

SYBASE®

リファレンス：ビルディング・ブロック、
テーブル、およびプロシージャ

Sybase IQ

15.1

ドキュメント ID : DC01135-01-1510-01

改訂 : 2009 年 7 月

Copyright © 1991-2009 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいエディションまたはテクニカル・ノートで特に示されない限り、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供され、使用や複製はこの契約に従って行う場合にのみ許可されます。

追加ドキュメントを注文する場合は、米国、カナダのお客様は、カスタマ・フルフィルメント事業部 (電話 800-685-8225、ファックス 617-229-9845) までご連絡ください。

米国のライセンス契約が適用されるその他の国のお客様は、上記のファックス番号でカスタマ・フルフィルメント事業部までご連絡ください。上記以外の国のお客様は、Sybase の関連会社または最寄りの販売代理店にお問い合わせください。アップグレードは定期ソフトウェア リリース日にのみ提供されます。このマニュアルの内容を弊社による事前許可を得ずに電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても複製、転載、翻訳することを禁じます。

Sybase の商標は、[Sybase の商標ページ \(http://www.sybase.com/detail?id=1011207\)](http://www.sybase.com/detail?id=1011207) で確認できます。Sybase および上記ページに記載されている商標は、Sybase, Inc の商標です。® は、米国で登録されていることを示します。

Java および Java 関連の商標は、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Unicode と Unicode のロゴは Unicode, Inc. の登録商標です。

このマニュアルに記載されているその他の社名および製品名は、当該各社の商標または登録商標である可能性があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

| | |
|-------------------------------------------------------------------|-------------------------------------|
| はじめに..... | xix |
| 第 1 章 | ファイル・ロケーションとインストール設定 1 |
| インストール・ディレクトリ構造 | 2 |
| Sybase IQ によるファイルの配置 | 3 |
| 簡単なファイル検索 | 5 |
| 広範囲なファイル検索 | 5 |
| 環境変数 | 6 |
| 環境変数の設定 | 6 |
| IQDIR15 環境変数 | 8 |
| IQPORT 環境変数 | 9 |
| IQLOGDIR15 環境変数 | 9 |
| IQTIMEOUT 環境変数 | 10 |
| IQTMP15 環境変数 | 10 |
| LIBPATH 環境変数と LD_LIBRARY_PATH 環境変数 | 12 |
| PATH 環境変数 | 12 |
| SACHARSET 環境変数 | 12 |
| SALANG 環境変数 | 13 |
| SQLCONNECT 環境変数 | 13 |
| SYBASE 環境変数 | 14 |
| \$SYBASE_JRE6_32、\$SYBASE_JRE6_64、 \$SYBASE_JRE5_64 環境変数 | 14 |
| SYBASE_OCS 環境変数 | 15 |
| レジストリ・エントリ | 15 |
| カレント・ユーザ設定とローカル・マシン設定 | 15 |
| レジストリ構造 | 16 |
| インストール時のレジストリ設定 | 17 |
| 第 2 章 | SQL 言語の要素 19 |
| キーワード | 19 |
| 予約語 | 20 |
| 識別子 | 22 |
| 文字列 | 24 |

| | |
|-----------------------|----|
| 式 | 25 |
| 式内の定数 | 26 |
| 式内のカラム名 | 26 |
| 式内のサブクエリ | 26 |
| SQL 演算子 | 27 |
| IF 式 | 30 |
| CASE 式 | 31 |
| 式の互換性 | 32 |
| 検索条件 | 34 |
| 比較条件 | 35 |
| 検索条件内のサブクエリ | 37 |
| ALL または ANY 条件 | 40 |
| BETWEEN 条件 | 41 |
| LIKE 条件 | 42 |
| IN 条件 | 44 |
| CONTAINS 条件 | 45 |
| EXISTS 条件 | 45 |
| IS NULL 条件 | 46 |
| 論理演算子を使用した条件 | 46 |
| NOT 条件 | 47 |
| 真理値条件 | 47 |
| 3 値的論理 | 47 |
| ユーザ指定の条件ヒント | 48 |
| 特別値 | 54 |
| CURRENT DATABASE 特別値 | 54 |
| CURRENT DATE 特別値 | 54 |
| CURRENT PUBLISHER 特別値 | 54 |
| CURRENT TIME 特別値 | 54 |
| CURRENT TIMESTAMP 特別値 | 55 |
| CURRENT USER 特別値 | 55 |
| LAST USER 特別値 | 55 |
| SQLCODE 特別値 | 56 |
| SQLSTATE 特別値 | 56 |
| TIMESTAMP 特別値 | 56 |
| USER 特別値 | 57 |
| 変数 | 57 |
| ローカル変数 | 58 |
| 接続レベル変数 | 59 |
| グローバル変数 | 60 |
| コメント | 65 |
| NULL 値 | 67 |

| | | |
|--------------|------------------------------------|------------|
| 第 3 章 | SQL データ型 | 69 |
| | 文字データ型 | 69 |
| | 数値データ型 | 73 |
| | バイナリ・データ型 | 77 |
| | Bit データ型 | 83 |
| | 日付と時刻のデータ型 | 83 |
| | 日付と時刻のデータベースへの送信 | 85 |
| | データベースから日付と時刻を取得する | 86 |
| | 日付と時刻の比較 | 86 |
| | あいまいさのない日付と時刻の使用 | 87 |
| | ドメイン | 89 |
| | データ型変換 | 90 |
| | Sybase IQ BINARY ロード形式 | 95 |
| | | |
| 第 4 章 | SQL 関数 | 103 |
| | 概要 | 104 |
| | 集合関数 | 104 |
| | 分析関数 | 106 |
| | データ型変換関数 | 110 |
| | 日付と時刻の関数 | 111 |
| | 日付要素 | 114 |
| | HTTP 関数 | 115 |
| | 数値関数 | 115 |
| | 文字列関数 | 117 |
| | システム関数 | 119 |
| | 接続プロパティ | 121 |
| | サーバで使用可能なプロパティ | 122 |
| | 各データベースで使用できるプロパティ | 122 |
| | SQL および Java のユーザ定義関数 | 123 |
| | 時系列関数および予測関数 | 124 |
| | 時系列関数および予測関数の IMSL ライブラリのロード | 125 |
| | その他の関数 | 127 |
| | アルファベット順の関数リスト | 128 |
| | ABS 関数 [数値] | 128 |
| | ACOS 関数 [数値] | 128 |
| | ARGN 関数 [その他] | 129 |
| | ASCII 関数 [文字列] | 129 |
| | ASIN 関数 [数値] | 130 |
| | ATAN 関数 [数値] | 130 |
| | ATAN2 関数 [数値] | 131 |
| | AVG 関数 [集合] | 131 |
| | BIGINTTOHEX 関数 [データ型変換] | 132 |
| | BIT_LENGTH 関数 [文字列] | 133 |
| | BYTE_LENGTH 関数 [文字列] | 133 |

| | |
|--------------------------------------|-----|
| CAST 関数 [データ型変換] | 134 |
| CEIL 関数 [数値] | 135 |
| CEILING 関数 [数値] | 135 |
| CHAR 関数 [文字列] | 136 |
| CHAR_LENGTH 関数 [文字列] | 136 |
| CHARINDEX 関数 [文字列] | 137 |
| COALESCE 関数 [その他] | 138 |
| COL_LENGTH 関数 [システム] | 138 |
| COL_NAME 関数 [システム] | 139 |
| CONNECTION_PROPERTY 関数 [システム] | 139 |
| CONVERT 関数 [データ型変換] | 140 |
| CORR 関数 [集合] | 142 |
| COS 関数 [数値] | 143 |
| COT 関数 [数値] | 144 |
| COVAR_POP 関数 [集合] | 144 |
| COVAR_SAMP 関数 [集合] | 145 |
| COUNT 関数 [集合] | 146 |
| CUME_DIST 関数 [ランク付け] | 147 |
| DATALength 関数 [システム] | 148 |
| DATE 関数 [日付および時刻] | 149 |
| DATEADD 関数 [日付および時刻] | 149 |
| DATECEILING 関数 [日付および時刻] | 150 |
| DATEDIFF 関数 [日付および時刻] | 152 |
| DATEFLOOR 関数 [日付および時刻] | 154 |
| DATEFORMAT 関数 [日付および時刻] | 156 |
| DATENAME 関数 [日付および時刻] | 157 |
| DATEPART 関数 [日付および時刻] | 157 |
| DATETIME 関数 [日付および時刻] | 158 |
| DATETIME 関数 [日付および時刻] | 160 |
| DAY 関数 [日付および時刻] | 160 |
| DAYNAME 関数 [日付および時刻] | 160 |
| DAYS 関数 [日付および時刻] | 161 |
| DB_ID 関数 [システム] | 162 |
| DB_NAME 関数 [システム] | 162 |
| DB_PROPERTY 関数 [システム] | 163 |
| DEGREES 関数 [数値] | 164 |
| DENSE_RANK 関数 [統計] | 164 |
| DIFFERENCE 関数 [文字列] | 165 |
| DOW 関数 [日付および時刻] | 166 |
| ERRORMSG 関数 [その他] | 167 |
| EVENT_CONDITION 関数 [システム] | 167 |
| EVENT_CONDITION_NAME 関数 [システム] | 169 |
| EVENT_PARAMETER 関数 [システム] | 169 |
| EXP 関数 [数値] | 170 |

| | |
|-------------------------------|-----|
| EXP_WEIGHTED_AVG 関数 [集合]..... | 171 |
| FIRST_VALUE 関数 [平均]..... | 172 |
| FLOOR 関数 [数値]..... | 174 |
| GETDATE 関数 [日付および時刻]..... | 175 |
| GRAPHICAL_PLAN 関数 [文字列]..... | 175 |
| GROUPING 関数 [集合]..... | 178 |
| GROUP_MEMBER 関数 [システム]..... | 178 |
| HEXTOBIGINT 関数 [データ型変換]..... | 179 |
| HEXTOINT 関数 [データ型変換]..... | 180 |
| HOUR 関数 [日付および時刻]..... | 181 |
| HOURS 関数 [日付および時刻]..... | 181 |
| HTML_DECODE 関数 [HTTP]..... | 182 |
| HTML_ENCODE 関数 [HTTP]..... | 183 |
| HTML_PLAN 関数 [文字列]..... | 184 |
| HTTP_DECODE 関数 [HTTP]..... | 186 |
| HTTP_ENCODE 関数 [HTTP]..... | 186 |
| HTTP_HEADER 関数 [HTTP]..... | 187 |
| HTTP_VARIABLE 関数 [HTTP]..... | 188 |
| IFNULL 関数 [その他]..... | 189 |
| INDEX_COL 関数 [システム]..... | 189 |
| INSERTSTR 関数 [文字列]..... | 190 |
| INTTOHEX 関数 [データ型変換]..... | 191 |
| ISDATE 関数 [日付および時刻]..... | 192 |
| ISNULL 関数 [その他]..... | 193 |
| ISNUMERIC 関数 [その他]..... | 193 |
| LAST_VALUE 関数 [集合]..... | 194 |
| LCASE 関数 [文字列]..... | 196 |
| LEFT 関数 [文字列]..... | 196 |
| LEN 関数 [文字列]..... | 197 |
| LENGTH 関数 [文字列]..... | 198 |
| LN 関数 [数値]..... | 198 |
| LOCATE 関数 [文字列]..... | 199 |
| LOG 関数 [数値]..... | 200 |
| LOG10 関数 [数値]..... | 201 |
| LOWER 関数 [文字列]..... | 201 |
| LTRIM 関数 [文字列]..... | 202 |
| MAX 関数 [集合]..... | 202 |
| MEDIAN 関数 [集合]..... | 203 |
| MIN 関数 [集合]..... | 204 |
| MINUTE 関数 [日付および時刻]..... | 205 |
| MINUTES 関数 [日付および時刻]..... | 205 |
| MOD 関数 [数値]..... | 206 |
| MONTH 関数 [日付および時刻]..... | 206 |
| MONTHNAME 関数 [日付および時刻]..... | 206 |

| | |
|--------------------------------------|-----|
| MONTHS 関数 [日付および時刻] | 207 |
| NEWID 関数 [その他] | 208 |
| NEXT_CONNECTION 関数 [システム] | 209 |
| NEXT_DATABASE 関数 [システム] | 210 |
| NEXT_HTTP_HEADER 関数 [HTTP] | 211 |
| NEXT_HTTP_VARIABLE 関数 [HTTP] | 211 |
| NOW 関数 [日付および時刻] | 212 |
| NTILE 関数 [統計] | 212 |
| NULLIF 関数 [その他] | 214 |
| NUMBER 関数 [その他] | 215 |
| OBJECT_ID 関数 [システム] | 216 |
| OBJECT_NAME 関数 [システム] | 216 |
| OCTET_LENGTH 関数 [文字列] | 217 |
| PATINDEX 関数 [文字列] | 217 |
| PERCENT_RANK 関数 [統計] | 219 |
| PERCENTILE_CONT 関数 [統計] | 220 |
| PERCENTILE_DISC 関数 [統計] | 223 |
| PI 関数 [数値] | 225 |
| POWER 関数 [数値] | 225 |
| PROPERTY 関数 [システム] | 225 |
| PROPERTY_DESCRIPTION 関数 [システム] | 226 |
| PROPERTY_NAME 関数 [システム] | 227 |
| PROPERTY_NUMBER 関数 [システム] | 228 |
| QUARTER 関数 [日付および時刻] | 228 |
| RADIANS 関数 [数値] | 229 |
| RAND 関数 [数値] | 229 |
| RANK 関数 [統計] | 230 |
| REGR_AVGX 関数 [集合] | 231 |
| REGR_AVGY 関数 [集合] | 232 |
| REGR_COUNT 関数 [集合] | 234 |
| REGR_INTERCEPT 関数 [集合] | 234 |
| REGR_R2 関数 [集合] | 236 |
| REGR_SLOPE 関数 [集合] | 237 |
| REGR_SXX 関数 [集合] | 238 |
| REGR_SXY 関数 [集合] | 239 |
| REGR_SYY 関数 [集合] | 240 |
| REMAINDER 関数 [数値] | 241 |
| REPEAT 関数 [文字列] | 242 |
| REPLACE 関数 [文字列] | 243 |
| REPLICATE 関数 [文字列] | 244 |
| REVERSE 関数 [文字列] | 245 |
| RIGHT 関数 [文字列] | 245 |
| ROUND 関数 [数値] | 246 |
| ROWID 関数 [その他] | 247 |

| | |
|-------------------------------------------|-----|
| RTRIM 関数 [文字列] | 248 |
| SECOND 関数 [日付および時刻] | 249 |
| SECONDS 関数 [日付および時刻] | 249 |
| SIGN 関数 [数値] | 250 |
| SIMILAR 関数 [文字列] | 250 |
| SIN 関数 [数値] | 251 |
| SORTKEY 関数 [文字列] | 251 |
| SOUNDEX 関数 [文字列] | 256 |
| SPACE 関数 [文字列] | 257 |
| SQRT 関数 [数値] | 257 |
| SQUARE 関数 [数値] | 257 |
| STDDEV 関数 [集合] | 258 |
| STDDEV_POP 関数 [集合] | 259 |
| STDDEV_SAMP 関数 [集合] | 260 |
| STR 関数 [文字列] | 261 |
| STR_REPLACE 関数 [文字列] | 262 |
| STRING 関数 [文字列] | 263 |
| STRTOUUID 関数 [文字列] | 264 |
| STUFF 関数 [文字列] | 265 |
| SUBSTRING 関数 [文字列] | 265 |
| SUM 関数 [集合] | 266 |
| SUSER_ID 関数 [システム] | 267 |
| SUSER_NAME 関数 [システム] | 267 |
| TAN 関数 [数値] | 268 |
| TODAY 関数 [日付および時刻] | 268 |
| TRIM 関数 [文字列] | 268 |
| TRUNCNUM 関数 [数値] | 269 |
| TS_ARMA_AR 関数 [時系列] | 270 |
| TS_ARMA_CONST 関数 [時系列] | 273 |
| TS_ARMA_MA 関数 [時系列] | 277 |
| TS_AUTOCORRELATION 関数 [時系列] | 280 |
| TS_AUTO_UNI_AR 関数 [時系列] | 284 |
| TS_BOX_COX_XFORM 関数 [時系列] | 287 |
| TS_DIFFERENCE 関数 [時系列] | 290 |
| TS_ESTIMATE_MISSING 関数 [時系列] | 294 |
| TS_LACK OF FIT 関数 [時系列] | 297 |
| TS_LACK OF FIT_P 関数 [時系列] | 300 |
| TS_MAX_ARMA_AR 関数 [時系列] | 304 |
| TS_MAX_ARMA_CONST 関数 [時系列] | 308 |
| TS_MAX_ARMA_LIKELIHOOD 関数 [時系列] | 311 |
| TS_MAX_ARMA_MA 関数 [時系列] | 315 |
| TS_OUTLIER_IDENTIFICATION 関数 [時系列] | 318 |
| TS_PARTIAL_AUTOCORRELATION 関数 [時系列] | 323 |
| TS_VWAP 関数 [時系列] | 326 |

| | | |
|--------------|----------------------------------------|------------|
| | UCASE 関数 [文字列]..... | 328 |
| | UPPER 関数 [文字列]..... | 329 |
| | USER_ID 関数 [システム]..... | 329 |
| | USER_NAME 関数[システム]..... | 330 |
| | UIDTOSTR 関数 [文字列]..... | 330 |
| | VAR_POP 関数 [集合]..... | 331 |
| | VAR_SAMP 関数 [集合]..... | 332 |
| | VARIANCE 関数 [集合]..... | 333 |
| | WEEKS 関数 [日付および時刻]..... | 334 |
| | WEIGHTED_AVG 関数 [集合]..... | 335 |
| | WIDTH_BUCKET 関数 [数値]..... | 337 |
| | YEAR 関数 [日付および時刻]..... | 338 |
| | YEARS 関数 [日付および時刻]..... | 339 |
| | YMD 関数 [日付および時刻]..... | 340 |
| 第 5 章 | 他の SQL 言語との違い | 341 |
| | Sybase IQ の機能..... | 342 |
| 第 6 章 | 物理的制限 | 345 |
| | サイズと数の制限 | 346 |
| 第 7 章 | システム・プロシージャ | 349 |
| | システム・プロシージャの概要..... | 350 |
| | ストアド・プロシージャの構文規則..... | 350 |
| | ストアド・プロシージャによりレポートされる統計情報に ついて..... | 351 |
| | システム・ストアド・プロシージャ..... | 351 |
| | sa_dependent_views プロシージャ..... | 352 |
| | sa_verify_password プロシージャ..... | 352 |
| | sa_get_user_status システム・プロシージャ..... | 352 |
| | sp_expireallpasswords プロシージャ..... | 352 |
| | sp_iqaddlogin プロシージャ..... | 353 |
| | sp_iqbackupdetails プロシージャ..... | 354 |
| | sp_iqbackupsummary プロシージャ..... | 356 |
| | sp_iqcheckdb プロシージャ..... | 357 |
| | sp_iqcheckoptions プロシージャ..... | 365 |
| | sp_iqclient_lookup プロシージャ..... | 367 |
| | sp_iqcolumn プロシージャ..... | 369 |
| | sp_iqcolumnuse プロシージャ..... | 371 |
| | sp_iqconnection プロシージャ..... | 372 |
| | sp_iqconstraint プロシージャ..... | 376 |
| | sp_iqcontext プロシージャ..... | 377 |

| | |
|--------------------------------------------|-----|
| sp_iqcopyloginpolicy プロシージャ | 379 |
| sp_iqcursorinfo プロシージャ | 380 |
| sp_iqdatatype プロシージャ | 382 |
| sp_iqdbsize プロシージャ | 385 |
| sp_iqdbspace プロシージャ | 387 |
| sp_iqdbspaceinfo プロシージャ | 389 |
| sp_iqdbspaceobjectinfo プロシージャ | 393 |
| sp_iqdbstatistics プロシージャ | 396 |
| sp_iqdroplogin プロシージャ | 398 |
| sp_iqemptyfile プロシージャ | 398 |
| sp_iqestjoin プロシージャ | 399 |
| sp_iqestdbspaces プロシージャ | 401 |
| sp_iqestspace プロシージャ | 403 |
| sp_iqevent プロシージャ | 403 |
| sp_iqfile プロシージャ | 406 |
| sp_iqhelp プロシージャ | 408 |
| sp_iqindex および sp_iqindex_alt プロシージャ | 415 |
| sp_iqindexadvice プロシージャ | 418 |
| sp_iqindexfragmentation プロシージャ | 419 |
| sp_iqindexinfo プロシージャ | 421 |
| sp_iqindexmetadata プロシージャ | 423 |
| sp_iqindexsize プロシージャ | 424 |
| sp_iqindexuse プロシージャ | 426 |
| sp_iqjoinindex プロシージャ | 427 |
| sp_iqjoinindexsize プロシージャ | 430 |
| sp_iqlmconfig プロシージャ | 431 |
| sp_iqlocks プロシージャ | 432 |
| sp_iqmodifyadmin プロシージャ | 435 |
| sp_iqmodifylogin プロシージャ | 436 |
| sp_iqobjectinfo プロシージャ | 436 |
| sp_iqpassword プロシージャ | 439 |
| sp_iqpkeys プロシージャ | 440 |
| sp_iqprocedure プロシージャ | 442 |
| sp_iqprocparm プロシージャ | 445 |
| sp_iqrebuildindex プロシージャ | 448 |
| sp_iqrename プロシージャ | 451 |
| sp_iq_reset_identity プロシージャ | 453 |
| sp_iqrestoreaction プロシージャ | 453 |
| sp_iqrowdensity プロシージャ | 454 |
| sp_iqshowpsexec プロシージャ | 456 |
| sp_iqspaceinfo プロシージャ | 458 |
| sp_iqspaceused プロシージャ | 459 |
| sp_iqstatistics プロシージャ | 460 |
| sp_iqstatus プロシージャ | 462 |

| | |
|---------------------------------------------------------------|-----|
| sp_iqsysmon プロシージャ | 465 |
| sp_iqtable プロシージャ | 471 |
| sp_iqtablesize プロシージャ | 473 |
| sp_iqtableuse プロシージャ | 475 |
| sp_iqtransaction プロシージャ | 476 |
| sp_iqunusedcolumn プロシージャ | 479 |
| sp_iqunusedindex プロシージャ | 481 |
| sp_iqunusedtable プロシージャ | 482 |
| sp_iqversionuse プロシージャ | 483 |
| sp_iqview プロシージャ | 484 |
| sp_iqwho プロシージャ | 486 |
| sp_iqworkmon プロシージャ | 489 |
| カタログ・ストアド・プロシージャ | 491 |
| sa_audit_string システム・プロシージャ | 491 |
| sa_checkpoint_execute システム・プロシージャ | 491 |
| sa_conn_activity システム・プロシージャ | 492 |
| sa_conn_info システム・プロシージャ | 493 |
| sa_conn_properties システム・プロシージャ | 494 |
| sa_db_info システム・プロシージャ | 495 |
| sa_db_properties システム・プロシージャ | 496 |
| sa_enable_auditing_type システム・プロシージャ | 497 |
| sa_eng_properties システム・プロシージャ | 498 |
| sa_table_page_usage システム・プロシージャ | 499 |
| sa_disable_auditing_type システム・プロシージャ | 500 |
| sa_flush_cache システム・プロシージャ | 500 |
| sa_make_object システム・プロシージャ | 501 |
| sa_rowgenerator システム・プロシージャ | 502 |
| sa_server_option システム・プロシージャ | 503 |
| sa_set_http_header システム・プロシージャ | 510 |
| sa_set_http_option システム・プロシージャ | 511 |
| sa_validate システム・プロシージャ | 511 |
| sa_verify_password システム・プロシージャ | 513 |
| sp_login_environment システム・プロシージャ | 513 |
| sp_remote_columns システム・プロシージャ | 513 |
| sp_remote_exported_keys システム・プロシージャ | 514 |
| sp_remote_imported_keys システム・プロシージャ | 515 |
| sp_remote_primary_keys システム・プロシージャ | 516 |
| sp_remote_tables システム・プロシージャ | 517 |
| sp_servercaps システム・プロシージャ | 518 |
| sp_tsql_environment システム・プロシージャ | 520 |
| Adaptive Server Enterprise のシステム・プロシージャと カタログ・プロシージャ | 521 |
| Adaptive Server Enterprise システム・プロシージャ | 522 |
| Adaptive Server Enterprise カタログ・プロシージャ | 523 |
| SQL Anywhere でサポートされているプロシージャ | 525 |

第 8 章

| | |
|---------------------------------------|-----|
| システム・ビュー | 527 |
| SYSARTICLE システム・ビュー | 532 |
| SYSARTICLECOL システム・ビュー | 532 |
| SYSARTICLECOLS 統合ビュー | 532 |
| SYSARTICLES 統合ビュー | 533 |
| SYSCAPABILITIES 統合ビュー | 533 |
| SYSCAPABILITY システム・ビュー | 533 |
| SYSCAPABILITYNAME システム・ビュー | 534 |
| SYSCATALOG 統合ビュー | 534 |
| SYSCHECK システム・ビュー | 534 |
| SYSCOLAUTH 統合ビュー | 535 |
| SYSCOLPERM システム・ビュー | 535 |
| SYSCOLLATION 互換ビュー (廃止) | 535 |
| SYSCOLLATIONMAPPINGS 互換ビュー (廃止) | 536 |
| SYSCOLSTAT システム・ビュー | 536 |
| SYSCOLSTATS 統合ビュー | 537 |
| SYSCOLUMN 互換ビュー (廃止) | 537 |
| SYSCOLUMNS 統合ビュー | 537 |
| SYSCOLUMNS ASE 互換ビュー | 538 |
| SYSCOMMENTS ASE 互換ビュー | 538 |
| SYSCONSTRAINT システム・ビュー | 538 |
| SYSDBFILE システム・ビュー | 539 |
| SYSDBSPACE システム・ビュー | 539 |
| SYSDBSPACEPERM システム・ビュー | 539 |
| SYSDEPENDENCY システム・ビュー | 540 |
| SYSDOMAIN システム・ビュー | 540 |
| SYSEVENT システム・ビュー | 541 |
| SYSEVENTTYPE システム・ビュー | 541 |
| SYSEXTERNENV システム・ビュー | 541 |
| SYSEXTERNENVOBJECT システム・ビュー | 542 |
| SYSEXTERNLOGIN システム・ビュー | 542 |
| SYSFILE 互換ビュー (廃止) | 543 |
| SYSFKCOL 互換ビュー (廃止) | 543 |
| SYSFKEY システム・ビュー | 543 |
| SYSFOREIGNKEY 互換ビュー (廃止) | 544 |
| SYSFOREIGNKEYS 統合ビュー | 544 |
| SYSGROUP システム・ビュー | 544 |
| SYSGROUPS 統合ビュー | 545 |
| SYSHISTORY システム・ビュー | 545 |
| SYSIDX システム・ビュー | 546 |
| SYSIDXCOL システム・ビュー | 546 |
| SYSINDEX 互換ビュー (廃止) | 546 |
| SYSINDEXES 統合ビュー | 547 |
| SYSINDEXES ASE 互換ビュー | 547 |

| | |
|-----------------------------------------|-----|
| SYSINFO 互換ビュー (廃止)..... | 548 |
| SYSIQBACKUPHISTORY システム・ビュー..... | 548 |
| SYSIQBACKUPHISTORYDETAIL システム・ビュー..... | 549 |
| SYSIQCOLUMN システム・ビュー (廃止)..... | 551 |
| SYSIQDBFILE システム・ビュー..... | 551 |
| SYSIQDBSPACE システム・ビュー..... | 552 |
| SYSIQFILE システム・ビュー (廃止)..... | 553 |
| SYSIQIDX システム・ビュー..... | 553 |
| SYSIQINFO システム・ビュー..... | 555 |
| SYSIQJOINIDX システム・ビュー..... | 556 |
| SYSIQJOININDEX システム・ビュー (廃止)..... | 557 |
| SYSIQJOINIXCOLUMN システム・ビュー..... | 557 |
| SYSIQJOINIXTABLE システム・ビュー..... | 559 |
| SYSIQMPXLOGINPOLICYOPTION システム・ビュー..... | 559 |
| SYSIQMPXSERVER システム・ビュー..... | 560 |
| SYSIQOBJECTS ASE 互換ビュー..... | 560 |
| SYSIQPARTITIONCOLUMN システム・ビュー..... | 560 |
| SYSIQTAB システム・ビュー..... | 561 |
| SYSIQTABCOL システム・ビュー..... | 562 |
| SYSIQTABLE システム・ビュー (廃止)..... | 563 |
| SYSIQVINDEXTABLE ASE 互換ビュー..... | 563 |
| SYSIXCOL 互換ビュー (廃止)..... | 564 |
| SYSJAR システム・ビュー..... | 564 |
| SYSJARCOMPONENT システム・ビュー..... | 564 |
| SYSJAVACLASS システム・ビュー..... | 565 |
| SYSLOGINMAP システム・ビュー..... | 565 |
| SYSLOGINPOLICY システム・ビュー..... | 565 |
| SYSLOGINPOLICYOPTION システム・ビュー..... | 566 |
| SYSLOGINS ASE 互換ビュー..... | 566 |
| SYSMVOPTION システム・ビュー..... | 566 |
| SYSMVOPTIONNAME システム・ビュー..... | 566 |
| SYSOBJECT システム・ビュー..... | 567 |
| SYSOBJECTS ASE 互換ビュー..... | 567 |
| SYSOPTION システム・ビュー..... | 568 |
| SYSOPTIONS 統合ビュー..... | 568 |
| SYSOPTSTAT システム・ビュー..... | 568 |
| SYSPARTITION システム・ビュー..... | 569 |
| SYSPARTITIONKEY システム・ビュー..... | 570 |
| SYSPARTITIONSCHEME システム・ビュー..... | 570 |
| SYSPHYSIDX システム・ビュー..... | 571 |
| SYSPROCAUTH 統合ビュー..... | 571 |
| SYSPROCEDURE システム・ビュー..... | 572 |
| SYSPROCPARM システム・ビュー..... | 572 |
| SYSPROCPARMS 統合ビュー..... | 572 |

| | |
|----------------------------------------|-----|
| SYSPROCPERM システム・ビュー | 573 |
| SYSPROCS 統合ビュー | 573 |
| SYSPROXYTAB システム・ビュー..... | 573 |
| SYSPUBLICATION システム・ビュー | 574 |
| SYSPUBLICATIONS 統合ビュー..... | 574 |
| SYSREMARK システム・ビュー..... | 574 |
| SYSREMOTEOPTION システム・ビュー | 575 |
| SYSREMOTEOPTION2 統合ビュー..... | 575 |
| SYSREMOTEOPTIONS 統合ビュー | 575 |
| SYSREMOTEOPTIONTYPE システム・ビュー..... | 576 |
| SYSREMOTETYPE システム・ビュー | 576 |
| SYSREMOTETYPES 統合ビュー..... | 576 |
| SYSREMOTEEUSER システム・ビュー..... | 577 |
| SYSREMOTEEUSERS 統合ビュー..... | 577 |
| SYSSCHEDULE システム・ビュー..... | 578 |
| SYSSERVER システム・ビュー | 578 |
| SYSSOURCE システム・ビュー..... | 578 |
| SYSSQLSERVERTYPE システム・ビュー..... | 579 |
| SYSSUBPARTITIONKEY システム・ビュー | 579 |
| SYSSUBSCRIPTION システム・ビュー | 579 |
| SYSSUBSCRIPTIONS 統合ビュー..... | 579 |
| SYSSYNC システム・ビュー | 580 |
| SYSSYNC2 統合ビュー..... | 580 |
| SYSSYNCPUBLICATIONDEFAULTS 統合ビュー | 581 |
| SYSSYNCS 統合ビュー..... | 581 |
| SYSSYNCSSCRIPT システム・ビュー | 581 |
| SYSSYNCSSCRIPTS 統合ビュー | 582 |
| SYSSYNCSUBSCRIPTIONS 統合ビュー..... | 582 |
| SYSSYNCSUSERS 統合ビュー..... | 582 |
| SYSTAB システム・ビュー | 583 |
| SYSTABLE 互換ビュー (廃止)..... | 583 |
| SYSTABAUTH 統合ビュー | 583 |
| SYSTABCOL システム・ビュー..... | 584 |
| SYSTABLEPERM システム・ビュー | 584 |
| SYSTEXTCONFIG システム・ビュー | 584 |
| SYSTEXTIDX システム・ビュー | 585 |
| SYSTEXTIDXTAB システム・ビュー | 585 |
| SYSTRIGGER システム・ビュー..... | 585 |
| SYSTRIGGERS 統合ビュー..... | 586 |
| SYSTYPEMAP システム・ビュー | 586 |
| SYSTYPES ASE 互換ビュー | 586 |
| SYSUSER システム・ビュー | 587 |
| SYSUSERAUTH 互換ビュー (廃止)..... | 587 |
| SYSUSERAUTHORITY システム・ビュー | 587 |

| | | |
|--------------|-----------------------------------------------------|------------|
| | SYSUSERLIST 互換ビュー (廃止)..... | 588 |
| | SYSUSERMESSAGE システム・ビュー | 588 |
| | SYSUSEROPTIONS 統合ビュー..... | 588 |
| | SYSUSERPERM 互換ビュー (廃止)..... | 589 |
| | SYSUSERPERMS 互換ビュー (廃止)..... | 589 |
| | SYSUSERTYPE システム・ビュー | 589 |
| | SYSUSERS ASE 互換ビュー | 590 |
| | SYSVIEW システム・ビュー | 590 |
| | SYSVIEWS 統合ビュー | 590 |
| | SYSWEBSERVICE システム・ビュー | 591 |
| | Transact-SQL 互換のビュー | 591 |
| 第 9 章 | システム・テーブル | 597 |
| | システム・テーブル・リスト | 597 |
| | DUMMY システム・テーブル | 600 |
| 付録 A | 他の Sybase データベースとの互換性 | 601 |
| | Transact-SQL サポートの概要..... | 602 |
| | Adaptive Server のアーキテクチャ..... | 604 |
| | サーバとデータベース | 604 |
| | 領域の割り付けとデバイス管理..... | 605 |
| | システム・テーブル、カタログ・ストア、IQ ストア | 606 |
| | 管理者の役割..... | 607 |
| | データ型..... | 608 |
| | Bit データ型..... | 608 |
| | 文字データ型..... | 609 |
| | バイナリ・データ型..... | 610 |
| | date、time、datetime、timestamp データ型..... | 611 |
| | 数値データ型..... | 612 |
| | 概数値データ型..... | 613 |
| | text データ型 | 613 |
| | image データ型..... | 614 |
| | Java データ型..... | 614 |
| | データ定義言語..... | 614 |
| | Transact-SQL と互換性のあるデータベースの作成 | 614 |
| | 大文字と小文字の区別 | 615 |
| | オブジェクト名の互換性..... | 616 |
| | CREATE TABLE 文..... | 617 |
| | CREATE DEFAULT、CREATE RULE、 CREATE DOMAIN 文..... | 620 |
| | CREATE TRIGGER 文..... | 621 |
| | CREATE INDEX 文 | 621 |
| | ユーザ、グループ、パーミッション..... | 622 |

| | |
|------------------------------------------------------|------------|
| ロード形式 | 625 |
| Transact-SQL との互換性を維持するためのオプション 設定 | 625 |
| データ操作言語 | 625 |
| 移植可能な SQL を書くための全般的なガイドライン | 625 |
| 互換性のあるクエリの書き方 | 626 |
| サブクエリ | 627 |
| GROUP BY 句 | 627 |
| COMPUTE 句 | 628 |
| WHERE 句 | 628 |
| ジョイン | 629 |
| Null 比較 | 630 |
| 長さがゼロの文字列 | 630 |
| HOLDLOCK、SHARED、FOR BROWSE | 631 |
| SQL 関数 | 631 |
| OLAP 関数 | 632 |
| システム関数 | 634 |
| ユーザ定義関数 | 634 |
| 日付の算術式 | 635 |
| SELECT INTO | 635 |
| 更新可能なビュー | 635 |
| UPDATE と DELETE の FROM 句 | 636 |
| Transact-SQL のプロシージャ言語の概要 | 636 |
| Transact-SQL のストアド・プロシージャの概要 | 636 |
| Transact-SQL のバッチの概要 | 637 |
| プロシージャとバッチ内の SQL 文 | 637 |
| ストアド・プロシージャの自動変換 | 639 |
| Transact-SQL プロシージャから結果セットを返す | 640 |
| Transact-SQL プロシージャ内の変数 | 641 |
| Transact-SQL プロシージャでのエラー処理 | 641 |
| プロシージャの中での RAISERROR 文の使用 | 643 |
| Watcom-SQL ダイアレクトでの Transact-SQL のような エラー処理 | 643 |
| SQL Anywhere と Sybase IQ | 644 |
| サーバとデータベースの起動と管理 | 644 |
| データベース・オプション | 645 |
| データ定義言語 (DDL) | 645 |
| データ操作言語 (DML) | 646 |
| ストアド・プロシージャ | 647 |
| Adaptive Server Enterprise と Sybase IQ | 647 |
| ストアド・プロシージャ | 648 |
| システム・ビュー | 648 |
| 索引 | 651 |

はじめに

このマニュアルの内容

このマニュアルでは、SQL 文、言語要素、データ型、関数、システム・プロシージャ、システム・テーブルなど、Sybase IQ のさまざまな側面に関するリファレンス資料を提供します。他のマニュアルでは、特定のタスクを実行する方法を詳しく説明します。このリファレンス・マニュアルは、使用可能な SQL 構文、パラメータ、オプションなどの情報を参照するためのものです。コマンドライン・ユーティリティの起動パラメータについては、『ユーティリティ・ガイド』を参照してください。

対象読者

このマニュアルは、すべての Sybase IQ ユーザを対象読者とするリファレンスです。

関連マニュアル

Sybase IQ 15.1 マニュアル・セットの構成は次のとおりです。

- 『リリース・ノート』では、製品およびマニュアルに加えられた最新の変更内容について説明しています。
- 『インストールおよび設定ガイド』では、Sybase IQ のインストール手順、新バージョンの Sybase IQ へのマイグレート、特定のプラットフォームでの Sybase IQ の設定について説明しています。
- 『Sybase IQ による高度なセキュリティ』は、Sybase IQ データ・レポジトリ内でのユーザによるカラムの暗号化の使用について説明しています。このオプションの製品をインストールするには、別のライセンスが必要です。
- 『エラー・メッセージ』は、Sybase IQ エラー・メッセージ (Sybase エラー・コード、SQLCode、SQLState、および SQL プリプロセッサのエラーと警告) を示します。
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド: 第 2/2 巻 C 統計ライブラリ』には、IMSL C 統計ライブラリの時系列 C 関数の簡潔な説明が記載されています。このマニュアルは、RAP — The Trading Edition™ Enterprise ユーザにのみ提供されています。
- 『Sybase IQ の概要』は、Sybase IQ や Sybase Central™ データベース管理ツールの操作に慣れていない場合に参照してください。実際に操作の練習ができます。

-
- 『Sybase IQ によるラージ・オブジェクト管理』は、Sybase IQ データ・レポジトリ内での BLOB (Binary Large Object) および CLOB (Character Large Object) の格納と取得について説明しています。このオプションの製品をインストールするには、別のライセンスが必要です。
 - 『Sybase IQ 15.0 の新機能』には、バージョン 15.0 の新機能と動作変更が記載されています。
 - 『新機能の概要 Sybase IQ 15.1』には、最新バージョンの新機能と動作変更がまとめられています。
 - 『パフォーマンス&チューニング・ガイド』では、巨大なデータベースのクエリ最適化、設計、チューニングについて説明しています。
 - 『クイック・スタート』には、Sybase IQ のソフトウェア・インストールを検証するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う手順が記載されています。デモ・データベースをマルチプレックスに変換するための情報も記載されています。
 - 『リファレンス・マニュアル』— Sybase IQ の 2 冊のリファレンス・ガイドで構成されています。
 - 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』(本書)では、Sybase IQ でサポートされる SQL、ストアド・プロシージャ、データ型、およびシステム・テーブルについて説明しています。
 - 『リファレンス：文とオプション』は、Sybase IQ でサポートされる SQL 文およびオプションについて説明します。
 - 『システム管理ガイド』— 2 巻構成です。
 - 『システム管理ガイド 第 1 巻』では、起動、接続、データベース作成、自動入力とインデックス作成、バージョン設定、照合、システムのバックアップとリカバリ、トラブルシューティング、およびデータベースの修復方法について説明しています。
 - 『システム管理ガイド 第 2 巻』では、プロシージャとバッチの作成および実行、OLAP でのプログラミング、リモート・データへのアクセス、Open Server としての IQ の設定、スケジューリングとイベント処理、XML でのプログラミング、およびデバッグについて説明しています。
 - 『ユーザ定義関数ガイド』には、ユーザ定義関数、パラメータ、および考えられる使用事例が記載されています。

- 『Sybase IQ マルチプレックスの使用』では、複数のノードにまたがって発生する大きなクエリの負荷を管理するために設計されているマルチプレックス機能の使用方法について説明しています。
- 『ユーティリティ・ガイド』では、Sybase IQ ユーティリティ・プログラムのリファレンス項目 (使用可能な構文、パラメータ、オプションなど) について説明しています。

Sybase IQ および SQL Anywhere

Sybase IQ は SQL Anywhere を拡張したものであり、SQL Anywhere® パッケージの 1 コンポーネントです。そのため Sybase IQ は SQL Anywhere の多くの同じ機能をサポートします。Sybase IQ のマニュアル・セットには、SQL Anywhere のマニュアルの該当する箇所が記載されています。

SQL Anywhere には、次のマニュアルがあります。

- 『SQL Anywhere Server — データベース管理ガイド』では、SQL Anywhere データベースの実行、管理、構成方法について説明します。データベース接続、データベース・サーバ、データベース・ファイル、バックアップ手順、セキュリティ、高可用性、Replication Server® での複製、管理ユーティリティおよびオプションについても説明します。
- 『SQL Anywhere Server — プログラミング・ガイド』では、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET プログラミング言語を使用したデータベース・アプリケーションの構築および展開方法について説明します。このマニュアルでは、ADO.NET や ODBC などの各種プログラミング・インタフェースについても説明します。
- 『SQL Anywhere Server — SQL リファレンス・マニュアル』には、システム・プロシージャおよびカタログ (システム・テーブルおよびビュー) に関する参照情報が記載されています。また、SQL 言語の SQL Anywhere の実装 (検索条件、構文、データ型、関数) についても説明します。
- 『SQL Anywhere サーバー SQL の使用法』では、データベースの設計／作成方法、データのインポート／エクスポート／変更方法、データの検索方法、およびストアド・プロシージャとトリガの作成方法について説明します。

Product Manuals (<http://www.sybase.com/support/manuals/>) の SQL Anywhere 11.0.1 コレクション、および DocCommentXchange (http://dcx.sybase.com/dcx_home.php) の SQL Anywhere マニュアルを参照することもできます。

Sybase ソフトウェア資産管理 (SySAM) には、次のマニュアルがあります。

- 『Sybase ソフトウェア資産管理 (SySAM) 2』では資産管理の概念を紹介し、SySAM 2 ライセンスの設定および管理方法について説明します。
- 『SySAM 2 クイック・スタート・ガイド』は、SySAM 対応の Sybase 製品を実行する方法について説明します。
- 『FLEXnet ライセンス・エンド・ユーザ・ガイド』では、管理者およびエンド・ユーザ向けに FLEXnet ライセンスについて説明し、Sybase から販売される標準的な FLEXnet ライセンス配布キットに含まれているツールの使用方法について説明しています。

その他の情報

Sybase Getting Started CD、SyBooks™ CD、Sybase Product Manuals Web サイトを利用すると、製品について詳しく知ることができます。

- Getting Started CD には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていないその他のマニュアルや更新情報が収録されています。この CD は製品のソフトウェアに同梱されています。Getting Started CD に収録されているマニュアルを参照または印刷するには、Adobe Acrobat Reader が必要です (CD 内のリンクを使用して Adobe の Web サイトから無料でダウンロードできます)。
- SyBooks CD には製品マニュアルが収録されています。この CD は製品のソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザでは、使いやすい HTML 形式のマニュアルにアクセスできます。

一部のマニュアルは PDF 形式で提供されています。それらのマニュアルは SyBooks CD の PDF ディレクトリに収録されています。PDF ファイルを開いたり印刷したりするには、Adobe Acrobat Reader が必要です。

SyBooks のインストールと起動の方法については、Getting Started CD の『SyBooks インストール・ガイド』、または SyBooks CD の *README.txt* ファイルを参照してください。

- Sybase Product Manuals Web サイトは、SyBooks CD のオンライン版であり、標準の Web ブラウザを使ってアクセスできます。また、製品マニュアルのほか、EBFs/Maintenance、Technical Documents、Case Management、Solved Cases、ニュース・グループ、Sybase Developer Network へのリンクもあります。

Sybase Product Manuals Web サイトは、[Product Manuals \(http://sybooks.sybase.com\)](http://sybooks.sybase.com) にあります。

Web 上の Sybase 製品の動作確認情報

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ 製品動作確認の最新情報にアクセスする

- 1 Web ブラウザで [Technical Documents \(http://certification.sybase.com/ucr/search.do\)](http://certification.sybase.com/ucr/search.do) を指定します。
- 2 [Search By Base Product] で製品ファミリーとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。
- 3 [Search] をクリックして、入手状況と動作確認レポートを表示します。

❖ コンポーネント動作確認の最新情報にアクセスする

- 1 Web ブラウザで [Availability and Certification Reports \(http://certification.sybase.com/\)](http://certification.sybase.com/) を指定します。
- 2 [Search By Base Product] で製品ファミリーとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。
- 3 [Search] をクリックして、入手状況と動作確認レポートを表示します。

❖ Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

- 1 Web ブラウザで [Technical Documents \(http://www.sybase.com/support/techdocs/\)](http://www.sybase.com/support/techdocs/) を指定します。
- 2 [MySybase] をクリックし、MySybase プロファイルを作成します。

❖ **EBF とソフトウェア・メンテナンスの最新情報にアクセスする**

- 1 Web ブラウザで [Sybase Support Page \(http://www.sybase.com/support\)](http://www.sybase.com/support) を指定します。
- 2 [EBFs/Maintenance] を選択します。ユーザ名とパスワードの入力が求められたら、MySybase のユーザ名とパスワードを入力します。
- 3 製品を選択します。
- 4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

鍵のアイコンは、自分が Technical Support Contact として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録ではあるが、Sybase 担当者またはサポート・コンタクトから有効な情報を得ている場合は、[Edit Roles] をクリックして、「Technical Support Contact」役割を MySybase プロファイルに追加します。

- 5 EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

SQL 構文の表記規則

このマニュアルでは、構文の説明に次の表記規則を使用します。

- **キーワード** SQL キーワードは大文字で示します。ただし、SQL キーワードは大文字と小文字の区別がないので、入力するときはどちらで入力してもかまいません。たとえば、SELECT は Select でも select でも同じです。
- **プレースホルダ** 適切な識別子または式で置き換えられる項目は、斜体で表記します。
- **継続** 省略記号 (...) で始まる行は、前の行から文が続いていることを表します。
- **繰り返し項目** 繰り返し項目のリストは、リストの要素の後ろに省略記号 (...) を付けて表します。1 つまたは複数の要素を指定できます。複数の要素を指定する場合は、各要素間はカンマで区切る必要があります。
- **オプション部分** 文のオプション指定部分は、角カッコで囲みます。例を示します。

```
RELEASE SAVEPOINT [ savepoint-name ]
```


この例では、*savepoint-name* がオプション部分です。大カッコは入力しないでください。

- **オプション** 項目リストから1つだけ選択しなければならない場合、また何も選択する必要のない場合は、項目間を縦線で区切り、リスト全体を角カッコで囲みます。次に例を示します。

[ASC | DESC]

この例では、ASC と DESC のどちらか1つを選択するか、どちらも選択しないことができます。大カッコは入力しないでください。

- **その他の方法** オプションの中の1つを必ず選択しなければならない場合は、選択肢を中カッコ { } で囲みます。例を示します。

QUOTES { ON | OFF }

中カッコは、ON か OFF のいずれかを含めなければいけないことを示します。大カッコは入力しないでください。

書体の表記規則

表 1 に、このマニュアルで使用している書体の表記規則を示します。

表 1：書体の表記規則

| 項目 | 説明 |
|-------------------|----------------------------------------------------------------------------|
| Code | SQL およびプログラム・コードは等幅 (固定幅) 文字フォントで表記します。 |
| User entry | ユーザが入力するテキストには等幅 (固定幅) 文字フォントを使用します。 |
| 「強調」 | 強調する言葉は「」で囲みます。 |
| <i>file names</i> | ファイル名は斜体で表記します。 |
| database objects | テーブル、プロシージャなどのデータベース・オブジェクトの名前は、印刷物では太字の sans serif フォントで、オンラインでは斜体で表記します。 |

デモ・データベース

Sybase IQ にはデモ・データベース (*iqdemo.db*) を作成するためのスクリプトが用意されています。このマニュアルで紹介している多くのクエリおよびコード例は、このデモ・データベースをデータ・ソースに使用しています。

デモ・データベースは、小規模会社の内部情報 (従業員、部署、財務データ) に加えて、製品と販売情報 (注文、顧客、担当者) で構成されています。

デモ・データベースの詳細については、使用しているプラットフォームの『Sybase IQ インストール・ガイド』を参照するか、システム管理者に相談してください。

アクセシビリティ機能

このマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示する方法により、その内容を理解できるよう配慮されています。

Sybase IQ 15.1 とその HTML マニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

アクセシビリティ・ツールの設定

アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (MixedCase Text など) は単語として発音します。構文規則を発音するようにツールを設定することをおすすめします。スクリーン・リーダーの使用方法については、使用しているツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、[Sybase Accessibility \(http://www.sybase.com/accessibility\)](http://www.sybase.com/accessibility) を参照してください。Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報のリンクもあります。

Sybase IQ の第 508 条準拠の声明については、[Sybase Accessibility \(http://www.sybase.com/products/accessibility\)](http://www.sybase.com/products/accessibility) を参照してください。

不明な点があるときは

サポート契約を購入済みの Sybase 製品のインストールには、定められた 1 人以上のユーザに対して、Sybase 製品の保守契約を結んでいるサポート・センタを利用する権利が付属します。マニュアルやオンライン・ヘルプで解決できない問題がある場合は、この担当者を通して最寄りの Sybase のサポート・センタまでご連絡ください。

ファイル・ロケーションとインストール設定

この章について

この章では、Sybase IQ で使用するインストール環境とオペレーティング・システムの設定について説明します。これらの設定は、オペレーティング・システムに応じて、環境変数、初期化ファイル・エントリ、またはレジストリ・エントリとして保管できます。

内容

| トピック名 | ページ |
|--------------------------------------|-----|
| インストール・ディレクトリ構造 | 2 |
| Sybase IQ によるファイルの配置 | 3 |
| 環境変数 | 6 |
| レジストリ・エントリ | 15 |

インストール・ディレクトリ構造

Sybase IQ をインストールするとき、いくつかのディレクトリが作成されます。作成されるディレクトリは、インストール時に選択したオプションと Sybase ディレクトリ内の既存のディレクトリ (UNIX では `$SYBASE` によって定義されたディレクトリ、Windows では `%SYBASE%` によって定義されたディレクトリ) によって決まります。この項ではディレクトリ構造について説明します。

デフォルトでは、Sybase IQ ソフトウェアは Sybase ディレクトリの下にユニークなサブディレクトリにインストールされます。このサブディレクトリは、インストール・ディレクトリと呼ばれます。Sybase IQ に付随する他のツールは、同様に Sybase ディレクトリの下にユニークなサブディレクトリを持ちます。この項では、Sybase IQ のサブディレクトリ構造についてのみ説明します。

Sybase IQ ディレクトリ

デフォルトでは、Sybase IQ ディレクトリは `IQ-15_1` です。`IQ-15_1` のロケーションは、Sybase IQ のインストール場所によって変わります。`IQ-15_1` ディレクトリは、UNIX の環境変数 `$IQDIR15` または Windows の環境変数 `%IQDIR15%` によっても参照されます。

Sybase IQ ディレクトリ内には、多くのディレクトリとファイルがあります。

- デモ・ディレクトリ (`%SYBASE%/SybaseIQ/demo`) — `iqdemo` データベースの構築に必要なツールがあります。`iqdemo` データベース・ファイルとして、`iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg`、および `iqdemo.iqtmp` があります。デモ・データベース自体は Sybase IQ に同梱されていません。

サブディレクトリ `/demo/adata` には、15.x の `iqdemo` データベースを作成するための 15.x データがあります。サブディレクトリ `/demo/demodata` には、IQ 12.7 `asiqdemo` データベースと同じスキーマ・レイアウトとデータを持つ `iqdemo` データベースを作成するための Sybase IQ 12.7 データがあります。15.x の `iqdemo` データベースを作成するには、Windows では `/demo/mkiqdemo.bat` を使用し、Unix では `demo/mkiqdemo.sh` を使用します。`iqdemo` データベースは、サポート・センタに問題を伝えるために使用できます。

- スクリプト・ディレクトリ (`IQ-15_1/scripts`) — 例として使用するスクリプトやストアド・プロシージャのようなカタログ・オブジェクト作成時に使用されるスクリプトがあります。これらのスクリプトは編集しないでください。これらのスクリプトを編集、削除、または移動すると、サーバが正しく動作しなくなります。

- サンプル・ディレクトリ *samples* ディレクトリには、SQL サンプルおよびユーザ定義関数 (UDF) サンプルがあります。 *%SYBASE%/SybaseIQ/samples/sqlanywhere* には、SQL サンプルの 24 個のディレクトリがあります。 *sqlanywhere/c* ディレクトリには、SQL Anywhere で ESQL (埋め込み SQL) と C を使用する方法を示した C++ サンプルがあります。SQL Anywhere と Sybase IQ は同じコードを共有しているため、これらのサンプルを Sybase IQ 用に変更できます。 *%SYBASE%/SybaseIQ/samples/udf* ディレクトリには、サンプルの C++ スカラーと集約 UDF があります。
- 実行可能ファイル用ディレクトリ — 実行プログラム、ライブラリ、ヘルプ・ファイルなどがあります。

UNIX では、実行可能ファイル用サブディレクトリとして、 *IQ-15_1* のサブディレクトリ */bin64*、 */lib64*、 */logfiles*、 */res*、 */tix* があります。Windows では、実行可能ファイル用サブディレクトリとして、 *IQ-15_1* のサブディレクトリ *¥h*、 *¥install*、 *¥java*、 *¥bin32* があります。

Sybase IQ によるファイルの配置

Sybase IQ は、起動時と実行時にいくつかのファイルを検出してアクセスする必要があります。正しいファイルが使用されるようにするには、Sybase IQ がこれらのファイルをどのように見つけるかを理解しておくことが重要です。システム上には、同じ名前を持つディレクトリまたはファイルが複数存在する可能性があります。Sybase IQ は、Adaptive Server™ Enterprise と SQL Anywhere の両方のライブラリを使用します。これらの製品のいずれかがすでにシステムにインストールされている場合は、混乱を避けるため、インストールされているディレクトリを確認しておいてください。

アクセスする必要があるファイル・タイプは、次のとおりですが、これだけとは限りません。

- ライブラリ — 製品ライブラリまたはシステム・ライブラリが含まれています。UNIX でのファイル名拡張子は `.so.nnn` または `.so`、Windows では `.dll` または `.lib` です。これらのファイルは、Sybase IQ の実行に必要です。不適切な DLL が見つかった場合、バージョン不一致エラーが発生する可能性があります。たとえば、ライブラリ・ファイルは、UNIX では `$IQDIR15/lib64` または `$SYBASE/$SYBASE_OCS/lib64` にあります。Windows では `%IQDIR15%\bin32` または `%SYBASE%\SYBASE_OCS\bin32` にあります。`start_iq` には、通常のライブラリ・ディレクトリよりも前に `usrlib` が含まれているため、空のディレクトリ `$IQDIR15/usrlib` では、デフォルトのライブラリをカスタム・ライブラリおよびパッチで置き換えることができます。
- インタフェース・ファイル — Sybase IQ の実行に必要です。たとえば、インタフェース・ファイルは、UNIX では `.odbc.ini` および `utility_db.ini`、Windows では `util_db.ini` です。これらのファイルの詳細については、『システム管理ガイド 第 1 巻』および『インストールおよび設定ガイド』を参照してください。
- 設定ファイル — 接続パラメータを指定するために使用します。たとえば、設定ファイルは、Windows では `default.cfg`、または `iqdemo.cfg` です。
- データベース・ファイル — データおよびメタデータを格納します。データベース・ファイルには、`iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg`、`iqdemo.iqtmp` などがあります。
- ログ・ファイル — サーバ上の現在のセッションおよび接続されているデータベースに関する情報を格納します。たとえば、サーバ・ログには、`%SYBASE%/SybaseIQ/IQ15_1/logfiles/yourservername.0006.srvlog` のような名前が付けられています。データベースに接続すると、データベース・ログ (`%SYBASE%/SybaseIQ/IQ-15_1/demo/iqdemo.log` など) が生成されます。これらのファイルの詳細については、『インストールおよび設定ガイド』を参照してください。
- プロダクト・スクリプト — データベースを作成、移植、アップグレードする方法が示されたサンプル・ファイルです。
- ユーザ・ファイル — LOAD コマンドで使用されるフラット・ファイルおよび Interactive SQL などのツールで使用される SQL スクリプトがあります。
- テンポラリ・ファイル — クエリのソートの実行などの操作に関するテンポラリ情報を格納するために、Sybase IQ が作成します。

いくつかのファイル名は、SQL 文で指定され、実行時に検出される必要があります。ファイル名を使用する SQL 文の例は次のとおりです。

- `INSTALL` 文 — Java クラスを保持するファイルの名前。
- `LOAD TABLE` 文 — データのロード元となるファイルの名前。
- `CREATE DATABASE` 文 — この文のように、ファイルを作成する可能性のある文には、ファイル名が必要です。

Sybase IQ は、簡単なアルゴリズムを使用してファイルを検出する場合があります。より広範囲の検索が実行される場合もあります。

簡単なファイル検索

多くの SQL 文 (`LOAD TABLE` や `CREATE DATABASE` など) では、ファイル名はデータベース・サーバの現在の作業ディレクトリ (サーバが起動されたディレクトリ) に対して相対的なものとして解釈されます。

また、データベース・サーバを起動し、データベース・ファイル名 (`DBF` パラメータ) を指定すると、パスは、サーバが起動されたディレクトリに対する相対パスとして解釈されます。

広範囲なファイル検索

データベース・サーバおよび管理ユーティリティを含む Sybase IQ プログラムは、DLL や共有ライブラリなど必要なファイルを広範囲に検索します。この場合、Sybase IQ プログラムは次の順序でファイルを検索します。

- 1 実行ディレクトリ — 実行プログラムのあるディレクトリです。また、実行プログラム・ディレクトリに対して次の相対パスを持つディレクトリです。
 - 実行ディレクトリの親ディレクトリ
 - `scripts` という親ディレクトリの子
- 2 現在の作業ディレクトリ — 開始したプログラムには、現在の作業ディレクトリ (そのプログラムが起動したディレクトリ) があります。必要なファイルは、このディレクトリ内で検索されます。

- 3 LOCATION レジストリ・エントリ — Windows インストール環境では、Sybase IQ は LOCATION レジストリ・エントリを追加します。指定されたディレクトリが検査され、その後に次のディレクトリが検索されます。
 - *scripts* という名前の子ディレクトリ
 - オペレーティング・システム名 (*bin32*、*bin* など) の付いた子ディレクトリ
- 4 システム固有ディレクトリ — 一般的なオペレーティング・システムが格納されているディレクトリが含まれます。たとえば、Windows では、Windows ディレクトリと Windows¥system ディレクトリがこれに含まれます。
- 5 CLASSPATH ディレクトリ — Java ファイルの場合は、CLASSPATH 環境変数に指定されているディレクトリ内のファイルを検索します。
- 6 PATH ディレクトリ — システム・パスおよびユーザのパスに含まれるディレクトリ内のファイルを検索します。
- 7 LIBRARY PATH ディレクトリ — *LIBPATH* 環境変数に指定されているディレクトリ内で共有ライブラリが検索されます。

環境変数

Sybase IQ では、一連の環境変数を使用してさまざまなタイプの情報を格納します。必ずしもすべての変数を指定する必要はありません。

環境変数の設定

必須の環境変数は、UNIX では環境ソース・ファイル、Windows では Sybase IQ インストールによって設定されます。

❖ UNIX 環境ソース・ファイルの実行

すべての必須環境変数を設定するには、次のコマンドを実行します。

- 1 Bourne/Korn シェルでは次のように入力します。

```
. $SYBASE/IQ-15_1/IQ-15_1.sh
```


2 C シェルでは、次のように入力します。

```
source $SYBASE/IQ-15_1/IQ-15_1.csh;  
rehash
```

Windows プラットフォームでは、すべての環境変数がインストール・プログラムによって自動的に設定されるため、変更は不要です。ただし、オプションの変数を設定したり、デフォルト値から変更する必要がある場合は、オペレーティング・システムに対応した、次のいずれかの手順を使用します。

❖ **Windows での環境変数の設定**

- 1 デスクトップで、[マイ コンピュータ] を右クリックし、サブメニューから [プロパティ] を選択します。
- 2 [詳細設定] タブをクリックします。
- 3 [環境変数] ボタンをクリックします。

[環境変数] ダイアログが開きます。

- a 環境変数がまだない場合は、[新規] をクリックして変数名とその値を入力し、[OK] をクリックします。
- b 変数がある場合、[システム環境変数] または [ユーザ環境変数] リストから変数を選択し、[編集] をクリックし、[値] フィールドを必要に応じて変更します。次に、[OK] をクリックして設定を保存します。

注意 ユーザ環境変数とシステム環境変数については、Microsoft Windows のマニュアルを参照してください。

❖ **UNIX での環境変数の設定**

- 1 環境変数の設定をチェックするには、次のコマンドを使用します。

```
echo $variable-name
```

たとえば、\$SYBASE 変数の設定を確認するには、次のコマンドを実行します。

```
% echo $SYBASE  
/server1/users/test/sybase
```

- 2 起動ファイル (.cshrc、.shrc、.login) の 1 つに変数を設定する行を追加します。

一部のシェル (sh、bash、ksh など) では、次のような行になります。

```
VARIABLE=value;export VARIABLE
```

その他のシェル (csh、tsch など) では、次のような行になります。

```
setenv VARIABLE "value"
```

Sybase IQ で使用する変数の詳細については、次を参照してください。

- 「IQDIR15 環境変数」(8 ページ)
- 「IQPORT 環境変数」(9 ページ)
- 「IQLOGDIR15 環境変数」(9 ページ)
- 「IQTIMEOUT 環境変数」(10 ページ)
- 「IQTMP15 環境変数」(10 ページ)
- 「LIBPATH 環境変数と LD_LIBRARY_PATH 環境変数」(12 ページ)
- 「PATH 環境変数」(12 ページ)
- 「SACHARSET 環境変数」(12 ページ)
- 「SALANG 環境変数」(13 ページ)
- 「SQLCONNECT 環境変数」(13 ページ)
- 「SYBASE 環境変数」(14 ページ)
- 「`$$SYBASE_JRE6_32`、`$$SYBASE_JRE6_64`、`$$SYBASE_JRE5_64` 環境変数」(14 ページ)
- 「SYBASE_OCS 環境変数」(15 ページ)

IQDIR15 環境変数

設定値

```
IQDIR15 = ${SYBASE}/IQ-15_1
```

オペレーティング・システム

(必須) 環境ソース・ファイルまたはインストール・プログラムによって設定されます。このデフォルト設定は、Windows では変更可能です。

説明

IQDIR15 は、Sybase IQ ディレクトリの場所を示すとともに、そのディレクトリ内の他のディレクトリおよびファイルの場所です。

- `IQDIR15/bin[64]/util_db.ini` には、ユーティリティ・データベース `utility_db` のログイン ID とパスワードが含まれています。インストール・プログラムでは、ログイン ID とパスワードのデフォルトの値 (それぞれ、“DBA” と “sql”) を変更できます。

- `$IQDIR15/logfiles` は、サーバ・ログとバックアップ/リストア・ログ(バックアップ履歴ファイル)のデフォルト・ロケーションです。`IQLOGDIR15` 環境変数を設定することで、このデフォルト設定を変更できます。
- `$IQDIR15/demo` は、`iqdemo` データベース・ファイルの場所です。

IQPORT 環境変数

設定値

`IQPORT = 5556`

オペレーティング・システム

オプション。環境ソース・ファイル内にユーザが `IQPORT` を指定していない場合、ポート番号はデフォルトで `1099` になります。このデフォルト値は、プラグインの起動前であれば変更できます。この変数は、「環境変数の設定」(6 ページ)で解説されている方法、または、Sybase Central の開始時に `-DIQPORT` 引数を `scjview` コマンドに指定することで指定できます。例を示します。

```
scjview -DIQPORT=3345
```

説明

Sybase IQ プラグインとエージェントとの間の通信に使用される、Sybase IQ Agent のポート番号のデフォルト値を上書きします。

注意 エージェントが起動すると、ポート番号を変更できません。

`1099` は、特定のポートでエージェント・プロセスを検索するときに使用される、プラグインのデフォルト値です。プラグインは、このポートでエージェントを検出できない場合、正しいポート番号を指定するよう求めるメッセージを表示します。

IQLOGDIR15 環境変数

設定値

`IQLOGDIR15 = path`

オペレーティング・システム

オプション。

説明

`IQLOGDIR15` 環境変数は、インストール・プログラムでは設定されません。この環境変数は、さまざまなログ・ファイルのロケーションを定義します。

- サーバ・ログは、`$IQLOGDIR15` で指定されたディレクトリ内の `servername.nnnn.srvlog` ファイル (nnnn は、サーバが起動された回数) 内にあります。

IQLOGDIR15 が有効で、書き込み可能なディレクトリに設定されていない場合、`start_iq` を含む大部分のユーティリティは、すべてのサーバ・ログにデフォルトの場所 `$IQDIR15/logfiles` を使用します。

IQTIMEOUT 環境変数

設定値

`IQTIMEOUT = nnn`

オペレーティング・システム

オプション、ただしマルチプレックス環境では推奨。マルチプレックス機能の詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。

説明

Sybase IQ Agent は、処理の完了を無期限で待機します。非常に大きなカタログ・ストアを持つマルチプレックスのクエリ・サーバを作成したり同期したりするときには、待機時間を設定することをおすすめします。カタログ・ストアが大きい場合、同期の `dbbackup` 部分に必要な時間が増大します。待機時間を増やすことによって、より長時間の同期に対応できます。

`IQTIMEOUT` 変数は、デフォルトの待機時間 (5 分) を変更します。引数 `nnn` は、Sybase IQ Agent が待機する時間 (分単位) です。例を示します。

- 待機時間を 45 分にするには、次のように指定します (Korn シェルまたは Bourne シェルの場合)。

```
IQTIMEOUT=45
export IQTIMEOUT
```

- 待機時間を 1 時間にするには、次のように指定します (C シェルの場合)。

```
setenv IQTIMEOUT 60
```

注意 `IQTIMEOUT` 変数の設定は、エージェント起動オプションを呼び出す前に行ってください。『インストールおよび設定ガイド』の「インストール前の作業」と「Sybase IQ Agent の起動」、および『Sybase IQ の概要』の「[Sybase IQ Agent の実行](#)」を参照してください。

IQTMP15 環境変数

設定値

`IQTMP15 = temp_directory`

オペレーティング・システム

UNIX の場合、オプションです。Windows プラットフォームでは使用されません。

説明

IQTMP15 環境変数は、インストール・プログラムでは設定されません。IQTMP15 は、テンポラリ・ファイルが格納されるディレクトリを示すために Sybase IQ で使用されます。

IQTMP15 環境変数は、NFS (Network File System) を使用しているローカル・ディレクトリを示す必要があります。これによって、IQTMP15 ディレクトリは、クライアント接続が終了したときに不必要となったディレクトリやファイルを消去できます。クライアント接続が行われるたびに、ディレクトリやファイルがテンポラリ・ディレクトリ内に作成されます。これらは、接続中にだけ必要です。サーバに接続するすべてのユーザが、このディレクトリに対する書き込みパーミッションを持っている必要があります。

注意 IQTMP15 変数によって定義されるロケーションにあるテンポラリ・ファイルは、クライアントとサーバによって使用されます。この変数は、テンポラリ IQ ストアのデフォルト・ロケーションを制御しません。Sybase IQ が、テンポラリ・ストアのロケーションを決定する仕組みについては、『リファレンス：文とオプション』の第 1 章 SQL 文に記載されている「CREATE DATABASE 文」を参照してください。

IQTMP15 変数を明示的に設定しない場合や、\$SYBASE または \$IQDIR15 に設定されている場合、Sybase IQ Agent は、IQTMP15 変数を UNIX ディレクトリ `/tmp` のサブディレクトリに設定します。

マシン上で複数のデータベース・サーバが実行されている場合、それぞれのサーバと関連するローカル・クライアントは、競合を避けるために別々のテンポラリ・ディレクトリを必要とします。接続に使用するポート番号またはエンジン番号を指定しない場合、Sybase IQ は、ネットワーク接続の代わりに共有メモリ接続を使用します。

共有メモリを使用したときに競合を避けるには、次の手順に従います。

- 各サーバに専用のテンポラリ・ディレクトリを作成します。両方の環境に明示的に IQTMP15 環境変数を設定して、各ローカル・クライアントがサーバと同じテンポラリ・ディレクトリを使用するようにします。
- 各サーバの `.odbc.ini` ファイル (UNIX の場合) にデータ・ソース名を作成し、詳細な接続情報を指定します。『インストールおよび設定ガイド』を参照してください。

- デフォルトでなく、明示的にパラメータを指定する接続文字列を使用します。
- 次のコマンドを実行して、接続を確認します。

```
SELECT "database name is" = db_name(),  
       "servername_is" = @@servername
```

LIBPATH 環境変数と LD_LIBRARY_PATH 環境変数

構文

AIX の場合 :

```
LIBPATH = installation_path/lib
```

他のすべての UNIX/LINUX プラットフォームの場合 :

```
LD_LIBRARY_PATH = installation_path/lib
```

オペレーティング・システム

必須。変数名はプラットフォームに依存します。UNIX のみ。

説明

Sybase IQ 共有ライブラリが配置されているディレクトリを指定します。UNIX では、環境ソース・ファイルを実行することによってライブラリ・パス変数を設定します。

PATH 環境変数

設定値

```
PATH = installation_path
```

オペレーティング・システム

必須。

説明

PATH 環境変数は、Sybase IQ 実行プログラムが配置されているディレクトリが含まれる、オペレーティング・システムで必須の変数です。Windows では、インストール・プログラムが PATH を変更します。UNIX では、環境ソース・ファイルを実行して必要なディレクトリを含めます。

Windows では、PATH は LIBRARY_PATH 変数の代わりになります。したがって、実行プログラムと DLL の場所は PATH 変数を使って参照できます。

SACHARSET 環境変数

設定値

```
SACHARSET=charset
```

説明 *Charset* は文字セットの名前です。たとえば、**SACHARSET=cp1252** と指定すると、デフォルトの文字セットが **cp1252** に設定されます。

次の値セットのうち、最初の値がデフォルトの文字セットを決定します。

- **SACHARSET** 環境変数
- OS をクエリ

文字セット情報が指定されていない場合は、UNIX では **iso_1**、その他の場合は **cp850** を使用します。

SALANG 環境変数

設定値 **SALANG=language_code**

オペレーティング・システム オプション、ただし英語以外の環境では推奨。

説明 *language_code* は、2 文字で表された言語名です。たとえば、**SALANG=DE** のように設定すると、デフォルトの言語がドイツ語に設定されます。

次の値セットのうち、最初の値がデフォルトの言語を決定します。

- **SALANG** 環境変数
- インストーラまたは *dblang.exe* によって設定されたレジストリ (Windows のみ)
- OS をクエリ

言語情報が設定されていない場合のデフォルト値は英語です。

SQLCONNECT 環境変数

構文 **SQLCONNECT = parameter#value ; ...**

オペレーティング・システム オプション。

説明 **SQLCONNECT** はオプションの環境変数で、インストール・プログラムでは設定を行いません。

SQLCONNECT では、データベース・サーバに接続するときに、DBISQL、DBINFO、DBCOLLAT、DBSTOP などのデータベース管理ユーティリティが使用する接続パラメータを指定します。この文字列は、`parameter=value` というフォームで指定されたパラメータ設定値のリストです (設定値ごとにセミコロンで区切ります)。

シャープ記号“#”は、等号(=)の代わりに使用します。SQLCONNECT 環境変数内に接続パラメータ文字列を設定する場合は、シャープ記号を使用してください。環境変数設定の中で“=”を使用すると構文エラーになります。“#”記号は、Windows でのみ使用できます。

注意 接続パラメータをコマンド・ラインではなく SQLCONNECT 変数で指定することで、UNIX システムのセキュリティが向上します。ユーザが、`ps -ef` コマンドで他のユーザのパスワードを表示できなくなります。これは、DBISQL を実行するとき、またはその他のユーティリティをクワイエット・モードで実行するとき特に有用です。

参照

『システム管理ガイド 第 1 巻』の第 4 章 [接続パラメータと通信パラメータ](#)の「[接続パラメータ](#)」を参照してください。

SYBASE 環境変数

設定値 `SYBASE = path`

オペレーティング・システム 必須。

説明 SYBASE 変数は、Open Client や Open Server などの Sybase アプリケーションの場所を指定します。Sybase IQ を UNIX システムにインストールする前に SYBASE 変数を設定する必要があります。この環境変数は、UNIX システムで Sybase Central を使用する場合に必要です。

\$\$SYBASE_JRE6_32、\$\$SYBASE_JRE6_64、\$\$SYBASE_JRE5_64 環境変数

設定値 `SYBASE_JRE= "${SYBASE}/shared/jre-6_0"`

説明 この変数は、Sybase IQ の Sybase Central プラグインで使用する Java Runtime Environment の場所を指定します。Windows および UNIX の場合、この環境変数は \$\$SYBASE_JRE6_32 または \$\$SYBASE_JRE6_64 です。AIX/LinuxIBM の場合、この環境変数は \$\$SYBASE_JRE5_64 です。

UNIX では、環境ソース・ファイル *SYBASE.csh* (C シェル) または *SYBASE.sh* (Bourne シェルまたは Korn シェル) を実行します。Windows では、インストール・プログラムが Open Client Software Developer's Kit をインストールするときにこの変数を設定します。

SYBASE_OCS 環境変数

設定値 `SYBASE_OCS = "OCS-15_1"`

オペレーティング・システム 必須。

説明 SYBASE_OCS 変数は、Open Client 製品のホーム・ディレクトリを指定します。この変数は Window でのみ使用されます。Windows では、インストール・プログラムが Open Client/Server Software Developers Kit をインストールするときに SYBASE_OCS を設定します。

レジストリ・エントリ

Windows オペレーティング・システムでは、Sybase IQ は複数のレジストリ設定を使用します。これらの設定はソフトウェアが行うため、通常、ユーザがレジストリにアクセスする必要はありません。ここでは、操作環境を変更するユーザのために、これらの設定について説明します。

警告！ Sybase は、レジストリを変更しないことを推奨しています。不正な変更を行うと、システムが損傷する場合があります。

カレント・ユーザ設定とローカル・マシン設定

Windows などのオペレーティング・システムには、2 レベルのシステム設定があります。そのユーザがログオンしたときにだけ使用できる、個々のユーザに固有の設定のことを、カレント・ユーザ設定と呼びます。また、マシン全体に関連し、どのユーザがログオンしているにかかわらず使用できる設定をローカル・マシン設定と呼びます。ローカル・マシン設定を行うには、マシンの管理者パーミッションを入手する必要があります。

Sybase IQ では、カレント・ユーザ設定とローカル・マシン設定の両方を行えます。Windows では、これらの設定はそれぞれ HKEY_CURRENT_USER レジストリと HKEY_LOCAL_MACHINE レジストリに格納されます。

Sybase IQ では、設定の対象をカレント・ユーザのみにするか、ローカル・マシン・レベルにするかを選択できます。

設定がカレント・ユーザ設定とローカル・マシン設定の両方で行われている場合、カレント・ユーザ設定が優先されます。

ローカル・マシン設定 が必要な場合

Windows で Sybase IQ プログラムをサービスとして実行する場合は、ローカル・マシン・レベルで設定されていることを確認してください。

サービスは、マシンをログオフしても、マシンを完全に停止しないかぎり、特別なアカウントのもとで実行し続けます。サービスは、個々のアカウントから独立して実行可能であり、ローカル・マシン設定にアクセスする必要があります。

一般的に、Sybase はローカル・マシン設定を使用することを推奨しています。

レジストリ構造

Windows では、レジストリ・エディタを使用してレジストリに直接アクセスできます。

エディタを起動するには、[スタート]>[ファイル名を指定して実行]を選択し、[名前]ボックスに次のように入力します。

```
regedt32
```

注意 読み込み専用モードは、間違ってレジストリ・データを変更するのを防ぐことができます。読み込み専用モードにするには、レジストリ・エディタを起動して [編集]>[パーミッション]を選択し、読み取りパーミッションをオンにします。

Sybase IQ レジストリ・エントリは、HKEY_LOCAL_MACHINE キーの次のロケーションにあります。

```
SOFTWARE
  Sybase
    IQ 15.0
```

インストール時のレジストリ設定

インストール・プログラムは、Sybase レジストリ内で次のレジストリ設定を自動的に行います。

- ロケーション — Sybase IQ レジストリでは、このエントリはインストール・ディレクトリのロケーションを保持します。例を示します。

```
Location:REG_SZ:C:\Program Files\Sybase  
IQ-15_1
```

Sybase IQ レジストリには、インストールされているアプリケーションの他のエントリが含まれています。Sybase Central レジストリには、Sybase Central のバージョンとインストール済みのプラグインの情報が含まれます。

この章について

この章では、Sybase IQ SQL の言語要素と規則について詳細に説明します。

内容

| トピック名 | ページ |
|------------------------|-----|
| キーワード | 19 |
| 識別子 | 22 |
| 文字列 | 24 |
| 式 | 25 |
| 検索条件 | 34 |
| 特別値 | 54 |
| 変数 | 57 |
| コメント | 65 |
| NULL 値 | 67 |

キーワード

各 SQL 文には 1 つまたは複数のキーワードが含まれています。SQL 文のキーワードでは大文字と小文字を区別しませんが、このマニュアルではキーワードを大文字で表記します。

たとえば、次の文では **SELECT** と **FROM** がキーワードになります。

```
SELECT *
FROM Employees
```

次の文は、上の文と同じです。

```
Select *
From Employees
select * from Employees
sELECT * FRoM Employees
```

予約語

SQL キーワードには**予約語**として定義されているものがあります。SQL 文で予約語を識別子として使用するには、二重引用符で囲む必要があります。SQL 文で使用するキーワードのうち、すべてではありませんがその多くが予約語です。たとえば、次の構文を使用して、SELECT テーブルの内容を取り出します。

```
SELECT *
FROM "SELECT"
```

SQL はキーワードの大文字と小文字を区別しないため、[表 2-1](#) の各語は大文字、小文字、またはその組み合わせで使用できます。次の語のいずれかと、大文字/小文字の区別のみが違う文字列はすべて予約語となります。

Embedded SQL を使用している場合、データベース・ライブラリ関数 `sql_needs_quotes` を使用すると、文字列に二重引用符が必要かどうかを判別できます。文字列が予約語であるか、通常は識別子に使用できない文字が文字列に含まれている場合は、文字列に二重引用符を付けます。

[表 2-1](#) は、Sybase IQ の SQL 予約語です。

表 2-1 : SQL 予約語

| | | | |
|-------------|-------------------|--------------|------------|
| active | add | all | algorithm |
| alter | and | any | append |
| as | asc | auto | backup |
| begin | between | bigint | binary |
| bit | bottom | break | by |
| calibrate | calibration | call | cancel |
| capability | cascade | case | cast |
| certificate | char | char_convert | character |
| check | checkpoint | checksum | clientport |
| close | columns | comment | commit |
| committed | comparisons | computes | conflict |
| connect | constraint | contains | continue |
| convert | create | cross | cube |
| current | current_timestamp | current_user | cursor |
| date | dbspace | dbspacename | deallocate |
| debug | dec | decimal | declare |
| decoupled | decrypted | default | delay |
| delete | deleting | density | desc |

| | | | |
|-----------------|----------------|--------------|--------------|
| deterministic | disable | distinct | do |
| double | drop | dynamic | elements |
| else | elseif | enable | encapsulated |
| encrypted | end | endif | escape |
| except | exception | exclude | exec |
| execute | existing | exists | explicit |
| express | externlogin | fastfirstrow | fetch |
| first | float | following | for |
| force | foreign | forward | from |
| full | gb | goto | grant |
| group | grouping | having | hidden |
| history | holdlock | identified | if |
| in | inactive | index | index_lparen |
| inner | inout | input | insensitive |
| insert | inserting | install | instead |
| int | integer | integrated | intersect |
| into | iq | is | isolation |
| jdk | join | kb | key |
| lateral | left | like | lock |
| logging | login | long | mb |
| match | membership | message | mode |
| modify | namespace | natural | new |
| no | noholdlock | nolock | not |
| notify | null | numeric | of |
| off | on | open | optimization |
| option | options | or | order |
| others | out | outer | over |
| pages | paglock | partial | partition |
| passthrough | password | plan | preceding |
| precision | prepare | primary | print |
| privileges | proc | procedure | proxy |
| publication | raiserror | range | raw |
| readcommitted | readonly | readpast | readtext |
| readuncommitted | readwrite | real | recursive |
| reference | references | release | relocate |
| remote | remove | rename | reorganize |
| repeatable | repeatableread | reserve | resizing |
| resource | restore | restrict | return |

| | | | |
|--------------|------------------|----------------|---------------|
| revoke | right | rollback | rollup |
| root | row | rowlock | rows |
| save | savepoint | schedule | scroll |
| secure | select | sensitive | serializable |
| service | session | set | setuser |
| share | smallint | soapaction | some |
| space | sqlcode | sqlstate | start |
| stop | subtrans | subtransaction | synchronize |
| syntax_error | table | tablock | tablockx |
| tb | temporary | then | ties |
| time | timestamp | tinyint | to |
| top | tran | transaction | transactional |
| transfer | tries | trigger | truncate |
| tsequal | unbounded | uncommitted | union |
| unique | uniqueidentifier | unknown | unsigned |
| update | updating | updlock | url |
| user | utc | using | validate |
| values | varbinary | varchar | variable |
| varying | virtual | view | wait |
| waitfor | web | when | where |
| while | window | with | withauto |
| with_cube | with_lparen | with_rollup | within |
| word | work | writeserver | writetext |
| xlock | xml | | |

識別子

機能

識別子は、ユーザ ID、テーブル、カラムなどのデータベースのオブジェクト名を表します。

説明

識別子の最大長は、128 バイトです。識別子は、次のいずれかの条件に当てはまる場合、二重引用符または角カッコで囲む必要があります。

- 識別子にスペースが含まれている。
- 識別子の最初の文字がアルファベット文字 (以下で説明) ではない。
- 識別子に予約語が含まれている。

- 識別子にアルファベット文字と数字以外の文字が含まれている。
アルファベット文字に含まれるのは、アルファベット、アンダースコア文字(`_`)、アット・マーク(`@`)、シャープ記号(`#`)、ドル記号(`$`)です。データベースの照合順によって、どの文字をアルファベットまたは数字として扱うかが決まります。

アポストロフィ (一重引用符) を識別子の中で使用するには、一重引用符を2つ続けます。

識別子には、次の制限事項があります。

- テーブル名に二重引用符を含めることはできない。
- ユーザ名とデータベース名に、二重引用符、一重引用符、およびセミコロン文字を含めることはできない。
- ユーザ名およびデータベース名の先頭または末尾にスペースを使用できない。
- CASE RESPECT** データベースでは、DB 領域名の大文字と小文字が区別されない。

QUOTED_IDENTIFIER データベース・オプションが **OFF** に設定されている場合、二重引用符は SQL 文字列を区切るために使用され、識別子には使用できません。ただし、角カッコは **QUOTED_IDENTIFIER** の設定に関係なく識別子の区切りに使用できます。

QUOTED_IDENTIFIER オプションのデフォルトの設定は、Open Client および jConnect 接続では **OFF**、その他の場合は **ON** です。

例

次の例は、いずれも有効な識別子です。

```
Surname
"Surname"
[Surname]
SomeBigName
"Client Number"
```

参照

予約語のリストについては、「[予約語](#)」(20 ページ) を参照してください。

QUOTED_IDENTIFIER オプションの詳細については、「[quoted_identifier オプション](#)」(33 ページ) を参照してください。

サーバおよびデータベースの名前の制約については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」の「[サーバのコマンド・ライン・スイッチ](#)」(8 ページ) を参照してください。

文字列

文字列には次のタイプがあります。

- リテラル文字列
- CHAR データ型または VARCHAR データ型を持つ式

組み込み関数やユーザ定義関数、またはそれ以外の多くの使用可能な式が CHAR データ型を持つことができます。

式の詳細については、「式」(25 ページ)を参照してください。

リテラル文字列とは、アポストロフィ ('一重引用符') で囲まれ、任意のシーケンスで並べられた文字のことです。文字データ型の SQL 変数には、文字列を入れることができます。次に、リテラル文字列の簡単な例を示します。

```
'This is a string.'
```

文字列中の特殊文字

文字列中の特殊文字を表すには、次のように、エスケープ・シーケンスを使用します。

- 文字列内でアポストロフィを表すには、アポストロフィを 2 つ続けて記述します。例を示します。

```
'John''s database'
```

- 文字列内で改行を表すには、円記号に続けて n (¥n) を記述します。例を示します。

```
'First line:¥nSecond line:'
```

- 円記号を表すには、円記号を 2 つ続けて (¥¥) 記述します。例を示します。

```
'c:¥¥temp'
```

- 16 進のエスケープ・シーケンスは、印刷できるかどうかに関係なくあらゆる文字に使用できます。16 進のエスケープ・シーケンスは、円記号とその後に x と 2 桁の 16 進数がある文字列です (たとえば、¥x6d は、文字 m を表します)。例を示します。

```
'¥x00¥x01¥x02¥x03'
```

互換性

Adaptive Server Enterprise との互換性を保つには、QUOTED_IDENTIFIER データベース・オプションを OFF に設定します。この設定では、文字列の先頭と末尾のマークに二重引用符を使用することもできます。このオプションは、デフォルトでは ON になっています。

式

構文

```
expression:
case-expression
| constant
| [ correlation-name. ] column-name [ java-ref ]
| - expression
| expression operator expression
| ( expression )
| function-name ( expression, ... )
| if-expression
| [ java-package-name. ] java-class-name java-ref
| ( subquery )
| variable-name [ java-ref ]
```

パラメータ

```
case-expression:
{ CASE search-condition
... WHEN expression THEN expression [ , ... ]
... [ ELSE expression ]
END
| CASE
... WHEN search-condition THEN expression [ , ... ]
... [ ELSE expression ]
END }
```

```
constant:
{ integer | number | 'string' | special-constant | host-variable }
```

```
special-constant:
{ CURRENT { DATE | TIME | TIMESTAMP | USER }
| LAST USER
| NULL
| SQLCODE
| SQLSTATE }
```

```
if-expression:
IF condition
... THEN expression
... [ ELSE expression ]
ENDIF
```

```
java-ref:
{ .field-name [ java-ref ]
| >> field-name [ java-ref ]
| .method-name ( [ expression ] [ , ... ] ) [ java-ref ]
| >> method-name ( [ expression ] [ , ... ] ) [ java-ref ] }
```

```
operator:
{ + | - | * | / | || | % }
```

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 使用法 | すべての場所 |
| 権限 | データベースに接続しておく必要があります。 |
| 関連する動作 | なし |
| 説明 | 式は、以降の項で説明する数種類の要素で構成されます。 |
| 互換性 | <ul style="list-style-type: none">Adaptive Server Enterprise では、IF 条件はサポートされません。Adaptive Server Enterprise では、Java 式は現在サポートされません。その他の相違点については、以降の項で説明する式の各クラスの説明を参照してください。 |

式内の定数

定数とは、数値または文字列です。文字列定数は、アポストロフィで囲まれています。文字列内でアポストロフィを表すには、アポストロフィを2つ続けて記述します。

式内のカラム名

カラム名は識別子の1つであり、前に関連名が付くことがあります。通常、関連名はテーブル名です。関連名の詳細については、『リファレンス：文とオプション』の「[FROM 句](#)」を参照してください。カラム名に英字、数字、アンダースコア以外の文字が使用されている場合は、二重引用符(“”) で囲んでください。以下は、有効なカラム名の例です。

```
Employees.Surname  
City  
"StartDate"
```

詳細については、「[識別子](#)」(22 ページ) を参照してください。

式内のサブクエリ

サブクエリとは、カッコに囲まれた SELECT 文です。SELECT 文には、リスト項目を1つだけ指定できます。式として使用すると、通常、スカラ・サブクエリはゼロまたは1つの値しか返せません。

最上位レベルの SELECT の SELECT リスト、または UPDATE 文の SET 句の中では、カラム名を使用できる場所ではどこでもスカラ・サブクエリを使用できます。ただし、条件式 (CASE、IF、NULLIF、ARGN) 内でサブクエリを使用することはできません。

たとえば、次の文は、各部署の従業員数を、部署名でグループ化して返します。

```
SELECT DepartmentName, COUNT(*), 'out of',  
       (SELECT COUNT(*) FROM Employees)  
FROM Departments AS D, Employees AS E  
WHERE D.DepartmentID = E.DepartmentID  
GROUP BY DepartmentName;
```

サブクエリのその他の使用法については、「[検索条件内のサブクエリ](#)」(37 ページ)を参照してください。

SQL 演算子

この項では、Sybase IQ で使用可能な算術演算子、文字列演算子、ビット処理演算子について説明します。比較演算子の詳細については、「[検索条件](#)」(34 ページ)を参照してください。

一般的な演算の優先度が適用されます。カッコ内の式が最初に評価され、続いて乗算と除算、最後に加算と減算が評価されます。その後、文字列の連結が行われます。

算術演算子

expression + expression 加算。いずれかの式が NULL 値の場合、結果は NULL 値になります。

expression - expression 減算。いずれかの式が NULL 値の場合、結果は NULL 値になります。

- expression 反転。式が NULL 値の場合、結果は NULL 値になります。

expression * expression 乗算。いずれかの式が NULL 値の場合、結果は NULL 値になります。

expression / expression 除算。いずれかの式が NULL 値か、または 2 番目の式が 0 の場合、結果は NULL 値になります。

expression % expression モジュロによる、2 つの整数での除算の余り (整数) の算出。たとえば、21 を 11 で割った場合の余りは 10 なので、 $21 \% 11 = 10$ になります。

文字列演算子

expression || expression 文字列連結 (2 本の縦線)。いずれかの文字列が NULL 値の場合、連結では空の文字列として扱われます。

expression + expression 代替の文字列連結。+ 連結演算子を使用する場合は、暗黙的データ変換を行わないで、必ずオペランドを文字データ型に明示設定してください。

文字列連結演算子の結果データ型は、LONG VARCHAR です。SELECT INTO 文で文字列連結演算子を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して LEFT を正しいデータ型とサイズに設定する必要があります。

詳細および使用方法については、「[REVERSE 関数 \[文字列\]](#)」(245 ページ) を参照してください。

標準と互換性

- **SQL92** || 演算子は、SQL92 の文字列連結演算子です。
- **Sybase** + 演算子は、Adaptive Server Enterprise でサポートされています。

ビット処理演算子

次の演算子は、Sybase IQ と Adaptive Server Enterprise で、位取りされていない整数データ型に使用できます。

| 演算子 | 説明 |
|-----|--------------|
| & | AND |
| | OR |
| ^ | EXCLUSIVE OR |
| ~ | NOT |

AND 演算子 (&)

| Bit 1 | Bit 2 | Bit 1 & Bit 2 |
|-------|-------|---------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

AND 演算子は 2 つのビットを比較します。両方とも 1 の場合、結果は 1 です。

ビット処理 OR (|)

| Bit 1 | Bit 2 | Bit 1 Bit 2 |
|-------|-------|---------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

OR 演算子は 2 つのビットを比較します。どちらかが 1 の場合、結果は 1 です。

EXCLUSIVE OR (^)

2 つのオペランドの両方ではなく一方だけが 1 である場合、EXCLUSIVE OR 演算子の結果は 1 です。

| Bit 1 | Bit 2 | Bit 1 ^ Bit 2 |
|-------|-------|---------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NOT (~)

NOT 演算子は、そのオペランドの逆の値を返す単項演算子です。

| Bit | ~ Bit |
|-----|-------|
| 1 | 0 |
| 0 | 1 |

ジョイン演算子

Sybase IQ では、FROM 句にテーブル式を使用する SQL92 のジョイン構文のほかにも、Transact-SQL™ 外部ジョイン演算子 *= と =* をサポートしています。

互換性

- **モジュロ** 新規データベースについては、デフォルト値は OFF です。
- **文字列連結** Sybase IQ で + 連結演算子を使用する場合は、暗黙的データ変換を行わずに、必ずオペランドを文字データ型に明示設定してください。たとえば、次のクエリは整数値 579 を返します。

```
SELECT 123 + 456
```

これに対し、次のクエリは文字列 123456 を返します。

```
SELECT '123' + '456'
```

CAST または CONVERT 関数を使用すると、データ型を明示的に変換できます。

注意 BINARY または VARBINARY データ型で使用する、+ 演算子は加算ではなく連結演算子として機能します。

Adaptive Server Enterprise では、|| 連結演算子をサポートしていません。

演算子の優先度

1つの式に複数の演算子を使用している場合は、Adaptive Server Enterprise と Sybase IQ で同一の演算子の優先度に依存しないで、カッコを使用して演算の順序を明示指定することをおすすめします。

IF 式

IF 式の構文は、次のとおりです。

```
IF condition  
THEN expression1  
[ ELSE expression2 ]  
ENDIF
```

この式は、次の値を返します。

- *condition* が TRUE の場合、IF 式は *expression1* を返します。
- *condition* が FALSE の場合、IF 式は *expression2* を返します。
- *condition* が FALSE で *expression2* がない場合、IF 式は NULL を返します。
- 条件が NULL の場合、IF 式は NULL を返します。

TRUE、FALSE、UNKNOWN の各条件については、「[NULL 値](#)」(67 ページ) および「[検索条件](#)」(34 ページ) を参照してください。

IF 文は IF 式とは違います

IF 式の構文と IF 文の構文を混同しないでください。

詳細については、『リファレンス：文とオプション』の「[IF 文](#)」を参照してください。

CASE 式

CASE 式は、条件付きの SQL 式を提供します。CASE 式は、式を使用できる場所ならどこでも使用できます。

CASE 式の構文は、次のとおりです。

```
CASE expression  
WHEN expression THEN expression [, ...]  
[ ELSE expression ] END
```

CASE 文で、値式としてサブクエリを使用することはできません。

CASE 文に続く式が WHEN 文に続く式と等しい場合、THEN 文に続く式が返されます。それ以外の場合、ELSE 文があれば、それに続く式が返されます。

たとえば、次のコードでは CASE 式を SELECT 文の 2 番目の句として使用しています。

```
SELECT ID,  
       (CASE name  
        WHEN 'Tee Shirt' THEN 'Shirt'  
        WHEN 'Sweatshirt' THEN 'Shirt'  
        WHEN 'Baseball Cap' THEN 'Hat'  
        ELSE 'Unknown'  
        END) as Type  
FROM "GROUP0".Products
```

次の構文も使用できます。

```
CASE  
WHEN search-condition THEN expression [, ...]  
[ ELSE expression ] END
```

WHEN 文に続く検索条件が満たされた場合は、THEN 文に続く式が返されます。それ以外の場合、ELSE 文があれば、それに続く式が返されます。

たとえば、次の文では、CASE 式を SELECT 文の 3 番目の句として使用し、検索条件と文字列を関連付けています。

```
SELECT ID, name,  
       (CASE  
        WHEN name='Tee Shirt' THEN 'Sale'  
        WHEN quantity >= 50 THEN 'Big Sale'  
        ELSE 'Regular price'  
        END) as Type  
FROM "GROUP0".Products
```

省略形 CASE 式の
NULLIF 関数

NULLIF 関数は、CASE 文を省略形で記述する方法の 1 つです。NULLIF の構文は、次のとおりです。

NULLIF (*expression-1*, *expression-2*)

NULLIF は、2 つの式の値を比較します。最初の式と次の式が一致する場合、NULLIF は NULL を返します。一致しない場合、NULLIF は最初の式の値を返します。

式の互換性

Adaptive Server Enterprise (ASE) と Sybase IQ の式と定数の互換性を、表 2-2 および表 2-3 に示します。これらの表は、簡単な説明を目的としてまとめたものです。両方と記されていても、状況や目的に左右されることなく、同じ方法で式を実行できるわけではありません。詳細については、Adaptive Server Enterprise と Sybase IQ の各マニュアルに記載された個々の式の説明を参照してください。

表 2-2 で、*expr* は式、*op* は演算子を表します。

表 2-2 : ASE と Sybase IQ の式の互換性

| 式 | サポート |
|-----------------------------------|--------------|
| constant | 両方 |
| column name | 両方 |
| variable name | 両方 |
| function (<i>expr</i>) | 両方 |
| - <i>expr</i> | 両方 |
| <i>expr</i> <i>op</i> <i>expr</i> | 両方 |
| (<i>expr</i>) | 両方 |
| (<i>subquery</i>) | 両方 |
| if-expression | Sybase IQ のみ |

表 2-3 : ASE と Sybase IQ の定数の互換性

| 定数 | サポート |
|------------------|-----------|
| integer | 両方 |
| number | 両方 |
| 'string' | 両方 |
| special-constant | 両方 |
| host-variable | Sybase IQ |

区切り文字列の デフォルト解釈

デフォルトでは、Adaptive Server Enterprise と Sybase IQ の区切り文字列の意味は異なります。区切り文字列とは、アポストロフィ (一重引用符) や引用符 (二重引用符) で囲まれた文字列です。

Sybase IQ では、アポストロフィで囲まれた文字列を定数式、二重引用符で囲まれた文字列を区切り識別子 (データベース・オブジェクト用の名前) とする SQL92 の表記規則を採用しています。Adaptive Server Enterprise では、二重引用符で囲まれた文字列を定数とし、デフォルトでは区切り識別子を識別子としてではなく文字列として扱う表記規則を採用しています。

quoted_identifier オプション

Adaptive Server Enterprise と Sybase IQ には、区切り文字列の解釈を変更できる `quoted_identifier` オプションがあります。デフォルトで、`quoted_identifier` オプションは、Adaptive Server Enterprise では OFF、Sybase IQ では ON に設定されています。

`quoted_identifier` オプションが OFF の場合、SQL 予約語を識別子として使用することはできません。

予約語のリストについては、[表 2-1 \(20 ページ\)](#) を参照してください。

オプションの設定

Transact-SQL の SET 文は、通常、Adaptive Server Enterprise 接続オプション用にサポートされていませんが、`quoted_identifier` オプション用にはサポートされています。

Sybase IQ または Adaptive Server Enterprise では、次の文を使って `quoted_identifier` オプションの設定を ON に変更します。

```
SET quoted_identifier ON
```

`quoted_identifier` オプションを ON に設定すると、Adaptive Server Enterprise では、テーブル、ビュー、カラム名を二重引用符で区切ることができます。他のオブジェクト名は、Adaptive Server Enterprise では区切ることができません。

Sybase IQ または Adaptive Server Enterprise では、次の文を使って `quoted_identifier` オプションの設定を OFF に変更します。

```
SET quoted_identifier OFF
```

Adaptive Server Enterprise と Sybase IQ の各 DBMS で `quoted_identifier` オプションが同じ値に設定されていれば、SQL92 の表記規則またはデフォルトの Transact-SQL の表記規則のどちらでも使用できます。

例 quoted_identifier オプションを ON (Sybase IQ のデフォルト設定) にして動作するように選択した場合、SQL キーワード `user` を指定した次の文はいずれの DBMS でも有効です。

```
CREATE TABLE "user" (
    coll char(5)
) ;
INSERT "user" ( coll )
VALUES ( 'abcde' ) ;
```

quoted_identifier オプションを OFF (Adaptive Server Enterprise のデフォルト設定) にして動作するように選択した場合、次の文はいずれの DBMS でも有効です。

```
SELECT *
FROM Employees
WHERE Surname = "Chin"
```

検索条件

機能 WHERE 句、HAVING 句、CHECK 句、JOIN 句、または IF 式の検索条件を指定します。

構文

```
{ expression compare expression
| expression compare { ANY | SOME | ALL } ( subquery )
| expression IS [ NOT ] NULL
| expression [ NOT ] BETWEEN expression AND expression
| expression [ NOT ] LIKE expression [ ESCAPE expression ]
| expression [ NOT ] IN ( { expression | subquery |
... value-expr1 , value-expr2 [ , value-expr3 ] ... } )
| column-name [ NOT ] CONTAINS ( ... word1 [ , word2, ] [ , word3 ] ... )
| EXISTS ( subquery )
| NOT condition
| condition AND condition
| condition OR condition
| ( condition )
| ( condition , estimate )
| condition IS [ NOT ] { TRUE | FALSE | UNKNOWN } }
```

パラメータ

```
compare:
{ = | > | < | >= | <= | <> | != | !< | !> }
```

使用法 すべての場所

権限 データベースに接続しておく必要があります。

例 たとえば、次のクエリは最年長の従業員の名前と誕生年を取得します。

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate <= ALL (SELECT BirthDate FROM
Employees);
```

限定比較述部に比較値を提供するサブクエリは、複数のローを取得できますが、カラムは 1 つしか持つことができません。

関連する動作

なし

参照

[「式」\(25 ページ\)](#)

説明

条件を使ってテーブルからローのサブセットを選択したり、IF 文などの制御文中で条件を使ってフロー制御を行います。

SQL の条件式は、条件が真 (TRUE) または偽 (FALSE) であるブール論理には従いません。SQL では、すべての条件が TRUE、FALSE、または UNKNOWN のいずれかに評価されます。これを、3 値的論理といいます。比較対象となる値のいずれかが NULL の場合、比較結果は UNKNOWN になります。3 値的論理での論理演算子の組み合わせについては、[「3 値的論理」\(47 ページ\)](#) を参照してください。

比較結果が TRUE の場合のみ、ローは検索条件を満たしたことになります。比較結果が UNKNOWN のローは、検索条件を満たしません。詳細については、[「NULL 値」\(67 ページ\)](#) を参照してください。

サブクエリは、多数の検索条件で使用される式の重要なクラスを構成します。詳細については、[「検索条件内のサブクエリ」\(37 ページ\)](#) を参照してください。

以降の項で、異なるタイプの検索条件について説明します。

比較条件

比較条件の構文は、次のとおりです。

```
expression compare expression
```

compare は比較演算子です。Sybase IQ で使用可能な比較演算子を [表 2-4](#) に示します。

表 2-4 : Sybase IQ で使用可能な比較演算子

| 演算子 | 説明 |
|-----|-------|
| = | 等価 |
| > | より大きい |

| 演算子 | 説明 |
|-----|------------|
| < | より小さい |
| >= | より大きいまたは等価 |
| <= | より小さいまたは等価 |
| != | 等価ではない |
| <> | 等価ではない |
| !> | 以下 |
| !< | 以上 |

例 たとえば、次のクエリは最年長の従業員の名前と誕生年を取得します。

```
SELECT Surname, BirthDate
FROM Employees
WHERE Surname <= ALL (SELECT MIN(BirthDate) FROM
Employees);
```

限定比較述部に比較値を提供するサブクエリは、前掲の例のとおり複数のローを取得できませんが、カラムは1つしか持つことができません。

注意 すべての文字列比較は

- データベースが **case respect** (デフォルト値) として作成されている場合、大文字と小文字を区別します。
- データベースが **case ignore** として作成されている場合、大文字と小文字を区別しません。

大文字と小文字を区別しないデータベースでの比較結果の詳細については、『リファレンス：文とオプション』の「[CREATE DATABASE 文](#)」の「[使用法](#)」を参照してください。

互換性

- **後続ブランク** Adaptive Server Enterprise で比較を行う場合、文字データの後続ブランクは無視されます。文字列を比較するときの Sybase IQ の動作は、[Ignore Trailing Blanks in String Comparison] データベース作成オプションによって制御されます。
- **大文字小文字の区別** デフォルトでは、Sybase IQ データベースは Adaptive Server Enterprise データベースと同様、大文字小文字を区別するように作成されます。比較処理は、該当のデータベースの文字の区別に合わせて実行されます。Sybase IQ データベースで大文字と小文字を区別するかどうかは、データベースの作成時に制御できます。

検索条件内のサブクエリ

サブクエリとは、カッコに囲まれた **SELECT** 文です。このような **SELECT** 文には、リスト項目を 1 つだけ指定してください。

サブクエリが複数のローを返す場合、比較条件 ($>$ 、 $<$ 、 $!=$ など) によってカラムをサブクエリと比較できます。(カラムを 1 つ持つ) サブクエリがローを 1 つ返す場合、そのローの値が式と比較されます。サブクエリがローを返さない場合、その値は **NULL** です。

1 カラムと任意の数のローを返すサブクエリは、**IN** 条件、**ANY** 条件、**ALL** 条件、または **EXISTS** 条件で使用できます。以降の項で、これらの条件について説明します。

Sybase IQ では、非相関サブクエリの述部でのみ **UNION** を使用できません。スカラ値サブクエリまたは相関サブクエリの述部では使用できません。

サブクエリは、**CONTAINS** または **LIKE** 述部内では使用できません。

Sybase IQ では、単独の **OR** 句で複数のサブクエリを使用することはできません。たとえば、次のクエリには、**OR** でジョインされた 2 つのサブクエリがあります。

```
CREATE VARIABLE @ln int;

SELECT @ln = 1;select count(*) FROM lineitem

WHERE l_shipdate IN (select l_shipdate FROM lineitem
WHERE l_orderkey IN (2,4,6))

OR l_shipdate IN (select l_shipdate FROM lineitem WHERE
l_orderkey IN (1,3,5))

OR l_linenumber = @ln;
```

AND および **BETWEEN** でジョインされた同様のサブクエリも使用できます。

詳細については、「[比較条件](#)」(35 ページ)を参照してください。

サブクエリ述部の分離

SQL89 規格では、サブクエリ述部をいくつかの形式で指定できます。各サブクエリは、**WHERE** 句または **HAVING** 句内で他の述部とともに指定し、**AND** 演算子または **OR** 演算子を使用して結合できます。**Sybase IQ** では、相関 (外部クエリ内のテーブルへの参照を含んでおり、単独で評価できない) サブクエリまたは非相関 (リモート・テーブルへの参照を含まない) サブクエリをサポートしています。

サブクエリ述部の形式は次のとおりです。

- 非限定比較述部

```
<scalar-expression> <comparison-operator>
<subquery>
```

比較演算子は =、<>、>、>=、<、または <= です。

非限定比較サブクエリは値を 1 つだけ返します。サブクエリが複数の値を返すと、エラー・メッセージが表示されます。このタイプのクエリは、スカラ・サブクエリ述部とも呼ばれます。

- IN 述部

```
<scalar-expression> [NOT] IN <subquery>
```

IN サブクエリ述部は、値のリストまたは 1 つの値を返します。このタイプのクエリは、限定サブクエリ述部とも呼ばれます。

- 存在述部

```
[NOT] EXISTS <subquery>
```

EXISTS 述部はサブクエリの存在を示します。EXISTS <subquery> という式は、サブクエリ結果が空でない場合にのみ true と評価されます。EXISTS 述部は、外部クエリ・ブロック内のカラムや式と結果を比較しません。通常は関連サブクエリとともに使用されます。

- 限定比較述部

```
<scalar-expression> <comparison-operator> [ANY |
ALL] <subquery>
```

限定比較述部は、サブクエリから返された 1 つの値または値の集合を比較します。

実行できるクエリのタイプは次のとおりです。

- WHERE 句または HAVING 句内で垂直に実行できない、非関連スカラ・サブクエリまたは IN サブクエリの分離
- WHERE 句または HAVING 句内の関連/非関連 EXISTS サブクエリの分離
- WHERE 句または HAVING 句内の任意の関連/非関連スカラ・サブクエリ、IN または EXISTS サブクエリ、あるいは限定比較サブクエリの分離
- AND/OR (連結/分離) および単純な述部またはサブクエリ述部と組み合わせた任意の非関連/関連サブクエリ述部

- ビュー/抽出テーブルの上にあるサブクエリ述部の連結/分離
- UPDATE、DELETE、および SELECT INTO 文内のサブクエリ述部の分離

SUBQUERY_CACHING_PREFERENCE オプションを使用すると、経験豊富な DBA は使用するサブクエリ・キャッシュ方法を選択できます。詳細については、『リファレンス：文とオプション』の「SUBQUERY_CACHING_PREFERENCE オプション」を参照してください。

例

例 1 非相関 EXISTS サブクエリと IN サブクエリの分離

```
SELECT COUNT(*)
FROM supplier
WHERE s_suppkey IN (SELECT MAX(l_suppkey)
                    FROM lineitem
                    GROUP BY l_linenumber)
OR EXISTS (SELECT p_brand
           FROM part
           WHERE p_brand = 'Brand#43');
```

例 2 非相関 EXISTS サブクエリの分離

```
SELECT COUNT(*)
FROM supplier
WHERE EXISTS (SELECT l_suppkey
              FROM lineitem
              WHERE l_suppkey = 12345)
OR EXISTS (SELECT p_brand
           FROM part
           WHERE p_brand = 'Brand#43');
```

例 3 非相関スカラまたは IN サブクエリ述部の分離

```
SELECT COUNT(*)
FROM supplier
WHERE s_acctbal*10 > (SELECT MAX(o_totalprice)
                     FROM orders
                     WHERE o_custkey = 12345)
OR substring(s_name, 1, 6) IN (SELECT c_name
                              FROM Customers
                              WHERE c_nationkey = 10);
```

例 4 相関/非相関限定比較サブクエリの分離

```
SELECT COUNT(*)
FROM lineitem
WHERE l_suppkey > ANY (SELECT MAX(s_suppkey)
                      FROM supplier)
```

```

WHERE s_acctbal >100
GROUP BY s_nationkey)
OR l_partkey >= ANY (SELECT MAX(p_partkey)
FROM part
GROUP BY p_mfgr);

```

例 5 関連サブクエリ述部の分離

```

SELECT COUNT(*)
FROM supplier S
WHERE EXISTS (SELECT l_suppkey
FROM lineitem
WHERE l_suppkey = S.s_suppkey)

OR EXISTS (SELECT p_brand FROM part
WHERE p_brand = 'Brand#43'
AND p_partkey > S.s_suppkey);

```

サブクエリの実行がサポートされる前は、2つの部分にクエリを記述してから、**UNION** を使用して最終結果をマージする必要がありました。

次のクエリは、「例 5」(40 ページ) のクエリと同じ結果が得られるマージされたクエリを示しています。マージされたクエリは **supplier** テーブルを 2 回スキャンしてから、各 **UNION** からの結果をマージして最終結果を返すため、最適なパフォーマンスは得られません。

```

SELECT COUNT(*)
FROM (SELECT s_suppkey FROM supplier S
WHERE EXISTS (SELECT l_suppkey
FROM lineitem
WHERE l_suppkey = S.s_suppkey)

UNION

SELECT s_suppkey
FROM supplier S
WHERE EXISTS (SELECT p_brand
FROM part
WHERE p_brand = 'Brand#43'
AND p_partkey > S.s_suppkey)) as UD;

```

ALL または ANY 条件

ANY 条件の構文は、次のとおりです。

expression compare **ANY** (*subquery*)

compare は比較演算子です。

たとえば、等号演算子のある ANY 条件は、*expression* がサブクエリ結果のいずれかの値に等しい場合は TRUE、式が NULL ではなく、サブクエリのいずれのカラムにも当てはまらない場合は FALSE です。

expression = ANY (*subquery*)

expression が NULL 値の場合、サブクエリ結果にローがあれば ANY 条件は UNKNOWN です。サブクエリ結果にローがなければ、条件は必ず FALSE になります。

ANY の代わりにキーワード SOME を使用できます。

ALL 条件の構文は、次のとおりです。

expression compare ALL (*subquery*)

compare は比較演算子です。

制限事項

限定比較述部の左右どちらかに複数の式があると、エラー・メッセージが返されます。例を示します。

```
Subquery allowed only one select list item
```

このような種類のクエリは、IN サブクエリ、または MIN セット関数や MAX セット関数を使用するスカラ・サブクエリに常に置き換えることができます。

互換性

ANY および ALL サブクエリは、Adaptive Server Enterprise と Sybase IQ 間で互換性があります。SOME を ANY の同意語として扱うのは、Sybase IQ だけです。

BETWEEN 条件

BETWEEN 条件の構文は、次のとおりです。

expr [NOT] BETWEEN *start-expr* AND *end-expr*

BETWEEN 条件は TRUE、FALSE、または UNKNOWN として評価できます。NOT キーワードがない場合、*expr* が *start-expr* と *end-expr* の間にあれば、条件は TRUE と評価されます。NOT キーワードを使用すると条件の意味が逆になりますが、UNKNOWN は変わりません。

BETWEEN 条件は、次のような 2 つの不等式の組み合わせに相当します。

expr >= *start-expr* AND *expr* <= *end-expr*

BETWEEN 述部は、“A between B and C” の形式で使用します。“B” または “C”、あるいは “B” と “C” の両方でサブクエリを指定できます。“A” は値式またはカラムでなければなりません。

互換性

BETWEEN 条件は、Sybase IQ と Adaptive Server Enterprise の間で互換性があります。

LIKE 条件

LIKE 条件の構文は、次のとおりです。

expression [NOT] LIKE *pattern* [ESCAPE *escape-expr*]

LIKE 条件は TRUE、FALSE、または UNKNOWN として評価できます。LIKE は、文字列データのみで使用できます。

サブクエリは、LIKE 述部内では使用できません。

HG インデックスまたは LF インデックスが使用可能な場合、ワイルドカード以外の文字で始まる LIKE 述部の実行を高速化できます。

WD インデックスが使用可能な場合、特定の LIKE 述部の実行が高速化します。

NOT キーワードがない場合、*expression* が *pattern* に一致すれば、条件は TRUE として評価されます。*expression* または *pattern* が NULL 値の場合、この条件は UNKNOWN です。NOT キーワードを使用すると条件の意味が逆になりますが、UNKNOWN は変わりません。

パターンには、任意の数のワイルドカード文字を指定できます。ワイルドカード文字は次のとおりです。

| ワイルドカード | 一致条件 |
|-------------|---------------------------|
| _ (アンダースコア) | 任意の 1 文字 |
| % (パーセント記号) | 0 個以上の文字からなる任意の文字列 |
| [] | 指定した範囲または文字セット内の任意の 1 文字 |
| [^] | 指定した範囲または文字セット以外の任意の 1 文字 |

ワイルドカード以外の指定文字は正確に一致しなければなりません。

たとえば、次のような検索条件があるとします。

name LIKE 'a%b_'

名前が *a* で始まり、末尾から 2 つ目の文字が *b* のローの場合、TRUE になります。

escape-expr が指定されている場合は、エスケープ文字は1文字として評価されます。*pattern* 内のパーセント記号、アンダースコア、左側の角カッコ、または別のエスケープ文字の前にエスケープ文字を置くことによって、特殊文字に特別な意味を持たせないようにできます。このようにすると、パーセント記号はパーセント記号として、アンダースコアはアンダースコアとして検出されます。

126文字以下のパターンは、すべてサポートされています。254文字よりも長いパターンは、サポートされていません。127～254文字の長さのパターンは、パターンの内容によってサポートされる場合があります。

文字セットからの探索

探索の対象となる文字セットは、角カッコ内に並べて指定します。たとえば、次の条件は文字列 *smith* と *smyth* を検出します。

```
LIKE 'sm[iy]th'
```

文字範囲からの探索

探索対象の文字範囲は、角カッコ内に範囲を書いて指定します。範囲の始めと終わりの間にハイフンを書きます。たとえば、次の条件は、文字列 *bough* と *rough* を検出しますが、*tough* は検出しません。

```
LIKE '[a-r]ough'
```

文字の範囲 [a-z] は「a以上z以下」と解釈され、データベースの照合では大なり演算と小なり演算が実行されます。照合順内の文字の順序の詳細については、『システム管理ガイド第1巻』の「[第11章 各国語と文字セット](#)」を参照してください。

範囲には、まず小さい方の値、次に大きい方の値を指定してください。たとえば、式 [z-a] がある LIKE 条件は、ローを返しません。これは、[z-a] の範囲に一致する文字がないためです。

データベースが大文字と小文字を区別するように作成されていないかぎり、文字の範囲では大文字と小文字を区別しません。たとえば、次の条件は文字列 *Bough*、*rough*、*TOUGH* を検出します。

```
LIKE '[a-z]ough'
```

データベースが大文字と小文字を区別するように作成されている場合は、検索条件でも大文字と小文字を区別します。

文字範囲と文字セットによる探索

文字範囲と文字セットを組み合わせて角カッコ内に指定できます。たとえば、次の条件は文字列 *bough*、*rough*、*tough* を検出します。

```
LIKE '[a-rt]ough'
```

角カッコ [a-mpqs-z] は、「aからmの範囲内の1文字か、pまたはqか、またはsからzの範囲内の1文字」と解釈されます。

範囲外の 1 文字の探索

脱字記号 (^) は、探索から除外する文字範囲を指定します。たとえば、次の条件は文字列 *tough* を検出しますが、文字列 *rough* や *bough* は検出しません。

```
LIKE '[^a-r]ough'
```

脱字記号は、カッコ内の内容をすべて否定します。たとえば、角カッコ `[^a-mpqs-z]` は、「*a* から *m* の範囲外で、*p* でも *q* でもなく、*s* から *z* の範囲外の 1 文字」と解釈されます。

特殊な文字範囲と文字セット

角カッコ内の任意の 1 文字は、その文字を指しています。たとえば、`[a]` は *a* という文字のみに一致します。`[^]` は脱字記号のみに一致し、`[%]` はパーセント文字 (この場合、パーセント文字はワイルドカード文字としては機能しません) のみに一致し、`[_]` はアンダースコア文字のみに一致します。また、`[[]]` は文字 `[` のみに一致します。

その他の特殊なケースには、次のものがあります。

- 式 `[a-]` は、文字 *a* または `-` に一致します。
- 式 `[]` は、一致することがないのでローを返しません。
- 式 `[` または `[abp-q` は、正しくないので構文エラーになります。
- 角カッコ内にワイルドカード文字を使用できません。式 `[a%b]` では、*a*、`%` または *b* を検出します。
- 脱字記号を使用しても、カッコ内の先頭になれば範囲を否定できません。式 `[a^b]` では、*a*、`^` または *b* を検出します。

互換性

ESCAPE 句をサポートしているのは Sybase IQ のみです。

IN 条件

IN 条件の構文は、次のとおりです。

```
{ expression [ NOT ] IN ( subquery )  
  | expression [ NOT ] IN ( expression )  
  | expression [ NOT ] IN ( value-expr1 , value-expr2  
  [, value-expr3 ] ... ) }
```

NOT キーワードがない場合、*expression* がリストされた値のいずれかに一致すれば IN 条件は TRUE、*expression* が NULL 値の場合は UNKNOWN、それ以外の場合は FALSE です。NOT キーワードを使用すると条件の意味が逆になりますが、UNKNOWN は変わりません。

IN 条件リスト内の値の最大数は 250,000 です。

互換性

IN 条件は、Adaptive Server Enterprise と Sybase IQ の間で互換性があります。

CONTAINS 条件

CONTAINS 条件の構文は、次のとおりです。

```
{ column-name [ NOT ] CONTAINS ( ( word1 [ , word2 ] [ , word3 ] ... )
```

column-name は、ベース・テーブル内の CHAR カラム、VARCHAR カラム、または LONG VARCHAR (CLOB) カラムのいずれかであり、WD インデックスを持つ必要があります。*word1*、*word2*、*word3* 式は、255 バイト以内の文字列定数で、それぞれ 1 単語を含む必要があります。この単語の長さは、カラムのワード・インデックスに許可されている最大長を超えてはなりません。

NOT キーワードがない場合、*column-name* が各単語を含んでいれば CONTAINS 条件は TRUE、*column-name* が NULL 値の場合は UNKNOWN、それ以外の場合は FALSE です。NOT キーワードを使用するとこれらの値が逆になりますが、UNKNOWN は変わりません。

たとえば、次のような検索条件があるとします。

```
varchar_col CONTAINS ('cat', 'mat')
```

これは、*varchar_col* の値が The cat is on the mat であれば TRUE です。*varchar_col* の値が The cat chased the mouse である場合は、この条件は FALSE です。

Sybase IQ が LIKE と CONTAINS の両方を含む文を実行するとき、CONTAINS 条件が優先されます。

ユーザ定義関数を含むビューで、CONTAINS 述部を使用すると CONTAINS の基準が無視されるため、使用しないでください。代わりに、LIKE 述部とワイルドカードを使用するか、ビューの外部でクエリを発行します。

EXISTS 条件

EXISTS 条件の構文は、次のとおりです。

```
EXISTS( subquery )
```

EXISTS 条件は、サブクエリ結果にローが少なくとも 1 つあれば TRUE で、ローがなければ FALSE です。EXISTS 条件には、UNKNOWN がありません。

互換性 EXISTS 条件は、Adaptive Server Enterprise と Sybase IQ の間で互換性があります。

IS NULL 条件

IS NULL 条件の構文は、次のとおりです。

expression IS [NOT] NULL

NOT キーワードがない場合、式が NULL 値なら IS NULL 条件は TRUE、それ以外の場合は FALSE です。NOT キーワードを使用すると条件の意味が逆になります。

互換性 IS NULL 条件は、Adaptive Server Enterprise と Sybase IQ の間で互換性があります。

論理演算子を使用した条件

検索条件は、AND、OR、NOT を使用して結合できます。

AND を使用して、次のように条件を結合します。

condition1 AND *condition2*

両方の条件が TRUE の場合、結合した条件は TRUE になります。一方の条件が FALSE の場合、結合した条件は FALSE になります。それ以外の場合は、結合した条件は UNKNOWN になります。

OR を使用して、次のように条件を結合します。

condition1 OR *condition2*

両方の条件が TRUE の場合、結合した条件は TRUE になります。一方の条件が FALSE の場合、結合した条件は FALSE になります。それ以外の場合は、結合した条件は UNKNOWN になります。*condition1* と *condition2* のどちらの条件が先に評価されるかという決まった順序はありません。

詳細については、「サブクエリ述部の分離」(37 ページ) を参照してください。

互換性 AND および OR 演算子は、Sybase IQ と Adaptive Server Enterprise の間で互換性があります。

NOT 条件

NOT 条件の構文は、次のとおりです。

NOT *condition1*

condition1 が FALSE の場合、NOT 条件は TRUE です。*condition1* が TRUE の場合は FALSE、*condition1* が UNKNOWN の場合は UNKNOWN になります。

真理値条件

真理値条件の構文は、次のとおりです。

IS [NOT] *truth-value*

NOT キーワードがない場合、*condition* が指定の *truth-value* (TRUE、FALSE、UNKNOWN のいずれか) と評価されれば条件は TRUE になります。それ以外の場合、値は FALSE です。NOT キーワードを使用すると条件の意味が逆になりますが、UNKNOWN は変わりません。

互換性

真理値条件がサポートされているのは、Sybase IQ のみです。

3 値的論理

次の表は、3 値的論理で SQL の論理演算子 AND、OR、NOT、および IS がどのように機能するかを示します。

AND 演算子

| AND | TRUE | FALSE | UNKNOWN |
|---------|---------|-------|---------|
| TRUE | TRUE | FALSE | UNKNOWN |
| FALSE | FALSE | FALSE | FALSE |
| UNKNOWN | UNKNOWN | FALSE | UNKNOWN |

OR 演算子

| OR | TRUE | FALSE | UNKNOWN |
|---------|------|---------|---------|
| TRUE | TRUE | TRUE | TRUE |
| FALSE | TRUE | FALSE | UNKNOWN |
| UNKNOWN | TRUE | UNKNOWN | UNKNOWN |

NOT 演算子

| TRUE | FALSE | UNKNOWN |
|-------|-------|---------|
| FALSE | TRUE | UNKNOWN |

IS 演算子

| IS | TRUE | FALSE | UNKNOWN |
|---------|-------|-------|---------|
| TRUE | TRUE | FALSE | FALSE |
| FALSE | FALSE | TRUE | FALSE |
| UNKNOWN | FALSE | FALSE | TRUE |

ユーザ指定の条件ヒント

Sybase IQ クエリ・オプティマイザは、使用可能なインデックスからの情報を使用して、クエリを実行するための適切な方式を選択します。クエリ内の各条件について、オプティマイザはインデックスを使用して条件を実行できるかどうかを決定します。条件を実行できる場合、オプティマイザはインデックスを選択し、そのテーブル上の他の条件に対する順序を決定します。これらの決定で最も重要な要因になるのは、条件の選択性、つまり条件を満たすテーブル・ローの端数です。

オプティマイザは通常、ユーザの介入なしに、一般的に最適な決定を行います。ただし、状況によっては、オプティマイザが条件の実行前にその選択性を正確に決定できない場合があります。これらの状況は通常、条件が適切なインデックスを使用できないカラムを対象としている場合、または算術演算または関数式が含まれるために条件が複雑すぎてオプティマイザが正確に予測できない場合に発生します。

頻繁に実行されるクエリが存在する場合、最適な実行方式を選択するために役立つ追加情報をオプティマイザに提供することによりクエリのパフォーマンスが向上するかどうかを実際に試してみたい場合があります。

ユーザ指定の条件の選択性

条件ヒントの最も簡単な形式は、オプティマイザが計算する値の代わりに使用される選択性の値を指定することです。

選択性ヒントは、クエリ・テキスト内で条件をカッコで囲むことにより指定します。次に、カッコ内の条件の後に、カンマと、選択性として使用する数値を追加します。

この選択性の値はテーブル・ローのパーセンテージとして表され、必要な条件を満たします。したがって、選択性の有効な数値は、100.0 ~ 0.0 です。

注意 クエリ・プランでは、選択性はパーセンテージではなく端数として表されます。したがって、ユーザ指定の選択性が 35.5 である場合、クエリ・プランでの選択性は 0.355000 となります。

例

- 次のクエリは、`ship_date` 値の 1.5% が 1994/06/30 より前であるという予測値を提供しています。

```
SELECT ShipDate
FROM SalesOrderItems
WHERE ( ShipDate < '2001/06/30', 1.5 )
ORDER BY ShipDate DESC
```

- 次のクエリでは、該当するロー全体の 0.5% が条件を満たすという予測値を提供しています。

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 0.5)
AND c.ID = o.customerID
```

端数のパーセンテージにより、ユーザはより正確な見積もりを指定できます。これは特に、大きなテーブルで重要になります。

互換性

SQL Anywhere Studio では、ユーザ指定の選択性が見積もりがサポートされます。

Adaptive Server Enterprise では、ユーザ指定の選択性が見積もりはサポートされていません。

ユーザ指定の条件ヒント文字列

Sybase IQ ではまた、ユーザ指定の選択性が見積もりのサポートに加えて、ユーザが条件ヒント文字列を使用してオプティマイザに追加ヒント情報を指定できます。これらの条件単位のヒント文字列により、ユーザは条件の実行設定を追加で指定できます。オプティマイザは、可能であればこの設定に従います。これらの設定には、条件に使用するインデックス、条件の選択性、条件の実行時の実行フェーズ、および 1 実行フェーズ内で実行される条件セット間の順序に影響する条件の有用性が含まれます。

ユーザ指定の選択性が見積もりなどの条件ヒント文字列は、クエリのテキスト内で条件をカッコで囲むことにより指定します。次に、このカッコ内の条件の後に、カンマを追加し、適切なヒントを含む文字列を引用符で囲んで指定します。この引用符で囲まれた文字列内では、各ヒントはヒント・タイプ識別子として表され、それにコロんとヒント・タイプ値が続きます。同じヒント文字列内に複数のヒントがある場合は、ヒントをカンマで区切ります。また、複数のヒントの順序は任意です。ヒント文字列内では、2 つの要素の間に空白文字を挿入できます。

サポートされるヒント・タイプには、次の4種類があります。

- [選択性ヒント](#)。ユーザ指定の選択性が見積もりと等価です
- [インデックス設定ヒント](#)
- [実行フェーズ・ヒント](#)
- [有用性ヒント](#)

選択性ヒント

ヒント文字列内に含めることのできる最初のヒント・タイプは、選択性ヒントです。選択性ヒントは、“S”または“s”のいずれかのヒント・タイプ識別子により識別されます。ユーザ指定の選択性が見積もりの場合と同様に、この選択性の値は常にテーブル・ローのパーセンテージとして表され、必要な条件を満たします。

例 次の例は、「[ユーザ指定の条件の選択性](#)」(48 ページ) の2番目の例とまったく同じです。

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 's: 0.5)
      AND c.ID = o.CustomerID
```

インデックス設定ヒント

サポートされる次のヒント・タイプは、インデックス設定ヒントです。これは、“I”または“i”のいずれかのヒント・タイプ識別子により識別されます。インデックス設定ヒントの値は -10 ~ 10 の整数です。正の整数値は、特定のインデックス・タイプが優先されることを意味し、負の値は特定のインデックス・タイプが回避されることを意味します。

インデックス設定ヒントの影響は、設定がクエリ内のすべての条件ではなく、関連する条件にのみ適用される点を除いて、INDEX_PREFERENCE オプションと同じです。指定のインデックス・タイプが該当カラムに存在し、関連条件の評価時に該当インデックス・タイプを使用できる場合、インデックス設定は条件の実行にのみ影響を及ぼすことができます。すべてのインデックス・タイプがすべての条件で使用できるわけではありません。-10 ~ 10 の整数のそれぞれの意味については、『リファレンス: 文とオプション』の「[第2章 データベース・オプション](#)」の「[INDEX_PREFERENCE オプション](#)」を参照してください。

例 次の例は、3%の選択性を指定し、可能であれば HG インデックスを使用して条件が評価されることを示します。

```
SELECT *
```

```
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 'S:3.00, I:+2')
      AND c.ID = o.CustomerID
```

次の例は、37.5%の選択性を指定し、可能であればHGインデックスを使用して条件が評価されないことを示します。

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (o.SalesRepresentative > 1000.0, 'i:-2,
s:37.500')
      AND c.ID = o.CustomerID
```

実行フェーズ・ヒント

サポートされる第3のヒント・タイプは、実行フェーズ・ヒントです。これは、“E”または“e”のいずれかのヒント・タイプ識別子により識別されます。

Sybase IQ クエリ・エンジン内には、条件を評価できる不変、遅延、バインド、水平という別個の実行フェーズがあります。

オプティマイザはデフォルトで、条件の評価のために必要な情報がすべて使用可能である最初の実行フェーズで、各条件を評価することを選択します。したがって、各条件は評価されるデフォルトの実行フェーズを持ちます。

必要な情報が使用可能になるまで条件を評価できないので、実行フェーズ・ヒントはデフォルトのフェーズより後のフェーズまで条件の実行を遅延する目的でのみ使用できます。実行フェーズ・ヒントを使用して、デフォルトのフェーズより前のフェーズで強制的に条件を評価することはできません。

次に、4つの条件実行フェーズについて、実行される順に説明します。

不変 単一のカラム(または同じテーブルの2つのカラム)を参照し、インデックスを使用して評価できる条件は一般的に、単純な不変条件と呼ばれます。単純な不変条件は通常、最適化処理内で早期に評価されます。

つまり、これらの不変条件をすべて満たすローの数を使用して、オプティマイザが、使用する最適なジョイン順序とジョイン・アルゴリズムを決定できることを意味します。これは最初の実行フェーズであるため、ユーザは条件をこのフェーズで強制的に実行することはできません。ただし、このフェーズからその後のフェーズに条件を実行させることはできます。

遅延 一部の条件は、他のクエリ部分が実行されるまで評価できません。この遅延条件は、関連するクエリ・ノードが最初にフェッチされたときに一度評価されます。これらの条件は、非関連のサブクエリ条件と、オプティマイザによって作成される IN または PROBABLY_IN のプッシュダウン・ジョイン条件の、2つのカテゴリに分類されます。

バインド 一部の条件は、複数回評価する必要があります。これらの条件は一般的に、関連サブクエリ内の外部参照を含む条件と、オプティマイザによって作成されるプッシュダウン等号ジョイン条件の、2つのカテゴリに分類されます。たとえば、外部参照条件は、クエリの実行中に外部参照値が変更されるたびに再評価されます。

水平 1つのテーブルの複数のカラムを含む条件など、一部の条件はインデックスを使用するのではなく、一度に1つずつローを評価する必要があります。

実行フェーズ・ヒントには、条件を評価する実行フェーズを識別する値を指定します。各値は、大文字と小文字が区別されない単一の文字です。

- D—遅延
- B—バインド
- H—水平

例 次の例の条件ヒント文字列は、条件を「遅延」実行フェーズに移動し、可能であれば LF インデックスを使用して条件を評価することを示します。

```
SELECT *  
FROM Customers c, SalesOrders o  
WHERE (o.SalesRepresentative > 10000.0, 'E:D, I:1')  
      AND c.id = o.CustomerID
```

有用性ヒント

サポートされる最後のヒント・タイプは、有用性ヒントです。これは、“U” または “u” のいずれかのヒント・タイプ識別子により識別されます。有用性ヒントの値は、0.0 ~ 10.0 の任意の数値です。オプティマイザでは、有用性の値は条件ごとに計算されます。この有用性の値は、同じ実行フェーズ内の条件セットの評価順序を決定するために使用されます。有用性の値が大きいほど、評価順序が前になります。ユーザは、有用性ヒントを指定することにより、評価順序内の特定位置に条件を配置できます。ただし、それによって、条件が評価される実行フェーズを変更することはできません。

例 次の例の条件ヒント文字列は、条件を「遅延」実行フェーズに移動し、「遅延」フェーズ内で有用性を 3.25 に設定することを示します。

```
SELECT *
FROM Customers c, SalesOrders o
WHERE (co.SalesRepresentative > 10000.0, 'U: 3.25, E:
D')
AND c.id = o.CustomerID
```

互換性 SQL Anywhere Studio では、ユーザ指定の条件ヒント文字列はサポートされていません。

Adaptive Server Enterprise では、ユーザ指定の条件ヒント文字列はサポートされていません。

ジョイン等号条件に関するユーザ指定のヒント

ジョイン・アルゴリズムの優先順位を指定できます。この順位によってクエリ内のすべてのジョインが影響を受けるとは限りません。

単純な等号ジョイン述部に述部ヒントのタグを付けることができます。このヒントにより、まさにその1つのジョインのためにジョインの優先順位を指定できます。ローカルなジョインの優先順位が設定されたジョイン条件が、同じジョインに複数あり、しかもそれらのヒントの値が異なる場合、そのジョインに対するローカルな優先順位がすべて無視されます。ローカルなジョインの優先順位は、オプティマイザが選択したジョインの順序に影響を与えません。

次の例はハッシュ・ジョインを要求します。

```
AND (T.X = 10 * R.x, 'J:4')
```

ユーザ指定の条件ヒントの使用ガイドライン

条件ヒントは一般的に、頻繁に実行するクエリでのみ使用します。

条件ヒントの試行は経験豊富なユーザのみが行ってください。一般的に、使用可能なインデックスからの条件に関する正確な情報を推定できない場合を除いて、オプティマイザが最適な決定を行います。

オプティマイザは頻繁に元の条件を書き直したり簡略化したりします。また、元の条件から新しい条件を推定します。条件ヒントは、オプティマイザにより推定された条件に至るまで、または簡略化された条件に至るまで、新規に実行されることはありません。

特別値

特別値は、式の中で使用したり、テーブル作成時にカラムのデフォルトとして使用したりできます。

CURRENT DATABASE 特別値

| | |
|------|---------------------------------------|
| 機能 | CURRENT DATABASE は、現在のデータベースの名前を返します。 |
| データ型 | STRING |
| 参照 | 「式」(25 ページ) |

CURRENT DATE 特別値

| | |
|------|---------------------------------------------------------------------|
| 機能 | 現在の年月日。 |
| データ型 | DATE |
| 参照 | 「式」(25 ページ) 「日付と時刻のデータ型」(83 ページ) |

CURRENT PUBLISHER 特別値

| | |
|------|---------------------------------------------------------------------------|
| 機能 | CURRENT PUBLISHER は、SQL Remote レプリケーション用データベースのパブリッシャ・ユーザ ID を含む文字列を返します。 |
| データ型 | STRING CURRENT PUBLISHER は、文字データ型のカラムにデフォルト値として使用できます。 |
| 参照 | 「式」(25 ページ) |

CURRENT TIME 特別値

| | |
|------|--------------------------|
| 機能 | 現在の時、分、秒(小数位あり)で構成される時刻。 |
| データ型 | TIME |

| | |
|----|-----------------------------------------------------------------------|
| 説明 | 秒の小数位は 6 桁まで格納されますが、現在の時刻の精度はシステム・クロックの精度によって制限されます。 |
| 参照 | 「式」 (25 ページ) 「日付と時刻のデータ型」 (83 ページ) |

CURRENT_TIMESTAMP 特別値

| | |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | CURRENT DATE と CURRENT TIME を結合して形成した、年、月、日、時、分、秒 (小数位あり) から成る TIMESTAMP 値。CURRENT TIME と同様に、秒の小数位の精度はシステム・クロックによって制限されます。 CURRENT_TIMESTAMP のデフォルト値は 3 桁です。 |
| データ型 | TIMESTAMP |
| 参照 | 「式」 (25 ページ) 「日付と時刻のデータ型」 (83 ページ) |

CURRENT_USER 特別値

| | |
|------|----------------------------------------------------------|
| 機能 | CURRENT_USER は、現在の接続のユーザ ID を含む文字列を返します。 |
| データ型 | STRING CURRENT_USER は、文字データ型のカラムにデフォルト値として使用できます。 |
| 説明 | UPDATE では、CURRENT_USER のデフォルト値を持つカラムは変更されません。 |
| 参照 | 「式」 (25 ページ) |

LAST_USER 特別値

| | |
|------|--------------------------------------------------------|
| 機能 | ローを最後に更新したユーザの名前。 |
| データ型 | STRING LAST_USER は、文字データ型のカラムにデフォルト値として使用できません。 |

| | |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 説明 | <p>INSERT および LOAD では、この定数は CURRENT USER と同じ効果があります。UPDATE では、LAST USER のデフォルト値を持つカラムが明示的に変更されない場合、現在のユーザの名前に変更されます。</p> <p>LAST USER のデフォルト値を DEFAULT TIMESTAMP と組み合わせて使用すると、ローを最後に変更したユーザと日時の両方を (別々のカラムに) 記録できます。</p> |
| 参照 | <p>「CURRENT USER 特別値」(55 ページ)</p> <p>「CURRENT TIMESTAMP 特別値」(55 ページ)</p> <p>『リファレンス：文とオプション』の「CREATE TABLE 文」</p> |

SQLCODE 特別値

| | |
|------|-------------------------------------------------------------------------------------------|
| 機能 | 現在の SQLCODE 値。 |
| データ型 | STRING |
| 説明 | SQLCODE 値は各文の後に設定されます。SQLCODE をチェックして、文の実行が成功したかどうかを確認することができます。 |
| 参照 | <p>「式」(25 ページ)</p> <p>『システム管理ガイド 第 2 巻』の「第 1 章 SQL 文」</p> |

SQLSTATE 特別値

| | |
|------|-------------------------------------------------------------------------------------------|
| 機能 | 現在の SQLSTATE 値。 |
| データ型 | STRING |
| 説明 | SQLSTATE 値は各文の後に設定されます。SQLSTATE をチェックして、文の実行が成功したかどうかを確認することができます。 |
| 参照 | <p>「式」(25 ページ)</p> <p>『システム管理ガイド 第 2 巻』の「第 1 章 SQL 文」</p> |

TIMESTAMP 特別値

| | |
|------|----------------------------------------|
| 機能 | TIMESTAMP は、テーブル内の各ローが最後に更新された日時を示します。 |
| データ型 | TIMESTAMP |

説明 DEFAULT TIMESTAMP によりカラムを宣言した場合、挿入処理およびロード処理のデフォルト値が提供されます。この値は、ローが更新されたときに常に、最新の日時に更新されます。

INSERT および LOAD では、DEFAULT TIMESTAMP は CURRENT TIMESTAMP と同じ効果があります。UPDATE では、TIMESTAMP のデフォルト値を持つカラムが明示的に変更されなければ、カラムの値は現在の日時に変更されません。

注意 Sybase IQ は、UTC TIMESTAMP または CURRENT UTC TIMESTAMP の DEFAULT 値をサポートしません。また、データベース・オプション DEFAULT_TIMESTAMP_INCREMENT もサポートしません。UTC TIMESTAMP または CURRENT UTC TIMESTAMP タイプのカラムで DEFAULT 値の挿入または更新が試みられるたびに、Sybase IQ はエラーを生成します。

参照 [「日付と時刻のデータ型」\(83 ページ\)](#)

USER 特別値

機能 USER は、現在の接続のユーザ ID を含む文字列を返します。

データ型 STRING

USER は、文字データ型のカラムにデフォルト値として使用できます。

説明 UPDATE では、USER のデフォルト値を持つカラムは変更されません。

参照 [「式」\(25 ページ\)](#)

[「CURRENT USER 特別値」\(55 ページ\)](#)

[「LAST USER 特別値」\(55 ページ\)](#)

変数

Sybase IQ は、3 つのレベルの変数をサポートしています。

- **ローカル変数** DECLARE 文を使用して、プロシージャまたはバッチ内の複合文に定義します。ローカル変数を指定できるのは複合文の中だけです。

- **接続レベル変数** CREATE VARIABLE 文で定義します。現在の接続に属しており、データベース接続を切断するか、DROP VARIABLE 文を使用すると消去されます。
- **グローバル変数** グローバル変数には、システム定義の値が指定されています。

ローカル変数と接続レベル変数は、ユーザが宣言します。これらの変数を SQL 文のプロシージャまたはバッチ内で使用して、情報を保持できます。グローバル変数は、システム定義の値が設定されるシステム定義の変数です。グローバル変数の名前はすべて、2つのアット・マーク (@) で始まります。たとえば、グローバル変数 @@version の値は、データベース・サーバの現在のバージョン番号です。ユーザはグローバル変数を定義できません。

ローカル変数

ローカル変数は DECLARE 文で宣言し、複合文 (BEGIN キーワードと END キーワードで囲まれた部分) 内でのみ使用できます。変数の初期設定値は NULL です。SET 文を使用して変数の値を設定できます。また、INTO 句のある SELECT 文を使用して値を代入できます。

DECLARE 文の構文は、次のとおりです。

```
DECLARE variable-name data-type
```

ローカル変数は、プロシージャが複合文中から呼び出されるかぎり、プロシージャに引数として引き渡すことができます。

例

- 次のバッチは、ローカル変数の使用例を示します。

```
BEGIN
DECLARE local_var INT ;
SET local_var = 10 ;
MESSAGE 'local_var = ', local_var ;
END
```

ISQL からこのバッチを実行すると、サーバのウィンドウに次のメッセージが表示されます。

```
local_var = 10
```

- 変数 local_var は、変数が宣言された複合文の外側には存在しません。次のバッチは無効で、「カラムが見つかりません」というエラーになります。

```
-- このバッチは無効です。
BEGIN
```

```

DECLARE local_var INT ;
SET local_var = 10 ;
MESSAGE 'local_var = ', local_var ;
END;
MESSAGE 'local_var = ', local_var ;

```

- 次の例は、INTO 句のある SELECT 文を使用してローカル変数の値を設定する方法を示します。

```

BEGIN
DECLARE local_var INT ;
SELECT 10 INTO local_var ;
MESSAGE 'local_var = ', local_var ;
END

```

ISQL からこのバッチを実行すると、サーバのウィンドウに次のメッセージが表示されます。

```

local_var = 10

```

互換性

- **名前** Adaptive Server Enterprise と Sybase IQ は、いずれもローカル変数をサポートしています。Adaptive Server Enterprise では、すべての変数名の先頭にアット・マーク (@) が付いています。Sybase IQ では、@ プレフィックスはオプションです。互換性のある SQL 文を書く場合は、変数名の先頭に必ずアット・マーク (@) を付けます。
- **スコープ** Sybase IQ と Adaptive Server Enterprise では、ローカル変数のスコープが異なります。Sybase IQ では、バッチ内でのローカル変数の宣言に DECLARE 文を使用できます。ただし、DECLARE が複合文で実行される場合、変数のスコープは複合文内に制限されます。
- **宣言** Sybase IQ では、各 DECLARE 文で宣言できる変数は 1 つのみです。Adaptive Server Enterprise の場合は、1 つの文中に複数の変数を宣言できます。

接続レベル変数

接続レベル変数は、CREATE VARIABLE 文で宣言します。CREATE VARIABLE 文は、複合文内でなければ任意の位置で使用できます。接続レベル変数は、プロシージャにパラメータとして引き渡すことができます。

CREATE VARIABLE の構文は、次のとおりです。

```

CREATE VARIABLE variable-name data-type

```

変数が作成されると、値は NULL に初期設定されます。接続レベル変数の値は、ローカル変数と同じように、SET 文または INTO 句のある SELECT 文を使用して設定できます。

接続レベル変数は、接続が終了するまで、または DROP VARIABLE 文を使用して変数が明示的に削除されるまで存在します。次の文は変数 *con_var* を削除します。

```
DROP VARIABLE con_var
```

- 例
- 次の SQL 文のバッチは、接続レベル変数の使用例を示します。

```
CREATE VARIABLE con_var INT;  
SET con_var = 10;  
MESSAGE 'con_var = ', con_var;
```

ISQL からこのバッチを実行すると、サーバのウィンドウに次のメッセージが表示されます。

```
con_var = 10
```

互換性

Adaptive Server Enterprise では、接続レベル変数はサポートされていません。

グローバル変数

グローバル変数は、Sybase IQ によって設定される値を持ちます。たとえば、グローバル変数 *@@version* の値は、データベース・サーバの現在のバージョン番号です。

グローバル変数は、名前の先頭に付けられた 2 つのアット・マーク (@) によって、ローカル変数および接続レベル変数と区別されます。たとえば、*@@error* はグローバル変数です。ユーザは、グローバル変数を作成したり、その値を直接更新したりできません。

一部のグローバル変数 (*@@spid* など) は、接続に固有の情報と値を保持します。その他の変数 (*@@connections* など) は、すべての接続に共通の値を保持します。

グローバル変数と特殊定数

特殊定数 (CURRENT DATE、CURRENT TIME、USER、SQLSTATE など) は、グローバル変数に類似しています。

次の文は、グローバル変数 *version* の値を取得します。

```
SELECT @@version
```

プロシージャでは、グローバル変数を変数リストに選択できます。次のプロシージャは、*ver* パラメータにサーバのバージョン番号を返します。

```
CREATE PROCEDURE VersionProc ( OUT ver
                               VARCHAR ( 100) )
BEGIN
    SELECT @@version
    INTO ver;
END
```

Embedded SQL では、グローバル変数をホスト変数リストに選択できます。

グローバル変数の
リスト

表 2-5 に、Sybase IQ で使用できるグローバル変数を示します。

表 2-5 : Sybase IQ グローバル変数

| 変数名 | 意味 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>@@error</i> | 通常、直前に実行された文のエラー・ステータス (成功または失敗) のチェックに使用する。直前のトランザクションが成功していれば値は 0 で、それ以外の場合、システムが生成した最新のエラー番号が設定される。エラーが発生した場合、 <i>if @@error != 0 return</i> のような文によって終了する。すべての SQL 文は <i>@@error</i> をリセットするため、実行の成否を判断する文の直後にステータス・チェックを行うこと。 |
| <i>@@fetch_status</i> | 最後の FETCH 文によって得られたステータス情報を保持する。 <i>@@fetch_status</i> には、次のいずれかの値が含まれる。 <ul style="list-style-type: none"> • 0 - fetch 文が正常に終了した。 • -1 - fetch 文がエラーになった。 • -2 - 結果セットにこれ以上データがない。 この機能は、返す値が異なる点を除いて <i>@@sqlstatus</i> と同様。これは Microsoft SQL Server との互換性のため。 |

| 変数名 | 意味 |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>@@identity</code> | insert、load、update 文により Identity/Autoincrement カラムに挿入された最後の値。ローがテーブルに挿入されると、 <code>@@identity</code> は必ずリセットされる。文が複数のローを挿入した場合、 <code>@@identity</code> は、最後に挿入されたローの Identity/Autoincrement 値を反映する。関連するテーブルに Identity/Autoincrement カラムがない場合、 <code>@@identity</code> は 0 に設定される。insert、load、update 文が正常に実行されなかったり、失敗した文を含むトランザクションがロールバックされても、 <code>@@identity</code> の値には影響しない。 <code>@@identity</code> は、Identity/Autoincrement カラムに最後の値を挿入した文がコミットに失敗した場合でも、その値を保持する。 |
| <code>@@isolation</code> | 現在の独立性レベル。 <code>@@isolation</code> には、アクティブ・レベルの値が入る。 |
| <code>@@procid</code> | 現在実行中のプロシージャのストアド・プロシージャ ID。 |
| <code>@@servername</code> | 現在のデータベース・サーバの名前。 |
| <code>@@sqlstatus</code> | 最後の FETCH 文によって得られたステータス情報を保持する。 |
| <code>@@version</code> | 現在の Sybase IQ のバージョン番号。 |

互換性

表 2-6 に、Sybase IQ でサポートされているすべての Adaptive Server Enterprise グローバル変数を示します。リストにない Adaptive Server Enterprise グローバル変数は、Sybase IQ ではサポートされていません。表 2-5 と対照的に、このリストに記載のグローバル変数はすべて値を返します。値には、NULL、1、-1、0 などの固定値のほか、意味のないものも含まれます。

表 2-6 : Sybase IQ がサポートする ASE グローバル変数

| グローバル変数 | 戻り値 |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>@@char_convert</code> | 0 を返す。 |
| <code>@@client_csname</code> | Adaptive Server Enterprise では、クライアントの文字セット名。クライアントの文字セットが一度も初期化されていない場合は NULL が設定され、それ以外の場合は、直前に使用された文字セットの名前が入る。Sybase IQ では NULL が返される。 |

| グローバル変数 | 戻り値 |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>@@client_csid</code> | Adaptive Server Enterprise では、クライアントの文字セット ID。クライアントの文字セットが一度も初期化されていない場合は、-1 に設定され、それ以外の場合は、直前に使用された文字セットの ID (syscharset から) が入る。Sybase IQ では -1 が返される。 |
| <code>@@connections</code> | 最後にサーバが起動されてからのログイン数。 |
| <code>@@cpu_busy</code> | Adaptive Server Enterprise では、最後に Adaptive Server Enterprise が起動されて以来 CPU が Adaptive Server Enterprise の作業に費やした時間数 (チック単位) を示す。Sybase IQ では 0 が返される。 |
| <code>@@error</code> | 通常、直前に実行された文のエラー・ステータス (成功または失敗) のチェックに使用する。直前のトランザクションが成功していれば値は 0 で、それ以外の場合、システムが生成した最新のエラー番号が設定される。次のような文の場合、 <pre>if @@error != 0 return</pre> エラーが発生すると終了する。PRINT 文や IF テストなどを含む、すべての SQL 文は <code>@@error</code> をリセットするため、実行の成否を判断する文の直後にステータス・チェックを行うこと。 |
| <code>@@identity</code> | Adaptive Server Enterprise では、INSERT、LOAD、または SELECT INTO 文によって IDENTITY カラムに挿入された最後の値。ローがテーブルに挿入されると、 <code>@@identity</code> は必ずリセットされる。文が複数のローを挿入した場合、 <code>@@identity</code> は、最後に挿入されたローの IDENTITY 値を反映する。関連するテーブルに IDENTITY カラムがない場合、 <code>@@identity</code> は 0 に設定される。INSERT または SELECT INTO 文が正常に実行されなかったり、失敗した文を含むトランザクションがロールバックされても、 <code>@@identity</code> の値には影響しない。 <code>@@identity</code> は、IDENTITY カラムに最後の値を挿入した文がコミットに失敗した場合でも、その値を保持する。 |
| <code>@@idle</code> | Adaptive Server Enterprise では、最後に Adaptive Server Enterprise が起動されて以来、アイドル状態になった時間数 (チック単位) を示す。Sybase IQ では 0 が返される。 |
| <code>@@io_busy</code> | Adaptive Server Enterprise では、最後にサーバが起動されて以来 Adaptive Server Enterprise が入出力処理に費やした時間数 (チック単位) を示す。Sybase IQ では 0 が返される。 |

| グローバル変数 | 戻り値 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| @@isolation | 接続の現在の独立性レベル。Adaptive Server Enterprise では、@@isolation にはアクティブ・レベルの値が入る。 |
| @@langid | Adaptive Server Enterprise では、現在使用中の言語のローカル言語 ID を定義する。Sybase IQ では 0 が返される。 |
| @@language | Adaptive Server Enterprise では、現在使用中の言語の名前を定義する。Sybase IQ では “English” が返される。 |
| @@maxcharlen | Adaptive Server Enterprise では、デフォルト文字セット内の 1 文字の最大長 (バイト単位)。Sybase IQ では 1 が返される。 |
| @@max_connections | ネットワーク・サーバの場合、アクティブ・クライアントの最大数 (各クライアントが複数の接続をサポートできるため、データベース接続数ではない)。Adaptive Server Enterprise では、サーバへの最大接続数。 |
| @@ncharsize | Adaptive Server Enterprise では、各国語 1 文字の平均の長さ (バイト単位)。Sybase IQ では 1 が返される。 |
| @@nestlevel | Adaptive Server Enterprise では、現在の実行のネスト・レベル (初期値 0)。ストアド・プロシージャまたはトリガが別のストアド・プロシージャまたはトリガを呼び出すたびに、ネスト・レベルが増加する。Sybase IQ では -1 が返される。 |
| @@pack_received | Adaptive Server Enterprise では、最後に起動して以来読み込んだ入力パケットの数。Sybase IQ では 0 が返される。 |
| @@pack_sent | Adaptive Server Enterprise では、最後に起動して以来書き込んだ出力パケットの数。Sybase IQ では 0 が返される。 |
| @@packet_errors | Adaptive Server Enterprise では、パケットの送受信中に発生したエラーの数。Sybase IQ では 0 が返される。 |
| @@procid | 現在実行中のプロシージャのストアド・プロシージャ ID。 |
| @@servername | ローカル Adaptive Server Enterprise または Sybase IQ サーバの名前。 |
| @@spid | Adaptive Server Enterprise では、現在のプロセスのサーバ・プロセス ID。Sybase IQ では、現在の接続の接続ハンドル。これは、sa_conn_info プロシージャによって表示されるものと同じ値。 |

| グローバル変数 | 戻り値 |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>@@sqlstatus</code> | 最後の FETCH 文によって得られたステータス情報を保持する。 <code>@@sqlstatus</code> に含まれる値は、次のとおり。 <ul style="list-style-type: none"> 0 - FETCH 文が正常に完了した 1 - FETCH 文でエラーが発生した 2 - 結果セットには値が残されていない |
| <code>@@thresh_hysteresis</code> | Adaptive Server Enterprise では、スレッシュホールドのアクティブ化に必要な空き領域の変更。Sybase IQ では 0 が返される。 |
| <code>@@timeticks</code> | Adaptive Server Enterprise では、1 チックあたりのマイクロ秒数。1 チックあたりの時間は、マシンによって異なる。Sybase IQ では 0 が返される。 |
| <code>@@total_errors</code> | Adaptive Server Enterprise では、読み込み中または書き込み中に発生したエラーの数。Sybase IQ では 0 が返される。 |
| <code>@@total_read</code> | Adaptive Server Enterprise では、最後に起動されてから行われたディスク読み込みの回数。Sybase IQ では 0 が返される。 |
| <code>@@total_write</code> | Adaptive Server Enterprise では、最後に起動されてから行われたディスク書き込みの回数。Sybase IQ では 0 が返される。 |
| <code>@@tranchained</code> | Transact-SQL プログラムの現在のトランザクション・モード。 <code>@@tranchained</code> は、非連鎖モードの場合は 0、連鎖モードの場合は 1 を返す。 |
| <code>@@trancount</code> | トランザクションのネスト・レベル。バッチ内の BEGIN TRANSACTION ごとに、トランザクション・カウントが増分する。 |
| <code>@@transtate</code> | Adaptive Server Enterprise では、文の実行後のトランザクションの現在の状態。Sybase IQ では -1 が返される。 |
| <code>@@version</code> | Adaptive Server Enterprise または Sybase IQ の現在のバージョン情報。 |

コメント

コメントは、SQL 文または文ブロックに説明テキストを付加するために使用します。データベース・サーバは、コメントを実行しません。

Sybase IQ では、いくつかのコメント・インジケータが使用可能です。

- **--(二重ハイフン)** データベース・サーバは、その行のハイフンに続く文字を無視します。これは、SQL92 のコメント・インジケータです。
- **//(二重スラッシュ)** 二重スラッシュは、二重ハイフンと同じ意味です。
- **/*...*/(スラッシューアスタリスク)** 2つのコメント・マーカの間にある文字は、すべて無視されます。2つのコメント・マーカは、同じ行にあっても別の行にあってもかまいません。このスタイルで示されたコメントはネストできます。このスタイルは、C スタイル・コメントとも呼ばれます。
- **% (パーセント記号)** パーセント記号は、二重ハイフンと同じ意味になります。% をコメント・インジケータとして使用しないようにおすすめます。

注意 --(二重ハイフン)と/*(スラッシューアスタリスク)のコメント・スタイルは、Adaptive Server Enterprise と互換性があります。

例

- 次に、二重ハイフンを使用したコメントの記述例を示します。

```
CREATE FUNCTION fullname (firstname CHAR(30),
                          lastname CHAR(30))
RETURNS CHAR(61)
-- fullname は firstname 引数と lastname 引数を連結し、
-- 2 つの引数の間にスペースを 1 つ挿入します。
BEGIN
    DECLARE name CHAR(61);
    SET name = firstname || ' ' || lastname;
    RETURN ( name );
END
```

- 次に、C スタイルを使ったコメントの記述例を示します。

```
/*
    営業部で働く従業員の名前と
    従業員 ID をリストします。
*/
CREATE VIEW SalesEmployee AS
SELECT emp_id, emp_lname, emp_fname
FROM "GROUPO".Employees
WHERE DepartmentID = 200
```

NULL 値

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 未知または適用不可の値を示します。 |
| 構文 | NULL |
| 使用法 | すべての場所 |
| パーミッション | データベースに接続しておく必要があります。 |
| 関連する動作 | なし |
| 説明 | NULL 値は、あらゆるデータ型の有効な値とは異なる特別な値です。ただし、NULL 値はすべてのデータ型で使用できます。NULL 値は、認識できないまたは適用不可の情報を表すために使用します。NULL 値が使用されるこれら 2 つのケースは、それぞれが個別で性質も異なることに注意してください。 |

| 状況 | 説明 |
|------|-----------------------|
| 未知の値 | フィールドには値があるが、未知の値である |
| 適用不可 | フィールドは、この特定のローに適用できない |

SQL では、NOT NULL 制限を使用してカラムを作成できます。このカラムには NULL 値を挿入できません。

SQL の 3 値的論理の概念は、NULL 値によって表されます。任意の比較演算子を使って、NULL 値を含む任意の値と NULL 値を比較すると UNKNOWN になります。TRUE が返る唯一の検索条件は、IS NULL 述部です。SQL では、WHERE 句の検索条件が TRUE と評価された場合のみ、ローが選択されます。UNKNOWN または FALSE と評価されたローは、選択されません。

IS [NOT] *truth-value* 句は、NULL 値があるローを選択するために使用します (*truth-value* は TRUE、FALSE、または UNKNOWN のいずれかです)。この句の詳細については、「[検索条件](#) (34 ページ) を参照してください。

次の例では、カラム Salary が NULL 値を持っています。

| 条件 | 真理値 | 選択 |
|----------------------|---------|----|
| Salary = NULL | UNKNOWN | なし |
| Salary <> NULL | UNKNOWN | なし |
| NOT (Salary = NULL) | UNKNOWN | なし |
| NOT (Salary <> NULL) | UNKNOWN | なし |
| Salary = 1000 | UNKNOWN | なし |
| Salary IS NULL | TRUE | あり |

| 条件 | 真理値 | 選択 |
|--------------------------|-------|----|
| Salary IS NOT NULL | FALSE | なし |
| Salary = 1000 IS UNKNOWN | TRUE | あり |

2 つの異なるテーブルのカラムを比較する場合、同じ規則が適用されます。そのため、2 つのテーブルを結合すると、比較したカラムに NULL 値があるローは選択されません。

数値式で使用する場合も、NULL 値は特別な性質を持っています。NULL 値が含まれる数値式の結果は、すべて NULL 値になります。NULL 値を数値に加算しても結果は NULL 値であり、数値にはなりません。NULL 値を 0 として扱う場合は、ISNULL(*expression*, 0) 関数を使用します。詳細については、「[第 4 章 SQL 関数](#)」を参照してください。

SQL クエリの作成で生じるエラーの多くは、NULL の性質によるものです。注意して、このような問題を避けるようにしてください。検索条件の組み合わせによる 3 値的論理の影響については、「[検索条件](#)」(34 ページ)を参照してください。

例

次の INSERT 文は、Borrowed_book テーブルの date_returned カラムに NULL を挿入します。

```
INSERT
INTO Borrowed_book
( date_borrowed, date_returned, book )
VALUES ( CURRENT DATE, NULL, '1234' )
```

この章について

この章では、Sybase IQ がサポートするデータ型について説明します。

内容

| トピック名 | ページ |
|----------------------------------------|-----|
| 文字データ型 | 69 |
| 数値データ型 | 73 |
| バイナリ・データ型 | 77 |
| Bit データ型 | 83 |
| 日付と時刻のデータ型 | 83 |
| 日付と時刻のデータベースへの送信 | 85 |
| データベースから日付と時刻を取得する | 86 |
| 日付と時刻の比較 | 86 |
| あいまいさのない日付と時刻の使用 | 87 |
| ドメイン | 89 |
| データ型変換 | 90 |
| Sybase IQ BINARY ロード形式 | 95 |

文字データ型

説明

文字、数字、記号などの文字列を格納します。

構文

```

CHAR [ ( max-length ) ]
CHARACTER [ ( max-length ) ]
CHARACTER VARYING [ ( max-length ) ]
VARCHAR [ ( max-length ) ]
UNIQUEIDENTIFIERSTR

```

使用法

CHAR 最大長 *max-length* バイトの文字データです。*max-length* が指定されなかった場合のデフォルトは 1 で、指定可能な最大サイズは 32KB - 1 です。255 バイトより大きい CHAR データに対する制限については「注意」をご覧ください。

データベース内の文字データ表現と、長い文字列の記憶領域に関する次の注意を参照してください。

CHAR 値では、BLANK PADDING オプションが指定されているかどうかにかかわらず、*max-length* までブランクが埋め込まれます。マルチバイト文字の文字列が CHAR 型として格納される場合でも、最大長は文字数ではなくバイト数になります。

CHARACTER CHAR と同じです。

CHARACTER VARYING VARCHAR と同じです。

LONG VARCHAR 任意の長さの文字データ。最大サイズは、データベース・ファイルの最大サイズ (現在は 2 ギガバイト) により制限されます。

TEXT これは、ユーザ定義データ型です。NULL を許可する LONG VARCHAR として実装されます。

VARCHAR 文字列の記憶領域にブランクが追加されないことを除き、CHAR と同じです。VARCHAR 文字列に指定可能な最大長は (32KB - 1) です。255 バイトより大きい VARCHAR データに対する制限については「注意」をご覧ください。

UNIQUEIDENTIFIERSTR CHAR(36) として実装されるドメインです。このデータ型は、Microsoft SQL Server の uniqueidentifier 列をマッピングするとき、リモート・データへのアクセスに使用されます。

注意

別途ライセンスが必要ですが、Sybase IQ オプションがサポートするキャラクタ・ラージ・オブジェクト (CLOB: Character Large Object) データを使用すると、IQ ページのサイズが 128KB であれば 0 ~ 512TB (テラバイト)、また IQ ページが 512KB であれば 0 ~ 2PB (ペタバイト) のサイズを扱うことができます。最大長は、4GB にデータベース・ページ・サイズを掛けた値になります。詳細については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。

記憶領域サイズ

表 3-1 は、文字データの記憶領域サイズを示します。

表 3-1 : 文字データの記憶領域サイズ

| データ型 | カラム定義 | 入力データ | 記憶領域 |
|---------------------------|-----------------|---------------|---------------|
| CHARACTER、CHAR | 幅 (32K - 1) バイト | (32K - 1) バイト | (32K - 1) バイト |
| VARCHAR、CHARACTER VARYING | 幅 (32K - 1) バイト | (32K - 1) バイト | (32K - 1) バイト |

文字セットとコード・ページ

アプリケーションから渡されるバイナリ表現とまったく同じバイナリ表現を使用して、データベースの中に文字データを配置します。通常、文字データは、システムで使用されている文字セットのバイナリ表現を使用してデータベースの中に格納されます。文字セットに関する説明は、オペレーティング・システムに添付のマニュアルに記載されています。

Windows では、コード・ページの最初の 128 文字は共通です。コード・ページの上位の半分にある特殊文字 (アクセントが付いた国際文字) を使用する場合は、データベースについて注意する必要があります。具体的には、別のコード・ページを使用している異なる種類のマシンにデータベースをコピーする場合に注意が必要です。このとき、特殊文字が元のコード・ページ表現を使用するデータベースから取得されますが、表示には新しいコード・ページが使われるため、これらの文字は画面上に異なる文字で表示されます。

2 つのクライアントが、異なるコード・ページを実行する同じマルチユーザ・サーバを使用している場合にも、このような問題が発生します。一方のクライアントから挿入または更新されたデータが、もう一方のクライアントでは正しく表示されない場合があります。

この問題は、データベースが複数のプラットフォームで使用される場合にも発生します。PowerBuilder などの多くの Windows アプリケーションは、標準 ANSI 文字セットでデータベースにデータを挿入します。Windows 以外のアプリケーションがこのデータを使用しようとしても、正しく表示しなかったり、拡張文字を正しく更新しなかったりします。

この問題は非常に複雑です。アプリケーションの上位の半分にあるコード・ページの拡張文字を使用する場合は、データベースを使用するすべてのクライアントとすべてのマシンで、同じまたは互換性のあるコード・ページを使用するようにしてください。

インデックス

DATE、TIME、DTTM を除くすべてのインデックス・タイプを、データ長が 255 バイト以下の CHAR データと VARCHAR データで使用できます。

VARCHAR データと後続ブランク

INSERT、UPDATE、または LOAD TABLE を使用して挿入したデータは、次のいずれかの形式になります。

- 引用符で囲む
- 引用符で囲まない
- Binary

データ型が VARCHAR のカラムの場合、挿入されるデータ内の後続ブランクは次のように処理されます。

- 引用符で囲まれたデータの場合、後続ブランクは削除されません。

- 引用符で囲まれていないデータの場合：
 - 後続ブランクは、挿入時と更新時に常に削除されます。
 - LOAD 文の場合は、STRIP RTRIM/OFF LOAD オプションを使用して、後続ブランクを削除するかどうかを指定できます。STRIP RTRIM/OFF オプションは、可変長の非バイナリ・データにのみ適用されます。たとえば、次のようなスキーマを想定できます。

```
CREATE TABLE t( c1 VARCHAR(3) );
LOAD TABLE t( c1 ',' ) ..... STRIP RTRIM // 後続ブランクは削除される

LOAD TABLE t( c1 ',' ) ..... STRIP OFF // 後続ブランクは削除されない

LOAD TABLE t( c1 ASCII(3) ) ... STRIP RTRIM // 後続ブランクは削除されない
LOAD TABLE t( c1 ASCII(3) ) ... STRIP OFF // 後続ブランクは削除される

LOAD TABLE t( c1 BINARY ) ..... STRIP RTRIM // 後続ブランクは削除される
LOAD TABLE t( c1 BINARY ) ..... STRIP OFF // 後続ブランクは削除される
```

- バイナリ・データの場合、後続ブランクは常に削除されます。

アプリケーションを作成する際に、VARCHAR カラムにおける後続ブランクの存在に依存しないでください。アプリケーションが後続ブランクに依存している場合は、VARCHAR カラムではなく CHAR カラムを使用してください。

255 バイトを超える CHAR データ と VARCHAR データに関する制限

255 バイトを超える CHAR カラムと VARCHAR カラムでは、WD と CMP のデフォルトのインデックス・タイプのみがサポートされます。これらのカラムに LF、HG、HNG、DATE、TIME、または DTTM インデックスを作成することはできません。

互換性

- CHAR の代用としての CHARACTER (n) は、Adaptive Server Enterprise ではサポートされていません。
- Sybase IQ は Adaptive Server Enterprise によって提供される NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR データ型をサポートしません。Sybase IQ は CHAR と VARCHAR のデータ型に格納される Unicode をサポートします。
- Sybase IQ は、SQL Anywhere よりも長い LONG VARCHAR データ型をサポートします。詳細については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。
- Sybase IQ と Adaptive Server Enterprise の互換性を保つため、文字データ型には必ず長さを指定してください。

長い文字列

SQL Anywhere は CHAR、VARCHAR、および LONG VARCHAR のカラムをすべて、同じ型として処理します。254 文字までの値は、短い文字列として保存され、先頭に長さのバイトが付加されます。255 バイトより長い値はすべて、長い文字列と見なされます。255 番目より後ろの文字は、長い文字列値を含むローとは個別に格納されます。

255 番目より後ろの文字をすべて無視するいくつかの関数が存在します(「SQL 関数」を参照)。それらは、**soundex**、**similar**、およびすべての日付関数です。また、長い文字列を数字に変換する算術演算はすべて、最初の 255 文字にのみ適用されます。こうした制限が問題になるのはきわめてまれです。

他のすべての関数および演算子は、長い文字列の全長を処理します。

数値データ型

説明

数値データを格納します。

構文

```
[ UNSIGNED ] BIGINT
[ UNSIGNED ] { INT | INTEGER }
SMALLINT
TINYINT
DECIMAL [ ( precision [ , scale ] ) ]
NUMERIC [ ( precision [ , scale ] ) ]
DOUBLE
FLOAT [ ( precision ) ]
REAL
```

使用法

BIGINT 記憶領域を 8 バイト必要とする符号付き 64 ビット整数です。

整数を **UNSIGNED** として指定できます。デフォルトでは、このデータ型は符号付きです。整数の範囲は、-9223372036854775808 から 9223372036854775807 の間 (符号付き) または 0 から 18446744073709551615 の間 (符号なし) です。

INT または **INTEGER** 記憶領域を 4 バイト必要とする、-2147483648 から 2147483647 の間の値を取る符号付き 32 ビット整数です。

INTEGER データ型は真数値データ型です。精度は算術演算後に保存されます。

整数を **UNSIGNED** として指定できます。デフォルトでは、このデータ型は符号付きです。符号なし整数値の範囲は、0 から 4294967295 の間です。

SMALLINT 記憶領域を 2 バイト必要とする、-32768 から 32767 の間の値を取る符号付き 16 ビット整数です。

SMALLINT データ型は真数値データ型です。精度は算術演算後に保存されます。

TINYINT 記憶領域を 1 バイト必要とする、0 から 255 の間の値を取る符号なし 8 ビット整数です。

TINYINT データ型は真数値データ型です。精度は算術演算後に保存されます。

DECIMAL 総桁数の *precision* と小数点以下の桁数の *scale* を持つ符号付き 10 進数です。精度は 1 から 126 の値を取ることができ、位取りは 0 から精度の値までを取ることができます。デフォルトでは、位取りが 38 で精度が 126 です。結果はカラムの実際のデータ型に基づき、精度を保って計算されますが、アプリケーションに返される結果に対して位取りの最大値を設定することもできます。詳細については、『[「MAX_CLIENT_NUMERIC_SCALE オプション」 \(419 ページ\)](#)』と『リファレンス：文とオプション』の「[SET OPTION 文](#)」を参照してください。

表 3-2 は、10 進数に必要な記憶領域を示します。

表 3-2 : 10 進数に必要な記憶領域

| 精度 | 記憶領域 |
|----------|-------|
| 1 ~ 4 | 2 バイト |
| 5 ~ 9 | 4 バイト |
| 10 ~ 18 | 8 バイト |
| 19 ~ 126 | 以下を参照 |

精度の値が 18 より大きい 10 進数の値に必要とされる記憶領域は、次の式を使用して計算できます。

$$4 + 2 * (\text{int}(((\text{prec} - \text{scale}) + 3) / 4) + 1) + \text{int}((\text{scale} + 3) / 4) + 1$$

ここで *int* は、引数の整数部分です。カラムが使用する記憶領域は、そのカラムの精度と位取りによって決まります。カラム内の各セルは、その精度と位取りの最大値を記憶するのに十分な領域を持っています。例を示します。

```
NUMERIC(18,4) takes 8 bytes per cell
NUMERIC(19,4) takes 16 bytes per cell
```

DECIMAL データ型は真数値データ型です。精度は算術演算後に、最小の有効桁数まで保存されます。その絶対値の最大値は、 $[precision - scale]$ で定義される桁数の 9 で構成される数値、次に小数点、そして最後に $scale$ で定義された桁数の 9 で構成される数値が続きます。ゼロを除く最小の絶対値は、小数点の後に、 $[scale - 1]$ の数のゼロが続き、そして最後に 1 が 1 つある値です。例を示します。

```
NUMERIC (3,2) Max positive = 9.99 Min non-zero = 0.01
Max negative = -9.99
```

NULL から **NUMERIC** の明示的な変換に、精度も位取りも指定されない場合は、デフォルトで **NUMERIC(1,0)** が設定されます。例を示します。

```
SELECT CAST( NULL AS NUMERIC ) A,
       CAST( NULL AS NUMERIC(15,2) ) B
```

上の式は次のように記述されます。

```
A NUMERIC(1,0)
B NUMERIC(15,2)
```

NUMERIC **DECIMAL** と同じです。

DOUBLE 8 バイトで格納される符号付きの倍精度浮動小数点数です。ゼロでない絶対値の範囲は、 $2.2250738585072014e-308$ から $1.797693134862315708e+308$ までです。**DOUBLE** として保持される値は、厳密には 15 有効桁数ですが、15 桁を超えると丸め誤差が出ます。

DOUBLE データ型は概数値データ型です。算術演算後に丸め誤差が出ます。

FLOAT *precision* を指定しない場合、**FLOAT** データ型は **REAL** データ型と同じです。*precision* を指定した場合、精度の値に応じて、**FLOAT** データ型は **REAL** データ型または **DOUBLE** データ型と同じになります。**REAL** データ型と **DOUBLE** データ型のどちらと同じになるかは、プラットフォームによって異なります。この境界は、プラットフォームの単精度浮動小数点数の少数部で使用されているビット数です。

FLOAT データ型を使用してカラムを作成した場合、すべてのプラットフォーム上でカラムが少なくとも指定の最小精度の値を持つことが保証されます。対照的に、**REAL** および **DOUBLE** ではプラットフォーム非依存の最小精度は保証されません。

FLOAT データ型は概数値データ型です。算術演算後に丸め誤差が出ます。

REAL 4 バイトで格納される符号付きの単精度浮動小数点数。ゼロでない絶対値の範囲は、1.175494351e-38 から 3.402823466e+38 までです。REAL として保持される値は、厳密には 6 有効桁数ですが、6 桁を超えると丸め誤差が出ます。

REAL データ型は概数値データ型です。算術演算後に丸め誤差が出ます。

注意

- INTEGER、NUMERIC、DECIMAL のデータ型が真数値データ型とも呼ばれるのに対し、FLOAT、DOUBLE、REAL は概算値データ型と呼ばれます。算術演算後に、指定した最少有効桁数までの正確性が保証されるのは、真数値データだけです。
- TINYINT カラムは、CHAR または UNSIGNED CHAR と定義された Embedded SQL 変数にフェッチしないでください。これは、カラムの値を文字列に変換し、最初のバイトをプログラムの変数に割り当てようとしてしまうためです。
- CMP と HNG のインデックス・タイプは、FLOAT、DOUBLE、REAL のデータ型をサポートしません。また、HG インデックス・タイプは推奨されません。
- WD、DATE、TIME、DTTM のインデックス・タイプは数値データ型をサポートしません。

インデックス

互換性

- Embedded SQL の TINYINT カラムは、2 バイトまたは 4 バイト整数カラムにフェッチしてください。また、TINYINT 値をデータベースに送信する場合、C 変数は整数でなければなりません。
- Adaptive Server Enterprise のバージョン 12.5.x は、符号なし整数をサポートしません。Sybase IQ の符号なし整数は、Adaptive Server Enterprise の符号付き整数または数値データにマップしてください。そうすれば、データが自動的に変換されます。
 - IQ UNSIGNED SMALLINT データは ASE INT にマップします。
 - 負の値がある場合は、IQ UNSIGNED BIGINT から ASE NUMERIC (*precision, scale*) にマップします。

UNSIGNED BIGINT カラムで、データベース間のジョインを行う際にパフォーマンスが低下するのを避けるため、Sybase IQ の側で (符号付き) BIGINT にキャストすることをおすすめします。
- NUMERIC と DECIMAL のデータ型は、精度と位取りのデフォルト設定が製品によって異なるため、デフォルトで使用するのは避けてください。以下にその違いを示します。

| データベース | デフォルトの精度 | デフォルトの位取り |
|----------------------------|----------|-----------|
| Sybase IQ | 126 | 38 |
| Adaptive Server Enterprise | 18 | 0 |
| SQL Anywhere | 30 | 6 |

- FLOAT (p) のデータ型は、REAL または DOUBLE (p によって決まります) と同じです。Adaptive Server Enterprise では、 p が 15 以下の場合には REAL が使用され、 p が 15 よりも大きい場合は DOUBLE が使用されます。Sybase IQ の場合、カットオフ値はプラットフォームによって異なりますが、すべてのプラットフォームでカットオフ値は 22 より大きい値になります。
- Sybase IQ では、MONEY と SMALLMONEY の 2 つのユーザ定義データ型を使用できます。これらはそれぞれ、NUMERIC(19,4) および NUMERIC(10,4) として実装されます。これらは主に Adaptive Server Enterprise との互換性を保つために用意されています。

バイナリ・データ型

説明 ピクチャなどのロー・バイナリ・データを、16 進数に似た表記で格納します。最大長は (32K - 1) バイトです。UNIQUEIDENTIFIER データ型は、UUID (GUID とも呼ばれます) 値の記憶領域として使用されます。

構文 **BINARY** [(*length*)]
VARBINARY [(*max-length*)]
UNIQUEIDENTIFIER

使用法 バイナリ・データは文字 “0x” または “0X” で始まり、数字と大文字および小文字の A ~ F との任意の組み合わせで構成されます。カラム長はバイト数を指定することも、デフォルト長 (1 バイト) を使用することもできます。各バイトは、2 桁の 16 進数を保管します。デフォルト長は 1 バイトですが、**BINARY** カラムと **VARBINARY** カラムの長さに、常に文字数を偶数で指定することをおすすめします。指定したカラム長より長い値が入力されると、Sybase IQ は入力値を指定された長さまでトランケートします。その際、警告やエラー・メッセージは表示されません。

BINARY 長さが *length* バイトのバイナリ・データです。 *length* が指定されない場合は、デフォルトで 1 バイトが設定されます。指定可能な最大サイズは 255 バイトです。すべての入力値の長さがほぼ等しいと予測されるデータには、固定長バイナリ型の **BINARY** を使用します。 **BINARY** カラムへのエントリは、 *length* のカラム長まで 0 が埋め込まれるので、 **VARBINARY** カラムへのエントリよりも多くの記憶領域が必要になる場合があります。

VARBINARY 最大長 *max-length* バイトのバイナリ データです。 *max-length* が指定されない場合は、デフォルトで 1 バイトが設定されます。指定可能な最大サイズは (32K - 1) バイトです。長さが大きく変化することが予測されるデータには、可変長バイナリ型の **VARBINARY** を使用してください。

注意

別途ライセンスが必要ですが、 Sybase IQ オプションがサポートするバイナリ・ラージ・オブジェクト (BLOB: Binary Large Object) データを使用すると、 IQ ページのサイズが 128KB であれば 0 ~ 512TB (テラバイト)、また IQ ページが 512KB であれば 0 ~ 2PB (ペタバイト) の長さを持つことができます。最大長は、 4GB にデータベース・ページ・サイズを掛けた値になります。詳細については、『 Sybase IQ によるラージ・オブジェクト管理 』を参照してください。

LONG BINARY および **IMAGE** データ型の詳細については、「 [バイナリ・データ型](#) 」 (610 ページ) を参照してください。

後続ゼロの扱い

BINARY カラムはすべて、カラム幅いっぱいまでゼロが埋め込まれます。すべての **VARBINARY** カラムで、後続ゼロがトランケートされます。

次の例では、 **BINARY** データ型と **VARBINARY** データ型に、それぞれ **NULL** と **NOT NULL** を定義した 4 種類のカラムがすべて揃っているテーブルを作成します。4 つのカラムすべてに同じデータを挿入しますが、ゼロが埋め込まれるか、トランケートされるかどうかは、カラムのデータ型によって異なります。

```
CREATE TABLE zeros (bnot BINARY(5) NOT NULL,  
                    bnull BINARY(5) NULL,  
                    vbnot VARBINARY(5) NOT NULL,  
                    vbnull VARBINARY(5) NULL);  
INSERT zeros VALUES (0x12345000, 0x12345000,  
                     0x12345000, 0x12345000);  
INSERT zeros VALUES (0x123, 0x123, 0x123, 0x123);  
INSERT zeros VALUES (0x0, 0x0, 0x0, 0x0);  
INSERT zeros VALUES ('002710000000ae1b',  
                     '002710000000ae1b', '002710000000ae1b',  
                     '002710000000ae1b');  
SELECT * FROM zeros;
```


| bnot | bnull | vbnot | vbnnull |
|--------------|--------------|--------------|----------------|
| 0x1234500000 | 0x1234500000 | 0x12345000 | 0x12345000 |
| 0x0123000000 | 0x0123000000 | 0x0123 | 0x0123 |
| 0x0000000000 | 0x0000000000 | 0x00 | 0x00 |
| 0x3030323731 | 0x3030323731 | 0x3030323731 | 0x3030323731 |

記憶領域の各バイトによってそれぞれ 2 桁の 16 進数が保持されるため、Sybase IQ は文字列 “0x” の後に偶数個の数字が続くバイナリ・エントリを期待します。“0x” の後に奇数個の数字が続く場合、Sybase IQ は先行ゼロが省略されているものとみなし、0 を追加します。

“0x00” と “0x0” の入力値は、可変長バイナリ・カラム (VARBINARY) に “0x00” として格納されます。固定長バイナリ・カラム (BINARY) では、値はフィールド幅いっぱいまでゼロが埋め込まれます。

```
INSERT zeros VALUES (0x0, 0x0, 0x0, 0x0);
SELECT * FROM zeros
```

| bnot | bnull | vbnot | vbnnull |
|--------------|--------------|--------------|----------------|
| 0x0000000000 | 0x0000000000 | 0x00 | 0x00 |

入力値に “0x” がない場合、Sybase IQ はその値が ASCII 値であると想定して、値を変換します。例を示します。

```
CREATE TABLE sample (col_bin BINARY(8));
INSERT sample VALUES ('002710000000aeb');
SELECT * FROM sample;
```

col_bin

0x3030323731303030

注意 上記の例で、string_truncation オプションが “off” に設定されていることを確認してください。

BINARY 値を選択する場合は、ゼロを埋め込んで値を指定するか、CAST 関数を使用する必要があります。例を示します。

```
SELECT * FROM zeros WHERE bnot = 0x0123000000;
```

または

```
SELECT * FROM zeros WHERE bnot = CAST(0x0123 as
binary(5));
```

フラット・ファイルからの ASCII データのロード

フラット・ファイルからバイナリ型のカラム (BINARY または VARBINARY) にロードされた ASCII データは、必ずニプルで格納されます。たとえば、フラット・ファイルからバイナリ・カラムに 0x1234 または 1234 が読み込まれた場合、Sybase IQ では 16 進数の 1234 として値が格納され、先頭の “0x” は無視されます。入力データに 0 ~ 9、a ~ f、A ~ F 以外の文字が含まれる場合、そのデータは拒否されます。

記憶領域サイズ

表 3-3 のリストは、バイナリ・データの記憶領域サイズを示します。

表 3-3 : バイナリ・データの記憶領域サイズ

| データ型 | カラム定義 | 入力データ | 記憶領域 |
|-----------|-----------------|----------------------|---------------|
| VARBINARY | 幅 (32K - 1) バイト | (32K - 1) バイトのバイナリ | (32K - 1) バイト |
| VARBINARY | 幅 (32K - 1) バイト | (64K - 2) バイトの ASCII | (32K - 1) バイト |
| BINARY | 幅 255 バイト | 255 バイトのバイナリ | 255 バイト |
| BINARY | 幅 255 バイト | 510 バイトの ASCII | 255 バイト |

プラットフォームの依存 特定の値の正確な入力形式は、使用しているプラットフォームに応じて異なります。このため、バイナリ・データを使う計算では、マシンによって異なった結果となることがあります。

16 進数の文字列と整数との間の変換で、どのプラットフォームでも同じ結果が得られるようにするには、プラットフォーム固有の CONVERT 関数ではなく、INTTOHEX 関数と HEXTOINT 関数を使用します。詳細については、「データ型変換関数」(110 ページ) を参照してください。

文字列演算子

文字列を連結する演算子である || と + は、どちらもバイナリ・データ型をサポートします。|| 演算子を使用する場合、バイナリ・オペランドから文字データ型への明示的な変換は必要ありません。ただし、明示的なデータ変換と暗示的なデータ変換とでは、結果に違いが生じます。

BINARY データと VARBINARY データに対する制限

BINARY データと VARBINARY データを格納するカラムには、以下の制限が適用されます。

- 集合関数 SUM、AVG、STDDEV、VARIANCE では、バイナリ・データ型を処理できません。集合関数 MIN、MAX、および COUNT は、バイナリ・データ型 (BINARY および VARBINARY) をサポートします。
- HNG、WD、DATE、TIME、および DTTM のインデックスは、BINARY または VARBINARY データをサポートしません。

- サイズが 255 バイトより大きい **VARBINARY** データの場合は、デフォルトと **CMP** のインデックス・タイプだけがサポートされます。
- ビット操作は、8 バイト以下の **BINARY** データと **VARBINARY** データでサポートされます。

互換性

バイナリ・データ型での後続ゼロの扱いは Sybase IQ、SQL Anywhere、Adaptive Server Enterprise でそれぞれ異なります。

表 3-4：後続ゼロの扱い

| データ型 | Sybase IQ | SQL Anywhere | ASE |
|--------------------|-----------------|-----------------|-----------------|
| BINARY NOT NULL | 埋め込みあり | 埋め込みなし | 埋め込みあり |
| BINARY NULL | 埋め込みあり | 埋め込みなし | 埋め込みなし |
| VARBINARY NOT NULL | トランケートする、埋め込みなし | トランケートする、埋め込みなし | トランケートする、埋め込みなし |
| VARBINARY NULL | トランケートする、埋め込みなし | トランケートする、埋め込みなし | トランケートする、埋め込みなし |

Adaptive Server Enterprise、SQL Anywhere、Sybase IQ はいずれも、**STRING_RTRUNCATION** データベース・オプションをサポートします。このオプションを使用すると、**INSERT** 文字列または **UPDATE** 文字列がトランケートされたとき表示されるエラー・メッセージに影響が出ます。Transact-SQL と互換性のある文字列比較を行う場合は、双方のデータベースで **STRING_RTRUNCATION** オプションを同じ値にセットしてください。

また、データをテーブルにロードする際に **STRING_RTRUNCATION** オプションを **ON** にすると、データが大きすぎてフィールドにロードできない場合に警告を表示させることができます。デフォルト値は **ON** です。

バイナリ型でのビット操作は Adaptive Server Enterprise ではサポートされていません。SQL Anywhere はバイナリ型データの先頭の 4 バイトに対してのみ、ビット操作をサポートします。Sybase IQ はバイナリ型データの先頭の 8 バイトに対して、ビット操作をサポートします。

UNIQUEIDENTIFIER UUID (GUID とも呼ばれます) 値の記憶領域で使用します。多くの場合、**UNIQUEIDENTIFIER** データ型は、プライマリ・キーやその他のユニーク・カラムで、ローを一意に識別する UUID (Universally Unique Identifier) 値を保持するために使用されます。**NEWID** 関数は UUID 値を、1 台のコンピュータで生成した UUID 値が別のコンピュータで生成した UUID 値と一致しないように生成します。したがって、**NEWID** を使用して生成された **UNIQUEIDENTIFIER** 値は、同期環境でキーとして使用できます。

たとえば、次の文では、テーブル `mytab` を更新し、カラム `uid_col` の値を、現在の値が `NULL` の場合は、`NEWID` 関数で生成された一意な識別子に設定します。

```
UPDATE mytab
  SET uid_col = NEWID()
  WHERE uid_col IS NULL
```

次の文を実行します。

```
SELECT NEWID()
```

一意な識別子は `BINARY(16)` として返されます。たとえば、値は `0xd3749fe09cf446e399913bc6434f1f08` になります。この文字列を `UUIDTOSTR()` 関数を使用して読みやすい形式に変換できます。

`UUID` は、`GUID (Globally Unique Identifier)` とも呼ばれます。

`STRTOUUID` 関数および `UUIDTOSTR` 関数を使用して、`UNIQUEIDENTIFIER` と文字列表記の間で値を変換できます。

`UNIQUEIDENTIFIER` 値は、`BINARY(16)` として格納され、返されます。

`UNIQUEIDENTIFIER` 値は大きいので、データベース間の一意な識別子が必要ない場合、`UNIQUEIDENTIFIER` の代わりに、`UNSIGNED BIGINT` または `UNSIGNED INT identity` カラムを使用した方がより効率的です。

`UNIQUEIDENTIFIER` の標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** `SQL Anywhere` でサポートされています。 `Adaptive Server Enterprise` ではサポートされていません。
- **下位互換性** `Sybase IQ 12.7` より前のバージョンで作成されたデータベースでは、`STRTOUUID`、`UUIDTOSTR`、および `NEWID` 関数が `CIS` の補正機能を通じてサポートされていました。バージョン `15.1` 以降では、`STRTOUUID`、`UUIDTOSTR`、および `NEWID` 関数は `Sybase IQ` のネイティブな関数です。

参照

`UNIQUEIDENTIFIER` の詳細については、次も参照してください。

- [「NEWID 関数 \[その他\]」 \(208 ページ\)](#)
- [「UUIDTOSTR 関数 \[文字列\]」 \(330 ページ\)](#)
- [「STRTOUUID 関数 \[文字列\]」 \(264 ページ\)](#)

Bit データ型

説明 ブール値を格納します。

| データ型 | 値 | サポート |
|------|---------|--------------------------|
| BIT | 0 または 1 | Sybase IQ および Enterprise |

使用法 BIT には、0 または 1 の値のみ格納されます。BIT カラムにゼロでない値を挿入すると、1 が格納されます。BIT カラムに値ゼロを挿入すると、0 が格納されます。

BIT データでは、デフォルトのインデックス・タイプだけがサポートされます。

互換性

Adaptive Server Enterprise の BIT データ型では、0 または 1 の値のみを扱うことができます。

日付と時刻のデータ型

説明 日付と時刻を格納します。

構文

DATE
DATETIME
SMALLDATETIME
TIME
TIMESTAMP

使用法 **DATE** 年、月、日などの暦日です。年の範囲は 0001 ~ 9999 です。日がゼロの値を持つことはできません。したがって、最小値は 0001-01-01 になります。DATE 値には、4 バイトの記憶領域が必要になります。

DATETIME **TIMESTAMP** で実装されるドメインです。DATETIME は主に Adaptive Server Enterprise との互換性を保つために用意されています。例外については、「[文字列から日付/時刻への変換の互換性](#)」(91 ページ)を参照してください。

SMALLDATETIME **TIMESTAMP** で実装されるドメインです。SMALLDATETIME は主に Adaptive Server Enterprise との互換性を保つために用意されています。例外については、「[文字列から日付/時刻への変換の互換性](#)」(91 ページ)を参照してください。

TIME 時、分、秒、秒以下で構成される時間です。秒の小数点以下は 6 桁まで格納されます。TIME 値には 8 バイトの記憶領域が必要です。(ODBC 規格では、TIME データ型の精度を秒の単位までに制限しています。このため、WHERE 句の比較に、秒の単位より高い精度に依存する TIME データ型を使用しないでください)。

TIMESTAMP 年、月、日、時、分、秒、秒以下で構成される時刻です。秒の小数点以下は 6 桁まで格納されます。日にはゼロでない値を格納してください。TIMESTAMP 値には 8 バイトの記憶領域が必要です。

TIMESTAMP データ型の有効範囲は 0001-01-01 00:00:00.000000 から 9999-12-31 23:59:59.999999 までです。1600-02-28 23:59:59 から 7911-01-01 00:00:00 の範囲外の TIMESTAMP データの表示は不完全になりますが、正確な日時はデータベースに格納されています。最初にそのデータを文字列に変換することにより、正確な値を表示することができます。これには、CAST() 関数を、以下の例のように使用します。この例では、最初に DATETIME カラムと TIMESTAMP カラムを持つテーブルを作成し、日付が 7911-01-01 より大きい値を挿入します。

```
create table mydates (id int, descript char(20),
    datetime_null datetime, timestamp_null timestamp);
insert into mydates values (1, 'example', '7911-12-30
    23:59:59', '7911-12-30 06:03:44');
commit;
```

CAST を使用せずに select を実行すると、時と分が 00:00: にセットされます。

```
select * from mydates;

1, 'example', '7911-12-30 00:00:59.000', '7911-12-30
00:00:44.000'
```

キャストを使用して select を実行すると、正しいタイムスタンプが表示されます。

```
select id, descript, cast(datetime_null as char(21)),
    cast(timestamp_null as char(21)) from mydates;

1, 'example', '7911-12-30 23:59:59.0', '7911-12-30
06:03:44.0'
```

注意

日付と時刻のデータでは、以下のインデックス・タイプがサポートされます。

- 日付および時刻のすべてのデータ型が CMP、HG、HNG、および LF のインデックス・タイプをサポートし、WD インデックス・タイプをサポートしません。

- DATE データは DATE インデックスをサポートします。
- TIME データは TIME インデックスをサポートします。
- DATETIME データと TIMESTAMP データは DTTM インデックスをサポートします。

日付と時刻のデータベースへの送信

説明

次のいずれかの方法で、日付と時刻をデータベースに送信します。

- インタフェースを使用して文字列として
- ODBC を使用して TIMESTAMP 構造体として
- Embedded SQL を使用して SQLDATETIME 構造体として

時刻をデータベースに文字列 (TIME データ型の場合) として、または文字列の一部 (TIMESTAMP データ型または DATE データ型の場合) として送信する場合、時間、分、秒をコロンで区切った *hh:mm:ss.sss* のフォーマットにする必要はありますが、文字列中のどこに置いてもかまいません。オプションとして、*hh:mm:ss.sss* のようにピリオドで秒と秒の小数点以下を区分することができます。以下は、有効かつ明確に時刻を指定するための文字列です。

```
21:35 -- am や pm で指定されていない場合は 24 時間表示です。  
10:00pm -- pm と指定されているため、12 時間表示と解釈します。  
10:00 -- pm がない場合は 10:00am を示します。  
10:23:32.234 -- 秒とそれ以下が含まれます。
```

データベースに日付を文字列として送信する場合、日付の変換は自動的に行われます。次の2つの方法のいずれかで、文字列を送信します。

- データベースに確実に解釈される *yyyy/mm/dd* または *yyyy-mm-dd* のいずれかの文字列として
- `DATE_ORDER` データベース・オプションに従って解釈される文字列として

日付フォーマットの文字列に、マルチバイト文字を格納することはできません。日付、時刻、日時のフォーマットの文字列に格納できるのは、シングルバイト文字だけです。これはデータベースの照合順序が、932JPN のようにマルチバイトの照合順序である場合も同じです。

データベースから日付と時刻を取得する

説明

次のいずれかの方法で、日付と時刻をデータベースから取得します。

- インタフェースを使用して文字列として
- ODBC を使用して `TIMESTAMP` 構造体として
- Embedded SQL を使用して `SQLDATETIME` 構造体として

日付または時刻を文字列として検索する場合は、データベース・オプション `DATE_FORMAT`、`TIME_FORMAT`、および `TIMESTAMP_FORMAT` によって指定されているフォーマットで検索します。これらのオプションの詳細については、『リファレンス：文とオプション』の「[SET OPTION 文](#)」を参照してください。

日付と時間を扱うための関数の詳細については、「[日付と時刻のデータ型](#)」(83 ページ)を参照してください。日付については次の演算子が許可されています。

- **timestamp + integer** 指定された値の日数を日付またはタイムスタンプに加えます。
- **timestamp - integer** 指定された値の日数を日付またはタイムスタンプから引きます。
- **date - date** 2つの日付(タイムスタンプ)間の日数を計算します。
- **date + time** 与えられた日付と時刻を結合するタイムスタンプを作成します。

日付と時刻の比較

説明

日付を文字列として比較する場合は、`DATEFORMAT` 関数または `CAST` 関数を使って日付を文字列に変換してから比較してください。例を示します。

```
DATEFORMAT(invoice_date, 'yyyy/mm/dd') = '1992/05/23'
```

`DATEFORMAT` の文字列式には、許可されている日付フォーマットをすべて使用できます。

日付フォーマットの文字列に、マルチバイト文字を格納することはできません。日付、時刻、日時フォーマットの文字列に格納できるのは、シングルバイト文字だけです。これはデータベースの照合順序が、932JPN のようにマルチバイトの照合順序である場合も同じです。

次の '?' がマルチバイト文字を表す場合、このクエリはエラーになります。

```
SELECT DATEFORMAT ( StartDate, 'yy?') FROM Employees;
```

代わりに、連結演算子を使用して、マルチバイト文字を日付フォーマットの文字列の外に移動します。

```
SELECT DATEFORMAT (StartDate, 'yy') + '?' FROM  
Employees;
```

あいまいさのない日付と時刻の使用

説明

yyyy/mm/dd または yyyy-mm-dd フォーマットの日付は、DATE_ORDER の設定に関係なく、日付として常に認識されます。区切り文字として他の文字 (疑問符、スペース文字、カンマなど) を使用することもできます。各ユーザの DATE_ORDER 設定が一定でないと思われる場合には、必ずこのフォーマットを使用してください。たとえば、ストアド・プロシージャでは、あいまいさのない日付フォーマットを使用することにより、ユーザの DATE_ORDER 設定に従った日付が間違っ て解釈されるのを防ぐことができます。

また、hh:mm:ss.sss のフォーマットの文字列は、あいまいさのない時刻として解釈されます。

日付と時刻を組み合わせる場合は、あいまいさのない日付と時刻を組み合わせることにより、結果的にあいまいさのない日付/時刻として評価されます。次の形式も、あいまいさのない日付/時刻値です。

```
YYYY-MM-DD HH.MM.SS.SSSSSS
```

ピリオドは、日付と組み合わせた時刻に対してのみ使用できます。

そのほかの場合にも、非常に柔軟な日付フォーマットを使用できます。Sybase IQ では、さまざまな文字列をフォーマットとして解釈することが可能です。この解釈は、DATE_ORDER データベース・オプションの設定に左右されます。DATE_ORDER データベース・オプションには、'MDY'、'YMD'、'DMY' の値を指定することができます。詳細については、『リファレンス：文とオプション』の「[SET OPTION 文](#)」を参照してください。たとえば、DATE_ORDER オプションを 'DMY' に設定するには、次のようにします。

```
SET OPTION DATE_ORDER = 'DMY' ;
```

デフォルトの `DATE_ORDER` 設定は、'YMD' です。ODBC ドライバは、接続が確立されたときに必ず `DATE_ORDER` オプションを 'YMD' に設定します。値を変更するには、`SET OPTION` 文を使用します。

データベース・オプション `DATE_ORDER` は、文字列 10/11/12 が、データベースによって 1912 年 10 月 11 日、1910 年 11 月 12 日、または 1912 年 11 月 10 日のどれに解釈されるかを決定します。日付文字列の年、月、日を何らかの文字 ('/'、'-', スペースなど) で区切ると、`DATE_ORDER` オプションで指定されている順序で表示されます。

年は、2 桁か 4 桁で入力できます。`NEAREST_CENTURY` オプションの値によって、2 桁の年の解釈が変わります。`NEAREST_CENTURY` よりも小さい値には 2000 が追加され、それ以外の値には 1900 が追加されます。このオプションのデフォルト値は 50 です。したがって、デフォルトでは 50 が 1950 と解釈され、49 が 2049 と解釈されます。詳細については、『リファレンス：文とオプション』の「[NEAREST_CENTURY オプション \[TSQL\]](#)」を参照してください。

月は月の名前または数字です。時間と分はコロンで区切りますが、文字列のどこにでも置けます。

Sybase では、年には常に 4 桁のフォーマットを使用して指定することを推奨しています。

`DATE_ORDER` の適切な設定を使用した次の文字列は、すべて有効な日付です。

```
99-05-23 21:35
99/5/23
1999/05/23
May 23 1999
23-May-1999
Tuesday May 23, 1999 10:00pm
```

文字列に部分的な日付指定だけがある場合、デフォルト値を使って日付が満たされます。次のデフォルトを使います。

年 1900

月 デフォルトなし

日 1 (月のフィールドにおいて便利です。たとえば、1999 年 5 月は、日付 1999-05-01 00:00 となります)

時、分、秒、秒以下 0

ドメイン

説明

ドメインは、必要に応じて精度と位取りを含めた組み込みデータ型のエイリアスです。

ドメインはユーザ定義データ型とも呼ばれ、データベース全体を通して、カラムを同じ NULL または NOT NULL 条件を持つ同じデータ型に自動的に定義できます。これにより、データベース全体の一貫性が高まります。ドメイン名の大文字と小文字は区別されません。大文字と小文字の表記だけが異なる既存のドメインと同じ名前のドメインを作成しようとする、エラーが返されます。

単純なドメイン

ドメインは `CREATE DOMAIN` 文を使用して作成します。詳細については、『リファレンス：文とオプション』の「[CREATE DOMAIN 文](#)」を参照してください。

次の文は `street_address` という名前の、35 文字の文字列のデータ型を作成します。

```
CREATE DOMAIN street_address CHAR( 35 )
```

`CREATE DOMAIN` の代わりに `CREATE DATATYPE` を使用することもできますが、`CREATE DOMAIN` は草案の SQL/3 規格で使用されている構文なので、こちらを使用することをおすすめします。

データ型を作成するには `RESOURCE` 権限が必要です。データ型を作成すると、`CREATE DOMAIN` 文を実行したユーザ ID がそのデータ型の所有者となります。このデータ型はすべてのユーザが使用できます。他のデータベース・オブジェクトとは異なり、所有者名をデータ型のプレフィックスとして使用しません。

カラムを定義する場合は、`street_address` データ型を他のデータ型とまったく同じように使用します。たとえば、2 つのカラムがある次のテーブルでは、2 番目のカラムが `street_address` カラムです。

```
CREATE TABLE twocol (id INT,  
street street_address)
```

所有者または DBA は、`COMMIT` を発行してから `DROP DOMAIN` 文を使用して、ドメインを削除できます。

```
DROP DOMAIN street_address
```

データベース内のどのテーブルもデータ型を使用していない場合にのみ、この文を実行できます。

ユーザ定義データ型を持つ制約およびデフォルト

NULL 値の許可、DEFAULT 値の設定など、カラムに関連する属性の多くは、ユーザ定義データ型に組み込むことができます。データ型で自動的に定義されるカラムはすべて、NULL 設定、CHECK 条件、および DEFAULT 値を継承します。これにより、データベース全体にわたって同様の意味を持つ一貫したカラムを作成できます。

たとえば、サンプル・データベース内の多くのプライマリ・キー・カラムは ID 番号を持つ整数カラムです。次の文は、こうしたカラムで役立つデータ型を作成します。

```
CREATE DOMAIN id INT
NOT NULL
DEFAULT AUTOINCREMENT
CHECK( @col > 0 )
```

データ型 ID を使用して作成されたカラムはすべて、NULL を持つことができず、デフォルトで自動インクリメント値になります。また、正の数を持つ必要があります。@col 変数内で、col の代わりに任意の ID を使用できます。

カラムに対して明示的に属性を指定することにより、必要に応じてデータ型の属性を上書きできます。あるデータ型 ID で作成したカラムで明示的に NULL 値を許可した場合、その ID データ型の設定にかかわらず、NULL が許可されます。

互換性

- **名前を付けた制約とデフォルト** Sybase IQ では、ユーザ定義データ型はベース・データ型、およびオプションで NULL または NOT NULL 条件とともに作成されます。名前付き制約と名前付きデフォルトはサポートしません。
- **データ型の作成** Sybase IQ では、sp_addtype システム・プロシージャを使用してドメインを追加するか、CREATE DOMAIN 文を使用します。Adaptive Server Enterprise の場合は、sp_addtype を使用する必要があります。

データ型変換

説明

型変換は自動的に発生することもあれば、CAST または CONVERT 関数を使用した明示的な型変換が必要になることもあります。

文字列が数値式で使用されているか、または数値引数の想定される関数への引数として使用されている場合、文字列は使用前に数値に変換されます。

数値が文字列式で使用されているか、または文字列関数の引数として使用されている場合、数値は使用前に文字列に変換されます。

すべての日付定数は文字列として指定されます。文字列は、自動的に日付に変換されてから使用されます。

自動的なデータ型変換が適切でない場合があります。

```
'12/31/90' + 5 -- 文字列を数値に変換します。
'a' > 0       -- 「a」を数値に変換します。
```

型変換を強制的に実行するには、CAST または CONVERT 関数を使用します。

以下の関数は、型変換を強制的に行うために使用することができます。

- **DATE(expression)** — 式を日付に変換し、時間、分、または秒を削除します。変換エラーがあればレポートします。
- **DATETIME(expression)** — 式をタイムスタンプに変換します。変換エラーがあればレポートします。
- **STRING(expression)** — CAST(value AS CHAR) と似ていますが、string(NULL) が空の文字列 (") であるのに対し、CAST(NULL AS CHAR) は NULL 値である点が異なります。

CAST 関数と CONVERT 関数については、「[データ型変換関数](#)」(110 ページ)を参照してください。

文字列から日付/時刻 への変換の互換性

文字列を日付データ型と時刻データ型に変更する場合、Sybase IQ と Adaptive Server Enterprise 間の動作にいくつかの相違があります。

時刻値だけ (日付なし) の文字列を日時/時刻データ型に変換する場合、Sybase IQ と Adaptive Server Enterprise は 1900 年 1 月 1 日のデフォルト日付を使用しますが、SQL Anywhere は現在の日付を使用します。

時刻のミリ秒の部分が 3 桁より小さい場合、Adaptive Server Enterprise は、前にピリオドが付くか、コロンが付くかによって違う値の解釈を行います。コロンの場合、値は 1000 分の 1 秒を意味します。ピリオドの場合は、1 桁であれば 10 分の 1、2 桁であれば 100 分の 1、3 桁であれば 1000 分の 1 秒を意味します。Sybase IQ と SQL Anywhere は、区切り文字に関係なく、値を同じ方法で解釈します。

例

- Adaptive Server Enterprise は、以下のように値を変換します。

```
12:34:56.7 to 12:34:56.700
12.34.56.78 to 12:34:56.780
12:34:56.789 to 12:34:56.789
12:34:56:7 to 12:34:56.007
12.34.56:78 to 12:34:56.078
12:34:56:789 to 12:34:56.789
```

- Sybase IQ は、どちらの場合も、Adaptive Server Enterprise がピリオドが前についた値を変換するのと同じ方法で、ミリ秒の値を変換します。

```
12:34:56.7 to 12:34:56.700
12.34.56.78 to 12:34:56.780
12:34:56.789 to 12:34:56.789
12:34:56:7 to 12:34:56.700
12.34.56:78 to 12:34:56.780
12:34:56:789 to 12:34:56.789
```

エクスポートされた日付の互換性

月の9日目まで、および時の10未満の値に対して、Adaptive Server Enterprise は1桁目のブランクをサポートし、Sybase IQ はゼロまたはブランクをサポートします。このようなデータを Adaptive Server Enterprise から Sybase IQ にロードする方法の詳細については、『システム管理ガイド 第1巻』の「[第7章 データベースへのデータの入出力](#)」を参照してください。

BIT から BINARY へのデータ型の変換

Sybase IQ では、BIT から BINARY へ、または BIT から VARBINARY への暗黙的な変換と明示的な変換をサポートします。これらの変換では、Adaptive Server Enterprise でのこれらの変換のサポートに準拠していません。Sybase IQ では、比較演算子と算術演算子がある式と、INSERT 文と UPDATE 文の中で BIT から BINARY へ、および BIT から VARBINARY へのデータ型変換を暗黙的に実行します。

BIT から BINARY への変換では、ビット値 'b' がバイナリ文字列の最初のバイトにコピーされ、残りのバイトに 0 が入力されます。たとえば、BIT 値 1 は、 2^n のニブルを持つ BINARY(n) 文字列 0x0100...00 に変換されます。BIT 値 0 は BINARY 文字列 0x00...00 に変換されます。

BIT から VARBINARY への変換では、BIT 値 'b' が BINARY 文字列の最初のバイトにコピーされ、残るバイトは使用されません。つまり、1 バイトのみが使用されます。たとえば、BIT 値 1 は、2 個のニブルを持つ VARBINARY(n) 文字列 0x01 に変換されます。

BIT データ型から BINARY データ型、および BIT データ型から VARBINARY データ型への変換は、暗黙的な変換も明示的な変換も同じ結果になります。次の表に、BIT から BINARY および VARBINARY への変換の例を示します。

| BIT 値 '1' を下記に変換 | 結果 |
|------------------|--------------------|
| BINARY(3) | 0x010000 |
| VARBINARY(3) | 0x01 |
| BINARY(8) | 0x0100000000000000 |
| VARBINARY(8) | 0x01 |

BIT から BINARY、および BIT から VARBINARY への変換の例 この例では、BIT データ型から BINARY データ型、および BIT データ型から VARBINARY データ型への暗黙的および明示的な変換を示しています。

次のテーブルとデータがあるとします。

```
CREATE TABLE tbin(c1 BINARY(9))
CREATE TABLE tvarbin(c2 VARBINARY(9))
CREATE TABLE tbar(c2 BIT)

INSERT tbar VALUES(1)
INSERT tbar VALUES(0)
```

BIT から BINARY への暗黙的な変換：

```
INSERT tbin SELECT c2 FROM tbar

c1
---
0x010000000000000000    (18 nibbles)
0x000000000000000000    (18 nibbles)
```

BIT から VARBINARY への暗黙的な変換：

```
INSERT tvarbin SELECT c2 FROM tbar

c2
---
0x01
0x00
```

BIT から BINARY への明示的な変換：

```
INSERT tbin SELECT CONVERT (BINARY(9), c2) FROM tbar

c1
---
0x010000000000000000    (18 nibbles)
0x000000000000000000    (18 nibbles)
```

BIT から VARBINARY への明示的な変換：

```
INSERT tvarbin SELECT CONVERT (VARBINARY(9), c2) FROM
tbar

c2
---
0x01
0x00
```

BIT と
CHAR/VARCHAR と
の間で行われる
データ型変換

Sybase IQ では、比較演算、算術演算、INSERT 文および UPDATE 文で、BIT と CHAR との間、BIT と VARCHAR との間の暗黙のデータ型変換がサポートされています。

BIT から VARCHAR へ、CHAR から BIT へ、および VARCHAR から BIT への変換の例 これらの例は、BIT と CHAR との間、BIT と VARCHAR との間の暗黙的および明示的なデータ型変換を示しています。

次のテーブルとデータがあるとします。

```
CREATE TABLE tchar(c1 CHAR(9))
CREATE TABLE tvarchar(c2 VARCHAR(9))
CREATE TABLE tbar(c2 BIT)
CREATE TABLE tbit(c2 BIT)

INSERT tbar VALUES(1)
INSERT tbar VALUES(0)
```

BIT から VARCHAR へ/VARCHAR から BIT への暗黙的な変換と BIT から VARCHAR への暗黙的な変換：

```
INSERT tvarchar SELECT c2 FROM tbar
SELECT c2, char_length(c2) FROM tvarchar

c2,char_length(tvarchar.c2)
-----
'1',1
'0',1
```

VARCHAR から BIT への暗黙的な変換：

```
INSERT tbit SELECT c2 FROM tvarchar
SELECT c2 FROM tbit

c2
--
0
1
```

BIT から CHAR へ/CHAR から BIT への明示的な変換と BIT から CHAR への明示的な変換：

```
INSERT tchar SELECT CONVERT (CHAR(9), c2) FROM tbar
SELECT c1, char_length(c1) FROM tchar

c1,char_length(tchar.c1)
-----
'1',9
'0',9
```


CHAR から BIT への明示的な変換 :

```
INSERT tbit SELECT CONVERT (BIT, c1) FROM tchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```

BIT から VARCHAR へ/VARCHAR から BIT への明示的な変換と BIT から VARCHAR への明示的な変換 :

```
INSERT tvarchar SELECT CONVERT (VARCHAR(9), c2)
FROM tbar
SELECT c2, char_length(c2) FROM tvarchar
```

```
c2,char_length(tvarchar.c2)
-----
'1',1
'0',1
```

VARCHAR から BIT への明示的な変換 :

```
INSERT tbit SELECT CONVERT (BIT, c2) FROM tvarchar
SELECT c2 FROM tbit
```

```
c2
--
0
1
```

Sybase IQ BINARY ロード形式

説明

Sybase IQ は、FORMAT BINARY および BINARY カラム指定句を使用して、LOAD TABLE 文で読み取ることができるデータ・ファイルを作成します。

Sybase IQ へのデータのロードを高速化するには、Sybase IQ のバイナリ・フォーマットでデータ・ファイルを作成して、LOAD TABLE コマンドの FORMAT BINARY 構文によりこのデータを Sybase IQ にロードできます。

LOAD TABLE 構文を使用したロード・スクリプトの作成、およびロードの指定の手順については、『リファレンス：文とオプション』を参照してください。

これらのバイナリ・フォーマットでデータ・ファイルを作成して、対応するデータ型のカラムにロードします。ほとんどの場合、Sybase IQ はプラットフォームに固有のバイナリ・フォーマットを使用します。これらのデータ型は、Sybase IQ に固有のバイナリ・フォーマットを使用する例外です。

- DATE
- TIME
- DATETIME
- NUMERIC

IQ の BINARY ロード形式とロード効率

Sybase IQ の BINARY ロード形式は、固定幅のフォーマットです。

一般的に、固定幅のロードは可変幅のロードよりも高速です。ロード・ロジックがカラムとローの長さを認識している場合、データの処理効率が向上します。幅が一定でないカラムとローをデリミタを使用して分割すると、ロード中に入力データをスキャンしてデリミタを探す時間がかかるようになります。

IQ の BINARY ロード形式は、固定幅のロードです。ロードでは、テーブルの定義の情報から各カラムの幅と各ローの長さを確認できます。

注意 BINARY ロード形式はビッグ・エンディアンとリトル・エンディアンを区別します。これは、BINARY ロード形式がネイティブ・バイナリ・データ型を使用してデータを表現するためです。

オペレーティング・システムのネイティブ・データ型

次のデータ型のデータは、オペレーティング・システムのネイティブ・バイナリ・フォーマットで保存され、そのフォーマットで直接データ・ファイルに書き込むことができます。Sybase IQ は、変換することなく各バイト数を関連データ型に直接読み込みます。

- BIT (1 バイト)
- TINYINT (1 バイト)
- SMALLINT (2 バイト)
- INT/UNSIGNED INT (4 バイト)
- BIGINT/UNSIGNED BIGINT (8 バイト)
- FLOAT (4 バイト)

- DOUBLE (8 バイト)
- CHAR/VARCHAR (文字データ)
- BINARY/VARBINARY (バイナリ・データ)

デフォルトでは、VARCHAR カラムと VARBINARY カラムが、LOAD TABLE *column-spec* により指定された値までのバイト数で読み取られます。

DATE

DATE カラム・データは、0000-01-01 からの日数を表す 4 バイト (32 ビットの符号なし整数) として Sybase IQ に格納されます。暦日を Sybase IQ のバイナリ・フォーマットに変換するには、以下を使用します。

ある年、月、日の場合：

```
year = current_year - 1;
days_in_year_0000 = 366;
binaryDateValue = (year * 365)
+ (year / 4)
- (year / 100)
+ (year / 400)
days_in_year_0000
day_of_current_year
-1;
```

上の式で *day_of_current_year* 値には、たとえば 2 月 12 日の場合、第 43 日と指定します。

TIME

TIME データは、マイクロ秒数 (つまり、 $1.0e-6$ 秒) を表す 64 ビットの符号なし数量として格納されます。マイクロ秒数は、次のように計算します。

ある時間、分、秒、マイクロ秒 (usec) の場合：

```
binaryTimeValue = (hour * 3600 + minute * 60 + second +
microsecond ) * 1000000
```

TIMESTAMP

TIMESTAMP データは、マイクロ秒数を表す 64 ビットの符号なし整数として格納されます。バイナリ TIMESTAMP 値は、次のように計算します。

ある年、月、日、時間、分、秒、マイクロ秒の場合：

前述の日付の *binaryDateValue* を計算します。

前述の時間の *binaryTimeValue* を計算します。

```
binaryDateTimeValue = binaryDateValue *
864000000000 + binaryTimeValue
```

NUMERIC と
DECIMAL

NUMERIC と DECIMAL のデータ型は、精度の関数であるために、フォーマットが変化します。値の右側には、値のフル・スケールまで 0 を埋め込む必要があります。さらに、左側にも 0 を埋め込む必要がありますが、埋め込みはバイナリ・プログラミングにより自動的に行われます。値に 0 が埋め込まれると、小数点が削除されます。たとえば、値 12.34 は次のようになります。

- NUMERIC(4,2): 1234
- NUMERIC(6,4): 123400
- NUMERIC(8,4): 00123400
- NUMERIC(12,6): 000012340000
- NUMERIC(16,8): 0000001234000000

値に 0 を埋め込み、小数点を削除すると、次の規則が当てはまります。

- 精度が 4 以下の場合、バイナリ・フォーマットは、2 バイトの整数量に対するオペレーティング・システムのネイティブ・バイナリ・フォーマットと同じです。
- 精度が 5 から 9 までの場合、バイナリ・フォーマットは、4 バイトの整数量に対するオペレーティング・システムのネイティブ・バイナリ・フォーマットと同じです。
- 精度が 10 から 18 までの場合、バイナリ・フォーマットは、8 バイトの整数量に対するオペレーティング・システムのネイティブ・バイナリ・フォーマットと同じです。
- 精度が 19 以上の場合、次の C++ 構造定義を使用する特別なフォーマットが存在します。

```
struct {
    unsigned char sign; // 正の符号は 1、負の符号は 0
    unsigned char ndig; // 桁数
    unsigned char exp; // 指数
    unsigned short digits[80];
};
```

値がゼロでない限り、指数は excess-80 形式です。「0」値は次のように表現されます。

```
sign = 1
ndig = 0
exp = 0
```

最大指数値は 159 です。サポートされる最大桁数は 288 です。“digits[0]” には、最下位桁が含まれます。数字は、「符号なし短整数型」(2 バイト) 数量あたり 2 桁のパックされた表現に格納されま
す。ある数の場合：

```
lower order digit = digit[i] & 0x00FF
```

```
high order digit => digit[i] & 0xFF00
```

たとえば、**NUMERIC(20)** カラムにロードされる値 **100** を考えてみます。この値のバイナリ・レイアウトは次のようになります。

```
0x0101 0x5000 0x0064 0x0000 0x0000 .....
```

```
Sign - 0x01
```

```
Number digits - 0x01
```

```
Exponent - 0x50
```

```
Digits - 0x0064
```

別の例として、値 **32769** を考えてみます。

```
0x0102 0x5000 0x0ad1 0x0003 0x0000 0x0000 ....
```

```
Sign - 0x01
```

```
Number digits - 0x02
```

```
Exponent - 0x50
```

```
Digits - 0x0ad1 0x0003
```

これらの数を基数 **10** に変換すると、次のようになります。

```
0x0ad1 => 2769 0x0003 => 3
```

NULL の挿入

NULL 値を挿入する最も適切な方法は、入力ファイルに NULL BYTE を使用し、LOAD TABLE 文のカラム指定に WITH NULL BYTE を指定することです。このためには、入力ファイルの各データ・フィールドの終わりを 'x00' か 'x01' にします。入力ファイルの各データ・フィールドの終わりを 'x01' とすると、ロードでカラムに NULL が挿入されま
す。例を示します。

```
create table d1 ( c1 date );
```

```
load table d1 ( c1 binary with null byte ) from  
'filename' quotes off escapes off format binary;
```

ロード入力ファイルの内容が **000b32cb00000b32cc00** の場合、テーブルに 2 つのローがロードされます。最初のローは、2009 年 5 月 7 日、2 番目のローは 2009 年 5 月 8 日です。入力ファイルの各バイナリ日付の後に NULL BYTE が追加されています。最初のローに NULL をロードする場合は、入力ファイルの NULL BYTE の値を 'x01' に変更します。

```
000b32cb01000b32cc00
```

カラム指定の NULL の部分は、テーブルのカラムにデータをロードするときに、特定の入力値を NULL として処理する方法を指定します。NULL として処理される文字には、BLANKS、ZEROS、または定義したその他のリテラルのリストなどがあります。NULL 値を指定するか、またはソース・ファイルから NULL 値を読み込む場合は、ロード先のカラムに NULL を格納できる必要があります。

ZEROS は次のように解釈されます。

- 入力データがすべてバイナリのゼロ (文字のゼロではない) の場合、カラムが NULL に設定されます。
- 入力データが文字のゼロの場合は、次のようになります。
 - NULL(ZEROS) を指定しても、カラムに NULL が設定されることはない。
 - NULL('0') を指定すると、カラムに NULL が設定される。例を示します。

ロード :

```
CREATE TABLE t1 ( c1 INT, c2 INT );
```

ビッグ・エンディアン・バイト順を使用する入力データ・ファイルの表示 :

```
od -x data.inp
3030 3030 0000 04d2
```

実行 :

```
LOAD TABLE t1 ( c1 ASCII(4) NULL( '0000' ),
                c2 BINARY )
FROM 'data.inp'
FORMAT BINARY
QUOTES OFF
ESCAPES OFF;
```

結果 :

```
SELECT * FROM t1;
c1      c2
NULL    1234
```

- 入力データがバイナリのゼロ (全ビットがクリア) の場合は、次のようになります。
 - NULL(ZEROS) を指定すると、カラムに NULL が設定される。
 - NULL('0') を指定しても、カラムに NULL が設定されることはない。次に例を示します。

ロード :

```
CREATE TABLE t1 ( c1 INT, C2 INT );
```

ビッグ・エンディアン・バイト順を使用する入力データ・ファイルの表示 :

```
od -x data.inp
0000 0000 0000 04d2
```

実行 :

```
LOAD TABLE t1 ( c1 ASCII(4) NULL( zeros ),
                c2 BINARY )
FROM 'data.inp'
FORMAT BINARY
QUOTES OFF
ESCAPES OFF;
```

結果 :

```
SELECT * FROM T1;
c1      c2
NULL    1234
```

別の例として、LOAD TABLE 文に col1 date('yymmdd') null(zeros) が記述され、ロードするデータが 000000 である場合は、000000 を DATE(4) に変換できないことを示すエラーが表示されます。LOAD TABLE 文で、データが 000000 である場合に col1 に NULL 値が挿入されるようにするには、NULL 句を null('000000') のように記述するか、データをバイナリのゼロに修正して NULL(ZEROS) を使用する必要があります。

BINARY ロード形式で NULL をロードする別の方法として、ロード先カラムが NULL 値を受け入れる場合に、LOAD TABLE 文のカラムにデータを指定しない方法があります。例を示します。

```
CREATE TABLE t1 ( c1 INT, c2 INT );
LOAD TABLE T1 ( c2 BINARY ) FROM 'data.inp'
FORMAT BINARY
QUOTES OFF
ESCAPES OFF;
```

```
SELECT * FROM T1;
c1      c2
NULL    1234
Null    1234
```

ビッグ・エンディアン・バイト順を使用する入力データ・ファイルの表示 :

```
od -x data.inp
0000 04d2 0000 04d2
```


この章について

この章では、Sybase IQ がサポートする組み込み関数について説明します。

内容

| トピック名 | ページ |
|---------------------------------------|-----|
| 概要 | 104 |
| 集合関数 | 104 |
| 分析関数 | 106 |
| データ型変換関数 | 110 |
| 日付と時刻の関数 | 111 |
| HTTP 関数 | 115 |
| 数値関数 | 115 |
| 文字列関数 | 117 |
| システム関数 | 119 |
| SQL および Java のユーザ定義関数 | 123 |
| 時系列関数および予測関数 | 124 |
| その他の関数 | 127 |
| アルファベット順の関数リスト | 128 |

概要

関数は、データベースから情報を返します。関数は、式が使用できる場所であればどこでも使用できます。

Sybase IQ で関数を使用する場合は、次のことに注意してください。

- 特に記述がないかぎり、NULL 値をパラメータとして受け取る関数は、NULL 値を返します。
- FROM 句を省略した場合、またはクエリ内のすべてのテーブルが SYSTEM DB 領域にある場合、クエリは Sybase IQ ではなく SQL Anywhere によって処理されます。このため、特に構文およびセキュリティの制限やオプションの設定方法の違いによって、動作が変わる可能性があります。処理に適用されるルールについては SQL Anywhere のマニュアルを参照してください。
- FROM 句を必要としないクエリがある場合は、“FROM iq_dummy” 句を追加することによって、強制的に Sybase IQ で処理させることができます。この iq_dummy は、ユーザが自分のデータベースに作成する 1 ロー 1 カラムのテーブルです。

集合関数

機能

集合関数は、データベースに含まれるローのグループのデータを要約します。SELECT 文の GROUP BY 句を使ってグループを作成します。

使用法

SUM()、MIN()、MAX()、AVG()、COUNT() などの単純な集合関数を使用できるのは、select リスト、および SELECT 文の HAVING 句と ORDER BY 句だけです。これらの関数は、データベースに含まれるローのグループのデータを要約します。SELECT 文の GROUP BY 句を使ってグループを作成します。

「ウィンドウ関数」と呼ばれる集約関数の新しいクラスでは、「ダウ工業株 30 種平均の四半期の移動平均」や「各部署のすべての従業員とその累積給与をリストする」などのクエリに対する回答を算出する移動平均および累積方法を提供します。

- AVG()、COUNT()、MAX()、MIN()、SUM() などの単純な集合関数は、データベースに含まれるローのグループのデータを要約します。SELECT 文の GROUP BY 句を使ってグループを作成します。

- 1つの引数を取る新しい統計集合関数には、STDDEV()、STDDEV_SAMP()、STDDEV_POP()、VARIANCE()、VAR_SAMP()、VAR_POP() があります。

単純な集合関数と新しい集合関数はどちらも、SQL クエリの指定に <ウィンドウ句> (「ウィンドウ」) を組み込むウィンドウ関数として使用できます。これにより、処理時に結果セットに対して概念的に移動ウィンドウを作成することができます。詳細については、「[分析関数](#)」(106 ページ) を参照してください。

ウィンドウ集合関数のもう 1 つのクラスは、時系列データの分析をサポートします。単純な集合関数と統計集合関数と同様に、これらのウィンドウ集合関数は、SQL クエリの指定 (または *window-spec*) と組み合わせ使用できます。時系列ウィンドウ集合関数は、相関、直線回帰、ランク付け、加重平均の結果を計算します。

- 時系列分析用の ANSI SQL:2008 OLAP 関数には、CORR()、COVAR_POP()、COVAR_SAMP()、CUME_DIST()、FIRST_VALUE()、LAST_VALUE()、REGR_AVGX()、REGR_AVGY()、REGR_COUNT()、REGR_INTERCEPT()、REGR_R2()、REGR_SLOPE()、REGR_SXX()、REGR_SXY()、REGR_SYY() があります。
- データベース業界で使用される ANSI 以外の SQL:2008 OLAP 集合関数の拡張機能には、FIRST_VALUE()、MEDIAN()、LAST_VALUE() があります。
- 加重移動平均を計算する OLAP の加重集合関数には、EXP_WEIGHTED_AVG() および WEIGHTED_AVG() があります。

表 4-1 : 集合関数

| 集合関数 | パラメータ |
|------------------|--------------------------------------------------|
| AVG | ([DISTINCT] { column-name numeric-expr }) |
| CORR | (dependent-expression, independent-expression) |
| COUNT | (*) |
| COUNT | ([DISTINCT] { column-name numeric-expr }) |
| COVAR_POP | (dependent-expression, independent-expression) |
| COVAR_SAMP | (dependent-expression, independent-expression) |
| CUME_DIST | () |
| EXP_WEIGHTED_AVG | (expression, period-expression) |
| FIRST_VALUE | (expression) |
| LAST_VALUE | (expression) |
| MAX | ([DISTINCT] { column-name numeric-expr }) |
| MEDIAN | (expression) |

| 集合関数 | パラメータ |
|----------------|--------------------------------------------------|
| MIN | ([DISTINCT] { column-name numeric-expr }) |
| REGR_AVGX | (dependent-expression, independent-expression) |
| REGR_AVGY | (dependent-expression, independent-expression) |
| REGR_COUNT | (dependent-expression, independent-expression) |
| REGR_INTERCEPT | (dependent-expression, independent-expression) |
| REGR_R2 | (dependent-expression, independent-expression) |
| REGR_SLOPE | (dependent-expression, independent-expression) |
| REGR_SXX | (dependent-expression, independent-expression) |
| REGR_SXY | (dependent-expression, independent-expression) |
| REGR_SYY | (dependent-expression, independent-expression) |
| STDDEV | ([ALL] expression) |
| SUM | ([DISTINCT] { column-name numeric-expr }) |
| VARIANCE | ([ALL] expression) |
| WEIGHTED_AVG | (expression, period-expression) |

AVG、SUM、STDDEV、VARIANCE の集合関数は、バイナリ・データ型 (BINARY および VARBINARY) をサポートしません。

参照

それぞれの関数の詳細については、この章の個別の分析関数の説明を参照してください。

OLAP 関数の使用の詳細については、『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」を参照してください。

分析関数

機能

分析関数には、次のものがあります。

- 単純な集合 — AVG、COUNT、MAX、MIN、SUM、STDDEV、VARIANCE

注意 Grouping() 関数以外の単純な集合関数は OLAP ウィンドウ関数と併用できません。

- ウィンドウ関数
 - ウィンドウ集合 — AVG、COUNT、MAX、MIN、SUM
 - ランク付け関数 — RANK、DENSE_RANK、PERCENT_RANK、NTILE

- 統計関数 — STDDEV、STDDEV_SAMP、STDDEV_POP、VARIANCE、VAR_SAMP、VAR_POP
- 分散統計関数 — PERCENTILE_CONT および PERCENTILE_DISC
- 数値関数 — WIDTH_BUCKET、CEIL、LN、EXP、POWER、SQRT、FLOOR

ウィンドウ集合関数の 使用法

OLAP に関する ANSI SQL 拡張で導入された主な機能は、「ウィンドウ」という名前の構成体です。このウィンドウ拡張により、ユーザはクエリの結果セット (クエリの論理パーティション) をパーティションと呼ばれるローのグループに分割し、現在のローについて集約するローのサブセットを決定することができます。

1 つのウィンドウで 3 つのウィンドウ関数クラス (ランク付け関数、ロー・ナンバリング関数、ウィンドウ集合関数) を使用できます。

ウィンドウ拡張は、ウィンドウ名または指定に対するウィンドウ関数の種類を指定し、1 つのクエリ式のスコープ内のパーティション化された結果セットに適用されます。ウィンドウ・パーティションは、特殊な OVER 句の 1 つ以上のカラムでの定義に基づいてクエリから返されるローのサブセットです。

```
OVER (PARTITION BY col1, col2...)
```

ウィンドウ操作では、パーティション内の各ローのランク付け、パーティション内のローの値の分布、および類似の操作などの情報を設定できます。また、データの移動平均や合計を計算し、データおよびそのデータの操作に対する影響を評価する機能を拡張することもできます。

ウィンドウ・パーティションは、特殊な OVER() 句の 1 つ以上のカラムでの定義に基づいてクエリから返されるローのサブセットです。

```
OVER (PARTITION BY col1, col2...)
```

ランク付け関数の 使用法

アプリケーション開発者は、OLAP ランク付け関数を使用して、「今年度出荷された製品の中で売上合計が上位 10 の製品名」または「15 社以上から受注した営業部員の上位 5%」などの質問に回答するクエリを 1 つの SQL 文で作成することができます。これらの関数には、ランク付け関数 RANK()、DENSE_RANK()、PERCENT_RANK()、NTILE() があり、PARTITION BY 句と一緒に使用します。

ランク付け統計関数は、グループ内の項目をランク付けし、分布を計算して結果セットを複数のグループに分類します。ランク付け統計関数 (RANK、DENSE_RANK、PERCENT_RANK、NTILE) にはすべて OVER (ORDER BY) 句が必要です。次に例を示します。

```
RANK() OVER ( [PARTITION BY] ORDER BY <expression>
[ ASC | DESC ] )
```

ORDER BY 句は、ランク付けを実行するパラメータ、および各グループでローをソートする順序を指定します。この ORDER BY 句は、OVER 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。ランク付けクエリ内の集合関数に DISTINCT を指定することはできません。

OVER 句は、関数がクエリの結果セットを操作することを示します。結果セットとは、FROM、WHERE、GROUP BY、HAVING 句がすべて評価された後に返されるローを指します。OVER 句には、ランク付け統計関数の計算の対象となるローのデータ・セットを定義します。

expression にはソートを指定します。カラムの参照、集合関数、またはこれらの項目を起動する式など、有効な式を何でも指定できます。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

ランク付け統計関数ができるのは、SELECT 文、INSERT 文、または SELECT 文の ORDER BY 句にある select リストのみです。ランク付け関数は、ビューまたは union に含めることができます。ランク付け関数は、サブクエリ、HAVING 句、および UPDATE または DELETE 文の select リストでは使用できません。Sybase IQ 15.1 では 1 つのクエリで複数のランク付け統計関数を使用できます。

統計集合分析関数の 使用法

データベースに含まれるローのグループのデータを要約します。SELECT 文の GROUP BY 句を使ってグループを作成します。集合関数は、select リストと、SELECT 文の HAVING 句および ORDER BY 句の中だけで使用できます。これらの関数には、STDDEV、STDDEV_POP、STDDEV_SAMP、VARIANCE、VAR_POP、VAR_SAMP があります。

OLAP 関数を、処理時に結果セットに対して概念的に移動ウィンドウを作成する SQL クエリの指定に OVER() 句があるウィンドウ関数として使用できます。

分散統計関数の使用法

逆分散統計関数である PERCENTILE_CONT および PERCENTILE_DISC は、パーセンタイル値を関数の引数として受け取り、WITHIN GROUP 句で指定されたデータ・グループまたはデータ・セット全体に対して処理を実行します。これらの関数は、グループごとに 1 つの値を返します。PERCENTILE_DISC では、結果のデータ型は WITHIN GROUP 句に指定した ORDER BY の項目のデータ型と同じになります。PERCENTILE_CONT では、結果のデータ型は、numeric (WITHIN GROUP 句の ORDER BY 項目が numeric の場合) または double (ORDER BY 項目が整数または浮動小数点の場合) となります。

逆分散統計関数では、WITHIN GROUP (ORDER BY) 句を指定する必要があります。次に例を示します。

```
PERCENTILE_CONT ( expression1 ) WITHIN GROUP ( ORDER BY
expression2 [ASC | DESC ] )
```

expression1 の値には、numeric データ型の定数を、0 以上 1 以下の範囲で指定します。引数が NULL であれば、“wrong argument for percentile” エラーが返ります。引数の値が 0 よりも小さいか、1 よりも大きい場合は、“data value out of range” エラーが返ります。

必須の ORDER BY には、パーセンタイル関数の実行の対象となる式と、各グループ内でのローのソート順を指定します。この ORDER BY 句は、WITHIN GROUP 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。

WITHIN GROUP 句は、クエリ結果を順序付けられたデータ・セットに分類します。関数はこのデータ・セットに基づいて結果を計算します。

expression2 には、カラム参照を含む 1 つの式でソートを指定します。このソート式に、複数の式やランク付け統計関数、set 関数、またはサブクエリを指定することはできません。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

逆分散統計関数は、サブクエリ、HAVING 句、ビュー、union で使用することが可能です。逆分散統計関数は、分析を行わない単純な集合関数が使用されるところであれば、どこでも使用できます。逆分散統計関数は、データ・セット内の NULL 値を無視します。

表 4-2 は、統計関数とそのパラメータを示します。表 4-1 に示した集合関数とは異なり、ウィンドウ関数で DISTINCT を指定することはできません。

表 4-2 : 分析関数

| 関数 | パラメータ |
|-----------------|--------------------------------------|
| AVG | ({ column-name numeric-expr }) |
| COUNT | (*) |
| COUNT | ({ column-name expression }) |
| DENSE_RANK | () |
| GROUPING * | ({ GROUPING group-by-expression }) |
| MAX | ({ column-name expression }) |
| MIN | ({ column-name expression }) |
| NTILE | (integer) |
| PERCENT_RANK | () |
| PERCENTILE_CONT | (numeric-expr) |

| 関数 | パラメータ |
|-----------------|------------------------------------------------|
| PERCENTILE_DISC | (<i>numeric-expr</i>) |
| RANK | () |
| STDDEV | ([ALL] <i>expression</i>) |
| STDDEV_POP | ([ALL] <i>expression</i>) |
| STDDEV_SAMP | ([ALL] <i>expression</i>) |
| SUM | ({ <i>column-name</i> <i>expression</i> }) |
| VAR_POP | ([ALL] <i>expression</i>) |
| VAR_SAMP | ([ALL] <i>expression</i>) |
| VARIANCE | ([ALL] <i>expression</i>) |

*OLAP SQL 標準では、Grouping() は GROUP BY CUBE または GROUP BY ROLLUP オペレーションでのみ使用できます。

互換性

ランク付け統計関数と逆分散統計関数は、Adaptive Server Anywhere や Adaptive Server Enterprise ではサポートされません。

参照

それぞれの関数の詳細については、この章の個別の分析関数の説明を参照してください。

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」を参照してください。

データ型変換関数

機能

データ型変換関数は、引数があるデータ型から別のデータ型に変換します。

表 4-3 は、データ型変換関数とそのパラメータを示します。

表 4-3 : データ型変換関数

| データ型変換関数 | パラメータ |
|-------------|------------------------------------------------------------|
| BIGINTTOHEX | (<i>integer-expression</i>) |
| CAST | (<i>expression AS data type</i>) |
| CONVERT | (<i>data type, expression</i> [, <i>format-style</i>]) |
| HEXTOBIGINT | (<i>hexadecimal-string</i>) |
| HEXTOINT | (<i>hexadecimal-string</i>) |
| INTTOHEX | (<i>integer-expr</i>) |
| ISDATE | (<i>string</i>) |
| ISNUMERIC | (<i>string</i>) |

説明

式を日付フォーマットに従った日付、タイムスタンプ、または文字列にそれぞれ変換する DATE、DATETIME、DATEFORMAT、YMD 関数の詳細については、「日付と時刻の関数」(111 ページ)を参照してください。式を文字列に変換する STRING 関数の詳細については、「文字列関数」(117 ページ)の項を参照してください。

データベース・サーバは、多数のデータ型変換を自動的に行っています。たとえば、数値式が必要なところに文字列が与えられた場合、文字列は自動的に数値に変換されます。Sybase IQ による自動データ型変換の詳細については、「データ型変換」(90 ページ)を参照してください。

日付と時刻の関数

機能

日付関数と時刻関数は、日付および時刻データ型の変換、抽出、操作を行い、日付および時刻の情報を返します。

表 4-4 と 表 4-5 は、日付関数と時刻関数、およびそのパラメータを示します。

構文 1

表 4-4 : 日付と時刻の関数

| 日付と時刻の関数 | パラメータ |
|-------------|------------------------------------------------------------------------|
| DATE | (<i>expression</i>) |
| DATECEILING | (<i>date-part</i> , <i>datetime-expr</i> , [<i>multiple-expr</i>]) |
| DATEFLOOR | (<i>date-part</i> , <i>datetime-expr</i> , [<i>multiple-expr</i>]) |
| DATEFORMAT | (<i>datetime-expr</i> , <i>string-expr</i>) |
| DATENAME | (<i>date-part</i> , <i>date-expr</i>) |
| DATEROUND | (<i>date-part</i> , <i>datetime-expr</i> , [<i>multiple-expr</i>]) |
| DATETIME | (<i>expression</i>) |
| DAY | (<i>date-expr</i>) |
| DAYNAME | (<i>date-expr</i>) |
| DAYS | (<i>date-expr</i>) |
| DAYS | (<i>date-expr</i> , <i>date-expr</i>) |
| DAYS | (<i>date-expr</i> , <i>integer-expr</i>) |
| DOW | (<i>date-expr</i>) |
| HOUR | (<i>datetime-expr</i>) |
| HOURS | (<i>datetime-expr</i>) |
| HOURS | (<i>datetime-expr</i> , <i>datetime-expr</i>) |
| HOURS | (<i>datetime-expr</i> , <i>integer-expr</i>) |
| ISDATE | (<i>string</i>) |

| 日付と時刻の関数 | パラメータ |
|-----------|---------------------------------------------------------|
| MINUTE | (<i>datetime-expr</i>) |
| MINUTES | (<i>datetime-expr</i>) |
| MINUTES | (<i>datetime-expr</i> , <i>datetime-expr</i>) |
| MINUTES | (<i>datetime-expr</i> , <i>integer-expr</i>) |
| MONTH | (<i>date-expr</i>) |
| MONTHNAME | (<i>date-expr</i>) |
| MONTHS | (<i>date-expr</i>) |
| MONTHS | (<i>date-expr</i> , <i>date-expr</i>) |
| MONTHS | (<i>date-expr</i> , <i>integer-expr</i>) |
| NOW | (*) |
| QUARTER | (<i>date-expr</i>) |
| SECOND | (<i>datetime-expr</i>) |
| SECONDS | (<i>datetime-expr</i>) |
| SECONDS | (<i>datetime-expr</i> , <i>datetime-expr</i>) |
| SECONDS | (<i>datetime-expr</i> , <i>integer-expr</i>) |
| TODAY | (*) |
| WEEKS | (<i>date-expr</i>) |
| WEEKS | (<i>date-expr</i> , <i>date-expr</i>) |
| WEEKS | (<i>date-expr</i> , <i>integer-expr</i>) |
| YEAR | (<i>date-expr</i>) |
| YEARS | (<i>date-expr</i>) |
| YEARS | (<i>date-expr</i> , <i>date-expr</i>) |
| YEARS | (<i>date-expr</i> , <i>integer-expr</i>) |
| YMD | (<i>year-num</i> , <i>month-num</i> , <i>day-num</i>) |

構文 2

表 4-5 : Transact-SQL と互換性のある日付関数と時刻関数

| Transact-SQL と互換性のある日付関数と時刻関数 | パラメータ |
|-------------------------------|---------------------------------------------------------------------|
| DATEADD | (<i>date-part</i> , <i>numeric-expression</i> , <i>date-expr</i>) |
| DATEDIFF | (<i>date-part</i> , <i>date-expr1</i> , <i>date-expr2</i>) |
| DATENAME | (<i>date-part</i> , <i>date-expr</i>) |
| DATEPART | (<i>date-part</i> , <i>date-expr</i>) |
| GETDATE | () |

説明

Sybase IQ には、日付と時刻の関数の 2 つのクラスがあり、どちらを使用してもかまいませんが、これらのクラスはそれぞれスタイルが異なります。片方のセットは、Transact-SQL 互換です。

表 4-4 にリストされている日付と時刻の関数では、時間単位で操作できます。ほとんどの時間単位 (たとえば MONTH) には、時間を操作するために 4 つの関数がありますが、使用する名前は 2 つだけ (MONTH と MONTHS) です。

表 4-5 にリストされている関数は、Transact-SQL の日付と時刻の関数です。これらは、日付および時刻情報へのアクセスおよび操作の代替方法です。

日付関数の引数は、日付に変換してから使用する必要があります。たとえば、以下は誤りです。

```
days ( '1995-11-17', 2 )
```

正しくは以下のとおりです。

```
days ( date( '1995-11-17' ), 2 )
```

Sybase IQ で使用する定数やデータ型は、SQL Anywhere とは異なりますが、ユーザ・インタフェースは共通です。SELECT 文を FROM 句なしで発行すると、文は SQL Anywhere に渡されます。次の文は SQL Anywhere によって排他的に処理されます。

```
SELECT WEEKS('1998/11/01');
```

Sybase IQ が処理する次の文では、上記の WEEKS 関数とは異なる開始ポイントを使用し、異なる結果を返します。

```
SELECT WEEKS('1998/11/01') FROM iq_dummy;
```

別の例を考えてみます。MONTHS 関数は「適当な開始日」からの月数を返します。Sybase IQ の「適当な開始日」である仮想日付 0000-01-01 を選択して、最も効率よく日付を計算し、さまざまなデータ部分で一貫した日付を使用できます。SQL Anywhere には、単一の開始日はありません。次の 2 つの文は、最初の文が SQL Anywhere によって処理され、次の文が Sybase IQ によって処理され、両方とも 11 を返します。

```
SELECT MONTHS('0001/01/01');  
SELECT MONTHS('0001/01/01') FROM iq_dummy;
```

しかし、以下の文も考えてみてください。

```
SELECT DAYS('0001/01/01');  
SELECT DAYS('0001/01/01') FROM iq_dummy;
```

最初の文は SQL Anywhere によって処理され、値 307 を生成しますが、2 つ目の文は Sybase IQ によって処理され、366 が生成されます。

結果の一貫性を維持するために、必要かどうかにかかわらず FROM 句にテーブル名を含めてください。

注意 カラムとローを 1 つだけ含むダミー テーブルを作成します。日付または時刻関数を使用する FROM 文の SELECT 句でこのテーブルを参照すれば、必ず Sybase IQ で処理することができ、一貫性のある結果が得られます。

日付要素

日付関数の多くは、日付要素で構成される日付を使用します。表 4-6 は、*date-part* に指定可能な値を示します。

表 4-6 : 日付要素の値

| 日付要素 | 省略形 | 値 |
|---------------|-----|---------------------------------------------------------------------------------------------------------------------------------------------|
| Year | yy | 0001 – 9999 |
| Quarter | qq | 1 – 4 |
| Month | mm | 1 – 12 |
| Week | wk | 1 – 54 |
| Day | dd | 1 – 31 |
| Dayofyear | dy | 1 – 366 |
| Weekday | dw | 1 – 7 (日曜日 ~ 土曜日) |
| Hour | hh | 0 – 23 |
| Minute | mi | 0 – 59 |
| Second | ss | 0 – 59 |
| Millisecond | ms | 0 – 999 |
| Calyearofweek | cyr | 整数値。指定した週が始まる年。週の一部分が年初にあたる場合、新年の最初の曜日によっては、その週は前年の最終週の一部となります。新年が木曜日～土曜日に始まる場合、その最初の週は前年の最後の日曜日から開始されます。新年が日曜日～水曜日に始まる場合、どの曜日も前年の一部となりません。 |
| Calweekofyear | cwk | 年の何週目に指定日が含まれるかを表す 1 ~ 54 の整数。 |
| Caldayofweek | cdw | 曜日番号 (日曜日 = 1、土曜日 = 7)。 |

注意 デフォルトでは、1 週間の始まりは日曜日です。月曜日を週の始まりに指定するには、次のオプションを使用します。

```
set option 'Date_First_Day_of_Week' = '1'
```

何曜日を週の始まりに指定するかの詳細については、『リファレンス：文とオプション』の「[DATE_FIRST_DAY_OF_WEEK オプション](#)」を参照してください。

互換性

Adaptive Server Enterprise と互換性をとる場合、Transact-SQL 日付/時刻関数を使用します。

HTTP 関数

機能

HTTP 関数は、Web Service 内での HTTP 要求の処理を容易にします。

表 4-7 は、すべての HTTP 関数とそのパラメータを示します。

表 4-7 : HTTP 関数

| HTTP 関数 | パラメータ |
|--------------------|--------------------------------------------------------------------|
| HTML_DECODE | (<i>string</i>) |
| HTML_ENCODE | (<i>string</i>) |
| HTTP_DECODE | (<i>string</i>) |
| HTTP_ENCODE | (<i>string</i>) |
| HTTP_VARIABLE | (<i>var-name</i> [[, <i>instance</i>], <i>header-field</i>]) |
| NEXT_HTTP_HEADER | <i>header-name</i> |
| NEXT_HTTP_VARIABLE | <i>var-name</i> |

数値関数

機能

数値関数は、数値データ型に対して算術演算を行うか、数値情報を返します。

Sybase IQ で使用する定数やデータ型は、SQL Anywhere とは異なりませんが、ユーザ・インタフェースは共通です。SELECT 文を FROM 句なしで発行すると、文は SQL Anywhere に渡されます。結果の一貫性を維持するために、必要かどうかにかかわらず FROM 句にテーブル名を含めてください。

注意 このような場合は、ダミー・テーブルの使用について検討してください。

表 4-8 に、数値関数とそのパラメータを示します。

表 4-8 : 数値関数

| 数値関数 | パラメータ |
|-----------|-------------------------------------------------|
| ABS | (<i>numeric-expr</i>) |
| ACOS | (<i>numeric-expr</i>) |
| ASIN | (<i>numeric-expr</i>) |
| ATAN | (<i>numeric-expr</i>) |
| ATAN2 | (<i>numeric-expr1</i> , <i>numeric-expr2</i>) |
| CEIL | (<i>numeric-expr</i>) |
| CEILING | (<i>numeric-expr</i>) |
| COS | (<i>numeric-expr</i>) |
| COT | (<i>numeric-expr</i>) |
| DEGREES | (<i>numeric-expr</i>) |
| EXP | (<i>numeric-expr</i>) |
| FLOOR | (<i>numeric-expr</i>) |
| LN | (<i>numeric-expr</i>) |
| LOG | (<i>numeric-expr</i>) |
| LOG10 | (<i>numeric-expr</i>) |
| MOD | (<i>dividend</i> , <i>divisor</i>) |
| PI | (*) |
| POWER | (<i>numeric-expr1</i> , <i>numeric-expr2</i>) |
| RADIANS | (<i>numeric-expr</i>) |
| RAND | ([<i>integer-expr</i>]) |
| REMAINDER | (<i>numeric-expr</i> , <i>numeric-expr</i>) |
| ROUND | (<i>numeric-expr</i> , <i>integer-expr</i>) |
| SIGN | (<i>numeric-expr</i>) |
| SIN | (<i>numeric-expr</i>) |
| SQRT | (<i>numeric-expr</i>) |
| SQUARE | (<i>numeric-expr</i>) |

| 数値関数 | パラメータ |
|--------------|-------------------------------------------------------------------------------------|
| TAN | (<i>numeric-expr</i>) |
| “TRUNCATE” | (<i>numeric-expr</i> , <i>integer-expr</i>) |
| TRUNCNUM | (<i>numeric-expression</i> , <i>integer-expression</i>) |
| WIDTH_BUCKET | (<i>expression</i> , <i>min_value</i> , <i>max_value</i> , <i>num_buckets</i>) |

文字列関数

機能

文字列関数は、文字列の変換、抽出、操作を行い、また文字列に関する情報を返します。

マルチバイト文字セットを操作するときは、使用している関数が文字情報を返すのか、バイト情報を返すのかに注意してください。

ほとんどの文字列関数は、*string-expr* パラメータに指定されたバイナリ・データ (16 進文字) を処理できますが、LCASE、UCASE、LOWER、および LTRIM など一部の関数では、文字列式に文字列以外を指定できません。

VARCHAR 型の結果を返す関数 (SPACE や REPEAT など) に定数の LENGTH 引数を指定しない場合、デフォルトの長さが最大許容値になります。表 6-1 (346 ページ) の「フィールド・サイズ」カラムを参照してください。

これらの関数を 1 つ以上含む Sybase IQ クエリは、次のエラーのいずれかを返す場合があります。

```
ASA Error -1009080: Key doesn't fit on a single database
page: 65560 (4, 1)
```

```
ASA Error -1009119: Record size too large for database
page size
```

次に例を示します。

```
SELECT COUNT(*) FROM test1 a WHERE (a.col1 + SPACE(4-
LENGTH(a.col1)) + a.col2 + space(2- LENGTH(a.col2))) IN
(SELECT (b.col3) FROM test1 b);
```

このようなエラーを防ぐには、次の例のように適切な最大長を使用して関数結果をキャストします。

```
SELECT COUNT(*) FROM test1 a WHERE (a.col1 +
CAST(SPACE(4-LENGTH(a.col1)) AS VARCHAR(4)) + a.col2 +
CAST(SPACE(2-LENGTH(a.col2)) AS VARCHAR(4))) IN
(SELECT (b.col3) FROM test1 b);
```

64K の IQ ページ・サイズやマルチバイト照合ではエラーが発生しやすくなります。

表 4-9 に、文字列関数とそのパラメータを示します。

表 4-9 : 文字列関数

| 文字列関数 | パラメータ |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|
| ASCII | (<i>string-expr</i>) |
| BIT_LENGTH | (<i>column-name</i>) |
| BYTE_LENGTH | (<i>string-expr</i>) |
| CHAR | (<i>integer-expr</i>) |
| CHAR_LENGTH | (<i>string-expr</i>) |
| CHARINDEX | (<i>string-expr1</i> , <i>string-expr2</i>) |
| DIFFERENCE | (<i>string-expr1</i> , <i>string-expr2</i>) |
| GRAPHICAL_PLAN | (<i>string-expr</i>) |
| HTML_PLAN | (<i>string-expr</i>) |
| INSERTSTR | (<i>numeric-expr</i> , <i>string-expr1</i> , <i>string-expr2</i>) |
| LCASE | (<i>string-expr</i>) |
| LEFT | (<i>string-expr</i> , <i>numeric-expr</i>) |
| LEN | (<i>string-expr</i>) |
| LENGTH | (<i>string-expr</i>) |
| LOCATE | (<i>string-expr1</i> , <i>string-expr2</i> [, <i>numeric-expr</i>]) |
| LOWER | (<i>string-expr</i>) |
| LTRIM | (<i>string-expr</i>) |
| OCTET_LENGTH | (<i>column-name</i>) |
| PATINDEX | ('% <i>pattern</i> %', <i>string_expr</i>) |
| REPEAT | (<i>string-expr</i> , <i>numeric-expr</i>) |
| REPLACE | (<i>original-string</i> , <i>search-string</i> , <i>replace-string</i>) |
| REVERSE | (<i>expression</i> <i>uchar_expr</i>) |
| REPLICATE | (<i>string-expr</i> , <i>integer-expr</i>) |
| RIGHT | (<i>string-expr</i> , <i>numeric-expr</i>) |
| RTRIM | (<i>string-expr</i>) |
| SIMILAR | (<i>string-expr1</i> , <i>string-expr2</i>) |
| SORTKEY | (<i>string-expression</i> [, { <i>collation-id</i> <i>collation-name</i> [(<i>collation-tailoring-string</i>)] }]) |
| SOUNDEX | (<i>string-expr</i>) |
| SPACE | (<i>integer-expr</i>) |
| STR | (<i>numeric_expr</i> [, <i>length</i> [, <i>decimal</i>]]) |

| 文字列関数 | パラメータ |
|-------------|------------------------------------------------------------------------------|
| STR_REPLACE | (<i>string_expr1</i> , <i>string_expr2</i> , <i>string_expr3</i>) |
| STRING | (<i>string1</i> [, <i>string2</i> , ..., <i>string99</i>]) |
| STUFF | (<i>string_expr1</i> , <i>start</i> , <i>length</i> , <i>string_expr2</i>) |
| SUBSTRING | (<i>string_expr</i> , <i>integer_expr</i> [, <i>integer_expr</i>]) |
| TRIM | (<i>string_expr</i>) |
| UCASE | (<i>string_expr</i>) |
| UPPER | (<i>string_expr</i>) |

システム関数

機能

システム関数はシステム情報を返します。

表 4-10 は、システム関数とそのパラメータを示します。

表 4-10 : システム関数

| システム関数 | パラメータ |
|----------------------|-----------------------------------------------------------------------------------------------------------|
| COL_LENGTH | (<i>table-name</i> , <i>column-name</i>) |
| COL_NAME | (<i>table-id</i> , <i>column-id</i> [, <i>database-id</i>]) |
| CONNECTION_PROPERTY | (({ <i>property-id</i> <i>property-name</i> } [, <i>connection-id</i>])) |
| DATALLENGTH | (<i>expression</i>) |
| DB_ID | ([<i>database-name</i>]) |
| DB_NAME | ([<i>database-id</i>]) |
| DB_PROPERTY | (({ <i>property-id</i> <i>property-name</i> } [, { <i>database-id</i> <i>database-name</i> }])) |
| EVENT_CONDITION | (<i>condition-name</i>) |
| EVENT_CONDITION_NAME | (<i>integer</i>) |
| EVENT_PARAMETER | (<i>context-name</i>) |
| GROUP_MEMBER | (<i>group-name-string-expression</i> [, <i>user-name-string-expression</i>]) |
| INDEX_COL | (<i>table-name</i> , <i>index-id</i> , <i>key_#</i> [, <i>user-id</i>]) |
| NEXT_CONNECTION | ([<i>connection-id</i>] [, <i>database-id</i>]) |
| NEXT_DATABASE | (({ NULL <i>database-id</i> })) |
| OBJECT_ID | (<i>object-name</i>) |
| OBJECT_NAME | (<i>object-id</i> [, <i>database-id</i>]) |
| PROPERTY | (({ <i>property-id</i> <i>property-name</i> })) |

| システム関数 | パラメータ |
|----------------------|-----------------------------------------------|
| PROPERTY_DESCRIPTION | (<i>property-id</i> <i>property-name</i>) |
| PROPERTY_NAME | (<i>property-id</i>) |
| PROPERTY_NUMBER | (<i>property-name</i>) |
| SUSER_ID | ([<i>user-name</i>]) |
| SUSER_NAME | ([<i>user-id</i>]) |
| USER_ID | ([<i>user-name</i>]) |
| USER_NAME | ([<i>user-id</i>]) |

説明

サーバ上で現在実行中のデータベースは、データベース名とデータベース ID 番号で識別されます。db_id 関数と db_name 関数を使用して、これらの情報を取得できます。

システム関数のセットは、データベース・サーバ上で現在実行中のデータベースのプロパティや接続のプロパティに関する情報を提供します。これらのシステム関数は、データベース名、ID、または接続名をオプションの引数として使用し、プロパティが要求されているデータベースまたは接続を識別します。

パフォーマンス

システム関数は、他の Sybase IQ 関数とは異なる方法で処理されます。Sybase IQ テーブルへのクエリにシステム関数を含めると、パフォーマンスが低下します。

互換性

表 4-11 は、Adaptive Server Enterprise システム関数、および Sybase IQ におけるそれぞれのステータスを示します。

表 4-11 : Sybase IQ における ASE システム関数のステータス

| 関数 | ステータス |
|-------------|-------------|
| col_length | 実装済み |
| col_name | 実装済み |
| db_id | 実装済み |
| db_name | 実装済み |
| index_col | 実装済み |
| object_id | 実装済み |
| object_name | 実装済み |
| proc_role | 常にゼロを返す |
| show_role | 常に NULL を返す |
| tsequal | 実装されていない |
| user_id | 実装済み |
| user_name | 実装済み |
| suser_id | 実装済み |
| suser_name | 実装済み |

| 関数 | ステータス |
|------------------|----------|
| datalength | 実装済み |
| curunreservedpgs | 実装されていない |
| data_pgs | 実装されていない |
| host_id | 実装されていない |
| host_name | 実装されていない |
| lct_admin | 実装されていない |
| reserved_pgs | 実装されていない |
| rowcnt | 実装されていない |
| used_pgs | 実装されていない |
| valid_name | 実装されていない |
| valid_user | 実装されていない |

注意

- システム関数の中には、Sybase IQ にシステム・ストアド・プロシージャとして実装されているものがあります。
- db_id 関数、db_name 関数、datalength 関数、suser_id 関数、suser_name 関数は、組み込み関数として実装されています。

接続プロパティ

接続プロパティは、個々の接続に対して適用されます。この項では、特定の接続プロパティの値、またはすべての接続プロパティの値を取得する方法について説明します。すべての接続プロパティの詳細については、『SQL Anywhere Server データベース管理ガイド』の [Database properties \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/connection-properties.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/connection-properties.html) を参照してください。

例

❖ 接続プロパティの値を取得する

- connection_property システム関数を使用します。次の文は、現在の接続によってファイルから読み取られたページ数を返します。

```
select connection_property ( 'DiskRead' )
```

❖ すべての接続プロパティの値を取得する

- sa_conn_properties システム・プロシージャを使用します。

```
call sa_conn_properties
```

接続ごと、プロパティごとに、個別のローが表示されます。

サーバで使用可能なプロパティ

サーバ・プロパティは、サーバ全体に対して適用されます。この項では、特定のサーバ・プロパティの値、またはすべてのサーバ・プロパティの値を取得する方法について説明します。すべてのサーバ・プロパティの詳細については、『SQL Anywhere Server データベース管理ガイド』の [Database server properties \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/server-properties-perfapp.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/server-properties-perfapp.html) を参照してください。

例

❖ サーバ・プロパティの値を取り出す

- property プロパティ・システム関数を使用します。次の文は、メイン・ヒープの保持に使用されるキャッシュ・ページの数返します。

```
select property ( 'MainHeapPages') from iq_dummy
```

❖ すべてのサーバ・プロパティの値を取り出す

- sa_eng_properties システム・プロシージャを使用します。

```
call sa_eng_properties
```

各データベースで使用できるプロパティ

データベース・プロパティは、データベース全体に対して適用されません。この項では、特定のデータベース・プロパティの値、またはすべてのデータベース・プロパティの値を取得する方法について説明します。すべてのデータベース・プロパティの詳細については、『SQL Anywhere Server データベース管理ガイド』の [Database server properties \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/server-properties-perfapp.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/server-properties-perfapp.html) を参照してください。

例

❖ データベース・プロパティの値を取り出す

- db_property システム関数を使用します。次の文は、現在のデータベースのページ・サイズをバイト単位で返します。

```
select db_property ( 'PageSize') from iq_dummy
```

❖ すべてのデータベース・プロパティの値を取り出す

- sa_db_properties システム・プロシージャを使用します。

```
call sa_db_properties
```

SQL および Java のユーザ定義関数

Sybase IQ には、ユーザ定義関数を作成するための仕組みが 2 つ用意されています。SQL 言語または Java 言語を使用して、関数を作成することができます。

注意 ユーザ定義関数は、SQL Anywhere で処理されます。Sybase IQ のパフォーマンス機能は使用されません。そのため、ユーザ定義関数を含むクエリは、そうでないクエリと比較して、少なくとも 10 倍の処理時間が必要です。

ごくまれに、SQL Anywhere と Sybase IQ のセマンティクスの違いによって、ユーザ定義関数から発行されたクエリの結果に違いが生じることがあります。たとえば、Sybase IQ では、CHAR と VARCHAR が区別されて異なるデータ型をして扱われますが、SQL Anywhere では CHAR データ型は VARCHAR と同じように扱われます。

SQL でのユーザ定義関数

CREATE FUNCTION 文を使用して、独自の関数を SQL に実装することができます。関数のデータ型は、CREATE FUNCTION 文の内部の RETURN 文に定義します。

SQL ユーザ定義関数を作成したら、データ型が同じである組み込み関数で使用される場所であれば、どこでも使用できます。

注意 ユーザ定義関数を含むビューで、CONTAINS 述部を使用するのは (CONTAINS 基準が無視されるため) 避けてください。LIKE 述部を使用するか、ビューの外部でクエリを発行してください。

SQL 関数の作成の詳細については、『システム管理ガイド 第 2 巻』の「[第 1 章 プロシージャとバッチの使用](#)」を参照してください。

Java でのユーザ定義関数

SQL 関数も便利ですが、Java クラスを使用すれば、必要に応じてデータベース・サーバからクライアント・アプリケーションへ関数を移動できるなど Java 固有の利点を活用して、より強力かつ柔軟にユーザ定義関数を実装することができます。

インストールされた Java クラスの「**クラス・メソッド**」は、データ型が同じである組み込み関数で使用される場所であればどこでも、ユーザ定義関数として使用できます。

インスタンス・メソッドは、クラスの特定のインスタンスに関連付けられます。そのため、通常のユーザ定義関数と異なる動作をします。

Java クラスの作成およびクラス・メソッドの詳細については、『SQL Anywhere Server プログラミング・ガイド』の [Java support in SQL Anywhere \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/pg-java.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/pg-java.html) を参照してください。

時系列関数および予測関数

機能

時系列関数および予測関数は、金融時系列と予測分析のために特に設計された集合関数（「[集合関数](#)」(104 ページ)を参照）です。

注意 時系列および予測の機能は、RAP — The Trading Edition™ Enterprise でのみ使用可能です。

表 4-12 : 時系列関数

| 時系列関数 | パラメータ |
|---------------------|----------------------------------------------------------------------------------------------------------|
| TS_ARMA_AR | (<i>timeseries_expression</i> , <i>ar_count</i> , <i>ar_elem</i> , <i>method</i>) |
| TS_ARMA_CONST | (<i>timeseries_expression</i> , <i>method</i>) |
| TS_ARMA_MA | (<i>timeseries_expression</i> , <i>ma_count</i> , <i>ma_elem</i> , <i>method</i>) |
| TS_AUTOCORRELATION | (<i>timeseries_expression</i> , <i>lagmax</i> , <i>lag_elem</i>) |
| TS_AUTO_UNI_AR | (<i>timeseries_expression</i> , <i>ar_count</i> , <i>ar_elem</i> , <i>method</i>) |
| TS_BOX_COX_XFORM | (<i>timeseries_expression</i> , <i>power</i> [, <i>shift</i> [, <i>inverse</i>]]) |
| TS_DIFFERENCE | (<i>timeseries_expression</i> , <i>period1</i> [, <i>period2</i> [, ... <i>period10</i>]]) |
| TS_ESTIMATE_MISSING | (<i>timeseries_expression</i> , <i>method</i>) |
| TS_LACK_OF_FIT | (<i>timeseries_expression</i> , <i>p_value</i> , <i>q_value</i> , <i>lagmax</i> , [<i>tolerance</i>]) |
| TS_LACK_OF_FIT_P | (<i>timeseries_expression</i> , <i>p_value</i> , <i>q_value</i> , <i>lagmax</i> , [<i>tolerance</i>]) |
| TS_MAX_ARMA_AR | (<i>timeseries_expression</i> , <i>ar_count</i> , <i>ar_elem</i>) |
| TS_MAX_ARMA_CONST | (<i>timeseries_expression</i>) |

| 時系列関数 | パラメータ |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TS_MAX_ARMA_MA | (<i>timeseries_expression</i> , <i>ma_count</i> , <i>ma_elem</i>) |
| TS_MAX_ARMA_LIKELIHOOD | (<i>timeseries_expression</i>) |
| TS_OUTLIER_IDENTIFICATION | (<i>timeseries_expression</i> , <i>p_value</i> , <i>q_value</i> , <i>s_value</i> , <i>d_value</i> , [, <i>delta_value</i> [, <i>critical_value</i>]]) |
| TS_PARTIAL_AUTOCORRELATION | (<i>timeseries_expression</i> , <i>lagmax</i> , <i>lag_elem</i>) |
| TS_VWAP | (<i>price_expression</i> , <i>volume_expression</i>) |

時系列関数および予測関数の IMSL ライブラリのロード

時系列関数および予測関数は、サードパーティ製の 2 つの外部ライブラリを呼び出します。Visual Numerics Inc. の IMSL™ C Stat ライブラリおよび C Math ライブラリには、金融時系列および予測計算のための C 関数が含まれています。

ラップ・ライブラリ **libtsudf** には、IMSL C Stat ライブラリと C Math ライブラリに含まれている関数を呼び出すユーザ定義関数 (UDF) が含まれています。

時系列 SQL 関数と予測 SQL 関数は、**libtsudf** ラップ・ライブラリを自動的に呼び出します。ユーザが時系列および予測分析用の有効なユーザ定義集合関数を呼び出すと、Sybase IQ は IMSL C Stat ライブラリおよび C Math ライブラリをロードします。

IMSL C Stat ライブラリおよび C Math ライブラリは、Sybase IQ がインストールされているプラットフォームにより異なります。

表 4-13 : IMSL ライブラリの位置およびファイル名

| | Windows 32 ビット版 | Windows 64 ビット版 | UNIX (AIX を除く) | AIX |
|--------------|--------------------------------------------------|--------------------------------------------------|----------------------------------------------|--------------------------------------------------|
| ライブラリのディレクトリ | bin32 | bin64 | lib64 | lib64 |
| ライブラリ・ファイル名 | imslcmath_imsl_dll.dll imslcstat_imsl_dll.dll | imslcmath_imsl_dll.dll imslcstat_imsl_dll.dll | libimslcmath_imsl.so libimslcstat_imsl.so | libimslcmath_imsl_r.so libimslcstat_imsl_r.so |

IMSL ライブラリ時系列関数のエラー処理

IMSL ライブラリを呼び出す時系列関数のエラー処理の動作を制御できます。IMSL ライブラリ関数の呼び出し時に実行時エラーが発生すると、Sybase IQ は、ユーザが選択したエラー処理オプションに従って応答します。エラー処理オプションは、すべてのエラーと警告を無視するデフォルトの動作に加えて、Sybase IQ で SQL 文をアボートしてエラー・メッセージを返すエラーの重大度を指定する 4 つのオプションから選択できます。

エラー処理を制御するには、次の SQL 文を使用します。

```
set option PUBLIC.Time_Series_Error_Level = '<value>'
```

有効な値は次のとおりです。

表 4-14 : IMSL ライブラリ時系列関数のエラー処理

| 値 | 説明 |
|-----------|------------------------------------------------------------------------------------------------------------------------------|
| 0 (デフォルト) | IMSL ライブラリ関数の呼び出し中に取得されるすべての種類の警告およびエラーが無視されます。そのような状況が発生すると、時系列関数は NULL 値を返します。 |
| 1 | IMSL ライブラリ関数の呼び出し中に時系列関数が警告またはエラー・メッセージを取得すると、IQ はエラー・メッセージを返し、SQL クエリをアボートします。 |
| 2 | IMSL ライブラリ関数の呼び出し中に時系列関数が致命的なエラー・メッセージを取得すると、IQ はエラー・メッセージを返し、SQL クエリをアボートします。しかし、警告が取得された場合、時系列関数は NULL 値を返します。 |
| 3 | IMSL ライブラリ関数の呼び出し中に時系列関数がターミナル・エラー・メッセージを取得すると、IQ はエラー・メッセージを返し、SQL クエリをアボートします。しかし、警告または致命的なエラーが取得された場合、時系列関数は NULL 値を返します。 |

IMSL ライブラリ時系列関数のエラー・ロギング

Sybase IQ が IMSL ライブラリ時系列関数エラー処理によって返されるエラー・メッセージをログする方法を制御できます。エラー・メッセージをログ・ファイルにログする 4 つのオプションがあります。

エラー・ロギングを制御するには、次の SQL 文を使用します。

```
set option PUBLIC.Time_Series_Log_Level = '<value>'
```

表 4-15 : IMSL ライブラリ時系列関数のエラー・ロギング

| 値 | 説明 |
|-----------|-----------------------------------------------------------------------------------------|
| 0 (デフォルト) | IMSL ライブラリ関数の呼び出し中に返されたすべての警告およびエラーは無視され、ログ・ファイルにログされません。 |
| 1 | IMSL ライブラリ関数の呼び出し中に時系列関数が警告またはエラー・メッセージを返すと、メッセージがログ・ファイルにログされます。 |
| 2 | IMSL ライブラリ関数の呼び出し中に時系列関数が致命的なエラー・メッセージを返すと、メッセージがログ・ファイルにログされます。警告はログされません。 |
| 3 | IMSL ライブラリ関数の呼び出し中に時系列関数がターミナル・エラー・メッセージを返すと、メッセージがログ・ファイルにログされます。警告および致命的なエラーはログされません。 |

その他の関数

機能

その他の関数は、他の関数の戻り値を含めた、算術式、文字列式、または日付/時刻式を操作します。

表 4-16 は、その他の関数とパラメータを示します。

表 4-16 : その他の関数

| その他の関数 | パラメータ |
|----------|-----------------------------------------------------------------------|
| ARGN | (<i>integer-expr</i> , <i>expression</i> [, ...]) |
| COALESCE | (<i>expression</i> , <i>expression</i> [, <i>expression</i> ...]) |

| その他の関数 | パラメータ |
|--------|-----------------------------------------------------------------------|
| IFNULL | (<i>expression1</i> , <i>expression2</i> [, <i>expression3</i>]) |
| ISNULL | (<i>expression</i> , <i>expression</i> [, <i>expression</i> ...]) |
| NULLIF | (<i>expression1</i> , <i>expression2</i>) |
| NUMBER | (*) |
| ROWID | (<i>table-name</i>) |

互換性

Adaptive Server Enterprise では、COALESCE 関数、ISNULL 関数、および NULLIF 関数はサポートされていません。

アルファベット順の関数リスト

この項では、それぞれの関数について個別に説明します。関数の種類 (数値、文字列など) は、関数名の隣に角カッコで囲んで示してあります。

例として示す実行結果の一部に、丸めまたはトランケートされているものがあります。

テーブルのオブジェクト ID やカラム ID など、例として示すデータベースのオブジェクト ID の値は、実際の値と異なる場合があります。

ABS 関数 [数値]

機能

数値式の絶対値を返します。

構文

ABS (*numeric-expression*)

パラメータ

numeric-expression 絶対値を返す数値。

例

次の文は、値 66 を返します。

```
SELECT ABS ( -66 ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

ACOS 関数 [数値]

機能

数値式のアーク・コサインをラジアンで返します。

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | ACOS (<i>numeric-expression</i>) |
| パラメータ | numeric-expression 角度の余弦。 |
| 例 | 次の文は、値 1.023945 を返します。 <pre>SELECT ACOS (0.52) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |
| 参照 | 「ASIN 関数 [数値]」 (130 ページ) 「ATAN 関数 [数値]」 (130 ページ) 「ATAN2 関数 [数値]」 (131 ページ) 「COS 関数 [数値]」 (143 ページ) |

ARGN 関数 [その他]

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 引数のリストから選択した引数を返します。 |
| 構文 | ARGN (<i>integer-expression</i> , <i>expression</i> [, ...]) |
| パラメータ | <p>integer-expression 式のリスト内での引数の位置。</p> <p>expression 関数に渡される、任意のデータ型の式。ここに記述する式のデータ型は、すべて同じであることが必要です。</p> |
| 例 | 次の文は、値 6 を返します。 <pre>SELECT ARGN (6, 1,2,3,4,5,6) FROM iq_dummy</pre> |
| 使用法 | <i>integer-expression</i> の値を <i>n</i> として、その後続く引数リストの <i>n</i> 番目の引数 (1 で開始) を返します。式のデータ型に制限はありませんが、すべての式でデータ型が同じであることが必要です。整数式には、1 からリストにある式の数までの値を指定します。それ以外を指定すると、NULL が返されます。式は、カンマで区切って入力します。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |

ASCII 関数 [文字列]

| | |
|----|-------------------------------------------|
| 機能 | 文字列式の先頭バイトの ASCII 値を整数で返します。 |
| 構文 | ASCII (<i>string-expression</i>) |

| | |
|--------|---------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | string-expression 文字列。 |
| 例 | 照合順序にデフォルトの ISO_BINENG が設定されている場合、次の文を実行すると値 90 が返ります。 <pre>SELECT ASCII('Z') FROM iq_dummy</pre> |
| 使用法 | 文字列が空の場合、ASCII ではゼロが返されます。リテラル文字列は引用符で囲む必要があります。 |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 ベンダの拡張機能。• Sybase Adaptive Server Enterprise 互換。 |

ASIN 関数 [数値]

| | |
|--------|---------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 数値のアーク・サインをラジアンで返します。 |
| 構文 | ASIN (<i>numeric-expression</i>) |
| パラメータ | numeric-expression 角度の正弦。 |
| 例 | 次の式を実行すると、値 0.546850 が返ります。 <pre>SELECT ASIN(0.52) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 ベンダの拡張機能。• Sybase Adaptive Server Enterprise 互換。 |

参照

- [「ACOS 関数 \[数値\]」 \(128 ページ\)](#)
- [「ATAN 関数 \[数値\]」 \(130 ページ\)](#)
- [「ATAN2 関数 \[数値\]」 \(131 ページ\)](#)
- [「SIN 関数 \[数値\]」 \(251 ページ\)](#)

ATAN 関数 [数値]

| | |
|-------|------------------------------------------------------------------------|
| 機能 | 数値のアーク・タンジェントをラジアンで返します。 |
| 構文 | ATAN (<i>numeric-expression</i>) |
| パラメータ | numeric-expression 角度の正接。 |
| 例 | 次の文は、値 0.479519 を返します。 <pre>SELECT ATAN(0.52) FROM iq_dummy</pre> |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase Adaptive Server Enterprise** 互換。
- 参照
- 「ACOS 関数 [数値]」 (128 ページ)
 - 「ASIN 関数 [数値]」 (130 ページ)
 - 「ATAN2 関数 [数値]」 (131 ページ)
 - 「TAN 関数 [数値]」 (268 ページ)

ATAN2 関数 [数値]

- 機能
- 2 つの数値の比率のアーク・タンジェントをラジアンで返します。
- 構文
- ATAN2** (*numeric-expression1*, *numeric-expression2*)
- パラメータ
- numeric-expression1** アーク・タンジェントを計算する比率における分子。
- numeric-expression2** アーク・タンジェントを計算する比率における分母。
- 例
- 次の文は、値 0.00866644968879073143 を返します。
- ```
SELECT ATAN2(0.52, 060) FROM iq_dummy
```
- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** ATAN2 は、Adaptive Server Enterprise ではサポートされていません。

- 参照
- 「ACOS 関数 [数値]」 (128 ページ)
  - 「ASIN 関数 [数値]」 (130 ページ)
  - 「ATAN 関数 [数値]」 (130 ページ)
  - 「TAN 関数 [数値]」 (268 ページ)

## AVG 関数 [集合]

- 機能
- ロー・セットに対する数値式の平均を計算します。または、ユニークな値のセットの平均を計算します。
- 構文
- AVG** ( *numeric-expression* | **DISTINCT** *column-name* )
- パラメータ
- numeric-expression** ロー・セットに対して平均を計算する値。

**DISTINCT column-name** *column-name* の一意な値の平均を計算します。この方法でを使用することはほとんどありませんが、万全を期すために含まれています。

例 次の文は、値 49988.6 を返します。

```
SELECT AVG (salary) FROM Employees
```

使用法 この平均には、*numeric-expression* が NULL 値であるローは含まれません。ローがまったくないグループに対しては、NULL 値が返されます。

標準と互換性

- **SQL92** SQL92 互換。
- **Sybase** Adaptive Server Enterprise 互換。

参照

[「COUNT 関数 \[集合\]」 \(146 ページ\)](#)

[「SUM 関数 \[集合\]」 \(266 ページ\)](#)

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

## BIGINTTOHEX 関数 [データ型変換]

機能 10 進の整数を VARCHAR(16) の同等の 16 進数に変換して返します。

構文 **BIGINTTOHEX** ( *integer-expression* )

パラメータ **integer-expression** 16 進数に変換される整数。

例 次の文は、値 0000000000000009 を返します。

```
SELECT BIGINTTOHEX(9) FROM iq_dummy
```

次の文は、値 FFFFFFFF7 を返します。

```
SELECT BIGINTTOHEX(-9) FROM iq_dummy
```

使用法 **BIGINTTOHEX** 関数は、**BIGINT** に評価される整数式を処理し、等価の 16 進文字列を返します。返された値は 16 桁になるまで左側に 0 が埋め込まれます。位取りされていない整数データ型のすべての型が、整数式として処理されます。

必要に応じて、変換は自動的に実行されます。小数值がゼロの場合にのみ、定数はトランケートされます。カラムが正の位取り値で宣言されている場合、カラムはトランケートできません。変換に失敗すると、**CONVERSION\_ERROR** オプションが **OFF** に設定されていない場合、Sybase IQ はエラーを返します。**OFF** に設定されている場合は、NULL を返します。

|        |                                                                                                                                                                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> Transact SQL の拡張機能。</li> <li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li> </ul>                                                                                                        |
| 参照     | <p>『リファレンス: 文とオプション』の「<a href="#">CONVERSION_ERROR オプション [TSQL]</a>」</p> <p><a href="#">「HEXTOBIGINT 関数 [データ型変換]」 (179 ページ)</a></p> <p><a href="#">「HEXTOINT 関数 [データ型変換]」 (180 ページ)</a></p> <p><a href="#">「INTTOHEX 関数 [データ型変換]」 (191 ページ)</a></p> |

## BIT\_LENGTH 関数 [文字列]

|        |                                                                                      |
|--------|--------------------------------------------------------------------------------------|
| 機能     | カラム・パラメータのビット長を、符号なし 64 ビット値で返します。                                                   |
| 構文     | <b>BIT_LENGTH</b> ( <i>column-name</i> )                                             |
| パラメータ  | <b>column-name</b> カラムの名前。                                                           |
| 使用法    | <p>引数が NULL の場合は、NULL 値を返します。</p> <p>BIT_LENGTH 関数は、すべての Sybase IQ データ型をサポートします。</p> |
| 標準と互換性 | <b>Sybase</b> SQL Anywhere および Adaptive Server Enterprise ではサポートされていません。             |
| 参照     | <a href="#">「OCTET_LENGTH 関数 [文字列]」 (217 ページ)</a>                                    |

## BYTE\_LENGTH 関数 [文字列]

|       |                                                                                                                                               |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | 文字列のバイト数を返します。                                                                                                                                |
| 構文    | <b>BYTE_LENGTH</b> ( <i>string-expression</i> )                                                                                               |
| パラメータ | <b>string-expression</b> 長さを計算する文字列。                                                                                                          |
| 例     | <p>次の文は、値 12 を返します。</p> <pre>SELECT BYTE_LENGTH( 'Test Message' ) FROM iq_dummy</pre>                                                         |
| 使用法   | <p>末尾にあるスペース文字を含めた長さが返されます。</p> <p>NULL 文字を指定すると、NULL 値が返ります。</p> <p>文字列がマルチバイトの文字セットで指定された場合、BYTE_LENGTH が返す文字数は、CHAR_LENGTH と同じではありません。</p> |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

[「CHAR\\_LENGTH 関数 \[文字列\]」 \(136 ページ\)](#)  
[「DATALENGTH 関数 \[システム\]」 \(148 ページ\)](#)  
[「LENGTH 関数 \[文字列\]」 \(198 ページ\)](#)

## CAST 関数 [データ型変換]

機能 式を指定されたデータ型に変換した値が返されます。

構文 **CAST ( *expression AS data type* )**

パラメータ **expression** 変換される式。  
**data type** 変換後のデータ型。

例 次のように関数を指定すると、文字列が日付として使用されます。

```
CAST ('2000-10-31' AS DATE)
```

次では、式 `1 + 2` の値が計算され、結果が 1 文字の文字列に、データ・サーバが割り当てる長さでキャストされます。

```
CAST (1 + 2 AS CHAR)
```

次のように、CAST 関数を使用して文字列を短縮できます。

```
SELECT CAST (Surname AS CHAR(5)) FROM Customers
```

使用法 文字列型への変換に長さを指定しない場合は、Sybase IQ によって適切な長さが選択されます。DECIMAL 変換に精度も小数位桁数も指定されない場合は、データベース・サーバが適切な値を設定します。

NULL から NUMERIC の明示的な変換に、精度も位取りも指定されない場合は、デフォルトで NUMERIC(1,0) が設定されます。例を示します。

```
SELECT CAST (NULL AS NUMERIC) A,
 CAST (NULL AS NUMERIC (15,2)) B
```

上の式は次のように記述されます。

```
A NUMERIC (1,0)
B NUMERIC (15,2)
```

- 標準と互換性
- **SQL92** この関数は SQL92 互換です。
  - **Sybase** Adaptive Server Enterprise 互換。

参照 [「CONVERT 関数 \[データ型変換\]」 \(140 ページ\)](#)



## CEIL 関数 [数値]

|        |                                                                                                                                        |
|--------|----------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | 指定された式以上の最小の整数を返します。<br>CEIL は CEILING と同意語です。                                                                                         |
| 構文     | <b>CEIL</b> ( <i>numeric-expression</i> )                                                                                              |
| パラメータ  | <b>expression</b> カラム、変数、またはデータ型が真数値、概数値、通貨、またはこれらの型の 1 つに暗黙的に変換できる式です。他のデータ型を指定すると、 <b>CEIL</b> ではエラーが返ります。戻り値のデータ型は、指定した値のデータ型と同じです。 |
| 使用法    | 指定された式について、 <b>CEIL</b> 関数は 1 つの引数を取ります。たとえば、 <b>CEIL</b> (-123.45) は、-123 を返し、 <b>CEIL</b> (123.45) は 124 を返します。                      |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> ベンダの拡張機能。</li> <li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li> </ul>     |
| 参照     | <a href="#">「CEILING 関数 [数値]」 (135 ページ)</a><br><a href="#">『システム管理ガイド 第 1 巻』の「第 11 章 各国語と文字セット」</a>                                    |

## CEILING 関数 [数値]

|        |                                                                                                                                                                                                                                                                                                         |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | 数値の上限値 (その値以上の最も小さい整数) を返します。<br>CEIL は CEILING と同意語です。                                                                                                                                                                                                                                                 |
| 構文     | <b>CEILING</b> ( <i>numeric-expression</i> )                                                                                                                                                                                                                                                            |
| パラメータ  | <b>numeric-expression</b> 上限値を計算する数値。                                                                                                                                                                                                                                                                   |
| 例      | 次の文は、値 60.00000 を返します。<br><pre>SELECT CEILING( 59.84567 ) FROM iq_dummy</pre> 次の文は、値 123 を返します。<br><pre>SELECT CEILING( 123 ) FROM iq_dummy</pre> 次の文は、値 124.00 を返します。<br><pre>SELECT CEILING( 123.45 ) FROM iq_dummy</pre> 次の文は、値 -123.00 を返します。<br><pre>SELECT CEILING( -123.45 ) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> ベンダの拡張機能。</li> <li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li> </ul>                                                                                                                                                                      |

参照 [「FLOOR 関数 \[数値\]」 \(174 ページ\)](#)

## CHAR 関数 [文字列]

機能 数値を ASCII 値とした文字を返します。

構文 **CHAR** ( *integer-expression* )

パラメータ **integer-expression** ASCII 文字に変換する数値。数値には、0 から 255 まで (0 と 255 を含む) を指定してください。

例 次の式を実行すると、値 “Y” が返ります。

```
SELECT CHAR(89) FROM iq_dummy
```

次の文は、値 “S” を返します。

```
SELECT CHAR(83) FROM iq_dummy
```

使用法 与えられた数値を 256 で割った余りの値に対応する、現在のデータベースの文字セットに含まれる文字が返されます。

255 より大きい値や、ゼロより小さい値を数値式に指定すると、NULL が返されます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** 互換。

## CHAR\_LENGTH 関数 [文字列]

機能 文字列の文字数を返します。

構文 **CHAR\_LENGTH** ( *string-expression* )

パラメータ **string-expression** 長さを計算する文字列。

使用法 末尾にあるスペース文字を含めた長さが返されます。

NULL 文字を指定すると、NULL 値が返ります。

マルチバイト文字セットの場合、CHAR\_LENGTH の結果は BYTE\_LENGTH の結果より小さくなる場合もあります。

例 次の文は、値 8 を返します。

```
SELECT CHAR_LENGTH('Chemical') FROM iq_dummy
```

## 標準と互換性

- **SQL92** この関数は SQL92 互換です。
- **Sybase** Adaptive Server Enterprise 互換。

## 参照

「[BYTE\\_LENGTH 関数 \[文字列\]](#)」 (133 ページ)

## CHARINDEX 関数 [文字列]

## 機能

指定した文字列が別の文字列で最初に出現する位置を返します。

## 構文

**CHARINDEX** ( *string-expression1*, *string-expression2* )

## パラメータ

**string-expression1** 検索する文字列。255 バイトまでの文字列を指定してください。

**string-expression2** 検索される文字列。検索される文字列内で、先頭の文字の位置が 1 になります。

## 例

次に例を示します。

```
SELECT Surname, GivenName
FROM Employees
WHERE CHARINDEX('K', Surname) = 1
```

この例では、次の値が返されます。

| Surname   | GivenName |
|-----------|-----------|
| Klobucher | James     |
| Kuo       | Felicia   |
| Kelly     | Moira     |

## 使用法

CHARINDEX 関数で返されるか指定される位置またはオフセットはすべて、常に文字オフセットであり、マルチバイト・データの場合はバイト・オフセットとは異なることがあります。

指定した文字列が検索対象の文字列に 2 つ以上含まれる場合は、最初の文字列の位置が返されます。

指定した文字列が検索対象の文字列に含まれない場合は、ゼロ (0) が返されます。

長さが 0 の文字列を検索すると、1 が返されます。

引数のどれか 1 つでも NULL の場合、結果は NULL になります。

CHARINDEX は CHAR カラムと VARCHAR カラムの場合、32 ビット符号付き整数で位置を返します。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise 互換。

参照 [「SUBSTRING 関数 \[文字列\]」 \(265 ページ\)](#)

『Sybase IQ によるラージ・オブジェクト管理』の「[第 4 章 関数のサポート](#)」

## COALESCE 関数 [その他]

機能 リストから、最初の NULL でない式を返します。

構文 **COALESCE** ( *expression*, *expression* [ , ... ] )

パラメータ **expression** 任意の式。

例 次の文は、値 34 を返します。

```
SELECT COALESCE(NULL, 34, 13, 0) FROM iq_dummy
```

- 標準と互換性
- **SQL92** SQL92。
  - **Sybase** Adaptive Server Enterprise 互換。

## COL\_LENGTH 関数 [システム]

機能 カラムの定義済みの長さを返します。

構文 **COL\_LENGTH** ( *table-name*, *column-name* )

パラメータ **table-name** テーブル名。

**column-name** カラム名。

例 次の文は、カラム長 35 を返します。

```
SELECT COL_LENGTH ('CUSTOMERS', 'Street') FROM
iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。

参照 [「DATALENGTH 関数 \[システム\]」 \(148 ページ\)](#)

## COL\_NAME 関数 [システム]

|        |                                                                                                                                                                                                                                                                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | カラム名を返します。                                                                                                                                                                                                                                                                                                                                                    |
| 構文     | <b>COL_NAME</b> ( <i>table-id</i> , <i>column-id</i> [, <i>database-id</i> ] )                                                                                                                                                                                                                                                                                |
| パラメータ  | <b>table-id</b> テーブルのオブジェクト ID。<br><b>column-id</b> カラムの ID。<br><b>database-id</b> データベース ID。                                                                                                                                                                                                                                                                 |
| 例      | 次の文は、カラム名 <i>lname</i> を返します。Customers テーブルのオブジェクト ID は 100209 です。この値は OBJECT_ID 関数で取得できます。カラム ID は、syscolumn システム・テーブルの <i>column_id</i> カラムに格納されています。iqdemo データベースのデータベース ID は 0 です。この値は DB_ID 関数で取得できます。<br><pre>SELECT COL_NAME( 100209, 3, 0 ) FROM iq_dummy</pre> 次の文は、カラム名 <i>city</i> を返します。<br><pre>SELECT COL_NAME ( 100209, 5 )FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> ベンダの拡張機能。</li> <li>• <b>Sybase</b> Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。</li> </ul>                                                                                                                                                                                                        |
| 参照     | <a href="#">「DB_ID 関数 [システム]」 (162 ページ)</a><br><a href="#">「OBJECT_ID 関数 [システム]」 (216 ページ)</a>                                                                                                                                                                                                                                                                |

## CONNECTION\_PROPERTY 関数 [システム]

|    |                                                                                                                                 |
|----|---------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定された接続プロパティの値を文字列で返します。                                                                                                        |
| 構文 | <b>CONNECTION_PROPERTY</b> ( { <i>integer-expression1</i>   <i>string-expression</i> }<br>... [, <i>integer-expression2</i> ] ) |

---

**注意** CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

---

|       |                                                                                                                                                        |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | <b>integer-expression1</b> ほとんどの場合、文字列式を最初の引数に指定したほうが便利です。 <i>integer-expression1</i> を指定した場合は、接続プロパティ ID となります。この値は PROPERTY_NUMBER 関数を使って調べることができます。 |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------|

**string-expression** 接続プロパティ名。プロパティ ID またはプロパティ名を指定する必要があります。

**integer-expression2** 現在のデータベース接続の接続 ID。この引数を省略すると、現在の接続を使用します。

例 次の文は、保持している準備文の数 (4 など) を返します。

```
SELECT connection_property('PrepStmt') FROM iq_dummy
```

使用法 第 2 の引数を省略すると、現在の接続を使用します。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

参照

[「接続プロパティ」 \(121 ページ\)](#)

[「PROPERTY\\_NUMBER 関数 \[システム\]」 \(228 ページ\)](#)

## CONVERT 関数 [データ型変換]

機能 式を指定されたデータ型に変換して返します。

構文 **CONVERT ( data-type, expression [ , format-style ] )**

パラメータ **data-type** 式の変換後のデータ型。

**expression** 変換される式。

**format-style** 文字列から日付または時刻データ型への変換 (あるいはその逆) では、使用する日付フォーマット文字列を表すスタイル・コード番号を *format-style* に指定します。表 4-17 は *format-style* 引数の値の意味を示します。

表 4-17 : CONVERT でのスタイル・コードの出力形式

| 世紀なし (yy) | 世紀あり (yyyy) | 出力                           |
|-----------|-------------|------------------------------|
| -         | 0 または 100   | mmm dd yyyy hh:nnAM (または PM) |
| 1         | 101         | mm/dd/yy[yy]                 |
| 2         | 102         | [yy]yy.mm.dd                 |
| 3         | 103         | dd/mm/yy[yy]                 |
| 4         | 104         | dd.mm.yy[yy]                 |
| 5         | 105         | dd-mm-yy[yy]                 |
| 6         | 106         | dd mmm yy[yy]                |
| 7         | 107         | mmm dd, yy[yy]               |
| 8         | 108         | hh:nn:ss                     |

| 世紀なし (yy) | 世紀あり (yyyy) | 出力                                                     |
|-----------|-------------|--------------------------------------------------------|
| -         | 9 または 109   | mmm dd yyyy hh:nn:ss:sssAM (または PM)                    |
| 10        | 110         | mm-dd-yy[yy]                                           |
| 11        | 111         | [yy]yy/mm/dd                                           |
| 12        | 112         | [yy]yymmdd                                             |
| 13        | 113         | dd mmm yyyy hh:nn:ss:sss (24 時間形式、欧州デフォルト + ミリ秒、年 4 桁) |
| 14        | 114         | hh:nn:ss (24 時間形式)                                     |
| 20        | 120         | yyyy-mm-dd hh:nn:ss (24 時間表示、ODBC 標準、年 4 桁)            |
| 21        | 121         | yyyy-mm-dd hh:nn:ss:sss (24 時間表示、ODBC 標準 + ミリ秒、年 4 桁)  |
| -         | 365         | yyyyjjj (文字列または整数で指定。jjj は年内のユリウス日の番号 (1 ~ 366))       |

*format-style* 引数を指定していない場合は、スタイル・コード **0** が使用されます。

例

*format-style* は次の文のように使用します。

```
SELECT CONVERT(CHAR(20), orderdate, 104)
FROM salesorder
```

#### **orderdate**

---

```
16.03.1993
20.03.1993
23.03.1993
25.03.1993
...
```

```
SELECT CONVERT(CHAR(20), orderdate, 7)
FROM salesorder
```

#### **orderdate**

---

```
mar 16, 93
mar 20, 93
mar 23, 93
mar 25, 93
...
```

365 の `format-style` は、次の文のように使用します。この文では、`DATE` 型および `DATETIME` 型のデータが、文字列型または整数型のデータ (およびその逆) に変換されます。

```
CREATE TABLE tab
 (date_col DATE, int_col INT, char7_col CHAR(7));
INSERT INTO tab (date_col, int_col, char7_col)
 VALUES ('Dec 17, 2004', 2004352, '2004352');
```

`SELECT CONVERT(VARCHAR(8), tab.date_col, 365) FROM tab;`  
`'2004352'` が返ります。

`SELECT CONVERT(INT, tab.date_col, 365) from tab;`  
`2004352` が返ります。

`SELECT CONVERT(DATE, tab.int_col, 365) FROM TAB;`  
`2004-12-17` が返ります。

`SELECT CONVERT(DATE, tab.char7_col, 365) FROM tab;`  
`2004-12-17` が返ります。

次の文は、整数への変換を示します。この文では値 `5` が返されます。

```
SELECT CONVERT(integer, 5.2) FROM iq_dummy
```

#### 使用法

`CONVERT` 関数の結果データ型は、`LONG VARCHAR` です。`SELECT INTO` 文で `CONVERT` を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、`CAST` を使用して `CONVERT` を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」を参照してください。

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise および SQL Anywhere 互換です。ただし、365 の `format-style` を除きます。この番号は、IQ でのみ使用できます。

#### 参照

[「CAST 関数 \[データ型変換\]」 \(134 ページ\)](#)

## CORR 関数 [集合]

#### 機能

一連の数値ペアの相関係数を返します。

#### 構文 1

```
CORR (dependent-expression, independent-expression)
```

#### 構文 2

```
CORR (dependent-expression, independent-expression)
OVER (window-spec)
```

*column-spec* : 以下の「使用法」の構文 2 の手順を参照してください。



|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ  | <b>dependent-expression</b> independent-expression の影響を受ける変数。<br><b>independent-expression</b> 結果に影響を与える変数。                                                                                                                                                                                                                                                                                                                                                                                         |
| 例      | 次の例は、相関を実行して年齢が収入に関連付けられているかどうかを調べます。この関数は、値 0.440227 を返します。<br><pre>SELECT CORR( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) ) FROM Employees;</pre>                                                                                                                                                                                                                                                                                                                                            |
| 使用法    | CORR 関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。<br><br><i>dependent-expression</i> および <i>independent-expression</i> は両方とも数値です。関数は、 <i>dependent-expression</i> または <i>independent-expression</i> が NULL のペアを排除した後、( <i>dependent-expression</i> , <i>independent-expression</i> ) のセットに適用されます。次の計算が行われます。<br><b>COVAR_POP (y, x) / (STDDEV_POP (x) * STDDEV_POP (y))</b><br>ここで、x は <i>dependent-expression</i> を表し、y は <i>independent-expression</i> を表します。 |
|        | <hr/> <b>注意</b> ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。 <hr/>                                                                                                                                                                                                                                                                                                                                                                                                                                |
|        | 構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで <i>window-spec</i> の要素を指定できます。ウィンドウの指定方法については、「 <a href="#">分析関数</a> 」(106 ページ) を参照してください。                                                                                                                                                                                                                                                                                                                      |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL2008</b> 主要な SQL に含まれない、SQL/foundation の機能です。</li> <li>• <b>Sybase</b> SQL Anywhere 互換。</li> </ul>                                                                                                                                                                                                                                                                                                                                                   |

## COS 関数 [数値]

|       |                                                                       |
|-------|-----------------------------------------------------------------------|
| 機能    | 数値の余弦をラジアンで返します。                                                      |
| 構文    | <b>COS ( numeric-expression )</b>                                     |
| パラメータ | <b>numeric-expression</b> 角度 (ラジアン)。                                  |
| 例     | 次の文は、値 0.86781 を返します。<br><pre>SELECT COS ( 0.52 ) FROM iq_dummy</pre> |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase Adaptive Server Enterprise** 互換。

- 参照
- 「ACOS 関数 [数値]」(128 ページ)
  - 「COT 関数 [数値]」(144 ページ)
  - 「SIN 関数 [数値]」(251 ページ)
  - 「TAN 関数 [数値]」(268 ページ)

## COT 関数 [数値]

機能 数値の余接をラジアンで返します。

構文 **COT** ( *numeric-expression* )

パラメータ **numeric-expression** 角度 (ラジアン)。

例 次の文は、値 1.74653 を返します。

```
SELECT COT(0.52) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase Adaptive Server Enterprise** 互換。

- 参照
- 「COS 関数 [数値]」(143 ページ)
  - 「SIN 関数 [数値]」(251 ページ)
  - 「TAN 関数 [数値]」(268 ページ)

## COVAR\_POP 関数 [集合]

機能 一連の数値ペアの母共分散を返します。

構文 1 **COVAR\_POP** ( *dependent-expression*, *independent-expression* )

構文 2 **COVAR\_POP** ( *dependent-expression*, *independent-expression* )  
**OVER** ( *window-spec* )

*column-spec* : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ **dependent-expression** 独立した変数の影響を受ける変数。

**independent-expression** 結果に影響を与える変数。

例 次の例は、従業員の年齢と給与の関連性の度合いを測定します。この関数は、値 73785.840059 を返します。

```
SELECT COVAR_POP(Salary, (YEAR(NOW()) - YEAR(
BirthDate))) FROM Employees;
```

使用法 この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

*dependent-expression* および *independent-expression* は両方とも数値です。関数は、*dependent-expression* または *independent-expression* が NULL のペアを排除した後、(*dependent-expression*, *independent-expression*) のセットに適用されます。次の計算が行われます。

$$(\text{SUM}(x*y) - \text{SUM}(x) * \text{SUM}(y) / n) / n$$

ここで、x は *dependent-expression* を表し、y は *independent-expression* を表します。

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage\\_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

---

**注意** ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

---

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能です。
- **Sybase** SQL Anywhere 互換。

## COVAR\_SAMP 関数 [集合]

機能 一連の数値ペアの標本共分散を返します。

構文 1 **COVAR\_SAMP** (*dependent-expression*, *independent-expression*)

構文 2 **COVAR\_SAMP** (*dependent-expression*, *independent-expression*)  
**OVER** (*window-spec*)

*column-spec* : 以下の「使用法」の構文 2 の手順を参照してください。

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ  | <p><b>dependent-expression</b> 独立した変数の影響を受ける変数。</p> <p><b>independent-expression</b> 結果に影響を与える変数。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 例      | <p>次の例は、従業員の年齢と給与の関連性の度合いを測定します。この関数は、値 74782.946005 を返します。</p> <pre>SELECT COVAR_SAMP( Salary, ( 2008 - YEAR( BirthDate ) ) ) FROM Employees;</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 使用法    | <p>この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。</p> <p><i>dependent-expression</i> および <i>independent-expression</i> は両方とも数値です。関数は、<i>dependent-expression</i> または <i>independent-expression</i> が NULL のペアを排除した後、(<i>dependent-expression</i>, <i>independent-expression</i>) のセットに適用されます。</p> <p><b>(SUM(x*y) - SUM(x) * SUM(y) / n) / (n-1)</b></p> <p>ここで、x は <i>dependent-expression</i> を表し、y は <i>independent-expression</i> を表します。</p> <p>『SQL Anywhere サーバー SQL の使用法』の <a href="http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html">Mathematical formulas for the aggregate functions (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html)</a> を参照してください。</p> <hr/> <p><b>注意</b> ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。</p> <hr/> <p>構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで <i>window-spec</i> の要素を指定できます。ウィンドウの指定方法については、「<a href="#">分析関数</a>」(106 ページ) を参照してください。</p> |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL2008</b> 主要な SQL に含まれない、SQL/foundation の機能です。</li> <li>• <b>Sybase</b> SQL Anywhere 互換。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

## COUNT 関数 [集合]

|    |                                                                             |
|----|-----------------------------------------------------------------------------|
| 機能 | グループ内で、指定されたパラメータに合致するローの数をカウントします。                                         |
| 構文 | <b>COUNT ( *   <i>expression</i>   <b>DISTINCT</b> <i>column-name</i> )</b> |

|        |                                                                                                                                                                                                              |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ  | <p>* 各グループのローの数を返します。</p> <p><b>expression</b> 各グループ内で、<i>expression</i> が NULL 値にならないローの数を返します。</p> <p><b>DISTINCT column-name</b> <i>column-name</i> に含まれる、共通する値を除いた値の数を返します。値が NULL 値であるローはカウントされません。</p> |
| 例      | <p>次の文は、一意な都市を (その都市を値に持つローの数と併せて) 返します。</p> <pre>SELECT city , Count(*) FROM Employees GROUP BY city</pre>                                                                                                  |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> SQL92 互換。</li> <li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li> </ul>                                                                           |
| 参照     | <p><a href="#">「AVG 関数 [集合]」 (131 ページ)</a></p> <p><a href="#">「SUM 関数 [集合]」 (266 ページ)</a></p> <p>『システム管理ガイド 第2巻』の「<a href="#">第2章 OLAP の使用</a>」</p>                                                          |

## CUME\_DIST 関数 [ランク付け]

|    |                                                                                                                                                                                                                                             |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | CUME_DIST 関数は、値の相対位置を行のグループ内で計算し、0 ~ 1 の 10 進値を返します。                                                                                                                                                                                        |
| 構文 | <p><b>CUME_DIST () OVER</b> (<i>window-spec</i>)</p> <p><i>column-spec</i> : 以下の「使用法」を参照してください。</p>                                                                                                                                         |
| 例  | <p>次の例は、カリフォルニア (CA) に住む従業員の給与の累積分布を示す結果セットを返します。</p> <pre>SELECT DepartmentID, Surname, Salary, CUME_DIST() OVER (PARTITION BY DepartmentID ORDER BY Salary DESC) "Rank" FROM Employees WHERE State IN ('CA');</pre> <p>次の結果セットが返されます。</p> |

表 4-18 : CUME\_DIST の結果セット

| DepartmentID | Surname  | Salary    | Rank     |
|--------------|----------|-----------|----------|
| 200          | Savarino | 72300.000 | 0.333333 |
| 200          | Clark    | 45000.000 | 0.666667 |
| 200          | Overbey  | 39300.000 | 1.000000 |

## 使用法

Sybase IQ は、次の式を使用してサイズ N のセット S に含まれる x の値の累積分布を計算します。

$CUME\_DIST(x) = (S \text{ に含まれ、指定された順番で } x \text{ 以前にある値の数}) \div N$

現在、CUME\_DIST 関数では、複合ソート・キーは許可されていません。複合ソート・キーは、その他のランク付け関数で使用できます。

関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。*window-spec* は ORDER BY 句を含む必要がありますが、ROWS および RANGE 句を含むことはできません。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

**注意** DISTINCT はサポートされていません。

## 標準と互換性

- **SQL2008** SQL/OLAP 機能 T612。
- **Sybase** SQL Anywhere 互換。

## DATALENGTH 関数 [システム]

## 機能

式の長さをバイト数で返します。

## 構文

**DATALENGTH** ( *expression* )

## パラメータ

**expression** *expression* には通常、カラム名を指定します。式が文字列定数の場合は、引用符で囲んでください。

## 使用法

表 4-19 は、DATALENGTH 関数の戻り値を示します。

表 4-19 : DATALENGTH の戻り値

| データ型     | DATALENGTH |
|----------|------------|
| SMALLINT | 2          |
| INTEGER  | 4          |
| DOUBLE   | 8          |
| CHAR     | データの長さ     |
| BINARY   | データの長さ     |

例 次の文は、`CompanyName` カラムで最も長い文字列の長さである 35 を返します。

```
SELECT MAX(DATALENGTH(CompanyName))
FROM Customers
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。

参照

[「CHAR\\_LENGTH 関数 \[文字列\]」 \(136 ページ\)](#)

[「COL\\_LENGTH 関数 \[システム\]」 \(138 ページ\)](#)

## DATE 関数 [日付および時刻]

機能 式を日付に変換し、時間、分、または秒を削除します。

構文 **DATE** ( *expression* )

パラメータ **expression** 日付フォーマットに変換される値。通常、文字列を指定します。

例 次の文は、値 1988-11-26 を日付として返します。

```
SELECT DATE('1988-11-26 21:20:53') FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

## DATEADD 関数 [日付および時刻]

機能 指定された日付要素と数値を、日付に加算して作成された日付を返します。

構文 **DATEADD** ( *date-part*, *numeric-expression*, *date-expression* )

パラメータ **date part** 日付に追加する日付要素。

指定可能な日付要素の一覧については、「[日付要素](#)」(114 ページ)を参照してください。

**numeric-expression** 日付に追加する *date parts* の数。*numeric-expression* には、任意の数値型を指定できます。この値は整数にトランケートされます。

**date-expression** 変更される日付。

- 例 次の文を実行すると、値 1995-11-02 00:00:00.000 が返ります。
- ```
SELECT DATEADD( month, 102, '1987/05/02' ) FROM iq_dummy
```
- 使用法 DATEADD は Transact-SQL 互換のデータ操作関数です。
- 標準と互換性
- **SQL92** Transact SQL の拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。

DATECEILING 関数 [日付および時刻]

機能 渡された値を、指定された粒度で、指定された倍数の最大値付近まで大きくして、新しい date、time、または datetime 値を計算します。

構文 **DATECEILING** (*date-part*, *datetime-expression* [*,multiple-expression*])

パラメータ **date part** 日付に追加する日付要素。

次の日付要素は、DATECEILING と互換性がありません。

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

日付要素の一覧については、「[日付要素の値](#)」(114 ページ) を参照してください。

datetime-expression 評価する値を含む date 式、time 式、または datetime 式。

multiple-expression (オプション)。date_part によって指定された単位を計算で何回使用するかを指定する 0 以外の正の整数値式。たとえば、multiple-expression を使用して、データを 10 分間隔で正規化するように指定できます。multiple-expression が 0 に評価された場合、負の数値に評価された場合、または明示的な NULL 定数である場合、Sybase IQ はエラーを生成します。この multiple-expression が NULL に評価された場合、関数の結果は NULL になります。

- 例 この例を実行すると、値 August 13, 2009 10:40.00.000AM (2009 年 8 月 13 日 10:40.00.000AM) が返ります。

```
SELECT DATECEILING( MI, 'August 13, 2009,  
10:32.00.132AM', 10) FROM iq_dummy
```


使用法

この関数は、渡された値を、指定された粒度で最大値付近まで大きくして、新しい `date`、`time`、または `datetime` 値を計算します。オプションの `multiple-expression` パラメータを含めた場合、`date` および `time` が、指定された粒度で、指定された倍数の最大近似値まで大きくなります。

計算された `date` および `time` のデータ型は、`multiple-expression` パラメータのデータ型に一致します。

`multiple-expression` を `millisecond`、`second`、`minute`、または `hour` 日付要素に指定した場合、倍数は、粒度の次の大きい単位の最初から適用されます。

- `millisecond` の倍数は、現在の `second` から開始します。
- `second` の倍数は現在の `minute` から開始します。
- `minute` の倍数は現在の `hour` から開始します。
- `hour` の倍数は現在の `day` から開始します。

たとえば、2 分の倍数を指定した場合、現在の `hour` から始まる 2 分の間隔が適用されます。

`millisecond`、`second`、`minute`、および `hour` の時間要素に対して、指定した日付要素の範囲を均等に分割する `multiple-expression` 値を指定します。

- `millisecond` の場合、有効な `multiple-expression` の値は、1、2、4、5、8、10、20、25、40、50、100、125、200、250、500、1000 です。
- `second` および `minute` の場合、有効な `multiple-expression` の値は、1、2、3、4、5、6、10、12、15、20、30、60 です。
- `hour` の場合、有効な `multiple-expression` の値は、1、2、3、4、6、8、12、24 です。

`multiple-expression` を `day`、`week`、`month`、`quarter`、または `year` の日付要素に指定した場合、間隔は最小の `date` 値 (0000-01-01)、最小の `time` 値 (00:00:00.000000)、または最小の `datetime` 値 (0000-01-01.00:00:00.000000) から開始します。たとえば、10 日の倍数を指定した場合、Sybase IQ では、0000-01-01 から始まる 10 日間隔が計算されます。

`day`、`week`、`month`、`quarter`、または `year` の日付要素の場合、時間粒度の次の大きな単元に均等に分割される倍数を指定する必要はありません。

`week` 日付要素の倍数が丸められる場合、`date` 値は常に Sunday (日曜日) です。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** および **SQL Anywhere** ではサポートされていません。

DATEDIFF 関数 [日付および時刻]

機能 2つの日付のインターバルを返します。

構文 **DATEDIFF** (*date-part*, *date-expression1*, *date-expression2*)

パラメータ **date-part** インターバルを計算する日付要素を指定します。

指定可能な日付要素の一覧については、「[日付要素](#)」(114 ページ)を参照してください。

date-expression1 間隔の起点となる日付。この値を *date-expression2* から引いて、2つの引数の間の日付要素の数を返します。

date-expression2 間隔の終点となる日付。*date-expression1* をこの値から引いて、2つの引数の間の日付要素の数を返します。

例 次の文は、1 を返します。

```
SELECT DATEDIFF( hour, '4:00AM', '5:50AM' ) FROM
iq_dummy
```

次の文は、102 を返します。

```
SELECT DATEDIFF( month, '1987/05/02', '1995/11/15' )
FROM iq_dummy
```

次の文は、0 を返します。

```
SELECT DATEDIFF( day, '00:00', '23:59' ) FROM iq_dummy
```

次の文は、4 を返します。

```
SELECT DATEDIFF( day, '1999/07/19 00:00', '1999/07/23
23:59' ) FROM iq_dummy
```

次の文は、0 を返します。

```
SELECT DATEDIFF( month, '1999/07/19', '1999/07/23' )
FROM iq_dummy
```

次の文は、1 を返します。

```
SELECT DATEDIFF( month, '1999/07/19', '1999/08/23' )
FROM iq_dummy
```

使用法

この関数は、指定した2つの日付の間における日付要素の数を計算します。結果は日付要素の (date2 - date1) に等しい符号付きの整数です。

DATEDIFF 関数の結果が日付要素の整数倍でない場合、結果は丸められずトランケートされます。

day を日付要素として使用すると、DATEDIFF 関数は2つの指定された時刻の間にある深夜の数を返します。このとき、2番目の日付は含みますが、最初の日付は含みません。たとえば、次の文は値5を返します。最初の日付である 2003/08/03 の深夜は結果に含まれていません。2番目の日付の深夜は、指定された時刻は深夜よりも前ですが、含まれています。

```
SELECT DATEDIFF( day, '2003/08/03 14:00', '2003/08/08  
14:00' ) FROM iq_dummy
```

日付要素に month を使用すると、DATEDIFF 関数は2つの日付の間にある、月の初日の数を返します。このとき、2番目の日付は含みますが、最初の日付は含みません。たとえば、次の文はどちらも9を返します。

```
SELECT DATEDIFF( month, '2003/02/01', '2003/11/15' )  
FROM iq_dummy;  
SELECT DATEDIFF( month, '2003/02/01', '2003/11/01' )  
FROM iq_dummy;
```

最初の日付、2003/02/01 は月の初日ですが、どちらのクエリの結果にも含まれていません。2つ目のクエリの2番目の日付、2003/11/01 も月の初日であり、これは結果に含まれています。

week を日付要素として使用すると、DATEDIFF 関数は2つの日付の間にある日曜日の数を返します。このとき、2番目の日付は含みますが、最初の日付は含みません。たとえば、2003/08月の日曜日は03、10、17、24、31日ですが、次のクエリを実行すると値4が返ります。

```
SELECT DATEDIFF( week, '2003/08/03', '2003/08/31' )  
FROM iq_dummy;
```

第1日曜日 (2003/08/03) は結果に含まれていません。

小さい時間単位を使用した場合、次の値でオーバーフローが発生します。

- **ミリ秒** 24日
- **秒** 68年
- **分** 4083年
- **その他** オーバフロー制限なし

これらの制限を超過すると、関数からオーバーフロー・エラーが返されます。

標準と互換性

- **SQL92** Transact SQL の拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

DATEFLOOR 関数 [日付および時刻]

機能

渡された値を、指定された粒度で、指定された倍数の最低値付近まで小さくして、新しい date、time、または datetime 値を計算します。

構文

DATEFLOOR (*date-part*, *datetime-expression* [*,multiple-expression*])

パラメータ

date part 日付に追加する日付要素。

次の日付要素は、DATEFLOOR と互換性がありません。

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

日付要素の一覧については、「[日付要素の値](#)」(114 ページ) を参照してください。

datetime-expression 評価する値を含む date 式、time 式、または datetime 式。

multiple-expression (オプション)。date_part によって指定された単位を計算で何回使用するかを指定する 0 以外の正の整数値式。たとえば、multiple-expression を使用して、データを 10 分間隔で正規化するように指定できます。multiple-expression が 0 に評価された場合、負の数値に評価された場合、または明示的な NULL 定数である場合、IQ はエラーを生成します。multiple-expression が NULL に評価された場合、関数の結果は NULL になります。

例

次の文を実行すると、値 August 13, 2009 10:35:00.000AM (2009 年 8 月 13 日 10:35:00.000AM) が返ります。

```
SELECT DATEFLOOR( MINUTE, 'August 13, 2009  
10:35.22.123AM') FROM iq_dummy
```

使用法

この関数は、渡された値を、指定された粒度で最小値付近まで小さくして、新しい `date`、`time`、または `datetime` 値を計算します。オプションの `multiple-expression` パラメータを含めた場合、`date` および `time` が、指定された粒度で、指定された倍数の最小近似値まで小さくなります。

計算された `date` および `time` のデータ型は、`multiple-expression` パラメータのデータ型に一致します。

`multiple-expression` を `millisecond`、`second`、`minute`、または `hour` 日付要素に指定した場合、倍数は、粒度の次の大きい単位の最初から適用されます。

- `millisecond` の倍数は、現在の `second` から開始します。
- `second` の倍数は現在の `minute` から開始します。
- `minute` の倍数は現在の `hour` から開始します。
- `hour` の倍数は現在の `day` から開始します。

たとえば、2 分の倍数を指定した場合、現在の `hour` から始まる 2 分の間隔が適用されます。

`millisecond`、`second`、`minute`、および `hour` の時間要素に対して、指定した日付要素の範囲を均等に分割する `multiple-expression` 値を指定します。

- `millisecond` の場合、有効な `multiple-expression` の値は、1、2、4、5、8、10、20、25、40、50、100、125、200、250、500、1000 です。
- `second` および `minute` の場合、有効な `multiple-expression` の値は、1、2、3、4、5、6、10、12、15、20、30、60 です。
- `hour` の場合、有効な `multiple-expression` の値は、1、2、3、4、6、8、12、24 です。

`multiple-expression` を `day`、`week`、`month`、`quarter`、または `year` の日付要素に指定した場合、間隔は最小の `date` 値 (0000-01-01)、最小の `time` 値 (00:00:00.000000)、または最小の `datetime` 値 (0000-01-01.00:00:00.000000) から開始します。たとえば、10 日の倍数を指定した場合、Sybase IQ では、0000-01-01 から始まる 10 日間隔が計算されます。

`day`、`week`、`month`、`quarter`、または `year` の日付要素の場合、時間粒度の次の大きな単元に均等に分割される倍数を指定する必要はありません。

`week` 日付要素の倍数が丸められる場合、`date` 値は常に Sunday (日曜日) です。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase Adaptive Server Enterprise** および **SQL Anywhere** ではサポートされません。

DATEFORMAT 関数 [日付および時刻]

- 機能** 日付式を指定されたフォーマットで表す文字列を返します。
- 構文** **DATEFORMAT** (*datetime-expression*, *string-expression*)
- パラメータ** **datetime-expression** 変換される日時。日付、時刻、タイムスタンプ、または文字列を指定する必要があります。
- string-expression** 変換後の日付のフォーマット。

例 次の文は、「Jan 01, 1989」のような文字列の値を返します。

```
SELECT DATEFORMAT( startdate, 'Mmm dd, yyyy' ) from
Employees;
```

次の文は、文字列「Feb 19, 1987」を返します。

```
SELECT DATEFORMAT( CAST ( '1987/02/19' AS DATE ), 'Mmm
Dd, yyyy' ) FROM iq_dummy
```

使用法 変換する *datetime-expression* は、日付、時刻、またはタイムスタンプのデータ型であることが必要ですが、**CHAR** や **VARCHAR** の文字列を使用することもできます。日付を文字列で指定すると、**Sybase IQ** によって文字列が自動的に日付、時刻、タイムスタンプのデータ型に変換されます。したがって、上記の例のように、キャストを明示的に行う必要はありません。

string-expression には、許可されている日付フォーマットをすべて使用できます。日付フォーマットの文字列に、マルチバイト文字を格納することはできません。日付、時刻、日時のフォーマットの文字列に格納できるのは、シングルバイト文字だけです。これはデータベースの照合順序が、**932JPN** のようにマルチバイトの照合順序である場合も同じです。

次の '?' がマルチバイト文字を表す場合、このクエリはエラーになります。

```
SELECT DATEFORMAT ( startdate, 'yy?') FROM Employees;
```

代わりに、連結演算子を使用して、マルチバイト文字を日付フォーマットの文字列の外に移動します。

```
SELECT DATEFORMAT (startdate, 'yy') + '?' FROM
Employees;
```

| | |
|--------|----------------------------------------------------------------------------------------------------------------------|
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase SQL Anywhere 互換。 |
| 参照 | 『リファレンス：文とオプション』の「 DATE_FORMAT オプション 」 |

DATENAME 関数 [日付および時刻]

| | |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定した日時の特要素の名前を(月ならば“June”のように)文字列で返します。 |
| 構文 | DATENAME (<i>date-part</i> , <i>date-expression</i>) |
| パラメータ | <p>date-part 名前を返す日付要素。</p> <p>指定可能な日付要素の一覧については、「日付要素」(114 ページ)を参照してください。</p> <p>date-expression 日付要素の名前を返す日付。必要な <i>date-part</i> を含めて指定します。</p> |
| 例 | 次の文を実行すると、値 May が返ります。 <pre>SELECT datename(month , '1987/05/02') FROM iq_dummy</pre> |
| 使用法 | DATENAME は、結果が 23 日などの数値であっても、文字列を返します。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 Transact SQL の拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |

DATEPART 関数 [日付および時刻]

| | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 日時の値の指定された要素に対する整数値を返します。 |
| 構文 | DATEPART (<i>date-part</i> , <i>date-expression</i>) |
| パラメータ | <p>date-part 返される日付要素。</p> <p>指定可能な日付要素の一覧については、「日付要素」(114 ページ)を参照してください。</p> <p>date-expression 要素を返す日付。<i>date-part</i> のフィールドを含める必要があります。</p> |
| 例 | 次の文は、値 5 を返します。 <pre>SELECT DATEPART(month , '1987/05/02') FROM iq_dummy</pre> |

- 使用法 DATE、TIME、および DTTM インデックスは、一部の日付要素 (Calyearofweek、Calweekofyear、Caldayofweek、Dayofyear、Millisecond) をサポートしません。
- 標準と互換性
- **SQL92** Transact SQL の拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。

DATEROUND 関数 [日付および時刻]

機能 渡された値を、指定された粒度で、指定された値に最も近い倍数まで丸めて、新しい date、time、または datetime 値を計算します。

構文 **DATEROUND** (*date-part*, *datetime-expression* [*,multiple-expression*])

パラメータ **date part** 日付に追加する日付要素。

次の日付要素は、DATEROUND と互換性がありません。

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

日付要素の一覧については、「[日付要素の値](#)」(114 ページ) を参照してください。

datetime-expression 評価する値を含む date 式、time 式、または datetime 式。

multiple-expression (オプション)。date_part によって指定された単位を計算で何回使用するかを指定する 0 以外の正の整数値式。たとえば、multiple-expression を使用して、データを 10 分間隔で正規化するように指定できます。multiple-expression が 0 に評価された場合、負の数値に評価された場合、または明示的な NULL 定数である場合、IQ はエラーを生成します。multiple-expression が NULL に評価された場合、関数の結果は NULL になります。

例 次の文を実行すると、値 August 13, 2009 10:30.000AM (2009 年 8 月 13 日 10:30.000AM) が返ります。

```
SELECT DATEROUND( MI, 'August 13, 2009 10:33:00.123AM',
10) FROM iq_dummy
```


使用法

この関数は、渡された値を、指定された粒度で近似値まで丸めて、新しい `date`、`time`、または `datetime` 値を計算します。オプションの `multiple-expression` パラメータを含めた場合、`date` および `time` が、指定された粒度で、指定された倍数の近似値まで丸められます。

計算された `date` および `time` のデータ型は、`multiple-expression` パラメータのデータ型に一致します。

`multiple-expression` を `millisecond`、`second`、`minute`、または `hour` 日付要素に指定した場合、倍数は、粒度の次の大きい単位の最初から適用されます。

- `millisecond` の倍数は、現在の `second` から開始します。
- `second` の倍数は現在の `minute` から開始します。
- `minute` の倍数は現在の `hour` から開始します。
- `hour` の倍数は現在の `day` から開始します。

たとえば、2 分の倍数を指定した場合、現在の `hour` から始まる 2 分の間隔が適用されます。

`millisecond`、`second`、`minute`、および `hour` の時間要素に対して、指定した日付要素の範囲を均等に分割する `multiple-expression` 値を指定します。

- `millisecond` の場合、有効な `multiple-expression` の値は、1、2、4、5、8、10、20、25、40、50、100、125、200、250、500、1000 です。
- `second` および `minute` の場合、有効な `multiple-expression` の値は、1、2、3、4、5、6、10、12、15、20、30、60 です。
- `hour` の場合、有効な `multiple-expression` の値は、1、2、3、4、6、8、12、24 です。

`multiple-expression` を `day`、`week`、`month`、`quarter`、または `year` の日付要素に指定した場合、間隔は最小の `date` 値 (0000-01-01)、最小の `time` 値 (00:00:00.000000)、または最小の `datetime` 値 (0000-01-01.00:00:00.000000) から開始します。たとえば、10 日の倍数を指定した場合、Sybase IQ では、0000-01-01 から始まる 10 日間隔が計算されます。

`day`、`week`、`month`、`quarter`、または `year` の日付要素の場合、時間粒度の次の大きな単元に均等に分割される倍数を指定する必要はありません。

`week` 日付要素の倍数が丸められる場合、`date` 値は常に Sunday (日曜日) です。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase Adaptive Server Enterprise** および **SQL Anywhere** ではサポートされません。

DATETIME 関数 [日付および時刻]

機能 式をタイムスタンプに変換します。

構文 **DATETIME** (*expression*)

パラメータ **expression** 変換される *expression*。通常、文字列を指定します。変換エラーがあればレポートします。

例 次の文を実行すると、値 1998-09-09 12:12:12.000 のタイムスタンプが返ります。

```
SELECT DATETIME( '1998-09-09 12:12:12.000' ) FROM  
iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase Adaptive Server Enterprise** ではサポートされていません。

DAY 関数 [日付および時刻]

機能 指定された日付の日に対応する 1 から 31 までの整数を返します。

構文 **DAY** (*date-expression*)

パラメータ **date-expression** 日付。

例 次の文は、値 12 を返します。

```
SELECT DAY( '2001-09-12' ) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase Adaptive Server Enterprise** 互換。

DAYNAME 関数 [日付および時刻]

機能 指定の日付について、曜日の名前を返します。

構文 **DAYNAME** (*date-expression*)

パラメータ **date-expression** 日付。

例 次の文を実行すると、値 `Saturday` が返ります。

```
SELECT DAYNAME ( '1987/05/02' ) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

DAYS 関数 [日付および時刻]

機能 任意の開始日からの日数を返すか、指定された 2 つの日付の間の日数を返すか、または *integer-expression* で指定された日数を、指定の日付に追加します。

構文

```
DAYS ( datetime-expression )
| ( datetime-expression, datetime-expression )
| ( datetime-expression, integer-expression )
```

パラメータ **datetime-expression** 日時。

integer-expression *datetime-expression* に追加する日数。*integer-expression* が負の場合、指定された日数が日時から減算されます。整数式を指定する場合、*datetime-expression* は日付として明示的にキャストする必要があります。

データ型のキャストについては、「[CAST 関数 \[データ型変換\]](#)」(134 ページ)を参照してください。

DAYS 関数では、時、分、秒は無視されます。

例 次の文を実行すると、整数値 `729948` が返ります。

```
SELECT DAYS ( '1998-07-13 06:07:12' ) FROM iq_dummy
```

次の文を実行すると、整数値 `-366` (2 つの日付の差) が返ります。

```
SELECT DAYS ( '1998-07-13 06:07:12',
'1997-07-12 10:07:12' ) FROM iq_dummy
```

次の文を実行すると、値 `1999-07-14` が返ります。

```
SELECT DAYS ( CAST('1998-07-13' AS DATE ), 366 )
FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

DB_ID 関数 [システム]

機能 データベース ID 番号を返します。

構文 **DB_ID** ([*database-name*])

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **database-name** データベース名を含む文字列式。 *database-name* が文字列定数の場合は、引用符で囲んでください。 *database-name* が指定されなければ、現在のデータベースの ID 番号が返されます。

例 実行中のデータベースが `iqdemo` だけの場合、次の文を実行すると値 0 が返されます。

```
SELECT DB_ID( 'iqdemo' ) FROM iq_dummy
```

次の文を、唯一動作中のデータベースに対して実行すると、値 0 が返されます。

```
SELECT DB_ID() FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。

参照 [「DB_NAME 関数 \[システム\]」 \(162 ページ\)](#)

[「OBJECT_ID 関数 \[システム\]」 \(216 ページ\)](#)

DB_NAME 関数 [システム]

機能 データベース名を返します。

構文 **DB_NAME** ([*database-id*])

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

| | |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | database-id データベースの ID。 <i>database-id</i> には数値式を指定してください。 |
| 例 | 次の文をサンプル・データベースに対して実行すると、データベース名 <i>iqdemo</i> が返されます。 <pre>SELECT DB_NAME(0) FROM iq_dummy</pre> |
| 使用法 | <i>database-id</i> を指定しない場合、現在のデータベースの名前が返されます。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。 |
| 参照 | 「COL_NAME 関数 [システム]」 (139 ページ) 「DB_ID 関数 [システム]」 (162 ページ) 「OBJECT_NAME 関数 [システム]」 (216 ページ) |

DB_PROPERTY 関数 [システム]

| | |
|----|----------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定されたプロパティの値を返します。 |
| 構文 | <pre>DB_PROPERTY ({ <i>property-id</i> <i>property-name</i> } [, { <i>database-id</i> <i>database-name</i> }])</pre> |

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されません。『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

| | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | <p>property-id データベースのプロパティ ID。</p> <p>property-name データベースのプロパティの名前。</p> <p>database-id データベースの ID 番号。DB_ID で取得できます。通常は、データベース名が使用されます。</p> <p>database-name データベースの名前。DB_NAME で取得できます。</p> |
| 例 | 次の文は、現在のデータベースのページ・サイズをバイト単位で返します。 <pre>SELECT DB_PROPERTY('PAGESIZE') FROM iq_dummy</pre> |

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 使用法 | 文字列を返します。2番目の引数を省略すると、現在のデータベースが使用されます。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | <p>「各データベースで使用できるプロパティ」(122 ページ)</p> <p>「DB_ID 関数 [システム]」(162 ページ)</p> <p>「DB_NAME 関数 [システム]」(162 ページ)</p> |

DEGREES 関数 [数値]

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 数値をラジアンから度に変換します。 |
| 構文 | DEGREES (<i>numeric-expression</i>) |
| パラメータ | numeric-expression 角度 (ラジアン)。 |
| 例 | 次の文は、値 29.793805 を返します。 <pre>SELECT DEGREES(0.52) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |

DENSE_RANK 関数 [統計]

| 機能 | グループ内の項目をランク付けします。 | | | | | | | | | | | | | | | |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------------|------------|--------------|--------|---|--------------|--------|---|--------------|--------|---|--------------|--------|---|
| 構文 | DENSE_RANK () OVER (ORDER BY <i>expression</i> [ASC DESC]) | | | | | | | | | | | | | | | |
| パラメータ | expression ソートを指定します。カラムの参照、集合関数、またはこれらの項目を起動する式など、有効な式を何でも指定できます。 | | | | | | | | | | | | | | | |
| 例 | 次の文は、DENSE_RANK 関数の使い方を示します。 <pre>SELECT s_suppkey, SUM(s_acctBal) as sum_acctBal, DENSE_RANK (OVER (ORDER BY SUM(s_acctBal) DESC) AS rank_dense FROM supplier GROUP BY s_suppkey;</pre> <table> <thead> <tr> <th>s_suppkey</th> <th>sum_acctBal</th> <th>rank_dense</th> </tr> </thead> <tbody> <tr> <td>supplier#011</td> <td>200000</td> <td>1</td> </tr> <tr> <td>supplier#002</td> <td>200000</td> <td>1</td> </tr> <tr> <td>supplier#013</td> <td>123000</td> <td>2</td> </tr> <tr> <td>supplier#004</td> <td>110000</td> <td>3</td> </tr> </tbody> </table> | s_suppkey | sum_acctBal | rank_dense | supplier#011 | 200000 | 1 | supplier#002 | 200000 | 1 | supplier#013 | 123000 | 2 | supplier#004 | 110000 | 3 |
| s_suppkey | sum_acctBal | rank_dense | | | | | | | | | | | | | | |
| supplier#011 | 200000 | 1 | | | | | | | | | | | | | | |
| supplier#002 | 200000 | 1 | | | | | | | | | | | | | | |
| supplier#013 | 123000 | 2 | | | | | | | | | | | | | | |
| supplier#004 | 110000 | 3 | | | | | | | | | | | | | | |

| | | |
|--------------|--------|---|
| supplier#035 | 110000 | 3 |
| supplier#006 | 50000 | 4 |
| supplier#021 | 10000 | 5 |

使用法

DENSE_RANK はランク付け統計関数です。ロー R のランクは、R 以前のローの数のうち、OVER 句で指定されたグループ間で同等なロー、または結果セット全体で同等なローを引いた数になります。DENSE_RANK と RANK では、順位が同じである場合に、DENSE_RANK はランク順に隔たりを置かない点が異なります。RANK では隔たりが置かれます。

DENSE_RANK には、OVER (ORDER BY) 句を指定する必要があります。ORDER BY 句は、ランク付けを実行するパラメータ、および各グループでローをソートする順序を指定します。この ORDER BY 句は、OVER 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。ランク付けクエリ内の集合関数に DISTINCT を指定することはできません。

OVER 句は、関数がクエリの結果セットを操作することを示します。結果セットとは、FROM、WHERE、GROUP BY、HAVING 句がすべて評価された後に返されるローを指します。OVER 句には、ランク付け統計関数の計算の対象となるローのデータ・セットを定義します。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

DENSE_RANK が使用できるのは、SELECT 文、INSERT 文、または SELECT 文の ORDER BY 句にある select リストのみです。DENSE_RANK は、ビューまたは union に含めることができます。DENSE_RANK 関数は、サブクエリ、HAVING 句、および UPDATE または DELETE 文の select リストでは使用できません。1 つのクエリで使用可能なランク付け統計関数は、1 つだけです。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** および **SQL Anywhere** ではサポートされていません。

参照

[「分析関数」\(106 ページ\)](#)

[「RANK 関数 \[統計\]」\(230 ページ\)](#)

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

DIFFERENCE 関数 [文字列]**機能**

2 つの文字列を比較し、それらの間の類似性を評価して 0 ～ 4 の値を返します。最高点は 4 です。

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | DIFFERENCE (<i>string-expression1</i> , <i>string-expression2</i>) |
| パラメータ | string-expression1 比較を行う最初の文字列。 string-expression2 比較を行う 2 つ目の文字列。 |
| 例 | 次の文は、値 4 を返します。 <pre>SELECT DIFFERENCE('Smith', 'Smith') FROM iq_dummy</pre> 次の文は、値 4 を返します。 <pre>SELECT DIFFERENCE('Smith', 'Smyth') FROM iq_dummy</pre> 次の文は、値 3 を返します。 <pre>SELECT DIFFERENCE('Smith', 'Sweeney') FROM iq_dummy</pre> 次の文は、値 2 を返します。 <pre>SELECT DIFFERENCE('Smith', 'Jones') FROM iq_dummy</pre> 次の文は、値 1 を返します。 <pre>SELECT DIFFERENCE('Smith', 'Rubin') FROM iq_dummy</pre> 次の文は、値 0 を返します。 <pre>SELECT DIFFERENCE('Smith', 'Wilkins') FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |
| 参照 | 「SOUNDEX 関数 [文字列]」 (256 ページ) |

DOW 関数 [日付および時刻]

| | |
|-------|----------------------------------------------------------------------------------------------------------|
| 機能 | Sunday = 1、Monday = 2 のような、指定された日付の曜日を表す 1 ~ 7 までの数字を返します。 |
| 構文 | DOW (<i>date-expression</i>) |
| パラメータ | date-expression 日付。 |
| 例 | 次の文は、値 5 を返します。 <pre>SELECT DOW('1998-07-09') FROM iq_dummy</pre> |
| 使用法 | 月曜日または別の日を週の最初の曜日にする方法については、『リファレンス: 文とオプション』の「 DATE_FIRST_DAY_OF_WEEK オプション 」を参照してください。 |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

ERRORMSG 関数 [その他]

機能 現在のエラー、または指定された SQLSTATE または SQLCODE 値のエラー・メッセージを提供します。

構文 `ERRORMSG ([sqlstate | sqlcode])`

sqlstate: string

sqlcode: integer

パラメータ **sqlstate** エラー・メッセージが返される SQLSTATE 値。

sqlcode エラー・メッセージが返される SQLCODE 値。

例 次の文を実行すると、SQLCODE -813 のエラー・メッセージを返します。

```
select errmsg( -813 )
```

戻り値 エラー・メッセージを含む文字列。引数を指定しない場合、現在の状態のエラー・メッセージが返されます。置き換え (テーブル名やカラム名など) が行われます。

引数を指定すると、指定した SQLSTATE または SQLCODE のエラー・メッセージが置き換えなしで返されます。テーブル名およびカラム名はプレースホルダ ("??") として返されます。

ERRORMSG 関数は、SQL Anywhere および Sybase IQ のエラー・メッセージを返します。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **SQL99** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照 『システム管理ガイド 第 2 巻』

EVENT_CONDITION 関数 [システム]

機能 イベント・ハンドラをトリガする条件を指定します。

構文

EVENT_CONDITION (*condition-name*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されま
す。『パフォーマンス&チューニング・ガイド』の「第3章 クエリと
削除の最適化」の「SQL Anywhere による処理を引き起こす条件」を
参照してください。

パラメータ

condition-name イベントをトリガする条件。指定可能な値はデー
タベースにあらかじめセットされており、大文字小文字を区別して指定
します。条件はそれぞれ、特定の種類のイベントに対してのみ有効
です。表 4-20 は条件と、それを指定可能なイベントを示します。

表 4-20 : 各イベントに対して有効な条件

| 条件名 | 単位 | イベント | コメント |
|-----------------|-------|---------------------|--------------------------------------------------------------------------------------|
| DBFreePercent | N/A | DBDiskSpace | DBDiskSpace は、IQ ス トアではなく、シス テム・データベース・ ファイル (db ファイ ル) の空き領域を表示 します。 |
| DBFreeSpace | メガバイト | DBDiskSpace | |
| DBSize | メガバイト | GrowDB | ハンドラが前回実行 されてからの時間。 |
| ErrorNumber | N/A | RAISERROR | |
| IdleTime | 秒 | ServerIdle | |
| Interval | 秒 | すべて | |
| LogFreePercent | N/A | LogDiskSpace | |
| LogFreeSpace | メガバイト | LogDiskSpace | |
| LogSize | メガバイト | GrowLog | |
| RemainingValues | 整数値 | GlobalAutoincrement | |
| TempFreePercent | N/A | TempDiskSpace | |
| TempFreeSpace | メガバイト | TempDiskSpace | |
| TempSize | メガバイト | GrowTemp | |

| | |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 例 | 次のイベント定義には、EVENT_CONDITION 関数が使用されています。 <pre>create event LogNotifier type LogDiskSpace where event_condition('LogFreePercent') < 50 handler begin message 'LogNotifier message' end</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | 『リファレンス：文とオプション』の「 CREATE EVENT 文 」 |

EVENT_CONDITION_NAME 関数 [システム]

| | |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | EVENT_CONDITION に指定可能なパラメータのリストを取得できます。 |
| 構文 | EVENT_CONDITION_NAME (<i>integer</i>) |
| | <hr/> <p>注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されません。『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「SQL Anywhere による処理を引き起こす条件」を参照してください。</p> <hr/> |
| パラメータ | integer 0以上の整数である必要があります。 |
| 使用法 | EVENT_CONDITION_NAME を使用すると、関数が NULL を返すまでループ処理を指定した回数だけ実行し、EVENT_CONDITION のすべての引数のリストを取得することができます。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | 『リファレンス：文とオプション』の「 CREATE EVENT 文 」 |

EVENT_PARAMETER 関数 [システム]

| | |
|----|------------------------------------------------|
| 機能 | イベント・ハンドラのコンテキスト情報を取得できます。 |
| 構文 | EVENT_PARAMETER (<i>context-name</i>) |

```

context-name:
'ConnectionID'
|'User'
|'EventName'
|'Executions'
|'IQDBMainSpaceName'
|'NumActive'
|'TableName'
condition-name

```

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ

context-name あらかじめセットされた文字列のいずれか。文字列は大文字と小文字を区別します。また、以下の情報を保持します。

- **ConnectionId** 接続 ID。次の文で返されます。

```
connection_property( 'id' )
```

- **User** イベントをトリガさせたユーザのユーザ ID。
- **EventName** トリガされたイベントの名前。
- **Executions** イベント・ハンドラが実行された回数。
- **NumActive** イベント・ハンドラのアクティブなインスタンスの数。これは、イベント・ハンドラの数制限して、一度に1つのインスタンスのみが実行されるようにする場合に便利です。
- **TableName** テーブルの名前。RemainingValues で使用します。

このほか、EVENT_CONDITION 関数に指定可能な *condition-name* 引数であればどれでも、EVENT_PARAMETER 関数で使用することができます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

[「EVENT_CONDITION 関数 \[システム\]」 \(167 ページ\)](#)

『リファレンス：文とオプション』の「[CREATE EVENT 文](#)」

EXP 関数 [数値]

機能

指数関数 (e の数値乗) を返します。

構文

```
EXP ( numeric-expression )
```

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | numeric-expression 指数。 |
| 例 | 次の文は、値 3269017.3724721107 を返します。 <pre>SELECT EXP(15) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |

EXP_WEIGHTED_AVG 関数 [集合]

| | |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指数加重移動平均を計算します。加重は、平均を構成する各数量の相対的な重要性を決定します。 |
| 構文 | <pre>EXP_WEIGHTED_AVG (<i>expression</i>, <i>period-expression</i>) OVER (<i>window-spec</i>)</pre> <p><i>column-spec</i> : 以下の「使用法」を参照してください。</p> |
| パラメータ | <p>expression 計算する加重値の数値式。</p> <p>period-expression 平均を計算する間隔を指定する数値式。</p> |
| 使用法 | <p>WEIGHTED_AVG 関数と同様に、EXP_WEIGHTED_AVG の加重は時間経過とともに減少します。しかし、WEIGHTED_AVG の加重は等差階級的に減少しますが、EXP_WEIGHTED_AVG の加重は指数関数的に減少します。指数関数的な加重は、最新の値に加重を適用し、加重を適用しながらも、古い値の加重を減少します。</p> <p>Sybase IQ は、次の式を使用して指数加重移動平均を計算します。</p> <p>S*C+(1-S)*PEMA</p> <p>上の計算では、現在の終値 (C) を補整定数 (S) の積に対して、1 から補整定数を引いた値に前日の指数加重移動平均値 (PEMA) を乗算した結果を追加することによって補整係数を適用します。</p> <p>Sybase IQ は、OVER 句によって指定された期間全体にわたる指数加重移動平均を計算します。<i>period-expression</i> は、指数加重平均の移動範囲を指定します。</p> |

関数構文 (インライン) または **SELECT** 文の **WINDOW** 句のいずれかで *window-spec* の要素を指定できます。 *window-spec* は **ORDER BY** 文を含む必要がありますが、フレーム指定を含むことはできません。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

注意 **ROLLUP** と **CUBE** は、**GROUP BY** 句でサポートされていません。**DISTINCT** はサポートされていません。

例 次の例は、フロリダの従業員の給与の指数加重移動平均、および平均のほとんどの加重に関する現在雇用されている従業員の給与を返します。加重では3つのローが使用されます。

```
SELECT DepartmentID, Surname, Salary,
       EXP_WEIGHTED_AVG(Salary, 3) OVER (ORDER BY
       YEAR(StartDate) DESC) as "W_AVG"
FROM Employees
WHERE State IN ('FL') ORDER BY StartDate DESC
```

次の結果セットが返されます。

表 4-21 : EXP_WEIGHTED_AVG 結果セット

| DepartmentID | Surname | Salary | W_AVG |
|--------------|----------|-----------|--------------|
| 400 | Evans | 68940.000 | 34470.000000 |
| 300 | Litton | 58930.000 | 46700.000000 |
| 200 | Sterling | 64900.000 | 55800.000000 |
| 200 | Kelly | 87500.000 | 71650.000000 |
| 400 | Charlton | 28300.000 | 49975.000000 |
| 100 | Lull | 87900.000 | 68937.500000 |
| 100 | Gowda | 59840.000 | 60621.875000 |
| 400 | Francis | 53870.000 | 61403.750000 |

標準と互換性

- **SQL2008** ベンダの拡張機能。

FIRST_VALUE 関数 [平均]

機能

一連の値の最初の値を返します。

構文

```
FIRST_VALUE (expression [IGNORE NULLS | RESPECT NULLS])
OVER (window-spec)
```

パラメータ

expression 順序付けられたセットの最初の値を決定する式。

使用法

FIRST_VALUE は、(通常は順序付けられた) 値のセット内の最初の値を返します。セットの最初の値が NULL の場合、IGNORE NULLS が指定されていない場合は NULL が返されます。IGNORE NULLS を指定した場合、FIRST_VALUE は、セット内で NULL でない最初の値を返します。すべての値が NULL の場合は NULL が返されます。

戻り値のデータ型は、入力値のデータ型と同じです。

expression に FIRST_VALUE およびその他の分析関数を使用することはできません。したがって、分析関数をネストすることはできませんが、その他の組み込み関数式を **expression** に使用できます。

window-spec に ORDER BY が含まれない場合、結果は任意です。*window-spec* がない場合、結果は任意です。

関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

注意 DISTINCT はサポートされていません。

例

次の例は、各従業員の給与と同じ部署で最も最近雇用された従業員の給与の関係をパーセンテージとして返します。

```
SELECT DepartmentID, EmployeeID,
       100 * Salary / ( FIRST_VALUE( Salary ) OVER (
         PARTITION BY DepartmentID ORDER BY Year(StartDate) DESC
       ) )
AS percentage
FROM Employees order by DepartmentID DESC;
```

次の結果セットが返されます。

表 4-22 : FIRST_VALUE 結果セット

| DepartmentID | EmployeeID | Percentage |
|--------------|------------|------------------------------------|
| 500 | 1658 | 100.000000000000000000000000000000 |
| 500 | 1570 | 138.842709713689113761394 |
| 500 | 1615 | 110.428462434244870095972 |
| 500 | 1013 | 109.585190539292454724330 |
| 500 | 750 | 137.734409508894510701521 |
| 500 | 921 | 167.449704854836766654619 |
| 500 | 868 | 113.239368750752921334778 |
| 500 | 703 | 222.867927558928643135365 |

| DepartmentID | EmployeeID | Percentage |
|--------------|------------|----------------------------|
| 500 | 191 | 119.664297474199895594908 |
| 400 | 1684 | 100.0000000000000000000000 |
| 400 | 1740 | 76.128652163477274215016 |
| 400 | 1751 | 76.353400685155687446813 |
| 400 | 1607 | 133.758100765890593292456 |
| 400 | 1507 | 77.996465120338650199655 |
| 400 | 1576 | 150.428767810774836893669 |

この例では、従業員 1658 は部署 500 の最初のローで、その部門で最も最近雇用されたこと、および 100% のパーセンテージを受け取っていることを示しています。部署 500 の残りの従業員のパーセンテージは、従業員 1658 のパーセンテージに相対的に計算されます。たとえば、従業員 1570 は、従業員 1658 の給与の 139% を受け取っています。

標準と互換性

- **SQL2008** SQL/OLAP 機能 T612。
- **Sybase** SQL Anywhere 互換。

FLOOR 関数 [数値]

機能

数値の下限值 (その値以下の最も大きい整数) を返します。

構文

FLOOR (*numeric-expression*)

パラメータ

numeric-expression 数値。通常は float 型です。

例

次の文は、値 123.00 を返します。

```
SELECT FLOOR ( 123 ) FROM iq_dummy
```

次の文は、値 123 を返します。

```
SELECT FLOOR ( 123.45 ) FROM iq_dummy
```

次の文は、値 -124.00 を返します。

```
SELECT FLOOR ( -123.45 ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

参照

[「CEILING 関数 \[数値\]」 \(135 ページ\)](#)

GETDATE 関数 [日付および時刻]

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 現在の日付と時刻を返します。 |
| 構文 | GETDATE () |
| 例 | 次の文は、システムの日付および時刻を返します。 <pre>SELECT GETDATE() FROM iq_dummy</pre> |
| 使用法 | GETDATE は Transact-SQL 互換のデータ操作関数です。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |

GRAPHICAL_PLAN 関数 [文字列]

| | |
|----|--------------------------------------------------------|
| 機能 | グラフィカルなクエリ・プランを Interactive SQL に XML フォーマットの文字列で返します。 |
| 構文 | GRAPHICAL_PLAN (<i>string-expression</i>) |

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

| | |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | <p>string-expression プランを生成する SQL 文。一般的に、<i>string-expression</i> は SELECT 文ですが、UPDATE、DELETE、INSERT SELECT、または SELECT INTO 文を指定することもできます。</p> <p>GRAPHICAL_PLAN に引数を指定しない場合、クエリ・プランはキャッシュから返されます。キャッシュにクエリ・プランがない場合は、次のメッセージが表示されます。</p> <p style="padding-left: 40px;">使用できるプランがありません。</p> <p>GRAPHICAL_PLAN 関数の動作は、データベース・オプション QUERY_PLAN_TEXT_ACCESS と QUERY_PLAN_TEXT_CACHING で制御されます。QUERY_PLAN_TEXT_ACCESS が OFF (デフォルト) の場合、次のメッセージが表示されます。</p> <p style="padding-left: 40px;">使用できるプランがありません。The database option QUERY_PLAN_TEXT_ACCESS is OFF</p> |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

ユーザがプランにアクセスする必要がある場合、DBA はそのユーザに対してオプション `QUERY_PLAN_TEXT_ACCESS` を `ON` に設定する必要があります。

`QUERY_PLAN_TEXT_ACCESS` が `ON` であり、しかもサーバに保持されているキャッシュに文字列式のクエリ・プランがある場合は、キャッシュからクエリ・プランが返されます。

クエリ・プランがキャッシュになく、しかもユーザがクライアント上でプランを表示する許可を得ている場合、オプティマイザの見積もりがあるクエリ・プランが生成され (`NOEXEC` オプションが `ON` のクエリ・プラン)、`dbisql` クライアント・プラン・ウィンドウに表示されます。

注意 `SELECT`、`AUPDATE`、`ADELETE`、`AINsert SELECT`、`ASELECT INTO` の各クエリに対しては、Sybase IQ は `NOEXEC` プランの生成をサポートしていません。

まだ実行されたことがないクエリ・プランを要求しても、そのクエリ・プランはキャッシュにないので、代わりにオプティマイザの見積もりがあるクエリ・プランが返されます。ただし、これには `QUERY_PLAN_AFTER_RUN` 統計はありません。

ストアド・プロシージャのクエリ・プランには、`GRAPHICAL_PLAN` 関数を使用してアクセスできません。

`IQ` クエリのためにオープンするカーソルのクエリ・プランを表示できます。カーソルは `DECLARE CURSOR` コマンドと `OPEN CURSOR` コマンドを使用して宣言、オープンされます。直近にオープンしたカーソルのクエリ・プランを取得するには、次の文を使用します。

```
SELECT GRAPHICAL_PLAN ( );
```

`QUERY_PLAN_AFTER_RUN` オプションが `OFF` の場合、プランは `OPEN CURSOR` または `CLOSE CURSOR` の実行後に表示されます。ただし、`QUERY_PLAN_AFTER_RUN` が `ON` の場合、`CLOSE CURSOR` を実行してから、プランを要求する必要があります。

Interactive SQL の [プラン・ビューワ] ウィンドウで SQL 文に対するクエリ・オプティマイザの実行プランを表示する方法の詳細については、[Viewing graphical plans in Interactive SQL \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/da-guitools-s-3768774.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/da-guitools-s-3768774.html) を参照してください。

Interactive SQL のユーザが UPDATE、DELETE、SELECT INTO、および INSERT SELECT の各クエリのプランを要求しても NOEXEC プランはサポートされていません。クエリ・プランにアクセスするには、最初にクエリを明示的に実行してから、プランを要求します。

例

次の例は SELECT 文を文字列パラメータとして渡して、クエリを実行するためのプランを返し、プランをファイル *gplan.xml* に保存します。

注意 フォーマットされたプラン出力を得るために、ASIS に設定された OUTPUT 文の HEXADEDECIMAL 句を使用すると、各文字の値に制御文字が含まれている場合でも、文字の値はエスケープされずに書き出されます。ASIS は、テキストにタブや復帰改行などのフォーマット文字列が含まれる場合に使用します。

```
SELECT GRAPHICAL_PLAN ('SELECT * FROM
Employees');OUTPUT to 'C:¥gplan.xml' HEXADEDECIMAL ASIS
quote '');
```

次の例は、クエリ・プランがキャッシュにある場合、そこから返します。

```
SELECT GRAPHICAL_PLAN ( );
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

「HTML_PLAN 関数 [文字列]」(184 ページ)

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「OUTPUT 文 [DBISQL]」

『リファレンス：文とオプション』の「NOEXEC オプション」、
「QUERY_PLAN_AFTER_RUN オプション」、
「QUERY_PLAN_AS_HTML オプション」、
「QUERY_PLAN_TEXT_ACCESS オプション」、
「QUERY_PLAN_TEXT_CACHING オプション」

『SQL Anywhere Server SQL リファレンス・マニュアル』の「PLAN 関数 [その他]」、「EXPLANATION 関数 [その他]」、「GRAPHICAL_ULPLAN 関数 [その他]」、「LONG_ULPLAN 関数 [その他]」、「SHORT_ULPLAN 関数 [その他]」

GROUPING 関数 [集合]

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | ROLLUP 処理または CUBE 処理の結果セットに含まれるカラムが NULL であるのは、それが小計ローの一部であるためか、それともデータそのものが NULL であるためかを識別します。 |
| 構文 | GROUPING (<i>group-by-expression</i>) |
| パラメータ | group-by-expression GROUP BY 句に ROLLUP または CUBE のキーワードを使用したクエリの結果セットで、グループ化したカラムとして表示される式。ROLLUP 処理または CUBE 処理によって結果セットに追加された小計ローが、関数によって識別されます。 現在、Sybase IQ は、GROUP BY CUBE を含む PERCENTILE_CONT 関数および PERCENTILE_DISC 関数をサポートしていません。 |
| 戻り値 | <ul style="list-style-type: none"> • 1 <i>group-by-expression</i> が、小計ローの一部なので NULL になっていることを示します。カラムは、小計ローの前にあるカラムではありません。 • 0 <i>group-by-expression</i> が小計ローの前にあるカラムであることを示します。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • SQL99 主要な SQL に含まれない、SQL/foundation の機能です。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | 『リファレンス：文とオプション』の「 SELECT 文 」 『システム管理ガイド 第 2 巻』の「 第 2 章 OLAP の使用 」 |

GROUP_MEMBER 関数 [システム]

| | |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | ユーザが指定されたグループに属しているかどうかを示します。 |
| 構文 | GROUP_MEMBER (<i>group-name-string-expression</i> [, <i>user-name-string-expression</i>]) |
| パラメータ | group-name-string-expression 対象となるグループを示します。 user-name-string-expression 対象となるユーザを示します。指定しなければ、現在のユーザ名とみなされます。 |
| 戻り値 | <ul style="list-style-type: none"> • 0 グループが存在しない場合、ユーザが存在しない場合、またはユーザが指定されたグループに属していない場合は 0 を返します。 • 1 ユーザが指定されたグループのメンバである場合は 0 以外の整数を返します。 |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

HEXTOBIGINT 関数 [データ型変換]

機能 16 進文字列に相当する BIGINT を返します。

構文 **HEXTOBIGINT** (*hexadecimal-string*)

パラメータ **hexadecimal-string** big integer (BIGINT) に変換される 16 進数値です。次のように入力します。先頭に小文字または大文字の 0x を付けるか、または省略することもできます。

```
0xhex-string
0Xhex-string
hex-string
```

例 次の文を実行すると、値 4294967287 が返ります。

```
SELECT HEXTOBIGINT ( '0xffffffff7' ) FROM iq_dummy
SELECT HEXTOBIGINT ( '0Xffffffff7' ) FROM iq_dummy
SELECT HEXTOBIGINT ( 'ffffffff7' ) FROM iq_dummy
```

使用法 HEXTOBIGINT 関数は 16 進数の整数を受け入れ、等価の BIGINT を返します。16 進数の整数は CHAR および VARCHAR 値の式としてだけでなく、BINARY および VARBINARY の式としても指定できます。

HEXTOBIGINT 関数は有効な 16 進文字列 (0x または 0X プレフィックスの有無に関係なく引用符で囲む) を受け入れます。

16 桁未満の入力は左に 0 が埋め込まれるものとします。

入力時にデータ型の変換に失敗すると、CONVERSION_ERROR オプションが OFF に設定されていない場合、Sybase IQ はエラーを返します。CONVERSION_ERROR が OFF に設定されている場合は、無効な 16 進入力に対して NULL が返されます。

BINARY または VARBINARY 値の場合は 8 バイト、CHAR または VARCHAR 値の場合は 16 文字を超えると、値に 0x が追加されている場合を除き、エラーが返されます。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。

参照 『リファレンス: 文とオプション』の「[CONVERSION_ERROR オプション \[TSQL\]](#)」

[「BIGINTTOHEX 関数 \[データ型変換\]」 \(132 ページ\)](#)

[「HEXTOINT 関数 \[データ型変換\]」 \(180 ページ\)](#)

[「INTTOHEX 関数 \[データ型変換\]」 \(191 ページ\)](#)

HEXTOINT 関数 [データ型変換]

機能 16 進文字列に相当する符号なしの BIGINT を返します。

構文 **HEXTOINT** (*hexadecimal-string*)

パラメータ **hexadecimal-string** 整数に変換される文字列。次のように入力します。先頭に小文字または大文字の x を付けるか、または省略することもできます。

```
0xhex-string  
0Xhex-string  
hex-string
```

例 次の文を実行すると、値 420 が返ります。

```
SELECT HEXTOINT ( '0x1A4' ) FROM iq_dummy  
SELECT HEXTOINT ( '0X1A4' ) FROM iq_dummy  
SELECT HEXTOINT ( '1A4' ) FROM iq_dummy
```

使用法 **CONVERSION_ERROR** オプションが **OFF** の場合を除き、無効な 16 進入力に対して **Sybase IQ** からエラーが返されます。**CONVERSION_ERROR** が **OFF** に設定されている場合は、無効な 16 進入力に対して **NULL** が返されます。

データベース・オプション **ASE_FUNCTION_BEHAVIOR** は、**INTTOHEX** と **HEXTOINT** を含む **Sybase IQ** 関数の出力が **Adaptive Server Enterprise** 関数の出力と一致するように指定します。**ASE_FUNCTION_BEHAVIOR** オプションが **ON** の場合、以下の処理が行われます。

- **Sybase IQ HEXTOINT** では、入力は 8 文字の 16 進文字列であるとみなされます。長さが 8 文字未満の場合、文字列の左側には 0 が埋め込まれます。
- **Sybase IQ HEXTOINT** は、0x で始まる最大 16 文字の文字列を処理します。入力値が大きいと、整数値によって 32 ビットの符号付き整数出力サイズがオーバーフローする可能性があるため、注意が必要です。
- **Sybase IQ HEXTOINT** 関数の出力のデータ型は、32 ビットの符号付き整数とみなされます。

- Sybase IQ HEXTOINT 関数は、32 ビットの 16 進整数を符号付き表現として受け入れます。
 - 8 文字を超える 16 進文字列の場合、Sybase IQ HEXTOINT は関連の文字のみ考慮します。
- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。
- 参照
- 『リファレンス：文とオプション』の「[ASE_FUNCTION_BEHAVIOR オプション](#)」
- 『リファレンス：文とオプション』の「[CONVERSION_ERROR オプション \[TSQL\]](#)」
- 「[INTTOHEX 関数 \[データ型変換\]](#)」 (191 ページ)

HOUR 関数 [日付および時刻]

機能 指定された日時から、時間 (hour) に対応する 0 から 23 までの数字を返します。

構文 `HOUR (datetime-expression)`

パラメータ `datetime-expression` 日時。

例 次の文は、値 21 を返します。

```
SELECT HOUR( '1998-07-09 21:12:13' ) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

HOURS 関数 [日付および時刻]

機能 任意の開始日時からの時間数を返すか、指定された 2 つの時刻の間の総時間数を返すか、または `integer-expression` で指定された時間数を時刻に追加します。

構文 `HOURS (datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression)`

パラメータ `datetime-expression` 日時。

integer-expression *datetime-expression* に追加する時間数。*integer-expression*が負の場合、指定された時間数が日時から減算されます。整数式を指定する場合、*datetime-expression* は **datetime** データ型として明示的にキャストする必要があります。

データ型のキャストについては、「[CAST 関数 \[データ型変換\]](#) (134 ページ) を参照してください。

例 次の文は、値 17518758 を返します。

```
SELECT HOURS( '1998-07-13 06:07:12' ) FROM iq_dummy
```

次の文を実行すると、2つの時刻の間の差である値 4 が返されます。

```
SELECT HOURS( '1999-07-13 06:07:12',
              '1999-07-13 10:07:12' ) FROM iq_dummy
```

次の文を実行すると、日時の値 1999-05-13 02:05:07.000 が返ります。

```
SELECT HOURS( CAST( '1999-05-12 21:05:07'
                   AS DATETIME ), 5 ) FROM iq_dummy
```

使用法 2つ目の構文は、最初の引数の日時から2番目の引数の日時までの総時間数を返します。負の値が返ることもあります。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

HTML_DECODE 関数 [HTTP]

機能 HTML リテラル文字列に含まれる特殊文字のエンティティをデコードします。

構文 **HTML_DECODE** (*string*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **string** HTML ドキュメント内で使用される、任意のリテラル文字列。

使用法 この関数は、引数の文字列を以下のように置換して返します。

| 文字列 | 置換 |
|-----|--------|
| " | " |
| ' | ' |

| 文字列 | 置換 |
|--------|-------|
| & | & |
| < | < |
| > | > |
| &#xnn; | 文字 nn |

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

[「HTML_ENCODE 関数 \[HTTP\]」 \(183 ページ\)](#)

[「HTTP_ENCODE 関数 \[HTTP\]」 \(186 ページ\)](#)

HTML_ENCODE 関数 [HTTP]

機能

HTML ドキュメントに挿入する文字列内の特殊文字をエンコードします。

構文

HTML_ENCODE (*string*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ

string HTML ドキュメント内で使用される、任意のリテラル文字列。

使用法

この関数は、引数の文字列を以下のように置換して返します。

| 文字列 | 置換 |
|-------------|--------|
| " | " |
| ' | ' |
| & | & |
| < | < |
| > | > |
| 0X20 以上のコード | &#xnn |

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

- 参照 [「HTML_DECODE 関数 \[HTTP\]」 \(182 ページ\)](#)
[「HTTP_ENCODE 関数 \[HTTP\]」 \(186 ページ\)](#)

HTML_PLAN 関数 [文字列]

機能 クエリ・プランを HTML フォーマット文字列で返します。

構文 **HTML_PLAN** (*string-expression*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されま
す。『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと
削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を
参照してください。

パラメータ **string-expression** プランを生成する SQL 文。通常は SELECT 文で
す、UPDATE 文と DELETE 文も可能です。

HTML_PLAN に引数を指定しない場合、クエリ・プランはキャッシュか
ら返されます。キャッシュにクエリ・プランがない場合は、次のメッ
セージが表示されます。

```
No plan available
```

HTML_PLAN 関数の動作は、データベース・オプション
QUERY_PLAN_TEXT_ACCESS と QUERY_PLAN_TEXT_CACHING で制御
されます。QUERY_PLAN_TEXT_ACCESS が OFF (デフォルト) の場合、
次のメッセージが表示されます。

```
plan not available. The database option  
QUERY_PLAN_TEXT_ACCESS is OFF
```

QUERY_PLAN_TEXT_ACCESS が ON であり、しかもサーバに保持され
ているキャッシュに文字列式のクエリ・プランがある場合は、キャッ
シュからクエリ・プランが返されます。

HTML_PLAN 関数を使用して、SELECT、UPDATE、DELETE、INSERT SELECT、
SELECT INTO を使用した Interactive SQL にクエリ・プランを返せます。

IQ クエリのためにオープンするカーソルのクエリ・プランを表示でき
ます。直近にオープンしたカーソルのクエリ・プランを取得するには、
次の文を使用します。

```
SELECT HTML_PLAN ( );
```

QUERY_PLAN_AFTER_RUN オプションが OFF の場合、プランは OPEN CURSOR または CLOSE CURSOR の実行後に表示されます。ただし、QUERY_PLAN_AFTER_RUN が ON の場合、CLOSE CURSOR を実行してから、プランを要求する必要があります。

Interactive SQL の [プラン・ビューワ] ウィンドウで SQL 文に対するクエリ・オプティマイザの実行プランを表示する方法の詳細については、[Viewing plans using the Interactive SQL Plan Viewer \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/da-guitools-s-5971976.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/da-guitools-s-5971976.html) を参照してください。

Interactive SQL のユーザが UPDATE、DELETE、SELECT INTO、および INSERT SELECT の各クエリのプランを要求しても NOEXEC プランはサポートされていません。クエリ・プランにアクセスするには、最初にクエリを明示的に実行してから、プランを要求します。

SQL Anywhere クエリまたは OMNI/CIS 分解クエリの HTML_PLAN を要求すると、次のメッセージが返されます。

```
プランがありません。このタイプの文またはデータベースでは、HTML_PLAN
関数はサポートされていません。
```

例

次の例は、SELECT 文を文字列パラメータとして渡して、クエリを実行するための HTML プランを返し、プランをファイル *hplan.html* に保存します。

```
SELECT HTML_PLAN ('SELECT * FROM Employees');
OUTPUT to 'C:\%hplan.html' HEXADECIMAL ASIS QUOTE '';
```

OUTPUT TO 句の HEXADECIMAL ASIS は、テキストにタブや復帰改行などのフォーマット文字列が含まれる場合に使用します。ASIS を設定すると、値はたとえ制御文字を含んでいても、そのまま (エスケープなしで) 書き込まれます。

次の例は、HTML クエリ・プランがキャッシュにある場合、そこから返します。

```
SELECT HTML_PLAN ( );
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

[「GRAPHICAL_PLAN 関数 \[文字列\]」 \(175 ページ\)](#)

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[OUTPUT 文 \[DBISQL\]](#)」

『リファレンス：文とオプション』の「[NOEXEC オプション](#)」、
「[QUERY_PLAN_AFTER_RUN オプション](#)」、
「[QUERY_PLAN_AS_HTML オプション](#)」、
「[QUERY_PLAN_TEXT_ACCESS オプション](#)」、
「[QUERY_PLAN_TEXT_CACHING オプション](#)」

『SQL Anywhere Server SQL リファレンス・マニュアル』の「[PLAN 関数 \[その他\]](#)」、「[EXPLANATION 関数 \[その他\]](#)」、「[GRAPHICAL_ULPLAN 関数 \[その他\]](#)」、「[LONG_ULPLAN 関数 \[その他\]](#)」、「[SHORT_ULPLAN 関数 \[その他\]](#)」

HTTP_DECODE 関数 [HTTP]

機能 文字列内の特殊文字をデコードして、HTTP で使えるようにします。

構文 `HTTP_DECODE (string)`

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されません。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **string** HTTP 要求に使用される、任意の文字列。

使用法 この関数は、引数の文字列のうち、`%nn` (`nn` は 16 進値) の形式の文字シーケンスを、`nn` をコードに持つ文字にすべて置換して返します。また、プラス記号 (+) はすべてスペースに置換します。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

参照 「[HTML_ENCODE 関数 \[HTTP\]](#)」 (183 ページ)

「[HTTP_ENCODE 関数 \[HTTP\]](#)」 (186 ページ)

HTTP_ENCODE 関数 [HTTP]

機能 文字列内の特殊文字をエンコードして、HTTP で使えるようにします。

構文 **HTML_ENCODE (string)**

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されません。『パフォーマンス&チューニング・ガイド』の「第 3 章 クエリと削除の最適化」の「SQL Anywhere による処理を引き起こす条件」を参照してください。

パラメータ **string** HTTP 要求に使用される、任意の文字列。

使用法 この関数は、引数の文字列を以下のように置換して返します。また、1F よりも小さいか 7E よりも大きい 16 進コードを持つ文字は、すべて %nn (nn は文字コード) に置換します。

| 文字 | 置換 |
|------|-----|
| スペース | %20 |
| " | %22 |
| # | %23 |
| & | %26 |
| , | %2C |
| ; | %3B |
| < | %3C |
| > | %3E |
| [| %5B |
| ¥ | %5C |
|] | %5D |
| ` | %60 |
| { | %7B |
| | %7C |
| } | %7D |

- 標準と互換性**
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照 [「HTML_ENCODE 関数 \[HTTP\]」 \(183 ページ\)](#)
[「HTTP_DECODE 関数 \[HTTP\]」 \(186 ページ\)](#)

HTTP_HEADER 関数 [HTTP]

機能 HTTP ヘッダの値を取得します。

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | HTML_HEADER (<i>field-name</i>) |
| | <hr/> <p>注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「SQL Anywhere による処理を引き起こす条件」を参照してください。</p> <hr/> |
| パラメータ | field-name HTTP ヘッダ・フィールドの名前。 |
| 使用法 | この関数は、指定した HTTP ヘッダ・フィールドの値を返します。HTTP 要求を Web サービス経由で処理しているときに使用されます。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | <p>「HTTP_VARIABLE 関数 [HTTP]」 (188 ページ)</p> <p>「NEXT_HTTP_HEADER 関数 [HTTP]」 (211 ページ)</p> <p>「NEXT_HTTP_VARIABLE 関数 [HTTP]」 (211 ページ)</p> |

HTTP_VARIABLE 関数 [HTTP]

| | |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | HTTP の変数の値を取得します。 |
| 構文 | HTML_VARIABLE ([<i>var-name</i> [[, <i>instance</i>] , <i>header-field</i>]) |
| | <hr/> <p>注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「SQL Anywhere による処理を引き起こす条件」を参照してください。</p> <hr/> |
| パラメータ | <p>var-name HTTP の変数の名前。</p> <p>instance 同じ名前の変数が2つ以上ある場合、フィールド・インスタンスのインスタンス番号。または、NULL を指定すると、最初のインスタンスが取得されます。リストで複数選択する際に使用します。</p> <p>header-field マルチパートの変数において、指定するフィールドに関連付けられたヘッダ・フィールド名。</p> |
| 使用法 | この関数は、指定された HTTP 変数の値を返します。HTTP 要求を Web サービス経由で処理しているときに使用されます。 |

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。
「HTML_DECODE 関数 [HTTP]」(182 ページ)
「NEXT_HTTP_HEADER 関数 [HTTP]」(211 ページ)
「NEXT_HTTP_VARIABLE 関数 [HTTP]」(211 ページ)

IFNULL 関数 [その他]

機能

最初の式が NULL 値の場合は、第 2 の式の値を返します。最初の式が NULL でない場合、第 3 の式の値を返します。第 3 の式が指定されておらず、最初の式が NULL でない場合は、NULL 値を返します。

構文

IFNULL (*expression1*, *expression2* [, *expression3*])

パラメータ

expression1 評価される式。この値によって、*expression2* と *expression3* のどちらを返すかが決まります。

expression2 *expression1* が NULL であった場合に返される値。

expression3 *expression1* が NULL でなかった場合に返される値。

例

次の文は、値 -66 を返します。

```
SELECT IFNULL( NULL, -66 ) FROM iq_dummy
```

次の文は、第 1 の式が NULL でなく、第 3 の式が指定されていないため、NULL を返します。

```
SELECT IFNULL( -66, -66 ) FROM iq_dummy
```

標準と互換性

- **SQL92** Transact SQL の拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

INDEX_COL 関数 [システム]

機能

インデックス・カラムの名前を返します。

構文

INDEX_COL (*table-name*, *index-id*, *key_#* [, *user-id*])

パラメータ

table-name テーブル名。

index-id *table-name* のインデックスのインデックス ID。

key_# *index-id* で指定されたインデックスのキー。このパラメータには、インデックス内のカラム番号を指定します。単一カラムのインデックスの場合、*key_#* は 0 になります。複数カラムのインデックスでは、1 目目のカラムの *key_#* は 0、2 目目のカラムは 1、というように指定します。

user-id *table-name* の所有者のユーザ ID。 *user-id* が省略された場合、この値はデフォルトで、関数を呼び出したユーザの ID になります。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。

参照

[「OBJECT_ID 関数 \[システム\]」 \(216 ページ\)](#)

INSERTSTR 関数 [文字列]

機能

文字列を、別の文字列の指定された位置に挿入します。

構文

INSERTSTR (*numeric-expression*, *string-expression1*, *string-expression2*)

パラメータ

numeric-expression この位置の後に、*string-expression2* が挿入されず。先頭に文字列を挿入するには、ゼロを使用します。

string-expression1 *string-expression2* が挿入される文字列。

string-expression2 挿入する文字列。

注意 INSERTSTR 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で INSERTSTR を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して INSERTSTR を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」を参照してください。

例

次の文は、値 “backoffice” を返します。

```
SELECT INSERTSTR( 0, 'office ', 'back' ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。STUFF 関数が同じ機能を持っていて、Adaptive Server Enterprise および Sybase IQ でサポートされています。

参照

[「STUFF 関数 \[文字列\]」 \(265 ページ\)](#)

INTTOHEX 関数 [データ型変換]

| | |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 10 進の整数を 16 進数に変換して返します。 |
| 構文 | INTTOHEX (<i>integer-expression</i>) |
| パラメータ | integer-expression 16 進数に変換される整数。 |
| 例 | 次の文は、値 3B9ACA00 を返します。 <pre>SELECT INTTOHEX(1000000000) FROM iq_dummy</pre> 次の文は、値 00000002540BE400 を返します。 <pre>SELECT INTTOHEX(10000000000) FROM iq_dummy</pre> |
| 使用法 | <p>INTTOHEX への入力のデータ変換に失敗すると、CONVERSION_ERROR オプションが OFF に設定されていない場合は、Sybase IQ はエラーを返します。OFF に設定されている場合は、NULL を返します。</p> <p>ASE_FUNCTION_BEHAVIOR オプション データベース・オプション ASE_FUNCTION_BEHAVIOR は、INTTOHEX と HEXTOINT を含む IQ 関数の出力が Adaptive Server Enterprise 関数の出力と一致するように指定します。ASE_FUNCTION_BEHAVIOR のデフォルト値は OFF です。</p> <p>ASE_FUNCTION_BEHAVIOR オプションが無効な場合 (値が OFF の場合)、次のようになります。</p> <ul style="list-style-type: none"> • INTTOHEX の出力は、SQL Anywhere と互換します。 • 入力に応じて、INTTOHEX の出力は 8 桁または 16 桁で、左側に 0 が埋め込まれます。戻り値のデータ型は VARCHAR です。 • INTTOHEX の出力には、0x および 0X プレフィクスはありません。 • INTTOHEX には、64 ビットまでの整数を入力できます。 <p>ASE_FUNCTION_BEHAVIOR オプションが有効な場合 (値が ON の場合) は、次のようになります。</p> <ul style="list-style-type: none"> • INTTOHEX の出力は、ASE 互換です。 • INTTOHEX の出力は常に 8 桁で、左側に 0 が埋め込まれます。戻り値のデータ型は VARCHAR です。 • INTTOHEX の出力には、0x および 0X プレフィクスはありません。 • Sybase IQ INTTOHEX では、入力を 32 ビットの符号付き整数とみなします。大きい値を指定するとオーバーフローが発生し、変換エラーが起こる場合があります。次の文を例にとります。 <pre>SELECT INTTOHEX(1000000000) FROM iq_dummy</pre> |

この結果は 3B9ACA00 になります。もう 1 つの例を示します。

```
SELECT INTOHEX( 10000000000 ) FROM iq_dummy
```

この場合は、変換エラーが発生します。

標準と互換性

- **SQL92** Transact SQL の拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

参照

『リファレンス：文とオプション』の「[CONVERSION_ERROR オプション \[TSQL\]](#)」

『リファレンス：文とオプション』の「[ASE_FUNCTION_BEHAVIOR オプション](#)」

「[HEXTOINT 関数 \[データ型変換\]](#)」 (180 ページ)

ISDATE 関数 [日付および時刻]

機能

引数の文字列が、日付に変換可能かどうかを調べます。変換可能であれば 1 が返され、可能でなければ 0 が返されます。引数が null に設定されている場合は、0 が返されます。

構文

ISDATE (*string*)

パラメータ

string 文字列表現が日付として有効かどうかを調べて判断される文字列。

例

次の例では、`birth_date` カラムに格納されている値が有効な日付かどうかを調べ、無効な日付であれば `NULL` を返し、有効な日付であれば `date` フォーマットで返しています。

```
select birth_date from MyData;
-----
1990/32/89
0101/32/89
1990/12/09

select
  case when isdate(birth_date)=0 then NULL
  else cast(birth_date as date)
  end
  from MyData;
-----
(NULL)
(NULL)
1990-12-09
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **SQL99** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

ISNULL 関数 [その他]

機能 パラメータ・リスト内で最初の NULL でない式の値を返します。

構文 **ISNULL** (*expression*, *expression* [..., *expression*])

パラメータ **expression** NULL かどうか調べられる式。
少なくとも2つの式を関数に渡す必要があります。

例 次の文は、値 -66 を返します。

```
SELECT ISNULL( NULL , -66, 55, 45, NULL, 16 ) FROM  
iq_dummy
```

使用法 ISNULL 関数は COALESCE 関数と同じです。

- 標準と互換性
- **SQL92** Transact SQL の拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照 「COALESCE 関数 [その他]」 (138 ページ)

ISNUMERIC 関数 [その他]

機能 引数の文字列が、数値に変換可能かどうかを調べます。変換可能であれば 1 が返され、可能でなければ 0 が返されます。引数が null に設定されている場合は、0 が返されます。

構文 **ISNUMERIC** (*string*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **string** 文字列表現が数値として有効かどうかを調べて判断される文字列。

使用法 パフォーマンスを最適化するため、述部で **ISNUMERIC** を使用しないでください。使用した場合、製品の **SQL Anywhere** の部分で処理され、**Sybase IQ** のパフォーマンス機能を利用することができません。

例 次の例では、**height_in_cms** カラムに格納されている値が有効な数値データかどうかを調べ、無効な数値データであれば **NULL** を返し、有効な数値データであれば **int** フォーマットで返しています。

```
data height_in_cms
-----
asde
asde
180
156

select case
  when isnumeric(height_in_cms)=0
  then NULL
  else cast(height_in_cms as int)
  end
from MyData
```

- 標準と互換性**
- **SQL92** ベンダの拡張機能。
 - **SQL99** ベンダの拡張機能。
 - **Sybase Adaptive Server Enterprise** ではサポートされていません。

LAST_VALUE 関数 [集合]

機能 一連の値の最後の値を返します。

構文 **LAST_VALUE** (*expression* [IGNORE NULLS | RESPECT NULLS])
OVER (*window-spec*)

パラメータ **expression** 順序付けられたセットの最後の値を決定する式。

使用法 **LAST_VALUE** は、(通常順序付けられた) 値のセット内の最後の値を返します。セットの最後の値が **NULL** の場合、**IGNORE NULLS** が指定されていなければ **NULL** が返されます。**IGNORE NULLS** を指定した場合、**LAST_VALUE** は、セット内で **NULL** でない最後の値を返します。すべての値が **NULL** の場合は **NULL** が返されます。

戻り値のデータ型は、入力値のデータ型と同じです。

expression に **LAST_VALUE** およびその他の分析関数を使用することはできません。したがって、分析関数をネストすることはできませんが、その他の組み込み関数式を **expression** に使用できます。

window-spec に **ORDER BY** が含まれない場合、結果は任意です *window-spec* がない場合、結果は任意です。

関数構文 (インライン) または **SELECT** 文の **WINDOW** 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

注意 **DISTINCT** はサポートされていません。

例

次の例は、各従業員の給与、およびその部署で最も高い給与の従業員の名前を返します。

```
SELECT GivenName + ' ' + Surname AS employee_name,
       Salary, DepartmentID,
       LAST_VALUE( employee_name ) OVER Salary_Window AS
highest_paid
FROM Employees
WINDOW Salary_Window AS ( PARTITION BY DepartmentID
ORDER BY Salary
RANGE BETWEEN UNBOUNDED PRECEDING
AND UNBOUNDED FOLLOWING )
ORDER BY DepartmentID DESC;
```

次の結果セットが返されます。

表 4-23 :

| employee_name | Salary | DepartmentID | highest_paid |
|--------------------|-----------|--------------|---------------|
| Michael Lynch | 24903.000 | 500 | Jose Martinez |
| Joseph Barker | 27290.000 | 500 | Jose Martinez |
| Sheila Romero | 27500.000 | 500 | Jose Martinez |
| Felicia Kuo | 28200.000 | 500 | Jose Martinez |
| Jeannette Bertrand | 29800.000 | 500 | Jose Martinez |
| Jane Braun | 34300.000 | 500 | Jose Martinez |
| Anthony Rebeiro | 34576.000 | 500 | Jose Martinez |
| Charles Crowley | 41700.000 | 500 | Jose Martinez |
| Jose Martinez | 55500.800 | 500 | Jose Martinez |
| Doug Charlton | 28300.000 | 400 | Scott Evans |
| Elizabeth Lambert | 29384.000 | 400 | Scott Evans |
| Joyce Butterfield | 34011.000 | 400 | Scott Evans |
| Robert Nielsen | 34889.000 | 400 | Scott Evans |
| Alex Ahmed | 34992.000 | 400 | Scott Evans |
| Ruth Wetherby | 35745.000 | 400 | Scott Evans |
| ... | ... | ... | ... |

- 標準と互換性
- **SQL2008** SQL/OLAP 機能 T612。
 - **Sybase** SQL Anywhere 互換。

LCASE 関数 [文字列]

機能 文字列内のすべての文字を小文字に変換します。

構文 **LCASE** (*string-expression*)

パラメータ **string-expression** 小文字に変換される文字列。

注意 LCASE 関数の結果データ型は LONG VARCHAR です。SELECT INTO 文で LCASE を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して LCASE を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」を参照してください。

例 次の文を実行すると、値 “lower case” が返ります。

```
SELECT LCASE( 'LOWER CasE' ) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** LCASE は、Adaptive Server Enterprise でサポートされていません。同じ機能を使用する場合は、LOWER を使用します。

参照 [「LOWER 関数 \[文字列\]」 \(201 ページ\)](#)
[「UCASE 関数 \[文字列\]」 \(328 ページ\)](#)
[「UPPER 関数 \[文字列\]」 \(329 ページ\)](#)

LEFT 関数 [文字列]

機能 文字列の先頭から、指定された数だけ文字を返します。

構文 **LEFT** (*string-expression, numeric-expression*)

パラメータ **string-expression** 文字列。

numeric-expression 返される文字数。

例 次の文は、値 “choco” を返します。

```
SELECT LEFT( 'chocolate', 5 ) FROM iq_dummy
```

使用法 文字列にマルチバイト文字が含まれ、照合が適切に使用されている場合は、返されるバイト数が、指定された文字数よりも多くなる場合があります。

注意 LEFT 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で LEFT を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して LEFT を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」を参照してください。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

参照

「[RIGHT 関数 \[文字列\]](#)」(245 ページ)

『システム管理ガイド 第1巻』の「[第11章 各国語と文字セット](#)」

LEN 関数 [文字列]

機能

BINARY データ型または STRING データ型の入力として1つの引数を取り、指定された文字列表現の後続ブランクを除いた、データベースの照合順序で定義されている文字数を返します。マルチバイト文字セットの場合、結果が文字列のバイト長と異なる場合があります。

BINARY および VARBINARY も使用できます。その場合は、LEN() によって入力のバイト数が返されます。

LEN は、LENGTH 関数のエイリアスです。

構文

LEN (*string_expr*)

パラメータ

string_expr 評価する文字列式です。

例

次の例は、値 3152 を返します。

```
select len(Photo) from Products
where ID = 500
```

使用法

この関数は、CHAR_LENGTH (*string_expression*) と同じです。

パーミッション

すべてのユーザが LEN 関数を実行できます。

標準と互換性

ANSI SQL 準拠レベル：Transact SQL の拡張機能。

参照

データ型 CHAR、NCHAR、VARCHAR、NVARCHAR

詳細については、「[第 3 章 SQL データ型](#)」を参照してください。

関数 「[CHAR_LENGTH 関数 \[文字列\]](#)」(136 ページ)、[「STR_REPLACE 関数 \[文字列\]](#)」(262 ページ)

文字列関数の一般的な説明については、「[文字列関数](#)」(117 ページ)を参照してください。

LENGTH 関数 [文字列]

機能 指定された文字列に含まれる文字数を返します。

構文 **LENGTH** (*string-expression*)

パラメータ **string-expression** 文字列。

例 次の文は、値 9 を返します。

```
SELECT LENGTH( 'chocolate' ) FROM iq_dummy
```

使用法 文字列にマルチバイト文字が含まれ、照合が適切に使用されていれば、LENGTH はバイト数ではなく文字数を返します。文字列のデータ型が BINARY である場合、LENGTH 関数は BYTE_LENGTH と同じように機能します。

LENGTH 関数は CHAR_LENGTH 関数と同じです。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。CHAR_LENGTH 関数を使用してください。

参照 [「BYTE_LENGTH 関数 \[文字列\]](#)」(133 ページ)

[「CHAR_LENGTH 関数 \[文字列\]](#)」(136 ページ)

『システム管理ガイド 第 1 巻』の「[第 11 章 各国語と文字セット](#)」

LN 関数 [数値]

機能 指定された式の自然対数を返します。

構文 **LN** (*numeric-expression*)

パラメータ **expression** カラム、変数、またはデータ型が真数値、概数値、通貨、またはこれらの型の 1 つに暗黙的に変換できる式です。他のデータ型を指定すると、LN 関数ではエラーが返ります。戻り値は、DOUBLE データ型です。

| | |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 使用法 | LN は 1 つの引数を取ります。たとえば LN (20) は、2.995732 を返します。 |
| | LN 関数は LOG 関数のエイリアスです。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。LOG 関数を使用してください。 |
| 参照 | <p>「LOG 関数 [数値]」(200 ページ)</p> <p>『システム管理ガイド 第 1 巻』の「第 11 章 各国語と文字セット」</p> |

LOCATE 関数 [文字列]

| | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | ある文字列の、別の文字列内の位置を返します。 |
| 構文 | LOCATE (<i>string-expression1</i> , <i>string-expression2</i> [, <i>numeric-expression</i>]) |
| パラメータ | <p>string-expression1 検索される文字列。</p> <p>string-expression2 検索する文字列。255 バイトまでの文字列を指定してください。</p> <p>numeric-expression 文字列内で検索を開始する文字位置。最初の文字の位置は 1 です。開始オフセットが負の場合、LOCATE は、最初ではなく最後にマッチする文字列のオフセットを返します。負のオフセットは、文字列の末尾から何文字を検索から除外するかを示します。除外するバイト数は、(-1 * offset) - 1 で計算します。</p> <p><i>numeric-expression</i> は、CHAR カラム、VARCHAR カラム、BINARY カラムの 32 ビット符号付き整数です。</p> |
| 例 | <p>次の文は、値 8 を返します。</p> <pre>SELECT LOCATE('office party this week - rsvp as soon as possible', 'party', 2) FROM iq_dummy</pre> <p>2 つ目の例は、検索の開始オフセットである <i>numeric-expression</i> に負の数が指定されています。</p> <pre>CREATE TABLE t1(name VARCHAR(20), dirname VARCHAR(60)); INSERT INTO t1 VALUES ('m1000', 'c:¥test¥functions¥locate.sql'); INSERT INTO t1 VALUES ('m1001', 'd:¥test¥functions¥trim.sql'); COMMIT;</pre> <pre>SELECT LOCATE(dirname, '¥', -1), dirname FROM t1;</pre> |

結果は次のようになります。

```
18 c:¥test¥functions¥locate.sql
18 d:¥test¥functions¥trim.sql
```

使用法

numeric-expression を指定した場合は、検索対象の文字列内のそのオフセット位置から検索を開始します。

numeric-expression を指定しなかった場合、LOCATE は、指定した文字列の最初のインスタンスの位置だけを返します。

最初の文字列には、サイズの大きい (255 バイトより大きい) 文字列を指定できますが、2 番目の文字列は 255 バイト以下である必要があります。2 番目の引数にサイズの大きい文字列が指定されると、関数からは NULL 値が返されます。

引数のどれか 1 つでも NULL の場合、結果は NULL になります。

長さが 0 の文字列を検索すると、1 が返されます。

指定した文字列が文字列に含まれない場合、LOCATE はゼロ (0) を返します。

LOCATE 関数で返されるか指定される位置またはオフセットはすべて、常に文字オフセットであり、マルチバイト・データの場合はバイト・オフセットとは異なることがあります。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

『Sybase IQ によるラージ・オブジェクト管理』の「[第 4 章 関数のサポート](#)」

LOG 関数 [数値]

機能

数値の自然対数を返します。

LN は LOG のエイリアスです。

構文

LOG (*numeric-expression*)

パラメータ

numeric-expression 数値。

例

次の文は、値 3.912023 を返します。

```
SELECT LOG( 50 ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

参照

「[LOG10 関数 \[数値\]](#)」 (201 ページ)

LOG10 関数 [数値]

| | |
|--------|---------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 数値の、底を 10 とする対数を返します。 |
| 構文 | LOG10 (<i>numeric-expression</i>) |
| パラメータ | numeric-expression 数値。 |
| 例 | 次の式を実行すると、値 1.698970 が返ります。 <pre>SELECT LOG10(50) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 ベンダの拡張機能。• Sybase Adaptive Server Enterprise 互換。 |
| 参照 | 「LOG 関数 [数値]」 (200 ページ) |

LOWER 関数 [文字列]

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 文字列内のすべての文字を小文字に変換します。 |
| 構文 | LOWER (<i>string-expression</i>) |
| パラメータ | string-expression 変換される文字列。 |
| <hr/> 注意 LOWER 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で LOWER を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して LOWER を正しいデータ型とサイズに設定する必要があります。 詳細については、「 REPLACE 関数 [文字列] 」を参照してください。 <hr/> | |
| 例 | 次の文を実行すると、値 “lower case” が返ります。 <pre>SELECT LOWER('LOWER CasE') FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 SQL92 互換。• Sybase Adaptive Server Enterprise 互換。 |
| 参照 | 「LCASE 関数 [文字列]」 (196 ページ) 「UCASE 関数 [文字列]」 (328 ページ) 「UPPER 関数 [文字列]」 (329 ページ) |

LTRIM 関数 [文字列]

| | |
|-------|-------------------------------------------|
| 機能 | 文字列から先行ブランクを削除します。 |
| 構文 | LTRIM (<i>string-expression</i>) |
| パラメータ | string-expression トリミングされる文字列。 |

注意 LTRIM 関数の結果データ型は LONG VARCHAR です。SELECT INTO 文で LTRIM を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して LTRIM を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」(243 ページ)を参照してください。

例 次の文は、先行ブランクがすべて削除された値 “Test Message” を返します。

```
SELECT LTRIM( '      Test Message' ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** 互換。

参照

- [「RTRIM 関数 \[文字列\]」](#) (248 ページ)
- [「TRIM 関数 \[文字列\]」](#) (268 ページ)

MAX 関数 [集合]

| | |
|-------|---------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | ローの各グループ内での <i>expression</i> の最大値を返します。 |
| 構文 | MAX (<i>expression</i> DISTINCT <i>column-name</i>) |
| パラメータ | expression 最大値が計算される式。通常はカラムを指定します。 DISTINCT column-name MAX (<i>expression</i>) を使用した場合と戻り値は同じです。万全を期すために含まれています。 |
| 例 | 次の文は、Employees テーブル内の salary の最大値である値 138948.000 を返します。 <pre>SELECT MAX (Salary) FROM Employees</pre> |
| 使用法 | <i>expression</i> が NULL になるローは無視されます。ローがまったくないグループに対しては、NULL 値を返します。 |

- 標準と互換性
- **SQL92** SQL92 互換。
 - **Sybase** Adaptive Server Enterprise 互換。
- 参照
- [「MIN 関数 \[集合\]」\(204 ページ\)](#)
『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

MEDIAN 関数 [集合]

- 機能
- 式の中央値を返します。
- 構文 1
- ```
MEDIAN([ALL | DISTINCT] expression)
```
- 構文 2
- ```
MEDIAN([ALL | DISTINCT] expression)  
OVER (window-spec)
```
- column-spec* : 以下の「使用法」の構文 2 の手順を参照してください。
- パラメータ
- expression** 計算する中央値の数値式。
- 使用法
- 中央値は、標本分布、人口分布、または確率分布の上半分と下半分を分割する数値です。

戻り値のデータ型は、入力値のデータ型と同じです。NULL は、中央値の計算で無視されます。オプションのキーワード **DISTINCT** を使用して、集合関数を適用する前に重複値を排除できます。すべてのローにオペレーションを実行する **ALL** がデフォルトです。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

注意 *window-spec* は、ROW、RANGE、または ORDER BY 指定を含むことはできません。*window-spec* が指定できるのは、PARTITION 句だけです。WINDOW 句を使用する場合、DISTINCT はサポートされません。

- 例
- 次のクエリは、フロリダの各部署の給与の中央値を返します。

```
SELECT DepartmentID, Surname, Salary,  
       MEDIAN(Salary) OVER (PARTITION BY DepartmentID)
```

```
"Median"
FROM Employees
WHERE State IN ('FL')
```

結果は次のようになります。

表 4-24 : MEDIAN 結果セット

| DepartmentID | Surname | Salary | Median |
|--------------|----------|-----------|-----------|
| 100 | Lull | 87900.000 | 73870.000 |
| 100 | Gowda | 59840.000 | 73870.000 |
| 200 | Sterling | 64900.000 | 76200.000 |
| 200 | Kelly | 87500.000 | 76200.000 |
| 300 | Litton | 58930.000 | 58930.000 |
| 400 | Francis | 53870.000 | 3870.000 |
| 400 | Charlton | 28300.000 | 53870.000 |
| 400 | Evans | 68940.000 | 53870.000 |

標準と互換性

- **SQL2008** ベンダの拡張機能。

MIN 関数 [集合]

機能

ローの各グループ内での *expression* の最小値を返します。

構文

```
MIN ( expression
| DISTINCT column-name )
```

パラメータ

expression 最小値が計算される式。通常はカラムを指定します。

DISTINCT column-name *MIN (expression)* を使用した場合と戻り値は同じです。万全を期すために含まれています。

例

次の文は、Employees テーブル内の salary の最小値である値 24903.000 を返します。

```
SELECT MIN ( Salary )
FROM Employees
```

使用法

expression が NULL になるローは無視されます。ローがまったくないグループに対しては、NULL 値を返します。

標準と互換性

- **SQL92** SQL92 互換。
- **Sybase** Adaptive Server Enterprise 互換。

参照

[「MAX 関数 \[集合\]」 \(202 ページ\)](#)

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

MINUTE 関数 [日付および時刻]

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定された日時の分に対応する 0 から 59 までの数値を返します。 |
| 構文 | MINUTE (<i>datetime-expression</i>) |
| パラメータ | datetime-expression 日時の値。 |
| 例 | 次の文は、値 22 を返します。 <pre>SELECT MINUTE('1998-07-13 12:22:34') FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |

MINUTES 関数 [日付および時刻]

| | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 任意の日時からの分の数を返すか、指定された 2 つの時刻の間の分の数 (何分以上あるか) を返すか、または <i>integer-expression</i> で指定された分の数を時刻に追加します。 |
| 構文 | MINUTES (<i>datetime-expression</i> <i>datetime-expression</i> , <i>datetime-expression</i> <i>datetime-expression</i> , <i>integer-expression</i>) |
| パラメータ | datetime-expression 日時。 integer-expression <i>datetime-expression</i> に追加する分数。 <i>integer-expression</i> が負の場合、指定された分数が日時から減算されます。整数式を指定する場合、 <i>datetime-expression</i> は <i>datetime</i> データ型として明示的にキャストする必要があります。 データ型のキャストについては、「 CAST 関数 [データ型変換] 」(134 ページ) を参照してください。 |
| 例 | 次の文は、値 1051125487 を返します。 <pre>SELECT MINUTES('1998-07-13 06:07:12') FROM iq_dummy</pre> 次の文を実行すると、2 つの時刻の間の差である値 240 が返されます。 <pre>SELECT MINUTES('1999-07-13 06:07:12', '1999-07-13 10:07:12') FROM iq_dummy</pre> 次の文を実行すると、日時の値 1999-05-12 21:10:07.000 が返ります。 <pre>SELECT MINUTES(CAST('1999-05-12 21:05:07' AS DATETIME), 5) FROM iq_dummy</pre> |
| 使用法 | 2 つ目の構文は、最初の引数の日時から 2 番目の引数の日時までが何分以上あるかを返します。負の値が返ることもあります。 |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

MOD 関数 [数値]

機能 ある整数を別の整数で割った余りを返します。

構文 **MOD** (*dividend*, *divisor*)

パラメータ **dividend** 割られる数。つまり分数の分子。

divisor 割る数。つまり分数の分母。

例 次の文は、値 2 を返します。

```
SELECT MOD ( 5, 3 ) FROM iq_dummy
```

使用法 負の *dividend* が含まれる除算の場合、結果は負または 0 になります。*divisor* の符号は結果に影響しません。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。Adaptive Server Enterprise では、% 演算子をモジュロ演算子として使用します。

参照 [「REMAINDER 関数 \[数値\]」 \(241 ページ\)](#)

MONTH 関数 [日付および時刻]

機能 指定された日付の月に対応する 1 から 12 までの数字を返します。

構文 **MONTH** (*date-expression*)

パラメータ **date-expression** 日時の値。

例 次の文は、値 7 を返します。

```
SELECT MONTH ( '1998-07-13' ) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

MONTHNAME 関数 [日付および時刻]

機能 指定された日付式から月の名前を返します。

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | MONTHNAME (<i>date-expression</i>) |
| パラメータ | date-expression 日時の値。 |
| 例 | DATE_ORDER オプションがデフォルト値である <i>ymd</i> にセットされている場合、次の文は値 September を返します。 <pre>SELECT MONTHNAME('1998-09-05') FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | 『リファレンス：文とオプション』の「 DATE_ORDER オプション 」 |

MONTHS 関数 [日付および時刻]

| | |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 任意の開始日時からの月数を返すか、指定された2つの日時の間の月数を返すか、または <i>integer-expression</i> で指定された月数を日時に追加します。 |
| 構文 | MONTHS (<i>date-expression</i> <i>date-expression, datetime-expression</i> <i>date-expression, integer-expression</i>) |
| パラメータ | <p>date-expression 日時。</p> <p>integer-expression <i>datetime-expression</i> に追加する月数。<i>integer-expression</i> が負の場合、指定された月数が日時の値から減算されます。整数式を指定する場合、<i>date-expression</i> は <i>datetime</i> データ型として明示的にキャストする必要があります。</p> <p>データ型のキャストについては、「CAST 関数 [データ型変換]」 (134 ページ) の項を参照してください。</p> |
| 例 | <p>次の文は、値 23982 を返します。</p> <pre>SELECT MONTHS('1998-07-13 06:07:12') FROM iq_dummy</pre> <p>次の文を実行すると、2つの日付の間の差である値 2 が返されます。</p> <pre>SELECT MONTHS('1999-07-13 06:07:12', '1999-09-13 10:07:12') FROM iq_dummy</pre> <p>次の文を実行すると、日時の値 1999-10-12 21:05:07.000 が返ります。</p> <pre>SELECT MONTHS(CAST('1999-05-12 21:05:07' AS DATETIME), 5) FROM iq_dummy</pre> |

使用法

最初の構文は、任意の開始日から経過した月数を返します。この値は、2つの日付/時刻式が同年同月かどうかを判別する場合に役立ちます。

```
MONTHS( invoice_sent ) = MONTHS( payment_received )
```

請求書が送られてから 12 か月後に支払が行われる場合、MONTH 関数の比較は正しく行われないうちに注意してください。

2番目の構文は、最初の引数の日付から2番目の引数の日付までの月数を返します。負の値が返ることもあります。MONTHS 関数の値は、2つの日付の間にある月初めの日(各月の1日)の数から算出されます。時間、分、秒は無視されます。

3番目の構文は、指定された日付に *integer-expression* の月数を追加します。新しい日付が月末 (MONTHS ('1992-01-31', 1) など) を過ぎている場合、結果を演算結果の月の末日に設定します。*integer-expression* が負の場合、指定された月数が日付から減算されます。時間、分、秒は無視されます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

NEWID 関数 [その他]

機能

UUID (Universally Unique Identifier) 値を生成します。返される UUID 値はバイナリです。UUID は GUID (Globally Unique Identifier) と同じです。

構文

NEWID ()

パラメータ

NEWID() に指定するパラメータはありません。

例

次の文では、テーブル t1 を作成して更新し、カラム uid_col の現在の値が NULL の場合、カラムの値を NEWID 関数で生成された一意な識別子に設定します。

```
CREATE TABLE t1 (uid_col int);
UPDATE t1
SET uid_col = NEWID()
WHERE uid_col IS NULL
```

次の文を実行します。

```
SELECT NEWID()
```

一意な識別子は BINARY(16) として返されます。たとえば、値は 0xd3749fe09cf446e399913bc6434f1f08 になります。この文字列を UUIDTOSTR() 関数を使用して読みやすい形式に変換できます。

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 使用法 | <p>NEWID() 関数は一意な識別子の値を生成します。</p> <p>UUID を使うと、データベース内のオブジェクトを一意に識別することができます。値は、1 台のコンピュータで生成された値が別のコンピュータで生成された値と一致しないように生成されます。このため、複写や同期の環境でキーとして使用することが可能です。</p> <p>NEWID 関数は、次の位置でのみサポートされます。</p> <ul style="list-style-type: none"> • 上位レベルのクエリ・ブロックの SELECT リスト • UPDATE 文の SET 句 • INSERT...VALUES の VALUES 句 <p>NEWID 関数によって生成された値は、Sybase IQ テーブルのカラムのデフォルト値として使用できます。</p> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • SQL99 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | <p>「STRTOUUID 関数 [文字列]」 (264 ページ)</p> <p>「UUIDTOSTR 関数 [文字列]」 (330 ページ)</p> <p>「バイナリ・データ型」 (77 ページ)</p> <p>「文字データ型」 (69 ページ)</p> |

NEXT_CONNECTION 関数 [システム]

| | |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 次の接続番号を返します。パラメータが NULL の場合は、最初の接続番号を返します。 |
| 構文 | NEXT_CONNECTION ({NULL <i>connection-id</i> }) |
| | <hr/> <p>注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「第 3 章 クエリと削除の最適化」の「SQL Anywhere による処理を引き起こす条件」を参照してください。</p> <hr/> |
| パラメータ | <p>connection-id 整数値。通常は、前回の NEXT_CONNECTION 呼び出しで返された値。<i>connection-id</i> が NULL の場合、NEXT_CONNECTION は最初の接続 ID を返します。</p> |

例 次の文は、最初の接続の ID を返します。ID は 569851433 などの整数値です。

```
SELECT NEXT_CONNECTION( NULL ) FROM iq_dummy
```

次の文は、1661140050 などの値を返します。

```
SELECT NEXT_CONNECTION( 569851433 ) FROM iq_dummy
```

使用法 NEXT_CONNECTION を使用して、データベースへの接続を列挙します。NULL を指定して最初の接続を取得し、前回の戻り値を指定して後続の各接続を取得します。それ以上接続がなくなると、NULL が返されます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

NEXT_DATABASE 関数 [システム]

機能 次のデータベース ID 番号を返します。パラメータが NULL の場合は、最初のデータベースを返します。

構文 **NEXT_DATABASE ({ NULL | database-id })**

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されません。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **database-id** データベースの ID 番号を指定する整数。

例 次の文は、最初のデータベースの値である 0 を返します。

```
SELECT NEXT_DATABASE( NULL ) FROM iq_dummy
```

次の文は NULL を返します。これは、サーバにそれ以上データベースが存在しないことを示します。

```
SELECT NEXT_DATABASE( 0 ) FROM iq_dummy
```

使用法 NEXT_DATABASE を使用すると、データベース・サーバ上で動作しているデータベースを列挙することができます。NULL を指定して最初のデータベースを取得し、前回の戻り値を指定して後続の各データベースを取得します。それ以上データベースがなくなると、NULL が返されます。

- 標準と互換性
- **SQL92** Transact SQL の拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。
- 参照
- 「[OBJECT_ID 関数 \[システム\]](#)」(216 ページ)

NEXT_HTTP_HEADER 関数 [HTTP]

機能 次の HTTP ヘッダ名を取得します。

構文 **NEXT_HTTP_HEADER** (*header-name*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **header-name** 前回のヘッダの名前。header-name に null を指定すると、最初の HTTP ヘッダの名前が返されます。

使用法 この関数は、要求内に含まれる HTTP ヘッダを繰り返し処理します。NULL を指定して呼び出すと、最初のヘッダの名前が返されます。前回のヘッダの名前を指定すると、後続のヘッダが取得できます。最後のヘッダの名前を指定して呼び出すと、NULL が返されます。

この関数を繰り返し呼び出すことにより、すべてのヘッダ・フィールドを確実に一度で取得できます。ただし、必ずしも HTTP 要求に記述された順序どおりに取得されるとは限りません。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

NEXT_HTTP_VARIABLE 関数 [HTTP]

機能 次の HTTP の変数名を取得します。

構文 **NEXT_HTTP_VARIABLE** (*var-name*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

| | |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | var-name 前回の変数の名前。var-name に null を指定すると、最初の HTTP 変数の名前が返されます。 |
| 使用法 | この関数は、要求内に含まれる HTTP 変数を繰り返し処理します。NULL を指定して呼び出すと、最初の変数の名前が返されます。前回の変数の名前を指定すると、後続の変数が取得できます。最後の変数の名前を指定して呼び出すと、NULL が返されます。 この関数を繰り返し呼び出すことにより、すべての変数を確実に一度で取得できます。ただし、必ずしも HTTP 要求に記述された順序どおりに取得されるとは限りません。URL PATH が ON に設定されていれば url が含まれ、ELEMENTS に設定されていれば変数 url1、url2 … url10 が含まれます。 |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | <p>「HTML_DECODE 関数 [HTTP]」 (182 ページ)</p> <p>「HTTP_VARIABLE 関数 [HTTP]」 (188 ページ)</p> <p>「NEXT_HTTP_HEADER 関数 [HTTP]」 (211 ページ)</p> |

NOW 関数 [日付および時刻]

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 現在の日付と時刻を返します。これは CURRENT_TIMESTAMP に対応する古い構文です。 |
| 構文 | NOW (*) |
| 例 | 次の文は、現在の日付および時刻を返します。 <pre>SELECT NOW(*) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise ではサポートされていません。 |

NTILE 関数 [統計]

| | |
|-------|------------------------------------------------------------------------------------------------------------------------|
| 機能 | クエリ結果を指定された数のバケットに分散させ、バケット内の各ローにバケット番号を割り当てます。 |
| 構文 | NTILE (<i>expression1</i>) OVER (ORDER BY <i>expression2</i> [ASC DESC]) |
| パラメータ | expression1 バケットの数を指定する、1 ~ 32767 の定数。 |

expression2 ソートを指定します。カラムの参照、集合関数、またはこれらの項目を起動する式など、有効な式を何でも指定できます。

例

次に、NTILE 関数を使用してカー・ディーラの販売状況を調査する例を示します。販売した車の数に基づいて、ディーラが4つのグループに分類されています。ntile = 1 になっているのは、車の販売台数で上位25% までのディーラです。

```
SELECT dealer_name, sales,
       NTILE(4) OVER ( ORDER BY sales DESC )
FROM carSales;
```

| dealer_name | sales | ntile |
|-------------|-------|-------|
| Boston | 1000 | 1 |
| Worcester | 950 | 1 |
| Providence | 950 | 1 |
| SF | 940 | 1 |
| Lowell | 900 | 2 |
| Seattle | 900 | 2 |
| Natick | 870 | 2 |
| New Haven | 850 | 2 |
| Portland | 800 | 3 |
| Houston | 780 | 3 |
| Hartford | 780 | 3 |
| Dublin | 750 | 3 |
| Austin | 650 | 4 |
| Dallas | 640 | 4 |
| Dover | 600 | 4 |

販売台数で上位10%のカー・ディーラを調べるには、この例のSELECT文にNTILE(10)と指定します。同様に、販売台数で50%の販売店を調べるには、NTILE(2)を指定します。

使用法

NTILE はクエリ結果を指定された数のバケットに分散させ、バケット内の各ローにバケット番号を割り当てるランク付け統計関数です。結果セットを100分位(パーセンタイル)、10分位(ディサイル)、4分位(クォータイル)、またはその他の数のグループに分類することができます。

NTILE には、OVER (ORDER BY) 句を指定する必要があります。ORDER BY 句は、ランク付けを実行するパラメータ、および各グループでローをソートする順序を指定します。このORDER BY 句は、OVER 句の内部でのみ使用するもので、SELECT のORDER BY とは異なることに注意してください。ランク付けクエリ内の集合関数にDISTINCTを指定することはできません。

OVER 句は、関数がクエリの結果セットを操作することを示します。結果セットとは、FROM、WHERE、GROUP BY、HAVING 句がすべて評価された後に返されるローを指します。OVER 句には、ランク付け統計関数の計算の対象となるローのデータ・セットを定義します。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

NTILE が使用できるのは、SELECT 文、INSERT 文、または SELECT 文の ORDER BY 句にある select リストのみです。NTILE は、ビューまたは union に含めることができます。NTILE 関数は、サブクエリ、HAVING 句、および UPDATE または DELETE 文の select リストでは使用できません。1つのクエリで使用可能な NTILE 関数は、1つだけです。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** および **SQL Anywhere** ではサポートされていません。

参照

「[分析関数](#)」(106 ページ)
『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

NULLIF 関数 [その他]

機能

式を比較する、CASE 式の省略形として使用できます。

構文

NULLIF (*expression1*, *expression2*)

パラメータ

expression1 比較される式。

expression2 比較される式。

例

次の文は、a を返します。

```
SELECT NULLIF( 'a', 'b' ) FROM iq_dummy
```

次の文は、NULL を返します。

```
SELECT NULLIF( 'a', 'a' ) FROM iq_dummy
```

使用法

NULLIF は、2つの式の値を比較します。

最初の式と次の式が一致する場合、NULLIF は NULL を返します。

最初の式が次の式と一致しない場合、または次の式が NULL の場合、NULLIF は最初の式を返します。

NULLIF 関数を使うと、CASE 式を手短かに書くことができます。NULLIF は次と同等です。

```
CASE WHEN expression1 = expression2 THEN NULL
ELSE expression1 END
```

標準と互換性

- **SQL92** Transact SQL の拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

[「CASE 式」\(31 ページ\)](#)

NUMBER 関数 [その他]

機能

1 で始まる数を、クエリの結果内にある連続したローのすべてに生成します。

構文

```
NUMBER ( * )
```

例

次の文は、番号の付いたリストを返します。

```
number(*)
```

```
1
2
3
4
5
```

```
SELECT NUMBER( * )
FROM Departments
WHERE DepartmentID > 10
```

使用法

NUMBER 関数は、UPDATE 文の select リストまたは SET 句の中でのみ使用してください。たとえば、次の文は seq_id カラムの各ローに対して直前のローより 1 大きい値で更新します。番号は、ORDER BY 句で指定された順番に適用されます。

```
update empl
set seq_id = number(*)
order by empl_id
```

UPDATE 文で NUMBER(*) 関数を SET 句に使用し、FROM 句に 1 対多のジョインが指定されている場合、NUMBER(*) は増加する一意な数値を生成しますが、ローが削除されるため、連続的には増加しません。

SELECT 文からの INSERT を使用する際に、NUMBER を使用してプライマリ・キーを生成することも可能ですが、連続的なプライマリ・キーを生成するために、IDENTITY/AUTOINCREMENT を使用することをお奨めします。

注意 NUMBER を DELETE 文、WHERE 句、HAVING 句、ORDER BY 句、サブクエリ、集合に関連するクエリ、制約、GROUP BY、DISTINCT、UNION ALL を含むクエリ、または派生テーブルで使用すると、構文エラーが発生します。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

OBJECT_ID 関数 [システム]

機能 オブジェクト ID を返します。

構文 **OBJECT_ID** (*object-name*)

パラメータ **object-name** オブジェクト名。

例 次の文は、*Customers* テーブルのオブジェクト ID である 100209 を返します。

```
SELECT OBJECT_ID ('CUSTOMERS') FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。

参照

[「COL_NAME 関数 \[システム\]」 \(139 ページ\)](#)

[「DB_ID 関数 \[システム\]」 \(162 ページ\)](#)

[「OBJECT_NAME 関数 \[システム\]」 \(216 ページ\)](#)

OBJECT_NAME 関数 [システム]

機能 オブジェクト名を返します。

構文 **OBJECT_NAME** (*object-id* [, *database-id*])

パラメータ **object id** オブジェクト ID。

database-id データベース ID。

| | |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 例 | 次の文は、名前 “customer” を返します。 <pre>SELECT OBJECT_NAME (100209) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。 |
| 参照 | <p>「COL_NAME 関数 [システム]」 (139 ページ)</p> <p>「DB_NAME 関数 [システム]」 (162 ページ)</p> <p>「OBJECT_ID 関数 [システム]」 (216 ページ)</p> |

OCTET_LENGTH 関数 [文字列]

| | |
|--------|---------------------------------------------------------------------------------|
| 機能 | カラムのバイト長を保持する、符号なしの 64 ビット値を返します。 |
| 構文 | OCTET_LENGTH (<i>column-name</i>) |
| パラメータ | column-name カラムの名前。 |
| 使用法 | 引数が NULL の場合は、NULL 値を返します。 OCTET_LENGTH 関数は、すべての Sybase IQ データ型をサポートします。 |
| 標準と互換性 | Sybase SQL Anywhere および Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | 「BIT_LENGTH 関数 [文字列]」 (133 ページ) |

PATINDEX 関数 [文字列]

| | |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定したパターンが最初に検出された先頭位置を返します。 |
| 構文 | PATINDEX ('% <i>pattern</i> %', <i>string-expression</i>) |
| パラメータ | pattern 検索するパターン。この文字列は、ワイルドカードを含むパターンの 126 バイトに制限されています。最初の % を省略すると、PATINDEX ではパターンが文字列の最初に出現する場合は 1 が、そうでない場合は 0 が返されます。 <i>pattern</i> の先頭の文字が % の場合は、% を 2 つ続けると 1 つとして処理されます。 パターンは LIKE 比較と同じワイルドカードを使用します。表 4-25 にパターンのワイルドカードを示します。 |

表 4-25 : PATINDEX でのパターンのワイルドカード

| ワイルドカード | 一致条件 |
|-------------|---------------------------|
| _ (アンダースコア) | 任意の 1 文字 |
| % (パーセント記号) | 0 個以上の文字からなる任意の文字列 |
| [] | 指定した範囲または文字セット内の任意の 1 文字 |
| [^] | 指定した範囲または文字セット以外の任意の 1 文字 |

ワイルドカードを含まないパターン (パーセント % またはアンダースコア _) の最大長は、255 バイトです。

string-expression パターンが検索される文字列。

例

次の文は、値 2 を返します。

```
SELECT PATINDEX( '%hoco%', 'chocolate' ) FROM iq_dummy
```

次の文は、値 11 を返します。

```
SELECT PATINDEX ( '%4_5_', '0a1A 2a3A 4a5A' ) FROM
iq_dummy
```

使用法

PATINDEX は、パターンが最初に検出された先頭位置を返します。文字列パターンが検索対象の文字列に 2 つ以上含まれる場合は、最初の文字列の位置だけが返されます。

パターンが検出されなければ、ゼロ (0) が返されます。

長さが 126 バイトを超えるパターンを検索すると、NULL が返されます。

長さが 0 の文字列を検索すると、1 が返されます。

引数のどれか 1 つでも NULL の場合、結果はゼロ (0) になります。

PATINDEX 関数で返されるか指定される位置またはオフセットはすべて、常に文字オフセットで、マルチバイト・データの場合はバイト・オフセットとは異なることがあります。

PATINDEX は CHAR カラムと VARCHAR カラムの場合、32 ビット符号なし整数で位置を返します。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

参照

[「LIKE 条件」 \(42 ページ\)](#)

[「LOCATE 関数 \[文字列\]」 \(199 ページ\)](#)

『Sybase IQ によるラージ・オブジェクト管理』の「[第 4 章 関数のサポート](#)」

PERCENT_RANK 関数 [統計]

機能 ORDER BY 句の定義に従い、クエリから返される 1 つのローの、クエリから返されるその他のローに対する (小数による) 位置を返します。0 ～ 1 の 10 進値を返します。

構文 **PERCENT_RANK () OVER (ORDER BY *expression* [ASC | DESC])**

パラメータ **expression** ソートを指定します。カラムの参照、集合関数、またはこれらの項目を起動する式など、有効な式を何でも指定できます。

例 PERCENT_RANK 関数は、次の文のように使用します。

```
SELECT s_suppkey, SUM(s_acctBal) AS sum_acctBal,
PERCENT_RANK() OVER ( ORDER BY SUM(s_acctBal) DESC )
AS percent_rank_all FROM supplier GROUP BY s_suppkey;
```

| s_suppkey | sum_acctBal | percent_rank_all |
|--------------|-------------|------------------|
| supplier#011 | 200000 | 0 |
| supplier#002 | 200000 | 0 |
| supplier#013 | 123000 | 0.3333 |
| supplier#004 | 110000 | 0.5 |
| supplier#035 | 110000 | 0.5 |
| supplier#006 | 50000 | 0.8333 |
| supplier#021 | 10000 | 1 |

使用法 PERCENT_RANK はランク付け統計関数です。ロー *R* のパーセント・ランクは、OVER 句に指定された各グループ内でのローのランクから 1 を引いた値を、OVER 句に指定された各グループのローの合計数から 1 を引いた数で割って算出します。PERCENT_RANK は 0 ～ 1 の値を返します。最初のローのパーセント・ランクはゼロになります。

ローの PERCENT_RANK は、以下のように計算されます。

$$(R_x - 1) / (N_{totalRow} - 1)$$

R_x はグループのローのランク位置で、*N_{totalRow}* は OVER 句に指定されたグループでのローの合計数です。

PERCENT_RANK には、OVER (ORDER BY) 句を指定する必要があります。ORDER BY 句は、ランク付けを実行するパラメータ、および各グループでローをソートする順序を指定します。この ORDER BY 句は、OVER 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。ランク付けクエリ内の集合関数に DISTINCT を指定することはできません。

OVER 句は、関数がクエリの結果セットを操作することを示します。結果セットとは、FROM、WHERE、GROUP BY、HAVING 句がすべて評価された後に返されるローを指します。OVER 句には、ランク付け統計関数の計算の対象となるローのデータ・セットを定義します。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

PERCENT_RANK が使用できるのは、SELECT 文、INSERT 文、または SELECT 文の ORDER BY 句にある select リストのみです。

PERCENT_RANK は、ビューまたは union に含めることができます。

PERCENT_RANK 関数は、サブクエリ、HAVING 句、および UPDATE または DELETE 文の select リストでは使用できません。1つのクエリで使用可能なランク付け統計関数は、1つだけです。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise および SQL Anywhere ではサポートされていません。

参照

[「分析関数」\(106 ページ\)](#)

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

PERCENTILE_CONT 関数 [統計]

機能

指定されたパーセンタイルから、対応する値を返します。連続分布データ・モデルを前提としています。

注意 パーセンタイルを計算するだけであれば、NTILE 関数に値 100 を指定して使用してください。

構文

```
PERCENTILE_CONT ( expression1 )  
WITHIN GROUP ( ORDER BY expression2 [ ASC | DESC ] )
```

パラメータ

expression1 数値データ型の定数を、0 以上 1 以下で指定します。引数が NULL であれば、“wrong argument for percentile” エラーが返ります。引数の値が 0 よりも小さいか、1 よりも大きい場合は、“data value out of range” エラーが返ります。

expression2 ソートを指定します。カラム参照を含む単一の式で指定してください。このソート式に、複数の式やランク付け統計関数、set 関数、またはサブクエリを指定することはできません。

例

次の例は PERCENTILE_CONT 関数を使用し、各地域の自動車販売の 10 番目のパーセンタイル値を調べています。

この例では、次のデータ・セットが使用されています。

| sales | region | dealer_name |
|-------|-----------|-------------|
| 900 | Northeast | Boston |
| 800 | Northeast | Worcester |
| 800 | Northeast | Providence |
| 700 | Northeast | Lowell |
| 540 | Northeast | Natick |
| 500 | Northeast | New Haven |
| 450 | Northeast | Hartford |
| 800 | Northwest | SF |
| 600 | Northwest | Seattle |
| 500 | Northwest | Portland |
| 400 | Northwest | Dublin |
| 500 | South | Houston |
| 400 | South | Austin |
| 300 | South | Dallas |
| 200 | South | Dover |

次の SELECT 文には PERCENTILE_CONT 関数が含まれています。

```
SELECT region, PERCENTILE_CONT(0.1)
WITHIN GROUP ( ORDER BY sales DESC )
FROM carSales GROUP BY region;
```

この SELECT 文の結果には、地域の自動車販売の 10 番目のパーセンタイル値が一覧表示されます。

| region | percentile_cont |
|-----------|-----------------|
| Northeast | 840 |
| Northwest | 740 |
| South | 470 |

使用法

逆分散統計関数は、K-理論パーセンタイル値を返します。これは、ひとまとまりのデータの値として許容し得るしきい値を決定する際に使用します。PERCENTILE_CONT 関数は、パーセンタイル値を引数として受け取り、WITHIN GROUP 句で指定されたデータ・グループか、データ・セット全体に対して処理を実行します。グループごとに1つの値を返し、クエリの GROUP BY に指定したカラムが存在しなければ、結果は単一のローになります。結果のデータ型は、WITHIN GROUP 句に指定した ORDER BY の項目のデータ型と同じになります。

PERCENTILE_CONT の ORDER BY 式のデータ型は、数値型である必要があります。

PERCENTILE_CONT には、WITHIN GROUP (ORDER BY) 句を指定する必要があります。

必須の ORDER BY には、パーセンタイル関数の実行の対象となる式と、各グループ内でのローのソート順を指定します。PERCENTILE_CONT 関数の場合、この式のデータ型は数値型であることが必要です。この ORDER BY 句は、WITHIN GROUP 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。

WITHIN GROUP 句は、クエリ結果を順序付けられたデータ・セットに分類します。関数はこのデータ・セットに基づいて結果を計算します。WITHIN GROUP 句には、単一のソート項目を含めてください。WITHIN GROUP 句に指定されたソート項目が1つより多い(または少ない)場合は、エラーが報告されます。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

PERCENTILE_CONT 関数は、サブクエリ、HAVING 句、ビュー、union で使用できます。PERCENTILE_CONT は、統計を行わない単純な集合関数が使用される場所であれば、どこでも使用できます。

PERCENTILE_CONT 関数では、データ・セット内の NULL 値は無視されます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise および SQL Anywhere ではサポートされていません。

参照

[「分析関数」\(106 ページ\)](#)

[「NTILE 関数 \[統計\]」\(212 ページ\)](#)

[「PERCENTILE_DISC 関数 \[統計\]」\(223 ページ\)](#)

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

PERCENTILE_DISC 関数 [統計]

機能 指定されたパーセンタイルから、対応する値を返します。離散分布データ・モデルを前提としています。

注意 パーセンタイルを計算するだけであれば、NTILE 関数に値 100 を指定して使用してください。

構文

```
PERCENTILE_DISC ( expression1 )
WITHIN GROUP ( ORDER BY expression2 [ ASC | DESC ] )
```

パラメータ

expression1 数値データ型の定数を、0 以上 1 以下で指定します。引数が NULL であれば、“wrong argument for percentile” エラーが返ります。引数の値が 0 よりも小さいか、1 よりも大きい場合は、“data value out of range” エラーが返ります。

expression2 ソートを指定します。カラム参照を含む単一の式で指定してください。このソート式に、複数の式やランク付け統計関数、set 関数、またはサブクエリを指定することはできません。

例

次の例は PERCENTILE_DISC 関数を使用し、各地域の自動車販売の 10 番目のパーセンタイル値を調べています。

この例では、次のデータ・セットが使用されています。

| sales | region | dealer_name |
|-------|-----------|-------------|
| 900 | Northeast | Boston |
| 800 | Northeast | Worcester |
| 800 | Northeast | Providence |
| 700 | Northeast | Lowell |
| 540 | Northeast | Natick |
| 500 | Northeast | New Haven |
| 450 | Northeast | Hartford |
| 800 | Northwest | SF |
| 600 | Northwest | Seattle |
| 500 | Northwest | Portland |
| 400 | Northwest | Dublin |
| 500 | South | Houston |
| 400 | South | Austin |
| 300 | South | Dallas |
| 200 | South | Dover |

次の SELECT 文には PERCENTILE_DISC 関数が含まれています。

```
SELECT region, PERCENTILE_DISC (0.1)
WITHIN GROUP ( ORDER BY sales DESC )
FROM carSales GROUP BY region;
```

この SELECT 文の結果には、地域の自動車販売の 10 番目のパーセンタイル値が一覧表示されます。

| region | percentile_cont |
|-----------|-----------------|
| Northeast | 900 |
| Northwest | 800 |
| South | 500 |

使用法

逆分散統計関数は、K-理論パーセンタイル値を返します。これは、ひとまとまりのデータの値として許容し得るしきい値を決定する際に使用します。PERCENTILE_DISC 関数は、パーセンタイル値を引数として受け取り、WITHIN GROUP 句で指定されたデータ・グループか、データ・セット全体に対して処理を実行します。グループごとに 1 つの値を返し、クエリの GROUP BY に指定したカラムが存在しなければ、結果は単一のローになります。結果のデータ型は、WITHIN GROUP 句に指定した ORDER BY の項目のデータ型と同じになります。

PERCENTILE_DISC は、Sybase IQ でソート可能なすべてのデータ型をサポートします。

PERCENTILE_DISC には、WITHIN GROUP (ORDER BY) 句を指定する必要があります。

必須の ORDER BY には、パーセンタイル関数の実行の対象となる式と、各グループ内でのローのソート順を指定します。この ORDER BY 句は、WITHIN GROUP 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。

WITHIN GROUP 句は、クエリ結果を順序付けられたデータ・セットに分類します。関数はこのデータ・セットに基づいて結果を計算します。WITHIN GROUP 句には、単一のソート項目を含めてください。WITHIN GROUP 句に指定されたソート項目が 1 つより多い (または少ない) 場合は、エラーが報告されます。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

PERCENTILE_DISC 関数は、サブクエリ、HAVING 句、ビュー、union で使用できます。PERCENTILE_DISC は、分析を行わない単純な集合関数で使用されるのであれば、どこでも使用できます。

PERCENTILE_DISC 関数では、データ・セット内の NULL 値は無視されます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise および SQL Anywhere ではサポートされていません。

- 参照
- 「分析関数」(106 ページ)
 - 「NTILE 関数 [統計]」(212 ページ)
 - 「PERCENTILE_CONT 関数 [統計]」(220 ページ)
- 『システム管理ガイド 第2巻』の「第2章 OLAP の使用」

PI 関数 [数値]

- 機能
- 円周率の値を数値で返します。
- 構文
- PI (*)**
- 例
- 次の文は、値 3.141592653... を返します。
- ```
SELECT PI (*) FROM iq_dummy
```
- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** PI() 関数は Adaptive Server Enterprise でサポートされていますが、PI(\*) はサポートされていません。

## POWER 関数 [数値]

- 機能
- ある数値を、別の数値で累乗します。
- 構文
- POWER ( numeric-expression1, numeric-expression2 )**
- パラメータ
- numeric-expression1** 底。
- numeric-expression2** 指数。
- 例
- 次の文は、値 64 を返します。
- ```
SELECT Power ( 2, 6 ) FROM iq_dummy
```
- 使用法
- numeric-expression1* を *numeric-expression2* で累乗します。
- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。

PROPERTY 関数 [システム]

- 機能
- 指定されたサーバ・レベル・プロパティの値を文字列で返します。

| | |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | PROPERTY ({ <i>property-id</i> <i>property-name</i> }) |
| | <hr/> 注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「 第3章 クエリと削除の最適化 」の「 SQL Anywhere による処理を引き起こす条件 」を参照してください。 <hr/> |
| パラメータ | property-id サーバ・レベル・プロパティのプロパティ番号である整数。この数値は PROPERTY_NUMBER 関数で調べることができます。プロパティのセットを繰り返し処理する場合は、 <i>property-id</i> がよく使用されます。 property-name プロパティ名を指定する文字列。サーバのプロパティ名の一覧は、「 サーバで使用可能なプロパティ 」(122 ページ)の項を参照してください。 |
| 例 | 次の文は、現在のデータベース・サーバの名前を返します。 <pre>SELECT PROPERTY('Name') FROM iq_dummy</pre> |
| 使用法 | 各プロパティには名前と番号がありますが、番号はバージョンごとに変わる場合があります。よって、プロパティを指定する際、この番号を信頼できる識別子として使用しないでください。 |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 ベンダの拡張機能。• Sybase Adaptive Server Enterprise ではサポートされていません。 |
| 参照 | 「サーバで使用可能なプロパティ」 (122 ページ) |

PROPERTY_DESCRIPTION 関数 [システム]

| | |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | プロパティの説明を返します。 |
| 構文 | PROPERTY_DESCRIPTION ({ <i>property-id</i> <i>property-name</i> }) |
| | <hr/> 注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「 第3章 クエリと削除の最適化 」の「 SQL Anywhere による処理を引き起こす条件 」を参照してください。 <hr/> |
| パラメータ | property-id プロパティの番号である整数。この数値は PROPERTY_NUMBER 関数で調べることができます。プロパティのセットを繰り返し処理する場合は、 <i>property-id</i> がよく使用されます。 |

property-name プロパティ名を指定する文字列。プロパティ名については、「[接続プロパティ](#)」(121 ページ)、「[サーバで使用可能なプロパティ](#)」(122 ページ)、「[各データベースで使用できるプロパティ](#)」(122 ページ)の項にあるリストを参照してください。

例 次の文は、「インデックス挿入数」という説明を返します。

```
SELECT PROPERTY_DESCRIPTION( 'IndAdd' ) FROM iq_dummy
```

使用法 各プロパティには名前と番号がありますが、番号はリリースごとに変わる場合があります。よって、プロパティを指定する際、この番号を信頼できる識別子として使用しないでください。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照 [「接続プロパティ」](#) (121 ページ)
[「サーバで使用可能なプロパティ」](#) (122 ページ)
[「各データベースで使用できるプロパティ」](#) (122 ページ)

PROPERTY_NAME 関数 [システム]

機能 指定されたプロパティ番号のプロパティ名を返します。

構文 **PROPERTY_NAME** (*property-id*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **property-id** プロパティのプロパティ番号。

例 次の文は、プロパティ番号 126 のプロパティを返します。この番号で参照される実際のプロパティは、バージョンの種類によって異なります。

```
SELECT PROPERTY_NAME( 126 ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照 [「接続プロパティ」](#) (121 ページ)
[「サーバで使用可能なプロパティ」](#) (122 ページ)
[「各データベースで使用できるプロパティ」](#) (122 ページ)

PROPERTY_NUMBER 関数 [システム]

機能 指定されたプロパティ名のプロパティ番号を返します。

構文 **PROPERTY_NUMBER** (*property-name*)

注意 CIS 機能補正のパフォーマンスに関する考慮事項が適用されません。『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。

パラメータ **property-name** プロパティ名。プロパティ名については、「[接続プロパティ](#)」(121 ページ)、「[サーバで使用可能なプロパティ](#)」(122 ページ)、「[各データベースで使用できるプロパティ](#)」(122 ページ)の項にあるリストを参照してください。

例 次の文は、整数の値を返します。実際の値は、バージョンの種類によって異なります。

```
SELECT PROPERTY_NUMBER( 'PAGESIZE' ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

参照

「[接続プロパティ](#)」(121 ページ)

「[サーバで使用可能なプロパティ](#)」(122 ページ)

「[各データベースで使用できるプロパティ](#)」(122 ページ)

QUARTER 関数 [日付および時刻]

機能 指定された日付式が、その年のどの四半期にあるかを示す数値を返します。

構文 **QUARTER**(*date-expression*)

パラメータ **date-expression** 日付。

例 DATE_ORDER オプションがデフォルトの *ymd* に設定されている場合、次の文は、値 2 を返します。

```
SELECT QUARTER ( '1987/05/02' ) FROM iq_dummy
```

使用法

[表 4-26](#) は、1 年の各四半期に含まれる日付を示します。

表 4-26 : 1 年の四半期の値

| Quarter | 期間 |
|---------|----------------------|
| 1 | 1 月 1 日 ~ 3 月 31 日 |
| 2 | 4 月 1 日 ~ 6 月 30 日 |
| 3 | 7 月 1 日 ~ 9 月 30 日 |
| 4 | 10 月 1 日 ~ 12 月 31 日 |

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

参照

『リファレンス：文とオプション』の「[DATE_ORDER オプション](#)」

RADIANS 関数 [数値]

機能

数値を度からラジアンに変換します。

構文

RADIANS (*numeric-expression*)

パラメータ

numeric-expression 数値 (度)。この角度がラジアンに変換されます。

例

次の文は、約 0.5236 の値を返します。

```
SELECT RADIANS( 30 ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** 互換。

RAND 関数 [数値]

機能

$0 \leq x < 1$ の範囲の乱数 x を **DOUBLE** 型の精度で返します。オプションでシードを使用できます。

構文

RAND ([*integer-expression*])

パラメータ

integer-expression 乱数の生成に使用されるオプションのシード。この引数を指定すると、再生可能な乱数シーケンスを作成できます。

RAND が **FROM** 句および **IQ** ストアのテーブルのみを含むクエリの引数を使用して呼び出されると、任意の繰り返し可能な値が返されます。

引数が呼び出されない場合、**RAND** は非決定性関数になります。**RAND** を連続して呼び出すと、異なる値を返す場合があります。クエリ・オプティマイザは、**RAND** 関数の結果をキャッシュしません。

注意 RAND から返される値は、FROM 句を使用しているかどうか、および参照されているテーブルが SYSTEM または IQ ストアのどちらで作成されているかによって変わります。

例 次の文は、テーブルの 5% のサンプリングを返します。

```
SELECT AVG(table1.number_of_cars),
       AVG(table1.number_of_tvcs)FROM table1 WHERE
       RAND(ROWID(table1)) < .05 and table1.income < 50000;
```

次の文は、約 941392926249216914 の値を返します。

```
SELECT RAND( 4 ) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

RANK 関数 [統計]

機能 グループ内の項目をランク付けします。

構文 **RANK () OVER (ORDER BY *expression* [ASC | DESC])**

パラメータ **expression** ソートを指定します。カラムの参照、集合関数、またはこれらの項目を起動する式など、有効な式を何でも指定できます。

例 RANK 関数は、次の文のように使用します。

```
SELECT s_suppkey, SUM(s_acctBal) AS sum_acctBal,
       RANK() OVER ( ORDER BY SUM(s_acctBal) DESC )
       AS rank_all FROM supplier GROUP BY s_suppkey;
```

| s_suppkey | sum_acctBal | rank_all |
|--------------|-------------|----------|
| supplier#011 | 200000 | 1 |
| supplier#002 | 200000 | 1 |
| supplier#013 | 123000 | 3 |
| supplier#004 | 110000 | 4 |
| supplier#035 | 110000 | 4 |
| supplier#006 | 50000 | 6 |
| supplier#021 | 10000 | 7 |

使用法

RANK はランク付け統計関数です。ロー R のランクは、R 以前にあり R と同等でないローの数で決まります。OVER 句で指定されたグループどうしで、2 つ以上のローが同等な場合、または結果セット全体で同等な場合は、ランク付けの順番に 1 つ以上の隔たりが生じます。RANK と DENSE_RANK では、順位が同じである場合に、DENSE_RANK はランク順に隔たりを置かない点が異なります。RANK では隔たりが置かれます。

RANK には、OVER (ORDER BY) 句を指定する必要があります。ORDER BY 句は、ランク付けを実行するパラメータ、および各グループでローをソートする順序を指定します。この ORDER BY 句は、OVER 句の内部でのみ使用するもので、SELECT の ORDER BY とは異なります。ランク付けクエリ内の集合関数に DISTINCT を指定することはできません。

OVER 句は、関数がクエリの結果セットを操作することを示します。結果セットとは、FROM、WHERE、GROUP BY、HAVING 句がすべて評価された後に返されるローを指します。OVER 句には、ランク付け統計関数の計算の対象となるローのデータ・セットを定義します。

ASC と DESC のパラメータでは、昇順または降順の順序付けシーケンスを指定します。昇順がデフォルトです。

RANK が使用できるのは、SELECT 文、INSERT 文、または SELECT 文の ORDER BY 句にある select リストのみです。RANK は、ビューまたは union に含めることができます。RANK 関数は、サブクエリ、HAVING 句、および UPDATE または DELETE 文の select リストでは使用できません。1 つのクエリで使用可能なランク付け統計関数は、1 つだけです。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** および **SQL Anywhere** ではサポートされていません。

参照

[「分析関数」\(106 ページ\)](#)

[「DENSE_RANK 関数 \[統計\]」\(164 ページ\)](#)

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

REGR_AVGX 関数 [集合]

機能

回帰線の独立変数の平均を計算します。

構文 1

REGR_AVGX (*dependent-expression, independent-expression*)

構文 2

REGR_AVGX (*dependent-expression, independent-expression*)
OVER (*window-spec*)

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

parameter **dependent-expression** 独立した変数の影響を受ける変数。

parameter **independent-expression** 結果に影響を与える変数。

usage この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

関数は、**dependent-expression** または **independent-expression** が NULL のペアをすべて排除した後、(**dependent-expression** と **independent-expression**) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。NULL 値を排除した後、次の計算が行われます (x は **independent-expression** を示します)。

AVG (x)

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、従属変数 (従業員の年齢) の平均を計算します。

```
SELECT REGR_AVGX( Salary, ( YEAR( NOW() ) - YEAR(
    BirthDate ) ) )
FROM Employees;
```

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。
- **Sybase** SQL Anywhere 互換。

REGR_AVGY 関数 [集合]

機能 回帰線の従変数の平均を計算します。

構文 1 **REGR_AVGY**(*dependent-expression*, *independent-expression*)

構文 2 **REGR_AVGY**(*dependent-expression*, *independent-expression*)
OVER (*window-spec*)

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ **dependent-expression** 独立した変数の影響を受ける変数。

independent-expression 結果に影響を与える変数。

使用法 この関数は、引数を **DOUBLE** に変換し、倍精度浮動小数点の計算を行って、**DOUBLE** を結果として返します。空のセットが指定された場合、**NULL** が返されます。

関数は、*dependent-expression* または *independent-expression* が **NULL** のペアをすべて排除した後、(*dependent-expression* と *independent-expression*) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。**NULL** 値を排除した後、次の計算が行われます (*y* は *dependent-expression* を示します)。

AVG(y)

『SQL Anywhere サーバ — SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 **ROLLUP** と **CUBE** は、構文 1 の **GROUP BY** 句でサポートされていません。**DISTINCT** はサポートされていません。

構文 2 は、**SELECT** 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、独立変数 (従業員の給与) の平均を計算します。この関数は、値 49988.6232 を返します。

```
SELECT REGR_AVGY( Salary, ( YEAR( NOW( ) ) - YEAR(
BirthDate ) ) )
FROM Employees;
```

標準と互換性 • **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。

 • **Sybase** SQL Anywhere 互換。

REGR_COUNT 関数 [集合]

機能 回帰線適合のために使用された非 NULL 値のペアの数を示す整数を返します。

構文 1 `REGR_COUNT(dependent-expression, independent-expression)`

構文 2 `REGR_COUNT(dependent-expression, independent-expression)
OVER (window-spec)`

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ **dependent-expression** 独立した変数の影響を受ける変数。

independent-expression 結果に影響を与える変数。

使用法 この関数は、結果として UNSIGNED BIGINT を返します。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、回帰直線の調整に使用される NULL 以外のペアの数を示す値を返します。この関数は、値 75 を返します。

```
SELECT REGR_COUNT( Salary, ( YEAR( NOW() ) - YEAR(
  BirthDate ) ) )
FROM Employees;
```

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。
- **Sybase** SQL Anywhere 互換。

REGR_INTERCEPT 関数 [集合]

機能 従変数と独立変数が最も適合する線形回帰線の y 切片を計算します。

構文 1 `REGR_INTERCEPT(dependent-expression, independent-expression)`

構文 2 `REGR_INTERCEPT(dependent-expression, independent-expression)
OVER (window-spec)`

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ **dependent-expression** 独立した変数の影響を受ける変数。
independent-expression 結果に影響を与える変数。

使用法 この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

関数は、*dependent-expression* または *independent-expression* が NULL のペアをすべて排除した後、(*dependent-expression* と *independent-expression*) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。NULL 値を排除した後、次の計算が行われます (*y* は *dependent-expression* を表し、*x* は *independent-expression* を表します)。

AVG(y) - REGR_SLOPE(y, x) * AVG(x)

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、値 1874.5805688517603 を返します。

```
SELECT REGR_INTERCEPT( Salary, ( YEAR( NOW() ) - YEAR(
BirthDate ) ) )
FROM Employees;
```

標準と互換性 • **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。

• **Sybase** SQL Anywhere 互換。

REGR_R2 関数 [集合]

機能 回帰線の決定係数 (R-squared または適合度統計とも呼ばれる) を計算します。

構文 1 `REGR_R2(dependent-expression, independent-expression)`

構文 2 `REGR_R2(dependent-expression, independent-expression)
OVER (window-spec)`

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ **dependent-expression** 独立した変数の影響を受ける変数。

independent-expression 結果に影響を与える変数。

使用法 この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

REGR_R2 関数は、*dependent-expression* または *independent-expression* が NULL のペアをすべて排除した後、(*dependent-expression* と *independent-expression*) ペアのセットに適用されます。次に、IQ は、次のアルゴリズムを適用します。

- $VAR_POP(x) = 0$ の場合、REGR_R2 は $VAR_POP(x)$ を計算して NULL を返します。 $VAR_POP(y) = 0$ の場合、 $VAR_POP(y)$ を計算して値 1 を返します。
- $VAR_POP(x)$ と $VAR_POP(y)$ がいずれも 0 でない場合、次の値が返されます。 $POWER(CORR(y,x),2)$

ここで、y は *dependent-expression* を表し、x は *independent-expression* を表します。

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例

次の例は、値 0.19379959710325653 を返します。

```
SELECT REGR_R2( Salary, ( YEAR( NOW() ) - YEAR(
BirthDate ) ) )
FROM Employees;
```

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。
- **Sybase** SQL Anywhere 互換。

REGR_SLOPE 関数 [集合]

機能

非 NULL 値のペアに適合された線形回帰線の傾斜を計算します。

構文 1

REGR_SLOPE(*dependent-expression*, *independent-expression*)

構文 2

REGR_SLOPE(*dependent-expression*, *independent-expression*)
OVER (*window-spec*)

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ

dependent-expression 独立した変数の影響を受ける変数。

independent-expression 結果に影響を与える変数。

使用法

この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

REGR_SLOPE 関数は、*dependent-expression* または *independent-expression* が NULL のペアをすべて排除した後、(*dependent-expression* と *independent-expression*) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。NULL 値を排除した後、次の計算が行われます (y は *dependent-expression* を表し、x は *independent-expression* を表します)。

COVAR_POP(x, y) / VAR_POP(y)

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、値 935.3429749445614 を返します。

```
SELECT REGR_SLOPE( Salary, ( YEAR( NOW() ) - YEAR(
  BirthDate ) ) )
FROM Employees;
```

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。
- **Sybase** SQL Anywhere 互換。

REGR_SXX 関数 [集合]

機能 非 NULL 値のペアに適合された線形回帰線の傾斜を計算します。

構文 1 **REGR_SXX**(*dependent-expression*, *independent-expression*)

構文 2 **REGR_SXX**(*dependent-expression*, *independent-expression*)
OVER (*window-spec*)

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ

dependent-expression 独立した変数の影響を受ける変数。

independent-expression 結果に影響を与える変数。

使用法

この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

関数は、*dependent-expression* または *independent-expression* が NULL のペアをすべて排除した後、(*dependent-expression* と *independent-expression*) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。NULL 値を排除した後、次の計算が行われます (y は *dependent-expression* を表し、x は *independent-expression* を表します)。

REGR_COUNT(y, x) * **VAR_POP**(x)

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例

次の例は、値 5916.4800000000105 を返します。

```
SELECT REGR_SXX( Salary, ( YEAR( NOW() ) - YEAR(
  BirthDate ) ) )
FROM Employees;
```

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。
- **Sybase** SQL Anywhere 互換。

REGR_SXY 関数 [集合]

機能

従変数および独立変数の積和を返します。REGR_SXY 関数は、回帰モデルの統計的な有効性を評価するときに使用できます。

構文 1

REGR_SXY(*dependent-expression*, *independent-expression*)

構文 2

REGR_SXY(*dependent-expression*, *independent-expression*)
OVER (*window-spec*)

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

パラメータ

dependent-expression 独立した変数の影響を受ける変数。

independent-expression 結果に影響を与える変数。

使用法

この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。

関数は、*dependent-expression* または *independent-expression* が NULL のペアをすべて排除した後、(*dependent-expression* と *independent-expression*) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。NULL 値を排除した後、次の計算が行われます (y は *dependent-expression* を表し、x は *independent-expression* を表します)。

REGR_COUNT(x, y) * COVAR_POP(x, y)

『SQL Anywhere サーバー SQL の使用法』の [Mathematical formulas for the aggregate functions \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。

注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。

構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、値 5533938.004400015 を返します。

```
SELECT REGR_SXY( Salary, ( YEAR( NOW() ) - YEAR(
  BirthDate ) ) )
FROM Employees;
```

標準と互換性

- **SQL2008** 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。
- **Sybase** SQL Anywhere 互換。

REGR_SYY 関数 [集合]

機能 回帰モデルの統計的な有効性を評価できる値を返します。

構文 1 **REGR_SYY**(*dependent-expression*, *independent-expression*)

構文 2 **REGR_SYY**(*dependent-expression*, *independent-expression*)
OVER (*window-spec*)

column-spec : 以下の「使用法」の構文 2 の手順を参照してください。

| | |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | <p>dependent-expression 独立した変数の影響を受ける変数。</p> <p>independent-expression 結果に影響を与える変数。</p> |
| 使用法 | <p>この関数は、引数を DOUBLE に変換し、倍精度浮動小数点の計算を行って、DOUBLE を結果として返します。空のセットが指定された場合、NULL が返されます。</p> <p>関数は、<i>dependent-expression</i> または <i>independent-expression</i> が NULL のペアをすべて排除した後、(<i>dependent-expression</i> と <i>independent-expression</i>) ペアのセットに適用されます。関数は、データを 1 回参照して同時に計算されます。NULL 値を排除した後、次の計算が行われます (y は <i>dependent-expression</i> を表し、x は <i>independent-expression</i> を表します)。</p> <p>REGR_COUNT(x, y) * VAR_POP(y)</p> <p>『SQL Anywhere サーバー SQL の使用法』の Mathematical formulas for the aggregate functions (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbusage_en11/ug-olap-s-6410540.html) を参照してください。</p> <hr/> <p>注意 ROLLUP と CUBE は、構文 1 の GROUP BY 句でサポートされていません。DISTINCT はサポートされていません。</p> <hr/> <p>構文 2 は、SELECT 文でのウィンドウ関数としての使用法を示します。その場合、関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで <i>window-spec</i> の要素を指定できます。ウィンドウの指定方法については、「分析関数」(106 ページ) を参照してください。</p> |
| 例 | <p>次の例は、値 26、708、672,843.3002 を返します。</p> <pre>SELECT REGR_SYY(Salary, (YEAR(NOW()) - YEAR(BirthDate))) FROM Employees;</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL2008 主要な SQL に含まれない、SQL/foundation の機能 (T621) です。 • Sybase SQL Anywhere 互換。 |

REMAINDER 関数 [数値]

| | |
|----|-------------------------------------------------------|
| 機能 | ある整数を別の整数で割った余りを返します。 |
| 構文 | REMAINDER (<i>dividend</i> , <i>divisor</i>) |

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | dividend 割られる数。つまり分数の分子。 divisor 割る数。つまり分数の分母。 |
| 例 | 次の文は、値 2 を返します。 <pre>SELECT REMAINDER(5, 3) FROM iq_dummy</pre> |
| 使用方法 | REMAINDER は MOD 関数と同じです。 |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 ベンダの拡張機能。• Sybase Adaptive Server Enterprise ではサポートされていません。 % (モジュロ) 演算子と除法演算子を使用して、余りを算出できます。 |
| 参照 | 「MOD 関数 [数値]」 (206 ページ) |

REPEAT 関数 [文字列]

| | |
|-------|----------------------------------------------------------------------------------------------------------------------------|
| 機能 | 文字列を指定された回数だけ連結します。 |
| 構文 | REPEAT (<i>string-expression</i> , <i>integer-expression</i>) |
| パラメータ | string-expression 繰り返す文字列。 integer-expression 文字列を繰り返す回数。 <i>integer-expression</i> が正数でない場合は、空の文字列を返します。 |

注意 REPEAT 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で REPEAT を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して REPEAT を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」を参照してください。

| | |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 例 | 次の文は、値 “repeatrepeatrepeat” を返します。 <pre>SELECT REPEAT('repeat', 3) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none">• SQL92 ベンダの拡張機能。• Sybase Adaptive Server Enterprise ではサポートされていませんが、REPLICATE で同じ機能を利用できます。 |
| 参照 | 「REPLICATE 関数 [文字列]」 (244 ページ) |

REPLACE 関数 [文字列]

- 機能** 検出されたすべての部分文字列を、別の部分文字列に置換します。
- 構文** `REPLACE (original-string, search-string, replace-string)`
- パラメータ** いずれかの引数が NULL であれば、関数から NULL が返されます。
- original-string** 検索される文字列。この文字列の長さに制限はありません。
- search-string** 検索して *replace-string* に置き換えられる文字列。255 バイトまでの文字列を指定してください。*search-string* が空の文字列であれば、元の文字列がそのまま返されます。
- replace-string** 置換文字列。*search-string* を置き換えます。この文字列の長さに制限はありません。*replace-string* が空の文字列であれば、検索されたすべての *search-string* が削除されます。
- replace-string* が *search-string* よりも長く、置換後のカラムの長さをコントロールする必要がある場合は、CAST 関数を使用してください。例を示します。

```
CREATE TABLE aa(a CHAR(5));
INSERT INTO aa VALUES('CCCCC');
COMMIT;
SELECT a, CAST(REPLACE(a,'C','ZZ') AS CHAR(5)) FROM aa;
```

例 次の文は値 “xx.def.xx.ghi” を返します。

```
SELECT REPLACE( 'abc.def.abc.ghi', 'abc', 'xx' ) FROM
iq_dummy
```

次の文は、ALTER PROCEDURE 文を含む結果セットを生成します。この結果セットを実行すると、名前が変更されたテーブルを参照するストアド・プロシージャが修復されます(使用するためには、テーブル名を一意にする必要があります)。

```
SELECT REPLACE(
    replace(proc_defn,'OldTableName','NewTableName'),
    'create procedure',
    'alter procedure')
FROM SYS.SYSPROCEDURE
WHERE proc_defn LIKE '%OldTableName%'
```

次の例では、LIST 関数の区切り文字に、コンマでないものを使用します。

```
SELECT REPLACE( list( table_id ), ',', '--')
FROM SYS.ISYSTAB
WHERE table_id <= 5
```

| | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 使用法 | <p>REPLACE 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で REPLACE を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して REPLACE を正しいデータ型とサイズに設定する必要があります。</p> <p>この問題には、次の 2 つの対処方法があります。</p> <ul style="list-style-type: none"> ローカル・テンポラリ・テーブルを宣言し、INSERT を実行します。 <pre>DECLARE local temporary table #mytable (name_column char(10)) on commit preserve rows; INSERT INTO #mytable SELECT REPLACE(name, '0', '1') FROM dummy_table01;</pre> CAST を使用します。 <pre>SELECT CAST(replace(name, '0', '1') AS Char(10)) into #mytable from dummy_table01;</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> SQL92 ベンダの拡張機能。 Sybase Adaptive Server Enterprise 互換。 |
| 参照 | 「SUBSTRING 関数 [文字列]」 (265 ページ) |

REPLICATE 関数 [文字列]

| | |
|-------|----------------------------------------------------------------------------------------------------|
| 機能 | 文字列を指定された回数だけ連結します。 |
| 構文 | REPLICATE (<i>string-expression</i> , <i>integer-expression</i>) |
| パラメータ | <p>string-expression 繰り返す文字列。</p> <p>integer-expression 文字列を繰り返す回数。</p> |
| 例 | <p>次の文は、値 “repeatrepeatrepeat” を返します。</p> <pre>SELECT REPLICATE('repeat', 3) FROM iq_dummy</pre> |
| 使用法 | REPLICATE は REPEAT 関数と同じです。 |

注意 REPLICATE 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で REPLICATE を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して REPLICATE を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]」 \(243 ページ\)](#) を参照してください。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。
- 参照 [「REPEAT 関数 \[文字列\]」 \(242 ページ\)](#)

REVERSE 関数 [文字列]

機能 BINARY データ型または STRING データ型の入力として 1 つの引数を取り、文字順を逆にした指定した文字列を返します。

構文 **REVERSE** (*expression* | *uchar_expr*)

パラメータ **expression** 文字またはバイナリ型のカラム名、変数、または CHAR、VARCHAR、NCHAR、NVARCHAR、BINARY、VARBINARY 型の定数式。

例 1

```
select reverse("abcd")
-----
dcba
```

例 2

```
select reverse(0x12345000)
-----
0x00503412
```

- 使用法**
- REVERSE は文字列関数で、式の文字順を逆にした文字列を返します。
 - 式が NULL の場合、reverse は NULL を返します。
 - サロゲート・ペアは分割不可能なデータとして処理されるため、文字順が逆にされることはありません。

パーミッション すべてのユーザが REVERSE 関数を実行できます。

標準と互換性 ANSI SQL 準拠レベル：Transact SQL の拡張機能。

参照 **関数** [「LOWER 関数 \[文字列\]」 \(201 ページ\)](#)、[「UPPER 関数 \[文字列\]」 \(329 ページ\)](#)

文字列関数の一般的な説明については、[「文字列関数」 \(117 ページ\)](#) を参照してください。

RIGHT 関数 [文字列]

機能 文字列の右端の文字を返します。

構文 **RIGHT** (*string-expression*, *numeric-expression*)

| | |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ | string-expression 左トランケートされる文字列。 numeric-expression 返す (文字列の末尾の) 文字数。 |
| 例 | 次の文は、値 “olate” を返します。 <pre>SELECT RIGHT('chocolate', 5) FROM iq_dummy</pre> |
| 使用法 | 文字列にマルチバイト文字が含まれ、照合が適切に使用されている場合は、返されるバイト数が、指定された文字数よりも多くなる場合があります。 |
| | <hr/> 注意 RIGHT 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で RIGHT を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して RIGHT を正しいデータ型とサイズに設定する必要があります。 <hr/> 詳細については、「 REPLACE 関数 [文字列] 」(243 ページ) を参照してください。 <hr/> |
| 標準と互換性 | <ul style="list-style-type: none"> • SQL92 ベンダの拡張機能。 • Sybase Adaptive Server Enterprise 互換。 |
| 参照 | <p>「LEFT 関数 [文字列]」(196 ページ)</p> <p>『システム管理ガイド 第1巻』の「第11章 各国語と文字セット」</p> |

ROUND 関数 [数値]

| | |
|-------|---------------------------------------------------------------------------------------------------------------------------------|
| 機能 | <i>numeric-expression</i> を、 <i>integer-expression</i> で指定した小数点以下の桁数で丸めます。 |
| 構文 | ROUND (<i>numeric-expression</i> , <i>integer-expression</i>) |
| パラメータ | numeric-expression 関数に渡され、丸められる数値。 integer-expression 正の整数は、丸めを行う小数点以下の有効桁数を指定します。負の式は、丸めを行う小数点の左側の有効桁数を指定します。 |
| 例 | 次の文は、値 123.200 を返します。 <pre>SELECT ROUND(123.234, 1) FROM iq_dummy</pre> このほかの、ROUND 関数の実行結果を次の表に示します。 |

| 値 | ROUND (値) |
|----------|-----------------|
| 123.4567 | round (a.n,4) |
| 123.4570 | round (a.n,3) |
| 123.4600 | round (a.n,2) |
| 123.5000 | round (a.n,1) |
| 123.0000 | round (a.n, 0) |
| 120.0000 | round (a.n, -1) |
| 100.0000 | round (a.n, -2) |
| 0.0000 | round (a.n, -3) |

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

参照

「[TRUNCNUM 関数 \[数値\]](#)」(269 ページ)

ROWID 関数 [その他]

機能

テーブルの各ローに対して、内部ロー ID を返します。

構文

ROWID (*table-name*) ...**FROM** *table-name*

パラメータ

table-name テーブル名。テーブル名は二重引用符を付け、カッコで括って指定します。二重引用符は省略できますが、引用符を使うことはできません。

例

次の文は、1 ~ 10 のロー ID の値を返します。

```
SELECT ROWID( "PRODUCTS" ) FROM PRODUCTS
```

rowid(Products)

```
1
2
3
.
.
.
10
```

次の文は、product ID の値が 400 よりも小さいすべてのローの product ID とロー ID の値を返します。

```
SELECT PRODUCTS.ID, ROWID ( PRODUCTS )
```

```
FROM PRODUCTS
WHERE PRODUCTS.ID < 400
```

| ID | rowid(Products) |
|-----|-----------------|
| 300 | 1 |
| 301 | 2 |
| 302 | 3 |

次の文は、ロー ID の値が 50 よりも大きいすべてのローを削除します。

```
DELETE FROM PRODUCTS
WHERE ROWID ( PRODUCTS ) > 50
```

使用法

ROWID 関数は、テーブルの特定のローを操作する場合に他の句と組み合わせることができます。

FROM *table-name* 句を指定する必要があります。

ROWID 関数には、テーブルのジョイン・インデックスを使用できないという制限事項があります。このため、ジョイン・インデックスを使用できる場合と比較すると、通常よりパフォーマンスが劣ります。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

RTRIM 関数 [文字列]

機能

後続ブランクを取り除いた文字列を返します。

構文

```
RTRIM ( string-expression )
```

パラメータ

string-expression トリミングされる文字列。

注意 RTRIM 関数の結果データ型は LONG VARCHAR です。SELECT INTO 文で RTRIM を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して RTRIM を正しいデータ型とサイズに設定する必要があります。

詳細については、「[REPLACE 関数 \[文字列\]](#)」(243 ページ) を参照してください。

例

次の文は、後続ブランクがすべて削除された文字列 “Test Message” を返します。

```
SELECT RTRIM( 'Test Message      ' ) FROM iq_dummy
```

- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise 互換。
- 参照 [「LTRIM 関数 \[文字列\]」 \(202 ページ\)](#)

SECOND 関数 [日付および時刻]

- 機能 指定された日時の値の秒に対応する 0 から 59 までの数字を返します。
- 構文 **SECOND** (*datetime-expression*)
- パラメータ **datetime-expression** 日時の値。
- 例 次の文は、値 5 を返します。
- ```
SELECT SECOND('1998-07-13 08:21:05') FROM iq_dummy
```
- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise 互換。

## SECONDS 関数 [日付および時刻]

- 機能 任意の開始日時から経過した秒数を返すか、2 つの時刻の間の秒数を返すか、または整数で指定された秒を時刻に追加します。
- 構文 **SECONDS** ( *datetime-expression*  
| *datetime-expression, datetime-expression*  
| *datetime-expression, integer-expression* )
- パラメータ **datetime-expression** 日時。
- integer-expression** *datetime-expression* に追加する秒数。*integer-expression* が負の場合、指定された秒数が日時の値から減算されます。整数式を指定する場合、*datetime-expression* は `datetime` データ型として明示的にキャストする必要があります。
- データ型のキャストについては、[「CAST 関数 \[データ型変換\]」 \(134 ページ\)](#) を参照してください。
- 例 次の文は、値 3600 を返します。
- ```
SELECT ( SECONDS( '1998-07-13 06:07:12' ) -
        SECONDS( '1998-07-13 05:07:12' ) ) FROM iq_dummy
```
- 次の文を実行すると、2 つの時刻の間の差である値 14400 が返されます。
- ```
SELECT SECONDS('1999-07-13 06:07:12',
 '1999-07-13 10:07:12') FROM iq_dummy
```

次の文を実行すると、日時の値 1999-05-12 21:05:12.000 が返ります。

```
SELECT SECONDS(CAST('1999-05-12 21:05:07'
AS TIMESTAMP), 5) FROM iq_dummy
```

使用法

2 つ目の構文は、最初の引数の日時から 2 番目の引数の日時までが何秒以上かを返します。負の値が返ることもあります。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

## SIGN 関数 [数値]

機能

数値の符号を返します。

構文

**SIGN** ( *numeric-expression* )

パラメータ

**numeric-expression** 符号を返す数値。

例

次の文は、値 -1 を返します。

```
SELECT SIGN(-550) FROM iq_dummy
```

使用法

負の数に対しては、**SIGN** は -1 を返します。

ゼロに対しては、**SIGN** は 0 を返します。

正の数に対しては、**SIGN** は 1 を返します。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

## SIMILAR 関数 [文字列]

機能

2 つの文字列の類似性を表す 0 から 100 の間の整数を返します。

構文

**SIMILAR** ( *string-expression1*, *string-expression2* )

パラメータ

**string-expression1** 比較を行う最初の文字列。

**string-expression2** 比較を行う 2 つ目の文字列。

例

次の文は、値 75 を返します。

```
SELECT SIMILAR('toast', 'coast') FROM iq_dummy
```

これは、2 つの値の類似性が 75% であることを示します。

**使用法** この関数は、2 つの文字列の類似性を表す 0 から 100 の間の整数を返します。結果は、2 つの文字列の間で文字が一致する割合と解釈することができます。値 100 は、2 つの文字列が同じであることを意味します。

この関数は、名前 (顧客名など) のリストを訂正するときに使用します。顧客の中には、少しだけ違う名前で何回もリストに追加される人がいるかもしれません。テーブルを連結し、類似度が 90% 以上 100% 未満であるものすべてについて、レポートを作成してみてください。

**標準と互換性**

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

## SIN 関数 [数値]

**機能** 数値の正弦をラジアンで返します。

**構文** `SIN ( numeric-expression )`

**パラメータ** `numeric-expression` 角度 (ラジアン)。

**例** 次の文は、値 0.496880 を返します。

```
SELECT SIN(0.52) FROM iq_dummy
```

**標準と互換性**

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** 互換。

**参照**

[「ASIN 関数 \[数値\]」 \(130 ページ\)](#)  
[「COS 関数 \[数値\]」 \(143 ページ\)](#)  
[「COT 関数 \[数値\]」 \(144 ページ\)](#)  
[「TAN 関数 \[数値\]」 \(268 ページ\)](#)

## SORTKEY 関数 [文字列]

**機能** 代替の照合規則に基づいて、文字列をソートするために使用する値を生成します。

**構文** `SORTKEY ( string-expression  
[, { collation-id  
| collation-name [(collation-tailoring-string)] } ]  
)`

パラメータ

**string-expression** 文字列式は、データベースの文字セットでエンコードされた文字を含む必要があります、**STRING** データ型である必要があります。

*string-expression* が NULL であれば、**SORTKEY** は NULL 値を返します。空の文字列は、データベース・カラムの NULL 文字列とは異なるソート順の値を持ちます。

**SORTKEY** 関数が処理できる入力文字列の長さに制限はありません。**SORTKEY** の結果は、常に 1024 バイトに制限された **VARBINARY** データ型です。実際の結果が 1024 バイトを超えた場合、結果には最初の 1024 バイトだけが含まれます。

**collation-name** 使用するソート順の名前を指定する文字列または文字変数。エイリアス `char_collation`、または `db_collation` を指定して、データベースによって使用される **CHAR** 照合で使用される *i* ソート・キーを生成することもできます。

同様に、エイリアス `NCHAR_COLLATION` を指定して、データベースによって使用される **NCHAR** 照合で使用されるソート・キーを生成できます。しかし、**Sybase IQ** は、**IQ** 固有のオブジェクトに対して `NCHAR_COLLATION` をサポートしません。`NCHAR_COLLATION` は、**IQ** サーバ上の **SQL Anywhere** オブジェクトに対してサポートされます。

**collation-id** 使用するソート順の ID 番号を指定する変数、整数定数、または文字列。このパラメータは、対応する照合 ID によって参照される **Adaptive Server Enterprise** 照合にのみ適用されます。

照合名または照合 ID を指定しない場合、デフォルトはデフォルト **Unicode** マルチ言語です。

有効な照合は次のとおりです。

- **SQL Anywhere** でサポートされる照合をラベル順に表示するには、`dbinit -l` を実行します。
- 次の表に **Adaptive Server Enterprise** 照合を示します。

| 説明                           | 照合名      | 照合 ID |
|------------------------------|----------|-------|
| デフォルト <b>Unicode</b> マルチ言語   | default  | 0     |
| CP 850 代替：アクセント記号なし          | altnoacc | 39    |
| CP 850 代替：小文字優先              | altdict  | 45    |
| CP 850 西ヨーロッパ：大文字小文字の優先指定なし  | altnocsp | 46    |
| CP 850 スカンジナビア語辞書            | scandict | 47    |
| CP 850 スカンジナビア：大文字小文字の優先指定なし | scannocp | 48    |

| 説明                                   | 照合名      | 照合 ID |
|--------------------------------------|----------|-------|
| GB ピンイン                              | gbpinyin | n/a   |
| バイナリ・ソート                             | binary   | 50    |
| Latin-1 英語、フランス語、ドイツ語辞書              | dict     | 51    |
| Latin-1 英語、フランス語、ドイツ語、大文字/小文字の区別なし   | nocase   | 52    |
| Latin-1 英語、フランス語、ドイツ語、大文字/小文字の優先設定なし | nocasep  | 53    |
| Latin-1 英語、フランス語、ドイツ語、アクセント記号なし      | noaccent | 54    |
| Latin-1 スペイン語辞書                      | espdict  | 55    |
| Latin-1 スペイン語、大文字/小文字の区別なし           | espnocs  | 56    |
| Latin-1 スペイン語、アクセント記号なし              | espnoc   | 57    |
| ISO 8859-5 ロシア語辞書                    | rusdict  | 58    |
| ISO 8859-5 ロシア語、大文字小文字の区別なし          | rusnocs  | 59    |
| ISO 8859-5 キリル語辞書                    | cyrdict  | 63    |
| ISO 8859-5 キリル語、大文字小文字の区別なし          | cymnocs  | 64    |
| ISO 8859-7 ギリシア語辞書                   | elldict  | 65    |
| ISO 8859-2 ハンガリー語辞書                  | hundict  | 69    |
| ISO 8859-2 ハンガリー語、アクセント記号なし          | hunnoac  | 70    |
| ISO 8859-2 ハンガリー語、大文字小文字の区別なし        | hunnocs  | 71    |
| ISO 8859-5 トルコ語辞書                    | turdict  | 72    |
| ISO 8859-5 トルコ語、アクセント記号なし            | turnoac  | 73    |
| ISO 8859-5 トルコ語、大文字小文字の区別なし          | turnocs  | 74    |
| CP 874 (TIS 620) タイ語辞書               | thaidict | 1     |
| ISO 14651 標準の順序                      | 14651    | 22    |
| Shift-JIS バイナリ順                      | sjisbin  | 179   |
| Unicode UTF-8 バイナリ・ソート               | utf8bin  | 24    |
| EUC JIS バイナリ順                        | ucjisbn  | 192   |
| GB2312 バイナリ順                         | gb2312bn | 137   |
| CP932 MS バイナリ順                       | cp932bin | 129   |
| Big5 バイナリ順                           | big5bin  | 194   |
| EUC KSC バイナリ順                        | euckscbn | 161   |

**collation-tailoring-string** (オプション) ソートおよび文字の比較に追加の制御を行う照合調整オプション (*collation-tailoring-string*) を指定します。これらのオプションは、照合名の後にカッコで囲まれた「キーワード=値」ペアの形を取ります。例を示します。

```
'UCA(locale=es;case=LowerFirst;accent=respect)'
```

これらのオプションを指定する構文は、CREATE DATABASE 文の COLLATION 句と同じです。『リファレンス：文とオプション』の「[第1章 SQL 文](#)」の「[CREATE DATABASE 文](#)」の「*collation-tailoring-string*」を参照してください。

---

**注意** Unicode Collation Algorithm (UCA) 照合を指定した場合、すべての照合調整オプションは SQL Anywhere データベースでサポートされません。その他のすべての照合では、大文字と小文字を区別する調整だけがサポートされます。

---

**例** 次の文は、Employees テーブルをクエリし、すべての従業員の GivenName および Surname を Surname カラムのソート・キー値でソートして返します。ソートには、dict 照合 (Latin-1、英語、フランス語、ドイツ語辞書) が使用されます。

```
SELECT Surname, GivenName FROM Employees ORDER BY
SORTKEY(Surname, 'dict');
```

**使用法** SORTKEY 関数は、事前定義されたソート順の動作に基づいて結果を並べ替える際に使用できる値を生成します。データベース照合で使用できない文字ソート順の動作を使用できます。戻り値は、入力文字列に対して SORTKEY 関数が保持するソート順の情報がコーディングされたバイナリ値です。

たとえば、元の文字列を指定して、SORTKEY 関数から返される値をカラムに格納できます。次の SELECT 文は、テーブル T1 からタイ語辞書に従って c1 のソート順でデータを取得します。

```
SELECT rid, c1 from T1 ORDER BY SORTKEY(c1)
```

あるいは、元の文字列を指定して、SORTKEY から返される値をカラムに格納することもできます。必要な順序で文字データを取得するために、SELECT 文は、SORTKEY 関数を実行した結果を含むカラムに 1 つの ORDER BY 句だけを含む必要があります。

```
UPDATE T1 SET shadowc1=SORTKEY(c1) FROM T1;
SELECT rid, c1 FROM T1 ORDER BY shadowc1
```



**SORTKEY** 関数により、指定されたソート順の基準のセットに対して返される値が、**VARBINARY** データ型について実行されるバイナリ比較に使用できることが保証されます。

クエリにソート・キーを生成すると負荷がかかる場合があります。頻繁に要求されるソート・キーの代替として、ソート・キー値を格納する計算カラムを作成して、クエリの **ORDER BY** 句でそのカラムを参照することを検討してください。

照合調整の面から、ソート・キーを作成するとき、一般的に完全な大文字と小文字の区別は意図されたものです。したがって、UCA 以外の照合を指定する場合は、適用されるデフォルトの調整は **case=Respect** と同等です。たとえば、次の 2 つの文は同じです。

```
SELECT SORTKEY('abc', '1252LATIN1');
SELECT SORTKEY('abc', '1252LATIN1(case=Respect)');
```

UCA 以外の照合を指定するとき、デフォルトでは、照合調整はアクセント記号付きで大文字と小文字が区別されます。しかし、UCA 以外の照合では、照合調整を使用して上書きできるのは大文字と小文字の区別だけです。次に例を示します。

```
SELECT SORTKEY('abc', '1252LATIN1(case=LowerFirst)');
```

調整オプションを指定せずにデータベースを作成した場合、**SORTKEY** 関数にデータベース照合名を指定しても、次の 2 つの句では異なるソート順が生成されることがあります。

```
ORDER BY string-expression
ORDER BY SORTKEY(string-expression, database-
collation-name)
```

データベース作成と **SORTKEY** 関数に使用されるデフォルトの調整設定が異なるので、異なるソート順が生成されることがあります。データベース照合と同じ動作を **SORTKEY** から得るには、データベース照合の設定に一致する調整構文を *collation-tailoring-string* に指定するか、照合名に **db\_collation** を指定します。次に例を示します。

```
SORTKEY(expression, 'db_collation')
```

照合調整および **SORTKEY** 関数と SQL Anywhere データベースの使用の詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の「SQL 関数」を参照してください。

---

**注意** 15.0 より前のバージョンの Sybase IQ を使用して作成されたソート・キー値には、15.0 以降のバージョンを使用して作成された同じ値は含まれません。これは、15.0 以前のデータベースにソート・キーが格納されている場合、特にソート・キー値の比較がアプリケーションで必要な場合に問題になることがあります。Sybase IQ の 15.0 より前のバージョンを使用して生成されたデータベースのソート・キー値を再生成してください。

---

## 参照

「[第 2 章 データベース・オプション](#)」の「[SORT\\_COLLATION オプション](#)」

「[文字列関数](#)」(117 ページ)

『システム管理ガイド 第 1 巻』の「[第 11 章 各国語と文字セット](#)」

## SOUNDEX 関数 [文字列]

**機能** 文字列の音を表す数値を返します。

**構文** **SOUNDEX** ( *string-expression* )

**パラメータ** **string-expression** 文字列。

**例** 次の文は、それぞれの名前の音を表す 2 つの数値を返します。引数に対する SOUNDEX の値は、どちらも 3827 です。

```
SELECT SOUNDEX('Smith'), SOUNDEX('Smythe') FROM
iq_dummy
```

SOUNDEX ( 'Smith' ) と SOUNDEX ( 'Smythe' ) は同じ値を返します。

## 使用法

文字列の SOUNDEX 関数値は、最初の英字と、それに続く H、Y、W 以外の 3 つの子音を基に生成されます。重なる英字は 1 字としてカウントします。次に例を示します。

```
SOUNDEX('apples') FROM iq_dummy
```

これは、A、P、L、S の英字に基づいて処理されます。

SOUNDEX 関数では、マルチバイト文字は無視されます。

完全ではありませんが、同じように発音し、同じ英字で始まる言葉に対して、SOUNDEX は通常同じ数値を返します。

## 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** 互換。ただし、Adaptive Server Enterprise は結果を CHAR(4) で返し、Sybase IQ は整数で返します。

## SPACE 関数 [文字列]

- 機能** 指定された数のスペースを返します。
- 構文** `SPACE ( integer-expression )`
- パラメータ** **integer-expression** 返されるスペースの数。
- 例** 次の文は、10 個のスペースを含む文字列を返します。  

```
SELECT SPACE(10) FROM iq_dummy
```
- 標準と互換性**
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise 互換。

## SQRT 関数 [数値]

- 機能** 数値の平方根を返します。
- 構文** `SQRT ( numeric-expression )`
- パラメータ** **numeric-expression** 平方根が計算される数値。
- 例** 次の文は、値 3 を返します。  

```
SELECT SQRT(9) FROM iq_dummy
```
- 標準と互換性**
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise 互換。

## SQUARE 関数 [数値]

- 機能** 指定した式の平方を、float で表したものを返します。
- 構文** `SQUARE ( numeric-expression )`
- パラメータ** **expression** カラム、変数、またはデータ型が真数値、概数値、通貨、またはこれらの型の 1 つに暗黙的に変換できる式です。他のデータ型を指定すると、SQUARE 関数ではエラーが返ります。戻り値は、DOUBLE データ型です。
- 使用法** SQUARE 関数は 1 つの引数を取ります。たとえば `SQUARE (12.01)` は、144.240100 を返します。
- 標準と互換性**
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise 互換。

## STDDEV 関数 [集合]

**機能** 数値セットの標準偏差を返します。

**構文** **STDDEV** ( [ ALL ] *expression* )

**パラメータ** **expression** 任意の数値式 (FLOAT、REAL、または DOUBLE 精度)。

**例** 次のようなデータがあるとします。

```
SELECT Salary FROM Employees WHERE DepartmentID = 300
```

| Salary     |
|------------|
| 51432.000  |
| 57090.000  |
| 42300.000  |
| 43700.00   |
| 36500.000  |
| 138948.000 |
| 31200.000  |
| 58930.00   |
| 75400.00   |

次の文は、値 32617.8446712838471 を返します。

```
SELECT STDDEV (Salary) FROM Employees
WHERE DepartmentID = 300
```

次のようなデータがあるとします。

```
SELECT UnitPrice FROM Products WHERE Name = 'Tee Shirt'
```

| 名前        | UnitPrice |
|-----------|-----------|
| Tee Shirt | 9.00      |
| Tee Shirt | 14.00     |
| Tee Shirt | 14.00     |

次の文は、値 2.88675134594813049 を返します。

```
SELECT STDDEV (UnitPrice) FROM Products
WHERE Name = 'Tee Shirt'
```

**使用法** STDDEV では、次の計算式が使用されます。

$$stddev = \sqrt{variance}$$

STDDEV は、結果を DOUBLE 型の精度浮動小数点数のデータ型で返します。空のセットに適用すると、結果は NULL になり、1 要素の入力セットに NULL が返されます。

STDDEV は、キーワード DISTINCT をサポートしません。DISTINCT を使用すると、構文エラーが返されます。

## 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

## 参照

[「STDDEV\\_SAMP 関数 \[集合\]」 \(260 ページ\)](#)

[「VARIANCE 関数 \[集合\]」 \(333 ページ\)](#)

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

## STDDEV\_POP 関数 [集合]

## 機能

1 つの数値式で構成される母標準偏差を DOUBLE 型として計算します。

## 構文

**STDDEV\_POP** ([ ALL ] *expression*)

## パラメータ

**expression** その母集団ベースの標準偏差がローのセットに対して計算される式 (通常はカラム名)。

## 例

次の文では、複数の期間における各注文の項目数の平均および分散を示します。

```
SELECT year(ship_date) AS Year, quarter(ship_date)
 AS Quarter, AVG(quantity) AS Average,
 STDDEV_POP (quantity) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter;
```

| 年    | 四半期 | 平均        | 偏差      |
|------|-----|-----------|---------|
| 2000 | 1   | 25.775148 | 14.2794 |
| 2000 | 2   | 27.050847 | 15.0270 |
| ...  | ... | ...       | ...     |

## 使用法

グループまたはパーティションの各ロー (DISTINCT が指定されている場合、重複が削除された後に残る各ロー) に対して評価される、指定された *value expression* の母標準偏差を計算します。これは、母分散の平方根として定義されます。

$$\sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

- 標準と互換性
- **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
  - **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照 [「分析関数」\(106 ページ\)](#)  
『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

## STDDEV\_SAMP 関数 [集合]

機能 1 つの数値式で構成される標本標準偏差を DOUBLE 型として計算します。

---

**注意** STDDEV\_SAMP は STDDEV のエイリアスです。

---

構文 **STDDEV\_SAMP** ([ ALL ] *expression* )

パラメータ **expression** その標本ベースの標準偏差がローのセットに対して計算される式 (通常はカラム名)。

例 次の文では、複数の期間における各注文の項目数の平均および分散を示します。

```
SELECT year(ship_date) AS Year, quarter(ship_date)
 AS Quarter, AVG(quantity) AS Average,
 STDDEV_SAMP(quantity) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter;
```

| 年    | 四半期 | 平均        | 偏差      |
|------|-----|-----------|---------|
| 2000 | 1   | 25.775148 | 14.3218 |
| 2000 | 2   | 27.050847 | 15.0696 |
| ...  | ... | ...       | ...     |

使用法 グループまたはパーティションの各ロー (DISTINCT が指定されている場合、重複が削除された後に残る各ロー) に対して評価される、指定された *value expression* の標本標準偏差を計算します。これは、標本分散の平方根として定義されます。

1 要素の入力セットの場合、NULL によって NULL が返されます。

標準偏差は次の式に従って計算されます。これは正規分布とみなされます。

$$\sqrt{\frac{\sum (x_i - \bar{x})^2}{(n-1)}}$$

#### 標準と互換性

- **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

#### 参照

「分析関数」(106 ページ)

「STDDEV 関数 [集合]」(258 ページ)

『システム管理ガイド 第2巻』の「第2章 OLAP の使用」

## STR 関数 [文字列]

#### 機能

数値に対応する文字列を返します。

#### 構文

**STR** ( *numeric-expression* [ , *length* [ , *decimal* ] ] )

#### パラメータ

**numeric-expression** 任意の概数 (FLOAT、REAL、または DOUBLE 精度) 式。

**length** 返される文字 (小数点、小数点の両側にあるすべての桁、必要な符号、ブランクを含む) の数。デフォルトは 10 で、最大長は 255 です。

**decimal** 返される文字の小数点以下の桁数。デフォルトは 0 です。

#### 例

次の文は、6 つのスペースの後に 1234 が続く、合計で 10 の文字から成る文字列を返します。

```
SELECT STR(1234.56) FROM iq_dummy
```

次の文は、1234.5を返します。

```
SELECT STR(1234.56, 6, 1) FROM iq_dummy
```

#### 使用法

数値の整数部分が指定された長さを超える場合は、NULL が返されません。たとえば、次の文は NULL を返します。

```
SELECT STR(1234.56, 3) FROM iq_dummy
```

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

## STR\_REPLACE 関数 [文字列]

|       |                                                                                                                                                                                                                                                                                                                                                                             |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | BINARY 型または STRING 型の入力として 3 つの引数を取り、最初の文字列式 ( <i>string_expr1</i> ) 内に出現する 2 番目の文字列式 ( <i>string_expr2</i> ) のすべてのインスタンスを、3 番目の式 ( <i>string_expr3</i> ) で置き換えます。                                                                                                                                                                                                          |
|       | STR_REPLACE は REPLACE 関数のエイリアスです。                                                                                                                                                                                                                                                                                                                                           |
| 構文    | <b>REPLACE</b> ( <i>string_expr1</i> , <i>string_expr2</i> , <i>string_expr3</i> )                                                                                                                                                                                                                                                                                          |
| パラメータ | <p><b>string_expr1</b> 検索される側のソース文字列または文字列式です。CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY データ型として表されます。</p> <p><b>string_expr2</b> 最初の式 (<i>string_expr1</i>) 内で検索するパターン文字列または文字列式です。CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY データ型として表されます。</p> <p><b>string_expr3</b> 置換文字列式です。CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY データ型として表されます。</p> |
| 例 1   | <p>文字列 <i>cdefghi</i> 内の文字列 <i>def</i> を <i>yyy</i> に置き換えます。</p> <pre>select replace("cdefghi", "def", "yyy") ----- cyyyghi (1 row(s) affected)</pre>                                                                                                                                                                                                                       |
| 例 2   | <p>すべてのスペースを “toyota” に置き換えます。</p> <pre>select str_replace ("chevy, ford, mercedes", " ", "toyota") ----- chevy, toyotaford, toyotamercedes (1 row(s) affected)</pre>                                                                                                                                                                                                       |
| 例 3   | <p>3 つ目のパラメータに NULL を使用できるようになりました。この場合、<i>string_expr2</i> が NULL に置き換えられます。この方法で STR_REPLACE を使用して、文字列を削除するオペレーションを行うことができます。“abcghijklm” が返されます。</p> <pre>select str_replace("abcdefghijklm", "def", NULL) ----- abcghijklm (1 row affected)</pre>                                                                                                                       |
| 使用法   | <ul style="list-style-type: none"> <li>入力として任意のデータ型を取り、STRING または BINARY を返します。</li> </ul>                                                                                                                                                                                                                                                                                  |



たとえば、引数として渡された空の文字列(“”)は、さらに評価される前に1つのスペース(“ ”)に置き換えられます。これは、BINARY型でもSTRING型でも同じです。

- すべての引数には、BINARY データ型と STRING データ型の組み合わせを指定できます。
- 結果の長さは、式がコンパイルされるときに、引数値が既知かどうかによって異なります。すべての引数が定数に割り当てられたカラムまたはホスト数である場合、結果の長さは、Sybase IQ によって次のように計算されます。

```
result_length = ((s/p)*(r-p)+s)
WHERE
 s = length of source string
 p = length of pattern string
 r = length of replacement string
IF (r-p) <= 0, result length = s
```

- 式のコンパイル時に引数値がわからないため Sybase IQ が結果の長さを計算できない場合は、結果の長さは 255 になります。
- RESULT\_LEN が 32767 を超えることはありません。

パーミッション

すべてのユーザが STR\_REPLACE 関数を実行できます。

標準と互換性

ANSI SQL — 準拠レベル：Transact SQL の拡張機能。

参照

**データ型** CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY。詳細については、「[第3章 SQL データ型](#)」を参照してください。

**機能** 「[LENGTH 関数 \[文字列\] \(198 ページ\)](#)」

文字列関数の一般的な説明については、「[文字列関数 \(117 ページ\)](#)」を参照してください。

## STRING 関数 [文字列]

機能

1つ以上の文字列を連結して、1つの大きな文字列にします。

構文

**STRING** ( *string-expression* [, … ] )

パラメータ

**string-expression** 文字列。

引数が1つしか指定されなければ、単一の式に変換されます。引数が複数指定されると、それらが1つの文字列に連結されます。

NULL は空の文字列(“)として扱われます。

- 例 次の文を実行すると、値 `testing123` が返ります。
- ```
SELECT STRING( 'testing', NULL, 123 )
FROM iq_dummy
```
- 使用法 数値または日付のパラメータは、文字列に変換されてから連結されます。`STRING` 関数を使用して、1つの式を文字列に変換することもできます。それには、変換する式を唯一のパラメータとして指定します。
- すべてのパラメータが `NULL` であれば、`STRING` は `NULL` を返します。
- 標準と互換性
- **SQL92** ベンダの拡張機能。
 - **Sybase** Adaptive Server Enterprise ではサポートされていません。

STRTOUUID 関数 [文字列]

- 機能 文字列の値を、一意な識別子 (UUID または GUID) の値に変換します。
- 構文 **STRTOUUID** (*string-expression*)
- パラメータ **string-expression** `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` の形式の文字列。
- 例
- ```
CREATE TABLE T (
 pk uniqueidentifier primary key,
 c1 int);
INSERT INTO T (pk, c1)
VALUES (STRTOUUID
 ('12345678-1234-5678-9012-123456789012'), 1);
```
- 使用法 `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` の形式の文字列 ( $x$  は 16 進の桁) を、一意な識別子の値に変換します。文字列が `UUID` として有効でなければ、`NULL` が返されます。
- `STRTOUUID` を使用して、`UUID` 値を Sybase IQ データベースに挿入できます。
- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **SQL99** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。
- 参照
- [「NEWID 関数 \[その他\]」 \(208 ページ\)](#)
  - [「UUIDTOSTR 関数 \[文字列\]」 \(330 ページ\)](#)
  - [「バイナリ・データ型」 \(77 ページ\)](#) の「`UNIQUEIDENTIFIER`」

## STUFF 関数 [文字列]

|        |                                                                                                                                                                                            |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | ある文字列からいくつかの文字を削除し、そこへ別の文字列を置きます。                                                                                                                                                          |
| 構文     | <b>STUFF</b> ( <i>string-expression1</i> , <i>start</i> , <i>length</i> , <i>string-expression2</i> )                                                                                      |
| パラメータ  | <p><b>string-expression1</b> STUFF 関数によって変更される文字列。</p> <p><b>start</b> 文字の削除を開始する位置です。文字列の先頭文字の位置が 1 になります。</p> <p><b>length</b> 削除される文字数です。</p> <p><b>string-expression2</b> 挿入する文字列。</p> |
| 例      | <p>次の文を実行すると、値 “chocolate pie” が返ります。</p> <pre>SELECT STUFF( 'chocolate cake', 11, 4, 'pie' ) FROM iq_dummy</pre>                                                                          |
| 使用法    | STUFF を使って文字列の一部を削除するには、置換後の文字列に NULL を指定します。STUFF を使って文字列を挿入するには、length をゼロに指定します。                                                                                                        |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> ベンダの拡張機能。</li> <li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li> </ul>                                                         |
| 参照     | <a href="#">「INSERTSTR 関数 [文字列]」 (190 ページ)</a>                                                                                                                                             |

## SUBSTRING 関数 [文字列]

|       |                                                                                                                                                                                                                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | 文字列の一部を返します。                                                                                                                                                                                                                                                                                                                         |
| 構文    | { <b>SUBSTRING</b>   <b>SUBSTR</b> } ( <i>string-expression</i> , <i>start</i> [ , <i>length</i> ] )                                                                                                                                                                                                                                 |
| パラメータ | <p><b>string-expression</b> 文字列。この文字列の一部が返されます。</p> <p><b>start</b> 返される部分文字列の開始位置。負の開始位置を指定する場合は、文字列の最初からの文字数ではなく、文字列の最後からの文字数を指定します。文字列の先頭文字の位置が 1 になります。</p> <p><b>length</b> 返される部分文字列の長さ。正の <i>length</i> を指定すると、開始位置から右側へ <i>length</i> 文字の位置で、部分文字列が終了します。一方、負の <i>length</i> を指定すると、開始位置から左側へ <i>length</i> 文字の位置で終了します。</p> |
| 例     | <p>次の文は、“back” を返します。</p> <pre>SELECT SUBSTRING ( 'back yard', 1 , 4 ) FROM iq_dummy</pre>                                                                                                                                                                                                                                           |

次の文は、`yard` を返します。

```
SELECT SUBSTR ('back yard', -1 , -4)
FROM iq_dummy
```

次の文は、`0x2233` を返します。

```
SELECT SUBSTR (0x112233445566, 2, 2)
FROM iq_dummy
```

#### 使用法

*length* を指定すると、指定した長さの部分文字列に制限されます。長さの指定を省略すると、*start* の位置を開始点として、文字列の残りがすべて返されます。

*start* と *length* の両方に負の値を指定できます。正と負を適切に組み合わせると、文字列の最初または最後から部分文字列を取得できます。

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** SUBSTR は、Adaptive Server Enterprise ではサポートされていません。代わりに SUBSTRING を使用してください。

## SUM 関数 [集合]

#### 機能

ローの各グループに対して、指定された式の合計を返します。

#### 構文

```
SUM (expression | DISTINCT column-name)
```

#### パラメータ

**expression** 合計する対象。通常はカラムを指定します。

**DISTINCT column-name** ローの各グループに対する *column-name* のうち、一意な値の合計を計算します。この方法で使用することはほとんどありませんが、万全を期すために含まれています。

#### 例

次の文は、値 3749146.740 を返します。

```
SELECT SUM(salary)
FROM Employees
```

#### 使用法

指定された式が NULL になるローは含まれません。

ローがまったくないグループに対しては、NULL 値を返します。

#### 標準と互換性

- **SQL92** SQL92 互換。
- **Sybase** Adaptive Server Enterprise 互換。

#### 参照

[「AVG 関数 \[集合\]」 \(131 ページ\)](#)

[「COUNT 関数 \[集合\]」 \(146 ページ\)](#)

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

## SUSER\_ID 関数 [システム]

|        |                                                                                                                                                         |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | ユーザ ID 番号を整数で返します。                                                                                                                                      |
| 構文     | <b>SUSER_ID</b> ( [ <i>user-name</i> ] )                                                                                                                |
| パラメータ  | <b>user-name</b> ユーザ名。                                                                                                                                  |
| 例      | 次の文は、ユーザ ID 番号 1 を返します。<br><pre>SELECT SUSER_ID ('DBA') FROM iq_dummy</pre> 次の文は、ユーザ ID 番号 0 を返します。<br><pre>SELECT SUSER_ID ('SYS') FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"><li>• <b>SQL92</b> ベンダの拡張機能。</li><li>• <b>Sybase</b> Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。</li></ul>     |
| 参照     | <a href="#">「SUSER_NAME 関数 [システム]」(267 ページ)</a><br><a href="#">「USER_ID 関数 [システム]」(329 ページ)</a>                                                         |

## SUSER\_NAME 関数 [システム]

|        |                                                                                                                                                                                                            |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | ユーザ名を返します。                                                                                                                                                                                                 |
| 構文     | <b>SUSER_NAME</b> ( [ <i>user-id</i> ] )                                                                                                                                                                   |
| パラメータ  | <b>user-id</b> ユーザ ID 番号。                                                                                                                                                                                  |
| 例      | 次の文は、値 DBA を返します。<br><pre>SELECT SUSER_NAME ( 1 ) FROM iq_dummy</pre> 次の文は、値 SYS を返します。<br><pre>SELECT SUSER_NAME ( 0 ) FROM iq_dummy</pre>                                                                |
| 標準と互換性 | <ul style="list-style-type: none"><li>• <b>SQL92</b> ベンダの拡張機能。</li><li>• <b>Sybase</b> Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。Adapter Server Enterprise では、SUSER_NAME はサーバのユーザ名を返します。</li></ul> |
| 参照     | <a href="#">「SUSER_ID 関数 [システム]」(267 ページ)</a><br><a href="#">「USER_NAME 関数 [システム]」(330 ページ)</a>                                                                                                            |

## TAN 関数 [数値]

|        |                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------|
| 機能     | 数値の正接を返します。                                                                                                                     |
| 構文     | <b>TAN</b> ( <i>numeric-expression</i> )                                                                                        |
| パラメータ  | <b>numeric-expression</b> 角度 (ラジアン)。                                                                                            |
| 例      | 次の文は、値 0.572561 を返します。<br><pre>SELECT TAN( 0.52 ) FROM iq_dummy</pre>                                                           |
| 標準と互換性 | <ul style="list-style-type: none"><li>• <b>SQL92</b> ベンダの拡張機能。</li><li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li></ul> |
| 参照     | <a href="#">「COS 関数 [数値]」 (143 ページ)</a><br><a href="#">「SIN 関数 [数値]」 (251 ページ)</a>                                              |

## TODAY 関数 [日付および時刻]

|        |                                                                                                                                            |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | 現在の日付を返します。これは <b>CURRENT DATE</b> に対応する古い構文です。                                                                                            |
| 構文     | <b>TODAY</b> ( * )                                                                                                                         |
| 例      | 次の文は、システム・クロックによる現在の日付を返します。<br><pre>SELECT TODAY( * ) FROM iq_dummy</pre>                                                                 |
| 標準と互換性 | <ul style="list-style-type: none"><li>• <b>SQL92</b> ベンダの拡張機能。</li><li>• <b>Sybase</b> Adaptive Server Enterprise ではサポートされていません。</li></ul> |

## TRIM 関数 [文字列]

|    |                                          |
|----|------------------------------------------|
| 機能 | 文字列から先行空白と後続空白を削除します。                    |
| 構文 | <b>TRIM</b> ( <i>string-expression</i> ) |

|        |                                                                                                                                                                                                                                                                                        |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パラメータ  | <p><b>string-expression</b> トリミングされる文字列。</p> <hr/> <p><b>注意</b> TRIM 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で TRIM を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して TRIM を正しいデータ型とサイズに設定する必要があります。</p> <p>詳細については、「<a href="#">REPLACE 関数 [文字列]</a> (243 ページ) を参照してください。</p> <hr/> |
| 例      | <p>次の文は、先行空白や後続空白がすべて削除された値 “chocolate” を返します。</p> <pre>SELECT TRIM( ' chocolate ' ) FROM iq_dummy</pre>                                                                                                                                                                               |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> ベンダの拡張機能。</li> <li>• <b>Sybase Adaptive Server Enterprise</b> ではサポートされていません。代わりに LTRIM と RTRIM を使用してください。</li> </ul>                                                                                                             |
| 参照     | <p><a href="#">LTRIM 関数 [文字列]</a> (202 ページ)</p> <p><a href="#">RTRIM 関数 [文字列]</a> (248 ページ)</p>                                                                                                                                                                                        |

## TRUNCNUM 関数 [数値]

|       |                                                                                                                                                                  |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | 数値を小数点以下の指定された桁数で切り捨てます。                                                                                                                                         |
| 構文    | <b>TRUNCNUM</b> ( <i>numeric-expression</i> , <i>integer-expression</i> )                                                                                        |
| パラメータ | <p><b>numeric-expression</b> 切り捨てられる数値。</p> <p><b>integer-expression</b> 正の整数は、丸めを行う小数点以下の有効桁数を指定します。負の式は、丸めを行う小数点の左側の有効桁数を指定します。</p>                            |
| 例     | <p>次の文は、値 600 を返します。</p> <pre>SELECT TRUNCNUM( 655, -2 ) FROM iq_dummy</pre> <p>次の文は、値 655.340 を返します。</p> <pre>SELECT TRUNCNUM( 655.348, 2 ) FROM iq_dummy</pre> |
| 使用法   | <p>この関数は TRUNCATE と同じですが、キーワードの矛盾が起こりません。</p> <p>ROUND、FLOOR、CEILING を組み合わせて使用することで類似した機能を提供できます。</p>                                                            |

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。
- 参照 [「ROUND 関数 \[数値\]」 \(246 ページ\)](#)

## TS\_ARMA\_AR 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

- 機能** 自己回帰移動平均 (ARMA) モデルのパラメータの最小二乗近似を計算し、要求された自己回帰近似を返します。
- 構文**
- ```
TS_ARMA_AR (timeseries_expression, ar_count, ar_elem, method)
OVER (window-spec)
```
- パラメータ**
- timeseries_expression** 数値式。一般的に、時系列の要素を含むカラム名。
- ar_count** 計算する自己回帰値の数を含む整数。
- ar_elem** 返される計算された AR 配列の要素を指定する整数。*ar_elem* は 0 よりも大きく、*ar_count* 以下である必要があります。
- method** (オプション) 推定の計算に使用するプロシージャのタイプを示す整数。0 (デフォルト値) = 最小 2 乗法、1 = モーメント法。
- window-spec** TS_ARMA_AR は、OVER () 句が必要な OLAP 関数です。
- 使用法** TS_ARMA_AR 時系列関数は、自己回帰近似を含む倍精度浮動小数点値を返します。TS_ARMA_AR は、IMSL ライブラリの関数 `imsls_d_arma` を呼び出します。
- IMSL マッピング** TS_ARMA_AR の引数は、次のように IMSL ライブラリ関数 `imsls_d_arma` にマップします。
- ```
params = imsls_d_arma(n_objs, z, p, q, methodID, 0);
```
- n\_objs** 現在のウィンドウ・フレームのローの数を含みます。
- z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。
- p** ユーザ定義の集合関数引数 *ar\_count* にマップします。
- q** = 1



**methodID** TS\_ARMA\_AR の *method* 引数にマップします。  
 IMSLS\_METHOD\_OF\_MOMENTS または IMSLS\_LEAST\_SQUARES に  
 設定できます。

`imsls_d_arma` が時系列計算を実行する方法の詳細については、『IMSL  
 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』  
 を参照してください。

例

この例は、入力データ・テーブル、TS\_ARMA\_AR 関数を含む SQL 文、  
 および関数から返されるデータ値を示します。この例は、次のテーブ  
 ル (DATASET) を入力データとして使用します。DATASET テーブルには、  
 時系列データの 50 のローが含まれています。

**表 4-27 : 入力データ・テーブル DATASET**

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |

| ロー番号 | データ       |
|------|-----------|
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、最小 2 乗法を使用して、**data** カラムの 1 つの値を含む自己回帰近似の最初の要素を返します。

```
SELECT TS_ARMA_AR(data,1,1,0) OVER (ORDER BY rownum
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-28 : TS\_ARMA\_AR から返される値**

| res      |
|----------|
| 0.898793 |
| 0.898793 |
| 0.898793 |

**res**

0.898793

0.898793

0.898793

0.898793

0.898793

0.898793

...

0.898793

## 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

## 参照

『システム管理ガイド 第 2 巻』の「第 2 章 OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

**TS\_ARMA\_CONST 関数 [時系列]**

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

## 機能

自己回帰移動平均 (ARMA) モデルのパラメータの最小二乗近似を計算し、推定定数を返します。

## 構文

**TS\_ARMA\_CONST** (*timeseries\_expression, method*)

**OVER** (*window-spec*)

## パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**method** 推定の計算に使用するプロシージャのタイプを示す整数。0 (デフォルト値) = 最小 2 乗法、1 = モーメント法

**window-spec** TS\_ARMA\_CONST は、OVER () 句が必要な OLAP 関数です。

**使用法** この時系列関数は、関数によって生成される定数推定を含む倍精度浮動小数点値を返します。TS\_ARMA\_CONST は、IMSL ライブラリの関数 `imsls_d_arma` を呼び出します。

**IMSL マッピング** TS\_ARMA\_CONST の引数は、次のように IMSL ライブラリ関数 `imsls_d_arma` にマップします。

```
params = imsls_d_arma(n_objs, z, p, q, IMSLS_CONSTANT,
method_id, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含まます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**p** = 1

**q** = 1

**MethodID** TS\_ARMA\_CONST の *method* 引数にマップします。

関数 `imsls_d_arma` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

**例 1** この例は、入力データ・テーブル、TS\_ARMA\_CONST 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-29 : 入力データ・テーブル DATASET**

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |
| 6    | 2.73657  |
| 7    | 2.64233  |
| 8    | 4.26118  |
| 9    | 3.13641  |
| 10   | 4.16566  |
| 11   | 2.95952  |
| 12   | 2.14504  |
| 13   | 1.98799  |
| 14   | 0.805859 |

| ロー番号 | データ       |
|------|-----------|
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、最小 2 乗法を使用して、**data** カラムから推定定数を返します。

```
SELECT TS_ARMA_CONST(data,0) OVER (ORDER BY ROWNUM rows
 BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS
 res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-30 : TS\_ARMA\_CONST の例 1 で返される値**

| res      |
|----------|
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| 0.082077 |
| ...      |
| 0.082077 |

例 2

この例は、AR、MA、および定数パラメータの推定を返すサンプル・クエリを示します。配列内の AR および MA の最初の要素には、1 つの要素が含まれます。DATASET テーブルの表 4-29 (274 ページ) を参照してください。

```
SELECT TS_ARMA_AR(DATA,1,1,0) OVER (ORDER BY rownum
 ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
 FOLLOWING) AS ar_param, ts_arma_ma(data,1,1,0) OVER
 (ORDER BY rownum ROWS BETWEEN UNBOUNDED PRECEDING AND
 UNBOUNDED FOLLOWING) AS ma_param, ts_arma_const(data,0)
 OVER (ORDER BY rownum ROWS BETWEEN UNBOUNDED PRECEDING
 AND UNBOUNDED FOLLOWING) AS const_param FROM DATASET
```

Sybase IQ は、データの 50 のローを返します。各ローには同じ 3 つの値が含まれます。

**表 4-31 : TS\_ARMA\_CONST の例 2 で返される値**

| ar_param | ma_param | const_param |
|----------|----------|-------------|
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |

| ar_param | ma_param | const_param |
|----------|----------|-------------|
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |
| 0.898793 | 0.105075 | 0.082077    |
| ...      | ...      | ...         |
| 0.898793 | 0.105075 | 0.082077    |

## 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

## 参照

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ』

## TS\_ARMA\_MA 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

## 機能

自己回帰移動平均 (ARMA) モデルのパラメータの最小二乗近似を計算し、要求された自己回帰移動平均近似を返します。

## 構文

**TS\_ARMA\_MA** (*timeseries\_expression*, *ma\_count*, *ma\_elem*, *method*)  
**OVER** (*window-spec*)

## パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**ma\_count** 計算する自己回帰値の数を含む整数。

**ma\_elem** 計算された移動平均配列から返す要素を示す整数。整数は、0 よりも大きく、*ma\_count* 以下である必要があります。

**method** (オプション) 推定の計算に使用するプロシージャを示す整数。0 (デフォルト値) = 最小2乗法、1 = モーメント法。

**window-spec** TS\_ARMA\_MA は、OVER () 句が必要な OLAP 関数です。

**使用法** この時系列関数は、移動平均推定を表す倍精度浮動小数点値を返します。TS\_ARMA\_MA は、IMSL ライブラリの関数 `imsls_d_arma` を呼び出します。

**IMSL マッピング** TS\_ARMA\_MA の引数は、次のように IMSL ライブラリ関数 `imsls_d_arma` にマップします。

```
params = imsls_d_arma(n_objs, z, p, q, method_id, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**p** = 1

**q** ユーザ定義の集合関数引数 *ma\_count* にマップします。

**method\_id** TS\_ARMA\_MA の *method* 引数にマップします。

関数 `imsls_d_arma` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

**例** この例は、入力データ・テーブル、TS\_ARMA\_MA 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-32：入力データ・テーブル DATASET**

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |
| 6    | 2.73657  |
| 7    | 2.64233  |
| 8    | 4.26118  |
| 9    | 3.13641  |
| 10   | 4.16566  |
| 11   | 2.95952  |
| 12   | 2.14504  |
| 13   | 1.98799  |
| 14   | 0.805859 |
| 15   | 0.833405 |



| ロー番号 | データ       |
|------|-----------|
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、最小 2 乗法を使用して、**data** カラムの 1 つの値を含む配列の最初の要素を返します。

```
SELECT TS_ARMA_MA(data,1,1,0) OVER (ORDER BY rownum
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-33 : TS\_ARMA\_MA から返される値**

| res      |
|----------|
| 0.105075 |
| 0.105075 |
| 0.105075 |
| 0.105075 |
| 0.105075 |
| 0.105075 |
| 0.105075 |
| 0.105075 |
| 0.105075 |
| ...      |
| 0.105075 |

**標準と互換性**

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

**参照**

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_AUTOCORRELATION 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

**機能**

定常時系列のサンプル自己相関関数を計算します。

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文         | <b>TS_AUTOCORRELATION</b> ( <i>timeseries_expression</i> , <i>lagmax</i> , <i>lag_elem</i> )<br><b>OVER</b> ( <i>window-spec</i> )                                                                                                                                                                                                                                                                                                                                              |
| パラメータ      | <p><b>timeseries_expression</b> 数値式。一般的に、時系列の要素を含むカラム名。</p> <p><b>lagmax</b> 自己共分散の最大遅延、自己相関、および自己相関の標準誤差を示す整数。整数は、1 以上で、時系列の要素数よりも少ない必要があります。</p> <p><b>lag_elem</b> 自己相関配列のどの要素を返すかを指定する整数。整数は、0 よりも大きく、<i>lagmax</i> 以下である必要があります。</p> <p><b>window-spec</b> TS_AUTOCORRELATION は、OVER () 句が必要な OLAP 関数です。</p>                                                                                                                                                               |
| 使用法        | この時系列関数は、自己相関値を表す倍精度浮動小数点値を返します。TS_AUTOCORRELATION は、IMSL ライブラリの関数 <code>imsls_d_autocorrelation</code> を呼び出します。                                                                                                                                                                                                                                                                                                                                                                |
| IMSL マッピング | <p>TS_AUTOCORRELATION の引数は、次のように IMSL ライブラリ関数 <code>imsls_d_autocorrelation()</code> にマップします。</p> <pre>params = imsls_d_autocorrelation(n_objs, x[], lagmax, 0);</pre> <p><b>n_objs</b> 現在のウィンドウ・フレームのローの数を含みます。</p> <p><b>x[]</b> 現在のウィンドウ・フレームの <i>timeseries_expression</i> の値を含みます。</p> <p><b>lagmax</b> ユーザ定義の集合関数引数 <i>lag_max</i> にマップします。</p> <p>関数 <code>imsls_d_autocorrelation</code> が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。</p> |
| 例          | <p>この例は、入力データ・テーブル、TS_AUTOCORRELATION 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。</p>                                                                                                                                                                                                                                                                                                                          |

表 4-34 : 入力データ・テーブル DATASET

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |

| ロー番号 | データ       |
|------|-----------|
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |

| ロー番号 | データ     |
|------|---------|
| 43   | 2.77188 |
| 44   | 4.00312 |
| 45   | 4.21298 |
| 46   | 5.00413 |
| 47   | 4.74498 |
| 48   | 4.89621 |
| 49   | 3.93273 |
| 50   | 4.31592 |

次の SQL 文を実行すると、**data** カラムの時系列データの自己相関を含む配列の 2 番目の要素が返されます。

```
SELECT TS_AUTOCORRELATION(data,2,2) OVER (ORDER BY
ROWNUM rows BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-35 : TS\_AUTOCORRELATION から返される値**

| res      |
|----------|
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| 0.803659 |
| ...      |
| 0.803659 |

#### 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

#### 参照

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ』

## TS\_AUTO\_UNI\_AR 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

機能

単変量自己回帰時系列モデルの自動選択と適合を実行します。

構文

```
TS_AUTO_UNI_AR (timeseries_expression, ar_count, ar_elem, method)
OVER (window-spec)
```

パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**ar\_count** 計算する自己回帰値の数を含む整数。

**ar\_elem** 返す計算済み自己回帰値を指定する整数。整数は、0 よりも大きく、*ar\_count* 以下である必要があります。

**method** (オプション) AR 係数の計算に使用する方法を示す整数。

0 = モーメント法

1 = 最小 2 乗法 (デフォルト値)

2 = 最尤推定

**window-spec** TS\_AUTO\_UNI\_AR は、OVER () 句が必要な OLAP 関数です。

使用法

この時系列関数は、自己回帰近似を含む倍精度浮動小数点数を返します。TS\_AUTO\_UNI\_AR は、IMSL ライブラリの関数 `imsls_d_auto_uni_ar` を呼び出します。

IMSL マッピング

TS\_AUTO\_UNI\_AR の引数は、次のように IMSL ライブラリ関数 `imsls_d_auto_uni_ar` にマップします。

```
params = imsls_d_auto_uni_ar (n_objs, z[], maxlag, p,
method, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**maxlag** ユーザ定義の集合関数引数 *ar\_count* にマップします。

**p** 最小 AIC を含むモデルの自己回帰パラメータの数を表す出力パラメータ。

**method** ユーザ定義の集合関数引数 *method* にマップします。*ar\_elem* が *p* よりも大きく、IMSL ライブラリ時系列関数のエラー処理値が 0 に設定されている場合、IQ は NULL を返します。IMSL ライブラリ時系列関数のエラー処理値が 0 以外の値に設定されている場合、IQ は、*ar\_elem* が *p* よりも大きいことを示すエラー・メッセージを表示します。詳細については、「IMSL ライブラリ時系列関数のエラー処理」(126 ページ) を参照してください。

関数 `imsls_d_auto_uni_ar` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例は、入力データ・テーブル、`TS_AUTO_UNI_AR` 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (`DATASET`) を入力データとして使用します。`DATASET` テーブルには、時系列データの 50 のローが含まれています。

表 4-36 : 入力データ・テーブル DATASET

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |

| ロー番号 | データ       |
|------|-----------|
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、**data** カラムの 2 つの要素を含む配列の最初の要素を返します。

```
SELECT TS_AUTO_UNI_AR(data,2,1,0) OVER (ORDER BY ROWNUM
rows BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。



表 4-37 : TS\_AUTO\_UNI\_AR から返される値

| res      |
|----------|
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| ...      |
| 0.883453 |

## 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

## 参照

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_BOX\_COX\_XFORM 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

## 機能

前方または逆 Box-Cox ベキ変換を実行します。

## 構文

**TS\_BOX\_COX\_XFORM** (*timeseries\_expression*, *power* [, *shift* [, *inverse*]])  
**OVER** (*window-spec*)

## パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**power** Box-Cox ベキ変換の指数パラメータを表す倍精度浮動小数点値。

**shift** (オプション) シフト・パラメータを表す倍精度浮動小数点値。値は次の関係を満たす必要があります。

**min(timeseries)+shift>0**

*shift* のデフォルトは 0.0 です。

**inverse** (オプション) *tinyint* 値。1 に設定すると、IQ は逆変換を実行します。0 または NULL の場合、IQ は前方変換を実行します。デフォルト値は 0 です。

**window-spec** *TS\_BOX\_COX\_XFORM* は、無制限ウィンドウを伴う *OVER ()* 句が必要な OLAP 関数です。*TS\_BOX\_COX\_XFORM* は値ベースのウィンドウをサポートしません。たとえば、*OVER ()* 句で範囲指定子を使用できません。

使用法

*TS\_BOX\_COX\_XFORM* は、時系列の各要素の対応する計算済み変換値を返します。IMSL ライブラリの関数 *imsls\_d\_box\_cox\_transform* を呼び出します。

IMSL マッピング

*TS\_BOX\_COX\_XFORM* の引数は、次のように IMSL ライブラリ関数 *imsls\_d\_box\_cox\_transform* にマップします。

```
params = imsls_d_box_cox_transform(n_objs, z[, power,
 IMSLS_SHIFT, shift [, IMSLS_INVERSE], 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含まます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**power** ユーザ定義の集合関数引数 *power* にマップします。

**shift** ユーザ定義の集合関数引数 *shift* にマップします。

**IMSLS\_INVERSE** ユーザ定義の集合関数の引数 *inverse* が 1 の場合、IQ は *IMSLS\_INVERSE* で Box-Cox 変換を呼び出します。そうでない場合は、この引数は、関数呼び出しから除外されます。

関数 *imsls\_d\_box\_cox\_transform* が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例は、入力データ・テーブル、*TS\_BOX\_COX\_XFORM* 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (*BOX\_COX\_XFORM\_DATASET*) を入力データとして使用します。*BOX\_COX\_XFORM\_DATASET* テーブルには、時系列データの 13 のローが含まれています。

表 4-38：入力データ・テーブル *BOX\_COX\_XFORM\_DATASET*

| ロー番号 | データ |
|------|-----|
| 1    | 7   |
| 2    | 26  |
| 3    | 6   |
| 4    | 60  |

| ロー番号 | データ  |
|------|------|
| 5    | 78.5 |
| 6    | 1    |
| 7    | 29   |
| 8    | 15   |
| 9    | 52   |
| 10   | 74.3 |
| 11   | 11   |
| 12   | 56   |
| 13   | 8    |

次の SQL 文を実行すると、**data** カラムから Box-Cox ベキ変換を実行します。

```
SELECT TS_BOX_COX_XFORM(data,1.0,1.0,0) OVER (ORDER BY
rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM BOX_COX_XFORM_DATASET
```

Sybase IQ は次の 13 のローを返します。

**表 4-39 : TS\_BOX\_COX\_XFORM から返される値**

| res  |
|------|
| 8    |
| 27   |
| 7    |
| 61   |
| 79.5 |
| 2    |
| 30   |
| 16   |
| 53   |
| 75.3 |
| 12   |
| 57   |
| 9    |

#### 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

参照 『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」  
『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ』

## TS\_DIFFERENCE 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

**機能** 季節時系列と非季節時系列の差異を計算します。

**構文** `TS_DIFFERENCE (timeseries_expression, period1 [, period2 [, ...period 10]]) OVER (window-spec)`

**パラメータ** **timeseries\_expression** 数値式。一般的に、差異を計算する時系列の要素を含むカラム名。  
**period1 ... period10** 各間隔は、時系列の差異を計算する間隔を含む正数式です。少なくとも1つの間隔を指定する必要があり、最大10の間隔を指定できます。  
**window-spec** TS\_DIFFERENCE は、無制限ウィンドウを伴う OVER () 句が必要な OLAP 関数です。この関数は値ベースのウィンドウをサポートしません。たとえば、OVER () 句で範囲指定子を使用できません。

**使用法** 時系列の各要素に対して、TS\_DIFFERENCE は、時系列の対応する計算済み差異値を返します。IMSL ライブラリの関数 `imsls_d_difference` を呼び出します。

**IMSL マッピング** TS\_DIFFERENCE の引数は、次のように IMSL ライブラリ関数 `imsls_d_difference` にマップします。

```
params = imsls_d_difference(n_objs, z[], n_differences,
 periods [], 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**n\_differences** TS\_DIFFERENCE で定義された *period* 引数にマップします。

**period** TS\_DIFFERENCE で定義された *period* 引数の配列。

関数 `imsls_d_difference` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例は、入力データ・テーブル、`TS_DIFFERENCE` 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (`DATASET`) を入力データとして使用します。`DATASET` テーブルには、時系列データの 50 のローが含まれています。

**表 4-40 : 入力データ・テーブル DATASET**

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |

| ロー番号 | データ       |
|------|-----------|
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、data カラムのデータの差異を計算します。

```
SELECT TS_DIFFERENCE(data,1) OVER (ORDER BY ROWNUM rows
 BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS
res FROM DATASET
```

Sybase IQ は 50 のローを返します。

**表 4-41 : TS\_DIFFERENCE から返される値**

| res       |
|-----------|
| NULL      |
| 0.170336  |
| 0.191027  |
| 1.29692   |
| 0.801743  |
| -0.038988 |
| -0.09424  |

| <b>res</b> |
|------------|
| 1.61886    |
| -1.12477   |
| 1.02925    |
| -1.20614   |
| -0.814478  |
| -0.157049  |
| -1.18213   |
| 0.027546   |
| 1.45734    |
| -0.990302  |
| -0.833325  |
| -0.637229  |
| -0.08655   |
| -0.12594   |
| -0.122914  |
| -1.39596   |
| 0.919785   |
| -0.449474  |
| 0.037273   |
| -0.954345  |
| -0.562983  |
| 1.98379    |
| 0.88304    |
| -0.345265  |
| 0.934656   |
| 0.069088   |
| -0.249428  |
| 0.795766   |
| -1.8145    |
| 1.27016    |
| 1.39266    |
| -0.141794  |
| 0.934752   |
| 0.982506   |
| 0.330772   |
| -1.34311   |
| 1.23124    |

**res**

---

0.209869

---

0.791146

---

-0.259155

---

0.15124

---

-0.963484

---

0.383186

---

**注意** IMSL ライブラリが結果の最初のローに 数値以外 (NaN) の値を返したので、結果の最初のローは NULL です。

---

標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

参照

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_ESTIMATE\_MISSING 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

機能

時系列内の欠落値を推定し、元の時系列に組み込み、新しい時系列として返します。

構文

**TS\_ESTIMATE\_MISSING** (*timeseries\_expression*, *method*)  
**OVER** (*window-spec*)

パラメータ

**timeseries\_expression** 数値式。一般的に、差異を計算する時系列の要素を含むカラム名。NULL 以外の値を指定した場合、時系列のギャップ (関数によって計算される値) が反映されます。



**method** (オプション) 欠落値を判別するときを使用する方法を指定する整数。

- **0** (デフォルト) — ギャップの前の最後の 4 つの時系列値とギャップの後の最初の 4 つの値の中央値でギャップの欠落時系列監視を推定します。
- **1** — 3 次スプライン補間法を使用して欠落値を推定します。ここでも、補間はギャップの前の最後の 4 つの時系列値とギャップの後の最初の 4 つの値に対して行われます。
- **2** — ギャップの前の時系列が AR(1) プロセスによって適切に記述されることが想定されます。
- **3** — AR(p) モデルを使用して、one-step-ahead 予測によって欠落値を推定します。

**window-spec** TS\_ESTIMATE\_MISSING は、無制限ウィンドウを伴う OVER () 句が必要な OLAP 関数です。この関数は値ベースのウィンドウをサポートしません。たとえば、OVER () 句で範囲指定子を使用できません。

#### 使用法

TS\_ESTIMATE\_MISSING を使用して、4 つの推定法の 1 つを使用して欠落している等間隔の時間ポイントを推定します。TS\_ESTIMATE\_MISSING は、IMSL ライブラリの関数 `imsls_d_estimate_missing` を呼び出します。

時間ポイントのセットに 2 つ以上の連続した NULL 値が存在する場合、TS\_ESTIMATE\_MISSING は使用できません。時間ポイント・セットの最初または最後の 2 つの値が NULL の場合、NULL が返されます。

#### IMSL マッピング

TS\_ESTIMATE\_MISSING の引数は、次のように IMSL ライブラリ関数 `imsls_d_estimate_missing` にマップします。

```
params = imsls_d_estimate_missing(n_objs, tpoints[],
 z[], method, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**tpoints** 時間ポイントのシーケンス内の欠落値を示すインデックスの配列。

**z[]** 累積された *timeseries\_expression* (`next_value` への呼び出しで取得されます)。

**method** TS\_ESTIMATE\_MISSING で定義された *method* 引数にマップします。

関数 `imsls_d_estimate_missing` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例 この例は、入力データ・テーブル、TS\_ESTIMATE\_MISSING 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (EST\_MISSING\_DATASET) を入力データとして使用します。EST\_MISSING\_DATASET テーブルには、時系列データの 9 つのローが含まれています。

**表 4-42 : 入力データ・テーブル EST\_MISSING\_DATASET**

| ロー番号 | データ     |
|------|---------|
| 1    | 2.8223  |
| 2    | -0.5721 |
| 3    | 2.2771  |
| 4    | NULL    |
| 5    | 1.2648  |
| 6    | 1.0278  |
| 7    | 0.6991  |
| 8    | -1.7539 |
| 9    | -2.8875 |

次の SQL 文を実行すると、4 番目のローから欠落しているデータの値を推定します。

```
SELECT TS_ESTIMATE_MISSING(DATA,0) OVER (ORDER BY
rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM EST_MISSING_DATASET
```

Sybase IQ は、次の 9 つのローを返し、NULL 値を 1.0278 で置き換えます。

**表 4-43 : TS\_ESTIMATE\_MISSING から返される値**

| res     |
|---------|
| 2.8223  |
| -0.5721 |
| 2.2771  |
| 1.0278  |
| 1.2648  |
| 1.0278  |
| 0.6991  |
| -1.7539 |
| -2.8875 |

|        |                                                                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL2008</b> Sybase 拡張。</li> <li>• <b>Sybase</b> SQL Anywhere および Adaptive Server Enterprise 互換ではありません。</li> </ul> |
| 参照     | <p>『システム管理ガイド 第2巻』の「<a href="#">第2章 OLAP の使用</a>」</p> <p>『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ』</p>                                                         |

## TS\_LACK OF FIT 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

|            |                                                                                                                                                                                                                                                                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能         | 適切な相関関数を前提として単変量時系列または伝達関数の適合性不足テスト (LOF) を実行します。                                                                                                                                                                                                                                                                                                            |
| 構文         | <pre><b>TS_LACK OF FIT</b> (<i>timeseries_expression</i>, <i>p_value</i>, <i>q_value</i>, <i>lagmax</i>, [<i>tolerance</i>]) <b>OVER</b> (<i>window-spec</i>)</pre>                                                                                                                                                                                          |
| パラメータ      | <p><b>timeseries_expression</b> 数値式。一般的に、時系列の要素を含むカラム名。</p> <p><b>p_value</b> 自己回帰パラメータの数を含む整数。</p> <p><b>q_value</b> 移動平均パラメータの数を含む整数。</p> <p><b>lagmax</b> 相関関数の最大遅延を含む整数。</p> <p><b>tolerance</b> (オプションのパラメータ)。非リニアの最小2乗アルゴリズムの収束を判別するために使用する浮動小数点値レベル。デフォルト値は0です。</p> <p><b>window-spec</b> <b>TS_LACK_OF_FIT</b> は、<b>OVER ()</b> 句が必要な OLAP 関数です。</p> |
| 使用法        | この関数は、時系列の適合性不足統計 (q) を含む倍精度浮動小数点値を返します。 <b>TS_LACK_OF_FIT</b> は、IMSL ライブラリの関数 <code>imsls_d_lack_of_fit</code> を呼び出します。                                                                                                                                                                                                                                     |
| IMSL マッピング | <p><b>TS_LACK_OF_FIT</b> の引数は、次のように IMSL ライブラリ関数 <code>imsls_d_lack_of_fit</code> にマップします。</p> <pre>params = imsls_d_arma(n_objs, z[], p, q, IMSL_S_LEAST_SQUARES, IMSL_S_CONVERGENCE_TOLERANCE, tolerance,</pre>                                                                                                                                            |

```

 IMSL_RESIDUAL, &residual, 0);

 correlations = imsls_d_autocorrelation(n_objs-
 p+lagmax, residuals, lagmax, 0);

 result = imsls_d_lack_of_fit(n_objs, correlations,
 lagmax, npfree, 0);

```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**p** TS\_LACK\_OF\_FIT で定義された *p\_value* 引数にマップします。

**q** TS\_LACK\_OF\_FIT で定義された *q\_value* 引数にマップします。

**lagmax** TS\_LACK\_OF\_FIT で定義された *lagmax* 引数にマップします。

**npfree** p+q から派生します。

**tolerance** IMSLS\_CONVERGENCE\_TOLERANCE を使用するオプションの引数。NULL の場合、IMSL ライブラリはデフォルト値を適用し、IMLSL\_CONVERGENCE\_TOLERANCE は使用されません。

IMSL 関数 *imsls\_d\_lack\_of\_fit* が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例 この例は、入力データ・テーブル、TS\_LACK\_OF\_FIT 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-44 : 入力データ・テーブル DATASET**

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |
| 6    | 2.73657  |
| 7    | 2.64233  |
| 8    | 4.26118  |
| 9    | 3.13641  |
| 10   | 4.16566  |
| 11   | 2.95952  |
| 12   | 2.14504  |

| ロー番号 | データ       |
|------|-----------|
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |

| ロー番号 | データ     |
|------|---------|
| 49   | 3.93273 |
| 50   | 4.31592 |

次の SQL 文を実行すると、**data** カラムから適合性不足統計を返します。

```
SELECT TS_LACK_OF_FIT(data,1,1,5,0.225) OVER (ORDER BY
rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-45 : TS\_LACK\_OF\_FIT から返される値**

| res     |
|---------|
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| 3.96751 |
| ...     |
| 3.96751 |

**標準と互換性**

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

**参照**

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_LACK OF FIT\_P 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能         | 単変量時系列に対して適合性不足テストを実行します。この関数は、 $q$ を返すのではなく、 $q$ の $p$ 値を返す点を除き、「 <b>TS_LACK OF FIT 関数 [時系列]</b> 」(297 ページ)と同じです。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 構文         | <b>TS_LACK OF FIT_P</b> ( <i>timeseries_expression</i> , <i>p_value</i> , <i>q_value</i> , <i>lagmax</i> , [ <i>tolerance</i> ])<br><b>OVER</b> ( <i>window-spec</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| パラメータ      | <b>timeseries_expression</b> 数値式。一般的に、時系列の要素を含むカラム名。<br><b>p_value</b> 自己回帰パラメータの数を含む整数。<br><b>q_value</b> 移動平均パラメータの数を含む整数。<br><b>lagmax</b> 相関関数の最大遅延を含む整数。<br><b>tolerance</b> (オプションのパラメータ)。非リニアの最小 2 乗アルゴリズムの収束を判別するために使用する浮動小数点値レベル。デフォルト値は 0 です。<br><b>window-spec</b> <b>TS_LACK_OF_FIT_P</b> は、 <b>OVER ()</b> 句が必要な OLAP 関数です。                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 使用法        | この関数は、時系列の適合性不足統計 ( $q$ ) の $p$ 値を含む倍精度浮動小数点値を返します。 <b>TS_LACK_OF_FIT_P</b> は、IMSL ライブラリの関数 <b>imsls_d_lack_of_fit</b> を呼び出します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| IMSL マッピング | <b>TS_LACK_OF_FIT_P</b> の引数は、次のように IMSL ライブラリ関数 <b>imsls_d_lack_of_fit</b> にマップします。<br><pre>params = imsls_d_arma(n_objs, z[], p, q,                     IMSLS_LEAST_SQUARES,                     IMSLS_CONVERGENCE_TOLERANCE, tolerance,                     IMSL_RESIDUAL, &amp;residual, 0); correlations = imsls_d_autocorrelation(n_objs- p+lagmax, residuals, lagmax, 0); result = imsls_d_lack_of_fit(n_objs, correlations, lagmax, npfree, 0);</pre><br><b>n_objs</b> 現在のウィンドウ・フレームのローの数を含みます。<br><b>z[]</b> 現在のウィンドウ・フレームの <i>timeseries_expression</i> の値を含みます。<br><b>p</b> <b>TS_LACK_OF_FIT_P</b> で定義された <i>p_value</i> 引数にマップします。<br><b>q</b> <b>TS_LACK_OF_FIT_P</b> で定義された <i>q_value</i> 引数にマップします。<br><b>lagmax</b> <b>TS_LACK_OF_FIT_P</b> で定義された <i>lagmax</i> 引数にマップします。 |

**npfree** p + q から派生します。

**tolerance** IMSLS\_CONVERGENCE\_TOLERANCE を使用するオプションの引数。NULL の場合、IMSL ライブラリはデフォルト値を適用し、IMSL\_CONVERGENCE\_TOLERANCE は使用されません。

IMSL 関数 `imsls_d_lack_of_fit` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

例

この例は、入力データ・テーブル、`TS_LACK_OF_FIT_P` 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (`DATASET`) を入力データとして使用します。`DATASET` テーブルには、時系列データの 50 のローが含まれています。

表 4-46 : 入力データ・テーブル DATASET

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |



| ロー番号 | データ       |
|------|-----------|
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、**data** カラムから適合性不足統計の **p** 値を返します。

```
SELECT TS_LACK_OF_FIT_P(data,1,1,5,0.225) OVER (ORDER
BY rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-47 : TS\_LACK\_OF\_FIT\_P から返される値**

**res**

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

0.735006

---

...

---

0.735006

参照

『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2/2巻 C 統計ライブラリ』

## TS\_MAX\_ARMA\_AR 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

機能

単変量 ARMA ( 自己回帰移動平均 ) 時系列モデルにある引数の正確な最尤推定を計算し、要求された自己回帰推定を返します。

構文

**TS\_MAX\_ARMA\_AR** (*timeseries\_expression*, *ar\_count*, *ar\_elem*)  
**OVER** (*window-spec*)

パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**ar\_count** 計算する自己回帰値の数を含む整数。

**ar\_elem** 計算された自己回帰配列のどの要素を返すかを指定する整数。整数は、0 よりも大きく、*ar\_count* 以下である必要があります。

**window-spec** TS\_MAX\_ARMA\_AR は、OVER () 句が必要な OLAP 関数です。

#### 使用法

この時系列関数は、自己回帰近似を含む倍精度浮動小数点値を返します。TS\_MAX\_ARMA\_AR は、IMSL ライブラリの関数 `imsls_d_max_arma` を呼び出します。

#### IMSL マッピング

TS\_MAX\_ARMA\_AR の引数は、次のように IMSL ライブラリ関数 `imsls_d_max_arma` にマップします。

```
params = imsls_d_max_arma(n_objs, z[], p, q, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含まます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含まます。

**p** *ar\_count* 引数にマップします。

**q** = 1

IMSL 関数 `imsls_d_max_arma` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

#### 例 1

この例は、入力データ・テーブル、TS\_MAX\_ARMA\_AR 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-48 : 入力データ・テーブル DATASET**

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |
| 6    | 2.73657  |
| 7    | 2.64233  |
| 8    | 4.26118  |
| 9    | 3.13641  |
| 10   | 4.16566  |
| 11   | 2.95952  |
| 12   | 2.14504  |
| 13   | 1.98799  |
| 14   | 0.805859 |

| ロー番号 | データ       |
|------|-----------|
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、**data** カラムのデータの 2 つの自己回帰推定を含む配列の 2 番目の要素が返されます。

```
SELECT TS_MAX_ARMA_AR(data,2,2) OVER (ORDER BY rownum
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-49 : TS\_MAX\_ARMA\_AR の例 1 で返される値**

| res      |
|----------|
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| 0.179748 |
| ...      |
| 0.179748 |

## 例 2

この例は、**DATASET** テーブルから結果の 2 つのカラム (自己回帰推定の 1 番目と 2 番目の要素) を返すサンプル・クエリを示します。**DATASET** テーブルの表 4-48 (305 ページ) を参照してください。

```
SELECT TS_MAX_ARMA_AR(data,2,1) OVER (ORDER BY rownum
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS ar_elem1, TS_MAX_ARMA_AR(data,2,2) OVER
(ORDER BY rownum ROWS BETWEEN UNBOUNDED PRECEDING AND
UNBOUNDED FOLLOWING) AS ar_elem2 FROM DATASET
```

Sybase IQ は、データの 50 のローを返します。各ローには同じ 2 つの値が含まれます。

**表 4-50 : TS\_MAX\_ARMA\_AR の例 2 で返される値**

| ar_elem1 | ar_elem2 |
|----------|----------|
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |

| ar_elem1 | ar_elem2 |
|----------|----------|
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| 0.731164 | 0.179748 |
| ...      | ...      |
| 0.731164 | 0.179748 |

標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

参照

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_MAX\_ARMA\_CONST 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

機能

単変量 ARMA (自己回帰移動平均) 時系列モデルにある引数の正確な最尤推定を計算し、定数推定を返します。

構文

**TS\_MAX\_ARMA\_CONST** (*timeseries\_expression*)  
**OVER** (*window-spec*)

パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**window-spec** TS\_MAX\_ARMA\_CONST は、OVER () 句が必要な OLAP 関数です。

使用法

この時系列関数は、定数推定を含む倍精度浮動小数点値を返します。TS\_MAX\_ARMA\_CONST は、IMSL ライブラリの関数 `imsls_d_arma` を呼び出します。

## IMSL マッピング

TS\_MAX\_ARMA\_CONST 引数は、次のように IMSL ライブラリ関数 `imsls_d_arma` にマップします。

```
params = imsls_d_max_arma(n_objs, z, p, q, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**p** = 1

**q** = 1

IMSL 関数 `imsls_d_arma` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

## 例

この例は、入力データ・テーブル、TS\_MAX\_ARMA\_CONST 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-51 : 入力データ・テーブル DATASET**

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |

| ロー番号 | データ       |
|------|-----------|
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、`data` カラムからデータの最尤推定自己回帰計算の定数推定を返します。

```
SELECT TS_MAX_ARMA_CONST(data) OVER (ORDER BY rownum
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```



Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-52 : TS\_MAX\_ARMA\_CONST から返される値**

| res      |
|----------|
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| 0.107555 |
| ...      |
| 0.107555 |

**標準と互換性**

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

**参照**

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_MAX\_ARMA\_LIKELIHOOD 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

**機能**

単変量 ARMA ( 自己回帰移動平均 ) 時系列モデルにある引数の正確な最尤推定を計算し、適合モデルの尤度値 (ln) を返します。

**構文**

**TS\_MAX\_ARMA\_LIKELIHOOD** (*timeseries\_expression*)  
**OVER** (*window-spec*)

**パラメータ**

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**window-spec** TS\_MAX\_ARMA\_LIKELIHOOD は、OVER () 句が必要な OLAP 関数です。

**使用法** この時系列関数は、 $-2*(\ln(\text{likelihood}))$  の値を含む倍精度浮動小数点値を返します。TS\_MAX\_ARMA\_LIKELIHOOD は、IMSL ライブラリの関数 `imsls_d_max_arma` を呼び出します。

**IMSL マッピング** TS\_MAX\_ARMA\_LIKELIHOOD の引数は、次のように IMSL ライブラリ関数 `imsls_d_max_arma` にマップします。

```
params = imsls_d_max_arma(n_objs, z, p, q,
 IMSLS_LOG_LIKELIHOOD, &likelihood, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含みます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

**p** = 1

**q** = 1

**likelihood** 関数呼び出しによって提供されます。ログ推定結果を含みます。

IMSL 関数 `imsls_d_max_arma` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

**例** この例は、入力データ・テーブル、TS\_MAX\_ARMA\_LIKELIHOOD 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-53 : 入力データ・テーブル DATASET**

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |
| 6    | 2.73657  |
| 7    | 2.64233  |
| 8    | 4.26118  |
| 9    | 3.13641  |
| 10   | 4.16566  |
| 11   | 2.95952  |

| ロー番号 | データ       |
|------|-----------|
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |

| ロー番号 | データ     |
|------|---------|
| 49   | 3.93273 |
| 50   | 4.31592 |

次の SQL 文を実行すると、**data** カラムからデータの最尤推定の推定値を返します。

```
SELECT TS_MAX_ARMA_LIKELIHOOD(data) OVER (ORDER BY
ROWNUM rows BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-54 : TS\_MAX\_ARMA\_LIKELIHOOD から返される値**  
**res**

|          |
|----------|
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| -11.7818 |
| ...      |
| -11.7818 |

標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

参照

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_MAX\_ARMA\_MA 関数 [時系列]

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能         | 単変量 ARMA (自己回帰移動平均) 時系列モデルにある引数の正確な最尤推定を計算し、要求された移動平均推定を返します。                                                                                                                                                                                                                                                                                                                                                                                     |
| 構文         | <b>TS_MAX_ARMA_MA</b> ( <i>timeseries_expression</i> , <i>ma_count</i> , <i>ma_elem</i> )<br><b>OVER</b> ( <i>window-spec</i> )                                                                                                                                                                                                                                                                                                                   |
| パラメータ      | <b>timeseries_expression</b> 数値式。一般的に、時系列の要素を含むカラム名。<br><b>ma_count</b> 計算する自己回帰値の数を含む整数。<br><b>ma_elem</b> 計算された移動平均配列から返す要素を示す整数。整数は、0 よりも大きく、 <i>ma_count</i> 以下である必要があります。<br><b>window-spec</b> TS_MAX_ARMA_MA は、OVER () 句が必要な OLAP 関数です。                                                                                                                                                                                                    |
| 使用法        | この時系列関数は、自己回帰近似を含む倍精度浮動小数点値を返します。TS_MAX_ARMA_MA は、IMSL ライブラリの関数 <code>imsls_d_max_arma</code> を呼び出します。                                                                                                                                                                                                                                                                                                                                            |
| IMSL マッピング | TS_MAX_ARMA_MA の引数は、次のように IMSL ライブラリ関数 <code>imsls_d_max_arma</code> にマップします。<br><pre>params = imsls_d_max_arma(n_objs, z[], p, q, 0);</pre> <b>n_objs</b> 現在のウィンドウ・フレームのローの数を含みます。<br><b>z[]</b> 現在のウィンドウ・フレームの <i>timeseries_expression</i> の値を含みます。<br><b>p</b> = 1<br><b>q</b> TS_MAX_ARMA_MA 引数 <i>ma_count</i> にマップします。<br>IMSL 関数 <code>imsls_d_max_arma</code> が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。 |
| 例          | この例は、入力データ・テーブル、TS_MAX_ARMA_MA 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。                                                                                                                                                                                                                                                                                                       |

表 4-55 : 入力データ・テーブル DATASET

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |

| ロー番号 | データ      |
|------|----------|
| 37   | 0.616094 |
| 38   | 2.00875  |
| 39   | 1.86696  |
| 40   | 2.80171  |
| 41   | 3.78422  |
| 42   | 4.11499  |
| 43   | 2.77188  |
| 44   | 4.00312  |
| 45   | 4.21298  |
| 46   | 5.00413  |
| 47   | 4.74498  |
| 48   | 4.89621  |
| 49   | 3.93273  |
| 50   | 4.31592  |

次の SQL 文を実行すると、**data** カラムからデータの最尤推定の移動平均を返します。

```
SELECT TS_MAX_ARMA_MA(DATA,5,4) OVER (ORDER BY ROWNUM
rows BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-56 : TS\_MAX\_ARMA\_MA から返される値**

| res       |
|-----------|
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| -0.035006 |
| ...       |
| -0.035006 |

|        |                                                                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL2008</b> Sybase 拡張。</li> <li>• <b>Sybase</b> SQL Anywhere および Adaptive Server Enterprise 互換ではありません。</li> </ul> |
| 参照     | <p>『システム管理ガイド 第 2 巻』の「<a href="#">第 2 章 OLAP の使用</a>」</p> <p>『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』</p>                                                   |

## TS\_OUTLIER\_IDENTIFICATION 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | 異常値の検出および判定を行うと同時に、異常値のない基礎系列が一般的な季節または非季節 ARMA モデルに従う時系列におけるモデル引数を推定します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 構文    | <p><b>TS_OUTLIER_IDENTIFICATION</b> (<i>timeseries_expression</i>, <i>p_value</i>, <i>q_value</i>, <i>s_value</i>, <i>d_value</i>, [, <i>delta_value</i>], <i>critical_value</i>)]</p> <p><b>OVER</b> (<i>window-spec</i>)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| パラメータ | <p><b>timeseries_expression</b> 数値式。一般的に、時系列の要素を含むカラム名。</p> <p><b>p_value</b> 異常値のない系列に従う自己回帰統合移動平均 (ARIMA) <math>(p, 0, q)x(0, d, 0)_s</math> モデルの <i>p</i> 部分を含む整数。</p> <p><b>q_value</b> 異常値のない系列に従う ARIMA <math>(p, 0, q)x(0, d, 0)_s</math> モデルの <i>q</i> 部分を含む整数。</p> <p><b>s_value</b> 異常値のない系列に従う ARIMA <math>(p, 0, q)x(0, d, 0)_s</math> モデルの <i>s</i> 部分を含む整数。</p> <p><b>d_value</b> 異常値のない系列に従う ARIMA <math>(p, 0, q)x(0, d, 0)_s</math> モデルの <i>d</i> 部分を含む整数。</p> <p><b>delta_value</b> (オプション) テンポラリ変更異常値の検出に使用する抑制効果パラメータを含む倍精度浮動値。整数は、0 よりも大きく、1 よりも小さい必要があります。デフォルト値は 0.7 です。</p> <p><b>critical_value</b> (オプション) 異常値の検出のしきい値として使用される倍精度浮動値。デフォルトは 3.0 です。</p> |



**window-spec** TS\_OUTLIER\_IDENTIFICATION は、無制限ウィンドウを伴う OVER () 句が必要な OLAP 関数です。この関数は値ベースのウィンドウをサポートしません。たとえば、OVER () 句で範囲指定子を使用できません。

#### 使用法

この関数は、異常値のない時系列を返します。  
TS\_OUTLIER\_IDENTIFICATION は、IMSL ライブラリの関数 `imsls_d_ts_outlier_identification` を呼び出します。

#### IMSL マッピング

TS\_OUTLIER\_IDENTIFICATION の引数は、次のように IMSL ライブラリ関数 `imsls_d_ts_outlier_identification` にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs,
model[], z[], 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含まます。

**model** TS\_OUTLIER\_IDENTIFICATION 引数 *p\_value*、*s\_value*、*q\_value*、*d\_value* を含む配列。

```
model[0] = p_value;
model[1] = s_value;
model[2] = q_value;
model[3] = d_value;
```

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含みます。

*delta\_value* が NULL でない場合、TS\_OUTLIER\_IDENTIFICATION の引数は、次に示すように IMSL ライブラリ関数 `imsls_d_ts_outlier_identification` にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs,
model[], z[], IMSL_DELTA, delta_value, 0);
```

*critical\_value* が NULL でない場合、TS\_OUTLIER\_IDENTIFICATION の引数は、次に示すように IMSL ライブラリ関数 `imsls_d_ts_outlier_identification` にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs,
model[], z[], IMSL_CRITICAL, critical_value, 0);
```

*delta\_value* と *critical\_value* の両方が NULL でない場合、TS\_OUTLIER\_IDENTIFICATION の引数は、次に示すように IMSL ライブラリ関数 `imsls_d_ts_outlier_identification` にマップします。

```
params = imsls_d_ts_outlier_identification(n_objs,
model[], z[], IMSL_DELTA, delta_value, IMSL_CRITICAL,
critical_value, 0);
```

IMSL 関数 `imsls_d_ts_outlier_identification` が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例 この例は、入力データ・テーブル、`TS_OUTLIER_IDENTIFICATION` 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

**表 4-57 : 入力データ・テーブル DATASET**

| ロー番号 | データ       |
|------|-----------|
| 1    | 0.315523  |
| 2    | 0.485859  |
| 3    | 0.676886  |
| 4    | 1.97381   |
| 5    | 2.77555   |
| 6    | 2.73657   |
| 7    | 2.64233   |
| 8    | 4.26118   |
| 9    | 3.13641   |
| 10   | 4.16566   |
| 11   | 2.95952   |
| 12   | 2.14504   |
| 13   | 1.98799   |
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |

| ロー番号 | データ       |
|------|-----------|
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、**data** カラムのデータの異常値を検出して判別します。

```
SELECT TS_OUTLIER_IDENTIFICATION(data,1,1,1,1,0.7,3.0)
OVER (ORDER BY rownum ROWS BETWEEN UNBOUNDED PRECEDING
AND UNBOUNDED FOLLOWING) AS res FROM DATASET
```

Sybase IQ は 50 のローを返します。

**表 4-58 : TS\_OUTLIER\_IDENTIFICATION から返される値**

| res      |
|----------|
| 0.315523 |
| 0.485859 |
| 0.676886 |
| 1.97381  |
| 2.77555  |
| 2.73657  |

**res**

---

2.64233

---

4.26118

---

3.13641

---

4.16566

---

2.95952

---

2.14504

---

1.98799

---

0.805859

---

0.833405

---

2.29075

---

1.30045

---

0.467122

---

-0.170107

---

-0.256657

---

-0.382597

---

-0.505511

---

-1.90147

---

-0.981688

---

-1.43116

---

-1.39389

---

-2.34823

---

-2.91122

---

-0.927423

---

-0.044383

---

-0.389648

---

0.545008

---

0.614096

---

0.364668

---

1.16043

---

-0.654063

---

0.616094

---

2.00875

---

1.86696

---

2.80171

---

3.78422

---

4.11499

---

2.77188

**res**

4.00312

4.21298

5.00413

4.74498

4.89621

3.93273

4.31592

## 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

## 参照

『システム管理ガイド 第 2 巻』の「第 2 章 OLAP の使用」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

**TS\_PARTIAL\_AUTOCORRELATION 関数 [時系列]**

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

## 機能

定常時系列のサンプル偏自己相関関数を計算します。

## 構文

**TS\_PARTIAL\_AUTOCORRELATION** (*timeseries\_expression*, *lagmax*, *lag\_elem*)

**OVER** (*window-spec*)

## パラメータ

**timeseries\_expression** 数値式。一般的に、時系列の要素を含むカラム名。

**lagmax** 計算する自己共分散の最大遅延、自己相関、および自己相関の標準誤差を示す整数。整数は、1 以上で、時系列の要素数よりも少ない必要があります。

**lag\_elem** 返す自己相関の要素を識別する整数。整数は、0 よりも大きく、*lagmax* 以下である必要があります。

**window-spec** **TS\_PARTIAL\_AUTOCORRELATION** は、**OVER ()** 句が必要な OLAP 関数です。

使用法

この関数は、異常値のない時系列を返します。  
 TS\_PARTIAL\_AUTOCORRELATION は、IMSL ライブラリの関数  
 imsls\_d\_autocorrelation および imsls\_d\_partial\_autocorrelation を呼び出します。

IMSL マッピング

TS\_PARTIAL\_AUTOCORRELATION の引数は、次のように IMSL ライブラリ関数 imsls\_d\_autocorrelation および imsls\_d\_partial\_autocorrelation にマップします。

```
params = imsls_d_autocorrelation(n_objs, z[], lagmax,
0);

result = imsls_d_partial_autocorrelation(lagmax,
params, 0);
```

**n\_objs** 現在のウィンドウ・フレームのローの数を含まます。

**z[]** 現在のウィンドウ・フレームの *timeseries\_expression* の値を含まます。

**lagmax** TS\_PARTIAL\_AUTOCORRELATION 引数 *lagmax* にマップします。

IMSL 関数 imsls\_d\_autocorrelation および imsls\_d\_partial\_autocorrelation が時系列計算を実行する方法の詳細については、『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』を参照してください。

例

この例は、入力データ・テーブル、TS\_PARTIAL\_AUTOCORRELATION 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (DATASET) を入力データとして使用します。DATASET テーブルには、時系列データの 50 のローが含まれています。

表 4-59 : 入力データ・テーブル DATASET

| ロー番号 | データ      |
|------|----------|
| 1    | 0.315523 |
| 2    | 0.485859 |
| 3    | 0.676886 |
| 4    | 1.97381  |
| 5    | 2.77555  |
| 6    | 2.73657  |
| 7    | 2.64233  |
| 8    | 4.26118  |
| 9    | 3.13641  |
| 10   | 4.16566  |
| 11   | 2.95952  |
| 12   | 2.14504  |
| 13   | 1.98799  |

| ロー番号 | データ       |
|------|-----------|
| 14   | 0.805859  |
| 15   | 0.833405  |
| 16   | 2.29075   |
| 17   | 1.30045   |
| 18   | 0.467122  |
| 19   | -0.170107 |
| 20   | -0.256657 |
| 21   | -0.382597 |
| 22   | -0.505511 |
| 23   | -1.90147  |
| 24   | -0.981688 |
| 25   | -1.43116  |
| 26   | -1.39389  |
| 27   | -2.34823  |
| 28   | -2.91122  |
| 29   | -0.927423 |
| 30   | -0.044383 |
| 31   | -0.389648 |
| 32   | 0.545008  |
| 33   | 0.614096  |
| 34   | 0.364668  |
| 35   | 1.16043   |
| 36   | -0.654063 |
| 37   | 0.616094  |
| 38   | 2.00875   |
| 39   | 1.86696   |
| 40   | 2.80171   |
| 41   | 3.78422   |
| 42   | 4.11499   |
| 43   | 2.77188   |
| 44   | 4.00312   |
| 45   | 4.21298   |
| 46   | 5.00413   |
| 47   | 4.74498   |
| 48   | 4.89621   |
| 49   | 3.93273   |
| 50   | 4.31592   |

次の SQL 文を実行すると、**data** カラムの偏自己相関含む配列の最初の要素を返します。

```
SELECT TS_PARTIAL_AUTOCORRELATION(data,1,1) OVER (ORDER
BY rownum ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED
FOLLOWING) AS res FROM DATASET
```

Sybase IQ は、50 のローを返します。各ローには同じ値が含まれます。

**表 4-60 : TS\_PARTIAL\_AUTOCORRELATION から返される値**

| res      |
|----------|
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| 0.883453 |
| ...      |
| 0.883453 |

標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

参照

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』

## TS\_VWAP 関数 [時系列]

---

**注意** この関数は、RAP - The Trading Edition Enterprise でのみ使用できます。

---



**機能** VWAP は、出来高加重平均価格 (volume-weighted average price) を意味します。TS\_VWAP は、特定の対象期間の取引値の総出来高に対する比率を計算します。VWAP は、証券の定義済みの取引期間における平均価格の評価基準です。TS\_VWAP は、単純な集合関数としても OLAP スタイルの集合関数としても使用できます。

その他の時系列関数とは異なり、TS\_VWAP は IMSL ライブラリを呼び出しません。

**構文 1** `TS_VWAP (price_expression, volume_expression)`

**構文 2** `TS_VWAP (price_expression, volume_expression)`

`OVER (window-spec)`

**パラメータ**

**price\_expression** 出来高加重平均に組み込む価格を指定する数値式。

**volume\_expression** 出来高加重平均の計算で使用する出来高を指定する数値式。

**window-spec** 構文 2 で使用した場合、TS\_VWAP は OVER () 句が必要な OLAP 関数です。

**使用法** Sybase IQ は、次の式を使用して TS\_VWAP を計算します。

**図 4-1 : VWAP の計算**

$$P_{vwap} = \frac{\sum_j P_j \cdot Q_j}{\sum_j Q_j}$$

$P_{vwap}$  = 出来高加重平均価格。

$P_j$  = 取引価格  $j$ 。

$Q_j$  = 取引高  $j$ 。

$j$  = 対象期間内に発生した 1 つの取引。

**例** この例は、入力データ・テーブル、TS\_VWAP 関数を含む SQL 文、および関数から返されるデータ値を示します。この例は、次のテーブル (VWAP\_DATASET) を入力データとして使用します。VWAP\_DATASET テーブルには、時系列データの 3 つのローが含まれています。

**表 4-61 : 入力データ・テーブル VWAP\_DATASET**

| ロー番号 | 価格 | 出来高 |
|------|----|-----|
| 1    | 1  | 1   |
| 2    | 2  | 2   |
| 3    | 5  | 1   |

次の SQL 文を実行すると、出来高加重平均価格を計算します。

```
SELECT TS_VWAP(price,volume) OVER (ORDER BY rownum ROWS
 BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS
 res FROM VWAP_DATASET
```

Sybase IQ は 3 つのローを返します。

**表 4-62 : TS\_VWAP から返される値**

| res |
|-----|
| 2.5 |
| 2.5 |
| 2.5 |

#### 標準と互換性

- **SQL2008** Sybase 拡張。
- **Sybase** SQL Anywhere および Adaptive Server Enterprise 互換ではありません。

#### 参照

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

## UCASE 関数 [文字列]

#### 機能

文字列内のすべての文字を大文字に変換します。

#### 構文

**UCASE** ( *string-expression* )

#### パラメータ

**string-expression** 大文字に変換される文字列。

詳細については、「[REPLACE 関数 \[文字列\]](#)」(243 ページ)を参照してください。

#### 例

次の文を実行すると、値 “CHOCOLATE” が返ります。

```
SELECT UCASE('ChocoLate') FROM iq_dummy
```

#### 使用法

UCASE 関数の結果データ型は、LONG VARCHAR です。SELECT INTO 文で UCASE を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して UCASE を正しいデータ型とサイズに設定する必要があります。

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise では UCASE がサポートされていませんが、UPPER が同じ機能を持っています。

#### 参照

[「LCASE 関数 \[文字列\]](#)」(196 ページ)

[「UPPER 関数 \[文字列\]](#)」(329 ページ)

## UPPER 関数 [文字列]

|        |                                                                                                                                            |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | 文字列内のすべての文字を大文字に変換します。                                                                                                                     |
| 構文     | <b>UPPER</b> ( <i>string-expression</i> )                                                                                                  |
| パラメータ  | <b>string-expression</b> 大文字に変換される文字列。<br>詳細については、「 <a href="#">REPLACE 関数 [文字列]</a> 」を参照してください。                                           |
| 例      | 次の文を実行すると、値 “CHOCOLATE” が返ります。<br><pre>SELECT UPPER( 'ChocoLate' ) FROM iq_dummy</pre>                                                     |
| 使用法    | UPPER 関数の結果データ型は LONG VARCHAR です。SELECT INTO 文で UPPER を使用する場合は、ラージ・オブジェクト管理オプションのライセンスを所有しているか、CAST を使用して UPPER を正しいデータ型とサイズに設定する必要があります。  |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> この関数は SQL92 互換です。</li> <li>• <b>Sybase</b> Adaptive Server Enterprise 互換。</li> </ul> |
| 参照     | <a href="#">「LCASE 関数 [文字列]」 (196 ページ)</a><br><a href="#">「LOWER 関数 [文字列]」 (201 ページ)</a><br><a href="#">「UCASE 関数 [文字列]」 (328 ページ)</a>     |

## USER\_ID 関数 [システム]

|        |                                                                                                                                                              |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能     | ユーザ ID 番号を整数で返します。                                                                                                                                           |
| 構文     | <b>USER_ID</b> ( [ <i>user-name</i> ] )                                                                                                                      |
| パラメータ  | <b>user-name</b> ユーザ名。                                                                                                                                       |
| 例      | 次の文は、ユーザ ID 番号 1 を返します。<br><pre>SELECT USER_ID ( 'DBA' ) FROM iq_dummy</pre><br>次の文は、ユーザ ID 番号 0 を返します。<br><pre>SELECT USER_ID ( 'SYS' ) FROM iq_dummy</pre> |
| 標準と互換性 | <ul style="list-style-type: none"> <li>• <b>SQL92</b> ベンダの拡張機能。</li> <li>• <b>Sybase</b> Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。</li> </ul>       |
| 参照     | <a href="#">「SUSER_ID 関数 [システム]」 (267 ページ)</a><br><a href="#">「USER_NAME 関数 [システム]」 (330 ページ)</a>                                                            |

## USER\_NAME 関数[システム]

機能 ユーザ名を返します。

構文 **USER\_NAME** ([ *user-id* ])

パラメータ **user-id** ユーザ ID 番号。

例 次の文は、値 “DBA” を返します。

```
SELECT USER_NAME (1) FROM iq_dummy
```

次の文は、値 “SYS” を返します。

```
SELECT USER_NAME (0) FROM iq_dummy
```

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Sybase IQ 用に実装された Adaptive Server Enterprise 関数です。Adapter Server Enterprise では、USER\_NAME はサーバのユーザ名ではなく、ユーザ名を返します。

参照

[「USER\\_NAME 関数 \[システム\]」 \(267 ページ\)](#)

[「USER\\_ID 関数 \[システム\]」 \(329 ページ\)](#)

## UUIDTOSTR 関数 [文字列]

機能 一意な識別子 (UUID、GUID と呼ばれる) の値を、文字列値に変換します。

構文 **UUIDTOSTR** ( *uuid-expression* )

パラメータ **uuid-expression** 一意な識別子の値。

例 一意な識別子の値を読みやすい形式に変換するには、次のようなクエリを実行します。

```
CREATE TABLE T3 (
pk uniqueidentifier primary key, c1 int);
INSERT INTO T3 (pk, c1)
values (0x12345678123456789012123456789012, 1)
SELECT UUIDTOSTR(pk) FROM T3
```

使用法

一意な識別子を、*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx* の形式の文字列値に変換します (x は 16 進の桁)。バイナリ値が一意な識別子として有効でない場合は、NULL が返されます。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **SQL99** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

- 参照 [「NEWID 関数 \[その他\]」 \(208 ページ\)](#)  
[「STRTOUUID 関数 \[文字列\]」 \(264 ページ\)](#)  
[「バイナリ・データ型」 \(77 ページ\)](#) の「UNIQUEIDENTIFIER」

## VAR\_POP 関数 [集合]

**機能** 1 つの数値式で構成される母集団の統計分散を DOUBLE 型として計算します。

**構文** `VAR_POP ([ ALL ] expression)`

**パラメータ** **expression** その母集団ベースの分散がローのセットに対して計算される式 (通常はカラム名)。

**例** 次の文では、複数の期間における各注文の項目数の平均および分散を示します。

```
SELECT year(ShipDate) AS Year, quarter(ShipDate)
 AS Quarter, AVG(Quantity) AS Average,
 VAR_POP(Quantity) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter
```

| 年    | 四半期 | 平均        | 偏差       |
|------|-----|-----------|----------|
| 2000 | 1   | 25.775148 | 203.9021 |
| 2000 | 2   | 27.050847 | 225.8109 |
| ...  | ... | ...       | ...      |

**使用法** グループまたはパーティションの各ロー (DISTINCT が指定されている場合、重複が削除された後に残る各ロー) に対して評価される、指定された *value expression* の母分散を計算します。これは、*value expression* から、グループまたはパーティション内の (残りの) ローの数で割られた *value expression* の平均を差し引いた値の 2 乗和として定義されます。

母集団ベースの分散は、次の式に従って計算されます。

$$\frac{\sum (x_i - \bar{x})^2}{n}$$

- 標準と互換性**
- **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
  - **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

「分析関数」(106 ページ)

『システム管理ガイド 第2巻』の「第2章 OLAP の使用」

## VAR\_SAMP 関数 [集合]

機能

1 つの数値式で構成される標本の統計分散を DOUBLE 型として計算します。

**注意** VAR\_SAMP は VARIANCE のエイリアスです。

構文

**VAR\_SAMP** ([ ALL ] *expression*)

パラメータ

**expression** その標本ベースの分散がローのセットに対して計算される式(通常はカラム名)。

例

次の文では、複数の期間における各注文の項目数の平均および分散を示します。

```
SELECT year(ShipDate) AS Year, quarter(ShipDate)
 AS Quarter, AVG(Quantity) AS Average,
 VAR_SAMP(Quantity) AS Variance
FROM SalesOrderItems GROUP BY Year, Quarter
ORDER BY Year, Quarter
```

| 年    | 四半期 | 平均        | 偏差       |
|------|-----|-----------|----------|
| 2000 | 1   | 25.775148 | 205.1158 |
| 2000 | 2   | 27.050847 | 227.0939 |
| ...  | ... | ...       | ...      |

使用法

グループまたはパーティションの各ロー (DISTINCT が指定されている場合、重複が削除された後に残る各ロー) に対して評価される *value expression* の標本分散を計算します。これは、*value expression* から、グループまたはパーティション内の残りのローより 1 少ない数で割った *value expression* の平均を差し引いた値の 2 乗和として定義されます。

Q 12.7 以降で 1 要素の入力セットの場合、NULL によって NULL が返されます。12.7 よりも前のバージョンの場合、NULL はゼロを返します。

分散は次の式に従って計算されます。これは、正規分布とみなされます。

$$\frac{\sum (x_i - \bar{x})^2}{n}$$

- 標準と互換性
- **SQL99** 主要な SQL に含まれない、SQL/foundation の機能です。
  - **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。

参照

[「分析関数」\(106 ページ\)](#)  
[「VARIANCE 関数 \[集合\]」\(333 ページ\)](#)  
『システム管理ガイド 第2巻』の「[第2章 OLAP の使用](#)」

## VARIANCE 関数 [集合]

機能 数値セットの分散を返します。

構文 **VARIANCE** ([ ALL ] *expression*)

パラメータ **expression** 任意の数値式 (FLOAT、REAL、または DOUBLE)。

例 次のようなデータがあるとします。

```
SELECT Salary FROM Employees WHERE DepartmentID = 300
```

```

salary
51432.000
57090.000
42300.000
43700.00
36500.000
138948.000
31200.000
58930.00
75400.00

```

次の文は、値 1063923790.99999994 を返します。

```
SELECT VARIANCE (Salary) FROM Employees
WHERE DepartmentID = 300
```

次のようなデータがあるとします。

```
SELECT UnitPrice FROM Products WHERE name = 'Tee Shirt'
```

```

UnitPrice
9.00
14.00
14.00

```

次の文は、値 8.333333333333334327 を返します。

```
SELECT VARIANCE (UnitPrice) FROM Products
WHERE name = 'Tee Shirt'
```

#### 使用法

VARIANCE では、次の計算式が使用されます。

$$var = \frac{n \sum x^2 - (\sum x)^2}{n(n-1)}$$

VARIANCE は、結果を倍精度浮動小数点数のデータ型で返します。空のセットに適用すると、結果は NULL になり、1 要素の入力セットに NULL が返されます。

VARIANCE は、キーワード DISTINCT をサポートしません。VARIANCE で DISTINCT を使用すると、構文エラーが返されます。

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

#### 参照

[「STDDEV 関数 \[集合\]」 \(258 ページ\)](#)

[「VAR\\_SAMP 関数 \[集合\]」 \(332 ページ\)](#)

『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

## WEEKS 関数 [日付および時刻]

#### 機能

任意の開始日時から経過した週の数か、指定された 2 つの日時の間の週の数か、または *integer-expression* で指定された週の日時に追加します。

#### 構文

```
WEEKS (datetime-expression
 | datetime-expression, datetime-expression
 | datetime-expression, integer-expression)
```

#### パラメータ

**datetime-expression** 日時。

**integer-expression** *datetime-expression* に追加する週数。*integer-expression* が負の場合、指定された週数が日時の値から減算されます。時間、分、秒は無視されます。整数式を指定する場合、*datetime-expression* は DATETIME データ型として明示的にキャストする必要があります。

データ型のキャストについては、「[CAST 関数 \[データ型変換\]](#)」 (134 ページ) を参照してください。



例

次の文は、値 104278 を返します。

```
SELECT WEEKS ('1998-07-13 06:07:12') FROM iq_dummy
```

次の文を実行すると、9 つの日付の間の差である値 2 が返されます。

```
SELECT WEEKS ('1999-07-13 06:07:12',
 '1999-09-13 10:07:12') FROM iq_dummy
```

次の文を実行すると、タイムスタンプの値 1999-06-16 21:05:07.000 が返ります。

```
SELECT WEEKS (CAST ('1999-05-12 21:05:07'
 AS TIMESTAMP), 5) FROM iq_dummy
```

使用法

週は、北米で使用されているカレンダーに従い、日曜日に始まり土曜日に終わるものとして定義されています。1 つ目の構文で返される数値は、2 つの日付が同じ週かどうかを判断するのによく利用されます。

```
WEEKS (invoice_sent) = WEEKS (payment_received) FROM
iq_dummy
```

2 つ目の構文では、WEEKS の値は 2 つの日付の間にある日曜日の数で計算されます。時間、分、秒は無視されます。この関数の計算結果は DATE\_FIRST\_DAY\_OF\_WEEK オプションの影響は受けません。

標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase Adaptive Server Enterprise** ではサポートされていません。

## WEIGHTED\_AVG 関数 [集合]

機能

算術 (または線形) 加重平均を計算します。加重平均は、平均に使用される各数量に加重が付加される平均です。加重は、平均を構成する各数量の相対的な重要性を決定します。

構文

```
WEIGHTED_AVG (expression)
OVER (window-spec)
```

*column-spec* : 以下の「使用法」を参照してください。

パラメータ

**expression** 計算する加重値の数値式。

使用法

WEIGHTED\_AVG 関数を使用して、加重移動平均を作成します。加重移動平均では、加重は時間の経過に従って等差階級的に減少します。加重は、直近のデータ・ポイントの最高加重から 0 に減少します。

図 4-2 : WEIGHTED\_AVG の計算

$$WMA_M = \frac{np_M + (n - 1)p_{M-1} + \dots + 2p_{M-n+2} + p_{M-n+1}}{n + (n - 1) + \dots + 2 + 1}$$

加重を計算するために、2 つ異常の加重移動平均を一緒に平均するか、EXP\_WEIGHTED\_AVG 関数を使用することができます。

関数構文 (インライン) または SELECT 文の WINDOW 句のいずれかで *window-spec* の要素を指定できます。

*column-spec* :

- 1 つの ORDER BY 指定子を含む必要があります。
- FOLLOWING および RANGE 指定子を含むことはできません。
- ROW 指定子 (指定した場合) の 2 番目の引数は CURRENT ROW である必要があります。
- NULL 値を含むことはできません。
- DISTINCT 指定子を含むことはできません。
- UNBOUNDED PRECEDING はサポートされていますが、使用した場合のパフォーマンスは低い場合があります。

ウィンドウの指定方法については、「[分析関数](#)」(106 ページ) を参照してください。

例 次の例は、フロリダの部門ごとの従業員の給与の加重移動平均、および平均のほとんどの加重に関する現在雇用されている従業員の給与を返します。

```
SELECT DepartmentID, Surname, Salary,
 WEIGHTED_AVG(Salary) OVER (PARTITION BY DepartmentID
 ORDER BY YEAR(StartDate) DESC) as "W_AVG"
FROM Employees
WHERE State IN ('FL') ORDER BY DepartmentID
```

次の結果セットが返されます。

表 4-63 : WEIGHTED\_AVG の結果セット

| DepartmentID | Surname  | Salary    | W_AVG        |
|--------------|----------|-----------|--------------|
| 100          | Lull     | 87900.000 | 87900.000000 |
| 100          | Gowda    | 59840.000 | 69193.333333 |
| 200          | Sterling | 64900.000 | 64900.000000 |
| 200          | Kelly    | 87500.000 | 79966.666667 |
| 300          | Litton   | 58930.000 | 58930.000000 |

| DepartmentID | Surname  | Salary    | W_AVG        |
|--------------|----------|-----------|--------------|
| 400          | Evans    | 68940.000 | 68940.000000 |
| 400          | Charlton | 28300.000 | 41846.666667 |
| 400          | Francis  | 53870.000 | 47858.333333 |

標準と互換性

- **SQL2008** ベンダの拡張機能。

## WIDTH\_BUCKET 関数 [数値]

機能

与えられた式に対して、WIDTH\_BUCKET 関数は、この式の評価後の結果に割り当てられるバケット番号を返します。

構文

**WIDTH\_BUCKET** ( *expression*, *min\_value*, *max\_value*, *num\_buckets* )

パラメータ

**expression** ヒストグラムが作成されている式です。この式は、数値または日時の値、または暗黙で数値または日時の値に変換できる値に評価される必要があります。*expr* が *null* に評価されると、式は *null* を返します。

**min\_value** *expr* に使用できる範囲の各ポイントに解決される式。数値または日時値にも評価される必要があります、*null* には評価できません。

**max\_value** *expr* に使用できる範囲の各ポイントに解決される式。数値または日時値にも評価される必要があります、*null* には評価できません。

**num\_buckets** バケット数を示す定数に解決される式。この式は正の整数に評価される必要があります。

例

次の例では、サンプル・テーブル内のマサチューセッツ州の顧客の *credit\_limit* カラムに 10 のバケット・ヒストグラムを作成し、各顧客のバケット数 (“Credit Group”) を返します。最大値を超える限度額が設定されている顧客は、オーバフロー・バケット 11 に割り当てられます。

```
select EmployeeID, Surname, Salary,
 WIDTH_BUCKET(Salary, 29000, 60000, 4) "Wages" from
Employees where State = 'FL' order by "Wages"
```

| EMPLOYEEID | SURNAME  | SALARY    | Wages |
|------------|----------|-----------|-------|
| 888        | Charlton | 28300.000 | 0     |
| 1390       | Litton   | 58930.000 | 4     |
| 207        | Francis  | 53870.000 | 4     |
| 266        | Gowda    | 59840.000 | 4     |
| 445        | Lull     | 87900.000 | 5     |
| 1021       | Sterling | 64900.000 | 5     |
| 902        | Kelly    | 87500.000 | 5     |
| 1576       | Evans    | 68940.000 | 5     |

範囲が逆の場合、バケットはオープン・クローズ間隔になります。次に例を示します。WIDTH\_BUCKET (*credit\_limit*, 5000, 0, 5)。この例では、バケット番号 1 は (4000, 5000]、バケット番号 2 は (3000, 4000]、およびバケット番号 5 は (0, 1000] です。オーバフロー・バケットには 0 (5000, +infinity) の番号が付き、アンダフロー・バケットには 6 (-infinity, 0] の番号が付きます。

#### 使用法

WIDTH\_BUCKET 関数を使用して等幅ヒストグラムを生成できます。等幅ヒストグラムでは、データ・セットを間隔サイズ(最も高い値から最も低い値まで)の同じバケットに分割します。保持されるロー数はバケットごとに異なります。関連する関数の NTILE は、等高バケットを作成します。

等幅ヒストグラムは数値、日付、日時データ型でのみ生成されるため、最初の 3 つのパラメータはすべて数値式またはすべて日付式にする必要があります。他の型の式は使用できません。最初のパラメータが NULL の場合、結果は NULL です。2 番目および 3 番目のパラメータが NULL の場合、エラー・メッセージが返されます。これは、NULL 値は日付または数値次元の範囲のどの終了ポイント(またはあらゆるポイント)も示すことができないためです。最後のパラメータ(バケットの数)は、正の整数値に評価される数値式にする必要があります。0、NULL、または負の値にすると、エラーが発生します。

バケットには 0 から (n+1) まで番号が付けられます。バケット 0 は、最小値未満の値のカウントを保持します。バケット (n+1) は、指定された最大値以上の値のカウントを保持します。

#### 標準と互換性

- **SQL03** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

#### 参照

「[NTILE 関数 \[統計\]](#) (212 ページ) (等高ヒストグラムの作成)。  
『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」

## YEAR 関数 [日付および時刻]

#### 機能

指定された日付/時刻の年に対応する 4 桁の数を返します。

#### 構文

**YEAR** ( *datetime-expression* )

#### パラメータ

**datetime-expression** 日時。

#### 例

次の文は、値 1998 を返します。

```
SELECT YEAR('1998-07-13 06:07:12') FROM iq_dummy
```

#### 使用法

YEAR 関数は、YEARS 関数の最初の構文と同じです。

- 標準と互換性
- **SQL92** ベンダの拡張機能。
  - **Sybase** Adaptive Server Enterprise ではサポートされていません。
- 参照 [「YEARS 関数 \[日付および時刻\]」 \(339 ページ\)](#)

## YEARS 関数 [日付および時刻]

**機能** 指定された日時の年に対応する 4 桁の数を返すか、指定された 2 つの日時の間の年数を返すか、または *integer-expression* で指定された年数を日時に追加します。

**構文**

```
YEARS (datetime-expression
 | datetime-expression, datetime-expression
 | datetime-expression, integer-expression)
```

**パラメータ** *datetime-expression* 日時。

*integer-expression* *datetime-expression* に追加する年数。*integer-expression* が負の場合は、指定された年数が日時の値から減算されます。整数式を指定する場合、*datetime-expression* は DATETIME データ型として明示的にキャストする必要があります。

データ型のキャストについては、[「CAST 関数 \[データ型変換\]」 \(134 ページ\)](#) を参照してください。

**例** 次の文は、値 1998 を返します。

```
SELECT YEARS ('1998-07-13 06:07:12') FROM iq_dummy
```

次の文を実行すると、2 つの日付の間の差である値 2 が返されます。

```
SELECT YEARS ('1997-07-13 06:07:12',
 '1999-09-13 10:07:12') FROM iq_dummy
```

次の文を実行すると、YEARS(cast('1999-05-12 21:05:07' as timestamp), 5) 値 2004-05-12 21:05:07.000 を返します。

```
SELECT YEARS (CAST ('1999-05-12 21:05:07'
 AS TIMESTAMP), 5) FROM iq_dummy
```

**使用法** YEARS 関数の 1 つ目の構文は、YEAR 関数と同じです。

2 つ目の構文は、最初の日付から次の日付までの年数を返します。これは、2 つの日付の間にある、年の初日の数で計算されます。負の値が返ることもあります。時間、分、秒は無視されます。たとえば、次の文は指定された日付の間にある、年の初日の数である 2 を返します。

```
SELECT YEARS ('2000-02-24', '2002-02-24') FROM
iq_dummy
```

次の文で指定されている日付の差は、暦年で完全な2年間に達していませんが、この文を実行しても2が返されます。値2は、2つの日付の間の、年の初日の数(この例では2001年1月1日と2002年1月1日)です。

```
SELECT YEARS ('2000-02-24', '2002-02-20') FROM
iq_dummy
```

3つ目の構文は、指定された日付に *integer-expression* の年数を追加します。処理後の日付が月の末日を過ぎてしまった場合 (`SSELECT YEARS ( CAST ( '1992-02-29' AS TIMESTAMP ), 1 )` など)、結果がその月の末日にセットされます。*integer-expression* が負の場合、指定された年数が日付から減算されます。時間、分、秒は無視されます。

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise ではサポートされていません。

#### 参照

[「YEAR 関数 \[日付および時刻\]」 \(338 ページ\)](#)

## YMD 関数 [日付および時刻]

#### 機能

指定された年、月、日に対応する日付を返します。

#### 構文

```
YMD (integer-expression1, integer-expression2, integer-expression3)
```

#### パラメータ

***integer-expression1*** 年。

***integer-expression2*** 月の番号。1 ~ 12 の範囲を超えて月を指定すると、それに応じて年が調整されます。

***integer-expression3*** 日の番号。任意の整数値を日として指定できます。指定の値に応じて、日付が調整されます。

#### 例

次の文を実行すると、値 1998-06-12 が返ります。

```
SELECT YMD (1998, 06, 12) FROM iq_dummy
```

通常の範囲を超えた値が指定されると、日付がその値に応じて調整されます。たとえば、次の文は値 1993-03-01 を返します。

```
SELECT YMD (1992, 15, 1) FROM iq_dummy
```

次の文を実行すると、値 1993-02-28 が返ります。

```
SELECT YMD (1992, 15, 1-1) FROM iq_dummy
```

次の文を実行すると、値 1992-02-29 が返ります。

```
SELECT YMD (1992, 3, 1-1) FROM iq_dummy
```

#### 標準と互換性

- **SQL92** ベンダの拡張機能。
- **Sybase** Adaptive Server Enterprise 互換。

## 他の SQL 言語との違い

### この章について

Sybase IQ は、基本的には ANSI SQL89 規格に準拠しますが、IBM の DB2 と SAA 仕様、および ANSI SQL92 規格で定義された機能が追加されています。

この章では、他の SQL ソフトウェアにはない、Sybase IQ の機能について説明します。

### 内容

| トピック名                         | ページ |
|-------------------------------|-----|
| <a href="#">Sybase IQ の機能</a> | 342 |

## Sybase IQ の機能

Sybase IQ の以下の機能は、他の多くの SQL ソフトウェアには見られません。

### 日付

Sybase IQ には日付、時刻、タイムスタンプのデータ型があり、年、月、日、時、分、秒、小数点以下の秒が含まれます。日付フィールドへの挿入または更新、および日付フィールド間の比較については、フリー・フォーマットの日付がサポートされています。

また、日付に関しては以下の演算が可能です。

- **date + integer** 指定された値の日数を日付に加えます。
- **date - integer** 指定された値の日数を日付から引きます。
- **date - date** 2つの日付間の日数を計算します。
- **date + time** 日付と時刻からタイムスタンプを作成します。

日付と時刻の処理に使用できる関数は、数多くあります。これらの詳細については、[第4章 SQL 関数](#)を参照してください。

### 整合性

Sybase IQ では、エンティティ整合性と参照整合性の両方がサポートされています。これは、**CREATE TABLE** コマンドおよび **ALTER TABLE** コマンドに対する次の2つの拡張機能によって実装されています。

```
PRIMARY KEY (column-name, ...)
[NOT NULL] FOREIGN KEY [role-name]
 [(column-name, ...)]
 REFERENCES table-name [(column-name, ...)]
 [CHECK ON COMMIT]
```

**PRIMARY KEY** 句では、関係のプライマリ・キーを宣言します。Adaptive Server IQ は、これにより、プライマリ・キーの一意性を確保し、プライマリ・キーのカラムに NULL 値が含まれないようにします。

**FOREIGN KEY** 句では、このテーブルと別のテーブルの間関係を定義します。この関係は、別のテーブルのプライマリ・キーの値を含むこのテーブルのカラム (1 つまたは複数) によって表されます。システムは、この定義に基づいて、これらのカラムの参照整合性を確認します。これらのカラムが変更されるか、このテーブルにローが挿入されると、これらのカラムについて、1 つまたは複数が NULL であること、または値が他のテーブルのプライマリ・キーのローの対応するカラムに一致していることが、常にチェックされます。詳細については、「**CREATE TABLE**文」を参照してください。



**ジョイン** Sybase IQ では、テーブル間での自動ジョインがサポートされています。他のソフトウェアでもサポートされる **NATURAL** ジョイン演算子と **OUTER** ジョイン演算子に加え、Sybase IQ では外部キー関係に基づく **KEY** ジョイン演算子がサポートされています。これにより、ジョイン実行時における **WHERE** 句の複雑さを軽減できます。

**更新** Sybase IQ では、複数のテーブルを **UPDATE** コマンドによって参照できます。複数のテーブルで定義されているビューを更新することもできます。多くの SQL ソフトウェアでは、ジョイン・テーブルに対して更新を行えません。

**テーブルの変更** **ALTER TABLE** コマンドが拡張されました。エンティティ整合性や参照整合性の変更に加えて、次の種類の変更が可能です。

```
ADD column data-type
MODIFY column data-type
DELETE column
RENAME new-table-name
RENAME old-column TO new-column
```

**MODIFY** を使用して文字カラムの最大長を変更できます。また、あるデータ型から別のデータ型に変換できます。『リファレンス：文とオプション』の「**SQL 文**」に記載されている「**ALTER TABLE 文**」を参照してください。

**サブクエリを使用できない場合**

SQL Anywhere とは異なり、Sybase IQ の場合、式を使用できる場所で、いつもサブクエリを使用できるわけではありません。Sybase IQ では、SQL-1989 の文法で許可されているケースと、最上位レベルのクエリ・ブロックの **SELECT** リスト、または **UPDATE** 文の **SET** 句内でのみサブクエリを使用できます。Sybase IQ では SQL Anywhere 拡張機能を使用できません。

多くの SQL ソフトウェアでは、比較演算子の右側でのみサブクエリを使用できます。たとえば、次のコマンドは Sybase IQ では有効ですが、他のほとんどの SQL ソフトウェアでは有効ではありません。

```
SELECT SurName,
 BirthDate,
 (SELECT DepartmentName
 FROM Departments
 WHERE DepartmentID = Employees.EmployeeID
 AND DepartmentID = 200)
FROM Employees
```

### その他の関数

Sybase IQ は、ANSI SQL 定義にはない関数をいくつかサポートしています。利用可能な全関数のリストについては、[第 4 章 SQL 関数](#)を参照してください。

### カーソル

Embedded SQL を使用する場合は、FETCH 文でカーソル位置を任意に移動できます。カーソルは、現在位置から相対的に、またはカーソルの最初または最後から指定のレコード数分、前後に移動できます。

## 物理的制限

### この章について

この章では、Sybase IQ データベースにおけるオブジェクトのサイズと数の制限について説明します。特定のプラットフォームにのみ適用される制限については、そのプラットフォームのマニュアルを参照してください。

### 内容

| トピック名                    | ページ |
|--------------------------|-----|
| <a href="#">サイズと数の制限</a> | 346 |

## サイズと数の制限

表 6-1 に、Sybase IQ データベースにおけるオブジェクトのサイズと数の制限を示します。ほとんどの場合、コンピュータのメモリおよびディスク容量から受ける制限の方が、大きな影響を持っています。

表 6-1 : Sybase IQ データベース・オブジェクトのサイズと数の制限

| 項目            | 制限                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| カタログ・ファイル・サイズ | FAT 32 ファイル・システムを使用する Windows システム (上限は 4GB) を除くすべてのプラットフォームで上限は 1TB。NTFS を使用する Windows システムの場合は、最大 1TB をサポートします。Sybase IQ は、NAS (ネットワーク接続ストレージ) デバイス上の DB 領域の作成をサポートしません。                                                                                                                                                                                                  |
| データベース名のサイズ   | 250 バイト。                                                                                                                                                                                                                                                                                                                                                                    |
| データベース・サイズ    | 最大データベース・サイズは、おおよそで、特定のプラットフォーム上のファイル数とファイル・サイズを乗算した値で、最大ディスク設定に依存する。<br>ファイルの最大数に影響するカーネル・パラメータについては、オペレーティング・システムのマニュアルを参照してください。                                                                                                                                                                                                                                         |
| DB 領域のサイズ     | ロー: 無制限 (デバイスの限界まで)。<br>オペレーティング・システム・ファイル: 4TB。                                                                                                                                                                                                                                                                                                                            |
| フィールド・サイズ     | BINARY では 255 バイト、VARBINARY では 32,767 バイト。<br>CHAR、VARCHAR では 32,767 バイト。<br>LONG BINARY、LONG VARCHAR では 128 KB ページの場合は最大 512 TB、512 KB ページの場合は 1 PB。                                                                                                                                                                                                                       |
| IQ ページ・サイズ    | 64KB から 512KB までの間。                                                                                                                                                                                                                                                                                                                                                         |
| キーの最大サイズ      | 単一カラムのインデックスでは 255 バイト。<br>複数カラムのインデックスでは 5,300 バイト。                                                                                                                                                                                                                                                                                                                        |
| SQL 文の最大長     | SQL 文の最大長は、IQ カタログに使用できるメモリ容量とカタログ・スタックのサイズに制限されています。<br>SQL 文が長すぎる場合は、 <code>-gss</code> を使用して、カタログ・スタックのサイズを大きくし、 <code>-c</code> または <code>-ch</code> と <code>-cl</code> の組み合わせを使用して、カタログ・メモリ・キャッシュの容量を増やしてください。<br>SQL 文をエラー・メッセージに出力する場合、テキストは IQ カタログ・ページのサイズに制限されます。<br><code>-gp</code> の設定を増やしてサーバを起動すると、長いコマンドを出力できますが、通常はデフォルトの <code>-gp 4096</code> を使用してください。 |

| 項目                                              | 制限                                                                                                                                                          |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 可変長 FILLER カラムの最大長<br>(同時接続中の) ユーザの最大数          | 512 バイト。<br>64 ビット・プラットフォーム (AIX、HP、Linux、Sun Solaris) では 1000。<br>Windows の 32 ビット・プラットフォームと 64 ビット・プラットフォームでは 200。                                        |
| 一時抽出ファイルの最大サイズ                                  | TEMP_EXTRACT_SIZE <sub>n</sub> オプションで設定。プラットフォーム別の制限は以下のとおり。<br>AIX、HP-UX : 0 ~ 64GB<br>Sun Solaris : 0 ~ 512GB<br>Windows : 0 ~ 128GB<br>Linux : 0 ~ 512GB |
| カラム数/テーブル                                       | Sybase IQ では 1 つのテーブルで 45,000 までのカラムをサポートする。テーブル内に 10,000 を超えるカラムがあると、パフォーマンスが低下する可能性があります。                                                                 |
| イベント数/データベース                                    | $2^{31} - 1 = 2\,147\,483\,647$                                                                                                                             |
| ファイル数/データベース                                    | ユーザが調整可能な (NOFILE を使うなど) OS による上限。一般的には、データベースごとに 2,047 ファイル。                                                                                               |
| インデックス数                                         | $2^{32}$ (~4,000,000)/テーブル。                                                                                                                                 |
| ロー数/テーブル                                        | テーブル・サイズにより制限される。上限は $2^{48}$ 。                                                                                                                             |
| データベースごとのストアド・プロシージャの数                          | $2^{32} - 1 = 4\,294\,967\,295$                                                                                                                             |
| 単独の FROM 句内のテーブルまたはビューの数                        | ジョイン・オプティマイザが有効になっている場合、クエリにより、16 から 64 の間。                                                                                                                 |
| クエリが参照するテーブルまたはビューの数                            | 512                                                                                                                                                         |
| テーブル数/データベース                                    | 4,293,918,719                                                                                                                                               |
| ジョイン・インデックスごとのテーブル数 (1 つのクエリ・ブロックでジョイン可能なテーブル数) | 32                                                                                                                                                          |
| 参照されるテーブル数/トランザクション                             | 制限なし。                                                                                                                                                       |
| クエリごとの UNION ブランチ数                              | 512。各ブランチの FROM 句に複数のテーブルがある場合、クエリごとのテーブル数の上限によって、UNION ブランチの許容数は少なくなる。                                                                                     |
| IN リスト内の値の数                                     | 250,000                                                                                                                                                     |
| ロー・サイズ                                          | Sybase ではページ・サイズの半分を上限として推奨。                                                                                                                                |
| テーブル・サイズ                                        | データベース・サイズにより制限。                                                                                                                                            |



## この章について

この章では、Sybase IQ データベースに提供されている、システム情報を取り出すためのストアド・プロシージャについて説明します。

## 内容

| トピック名                                                                    | ページ |
|--------------------------------------------------------------------------|-----|
| <a href="#">システム・プロシージャの概要</a>                                           | 350 |
| <a href="#">システム・ストアド・プロシージャ</a>                                         | 351 |
| <a href="#">カタログ・ストアド・プロシージャ</a>                                         | 491 |
| <a href="#">Adaptive Server Enterprise のシステム・プロシージャと<br/>カタログ・プロシージャ</a> | 521 |
| <a href="#">SQL Anywhere でサポートされているプロシージャ</a>                            | 525 |

## システム・プロシージャの概要

Sybase IQ には、次の種類のシステム・プロシージャが用意されています。

- ストアド・プロシージャとして実装されるシステム関数。
- システム情報を表形式で表示する、カタログ・ストアド・プロシージャ。
- 上記 2 種類のプロシージャをマルチプレックス・サーバの操作向けにしたマルチプレックス・ストアド・プロシージャ。『Sybase IQ マルチプレックスの使用』の「付録 A マルチプレックス・リファレンス」の「システム・プロシージャ」を参照してください。
- Transact-SQL システムおよびカタログ・プロシージャ。これらのシステム・プロシージャの一覧については、「[Adaptive Server Enterprise のシステム・プロシージャとカタログ・プロシージャ](#)」(521 ページ)を参照してください。

この章では、システム・プロシージャについて説明します。

sp\_iqsetcompression、sp\_iqshowcompression など、特にラージ・オブジェクト・データに関連付けられたシステム・ストアド・プロシージャについては、『Sybase IQ によるラージ・オブジェクト管理』の「[第 5 章 ストアド・プロシージャのサポート](#)」を参照してください。

## ストアド・プロシージャの構文規則

Interactive SQL と同様にストアド・プロシージャ名を直接入力する場合と、CALL 文で呼び出す場合とでは、ストアド・プロシージャ・コールでのカッコと引用符の使い方が異なります。この製品は Sybase IQ SQL と Transact-SQL のどちらの構文もサポートしているため、多少の変形は許容されています。Transact-SQL との互換性を維持する必要がある場合は、Transact-SQL の構文に従ってください。

構文の変形については、[表 7-1](#) を参照してください。

表 7-1 : ストアド・プロシージャの構文のバリエーション

| 構文                                    | 構文タイプ     | 説明                          |
|---------------------------------------|-----------|-----------------------------|
| <code>procedure_name ('param')</code> | Sybase IQ | パラメータをカッコで囲むときは引用符が必要です。    |
| <code>procedure_name 'param'</code>   | Sybase IQ | パラメータを引用符で囲む場合、カッコはオプションです。 |



| 構文                                         | 構文タイプ                      | 説明                                                                 |
|--------------------------------------------|----------------------------|--------------------------------------------------------------------|
| <i>procedure_name param</i>                | Transact-SQL               | パラメータ前後の引用符を省く場合は、カッコも省く必要があります。                                   |
| <i>procedure_name</i>                      | Sybase IQ または Transact-SQL | DBISQL の中でパラメータなしのプロシージャを直接実行する場合で、そのプロシージャにパラメータがない場合、この構文を使用します。 |
| <i>call procedure_name (param='value')</i> | Sybase IQ                  | パラメータ値を渡すプロシージャを呼び出すときはこの構文を使用します。                                 |

Transact-SQL ストアド・プロシージャを使用するときは、Transact-SQL 構文を使用してください。

## ストアド・プロシージャによりレポートされる統計情報について

多くのストアド・プロシージャは、プロシージャの実行時に、Sybase IQ の状態に関する情報をレポートします。つまり、スナップショットが得られます。たとえば、接続で使用されている領域を列挙するレポート・カラムは、プロシージャが実行された瞬間の領域だけを示し、接続で使用される最大領域を示すものではありません。

長期間にわたって Sybase IQ の利用状況をモニタするには、Sybase IQ モニタを利用します。このモニタは、開始から終了まで、統計情報を収集し、指定した間隔でレポートします。

## システム・ストアド・プロシージャ

システム・ストアド・プロシージャは、ユーザ ID `dbo` によって所有されています。この項のシステム・プロシージャは、IQ ストアドでのシステム管理者タスクを実行します。

**注意** デフォルトでは、DBISQLC が表示できるカラム値の最大長は 30 文字です。これでは、`sp_iqstatus` などのストアド・プロシージャの出力を表示するには不十分です。出力がトランケートされないようにするため、`dbisql` メニューから [コマンド]-[オプション] を選択し、[表示カラムの制限値]、[出力カラムの制限値]、またはその両方の値を増やすことで最大長を大きくしてください。

## sa\_dependent\_views プロシージャ

**機能** 指定のテーブルまたはビューのすべての依存ビューのリストを返します。

詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [sa\\_dependent\\_views system procedure \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference\\_en11/sa-dependent-views-sysproc.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sa-dependent-views-sysproc.html) を参照してください。

## sa\_verify\_password プロシージャ

**機能** 現在のユーザのパスワードを検証します。

詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [sa\\_verify\\_password system procedure \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference\\_en11/sysproc-s-4759676.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysproc-s-4759676.html) を参照してください。

## sa\_get\_user\_status システム・プロシージャ

**機能** ユーザの現在のステータスを検出できます。

詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [sa\\_get\\_user\\_status system procedure \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference\\_en11/sa-get-user-status.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sa-get-user-status.html) を参照してください。

## sp\_expireallpasswords プロシージャ

**機能** すべてのユーザ・パスワードをただちに期限切れにします。

**構文 1** `call sp_expireallpasswords`

**構文 2** `sp_expireallpasswords`

**参照** 「[sp\\_iqpassword プロシージャ](#)」 (439 ページ)

『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」の「CREATE USER 文」

**パーミッション** DBA 権限が必要です。

**例** すべてのユーザ・パスワードをただちに期限切れにします。

```
call sp_expireallpasswords
```

## sp\_iqaddlogin プロシージャ

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能      | 新しい Sybase IQ ユーザ・アカウントを指定のログイン・ポリシーに追加します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 構文 1    | <pre>call sp_iqaddlogin ('username_in', 'pwd',<br/>[ 'password_expiry_on_next_login' ][, 'policy_name' ])</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 構文 2    | <pre>sp_iqaddlogin 'username_in', 'pwd',<br/>[ 'password_expiry_on_next_login' ][, 'policy_name' ]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 構文 3    | <pre>sp_iqaddlogin username_in, pwd, [ password_expiry_on_next_login ][,<br/>policy_name ]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 使用法     | <p><b>username_in</b> ユーザのログイン名。ログイン名は識別子の規則に従う必要があります。</p> <p><b>pwd</b> ユーザのパスワード。パスワードは、パスワード規則に準拠する必要があります。つまり、有効な識別子である必要があります。</p> <p><b>password_expiry_on_next_login</b> (オプション) ユーザのログインが作成されたらすぐに、ユーザのパスワードを失効させるかどうかを指定します。デフォルトの設定は OFF (パスワードに有効期限はない) です。</p> <p><b>policy_name</b> (オプション) 指定のログイン・ポリシーの下にユーザを作成します。指定しないと、ルート・ログイン・ポリシーの下にユーザが作成されます。</p> <p>sp_iqaddlogin を使用して作成し、1 日で有効期限が切れるように設定した <b>username_in/pwd</b> は、翌日は終日有効であり、翌々日に無効になります。つまり、ログインを今日作成し、<i>n</i> 日で有効期限が切れるように設定した場合、日付が <i>n+1</i> 日目になると使用できなくなります。</p> |
| パーミッション | DBA 権限が必要です。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 参照      | <p>「sp_iqpassword プロシージャ」(439 ページ)</p> <p>『リファレンス：文とオプション』の「第 1 章 SQL 文」の「CREATE USER 文」</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 説明      | <p>新しい Sybase IQ ユーザ・アカウントを追加し、ログイン・ポリシーをユーザに割り当てて、ユーザを ISYSUSER システム・テーブルに追加します。ユーザがそのデータベースですでにユーザ ID を持っているのに、ISYSUSER 内に登録されていない場合 (たとえば、GRANT CONNECT statement 文または Sybase Central でユーザ ID が追加された場合など) は、sp_iqaddlogin によってユーザがテーブルに追加されます。</p> <p>Sybase IQ では、プロシージャを呼び出すときにログイン・ポリシー名を指定しないと、ユーザがルート・ログイン・ポリシーに割り当てられます。</p>                                                                                                                                                                                                                    |

**注意** ログイン・ポリシーに対する最大ログイン数が無制限の場合、そのログイン・ポリシーに属するユーザが持つことができる接続は無制限になります。

最初のユーザ・ログインでは、パスワードの変更が強制され、ログイン・ポリシーが新しく作成されたユーザに割り当てられます。CREATE USER を使用して新しいユーザを作成することをおすすめしますが、下位互換性のために sp\_iqaddlogin も引き続きサポートされています。

例 これら呼び出すと、パスワード irk324 を持つユーザ rose が、expired\_password という名前のログイン・ポリシーの下に追加されます。この例では、expired\_password ログイン・ポリシーはすでに存在するものとします。

```
call sp_iqaddlogin('rose', 'irk324', 'ON',
'expired_password')

sp_iqaddlogin 'rose','irk324', 'ON', 'expired_password'
```

## sp\_iqbackupdetails プロシージャ

**機能** 特定のバックアップに含まれるすべての dbfiles を表示します。

**構文** `sp_iqbackupdetails backup_id`

**パラメータ** `backup_id` バックアップ操作のトランザクション識別子を指定します。

**注意** SYSIQBACKUPHISTORY テーブルから `backup_id` 値を取得できません。次のクエリを実行します。

```
select * from sysiqbackuphistory
```

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**説明** sp\_iqbackupdetails は次の値を返します。

**表 7-2 : sp\_iqbackupdetails のカラム**

| カラム名        | 説明                                                                            |
|-------------|-------------------------------------------------------------------------------|
| backup_id   | バックアップ・トランザクションの識別子。                                                          |
| backup_time | バックアップの時間。                                                                    |
| backup_type | バックアップの種類：“Full”、“Incremental since incremental”、または“Incremental since full”。 |

| カラム名                | 説明                                                                                                                       |
|---------------------|--------------------------------------------------------------------------------------------------------------------------|
| selective_type      | バックアップのサブタイプ: “All inclusive”、“All RW files in RW dbspaces”、“Set of RO dbspace/file”。                                    |
| depends_on_id       | バックアップが依存する以前のバックアップの識別子。                                                                                                |
| dbspace_id          | バックアップされる DB 領域の識別子。                                                                                                     |
| dbspace_name        | SYSIQBACKUPHISTORYDETAIL からの DB 領域の名前。DB 領域名が、指定の dbspace_id の SYSDBSPACE の DB 領域名と一致する場合。それ以外は “null”。                  |
| dbspace_rwstatus    | “ReadWrite” または “Read Only”。                                                                                             |
| dbspace_createid    | DB 領域作成トランザクション識別子。                                                                                                      |
| dbspace_alterid     | Alter DBSPACE 読み込み/書き込みモード・トランザクション識別子。                                                                                  |
| dbspace_online      | ステータス。値は “Online” または “Offline”。                                                                                         |
| dbspace_size        | バックアップ時の DB 領域のサイズ (キロバイト)。                                                                                              |
| dbspace_backup_size | DB 領域でのバックアップ時のデータのサイズ (キロバイト)。                                                                                          |
| dbfile_id           | バックアップされる dbfile の識別子。                                                                                                   |
| dbfile_name         | バックアップ操作後に名前が変更されなかった場合は、論理ファイル名。変更された場合は “null”。                                                                        |
| dbfile_rwstatus     | “ReadWrite” または “Read Only”。                                                                                             |
| dbfile_createid     | dbfile 作成トランザクション識別子。                                                                                                    |
| dbfile_alterid      | Alter DBSPACE alter FILE 読み込み/書き込みモード・トランザクション識別子。                                                                       |
| dbfile_size in MB   | dbfile のサイズ (キロバイト)。                                                                                                     |
| dbfile_backup_size  | dbfile バックアップのサイズ (キロバイト)。                                                                                               |
| dbfile_path         | 指定された dbspace_id および dbfile_id の SYSDBFILE の物理ファイル・パス (“file_name”) と一致する場合は、SYSBACKUPDETAIL からの dbfile パス。それ以外は “null”。 |

例 `sp_iqbackupdetails` からの出力例を次に示します。

```

backup_id backup_time backup_type selective_type depends_on_id
 883 2008-09-23 13:58:49.0 Full All inclusive 0

dbspace_id dbspace_name dbspace_rwstatus dbspace_createid
 0 system ReadWrite 0

dbspace_alterid dbspace_online dbspace_size dbspace_backup_size dbfile_id
 0 0 2884 2884 0

dbfile_name dbfile_rwstatus dbfile_createid dbfile_alterid dbfile_size
 system ReadWrite 0 0 2884

dbfile_backup_size dbfile_path
 2884 C:\¥¥Documents and Settings¥¥All Users¥¥SybaseIQ¥¥demo¥¥iqdemo.db

```

参照

「SYSIQBACKUPHISTORY システム・ビュー」(548 ページ)

## sp\_iqbackupsummary プロシージャ

**機能** 実行されたバックアップ操作の概要を示します。

**構文** `sp_iqbackupsummary [ timestamp or backup_id ]`

**パラメータ** **timestamp または backup\_id** バックアップ操作をレポートする間隔を指定します。タイムスタンプまたはバックアップ ID を指定した場合、指定した時間に等しいか、それより大きい backup\_time を持つレコードのみが返されます。タイムスタンプを指定しない場合、ISYSIQBACKUPHISTORY のすべてのバックアップ・レコードが返されます。

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**説明** sp\_iqbackupsummary は次の値を返します。

**表 7-3 : sp\_iqbackupsummary のカラム**

| カラム名           | 説明                                                                                  |
|----------------|-------------------------------------------------------------------------------------|
| backup_id      | バックアップ・トランザクションの識別子                                                                 |
| backup_time    | バックアップの時間                                                                           |
| backup_type    | バックアップの種類：“Full”、“Incremental since incremental”、または “Incremental since full”       |
| selective_type | バックアップのサブタイプ：“All Inclusive”、“All RW files in RW dbspaces”、“Set of RO dbspace/file” |
| virtual_type   | 仮想バックアップの種類：“Non-virtual”、“Decoupled”、または “Encapsulated”                            |
| depends_on_id  | バックアップが依存するバックアップの識別子                                                               |
| creator        | バックアップの作成者                                                                          |
| backup_size    | バックアップのサイズ (キロバイト)                                                                  |
| user_comment   | ユーザ・コメント                                                                            |
| backup_command | 発行された backup 文 (コメントなし)                                                             |

**例** sp\_iqbackupsummary からの出力例を次に示します。

```

backup_id backup_time backup_type selective_type virtual_type
 883 2008-09-23 13:58:49.0 Full All inclusive Non virtual

depends_on_id creator backup_size user_comment backup_command
 0 DBA 10864 backup database to
 'c:¥¥¥¥temp¥¥¥¥b1'
```

## sp\_iqcheckdb プロシージャ

### 機能

現在のデータベースの妥当性を確認します。オプションで、DB 領域またはデータベースの割り付けの問題を解決します。

sp\_iqcheckdb はデータベース内のすべての記憶領域を読み込みます。成功すると、データベースのフリー・リスト (内部割り付けマップ) が更新され、データベースの実際の記憶領域割り当てがリストに反映されます。sp\_iqcheckdb は次に、実行した作業をリストしたレポートを生成します。

エラーが検出されると、sp\_iqcheckdb がエラーのタイプとオブジェクト名をレポートします。エラーが検出された場合、sp\_iqcheckdb はフリー・リストを更新しません。

sp\_iqcheckdb でも、指定されたテーブル、インデックス、インデックス・タイプ、またはデータベース全体の一貫性を検査できます。

---

**注意** sp\_iqcheckdb は、IQ データベース一貫性チェッカ (DBCC) のユーザ・インタフェースで、DBCC と呼ばれる場合もあります。

---

### 構文

```
sp_iqcheckdb 'mode target […] [resources resource-percent]'
```

これは sp\_iqcheckdb の一般的な構文です。データベースの一貫性をチェックするモードは 3 種類あり、アロケーション・マップをリセットするモードは 1 つです。各モードの構文を、それぞれ次に示します。パラメータ文字列でモードとターゲットの両方を指定しないと、Sybase IQ により次のエラー・メッセージが返されます。

少なくとも 1 つのモードおよびターゲットを DBCC に指定する必要があります。

### パラメータ

*mode:*

```
{ allocation | check | verify } | dropleaks
```

*target:*

```
[indextype index-type […]]
database | database resetclocks |
{ [indextype index-type] […] table table-name
[partition partition-name] […] |
index index-name | […]
dbspace dbspace-name }
```

allocation モード **sp\_iqcheckdb** 'allocation target [ resources resource-percent ]'

check モード **sp\_iqcheckdb** 'check target [ resources resource-percent ]'

verify モード **sp\_iqcheckdb** 'verify target [ resources resource-percent ]'

## Dropleaks モード

## 使用法

```
sp_iqcheckdb 'dropleaks target [resources resource-percent]'
```

**database** ターゲットがデータベースの場合、すべての DB 領域がオンラインである必要があります。

**index-type** 次のいずれかのインデックス・タイプ。FP、CMP、LF、HG、HNG、WD、DATE、TIME、DTTM。

指定した *index-type* がターゲットに存在しない場合、エラー・メッセージが返されます。複数のインデックス・タイプが指定されていて、ターゲットにこれらのインデックス・タイプの一部のみが含まれる場合、*sp\_iqcheckdb* では存在するインデックス・タイプが処理されます。

**index-name** 所有者とテーブル識別子を含めることができます。  
[[owner.]table-name.]index-name

*owner* を指定しない場合は、現在のユーザとデータベース所有者 (dbo) がこの順序で指定されます。*table* を指定しない場合、*index-name* はユニークである必要があります。

**table-name** 所有者の識別子を含めることができます：  
[owner.]table-name

*owner* を指定しない場合は、現在のユーザとデータベース所有者 (dbo) がこの順番で代わりに使用されます。*table-name* にテンポラリ・テーブルやプリジョイン・テーブルを指定することはできません。

---

**注意** テーブル名またはインデックス名にスペースが含まれる場合は、*table-name* または *index-name* パラメータを次のように二重引用符で囲みます。

```
sp_iqcheckdb 'check index "dbo.sstab.i2" resources 75'
```

---

**partition-name** *partition-name* パラメータには、修飾子は含まれません。スペースが含まれる場合は、二重引用符で囲みます。

**dbspace-name** *dbspace-name* パラメータには、修飾子は含まれません。スペースが含まれる場合は、二重引用符で囲みます。

パーティション・フィルタを使用すると、*sp\_iqcheckdb* は、そのパーティションに属する対応テーブルのローのサブセットを調べます。テーブルのパーティション・フィルタと、パーティション・フィルタのないテーブル・ターゲットは、テーブルに 1 つのパーティションのみがある場合、意味的に同じこととなります。



DB 領域ターゲットは、DB 領域に属するデータベースのページのサブセットを調べます。DB 領域はオンラインである必要があります。DB 領域とデータベース・ターゲットは、テーブルに1つのDB 領域のみがある場合、意味的に同じこととなります。

**resource-percent** 入力パラメータ *resource-percent* は0以上の整数である必要があります。リソースのパーセンテージを利用して、CPUの数に応じてスレッドの数を制御すれば、データベース一貫性チェックのCPUの使用率を制限できます。*resource-percent* = 100 (デフォルト値)の場合、1つのCPUに1つのスレッドが作成されます。*resource-percent* が100よりも大きい場合、CPUの数以上のスレッドがあり、一部のマシン構成ではパフォーマンスが向上することがあります。スレッドの最小数は1です。

**注意** `sp_iqcheckdb` のパラメータ文字列は、一重引用符で囲みます。また、文字列の長さが255バイトを超えることはできません。

割り付けの問題は `dropleaks` モードで修正できます。

#### パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

#### 説明

`sp_iqcheckdb` は、データベース内の各ブロックの割り付けをチェックし、次の `sp_iqdbstatistics` プロシージャが実行されるまで現在のセッション内の情報を保存します。`sp_iqdbstatistics` は、最近実行された `sp_iqcheckdb` の最新の結果を表示します。

`sp_iqcheckdb` は、指定されたパラメータに応じて、いくつかの関数を実行できます。データベースの一貫性の検査と修復には、次のモードがあります。

**allocation モード** データベース全体、特定のインデックス、特定のインデックス・タイプ、特定のパーティション、特定のテーブル、または特定のDB 領域のブロックマップ情報で割り付けをチェックします。インデックスの一貫性は検査しません。

重複ブロック (複数のオブジェクトが所有権を主張するブロック) または余分なブロック (オブジェクトが所有する割り付けられていないブロック) を検出します。

データベースまたはDB 領域ターゲットのリーク・ブロック (指定されたターゲットのオブジェクトによって要求されていない割り付けブロック) を検出します。

ターゲットが分割されたテーブルの場合、**allocation mode** では次のことが行われます。

- すべてのテーブルのパーティション割り付けビットマップのメタデータのチェック
- テーブル割り付けビットマップのメタデータのチェック
- ブロックマップ・エントリがテーブルの割り付けビットマップと一致していることの確認
- テーブルのパーティション割り付けビットマップが重複していないことの確認
- テーブルのパーティション割り付けビットマップで定義されたローが、テーブルの存在ビットマップのスーパーセットを構成していることのチェック
- テーブルのパーティション割り付けビットマップで定義されたローが、テーブルの割り付けビットマップのスーパーセットを構成していることのチェック

---

**注意** `sp_iqcheckdb` の入力パラメータ文字列で、単一のインデックス、インデックス・タイプ、またはテーブルの名前を指定すると、すべての割り付けの問題をチェックできません。

---

次の場合に **allocation** モードで実行します。

- 重複ブロックや未所有ブロックを検出する (データベースまたは特定のテーブルやインデックスをターゲットとして使用する)
- ページ・ヘッダ・エラーが発生した場合

DBCC オプション **resetclocks** は、**allocation** モードでのみ使用されます。**resetclocks** は、強制リカバリとともに使用し、マルチプレックス・セカンダリ・サーバをコーディネータに変換します。マルチプレックス機能については、『Sybase IQ マルチプレックスの使用』を参照してください。**resetclocks** は、内部データベース・バージョン管理クロックが遅れている場合に、クロックの値を修正します。Sybase IQ の保守契約を結んでいるサポート・センタに連絡した場合を除き、**resetclocks** オプションは使用しないでください。

**resetclocks** オプションは、シングル・ユーザ・モードで実行する必要があります。DBCC 文 **allocation database** でのみ使用できます。**resetclocks** コマンドの構文は次のとおりです。

```
sp_iqcheckdb 'allocation database resetclocks'
```

**check モード** すべてのデータベース・ページが、データベース全体、特定のインデックス、特定のインデックス・タイプ、特定のテーブル、特定のパーティション、または特定の DB 領域について読み込めることを確認します。テーブルが分割されている場合、check モードはテーブルのパーティション割り付けビットマップをチェックします。

クエリを実行したときに、メタデータ、NULL カウント、または重複した値を除くカウントのエラーが返された場合、check モードで実行します。

**verify モード** データベース全体、特定のインデックス、特定のインデックス・タイプ、特定のテーブル、特定のパーティション、または特定の DB 領域について、非 FP インデックスの内容を、対応する FP インデックスで確認します。FP および対応する非 FP インデックスのすべてのデータ・ページが指定のターゲットに含まれている場合、verify モードで次の不整合が検出されます。

- 欠落キー — FP インデックスに存在するが、非 FP インデックスには存在しないキー。
- 余分なキー — 非 FP インデックスに存在するが、FP インデックスには存在しないキー。
- 欠落ロー — FP インデックスに存在するが、非 FP インデックスには存在しないロー。
- 余分なロー — 非 FP インデックスに存在するが、FP インデックスには存在しないロー。

指定のターゲットに FP ページのサブセットのみが含まれる場合、verify モードでは次の不整合のみが検出できます。

- 欠落キー
- 欠落ロー

ターゲットが分割されたテーブルの場合、verify モードでは、テーブルまたはテーブル・パーティション内の各ローが正確なパーティションに割り当てられていることも確認します。

クエリを実行したときに、メタデータ、NULL カウント、または重複した値を除くカウントのエラーが返された場合、verify モードで実行します。

---

**注意** `sp_iqcheckdb` は、参照整合性の検査や、参照整合性違反の修復は行いません。

---

**dropleaks モード** Sybase IQ サーバがシングルノード・モードで実行されている場合は、dropleak モードをデータベースまたは DB 領域ターゲットで使用して、データベース全体または指定の DB 領域ターゲットの割り付けマップをリセットできます。ターゲットが DB 領域の場合、dropleaks 操作は指定の DB 領域での読み込み/書き込み操作も防ぎます。データベースまたは DB 領域リスト内のすべての DB 領域はオンラインである必要があります。

マルチプレックスで dropleaks モードを実行する方法については、『Sybase IQ マルチプレックスの使用』を参照してください。

sp\_iqcheckdb プロシージャは、次の例のように使用します。

例 1 次の例では、sp\_iqcheckdb はデータベース全体の割り付けをチェックします。

```
sp_iqcheckdb 'allocation database'
```

例 2 2 番目の例では、sp\_iqcheckdb がインデックス i1、i2、および dbo.t1.i3 に対して詳細なチェックを行います。新しいモードを指定しない場合、次のコマンドで示すように、sp\_iqcheckdb は残りのターゲットにも同じモードを適用します。

```
sp_iqcheckdb 'verify index i1 index i2 index dbo.t1.i3'
```

例 3 すべてのモードは、組み合わせて、1 つのセッションで複数のチェックをデータベースに対して行うことができます。次の例では、sp\_iqcheckdb は、CPU の半分を使用して、テーブル t2 のパーティション p1 の簡単なチェック、インデックス i1 の詳細なチェック、データベース全体に対する割り付けチェックを行います。

```
sp_iqcheckdb 'check table t2 partition p1 verify index
i1
allocation database resources 50'
```

例 4 次の例では、データベース内の FP タイプのインデックスをすべてチェックします。

```
sp_iqcheckdb 'check indextype FP database'
```

例 5 次の例では、テーブル t1 内の FP および HG インデックスと、テーブル t2 の LF インデックスを検証します。

```
sp_iqcheckdb 'verify indextype FP indextype HG table t1
indextype LF table t2'
```

例 6 次の例は、sp\_iqcheckdb の出力にある 3 つの“LVC cells”メッセージの 1 つを示します。

```
sp_iqcheckdb 'check index
EFG2JKL.ASIQ_IDX_T208_C504_FP'
```

```

Index Statistics:
** Inconsistent Index:
abcd.EFG2JKL.ASIQ_IDX_T208_C504_FP ***** FP
Indexes Checked: 1
** Unowned LVC Cells: 212 *****

```

sp\_iqcheckdb LVC セル・メッセージには、次の項目が含まれています。

- 未所有の LVC セル
- 重複した LVC セル・ロー
- 未割り付けの LVC セル・ロー

これらのメッセージには、VARCHAR、VARBINARY、LONG BINARY (BLOB)、または LONG VARCHAR (CLOB) カラムとの矛盾が示されています。未所有の LVC セルは、小容量の使用不可ディスク領域を表し、無視しても問題ありません。重複した LVC セルと未割り付けの LVC セルは、ダメージを受けたカラムを削除しないと解決できない重大なエラーです。

ダメージを受けたカラムを削除するには、古いカラムのコピーから新しいカラムを作成した後で、元のカラムを削除し、新しいカラムの名前を古いカラムの名前に変更します。

---

**注意** LVC は、幅が 255 より大きい VARCHAR または VARBINARY カラムです。LONG BINARY (BLOB) および LONG VARCHAR (CLOB) も LVC を使用します。

---

## DBCC の パフォーマンス

DBCC の実行時間は、データベース検査全体のデータベースのサイズ、指定するテーブルやインデックスの数、マシンのサイズによって異なります。データベースの一部、つまり指定したテーブル、インデックス、またはインデックス・タイプだけをチェックする場合は、データベース全体を検査する場合より時間がかかりません。

sp\_iqcheckdb dropleaks モードの処理時間は、DB 領域ターゲットの数によって異なります。

4 つの sp\_iqcheckdb モードの動作と出力内容を表 7-4 にまとめます。

**表 7-4 : sp\_iqcheckdb の各モードの動作と出力**

| モード        | 検出されるエラー                   | 出力            | 処理速度   |
|------------|----------------------------|---------------|--------|
| allocation | 割り付けのエラー                   | 割り付けの統計情報のみ   | 4TB/時  |
| check      | 割り付けのエラー<br>大部分のインデックス・エラー | 表示可能なすべての統計情報 | 60GB/時 |

| モード       | 検出されるエラー                   | 出力            | 処理速度   |
|-----------|----------------------------|---------------|--------|
| verify    | 割り付けのエラー<br>すべてのインデックス・エラー | 表示可能なすべての統計情報 | 15GB/時 |
| dropleaks | 割り付けのエラー                   | 割り付けの統計情報のみ   | 4TB/時  |

出力

実行モードによって、`sp_iqcheckdb` 出力には、結果の要約、エラー、統計情報、および修復の統計が含まれます。1つのセッションで複数のモードを指定した場合、出力には最高で3つの結果セットが含まれます。エラーの統計情報は、エラーが検出された場合のみ、アスタリスク(\*\*\*\*\*)で表示されます。

`sp_iqcheckdb` の出力は、Sybase IQ メッセージ・ファイル `.iqmsg` にもコピーされます。`DBCC_LOG_PROGRESS` オプションが ON の場合、`sp_iqcheckdb` は進行メッセージを IQ メッセージ・ファイルに送信し、ユーザが DBCC の実行状況を把握できるようにします。

出力例

`sp_iqcheckdb 'allocation database'` を実行し、リークした領域が見つかった場合に表示される出力の例を次に示します。リークされた領域とは、データベースのフリー・リスト(内部割り付けマップ)に従って割り付けられたが、どのデータベース・オブジェクトの一部でもない DBCC が判断したブロックのことです。この例では、DBCC は 32 のリーク・ブロックをレポートしています。

```

 Stat Value Flags
=====|=====|=====
DBCC Allocation Mode Report
=====|=====|=====
** DBCC Status |Errors Detected |*****
=====|=====|=====

Allocation Summary
=====|=====|=====
 Blocks Total |8192 |
 Blocks in Current Version |4954 |
 Blocks in All Versions |4954 |
 Blocks in Use |4986 |
 % Blocks in Use |60 |
** Blocks Leaked |32 |*****
=====|=====|=====

Allocation Statistics
=====|=====|=====
 Marked Logical Blocks |8064 |

```

```

Marked Physical Blocks |4954 |
Marked Pages |504 |
Blocks in Freelist |126553 |
Imaginary Blocks |121567 |
Highest PBN in Use |5432 |
** 1st Unowned PBN |452 |*****
Total Free Blocks |3206 |
Usable Free Blocks |3125 |
% Free Space Fragmented |2 |
Max Blocks Per Page |16 |
1 Block Page Count |97 |
3 Block Page Count |153 |
4 Block Page Count |14 |
...
9 Block Hole Count |2 |
16 Block Hole Count |194 |
Database Objects Checked |1 |
B-Array Count |1 |
Blockmap Identity Count |1 |
=====|=====
====Connection Statistics | |

```

## sp\_iqcheckoptions プロシージャ

**機能** 接続されているユーザについては、sp\_iqcheckoptions は、現在の値とデータベースのデフォルト値、そして、デフォルト値から変更されたサーバ起動オプションを表示します。

**構文** `sp_iqcheckoptions`

**パーミッション** なし。DBA には、すべてのグループとユーザに対して永続的に設定されたすべてのオプションと、DBA に設定された temporary オプションが表示されます。DBA 以外のユーザには、そのユーザ自身の temporary オプションが表示されます。デフォルトでないサーバ起動オプションはすべてのユーザに表示されます。

**使用法** パラメータは不要です。デフォルト値から変更された各オプションに対して 1 つのローを返します。出力はオプション名、ユーザ名の順でソートされます。

## 説明

接続されているユーザについては、`sp_iqcheckoptions` ストアド・プロシージャが、現在の値とデータベースのデフォルト値、そして、デフォルト値から変更されたサーバ起動オプションを表示します。

`sp_iqcheckoptions` は、すべての Sybase IQ および SQL Anywhere データベース・オプションを考慮します。Sybase IQ は一部の SQL Anywhere オプションのデフォルトを変更し、変更されたこれらの値が新しいデフォルト値になります。新しい Sybase IQ のデフォルト値が再度変更されない限り、`sp_iqcheckoptions` はこのオプションを一覧に表示しません。

`sp_iqcheckoptions` を実行すると、DBA には、すべてのグループとユーザに永続的に設定されているすべてのオプションと、DBA に設定されているテンポラリー・オプションが表示されます。DBA 以外のユーザには、そのユーザ自身の temporary オプションが表示されます。デフォルトでないサーバ起動オプションはすべてのユーザに表示されます。

表 7-5 : `sp_iqcheckoptions` のカラム

| カラム名          | 説明                                                                                                       |
|---------------|----------------------------------------------------------------------------------------------------------|
| User_name     | オプションが設定されたユーザまたはグループの名前。データベース作成時に、public グループにすべてのオプションが設定されます。public 以外のグループまたはユーザに設定されたオプションが表示されます。 |
| Option_name   | オプション名。                                                                                                  |
| Current_value | オプションの現在の値。                                                                                              |
| Default_value | オプションのデフォルト値。                                                                                            |
| Option_type   | TEMPORARY オプションの場合は“Temporary”、その他の場合は“Permanent”。                                                       |

## 例

これらの例では、temporary オプション `APPEND_LOAD` が ON に設定されており、グループ `mygroup` では `MAX_WARNINGS` オプションが 9 に設定されています。ユーザ `joel` では、`MAX_WARNINGS` に一時的な値 55 が設定されています。

最初の例では、`sp_iqcheckoptions` が DBA によって実行されています。

| User_name | Option_name        | Current_value                                | Default_value   | Option_type |
|-----------|--------------------|----------------------------------------------|-----------------|-------------|
| DBA       | Ansi_update_constr | CURSORS                                      | Off             | Permanent   |
| PUBLIC    | Ansi_update_constr | Cursors                                      | Off             | Permanent   |
| DBA       | Append_Load        | ON                                           | OFF             | Temporary   |
| DBA       | Checkpoint_time    | 20                                           | 60              | Temporary   |
| DBA       | Connection_authent | Company=MyComp;<br>Application=DBTools;Signa |                 | Temporary   |
| DBA       | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| PUBLIC    | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| mygroup   | Max_Warnings       | 9                                            | 281474976710655 | Permanent   |
| DBA       | Thread_count       | 25                                           | 0               | Temporary   |



2 番目の例では、`sp_iqcheckoptions` がユーザ `joel` によって実行されています。

| User_name | Option_name        | Current_value                                | Default_value   | Option_type |
|-----------|--------------------|----------------------------------------------|-----------------|-------------|
| joel      | Ansi_update_constr | CURSORS                                      | Off             | Permanent   |
| PUBLIC    | Ansi_update_constr | Cursors                                      | Off             | Permanent   |
| joel      | Checkpoint_time    | 20                                           | 60              | Temporary   |
| joel      | Connection_authent | Company=MyComp;<br>Application=DBTools;Signa |                 | Temporary   |
| joel      | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| PUBLIC    | Login_procedure    | DBA.sp_iq_proce                              | sp_login_envir  | Permanent   |
| joel      | Max_Warnings       | 55                                           | 281474976710655 | Temporary   |
| joel      | Thread_count       | 25                                           | 0               | Temporary   |

## sp\_iqclient\_lookup プロシージャ

### 機能

特定のクライアント IP アドレス/ポートを起点とするネットワーク・アナライザと同様に、特定のデータ・ストリームを担当する Sybase IQ ユーザ・アカウントを特定するために、クライアント・アプリケーションによって使用されます。

### 構文

```
sp_iqclient_lookup ['IPaddress'], [Port], [UserID]
```

### パラメータ

**IPaddress** 起点となるクライアント・アプリケーションの IP アドレスを指定します。

**Port** 起点となるクライアント・アプリケーションのポート番号を指定します。

**UserID** Sybase IQ ユーザの ID を指定します。

### パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

### 説明

`sp_iqclient_lookup` プロシージャは、クライアントの IP アドレスとポート番号を受け取って、Number (接続 ID)、IPaddress、Port、および UserID が含まれた単一のローを返します。

```
1> sp_iqclient_lookup '158.76.235.71',3360
2> go
```

| Number | IPaddress     | Port | UserID  |
|--------|---------------|------|---------|
| -----  | -----         | ---- | -----   |
| 15     | 158.76.235.71 | 3360 | rdeniro |

オプションで、3番目の引数を渡して、UserIDのみを選択することもできます。引数が渡されなかった場合、`sp_iqclient_lookup` は、現在のすべてのログインを IP アドレスとポート番号とともに返します。例を示します。

```
sp_iqclient_lookup

Number IPAddress Port UserID
----- -
11 162.66.131.36 2082 mbrando
21 162.66.100.233 1863 apacino
22 162.66.100.206 8080 jcaan
23 162.66.100.119 6901 rduvall
24 162.66.100.125 7001 dkeaton
25 162.66.100.124 6347 jcazale
```

```
(6 rows affected)
(return status = 0)
```

クライアント・アプリケーションが内部接続に TCP/IP を使用していない場合、アドレスは 127.0.0.1 と表示されます。

---

**注意** これらの情報にアクセスできるのは、ログオンしているユーザのみです。履歴ログイン・データがこの目的でサーバ上に維持されることはありません。

---

#### 関連する動作

`sp_iqclient_lookup` ストアド・プロシージャは、インストールごとに異なるサーバのパフォーマンスに影響を与える可能性があります。ログイン名を検索するには、サーバ上で現在アクティブになっている全接続をスキャンする必要があります。そのため、接続数の多いサーバほど、パフォーマンスに与える影響も大きくなります。また、これらの情報は動的(場合によっては高度に動的)であるため、キャッシュできません。したがって、サーバ機能を使用する他のクライアント・アプリケーションと同様に、このストアド・プロシージャの使用を管理することと、サーバに対する影響をモニタすることは、ローカル・システムの管理者の責任となります。

#### 例

UserID `jcazale1` の IP アドレスを表示します。

```
sp_iqclient_lookup null, null, jcazale

Number IPAddress Port UserID
----- -
11 162.66.131.36 2082 jcazale
15 164.66.131.36 1078 jcazale
```

クライアント IP 162.66.131.36 の IP アドレスを表示します。

```
sp_iqclient_lookup '162.66.131.36'
```

| Number | IPAddress     | Port | UserID  |
|--------|---------------|------|---------|
| -----  | -----         | ---- | -----   |
| 11     | 162.66.131.36 | 2082 | jcazale |
| 12     | 162.66.131.36 | 1078 | jcaan   |

**注意** ユーザによって指定された引数が正しくない場合、結果は空になります。

## sp\_iqcolumn プロシージャ

機能

データベース内のカラムに関する情報を表示します。

構文 1

```
sp_iqcolumn ([table_name],[table_owner], [table_loc])
```

構文 2

```
sp_iqcolumn
[table_name='table_name'],[table_owner='tableowner'],[table_loc='table_loc']
```

使用法

**構文 1** *table\_name* を指定せずに *table\_owner* を指定する場合は、*table\_name* を NULL に置き換える必要があります。たとえば、`sp_iqcolumn NULL,DBA` のように指定します。

**構文 2** パラメータはどのような順番で指定しても構いません。*'table\_name'* と *'table\_owner'* を一重引用符で囲みます。

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

データベース内のカラムに関する情報を表示します。*table\_name* パラメータを指定すると、その名前のテーブル内のカラムだけが返されます。*table\_owner* パラメータを指定すると、そのユーザが所有するテーブルだけが返されます。*table\_name* および *table\_owner* パラメータを両方とも指定すると、ユニークなテーブル (存在する場合) からカラムが選択されます。*table\_loc* を指定すると、そのセグメント・タイプで定義されたテーブルだけが返されます。パラメータを指定しないと、データベース内のすべてのテーブルのすべてのカラムが返されます。`sp_iqcolumn` は、システム・テーブルのカラム情報は返しません。

表 7-6 : sp\_iqcolumn のカラム

| カラム名            | 説明                                                                                                                                                      |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| table_name      | テーブルの名前                                                                                                                                                 |
| table_owner     | テーブルの所有者                                                                                                                                                |
| column_name     | カラム名                                                                                                                                                    |
| domain_name     | データ型                                                                                                                                                    |
| width           | 精度と位取りを持つ数値データ型の精度、または位取りを持たない数値データ型の格納領域の幅、または文字データ型の幅                                                                                                 |
| scale           | 数値データ型の位取り                                                                                                                                              |
| nulls           | カラムが NULL を許容する場合は 'Y'、許容しない場合は 'N'                                                                                                                     |
| default         | カラムが identity/autoincrement カラムである場合は 'Identity/Autoincrement'。そうでない場合は null                                                                            |
| cardinality     | 重複する値を除いたインデックスによるカウント (判明している場合)                                                                                                                       |
| est_cardinality | 重複した値を除いた値の予測数。MINIMIZE_STORAGE オプションを ON にしてカラムを作成した場合は自動的に 255 に設定されます。または、CREATE TABLE に指定された IQ UNIQUE 制約内のユーザが指定した値です                              |
| location        | TEMP = IQ テンポラリ・ストア、MAIN = IQ メイン・ストア、SYSTEM = カタログ・ストア                                                                                                 |
| isPartitioned   | カラムが、分割されたテーブルに属しており、かつテーブル・パーティションの DB 領域と異なる DB 領域を持つ 1 つ以上のパーティションを持っている場合は 'Y'。カラムのテーブルが分割されていないか、またはカラムの各パーティションがテーブル・パーティションと同じ DB 領域に存在する場合は 'N' |
| remarks         | COMMENT 文で追加されたユーザ・コメント                                                                                                                                 |
| check           | 検査制約の式                                                                                                                                                  |

例 次の構文は両方とも、テーブル **Departments** のすべてのカラムを返します。

```
sp_iqcolumn Departments
call sp_iqcolumn (table_name='Departments')
```

| table_name  | table_owner | column_name    | domain_name | width | scale | nulls | default |
|-------------|-------------|----------------|-------------|-------|-------|-------|---------|
| Departments | GROUPO      | DepartmentID   | integer     | 4     | 0     | N     | (NULL)  |
| Departments | GROUPO      | DepartmentName | char        | 40    | 0     | N     | (NULL)  |
| Departments | GROUPO      | DepartmentHead | integer     | 4     | 0     | Y     | (NULL)  |

| cardinality | est_cardinality | location | isPartitioned | remarks | check  |
|-------------|-----------------|----------|---------------|---------|--------|
| 5           | 5               | Main     | N             | (NULL)  | (NULL) |
| 0           | 5               | Main     | N             | (NULL)  | (NULL) |
| 5           | 5               | Main     | N             | (NULL)  | (NULL) |

次の構文は、DBA が所有するすべてのテーブルのすべてのカラムを返します。

```
sp_iqcolumn table_owner='DBA'
```

## sp\_iqcolumnuse プロシージャ

**機能** 負荷によってアクセスされるカラムの使用状況の情報を詳細にレポートします。

**構文** `sp_iqcolumnuse`

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**説明** SYSTEM で作成されたテーブルのカラムはレポートされません。

**表 7-7 : sp\_iqcolumnuse のカラム**

| カラム名       | 説明           |
|------------|--------------|
| TableName  | テーブル名        |
| ColumnName | カラム名         |
| Owner      | カラム所有者のユーザ名  |
| UID**      | カラムのユニークな識別子 |
| LastDT     | 前回のアクセスの日時   |
| NRef       | クエリ参照の数      |

\*\*UID はシステムが割り当てた番号であり、カラムのインスタンスをユニークに識別します (インスタンスはオブジェクト作成時に定義されます)。

**例** `sp_iqcolumnuse` プロシージャからの出力例を示します。

| TableName | ColumnName     | Owner | UID | LastDT              | NRef |
|-----------|----------------|-------|-----|---------------------|------|
| orders    | o_orderdate    | DBA   | 151 | 20070917 22:41:22.. | 13   |
| orders    | o_shippriority | DBA   | 154 | 20070917 22:41:22.. | 13   |
| lineitem  | l_orderkey     | DBA   | 186 | 20070917 22:41:22.. | 13   |
| lineitem  | l_extendedp..  | DBA   | 191 | 20070917 22:41:22.. | 13   |
| lineitem  | l_discount     | DBA   | 192 | 20070917 22:41:22.. | 13   |
| lineitem  | l_shipdate     | DBA   | 196 | 20070917 22:41:22.. | 13   |

|       |            |     |                |                     |   |
|-------|------------|-----|----------------|---------------------|---|
| #tmp1 | expression | DBA | 10000000001218 | 20070917 22:57:36.. | 1 |
| #tmp1 | expression | DBA | 10000000001222 | 20070917 22:41:58.. | 1 |
| ...   |            |     |                |                     |   |

---

**注意** 前述の例での長い番号は、テンポラリ ID です。

---

**参照** 『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「負荷のモニタリング」

「sp\_iqdbspace プロシージャ」(387 ページ)、 「sp\_iqindexadvice プロシージャ」(418 ページ)、 「sp\_iqindexuse プロシージャ」(426 ページ)、 「sp\_iqtableuse プロシージャ」(475 ページ)、 「sp\_iqunusedcolumn プロシージャ」(479 ページ)、 「sp\_iqunusedindex プロシージャ」(481 ページ)、 「sp\_iqunusedtable プロシージャ」(482 ページ)、 および 「sp\_iqworkmon プロシージャ」(489 ページ)

『リファレンス：文とオプション』の「第2章 データベース・オプション」の「INDEX\_ADVISOR オプション」

## sp\_iqconnection プロシージャ

**機能** テンポラリ DB 領域を使用しているユーザ、バージョンを有効にしているユーザ、各接続が Sybase IQ 内で行っている作業、接続ステータス、データベース・バージョン・ステータスなど、接続およびバージョンに関する情報を表示します。

**構文** `sp_iqconnection [ connhandle ]`

**使用方法** 入力パラメータ *connhandle* は、Number 接続プロパティに等しい、接続の ID 番号です。connection\_property システム関数は、次のように接続 ID を返します。

```
SELECT connection_property ('Number')
```

有効な *connhandle* の入力パラメータで呼び出すと、sp\_iqconnection はその接続に対応する 1 つのローのみを返します。

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

## 説明

sp\_iqconnection は、アクティブな各接続に対して1つのローを返します。カラム ConnHandle、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr、および LastIdle は、接続プロパティ Number、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr、および LastIdle にそれぞれ対応しており、システム関数 sa\_conn\_info と同じ値を返します。追加のカラムは、Sybase IQ エンジンの Sybase IQ 側から接続データを返します。ローは、ConnCreateTime の順で並べられます。

カラム MPXServerName は、表 7-8 に示すように、マルチプレックス・ノード間通信 (INC) に関連する情報を格納します。

表 7-8 : MPXServerName のカラム値

| 実行場所のサーバ         | MPXServerName カラムの内容                                                                                                                                |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| シンプレックス・サーバ      | NULL (すべての接続がローカル/ユーザ接続)。                                                                                                                           |
| マルチプレックス・コーディネータ | <ul style="list-style-type: none"> <li>ローカル/ユーザ接続の場合は NULL。</li> <li>すべての INC 接続 (オンデマンドまたは専用のハートビート接続) について、セカンダリ・ノードのサーバ名 (接続元) の値を含む。</li> </ul> |
| マルチプレックス・セカンダリ   | <ul style="list-style-type: none"> <li>ローカル/ユーザ接続の場合は NULL。</li> <li>コーディネータのサーバ名 (接続元) の値を含む。</li> </ul>                                           |

マルチプレックス機能の詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。

Java アプリケーションでは、RemotePWD フィールドに、TDS クライアントから Sybase IQ 固有の接続プロパティを指定します。次の例は、IQ 固有の接続パラメータの指定方法を示しています。ここで、myconnection は IQ 接続名になります。

```
p.put ("RemotePWD", "", "CON=myconnection");
```

RemotePWD パラメータの使用方法の詳細については、『SQL Anywhere Server プログラミング・ガイド』の [Specifying a database on a server](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/dbname-jconnect-jdbc.html) ([http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming\\_en11/dbname-jconnect-jdbc.html](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbprogramming_en11/dbname-jconnect-jdbc.html)) を参照してください。

表 7-9 : sp\_iqconnection のカラム

| カラム名        | 説明                        |
|-------------|---------------------------|
| ConnHandle  | 接続の ID 番号。                |
| Name        | サーバ名。                     |
| Userid      | 接続のユーザ ID。                |
| LastReqTime | 指定された接続に対する直前の要求が開始された時刻。 |

| カラム名                | 説明                                                                                                                                                                                                                                                                                                                                |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReqType             | 最後の要求のタイプを示す文字列。                                                                                                                                                                                                                                                                                                                  |
| IQCmdType           | Sybase IQ 側で現在実行されているコマンド (ある場合)。コマンドの種類には、エンジンの実装レベルで定義されたコマンドが反映されます。これらのコマンドは、トランザクション・コマンド、IQ ストア内のデータを対象とした DDL および DML コマンド、内部 IQ カーソル・コマンドと、OPEN、CLOSE DB、BACKUP、RESTORE などの特殊な制御コマンドから構成されます。                                                                                                                             |
| LastIQCmdTime       | この接続の Sybase IQ エンジンの IQ 側で最後の IQ コマンドが開始または完了した時刻。                                                                                                                                                                                                                                                                               |
| IQCursors           | この接続の IQ ストアでオープンしているカーソルの数。                                                                                                                                                                                                                                                                                                      |
| LowestIQCursorState | IQ カーソルの状態 (あれば)。接続に複数のカーソルがある場合、すべてのカーソルの中で最小のカーソル状態、つまり完了までの時間が最も長いものが表示されます。カーソル状態は内部の Sybase IQ 実装の詳細を反映するもので、将来的に変更される可能性があります。このバージョンのカーソル状態は、NONE、INITIALIZED、PARSED、DESCRIBED、COSTED、PREPARED、EXECUTED、FETCHING、END_OF_DATA、CLOSED、COMPLETED となります。名前からもわかるように、カーソル状態は操作の最後に変更されます。たとえば、状態 PREPARED は、カーソルが実行中であることを示します。 |
| IQthreads           | 現在、接続に割り当てられている Sybase IQ スレッドの数。割り当て済みのスレッドでも、アイドルである可能性があります。このカラムから、どの接続がリソースを最も多く使用しているかを判断できます。                                                                                                                                                                                                                              |
| TxnID               | 接続の現在のトランザクションのトランザクション ID。これは、BeginTxn、CmtTxn、および PostCmtTxn メッセージによって <i>.iqmsg</i> ファイルに表示されるトランザクション ID、また、データベースが開かれたときにログ記録される Txn ID Seq と同じです。                                                                                                                                                                            |
| ConnCreateTime      | 接続が作成された時刻。                                                                                                                                                                                                                                                                                                                       |
| TempTableSpaceKB    | この接続が IQ テンポラリー・テーブルに格納されているデータに使用している IQ テンポラリー・ストアの領域 (キロバイト)。                                                                                                                                                                                                                                                                  |
| TempWorkSpaceKB     | この接続が、ソート、ハッシュ、テンポラリー・ビットマップなどの作業領域として使用している IQ テンポラリー・ストアの領域 (キロバイト)。ビットマップや、Sybase IQ テンポラリー・テーブルのインデックスの一部分であるその他のオブジェクトによって使用されている領域は、TempTableSpaceKB に反映されます。                                                                                                                                                               |
| IQConnID            | <i>.iqmsg</i> ファイル内のすべてのメッセージに表示される 10 桁の接続 ID。これは、サーバ・セッション内でユニークな、単純増加する整数です。                                                                                                                                                                                                                                                   |
| satoiq_count        | Sybase IQ エンジンの SQL Anywhere 側から IQ 側への超過の数を表示するのに使用される内部カウンタ。これは、接続のアクティビティを確認するのに役立つことがあります。結果セットはローのバッファに返され、satoiq_count や iqtosa_count をローごとに 1 回増分することはありません。                                                                                                                                                               |
| iqtosa_count        | Sybase IQ エンジンの IQ 側から SQL Anywhere 側への超過の数を表示するのに使用される内部カウンタ。このカラムは、接続のアクティビティを確認するのに役立つことがあります。                                                                                                                                                                                                                                 |



| カラム名          | 説明                                                                                |
|---------------|-----------------------------------------------------------------------------------|
| CommLink      | 接続用の通信リンク。これは、Sybase IQ がサポートするネットワーク・プロトコルであり、同一マシン接続用の「ローカル」なものです。              |
| NodeAddr      | クライアント/サーバ接続のクライアント用ノード。                                                          |
| LastIdle      | 要求間のチックの数。                                                                        |
| MPXServerName | INC 接続の場合、varchar(128) 値には、INC 接続が開始するマルチプレックス・サーバの名前が含まれます。INC 接続でない場合は NULL です。 |

例 `sp_iqconnection` 出力の例を次に示します。

| ConnHandle | Name               | Userid | LastReqTime               | ReqType |
|------------|--------------------|--------|---------------------------|---------|
| 9          | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 13:15:00.035' | 'EXEC'  |
| 11         | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 13:15:00.046' | 'EXEC'  |
| 13         | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 14:52:55.003' | 'EXEC'  |
| 15         | 'IQ_MPX_SERVER_H'  | 'dbo'  | '2008-11-18 14:53:25.005' | 'EXEC'  |
| 17         | 'SQL_DBC_49450e8'  | 'DBA'  | '2008-11-18 14:59:45.680' | 'OPEN'  |
| 44         | 'Sybase Central 1' | 'DBA'  | '2008-11-18 14:59:45.023' | 'CLOSE' |

| IQCmdType             | LastIQCmdTime         | IQCursors | LowestIQCursorState |
|-----------------------|-----------------------|-----------|---------------------|
| 'NONE'                | 2008-11-18 13:15:00.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 13:15:00.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 14:52:55.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 14:53:25.0 | 0         | 'NONE'              |
| 'IQUTILITYOPENCURSOR' | 2008-11-18 14:59:45.0 | 0         | 'NONE'              |
| 'NONE'                | 2008-11-18 14:43:33.0 | 0         | 'NONE'              |

| IQthreads | TxnID | ConnCreateTime        | TempTableSpaceKB | TempWorkSpaceKB |
|-----------|-------|-----------------------|------------------|-----------------|
| 1         | 0     | 2008-11-18 13:14:09.0 | 0                | 0               |
| 1         | 0     | 2008-11-18 13:14:34.0 | 0                | 0               |
| 1         | 0     | 2008-11-18 13:14:55.0 | 0                | 0               |
| 1         | 0     | 2008-11-18 13:15:25.0 | 0                | 0               |
| 1         | 50024 | 2008-11-18 13:28:08.0 | 0                | 0               |
| 1         | 50545 | 2008-11-18 14:03:50.0 | 0                | 0               |

| IQconnID | satoiq_count | iqtos_a_count | CommLink | NodeAddr       | LastIdle | MPXServerName |
|----------|--------------|---------------|----------|----------------|----------|---------------|
| 23198    | 28           | 12            | 'local'  | ''             | 2977     | 'mpx0631_r1'  |
| 23202    | 28           | 12            | 'local'  | ''             | 1503     | 'mpx0631_r2'  |
| 23207    | 127          | 12            | 'local'  | ''             | 10000    | 'mpx0631_w1'  |
| 23212    | 127          | 12            | 'local'  | ''             | 10000    | 'mpx0631_w2'  |
| 23267    | 658          | 66            | 'TCP/IP' | '10.18.60.181' | 9375     |               |
| 23443    | 510          | 54            | 'local'  | ''             | 1238     |               |

## sp\_iqconstraint プロシージャ

|         |                                                                                                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能      | CREATE TABLE または ALTER TABLE を使用して、指定されたテーブルまたはカラムに定義した参照整合性制約を一覧表示します。                                                                                          |
| 構文      | <b>sp_iqconstraint</b> [ <i>table-name</i> ', <i>column-name</i> ', <i>table-owner</i> ']                                                                        |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。                                                                                    |
| 説明      | テーブル名とカラム名を指定しなかった場合、現在接続中のデータベースにある、テンポラリ・テーブルも含むすべてのテーブルのすべての参照整合性制約が返されます。この情報には、ユニークまたはプライマリ・キー制約、参照制約を含み、CREATE TABLE や ALTER TABLE 文で定義されたロール名が割り当てられています。 |
| 例       | これは、すべてのテーブルのうち、候補キーまたは外部キーに、所有者が bob のカラム ck1 が含まれているすべてのプライマリ・キー/外部キーの組み合わせを表示した出力例です。                                                                         |

```
call sp_iqconstraint('', 'ck1', 'bob')

PTAB1 bob ASIQ_IDX_T27_HG unique ck1,ck2 selftab bob CK6FK3 Y
ASIQ_IDX_T42_HG ck1,ck2

PTAB2 bob ASIQ_IDX_T27_HG unique ck1,ck2 selftab bob CK6FK4 Y
ASIQ_IDX_T206_I42_HG ck1,ck2

selftab bob ASIQ_IDX_T26_HG unique ck1,ck2 selftab bob CK3FK1 Y
ASIQ_IDX_T206_I42_HG ck1,ck2
```

表示されるカラムは、次のとおりです。

- プライマリ実施テーブル
- テーブルの所有者
- 候補キー・インデックス
- プライマリ・キーまたはユニーク・キー
- プライマリ・キー・カラム
- 外部テーブル
- 外部テーブルの所有者
- 外部キー・ロール名
- 実施ステータス (実施されている場合は“Y”、実施されていない場合は“N”)

- 外部キー・インデックス
- 外部キー・カラム
- 場所 (“TEMP”、“MAIN”、または “SYSTEM”)

## sp\_iqcontext プロシージャ

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能      | 接続ごとに、現在実行されている文に関する情報を追跡して表示します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 構文      | <code>sp_iqcontext [ connhandle ]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 使用法     | 入力パラメータ <i>connhandle</i> は、Number 接続プロパティに等しい、接続の ID 番号です。<br><br>有効な <i>connhandle</i> の入力パラメータで呼び出されると、 <code>sp_iqcontext</code> はその接続に対応する 1 つの情報のみを返します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 説明      | <code>sp_iqcontext</code> では、DBA が、特定の時間にシステム上で実行中の文を確認したり、その文を発行したユーザおよび接続を調べたりすることができます。この情報を基に、このユーティリティでは次のような作業を行うことができます。 <ul style="list-style-type: none"><li>• 文のテキストを、<code>sp_iqconnection</code> 内の相当する行に対応させ、各接続でのリソースの利用状況やトランザクション情報を入手する。</li><li>• 文のテキストを、<code>-zr</code> サーバ・オプションが ALL または SQL に設定されている場合に作成される SQL ログ内の相当する行に対応させる。</li><li>• 接続情報を使用して、<code>sp_iqcontext</code> 内の文のテキストを、ファイル <i>.iqmsg</i> (Sybase IQ がクエリ・プランを収集できる場合はクエリ・プランが含まれる) 内の相当する行に対応させる。</li><li>• IQ スタック・トレース (<i>stktre-yyyymmdd-hhnnss_#.iq</i>) が作成されていれば、文のテキストをこれに対応させる。</li><li>• OS のスタック・トレース (Sun Solaris の <i>pstack</i> など) が生成されている場合、この情報をそれと照合する。</li></ul> 収集された文のテキストの最大サイズは、カタログ・ストアのページ・サイズです。 |

表 7-10 : sp\_iqcontext のカラム

| カラム名                | 説明                                                                                                                                                                                                                                   |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConnOrCursor        | CONNECTION または CURSOR。                                                                                                                                                                                                               |
| ConnHandle          | 接続の ID 番号。                                                                                                                                                                                                                           |
| Name                | サーバ名。                                                                                                                                                                                                                                |
| Userid              | 接続またはカーソルのユーザ ID。                                                                                                                                                                                                                    |
| numIQCursors        | カラム 1 が CONNECTION の場合、この接続で開かれているカーソルの数を表します。<br>カラム 1 が CURSOR の場合、この接続に関連付けられているカーソルに割り当てられた通し番号を表します。                                                                                                                            |
| IQthreads           | 現在、接続に割り当てられている IQ スレッドの数。割り当て済みのスレッドでも、アイドルである可能性があります。                                                                                                                                                                             |
| TxnID               | 現在のトランザクションのトランザクション ID。                                                                                                                                                                                                             |
| ConnOrCurCreateTime | この接続またはカーソルが作成された時刻。                                                                                                                                                                                                                 |
| IQConnID            | .iqmsg ファイル内のすべてのメッセージに表示される 10 桁の接続 ID。これは、サーバ・セッション内でユニークな、単純増加する整数です。                                                                                                                                                             |
| IQGovernPriority    | ユーザのクエリが実行キューに入れられる順番を示す値。1 は優先度高、2 (デフォルト) は優先度中、3 は優先度低を示します。値 -1 は、IQGovernPriority を操作に適用しないことを示します。IQGovernPriority 値は、データベース・オプション IQGOVERN_PRIORITY を使用して設定します。『パフォーマンス&チューニング・ガイド』の「第 3 章 クエリと削除の最適化」の「クエリの優先度の設定」を参照してください。 |
| CmdLine             | 実行されるユーザ・コマンドの最初の 4096 文字。                                                                                                                                                                                                           |

例 次の例は、sp\_iqcontext をパラメータなしで発行し、現在のすべての接続の結果を表示した場合の出力の要約です。

```

CONNECTION 701773517 dba7 DBA 6 1 1324 2009-06-04 09:24:17.000 4 NO COMMAND
CURSOR 701773517 dba7 DBA 1 0 1324 2009-06-04 09:24:46.000 4 2 select * from fool
CURSOR 701773517 dba7 DBA 2 0 1324 2009-06-04 09:24:47.000 4 2 select a from fool
...
CURSOR 701773517 dba7 DBA 6 0 1324 2009-06-04 09:24:47.000 4 2 select e from fool
CONNECTION 1271624950 dba7 DBA 0 12 1377 2009-06-04 09:24:12.000 3 2 sp_iqcheckdb
CONNECTION 1841476383 dba7 DBA 10 1 1337 2009-06-04 09:24:19.000 5 2 call sp_iqcontext()
CURSOR 1841476383 dba7 DBA 1 0 1337 2009-06-04 09:24:47.000 5 2 select * from foo
...
CURSOR 1841476383 dba7 DBA 10 0 1337 2009-06-04 09:24:48.000 5 2 select i from foo

```

出力の1行目には、connection 701773517 (IQ 接続 ID は 4) が表示されています。この接続のサーバは dba7、ユーザは DBA です。この接続は、アクティブなカーソルを 6 つと IQ スレッドを 1 つ持ち、トランザクション 1324 から作成されていることがわかります。また、sp\_iqcontext が発行されたときには、コマンドを実行していません。出力の次の 6 行では、この接続で使用中のカーソルが列挙されています (ここでは 3 つのみ表示しています)。

2 つの接続がストアド・プロシージャを実行しています。connection 1271624950 は、sp\_iqcheckdb を dbisql から直接実行しており、アクティブなカーソルはなく、12 の IQ スレッドを使用しています。connection 1841476383 は、sp\_iqcontext をプロシージャとして呼び出し、IQ スレッドは 1 つしか使用していません。10 のアクティブなカーソルを持ちます (ここでは、最初と最後のカーソルのみ表示しています)。いずれの場合でも、ストアド・プロシージャの名前は表示されていますが、その中で実行されているコードは表示されていません。

接続ハンドル (この例の最初の接続では 701773517) は、-zr ログ内の結果を識別します。IQ 接続 ID (この例の最初の接続では 4) は、.iqmsg ファイル内の結果を識別します。UNIX システムでは、grep コマンドを使用して、接続ハンドルまたは接続 ID のすべてのインスタンスを探ることができます。これにより、すべてのソースからの情報を簡単に関連付けることができます。ユーザ・コマンド・フラグメントの前の 2 は、これが優先度中のクエリであることを示します。

## sp\_iqcopyloginpolicy プロシージャ

|         |                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------|
| 機能      | 既存のログイン・ポリシーをコピーして、新しいログイン・ポリシーを作成します。                                                                 |
| 構文 1    | <code>call sp_iqcopyloginpolicy ('existing-policy-name', 'new-policy-name')</code>                     |
| 構文 2    | <code>sp_iqcopyloginpolicy 'existing-policy-name', 'new-policy-name'</code>                            |
| 構文 3    | <code>sp_iqcopyloginpolicy existing-policy-name, new-policy-name<br/>policy_name ]</code>              |
| 使用法     | <b>existing policy name</b> コピーするログイン・ポリシー。<br><b>new policy name</b> 作成する新しいログイン・ポリシーの名前 (CHAR(128))。 |
| パーミッション | DBA 権限が必要です。                                                                                           |
| 参照      | <a href="#">「sp_iqpassword プロシージャ」 (439 ページ)</a>                                                       |

『リファレンス：文とオプション』の「第1章 SQL 文」の「CREATE USER 文」

例 次に示すストアド・プロシージャは、ログイン・ポリシー・オプション値を、*root* という既存のログイン・ポリシーからコピーすることで、*lockeduser* という名前の新しいログイン・ポリシーを作成します。

```
call sp_iqcopyloginpolicy ('root','lockeduser')
```

## sp\_iqcursorinfo プロシージャ

機能 サーバで現在オープンしているカーソルに関する詳細情報を表示します。

構文 `sp_iqcursorinfo [ cursor-name ] [, conn-handle ]`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法 **cursor-name** カーソルの名前。このパラメータのみを指定した場合、`sp_iqcursorinfo` はすべての接続内で指定の名前を持つすべてのカーソルに関する情報を返します。

**conn-handle** 接続 ID を表す整数。このパラメータのみを指定した場合、`sp_iqcursorinfo` は指定の接続内のすべてのカーソルに関する情報を返します。

`sp_iqcursorinfo` プロシージャは、パラメータなしで呼び出せます。パラメータを指定しない場合、`sp_iqcursorinfo` は、現在サーバ上でオープンしているすべてのカーソルに関する情報を返します。両方のパラメータを指定した場合、`sp_iqcursorinfo` は、指定の名前を持つすべてのカーソルに関する情報と、指定の接続内のすべてのカーソルに関する情報をレポートします。

最初のパラメータを指定せずに、2 番目のパラメータを指定する場合は、指定しないパラメータの位置に NULL を指定する必要があります。たとえば、`sp_iqcursorinfo NULL, 1` のように指定します。

表 7-11 : `sp_iqcursorinfo` の使用例

| 構文                                     | 出力                                                        |
|----------------------------------------|-----------------------------------------------------------|
| <code>sp_iqcursorinfo</code>           | サーバで現在オープンしているすべてのカーソルに関する情報を表示します。                       |
| <code>sp_iqcursorinfo 'cursor1'</code> | すべての接続内で <code>cursor1</code> という名前のすべてのカーソルに関する情報を表示します。 |

| 構文                                        | 出力                                                       |
|-------------------------------------------|----------------------------------------------------------|
| <code>sp_iqcursorinfo NULL, 3</code>      | 接続 3 のすべてのカーソルに関する情報を表示します。                              |
| <code>sp_iqcursorinfo 'cursor2', 4</code> | 接続 4 内で <code>cursor2</code> という名前のすべてのカーソルに関する情報を表示します。 |

## 説明

`sp_iqcursorinfo` ストアド・プロシージャは、サーバ上で現在オープンしているカーソルについて詳細情報を表示します。`sp_iqcursorinfo` プロシージャにより、データベース管理者はこの1つのストアド・プロシージャを使用するだけで、カーソル・ステータスを監視し、更新、削除、および挿入されたローの数などの統計情報を表示できます。

1つ以上のパラメータを指定した場合、指定したパラメータにより結果がフィルタされます。たとえば、`cursor-name` を指定した場合、指定のカーソルに関する情報のみが表示されます。`conn-handle` を指定した場合、`sp_iqcursorinfo` は指定の接続内のカーソルに関する情報のみを返します。パラメータを指定しない場合、`sp_iqcursorinfo` は現在サーバ上でオープンしているすべてのカーソルに関する情報を表示します。

`sp_iqcursorinfo` プロシージャは、次のカラムの情報を返します。

表 7-12 : `sp_iqcursorinfo` のカラム

| カラム名       | 説明                                                                                   |
|------------|--------------------------------------------------------------------------------------|
| Name       | カーソルの名前                                                                              |
| ConnHandle | 接続の ID 番号                                                                            |
| IsUpd      | Y : カーソルは更新可能、N : カーソルを更新できない                                                        |
| IsHold     | Y : カーソルはホールド・カーソル、N : カーソルはホールド・カーソルではない                                            |
| IQConnID   | <code>.iqmsg</code> ファイル内のすべてのメッセージに表示される 10 桁の接続 ID。これは、サーバ・セッション内でユニークな、単純増加する整数です |
| UserID     | カーソルを作成し、実行したユーザのユーザ ID (またはユーザ名)                                                    |
| CreateTime | カーソルが作成された時刻                                                                         |
| CurrentRow | 結果セット内のカーソルの現在位置                                                                     |
| NumFetch   | カーソルがローをフェッチする回数。同じローを複数回フェッチできます                                                    |
| NumUpdate  | カーソルが更新可能な場合に、カーソルがローを更新する回数。同じローを複数回更新できます                                          |
| NumDelete  | カーソルが更新可能な場合に、カーソルがローを削除する回数                                                         |
| NumInsert  | カーソルが更新可能な場合に、カーソルがローを挿入する回数                                                         |

| カラム名       | 説明                           |
|------------|------------------------------|
| RWTabOwner | カーソルによって RW モードで開かれるテーブルの所有者 |
| RWTabName  | カーソルによって RW モードで開かれるテーブルの名前  |
| CmdLine    | ユーザが実行したコマンドの最初の 4096 文字     |

例 サーバで現在オープンしているすべてのカーソルに関する情報を表示します。

```
sp_iqcursorinfo
```

| Name  | ConnHandle | IsUpd | IsHold | IQConnID | UserID |
|-------|------------|-------|--------|----------|--------|
| crsr1 | 1          | Y     | N      | 118      | DBA    |
| crsr2 | 3          | N     | N      | 118      | DBA    |

  

| CreateTime              | CurrentRow | NumFetch  | NumUpdate |
|-------------------------|------------|-----------|-----------|
| 2009-06-26 15:24:36.000 | 19         | 100000000 | 200000000 |
| 2009-06-26 15:38:38.000 | 20000      | 200000000 |           |

  

| NumDelete | NumInsert  | RWTabOwner | RWTabName | CmdLine                        |
|-----------|------------|------------|-----------|--------------------------------|
| 20000000  | 3000000000 | DBA        | test1     | call proc1 ()<br>call proc2 () |

参照 『リファレンス：文とオプション』の「[DECLARE CURSOR 文 \[ESQL\] \[SP\]](#)」、[「DECLARE CURSOR 文 \[T-SQL\]](#)」、[「UPDATE \(位置付け\) 文 \[ESQL\] \[SP\]](#)」、[「DELETE \(位置付け\) 文 \[ESQL\] \[SP\]](#)」、[「FORCE\\_NO\\_SCROLL\\_CURSORS オプション](#)」、および [FORCE\\_UPDATABLE\\_CURSORS オプション](#)

『システム管理ガイド 第 2 巻』の「[第 1 章 プロシージャとバッチの使用](#)」の「[プロシージャでのカーソルの使用](#)」

『システム管理ガイド 第 1 巻』の「[第 10 章 トランザクションとバージョン管理](#)」の「[トランザクション内のカーソル](#)」

## sp\_iqdatatype プロシージャ

機能 システム・データ型およびユーザ定義データ型に関する情報を表示します。

構文 **sp\_iqdatatype** [ *type-name* ], [ *type-owner* ], [ *type-type* ]



## パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

## 使用法

**type-name** データ型の名前。

**type-owner** データ型の作成者の名前。

**type-type** データ型のタイプ。指定できる値は次のとおりです。

- **SYSTEM** : システム定義のデータ型 (ユーザ **SYS** または **dbo** によって所有されているデータ型) に関する情報のみを表示します。
- **ALL** : ユーザおよびシステムのデータ型に関する情報を表示します。
- その他の値 : ユーザのデータ型に関する情報を表示します。

**sp\_iqdatatype** プロシージャは、パラメータなしで呼び出せます。パラメータを指定しない場合、ユーザ定義データ型 (**dbo** または **SYS**) によって所有されていないデータ型) に関する情報のみがデフォルトで表示されます。

最初の 2 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に **NULL** を指定する必要があります。たとえば、**sp\_iqdatatype NULL, NULL, SYSTEM** や **sp\_iqdatatype NULL, user1** のように指定します。

表 7-13 : **sp\_iqdatatype** の使用例

| 構文                                     | 出力                                                                                                        |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>sp_iqdatatype</b>                   | データベース内のすべてのユーザ定義データ型に関する情報を表示します。                                                                        |
| <b>sp_iqdatatype address</b>           | <b>address</b> という名前のユーザ定義データ型に関する情報を表示します。                                                               |
| <b>sp_iqdatatype non_existing_type</b> | データ型 <b>non_existing_type</b> が存在しないため、ローは返されません。                                                         |
| <b>sp_iqdatatype NULL, DBA</b>         | <b>DBA</b> によって所有されているすべてのユーザ定義データ型に関する情報を表示します。                                                          |
| <b>sp_iqdatatype address, DBA</b>      | <b>DBA</b> によって所有されているデータ型 <b>address</b> に関する情報を表示します。                                                   |
| <b>sp_iqdatatype rowid</b>             | <b>rowid</b> は、システム定義のデータ型です。 <b>rowid</b> という名前のユーザ定義データ型が存在しない場合、ローは返されません (デフォルトでは、ユーザ定義データ型のみが返されます)。 |
| <b>sp_iqdatatype rowid, SYS</b>        | <b>rowid</b> データ型はユーザ定義データ型ではないため、ローは返されません (デフォルトでは、ユーザ定義データ型のみが返されます)。                                  |

| 構文                                | 出力                                                          |
|-----------------------------------|-------------------------------------------------------------|
| sp_iqdatatype NULL, NULL, SYSTEM  | すべてのシステム定義のデータ型 (dbo または SYS によって所有されているデータ型) に関する情報を表示します。 |
| sp_iqdatatype rowid, NULL, SYSTEM | システム・データ型 rowid に関する情報を表示します。                               |
| sp_iqdatatype NULL, NULL, 'ALL'   | rowid データ型に関する情報を表示します。                                     |

## 説明

sp\_iqdatatype ストアド・プロシージャは、データベース内のシステムおよびユーザ定義データ型に関する情報を表示します。ユーザ定義データ型は、ドメインと呼ばれる場合もあります。あらかじめ定義されているドメイン名は、sp\_iqdatatype 出力には含まれません。

1 つ以上のパラメータを指定した場合、指定したパラメータにより sp\_iqdatatype の結果がフィルタされます。たとえば、*type-name* を指定した場合、指定のデータ型に関する情報のみが表示されます。*type-owner* を指定した場合、sp\_iqdatatype は指定の所有者によって所有されているデータ型に関する情報のみを返します。パラメータを指定しない場合、sp\_iqdatatype はデータベース内のすべてのユーザ定義データ型に関する情報を表示します。

sp\_iqdatatype プロシージャは、次のカラムに情報を返します。

表 7-14 : sp\_iqdatatype のカラム

| カラム名      | 説明                                                               |
|-----------|------------------------------------------------------------------|
| type_name | データ型の名前。                                                         |
| creator   | データ型の所有者。                                                        |
| nulls     | Y は、ユーザ定義データ型で NULL を許可することを示し、N はユーザ定義データ型で NULL を許可しないことを示します。 |
| width     | 文字列カラムでは長さ、数値カラムでは精度、その他のデータ型では格納サイズ (バイト数) を表示します。              |
| scale     | 数値データ型カラムでは小数点以下の桁数、その他のデータ型では 0 を表示します。                         |
| “default” | データ型のデフォルト値。                                                     |
| “check”   | データ型の CHECK 条件。                                                  |

例 address という名前のユーザ定義データ型に関する情報を表示します。

```
sp_iqdatatype address
```

```
type_name creator nulls width scale “default” “check”
address DBA Y 5 0 (NULL) (NULL)
```

参照 『リファレンス：文とオプション』の「[CREATE DOMAIN 文](#)」  
[第3章 SQL データ型](#)

## sp\_iqdbsize プロシージャ

機能 現在のデータベースのサイズを表示します。

構文 `sp_iqdbsize(  
 [ main ]  
 )`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 データベースの合計サイズを返します。また、メモリにデータベースを保持するために必要なページ数と、(ディスク上で) データベースを圧縮した場合の IQ ページ数も返します。

表 7-15 : sp\_iqdbsize のカラム

| カラム名             | 説明                                                                      |
|------------------|-------------------------------------------------------------------------|
| Database         | データベース・ファイルのパス名。                                                        |
| Physical Blocks  | データベースの合計サイズ(ブロック数)。                                                    |
| KBytes           | データベースの合計サイズ(キロバイト)。                                                    |
| Pages            | IQ ページの合計数。                                                             |
| Compressed Pages | (ディスク上で)圧縮された IQ ページの合計数。Pages のサブセット。                                  |
| NBlocks          | テーブルおよびジョイン・インデックスへのデータの格納に使用されている IQ ブロック内の合計サイズ。                      |
| Catalog Blocks   | テーブルおよびジョイン・インデックスのメタデータの格納に使用されている領域の合計サイズ (IQ ブロック単位)。Nblocks のサブセット。 |

sp\_iqdbsize のカラムの説明を次に示します。

**Database** 現在のデータベース・ファイルのパス名を返します。

**Physical Blocks** IQ データベースは、1 つまたは複数の DB 領域で構成されています。各 DB 領域は、当初メガバイト単位で指定されたサイズに固定されています。このメガバイトの値は、IQ ページ・サイズと、その IQ ページ・サイズに対応するブロック・サイズを使用してブロックに変換されます。Physical Blocks カラムには、各 Sybase IQ DB 領域のサイズの累積値がブロック単位で反映されます。

IQ ページ・サイズとブロック・サイズの対応については、『パフォーマンス&チューニング・ガイド』の「[第4章 システム・リソースの管理](#)」を参照してください。

**KBytes** データベースの合計サイズ (キロバイト) です。この値は、ブロックで表したデータベースの合計サイズ (前の `sp_iqdbsize` カラムの Physical Blocks) に、ブロック・サイズを乗算したものです。このブロック・サイズは IQ ページ・サイズに依存します。

**Pages** テーブルおよびジョイン・インデックスに格納されているすべてのデータとそれらのオブジェクトのメタデータをメモリ内に表示するために必要な IQ ページの合計数です。この値は常に、Compressed Pages (次の `sp_iqdbsize` カラム) の値よりも大きいかこの値に等しくなります。

**Compressed Pages** テーブルおよびジョイン・インデックスに格納されているすべてのデータとそれらのオブジェクトのメタデータをディスクに格納するために必要な IQ ページの合計数です。IQ ページがメモリからディスクに書き込まれるとき、Sybase IQ はページを圧縮するため、この値は、Pages (前の `sp_iqdbsize` カラム) の値よりも小さいかこの値に等しくなります。`sp_iqdbsize` Compressed Pages カラムは、圧縮されたページの数を表します。

**NBlocks** テーブルおよびジョイン・インデックスへのデータの格納に使用されている領域の合計サイズ (ブロック単位)。この値は常に、`sp_iqdbsize` Physical Blocks の値よりも小さいかこの値に等しくなります。

**Catalog Blocks** テーブルおよびジョイン・インデックスのメタデータの格納に使用されている領域の合計サイズ (ブロック単位)。

例 この例は、データベース `iqdemo` のサイズ情報を表示します。

```
sp_iqdbsize

Database
PhysicalBlocks KBytes Pages CompressedPages NBlocks CatalogBlocks
=====
/system1/sybase/IQ-15_1/demo/iqdemo.db
 1280 522 688 257 1119 18
```

- 参照 『パフォーマンス&チューニング・ガイド』の「第4章 システム・リソースの管理」の「メモリ使用の概要」の「ページ・サイズの指定」  
『システム管理ガイド 第1巻』の「第5章 データベース・オブジェクトの使用」の「データベース・オブジェクトの使用」

## sp\_iqdbspace プロシージャ

- 機能 各 IQ DB 領域についての詳細情報を表示します。
- 構文 `sp_iqdbspace [ dbspace-name ]`
- パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。
- 説明 iqdbspace の情報を使用して、データの移動が必要かどうかを判断できます。また、移動されたデータに関しては、古いバージョンの割り付けが解除されているかどうかを確認できます。sp\_iqdbspace は次の情報を表示します。

表 7-16 : sp\_iqdbspace のカラム

| カラム名        | 説明                                                                                       |
|-------------|------------------------------------------------------------------------------------------|
| DBSpaceName | CREATE DBSPACE 文で指定された DB 領域の名前。CASE RESPECT を指定して作成されたデータベースの場合、DB 領域名は大文字と小文字は区別されません。 |
| DBSpaceType | DB 領域のタイプ (MAIN または TEMPORARY のみ)。                                                       |
| Writable    | T (書き込み可能) または F (書き込み不可)。                                                               |
| Online      | T (オンライン) または F (オフライン)。                                                                 |
| Usage       | DB 領域のすべてのファイルで現在使用されている DB 領域の割合。                                                       |
| TotalSize   | DB 領域のすべてのファイルの合計サイズ。単位は、B (バイト)、K (キロバイト)、M (メガバイト)、G (ギガバイト)、T (テラバイト)、または P (ペタバイト)。  |
| Reserve     | DB 領域のすべてのファイルに追加できる予約領域の合計。                                                             |
| NumFiles    | DB 領域内のファイルの数。                                                                           |
| NumRWFiles  | DB 領域内の読み込み/書き込みファイルの数。                                                                  |
| Stripingon  | T (オン) または F (オフ)。                                                                       |
| StripeSize  | ディスク・ストライピングが有効になっている場合、次の DB 領域に移動するまでに DB 領域に書き込まれたデータの量。                              |

| カラム名     | 説明                                      |
|----------|-----------------------------------------|
| BlkTypes | ユーザ・データと内部システム構造が使用している領域 (表 7-17 を参照)。 |
| OkToDrop | DB 領域を削除できる場合は 'Y'、それ以外の場合は 'N'。        |

表 7-17 に、ブロック・タイプ識別子の値を示します。

**表 7-17 : sp\_iqdbspace のブロック・タイプ**

| 識別子 | ブロック・タイプ            |
|-----|---------------------|
| A   | アクティブなバージョン         |
| B   | バックアップ構造            |
| C   | チェックポイント・ログ         |
| D   | データベースの識別情報         |
| F   | フリー・リスト             |
| G   | グローバル・フリー・リスト・マネージャ |
| H   | フリー・リストのヘッダ・ブロック    |
| I   | インデックス・アドバイスの格納     |
| M   | マルチプレックス CM*        |
| O   | 旧バージョン              |
| T   | テーブルの使用             |
| U   | インデックスの使用           |
| N   | カラムの使用              |
| X   | チェックポイントでの削除        |

\*マルチプレックス・コミット ID ブロック (実際は 128 ブロック) は、シンプレックス・データベースで使用されていない場合でも、すべての IQ データベースに存在します。マルチプレックス機能の詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。

例

次の出力には、DB 領域に関する情報が表示されています。

```
sp_iqdbspace;
```

| DBSpace Name     | DBSpaceType | Writable | Online | Usage | Total Size | Reserve | Num Files | Num RWFiles | Striping | Stripe Size | Block Types          | Ok To Drop |
|------------------|-------------|----------|--------|-------|------------|---------|-----------|-------------|----------|-------------|----------------------|------------|
| IQ_MAIN          | MAIN        | T        | T      | 55    | 75M        | 200M    | 1         | 1           | T        | 1K          | 1H, 5169A, 190       | N          |
| IQ__SYSTEM__MAIN | MAIN        | T        | T      | 21    | 300M       | 50M     | 1         | 1           | F        | 8K          | 1H, 7648F, 32D, 128M | N          |
| IQ_SYSTEM__TEMP  | TEMPORARY   | T        | T      | 1     | 100M       | 50M     | 1         | 1           | F        | 8K          | 1H, 64F, 32A         | N          |

参照

[「sp\\_iqdbspaceinfo プロシージャ」 \(389 ページ\)](#)[「sp\\_iqdbspaceobjectinfo プロシージャ」 \(393 ページ\)](#)[「sp\\_iqindexinfo プロシージャ」 \(421 ページ\)](#)

## sp\_iqdbspaceinfo プロシージャ

機能

指定のテーブルまたはジョイン・インデックスで使用される各オブジェクトおよびサブオブジェクトのサイズを表示します。

構文

```
sp_iqdbspaceinfo [dbspace-name] [, owner_name] [,
object_name] [, object-type]
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法

**dbspace\_name** 指定した場合、sp\_iqdbspaceinfo は、指定の DB 領域内のコンポーネントを持つ各テーブルを 1 行に表示します。指定しない場合、プロシージャはデータベース内のすべての DB 領域の情報を表示します。

**owner\_name** オブジェクトの所有者。指定した場合、`sp_iqdbspaceinfo` は、指定の所有者を持つテーブルおよびジョイン・インデックスについてのみ出力を表示します。指定しない場合、`sp_iqdbspaceinfo` は、データベース内のすべてのユーザのテーブルおよびジョイン・インデックスに関する情報を表示します。

**object\_name** テーブルまたはジョイン・インデックスの名前。指定しない場合、`sp_iqdbspaceinfo` は、データベース内のすべてのテーブルおよびジョイン・インデックスに関する情報を表示します。

**object\_type** 有効なオブジェクト・タイプは、`table` (テーブル情報の場合) または `joinindex` (ジョイン・インデックス情報の場合) です。指定しない場合、オブジェクト・タイプのデフォルトは `table` になります。

すべてのパラメータがオプションであり、どのパラメータも別のパラメータの値とは関係なく指定できます。

`sp_iqdbspaceinfo` ストアド・プロシージャは、`dbspace_name`、`object_name`、および `owner_name` の解釈のために、ワイルドカード文字をサポートしています。これは、`LIKE` 句がクエリ内のパターンを照合するのと同じ方法で、指定のパターンと一致するすべての DB 領域の情報を表示します。

## 説明

`sp_iqdbspaceinfo` は、各 DB 領域に存在するオブジェクトによって使用される領域の大きさを `DBA` に示します。`DBA` はこの情報を使用して、DB 領域を削除する前に移動する必要があるオブジェクトを判断できます。サブオブジェクト・カラムには、整数の量でレポートされるサイズが表示されます。各値の後ろには、サフィックス `B`、`K`、`M`、`G`、`T`、または `P` が付き、これらはそれぞれキロバイト、メガバイト、ギガバイト、テラバイト、およびペタバイトを表します。

テーブルの場合、`sp_iqdbspaceinfo` は、すべてのサブオブジェクトのサイジング情報を表示します(サフィックス `B`、`K`、`M`、`G`、`T`、または `P` を持つ整数の量を使用します)。ジョイン・インデックスの場合は、ジョイン・インデックスと、それに関連するすべてのサブオブジェクトについてのサイジング情報を表示します。出力は、`dbspace_name`、`object_name`、および `owner_name` でソートされます。

**表 7-18 : `sp_iqdbspaceinfo` のカラム**

| カラム名                      | 説明                                                              |
|---------------------------|-----------------------------------------------------------------|
| <code>dbspace_name</code> | DB 領域の名前。                                                       |
| <code>object_type</code>  | オブジェクトのタイプ ( <code>table</code> または <code>joinindex</code> のみ)。 |
| <code>owner</code>        | オブジェクトの所有者の名前。                                                  |
| <code>object_name</code>  | DB 領域にあるオブジェクトの名前 (テーブルおよびジョイン・インデックス・タイプのみ)。                   |
| <code>object_id</code>    | オブジェクトのグローバル・オブジェクト ID。                                         |



| カラム名              | 説明                                                                                    |
|-------------------|---------------------------------------------------------------------------------------|
| id                | オブジェクトのテーブル ID またはジョイン・インデックス ID。                                                     |
| columns           | 指定の DB 領域のカラム記憶領域のサイズ。                                                                |
| indexes           | 指定の DB 領域のインデックス記憶領域のサイズ。システムで作成されたインデックス (一意性制約または FP インデックスの HG インデックスなど) は使用できません。 |
| metadata          | 指定の DB 領域のメタデータ・オブジェクトの記憶領域サイズ。                                                       |
| primary_key       | 指定の DB 領域のプライマリ・キー関連オブジェクトの記憶領域サイズ。                                                   |
| unique_constraint | 指定の DB 領域の一意性制約関連オブジェクトの記憶領域サイズ。                                                      |
| foreign_key       | 指定の DB 領域の外部キー関連オブジェクトの記憶領域サイズ。                                                       |

**注意**

-r スイッチ (読み込み専用) で開始したサーバに対して `sp_iqdbspaceinfo` を実行した場合、Msg 13768, Level 14, State 0: SQL Anywhere Error -757: Modifications not permitted for read-only database というエラーが表示されます。これは予期された動作です。`sp_iqdbspace`、`sp_iqfile`、`sp_iqdbspaceobjectinfo`、`sp_iqobjectinfo` などの他のストアド・プロシージャでは、エラーは発生しません。

**例**

データベース内のすべての DB 領域のすべてのテーブルにある、すべてのオブジェクトおよびサブオブジェクトのサイズを表示します。

```
sp_iqdbspaceinfo
```

| dbspace_name | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | table       | DBA    | empl            | 3689      | 741 | 96K     |
| iq_main      | table       | DBA    | iq_dummy        | 3686      | 740 | 24K     |
| iq_main      | table       | DBA    | sale            | 3698      | 742 | 96K     |
| iq_main      | table       | GROUPO | Contacts        | 3538      | 732 | 288K    |
| iq_main      | table       | GROUPO | Customers       | 3515      | 731 | 240K    |
| iq_main      | table       | GROUPO | Departments     | 3632      | 738 | 72K     |
| iq_main      | table       | GROUPO | Employees       | 3641      | 739 | 408K    |
| iq_main      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 72K     |
| iq_main      | table       | GROUPO | FinancialData   | 3621      | 737 | 96K     |
| iq_main      | table       | GROUPO | Products        | 3593      | 735 | 272K    |
| iq_main      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 120K    |
| iq_main      | table       | GROUPO | SalesOrders     | 3565      | 733 | 144K    |

| indexes | metadata | primary_key | unique_constraint | foreign_key |
|---------|----------|-------------|-------------------|-------------|
| 0B      | 1.37M    | 0B          | 0B                | 0B          |
| 0B      | 464K     | 0B          | 0B                | 0B          |
| 0B      | 1.22M    | 0B          | 0B                | 0B          |
| 0B      | 5.45M    | 24K         | 0B                | 48K         |
| 48K     | 4.63M    | 24K         | 0B                | 0B          |
| 0B      | 1.78M    | 24K         | 0B                | 48K         |
| 0B      | 8.03M    | 24K         | 0B                | 48K         |
| 0B      | 1.53M    | 24K         | 0B                | 0B          |
| 0B      | 2.19M    | 24K         | 0B                | 48K         |
| 192K    | 4.67M    | 24K         | 0B                | 0B          |
| 0B      | 2.7M     | 24K         | 0B                | 104K        |
| 0B      | 3.35M    | 24K         | 0B                | 144K        |

データベース内の指定の DB 領域の指定のユーザによって所有されている、すべてのオブジェクトおよびサブオブジェクトのサイズを表示します。

```
sp_iqdbspaceinfo iq_main,GROUPO
```

| dbspace_name | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | table       | GROUPO | Contacts        | 3538      | 732 | 288K    |
| iq_main      | table       | GROUPO | Customers       | 3515      | 731 | 240K    |
| iq_main      | table       | GROUPO | Departments     | 3632      | 738 | 72K     |
| iq_main      | table       | GROUPO | Employees       | 3641      | 739 | 408K    |
| iq_main      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 72K     |
| iq_main      | table       | GROUPO | FinancialData   | 3621      | 737 | 96K     |
| iq_main      | table       | GROUPO | Products        | 3593      | 735 | 272K    |
| iq_main      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 120K    |
| iq_main      | table       | GROUPO | SalesOrders     | 3565      | 733 | 144K    |

| indexes | metadata | primary_key | unique_constraint | foreign_key |
|---------|----------|-------------|-------------------|-------------|
| 0B      | 5.45M    | 24K         | 0B                | 48K         |
| 48K     | 4.63M    | 24K         | 0B                | 0B          |
| 0B      | 1.78M    | 24K         | 0B                | 48K         |
| 0B      | 8.03M    | 24K         | 0B                | 48K         |
| 0B      | 1.53M    | 24K         | 0B                | 0B          |
| 0B      | 2.19M    | 24K         | 0B                | 48K         |
| 192K    | 4.67M    | 24K         | 0B                | 0B          |
| 0B      | 2.7M     | 24K         | 0B                | 104K        |
| 0B      | 3.35M    | 24K         | 0B                | 144K        |

データベース内の指定の DB 領域の指定のユーザによって所有されている、指定のオブジェクトとそのサブオブジェクトのサイズを表示します。

```
sp_iqdbspaceinfo iq_main,GROUPO,Departments
```

| dbspace_name | object_type | owner  | object_name | object_id | id  | columns |
|--------------|-------------|--------|-------------|-----------|-----|---------|
| iq_main      | table       | GROUPO | Departments | 3632      | 738 | 72K     |

| indexes | metadata | primary_key | unique_constraint | foreign_key |
|---------|----------|-------------|-------------------|-------------|
| 0B      | 1.78M    | 24K         | 0B                | 48K         |

参照 [「sp\\_iqdbspace プロシージャ」\(387 ページ\)](#) および [「sp\\_iqindexinfo プロシージャ」\(421 ページ\)](#)

## sp\_iqdbspaceobjectinfo プロシージャ

**機能** 指定の DB 領域のテーブルおよびジョイン・インデックス・タイプのオブジェクトと、そのサブオブジェクト (カラム、インデックス、メタデータ、プライマリ・キー、一意性制約、外部キー、およびパーティション) をリストします。

**構文** `sp_iqdbspaceobjectinfo [ dbspace-name ] [ , owner_name ] [ , object_name ] [ , object-type ]`

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**使用法** **dbspace-name** 指定した場合、sp\_iqdbspaceobjectinfo は、指定の DB 領域についてのみ出力を表示します。指定しない場合、データベース内のすべての DB 領域の情報を表示します。

**owner-name** オブジェクトの所有者。指定した場合、sp\_iqdbspaceobjectinfo は、指定の所有者を持つテーブルおよびジョイン・インデックスについてのみ出力を表示します。指定しない場合、sp\_iqdbspaceobjectinfo は、データベース内のすべてのユーザのテーブルおよびジョイン・インデックスに関する情報を表示します。

**object-name** テーブルまたはジョイン・インデックスの名前。指定しない場合、sp\_iqdbspaceobjectinfo は、データベース内のすべてのテーブルおよびジョイン・インデックスに関する情報を表示します。

**object-type** 有効なオブジェクト・タイプは、table (テーブル情報の場合) または joinindex (ジョイン・インデックス情報の場合) です。指定しない場合、オブジェクト・タイプのデフォルトは table になります。

すべてのパラメータがオプションであり、どのパラメータも他のパラメータの値とは関係なく指定できます。

sp\_iqdbspaceobjectinfo ストアド・プロシージャは、*dbspace\_name*、*object\_name*、および *owner\_name* の解釈のために、ワイルドカード文字をサポートしています。これは、LIKE 句がクエリ内のパターンを照合するのと同じ方法で、指定のパターンと一致するすべての DB 領域の情報を表示します。

## 説明

テーブルの場合、`sp_iqdbspaceobjectinfo` は、関連するすべてのサブオブジェクトの要約情報を表示します。ジョイン・インデックスの場合は、ジョイン・インデックスと、それに関連するすべてのサブオブジェクトについてのサイジング情報を表示します。ストアド・プロシージャの出力は、`dbspace_name`、`owner`、および `object_name` でソートされます。

`sp_iqdbspaceobjectinfo` は、入力パラメータ値に基づいて次の情報を表示します。

表 7-19 : `sp_iqdbspaceobjectinfo` のカラム

| カラム名                           | 説明                                                                                                                                                        |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dbspace_name</code>      | DB 領域の名前。                                                                                                                                                 |
| <code>object_type</code>       | テーブルまたはジョイン・インデックス。                                                                                                                                       |
| <code>owner</code>             | オブジェクトの所有者の名前。                                                                                                                                            |
| <code>object_name</code>       | DB 領域にあるオブジェクトの名前 (テーブルおよびジョイン・インデックス・タイプのみ)。                                                                                                             |
| <code>object_id</code>         | オブジェクトのグローバル・オブジェクト ID。                                                                                                                                   |
| <code>id</code>                | オブジェクトのテーブル ID またはジョイン・インデックス ID。                                                                                                                         |
| <code>columns</code>           | 指定の DB 領域にあるテーブル・カラムの数。カラム、またはいずれかのカラムパーティションが DB 領域にある場合、その DB 領域に存在しているものとして数えられます。結果は n/N フォームで表示されます (テーブルの合計 N 個のカラムのうち n 個が、指定された DB 領域に存在します)。     |
| <code>indexes</code>           | 指定の DB 領域にあるテーブルのユーザ定義インデックスの数。n/N フォームで表示されます (テーブル上の合計 N 個のインデックスのうち n 個が、指定された DB 領域に存在します)。一意性制約の場合、これには、FP インデックスや HG インデックスなどのシステム作成のインデックスは含まれません。 |
| <code>metadata</code>          | サブオブジェクトのメタデータ情報もこの DB 領域にあるかどうかを示すブール・フィールド (Y/N)。                                                                                                       |
| <code>primary_key</code>       | テーブルのプライマリ・キー (存在する場合) がこの DB 領域にあるかどうかを示すブール・フィールド (1/0)。                                                                                                |
| <code>unique_constraint</code> | 指定の DB 領域にあるテーブルの一意性制約の数。n/N フォームで表示されます (テーブル上の合計 N 個の一意性制約のうち n 個が、指定された DB 領域に存在します)。                                                                  |
| <code>foreign_key</code>       | 指定の DB 領域にあるテーブルの外部キーの数。n/N フォームで表示されます (テーブル上の合計 N 個の外部キーのうち n 個が、指定された DB 領域に存在します)。                                                                    |
| <code>partitions</code>        | 指定の DB 領域にあるテーブルのパーティションの数。n/N フォームで表示される (テーブルの合計 N 個のパーティションのうち n 個が、指定された DB 領域に存在します)。                                                                |

例

データベース内の特定の DB 領域に関する情報を表示します。

```
sp_iqdbspaceobjectinfo iq_main
```

| dbspace_name | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | table       | DBA    | empl            | 3689      | 741 | 4/4     |
| iq_main      | table       | DBA    | iq_dummy        | 3686      | 740 | 1/1     |
| iq_main      | table       | DBA    | sale            | 3698      | 742 | 4/4     |
| iq_main      | table       | GROUPO | Contacts        | 3538      | 732 | 12/12   |
| iq_main      | table       | GROUPO | Customers       | 3515      | 731 | 10/10   |
| iq_main      | table       | GROUPO | Departments     | 3632      | 738 | 3/3     |
| iq_main      | table       | GROUPO | Employees       | 3641      | 739 | 21/21   |
| iq_main      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 3/3     |
| iq_main      | table       | GROUPO | FinancialData   | 3621      | 737 | 4/4     |
| iq_main      | table       | GROUPO | Products        | 3593      | 735 | 8/8     |
| iq_main      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 5/5     |
| iq_main      | table       | GROUPO | SalesOrders     | 3565      | 733 | 6/6     |

| indexes | metadata | primary_key | unique_constraint | foreign_key | partitions |
|---------|----------|-------------|-------------------|-------------|------------|
| 0/0     | Y        | 0           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 0           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 0           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |
| 1/1     | Y        | 1           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |
| 4/4     | Y        | 1           | 0/0               | 0/0         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 2/2         | 0/0        |
| 0/0     | Y        | 1           | 0/0               | 3/3         | 0/0        |

データベース内の指定の DB 領域の指定のユーザによって所有されているオブジェクトに関する情報を表示します。

```
sp_iqdbspaceobjectinfo iq_main,GROUPO
```

| dbspace_name | object_type | owner  | object_name     | object_id | id  | columns |
|--------------|-------------|--------|-----------------|-----------|-----|---------|
| iq_main      | table       | GROUPO | Contacts        | 3538      | 732 | 12/12   |
| iq_main      | table       | GROUPO | Customers       | 3515      | 731 | 10/10   |
| iq_main      | table       | GROUPO | Departments     | 3632      | 738 | 3/3     |
| iq_main      | table       | GROUPO | Employees       | 3641      | 739 | 21/21   |
| iq_main      | table       | GROUPO | FinancialCodes  | 3612      | 736 | 3/3     |
| iq_main      | table       | GROUPO | FinancialData   | 3621      | 737 | 4/4     |
| iq_main      | table       | GROUPO | Products        | 3593      | 735 | 8/8     |
| iq_main      | table       | GROUPO | SalesOrderItems | 3580      | 734 | 5/5     |
| iq_main      | table       | GROUPO | SalesOrders     | 3565      | 733 | 6/6     |

| indexes | metadata | primary_key | unique_constraint | foreign_key | partitions |
|---------|----------|-------------|-------------------|-------------|------------|
| 0/0     | Y        | 1           | 0/0               | 1/1         | 0/0        |
| 1/1     | Y        | 1           | 0/0               | 0/0         | 0/0        |

|     |   |   |     |     |     |
|-----|---|---|-----|-----|-----|
| 0/0 | Y | 1 | 0/0 | 1/1 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 1/1 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 0/0 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 1/1 | 0/0 |
| 4/4 | Y | 1 | 0/0 | 0/0 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 2/2 | 0/0 |
| 0/0 | Y | 1 | 0/0 | 3/3 | 0/0 |

`sp_iqdbspaceobjectinfo` を使用して、オブジェクトの移動に使用できるコマンドを構成します。次の例では、コマンドで `dbspace_x` 上のすべてのテーブルを `dbspace_y` に移動します。

```
SELECT 'ALTER TABLE ' || owner || '.' ||
object_name || ' MOVE TO dbspace_y; '
FROM sp_iqdbspaceobjectinfo()
WHERE object_type = 'table' AND
dbspace_name = 'dbspace_x';
```

次の ALTER TABLE コマンドはその結果です。

```
ALTER TABLE DBA.dt1 MOVE TO dbspace_y;
ALTER TABLE DBA.dt2 MOVE TO dbspace_y;
ALTER TABLE DBA.dt3 MOVE TO dbspace_y;
```

参照

[「sp\\_iqdbspace プロシージャ」\(387 ページ\)](#) および [「sp\\_iqindexinfo プロシージャ」\(421 ページ\)](#)

## sp\_iqdbstatistics プロシージャ

機能

最後に実行された `sp_iqcheckdb` の結果をレポートします。

構文

**sp\_iqdbstatistics**

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

最後に実行された `sp_iqcheckdb` が収集したデータベースの統計情報を表示します。

例

次の例は、`sp_iqdbstatistics` からの出力を表示します。この例では、最後に実行された `sp_iqcheckdb` は、コマンド `sp_iqcheckdb 'allocation database'` です。

| DB Statistics               | Value | Flags |
|-----------------------------|-------|-------|
| ===== ===== =====           |       |       |
| DBCC Allocation Mode Report |       |       |
| ===== ===== =====           |       |       |

```

** DBCC Status |Errors Detected |*****
 DBCC Work units Dispatched |163 |
 DBCC Work units Completed |163 |
=====|=====|=====
Allocation Summary | |
=====|=====|=====
 Blocks Total |8192 |
 Blocks in Current Version |4954 |
 Blocks in All Versions |4954 |
 Blocks in Use |4986 |
 % Blocks in Use |60 |
** Blocks Leaked |32 |*****
=====|=====|=====
Allocation Statistics | |
=====|=====|=====
 Blocks Created in Current TXN |382 |
 Blocks To Drop in Current TXN |382 |
 Marked Logical Blocks |8064 |
 Marked Physical Blocks |4954 |
 Marked Pages |504 |
 Blocks in Freelist |126553 |
 Imaginary Blocks |121567 |
 Highest PBN in Use |5432 |
** 1st Unowned PBN |452 |*****
 Total Free Blocks |3206 |
 Usable Free Blocks |3125 |
 % Free Space Fragmented |2 |
 Max Blocks Per Page |16 |
 1 Block Page Count |97 |
 3 Block Page Count |153 |
 4 Block Page Count |14 |
 ...
 9 Block Hole Count |2 |
 16 Block Hole Count |194 |
 Database Objects Checked |1 |
 B-Array Count |1 |
 Blockmap Identity Count |1 |
=====|=====|=====
Connection Statistics | |
=====|=====|=====

```

参照 [sp\\_iqcheckdb の使用法と sp\\_iqcheckdb の出力の解釈の詳細](#)については、『システム管理ガイド 第1巻』の「[第13章 システムのリカバリとデータベースの修復](#)」を参照してください。

## sp\_iqdroplogin プロシージャ

|         |                                                                                                                                                                |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能      | Sybase IQ のユーザ・アカウントを削除します。                                                                                                                                    |
| 構文 1    | <b>call sp_iqdroplogin ('userid')</b>                                                                                                                          |
| 構文 2    | <b>sp_iqdroplogin 'userid'</b>                                                                                                                                 |
| 構文 3    | <b>sp_iqdroplogin userid</b>                                                                                                                                   |
| 構文 4    | <b>sp_iqdroplogin ('userid')</b>                                                                                                                               |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。                                                                                  |
| 使用法     | <b>userid</b> 削除するユーザのユーザ ID。                                                                                                                                  |
| 説明      | sp_iqdroplogin は指定のユーザを削除します。                                                                                                                                  |
| 例       | 次のストアド・プロシージャ・コールは、ユーザ <b>rose</b> を削除します。<br><pre>sp_iqdroplogin 'rose'<br/>sp_iqdroplogin rose<br/>call sp_iqdroplogin ('rose')</pre>                        |
| 参照      | <a href="#">「sp_iqaddlogin プロシージャ」(353 ページ)</a><br>『リファレンス：文とオプション』の「 <a href="#">REVOKE 文</a> 」<br>『システム管理ガイド 第1巻』の「 <a href="#">第8章 ユーザ ID とパーミッションの管理</a> 」 |

## sp\_iqemptyfile プロシージャ

|         |                                                                               |
|---------|-------------------------------------------------------------------------------|
| 機能      | DB 領域ファイルを空にし、ファイル内のオブジェクトを別の使用可能な読み込み/書き込み dbpace ファイルに移動します。                |
| 構文      | <b>sp_iqemptyfile ( <i>logical-file--name</i> )</b>                           |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |



**説明** `sp_iqemptyfile` は、DB 領域ファイルを空にします。`sp_iqemptyfile` プロシージャを実行するには、DB 領域を読み込み/書き込みにする必要があります。プロシージャは、ファイル内のオブジェクトを別の使用可能な読み込み/書き込み DB 領域ファイルに移動します。使用可能な他の読み込み/書き込み DB 領域ファイルがない場合は、Sybase IQ によってエラー・メッセージが表示されます。

---

**注意** マルチプレックス環境では、コーディネータで `sp_iqemptyfile` のみを実行できます。

プロシージャを正常に完了するには、1 つの読み込み/書き込み DB 領域が使用可能である必要があります。

---

**例** 次の例は、DB 領域 `dbspace1` を空にします。

```
sp_iqemptyfile 'dbspace1'
```

## sp\_iqestjoin プロシージャ

**機能** 指定したテーブルのジョイン・インデックスを作成するために必要な領域を見積もります。

**構文**

```
sp_iqestjoin (table1_name, table1_row_#, table2_name,
 table2_row_#, relation, iq_page_size)
```

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**説明** ジョインするテーブルに基づいて、ジョイン・インデックスで使われる領域の見積もりを返します。このプロシージャでは、指定された IQ ページ・サイズのデフォルト・ブロック・サイズでデータベースが作成されていることを前提として計算します(それ以外の場合、正確な値が返されません)。

修飾されていないテーブル名を指定した場合、ジョインするテーブルを所有するユーザであることを確認します。テーブルの所有者でないユーザは、各テーブルに、'owner.tablename' のような修飾されたテーブル名を指定します。

表 7-20 に、`sp_iqestjoin` のパラメータを示します。

表 7-20 : sp\_iqestjoin のパラメータ

| 名前                  | データ型      | 説明                                                                          |
|---------------------|-----------|-----------------------------------------------------------------------------|
| <i>table1_name</i>  | char(256) | 最初にジョインするテーブルの名前。                                                           |
| <i>table1_row_#</i> | int       | 最初のテーブル内で、ジョイン対象となるロー数。                                                     |
| <i>table2_name</i>  | char(256) | 2 番目にジョインするテーブルの名前。                                                         |
| <i>table2_row_#</i> | int       | 2 番目のテーブル内で、ジョイン対象となるロー数。                                                   |
| <i>relation</i>     | char(9)   | ジョインのタイプ。“one>>many”または“one>>one”(単語と演算子の間にスペースは入らない)。デフォルト値は“one>>many”。   |
| <i>iq_page_size</i> | smallint  | データベースの IQ セグメント用に定義されているページ・サイズ。2 の累乗で、範囲は 1024 ~ 524288、デフォルトは 131072 です。 |

例

```
call sp_iqestjoin ('Customers', 1500000,
'SalesOrders', 15000000, 'one>>many', 65536)
```

| Cases               | Indexsize | Create time | Msg |
|---------------------|-----------|-------------|-----|
| Table1: Customers   |           |             |     |
| Rows: 1500000       |           |             |     |
| Columns:            |           |             |     |
| 8                   |           |             |     |
| Width:              |           |             |     |
| 223                 |           |             |     |
| Table2: SalesOrders |           |             |     |
| Rows: 15000000      |           |             |     |
| Columns:            |           |             |     |
| 9                   |           |             |     |
| Width:              |           |             |     |
| 134                 |           |             |     |
| IQpagesize:         |           |             |     |
| 65536               |           |             |     |
| Min Case            | 48001024  | 3h0m/CPU    |     |
| Max Case            | 95449088  | 9h6m/CPU    |     |
| Avg Case            | 70496256  | 5h53m/CPU   |     |

## sp\_iquestdbspaces プロシージャ

**機能** 指定した合計インデックス・サイズに必要な DB 領域の数とサイズを見積もります。

**構文** `sp_iquestdbspaces ( db_size_in_bytes, iq_page_size, min_#_of_bytes, max_#_of_bytes )`

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**説明** データベース・サイズ、IQ ページ・サイズ、DB 領域セグメントあたりのバイト数の範囲に基づいて、DB 領域セグメントの数とサイズに関する情報を表示します。このプロシージャでは、指定された IQ ページ・サイズのデフォルト・ブロック・サイズでデータベースが作成されていることを前提として計算します (それ以外の場合、正確な値が返されません)。表 7-21 に、sp\_iquestdbspaces のパラメータを示します。

表 7-21 : sp\_iquestdbspaces のパラメータ

| 名前                      | データ型        | 説明                                                                           |
|-------------------------|-------------|------------------------------------------------------------------------------|
| <i>db_size_in_bytes</i> | decimal(16) | データベースのサイズ (バイト)。                                                            |
| <i>iq_page_size</i>     | smallint    | データベースの IQ セグメント用に定義されているページ・サイズ。2 の累乗で、範囲は 65536 ~ 524288、デフォルトは 131072 です。 |
| <i>min_#_of_bytes</i>   | int         | DB 領域セグメントあたりの最小バイト数。デフォルト値は 20,000,000 (20MB) です。                           |
| <i>max_#_of_bytes</i>   | int         | DB 領域セグメントあたりの最大バイト数。デフォルト値は 2,146,304,000 (2.146GB) です。                     |

**使用法** sp\_iquestdbspaces は、データのユニーク性に応じて、4 種類の推奨を表示します。

**min** データの差が少ない場合は、min で推奨されているサイズの DB 領域セグメントだけを作成することもできます。これらの推奨は、最小限の差を持つデータを最大限に圧縮した場合のものです。

**avg** データの差が平均的であれば、min で推奨されているとおりに DB 領域セグメントを作成し、avg で推奨されているサイズで追加のセグメントを作成します。

**max** データの差が大きい (ユニークな値が多い) 場合は、min、avg、および max で推奨されているとおりに DB 領域セグメントを作成します。

**spare** データ内のユニークな値の数がわからない場合は、**min**、**avg**、**max**、および **spare** で推奨されているとおりに DB 領域セグメントを作成します。データをロードした後で、使用していないセグメントはいつでも削除できますが、作成するセグメントが少なすぎると時間がかかります。

❖ **sp\_iqestdbspaces** を他のシステム・ストアド・プロシージャと一緒に使用する

- 1 頻繁にジョインすることが予想されるテーブルの組み合わせすべてに対して、**sp\_iqestjoin** を実行します。
- 2 テーブルの各ペアに対して、推奨されたインデックス・サイズのいずれかを選択します。
- 3 すべてのテーブルに対して選択したインデックス・サイズを合計します。
- 4 すべてのテーブルに対して **sp\_iqestspace** を実行します。
- 5 **sp\_iqestspace** が返したすべての RAW DATA インデックス・サイズを合計します。
- 6 手順 3 の合計を手順 5 の合計に加え、インデックス・サイズの合計を出します。
- 7 手順 6 で出したインデックス・サイズの合計を、**sp\_iqestdbspaces** の **db\_size\_in\_bytes** パラメータとして使用します。

**sp\_iqestdbspaces** の結果は、インデックスの平均サイズを基にして見積もっただけのものです。実際のサイズは、テーブルに格納されたデータ、特に、データ間の差がどれだけあるかに依存します。

Sybase では、予備の DB 領域セグメントを作成することを強くおすすめしています。使用しなかった場合は、後から削除できます。

例

```
sp_iqestdbspaces 12000000000, 65536, 500000000,
2146304000
```

| dbspace files | Type  | Size       | Msg |
|---------------|-------|------------|-----|
| 1             | min   | 2146304000 |     |
| 2             | min   | 2146304000 |     |
| 3             | min   | 507392000  |     |
| 4             | avg   | 2146304000 |     |
| 5             | max   | 2053697536 |     |
| 6             | spare | 1200001024 |     |

この例では、12GB のデータベースに必要な DB 領域セグメントのサイズと数を見積もります。Sybase IQ では、データ間の差が少ない場合、最大限の圧縮をするためには最低でも 3 つのセグメント (min と表示) を作成することをおすすめします。データの差が標準的であれば、もう 1 つセグメント (avg と表示) を作成します。データの差が大きい (ユニークな値が多く、高度なインデックス化が必要) 場合は、さらに 1 つのセグメント (max と表示) を作成する必要があります。1200001024 バイトの予備のセグメントを作成すれば、最初のロードを必ず成功させることができます。データベースをロードした後で、使用していない DB 領域セグメントはすべて削除できます。

## sp\_iqestspace プロシージャ

|         |                                                                                                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能      | テーブル内のロー数に基づいて、インデックスの作成に必要な領域を見積もります。                                                                                                                                                                 |
| 構文      | <code>sp_iqestspace ( table_name, #_of_rows, iq_page_size )</code>                                                                                                                                     |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。                                                                                                                          |
| 説明      | 基本となるデータベース・テーブルのロー数と、データベースの IQ ページ・サイズに基づいて、データベースに必要な領域の見積もりを表示します。このプロシージャでは、指定された IQ ページ・サイズのデフォルト・ブロック・サイズでデータベースが作成されていることを前提として計算します (それ以外の場合、正確な値が返されません)。表 7-22 に、sp_iqestspace のパラメータを示します。 |

表 7-22 : sp\_iqestspace のパラメータ

| 名前                        | データ型      | 説明                                                                           |
|---------------------------|-----------|------------------------------------------------------------------------------|
| <code>table_name</code>   | char(256) | テーブルの名前。                                                                     |
| <code>#_of_rows</code>    | int       | テーブルにあるロー数。                                                                  |
| <code>iq_page_size</code> | smallint  | データベースの IQ セグメント用に定義されているページ・サイズ。2 の累乗で、範囲は 65536 ~ 524288、デフォルトは 131072 です。 |

## sp\_iqevent プロシージャ

|    |                                    |
|----|------------------------------------|
| 機能 | システム・イベントおよびユーザ定義イベントに関する情報を表示します。 |
|----|------------------------------------|

|         |                                                                                                                                                                                                                                                                                                                                                                        |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文      | <b>sp_iqevent</b> [ <i>event-name</i> ], [ <i>event-owner</i> ], [ <i>event-type</i> ]                                                                                                                                                                                                                                                                                 |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。                                                                                                                                                                                                                                                                                          |
| 使用法     | <p><b>event-name</b> イベント名。</p> <p><b>event-owner</b> イベントの所有者。</p> <p><b>event-type</b> イベントのタイプ。指定できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>SYSTEM</b> : システム・イベント (ユーザ <b>SYS</b> または <b>dbo</b> によって所有されているイベント) に関する情報のみを表示します。</li> <li>• <b>ALL</b> : ユーザ・イベントおよびシステム・イベントに関する情報を表示します。</li> <li>• その他の値 : ユーザ・イベントに関する情報を表示します。</li> </ul> |

**sp\_iqevent** プロシージャは、パラメータなしで呼び出せます。パラメータを指定しない場合、ユーザ・イベント (**dbo** または **SYS** によって所有されていないイベント) に関する情報のみがデフォルトで表示されます。

最初の 2 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に **NULL** を指定する必要があります。たとえば、**sp\_iqevent NULL, NULL, SYSTEM** や **sp\_iqevent NULL, user1** のように指定します。

**表 7-23 : sp\_iqevent の使用例**

| 構文                                   | 出力                                                                                                                        |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>sp_iqevent</b>                    | データベース内のすべてのユーザ・イベントに関する情報を表示します。                                                                                         |
| <b>sp_iqevent e1</b>                 | イベント <b>e1</b> に関する情報を表示します。                                                                                              |
| <b>sp_iqevent non_existing_event</b> | イベント <b>non_existing_event</b> が存在しないため、ローは返されません。                                                                        |
| <b>sp_iqevent NULL, DBA</b>          | <b>DBA</b> によって所有されているすべてのイベントに関する情報を表示します。                                                                               |
| <b>sp_iqevent e1, DBA</b>            | <b>DBA</b> によって所有されているイベント <b>e1</b> に関する情報を表示します。                                                                        |
| <b>sp_iqevent ev_iqbegintxn</b>      | <b>ev_iqbegintxn</b> は、システム定義のイベントです。 <b>ev_iqbegintxn</b> という名前のユーザ定義イベントが存在しない場合、ローは返されません (デフォルトでは、ユーザ定義イベントのみが返されます)。 |
| <b>sp_iqevent ev_iqbegintxn, dbo</b> | イベント <b>ev_iqbegintxn</b> はユーザ・イベントではないため、ローは返されません (デフォルトでは、ユーザ・イベントのみが返されます)。                                           |

| 構文                                                  | 出力                                                                              |
|-----------------------------------------------------|---------------------------------------------------------------------------------|
| <code>sp_iqevent NULL, NULL, SYSTEM</code>          | ( <code>dbo</code> または <code>SYS</code> によって所有されている) すべてのシステム・イベントに関する情報を表示します。 |
| <code>sp_iqevent ev_iqbegintxn, NULL, SYSTEM</code> | システム・イベント <code>ev_iqbegintxn</code> に関する情報を表示します。                              |
| <code>sp_iqevent ev_iqbegintxn, dbo, ALL</code>     | <code>dbo</code> によって所有されているシステム・イベント <code>ev_iqbegintxn</code> に関する情報を表示します。  |

## 説明

`sp_iqevent` ストアド・イベントは、データベース内のイベントに関する情報を表示します。1 つ以上のパラメータを指定した場合、指定したパラメータにより結果がフィルタされます。たとえば、`event-name` を指定した場合、指定のイベントに関する情報のみが表示されます。`event-owner` を指定した場合、`sp_iqevent` は指定の所有者によって所有されているイベントに関する情報のみを返します。パラメータを指定しない場合、`sp_iqevent` はデータベース内のすべてのユーザ・イベントに関する情報を表示します。

`sp_iqevent` プロシージャは、次のカラムに情報を返します。

表 7-24 : `sp_iqevent` のカラム

| カラム名                     | 説明                                                                                                                                                 |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>event_name</code>  | イベント名                                                                                                                                              |
| <code>event_owner</code> | イベントの所有者                                                                                                                                           |
| <code>event_type</code>  | システム・イベントの場合、 <code>SYSEVENTTYPE</code> システム・テーブルに挙げられたイベント・タイプ                                                                                    |
| <code>enabled</code>     | イベントの発生が許可されているかどうかを示す (Y/N)                                                                                                                       |
| <code>action</code>      | イベント・ハンドラの定義                                                                                                                                       |
| <code>condition</code>   | イベント・ハンドラの起動を制御するのに使用する <code>WHERE</code> 条件                                                                                                      |
| <code>location</code>    | イベントの発生が許可されている場所。 <ul style="list-style-type: none"> <li>• C = 統合 (consolidated)</li> <li>• R = リモート (remote)</li> <li>• A = すべて (all)</li> </ul> |
| <code>remarks</code>     | コメント文字列                                                                                                                                            |

## 例

ユーザ定義イベント `e1` に関する情報を表示します。

```
sp_iqevent e1
```

```
event_name event_owner event_type enabled action
e1 DBA (NULL) Y (NULL)
```

|           |          |         |
|-----------|----------|---------|
| condition | location | remarks |
| (NULL)    | A        | (NULL)  |

すべてのシステム・イベントに関する情報を表示します。

sp\_iqevent NULL, NULL, SYSTEM

|                 |             |                |         |                               |
|-----------------|-------------|----------------|---------|-------------------------------|
| event_name      | event_owner | event_type     | enabled | action                        |
| ev_iqbegintxn   | dbo         | IQTLVAvailable | Y       | begin call<br>dbo.sp_iqlog... |
| ev_iqmpxcompact | dbo         | (NULL)         | N       | begin Declare<br>_Catalog...  |

|           |          |         |
|-----------|----------|---------|
| condition | location | remarks |
| (NULL)    | A        | (NULL)  |
| (NULL)    | A        | (NULL)  |

## 参照

『リファレンス：文とオプション』の「[CREATE EVENT 文](#)」

『システム管理ガイド 第2巻』の「[第6章 スケジューリングとイベント処理によるタスクの自動化](#)」

## sp\_iqfile プロシージャ

**機能** DB 領域の各 dbfile についての詳細情報を表示します。

**構文** `sp_iqfile [ dbspace-name ]`

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**説明** sp\_iqfile は、DB 領域の各 dbfile のデータの利用率、プロパティ、およびタイプを表示します。この情報を使用して、データの移動が必要かどうかを判断できます。また、移動されたデータに関しては、古いバージョンの割り付けが解除されているかどうかを確認できます。

sp\_iqfile は次の情報を表示します。

**表 7-25 : sp\_iqfile のカラム**

| カラム名        | 説明                                                                                       |
|-------------|------------------------------------------------------------------------------------------|
| DBSpaceName | CREATE DBSPACE 文で指定された DB 領域の名前。CASE RESPECT を指定して作成されたデータベースの場合、DB 領域名は大文字と小文字は区別されません。 |
| DBFileName  | 論理ファイル名。                                                                                 |



| カラム名        | 説明                                                                 |
|-------------|--------------------------------------------------------------------|
| Path        | 物理ファイルまたはロー・パーティションの場所。                                            |
| SegmentType | DB 領域のタイプ (MAIN または TEMPORARY)。                                    |
| RWMode      | DB 領域のモード。読み込み/書き込み (RW) または読み取り専用 (RO)。                           |
| Online      | T (オンライン) または F (オフライン)。                                           |
| Usage       | DB 領域のこのファイルで現在使用されている DB 領域の割合。                                   |
| DBFileSize  | ファイルまたはロー・パーティションの現在のサイズ。ロー・パーティションでは、このサイズ値は物理サイズよりも小さくなる場合があります。 |
| Reserve     | DB 領域のこのファイルに追加できる予約領域。                                            |
| StripeSize  | ディスク・ストライピングが有効になっている場合、次のファイルに移動するまでにファイルに書き込まれたデータの量。            |
| BlkTypes    | ユーザ・データと内部システム構造が使用している領域 (識別子の値については、表 7-26 を参照)。                 |
| FirstBlk    | ファイルに割り当てられている最初の IQ ブロック番号。                                       |
| LastBlk     | ファイルに割り当てられている最後の IQ ブロック番号。                                       |
| OkToDrop    | ファイルを削除できる場合は 'Y'、それ以外の場合は 'N'。                                    |

表 7-26 に、ブロック・タイプ識別子の値を示します。

表 7-26 : sp\_iqfile のブロック・タイプ

| 識別子 | ブロック・タイプ            |
|-----|---------------------|
| A   | アクティブなバージョン         |
| B   | バックアップ構造            |
| C   | チェックポイント・ログ         |
| D   | データベースの識別情報         |
| F   | フリー・リスト             |
| G   | グローバル・フリー・リスト・マネージャ |
| H   | フリー・リストのヘッダ・ブロック    |
| I   | インデックス・アドバイスの格納     |
| M   | マルチプレックス CM*        |
| O   | 旧バージョン              |
| T   | テーブルの使用             |
| U   | インデックスの使用           |

| 識別子 | ブロック・タイプ     |
|-----|--------------|
| N   | カラムの使用       |
| X   | チェックポイントでの削除 |

\*マルチプレックス・コミット ID ブロック (実際は 128 ブロック) は、シンプレックス・データベースで使用されていない場合でも、すべての IQ データベースに存在します。

例 DB 領域のファイルに関する情報を表示します。

```
sp_iqfile;
```

| DBSpace<br>Name | DBFile<br>Name | Path                             | Segment<br>Type | RWMode |
|-----------------|----------------|----------------------------------|-----------------|--------|
| IQ_SYSTEM_MAIN  | IQ_SYSTEM_MAIN | /sunopt/users/user1/iqdemo.iq    | MAIN            | RW     |
| IQ_SYSTEM_TEMP  | IQ_SYSTEM_TEMP | /sunopt/users/user1/iqdemo.iqtmp | TEMPORARY       | RW     |

  

| Online | Usage | DBFileSize | Reserve | Stripesize | BlkTypes              | FirstBlk | LastBlk | OkToDrop |
|--------|-------|------------|---------|------------|-----------------------|----------|---------|----------|
| T      | 21    | 300M       | 50M     | 8K         | 1H,7648F,<br>32D,128M | 1        | 38400   | N        |
| T      | 1     | 100M       | 50M     | 8K         | 1H,64F,16A            | 1        | 12800   | N        |

参照 [「sp\\_iqdbspaceinfo プロシージャ」 \(389 ページ\)](#) および [「sp\\_iqindexinfo プロシージャ」 \(421 ページ\)](#)

『Sybase IQ システム管理ガイド』の「[第 5 章 データベース・オブジェクトの使用](#)」

## sp\_iqhelp プロシージャ

**機能** システムおよびユーザ定義のオブジェクトおよびデータ型に関する情報を表示します。

**構文** `sp_iqhelp [ obj-name ], [ obj-owner ], [ obj-category ], [ obj-type ]`

**パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

**使用法**

- obj-name** オブジェクト名。
- obj-owner** オブジェクトの所有者。
- obj-category** オブジェクトのカテゴリを指定するオプション・パラメータ。

表 7-27 : `sp_iqhelp obj-category` のパラメータ値

| object-type パラメータ | 指定される内容                     |
|-------------------|-----------------------------|
| "table"           | オブジェクトはベース・テーブルです。          |
| "view"            | オブジェクトはビューです。               |
| "procedure"       | オブジェクトはストアド・プロシージャまたは関数です。  |
| "event"           | オブジェクトはイベントです。              |
| "datatype"        | オブジェクトはシステムまたはユーザ定義のデータ型です。 |

カラム、制約、およびインデックスは、テーブルに関連付けられており、直接クエリできません。テーブルをクエリすると、そのテーブルに関連するカラム、インデックス、および制約に関する情報が表示されます。

指定したオブジェクト・カテゴリが許可された値ではない場合、“Invalid object category” エラーが返されます。

**obj-type** オブジェクトのタイプ。指定できる値は次のとおりです。

- **SYSTEM** : システム・オブジェクト (ユーザ **SYS** または **dbo** によって所有されているオブジェクト) に関する情報のみを表示します。
- **ALL** : すべてのオブジェクトに関する情報を表示します。

デフォルトでは、システム・オブジェクト以外のオブジェクトに関する情報のみが表示されます。指定したオブジェクト・タイプが **SYSTEM** または **ALL** ではない場合、“Invalid object type” エラーが返されます。

`sp_iqhelp` プロシージャは、パラメータなしで呼び出せます。パラメータを指定しない場合、`sp_iqhelp` はデータベース内のすべての独立したオブジェクト、つまりベース・テーブル、ビュー、ストアド・プロシージャ、関数、イベント、およびデータ型に関する情報を表示します。

最初の 3 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に **NULL** を指定する必要があります。たとえば `sp_iqhelp NULL, NULL, NULL, SYSTEM` や `sp_iqhelp NULL, user1, "table"` のように指定します。

`obj-category` パラメータは、**NULL** 以外の場合、一重引用符または二重引用符で囲みます。

`sp_iqhelp` で指定の記述を満たすオブジェクトがデータベースに見つからない場合、エラー “No object found for the given description” が返されます。

表 7-28 : sp\_iqhelp の使用例

| 構文                                 | 出力                                                                                           |
|------------------------------------|----------------------------------------------------------------------------------------------|
| sp_iqhelp                          | データベース内のすべてのユーザ定義テーブル、ビュー、プロシージャ、イベント、およびデータ型に関する要約情報を表示します。                                 |
| sp_iqhelp t1, u1, "table"          | ユーザ <b>u1</b> の所有するテーブル <b>t1</b> と、 <b>t1</b> に関連するカラム、インデックス、および制約に関する情報を表示します。            |
| sp_iqhelp NULL, u1, "view"         | ユーザ <b>u1</b> の所有するビュー <b>v1</b> と、 <b>v1</b> に関連するカラムに関する情報を表示します。                          |
| sp_iqhelp sp2                      | プロシージャ <b>sp2</b> と <b>sp2</b> のパラメータに関する情報を表示します。                                           |
| sp_iqhelp e1                       | イベント <b>e1</b> に関する情報を表示します。                                                                 |
| sp_iqhelp dt1                      | データ型 <b>dt1</b> に関する情報を表示します。                                                                |
| sp_iqhelp NULL, NULL, NULL, SYSTEM | ( <b>dbo</b> または <b>SYS</b> によって所有されている) すべてのシステム・オブジェクトに関する要約情報を表示します。                      |
| sp_iqhelp non_existing_obj         | <b>non_existing_obj</b> が存在しないため、エラー "Object 'non_existing_obj' not found" が返されます。           |
| sp_iqhelp NULL, non_existing_user  | <b>non_existing_user</b> が存在しないため、エラー "User 'non_existing_user' not found" が返されます。           |
| sp_iqhelp t1, NULL, "apple"        | <b>obj-category</b> に対して "apple" は許可された値ではないため、エラー "Invalid object category 'apple'" が返されます。 |
| sp_iqhelp t1, NULL, NULL, "USER"   | <b>obj-type</b> に対して "USER" は許可された値ではないため、エラー "Invalid object type 'USER'" が返されます。           |

## 説明

sp\_iqhelp ストアド・プロシージャは、IQ データベース内のシステム、ユーザ定義オブジェクト、およびユーザ定義データ型に関する情報を表示します。sp\_iqhelp でサポートされるオブジェクトは、テーブル、ビュー、カラム、インデックス、ジョイン・インデックス、制約、ストアド・プロシージャ、関数、イベント、およびデータ型です。

1 つ以上のパラメータを指定した場合、指定したパラメータにより結果がフィルタされます。たとえば、**obj-name** を指定した場合、指定のオブジェクトに関する情報のみが表示されます。**obj-owner** を指定した場合、sp\_iqhelp は指定の所有者によって所有されているオブジェクトに関する情報を返します。パラメータを指定しない場合、sp\_iqhelp はデータベース内のすべてのユーザ定義テーブル、ビュー、プロシージャ、イベント、およびデータ型に関する要約情報を表示します。

`sp_iqhelp` プロシージャは、指定のパラメータが複数のオブジェクトまたは1つのオブジェクトのいずれに一致するかどうかで、要約情報または詳細情報のいずれかを返します。`sp_iqhelp` の出力カラムは、ストアド・プロシージャ `sp_iqtable`、`sp_iqindex`、`sp_iqview`、および `sp_iqconstraint` により表示されるカラムと同様のものです。

複数のオブジェクトが指定の `sp_iqhelp` パラメータに一致する場合、`sp_iqhelp` はそれらのオブジェクトに関する要約情報を表示します。

**表 7-29 : `sp_iqhelp` 要約情報**

| オブジェクト・タイプ       | 表示されるカラム                                                                                                                                                                                                                           |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ベース・テーブル         | <code>table_name</code> 、 <code>table_owner</code> 、 <code>server_type</code> 、 <code>location</code> 、 <code>table_constraints</code> 、 <code>remarks</code>                                                                      |
| ビュー              | <code>view_name</code> 、 <code>view_creator</code> 、 <code>view_def</code> 、 <code>server_type</code> 、 <code>location</code> 、 <code>remarks</code>                                                                               |
| ストアド・プロシージャ      | <code>proc_name</code> 、 <code>proc_creator</code> 、 <code>proc_defn</code> 、 <code>replicate</code> 、 <code>srvld</code> 、 <code>remarks</code>                                                                                   |
| 関数               | <code>proc_name</code> 、 <code>proc_creator</code> 、 <code>proc_defn</code> 、 <code>replicate</code> 、 <code>remarks</code>                                                                                                        |
| イベント             | <code>event_name</code> 、 <code>event_creator</code> 、 <code>enabled</code> 、 <code>location</code> 、 <code>event_type</code> 、 <code>action</code> 、 <code>external_action</code> 、 <code>condition</code> 、 <code>remarks</code> |
| システムおよびユーザ定義データ型 | <code>type_name</code> 、 <code>creator</code> 、 <code>nulls</code> 、 <code>width</code> 、 <code>scale</code> 、 <code>default</code> 、 <code>check</code>                                                                           |

1つのオブジェクトが指定の `sp_iqhelp` パラメータに一致する場合、`sp_iqhelp` はオブジェクトに関する詳細情報を表示します。

表 7-30 : sp\_iqhelp 詳細情報

| オブジェクト・タイプ  | 説明                                                                                 | コラム                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| テーブル        | 指定のベース・テーブルとそのコラム、インデックス、制約、およびジョイン・インデックス (テーブルがジョイン・インデックスに関与する場合) に関する情報を表示します。 | <ul style="list-style-type: none"> <li>• テーブル・コラム : table_name、table_owner、server_type、location、table_constraints、remarks</li> <li>• コラムのコラム : column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks</li> <li>• インデックス・コラム : index_name、column_name、index_type、unique_index、location、remarks</li> <li>• 制約コラム : constraint_name (役割)、column_name、index_name、constraint_type、foreigntable_name、foreigntable_owner、foreigncolumn_name、foreignindex_name、location</li> <li>• ジョイン・インデックス・コラム : joinindex_name、creator、left_table_name、left_table_owner、left_column_name、join_type、right_table_name、right_table_owner、right_column_name、key_type、valid、remarks</li> </ul> |
| ビュー         | 指定のビューとそのコラムに関する情報を表示します。                                                          | <ul style="list-style-type: none"> <li>• ビュー・コラム : view_name、view_creator、view_def、server_type、location、remarks</li> <li>• コラムのコラム : column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ストアド・プロシージャ | 指定のプロシージャとそのパラメータに関する情報を表示します。                                                     | <ul style="list-style-type: none"> <li>• プロシージャ・コラム : proc_name、proc_creator、proc_defn、replicate、srvid、remarks</li> <li>• パラメータ・コラム : parameter_name、type、width、scale、default、mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 関数          | 指定の関数とそのパラメータに関する情報を表示します。                                                         | <ul style="list-style-type: none"> <li>• 関数コラム : proc_name、proc_creator、proc_defn、replicate、srvid、remarks</li> <li>• パラメータ・コラム : parameter_name、type、width、scale、default、mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| イベント        | 指定のイベントに関する情報を表示します。                                                               | <ul style="list-style-type: none"> <li>• イベント・コラム : event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| データ型        | 指定のデータ型に関する情報を表示します。                                                               | <ul style="list-style-type: none"> <li>• データ型のコラム : type_name、creator、nulls、width、scale、default、check</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

表 7-30 の各出力カラムについては、次のストアド・プロシージャの説明を参照してください。

- テーブル : 「[sp\\_iqtable プロシージャ](#)」 (471 ページ)
- カラム : 「[sp\\_iqcolumn プロシージャ](#)」 (369 ページ)
- インデックス : 「[sp\\_iqindex](#) および [sp\\_iqindex\\_alt プロシージャ](#)」 (415 ページ)
- 制約 : 「[sp\\_iqconstraint プロシージャ](#)」 (376 ページ)
- ジョイン・インデックス : 「[sp\\_iqjoinindex プロシージャ](#)」 (427 ページ)
- ビュー : 「[sp\\_iqview プロシージャ](#)」 (484 ページ) および Adaptive Server Enterprise カタログ・プロシージャ [sp\\_columns](#) (ビュー・カラムの場合)
- ストアド・プロシージャと関数 : 「[sp\\_iqprocedure プロシージャ](#)」 (442 ページ) および 「[sp\\_iqprocparm プロシージャ](#)」 (445 ページ) (プロシージャ・パラメータの場合)
- イベント : 「[sp\\_iqevent プロシージャ](#)」 (403 ページ)
- データ型 : 「[sp\\_iqdatatype プロシージャ](#)」 (382 ページ)

**Adaptive Server Enterprise の互換性** Sybase IQ [sp\\_iqhelp](#) ストアド・プロシージャは、Adaptive Server Enterprise [sp\\_help](#) プロシージャと同様のもので、**SYSOBJECTS** システム・テーブルに挙げられたデータベース・オブジェクトと、システムおよびユーザ定義のデータ型に関する情報を表示します。

サポートされるオブジェクト型とオブジェクトのネームスペースについて、Sybase IQ のアーキテクチャは Adaptive Server とは異なります。Adaptive Server では、すべてのオブジェクト (テーブル、ビュー、ストアド・プロシージャ、ログ、規則、デフォルト、トリガ、検査制約、参照制約、およびテンポラリ・オブジェクト) は、**SYSOBJECTS** システム・テーブルに保存され、同じネームスペースに属します。Sybase IQ によりサポートされるオブジェクト (テーブル、ビュー、ストアド・プロシージャ、イベント、プライマリ・キー、一意性制約、検査制約、および参照制約) は、異なるシステム・テーブルに保存され、異なるネームスペースに属します。たとえば、Sybase IQ ではテーブルの名前がイベントまたはストアド・プロシージャの名前と同じであっても構いません。

Sybase IQ と Adaptive Server のアーキテクチャの相違のために、Sybase IQ `sp_iqhelp` のサポートするオブジェクト・タイプと構文は、Adaptive Server `sp_help` のサポートするオブジェクトと構文とは異なります。ただし、両ストアド・プロシージャにより表示されるデータベース・オブジェクトに関する情報の種類は同様のものです。

例 テーブル `sale` に関する詳細情報を表示します。

```
sp_iqhelp sale
```

```
Table_name Table_owner Server_type Location dbspace_id isPartitioned
table_constraints
```

```
=====
sale DBA IQ Main 16387 N
```

```
Remarks table_constraints
=====
```

```
(NULL) (NULL)
```

```
column_name domain_name width scale nulls default cardinality
=====
```

```
prod_id integer 4 0 Y (NULL) 0
month_num integer 4 0 Y (NULL) 0
rep_id integer 4 0 Y (NULL) 0
sales integer 4 0 Y (NULL) 0
```

```
est_cardinality isPartitioned remarks check
=====
```

```
0 N (NULL) (NULL)
0 N (NULL) (NULL)
0 N (NULL) (NULL)
0 N (NULL) (NULL)
```

```
index_name column_name index_type unique_index location
=====
```

```
ASIQ_IDX_T463_C2_FP month_num FP N Main
ASIQ_IDX_T463_C1_FP prod_id FP N Main
ASIQ_IDX_T463_C3_FP rep_id FP N Main
ASIQ_IDX_T463_C4_FP sales FP N Main
```

```
remarks
=====
(NULL)
(NULL)
```



(NULL)

(NULL)

プロシージャ `sp_customer_list` に関する詳細情報を表示します。

```

sp_iqhelp sp_customer_list
proc_name proc_owner proc_defn
=====
sp_customer_list DBA create procedure DBA.sp_customer_list()
 result(id integer company_name char(35))
 begin
 select id company_name from Customers
 end

 replicate srvid remarks
 =====
 N (NULL) (NULL)

parm_name parm_type parm_mode domain_name width scale
=====
id result out integer 4 0
company_name result out char 35 0

default
=====
(NULL)

```

## sp\_iqindex および sp\_iqindex\_alt プロシージャ

機能

インデックスに関する情報をリストします。

構文 1

```
sp_iqindex ([table_name], [column_name], [table_owner])
```

構文 2

```
sp_iqindex [table_name='tablename'],
[column_name='columnname'], [table_owner='tableowner']
```

構文 3

```
sp_iqindex_alt ([table_name], [column_name], [table_owner])
```

構文 4

```
sp_iqindex_alt [table_name='tablename'],
[column_name='columnname'], [table_owner='tableowner']
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

## 使用法

**構文 1** 最初の 2 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に NULL を指定する必要があります。たとえば、`sp_iqindex NULL,NULL,DBA` や `sp_iqindex Departments,NULL,DBA` のようになります。

**構文 2** パラメータは、どのような順序で指定しても構いません。一重引用符で囲みます。

**構文 3 および 4** 複数カラムのインデックスがある場合、出力は多少異なります。構文 1 および 2 と同じオプションを使用できます。

## 説明

データベース内のインデックスに関する情報を表示します。いずれかのパラメータを指定すると、そのテーブル、カラム、または指定したユーザが所有するテーブルからのみインデックスが返されます。複数のパラメータを指定すると、指定されたすべてのパラメータにより結果がフィルタされます。パラメータを指定しない場合、データベース内のすべてのテーブルのすべてのインデックスが返されます。

**表 7-31 : sp\_iqindex および sp\_iqindex\_alt のカラム**

| カラム名         | 説明                                                  |
|--------------|-----------------------------------------------------|
| table_name   | テーブルの名前                                             |
| table_owner  | テーブルの所有者                                            |
| column_name  | カラムの名前。複数カラムのインデックスでは複数の名前が表示されます                   |
| index_type   | インデックスのタイプの省略形 (HG、LF など)                           |
| index_name   | インデックス名                                             |
| unique_index | ユニーク・インデックスの場合は 'U'、そうでない場合は 'N'                    |
| location     | TEMP = IQ テンポラリ・ストア、MAIN = IQ ストア、SYSTEM = カタログ・ストア |
| remarks      | COMMENT 文で追加されたユーザ・コメント                             |

`sp_iqindex` は常に、インデックスごとに 1 行生成します。`sp_iqindex_alt` は、複数カラムのインデックスがある場合、カラムのインデックスごとに 1 行生成します。

## 例

次の構文ではいずれも、`DepartmentID` という名前のカラムのすべてのインデックスが返されます。

```
call sp_iqindex (NULL,'DepartmentID')
sp_iqindex column_name='DepartmentID'
```

| table_name  | table_owner | column_name     | index_type | index_name          | unique_index | location | dbspace_id | remarks |
|-------------|-------------|-----------------|------------|---------------------|--------------|----------|------------|---------|
| Departments | GROUPO      | DepartmententID | FP         | ASIQ_IDX_T201_C1_FP | N            | Main     | 16387      | (NULL)  |
| Departments | GROUPO      | DepartmententID | HG         | ASIQ_IDX_T201_C1_HG | U            | Main     | 16387      | (NULL)  |
| Employees   | GROUPO      | DepartmententID | FP         | ASIQ_IDX_T202_C5_FP | N            | Main     | 16387      | (NULL)  |

次の構文ではいずれも、GROUPO が所有しているテーブル Departments のすべてのインデックスが返されます。

```
sp_iqindex Departments,NULL,GROUPO
sp_iqindex table_name='Departments',table_owner='DBA'
```

| table_name  | table_owner | column_name      | index_type | index_name          | unique_index | location | dbspace_id | remarks |
|-------------|-------------|------------------|------------|---------------------|--------------|----------|------------|---------|
| Departments | GROUPO      | DepartmentHeadID | FP         | ASIQ_IDX_T201_C3_FP | N            | Main     | 16387      | (NULL)  |
| Departments | GROUPO      | DepartmentID     | FP         | ASIQ_IDX_T201_C1_FP | N            | Main     | 16387      | (NULL)  |
| Departments | GROUPO      | DepartmentID     | HG         | ASIQ_IDX_T201_C1_HG | U            | Main     | 16387      | (NULL)  |
| Departments | GROUPO      | DepartmentName   | FP         | ASIQ_IDX_T201_C2_FP | N            | Main     | 16387      | (NULL)  |

sp\_iqindex\_alt の次の構文ではいずれも、カラム City を含む Employees テーブルのインデックスが返されます。インデックス emp\_loc は、カラム City および State の複数カラムのインデックスです。sp\_iqindex\_alt では、複数カラムのインデックスではカラムごとに 1 行が表示されます。

```
sp_iqindex_alt Employees,City
sp_iqindex_alt table_name='Employees',
column_name='City'
```

| table_name | table_owner | column_name | index_type | index_name          | unique_index | dbspace_id | remarks |
|------------|-------------|-------------|------------|---------------------|--------------|------------|---------|
| Employees  | GROUPO      | City        | FP         | ASIQ_IDX_T452_C7_FP | N            | 16387      | (NULL)  |
| Employees  | GROUPO      | City        | HG         | emp_loc             | N            | 16387      | (NULL)  |
| Employees  | GROUPO      | State       | HG         | emp_loc             | N            | 16387      | (NULL)  |

sp\_iqindex の出力は、同じテーブル、同じカラムでも多少異なります。

```
sp_iqindex Employees, City
sp_iqindex table_name='Employee', column_name='City'
```

| table_ name | table_ owner | column_ name | index_ type | index_ name          | unique_ index | dbspace_ id | location | remarks |
|-------------|--------------|--------------|-------------|----------------------|---------------|-------------|----------|---------|
| Employees   | GROU<br>PO   | City         | FP          | ASIQ_IDX_T452_C7_ FP | N             | 16387       | Main     | (NULL)  |
| Employees   | GROU<br>PO   | City, State  | HG          | emp_loc              | N             | 16387       | Main     | (NULL)  |

#### 参照

『リファレンス：文とオプション』の「第2章 データベース・オプション」の「FP\_LOOKUP\_SIZE オプション」、「INDEX\_ADVISOR オプション」、および「MINIMIZE\_STORAGE オプション」

『システム管理ガイド 第1巻』の「第6章 Sybase IQ インデックスの使用」

## sp\_iqindexadvice プロシージャ

#### 機能

格納されたインデックス・アドバイス・メッセージを表示します。オプションで、アドバイス記憶領域をクリアします。

#### 構文

```
sp_iqindexadvice ([resetflag])
```

#### パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

#### 使用法

**resetflag** 呼び出し側からインデックス・アドバイス記憶領域をクリアできます。*resetflag* が 0 以外である場合、最後のローの取得後に、すべてのアドバイスが削除されます。

#### 説明

このプロシージャを使用すると、蓄積されたインデックス・アドバイザのメッセージを SQL でクエリできます。情報は、どのインデックスまたはスキーマの変更が多くクエリに影響を与えるかを判断するために役立てることができます。

表 7-32 に INDEX\_ADVISOR のカラムを示します。

表 7-32 : sp\_iqindexadvice のカラム

| カラム名   | 説明               |
|--------|------------------|
| Advice | ユニークなアドバイス・メッセージ |
| NInst  | メッセージのインスタンス数    |
| LastDT | アドバイスが生成された最終日時  |

例 表 7-33 に、`sp_iqindexadvice` プロシージャの出力例を示します。

表 7-33 : `sp_iqindexadvice` の出力例

| Advice                                                                 | NInst | LastDT                  |
|------------------------------------------------------------------------|-------|-------------------------|
| Add a CMP index on DBA.tb (c2, c3)<br>Predicate: (tb.c2 = tb.c3)       | 2073  | 2009-04-07 16:37:31.000 |
| Convert HG index on DBA.tb.c4 to a unique<br>HG                        | 812   | 2009-04-06 10:01:15.000 |
| Join Key Columns DBA.ta.c1 and DBA.tb.c1<br>have mismatched data types | 911   | 2009-02-25 20:59:01.000 |

参照

「`sp_iqcolumnuse` プロシージャ」(371 ページ)、 「`sp_iqdbspace` プロシージャ」(387 ページ)、 「`sp_iqindexuse` プロシージャ」(426 ページ)、 「`sp_iqtableuse` プロシージャ」(475 ページ)、 「`sp_iqunusedcolumn` プロシージャ」(479 ページ)、 「`sp_iqunusedindex` プロシージャ」(481 ページ)、 「`sp_iqunusedtable` プロシージャ」(482 ページ)、 および 「`sp_iqworkmon` プロシージャ」(489 ページ)

『リファレンス：文とオプション』の「第2章 データベース・オプション」の「`FP_LOOKUP_SIZE` オプション」、 「`INDEX_ADVISOR` オプション」、 および 「`MINIMIZE_STORAGE` オプション」

『システム管理ガイド 第1巻』の「第6章 Sybase IQ インデックスの使用」

## sp\_iqindexfragmentation プロシージャ

機能

Sybase IQ インデックスの B-tree、garray、および bitmap 構造内でページ領域が占める割合に関する情報をレポートします。

garray の場合、フィル・パーセンテージの計算では、`GARRAY_FILL_FACTOR_PERCENT` オプションによって制御される garray グループ内の予約領域が考慮されません。

構文

```
dbo.sp_iqindexfragmentation ('target')
```

```
target: table table-name | index index-name [...]
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、`EXECUTE` パーミッションが付与される必要があります。

使用法

**table-name** target の table *table-name* では、指定されたテーブル内の、デフォルト以外のすべてのインデックスについてレポートされます。

**index-name** target の index *index-name* では、指定されたインデックスについてレポートされます。各 **index-name** は修飾されたインデックス名です。テーブル内の複数のインデックスを指定することもできますが、指定するインデックスごとに **index** キーワードを繰り返し指定する必要があります。

例 Customers テーブル内のユニークでない HG インデックスである cidhg に関して、内部インデックスの断片化がレポートされます。

```
dbo.sp_iqindexfragmentation ('index customers.cidhg')
```

| インデックス              | インデックス・タイプ | btree ノード・ページ | GARRAY_FILL_FACTOR_PERCENT |
|---------------------|------------|---------------|----------------------------|
| dba.customers.cidhg | HG         | 3             | 75                         |
| SQLCODE             | 0          |               |                            |
|                     |            |               |                            |
| フィル・パーセント           | btree ページ  | garray ページ    | bitmap ページ                 |
| 0 - 10%             | 0          | 0             | 0                          |
| 11 - 20%            | 0          | 0             | 0                          |
| 21 - 30%            | 0          | 0             | 0                          |
| 31 - 40%            | 0          | 0             | 22                         |
| 41 - 50%            | 0          | 0             | 0                          |
| 51 - 60%            | 0          | 0             | 10                         |
| 61 - 70%            | 2          | 0             | 120                        |
| 71 - 80%            | 138        | 3             | 64                         |
| 81 - 90%            | 24         | 122           | 14                         |
| 91 - 100%           | 18         | 1             | 0                          |

この出力では、ユニークでない HG インデックスである cidhg 内の 182 B-tree ページのうち、2 ページが 61% ~ 70%、138 ページが 71% ~ 80%、24 ページが 81% ~ 90%、18 ページが 91% ~ 100% 使用されていることがわかります。garray および bitmap ページの利用率も、同様にレポートされます。すべてのパーセンテージは、一番近いパーセンテージ・ポイントにトランケートされます。HG インデックスでは、GARRAY\_FILL\_FACTOR\_PERCENT オプションの値も表示されます。B-tree を使用するインデックス・タイプでは、ノード (リーフでない) ページの数も表示されます。HG、LF、WD、DATE、および DTTM がこれに当たります。

このインデックスでのストアド・プロシージャの実行中にエラーが発生した場合、SQLCODE は 0 以外の値を取ります。

- 参照 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第2章 データベース・オプション」の「GARRAY\_FILL\_FACTOR\_PERCENT オプション」および「GARRAY\_PAGE\_SPLIT\_PAD\_PERCENT オプション」
- 『リファレンス：文とオプション』の「第2章 データベース・オプション」の「FP\_LOOKUP\_SIZE オプション」、「INDEX\_ADVISOR オプション」、および「MINIMIZE\_STORAGE オプション」
- 『システム管理ガイド 第1巻』の「第6章 Sybase IQ インデックスの使用」

## sp\_iqindexinfo プロシージャ

- 機能 特定のオブジェクトに使用しているブロックの数を、メインの DB 領域のインデックスごとに表示します。オブジェクトが複数の DB 領域に含まれている場合、sp\_iqindexinfo は、すべての DB 領域で使用される領域を返します。次に例を示します。
- 構文
- ```
sp_iqindexinfo { database
| [ table table-name | index index-name ] [...]
| resources resource-percent }
```
- パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。
- 使用法
- データベース全体のインデックス情報を要求することも、任意の数のテーブルやインデックス・パラメータを指定して要求することもできます。テーブル名を指定した場合、sp_iqindexinfo はテーブル内のすべてのインデックスに関する情報を返します。インデックス名を指定した場合、そのインデックスに関する情報のみが返されます。
- ジョイン・インデックスを名前では指定することはできません。ジョイン・インデックスを表示するには、database キーワードを使用します。
- 指定された table-name または index-name があいまいである場合や、オブジェクトが見つからなかった場合は、エラーが返されます。
- マルチプレックス・データベースでは、sp_iqindexinfo は、デフォルトで、セカンダリ・ノード上の共有 IQ ストアに関する情報を表示します。個別のテーブルまたはインデックスが指定された場合、表示するストアは自動的に選択されます。

resource-percent は 0 より大きい整数である必要があります。リソースの割合を使用して CPU の合計使用率を指定すれば、`sp_iqindexinfo` プロシージャによる CPU の使用率を制限できます。

説明

`sp_iqindexinfo` では、特定のオブジェクトがどの DB 領域にあるかを DBA に示します。DBA はこの情報を使用して、`relocate` モードにしてオブジェクトを移動する必要がある DB 領域を判断できます。

`sp_iqindexinfo` の結果は、コマンドを実行しているトランザクションから見たバージョンが基準になります。他のバージョンで使用されているブロックは表示されません。

表 7-34 : `sp_iqindexinfo` のカラム

| カラム名 | 説明 |
|--------------|-------------------------------------------------------------------------|
| Object | テーブル、インデックス、ジョイン・インデックスの名前 |
| Dbspace_name | DB 領域の名前 |
| ObjSize | この DB 領域での、このオブジェクトのデータのサイズ |
| DBSpPct | このオブジェクトが使用している DB 領域の割合 |
| MinBlk | この DB 領域で、このオブジェクトが使用する最初のブロック |
| MaxBlk | この DB 領域で、このオブジェクトが使用する最後のブロック。DB 領域のサイズを縮小する前に移動が必要なオブジェクトを判断するのに役立ちます |

例

テーブル `t2` に関するインデックス情報を表示します。

```
sp_iqindexinfo 'table t2';
```

| Object | dbspace_name | ObjSize | DBSpPct | MinBlk | MaxBlk |
|----------------------------|----------------|---------|---------|---------|---------|
| t2 | IQ_SYSTEM_MAIN | 32K | 1 | 84 | 107 |
| t2 | dbspacedb2 | 160K | 2 | 1045495 | 1045556 |
| t2 | dbspacedb3 | 8K | 1 | 2090930 | 2090930 |
| t2.DBA.ASIQ_IDX_T430_C1_FP | IQ_SYSTEM_MAIN | 136K | 2 | 126 | 321 |
| t2.DBA.ASIQ_IDX_T430_C1_FP | dbspacedb3 | 152K | 2 | 2091032 | 2091053 |
| t2.DBA.t2c1hng | dbspacedb2 | 136K | 2 | 1045537 | 1045553 |

参照

「[sp_iqdbspace プロシージャ](#)」(387 ページ)、[「sp_iqdbspaceinfo プロシージャ」](#) (389 ページ)、[「sp_iqspaceinfo プロシージャ」](#) (458 ページ)

『システム管理ガイド 第 1 巻』の「[第 5 章 データベース・オブジェクトの使用](#)」

『Sybase IQ マルチプレックスの使用』の「[付録 A マルチプレックス・リファレンス](#)」

sp_iqindexmetadata プロシージャ

| | |
|---------|---------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定のインデックスのインデックス・メタデータを表示します。オプションで、指定したテーブル上のインデックスや、指定した所有者に属するインデックスのみに出力を制限できます。 |
| 構文 | <pre>dbo.sp_iqindexmetadata {'index-name' [, 'table-name' [, 'owner-name']]}</pre> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 使用法 | テーブル名を指定すると、そのテーブルに属するインデックスに出力が制限されます。所有者名を指定すると、その所有者が所有するインデックスに出力が制限されます。パラメータを省略した場合のデフォルトは NULL です。プロシージャごとに1つのインデックスのみを指定できます。 |
| 説明 | 出力の最初のローは、インデックスの所有者名、テーブル名、およびインデックス名です。 それ以降出力されるローは、指定したインデックス・タイプに規定されます。 |

表 7-35 : sp_iqindexmetadata の出力ロー

| インデックス・タイプ | 返されるメタデータ |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| CMP、DATE、DTTM、TIME | Type、Version |
| FP | Type、Version、LookupPages、Style、LookupEntries、1stLookupPage、LargeLOBs、SmallLOBs、IQ Unique、LOB Compression (カラムのデータ型が LONG VARCHAR または LONG BINARY の場合のみ) |
| HG | Type、Version、Distinct Keys |
| HNG | Type、Version、BitsPerBlockmap、NumberOfBits |
| LD | Type、Version<ld>、Version、Distinct Keys |
| LF | Type、Version、IndexStatus、NumberOfBlockmaps、BitsPerBlockmap、Distinct Keys |
| WD | Type、Version、KeySize、Delimiters、DelimiterCount、MaxKeywordLength、PermitEmptyWord |

例 次のコマンドは、HG インデックス hg_index_col54 についてのインデックス情報を表示します。

```
sp_iqindexmetadata 'hg_index_col54' , 'metal' , 'DBA';
```

```
'DBA',           'metal'      'hg_index_col54'
'Type',          'HG',        "
'Version',       '2',         "
'Distinct Keys', '0',         "
```

参照

「[sp_iqindex](#) および [sp_iqindex_alt](#) プロシージャ」(415 ページ)、
「[sp_iqindexfragmentation](#) プロシージャ」(419 ページ)、
「[sp_iqindexinfo](#) プロシージャ」(421 ページ)、および「[sp_iqindexsize](#) プロシージャ」
(424 ページ)

『システム管理ガイド 第 1 巻』の「[第 5 章 データベース・オブジェクトの使用](#)」および「[第 6 章 Sybase IQ インデックスの使用](#)」

『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」の「[FP_LOOKUP_SIZE](#) オプション」および「[MINIMIZE_STORAGE](#) オプション」

『システム管理ガイド 第 1 巻』の「[第 6 章 Sybase IQ インデックスの使用](#)」

sp_iqindexsize プロシージャ

機能

指定したインデックスのサイズを返します。

構文

```
sp_iqindexsize [ [ owner.] table.] index_name
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

表 7-36 : sp_iqindexsize のカラム

| カラム名 | 説明 |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Username | インデックスの所有者。 |
| Indexname | 結果を返すインデックス (テーブル名を含む)。 |
| Type | インデックスのタイプ。 |
| Info | KBytes、Pages、Compressed Pages がレポートされる IQ インデックスのコンポーネント。コンポーネントは、インデックスのタイプによって異なります。たとえば、デフォルト (FP) インデックスには、BARRAY (barray) と Bitmap (bm) コンポーネントが含まれます。Low_Fast (LF) インデックスには、B-tree (bt) と Bitmap (bm) コンポーネントが含まれます。 |

| カラム名 | 説明 |
|------------------|-----------------------------------|
| KBytes | 物理オブジェクト・サイズ(キロバイト)。 |
| Pages | メモリ内にオブジェクトを保持するために必要な IQ ページの数。 |
| Compressed Pages | (ディスク上で) オブジェクトを圧縮した場合の IQ ページの数。 |

インデックスの合計サイズ(バイト、キロバイト)と、Info カラムを返します。Info カラムには、KBytes、Pages、Compressed Pages をレポートする IQ インデックスのコンポーネントが返されます。コンポーネントは、インデックスのタイプによって異なります。たとえば、デフォルト(FP)インデックスには、BARRAY (barray) と Bitmap (bm) コンポーネントが含まれます。Low_Fast (LF) インデックスには、B-tree (bt) と Bitmap (bm) コンポーネントが含まれます。

また、メモリにオブジェクトを保持するために必要なページ数と、(ディスク上で)インデックスを圧縮した場合の IQ ページ数も返します。

このプロシージャには、必ず *index_name* パラメータを指定してください。単独のテーブル内のこのインデックス名のみ結果を絞り込むには、インデックスを指定するときに *owner.table* を含めます。

例

```
sp_iqindexsize ASIQ_IDX_T452_C19_FP
```

| Username | Indexname | Type | Info | KBytes | Pages | Compressed Pages |
|----------|--------------------------------|------|----------|--------|-------|------------------|
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | Total | 288 | 4 | 2 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | vdo | 0 | 0 | 0 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | bt | 0 | 0 | 0 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | garray | 0 | 0 | 0 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | bm | 136 | 2 | 1 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | barray | 152 | 2 | 1 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | dpstore | 0 | 0 | 0 |
| DBA | Employees.ASIQ_IDX_T452_C19_FP | FP | largelob | 0 | 0 | 0 |

参照

『システム管理ガイド 第1巻』の「第6章 Sybase IQ インデックスの使用」

『リファレンス：文とオプション』の「第2章 データベース・オプション」の「FP_LOOKUP_SIZE オプション」および「MINIMIZE_STORAGE オプション」

sp_iqindexuse プロシージャ

機能 負荷によってアクセスされる二次的な (非 FP) インデックスの使用状況の情報を詳細にレポートします。

構文 `sp_iqindexuse`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 負荷によってアクセスされる二次的なインデックスごとに、ローが 1 つ表示されます。アクセスされないインデックスは、表示されません。インデックスの使用状況は、オブティマイザ、制約、およびクエリの使用状況別に分割されます。

SYSTEM で作成されたテーブルのインデックスはレポートされません。

表 7-37 : sp_iqindexuse のカラム

| カラム名 | 説明 |
|-------------|--------------------------|
| IndexName | インデックス名 |
| TableName | テーブル名 |
| Owner | インデックス所有者のユーザ名 |
| UID** | インデックスのユニークな識別子 |
| Type | インデックス・タイプ |
| LastDT | 前回のアクセスの日時 |
| NOpt | メタデータ/ユニーク性のアクセスの数 |
| NQry | クエリ・アクセスの数 |
| NConstraint | ユニークまたは参照の整合性チェックのアクセスの数 |

**UID はシステムが割り当てた番号であり、インデックスのインスタンスをユニークに識別します (インスタンスはオブジェクト作成時に定義されます)。

例 sp_iqindexuse プロシージャからの出力例を示します。

| IndexName | TableName | Owner | UID | Type | LastDT | NOpt | NQry | NConstraint |
|----------------|-----------|-------|-----|------|--------------------|------|------|-------------|
| n_nationkey_hg | nation | DBA | 29 | HG | 20070917 22:08:06~ | 12 | 0 | 12 |
| n_regionkey_hg | nation | DBA | 31 | HG | 20070917 22:08:06~ | 12 | 0 | 0 |
| r_regionkey_hg | region | DBA | 47 | HG | 20070917 22:08:06~ | 12 | 0 | 12 |
| s_suppkey_hg | supplier | DBA | 64 | HG | 20070917 22:08:06~ | 12 | 0 | 12 |
| p_partkey_hg | part | DBA | 87 | HG | 20070917 22:08:06~ | 6 | 0 | 6 |
| s_suppkey_hg | supplier | DBA | 64 | HG | 20070917 22:08:06~ | 12 | 0 | 12 |
| ... | | | | | | | | |

参照 『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「負荷のモニタリング」

「[sp_iqcolumnuse](#) プロシージャ」(371 ページ)、[「sp_iqdbspace](#) プロシージャ」(387 ページ)、[「sp_iqindexadvice](#) プロシージャ」(418 ページ)、[「sp_iqtableuse](#) プロシージャ」(475 ページ)、[「sp_iqunusedcolumn](#) プロシージャ」(479 ページ)、[「sp_iqunusedindex](#) プロシージャ」(481 ページ)、[「sp_iqunusedtable](#) プロシージャ」(482 ページ)、および [「sp_iqworkmon](#) プロシージャ」(489 ページ)

『リファレンス : 文とオプション』の「[第2章 データベース・オプション](#)」の「[INDEX_ADVISOR オプション](#)」

sp_iqjoinindex プロシージャ

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | ジョイン・インデックスに関する情報を表示します。 |
| 構文 | sp_iqjoinindex [<i>left-table-name</i>], [<i>left-column-name</i>], [<i>left-table-owner</i>], [<i>right-table-name</i>], [<i>right-column-name</i>], [<i>right-table-owner</i>] |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 使用法 | <p>left-table-name ジョイン操作の左側を構成するテーブルの名前。</p> <p>left-column-name ジョインの左側の一部を構成するカラムの名前。</p> <p>left-table-owner ジョイン操作の左側を構成するテーブルの所有者。</p> <p>right-table-name ジョイン操作の右側を構成するテーブルの名前。</p> <p>right-column-name ジョインの右側の一部を構成するカラムの名前。</p> <p>right-table-owner ジョイン操作の右側を構成するテーブルの所有者。</p> <p>sp_iqjoinindex プロシージャは、パラメータなしで呼び出せます。パラメータを指定しない場合、sp_iqjoinindex は IQ ベース・テーブルのすべてのジョイン・インデックスに関する情報を表示します。ジョイン・インデックス・テーブルは常に、IQ ベース・テーブルであることに注意してください。テンポラリ・テーブル、リモート・テーブル、またはプロキシ・テーブルは、ジョイン・インデックス・テーブルとして使用することはできません。</p> <p>最初の 5 つのパラメータを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に NULL を指定する必要があります。たとえば、sp_iqjoinindex NULL, NULL, NULL, t2, n2, DB' や sp_iqjoinindex t1, NULL, NULL, t2 のように指定します。</p> |

表 7-38 : sp_iqjoinindex の使用例

| 構文 | 出力 |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------|
| sp_iqjoinindex | すべてのジョイン・インデックスに関する情報を表示します。 |
| sp_iqjoinindex t1, NULL, DBA | DBA によって所有されている t1 が操作の左側を構成するすべてのジョイン・インデックスに関する情報を表示します。 |
| sp_iqjoinindex t2, n1, DBA | ジョインの左側として DBA によって所有されているテーブル t2 のカラム n1 についてのジョイン・インデックス情報を表示します。 |
| sp_iqjoinindex NULL, NULL, DBA, NULL, NULL, DBA | 左右のテーブルが DBA によって所有されているすべてのジョイン・インデックスに関する情報を表示します。 |
| sp_iqjoinindex NULL, NULL, NULL, t2, NULL, NULL | テーブル t2 がジョイン操作の右側にあるすべてのジョイン・インデックスに関する情報を表示します。 |
| sp_iqjoinindex t1, n1, DBA, t2, n1, DBA | 左側が DBA によって所有されているテーブル t1 の n1 カラム、右側が DBA によって所有されているテーブル t2 の n1 カラムであるジョイン・インデックスに関する情報を表示します。 |
| sp_iqjoinindex non_existing_table | テーブル non_existing_table が存在しないため、ローは返されません。 |
| sp_iqjoinindex NULL, NULL, non_existing_user | ユーザ non_existing_user が存在しないため、ローは返されません。 |

参照

『リファレンス：文とオプション』の「[CREATE JOIN INDEX 文](#)」

『システム管理ガイド 第 1 巻』の「[第 6 章 Sybase IQ インデックスの使用](#)」

説明

sp_iqjoinindex ストアド・プロシージャは、データベース内のジョイン・インデックスに関する情報を表示します。1 つ以上のパラメータを指定した場合、指定したパラメータにより結果がフィルタされます。たとえば、*left-table-name* を指定した場合、sp_iqjoinindex は該当テーブルがジョインの左側を構成するすべてのジョイン・インデックスを表示します。*left-table-owner* を指定した場合、sp_iqjoinindex は左のテーブルが指定の所有者によって所有されているジョイン・インデックスのみを返します。パラメータを指定しない場合、sp_iqjoinindex はデータベース内のすべてのジョイン・インデックスに関する情報を表示します。

sp_iqjoinindex プロシージャは、次のカラムに情報を返します。

表 7-39 : sp_iqjoinindex のカラム

| カラム名 | 説明 |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| joinindex_name | ジョイン・インデックス名。 |
| creator | ジョイン・インデックスの所有者。 |
| left_table_name | ジョイン操作の左側を構成するテーブルの名前。 |
| left_table_owner | ジョイン操作の左側を構成するテーブルの所有者名。 |
| left_column_name | ジョインの左側の一部を構成するカラムの名前。 |
| join_type | 現在サポートしている値は“=”のみ。 |
| right_table_name | ジョイン操作の右側を構成するテーブルの名前。 |
| right_table_owner | ジョイン操作の右側を構成するテーブルの所有者名。 |
| right_column_name | ジョインの右側の一部を構成するカラムの名前。 |
| key_type | 次のキーのジョインのタイプを定義します。 <ul style="list-style-type: none"> • NATURAL : ナチュラル・ジョイン • KEY : キー・ジョイン • ON : 左外部ジョイン/右外部ジョイン/フル・ジョイン |
| valid | このジョイン・インデックスを同期する必要があるかどうかを示します。‘Y’の場合は同期の必要はなく、‘N’の場合は同期の必要があります。 |
| remarks | コメント文字列。 |
| dbspace_id | 指定のジョイン・インデックスが存在する DB 領域の名前。 |

例 テーブル **t1** がジョイン操作の左側を構成するジョイン・インデックスに関する情報を表示します。

```
sp_iqjoinindex t1
```

```
joindex_name creator left_table_name left_table_owner left_column_name
join_type right_table_name right_table_owner right_column_name key_type
valid dbspace_id remarks
t1_t2_t3_join DBA t1 DBA n1
= t2 DBA n1 NATURAL
Y 16387 (NULL)
```

テーブル **t2** がジョイン操作の左側を構成するジョイン・インデックスに関する情報を表示します。

```
sp_iqjoinindex t2
```

```
joindex_name creator left_table_name left_table_owner left_column_name
join_type right_table_name right_table_owner right_column_name key_type
valid dbspace_id remarks
t1_t2_t3_join DBA t2 DBA n1
```

```

=          t3          DBA          n1          NATURAL
Y          (NULL)
t1_t2_t3_join DBA      t2          DBA          name
=          t3          DBA          name          NATURAL
Y          16387      ((NULL))

```

左側が DBA によって所有されているテーブル t2 のカラム name、右側が DBA によって所有されているテーブル t3 のカラム name であるジョイン・インデックスに関する情報を表示します。

```
sp_iqjoinindex t2, name, DBA, t3, name, DBA
```

```

joinindex_name creator left_table_name left_table_owner left_column_name
join_type right_table_name right_table_owner right_column_name key_type
valid dbspace_id remarks
t1_t2_t3_join DBA      t2          DBA          name
=          t3          DBA          name          NATURAL
Y          16387      ((NULL))

```

sp_iqjoinindexsize プロシージャ

機能 指定されたジョイン・インデックスのサイズを取得します。

構文 `sp_iqjoinindexsize (join_index_name)`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 インデックスの合計サイズを、バイト、キロバイト、および Nblocks (IQ ブロック) で返します。また、メモリにジョイン・インデックスを保持するために必要なページ数と、(ディスク上で)ジョイン・インデックスを圧縮した場合の IQ ページ数も返します。このプロシージャには、必ず `join_index_name` パラメータを指定してください。

表 7-40 : sp_iqjoinindexsize のカラム

| カラム名 | 説明 |
|------------------|---------------------------------|
| Username | ジョイン・インデックスの所有者 |
| JoinIndexName | 結果を返すジョイン・インデックス |
| Number of Tables | ジョイン・インデックス内のテーブル数 |
| KBytes | 物理オブジェクト・サイズ(キロバイト) |
| Pages | メモリ内にオブジェクトを保持するために必要な IQ ページの数 |

| カラム名 | 説明 |
|------------------|----------------------------------|
| Compressed Pages | (ディスク上で) オブジェクトを圧縮した場合の IQ ページの数 |
| NBlocks | IQ ブロックの数 |

例 `sp_iqjoinindexsize ('t1t2')`

| Username | JoinIndexName | Number of Tables | KBytes | Pages | Compressed Pages | NBlocks |
|----------|---------------|------------------|--------|-------|------------------|---------|
| DBA | t1t2 | 2 | 13 | 15 | 4 | 26 |

sp_iqlmconfig プロシージャ

機能 ライセンス管理設定を制御し、ライセンスのタイプとステータスを表示して設定します。

構文 1 `sp_iqlmconfig 'edition', { 'SE' | 'SA' | 'EE' }`

表 7-41 : "edition" パラメータの要約情報

| トピック名 | 値 |
|-------|---------------------------------------------------------------------------------|
| デフォルト | 'EE' (Enterprise Edition) |
| 値の範囲 | 'SE' (Small Business) 'SA' (Single Application) 'EE' (Enterprise Edition) |
| ステータス | Static |

構文 2 `sp_iqlmconfig 'license type', { 'CP' | 'DT' | 'SF' | 'AC' | 'BC' | 'CH' | 'DH' | 'SH' | 'AH' | 'BH' }`

表 7-42 : "license type" パラメータの要約情報

| トピック名 | 値 |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| デフォルト | 'DT' (開発およびテスト) |
| 値の範囲 | 'AC' (OEM CPU ライセンス) 'AH' (OEM CPU ライセンス・チップ) 'BC' (OEM スタンバイ・ライセンス) 'BH' (OEM スタンバイ・ライセンス・チップ) 'CP' (CPU ライセンス) 'CH' (CPU ライセンス・チップ) 'DH' (開発およびテスト・ライセンス・チップ) 'DT' (開発およびテスト) 'EV' (評価) 'SF' (スタンバイ CPU ライセンス) 'SH' (スタンバイ CPU ライセンス・チップ) |
| ステータス | Static |

| | |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 3 | <pre>sp_iqilmconfig 'email severity', { 'ERROR' 'WARNING' 'INFORMATIONAL' 'NONE' }</pre> |
| | NONE は、電子メール通知の無効を指定します。 |
| 構文 4 | <pre>sp_iqilmconfig 'smtp host', '<hostname>' '</pre> <p>hostname 電子メール通知に使用する SMTP ホストを指定します。</p> |
| 構文 5 | <pre>sp_iqilmconfig 'email sender', '<email address>' </pre> <p><email address> 電子メール通知に使用する送信側の電子メール・アドレスを指定します。</p> |
| 構文 6 | <pre>sp_iqilmconfig 'email recipients', '<email recipients>' </pre> <p><email recipients> 電子メール通知を送信する電子メール・アドレスのカンマで区切られたリストを指定します。</p> |
| 構文 7 | <pre>sp_iqilmconfig </pre> |
| パーミッション | DBA パーミッションが必要です。 |
| 使用法 | 起動時に、 <code>sp_iqilmconfig</code> は、指定のライセンスのエディション・タイプとライセンス・タイプをチェックします。 |
| | <ul style="list-style-type: none">指定のライセンスが見つからない場合、サーバは猶予モードになります。指定のライセンス・タイプは、NULL 以外のエディション値が指定された場合のみ有効になります。<code>sp_iqilmconfig</code> はパラメータなしで呼び出され (構文 3)、上記のすべての情報に加え、次に示す情報も表示します。<ul style="list-style-type: none">製品エディションまたはライセンス・タイプ使用しているオプションのライセンスライセンス・カウント電子メール情報ライセンスに関する一般情報 |

sp_iqlocks プロシージャ

| | |
|----|--------------------------------------------------------------------------------------|
| 機能 | IQ ストアとカタログ・ストアの両方に関して、データベース内のロックに関する情報を表示します。 |
| 構文 | <pre>sp_iqlocks ([connection,] [[owner.]table_name] max_locks,] [sort_order])</pre> |

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法

表 7-43 は、結果の絞り込みに使用できる、オプションの `sp_iqlocks` のパラメータを示します。

表 7-43 : `sp_iqlocks` のパラメータ (オプション)

| 名前 | データ型 | 説明 |
|-------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>connection</code> | integer | 接続 ID。このオプションを指定すると、指定された接続内のロックに関する情報のみが返されます。デフォルトは 0 (すべての接続に関する情報を返します)。 |
| <code>owner.table_name</code> | char (128) | テーブル名。このオプションを指定すると、指定されたテーブル内のロックに関する情報のみが返されます。デフォルトは NULL (データベース内のすべてのテーブルに関する情報を返します)。owner を指定しなかった場合、プロシージャを呼び出したユーザがテーブルの所有者と見なされます。 |
| <code>max_locks</code> | integer | 情報を返すロックの最大数。デフォルトは 0 (すべてのロック情報を返します)。 |
| <code>sort_order</code> | char(1) | 情報を返す順序。 <ul style="list-style-type: none"> • C : 接続ごとにソート (デフォルト) • T : table_name ごとにソート |

説明

データベース内の現在のロックに関する情報を表示します。指定する接続によって、単一の接続内のロック、単一のテーブル内のロック、指定された数のロック、というように結果を絞り込むことができます。

`sp_iqlocks` は、次の情報を `sort_order` パラメータに指定されたとおりにソートして表示します。

表 7-44 : `sp_iqlocks` のカラム

| カラム | 説明 |
|-------------------------|----------------------------------------------------------------------------------|
| <code>conn_name</code> | 現在の接続の名前。 |
| <code>conn_id</code> | ロックが存在する接続 ID。 |
| <code>user_id</code> | この接続 ID に関連付けられているユーザ。 |
| <code>table_type</code> | テーブルの種類。これは、テーブルを表す BASE、グローバル一時テーブルを表す GLBTMP、マテリアライズド・ビューを表す MVIEW のいずれかになります。 |

| カラム | 説明 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| creator | テーブルの所有者。 |
| table_name | ロックが存在するテーブル。 |
| index_id | インデックス ID または NULL。 |
| lock_class | <p>ロックのタイプを示す文字列。</p> <p>S – Share (共有) SW – Share and Write (共有/書き込み) EW – Exclusive and Write (排他/書き込み) E – Exclusive (排他) P – Phantom (幻) A – Antiphantom (幻でない) W – Write (書き込み)</p> <p>すべてのロックには、S、E、EW、SW のいずれかと、P、A のいずれかまたは両方が表示されます。幻ロックと幻でないロックには、T または * の修飾子が付きます。</p> <p>T – ロックは逐次スキャンに関連します。 * – ロックはすべてのスキャンに関連します。 <i>nnn</i> – インデックス番号。ロックが特定のインデックスに関連する場合。</p> <p>Sybase IQ は、共有ロックを取得してから書き込みロックを取得します。接続に排他ロックがある場合、共有ロックは表示されません。書き込みロックの場合は、排他、共有、書き込みのすべてのロックが接続にあれば、EW が表示されます。</p> |
| lock_type | ロックを識別する値 (ロック・クラスに依存)。 |
| row_identifier | ローまたは NULL の識別子。 |

接続 ID またはテーブルでロックされているユーザのユーザ名を見つけれない場合、`sp_iqlocks` には接続 ID に 0 (ゼロ) が表示され、ユーザ名には `User unavailable` が表示されます。

注意 排他ロック、幻ロック、幻でないロックは、SQL Anywhere のテーブルでは使用できますが、Sybase IQ テーブルでは使用できません。カタログ・ストア内のテーブルで明示的にロックを解除しない限り、これらの種類のロックが Sybase IQ データベースで起きる (修飾子 T、*、および *nnn* が表示される) ことはありません。SQL Anywhere テーブルでのロックについては、『SQL Anywhere サーバー SQL の使用法』を参照してください。

例 次の例は、`sp_iqlocks` プロシージャ・コールと、Sybase IQ データベースでのその出力を示します。このプロシージャは、すべてのデフォルト・オプションと共に呼び出されており、すべてのロックが接続ごとにソートされて表示されます。

```
call sp_iqlocks()

conn_name  conn_id  user_id  table_type  creator  table_name
=====  =====  =====  =====  =====  =====
con1       70187172 'mary'   BASE       DBA       t1

index_id          lock_class  lock_duration  lock_type  row_identifier
=====          =====  =====  =====  =====
ASIQ_IDX_T452_C19_FP  Table      Position      Table      1
```

sp_iqmodifyadmin プロシージャ

機能 名前付きログイン・ポリシーのオプションを所定の値に設定します。ログイン・ポリシーを指定しないと、オプションがルート・ポリシーに設定されます。マルチプレックスでは、`sp_iqmodifyadmin` は、マルチプレックス・サーバ名であるオプションのパラメータを指定します。

構文 1 `call sp_iqmodifyadmin ('policy_option_name', 'value_in' [,login_policy_name])`

構文 2 `sp_iqmodifyadmin 'policy_option_name', 'value_in' [,login_policy_name]`

構文 3 `sp_iqmodifyadmin policy_option_name, value_in, login_policy_name`

構文 4 `sp_iqmodifyadmin 'policy_option_name', 'value_in' [,login_policy_name] [,server_name]`

使用法 `policy_option_name` 変更するログイン・ポリシー・オプション。

`value_in` ログイン・ポリシー・オプションの新しい値。

`login_policy_name` 変更に必要なログイン・ポリシー・オプションを持つログイン・ポリシーの名前。

パーミッション DBA 権限が必要です。

参照 [「sp_iqpassword プロシージャ」\(439 ページ\)](#)

『リファレンス：文とオプション』の「第1章 SQL 文」の「ALTER LOGIN POLICY 文」

例 **例 1** *lockeduser* という名前のポリシーについて、ログイン・オプション *locked* を ON に設定します。

```
call sp_iqmodifyadmin ('locked','on','lockeduser')
```

例 2 *Writer1* という名前のマルチプレックス・サーバ上の *lockeduser* というポリシーについて、ログイン・オプション *locked* を ON に設定します。

```
call sp_iqmodifyadmin  
( 'locked','on','lockeduser','Writer1')
```

sp_iqmodifylogin プロシージャ

機能 ユーザをログイン・ポリシーに割り当てます。

構文 1 **call sp_iqmodifylogin 'userid', ['login_policy_name']**

構文 2 **sp_iqmodifylogin 'userid', ['login_policy_name']**

パーミッション DBA 権限が必要です。

使用法 **userid** 変更するアカウントの名前を保持する変数。

login_policy_name (オプション) ユーザを割り当てるログイン・ポリシーの名前を指定します。ログイン・ポリシー名を指定しないと、ユーザがルート・ログイン・ポリシーに割り当てられます。

例 **例 1** ユーザ *joe* を、*expired_password* という名前のログイン・ポリシーに割り当てます。

```
sp_iqmodifylogin 'joe', 'expired_password'
```

例 2 ユーザ *joe* をルート・ログイン・ポリシーに割り当てます。

```
call sp_iqmodifylogin ('joe')
```

参照 [「sp_iqmodifyadmin プロシージャ」\(435 ページ\)](#)

sp_iqobjectinfo プロシージャ

機能 データベースのオブジェクトおよびサブオブジェクトのパーティションと DB 領域の割り当てを返します。

構文 **sp_iqobjectinfo [owner_name] [, object_name] [, object-type]**

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法

owner_name オブジェクトの所有者。指定した場合、`sp_iqobjectinfo` は、指定の所有者を持つテーブルおよびジョイン・インデックスについてのみ出力を表示します。指定しない場合、`sp_iqobjectinfo` は、データベース内のすべてのユーザのテーブルおよびジョイン・インデックスに関する情報を表示します。

object_name テーブルまたはジョイン・インデックスの名前。指定しない場合、`sp_iqobjectinfo` は、データベース内のすべてのテーブルおよびジョイン・インデックスに関する情報を表示します。

object-type 有効なオブジェクト・タイプは、`table` (デフォルト) または `joinindex` です。

オブジェクト・タイプが `table` の場合は、引用符で囲んでください。

すべてのパラメータがオプションであり、どのパラメータも他のパラメータの値とは関係なく指定できます。

入力パラメータは `sp_iqobjectinfo` と共に使用することをおすすめします。`sp_iqobjectinfo` の結果にクエリを実行でき、クエリの `WHERE` 句で述部を使用する代わりに、入力パラメータを使用することで、クエリのパフォーマンスが向上します。たとえば、クエリ A は次のように記述します。

```
SELECT COUNT(*) FROM sp_iqobjectinfo()  
WHERE owner = 'DBA'  
AND object_name = 'tab_case510'  
AND object_type = 'table'  
AND sub_object_name is NULL  
AND dbspace_name = 'iqmain7'  
AND partition_name = 'P1'
```

クエリ B は、クエリ A を `sp_iqobjectinfo` 入力パラメータを使用するように記述し直したものです。

```
SELECT COUNT(*) FROM  
sp_iqobjectinfo('DBA','tab_case510','table')  
WHERE sub_object_name is NULL  
AND dbspace_name = 'iqmain7'  
AND PARTITION_NAME = 'P1'
```

クエリ B は、クエリ A よりも速く結果を返します。入力パラメータが `sp_iqobjectinfo` に渡されたとき、プロシージャはシステム・テーブル内の少数のレコードを比較してジョインします。つまり、クエリ A に比べ作業が少なくなります。クエリ B では、述部はプロシージャ自体に適用され、プロシージャは小さい結果セットを返します。そのため、より少数の述部がクエリに適用されます。

`sp_iqobjectinfo` ストアド・プロシージャは、`owner_name`、`object_name`、および `object_type` の解釈のために、ワイルドカード文字をサポートしています。これは、LIKE 句がクエリ内のパターンを照合するのと同じ方法で、指定のパターンと一致するすべての DB 領域の情報を表示します。

説明

特定のまたはすべてのデータベース・オブジェクト (テーブルおよびジョイン・インデックス・タイプのみ) とそのサブオブジェクトの、すべてのパーティションと DB 領域割り当てを返します。サブオブジェクトは、カラム、インデックス、プライマリ・キー、一意性制約、および外部キーです。

表 7-45 : `sp_iqobjectinfo` のカラム

| カラム名 | 説明 |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <code>owner</code> | オブジェクトの所有者の名前。 |
| <code>object_name</code> | DB 領域にあるオブジェクトの名前 (テーブルおよびジョイン・インデックス・タイプのみ)。 |
| <code>sub_object_name</code> | DB 領域に存在するオブジェクトの名前。 |
| <code>object_type</code> | オブジェクトのタイプ (カラム、インデックス、プライマリ・キー、一意性制約、外部キー、パーティション、ジョイン・インデックスまたはテーブル)。 |
| <code>object_id</code> | オブジェクトのグローバル・オブジェクト ID。 |
| <code>id</code> | オブジェクトのテーブル ID またはジョイン・インデックス ID。 |
| <code>dbspace_name</code> | オブジェクトが存在する DB 領域の名前。文字列 “[multiple]” は、分割されたオブジェクトの特別なメタ・ローの場合に表示されます。[multiple] ローは、テーブルまたはカラムを説明するために、出力に複数のローが伴うことを示します。 |
| <code>partition_name</code> | 指定のオブジェクトのパーティションの名前。 |

例

指定のユーザによって所有されている特定のデータベース・オブジェクトおよびサブオブジェクトのパーティションおよび DB 領域割り当てに関する情報を表示します。

```
sp_iqobjectinfo GROUPO,Departments
```

| owner | object_name | sub_object_name | object_type | object_id | id |
|--------|-------------|--------------------------------|-------------|-----------|-----|
| GROUPO | Departments | (NULL) | table | 3632 | 738 |
| GROUPO | Departments | DepartmentID | column | 3633 | 738 |
| GROUPO | Departments | DepartmentName | column | 3634 | 738 |
| GROUPO | Departments | DepartmentHeadID | column | 3635 | 738 |
| GROUPO | Departments | DepartmentsKey | primary key | 83 | 738 |
| GROUPO | Departments | FK_DepartmentHeadID_EmployeeID | foreign key | 92 | 738 |

| | |
|--------------|----------------|
| dbspace_name | partition_name |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |

object-type table の指定のユーザによって所有されている特定のデータベース・オブジェクトおよびサブオブジェクトのパーティションおよび DB 領域割り当てに関する情報を表示します。

```
sp_iqobjectinfo DBA,sale,'table'
```

| owner | object_name | sub_object_name | object_type | object_id | id |
|-------|-------------|-----------------|-------------|-----------|-----|
| DBA | sale | (NULL) | table | 3698 | 742 |
| DBA | sale | prod_id | column | 3699 | 742 |
| DBA | sale | month_num | column | 3700 | 742 |
| DBA | sale | rep_id | column | 3701 | 742 |
| DBA | sale | sales | column | 3702 | 742 |

| | |
|--------------|----------------|
| dbspace_name | partition_name |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |
| iq_main | (NULL) |

sp_iqpassword プロシージャ

機能 ユーザのパスワードを変更します。ユーザの作成には、CREATE USER 文を使用する方法が適しています。『リファレンス：文とオプション』の「第1章 SQL 文」の「CREATE USER 文 CREATE USER 文」を参照してください。

構文 1 `call sp_iqpassword ('caller_password', 'new_password' [, 'user_name'])`

構文 2 `sp_iqpassword 'caller_password', 'new_password' [, 'user_name']`

パーミッション 自分のパスワードを設定する場合には不要ですが、他のユーザのパスワードを設定する場合には DBA 権限が必要です。

使用法 **caller_password** 自分のパスワード。自分のパスワードを変更する場合は、古い方のパスワードを指定します。DBA が他のユーザのパスワードを変更する場合、caller_password は DBA のパスワードです。

new_password ユーザまたは *loginname* の新しいパスワードです。

| | |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>user_name DBA がパスワードを変更するユーザのログイン名。自分のパスワードを変更する場合は user_name を指定しません。</p> |
| 説明 | <p>すべてのユーザが sp_iqpassword によって自分のパスワードを変更できます。DBA は、すべての既存のユーザのパスワードを変更できます。</p> |
| 例 | <p>例 1 ログインしたユーザのパスワードを irk103 から exP984 に変更します。</p> <pre>sp_iqpassword 'irk103', 'exP984'</pre> <p>例 2 ログインしているユーザが DBA 権限を持っているか、またはユーザが joe 自身の場合のみ、ユーザ joe のパスワードを epr45 から pdi032 に変更します。</p> <pre>call sp_iqpassword ('epr45', 'pdi932', 'joe')</pre> |

sp_iqkeys プロシージャ

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | <p>プライマリ・キーとプライマリ・キー制約に関する情報を、テーブル、カラム、テーブル所有者別に表示します。または、データベース内のすべての Sybase IQ テーブルについて表示します。</p> |
| 構文 | <pre>sp_iqkeys { [<i>table-name</i>], [<i>column-name</i>], [<i>table-owner</i>] }</pre> |
| パーミッション | <p>DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。</p> |
| 使用法 | <p>table-name ベース・テーブルまたはグローバル・テンポラリ・テーブルの名前。このオプションを指定すると、指定されたテーブルで定義されたプライマリ・キーに関する情報のみが返されます。</p> <p>column-name カラムの名前。このオプションを指定すると、指定されたカラムのプライマリ・キーに関する情報のみが返されます。</p> <p>table-owner テーブルの所有者。このオプションを指定すると、指定された所有者によって所有されているテーブルのプライマリ・キーに関する情報のみが返されます。</p> <p>1 つ以上のパラメータを指定できます。最初の 2 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に NULL を指定する必要があります。パラメータを指定しない場合、データベース内のすべてのテーブルのすべてのプライマリ・キーの説明が表示されます。指定したいいずれかのパラメータが無効な場合、出力にローが表示されません。</p> |

表 7-46 : sp_iqpkeys の使用例

| 構文 | 出力 |
|---------------------------------|---------------------------------------------------------------------|
| sp_iqpkeys sales | テーブル sales 上で定義されたプライマリ・キーに関する情報を表示します。 |
| sp_iqpkeys sales, NULL, DBA | DBA によって所有されているテーブル sales 上で定義されたプライマリ・キーに関する情報を表示します。 |
| sp_iqpkeys sales, store_id, DBA | DBA によって所有されているテーブル sales のカラム store_id で定義されたプライマリ・キーに関する情報を表示します。 |
| sp_iqpkeys NULL, NULL, DBA | DBA によって所有されているすべてのテーブル上で定義されたプライマリ・キーに関する情報を表示します。 |

説明

sp_iqpkeys ストアド・プロシージャは、データベース内のベース・テーブルおよびグローバル・テンポラリ・テーブル上のプライマリ・キーに関する次の情報を表示します。

表 7-47 : sp_iqpkeys のカラム

| カラム名 | 説明 |
|-----------------|------------------------|
| table_name | テーブルの名前 |
| table_owner | テーブルの所有者 |
| column_name | プライマリ・キーが定義されているカラムの名前 |
| column_id | カラム ID |
| constraint_name | プライマリ・キー制約の名前 |
| constraint_id | プライマリ・キー制約の ID |

注意 sp_iqpkeys ストアド・プロシージャは、Sybase IQ バージョン 12.6 以降で作成されたデータベースにのみ存在します。

例

テーブル sales1 のカラムで定義されたプライマリ・キーを表示します。

```
sp_iqpkeys sales1
```

```
table_name table_owner column_name column_id constraint_name constraint_id
sales1      DBA          store_id    1          MA114        114
```

テーブル sales2 のカラムで定義されたプライマリ・キーを表示します。

```
sp_iqpkeys sales2
```

```
table_name table_owner column_name column_id constraint_name constraint_id
```

```
sales2          DBA          store_id,      1,2          MA115          115
                order_num
```

テーブル **sales2** のカラム **store_id** で定義されたプライマリ・キーを表示します。

```
sp_iqpkkeys sales2, store_id
```

```
table_name table_owner column_name column_id constraint_name constraint_id
sales2     DBA          store_id      1          MA115        115
```

参照

[「sp_iqindex および sp_iqindex_alt プロシージャ」 \(415 ページ\)](#)

[「sp_iqcolumn プロシージャ」 \(369 ページ\)](#)

sp_iqprocedure プロシージャ

機能

システム・プロシージャおよびユーザ定義プロシージャに関する情報を表示します。

構文

```
sp_iqprocedure [ proc-name ], [ proc-owner ], [ proc-type ]
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法

proc-name プロシージャ名。

proc-owner プロシージャの所有者。

proc-type プロシージャのタイプ。指定できる値は次のとおりです。

- **SYSTEM** : システム・プロシージャ (ユーザ **SYS** または **dbo** によって所有されているプロシージャ) に関する情報のみを表示します。
- **ALL** : ユーザおよびシステムのプロシージャに関する情報を表示します。
- その他の値 : ユーザのプロシージャに関する情報を表示します。

sp_iqprocedure プロシージャは、パラメータなしで呼び出せます。パラメータを指定しない場合、すべてのユーザ定義プロシージャ (**dbo** または **SYS** によって所有されていないプロシージャ) に関する情報のみがデフォルトで表示されます。

最初の 2 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に **NULL** を指定する必要があります。たとえば、**sp_iqprocedure NULL, NULL, SYSTEM** や **sp_iqprocedure NULL, user1** のように指定します。

表 7-48 : `sp_iqprocedure` の使用例

| 構文 | 出力 |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sp_iqprocedure</code> | <code>dbo</code> または <code>SYS</code> によって所有されていないデータベース内のすべてのプロシージャに関する情報を表示します。 |
| <code>sp_iqprocedure sp_test</code> | プロシージャ <code>sp_test</code> に関する情報を表示します。 |
| <code>sp_iqprocedure non_existing_proc</code> | プロシージャ <code>non_existing_proc</code> が存在しないため、ローは返されません。 |
| <code>sp_iqprocedure NULL, DBA</code> | <code>DBA</code> によって所有されているすべてのプロシージャに関する情報を表示します。 |
| <code>sp_iqprocedure sp_test, DBA</code> | <code>DBA</code> によって所有されているプロシージャ <code>sp_test</code> に関する情報を表示します。 |
| <code>sp_iqprocedure sp_iqtable</code> | プロシージャ <code>sp_iqtable</code> はシステム・プロシージャではありません。 <code>sp_iqtable</code> という名前のユーザ定義プロシージャが存在しない場合、ローは返されません (デフォルトでは、ユーザ定義プロシージャのみが返されます)。 |
| <code>sp_iqprocedure sp_iqtable, dbo</code> | <code>sp_iqtable</code> プロシージャはユーザ・プロシージャではないため、ローは返されません (デフォルトではユーザ・プロシージャのみが返されます)。 |
| <code>sp_iqprocedure NULL, NULL, SYSTEM</code> | (<code>dbo</code> または <code>SYS</code> によって所有されている) すべてのシステム・プロシージャに関する情報を表示します。 |
| <code>sp_iqprocedure sp_iqtable, NULL, 'SYSTEM</code> | システム・プロシージャ <code>sp_iqtable</code> に関する情報を表示します。 |
| <code>sp_iqprocedure sp_iqtable, dbo, ALL</code> | <code>dbo</code> によって所有されているシステム・プロシージャ <code>sp_iqtable</code> に関する情報を表示します。 |

説明

`sp_iqprocedure` ストアド・プロシージャは、データベース内のプロシージャに関する情報を表示します。1 つ以上のパラメータを指定した場合、指定したパラメータにより結果がフィルタされます。たとえば、`proc-name` を指定した場合、指定のプロシージャに関する情報のみが表示されます。`proc-owner` を指定した場合、`sp_iqprocedure` は指定の所有者によって所有されているプロシージャに関する情報のみを返します。パラメータを指定しない場合、`sp_iqprocedure` はデータベース内のすべてのユーザ定義プロシージャに関する情報を表示します。

`sp_iqprocedure` プロシージャは、次のカラムに情報を返します。

表 7-49 : sp_iqprocedure のカラム

| カラム名 | 説明 |
|------------|-------------------------------------------------------------------------------------------|
| proc_name | プロシージャの名前。 |
| proc_owner | プロシージャの所有者。 |
| proc_defn | プロシージャの作成に使用するコマンド。隠しプロシージャでは、キーワード 'HIDDEN' が表示されます。 |
| replicate | このプロシージャが Replication Server のインストール環境においてプライマリ・データ・ソースである場合は Y が表示され、それ以外の場合は N が表示されます。 |
| srvid | プロシージャがリモート・データベース・サーバ上にある場合にリモート・サーバを示します。 |
| remarks | コメント文字列。 |

例 ユーザ定義プロシージャ sp_test に関する情報を表示します。

```
sp_iqprocedure sp_test
```

| proc_name | proc_owner | proc_defn | replicate | srvid | remarks |
|-----------|------------|---------------------------------------------------------------------------------|-----------|--------|---------|
| sp_test | DBA | create procedure DBA.sp_test(in n1 integer) begin messageÅsp_testÅfend | N | (NULL) | (NULL) |

ユーザ DBA によって所有されているすべてのプロシージャに関する情報を表示します。

```
sp_iqprocedure NULL, DBA
```

| proc_name | proc_owner | proc_defn | replicate | srvid | remarks |
|-----------|------------|--------------------------------------------------------------------------------|-----------|--------|---------|
| sp_test | DBA | create procedure DBA.sp_test(in n1 integer) begin message'sp_test'end | N | (NULL) | (NULL) |
| sp_dept | DBA | create procedure DBA.sp_dept() begin end | N | (NULL) | (NULL) |

参照 『リファレンス：文とオプション』の「第 1 章 SQL 文」の「CREATE USER 文」

sp_iqprocparm プロシージャ

機能 ストアド・プロシージャに関する情報を表示します。結果セット変数と SQLSTATE/SQLCODE エラー値も含まれます。

構文 `sp_iqprocparm [proc-name], [proc-owner], [proc-type]`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法 **proc-name** プロシージャ名。

proc-owner プロシージャの所有者。

proc-type プロシージャのタイプ。指定できる値は次のとおりです。

- **SYSTEM** : システム・プロシージャ (ユーザ **SYS** または **dbo** によって所有されているプロシージャ) に関する情報のみを表示します。
- **ALL** : ユーザおよびシステムのプロシージャに関する情報を表示します。
- **その他の値** : ユーザのプロシージャに関する情報を表示します。

`sp_iqprocparm` はパラメータなしで呼び出せます。パラメータを指定しない場合、すべてのユーザ定義プロシージャ (**dbo** または **SYS** によって所有されていないプロシージャ) の入出力パラメータおよび結果パラメータがデフォルトで表示されます。

最初の 2 つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に **NULL** を指定する必要があります。たとえば、`sp_iqprocparm NULL, NULL, SYSTEM` や `sp_iqprocparm NULL, user1` のように指定します。

表 7-50 : sp_iqprocparm の使用例

| 構文 | 出力 |
|----------------------------------------------|----------------------------------------------------------------------------|
| <code>sp_iqprocparm</code> | dbo または SYS によって所有されていないデータベース内のすべてのシステム・プロシージャのパラメータを表示します。 |
| <code>sp_iqprocparm sp_test</code> | プロシージャ <code>sp_test</code> に関する情報を表示します。 |
| <code>sp_iqprocparm non_existing_proc</code> | プロシージャ <code>non_existing_proc</code> が存在しないため、ローは返されません。 |
| <code>sp_iqprocparm NULL, DBA</code> | DBA によって所有されているすべてのプロシージャのパラメータを表示します。 |
| <code>sp_iqprocparm sp_test, DBA</code> | DBA によって所有されているプロシージャ <code>sp_test</code> のパラメータを表示します。 |

| 構文 | 出力 |
|----------------------------------------|---------------------------------------------------------------------------------------------------------|
| sp_iqprocparm sp_iqtable | sp_iqtable はシステム・プロシージャです。sp_iqtable という名前のユーザ定義プロシージャが存在しない場合、ローは返されません (デフォルトでは、ユーザ定義プロシージャのみが返されます)。 |
| sp_iqprocparm sp_iqtable, dbo | sp_iqtable プロシージャはユーザ・プロシージャではないため、ローは返されません (デフォルトでは、ユーザ・プロシージャのみが返されます)。 |
| sp_iqprocparm NULL, NULL, SYSTEM | (dbo または SYS によって所有されている) すべてのシステム・プロシージャのパラメータを表示します。 |
| sp_iqprocparm sp_iqtable, NULL, SYSTEM | システム・プロシージャ sp_iqtable のパラメータを表示します。 |
| sp_iqprocparm sp_iqtable, dbo, ALL | dbo によって所有されているシステム・プロシージャ sp_iqtable のパラメータ情報を表示します。 |

説明

sp_iqprocparm ストアド・プロシージャは、ストアド・プロシージャのパラメータに関する情報を表示します。結果セット変数と SQLSTATE/SQLCODE エラー値も含まれます。1 つ以上のパラメータを指定した場合、指定したパラメータにより結果がフィルタされます。たとえば、*proc-name* を指定した場合、指定のプロシージャに対するパラメータの情報のみが表示されます。*proc-owner* を指定した場合、sp_iqprocparm は指定の所有者によって所有されているプロシージャに対するパラメータの情報のみを返します。パラメータを指定しない場合、sp_iqprocparm はデータベース内のすべてのユーザ定義プロシージャに対するパラメータの情報を表示します。

sp_iqprocparm プロシージャは、次のカラムに情報を返します。

表 7-51 : sp_iqprocparm のカラム

| カラム名 | 説明 |
|------------|------------|
| proc_name | プロシージャの名前 |
| proc_owner | プロシージャの所有者 |
| parm_name | パラメータの名前 |

| カラム名 | 説明 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| parm_type | パラメータは、次に示すタイプのいずれかに該当します。 <ul style="list-style-type: none"> 標準のパラメータ (変数) 結果変数：結果セットを返すプロシージャで使用されます。 SQLSTATE エラー値 SQLCODE エラー値 |
| parm_mode | パラメータのモード：プロシージャにパラメータの値を渡すか、戻り値を返すか、それらの両方、あるいはいずれも行いません。パラメータ・モードは、次のいずれかに該当します。 <ul style="list-style-type: none"> in：パラメータはプロシージャに値を渡します。 out：パラメータは値を返します。 inout：パラメータは、値を渡すとともに値を返します。 NULL：パラメータは値を渡すことも返すこともしません。 |
| domain_name | SYSDOMAIN システム・テーブルに挙げられたパラメータのデータ型の名前 |
| width | 文字列パラメータの長さ、数値パラメータの精度、その他すべてのデータ型の格納バイト数 |
| scale | 数値データ型パラメータでは小数点以下の桁数、その他のデータ型では 0 |
| default | 文字列として保持されたパラメータのデフォルト値 |

例 ユーザ定義プロシージャ `sp_test` のパラメータに関する情報を表示します。

```
sp_iqprocparm sp_test
```

```
proc_name proc_owner parm_name parm_type parm_mode domain_name width scale default
sp_test DBA ID normal in integer 4 0 (NULL)
```

システム・プロシージャ `sp_iqshowcompression` のパラメータに関する情報を表示します。

```
sp_iqprocparm sp_iqshowcompression, dbo, system
```

```
proc_name          proc_owner parm_name    parm_type  parm_mode
domain_name width  scale  default
sp_iqshowcompression  dbo          @owner_name normal    in
char          128    0      (NULL)
```

| | | | | |
|----------------------|-----|--------------|--------|-----|
| sp_iqshowcompression | dbo | @table_name | normal | in |
| char | 128 | 0 | (NULL) | |
| sp_iqshowcompression | dbo | @column_name | normal | in |
| char | 128 | 0 | (NULL) | |
| sp_iqshowcompression | dbo | Column | result | out |
| char | 128 | 0 | (NULL) | |
| sp_iqshowcompression | dbo | Compression | result | out |
| char | 3 | 0 | (NULL) | |

参照 『リファレンス：文とオプション』の「[第1章 SQL 文](#)」の「[CREATE PROCEDURE 文](#)」

sp_iqrebuildindex プロシージャ

機能 テーブル上の1つまたは複数のインデックスを、CREATE TABLE 文で指定された元の IQ UNIQUE の値、または、新しい IQ UNIQUE 値で再構築し、必要な記憶領域やクエリのパフォーマンスを変えます。デフォルト・インデックス以外のインデックスを再構築するには、インデックス名を指定します。

構文 `sp_iqrebuildindex (table_name, index_clause)`

パーミッション テーブル上のインデックスを再構築するには、そのテーブルの INSERT パーミッションが必要です。

使用法 **table_name** インデックスの再構築作業が行われるテーブルの、部分的または完全に修飾された名前。テーブルの所有者であるユーザがこのプロシージャを実行する場合、部分的に修飾された名前を使用できます。そうでない場合は、完全に修飾された名前である必要があります。

index_clause 次に示す1つ以上の文字列。スペースで区切って指定します。

`column column_name [count]`

`[dbspace dbspace_name]`

`index index_name`

`[dbspace dbspace_name]`

各 `column_name` または `index_name` は、指定されたテーブル上のカラムまたはインデックスを参照するものである必要があります。`column_name` または `index_name` を複数指定した場合、プロシージャはエラーを返し、インデックスは再構築されません。

count は、IQ UNIQUE の値を示す、負でない数値です。CREATE TABLE 文では、IQ UNIQUE (*count*) が、特定のカラム内にある、重複しない値の数を概算します。重複しない値の数は、クエリ の速度と格納領域の要件に影響します。詳細については、『システム管理ガイド 第1巻』の「第5章 データベース・オブジェクトの使用」の「記憶領域とクエリ・パフォーマンスの最適化」を参照してください。

キーワード *column* および *index* を指定する必要があります。キーワード *dbspace* はオプションです。これらのキーワードでは大文字と小文字が区別されません。

Sybase IQ は、カラムまたはインデックスの再構築の場合に *dbspace dbspace-name* が指定されないと、カラムまたはインデックスのオリジナルが存在していた DB 領域に再構築します。

参照

「[sp_iqindexfragmentation プロシージャ](#)」(419 ページ) および
「[sp_iqrowdensity プロシージャ](#)」(454 ページ)

『システム管理ガイド 第1巻』の「第6章 Sybase IQ インデックスの使用」

『リファレンス: 文とオプション』の「第2章 データベース・オプション」の「[FP_LOOKUP_SIZE オプション](#)」および「[MINIMIZE_STORAGE オプション](#)」

説明

カラム名を指定した場合、このプロシージャはそのカラムのデフォルト・インデックスを再構築します。インデックス名を指定する必要はありません。この場合に、カラム名に加えて Sybase IQ によって割り当てられたデフォルト・インデックスの名前を指定すると、エラーが発生します。*column_name* の後の *count* を省略すると、0 (ゼロ) がデフォルトとして使用されます。

デフォルト・インデックスが1バイト・インデックスの場合、*sp_iqrebuildindex* は IQ UNIQUE に指定された値にかかわらず、これを必ず1バイト・インデックスとして再構築します。

1バイトのデフォルト・インデックスでは、0、または256より大きい数字が *column_name (count)* に指定された場合、SYS.SYSIQCOLUMN 内の *approx_unique_count* カラムはカラムのカーディナリティの値を使用して更新されます。

カラムのデータ型が255バイトよりも大きい VARCHAR または VARBINARY である場合、*sp_iqrebuildindex* はデフォルト・インデックスを再構築しません。

sp_iqrebuildindex は、データ型 LONG VARCHAR (CLOB) のカラムに WD インデックスを再構築します。

デフォルト・インデックスが 2 バイト・インデックスであり、指定されたカウントが 0、または 65536 よりも大きい数である場合、カラムのカーディナリティの値によって、デフォルトを 1 バイト・インデックスと 2 バイト・インデックスのどちらで再構築するかが決定されます。このカーディナリティの値は、SYS.SYSIQCOLUMN の approx_unique_count カラムを更新するのに使用されます。

IQ UNIQUE に 0 以外の値を指定した場合、デフォルト・インデックスは 1 バイト、2 バイト、またはフラット・デフォルト・インデックスとして再構築されます (前述の例外あり)。

IQ UNIQUE の値として 0 を指定した場合や、IQ UNIQUE に値を指定しなかった場合は、次のように、MINIMIZE_STORAGE オプションがインデックスの再構築方法を制御します。

- MINIMIZE_STORAGE オプションが ON の場合、インデックスは 1 バイトのデフォルト・インデックスとしてまず再構築され、その後必要に応じて 2 バイトまたはフラットに変換される。
- MINIMIZE_STORAGE が OFF の場合、インデックスはデータ型のデフォルトに従って再構築される。詳細については、『システム管理ガイド 第 1 巻』の「第 7 章 システム・プロシージャ」の「[Sybase IQ インデックス・タイプ](#)」を参照。

例

カラム *Surname* のデフォルト・インデックスを再構築します。

```
sp_iqrebuildindex 'empl', 'column dept_id'
```

または

```
call sp_iqrebuildindex ('empl', 'column dept_id')
```

カラム *c1* のデフォルト・インデックスを作成します。

```
CREATE TABLE mytable (c1 int IQ UNIQUE 1000000000)
```

デフォルトの 1 バイト・インデックスを 2 バイト・インデックスに変換します。

```
sp_iqrebuildindex 'mytable', 'column c1 1024'
```

または

```
call sp_iqrebuildindex ('mytable', 'column c1 1024')
```

参照

「[sp_iqindexfragmentation プロシージャ](#)」(419 ページ) および
「[sp_iqrowdensity プロシージャ](#)」(454 ページ)

『リファレンス：文とオプション』の「第 2 章 データベース・オプション」の「[FP_LOOKUP_SIZE オプション](#)」および「[MINIMIZE_STORAGE オプション](#)」

『システム管理ガイド 第1巻』の「第6章 Sybase IQ インデックスの使用」

sp_iqrename プロシージャ

機能 ユーザが作成したテーブル、カラム、インデックス、制約(一意性、プライマリ・キー、外部キー、検査)、ストアド・プロシージャ、関数の名前を変更します。

構文 `sp_iqrename object-name, new-name [, object-type]`

パーミッション テーブルの所有者であるか、またはオブジェクトの DBA 権限または alter パーミッションを持っていないければなりません。オブジェクトへの排他的アクセスが必要です。

使用法 **object-name** ユーザの作成したオブジェクトの元の名前。

オプションで、*owner-name* は、*object-name* の一部として *owner-name.object-name* の形式で指定できます。ここで、*owner-name* は名前を変更するオブジェクトの所有者名です。*owner-name* を指定しない場合、`sp_iqrename` を呼び出したユーザがオブジェクトの所有者と見なされます。`sp_iqrename` を呼び出したユーザがオブジェクトの名前を変更するために必要なパーミッションを持つ場合にのみ、オブジェクトの名前が正しく変更されることに注意してください。

名前を変更するオブジェクトがカラム、インデックス、または制約である場合、オブジェクトが関連付けられているテーブルの名前を指定する必要があります。カラム、インデックス、または制約の場合、*object-name* は *table-name.object-name* または *owner-name.table-name.object-name* の形式で指定できます。

new-name オブジェクトの新しい名前。この名前は、ID の規則に従っており、名前を変更するオブジェクト・タイプに対してユニークであることが必要です。

object-type 名前を変更するユーザ作成オブジェクトのタイプを指定するオプション・パラメータ。つまり、オブジェクト *object-name* のタイプです。*object-type* パラメータは、大文字または小文字のいずれかで指定できます。

表 7-52 : sp_iqrename object-type のパラメータ値

| object-type パラメータ | 指定される内容 |
|-------------------|----------------------|
| column | 名前を変更するオブジェクトはカラム |
| index | 名前を変更するオブジェクトはインデックス |

| object-type パラメータ | 指定される内容 |
|---------------------------|-------------------------------------------------|
| constraint | 名前を変更するオブジェクトは、一意性、プライマリ・キー、検査、または参照 (外部キー) の制約 |
| procedure | 名前を変更するオブジェクトは関数 |
| <i>object-type</i> を指定しない | 名前を変更するオブジェクトはテーブル |

警告！ `sp_iqrename` により名前を変更するオブジェクト上で、依存オブジェクト (プロシージャ、関数、およびビュー) の定義を適切に変更する必要があります。 `sp_iqrename` プロシージャは、従属オブジェクトの定義を自動的に更新しません。これらの定義は手動で変更する必要があります。

説明

`sp_iqrename` ストアド・プロシージャは、ユーザ作成のテーブル、カラム、インデックス、制約 (一意性、プライマリ・キー、外部キー、検査)、および関数の名前を変更します。

オブジェクト・タイプに対してユニークではない名前を持つオブジェクトの名前を変更しようとすると、`sp_iqrename` はメッセージ “項目はすでに存在します。” を返します。

`sp_iqrename` は、ビュー、プロシージャ、イベント、またはデータ型の名前を変更できません。 *object-type* パラメータとして、`event` または `datatype` を指定した場合、`sp_iqrename` により “機能はサポートされていません。” が返されます。

例

ユーザ `shweta` によって所有されているテーブル `titles` の名前を `books` に変更します。

```
sp_iqrename shweta.titles, books
```

テーブル `books` のカラム `id` の名前を `isbn` に変更します。

```
sp_iqrename shweta.books.id, isbn, column
```

テーブル `books` 上のインデックス `idindex` の名前を `isbnindex` に変更します。

```
sp_iqrename books.idindex, isbnindex, index
```

テーブル `books` 上のプライマリ・キー制約 `prim_id` の名前を `prim_isbn` に変更します。

```
sp_iqrename books.prim_id, prim_isbn, constraint
```

参照

『リファレンス：文とオプション』の「[ALTER TABLE 文 RENAME clause](#)」および「[ALTER INDEX 文 RENAME clause](#)」

sp_iq_reset_identity プロシージャ

| | |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定されたテーブルに関連付けられた Identity/Autoincrement カラムのシード値を、指定された値に設定します。 |
| 構文 | sp_iq_reset_identity (<i>table_name</i> , <i>table_owner</i> , <i>value</i>) |
| 使用法 | 構文 <i>table_name</i> 、 <i>table owner</i> 、および <i>value</i> を指定する必要があります。 |
| パーミッション | 不要。 |
| 説明 | Identity/Autoincrement カラムには、自動生成された数字が格納されます。生成された値は、受信データのユニークな識別子です。値は連続したもので、自動生成され、ローがテーブルから削除されても再利用されることはありません。指定されたシード値は、このデフォルトのシード値の代わりとなるもので、データベースのシャットダウンや障害があっても持続します。 |
| 例 | 次の例は、最初のシード値を 50 として Identity カラムを作成します。 <pre>CREATE TABLE mytable(c1 INT identity) call sp_iq_reset_identity('mytable', 'dba', 50)</pre> |
| 参照 | 「sp_iqcolumn プロシージャ」(369 ページ) |

sp_iqrestoreaction プロシージャ

| | |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定された過去の日付の矛盾のない状態にデータベースを戻すために必要なリストア・アクションを示します。 |
| 構文 | sp_iqrestoreaction [<i>timestamp</i>] |
| パラメータ | timestamp 対象の過去の日付を指定します。 |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | データベースを指定のタイムスタンプの矛盾のない状態に戻すことができない場合、 sp_iqrestoreaction は、エラーを返します。それ以外の場合、データベースを矛盾のない状態に戻すリストア・アクションを示します。 データベースをリストアできる時点は通常、指定のタイムスタンプの直前に読み書き可能なファイルをバックアップした最後のバックアップ時間と一致します。バックアップは、包括的なファイルまたは読み書き可能なファイルのいずれかに限られます。 |

出力は、バックアップ時間に基づいた正確な昇順でないことがあります。バックアップ・アーカイブが複数の読み込み専用 dbfile で構成されている場合、複数のロー (同じバックアップ時間とバックアップ ID を持つ) が含まれている可能性があります。

読み込み専用 dbfile または DB 領域を複数回バックアップした場合、リストアには最後のバックアップが使用されます。DB 領域/dbfile の alter ID が、リストアされる最後の読み込み/書き込みバックアップに記録された DB 領域/dbfile の alter ID と一致していれば、対応するバックアップ時間を指定のタイムスタンプ後にすることもできます。

sp_iqrestoreaction は、次の情報を返します。

表 7-53 : sp_iqrestoreaction のカラム

| カラム名 | 説明 |
|---------------------|--------------------------------------------------------------------|
| sequence_number | 処理を行う順序。 |
| backup_id | バックアップ・トランザクションの識別子。 |
| backup_archive_list | バックアップのアーカイブ・ファイルのリスト。 |
| backup_time | バックアップを行った時間。 |
| virtual_type | 仮想バックアップの種類: “Non-virtual”、 “Decoupled”、または “Encapsulated”。 |
| restore_dbSPACE | 空にできます。すべての DB 領域がバックアップ・ アーカイブからリストアされることを示します。 |
| restore_dbfile | 空にできます。指定の DB 領域のすべての dbfile がバックアップ・アーカイブからリストアされる ことを示します。 |
| backup_comment | ユーザ・コメント。 |

例

sp_iqrestoreaction の出力例を示します。

```

sequence_number  backup_id  backup_archive_list  backup_time
1                1192      c:YYYYYtempYYYYYb1  2008-09-23 14:47:40.0
2                1201      c:YYYYYtempYYYYYb2.inc  2008-09-23 14:48:05.01
3                1208      c:YYYYYtempYYYYYb3.inc  2008-09-23 14:48:13.0

virtual_type  restore_dbSPACE  restore_dbfile  backup_comment
Nonvirtual
Nonvirtual
Nonvirtual

```

sp_iqrowdensity プロシージャ

機能

FP インデックス・レベルのテーブルの内部ローの断片化に関する情報をレポートします。

| 構文 | dbo.sp_iqrowdensity ('target ' target:(table table-name (column column-name (...)) | | | | | | | | |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------|-----------|---------|----------------------|------|-----------------|-------|
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 | | | | | | | | |
| 使用法 | table-name ターゲット・テーブル <i>table-name</i> では、指定されたテーブル内の、すべてのカラムについてレポートします。 column-name ターゲット・カラム <i>column-name</i> では、ターゲット・テーブル内の、指定されたカラムについてレポートします。ターゲット・カラムは複数指定できますが、キーワードは毎回指定する必要があります。 キーワード table および column を指定する必要があります。これらのキーワードでは大文字と小文字が区別されません。 | | | | | | | | |
| 説明 | sp_iqrowdensity はデフォルト・インデックス・レベルでのローの断片化を計測します。密度は、既存のテーブルのローに対してインデックスが必要とする最低限のページ数に対する、実際にインデックスが使用するページ数の割合です。このプロシージャは、密度を $0 < density \leq 1$ のように示します。たとえば、格納領域として最低 8 ページを必要とするインデックスが実際には 10 ページを占有している場合、密度は 0.8 です。 レポートされる密度は、デフォルト・インデックスの再作成または再構成によって再利用可能となるディスク・ページの数を示すものではありません。 このプロシージャは、カラムのロー密度を表示するだけで、それ以上のアクションを推奨するものではありません。インデックスの再作成、再構成、再構築を行うかどうかはユーザが判断する必要があります。 | | | | | | | | |
| 例 | 次のプロシージャは、 <i>SalesOrders</i> テーブル内のカラム <i>ID</i> のロー密度をレポートします。 <pre>sp_iqrowdensity('column group0.SalesOrders.ID')</pre> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Tablename</th> <th style="text-align: left;">ColumnName</th> <th style="text-align: left;">IndexType</th> <th style="text-align: left;">Density</th> </tr> </thead> <tbody> <tr> <td>'GROUPO.SalesOrders'</td> <td>'ID'</td> <td>'Flat style FP'</td> <td>'1.0'</td> </tr> </tbody> </table> | Tablename | ColumnName | IndexType | Density | 'GROUPO.SalesOrders' | 'ID' | 'Flat style FP' | '1.0' |
| Tablename | ColumnName | IndexType | Density | | | | | | |
| 'GROUPO.SalesOrders' | 'ID' | 'Flat style FP' | '1.0' | | | | | | |
| 参照 | 『リファレンス：文とオプション』の「 第2章 データベース・オプション 」の「 FP_LOOKUP_SIZE オプション 」および「 MINIMIZE_STORAGE オプション 」 『システム管理ガイド 第1巻』の「 第6章 Sybase IQ インデックスの使用 」 | | | | | | | | |

sp_iqshowpsexexe プロシージャ

機能 接続のタスクとリソース使用の優先順位を制御するデータベース・オプションの設定に関する情報を表示します。

構文 `sp_iqshowpsexexe [connection-id]`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法 `connection-id` 接続 ID を表す整数。

`connection-id` を指定した場合、`sp_iqshowpsexexe` は指定の接続内のカーソルに関する情報のみを返します。`connection-id` を指定しない場合、`sp_iqshowpsexexe` はすべての接続に関する情報を返します。

指定の `connection-id` が存在しない場合、`sp_iqshowpsexexe` はローを返しません。

説明 `sp_iqshowpsexexe` ストアド・プロシージャは、接続のタスクおよびリソースの使用優先度を管理するデータベース・オプションの設定に関する情報を表示します。これは、データベース管理者がパフォーマンス・チューニングを行うときに役立ちます。

表 7-54 : sp_iqshowpsexexe のカラム

| カラム名 | 説明 |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| connectionid | 接続 ID |
| application | 接続を開いたクライアント・アプリケーションに関する情報。次の AppInfo 接続プロパティ情報を含みます。 HOST : クライアント・マシンのホスト名 EXE : クライアントの実行ファイルの名前 (Windows のみ) APPINFO : クライアント接続文字列の APPINFO (指定した場合) |
| userid | 接続を開いたユーザのログイン名 |
| iqgovern_priority | -iqgovern キュー内で待機する各クエリに優先度を割り当てるデータベース・オプション IQGOVERN_PRIORITY の値。デフォルトでは、このオプションの値は 2 (MEDIUM) です。値 1、2、および 3 はそれぞれ、HIGH、MEDIUM、および LOW として表示されます。 |
| max_query_time | オプティマイザで非常に長いクエリを拒否するための制限を設定するデータベース・オプション MAX_QUERY_TIME の値。デフォルトでは、このオプションは無効であり、値は 0 です。 |

| カラム名 | 説明 |
|------------------------|-------------------------------------------------------------------------------------------------------------------------|
| query_row_limit | データベース・オプション QUERY_ROWS_RETURNED_LIMIT を使用する場合に、結果セットの予測サイズに基づいてクエリを拒否するためのロー・スレッシュホールドを設定する値。デフォルトは0で、制限がないことを示します。 |
| query_temp_space_limit | ユーザ・クエリによりテンポラリ IQ DB 領域の使用を制限するデータベース・オプション QUERY_TEMP_SPACE_LIMIT の値 (メガバイト)。デフォルト値は 2000MB です。 |
| max_cursors | 接続が一度に使用できるカーソルの最大数を制限するためにリソース・ガバナを指定するデータベース・オプション MAX_CURSOR_COUNT の値。デフォルト値は 50 です。値 0 は、制限がないことを示します。 |
| max_statements | 接続が一度に使用できる準備文の最大数を制限するためにリソース・ガバナを指定するデータベース・オプション MAX_STATEMENT_COUNT の値。デフォルト値は 100 です。値 0 は、制限がないことを示します。 |

注意 ApplInfo プロパティは、Interactive SQL (dbisql) の Java バージョンや Sybase Central などの Open Client または jConnect アプリケーションからは使用できない場合があります。ApplInfo プロパティを使用できない場合、application カラムはブランクになります。

例 接続 ID 2 のタスクとリソース使用の優先順位を制御するデータベース・オプションの設定に関する情報を表示します。

```
sp_iqshowpsexec 2
```

```
connectionid      application
                2      HOST=GOODGUY-XP;EXE=C:\Program Files\Sybase\
                IQ-15_1\bin32\dbisqlg.exe;

userid      iqgovern_priority      max_query_time      query_row_limit
DBA          MEDIUM                      0                      0

query_temp_space_limit      max_statements      max_cursors
                2000                      50                      100
```

参照 [第7章 システム・プロシージャの「sp_iqconnection プロシージャ」\(372 ページ\)](#)、[「sp_iqcontext プロシージャ」\(377 ページ\)](#)、および [「sa_conn_info システム・プロシージャ」\(493 ページ\)](#)

「CONNECTION_PROPERTY 関数 [システム]」 (139 ページ)

『リファレンス：文とオプション』の「第2章 データベース・オプション」の「IQGOVERN_MAX_PRIORITY オプション」、
「IQGOVERN_PRIORITY オプション」、
「IQGOVERN_PRIORITY_TIME オプション」、
「MAX_QUERY_TIME オプション」、
「QUERY_ROWS_RETURNED_LIMIT オプション」、
「QUERY_TEMP_SPACE_LIMIT オプション」、
「MAX_CURSOR_COUNT オプション」、および
「MAX_STATEMENT_COUNT オプション」

『システム管理ガイド 第1巻』の「第4章 接続パラメータと通信パラメータ」の「AppInfo 接続パラメータ (App)」

sp_iqspaceinfo プロシージャ

機能 現在のデータベース内の各オブジェクトが使用しているブロック数と、オブジェクトが置かれている DB 領域の名前を表示します。

構文 `sp_iqspaceinfo ['main
| [table table-name | index index-name] [...]']`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 現在のデータベースに関して、オブジェクト名、各オブジェクトが使用するブロック数、DB 領域の名前を表示します。sp_iqspaceinfo にはパラメータは不要です。

sp_iqspaceinfo が返した情報は、DB 領域の管理に役立ちます。

例 iqdemo データベースで実行された sp_iqspaceinfo ストアド・プロシージャの出力を次に示します。この例では、一部のテーブルやインデックスに対する出力は省略されています。

| Name | NBlocks | dbspace_name |
|-----------------------------------------|---------|----------------|
| Contacts | 19 | IQ_SYSTEM_MAIN |
| SalesOrderItems.DBA.ASIQ_IDX_T205_C5_FP | 56 | IQ_SYSTEM_MAIN |
| Contacts.DBA.ASIQ_IDX_T206_C10_FP | 55 | IQ_SYSTEM_MAIN |
| Contacts.DBA.ASIQ_IDX_T206_C1_FP | 61 | IQ_SYSTEM_MAIN |
| ... | | |
| Contacts.DBA.ASIQ_IDX_T206_C9_FP | 55 | IQ_SYSTEM_MAIN |
| Contacts.DBA.ASIQ_IDX_T206_I11_HG | 19 | IQ_SYSTEM_MAIN |
| Customers | 20 | IQ_SYSTEM_MAIN |
| Customers.DBA.ASIQ_IDX_T207_C1_FP | 61 | IQ_SYSTEM_MAIN |

| | | |
|------------------------------------|----|----------------|
| Customers.DBA.ASIQ_IDX_T207_C2_FP | 55 | IQ_SYSTEM_MAIN |
| ... | | |
| Customers.DBA.ASIQ_IDX_T207_I10_HG | 19 | IQ_SYSTEM_MAIN |
| ... | | |

参照 「sp_iqindexinfo プロシージャ」(421 ページ)、 「sp_iqdbspace プロシージャ」(387 ページ)、および「sp_iqdbspaceinfo プロシージャ」(389 ページ)
『システム管理ガイド 第1巻』の「第5章 データベース・オブジェクトの使用」

sp_iqspaceused プロシージャ

機能 IQ ストアおよび IQ テンポラリ・ストアの空き領域と使用領域に関する情報を表示します。

構文 **sp_iqspaceused**(out mainKB unsigned bigint,
out mainKBUsed unsigned bigint,
out tempKB unsigned bigint,
out tempKBUsed unsigned bigint)

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法 sp_iqspaceused は、unsigned bigint の out パラメータとして 4 つの値を返します。このシステム・ストアド・プロシージャは、ユーザ定義のストアド・プロシージャから呼び出し、メインおよびテンポラリ IQ ストアの領域の使用量を確認できます。

説明 sp_iqspaceused は、sp_iqstatus によって提供された情報のサブセットを返しますが、計算に使用する SQL 変数内の情報をユーザが返すこともできます。

表 7-55 : sp_iqspaceused のカラム

| カラム名 | 説明 |
|------------|----------------------------------------|
| mainKB | IQ メイン・ストアの領域の合計 (キロバイト)。 |
| mainKBUsed | データベースが使用している IQ メイン・ストアの領域 (キロバイト)。 |
| tempKB | IQ テンポラリ・ストアの領域の合計 (キロバイト)。 |
| tempKBUsed | データベースが使用している IQ テンポラリ・ストアの領域 (キロバイト)。 |

例 `sp_iqspaceused` では 4 つの出力パラメータが必要です。次の例は、ユーザ定義のストアド・プロシージャ `myspace` を作成するものです。このストアド・プロシージャは、4 つの出力パラメータを宣言し、`sp_iqspaceused` を呼び出します。

```
create procedure dbo.myspace()
begin
  declare mt unsigned bigint;
  declare mu unsigned bigint;
  declare tt unsigned bigint;
  declare tu unsigned bigint;
  call sp_iqspaceused(mt,mu,tt,tu);
  select cast(mt/1024 as unsigned bigint) as mainMB,
         cast(mu/1024 as unsigned bigint) as mainusedMB,
         mu*100/mt as mainPerCent,
         cast(tt/1024 as unsigned bigint) as tempMB,
         cast(tu/1024 as unsigned bigint) as tempusedMB,
         tu*100/tt as tempPerCent;
end
```

`sp_iqspaceused` の出力を表示するには、プロシージャ `myspace` を実行します。

```
myspace
```

sp_iqstatistics プロシージャ

機能 使用可能な各統計または指定の統計について、シリアル番号、名前、説明、値、および単位指定子を返します。

構文 `sp_iqstatistics [stat_name]`

パラメータ **stat_name** (オプション) 統計の名前を指定する VARCHAR パラメータ。

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 `stat_name` を指定すると、`sp_iqstatistics` は、指定の統計について 1 つのローを返すか、または名前が無効な場合はローを返しません。パラメータを指定せずに呼び出すと、`sp_iqstatistics` はすべての統計を返します。

結果セット

| カラム名 | データ型 | 説明 |
|-----------|------------------|-----------|
| stat_num | UNSIGNED INTEGER | 統計のシリアル番号 |
| stat_name | VARCHAR(255) | 統計の名前 |
| stat_desc | VARCHAR(255) | 統計の説明 |

| カラム名 | データ型 | 説明 |
|------------|--------------|-------|
| stat_value | LONG VARCHAR | 統計の値 |
| stat_unit | VARCHAR(128) | 単位指定子 |

次の統計が返されます。

| stat_num | stat_name | stat_desc | stat_unit |
|----------|--------------------------------|----------------------------------------------|-----------|
| 0 | CpuTotalTime | 最後にサーバを起動してから IQ サーバによって消費された合計 CPU 時間 (秒) | Second |
| 1 | CpuUserTime | 最後にサーバを起動してから IQ サーバによって消費された CPU ユーザ時間 (秒) | Second |
| 2 | CpuSystemTime | 最後にサーバを起動してから IQ サーバによって消費された CPU システム時間 (秒) | Second |
| 3 | ThreadsFree | 空き IQ スレッドの数 | N/A |
| 4 | ThreadsInUse | 使用中の IQ スレッドの数 | N/A |
| 5 | MemoryAllocated | 割り付けられたメモリ (メガバイト) | MB |
| 6 | MemoryMaxAllocated | 割り付けられた最大メモリ (メガバイト) | MB |
| 7 | MainCacheCurrentSize | メイン・キャッシュの現在のサイズ (メガバイト) | MB |
| 8 | MainCacheFinds | メイン・キャッシュの検索要求の総数 | N/A |
| 9 | MainCacheHits | メイン・キャッシュのヒットの総数 | N/A |
| 10 | MainCachePagesPinned | メイン・キャッシュの固定ページの数 | Page |
| 11 | MainCachePagesPinnedPercentage | メイン・キャッシュの固定ページの割合 | % |
| 12 | MainCachePagesDirtyPercentage | メイン・キャッシュのダーティ・ページの割合 | % |
| 13 | MainCachePagesInUsePercentage | メイン・キャッシュの使用ページ中の割合 | % |
| 14 | TempCacheCurrentSize | テンポラリ・キャッシュの現在のサイズ (メガバイト) | MB |
| 15 | TempCacheFinds | テンポラリ・キャッシュの検索要求の総数 | N/A |
| 16 | TempCacheHits | テンポラリ・キャッシュのヒットの総数 | N/A |
| 17 | TempCachePagesPinned | テンポラリ・キャッシュの固定ページの数 | Page |
| 18 | TempCachePagesPinnedPercentage | テンポラリ・キャッシュの固定ページの割合 | % |
| 19 | TempCachePagesDirtyPercentage | テンポラリ・キャッシュのダーティ・ページの割合 | % |

| stat_num | stat_name | stat_desc | stat_unit |
|----------|-------------------------------------|----------------------------------|-----------|
| 20 | TempCachePagesInUsePercentage | テンポラリ・キャッシュの使用ページ割合 | % |
| 21 | MainStoreDiskReads | メイン・ストアから読み込まれたデータ量 (キロバイト) | KB |
| 22 | MainStoreDiskWrites | メイン・ストアから書き込まれたデータ量 (キロバイト) | KB |
| 23 | TempStoreDiskReads | メイン・ストアから読み込まれたデータ量 (キロバイト) | KB |
| 24 | TempStoreDiskWrites | メイン・ストアから書き込まれたデータ量 (キロバイト) | KB |
| 25 | ConnectionsTotalConnections | サーバ起動後の総接続数 | N/A |
| 26 | ConnectionsTotalDisconnections | サーバ起動後の総切断数 | N/A |
| 27 | ConnectionsActive | アクティブな接続の数 | N/A |
| 28 | OperationsWaiting | IQ リソース・ガバナを待機している処理の数 | N/A |
| 29 | OperationsActive | IQ リソース・ガバナによって認められたアクティブな同時処理の数 | N/A |
| 30 | OperationsActiveLoadTableStatements | アクティブな LOAD TABLE 文の数 | N/A |

例 **例 1** 1 つの統計 (総 CPU 時間) を表示します。

```
sp_iqstatistics 'CPUTotalTime'
```

例 2 MainCache% のすべての統計を表示します。

```
SELECT * from sp_iqstatistics() WHERE stat_name LIKE 'MainCache%'
```

sp_iqstatus プロシージャ

機能 現在のデータベースについて、さまざまな Sybase IQ ステータス情報を表示します。

構文 `sp_iqstatus`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

現在のデータベースについて、ステータス情報を表示します。データベース名、作成日、ページ・サイズ、DB 領域セグメントの数、ブロック使用状況、バッファ使用状況、I/O、バックアップ情報などが含まれます。

`sp_iqstatus` は、メインおよびテンポラリのストアの領域不足状態を表示します。任意のストアが領域不足状態に陥った場合、`sp_iqstatus` はそのストアの領域不足状態表示欄に Y を表示します。

`sp_iqspaceused` は、`sp_iqstatus` によって提供される情報と同じ情報のサブセットを返しますが、計算に使用する SQL 変数内の情報をユーザが返すこともできます。詳細については、「[sp_iqspaceused プロシージャ](#)」(459 ページ) を参照してください。

接続を切断することによって再利用できる領域を表示するには、`sp_iqstatus` を使用して、返された 2 つのローからの結果を追加します。

```
(DBA)> select * from sp_iqstatus() where name like
'%Versions:%'
Execution time: 6.25 seconds
Name                Value
-----
Other Versions: 2 = 1968Mb
Active Txn Versions: 1 = C:2175Mb/D:2850Mb

(First 2 rows)
```

上記の出力例には、1 つのアクティブな書き込みトランザクションによって 2175MB のデータが作成され、2850MB のデータが破棄されたことが示されています。トランザクションで消費され、まだ解放されていないデータの合計サイズは、4818MB (1968MB + 2850MB = 4818MB) です。

`sp_iqstatus` は、次のチェックポイントで割り付けが解除されるブロックは示しません。ただし、これらのブロックは、`sp_iqdbspace` 出力にタイプ X として表示されます。

例

`sp_iqstatus` ストアド・プロシージャの出力を次に示します。

```
Sybase IQ (TM)                Copyright (c) 1992-2009 by Sybase, Inc.
                                All rights reserved.
Version:                        15.1.0/090416/P/MS/Windows/2003/
                                32bit/2009-04-16 02:11:41
Time Now:                        2009-04-21 13:48:22.319
Build Time:                      2009-04-16 02:15:39
File Format:                      23 on 03/18/1999
Server mode:                      IQ Server
Catalog Format:                    2
Stored Procedure Revision:        1
```

```

Page Size:                               131072/8192blksz/16bpp
Number of Main DB Files :                 2
Main Store Out Of Space:                  N
Number of Temp DB Files :                 1
Temp Store Out Of Space:                  N
DB Blocks: 1-3200                          IQ_SYSTEM_MAIN
DB Blocks: 1045440-1055039                 iq_main
Temp Blocks: 1-1600                       IQ_SYSTEM_TEMP
Create Time:                               2009-04-03 11:30:20.674
Update Time:                              2009-04-03 11:34:33.040
Main IQ Buffers:                           255, 32Mb
Temporary IQ Buffers:                      191, 24Mb
Main IQ Blocks Used:                       5915 of 11200, 52%=46Mb, Max Block#:105278
Temporary IQ Blocks Used:                  65 of 800, 8%=0Mb, Max Block#: 0
Main Reserved Blocks Available:            1600 of 1600, 100%=6Mb
Temporary Reserved Blocks Available:       6400 of 6400, 100%=50Mb
IQ Dynamic Memory:                         Current: 69mb, Max: 70mb
Main IQ Buffers:                           Used: 17, Locked: 0
Temporary IQ Buffers:                      Used: 4, Locked: 0
Main IQ I/O:                               I: L1581/P14 O: C3/D163/P161 D:34 C:97.1
Temporary IQ I/O:                         I: L6627/P0 O: C1086/D1166/P83
                                           D:1082 C:100.0
Other Versions:                            0 = 0Mb
Active Txn Versions:                      0 = C:0Mb/D:0Mb
Last Full Backup ID:                      0
Last Full Backup Time:                    0
Last Backup ID:                           0
Last Backup Type:                         None
Last Backup Time:                         0
DB Updated:                               1
Blocks in next ISF Backup:                 0 Blocks: =0Mb
Blocks in next ISI Backup:                 0 Blocks: =0Mb
DB File Encryption Status:                OFF

```

Main IQ I/O と Temporary IQ I/O の出力コードの意味は、次のとおりです。

- I: 入力
- L: 読み込まれた論理ページ数 (“Finds”)
- P: 読み込まれた物理ページ数
- O: 出力
- C: 作成されたページ数
- D: ダーティ・ページの数
- P: 物理的書き込み
- D: 破損したページ数
- C: 圧縮率

参照

[sp_iqtransaction](#) プロシージャ および [sp_iqversionuse](#) プロシージャ

sp_iqsysmon プロシージャ

機能

Sybase IQ の複数のコンポーネントをモニタします。モニタ対象には、キャッシュ、メモリ、スレッド、ロック、入出力機能、CPU 使用率の管理を含みます。

バッチ・モードでの

構文

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor [, "section(s)"]
or
sp_iqsysmon "time-period" [, "section(s)"]
```

ファイル・モードでの

構文

```
sp_iqsysmon start_monitor, 'filemode' [, "monitor-options"]
sp_iqsysmon stop_monitor
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

バッチ・モードでの

使用法

start_monitor モニタリングを開始します。

stop_monitor モニタリングを停止し、レポートを表示します。

time-period モニタリングの期間。HH:MM:SS の形式で指定してください。

section(s) sp_iqsysmon によって表示される 1 つ以上のセクションの省略形。複数のセクションを指定する場合、セクションの省略形をスペースで区切り、そのリストを一重引用符または二重引用符で囲む必要があります。デフォルトでは、すべてのセクションが表示されます。

IQ ストアに関連するセクションの場合、セクションの省略形に “m” または “t” のプレフィックスを付けることによって、それぞれメイン・ストアまたはテンポラリ・ストアを指定できます。詳細については、[表 7-56](#) を参照してください。プレフィックスを使用しない場合、両方のストアがモニタされます。たとえば、“mbufman” を指定した場合、IQ メイン・ストア・バッファ・マネージャのみがモニタされます。“mbufman tbufman” または “bufman” を指定した場合、メイン・ストアとテンポラリ・ストアの両方のバッファ・マネージャがモニタされます。

表 7-56 : sp_iqsysmon レポート・セクションの省略形

| レポート・セクションまたは IQ コンポーネント | 省略形 |
|--------------------------|---------------|
| バッファ・マネージャ | (m/t)bufman |
| バッファ・プール | (m/t)bufpool |
| プリフェッチ管理 | (m/t)prefetch |
| フリー・リスト管理 | (m/t)freelist |
| バッファ割り付け | (m/t)bufalloc |
| メモリ管理 | memory |
| スレッド管理 | threads |
| CPU の使用率 | cpu |
| トランザクション管理 | txn |
| サーバ・コンテキスト統計 | server |
| カタログの統計 | catalog |

注意 Sybase IQ コンポーネント・ディスク I/O およびロック・マネージャは、現在 sp_iqsysmon ではサポートされていません。

ファイル・モードでの
使用法

start_monitor モニタリングを開始します。

stop_monitor モニタリングを停止し、残りの出力をログ・ファイルに書き込みます。

filemode sp_iqsysmon をファイル・モードで実行することを指定します。ファイル・モードでは、モニタリング期間中一定の時間間隔でサンプリングした統計情報が示されます。デフォルトでは、*dbname.connid-iqmon* という名前のログ・ファイルに出力が書き込まれます。file_suffix オプションを使用して、出力ファイルのサフィックスを変更します。file_suffix オプションの説明については、*monitor_options* パラメータを参照してください。

monitor_options *monitor_options* 文字列には次のオプションを 1 つ以上指定できます。

- **-interval seconds**
レポート間隔を秒単位で指定します。モニタ統計のサンプリング情報が、一定の時間間隔でログ・ファイルに出力されます。**-interval** オプションを指定しない場合のデフォルトは 60 秒間隔です。最小レポート間隔は 2 秒です。このオプションで指定した間隔が無効であるか、2 秒未満である場合、2 秒間隔に設定されます。

最初の記録では、サーバの起動からのカウンタが示されます。それ以降の記録では、前の表示との差が示されます。通常は、パフォーマンスに問題がある期間 (クエリ実行時や特定の時間帯) に 60 秒のデフォルト間隔でモニタを実行すると、意味のある結果を得ることができます。間隔が短すぎると、意味のある結果を取得できないことがあります。ジョブ時間に見合った間隔を指定してください。通常は 60 秒で十分です。

- **-file_suffix suffix**
dbname.connid-suffix という名前のモニタリング出力ファイルを作成します。**-file_suffix** オプションを指定しないと、サフィックスはデフォルトで *iqmon* に設定されます。**-file_suffix** オプションを指定した場合、サフィックスを指定しないか、サフィックスとしてブランクの文字列を指定したときは、サフィックスは使用されません。
- **-append** または **-truncate**
 前者は既存の出力ファイルに追加、後者は既存の出力ファイルをトランケートするよう *sp_iqsysmon* に指示します。デフォルトでは、トランケートされます。両方のオプションを指定した場合、文字列内で後ろに指定されているオプションが有効になります。
- **-section section(s)**
 モニタ・ログ・ファイルに書き込む 1 つ以上のセクションの省略形を指定します。デフォルトでは、すべてのセクションが書き込まれます。ファイル・モードのセクション・リストで指定する省略形は、バッチ・モードで使用する省略形と同じです。省略形のリストについては、[表 7-56](#) を参照してください。複数のセクションを指定する場合、セクションの省略形をスペースで区切る必要があります。
 セクションなしで **-section** オプションを指定した場合、どのセクションもモニタされません。無効なセクション省略形は無視され、IQ メッセージ・ファイル内に警告が示されます。

使用構文の例

表 7-57 : *sp_iqsysmon* の使用例

| 構文 | 結果 |
|-------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <i>sp_iqsysmon start_monitor</i> <i>sp_iqsysmon stop_monitor</i> | バッチ・モードでモニタを開始し、メイン・ストアおよびテンポラリ・ストアのすべてのセクションを表示します。 |
| <i>sp_iqsysmon start_monitor</i> <i>sp_iqsysmon stop_monitor</i> "mbufman mbufpool" | バッチ・モードでモニタを開始し、メイン・ストアのバッファ・マネージャとバッファ・プールの統計情報を表示します。 |

| 構文 | 結果 |
|--------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| <code>sp_iqsysmon "00:00:10", "mbufpool memory"</code> | バッチ・モードでモニタを 10 秒間実行し、10 秒間の経過後に集約した統計情報を表示します。 |
| <code>sp_iqsysmon start_monitor, 'filemode', "-interval 5 -sections mbufpool memory"</code> <code>sp_iqsysmon stop_monitor</code> | ファイル・モードでモニタを開始し、メイン・バッファ・プールとメモリ・マネージャの統計情報を 5 秒間隔でログ・ファイルに書き込みます。 |

説明

`sp_iqsysmon` ストアド・プロシージャは、Sybase IQ の複数のコンポーネントをモニタします。モニタ対象には、キャッシュ、メモリ、スレッド、ロック、入出力機能、CPU の使用率の管理を含みます。

`sp_iqsysmon` プロシージャは、次の 2 つのモニタリング・モードをサポートします。

- バッチ・モード

バッチ・モードでは、`sp_iqsysmon` はモニタを開始してから停止するまでの期間、または *time-period* パラメータで指定した期間、モニタ統計情報を収集します。モニタリング期間の経過後、`sp_iqsysmon` は集約した統計情報のリストを表示します。

バッチ・モードの `sp_iqsysmon` は、Adaptive Server Enterprise プロシージャ `sp_sysmon` と同様のものです。

- ファイル・モード

ファイル・モードでは、`sp_iqsysmon` はモニタを開始してから停止するまで、一定の時間間隔でサンプリングした統計情報をログ・ファイルに書き込みます。

ファイル・モードの最初の記録では、サーバの起動からのカウンタが示されます。それ以降の記録では、前の表示との差が示されます。

ファイル・モードの `sp_iqsysmon` は、IQ UTILITIES コマンド `START MONITOR` および `STOP MONITOR` インタフェースに似ています。

バッチ・モードの例

10 分後にモニタ情報を出力します。

```
sp_iqsysmon "00:10:00"
```

5 分後に、`sp_iqsysmon` レポートのメモリ・マネージャのセクションのみを出力します。

```
sp_iqsysmon "00:05:00", memory
```

モニタを開始した後、2つのプロシージャと1つのクエリを実行し、モニタを停止して、レポートのバッファ・マネージャのセクションのみを出力します。

```
sp_iqsysmon start_monitor
go
execute procl
go
execute proc2
go
select sum(total_sales) from titles
go
sp_iqsysmon stop_monitor, bufman
go
```

20分後に、レポートのメイン・バッファ・マネージャおよびメイン・バッファ・プールのセクションのみを出力します。

```
sp_iqsysmon "00:02:00", "mbufman mbufpool"
```

ファイル・モードの例

モニタを開始してから停止するまで、2秒ごとに情報をトランケートし、ログ・ファイルに書き込みます。

```
sp_iqsysmon start_monitor, 'filemode', '-interval 2'
.
.
.
sp_iqsysmon stop_monitor
```

dbname.connid-testmon という名前の ASCII ファイルに対して、メイン・バッファ・マネージャおよびメモリ・マネージャのセクションのみの出力を追加します。データベース *iqdemo* について、ファイル *iqdemo.2-testmon* に結果を書き込みます。

```
sp_iqsysmon start_monitor, 'filemode',
"-file_suffix testmon -append -section mbufman memory"
.
.
.
sp_iqsysmon stop_monitor
```

例

バッチ・モードでモニタを 10 秒間実行し、10 秒間の経過後に集約した統計情報を表示します。

```
sp_iqsysmon "00:00:10", "mbufpool memory"
```

```
=====
Buffer Pool (Main)
=====
```

| STATS-NAME | TOTAL | NONE | BTREEV | BTREEF | BV | VDO | DBEXT | DBID | SORT |
|-------------------|-------|------|--------|--------|----|-----|-------|------|------|
| MovedToMRU | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MovedToWash | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RemovedFromLRU | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RemovedFromWash | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RemovedInScanMode | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| STORE | GARRAY | BARRAY | BLKMAP | HASH | CKPT | BM | TEST | CMID | RIDCA | LOB |
|-------|--------|--------|--------|------|------|----|------|------|-------|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| STATS-NAME | VALUE |
|---------------------|----------------|
| Pages | 127 (100.0 %) |
| InUse | 4 (3.1 %) |
| Dirty | 1 (0.8 %) |
| Pinned | 0 (0.0 %) |
| Flushes | 0 |
| FlushedBufferCount | 0 |
| GetPageFrame | 0 |
| GetPageFrameFailure | 0 |
| GotEmptyFrame | 0 |
| Washed | 0 |
| TimesSweepersWoken | 0 |
| washTeamSize | 0 |
| WashMaxSize | 26 (20.5 %) |
| washNBuffers | 4 (3.1 %) |
| washNDirtyBuffers | 1 (0.8 %) |
| washSignalThreshold | 3 (2.4 %) |
| washNActiveSweepers | 0 |
| washIntensity | 1 |

=====
 Memory Manager
 =====

| STATS-NAME | VALUE |
|-------------------|----------------------|
| MemAllocated | 43616536 (42594 KB) |
| MemAllocatedMax | 43735080 (42710 KB) |
| MemAllocatedEver | 0 (0 KB) |
| MemNAllocated | 67079 |
| MemNAllocatedEver | 0 |
| MemNTimesLocked | 0 |
| MemNTimesWaited | 0 (0.0 %) |

- 参照 『リファレンス：文とオプション』の「第1章 SQL 文」の「IQ UTILITIES 文」
『パフォーマンス&チューニング・ガイド』の「第5章 パフォーマンスのモニタリングとチューニング」

sp_iqtable プロシージャ

- 機能** データベース内のテーブルに関する情報を表示します。
- 構文 1** `sp_iqtable ([table_name],[table_owner],[table_type])`
- 構文 2** `sp_iqtable [table_name='tablename'],
[table_owner='tableowner'],[table_type='tabletype']`
- パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。
- 使用法** **構文 1** 最初の2つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に NULL を指定する必要があります。たとえば、`sp_iqtable NULL,NULL,TEMP` や `sp_iqtable NULL,dbo,SYSTEM` のように指定します。

注意 `table_type` の値 ALL および VIEW は、構文 1 では一重引用符で囲む必要があります。

構文 2 パラメータはどのような順番で指定しても構いません。一重引用符で囲みます。

表 7-58 に `table_type` パラメータに指定できる値を示します。

表 7-58 : sp_iqtable の table_type の値

| table_type の値 | 表示される情報 |
|---------------|-----------------------|
| SYSTEM | システム・テーブル |
| TEMP | グローバル・テンポラリ・テーブル |
| VIEW | ビュー |
| ALL | IQ テーブル、システム・テーブル、ビュー |
| その他の値 | IQ テーブル |

説明

いずれかのパラメータを指定すると、そのパラメータに合致するテーブルだけが返されます。複数のパラメータを指定すると、指定されたすべてのパラメータにより結果がフィルタされます。パラメータを指定しない場合は、データベース内のすべての Sybase IQ テーブルが返されます。ローカル・テンポラリ・テーブルの名前を返す方法はありません。

表 7-59 : sp_iqtable のカラム

| カラム名 | 説明 |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| table_name | テーブル名。 |
| table_type | BASE - ベース・テーブル。 MAT VIEW - マテリアライズド・ビュー。 GBL TEMP - グローバル・テンポラリ・テーブル。 PARTITION - テーブル・パーティション(このテーブルは内部でのみ使用可能であり、Sybase IQ ユーザが使用することはできない)。 VIEW - ビュー。 JVT - ジョイン仮想テーブル。 |
| table_owner | テーブルの所有者。 |
| server_type | IQ - IQ ストアで作成されたオブジェクト。 SA - SA ストアで作成されたオブジェクト。 ビューはすべて SA ストアで作成されます。 |
| location | TEMP - IQ テンポラリ・ストア。 MAIN - IQ ストア。 SYSTEM - カタログ・ストア。 |
| dbspace_id | テーブルが存在する DB 領域の名前。 |
| isPartitioned | カラムが、分割されたテーブルに属しており、かつテーブル・パーティションの DB 領域と異なる DB 領域を持つ 1 つ以上のパーティションを持っている場合は 'Y'。カラムのテーブルが分割されていないか、またはカラムの各パーティションがテーブル・パーティションと同じ DB 領域に存在する場合は 'N'。 |
| remarks | COMMENT 文で追加されたユーザ・コメント。 |
| table_constraints | テーブルに対する制約。 |

例

次の構文は、いずれもテーブル Departments に関する情報を返します。

```
sp_iqtable ('Departments')
sp_iqtable table_name='Departments'
```

| Table_name | Table_type | Table_owner | Server_type | location |
|-------------|------------|-------------|-------------|----------|
| Departments | BASE | GRUPO | IQ | Main |

| dbspace_id | isPartitioned | Remarks | table_constraints |
|------------|---------------|---------|-------------------|
| 16387 | N | (NULL) | (NULL) |

次の構文は、いずれも **GRUPO** が所有するすべてのテーブルを返します。

```
sp_iqtable NULL,GRUPO
sp_iqtable table_owner='GRUPO'
```

| Table_name | Table_type | Table_owner | Server_type | location |
|-----------------|------------|-------------|-------------|----------|
| Contacts | BASE | GRUPO | IQ | Main |
| Customers | BASE | GRUPO | IQ | Main |
| Departments | BASE | GRUPO | IQ | Main |
| Employees | BASE | GRUPO | IQ | Main |
| FinancialCodes | BASE | GRUPO | IQ | Main |
| FinancialData | BASE | GRUPO | IQ | Main |
| Products | BASE | GRUPO | IQ | Main |
| SalesOrders | BASE | GRUPO | IQ | Main |
| SalesOrderItems | BASE | GRUPO | IQ | Main |

| dbspace_id | isPartitioned | Remarks | table_constraints |
|------------|---------------|---------|-------------------|
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |
| 16387 | N | (NULL) | (NULL) |

sp_iqtablesizer プロシージャ

機能

指定したテーブルのサイズを返します。

構文 `sp_iqtablesiz (table_owner.table_name)`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 テーブルの合計サイズを、ブロック、キロバイト、Nblocks (IQ ブロック) で返します。また、メモリにテーブルを保持するために必要なページ数と、(ディスク上で) テーブルを圧縮した場合に圧縮された IQ ページ数も返します。このプロシージャには、必ず `table_name` パラメータを指定してください。`table_name` の所有者がこれを実行する場合は、`table_owner` パラメータを指定する必要はありません。

表 7-60 : sp_iqtablesiz のコラム

| コラム名 | 説明 |
|-----------------|-------------------------------------|
| Ownername | 所有者の名前 |
| Tablename | テーブルの名前 |
| Columns | テーブルのコラム数 |
| KBytes | 物理テーブル・サイズ (キロバイト) |
| Pages | メモリ内にテーブルを保持するために必要な IQ ページの数 |
| CompressedPages | (ディスク上で) テーブルを圧縮した場合に圧縮される IQ ページの数 |
| NBlocks | IQ ブロックの数 |

Pages は、テーブルの IQ ページの合計数です。ページの測定単位は、IQ ページ・サイズです。すべてのメモリ内バッファ (IQ バッファ・キャッシュ内のバッファ) は同じサイズです。

ディスク上の IQ ページは圧縮されています。ディスク上の各 IQ ページは、1 から 16 ブロックを使用します。IQ ページ・サイズが 128KB の場合、IQ ブロック・サイズは 8KB です。この場合、ディスク上の個別のページは 8、16、24、32、40、48、56、64、72、80、88、96、104、112、120、128 KB のいずれかです。

KBytes の値をページ・サイズで除算すると、ディスク上のページ・サイズの平均がわかります。

注意 Sybase IQ は常に、ブロックではなくページ全体を読み書きします。たとえば、個別のページが 88K まで圧縮された場合、IQ は 1 回の I/O で 88K を読み書きできます。平均的なページは、3 分の 1 から 2 分の 1 に圧縮されます。

NBlocks は、**Kbytes** を IQ ブロック・サイズで除算したものです。

CompressedPages は、圧縮されたページの数です。たとえば、Pages が 1000 で CompressedPages が 992 である場合、1000 ページ中 992 ページが圧縮されたことになります。大部分のページは圧縮されるため、CompressedPages を Pages で割った結果は、通常はほぼ 100% になります。Sybase IQ は空のページを書き込まないため、空のページは圧縮されません。IQ ページは、ページの満杯度にかかわらず、高い圧縮率で圧縮されます。

例 `call sp_iqtablesize ('dba.emp1')`

| Ownername | Tablename | Columns | KBytes | Pages | CompressedPages | NBlocks |
|-----------|-----------|---------|--------|-------|-----------------|---------|
| DBA | emp1 | 4 | 1504 | 24 | 14 | 188 |

sp_iqtableuse プロシージャ

機能 負荷によってアクセスされるテーブルの使用状況の情報を詳細にレポートします。

構文 `sp_iqtableuse`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 SYSTEM で作成されたテーブルはレポートされません。

表 7-61 : sp_iqtableuse のカラム

| カラム名 | 説明 |
|-----------|---------------|
| TableName | テーブル名 |
| Owner | テーブル所有者のユーザ名 |
| UID** | テーブルのユニークな識別子 |
| LastDT | 前回のアクセスの日時 |
| NRef | クエリ参照の数 |

**UID はシステムが割り当てた番号であり、テーブルのインスタンスをユニークに識別します (インスタンスはオブジェクト作成時に定義されます)。

参照 『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「負荷のモニタリング」

「sp_iqcolumnuse プロシージャ」(371 ページ)、 「sp_iqdbspace プロシージャ」(387 ページ)、 「sp_iqindexadvice プロシージャ」(418 ページ)、 「sp_iqindexuse プロシージャ」(426 ページ)、 「sp_iqunusedcolumn プロシージャ」(479 ページ)、 「sp_iqunusedindex プロシージャ」(481 ページ)、 「sp_iqunusedtable プロシージャ」(482 ページ)、 および 「sp_iqworkmon プロシージャ」(489 ページ)

『リファレンス：文とオプション』の「第 2 章 データベース・オプション」の「INDEX_ADVISOR オプション」

sp_iqtransaction プロシージャ

機能 トランザクションとバージョンに関する情報を表示します。

構文 `sp_iqtransaction`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 sp_iqtransaction は、Sybase IQ トランザクション・マネージャ内の各トランザクション制御ブロックのローを返します。カラム Name、Userid、ConnHandle はそれぞれ、接続プロパティ Name、Userid、Number です。ローは TxnID の順に並べられます。

sp_iqtransaction の出力には、実行中のトランザクションを持たない接続に対応するローは含まれません。すべての接続を表示するには、sp_iqconnection を使用します。

注意 sp_iqtransaction を使用して、他のユーザによるテーブルへの書き込みをブロックしているユーザを確認することもできますが、この場合は sp_iqllocks を使用することをおすすめします。

表 7-62 : sp_iqtransaction のカラム

| カラム名 | 説明 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | アプリケーション名。 |
| Userid | 接続のユーザ ID。 |
| TxnID | このトランザクション制御ブロックのトランザクション ID。トランザクション ID は、トランザクションの開始中に割り当てられます。これは、BeginTxn、CmtTxn、および PostCmtTxn メッセージによって .iqmsg ファイルに表示されるトランザクション ID、また、データベースが開かれたときにログ記録される Txn ID Seq と同じです。 |

| カラム名 | 説明 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CmtID | トランザクションがコミットしたときにトランザクション・マネージャによって割り当てられる ID。アクティブなトランザクションでは 0 です。 |
| VersionID | シンプレックス・データベースでは、VersionID は TxnID と同一です。マルチプレックス・コーディネータの場合、VersionID はアクティブ・トランザクションの TxnID と同じであり、VersionID はコミットされたトランザクションの CmtID と同じです。マルチプレックス・サカンダリ・サーバでは、VersionID は、マルチプレックス・コーディネータでデータベース・バージョンを作成したトランザクションの CmtID です。これは、マルチプレックス・データベース内のすべてのノードに対してデータベース・バージョンを一意に識別するために、Sybase IQ のメモリ内カタログと IQ トランザクション・マネージャで使用されます。 |
| State | トランザクション制御ブロックの状態。この変数は内部の Sybase IQ 実装の詳細を反映するもので、将来的に変更される可能性があります。本稿の執筆時点では、トランザクションの状態は NONE、ACTIVE、ROLLING_BACK、ROLLED_BACK、COMMITTING、COMMITTED、APPLIED です。 |
| ConnHandle | 接続の ID 番号。 |
| IQConnID | .iqmsg ファイル内のすべてのメッセージに表示される 10 桁の接続 ID。これは、サーバ・セッション内でユニークな、単純増加する整数です。 |
| MainTableKBCr | このトランザクションによって作成された IQ ストアの領域 (キロバイト)。 |
| MainTableKBDr | このトランザクションによって削除されたが、他のデータベース・バージョン、またはこのトランザクションの他のセーブポイントでこの領域が表示されているためにストア上のディスクに残っている IQ ストアの領域 (キロバイト)。 |
| TempTableKBCr | このトランザクションが IQ テンポラリー・テーブルのデータの格納用に作成した IQ テンポラリー・ストアの領域 (キロバイト)。 |
| TempTableKBDr | このトランザクションによって削除されたが、IQ カーソルで表示されている、またはこのトランザクションの他のセーブポイントが所有しているためにテンポラリー IQ ストア上のディスクに残っている IQ テンポラリー・テーブルの領域 (キロバイト)。 |
| TempWorkSpaceKB | ACTIVE なトランザクションでは、これは、このトランザクションが使用中のワークスペースのスナップショットである (ソート、ハッシュ、テンポラリー・ビットマップなど)。この数字は、sp_iqtransaction を実行するタイミングによって変わります。たとえば、クエリ・エンジンがテンポラリー・キャッシュに 60MB を作成し、クエリ処理が継続中でも、その大部分をすぐに解放することがあります。したがって、クエリが完了した後に sp_iqtransaction を実行すると、このカラムの数字がずっと小さくなっていることがあります。トランザクションがアクティブでなくなると、このカラムは 0 になります。 ACTIVE なトランザクションでは、このカラムは sp_iqconnection の TempWorkSpaceKB カラムと同一です。 |

| カラム名 | 説明 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TxnCreateTime | トランザクションが開始した時間。すべての Sybase IQ トランザクションは、アクティブな接続が確立されてすぐに、または前のトランザクションがコミットまたはロールバックしたときに暗黙的に始まります。 |
| CursorCount | このトランザクション制御ブロックを参照している、オープンしている Sybase IQ カーソルの数。トランザクションが ACTIVE である場合、トランザクション内で作成されたオープンしているカーソルの数を示します。トランザクションが COMMITTED の場合は、このトランザクション制御ブロックが所有するデータベース・バージョンを参照する HOLD カーソルの数を示します。 |
| SpCount | トランザクション制御ブロック内に存在する、セーブポイント構造の数。セーブポイントは、暗黙的に作成および開放されます。したがって、この番号はトランザクション内のユーザ作成セーブポイントの数を示すものではありません。 |
| SpNumber | トランザクションの、アクティブなセーブポイントの数。これは実装の詳細を反映するものであり、ユーザ作成のセーブポイントは反映しません。 |
| MPXServerName | アクティブ・トランザクションがノード間通信 (INC) 接続からのものであるかどうかを示す値。INC 接続からの場合、値は、トランザクションが開始するマルチプレックス・サーバの名前となります。INC 接続からでない場合は NULL です。トランザクションがアクティブでない場合は常に NULL です。 |
| GlobalTxnID | 現在のトランザクションに関連付けられているグローバル・トランザクション ID を示す値。グローバル・トランザクションに何も関連付けられていない場合は 0 です。 |

例 `sp_iqtransaction` の出力の例を次に示します。

| Name | Userid | TxnID | CmtID | VersionID | State | ConnHandle | IQConnID |
|---------|--------|-------|-------|-----------|-----------|------------|----------|
| red2 | DBA | 10058 | 10700 | 10058 | COMMITTED | 419740283 | 14 |
| blue1 | DBA | 10568 | 0 | 10568 | ACTIVE | 640038605 | 17 |
| | DBA | 10604 | 0 | 10604 | ACTIVE | 2094200996 | 18 |
| fromSCJ | DBA | 10619 | 0 | 10619 | ACTIVE | 954498130 | 20 |
| blue2 | DBA | 10634 | 10677 | 10634 | COMMITTED | 167015670 | 21 |
| ntJava2 | DBA | 10676 | 0 | 10676 | ACTIVE | 1779741471 | 24 |
| blue2 | DBA | 10678 | 0 | 10678 | ACTIVE | 167015670 | 21 |
| nt1 | DBA | 10699 | 0 | 10699 | ACTIVE | 710225777 | 28 |
| red2 | DBA | 10701 | 0 | 10701 | ACTIVE | 419740283 | 14 |
| | DBA | 16687 | 0 | 16687 | ACTIVE | 1306718536 | 23 |

| MainTableKBCr | MainTableKBDr | TempTableKBCr | TempTableKBDr |
|---------------|---------------|---------------|---------------|
| 0 | 0 | 65824 | 0 |
| 0 | 0 | 0 | 0 |

| | | | |
|------|------|------|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 3960 | 152 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 2400 | 1992 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 2912 | 22096 |
| 0 | 0 | 0 | 0 |

| TempWorkSpaceKB | TxnCreateTime | CursorCount | SpCount | SpNumber |
|-----------------|-------------------------|-------------|---------|----------|
| ===== | ===== | ===== | ===== | ===== |
| 0 | 2009-06-26 13:17:27.612 | 1 | 3 | 2 |
| 102592 | 2009-06-26 13:27:28.491 | 1 | 1 | 0 |
| 0 | 2009-06-26 13:30:27.548 | 0 | 1 | 0 |
| 0 | 2009-06-26 13:31:27.151 | 0 | 24 | 262 |
| 0 | 2009-06-26 13:35:02.128 | 0 | 0 | 0 |
| 0 | 2009-06-26 13:43:58.805 | 0 | 39 | 408 |
| 128 | 2009-06-26 13:45:28.379 | 0 | 1 | 0 |
| 0 | 2009-06-26 14:05:15.759 | 0 | 42 | 413 |
| 680 | 2009-06-26 14:57:51.104 | 1 | 2 | 20 |
| 0 | 2009-06-26 15:09:30.319 | 0 | 1 | 0 |

| MPXServerName | GlobalTxnID |
|---------------|-------------|
| ===== | ===== |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |
| (NULL) | 0 |

sp_iqunusedcolumn プロシージャ

機能 負荷によって参照されなかった IQ カラムをレポートします。

構文 `sp_iqunusedcolumn`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 SYSTEM で作成されたテーブルまたはローカル・テンポラリ・テーブルのカラムはレポートされません。

表 7-63 : sp_iqunusedcolumn のカラム

| カラム名 | 説明 |
|------------|-------------|
| TableName | テーブル名 |
| ColumnName | カラム名 |
| Owner | カラム所有者のユーザ名 |

例 sp_iqunusedcolumn プロシージャからの出力例を示します。

| TableName | ColumnName | Owner |
|-----------------|---------------------|--------|
| SalesOrders | ID | GROUP0 |
| SalesOrders | CustomerID | GROUP0 |
| SalesOrders | OrderDate | GROUP0 |
| SalesOrders | FinancialCode | GROUP0 |
| SalesOrders | Region | GROUP0 |
| SalesOrders | SalesRepresentative | GROUP0 |
| SalesOrderItems | ID | GROUP0 |
| SalesOrderItems | LineID | GROUP0 |
| SalesOrderItems | ProductID | GROUP0 |
| SalesOrderItems | Quantity | GROUP0 |
| SalesOrderItems | ShipDate | GROUP0 |
| Contacts | ID | GROUP0 |
| Contacts | Surname | GROUP0 |
| Contacts | GivenName | GROUP0 |
| ... | | |

参照 『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「負荷のモニタリング」

「sp_iqcolumnuse プロシージャ」(371 ページ)、[「sp_iqdbspace プロシージャ」](#)(387 ページ)、[「sp_iqindexadvice プロシージャ」](#)(418 ページ)、[「sp_iqindexuse プロシージャ」](#)(426 ページ)、[「sp_iqtableuse プロシージャ」](#)(475 ページ)、[「sp_iqunusedindex プロシージャ」](#)(481 ページ)、[「sp_iqunusedtable プロシージャ」](#)(482 ページ)、および [「sp_iqworkmon プロシージャ」](#)(489 ページ)

『リファレンス: 文とオプション』の「第2章 データベース・オプション」の「[INDEX_ADVISOR オプション](#)」

sp_iqunusedindex プロシージャ

機能 負荷によって参照されなかった IQ の二次的な (非 FP) インデックスをレポートします。

構文 `sp_iqunusedindex`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 SYSTEM で作成されたテーブルまたはローカル・テンポラリ・テーブルのインデックスはレポートされません。

表 7-64 : sp_iqunusedindex のカラム

| カラム名 | 説明 |
|-----------|----------------|
| IndexName | インデックス名 |
| TableName | テーブル名 |
| Owner | インデックス所有者のユーザ名 |
| IndexType | インデックス・タイプ |

例 sp_iqunusedindex プロシージャからの出力例を示します。

| IndexName | TableName | Owner | IndexType |
|----------------------|-----------------|--------|-----------|
| ASIQ_IDX_T450_I7_HG | SalesOrders | GROUP0 | HG |
| ASIQ_IDX_T450_C6_HG | SalesOrders | GROUP0 | HG |
| ASIQ_IDX_T450_C4_HG | SalesOrders | GROUP0 | HG |
| ASIQ_IDX_T450_C2_HG | SalesOrders | GROUP0 | HG |
| ASIQ_IDX_T451_I6_HG | SalesOrderItems | GROUP0 | HG |
| ASIQ_IDX_T451_C3_HG | SalesOrderItems | GROUP0 | HG |
| ASIQ_IDX_T451_C1_HG | SalesOrderItems | GROUP0 | HG |
| ASIQ_IDX_T452_I11_HG | Contacts | GROUP0 | HG |
| ASIQ_IDX_T453_I10_HG | Contacts | GROUP0 | HG |
| ASIQ_IDX_T454_I4_HG | FinancialCodes | GROUP0 | HG |
| ASIQ_IDX_T455_I5_HG | FinancialData | GROUP0 | HG |
| ASIQ_IDX_T455_C3_HG | FinancialData | GROUP0 | HG |
| ASIQ_IDX_T456_I8_HG | Products | GROUP0 | HG |
| ASIQ_IDX_T457_I4_HG | Departments | GROUP0 | HG |
| ASIQ_IDX_T457_C3_HG | Departments | GROUP0 | HG |
| ASIQ_IDX_T458_I21_HG | Departments | GROUP0 | HG |
| ASIQ_IDX_T458_C5_HG | Departments | GROUP0 | HG |

参照 『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」の「[負荷のモニタリング](#)」

「[sp_iqcolumnuse プロシージャ](#)」(371 ページ)、[「sp_iqdbspace プロシージャ](#)」(387 ページ)、[「sp_iqindexadvice プロシージャ](#)」(418 ページ)、[「sp_iqindexuse プロシージャ](#)」(426 ページ)、[「sp_iqtableuse プロシージャ](#)」(475 ページ)、[「sp_iqunusedcolumn プロシージャ](#)」(479 ページ)、[「sp_iqunusedtable プロシージャ](#)」(482 ページ)、および [「sp_iqworkmon プロシージャ](#)」(489 ページ)

『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」の「[INDEX_ADVISOR オプション](#)」

sp_iqunusedtable プロシージャ

機能 負荷によって参照されなかった IQ テーブルをレポートします。

構文 `sp_iqunusedtable`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 SYSTEM で作成されたテーブルとローカル・テンポラリ・テーブルはレポートされません。

表 7-65 : sp_iqunusedtable のカラム

| カラム名 | 説明 |
|-----------|--------------|
| TableName | テーブル名 |
| Owner | テーブル所有者のユーザ名 |

例 次の表は sp_iqunusedtable プロシージャからの出力例です。

| TableName | Owner |
|-----------------|--------|
| FinancialCodes | GROUP0 |
| Contacts | GROUP0 |
| Employees | GROUP0 |
| empl | DBA |
| SalesOrders | GROUP0 |
| FinancialData | GROUP0 |
| Departments | GROUP0 |
| SalesOrderItems | GROUP0 |
| Products | GROUP |
| iq_dummy | DBA |
| Customers | GROUP0 |
| sale | DBA |

- 参照 『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「負荷のモニタリング」
- 「sp_iqcolumnuse プロシージャ」(371 ページ)、「sp_iqdbspace プロシージャ」(387 ページ)、「sp_iqindexadvice プロシージャ」(418 ページ)、「sp_iqindexuse プロシージャ」(426 ページ)、「sp_iqtableuse プロシージャ」(475 ページ)、「sp_iqunusedcolumn プロシージャ」(479 ページ)、「sp_iqunusedindex プロシージャ」(481 ページ)、および「sp_iqworkmon プロシージャ」(489 ページ)
- 『リファレンス : 文とオプション』の「第2章 データベース・オプション」の「INDEX_ADVISOR オプション」

sp_iqversionuse プロシージャ

- 機能** IQ メイン・ストアで使用されているバージョンを表示します。
- 構文** `sp_iqversionuse`
- パーミッション** DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。
- 説明** `sp_iqversionuse` システム・ストアド・プロシージャは、テーブルのバージョンが複数あるためにデータベースが記憶領域を過度に使用するという問題のある状況に対するトラブルシューティングに役立ちます。
- 領域不足の状況が発生している場合、またはマルチプレックス・サーバで `sp_iqstatus` によるメイン・ブロックの使用が高いパーセンテージを示している場合は、`sp_iqversionuse` を実行して、使用されているバージョン、およびそのバージョンでリカバリできる領域の量を確認します。マルチプレックス機能の詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。
- 領域の大きさは範囲として表されます。これは、一般的に他にどのバージョンが解放されたかによって実際の大きさが異なるためです。解放される実際の領域の大きさは、MinKBRelease ~ MaxKBRelease の範囲です。最も古いバージョンでは必ず、MaxKBRelease と MinKBRelease が一致します。

WasReported は、バージョン使用情報がセカンダリ・サーバからコーディネータに送信されたかどうかを示します。新しいバージョンでは、コーディネータ上での WasReported の初期値は 0 です。SQL Remote がコーディネータにバージョン使用情報をレプリケートすると、WasReported が 1 になります。WasReported が長期間にわたり 0 である場合、SQL Remote が停止している可能性があります。

注意 WasReported カラムは、マルチプレックス設定で使用されます。マルチプレックスの詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。

表 7-66 : sp_iqversionuse のカラム

| カラム名 | 説明 |
|--------------|----------------------------------|
| VersionID | バージョン識別子 |
| Server | このバージョンのユーザが接続するサーバ |
| IQConnID | このバージョンを使用する接続 ID |
| WasReported | このバージョンの使用情報をサーバが受信したかどうかを示します |
| MinKBRelease | このバージョンが使用されなくなったときに返される領域の最小サイズ |
| MaxKBRelease | このバージョンが使用されなくなったときに返される領域の最大サイズ |

例 次の表は sp_iqversionuse システム・プロシージャからの出力例です。

```

VersionID Server          IQConnID WasReported
=====  =====
          0 ab2ab_iqdemo          9          0

MinKBRelease MaxKBRelease
=====
          0          0

```

sp_iqview プロシージャ

機能 テーブル内のビューに関する情報を表示します。

構文 1 **sp_iqview** ([view_name],[view_owner],[view_type])

構文 2 **sp_iqview** [view_name='viewname'],
[view_owner='viewowner'],[view_type='viewtype']

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法

構文 1 `sp_iqview NULL,NULL,SYSTEM` 最初の2つのパラメータのいずれかを指定せずに、その次のパラメータを指定する場合は、指定しないパラメータの位置に NULL を指定する必要があります。たとえば、`sp_iqview NULL,NULL,SYSTEM` や `sp_iqview deptview,NULL,'ALL'` のように指定します。

注意 `view_type` の値 ALL は、構文 1 では一重引用符で囲む必要があります。

構文 2 パラメータはどのような順番で指定しても構いません。一重引用符で囲みます。

`view_type` パラメータに指定できる値を表 7-67 に示します。

表 7-67 : sp_iqview の view_type の値

| view_type の値 | 表示される情報 |
|--------------|------------------|
| SYSTEM | システム・ビュー |
| ALL | ユーザ・ビューとシステム・ビュー |
| その他の値 | ユーザ・ビュー |

説明

いずれかのパラメータを指定すると、指定されたビュー名のビュー、または指定されたユーザが所有するビューのみが返されます。複数のパラメータを指定すると、指定されたすべてのパラメータにより結果がフィルタされます。パラメータを指定しない場合は、データベース内のすべてのユーザ・ビューが返されます。

表 7-68 : sp_iqview のカラム

| カラム名 | 説明 |
|-------------------------|---------------------------|
| <code>view_name</code> | ビューの名前 |
| <code>view_owner</code> | ビューの所有者 |
| <code>view_def</code> | CREATE VIEW 文で指定されたビューの定義 |
| <code>remarks</code> | COMMENT 文で追加されたユーザ・コメント |

`sp_iqview` は、32K 文字より大きいビュー定義をトランケーションなしで返します。

例

次の構文は、いずれもビュー `deptview` に関する情報を返します。

```
call sp_iqview('ViewSalesOrders')
sp_iqview view_name='ViewSalesOrders'
```

次の構文は、いずれも **GROUPO** が所有するすべてのビューを返します。

```
sp_iqview NULL, GROUPO
sp_iqview view_owner='GROUPO'
```

| view_name | view_owner | view_def | remarks |
|-----------------|------------|------------------------------------------------------------------------------------------------------------------------------------------|---------|
| ViewSalesOrders | GROUPO | Create views GROUPO , ViewSalesOrders(ID, LineID, ProductID, Quantity, OrderDate, ShipDate, Region, SalesRepresentativeName | (NULL) |

sp_iqwho プロシージャ

機能 現在のすべてのユーザと接続に関する情報、または特定のユーザまたは接続に関する情報を表示します。

構文 `sp_iqwho [{ connhandle | user-name } [, arg-type]]`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 `sp_iqwho` ストアド・プロシージャは、現在のすべてのユーザと接続に関する情報、または特定のユーザまたは接続に関する情報を表示します。

表 7-69 : sp_iqwho のカラム

| カラム名 | 説明 |
|-------------|-----------------------------------------------------|
| ConnHandle | SA 接続ハンドル。 |
| IQConnID | Sybase IQ 固有の接続 ID。 |
| Userid | 接続 “ConnHandle” を開いたユーザの名前。 |
| BlockedOn | 特定の接続をブロックする接続。どの接続もブロックしない場合は 0 です。 |
| BlockUserid | ブロックする接続の所有者。ブロック接続が存在しない場合は NULL です。 |
| ReqType | 接続を通じて行われた要求のタイプ。コマンドを発行しない場合は DO_NOTHING です。 |
| IQCmdType | 接続から発行された Sybase IQ コマンドのタイプ。コマンドを発行しない場合は NONE です。 |

| カラム名 | 説明 |
|------------------|----------------------------------------------------------------------------------------------------------|
| IQIdle | この接続を通じて最後の Sybase IQ コマンドを発行してから経過した時間 (秒)。最後に発行した Sybase IQ コマンドが存在しない場合、2000 年 1 月 1 日からの経過時間が表示されます。 |
| SAIdle | この接続を通じて最後の SA 要求を発行してから経過した時間 (秒)。最後に発行した SA コマンドが存在しない場合、2000 年 1 月 1 日からの経過時間が表示されます。 |
| IQCursors | 接続内のアクティブなカーソルの数。カーソルが存在しない場合は 0 です。 |
| IQThreads | 接続を持つスレッドの数。接続を開くとすぐに、少なくとも 1 つのスレッドが開始するので、IQThreads の最小値は 1 です。 |
| TempTableSpaceKB | テンポラリ・テーブル領域のサイズ (キロバイト)。テンポラリ・テーブル領域が使用されていない場合は 0 です。 |
| TempWorkSpaceKB | テンポラリ作業領域のサイズ (キロバイト)。テンポラリ作業領域が使用されていない場合は 0 です。 |

Adaptive Server Enterprise の互換性 Sybase IQ sp_iqwho ストアド・プロシージャは、Adaptive Server Enterprise sp_who プロシージャにより表示されたカラムに対応する Sybase IQ カラムを取り込みます。いくつかの Adaptive Server Enterprise カラムは、Sybase IQ に適用できないため省略されます。表 7-70 に、Adaptive Server Enterprise sp_who カラムと、sp_iqwho により表示されるカラムのマッピングを示します。

表 7-70 : sp_who カラムと sp_iqwho カラムのマッピング

| sp_who カラム | sp_iqwho カラム |
|-------------|-----------------------------------------------------------------|
| fid | ロックの属するファミリ。Sybase IQ には適用されないため省略されます。 |
| spid | ConnHandle、IQConnID。 |
| status | IQIdle、SAIdle。 |
| loginame | Userid。 |
| origname | ユーザのエイリアス。Sybase IQ には適用されないため省略されます。 |
| hostname | サーバが実行されているホストの名前。現在はサポートされていません。 |
| blk_spid | BlockedOn。 |
| dbname | Sybase IQ に対して 1 つのサーバと 1 つのデータベースが存在し、これらが接続ごとに同じであるため、省略されます。 |
| cmd | ReqType、IQCmdType。 |
| block_xloid | BlockUserid。 |

使用法

connhandle 接続 ID を表す整数。このパラメータを指定した場合、**sp_iqwho** は指定の接続に関する情報のみを返します。指定の接続が開いていない場合、出力にローが表示されません。

user-name ユーザ・ログイン名を表す **char(255)** パラメータ。このパラメータを指定した場合、**sp_iqwho** は指定のユーザに関する情報のみを返します。指定のユーザが接続を開いていない場合、出力にローが表示されません。指定のユーザがデータベースに存在しない場合、**sp_iqwho** はエラー・メッセージ "User **user-name** does not exist" を返します。

arg-type *arg-type* パラメータはオプションであり、最初のパラメータを指定した場合に限り指定できます。*arg-type* の唯一の値は "user" です。*arg-type* 値を "user" として指定すると、**sp_iqwho** は最初のパラメータが数値であってもユーザ名として解釈します。*arg-type* に "user" 以外の値を指定した場合、**sp_iqwho** は次のエラーを返します。

```
"Invalid parameter."
```

arg-type 値は二重引用符で囲みます。

パラメータを指定しない場合、**sp_iqwho** では現在アクティブなすべての接続およびユーザに関する情報が表示されます。

最初の **sp_iqwho** パラメータとして、接続ハンドルまたはユーザ名のいずれかを指定できます。パラメータ *connhandle* と *user-name* は排他的なオプションです。これらのパラメータは、一度に片方だけ指定できます。デフォルトでは、最初のパラメータが数値である場合、パラメータは接続ハンドルと見なされます。最初のパラメータが数値でない場合、ユーザ名と見なされます。

Sybase IQ では数値のユーザ名を使用できます。*arg-type* パラメータは、最初のパラメータの数値をユーザ名として解釈するように **sp_iqwho** に指示します。例を示します。

```
sp_iqwho 1, "user"
```

arg-type "user" を指定すると、**sp_iqwho** は最初のパラメータ 1 を、接続 ID ではなく、ユーザ名として解釈します。1 という名前のユーザがデータベースに存在する場合、**sp_iqwho** はユーザ 1 により開かれた接続に関する情報を表示します。

表 7-71 : sp_iqwho の使用例

| 構文 | 出力 |
|-----------------------|--------------------------|
| sp_iqwho | アクティブなすべての接続を表示します。 |
| sp_iqwho 3 | 接続 3 に関する情報を表示します。 |
| sp_iqwho "DBA" | ユーザ DBA により開かれた接続を表示します。 |

| 構文 | 出力 |
|-----------------------------------------|---------------------------------------------------------------------------------------|
| <code>sp_iqwho 3, "user"</code> | 3 をユーザ名として解釈し、ユーザ 3 により開かれた接続を表示します。ユーザ 3 が存在しない場合、エラー "User 3 does not exist" を返します。 |
| <code>sp_iqwho non-existing-user</code> | エラー "User non-existing-user does not exist" を返します。 |
| <code>sp_iqwho 3, "xyz"</code> | エラー "Invalid parameter: xyz" を返します。 |

例 すべてのアクティブな接続を表示します。

```

ConnHandle IQConnID Userid ReqType      IQCmdType          BlockedOn BlockUserid IQCursors
IQThreads  IQIdle  SAIdle  TempTableSpaceKB  TempWorkSpaceKB

12          118      DBA     CURSOR_OPEN      IQUTILITYOPENCURSOR 0          (NULL)      0
1           1         0       0                 0
13          119      shweta  DO_NOTHING       NONE                 0          (NULL)      0
1          16238757  470     0                 0

```

参照

[「sp_iqconnection プロシージャ」\(372 ページ\)](#)

[「sa_conn_info システム・プロシージャ」\(493 ページ\)](#)

sp_iqworkmon プロシージャ

機能

負荷モニタリングの使用状況情報の収集処理を制御し、モニタリング収集ステータスをレポートします。

構文

```

sp_iqworkmon [ 'action' ] [ , 'mode' ]
action = 'start', 'stop', 'status', 'reset'
mode = 'index', 'table', 'column', 'all'

```

次に例を示します。

```

sp_iqworkmon 'start', 'all'

```

引数を 1 つしか指定しない場合、*action* 以外は指定できません。例を示します。

```

sp_iqworkmon 'stop'

```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

使用法

action 適用する制御アクションを指定します。値 *start* を指定すると、指定されたモードでモニタリングがすぐに開始されます。値を *stop* にすると、モニタリングがすぐに停止されます。値 *reset* を指定すると、収集された使用状況の情報がすべて消去されます。値 *status* (デフォルト値) を指定すると、状態を変えずに現在のステータスが表示されます。

mode 制御するモニタリングの種類を指定します。INDEX、TABLE、COLUMN の各キーワードは、インデックスの使用状況、テーブルの使用状況、カラムの使用状況のモニタリングをそれぞれ個別に制御します。デフォルトの ALL キーワードでは、すべての使用状況モニタリング機能のモニタリングが同時に制御されます。

`sp_iqworkmon` の実行には常に結果セットが伴います。特定のモード (インデックスなど) を指定すると、そのモードのローだけが表示されます。

使用状況が収集されるのは、SELECT、UPDATE、DELETE など、FROM 句を含む SQL 文だけです。

表 7-72 : `sp_iqworkmon` のカラム

| カラム名 | 説明 |
|----------|------------------------|
| MonMode | table、index、または column |
| Status | started または stopped |
| Rowcount | 現在までに収集されたローの数 |

例

`sp_iqworkmon` プロシージャからの出力例を示します。

| MonMode | Status | Rowcount |
|---------|---------|----------|
| index | started | 15 |
| table | started | 10 |
| column | started | 31 |

参照

『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」の「負荷のモニタリング」

「`sp_iqcolumnuse` プロシージャ」(371 ページ)、`「sp_iqindexadvice` プロシージャ」(418 ページ)、`「sp_iqdbspace` プロシージャ」(387 ページ)、`「sp_iqindexuse` プロシージャ」(426 ページ)、`「sp_iqtableuse` プロシージャ」(475 ページ)、`「sp_iqunusedcolumn` プロシージャ」(479 ページ)、`「sp_iqunusedindex` プロシージャ」(481 ページ)、および `「sp_iqunusedtable` プロシージャ」(482 ページ)

『リファレンス：文とオプション』の「第2章 データベース・オプション」の「INDEX_ADVISOR オプション」

カタログ・ストアド・プロシージャ

次のカタログ・ストア・ストアド・プロシージャは、データベース・サーバ、データベース、接続プロパティの情報を表にまとめて、結果セットとして返します。これらのプロシージャは、ユーザ ID `dbo` が所有しており、PUBLIC グループが EXECUTE パーミッションを持っています。

sa_audit_string システム・プロシージャ

| | |
|---------|--------------------------------------------------------------------------------|
| 機能 | トランザクション・ログに文字列を追加します。 |
| 構文 | sa_audit_string (<i>string</i>) |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | 監査が有効になっている場合、このシステム・プロシージャは監査ログにコメントを追加します。この文字列は最長 200 バイトです。 |
| 例 | 次の呼び出しは、監査ログにコメントを追加します。 <pre>CALL sa_audit_string('Auditing test')</pre> |

sa_checkpoint_execute システム・プロシージャ

| | |
|---------|------------------------------------------------------------------------------------------------------------------------|
| 機能 | チェックポイント中にシェル・コマンドの実行を可能にします。 |
| 構文 | sa_checkpoint_execute 'shell_commands' |
| パラメータ | shell_commands システム・シェル内で1つまたは複数のユーザ・コマンドを実行します。シェル・コメントはシステム・シェルに固有のもので、コマンドはセミコロン (;) で区切られます。 |
| パーミッション | なし。 |
| 説明 | サーバがクワイス (静止) 状態のとき、シェル・コマンドを実行して、実行中のデータベースをチェックポイント操作中にコピーできます。コピーされたデータベースは、システム障害の後と同じように、開始して通常のリカバリを行うことができます。 |

`sa_checkpoint_execute` は、チェックポイントを開始し、チェックポイントの中からシステム・シェルを実行し、ユーザ・コマンドをシェルに渡します。サーバは、シェルが完了するのを待って、データベース・ファイルをコピーするための任意のサイズの時間枠を作成します。チェックポイントの実行中、ほとんどのデータベース・アクティビティは停止します。したがって、シェル・コマンドの実行時間は、ユーザへの応答時間として許容できる範囲の時間に制限されます。

シェル・コマンドが 0 以外のステータスを返すと、`sa_checkpoint_execute` はエラーを返します。

`sa_checkpoint_execute` は、対話型コマンドと共に使用すると、対話型コマンドが強制終了されるまでサーバが待機しなければならないため、その場合の使用は避けてください。対話型にすることが可能なシェル・コマンドでプロンプトが表示されるのを防ぐために (COPY、MOVE、DELETE などのコマンドでは確認のプロンプトが表示されることがあります)、上書きフラグを立てます。

`sa_checkpoint_execute` の本来の用途は、ディスク・ミラーリングと共に使用して、ミラーリングされたデバイスを分割することです。

`sa_checkpoint_execute` を使用して `iqdemo.*` ファイルを別のディレクトリにコピーすると、`.db` ファイルと `.log` ファイルを除くすべてのファイルがコピーされ、エラー -910 が返されます。

このエラーは、製品異常ではなく Windows の制限事項です。つまり、Windows のコピー・コマンドでは、データベースによって開かれているカタログ・ファイルをコピーできません。

例 `backup` というサブディレクトリが作成されていると仮定した場合、次の文は、チェックポイントを発行し、`iqdemo` データベースの全ファイルを `backup` サブディレクトリにコピーして、チェックポイントを完了します。

```
sa_checkpoint_execute 'cp iqdemo.* backup/'
```

sa_conn_activity システム・プロシージャ

機能 サーバ上のデータベースへの各接続で、最後に準備された SQL 文を返します。

構文 `sa_conn_activity`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

| | |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 関連する動作 | なし |
| 説明 | <p><code>sa_conn_activity</code> プロシージャは、サーバがその情報を収集するよう指示されている場合、各接続に対して最後に準備された SQL 文で構成された結果セットを返します。結果セットを取得するには、データベース・サーバを起動するときに <code>-zi</code> オプションを指定するか、次のコマンドを実行します。</p> <pre>CALL sa_server_option('Remember_last_statement','ON')</pre> <p>このプロシージャは、データベース・サーバがビジー状態であるときに、各接続に対して準備された SQL 文に関する情報を取得する場合に役立ちます。この機能は、要求レベル・ログの代わりに使用できます。</p> <p>これらの値を派生させる <code>LastStatement</code> プロパティの詳細については、『SQL Anywhere Server データベース管理ガイド』を参照してください。</p> <p><code>-zi</code> コマンド・ライン・オプションの詳細については、『ユーティリティ・ガイド』の「第1章 データベース・サーバの実行」を参照してください。</p> <p><code>remember_last_statement</code> 設定の詳細については、「sa_server_option システム・プロシージャ」(503 ページ)を参照してください。</p> |

sa_conn_info システム・プロシージャ

| | |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 接続プロパティ情報をレポートします。 |
| 構文 | <code>sa_conn_info ([connection-id])</code> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | <p>指定した接続に関して、次の接続プロパティを含む結果セットを返します。<code>connection-id</code> を指定しなかった場合は、そのサーバにあるデータベースへの現在の全接続について情報が返されます。</p> <ul style="list-style-type: none"> • Number • Name • Userid • DBNumber • LastReqTime • ReqType |

- CommLink
- NodeAddr
- ClientPort
- ServerPort
- BlockedOn
- LockRowID
- LockIndexID
- LockTable
- UncommitOps

デッドロックの状況下でこのプロシージャから返される **BlockedOn** 値により、ブロックされているユーザと、そのユーザのブロックの競合ユーザをチェックできます。

例

```
sa_conn_info
569851433, '', 'DBA', 0, '', '0.0', 1,
'Cursors_OPEN', 'local', '', 6821, 0, 0, 1008
```

sa_conn_properties システム・プロシージャ

機能 接続プロパティ情報をレポートします。

構文 `sa_conn_properties ([connection-id])`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 使用可能な各接続プロパティについて Number、PropNum、PropName、PropDescription、Value の接続 ID を返します。connection-id を指定しなかった場合は、全接続の情報が返されます。

使用可能な接続プロパティのリストについては、『SQL Anywhere Server データベース管理ガイド』の [Connection properties \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/connection-properties.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbadmin_en11/connection-properties.html) を参照してください。

例

```
sa_conn_properties

Number      PropNum PropName      PropDescription      Value
```



```

1,2,'BytesReceived','Bytes received by server','4157'
1,3,'BytesReceivedUncomp','Bytes received after decompression','4241'
1,4,'BytesSent','Bytes sent to client','15192'
1,10,'CacheHits','Cache Hits','1291'
1,11,'CacheReadIndInt','Cache index interior reads','58'
1,12,'CacheReadIndLeaf','Cache index leaf reads','121'
1,15,'CacheRead','Cache reads','1318'
1,19,'CacheReadTable','Cache table reads','387'
1,20,'CacheReadWorkTable','Cache work table reads','4'
1,21,'CarverHeapPages','Cache pages used for carvers','0'
1,39,'ClientStmtCacheHits','Number of prepares not required because of
the client statement cache','0'
1,40,'ClientStmtCacheMisses','Number of prepares in the client
statement cache which were prepared again','0'
1,41,'Commit','Number of commit requests','1'
1,48,'Cursor','Declared cursors','4'1,41,'Commit','Number of commit
requests','1'

```

注意 キャッシュ全体のキャッシュ・ヒット率の統計情報を取得するには、`sa_eng_properties` を使用し、出力中の `CacheHitsEng`、`CacheReadEng`、`DiskReadEng` を確認します。同じクエリを繰り返しカタログ・ストアで実行する場合、初回は、読み取りは増加しているのにキャッシュ・ヒットはないという結果が出ます。クエリを繰り返すと、キャッシュの読み取りと同時にキャッシュ・ヒットが増えていきます。

sa_db_info システム・プロシージャ

| | |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | データベースのプロパティ情報をレポートします。 |
| 構文 | <code>sa_db_info ([database-id])</code> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | 指定したデータベースの Number、Alias、File、ConnCount、PageSize、LogName を含む 1 つのローを返します。 |
| 例 | <ul style="list-style-type: none"> 次の文は、現在のデータベースの詳細を含む 1 つのローを返します。サンプルの値を表 7-73 に示します。 <pre> sa_db_info 0,'iqdemo',' </pre> |

```
/sys1/users/test/sybase/IQ-15_1/demo/iqdemo.db',
1,4096, '/sys1/users/test/sybase/IQ-15_1/demo/
iqdemo.log'
```

表 7-73 : sa_db_info の値のサンプル

| プロパティ | 値 |
|-----------|--------------------------------------------------------------|
| Number | 0 |
| Alias | iqdemo |
| File | C:\Documents and Settings\All Users\SybaseIQ\demo\iqdemo.db |
| ConnCount | 1 |
| PageSize | 4096 |
| LogName | C:\Documents and Settings\All Users\SybaseIQ\demo\iqdemo.log |

参照

[「sa_db_properties システム・プロシージャ」 \(496 ページ\)](#)

sa_db_properties システム・プロシージャ

機能

データベースのプロパティ情報をレポートします。

構文

```
sa_db_properties ([ database-id ])
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

sa_db_info システム・プロシージャによって返された各プロパティのデータベース ID 番号、Number、PropNum、PropName、PropDescription、Value を返します。

例

```
sa_db_properties

0,10,'CacheHits','Cache Hits','4660'
0,11,'CacheReadIndInt','Cache index interior reads','334'
0,12,'CacheReadIndLeaf','Cache index leaf reads','1117'
0,15,'CacheRead','Cache reads','4887'
0,19,'CacheReadTable','Cache table reads','2025'
0,20,'CacheReadWorkTable','Cache work table reads','4'
0,22,'ChkptFlush','Checkpoint flushed pages','64'/
0,23,'ChkptPage','Checkpoint log page images saved','64'
0,24,'CheckpointUrgency','Checkpoint Urgency','7'
0,25,'Chkpt','Checkpoints','2'
0,26,'CheckpointLogBitmapSize','Checkpoint log bitmap
size','0'
```

```
'0,27,'CheckpointLogBitmapPagesWritten','Checkpoint log
writes to bitmap','0'
0,28,'CheckpointLogCommitToDisk','Checkpoint log commit to
disk','8
0,29,'CheckpointLogPageInUse','Checkpoint log pages in
use','62'
0,30,'CheckpointLogPagesRelocated','Checkpoint log pages
relocated','0'
0,31,'CheckpointLogSavePreimage','Checkpoint log save
preimage','64
```

参照 [「sa_db_info システム・プロシージャ」\(495 ページ\)](#)

sa_enable_auditing_type システム・プロシージャ

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 監査を有効にし、監査対象のイベントを指定します。 |
| 構文 | sa_enable_auditing_type ([<i>string</i>]) |
| パラメータ | <i>string</i> は、カンマで区切った文字列です。 |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | <p>sa_enable_auditing_type は、PUBLIC.AUDITING オプションと共に実行し、特定の種類の情報の監査を有効にします。</p> <p>PUBLIC.AUDITING オプションを ON に設定し、監査対象の情報の種類を指定しなかった場合、デフォルトの設定 (all) が適用されます。この場合、あらゆる種類の監査情報が記録されます。</p> <p>PUBLIC.AUDITING オプションを ON に設定し、sa_disable_auditing_type を使用してすべての種類の監査を無効にすると、監査情報は記録されません。監査を再確立するには、sa_enable_auditing_type を使用して、監査する情報の種類を指定する必要があります。</p> <p>PUBLIC.AUDITING オプションを OFF に設定すると、sa_enable_auditing_type の設定にかかわらず、監査情報は記録されません。</p> |
| 例 | <p>オプションの監査のみを有効にするには、次のように指定します。</p> <pre>sa_disable_auditing_type('all') sa_enable_auditing_type('options')</pre> |

参照 『リファレンス：文とオプション』の「第2章 データベース・オプション」の「AUDITING オプション [データベース]」

sa_eng_properties システム・プロシージャ

機能 データベース・サーバのプロパティ情報をレポートします。

構文 **sa_eng_properties**

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 使用可能な各サーバ・プロパティの PropNum、PropName、PropDescription、Value を返します。

表 7-74 : sa_eng_properties の結果セット

| カラム名 | データ型 | 説明 |
|-----------------|--------------|---------------------|
| PropNum | integer | データベース・サーバ・プロパティの番号 |
| PropName | varchar(255) | データベース・サーバ・プロパティの名前 |
| PropDescription | varchar(255) | データベース・サーバ・プロパティの説明 |
| Value | long varchar | データベース・サーバ・プロパティの値 |

使用可能なサーバのプロパティの一覧については、『SQL Anywhere Server SQL リファレンス・マニュアル』の「System 関数」を参照してください。『SQL Anywhere Server データベース管理ガイド』の「サーバ・レベルのプロパティ」も参照してください。

例 次の文は、使用可能な一連のサーバ・プロパティを返します。

```
call sa_eng_properties()
```

| PropNum | PropName | ... |
|---------|-----------|-----|
| 1 | IdleWrite | ... |
| 2 | IdleChkPt | ... |
| ... | ... | ... |

次の文は、使用可能な一連のサーバ・プロパティを返します。

```
sa_eng_properties
0,'ActiveReq','Active requests','1'
1,'AvailIO','Number of available I/O control blocks','255'
2,'BytesReceived','Bytes received by server','85898'
3,'BytesReceivedUncomp','Bytes received after
```

```

decompression','85898'
4,'BytesSent','Bytes sent to client','145053'
5,'BytesSentUncomp','Bytes sent before compression','145053'
6,'CacheAllocated','Cache pages that have been allocated for
server data structures','546'
7,'CacheFile','Cache pages used to hold data from database
files','600'
8,'CacheFileDirty','Cache pages that are dirty (needing a
write)','2'
9,'CacheFree','Number of cache pages not being used','270'
13,'CachePanics','Number of times the cache manager has failed
to find a page to allocate','0'
14,'CachePinned','Pinned cache pages','591'
15,'CacheRead','Cache reads','456801'
16,'CacheReplacements','Cache replacements','0'
17,'CacheScavengeVisited','Number of pages visited while
scavenging for a page to allocate','1416'
18,'CacheScavenges','Number of times the cache manager has
scavenged for a page to allocate','1416'
21,'CarverHeapPages','Cache pages used for carvers','0'
39,'ClientStmtCacheHits','Number of prepares not required
because of the client statement cache','23'
40,'ClientStmtCacheMisses','Number of prepares in the client
statement cache which were prepared again','3'
44,'CurrentCacheSize','Current cache size in
kilobytes','49152'
51,'DiskRead','Disk reads','938'
54,'DiskReadHintScatterLimit','Imposed limit on the size (in
bytes) of a scatter read hint','0'
55,'DiskRetryRead','Disk read retries','0'

```

sa_table_page_usage システム・プロシージャ

| | |
|---------|------------------------------------------------------------------------------|
| 機能 | データベース・テーブルの使用状況に関する情報をレポートします。 |
| 構文 | sa_table_page_usage |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | 結果には、情報ユーティリティから提供されるものと同じ情報が含まれます。 |

sa_disable_auditing_type システム・プロシージャ

| | |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 特定のイベントの監査を無効にします。 |
| 構文 | sa_disable_auditing_type ([<i>string</i>]) |
| パラメータ | <p><i>string</i> は、次のいずれかを含む、カンマで区切った文字列です。</p> <p>all すべての種類の監査を有効にします。</p> <p>connect 成功した接続と失敗した接続の両方の監査を有効にします。</p> <p>connectFailed 失敗した接続の監査を有効にします。</p> <p>DDL DDL 文の監査を有効にします。</p> <p>options public オプションの監査を有効にします。</p> <p>permission パーミッション・チェック、ユーザ・チェック、setuser 文の監査を有効にします。</p> <p>permissionDenied 失敗したパーミッションおよびユーザのチェックの監査を有効にします。</p> <p>triggers トリガ・イベント後の監査を有効にします。</p> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | <p>sa_disable_auditing_type システム・プロシージャを使用して、1 つまたは複数の種類の情報の監査を無効にすることができます。</p> <p>このオプションを all に設定すると、すべての監査が無効になります。public.auditing オプションを OFF にして監査を無効にすることもできます。</p> |

sa_flush_cache システム・プロシージャ

| | |
|---------|----------------------------------------------------------------------------------------------------------------------|
| 機能 | データベース・サーバ・キャッシュのすべてのページを空にします。 |
| 構文 | sa_flush_cache |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | データベース管理者が、このプロシージャを使用してデータベース・サーバのキャッシュを空にすることができます。このプロシージャは、カタログ・ストアに作用します。パフォーマンス計測時に、繰り返し得られる結果の精度を高めるために使用します。 |

sa_make_object システム・プロシージャ

機能 SQL スクリプトで使用し、実際の定義を提供する ALTER 文を実行する前に、オブジェクトの骨組みのインスタンスが存在することを確実にします。

構文 `sa_make_object (objtype, objname [, owner [, tabname]]`
`object-type: 'procedure' | 'function' | 'view' | 'trigger'`

パーミッション データベース・オブジェクトの作成および変更には、RESOURCE 権限が必要です。

説明 このプロシージャは、データベース・スキーマを作成または変更するために繰り返し実行されるスクリプトまたはコマンド・ファイルで特に役立ちます。このようなスクリプトで通常問題になるのが、最初の実行時には CREATE 文を実行する必要があり、次回からは ALTER 文を実行する必要があることです。このプロシージャを利用することで、システム・テーブルをクエリしてオブジェクトが存在するかどうかを確認する必要がなくなります。

このプロシージャを使用するには、完全なオブジェクト定義を含む ALTER 文を後に続けます。

sa_make_object システム・プロシージャを使用して、スケルトン Web サービスを追加することもできます。

```
CALL sa_make_object( 'service', 'my_web_service' )
```

表 7-75 に、sa_make_object のパラメータの意味を示します。

表 7-75 : sa_make_object のオプション

| オプション名 | 値 |
|-----------|---------------------------------------------------------------------------------------------|
| objtype | 作成するオブジェクトのタイプ。このパラメータには、'procedure'、'function'、'view'、'service'、または 'trigger' のいずれかを指定します。 |
| objname | 作成するオブジェクトの名前。 |
| owner | 作成するオブジェクトの所有者。デフォルト値は CURRENT USER です。 |
| tablename | objtype が 'trigger' である場合のみ、このオプションでトリガを作成するテーブル名を指定する必要があります。 |

例

- 次の文は、スケルトン・プロシージャの定義が作成されていることを確認した後、プロシージャを定義し、それにパーミッションを付与します。これらの命令を含むコマンド・ファイルは、データベースに対して繰り返し実行してもエラーは発生しません。

```
CALL sa_make_object( 'procedure','myproc' );
ALTER PROCEDURE myproc( in p1 int, in p2 char(30) )
BEGIN
    // ...
END;
GRANT EXECUTE ON myproc TO public;
```

- 次の例では、**sa_make_object** システム・プロシージャを使用して、スケルトン Web サービスを追加します。

```
CALL sa_make_object( 'service','my_web_service' )
```

参照

[「sa_db_info システム・プロシージャ」\(495 ページ\)](#)

sa_rowgenerator システム・プロシージャ

機能

指定の開始値と終了値の間にあるローを含む結果セットを返します。

構文

```
sa_rowgenerator ( [ rstart [ , rend [ , rstep ] ] ] )
```

パラメータ

- **rstart** 開始値を指定する、オプションの整数パラメータです。デフォルト値は 0 です。
- **rend** 終了値を指定する、オプションの整数パラメータです。デフォルト値は 100 です。
- **rstep** シーケンス値の増分を指定する、オプションの整数パラメータです。デフォルト値は 1 です。

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

結果セット

| カラム名 | データ型 | 説明 |
|---------|---------|---------|
| row_num | integer | シーケンス番号 |

備考

sa_rowgenerator プロシージャをクエリの FROM 句で使用することにより、シーケンス番号を生成できます。このプロシージャは、RowGenerator システム・テーブルの代わりに使用できます。**sa_rowgenerator** は、次のようなタスクで使用できます。

- 結果セット内の既知の数のローに対してテスト・データを生成する。
- 各範囲内の値に対するローを含む結果セットを生成する。たとえば、月の各日に対応するローや、郵便番号の範囲を生成できます。
- 結果セット内の指定の数のローを含むクエリを生成する。これは、クエリのパフォーマンスをテストするために使用できます。

次の文により、RowGenerator テーブルの動作をエミュレートできます。

```
SELECT row_num FROM sa_rowgenerator(1255)
```

関連する動作

なし

例

次のクエリは、当月の各日について1つのローを含む結果セットを返します。

```
SELECT dateadd(day,row_num-1,
ymd(datepart(year,CURRENT DATE),
datepart(month,CURRENT DATE), 1)) AS
day_of_month FROM sa_RowGenerator(1,31,1) WHERE
datepart(month,day_of_month) =
datepart(month,CURRENT DATE) ORDER BY row_num
```

次のクエリは、住所の郵便番号が(0~9999)、(10000~19999)、~(90000~99999)の範囲にある従業員の数を表示します。これらの範囲のいくつかでは従業員が含まれません。したがって、警告 Null value eliminated in aggregate function (-109) が発生します。sa_rowgenerator プロシージャは、たとえ範囲内の郵便番号を持つ従業員が存在しない場合でも、それらの範囲を生成するために使用できます。

```
SELECT row_num AS r1, row_num+9999 AS r2,
count(PostalCode) AS zips_in_range
FROM sa_rowgenerator(0,99999,10000) D LEFT JOIN
employees ON PostalCode
BETWEEN r1 AND r2 GROUP BY r1, r2 ORDER BY 1
```

次の例では、10件のローから成るデータを生成し、それらを emp テーブルに挿入します。

```
INSERT INTO emp1(dept_id, salary, name) SELECT row_num,
CAST( rand() * 1000 AS INTEGER), 'Mary' FROM
sa_rowgenerator(1, 10)
```

sa_server_option システム・プロシージャ

機能

データベース・サーバの実行中に、データベース・サーバのコマンド・ライン・オプションを上書きします。

構文

```
sa_server_option ( option_name, option_value )
```

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

データベース管理者はこのプロシージャを使用して、データベース・サーバを再起動せずにデータベース・サーバ・オプションを上書きできます。

次のオプションをリセットできます。

| オプション名 | 値 | デフォルト |
|-------------------------|-------------------------------------|-------|
| Disable_connections | ON、OFF | OFF |
| IQMsgMaxSize | 0 ~ 2047 の整数 (メガバイト) | 0 |
| IQMsgNumFiles | 0 ~ 64 の整数 | 0 |
| Liveness_timeout | 整数、秒単位 | 120 |
| Main_Cache_Cemory_MB | 1 - 4294967295 (2 ³² -1) | 32MB |
| Temp_Cache_Memory_MB | 1 - 4294967295 (2 ³² -1) | 24MB |
| Procedure_profiling | ON、OFF、RESET、CLEAR | OFF |
| Profile_filter_conn | connection-id | |
| Profile_filter_user | user-id | |
| Quitting_time | 有効な日付および時刻 | |
| Remember_last_statement | ON、OFF | OFF |
| Request_level_log_file | ファイル名 | |
| Request_level_log_size | ファイルサイズ (バイト) | |
| Request_level_logging | ALL、SQL、NONE、SQL+hostvars | NONE |
| Requests_for_connection | connection-id、-1 | |
| Requests_for_database | database-id、-1 | |

disable_connections ON に設定すると、データベース・サーバ上のすべてのデータベースに対して他の接続が許可されません。

IQMsgMaxSize メッセージ・ログの最大サイズを制限します。IQMsgMaxSize は、0 ~ 2047 の整数で、単位はメガバイトです。デフォルトは 0 で、メッセージ・ログのサイズの制限がないことを示します。IQMsgMaxSize は、-iqmsgsz サーバ・スイッチに対応し、-iqmsgsz の値よりも優先されます。

0 よりも大きい IQMsgMaxSize 値 *n* は、メッセージ・ログが最大 *n* メガバイトまで大きくなることを意味します。たとえば、次の文はメッセージ・ログのサイズを 50MB に制限します。

```
CALL sa_server_option('IQMsgMaxSize','50')
```

0 よりも大きい `-iqmsgsz` 値 n は、メッセージ・ログが最大 n メガバイトまで大きくなることを意味します。たとえば、次のコマンドはメッセージ・ログのサイズを 100MB に制限します。

```
start_iq -n iqdemo iqdemo.db ... <other options> ...
-iqmsgsz 100
```

`-iqmsgsz` サーバ起動スイッチの詳細については、『ユーティリティ・ガイド』の「第1章 データベース・サーバの実行」の「データベース・サーバの起動」を参照してください。

メッセージ・ログ管理については、『システム管理ガイド 第1巻』の「第1章 Sybase IQ システム管理の概要」の「メッセージ・ロギング」を参照してください。

IQMsgNumFiles サーバによって管理される古いメッセージ・ログのアーカイブの数を指定します。IQMsgNumFiles の値は、IQMsgMaxSize サーバ・プロパティまたは `-iqmsgsz` サーバ起動スイッチが 0 以外の場合のみ有効です。IQMsgNumFiles は、`-iqmsgnum` サーバ・スイッチに対し、`-iqmsgnum` の値よりも優先されます。

IQMsgNumFiles は 0 ~ 64 の整数です。デフォルトは 0 で、メッセージはメインのメッセージ・ログ・ファイル内でラッピングされることを示します。

0 よりも大きい IQMsgNumFiles 値 n は、サーバで n 個のメッセージ・ログ・アーカイブを維持することを意味します。たとえば、次の文は、メッセージ・ログの 5 個のアーカイブをサーバで維持することを指定します。

```
CALL sa_server_option('IQMsgNumFiles','5')
```

`-iqmsgnum` サーバ起動スイッチの詳細については、『ユーティリティ・ガイド』の「第1章 データベース・サーバの実行」の「データベース・サーバの起動」を参照してください。

メッセージ・ログ管理については、『システム管理ガイド 第1巻』の「第1章 Sybase IQ システム管理の概要」の「メッセージ・ロギング」を参照してください。

liveness_timeout 接続が維持されていることを確認するため、クライアント/サーバの TCP/IP または SPX ネットワークを介して、定期的に活性パケットが送信されます。活性パケットを検出することなく、`liveness_timeout` で指定した時間にわたってネットワーク・サーバが実行されていると、通信は切断されます。

`-tl` コマンド・ライン・オプションの詳細については、『システム管理ガイド 第1巻』の「第2章 Sybase IQ の実行」の「非アクティブ接続の制限」を参照してください。

main_cache_memory_mb メイン共有メモリ・バッファ・キャッシュのデフォルトを動的に変更します。このオプションは実行中のサーバで設定できますが、実行中のデータベースでキャッシュ・サイズを変更することはできません。2つのデータベースを異なるキャッシュ・サイズで実行する必要がある場合は、各データベースを起動する前にオプションを設定します。`-iqmc` サーバ起動スイッチを使用してキャッシュ・サイズを設定すると、`sa_server_option` で新しい値を指定しない限り、Sybase IQ は、そのサーバで起動されたすべてのデータベースに対し指定の値を使用します。詳細については、『パフォーマンス&チューニング・ガイド』の「[第4章 システム・リソースの管理](#)」の「[バッファ・キャッシュと物理メモリ](#)」を参照してください。

procedure_profiling ストアド・プロシージャ、関数、イベント、トリガのプロシージャ・プロファイリングを制御します。プロシージャ・プロファイリングでは、ストアド・プロシージャ、関数、イベント、およびトリガの実行にかかる時間、および各行の実行にかかる時間が表示されます。プロシージャ・プロファイリングのオプションは、Sybase Central のデータベース・プロパティ・シートでも設定できます。収集された情報は、Sybase Central の右側のペインにある [プロファイル] タブに表示されます。

- **ON** 現在接続中のデータベースのプロシージャ・プロファイリングを有効にします。
- **OFF** プロシージャ・プロファイリングを無効にし、プロファイリング・データは表示可能にします。
- **RESET** ON または OFF の設定はそのまま、プロファイリング・カウンタを 0 に戻します。
- **CLEAR** プロファイリング・カウンタを 0 に戻し、プロシージャ・プロファイリングを無効にします。

プロファイリングが有効になると、`sa_procedure_profile_summary` および `sa_procedure_profile` ストアド・プロシージャを使用して、データベースからプロファイリング情報を取得できるようになります。これらのプロシージャの詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』を参照してください。

Sybase Central でのプロシージャ・プロファイリング情報の表示の詳細については、『パフォーマンス&チューニング・ガイド』の「[データベース・プロシージャのプロファイリング](#)」を参照してください。

profile_filter_conn 特定の接続 ID のプロファイル情報を収集するようにデータベース・サーバに指示します。

profile_filter_user 特定のユーザ ID のプロファイル情報を収集するようにデータベース・サーバに指示します。

quitting_time データベース・サーバが指定された時間に停止するよう指示します。

quitting_time の詳細については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」の `-tq` サーバ・オプションを参照してください。

remember_last_statement データベース・サーバが、サーバ上のデータベースへの各接続で最後に準備された SQL 文を取得するよう指示します。ストアド・プロシージャ・コールの場合、最も外側のプロシージャ・コールだけが表示され、プロシージャ内部の文は表示されません。

RememberLastStatement プロパティ関数を次のように使用すると、**remember_last_statement** 設定の現在の値を取得できます。

```
SELECT property( 'RememberLastStatement' )
```

詳細については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」の `-zl` サーバ・オプションを参照してください。

remember_last_statement が有効になっている場合、次の文は、指定された接続用に最後に準備された文を返します。

```
SELECT connection_property( 'LastStatement', conn_id )
```

ストアド・プロシージャ **sa_conn_activity** は、これと同じ情報をすべての接続に関して返します。

request_level_log_file ログ情報を記録するのに使用するファイルの名前。名前に NULL を指定した場合、ファイルのログが停止します。SQL 文字列として使用するので、ファイル名に含まれるバックスラッシュ文字は2つ指定する必要があります。

request_level_log_size ログ情報を記録するファイルの最大サイズ(バイト)。

要求レベル・ログ・ファイルが、**sa_server_option** システム・プロシージャまたは `-zs` サーバ・オプションで指定されたサイズに達すると、ファイルは `.old` という拡張子を付けられ、名前が変更されます(同じ名前のファイルがある場合はこれを上書きします)。その後、要求レベル・ログ・ファイルは再開されます。

request_level_logging ログ・オプションは、ALL、SQL、NONE、HOSTVARS、PLAN、PROCEDURES、および TRIGGERS で、“+”で区切ります。ON と ALL は同義です。OFF と NONE は同義です。この呼び出しにより、データベース・サーバに送信され、データベース・サーバの `-zr` および `-zo` オプションと組み合わせて、トラブルシューティングに使用される個別の SQL 文のログが有効になります。

SQL には、要求に関連する基本の SQL 文が含まれます。ALL には、SQL 要求とその他の要求が含まれ、これを使用すると、ログのサイズが大幅に増加することがあります。また、ALL は、ホスト変数の値の記録を有効にします。TRIGGERS を指定すると、すべてのストアド・プロシージャ文 (トリガ内の文も含む) が要求ログに記録されます。

request_level_debugging と request_level_logging の 2 つの設定は同義です。request_level_logging を OFF に設定すると、要求レベル・ログ・ファイルが閉じられます。

SQL を選択すると、次のタイプの要求のみが記録されます。

- START DATABASE
- STOP ENGINE
- STOP DATABASE
- 文の準備
- 文の実行
- EXECUTE IMMEDIATE 文
- オプションの設定
- COMMIT 文
- ROLLBACK 文
- PREPARE TO COMMIT 操作
- 接続
- 接続の切断
- トランザクションの開始
- DROP STATEMENT 文
- カーソルの説明
- カーソルのクローズ
- カーソルの再開
- エラー

request_level_logging を SQL+HOSTVARS に設定すると、SQL (request_level_logging=SQL を指定した場合と同様) とホスト変数の値の両方がログに出力されます。

`request_level_logging` 設定の現在の値は、`property('RequestLogging')` を使用して確認できます。

詳細については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」の `-z`、`-zr`、`-zs`、`-zo`、および `-o` コマンド・ライン・オプションを参照してください。

SQL 要求レベル・ロギングでログが取られる要求のリストについては、『ユーティリティ・ガイド』の「[-zr level](#)」(32 ページ)を参照してください。要求ロギングの使用法の詳細については、『システム管理ガイド 第1巻』の「[第14章 トラブルシューティングのヒント](#)」の「[サーバ要求のロギング](#)」を参照してください。『SQL Anywhere サーバー SQL の使用法 > データベース・パフォーマンスのモニタリングと改善 > パフォーマンスのモニタリングと改善 > その他の診断ツールと方法』の「要求ロギング」も参照してください。

requests_for_connection 要求レベル・ロギングの情報をフィルタし、特定の接続に関する情報だけをログに記録します。これは、多数のアクティブな接続や複数のデータベースがあるサーバをモニタする場合に、要求レベル・ログ・ファイルのサイズを小さくするのに役立ちます。接続 ID を取得するには、次の文を実行します。

```
CALL sa_conn_info()
```

接続 ID を取得した後に、特定の接続をロギングするよう指定するには、次の文を実行します。

```
CALL sa_server_option( 'requests_for_connection',  
connection-id )
```

フィルタリングは、明示的にリセットされるか、データベース・サーバが停止するまで有効です。フィルタリングをリセットするには、次の文を使用します。

```
CALL sa_server_option( 'requests_for_connection', -1)
```

requests_for_database 要求レベル・ロギングの情報をフィルタし、特定のデータベースに関する情報だけをログに記録します。これは、多数のアクティブな接続や複数のデータベースがあるサーバをモニタする場合に、要求レベル・ログ・ファイルのサイズを小さくするのに役立ちます。現在接続しているデータベースのデータベース ID を取得するには、次の文を実行します。

```
SELECT connection_property( 'DBNumber' )
```

特定のデータベースに関する情報だけをログに記録するよう指定するには、次の文を実行します。

```
CALL sa_server_option( 'requests_for_database',  
database-id )
```

フィルタリングは、明示的にリセットされるか、データベース・サーバが停止するまで有効です。フィルタリングをリセットするには、次の文を使用します。

```
CALL sa_server_option( 'requests_for_database', -1 )
```

temp_cache_memory_mb テンポラリ共有メモリ・バッファ・キャッシュのデフォルトを動的に変更します。このオプションは実行中のサーバで設定できますが、実行中のデータベースでキャッシュ・サイズを変更することはできません。2つのデータベースを異なるキャッシュ・サイズで実行する必要がある場合は、各データベースを起動する前にオプションを設定します。**-iqtc** サーバ起動スイッチを使用してキャッシュ・サイズを設定すると、**sa_server_option** で新しい値を指定しない限り、Sybase IQ は、そのサーバで起動されたすべてのデータベースに対し指定の値を使用します。詳細については、『パフォーマンス&チューニング・ガイド』の「[第4章 システム・リソースの管理](#)」の「[バッファ・キャッシュと物理メモリ](#)」を参照してください。

例 次の文は、データベース・サーバに新しい接続を確立できないようにします。

```
call sa_server_option( 'disable_connections', 'ON')
```

次の文は、メイン共有メモリ・バッファ・キャッシュのサイズを変更します。

```
call sa_server_option( 'main_cache_memory_mb', '200')
```

新しいサイズを有効にするには、データベースを再起動する必要があります。

参照 『SQL Anywhere Server SQL リファレンス・マニュアル』の「[sa_get_request_profile system procedure](#)」、[sa_get_request_times system procedure](#)」、および「[sa_statement_text system procedure](#)」

sa_set_http_header システム・プロシージャ

機能 Web サービスが、結果内の HTTP ヘッダを設定できるようにします。

構文 **sa_set_http_header** (*field-name*, *value*)

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

```
call dbo.sa_set_http_header( 'Content-Type', 'text/html' )
```


特殊なヘッダ・フィールド `@HttpStatus` を設定すると、要求に対して返されるステータス・コードが設定されます。たとえば、次のコマンドでは、ステータス・コードを "404 Not Found" に設定します。

```
dbo.sa_set_http_header( '@HttpStatus', '404' )
```

エラー・メッセージ本文は自動的に挿入されます。使用できるのは、有効な HTTP エラー・コードだけです。ステータスに無効なコードを設定すると、SQL エラーが発生します。

参照 [「sa_set_http_option システム・プロシージャ」 \(511 ページ\)](#)

sa_set_http_option システム・プロシージャ

機能 Web サービスが、結果内の HTTP オプションを設定できるようにします。

構文 `sa_set_http_option (option-name, value)`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明 Web サービスを操作して HTTP 結果セット内のオプションを設定する文またはプロシージャ内でこのプロシージャを使用します。

現在サポートされているオプションは1つだけです。

- **CharsetConversion** データベースの文字セットからクライアントの文字セットに、結果セットを自動的に変換するかどうかを制御します。設定可能な値は ON および OFF です。デフォルト値は ON です。

参照 [「sa_set_http_header システム・プロシージャ」 \(510 ページ\)](#)

sa_validate システム・プロシージャ

機能 カタログ・ストア内のすべてのテーブルを検証します。

構文 `sa_validate [tbl_name,][owner_name,][check_type]`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストア・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

このプロシージャは、SQL Anywhere の各テーブルまたはカタログ・ストア内のインデックスを検証します。

詳細については、『ユーティリティ・ガイド』の「第3章 データベース管理ユーティリティ」の「検証ユーティリティ (dbvalid)」を参照してください。

表 7-76 に、sa_validate のパラメータの意味を示します。

表 7-76 : sa_validate のオプション

| オプション名 | 値 |
|------------|--------------------------------------------------------------------------------------------------------------|
| tbl_name | 指定されたテーブルのみを検証します。NULL (デフォルト値) の場合、すべてのテーブルを検証します。 |
| owner_name | 指定されたユーザが所有するテーブルのみを検証します。NULL (デフォルト値) の場合、すべてのユーザのテーブルを検証します。 |
| check_type | NULL (デフォルト) の場合、各テーブルは追加のチェックなしでチェックされます。check_type には、data、express、full、index、または checksum のいずれかの値を指定できます。 |

tbl_name、owner_name、および check_type パラメータの値は、文字列を引用符で囲んだものである必要があります。

このプロシージャは、Messages という名前の単独のカラムを返します。すべてのテーブルが有効である場合、このカラムの内容は次のとおりです。

```
No errors detected
```

警告！ テーブルやカタログ・ストア全体を検証できるのは、データベースに変更を加えている接続がないときのみです。データベースの変更中にこれを行うと、データベースに実際には破損がないにもかかわらず、何らかの破損があることを示す擬似エラーがレポートされることがあります。

例

次の文は、DBA が所有するすべてのカタログ・ストア・テーブルを検証し、インデックスのチェックを行います。

```
CALL sa_validate (owner_name='DBA', check_type =
'index')
```

sa_verify_password システム・プロシージャ

| | |
|---------|------------------------------------------------------------------------------------------------------------------|
| 機能 | 現在のユーザのパスワードを検証します。 |
| 構文 | sa_verify_password (<i>string</i>) |
| パラメータ | <ul style="list-style-type: none"> • string この char(128) パラメータは、現在のデータベース・ユーザのパスワードを指定します。 |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 備考 | このプロシージャは、sp_password により使用されます。パスワードが一致した場合、プロシージャは何も返しません。パスワードが一致しない場合、プロシージャからエラー文字列が返されます。 |
| 関連する動作 | なし。 |

sp_login_environment システム・プロシージャ

sp_login_environment システム・プロシージャは、SQL Anywhere システム・プロシージャです。詳細については、『SQL Anywhere Server - SQL Reference』の [sp_login_environment system procedure \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sp-login-environment-sysproc.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sp-login-environment-sysproc.html) を参照してください。

sp_remote_columns システム・プロシージャ

| | |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | リモート・テーブルにあるカラムのリストと、それらのカラムの説明を生成します。カラムごとに、データベース、所有者、テーブル、カラム、ドメイン ID、幅、小数点以下の桁数、および NULL 入力属性が返されます。 このシステム・プロシージャを使用するには、サーバを CREATE SERVER 文で定義する必要があります。 |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

注意 このプロシージャからの出力を、ファイルに取り込むことはできません。リダイレクト演算子を使用すると、“カーソルは FETCH NEXT 操作に制限されています。”というメッセージが返されます。

| | |
|----|---------------------------------------------------------------------------------------------------------|
| 構文 | sp_remote_columns <i>servername</i> [, <i>tablename</i>] [, <i>owner</i>] [, <i>database</i>] |
|----|---------------------------------------------------------------------------------------------------------|

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | CREATE EXISTING 文を入力していて、カラム・リストを指定している場合は、リモート・テーブルで使用可能なカラムのリストを取得しておく役立つ場合があります。sp_remote_columns はリモート・テーブルのカラムのリストと、それらのデータ型の説明を生成します。 |
| 例 | asetest という名前の Adaptive Server サーバの production データベースにある sysobjects テーブル内のカラムのリストを取得します。 <pre>sp_remote_columns asetest, sysobjects, null, production</pre> |
| 標準と互換性 | Sybase Open Client/Open Server でサポートされています。 |
| 参照 | 『システム管理ガイド 第 2 巻』の「 第 5 章 リモート・データ・アクセス用のサーバ・クラス 」および「 第 4 章 リモート・データへのアクセス 」 『リファレンス：文とオプション』の「 CREATE SERVER 文 」 |

sp_remote_exported_keys システム・プロシージャ

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | 指定されたプライマリ・キー・テーブルで外部キーを持つテーブルに関する情報を表示します。 このシステム・プロシージャを使用するには、サーバを CREATE SERVER 文で定義する必要があります。 |
| 構文 | <pre>sp_remote_exported_keys @server_name , @sp_name [, @sp_owner][, @sp_qualifier]</pre> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | sp_remote_exported_keys 結果セットには、データベース、所有者、テーブル、カラム、プライマリ・キーおよび外部キーの名前、そして外部キー・カラムの外部キー・シーケンスが含まれます。基本となる ODBC および JDBC 呼び出しによって結果セットは異なりますが、外部キーのテーブルとカラムに関する情報は必ず返されます。 sp_remote_exported_keys を使用するには、Sybase IQ のバージョン 12.4.3 以降でデータベースが作成またはアップグレードされている必要があります。 |
| パラメータ | sp_remote_exported_keys のパラメータを 表 7-77 に示します。 |

表 7-77 : `sp_remote_exported_keys` のパラメータ

| 名前 | データ型 | 説明 |
|----------------------------|-------------|-------------------------------|
| <code>@server_name</code> | varchar | プライマリ・キー・テーブルのあるサーバ。必須。 |
| <code>@sp_name</code> | varchar(30) | プライマリ・キーを含むテーブル。必須。 |
| <code>@sp_owner</code> | varchar | プライマリ・キー・テーブルの所有者。オプション。 |
| <code>@sp_qualifier</code> | varchar | プライマリ・キー・テーブルを含むデータベース。オプション。 |

例 `asetest` という名前のサーバにある `production` データベースの `sysobjects` テーブル上に外部キーを持つリモート・テーブルに関する情報を取得するには、次のように指定します。

```
call sp_remote_exported_keys
(@server_name='asetest',@sp_name='sysobjects',
 @sp_qualifier='production')
```

参照 『システム管理ガイド 第2巻』の「第4章 リモート・データへのアクセス」および「第5章 リモート・データ・アクセス用のサーバ・クラス」
『リファレンス：文とオプション』の「[CREATE SERVER 文](#)」

`sp_remote_imported_keys` システム・プロシージャ

機能 指定された外部キーに対応するプライマリ・キーを持つリモート・テーブルに関する情報を表示します。

このシステム・プロシージャを使用するには、サーバを `CREATE SERVER` 文で定義する必要があります。

構文 `sp_remote_imported_keys @server_name , @sp_name [, @sp_owner] [, @sp_qualifier]`

パーミッション DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、`EXECUTE` パーミッションが付与される必要があります。

説明

外部キーは、対応するプライマリ・キーを含む別のテーブル上のローを参照します。このプロシージャを使用すると、特定の外部キー・テーブルに対応するプライマリ・キーを含むリモート・テーブルのリストを取得できます。sp_remote_imported_keys の結果セットには、データベース、所有者、テーブル、カラム、プライマリ・キーおよび外部キーの名前、そして外部キー・カラムの外部キー・シーケンスが含まれます。基本となる ODBC および JDBC 呼び出しによって結果セットは異なりますが、プライマリ・キーのテーブルとカラムに関する情報は必ず返されます。

sp_remote_exported_keys を使用するには、Sybase IQ のバージョン 12.4.3 以降でデータベースが作成またはアップグレードされている必要があります。

パラメータ

sp_remote_imported_keys のパラメータを表 7-78 に示します。

表 7-78 : sp_remote_imported_keys のパラメータ

| 名前 | データ型 | 説明 |
|---------------|-------------|---------------------------|
| @server_name | varchar | 外部キー・テーブルのあるサーバ。必須。 |
| @sp_name | varchar(30) | 外部キーを含むテーブル。必須。 |
| @sp_owner | varchar | 外部キー・テーブルの所有者。オプション。 |
| @sp_qualifier | varchar | 外部キー・テーブルを含むデータベース。オプション。 |

例

asetest サーバ上の、“fred” が所有する sysobjects テーブル上にある外部キーに対応するプライマリ・キーが含まれるテーブルに関する情報を取得します。

```
call sp_remote_imported_keys
(@server_name='asetest', @sp_name='sysobjects',
@sp_qualifier='production')
```

参照

『システム管理ガイド 第 2 巻』の「[第 4 章 リモート・データへのアクセス](#)」および「[第 5 章 リモート・データ・アクセス用のサーバ・クラス](#)」
『リファレンス：文とオプション』の「[CREATE SERVER 文](#)」

sp_remote_primary_keys システム・プロシージャ

機能

リモート・データ・アクセスを使用して、リモート・テーブルのプライマリ・キー情報を表示します。

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | <pre>sp_remote_primary_keys @server_name [, @table_name] [, @table_owner] [, @table_qualifier]</pre> <p>次のパラメータを使用できます。</p> <p>@server_name リモート・テーブルが置かれているサーバを選択します。</p> <p>@table_name リモート・テーブルを選択します。</p> <p>@table_owner リモート・テーブルの所有者を選択します。</p> <p>@table_qualifier データベースを選択します。</p> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 関連する動作 | なし |
| 説明 | 基本となる ODBC/JDBC 呼び出しの違いによって、返される情報のうち、サーバに指定されたリモート・データ・アクセスのクラスに依存するカタログ/データベースの値が多少異なります。重要な情報 (カラム名など) は変わりません。 |
| 標準と互換性 | Sybase Open Client/Open Server でサポートされています。 |

sp_remote_tables システム・プロシージャ

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能 | サーバ上のテーブルのリストを返します。 |
| 構文 | <pre>sp_remote_tables servername [, tablename] [, owner] [, table_qualifier] [, with_table_type]</pre> <p>このシステム・プロシージャを使用するには、サーバを CREATE SERVER 文で定義する必要があります。</p> |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | <p>データベース・サーバを設定するとき、特定のサーバで利用できるリモート・テーブルのリストがあると便利です。sp_remote_tables は、サーバ上のテーブルのリストを返します。</p> <p>このプロシージャでは次のパラメータを使用できます。</p> <p>server_name リモート・テーブルが置かれているサーバを選択します。</p> <p>table_name リモート・テーブルを選択します。</p> |

table_owner リモート・テーブルの所有者を選択します。

table_qualifier データベースを選択します。

with_table_type リモート・テーブルのタイプを選択します。このパラメータは bit 型で、値は 0 (デフォルト) または 1 です。テーブルのタイプを示すカラムを結果セットに含めるには 1 を指定します。

with_table_type パラメータは、SQL Anywhere 7.0.2 以降で作成されたデータベースでのみ使用できます。それ以前のバージョンで作成されたデータベースでこのパラメータを使用すると、次のエラー・メッセージが返されます。

```
Wrong number of parameters to function 'sp_remote_tables'
```

テーブル名、所有者、またはデータベース名が指定されている場合、これらのパラメータに一致するテーブルだけがリストされます。

注意 このプロシージャからの出力を、ファイルに取り込むことはできません。リダイレクト演算子を使用すると、“カーソルは FETCH NEXT 操作に制限されています。” というメッセージが返されます。

例

- “exce” という名前の ODBC データ・ソースから、使用可能なすべての Microsoft Excel ワークシートのリストを取得するには、次のように指定します。

```
sp_remote_tables excel
```

- ユーザ fred が所有する asetest という名前の Adaptive Server Enterprise サーバにある production データベース内のすべてのテーブルを表示するには、次のように指定します。

```
sp_remote_tables asetest, null, fred, production
```

標準と互換性

Sybase Open Client/Open Server でサポートされています。

参照

『リファレンス：文とオプション』の「[CREATE SERVER 文](#)」

『システム管理ガイド 第 2 巻』の「[第 4 章 リモート・データへのアクセス](#)」および「[第 5 章 リモート・データ・アクセス用のサーバ・クラス](#)」

sp_servercaps システム・プロシージャ

機能

リモート・サーバの機能に関する情報を表示します。

このシステム・プロシージャを使用するには、サーバを CREATE SERVER 文で定義する必要があります。

| | |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 構文 | sp_servercaps servername |
| パーミッション | DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。 |
| 説明 | Sybase IQ は機能情報に基づいて、リモート・サーバに転送することができる SQL 文の量を判断します。サーバの機能情報を保持するシステム・テーブルには、Sybase IQ がリモート・サーバに接続した後でないと、値が入りません。この情報は、 syscapability と syscapabilityname システム・テーブルから取得されます。指定した <i>servername</i> は、CREATE SERVER 文で使用するサーバ名と同じでなければなりません。 |
| 例 | リモート・サーバ testiq に関する情報を表示します (出力はトランケートされています)。 |

```

sp_servercaps testiq

1,'Alter table with add','T'
2,'Alter table with drop','T'
3,'Owner supported','T'
4,'Primary key requires index','F'
5,'Create table constraints','T'
6,'Truncate table','T'
7,'Create index','T' 7,'Create index','T'
8,'Create unique index','T'
9,'Syscapability system table initialized','T'
10,'Subquery','T'
11,'Subquery in group by','T'
12,'Subquery in comparison','T'
13,'Subquery in exist','T'
14,'Subquery in IN','T'
15,'Subquery correlated','T'
16,'Subquery in select list','T'
17,'Subquery in update','T'
20,'Order by','T'
21,'Order by expressions','T'
22,'Order by column not in select list','T'
23,'Order by allowed in update','T'
25,'Joins','T'
26,'Outer joins','T'
27,'Full outer joins','T'
28,'Multiple outer joins','T'
29,'Logical operators in outer join','T'
30,'Outer joins mixed with normal joins','T'
31,'ANSI join syntax','T'
32,'TSQL join syntax','F'
33,'ODBC outer join syntax','F'

```

```

34,'Unrestricted ANSI ON','T'
40,'Group by','T'
41,'Group by ALL','T'
45,'Aggregates','T'
46,'Aggregates with column name','T'
50,'And','T'
51,'Or','T'
52,'Like','T'
53,'Like - TSQL','T'
54,'Distinct','T'
55,'In','T'

```

標準と互換性

Sybase Open Client/Open Server でサポートされています。

参照

『リファレンス：文とオプション』の「[CREATE SERVER 文](#)」

『システム管理ガイド 第2巻』の「[第4章 リモート・データへのアクセス](#)」および「[第5章 リモート・データ・アクセス用のサーバ・クラス](#)」

sp_tsql_environment システム・プロシージャ

機能

ユーザが jConnect または Open Client アプリケーションから接続するときに、接続オプションを設定します。

構文

sp_tsql_environment

パーミッション

DBA 権限が必要です。DBA 権限を持たないユーザがこのストアド・プロシージャを実行するには、EXECUTE パーミッションが付与される必要があります。

説明

接続で TDS 通信プロトコルを使用する場合 (つまり、Open Client 接続の場合)、今度は `sp_login_environment` が `sp_tsql_environment` を呼び出します。

このプロシージャは、デフォルトの Sybase Adaptive Server Enterprise の動作との互換性を持つように、データベース・オプションを設定します。

デフォルトの動作を変更する場合は、新しいプロシージャを作成して、その新しいプロシージャを指すように LOGIN_PROCEDURE オプションを変更します。

LOGIN_PROCEDURE に新しいプロシージャの名前を設定する方法の詳細については、『システム管理ガイド 第2巻』の「[第3章 データ・サーバとしての Sybase IQ](#)」を参照してください。

`sp_tsql_environment` のテキストを次に示します。

```
create procedure dbo.sp_tsql_environment()
begin
if db_property('IQStore') = 'Off' then

    -- ASA datastore
set temporary option close_on_endtrans = 'OFF'
end if;
set temporary option ansinull = 'OFF';
set temporary option tsql_variables = 'ON';
set temporary option ansi_blanks = 'ON';
set temporary option chained = 'OFF';
set temporary option quoted_identifier = 'OFF';
set temporary option allow_nulls_by_default = 'OFF';
set temporary option on_tsql_error = 'CONTINUE';
set temporary option isolation_level = '1';
set temporary option date_format = 'YYYY-MM-DD';
set temporary option timestamp_format = 'YYYY-MM-DD
HH:NN:SS.SSS';
set temporary option time_format = 'HH:NN:SS.SSS';
set temporary option date_order = 'MDY';
set temporary option escape_character = 'OFF'
end
```

参照

『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」の「[LOGIN_PROCEDURE オプション](#)」

Adaptive Server Enterprise のシステム・プロシージャと カタログ・プロシージャ

Adaptive Server Enterprise では、多くの管理機能を実行し、またシステム情報を取得するためのシステム・プロシージャとカタログ・プロシージャが提供されています。Sybase IQ には、これらのプロシージャの一部が実装されています。

システム・プロシージャは、システム・テーブルから情報を取得して更新するための組み込みストアド・プロシージャです。カタログ・ストアド・プロシージャは、システム・テーブルから表形式で情報を取り出します。

注意 これらのプロシージャは、Adaptive Server Enterprise およびバージョン 12 より前の Sybase IQ に組み込まれていたものと変わりませんが、まったく同じというわけではありません。既存のスクリプトでこれらのプロシージャが使用されている場合、動作をチェックしておいたほうがいいでしょう。ストアド・プロシージャの説明を表示するには、次を実行します。

```
sp_helptext 'owner.procedure_name'
```

Sybase が提供するすべてのシステム・ストアド・プロシージャでは、所有者は `dbo` です。別のユーザが所有する同じ名前のストアド・プロシージャのテキストを参照するには、そのユーザを指定する必要があります。次に示すのはその例です。

```
sp_helptext 'myname.myprocedure'
```

Adaptive Server Enterprise システム・プロシージャ

Sybase IQ が提供している Adaptive Server Enterprise のシステム・プロシージャを表 7-79 に示します。

表 7-79 : Sybase IQ に用意されている ASE システム・プロシージャ

| システム・プロシージャ | 説明 |
|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>sp_addgroup group-name</code> | データベースにグループを追加します。 |
| <code>sp_addlogin userid, password[, defdb [, deflanguage [, fullname]]]</code> | データベースに新規ユーザ・アカウントを追加します。 |
| <code>sp_addmessage message-num, message_text [, language]</code> | ユーザ定義メッセージを <code>SYSUSERMESSAGES</code> に追加して、ストアド・プロシージャの <code>PRINT</code> 呼び出しと <code>RAISERROR</code> 呼び出しで使用します。 |
| <code>sp_addtype typename, datatype, [, "identity" nulltype]</code> | ユーザ定義データ型を作成します。Sybase IQ では <code>IDENTITY</code> カラムはサポートしていません。 |
| <code>sp_adduser userid [, name_in_db [, grpname]]</code> | データベースに新規ユーザを追加します。 |
| <code>sp_changegroup new-group-name, userid</code> | ユーザのグループを変更する、またはグループにユーザを追加します。 |
| <code>sp_dboption [dbname, optname, {true false}]</code> | データベース・オプションを表示または変更します。 |
| <code>sp_dropgroup group-name</code> | データベースからグループを削除します。 |
| <code>sp_droplogin userid</code> | データベースからユーザを削除します。 |

| システム・プロシージャ | 説明 |
|----------------------------------------------------------------------|------------------------------------------------------------------------|
| <code>sp_dropmessage message-number [, language]</code> | ユーザ定義メッセージを削除します。 |
| <code>sp_droptype typename</code> | ユーザ定義データ型を削除します。 |
| <code>sp_dropuser userid</code> | データベースからユーザを削除します。 |
| <code>sp_getmessage message-num, @msg-var output [, language]</code> | 格納されているメッセージ文字列を SYSUSERMESSAGES から取り出して、PRINT 文および RAISERROR 文で使用します。 |
| <code>sp_helptext 'owner.object-name'</code> | システム・プロシージャまたはビューのテキストを表示します。 |
| <code>sp_password caller_passwd, new_passwd [, userid]</code> | ユーザ ID のパスワードを追加または変更します。 |

注意 `sp_dropuser` などのプロシージャには、Adaptive Server Enterprise のストアド・プロシージャとの互換性はほとんどありません。Adaptive Server Enterprise (または Sybase IQ 11.x) のストアド・プロシージャを使い慣れている場合は、それらのテキストと Sybase IQ 12 のプロシージャとを比較してから、`dbisql` でプロシージャを使用してください。比較には、次のコマンドを使用します。

```
sp_helptext 'owner.procedure_name'
```

Sybase が提供するシステム・ストアド・プロシージャでは、所有者は常に `dbo` です。別のユーザが所有する同じ名前のストアド・プロシージャのテキストを参照するには、そのユーザを指定する必要があります。次に示すのはその例です。

```
sp_helptext 'myname.myprocedure'
```

Adaptive Server Enterprise カタログ・プロシージャ

Sybase IQ には、`sp_column_privileges` 以外の Adaptive Server Enterprise のカタログ・プロシージャがほとんど実装されています。実装されているカタログ・プロシージャを表 7-80 に示します。また、Sybase IQ では、これらの Adaptive Server カタログ・プロシージャの一部について、同様にカスタマイズされたストアド・プロシージャが用意されています。

表 7-80 : Sybase IQ に実装されている ASE カタログ・プロシージャ

| ASE カタログ・プロシージャ | 説明 | IQ プロシージャ |
|-----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|-----------------------------|
| <code>sp_columns table-name [, table-owner] [, table-qualifier] [, column-name]</code> | 指定したカラムのデータ型を返します。 | |
| <code>sp_fkeys phtable_name [, phtable-owner] [, phtable-qualifier] [, fhtable-name] [, fhtable_owner] [, fhtable-qualifier]</code> | 指定したテーブルの外部キー情報を返します。 | |
| <code>sp_pkeys table-name [, table_owner] [, table_qualifier]</code> | 指定したテーブルのプライマリ・キー情報を返します。 | <code>sp_iqpkeys</code> |
| <code>sp_special_columns table_name [, table-owner] [, table-qualifier] [, col-type]</code> | テーブルのローをユニークに識別するのに最適なカラム・セットを返します。 | |
| <code>sp_sproc_columns proc-name [, proc_owner] [, proc-qualifier] [, column-name]</code> | ストアド・プロシージャの入力パラメータおよびリターン・パラメータ情報を返します。 | <code>sp_iqprocparm</code> |
| <code>sp_stored_procedures [sp-name] [, sp-owner] [, sp-qualifier]</code> | 1 つ以上のストアド・プロシージャの情報を返します。 | <code>sp_iqprocedure</code> |
| <code>sp_tables table-name [, table-owner] [, table-qualifier] [, table-type]</code> | FROM 句に指定できるオブジェクトのリストを返します。 | |

以下の Adaptive Server Enterprise カタログ・プロシージャはサポートされていません。

- `sp_column_privileges`
- `sp_databases`
- `sp_datatype_info`
- `sp_server_info`

SQL Anywhere でサポートされているプロシージャ

Sybase IQ は、『SQL Anywhere Server SQL リファレンス・マニュアル』の [Alphabetical list of system procedures \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/spcsp.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/spcsp.html) にリストされている SQL Anywhere プロシージャをサポートしています。

注意 『SQL Anywhere Server SQL リファレンス・マニュアル』の「システム拡張ストアド・プロシージャ」の項には、ユーザは EXECUTE パーミッションを付与されているか、権限を持っている必要があると記載されています。ただし、後続のプロシージャの中には、パーミッションとして NONE が記載されているものがあります。

次のプロシージャには、DBA 権限が必要です。

-
- `xp_cmdshell` – 外部シェル・コマンドを実行するためにデータベース・サーバによって使用されるシステム拡張プロシージャ。

この章について

この章では、Sybase IQ システム・テーブルの定義済みビューを順次説明します。

システム・ビューについて

システム・テーブルは内部での使用を目的として設計されています。システム・テーブルの内容を見るには、システム・ビューを使用します。システム・テーブル内の情報を読みやすい形で表示するために、定義済みシステム・ビューが多数用意されています。

システム・ビューの記述ではその定義も示します。複雑なものもありますが、ビューを使用するには理解する必要はありません。定義は、SELECT コマンドとビューで何が実行できるかを示す格好の例になっています。

統合ビューについて

統合ビューにはユーザがよく要求する形式でデータが表示されます。たとえば、統合ビューにはよく一般に必要とされるジョインが用意されています。統合ビューはシステム・ビューとは違って、システム・テーブルからの生データをそのまま表示するビューではありません。たとえば、システム・ビューのカラムの多くは意味のない ID の値ですが、統合ビューでは、それは意味のある名前になっています。

SYSCATALOG や SYSINDEXES などの統合ビューは、Sybase IQ と SQL Anywhere の両方に共通です。各種の統合ビューの定義については、[Consolidated views \(PLUGINS_ROOT/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4117677.html\)](http://com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4117677.html) を参照してください。

互換ビューについて

互換ビューは廃止されたビューですが、旧バージョンの SQL Anywhere と Sybase IQ との互換性のために提供されています。できるだけ、互換ビューの代わりにシステム・ビューと統合ビューを使用するようにしてください。互換ビューは Sybase IQ の将来のバージョンから削除される可能性があります。

互換ビューの詳細については、[Compatibility views \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5195671.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5195671.html) を参照してください。

ASE T-SQL 互換 ビューについて

Sybase IQ には DBO という特殊なユーザが所有するビューのセットがあります。これは Adaptive Server Enterprise のシステム・テーブルとビューに対応しています。詳細については、「[Transact-SQL 互換のビュー](#)」(591 ページ)を参照してください。

内容

| トピック名 | ページ |
|-------------------------------------------------|-----|
| SYSARTICLE システム・ビュー | 532 |
| SYSARTICLECOL システム・ビュー | 532 |
| SYSARTICLECOLS 統合ビュー | 532 |
| SYSARTICLES 統合ビュー | 533 |
| SYSCAPABILITIES 統合ビュー | 533 |
| SYSCAPABILITY システム・ビュー | 533 |
| SYSCAPABILITYNAME システム・ビュー | 534 |
| SYSCATALOG 統合ビュー | 534 |
| SYSCHECK システム・ビュー | 534 |
| SYSCOLAUTH 統合ビュー | 535 |
| SYSCOLPERM システム・ビュー | 535 |
| SYSCOLLATION 互換ビュー (廃止) | 535 |
| SYSCOLLATIONMAPPINGS 互換ビュー (廃止) | 536 |
| SYSCOLSTAT システム・ビュー | 536 |
| SYSCOLSTATS 統合ビュー | 537 |
| SYSCOLUMN 互換ビュー (廃止) | 537 |
| SYSCOLUMNS 統合ビュー | 537 |
| SYSCOLUMNS ASE 互換ビュー | 538 |
| SYSCOMMENTS ASE 互換ビュー | 538 |
| SYSCONSTRAINT システム・ビュー | 538 |
| SYSDBFIL システム・ビュー | 539 |
| SYSDBSpace システム・ビュー | 539 |
| SYSDBSpacePERM システム・ビュー | 539 |
| SYSDependency システム・ビュー | 540 |
| SYSDomain システム・ビュー | 540 |
| SYSEvent システム・ビュー | 541 |
| SYSEVENTTYPE システム・ビュー | 541 |
| SYSEXTERNENV システム・ビュー | 541 |
| SYSEXTERNENVOBJECT システム・ビュー | 542 |
| SYSEXTERNLOGIN システム・ビュー | 542 |
| SYSFILE 互換ビュー (廃止) | 543 |
| SYSFKCOL 互換ビュー (廃止) | 543 |

| トピック名 | ページ |
|----------------------------------------------------|-----|
| SYSFKEY システム・ビュー | 543 |
| SYFOREIGNKEY 互換ビュー (廃止) | 544 |
| SYFOREIGNKEYS 統合ビュー | 544 |
| SYSGROUPS 統合ビュー | 545 |
| SYSHISTORY システム・ビュー | 545 |
| SYSIDX システム・ビュー | 545 |
| SYSIDXCOL システム・ビュー | 546 |
| SYSINDEX 互換ビュー (廃止) | 546 |
| SYSINDEXES 統合ビュー | 547 |
| SYSINDEXES ASE 互換ビュー | 547 |
| SYSINFO 互換ビュー (廃止) | 548 |
| SYSIQBACKUPHISTORY システム・ビュー | 548 |
| SYSIQBACKUPHISTORYDETAIL システム・ビュー | 549 |
| SYSIQCOLUMN システム・ビュー (廃止) | 551 |
| SYSIQDBFILE システム・ビュー | 551 |
| SYSIQDBSPACE システム・ビュー | 552 |
| SYSIQFILE システム・ビュー (廃止) | 553 |
| SYSIQIDX システム・ビュー | 553 |
| SYSIQINFO システム・ビュー | 555 |
| SYSIQJOINIDX システム・ビュー | 556 |
| SYSIQJOININDEX システム・ビュー (廃止) | 557 |
| SYSIQJOINIXCOLUMN システム・ビュー | 557 |
| SYSIQJOINIXTABLE システム・ビュー | 559 |
| SYSIQMPXLOGINPOLICYOPTION システム・ビュー | 559 |
| SYSIQMPXSERVER システム・ビュー | 560 |
| SYSIQOBJECTS ASE 互換ビュー | 560 |
| SYSIQPARTITIONCOLUMN システム・ビュー | 560 |
| SYSIQTAB システム・ビュー | 561 |
| SYSIQTABCOL システム・ビュー | 562 |
| SYSIQTABLE システム・ビュー (廃止) | 563 |
| SYSIQVINDEXT ASE 互換ビュー | 563 |
| SYSIXCOL 互換ビュー (廃止) | 564 |
| SYSJAR システム・ビュー | 564 |
| SYSJARCOMPONENT システム・ビュー | 564 |
| SYSJAVACLASS システム・ビュー | 565 |
| SYSLOGINMAP システム・ビュー | 565 |
| SYSLOGINPOLICY システム・ビュー | 565 |

| トピック名 | ページ |
|-----------------------------------------------|-----|
| SYSLOGINPOLICYOPTION システム・ビュー | 566 |
| SYSLOGINS ASE 互換ビュー | 566 |
| SYSMVOPTION システム・ビュー | 566 |
| SYSMVOPTIONNAME システム・ビュー | 566 |
| SYSOBJECT システム・ビュー | 567 |
| SYSOBJECTS ASE 互換ビュー | 567 |
| SYSOPTION システム・ビュー | 568 |
| SYSOPTIONS 統合ビュー | 568 |
| SYSOPTSTAT システム・ビュー | 568 |
| SYSPARTITION システム・ビュー | 569 |
| SYSPARTITIONKEY システム・ビュー | 570 |
| SYSPARTITIONSCHEME システム・ビュー | 570 |
| SYSPHYSIDX システム・ビュー | 571 |
| SYSPROCAUTH 統合ビュー | 571 |
| SYSPROCEDURE システム・ビュー | 572 |
| SYSPROCPARM システム・ビュー | 572 |
| SYSPROCPARMS 統合ビュー | 572 |
| SYSPROCPERM システム・ビュー | 573 |
| SYSPROCS 統合ビュー | 573 |
| SYSPROXYTAB システム・ビュー | 573 |
| SYS PUBLICATION システム・ビュー | 574 |
| SYS PUBLICATIONS 統合ビュー | 574 |
| SYSREMARK システム・ビュー | 574 |
| SYSREMOTEOPTION システム・ビュー | 575 |
| SYSREMOTEOPTION2 統合ビュー | 575 |
| SYSREMOTEOPTIONS 統合ビュー | 575 |
| SYSREMOTEOPTIONTYPE システム・ビュー | 576 |
| SYSREMOTETYPE システム・ビュー | 576 |
| SYSREMOTETYPES 統合ビュー | 576 |
| SYSREMOTEUER システム・ビュー | 577 |
| SYSREMOTEUERS 統合ビュー | 577 |
| SYSSCHEDULE システム・ビュー | 578 |
| SYSSERVER システム・ビュー | 578 |
| SYSSOURCE システム・ビュー | 578 |
| SYSSQLSERVERTYPE システム・ビュー | 579 |
| SYSSUBPARTITIONKEY システム・ビュー | 579 |
| SYSSUBSCRIPTION システム・ビュー | 579 |

| トピック名 | ページ |
|----------------------------------|-----|
| SYSSUBSCRIPTIONS 統合ビュー | 579 |
| SYSSYNC システム・ビュー | 580 |
| SYSSYNC2 統合ビュー | 580 |
| SYSSYNCPUBLICATIONDEFAULTS 統合ビュー | 581 |
| SYSSYNCS 統合ビュー | 581 |
| SYSSYNCSCRIPT システム・ビュー | 581 |
| SYSSYNCSCRIPTS 統合ビュー | 582 |
| SYSSYNCSUBSCRIPTIONS 統合ビュー | 582 |
| SYSSYNCSUSERS 統合ビュー | 582 |
| SYSTAB システム・ビュー | 583 |
| SYSTABLE 互換ビュー (廃止) | 583 |
| SYSTABAUTH 統合ビュー | 583 |
| SYSTABCOL システム・ビュー | 584 |
| SYSTABLEPERM システム・ビュー | 584 |
| SYSTEXTCONFIG システム・ビュー | 584 |
| SYSTEXTIDX システム・ビュー | 585 |
| SYSTEXTIDXTAB システム・ビュー | 585 |
| SYSTRIGGER システム・ビュー | 585 |
| SYSTRIGGERS 統合ビュー | 586 |
| SYSTYPEMAP システム・ビュー | 586 |
| SYSTYPES ASE 互換ビュー | 586 |
| SYSUSER システム・ビュー | 587 |
| SYSUSERAUTH 互換ビュー (廃止) | 587 |
| SYSUSERAUTHORITY システム・ビュー | 587 |
| SYSUSERLIST 互換ビュー (廃止) | 588 |
| SYSUSERMESSAGE システム・ビュー | 588 |
| SYSUSEROPTIONS 統合ビュー | 588 |
| SYSUSERPERM 互換ビュー (廃止) | 589 |
| SYSUSERPERMS 互換ビュー (廃止) | 589 |
| SYSUSERTYPE システム・ビュー | 589 |
| SYSUSERS ASE 互換ビュー | 590 |
| SYSVIEW システム・ビュー | 590 |
| SYSVIEWS 統合ビュー | 590 |
| SYSWEBSERVICE システム・ビュー | 591 |
| Transact-SQL 互換のビュー | 591 |

SYSARTICLE システム・ビュー

SYSARTICLE システム・ビューの各ローは、パブリケーション内の個々のアートを記述します。このビューの元になっているシステム・テーブルは ISYSARTICLE です。

SYSARTICLE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSARTICLE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysrt.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysrt.html) を参照してください。

SYSARTICLECOL システム・ビュー

SYSARTICLECOL システム・ビューの各ローは、アートの個々のカラムを示します。このビューの元になっているシステム・テーブルは ISYSARTICLECOL です。

SYSARTICLECOL ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSARTICLECOL system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysrtc.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysrtc.html) を参照してください。

SYSARTICLECOLS 統合ビュー

このビューの各ローは、アートの個々のカラムを示します。

SYSARTICLECOLS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSARTICLECOLS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysarticlecols.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysarticlecols.html) を参照してください。

SYSARTICLES 統合ビュー

SYSARTICLES ビューの各ローは、パブリケーション内の個々のアーティクルの記述です。

SYSARTICLES ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSARTICLECOLS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysarticless.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysarticless.html) を参照してください。

SYSCAPABILITIES 統合ビュー

SYSCAPABILITIES ビューの各ローは個々の機能を記述します。このビューのデータは ISYSCAPABILITY システム・テーブルと ISYSCAPABILITYNAME システム・テーブルから取得されます。

SYSCAPABILITIES ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCAPABILITIES consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscapabilities-sysviews.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscapabilities-sysviews.html) を参照してください。

SYSCAPABILITY システム・ビュー

SYSCAPABILITY システム・ビューの各ローは、リモート・サーバの個々の機能を示します。このビューの元になっているシステム・テーブルは ISYSCAPABILITY です。

SYSCAPABILITY ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCAPABILITY system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscap-systab.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscap-systab.html) を参照してください。

SYSCAPABILITYNAME システム・ビュー

SYSCAPABILITYNAME システム・ビューの各ローは、SYSCAPABILITY システム・ビューで定義されている個々の機能の名前です。

SYSCAPABILITYNAME ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCAPABILITYNAME system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscap-systab3453.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscap-systab3453.html) を参照してください。

SYSCATALOG 統合ビュー

SYSCATALOG ビューの各ローは、個々のシステム・テーブルを記述します。

SYSCATALOG ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCATALOG consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscatalog.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscatalog.html) を参照してください。

SYSCHECK システム・ビュー

SYSCHECK システム・ビューの各ローは、テーブル内の個々の名前付き検査制約の定義です。このビューの元になっているシステム・テーブルは ISYSCHECK です。

SYSCHECK ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCHECK system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systb-s-5962966.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systb-s-5962966.html) を参照してください。

SYSCOLAUTH 統合ビュー

SYSCOLAUTH ビューの各ローは、カラムに与えられた権限 (UPDATE、SELECT、REFERENCES) を記述します。SYSCOLAUTH ビューは、SYSCOLPERM システム・ビュー内のデータを読みやすい形式で表示します。

SYSCOLAUTH ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLAUTH consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvcola.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvcola.html) を参照してください。

SYSCOLPERM システム・ビュー

GRANT コマンドを使って、テーブルの各カラムに UPDATE、SELECT、または REFERENCES のパーミッションを与えることができます。UPDATE、SELECT、または REFERENCES パーミッションを持つ個々のカラムが、SYSCOLPERM システム・ビューの各ローに記録されています。このビューの元になっているシステム・テーブルは ISYSCOLPERM です。

SYSCOLPERM ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLPERM system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sycolprm.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sycolprm.html) を参照してください。

SYSCOLLATION 互換ビュー (廃止)

SYSCOLLATION 互換ビューにはそのデータベースの照合順情報が入っています。この情報は組み込み関数によって取得されます。カタログには保存されません。

SYSCOLLATION ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLLATION compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscollat345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscollat345.html) を参照してください。

SYSCOLLATIONMAPPINGS 互換ビュー (廃止)

SYSCOLLATIONMAPPINGS 互換ビューにはデータベース照合マッピングの入ったローが 1 つあるだけです。この情報は組み込み関数によって取得されます。カタログには保存されません。

SYSCOLLATIONMAPPINGS ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLLATIONMAPPINGS compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscollationmappings345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscollationmappings345.html) を参照してください。

SYSCOLSTAT システム・ビュー

SYSCOLSTAT システム・ビューには、オプティマイザが使用するカラム統計 (ヒストグラムを含む) が入っています。このビューの内容の取得には、`sa_get_histogram` ストアド・プロシージャまたは `Histogram` ユーティリティが最も適しています。このビューの元になっているシステム・テーブルは `ISYSCOLSTAT` です。

このビューの元になっているテーブルは不正なアクセスからデータを保護するために、常に暗号化されています。

SYSCOLSTAT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLSTAT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscolstt-systab.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscolstt-systab.html) を参照してください。

SYSCOLSTATS 統合ビュー

SYSCOLSTATS ビューにはオプティマイザが使用するカラム統計がヒストグラムとして入っています。

SYSCOLSTATS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLSTATS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscolstats-sysviews.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscolstats-sysviews.html) を参照してください。

SYSCOLUMN 互換ビュー (廃止)

SYSCOLUMN ビューは、SYSCOLUMN システム・テーブルを使用していた旧バージョンの Sybase IQ との互換性を維持するために提供されています。ただし、以前の SYSCOLUMN テーブルは ISYSTABCOL システム・テーブルとそれに対応する SYSTABCOL システム・ビューによって置き換えられましたので、そちらを使用するようにしてください。

SYSCOLUMN ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLUMN compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5215939.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5215939.html) を参照してください。

SYSCOLUMNNS 統合ビュー

SYSCOLUMNNS ビューの各ローは、カタログ内の各テーブルとビューの個々のカラムを記述します。

SYSCOLUMNNS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCOLUMNNS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvcoll.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvcoll.html) を参照してください。

SYSCOLUMNS ASE 互換ビュー

このビューは DBO というユーザが所有しています。syscolumns の各ローは、あらゆるテーブルとビュー内の各カラムまたはプロシージャ内の各パラメータに対応しています。詳細については、表 8-1 (592 ページ) を参照してください。

SYSCOMMENTS ASE 互換ビュー

このビューは DBO というユーザが所有しています。syscomments には個々のビュー、ルール、デフォルト、トリガ、テーブル制約、パーティション、プロシージャ、計算カラム、機能ベースのインデックス・キー、その他の形式のコンパイル済みオブジェクトがエントリとして入っています。テキスト・カラムには元の定義文が入っています。テキスト・カラムが 255 バイトを超えるとエントリは複数ローにまたがります。個々のオブジェクトは最大 65,025 ローを占めることができます。詳細については、表 8-1 (592 ページ) を参照してください。

SYSCONSTRAINT システム・ビュー

SYSCONSTRAINT システム・ビューの各ローは、そのデータベース内の名前付き制約を記述します。このビューの元になっているシステム・テーブルは ISYSCONSTRAINT です。

SYSCONSTRAINT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSCONSTRAINT system view \(http://infocenter.sybase.com/help/topic.com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792802.html\)](http://infocenter.sybase.com/help/topic.com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792802.html) を参照してください。

SYSDBFILe システム・ビュー

SYSDBFILe システム・ビューの各ローは DB 領域ファイルを記述します。このビューの元になっているシステム・テーブルは ISYSDBFILe です。

注意 このビューは廃止になった SYSDBFILe システム・ビューの代わりに使用されます。

SYSDBFILe ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSDBFILe system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdbfile-sys-view.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdbfile-sys-view.html) を参照してください。

SYSDBSPACE システム・ビュー

SYSDBSPACE システム・ビューの各ローは、個々の DB 領域ファイルを記述します。このビューの元になっているシステム・テーブルは ISYSDBSPACE です。

注意 このビューは廃止になった SYSDBSPACE システム・ビューの代わりに使用されます。

SYSDBSPACE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSDBSPACE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdbspace-sys-view.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdbspace-sys-view.html) を参照してください。

SYSDBSPACEPERM システム・ビュー

SYSDBSPACEPERM システム・ビューの各ローは、DB 領域ファイル上の個々のパーミッションを記述します。このビューの元になっているシステム・テーブルは ISYSDBSPACEPERM です。

SYSDBSPACEPERM ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSDBSPACEPERM system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdbspaceperm-sys-view.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdbspaceperm-sys-view.html) を参照してください。

SYSDEPENDENCY システム・ビュー

SYSDEPENDENCY システム・ビューの各ローは、2 つのデータベース・オブジェクトの間の個々の依存関係を記述します。このビューの元になっているシステム・テーブルは ISYSDEPENDENCY です。

オブジェクトがその定義の中で他のオブジェクトを参照している場合、両者の間に依存関係があります。たとえば、ビューのクエリ仕様でテーブルを参照している場合、そのビューはそのテーブルに依存していると言います。データベース・サーバはテーブル、ビュー、マテリアライズド・ビュー、カラムに対するビューの依存関係を追跡します。

SYSDEPENDENCY ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSDEPENDENCY system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792803.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792803.html) を参照してください。

SYSDOMAIN システム・ビュー

SYSDOMAIN システム・ビューは組み込みデータ型 (ドメインとも呼ばれる) の情報を記録します。このビューの内容は通常の作業では変化しません。このビューの元になっているシステム・テーブルは ISYSDOMAIN です。

SYSDOMAIN ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSDOMAIN system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdomayn.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdomayn.html) を参照してください。

SYSEVENT システム・ビュー

SYSEVENT システム・ビュー内の各ローは、CREATE EVENT で作成された個々のイベントを記述します。このビューの元になっているシステム・テーブルは ISYSEVENT です。

SYSEVENT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSEVENT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysevent-systab34543.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysevent-systab34543.html) を参照してください。

SYSEVENTTYPE システム・ビュー

SYSEVENTTYPE システム・ビューは CREATE EVENT が参照できるシステム・イベント・タイプを定義します。

SYSEVENTTYPE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSEVENTTYPE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syseventtype-systab345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syseventtype-systab345.html) を参照してください。

SYSEXTERNENV システム・ビュー

Sybase IQ は 6 つの外部ランタイム環境をサポートします。これには、C/C++ で書かれた Embedded SQL と ODBC アプリケーション、Java、Perl、PHP、Microsoft .NET Framework Common Language Runtime (CLR) に基づく C# や Visual Basic などの言語で書かれたアプリケーションが含まれます。

SYSEXTERNENV システム・ビューの各ローは、各外部環境を識別して起動するときに必要な情報を記述します。このビューの元になっているシステム・テーブルは ISYSEXTERNENV です。

SYSEXTERNENV ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSEXTERNENV system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-s-5305299.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-s-5305299.html) を参照してください。

SYSEXTERNENVOBJECT システム・ビュー

Sybase IQ は 6 つの外部ランタイム環境をサポートします。これには、C/C++ で書かれた Embedded SQL と ODBC アプリケーション、Java、Perl、PHP、Microsoft .NET Framework Common Language Runtime (CLR) に基づく C# や Visual Basic などの言語で書かれたアプリケーションが含まれます。

SYSEXTERNENVOBJECT システム・ビューの各ローは、インストールされている個々の外部オブジェクトを記述します。このビューの元になっているシステム・テーブルは ISYSEXTERNENVOBJECT です。

SYSEXTERNENVOBJECT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSEXTERNENVOBJECT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-sysexternenvobj.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-sysexternenvobj.html) を参照してください。

SYSEXTERNLOGIN システム・ビュー

SYSEXTERNLOGIN システム・ビューの各ローは、リモート・データ・アクセスの個々の外部ログインを記述します。このビューの元になっているシステム・テーブルは ISYSEXTERNLOGIN です。

SYSEXTERNLOGIN ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSEXTERNLOGIN system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysexternlogns345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysexternlogns345.html) を参照してください。

SYSFILE 互換ビュー (廃止)

SYSFILE システム・ビューの各ローはデータベースの個々の DB 領域 ファイルを記述します。個々のデータベースは 1 つ以上の DB 領域で構成されています。個々の DB 領域は個々のオペレーティング・システム・ファイルに対応しています。

SYSFILE ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSFILE compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systfile34534.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systfile34534.html) を参照してください。

SYSFKCOL 互換ビュー (廃止)

SYSFKCOL の各ローは、外部テーブルの外部カラムと、プライマリ・テーブルのプライマリ・カラムの個々の関係を記述します。このビューは廃止になりました。代わりに SYSIDX と SYSIDXCOL システム・ビューを使用してください。

SYSFKCOL ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSFKCOL compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5203646.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5203646.html) を参照してください。

SYSFKEY システム・ビュー

SYSFKEY システム・ビューの各ローは、そのシステム内の個々の外部キー制約を記述します。このビューの元になっているシステム・テーブルは ISYSFKEY です。

SYSFKEY ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSFKEY system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792805.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792805.html) を参照してください。

SYSFOREIGNKEY 互換ビュー (廃止)

SYSFOREIGNKEY ビューは、SYSFOREIGNKEY システム・テーブルを使用していた旧バージョンの Sybase IQ との互換性を維持するために提供されています。ただし、以前の SYSFOREIGNKEY システム・テーブルは ISYSFKEY システム・テーブルとそれに対応する SYSFKEY システム・ビューによって置き換えられましたので、そちらを使用するようにしてください。

SYSFOREIGNKEY ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSFOREIGNKEY compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5227056.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5227056.html) を参照してください。

SYSFOREIGNKEYS 統合ビュー

SYSFOREIGNKEYS ビューの各ローは、カタログ内の各テーブルの個々の外部キーを記述します。

SYSFOREIGNKEYS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSFOREIGNKEYS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvfk.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvfk.html) を参照してください。

SYSGROUP システム・ビュー

SYSGROUP システム・ビューには各グループの各メンバに対して1つのローがあります。このテーブルは、グループとメンバの多対多の関係を示します。1グループに複数のメンバが属することができ、1ユーザが複数のグループのメンバになれます。このビューの元になっているシステム・テーブルは ISYSGROUP です。

SYSGROUP ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSGROUP system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysgroup345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysgroup345.html) を参照してください。

SYSGROUPS 統合ビュー

SYSGROUPS ビューには各グループの各メンバに対して 1 つのローがあります。このテーブルは、グループとメンバの多対多の関係を示します。1 グループに複数のメンバが属することができ、1 ユーザが複数のグループのメンバになれます。

SYSGROUPS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSGROUPS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvggrp.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvggrp.html) を参照してください。

SYSHISTORY システム・ビュー

SYSHISTORY システム・ビューの各ローには、データベースに対するシステム操作(データベースの起動や調整など)が記録されます。このビューの元になっているシステム・テーブルは ISYSHISTORY です。

SYSHISTORY ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSHISTORY system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systb-s-5122766.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systb-s-5122766.html) を参照してください。

SYSIDX システム・ビュー

SYSIDX システム・ビューの各ローはデータベースの個々の論理インデックスを定義します。このビューの元になっているシステム・テーブルは ISYSIDX です。

注意 このビューは廃止になった SYSINSDX システム・ビューの代わりに使用されます。

SYSIDX ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSIDX system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792806.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792806.html) を参照してください。

SYSIDXCOL システム・ビュー

SYSIDXCOL システム・ビューの各ローは、SYSIDX システム・ビューで記述されている個々のインデックスの個々のカラムを記述します。このビューの元になっているシステム・テーブルは ISYSIDXCOL です。

SYSIDXCOL ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSIDX system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syxicol345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syxicol345.html) を参照してください。

SYSINDEX 互換ビュー (廃止)

SYSINDEX ビューは、SYSINDEX システム・テーブルを使用していた旧バージョンの Sybase IQ との互換性を維持するために提供されています。ただし、以前の SYSINDEX システム・テーブルは ISYSIDX システム・テーブルとそれに対応する SYSIDX システム・ビューによって置き換えられましたので、そちらを使用するようにしてください。

SYSINDEX ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSINDEX compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5239986.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5239986.html) を参照してください。

SYSINDEXES 統合ビュー

SYSINDEXES ビューの各ローはデータベース内の個々のインデックスを記述します。このビューの代わりに、SYSIDX と SYSIDXCOL システム・ビューを使用できます。

SYSINDEXES ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSINDEXES consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syindex.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syindex.html) を参照してください。

SYSINDEXES ASE 互換ビュー

このビューは DBO というユーザが所有しています。sysindexes には、各クラスタード・インデックスに対して 1 つのロー、各ノンクラスタード・インデックスに対して 1 つのロー、クラスタード・インデックスがない各テーブルに対して 1 つのロー、テキストまたは画像カラムを持つ各テーブルに対して 1 つのローが入っています。このテーブルの各ローには個々の機能ベースのインデックスまたは計算カラムに基づいて作成されたインデックスも入っています。詳細については、[表 8-1 \(592 ページ\)](#) を参照してください。

SYSINFO 互換ビュー (廃止)

SYSINFO ビューには、データベースが作成されたときに定義されたデータベース特性が表示されます。このテーブルには常にローが 1 つしかありません。このビューは組み込み関数によって取得されます。カタログには保存されません。

SYSINFO ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSINFO compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysinfo345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysinfo345.html) を参照してください。

SYSIQBACKUPHISTORY システム・ビュー

このビューは ISYSIQBACKUPHISTORY からグループの情報を読みやすい形式で表示します。このビューの各ローは、正常に終了した個々のバックアップ処理を記述します。

SYSIQBACKUP ビューは、カラムのタイプ、サブタイプ、bkp_virtual に対する等価文字列値を示します。

| カラム名 | カラム・タイプ | カラム制約 | 説明 |
|---------|-----------------|----------|--------------------------------------------------------------------------|
| bu_id | unsigned bigint | NOT NULL | 操作のチェックポイントのトランザクション識別子。バックアップ処理のバックアップ ID |
| bu_time | timestamp | NOT NULL | バックアップ記録に記録されたバックアップ処理が行われた時刻 |
| 種類 | tinyint | NOT NULL | バックアップのタイプ： 0 = FULL 1 = INCREMENTAL 2 = INCREMENTAL SINCE FULL |

| カラム名 | カラム・タイプ | カラム制約 | 説明 |
|----------------|-----------------|----------|-------------------------------------------------------------------------------------------------------------------------------------|
| selective_type | tinyint | NOT NULL | バックアップのサブタイプ： 0 = ALL (すべてのdbfiles をバックアップ) 1 = READ/WRITE ONLY (すべての読み書きファイルをバックアップ) 2 = READ ONLY (特定の読み取り専用ファイルをバックアップ) |
| virtual_type | tinyint | NOT NULL | 仮想バックアップの種類 0 = NONE 1 = DECOUPLED 2 = ENCAPSULATED |
| dependson_id | unsigned bigint | NULL | FULL バックアップには NULL |
| cmd | long varchar | NOT NULL | コマンドの完全テキスト |
| creator | char(128) | NOT NULL | バックアップ・コマンドを出したユーザ |
| バージョン | unsigned int | NOT NULL | バックアップのバージョン |

元になっているシステム・テーブルの制約

プライマリ・キー (bu_id)

SYSIQBACKUPHISTORYDETAIL システム・ビュー

このビューはバックアップ時にデータベース内に存在するすべてのdbfile レコードを記述します。このビューの各ローは、正常に終了した個々のバックアップ処理を記述します。ISYSIQBACKUPHISTORYDETAIL からのグループ情報を読みやすい形式で表示します。各カラムのカラム制約は NOT NULL です。

| カラム名 | カラム・タイプ | 説明 |
|----------------------|-----------------|---------------------------------------------------|
| bu_id | unsigned bigint | 操作のチェックポイントのトランザクション識別子。バックアップ処理のバックアップ ID |
| dbspace_id | smallint | この dbfile レコードに関連付けられている DB 領域 ID |
| dbfile_id | smallint | バックアップ処理の進行中に DB 領域に存在する dbfile ID |
| dbspace_rwstatus | char(1) | T は読み書き可能を示す |
| dbspace_createid | unsigned bigint | DB 領域を作成したトランザクションのトランザクション ID |
| dbspace_alterid | unsigned bigint | DB 領域に RO マークを付けたトランザクションの ID。マークなしの場合は ID を作成する。 |
| dbspace_online | char(1) | T はオンラインを示す |
| dbfile_rwstatus | char(1) | T は読み書き可能を示す |
| dbfile_createid | unsigned bigint | この dbfile を作成したトランザクションのトランザクション ID |
| dbfile_alterid | unsigned bigint | この dbfile の読み書きステータスを最後に変更したトランザクションのトランザクション ID |
| is_backed_up | char(1) | このバックアップでその dbfile ファイルがバックアップされたことを示す |
| start_block | unsigned bigint | dbfile の開始ブロック |
| num_blocks | unsigned bigint | dbfile 内の総ブロック数 |
| num_blocks_backed_up | unsigned bigint | バックアップされた総 IQ ブロック数 |
| dbspace_name | char(128) | DB 領域名 |
| dbfile_name | char(128) | dbfile ファイルの論理ファイル名 |
| dbfile_path | long varchar | ファイルの物理パス |

元になっているシステム・テーブルの制約

プライマリ・キー (bu_id、dbfile_id)

外部キー (txn_id) が SYS.ISYSBACKUPHISTORY を参照

SYSIQCOLUMN システム・ビュー (廃止)

SYSIQCOLUMN の代わりに SYSIQTABCOL システム・ビューが使用されます。詳細については、「[SYSIQTABCOL システム・ビュー](#)」(562 ページ) を参照してください。

SYSIQDBFILE システム・ビュー

ISYSIQDBFILE からのグループ情報を読みやすい形式で表示します。

注意 このビューは廃止になった SYSIQFILE システム・ビューの代わりに使用されます。

| カラム名 | カラム・タイプ | 説明 |
|--------------|--------------|---------------------------------------------------------|
| dbfile_id | small int | dbfile のユニークな ID |
| start_block | rowid | 最初のブロックの番号 |
| block_count | rowid | このファイル (DB 領域) のブロック数 |
| reserve_size | rowid | DB 領域に事前に割り付けられているファイル・システム領域 |
| allocated | char(1) | セグメントが事前割り付けされるか (T)、または自動割り付けされるか (F) を定義 |
| data_offset | unsigned int | Sybase IQ のデータが開始する場所の、ロー・パーティションの開始点から見た相対的なバイト・ロケーション |

| カラム名 | カラム・タイプ | 説明 |
|---------------|--------------|-------------------------------------------|
| create_time | timestamp | ファイルの作成日時 |
| last_modified | timestamp | ファイルの最終変更日時 |
| read_write | char(1) | T は読み書き可能を示す |
| online | char(1) | T はオンラインを示す |
| create_txn_id | xact_id | その dbfile ファイルを作成したトランザクション ID |
| alter_txn_id | xact_id | 読み書きステータスを最後に変更したトランザクション ID |
| server_id | unsigned int | マルチプレックス・サーバの名前 |
| file_name | text | IQ DB 領域を開くためにマルチプレックス・サーバが使用する IQ DB 領域名 |

元になっているシステム・テーブルの制約

外部キー (server_id) が SYS.ISYSIQMPXSERVER を参照

ユニーク (server_id, file_name)

SYSIQDBSPACE システム・ビュー

ISYSIQDBSPACE からのグループ情報を読みやすい形式で表示します。

| カラム名 | カラム・タイプ | 説明 |
|---------------|-----------|-----------------------------------------|
| dbspace_id | small int | データベース内の各 DB 領域に与えられるユニークな番号 (DB 領域 ID) |
| last_modified | timestamp | 最後に DB 領域の読み書きステータスが変更された日時 |
| segment_type | char(8) | セグメントのタイプ : Main、Temp、Msg |
| read_write | char(1) | 'T' - 読み書き可能、 'F' - 読み取り専用 |

| カラム名 | カラム・タイプ | 説明 |
|----------------|--------------|-------------------------------------------------------------------|
| online | char(1) | 'T' – オンライン、'F' – オフライン |
| create_txn_id | xact_id | DB 領域を作成するトランザクション ID |
| alter_txn_id | xact_id | 読み書きステータスを最後に変更したトランザクション ID |
| striping_on | char(1) | 'T' – ディスク・ストライピングがオン、'F' – ディスク・ストライピングがオフ |
| stripe_size_kb | unsigned int | ディスク・ストライピング・アルゴリズムが次の dbfile に移動する前に、DB 領域の各ファイルに書き込まれるデータ量 (KB) |

元になっているシステム・テーブルの制約

プライマリ・キー (dbspace_id)

外部キー (dbspace_id) が SYS.ISYSDBSPACE(dbspace_id) を参照

SYSIQFILE システム・ビュー (廃止)

SYSIQFILE の代わりに SYSIQDBFILE システム・ビューが使用されます。詳細については、「[SYSIQDBFILE システム・ビュー](#)」(551 ページ)を参照してください。

SYSIQIDX システム・ビュー

ISYSIQIDX からのグループ情報を読みやすい形式で表示します。SYSIQIDX ビューの各ローは IQ インデックスを記述します。

注意 このビューは廃止になった SYSIQINDEX システム・ビューの代わりに使用されます。

| カラム名 | カラム・タイプ | 説明 |
|------------------------|---------------|-----------------------------------------------------------------------|
| table_id | unsigned int | このインデックスが適用されるテーブルを識別するユニークなテーブル番号 |
| index_id | unsigned int | 個々のテーブルの各インデックスに割り当てられたユニークなインデックス番号 |
| index_type | char(4) | インデックス・タイプ |
| index_owner | char(4) | インデックスの所有者 |
| max_key | unsigned int | 内部用 |
| identity_location | hs_vdorecid | 内部用 |
| identity_size | unsigned int | 内部用 |
| identity_location_size | unsigned int | 内部用 |
| link_table_id | unsigned int | 内部用 |
| link_index_id | unsigned int | 内部用 |
| delimited_by | varchar(1024) | (WD インデックスのみ) カラムの文字列を、そのカラムの WD インデックスに格納する単語に解析する際に使われるセパレータのリスト |
| limit | unsigned int | (WD インデックスのみ) WD インデックスの単語の最大長 |

元になっているシステム・テーブルの制約

プライマリ・キー (table_id、index_id)

外部キー (table_id、index_id) が SYS.ISYIDX を参照

外部キー (link_table_id、link_index_id、table_id、index_id) が SYS.ISYSIDX を参照

SYSIQINFO システム・ビュー

ISYSIQINFO からのグループ情報を読みやすい形式で表示します。

ISYSIQINFO システム・テーブルは、CREATE DATABASE を使って Sybase IQ データベースを作成したときに定義されたデータベースの特性を示します。このテーブルには常にローが 1 つしかありません。multiplex_name と last_multiplex_mode カラムは Sybase IQ 15.0 で追加されたものです。

| カラム名 | カラム・タイプ | 説明 |
|---------------------|-----------------------|---------------------------------------------------|
| create_time | TIMESTAMP NOT NULL | データベースの作成日時。 |
| update_time | TIMESTAMP NOT NULL | 前回の更新日時。 |
| file_format_version | UNSIGNED INT NOT NULL | このデータベースのファイルのファイル・フォーマット番号。 |
| cat_format_version | UNSIGNED INT NOT NULL | このデータベースのカatalog・フォーマット番号。 |
| sp_format_version | UNSIGNED INT NOT NULL | このデータベースのストアド・プロシージャ・フォーマット番号。 |
| block_size | UNSIGNED INT NOT NULL | データベースに指定されているブロック・サイズ。 |
| chunk_size | UNSIGNED INT NOT NULL | データベースに指定されているブロック・サイズとページ・サイズを基にした、ページあたりのブロック数。 |
| file_format_date | CHAR(10) NOT NULL | ファイル・フォーマット番号の最終更新日付。 |
| dbsig | BINARY(136) NOT NULL | カタログが内部で使用。 |
| commit_txn_id | xact_id | 内部用。 |
| rd_commit_txn_id | xact_id | 内部用。 |
| multiplex name | CHAR(128) NULL | このデータベースがメンバになっているマルチプレックスの名前。 |

| カラム名 | カラム・タイプ | 説明 |
|---------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| last_multiplex_mode | TINYINT NULL | (Sybase IQ 15.1 では使用されないカラム) カタログを読み書きアクセスで最後に開いたサーバのモード。次の値のいずれか。 <ul style="list-style-type: none"> • 0 – シングル・ノード • 1 – リーダ • 2 – コーディネータ • 3 – ライタ |

SYSIQJOINIDX システム・ビュー

ISYSIQJOINIDX からのグループ情報を読みやすい形式で表示します。SYSIQJOINIDX ビューの各ローは IQ ジョイン・インデックスを記述します。

注意 このビューは廃止になった SYSIQJOININDEX システム・ビューの代わりに使用されます。

| カラム名 | カラム・タイプ | 説明 |
|--------------------|--------------|-------------------------------------------|
| joinindex_id | unsigned int | 各ジョイン・インデックスには、プライマリ・キーになるユニークな番号が付いています。 |
| jvt_id | unsigned int | 内部用。 |
| dbspace_id | smallint | DB 領域の ID。 |
| joinindex_name | char(128) | ジョイン・インデックスの名前を定義します。 |
| joinindex_type | char(12) | 内部用。 |
| creator | unsigned int | ジョイン・インデックスを作成したユーザの番号。 |
| join_info_location | hs_vdorecid | 内部用。 |
| join_info_loc_size | unsigned int | 内部用。 |

| カラム名 | カラム・タイプ | 説明 |
|----------------|---------------------|-------------------------------------------------------------------|
| join_info_size | unsigned int | 内部用。 |
| block_map | hs_blockmapidentity | 内部用。 |
| block_map_size | unsigned int | 内部用。 |
| vdo | hs_vdoidentity | 内部用。 |
| vdo_size | unsigned_int | 内部用。 |
| commit_txn_id | xact_id | 内部用。 |
| txn_id | xact_id | 内部用。 |
| valid | char(1) | このジョイン・インデックスを同期する必要があるかどうかを示します。Y の場合は同期の必要はなく、N の場合は同期の必要があります。 |

元になっているシステム・テーブルの制約

プライマリ・キー (joinindex_id)

外部キー (jvt_id) が SYS.ISYSTAB を参照

外部キー (dbspace_id) が SYS.ISYSDBSPACE を参照

外部キー (creator) が SYS.ISYSUSER が参照

ユニーク (jvt_id, commit_txn_id, txn_id)

SYSIQJOININDEX システム・ビュー (廃止)

SYSIQJOININDEX の代わりに SYSIQJOINIDX システム・ビューが使用されます。詳細については、「[SYSIQJOINIDX システム・ビュー](#)」(556 ページ) を参照してください。

SYSIQJOINIXCOLUMN システム・ビュー

```
ALTER VIEW "SYS"."SYSIQJOINIXCOLUMN"
as select * from SYS.ISYSIQJOINIXCOLUMN
```

ISYSIQJOINIXCOLUMN からのグループ情報を読みやすい形式で表示します。SYSIQJOINIXCOLUMN ビューの各ローは IQ ジョイン・インデックスを記述します。

| カラム名 | カラム・タイプ | 説明 |
|-----------------|-----------------|--------------------------------------------------------------------------------------|
| joinindex_id | unsigned bigint | SYSIQJOINIDX のジョイン・インデックス値に対応します。 |
| left_table_id | unsigned int | ジョイン演算の左項に指定する SYSTAB のテーブル値に対応します。 |
| left_column_id | unsigned int | ジョイン演算の左項の一部として指定する SYSTABCOL のカラム値に対応します。 |
| join_type | char(4) | 現在サポートしている値は "=" だけです。 |
| right_table_id | unsigned int | ジョイン演算の右項に指定する SYSTAB のテーブル値に対応します。 |
| right_column_id | unsigned int | ジョイン演算の右項の一部として指定する SYSTABCOL のカラム値に対応します。 |
| order_num | unsigned int | 内部用。 |
| left_order_num | unsigned int | 内部用。 |
| right_order_num | unsigned int | 内部用。 |
| key_type | char(8) | キーに対するジョインのタイプを定義します。NATURAL はナチュラル・ジョイン、KEY はキー・ジョイン、ON は左外部ジョイン/右外部ジョイン/フル・ジョインです。 |
| coalesce | char(1) | 未使用。 |

元になっているシステム・テーブルの制約

プライマリ・キー (joinindex_id, left_table_id, left_column_id, right_table_id, right_column_id)

外部キー (joinindex_id) が SYS.ISYSIQJOINIDX を参照

外部キー (left_table_id, column_id) が SYS.ISYSTABCOL を参照

外部キー (right_table_id, column_id) が SYS.ISYSTABCOL を参照

SYSIQJOINIXTABLE システム・ビュー

```
ALTER VIEW "SYS"."SYSIQJOINIXTABLE"
as select * from SYS.ISYSIQJOINIXTABLE
```

SYSIQJOINIXTABLE からのグループ情報を読みやすい形式で表示します。SYSIQJOINIXTABLE ビューの各ローは IQ ジョイン・インデックスを記述します。

| カラム名 | カラム・タイプ | 説明 |
|--------------|-----------------|-----------------------------------|
| table_id | unsigned int | ジョイン演算に含まれる SYSTAB のテーブル値に対応します。 |
| joinindex_id | unsigned bigint | SYSIQJOINIDX のジョイン・インデックス値に対応します。 |
| active | unsigned int | ジョイン・インデックスでテーブルが使用される回数を定義します。 |

元になっているシステム・テーブルの制約

プライマリ・キー (table_id, joinindex_id)

外部キー (table_id) が SYS.ISYSTAB を参照

外部キー (joinindex_id) が SYS.ISYSIQJOINIDX を参照

SYSIQMPXLOGINPOLICYOPTION システム・ビュー

『Sybase IQ マルチプレックスの使用』の「付録 A マルチプレックス・リファレンス」の「SYSIQMPXLOGINPOLICYOPTION システム・ビュー」を参照してください。

SYSIQMPXSERVER システム・ビュー

『Sybase IQ マルチプレックスの使用』の「[付録 A マルチプレックス・リファレンス](#)」の「[SYSIQMPXSERVER システム・ビュー](#)」を参照してください。

SYSIQOBJECTS ASE 互換ビュー

SYSIQOBJECTS は各ローに個々のシステム・テーブル、ユーザ・テーブル、ビュー、プロシージャ、トリガ、イベント、ジョイン・インデックス、制約、ドメイン (sysdomain)、ドメイン (sysusertype)、カラム、インデックスを表示します。このビューは、dbo が所有します。詳細については、[表 8-1 \(592 ページ\)](#) を参照してください。

SYSIQPARTITIONCOLUMN システム・ビュー

```
ALTER VIEW "SYS"."SYSIQPARTITIONCOLUMN"
as select * from SYS.ISYSIQPARTITIONCOLUMN
```

ISYSIQPARTITIONCOLUMN からのグループ情報を読みやすい形式で表示します。SYSIQPARTITIONCOLUMN ビューの各ローで記述されているパーティションのカラムは、SYSPARTITIONSCHEME ビュー内で記述されている分割されたテーブル内の SYSIQPARTITION ビュー内で記述されているパーティション内のカラムです。SYSIQPARTITIONCOLUMN では、そのパーティションの DB 領域に保存されていないカラムのパーティションしか記述しません。

| カラム名 | カラム・タイプ | 説明 |
|-----------------------|-----------------|--------------------------------------|
| partitioned_object_id | unsigned bigint | 分割されたオブジェクト (テーブル) に割り当てられたユニークな ID。 |
| partition_id | unsigned int | 分割されたテーブル内のパーティションを識別します。 |
| column_id | unsigned int | カラムの ID。 |
| dbspace_id | smallint | パーティションの指定されたカラムが保存されている DB 領域の ID。 |

元になっているシステム・テーブルの制約

プライマリ・キー (partitioned_object_id, partition_id, column_id)

外部キー (partitioned_object_id, partition_id) が SYS.ISYSPARTITION を参照

外部キー (dbspace_id) が SYS.ISYSDBSPACE を参照

SYSIQTAB システム・ビュー

```
ALTER VIEW "SYS"."SYSIQTAB"
as select * from SYS.ISYSIQTAB
```

ISYSIQTAB からのグループ情報を読みやすい形式で表示します。
SYSIQTAB ビューの各ローは IQ テーブルを記述します。

注意 このビューは廃止になった SYSIQTABLE システム・ビューの代わりに使用されます。

| カラム名 | カラム・タイプ | 説明 |
|--------------------|---------------------|----------------------------------------------|
| table_id | unsigned int | 各テーブルには、プライマリ・キーになるユニークな番号 (テーブル番号) が付いています。 |
| block_map | hs_blockmapidentity | 内部用。 |
| block_map_size | unsigned int | 内部用。 |
| vdo | hs_vdoidentity | 内部用。 |
| vdo_id_size | unsigned int | 内部用。 |
| info_location | hs_vdorecid | 未使用。常にゼロ。 |
| info_recid_size | unsigned int | 未使用。常にゼロ。 |
| info_location_size | unsigned int | 未使用。常にゼロ。 |
| commit_txn_id | xact_id | 内部用。 |
| txn_id | xact_id | 内部用。 |
| join_id | unsigned int | 内部用。 |
| update_time | timestamp | IQ テーブルの最終変更日時。 |

元になっているシステム・テーブルの制約

プライマリ・キー (table_id)

SYSIQTABCOL システム・ビュー

```
ALTER VIEW "SYS"."SYSIQTABCOL"
as select * from SYS.ISYSIQTABCOL
```

ISYSIQTABCOL からのグループ情報を読みやすい形式で表示します。
SYSIQTABCOL ビューの各ローは IQ テーブル内のカラムを記述します。

注意 このビューは廃止になった SYSIQCOLUMN システム・ビューの代わりに使用されます。

| カラム名 | カラム・タイプ | 説明 |
|---------------------|--------------|-------------------------------------------------------------------------------------------|
| table_id | unsigned int | このカラムが属するテーブルを識別するユニークなテーブル番号。これは、SYSTAB の table_id カラムに対応します。 |
| column_id | unsigned int | 各テーブルでは、カラムに 1 から始まる番号が付きます。 select * from table コマンドを実行すると、このカラム番号の順番でカラムが表示されます。 |
| link_table_id | unsigned int | 内部用。 |
| link_column_id | unsigned int | 内部用。 |
| max_length | unsigned int | カラムで使用できる最大長を示します。 |
| approx_unique_count | rowid | このカラムのユニークな値(カーディナリティ)の概数。 |
| cardinality | rowid | このカラムのユニークな値(カーディナリティ)の実数。 |
| has_data | char(1) | カラムにデータがあるかどうかを示します (T/F)。 |
| has_original | char(1) | ジョイン・インデックスにオリジナル・データがあるかどうかを示します (T/F)。 |

| カラム名 | カラム・タイプ | 説明 |
|-------------------|---------|------------------------------------------------------------|
| original_not_null | char(1) | オリジナル・データがあるジョイン・インデックス・カラムが NOT NULL であったかどうかを示します (T/F)。 |
| original_unique | char(1) | オリジナル・データがあるジョイン・インデックス・カラムが UNIQUE であったかどうかを示します (T/F)。 |

元になっているシステム・テーブルの制約

プライマリ・キー (table_id)

SYSIQTABLE システム・ビュー (廃止)

SYSIQTABLE の代わりに SYSIQTAB システム・ビューが使用されます。詳細については、「[SYSIQTAB システム・ビュー](#)」(561 ページ)を参照してください。

SYSIQVINDEXT ASE 互換ビュー

SYSIQVINDEXT では、FP IQ 以外の個々のインデックスに 1 つのローが割り当てられます。このビューは、DBO が所有します。詳細については、[表 8-1](#) (592 ページ)を参照してください。

SYSIXCOL 互換ビュー (廃止)

SYSIXCOL ビューは、SYSIXCOL システム・テーブルを使用していた旧バージョンの Sybase IQ との互換性を維持するために提供されています。ただし、SYSIXCOL システム・テーブルの代わりに ISYSIDXCOL システム・テーブルとそれに対応する SYSIDXCOL システム・ビューが使用されます。

SYSIXCOL ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSIXCOL compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5249307.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-5249307.html) を参照してください。

SYSJAR システム・ビュー

SYSJAR システム・ビューの各ローはデータベースに保存されている JAR ファイルを定義します。このビューの元になっているシステム・テーブルは ISYSJAR です。

SYSJAR ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSJAR system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysjar345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysjar345.html) を参照してください。

SYSJARCOMPONENT システム・ビュー

SYSJARCOMPONENT システム・ビューの各ローは JAR ファイルの個々のコンポーネントの定義です。このビューの元になっているシステム・テーブルは ISYSJARCOMPONENT です。

SYSJARCOMPONENT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSJARCOMPONENT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysjarcomponent345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysjarcomponent345.html) を参照してください。

SYSJAVACLASS システム・ビュー

SYSJAVACLASS システム・ビューの各ローはそのデータベースに保存されている個々の Java クラスを記述します。このビューの元になっているシステム・テーブルは ISYSJAVACLASS です。

SYSJAVACLASS ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSJAVACLASS system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-sectb-4248641.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-sectb-4248641.html) を参照してください。

SYSLOGINMAP システム・ビュー

SYSLOGINMAP システム・ビューでは、統合ログインまたは Kerberos ログインを使ってデータベースに接続できる個々のユーザに 1 つのローが割り当てられています。セキュリティ上の理由から、このテーブルの内容は DBA 権限を保有するユーザだけが表示できます。このビューの元になっているシステム・テーブルは ISYSLOGINMAP です。

SYSLOGINMAP ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSLOGINMAP system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syslogin34s5.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syslogin34s5.html) を参照してください。

SYSLOGINPOLICY システム・ビュー

このビューの元になっているシステム・テーブルは ISYSLOGINPOLICY です。

SYSLOGINPOLICY ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSLOGINPOLICY system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-s-5437600.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-s-5437600.html) を参照してください。

SYSLOGINPOLICYOPTION システム・ビュー

このビューの元になっているシステム・テーブルは ISYSLOGINPOLICYOPTION です。

SYSLOGINPOLICYOPTION ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSLOGINPOLICYOPTION system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-s-5437600.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/rf-views-s-5437600.html) を参照してください。

SYSLOGINS ASE 互換ビュー

このビューの所有者は DBO というユーザです。syslogins の各ローは、有効な Adaptive Server の各ユーザ・アカウントです。詳細については、[表 8-2 \(594 ページ\)](#) を参照してください。

SYSMVOPTION システム・ビュー

SYSMVOPTION システム・ビューの各ローは、マテリアライズド・ビューの作成時に設定された各オプション値を記述します。ただし、この記述にオプション名は含まれません。このビューの元になっているシステム・テーブルは ISYSMVOPTION です。

SYSMVOPTION ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSMVOPTION system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792807.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792807.html) を参照してください。

SYSMVOPTIONNAME システム・ビュー

SYSMVOPTIONNAME システム・ビューの各ローは SYSMVOPTION システム・ビューに定義のあるオプションの名前です。このビューの元になっているシステム・テーブルは ISYSMVOPTIONNAME です。

SYSMVOPTIONNAME ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSMVOPTIONNAME system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407925433807.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407925433807.html) を参照してください。

SYSOBJECT システム・ビュー

SYSOBJECT システム・ビューの各ローは個々のデータベース・オブジェクトを記述します。このビューの元になっているシステム・テーブルは ISYSOBJECT です。

SYSOBJECT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSOBJECT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407967862807.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407967862807.html) を参照してください。

SYSOBJECTS ASE 互換ビュー

このビューの所有者は DBO というユーザです。sysobjects の各ローは、個々のテーブル、ビュー、ストアド・プロシージャ、拡張ストアド・プロシージャ、ログ、規則、デフォルト、トリガ、検査制約、参照制約、計算カラム、機能ベースのインデックス・キー、テンポラリ・オブジェクト、その他の形式のコンパイル済みオブジェクトです。オブジェクト・タイプが N の場合は、パーティション条件 ID のローも含まれます。詳細については、[表 8-1 \(592 ページ\)](#) を参照してください。

SYSOPTION システム・ビュー

SYSOPTION システム・ビューの各ローはデータベースに保存されている各オプション設定です。各ユーザはオプションごとに自分の設定を保存できます。また、PUBLIC ユーザ ID に対する設定は、自分の設定を持たないユーザのデフォルト設定として使用されます。このビューの元になっているシステム・テーブルは ISYSOPTION です。

SYSOPTION ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSOBJECT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysoption345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysoption345.html) を参照してください。

SYSOPTIONS 統合ビュー

SYSOPTIONS ビューの各ローは、SET コマンドを使って作成した各オプションです。各ユーザはオプションごとに自分の設定を保存できます。また、PUBLIC ユーザに対する設定は、自分の設定を持たないユーザのデフォルト設定として使用されます。

SYSOPTIONS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSOPTIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvopt.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvopt.html) を参照してください。

SYSOPTSTAT システム・ビュー

SYSOPTSTAT システム・ビューには、ALTER DATABASE CALIBRATE 文によって計算されたコスト・モデル調整情報が入ります。このビューの内容は内部使用専用であり、sa_get_dtt システム・プロシージャによってアクセスするのが最も良い方法です。このビューの元になっているシステム・テーブルは ISYSOPTSTAT です。

SYSOPTSTAT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSOPTSTAT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-3753748.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-3753748.html) を参照してください。

SYSPARTITION システム・ビュー

```
ALTER VIEW "SYS"."SYSPARTITION"
as select * from SYS.ISYSPARTITION
```

ISYSPARTITION からのグループ情報を読みやすい形式で表示します。SYSPARTITION ビューの各ローは、データベース内の分割された個々のオブジェクト (テーブルやインデックス) です。

| カラム名 | カラム・タイプ | 説明 |
|-----------------------|-----------------|--------------------------------------------------------------------------------|
| partitioned_object_id | unsigned bigint | 分割されたオブジェクト(テーブル)に割り当てられたユニークな ID。 |
| partition_id | unsigned int | 分割されたテーブル内のパーティションを識別します。 |
| partition_object_id | unsigned bigint | 各テーブル・パーティションはそれぞれが1つのオブジェクトなので、テーブル・オブジェクトまたはインデックス・オブジェクトからユニークな番号が割り当てられます。 |
| partition_values | long varchar | この範囲パーティションの上限です。 |
| position | unsigned int | パーティションの元の番号。 |
| partition_name | char(128) | パーティションの名前。 |

元になっているシステム・テーブルの制約

プライマリ・キー (partitioned_object_id, partition_id)

ユニーク (partition_object_id, position)

外部キー (partition_object_id) が SYS.ISYSOBJECT を参照

外部キー (partitioned_object_id) が SYS.ISYSOBJECT を参照

SYSPARTITIONKEY システム・ビュー

```
ALTER VIEW "SYS"."SYSPARTITIONKEY"
as select * from SYS.ISYSPARTITIONKEY
```

ISYSPARTITIONKEY からのグループ情報を読みやすい形式で表示します。SYSPARTITIONKEY ビューの各ローは、データベース内の分割された個々のオブジェクト (テーブルやインデックス) です。

| カラム名 | カラム・タイプ | 説明 |
|-----------------------|-----------------|-----------------------------------------------------------------|
| partitioned_object_id | unsigned bigint | 分割された各オブジェクト (テーブル) にはユニークなオブジェクト番号が付いています。 |
| column_id | unsigned int | カラム ID はパーティション・キーの一部となってテーブル・カラムを識別します。 |
| position | smallint | パーティション・キー内でのこのカラムの位置。0 から始まります。位置 0 はそのパーティション・キーの 1 番目のカラムです。 |

元になっているシステム・テーブルの制約

プライマリ・キー (partitioned_object_id, column_id)

外部キー (partitioned_object_id) が SYS.ISYSOBJECT を参照

SYSPARTITIONSCHEME システム・ビュー

```
ALTER VIEW "SYS"."SYSPARTITIONSCHEME"
as select * from SYS.ISYSPARTITIONSCHEME
```

ISYSPARTITIONSCHEME からのグループ情報を読みやすい形式で表示します。SYSPARTITIONSCHEME ビューの各ローは、データベース内の分割された各オブジェクト (テーブルやインデックス) です。

| カラム名 | カラム・タイプ | 説明 |
|-----------------------|-----------------|---------------------------------------|
| partitioned_object_id | unsigned bigint | 分割された各オブジェクト (テーブル) にはユニークな番号が付いています。 |

| カラム名 | カラム・タイプ | 説明 |
|---------------------|---------|---------------------------------------------------------------|
| partition_method | tinyint | このテーブルの分割方法。有効な値 1- 範囲分割 Sybase IQ は範囲分割しかサポートし ません。 |
| subpartition_method | tinyint | 将来の使用に備えて予約されている。 |

元になっているシステム・テーブルの制約

プライマリ・キー (partitioned_object_id)

外部キー (partitioned_object_id) が SYS.ISYSOBJECT を参照

SYSphysIDX システム・ビュー

SYSphysIDX システム・ビューの各ローはデータベースの個々の物理インデックスを定義します。このビューの元になっているシステム・テーブルは ISYSphysIDX です。

SYSphysIDX ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSphysIDX system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407d92806.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407d92806.html) を参照してください。

SYSprocauth 統合ビュー

SYSprocauth ビューの各ローは、個々のプロシージャに与えられた権限のセットを記述します。代わりに SYSprocpem システム・ビューを使用することもできます。

SYSprocauth ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSprocauth consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysprocauth.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysprocauth.html) を参照してください。

SYSPROCEDURE システム・ビュー

SYSPROCEDURE システム・ビューの各ローは、そのデータベースに保存されている個々のプロシージャを記述します。このビューの元になっているシステム・テーブルは ISYSPROCEDURE です。

SYSPROCEDURE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPROCEDURE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysproc345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysproc345.html) を参照してください。

SYSPROCPARM システム・ビュー

SYSPROCPARM システム・ビューの各ローは、そのデータベースに保存されている個々のプロシージャの個々のパラメータを記述します。このビューの元になっているシステム・テーブルは ISYSPROCPARM です。

SYSPROCPARM ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPROCPARM system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syprpar345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syprpar345.html) を参照してください。

SYSPROCPARMS 統合ビュー

SYSPROCPARMS システム・ビューの各ローは、そのデータベースに保存されている個々のプロシージャの個々のパラメータを記述します。

SYSPROCPARMS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPROCPARMS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvrpar.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvrpar.html) を参照してください。

SYSPROCPerm システム・ビュー

SYSPROCPerm システム・ビューの各ローは、プロシージャを実行するパーミッションが与えられている個々のユーザを記述します。プロシージャを呼び出すことができるのは、パーミッションを付与されたユーザだけです。このビューの元になっているシステム・テーブルは ISYSPROCPerm です。

SYSPROCPerm ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPROCPerm system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysprper345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysprper345.html) を参照してください。

SYSPROCCons 統合ビュー

SYSPROCCons ビューには、プロシージャまたは関数の名前、その作成者の名前、そのプロシージャまたは関数に記録されたコメントが表示されます。

SYSPROCCons ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPROCCons consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4771349.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4771349.html) を参照してください。

SYSPROXYTAB システム・ビュー

SYSPROXYTAB システム・ビューの各ローは、個々のプロキシ・テーブルのリモート・パラメータを記述します。このビューの元になっているシステム・テーブルは ISYSPROXYTAB です。

SYSPROXYTAB ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPROXYTAB system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792808.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792808.html) を参照してください。

SYSPUBLICATION システム・ビュー

SYSPUBLICATION システム・ビューの各ローは、個々の SQL Remote または MobiLink のパブリケーションを記述します。このビューの元になっているシステム・テーブルは ISYSPUBLICATION です。

SYSPUBLICATION ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPUBLICATION system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syspub345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syspub345.html) を参照してください。

SYSPUBLICATIONS 統合ビュー

SYSPUBLICATIONS システム・ビューの各ローは、個々の SQL Remote または MobiLink のパブリケーションを記述します。

SYSPUBLICATIONS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSPUBLICATIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syspublications.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syspublications.html) を参照してください。

SYSDATABASE システム・ビュー

SYSDATABASE システム・ビューの各ローは、オブジェクトに付けた個々の注釈 (コメント) を記述します。このビューの元になっているシステム・テーブルは ISYSDATABASE です。

SYSDATABASE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSDATABASE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792809.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-40792809.html) を参照してください。

SYSREMOTEOPTION システム・ビュー

SYSREMOTEOPTION システム・ビューの各ローは、SQL Remote メッセージの個々のリンク・パラメータの値を記述します。このビューの元になっているシステム・テーブルは ISYSREMOTEOPTION です。

このビューには、一般に公開すべきではないデータが含まれている可能性のあるカラムもあります。そのため、このビューへのアクセスは DBA 権限を持つユーザに限定されます。SYSREMOTEOPTION2 ビューは、このビューにあるデータへのパブリック・アクセスを提供しますが、一般に公開すべきではないカラムは除外されます。

SYSREMOTEOPTION ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の **SYSREMOTEOPTION system view** (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremoteoption-systab345.html) を参照してください。

SYSREMOTEOPTION2 統合ビュー

SYSREMOTEOPTION と SYSREMOTEOPTIONTYPE のカラムを読みやすい形式で表示します。一般に公開すべきではないカラムは除外されます。

SYSREMOTEOPTION2 ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の **SYSREMOTEOPTION2 consolidated view** (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysviews-s-3703906.html) を参照してください。

SYSREMOTEOPTIONS 統合ビュー

SYSREMOTEOPTIONS ビューの各ローは、SQL Remote メッセージの個々のリンク・パラメータの値を記述します。このビューには、一般に公開すべきではないデータが含まれている可能性のあるカラムもあります。そのため、このビューへのアクセスは DBA 権限を持つユーザに限定されます。SYSREMOTEOPTION2 ビューは、公開しても問題のないデータへのパブリック・アクセスを提供します。

SYREMOTEOPTIONS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYREMOTEOPTIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotEOPTIONS-system-sysviews.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotEOPTIONS-system-sysviews.html) を参照してください。

SYREMOTEOPTIONTYPE システム・ビュー

SYREMOTEOPTIONTYPE システム・ビューの各ローは、SQL Remote メッセージの個々のリンク・パラメータを記述します。このビューの元になっているシステム・テーブルは ISYREMOTEOPTIONTYPE です。

SYREMOTEOPTIONTYPE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYREMOTEOPTIONTYPE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotEOPTIONTYPE-systab345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotEOPTIONTYPE-systab345.html) を参照してください。

SYREMOTETYPE システム・ビュー

SYREMOTETYPE システム・ビューには、SQL Remote に関する情報があります。このビューの元になっているシステム・テーブルは ISYREMOTETYPE です。

SYREMOTETYPE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYREMOTETYPE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotetype345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotetype345.html) を参照してください。

SYREMOTETYPES 統合ビュー

SYREMOTETYPES ビューの各ローは、個々の SQL Remote メッセージ・タイプを記述します。パブリッシャのアドレスも含まれます。

SYSREMOTETYPES ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSREMOTETYPES consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotetypes.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysremotetypes.html) を参照してください。

SYSREMOTEUSER システム・ビュー

SYSREMOTEUSER の各ローは、REMOTE パーミッションを持つユーザの ID (サブスクライバ) と、そのユーザが送受信した SQL Remote メッセージのステータスを記述します。このビューの元になっているシステム・テーブルは ISYSREMOTEUSER です。

SYSREMOTEUSER ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSREMOTEUSER system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysrem345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysrem345.html) を参照してください。

SYSREMOTEUSERS 統合ビュー

SYSREMOTEUSERS ビューの各ローは、REMOTE パーミッションを持つユーザの ID (サブスクライバ) と、そのユーザが送受信した SQL Remote メッセージのステータスを記述します。

SYSREMOTEUSERS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSREMOTEUSERS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvremusr.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvremusr.html) を参照してください。

SYSSCHEDULE システム・ビュー

SYSSCHEDULE システム・ビューの各ローには、CREATE EVENT の SCHEDULE 句で指定された、イベントが発生する時刻が記述されています。このビューの元になっているシステム・テーブルは ISYSSCHEDULE です。

SYSSCHEDULE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の SYSSCHEDULE system view (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-sectb-4249019.html) を参照してください。

SYSSERVER システム・ビュー

SYSSERVER システム・ビューの各ローはリモート・サーバを記述します。このビューの元になっているシステム・テーブルは ISYSSERVER です。

SYSSERVER ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の SYSSCHEDULE system view (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssservers345.html) を参照してください。

SYSSOURCE システム・ビュー

SYSSOURCE システム・ビューの各ローは、SYSOBJECT システム・ビューにある個々のオブジェクトのソース・コード(あれば)です。

SYSSOURCE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の SYSSOURCE system view (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928010.html) を参照してください。

SYSSQLSERVERTYPE システム・ビュー

SYSSQLSERVERTYPE システム・ビューには、Adaptive Server Enterprise との互換性に関する情報が含まれます。このビューの元になっているシステム・テーブルは ISYSSQLSERVERTYPE です。

SYSSQLSERVERTYPE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSQLSERVERTYPE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssqlservertype345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssqlservertype345.html) を参照してください。

SYSSUBPARTITIONKEY システム・ビュー

このビューは将来の使用に備えて予約されています。Sybase IQ 15.1 はサブパーティションをサポートしません。

SYSSUBSCRIPTION システム・ビュー

SYSSUBSCRIPTION システム・ビューの各ローは、REMOTE パーミッションを持つユーザ ID からの、個々のパブリケーションに対するサブスクリプションを記述します。このビューの元になっているシステム・テーブルは ISYSSUBSCRIPTION です。

SYSSUBSCRIPTION ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSUBSCRIPTION system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssub345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssub345.html) を参照してください。

SYSSUBSCRIPTIONS 統合ビュー

各ローは、REMOTE パーミッションを持つユーザ ID からの、個々のパブリケーションに対するサブスクリプションを記述します。

SYSSUBSCRIPTIONS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSUBSCRIPTIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssubscriptions.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssubscriptions.html) を参照してください。

SYSSYNC システム・ビュー

SYSSYNC システム・ビューには MobiLink 同期に関する情報が含まれています。このビューには、一般に公開すべきではないデータが含まれている可能性のあるカラムもあります。そのため、このビューへのアクセスは DBA 権限を持つユーザーに限定されます。SYSSYNC2 ビューは、このビューにあるデータへのパブリック・アクセスを提供しますが、一般に公開すべきではないカラムは除外されます。このビューの元になっているシステム・テーブルは ISYSSYNC です。

SYSSYNC ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNC system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssync345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssync345.html) を参照してください。

SYSSYNC2 統合ビュー

SYSSYNC2 ビューは、SYSSYNC システム・ビューにあるデータ (MobiLink 同期に関連した情報) へのパブリック・アクセスを提供します。一般に公開すべきではないデータは公開されません。

SYSSYNC2 ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNC2 consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysviews-s-3675006.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysviews-s-3675006.html) を参照してください。

SYSSYNCPUBLICATIONDEFAULTS 統合ビュー

SYSSYNCPUBLICATIONDEFAULTS ビューは、MobiLink 同期に関するパブリケーションで使用されるデフォルト同期設定を表示します。

SYSSYNCPUBLICATIONDEFAULTS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNCPUBLICATIONDEFAULTS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssyncpublications.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssyncpublications.html) を参照してください。

SYSSYNCS 統合ビュー

SYSSYNCS ビューには MobiLink 同期に関する情報が含まれています。このビューには、一般に公開すべきではないデータが含まれている可能性のあるカラムもあります。そのため、このビューへのアクセスは DBA 権限を持つユーザに限定されます。

SYSSYNCS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNCS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssyncs.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssyncs.html) を参照してください。

SYSSYNCSRIPT システム・ビュー

SYSSYNCSRIPT システム・ビューの各ローは、MobiLink のスクリプトによるアップロードで使用する個々のストアド・プロシージャを示します。このビューは SYSSYNCSRIPTS ビューとほとんど同じですが、このビューでは値が生の形式で表示されます。人間が読み取ることのできるフォーマットで表示するには、「[SYSSYNCSRIPTS 統合ビュー](#)」を参照してください。

SYSSYNCSRIPT ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNCSRIPT system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928011.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928011.html) を参照してください。

SYSSYNCSCRIPTS 統合ビュー

SYSSYNCSCRIPTS ビューの各ローは、MobiLink のスクリプトによるアップロードで使用する個々のストアド・プロシージャを示します。このビューは SYSSYNCSCRIPT システム・ビューとほとんど同じですが、このビューでは、生データではなく、人間が読み取ることのできるフォーマットで表示されます。

SYSSYNCSCRIPTS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNCSCRIPTS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4079d28011.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4079d28011.html) を参照してください。

SYSSYNCSUBSCRIPTIONS 統合ビュー

SYSSYNCSUBSCRIPTIONS ビューには、MobiLink 同期サブスクリプションに関連する同期の設定が含まれています。

SYSSYNCSUBSCRIPTIONS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNCSUBSCRIPTIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssynsubscriptions.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syssynsubscriptions.html) を参照してください。

SYSSYNCUSERS 統合ビュー

MobiLink 同期ユーザに関連した同期の設定を表示します。

SYSSYNCUSERS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSSYNCUSERS system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928011.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928011.html) を参照してください。

SYSTAB システム・ビュー

SYSTAB システム・ビューの各ローはそのデータベース内の個々のテーブルまたはビューを記述します。ビューについての追加情報は、SYSVIEW システム・ビューの説明を参照してください。このビューの元になっているシステム・テーブルは ISYSTAB です。

SYSTAB ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTAB system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928012.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928012.html) を参照してください。

SYSTABLE 互換ビュー (廃止)

SYSTABLE ビューは、SYSTABLE システム・テーブルを使用していた旧バージョンの Sybase IQ との互換性を維持するために提供されています。ただし、SYSTABLE システム・テーブルの代わりに ISYSTAB システム・テーブルとそれに対応する SYSTAB システム・ビューが使用されています。

SYSTABLE ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTABLE compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere/dbreference_en11/views-s-5747733.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere/dbreference_en11/views-s-5747733.html) を参照してください。

SYSTABAUTH 統合ビュー

SYSTABAUTH ビューには、SYSTABLEPERM システム・ビューからの情報が読みやすい形式で表示されます。

SYSTABAUTH ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTABAUTH consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvtaba.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvtaba.html) を参照してください。

SYSTABCOL システム・ビュー

SYSTABCOL システム・ビューには、データベース内の各テーブルとビューに対して 1 つのローがあります。このビューの元になっているシステム・テーブルは ISYSTABCOL です。

SYSTABCOL ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTABCOL system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscolumn345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syscolumn345.html) を参照してください。

SYSTABLEPERM システム・ビュー

GRANT 文によって与えられたパーミッションは SYSTABLEPERM システム・ビューに保存されます。このビューの各ローは、1 つのテーブル、パーミッションを付与する 1 ユーザ ID (grantor)、そしてパーミッションを付与された 1 ユーザ ID (grantee) に対応しています。このビューの元になっているシステム・テーブルは ISYSTABLEPERM です。

SYSTABLEPERM ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTABLEPERM system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systabperm345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systabperm345.html) を参照してください。

SYSTEXTCONFIG システム・ビュー

SYSTEXTCONFIG システム・ビューの各ローは、完全テキスト検索で使用する個々のテキスト設定オブジェクトを記述します。このビューの元になっているシステム・テーブルは ISYSTEXTCONFIG です。

SYSTEXTCONFIG ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTEXTCONFIG system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systextconfi.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systextconfi.html) を参照してください。

SYSTEXTIDX システム・ビュー

SYSTEXTIDX システム・ビューの各ローは、個々のテキスト・インデックスを記述します。このビューの元になっているシステム・テーブルは ISYSTEXTIDX です。

SYSTEXTIDX ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTEXTIDX system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systextidx.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systextidx.html) を参照してください。

SYSTEXTIDXTAB システム・ビュー

SYSTEXTIDXTAB システム・ビューの各ローは、テキスト・インデックスの一部になっている個々の生成されたテーブルを記述します。このビューの元になっているシステム・テーブルは ISYSTEXTIDXTAB です。

SYSTEXTIDXTAB ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTEXTIDXTAB system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systextidxtab.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systextidxtab.html) を参照してください。

SYSTRIGGER システム・ビュー

SYSTRIGGER システム・ビューの各ローは、そのデータベース内の個々のトリガを記述します。このビューは、参照トリガ・アクション (ON DELETE CASCADE など) を持つ、外部キー定義に対して自動的に作成されたトリガも含みます。このビューの元になっているシステム・テーブルは ISYSTRIGGER です。

SYSTRIGGER ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTRIGGER system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systrig345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systrig345.html) を参照してください。

SYSTRIGGERS 統合ビュー

SYSTRIGGERS ビューの各ローは、そのデータベース内の個々のトリガを記述します。このビューは、参照トリガ・アクション (ON DELETE CASCADE など) を持つ、外部キー定義に対して自動的に作成されたトリガも含みます。

SYSTRIGGERS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTRIGGERS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvtrig.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvtrig.html) を参照してください。

SYSTYPEMAP システム・ビュー

SYSTYPEMAP システム・ビューには、「[SYSSQLSERVERTYPE システム・ビュー](#)」のエントリに対する互換マッピング値があります。このビューの元になっているシステム・テーブルは ISYSTYPEMAP です。

SYSTYPEMAP ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSTYPEMAP system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systypemap345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/systypemap345.html) を参照してください。

SYSTYPES ASE 互換ビュー

このビューの所有者は DBO というユーザです。systypes の各ローには、システム提供の各データ型とユーザ定義の各データ型が入っています。ドメイン (規則によって定義された) とデフォルトがあれば、指定されています。システム提供のデータ型を記述するローは変更できません。詳細については、[表 8-1 \(592 ページ\)](#) を参照してください。

SYSUSER システム・ビュー

SYSUSER システム・ビューの各ローは、そのシステム内の個々のトリガを記述します。このビューの元になっているシステム・テーブルは ISYSUSER です。

SYSUSER ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSER system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928014.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928014.html) を参照してください。

SYSUSERAUTH 互換ビュー (廃止)

SYSUSERAUTH ビューは古いバージョンの Sybase IQ との互換性を維持するために提供されています。代わりに SYSUSERAUTHORITY システム・ビューを使ってください。

SYSUSERAUTH ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSERAUTH compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvauth.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvauth.html) を参照してください。

SYSUSERAUTHORITY システム・ビュー

SYSUSERAUTHORITY システム・ビューの各ローは、ユーザ ID とそれに与えられている個々の権限を記述します。このビューの元になっているシステム・テーブルは ISYSUSERAUTHORITY です。

SYSUSERAUTHORITY ビューは SQL Anywhere のシステム・ビューです。詳細については、[SYSUSERAUTHORITY system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928015.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-407928015.html) を参照してください。

SYSUSERLIST 互換ビュー (廃止)

SYSUSERLIST ビューは古いバージョンの Sybase IQ との互換性を維持するために提供されています。SYSUSERLIST ビューの各ローは、ユーザを記述しますが、ユーザ ID とパスワードは公開されません。ユーザはユーザ名によって識別されます。

SYSUSERLIST ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSERLIST compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syul.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syul.html) を参照してください。

SYSUSERMESSAGE システム・ビュー

SYSUSERMESSAGE システム・ビューの各ローは、エラー条件に対する個々のユーザ定義メッセージです。このビューの元になっているシステム・テーブルは ISYSUSERMESSAGE です。

SYSUSERMESSAGE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSERMESSAGE system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysusermessages345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysusermessages345.html) を参照してください。

SYSUSEROPTIONS 統合ビュー

SYSUSEROPTIONS ビューには、各ユーザの有効なオプション設定が入っています。ユーザ固有の設定がないオプションについては、そのオプションのパブリック設定が表示されます。

SYSUSEROPTIONS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSEROPTIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvuopt.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvuopt.html) を参照してください。

SYSUSERPERM 互換ビュー (廃止)

このビューは、旧バージョンでの権限とパーミッションしか表示しないので、廃止になりました。代わりに SYSUSERAUTHORITY システム・ビューを使用するようにアプリケーションを変更してください。SYSUSERPERM ビューの各ローは個々のユーザ ID を記述します。

SYSUSERPERM ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSERPERM compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysup345.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysup345.html) を参照してください。

SYSUSERPERMS 互換ビュー (廃止)

このビューは、旧バージョンでの権限とパーミッションしか表示しないので、廃止になりました。代わりに SYSUSERAUTHORITY システム・ビューを使用するようにアプリケーションを変更してください。SYSUSERPERM ビューと同様に、SYSUSERPERMS ビューの各ローは、個々のユーザ ID を記述しますが、パスワード情報は含まれていません。このビューはすべてのユーザが読み取りを許可されます。

SYSUSERPERMS ビューは SQL Anywhere の互換ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSERPERMS compatibility view \(deprecated\) \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvuper.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvuper.html) を参照してください。

SYSUSERTYPE システム・ビュー

SYSUSERTYPE システム・ビューの各ローは、ユーザ定義のデータ型の記述です。このビューの元になっているシステム・テーブルは ISYSUSERTYPE です。

SYSUSERTYPE ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSUSEROPTIONS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvuopt.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvuopt.html) を参照してください。

SYSUSERS ASE 互換ビュー

このビューの所有者は DBO というユーザです。sysusers には、そのデータベースの使用を許可されている各ユーザのローと各グループまたはロールのローがあります。詳細については、[表 8-1 \(592 ページ\)](#) を参照してください。

SYSVIEW システム・ビュー

SYSVIEW システム・ビューの各ローは、そのデータベース内の個々のビューを記述します。ビューの詳細については、「[SYSTAB システム・ビュー](#)」を参照してください。このビューの元になっているシステム・テーブルは ISYSVIEW です。

マテリアライズド・ビューの情報を読みやすい形式で表示するには、sa_materialized_view_info システム・プロシージャも使用できます。

SYSVIEW ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSVIEW system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4079280.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4079280.html) を参照してください。

SYSVIEWS 統合ビュー

SYSVIEWS ビューの各ローは、個々のビューとその定義を記述します。

SYSVIEWS ビューは SQL Anywhere の統合ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSVIEWS consolidated view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvviews.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/syvviews.html) を参照してください。

SYSWEBSERVICE システム・ビュー

SYSWEBSERVICE システム・ビューの各ローは、個々の Web サービスを記述します。

SYSWEBSERVICE ビューは SQL Anywhere のシステム・ビューです。詳細については、『SQL Anywhere Server SQL リファレンス・マニュアル』の [SYSVIEW system view \(http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4079280.html\)](http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/views-s-4079280.html) を参照してください。

Transact-SQL 互換のビュー

Adaptive Server Enterprise および Sybase IQ には、両製品の用途の違いを反映して別々のシステム・カタログがあります。

Adaptive Server Enterprise の場合、1 つのサーバ内に 1 つの master データベースが存在します。master データベースは、サーバ内にある全てのデータベースに適用される情報を格納する一連のシステム・テーブルで構成されています。master データベース内には多数のデータベースがあり、各データベースにはそれに関連する複数のシステム・テーブルがあります。

Sybase IQ では、各データベースは独立して存在し、独自のシステム・テーブルを持っています。データベースの集合体に関するシステム情報を持つ master データベースはありません。各サーバは、必要に応じて各データベースを動的にロードしたりアンロードしたりしながら、複数のデータベースを同時に実行できます。

Adaptive Server Enterprise と Sybase IQ のシステム・カタログは異なります。Adaptive Server Enterprise のシステム・テーブルおよびビューは、特殊なユーザ `dbo` が所有し、一部は `master` データベース内に、一部は `sybsecurity` データベース内に、一部は個別のデータベース内に、と分散しています。一方、Sybase IQ のシステム・テーブルおよびビューは、特殊なユーザ `SYS` が所有し、各データベース内に別々に存在します。

互換性のあるアプリケーションを作成できるように、Sybase IQ には特殊ユーザ `dbo` が所有する一連のビュー (Adaptive Server Enterprise のシステム・テーブルおよびビューに対応する) が用意されています。構造上の違いのために、Sybase IQ のコンテキストではその内容が無意味になる Adaptive Server Enterprise のテーブルまたはビューに対応するビューは、カラム名とデータ型だけの空のビューになります。

表 8-1、表 8-2、表 8-3 は、Adaptive Server Enterprise のシステム・テーブルと、Sybase IQ システム・カタログ内での実装を示します。すべてのテーブルの所有者は、各 DBMS の `dbo` です。

各 Adaptive Server Enterprise データベースに存在するテーブル

表 8-1 : 各 ASE データベース内のテーブル

| テーブル名 | 説明 | データ? | IQによるサポート? |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|------|------------|
| <code>sysalternates</code> | データベース・ユーザにマップされたユーザごとのロー。 | なし | なし |
| <code>syscolumns</code> | テーブルまたはビューのカラムごとのロー、プロシージャのパラメータごとのロー。 Sybase IQ では、クエリするときに所有者名として <code>dbo</code> (つまり、 <code>dbo.syscolumns</code>) を使用してください。 | あり | あり |
| <code>syscomments</code> | ビュー、ルール、デフォルト、プロシージャごとに 1 ロー (複数ローにまたがる可能性あり)、SQL 定義文を表示。 | あり | あり |
| <code>sysconstraints</code> | テーブルまたはカラムに関連付けられた参照制約と検査制約ごとに 1 ロー。 | なし | なし |
| <code>sysdepends</code> | プロシージャまたはビューが参照するプロシージャ、ビュー、またはテーブルごとに 1 ロー。 | なし | なし |

| テーブル名 | 説明 | データ? | IQによるサポート? |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------|
| sysindexes | クラスタード・インデックスまたはノンクラスタード・インデックスごとに1ロー。インデックスのないテーブルごとに1ロー。さらにtextまたはimageデータを持つテーブルごとに1ロー。 Sybase IQでは、クエリするときに所有者名としてdbo(つまり、dbo.sysindexes)を使用してください。 | あり | あり |
| sysiqobjects | システム・テーブル、ユーザ・テーブル、ビュー、プロシージャ、トリガ、イベント、ジョイン・インデックス、制約、ドメイン(sysdomain)、ドメイン(sysusertype)、カラム、インデックスごとに1ロー。 | あり | あり |
| sysiqvindex | FP IQ以外のインデックスごとに1ロー。 | あり | あり |
| syskeys | ユーザが設定した(Adaptive Server Enterpriseによって維持されない)プライマリ・キー、外部キー、または共通キーごとに1ロー。 | なし | なし |
| syslogs | トランザクション・ログ。 | なし | なし |
| sysobjects | テーブル、ビュー、プロシージャ、ルール、デフォルト、ログ、テンポラリ・オブジェクト(tempdb内でのみ)ごとに1ロー。 | 互換性のあるデータだけを含む。 | あり |
| sysprocedures | ビュー、ルール、デフォルト、プロシージャごとに1ロー、内部定義を表示。 | なし | なし |
| sysprotects | ユーザ・パーミッション情報。 | なし | なし |
| sysreferences | テーブルまたはカラムに宣言された参照整合性制約ごとに1ロー。 | なし | なし |
| sysroles | サーバ全体に適用される役割をローカル・データベース・グループにマップ。 | なし | なし |

| テーブル名 | 説明 | データ? | IQによるサポート? |
|---------------|-------------------------------------|------|------------|
| syssegments | セグメント (ディスクの断片の集合に名前を付けたもの) ごとに1ロー。 | なし | なし |
| systhresholds | データベースに定義されたスレッシュホールドごとに1ロー。 | なし | なし |
| systypes | システムが提供するデータ型またはユーザ定義のデータ型ごとに1ロー。 | あり | あり |
| sysusers | データベースのアクセス権を保持するユーザごとに1ロー。 | あり | あり |

Adaptive Server Enterprise の master データベースに存在するテーブル

表 8-2 : ASE master データベース・テーブル

| テーブル名 | 説明 | データ? | IQによるサポート? |
|---------------|---------------------------------------------------------------------|------|------------|
| syscharsets | 文字セットまたはソート順ごとに1ロー。 | なし | なし |
| sysconfigures | ユーザが設定できる設定パラメータごとに1ロー。 | なし | なし |
| syscurconfigs | サーバが現在使用中の設定パラメータについての情報。 | なし | なし |
| sysdatabases | サーバ上のデータベースごとに1ロー。 | なし | なし |
| sysdevices | テープ・ダンプ・デバイス、ディスク・ダンプ・デバイス、データベース用のディスク、データベース用のディスク・パーティションごとに1ロー。 | なし | なし |
| sysengines | 現在オンラインのサーバごとに1ロー。 | なし | なし |
| syslanguages | サーバが認識している言語 (アメリカ英語を除く) ごとに1ロー。 | なし | なし |
| syslocks | アクティブ・ロックについての情報。 | なし | なし |
| sysloginroles | システム定義された役割を持つサーバ・ログインごとに1ロー。 | なし | なし |
| syslogins | 有効なユーザ・アカウントごとに1ロー。 | あり | あり |
| sysmessages | システム・エラーまたは警告ごとに1ロー。 | なし | なし |
| sysprocesses | サーバ・プロセスについての情報。 | なし | なし |

| テーブル名 | 説明 | データ? | IQによるサポート? |
|-----------------|------------------------------|------|------------|
| sysremotelogins | リモート・ユーザごとに1ロー。 | なし | なし |
| sysserverroles | サーバ全体に適用される役割ごとに1ロー。 | なし | なし |
| syssservers | リモート・サーバごとに1ロー。 | なし | なし |
| sysusages | データベースに割り当てられたディスクの断片ごとに1ロー。 | なし | なし |

Adaptive Server Enterprise の sybsecurity データベースに存在するテーブル

表 8-3 : ASE sybsecurity データベース・テーブル

| テーブル名 | 説明 | データ? | IQによるサポート? |
|-----------------|---------------------|------|------------|
| sysaudits | 監査レコードごとに1ロー。 | なし | なし |
| sysauditoptions | グローバル監査オプションごとに1ロー。 | なし | なし |

この章について

すべての Sybase IQ データベースの構造は、数多くのシステム・テーブルに記述されています。

DUMMY システム・テーブルは、ユーザが直接アクセスすることが許された唯一のシステム・テーブルです。その他のすべてのシステム・テーブルについては、その対応するビューを通じて基本データにアクセスします。「第 8 章 システム・ビュー」を参照してください。

内容

| トピック名 | ページ |
|---------------------------------|-----|
| システム・テーブル・リスト | 597 |
| DUMMY システム・テーブル | 600 |

システム・テーブル・リスト

表 9-1 には、すべての Sybase IQ システム・テーブルの一覧が示されています。

表 9-1 : システム・テーブルのリスト

| システム・テーブル | 内部使用の有無 |
|----------------------------|---------|
| DUMMY | なし |
| ISYSARTICLE | あり |
| ISYSARTICLECOL | あり |
| ISYSATTRIBUTE | あり |
| ISYSATTRIBUTENAME | あり |
| ISYSCAPABILITY | あり |
| ISYSCHECK | あり |
| ISYSCOLPERM | あり |
| ISYSCOLSTAT | あり |
| ISYSCONSTRAINT | あり |
| ISYSDBFILE | あり |
| ISYSDBSPACE | あり |
| ISYSDBSPACEPERM | あり |
| ISYSDEPENDENCY | あり |
| ISYSDOMAIN | あり |
| ISYSEVENT | あり |
| ISYSEXTERNENV | あり |
| ISYSEXTERNENVOBJECT | あり |
| ISYSEXTERNLOGIN | あり |
| ISYSFKEY | あり |
| ISYSGROUP | あり |
| ISYSHISTORY | あり |
| ISYSIDX | あり |
| ISYSIDXCOLUMN | あり |
| ISYSIQBACKUPHISTORY | あり |
| ISYSIQBACKUPHISTORYDETAIL | あり |
| ISYSIQDBFILE | あり |
| ISYSIQDBSPACE | あり |
| ISYSIQIDX | あり |
| ISYSIQINFO | あり |
| ISYSIQJOINIDX | あり |
| ISYSIQJOINIXCOLUMN | あり |
| ISYSIQMPXLOGINPOLICYOPTION | あり |
| ISYSIQMPXSERVER | あり |
| ISYSIQPARTITIONCOLUMN | あり |
| ISYSIQTAB | あり |

| システム・テーブル | 内部使用の有無 |
|-----------------------|---------|
| ISYSIQTABCOL | あり |
| ISYSJAR | あり |
| ISYSJARCOMPONENT | あり |
| ISYSJAVACLASS | あり |
| ISYSLOGINMAP | あり |
| ISYSLOGINPOLICY | あり |
| ISYSLOGINPOLICYOPTION | あり |
| ISYSMVOPTION | あり |
| ISYSMVOPTIONNAME | あり |
| ISYSOBJECT | あり |
| ISYSOPTION | あり |
| ISYSOPTSTAT | あり |
| ISYSPARTITION | あり |
| ISYSPARTITIONKEY | あり |
| ISYSPARTITIONScheme | あり |
| ISYSPHYSIDX | あり |
| ISYSPROCEDURE | あり |
| ISYSPROCPARM | あり |
| ISYSPROCPERM | あり |
| ISYSPROXYTAB | あり |
| ISYSPUBLICATION | あり |
| ISYSREMARK | あり |
| ISYSREMOTEOPTION | あり |
| ISYSREMOTEOPTIONTYPE | あり |
| ISYSREMOTETYPE | あり |
| ISYSREMOTEUSER | あり |
| ISYSSCHEDULE | あり |
| ISYSSEVER | あり |
| ISYSSOURCE | あり |
| ISYSSQLSERVERTYPE | あり |
| ISYSSUBPARTITIONKEY | あり |
| ISYSSUBSCRIPTION | あり |
| ISYSSYNC | あり |
| ISYSSYNCPROFILE | あり |
| ISYSSYNCSRIPT | あり |
| ISYSTAB | あり |
| ISYSTABCOL | あり |

| システム・テーブル | 内部使用の有無 |
|-------------------|---------|
| ISYSTABLEPERM | あり |
| ISYSTEEXTCONFIG | あり |
| ISYSTEEXTIDX | あり |
| ISYSTEEXTIDXTAB | あり |
| ISYSTRIGGER | あり |
| ISYSTYPEMAP | あり |
| ISYSUSER | あり |
| ISYSUSERAUTHORITY | あり |
| ISYSUSERMESSAGE | あり |
| ISYSUSERTYPE | あり |
| ISYSVIEW | あり |
| ISYSWEBSERVICE | あり |

DUMMY システム・テーブル

DUMMY テーブルは、ローを常に 1 つだけ持つテーブルとして提供されています。これはデータベースから情報を抽出するのに役立ちます。次に、データベースから現在のユーザ ID と今日の日付を取り出す例を示します。

```
SELECT USER, today(*) FROM SYS.DUMMY
```

DUMMY テーブルは SQL Anywhere システム・テーブルです。DUMMY system table (http://infocenter.sybase.com/help/topic/com.sybase.help.sqlanywhere.11.0.1/dbreference_en11/sysdummy.html) にある SQL Anywhere のマニュアル『SQL Anywhere Server — SQL リファレンス・マニュアル』を参照してください。

他の Sybase データベースとの 互換性

この付録について

この付録の情報は、他の Sybase データベースから Sybase IQ への移行ガイドとして、および、Adaptive Server Enterprise または SQL Anywhere と互換性のある Sybase IQ アプリケーションの作成ガイドとして提供されています。最初に Transact-SQL の概要を紹介し、次に Sybase IQ に移行する際に知っておく必要のある以下の事項についてこれらのデータベースを比較します。

- アーキテクチャ
- データ型
- データ定義言語
- データ操作言語
- ストアド・プロシージャ言語

Sybase IQ の各新バージョンでは互換性を維持するための機能が提供されています。この付録では、Sybase IQ 15.1 と Adaptive Server Enterprise 15.0.3 (およびそれ以前のバージョン)、および SQL Anywhere 11.0.1 を比較します。

トピック

| トピック名 | ページ |
|--------------------------------------------------------|-----|
| Transact-SQL サポートの概要 | 602 |
| Adaptive Server のアーキテクチャ | 604 |
| データ型 | 608 |
| データ定義言語 | 614 |
| データ操作言語 | 625 |
| Transact-SQL のプロシージャ言語の概要 | 636 |
| ストアド・プロシージャの自動変換 | 639 |
| Transact-SQL プロシージャから結果セットを返す | 640 |
| Transact-SQL プロシージャ内の変数 | 641 |
| Transact-SQL プロシージャでのエラー処理 | 641 |
| SQL Anywhere と Sybase IQ | 644 |
| Adaptive Server Enterprise と Sybase IQ | 647 |

その他の互換性情報

以下の章にも、互換性情報が記載されています。

- 『リファレンス：文とオプション』の「第2章 データベース・オプション」の「[Transact-SQL 互換性オプション](#)」を参照してください。
- 「第3章 SQL データ型」には、各データ型の互換性情報が提供されています。また、「[データ型変換](#)」(90 ページ)も参照してください。
- 『リファレンス：文とオプション』で、各コマンドの互換性情報を参照してください。

SQL Anywhere についての注意

Sybase IQ は、SQL Anywhere を拡張したものです。SQL 構文、関数、オプション、ユーティリティ、プロシージャ、その他の機能のほとんどは、両製品に共通ですが、両製品には相違点もあるので、SQL Anywhere のマニュアルに記載されている機能が Sybase IQ でもサポートされていると想定することはできません。

Sybase IQ のマニュアル・セットでは相違点の多くが説明されていますが、すべてをカバーしていません。Sybase IQ のマニュアルは常に SQL Anywhere のマニュアルに優先します。Sybase IQ のマニュアルで SQL Anywhere のマニュアルを参照するように指示されているトピックを除いて、常に、Sybase IQ の各マニュアルの目次のすぐ後にある「はじめに」の Sybase IQ の「関連マニュアル」に示されている各種マニュアルを参照してください。

Transact-SQL サポートの概要

Sybase IQ は、SQL Anywhere と同様、Sybase Adaptive Server Enterprise がサポートする SQL の **Transact-SQL** ダイアレクトの大部分のサブセットをサポートしています。

Sybase IQ では、Transact-SQL をサポートすることにより、アプリケーションの移植性を実現しています。アプリケーション、ストアド・プロシージャ、バッチ・ファイルの多くは、Adaptive Server Enterprise と Sybase IQ の両方のデータベースで使用できるように作成できます。

目的は、Adaptive Server Enterprise と Sybase IQ の両方で動作するアプリケーションを作成することです。通常は、既存の Adaptive Server Enterprise アプリケーションを SQL Anywhere または Sybase IQ データベース上で実行する場合、多少の変更が必要です。

次に、Sybase IQ での Transact-SQL に対するサポート方法を示します。

- SQL 文のほとんどは、Sybase IQ と Adaptive Server Enterprise の間で互換性があります。
- 一部の文、特にプロシージャ言語を使ってプロシージャやバッチ用に使われた文では、以前のバージョンの Sybase IQ でサポートされた構文に加え、別の Transact-SQL 文もサポートされます。これらの文では、SQL Anywhere も Sybase IQ も 2 種類の SQL ダイアレクトをサポートしています。この付録では、それらのダイアレクトを Transact-SQL と Watcom-SQL と呼びます。
- プロシージャまたはバッチは、Transact-SQL ダイアレクトと Watcom-SQL ダイアレクトのどちらでも実行できます。1 つのバッチまたはプロシージャ内では、全体を通して、いずれかのダイアレクトだけで制御文を記述する必要があります。たとえば、フロー制御文はダイアレクトによって異なります。

類似点と相違点

Sybase IQ は、既存データの処理に使用する、Transact-SQL 言語の要素、関数、文の多くをサポートしています。

さらに、Sybase IQ では、Transact-SQL のストアド・プロシージャ言語 (CREATE PROCEDURE 構文、制御文など) の大部分と、Transact-SQL のデータ定義言語文のほとんどがサポートされています。

それぞれの製品でサポートされるアーキテクチャと構成には設計上の相違点があります。デバイス管理、ユーザ管理、バックアップなどの管理タスクの多くはシステム固有のものです。そのため、Sybase IQ は Transact-SQL のシステム・テーブルをビューとして提供しますが、Sybase IQ にとって意味のないテーブルを示すローを除外します。また、Sybase IQ は、一般的な管理タスクの一部を一連のシステム・プロシージャとして提供します。

Adaptive Server のアーキテクチャ

Adaptive Server Enterprise、SQL Anywhere、Sybase IQ はそれぞれの異なった目的に合わせたアーキテクチャを持つ、相互に補完的な製品です。Sybase IQ は、データ・ウェアハウジングと分析処理専用に設計された、高パフォーマンスの意思決定支援サーバです。SQL Anywhere は、管理作業をほとんど必要とせず、ワークグループや部署単位のサーバとして、さらにはパーソナル・データベースとして適切な製品です。Adaptive Server Enterprise は、トランザクション処理を主眼とした大規模データベース用のエンタープライズ・レベルのサーバとして適切な製品です。

この項では、これら 3 つの製品のアーキテクチャ上の違いを解説します。また、データベース管理の互換性のために Sybase IQ と SQL Anywhere に含まれている、Adaptive Server Enterprise に似たツールについても説明します。

サーバとデータベース

サーバとデータベースの関係は、Adaptive Server Enterprise と Sybase IQ や SQL Anywhere とでは異なります。

Adaptive Server Enterprise では各データベースはサーバ内に存在し、各サーバは複数のデータベースを持つことができます。ユーザはサーバに対するログイン権限を与えられ、サーバに接続できます。ユーザはそのサーバ上のパーミッションを与えられたデータベースに接続できます。master データベースに保持され、システム全体に適用されるシステム・テーブルには、サーバ上のすべてのデータベースに共通な情報が含まれています。

Sybase IQ には Adaptive Server Enterprise の master データベースに対応するものではありません。その代わり、それぞれのデータベースが独立したエンティティであり、独自のシステム・テーブルを持っています。ユーザは、サーバに対してではなく、個々のデータベースに対する接続権限を与えられます。ユーザが接続するのは、個々のデータベースです。master データベース・レベルで維持されるシステム全体に関する一連のシステム・テーブルはありません。各 Sybase IQ データベース・サーバは、動的に個々のデータベースを開始および停止できます。ユーザはデータベースに対する接続を個別に維持できます。1 つのサーバで実行する Sybase IQ データベースは 1 つだけにすることを強くおすすめします。

SQL Anywhere と Sybase IQ は、Transact-SQL のサポートと Open Server のサポートによって、Adaptive Server Enterprise と同様の方法でタスクを実行するツールを提供します。ただし、これらのツールの実装方法には細かい違いがあります。

複数のデータベースを持つサーバ上の特定のデータベースに接続するときに `isql` を使用する方法の詳細については、『システム管理ガイド 第 2 巻』の「第 3 章 データ・サーバとしての Sybase IQ」を参照してください。

領域の割り付けとデバイス管理

3 つの製品は、それぞれの製品の用途を反映して、デバイスの管理とディスク領域の割り付けに、初期設定でもその後の管理でも別々のモデルを使用します。次に例を示します。

- Adaptive Server Enterprise では、最初に `DISK INIT` を使ってデータベース・デバイス内の領域を割り付け、その後 1 つまたは複数のデータベース・デバイス上に個々のデータベースを作成します。`ALTER DATABASE` を使って領域を追加するか、スレッショルドを使って自動的に領域を追加することができます。
- Sybase IQ では、最初に `CREATE DATABASE` 文に生のデバイスをリストアップすることによって、領域を割り付けます。`CREATE DBSPACE` を使用することによって、手動で領域を追加できます。領域を自動的に追加することはできませんが、追加のスペースが実際に必要になる前に `DBA` に警告するイベントを作成することはできます。Sybase IQ では、ファイル・システム領域も利用できます。Sybase IQ は、`DISK INIT`、`DISK MIRROR`、`DISK REFIT`、`DISK REINIT`、`DISK REMIRROR`、`DISK UNMIRROR` などの Transact-SQL `DISK` 文をサポートしません。
- SQL Anywhere は、最初の `CREATE DATABASE` 文が生のデバイスのリストではなく単一のファイル・システム・ファイルを取るという点を除いては、Sybase IQ と同様です。SQL Anywhere ではコマンド・ユーティリティの `dbinit` を使ってデータベースの初期化ができます。これは Sybase IQ ではサポートされません。

ディスク管理の詳細については、『システム管理ガイド 第 1 巻』を参照してください。

システム・テーブル、カタログ・ストア、IQ ストア

IQ データベースは、次の部分から構成されるジョイント・データ・ストアです。

- カタログ・ストアは、システム・テーブルとストアド・プロセスージャを含み、SQL Anywhere と互換性のある一連のテーブル内にあります。
- 永続的な IQ ストアは、一連の Sybase IQ テーブルです。テーブル・データは、インデックスに格納されます。
- テンポラリ・ストアは、データベース・サーバが、ソートやその他の一時的な処理に使用する一連のテンポラリ・テーブルです。

カタログの差異と互換性サポートには次のものが含まれます。

- SQL Anywhere と Sybase IQ は、カタログ (テーブル、カラムなど) に Adaptive Server Enterprise とは異なるスキーマを使用します。
- SQL Anywhere と Sybase IQ には、Adaptive Server Enterprise のシステム・テーブルの対応する部分を模倣する互換ビューが用意されています。ただし、これらのビューを利用するとパフォーマンスに影響します。互換ビューの一覧と個別の説明は、「[第 8 章 システム・ビュー](#)」および「[第 9 章 システム・テーブル](#)」を参照してください。SQL Anywhere 互換ビューの完全なリストは、『SQL Anywhere Server SQL リファレンス・マニュアル』を参照してください。
- Adaptive Server Enterprise では、データベース所有者 (ユーザ ID dbo) がカタログ・オブジェクトの所有者です。
- SQL Anywhere と Sybase IQ では、システムの所有者 (ユーザ ID SYS) がカタログ・オブジェクトの所有者です。

注意 ユーザ ID dbo は、Sybase IQ が提供する、Adaptive Server Enterprise と互換性のあるシステム・ビューの所有者です。ユーザ ID dba は、Sybase IQ のユーザとマルチプレックスの管理に使用する少数の Sybase IQ システム・テーブルの所有者です。

管理者の役割

Adaptive Server Enterprise は、SQL Anywhere や Sybase IQ より管理者の役割が充実しています。Adaptive Server Enterprise には、複数の異なった役割が用意されていますが、複数のログイン・アカウントに同じ役割を与え、1つのアカウントに複数の役割を持たせることができます。

Adaptive Server Enterprise には次の役割の区別があります。

- **システム管理者** 特定のアプリケーションに関連していない一般的な管理タスクを担当し、あらゆるデータベース・オブジェクトにアクセスできます。
- **システム・セキュリティ担当者** セキュリティが問題となる Adaptive Server Enterprise のタスクを担当しますが、データベース・オブジェクトに対する特別なパーミッションは持ちません。
- **データベース所有者** 自分が所有するデータベース内のオブジェクトに対して、フル・パーミッションを持ちます。また、データベースにユーザを追加したり、データベース内でオブジェクトの作成やコマンドの実行を行うパーミッションを他のユーザへ付与したりできます。
- **データ定義文** ユーザがデータベース・オブジェクトを作成できるように、CREATE TABLE や CREATE VIEW などの特定のデータ定義文に対するパーミッションをユーザに与えることができます。
- **オブジェクト所有者** 各データベース・オブジェクトの所有者は、そのオブジェクトにアクセスするパーミッションを他のユーザに付与できます。オブジェクトの所有者は、自動的にそのオブジェクトに対するすべてのパーミッションを持ちます。

SQL Anywhere と Sybase IQ では、データベース全体に適用される次のパーミッションが管理者の役割を持ちます。

- **データベース管理者 (DBA 権限)** Adaptive Server Enterprise のデータベース所有者のように、データベース内のすべてのオブジェクト (SYS が所有するオブジェクトは除く) に対してフル・パーミッションを持ち、データベース内でのオブジェクト作成とコマンド実行のパーミッションを他のユーザに付与できます。デフォルトのデータベース管理者は、ユーザ ID DBA です。

- **RESOURCE パーミッション** ユーザがデータベース内で任意のオブジェクトを作成することを許可します。Adaptive Server Enterprise では個々の CREATE 文に対するパーミッションを与える方法が使用されますが、これはその代わりに使用されます。
- **オブジェクト所有者** Sybase IQ のオブジェクトには、Adaptive Server Enterprise の場合と同じように所有者がいます。オブジェクト所有者は、パーミッションを付与する権限も含めて、そのオブジェクトに関するすべてのパーミッションを自動的に持ちます。

Adaptive Server Enterprise および Sybase IQ の両方に格納されたデータと同じようにアクセスするためには、データベース内に適切なパーミッション (Sybase IQ では RESOURCE、Adaptive Server Enterprise では CREATE 文ごとのパーミッション) を持つユーザ ID を作成し、そのユーザ ID を使ってオブジェクトを作成します。同じユーザ ID を両方の環境で使用すると、オブジェクト名と修飾子が 2 つのデータベースで同一になり、互換性のあるアクセスが可能になります。

データ型

この項では、データ型の互換性情報について説明します。Sybase IQ データ型の詳細については、「[第 3 章 SQL データ型](#)」を参照してください。

注意 この項で説明していないデータ型は、現在 3 つの製品すべてでサポートされています。

Bit データ型

3 つの製品はいずれも BIT データ型をサポートしていますが、以下の違いがあります。

- SQL Anywhere では 0 または 1 のみを許可します。
- Adaptive Server Enterprise および Sybase IQ では、integral データ型を BIT に暗黙的に変換します。ゼロ以外の値は 1 (TRUE) として格納されます。

文字データ型

3 つの製品はいずれも、CHAR および VARCHAR データを許可しますが、その取り扱い方法がそれぞれ異なります。

- SQL Anywhere は、空白がパディングされたデータベースにおいても、すべての文字列を VARCHAR として扱います。
- Adaptive Server Enterprise および Sybase IQ は、CHAR (固定長) と VARCHAR (可変長) のデータを区別します。

Adaptive Server Enterprise では、VARCHAR 値の末尾の空白文字が削除されます。Sybase IQ では、VARCHAR 値の末尾の空白文字が削除されるかどうかは、データの形式と操作によって決まります。詳細については、「第 3 章 SQL データ型」の「文字データ型」を参照してください。

CHAR または VARCHAR への挿入に次のような違いがあります。

- SQL Anywhere では integral データ型を CHAR または VARCHAR (暗黙の変換) に挿入することを許可します。
- Adaptive Server Enterprise および Sybase IQ では、明示的な変換が必要です。

カラムの最大サイズは次のようにして決定されます。

- Adaptive Server Enterprise では、CHAR および VARCHAR は論理ページ・サイズ (2K、4K、8K、16K のいずれか) に依存します。次に例を示します。
 - ページ・サイズが 2K の場合、カラムの最大サイズは単一のローと同じ大きさ (約 1962 バイト) です。
 - ページ・サイズが 4K の場合、カラムの最大サイズは約 4010 バイトです。
- SQL Anywhere は最大 32K-1 の CHAR と VARCHAR をサポートしています。LONG VARCHAR では最大 2GB までサポートしています。
- SQL Anywhere では、名前として LONG VARCHAR とそのシノニム TEXT がサポートされますが、Adaptive Server Enterprise では名前として TEXT だけがサポートされ、LONG VARCHAR はサポートされません。
- Sybase IQ では、32K - 1 バイトまでの CHAR および VARCHAR をサポートしています。

Sybase IQ はさらに、LONG VARCHAR で 512TB (IQ ページ・サイズが 128KB の場合) および 2PB (IQ ページ・サイズが 512KB の場合) までサポートしています。Sybase IQ の LONG VARCHAR データ型については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。

- Adaptive Server Enterprise では、NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR データ型をサポートしています。N はマルチバイトの文字セットを表し、UNI はシングル・バイトの文字セットを表します。
- SQL Anywhere と Sybase IQ は、Unicode を別のデータ型としてではなく、CHAR と VARCHAR データ型の中でサポートしています。
- Sybase IQ と Adaptive Server Enterprise の互換性を保つため、文字データ型には必ず長さを指定してください。

バイナリ・データ型

次の表は、データ型のサポートの概要を示したものです。

表 A-1 : バイナリ・データ型の許容されるサイズ

| データ型 | Adaptive Server Enterprise | SQL Anywhere | Sybase IQ |
|--------------|----------------------------|--------------|----------------------------------------------------|
| BINARY | <ページ・サイズ | 32KB - 1 | 255 |
| VARBINARY | <ページ・サイズ | 32KB - 1 | 32KB - 1 |
| LONG BINARY* | サポートなし | 2GB - 1 | 512TB (IQ ページ・サイズ 128KB) 2PB (IQ ページ・サイズ 512KB) |
| IMAGE | 2GB | 2GB - 1 | LONG BINARY* を使用してください |

*Sybase IQ の LONG BINARY データ型の詳細については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。この機能には、別のライセンスが必要です。

Adaptive Server Enterprise と SQL Anywhere では、バイナリ・データの表示方法が異なります。

- Sybase IQ は Adaptive Server Enterprise と SQL Anywhere の両方の表示形式をサポートしています。
- '123' が BINARY フィールドに入力されると、SQL Anywhere の表示形式ではバイト単位の '123' になります。Adaptive Server Enterprise の表示形式ではニブル単位の '0x616263' になります。

date、time、datetime、timestamp データ型

3 つの製品はいずれも日付と時刻のデータを何らかの形式でサポートしていますが、それぞれに違いがあります。

- SQL Anywhere と Sybase IQ は、4 バイトの日付と時刻のデータ型をサポートしています。
- Adaptive Server Enterprise は、バイナリ (8) で実装されているユーザ定義のデータ型 (ドメイン) として、8 バイトの **datetime** データ型と **timestamp** データ型をサポートしています。
- SQL Anywhere と Sybase IQ は、8 バイトの **timestamp** 型と、**timestamp** として実装されている 8 バイトの **datetime** ドメインをサポートしています。Anywhere/Sybase IQ の **datetime** データ型のミリ秒単位の精度は、Adaptive Server Enterprise の精度とは異なります。

それぞれの製品におけるデフォルトの日付表示形式には次のような違いがあります。

- Adaptive Server Enterprise のデフォルトでは、日付を MMM-DD-YYYY 形式で表示しますが、オプションを設定してこれを変更することができます。
- SQL Anywhere/Adaptive Server Anywhere と Sybase IQ のデフォルトは、ISO の YYYY-MM-DD という形式ですが、オプションを設定してこれを変更することができます。

時刻の変換は次のとおりです。

- Adaptive Server Enterprise では、文字列に格納された時刻を内部時刻に変換する方法が、秒の少数部分がコロンとピリオドのどちらで区切られているかによって変わります。
- SQL Anywhere と Sybase IQ では、区切り文字に関係なく同じ方法で時刻が変換されます。

DATETIME カラムに時刻を挿入する場合

- Adaptive Server Enterprise と Sybase IQ では、1900 年 1 月 1 日がデフォルトで設定されます。
- SQL Anywhere では、現在の日付がデフォルトです。

Adaptive Server Enterprise データベースから取得した TIME 値と DATETIME 値は、INSERTAcLOCATION を使用して DATETIME カラムがある Sybase IQ テーブルに挿入すると、変更されます。

INSERTAcLOCATION 文は、DATETIME の精度が 300 分の 1 秒である Open Client を使用します。

たとえば、Adaptive Server Enterprise データベースのテーブル・カラムに、次の値が格納されているとします。

```
2004-11-08 10:37:22.823
```

INSERT..LOCATION を使用してこの値を取得し、Sybase IQ テーブルに格納すると、値は次のようになります。

```
2004-11-08 10:37:22.823333
```

ASE の日時値と時刻値との互換性

INSERT..LOCATION を使用して Adaptive Server Enterprise データベースから取得した DATETIME 値または TIME 値は、Open Client の日時の精度が原因で、元の値とは異なる値になることがあります。

たとえば、Adaptive Server Enterprise データベースでの DATETIME 値が '2004-11-08 10:37:22.823' のとき、INSERT..LOCATION を使用して取得した値は '2004-11-08 10:37:22.823333' になります。

数値データ型

Adaptive Server Enterprise、SQL Anywhere、Sybase IQ のそれぞれで、デフォルトの精度と位取りが異なります。

- Adaptive Server Enterprise のデフォルトは、精度 18、位取り 0 です。
- SQL Anywhere のデフォルトは精度 30 位取り 6 です。
- Sybase IQ のデフォルトは、精度 126、位取り 38 です。これらのデフォルトは TDS や一部のクライアント・ツールには大きすぎるため、Sybase IQ の真数値型では精度と位取りを必ず指定する必要があります。

概数値データ型

Adaptive Server Enterprise では SQL Anywhere や Sybase IQ とは、FLOAT(precision) データ型の解釈が異なります。つまり、4 バイト・データ型と 8 バイト・データ型の作成される基準が異なります。

REAL 値は 4 バイトで、DOUBLE 値は 8 バイトです。精度が指定されていない場合、FLOAT_AS_DOUBLE オプションをオンにすると、IQ FLOAT キーワードに Adaptive Server Enterprise の FLOAT キーワードと同様の動作をさせることができます。Adaptive Server Enterprise はその FLOAT 値を DOUBLE として処理するので、このオプションを有効にすると、Sybase IQ は Adaptive Server Enterprise と同じ方法で FLOAT 値を処理します。

FLOAT_AS_DOUBLE オプションをオフにすると、FLOAT キーワードは Sybase IQ や SQL Anywhere の 4 バイト REAL 値と同じになります。

Adaptive Server Enterprise の場合、FLOAT(precision) で指定された精度は 10 進数の精度を意味します。SQL Anywhere と Sybase IQ の場合、精度は、浮動小数点数の小数部で使用されるビット数を指定した整数式になります。

Sybase IQ FLOAT データ型の詳細については、「[第 3 章 SQL データ型](#)」の「[数値データ型](#)」を参照してください。

text データ型

TEXT データのサポートは、現在次のように実装されています。

- Adaptive Server Enterprise と SQL Anywhere は LONG VARBINARY と TEXT で最大 2GB までサポートしています。
- Sybase IQ は VARCHAR で最大 32KB - 1 までサポートしています。Sybase IQ はさらに LONG VARCHAR で 512TB (IQ ページ・サイズが 128KB の場合) と 2PB (IQ ページ・サイズが 512KB の場合) までサポートしています。Sybase IQ の LONG VARCHAR データ型については、『[Sybase IQ によるラージ・オブジェクト管理](#)』を参照してください。

image データ型

IMAGE データのサポートは、現在次のように実装されています。

- Adaptive Server Enterprise と SQL Anywhere は最大 2GB の IMAGE をサポートしています。
- Sybase IQ は、LONG BINARY で 512TB (IQ ページ・サイズが 128KB の場合) または 2PB (IQ ページ・サイズが 512KB の場合) までサポートしています。Sybase IQ の LONG BINARY データ型については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。

Java データ型

Adaptive Server Enterprise では、データベースで Java データ型を使用できますが、SQL Anywhere と Sybase IQ ではできません。

データ定義言語

この項では、データベース・オブジェクトを作成する場合の互換性について説明します。関連情報については、「[システム・テーブル、カタログ・ストア、IQ ストア](#)」(606 ページ) および「[領域の割り付けとデバイス管理](#)」(605 ページ) を参照してください。

Transact-SQL と互換性のあるデータベースの作成

ここでは、データベースの作成時や再構築時に選択する必要のあるオプションについて説明します。

まず、Transact-SQL と互換性のあるデータベースを作成するときに必要な基本手順を示します。その後は、設定する必要があるオプションについて説明します。

- ❖ **Sybase Central から Transact-SQL と互換性のあるデータベースを作成するには**
 - 1 [データベースの作成] ウィザードに [デフォルト・データベース属] というページがあります。

- 2 Adaptive Server Enterprise をエミュレートするには、[Adaptive Server Enterprise のエミュレート] を選択します。これにより、比較での大文字小文字の区別とパスワードでの大文字小文字の区別とが自動的にオンになります。

❖ **CREATE DATABASE 文を使用して Transact-SQL と互換性のあるデータベースを作成するには**

- 次の文を、Interactive SQL などで入力します。

```
CREATE DATABASE 'db-name.db'
CASE RESPECT BLANK PADDING ON
```

大文字と小文字の区別

データベースにおける大文字と小文字の区別は、次のことに関連があります。

- **データ** データの大文字と小文字を区別するかしないかは、インデックス、クエリの結果などに反映されます。
- **識別子** 識別子には、テーブル名、カラム名、ユーザ ID などがあります。
- **パスワード** パスワードでの大文字と小文字の区別は、他の識別子とは違う扱いを受けます。

データの大文字と小文字の区別

Sybase IQ のデータ比較を行う場合の大文字と小文字の区別については、データベース作成時に決定します。Sybase IQ データベースは、デフォルトでは、データは常に入力されたとおりの大文字と小文字で保持されていますが、比較において大文字と小文字を区別しません。

Adaptive Server Enterprise での大文字と小文字の区別の有無は、Adaptive Server Enterprise システムにインストールされているソート順によって決まります。シングル・バイト文字セットを使用している場合は、Adaptive Server Enterprise のソート順を再設定することによって、大文字と小文字の区別の取り扱い方を変更できます。

識別子の大文字と小文字の区別

Sybase IQ は、大文字と小文字を区別する識別子をサポートしていません。Adaptive Server Enterprise では、識別子の大文字と小文字の区別はデータの大文字と小文字の区別に従います。

Adaptive Server Enterprise では、ユーザ定義データ型の名前については、大文字と小文字を区別します。Sybase IQ では、大文字と小文字は区別されません。

ユーザ ID とパスワード

Sybase IQ と SQL Anywhere では、新しく作成されたデータベース内のパスワードには大文字小文字の区別があります。これはデータベース自体の大文字小文字の区別の設定に依存しません。デフォルトのユーザ ID は DBA です。このユーザのパスワードは小文字の `sql` です。

既存のデータベースを再構築するとき、Sybase IQ と SQL Anywhere はパスワードの大文字小文字の区別を次のようにして決定します。

- 元々大文字小文字の区別がないデータベースで入力されたデータベースの場合は、パスワードの大文字小文字の区別は元通り無視されます。
- 元々大文字小文字の区別があるデータベースに入力されたパスワードでは、大文字のみのパスワードと大文字小文字が混在しているパスワードの大文字小文字の区別が維持されます。すべて小文字で入力されていたパスワードには、大文字小文字の区別が適用されません。
- 既存のパスワードでも新しいパスワードでも、変更されると大文字小文字の区別が有効になります。

Adaptive Server Enterprise では、ユーザ ID とパスワードの大文字と小文字の区別は、サーバの大文字と小文字の区別に従います。

オブジェクト名の互換性

各データベース・オブジェクトは、特定の「**ネーム・スペース**」内でユニークな名前を持っている必要があります。そのネーム・スペースの外では同じ名前があってもかまいません。一部のデータベース・オブジェクトは Adaptive Server Enterprise 内に SQL Anywhere や Sybase IQ とは異なるネーム・スペースを持っています。

テーブル名のユニーク性

テーブル名は、データベース内でユニークである必要があります。

- Sybase IQ と SQL Anywhere では、テーブル名は個々の所有者の個々のデータベース内でユニークである必要があります。たとえば、`user1` と `user2` の両者が、`employee` という名前のテーブルを作成できます。ユニーク性は、`user1.employee` および `user2.employee` という完全修飾名で実現されます。
- Adaptive Server Enterprise では、テーブル名はデータベース内でも所有者にとってもユニークである必要があります。

**インデックス名の
ユニーク性**

インデックス名は、テーブル内でユニークである必要があります。3つの製品ではいずれも、インデックスが作成されたテーブルの所有者が、そのインデックスを所有します。個々のテーブル内ではインデックス名はユニークである必要がありますが、別々のテーブルでは、所有者が同じであっても同一のインデックス名を持つことができます。たとえば、3つの製品ではいずれも、テーブル t1 と t2 は、所有者が同じであっても別であっても、同じインデックス名を持つことができます。

**インデックスと外部キーの
名前の変更**

Sybase IQ では、ALTER INDEX 文を使って、明示的に作成されたインデックス、インデックスの外部キー・ロール名、外部キーの名前を変更できます。Adaptive Server Anywhere では、ALTER INDEX 文を使って、インデックス、外部キー・ロール名、外部キーを変更できます。Adaptive Server Enterprise では、これらのオブジェクトの名前を変更することはできません。

CREATE TABLE 文

互換性のあるテーブルを作成する際は、次の事柄に注意してください。

カラム内の NULL

互換性のある NULL の取り扱い

- SQL Anywhere と Sybase IQ では、カラムの定義で NOT NULL が宣言されていない限り、カラムが null になることを認めます。この動作は、データベース・オプション ALLOW_NULLS_BY_DEFAULT で、Transact-SQL 互換性設定を OFF にすることで変更できます。
- SQL Anywhere では、BIT カラムのみが NULL になれないと想定します。
- Adaptive Server Enterprise では、NULL が宣言されない限り、カラムは NULL になれないと想定されます。

検査制約

Sybase IQ では、ベース・テーブル、グローバル・テンポラリ・テーブル、ローカル・テンポラリ・テーブル、ユーザ定義のデータ型に検査制約を適用します。ユーザは、検査整合性制約の違反のログを記録して、LOAD 文がロールバックするまでに発生を許可する違反の数を指定できます。

Sybase IQ では、ユーザ定義関数、プロキシ・テーブル、非 Sybase IQ テーブルなどで構成された検査制約のように、評価できない検査制約の作成を許可しません。評価できない制約は、検査制約が定義されているテーブルが LOAD、INSERT、または UPDATE 文で最初に使用されたときに検出されます。Sybase IQ では、次のものを含む検査制約を許可しません。

- サブクエリ
- データ値のターゲットとして、ホスト言語パラメータ、SQL パラメータ、またはカラムを指定している式
- set 関数
- 非決定的関数、またはデータを変更する関数の呼び出し

Adaptive Server Enterprise と SQL Anywhere は CHECK 制約を適用します。SQL Anywhere はサブクエリを検査制約で許可します。

Sybase IQ がサポートするユーザ定義データ型では、データ型の定義内に制約をカプセル化できます。

参照整合性制約

Sybase IQ では、『システム管理ガイド 第1巻』の「[第9章 データ整合性の確保](#)」で説明されているように参照整合性が強制的に実現されます。

整合性の強制実現は次のようなアクションによってサポートされています。

- SQL Anywhere は、SET NULL、CASCADE、DEFAULT、RESTRICT のすべての ANSI アクションをサポートしています。
- Adaptive Server Enterprise では、そのうち SET NULL と DEFAULT の2つのアクションをサポートしています。

注意 Adaptive Server Enterprise で CASCADE を行うには、参照整合性ではなく、トリガを使用します。

- Sybase IQ では RESTRICT アクションのみがサポートされています。
- Sybase IQ では NOT NULL FOREIGN KEY はサポートされていません。
- Sybase IQ には、カラムが候補キーと外部キーの両方に同時にすることはできないという制約があります。

カラムのデフォルト値

デフォルト値のサポートは、次のようにそれぞれ異なります。

- Adaptive Server Enterprise と SQL Anywhere では、カラムにデフォルト値を指定できます。
- DEFAULT UTC TIMESTAMP をサポートするのは SQL Anywhere だけです。
- Sybase IQ では、カラムにデフォルト値を指定できますが、特殊な値である DEFAULT UTC TIMESTAMP と DEFAULT CURRENT UTC TIMESTAMP は指定できません。また、Sybase IQ では、DEFAULT_TIMESTAMP_INCREMENT データベース・オプションの設定が無視されます。

identity カラム

identity カラムのサポートには次のような違いがあります。

- Sybase IQ は、IDENTITY または DEFAULT AUTOINCREMENT をデフォルト値としてサポートしています。Sybase IQ では、任意の精度、位取り 0 の数値型の identity カラムがサポートされ、カラムは NULL でもかまいません。Sybase IQ の identity カラムは正の数である必要があり、データ型の範囲によって制限されます。Sybase IQ では、1 つのテーブルにつき 1 つの identity カラムがサポートされ、明示的な挿入や更新のためにはデータベース・オプション IDENTITY_INSERT にテーブル名を指定する必要があります。IDENTITY カラムを持つテーブルを削除するときは、IDENTITY_INSERT にそのテーブルを指定しておくことはできません。identity カラムの追加時には、テーブルにデータが含まれていてもかまいません。SELECT INTO を使って作成されたテーブルには、Identity/Autoincrement カラムがありません。Sybase IQ のビューには、IDENTITY/DEFAULT AUTOINCREMENT カラムを含めることはできません。
- SQL Anywhere は AUTOINCREMENT デフォルト値をサポートしています。SQL Anywhere は、可能なあらゆる位取りと精度を持つ任意の数値型の identity カラムをサポートしています。identity カラムの値は、正、負、ゼロのいずれでもかまいませんが、データ型の範囲によって制限されます。SQL Anywhere は、テーブル内で任意の数の identity カラムをサポートし、明示的な挿入、削除、更新に identity_insert を必要としません。identity カラムの追加時には、テーブルは空になっている必要があります。SQL Anywhere では、identity カラムを非 identity カラムに、またその逆に変更できます。SQL Anywhere ビューでは、AUTOINCREMENT カラムを追加または削除できます。

- Adaptive Server Enterprise では、1 つのテーブルにつき 1 つの `identity` カラムをサポートしています。ASE の `identity` カラムは、位取り 0、最大精度 38 の数値データ型に限定されます。また、値は正の数である必要があり、データ型の範囲によって制限され、`null` にはできません。Adaptive Server Enterprise では、明示的な挿入と削除には `identity_insert` が必要ですが、`identity` カラムの明示的な更新には必要ありません。`identity` カラムの追加時に、テーブルにデータが含まれていてもかまいません。ASE ユーザは、`identity` カラムに選択される次の値を明示的に設定することはできません。ASE ビューには、`IDENTITY/AUTOINCREMENT` カラムを含めることはできません。`SELECT INTO` を使用するとき、ASE では、条件によっては、元のテーブルに `Identity/Autoincrement` カラムが含まれている場合、結果テーブルにもこれらを含めることができます。

計算カラム

計算カラムのサポートには次の違いがあります。

- SQL Anywhere は、インデックスできる計算カラムをサポートしています。
- Adaptive Server Enterprise と Sybase IQ ではサポートしていません。

テンポラリ・テーブル

テンポラリ・テーブルを作成するには、所有者を指定せずに、`CREATE TABLE` 文の中でテーブル名の前にポンド記号 (#) を指定します。このようなテンポラリ・テーブルは、Sybase IQ の宣言されたテンポラリ・テーブルであり、現在の接続内でのみ使用できます。Sybase IQ の宣言されたテンポラリ・テーブルの詳細については、『リファレンス：文とオプション』の「[DECLARE LOCAL TEMPORARY TABLE 文](#)」を参照してください。

テーブルの作成方法の詳細については、『リファレンス：文とオプション』の「[CREATE TABLE 文](#)」を参照してください。

テーブルの配置

テーブルの物理的な配置方法は、Adaptive Server Enterprise と Sybase IQ では異なります。Sybase IQ は `ON segment-name` 句をサポートしますが、`segment-name` は Sybase IQ の DB 領域を意味します。

CREATE DEFAULT、CREATE RULE、CREATE DOMAIN 文

Sybase IQ には、ルールを組み込む別の方法が用意されています。

- Adaptive Server Enterprise は、名前付きデフォルトを作成するための、`Create Default` および `Create Rule` 文をサポートしています。

- SQL Anywhere と Sybase IQ は同じ目的のために CREATE DOMAIN 文をサポートしています。

CREATE TRIGGER 文

トリガのサポートには次の違いがあります。

- SQL Anywhere は、ロー・レベルと文レベル両方のトリガをサポートしています。
- Adaptive Server Enterprise は、文レベルのトリガのみをサポートしています。
- Sybase IQ ではトリガをサポートしていません。

注意 トリガは、実質的にはストアド・プロシージャであり、INSERT、UPDATE、DELETE の直前、もしくは直後に同じトランザクションの一部として、自動的に実行され、依存する変更 (従業員が別の部署に異動になったときに上司の名前を自動的に更新するなど) を行うために利用できます。また、トリガは、どの修正がいつデータベースにどの変更をいつ加えたのかを識別するための監査証跡を記録するために利用することもできます。

CREATE INDEX 文

CREATE INDEX の構文は、3 つの製品間で多少異なります。

- Adaptive Server Enterprise と SQL Anywhere では、次の構文で、クラスタード・インデックスまたはノンクラスタード・インデックスをサポートしています。

```
CREATE [UNIQUE] [CLUSTERED] INDEX name
ON table (column,...)
ON dbspace
```

Adaptive Server Enterprise でも NONCLUSTERED キーワードを使用できます。どちらの製品でもデフォルトは NONCLUSTERED です。

- Adaptive Server Enterprise の CREATE INDEX 文は、SQL Anywhere でも機能しますが、SQL Anywhere は FILLFACTOR、IGNORE_DUP_KEY、ASORTED_DATA、IGNORE_DUP_ROW、ALLOW_DUP_ROW の各キーワードを黙認して無視します。
- SQL Anywhere の CREATE INDEX 構文は、VIRTUAL キーワードをその Index Consultant が使用するのをサポートしますが、クエリの実行ではサポートしません。
- Sybase IQ は特殊なインデックスの種類として LF、HG、HNG、DATE、TIME、DTTM、WD の 7 つをサポートしています。Sybase IQ では、同じデータ型、精度、位取りを持つ 2 つのカラム間の関係に対する CMP インデックスもサポートしています。Sybase IQ では、CREATE INDEX 文でインデックスの種類が指定されない場合、デフォルトで HG インデックスを作成します。

```
CREATE [UNIQUE] [type] INDEX name  
ON table (column,...)
```

注意 Sybase IQ では、クエリで定期的かつ頻繁にジョインされるカラムの組み合わせにジョイン済みのインデックスを作成できる CREATE JOIN INDEX もサポートしています。

詳細については、『システム管理ガイド 第 1 巻』の「[第 6 章 Sybase IQ インデックスの使用](#)」を参照してください。

ユーザ、グループ、パーミッション

Adaptive Server Enterprise のユーザとグループのモデルは、SQL Anywhere や Sybase IQ のものとは多少異なります。

ユーザの接続方法

Adaptive Server Enterprise では、サーバに接続する各ユーザは、サーバに対するログイン ID とパスワード、およびそのサーバ上でアクセスする各データベースに対するユーザ ID が必要です。

SQL Anywhere と Sybase IQ のユーザはサーバへのログイン ID を必要としません。SQL Anywhere と Sybase IQ のユーザは個々のデータベースに対するユーザ ID とパスワードを受け取ります。

ユーザ・グループ

3 つの製品のいずれもユーザ・グループをサポートしているので、複数のユーザに一度にパーミッションを付与できます。ただし、グループの詳細が異なります。

- Adaptive Server Enterprise では、ユーザは 1 つのグループのメンバにしかありません。

データベース・オブジェクトのパーミッション

- SQL Anywhere と Sybase IQ では、ユーザは複数のグループのメンバになることができ、グループを階層構造にすることができます。

3 つの製品すべてに、デフォルトのパーミッションを定義するための **public** グループがあります。すべてのユーザは、自動的に **public** グループのメンバになります。

個々のデータベース・オブジェクトにパーミッションを付与する **GRANT** 文および **REVOKE** 文は、3 つの製品でかなり共通しています。

- いずれの製品でも、**SELECT**、**INSERT**、**DELETE**、**UPDATE**、**REFERENCES** のパーミッションをデータベース・テーブルとビューに設定でき、**UPDATE** パーミッションをデータベース・テーブルの特定のカラムに設定できます。

たとえば、次の文は 3 つの製品のいずれにおいても有効です。

```
GRANT INSERT, DELETE
ON TITLES
TO MARY, SALES
```

この文は、**TITLES** テーブルで **INSERT** 文と **DELETE** 文を使用するパーミッションをユーザ **MARY** と **SALES** というグループに与えます。

- すべての製品で、ストアド・プロシージャに **EXECUTE** パーミッションを与えることができます。
- Adaptive Server Enterprise では、**GRANT** と **REVOKE** を次のアイテムにも使用できます。
 - オブジェクト：テーブル内のカラム、ビュー内のカラム、ストアド・プロシージャ
 - ユーザが可能な操作：**CREATE DATABASE**、**CREATE DEFAULT**、**CREATE PROCEDURE**、**CREATE RULE**、**CREATE TABLE**、**CREATE VIEW**
- SQL Anywhere と Sybase IQ では、ユーザがデータベース・オブジェクトを作成するには **RESOURCE** 権限が必要です。(Adaptive Server Enterprise では、このパーミッションにほぼ相当するのは、データベース所有者が使用する **GRANT ALL** です。)

- 3つの製品はいずれも WITH GRANT OPTION 句をサポートしています。この句は、パーミッションを付与されたユーザがそのパーミッションを他のユーザに付与できるようにします。ただし、Sybase IQ では GRANT EXECUTE 文で WITH GRANT OPTION は使用できません。

データベース全体に適用されるパーミッション

Adaptive Server Enterprise では、データベース全体のユーザ・パーミッションには別のモデルを使用します。

- SQL Anywhere と Sybase IQ では DBA パーミッションを使用して、ユーザに個々のデータベース内での完全な権限を与えます。
- Adaptive Server Enterprise のシステム管理者は、サーバ上のすべてのデータベースに対して完全なパーミッションを与えられます。しかし、Sybase IQ データベース上での DBA 権限は、Adaptive Server Enterprise のデータベース所有者のパーミッションとは異なります。Adaptive Server Enterprise のデータベース所有者は、他のユーザが所有するオブジェクトに対するパーミッションを得るには、Adaptive Server Enterprise の SETUSER 文を使用する必要があります。

ユーザの追加

Adaptive Server Enterprise でユーザを追加するには、sp_addlogin と sp_add_user の2つのステップを経る必要があります。

SQL Anywhere と Sybase IQ では1つのステップでユーザを追加できます。

Sybase IQ ログイン管理のストアド・プロシージャでは、ユーザを追加または削除する必要はありませんが、DBA は Sybase IQ ユーザ・アカウントを追加または削除できます。Sybase IQ ユーザ管理が有効になっていれば、DBA は Sybase IQ のユーザ・アカウントのユーザ接続やパスワードの期限を管理できます。

詳細については、『システム管理ガイド 第1巻』の「[第8章 ユーザ ID とパーミッションの管理](#)」と『システム管理ガイド 第2巻』の「[第3章 データ・サーバとしての Sybase IQ](#)」を参照してください。

SQL Anywhere と Sybase IQ では、Adaptive Server Enterprise のシステム・プロシージャを使ってユーザやグループを管理することができますが、これらのプロシージャの厳密な構文や機能は同じでない場合があります。詳細については、「[Adaptive Server Enterprise システム・プロシージャ](#)」(522 ページ)を含めて「[第7章 システム・プロシージャ](#)」を参照してください。

ロード形式

3つの製品でサポートされているロード形式は次のとおりです。

- Sybase IQ は ASCII、BINARY、BCP ロード形式を扱います。
- SQL Anywhere では、ASCII と BINARY の他に、dBase、Excel、FoxPro、Lotus ファイル形式をインポートできます。
- Adaptive Server Enterprise は、BCP を通じて ASCII と BINARY ロード形式を扱います。

注意 Sybase IQ と SQL Anywhere の LOAD 文の構文は BCP に基づいており、まったく同じ機能を果たします。

Transact-SQL との互換性を維持するためのオプション設定

Sybase IQ データベース・オプションは、SET OPTION 文を使用して設定します。互換性のある動作に必要なオプション設定の詳細については、『リファレンス：文とオプション』、「データベース・オプション」の「[Transact-SQL 互換性オプション](#)」を参照してください。

データ操作言語

この項では、移植可能なクエリ作成の全般的なガイドラインを示した後、特定のクエリ要件を説明します。

移植可能な SQL を書くための全般的なガイドライン

複数のデータベース管理システムで使用するために SQL を書く場合は、SQL 文をできるだけ明示的にします。指定した SQL 文を複数のサーバがサポートしている場合でも、デフォルトの動作が各システムで同じであると仮定できない場合もあります。次に、互換性のある SQL を書くときの全般的なガイドラインを示します。

- デフォルトの動作を使用しないで、使用可能なオプションをすべて略さずに書く。

- カッコを使用して文中での実行順序を明示的に指定し、演算子の優先順位のデフォルトが同じであると想定しない。
- Adaptive Server Enterprise に移植できるように、変数名には @ をプレフィクスとして付ける Transact-SQL の規則に従う。
- BEGIN 文のすぐ後で、プロシージャとバッチ内の変数とカーソルを宣言する。Sybase IQ ではこれが必須ですが、Adaptive Server Enterprise ではプロシージャまたはバッチ内のどこでも宣言できます。
- Adaptive Server Enterprise または Sybase IQ の予約語をデータベース内の識別子として使用しない。

互換性のあるクエリの書き方

Sybase IQ と Adaptive Server Enterprise の両方のデータベースで実行できるクエリを書くには、次の 2 つの基準に従う必要があります。

- クエリ中のデータ型、式、検索条件が互換性を持つこと。
- SELECT 文の構文自体が互換性を持つこと。

Sybase IQ がサポートする、Transact-SQL の SELECT 文のサブセットを次に示します。

構文

```
SELECT [ ALL | DISTINCT ] select-list
...[ INTO #temporary-table-name ]
...[ FROM table-spec,
...  table-spec, ... ]
...[ WHERE search-condition ]
...[ GROUP BY column-name, ... ]
...[ HAVING search-condition ]
...[ ORDER BY expression [ ASC | DESC ], ... ] |
  [ ORDER BY integer [ ASC | DESC ], ... ] |
```

パラメータ

```
select-list:
{ table-name. * }...
{ * }...
{ expression }...
{ alias-name = expression }...
{ expression as identifier }...
{ expression as T_string }...

table-spec:
[ owner. ]table-name
```

```
... [[ AS ] correlation-name ]
...
```

```
alias-name:
  identifier | 'string' | "string"
```

SELECT 文の詳細な説明は、『リファレンス：文とオプション』の「[SELECT 文](#)」を参照してください。

次の項では、互換性のあるクエリを作成するために注意する必要がある事項を説明します。

サブクエリ

Sybase IQ でのサブクエリのサポートは、現在、Adaptive Server Enterprise および SQL Anywhere でのサポートとは多少異なります。Adaptive Server Enterprise と SQL Anywhere はサブクエリを ON 句の中でサポートしていますが、Sybase IQ は現在サポートしていません。

サブクエリ内の UNION は次のようにサポートされます。

- SQL Anywhere は UNION を相関サブクエリと非相関サブクエリの両方でサポートしています。
- Sybase IQ では、非相関クエリでのみ UNION をサポートしています。
- Adaptive Server Enterprise では、どのサブクエリにおいても UNION をサポートしていません。

SQL Anywhere は、文法にスカラ値が登場するその他の多くの場面でサブクエリをサポートしています。Adaptive Server Enterprise と Sybase IQ では、サブクエリを指定できる場所について ANSI 標準に従っています。

GROUP BY句

GROUP BY ALL のサポートは次のとおりです。

- Adaptive Server Enterprise は、WHERE 句および HAVING 句によって除外されたグループも含む、すべてのグループを返す GROUP BY ALL をサポートしています。この場合、集約はすべて NULL 値になります。

- SQL Anywhere は GROUP BY ALL Transact-SQL 拡張機能をサポートしません。

GROUP BY 句内の ROLLUP と CUBE のサポートは次のとおりです。

- Sybase IQ と SQL Anywhere は ROLLUP と CUBE を GROUP BY 句の中でサポートしています。
- Adaptive Server Enterprise は、現在 ROLLUP と CUBE をサポートしていません。

Adaptive Server Enterprise は、SELECT 句内のグループ化されていないカラムの表示をサポートしています。これは、セマンティックによる拡張グループと呼ばれ、一連の値を返します。Sybase IQ はセマンティックによる拡張グループをサポートしていますが、SQL Anywhere はサポートしていません。値のリストを返す List() 集約関数をサポートしているのは、SQL Anywhere のみです。

GROUP BY を OLAP 関数で使用方法については、『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」を参照してください。

COMPUTE 句

COMPUTE 句のサポートは次のとおりです。

- Adaptive Server Enterprise は、Transact-SQL の COMPUTE 句をサポートしています。
- Transact-SQL の COMPUTE 句は、ANSI 標準になく、また、この機能は大部分のサード・パーティのフロントエンド・ツールによって実現されているため、SQL Anywhere と Sybase IQ ではサポートしていません。

WHERE 句

WHERE 句での Contains() 述部のサポートと、Like() 述部での末尾の空白の取り扱いは製品ごとに異なります。

- Sybase IQ は、文字データ内のワード検索で Contains() 述部をサポートしています (MS SQL Server や Verity での Contains と同様)。Sybase IQ は、可能な場合 WORD インデックスを使って検索を最適化します。

- SQL Anywhere と Adaptive Server Enterprise は Contains() をサポートしていません。

ジョイン

この項では、Transact-SQL の外部ジョインと ANSI ジョインのサポートについて説明します。

Transact-SQL 外部ジョイン

外部ジョインでサポートされている構文は次のとおりです。

- Adaptive Server Enterprise は、*= および=* の Transact-SQL 外部ジョイン構文をサポートしています。
- SQL Anywhere と Sybase IQ は、Transact-SQL の外部ジョインをサポートしていますが、曖昧になる可能性のある一部の複雑な Transact-SQL 外部ジョインは拒否されます。
- Sybase IQ は、Transact-SQL の連鎖 (ネストした) 外部ジョインをサポートしていません。このような複数の外部ジョインには、ANSI 構文を使用してください。

注意 Transact-SQL の外部ジョイン構文は SQL Anywhere と Sybase IQ では非推奨です。

ANSI ジョイン

ANSI ジョインの構文のサポートは次のとおりです。

- Sybase IQ は、現在、ON 句内のサブクエリをサポートしていません。
- Adaptive Server Enterprise と SQL Anywhere は ON 句内のサブクエリをサポートしています。

完全外部ジョインのサポートは次のとおりです。

- SQL Anywhere と Sybase IQ は FULL OUTER JOIN をサポートしています。
- Adaptive Server Enterprise は FULL OUTER JOIN をサポートしていません。

外部ジョインについての詳細情報

ANSI 構文の代替を含む、Transact-SQL 外部ジョインの詳細情報は、[MySybase \(http://www.sybase.com/support/\)](http://www.sybase.com/support/) で閲覧可能なホワイト・ペーパー『Semantics and Compatibility of Transact-SQL Outer Joins』を参照してください。これは SQL Anywhere 用に書かれたものですが、このホワイト・ペーパー内の情報は Sybase IQ にも適用できます。

Null 比較

Adaptive Server Enterprise には、NULL 値を比較する述部を許可する Transact-SQL 拡張機能が用意されています。たとえば、{col} = Null は、{col} Is Null と同義です。

SQL Anywhere と Sybase IQ は、ANSINULL オプションが OFF に設定されていない限り、Null 比較に ANSI セマンティックを使用します。OFF に設定されている場合は、比較は Adaptive Server Enterprise 互換で行われます。

注意 SQL Anywhere 8.0 以降では、NULL 値に空の文字列をマッピングする際、Adaptive Server Enterprise との互換性を実現するために TDS_EMPTY_STRING_AS_NULL のサポートが追加されています。

長さがゼロの文字列

長さがゼロの文字列は次のように取り扱われます。

- Adaptive Server Enterprise は、長さがゼロの文字列を NULL 値と見なします。

Adaptive Server Enterprise で空の文字列を作成するには、空白を 1 つ格納します。

- SQL Anywhere と Sybase IQ は、ANSI セマンティックに従い、長さがゼロの文字列を null ではなく値として扱います。

HOLDLOCK、SHARED、FOR BROWSE

この構文のサポートは次のとおりです。

- Adaptive Server Enterprise は HOLDLOCK、SHARED、FOR BROWSE の各構文をサポートしています。
- SQL Anywhere は HOLDLOCK をサポートしていますが、SHARED と FOR BROWSE はサポートしていません。
- Sybase IQ はこれらのキーワードをサポートしていません。

SQL 関数

Sybase IQ は SQL Anywhere や Adaptive Server Enterprise とほぼ同じ関数をサポートしていますが、次のような違いがあります。

- Adaptive Server Enterprise は USING CHARACTERS | USING BYTES 構文を PatIndex() でサポートしていますが、SQL Anywhere と Sybase IQ はサポートしていません。
- Adaptive Server Enterprise は Reverse() 関数をサポートしていますが、SQL Anywhere と Sybase IQ はサポートしていません。
- Adaptive Server Enterprise は Len() を Length() の代わりにサポートしていますが、SQL Anywhere はサポートしていません。
- Adaptive Server Enterprise は Square() と Str_Replace() を Microsoft 互換の関数としてサポートしていますが、SQL Anywhere はサポートしていません。
- Sybase IQ は Str_Replace() をサポートしています。
- Adaptive Server Enterprise と SQL Anywhere では、変更時刻を確認するための 2 つのタイムスタンプの比較に TSEQUAL() を使用できますが、Sybase IQ では TSEQUAL() はサポートされていません。(TSEQUAL は Sybase IQ テーブル・レベルのバージョン・モデルには関係しません。)
- Sybase IQ は ROWID() をサポートしています。Adaptive Server Enterprise と SQL Anywhere はサポートしていません。
- SQL Anywhere と Sybase IQ は、データ型の変換に、Adaptive Server Enterprise の Convert() に加え、Cast() をサポートしています。

注意 Cast() は ANSI 互換の名前です。

- SQL Anywhere と Sybase IQ は、Lcase() と Ucase() を Lower() と Upper() のシノニムとしてサポートしています。Adaptive Server Enterprise はサポートしていません。
- SQL Anywhere と Sybase IQ は、文字列関数 Locate() をサポートしていますが、Adaptive Server Enterprise はサポートしていません。
- SQL Anywhere は、文字列を適切なデータ型に変換する機能をテストする IsDate() および IsNumeric() 関数をサポートしていますが、Adaptive Server Enterprise はサポートしていません。Sybase IQ は IsDate() をサポートしています。IsNumeric は Sybase IQ で使用できますが、CIS 機能補正のパフォーマンスへの影響を考慮する必要があります。
- SQL Anywhere は NEWID、STRTOUID、UIDTOSTR の各関数をサポートしていますが、Adaptive Server Enterprise はサポートしていません。これらの関数は Sybase IQ のネイティブ関数なので、CIS 機能補正のパフォーマンスへの影響を考慮する必要はありません。

注意 SOUNDEX 文字列関数、DIFFERENCE 文字列関数、日付関数の一部など、Sybase IQ と SQL Anywhere では動作が異なる SQL 関数があります。Sybase IQ のデータベース・オプション ASE_FUNCTION_BEHAVIOR は、Sybase IQ の一部のデータ型変換関数 (HEXTOINT や INTTOHEX など) の出力を、Adaptive Server Enterprise の関数の出力と整合性があるものになるよう指定します。

OLAP 関数

Sybase IQ が現在サポートしている OLAP 関数は次のとおりです。

- Corr()
- Covar_Pop()
- Covar_Samp()
- Cume_Dist
- Dense_Rank()
- Exp_Weighted_Avg
- First_Value

- Last_Value
- Median
- Ntile()
- Percent_Rank()
- Percentile_Cont()
- Percentile_Disc()
- Rank()
- Regr_Avgx()
- Regr_Avgy()
- Regr_Intercept()
- Regr_R2
- Regr_Slope()
- Regr_Sxx()
- Regr_Sxy()
- Regr_Syy()
- StdDev()
- Stddev_Pop
- Stddev_Samp
- Var_Pop
- Var_Samp
- Variance()
- Weighted_Avg

SQL Anywhere はすべての Sybase IQ OLAP 関数をサポートしています。

現在、Adaptive Server Enterprise は OLAP 関数をサポートしていません。

CIS の機能補正では、OLAP 関数はサポートされていません。

注意 OLAP 関数のサポートは、Sybase の製品開発で急速に発展している分野です。詳細については、「[第 4 章 SQL 関数](#)」を参照してください。さらに、『システム管理ガイド 第 2 巻』の「[第 2 章 OLAP の使用](#)」も参照してください。

システム関数

SQL Anywhere と Sybase IQ は次の Adaptive Server Enterprise システム関数をサポートしていません。

- `curunreservedpgs()` – DB 領域の空きページ数。
- `data_pgs()` – 個々のテーブルまたはインデックスによって使用されているページ数。
- `host_id()` – サーバ・プロセスの UNIX pid。
- `hos_name()` – サーバを実行しているマシンの名前。
- `lct_admin()` – トランザクション・マネージャの「ラストチャンス・スレッシュホールド」の管理に使用します。
- `reserved_pgs()` – テーブルまたはインデックスに割り当てられたページ数。
- `rowcnt()` – 指定されたテーブル内のロー数。
- `valid_name()` – 名前が、テーブルなどに使用された場合に有効かどうかを示します。
- `valid_user()` – そのユーザが接続パーミッションを持っている場合に TRUE を返します。
- `ptn_data_pgs()` – パーティション内のデータ・ページ数。
- `index_colorder()` – インデックス内のカラムの順番を返します。

ユーザ定義関数

ユーザ定義関数 (UDF: User Defined Function) のサポートには次のような違いがあります。

- SQL Anywhere は、SQL、Java、C で UDF をサポートしています。

- Adaptive Server Enterprise は、Java で書かれた UDF しかサポートしません。
- Sybase IQ では、CIS クエリ分解を通じて UDF をサポートしていますが、パフォーマンスに影響があります。

日付の算術式

SQL Anywhere と Sybase IQ は、日付に使われる算術式を、さまざまな日付関数の省略形として解釈します。Adaptive Server Enterprise はこのような解釈をしません。

- 日付 +/- 整数 は、Dateadd() と同義です。
- 日付 - 日付 は、Datediff() と同義です。
- 日付 + 時刻 は、両者を合わせてタイムスタンプを作成します。

SELECT INTO

次のような文で許可されるテーブルの種類に違いがあります。

```
select into table1 from table2
```

- Adaptive Server Enterprise では、*table1* が永久テーブル、テンポラリ・テーブル、プロキシ・テーブルのいずれであってもかまいません。Adaptive Server Enterprise ではさらに、SELECT INTO EXISTING TABLE をサポートしています。
- SQL Anywhere と Sybase IQ では、*table1* は永久テーブルまたはテンポラリ・テーブルのいずれかになります。永久テーブルは、select into *table* を実行して複数のカラムを指定した場合にのみ作成されます。所有者を指定せずに SELECT INTO #*table* を実行すると、指定されたカラムの数にかかわらずテンポラリ・テーブルが作成されます。1つしかカラムを持たないテーブルで SELECT INTO を実行すると、ホスト変数が選択されます。

更新可能なビュー

Adaptive Server Enterprise と SQL Anywhere では、WITH CHECK オプションが要求されていない場合に更新可能となるビュー定義に関しては ANSI よりも自由度が高くなっています。

SQL Anywhere には、SQL92 でサポートされているもののみを更新可能にするか、あるいはもっと自由度の高いルールを設定するかを制御できる ANSI_UPDATE_CONSTRAINTS オプションが用意されています。

Sybase IQ は、フラット化できる単一のテーブルのビューでの UPDATE のみをサポートしています。Sybase IQ は WITH CHECK をサポートしていません。

UPDATE と DELETE の FROM 句

3つの製品はすべて、UPDATE と DELETE で複数のテーブルの FROM 句をサポートしています。

Transact-SQL のプロシージャ言語の概要

「ストアド・プロシージャ言語」は SQL の一部として、ストアド・プロシージャとバッチで使用します。

SQL Anywhere と Sybase IQ は、SQL92 に基づく Watcom-SQL ダイアレクトに加えて、Transact-SQL ストアド・プロシージャ言語の大部分をサポートしています。

Transact-SQL のストアド・プロシージャの概要

SQL Anywhere と Sybase IQ のストアド・プロシージャ言語は、ISO/ANSI 規格を基準にしているので、Transact-SQL ダイアレクトとは多くの点で異なっています。概念と機能の多くは似ていますが、構文が異なります。概念が似ているので、SQL Anywhere と Sybase IQ の Transact-SQL サポートでは、両者の間の自動変換ができます。ただし、プロシージャはどちらかのダイアレクトだけで記述する必要があり、混在させることはできません。

ここでは、Transact-SQL ストアド・プロシージャに対する SQL Anywhere と Sybase IQ のサポートについて、次のような側面から説明します。

- パラメータを引き渡す

- 結果セットを返す
- ステータス情報を返す
- パラメータにデフォルト値を提供する
- 制御文
- エラー処理

Transact-SQL のバッチの概要

Transact-SQL のバッチは、一緒に送信されてグループとして次々に実行される一連の SQL 文です。バッチはコマンド・ファイルとして保存できます。SQL Anywhere と Sybase IQ の ISQL ユーティリティと Adaptive Server Enterprise の *isql* ユーティリティは、バッチを対話型で実行するためのよく似た機能を持っています。

プロシージャで使われる制御文はバッチでも使えます。SQL Anywhere と Sybase IQ は、バッチでの制御文の使用をサポートし、Transact-SQL のように、一連の文を区切りなしで使用できますが、GO 文がバッチの終わりを示します。

コマンド・ファイルに保存されているバッチでは、SQL Anywhere と Sybase IQ はコマンド・ファイルでのパラメータの使用をサポートしています。Adaptive Server Enterprise はパラメータをサポートしません。

パラメータの詳細については、『リファレンス：文とオプション』の「PARAMETERS 文 [DBISQL]」を参照してください。

プロシージャとバッチ内の SQL 文

Sybase IQ がサポートする SQL 文には、一方のダイアレクトの一部になっていて、もう一方のダイアレクトの一部になっていないものがあります。そのため、2 つのダイアレクトを 1 つのプロシージャやバッチ内で混在させることはできません。したがって、次の条件で使用してください。

- Transact-SQL 専用の文は、両方のダイアレクトに含まれている文と共に、バッチまたはプロシージャ内に含めることができます。

- Adaptive Server Enterprise によってサポートされない文は、両サーバによってサポートされる文とともに、バッチまたはプロシージャ内に含めることができます。
- Transact-SQL 専用の文と Sybase IQ 専用の文は、バッチまたはプロシージャ内に混在させることはできません。

セミコロンで区切られていない SQL 文は、Transact-SQL のプロシージャまたはバッチの一部です。個々の文の詳細については、『リファレンス：文とオプション』を参照してください。

IF 文内の式サブクエリ

Adaptive Server Enterprise と SQL Anywhere は、式サブクエリによって返されたスカラ値と変数との比較をサポートしています。次に例を示します。

```
create procedure testIf ()
begin
  declare var4 int;
set var4 = 10;
  if var4 = (select MIN (a_i1) from a) then set
    var4 = 100;
end if;
end;
```

CASE 文

Sybase IQ と SQL Anywhere では、CASE 文の許可されている使用方法が異なります。

Adaptive Server Enterprise では、case 式のみがサポートされ、CASE 文はサポートされません。

Sybase IQ と Adaptive Server Enterprise でサポートされる case 式の詳細な比較については、「[式](#) (25 ページ) を参照してください。

ロー・レベルのカーソル演算

3つの製品いずれにおいても、次の例のように UPDATE と DELETE でカーソルを使用できます。

```
UPDATE WHERE CURRENT OF {cursor}
DELETE WHERE CURRENT OF {cursor}
```


Sybase IQ では、更新可能なカーソルは、`asensitive` のみ、1つのテーブル用のみ、連鎖のみです。更新可能なホールド・カーソルは使用できません。Sybase IQ では、更新可能なカーソルはテーブル・ロックされます。

Print コマンド

PRINT コマンドの実行結果はクライアントに依存します。

- Adaptive Server Enterprise では、PRINT は常にクライアントにメッセージを送信します。
- SQL Anywhere と Sybase IQ では、PRINT は Open Client と JDBC 接続のクライアントにメッセージを送信します。
- PRINT に依存する Adaptive Server Enterprise のストアド・プロシージャは、Sybase IQ では DBISQL を使って機能します。

注意 Sybase IQ ユーザは、iAdaptive Server Anywhere JDBC ドライバ (以前 JDBC-ODBC ブリッジと呼ばれていたもの) ではなく、JDBC で `dbisql` を使用するようおすすめします。

ストアド・プロシージャの自動変換

Transact-SQL 代替構文のサポートのほかに、SQL Anywhere と Sybase IQ では、Watcom-SQL ダイアレクトと Transact-SQL ダイアレクト間での文の変換を支援します。次に示す関数は、SQL 文に関する情報を返し、SQL 文を自動変換できるようにします。

- **SQLDialect(文)** Watcom-SQL または Transact-SQL を返します。
- **WatcomSQL(文)** その文の Watcom-SQL 構文を返します。
- **TransactSQL(文)** その文の Transact-SQL 構文を返します。

これらは関数なので、ISQL から SELECT 文を使用してアクセスできます。たとえば、次の文は、Watcom-SQL という値を返します。

```
SELECT SqlDialect('select * from Employees')
```

Transact-SQL プロシージャから結果セットを返す

SQL Anywhere と Sybase IQ は RESULT 句を使用して、返される結果セットを指定します。Transact-SQL プロシージャでは、最初のクエリのカラム名またはエイリアス名が呼び出し元の環境に返されます。

Transact-SQL プロシージャの例

次の Transact-SQL プロシージャは、Transact-SQL ストアド・プロシージャが結果セットを返す方法を示します。

```
CREATE PROCEDURE showdept (@deptname varchar(30))
AS
    SELECT Employees.Surname, Employees.GivenName
    FROM Departments, Employees
    WHERE Departments.DepartmentName = @deptname
    AND Departments.DepartmentID =
    Employees.DepartmentID
```

Watcom-SQL プロシージャの例

対応する SQL Anywhere または Sybase IQ のプロシージャは次のようになります。

```
CREATE PROCEDURE showdept(in deptname varchar(30))
RESULT ( lastname char(20), firstname char(20))
BEGIN
    SELECT Employees.Surname, Employees.GivenName
    FROM Departments, Employees
    WHERE Departments.DepartmentName = deptname
    AND Departments.DepartmentID =
    Employees.DepartmentID
END
```

複数の結果セット

3 つの Sybase クライアント・ツールが複数の結果セットをクライアントに表示する方法には多少違いがあります。

- ISQL は 1 つのストリームにすべての結果を表示します。
- DBISQL は結果セットをそれぞれ別のタブに表示します。ユーザはこの機能を [オプション] メニューで有効にする必要があります。これを永続的な変更とした後、DBISQL を再起動するか再接続してください。
- DBISQLC には、各結果セットを次々に表示する RESUME コマンドが用意されています。

プロシージャと結果の詳細については、『システム管理ガイド第2巻』の「[第1章 プロシージャとバッチの使用](#)」を参照してください。

Transact-SQL プロシージャ内の変数

SQL Anywhere と Sybase IQ は SET 文を使用して、プロシージャ中の変数に値を割り当てます。Transact-SQL では、SELECT 文と空のテーブル・リストを使用して値を割り当てます。次の単純なプロシージャは、Transact-SQL 構文の働きを示します。

```
CREATE PROCEDURE multiply
    @mult1 int,
    @mult2 int,
    @result int output
AS
SELECT @result = @mult1 * @mult2
```

このプロシージャを呼び出すには、次のようにします。

```
CREATE VARIABLE @product int
go
EXECUTE multiply 5, 6, @product OUTPUT
go
```

プロシージャの実行後、変数 `@product` の値は 30 になります。

変数の順序と持続性

変数宣言の順序と持続性にはいくつかの違いがあります。

- Adaptive Server Enterprise では、ストアド・プロシージャ本体のどこでも変数を宣言できます。変数はプロシージャの実行中持続します。
- SQL Anywhere と Sybase IQ では、複合文の最初(たとえば、BEGIN...END ペア内で BEGIN の直後)に変数を宣言する必要があります。変数は、複合文の実行中しか持続しません。

SELECT 文の使用による変数割り当ての詳細については、「[互換性のあるクエリの書き方](#)」(626 ページ)を参照してください。SET 文の使用による変数割り当ての詳細については、『リファレンス：文とオプション』の「[SET 文 \[ESQL\]](#)」を参照してください。

Transact-SQL プロシージャでのエラー処理

デフォルトでのプロシージャのエラー処理は、Watcom-SQL ダイアレクトと Transact-SQL ダイアレクトでは異なります。デフォルトでは、Watcom-SQL ダイアレクトのプロシージャはエラーが発生すると終了し、呼び出し元の環境に SQLSTATE 値と SQLCODE 値を返します。

EXCEPTION 文を使って、Watcom-SQL スタアド・プロシージャに明示的なエラー処理を組み込むことができます。または、ON EXCEPTION RESUME 文を使って、エラーが発生したら、その次の文から実行を再開するよう、プロシージャに指示することもできます。

Transact-SQL ダイアレクトのプロシージャでエラーが発生した場合は、その次の文から実行が継続されます。グローバル変数 @@error には、最後に実行された文のエラー・ステータスが保存されます。文の後でこの変数をチェックして、プロシージャから戻るように強制できます。たとえば、次の文によって、エラーが発生すると終了して戻るようにできます。

```
IF @@error != 0 RETURN
```

プロシージャが実行を終了したときの戻り値で、プロシージャが成功したかどうかわかります。この戻り値は整数で、次のようにしてアクセスできます。

```
DECLARE @status INT
EXECUTE @status = proc_sample
IF @status = 0
    PRINT 'procedure succeeded'
ELSE
    PRINT 'procedure failed'
```

表 A-2 に、組み込みプロシージャの戻り値とその意味を示します。

表 A-2 : 組み込みプロシージャの戻り値

| 値 | 意味 |
|-----|--------------------------|
| 0 | プロシージャはエラーなしで実行された |
| -1 | オブジェクトがない |
| -2 | データ型エラー |
| -3 | プロセスがデッドロックの被害対象として選択された |
| -4 | パーミッション・エラー |
| -5 | 構文エラー |
| -6 | その他のさまざまなユーザ・エラー |
| -7 | 領域不足などのリソース・エラー |
| -8 | 致命的ではない内部の問題 |
| -9 | システムの限界に達した |
| -10 | 致命的な内部不整合 |
| -11 | 致命的な内部不整合 |
| -12 | テーブルまたはインデックスが破壊されている |

| 値 | 意味 |
|-----|----------------|
| -13 | データベースが破壊されている |
| -14 | ハードウェア・エラー |

RETURN 文は、この表の値以外の、ユーザが意味を定義した整数値を返すこともできます。

プロシージャの中での RAISERROR 文の使用

RAISERROR 文は、ユーザ定義エラーを生成するための Transact-SQL 文です。これは、SIGNAL 文と似た機能を持っています。

RAISERROR 文の詳細な説明は、『リファレンス：文とオプション』の「[RAISERROR 文 \[T-SQL\]](#)」を参照してください。

RAISERROR 文そのものは、プロシージャを終了しませんが、ユーザ定義エラー後の実行を制御するために、RETURN 文やグローバル変数 @@error のテストと組み合わせることができます。

ON_TSQL_ERROR データベース・オプションを CONTINUE に設定すると、RAISERROR 文はエラーによる実行終了を通知しません。代わりに、プロシージャは実行を完了して RAISERROR のステータス・コードとメッセージを保存し、最新の RAISERROR を返します。RAISERROR を返すプロシージャが別のプロシージャから呼び出された場合は、もっとも外側のプロシージャが終了してから RAISERROR が返ります。

途中の RAISERROR ステータスとコードは、プロシージャが終了すると失われます。RAISERROR が返るときにエラーが発生した場合は、エラー情報が返され、RAISERROR 情報は失われます。アプリケーションは異なる実行ポイントで @@error グローバル変数を調べて、途中の RAISERROR ステータスを問い合わせることができます。

Watcom-SQL ダイアレクトでの Transact-SQL のようなエラー処理

CREATE PROCEDURE 文に ON EXCEPTION RESUME 句を追加して、Watcom-SQL ダイアレクトのプロシージャが Transact-SQL に似た方法でエラー処理を行うようにできます。

```
CREATE PROCEDURE sample_proc ()
ON EXCEPTION RESUME
```

```
BEGIN
    ...
END
```

ON EXCEPTION RESUME 句があると、明示的な例外処理コードは実行されないの、これらの 2 つの句は一緒に使用しないようにしてください。

SQL Anywhere と Sybase IQ

ここまでの各項では、Transact-SQL との互換性に焦点を当てつつ、Sybase IQ と SQL Anywhere の数々の差異を明らかにしてきました。

この項では、Sybase IQ と SQL Anywhere のその他の差異と機能の共通点を指摘します。

詳細については、Sybase IQ を使用している場合は必ずそのマニュアル・セットを参照してください。SQL Anywhere を使用している場合や、Sybase IQ のマニュアルで具体的な SQL Anywhere の機能のみを参照している場合は、SQL Anywhere のマニュアル・セットを参照してください。

サーバとデータベースの起動と管理

データベースおよびサーバの起動と管理では、次の違いに注意してください。

- Sybase IQ は、サーバ起動コマンド `start_iq` を使用し、SQL Anywhere のネットワーク・サーバ起動コマンドは使用しません。
- Sybase IQ では、パーソナル・サーバはサポートされていません。
- Sybase IQ は、SQL Anywhere のサーバ用コマンドライン・オプションの多くをサポートしていますが、すべてではありません。Sybase IQ でサポートされていて SQL Anywhere ではサポートされていないサーバ・オプションもあります。
- Sybase IQ には、サーバをシャットダウンするための `stop_iq` ユーティリティが用意されています。

- BACKUP と RESTORE 文で使用できる句は Sybase IQ と SQL Anywhere では異なります。
- Sybase IQ では、マルチプレックス操作にのみ SQL Remote がサポートされています。

Sybase IQ は、SQL Anywhere のデータベース管理ユーティリティの多くをサポートしていますが、すべてではありません。

- Sybase IQ でサポートされていない SQL Anywhere ユーティリティは、backup、compression、console、initialization、license、log transfer、log translation、rebuild、spawn、transaction log オプションの一部 (-g、-il、-ir、-n、-x、-z)、uncompression、unload、upgrade、write file です。
- Sybase IQ は SQL Anywhere の validation ユーティリティをカタログ・ストアでのみサポートしています。IQ ストアを検証するには、sp_iqcheckdb を使用します。

データベース・オプション

SQL Anywhere のデータベース・オプションの中には、Sybase IQ によってサポートされていないものがあります。

DEFAULT_TIMESTAMP_INCREMENT もその 1 つです。

データベース・オプションには、カタログ・ストアのみに適用されるものがあります。これには、FOR_XML_NULL_TREATMENT、ISOLATION_LEVEL、PREFETCH、PRECISION、SCALE があります。

動作、デフォルト値、または指定可能な値が異なるオプションには、DELAYED_COMMITS、TIME_FORMAT、TIMESTAMP_FORMAT があります。

Sybase IQ には、SQL Anywhere でサポートされていない多くのオプションがあります。詳細については、『リファレンス：文とオプション』の「第 2 章 データベース・オプション」を参照してください。

データ定義言語 (DDL)

前述の DDL の差異に加え、次の点に注意してください。

- ALTER TABLE 文の DELETE/DROP 句や PRIMARY KEY 句では、Sybase IQ は RESTRICT アクションを取ります (関連付けられた外部キーがある場合はエラーが報告されます)。SQL Anywhere は常に CASCADE アクションを取ります。

- 同様に、Sybase IQ では、関連付けられた外部キー制約がある場合、DROP TABLE 文もエラーを報告します。
- Sybase IQ は、DDL 文の CREATE COMPRESSED DATABASE、CREATE TRIGGER、SETUSER をサポートしていません。
- SQL Anywhere では CREATE TABLE の CHECK ON COMMIT 句でトランザクション・レベルの整合性をサポートしていますが、Sybase IQ は文レベルの参照整合性をサポートしています。
- Sybase IQ テーブルは SQL Anywhere (または Catalog) テーブルを参照する外部キーを持つことができません。また、SQL Anywhere テーブルは Sybase IQ テーブルを参照する外部キーを持つことができません。
- Sybase IQ データベースでは、パブリケーションは、SQL Anywhere テーブル上でしか作成できません。
- CREATE DATABASE では、大文字小文字の区別と照合のデフォルトに違いがあります。Sybase IQ でのデフォルトは CASE RESPECT と ISO_BINENG 照合です。SQL Anywhere では、デフォルトは CASE IGNORE で、照合はオペレーティング・システムの言語と文字セットから判断されます。

データ操作言語 (DML)

- Sybase IQ でサポートしていない DML および手続き型文は、EXPLAIN、GET DATA、INPUT、PREPARE TO COMMIT、PUT、READTEXT、ROLLBACK TRIGGER、SYSTEM、UNLOAD TABLE、VALIDATE TABLE です。

注意 一連の抽出オプションが、UNLOAD TABLE と類似した役割をします。詳細については、『システム管理ガイド 第 1 巻』、「[第 7 章 データベースへのデータの入出力](#)」の「[データ抽出オプション](#)」の項を参照してください。

- Sybase IQ は INSERT...LOCATION 構文をサポートしますが、SQL Anywhere はサポートしません。
- LOAD TABLE オプションは Sybase IQ と SQL Anywhere では異なります。
- Sybase IQ の OPEN 文 では BLOCK 句と ISOLATION LEVEL 句を使用できません。

- Sybase IQ ではトリガをサポートしていません。
- トランザクション、独立性レベル、チェックポイント、自動生成される COMMIT の使用方法は、カーソル・サポートと同様に Sybase IQ と SQL Anywhere では異なります。詳細については、『システム管理ガイド 第 1 巻』の「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。
- Sybase IQ のストアド・プロシージャから SELECT を実行する場合、CIS の機能補正によるパフォーマンスへの影響を考慮する必要があります。詳細については、『パフォーマンス&チューニング・ガイド』の「[SQL Anywhere による処理を引き起こす条件](#)」を参照してください。
- Sybase IQ は Adaptive Server Enterprise の SELECT 文内にある完全修飾名のデータベース名の修飾子を無視します。たとえば、FROM 句の `<database name>.<owner>.<table name>` という構文では、Sybase IQ は `SELECT * FROM XXX..TEST` を `SELECT * FROM TEST` として解釈します。

ストアド・プロシージャ

詳細については、「[SQL Anywhere でサポートされているプロシージャ](#)」(525 ページ)を参照してください。

Sybase IQ 12.6 ESD #2 かそれ以上を使用して ALTER DATABASE UPGRADE コマンドを実行すると、旧式のプロシージャとビューが Sybase IQ 12.6 ESD #2 より前に作成されたデータベースから削除されます。

ALTER DATABASE UPGRADE を実行しない場合、それらのプロシージャはデータベースに残ったままになります。

Adaptive Server Enterprise と Sybase IQ

この項では、Sybase IQ と Adaptive Server Enterprise のその他の差異と機能の共通点を指摘します。

詳細については、Sybase IQ を使用している場合は必ずそのマニュアル・セットを参照してください。Adaptive Server Enterprise を使用している場合や、Sybase IQ のマニュアルで具体的な Adaptive Server Enterprise の機能を参照している場合は、Adaptive Server Enterprise のマニュアル・セットを参照してください。

ストアド・プロシージャ

Sybase IQ では次の Adaptive Server ストアド・プロシージャが使用できなくなりました。

- `sp_addserver`
- `sp_configure`
- `sp_estspace`
- `sp_help`
- `sp_helpuser`
- `sp_who`

Sybase IQ では次のカタログ・プロシージャが使用できなくなりました。

- `sp_column_privileges`
- `sp_databases`
- `sp_datatype_info`
- `sp_server_info`

Sybase IQ 15.1 ESD #2 かそれ以上を使用して ALTER DATABASE UPGRADE コマンドを実行すると、旧式のプロシージャとビューが Sybase IQ 15.1 ESD #2 より前に作成されたデータベースから削除されます。

ALTER DATABASE UPGRADE を実行しない場合、それらのプロシージャはデータベースに残ったままになります。

システム・ビュー

Sybase IQ では次の Adaptive Server Enterprise ビューが使用できなくなりました。

- `sysalternates`

- sysaudits
- sysauditoptions
- sysconstraints
- syscharsets
- sysconfigures
- syscurconfigs
- sysdatabases
- sysdepends
- sysdevices
- sysengines
- syskeys
- syslanguages
- syslocks
- syslogs
- sysloginroles
- sysmessages
- sysprocedures
- sysprocesses
- sysprotects
- sysreferences
- sysremotelogins
- sysroles
- syssegments
- syssservers
- sysssrvroles
- systhresholds
- sysusages

カラム名の違い

Adaptive Server Enterprise の SYSTYPES ビューで使用されるカラム名は“allownulls”です。Sybase IQ の SYSTYPES ビューで使用されるカラム名は“allowsnulls”です。

索引

記号

¥ 円記号

SQL 識別子で使用できない文字 22

数字

3 値的論理

NULL 値 67

説明 47

A

ABS 関数 128

ACOS 関数 128

Adaptive Server Enterprise

互換性 601

ALL

条件 40

AND 条件 46

ANY

条件 40

ARGN 関数 129

ASCII 関数 129

ASCII 値 129, 136

ASIN 関数 130

ATAN2 関数 131

ATAN 関数 130

AVG 関数 131

B

BETWEEN 条件 41

BIGINTTOHEX 関数 132

BINARY データ型 78

BINARY ロード形式

LOAD TABLE 95

データ・ファイル 95

BIT_LENGTH 関数 133

BIT データ型

Transact-SQL 83

互換性 608

BYTE_LENGTH 関数 133

C

CASE 式 31

NULLIF 関数 214

CAST 関数 90, 134

CEILING 関数 135

CEIL 関数 135

CHAR_LENGTH 関数 136

CHARACTER VARYING データ型

後続ブランクの削除 71, 609

説明 70

CHARACTER データ型

説明 70

CHARINDEX 関数 137

CHAR 関数 136

CHAR データ型

説明 69

CHECKPOINT 文

チェックポイント中のバックアップ 491

CHECK 条件

Transact-SQL 617

COALESCE 関数 138

COL_LENGTH 関数 138

COL_NAME 関数 139

command-line options

上書き 503

COMPUTE 句

Transact-SQL 628

CONNECTION_PROPERTY 関数 139

索引

CONTAINS 条件 45
CONVERT 関数 90, 140
 整数から日付への変換 142
 日付から整数への変換 142
 日付から文字列への変換 142
 文字列から日付への変換 142
CORR 関数 142
COS 関数 143
COT 関数 144
COUNT 関数 146
COVAR_POP 関数 144
COVAR_SAMP 関数 145
CPU の使用率
 データベース一貫性チェッカ 359
CREATE DEFAULT 文
 サポート対象外 620
CREATE DOMAIN 文
 Transact-SQL との互換性 620
 使用 89
CREATE EXISTING TABLE 文
 プロキシ・テーブル 514
CREATE INDEX 文
 IQ 621
 Transact-SQL 621
CREATE RULE 文
 サポート対象外 620
CREATE TABLE 文
 Transact-SQL 617
CREATE TRIGGER
 サポートなし 621
CUBE 処理
 GROUPING 関数 178
CUME_DIST 関数 147
CURRENT DATABASE
 特別値 54
CURRENT DATE
 デフォルト 54
 特別値 54
CURRENT PUBLISHER
 デフォルト 54
 特別値 54
CURRENT TIME
 デフォルト 54
 特別値 54

CURRENT TIMESTAMP
 デフォルト 55
 特別値 55
CURRENT USER
 デフォルト 55
 特別値 55

D

DATALength 関数 148
DATE_ORDER オプション 87
DATEADD 関数 149
DATECEILING 関数 150
DATEDIFF 関数 152
DATEFLOOR 関数 154
DATEFORMAT 関数 156
DATENAME 関数 157
DATEPART 関数 157
DATEROUND 関数 158
DATETIME 関数 160
DATE 関数 149
DATE データ型 83
DAYNAME 関数 160
DAYS 関数 161
DAY 関数 160
DB_ID 関数 162
DB_NAME 関数 162
DB_PROPERTY 関数 163
DBCC
 実行時間 363
 出力 364
 データベースの検証 357
 パフォーマンス 363
dbcc
 スレッド使用 359
DBCC_LOG_PROGRESS オプション 364
dbinit
 サポートなし 605
DB 領域
 管理 605
 読み込み／書き込み操作の防止 362
DDL
 SQL Anywhere 645
DECIMAL データ型 73

DEGREES 関数 164
 DENSE_RANK 関数 164
 DIFFERENCE 関数 165
 DISK 文
 サポート対象外 605
 DML
 SQL Anywhere 646
 DOUBLE データ型 75
 DOW 関数 166
 dropleaks モード 362
 DUMMY テーブル 600

E

ELSE
 IF 式 30
 ENDIF
 IF 式 30
 ERRORMSG 関数
 SQL 構文 167
 EVENT_CONDITION_NAME 関数 169
 EVENT_CONDITION 関数 167
 EVENT_PARAMETER 関数 169
 EXISTS 条件 45
 EXP_WEIGHTED_AVG 関数 171
 EXP 関数 170

F

FIRST_VALUE 関数 172
 FLOAT_AS_DOUBLE オプション 613
 FLOAT データ型 75
 FLOOR 関数 174
 FOR BROWSE 構文
 Transact-SQL 631
 FP インデックス
 確認 361
 FROM 句 115
 UPDATE と DELETE 636

G

GETDATE 関数 175
 Globally Unique Identifier
 NEWID 関数の SQL 構文 208
 GRAPHICAL_PLAN 関数 175
 GROUP BY
 互換性 627
 GROUP_MEMBER 関数
 SQL 構文 178
 GROUPING 関数 178
 GUID
 NEWID 関数の SQL 構文 208
 STRTOUUID 関数の SQL 構文 264
 UUIDTOSTR 関数の SQL 構文 330

H

HEXTOBIGINT 関数 179
 HEXTOINT 関数 180
 ASE_FUNCTION_BEHAVIOR オプション 180
 HOLDLOCK 構文
 Transact-SQL 631
 HOURS 関数 181
 HOUR 関数 181
 HTML_DECODE 関数 182
 HTML_ENCODE 関数 183
 HTML_PLAN 184
 HTML_PLAN 関数 184
 HTTP
 オプションの設定 511
 ヘッダの設定 510
 HTTP_DECODE 関数 186
 HTTP_ENCODE 関数 186
 HTTP_HEADER 関数 187
 HTTP_VARIABLE 関数 188
 HTTP 関数 115
 HTML_DECODE 182
 HTML_ENCODE 183
 HTTP_DECODE 186
 HTTP_ENCODE 186
 HTTP_HEADER 187
 HTTP_VARIABLE 188
 NEXT_HTTP_HEADER 211
 NEXT_HTTP_VARIABLE 211

索引

I

identity カラム
 互換性 619
 デフォルト値としてのサポート 619

IFNULL 関数 189

IF 式 30

IMAGE データ型 78, 610
 互換性 614

IMSL ライブラリ
 エラー処理 126
 エラー・ロギング 127
 接続 125

INDEX_COL 関数 189

INSERTSR 関数 190

INTEGER データ型 73

INTTOHEX 関数 191
 ASE_FUNCTION_BEHAVIOR オプション 191

IN 条件 44

IQ Agent
 待機時間 10
 ポート 9

iq_dummy テーブル 104

IQDIR15 環境変数 8

IQLOGDIR15 環境変数 9

IQMsgMaxSize サーバ・オプション 504

IQMsgNumFiles サーバ・プロパティ 505

IQPORT 環境変数 9

IQTIMEOUT 環境変数
 IQ Agent の待機時間の指定 10

IQTMP15 環境変数 10

IQ ストア 606

IS NULL 条件 46

ISDATE 関数
 SQL 構文 192

ISNULL 関数 193

ISNUMERIC 関数
 SQL 構文 193

J

Java
 ユーザ定義関数 123

Java Runtime Environment
 設定 14

Java データ型
 互換性 614

L

LAST USER
 特別値 55

LAST_VALUE 関数 194

LCASE 関数 196

LD_LIBRARY_PATH 環境変数 12

LEFT 関数 196

LENGTH 関数 198

LEN 関数
 SQL 構文 197

LIBPATH 環境変数 12

LIKE 条件 42

LOAD TABLE
 BINARY 95
 バイナリ・フォーマット 95
 バイナリ・フォーマット・データ・
 ファイル 95

LOCATE 関数 199

LOG10 関数 201

LOG 関数 200

LONG BINARY データ型 78, 610, 614

LOWER 関数 201

LTRIM 関数 202

LVC セル 362

M

main_cache_memory_mb
 sa_server_option パラメータ 506

master データベース
 サポート対象外 604

MAX 関数 202

MEDIAN 関数 203

MINUTES 関数 205

MINUTE 関数 205

MIN 関数 204

MOD 関数 206

MONEY データ型 77

MONTHNAME 関数 206

MONTHS 関数 207
 MONTH 関数 206
 MPXServerName カラム 373

N

NEWID 関数
 SQL 構文 208
 NEXT_CONNECTION 関数 209
 NEXT_DATABASE 関数 210
 NEXT_HTTP_HEADER 関数 211
 NEXT_HTTP_VARIABLE 関数 211
 NOT 条件 47
 NOW 関数 212
 NTILE 関数 212
 NULL
 Transact-SQL との互換性 617
 NULLIF 関数 32, 214
 NULL 値
 説明 67
 null 比較
 Transact-SQL 630
 NUMBER 関数 215
 NUMERIC 75

O

OBJECT_ID 関数 216
 OBJECT_NAME 関数 216
 OCTET_LENGTH 関数 217
 OLAP
 DENSE_RANK 関数 164
 GROUPING 関数 178
 NTILE 関数 212
 PERCENT_RANK 関数 219
 PERCENTILE_CONT 関数 220
 PERCENTILE_DISC 関数 223
 RANK 関数 230
 STDDEV 関数 258
 VARIANCE 関数 333
 ウィンドウ関数 107
 ウィンドウ関数の種類 107
 ウィンドウ指定 107

ウィンドウ集合関数 106
 ウィンドウ名 107
 数値関数 107
 統計関数 107
 分散統計関数 107
 ランク付け関数 106
 OLAP OVER 句 107
 OLAP 関数
 互換性 632
 ON EXCEPTION RESUME 句
 Transact-SQL 643
 Open Client の設定値 15
 OR キーワード 46
 OVER 句 107

P

PATH 環境変数 12
 PATINDEX 関数 217
 PERCENT_RANK 関数 219
 PERCENTILE_CONT 関数 220
 PERCENTILE_DISC 関数 223
 PI 関数 225
 POWER 関数 225
 PRINT コマンド
 Transact-SQL 639
 PROPERTY_DESCRIPTION 関数 226
 PROPERTY_NAME 関数 227
 PROPERTY_NUMBER 関数 228
 PROPERTY 関数 225

Q

QUARTER 関数 228
 QUOTED_IDENTIFIER オプション 33

R

RADIANS 関数 229
 RAISERROR 文
 ON EXCEPTION RESUME 643
 Transact-SQL 643

索引

- RAND 関数 229
 - RANK 関数 230
 - REGR_AVGX 関数 231
 - REGR_AVGY 関数 232
 - REGR_COUNT 関数 234
 - REGR_INTERCEPT 関数 234
 - REGR_R2 関数 236
 - REGR_SLOPE 関数 237
 - REGR_SXX 関数 238
 - REGR_SXY 関数 239
 - REGR_SYY 関数 240
 - REMAINDER 関数 241
 - REPEAT 関数 242
 - REPLACE 関数 243
 - SELECT INTO 文 28, 142, 190, 196, 197, 201, 202, 242, 244, 246, 248, 269, 328, 329
 - REPLICATE 関数 244
 - request_level_debugging 説明 503
 - resetclocks
 - sp_iqcheckdb オプション 360
 - REVERSE 関数
 - SQL 構文 245
 - RIGHT 関数 245
 - ROLLUP 処理
 - GROUPING 関数 178
 - ROUND 関数 246
 - ROWID 関数 247
 - RTRIM 関数 248
- S**
- sa_audit_string システム・プロシージャ 491
 - sa_checkpoint_execute システム・プロシージャ 491
 - sa_conn_activity システム・プロシージャ
 - 構文 492
 - sa_conn_info システム・プロシージャ 493
 - sa_conn_properties システム・プロシージャ 494
 - sa_db_info システム・プロシージャ 495
 - sa_db_properties システム・プロシージャ 496
 - sa_dependent_views system プロシージャ 352
 - a_disable_auditing_type システム・プロシージャ 500
 - sa_enable_auditing_type システム・プロシージャ 497
 - sa_eng_properties システム・プロシージャ 498
 - sa_flush_cache システム・プロシージャ 500
 - sa_make_object system システム・プロシージャ 501
 - sa_rowgenerator システム・プロシージャ
 - 構文 502
 - sa_server_option システム・プロシージャ 503
 - sa_set_http_header システム・プロシージャ 510
 - sa_set_http_option システム・プロシージャ 511
 - sa_table_page_usage システム・プロシージャ 499
 - sa_validate システム・プロシージャ
 - 構文 511
 - sa_verify_password システム・プロシージャ 352
 - SACHARSET 環境変数 12
 - SALANG 環境変数 13
 - SECONDS 関数 249
 - SECOND 関数 249
 - SELECT INTO
 - REPLACE 関数の使用 28, 142, 190, 196, 197, 201, 202, 242, 244, 246, 248, 269, 328, 329
 - Transact-SQL 635
 - SELECT 文
 - Transact-SQL 626
 - 例 527
 - SET OPTION 文
 - Transact-SQL 625
 - SHARED 構文
 - Transact-SQL 631
 - SIGNAL 文
 - Transact-SQL 643
 - SIGN 関数 250
 - SIMILAR 関数 250
 - SIN 関数 251
 - SMALLDATETIME データ型 83
 - SMALLINT データ型 74
 - SMALLMONEY データ型 77
 - SOME 条件 40
 - SORTKEY 関数 251
 - SYSPARTITION システム・ビュー 569
 - SOUNDEX 関数 256
 - sp_expireallpasswords システム・プロシージャ 352
 - sp_iq_reset_identity システム・プロシージャ 453
 - sp_iqaddlogin システム・プロシージャ 353
 - sp_iqbackupdetails ストアド・プロシージャ 354
 - sp_iqbackupsummary ストアド・プロシージャ 356
 - sp_iqbreaction ストアド・プロシージャ 453

- sp_iqcheckdb
 - allocation モード 359
 - check モード 361
 - DBCC_LOG_PROGRESS オプション 364
 - dropleaks モード 362
 - resetclocks オプション 360
 - verify モード 361
 - 構文 357
 - 実行時間 363
 - 出力 364
 - 出力例 364
 - パフォーマンス 363
- sp_iqcheckdb システム・プロシージャ 357
- sp_iqcheckoptions システム・プロシージャ 465
- sp_iqcolumnuse システム・プロシージャ 371
- sp_iqcolumn システム・プロシージャ 369
- sp_iqconnection システム・プロシージャ 372
- sp_iqcontext システム・プロシージャ 377
- sp_iqcopyloginpolicy システム・プロシージャ 379, 435
- sp_iqcursorinfo システム・プロシージャ 380
- sp_iqdatatype システム・プロシージャ 382
- sp_iqdbsize システム・プロシージャ 385
- sp_iqdbspaceinfo システム・プロシージャ 389
- sp_iqdbspaceobjectinfo システム・プロシージャ 393
- sp_iqdbspace システム・プロシージャ 387
- sp_iqdbstatistics システム・プロシージャ 396
- sp_iqdroplogin システム・プロシージャ 398
- sp_iqemptyfile システム・プロシージャ 398
- sp_iqestdbspaces システム・プロシージャ 401
- sp_iqestjoin システム・プロシージャ 399
- sp_iqestspace システム・プロシージャ 403
- sp_iqevent システム・プロシージャ 403
- sp_iqfile システム・プロシージャ 406
- sp_iqhelp システム・プロシージャ 408
- sp_iqindex_alt システム・プロシージャ 415
- sp_iqindexadvice システム・プロシージャ 418
- sp_iqindexfragmentation システム・プロシージャ 419
- sp_iqindexinfo
 - インデックス情報の表示 422, 423
- sp_iqindexinfo システム・プロシージャ 421
- sp_iqindexmetadata システム・プロシージャ 423
- sp_iqindexsize システム・プロシージャ 424
- sp_iqindexuse システム・プロシージャ 426
- sp_iqindex システム・プロシージャ 415
- sp_iqjoinindexsize システム・プロシージャ 430
- sp_iqjoinindex システム・プロシージャ 427
- sp_iqlocks システム・プロシージャ 432
- sp_iqmodifylogin 436
- sp_iqmodifylogin システム・プロシージャ 436
- sp_iqobjectinfo システム・プロシージャ 436
- sp_iqpassword システム・プロシージャ 439
- sp_iqpkkeys システム・プロシージャ 440
- sp_iqprocedure システム・プロシージャ 442
- sp_iqprocparm システム・プロシージャ 445
- sp_iqrebuildindex システム・プロシージャ 448, 454
- sp_iqrename システム・プロシージャ 451
- sp_iqsetcompression システム・プロシージャ 350
- sp_iqshowcompression システム・プロシージャ 350
- sp_iqshowpsexec システム・プロシージャ 456
- sp_iqspaceinfo システム・プロシージャ 458
 - 出力例 458
- sp_iqspaceused システム・プロシージャ 459
- sp_iqstatistics システム・プロシージャ 460
- sp_iqstatus システム・プロシージャ 462
 - 出力例 463
- sp_iqsysmon システム・プロシージャ 465
- sp_iqtablesize システム・プロシージャ 473
- sp_iqtableuse システム・プロシージャ 475
- sp_iqtable システム・プロシージャ 471
- sp_iqtransaction システム・プロシージャ 476
- sp_iqunusedcolumn システム・プロシージャ 479
- sp_iqunusedindex システム・プロシージャ 481
- sp_iqunusedtable システム・プロシージャ 482
- sp_iqversionuse システム・プロシージャ 483
- sp_iqview システム・プロシージャ 484
- sp_iqwho システム・プロシージャ 486
- sp_iqworkmon システム・プロシージャ 489
- sp_login_environment システム・プロシージャ 513
- sp_remote_columns システム・プロシージャ 513
- sp_remote_exported_keys システム・プロシージャ 514, 515
- sp_remote_primary_keys システム・プロシージャ
 - 構文 516
- sp_remote_tables システム・プロシージャ 517
- sp_servercaps システム・プロシージャ 518

索引

- sp_tsql_environment システム・プロシージャ 520
 - SPACE 関数 257
 - SQL
 - IQ 言語の違い 341
 - ユーザ定義関数 123
 - SQL Anywhere 602
 - 管理者の役割 607
 - 参照整合性制約 618
 - マニュアル xxi
 - SQL92 準拠 341
 - SQLCODE
 - 特別値 56
 - SQLCONNECT 環境変数 13
 - SQLSTATE
 - 特別値 56
 - SQL 関数
 - ERRORMSG 関数の構文 167
 - GRAPHICAL_PLAN 関数の構文 175
 - GROUP_MEMBER 関数の構文 178
 - GROUPING 関数の構文 178
 - HTML_PLAN 関数の構文 184
 - ISDATE 関数の構文 192
 - ISNUMERIC 関数の構文 193
 - LEN 関数の構文 197
 - NEWID 関数の構文 208
 - REVERSE 関数の構文 245
 - STR_REPLACE 関数の構文 262
 - STRTOUUID 関数の構文 264
 - UUIDTOSTR 関数の構文 330
 - 互換性 631
 - SQL 構文
 - CURRENT DATABASE 特別値 54
 - CURRENT PUBLISHER 特別値 54
 - CURRENT USER 特別値 55
 - LAST USER 特別値 55
 - TIMESTAMP 特別値 56
 - USER 特別値 57
 - 識別子 22
 - SQL 文字列の区切り 22
 - SQRT 関数 257
 - SQUARE 関数 257
 - STDDEV_POP 関数 259
 - STDDEV_SAMP 関数 260
 - STDDEV 関数 258
 - STR_REPLACE 関数
 - SQL 構文 262
 - STRING 関数 263
 - STRTOUUID 関数
 - SQL 構文 264
 - STR 関数 261
 - STUFF 関数 265
 - SUBSTRING 関数 265
 - SUBSTR 関数 265
 - SUM 関数 266
 - SUSER_ID 関数 267
 - SUSER_NAME 関数 267
 - Sybase IQ ユーザ管理
 - sp_iqdroplogin 398
 - SYBASE_JRE 環境変数 14
 - SYBASE_OCS 環境変数 15
 - SYBASE 環境変数 14
 - SYSIQBACKUPHISTORYDETAIL システム・ビュー 549
 - SYSIQBACKUPHISTORY システム・ビュー 548
 - SYSIQDBFILE システム・ビュー 551
 - SYSIQDBSPACE システム・ビュー 552
 - SYSIQIDX システム・ビュー 553
 - SYSIQJOINIDX システム・ビュー 556
 - SYSIQJOINIXCOLUMN システム・ビュー 557
 - SYSIQJOINIXTABLE システム・ビュー 559
 - SYSIQPARTITIONCOLUMN システム・ビュー 560
 - SYSIQTAB システム・ビュー 561
 - SYSPARTITIONKEY システム・ビュー 570
 - SYSPARTITIONSCHEME システム・ビュー 570
 - SYSSUBPARTITIONKEY システム・ビュー 579
- ## T
- TAN 関数 268
 - temp_cache_memory_mb
 - sa_server_option パラメータ 510
 - TEXT データ型 70, 609
 - 互換性 609, 613
 - THEN
 - IF 式 30
 - TIMESTAMP
 - 式の変換 160

データ型 84
 データ型の互換性 611, 612
 特別値 56
 TIME データ型 83
 TINYINT データ型 74
 TODAY 関数 268, 600
 Transact-SQL
 移植可能な SQL を書 625
 外部ジョイン演算子 29
 概要 602
 結果セット 640
 互換性のあるデータベースの作成 614
 参照整合性制約 618
 式 32
 システム・カタログ 591
 ジョイン 629
 説明 601
 定数 32
 バッチ 637
 比較条件 36
 ビット処理演算子 28
 プロシージャ 636
 プロシージャ言語の概要 636
 変数 641
 文字列 33
 ユーザ定義データ型 90
 ローカル変数 59
 Transact-SQL との互換性
 データベース 615
 TRIM 関数 268
 TRUNCNUM 関数 269
 TS_ARMA_AR 関数 270
 TS_ARMA_CONST 関数 273
 TS_ARMA_MA 関数 277
 TS_AUTO_UNI_AR 関数 284
 TS_AUTOCORRELATION 関数 280
 TS_BOX_COX_XFORM 関数 287
 TS_DIFFERENCE 関数 290
 TS_ESTIMATE_MISSING 関数 294
 TS_LACK OF FIT_P 関数 300
 TS_LACK OF FIT 関数 297
 TS_MAX_ARMA_AR 関数 304
 TS_MAX_ARMA_CONST 関数 308
 TS_MAX_ARMA_LIKELIHOOD 関数 311

TS_MAX_ARMA_MA 関数 315
 TS_OUTLIER_IDENTIFICATION 関数 318
 TS_PARTIAL_AUTOCORRELATION 関数 323
 TS_VWAP 関数 326

U

UCASE 関数 328
 UNION
 サブクエリ内 627
 UNIQUEIDENTIFIERSTR データ型
 説明 69
 UNIQUEIDENTIFIER データ型 81
 Universally Unique Identifier
 NEWID 関数の SQL 構文 208
 UPPER 関数 329
 USER
 特殊定数 600
 特別値 57
 USER_ID 関数 329
 USER_NAME 関数 330
 UUID
 NEWID 関数の SQL 構文 208
 STRTOUUID 関数の SQL 構文 264
 UUIDTOSTR 関数の SQL 構文 330
 UUIDTOSTR 関数
 SQL 構文 330

V

VAR_POP 関数 331
 VAR_SAMP 関数 332
 VARBINARY データ型 78
 VARCHAR データ型
 後続ブランクの削除 71, 609
 説明 69, 70
 VARIANCE 関数 333

W

WEEKS 関数 334
 WEIGHTED_AVG 関数 335

索引

WHERE 句

Transact-SQL 628

WIDTH_BUCKET 関数 337

WITHIN GROUP 句 108

X

xp_cmdshell システム・プロシージャ
構文 525

Y

YEARS 関数 339

YEAR 関数 338

YMD 関数 340

あ

アーキテクチャ

Adaptive Server 604

アーク・コサイン 128

アーク・サイン 130

アーク・タンジェント 130

アーク・タンジェント率 131

値の対数 200

アドバイス

格納 418

クリア 418

表示 418

アポストロフィ

文字列中 24

アルファベット文字

定義 22

アロケーション・マップ

リセット 362

い

移植可能な SQL 625

一貫性の検査

パーティション 358

イベント

EVENT_CONDITION_NAME 関数 169

EVENT_CONDITION 関数 167

EVENT_PARAMETER 関数 169

情報の表示 403, 408

インストール・ディレクトリ

説明 2

インデックス

Adaptive Server Enterprise 621

IQ 621

SQL Anywhere 621

Transact-SQL 616

システム・ビュー 547

インデックス設定ヒント 50

引用符

SQL 識別子 22

データベース・オブジェクト 22

文字列 24

う

ウィンドウ関数

ウィンドウ関数の種類 107

ウィンドウ・パーティション 107

ウィンドウ名または指定 107

ウィンドウ関数、定義 107

ウィンドウ指定 107

ウィンドウ集合関数 106

ウィンドウの種類 107

ウィンドウの定義 107

ウィンドウ名 107

え

エラー

Transact-SQL 641, 643

エラー処理

IMSL ライブラリ 126

エラー・メッセージ

ERRORMSG 関数 167

テキストの取得 167

エラー・ロギング
 IMSL ライブラリ 127
 演算子
 比較演算子 35
 優先度 30

お

大文字と小文字の区別
 Transact-SQL との互換性 615
 識別子 615
 データ 615
 データベース 615
 パスワード 616
 パターン・マッチング 43
 比較条件 35
 ユーザ ID 616
 ユーザ定義データ型 615
 オブジェクト
 ID の調査 216
 情報の表示 408
 定義 501
 名前の調査 216
 名前の変更 451
 オブジェクト名の変更
 sp_iqrename プロシージャ 451
 オプション
 DBCC_LOG_PROGRESS 364
 FLOAT_AS_DOUBLE 613
 QUOTED_IDENTIFIER 33
 SQL Anywhere 645
 システム・ビュー 568, 588
 オプティマイザ
 見積もり 48
 ユーザ定義の選択性 48

か

カーソル
 IQ のロー・レベル 638
 Transact-SQL 638
 情報の表示 380

概数値データ型
 互換性 613
 外部キー
 システム・ビュー 543, 544
 外部ジョイン
 Transact-SQL 629
 演算子 29
 サブクエリ 26
 ネスト 629
 連鎖 629
 概要 354
 角カッコ
 SQL 識別子 22
 データベース・オブジェクト 22
 確認
 インデックス 361
 キー 361
 パスワード 352, 513
 分割されたテーブル 361
 型
 データ型について 69
 型変換 90
 カタログ
 Adaptive Server Enterprise の互換性 606
 システム・テーブル 597
 カタログ・ストア
 IQ 606
 検証 511
 モニタリング 465
 カッコ
 SQL 識別子 22
 データベース・オブジェクト 22
 活性タイムアウト
 データベース・サーバ 503
 カラム
 SYSCOLUMNS システム・ビュー 537
 命名 26
 ユーザ定義データ型 90
 カラム長 138
 カラムのデフォルト
 サポートなし 619
 カラム名 139
 カレント・ユーザ
 環境設定 15

索引

環境変数

IQLOGDIR15 9
IQPORT 9
IQTIMEOUT 10
PATH 12
SQLCONNECT 13
SYBASE 14
SYBASE_JRE 14
SYBASE_OCS 15
説明 6

監査

コメントの追加 491
無効化 500
有効化 497

関数 103

ABS 関数 128
ACOS 関数 128
Adaptive Server Enterprise システム関数 120
ARGN 関数 129
ASCII 関数 129
ASIN 関数 130
ATAN2 関数 131
ATAN 関数 130
AVG 関数 131
BIGINTTOHEX 関数 132
BIT_LENGTH 関数 133
BYTE_LENGTH 関数 133
CAST 関数 134
CEILING 関数 135
CEIL 関数 135
CHAR_LENGTH 関数 136
CHARINDEX 関数 137
CHAR 関数 136
COALESCE 関数 138
COL_LENGTH 関数 138
COL_NAME 関数 139
CONNECTION_PROPERTY 関数 139
CONVERT 関数 140
CORR 関数 142
COS 関数 143
COT 関数 144
COUNT 関数 146
COVAR_POP 関数 144

COVAR_SAMP 関数 145
CUME_DIST 関数 147
DATALENGTH 関数 148
DATEADD 関数 149
DATECEILING 関数 150
DATEDIFF 関数 152
DATEFLOOR 関数 154
DATEFORMAT 関数 156
DATENAME 関数 157
DATEPART 関数 157
DATEROUND 関数 158
DATETIME 関数 160
DATE 関数 149
DAYNAME 関数 160
DAYS 関数 161
DAY 関数 160
DB_ID 関数 162
DB_NAME 関数 162
DB_PROPERTY 関数 163
DEGREES 関数 164
DENSE_RANK 関数 164
DIFFERENCE 関数 165
DOW 関数 166
ERRORMSG 関数の SQL 構文 167
EVENT_CONDITION_NAME 関数 169
EVENT_CONDITION 関数 167
EVENT_PARAMETER 関数 169
EXP_WEIGHTED_AVG 関数 171
EXP 関数 170
FIRST_VALUE 関数 172
FLOOR 関数 174
GETDATE 関数 175
GRAPHICAL_PLAN 175
GROUP_MEMBER 関数の SQL 構文 178
GROUPING 関数の SQL 構文 178
HEXTOBIGINT 関数 179
HEXTOINT 関数 180
HOURS 関数 181
HOUR 関数 181
HTML_DECODE 関数 182
HTML_ENCODE 関数 183
HTML_PLAN 関数 184

HTTP 115
HTTP_DECODE 関数 186
HTTP_ENCODE 関数 186
HTTP_HEADER 関数 187
HTTP_VARIABLE 関数 188
IFNULL 関数 189
INDEX_COL 関数 189
INSERTSR 関数 190
INTTOHEX 関数 191
IQ の拡張機能 344
ISDATE 関数の SQL 構文 192
ISNULL 関数 193
ISNUMERIC 関数の SQL 構文 193
LAST_VALUE 194
LCASE 関数 196
LEFT 関数 196
LENGTH 関数 198
LEN 関数の SQL 構文 197
LOCATE 関数 199
LOG10 関数 201
LOG 関数 200
LOWER 関数 201
LTRIM 関数 202
MAX 関数 202
MEDIAN 関数 203
MINUTES 関数 205
MINUTE 関数 205
MIN 関数 204
MOD 関数 206
MONTHNAME 関数 206
MONTHS 関数 207
MONTH 関数 206
NEWID 関数の SQL 構文 208
NEXT_CONNECTION 関数 209
NEXT_DATABASE 関数 210
NEXT_HTTP_HEADER 関数 211
NEXT_HTTP_VARIABLE 関数 211
NOW 関数 212
NTILE 関数 212
NULLIF 関数 214
NUMBER 関数 215
OBJECT_ID 関数 216
OBJECT_NAME 関数 216
OCTET_LENGTH 関数 217
PATINDEX 関数 217
PERCENT_RANK 関数 219
PERCENTILE_CONT 関数 220
PERCENTILE_DISC 関数 223
PI 関数 225
POWER 関数 225
PROPERTY_DESCRIPTION 関数 226
PROPERTY_NAME 関数 227
PROPERTY_NUMBER 関数 228
PROPERTY 関数 225
QUARTER 関数 228
RADIANS 関数 229
RAND 関数 229
RANK 関数 230
REGR_AVGX 関数 231
REGR_AVGY 関数 232
REGR_COUNT 関数 234
REGR_INTERCEPT 関数 234
REGR_R2 関数 236
REGR_SLOPE 関数 237
REGR_SXX 関数 238
REGR_SXY 関数 239
REGR_SYY 関数 240
REMAINDER 関数 241
REPEAT 関数 242
REPLACE 関数 243
REPLICATE 関数 244
REVERSE 関数の SQL 構文 245
RIGHT 関数 245
ROUND 関数 246
ROWID 関数 247
RTRIM 関数 248
SECONDS 関数 249
SECOND 関数 249
SIGN 関数 250
SIMILAR 関数 250
SIN 関数 251
SORTKEY 関数 251
SOUNDEX 関数 256
SPACE 関数 257

- SQRT 関数 257
 - SQUARE 関数 257
 - STDDEV_POP 関数 259
 - STDDEV_SAMP 関数 260
 - STDDEV 関数 258
 - STR_REPLACE 関数の SQL 構文 262
 - STRING 関数 263
 - STRTOUUID 関数の SQL 構文 264
 - STR 関数 261
 - STUFF 関数 265
 - SUBSTRING 関数 265
 - SUBSTR 関数 265
 - SUM 関数 266
 - SUSER_ID 関数 267
 - SUSER_NAME 関数 267
 - TAN 関数 268
 - today 600
 - TODAY 関数 268
 - Transact-SQL 631
 - TRIM 関数 268
 - TRUNCNUM 関数 269
 - TS_ARMA_AR 関数 270
 - TS_ARMA_CONST 関数 273
 - TS_ARMA_MA 関数 277
 - TS_AUTO_UNI_AR 関数 284
 - TS_AUTOCORRELATION 関数 280
 - TS_BOX_COX_XFORM 関数 287
 - TS_DIFFERENCE 関数 290
 - TS_ESTIMATE_MISSING 関数 294
 - TS_LACK OF FIT_P 関数 300
 - TS_LACK OF FIT 関数 297
 - TS_MAX_ARMA_AR 関数 304
 - TS_MAX_ARMA_CONST 関数 308
 - TS_MAX_ARMA_LIKELIHOOD 関数 311
 - TS_MAX_ARMA_MA 関数 315
 - TS_OUTLIER_IDENTIFICATION 関数 318
 - TS_PARTIAL_AUTOCORRELATION 関数 323
 - TS_VWAP 関数 326
 - UCASE 関数 328
 - UPPER 関数 329
 - USER_ID 関数 329
 - USER_NAME 関数 330
 - UUIDTOSTR 関数の SQL 構文 330
 - VAR_POP 関数 331
 - VAR_SAMP 関数 332
 - VARIANCE 関数 333
 - WEEKS 関数 334
 - WEIGHTED_AVG 関数 335
 - WIDTH_BUCKET 関数 337
 - YEARS 関数 339
 - YEAR 関数 338
 - YMD 関数 340
 - アルファベット順のリスト 128
 - ウィンドウ集合 106
 - 結果の一貫性 115
 - 時系列 124
 - 集合 104
 - 数値 107, 115
 - その他 127
 - データ型変換 110
 - 統計 107
 - 日付および時刻 111
 - 分散統計 107
 - 分析 106
 - 文字列 117
 - ユーザ定義 123
 - ランク付け 106
 - 関数、集合
 - GROUPING 178
 - 関数、その他
 - ERRORMSG 167
 - ISNUMERIC 193
 - NEWID 208
 - 関数、データ型変換
 - ISDATE 192
 - 関数、文字列 184, 197, 245, 262
 - STRTOUUID 264
 - UUIDTOSTR 330
- き**
- キー
- 確認 361
 - 情報の表示 440

キーワード
 SQL 19
 リスト 20
 キャッシュ
 テンポラリ 510
 フラッシュ 500
 メイン 506

く

句
 ON EXCEPTION RESUME 643

クエリ
 Transact-SQL 626

クエリ・サーバ
 同期 10

グループ
 Adaptive Server Enterprise 622

グローバル変数
 互換性 62
 説明 58, 60
 リスト 61

け

計算カラム
 サポートなし 620

結果セット
 Transact-SQL 640

言語
 指定 13

検査制約 617
 Transact-SQL との互換性 617
 適用 617

検証
 カタログ・ストア 511

こ

後続ブランク
 削除 71, 609
 後続ブランクの削除 71, 609

構文
 CURRENT DATABASE 特別値 54
 CURRENT PUBLISHER 特別値 54
 CURRENT USER 特別値 55
 LAST USER 特別値 55
 SQL 識別子 22
 TIMESTAMP 特別値 56
 USER 特別値 57

コード・ページ
 データ記憶領域 71

互換性
 Adaptive Server Enterprise 601
 概数値データ 613
 参照整合性制約 618

コメント
 コメント・インジケータ 65

さ

サーバ
 プロパティ 122

サーバ管理
 SQL Anywhere と IQ 644

サービス
 レジストリ・エントリ 16

作成
 データ型 89

サブクエリ
 Adaptive Server Enterprise 627
 IQ 627
 IQ の実装 343
 SQL Anywhere 627
 式内 26
 検索条件内 37
 分離 37

サブクエリ、分離 38

サブクエリ述部の分離 37

算術式 27
 日付 635

参照整合性制約
 CASCADE がサポートされない 618
 互換性 618

し

- 式 25
 - CASE 31
 - Transact-SQL 32
 - タイムスタンプへの変換 160
 - 長さ(バイト数) 148
- 式サブクエリ
 - IF 文内 638
- 識別子
 - SQL Anywhere での最大長 22
 - SQL 構文 22
 - 大文字と小文字の区別 615
 - 説明 22
 - ユニーク性 616
- 時系列関数 124, 125
 - IMSL ライブラリ 125
 - エラー処理 126
 - エラー・ロギング 127
- 時刻
 - クエリ 86
- 指数関数 170
- システム・カタログ 534
 - Adaptive Server Enterprise の互換性 606
 - Transact-SQL 591
- システム関数 119
 - COL_LENGTH 138
 - COL_NAME 139
 - CONNECTION_PROPERTY 139
 - DATALENGTH 148
 - DB_ID 162
 - DB_NAME 162
 - DB_PROPERTY 163
 - EVENT_CONDITION 167
 - EVENT_CONDITION_NAME 169
 - EVENT_PARAMETER 169
 - INDEX_COL 189
 - NEXT_CONNECTION 209
 - NEXT_DATABASE 210
 - OBJECT_ID 216
 - OBJECT_NAME 216
 - PROPERTY 225
 - PROPERTY_DESCRIPTION 226
 - PROPERTY_NAME 227
 - PROPERTY_NUMBER 228
 - SUSER_ID 267
 - SUSER_NAME 267
 - Transact-SQL 634
 - USER_ID 329
 - USER_NAME 330
- システム管理者
 - Adaptive Server Enterprise 607
- システム・セキュリティ担当者
 - Adaptive Server Enterprise 607
- システム・テーブル
 - Adaptive Server Enterprise の互換性 606
 - DUMMY 600
 - 情報の表示 408
 - 説明 597
- システム・ビュー
 - SYSIQCOLUMN 551
 - SYSIQFILE 553
 - SYSIQJOININDEX 557
 - SYSIQMPXLOGINPOLICYOPTION 559
 - SYSPARTITIONKEY 570
 - SYSCAPABILITYESSYSCAPABILITIES 533
 - Adaptive Server Enterprise 648
 - SYSARTICLE 532
 - SYSARTICLECOL 532
 - SYSARTICLECOLS 532
 - SYSARTICLES 533
 - SYSCAPABILITY 533
 - SYSCAPABILITYNAME 534
 - SYSCATALOG 534
 - SYSCHECK 534
 - SYSCOLAUTH 535
 - SYSCOLLATION 535
 - SYSCOLLATIONMAPPINGS 536
 - SYSCOLPERM 535
 - SYSCOLSTAT 536
 - SYSCOLSTATS 537
 - SYSCOLUMN 537
 - SYSCOLUMNS 537
 - SYSCOLUMNS ASE 互換ビュー 538
 - SYSCONSTRAINT 538
 - SYSDBFILE 539
 - SYSDBSPACE 539
 - SYSDBSPACEPERM 539
 - SYSDEPENDENCY 540
 - SYSDOMAIN 540
 - SYSEVENT 541

SYSEVENTTYPE 541
 SYSEXTERNALLOGIN 542
 SYSEXTERNENV 541
 SYSEXTERNENVOBJECT 542
 SYSFILE 543
 SYSFKCOL 543
 SYSFKEY 543
 SYSFORIGNKEY 544
 SYSFORIGNKEYS 544
 SYSGROUP 544
 SYSGROUPS 545
 SYSHISTORY 545
 SYSIDX 546
 SYSIDXCOLUMN 546
 SYSINDEX 546
 SYSINDEXES 547
 SYSINDEXES ASE 互換ビュー 547
 SYSINFO 548
 SYSIQBACKUPHISTORY 548
 SYSIQBACKUPHISTORYDETAIL 549
 SYSIQDBFILE 551
 SYSIQDBSPACE 552
 SYSIQIDX 553
 SYSIQINFO 555
 SYSIQITAB 561
 SYSIQJOINIDX 556
 SYSIQJOINIXCOLUMN 557
 SYSIQJOINIXTABLE 559
 SYSIQMPXSERVER 560
 SYSIQOBJECTS ASE 互換ビュー 560
 SYSIQPARTITIONCOLUMN 560
 SYSIQTABCOL 562
 SYSIQTABLE システム・ビュー 563
 SYSIQVINDEXTAB ASE 互換ビュー 563
 SYSIXCOL 互換ビュー 564
 SYSJAR 564
 SYSJARCOMPONENT 564
 SYSJAVACLASS 565
 SYSLOGINMAP 565
 SYSLOGINPOLICY 565
 SYSLOGINPOLICYOPTION 566
 SYSLOGINS ASE 互換ビュー 566
 SYSMVOPTION 566
 SYSOBJECTS ASE 互換ビュー 567
 SYSOPTIONS 568
 SYSPARTITION 569
 SYSPARTITIONSCHEME 570
 SYSPHYSIDX 571
 SYSPROCAUTH 571
 SYSPROCEDURE 572
 SYSPROCPARMS 572
 SYSPROCPERM 573
 SYSPROCS 573
 SYSPROXYTAB 573
 SYSPUBLICATION 574
 SYSPUBLICATIONS 574
 SYSREMARK 574
 SYSREMOTEOPTION 575
 SYSREMOTEOPTION2 575
 SYSREMOTEOPTIONS 575
 SYSREMOTEOPTIONTYPE 576
 SYSREMOTETYPE 576
 SYSREMOTETYPES 576
 SYSREMOTEUSER 577
 SYSREMOTEUSERS 577
 SYSSCHEDULE 578
 SYSSOURCE 578
 SYSSQLSERVERTYPE 579
 SYSSUBPARTITIONKEY 579
 SYSSUBSCRIPTION 579
 SYSSUBSCRIPTIONS 579
 SYSSYNC 580
 SYSSYNCPROFILE 581
 SYSSYNCS 581
 SYSSYNCSUBSCRIPTIONS 582
 SYSSYNCSUSERS 582
 SYSTAB 583
 SYSTABAUTH 583
 SYSTABCOL 584
 SYSTABLE 583
 SYSTABLEPERM 584
 SYSTEXTCONFIG 584
 SYSTEXTIDX 585
 SYSTEXTIDXTAB 585
 SYSTRIGGER 585
 SYSTRIGGERS 586
 SYSTYPEMAP 586
 SYSTYPES ASE 互換ビュー 586
 SYSUSER 587
 SYSUSERAUTH 587
 SYSUSERAUTHORITY 587
 SYSUSERLIST 588

索引

- SYSUSERMESSAGE 588
- SYSUSEROPTIONS 588
- SYSUSERPERM 589
- SYSUSERPERMS 589
- SYSUSERS ASE 互換ビュー 590
- SYSUSERTYPE 589
- SYSVIEW 590
- SYSVIEWS 590
- SYSWEBSERVICE 591
- 統合 527
- システム・プロシージャ
 - sa_audit_string 491
 - sa_checkpoint_execute 491
 - sa_conn_activity 492
 - sa_conn_info 493
 - sa_conn_properties 494
 - sa_db_info 495
 - sa_db_properties 496
 - sa_dependent_views 352
 - sa_disable_auditing_type 500
 - sa_enable_auditing_type 497
 - sa_eng_properties 498
 - sa_flush_cache 500
 - sa_make_object 501
 - sa_rowgenerator 502
 - sa_server_option 503
 - sa_set_http_header 510
 - sa_set_http_option 511
 - sa_table_page_usage 499
 - sa_validate 511
 - sa_verify_password 352, 513
- sp_expireallpasswords 352
- sp_iqaddlogin 353
- sp_iqbackupdetails 354
- sp_iqbackupsummary 356
- sp_iqcheckdb 357
- sp_iqcheckoptions 365
- sp_iqclient_lookup 367
- sp_iqcolumn 369
- sp_iqcolumnuse 371
- sp_iqconnection 372
- sp_iqcontext 377
- sp_iqcopyloginpolicy 379, 435
- sp_iqcursorsinfo 380
- sp_iqdatatype 382
- sp_iqdbsize 385
- sp_iqdbspaceobjectinfo 393
- sp_iqdbstatistics 396
- sp_iqdroplogin 398
- sp_iqemptyfile 398
- sp_iquestdbspaces 401
- sp_iquestjoin 399
- sp_iquestspace 403
- sp_iqevent 403
- sp_iqfile 406
- sp_iqhelp 408
- sp_iqindex 415
- sp_iqindex_alt 415
- sp_iqindexadvice 418
- sp_iqindexsize 424
- sp_iqindexuse 426
- sp_iqjoinindex 427
- sp_iqjoinindexsize 430
- sp_iqmodifylogin 436
- sp_iqobjectinfo 436
- sp_iqpassword 439
- sp_iqpkeys 440
- sp_iqprocedure 442
- sp_iqprocparm 445
- sp_iqrename 451
- sp_iqrestoreaction 453
- sp_iqsetcompression 350
- sp_iqshowcompression 350
- sp_iqshowpsex 456
- sp_iqspaceinfo 458
- sp_iqspaceused 459
- sp_iqstatistics 460
- sp_iqstatus 462
- sp_iqsysmon 465
- sp_iqtable 471
- sp_iqtablesize 473
- sp_iqtableuse 475
- sp_iqtransaction 476
- sp_iqunusedcolumn 479
- sp_iqunusedindex 481
- sp_iqunusedtable 482
- sp_iqversionuse 483
- sp_iqview 484
- sp_iqwho 486
- sp_iqworkmon 489
- sp_login_environment 513
- sp_remote_columns 513

sp_remote_exported_keys 514, 515
 sp_remote_primary_keys 516
 sp_remote_tables 517
 sp_servercaps 518
 sp_tsql_environment 520
 xp_cmdshell 525
 情報の表示 408
 説明 349
 システム変数 60
 システム呼び出し
 xp_cmdshell システム・プロシージャ 525
 ストアド・プロシージャ 525
 実行フェーズ・ヒント 51
 自動ジョイン
 外部キー 342
 集合関数 104
 AVG 131
 COUNT 146
 MAX 202
 MIN 204
 STDDEV 258
 STDDEV_POP 259
 STDDEV_SAMP 260
 SUM 266
 VAR_POP 331
 VAR_SAMP 332
 VARIANCE 333
 終了時刻
 データベース・サーバ 503
 述部
 説明 34
 分離 37
 ジョイン
 Transact-SQL 629
 外部演算子 29
 自動 343
 ジョイン・インデックス
 情報の表示 427
 テーブル数
 クエリ
 テーブル数/ブロック 347
 ジョイン演算子
 ANSI 629
 Transact-SQL 629
 ジョイン等号条件 53

ジョイン等号条件に関するユーザ指定の
 ヒント 53
 条件
 クエリのユーザ指定の条件 48
 条件ヒント文字列 49
 照合
 SORTKEY 関数 251

す

数字 26
 数値関数 107, 115
 ABS 128
 ACOS 128
 ASIN 130
 ATAN 130
 ATAN2 131
 CEIL 135
 CEILING 135
 COS 143
 COT 144
 DEGREES 164
 EXP 170
 FLOOR 174
 LOG 200
 LOG10 201
 MOD 206
 PI 225
 POWER 225
 RADIANS 229
 RAND 229
 REMAINDER 241
 ROUND 246
 SIGN 250
 SIN 251
 SQRT 257
 SQUARE 257
 TAN 268
 TRUNCNUM 269
 WIDTH_BUCKET 337
 結果の一貫性 115
 数値式 27
 数値データ型
 互換性 612, 613

索引

ストアド・プロシージャ
Adaptive Server Enterprise 648
sa_dependent_views 352
sa_rowgenerator 502
sa_verify_password 352, 513
sp_iqbackupdetails 354
sp_iqbackupsummary 356
sp_iqclient_lookup 367
sp_iqrestoreaction 453
SQL Anywhere 647
xp_cmdshell 525
ストアド・プロシージャ言語
概要 636
スレッド
dbcc 359

せ

正接 268
接続
ID 番号の調査 209
IMSL ライブラリ 125
数 372
情報の表示 486
プロパティ 121
接続情報
sp_iqcontext 377
接続数
決定 372
接続プロパティ値 139
接続レベル変数
説明 59
絶対値 128
選択性
ヒント 50
明示的 48
ユーザ指定の条件 48

そ

挿入
SQL Anywhere 646
その他の関数 127
ARGN 129

COALESCE 138
IFNULL 189
ISNULL 193
NULLIF 214
NUMBER 215
ROWID 247

た

対数 (底が 10) 201
ダミーの IQ テーブル 104
一貫性のある結果を取得 113
検索条件
3 値的論理 47
ALL または ANY 条件 40
BETWEEN 条件 41
CONTAINS 条件 45
EXISTS 条件 45
IN 条件 44
IS NULL 条件 46
LEADING SUBSTRING SEARCH 条件 42
LIKE 条件 42
NOT 条件 47
サブクエリ 37
真理値条件 47
説明 34
比較条件 35

て

定数
Transact-SQL 32
式内 26
ディレクトリ構造 2
データ
大文字と小文字の区別 615
データ型
FLOAT 613
Adaptive Server Enterprise 608
IMAGE 78, 610, 614
IQ 608
LONG BINARY 78, 610, 614

- SQL Anywhere 608
- TEXT 70, 609, 613
- UNIQUEIDENTIFIERSTR 69
- 互換性 91
- 情報の表示 382, 408
- 数値 73
- 説明 69
- バイナリ 77
- 日付および時刻 83
- 丸めエラー 76
- 文字 69
- ユーザ定義 89
- データ型の互換性
 - bit データ 608
 - IMAGE データ 614
 - Java データ 614
 - TEXT データ 613
 - 概数値データ 613
 - 数値データ 612
 - 日時と時刻のデータ 612
 - バイナリ・データ 610
 - 日付と時刻のデータ 611
 - 文字データ 609
- データ型変換
 - BIT から BINARY に変換 92
 - BIT から CHAR へ 94
 - BIT から VARBINARY に変換 92
 - BIT から VARCHAR へ 94
 - CHAR から BIT へ 94
 - VARCHAR から BIT へ 94
 - 関数 110
 - 説明 90
- データ型変換関数 110
 - BIGINTTOHEX 132
 - CAST 134
 - CONVERT 140
 - HEXTOBIGINT 179
 - HEXTOINT 180
 - INTTOHEX 191
- データベース
 - ID 番号の調査 162, 210
 - 大文字と小文字の区別 615
 - カタログ・ストアの検証 511
 - システム・テーブル 597
 - システム・プロシージャ 349
 - 名前の調査 162
 - プロパティ 122
 - プロパティ値 163
- データベース・オブジェクト
 - ID の調査 216
 - 識別 22
 - 名前の調査 216
- データベース・オプション
 - DATE_ORDER 87
 - FLOAT_AS_DOUBLE 613
 - QUOTED_IDENTIFIER 33
- データベース管理者
 - 役割 607
- データベース・サーバ
 - コマンド・ライン・オプションの上書き 503
- テーブル
 - iq_dummy 104
 - Transact-SQL 617
 - ジョイン・インデックスごとの数 347
 - 情報の表示 408
- デバイス
 - 管理 605
- デフォルト
 - CURRENT DATE 54
 - CURRENT PUBLISHER 54
 - CURRENT TIME 54
 - CURRENT TIMESTAMP 55
 - CURRENT USER 55
 - Transact-SQL 620
- デフォルト値
 - CURRENT DATABASE 54
 - CURRENT PUBLISHER 54
 - CURRENT USER 55
 - LAST USER 55
 - TIMESTAMP 56
 - USER 57
 - サポートなし 619
- テンポラリ・テーブル
 - Transact-SQL 620

索引

と

- 統計関数 107
- 特殊文字
 - 文字列中 24
- 特別値
 - CURRENT DATABASE 54
 - CURRENT DATE 54
 - CURRENT PUBLISHER 54
 - CURRENT TIME 54
 - CURRENT TIMESTAMP 55
 - CURRENT USER 55
 - LAST USER 55
 - SQLCODE 56
 - SQLSTATE 56
 - TIMESTAMP 56
 - USER 57
- ドメイン
 - 説明 89
- トラブルシューティング
 - request_level_logging 503
 - ロギング操作 507
- トランザクション管理
 - sp_iqsysmon を使用したモニタリング 465
- トランザクション・ログ
 - 文字列の追加 491
- トリガ
 - サポートなし 621

な

- 長さがゼロの文字列
 - Transact-SQL 630

に

- 二重引用符
 - SQL 識別子で使用できない文字 22
 - データベース・オブジェクト 22
- 日時と時刻のデータ型
 - 互換性 612

ね

- ネーム・スペース
 - インデックス 616
- ネストした外部ジョイン 629

は

- パーセントイル
 - NTILE 関数による計算 212
- パーティション
 - 一貫性の検査 358
- パーミッション
 - Adaptive Server Enterprise 622
 - SYSCOLAUTH システム・ビュー 535
 - システム・ビュー 583
- バイト長 217
- バイナリ・データ
 - 互換性 610
- パスワード
 - sa_verify_password システム・プロシージャ 352, 513
 - 大文字と小文字の区別 616
 - 使用期限 352
 - 追加または変更 439
- パターン・マッチング
 - 照合 43
 - 制限 43
 - 説明 42
- バックアップ
 - アクティビティが活発でない間 491
 - チェックポイント中 491
- バックアップ操作
 - 概要 356
- バックアップ履歴ファイル
 - ロケーション 9
- バッチ
 - Transact-SQL の概要 637
 - 作成 637
- バッファ・キャッシュ
 - sp_iqsysmon を使用したモニタリング 465
- パフォーマンス
 - sp_iqshowpsexec 接続情報 456

sp_iqsysmon プロシージャ 465
 モニタリング 465
 パブリッシャ
 SQL Remote 54

ひ

比較

説明 34

引数の選択 129

日付

クエリ 86

現在の調査 212, 268

算術式 635

文字列を日付として解釈 87

日付と時刻の関数 111

DATE 149

DATEADD 149

DATECEILING 150

DATEDIFF 152

DATEFLOOR 154

DATEFORMAT 156

DATENAME 157

DATEPART 157

DATEROUND 158

DATETIME 160

DAY 160

DAYNAME 160

DAYS 161

DOW 166

GETDATE 175

HOUR 181

HOURS 181

IQ の機能 342

MINUTE 205

MINUTES 205

MONTH 206

MONTHNAME 206

MONTHS 207

NOW 212

QUARTER 228

SECOND 249

SECONDS 249

TODAY 268

WEEKS 334

YEAR 338

YEARS 339

YMD 340

一貫性のある結果を取得 113

日付と時刻のデータ型

互換性 611

日付と時刻の比較 86

ビット処理演算子 28

ビット長 133

ビュー

更新可能 635

システム・ビュー 590

情報の表示 408

標準偏差

関数 258

標本関数 260

母関数 259

標本分散関数 332

ヒント

インデックス設定 50

実行フェーズ 51, 52

ふ

ファイル

ロケーション 3

プライマリ・キー

UUID と GUID 208

UUID による一意な値の生成 208

一意な値の生成 208

情報の表示 440

ブランク

後続の削除 71, 609

プリフェッチ

sp_iqsysmon を使用したモニタリング 465

プロシージャ

Transact-SQL 639

Transact-SQL の概要 636

エラー処理 641, 643

情報の表示 408, 442

パラメータ情報の表示 445

変換 639

戻り値 642

索引

プロシージャ言語
概要 636

プロパティ
ID の記述 226
サーバ 122
サーバ・レベル 225
接続 121
データベース 122
名前の調査 227
番号の調査 228

文
CREATE DEFAULT 620
CREATE DOMAIN 620
CREATE RULE 620
CREATE TABLE 617
DISK INIT 605
DISK MIRROR 605
DISK REFIT 605
DISK REINIT 605
DISK REMIRROR 605
DISK UNMIRROR 605
RAISERROR 643
SELECT 626
SIGNAL 643

分割されたテーブル
確認 361

分散統計関数 107
分析関数 106
DENSE_RANK 164
NTILE 212
PERCENT_RANK 219
PERCENTILE_CONT 220
PERCENTILE_DISC 223
RANK 230

分離、サブクエリ 38

へ

平均 131
平方根関数 257
変数
Transact-SQL 641
グローバル 58, 60
接続レベル 59

説明 57
ローカル 58

ほ

母分散関数 331

ま

マニュアル
SQL Anywhere xxi
マルチプレックス
クエリ・サーバの同期 10
システム・プロシージャ 372

み

見積もり
オブティマイザ 48

む

矛盾のない状態
リストア 453

め

メッセージ・ログ
アーカイブ数の指定 505
サイズの指定 504
メモリ
sp_iqsysmon を使用したモニタリング 465

も

文字セット
指定 12
文字データ
互換性 609

文字列

SOUNDEX 関数 256
 Transact-SQL 33
 位置 137
 大文字への変換 328
 区切り文字 33
 後続ブランクの削除 248
 小文字への変換 196, 201
 説明 24
 先行ブランクの削除 202
 挿入 190
 定数 24, 26
 特殊文字 24
 長さ 133, 136
 長さの調査 198
 部分文字列の置換 243
 部分文字列を返す 265
 ブランクの削除 268
 リテラル文字列 24
 類似性の調査 250
 連結 28, 244, 263
 連結演算子 28
 文字列関数 117
 ASCII 129
 BIT_LENGTH 133
 BYTE_LENGTH 133
 CHAR 136
 CHAR_LENGTH 136
 CHARINDEX 137
 DIFFERENCE 165
 INSERTSTR 190
 LCASE 196
 LEFT 196
 LENGTH 198
 LOCATE 199
 LOWER 201
 LTRIM 202
 OCTET_LENGTH 217
 PATINDEX 217
 REPEAT 242
 REPLACE 243
 REPLICATE 244
 RIGHT 245
 RTRIM 248

SIMILAR 250
 SORTKEY 251
 SOUNDEX 256
 SPACE 257
 STR 261
 STRING 263
 STUFF 265
 SUBSTR 265
 SUBSTRING 265
 TRIM 268
 UCASE 328
 UPPER 329

戻り値

プロシージャ 642

モニタ

sp_iqsysmon プロシージャ 465

や

役割

Adaptive Server Enterprise 607

ゆ

ユーザ

削除 398
 情報の表示 486
 接続数 372
 追加 353
 変更 436

ユーザ ID

Adaptive Server Enterprise 622
 大文字と小文字の区別 616
 ユーザ名からの調査 267, 329

ユーザ管理、「ログイン管理」参照

ユーザ指定の条件

クエリ 48

ユーザ指定の条件の選択性 48

ユーザ指定の条件ヒント、ガイドラインと使用方法 53

ユーザ指定の条件ヒント文字列 49

ユーザ定義関数 123

互換性 634

索引

ユーザ定義データ型

Transact-SQL 90

大文字と小文字の区別 615

説明 89

ユーザ名

ユーザ ID からの調査 267, 330

優先度、演算子 30

ユーティリティ

SQL Anywhere 644

有用性ヒント 52

よ

要求レベル・ロギング

Interactive SQL からの有効化 507

説明 503

曜日 (DOW) 166

余弦 143

余接 144

予約語 20

リスト 20

ら

ライブラリ

IMSL エラー処理 126

IMSL エラー・ロギング 127

ランク付け関数 106

り

リストア操作

矛盾のない状態 453

リテラル文字列 24, 26

リモート・サーバ

機能 518

リモート・テーブル

カラム 513, 514, 515

リスト 517

る

ルール

Transact-SQL 620

れ

レジストリ・エントリ

説明 15

連結文字列 28

連鎖外部ジョイン 629

ろ

ロー

カウント 146

ローカル変数

説明 58

ローカル・マシン

環境設定 15

ロード形式

Transact-SQL と SQL Anywhere 625

ログイン管理

sp_expireallpasswords 352

sp_iqaddlogin 353

sp_iqcopyloginpolicy 379, 435

ログイン・ポリシー

コピー 379, 435

ユーザの割り当て 436

ロック

表示 432