



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

SAP Sybase IQ 16.0 SP03

ドキュメント ID：DC01135-01-1603-01

改訂：2013 年 11 月

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG による明示的な許可なく複製または転載することは、形態や目的を問わず禁じられています。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダ独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

目次

対象読者	1
ファイルロケーションとインストール設定	3
インストールディレクトリ構造	3
SAP Sybase IQ によるファイルの検出方法	4
簡単なファイル検索	6
広範囲なファイル検索	6
環境変数	7
Windows での環境変数の設定	7
UNIX 環境ソースファイルの実行	7
UNIX での環境変数の設定	8
IQCHARSET 環境変数	8
IQDIR16 環境変数	8
IQLANG 環境変数	9
IQLOGDIR16 環境変数	11
IQTMP16 環境変数	11
IQ_USE_DIRECTIO 環境変数	12
JAVA_HOME 環境変数	13
LIBPATH 環境変数と LD_LIBRARY_PATH 環境 変数	13
PATH 環境変数	14
SQLCONNECT 環境変数	14
SYBASE 環境変数	15
\$SYBASE_JRE7_64、\$SYBASE_JRE7_32 環 境変数	15
SYBASE_OCS 環境変数	16
レジストリエントリ	16
カレントユーザ設定とローカルマシン設定	16
レジストリ構造	17

インストール時のレジストリ設定	17
SQL 言語の要素	19
キーワード	19
予約語	19
識別子	26
データベースサーバの命名制限	28
文字列	28
式	29
式内の定数	31
式内のカラム名	31
式のサブクエリ	32
SQL 演算子	33
IF 式	38
CASE 式	38
式と定数の互換性	40
検索条件	43
比較条件	44
3 値的論理	46
検索条件内のサブクエリ	47
ALL または ANY 条件	50
BETWEEN 条件	51
論理演算子を使用した条件	52
CONTAINS 条件	52
EXISTS 条件	53
IN 条件	53
IS DISTINCT FROM 探索条件	54
IS NULL 条件	54
LIKE 条件	55
NOT 条件	59
真理値条件	59
ユーザ指定の条件ヒント	60
特別値	69

CURRENT DATABASE 特別値	69
CURRENT DATE 特別値	69
CURRENT PUBLISHER 特別値	70
CURRENT TIME 特別値	70
CURRENT TIMESTAMP 特別値	70
CURRENT USER 特別値	71
LAST USER 特別値	71
SQLCODE 特別値	72
SQLSTATE 特別値	72
TIMESTAMP 特別値	72
USER 特別値	73
変数	73
ローカル変数	73
接続レベル変数	75
グローバル変数	75
コメント	80
NULL 値	82
SQL 関数	85
集合関数	85
分析関数	87
ウィンドウ集合関数の使用法	89
ランク付け関数の使用法	90
統計集合分析関数の使用法	91
分散統計関数の使用法	91
Interrow 関数の使用法	92
データ型変換関数	94
日付と時刻の関数	95
日付要素	97
HTTP 関数	98
数値関数	99
文字列関数	100
システム関数	102

Adaptive Server システム関数の互換性	103
接続プロパティ	104
サーバで使用可能なプロパティ	105
各データベースで使用可能なプロパティ	134
SQL および 外部環境ユーザ定義関数	155
SQL のユーザ定義関数	155
Java のユーザ定義関数	155
その他の関数	156
アルファベット順の関数リスト	156
ABS 関数 [数値]	157
ACOS 関数 [数値]	157
ARGN 関数 [その他]	158
ASCII 関数 [文字列]	159
ASIN 関数 [数値]	160
ATAN 関数 [数値]	161
ATAN2 関数 [数値]	162
AVG 関数 [集合]	162
BFILE 関数 [データ抽出]	163
BIGINTTOHEX 関数 [データ型変換]	164
BIT_LENGTH 関数 [文字列]	165
BYTE_LENGTH 関数 [文字列]	166
BYTE_LENGTH64 関数	167
BYTE_SUBSTR64 関数と BYTE_SUBSTR 関 数	167
CAST 関数 [データ型変換]	168
CEIL 関数 [数値]	169
CEILING 関数 [数値]	170
CHAR 関数 [文字列]	171
CHAR_LENGTH 関数 [文字列]	171
CHAR_LENGTH64 関数	173
CHARINDEX 関数 [文字列]	173
COALESCE 関数 [その他]	174

COL_LENGTH 関数 [システム]	175
COL_NAME 関数 [システム]	176
CONNECTION_PROPERTY 関数 [システム]	177
CONVERT 関数 [データ型変換]	178
CORR 関数 [集合]	182
COS 関数 [数値]	183
COT 関数 [数値]	184
COVAR_POP 関数 [集合]	185
COVAR_SAMP 関数 [集合]	186
COUNT 関数 [集合]	187
CUME_DIST 関数 [統計]	188
DATALLENGTH 関数 [システム]	189
DATE 関数 [日付と時刻]	191
DATEADD 関数 [日付と時刻]	191
DATECEILING 関数 [日付と時刻]	193
DATEDIFF 関数 [日付と時刻]	196
DATEFLOOR 関数 [日付と時刻]	198
DATEFORMAT 関数 [日付と時刻]	201
DATENAME 関数 [日付と時刻]	202
DATEPART 関数 [日付と時刻]	203
DATEROUND 関数 [日付と時刻]	205
DATETIME 関数 [日付と時刻]	208
DAY 関数 [日付と時刻]	208
DAYNAME 関数 [日付と時刻]	209
DAYS 関数 [日付と時刻]	209
DB_ID 関数 [システム]	211
DB_NAME 関数 [システム]	212
DB_PROPERTY 関数 [システム]	213
DEGREES 関数 [数値]	214
DENSE_RANK 関数 [統計]	214
DIFFERENCE 関数 [文字列]	216
DOW 関数 [日付と時刻]	217

ERRORMSG 関数 [その他]	218
EVENT_CONDITION 関数 [システム]	219
EVENT_CONDITION_NAME 関数 [システム]	221
EVENT_PARAMETER 関数 [システム]	221
EXP 関数 [数値]	223
EXP_WEIGHTED_AVG 関数 [集合]	223
FIRST_VALUE 関数 [集合]	225
FLOOR 関数 [数値]	227
GETDATE 関数 [日付と時刻]	228
GRAPHICAL_PLAN 関数 [文字列]	228
GROUPING 関数 [集合]	231
GROUP_MEMBER 関数 [システム]	232
HEXTOBIGINT 関数 [データ型変換]	232
HEXTOINT 関数 [データ型変換]	234
HOUR 関数 [日付と時刻]	235
HOURS 関数 [日付と時刻]	236
HTML_DECODE 関数 [その他]	237
HTML_ENCODE 関数 [その他]	239
HTML_PLAN 関数 [文字列]	240
HTTP_DECODE 関数 [Web サービス]	241
HTTP_ENCODE 関数 [Web サービス]	242
HTTP_HEADER 関数 [Web サービス]	243
HTTP_VARIABLE 関数 [Web サービス]	245
IFNULL 関数 [その他]	247
INDEX_COL 関数 [システム]	248
INSERTSTR 関数 [文字列]	249
INTTOHEX 関数 [データ型変換]	250
ISDATE 関数 [日付および時刻]	251
ISNULL 関数 [その他]	252
ISNUMERIC 関数 [その他]	253
LAG 関数 [統計]	254
LAST_VALUE 関数 [集合]	256

LCASE 関数 [文字列]	258
LEAD 関数 [統計]	259
LEFT 関数 [文字列]	261
LEN 関数 [文字列]	262
LENGTH 関数 [文字列]	263
LIST 関数 [集合]	264
LN 関数 [数値]	267
LOCATE 関数 [文字列]	268
LOG 関数 [数値]	270
LOG10 関数 [数値]	270
LOWER 関数 [文字列]	271
LTRIM 関数 [文字列]	272
MAX 関数 [集合]	273
MEDIAN 関数 [集合]	274
MIN 関数 [集合]	276
MINUTE 関数 [日付と時刻]	276
MINUTES 関数 [日付と時刻]	277
MOD 関数 [数値]	278
MONTH 関数 [日付と時刻]	279
MONTHNAME 関数 [日付と時刻]	280
MONTHS 関数 [日付と時刻]	281
NEWID 関数 [その他]	282
NEXT_CONNECTION 関数 [システム]	284
NEXT_DATABASE 関数 [システム]	285
NEXT_HTTP_HEADER 関数 [Web サービス] ..	286
NEXT_HTTP_VARIABLE 関数 [Web サービス]	288
NOW 関数 [日付と時刻]	289
NTILE 関数 [統計]	289
NULLIF 関数 [その他]	291
NUMBER 関数 [その他]	292
OBJECT_ID 関数 [システム]	293

OBJECT_NAME 関数 [システム].....	294
OCTET_LENGTH 関数 [文字列].....	295
PATINDEX 関数 [文字列].....	296
PERCENT_RANK 関数 [統計].....	298
PERCENTILE_CONT 関数 [統計].....	299
PERCENTILE_DISC 関数 [統計].....	302
PI 関数 [数値].....	304
POWER 関数 [数値].....	304
PROPERTY 関数 [システム].....	305
PROPERTY_DESCRIPTION 関数 [システム]...	306
PROPERTY_NAME 関数 [システム].....	307
PROPERTY_NUMBER 関数 [システム].....	307
QUARTER 関数 [日付と時刻].....	308
RADIANS 関数 [数値].....	309
RAND 関数 [数値].....	310
RANK 関数 [統計].....	311
REGR_AVGX 関数 [集合].....	312
REGR_AVGY 関数 [集合].....	313
REGR_COUNT 関数 [集合].....	315
REGR_INTERCEPT 関数 [集合].....	316
REGR_R2 関数 [集合].....	317
REGR_SLOPE 関数 [集合].....	318
REGR_SXX 関数 [集合].....	320
REGR_SXY 関数 [集合].....	321
REGR_SYY 関数 [集合].....	322
REMAINDER 関数 [数値].....	323
REPEAT 関数 [文字列].....	324
REPLACE 関数 [文字列].....	325
REPLICATE 関数 [文字列].....	327
REVERSE 関数 [文字列].....	329
RIGHT 関数 [文字列].....	330
ROUND 関数 [数値].....	331

ROW_NUMBER 関数 [統計]	332
ROWID 関数 [その他]	334
RTRIM 関数 [文字列]	335
SECOND 関数 [日付と時刻]	336
SECONDS 関数 [日付と時刻]	337
SIGN 関数 [数値]	338
SIMILAR 関数 [文字列]	338
SIN 関数 [数値]	339
SORTKEY 関数 [文字列]	340
SOUNDEX 関数 [文字列]	345
SP_HAS_ROLE 関数 [システム]	346
SPACE 関数 [文字列]	349
SQLFLAGGER 関数 [その他]	349
SQRT 関数 [数値]	351
SQUARE 関数 [数値]	351
STDDEV 関数 [集合]	352
STDDEV_POP 関数 [集合]	354
STDDEV_SAMP 関数 [集合]	355
STR 関数 [文字列]	356
STR_REPLACE 関数 [文字列]	357
STRING 関数 [文字列]	359
STRTOUUID 関数 [文字列]	360
STUFF 関数 [文字列]	361
SUBSTRING 関数 [文字列]	362
SUBSTRING64 関数 [文字列]	365
SUM 関数 [集合]	365
SUSER_ID 関数 [システム]	366
SUSER_NAME 関数 [システム]	367
TAN 関数 [数値]	368
TODAY 関数 [日付と時刻]	368
TRIM 関数 [文字列]	369
TRUNCNUM 関数 [数値]	370

UCASE 関数 [文字列]	371
UPPER 関数 [文字列]	372
USER_ID 関数 [システム]	373
USER_NAME 関数 [システム]	373
UUIDTOSTR 関数 [文字列]	374
VAR_POP 関数 [集合]	375
VAR_SAMP 関数 [集合]	377
VARIANCE 関数 [集合]	378
WEEKS 関数 [日付と時刻]	380
WEIGHTED_AVG 関数 [集合]	381
WIDTH_BUCKET 関数 [数値]	383
YEAR 関数 [日付と時刻]	385
YEARS 関数 [日付と時刻]	386
YMD 関数 [日付と時刻]	388
SQL データ型	389
文字データ型	389
記憶領域サイズ	390
文字セットとコードページ	391
インデックス	392
VARCHAR データと後続ブランク	392
255 バイトを超える CHAR データと VARCHAR データに関する制限	393
文字データの互換性	393
長い文字列	393
数値データ型	394
数値データ型の使用法	394
バイナリデータ型	398
バイナリデータ型の使用法	399
Bit データ型	404
Bit データの互換性	404
日付と時刻のデータ型	405
日付と時刻のデータ型の使用法	405

ドメイン	410
単純なドメイン	411
CREATE DOMAIN 文	412
ドメインの互換性	414
データ型変換	415
文字列から日付／時刻への変換の互換性	416
エクスポートされた日付の互換性	417
BIT から BINARY へのデータ型の変換	417
BIT と CHAR/VARCHAR 間のデータ型変換	418
他の SQL 言語との違い	421
日付	421
整合性	421
ジョイン	422
更新	422
テーブルの変更	422
サブクエリが許容されない場合	423
その他の関数	423
カーソル	423
物理的制限	425
システムプロシージャ	429
権限付きシステムプロシージャの実行管理	429
互換ロールの移行の意味	430
権限付きシステムプロシージャを実行する機能の付与	431
権限付きシステムプロシージャを実行する機能の取り消し	432
データベースが使用するセキュリティモデルの特定	432
16.0 より前の権限付きシステムプロシージャ	433
ストアドプロシージャの構文規則	435

ストアードプロシージャが報告する統計情報を理解する	436
SQL Anywhere でサポートされているプロシージャ	436
システムストアードプロシージャのアルファベット順リスト	436
sp_iqaddlogin プロシージャ	437
sp_iqbackupdetails プロシージャ	438
sp_iqbackupsummary プロシージャ	441
sp_iqcardinality_analysis プロシージャ	442
sp_iqcheckdb プロシージャ	443
sp_iqcheckoptions プロシージャ	453
sp_iqclient_lookup プロシージャ	455
sp_iqcolumn プロシージャ	457
sp_iqcolumnmetadata プロシージャ	459
sp_iqcolumnuse プロシージャ	460
sp_iqconnection プロシージャ	461
sp_iqconstraint プロシージャ	465
sp_iqcontext プロシージャ	466
sp_iqcopyloginpolicy プロシージャ	469
sp_iqcursorinfo プロシージャ	470
sp_iqdatatype プロシージャ	473
sp_iqdbsize プロシージャ	476
sp_iqdbspace プロシージャ	477
sp_iqdbspaceinfo プロシージャ	481
sp_iqdbspaceobjectinfo プロシージャ	485
sp_iqdbstatistics プロシージャ	489
sp_iqdroplogin プロシージャ	490
sp_iqemptyfile プロシージャ	491
sp_iqestdbspaces プロシージャ	492
sp_iqestspace プロシージャ	495
sp_iqevent プロシージャ	496

sp_iqfile プロシージャ	499
sp_iqhelp プロシージャ	502
sp_iqindex および sp_iqindex_alt プロシージャ	509
sp_iqindexadvice プロシージャ	513
sp_iqindexfragmentation プロシージャ	514
sp_iqindexinfo プロシージャ	516
sp_iqindexmetadata プロシージャ	518
sp_iqindexrebuildwidedata プロシージャ	521
sp_iqindexsize プロシージャ	522
sp_iqindexuse プロシージャ	525
sp_iqlmconfig プロシージャ	527
sp_iqlocks プロシージャ	529
sp_iqmergerlvstore プロシージャ	534
sp_iqmodifyadmin プロシージャ	534
sp_iqmodifylogin プロシージャ	535
sp_iqmpxcheckdqpconfig プロシージャ	536
sp_iqmpxdumptlvlog プロシージャ	538
sp_iqmpxfilestatus プロシージャ	539
sp_iqmpxinconnpoolinfo プロシージャ	540
sp_iqmpxincheartbeatinfo プロシージャ	541
sp_iqmpxincstatistics プロシージャ	542
sp_iqmpxinfo プロシージャ	543
sp_iqmpxsuspendedconninfo プロシージャ	545
sp_iqmpxvalidate プロシージャ	546
sp_iqmpxversioninfo プロシージャ	547
sp_iqobjectinfo プロシージャ	548
sp_iqpassword プロシージャ	551
sp_iqpkkeys プロシージャ	553
sp_iqprocedure プロシージャ	555
sp_iqprocparm プロシージャ	558
sp_iqrebuildindex プロシージャ	561

sp_iqrebuildindexwide プロシージャ	564
sp_iqrename プロシージャ	568
sp_iq_reset_identity プロシージャ	570
sp_iqrestoreaction プロシージャ	571
sp_iqrlvmemory プロシージャ	573
sp_iqrowdensity プロシージャ	574
sp_iqsetcompression プロシージャ	576
sp_iqsharedtempdistrib プロシージャ	577
sp_iqshowcompression プロシージャ	578
sp_iqshowpsexec プロシージャ	579
sp_iqspaceinfo プロシージャ	581
sp_iqspaceused プロシージャ	582
sp_iqstatistics プロシージャ	585
sp_iqstatus プロシージャ	589
sp_iqsysmon プロシージャ	592
sp_iqtable プロシージャ	614
sp_iqtablesize プロシージャ	618
sp_iqtableuse プロシージャ	620
sp_iqtransaction プロシージャ	621
sp_iqunusedcolumn プロシージャ	625
sp_iqunusedindex プロシージャ	627
sp_iqunusedtable プロシージャ	628
sp_iqversionuse プロシージャ	629
sp_iqview プロシージャ	632
sp_iqwho プロシージャ	634
sp_iqworkmon プロシージャ	637
カタログストアードプロシージャのアルファベット順 リスト	639
sa_ansi_standard_packages システムプロシ ージャ	639
sa_audit_string システムプロシージャ	640
sa_char_terms システムプロシージャ	640

sa_checkpoint_execute システムプロシージャ	642
sa_conn_activity システムプロシージャ	643
sa_conn_info システムプロシージャ	645
sa_conn_list システムプロシージャ	648
sa_conn_properties システムプロシージャ	649
sa_db_info システムプロシージャ	651
sa_db_properties システムプロシージャ	652
sa_describe_shapefile システムプロシージャ	654
sa_dependent_views システムプロシージャ ...	656
sa_disable_auditing_type システムプロシ ージャ	657
sa_disk_free_space システムプロシージャ	658
sa_enable_auditing_type システムプロシ ージャ	659
sa_eng_properties システムプロシージャ	661
sa_external_library_unload システムプロシ ージャ	662
sa_flush_cache システムプロシージャ	662
sa_get_user_status システムプロシージャ	663
sa_get_ldapservers_status システムプロシ ージャ	665
sa_list_external_library システムプロシ ージャ	666
sa_locks システムプロシージャ	666
sa_make_object システムプロシージャ	670
sa_nchar_terms システムプロシージャ	673
sa_performance_diagnostics システムプロ シージャ	673
sa_procedure_profile_summary システムプロ シージャ	679

sa_procedure_profile システムプロシージャ	681
sa_report_deadlocks システムプロシージャ ...	683
sa_rowgenerator システムプロシージャ	684
sa_server_option システムプロシージャ	687
sa_set_http_header システムプロシージャ	704
sa_set_http_option システムプロシージャ	705
sa_table_page_usage システムプロシージャ	710
sa_text_index_stats システムプロシージャ	711
sa_text_index_vocab システムプロシージャ ...	712
sa_validate システムプロシージャ	713
sa_verify_password システムプロシージャ	715
sp_alter_secure_feature_key システムプロ シージャ	716
sp_auth_sys_role_info システムプロシージャ	716
sp_create_secure_feature_key システムプロ シージャ	717
sp_displayroles システムプロシージャ	718
sp_drop_secure_feature_key システムプロ シージャ	721
sp_expireallpasswords システムプロシージャ	721
sp_list_secure_feature_key システムプロシ ージャ	722
sp_login_environment システムプロシージャ	723
sp_objectpermission システムプロシージャ	723
sp_proc_priv システムプロシージャ	727
sp_remote_columns システムプロシージャ	729

sp_remote_exported_keys システムプロシ ージャ	731
sp_remote_imported_keys システムプロシ ージャ	732
sp_remote_primary_keys システムプロシ ージャ	734
sp_remote_tables システムプロシージャ	735
sp_servercaps システムプロシージャ	737
sp_sys_priv_role_info システム権限	738
sp_tsq_environment システムプロシージャ	738
sp_use_secure_feature_key システムプロシ ージャ	739
Adaptive Server のシステムプロシージャおよびカタ ログプロシージャ	740
Adaptive Server のシステムプロシージャ	741
Adaptive Server のカタログプロシージャ	742
システムテーブルとシステムビュー	745
システムテーブル	745
SYS.DUMMY テーブルと IQ_DUMMY テーブ ルの比較	748
システムビュー	749
統合ビュー	749
互換ビュー	750
システムビューのアルファベット順リスト	750
他の Sybase データベースとの互換性	869
SQL Anywhere について	869
Transact-SQL サポートの概要	869
Adaptive Server、SQL Anywhere、および SAP Sybase IQ のアーキテクチャ	870
サーバとデータベース	871
領域の割り付けとデバイス管理	871

システムテーブル、カタログストア、IQ メイ ンストア	872
データ型	872
Bit データ型	873
文字データ型の互換性	873
バイナリデータ型	874
date、time、datetime、timestamp データ型	875
数値データ型	877
text データ型	877
image データ型	878
Java データ型	878
データ定義言語	878
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	878
大文字と小文字の区別	878
オブジェクト名の互換性の確保	879
CREATE TABLE 文使用時の考慮事項	880
CREATE DEFAULT 文、CREATE RULE 文、 CREATE DOMAIN 文使用時の考慮事項	884
CREATE TRIGGER 文使用時の考慮事項	885
CREATE INDEX 文使用時の考慮事項	885
ユーザ、グループ/ロール、およびパーミッ ション	886
ロード形式	888
Transact-SQL 互換のオプション	888
データ操作言語	888
移植可能な SQL を記述するための一般的なガ イドライン	888
互換性のあるクエリの記述方法の基準	889
サブクエリのサポート	890
GROUP BY 句のサポート	890

COMPUTE 句のサポート	891
WHERE 句のサポート	891
Transact-SQL 外部ジョインのサポート	891
ANSI ジョインのサポート	892
Null 比較のサポート	892
長さがゼロの文字列のサポート	892
HOLDLOCK、SHARED、FOR BROWSE のサ ポート	893
SQL 関数のサポート	893
OLAP 関数のサポート	894
システム関数のサポート	895
ユーザ定義関数のサポート	896
日付の算術式の異なる解釈	896
SELECT INTO 文のサポート	896
更新可能なビューのサポート	897
UPDATE と DELETE の FROM 句のサポート ..	897
Transact-SQL のプロシージャ言語の概要	897
Transact-SQL のストアードプロシージャの概要	897
Transact-SQL のバッチの概要	898
プロシージャとバッチ内の SQL 文	898
ストアードプロシージャの自動変換	900
Transact-SQL プロシージャから返される結果セット	900
Transact-SQL プロシージャ内の変数	901
Transact-SQL プロシージャでのエラー処理	902
プロシージャ内での RAISERROR 文の使用	903
Watcom-SQL ダイアレクトでの Transact-SQL に似たエラー処理	904
SQL Anywhere と SAP Sybase IQ の相違点および共 通機能	904

SQL Anywhere サーバおよびデータベースの起 動と管理	904
SQL Anywhere データ定義言語 (DDL) の相違点	905
SQL Anywhere データ操作言語 (DML) の相違 点	906
Adaptive Server と SAP Sybase IQ の相違点および共 通機能	907
Adaptive Server ストアドプロシージャ	907
Adaptive Server システムビュー	907
索引	909

対象読者

このマニュアルは、SQL 文、言語要素、データ型、関数、システムプロシージャ、システムテーブルに関するリファレンス資料を必要とする SAP® Sybase® IQ ユーザを対象としています。

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。このマニュアルを使用して、SQL 構文、パラメータ、オプションの情報を確認してください。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

ファイルロケーションとインストール設定

ここでは、SAP 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。

インストールディレクトリ構造

SAP Sybase IQ をインストールすると、いくつかのディレクトリが作成されます。この項ではディレクトリ構造について説明します。

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。

デフォルトでは、SAP 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。SAP Sybase IQ に付随する他のツールも、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にインストールされます。この項では、SAP Sybase IQ のサブディレクトリ構造についてのみ説明します。

デフォルトでは、SAP Sybase IQ ディレクトリは `IQ-16_0` です。`IQ-16_0` のロケーションは、SAP Sybase IQ のインストール場所によって変わります。`IQ-16_0` ディレクトリは、UNIX の環境変数 `$IQDIR16` または Windows の環境変数 `%IQDIR16%` によっても参照されます。

SAP Sybase IQ ディレクトリ内には、次のようなディレクトリとファイルがあります。

- **デモディレクトリ** (`%ALLUSERSPROFILE%/SybaseIQ/demo`) – `iqdemo` データベースの構築に必要なツールが格納されています。`iqdemo` データベースファイルとして、`iqdemo.db`、`iqdemo.iq`、`iqdemo.iqmsg`、`iqdemo.iqtmp` があります。デモデータベース自体は SAP Sybase IQ に同梱されていません。
- **サブディレクトリ** `/demo/adata-16.x` の `iqdemo` データベースを作成するための `16.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 SAP Sybase IQ 12.7 データがあります。`16.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-16_0/scripts)** – 例として使用するスクリプトやストアードプロシージャのようなカタログオブジェクトの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 サンプルとユーザ定義関数 (UDF) サンプルがあります。%ALLUSERSPROFILE%/SybaseIQ/samples/sqlanywhere には、SQL サンプルのディレクトリがあります。sqlanywhere/c directory には、SAP Sybase SQL Anywhere® で ESQL (埋め込み SQL) と C を使用する方法を示した C++ のサンプルがあります。SQL Anywhere と SAP 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 SAP Sybase IQ 用に変更できます。%ALLUSERSPROFILE%/SybaseIQ/samples/udf ディレクトリには、サンプルの C++ スカラと集合 UDF があります。
- **実行プログラムディレクトリ** – 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-16_0 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-16_0 のサブディレクトリ %h、%install、%java、%bin32 があります。

SAP Sybase IQ によるファイルの検出方法

SAP 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。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。

適切なファイルを使用するには、SAP Sybase IQ によるファイルの検出方法を理解しておくことが重要です。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。ファイル名拡張子は、UNIX では .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、SAP 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 \$IQDIR16/lib64 または \$SYBASE/\$SYBASE_OCS/lib64 にあり、Windows では %IQDIR16%

¥bin32 または %SYBASE¥%SYBASE_OCS¥dll 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 \$IQDIR16/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。SAP Sybase IQ は、SAP 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ください。

- インタフェースファイル – SAP Sybase IQ の実行に必要です。たとえば、UNIX では .odbc.ini と utility_db.ini、Windows では util_db.ini が該当します。これらのファイルの詳細については、『管理：ユーザ管理とセキュリティ』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 – データとメタデータを格納します。たとえば、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 %ALLUSERSPROFILE%/logfiles/yourservername.0006.srvlog などの名前が付けられることもあります。データベースに接続すると、データベースログ(%ALLUSERSPROFILE%/demo/iqdemo.log など)が作成されます。これらのファイルの詳細については、『管理：データベース』と『ユーティリティガイド』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作成されます。

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 – この文およびファイルを作成できる同様の文には、ファイル名が必要です。

SAP 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、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 SAP 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、SAP Sybase IQ プログラムは次の順序でファイルを検索します。

- **実行プログラムディレクトリ** – 実行プログラムが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。
 - 実行プログラムディレクトリの親
 - `scripts.` という親ディレクトリの子
- **現在の作業ディレクトリ** – 起動したプログラムには、現在の作業ディレクトリ (プログラムの起動元のディレクトリ) 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。
- **LOCATION レジストリエントリ** – Windows インストール環境では、SAP 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索され、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の付いた子ディレクトリ (`bin32`、`bin` など)
- **システム固有のディレクトリ** – 一般的なオペレーティングシステムファイルが格納されているディレクトリが含まれます。Windows では、Windows ディレクトリや `Windows¥system` ディレクトリなどです。
- **CLASSPATH ディレクトリ** – Java ファイルの場合、CLASSPATH 環境変数で指定されているディレクトリでファイルが検索されます。
- **PATH ディレクトリ** – システムパスとユーザパスのディレクトリでファイルが検索されます。
- **LIBRARY PATH ディレクトリ** – LIBPATH 環境変数で指定されているディレクトリで共有ライブラリが検索されます。

環境変数

SAP 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。

Windows での環境変数の設定

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、次の手順に従ってください。

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 環境ソースファイルの実行

UNIX では、環境ソースファイルによって必須の環境変数を設定します。

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-16_0/IQ-16_0.sh
```

2. C シェルでは、次のように入力します。

```
source $SYBASE/IQ-16_0/IQ-16_0.csh;  
rehash
```

UNIX での環境変数の設定

UNIX プラットフォームでは、環境ソースファイルを実行して必須環境変数を設定します。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、次の手順に従ってください。

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"
```

IQCHARSET 環境変数

`IQCHARSET` は、デフォルトの文字セットを設定します。

`Charset` は文字セットの名前です。たとえば、`IQCHARSET=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。

- `IQCHARSET` 環境変数
- OS での設定

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。

設定値

```
IQCHARSET=charset
```

IQDIR16 環境変数

`IQDIR16` は、SAP Sybase IQ ディレクトリのロケーションを示すとともに、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ロケーションです。

- `IQDIR16/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。

- \$IQDIR16/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。IQLOGDIR16 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。
- \$IQDIR16/demo は、iqdemo データベースファイルのロケーションです。

設定値

```
IQDIR16 = ${SYBASE}/IQ-16_0
```

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。

IQLANG 環境変数

IQLANG はデフォルト言語を設定します。

Language_code は、2 文字で表された言語名です。たとえば、**IQLANG=DE** の場合、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。

- IQLANG 環境変数
- インストーラによって設定されたレジストリ (Windows のみ)
- OS での設定

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。

設定値

```
IQLANG=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。

言語ラベル値のリスト

有効な言語ラベルの値と対応する ISO 639 言語コードは次のとおりです。IQ_LANG 環境変数に 2 文字の ISO_639 言語コードを設定してください。

言語	ISO_639 言語コード	言語ラベル	代替ラベル
アラビア語	AR	arabic	該当なし
チェコ語	CS	czech	該当なし

ファイルロケーションとインストール設定

言語	ISO_639 言語コード	言語ラベル	代替ラベル
デンマーク語	DA	danish	該当なし
オランダ語	NL	dutch	該当なし
英語	EN	us_english	english
フィンランド語	FI	finnish	該当なし
フランス語	FR	french	該当なし
ドイツ語	DE	german	該当なし
ギリシャ語	EL	greek	該当なし
ヘブライ語	HE	hebrew	該当なし
ハンガリー語	HU	hungarian	該当なし
イタリア語	IT	italian	該当なし
日本語	JA	japanese	該当なし
韓国語	KO	korean	該当なし
リトアニア語	LT	lithuanian	該当なし
ノルウェー語	NO	norwegian	norweg
ポーランド語	PL	polish	該当なし
ポルトガル語	PT	portuguese	portugue
ロシア語	RU	russian	該当なし
中国語 (簡体字)	ZH	chinese	simpchin
スペイン語	ES	spanish	該当なし
スウェーデン語	SV	swedish	該当なし
タイ語	TH	thai	該当なし
中国語 (繁体字)	TW	tchinese	tradchin
トルコ語	TR	turkish	該当なし
ウクライナ語	UK	ukrainian	該当なし

IQLOGDIR16 環境変数

IQLOGDIR16 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。IQLOGDIR16 は、インストールプログラムでは設定されません。

- サーバログは、\$IQLOGDIR16 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。

IQLOGDIR16 が有効な書き込み可能ディレクトリとして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対してデフォルトのロケーション `$IQDIR16/logfiles` を使用します。

設定値

```
IQLOGDIR16 = path
```

オペレーティングシステム
オプション。

IQTMP16 環境変数

IQTMP16 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。IQTMP16 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 SAP Sybase IQ で使用されます。

IQTMP16 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、IQTMP16 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： IQTMP16 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、IQ テンポラリストアのデフォルトロケーションを制御しません。IQ テンポラリストアのデフォルトロケーションは、**CREATE DATABASE** 文によって制御します。

警告！ IQTMP16 を \$SYBASE や \$IQDIR16 に設定しないでください。

IQTMP16 を明示的に設定しない場合、IQTMP16 は UNIX ディレクトリ `/tmp` のサブディレクトリに設定されます。

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、SAP 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 IQTMP16 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。
- 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
```

設定値

```
IQTMP16 = 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。

IQ_USE_DIRECTIO 環境変数

IQ_USE_DIRECTIO は、サポートされるシンプレックスプラットフォームでダイレクト I/O を有効にします。

IQ_USE_DIRECTIO 環境変数のデフォルト設定は 0 です。サポートされるプラットフォームでダイレクト I/O を有効にするには、IQ_USE_DIRECTIO を 1 に設定します。マルチプレックスデータベースはダイレクト I/O をサポートしていません。

設定

```
IQ_USE_DIRECTIO = value
```

オペレーティングシステム

Sun Solaris UFS、Linux、Linux IBM、AIX、Windows ファイルシステムでのみ使用できるオプションです。Linux では、ダイレクト I/O はカーネルバージョン 2.6.x でサポートされます。

HP-UX および HP-UXi プラットフォームでは使用されません。

IQ_USE_DIRECTIO は、Sun Solaris と Windows には影響しません。

JAVA_HOME 環境変数

bin/java を含むディレクトリを指す JRE ホームを定義します。

Java VM のロケーションが \$SYBASE_JRE6_32、\$SYBASE_JRE6_64、または \$SYBASE_JRE5_64 環境変数で設定されていない場合に使用されます。

JAVA_HOME は、通常、VM のインストール時に作成されます。

UNIX では、SYBASE.csh (C シェル) または SYBASE.sh (Bourne または Korn シェル) 環境ソースファイルを実行することによって、IQ エンジンの正しい JRE を検索し、開始します。JAVA_HOME で指定されている Java VM のロケーションは、SYBASE.csh または SYBASE.sh によって返されるロケーションよりも優先されます。JAVA_HOME またはスクリプト (SYBASE.csh または SYBASE.sh) のいずれでも Java VM を見つけられなかった場合、IQ は Java VM をロードしません。

設定

```
JAVA_HOME = Sybase/shared/JRE<version>
```

オペレーティングシステム
必須。

LIBPATH 環境変数と LD_LIBRARY_PATH 環境変数

LIBPATH または LD_LIBRARY_PATH は、SAP 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。

設定

AIX の場合:

```
LIBPATH = installation_path/lib
```

その他の 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。

PATH 環境変数

PATH は SAP Sybase IQ 実行プログラムの配置ディレクトリを含んでいる、オペレーティングシステムの必須変数です。

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。

設定値

```
PATH = installation_path
```

オペレーティングシステム
必須。

SQLCONNECT 環境変数

SQLCONNECT は、データベースサーバに接続するとき、Interactive SQL、**dbinfo**、**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。

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。

この文字列は、**parameter = value** 形式のパラメータ設定をセミコロンで区切ったリストで指定します。

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。これにより、ユーザは **ps -ef** 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、Interactive SQL 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に特に有用です。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セキュリティは向上しますが、それで完全になるわけではないことに注意してください。パスワードはプレーンテキストであるため、悪意を持って操作すれば、環境コンテキストからパスワードを抽出することが可能です。詳細については、『管理：データベース』の「接続パラメータと通信パラメータリファレンス」を参照してください。

設定

```
SQLCONNECT = parameter#value ; ...
```

オペレーティングシステム
オプション。

SYBASE 環境変数

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。

SAP 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。

設定値

```
SYBASE = path
```

オペレーティングシステム
必須。

\$\$SYBASE_JRE7_64、\$\$SYBASE_JRE7_32 環境変数

この変数は、Sybase Control Center で使用する Java Runtime Environment のロケーションを指定します。

起動時に、Sybase Control Center は SCC_JAVA_HOME で Java バージョンの定義をチェックします。SCC_JAVA_HOME が定義されていない場合、Sybase Control Center は、インストールされている JRE を次の順序でチェックします。

- SYBASE_JRE7_64
- SYBASE_JRE7
- SYBASE_JRE7_32

その後、このリストで検出された最初の値に SCC_JAVA_HOME を設定します。

設定値

IQ.sh (Bourne/Korn シェル) ファイル、IQ.csh (C シェル) ファイルを参照します。

ヒント： また、次のように手動で JRE を設定することもできます。

```
SCC_JAVA_HOME=${SYBASE}/shared/JRE-7_(minor_version)_64BIT
```

または

```
SCC_JAVA_HOME=${SYBASE}/shared/JRE-7_(minor_version)_32BIT
```

SYBASE_OCS 環境変数

SYBASE_OCS 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。

設定値

```
SYBASE_OCS = "OCS-16_0"
```

オペレーティングシステム
必須。

レジストリエントリ

Windows オペレーティングシステムでは、SAP 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！ 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ため、レジストリを変更することはおすすめしません。

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つのレベルのシステム設定(ユーザ設定とローカルマシン設定)があります。

カレントユーザ設定は、特定のユーザがログオンしたときにだけ使用できる、ユーザ固有の設定です。ローカルマシン設定は、ログオンしているユーザに関わりなく使用できる、マシン全体に関連する設定です。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。

SAP Sybase IQ は、カレントユーザ設定とローカルマシン設定の両方を許可します。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。

SAP 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。

Windows で SAP 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。

通常、ローカルマシン設定を使用します。

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せん。

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]- [アクセス許可] を選択し、読み取りパーミッションをオンにします。

SAP 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 > SAP Sybase IQ 16.0

レジストリエディタの起動

レジストリエディタを起動して、Windows レジストリにアクセスします。

1. [スタート]> [ファイル名を指定して実行] を選びます。
2. [名前] ボックスに、次のように入力します。

```
regedt32
```

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 レジストリ内でレジストリ設定を自動的に行います。

- Location - SAP 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。次に例を示します。

```
Location:REG_SZ:C:\Program Files\Sybase  
¥IQ-16_0
```

SAP 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。

SQL 言語の要素

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

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 SAP Sybase IQ 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 がキーワードです。

```
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、SELECT という名前のテーブルの内容を取得するには、次の構文を使用する必要があります。

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

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。

次の表は、SAP Sybase IQ の SQL 予約語の一覧です。SQL はキーワードの大文字と小文字を区別しないため、この表の各語は大文字、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。

SQL 予約語

- add
- all
- alter
- and
- any
- array
- as
- asc
- attach
- backup
- begin
- between
- bigint
- binary
- bit
- bottom
- break
- by
- call
- capability
- cascade
- case
- cast
- char
- char_convert
- character
- check
- checkpoint
- close
- comment
- commit
- compressed
- conflict
- connect
- constraint
- contains
- continue

- convert
- create
- cross
- cube
- current
- current_timestamp
- current_user
- cursor
- date
- datetimeoffset
- dbspace
- deallocate
- dec
- decimal
- declare
- default
- delete
- deleting
- desc
- detach
- distinct
- do
- double
- drop
- dynamic
- else
- elseif
- encrypted
- end
- endif
- escape
- except
- exception
- exec
- execute
- existing
- exists
- externlogin
- fetch

SQL 言語の要素

- first
- float
- for
- force
- foreign
- forward
- from
- full
- goto
- grant
- group
- having
- holdlock
- identified
- if
- in
- index
- inner
- inout
- insensitive
- insert
- inserting
- install
- instead
- int
- integer
- integrated
- Intersect
- into
- is
- isolation
- join
- json
- kerberos
- key
- lateral
- left
- like
- limit

- lock
- login
- long
- match
- membership
- merge
- message
- mode
- modify
- natural
- nchar
- new
- no
- noholdlock
- not
- notify
- null
- numeric
- nvarchar
- of
- off
- on
- open
- openstring
- openxml
- option
- options
- or
- order
- others
- out
- outer
- over
- passthrough
- precision
- prepare
- primary
- print
- privileges

SQL 言語の要素

- proc
- procedure
- publication
- raiserror
- readtext
- real
- reference
- references
- refresh
- release
- remote
- remove
- rename
- reorganize
- resource
- restore
- restrict
- return
- revoke
- right
- rollback
- rollup
- row
- rowtype
- save
- savepoint
- scroll
- select
- sensitive
- session
- set
- setuser
- share
- smallint
- some
- spatial
- sqlcode
- sqlstate
- start

- stop
- subtrans
- subtransaction
- synchronize
- table
- temporary
- then
- time
- timestamp
- tinyint
- to
- top
- tran
- treat
- trigger
- truncate
- tsequal
- unbounded
- union
- unique
- uniqueidentifier
- unknown
- unnest
- unsigned
- update
- updating
- user
- using
- validate
- values
- varbinary
- varbit
- varchar
- variable
- varray
- varying
- view
- wait
- waitfor

SQL 言語の要素

- when
- where
- while
- window
- with
- within
- work
- writetext
- xml

参照：

- 識別子 (26 ページ)
- quoted_identifier オプション (41 ページ)
- 検索条件内のサブクエリ (47 ページ)
- 式内のカラム名 (31 ページ)

識別子

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。

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。
- 識別子の先頭文字がアルファベット文字ではない (以下を参照)。
- 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。

次の文字を識別子の中で使用することはできません。

- " (二重引用符)
- 制御文字 (0x20 未満の文字)
- 円記号
- 角カッコ (開く)
- 角カッコ (閉じる)

- 逆引用符／抑音符号

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。

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、QUOTED_IDENTIFIER の設定にかかわらず、角カッコは常に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。

制限事項

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。
- ユーザ名に二重引用符またはセミコロン文字を含めることはできない (一重引用符は使用できる)。
- データベース名に二重引用符、一重引用符、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。
- **CREATE DATABASE...CASE IGNORE** または **CASE RESPECT** の指定にかかわらず、データベース名は常に大文字と小文字によって区別される。

-n start_iq サーバオプションを使用する場合には、データベースサーバの命名制限が存在します。

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 (19 ページ)
- quoted_identifier オプション (41 ページ)
- 検索条件内のサブクエリ (47 ページ)
- 式内のカラム名 (31 ページ)

データベースサーバの命名制限

`start iq [server-options]` で `-n` スイッチを使用すると、特定の命名制限が適用されません。

サーバ名では文字セットの変換が実行されません。クライアントの文字セットとデータベースサーバの文字セットが異なる場合、サーバ名で拡張文字を使用すると、サーバが見つからなくなることがあります。クライアントとサーバが異なるオペレーティングシステムまたはロケールで実行されている場合、サーバ名には 7 ビット ASCII 文字を使用します。

データベースサーバ名は、有効な識別子にしてください。長いデータベースサーバ名は、プロトコルに応じてさまざまな長さにトランケートされます。データベースサーバ名では次の操作を行うことができません。

- 最初の文字をスペース、一重引用符または二重引用符にする。
- 最後の文字をスペースにする。
- セミコロンを含める。
- 128 バイトを超える。

サーバ名は、クライアントアプリケーション接続文字列またはプロファイルで使用する名前を指定します。同じ名前のデータベースサーバを複数実行することはおすすめしません。

文字列

文字列は、リテラル文字列またはデータ型 `CHAR`/`VARCHAR` を持つ式です。

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。

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。

```
'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。次に例を示します。

```
'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。次に例を示します。

```
'First line:\nSecond line:'
```

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

```
'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)。次に例を示します。

```
'¥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。

参照：

- 比較条件 (44 ページ)
- 式 (29 ページ)
- NULL 値 (82 ページ)
- 検索条件 (43 ページ)
- 3 値的論理 (46 ページ)
- SQL 演算子 (33 ページ)
- 検索条件内のサブクエリ (47 ページ)

式

式は、定数、カラム名、SQL 演算子、サブクエリなどの数種類の要素で構成されます。

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。

互換性

- Adaptive Server では、IF 条件はサポートされません。
- Adaptive Server では、Java 式は現在サポートされていません。

- 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。

参照：

- 比較条件 (44 ページ)
- NULL 値 (82 ページ)
- 検索条件 (43 ページ)
- 文字列 (28 ページ)
- 3 値的論理 (46 ページ)
- SQL 演算子 (33 ページ)
- 検索条件内のサブクエリ (47 ページ)
- 特別値 (69 ページ)
- CASE 文のサポート (899 ページ)

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。

参照：

- 式内のカラム名 (31 ページ)
- 式のサブクエリ (32 ページ)
- SQL 演算子 (33 ページ)
- IF 式 (38 ページ)
- CASE 式 (38 ページ)
- 式と定数の互換性 (40 ページ)

式内のカラム名

カラム名は識別子の1つであり、前に相関名が付くことがあります。通常、相関名はテーブル名です。

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"
```

参照：

- 式内の定数 (31 ページ)

SQL 言語の要素

- 式のサブクエリ (32 ページ)
- SQL 演算子 (33 ページ)
- IF 式 (38 ページ)
- CASE 式 (38 ページ)
- 式と定数の互換性 (40 ページ)
- 検索条件内のサブクエリ (47 ページ)
- 予約語 (19 ページ)
- 識別子 (26 ページ)

式のサブクエリ

サブクエリとは、カッコで囲まれた **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、条件式内でサブクエリを使用することはできません。

- **CASE**
- **IF**
- **NULLIF**
- **ARGN**
- **COALESCE**
- **ISNULL**

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;
```

参照：

- 式内の定数 (31 ページ)
- 式内のカラム名 (31 ページ)
- SQL 演算子 (33 ページ)
- IF 式 (38 ページ)
- CASE 式 (38 ページ)
- 式と定数の互換性 (40 ページ)

SQL 演算子

ここでは、SAP 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。

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。

参照：

- 式内の定数 (31 ページ)
- 式内のカラム名 (31 ページ)
- 式のサブクエリ (32 ページ)
- IF 式 (38 ページ)
- CASE 式 (38 ページ)
- 式と定数の互換性 (40 ページ)
- 比較条件 (44 ページ)
- 式 (29 ページ)
- NULL 値 (82 ページ)
- 検索条件 (43 ページ)
- 文字列 (28 ページ)
- 3 値的論理 (46 ページ)

算術演算子

SAP Sybase IQ では、次の算術演算子を使用できます。

表 1：算術演算子

演算子	説明
expression + expression	加算。いずれかの式が NULL 値の場合、結果は NULL 値になる。
expression - expression	減算。いずれかの式が NULL 値の場合、結果は NULL 値になる。
- expression	反転。式が NULL 値の場合、結果は NULL 値になる。
expression * expression	乗算。いずれかの式が NULL 値の場合、結果は NULL 値になる。
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 値になる。
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。

参照：

- 文字列演算子 (34 ページ)
- ビット処理演算子 (35 ページ)
- ジョイン演算子 (37 ページ)
- 演算子の優先度 (37 ページ)

文字列演算子

SAP Sybase IQ では、次の文字列演算子を使用できます。

表 2：文字列演算子

演算子	説明
expression ex- pression	文字列連結 (2 本の縦線)。いずれかの文字列が NULL 値の場合、連結では空文字列として扱われる。
expression + ex- pression	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ずに、必ずオペランドを文字データ型に明示設定すること。

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、非構造化データ分析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。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。|| 演算子は、ISO/ANSI SQL の文字列連結演算子です。
- Sybase - + 演算子は Adaptive Server でサポートされています。

参照：

- 算術演算子 (33 ページ)
- ビット処理演算子 (35 ページ)
- ジョイン演算子 (37 ページ)
- 演算子の優先度 (37 ページ)
- REVERSE 関数 [文字列] (329 ページ)

ビット処理演算子

SAP Sybase IQ と Adaptive Server では、位取りされていないすべての整数データ型で次のビット処理演算子を使用できます。

演算子	説明
&	AND
	OR
^	EXCLUSIVE OR
~	NOT

参照：

- 算術演算子 (33 ページ)
- 文字列演算子 (34 ページ)
- ジョイン演算子 (37 ページ)
- 演算子の優先度 (37 ページ)

AND 演算子 (&)

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。

Bit 1	Bit 2	Bit 1 & Bit 2
0	0	0
0	1	0
1	0	0
1	1	1

参照：

- ビット処理 OR (|) (35 ページ)
- EXCLUSIVE OR (^) (36 ページ)
- NOT (~) (36 ページ)

ビット処理 OR (|)

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。

Bit 1	Bit 2	Bit 1 Bit 2
0	0	0

Bit 1	Bit 2	Bit 1 Bit 2
0	1	1
1	0	1
1	1	1

参照：

- AND 演算子 (&) (35 ページ)
- EXCLUSIVE OR (^) (36 ページ)
- NOT (~) (36 ページ)

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

参照：

- AND 演算子 (&) (35 ページ)
- ビット処理 OR (|) (35 ページ)
- NOT (~) (36 ページ)

NOT (~)

NOT 演算子は、そのオペランドの逆の値を返す単行演算子です。

Bit	~ Bit
1	0
0	1

参照：

- AND 演算子 (&) (35 ページ)
- ビット処理 OR (|) (35 ページ)
- EXCLUSIVE OR (^) (36 ページ)

ジョイン演算子

SAP Sybase IQ では、**FROM** 句でテーブル式を使用する ISO/ANSI SQL 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

- モジューロ – 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 – SAP 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と、+ 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 では、|| 連結演算子をサポートしていません。

参照：

- 算術演算子 (33 ページ)
- 文字列演算子 (34 ページ)
- ビット処理演算子 (35 ページ)
- 演算子の優先度 (37 ページ)

演算子の優先度

次の推奨事項に従って、実行順序を明確に指定してください。

1 つの式に複数の演算子を使用している場合は、Adaptive Server と SAP 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します。

参照：

- 算術演算子 (33 ページ)
- 文字列演算子 (34 ページ)
- ビット処理演算子 (35 ページ)
- ジョイン演算子 (37 ページ)

IF 式

IF 式は IF-THEN-ELSE SQL 式を提供します。

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。

注意： IF 文は IF 式とは異なります。

IF 式の構文と IF 文の構文を混同しないでください。

参照：

- 式内の定数 (31 ページ)
- 式内のカラム名 (31 ページ)
- 式のサブクエリ (32 ページ)
- SQL 演算子 (33 ページ)
- CASE 式 (38 ページ)
- 式と定数の互換性 (40 ページ)

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、下記のコードでは CASE 式が **SELECT** 文の 2 番目の句として使用されています。

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

参照：

- 式内の定数 (31 ページ)
- 式内のカラム名 (31 ページ)
- 式のサブクエリ (32 ページ)
- SQL 演算子 (33 ページ)
- IF 式 (38 ページ)
- 式と定数の互換性 (40 ページ)
- NULLIF 関数 [その他] (291 ページ)
- 省略形 CASE 式の NULLIF 関数 (39 ページ)

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。1 番目の式と 2 番目の式が等しい場合、**NULLIF** は NULL を返します。1 番目の式と 2 番目の式が異なる場合、**NULLIF** は 1 番目の式を返します。

参照：

- CASE 式 (38 ページ)
- NULLIF 関数 [その他] (291 ページ)

式と定数の互換性

ここでは、Adaptive Server と SAP Sybase IQ の式と定数の互換性について説明します。

参照：

- 式内の定数 (31 ページ)
- 式内のカラム名 (31 ページ)
- 式のサブクエリ (32 ページ)
- SQL 演算子 (33 ページ)
- IF 式 (38 ページ)
- CASE 式 (38 ページ)

式の互換性

次の表は、Adaptive Server と SAP Sybase IQ の式の互換性を示したものです。

この表は、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 と SAP 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。

この表では、**expr** は式、**op** は演算子を表しています。

表 3 : Adaptive Server と SAP Sybase IQ の式の互換性

式	サポートする製品
constant	両方
column name	両方
variable name	両方
function (expr)	両方
- expr	両方

式	サポートする製品
expr op expr	両方
(expr)	両方
(subquery)	両方
if-expression	SAP Sybase IQ のみ

定数の互換性

次の表は、Adaptive Server と SAP Sybase IQ の定数の互換性を示したものです。

この表は、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 と SAP 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 4 : Adaptive Server と SAP Sybase IQ の定数の互換性

定数	サポートする製品
integer	両方
number	両方
'string'	両方
special-constant	両方
host-variable	SAP 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 と SAP 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。

SAP Sybase IQ では、アポストロフィで囲まれた文字列を定数式、引用符 (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 では、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 と SAP 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 では OFF、SAP 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 の予約語は識別子として使用できません。

Transact-SQL の **SET** 文は、Adaptive Server の大部分の接続オプションに対応していませんが、**SET** は **quoted_identifier** オプションには対応しています。

SAP Sybase IQ または Adaptive Server では、次の文を使用して **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 では、テーブル、ビュー、カラムの名前を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 では区切ることができません。

SAP Sybase IQ または Adaptive Server では、次の文を使用して **quoted_identifier** オプションの設定を OFF に変更します。

```
SET quoted_identifier OFF
```

Adaptive Server と SAP Sybase IQ の各 DBMS で **quoted_identifier** オプションが同じ値に設定されている場合は、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も使用できます。

例

quoted_identifier オプションを ON (SAP 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 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"
```

参照：

- 予約語 (19 ページ)
- 識別子 (26 ページ)

検索条件

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。

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。

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。

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。

WHERE 句、**HAVING** 句、**CHECK** 句、**JOIN** 句、または **IF** 式に対して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
| expression1 IS [ NOT ] DISTINCT FROM expression2
| 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 ] ... )
| CONTAINS ( column-name [ ,...], contains-query string )
| 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);
```

限定比較述語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：

- 比較条件 (44 ページ)
- 式 (29 ページ)
- NULL 値 (82 ページ)
- 文字列 (28 ページ)
- 3 値的論理 (46 ページ)
- SQL 演算子 (33 ページ)
- 検索条件内のサブクエリ (47 ページ)

比較条件

検索条件で比較条件を指定する場合は、比較演算子を使用します。

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。次の表は、SAP Sybase IQ で使用可能な比較演算子の一覧です。

演算子	説明
=	等価
>	より大きい
<	より小さい
>=	より大きいまたは等価
<=	以下

演算子	説明
!=	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 BirthDate <= 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すが、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次のとおりです。

- 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。

互換性

- 後続ブランク - Adaptive Server 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 SAP Sybase IQ の動作は、[文字列比較における後続ブランクの無視] データベース作成オプションによって制御されます。
- 大文字と小文字の区別 - デフォルトでは、SAP Sybase IQ データベースは Adaptive Server 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。SAP 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。

参照：

- 式 (29 ページ)
- NULL 値 (82 ページ)
- 検索条件 (43 ページ)
- 文字列 (28 ページ)
- 3 値的論理 (46 ページ)

SQL 言語の要素

- SQL 演算子 (33 ページ)
- 検索条件内のサブクエリ (47 ページ)

3 値的論理

SQL の論理演算子 AND、OR、NOT、IS は 3 値的論理で機能します。

以下の表は、3 値的論理を示しています。

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

参照：

- 比較条件 (44 ページ)
- 式 (29 ページ)
- NULL 値 (82 ページ)

- 検索条件 (43 ページ)
- 文字列 (28 ページ)
- SQL 演算子 (33 ページ)
- 検索条件内のサブクエリ (47 ページ)

検索条件内のサブクエリ

サブクエリとは、カッコで囲まれた **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つのカラムと任意の数のローを返すサブクエリは、**IN**、**ANY**、**ALL**、**EXISTS** の各検索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。

SAP Sybase IQ では、非相関サブクエリの述部でのみ **UNION** を使用できます。スカラ値サブクエリおよび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。

SAP 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  
lineitemWHERE 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。

参照：

- 式内のカラム名 (31 ページ)
- 予約語 (19 ページ)
- 識別子 (26 ページ)
- 比較条件 (44 ページ)
- 式 (29 ページ)
- NULL 値 (82 ページ)
- 検索条件 (43 ページ)
- 文字列 (28 ページ)
- 3 値的論理 (46 ページ)

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。SAP 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) サブクエリおよび非相関 (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せん。

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。通常は相関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、非相関スカラサブクエリまたは **IN** サブクエリ
- **WHERE** 句または **HAVING** 句内の相関/非相関 **EXISTS** サブクエリ
- **WHERE** 句または **HAVING** 句内の任意の相関/非相関スカラサブクエリ、**IN** または **EXISTS** サブクエリ、あるいは限定比較サブクエリ

- 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相関/相関サブクエリ述部
- 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** 文、**DELETE** 文、**SELECT INTO** 文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。

例

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');
```

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');
```

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);
```

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);
```

相関サブクエリ述部の分離

SQL 言語の要素

```
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);
```

サブクエリの分離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。

次のマージされたクエリは、関連サブクエリの述部の分離例で示したクエリと同じ結果を得ることができます。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

ALL または **ANY** 条件は検索条件のサブクエリで使用します。

ALL 条件の構文を次に示します。

```
expression compare
      ALL ( subquery )
```

compare は比較演算子です。

ANY 条件の構文を次に示します。

```
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列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。

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ず。次に例を示します。

```
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 と SAP Sybase IQ の間で互換性があります。**SOME** を **ANY** の同意語として扱うのは、SAP Sybase IQ のみです。

BETWEEN 条件

サブクエリで **BETWEEN** 条件を使用して、一定範囲内の値を取得します。

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は、*expr* が *start-expr* と *end-expr* との間であれば、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 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" のいずれか、あるいは "B" と "C" の両方をサブクエリにできます。"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、SAP Sybase IQ と Adaptive Server の間で互換性があります。

論理演算子を使用した条件

AND、**OR**、**NOT** を使用して、サブクエリの検索条件を結合します。

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。

互換性

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

CONTAINS 条件

サブクエリで **CONTAINS** 条件を使用して、テキスト一致条件を定義します。

WD インデックスのあるカラムに対する **CONTAINS** 条件の構文は、次のとおりです。

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

column-name は、ベーステーブルの **CHAR**、**VARCHAR**、または **LONG VARCHAR** (**CLOB**) カラムであり、**WD** インデックスが付いている必要があります。*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')
```

`varchar_col` の値が `The cat is on the mat` の場合は、`TRUE` になります。
`varchar_col` の値が `The cat chased the mouse` の場合、この条件は `FALSE` になります。

SAP Sybase IQ で `LIKE` と `CONTAINS` の両方を含む文を実行する場合は、`CONTAINS` 条件が優先されます。

ユーザ定義関数を含むビューでは、`CONTAINS` の基準が無視されるため、`CONTAINS` 述部を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。

`CONTAINS` 条件での `TEXT` インデックスの使用法については、『非構造化データ分析』を参照してください。

EXISTS 条件

`EXISTS` 条件は、サブクエリ結果にローが 1 つでもあれば満たされます。

`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` 条件は、`UNKNOWN` にはなりません。

互換性

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

IN 条件

サブクエリで `IN` 条件を使用すると、`OR` 条件の使用回数を減らすことができます。

`IN` 条件の構文を次に示します。

```
{ 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 と SAP Sybase IQ の間で互換性があります。

IS DISTINCT FROM 探索条件

IS DISTINCT FROM および IS NOT DISTINCT FROM 探索条件は比較演算子として使用します。

構文

```
expression1 IS [ NOT ] DISTINCT FROM expression2
```

備考

IS DISTINCT FROM および IS NOT DISTINCT FROM 探索条件は、検索引数可能であり、TRUE または FALSE と評価されます。

IS NOT DISTINCT FROM 探索条件は、*expression1* が *expression2* と等しい場合、または両方の式が NULL の場合、TRUE と評価されます。これは、次のような2つの探索条件の組み合わせと同じです。

```
expression1 = expression2 OR ( expression1 IS NULL AND expression2 IS NULL )
```

IS DISTINCT FROM 構文は、逆の意味を持ちます。つまり、IS DISTINCT FROM は、*expression1* が *expression2* と等しくなく、少なくとも一方の式が NULL ではない場合、TRUE と評価されます。これは、次の探索条件と同じです。

```
NOT( expression1 = expression2 OR ( expression1 IS NULL AND expression2 IS NULL ) )
```

標準と互換性

- **SQL/2008** – IS [NOT] DISTINCT FROM 述部は、SQL/2008 標準に定義されています。IS DISTINCT FROM 述部は、SQL/2008 標準の機能 T151 の "DISTINCT predicate" です。IS NOT DISTINCT FROM 述部は、SQL/2008 標準の機能 T152 の "DISTINCT predicate with negation" です。

IS NULL 条件

サブクエリで IS NULL 条件を使用すると、欠落している未知のデータがある場合は NULL 値が示されます。

IS NULL 条件の構文を次に示します。

```
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 と SAP Sybase IQ の間で互換性があります。

LIKE 条件

サブクエリで **LIKE** 条件を指定して、**WHERE** 句でワイルドカードを使用し、パターン一致を行います。

LIKE 条件の構文を次に示します。

```
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。

次の場合、特定の **LIKE** 述部の実行が高速化します。

- **WD** インデックスが使用可能な場合 (**LIKE** パターンの左端にスペースで区切って、またはパターンの末尾に少なくとも 1 つの語が含まれている場合)
- **NGRAM TEXT** インデックスが使用可能な場合 (*N* 文字以上の連続するワイルドカード以外の文字が **LIKE** パターンに含まれている場合)

NOT キーワードがない場合は、*expression* が *pattern* と一致すれば、条件は TRUE と評価されます。*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_'
```

文字 *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ず。*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。

サポートされているパターン

126 文字以下のパターンは、すべてサポートされています。

127 ~ 254 文字のパターンの中には、特定の条件のみサポートされるものがあります。例として、以下のサブクエリを参照してください。

255 文字以上のパターンはまったくサポートされません。

127 ~ 254 文字のパターン：例 1

パターン内に定数文字が隣接して存在する場合、長さが 127 ~ 254 文字のパターンがサポートされる場合があります。文字列パターン内の定数文字の 1 文字は、その文字が 1 バイト文字の場合でも 2 バイトを必要とします。**LIKE** 述部内の文字列パターンは、256 バイト (または 255/2 文字) 未満でなければなりません。これを超えると、次のようなエラーが表示されます。

```
There was an error reading the results of the SQL statement. The
displayed results may be incorrect or incomplete. Cannot compile Like
pattern: either bad pattern or pattern too long.
```

SAP Sybase IQ は、隣接する定数文字をまとめて 1 文字とみなします。たとえば、次のように、**LIKE** 述部に長さが 130 文字の文字列を指定した場合を考えてみましょう。

```
select col2 from tablen where col2 like
'123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890';
```

SAP Sybase IQ は、文字列の最後にある隣接する 4 つの定数文字 %%%% をまとめて 1 つの % とします。これにより、文字列の長さが 130 文字から 127 文字に短縮されます。これは最大長 256 バイト (または 255/2 文字) より短いため、エラーは生成されません。

つまり、**LIKE** 述部の文字列内で定数が隣接する場合、それらをまとめた文字列全体の長さが 256 バイト (または 255/2 文字) 未満であれば、長さが 127 ~ 254 文字のパターンはサポートされます。

127 ~ 254 文字のパターン：例 2

次の例では、130 文字の **LIKE** 述部の最後にある隣接する 4 つの定数文字 %%%% を、定数文字 7890 に置き換えます。

```
select col2 from tablen where col2 like
'123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890';
```

この場合、文字がまとめられることはありません。文字列の長さは 130 文字のままであるため、SAP Sybase IQ はエラーを生成します。

127 ~ 254 文字のパターン：例 3

次の例では、130 文字の **LIKE** 述部の最後にある 4 つの定数文字 %%% を、隣接する 4 つのアンダースコア _____ (特殊文字) に置き換えます。

```
select col2 from tablen where col2 like
'1234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456_____';
```

SAP Sybase IQ では、隣接する特殊文字がまとめられることはありません。文字列の長さは 130 文字のままであるため、SAP Sybase IQ はエラーを生成します。

127 ~ 254 文字のパターン：例 4

次の例では、130 文字の **LIKE** 述部の最後にある 4 つの定数文字 %%% を、範囲を表す [1-3] に置き換えます。

```
select col2 from tablen where col2 like
'1234567890123456789012345678901234567890123456789012345678901234567890123456
7890123456789012345678901234567890123456789012345678901234567890123456[1-3]';
```

LIKE 述部の長さのバイト数は次のように計算されます。126 (定数文字) * 2 + 1 (角カッコ内の 1) + 1 (角カッコ内の 3) + 2 (セットと範囲の式)。

このように計算すると 256 バイトとなるので、SAP Sybase IQ はエラーを生成しません。

文字セットからの検索

検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'
```

範囲内の文字の検索

検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。照合での文字の順序については、『管理：ローバリゼーション』の「照合を使用した文字のソート」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検索

角カッコ内で、文字範囲と文字セットを結合できます。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、あるいは s から z の範囲内の 1 文字」と解釈されます。

範囲外の 1 文字の検索

脱字記号 (^) は、検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] は文字 a にのみ一致します。[^] は脱字記号にのみ、[%] はパーセント記号にのみ (この場合、パーセント記号はワイルドカードとして機能しません)、[_] はアンダースコア文字にのみ一致します。また、[/] は文字 / にのみ一致します。

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] は、a、%、または b のいずれかを検出します。

- 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 $[a^b]$ は a 、 $^$ 、または b のいずれかを検出します。

互換性

ESCAPE 句をサポートしているのは SAP Sybase IQ のみです。

注意： **LIKE** 述部でのラージオブジェクトデータおよび変数のサポートについては、『非構造化データ分析』の「非構造化データのクエリ」を参照してください。

ラージオブジェクトデータ型 LONG BINARY と LONG VARCHAR を使用するには、専用のライセンスを取得しておく必要があります。非構造化データ分析オプションの詳細については、『非構造化データ分析』を参照してください。

参照：

- PATINDEX 関数 [文字列] (296 ページ)
- LOCATE 関数 [文字列] (268 ページ)

NOT 条件

NOT 条件は TRUE、FALSE、または UNKNOWN として評価できます。

NOT 条件の構文を次に示します。

```
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

条件の真理値は TRUE または FALSE です。

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* が指定された *truth-value* (TRUE、FALSE、UNKNOWN のいずれか) と評価されれば、検索条件は 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、SAP Sybase IQ のみです。

ユーザ指定の条件ヒント

条件の選択性とは、テーブル内の条件を満たすローの割合のことです。

SAP 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、条件を満たすテーブルローのパーセンテージとして表されません。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せん。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 では、ユーザ指定の選択性的見積もりがサポートされています。

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

参照：

- ユーザ指定の条件ヒント文字列 (61 ページ)
- ジョイン等号条件に関するユーザ指定のヒント (67 ページ)
- ユーザ指定の条件ヒントの使用ガイドライン (69 ページ)
- 選択性ヒント (62 ページ)

ユーザ指定の条件ヒント文字列

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的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、それにコロン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。

参照：

- ユーザ指定の条件の選択性 (60 ページ)
- ジョイン等号条件に関するユーザ指定のヒント (67 ページ)
- ユーザ指定の条件ヒントの使用ガイドライン (69 ページ)

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、条件を満たすテーブルローのパーセンテージとして表されます。

例

次の例は、選択性の項で示した 2 番目のユーザ指定条件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
```

参照：

- インデックス設定ヒント (62 ページ)
- INDEX_PREFERENCE オプション (63 ページ)
- 実行フェーズヒント (65 ページ)
- 有用性ヒント (66 ページ)
- ユーザ指定の条件の選択性 (60 ページ)

インデックス設定ヒント

サポートされている 2 番目のヒントタイプは、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。

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
```

参照：

- 選択性ヒント (62 ページ)
- INDEX_PREFERENCE オプション (63 ページ)
- 実行フェーズヒント (65 ページ)
- 有用性ヒント (66 ページ)

INDEX_PREFERENCE オプション

クエリ処理に使用するインデックスの選択を制御します。

指定できる値

値	アクション
0	オプティマイザの選択に従う。
1	LF インデックスを優先する。
2	HG インデックスを優先する。
3	HNG インデックスを優先する。
4	CMP インデックスを優先する。
5	デフォルトのインデックスを優先する。
6	WD インデックスを優先する。
8	DATE インデックスを優先する。
9	TIME インデックスを優先する。
10	DTTM インデックスを優先する。
-1	LF インデックスを回避する。
-2	HG インデックスを回避する。

値	アクション
-3	HNG インデックスを回避する。
-4	CMP インデックスを回避する。
-5	デフォルトのインデックスを回避する。
-6	WD インデックスを回避する。
-8	DATE インデックスを回避する。
-9	TIME インデックスを回避する。
-10	DTTM インデックスを回避する。

デフォルト値

0

スコープ

オプションは、データベース (PUBLIC) レベルまたは ユーザレベルで設定できます。データベースレベルで設定した場合、値は新しいユーザのデフォルト値になりますが、既存のユーザには影響を与えません。ユーザレベルで設定した場合は、そのユーザの PUBLIC 値のみが上書きされます。自分のオプションを設定する場合は、システム権限は必要ありません。自分以外のユーザのオプションをデータベースレベルまたはユーザレベルで設定する場合は、システム権限が必要です。

このオプションを設定するには、SET ANY PUBLIC OPTION システム権限が必要です。個々の接続または PUBLIC ロールに一時的に設定できます。すぐに有効になります。

備考

通常、SAP Sybase IQ オプティマイザは、ローカルな WHERE 句の述部の処理など、1つの IQ インデックスの範囲内で実行できる操作を行うために、使用可能な最適なインデックスを選択します。INDEX_PREFERENCE は、テスト目的でオプティマイザの選択を無効にするために使用します。通常の使用では、このオプションの値を変更しないでください。

参照：

- 選択性ヒント (62 ページ)
- インデックス設定ヒント (62 ページ)
- 実行フェーズヒント (65 ページ)
- 有用性ヒント (66 ページ)

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。

SAP Sybase IQ クエリエンジン内には、条件を評価できる次の実行フェーズがあります。

- 不変
- 遅延
- バインド
- 水平

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** – 単一の列 (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
```

参照：

- 選択性ヒント (62 ページ)
- インデックス設定ヒント (62 ページ)
- INDEX_PREFERENCE オプション (63 ページ)
- 有用性ヒント (66 ページ)

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 (o.SalesRepresentative > 10000.0, 'U: 3.25, E: D')
      AND c.id = o.CustomerID
```

互換性

SQL Anywhere 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 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。

参照：

- 選択性ヒント (62 ページ)
- インデックス設定ヒント (62 ページ)
- INDEX_PREFERENCE オプション (63 ページ)
- 実行フェーズヒント (65 ページ)

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。

例

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')
```

表 5 : JOIN_PREFERENCE 値

値	アクション
0	オプティマイザの選択に従う。
1	ソート／マージを優先する。
2	ネストされたループを優先する。
3	ネストされたループのプッシュダウンを優先する。
4	ハッシュを優先する。
5	ハッシュプッシュダウンを優先する。
6	非対称ソート／マージのジョインを優先する。
7	ソート／マージのプッシュダウンを優先する。
8	非対称ソート／マージのプッシュダウンのジョインを優先する。
9	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ハッシュのジョインを優先する。
10	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ハッシュプッシュダウンのジョインを優先する。

値	アクション
11	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ソート／マージのジョインを優先する。
12	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ソート／マージプッシュダウンのジョインを優先する。
-1	ソート／マージを回避する。
-2	ネストされたループを回避する。
-3	ネストされたループのプッシュダウンを回避する。
-4	ハッシュを回避する。
-5	ハッシュプッシュダウンを回避する。
-6	非対称ソート／マージのジョインを回避する。
-7	ソート／マージのプッシュダウンを回避する。
-8	非対称ソート／マージのプッシュダウンのジョインを回避する。
-9	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ハッシュのジョインを回避する。
10	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ハッシュプッシュダウンのジョインを回避する。
11	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ソート／マージのジョインを回避する。
12	ジョインキーにハッシュ分割テーブルの全分割キーが含まれる場合、分割ソート／マージプッシュダウンのジョインを回避する。

参照先

『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「JOIN_PREFERENCE オプション」

参照：

- ユーザ指定の条件の選択性 (60 ページ)
- ユーザ指定の条件ヒント文字列 (61 ページ)
- ユーザ指定の条件ヒントの使用ガイドライン (69 ページ)

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、オプティマイザがオプションの決定を行います。

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。

参照：

- ユーザ指定の条件の選択性 (60 ページ)
- ユーザ指定の条件ヒント文字列 (61 ページ)
- ジョイン等号条件に関するユーザ指定のヒント (67 ページ)

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。

参照：

- 式 (29 ページ)

CURRENT DATABASE 特別値

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

CURRENT DATE 特別値

CURRENT DATE は、現在の年、月、日を返します。

データ型
DATE

参照：

- **TIMESTAMP** 特別値 (72 ページ)
- **CURRENT TIMESTAMP** 特別値 (70 ページ)

SQL 言語の要素

- CURRENT TIME 特別値 (70 ページ)
- 日付と時刻のデータ型 (405 ページ)
- 日付と時刻の取得 (408 ページ)

CURRENT PUBLISHER 特別値

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でデフォルト値として使用できます。

CURRENT TIME 特別値

CURRENT TIME は、現在の時、分、秒 (小数位あり) で構成される時刻を返します。

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：

- TIMESTAMP 特別値 (72 ページ)
- CURRENT TIMESTAMP 特別値 (70 ページ)
- CURRENT DATE 特別値 (69 ページ)
- 日付と時刻のデータ型 (405 ページ)
- 日付と時刻の取得 (408 ページ)

CURRENT TIMESTAMP 特別値

CURRENT DATE と **CURRENT TIME** を結合して形成された **TIMESTAMP** 値です。年、月、日、時、分、秒、秒の小数位で構成されます。

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：

- **TIMESTAMP** 特別値 (72 ページ)
- **CURRENT TIME** 特別値 (70 ページ)
- **CURRENT DATE** 特別値 (69 ページ)
- 日付と時刻のデータ型 (405 ページ)
- 日付と時刻の取得 (408 ページ)
- **CURRENT USER** 特別値 (71 ページ)
- **LAST USER** 特別値 (71 ページ)
- **USER** 特別値 (73 ページ)

CURRENT USER 特別値

CURRENT USER は、現在の接続のユーザ ID を含む文字列を返します。

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。

データ型
STRING

CURRENT USER は文字データ型のカラムでデフォルト値として使用できます。

参照：

- **CURRENT TIMESTAMP** 特別値 (70 ページ)
- **LAST USER** 特別値 (71 ページ)
- **USER** 特別値 (73 ページ)

LAST USER 特別値

LAST USER は、ローを最後に更新したユーザの名前を返します。

INSERT と **LOAD** の場合、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ていない場合、現在のユーザ名に変更されます。

LAST USER のデフォルト値を **DEFAULT TIMESTAMP** と組み合わせると、ローを最後に変更したユーザと日時の両方を (別々カラムに) 記録できます。

データ型
STRING

LAST USER は、文字データ型のカラムでデフォルト値として使用できます。

参照：

- **CURRENT USER** 特別値 (71 ページ)

SQL 言語の要素

- CURRENT TIMESTAMP 特別値 (70 ページ)
- USER 特別値 (73 ページ)

SQLCODE 特別値

SQLCODE は現在の **SQLCODE** 値を返します。

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できます。

データ型
STRING

SQLSTATE 特別値

SQLSTATE は現在の **SQLSTATE** 値を返します。

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できます。

データ型
STRING

TIMESTAMP 特別値

TIMESTAMP は、テーブルの各ローが最後に修正された日時を示します。

DEFAULT TIMESTAMP 型としてカラムを宣言すると、挿入処理およびロード処理に対して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 と **LOAD** の場合、**DEFAULT TIMESTAMP** は **CURRENT TIMESTAMP** と同じ効果があります。**UPDATE** の場合、**TIMESTAMP** のデフォルト値を持つカラムは、明示的に変更されないかぎり、その値が現在の日時に変更されます。

注意： SAP Sybase IQ は、**UTC TIMESTAMP** または **CURRENT UTC TIMESTAMP** の **DEFAULT** 値をサポートしていません。また、データベースオプション **DEFAULT_TIMESTAMP_INCREMENT** もサポートしていません。データ型 **UTC TIMESTAMP** または **CURRENT UTC TIMESTAMP** のカラムの **DEFAULT** 値を挿入または更新しようとするたびに、SAP Sybase IQ はエラーを生成します。

データ型
TIMESTAMP

参照：

- CURRENT TIMESTAMP 特別値 (70 ページ)

- CURRENT TIME 特別値 (70 ページ)
- CURRENT DATE 特別値 (69 ページ)
- 日付と時刻のデータ型 (405 ページ)
- 日付と時刻の取得 (408 ページ)

USER 特別値

USER は、現在の接続のユーザ ID を含む文字列を返します。

UPDATE では、**USER** のデフォルト値を含むカラムは変更されません。

データ型
STRING

USER は、文字データ型のカラムでデフォルト値として使用できます。

参照：

- CURRENT USER 特別値 (71 ページ)
- CURRENT TIMESTAMP 特別値 (70 ページ)
- LAST USER 特別値 (71 ページ)

変数

SAP Sybase IQ は、ローカル変数、接続レベル変数、グローバル変数をサポートしています。

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

ローカル変数はユーザが宣言します。この変数を 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。

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) の中でのみ使用できます。変数の初期設定値は NULL です。変数の値は、**SET** 文によって設定するか、**INTO** 句のある **SELECT** 文で割り当てることができます。

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。

```
-- This batch is invalid.
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 と SAP 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 では、すべての変数名の先頭にアットマーク (@) を付ける必要があります。SAP Sybase IQ では、@ プレフィ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 - SAP Sybase IQ と Adaptive Server では、ローカル変数のスコープが異なります。SAP 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 - SAP Sybase IQ では、各 **DECLARE** 文で宣言できる変数は 1 つのみです。Adaptive Server では、1 つの文で複数の変数を宣言できます。

接続レベル変数

接続レベル変数はユーザが宣言します。この変数を 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。

接続レベル変数は、**CREATE VARIABLE** 文で宣言します。**CREATE VARIABLE** 文は、複合文内を除き、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 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、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。

グローバル変数の値は SAP Sybase IQ によって設定されます。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 つのアット記号 (@) 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

SQL 言語の要素

ローカル変数です。ユーザは、グローバル変数を定義したり、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

次の表は、SAP Sybase IQ で使用可能なグローバル変数の一覧です。

変数名	意味
@@error	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、if @@error != 0 return のような文によって終了する。すべての SQL 文は @@error 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。
@@fetch_status	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。@@fetch_status には、次のいずれかの値が含まれる。 <ul style="list-style-type: none">0: フェッチ文は正常終了した。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 @@sqlstatus と同様。これは Microsoft SQL Server との互換性のため。

変数名	意味
<i>@@identity</i>	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、 <i>@@identity</i> は必ずリセットされる。文によって複数のローが挿入された場合、 <i>@@identity</i> 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、 <i>@@identity</i> 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、 <i>@@identity</i> の値には影響しない。 <i>@@identity</i> 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。
<i>@@isolation</i>	現在の独立性レベル。 <i>@@isolation</i> には、アクティブレベルの値が入る。
<i>@@procid</i>	現在実行中のプロシージャのストアドプロシージャ ID。
<i>@@rowcount</i>	最後の文の影響を受けるローの数。 <i>@@rowcount</i> の値は、文の直後にチェックする必要があります。挿入、更新、削除を実行すると、影響を受けたローの数が <i>@@rowcount</i> に設定されます。カーソルを使用する場合、 <i>@@rowcount</i> は、最後のフェッチ要求までにカーソルの結果セットからクライアントに返されたローの累積数を表します。 <i>@@rowcount</i> が、IF 文のようにローに影響を及ぼさない文によって 0 にリセットされることはありません。
<i>@@servername</i>	現在のデータベースサーバの名前。
<i>@@sqlstatus</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>@@version</i>	現在の SAP Sybase IQ のバージョン番号。

SAP Sybase IQ でサポートされている Adaptive Server グローバル変数

次の表は、SAP Sybase IQ でサポートされているすべての Adaptive Server グローバル変数を示したものです。リストにない Adaptive Server グローバル変数は、SAP Sybase IQ ではサポートされていません。

この表に記載されている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 6 : SAP Sybase IQ でサポートされている Adaptive Server グローバル変数

グローバル変数	戻り値
<i>@@char_convert</i>	0 を返す。

グローバル変数	戻り値
@@client_csname	Adaptive Server 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。SAP Sybase IQ では NULL が返される。
@@client_csid	Adaptive Server 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、syscharset から直前に使用されたクライアントの文字セット ID が入る。SAP Sybase IQ では -1 が返される。
@@connections	最後にサーバが起動されてからのログイン数。
@@cpu_busy	Adaptive Server では、最後に Adaptive Server が起動されてから CPU が Adaptive Server の作業に費やした時間数 (チック単位)。SAP Sybase IQ では 0 が返される。
@@error	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 テストなどを含め、すべての SQL 文は @@error をリセットしてしまうので、成否の確認が必要な文については、その直後にステータスチェックを行う必要がある。
@@identity	Adaptive Server では、INSERT 文、LOAD 文、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たびに、@@identity はリセットされる。文によって複数のローが挿入された場合、@@identity 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、@@identity 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、@@identity の値には影響しない。@@identity 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。
@@idle	Adaptive Server では、最後にサーバが起動されてから Adaptive Server がアイドル状態になった時間数 (チック単位)。SAP Sybase IQ では 0 が返される。
@@io_busy	Adaptive Server では、最後にサーバが起動されてから Adaptive Server が入出力処理に費やした時間数 (チック単位)。SAP 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 では、 <code>@@isolation</code> にはアクティブレベルの値が入る。
@@langid	Adaptive Server では、現在使用中の言語のローカル言語 ID が定義される。SAP Sybase IQ では 0 が返される。
@@language	Adaptive Server では、現在使用中の言語の名前が定義される。SAP Sybase IQ では "English" が返される。
@@maxcharlen	Adaptive Server では、Adaptive Server のデフォルト文字セット内の 1 文字の最大長 (バイト単位)。SAP 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、データベース接続とは異なる)。Adaptive Server の場合は、サーバへの最大接続数。
@@ncharsize	Adaptive Server では、各国語 1 文字の平均の長さ (バイト単位)。SAP Sybase IQ では 1 が返される。
@@nestlevel	Adaptive Server では、現在の実行のネストレベル (初期値 0)。ストアードプロシージャまたはトリガが別のストアードプロシージャやトリガを呼び出すたびに、ネストレベルは増加する。SAP Sybase IQ では -1 が返される。
@@pack_received	Adaptive Server では、最後にサーバが起動されてから Adaptive Server が読み込んだ入力パケットの数。SAP Sybase IQ では 0 が返される。
@@pack_sent	Adaptive Server では、最後にサーバが起動されてから Adaptive Server が書き込んだ出力パケットの数。SAP Sybase IQ では 0 が返される。
@@packet_errors	Adaptive Server では、Adaptive Server がパケットを送受信中に発生したエラーの数。SAP 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ードプロシージャ ID。
@@servername	ローカル Adaptive Server サーバまたはローカル SAP Sybase IQ サーバの名前。
@@spid	Adaptive Server では、現在のプロセスのサーバプロセス ID 番号。SAP Sybase IQ では、現在の接続の接続ハンドル。これは、 <code>sa_conn_info</code> 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
@@sqlstatus	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。 @@sqlstatus は次の値を含む。 <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 – 結果セットには値が残されていない。
@@thresh_hysteresis	Adaptive Server 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。SAP Sybase IQ では 0 が返される。
@@timeticks	Adaptive Server では、1 チックあたりのマイクロ秒数。1 チックの長さはマシンによって異なる。SAP Sybase IQ では 0 が返される。
@@total_errors	Adaptive Server では、Adaptive Server が読み込み中または書き込み中に発生したエラーの数。SAP Sybase IQ では 0 が返される。
@@total_read	Adaptive Server では、最後にサーバが起動されてから Adaptive Server が行ったディスク読み込みの回数。SAP Sybase IQ では 0 が返される。
@@total_write	Adaptive Server では、最後にサーバが起動されてから Adaptive Server が行ったディスク書き込みの回数。SAP Sybase IQ では 0 が返される。
@@tranchained	Transact-SQL プログラムの現在のトランザクションモード。 @@tranchained 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。
@@trancount	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。
@@transtate	Adaptive Server では、文の実行後のトランザクションの現在の状態。SAP Sybase IQ では -1 が返される。
@@version	Adaptive Server または SAP 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。

SAP Sybase IQ では、次のコメントインジケータを使用できます。

コメントインジケータ	説明
-- (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 - 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る。% をコメントインジケータとして使用しないことが推奨される。

注意： -- (二重ハイフン) と /* (スラッシュ - アスタリスク) のコメントスタイルは、Adaptive Server 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 concatenates the firstname and lastname
-- arguments with a single space between.
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。

```
/*
    Lists the names and employee IDs of employees
    who work in the sales department.
*/
CREATE VIEW SalesEmployee AS
SELECT emp_id, emp_lname, emp_fname
FROM "GROUP0".Employees
WHERE DepartmentID = 200
```

NULL 値

NULL 値を使用して、未知、欠落、適用不可の値を指定します。

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
欠落	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。

NULL 値によって、SQL に 3 値的論理の概念が導入されました。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 述語です。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、UNKNOWN のいずれかです)。

次の例では、カラム salary に NULL 値が含まれています。

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
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)** 関数を使用する必要があります。

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。検索条件を組み合わせる場合は 3 値的論理の影響に注意してください。

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

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' )
```

参照：

- 比較条件 (44 ページ)
- 式 (29 ページ)
- 検索条件 (43 ページ)
- 文字列 (28 ページ)
- 3 値的論理 (46 ページ)
- SQL 演算子 (33 ページ)
- 検索条件内のサブクエリ (47 ページ)

SQL 関数

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。

SAP Sybase IQ で関数を使用する場合は、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 SAP Sybase IQ ではなく SQL Anywhere によって処理されます。このため、特に構文およびセマンティック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。

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 SAP Sybase IQ で処理させることができます。この iq_dummy は、ユーザが自身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。

参照：

- その他の関数 (423 ページ)

集合関数

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** 句を使用してグループを作成します。

使用法

単純な集合関数 (**SUM()**、**MIN()**、**MAX()**、**AVG()**、**COUNT()** など) は、**SELECT** 文の select リストの中および **HAVING** 句や **ORDER BY** 句の中だけで使用できます。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** 句を使用し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。

- 単純な集合関数 (**AVG()**、**COUNT()**、**MAX()**、**MIN()**、**SUM()** など) 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** 句を使用し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()** などがあります。

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 **<window clause>** (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できます。

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せん。単純な集合関数および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*) と併用できます。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。

- 時系列分析用として、次のような ISO/ANSI SQL:2008 OLAP 関数があります。**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()**。
- データベース業界で使用される ISO/ANSI SQL:2008 以外の 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。

金融時系列の予測と分析専用設計された時系列関数は、名前が "TS_" で始まります。

OLAP 関数の使用方法の詳細については、『プログラミング』の「付録：OLAP の使用」を参照してください。

集合関数による LONG BINARY データ型と LONG VARCHAR データ型のサポートについては、『非構造化データ分析』を参照してください。

集合関数 **AVG**、**SUM**、**STDDEV**、**VARIANCE** は、バイナリデータ型 (BINARY と VARBINARY) をサポートしていません。

参照：

- 分析関数 (87 ページ)
- 集合関数 (85 ページ)
- AVG 関数 [集合] (162 ページ)
- CORR 関数 [集合] (182 ページ)
- COVAR_POP 関数 [集合] (185 ページ)
- COVAR_SAMP 関数 [集合] (186 ページ)
- COUNT 関数 [集合] (187 ページ)
- CUME_DIST 関数 [統計] (188 ページ)

- EXP_WEIGHTED_AVG 関数 [集合] (223 ページ)
- FIRST_VALUE 関数 [集合] (225 ページ)
- LAST_VALUE 関数 [集合] (256 ページ)
- LIST 関数 [集合] (264 ページ)
- MAX 関数 [集合] (273 ページ)
- MEDIAN 関数 [集合] (274 ページ)
- MIN 関数 [集合] (276 ページ)
- REGR_AVGX 関数 [集合] (312 ページ)
- REGR_AVGY 関数 [集合] (313 ページ)
- REGR_COUNT 関数 [集合] (315 ページ)
- REGR_INTERCEPT 関数 [集合] (316 ページ)
- REGR_R2 関数 [集合] (317 ページ)
- REGR_SLOPE 関数 [集合] (318 ページ)
- REGR_SXX 関数 [集合] (320 ページ)
- REGR_SXY 関数 [集合] (321 ページ)
- REGR_SYY 関数 [集合] (322 ページ)
- STDDEV 関数 [集合] (352 ページ)
- STDDEV_POP 関数 [集合] (354 ページ)
- STDDEV_SAMP 関数 [集合] (355 ページ)
- SUM 関数 [集合] (365 ページ)
- VAR_POP 関数 [集合] (375 ページ)
- VAR_SAMP 関数 [集合] (377 ページ)
- VARIANCE 関数 [集合] (378 ページ)
- WEIGHTED_AVG 関数 [集合] (381 ページ)

分析関数

分析関数として、単純な集合関数、ウィンドウ関数、数値関数があります。

- 単純な集合関数 — **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
 - ウィンドウ集合関数 — **AVG**、**COUNT**、**MAX**、**MIN**、**SUM**。

- ランク付け関数 — **RANK**、**DENSE_RANK**、**PERCENT_RANK**、**ROW_NUMBER**、**NTILE**。
- 統計関数 — **STDDEV**、**STDDEV_SAMP**、**STDDEV_POP**、**VARIANCE**、**VAR_SAMP**、**VAR_POP**。
- 分散統計関数 — **PERCENTILE_CONT**、**PERCENTILE_DISC**。
- Interrow 関数 — **LAG**、**LEAD**。
- 数値関数 — **WIDTH_BUCKET**、**CEIL**、**LN**、**EXP**、**POWER**、**SQRT**、**FLOOR**。

注意： ランク付け統計関数と逆分散統計関数は、Adaptive Server ではサポートされません。

一部の集合関数とは異なり、ウィンドウ関数で **DISTINCT** は指定できません。

* OLAP SQL 標準では、Grouping() は **GROUP BYCUBE** または **GROUP BY ROLLUP** オペレーションでのみ使用できます。

参照：

- 集合関数 (85 ページ)
- AVG 関数 [集合] (162 ページ)
- COUNT 関数 [集合] (187 ページ)
- DENSE_RANK 関数 [統計] (214 ページ)
- GROUPING 関数 [集合] (231 ページ)
- MAX 関数 [集合] (273 ページ)
- MIN 関数 [集合] (276 ページ)
- NTILE 関数 [統計] (289 ページ)
- PERCENT_RANK 関数 [統計] (298 ページ)
- PERCENTILE_CONT 関数 [統計] (299 ページ)
- PERCENTILE_DISC 関数 [統計] (302 ページ)
- RANK 関数 [統計] (311 ページ)
- ROW_NUMBER 関数 [統計] (332 ページ)
- STDDEV 関数 [集合] (352 ページ)
- STDDEV_POP 関数 [集合] (354 ページ)
- STDDEV_SAMP 関数 [集合] (355 ページ)
- SUM 関数 [集合] (365 ページ)
- VAR_POP 関数 [集合] (375 ページ)
- VAR_SAMP 関数 [集合] (377 ページ)
- VARIANCE 関数 [集合] (378 ページ)

ウィンドウ集合関数の使用法

ISO/ANSI SQL 拡張で導入された OLAP 用の主な機能として、「ウィンドウ」という名前の構成体があります。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計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つ以上のカラムで定義されている、クエリから返されるローのサブセットです。

```
OVER (PARTITION BY col1, col2...)
```

分析関数による LONG BINARY データ型および LONG VARCHAR データ型のサポートについては、『非構造化データ分析』の「関数のサポート」を参照してください。

参照：

- CORR 関数 [集合] (182 ページ)
- COUNT 関数 [集合] (187 ページ)
- EXP_WEIGHTED_AVG 関数 [集合] (223 ページ)
- FIRST_VALUE 関数 [集合] (225 ページ)
- GROUPING 関数 [集合] (231 ページ)
- LAST_VALUE 関数 [集合] (256 ページ)
- MAX 関数 [集合] (273 ページ)
- MEDIAN 関数 [集合] (274 ページ)
- MIN 関数 [集合] (276 ページ)
- REGR_AVGX 関数 [集合] (312 ページ)
- REGR_COUNT 関数 [集合] (315 ページ)
- REGR_INTERCEPT 関数 [集合] (316 ページ)
- REGR_R2 関数 [集合] (317 ページ)

- REGR_SLOPE 関数 [集合] (318 ページ)
- REGR_SXX 関数 [集合] (320 ページ)
- REGR_SXY 関数 [集合] (321 ページ)
- REGR_SYY 関数 [集合] (322 ページ)
- STDDEV 関数 [集合] (352 ページ)
- SUM 関数 [集合] (365 ページ)
- STDDEV_POP 関数 [集合] (354 ページ)
- STDDEV_SAMP 関数 [集合] (355 ページ)
- VAR_POP 関数 [集合] (375 ページ)
- VAR_SAMP 関数 [集合] (377 ページ)
- VARIANCE 関数 [集合] (378 ページ)
- WEIGHTED_AVG 関数 [集合] (381 ページ)

ランク付け関数の使用法

OLAP ランク付け関数を使用すると、アプリケーション開発者は、「今年度出荷した製品の中で総売り上げが上位 10 位の製品名」や「15 社以上から受注した営業部員の上位 5%」などの情報を取得するクエリを、単一の SQL 文で作成することができます。

このようなランク付け関数には、**RANK()**、**DENSE_RANK()**、**PERCENT_RANK()**、**ROW_NUMBER()**、**NTILE()** があります。

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()**、**ROW_NUMBER()**、**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と、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** 句内でのみ使用するもので、**SELECT** 文の **ORDER BY** とは異なります。ランク付けクエリ **ROW** の集合関数に **DISTINCT** を指定することはできません。

注意： **ROW_NUMBER()** 関数の **OVER (ORDER BY)** 句に **ROWS** や **RANGE** 句を含めることはできません。

OVER 句は、関数がクエリの結果セットに対して処理を行うことを示します。結果セットは、**FROM**、**WHERE**、**GROUP BY**、**HAVING** の各句がすべて評価された後で返されるローです。**OVER** 句は、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は、**SELECT** 文や **INSERT** 文の **SELECT** リスト、または **SELECT** 文の **ORDER BY** 句のみで使用できます。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。サブクエリ、**HAVING** 句、**UPDATE** 文や **DELETE** 文の **SELECT** リストでランク付け関数を使用することはできません。SAP Sybase IQ 16.0 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

統計集合分析関数は、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** 句を使用してグループを作成します。集合関数は、**SELECT** 文の select リストおよび **HAVING** 句と **ORDER BY** 句でのみ使用できます。これらの関数としては、**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 (**PERCENTILE_CONT** と **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** 句で指定されている **ORDER BY** 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" エラーが返されます。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" エラーが返されます。

必須の **ORDER BY** 句には、パーセンタイル関数の実行対象となる式と、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** 句の内部のみで使用するもので、**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のであれば、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。

Interrow 関数の使用法

Interrow 関数 **LAG** と **LEAD** は、一連のデータ内で前の値または後ろの値にアクセスすることを可能にします。

これらの関数は、セルフジョインなしで、テーブルまたはパーティションの複数のローに同時にアクセスできます。**LAG** 関数は、テーブルまたはパーティション内の **CURRENT ROW** から特定の物理的オフセット分だけ前にあるローにアクセスします。**LEAD** 関数は、テーブルまたはパーティション内の **CURRENT ROW** から特定の物理的オフセット分だけ後ろにあるローにアクセスします。**LAG** 関数と **LEAD** 関数を使用して、「現在のローよりインターバル 2 つ前の株価」や、「現在のローよりインターバル 1 つ後ろの株価」といったクエリを作成できます。

Interrow 関数は **OVER (ORDER_BY)** 句を必要とします。

Interrow 関数の使用法

Interrow 関数 **LAG** および **LEAD** を使用すると、一連のデータ内では前後の値にアクセスでき、テーブル内では複数のローにアクセスできます。

また、Interrow 関数は、セルフジョインなしで同時にパーティション分けします。**LAG** では、テーブル内またはパーティション内の **CURRENT ROW** から特定の物理的オフセット分だけ前にあるローにアクセスできます。**LEAD** では、テーブル内またはパーティション内の **CURRENT ROW** から特定の物理的オフセット分だけ後ろにあるローにアクセスできます。

LAG と **LEAD** の構文は同じです。どちらの関数にも **OVER (ORDER_BY)** ウィンドウ指定が必要です。例を示します。

```
LAG (value_expr) [, offset [, default]] OVER ([PARTITION BY
      window partition] ORDER BY
      window ordering)
```

および

```
LEAD (value_expr) [, offset [, default]] OVER ([PARTITION BY
      window partition] ORDER BY
      window ordering)
```

OVER (ORDER BY) 句内の **PARTITION BY** 句はオプションです。 **OVER (ORDER BY)** 句に、ウィンドウフレーム **ROWS/RANGE** 指定を含めることはできません。

value_expr は、テーブルから返すオフセットデータを定義するテーブルカラムまたは式です。 *value_expr* では他の関数を定義できます。ただし、分析関数を除きます。

両方の関数について、物理的なオフセットを入力してターゲットローを指定します。 *offset* 値は、現在のローより上または下のロー数です。負でない数値式を入力してください(負の値を入力するとエラーが生成されます)。0を入力すると、SAP Sybase IQ から現在のローが返されます。

オプションの *default* 値は、 *offset* 値がテーブルのスコープを超える場合に返される値を定義します。 *default* のデフォルト値は **NULL** です。 *default* のデータ型は、 *value_expr* 値のデータ型に暗黙的に変換可能である必要があります。変換可能でない場合、SAP Sybase IQ は変換エラーを生成します。

LAG の例 1 - Interrow 関数は、証券取引などのデータストリームに対して計算を実行する金融サービスアプリケーションで使用すると有用です。この例では、 **LAG** 関数を使用して、特定の株式取引価格の変化率を計算しています。次の `stock_trades` という架空のテーブルの取引データについて考えてみます。

traded at	symbol	price
2009-07-13 06:07:12	SQL	15.84
2009-07-13 06:07:13	TST	5.75
2009-07-13 06:07:14	TST	5.80
2009-07-13 06:07:15	SQL	15.86
2009-07-13 06:07:16	TST	5.90
2009-07-13 06:07:17	SQL	15.86

注意： 架空の `stock_trades` テーブルは、 `iqdemo` データベースには含まれていません。

このクエリでは、銘柄記号ごとに取引をパーティションに分け、取引時間に基づいて順序付けします。また、 **LAG** 関数を使用して、現在の取引価格と前の取引価格を比較し、その増加率または減少率を計算します。

```
select stock_symbol as 'Stock',
       traded_at     as 'Date/Time of Trade',
```

```

trade_price as 'Price/Share',
cast ( ( ( trade_price
- (lag(trade_price, 1)
over (partition by stock_symbol
order by traded_at)))
/ trade_price)
* 100.0) as numeric(5, 2) )
as '% Price Change vs Previous Price'
from stock_trades
order by 1, 2

```

このクエリは次の結果を返します。

Stock symbol	Date/Time of Trade	Price/Share	% Price Change vs Previous Price
SQL	2009-07-13 06:07:12	15.84	NULL
SQL	2009-07-13 06:07:15	15.86	0.13
SQL	2009-07-13 06:07:17	15.86	0.00
TST	2009-07-13 06:07:13	5.75	NULL
TST	2009-07-13 06:07:14	5.80	0.87
TST	2009-07-13 06:07:16	5.90	1.72

1 番目と 4 番目の出力ローの NULL は、**LAG** 関数が、2 つの各パーティションの最初のローについてはスコープ外であることを示します。比較対象となる前のローがないため、*default* 変数で指定されているように、SAP Sybase IQ は NULL を返します。

データ型変換関数

データ型変換関数は、引数にあるデータ型から別のデータ型に変換します。

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。

参照：

- データ型変換 (415 ページ)
- 記憶領域サイズ (401 ページ)
- BIGINTTOHEX 関数 [データ型変換] (164 ページ)
- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HEXTOBIGINT 関数 [データ型変換] (232 ページ)
- HEXTOINT 関数 [データ型変換] (234 ページ)
- INTTOHEX 関数 [データ型変換] (250 ページ)
- ISNUMERIC 関数 [その他] (253 ページ)

日付と時刻の関数

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。

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 (MONTH など) には、時間を操作するために 4 つの関数がありますが、2 つの名前 (MONTH と MONTHS など) だけが使用されます。

これらの関数は、Transact-SQL の日付および時刻関数です。これらは、日付および時刻関数へのアクセスおよび操作の代替方法です。

- DATEADD
- DATEDIFF
- DATENAME
- DATEPART
- GETDATE

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 )
```

SAP 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');
```

SAP 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、「任意の開始日」から経過した月数を返します。SAP Sybase IQ の「任意の開始日」である仮想日付 0000-01-01 は、最も効率的に日付計算を行うために決められた日付で、さまざまなデータ部分で一貫しています。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 SAP Sybase IQ によって処理され、両方とも 12 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;
```

SQL Anywhere によって処理される最初の文は値 307 を生成しますが、SAP Sybase IQ によって処理される 2 番目の文は 166 を生成します。

結果の一貫性を保つために、必要かどうかにかかわらず、必ず **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。日付または時刻の関数を使用する **SELECT** 文の **FROM** 句でこのテーブルを参照すると、必ず SAP Sybase IQ によって処理され、一貫性のある結果を得ることができます。

参照：

- ISDATE 関数 [日付および時刻] (251 ページ)
- DATE 関数 [日付と時刻] (191 ページ)
- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEFORMAT 関数 [日付と時刻] (201 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- DATETIME 関数 [日付と時刻] (208 ページ)
- DAY 関数 [日付と時刻] (208 ページ)
- DAYNAME 関数 [日付と時刻] (209 ページ)
- DAYS 関数 [日付と時刻] (209 ページ)
- DOW 関数 [日付と時刻] (217 ページ)
- GETDATE 関数 [日付と時刻] (228 ページ)
- HOUR 関数 [日付と時刻] (235 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTE 関数 [日付と時刻] (276 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTH 関数 [日付と時刻] (279 ページ)
- MONTHNAME 関数 [日付と時刻] (280 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)

- NOW 関数 [日付と時刻] (289 ページ)
- QUARTER 関数 [日付と時刻] (308 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- SECONDS 関数 [日付と時刻] (337 ページ)
- TODAY 関数 [日付と時刻] (368 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)
- YMD 関数 [日付と時刻] (388 ページ)

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。

次の表に、*date-part* に指定可能な値を示します。

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 ~ 土曜)
Hour	hh	0 – 23
Minute	mi	0 – 59
Second	ss	0 – 59
Millisecond	ms	0 – 999
Microsecond	mcs または us	0 – 999999
Calyearofweek	cyr	integer 型。指定した週が始まる年。週の一部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る。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る。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らない。

日付要素	省略形	値
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'
```

互換性

Adaptive Server と互換性をとる場合、Transact-SQL 日付/時刻関数を使用します。

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)

HTTP 関数

HTTP 関数は、Web Service 内での HTTP 要求の処理を容易にします。

注意： クロスサイトスクリプティング (XSS) 攻撃を防ぐために、Web サービスにコーディングのベストプラクティスを使用していることを確認してください。オープンソースリソースは OWASP などの組織から入手できます。

参照：

- HTML_DECODE 関数 [その他] (237 ページ)
- HTML_ENCODE 関数 [その他] (239 ページ)
- HTTP_DECODE 関数 [Web サービス] (241 ページ)
- HTTP_ENCODE 関数 [Web サービス] (242 ページ)
- HTTP_HEADER 関数 [Web サービス] (243 ページ)
- HTTP_VARIABLE 関数 [Web サービス] (245 ページ)
- NEXT_HTTP_HEADER 関数 [Web サービス] (286 ページ)

- NEXT_HTTP_VARIABLE 関数 [Web サービス] (288 ページ)

数値関数

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。

SAP 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。結果の一貫性を保つために、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。

参照：

- ABS 関数 [数値] (157 ページ)
- ACOS 関数 [数値] (157 ページ)
- ASIN 関数 [数値] (160 ページ)
- ATAN 関数 [数値] (161 ページ)
- ATAN2 関数 [数値] (162 ページ)
- CEIL 関数 [数値] (169 ページ)
- CEILING 関数 [数値] (170 ページ)
- COS 関数 [数値] (183 ページ)
- COT 関数 [数値] (184 ページ)
- DEGREES 関数 [数値] (214 ページ)
- EXP 関数 [数値] (223 ページ)
- FLOOR 関数 [数値] (227 ページ)
- LN 関数 [数値] (267 ページ)
- LOG 関数 [数値] (270 ページ)
- LOG10 関数 [数値] (270 ページ)
- MOD 関数 [数値] (278 ページ)
- PI 関数 [数値] (304 ページ)
- POWER 関数 [数値] (304 ページ)
- RADIANS 関数 [数値] (309 ページ)
- RAND 関数 [数値] (310 ページ)
- REMAINDER 関数 [数値] (323 ページ)
- ROUND 関数 [数値] (331 ページ)
- SIGN 関数 [数値] (338 ページ)

- SIN 関数 [数値] (339 ページ)
- SQRT 関数 [数値] (351 ページ)
- SQUARE 関数 [数値] (351 ページ)
- TAN 関数 [数値] (368 ページ)
- TRUNCNUM 関数 [数値] (370 ページ)
- WIDTH_BUCKET 関数 [数値] (383 ページ)

文字列関数

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。

大部分の文字列関数は、*string-expr* パラメータで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**、**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。

LONG VARCHAR 型の結果を返す関数 (**SPACE**、**REPEAT** など) に定数の **LENGTH** 引数を指定しない場合、デフォルトの長さは最大許容値となります。

これらの関数を 1 つ以上含む SAP 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指定して関数の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。

注意：文字列関数による LONG BINARY データ型と LONG VARCHAR データ型のサポートについては、『非構造化データ分析』の「関数のサポート」を参照してください。

参照：

- ASCII 関数 [文字列] (159 ページ)
- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR 関数 [文字列] (171 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- CHARINDEX 関数 [文字列] (173 ページ)
- DIFFERENCE 関数 [文字列] (216 ページ)
- GRAPHICAL_PLAN 関数 [文字列] (228 ページ)
- HTML_PLAN 関数 [文字列] (240 ページ)
- INSERTSTR 関数 [文字列] (249 ページ)
- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- LOCATE 関数 [文字列] (268 ページ)
- LOWER 関数 [文字列] (271 ページ)
- LTRIM 関数 [文字列] (272 ページ)
- PATINDEX 関数 [文字列] (296 ページ)
- REPEAT 関数 [文字列] (324 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REPLICATE 関数 [文字列] (327 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- RTRIM 関数 [文字列] (335 ページ)
- SIMILAR 関数 [文字列] (338 ページ)
- SORTKEY 関数 [文字列] (340 ページ)
- SOUNDEX 関数 [文字列] (345 ページ)
- SPACE 関数 [文字列] (349 ページ)
- STR 関数 [文字列] (356 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)
- STRING 関数 [文字列] (359 ページ)
- STRTOUUID 関数 [文字列] (360 ページ)
- STUFF 関数 [文字列] (361 ページ)
- SUBSTRING 関数 [文字列] (362 ページ)

- SUBSTRING64 関数 [文字列] (365 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)
- UUIDTOSTR 関数 [文字列] (374 ページ)

システム関数

システム関数はシステム情報を返します。

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 SAP Sybase IQ 関数とは異なる方法で処理されます。SAP 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。

参照：

- CEIL 関数 [数値] (169 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- COL_NAME 関数 [システム] (176 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- DB_ID 関数 [システム] (211 ページ)
- DB_NAME 関数 [システム] (212 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- EVENT_CONDITION 関数 [システム] (219 ページ)
- EVENT_CONDITION_NAME 関数 [システム] (221 ページ)
- EVENT_PARAMETER 関数 [システム] (221 ページ)
- GROUP_MEMBER 関数 [システム] (232 ページ)
- INDEX_COL 関数 [システム] (248 ページ)
- NEXT_CONNECTION 関数 [システム] (284 ページ)

- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- PROPERTY 関数 [システム] (305 ページ)
- PROPERTY_DESCRIPTION 関数 [システム] (306 ページ)
- PROPERTY_NAME 関数 [システム] (307 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)
- SUSER_ID 関数 [システム] (366 ページ)
- SUSER_NAME 関数 [システム] (367 ページ)
- USER_ID 関数 [システム] (373 ページ)
- USER_NAME 関数 [システム] (373 ページ)

Adaptive Server システム関数の互換性

すべての Adaptive Server システム関数が SAP Sybase IQ に実装されているわけではありません。

表 7 : SAP Sybase IQ における Adaptive Server 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	未実装
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
<code>datalength</code>	実装済み
<code>curunreservedpgs</code>	未実装
<code>data_pgs</code>	未実装
<code>host_id</code>	未実装
<code>host_name</code>	未実装
<code>lct_admin</code>	未実装
<code>reserved_pgs</code>	未実装
<code>rowcnt</code>	未実装
<code>used_pgs</code>	未実装
<code>valid_name</code>	未実装
<code>valid_user</code>	未実装

説明

- 一部のシステム関数は、システムストアプロシージャとして SAP Sybase IQ に実装されています。
- `db_id`、`db_name`、`datalength`、`suser_id`、および `suser_name` 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

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します。

例

特定の接続プロパティの値を取得するには、`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には、`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。

参照：

- PROPERTY 関数 [システム] (305 ページ)

- PROPERTY_NAME 関数 [システム] (307 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)

サーバで使用可能なプロパティ

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します。

サーバプロパティは、サーバ全体に対して適用されます。

Server Edition プロパティは、SAP Sybase IQ のエディションではなく、SQL Anywhere のエディションを返します。SQL Anywhere のライセンス情報を表示するには、`sp_iqlmconfig` システムプロシージャを使用します。

例

特定のサーバプロパティの値を取得するには、`property` 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です。

```
select property ( 'MainHeapPages' ) from iq_dummy
```

すべてのサーバプロパティの値を取得するには、`sa_eng_properties` システムプロシージャを使用します。

```
call sa_eng_properties
```

参照：

- `sp_iqlmconfig` プロシージャ (527 ページ)
- PROPERTY 関数 [システム] (305 ページ)
- PROPERTY_NAME 関数 [システム] (307 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)

接続プロパティの一覧

接続プロパティは、SAP Sybase IQ データベースへの接続ごとに使用できます。接続プロパティ名では大文字と小文字が区別されません。

接続プロパティ

プロパティ名	説明
allow_nulls_by_default	NULL と NOT NULL のいずれも指定せずに作成されたカラムについて NULL 値の入力を許可するかどうかを示す値を返します。
allow_read_client_file	データベースサーバがクライアントコンピュータ上のファイルの読み込みを許可するかどうかを示す値を返します。
allow_snapshot_isolation	スナップショットアイソレーションが有効であるか無効であるかを示す値を返します。
allow_write_client_file	データベースサーバがクライアントコンピュータへのファイルの書き込みを許可するかどうかを示す値を返します。
ansi_blanks	どのような場合に文字データがクライアント側でトランケートされるかを示す値を返します。
ansi_close_cursors_on_rollback	WITH HOLD で開かれたカーソルが ROLLBACK の実行時に閉じるかどうかを示す値を返します。
ansi_permissions	DELETE および UPDATE 文について権限がチェックされるかどうかを示す値を返します。
ansi_substring	start パラメータまたは length パラメータに負の値が設定された場合に SUBSTRING (SUBSTR) 関数がどのような動作をするかを示す値を返します。
ansi_update_constraints	更新可能な範囲を示す値を返します。
ansinull	NULL をどのように解釈するかを示す値を返します。

プロパティ名	説明
AppInfo	<p>接続を確立したクライアントに関する情報を返します。HTTP 接続では、ブラウザの情報が含まれています。jConnect または Sybase Open Client の古いバージョンを使った接続については、情報は不完全の場合があります。</p> <p>API 値は、DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD、DBEXPRESS のいずれかです。</p>
ApproximateCPU-Time	<p>特定の接続で使用された累積 CPU 時間の概算値(秒単位)を返します。この値と実際の CPU 時間との誤差は、通常は 5～10 % ですが、最大で 50 % に達することもあります。マルチプロセッサコンピュータでは、CPU (ハイパースレッドまたはコア) ごとに使用時間が積算されるため、すべての接続の累積時間を合計した値が実際の経過時間より大きくなる場合もあります。このプロパティは Windows と Linux でサポートされています。</p>
auditing	<p>PUBLIC.auditing オプションが On に設定されている場合は、On を返します。それ以外の場合は、Off を返します。</p> <p>PUBLIC.auditing オプションが On に設定され、conn_auditing オプションが Off に設定されている場合は、現在の接続が監査されていない場合でも、auditing 接続プロパティは On を返します。</p>
auditing_options	<p>このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。</p>
Authenticated	<p>アプリケーションが有効な接続認証文字列を送信した場合は Yes を返します。アプリケーションが有効な接続認証文字列を送信しなかった場合は No を返します。</p>
AuthType	<p>接続時に使用される認証のタイプを返します。戻り値は、Standard、Integrated、Kerberos、LDAPUA、または空の文字列です。空の文字列は、内部接続と AUTHORIZATION OFF を使用する HTTP サービスの接続の場合に返されます。</p>
auto_commit_on_create_local_temp_index	<p>auto_commit_on_create_local_temp_index オプションの設定を返します。</p>

プロパティ名	説明
background_priority	このプロパティは使用されなくなりました。現在の接続が他の接続のパフォーマンスにもたらす影響の大きさを示す値を返します。
BlockedOn	現在の接続がブロックされていない場合は 0 を返し、ブロックされている場合はロック競合によってブロックされた接続の接続番号を返します。
blocking	ロック競合に対するデータベースサーバの動作を示す値を返します。
blocking_others_timeout	現在の接続がロールバックされる前に、別の接続が現在の接続のローおよびテーブルロックでブロックされる可能性がある時間の長さを返します。
blocking_timeout	トランザクションがロックを獲得するまでの待ち時間(ミリ秒単位)を返します。
BytesReceived	クライアント/サーバ通信中に受信したバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されません。
BytesReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したバイト数を返します(この値は、圧縮が無効の場合は BytesReceived の値と同じ)。
BytesSent	クライアント/サーバ通信中に送信されたバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されません。
BytesSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたバイト数を返します(この値は、圧縮が無効の場合は BytesSent の値と同じ)。
CacheHits	成功したキャッシュ読み込み回数を返します。
CacheRead	キャッシュ内で検索されたデータベースページの数です。
CacheReadIndInt	キャッシュから読み込まれたインデックス内部ノードページの数です。
CacheReadIndLeaf	キャッシュから読み込まれたインデックスリーフページの数です。

プロパティ名	説明
CacheReadTable	キャッシュから読み込まれたテーブルページの数を読み返します。
CacheReadWorkTable	キャッシュワークテーブルの読み込み数を返します。
CarverHeapPages	クエリ最適化などの短期的な目的に使用されたヒープページの数を読み返します。
chained	BEGIN TRANSACTION 文が指定されていない場合に使用されるトランザクションモードを読み返します。
CharSet	接続で使用される CHAR 文字セットを読み返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
checkpoint_time	データベースサーバが、あるチェックポイントを実行してから次のチェックポイントを実行するまでの最大時間 (分単位) を読み返します。
cis_option	リモートデータアクセス用のデバッグ情報がデータベースサーバメッセージウィンドウに表示される場合は 7 を返し、表示されない場合は 0 を読み返します。
cis_rowset_size	各フェッチにおいてリモートサーバから返されるローの数を読み返します。
ClientLibrary	jConnect 接続の場合は jConnect、Sybase Open Client 接続の場合は CT_Library、HTTP 接続の場合は None、ODBC 接続、Embedded SQL 接続、OLE DB 接続、ADO.NET 接続、SAP Sybase IQ JDBC ドライバ接続の場合は CmdSeq を読み返します。
ClientNodeAddress	クライアント/サーバ接続のクライアント側に対応するノードを読み返します。クライアントとサーバの両方が同じコンピュータにある場合は、空の文字列を読み返します。これは NodeAddress プロパティの同意語です。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を読み返します。
ClientPort	クライアントの TCP/IP ポート番号を読み返します。接続の種類が TCP/IP でない場合は、0 を読み返します。

プロパティ名	説明
ClientStmtCache-Hits	クライアントに文がキャッシュされていることが理由で、この接続には不要な準備文の数を返します。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。
ClientStmtCache-Misses	この接続において、クライアントのキャッシュに存在した文の中で、再び準備された文の数を返します。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。
close_on_endtrans	トランザクションの終了時にカーソルを閉じるかどうかを示す値として、On または Off を返します。
collect_statistics_on_dml_updates	INSERT、DELETE、UPDATE などのデータを変更する DML 文の実行中に統計情報を収集するかどうかを示す値として、On または Off を返します。
Commit	処理されたコミット要求の数を返します。
CommLink	接続の通信リンクを返します。これは SAP Sybase IQ がサポートするネットワークプロトコルであり、同一コンピュータ接続の場合は local となります。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。
CommNetworkLink	接続の通信リンクを返します。これは SAP Sybase IQ がサポートするネットワークプロトコルです。値には SharedMemory と TCPIP があります。CommNetworkLink プロパティは、同一コンピュータかどうかにかかわらず、常にリンク名を返します。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。
CommProtocol	Sybase Open Client 接続と jConnect 接続の場合は TDS、HTTP 接続の場合は HTTP、HTTPS 接続の場合は HTTPS、ODBC 接続、Embedded SQL 接続、OLE DB 接続、ADO.NET 接続、SAP Sybase IQ JDBC ドライバ接続の場合は CmdSeq を返します。

プロパティ名	説明
Compression	この接続で通信圧縮が有効であるかどうかを示す値として、On または Off を返します。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。
conn_auditing	この接続に対して監査が有効になっている場合は、auditing オプションが Off に設定されていても、On を返します。
ConnectedTime	接続がこれまで接続してきた合計の時間数を秒単位で返します。
connection_authentication	クライアントの認証に使用される文字列を返します。データベースが変更可能になる前に、認証が必要です。
continue_after_raiserror	RAISERROR 文が見つかるたびにプロシージャまたはトリガの実行を停止するかどうかを示す値として、On または Off を返します。
conversion_error	データベースから情報をフェッチするときにデータ型変換障害をレポートするかどうかを示す値として、On または Off を返します。
cooperative_commit_timeout	データベースサーバがディスクへの書き込み前に他の接続の書き込みによって1つのログページが満杯になるまで待っている時間(ミリ秒単位)を返します。
cooperative_commits	コミットをいつディスクに書き込むかを示す値として、On または Off を返します。
CurrentLineNumber	接続で実行されているプロシージャまたは複合文の現在の行番号を返します。実行されているプロシージャの名前は CurrentProcedure プロパティで取得できます。現在の行がクライアント側からの複合文の一部である場合は、空の文字列を返します。
CurrentProcedure	接続で現在実行されているプロシージャの名前を返します。接続でネストされたプロシージャコールが実行されている場合は、現在のプロシージャの名前を返します。実行されているプロシージャがない場合は、空の文字列を返します。
Cursor	現在データベースサーバが保持している宣言されたカーソルの数を返します。

プロパティ名	説明
CursorOpen	現在データベースサーバが保持している開いたカーソルの数を返します。
database_authentication	データベースの認証に使用される文字列を返します。データベースサーバの認証が完了してからでなければ、データベースに変更を加えることはできません。
date_format	データベースから取り出した日付の形式を示す文字列を返します。
date_order	日付形式を設定する方法を示す文字列を返します。
db_publisher	データベースパブリッシャのユーザ ID を返します。
DBNumber	データベースの ID 番号を返します。
debug_messages	DEBUG ONLY 句を含む MESSAGE 文を実行するかどうかを示す値として、On または Off を返します。
dedicated_task	要求処理タスクがこの接続からの要求の処理専用であるかどうかを示す値として、On または Off を返します。
default_dbpace	デフォルト DB 領域の名前を返します。デフォルト DB 領域が指定されていない場合は、空の文字列を返します。
default_timestamp_increment	TIMESTAMP 型のカラム内の値を一意に保つために加算する値 (ミリ秒単位) を返します。
delayed_commit_timeout	COMMIT の後でデータベースサーバがアプリケーションに制御を戻すまでの待ち時間 (ミリ秒単位) を返します。
delayed_commits	COMMIT の後でデータベースサーバがアプリケーションに制御を戻す時点を示す値として、On または Off を返します。
disk_sandbox	On または Off を返し、データベースの読み込み/書き込みファイル操作が、メインのデータベースファイルがあるディレクトリに制限されているかどうかを示します。
DiskRead	ディスクから読み込まれたページの数と返します。
DiskReadHint	ディスクの読み込みヒント数を返します。
DiskReadHintPages	ディスクの読み込みヒントページの数と返します。
DiskReadIndInt	ディスクから読み込まれたインデックス内部ノードページの数と返します。

プロパティ名	説明
DiskReadIndLeaf	ディスクから読み込まれたインデックスリーフページの数 を返します。
DiskReadTable	ディスクから読み込まれたテーブルページの数 を返します。
DiskReadWorkTable	ディスクワークテーブルの読み込み数 を返します。
DiskSyncRead	同期的に発行されたディスクの読み込み数 を返します。
DiskSyncWrite	同期的に発行された書き込み数 を返します。
DiskWaitRead	データベースサーバによる非同期読み込みの 待機回数を返します。
DiskWaitWrite	データベースサーバによる非同期書き込みの 待機回数を返します。
DiskWrite	ディスクに書き込まれた修正ページの数 を返します。
DiskWriteHint	ディスクの書き込みヒント数を返 します。
DiskWriteHintPages	ディスクの書き込みヒントページ の数を返します。
divide_by_zero_error	ゼロ除算によってエラーが発生する 場合は On を返し、ゼロ除算によっ てエラーが発生しない場合は Off を返します。
Encryption	接続が暗号化されるかどうかを示 す値を返します。
escape_character	このプロパティはシステムで使 用するために予約されています。 このオプションの設定は変更しな いください。
EventName	接続でイベントハンドラが実行 されている場合、関連付けられ ているイベントの名前を返しま す。それ以外の場合は、空の文 字列が返されます。
exclude_operators	このプロパティはシステムで使 用するために予約されています。 このオプションの設定は変更しな いください。
ExprCacheAbandons	ヒット率が低すぎるために式 キャッシュが中止された回数 を返します。
ExprCacheDropsToReadOnly	ヒット率が低いために式 キャッシュが読み込み専用ステ ータスに変更された回数 を返します。
ExprCacheEvicts	式キャッシュからの出力数 を返します。
ExprCacheHits	式キャッシュ内のヒット数 を返します。

プロパティ名	説明
ExprCacheInserts	式キャッシュに挿入された値の数を返します。
ExprCacheLookups	式キャッシュ内で実行されたルックアップの回数を返します。
ExprCacheResume- sOfReadWrite	ヒット率の上昇によって式キャッシュが読み込み/書き込みステータスに戻った回数を返します。
ExprCacheStarts	式キャッシュが開始された回数を返します。
extended_join_ syntax	複数テーブルのジョインに対して、重複する相関名構文を持つクエリが許可されている場合は On、エラーとしてレポートされる場合は Off を返します。
extern_login_cre- dentials	リモート接続の試行で、ログインしたユーザの外部ログインクレデンシアルを使用するか、効果的なユーザの外部ログインクレデンシアルを使用するかを返します。
fire_triggers	データベースにおいてトリガが起動する場合は On を返し、起動しない場合は Off を返します。
first_day_of_week	週の最初の曜日を表す番号を返します。7 は日曜日を表し、1 は月曜日を表します。
for_xml_null_ treatment	NULL 値を含む要素と属性が結果から除外される場合は OMIT を返し、クエリ内で FOR XML 句が使用されている場合に NULL 値に対応する空の要素または属性が生成される場合は EMPTY を返します。
force_view_crea- tion	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。
FullCompare	インデックスのハッシュ値を超えて実行された比較の回数を返します。
GetData	GETDATA 要求の数を返します。
global_database_ id	DEFAULT GLOBAL AUTOINCREMENT が指定されたカラムの開始値を返します。
HashForcedParti- tions	メモリ競合が原因でハッシュ演算子が分割された回数を返します。
HashRowsFiltered	ビットベクトルフィルタによって拒否されたプローブロー数を返します。

プロパティ名	説明
HashRowsPartitioned	ハッシュワークテーブルに書き込まれたロー数を返します。
HashWorkTables	ハッシュベースの演算用に作成されたワークテーブルの数を返します。
HeapsCarver	クエリ最適化などの短期的な目的に使用されたヒープの数を返します。
HeapsLocked	キャッシュ内で現在ロックされている再配置可能なヒープの数を返します。
HeapsQuery	クエリ処理 (ハッシュ操作とソート操作) に使用されるヒープの数を返します。
HeapsRelocatable	再配置可能なヒープの数を返します。
http_connection_pool_basesize	データベース接続の公称スレッシュホールドサイズを返します。
http_connection_pool_timeout	未使用の接続が接続プールに格納される最長時間を返します。
http_session_timeout	現在の HTTP セッションタイムアウトを返します (分単位)。
HttpServiceName	現在の HTTP 要求のサービス名エンリポイントを返します。このプロパティは、エラーレポートやフロー制御の場合に便利です。このプロパティを HTTP 要求以外により実行されたストアプロシージャから選択した場合、または接続が現在アクティブではないか、HTTP セッションの再開を待っている状態である場合は、空の文字列を返します。
IdleTimeout	接続のアイドルタイムアウト値を返します。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。
IndAdd	インデックスに追加されたエンリの数を返します。
IndLookup	インデックス内で検索されたエンリの数を返します。
integrated_server_name	統合化ログインを目的とする Windows ユーザグループメンバーシップの検索に使用されるドメインコントローラサーバの名前を返します。

プロパティ名	説明
IsDebugger	SAP Sybase IQ デバッガの実行に使用されている接続を区別するために Yes または No を返します。現在の接続番号がデバッガ接続の接続番号と対応する場合は値が Yes、それ以外の場合は No になります。
isolation_level	接続の独立性レベルを返します (0、1、2、3、Snapshot、Statement-snapshot、または Readonly-statement-snapshot)。
java_class_path	クラスを検索する追加のディレクトリまたは JAR ファイルのリストを返します。
java_location	データベースに対して Java VM が指定されている場合、そのパスを返します。
java_main_userid	このプロパティは使用されなくなりました。
java_vm_options	Java VM の起動時にデータベースサーバが使用するコマンドラインオプションを返します。
Language	ロケール言語を返します。
LastCommitRedoPos	接続によって最後の COMMIT 操作がトランザクションログに書き込まれた後で REDO ログの位置を返します。
LastIdle	要求間のチック数を返します。
LastPlanText	接続で最後に実行されたクエリの長いテキストプランを返します。最後のプランを記憶するかどうかは、sa_server_option システムプロシージャの RememberLastPlan オプションを設定するか、-zp サーバオプションを使用することで制御します。
LastReqTime	指定された接続において最後の要求が開始された時刻を返します。このプロパティは、イベントなどの内部接続の場合は空の文字列を返すことがあります。

プロパティ名	説明
LastStatement	<p>現在の接続で最後に準備された SQL 文を返します。</p> <p>LastStatement の値は、文が準備されると同時に設定され、文が削除されると同時にクリアされます。各接続につき 1 つの文の文字列のみが記憶されます。</p> <p>ある接続について sa_conn_activity が空でない値を返した場合、その接続で現在実行されている文である可能性が高くなります。その文が完了している場合は、文がすでに削除され、このプロパティの値がクリアされている可能性があります。アプリケーションが複数の文を準備し、それらの文のステートメントハンドルを保持している場合、LastStatement が返す値は接続で現在実行されている処理を表しません。</p> <p>クライアントでの文のキャッシュが有効であり、キャッシュされた文が再使用されているとき、このプロパティは空の文字列を返します。</p>
LivenessTimeout	<p>現在の接続の活性タイムアウト時間を返します。</p> <p>現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。</p>
lock_rejected_rows	<p>このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。</p>
LockCount	<p>接続で保持されているロックの数を返します。</p>
LockIndexID	<p>ロックされたインデックスの識別子を返します。</p>
LockName	<p>接続が解放を待っているロックを示す 64 ビット符号なし整数を返します。</p>
LockRowID	<p>ロックされたローの識別子を返します。</p>
LockTableOID	<p>接続がブロックされていない場合、または接続が CONNECTION_PROPERTY を呼び出した接続とは別のデータベース上にある場合は、0 を返します。それ以外の場合は、接続が解放を待っているロックが設定されているテーブルのオブジェクト ID を返します。このオブジェクト ID によって、SYSTAB システムビューを使用してテーブル情報を検索できます。</p>
log_deadlocks	<p>デッドロック情報がレポートされる場合は On を返し、それ以外の場合は Off を返します。</p>

プロパティ名	説明
LogFreeCommit	Redo Free Commit の数を返します。Redo Free Commit が発生するのは、トランザクションログのコミットが要求されているが、ログはすでに書き込まれている (コミットが実行されている) 場合です。
login_mode	サポートされているログインのタイプを示す値として、Standard、Integrated、Kerberos、または LDAPUA の中から 1 つまたは複数返します。
login_procedure	起動時に互換性オプションを設定するストアプロシージャの名前を返します。
LoginTime	接続が確立された日付と時刻を返します。
LogWrite	トランザクションログに書き込まれたページの数返します。
materialized_view_optimization	マテリアライズドビューをクエリ最適化の実行中に使用するかどうかを示す値を返します。 Disabled Fresh Stale NMinute[s] NHour[s] NDay[s] NWeek[s] NMonth[s]
max_client_statements_cached	クライアントでキャッシュされる文の数を返します。
max_cursor_count	接続で一度に使用できるカーソルの最大数を返します。
max_hash_size	このプロパティは使用されなくなりました。
max_plans_cached	キャッシュに格納される実行プランの最大数を返します。
max_priority	接続に設定できる最高優先度レベルを示す値を返します。
max_query_tasks	データベースサーバがクエリの処理に使用できる要求の最大数を返します。

プロパティ名	説明
max_recursive_ iterations	再帰共通テーブル式が実行できる反復処理の最大回数を返します。
max_statement_ count	接続で一度に使用できる準備文の最大数を返します。
max_temp_space	1つの接続で使用できるテンポラリファイル領域の最大サイズを返します。
MessageReceived	MESSAGE 文によって生成され、WAITFOR 文が中断する原因となった文字列を返します。それ以外の場合は、空の文字列が返されます。
min_password_ length	データベースにおける新しいパスワードの最小長を返します。
min_role_admins	ロールに必要な管理者の最小数を返します。min_role_admins option を参照してください。

プロパティ名	説明
Name	<p>現在の接続の名前を返します。</p> <p>ConnectionString (CON) 接続パラメータを使用して接続名を指定できます。</p> <p>データベースサーバによって作成されるテンポラリ接続には次の名前が使用されます。</p> <p>INT:ApplyRecovery INT:BackupDB INT:Checkpoint INT:Cleaner INT:CloseDB INT>CreateDB INT>CreateMirror INT:DelayedCommit INT:DiagRcvr INT:DropDB INT:EncryptDB INT:Exchange INT:FlushMirrorLog INT:FlushStats INT:HTTPReq INT:PromoteMirror INT:PurgeSnapshot INT:ReconnectMirror INT:RecoverMirror INT:RedoCheckpoint INT:RefreshIndex INT:ReloadTrigger INT:RenameMirror INT:RestoreDB INT:StartDB INT:VSS</p>
NcharCharSet	<p>接続で使用される NCHAR 文字セットを返します。このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。</p>

プロパティ名	説明
nearest_century	文字列から日付への変換において 2 桁の年がどのように解釈されるかを示す値を返します。
NodeAddress	クライアント/サーバ接続のクライアント側に対応するノードを返します。クライアントとサーバの両方が同じコンピュータにある場合は、空の文字列を返します。
non_keywords	オフになっているため識別子として使用できるキーワードのリストを返す (そのようなキーワードが存在する場合)。
Number	現在の接続の接続 ID (数値) を返します。
NumLocalTempTables	接続別の使用中ローカルテンポラリテーブルの数を返します。ローカルテンポラリテーブルは、削除されたか、スコープ外になったときでも、次の COMMIT までは使用中とみなされます。
odbc_describe_binary_as_varbinary	SAP Sybase IQ ODBC ドライバが BINARY カラムと VARBINARY カラムをどちらも SQL_BINARY として記述する場合は Off を返し、ODBC ドライバが BINARY カラムと VARBINARY カラムを SQL_VARBINARY として記述する場合は On を返します。
odbc_distinguish_char_and_varchar	CHAR カラムが SQL_VARCHAR として記述される場合は Off を返し、SQL_CHAR として記述される場合は On を返します。
oem_string	データベースファイルのヘッダページ内の文字列を返します。
on_charset_conversion_failure	文字セットの変換中にエラーが発生した場合の動作を示す値として、Ignore、Warning、Error のいずれかを返します。
on_tsql_error	ストアプロシージャまたは T-SQL バッチの実行中にエラーが発生した場合の動作を示す値として、Stop、Conditional、Continue のいずれかを返します。
optimization_goal	クエリ処理を最適化する方法を示す値として、First-row または All-rows を返します。
optimization_level	0 ~ 15 の範囲の値を返します。この値を使用して、SAP Sybase IQ クエリオプティマイザが SQL 文のアクセスプランの検索に費やす作業レベルを指定します。

プロパティ名	説明
optimization_workload	SAP Sybase IQ クエリオプティマイザが SQL 文のアクセスプランの検索に費やす作業レベルを示す値を返します。
OSUser	クライアントプロセスに関連付けられたオペレーティングシステムユーザ名を返します。クライアントプロセスで別のユーザを同一化している場合 (UNIX の場合は、セット ID ビットが設定されている場合)、同一化されたユーザ名が返されます。バージョン 10.0.1 以前のクライアントと HTTP や TDS クライアントでは、空の文字列を返します。
PacketSize	接続で使用されるパケットサイズ (バイト単位) を返します。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。
PacketsReceived	受信したクライアント/サーバ通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したパケットの数を返します (この値は、圧縮が無効の場合は PacketsReceived の値と同じ)。
PacketsSent	送信されたクライアント/サーバ通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたパケットの数を返します (この値は、圧縮が無効の場合は PacketsSent の値と同じ)。
ParentConnection	データベースオペレーション (データベースのバックアップや作成など) を実行するためにテンポラリ接続を作成した接続の接続 ID を返します。その他のタイプの接続について、このプロパティは NULL を返します。
pinned_cursor_percent_of_cache	カーソルを固定するために使用できるキャッシュの割合を返します。
post_login_procedure	ユーザが接続したときにアプリケーションが表示する必要があるメッセージがプロシージャの結果セットに含まれている場合、そのプロシージャの名前を返します。
precision	10 進数精度と数値精度の設定を返します。

プロパティ名	説明
prefetch	プリフェッチが行われない場合は Off、カーソルタイプが SENSITIVE であるかクエリにプロキシテーブルが含まれていないかぎりプリフェッチが行われる場合は Conditional、カーソルタイプが SENSITIVE であり、カーソルにプロキシテーブルが含まれていても常にプリフェッチが行われる場合は Always を返します。
Prepares	接続に対して実行された文の準備作業の数を返します。
PrepStmt	現在データベースサーバが保持している準備文の数を返します。
preserve_source_format	プロシージャ、トリガ、ビュー、イベントハンドラの元のソース定義がシステムファイルに保存される場合は On を返し、それ以外の場合は Off を返します。
prevent_article_pkey_update	パブリケーションで使用されるテーブルのプライマリキーカラムの更新が許可されている場合は On を返し、それ以外の場合は Off を返します。
priority	接続の優先度レベルを示す値を返します。
Progress	クエリが実行されている時間に関する情報を返します。次に例を示します。 <pre>43% (9728 of 22230 pages) complete after 00:00:05; estimated 00:00:06 remaining</pre> このプロパティには、プロパティ値を問い合わせるときに指定できる拡張機能があります。
progress_messages	progress_messages オプションの値を返します。
query_mem_timeout	query_mem_timeout オプションの値を返します。
QueryBypassed	オプティマイザバイパスによって最適化された要求の数を返します。
QueryBypassedCosted	オプティマイザバイパスによってコストを使用して処理された要求の数を返します。
QueryBypassedHeuristic	オプティマイザバイパスによってヒューリスティックを使用して処理された要求の数を返します。

プロパティ名	説明
QueryBypassedOptimized	オプティマイザバイパスによって最初に処理されてから、SAP Sybase IQ オプティマイザによって完全に最適化された要求の数を返します。
QueryCachedPlans	接続において現在キャッシュされているクエリ実行プランの数を返します。
QueryCachePages	実行プランの保存に使用されるキャッシュページの数を返します。
QueryDescribedBy-pass	オプティマイザバイパスによって処理された DESCRIBE 要求の数を返します。
QueryDescribedOptimizer	オプティマイザによって処理された DESCRIBE 要求の数を返します。
QueryHeapPages	クエリ処理 (ハッシュ操作とソート操作) に使用されるキャッシュページの数を返します。
QueryJHToJNLOptUsed	ハッシュジョインがネストループジョインに変換された回数を返します。
QueryLowMemoryStrategy	メモリ不足が原因でサーバが実行プランを実行中に変更した回数を返します。現在使用できるメモリがオプティマイザの推定よりも少ない、または実行プランが必要とするメモリがオプティマイザの推定よりも多い場合に、プランが変更されることがあります。
QueryMemActive-Curr	クエリメモリをアクティブに使用する要求数を返します。
QueryMemGrant-Failed	要求がクエリメモリ待ちになり、取得できなかった合計回数を返します。
QueryMemGrant-Granted	要求に現在付与されているページ数を返します。
QueryMemGrantRequested	要求がクエリメモリを取得しようとした合計回数を返します。
QueryMemGrantWait-ed	要求がクエリメモリ待ちになった合計回数を返します。
QueryMemGrant-Waiting	クエリメモリ待ちになっている現在の要求数を返します。

プロパティ名	説明
QueryOpened	実行対象の OPEN 要求の数を返します。
QueryOptimized	完全に最適化された要求の数を返します。
QueryReused	プランキャッシュから再利用された要求の数を返します。
QueryRowsFetched	この接続について、逐次スキャンまたはインデックススキャンによってベーステーブルから読み込まれたローの数を返します。
QueryRowsMaterialized	クエリ処理中にワークテーブルに書き込まれるロー数を返します。
quoted_identifier	二重引用符で囲まれた文字列が識別子として解釈される場合は On を返し、リテラル文字列として解釈される場合は Off を返します。
read_past_deleted	独立性レベル 1 または 2 の逐次スキャンにおいてコミットされていない削除ローが無視される場合は On を返し、独立性レベル 1 または 2 の逐次スキャンがコミットされていない削除ローでブロックされる場合は Off を返します。
recovery_time	データベースサーバがシステム障害から回復するのに要する最長時間 (分単位) を返します。
RecursiveIterations	再帰ユニオンの反復回数を返します。
RecursiveIterationsHash	再帰ハッシュジョインでハッシュ方式が使用された回数を返します。
RecursiveIterationsNested	再帰ハッシュジョインでネストループ方式が使用された回数を返します。
RecursiveJNLMisses	再帰ハッシュジョインでのインデックスプローブキャッシュミスの数を返します。
RecursiveJNLProbes	再帰ハッシュジョインにおけるインデックスプローブの試行回数を返します。
remote_idle_timeout	Web サービスのクライアントプロシージャと関数で許容される休止時間 (秒単位) を返します。
replicate_all	内部でのみ使用されます。

プロパティ名	説明
ReqCountActive	処理が完了した要求の数を返します。RequestTiming サーバプロパティが Off に設定されている場合は、NULL を返します。
ReqCountBlockContention	接続がアトミックアクセスを待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。
ReqCountBlockIO	接続が I/O 処理の完了を待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。
ReqCountBlockLock	接続がロックの解放を待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。
ReqCountUnscheduled	接続がスケジューリングを待った回数を返します。-zt オプションが指定されていない場合は、NULL を返します。
ReqStatus	<p>要求のステータスを返します。値は次のいずれかです。</p> <ul style="list-style-type: none"> • Idle – この接続では現在、要求を処理していない。 • Unscheduled* – この接続では他の処理が実行されており、使用可能なデータベースサーバワーカを待っている。 • BlockedIO* – この接続はブロックされ、I/O 処理の完了を待っている。 • BlockedContention* – この接続はブロックされ、共有データベースサーバデータ構造体へのアクセスを待っている。 • BlockedLock – この接続はブロックされ、オブジェクトのロックの解放を待っている。 • Executing – この接続では現在、要求を実行している。 <p>アスタリスク (*) が付いている値が返されるのは、-zt サーバオプションによってデータベースサーバで要求タイミング情報のロギングがオンになっている場合に限られます。要求タイミング情報のロギングが実行されていない場合は (デフォルト)、値は Executing として返されます。</p>
ReqTimeActive	要求の処理にかかった時間 (秒単位)、または、-zt オプションが指定されていなかった場合は NULL を返します。
ReqTimeBlockContention	アトミックアクセスを待っていた時間 (秒単位)、または、RequestTiming サーバプロパティが Off に設定されている場合は NULL を返します。

プロパティ名	説明
ReqTimeBlockIO	I/O 処理の完了を待っていた時間 (秒単位)、または、-zt オプションが指定されていなかった場合は NULL を返します。
ReqTimeBlockLock	ロックを待っていた時間 (秒単位)、または、-zt オプションが指定されていなかった場合は NULL を返します。
ReqTimeUnscheduled	未スケジュール時間を返します。-zt オプションが指定されていない場合は、NULL を返します。
ReqType	最後の要求のタイプを返します。接続が接続プールによってキャッシュされる場合、その ReqType 値は CONNECT_POOL_CACHE になります。
request_timeout	1つの要求を実行できる最大時間を返します。
RequestsReceived	クライアント/サーバ通信要求またはラウンドトリップの数を返します。このプロパティでは、PacketsReceived とは異なり、マルチパケット要求を1つの要求として数え、活性パケットを計数の対象から除外します。
reserved_keywords	データベースで有効なデフォルト以外の予約キーワードを返します。
return_date_time_as_string	DATE、TIME、TIMESTAMP の値が文字列としてアプリケーションに返される場合は On を返し、DATE データ型または TIME データ型として返される場合は Off を返します。
Rlbnk	処理されたロールバック要求の数を返します。
rollback_on_deadlock	参照整合性アクションが UPDATE または DELETE の後に実行される場合は After を返し、UPDATE または DELETE の前に実行される場合は Before を返します。
RollbackLogPages	ロールバックログのページ数を返します。
row_counts	ローの数が常に正確である場合は On を返し、ローの数が通常は推定値である場合は Off を返します。
scale	接続における 10 進数と数値の位取りを返します。
secure_feature_key	データベースサーバにおいて機能を有効または無効にするキーを格納します。このプロパティの値を選択すると、常に空の文字列が返されます。

プロパティ名	説明
ServerNodeAddress	クライアント/サーバ接続のサーバ側に対応するノードを返します。クライアントとサーバの両方が同じコンピュータにある場合は、空の文字列を返します。 現在実行中の要求がイベントハンドラの一部である場合、このプロパティは NA を返します。
ServerPort	データベースサーバの TCP/IP ポート番号または 0 を返します。
SessionCreateTime	HTTP セッションが作成された時刻を返します。
SessionID	接続のセッション ID がある場合は、そのセッション ID を返し、それ以外の場合は空の文字列を返します。
SessionLastTime	HTTP セッションにおける最後の要求の時刻を返します。
SessionTimeout	非アクティブ状態の HTTP セッションが維持される時間(分単位)を返します。
SnapshotCount	接続に関連付けられているスナップショットの数を返します。
sort_collation	ORDER BY 句が変更されていない場合は Internal を返し、それ以外の場合は照合名または照合 ID を返します。
SortMergePasses	ソート中に使用されたマージパスの数を返します。
SortRowsMaterialized	ソートワークテーブルに書き込まれたロー数を返します。
SortRunsWritten	ソート中に書き込まれたソート実行の数を返します。
SortSortedRuns	実行の生成中に作成されたソート実行の数を返します。
SortWorkTables	ソート用に作成されたワークテーブルの数を返します。
sql_flagger_error_level	指定された SQL/2003 に含まれていない場合にエラーとして通知する SQL を示す値として、次のいずれかを返します。 <ul style="list-style-type: none"> • E – 初級レベル SQL/2003 構文でない構文を通知します。 • I – 中級レベル SQL/2003 構文でない構文を通知します。 • F – 上級レベル SQL/2003 構文でない構文を通知します。 • W – サポートされている構文をすべて許可します。

プロパティ名	説明
sql_flagger_warning_level	指定された SQL/2003 に含まれていない場合に警告として通知する SQL を示す値として、次のいずれかを返します。 <ul style="list-style-type: none"> • E – 初級レベル SQL/2003 構文でない構文を通知します。 • I – 中級レベル SQL/2003 構文でない構文を通知します。 • F – 上級レベル SQL/2003 構文でない構文を通知します。 • W – サポートされている構文をすべて許可します。
st_geometry_asbinary_format	空間値がジオメトリからバイナリフォーマットに変換される方法を示す値を返します。
st_geometry_as_text_format	空間値がジオメトリからテキストに変換される方法を示す値を返します。
st_geometry_asxml_format	空間値がジオメトリから xml に変換される方法を示す値を返します。
st_geometry_describe_type	空間データ値を記述する方法を示す値を返します。
st_geometry_interpolation	ST_CircularString ジオメトリの内挿法設定を返します。
st_geometry_on_invalid	ジオメトリが表面検証に失敗した場合の動作を示す値を返します。
StatementDescribes	DESCRIBE 要求によって処理された文の合計数を返します。
StatementPostAnnotates	セマンティックなクエリ変換フェーズによって処理された文の数を返します。
StatementPostAnnotatesSimple	セマンティックなクエリ変換フェーズによって処理されたが、一部のセマンティック変換が省略された文の数を返します。
StatementPostAnnotatesSkipped	セマンティックなクエリ変換フェーズが完全に省略された文の数を返します。
string_rtruncation	文字列がトランケートされたときにエラーが発生する場合は On を返し、エラーなしで文字列がトランケートされる場合は Off を返します。

プロパティ名	説明
subsume_row_locks	データベースサーバがテーブルに対する個別ローロックを取得する場合は On を返し、それ以外の場合は Off を返します。
suppress_tds_debugging	TDS デバッグ情報がデータベースサーバメッセージウィンドウに表示される場合は Off を返し、表示されない場合は On を返します。
synchronize_mirror_on_commit	データベースミラーサーバがコミット時に同期される場合は On を返し、それ以外の場合は Off を返します。
tds_empty_string_is_null	TDS 接続において空の文字列が NULL として返される場合は On を返し、TDS 接続において1つのブランク文字を含む文字列が返される場合は Off を返します。
temp_space_limit_check	データベースサーバが接続で使用できるテンポラリ領域のサイズをチェックする場合は On を返し、データベースサーバが接続で使用できる領域のサイズをチェックしない場合は Off を返します。
TempFilePages	接続で使用されるテンポラリファイルページの数を返します。
TempTablePages	テンポラリテーブルで使用されるテンポラリファイルのページ数を返します。
time_format	データベースから取り出される時刻の文字列形式を返します。
time_zone_adjustment	接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位) を返します。
timestamp_format	データベースから取り出したタイムスタンプの形式を返します。
timestamp_with_time_zone_format	データベースから取り出した TIMESTAMP WITH TIME ZONE 値の形式を返します。
TimeZoneAdjustment	接続のローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位) を返します。

プロパティ名	説明
TransactionStart-Time	COMMIT または ROLLBACK の後にデータベースが最初に変更された時刻を含む文字列を返します。最後の COMMIT または ROLLBACK 以降にデータベースが変更されていない場合は、空の文字列を返します。
truncate_time-stamp_values	TIMESTAMP 値で使用される小数点以下の桁数に制限がある場合は On を返し、ない場合は Off を返します。
trusted_certificates_file	データベースサーバが LDAP サーバのクライアントとして動作するときに、信頼できる認証局の証明書のリストを含むファイルを返します。
tsql_outer_joins	DML 文で Transact-SQL 外部ジョインが使用できる場合は On を返します。
tsql_variables	Embedded SQL のホスト変数名のプレフィクスとしてコロンの代わりに @ 符号を使用できる場合は On を返し、それ以外の場合は Off を返します。
UncommitOp	コミットされていない操作の数を返します。
updatable_statement_isolation	isolation_level オプションが Readonly-statement-snapshot に設定されている場合に更新可能な文に使用される独立性レベル (0、1、2、または 3) を返します。
update_statistics	この接続が統計ガバナーにクエリフィードバックを送信できる場合、On を返します。update_statistics オプションが Off に設定されている場合、統計ガバナーは現在の接続からクエリフィードバックを受け取りません。
upgrade_database_capability	このプロパティはシステムで使用するために予約されています。このオプションの設定は変更しないでください。

プロパティ名	説明
user_estimates	<p>クエリオプティマイザがクエリの述部に含まれる選択性推定を尊重するか無視するかを示す値として、次のいずれかを返します。</p> <ul style="list-style-type: none"> • Enabled – ユーザが提供する選択性推定をすべて尊重します。このオプションは、On を使用して有効にすることもできます。 • Override-Magic – ユーザ選択性推定は尊重されますが、オプティマイザが最後の手段であるヒューリスティック値 (マジック値とも呼ばれます) の使用を選択する以外に方法がない場合のみ使用されます。 • Disabled – 他の推定データが使用できないときは、ユーザ推定は無視され、マジック値が使用されます。このオプションは、Off を使用して無効にすることもできます。
UserAppInfo	AppInfo 接続パラメータによって接続文字列に指定された文字列を返します。
UserDefinedCounterRate01	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate02	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate03	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate04	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。

プロパティ名	説明
UserDefinedCounterRate05	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRaw01	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw02	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw03	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw04	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw05	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserID	接続のユーザ ID を返します。
UtilCmdsPermitted	この接続で CREATE DATABASE、DROP DATABASE、RESTORE DATABASE などのユーティリティコマンドの使用が許可されているかどうかを示す値として、On または Off を返します。
uuid_has_hyphens	このプロパティでは、ユニークな識別子の値を文字列に変換するときのフォーマットを設定します。

プロパティ名	説明
verify_password_function	パスワードの検証に使用される関数が指定されている場合は、その関数名を返します。
wait_for_commit	データベースが次の COMMIT 文まで外部キー整合性をチェックしない場合は、On を返します。それ以外の場合は Off を返します。この場合、check_on_commit オプションで作成されていないすべての外部キーが、挿入、更新、削除時にチェックされます。
WaitStartTime	接続が待機を始めた時刻 (接続が待機していない場合は空の文字列) を返します。
WaitType	待機の理由を返します (ある場合)。WaitType には次の値があります。 <ul style="list-style-type: none"> • lock – 接続がロックを待っている場合に返されます。 • waitfor – 接続が WAITFOR 文を実行している場合に返されます。 • 空の文字列 – 接続が待機していないか、待機の理由が不明な場合に返されます。
webservice_name-space_host	生成された WSDL ドキュメント内で XML ネームスペースとして使用されるホスト名を返す (そのようなホスト名が指定されている場合)。
webservice_sessionid_name	セッション管理が使用されているかどうかを判別するときに Web サーバが使用するセッションの識別子名を返します。

各データベースで使用可能なプロパティ

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できます。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す。

サーバプロパティ **QueryBypassedCosted**、**QueryBypassedOptimized**、**QueryDescribedOptimizer**、**StatementPostAnnotatesSimple** は、カタログストアテーブルに対するクエリについてのみ更新されます。

例

特定のデータベースプロパティの値を取得するには、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返します。

```
select db_property ( 'PageSize' )
```

すべてのデータベースプロパティの値を取得するには、sa_db_properties システムプロシージャを使用します。

```
call sa_db_properties
```

参照：

- PROPERTY 関数 [システム] (305 ページ)
- PROPERTY_NAME 関数 [システム] (307 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)

データベースサーバプロパティの一覧

データベースサーバプロパティ

プロパティ名	説明
ActiveReq	現在クライアント側の要求を処理中のサーバワーカの数 を返します。
ApproximateCPU- Time	データベースサーバで使用された累積 CPU 時間の概算値 (秒 単位) を返します。この値と実際の CPU 時間との誤差は、通 常は 5 ~ 10 % ですが、最大で 50 % に達することもありま す。マルチプロセッサコンピュータでは、CPU (ハイパース レッドまたはコア) ごとに使用時間が積算されるため、すべ ての接続の累積時間を合計した値が実際の経過時間より大 きくなる場合もあります。このプロパティは Windows と Linux でサポートされています。
AutoMultiProgram- mingLevel	データベースサーバがマルチプログラミングレベルを自動 的に調整しているかどうかを示す値を返します。
AutoMultiProgram- mingLevelStatis- tics	データベースサーバのマルチプログラミングレベルの自動 調整に関するメッセージが、データベースサーバのメッ セージログに表示されるかどうかを示す値を返します。
AvailIO	現在使用可能な I/O 制御ブロックの数を返します。
BuildChange	このプロパティはシステムで使用するために予約されてい ます。このプロパティの設定は変更しないでください。
BuildClient	このプロパティはシステムで使用するために予約されてい ます。このプロパティの設定は変更しないでください。

プロパティ名	説明
BuildProduction	データベースサーバが、実際の運用で使用するようコンパイルされている場合は Yes を返し、デバッグ用のビルドである場合は No を返します。
BuildReproducible	このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。
BytesReceived	クライアント/サーバ通信中に受信したバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。
BytesReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したバイト数を返します (この値は、圧縮が無効の場合は BytesReceived の値と同じ)。
BytesSent	クライアント/サーバ通信中に送信されたバイト数を返します。この値は、HTTP 接続と HTTPS 接続の場合に更新されます。
BytesSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたバイト数を返します (この値は、圧縮が無効の場合は BytesSent の値と同じ)。
CacheAllocated	サーバデータ構造に割り付けられたキャッシュページの数 を返します。
CacheFile	データベースファイルから取得したデータを格納する キャッシュページの数 を返します。
CacheFileDirty	ダーティな (書き込みが必要な) キャッシュページの数 を返します。
CacheFree	未使用のキャッシュページ数を返します。
CacheHits	データベースページのルックアップ回数を返します。
CachePanics	キャッシュマネージャが割り付けるページの検索に失敗した 回数を返します。
CachePinned	固定キャッシュページ数を返します。
CacheRead	キャッシュの読み込み数を返します。
CacheReplacements	キャッシュ内の置換されたページの数 を返します。
CacheScavenges	キャッシュマネージャが割り付けるページをスカベンジした 回数を返します。

プロパティ名	説明
CacheScavengeVisited	割り付けるページのスカベンジ中にアクセスしたページの数返します。
CacheSizingStatistics	キャッシュのサイズを変更するときサーバがキャッシュサイズ設定の統計を表示する場合は Yes を返し、それ以外の場合は No を返します。
CarverHeapPages	クエリ最適化など、短期的な目的に使用されたヒープページの数返します。
CharSet	データベースサーバが使用している CHAR 文字セットを返します。
ClientStmtCacheHits	クライアントに文がキャッシュされていることが理由で、不要な準備文の数返します。これは、クライアントでの文のキャッシュが無効になった場合に必要となる、追加の準備文の数です。
ClientStmtCacheMisses	クライアントのキャッシュに存在した文の中で、再び準備された文の数返します。これは、キャッシュされた文が再使用できるか検討されたものの、スキーマの変更、データベースのオプション設定の変更、DROP VARIABLE 文によって再使用できなかった回数です。
CollectStatistics	データベースサーバがパフォーマンス統計を収集するかどうかを示す値として、Yes または No を返します。
CommandLine	データベースサーバの起動に使用されたコマンドライン引数返します。 -ek オプションを使用してデータベース用の暗号化キーを指定した場合、キーはこのプロパティで返される値にあるアスタリスクの定数文字列で置き換えられます。
Commit	処理されたコミット要求の数返します。
CompactPlatformVersion	PlatformVer プロパティの縮小バージョン返します。
CompanyName	このソフトウェアを所有している会社の名前返します。

プロパティ名	説明
ConnectedTime	すべての接続が、これまでデータベースサーバに接続してきた合計の時間数を秒単位で返します。 要求が接続に対して完了したとき、または接続が切断されたときにだけ値が更新されます。その結果、値は、データベースサーバ内で長いことアイドル状態であるか、実行中でビジーである接続に対しては遅れる可能性があります。値には、データベースイベントやバックグラウンドサーバ接続(データベースクリーナなど)といった任意の接続によって増えた時間も含まれます。
ConnsDisabled	新しい接続を禁止するサーバオプションの現在の設定を示す値として、Yes または No を返します。
ConsoleLogFile	-o オプションが指定されている場合は、データベースサーバメッセージがロギングされるファイルの名前を返します。それ以外の場合は、空の文字列を返します。
ConsoleLogMaxSize	データベースサーバメッセージのロギングに使用するファイルの最大サイズ(バイト単位)を返します。
CurrentCacheSize	現在のキャッシュサイズ(キロバイト単位)を返します。
CurrentMultiProgrammingLevel	データベースサーバが同時に処理できる現在のタスク数を返します。
Cursor	現在データベースサーバが保持している宣言されたカーソルの数を返します。
CursorOpen	現在データベースサーバが保持している開いたカーソルの数を返します。
DebuggingInformation	サーバがトラブルシューティング用の診断メッセージを表示する場合は Yes を返し、それ以外の場合は No を返します。
DefaultCollation	新規データベースに使用される照合を返します(明示的に指定されている照合がない場合)。
DefaultNcharCollation	サーバコンピュータのデフォルト NCHAR 照合の名前を返します(ICU がインストールされている場合は UCA、インストールされていない場合は UTF8BIN)。
DiskRead	ディスクの読み込み数を返します。
DiskReadHintScatterLimit	分散読み込みヒントのサイズ制限(バイト単位)を返します。

プロパティ名	説明
DiskRetryRead	ディスクの読み込みリトライ数を返します。
DiskRetryReadScatter	分散読み込みのためのディスク読み込みリトライ数を返します。
DiskRetryWrite	ディスクの書き込みリトライ数を返します。
DiskSandbox	On または Off を返し、データベースの読み込み／書き込みファイル操作が、メインのデータベースファイルがあるディレクトリに制限されているかどうかを示します。
EventTypeDesc	特定のイベントタイプ ID に関連付けられたイベントタイプの説明を返します。
EventTypeName	特定のイベントタイプ ID に関連付けられたシステムイベントタイプの名前を返します。
ExchangeTasks	クエリの並列実行に現在使用されているタスクの数を返します。
ExchangeTasksCompleted	データベースサーバが起動された後、クエリ内並列処理に使用された内部タスクの総数を返します。
FipsMode	データベースサーバの起動時に <code>-fips</code> オプションが使用された場合は Yes を返し、それ以外の場合は No を返します。
FirstOption	データベースオプションに対応する最初の接続プロパティを表す番号を返します。
FreeBuffers	使用できるネットワークバッファの数を返します。
FunctionMaxParms	関数で指定可能なパラメータの最大数を返します。関数は、正の整数である <i>function-number</i> によって指定される値で識別されます。次に例を示します。 <pre>SELECT PROPERTY ('FunctionMaxParms', function-number);</pre> <i>function-number</i> は、リリース間で変更されることがあります。

プロパティ名	説明
FunctionMinParms	関数で指定しなければならないパラメータの最小数を返します。関数は、正の整数である <i>function-number</i> によって指定される値で識別されます。次に例を示します。 <pre>SELECT PROPERTY ('FunctionMinParms', function-number);</pre> <i>function-number</i> は、リリース間で変更されることがあります。
FunctionName	正の整数である <i>function-number</i> によって指定される値で識別される関数名を返します。 <pre>SELECT PROPERTY ('FunctionName', function- number);</pre> <i>function-number</i> は、リリース間で変更されることがあります。
HeapsCarver	クエリ最適化などの短期的な目的に使用されたヒープの数を返します。
HeapsLocked	キャッシュ内で現在ロックされている再配置可能なヒープの数を返します。
HeapsQuery	クエリ処理 (ハッシュ操作とソート操作) に使用されるヒープの数を返します。
HeapsRelocatable	再配置可能なヒープの数を返します。
HttpAddresses	サーバが HTTP 接続で受信するセミコロンで区切られた TCP/IP アドレスのリストを返します。次に例を示します。 <pre>(::1):80;127.0.0.1:80</pre>
HttpNumActiveReq	HTTP 要求をアクティブに処理している HTTP 接続数を返します。応答を送信した HTTP 接続は含まれません。
HttpNumConnections	データベースサーバで現在開いている HTTP 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。
HttpNumSessions	データベースサーバ内のアクティブおよび休止状態の HTTP セッション数を返します。
HttpPorts	Web サーバの HTTP ポート番号をカンマで区切られたリストで返します。

プロパティ名	説明
HttpsAddresses	サーバが HTTPS 接続で受信するセミコロンで区切られた TCP/IP アドレスのリストを返します。次に例を示します。 <code>(::1):443;127.0.0.1:443</code>
HttpsNumActiveReq	HTTPS 要求をアクティブに処理しているセキュア HTTPS 接続数を返します。応答を送信した HTTPS 接続は含まれません。
HttpsNumConnections	データベースサーバで現在開いている HTTPS 接続数を返します。この接続は、要求をアクティブに処理しているか、長期 (keep-alive) 接続のキューで待機中である可能性があります。
HttpsPorts	Web サーバの HTTPS ポート番号をカンマで区切られたリストで返します。
IdleTimeout	デフォルトのアイドルタイムアウトを返します。
IPAddressMonitorPeriod	データベースサーバが新しい IP アドレスをチェックする時間 (秒単位) を返します。
IsAesniAvailable	データベースサーバコンピュータの CPU が Intel AES-NI 命令セットをサポートしており、このコンピュータが、サポートされているオペレーティングシステムを実行している場合は Yes を、それ以外の場合は No を返します。
IsFipsAvailable	FIPS 認定の DLL がインストールされている場合は Yes を返し、それ以外の場合は No を返します。
IsIQ	このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。
IsNetworkServer	ネットワークデータベースサーバに接続している場合は Yes を返し、パーソナルデータベースサーバに接続している場合は No を返します。

プロパティ名	説明
IsPortableDevice	データベースサーバがラップトップ、ノートブック、またはその他のポータブルデバイスで実行されている場合は Yes、それ以外の場合は No を返します。Windows Mobile では常に Yes が返されます。VMWare は考慮されないため、ラップトップで実行されている VM で実行中のデータベースサーバでは No が返されます。 Windows では、デバイスがポータブルかどうかを判別できない場合、このプロパティは NULL を返します。 UNIX の場合、このプロパティは常に NULL を返します。
IsRsaAvailable	RSA DLL がインストールされている場合は Yes を返し、それ以外の場合は No を返します。
IsRuntimeServer	データベースサーバのすべてのバージョンについて No を返します。
IsService	データベースサーバがサービスとして実行されている場合は Yes、それ以外の場合は No を返します。
Language	サーバのロケール言語を返します。
LastConnection-Property	最後の接続プロパティを表す番号を返します。
LastDatabaseProperty	最後のデータベースプロパティを表す番号を返します。
LastOption	データベースオプションに対応する最後の接続プロパティを表す番号を返します。
LastServerProperty	最後のサーバプロパティを表す番号を返します。
LegalCopyright	ソフトウェアの著作権を示す文字列を返します。
LegalTrademarks	ソフトウェアの商標を示す文字列を返します。
LicenseCount	ライセンスされたシートまたはプロセッサの数を返します。
LicensedCompany	ライセンスされた会社の名前を返します。
LicensedUser	ライセンスされたユーザの名前を返します。
LicenseKey	内部でのみ使用。

プロパティ名	説明
LicenseType	ライセンスタイプを返します。ネットワークシート (per-seat) または CPU ベースのいずれかです。
LivenessTimeout	クライアント活性タイムアウトのデフォルトを返します。
LockedCursorPages	メモリ内でカーソルヒープを固定するために使用されているページ数を返します。
LockedHeapPages	キャッシュ内でロックされているヒープページの数を返します。
MachineName	データベースサーバを実行しているコンピュータの名前を返します。通常は、コンピュータのホスト名です。
MainHeapBytes	グローバルサーバデータ構造に使用されているバイト数を返します。
MainHeapPages	グローバルサーバデータ構造に使用されているページ数を返します。
MapPhysicalMemoryEng	Address Windowing Extensions を使用して、キャッシュ内の物理メモリにマッピングされているデータベースページのアドレス領域ウィンドウの数を返します。
MaxCacheSize	許容最大キャッシュサイズ(キロバイト単位)を返します。
MaxConnections	サーバが許容する同時接続の最大数を返します。デフォルト値は、パーソナルサーバの場合は 10、ネットワークサーバの場合は約 32000 です。この値を小さくするには、-gm サーバオプションを使用します。 通常は、コンピュータリソースの制約から、ネットワークサーバへの接続の最大数はデフォルト値より小さくなります。
MaxEventType	有効な最大イベントタイプ ID を返します。
MaxMessage	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウから取得できる行番号の現在の最大値を返します。この値は、データベースサーバメッセージウィンドウに最後に表示されたメッセージを表します。
MaxMultiProgrammingLevel	データベースサーバが同時に処理できるタスクの最大数を返します。

プロパティ名	説明
MaxRemoteCapability	有効な最大機能 ID を返します。
Message	<p>このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウから取得したメッセージ行を返します。行の先頭にはメッセージが表示された日付と時刻が追加されています。2 番目のパラメータは行番号を指定します。</p> <p>PROPERTY ("message") によって返された値が、データベースサーバメッセージウィンドウに書き込まれた出力の最初の行です。PROPERTY ("message", n) を呼び出すと、サーバ出力の n 行目 (0 を最初の行とする) が返されます。バッファは有限であるため、メッセージの生成とともに最初の行が削除されていき、メモリからなくなる場合があります。この場合は、NULL が返されます。</p>
MessageCategoryLimit	sa_server_messages システムプロシージャを使用して取得できる重要度とカテゴリそれぞれの最小メッセージ数を返します。デフォルト値は 400 です。
MessageText	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウ内の指定された行番号に対応するテキスト (日付と時刻を含まない) を返します。2 番目のパラメータは行番号を指定します。
MessageTime	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウ内の指定された行番号に対応する日付と時刻を返します。2 番目のパラメータは行番号を指定します。
MessageWindowSize	このプロパティは使用されなくなりました。データベースサーバメッセージウィンドウから取得できる行の最大数を返します。
MinCacheSize	許容最小キャッシュサイズ (キロバイト単位) を返します。
MinMultiProgrammingLevel	サーバが同時に処理できるタスクの最小数を返します。
MultiPacketsReceived	クライアント/サーバ通信中に受信したマルチパケット要求の数を返します。

プロパティ名	説明
MultiPacketsSent	クライアント/サーバ通信中に送信されたマルチパケット応答の数を返します。
MultiPageAllocs	マルチページキャッシュ割り付けの数を返します。
MultiProgrammingLevel	サーバが一度に処理できる同時タスクの最大数を返します。この値を超える数の同時タスクが存在する場合、要求はキューイングされます。この動作を変更するには、-gn サーバオプションを使用します。
Name	データベースへの接続に使用するサーバの別名を返す (別名が指定されている場合)。別名が指定されていない場合は、実際のサーバ名を返します。 クライアントがコピーノードに接続され、接続文字列で NodeType=COPY が指定されている場合、このプロパティの値は、ServerName 接続パラメータでクライアントの接続文字列に指定されたデータベースサーバ名と異なることがあります。
NativeProcessorArchitecture	ソフトウェアが実行されているネイティブプロセッサタイプを表す文字列を返します。プロセッサをエミュレートできるプラットフォーム (x64 や x86 など) の場合は、OS アーキテクチャタイプではなく、実際のプロセッサタイプが返されます。 このプロパティが返す値は、オペレーティングシステムが 32 ビットと 64 ビットのいずれであるかを示す値ではありません。 値には、次のものがあります。 Windows Mobile – ARM Solaris – SPARC または X86_64 AIX – PPC HP – IA64 Linux – X86 または X86_64 X86 は 32 ビットハードウェアアーキテクチャを表します。X86_64 は 64 ビットハードウェアアーキテクチャを表します。 サポートされるプラットフォームの詳細については、 http://www.sybase.com/detail?id=1002288 を参照してください。

プロパティ名	説明
NumLogicalProcessors	サーバコンピュータ上の論理プロセッサ数 (コアとハイパースレッドを含む) を返します。
NumLogicalProcessorsUsed	データベースサーバが使用する論理プロセッサの数を返します。Windows の場合、使用する論理プロセッサ数を変更するには、-gtc オプションを使用します。
NumPhysicalProcessors	サーバコンピュータで有効となっている物理プロセッサの数を返します。この値は、NumLogicalProcessors の値を物理プロセッサあたりのコア数またはハイパースレッド数で割った数に相当します。Windows 以外のプラットフォームでは、コアまたはハイパースレッドが物理プロセッサとして計数される場合があります。
NumPhysicalProcessorsUsed	データベースサーバが使用する物理プロセッサの数を返します。パーソナルサーバでは、プラットフォームによっては、1つの CPU 上のコアが4つに制限されます。Windows でネットワークデータベースサーバが使用する物理プロセッサの数を変更するには、-gt オプションを使用します。
ObjectType	データベースオブジェクトのタイプを返します。この値は、SYSOBJECT システムビューによって使用されます。
OmniIdentifier	このプロパティはシステムで使用するために予約されています。このプロパティの設定は変更しないでください。
PacketsReceived	受信したクライアント/サーバ通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsReceivedUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に受信したパケットの数を返します(この値は、圧縮が無効の場合は PacketsReceived の値と同じ)。
PacketsSent	送信されたクライアント/サーバ通信パケットの数を返します。この値は、HTTP 接続と HTTPS 接続の場合は更新されません。
PacketsSentUncomp	圧縮が無効になっている場合にクライアント/サーバ通信中に送信されたパケットの数を返します(この値は、圧縮が無効の場合は PacketsSent の値と同じ)。

プロパティ名	説明
PageSize	データベースサーバのキャッシュページのサイズを返します。 <code>-gp</code> オプションを使用して設定可能です。設定しない場合はコマンドラインで指定したデータベースの最大ページサイズです。
PeakCacheSize	現在のセッションでキャッシュが到達した最大値 (キロバイト単位) を返します。
Platform	ソフトウェアが実行されているオペレーティングシステムを返します。たとえば、Windows XP を実行中の場合、このプロパティは WindowsXP を返します。
PlatformVer	ソフトウェアを実行しているオペレーティングシステムの名前を返します。この名前には、ビルド番号やサービスパックなどの情報も含まれています。
PrepStmt	現在データベースサーバが保持している準備文の数を返します。
ProcessCPU	データベースサーバプロセスにおける CPU の使用量を返します。値は秒数単位です。このプロパティは Windows と UNIX でサポートされています。このプロパティは Windows Mobile ではサポートされていません。 このプロパティが返す値は、データベースサーバが起動してからの累積値です。この値は Windows タスクマネージャや Windows パフォーマンスモニタなどのアプリケーションに表示される瞬時値とは一致しません。
ProcessCPUSystem	データベースサーバプロセス CPU におけるシステム CPU の使用量を返します。これは、データベースサーバがオペレーティングシステムカーネルの内部で使用した CPU 時間を表します。値は秒数単位です。このプロパティは Windows と UNIX でサポートされています。このプロパティは Windows Mobile ではサポートされていません。 このプロパティが返す値は、データベースサーバが起動してからの累積値です。この値は Windows タスクマネージャやパフォーマンスモニタなどのアプリケーションに表示される瞬時値とは一致しません。

プロパティ名	説明
ProcessCPUUser	<p>データベースサーバプロセスにおけるユーザ CPU の使用量を返します。値は秒数単位です。これには、データベースサーバがオペレーティングシステムカーネルの内部で使用した CPU 時間は含まれません。このプロパティは Windows と UNIX でサポートされています。このプロパティは Windows Mobile ではサポートされていません。</p> <p>このプロパティが返す値は、データベースサーバが起動してからの累積値です。この値は Windows タスクマネージャやパフォーマンスモニタなどのアプリケーションに表示される瞬時値とは一致しません。</p>
ProcessID	データベースサーバプロセスのプロセス ID を返します。
ProcessorAffinity	-gta オプション、または sa_server_option システムプロシージャと ProcessorAffinity オプションによる指定に従ってデータベースサーバが使用する論理プロセッサを返します。
ProcessorArchitecture	<p>現在のソフトウェアが構築された対象のプロセッサタイプを表す文字列を返します。次のような値があります。</p> <p>Windows Mobile – ARM Solaris – SPARC または X86_64 AIX - PPC HP - IA64 Linux - X86 または X86_64</p> <p>X86 は 32 ビットデータベースサーバを表します。X86_64 は 64 ビットデータベースサーバを表します。</p>
ProductName	ソフトウェアの名前を返します。
ProductVersion	実行中のソフトウェアのバージョンを返します。
ProfileFilterConn	特定の接続をプロファイルするプロシージャがオンの場合にモニタされる接続の ID を返します。それ以外の場合、空の文字列を返します。ユーザによるプロシージャのプロファイルは、sa_server_option プロシージャを使って制御します。

プロパティ名	説明
ProfileFilterUser	特定のユーザのプロシージャプロファイルがオンの場合にモニタされるユーザ ID を返します。それ以外の場合、空の文字列を返します。ユーザによるプロシージャのプロファイルは、sa_server_option プロシージャを使って制御します。
QueryHeapPages	クエリ処理 (ハッシュ操作とソート操作) に使用されるキャッシュページの数 を返します。
QueryMemActive-Curr	クエリメモリをアクティブに使用する要求数を返します。
QueryMemActiveEst	安定した状態のデータベースサーバで、クエリメモリをアクティブに使用する要求の推定平均数を返します。
QueryMemActiveMax	クエリメモリをアクティブに使用できる要求の最大数を返します。
QueryMemExtraAvail	基本的なメモリ集約型の付与を超えて付与可能なメモリ量を返します。
QueryMemGrantBase	すべての要求に付与される最小メモリ量を返します。
QueryMemGrantBaseMI	メモリ集約型の要求に付与される最小メモリ量を返します。
QueryMemGrantExtra	アクティブなメモリ集約型の要求間で QueryMemGrantBaseMI の値を超えて分配されるクエリメモリページの数 を返します。
QueryMemGrant-Failed	要求がクエリメモリ待ちになり、取得できなかった合計回数を返します。
QueryMemGrant-Granted	要求に現在付与されているページ数を返します。
QueryMemGrantRequested	要求がクエリメモリを取得しようとした合計回数を返します。
QueryMemGrantWait-ed	要求がクエリメモリ待ちになった合計回数を返します。
QueryMemGrant-Waiting	クエリメモリ待ちになっている現在の要求数を返します。
QueryMemPages	クエリ実行アルゴリズムで使用可能なメモリの容量をページ数で返します。

プロパティ名	説明
QueryMemPercentOfCache	クエリ実行アルゴリズムで使用可能なメモリの容量を最大キャッシュサイズの割合 (パーセント) で返します。
QuittingTime	サーバのシャットダウン時間を返します。シャットダウン時間が指定されていない場合は、none を返します。
RememberLastPlan	オプティマイザが返す最後のクエリ最適化プランをサーバが記録する場合は、Yes を返します。
RememberLastStatement	サーバが各接続で準備された最後の文を記録している場合は Yes を返し、それ以外の場合は No を返します。
RemoteCapability	特定の機能 ID に関連付けられたリモート機能名を返します。
RemoteputWait	通信パケットの送信中にサーバがブロックしなければならなかった回数を返します。通常ブロックは、クライアントまたはネットワークが受信可能な速度を超えてデータベースサーバがデータを送信した場合にのみ起こります。エラー条件は示しません。
Req	サーバが新しい要求を処理するか、または既存の要求の処理を続行できる状態になった回数を返します。
ReqCountActive	サーバが新しい要求を処理するか、または既存の要求の処理を続行できる状態になった回数を返します。
ReqCountBlockContention	内部サーバのリソースの競合により接続がブロックされた回数を返します。
ReqCountBlockIO	IO 要求の完了を待つ間に接続がブロックされた回数を返します。
ReqCountBlockLock	別の接続が保持しているローロックを待つ間に接続がブロックされた回数を返します。
ReqCountUnscheduled	サーバスレッドが接続を処理するのを待つ間に接続がブロックされた回数を返します。
ReqTimeActive	サーバが要求の処理で直接費やした合計時間を返します。
ReqTimeBlockContention	内部サーバのリソースの競合により接続がブロックされた合計時間を返します。
ReqTimeBlockIO	IO 要求の完了を待つ間に接続がブロックされた合計時間を返します。

プロパティ名	説明
ReqTimeBlockLock	別の接続が保持しているローロックを待つ間に接続がブロックされた合計時間を返します。
ReqTimeUnscheduled	サーバスレッドが接続を処理するのを待つ間に接続がブロックされた合計時間を返します。
RequestFilterConn	接続のロギング情報をフィルタしている場合は、その接続の ID を返し、それ以外の場合は -1 を返します。
RequestFilterDB	データベースのロギング情報をフィルタしている場合は、そのデータベースの ID を返し、それ以外の場合は -1 を返します。
RequestLogFile	要求ロギングファイルの名前を返します。要求がロギングされていない場合は、空の文字列を返します。
RequestLogging	現在の要求ロギング設定を示す値として、SQL、PLAN、HOSTVARS、PROCEDURES、TRIGGERS、OTHER、BLOCKS、REPLACE、ALL、NONE のいずれかを返します。
RequestLogMaxSize	要求ログファイルの最大サイズを返します。
RequestLogNumFiles	保管される要求ログファイルの数を返します。
RequestsReceived	クライアント/サーバ通信要求またはラウンドトリップの数を返します。このプロパティでは、PacketsReceived とは異なり、マルチパケット要求を1つの要求として数え、活性パケットを計数の対象から除外します。
RequestTiming	要求タイミングがオンになっている場合は Yes を返し、オフになっている場合は No を返します。要求タイミングをオンにするには、-zr データベースサーバオプションを使用します。
Rlbnk	処理されたロールバック要求の数を返します。
SendFail	通信プロトコルがパケット送信に失敗した回数を返します。

プロパティ名	説明
ServerEdition	<p>データベースサーバのタイプを記述する単語のスペースで区切られたリストを返します。次のような値があります。</p> <p>Evaluation Developer Web Educational Standard Advanced Workgroup OEM Authenticated</p> <p>次のいずれかの機能のためのライセンスを別途保有している場合は、返されるライセンス文字列に、次の該当する文字列が追加されます。</p> <p>HighAvailability InMemory FIPS</p>
ServerName	<p>実際のサーバ名 (サーバの代替名でない) を返します。この値によって、稼働しているサーバのうちどれがデータベースミラーリング構成のプライマリサーバであるかを特定できます。</p>
StartDBPermission	<p>-gd サーバオプションの設定を返します。値は、DBA、all、none のいずれかです。</p>
StartTime	<p>サーバが起動された日付/時刻を返します。</p>
StreamsUsed	<p>使用中のデータベースサーバストリーム数を返します。</p>
TcpIpAddresses	<p>サーバが Command Sequence と TDS 接続で受信するセミコロンの区切られた TCP/IP アドレスのリストを返します。次に例を示します。</p> <p>(::1):2638;127.0.0.1:2638</p>
TempDir	<p>テンポラリファイルが格納されているサーバ上のディレクトリを返します。</p>

プロパティ名	説明
ThreadDeadlocksAvoided	スレッドデッドロックエラーが検出されたがクライアントアプリケーションにレポートされなかった回数を返します。データベースサーバが起動すると、このプロパティの値は 0 になります。 スレッドデッドロックエラーを返すのを防ぐため、データベースサーバはマルチプログラミングレベルを動的に上げます。マルチプログラミングレベルを上げることができない場合、スレッドデッドロックエラーはクライアントアプリケーションに返され、ThreadDeadlocksReported プロパティが増分されます。パーソナルサーバ上ではマルチプログラミングレベルの動的な調整がデフォルトで無効になっているため、ThreadDeadlocksAvoided プロパティは常に 0 になります。
ThreadDeadlocksReported	スレッドデッドロックエラーがクライアントアプリケーションにレポートされた回数を返します。データベースサーバが起動すると、このプロパティの値は 0 になります。
TimeZoneAdjustment	サーバのローカル時間を表示するために協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある時間 (分単位) を返します。
TotalBuffers	ネットワークバッファの総数を返します。
UniqueClientAddresses	ネットワークサーバに接続しているユニークなクライアントネットワークアドレスの数を返します。
UnschReq	使用できるサーバワーカの解放を待つキューイングされている要求の数を返します。
UserDefinedCounterRate01	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate02	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。

プロパティ名	説明
UserDefinedCounterRate03	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate04	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRate05	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタは、カウンタ値の経時変化を表示します。
UserDefinedCounterRaw01	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw02	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw03	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
UserDefinedCounterRaw04	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。

プロパティ名	説明
UserDefinedCounterRaw05	ユーザが定義したパフォーマンスカウンタの現在の値を返します。このプロパティのセマンティックはクライアントアプリケーションによって定義されます。このカウンタは、パフォーマンスモニタからもアクセスできます。パフォーマンスモニタはカウンタの絶対値を表示します。
WebClientLogFile	Web サービスクライアントログファイルの名前を返します。
WebClientLogging	Web サービスクライアント情報がファイルにロギングされているかどうかを示す値を返します。

SQL および 外部環境ユーザ定義関数

SAP Sybase IQ には、ユーザ定義関数を作成するためのしくみが2つ用意されています。SQL 言語を使用して関数を記述するか、ESQL、ODBC、Java、Perl、または PHP 外部環境を使用できます。

SQL UDF と外部 C および C++ UDF を混同しないでください。外部 UDF には特別なライセンスが必要です。外部 UDF の詳細については、『ユーザ定義関数』マニュアルを参照してください。

SQL のユーザ定義関数

CREATE FUNCTION 文を使用して、独自の関数を SQL に実装できます。関数のデータ型は、**CREATE FUNCTION** 文内の **RETURN** 文で定義します。

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が使用されるのであれば、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は、CONTAINS の基準が無視されるため、CONTAINS 述部を使用しないでください。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。

Java 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が使用されるのであればどこでも、ユーザ定義関数として使用できます。

SQL 関数

インスタンスメソッドは、クラス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 クラスの作成とクラスメソッドの詳細については、『プログラミング』ガイドの「データベース内の Java」を参照してください。

その他の関数

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。

互換性

Adaptive Server では、COALESCE 関数、ISNULL 関数、および NULLIF 関数のみをサポートしています。

参照：

- ARGN 関数 [その他] (158 ページ)
- COALESCE 関数 [その他] (174 ページ)
- IFNULL 関数 [その他] (247 ページ)
- ISNULL 関数 [その他] (252 ページ)
- ISNUMERIC 関数 [その他] (253 ページ)
- NEWID 関数 [その他] (282 ページ)
- NULLIF 関数 [その他] (291 ページ)
- NUMBER 関数 [その他] (292 ページ)
- ROWID 関数 [その他] (334 ページ)

アルファベット順の関数リスト

この項では、各 SQL 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]

数値式の絶対値を返します。

構文

```
ABS ( numeric-expression )
```

パラメータ

パラメータ	説明
numeric-expression	絶対値を返す数値。

戻り値

数値式の絶対値。

数値式データ型	戻り値
INT	INT
FLOAT	FLOAT
DOUBLE	DOUBLE
NUMERIC	NUMERIC

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文は、値 66 を返します。

```
SELECT ABS ( -66 ) FROM iq_dummy
```

ACOS 関数 [数値]

数値式のアークコサインをラジアンで返します。

構文

```
ACOS ( numeric-expression )
```

パラメータ

表 8 : パラメータ

パラメータ	説明
numeric-expression	角度の余弦。

戻り値

DOUBLE

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文は、値 1.023945 を返します。

```
SELECT ACOS ( 0.52 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)
- ATAN2 関数 [数値] (162 ページ)
- ATAN 関数 [数値] (161 ページ)
- ASIN 関数 [数値] (160 ページ)
- COT 関数 [数値] (184 ページ)
- SIN 関数 [数値] (339 ページ)
- TAN 関数 [数値] (368 ページ)

ARGN 関数 [その他]

引数のリストから選択した引数を返します。

構文

```
ARGN ( integer-expression, expression [ , ... ] )
```

パラメータ

表 9: パラメータ

パラメータ	説明
integer-expression	式のリスト内での引数の位置。
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。

戻り値

integer-expression の値を *n* とした場合、引数リストの *n* 番目の引数 (1 から開始) を返します。

備考

integer-expression の値を *n* とした場合、引数リストの *n* 番目の引数 (1 から開始) 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 が返ります。式は、カンマで区切って入力します。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文は、値 6 を返します。

```
SELECT ARGN ( 6, 1,2,3,4,5,6 ) FROM iq_dummy
```

ASCII 関数 [文字列]

文字列式の先頭バイトの ASCII 値を整数で返します。

構文

```
ASCII ( string-expression )
```

パラメータ

パラメータ	説明
string-expression	文字列。

SQL 関数

戻り値
SMALLINT

備考
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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

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。

```
SELECT ASCII ( 'Z' ) FROM iq_dummy
```

ASIN 関数 [数値]

数値のアークサインをラジアンで返します。

構文
ASIN (*numeric-expression*)

パラメータ

パラメータ	説明
numeric-expression	角度のサイン。

戻り値
DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例
次の文は、値 0.546850 を返します。

```
SELECT ASIN ( 0.52 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)

- ATAN2 関数 [数値] (162 ページ)
- ATAN 関数 [数値] (161 ページ)
- ACOS 関数 [数値] (157 ページ)
- COT 関数 [数値] (184 ページ)
- SIN 関数 [数値] (339 ページ)
- TAN 関数 [数値] (368 ページ)

ATAN 関数 [数値]

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 ( numeric-expression )
```

パラメータ

パラメータ	説明
numeric-expression	角度のタンジェント。

戻り値

DOUBLE

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文は、値 0.479519 を返します。

```
SELECT ATAN( 0.52 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)
- ATAN2 関数 [数値] (162 ページ)
- ASIN 関数 [数値] (160 ページ)
- ACOS 関数 [数値] (157 ページ)
- COT 関数 [数値] (184 ページ)
- SIN 関数 [数値] (339 ページ)
- TAN 関数 [数値] (368 ページ)

ATAN2 関数 [数値]

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

パラメータ	説明
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。

戻り値

DOUBLE

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — ATAN2 は、Adaptive Server ではサポートされていません。

例

次の文は、値 0.00866644968879073143 を返します。

```
SELECT ATAN2( 0.52, 060 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)
- ATAN 関数 [数値] (161 ページ)
- ASIN 関数 [数値] (160 ページ)
- ACOS 関数 [数値] (157 ページ)
- COT 関数 [数値] (184 ページ)
- SIN 関数 [数値] (339 ページ)
- TAN 関数 [数値] (368 ページ)

AVG 関数 [集合]

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

パラメータ	説明
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。

戻り値

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。

引数が DOUBLE の場合は DOUBLE を、それ以外の場合は NUMERIC を返します。

備考

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

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server 互換。

例

次の文は、値 49988.6 を返します。

```
SELECT AVG ( salary ) FROM Employees
```

参照：

- COUNT 関数 [集合] (187 ページ)
- SUM 関数 [集合] (365 ページ)

BFIL 関数 [データ抽出]

個別の LONG BINARY セルと LONG VARCHAR セルをサーバ上の個別のオペレーティングシステムファイルに抽出します。

備考

IQ データ抽出機能には、個別の LONG BINARY セルと LONG VARCHAR セルをサーバ上の個別のオペレーティングシステムファイルに抽出できる **BFIL** 関数が含まれています。**BFIL** は、データ抽出機能と一緒に使用できるほか、単独でも使用できます。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。

BIGINTTOHEX 関数 [データ型変換]

10 進の整数をデータ型 VARCHAR(16) の 16 進数に変換して返します。

構文

```
BIGINTTOHEX ( integer-expression )
```

パラメータ

パラメータ	説明
integer-expression	16 進数に変換される整数。

備考

BIGINTTOHEX は、BIGINT データ型として評価される整数式を受け取り、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せん。

必要に応じて、変換は自動的に実行されます。小数値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、SAP 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

- SQL — ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase — Adaptive Server 互換。

例

次の文は、値 00000000000000009 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
```

参照：

- HEXTOBIGINT 関数 [データ型変換] (232 ページ)
- HEXTOINT 関数 [データ型変換] (234 ページ)
- INTTOHEX 関数 [データ型変換] (250 ページ)

BIT_LENGTH 関数 [文字列]

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

```
BIT_LENGTH( column-name )
```

パラメータ

パラメータ	説明
column-name	カラムの名前。

戻り値

INT

備考

引数が NULL の場合は、NULL 値を返します。

BIT_LENGTH 関数は、すべての SAP Sybase IQ データ型をサポートしています。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラジオブジェクトデータを使用できます。

非構造化データ分析の「関数のサポート」を参照してください。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — SQL Anywhere または Adaptive Server でサポートされていません。

参照：

- **BYTE_LENGTH** 関数 [文字列] (166 ページ)
- **CHAR_LENGTH** 関数 [文字列] (171 ページ)
- **COL_LENGTH** 関数 [システム] (175 ページ)
- **DATALength** 関数 [システム] (189 ページ)
- **LEN** 関数 [文字列] (262 ページ)
- **LENGTH** 関数 [文字列] (263 ページ)
- **OBJECT_NAME** 関数 [システム] (294 ページ)
- **OCTET_LENGTH** 関数 [文字列] (295 ページ)
- **STR_REPLACE** 関数 [文字列] (357 ページ)

BYTE_LENGTH 関数 [文字列]

文字列のバイト数を返します。

構文

```
BYTE_LENGTH ( string-expression )
```

パラメータ

パラメータ	説明
string-expression	長さが計算される文字列。

戻り値

INT

備考

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文字列の場合、**BYTE_LENGTH** の値は **CHAR_LENGTH** で返される文字数と異なります。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。

- **BYTE_LENGTH** 関数が LONG BINARY および LONG VARCHAR のカラムと変数の両方をサポートするのは、クエリの戻り値が 2GB 未満の場合のみです。返された LONG BINARY データまたは LONG VARCHAR データのバイト長が 2GB 以上の場合は、**BYTE_LENGTH** によって、**BYTE_LENGTH64** 関数を使用する必要があります。あることを示すエラーが返されます。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase — Adaptive Server では、サポートされていません。

例

次の文は、値 12 を返します。

```
SELECT BYTE_LENGTH( 'Test Message' ) FROM iq_dummy
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)

- COL_LENGTH 関数 [システム] (175 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- OCTET_LENGTH 関数 [文字列] (295 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)

BYTE_LENGTH64 関数

BYTE_LENGTH64 は、LONG BINARY カラムパラメータのバイト長を表す符号なし 64 ビット値を返します。

備考

BYTE_LENGTH64 は、LONG VARCHAR データ型、および任意のデータサイズの LONG BINARY 変数と LONG VARCHAR 変数もサポートしています。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラジオブジェクトデータを使用できます。

非構造化データ分析の「関数のサポート」を参照してください。

BYTE_SUBSTR64 関数と BYTE_SUBSTR 関数

BYTE_SUBSTR64 関数と **BYTE_SUBSTR** 関数は、LONG BINARY カラムパラメータの long binary 型バイト部分文字列を返します。

備考

BYTE_SUBSTR64 関数と **BYTE_SUBSTR** 関数は、LONG VARCHAR データ型、および任意のデータサイズの LONG BINARY 変数と LONG VARCHAR 変数もサポートしています。

CHAR_LENGTH64 は、任意のデータサイズの LONG VARCHAR 変数もサポートしています。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラジオブジェクトデータを使用できます。

『非構造化データ分析』の「**BYTE_SUBSTR64 関数と BYTE_SUBSTR 関数**」を参照してください。

CAST 関数 [データ型変換]

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

パラメータ	説明
expression	変換される式。
data type	ターゲットのデータ型。

戻り値

指定されたデータ型。

備考

文字列型への変換に長さを指定しない場合は、SAP Sybase IQ によって適切な長さが選択されます。DECIMAL 変換で精度も位取りも指定しない場合は、データベースサーバによって適切な値が選択されます。

NULL から NUMERIC への明示的な変換に対して、精度も位取りも指定しない場合は、デフォルトで NUMERIC(1,0) が設定されます。たとえば、

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

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

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server 互換。

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、データベース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 ( lname AS CHAR(5) ) FROM Customers
```

参照：

- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)
- DAYS 関数 [日付と時刻] (209 ページ)

CEIL 関数 [数値]

指定された式以上の最小の整数を返します。

CEIL は **CEILING** と同意語です。

構文

```
CEIL ( numeric-expression )
```

パラメータ

パラメータ	説明
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、 CEIL 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。

備考

指定された式について、**CEIL** 関数は 1 つの引数を取ります。たとえば、**CEIL (-123.45)** は -123 を返し、**CEIL (123.45)** は 124 を返します。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server Enterprise 互換。

参照：

- FLOOR 関数 [数値] (227 ページ)
- CEILING 関数 [数値] (170 ページ)

CEILING 関数 [数値]

数値の上限値 (その値以上の最も小さい整数) を返します。

CEIL は **CEILING** と同意語です。

構文

```
CEILING ( numeric-expression )
```

パラメータ

パラメータ	説明
numeric-expression	切り上げ値を計算する対象の数値。

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文は、値 60.00000 を返します。

```
SELECT CEILING( 59.84567 ) FROM iq_dummy
```

次の文は、値 123 を返します。

```
SELECT CEILING( 123 ) FROM iq_dummy
```

次の文は、値 124.00 を返します。

```
SELECT CEILING( 123.45 ) FROM iq_dummy
```

次の文は、値 -123.00 を返します。

```
SELECT CEILING( -123.45 ) FROM iq_dummy
```

参照：

- FLOOR 関数 [数値] (227 ページ)
- CEIL 関数 [数値] (169 ページ)

CHAR 関数 [文字列]

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

パラメータ	説明
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。

戻り値

VARCHAR

備考

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。

整数式の値が 255 より大きいか、0 より小さい場合、**CHAR** は NULL を返します。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文を実行すると、値 "Y" が返ります。

```
SELECT CHAR( 89 ) FROM iq_dummy
```

次の文を実行すると、値 "S" が返ります。

```
SELECT CHAR( 83 ) FROM iq_dummy
```

CHAR_LENGTH 関数 [文字列]

文字列の文字数を返します。

構文

```
CHAR_LENGTH ( string-expression )
```

パラメータ

パラメータ	説明
string-expression	長さが計算される文字列。

戻り値

INT

備考

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。

CHAR_LENGTH64 は、任意のデータサイズの LONG VARCHAR 変数もサポートしています。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。『非構造化データ分析』の「**CHAR_LENGTH 関数**」を参照してください。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server 互換。

例

次の文は、値 8 を返します。

```
SELECT CHAR_LENGTH( 'Chemical' ) FROM iq_dummy
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- OCTET_LENGTH 関数 [文字列] (295 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)

CHAR_LENGTH64 関数

CHAR_LENGTH64 関数は、LONG VARCHAR カラムパラメータの文字長(後続ブランクを含む)を表す符号なし 64 ビット値を返します。

備考

CHAR_LENGTH64 は、任意のデータサイズの LONG VARCHAR 変数もサポートしています。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラジオブジェクトデータを使用できます。『非構造化データ分析』の「CHAR_LENGTH64 関数」を参照してください。

CHARINDEX 関数 [文字列]

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

パラメータ	説明
<i>string-expression1</i>	検索する文字列。255 バイトまでの文字列を指定する。
<i>string-expression2</i>	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 になる。

戻り値

INT

備考

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、**CHARINDEX** は最初の文字列の位置を返します。

指定した文字列が検索対象の文字列に含まれない場合、**CHARINDEX** はゼロ (0) を返します。

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。

SQL 関数

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。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。『非構造化データ分析』の「CHARINDEX 関数」を参照してください。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server 互換。

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	Moir

参照：

- SUBSTRING 関数 [文字列] (362 ページ)

COALESCE 関数 [その他]

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

パラメータ	説明
expression	任意の式。

戻り値

ANY

標準と互換性

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server 互換。

例

次の文は、値 34 を返します。

```
SELECT COALESCE( NULL, 34, 13, 0 ) FROM iq_dummy
```

参照：

- ISNULL 関数 [その他] (252 ページ)

COL_LENGTH 関数 [システム]

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	説明
table-name	テーブル名。
column-name	カラム名。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

次の文は、カラム長 35 を返します。

```
SELECT COL_LENGTH ( 'CUSTOMERS', 'ADDRESS' ) FROM iq_dummy
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)

SQL 関数

- OCTET_LENGTH 関数 [文字列] (295 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)

COL_NAME 関数 [システム]

カラム名を返します。

構文

```
COL_NAME ( table-id, column-id [ , database-id ] )
```

パラメータ

表 10 : パラメータ

パラメータ	説明
table-id	テーブルのオブジェクト ID。
column-id	カラムの ID。
database-id	データベース ID。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

次の文は、カラム名 `lname` 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 `column_id` 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。

```
SELECT COL_NAME( 100209, 3, 0 ) FROM iq_dummy
```

次の文は、カラム名 **city** を返します。

```
SELECT COL_NAME ( 100209, 5 ) FROM iq_dummy
```

参照：

- DB_ID 関数 [システム] (211 ページ)
- DB_NAME 関数 [システム] (212 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)

- OBJECT_NAME 関数 [システム] (294 ページ)

CONNECTION_PROPERTY 関数 [システム]

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

```
CONNECTION_PROPERTY ( { integer-expression1 | string-expression }
... [ , integer-expression2 ] )
```

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。

パラメータ

パラメータ	説明
integer-expression1	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。integer-expression1 を指定する場合、これは接続プロパティ ID になります。これは、PROPERTY_NUMBER 関数を使用して確認できます。
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。

戻り値

VARCHAR

備考

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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

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( 'PrepStmnt' ) FROM iq_dummy
```

参照：

- PROPERTY 関数 [システム] (305 ページ)
- PROPERTY_NAME 関数 [システム] (307 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)

SQL 関数

- サーバで使用可能なプロパティ (105 ページ)
- 各データベースで使用可能なプロパティ (134 ページ)
- 接続プロパティ (104 ページ)
- sp_iqshowpsexec プロシージャ (579 ページ)
- sp_iqcontext プロシージャ (466 ページ)

CONVERT 関数 [データ型変換]

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

パラメータ	説明
data-type	変換後の式のデータ型。
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に変換 (あるいはその逆) する場合に、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指定する。

format-style 引数が指定されない場合は、データベースオプション設定が使用されます。

表 11 : CONVERT 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]

世紀なし (yy)	世紀あり (yyyy)	出力
8	108	hh:nn:ss
-	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]lyymmdd
-	13 または 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 または 120	yyyy-mm-dd hh:nn:ss (24 時間表示、ODBC 標準、年 4 桁)
-	21 または 121	yyyy-mm-dd hh:nn:ss:sss (24 時間表示、ODBC 標準 + ミリ秒、年 4 桁)
36	136	hh:nn:ss.ssssssAM (または PM)
37	137	hh:nn:ss.ssssss
38	138	mmm dd yy[yy] hh:nn:ss.ssssssAM (または PM)
39	139	mmm dd yy[yy] hh:nn:ss.ssssss
40	140	[yy]yy-mm-dd hh:nn:ss.ssssss
-	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))

次の表は、**CONVERT** フォーマットスタイルでの日付要素の省略形と値を示します。

省略形	日付要素	値
hh	時間	0 - 23
nn	分	0 - 59
ss	秒	0 - 59
sss	ミリ秒	0 - 999
sssss	マイクロ秒	0 - 999999

SQL 関数

省略形	日付要素	値
mmm	月	Jan から Dec
dd	日	1 - 31
yyyy	年	0001 - 9999
mm	月	1 - 12

戻り値

指定されたデータ型。

備考

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、非構造化データ分析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。

標準と互換性

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- Sybase - Adaptive Server および SQL Anywhere 互換。ただし、format-style 365 は SAP Sybase IQ でのみ使用できます。

例

format-style は次の文のように使用します。

```
SELECT CONVERT( CHAR( 20 ), order_date, 104 )
FROM sales_order
```

order_date
16.03.1993
20.03.1993
23.03.1993
25.03.1993
...

```
SELECT CONVERT( CHAR( 20 ), order_date, 7 )
FROM sales_order
```

order_date
mar 16, 93
mar 20, 93
mar 23, 93
mar 25, 93
...

```
SELECT order_datetime, CONVERT (CHAR(30), order_datetime, 40)
order_datetime40,
CONVERT (CHAR (30), order_datetime, 140) order_datetime140
FROM sales_order;
```

order_datetime	order_datetime40	order_datetime140
03/05/2009 01:03:05.123456	09-03-05 01:03:05.123456	2009-03-05 01:03:05.123456
03/05/2009 13:05:07.654321	09-03-05 13:05:07.654321	2009-03-05 13:05:07.654321

```
SELECT CONVERT (CHAR (50), DATETIME ('2009-11-03
11:10:42.033189'), 136) FROM iq_dummy が返す値 11:10:42.033189AM
```

```
SELECT CONVERT (CHAR (50), NOW (), 137) FROM iq_dummy が返す値
14:54:48.794122
```

次の文は format-style 365 の使用例です。この文では、DATE 型と 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; が返す値
'2004352'
```

```
SELECT CONVERT (INT, tab.date_col, 365) from tab; が返す値
2004352
```

```
SELECT CONVERT (DATE, tab.int_col, 365) FROM TAB; が返す値
2004-12-17
```

```
SELECT CONVERT (DATE, tab.char7_col, 365) FROM tab; が返す値
2004-12-17
```

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
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)

CORR 関数 [集合]

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*)

パラメータ

表 12 : パラメータ

パラメータ	説明
dependent-expression	independent-expression の影響を受ける変数。
independent-expression	結果に影響を与える変数。

戻り値

DOUBLE

備考

CORR 関数は、引数を DOUBLE に変換して、倍精度浮動小数点計算を実行し、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 であるペアをすべて排

除した後に、(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。

$$\text{COVAR_POP}(y, x) / (\text{STDDEV_POP}(x) * \text{STDDEV_POP}(y))$$

x は dependent-expression を表し、y は independent-expression を表します。

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句ではサポートされていま
せん。

構文 2 は、**SELECT** 文でウィンドウ関数として使用する場合の用法を示していま
す。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、
window-spec の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。主要な SQL に含まれない、SQL/foundation の機能
です。
- Sybase — SQL Anywhere 互換。

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。

```
SELECT CORR( Salary, ( YEAR( NOW( ) ) - YEAR( BirthDate ) ) ) FROM
Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

COS 関数 [数値]

数値の余弦をラジアンで返します。

構文

```
COS ( numeric-expression )
```

パラメータ

表 13：パラメータ

パラメータ	説明
numeric-expression	角度 (ラジアン)。

戻り値

この関数は、引数を DOUBLE に変換し、計算を倍精度浮動小数点で行い、結果を DOUBLE で返します。パラメータが NULL 値の場合、結果は NULL 値になります。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 0.86781 を返します。

```
SELECT COS ( 0.52 ) FROM iq_dummy
```

参照：

- ATAN2 関数 [数値] (162 ページ)
- ATAN 関数 [数値] (161 ページ)
- ASIN 関数 [数値] (160 ページ)
- ACOS 関数 [数値] (157 ページ)
- COT 関数 [数値] (184 ページ)
- SIN 関数 [数値] (339 ページ)
- TAN 関数 [数値] (368 ページ)

COT 関数 [数値]

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

表 14：パラメータ

パラメータ	説明
numeric-expression	角度 (ラジアン)。

戻り値

この関数は、引数を DOUBLE に変換し、計算を倍精度浮動小数点で行い、結果を DOUBLE で返します。パラメータが NULL 値の場合、結果は NULL 値になります。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、値 1.74653 を返します。

```
SELECT COT( 0.52 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)
- ATAN2 関数 [数値] (162 ページ)
- ATAN 関数 [数値] (161 ページ)
- ASIN 関数 [数値] (160 ページ)
- ACOS 関数 [数値] (157 ページ)
- SIN 関数 [数値] (339 ページ)
- TAN 関数 [数値] (368 ページ)

COVAR_POP 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

備考

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、**COVAR_POP** は NULL を返します。

SQL 関数

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 であるペアをすべて排除した後に、(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。

$$(\text{SUM}(x*y) - \text{SUM}(x) * \text{SUM}(y) / n) / n$$

x は dependent-expression を表し、y は independent-expression を表します。

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。主要な SQL に含まれない、SQL/foundation の機能です。
- Sybase — SQL Anywhere 互換。

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;
```

COVAR_SAMP 関数 [集合]

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)
```

パラメータ

パラメータ	説明
dependent-expression	独立した変数の影響を受ける変数。

パラメータ	説明
independent-expression	結果に影響を与える変数。

備考

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、**COVAR_SAMP** は NULL を返します。

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 であるペアをすべて排除した後に、(dependent-expression, independent-expression) のセットに適用されません。

```
(SUM(x*y) - SUM(x) * SUM(y) / n) / (n-1)
```

x は dependent-expression を表し、y は independent-expression を表します。

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。主要な SQL に含まれない、SQL/foundation の機能です。
- Sybase — SQL Anywhere 互換。

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 74782.946005 を返します。

```
SELECT COVAR_SAMP( Salary, ( 2008 - YEAR( BirthDate ) ) ) FROM Employees;
```

COUNT 関数 [集合]

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

```
COUNT ( * | expression | DISTINCT column-name )
```

パラメータ

パラメータ	説明
*	各グループのローの数を返す。
expression	各グループ内で、expression が NULL 値にならないローの数を返す。
DISTINCT column-name	column-name に含まれる、共通する値を除いた値の数を返す。値が NULL 値であるローはカウントされない。

注意： クエリ結果が表示されるときに、カラムヘッダに * は表示されません。次のように表示されます。

```
Count ()
```

戻り値

INT

標準と互換性

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server Enterprise 互換。

例

一意な各都市と、その都市を値に持つローの数を返します。

```
SELECT city , Count(*)
FROM Employees
GROUP BY city
```

参照：

- AVG 関数 [集合] (162 ページ)
- SUM 関数 [集合] (365 ページ)
- ウィンドウ集合関数の使用法 (89 ページ)

CUME_DIST 関数 [統計]

CUME_DIST 関数はランク付け統計関数です。1つの値について、ローのグループ内での相対位置を取得します。0～1の10進値を返します。

構文

```
CUME_DIST () OVER (window-spec)
```

戻り値

0～1のDOUBLE値

備考

SAP Sybase IQ は、 $CUME_DIST(x) = (S \text{ に含まれ、指定された順番で } x \text{ 以前にある値の数}) \div N$ という式を使用して、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。

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** 句で、*window-spec* の要素を指定できます。*window-spec* には **ORDER BY** 句を含める必要がありますが、**ROWS** や **RANGE** 句を含めることはできません。

注意： **DISTINCT** はサポートされていません。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、カリフォルニアに住む従業員の給与に関する累積分布を示す結果セットを返します。

```
SELECT DepartmentID, Surname, Salary, CUME_DIST() OVER (PARTITION BY
DepartmentID ORDER BY Salary DESC) "Rank" FROM Employees WHERE State IN
('CA');
```

次の結果セットが返されます。

表 15 : CUME_DIST の結果セット

DepartmentID	Surname	Salary	Rank
200	Savarino	72,300.000	0.333333
200	Clark	45,000.000	0.666667
200	Overbey	39,300.000	1.000000

DATALENGTH 関数 [システム]

式の長さをバイト数で返します。

構文

```
DATALENGTH ( expression )
```

パラメータ

パラメータ	説明
expression	expression には通常、カラム名を指定する。式が文字列定数の場合は、引用符で囲む。

戻り値

UNSIGNED INT

備考

表 16 : 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

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

company_name カラムの最も長い文字列の長さである 35 を返します。

```
SELECT MAX( DATALENGTH( company_name ) )
FROM Customers
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)

- OCTET_LENGTH 関数 [文字列] (295 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)

DATE 関数 [日付と時刻]

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

表 17: パラメータ

パラメータ	説明
expression	日付フォーマットに変換される値。通常、文字列を指定する。

戻り値

DATE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

DATEADD 関数 [日付と時刻]

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

パラメータ	説明
date part	日付に追加する日付要素。

パラメータ	説明
numeric-expression	日付に加算される日付の単位の数。numeric-expression には、任意の数値型を指定できる。この値は整数にトランケートされる。numeric-expression の最大マイクロ秒は 2147483647。これは、35:47.483647 (35 分 47 秒 483647 マイクロ秒) である。
date-expression	変更される日付。

戻り値

TIMESTAMP

備考

DATEADD は Transact-SQL 互換のデータ操作関数です。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 1995-11-02 00:00:00.000 を返します。

```
SELECT DATEADD( MONTH, 102, '1987/05/02' ) FROM iq_dummy
```

次の文は、値 2009-11-10 14:57:52.722016 を返します。

```
SELECT DATEADD(MICROSECOND, 15, '2009-11-10
14:57:52.722001') FROM iq_dummy
```

次の文は、値 1985-05-02 00:00:00.123456 を返します。

```
SELECT DATEADD(MICROSECOND, 123456, '1985/05/02')
FROM iq_dummy
```

次の文は、値 1985-05-01 23:59:59.876544 を返します。

```
SELECT DATEADD(MICROSECOND, -123456, '1985/05/02')
FROM iq_dummy
```

次の文は、値 2009-11-03 11:10:42.033192 を返します。

```
SELECT DATEADD(MCS, 2, '2009-11-03 11:10:42.033190')
FROM iq_dummy
```

参照：

- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)

- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- 日付要素 (97 ページ)

DATECEILING 関数 [日付と時刻]

渡された値より大きく、指定された粒度の最も近い値で、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

パラメータ	説明
date-part	日付に追加する日付要素。
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 パラメータによって指定された単位の何倍を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 200 マイクロ秒間隔または 10 分間隔で正規化するように指定できる。 multiple-expression が 0 に評価された場合、負の数値に評価された場合、明示的な NULL 定数である場合、指定された date-part に対して有効な値でない場合、SAP Sybase IQ はエラーを生成する。multiple-expression が NULL に評価された場合、関数の結果は NULL になる。

備考

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。

次の日付要素は、**DATECEILING** と互換性がありません。

- DayofYear
- WeekDay
- CalYearofWeek

SQL 関数

- CalWeekofYear
- CalDayofWeek

multiple-expression を microsecond、millisecond、second、minute、または hour 日付要素に指定した場合、SAP Sybase IQ では、粒度が次に大きい単位の最初から倍数が適用されるものとします。

- microsecond の倍数は現在の second から開始します。
- 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分という倍数を指定した場合、SAP Sybase IQ は現在の hour から始まる2分の間隔を適用します。

microsecond、millisecond、second、minute、および hour の日付要素の場合は、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。

- hour の場合、有効な *multiple-expression* の値は、1、2、3、4、6、8、12、24 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。
- 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。
- microsecond の場合、有効な *multiple-expression* の値は、次のとおりです。

1	40	400	4000	40000
2	50	500	5000	50000
4	64	625	6250	62500
5	80	800	8000	100000
8	100	1000	10000	125000
10	125	1250	12500	200000
16	160	1600	15625	250000
20	200	2000	20000	500000
25	250	2500	25000	1000000
32	320	3125	31250	

multiple-expression を day、week、month、quarter、または year の日付要素に指定した場合、SAP Sybase IQ は、最小の date 値 (0001-01-01)、最小の time 値

(00:00:00.000000)、または最小の datetime 値 (0001-01-01.00:00:00.000000) から間隔が開始するものとします。たとえば、10 日という倍数を指定した場合、SAP Sybase IQ では、0001-01-01 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位に均等に分割される倍数を指定する必要はありません。

SAP Sybase IQ が week 日付要素の倍数に丸める場合、date 値は常に Sunday (日曜日) です。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の文は、値 August 13, 2009 10:40.00.000 AM を返します。

```
SELECT DATECEILING( MI, 'August 13, 2009, 10:32.00.132AM', 10) FROM iq_dummy
```

次の文は、値 August 13, 2009 10:32.35.456800 AM を返します。

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM', 200 ) FROM iq_dummy
```

次の文は、値 August 13, 2009 10:32.35.600000 AM を返します。

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM', 200000 ) FROM iq_dummy
```

次の文は、値 August 13, 2009 10:32.35.456789 AM を返します。

```
SELECT DATECEILING( US, 'August 13, 2009, 10:32.35.456789AM') FROM iq_dummy
```

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- 日付要素 (97 ページ)

DATEDIFF 関数 [日付と時刻]

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

パラメータ	説明
date-part	インターバルを計算する日付要素を指定する。
date-expression1	インターバルの起点となる日付。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す。
date-expression2	インターバルの終点となる日付。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す。

戻り値

INT

備考

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** 関数は指定された2つの時刻の間にある深夜(午前0時)の回数を返します。このとき、2番目の日付は計算に含まれますが、最初の日付は含まれません。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
```

month 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月の初日の数を返します。このとき、2番目の日付は計算に含まれますが、最初の日付は含まれません。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番目の日付は計算に含まれますが、最初の日付は含まれません。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。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise 互換。

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
```

次の文は、4を返します。

```
SELECT DATEDIFF(MCS, '2009-11-03 11:10:42.033185',
'2009-11-03 11:10:42.033189') FROM iq_dummy
```

次の文は、15を返します。

SQL 関数

```
SELECT DATEDIFF(MICROSECOND, '2009-11-10
14:57:52.722001', '2009-11-10 14:57:52.722016')
FROM iq_dummy
```

次の文は、1,500,000 を返します。

```
SELECT DATEDIFF(MCS, '2000/07/07/07 07:07:06.277777',
'2000/07/07/07 07:07:07.777777') FROM iq_dummy
```

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- 日付要素 (97 ページ)

DATEFLOOR 関数 [日付と時刻]

渡された値を、指定された粒度で、指定された倍数の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。

構文

```
DATEFLOOR ( date-part, datetime-expression [,multiple-expression])
```

パラメータ

パラメータ	説明
date part	日付に追加する日付要素。
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の何倍を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 200 マイクロ秒間隔または 10 分間隔で正規化するように指定できる。 multiple-expression が 0 に評価された場合、負の数値に評価された場合、明示的な NULL 定数である場合、指定された date-part に対して有効な値でない場合、SAP Sybase IQ はエラーを生成する。multiple-expression が NULL に評価された場合、関数の結果は NULL になる。

備考

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。

次の日付要素は、**DATEFLOOR** と互換性がありません。

- DayofYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

multiple-expression を microsecond、millisecond、second、minute、または hour 日付要素に指定した場合、SAP Sybase IQ では、粒度が次に大きい単位の最初から倍数が適用されるものとします。

- microsecond の倍数は現在の second から開始します。
- 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分という倍数を指定した場合、SAP Sybase IQ は現在の hour から始まる 2分の間隔を適用します。

microsecond、millisecond、second、minute、および hour の日付要素の場合は、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。

- hour の場合、有効な *multiple-expression* の値は、1、2、3、4、6、8、12、24 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。
- 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。
- microsecond の場合、有効な *multiple-expression* の値は、次のとおりです。

1	40	400	4000	40000
---	----	-----	------	-------

2	50	500	5000	50000
4	64	625	6250	62500
5	80	800	8000	100000
8	100	1000	10000	125000
10	125	1250	12500	200000
16	160	1600	15625	250000
20	200	2000	20000	500000
25	250	2500	25000	1000000
32	320	3125	31250	

multiple-expression を day、week、month、quarter、または year の日付要素に指定した場合、SAP Sybase IQ は、最小の date 値 (0001-01-01)、最小の time 値 (00:00:00.000000)、または最小の datetime 値 (0001-01-01.00:00:00.000000) から間隔が開始するものとします。たとえば、10 日という倍数を指定した場合、SAP Sybase IQ では、0001-01-01 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位に均等に分割される倍数を指定する必要はありません。

SAP Sybase IQ が week 日付要素の倍数に丸める場合、date 値は常に Sunday (日曜日) です。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

- 次の文は、値 August 13, 2009 10:35:00.000 AM を返します。

```
SELECT DATEFLOOR( MINUTE, 'August 13, 2009 10:35:22.123AM') FROM iq_dummy
```

- 次の文は、値 August 13, 2009 10:32:35.456600 AM を返します。

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM', 200 ) FROM iq_dummy
```

- 次の文は、値 August 13, 2009 10:32:35.400000 AM を返します。

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM', 200000 ) FROM iq_dummy
```

- 次の文は、値 August 13, 2009 10:32:35.456789 AM を返します。

```
SELECT DATEFLOOR( US, 'August 13, 2009, 10:32:35.456789AM') FROM
iq_dummy
```

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- 日付要素 (97 ページ)

DATEFORMAT 関数 [日付と時刻]

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

パラメータ	説明
<i>datetime-expression</i>	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る。
<i>string-expression</i>	変換後の日付のフォーマット。

戻り値

VARCHAR

備考

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、SAP 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。

SQL 関数

```
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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の文は、"Jan 01, 1989" のような文字列の値を返します。

```
SELECT DATEFORMAT( start_date, '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
```

DATENAME 関数 [日付と時刻]

指定した日時の特定要素の名前を (月ならば "June" のように) 文字列で返します。

構文

```
DATENAME ( date-part, date-expression )
```

パラメータ

表 18 : パラメータ

パラメータ	説明
date-part	名前を返す日付要素。
date-expression	日付要素の名前を返す日付。必要な date-part を含めて指定する。

戻り値

VARCHAR

備考

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

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。

- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、値 May を返します。

```
SELECT DATENAME( MONTH , '1987/05/02' )
FROM iq_dummy
```

次の文は、値 722,001 を返します。

```
SELECT DATENAME(MICROSECOND, '2009-11-10
14:57:52.722001') FROM iq_dummy
```

次の文は、値 777,777 を返します。

```
SELECT DATENAME(MICROSECOND, '2000/07/07
07:07:07.777777') FROM iq_dummy
```

次の文は、値 33,189 を返します。

```
SELECT DATENAME(MCS, '2009-11-03 11:10:42.033189')
FROM iq_dummy
```

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- 日付要素 (97 ページ)

DATEPART 関数 [日付と時刻]

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 ( date-part, date-expression )
```

パラメータ

表 19 : パラメータ

パラメータ	説明
date-part	返される日付要素。
date-expression	要素を返す日付。date-part のフィールドを含める必要がある。

戻り値

INT

備考

DATE、**TIME**、**DTTM** インデックスは、一部の日付要素 (Calyearofweek、Calweekofyear、Caldayofweek、Dayofyear、Millisecond、Microsecond) をサポートしていません。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 5 を返します。

```
SELECT DATEPART( MONTH, '1987/05/02' )  
FROM iq_dummy
```

次の文は、値 722,001 を返します。

```
SELECT DATEPART(MICROSECOND, '2009-11-10  
14:57:52.722001') FROM iq_dummy
```

次の文は、値 777,777 を返します。

```
SELECT DATEPART(MICROSECOND, '2000/07/07  
07:07:07.777777') FROM iq_dummy
```

次の文は、値 33,189 を返します。

```
SELECT DATEPART(MCS, '2009-11-03 11:10:42.033189')  
FROM iq_dummy
```

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- DATEROUND 関数 [日付と時刻] (205 ページ)
- 日付要素 (97 ページ)

DATEROUND 関数 [日付と時刻]

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

パラメータ	説明
日付要素	返される日付要素。
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の何倍を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 200 マイクロ秒間隔または 10 分間隔で正規化するように指定できる。 <i>multiple-expression</i> が 0 に評価された場合、負の数値に評価された場合、明示的な NULL 定数である場合、指定された <i>date-part</i> に対して有効な値でない場合、SAP Sybase IQ はエラーを生成する。 <i>multiple-expression</i> が NULL に評価された場合、関数の結果は NULL になる。

備考

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。

次の日付要素は、**DATEROUND** と互換性がありません。

- DayOfYear
- WeekDay
- CalYearofWeek
- CalWeekofYear
- CalDayofWeek

multiple-expression を *microsecond*、*millisecond*、*second*、*minute*、または *hour* 日付要素に指定した場合、SAP Sybase IQ では、粒度が次に大きい単位の最初から倍数が適用されるものとします。

- *microsecond* の倍数は現在の *second* から開始します。
- *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分という倍数を指定した場合、SAP Sybase IQ は現在の *hour* から始まる2分の間隔を適用します。

microsecond、*millisecond*、*second*、*minute*、および *hour* の日付要素の場合は、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。

- *hour* の場合、有効な *multiple-expression* の値は、1、2、3、4、6、8、12、24 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。
- *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。
- *microsecond* の場合、有効な *multiple-expression* の値は、次のとおりです。

1	40	400	4000	40000
2	50	500	5000	50000
4	64	625	6250	62500
5	80	800	8000	100000
8	100	1000	10000	125000
10	125	1250	12500	200000
16	160	1600	15625	250000
20	200	2000	20000	500000
25	250	2500	25000	1000000
32	320	3125	31250	

multiple-expression を *day*、*week*、*month*、*quarter*、または *year* の日付要素に指定した場合、SAP Sybase IQ は、最小の *date* 値 (0001-01-01)、最小の *time* 値 (00:00:00.000000)、または最小の *datetime* 値 (0001-01-01-00:00:00.000000) から間隔

が開始するものとします。たとえば、10 日という倍数を指定した場合、SAP Sybase IQ では、0001-01-01 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位に均等に分割される倍数を指定する必要はありません。

SAP Sybase IQ が week 日付要素の倍数に丸める場合、date 値は常に Sunday (日曜日) です。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の文は、値 August 13, 2009, 10:30.000 AM を返します。

```
SELECT DATEROUND( MI, 'August 13, 2009 10:33.123AM', 10) FROM  
iq_dummy
```

次の文は、値 August 13, 2009 10:32:35.456600 AM を返します。

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456500AM', 200 )  
FROM iq_dummy
```

次の文は、値 August 13, 2009 10:32:35.456789 AM を返します。

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456789AM') FROM  
iq_dummy
```

次の文は、値 August 13, 2009 10:32:35.456400 AM を返します。

```
SELECT DATEROUND( US, 'August 13, 2009, 10:32:35.456499AM', 200 )  
FROM iq_dummy
```

参照：

- DATEADD 関数 [日付と時刻] (191 ページ)
- DATECEILING 関数 [日付と時刻] (193 ページ)
- DATEDIFF 関数 [日付と時刻] (196 ページ)
- DATEFLOOR 関数 [日付と時刻] (198 ページ)
- DATEPART 関数 [日付と時刻] (203 ページ)
- DATENAME 関数 [日付と時刻] (202 ページ)
- 日付要素 (97 ページ)

DATETIME 関数 [日付と時刻]

式をタイムスタンプに変換します。

構文

```
DATETIME ( expression )
```

パラメータ

パラメータ	説明
<i>expression</i>	変換される式。通常、文字列を指定する。変換エラーがあればレポートする。

戻り値

TIMESTAMP

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の文は、値 1998-09-09 12:12:12.000 のタイムスタンプを返します。

```
SELECT DATETIME( '1998-09-09 12:12:12.000' ) FROM iq_dummy
```

DAY 関数 [日付と時刻]

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

表 20 : パラメータ

パラメータ	説明
<i>date-expression</i>	日付。

戻り値

SMALLINT

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、値 12 を返します。

```
SELECT DAY( '2001-09-12' ) FROM iq_dummy
```

DAYNAME 関数 [日付と時刻]

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

表 21 : パラメータ

パラメータ	説明
date-expression	日付。

戻り値

VARCHAR

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise ではサポートされていません。

例

次の文は、値 Saturday を返します。

```
SELECT DAYNAME ( '1987/05/02' ) FROM iq_dummy
```

DAYS 関数 [日付と時刻]

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。

DAYS 関数では、時、分、秒は無視されます。

構文

```
DAYS ( datetime-expression )
| ( datetime-expression, datetime-expression )
| ( datetime-expression, integer-expression )
```

パラメータ

表 22 :

パラメータ	説明
<i>datetime-expression</i>	日時。
<i>integer-expression</i>	<i>datetime-expression</i> に追加する日数。 <i>integer-expression</i> が負の場合、指定された日数が日時から減算される。整数式を指定する場合、 <i>datetime-expression</i> は日付として明示的にキャストする必要がある。

戻り値

datetime-expression を 2 つ指定した場合は INT。

2 番目の引数が整数の場合は TIMESTAMP。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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、すなわち 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
```

参照：

- CAST 関数 [データ型変換] (168 ページ)

DB_ID 関数 [システム]

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。

パラメータ

パラメータ	説明
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、引用符で囲む。database-name を指定しない場合、現在のデータベースの ID 番号が返される。

戻り値

INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

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、唯一実行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
```

参照：

- COL_NAME 関数 [システム] (176 ページ)
- DB_NAME 関数 [システム] (212 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)

DB_NAME 関数 [システム]

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。

パラメータ

パラメータ	説明
database-id	データベースの ID。database-id には数値式を指定する。

戻り値

VARCHAR

備考

database-id 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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

次の文をデモデータベースに対して実行すると、データベース名 **iqdemo** が返されます。

```
SELECT DB_NAME( 0 ) FROM iq_dummy
```

参照：

- COL_NAME 関数 [システム] (176 ページ)
- DB_ID 関数 [システム] (211 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)

DB_PROPERTY 関数 [システム]

指定されたプロパティの値を返します。

構文

```
DB_PROPERTY ( { property-id | property-name }  
[ , { database-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。

パラメータ

表 23 : パラメータ

パラメータ	説明
property-id	データベースのプロパティ ID。
property-name	データベースのプロパティの名前。
database-id	データベースの ID 番号。DB_ID で取得できる。通常は、データベース名が使用される。
database-name	データベースの名前。DB_NAME で取得できる。

戻り値

VARCHAR

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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。

```
SELECT DB_PROPERTY( 'PAGESIZE' ) FROM iq_dummy
```

参照：

- COL_NAME 関数 [システム] (176 ページ)
- DB_ID 関数 [システム] (211 ページ)
- DB_NAME 関数 [システム] (212 ページ)

SQL 関数

- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)

DEGREES 関数 [数値]

数値をラジアンから度に変換します。

構文

```
DEGREES ( numeric-expression )
```

パラメータ

表 24 : パラメータ

パラメータ	説明
<i>numeric-expression</i>	角度 (ラジアン)。

戻り値

numeric-expression で指定された角度を返します。

DOUBLE

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、値 29.793805 を返します。

```
SELECT DEGREES ( 0.52 ) FROM iq_dummy
```

DENSE_RANK 関数 [統計]

グループ内の項目をランク付けします。

構文

```
DENSE_RANK () OVER ( ORDER BY expression [ ASC | DESC ] )
```

パラメータ

表 25 : パラメータ

パラメータ	説明
expression	ソートを指定する。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る。

戻り値

INTEGER

備考

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** の相違点は、順位が同じである場合に、**DENSE_RANK** ではランク順に隔たりが置かれられないことです。**RANK** では隔たりが置かれます。

DENSE_RANK には **OVER (ORDER BY)** 句が必須です。**ORDER BY** 句は、ランク付けを実行するパラメータと、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** 句の内部のみで使用するもので、**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で **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に対して処理を行うことを示します。結果セットは、**FROM**、**WHERE**、**GROUP BY**、**HAVING** の各句がすべて評価された後で返されるローです。**OVER** 句は、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 は、**SELECT** または **INSERT** 文の select リスト、および **SELECT** 文の **ORDER BY** 句でのみ使用できます。**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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の文は、**DENSE_RANK** 関数の使い方を示しています。

```
SELECT s_suppkey, DENSE_RANK()
OVER ( ORDER BY ( SUM(s_acctBal) DESC )
AS rank_dense FROM supplier GROUP BY s_suppkey;
```

s_suppkey	sum_acctBal	rank_dense
supplier#011	200,000	1
supplier#002	200,000	1
supplier#013	123,000	2
supplier#004	110,000	3
supplier#035	110,000	3
supplier#006	50,000	4
supplier#021	10,000	5

参照：

- RANK 関数 [統計] (311 ページ)

DIFFERENCE 関数 [文字列]

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 ( string-expression1, string-expression2 )
```

パラメータ

表 26 : パラメータ

パラメータ	説明
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。

戻り値

SMALLINT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

参照：

- SOUNDEX 関数 [文字列] (345 ページ)

DIFFERENCE 関数の例

以下の **DIFFERENCE** 関数の使用例を参考にしてください。

次の文は、値 4 を返します。

```
SELECT DIFFERENCE( 'Smith', 'Smith' ) FROM iq_dummy
```

次の文は、値 4 を返します。

```
SELECT DIFFERENCE( 'Smith', 'Smyth' ) FROM iq_dummy
```

次の文は、値 3 を返します。

```
SELECT DIFFERENCE( 'Smith', 'Sweeney' ) FROM iq_dummy
```

次の文は、値 2 を返します。

```
SELECT DIFFERENCE( 'Smith', 'Jones' ) FROM iq_dummy
```

次の文は、値 1 を返します。

```
SELECT DIFFERENCE( 'Smith', 'Rubin' ) FROM iq_dummy
```

次の文は、値 0 を返します。

```
SELECT DIFFERENCE( 'Smith', 'Wilkins' ) FROM iq_dummy
```

DOW 関数 [日付と時刻]

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 ( date-expression )
```

パラメータ

表 27 : パラメータ

パラメータ	説明
date-expression	日付。

戻り値

SMALLINT

備考

月曜日または別の日を週の最初の曜日にする必要がある場合は、DATE_FIRST_DAY_OF_WEEK オプションを使用してください。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise ではサポートされていません。

例

次の文は、値 5 を返します。

```
SELECT DOW( '1998-07-09' ) FROM iq_dummy
```

ERRORMSG 関数 [その他]

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

パラメータ	定義
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。

戻り値

エラーメッセージを含む文字列。

VARCHAR

備考

引数を指定しない場合は、現在のステータスのエラーメッセージが返されます。テーブル名やカラム名などが代入されます。

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 SAP Sybase IQ のエラーメッセージを返します。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、サポートされていません。

例

次の文は、SQLCODE -813 のエラーメッセージを返します。

```
select errormsg( -813 )
```

EVENT_CONDITION 関数 [システム]

イベントハンドラがトリガされる条件を指定します。

イベントおよび関連するハンドラを定義するには、**CREATE EVENT** 文を使用します。

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。

パラメータ

表 28 : パラメータ

パラメータ	定義
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する。各条件は、特定のイベントタイプにのみ有効。

表 29 : 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	該当なし	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) の空き領域を表示する。
DBFreeSpace	メガバイト	DBDiskSpace	
DBSize	メガバイト	GrowDB	
ErrorNumber	該当なし	RAISERROR	

条件名	単位	イベント	コメント
IdleTime	秒	ServerIdle	
Interval	秒	ALL	ハンドラが前回実行されてからの時間。
LogFreePercent	該当なし	LogDiskSpace	
LogFreeSpace	メガバイト	LogDiskSpace	
LogSize	メガバイト	GrowLog	
RemainingValues	整数値	GlobalAutoincrement	残った値の数。
TempFreePercent	該当なし	TempDiskSpace	TempDiskSpace は、IQ テンポラリストアではなく、システムテンポラリファイル (TEMP または IQTMP16 の環境変数でポイントされる) の空き領域を表示する。
TempFreeSpace	メガバイト	TempDiskSpace	
TempSize	メガバイト	GrowTemp	

戻り値
INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次のイベント定義には、**EVENT_CONDITION** 関数が使用されています。

```
create event LogNotifier
type LogDiskSpace
where event_condition( 'LogFreePercent' ) < 50
handler
begin
    message 'LogNotifier message'
end
```

参照：

- EVENT_PARAMETER 関数 [システム] (221 ページ)

EVENT_CONDITION_NAME 関数 [システム]

この関数を使用すると、**EVENT_CONDITION** に対して指定可能なパラメータのリストを取得できます。

イベントおよび関連するハンドラを定義するには、**CREATE EVENT** 文を使用します。

構文

```
EVENT_CONDITION_NAME ( integer )
```

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。

パラメータ

パラメータ	説明
integer	0 以上の整数である。

戻り値

VARCHAR

備考

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できません。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

EVENT_PARAMETER 関数 [システム]

イベントハンドラのコンテキスト情報を取得できます。

イベントおよび関連するハンドラを定義するには、**CREATE EVENT** 文を使用します。

構文

```
EVENT_PARAMETER ( context-name )
```

```
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。

パラメータ

表 30 : パラメータ

パラメータ	説明
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する。また、以下の情報を保持する。
ConnectionId	接続 ID。次の文で返される。 <code>connection_property('id')</code>
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役立つ。
TableName	RemainingValues で使用するテーブル名。

さらに、**EVENT_PARAMETER** 関数からは、**EVENT_CONDITION** 関数に対して有効なすべての *condition-name* 引数にアクセスできます。

戻り値

VARCHAR

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- 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] (219 ページ)

EXP 関数 [数値]

指数関数 (e の数値乗) を返します。

構文

```
EXP ( numeric-expression )
```

パラメータ

表 31 : パラメータ

パラメータ	説明
numeric-expression	指数。

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 3269017.3724721107 を返します。

```
SELECT EXP( 15 ) FROM iq_dummy
```

EXP_WEIGHTED_AVG 関数 [集合]

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

```
EXP_WEIGHTED_AVG ( expression, period-expression )
```

```
OVER ( window-spec )
```

パラメータ

パラメータ	説明
expression	計算する加重値の数値式。
period-expression	平均を計算する間隔を指定する数値式。

備考

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。

SAP 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。

$$S * C + (1 - S) * PEMA$$

上の計算では、SAP Sybase IQ 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。

SAP 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。 *period-expression* は、指数加重平均の移動範囲を指定します。

関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*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。

注意： ROLLUP と CUBE は、**GROUP BY** 句ではサポートされていません。
DISTINCT はサポートされていません。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

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 32 : EXP_WEIGHTED_AVG の結果セット

DepartmentID	Surname	Salary	W_AVG
400	Evans	68,940.000	34,470.000000
300	Litton	58,930.000	46,700.000000
200	Sterling	64,900.000	55,800.000000

DepartmentID	Surname	Salary	W_AVG
200	Kelly	87,500.000	71,650.000000
400	Charlton	28,300.000	49,975.000000
100	Lull	87,900.000	68,937.500000
100	Gowda	59,840.000	60,621.875000
400	Francis	53,870.000	61,403.750000

参照：

- WEIGHTED_AVG 関数 [集合] (381 ページ)
- ウィンドウ集合関数の使用法 (89 ページ)

FIRST_VALUE 関数 [集合]

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

パラメータ	説明
<i>expression</i>	順序付けられたセットの最初の値を決定する式。

戻り値

引数のデータ型。

備考

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。**IGNORE NULLS** が指定されていない場合、セット内の最初の値が NULL のときには、NULL が返されます。**IGNORE NULLS** が指定されている場合、**FIRST_VALUE** は、セット内の NULL では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** や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 が **ORDER BY** 式を含まない場合、または **ORDER BY** 式の精度が不十分で一意の順序を確保できない場合、結果は不定になります。window-spec がない場合も、結果が不定になります。

関数構文(インライン)または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

注意： DISTINCT はサポートされていません。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、各従業員の給与と同じ部署で直近に雇用された従業員の給与と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 33 : FIRST_VALUE の結果セット

DepartmentID	EmployeeID	Percentage
500	1,658	100.00000000000000000000000000000000
500	1,570	138.842709713689113761394
500	1,615	110.428462434244870095972
500	1,013	109.585190539292454724330
500	750	137.734409508894510701521
500	921	167.449704854836766654619
500	868	113.239368750752921334778
500	703	222.867927558928643135365
500	191	119.664297474199895594908
400	1,684	100.00000000000000000000000000000000
400	1,740	76.128652163477274215016

DepartmentID	EmployeeID	Percentage
400	1,751	76.353400685155687446813
400	1,607	133.758100765890593292456
400	1,507	77.996465120338650199655
400	1,576	150.428767810774836893669

この例では、従業員 1658 は部署 500 の最初のローで、その部門で直前に雇用されたこと、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。

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

FLOOR 関数 [数値]

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

表 34 : パラメータ

パラメータ	説明
<i>numeric-expression</i>	数値。通常は float 型。

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 123.00 を返します。

```
SELECT FLOOR ( 123 ) FROM iq_dummy
```

SQL 関数

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
```

参照：

- CEILING 関数 [数値] (170 ページ)
- CEIL 関数 [数値] (169 ページ)

GETDATE 関数 [日付と時刻]

現在の日付と時刻を返します。

構文

```
GETDATE ( )
```

戻り値

TIMESTAMP

備考

GETDATE は Transact-SQL 互換のデータ操作関数です。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、システムの日付および時刻を返します。

```
SELECT GETDATE ( ) FROM iq_dummy
```

GRAPHICAL_PLAN 関数 [文字列]

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 ( string-expression  
[, statistics-level  
[, cursor-type  
[, update-status ]]])
```

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。

パラメータ

パラメータ	説明
string-expression	プランを生成する SQL 文。通常、string-expression は SELECT 文だが、 UPDATE 、 DELETE 、 INSERT SELECT 、 SELECT INTO 文を指定することもできる。
statistics-level	整数。Statistics-level は次のいずれか。 <ul style="list-style-type: none"> 0 - オプティマイザの推定のみ (デフォルト) 2 - ノード統計値を含む詳細な統計情報 3 - 詳細な統計情報
cursor-type	カーソルタイプ。文字列として表現される。有効な値は次のとおり。 asensitive、insensitive、sensitive、または keyset-driven。cursor-type が指定されない場合、デフォルトで asensitive が使用される。
update-status	次のいずれかの値を受け入れる文字列パラメータ。これらの値は、指定されたカーソルをオプティマイザがどのように処理するかを示す。 READ-ONLY - このカーソルは読み込み専用。 READ-WRITE (デフォルト) - このカーソルは読み込みや書き込みが可能。

戻り値

LONG VARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **GRAPHICAL_PLAN** を使用する場合は、非構造化データ分析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 **GRAPHICAL_PLAN** を正しいデータ型とサイズに設定する必要があります。

備考

GRAPHICAL_PLAN 関数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。

```
plan not available
```

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。

```
Plan not available. The database option QUERY_PLAN_TEXT_ACCESS is OFF
```

ユーザがプランにアクセスする必要がある場合、SET ANY SYSTEM OPTION システム権限が付与されたユーザが、そのユーザのオプション 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、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)、Interactive SQL クライアントプランウィンドウに表示されます。

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。

SAP Sybase 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せん。カーソルは、DECLARE CURSOR コマンドと OPEN CURSOR コマンドによって、宣言およびオープンされます。直前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 の場合は、CLOSE CURSOR を実行してから、プランを要求する必要があります。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、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、クエリを実行するためのプランを返します。プランは gplan.xml ファイルに保存されます。

注意：フォーマットされたプラン出力を得る際に、OUTPUT 文の HEXADECIMAL 句を ASIS に設定すると、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' HEXADECIMAL 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 ( );
```

参照：

- HTML_PLAN 関数 [文字列] (240 ページ)

GROUPING 関数 [集合]

ROLLUP または **CUBE** 演算の結果セット内のカラムが NULL である場合、その理由が小計ローの一部であるためか、または基本データによるためかを識別します。

構文

```
GROUPING ( group-by-expression )
```

パラメータ

パラメータ	説明
group-by-expression	GROUP BY 句に ROLLUP または CUBE キーワードを使用したクエリの結果セットにおいて、グループ化カラムとして表示される式。この関数を使用すると、ROLLUP 演算または CUBE 演算によって結果セットに追加された小計ローを識別できる。

現在、SAP Sybase IQ は **PERCENTILE_CONT** 関数および **PERCENTILE_DISC** 関数での **GROUP BY CUBE** 演算をサポートしていません。

戻り値

値	説明
1	group-by-expression が、小計ローの一部なので NULL になっていることを示す。カラムは、小計ローの前にあるカラムではない。
0	group-by-expression が小計ローの前にあるカラムであることを示す。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、サポートされていません。

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

GROUP_MEMBER 関数 [システム]

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 ( group-name-string-expression [ , user-name-string-expression ] )
```

パラメータ

パラメータ	説明
<i>group-name-string-expression</i>	対象となるグループを示す。
<i>user-name-string-expression</i>	対象となるユーザを示す。指定しなければ、現在のユーザ名とみなされる。

戻り値

表 35 : 戻り値

値	説明
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 以外の整数を返す。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- 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]

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

パラメータ	説明
hexadecimal-string	big integer (BIGINT) に変換される 16 進数値。次のように入力する。先頭に小文字または大文字の 0x を付けることも、省略することもできる。 0xhex-string 0Xhex-string hex-string

備考

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、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。

『リファレンス：文とオプション』の「データベースオプション」>「アルファベット順のオプションリスト」>「CONVERSION_ERROR オプション [TSQL]」を参照してください。

値に 0x が追加されている場合を除き、BINARY または VARBINARY 値では 8 バイトを超えるとエラーが返され、CHAR または VARCHAR 値では 16 文字を超えるとエラーが返されます。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

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
```

参照：

- BIGINTTOHEX 関数 [データ型変換] (164 ページ)
- HEXTOINT 関数 [データ型変換] (234 ページ)
- INTTOHEX 関数 [データ型変換] (250 ページ)

HEXTOINT 関数 [データ型変換]

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

パラメータ	説明
hexadecimal-string	整数に変換される文字列。次のように入力する。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る。 <i>0xhex-string</i> <i>0Xhex-string</i> <i>hex-string</i>

戻り値

HEXTOINT 関数は、プラットフォームに依存しない SQL INTEGER 相当の 16 進文字列を返します。右から 8 桁目が数値 8 ~ 9 か、大文字または小文字の A ~ F のいずれかであり、その前の桁がすべて大文字または小文字の F の場合は、16 進値が負の整数値になります。次の HEXTOINT は無効な使用例です。これは、引数が符号付き 32 ビット整数で表現できない正の整数値を示しているためです。

```
SELECT HEXTOINT( '0x0080000001' );
```

INT

備考

CONVERSION_ERROR オプションが OFF に設定されていない場合は、無効な 16 進入力に対して SAP 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。

『リファレンス：文とオプション』の「CONVERSION_ERROR オプション [TSQL]」を参照してください。

データベースオプション ASE_FUNCTION_BEHAVIOR は、SAP Sybase IQ の関数 (INTTOHEX や HEXTOINT など) の出力が Adaptive Server の関数の出力と一致するように指定します。

『リファレンス：文とオプション』の「ASE_FUNCTION_BEHAVIOR オプション」を参照してください。

ASE_FUNCTION_BEHAVIOR オプションが ON の場合は次の処理が行われます。

- SAP Sybase IQ の **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。
- SAP Sybase IQ の **HEXTOINT** は、先頭に 0x が付いた最大 16 文字の文字列 (合計で 18 文字) を受け入れます。入力値が大きいと、整数値が 32 ビットの符号付き整数出力サイズをオーバフローする可能性があるため、注意が必要です。
- SAP Sybase IQ の **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。
- SAP Sybase IQ の **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、SAP Sybase IQ の **HEXTOINT** は関連の文字のみを考慮します。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

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
```

参照：

- BIGINTTOHEX 関数 [データ型変換] (164 ページ)
- HEXTOBIGINT 関数 [データ型変換] (232 ページ)
- INTTOHEX 関数 [データ型変換] (250 ページ)

HOUR 関数 [日付と時刻]

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 )
```

表 36 : パラメータ

パラメータ	定義
datetime-expression	日時。

戻り値
SMALLINT

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise ではサポートされていません。

例
次の文は、値 21 を返します。

```
SELECT HOUR ( '1998-07-09 21:12:13' ) FROM iq_dummy
```

HOURS 関数 [日付と時刻]

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

表 37 : パラメータ

パラメータ	説明
datetime-expression	日時。
integer-expression	<i>datetime-expression</i> . に加算する時間数。 <i>integer-expression</i> が負の場合、日時から適切な時間数が引かれる。整数式を指定する場合、 <i>datetime-expression</i> は日時データ型として明示的にキャストする必要がある。

戻り値
INT

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文を実行すると、値 17518758 が返されます。

```
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 が返されます。

```
SELECT HOURS ( CAST ( '1999-05-12 21:05:07'  
                    AS DATETIME ) , 5 ) FROM iq_dummy
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)

HTML_DECODE 関数 [その他]

HTML リテラル文字列で表示される特殊文字エンティティを復号化します。

構文

```
HTML_DECODE ( string )
```

パラメータ

- **string** – HTML ドキュメントで使用される任意のリテラル文字列。

戻り値

LONG VARCHAR または LONG NVARCHAR。

注意： 結果のデータ型は LONG VARCHAR です。SELECT INTO 文で HTML_DECODE を使用する場合は、Unstructured Data Analytics Option ライセンスが必要になります。または CAST を使用して HTML_DECODE を設定して、データ型とサイズを修正する必要があります。

備考

この関数は、適切な置換を行った後に文字列引数を返します。次の表は、有効な文字エンティティのサンプルを示します。

文字	置換
"	"
'	'
&	&
<	<
>	>
&#xhexadecimal-number;	Unicode のコードポイント。16 進数で指定します。たとえば、' は一重引用符を返します。
&#decimal-number;	Unicode のコードポイント。10 進数で指定します。たとえば、™ は商標記号を返します。

指定した Unicode のコードポイントが、データベース側文字セットの文字に変換できる場合は、その文字に変換されます。それ以外の場合は、解釈されないまま返されます。

SAP Sybase IQ は、HTML 4.01 仕様で指定されているすべての文字エンティティ参照をサポートします。<http://www.w3.org/TR/html4/>と <http://www.w3.org/TR/html4/sgml/entities.html#h-24.2> を参照してください。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

次の文は、文字列 `<p>The piano was made for 'Steinway & Sons'.</p>` を返します。

```
SELECT HTML_DECODE('&lt;p&gt;The piano was made ' ||
  'by &lsquo;Steinway &amp; Sons&rsquo;.&lt;/p&gt;')
```

次の文は、文字列 `<p>It cost €8364;85.000,000. </p>` を返します。

```
SELECT HTML_DECODE('&lt;p&gt;It cost &euro;85.000,000.&lt;/p&gt;')
```

HTML_ENCODE 関数 [その他]

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 )
```

パラメータ

- **string** – HTML ドキュメントで使用される任意の文字列。

戻り値

LONG VARCHAR または LONG NVARCHAR。

注意： 結果のデータ型は LONG VARCHAR です。SELECT INTO 文で HTML_ENCODE を使用する場合は、Unstructured Data Analytics Option ライセンスが必要になります。または CAST を使用して HTML_DECODE を設定して、データ型とサイズを修正する必要があります。

備考

この関数は、次の一連の置換を行った後に文字列引数を返します。

文字	置換
"	"
'	'
&	&
<	<
>	>
0x20 より小さいコード <i>nn</i>	&#x <i>nn</i> ;

この関数は NCHAR の入力または出力をサポートしています。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

次の例では、文字列 '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">' を返します。

```
SELECT HTML_ENCODE('<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">')
```

HTML_PLAN 関数 [文字列]

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。

パラメータ

パラメータ	説明
string-expression	プランを生成する SQL 文。通常は SELECT だが、 UPDATE 文と DELETE 文も可能。

HTML_PLAN 関数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すことができます。

SAP Sybase 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せん。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 の場合は、**CLOSE CURSOR** を実行してから、プランを要求する必要があります。

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。

```
No plan. HTML_PLAN function is not supported for this type of statement or database.
```

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、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、クエリを実行するための **HTML** プランを返します。プランは hplan.html ファイルに保存されます。

```
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** に設定すると、値はそのまま書き込まれます。値が制御文字を含む場合も、エスケープはされません。

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 ( );
```

参照：

- GRAPHICAL_PLAN 関数 [文字列] (228 ページ)

HTTP_DECODE 関数 [Web サービス]

HTTP のコード化された文字列を復号化します。これは URL 復号化とも呼ばれます。

構文

```
HTTP_DECODE ( string )
```

SQL 関数

パラメータ

- **string** – URL またはコード化された URL の要求本文から取得された任意の文字列。

戻り値

LONG VARCHAR または LONG NVARCHAR

備考

この関数は、%*nn* という形式 (*nn* は 16 進値) のすべての文字シーケンスをコード *nn* の文字で置換した後に、文字列引数を返します。また、すべてのプラス記号 (+) はスペースで置換されます。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

次の文は、文字列 `http://dcx.sybase.com` を返します。

```
SELECT HTTP_DECODE ('http%3A%2F%2Fdcx.sybase.com')
```

HTTP_ENCODE 関数 [Web サービス]

HTTP で使用するために文字列をコード化します。これは URL コード化とも呼ばれます。

構文

```
HTTP_ENCODE ( string )
```

パラメータ

- **string** – HTTP トランスポート用にコード化される任意の文字列。

戻り値

LONG VARCHAR または LONG NVARCHAR

備考

この関数は、次の一連の置換を行った後に文字列引数を返します。また、16 進コードが 20 より小さいか 7E より大きいすべての文字は、%*nn* (*nn* は文字コード) で置換されます。

文字	置換
スペース	%20

文字	置換
"	%22
#	%23
%	%25
&	%26
,	%2C
;	%3B
<	%3C
>	%3E
[%5B
¥	%5C
]	%5D
`	%60
{	%7B
	%7C
}	%7D
0x20 未満で 0x7f を超える文字コード <i>nn</i>	% <i>nn</i>

この関数は NCHAR の入力または出力をサポートしています。

標準と互換性

- SQL/2008 – ベンダー拡張。

例

次の文は、文字列 /opt%26id=123%26text='oid:c%09d%20ef' を返します。

```
SELECT HTTP_ENCODE('/opt&id=123&text=''oid:c¥x09d ef''')
```

HTTP_HEADER 関数 [Web サービス]

HTTP 要求ヘッダの値を返します。

構文

```
HTTP_HEADER( header-field-name )
```

パラメータ

- **header-field-name** – HTTP 要求ヘッダフィールドの名前。

戻り値

LONG VARCHAR。

注意：結果のデータ型は LONG VARCHAR です。SELECT INTO 文で HTTP_HEADER を使用する場合は、Unstructured Data Analytics Option ライセンスが必要になります。または CAST を使用して HTTP_HEADER を設定して、データ型とサイズを修正する必要があります。

備考

この関数は、指定された HTTP 要求ヘッダフィールドの値を返します。ヘッダが存在しない場合、または HTTP サービスから呼び出されていない場合は NULL を返します。Web サービスを介して HTTP 要求を処理する場合に使用します。

HTTP Web サービス要求を処理するとき、次のようなヘッダが有用です。

- **Cookie** – 要求された URI に関連付けられた cookie 値 (クライアントに格納されている場合)。
- **Referer** – 要求された URI へのリンクが指定されたページの URL (`http://documents.sample.com:80/index.html` など)。
- **Host** – ユーザが指定するリソースを参照して元の URI から取得された、要求されたリソースのインターネットホスト名または IP アドレスとポート番号 (`webserver.sample.com:8082` など)。
- **User-Agent** – クライアントアプリケーション名 (Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0 など)。
- **Accept-Encoding** – クライアントアプリケーションが使用できる応答のエンコーディングリスト (`gzip, deflate` など)。

これらのヘッダの詳細については、<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html> を参照してください。

次の特別なヘッダでは、クライアント要求の要求行で要素にアクセスできます。

- **@HttpMethod** – 処理されている要求の種類を返します。可能な値には DELETE、HEAD、GET、PUT、POST があります。
- **@HttpURI** – HTTP 要求で指定された、要求の完全な URI (`/myservice?&id=-123&version=109&lang=en` など)。
- **@HttpVersion** – 要求の HTTP バージョン (HTTP/1.0 または HTTP/1.1 など)。

- **@HttpQueryString** – 要求された URI が存在する場合に、そのクエリ部分を返します (id=-123&version=109&lang=en など)。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

HTTP Web サービスによって呼び出されるストアプロシージャ内で次の文が使用されると、Cookie ヘッダの値が取得されます。

```
SET cookie_value = HTTP_HEADER( 'Cookie' );
```

HTTP Web サービスによって呼び出されるストアプロシージャ内で次の文が使用されると、データベースサーバメッセージウィンドウの HTTP 要求ヘッダの名前と値が表示されます。

```
BEGIN
  declare header_name long varchar;
  declare header_value long varchar;
  set header_name = NULL;
header_loop:
  LOOP
    SET header_name = NEXT_HTTP_HEADER( header_name );
    IF header_name IS NULL THEN
      LEAVE header_loop
    END IF;
    SET header_value = HTTP_HEADER( header_name );
    MESSAGE 'HEADER: ', header_name, '=',
            header_value TO CONSOLE;
  END LOOP;
END;
```

HTTP_VARIABLE 関数 [Web サービス]

HTTP 変数の値を返します。

構文

```
HTTP_VARIABLE( var-name [ , instance [ , attribute ] ] )
```

パラメータ

- **var-name** – HTTP 変数の名前。
- **instance** – 同じ名前の変数が複数ある場合、フィールドインスタンスのインスタンス番号、または最初の NULL。複数選択を許可している SELECT リストで使用すると便利です。
- **attribute** – マルチパート要求では、マルチパート名のヘッダの値を返すヘッダフィールド名を属性で指定できます。

属性を指定しないと、戻り値が % で復号化され、文字セットがデータベースの文字セットに変換されます。このモードでは、UTF % でコード化されたデータがサポートされています。

属性には、次のいずれかのモードも使用できます。

- '@**BINARY**' – x-www-form-urlencoded バイナリデータ値を返します。このモードは、戻り値が % で復号化されており、変換された文字セットではないことを示します。UTF-8 % エンコードは、このモードではサポートされません。% コード化されたデータは単純に等価のバイト表現にデコードされるからです。
- '@**TRANSPORT**' – 未加工の HTTP トランスポート形式の値を返し、% でのコード化が保持されます。

戻り値

LONG VARCHAR。

注意：結果のデータ型は LONG VARCHAR です。SELECT INTO 文で HTML_VARIABLE を使用する場合は、Unstructured Data Analytics Option ライセンスが必要になります。または CAST を使用して HTML_VARIABLE を設定して、データ型とサイズを修正する必要があります。

備考

この関数は、指定された HTTP 変数の値を返します。Web サービス内で HTTP 要求を処理する場合に使用されます。

var-name が存在しない場合、戻り値は NULL になります。

Web サービス要求が POST で、変数データが multipart/form-data と通知された場合、HTTP サーバは個々の変数の HTTP ヘッダを受信します。*attribute* パラメータを指定すると、特定の変数の POST 要求から関連する multipart/form-data ヘッダ値が HTTP_VARIABLE 関数から返されます。ファイルを表す変数の場合、Content-Disposition、Content-Type、@BINARY の属性はそれぞれファイル名、メディアタイプ、ファイルの内容を返します。

通常は、クライアント (ブラウザなど) の文字セットとデータベース側文字セットの間で、どの入力データも文字セットの変換が行われます。ただし、*attribute* に @BINARY を指定すると、文字セットの変換または % による復号化を行わずに変数値が返されます。@BINARY は、画像などのデータをクライアントから受信するときに便利です。

この関数は、指定されたインスタンスが存在しない場合、または Web サービスの実行外から関数が呼び出された場合には、NULL を返します。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

HTTP Web サービスによって呼び出されるストアプロシージャ内で次の文が使用されると、サンプル URL で示される HTTP 変数の値が取得されます。

```
-- http://sample.com/demo/ShowDetail?product_id=300&customer_id=101
BEGIN
  DECLARE v_customer_id LONG VARCHAR;
  DECLARE v_product_id LONG VARCHAR;
  SET v_customer_id = HTTP_VARIABLE( 'customer_id' );
  SET v_product_id = HTTP_VARIABLE( 'product_id' );
  CALL ShowSalesOrderDetail( v_customer_id, v_product_id );
END;
```

HTTP Web サービスによって呼び出されるストアプロシージャ内で使用された場合、次の文はイメージ変数の Content-Disposition ヘッダと Content-Type ヘッダを要求します。

```
SET v_name = HTTP_VARIABLE( 'image', NULL, 'Content-Disposition' );
SET v_type = HTTP_VARIABLE( 'image', NULL, 'Content-Type' );
```

HTTP Web サービスによって呼び出されるストアプロシージャ内で使用された場合、次の文は現在の文字セットでのイメージ変数値 (文字セットを変換しない値) を要求します。

```
SET v_image = HTTP_VARIABLE( 'image', NULL, '@BINARY' );
```

IFNULL 関数 [その他]

最初の NULL 以外の式、または NULL を返します。

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

表 38 : パラメータ

パラメータ	説明
expression1	評価される式。この値によって、 <i>expression2</i> と <i>expression3</i> のどちらを返すかが決まる。
expression2	<i>expression1</i> が NULL の場合の戻り値。
expression3	<i>expression1</i> が NULL でない場合の戻り値。

戻り値

返されるデータ型は、*expression-2* と *expression-3* のデータ型によって異なります。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

INDEX_COL 関数 [システム]

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

パラメータ	定義
table-name	テーブル名。
index-id	<i>table-name</i> のインデックスのインデックス ID。

パラメータ	定義
key_#	<i>index-id</i> 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する。単一カラムのインデックスの場合、 <i>key_#</i> は 0 になる。複数カラムのインデックスでは、1 つ目のカラムの <i>key_#</i> は 0、2 つ目のカラムは 1、というように指定する。
user-id	<i>table-name</i> の所有者のユーザ ID。 <i>user-id</i> 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 になる。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

参照：

- OBJECT_ID 関数 [システム] (293 ページ)

INSERTSTR 関数 [文字列]

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

パラメータ	定義
<i>numeric-expression</i>	この位置の後に、 <i>string-expression2</i> が挿入される。先頭に文字列を挿入するには、ゼロを使用する。
<i>string-expression1</i>	<i>string-expression2</i> が挿入される文字列。
<i>string-expression2</i>	挿入する文字列。

戻り値

LONG VARCHAR

注意： 結果データ型は LONG VARCHAR です。SELECT INTO 文で INSERTSTR を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、サポートされていません。STUFF 関数が同等の機能を持っており、Adaptive Server および SAP Sybase IQ でサポートされています。

例

次の文は、値 "backoffice" を返します。

```
SELECT INSERTSTR( 0, 'office ', 'back' ) FROM iq_dummy
```

INTTOHEX 関数 [データ型変換]

10 進の整数を 16 進数に変換して返します。

構文

```
INTTOHEX ( integer-expression )
```

パラメータ

パラメータ	説明
integer-expression	16 進数に変換される整数。

戻り値

VARCHAR

備考

INTTOHEX への入力データ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は、SAP 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。

データベースオプション ASE_FUNCTION_BEHAVIOR は、SAP Sybase IQ の関数 (**INTTOHEX** や **HEXTOINT** など) の出力が Adaptive Server の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。

ASE_FUNCTION_BEHAVIOR オプションが無効な場合 (値が OFF の場合) は、次のようになります。

- **INTTOHEX** の出力は SQL Anywhere 互換になります。
- **INTTOHEX** の出力は、入力に応じて 8 桁または 16 桁となり、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。

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。

- **INTTOHEX** の出力は Adaptive Server 互換になります。
- **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。
- SAP Sybase IQ の **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フローが発生し、変換エラーが起こる場合があります。たとえば、次の文を実行すると

```
SELECT INTTOHEX( 1000000000 ) FROM iq_dummy
```

値 3B9ACA00 が返ります。しかし次の文では

```
SELECT INTTOHEX( 10000000000 ) FROM iq_dummy
```

変換エラーが発生します。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

例

次の文を実行すると、値 3B9ACA00 が返ります。

```
SELECT INTTOHEX( 1000000000 ) FROM iq_dummy
```

次の文を実行すると、値 00000002540BE400 が返ります。

```
SELECT INTTOHEX ( 10000000000) FROM iq_dummy
```

参照：

- BIGINTTOHEX 関数 [データ型変換] (164 ページ)
- HEXTOBIGINT 関数 [データ型変換] (232 ページ)
- HEXTOINT 関数 [データ型変換] (234 ページ)

ISDATE 関数 [日付および時刻]

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

表 39 : パラメータ

パラメータ	説明
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。

戻り値

INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase :
 - SQL Anywhere では、ISO 8601 の日付交換形式を使用します。
 - Adaptive Server 15.0.1 以降でサポートされています。

例

次の例は、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
```

ISNULL 関数 [その他]

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。

少なくとも 2 つの式を関数に渡す必要があります。

構文

```
ISNULL ( expression, expression [ ..., expression ] )
```

パラメータ

表 40 : パラメータ

パラメータ	説明
expression	NULL かどうか調べられる式。

戻り値

この関数の戻り値は、指定した式によって異なります。具体的には、データベースサーバが関数を評価するとき、まず、式の比較が可能なデータ型を検索します。該当するデータ型が見つかったら、データベースサーバは式を比較し、比較に使用したデータ型で結果を返します。データベースサーバは、一般に比較が可能なデータ型を見つけることができないと、エラーを返します。

備考

ISNULL 関数は **COALESCE** 関数と同じです。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、値 -66 を返します。

```
SELECT ISNULL( NULL , -66, 55, 45, NULL, 16 ) FROM iq_dummy
```

参照：

- **COALESCE** 関数 [その他] (174 ページ)

ISNUMERIC 関数 [その他]

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 )
```

SQL 関数

パラメータ

パラメータ	説明
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。

戻り値

INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase - Adaptive Server 15.0.1 以降でサポートされています。

例

次の例は、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
```

LAG 関数 [統計]

テーブルまたはパーティション内の前のローの属性値を返す Interrow 関数。

構文

```
LAG (value_expr) [, offset [, default]] OVER ([PARTITION BY window partition] ORDER BY window ordering)
```

パラメータ

パラメータ	説明
value_expr	テーブルから返すオフセットデータを定義するテーブルカラムまたは式。

パラメータ	説明
offset	現在のローより上のロー数。負でない真数値リテラル、または真数値データの SQL 変数で表現する。許容範囲は 0 ~ 231。
default	<i>offset</i> 値がテーブルまたはパーティションのカーディナリティの範囲を超える場合に返される値。
window partition	(オプション) 結果ローセットの分割方法を示す、カンマ区切りの 1 つ以上の値式。
window ordering	ローのソートのための式を定義する。ウィンドウパーティションを指定した場合はウィンドウパーティション内でのソート、指定しなかった場合は結果セット内でのソート。

備考

LAG 関数は、**OVER (ORDER BY)** ウィンドウ指定を必要とします。**OVER (ORDER BY)** 句内のウィンドウパーティション句はオプションです。**OVER (ORDER BY)** 句に、ウィンドウフレーム **ROWS/RANGE** 指定を含めることはできません。

value_expr に分析式を定義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 *value_expr* に使用できます。

offset には、負でない数値データ型を入力する必要があります。**0** を入力すると、現在のローが返されます。負の数を入力すると、エラーが生成されます。

default のデフォルト値は **NULL** です。*default* のデータ型は、*value_expr* 値のデータ型に暗黙的に変換可能である必要があります。変換可能でない場合、SAP Sybase IQ は変換エラーを生成します。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

例

次の例は、Employees テーブルから給与データを返し、部署 ID でデータをパーティションに分割し、入社日でデータを並べ替えます。**LAG** 関数は、前のロー (物理的なオフセットは 1 ロー) から給与を返し、それを **LAG (Salary)** カラムに表示します。

```
SELECT DepartmentID dID, StartDate, Salary, LAG(Salary, 1)
OVER(PARTITION BY dID ORDER BY StartDate) FROM Employees ORDER BY
1,2;
```

次の結果セットが返されます。

dID	StartDate	Salary	Lag(Salary)
100	1984-08-28	45,700.000	NULL
100	1985-01-01	62,000.000	45,700.000
100	1985-06-17	57,490.000	62,000.000
100	1986-06-07	72,995.000	57,490.000
100	1986-07-01	48,023.690	72,995.000
...			
200	1985-02-03	38,500.000	NULL
200	1985-12-06	54,800.000	38,500.000
200	1987-02-19	39,300.000	54,800.000
200	1987-07-10	49,500.000	39,300.000
200	1988-10-04	54,600.000	49,500.000
200	1988-11-12	39,800.000	54,600.000
...			

参照：

- LEAD 関数 [統計] (259 ページ)

LAST_VALUE 関数 [集合]

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

パラメータ	定義
expression	順序付けられたセットの最後の値を決定する式。

戻り値

引数のデータ型。

備考

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 が指定されていない場合は NULL が返されます。IGNORE NULLS を指定した場合、**LAST_VALUE** は、セット内の NULL では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** や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 が **ORDER BY** 式を含まない場合、または **ORDER BY** 式の精度が不十分で一意の順序を確保できない場合、結果は不定になります。window-spec がない場合も、結果が不定になります。

関数構文(インライン)または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

注意： DISTINCT はサポートされていません。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

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 41 : LAST_VALUE の結果セット

employee_name	Salary	DepartmentID	highest_paid
Michael Lynch	24,903.000	500	Jose Martinez
Joseph Barker	27,290.000	500	Jose Martinez
Sheila Romero	27,500.000	500	Jose Martinez
Felicia Kuo	28,200.000	500	Jose Martinez
Jeannette Bertrand	29,800.000	500	Jose Martinez
Jane Braun	34,300.000	500	Jose Martinez
Anthony Rebeiro	34,576.000	500	Jose Martinez
Charles Crowley	41,700.000	500	Jose Martinez
Jose Martinez	55,500.800	500	Jose Martinez

employee_name	Salary	DepartmentID	highest_paid
Doug Charlton	28,300.000	400	Scott Evans
Elizabeth Lambert	29,384.000	400	Scott Evans
Joyce Butterfield	34,011.000	400	Scott Evans
Robert Nielsen	34,889.000	400	Scott Evans
Alex Ahmed	34,992.000	400	Scott Evans
Ruth Wetherby	35,745.000	400	Scott Evans
...

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

LCASE 関数 [文字列]

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

パラメータ	説明
string-expression	小文字に変換される文字列。

戻り値

CHAR

NCHAR

LONG VARCHAR

VARCHAR

NVARCHAR

注意： 結果データ型は LONG VARCHAR です。SELECT INTO 文で LCASE を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – **LCASE** は、Adaptive Server でサポートされていません。**LOWER** を使用して同じ機能を実現できます。

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
```

参照：

- LEFT 関数 [文字列] (261 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)

LEAD 関数 [統計]

テーブルまたはパーティション内の後ろのローの属性値を返す Interrow 関数。

構文

```
LEAD (value_expr) [, offset [, default]] OVER ([PARTITION BY window partition] ORDER BY window ordering)
```

パラメータ

パラメータ	説明
value_expr	テーブルから返すオフセットデータを定義するテーブルカラムまたは式。
offset	現在のローより下のロー数。負でない真数値リテラル、または真数値データの SQL 変数で表現する。許容範囲は 0 ~ 231。
default	offset 値がテーブルまたはパーティションの範囲を超える場合に返される値。
window partition	(オプション) 結果ローセットの分割方法を示す、カンマ区切りの 1 つ以上の値式。

パラメータ	説明
window ordering	ローのソートのための式を定義する。ウィンドウパーティションを指定した場合はウィンドウパーティション内でのソート、指定しなかった場合は結果セット内でのソート。

備考

LEAD 関数は、**OVER (ORDER BY)** ウィンドウ指定を必要とします。**OVER (ORDER BY)** 句内のウィンドウパーティション句はオプションです。**OVER (ORDER BY)** 句に、ウィンドウフレーム **ROWS/RANGE** 指定を含めることはできません。

value_expr に分析式を定義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 *value_expr* に使用できます。

offset には、負でない数値データ型を入力する必要があります。**0** を入力すると、現在のローが返されます。負の数を入力すると、エラーが生成されます。

default のデフォルト値は **NULL** です。*default* のデータ型は、*value_expr* 値のデータ型に暗黙的に変換可能である必要があります。変換可能でない場合、SAP Sybase IQ は変換エラーを生成します。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

例

次の例は、Employees テーブルから給与データを返し、部署 ID でデータをパーティションに分割し、入社日でデータを並べ替えます。**LEAD** 関数は、次のロー (物理的なオフセットは 1 ロー) から給与を返し、それを **LEAD (Salary)** カラムに表示します。

```
SELECT DepartmentID dID, StartDate, Salary, LEAD(Salary, 1)
OVER(PARTITION BY dID ORDER BY StartDate) FROM Employees ORDER BY
1,2;
```

次の結果セットが返されます。

dID	StartDate	Salary	Lead(Salary)
100	1984-08-28	45,700.000	62,000.000
100	1985-01-01	62,000.000	57,490.000
100	1985-06-17	57,490.000	72,995.000
100	1986-06-07	72,995.000	48,023.690
...			
100	1990-08-19	54,900.000	NULL
200	1985-02-03	38,500.000	39,300.000
200	1987-02-19	39,300.000	49,500.000
200	1987-07-10	49,500.000	54,600.000

200	1988-11-28	46,200.000	34,892.000
200	1989-06-01	34,892.000	87,500.000
...			
200	1993-08-12	47,653.000	NULL

参照：

- LAG 関数 [統計] (254 ページ)

LEFT 関数 [文字列]

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

表 42：パラメータ

パラメータ	説明
string-expression	文字列。
numeric-expression	返される文字数。

戻り値

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。SELECT INTO 文で LEFT を使用する場合は、非構造化データ分析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。

備考

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ことがあります。

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、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

SQL 関数

- Sybase — Adaptive Server Enterprise ではサポートされていません。

例

次の文は、値 "choco" を返します。

```
SELECT LEFT( 'chocolate', 5 ) FROM iq_dummy
```

参照：

- LCASE 関数 [文字列] (258 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)

LEN 関数 [文字列]

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

表 43 : パラメータ

パラメータ	説明
<i>string_expr</i>	評価する文字列式。

備考

この関数は、CHAR_LENGTH (*string_expression*) と同じです。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。

例

次の例は、値 3152 を返します。

```
select len(Photo) from Productswhere ID = 500
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- OCTET_LENGTH 関数 [文字列] (295 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)

LENGTH 関数 [文字列]

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

表 44：パラメータ

パラメータ	説明
string-expression	文字列。

戻り値

INT

備考

文字列にマルチバイト文字があり、適切な照合が使用されている場合、**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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。代わりに、**CHAR_LENGTH** 関数を使用してください。

例

次の文は、値 9 を返します。

```
SELECT LENGTH( 'chocolate' ) FROM iq_dummy
```

参照：

- **BIT_LENGTH** 関数 [文字列] (165 ページ)
- **BYTE_LENGTH** 関数 [文字列] (166 ページ)
- **CHAR_LENGTH** 関数 [文字列] (171 ページ)
- **COL_LENGTH** 関数 [システム] (175 ページ)
- **DATALength** 関数 [システム] (189 ページ)
- **LEN** 関数 [文字列] (262 ページ)
- **OBJECT_NAME** 関数 [システム] (294 ページ)
- **OCTET_LENGTH** 関数 [文字列] (295 ページ)
- **STR_REPLACE** 関数 [文字列] (357 ページ)

LIST 関数 [集合]

グループ内のローごとに値のデリミタ付きリストを返します。

構文

```
LIST(  
[ALL | DISTINCT] string-expression  
[, 'delimiter-string']  
[ORDER BY order-by-expression [ ASC | DESC ], ... ] )
```

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。

パラメータ

- **string-expression** – 文字列式。通常はカラム名です。ALL が指定されていると (デフォルト)、グループ内のローごとに *string-expression* の値が結果文字列に追加されます。各値は、*delimiter-string* で区切られます。DISTINCT が指定されていると、ユニークな *string-expression* 値のみが追加されます。

- **delimiter-string** – リスト項目のデリミタ文字列。デフォルト設定はカンマです。NULL 値または空の文字列を指定した場合は、デリミタはありません。delimiter-string は定数でなければなりません。
- **order-by-expression** – 関数によって返された項目を並べ替えます。この引数の前にカンマは必要ありません。このため、delimiter-string を指定しない場合、使用が簡単になります。

order-by-expression には整数リテラルを指定できません。ただし、整数リテラルを含む変数を指定できます。

ORDER BY 句に定数が含まれている場合、それらの定数はオプティマイザによって解釈され、同義の ORDER BY 句に置き換えられます。たとえば、オプティマイザは ORDER BY 'a' を ORDER BY 式として解釈します。

クエリブロックに、有効な ORDER BY 句が指定された複数の集合関数が含まれているとき、それらの ORDER BY 句を単一の ORDER BY 句に論理的に結合できる場合は、そのクエリブロックを実行できます。たとえば、次の ORDER BY 句の場合は、

```
ORDER BY expression1, 'a', expression2
```

```
ORDER BY expression1, 'b', expression2, 'c', expression3
```

次の ORDER BY 句として結合されます。

```
ORDER BY expression1, expression2, expression3
```

戻り値

LONG VARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **LIST** を使用する場合は、非構造化データ分析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 **LIST** を正しいデータ型とサイズに設定する必要があります。

備考

LIST 関数は、グループの各ローの、NULL 以外のすべての X の値を (デリミタ付きで) 連結して返します。グループ内に明確な X 値を持ったローが 1 つ以上存在しない場合、LIST(X) は空の文字列を返します。

LIST 関数では、NULL 値と空の文字列は無視されます。

LIST 関数は Window 関数として使用できませんが、Window 関数の入力には使用できます。

この関数は NCHAR の入力または出力をサポートしています。

標準と互換性

- **SQL/2008** – ベンダ拡張。

SAP Sybase IQ では、SQL/2008 言語機能 F441、「Extended set function support」がサポートされています。これにより、カラム参照ではない任意の式を集合関数のオペランドで使用できます。

SAP Sybase IQ では、オプションの SQL/2008 機能 F442、「Mixed column references in set function」がサポートされていません。SAP Sybase IQ では、LIST 関数を含むクエリブロックからのカラム参照と外部参照の両方を、集合関数の引数に含めることはできません。

例

次の文は、値 487 Kennedy Court, 547 School Street を返します。

```
SELECT LIST( Street ) FROM Employees
WHERE GivenName = 'Thomas';
```

次の文は、従業員 ID をリストします。結果セットの各ローには、部門ごとの従業員 ID のカンマで区切られたリストが入っています。

```
SELECT LIST( EmployeeID )
FROM Employees
GROUP BY DepartmentID;
```

LIST(EmployeeID)
102,105,160,243,247,249,266,278,...
129,195,299,467,641,667,690,856,...
148,390,586,757,879,1293,1336,...
184,207,318,409,591,888,992,1062,...
191,703,750,868,921,1013,1570,...

次の文は、従業員の姓を基準に従業員 ID をソートします。

```
SELECT LIST( EmployeeID ORDER BY Surname ) AS "Sorted IDs"
FROM Employees
GROUP BY DepartmentID;
```

ソートされた ID
1013,191,750,921,868,1658,...
1751,591,1062,1191,992,888,318,...

ソートされた ID
1336,879,586,390,757,148,1483,...
1039,129,1142,195,667,1162,902,...
160,105,1250,247,266,249,445,...

次の文は、セミコロンで区切られたリストを返します。ORDER BY 句とリストセパレータの位置に注意してください。

```
SELECT LIST( EmployeeID, ';' ORDER BY Surname ) AS "Sorted IDs"
FROM Employees
GROUP BY DepartmentID;
```

ソートされた ID
1013;191;750;921;868;1658;703;...
1751;591;1062;1191;992;888;318;...
1336;879;586;390;757;148;1483;...
1039;129;1142;195;667;1162;902; ...
160;105;1250;247;266;249;445;...

前の文と次の文は区別してください。次の文では、(Surname, ';')を複合ソートキーを使用してソートした従業員 ID のカンマ区切りのリストが返ります。

```
SELECT LIST( EmployeeID ORDER BY Surname, ';' ) AS "Sorted IDs"
FROM Employees
GROUP BY DepartmentID;
```

LN 関数 [数値]

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

パラメータ	説明
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。他のデータ型を指定すると、LN 関数ではエラーが返る。戻り値は、DOUBLE データ型。

備考

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。

SQL 関数

LN 関数は LOG 関数のエイリアスです。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。代わりに、LOG 関数を使用してください。

参照：

- LOG 関数 [数値] (270 ページ)
- LOG10 関数 [数値] (270 ページ)

LOCATE 関数 [文字列]

ある文字列の、別の文字列内の位置を返します。

構文

```
LOCATE ( string-expression1, string-expression2  
[ , numeric-expression ] )
```

パラメータ

パラメータ	説明
string-expression1	検索される文字列。
string-expression2	検索する文字列。255 バイトまでの文字列を指定する。
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。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す。除外されるバイト数は、 $(-1 * \text{offset}) - 1$ で計算される。

numeric-expression 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。

戻り値

INT

備考

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 255 バイトを超える長い文字列を指定できますが、2 番目の文字列は 255 バイトに制限されます。2 番目の文字列が 255 バイトを超えるとエラーが発生します。

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。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。

『非構造化データ分析』の「**LOCATE** 関数」を参照してください。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、サポートされていません。

例

次の文は、値 8 を返します。

```
SELECT LOCATE( 'office party this week - rsvp as soon as possible',
'party', 2 ) FROM iq_dummy
```

2 つ目の例は、検索の開始オフセットである *numeric-expression* に負の数が指定されています。

```
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;

SELECT LOCATE(dirname, '¥', -1), dirname FROM t1;
```

結果は次のようになります。

```
18 c:¥test¥functions¥locate.sql
18 d:¥test¥functions¥trim.sql
```

参照：

- PATINDEX 関数 [文字列] (296 ページ)

SQL 関数

- LIKE 条件 (55 ページ)

LOG 関数 [数値]

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

表 45 : パラメータ

パラメータ	説明
numeric-expression	数値。

戻り値

この関数は、引数を DOUBLE に変換し、計算を倍精度浮動小数点で行い、結果を DOUBLE で返します。パラメータが NULL 値の場合、結果は NULL 値になります。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、値 3.912023 を返します。

```
SELECT LOG( 50 ) FROM iq_dummy
```

参照：

- LN 関数 [数値] (267 ページ)
- LOG10 関数 [数値] (270 ページ)

LOG10 関数 [数値]

数値の、底を 10 とする対数を返します。

構文

```
LOG10 ( numeric-expression )
```

パラメータ

表 46 : パラメータ

パラメータ	説明
numeric-expression	数値。

戻り値

この関数は、引数を DOUBLE に変換し、倍精度浮動小数点で計算を行います。パラメータが NULL 値の場合、結果は NULL 値になります。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 1.698970 を返します。

```
SELECT LOG10( 50 ) FROM iq_dummy
```

参照：

- LN 関数 [数値] (267 ページ)
- LOG 関数 [数値] (270 ページ)

LOWER 関数 [文字列]

文字列内のすべての文字を小文字に変換します。

構文

```
LOWER ( string-expression )
```

パラメータ

表 47 : パラメータ

パラメータ	説明
string-expression	変換される文字列。

戻り値

CHAR

NCHAR

SQL 関数

LONG VARCHAR

VARCHAR

NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **LOWER** を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文を実行すると、値 "lower case" が返ります。

```
SELECT LOWER ( 'LOWER CasE' ) FROM iq_dummy
```

参照：

- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)

LTRIM 関数 [文字列]

文字列から先行ブランクを削除します。

構文

```
LTRIM ( string-expression )
```

パラメータ

パラメータ	説明
string-expression	削除される文字列。

戻り値

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **LTRIM** を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、先行空白がすべて削除された値 "Test Message" を返します。

```
SELECT LTRIM( '      Test Message' ) FROM iq_dummy
```

参照：

- RTRIM 関数 [文字列] (335 ページ)

MAX 関数 [集合]

ローの各グループ内での *expression* の最大値を返します。

構文

```
MAX ( expression
| DISTINCT column-name )
```

パラメータ

パラメータ	説明
<i>expression</i>	最大値が計算される式。通常はカラムを指定する。
DISTINCT <i>column-name</i>	MAX (<i>expression</i>) を使用した場合と戻り値は同じ。万全を期すために含まれている。

戻り値

引数と同じデータ型。

備考

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

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、Employees テーブル内の給与の最大値 138948.000 を返します。

```
SELECT MAX ( Salary )  
FROM Employees
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)
- MIN 関数 [集合] (276 ページ)

MEDIAN 関数 [集合]

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)
```

パラメータ

パラメータ	説明
expression	計算する中央値の数値式。

備考

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** 句ではサポートされていません。

構文 2 は、**SELECT** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

注意： *window-spec* には、**ROW**、**RANGE**、**ORDER BY** 指定を含めることはできません。 *window-spec* では **PARTITION** 句のみ指定できます。**WINDOW** 句を使用する場合、**DISTINCT** はサポートされません。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

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 48 : MEDIAN の結果セット

DepartmentID	Surname	Salary	Median
100	Lull	87,900.000	73,870.000
100	Gowda	59,840.000	73,870.000
200	Sterling	64,900.000	76,200.000
200	Kelly	87,500.000	76,200.000
300	Litton	58,930.000	58,930.000
400	Francis	53,870.000	38,70.000
400	Charlton	28,300.000	53,870.000
400	Evans	68,940.000	53,870.000

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

MIN 関数 [集合]

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

パラメータ	説明
<code>expression</code>	最小値が計算される式。通常はカラムを指定する。
<code>DISTINCT column-name</code>	<code>MIN (expression)</code> を使用した場合と戻り値は同じ。万全を期すために含まれている。

戻り値

引数と同じデータ型。

備考

`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

- SQL — ISO/ANSI SQL 準拠。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、`Employees` テーブル内の給与の最小値 24903.000 を返します。

```
SELECT MIN ( Salary )
FROM Employees
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)
- MAX 関数 [集合] (273 ページ)

MINUTE 関数 [日付と時刻]

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 ( datetime-expression )
```

パラメータ

パラメータ	説明
datetime-expression	日時の値。

戻り値

SMALLINT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 22 を返します。

```
SELECT MINUTE ( '1998-07-13 12:22:34' ) FROM iq_dummy
```

MINUTES 関数 [日付と時刻]

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 integer-expression で指定された分の数を時刻に追加します。

構文

```
MINUTES ( datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression )
```

パラメータ

パラメータ	説明
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、日時から適切な分数が引かれる。整数式を指定する場合は、 <i>datetime-expression</i> を datetime データ型として明示的にキャストする必要がある。

戻り値

INT

TIMESTAMP

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

値 1051125487 が返されます。

```
SELECT MINUTES( '1998-07-13 06:07:12' ) FROM iq_dummy
```

2つの時刻の差である値 240 が返されます。

```
SELECT MINUTES( '1999-07-13 06:07:12',  
                '1999-07-13 10:07:12' ) FROM iq_dummy
```

日時の値 1999-05-12 21:10:07.000 が返されます。

```
SELECT MINUTES( CAST( '1999-05-12 21:05:07'  
                    AS DATETIME ), 5) FROM iq_dummy
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)

MOD 関数 [数値]

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

パラメータ	説明
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。

戻り値

SMALLINT

INT

NUMERIC

備考

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* の符号は計算結果に影響を与えません。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、サポートされていません。Adaptive Server では、% 演算子をモジュロ演算子として使用します。

例

次の文は、値 2 を返します。

```
SELECT MOD( 5, 3 ) FROM iq_dummy
```

参照：

- REMAINDER 関数 [数値] (323 ページ)

MONTH 関数 [日付と時刻]

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

パラメータ	説明
date-expression	日時の値。

戻り値
SMALLINT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例
次の文は、値 7 を返します。

```
SELECT MONTH ( '1998-07-13' ) FROM iq_dummy
```

MONTHNAME 関数 [日付と時刻]

指定された日付式から月の名前を返します。

構文

```
MONTHNAME ( date-expression )
```

パラメータ

パラメータ	説明
date-expression	日時の値。

戻り値
VARCHAR

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例
DATE_ORDER オプションがデフォルト値の *ymd* に設定されている場合、次の文は、値 **September** を返します。

```
SELECT MONTHNAME ( '1998-09-05' ) FROM iq_dummy
```

MONTHS 関数 [日付と時刻]

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 `integer-expression` で指定された月数を日時に追加します。

構文

```
MONTHS ( date-expression
| date-expression, datetime-expression
| date-expression, integer-expression )
```

パラメータ

パラメータ	説明
<code>date-expression</code>	日時。
<code>integer-expression</code>	<code>date-expression</code> に追加する月数。 <code>integer-expression</code> が負の場合、日時から適切な月数が引かれる。整数式を指定する場合は、 <code>date-expression</code> を <code>datetime</code> データ型として明示的にキャストする必要がある。

戻り値

INT

TIMESTAMP

備考

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 )
```

MONTH 関数の比較は、間違っていて、請求書が送られてから 12 か月後の支払も含む可能性があります。

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、日時から適切な月数が引かれます。時間、分、秒は無視されます。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。

SQL 関数

- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文を実行すると、値 23982 が返されます。

```
SELECT MONTHS ( '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 MONTHS ( '1999-07-13 06:07:12',  
                '1999-09-13 10:07:12' ) FROM iq_dummy
```

次の文を実行すると、日時の値 1999-10-12 21:05:07.000 が返されます。

```
SELECT MONTHS ( CAST ( '1999-05-12 21:05:07'  
AS DATETIME ), 5) FROM iq_dummy
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)

NEWID 関数 [その他]

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。

戻り値

UNIQUEIDENTIFIER

備考

NEWID() 関数は一意的な識別子の値を生成します。

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。

NEWID 関数は、次の位置でのみサポートされます。

- 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

NEWID 関数によって生成された値は、テーブルのカラムのデフォルト値として使用できます。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server では、サポートされていません。

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()
```

一意の識別子が `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。

参照：

- バイナリデータ型 (398 ページ)
- **STRTOUUID** 関数 [文字列] (360 ページ)
- **UUIDTOSTR** 関数 [文字列] (374 ページ)
- 文字データ型 (389 ページ)
- バイナリデータ型 (874 ページ)

NEXT_CONNECTION 関数 [システム]

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 ( { connection-id }, { 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。

戻り値

INT

パラメータ

パラメータ	説明
connection-id	整数。通常は、前回の NEXT_CONNECTION の呼び出しで返された整数。 <i>connection-id</i> が NULL の場合、 NEXT_CONNECTION は最新の接続 ID を返す。
database-id	現在のサーバ上のデータベースの 1 つを表す整数。 <i>database-id</i> を指定しない場合は、現在のデータベースが使用される。NULL を指定すると、 NEXT_CONNECTION はデータベースに関係なく次の接続を返す。

備考

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。

NEXT_CONNECTION を使用して、データベースへの接続を列挙できます。接続 ID は、通常、単純増加する昇順で作成されます。この関数は、反対の降順で次の接続 ID を返します。

最新の接続の接続 ID 値を取得するには、*connection-id* に NULL を入力します。その次の接続を取得するには、前の戻り値を入力します。その順序で次の接続がなくなると、NULL が返されます。

NEXT_CONNECTION は、特定の時点より前に作成されたすべての接続を切断する場合に便利です。ただし、**NEXT_CONNECTION** は接続 ID を降順で返すため、関数の開始後に作成された接続は返されません。すべての接続を確実に切断するには、新しい接続が作成されないようにしてから **NEXT_CONNECTION** を実行します。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

例

次の文は、現在のデータベースの最初の接続の ID を返します。ID は 10 などの整数値です。

```
SELECT NEXT_CONNECTION( NULL );
```

次の文は、5 などの値を返します。

```
SELECT NEXT_CONNECTION( 10 );
```

次の呼び出しは、現在のデータベースについて、指定された *connection-id* から降順で次の接続 ID を返します。

```
SELECT NEXT_CONNECTION( connection-id );
```

次の呼び出しは、データベースに関係なく、指定された *connection-id* から降順で次の接続 ID を返します。

```
SELECT NEXT_CONNECTION( connection-id, NULL );
```

次の呼び出しは、指定されたデータベースについて、指定された *connection-id* から降順で次の接続 ID を返します。

```
SELECT NEXT_CONNECTION( connection-id, database-id );
```

次の呼び出しは、データベースに関係なく、最初の (最も古い) 接続を返します。

```
SELECT NEXT_CONNECTION( NULL, NULL );
```

次の呼び出しは、指定されたデータベースについて、最初の (最も古い) 接続を返します。

```
SELECT NEXT_CONNECTION( NULL, database-id );
```

NEXT_DATABASE 関数 [システム]

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す。

パラメータ

パラメータ	説明
database-id	データベースの ID 番号を指定する整数。

戻り値
INT

備考

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

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

参照：

- COL_NAME 関数 [システム] (176 ページ)
- DB_ID 関数 [システム] (211 ページ)
- DB_NAME 関数 [システム] (212 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)

NEXT_HTTP_HEADER 関数 [Web サービス]

次の HTTP ヘッダの名前を返します。

構文

```
NEXT_HTTP_HEADER( header-name )
```

パラメータ

- **header-name** – 前の要求ヘッダの名前。header-name が NULL の場合、この関数は最初の HTTP 要求ヘッダの名前を返します。

戻り値

LONG VARCHAR。

注意： 結果のデータ型は LONG VARCHAR です。SELECT INTO 文で NEXT_HTTP_HEADER を使用する場合は、Unstructured Data Analytics Option ライセンスが必要になります。または CAST を使用して HTML_DECODE を設定して、データ型とサイズを修正する必要があります。

備考

この関数を HTTP 要求ヘッダで繰り返し使用することで、次の HTTP ヘッダ名が返されます。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。後続のヘッダは、前のヘッダの名前を関数に渡すことによって取得されます。この関数を最後のヘッダ名を使用して呼び出した場合、または Web サービスから呼び出していない場合、NULL を返します。

この関数を繰り返し呼び出すと、すべてのヘッダフィールドが一度だけ返されます。ただし、必ずしも HTTP 要求での表示順に表示されるとはかぎりません。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

HTTP Web サービスによって呼び出されるストアードプロシージャ内で次の文が使用されると、データベースサーバメッセージウィンドウの HTTP 要求ヘッダの名前と値が表示されます。

```
BEGIN
  declare header_name long varchar;
  declare header_value long varchar;
  set header_name = NULL;
header_loop:
  LOOP
    SET header_name = NEXT_HTTP_HEADER( header_name );
    IF header_name IS NULL THEN
      LEAVE header_loop
    END IF;
    SET header_value = HTTP_HEADER( header_name );
    MESSAGE 'HEADER: ', header_name, '=',
            header_value TO CONSOLE;
  END LOOP;
END;
```

NEXT_HTTP_VARIABLE 関数 [Web サービス]

次の HTTP 変数の名前を返します。

構文

```
NEXT_HTTP_VARIABLE ( var-name )
```

パラメータ

- **var-name** – 前の変数の名前。var-name が NULL の場合、この関数は最初の HTTP 変数の名前を返します。

戻り値

LONG VARCHAR。

注意： 結果のデータ型は LONG VARCHAR です。SELECT INTO 文で NEXT_HTTP_VARIABLE を使用する場合は、Unstructured Data Analytics Option ライセンスが必要になります。または CAST を使用して NEXT_HTTP_VARIABLE を設定して、データ型とサイズを修正する必要があります。

備考

この関数は、要求内の HTTP 変数に対して反復して適用されます。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。後続の変数は、前の変数の名前を関数に渡すことによって取得されます。この関数は、最後の変数名を使用して呼び出した場合、または Web サービスから呼び出していない場合、NULL を返します。

この関数を繰り返し呼び出すと、すべての変数が一度だけ返されます。ただし、必ずしも HTTP 要求での表示順に表示されるとはかぎりません。URL PATH が ON または ELEMENTS に設定される場合、変数 url または url1、url2、...、url10 が個々に含められます。

標準と互換性

- **SQL/2008** – ベンダー拡張。

例

HTTP Web サービスによって呼び出されるストアプロシージャ内で次の文が使用されると、最初の HTTP 変数の名前が返されます。

```
BEGIN
DECLARE variable_name LONG VARCHAR;
DECLARE variable_value LONG VARCHAR;
SET variable_name = NULL;
SET variable_name = NEXT_HTTP_VARIABLE( variable_name );
```

```
SET variable_value = HTTP_VARIABLE( variable_name );
END;
```

NOW 関数 [日付と時刻]

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 ( * )
```

戻り値

TIMESTAMP

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、現在の日付および時刻を返します。

```
SELECT NOW(*) FROM iq_dummy
```

NTILE 関数 [統計]

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 ( expression1 )
OVER ( ORDER BY expression2 [ ASC | DESC ] )
```

パラメータ

パラメータ	説明
expression1	バケットの数を指定する、1～32767の定数。
expression2	ソートを指定する。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る。

備考

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できます。

NTILE には **OVER (ORDER BY)** 句が必須です。**ORDER BY** 句は、ランク付けを実行するパラメータと、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** 句の内部のみで使用するもので、**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で **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に対して処理を行うことを示します。結果セットは、**FROM**、**WHERE**、**GROUP BY**、**HAVING** の各句がすべて評価された後で返されるローです。**OVER** 句は、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 は、**SELECT** または **INSERT** 文の **select** リスト、および **SELECT** 文の **ORDER BY** 句でのみ使用できます。**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、**UPDATE** 文や **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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

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、販売台数で上位 50% のカーディーラを調べるには、**NTILE(2)** を指定します。

参照：

- PERCENTILE_CONT 関数 [統計] (299 ページ)
- PERCENTILE_DISC 関数 [統計] (302 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)

NULLIF 関数 [その他]

式を比較します。**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

パラメータ	説明
expression1	比較される式。
expression2	比較される式。

戻り値

最初の引数のデータ型。

備考

NULLIF は 2 つの式の値を比較します。

1 番目の式と 2 番目の式が等しい場合、**NULLIF** は **NULL** を返します。

1 番目の式と 2 番目の式が異なる場合、または 2 番目の式が **NULL** の場合、**NULLIF** は 1 番目の式を返します。

NULLIF 関数を使用すると、**CASE** 式を手短かに書くことができます。**NULLIF** は次の式と同等です。

```
CASE WHEN expression1 = expression2 THEN NULL  
ELSE expression1 END
```

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。

- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

参照：

- CASE 式 (38 ページ)
- 省略形 CASE 式の NULLIF 関数 (39 ページ)

NUMBER 関数 [その他]

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 ( * )
```

戻り値

INT

備考

NUMBER 関数は、**UPDATE** 文の **SELECT** リストまたは **SET** 句でのみ使用してください。たとえば、次の文は seq_id カラムの各ローを直前のローより 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(*)** 関数が **SET** 句に使用され、**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。

NUMBER によるプライマリキーの生成は、**SELECT** 文から **INSERT** を使用して行うことも可能です。しかし、連続したプライマリキーの生成には、**IDENTITY/ AUTOINCREMENT** を使用することをおすすめします。

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** 句、**DISTINCT** 句、**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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

OBJECT_ID 関数 [システム]

オブジェクト ID を返します。

構文

```
OBJECT_ID ( object-name )
```

パラメータ

パラメータ	説明
object-name	オブジェクト名。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

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
```

参照：

- COL_NAME 関数 [システム] (176 ページ)
- DB_ID 関数 [システム] (211 ページ)
- DB_NAME 関数 [システム] (212 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- INDEX_COL 関数 [システム] (248 ページ)

OBJECT_NAME 関数 [システム]

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

パラメータ	説明
object id	オブジェクト ID。
database-id	データベース ID。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

次の文は、名前 "customer" を返します。

```
SELECT OBJECT_NAME ( 100209 ) FROM iq_dummy
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)

- OCTET_LENGTH 関数 [文字列] (295 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)
- COL_NAME 関数 [システム] (176 ページ)
- DB_ID 関数 [システム] (211 ページ)
- DB_NAME 関数 [システム] (212 ページ)
- DB_PROPERTY 関数 [システム] (213 ページ)
- NEXT_DATABASE 関数 [システム] (285 ページ)
- OBJECT_ID 関数 [システム] (293 ページ)

OCTET_LENGTH 関数 [文字列]

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 ( column-name )
```

パラメータ

パラメータ	説明
column-name	カラムの名前。

備考

引数が NULL の場合は、NULL 値を返します。

OCTET_LENGTH 関数は、すべての SAP Sybase IQ データ型をサポートしています。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラジオブジェクトデータを使用できます。

『非構造化データ分析』の「OCTET_LENGTH 関数」を参照してください。

標準と互換性

- Sybase — SQL Anywhere または Adaptive Server でサポートされていません。

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LEN 関数 [文字列] (262 ページ)

SQL 関数

- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- STR_REPLACE 関数 [文字列] (357 ページ)

PATINDEX 関数 [文字列]

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 ( '% pattern%', string-expression )
```

パラメータ

パラメータ	説明
pattern	検索するパターン。パターンは、ワイルドカードを使用して 126 バイトまでの文字列を指定する。最初の % ワイルドカードを省略すると、 PATINDEX は、パターンが文字列の最初に出現する場合に 1 を、そうでない場合は 0 を返す。 <i>pattern</i> の先頭の文字が % の場合は、% を 2 つ続けると 1 つとして処理される。 パターンにワイルドカード (パーセント % またはアンダースコア _) を使用しない場合は、255 バイトの長さまで指定できる。
string-expression	パターンが検索される文字列。

戻り値

INT

備考

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、**PATINDEX** は最初の文字列の位置だけを返します。

パターンに使用するワイルドカードは、**LIKE** での比較の場合と同じです。次の表は、パターンのワイルドカードを示します。

表 49 : **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

ワイルドカード	一致条件
[]	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

パターンを検出できなかった場合、**PATINDEX** はゼロ (0) を返します。

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) となります。

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。

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。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。

『非構造化データ分析』の「**PATINDEX** 関数」を参照してください。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server 互換。

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
```

参照：

- LIKE 条件 (55 ページ)
- LOCATE 関数 [文字列] (268 ページ)

PERCENT_RANK 関数 [統計]

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

パラメータ	説明
expression	ソートを指定する。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る。

戻り値

PERCENT_RANK 関数は、0 ~ 1 の DOUBLE 値を返します。

備考

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。

```
(Rx - 1) / (NtotalRow - 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)** 句が必須です。**ORDER BY** 句は、ランク付けを実行するパラメータと、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** 句の内部のみで使用するもので、**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で **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に対して処理を行うことを示します。結果セットは、**FROM**、**WHERE**、**GROUP BY**、**HAVING** の各句がすべて評価された後で返されるローです。**OVER** 句は、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 は、**SELECT** または **INSERT** 文の select リスト、および **SELECT** 文の **ORDER BY** 句でのみ使用できます。**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、**UPDATE** 文や **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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の文は、**PERCENT_RANK** 関数の使い方を示しています。

```
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

PERCENTILE_CONT 関数 [統計]

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

パラメータ	説明
expression1	数値データ型の定数を、0 以上 1 以下で指定する。引数が NULL であれば、"wrong argument for percentile" エラーが返される。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" エラーが返される。
expression2	ソートを指定する。カラム参照を含む単一の式で指定する。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ない。

備考

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、データセットのスレッシュホールド許容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** 句には、パーセンタイル関数の実行対象となる式と、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** 句の内部のみで使用するもので、**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 で使用できません。**PERCENTILE_CONT** は、統計を行わない単純な集合関数が使用されるのであれば、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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の例では、**PERCENTILE_CONT** 関数を使用して、各地域の自動車販売の 10 番目のパーセンタイル値を求めます。

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、各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

参照：

- NTILE 関数 [統計] (289 ページ)
- PERCENTILE_DISC 関数 [統計] (302 ページ)

PERCENTILE_DISC 関数 [統計]

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

パラメータ	説明
<i>expression1</i>	数値データ型の定数を、0 以上 1 以下で指定する。引数が NULL であれば、"wrong argument for percentile" エラーが返される。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" エラーが返される。
<i>expression2</i>	ソートを指定する。カラム参照を含む単一の式で指定する。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ない。

備考

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、データセットのスレッシュホールド許容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、SAP 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** 句には、パーセンタイル関数の実行対象となる式と、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** 句の内部のみで使用するもので、**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 で使用できません。**PERCENTILE_DISC** は、分析を行わない単純な集合関数が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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

次の例では、**PERCENTILE_DISC** 関数を使用して、各地域の自動車販売の10番目のパーセンタイル値を求めます。

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、各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

SQL 関数

参照：

- NTILE 関数 [統計] (289 ページ)
- PERCENTILE_CONT 関数 [統計] (299 ページ)

PI 関数 [数値]

円周率の値を数値で返します。

構文

```
PI ( * )
```

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server では、**PI()** 関数はサポートされていますが、**PI(*)** はサポートされていません。

例

次の文は、値 3.141592653... を返します。

```
SELECT PI ( * ) FROM iq_dummy
```

POWER 関数 [数値]

数値のべき乗を求めます。

構文

```
POWER ( numeric-expression1, numeric-expression2 )
```

パラメータ

パラメータ	説明
numeric-expression1	底。
numeric-expression2	指数。

戻り値

DOUBLE

備考

numeric-expression1 を *numeric-expression2* で累乗します。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 64 を返します。

```
SELECT Power( 2, 6 ) FROM iq_dummy
```

PROPERTY 関数 [システム]

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 ( { property-id | 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す。

パラメータ

表 50 :

パラメータ	説明
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。この番号は、 PROPERTY_NUMBER 関数で調べることができる。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> がよく使用される。
property-name	プロパティ名を指定する文字列。

戻り値

VARCHAR

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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。

SQL 関数

```
SELECT PROPERTY( 'Name' ) FROM iq_dummy
```

参照：

- PROPERTY_NAME 関数 [システム] (307 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)
- サーバで使用可能なプロパティ (105 ページ)
- 各データベースで使用可能なプロパティ (134 ページ)
- 接続プロパティ (104 ページ)

PROPERTY_DESCRIPTION 関数 [システム]

プロパティの説明を返します。

構文

```
PROPERTY_DESCRIPTION ( { property-id | 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す。

パラメータ

パラメータ	説明
property-id	プロパティの番号である整数。この番号は、 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。

戻り値

VARCHAR

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

PROPERTY_NAME 関数 [システム]

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。

パラメータ

パラメータ	説明
property-id	プロパティのプロパティ番号。

戻り値

VARCHAR

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

参照：

- PROPERTY 関数 [システム] (305 ページ)
- PROPERTY_NUMBER 関数 [システム] (307 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)
- サーバで使用可能なプロパティ (105 ページ)
- 各データベースで使用可能なプロパティ (134 ページ)
- 接続プロパティ (104 ページ)

PROPERTY_NUMBER 関数 [システム]

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す。

パラメータ

パラメータ	説明
property-name	プロパティ名。

戻り値

INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

参照：

- PROPERTY 関数 [システム] (305 ページ)
- PROPERTY_NAME 関数 [システム] (307 ページ)
- CONNECTION_PROPERTY 関数 [システム] (177 ページ)
- サーバで使用可能なプロパティ (105 ページ)
- 各データベースで使用可能なプロパティ (134 ページ)
- 接続プロパティ (104 ページ)

QUARTER 関数 [日付と時刻]

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

パラメータ	説明
date-expression	日付。

戻り値

INT

備考

次の表は、1年の各四半期に含まれる日付を示します。

表 51 : 1年の四半期の値

四半期	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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

DATE_ORDER オプションがデフォルト値の *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
```

RADIANS 関数 [数値]

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

パラメータ	説明
<i>numeric-expression</i>	数値 (度)。この角度がラジアンに変換されます。

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、約 0.5236 の値を返します。

```
SELECT RADIANS( 30 ) FROM iq_dummy
```

RAND 関数 [数値]

$0 \leq x < 1$ 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

パラメータ	説明
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る。

戻り値

DOUBLE

備考

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。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

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_tvs) 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
```

RANK 関数 [統計]

グループ内の項目をランク付けします。

構文

```
RANK ( ) OVER ( [ PARTITION BY ] ORDER BY expression [ ASC | DESC ] )
```

パラメータ

パラメータ	説明
expression	ソートを指定する。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る。

戻り値

INTEGER

備考

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て、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** の相違点は、順位が同じである場合に、**DENSE_RANK** ではランク順に隔たりが置かれないことです。**RANK** では隔たりが置かれます。

RANK には **OVER (ORDER BY)** 句が必須です。**ORDER BY** 句は、ランク付けを実行するパラメータと、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** 句の内部のみで使用するもので、**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 (ORDER BY) 句内の **PARTITION BY** ウィンドウパーティション句はオプションです。

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。

OVER 句は、関数がクエリの結果セットに対して処理を行うことを示します。結果セットは、**FROM**、**WHERE**、**GROUP BY**、**HAVING** の各句がすべて評価された後で返されるローです。**OVER** 句は、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。

RANK は、**SELECT** または **INSERT** 文の **SELECT** リスト、および **SELECT** 文の **ORDER BY** 句でのみ使用できます。**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、**UPDATE** 文や **DELETE** 文の

SQL 関数

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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server または SQL Anywhere でサポートされていません。

例

RANK 関数は、次の文のように使用します。

```
SELECT Surname, Sex, Salary, RANK() OVER (PARTITION BY Sex
ORDER BY Salary DESC) AS RANK FROM Employees
WHERE State IN ('CA', 'AZ') AND DepartmentID IN (200, 300)
ORDER BY Sex, Salary DESC;
```

このクエリの結果セットを次に示します。

Surname	Sex	Salary	RANK
Savarino	F	72300.000	1
Jordan	F	51432.000	2
Clark	F	45000.000	3
Coleman	M	42300.000	1
Overbey	M	39300.000	2

参照：

- DENSE_RANK 関数 [統計] (214 ページ)

REGR_AVGX 関数 [集合]

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)
```

パラメータ

パラメータ	説明
<i>dependent-expression</i>	独立した変数の影響を受ける変数。
<i>independent-expression</i>	結果に影響を与える変数。

戻り値
DOUBLE

備考

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、**REGR_AVGX** は NULL を返します。

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)

注意：ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

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;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_AVGY 関数 [集合]

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*)

パラメータ

パラメータ	説明
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。

戻り値

DOUBLE

備考

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、**REGR_AVGY** は NULL を返します。

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)

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、独立変数である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;
```


REGR_COUNT 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

戻り値

INTEGER

備考

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** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、回帰直線の調整に使用された 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;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_INTERCEPT 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

戻り値

DOUBLE

備考

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、**REGR_INTERCEPT** は NULL を返します。

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)
```

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、値 1874.5805688517603 を返します。

```
SELECT REGR_INTERCEPT( Salary, ( YEAR( NOW() ) -
YEAR( BirthDate ) ) ) FROM Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_R2 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

戻り値

DOUBLE

備考

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、**REGR_R2** は NULL を返します。

REGR_R2 は、dependent-expression または *independent-expression* が NULL であるペアをすべて除外した後で、(dependent-expression と *independent-expression* の) ペアのセットに適用されます。その後、SAP Sybase IQ は次のアルゴリズムを適用します。

SQL 関数

- $\text{VAR_POP}(x) = 0$ の場合、**REGR_R2** は $\text{VAR_POP}(x)$ を計算して NULL を返します。 $\text{VAR_POP}(y) = 0$ の場合、 $\text{VAR_POP}(y)$ を計算して値 1 を返します。
- $\text{VAR_POP}(x)$ と $\text{VAR_POP}(y)$ がいずれも 0 でない場合、 $\text{POWER}(\text{CORR}(y,x),2)$ が返されます。

y は *dependent-expression* を表し、 x は *independent-expression* を表します。

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、値 0.19379959710325653 を返します。

```
SELECT REGR_R2( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) ) FROM Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_SLOPE 関数 [集合]

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)
```

パラメータ

パラメータ	説明
<i>dependent-expression</i>	独立した変数の影響を受ける変数。

パラメータ	説明
independent-expression	結果に影響を与える変数。

戻り値
DOUBLE

備考

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、**REGR_SLOPE** は NULL を返します。

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)
```

注意：ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、値 935.3429749445614 を返します。

```
SELECT REGR_SLOPE( Salary, ( YEAR( NOW() ) -  
YEAR( BirthDate ) ) ) FROM Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_SXX 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

戻り値

DOUBLE

備考

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、**REGR_SXX** は NULL を返します。

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)
```

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、window-spec の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、値 5916.4800000000105 を返します。

```
SELECT REGR_SXX( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) FROM
Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_SXY 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

戻り値

DOUBLE

備考

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

SQL 関数

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)
```

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、値 5533938.004400015 を返します。

```
SELECT REGR_SXY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) )FROM  
Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REGR_SYY 関数 [集合]

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)
```

パラメータ

パラメータ	説明
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。

戻り値 DOUBLE

備考

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、**REGR_SYY** は NULL を返します。

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) * VAR_POP(y)
```

注意： ROLLUP と CUBE は、構文 1 の **GROUP BY** 句では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** 文でウィンドウ関数として使用する場合の用法を示しています。その場合は、関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T612 です。
- Sybase — SQL Anywhere 互換。

例

次の例は、値 26、708、672,843.3002 を返します。

```
SELECT REGR_SYY( Salary, ( YEAR( NOW() ) - YEAR( BirthDate ) ) ) FROM Employees;
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

REMAINDER 関数 [数値]

ある整数を別の整数で割った余りを返します。

構文

```
REMAINDER ( dividend, divisor )
```

パラメータ

パラメータ	説明
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。

戻り値

INTEGER

NUMERIC

備考

REMAINDER 関数は **MOD** 関数と同じです。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- 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、余りを算出してください。

例

次の文は、値 2 を返します。

```
SELECT REMAINDER( 5, 3 ) FROM iq_dummy
```

参照：

- MOD 関数 [数値] (278 ページ)

REPEAT 関数 [文字列]

文字列を指定された回数だけ連結します。

構文

```
REPEAT ( string-expression, integer-expression )
```

パラメータ

パラメータ	説明
string-expression	繰り返される文字列。
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。

戻り値

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **REPEAT** を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- 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。

例

次の文は、値 "repeatrepeatrepeat" を返します。

```
SELECT REPEAT( 'repeat', 3 ) FROM iq_dummy
```

参照：

- REPLACE 関数 [文字列] (325 ページ)
- REPLICATE 関数 [文字列] (327 ページ)

REPLACE 関数 [文字列]

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。

パラメータ	説明
<i>original-string</i>	検索される文字列。この文字列の長さに制限はない。
<i>search-string</i>	検索して <i>replace-string</i> に置き換えられる文字列。この文字列は 255 バイトに制限されている。 <i>search-string</i> が空の文字列の場合は、元の文字列がそのまま返される。
<i>replace-string</i>	置換文字列。 <i>search-string</i> を置き換える。任意の長さの文字列を指定できる。 <i>replace-string</i> が空の文字列の場合は、検索されたすべての <i>search-string</i> が削除される。

戻り値

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **REPLACE** を使用する場合は、非構造化データ分析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。

備考

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、非構造化データ分析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。

この問題には、次の2つの対処方法があります。

- ローカルテナポラリテーブルを宣言し、**INSERT** を実行します。

```
DECLARE local temporary table #mytable
  (name_column char(10)) on commit preserve rows;
INSERT INTO #mytable SELECT REPLACE(name, '0', '1') FROM
dummy_table01;
```

- CAST** を使用します。

```
SELECT CAST(replace(name, '0', '1') AS Char(10)) into #mytable
from dummy_table01;
```

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。次に例を示します。

```
CREATE TABLE aa(a CHAR(5));
INSERT INTO aa VALUES ('CCCCC');
COMMIT;
SELECT a, CAST(REPLACE(a, 'C', 'ZZ') AS CHAR(5)) FROM aa;
```

標準と互換性

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- Sybase - Adaptive Server Enterprise 互換。

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。この文を実行すると、名前が変更されたテーブルを参照するストアプロシージャが修復されます(テーブル名を一意にすることをおすすめします)。

```
SELECT REPLACE (
    replace(proc_defn, 'OldTableName', 'NewTableName'),
    'create procedure',
    'alter procedure')
FROM SYS.SYSPROCEDURE
WHERE proc_defn LIKE '%OldTableName%'
```

次の例では、カンマ以外の区切り文字を **LIST** 関数に使用します。

```
SELECT REPLACE( list( table_id ), ',', '--')
FROM SYS.ISYSTAB
WHERE table_id <= 5
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)
- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)
- REPEAT 関数 [文字列] (324 ページ)
- REPLICATE 関数 [文字列] (327 ページ)

REPLICATE 関数 [文字列]

文字列を指定された回数だけ連結します。

構文

```
REPLICATE ( string-expression, integer-expression )
```

パラメータ

パラメータ	説明
string-expression	繰り返される文字列。
integer-expression	文字列を繰り返す回数。

戻り値

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **REPLICATE** を使用する場合は、非構造化データ分析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。

備考

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、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 "repeatrepeatrepeat" を返します。

```
SELECT REPLICATE( 'repeat', 3 ) FROM iq_dummy
```

参照：

- REPEAT 関数 [文字列] (324 ページ)
- REPLACE 関数 [文字列] (325 ページ)

REVERSE 関数 [文字列]

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、指定された文字列の文字順を逆にして返します。

構文

```
REVERSE ( expression | uchar_expr )
```

パラメータ

パラメータ	説明
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。

戻り値

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **REVERSE** を使用する場合は、非構造化データ分析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 **REVERSE** を正しいデータ型とサイズに設定する必要があります。

備考

- **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。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。

例

```
select reverse("abcd")
-----
dcba
```

```
select reverse(0x12345000)
-----
0x00503412
```

参照：

- 文字列演算子 (34 ページ)

SQL 関数

- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)

RIGHT 関数 [文字列]

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

パラメータ	説明
<i>string-expression</i>	左トランケートされる文字列。
<i>numeric-expression</i>	返す (文字列の末尾の) 文字数。

戻り値

LONG VARCHAR

LONG NVARCHAR

注意： 結果データ型は LONG VARCHAR です。 **SELECT INTO** 文で **RIGHT** を使用する場合は、非構造化データ分析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。

備考

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ことがあります。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 "olate" を返します。

```
SELECT RIGHT( 'chocolate', 5 ) FROM iq_dummy
```


参照：

- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- UCASE 関数 [文字列] (371 ページ)
- UPPER 関数 [文字列] (372 ページ)

ROUND 関数 [数値]

numeric-expression を *integer-expression* で指定した小数点以下の桁数に丸めます。

構文

```
ROUND ( numeric-expression, integer-expression )
```

パラメータ

パラメータ	説明
<i>numeric-expression</i>	関数に渡され、丸められる数値。
<i>integer-expression</i>	正の整数は、丸めを行う小数点以下の有効桁数を指定する。負の式は、丸めを行う小数点の左側の有効桁数を指定する。

戻り値

NUMERIC

ROUND_TO_EVEN データベースオプションが ON に設定されている場合、**ROUND** 関数は SAP Sybase IQ テーブルデータの端数を *integer-expression* に最も近い偶数に丸め、SQL Anywhere テーブルデータの動作に合わせます。オプションが OFF に設定されている場合、**ROUND** 関数は SAP Sybase IQ データの端数を四捨五入します。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 123.200 を返します。

```
SELECT ROUND( 123.234, 1 ) FROM iq_dummy
```

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)

次の例では、ROUND_TO_EVEN の設定が、返される値に影響します。

ROUND (値)	ROUND_TO_EVEN ON	ROUND_TO_EVEN OFF	注意
ROUND (convert (double, 123.45001), 1)	123.5	123.5	0.05001 は、0.1 の半分より大きい
ROUND (convert (double, 123.45000), 1)	123.4	123.5	0.0500 は 0.1 の半分

参照：

- TRUNCNUM 関数 [数値] (370 ページ)

ROW_NUMBER 関数 [統計]

ウィンドウパーティション内の各ローにユニークなロー番号を返すランク付け関数です。ウィンドウパーティションごとに新しい番号を割り当てます。

ウィンドウパーティションが存在しない場合は、結果セット内のローに 1 からテーブルのカーディナリティまでの番号が割り当てられます。

構文

```
ROW_NUMBER () OVER ([PARTITION BY window partition] ORDER BY window ordering)
```

パラメータ

パラメータ	説明
window partition	(オプション) 結果ローセットの分割方法を示す、カンマ区切りの 1 つ以上の値式。
window ordering	ローのソートのための式を定義する。ウィンドウパーティションを指定した場合はウィンドウパーティション内でのソート、指定しなかった場合は結果セット内でのソート。

備考

ROW_NUMBER 関数は、**OVER (ORDER BY)** ウィンドウ指定を必要とします。**OVER (ORDER BY)** 句内のウィンドウパーティション句はオプションです。**OVER (ORDER BY)** 句に、ウィンドウフレーム **ROWS/RANGE** 指定を含めることはできません。

標準と互換性

- SQL — ISO/ANSI SQL 準拠。SQL/OLAP 機能 T611 です。

例

次の例は、Employees テーブルから給与データを返し、部署 ID で結果セットをパーティションに分割し、入社日でデータを並べ替えます。**ROW_NUMBER** 関数は、各ローにロー番号を割り当てます。ウィンドウパーティションごとに新しいロー番号を割り当てます。

```
SELECT DepartmentID dID, StartDate, Salary,
ROW_NUMBER()OVER(PARTITION BY dID ORDER BY StartDate) FROM Employees
ORDER BY 1,2;
```

次の結果セットが返されます。

dID	StartDate	Salary	Row_number()
100	1986-10-14	42,998.000	1
100	1987-07-23	39,875.500	2
100	1988-03-23	37,400.000	3
100	1989-04-20	42,500.000	4
100	1990-01-15	42,100.000	5
200	1985-02-03	38,500.000	1
200	1987-02-19	39,300.000	2
200	1988-11-22	39,800.000	3
200	1989-06-01	34,892.000	4
200	1990-05-13	33,890.000	5
200	1990-07-11	37,803.000	6

ROWID 関数 [その他]

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

パラメータ	説明
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する。二重引用符は省略できるが、引用符を使うことはできない。

戻り値

UNSIGNED BIGINT

備考

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て使用できます。

FROMtable-name 句を指定する必要があります。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
.
.
.

rowid(Products)

10

次の文は、製品 ID の値が 400 よりも小さいすべてのローの製品 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
```

RTRIM 関数 [文字列]

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

パラメータ	説明
string-expression	削除される文字列。

戻り値

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **RTRIM** を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

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
```

参照：

- LTRIM 関数 [文字列] (272 ページ)

SECOND 関数 [日付と時刻]

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

パラメータ	説明
<i>datetime-expression</i>	日時の値。

戻り値

SMALLINT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 5 を返します。

```
SELECT SECOND( '1998-07-13 08:21:05' ) FROM iq_dummy
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)

- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)

SECONDS 関数 [日付と時刻]

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

パラメータ	説明
<i>datetime-expression</i>	日時。
<i>integer-expression</i>	<i>datetime-expression</i> に追加する秒数。 <i>integer-expression</i> 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、 <i>datetime-expression</i> は日時データ型として明示的にキャストする必要があります。

戻り値

INTEGER

TIMESTAMP

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文を実行すると、値 3600 が返ります。

```
SELECT ( SECONDS( '1998-07-13 06:07:12' ) -
SECONDS( '1998-07-13 05:07:12' ) ) FROM iq_dummy
```

SQL 関数

次の文を実行すると、2つの時刻の間の差である値 14400 が返ります。

```
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
```

SIGN 関数 [数値]

数値の符号を返します。

構文

```
SIGN ( numeric-expression )
```

パラメータ

パラメータ	説明
<i>numeric-expression</i>	符号を返す数値。

戻り値

SMALLINT

備考

負の数を指定すると、**SIGN** 関数は -1 を返します。

0 を指定すると、**SIGN** 関数は 0 を返します。

正の数を指定すると、**SIGN** 関数は 1 を返します。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 -1 を返します。

```
SELECT SIGN( -550 ) FROM iq_dummy
```

SIMILAR 関数 [文字列]

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

パラメータ	説明
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。

戻り値

SMALLINT

備考

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も何度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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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。

SIN 関数 [数値]

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

パラメータ	説明
numeric-expression	角度 (ラジアン)。

SQL 関数

戻り値
DOUBLE

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文は、値 0.496880 を返します。

```
SELECT SIN( 0.52 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)
- ATAN2 関数 [数値] (162 ページ)
- ATAN 関数 [数値] (161 ページ)
- ASIN 関数 [数値] (160 ページ)
- ACOS 関数 [数値] (157 ページ)
- COT 関数 [数値] (184 ページ)
- TAN 関数 [数値] (368 ページ)

SORTKEY 関数 [文字列]

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

パラメータ	説明
string-expression	<p>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。</p> <p>string-expression が NULL の場合、SORTKEY 関数は NULL 値を返す。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つ。</p> <p>SORTKEY 関数が処理できる入力文字列の長さに制限はない。SORTKEY の結果は、常に 1024 バイトに制限された VARBINARY データ型。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る。</p>
collation-name	<p>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 照合で使用されるソートキーを生成することもできる。</p> <p>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る。ただし、SAP Sybase IQ は SAP Sybase IQ 固有のオブジェクトに対して NCHAR_COLLATION をサポートしない。NCHAR_COLLATION は、SAP Sybase IQ サーバ上の SQL Anywhere オブジェクトに対してサポートされる。</p>
collation-id	<p>使用するソート順の ID 番号を指定する変数、定数 (整数)、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 照合にのみ適用される。</p>
collation-tailoring-string	<p>(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 (<i>collation-tailoring-string</i>) を指定する。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」ペアの形を取る。次に例を示す。</p> <pre>'UCA(locale=es;case=LowerFirst;accent=respect)'</pre> <p>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。</p> <p>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が SQL Anywhere データベースでサポートされます。その他のすべての照合については、大文字と小文字の区別の調整のみがサポートされます。</p>

戻り値
BINARY

備考

SORTKEY 関数が生成する値を使用し、事前定義済みのソート順の動作に基づいて、結果を順序付けることができます。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 関数は、特定のソート順の基準セットに対して返された値が、VARBINARY データ型で実行されるバイナリ比較で使用できることを保証します。

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。

- SAP Sybase IQ でサポートされる照合をラベル順に表示するには、`iqinit -l` を実行します。
- 次の表に Adaptive Server の照合を示します。

説明	照合名	照合 ID
デフォルト Unicode 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

説明	照合名	照合 ID
CP 850 スカンジナビア：大文字小文字の優先指定なし	scannocp	48
GB ピンイン	gbpinyin	該当なし
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	espnoac	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	cyrnocs	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	sjsbin	179
Unicode UTF-8 バイナリソート	utf8bin	24

説明	照合名	照合 ID
EUC JIS バイナリ順	eucjisbn	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

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ため、異なるソート順が生成されることがあります。**SORTKEY** でもデータベース照合と同じ動作が実行されるようにするには、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、*collation-name* に `db_collation` を指定します。次に例を示します。

```
SORTKEY( expression, 'db_collation' )
```

注意： 15.0 より前のバージョンの SAP 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値と同じ値は含まれていません。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。15.0 より前のバージョンの SAP Sybase IQ を使用して生成されたデータベース内のソートキー値は、すべて再生成してください。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

例

次の文は、Employees テーブルをクエリし、すべての従業員の FirstName 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' );
```

SOUNDEX 関数 [文字列]

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

パラメータ	説明
string-expression	文字列。

戻り値

SMALLINT

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

SQL 関数

- Sybase — Adaptive Server 互換。ただし、Adaptive Server は結果を CHAR(4) で返し、SAP Sybase IQ は整数で返します。

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。

参照：

- DIFFERENCE 関数 [文字列] (216 ページ)

SP_HAS_ROLE 関数 [システム]

指定されたシステム権限またはユーザ定義ロールが呼び出し側ユーザに付与されているかどうかを示す整数値を返します。ユーザ定義ストアプロシージャ内で権限チェックに使用された場合、ユーザが権限チェックに失敗すると、**SP_HAS_ROLE** はエラーメッセージを返します。

構文

```
dbo.sp_has_role( [rolename], [grant_type], [throw_error] )
```

パラメータ

パラメータ	説明
rolename	システム権限またはユーザ定義ロールの名前。
grant_type	有効な値は ADMIN と NO ADMIN。NULL が指定されるか、または何も指定されない場合、NO ADMIN がデフォルトで使用される。
throw_error	有効な値： <ul style="list-style-type: none">• [1] – システム権限またはユーザ定義ロールが呼び出し側ユーザに付与されていない場合、指定されたエラーメッセージを表示する。• [0] – (デフォルト) 指定されたシステム権限またはユーザ定義ロールが呼び出し側ユーザに付与されていない場合、エラーメッセージを表示しない。

戻り値

値	説明
1	システム権限またはユーザ定義ロールは呼び出し側ユーザに付与されている。
0 またはパーミッションがありません: このコマンド / プロシージャを実行するためのパーミッションがありません。	システム権限またはユーザ定義ロールは呼び出し側ユーザに付与されていない。throw_error 引数が 1 に設定されている場合は、値 0 の代わりにエラーメッセージが返される。
-1	指定されたシステム権限またはユーザ定義ロールが存在しない。throw_error 引数が 1 に設定されている場合でも、エラーメッセージは表示されない。

備考

grant_type 引数の値が ADMIN である場合、この関数は、呼び出し側ユーザがシステム権限の管理権限を持っているかどうかをチェックします。grant_type 引数の値が NO ADMIN である場合、この関数は、呼び出し側ユーザがシステム権限またはロールを使用する権限を持っているかどうかをチェックします。

grant_type 引数が指定されていない場合、デフォルトで NO ADMIN が使用され、出力は、指定されたシステム権限またはユーザ定義ロールが直接的または間接的に呼び出し側ユーザに付与されているかどうかのみを示します。

rolename と grant_type の両方の引数が NULL で、throw_error 引数が 1 である場合、エラーメッセージが表示されます。この処理は、呼び出し側ユーザの特定のシステム権限の存在をチェックした後ではなく、カタログテーブルから特定の値を読み込んだ後にエラーメッセージを表示するようなストアードプロシージャの場合に役に立つ可能性があります。

注意: 引数 rolename と grant_type が NULL、throw_error が 1 に設定されている場合、またはこれら 3 つの引数がすべて NULL に設定されている場合は、パーミッションがないというエラーメッセージが返されます。

例

次のシナリオを検討します。

- u1 は、WITH NO ADMIN OPTION 句を使用して CREATE ANY PROCEDURE システム権限が付与されている。
- u1 は、CREATE ANY TABLE システム権限を付与されていない。
- u1 は、WITH ADMIN ONLY OPTION 句を使用してユーザ定義ロール Role_A が付与されている。
- Role_B は存在するが、u1 には付与されていない。
- ロール Role_C は存在しない。

このシナリオにもとづいて、次の各コマンドを実行します。

- `sp_has_role 'create any procedure'`

値 1 を返します。これは、u1 に CREATE ANY PROCEDURE システム権限が付与されていることを示します。

- `sp_has_role 'create any table'`

値 0 を返します。これは、u1 に CREATE ANY TABLE システム権限が付与されていないことを示します。throw_error 引数は指定されていないので、エラーメッセージは返されません。

- `sp_has_role 'create any procedure','admin',1`

「Permission denied」エラーメッセージを返します (throw_error=1)。u1 に CREATE ANY PROCEDURE システム権限は付与されていますが、それに対する管理権限は u1 には付与されていません。

- `sp_has_role 'Role_A'`

値 1 を返します。これは、u1 にロール Role_A が付与されていることを示します。

- `sp_has_role 'Role_A','admin',1`

値 1 を返します。これは、u1 にロール Role_A が管理権限付きで付与されていることを示します。

- `sp_has_role 'Role_B'`

値 0 を返します。これは、u1 にロール Role_B が付与されていないことを示します。throw_error 引数は指定されていないので、エラーメッセージは返されません。

- `sp_has_role 'Role_C'`

値 -1 を返します。これは、ロール Role_C が存在しないことを示します。

- `sp_has_role 'Role_C',NULL,1`

値 -1 を返します。これは、ロール Role_C が存在しないことを示します。

SPACE 関数 [文字列]

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

パラメータ	説明
integer-expression	返されるスペースの数。

戻り値

LONG VARCHAR

注意： 結果データ型は LONG VARCHAR です。SELECT INTO 文で SPACE を使用する場合は、非構造化データ分析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 SPACE 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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

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
```

SQLFLAGGER 関数 [その他]

特定の SQL 文が、指定した標準に準拠しているかどうかを返します。

構文

```
SQLFLAGGER ( sql-standard-string, sql-statement-string )
```

パラメータ

パラメータ	説明
sql-standard-string	<p>準拠をテストする規格レベル。指定できる値は、SQL_FLAGGER_ERROR_LEVEL データベースオプションと同じ。</p> <ul style="list-style-type: none"> SQL:2003/Core コア SQL/2003 構文に対する準拠性をテストする。 SQL:2003/Package 上級 SQL/2003 構文に対する準拠性をテストする。 SQL:1999/Core コア SQL/1999 構文に対する準拠性をテストする。 SQL:1999/Package 上級 SQL/1999 構文に対する準拠性をテストする。 SQL:1992/Entry 初級レベル SQL/1992 構文に対する準拠性をテストする。 SQL:1992/Intermediate 中級レベル SQL/1992 構文に対する準拠性をテストする。 SQL:1992/Full 上級 SQL/1992 構文に対する準拠性をテストする。
sql-statement-string	準拠性をチェックする SQL 文。

戻り値

LONG VARCHAR

注意： 結果データ型は LONG VARCHAR です。SELECT INTO 文で SQLFLAGGER を使用する場合は、非構造化データ分析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 SQLFLAGGER を正しいデータ型とサイズに設定する必要があります。

備考

また、SQL プリプロセッサユーティリティ iqsqpp を使用して、特定の SQL92 セットの一部ではない Embedded SQL にフラグを設定することもできます。『ユーティリティガイド』の「iqsqpp SQL プリプロセッサユーティリティ」を参照してください。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、使用できない拡張機能が検出されたときに返されるメッセージの例を示します。

```
SELECT SQLFLAGGER( 'SQL:2003/Package', 'SELECT top 1 dummy_col FROM
sys.dummy ORDER BY dummy_col' );
```

この文は、メッセージ '0AW03 Disallowed language extension detected in syntax near 'top' on line 1' を返します。

次の文は、使用できない拡張性を含まないので、'00000' を返します。

```
SELECT SQLFLAGGER( 'SQL:2003/Package', 'SELECT dummy_col FROM
sys.dummy' );
```

SQRT 関数 [数値]

数値の平方根を返します。

構文

```
SQRT ( numeric-expression )
```

パラメータ

パラメータ	説明
<i>numeric-expression</i>	平方根が計算される数値。

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 3 を返します。

```
SELECT SQRT( 9 ) FROM iq_dummy
```

SQUARE 関数 [数値]

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

パラメータ	説明
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。他のデータ型を指定すると、 SQUARE 関数ではエラーが返る。戻り値は、 DOUBLE データ型。

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

STDDEV 関数 [集合]

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

パラメータ	説明
expression	任意の数値式 (FLOAT、REAL、または DOUBLE 精度)。

戻り値

DOUBLE

備考

STDDEV では、次の計算式が使用されます。

$$stddev = \sqrt{variance}$$

STDDEV は、結果を **DOUBLE** 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** が返されません。

STDDEV は、キーワード **DISTINCT** をサポートしません。**STDDEV** 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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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'
```

Name	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'
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)
- STDDEV_SAMP 関数 [集合] (355 ページ)
- VARIANCE 関数 [集合] (378 ページ)

STDDEV_POP 関数 [集合]

数値式からなる母標準偏差を DOUBLE として計算します。

構文

```
STDDEV_POP ( [ ALL ] expression )
```

パラメータ

パラメータ	説明
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)。

戻り値

DOUBLE

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、異なる期間における注文ごとの項目数で平均と平方偏差をリストします。

```
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;
```


Year	Quarter	Average	Variance
2000	1	25.775148	14.2794
2000	2	27.050847	15.0270
...

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

STDDEV_SAMP 関数 [集合]

数値式からなる標本標準偏差を DOUBLE として計算します。

構文

```
STDDEV_SAMP ( [ ALL ] expression )
```

パラメータ

パラメータ	説明
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)。

戻り値

DOUBLE

備考

注意： STDDEV_SAMP は、STDDEV のエイリアスです。

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

SQL 関数

- Sybase — Adaptive Server Enterprise ではサポートされていません。

例

次の文は、異なる期間における注文ごとの項目数で平均と平方偏差をリストします。

```
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;
```

Year	Quarter	Average	Variance
2000	1	25.775148	14.3218
2000	2	27.050847	15.0696
...

参照：

- ウィンドウ集合関数の使用法 (89 ページ)
- STDDEV 関数 [集合] (352 ページ)
- VARIANCE 関数 [集合] (378 ページ)

STR 関数 [文字列]

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

パラメータ	説明
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。
decimal	返される文字の小数点以下の桁数。デフォルトは 0。

戻り値

VARCHAR

備考

数値の整数部分が指定された長さを超える場合は、NULL が返されます。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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、6つのスペースの後に1234が続く、合計で10の文字からな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
```

STR_REPLACE 関数 [文字列]

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 (*string_expr1*) 内に出現する2番目の文字列式 (*string_expr2*) のすべてのインスタンスを、3番目の式 (*string_expr3*) で置き換えます。

構文

```
REPLACE ( string_expr1, string_expr2, string_expr3 )
```

パラメータ

パラメータ	説明
<i>string_expr1</i>	検索される側のソース文字列または文字列式。CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY データ型として表される。
<i>string_expr2</i>	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。CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY データ型として表される。
<i>string_expr3</i>	置換文字列式。CHAR、VARCHAR、UNICHAR、UNIVARCHAR、VARBINARY、または BINARY データ型として表される。

備考

STR_REPLACE は、**REPLACE** 関数のエイリアスです。

- 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。
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、**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変数である場合、結果の長さは、**SAP 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 **SAP 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。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。

例

文字列 *cdefghi* 内の文字列 *def* を *yyy* に置き換えます。

```
select replace("cdefghi", "def", "yyy")
-----
cyyyghi
(1 row(s) affected)
```

すべてのスペースを "toyota" に置き換えます。

```
select str_replace ("chevy, ford, mercedes", " ", "toyota")
-----
chevy, toyotaford, toyotamercedes
(1 row(s) affected)
```

3 つ目のパラメータで **NULL** を許可し、*string_expr2* が **NULL** に置換されるようにします。これにより、**STR_REPLACE** が事実上「文字列削除」の操作に変わります。*"abcdefghijklm"* が返されます。

```
select str_replace("abcdefghijklm", "def", NULL)
-----
```

```
abcghijklm
(1 row affected)
```

参照：

- BIT_LENGTH 関数 [文字列] (165 ページ)
- BYTE_LENGTH 関数 [文字列] (166 ページ)
- CHAR_LENGTH 関数 [文字列] (171 ページ)
- COL_LENGTH 関数 [システム] (175 ページ)
- DATALENGTH 関数 [システム] (189 ページ)
- LEN 関数 [文字列] (262 ページ)
- LENGTH 関数 [文字列] (263 ページ)
- OBJECT_NAME 関数 [システム] (294 ページ)
- OCTET_LENGTH 関数 [文字列] (295 ページ)

STRING 関数 [文字列]

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

パラメータ	説明
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る。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る。NULL は空の文字列 ("") として扱われる。

戻り値

LONG VARCHAR

LONG NVARCHAR

LONG BINARY

注意： 結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **STRING** を使用する場合は、非構造化データ分析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 **STRING** を正しいデータ型とサイズに設定する必要があります。

備考

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。

SQL 関数

すべてのパラメータが NULL の場合、**STRING** は NULL を返します。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

STRTOUUID 関数 [文字列]

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

パラメータ	説明
string-expression	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX 形式の文字列。

戻り値

UNIQUEIDENTIFIER

備考

XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXXX 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 SAP Sybase IQ データベースに挿入できます。

標準と互換性

- SQL – ISO/ANSI SQL 文法の Transact-SQL 拡張。
- Sybase – Adaptive Server では、サポートされていません。

例

```
CREATE TABLE T (
  pk uniqueidentifier primary key,
  c1 int);
INSERT INTO T (pk, c1)
```

```
VALUES (STRTOUUID
('12345678-1234-5678-9012-123456789012'), 1);
```

参照：

- バイナリデータ型 (398 ページ)
- NEWID 関数 [その他] (282 ページ)
- UUIDTOSTR 関数 [文字列] (374 ページ)
- 文字データ型 (389 ページ)
- バイナリデータ型 (874 ページ)

STUFF 関数 [文字列]

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

```
STUFF ( string-expression1, start, length, string-expression2 )
```

パラメータ

パラメータ	説明
string-expression1	STUFF 関数によって変更される文字列。
start	文字の削除を開始する位置。文字列の先頭文字の位置が 1 になる。
length	削除される文字数。
string-expression2	挿入する文字列。 STUFF 関数を使用して文字列の一部を削除するには、NULL の置換文字列を使用する。

戻り値

LONG NVARCHAR

備考

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。

STUFF 関数は、次のような場合は NULL を返します。

- 最初の 3 つのパラメータのいずれかが NULL 値である。
- start パラメータまたは length パラメータのいずれかが負の値である。
- start パラメータが string-expression1 の長さよりも大きい。

標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文を実行すると、値 "chocolate pie" が返ります。

```
SELECT STUFF( 'chocolate cake', 11, 4, 'pie' )
FROM iq_dummy
```

SUBSTRING 関数 [文字列]

文字列の一部を返します。

構文

```
{ SUBSTRING | SUBSTR } ( string-expression, start [ , length ] )
```

パラメータ

パラメータ	説明
string-expression	部分文字列が返される文字列。
start	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する。文字列の先頭文字の位置が 1 になる。
length	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、部分文字列が終了する。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> 文字の位置で終了する。

戻り値

LONG VARCHAR

LONG NVARCHAR

LONG BINARY

注意： 結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **STRING** を使用する場合は、非構造化データ分析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 **STRING** を正しいデータ型とサイズに設定する必要があります。

備考

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て使うと、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。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラジオブジェクトデータを使用できます。

ansi_substring データベースオプションを ON (デフォルト) に設定すると、負の値は無効となります。

『非構造化データ分析』の「SUBSTRING 関数」を参照してください。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — **SUBSTR** は、Adaptive Server 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。

例

次の文は、"back" を返します。

```
SELECT SUBSTRING ( 'back yard', 1 , 4 )  
FROM iq_dummy
```

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
```

参照：

- CHARINDEX 関数 [文字列] (173 ページ)

ANSI SUBSTRING オプション [TSQL]

start または *length* パラメータに負の値が指定されている場合に、SUBSTRING (SUBSTR) 関数の動作を制御します。

指定できる値

ON、OFF

デフォルト値

ON

スコープ

オプションは、データベース (PUBLIC) レベルまたはユーザレベルで設定できます。データベースレベルで設定した場合、値は新しいユーザのデフォルト値になりますが、既存のユーザには影響を与えません。ユーザレベルで設定した場合は、そのユーザの PUBLIC 値のみが上書きされます。自分のオプションを設定する場合は、システム権限は必要ありません。自分以外のユーザのオプションをデータベースレベルまたはユーザレベルで設定する場合は、システム権限が必要です。

このオプションを設定するには、SET ANY PUBLIC OPTION システム権限が必要です。個々の接続または PUBLIC ロールに一時的に設定できます。すぐに有効になります。

備考

ANSI_SUBSTRING オプションを ON に設定した場合、**SUBSTRING** 関数は ANSI/ISO SQL/2003 と同じ動作をします。開始オフセットが負またはゼロの場合は、文字列の左側に非文字が埋め込まれているものと解釈され、負の長さが指定された場合はエラーが発生します。

このオプションを OFF に設定した場合、**SUBSTRING** 関数の動作は以前のバージョンの SAP Sybase IQ と同じになります。負の開始オフセットは、文字列の末尾からのオフセットになり、負の長さは、開始オフセットから左に向かって指定の部分文字列の終端までの長さになります。開始オフセット 0 を使用した場合、開始オフセット 1 と同じ結果となります。

SUBSTRING 関数では、正以外の開始オフセットおよび負の長さを使用しないでください。可能な限り、**SUBSTRING** 関数の代わりに **LEFT** 関数または **RIGHT** 関数を使用してください。

例

次の例は、**ANSI_SUBSTRING** オプションの設定に応じて **SUBSTRING** 関数によって返される値の違いを示しています。

```
SUBSTRING( 'abcdefgh',-2,4 );
ansi_substring = Off ==> 'gh'
// substring starts at second-last character
ansi_substring = On ==> 'gh'
// takes the first 4 characters of
// ???abcdefgh and discards all ?
```

```
SUBSTRING( 'abcdefgh',4,-2 );
ansi_substring = Off ==> 'cd'
ansi_substring = On ==> value -2 out of range
for destination
```

```
SUBSTRING( 'abcdefgh',0,4 );
ansi_substring = Off ==> 'abcd'
ansi_substring = On  ==> 'abcd'
```

SUBSTRING64 関数 [文字列]

SUBSTRING64 関数は、ラージオブジェクトカラムまたは変数パラメータの可変長文字列を返します。

備考

SUBSTRING64 は、LONG VARCHAR カラムと LONG BINARY カラム、および任意のデータサイズの LONG VARCHAR 変数と LONG BINARY 変数の検索をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。

『非構造化データ分析』の「SUBSTRING64 関数」を参照してください。

SUM 関数 [集合]

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

パラメータ	説明
expression	合計する対象。通常はカラムを指定する。
DISTINCT column-name	ローの各グループに対する <i>column-name</i> のうち、一意な値の合計を計算する。この方法で使用することはほとんどないが、万全を期すために含まれている。

戻り値

INTEGER

DOUBLE

NUMERIC

備考

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。

SQL 関数

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

次の文は、値 3749146.740 を返します。

```
SELECT SUM( salary )  
FROM Employees
```

参照：

- AVG 関数 [集合] (162 ページ)
- COUNT 関数 [集合] (187 ページ)
- ウィンドウ集合関数の使用法 (89 ページ)

SUSER_ID 関数 [システム]

ユーザ ID 番号を整数で返します。

構文

```
SUSER_ID ( [ user-name ] )
```

パラメータ

パラメータ	説明
user-name	ユーザ名。

戻り値

INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

次の文は、ユーザ ID 番号 1 を返します。

```
SELECT SUSER_ID ('DBA') FROM iq_dummy
```

次の文は、ユーザ ID 番号 0 を返します。

```
SELECT SUSER_ID ('SYS') FROM iq_dummy
```

参照：

- SUSER_NAME 関数 [システム] (367 ページ)
- USER_ID 関数 [システム] (373 ページ)
- USER_NAME 関数 [システム] (373 ページ)

SUSER_NAME 関数 [システム]

ユーザ名を返します。

構文

```
SUSER_NAME ( [ user-id ] )
```

パラメータ

パラメータ	説明
user-id	ユーザ ID 番号。

戻り値

LONG VARCHAR

注意： 結果データ型は LONG VARCHAR です。SELECT INTO 文で SUSER_NAME を使用する場合は、非構造化データ分析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 SUSER_NAME 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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。Adaptive Server では、SUSER_NAME はサーバのユーザ名を返します。

例

次の文は、値 DBA を返します。

```
SELECT SUSER_NAME ( 1 ) FROM iq_dummy
```

次の文は、値 SYS を返します。

```
SELECT SUSER_NAME ( 0 ) FROM iq_dummy
```

参照：

- SUSER_ID 関数 [システム] (366 ページ)
- USER_ID 関数 [システム] (373 ページ)
- USER_NAME 関数 [システム] (373 ページ)

TAN 関数 [数値]

数値の正接を返します。

構文

```
TAN ( numeric-expression )
```

パラメータ

パラメータ	説明
numeric-expression	角度 (ラジアン)。

戻り値

DOUBLE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 互換。

例

値 0.572561 を返します。

```
SELECT TAN( 0.52 ) FROM iq_dummy
```

参照：

- COS 関数 [数値] (183 ページ)
- ATAN2 関数 [数値] (162 ページ)
- ATAN 関数 [数値] (161 ページ)
- ASIN 関数 [数値] (160 ページ)
- ACOS 関数 [数値] (157 ページ)
- COT 関数 [数値] (184 ページ)
- SIN 関数 [数値] (339 ページ)

TODAY 関数 [日付と時刻]

現在の日付を返します。これは **CURRENT DATE** に対応する古い構文です。

構文

```
TODAY ( * )
```

戻り値

DATE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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。

```
SELECT TODAY( * ) FROM iq_dummy
```

TRIM 関数 [文字列]

文字列から先行空白と後続空白を削除します。

構文

```
TRIM ( string-expression )
```

パラメータ

パラメータ	説明
string-expression	削除される文字列。

戻り値

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **TRIM** を使用する場合は、非構造化データ分析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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise 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。

例

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。

```
SELECT TRIM( ' chocolate ' ) FROM iq_dummy
```

TRUNCNUM 関数 [数値]

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

```
TRUNCNUM ( numeric-expression, integer-expression )
```

パラメータ

パラメータ	説明
<i>numeric-expression</i>	切り捨てられる数値。
<i>integer-expression</i>	正の整数は、丸めを行う小数点以下の有効桁数を指定する。負の式は、丸めを行う小数点の左側の有効桁数を指定する。

戻り値

NUMERIC

備考

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。

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。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、値 600 を返します。

```
SELECT TRUNCNUM( 655, -2 ) FROM iq_dummy
```

次の文は、値 655.340 を返します。

```
SELECT TRUNCNUM( 655.348, 2 ) FROM iq_dummy
```

参照：

- **ROUND 関数 [数値]** (331 ページ)

UCASE 関数 [文字列]

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

パラメータ	説明
string-expression	大文字に変換される文字列。

戻り値

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

注意： 結果データ型は LONG VARCHAR です。SELECT INTO 文で UCASE を使用する場合は、非構造化データ分析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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server 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。

例

次の文を実行すると、値 "CHOCOLATE" が返ります。

```
SELECT UCASE ( 'ChocoLate' ) FROM iq_dummy
```

参照：

- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UPPER 関数 [文字列] (372 ページ)

UPPER 関数 [文字列]

文字列内のすべての文字を大文字に変換します。

構文

```
UPPER ( string-expression )
```

パラメータ

パラメータ	説明
string-expression	大文字に変換される文字列。

戻り値

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

注意： 結果データ型は LONG VARCHAR です。 **SELECT INTO** 文で **UPPER** を使用する場合は、非構造化データ分析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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server Enterprise 互換。

例

次の文を実行すると、値 "CHOCOLATE" が返ります。

```
SELECT UPPER( 'ChocoLate' ) FROM iq_dummy
```

参照：

- LCASE 関数 [文字列] (258 ページ)
- LEFT 関数 [文字列] (261 ページ)
- LOWER 関数 [文字列] (271 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- REVERSE 関数 [文字列] (329 ページ)
- RIGHT 関数 [文字列] (330 ページ)
- UCASE 関数 [文字列] (371 ページ)

USER_ID 関数 [システム]

ユーザ ID 番号を整数で返します。

構文

```
USER_ID ( [ user-name ] )
```

パラメータ

パラメータ	説明
user-name	ユーザ名。

戻り値

INT

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。

例

次の文は、ユーザ ID 番号 1 を返します。

```
SELECT USER_ID ('DBA') FROM iq_dummy
```

次の文は、ユーザ ID 番号 0 を返します。

```
SELECT USER_ID ('SYS') FROM iq_dummy
```

参照：

- SUSER_ID 関数 [システム] (366 ページ)
- SUSER_NAME 関数 [システム] (367 ページ)
- USER_NAME 関数 [システム] (373 ページ)

USER_NAME 関数 [システム]

ユーザ名を返します。

構文

```
USER_NAME ( [ user-id ] )
```

パラメータ

パラメータ	説明
user-id	ユーザ ID 番号。

戻り値

LONG VARCHAR

注意：結果データ型は LONG VARCHAR です。**SELECT INTO** 文で **USER_NAME** を使用する場合は、非構造化データ分析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 **USER_NAME** 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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SAP Sybase IQ 用に実装された Adaptive Server 関数です。Adaptive Server 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。

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
```

参照：

- SUSER_ID 関数 [システム] (366 ページ)
- SUSER_NAME 関数 [システム] (367 ページ)
- USER_ID 関数 [システム] (373 ページ)

UUIDTOSTR 関数 [文字列]

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

表 52 : パラメータ

パラメータ	説明
uuid-expression	一意な識別子の値。

戻り値

VARCHAR

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

参照：

- バイナリデータ型 (398 ページ)
- NEWID 関数 [その他] (282 ページ)
- STRTOUUID 関数 [文字列] (360 ページ)
- 文字データ型 (389 ページ)
- バイナリデータ型 (874 ページ)

VAR_POP 関数 [集合]

数値式からなる母集団の統計上の平方偏差を DOUBLE 型として計算します。

構文

```
VAR_POP ( [ ALL ] expression )
```

パラメータ

パラメータ	説明
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)。

戻り値

DOUBLE

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、異なる期間における注文ごとの項目数で平均と平方偏差をリストします。

```
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
```

Year	Quarter	Average	Variance
2000	1	25.775148	203.9021
2000	2	27.050847	225.8109
...

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

VAR_SAMP 関数 [集合]

数値式からなる標本の統計上の平方偏差を DOUBLE 型として計算します。

構文

```
VAR_SAMP ( [ ALL ] expression )
```

パラメータ

パラメータ	説明
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)。

戻り値

DOUBLE

備考

注意： VAR_SAMP は、VARIANCE のエイリアスです。

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。

SAP Sybase IQ では、1 要素の入力セットの場合、NULL によって NULL が返されます。

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

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- Sybase — Adaptive Server では、サポートされていません。

例

次の文は、異なる期間における注文ごとの項目数で平均と平方偏差をリストします。

```
SELECT year( ShipDate ) AS Year, quarter( ShipDate )
       AS Quarter, AVG( Quantity ) AS Average,
       VAR_SAMP( Quantity ) AS Variance
```

SQL 関数

```
FROM SalesOrderItems GROUP BY Year, Quarter  
ORDER BY Year, Quarter
```

Year	Quarter	Average	Variance
2000	1	25.775148	205.1158
2000	2	27.050847	227.0939
...

参照：

- ウィンドウ集合関数の使用法 (89 ページ)

VARIANCE 関数 [集合]

数値セットの分散を返します。

構文

```
VARIANCE ( [ ALL ] expression )
```

パラメータ

パラメータ	説明
expression	任意の数値データ型 (FLOAT、REAL、または DOUBLE) の式。 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)。

戻り値

DOUBLE

備考

VARIANCE では、次の計算式が使用されます。

$$\text{var} = \frac{n \sum x^2 - (\sum x)^2}{n(n-1)}$$

VARIANCE は、結果を double-precision floating-point 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 要素の入力セットの場合は 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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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 8.33333333333334327 を返します。

```
SELECT VARIANCE ( UnitPrice ) FROM Products  
WHERE name = 'Tee Shirt'
```

参照：

- ウィンドウ集合関数の使用法 (89 ページ)
- STDDEV 関数 [集合] (352 ページ)
- STDDEV_SAMP 関数 [集合] (355 ページ)

WEEKS 関数 [日付と時刻]

任意の開始日時から経過した週の数を返すか、指定された2つの日時の間の週の数を返すか、または `integer-expression` で指定された週の数を日時に追加します。

構文

```
WEEKS ( datetime-expression
| datetime-expression, datetime-expression
| datetime-expression, integer-expression )
```

パラメータ

パラメータ	説明
<code>datetime-expression</code>	日時。
<code>integer-expression</code>	<code>datetime-expression</code> に追加する週数。 <code>integer-expression</code> が負の場合、日時の値から適切な週数が引かれる。時間、分、秒は無視される。整数式を指定する場合は、 <code>datetime-expression</code> を DATETIME データ型として明示的にキャストする必要がある。

戻り値

構文 1 は、INTEGER を返します。

構文 2 は、TIMESTAMP を返します。

備考

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せん。時間、分、秒は無視されます。この関数は、`DATE_FIRST_DAY_OF_WEEK` オプションの影響を受けません。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。

- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文を実行すると、値 104278 が返されます。

```
SELECT WEEKS ( '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9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
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)

WEIGHTED_AVG 関数 [集合]

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)
```

window-spec：以下の「使用法」を参照してください。

パラメータ

パラメータ	説明
expression	計算する加重値の数値式。

備考

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 1 : **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 つ異なる加重移動平均と一緒に平均するか、**EXP_WEIGHTED_AVG** 関数を使用できます。

関数構文 (インライン) または **SELECT** 文の **WINDOW** 句で、*window-spec* の要素を指定できます。

window-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。

標準と互換性

- **SQL – ISO/ANSI SQL** 文法のベンダ拡張。

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係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 53 : WEIGHTED_AVG の結果セット

DepartmentID	Surname	Salary	W_AVG
100	Lull	87,900.000	87,900.000000
100	Gowda	59,840.000	69,193.333333
200	Sterling	64,900.000	64,900.000000
200	Kelly	87,500.000	79,966.666667
300	Litton	58,930.000	58,930.000000
400	Evans	68,940.000	68,940.000000
400	Charlton	28,300.000	41,846.666667
400	Francis	53,870.000	47,858.333333

参照：

- EXP_WEIGHTED_AVG 関数 [集合] (223 ページ)
- ウィンドウ集合関数の使用法 (89 ページ)

WIDTH_BUCKET 関数 [数値]

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

パラメータ	説明
<i>expression</i>	ヒストグラムが作成されている式。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る。 <i>expr</i> が null に評価されると、式は null を返す。
<i>min_value</i>	<i>expr</i> 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、null には評価できない。

パラメータ	説明
max_value	<i>expr</i> 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、nullには評価できない。
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る。

備考

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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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、オーバーフローバケット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。たとえば、**WIDTH_BUCKET** (*credit_limit*, 5000, 0, 5) では、バケット番号 1 は (4000, 5000)、バケット番号 2 は (3000, 4000)、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。

YEAR 関数 [日付と時刻]

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

パラメータ	説明
datetime-expression	日時。

戻り値

SMALLINT

備考

YEAR 関数は **YEARS** 関数の最初の構文と同じです。

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文は、値 1998 を返します。

```
SELECT YEAR( '1998-07-13 06:07:12' ) FROM iq_dummy
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)

- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEARS 関数 [日付と時刻] (386 ページ)
- NTILE 関数 [統計] (289 ページ)

YEARS 関数 [日付と時刻]

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

パラメータ	説明
<i>datetime-expression</i>	日時。
<i>integer-expression</i>	<i>datetime-expression</i> に追加する年数。 <i>integer-expression</i> が負の場合、日時の値から適切な年数が引かれる。整数式を指定する場合は、 <i>datetime-expression</i> を DATETIME データ型として明示的にキャストする必要がある。

戻り値

構文 1 は、INTEGER を返します。

構文 2 は、TIMESTAMP を返します。

備考

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 (**SELECT YEARS (CAST ('1992-02-29' AS TIMESTAMP), 1)** など)、結果は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

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

例

次の文を実行すると、値 1998 が返されます。

```
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
```

参照：

- CAST 関数 [データ型変換] (168 ページ)
- CONVERT 関数 [データ型変換] (178 ページ)
- HOURS 関数 [日付と時刻] (236 ページ)
- MINUTES 関数 [日付と時刻] (277 ページ)
- MONTHS 関数 [日付と時刻] (281 ページ)
- REPLACE 関数 [文字列] (325 ページ)
- SECOND 関数 [日付と時刻] (336 ページ)
- WEEKS 関数 [日付と時刻] (380 ページ)
- YEAR 関数 [日付と時刻] (385 ページ)

YMD 関数 [日付と時刻]

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

パラメータ	説明
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る。
integer-expression3	日の番号。任意の整数値を日として指定できる。指定の値に応じて、日付が調整される。

戻り値

DATE

標準と互換性

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – Adaptive Server Enterprise ではサポートされていません。

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
```

SQL データ型

SQL データ型によって、格納するデータのタイプ (文字列、数値、日付など) を定義します。

文字データ型

文字データ型は、文字、数字、記号などの文字列を格納するために使用します。

構文

文字データ型には次の構文を使用します。

CHAR [(*max-length*)]

CHARACTER [(*max-length*)]

CHARACTER VARYING [(*max-length*)]

VARCHAR [(*max-length*)]

UNIQUEIDENTIFIERSTR

使用法

次の表は、文字データ型の説明です。

表 54 : 文字データ型

文字データ型	説明
CHAR	<p>最大長 <i>max-length</i> バイトの文字データ。 <i>max-length</i> を指定し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。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」を参照。</p> <p>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。</p> <p>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、<i>max-length</i> までブランクが埋め込まれる。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る。</p>
CHARACTER	CHAR と同じ。

文字データ型	説明
CHARACTER VARYING	VARCHAR と同じ。
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) により制限される。
TEXT	ユーザ定義データ型。NULL を許可する LONG VARCHAR として実装される。
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。VARCHAR 文字列に指定可能な最大長は (32KB - 1)。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」を参照。
UNIQUEIDENTIFIERSTR	CHAR (36) として実装されるドメイン。このデータ型は、Microsoft SQL Server の uniqueidentifier カラムをマッピングするとき、リモートデータアクセスに使用される。

注意： 別途ライセンスが必要ですが、SAP Sybase IQ オプションがサポートするキャラクターラージオブジェクト (CLOB) データを使用すると、SAP Sybase IQ のページサイズが 128KB であれば 0 ~ 512TB (テラバイト)、また SAP Sybase IQ のページサイズが 512KB であれば 0 ~ 2PB (ペタバイト) の長さ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。『非構造化データ分析』を参照してください。

参照：

- バイナリデータ型 (398 ページ)
- NEWID 関数 [その他] (282 ページ)
- STRTOUUID 関数 [文字列] (360 ページ)
- UUIDTOSTR 関数 [文字列] (374 ページ)
- バイナリデータ型 (874 ページ)
- 文字データ型の互換性 (873 ページ)

記憶領域サイズ

文字データの記憶領域サイズ、所定のカラム定義サイズと入力データサイズは次のとおりです。

表 55 : 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

データ型	カラム定義	入力データ	記憶領域
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用されるため、これらの文字はウィンドウ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

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、データが引用符で囲まれているかどうかに応じて異なる方法で処理されます。

INSERT、**UPDATE**、**LOAD TABLE** によって挿入されるデータは、次のいずれかです。

- 引用符で囲まれている
- 引用符で囲まれていない
- バイナリ

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/OFFLOAD** 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 // trailing blanks trimmed LOAD TABLE t( c1
',' ) ..... STRIP OFF // trailing blanks not trimmed
LOAD TABLE t( c1 ASCII(3) ) ... STRIP RTRIM // trailing
blanks not trimmed LOAD TABLE t( c1 ASCII(3) ) ... STRIP
OFF // trailing blanks trimmed LOAD TABLE t( c1
BINARY ) ..... STRIP RTRIM // trailing blanks trimmed LOAD
TABLE t( c1 BINARY ) ..... STRIP OFF // trailing blanks
trimmed
```

- 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 カラムについては、デフォルトインデックス、**WD** インデックス、**TEXT** インデックス、**COMP** インデックスのみがサポートされます。

これらのカラムに対して、**LF**、**HG**、**HNG**、**DATE**、**TIME**、**DTTM** インデックスは作成できません。

文字データの互換性

文字データの互換性は、SAP Sybase IQ、Adaptive Server、SQL Anywhere の間で異なります。

- CHARACTER(*n*)を CHAR の代わりに使用することは、Adaptive Server ではサポートされていません。
- SAP Sybase IQ は、Adaptive Server が提供しているデータ型 NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR をサポートしていません。SAP Sybase IQ は、CHAR データ型と VARCHAR データ型で Unicode をサポートしています。
- SAP Sybase IQ は、SQL Anywhere よりも長い LONG VARCHAR データ型をサポートしています。『非構造化データ分析』を参照してください。
- SAP Sybase IQ と Adaptive Server 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

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。

SQL Anywhere は、CHAR、VARCHAR、LONG VARCHAR のカラムをすべて同じ型として扱います。

255 番目の文字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。該当するのは、**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

数値データ型は、数値データを格納するために使用します。

構文

数値データ型には次の構文を使用します。

```
[ UNSIGNED ] BIGINT
```

```
[ UNSIGNED ] { INT | INTEGER }
```

```
SMALLINT
```

```
TINYINT
```

```
DECIMAL [ ( precision [ , scale ] ) ]
```

```
NUMERIC [ ( precision [ , scale ] ) ]
```

```
DOUBLE
```

```
FLOAT [ ( precision ) ]
```

```
REAL
```

数値データ型の使用法

数値データ型を使用する場合は、次の点に注意してください。

- 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。
- 小数セパレータ (小数点) はピリオドだけです。カンマは小数セパレータとしてサポートされていません。

表 56 : 数値データ型

数値データ型	説明
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。 整数を UNSIGNED として指定できる。デフォルトでは、このデータ型は符号付きである。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) である。
INT または INTEGER	-2147483648 から 2147483647 の間の値を取る符号付き 32 ビット整数。4 バイトの記憶領域が必要。 INTEGER データ型は真数値データ型である。精度は算術演算の後で保存される。 整数を UNSIGNED として指定できる。デフォルトでは、このデータ型は符号付きである。符号なし整数値の範囲は、0 から 4294967295 の間。
SMALLINT	-32768 から 32767 の間の値を取る符号付き 16 ビット整数。2 バイトの記憶領域が必要。 SMALLINT データ型は真数値データ型である。精度は算術演算の後で保存される。
TINYINT	0 から 255 の間の値を取る符号なし 8 ビット整数。1 バイトの記憶領域が必要。 TINYINT データ型は真数値データ型である。精度は算術演算の後で保存される。
DECIMAL	総桁数の <i>precision</i> と小数点以下の桁数の <i>scale</i> を持つ符号付き 10 進数。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る。デフォルトでは、位取りが 38 で精度が 126。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るが、MAX_CLIPRENT_NUMERIC_SCALE オプションを使用すると、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ができる。
NUMERIC	DECIMAL と同じ。
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。ゼロでない絶対値の範囲は、2.2250738585072014e-308 ~ 1.797693134862315708e+308。 DOUBLE として保持される値の有効桁数は厳密には 15 桁であるが、15 桁を超えると丸め誤差が出る可能性がある。 DOUBLE データ型は概数値データ型である。算術演算後に丸め誤差が出る。

数値データ型	説明
FLOAT	<p><i>precision</i> を指定しない場合、FLOAT データ型は REAL データ型と同じになる。<i>precision</i> を指定した場合、FLOAT データ型は、精度の値に応じて、REAL データ型または DOUBLE データ型と同じになる。REAL または DOUBLE のどちらのデータ型と同じになるかは、プラットフォームによって異なる。その境界となるのは、プラットフォームの単精度浮動小数点数の小数部で使用されているビット数である。</p> <p>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る。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ない。</p> <p>FLOAT データ型は概数値データ型である。算術演算後に丸め誤差が出る。</p>
REAL	<p>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 ~ 3.402823466e+38。REAL として保持される値の有効桁数は厳密には 6 桁であるが、6 桁を超えると丸め誤差が出る可能性がある。</p> <p>REAL データ型は概数値データ型である。算術演算後に丸め誤差が出る。</p>

次の表は、10 進数に必要な記憶領域を示します。

表 57 : 10 進数に必要な記憶領域

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

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

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

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。次に例を示します。

```
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、算術演算の後、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]` で定義された桁数だけ 9 が並び、次に小数点、その後に `scale` で定義された桁数だけ 9 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]` の数だけゼロが続き、最後に 1 が 1 つ置かれます。次に例を示します。

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

NULL から NUMERIC への明示的な変換に対して、精度も位取りも指定しない場合は、デフォルトで NUMERIC(1,0) が設定されます。たとえば、

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

は、次のように記述されます。

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

注意： SQL Anywhere では、数値関数に対してサポートされる最大値は 255 です。数値関数の精度が SQL Anywhere でサポートされる最大値を超えると、次のようなエラーが発生します。"The result datatype for function 'funcname' exceeds the maximum supported numeric precision of 255. Please set the proper value for precision in numeric function, 'location'"

数値データの互換性

数値データの互換性は、SAP Sybase IQ、Adaptive Server、SQL Anywhere の間で異なります。

- Embedded SQL の TINYINT カラムは、2 バイトまたは 4 バイト整数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 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 (バージョン 15.x は、符号なし整数データ型をサポートします)。SAP Sybase IQ の符号なし整数は、Adaptive Server 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。マップすると、データが自動的に変換されます。
 - SAP Sybase IQ の UNSIGNED SMALLINT データは ASE INT にマップします。
 - 負の値がある場合は、SAP Sybase IQ の UNSIGNED BIGINT を 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、SAP 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。

SQL データ型

- NUMERIC と DECIMAL 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を避けてください。以下にその違いを示します。

Database	デフォルトの精度	デフォルトの位取り
SAP Sybase IQ	126	38
Adaptive Server	18	0
SQL Anywhere	30	6

- FLOAT (p) データ型は、REAL または DOUBLE と同じです (p によって決まります)。Adaptive Server 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。SAP 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。
- SAP 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) として実装されています。これらは主に Adaptive Server との互換性を保つために用意されています。

インデックス

この項では、インデックスタイプと数値データ型の関係について説明します。

- CMP** と **HNG** インデックスタイプは、FLOAT、DOUBLE、REAL データ型をサポートしていません。また、**HG** インデックスタイプは使用しないことをおすすめます。
- WD**、**DATE**、**TIME**、**DTTM** インデックスタイプは、数値データ型をサポートしていません。

バイナリデータ型

バイナリデータ型は、ピクチャなどのローバイナリデータを格納するために使用します。16 進に似た表記で、最大 (32K - 1) バイトまで格納できます。

構文

```
BINARY [ ( length ) ]
```

```
VARBINARY [ ( max-length ) ]
```

```
UNIQUEIDENTIFIER
```

参照：

- NEWID 関数 [その他] (282 ページ)

- STRTOUUID 関数 [文字列] (360 ページ)
- UUIDTOSTR 関数 [文字列] (374 ページ)
- 文字データ型 (389 ページ)
- バイナリデータ型 (874 ページ)

バイナリデータ型の使用法

バイナリデータは文字 "0x" または "0X" から始まり、数字と A ~ F (大文字および小文字) 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単位で指定するか、デフォルトの長さ 1 バイトを使用することができます。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 カラムの長さに対して、常に偶数の文字数を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、SAP 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。

バイナリデータ型	説明
BINARY	長さが <i>length</i> バイトのバイナリデータ。 <i>length</i> を指定しない場合は、デフォルトで 1 バイトが設定される。指定可能な最大サイズは 32767 バイト。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 を使用する。 BINARY カラムのエントリには、 <i>length</i> カラム長まで 0 が埋め込まれるので、VARBINARY カラム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る。
VARBINARY	最大長 <i>max-length</i> バイトのバイナリデータ。 <i>max-length</i> を指定しない場合は、デフォルトで 1 バイトが設定される。指定可能な最大サイズは (32K - 1) バイト。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 を使用する。
UNIQUEIDENTIFIER	UNIQUEIDENTIFIER データ型は、UUID (GUID と呼ばれる) 値の記憶領域として使用される。

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,
```

SQL データ型

```

        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 ('002710000000aeb',
        '002710000000aeb', '002710000000aeb',
        '002710000000aeb');
SELECT * FROM zeros;

```

bnot	bnull	vbnot	vbnull
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、SAP 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、SAP 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	vbnull
0x0000000000	0x0000000000	0x00	0x00

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

```

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

```

col_bin
0x3030323731303030

注意：上記の例では、string_rtruncation オプションを "off" に設定しておいてください。

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。次に例を示します。

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

または

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

フラットファイルの ASCII データ

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、必ずニブルで格納されます。

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、SAP 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

バイナリデータの記憶領域サイズをよく理解してください。

表 58 : 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 (32K - 1) バイト	(32K - 1) バイトのバイナリ	(32K - 1) バイト
BINARY	幅 (32K - 1) バイト	(64K - 2) バイトの ASCII	(32K - 1) バイト

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変換において、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。

参照：

- データ型変換関数 (94 ページ)
- データ型変換 (415 ページ)

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 データや VARBINARY データを含むカラムには、次のような制限があります。

- 集合関数 **SUM**、**AVG**、**STDDEV**、**VARIANCE** では、バイナリデータ型を使用できません。集合関数 **MIN**、**MAX**、**COUNT** は、バイナリデータ型 BINARY と VARBINARY をサポートしています。
- **HNG**、**WD**、**DATE**、**TIME**、**DTTM** インデックスは、BINARY データや VARBINARY データをサポートしていません。
- サイズが 255 バイトより大きい BINARY および VARBINARY データの場合は、デフォルト、**CMP**、**TEXT** 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。

バイナリデータの互換性

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

表 59 : 後続ゼロの扱い

データ型	SAP Sybase IQ	SQL Anywhere	Adaptive Server
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、SQL Anywhere、SAP 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ています。このオプションは、**INSERT** 文字列または **UPDATE** 文字列がトランケートされたときのエラーメッセージの出力に

影響を与えます。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設定してください。

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。

Adaptive Server は、バイナリ型データに対するビット操作をサポートしていません。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ています。SAP 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ています。

UNIQUEIDENTIFIER

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。

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、次の文は、テーブル mytab を更新し、カラム uid_col の現在値が NULL の場合は、その値に **NEWID** 関数で生成された一意の識別子を設定します。

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

次の文を実行します。

```
SELECT NEWID()
```

一意の識別子が 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 **UIDTOSTR()** 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 関数と **UIDTOSTR** 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。

SQL データ型

UNIQUEIDENTIFIER 値は大きいので、データベース間に一意の識別子が必要な場合は、UNIQUEIDENTIFIER の代わりに、UNSIGNED BIGINT または UNSIGNED INT identity カラムを使用する方がより効率的です。

UNIQUEIDENTIFIER の標準と互換性

UNIQUEIDENTIFIER の値に関する標準と互換性は、次のとおりです。

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
- Sybase – SQL Anywhere でサポートされています。Adaptive Server ではサポートされていません。

バイナリラージオブジェクトデータ

別途ライセンスが必要ですが、SAP 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) データを使用すると、ページサイズが 128KB であれば 0～512TB (テラバイト)、ページサイズが 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。『非構造化データ分析』を参照してください。

Bit データ型

BIT データ型は、ブール値の格納に使用します。

データ型	値	サポート
BIT	0 または 1	SQL Anywhere と Adaptive Server

使用法

BIT は 0 または 1 のみ格納します。

BIT カラムにゼロ以外の値を挿入すると、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 データに対しては、デフォルトのインデックスタイプだけがサポートされています。

Bit データの互換性

Adaptive Server 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

日付と時刻のデータ型は、日付と時刻の格納に使用されます。

構文

日付と時刻のデータ型には次の構文を使用します。

DATE

DATETIME

SMALLDATETIME

TIME

TIMESTAMP

参照：

- **TIMESTAMP 特別値** (72 ページ)
- **CURRENT_TIMESTAMP 特別値** (70 ページ)
- **CURRENT TIME 特別値** (70 ページ)
- **CURRENT DATE 特別値** (69 ページ)
- **日付と時刻の取得** (408 ページ)
- **BIGTIME と BIGDATETIME のサポート** (876 ページ)

日付と時刻のデータ型の使用法

日付と時刻のデータ型は、次の使用上の考慮事項をよく理解してから使用してください。

日付と時刻のデータ型	説明
DATE	年、月、日などの暦日。年の範囲は 0001 ~ 9999。日がゼロの値を持つことはできない。したがって、最小値は 0001-01-01。DATE 値には 4 バイトの記憶領域が必要。
DATETIME	TIMESTAMP として実装されたドメイン。DATETIME は主に Adaptive Server との互換性を保つために用意されている。
SMALLDATETIME	TIMESTAMP として実装されたドメイン。SMALLDATETIME は主に Adaptive Server との互換性を保つために用意されている。

日付と時刻のデータ型	説明
時刻	時、分、秒、秒以下で構成される時間。秒以下は 6 桁まで格納される。TIME 値には 8 バイトの記憶領域が必要。(ODBC 規格では、TIME データ型の精度を秒単位までに制限している。そのため、 WHERE 句の比較に、秒単位よりも高い精度に基づく TIME データ型を使用しない。)
TIMESTAMP	年、月、日、時、分、秒、秒以下で構成される時刻。秒以下は 6 桁まで格納される。日にはゼロでない値を格納する必要がある。TIMESTAMP 値には 8 バイトの記憶領域が必要。

TIMESTAMP データ型の有効な値の範囲は、0001-01-01 00:00:00.000000 から 9999-12-31 23:59:59.999999 です。1600-02-28 23:59:59 から 7911-01-01 00:00:00 の範囲外の TIMESTAMP データは表示が不完全になりますが、データベースには正確な日時値が格納されます。最初に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。それを行うには、**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'
```

参照：

- 文字列から日付／時刻への変換の互換性 (416 ページ)

サポートされているインデックスタイプ

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ています。

- 日付と時刻のデータ型はすべて、**CMP**、**HG**、**HNG**、**LF** インデックスタイプをサポートしています。**WD** インデックスタイプはサポートされていません。
- DATE データは、**DATE** インデックスをサポートしています。
- TIME データは、**TIME** インデックスをサポートしています。
- DATETIME と TIMESTAMP データは、**DTTM** インデックスをサポートしています。

日付と時刻の送信

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うときは、文字列として
- ODBC を使用し、TIMESTAMP 構造体として
- Embedded SQL を使用し、SQLDATETIME 構造体として

データベースに時刻を文字列として (TIME データ型の場合)、または文字列の一部として (TIMESTAMP または DATE データ型の場合) 送信する場合は、時間、分、秒をコロンで区切り、*hh:mm:ss.sss* のフォーマットにする必要がありますが、文字列中のどこに置いてかまいません。オプションとして、*hh:mm:ss.sss* のように秒と小数点以下の秒をピリオドで区切ることができます。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 -- 24 hour clock if no am or pm specified
10:00pm -- pm specified, so interpreted as 12 hour clock
10:00 -- 10:00am in the absence of pm
10:23:32.234 -- seconds and fractions of a
                second included
```

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。

- データベースで確実に解釈される *yyyy/mm/dd* または *yyyy-mm-dd* フォーマットの文字列として
- 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。

日付と時刻の取得

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うときは、文字列として
- ODBC を使用し、TIMESTAMP 構造体として
- Embedded SQL を使用し、SQLDATETIME 構造体として

使用法

日付や時刻を文字列として取得する場合は、データベースオプション `DATE_FORMAT`、`TIME_FORMAT`、`TIMESTAMP_FORMAT` によって指定されているフォーマットを使用します。

日付については次の演算子が許可されています。

表 60 : 演算子

演算子	説明
<code>timestamp + integer</code>	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る。
<code>timestamp - integer</code>	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く。
<code>date - date</code>	2つの日付(タイムスタンプ)間の日数を計算する。
<code>date + time</code>	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する。

参照：

- `TIMESTAMP` 特別値 (72 ページ)
- `CURRENT_TIMESTAMP` 特別値 (70 ページ)
- `CURRENT_TIME` 特別値 (70 ページ)
- `CURRENT_DATE` 特別値 (69 ページ)
- 日付と時刻のデータ型 (405 ページ)

日付と時刻の比較

日付を文字列として比較する場合は、`DATEFORMAT` 関数または `CAST` 関数によって日付を文字列に変換してから比較します。

使用法

```
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;
```

あいまいさのない日付と時刻

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。

使用法

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、常に日付として認識されます。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 設定が異なる可能性がある場合は、必ずこのフォーマットを使用してください。たとえば、ストアー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。

他のコンテキストでは、より柔軟性のある日付フォーマットを使用できます。SAP Sybase IQ は、さまざまな文字列をフォーマットとして解釈できます。この解釈は、DATE_ORDER データベースオプションに左右されます。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。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、接続されるときに必ず 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。

SQL データ型

データベースオプション `DATE_ORDER` は、文字列 `10/11/12` が、データベースで 1912 年 10 月 11 日、1910 年 11 月 12 日、または 1912 年 11 月 10 日のいずれとして解釈されるかを決定します。日付文字列の年、月、日を記号 (/、-、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 [TSQL] 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 は 1950、49 は 2049 と解釈されます。

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。

4 桁のフォーマットを使用して年を指定します。

`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 (月のフィールドで役に立ちます。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

ドメインは、必要に応じて精度と小数点以下の桁数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と、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。

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 ISO/ANSI SQL 標準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。

`CREATE DATATYPE` システム権限が必要です。データ型を一度作成すると、**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、`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できます。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のみ、この文を実行できます。

ユーザ定義データ型に関する制約とデフォルト

`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、デモ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 変数では、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。

CREATE DOMAIN 文

データベースにユーザ定義データ型を作成します。

クイックリンク：

「パラメータ」 (413 ページ)

「例」 (413 ページ)

「使用法」 (413 ページ)

「標準」 (414 ページ)

「パーミッション」 (414 ページ)

構文

```
CREATE { DOMAIN | DATATYPE } domain-name data-type
... [ NOT ] NULL ]
... [ DEFAULT default-value ]
```

default-value - (構文に戻る)

special-value

```
| string
| global variable
| [ - ] number
| ( constant-expression )
| built-in-function( constant-expression )
| AUTOINCREMENT
| CURRENT DATABASE
| CURRENT REMOTE USER
| NULL
| TIMESTAMP
| LAST USER
```

special-value (back to default-value)

CURRENT

```
{ DATE
  | TIME
  | TIMESTAMP
  | USER
  | PUBLISHER }
| USER
```

パラメータ

(先頭に戻る) (412 ページ)

- **data-type** – 精度と位取りを指定した組み込みデータ型

例

(先頭に戻る) (412 ページ)

- **例 1** – 35 文字の文字列を格納し、NULL が使用できる `address` という名前のデータ型を作成します。

```
CREATE DOMAIN address CHAR( 35 ) NULL
```

使用法

(先頭に戻る) (412 ページ)

ユーザ定義データ型は、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。データベース内の使いやすさを改善し、一貫性を高めます。

注意： `CREATE DOMAIN` は ANSI/ISO SQL3 用語なので、`CREATE DATATYPE` ではなく、`CREATE DOMAIN` を使用してください。

データ型を作成するユーザは、自動的にそのデータ型の所有者となります。`CREATE DATATYPE` 文の中では、所有者を指定できません。ユーザ定義データ型の名前はユニークでなければなりません。また、プレフィクスとして所有者を使用しなくても、すべてのユーザがデータ型にアクセスできます。

ユーザ定義データ型は、データベースの中のオブジェクトです。識別子のルールに従って名前を付ける必要があります。ユーザ定義データ型の名前では、組み込みデータ型の名前と同様に、常に大文字と小文字が区別されません。

ユーザ定義データ型では、デフォルトで NULL を使用できます。ただし、**allow_nulls_by_default** データベースオプションが OFF に設定されている場合は除きます。この場合、新しいユーザ定義データ型では、デフォルトで NULL を使用できません。ユーザ定義データ型で作成したカラムで NULL を使用できるかどうかは、そのカラムを参照するときの **allow_nulls_by_default** オプションの設定ではなく、ユーザ定義データ型の定義の設定によって決まります。カラム定義で NULL または NOT NULL を明示的に設定すると、ユーザ定義データ型の設定より優先されます。

`CREATE DOMAIN` 文を使用すると、ユーザ定義データ型の DEFAULT 値を指定できます。DEFAULT 値の指定は、そのデータ型で定義されたすべてのカラムに継承されます。カラムに対して明示的に指定した DEFAULT 値は、データ型に対して指定したものより優先されます。

CREATE DOMAIN 文を使用して、CHECK 条件と呼ばれるルールをユーザ定義データ型の定義に組み込むことができます。

SAP 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、およびユーザ定義データ型に CHECK 制約が適用されます。

データベースからデータ型を削除するには **DROP** 文を使用します。ユーザ定義データ型を削除するには、そのデータ型の所有者であるか、**CREATE DATATYPE** または **CREATE ANY OBJECT** システム権限を持っている必要があります。

関連する動作：

- オートコミット

標準

(先頭に戻る) (412 ページ)

- SQL - ISO/ANSI SQL 準拠。
- SAP Sybase Database 製品 - Adaptive Server ではサポートされていません。Transact-SQL は **sp_addtype** システムプロシージャ、**CREATE DEFAULT** 文、**CREATE RULE** 文を使用して、同様の機能を提供しています。

パーミッション

(先頭に戻る) (412 ページ)

次のいずれかが必要です。

- **CREATE DATATYPE** システム権限
- **CREATE ANY OBJECT** システム権限

ドメインの互換性

ドメインの互換性は、SAP Sybase IQ、Adaptive Server、SQL Anywhere の間で異なります。

- 名前付き制約と名前付きデフォルト – SAP Sybase IQ では、ユーザ定義データ型はベースデータ型で作成され、オプションとして NULL または NOT NULL 条件も使用できます。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。
- SAP Sybase IQ では、**sp_addtype** システムプロシージャまたは **CREATE DOMAIN** 文を使用してドメインを追加できます。Adaptive Server の場合は、**sp_addtype** を使用する必要があります。**sp_addtype**、および SQL Anywhere から継承した他のストアードプロシージャの所有者は **dbo** です。SQL Anywhere ストアドプロシージャを使用して作成されたオブジェクトの作成者も **dbo** です。そのため、**ALTER DATATYPE** または **ALTER ANY OBJECT** システ

ム権限、および DROP DATATYPE または DROP ANY OBJECT システム権限のないユーザは、sp_addtype を使用して作成されたドメインを変更または削除できません。dbo が所有するドメインの変更および削除には、これらのシステム権限が必要です。

データ型変換

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。

使用法

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 -- Tries to convert the string to a number  
'a' > 0      -- Tries to convert 'a' to a number
```

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、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) に類似していますが、string(NULL) が空の文字列 ("") であるのに対して、CAST(NULL AS CHAR) は NULL 値です。

注意：TINYINT、SMALLINT、INT、DOUBLE、DATE、DATETIME、TIME、BIGINT、UNSIGNED BIGINT、UNSIGNED INT から CHAR または VARCHAR への変換時に問題が発生すると、SAP Sybase IQ はその問題をエラーまたは警告として .iqmsg ファイルに記録します。CONVERSION_ERROR オプションが ON の場合、SAP Sybase IQ は問題をエラーとして報告します。CONVERSION_ERROR オプションが OFF の場合、問題は警告として報告されます。

参照：

- データ型変換関数 (94 ページ)
- 記憶領域サイズ (401 ページ)

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や時刻データ型に変更する場合、SAP Sybase IQ と Adaptive Server の動作には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、SAP Sybase IQ と Adaptive Server 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。

使用法

時刻のミリ秒の部分が 3 桁より小さい場合、Adaptive Server 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 秒、2 桁であれば 100 分の 1 秒、3 桁であれば 1000 分の 1 秒を意味します。SAP 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。

- Adaptive Server 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
```

- SAP Sybase IQ は、どちらの場合も、Adaptive Server 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
```

参照：

- 日付と時刻のデータ型の使用法 (405 ページ)

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 は1桁目のブランクをサポートし、SAP Sybase IQ はゼロまたはブランクをサポートします。

Adaptive Server でサポートされるデータ型とサポートされないデータ型の詳細については、『管理：ロード管理』を参照してください。

BIT から BINARY へのデータ型の変換

SAP Sybase IQ は、BIT から BINARY、および BIT から VARBINARY への暗黙的および明示的な変換をサポートしています。これらの変換は、Adaptive Server でサポートされている変換と互換性があります。

SAP Sybase IQ は、比較演算子、算術演算子、INSERT 文と UPDATE 文に対して、BIT から BINARY へ、および BIT から VARBINARY へと暗黙的にデータ型を変換します。

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、2ⁿ 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。

ビット値 '1' を下記に変換	結果
BINARY(3)	0x010000
VARBINARY(3)	0x01
BINARY(8)	0x0100000000000000
VARBINARY(8)	0x01

以下の例は、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。

次のようなテーブルとデータがあるとします。

SQL データ型

```
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
---
0x010000000000000000000000 (18 nibbles)
0x000000000000000000000000 (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
---
0x010000000000000000000000 (18 nibbles)
0x000000000000000000000000 (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 間のデータ型変換

SAP Sybase IQ は、比較演算子、算術演算子、**INSERT** 文と **UPDATE** 文に対して、BIT から CHAR へ、および BIT から VARCHAR への暗黙的なデータ型の変換をサポートしています。

以下の例は、BIT データ型と CHAR データ型、および BIT データ型と VARCHAR データ型との間の暗黙的および明示的な変換を示しています。

次のような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 または VARCHAR から BIT への暗黙的な変換、および 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 または CHAR から BIT への明示的な変換、および 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 または VARCHAR から BIT への明示的な変換、および 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)
```

SQL データ型

```
-----  
\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
```

他の SQL 言語との違い

SAP 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。

SAP 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

SAP 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。

表 61 : 日付の演算

日付の演算	説明
date + integer	指定された値の日数を日付に加える。
date - integer	指定された値の日数を日付から引く。
date - date	2つの日付間の日数を計算する。
date + time	日付と時刻からタイムスタンプを作成する。

日付と時刻の処理に使用できる関数は、数多くあります。

整合性

SAP 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** 文および **ALTER TABLE** 文に対する次の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 ]
```

他の SQL 言語との違い

PRIMARY KEY 句では、関係のプライマリキーを宣言します。SAP Sybase 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 句は、このテーブルと他のテーブルの関係を定義します。その関係は、このテーブルのカラムが他のテーブルのプライマリキーの値を保有することによって確立しています。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 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。

ジョイン

SAP Sybase IQ では、テーブル間の **automatic joins** を使用できます。

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、SAP Sybase IQ では外部キー関係に基づくテーブル間の **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

SAP Sybase IQ では、複数のテーブルを **UPDATE** 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 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。

```
DELETE column  
RENAME new-table-name  
RENAME old-column TO new-column
```

ヒント： カラムの作成後にカラムのデータ型を変更することはできません。データ型を変更するには、カラムを削除し、適切なデータ型でカラムを再作成してください。

サブクエリが許容されない場合

SQL Anywhere とは異なり、SAP Sybase IQ では、式を使用できる場所であればどこでもサブクエリを使用できるわけではありません。

SAP 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。SAP Sybase IQ は SQL Anywhere 拡張をサポートしていません。

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 SAP 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

SAP 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。

参照：

- SQL 関数 (85 ページ)

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。

他の SQL 言語との違い

物理的制限

SAP Sybase IQ データベースでは、オブジェクトのサイズと数に制限があります。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。

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。

表 62 : データベースオブジェクトのサイズと数の制限

項目	制限事項
カタログファイルサイズ	すべてのプラットフォームで上限は 1TB。NTFS を使用する Windows システムは、最大 1TB をサポートする。
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。 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。
DB ファイルのサイズ	ファイルシステムデバイス：最大サイズは 4TB。 オペレーティングシステムファイル：オペレーティングシステムでサポートされる最大サイズ。
DB 領域のサイズ	DB 領域ごとに最大 2000 dbfile。 NAS (ネットワーク接続ストレージ) デバイス上に DB 領域を作成することは推奨しない。
フィールドサイズ	BINARY では 255 バイト、VARBINARY では 32,767 バイト。 CHAR、VARCHAR では 32,767 バイト。 LONG BINARY、LONG VARCHAR では、128KB ページの場合は最大 512TB、512KB ページの場合は 1PB。
IQ ページサイズ	64 ~ 512KB 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。複数カラムのインデックスでは 5300 バイト。
文字列リテラルの最大長	32KB

物理的制限

項目	制限事項
SQL 文の最大長	<p>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までに制限される。</p> <p>SQL 文が長すぎる場合は、-gss を使用してカタログスタックのサイズを大きくし、-c、または -ch と -cl 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す。</p> <p>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る。-gp 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るが、通常はデフォルトの -gp 4096 を使用することを推奨する。</p>
可変長 FILLER カラムの最大長	512 バイト
DB ファイルの最大数	開けるファイルの総数は、オペレーティングシステムがサポートできるユニークファイル記述子の数によって異なる。
(同時接続中の) ユーザの最大数	<p>64 ビットプラットフォーム (AIX、HP、Linux、Sun Solaris) では 1000。</p> <p>Windows の 64 ビットプラットフォームでは 200。</p>
一時抽出ファイルの最大サイズ	<p>TEMP_EXTRACT_SIZEn オプションで設定される。プラットフォーム別の制限は次のとおり。</p> <p>AIX、HP-UX : 0 ~ 64GB</p> <p>Sun Solaris : 0 ~ 512GB</p> <p>Windows : 0 ~ 128GB</p> <p>Linux : 0 ~ 512GB</p>
Table_name サイズ	128 バイト
カラム数/テーブル	SAP 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る。NULL を許可するカラム数のテーブルごとの制限は、約 $8 * (\text{database-page-size} - 30)$ 。
Column_name サイズ	128 バイト
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 NOFILE を使用するなどして調整可能な 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} (~4,000,000,000)/テーブル。
Index_name サイズ	128 バイト

項目	制限事項
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 $2^{48} - 1$ 。
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、クエリに応じて、16 ~ 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
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	SAP 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。

システムプロシージャ

システム情報を取得するには、SAP Sybase IQ データベースでシステム提供のストアプロシージャを使用します。

SAP 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プロシージャ。『管理：マルチプレックス』の「システムプロシージャ」を参照してください。
- Transact-SQL システムとカタログプロシージャ。

`sp_iqsetcompression`、`sp_iqshowcompression` など、特にラージオブジェクトデータに関連するシステムストアプロシージャについては、『非構造化データ分析』の「ストアプロシージャのサポート」を参照してください。

権限付きシステムプロシージャの実行管理

権限付きシステムプロシージャが動作できるセキュリティモデルは 2 つあります。各モデルでは、異なる方法でシステムプロシージャを実行できます。

注意： 次の情報は、SAP Sybase IQ の権限付きシステムプロシージャのみに適用されます。ユーザ定義のストアプロシージャには適用されません。

最初のモデルは、SYSTEM PROCEDURE DEFINER モデルと呼ばれます。このモデルでは、権限付きシステムプロシージャがその所有者 (通常、dbo) の権限で実行されます。2 番目のモデルは、SYSTEM PROCEDURE INVOKER モデルと呼ばれます。このモデルでは、権限付きシステムプロシージャがその実行者の権限で実行されます。

SYSTEM PROCEDURE DEFINER モデルを使用して権限付きシステムプロシージャを実行するには、そのプロシージャに対する明示的な EXECUTE オブジェクトレベル権限を付与します。そのシステムプロシージャの基本となる承認済みタスクの実行に必要なシステム権限は、所有者 (システムプロシージャ definer) から自動的に継承されます。

SYSTEM PROCEDURE INVOKER モデルを使用した権限付きシステムプロシージャについては、PUBLIC ロールに EXECUTE オブジェクトレベル権限が付与されます。デフォルトで、ユーザはいずれも PUBLIC ロールのメンバーであるため、すべてのユーザが EXECUTE 権限を自動的に継承します。しかし、PUBLIC ロール

はそのシステムプロシージャの所有者ではなく、またシステム権限は何も付与されていないため、基礎となる承認済みタスクの実行に必要なシステム権限をユーザに直接または間接的に付与する必要があります。

デフォルトでは、バージョン 16.0 以降で作成されたデータベースでは、SYSTEM PROCEDURE INVOKER モデルを使用して、すべての権限付きシステムプロシージャが実行されます。16.0 より前のバージョンで作成され、16.0 以降にアップグレードされたデータベースは、SYSTEM PROCEDURE DEFINER モデルと

SYSTEM PROCEDURE INVOKER モデルの両方の組み合わせを使用して、権限付きシステムプロシージャを実行します。この結合モデルでは、16.0 より前の権限付きシステムプロシージャはすべて、SYSTEM PROCEDURE DEFINER モデルを使用して実行され、16.0 (または以降のいずれかのリリース) で導入された権限付きシステムプロシージャは、SYSTEM PROCEDURE INVOKER モデルを使用して実行されます。デフォルトのセキュリティモデルは、データベースの作成時、アップグレード時、またはそれ以降に随時、上書きすることができます。ただし、カスタムのストアードプロシージャおよびアプリケーションの機能が失われることがあるため、これはおすすしめしません。

SYSTEM PROCEDURE DEFINER モデルを使用した権限付きシステムプロシージャの実行では、通常、DBO システムロールがプロシージャの所有者になります。デフォルトの場合、dbo システムロールは SYS_AUTH_DBA_ROLE 互換ロールに付与されます。これにより、システムプロシージャの実行に必要なすべての権限が間接的に付与されることが保証されます。SYS_AUTH_DBA_ROLE 互換ロールを移行すると、dbo システムロールの権限付きシステムプロシージャの実行能力が失われる場合があります。詳細については、「互換性ロールの移行がシステムロールに及ぼす影響」を参照してください。

互換ロールの移行の意味

一部のシステムロールには、互換ロールのメンバーシップを介して権限付きのタスクの実行に必要なシステム権限が間接的に付与されます。

互換ロールの基礎となるシステム権限を取り消すには、まずその権限をユーザ定義ロールに移行する必要があります。移行した場合のみ、新しいロールから基礎となるシステム権限を取り消し、組織のセキュリティ要件に従って他のユーザ定義ロールに付与することができます。これによって任務分散が実施されます。

互換ロールは、自動または手動で移行できます。移行の方法によっては、承認済みタスクをシステムロールで引き続き実行する能力に影響が生じることがあります。

使用した移行方法に関係なく、互換ロールが削除された後に、新しいユーザ定義ロールからシステム権限を取り消して別のユーザ定義ロールに付与する場合は、次のいずれかを実行して、システムロールが実行に必要なシステム権限をすべてを保持するには、該当する権限付きのタスクの

- システムロールに 移行されたユーザ定義ロールから取り消された各システム権限を付与
- システムロールにシステム権限が付与されているユーザ定義ロールのメンバーシップを付与。

互換ロールのメンバーで、移行の影響を受ける可能性のあるシステムロールは、次のとおりです。

システムロール	互換ロール
dbo	SYS_AUTH_DBA_ROLE SYS_AUTH_RESOURCE_ROLE
SYS_RUN_REPLICATION_ROLE	SYS_AUTH_DBA_ROLE

自動移行

ALTER ROLE 文は、新しいユーザ定義ロールを作成して、互換ロールの基礎となるシステム権限をすべて新しいユーザ定義ロールに自動的に付与し、互換ロールの各メンバーを新しいユーザ定義ロールのメンバーにしてから、互換ロールを削除します。

自動移行では、移行先ユーザ定義ロールが存在せず、すべてのシステム権限が同一の新しいユーザ定義ロールに移行されることを前提としています。

手動移行

新しいユーザ定義ロールを作成するには、**CREATE ROLE** 文を使用します。基礎となる各システム権限を1つ以上のユーザまたはロールに付与するには、**GRANT** 文を使用します。基礎となるシステム権限のすべてを1つ以上の他のユーザまたはロールに付与してから、互換ロールを削除するには、**DROP** 文を使用します。

移行された互換ロールのメンバーに対する新しいユーザ定義ロールのメンバーシップの自動付与は行われません。このため、互換ロールが削除されると、一部のシステムロールのメンバーは、想定されている権限付きのタスクを実行できなくなることがあります。影響を受けるシステムロールに新しいユーザ定義ロールのメンバーシップを付与するか、影響を受けるメンバーに必要なシステム権限を直接付与する必要があります。

権限付きシステムプロシージャを実行する機能の付与

権限付きシステムプロシージャを実行する機能を付与する際に使用するプロセスは、そのシステムプロシージャが実行されるセキュリティモデルによって異なります。

SYSTEM PROCEDURE DEFINER モデルを使用する権限付きシステムプロシージャの場合は、そのシステムプロシージャに対する EXECUTE オブジェクトレベル権限をユーザに付与します。

```
GRANT EXECUTE ON sys_procedure_name
TO grantee [,...]
```

SYSTEM PROCEDURE INVOKER モデルを使用する権限付きシステムプロシージャの場合は、そのシステムプロシージャに必要な基礎となるシステム権限をユーザに付与します。 **sp_proc_priv()** を使用して、システムプロシージャの実行に必要なシステム権限を指定します。

```
GRANT system_privilege_name
TO grantee [,...]
```

権限付きシステムプロシージャを実行する機能の取り消し

権限付きシステムプロシージャを実行する機能を取り消す際に使用するプロセスは、そのシステムプロシージャが実行されるセキュリティモデルによって異なります。

SYSTEM PROCEDURE DEFINER モデルを使用する権限付きシステムプロシージャの場合は、そのシステムプロシージャに対する EXECUTE オブジェクトレベル権限をユーザから取り消します。

```
REVOKE EXECUTE ON sys_procedure_name
FROM grantee [,...]
```

SYSTEM PROCEDURE INVOKER モデルを使用する権限付きシステムプロシージャの場合は、ユーザが持つそのシステムプロシージャに必要な基礎となるシステム権限を取り消します。

```
REVOKE system_privilege_name
FROM grantee [,...]
```

データベースが使用するセキュリティモデルの特定

データベースが使用できるセキュリティモデルは2つあります。

データベースが使用しているセキュリティモデルを特定するには、次の文を実行します。

```
select IF ((HEXTOINT(substring(db_property('Capabilities'),
1,length(db_property('Capabilities'))-20)) & 8) = 8)
THEN 1
ELSE 0
END IF
```

1は、データベースがSYSTEM PROCEDURE INVOKER モデルを使用していることを示します。0は、データベースが複合モデルを使用していることを示します。

複合モデルでは、16.0 より前の権限付きシステムプロシージャのみが SYSTEM PROCEDURE DEFINER を使用して実行されます。これらのシステムプロシージャを特定するには、16.0 より前の権限付きシステムプロシージャのリストを参照してください。

新しい、またはアップグレードされた 16.0 以降のデータベースを、SYSTEM PROCEDURE DEFINER モデルを使用してすべてのシステムプロシージャを実行するように設定することはできません。

16.0 より前の権限付きシステムプロシージャ

16.0 より前の権限付きシステムプロシージャのリストです。

複合セキュリティモデルを使用する権限付きシステムプロシージャ
これらの権限付きシステムプロシージャの場合、SYSTEM PROCEDURE DEFINER を使用するようにデータベースが設定されていると、実行するプロシージャに対する EXECUTE オブジェクトレベル権限のみが必要になります。データベースが SYSTEM PROCEDURE INVOKER を使用するように設定されている場合は、各プロシージャで必要とされる個別のシステム権限も必要です。各システムプロシージャの実行に必要なシステム権限については、『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 ガイド』を参照してください。

<ul style="list-style-type: none"> • sa_audit_string • sa_checkpoint_execute • sa_disable_auditing_type • sa_disk_free_space • sa_enable_auditing_type • sa_external_library_unload • sa_flush_cache • sa_list_external_library • sa_server_option • sa_procedure_profile • sa_procedure_profile_summary • sa_table_page_usage • sa_validate • sp_iq_reset_identity • sp_iqaddlogin • sp_iqbackupdetails • sp_iqbackupsummary • sp_iqcardinality_analysis • sp_iqcheckdb • sp_iqcheckoptions • sp_iqclient_lookup • sp_iqcolumn • sp_iqcolumnuse • sp_iqconnection • sp_iqconstraint • sp_iqcontext • sp_iqconstraint • sp_iqcontext • sp_iqcursorinfo • sp_iqdatatype • sp_iqdbsize 	<ul style="list-style-type: none"> • sp_iqdbspace • sp_iqdbspaceinfo • sp_iqdbspaceobjectinfo • sp_iqdbstatistics • sp_iqdroplogin • sp_iqemptyfile • sp_iquestdbspaces • sp_iquestspace • sp_iqevent • sp_iqfile • sp_iqhelp • sp_iqindex • sp_iqindex_alt • sp_iqindexadvice • sp_iqindexfragmentation • sp_iqindexinfo • sp_iqindexmetadata • sp_iqindexsize • sp_iqindexuse • sp_iqlmconfig • sp_iqlocks • sp_iqmodifyadmin • sp_iqmodifylogin • sp_iqmpxcheckdqpconfig • sp_iqmpxdumptlvlog • sp_iqmpxfilestatus • sp_iqmpxinconnpoolinfo • sp_iqmpxinheartbeatinfo • sp_iqcopyloginpolicy • sp_iqmpxinconnpoolinfo • sp_iqmpxinheartbeatinfo 	<ul style="list-style-type: none"> • sp_iqmpxinfo • sp_iqmpxversioninfo • sp_iqobjectinfo • sp_iqkeys • sp_iqprocedure • sp_iqprocparm • sp_iqrebuildindex • sp_iqrename • sp_iqrestoreaction • sp_iqrowdensity • sp_iqsetcompression • sp_iqsharedtempdistrib • sp_iqshowcompression • sp_iqshowpsex • sp_iqspaceinfo • sp_iqspaceused • sp_iqstatistics • sp_iqstatus • sp_iqsysmon • sp_iqtable • sp_iqtablesize • sp_iqtableuse • sp_iqtransaction • sp_iqunusedcolumn • sp_iqunusedindex • sp_iqunusedtable • sp_iqversionuse • sp_iqview • sp_iqwho • sp_iqworkmon
---	---	--

呼び出し側の権限を使用する権限付きシステムプロシージャ

これらの 16.0 より前の権限付きシステムプロシージャは、セキュリティモデルの設定に関係なく、そのプロシージャの所有者ではなくそのプロシージャを実行するユーザの権限で実行されます。したがって、システムプロシージャに対する EXECUTE オブジェクトレベル権限 (デフォルトでは PUBLIC ロールのメンバーシップによって付与) のほかに、そのシステムプロシージャに必要な追加のシステ

ム権限が付与されている必要があります。各システムプロシージャの実行に必要なシステム権限については、『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 ガイド』を参照してください。

- sa_describe_shapefile
- sa_get_user_status
- sa_locks
- sa_performance_diagnostics
- sa_report_deadlocks
- sa_text_index_stats

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 SAP Sybase IQ SQL と Transact-SQL のどちらの構文もサポートしているため、多少の変化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。

表 63 : 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>	SAP Sybase IQ	パラメータをカッコで囲むときは引用符が必要
<code>procedure_name 'param'</code>	SAP Sybase IQ	パラメータを引用符で囲むときは、カッコはオプション
<code>procedure_name param</code>	Transact-SQL	<p>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る。</p> <p>注意：所有者を指定する場合は、常にパラメータを引用符で囲む必要があります。たとえば、所有者が <i>dba</i> の場合、</p> <pre>sp_iqtablesiz e 'dba.empl'</pre> <p>はパラメータを引用符で囲む必要がありますが、</p> <pre>sp_iqtablesiz e empl</pre> <p>では不要です。</p>

構文	構文タイプ	説明
<code>procedure_name</code>	SAP Sybase IQ または Transact-SQL	Interactive SQL でパラメータなしにプロシージャを直接実行する場合は、この構文を使用する。プロシージャにパラメータはない。
<code>call procedure_name (param='value')</code>	SAP 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する。

Transact-SQL ストアドプロシージャを使うときは、Transact-SQL 構文を使用してください。

ストアドプロシージャが報告する統計情報を理解する

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、SAP Sybase IQ の状態に関する情報を報告します。

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 SAP Sybase IQ の利用状況を監視するには、SAP 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、指定した間隔で統計情報を収集し報告します。

SQL Anywhere でサポートされているプロシージャ

SAP Sybase IQ は SQL Anywhere のシステムプロシージャをサポートしています。

ヒント： SQL Anywhere のストアドプロシージャの名前には `iq` が含まれません。

`sa_get_table_definition` プロシージャは、SQL Anywhere テーブルでのみサポートされています。このプロシージャは、SAP Sybase IQ テーブルに対して実行すると、エラー `not implemented for IQ tables` を返します。

システムストアドプロシージャのアルファベット順リスト

システムプロシージャは、IQ メインストアでのシステム管理者タスクを実行します。

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。

注意： デフォルトでは、Interactive SQL Classic 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** などのストアー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、Interactive SQL 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。

sp_iqaddlogin プロシージャ

新しい SAP 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

```
call sp_iqaddlogin ('username_in', 'pwd' [,
'password_expiry_on_next_login'] [, 'policy_name'] )
```

構文 2

```
sp_iqaddlogin 'username_in', 'pwd' [, 'password_expiry_on_next_login']
[, 'policy_name']
```

構文 3

```
sp_iqaddlogin username_in, pwd [, password_expiry_on_next_login] [,
policy_name]
```

パラメータ

- **username_in** – 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。
- **pwd** – 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。
- **password_expiry_on_next_login** – (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 です (パスワードに有効期限はありません)。
- **policy_name** – (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。

sp_iqaddlogin を使って作成し、1 日で有効期限が切れるように設定した *username_in/pwd* 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、*n* 日で有効期限が切れるように設定した場合、日付が (*n* +1) 日目になると使用できなくなります。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。
MANAGE ANY USER システム権限

備考

新しい SAP 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。ユーザがすでにそのデータベースのユーザ ID を持っているが、ISYSUSER 内に登録されていない場合 (**GRANT CONNECT** 文または SAP Control Center によってユーザ 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。

SAP 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。

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。新しいユーザの作成には **CREATE USER** が使用されますが、下位互換性保持のため、**sp_iqaddlogin** も引き続きサポートされます。

例

この呼び出しでは、パスワード `irk324` を持つユーザ `rose` が `expired_password` というログインポリシーに追加されます。この例では、`expired_password` ログインポリシーがすでに存在しているものとします。

```
call sp_iqaddlogin('rose', 'irk324', 'ON', 'expired_password')
sp_iqaddlogin 'rose', 'irk324', 'ON', 'expired_password'
```

参照：

- `sp_expireallpasswords` システムプロシージャ (721 ページ)
- `sp_iqcopyloginpolicy` プロシージャ (469 ページ)
- `sp_iqmodifylogin` プロシージャ (535 ページ)
- `sp_iqpassword` プロシージャ (551 ページ)
- `sp_iqdroplogin` プロシージャ (490 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqbackupdetails プロシージャ

特定のバックアップに含まれるすべての `dbfile` 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： 次のクエリを実行すると、SYSIQBACKUPHISTORY テーブルから backup_id 値を取得できます。

```
select * from sysiqbackuphistory
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

sp_iqbackupdetails は、次の値を返します。

表 64 : 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"。
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。それ以外の場合は "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 "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)。

カラム名	説明
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(メガバイト)。
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 dbfile パス。それ以外の場合は "null"。

例

sp_iqbackupdetails の出力例を次に示します。

```

backup_id      backup_time      backup_type      selective_type  d
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 (783 ページ)

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqbackupsummary プロシージャ

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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

表 65 : 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 の出力例を次に示します。

```

backup_id      backup_time      backup_type      selective_type   v
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:YYYYtemp
YYYYb1'
    
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcardinality_analysis プロシージャ

テーブル内のカラムのカーディナリティを分析します。

注意： **sp_iqcardinality_analysis** は、インデックスタイプの値やインデックスの推奨を返さなくなりました。追加のカラムインデックスに関する推奨事項を参照する場合は、インデックスアドバイザの実行が推奨されます。

sp_iqcardinality_analysis は廃止され、今後のリリースで削除される予定です。

構文

```
sp_iqcardinality_analysis [ 'table_name' ] [, 'table_owner' ] [, 'script' ]
```

パラメータ

- **table_name** – テーブルの名前。
- **table_owner** – テーブル所有者名。このパラメータが指定されない場合、プロシージャは現在のユーザが所有するテーブルを検索します。
- **script** – スクリプト：
 - table_name
 - table_owner
 - column_name
 - cardinality
 - index_type
 - index recommendation

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。：

- SELECT ANY TABLE システム権限
- テーブルを所有している

さらに、次のシステム権限の1つ以上を所有する必要があります。

- CREATE ANY INDEX
- ALTER ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT

備考

パラメータをまったく指定しない場合、SAP Sybase IQ は現在のユーザが所有するすべてのテーブルのすべてのカラムについて **create_index** SQL 文を表示します。

script を指定すると、次のように出力をリダイレクトしてスクリプトファイルを生成できます。

```
OUTPUT TO 'indexfile.sql' FORMAT ASCII QUOTE '';
```

例

```
sp_iqcardinality_analysis 'Departments', 'GROUPO'
```

table_name	table_owner	column_name	cardinality	index type	Index Recommendation
Departments	GROUPO	DepartmentID	5		Run Index Advisor
Departments	GROUPO	DepartmentName	5		Run Index Advisor
Departments	GROUPO	DepartmentHeadID	5		Run Index Advisor

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcheckdb プロシージャ

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** では、分割されたデータがオフラインの DB 領域に存在する場合は、分割されたテーブルをチェックしません。

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、SAP Sybase 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 ]'

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}
  | cache main-cache-name
```

データベースの一貫性をチェックするモードは3種類、アロケーションマップをリセットするモードは1つ存在します。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、SAP Sybase IQ により次のエラーメッセージが返されます。

```
At least one mode and target must be specified to DBCC.
```

パラメータ

- **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、TEXT。

指定した *index-type* がターゲットに存在しない場合は、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、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、存在するインデックスタイプが **sp_iqcheckdb** によって処理されます。

- **index-name** –所有者とテーブル修飾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) がこの順序で代わりに使用されます。*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** –所有者の修飾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* パラメータに修飾子含まれません。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。テーブルに存在するパーティションが1つのみの場合、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、意味的に同じこととなります。

- **dbspace-name** – *dbspace-name* パラメータに修飾子含まれません。スペースが含まれる場合は、二重引用符で囲みます。

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なければなりません。テーブルに存在する DB 領域が1つのみの場合、DB 領域とデータベースターゲットは、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 より大きい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カの CPU 使用率を制限することができます。*resource-percent*=100 (デフォルト値) の場合、CPU ごとにスレッドが 1 つ作成されます。*resource-percent* > 100 の場合、CPU の数を超えるスレッドが作成され、マシンの構成によって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。
- **main-cache-name** – キャッシュターゲットにより、メインキャッシュ DB 領域のページが IQ メインストア内にある元のページと比較されます。

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。

適用対象

シンプレックスとマルチプレックス。

権限

ALTER DATABASE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

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。

モード	説明
Allocation	<p>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を使用して割り付けをチェックする。インデックスの一貫性は検査しない。</p> <p>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) または余剰なブロック (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) を検出する。</p> <p>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どのオブジェクトにも要求されていない割り付けブロック) を検出する。</p> <p>ターゲットが分割されたテーブルの場合、allocation mode は次のように処理する。</p> <ul style="list-style-type: none"> • 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をチェック • テーブル割り付けビットマップのメタデータをチェック • 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かの確認 • テーブルのパーティション割り付けビットマップに重複がないことの確認 • 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となっているかの確認 • 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、テーブルの割り付けビットマップのスーパーセットとなっているかの確認 • メインキャッシュページが IQ メインストアページと一致しているかの確認 <hr/> <p>注意： <code>sp_iqcheckdb</code> 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確認しきれないことになります。</p> <hr/> <p>次の場合は allocation モードで実行する。</p> <ul style="list-style-type: none"> • 重複ブロックや未所有ブロックを検出する場合 (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) • ページヘッダエラーが発生した場合 <p>DBCC オプション resetclocks は、allocation モードでのみ使用される。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する。マルチプレックス機能については、</p>

モード	説明
	<p>『管理：マルチプレックス』を参照。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する。SAP 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。</p> <p>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る、DBCC 文 allocation database でのみ使用できる。resetclocks の構文は、次のとおり。</p> <pre>sp_iqcheckdb 'allocation database resetclocks'</pre>
Check	<p>すべてのデータベースページが、データベース全体、メインキャッシュ、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する。テーブルがパーティションに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する。</p> <p>クエリの実行時に、メタデータ、NULL カウント、または個別カウントのエラーが返された場合は、check モードで実行する。</p>

モード	説明
Verify	<p>データベース全体、メインキャッシュ、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インデックスを使用して確認する。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る。</p> <ul style="list-style-type: none"> • 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 <p>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を検出できる。</p> <ul style="list-style-type: none"> • 欠落キー • 欠落ロー <p>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する。</p> <p>クエリの実行時に、メタデータ、NULLカウント、または個別カウントのエラーが返された場合は、verifyモードで実行する。</p> <p>注意： <code>sp_iqcheckdb</code> 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。</p>
Dropleaks	<p>SAP 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る。ターゲットがDB領域の場合のdropleaks操作は、そのDB領域での読み/書き操作も防止しなければならない。データベースまたはDB領域リスト内のすべてのDB領域がオンラインでなければならない。</p> <p>マルチプレックスコーディネータノードのdropleaksモードは、マルチプレックス全体のリークブロック、重複ブロック、または余分なブロックも検出する。</p>

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と、データベース全体を検査するより時間を短縮できます。

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。

次の表は、4 つの **sp_iqcheckdb** モードの動作と出力内容を示します。

表 66 : 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	1 時間あたり 4TB
Check	割り付けのエラー 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	1 時間あたり 60GB
Verify	割り付けのエラー 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	1 時間あたり 15GB
Dropleaks	割り付けのエラー	割り付けの統計情報のみ	1 時間あたり 4TB

出力 :

実行モードに応じて、**sp_iqcheckdb** の出力に、結果の要約、エラー、統計情報、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、出力には最大で 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、SAP 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ます。これにより、ユーザは DBCC 操作の実行状況を把握できます。

例

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'
```

インデックス i1、i2、および dbo.t1.i3 の詳細チェックを実行します。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'
```


すべてのモードを組み合わせて、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。CPUの半分を使って、テーブル t2 のパーティション p1 に対しては簡単なチェックを、インデックス i1 に対しては詳細なチェックを、データベース全体に対しては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'
```

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'
```

テーブル t1 内の **FP** および **HG** インデックスと、テーブル t2 内の **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'
```

LVC セルの不整合をチェックします。

```
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
- 割り付けられていない LVC セルロー

これらのメッセージは、VARCHAR カラム、VARBINARY カラム、LONG BINARY (BLOB) カラム、または LONG VARCHAR (CLOB) カラムに不整合があることを示しています。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 セルは重大なエラーで、場合によっては、ダメージを受けたカラムを削除しないと解決できません。

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。LVC は LONG BINARY (BLOB) および LONG VARCHAR (CLOB) によっても使用されます。

出力例：

sp_iqcheckdb 'allocation database' を実行します。

```

=====
DBCC Allocation Mode Report
=====
      DBCC Status                               No Errors Detected
=====
Allocation Summary
=====
      Blocks Total                               25600
      Blocks in Current Version                 5917
      Blocks in All Versions                   5917
      Blocks in Use                             5917
      % Blocks in Use                           23
=====
Allocation Statistics
=====
      Marked Logical Blocks                     8320
      Marked Physical Blocks                   5917
      Marked Pages                             520
      Blocks in Freelist                       2071196
      Imaginary Blocks                         2014079
      Highest PBN in Use                       1049285
      Total Free Blocks                        19683
      Usable Free Blocks                       19382
      % Total Space Fragmented                  1
      % Free Space Fragmented                  1
      Max Blocks Per Page                      16
      1 Block Page Count                       165
      3 Block Page Count                       200
      4 Block Page Count                       1
      10 Block Page Count                      1
      16 Block Page Count                      153
      2 Block Hole Count                       1
      3 Block Hole Count                       19
      6 Block Hole Count                       12
      7 Block Hole Count                       1
      10 Block Hole Count                      1
      15 Block Hole Count                      1
      16 Block Hole Count                      1220
=====
Partition Summary
      Database Objects Checked                 2
      Blockmap Identity Count                 2
      Bitmap Count                            2
=====
Connection Statistics
=====
      Sort Records                             3260
      Sort Sets                                2
=====
DBCC Info
=====
      DBCC Work units Dispatched              197
      DBCC Work units Completed               197
      DBCC Buffer Quota                        255
=====

```

DBCC Per-Thread Buffer Quota	255
Max Blockmap ID found	200
Max Transaction ID found	404

注意： このレポートでは、リークされた領域が示されることがあります。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。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcheckoptions プロシージャ

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
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

DBA ユーザには、すべてのロールとユーザに対して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、そのユーザ自身のテンポラリ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。

備考

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。

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は、データベースの現在の値とデフォルト値、および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、SAP Sybase IQ と SQL Anywhere データベースのすべてのオプションを対象としています。SAP 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 SAP Sybase IQ のデフォルト値が再変更されない限り、**sp_iqcheckoptions** はそのオプションをリストに含めません。

sp_iqcheckoptions を実行すると、DBA にはすべてのロール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、そのユーザ自身のテンポラリオプ

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 67 : `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ロールの名前。データベース作成時には、すべてのオプションが PUBLIC ロールに設定される。PUBLIC 以外のロールまたはユーザに設定されたオプションが表示される。
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

次の例では、テンポラリオプション `APPEND_LOAD` が ON に設定され、ロール `myrole` のオプション `MAX_WARNINGS` が 9 に設定されています。ユーザ `joel` では、`MAX_WARNINGS` にテンポラリ値として 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	Checkpoint_time	20	60	Temporary
DBA	Connection_authent	Company=MyComp; 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
myrole	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; 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

nent				
joel	Max_Warnings	55	281474976710655	Temporary
joel	Thread_count	25	0	Temporary

sp_iqclient_lookup プロシージャ

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 SAP 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 アドレスを指定する。
- **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する。
- **UserID** – SAP Sybase IQ ユーザの ID を指定する。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- SELECT ANY TABLE
- MONITOR
- DROP CONNECTION
- SERVER OPERATOR

備考

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、UserID を含む、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。次に例を示します。

```
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、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 jcazale 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：ユーザの指定した引数が正しくない場合、結果は空になります。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcolumn プロシージャ

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' ]
```

パラメータ

- **構文 1** – *table_name* を指定しないで *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 のようになります。
- **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。'table_name' と 'table_owner' を一重引用符で囲みます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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* パラメータと *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。

カラム名	説明
table_name	テーブル名。
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。

カラム名	説明
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)。
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。

参照：

- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcolumn プロシージャの例

sp_iqcolumn を使用する場合は、この例を参照してください。

次の構文はどちらも、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
Departments	GROUPO	DepartmentID	integer	4	0
Departments	GROUPO	DepartmentName	char	40	0
Departments	GROUPO	DepartmentHead	integer	4	0
cardinality	location	isPartitioned	remarks	check	
5	Main	N	(NULL)	(NULL)	
0	Main	N	(NULL)	(NULL)	
0	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_iqcolumnmetadata プロシージャ

1 つまたは複数のテーブルのカラムインデックスに関する詳細情報を返します。

構文

```
sp_iqcolumnmetadata [ table.name [, owner-name ] ]
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- ALTER ANY INDEX システム権限
- ALTER ANY OBJECT システム権限
- テーブルに対する REFERENCE パーミッション
- テーブルを所有している

備考

sp_iqcolumnmetadata は、インデックスメタデータを読み込んで、ベーステーブルとグローバルテンポラリテーブル両方のカラムインデックスに関する詳細情報を返します。グローバルテンポラリテーブルについてレポートされたインデックスメタデータは、そのテーブルの個別インスタンスのものであります。

オプションの *[table.name]* パラメータを指定すると、そのテーブルに関する詳細情報が生成されます。*[table.name]* パラメータを省略すると、データベース内のすべてのテーブルに関する詳細情報が生成されます。

sp_iqcolumnuse プロシージャ

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

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。 MONITOR システム権限。

備考

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 68 : 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)。

ヒント： INDEX_ADVISOR オプションは、クエリのパフォーマンスを向上させる可能性がある追加のカラムインデックスを推奨するメッセージを生成します。

例

sp_iqcolumnuse プロシージャからの出力例を示します。

TableName	NRef	ColumnName	Owner	UID	LastDT
orders		o_orderdate	DBA		20070917
22:41:22..	13				
orders		o_shippriority	DBA		20070917
22:41:22..	13				
lineitem		l_orderkey	DBA		20070917
22:41:22..	13				
lineitem		l_extendedp..	DBA		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：

- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)
- sp_iqworkmon プロシージャ (637 ページ)

sp_iqconnection プロシージャ

接続およびバージョンについての情報を表示します。この情報には、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 SAP 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が含まれます。

構文

```
sp_iqconnection [ connhandle ]
```

適用対象

シンプレックスとマルチプレックス。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

備考

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されると、**sp_iqconnection** はその接続に対応する 1 つのローのみを返します。

sp_iqconnection は、有効な各接続に対して 1 つのローを返します。ConnHandle、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr、LastIdle の各カラムは、Number、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr、LastIdle の各接続プロパティ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、SAP Sybase IQ エンジンの SAP 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。

MPXServerName カラムには、次の表に示すようにノード間通信 (INC) に関連する情報が格納されています。

実行されているサーバ	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"> ローカル/ユーザ接続の場合は NULL。 各 INC 接続 (オンデマンド接続または専用ハートビート接続のいずれか) のセカンダリノードのサーバ名 (接続元) の値を含む。
マルチプレックスセカンダリ	<ul style="list-style-type: none"> ローカル/ユーザ接続の場合は NULL。 コーディネータのサーバ名 (接続元) の値を含む。

Java アプリケーションでは、TDS クライアントから SAP Sybase IQ 固有の接続プロパティを RemotePWD フィールドで指定します。次の例は、IQ 固有の接続パラメータの指定方法を示します。**myconnection** は IQ 接続名です。

```
p.put ("RemotePWD", " , , CON=myconnection" );
```

カラム名	説明
ConnHandle	接続の ID 番号。
ConnectionName	パラメータで指定される接続名。
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。
ReqType	最後の要求のタイプを示す文字列。

カラム名	説明
IQCmdType	SAP Sybase IQ 側で現在実行されているコマンド (存在する場合)。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る。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、特殊な制御コマンド (OPEN と CLOSE 、 BACKUP DATABASE 、 RESTORE DATABASE など) で構成される。
LastIQCmdTime	この接続の SAP 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 (存在する場合)。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る。カーソル状態は内部の SAP 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る。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、および COMPLETED。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る。たとえば、状態 PREPARED は、カーソルが実行中であることを示す。
IQthreads	現在、接続に割り当てられている SAP 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る。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る。
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。この ID 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 と同じである。
ConnCreateTime	接続が作成された時刻。
TempTableSpaceKB	この接続が IQ テンポラリテーブルに格納されているデータに使用している IQ テンポラリストアの領域 (KB 単位)。
TempWorkSpaceKB	この接続が、ソート、ハッシュ、テンポラリビットマップなどの作業領域として使用している IQ テンポラリストアの領域 (KB 単位)。ビットマップや、SAP Sybase IQ テンポラリ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 に反映される。

カラム名	説明
IQConnID	.iqmsg ファイル内のすべてのメッセージの一部として含まれている 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ある。
satoiq_count	SAP Sybase IQ エンジンの SQL Anywhere 側から IQ 側への超過の数の表示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場合がある。結果セットはローのバッファに返され、satoiq_count や iqtosa_count がローごとに 1 回増分することはない。
iqtosa_count	SAP Sybase IQ エンジンの IQ 側から SQL Anywhere 側への超過の数の表示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場合がある。
CommLink	接続用の通信リンク。これは、SAP Sybase IQ がサポートするネットワークプロトコルのいずれかで、同一マシン接続の場合は「local」になる。
NodeAddr	クライアント/サーバ接続のクライアント側に対応するノード。
LastIdle	要求間のチックの数。
MPXServerName	INC 接続の場合、varchar(128) 値には、INC 接続を開始したマルチプレックスサーバの名前が含まれる。INC 接続でない場合は NULL になる。
LSName	接続の論理サーバ名。論理サーバのコンテキストが未知または適用不可の場合、NULL となる。
INCConnName	ユーザ接続の基礎となる INC 接続の名前。このカラムのデータ型は varchar(255)。sp_iqconnection でサスペンドされたユーザ接続の INC 接続名が表示される場合、そのユーザ接続にはサスペンドされた関連 INC 接続も存在する。
INCConnSuspended	このカラムの値 "Y" は、ユーザ接続の基礎となる INC 接続がサスペンド状態であることを示す。値 "N" は、接続がサスペンドされていないことを示す。

例

sp_iqconnection

```

ConnHandle      Name      Userid      LastReqTime      ReqType
=====
1  'SQL_DBC_100525210'  'DBA'      '2011-03-28 09:29:24.466'  'OPEN'

            IQCmdType      LastIQCmdTime      IQCursors      LowestIQCursorState
=====
'IQUTILITYOPENCURSOR'  2011-03-28 09:29:24.0      0      'NONE'

IQthreads      TxnID      ConnCreateTime      TempTableSpaceKB      TempWorkSpaceKB
=====

```

```

=====
0 3352568 2011-03-28 09:29:20.0 0 0
=====
IQconnID satoiq_count iqtosa_count CommLink NodeAdd LastIdle MPXServerName
=====
34 43 2 'local' '' 244 (NULL)
=====
LSName INCConnName INCConnSuspended
=====
Finance_LS 'IQ_MPX_SERVER_P54' 'Y'
=====

```

sp_iqconstraint プロシージャ

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

```
sp_iqconstraint ['table-name', 'column-name', 'table-owner']
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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、参照制約が含まれ、**CREATE TABLE** 文と **ALTER TABLE** 文の両方またはいずれか一方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,ck2PTAB2 bob ASIQ_IDX_T27_HG unique ck1,ck2
selftab bob CK6FK4 Y
ASIQ_IDX_T206_I42_HG ck1,ck2selftab 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

システムプロシージャ

- 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
- ロケーション ("TEMP"、"MAIN"、または "SYSTEM")

参照：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcontext プロシージャ

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

```
sp_iqcontext [ connhandle ]
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- MANAGE ANY USER
- MONITOR

備考

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。たとえば、**SELECT CONNECTION_PROPERTY('NUMBER')** のようになります。

有効な *connhandle* の入力パラメータで呼び出されると、**sp_iqcontext** はその接続に対応する情報のみを返します。

sp_iqcontext 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。

- 文のテキストを、**sp_iqconnection** 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。
- 文のテキストを、**-zr** 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。
- 接続情報を使用して、**sp_iqcontext** 内の文のテキストを、ファイル `.iqmsg` (SAP 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) 内の相当する行と照合する。
- SAP Sybase IQ スタックトレース (`stktrc-yyyyymmdd-hhnnss_#.iq`) が作成されていれば、文のテキストをこれと照合する。
- OS のスタックトレース (Sun Solaris の `pstack` など) が生成されている場合、この情報をそれと照合する。

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 69 : **sp_iqcontext** のカラム

カラム名	説明
ConnOrCursor	CONNECTION、CURSOR、または DQP。
ConnHandle	接続の ID 番号。DQP の場合は 0。
Name	サーバの名前 (リーダ名)。
Userid	接続、カーソル、または DQP ワークのユーザ 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す。 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す。 カラム 1 が DQP の場合、CONNECTION は 0 の値を返す。
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はある。DQP スレッドの場合、DQP ワークに割り当てられているスレッドの数を表す。
TxnID	現在のトランザクションのトランザクション ID。ワークスレッドの場合、リーダのトランザクション ID を表す。
ConnOrCurCreate-Time	この接続、カーソル、または DQP が作成された時間。

カラム名	説明
IQConnID	.iqmsg ファイル内のすべてのメッセージの一部として表示される接続 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ある。
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 は優先度低を示す。値 -1 は、IQGovernPriority を操作に適用しないことを示す。IQGovernPriority 値は、データベースオプション IQGOVERN_PRIORITY を使用して設定する。 DQP 接続の場合、このカラムには、コマンドはありませんと表示される。
CmdLine	実行されるユーザコマンドの最初の 4096 文字。 DQP 接続の場合、このカラムには、コマンドはありませんと表示される。
Attribute	分散されているクエリのユニークな ID。

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について結果を表示した場合の出力の一部です。カラム名は、表示幅の制限によってトランケートされます。

```

ConnOrCu.. ConnHandle Name UserId numIQ.. IQthr.. TxnID Conn.. IQcon..
IQGov.. Cmd.. Attributes
CONNECTION 2 sun7bar dbo 0 0 0 2010-08-04 15:15:40.0 15 No command NO
COMMAND
CONNECTION 7 sun7bar dbo 0 0 0 2010-08-04 15:16:00.0 32 No command NO
COMMAND
CONNECTION 10 sun7bar dbo 0 0 0 2010-08-04 15:16:21.0 46 No command NO
COMMAND
...
CONNECTION 229 sun7bar DBA 0 0 1250445 2010-08-05 18:28:16.0 50887 2
select server_name,
inc_state, coordinator_failover from sp_iqmpxinfo() order by server_name
...
DQP 0 dbsrv2873_node_c1DBA 0 1 10000 2010-08-05 18:28:16.0 no command no
command Query ID:
12345; Condition: c1 > 100;
DQP 0 dbsrv2873_node_c1DBA 0 1 10001 2010-08-05 18:28:16.0 no command no
command Query ID:
12346; Node #12 Join (Hash);

```

出力の 1 行目には、connection 2 (IQ 接続 ID は 15) が表示されています。この接続のサーバは sun7bar、ユーザは dbo です。sp_iqcontext の発行時に、この接続はコマンドを実行していませんでした。

connection 229 は、実行中のユーザコマンド (カラムに表示できる最大 4096 文字よりも少ない文字のコマンド) を示します。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。

接続ハンドル (この例の最初の接続では 2) 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の 15) は、.iqmsg ファイル内の結果を識別します。UNIX システムでは、**grep** 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することができます。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。

最終行から 2 番目の行 (TxnID 10000) は、DQP ワークスレッドを示します。ワーク接続は 2 つの不変条件を実行しています。

最終行 (TxnID 10001) は、接続がハッシュジョインを実行中であることを示します。

参照：

- CONNECTION_PROPERTY 関数 [システム] (177 ページ)
- sp_iqshowpsexec プロシージャ (579 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcopyloginpolicy プロシージャ

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

```
call sp_iqcopyloginpolicy ('existing-policy-name', 'new-policy-name' )
```

構文 2

```
sp_iqcopyloginpolicy 'existing-policy-name', 'new-policy-name'
```

パラメータ

- **existing-policy-name** – コピーするログインポリシー。
- **new-policy-name** – 作成する新しいログインポリシーの名前 (CHAR(128))。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MANAGE ANY LOGIN POLICY システム権限。

例

root という既存のログインポリシーからログインポリシーオプション値をコピーして、*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')
```

参照：

- `sp_expireallpasswords` システムプロシージャ (721 ページ)
- `sp_iqaddlogin` プロシージャ (437 ページ)
- `sp_iqmodifylogin` プロシージャ (535 ページ)
- `sp_iqpassword` プロシージャ (551 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqcursorinfo プロシージャ

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 ]
```

パラメータ

- **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。
コピーするログインポリシー。
- **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** は指定された接続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。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。 MONITOR システム権限。

備考

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には、指定しないパラメータの位置に NULL を入力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 70 : `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する。
<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する。
<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する。
<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する。

`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` プロシージャによって、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できます。また、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って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` は指定した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、次のカラムに情報を返します。

表 71 : `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 : カーソルは更新可能ではない。

カラム名	説明
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	.iqmsg ファイル内のすべてのメッセージの一部として表示される 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ある。
UserID	カーソルを作成、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間。
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る。
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る。
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。
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() call proc2()	

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdatatype プロシージャ

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 ]
```

パラメータ

- **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 が所有するデータ型) に関する情報のみを表示します。
 - **ALL**: ユーザデータ型およびシステムデータ型に関する情報を表示します。
 - その他の値： ユーザ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 も SYS も所有していないデータ型) に関する情報のみがデフォルトで表示されます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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` および `sp_iqdatatype NULL, user1` とします。

表 72 : **sp_iqdatatype** の使用例

構文	出力
<code>sp_iqdatatype</code>	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する。
<code>sp_iqdatatype country_t</code>	<code>country_t</code> という名前のユーザ定義データ型に関する情報を表示する。

構文	出力
<code>sp_iqdatatype non_existing_type</code>	データ型 <code>non_existing_type</code> が存在しないため、ローは返されない。
<code>sp_iqdatatype NULL, DBA</code>	DBA が所有するすべてのユーザ定義データ型に関する情報を表示する。
<code>sp_iqdatatype country_t, DBA</code>	DBA が所有するデータ型 <code>country_t</code> に関する情報を表示する。
<code>sp_iqdatatype rowid</code>	<code>rowid</code> は、システム定義のデータ型である。 <code>rowid</code> という名前のユーザ定義データ型が存在しない場合、ローは返されない (デフォルトでは、ユーザ定義データ型のみが返される)。
<code>sp_iqdatatype rowid, SYS</code>	データ型 <code>rowid</code> はユーザ定義データ型ではないため、ローは返されない (デフォルトでは、ユーザ定義データ型のみが返される)。
<code>sp_iqdatatype NULL, NULL, SYSTEM</code>	(<code>dbo</code> または <code>SYS</code> が所有する) すべてのシステム定義データ型に関する情報を表示する。
<code>sp_iqdatatype rowid, NULL, SYSTEM</code>	システムデータ型 <code>rowid</code> に関する情報を表示する。
<code>sp_iqdatatype NULL, NULL, 'ALL'</code>	ユーザ定義データ型およびシステムデータ型に関する情報を表示する。

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って、**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** は指定の所有者が所有す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 73 : sp_iqdatatype 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。
NULL	Y はユーザ定義データ型が NULL を許可することを示し、N は許可しないことを示す。U は、データ型について NULL 値が指定されていないことを示す。
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) を表示する。
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 を表示する。
"default"	データ型のデフォルト値
"check"	データ型の CHECK 条件

例

country_t 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 country_t
```

```
type_name    creator    nulls     width    scale    "default"    "check"
country_t    DBA       U         15       0        (NULL)       (NULL)
```

参照：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdbsize プロシージャ

現在のデータベースのサイズを表示します。

構文

```
sp_iqdbsize ( [ main ] )
```

適用対象

シンプレックスとマルチプレックス。

権限

ALTER DATABASE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

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。

マルチプレックスデータベース上で実行する場合、デフォルトのパラメータは main です。これにより、共有 IQ ストアのサイズが返されます。

RLV 対応テーブルにローが存在しないときに実行すると、Physical Blocks、RLVLogBlocks、および RLVLogKBytes カラムには 0 以外のエントリが格納され、その他のカラムには 0 が格納されます。これは、ローレベルのバージョン管理テーブルが存在しないことを示します。

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)。 IQ データベースは、1 つ以上の DB 領域で構成される。各 DB 領域は、最初にメガバイト単位で指定されたサイズに固定されている。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ってブロックに変換される。Physical Blocks カラムは、各 SAP Sybase IQ DB 領域のサイズの累計値が反映され、ブロック単位で示される。
KBytes	データベースの合計サイズ (KB 単位)。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ある。このブロックサイズは IQ ページサイズに依存する。

カラム名	説明
Pages	テーブル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で表示するために必要な IQ ページの合計数。この値は常に、Compressed Pages (次の sp_iqdbsize カラム) の値以上になる。
Compressed Pages	テーブルに格納されている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。IQ ページがメモリからディスクに書き込まれるときに、SAP Sybase IQ によってページが圧縮されるため、この値は、常に Pages (前の sp_iqdbsize カラム) の値以下になる。sp_iqdbsize Compressed Pages カラムは、圧縮されたページの数を表す。
NBlocks	テーブルのデータの格納に使用されるブロック単位の合計サイズ。この値は常に、sp_iqdbsize Physical Blocks の値以下になる。
Catalog Blocks	テーブルのメタデータの格納に使用されているブロック単位の合計サイズ。
RLVLogBlocks	RLV ストアのログ情報に使用されるブロックの数。
RLVLogKBytes	RLV ログの合計サイズ (KB 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
RLVLogBlocks RLVLogKBytes
```

```
=====
=====
/system1/sybase/IQ-16_0/demo/iqdemo.db
      1280      522      688                257      1119                18
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdbspace プロシージャ

各 SAP Sybase 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 ]
```

適用対象

シンプレックスとマルチプレックス。

権限

MANAGE ANY DBSPACE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqdbspace の情報は、データを移動する必要があるかどうかの判断に、また移動されたデータについては旧バージョンの割り付けが解除されているかどうかの判定に使用されます。

カラム名	説明
DBSpaceName	CREATE DBSPACE 文で指定された DB 領域の名前。 CREATE DATABASE...CASE IGNORE または CASE RESPECT の指定に関係なく、DB 領域名は常に大文字と小文字が区別されない。
DBSpaceType	DB 領域のタイプ (MAIN、SHARED_TEMP、TEMPORARY、RLV、または CACHE)。
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	F (オフ)。
StripeSize	ディスクストライピングの有効化時は、常に 1。
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。
OkToDrop	DB 領域を削除できる場合は "Y"、それ以外の場合は "N"。

BlkTypes カラムのブロックタイプ識別子の値は、次のとおりです。

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
R	RLV フリーリストマネージャ
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。

例

DB 領域に関する情報を表示します。

```
sp_iqdbspace;
```

注意： 出力内容をわかりやすくするため、次の例は iqdemo データベース内のオブジェクトを示しています。iqdemo には iq_main というサンプルのユーザ DB 領域が含まれていますが、この領域はユーザ独自のデータベースには存在しない場合があります。

DBSpaceName	DBSpaceType	Writable
IQ_MAIN	MAIN	T

DBSpaceName	DBSpaceType	Writable
IQ__SYSTEM_MAIN	MAIN	T
IQ_SYSTEM_TEMP	TEMPORARY	T
myDas	CACHE	T

(続き) Online	Usage	DBSpaceName
T	55	IQ_MAIN
T	21	IQ__SYSTEM_MAIN
T	1	IQ_SYSTEM_TEMP
T	1	myDas

(続き) Reserve	NumFiles	NumRWFiles
200M	1	1
50M	1	1
50M	1	1
0B	5	5

(続き) DBSpaceName	Stripingon	Stripe Size
IQ_MAIN	T	1K
IQ__SYSTEM_MAIN	F	8K
IQ_SYSTEM_TEMP	F	8K
myDas	T	1K

(続き) Blk Types	OkTo Drop
1H, 5169A, 190	N

(続き)	OkTo Drop
Blk Types	
1H, 7648F, 32D, 128M	N
1H, 64F, 32A	N
5, 192FH	Y

参照：

- sp_iqindexinfo プロシージャ (516 ページ)
- sp_iqdbspaceinfo プロシージャ (481 ページ)
- sp_iqspaceinfo プロシージャ (581 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdbspaceinfo プロシージャ

指定のテーブル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。RLV DB 領域はサポートされていません。

構文

```
sp_iqdbspaceinfo [ dbspace-name ] [ , owner_name ] [ ,  
object_name ] [ , object-type ]
```

パラメータ

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に依存することなく指定できます。

- **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 行ごとに表示します。指定しない場合、この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、指定の所有者のテーブルのみの出力を表示します。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に関する情報を表示します。
- **object_name** – テーブルの名前。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に関する情報を表示します。
- **object_type** – 有効な **table** オブジェクト。

sp_iqdbspaceinfo ストアドプロシージャでは、*dbspace_name*、*object_name*、および *owner_name* の解釈に、ワイルドカード文字がサポートされています。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。

適用対象

シンプレックスとマルチプレックス。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- BACKUP DATABASE
- SERVER OPERATOR
- MANAGE ANY DBSPACE

備考

RLV DB 領域を指定した場合、このプロシージャは結果を返しません。

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量を 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の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、これらはそれぞれバイト、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)。この情報は、*dbspace_name*、*object_name*、および *owner_name* でソートされます。

表 74 : sp_iqdbspaceinfo のカラム

カラム名	説明
dbspace_name	DB 領域の名前。
object_type	オブジェクトのタイプ (table または joinindex のみ)。
owner	オブジェクトの所有者の名前。
object_name	DB 領域にあるオブジェクトの名前。
object_id	オブジェクトのグローバルオブジェクト ID。
id	オブジェクトのテーブル ID。
columns	指定の DB 領域のカラム記憶領域のサイズ。

カラム名	説明
indexes	指定の DB 領域のインデックス記憶領域のサイズ。システムで生成されたインデックス(一意性制約の HG インデックス、FP インデックスなど)は使用できない。
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 領域のプライマリー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。
dbspace_online	DB 領域がオンライン (Y) か、オフライン (N) かを示す。

-r スイッチ (読み込み専用) で起動したサーバに対して `sp_iqdbspaceinfo` を実行すると、「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` などの他のストアプロシージャでは、このエラーは発生しません。

例

注意： 出力内容をわかりやすくするため、次の例は `iqdemo` データベース内のオブジェクトを示しています。`iqdemo` には `iq_main` というサンプルのユーザ DB 領域が含まれていますが、この領域はユーザ独自のデータベースには存在しない場合があります。

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 288K	table	GROUPO	Contacts	3538	732
iq_main 240K	table	GROUPO	Customers	3515	731
iq_main	table	GROUPO	Departments	3632	738 72K
iq_main 408K	table	GROUPO	Employees	3641	739
iq_main 72K	table	GROUPO	FinancialCodes	3612	736
iq_main	table	GROUPO	FinancialData	3621	737 96K

システムプロシージャ

iq_main 3593	735	table	272K	GROUPO	Products		
iq_main 120K		table		GROUPO	SalesOrderItems	3580	734
iq_main 144K		table		GROUPO	SalesOrders	3565	733
indexes ace_online	metadata	primary_key	unique_constraint	foreign_key	dbsp		
0B	1.37M	0B	0B	0B	Y		
0B	464K	0B	0B	0B	Y		
0B	1.22M	0B	0B	0B	Y		
0B	5.45M	24K	0B	48K	Y		
48K	4.63M	24K	0B	0B	Y		
0B	1.78M	24K	0B	48K	Y		
0B	8.03M	24K	0B	48K	Y		
0B	1.53M	24K	0B	0B	Y		
0B	2.19M	24K	0B	48K	Y		
192K	4.67M	24K	0B	0B	Y		
0B	2.7M	24K	0B	104K	Y		
0B	3.35M	24K	0B	144K	Y		

データベース内の指定 DB 領域に指定ユーザが所有する、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 288K	table	GROUPO	Contacts	3538	732		
iq_main 240K	table	GROUPO	Customers	3515	731		
iq_main	table	GROUPO	Departments	3632	738	72K	
iq_main 408K	table	GROUPO	Employees	3641	739		
iq_main 72K	table	GROUPO	FinancialCodes	3612	736		
iq_main	table	GROUPO	FinancialData	3621	737	96K	
iq_main 272K	table	GROUPO	Products	3593	735		
iq_main 120K	table	GROUPO	SalesOrderItems	3580	734		
iq_main 144K	table	GROUPO	SalesOrders	3565	733		
indexes ace_online	metadata	primary_key	unique_constraint	foreign_key	dbsp		
0B	5.45M	24K	0B	48K	Y		
48K	4.63M	24K	0B	0B	Y		
0B	1.78M	24K	0B	48K	Y		
0B	8.03M	24K	0B	48K	Y		
0B	1.53M	24K	0B	0B	Y		
0B	2.19M	24K	0B	48K	Y		
192K	4.67M	24K	0B	0B	Y		

0B	2.7M	24K	0B	104K	Y
0B	3.35M	24K	0B	144K	Y

データベース内の指定 DB 領域に指定ユーザが所有する、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	dbspace_online
0B	1.78M	24K	0B	48K	Y

参照：

- sp_iqindexinfo プロシージャ (516 ページ)
- sp_iqdbspace プロシージャ (477 ページ)
- sp_iqspaceinfo プロシージャ (581 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdbspaceobjectinfo プロシージャ

指定の DB 領域のテーブル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。RLV DB 領域はサポートされていません。

構文

```
sp_iqdbspaceobjectinfo [ dbspace-name ] [ , owner_name ] [ , object_name ] [ , object-type ]
```

パラメータ

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。

- **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する。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、指定の所有者のテーブルのみの出力を表示します。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の情報を表示する。
- **object-name** – テーブルの名前。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の情報を表示します。
- **object-type** – **table** オブジェクトの有効なオブジェクトタイプ。

sp_iqdbspaceobjectinfo ストアドプロシージャでは、*dbspace_name*、*object_name*、および *owner_name* の解釈に、ワイルドカード文字がサポートされています。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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

RLV DB 領域を指定した場合、このプロシージャは結果を返しません。

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 *dbspace_name* および *owner and object_name* でソートして表示します。

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 75 : sp_iqdbspaceobjectinfo のカラム

カラム名	説明
<i>dbspace_name</i>	DB 領域の名前。
<i>dbspace_id</i>	DB 領域の識別子。
<i>object_type</i>	テーブル。
<i>owner</i>	オブジェクトの所有者の名前。
<i>object_name</i>	DB 領域にあるテーブルオブジェクトの名前。
<i>object_id</i>	オブジェクトのグローバルオブジェクト ID。
<i>id</i>	オブジェクトのテーブル ID。
<i>columns</i>	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る。結果は n/N フォームで表示される (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)。
<i>indexes</i>	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る (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)。一意性制約の場合、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ない。
<i>metadata</i>	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)。

カラム名	説明
primary_key	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)。
unique_constraint	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る (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)。
foreign_key	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る (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)。
partitions	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)。

例

出力内容をわかりやすくするため、次の例は iqdemo データベース内のオブジェクトを示しています。iqdemo には iq_main というサンプルのユーザ DB 領域が含まれていますが、この領域はユーザ独自のデータベースには存在しない場合があります。

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 d id columns	dbspace_id	object_type	owner	object_name	object_i
iq_main 741 4/4	16387	table	DBA	emp1	3689
iq_main 740 1/1	16387	table	DBA	iq_dummy	3686
iq_main 742 4/4	16387	table	DBA	sale	3698
iq_main 732 12/12	16387	table	GROUP0	Contacts	3538
iq_main 731 10/10	16387	table	GROUP0	Customers	3515
iq_main 738 3/3	16387	table	GROUP0	Departments	3632
iq_main 739 21/21	16387	table	GROUP0	Employees	3641
iq_main 736 3/3	16387	table	GROUP0	FinancialCodes	3612
iq_main 737 4/4	16387	table	GROUP0	FinancialData	3621
iq_main 735 8/8	16387	table	GROUP0	Products	3593
iq_main	16387	table	GROUP0	SalesOrderItems	3580

システムプロシージャ

id	id	columns	dbspace_name	dbspace_id	object_type	owner	object_name	object_id
iq_main	734	5/5	16387	table	GROUPO	SalesOrders	3565	
iq_main	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			

データベース内で指定した DB 領域の指定ユーザが所有す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	dbspace_id	object_type	owner	object_name	object_id
iq_main	16387	table	GROUPO	Contacts	3538
iq_main	732	2/12			
iq_main	16387	table	GROUPO	Customers	3515
iq_main	731	10/10			
iq_main	16387	table	GROUPO	Departments	3632
iq_main	738	3/3			
iq_main	16387	table	GROUPO	Employees	3641
iq_main	739	21/21			
iq_main	16387	table	GROUPO	FinancialCodes	3612
iq_main	736	3/3			
iq_main	16387	table	GROUPO	FinancialData	3621
iq_main	737	4/4			
iq_main	16387	table	GROUPO	Products	3593
iq_main	735	8/8			
iq_main	16387	table	GROUPO	SalesOrderItems	3580
iq_main	734	5/5			
iq_main	16387	table	GROUPO	SalesOrders	3565
iq_main	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

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：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdbstatistics プロシージャ

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
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。ALTER DATABASE システム権限。

備考

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		
===== =====		

システムプロシージャ

```
=====  
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:

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqdroplogin プロシージャ

SAP Sybase IQ のユーザアカウントを削除します。

構文 1

```
call sp_iqdroplogin ('userid')
```


構文 2

```
sp_iqdroplogin 'userid'
```

構文 3

```
sp_iqdroplogin  userid
```

構文 4

```
sp_iqdroplogin ('userid')
```

パラメータ

- **userid** – 削除するユーザの ID。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

sp_iqdroplogin は、指定されたユーザを削除します。

例

次のコマンドはすべて、ユーザ `rose` を削除します。

```
sp_iqdroplogin 'rose'
```

```
sp_iqdroplogin rose
```

```
call sp_iqdroplogin ('rose')
```

参照：

- `sp_iqaddlogin` プロシージャ (437 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqemptyfile プロシージャ

`dbfile` を空にし、`dbfile` 内のオブジェクトを、同じ DB 領域にある別の使用可能な読み込み／書き込み `dbfile` に移動します。RLV DB 領域にあるファイルには使用できません。

構文

```
sp_iqemptyfile ( logical-file--name )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- BACKUP DATABASE
- SERVER OPERATOR
- ALTER DATABASE

さらに、次のシステム権限のいずれかを所有している必要があります。

- INSERT ANY TABLE
- UPDATE ANY TABLE
- DELETE ANY TABLE
- ALTER ANY TABLE
- LOAD ANY TABLE
- TRUNCATE ANY TABLE
- ALTER ANY OBJECT

備考

sp_iqemptyfile は、dbfile を空にします。**sp_iqemptyfile** プロシージャを実行するには、DB 領域を読み込み専用にしておく必要があります。このプロシージャは、ファイル内のオブジェクトを、同じ DB 領域にある別の使用可能な読み込み／書き込み dbfile に移動します。使用可能な他の読み込み／書き込み dbfile がない場合は、SAP Sybase IQ から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。

dbfile が RLV DB 領域にある場合は、次のエラーメッセージが表示されます。

```
Cannot empty files in an rlv store dbspace.
```

例

dbfile **dbfile1** を空にします。

```
sp_iqemptyfile 'dbfile1'
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iquestdbspaces プロシージャ

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 )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- MANAGE ANY DBSPACE
- ALTER DATABASE

備考

sp_iqestdbspaces は、データのユニーク性に応じて、いくつかの推奨をレポートします。

推奨	説明
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 、 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 、 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。

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、正しい推定値は返されません。

表 76 : sp_iqestdbspaces のパラメータ

名前	データ型	説明
db_size_in_bytes	decimal(16)	データベースのサイズ (バイト)。
iq_page_size	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。

名前	データ型	説明
min_#_of_bytes	int	dbspace セグメントあたりの最小バイト数。デフォルトは 20,000,000 (20MB) です。
max_#_of_bytes	int	DB 領域セグメントあたりの最大バイト数。デフォルトは 2,146,304,000 (2.146GB) です。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iquestdbspaces プロシージャの例

sp_iquestdbspaces を使用する場合は、この例を参照してください。

```
sp_iquestdbspaces 12000000000, 65536, 500000000, 2146304000
```

dbspace files	種類	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。SAP 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 dbspace セグメントはすべて削除できます。

sp_iquestdbspaces を他のシステムストアプロシージャと一緒に使用する

sp_iquestdbspaces に必要なパラメータ *db_size_in_bytes* を提供するには、2 つのストアプロシージャを実行する必要があります。

sp_iquestdbspaces 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。

SAP 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。

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_iqestspace プロシージャ

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

```
sp_iqestspace ( table_name, #_of_rows, iq_page_size )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- CREATE ANY INDEX
- ALTER ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- テーブルを所有している

備考

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)。次の表に、**sp_iqestspace** パラメータを示します。

表 77 : sp_iqestspace のパラメータ

名前	データ型	説明
<i>table_name</i>	char (256)	テーブルの名前。
<i>#_of_rows</i>	int	テーブルにあるロー数。
<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。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqevent プロシージャ

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

```
sp_iqevent [ event-name ], [ event-owner ], [ event-type ]
```

パラメータ

- **event-name** – イベント名。
- **event-owner** – イベントの所有者。
- **event-type** – イベントのタイプ。指定できる値は次のとおりです。
 - **SYSTEM** – システムイベント (ユーザ SYS または dbo が所有するイベント) に関する情報のみを表示します。
 - **ALL** – 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。
 - **その他の値** – 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、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 が所有していないイベント) に関する情報のみがデフォルトで表示されます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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 および sp_iqevent NULL, user1 とします。

表 78 : sp_iqevent の使用例

構文	出力
sp_iqevent	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する。
sp_iqevent e1	イベント e1 に関する情報を表示する。
sp_iqevent non_existing_event	イベント non_existing_event が存在しないため、ローは返されない。
sp_iqevent NULL, DBA	DBA が所有するすべてのイベントに関する情報を表示する。
sp_iqevent e1, DBA	DBA が所有するイベント e1 に関する情報を表示する。
sp_iqevent ev_iqbegin_txn	ev_iqbegin_txn は、システム定義のイベントである。ev_iqbegin_txn という名前のユーザ定義イベントが存在しない場合、ローは返されない (デフォルトでは、ユーザ定義イベントのみが返される)。
sp_iqevent ev_iqbegin_txn, dbo	イベント ev_iqbegin_txn はユーザイベントではないため、ローは返されない (デフォルトでは、ユーザイベントのみが返される)。
sp_iqevent NULL, NULL, SYSTEM	(dbo または SYS が所有する) すべてのシステム定義イベントに関する情報を表示する。
sp_iqevent ev_iqbegin_txn, NULL, SYSTEM	システムイベント ev_iqbegin_txn に関する情報を表示する。
sp_iqevent ev_iqbegin_txn, dbo, ALL	dbo が所有するシステムイベント ev_iqbegin_txn に関する情報を表示する。

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って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** は指定の所有者が所有す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 79 : sp_iqevent のカラム

カラム名	説明
event_name	イベント名。
event_owner	イベントの所有者。
event_type	システムイベントの場合、SYSEVENTTYPE システムテーブルに挙げられたイベントタイプ。
enabled	イベントの発生が許可されているかどうかを示す (Y/N)
action の値	イベントハンドラの定義
condition	イベントハンドラの起動を制御するのに使用される WHERE 条件
location	イベントの発生が許可されている場所。 <ul style="list-style-type: none"> • C = 統合 (consolidated) • R = リモート (remote) • A = すべて (all)
remarks	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
ev_iqmpxcompact dbo              (NULL)         N            dbo.sp_iqlog...
               begin Declare
               _Catalog...

condition       location         remarks
(NULL)         A               (NULL)
(NULL)         A               (NULL)
```

参照：

- sp_iqcolumn プロシージャ (457 ページ)

- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqfile プロシージャ

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 ]
```

適用対象

シンプレックスとマルチプレックス。

権限

MANAGE ANY DBSPACE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqfile は、DB 領域の各 dbfile のデータの使用率、プロパティ、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、旧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。

カラム名	説明
DBSpaceName	CREATE DBSPACE 文で指定された DB 領域の名前。 CREATE DATABASE...CASE IGNORE または CASE RESPECT の指定に関係なく、DB 領域名は常に大文字と小文字が区別されない。
DBFileName	論理ファイル名。
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、TEMPORARY、RLV、または CACHE)。

カラム名	説明
RWMode	DB 領域のモード。常に、読み書き (RW)。
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。マルチプレックス設定のセカンダリノードに対して実行した場合は、NAがこのカラムに表示される。
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る。
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の有効化時は、常に 1。
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。
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"。

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
R	RLV フリーリストマネージャ
T	テーブルの使用

識別子	ブロックタイプ
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

*マルチブックスコミット 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;
```

```
sp_iqfile;
DBSpaceName,DBFileName,Path,SegmentType,RWMode,Online,
Usage,DBFileSize,Reserve,StripeSize,BlkTypes,FirstBlk,
LastBlk,OkToDrop

'IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','/sun1-c1/users/smith/mpx/m/
mpx_db.iq','MAIN','RW','T','21','
2.92G','0B','1K','1H',76768F,32D,19A,1850,128M,34B,32C'
,1,384000,'N'

'mpx_main1','mpx_main1','/sun1-c1/users/smith/mpx/m/
mpx_main1.iq','MAIN','RW','T','1'
,'100M','0B','1K','1H',1045440,1058239,'N'

'IQ_SHARED_TEMP','sharedfile1_bcp','/sun1-c1/users/smith/mpx/m/
f1','SHARED_TEMP','RO','T','0'
,'50M','0B','1K','1H',1,6400,'N'

'IQ_SHARED_TEMP','sharedfile2_bcp','/sun1-c1/users/smith/mpx/m/
f2','SHARED_TEMP','RO','T','0'
,'50M','0B','1K','1H',1045440,1051839,'N'

'myDAS','ssd_dev_1','/dev/raw/ssd_dev_1','CACHE','RW','T','2',
'20M','0B','1K','1H','64F','1','5120','N'
'myDAS','ssd_dev_2','/dev/raw/ssd_dev_2','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','522208','527327','N'
'myDAS','ssd_dev_3','/dev/raw/ssd_dev_3','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','1044416','1049535','N'
'myDAS','ssd_dev_4','/dev/raw/ssd_dev_4','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','1566624','1571743','N'
'myDAS','ssd_dev_5','/dev/raw/ssd_dev_5','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','2088832','2093951','N'

'IQ_SYSTEM_TEMP','IQ_SYSTEM_TEMP','/sun1-c1/users/smith/mpx/m/
mpx_db.iqtmp','TEMPORARY','RW',
'T','1','2.92G','0B','1K','1H',64F,33A',1,384000,'N'
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqhelp プロシージャ

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 ]
```

```
obj-type :  
    table  
    | view  
    | procedure  
    | event  
    | datatype
```

パラメータ

- **obj-name** – オブジェクトの名前。
- **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。

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、そのテーブルと関連するカラム、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。

指定できる値は次のとおりです。

- **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ー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。
- **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 が所有するオブジェクト) 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、ストアードプロシージャ、ファンクション、イベント、データ型に関する情報を表示します。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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には、指定しないパラメータの位置に NULL を入力する必要があります。たとえば、`sp_iqhelp NULL, NULL, NULL, SYSTEM` および `sp_iqhelp NULL, user1, "table"` とします。

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” が返されません。

表 80 : **sp_iqhelp** の使用例

構文	出力
<code>sp_iqhelp</code>	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する。
<code>sp_iqhelp t1, u1, "table"</code>	ユーザ <code>u1</code> が所有するテーブル <code>t1</code> に関する情報と、 <code>t1</code> に関連付けられているカラム、インデックス、制約に関する情報を表示する。
<code>sp_iqhelp NULL, u1, "view"</code>	ユーザ <code>u1</code> が所有するビュー <code>v1</code> に関する情報と、 <code>v1</code> に関連付けられているカラム、インデックス、制約に関する情報を表示する。
<code>sp_iqhelp sp2</code>	プロシージャ <code>sp2</code> と <code>sp2</code> のパラメータに関する情報を表示する。
<code>sp_iqhelp e1</code>	イベント <code>e1</code> に関する情報を表示する。
<code>sp_iqhelp dt1</code>	データ型 <code>dt1</code> に関する情報を表示する。

構文	出力
sp_iqhelp NULL, NULL, NULL, SYS- TEM	(dbo または SYS が所有する) 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する。
sp_iqhelp non_Existing_obj	オブジェクト non_existing_obj が存在しないため、エラー “Object ‘non_existing_obj’ not found” が返される。
sp_iqhelp NULL, non_Existing_user	ユーザ non_existing_user が存在しないため、エラー “User ‘non_existing_user’ not found” が返される。
sp_iqhelp t1, NULL, “apple”	“apple” は <i>obj-category</i> に入力できない値のため、エラー “Invalid object category ‘apple’ ” が返される。
sp_iqhelp t1, NULL, NULL, “USER”	“USER” は <i>obj-type</i> に入力できない値のため、エラー “Invalid object type ‘USER’ ” が返される。

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、制約、ストアドプロシージャ、ファンクション、イベント、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って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** は指定の所有者が所有するオブジェク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_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、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**、**sp_iqconstraint** によって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 81 : 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	table_name、table_owner、server_type、location、table_constraints、remarks
ビュー	view_name、view_creator、view_def、server_type、location、remarks
ストアドプロシージャ	proc_name、proc_creator、proc_defn、replicate、srvid、remarks
関数	proc_name、proc_creator、proc_defn、replicate、remarks
イベント	event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks
システムおよびユーザ定義のデータ型	type_name、creator、nulls、width、scale、default、check

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 82 : sp_iqhelp 詳細情報

オブジェクトタイプ	説明	カラム
テーブル	指定されたベーステーブル、そのカラム、インデックス、制約に関する情報を表示する。	<ul style="list-style-type: none"> テーブルカラム：table_name、table_owner、server_type、location、table_constraints、remarks カラムのカラム：column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks インデックスカラム：index_name、column_name、index_type、unique_index、location、remarks 制約カラム：constraint_name (役割)、column_name、index_name、constraint_type、foreign_table_name、foreign_table_owner、foreign_column_name、foreign_index_name、location

オブジェクトタイプ	説明	カラム
ビュー	指定のビューとそのカラムに関する情報を表示する。	<ul style="list-style-type: none"> ビューカラム：view_name、view_creator、view_def、server_type、location、remarks カラムのカラム：column_name、domain_name、width、scale、nulls、default、check、pkey、user_type、cardinality、est_cardinality、remarks
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する。	<ul style="list-style-type: none"> プロシージャカラム：proc_name、proc_creator、proc_defn、replicate、srvid、remarks パラメータカラム：parameter_name、type、width、scale、default、mode
関数	指定の関数とそのパラメータに関する情報を表示する。	<ul style="list-style-type: none"> 関数カラム：proc_name、proc_creator、proc_defn、replicate、srvid、remarks パラメータカラム：parameter_name、type、width、scale、default、mode
イベント	指定のイベントに関する情報を表示する。	<ul style="list-style-type: none"> イベントカラム：event_name、event_creator、enabled、location、event_type、action、external_action、condition、remarks
データ型	指定のデータ型に関する情報を表示する。	<ul style="list-style-type: none"> データ型のカラム：type_name、creator、nulls、width、scale、default、check

注意： システムプロシージャのプロシージャ定義 (proc-defn) は暗号化され、ビューに表示されません。

各出力カラムの詳細については、関連するストアドプロシージャを参照してください。たとえば、このテーブルカラムの詳細については、**sp_iqtable** プロシージャを参照してください。

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_ighelp 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)
```

参照：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqhelp の Adaptive Server との互換性

The SAP Sybase IQ **sp_iqhelp** ストアドプロシージャは、Adaptive Server **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、SAP 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。SAP 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、SAP 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。

SAP Sybase IQ と Adaptive Server のアーキテクチャの相違のために、SAP 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が表示す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。

sp_iqindex および sp_iqindex_alt プロシージャ

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' ]
```

パラメータ

- **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 および sp_iqindex Departments, NULL, DBA とします。
- **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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

データベース内のインデックスについての情報を表示します。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 83 : sp_iqindex および sp_iqindex_alt 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る。
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 行を生成します。**sp_iqindex_alt** は、複数カラムのインデックスがある場合、カラムごと、インデックスごとに 1 行を生成します。

参照：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqindex および sp_iqindex_alt のプロシージャ例

sp_iqindex および **sp_iqindex_alt** を使用する場合は、この例を参照してください。

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
Departments	GROUPO	DepartmentID	FP
Departments	GROUPO	DepartmentID	HG
Employees	GROUPO	DepartmentID	FP

(続き) index_name	unique_index	location	dbspace_id	remarks
ASIQ_IDX_T201_C1_FP	N	Main	16387	(NULL)
ASIQ_IDX_T201_C1_HG	U	Main	16387	(NULL)
ASIQ_IDX_T202_C5_FP	N	Main	16387	(NULL)

次の構文ではいずれも、テーブル所有者 GROUPO が所有するテーブル 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
Departments	GROUPO	DepartmentHeadID	FP
Departments	GROUPO	DepartmentID	FP
Departments	GROUPO	DepartmentID	HG
Departments	GROUPO	DepartmentName	FP

(続き) index_name	unique_index	location	dbspace_id	remarks
ASIQ_IDX_T201_C3_FP	N	Main	16387	(NULL)
ASIQ_IDX_T201_C1_FP	N	Main	16387	(NULL)
ASIQ_IDX_T201_C1_HG	U	Main	16387	(NULL)
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 と

State の複数カラムインデックスです。sp_iqindex_alt では、複数カラムインデックスのカラムごとに 1 ローが表示されます。

```
sp_iqindex_alt Employees, City
```

```
sp_iqindex_alt table_name='Employees',
               column_name='City'
```

table_name	table_owner	column_name	index_type
Employees	GROUPO	City	FP
Employees	GROUPO	City	HG
Employees	GROUPO	State	HG

(続き) index_name	unique_index	dbspace_id	remarks
ASIQ_IDX_T452_C7_FP	N	16387	(NULL)
emp_loc	N	16387	(NULL)
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
Employees	GROUPO	City	FP
Employees	GROUPO	City, State	HG

(続き) index_name	unique_index	dbspace_id	location	remarks
ASIQ_IDX_T452_C7_FP	N	16387	Main	(NULL)
emp_loc	N	16387	Main	(NULL)

sp_iqindexadvice プロシージャ

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、アドバイスメッセージ領域をクリアします。

構文

```
sp_iqindexadvice ( [ resetflag ] )
```

パラメータ

- **resetflag** – 呼び出し側からインデックスアドバイスメッセージ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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- ALTER ANY INDEX
- ALTER ANY OBJECT

備考

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。

INDEX_ADVISOR カラム:

表 84 : 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

sp_iqindexadvice プロシージャからの出力例を示します。

表 85 : sp_iqindexadvice の出力例

Advice	NInst	LastDT
Add a CMP index on DBA.tb (c2, c3) Predicate:(tb.c2 = tb.c3)	2073	2009-04-07 16:37:31.000
Convert HG index on DBA.tb.c4 to a unique HG	812	2009-04-06 10:01:15.000
Join Key Columns DBA.ta.c1 and DBA.tb.c1 have mismatched data types	911	2009-02-25 20:59:01.000

参照：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)
- sp_iqworkmon プロシージャ (637 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqindexfragmentation プロシージャ

SAP 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 [...]
```

パラメータ

- **table-name** – ターゲットとなる **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** – ターゲットとなる **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。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが必要です。MANAGE ANY DBASPACE システム権限。

例

DBA.prop_nu.prop_nu_a テーブル内のユニークな **HG** インデックスについて、内部インデックスの断片化がレポートされます。

Index	IndexType	Btree_Node_pages	GARRAY_FILL_FACTOR_PERCENT
DBA.prop_nu.prop_nu_a	HG	8	25
SQLCODE:	0		
フィルパーセント	btree ページ	garray ページ	bitmap ページ
0-10%	13	2	8
11-20%	1	8	0
21-30%	0	4	0
31-40%	3	20	0
41-50%	4	116	0
51-60%	6	4	0
61-70%	3	3	0
71-80%	4	1	0
81-90%	1	1	0
91-100%	192	276	0

注意：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クートされます。**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：

- `sp_iqindexmetadata` プロシージャ (518 ページ)
- `sp_iqindexinfo` プロシージャ (516 ページ)
- `sp_iqindexsize` プロシージャ (522 ページ)
- `sp_iqrebuildindex` プロシージャ (561 ページ)
- `sp_iqrowdensity` プロシージャ (574 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

`sp_iqindexinfo` プロシージャ

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 ]`
```

適用対象

シンプレックスとマルチプレックス。

権限

MANAGE ANY DBSPACE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

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。テーブル名が指定されている場合、`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。

指定された `table-name` または `index-name` があいまいな場合、またはオブジェクトが見つからない場合には、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。リソースのパーセンテージを使用して 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。

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。この情報によって、DBA はどの DB 領域を **relocate** モードにしてオブジェクトを移動する必要があるのかを判断できます。

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 86 : **sp_iqindexinfo** のカラム

カラム名	説明
Object	テーブル名またはインデックス名
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で、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の判断に役立つ

例

Departments テーブルのインデックスに関する情報を次のように表示します。

```
sp_iqindexinfo 'table GROUPO.Departments';
```

Object	DbspaceName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Departments	iq_main	288K	1	1,045,496.00	1,048,891.00
GROUPO.Departments.ASIQ_IDX_T779_C1_FP	iq_main	176K	1	1,047,197.00	1,047,328.00
GROUPO.Departments.ASIQ_IDX_T779_C2_FP	iq_main	160K	1	1,047,213.00	1,047,324.00
GROUPO.Departments.ASIQ_IDX_T779_C3_FP	iq_main	184K	1	1,047,229.00	1,047,317.00

Object	DbSpaceName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Departments.ASIQ_IDX_T779_C3_HG	iq_main	440K	1	1,048,421.00	1,048,796.00
GROUPO.Departments.ASIQ_IDX_T779_I4_HG	iq_main	288K	1	1,047,261.00	1,047,306.00

参照：

- sp_iqdbspace プロシージャ (477 ページ)
- sp_iqdbspaceinfo プロシージャ (481 ページ)
- sp_iqspaceinfo プロシージャ (581 ページ)
- sp_iqindexmetadata プロシージャ (518 ページ)
- sp_iqindexfragmentation プロシージャ (514 ページ)
- sp_iqindexsize プロシージャ (522 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqindexmetadata プロシージャ

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

```
dbo.sp_iqindexmetadata (index-name
[ , table-name [ , owner-name ] ] )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- ALTER ANY INDEX システム権限
- ALTER ANY OBJECT システム権限
- そのテーブルに対する REFERENCES 権限
- テーブルを所有している

備考

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。

sp_iqindexmetadata を通じて、カラムのユーザサプライヤ IQ UNIQUE 値が入手可能です。ユニークな HG または LF が存在している場合は、正確なカーディナリティがレポートされます。ユニークでない内 HG (のみ) が存在している場合は、カーディナリティとして 0 がレポートされます。

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 87 : 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、Style、Version、DBType、Maximum Width、EstUnique、TokenCount、NBit、CountSize、DictSize、CountLen、MaxKeyToken、MinKeyToken、MinCount、MaxCount、DistinctKey、IQ Unique
HG	Type、Version、Maintains Exact Distinct、Level 0 Threshold、Force Physical Delete、Maximum Level Count、Tier ratio、Auto sizing、Average Load Size (records)、Active Subindex count、Cardinality Range Min - Max、Estimated Cardinality、Accuracy of Cardinality
HNG	Type、Version、BitsPerBlockmap、NumberOfBits
LF	Type、Version、IndexStatus、NumberOfBlockmaps、BitsPerBlockmap、Distinct Keys
WD	Type、Version、KeySize、Delimiters、DelimiterCount、MaxKeywordLength、PermitEmptyWord

例

Value1	Value2	Value3
DBA	test2	test2_c1_hg
Type	HG	
Version	3	

Value1	Value2	Value3
Maintains Exact Distinct	No	
Level 0 Threshold	3000000	
Force Physical Delete	Yes	
Maximum Level Count	10	
Tier ratio	30	
Auto sizing	On	
Avarage Load Size (records)	58622	
Active Subindex count	3	
Cardinality Range Min - Max	5-5	
Estimated Cardinality	5	
Accuracy of Cardinality	100	
レベル :0 メインインデックス 合計ローカウント	1	
レベル :0 メインインデックス 削除済ローカウント	0	
レベル :0 メインインデックス メイン btree における btree ページの数	1	
レベル :0 メインインデックス garray ページの数	1	
レベル :0 メインインデックス メイン btree におけるキーの数	1	
レベル :0 メインインデックス btree で調査されるキーの数	0	
レベル :0 メインインデックス btree で重複が見つかったキーの数	0	
レベル :0 メインインデックス btree で可能な明確なキーの数	0	
レベル :1 メインインデックス 合計ローカウント	3145747	
レベル :1 メインインデックス 削除済ローカウント	0	
レベル :1 メインインデックス メイン btree における btree ページの数	1	
レベル :1 メインインデックス 共役 btree における btree ページの数	1	
レベル :1 メインインデックス garray ページの数	2	

Value1	Value2	Value3
レベル:1 メインインデックス メイン btree におけるキーの数	8	
レベル:1 メインインデックス 共役 btree におけるキーの数	3	
レベル:1 メインインデックス 共役 btree におけるローの数	2949127	
レベル:1 メインインデックス btree で調査されるキーの数	0	
レベル:1 メインインデックス btree で重複が見つかったキーの数	0	
レベル:1 メインインデックス btree で可能な明確なキーの数	0	
レベル:1 インクリメンタルインデックス 合計ローカウント	1	
レベル:1 インクリメンタルインデックス 削除済ローカウント	0	
レベル:1 インクリメンタルインデックス btree ページの数	1	
レベル:1 インクリメンタルインデックス garray ページの数	1	
レベル:1 インクリメンタルインデックス btree におけるキーの数	1	
レベル:1 インクリメンタルインデックス btree で調査されるキーの数	0	
レベル:1 インクリメンタルインデックス btree で重複が見つかったキーの数	0	
レベル:1 インクリメンタルインデックス btree で可能な明確なキーの数	0	

参照：

- sp_iqindexfragmentation プロシージャ (514 ページ)
- sp_iqindexinfo プロシージャ (516 ページ)
- sp_iqindexsize プロシージャ (522 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqindexrebuildwidedata プロシージャ

事前に再構築しないと読み込み／書き込みアクティビティに利用できない、移行されたデータベース内の幅の広いカラムを識別します。

構文

sp_iqindexrebuildwidedata [*table.name*]

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- INSERT ANY TABLE システム権限
- インデックスを再構築するテーブルに対する INSERT 権限
- テーブルを所有している

備考

SAP Sybase IQ 16.0 に移行されたデータベース内のすべての Long Varchar カラムと Long Binary カラム、および 255 文字を超える CHAR、VARCHAR、BINARY、VARBINARY の各カラムに対してデータベースエンジンが読み込み／書き込みアクティビティを実行できるようにするには、これらのカラムを再構築する必要があります。sp_iqindexrebuildwidedata は、これらのカラムを識別し、sp_iqrebuildindex プロシージャでカラムを再構築するために使用可能な文のリストを生成します。

オプションの [table.name] パラメータを指定すると、そのテーブルについて、幅の広いカラムのリストが生成されます。[table.name] パラメータを省略すると、データベース内のすべてのテーブルについて、幅の広いカラムのリストが生成されません。

例

テーブル T2 について、幅の広いカラムの再構築文を生成します。

```
sp_iqindexrebuildwidedata T2
```

出力：

Owner	Table	Column	Domain	Width	IndexType	sp_iqrebuild
DBA	T2	C1	char	1020	Long varchar FP	sp_iqrebuildindex 'DBA.T2' 'column
"C1"	0';					

参照：

- sp_iqrebuildindex プロシージャ (561 ページ)

sp_iqindexsize プロシージャ

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
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- `MANAGE ANY DBSPACE`
- `ALTER ANY INDEX`
- テーブルを所有している

備考

表 88 : `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る。たとえば、デフォルト (FP) インデックスには、 <code>BARRAY</code> (<code>barray</code>) と <code>Bitmap</code> (<code>bm</code>) コンポーネントが含まれる。Low_Fast (LF) インデックスには、 <code>B-tree</code> (<code>bt</code>) と <code>Bitmap</code> (<code>bm</code>) コンポーネントが含まれる。
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。単一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_T780_I4_HG
```

Username	Indexname	Type	Info	Kbytes	Pages	Compressed Pages
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	Total	288	4	2
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	vdo	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	bt	152	2	1
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	garray	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	bm	136	2	1
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	barray	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	dpstore	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	largelob	0	0	0
GROUPO	GROUPO.Departments.ASIQ_IDX_T780_I4_HG	HG	txtPst	0	0	0

```
CREATE TEXT INDEX ti ON Employees( Street ) IMMEDIATE
REFRESH;sp_iqindexsize 'ti';
```

Username	Indexname	Type	Info	KBytes	Pages	Compressed Pages
GROUPO	GROUPO.Employees.ti	TEXT	Total	896	12	6

Username	Indexname	Type	Info	KBytes	Pages	Compressed Pages
GROUPO	GROUPO.Employees.ti	TEXT	vdo	0	0	0
GROUPO	GROUPO.Employees.ti	TEXT	bt	304	4	2
GROUPO	GROUPO.Employees.ti	TEXT	garray	152	2	1
GROUPO	GROUPO.Employees.ti	TEXT	bm	136	2	1
GROUPO	GROUPO.Employees.ti	TEXT	barray	152	2	1
GROUPO	GROUPO.Employees.ti	TEXT	dpstore	0	0	0
GROUPO	GROUPO.Employees.ti	TEXT	largelob	0	0	0
GROUPO	GROUPO.Employees.ti	TEXT	txtPst	304	4	2

参照：

- sp_iqindexmetadata プロシージャ (518 ページ)
- sp_iqindexfragmentation プロシージャ (514 ページ)
- sp_iqindexinfo プロシージャ (516 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqindexuse プロシージャ

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

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。 MONITOR システム権限。

備考

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、オプティマイザ、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 89 : 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：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)
- sp_iqworkmon プロシージャ (637 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqlmconfig プロシージャ

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 90 : "edition" パラメータの要約情報

トピック	値
デフォルト値	'EE' (Enterprise Edition)
値の範囲	'SE' (Small Business) 'SA' (Single Application) 'EE' (Enterprise Edition)
ステータス	静的

構文 2

```
sp_iqlmconfig 'license type', { 'CP' | 'DT' | 'SF' | 'AC' | 'BC' | 'CH' | 'DH' | 'SH' | 'AH' | 'BH' }
```

表 91 : "license type" パラメータの要約情報

トピック	値
デフォルト値	'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)

トピック	値
ステータス	静的

構文 3

```
sp_iqlmconfig 'email severity', { 'ERROR' | 'WARNING' | 'INFORMATIONAL' | 'NONE' }
```

NONE は、電子メール通知の無効を指定します。

構文 4

```
sp_iqlmconfig 'smtp host', '<host name>'
```

表 92 : パラメータ

パラメータ	説明
host name	電子メール通知に使用する SMTP ホストを指定する。

構文 5

```
sp_iqlmconfig 'email sender', '<email address>'
```

表 93 : パラメータ

パラメータ	説明
<email address>	電子メール通知に使用する送信者の電子メールアドレスを指定する。

構文 6

```
sp_iqlmconfig 'email recipients', '<email recipients>'
```

表 94 : パラメータ

パラメータ	説明
<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する。

構文 7

```
sp_iqlmconfig
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが必要です。 SERVER OPERATOR システム権限。

備考

起動時に、**sp_iqlmconfig** 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。

- 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。
- **sp_iqlmconfig** 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。
 - 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

参照：

- サーバで使用可能なプロパティ (105 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqlocks プロシージャ

IQ メインストアと IQ カタログストアの両方について、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

```
sp_iqlocks ([connection,] [[owner.]table_name,] max_locks,]
[sort_order])
```

パラメータ

結果を絞り込むためのすべてのパラメータはオプションです。

パラメータ	データ型	説明
connection	integer	接続 ID。このオプションを指定すると、指定された接続のロックに関する情報のみが返される。デフォルトはゼロ (すべての接続についての情報を返す)。

パラメータ	データ型	説明
owner.table_name	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ついての情報のみが返される。デフォルトは NULL (データベース内のすべてのテーブルについての情報を返す)。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る。
max_locks	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す)。
sort_order	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

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR システム権限。

備考

データベース内の現在のロックについての情報を表示します。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。

カラム	データ型	説明
conn_name	VARCHAR(128)	現在の接続の名前。
conn_id	INTEGER	ロックが存在する接続 ID。
user_id	CHAR(128)	この接続 ID に関連付けられているユーザ。
table_type	CHAR(6)	テーブルのタイプ。タイプは、テーブルの場合は BASE、グローバルテンプラリテーブルの場合は GLBTMP、マテリアライズドビューの場合は MVIEW。 マテリアライズドビューは、IQ カタログストアの SQL Anywhere テーブルでのみサポートされる。
creator	VARCHAR(128)	テーブルの所有者。

カラム	データ型	説明
table_name	VARCHAR(128)	ロックが存在するテーブル。
index_id	INTEGER	インデックス ID または NULL。
lock_class	CHAR(8)	ロッククラス。Schema、Row、Table、Position のいずれか。
lock_duration	CHAR(11)	ロックの期間。Transaction、Position、Connection のいずれか。
lock_type	CHAR(9)	ロックタイプ (ロッククラスに依存する)。
row_identifier	UNSIGNED BIGINT	ロックが開始されるローの識別子、または NULL。
row_range	BIGINT	ロックされた連続するローの数。RLV ストア内のローロックは、単一のローまたは一連のローのいずれかになる。

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。

lock_type カラムの値は、lock_class カラムのロック分類によって変わります。次の値が返されます。

ロッククラス	ロックタイプ	コメント
Schema	<ul style="list-style-type: none"> • Shared – 共有スキーマロック。 • Exclusive – (IQ カタログストア テーブルのみ) 排他スキーマロック。 	スキーマロックの場合、row_identifier とインデックス ID の値は NULL。

ロック クラス	ロックタイプ	コメント
Row	<ul style="list-style-type: none"> • Read – 読み込みロック。 • Intent – 意図的ロック。 • ReadPK – 読み込みロック。 • Write – 書き込みロック。 • WriteNoPK – 書き込みロック。 • Surrogate – 代理ロック。 	<p>ローの読み取りロックは、短期のロック (独立性レベル 1 でスキャン) にするか、高い独立性レベルの長期ロックにすることができる。lock_duration カラムは、読み取りロックがカーソル安定性 (Position) のために短期になるか、COMMIT/ROLLBACK (Transaction) まで保持される長期になるかを指定する。ローのロックは、常に特定のローに保持される。この 8 バイトのロー識別子は、row_identifier カラムの 64 ビット整数値としてレポートされる。</p> <p>代理ロックは、ローロックの特殊なケースである。代理ロックは代理エントリで保持される。代理エントリは、参照整合性のチェックが遅延するときに作成される。テーブルで作成されるすべての代理エントリについてユニークな代理ロックはない。ただし、代理ロックは、指定した接続で指定されるテーブルで作成される代理エントリのセットに対応する。row_identifier 値は、代理ロックに関連付けられているテーブルと接続についてユニークである。</p> <p>必要に応じて、ローのキーとキーではない部分を別々にロックできる。1 つの接続で共有 (読み取り) アクセス用にローのキー部分に対する読み取りロックを取得して、他の接続でローのキーではない他のカラムに対する書き込みロックを取得できるようにすることができる。ローのキーではないカラムを更新しても、そのローを参照する外部ローの挿入と削除は妨げられない。</p>

ロッククラス	ロックタイプ	コメント
Table	<ul style="list-style-type: none"> • Shared – 共有テーブルロック。 • Intent – 意図的にテーブルロックを更新。 • Exclusive – (IQ カタログストアテーブルのみ) 排他テーブルロック。 	なし
Position	<ul style="list-style-type: none"> • Phantom – (IQ カタログストアテーブルのみ) 幻ロック。 • Insert – 挿入ロック。 	通常、位置ロックも特定のローに保持され、そのローの 64 ビットロー識別子は結果セットの row_identifier カラムに表示される。ただし、row_identifier カラムが NULL の場合、位置ロックはスキャン (インデックススキャンまたは逐次スキャン) 全体に保持される。

注意： 排他ロック、幻ロック、幻でないロックは、IQ カタログストアのテーブルでは使用できますが、IQ メインストアの SAP 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かぎり、これらの種類のロックが SAP Sybase IQ データベースで起きることはありません。

例

次の例は、**sp_iqlocks** プロシージャコールと、SAP 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
=====
=====
SQL_DBC_13cd6038 3          DBA      BASE      DBA      rv_locks2
SQL_DBC_13cd6038 3          DBA      BASE      DBA      rv_locks2
SQL_DBC_13cd6038 3          DBA      BASE      DBA      rv_locks2
RVL_CONN_T775    1000000407 BASE     DBA      rv_locks2

index_id lock_class lock_duration lock_type row_identifier
row_range
=====
=====
Schema Transaction Shared
Row Transaction Row 1 4
```

Row	Transaction	Row	281474976710656	1
Table	Transaction	Intent		

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmergerlvstore プロシージャ

ローレベルのバージョン管理 (RLV) ストアと IQ メインストアとのマージをトリガします。

構文

```
sp_iqmergerlvstore 'merge_type', 'table_name', [ 'table_owner' ]
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

- マージタイプには BLOCKING または NON-BLOCKING を指定できます。
- マージの実行後は、ストアドプロシージャによってマージトランザクションが自動的にコミットされます。

sp_iqmodifyadmin プロシージャ

指定した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 '
```

パラメータ

- **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** – ログインポリシーオプションが変更されるポリシー。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが必要です。MANAGE ANY LOGIN POLICY システム権限。

例

lockeduser という名前のポリシーで、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 という名前のマルチプレックスサーバ上の *lockeduser* というポリシーで、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')
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmodifylogin プロシージャ

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']
```

パラメータ

- **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。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが必要です。MANAGE ANY USER システム権限。

例

ユーザ joe を expired_password というログインポリシーに割り当てます。

```
sp_iqmodifylogin 'joe', 'expired_password'
```

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_expireallpasswords システムプロシージャ (721 ページ)
- sp_iqaddlogin プロシージャ (437 ページ)
- sp_iqcopyloginpolicy プロシージャ (469 ページ)
- sp_iqpassword プロシージャ (551 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmpxcheckdqpconfig プロシージャ

sp_iqmpxcheckdqpconfig は、現在の接続の DQP 設定をチェックする診断ツールです。DQP に失敗した場合は、**sp_iqmpxcheckdqpconfig** を実行して、DQP 設定の問題によってクエリ分散エラーが発生しているかどうかを調べます。

構文

```
sp_iqmpxcheckdqpconfig
```

適用対象

マルチプレックスのみ。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

表 95 : カラムの説明

カラム名	説明
DiagMsgID	診断メッセージの一意の ID
Description	見つかった DQP 設定の問題を説明する診断メッセージ

表 96 : 診断メッセージ

DiagMsgID	説明
0	DQP の設定で問題は検出されませんでした
1	データベースはシンプレックスです
2	Multiplex がシングルノード設定モードで実行されています
3	論理サーバポリシーオプション dqp_enabled が 0 に設定されています
4	テンポラリ dqp_enabled 接続オプションが OFF に設定されています
5	論理サーバのコンテキストにメンバーノードが 1 つしかありません
6	論理サーバで指定されたコーディネータメンバーシップが現在無効なため、コーディネータは DQP に加えられていません
7	ルート論理サーバのポリシーで ALLOW_COORDINATOR_AS_MEMBER オプションが OFF に設定されているため、論理サーバでのコーディネータメンバーシップが現在無効となっており、コーディネータは DQP に加えられていません
8	IQ_SHARED_TEMP DB 領域に DB ファイルがありません
9	IQ_SHARED_TEMP DB 領域のすべての DB ファイルが READ ONLY です
10	IQ_SHARED_TEMP DB 領域は動的にオフラインです

例

sp_iqmpxcheckdqpconfig プロシージャからの出力例を示します。

```
diagmsgid      description
3              Logical server policy option dqp_enabled is set to 0
5              Logical server context has only one member node
6              Coordinator does not participate in DQP since its
               named membership in the logical server is
               currently ineffective
7              Coordinator does not participate in DQP since
               its logical membership in the logical server
```

```

8      is currently ineffective because
      ALLOW_COORDINATOR_AS_MEMBER option in Root
      Logical server policy set to OFF
      There is no dbfile in IQ_SHARED_TEMP dbspace
    
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmpxdumptlvlog プロシージャ

テーブルバージョンログの内容を読みやすい形式で返します。

構文

```

sp_iqmpxdumptlvlog
[main], [asc | desc]
    
```

適用対象

マルチプレックスのみ。

権限

MANAGE MULTIPLEX システム権限。そのシステムプロシージャに対するEXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqmpxdumptlvlog は、コーディネータが DML および DDL コマンドをセカンダリノードに送信するのに用いるキューの内容を返します。

引数 **asc** または **desc** によってローの順序を指定できます。これらの引数を指定する場合は **main** 引数が必要です。デフォルトのオプションは次のとおりです。

```

'main', 'asc'.
    
```

例

sp_iqmpxdumptlvlog の出力を表示します。

RowID	Contents
1	Txn CatId:196 CmtId:196 TxnId:195 Last Rec:1 UpdateTime: 2011-08-08 15:41:43.621
2	Txn CatId:243 CmtId:243 TxnId:242 Last Rec:5 UpdateTime: 2011-08-08 15:42:25.070
3	DDL: Type=34, CatID=0, IdxID=0, Object=IQ_SYSTEM_TEMP, Owner=mpx4022_w1
4	CONN: CatID=0, ConnUser=
5	SQL: ALTER DBSPACE "IQ_SYSTEM_TEMP" ADD FILE "w1 templ" '/dev/raw/raw25' FILE ID 16391 PREFIX 65536 FINISH 0 FIRST BLOCK
1	BLOCK COUNT 3276792 RESERVE 0 MULTIPLEX SERVER "mpx4022_w1" COMMITID 242 CREATETIME '2011-08-08 15:42:24.860'


```

6 Txn CatId:283 CmtId:283 TxnId:282 Last Rec:7
  UpdateTime: 2011-08-08 15:42:50.827
7 RFRB TxnID: 242 CmtID:243 ServerID 0 BlkmapID:
  0d00000000000000d2000a0000000000200000000000000000
  0000000000000000000000008003501010000000c38000000000000
  01000000000000000000000000RFID:01000501000000001300000
  0000000000100000000000100RBID:010005010000000013000

```

sp_iqmpxfilestatus プロシージャ

コーディネータノードで実行した場合は、コーディネータのファイルステータスと、含まれている各セカンダリノード上のすべての共有 DB 領域ファイルのファイルステータスが表示されます。セカンダリノードで実行した場合は、現在のノードのファイルステータスのみが表示されます。

構文

```
sp_iqmpxfilestatus
```

適用対象

マルチプレックスのみ。

権限

MANAGE MULTIPLEX システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqmpxfilestatus の戻り値。

カラム名	データ型	説明
server_id	unsigned int	SYSIQMPXINFO からの、マルチプレックスサーバの識別子
dbspace_name	char(128)	領域の予約元の DB 領域。
FileName	char(128)	DB 領域ファイルの論理ファイル名。
FileStatus	char(2)	DB 領域ファイルのステータス。 <ul style="list-style-type: none"> VALID – ファイルパスと権限は正しい。 INVALID_PATH – パス名はアクセス不能。 INVALID_PERM – ファイル権限が正しくない。

例

次に sp_iqmpxfilestatus の出力例を示します。

```

server_id,server_name,DBSpace_name,FileName,FileStatus
1,'mpx2422_m','IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','VALID'

```

```

1, 'mpx2422_m', 'mpx_main1', 'mpx_main1', 'VALID'
1, 'mpx2422_m', 'IQ_SHARED_TEMP', 'sharedfile_dba', 'VALID'
1, 'mpx2422_m', 'IQ_SHARED_TEMP', 'sharedfile_dba1', 'VALID'
2, 'mpx2422_w1', 'IQ_SYSTEM_MAIN', 'IQ_SYSTEM_MAIN', 'VALID'
2, 'mpx2422_w1', 'mpx_main1', 'mpx_main1', 'VALID'
2, 'mpx2422_w1', 'IQ_SHARED_TEMP', 'sharedfile_dba', 'VALID'
2, 'mpx2422_w1', 'IQ_SHARED_TEMP', 'sharedfile_dba1', 'VALID'
3, 'mpx2422_r1', 'IQ_SYSTEM_MAIN', 'IQ_SYSTEM_MAIN', 'VALID'
3, 'mpx2422_r1', 'mpx_main1', 'mpx_main1', 'VALID'
3, 'mpx2422_r1', 'IQ_SHARED_TEMP', 'sharedfile_dba', 'VALID'
3, 'mpx2422_r1', 'IQ_SHARED_TEMP', 'sharedfile_dba1', 'VALID'

```

sp_iqmpxinconnpoolinfo プロシージャ

コーディネータノードで実行した場合、各ノードの INC 接続プールステータスが表示されます。セカンダリノードで実行した場合、現在のノードのみの INC 接続プールステータスが表示されます。

構文

```
sp_iqmpxinconnpoolinfo
```

適用対象

マルチプレックスのみ。

権限

MANAGE MULTIPLEX システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

このプロシージャをコーディネータノードで実行したときに、セカンダリノードが応答しない場合やタイムアウトになった場合は、そのノードのローは結果セットには含まれません。これは、そのノードが実行されていないとこのデータへのアクセスができないためです。

sp_iqmpxinconnpoolinfo プロシージャは、次の出力を返します。

カラム名	データ型	説明
server_id	unsigned int	サーバの識別子
server_name	char(128)	サーバの名前
current_pool_size	unsigned int	接続プールの現在のサイズ
idle_connection_count	unsigned int	プール内でアイドル中の接続数
connections_in_use	unsigned int	使用中の接続の数

例

次に **sp_iqmpxinconnpoolinfo** の出力例を示します。

```
server_id,server_name,current_pool_size,
idle_connection_count,connections_in_use

2,'r2_dbsrv90210',0,0,0
3,'w3_dbsrv90210',0,0,0
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmpxinheartbeatinfo プロシージャ

コーディネータノードで実行した場合、各ノードの INC ハートビートステータスが表示されます。セカンダリノードで実行した場合、現在のノードのみの INC ハートビートステータスが表示されます。

構文

```
sp_iqmpxinheartbeatinfo
```

適用対象

マルチプレックスのみ。

権限

MANAGE MULTIPLEX システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqmpxinheartbeatinfo プロシージャは、次の出力を返します。

カラム名	データ型	説明	値
server_id	unsigned int	サーバの識別子	
server_name	char(128)	サーバの名前	
last_positive_hb	TIMESTAMP	正常に実行された最後のハートビート ping の日付/時刻	DD:MM: YYYY:HH: MM:SS
time_not_responding	TIME	正常に実行された最後のハートビート ping からの経過時間	HH:MM:SS

カラム名	データ型	説明	値
time_until_timeout	TIME	ノードが応答していない場合に、ノードのオフラインが宣言されるまでの残り時間	

例

次に **sp_iqmpxinheartbeatinfo** の出力例を示します。

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_dbsrv90210',2012-11-17
15:48:42.0,00:00:00,00:00:00
3,'w3_dbsrv90210',2012-11-17
15:48:42.0,00:00:00,00:00:00
```

経過時間が 24 時間を超えると、SAP Sybase IQ は次のような **sp_iqmpxinheartbeatinfo** 出力を返します。

```
server_id,server_name,last_positive_hb,
time_not_responding,time_until_timeout
2,'r2_mpx_cr_srv',Jan 14 2013 11:57AM,11:59PM,11:59PM
3,'w4_mpx_cr_srv',Jan 14 2013
11:57AM,11:59PM,11:59PM
(2 rows affected)
(return status = 0)
```

time_not_responding カラムと time_until_timeout カラムの 11:59PM という値は、時間が 24 時間の制限を超えたことを意味します。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmpxincstatistics プロシージャ

サーバ起動後から実行時までのノード間通信 (INC) ステータスの累計統計のスナップショットを表示します。

構文

```
sp_iqmpxincstatistics
```

適用対象

マルチプレックスのみ。

権限

MANAGE ANY STATISTICS システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考
戻り値

表 97 : sp_iqmpxincstatistics カラム

カラム名	データ型	説明
stat_name	char(128)	INC 統計名
stat_value	符号なし整数	INC 統計値

表 98 : 有効な stat_name 値

値	説明
NumSuspendedINC	サーバ起動後、サスペンドされた INC 接続の数
NumResumedINC	サーバ起動後、再開された INC 接続の数
NumDroppedSuspendedINC	サスペンドされていた接続のうち削除された INC 接続の数 (コーディネータ上のみ)
NumSuspendedTxnRollback-INC	INC の障害によりロールバックされたグローバル DML トランザクションの数 (ライター上のみ)

例

サスペンドされたトランザクションと再開されたトランザクションを 1 つずつ示します。

sp_iqmpxincstatistics

```
stat_name          stat_value
NumSuspendedINC   1
NumResumedINC     1
NumSuspendedTXNRollBackINC  0
```

sp_iqmpxinfo プロシージャ

マルチプレックス内のすべてのノードのローを返します。任意のマルチプレックスノードから実行できます。

構文

```
sp_iqmpxinfo
```

適用対象

マルチプレックスのみ。

権限

MANAGE MULTIPLEX システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqmpxinfo プロシージャは、次の出力を返します。

カラム名	データ型	説明
server_id	unsigned int	情報が表示されるサーバの識別子
server_name	char(128)	サーバの名前
connection_info	long varchar	マルチプレックスサーバ間の TCP/IP 接続に使用される接続文字列のホスト/ポート部分を含むフォーマット済みの文字列
db_path	long varchar	データベースへのフルパス
role	char(16)	'coordinator' 'writer' 'reader'
status	char(8)	'included' 'excluded'
mpx_mode	char(16)	'single' 'coordinator' 'writer' 'reader' 'unknown'
inc_state	char(16)	'active' 'not responding' 'timed out'
coordinator_failover	char(128)	フェールオーバーサーバの名前
current_version	unsigned bigint	10 進数形式のバージョン ID
active_versions	long varchar	10 進数形式のバージョン ID のカンマ区切りリスト
private_connection_info	long varchar	マルチプレックスサーバ間のプライベート TCP/IP 接続に使用される接続文字列のホスト/ポート部分を含むフォーマット済みの文字列
mipc_priv_state	char(16)	'active' - このノードへの MIPC 接続は、プライベート相互接続上でアクティブである。 'not responding' - このノードへの MIPC 接続は、プライベート相互接続上で応答しない。
mipc_public_state	char(16)	'active' - このノードへの MIPC 接続は、パブリック相互接続上でアクティブである。 'not responding' - このノードへの MIPC 接続は、パブリック相互接続上で応答しない。

例

次に `sp_iqmpxinfo` の出力例を示します。

```
server_id,server_name,connection_info,db_path,role,
status,mpx_mode,inc_state,coordinator_failover,
current_version,active_versions,private_connection_
info,mipc_priv_state,mipc_public_state

1,'my_mpx1','host=(fe80::214:4fff:fe45:be26%2):1362
0,(fd77:55d:59d9:329:214:4fff:fe45:be2
6%2):13620,10.18.41.196:13620','/system3/users
/devices/s16900269/iqmpx1/mpx1.db',
'coordinator','included','coordinator','N/A',
'my_mpx2',0,,,'active','active'

2,'IQ_mpx2','host=system3:13625',
'/system3/users/devices/s16900269
/iqmpx_2/wk0001.db','writer','included',
'writer','active','IQ_mpx20','not responding','active'

3,'IQ_mpx3','host=system3:13630/system3/users/devi
ces/s16900269/iqmpx_3/mpx1.db','reader','included',
'unknown',timed out',
'IQ_mpx20','not responding',
'not responding'
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqmpxsuspendedconninfo プロシージャ

コーディネータノードで現在サスペンドされている接続およびトランザクションの詳細を表示します。

構文

sp_iqmpxsuspendedconninfo

適用対象

マルチプレックスのみ。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。サスペンド中の自分の接続を表示する場合、システム権限は必要ありません。また、次のものも必要です。データベース内のサスペンド中の接続をすべて表示する場合は、次のいずれかのシステム権限が必要です。

- DROP CONNECTION
- MONITOR

- SERVER OPERATOR

備考
戻り値

表 99 : sp_iqmpxsuspendedconninfo カラム

カラム名	データ型	説明
ConnName	char(128)	接続名
ConnHandle	符号なし整数	接続識別子
GlobalTxnID	unsigned bigint	この接続のアクティブなトランザクションのグローバルトランザクション識別子
MPXServerName	char(128)	INC 接続を開始したマルチプレックスサーバの名前
TimeInSuspended-State	integer	接続がサスペンド状態であった時間の合計 (秒数)
SuspendTimeout	integer	サスペンドのタイムアウト (秒数) (2*MPX_LIVENESS_TIMEOUT)

例

sp_iqmpxsuspendedconninfo

```

ConnName          ConnHandle      GlobalTxnId
=====          =====
'IQ_MPX_SERVER_P54'          14          112753

MPXServerName     TimeInSuspendedState
=====          =====
'HP1_12356_IQ_mpx2'          37

SuspendTimeout
=====
360
    
```

sp_iqmpxvalidate プロシージャ

マルチプレックス設定の矛盾をチェックします。

構文

```
call dbo.sp_iqmpxvalidate ( 'show_msgs' )
```

適用対象

マルチプレックスのみ。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

テーブル SYS.SYSIQDBFILE、他のマルチプレックスイベントとストアドプロシージャで複数のチェックを実行します。どのサーバでも実行可能です。呼び出し元に、重大度を示す結果を次の値で返します。

値	説明
0	エラーは検出されていません
1	予想外の動的な状態
2	致命的でない設定エラー (マルチプレックス操作の障害など)
3	致命的な設定の問題 (1 つまたは複数のサーバが起動しないおそれがあるなど)

また、対話的に呼び出した場合は、呼び出し時のパラメータが 'Y' でなかった場合を除いて、エラーが検出されたときにそのテーブルを返します。

各エラーは、その重大度を表します。エラーがなかった場合、プロシージャは No errors detected を返します。

sp_iqmpxversioninfo プロシージャ

このサーバの現在のバージョン情報を表示します。この情報には、サーバタイプ (ライターサーバ、クエリサーバ、シングルノードモード) と同期ステータスが含まれます。

構文

```
sp_iqmpxversioninfo
```

適用対象

マルチプレックスのみ。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

表 100 : `sp_iqmpxversioninfo` のカラム

カラム	データ型	説明
CatalogID	unsigned bigint	このサーバのカタログバージョン。
VersionID	unsigned bigint	このサーバで利用可能な最新バージョン。
OAVID	unsigned bigint	このサーバ上の最も古いバージョン。
ServerType	char(1)	サーバの種類。"C" (コーディネータ)、"W" (ライターサーバ)、"Q" (クエリサーバ)。
CatalogSync	char(1)	カタログ同期。"T" (同期)、"F" (非同期)。
WCatalogID	unsigned bigint	ライターサーバ上のカタログバージョン。
WVersionID	unsigned bigint	ライターサーバ上で利用可能な最新バージョン。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqobjectinfo プロシージャ

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 ]
```

パラメータ

- owner_name** – オブジェクトの所有者。指定した場合、**sp_iqobjectinfo** は、指定の所有者のテーブルのみの出力を表示します。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に関する情報を表示します。
- object_name** – テーブルの名前。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に関する情報を表示します。
- object-type** – 有効な **table** オブジェクトタイプ。

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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

入力パラメータは **sp_iqobjectinfo** とともに使用します。**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 を記述し直して、**sp_iqobjectinfo** 入力パラメータを使用できるようにしたものです。

```
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 よりも短時間で結果を返します。入力パラメータが **sp_iqobjectinfo** に渡されると、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せん。クエリ B では、述部はプロシージャ自体に適用され、プロシージャで返される結果セットが小さくなります。そのため、クエリで適用される述部の数が少なくなります。

sp_iqobjectinfo ストアドプロシージャでは、*owner_name*、*object_name*、および *object_type* の解釈に、ワイルドカード文字がサポートされています。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。

特定のまたはすべてのデータベースオブジェクト (テーブルタイプ) 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 101 : sp_iqobjectinfo のカラム

カラム名	説明
owner	オブジェクトの所有者の名前。

カラム名	説明
object_name	DB 領域にあるオブジェクト (テーブルタイプ) の名前。
sub_object_name	DB 領域に存在するオブジェクトの名前。
object_type	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、テーブル)。
object_id	オブジェクトのグローバルオブジェクト ID。
id	オブジェクトのテーブル ID。
dbspace_name	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に表示される。出力内の [multiple] ローは、その後ろにテーブルまたはカラムを記述する複数のローが続くことを示す。
partition_name	指定のオブジェクトのパーティションの名前。

例

注意: 出力内容をわかりやすくするため、次の例は iqdemo データベース内のオブジェクトを示しています。iqdemo には iq_main というサンプルのユーザ DB 領域が含まれていますが、この領域はユーザ独自のデータベースには存在しない場合があります。

指定のユーザが所有す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  obj
ect_id     id
GROUPO    Departments  (NULL)               table        3
632       738
GROUPO    Departments  DepartmentID         column       3
633       738
GROUPO    Departments  DepartmentName       column       3
634       738
GROUPO    Departments  DepartmentHeadID     column       3
635       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 で、指定のユーザが所有す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)

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqpassword プロシージャ

ユーザのパスワードを変更します。

構文 1

```
call sp_iqpassword ('caller_password', 'new_password' [, 'user_name'])
```

構文 2

```
sp_iqpassword 'caller_password', 'new_password' [, 'user_name']
```

パラメータ

- **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。CHANGE PASSWORD システム権限を持つユーザが別のユーザのパスワードを変更する場合は、caller_password には変更を実行するユーザのパスワードを指定する。
- **new_password** – ユーザまたは *loginname* の新しいパスワード。
- **user_name** – CHANGE PASSWORD システム権限を持つ別のユーザによってパスワードが変更され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 を指定しない。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。各自のパスワードを設定する際に要求される追加のシステム権限はありません。他のユーザのパスワードを変更する場合は、CHANGE PASSWORD システム権限が必要です。

備考

ユーザパスワードは識別子です。すべてのユーザが `sp_iqpassword` を使用して自分のパスワードを変更できます。既存のユーザのパスワードを変更するには、CHANGE PASSWORD システム権限が必要です。

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。
- 識別子の先頭文字がアルファベット文字ではない (以下を参照)。
- 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。

例

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。

```
sp_iqpassword 'irk103', 'exP984'
```

ログインしたユーザが、`joe` に対する CHANGE PASSWORD システム権限を持っている場合に、ユーザ `joe` のパスワードを `epr45` から `pdi032` に変更します。

```
call sp_iqpassword ('epr45', 'pdi932', 'joe')
```

参照：

- `sp_expireallpasswords` システムプロシージャ (721 ページ)
- `sp_iqaddlogin` プロシージャ (437 ページ)
- `sp_iqcopyloginpolicy` プロシージャ (469 ページ)
- `sp_iqmodifylogin` プロシージャ (535 ページ)

sp_iqpkeys プロシージャ

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 SAP Sybase IQ テーブルについて表示します。

構文

```
sp_iqpkeys { [ table-name ], [ column-name ], [ table-owner ] }
```

パラメータ

- **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。
- **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。
- **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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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。

表 102 : 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する。
sp_iqpkeys sales, NULL, DBA	DBA が所有するテーブル sales 上で定義された、プライマリキーに関する情報を表示する。
sp_iqpkeys sales, store_id, DBA	DBA が所有するテーブル sales のカラム store_id 上で定義された、プライマリキーに関する情報を表示する。

構文	出力
sp_iqpkeys NULL, NULL, DBA	DBA が所有す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する。

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 103 : 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。

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_iqpkeys 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_iqcolumn プロシージャ (457 ページ)

- `sp_iqconstraint` プロシージャ (465 ページ)
- `sp_iqdatatype` プロシージャ (473 ページ)
- `sp_iqevent` プロシージャ (496 ページ)
- `sp_iqhelp` プロシージャ (502 ページ)
- `sp_iqindex` および `sp_iqindex_alt` プロシージャ (509 ページ)
- `sp_iqprocparm` プロシージャ (558 ページ)
- `sp_iq_reset_identity` プロシージャ (570 ページ)
- `sp_iqtable` プロシージャ (614 ページ)
- `sp_iqview` プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

`sp_iqprocedure` プロシージャ

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 ]
```

パラメータ

- **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` が所有するプロシージャ) 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、ユーザ定義プロシージャ (`dbo` または `sys` が所有し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。

権限

そのシステムプロシージャに対する `EXECUTE` 権限が必要です。

備考

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` および `sp_iqprocedure NULL, user1` とします。

表 104 : `sp_iqprocedure` の使用例

構文	出力
<code>sp_iqprocedure</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プロシージャに関する情報を表示する。
<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する。
<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、ローは返されない。
<code>sp_iqprocedure NULL, DBA</code>	DBA が所有するすべてのプロシージャに関する情報を表示する。
<code>sp_iqprocedure sp_test, DBA</code>	DBA が所有するプロシージャ <code>sp_test</code> に関する情報を表示する。
<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ない。 <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、ローは返されない(デフォルトでは、ユーザ定義プロシージャのみが返される)。
<code>sp_iqprocedure sp_iqtable, dbo</code>	プロシージャ <code>sp_iqtable</code> はユーザプロシージャではないため、ローは返されない(デフォルトではユーザプロシージャのみが返される)。
<code>sp_iqprocedure NULL, NULL, SYSTEM</code>	(dbo または SYS が所有する) すべてのシステムプロシージャに関する情報を表示する。
<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する。
<code>sp_iqprocedure sp_iqtable, dbo, ALL</code>	dbo が所有するシステムプロシージャ <code>sp_iqtable</code> に関する情報を表示する。

`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って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` は指定の所有者が所有す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 105 : 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' が表示される。
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 が表示される。
srv_id	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      srv_id      r
emarks
sp_test        DBA              create procedure N              (NULL)      (NULL)
                DBA.sp_test(in n1
                integer)
                begin message 'sp_test' end
```

DBA が所有す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      srv_id      r
emarks
sp_test        DBA              create procedure N              (NULL)      (NULL)
                DBA.sp_test(in n1
                integer)
                begin message 'sp_test' end
sp_dept        DBA              create procedure N              (NULL)      (NULL)
                DBA.sp_dept() begin end
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqprocparm プロシージャ

ストアー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 ]
```

パラメータ

- **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 が所有するプロシージャ) 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 が所有し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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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 および sp_iqprocparm NULL, user1 とします。

表 106 : sp_iqprocparm の使用例

構文	出力
sp_iqprocparm	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プロシージャのパラメータを表示する。
sp_iqprocparm sp_test	プロシージャ sp_test に関する情報を表示する。
sp_iqprocparm non_existing_proc	プロシージャ non_existing_proc が存在しないため、ローは返されない。

構文	出力
sp_iqprocparm NULL, DBA	DBA が所有するすべてのプロシージャのパラメータを表示する。
sp_iqprocparm sp_test, DBA	DBA が所有するプロシージャ sp_test のパラメータを表示する。
sp_iqprocparm sp_iqtable	sp_iqtable はシステムプロシージャである。 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、ローは返されない (デフォルトでは、ユーザ定義プロシージャのみが返される)。
sp_iqprocparm sp_iqtable, dbo	プロシージャ sp_iqtable はユーザプロシージャではないため、ローは返されない (デフォルトでは、ユーザプロシージャのみが返される)。
sp_iqprocparm NULL, NULL, SYSTEM	(dbo または SYS が所有する) すべてのシステムプロシージャのパラメータを表示する。
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する。
sp_iqprocparm sp_iqtable, dbo, ALL	dbo が所有するシステムプロシージャ sp_iqtable のパラメータを表示する。

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って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** は指定の所有者が所有するプロシージャに関する情報を返すだけです。パラメータを指定しない場合、**sp_iqprocparm** はパラメータに関する情報をデータベース内のすべてのユーザ定義プロシージャに表示します。

sp_iqprocparm プロシージャは、次のカラムに情報を返します。

表 107 : 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	<p>パラメータは、次に示すタイプのいずれかに該当する。</p> <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る。 SQLSTATE エラー値。 SQLCODE エラー値。
parm_mode	<p>パラメータのモード。プロシージャに値を渡すパラメータ、値を返すパラメータ、この両方を行うパラメータ、またはいずれも行わないパラメータ。パラメータモードは、次のいずれかである。</p> <ul style="list-style-type: none"> in：パラメータはプロシージャに値を渡す。 out：パラメータは値を返す。 inout：パラメータは、値を渡すとともに値を返す。 NULL：パラメータは値を渡すことも返すこともしない。
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
h 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：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqkeys プロシージャ (553 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqrebuildindex プロシージャ

カラムインデックスを再構築します。

デフォルトの FP 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。**sp_iqrebuildindex** の動作は、FP_NBIT_IQ15_COMPATIBILITY の設定に関係なく同じです。

構文

```
sp_iqrebuildindex table_name, index_clause
```

注意： サードパーティの参考資料には、サポートされていない **sp_iqrebuildindex** 構文についても記述されています。index 句でテーブル名を指定するとエラーが発生することに注意してください。

```
sp_iqrebuildindex tbl, 'column tbl.c1'
```

パラメータ

- **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*]

index *index_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* を複数回指定した場合、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** の値を表す、負数以外の数です。**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。

MERGEALL と **RETIER** は、HG インデックス操作に固有のキーワードです。

```
sp_iqrebuildindex('table name', 'index index name [ MERGEALL | RETIER ]')
```

MERGEALL または **RETIER** が HG インデックスの操作から省略された場合、**sp_iqrebuildindex** は、カラムデータから HG インデックス全体をトランケートおよび再構築します。

MERGEALL は、多層構造の HG インデックスのすべての層をマージし、内容を適切な層に移動します。

```
sp_iqrebuildindex(' table name ', ' index index name MERGEALL ')
```

このマージにより、多層構造の HG インデックスでは、アクティブなサブインデックスが 1 つのみとなります。(**sp_iqindexmetadata** で示されているように) 削除されたレコードが多すぎる場合は、**MERGEALL** 操作を行うと、多層構造のインデックスのクエリアクセス時間が短縮されることがあります。**MERGEALL** がサポートされるのは、**index** 句が併用されていて、指定されたインデックスが HG インデックスである場合のみです。

RETIER は、HG インデックスに固有のキーワードであり、HG インデックスの形式を、多層構造でない HG から多層構造の HG、または多層構造の HG から多層構造でない HG に変更します。

```
sp_iqrebuildindex(' table name ', ' index index name RETIER ')
```

RETIER は、HG インデックスの形式を切り替えます。

- RETIER は、多層構造の HG インデックスを多層構造でない単一の HG インデックスに変換します。メタデータの多層構造化は無効になっていて、1つのサブインデックスのみが保持されます。
- RETIER は、多層構造でない HG インデックスを多層構造の HG インデックスに変換し、すべてのデータが格納された単一のサブインデックスを適切な階層にプッシュします。

MERGEALL と RETIER がサポートされるのは、`index` 句が併用されていて、指定されたインデックスが HG インデックスである場合のみです。

キーワード **column** および **index** 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。

注意： このプロシージャは **TEXT** インデックスをサポートしません。**TEXT** インデックスを再構築するには、インデックスを削除してから再作成する必要があります。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- INSERT ANY TABLE システム権限
- インデックスを再構築するテーブルに対する INSERT 権限
- テーブルを所有している

備考

カラム名を指定した場合、`sp_iqrebuildindex` はそのカラムのデフォルトの FP 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 SAP Sybase IQ によって割り当てられたデフォルトの FP インデックスの名前を指定すると、`sp_iqrebuildindex` はエラーを返します。

`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。

IQ UNIQUE の n 値が設定されているカラムでは、`sp_iqrebuildindex` によって再構築されるカラムが Flat FP か NBit かを決定します。IQ UNIQUE の n 値を 0 に設定すると、インデックスは Flat FP として再構築されます。0 より大きく、2,147,483,647 より小さい n 値を設定すると、インデックスは NBit として再構築されます。 n 値を設定しない NBit カラムは、NBit として再構築されます。

`sp_iqrebuildindex` は、カウントを指定していなくても、NBit カラムを NBit として再構築します。カウントを指定している場合、 n には、インデックスにすでに存在しているユニークな値の数より大きな値を設定する必要があります。

Flat FP インデックスの設定されたカラムを再構築し、そのカラムに IQ UNIQUE の n 値が含まれていない場合、`sp_iqrebuildindex` では、

FP_NBIT_AUTOSIZE_LIMIT オプションおよび FP_NBIT_LOOKUP_MB オプションで設定されている限度まで、インデックスが Flat FP として再構築されます。フラットカラムに *n* 値を指定すると、FP_NBIT_ENFORCE_LIMITS=ON になっていてカーディナリティがカウントを超過した場合にエラーがスローされます。

sp_iqrebuildindex のデフォルトのインタフェースでは、ユーザは、既存の FP インデックスから HG インデックス全体を再作成することができます。

sp_iqrebuildindex は、FP インデックスのカラム値をすべて再読み込みし、HG インデックスを作成します。ただし、この場合は、層のサイズや継続ロードのサイズなどに関するメタデータがすべて保持されます。

例

カラム *Surname* のデフォルトの FP インデックスを再構築します。

```
sp_iqrebuildindex 'empl1', 'column dept_id'
```

または

```
call sp_iqrebuildindex ('empl1', 'column dept_id')
```

カラム *c1* のフラットな FP インデックスを作成します。

```
CREATE TABLE mytable (c1 int IQ UNIQUE (0))
```

デフォルトの Flat FP インデックスを、重複する値を除いた推定カウントが 1024 の Nbit 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')
```

注意： 大きな HG インデックスに対して **sp_iqrebuildindex** が実行されると、パフォーマンスが一時的に低下することが考えられます。

参照：

- sp_iqindexfragmentation プロシージャ (514 ページ)
- sp_iqrowdensity プロシージャ (574 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqrebuildindexwide プロシージャ

255 バイトを超える、16.0 より前の FP インデックスを再構築します。

構文

```
sp_iqrebuildindexwide [ table_name ] [, table_owner] [, level]
```

パラメータ

- **table_name** – テーブルを指定します。このパラメータは必須ですが、空の文字列を指定できます。**table_name** に空の文字列を代入すると、コマンドで指定した **table owner** が所有する、データベース内のすべてのワイドカラムテーブルが再構築されます。**table_name** と **table owner** に空の文字列を代入すると、データベース内のすべてのワイドカラムテーブルが再構築されます。
- **table_owner** – テーブルの所有者。明示的な **table owner** 名はオプションです。デフォルトは空の文字列です。**table_name** に空の文字列を代入すると、コマンドで指定した **table owner** が所有する、データベース内のすべてのワイドカラムテーブルが再構築されます。明示的な **table_name** を使用し、**table_owner** として空の文字列を指定すると、すべてのユーザのテーブルが再構築されます。**table_owner** と **table owner** に空の文字列を代入すると、データベース内のすべてのワイドカラムテーブルが再構築されます。
- **level – sp_iqrebuildindexwide** がテーブルを再構築する方法を決定します。このパラメータは省略可能で、次の4つのオプションがあります。
 - レベル '1' を指定すると、指定したユーザについて、255 バイトを超える、16.0 より前のすべてのカラムが再構築されます。
 - レベル '2' を指定すると、VARCHAR または VARBINARY カラム、255 バイトを超える、16.0 より前のすべてのカラムに加えて、すべてのトークン化 FP (16.0 より前の 1/2/3 バイトの FP、射影可能な 1 バイトおよび 2 バイトの FP、16.0 NBit FP) が再構築されます。
 - レベル '3' を指定すると、すべての固定フラット FP と、255 バイトを超える、16.0 より前のすべてのカラムが再構築されます。
 - レベル '4' を指定すると、レベル 1、2、3 が適用され、255 バイトを超える 16.0 より前のすべてのカラム、すべてのトークン化 FP、すべての VARCHAR および VARBINARY カラム、すべての固定フラット FP が再構築されます。

level パラメータを省略すると、**sp_iqrebuildindexwide** はレベル '1' で実行されます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。:

- INSERT ANY TABLE システム権限
- インデックスを再構築するテーブルに対する INSERT 権限
- ユーザがテーブルを所有している

備考

SAP Sybase IQ 16.0 に移行されたデータベース内のすべての LONG VARCHAR カラムと LONG BINARY カラム、および 255 文字を超える CHAR、VARCHAR、BINARY、VARBINARY の各カラムに対してデータベースエンジンが読み込み／書き込みアクティビティを実行できるようにするには、これらのカラムを再構築する必要があります。

SAP Sybase IQ では、読み書きアクセスのために初めてテーブルを開いたときに、これらの型のカラムが暗黙的に再構築されます。**sp_iqrebuildindexwide** は、level パラメータで定義された状態にこれらのカラムを明示的に再構築します。

sp_iqrebuildindexwide は、実行結果を SAP Sybase IQ メッセージファイル (<dbname>.iqmsg) に書き込みます。

例

- **例 1** – vartab テーブルを DBA が所有しており、

```
select * from sp_iqrowdensity('table user1.vartab')
```

を実行すると、次の結果が返されるとします。

```
DBA.vartab  rid  Flat style FP
DBA.vartab  lvb1 Long varbinary FP
DBA.vartab  blob2 Long binary FP
DBA.vartab  blob1 Long binary FP
DBA.vartab  vc1  Flat style FP
DBA.vartab  lvc1 Long varchar FP
DBA.vartab  clob2 Long binary FP
DBA.vartab  clob1 Long binary FP
DBA.vartab  part Flat style FP
DBA.vartab  vb1  Flat style FP
```

table_name として vartab、**table_owner** として DBA を指定してレベル '1' で **sp_iqrebuildindexwide** を実行します。

```
call sp_iqrebuildindexwide('vartab', 'DBA', 1)
```

カラム clob1、colb2、lvc1、lvb1、blob1、blob2 が再構築され、次のメッセージが .iqmsg に書き込まれます。

```
Index Rebuild in progress for "DBA"."vartab" column "clob1" column
"clob2" column "lvc1" column "lvb1" column "blob1" column "blob2"
```

- **例 2** – vartab3 テーブルを user1 が所有しており、

```
select * from sp_iqrowdensity('table user1.vartab3')
```

を実行すると、次の結果が返されるとします。

```
user1.vartab3  rid  Flat style FP
user1.vartab3  part Flat style FP
user1.vartab3  vb1  Long varbinary FP
user1.vartab3  tk5  Two Byte FP
```

```

user1.vartab3 tk4 One Byte FP
user1.vartab3 tk3 Three Byte FP
user1.vartab3 vc1 Long varchar FP
user1.vartab3 tk2 Projectable Two Byte FP
user1.vartab3 b1 Flat style FP
user1.vartab3 c1 Long varchar FP
user1.vartab3 tk1 Projectable One Byte FP

```

table_name として vartab3、**table_owner** として user1 を指定してレベル '2' で **sp_iqrebuildindexwide** を実行します。

```
sp_iqrebuildindexwide('vartab3', 'user1', 2)
```

カラム vc1、vb1、c1、b1、tk1、tk2、tk3、tk4、tk5 が再構築され、次のメッセージが .iqmsg に書き込まれます。

```

Index Rebuild in progress for "user1"."vartab3" column "vc1"
column "vb1" column "c1" column "b1" column "tk1" column "tk2"
column "tk3" column "tk4" column "tk5"

```

- **例 3** – vartab3 テーブルを user1 が所有しており、

```
select * from sp_iqrowdensity('table user1.vartab3')
```

を実行すると、次の結果が返されるとします。

```

user1.vartab3 part Flat style FP
user1.vartab3 c1 2 Bit FP
user1.vartab3 tk4 2 Bit FP
user1.vartab3 tk5 2 Bit FP
user1.vartab3 vb1 2 Bit FP
user1.vartab3 tk3 2 Bit FP
user1.vartab3 tk2 2 Bit FP
user1.vartab3 tk1 2 Bit FP
user1.vartab3 b1 2 Bit FP
user1.vartab3 rid Flat style FP
user1.vartab3 vc1 2 Bit FP

```

table_name として vartab3、**table_owner** として user1 を指定してレベル '3' で **sp_iqrebuildindexwide** を実行します。

```
call sp_iqrebuildindexwide('vartab3', 'user1', 3)
```

カラム rid と part が再構築され、次のメッセージが .iqmsg に書き込まれます。

```

Index Rebuild in progress for "user1"."vartab3" column "rid"
column "part"

```

再構築されたテーブルに対して

```
select * from sp_iqrowdensity('table user1.vartab3')
```

を実行すると、次の結果が返されます。

```

user1.vartab3 rid 2 Bit FP
user1.vartab3 vc1 2 Bit FP
user1.vartab3 vb1 2 Bit FP
user1.vartab3 b1 2 Bit FP
user1.vartab3 tk1 2 Bit FP

```

```

user1.vartab3 tk4 2 Bit FP
user1.vartab3 tk5 2 Bit FP
user1.vartab3 tk3 2 Bit FP
user1.vartab3 c1 2 Bit FP
user1.vartab3 part 2 Bit FP
user1.vartab3 tk2 2 Bit FP

```

参照：

- `sp_iqindexrebuildwidedata` プロシージャ (521 ページ)
- `sp_iqrebuildindex` プロシージャ (561 ページ)

sp_iqrename プロシージャ

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)、ストアードプロシージャ、ファンクションの名前を変更します。

構文

```
sp_iqrename object-name, new-name [, object-type ]
```

パラメータ

- **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` を呼び出しているユーザがオブジェクトの所有者と見なされます。`sp_iqrename` を呼び出しているユーザがオブジェクトの名前を変更するために必要な権限を持つ場合にのみ、オブジェクトの名前が正しく変更されます。

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せん。

- **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。

有効値:

- **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
- **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
- **object-type not specified** – 名前を変更するオブジェクトはテーブル

警告！ `sp_iqrename` によって名前を変更されてい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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。以下のように、具体的なオプションタイプの名前を変更するには、さまざまな権限が必要です。

- **ALTER ANY OBJECT システム権限** – いずれのオブジェクトでも、名前を変更することができます。
- **ALTER ANY TABLE システム権限** – いずれのテーブル、カラム、制約でも、名前を変更することができます。
- **ALTER ANY INDEX システム権限** – いずれのインデックスでも、名前を変更することができますが、テーブルまたはカラムの名前を変更することはできません。
- **テーブルに対する REFERENCES 権限** – そのテーブルのインデックスのみ、名前を変更することができます。
- **そのテーブルに対する ALTER 権限** – そのテーブル、カラム、および制約のみ、名前を変更することができます。
- **テーブルを所有している** – 自分が所有しているいずれのオブジェクトでも、名前を変更することができます。

このプロシージャを実行するには、オブジェクトに対する排他的なアクセス権限も必要です。

備考

`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、`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 メッセージ "機能はサポートされていません。" が返されません。

ALTER TABLE 文と **ALTER INDEX** 文の **RENAME** 句を使用して名前を変更することもできます。

例

ユーザ shweta が所有するテーブル 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 上のプライマリーキー制約 prim_id の名前を prim_isbn に変更します。

```
sp_iqrename books.prim_id, prim_isbn, constraint
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iq_reset_identity プロシージャ

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 table_name, table_owner, value
```

パラメータ

table_name、*table owner*、*value* を指定してください。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- ALTER ANY TABLE システム権限
- ALTER ANY OBJECT システム権限
- そのテーブルに対する ALTER 権限
- テーブルを所有している

備考

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。

参照：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iqtable プロシージャ (614 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iq_reset_identity プロシージャ例

sp_iq_reset_identity を使用する場合は、この例を参照してください。

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。

```
CREATE TABLE mytable(c1 INT identity)
call sp_iq_reset_identity('mytable', 'dba', 50)
```

sp_iqrestoreaction プロシージャ

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クションを示します。

構文

```
sp_iqrestoreaction 'timestamp'
```

パラメータ

- **timestamp** – 対象の過去の日付を指定します。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

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` によって、次の結果が返されます。

表 108 : `sp_iqrestoreaction` のカラム

カラム名	説明
<code>sequence_number</code>	処理を行う順序
<code>backup_id</code>	バックアップトランザクションの識別子
<code>backup_archive_list</code>	バックアップのアーカイブファイルのリスト
<code>backup_time</code>	バックアップを行った時間
<code>virtual_type</code>	仮想バックアップの種類: "Non-virtual"、"Decoupled"、または "Encapsulated"。
<code>restore_dbSPACE</code>	空にできる。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す。
<code>restore_dbfile</code>	空にできる。指定の DB 領域のすべての <code>dbfile</code> 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す。
<code>backup_comment</code>	ユーザコメント

例

`sp_iqrestoreaction` の出力例を以下に示します。

```

sequence_number  backup_id  backup_archive_list  backup_time
1                1192      c:YYYYtempYYYYb1   2008-09-23
14:47:40.0
2                1201      c:YYYYtempYYYYb2.inc 2008-09-23
14:48:05.01
3                1208      c:YYYYtempYYYYb3.inc 2008-09-23
14:48:13.0

virtual_type     restore_dbpace  restore_dbfile  backup_comment
Nonvirtual
Nonvirtual
Nonvirtual

```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqrlvmemory プロシージャ

テーブルごとに RLV ストアのメモリ使用量をモニタリングします。

構文

```
sp_iqrlvmemory ( [table_name [,table_owner ] ])
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR システム権限。

備考

バージョンのビットマップ、要求に応じたインデックスなど、バージョン固有のデータは RLV のメモリ計算に含まれません。これらは RLV メモリ制限にカウントされず、sp_iqrlvmemory でレポートされません。

パラメータが指定されていない場合は、すべての RLV テーブルのメモリ消費に関する情報が返されます。出力を1つのテーブルに限定するには、table_name と追加の table_owner オプションを指定できます。table_owner が指定されていない場合は、デフォルトは現在のユーザになります。

sp_iqrlvmemory は RLV ストアメモリを消費するテーブルごとに1つのローを、次の出力カラムとともに出力します。

カラム名	説明
table_id	このローが表すテーブルの ID。
fragments	このテーブルのストアフラグメントの数。
total	このテーブルが使用する RLV ストアメモリ合計 (MB)。

カラム名	説明
data	このテーブルのカラムフラグメントに使用される RLV ストアメモリ (MB)。
dictionary	このテーブルのディクショナリに使用される RLV ストアメモリ (MB)。
bitmap	テーブルレベルのビットマップの格納に使用される RLV ストアメモリ (MB)。

例

この例では、ユーザ DBA が所有するテーブル `rlv_table1` の現在の RLV メモリ使用量が返されます。

```
sp_iqrlvmemory 'rlv_table1', 'DBA'
```

出力：

	table_id	fragments	total	data	dictionary	bitmap
1	778	1	1	0	1	1
2	779	1	48	48	0	1
3	785	2	1596	1584	0	12

sp_iqrowdensity プロシージャ

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 (...))
```

パラメータ

- **table-name** – 指定し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** – ターゲットテーブル内の指定し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。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- MONITOR
- MANAGE ANY DBSPACE
- CREATE ANY INDEX
- ALTER ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- テーブルを所有している

備考

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 $0 < density < 1$ 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 .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。

sp_iqrowdensity の IndexType カラムでは常に、このカラムのエンコードに必要なビットの最大数が返されます。

以前のリリースにあった FP(1)、FP(2)、FP(3) の各ディクショナリ圧縮では、各ページで同じビット数を使用していましたが、それとは違って、NBit では各ページを動的にエンコードします。sp_iqrowdensity では常に、ページのすべてで使用されるビットの最大数が返されます。

例

次のプロシージャは、SalesOrders テーブル内のカラム ID のロー密度をレポートします。

```
sp_iqrowdensity('column group0.SalesOrders.ID')
```

Tablename	ColumnName	IndexType	Density
GROUP0.SalesOrders	ID	NBit FP	1.0

参照：

- sp_iqindexfragmentation プロシージャ (514 ページ)
- sp_iqrebuildindex プロシージャ (561 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqsetcompression プロシージャ

LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムのデータの圧縮を設定します。

構文

```
sp_iqsetcompression ( owner, table, column, on_off_flag )
```

パーミッション

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- ALTER ANY TABLE
- ALTER ANY OBJECT

備考

sp_iqsetcompression は、LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムの圧縮を制御します。圧縮設定は、ベーステーブルにのみ適用されます。

sp_iqsetcompression には、圧縮設定を変更した後に **COMMIT** が実行されるという二次的な影響があります。

表 109 : sp_iqsetcompression パラメータ

名前	説明
<i>owner</i>	圧縮を設定するテーブルの所有者
<i>table</i>	圧縮を設定するテーブル
<i>column</i>	圧縮を設定するカラム
<i>on_off_flag</i>	圧縮設定：圧縮を有効にする場合は ON、無効にする場合は OFF

例

次のテーブル定義を想定します。

```
CREATE TABLE USR.pixTable (picID INT NOT NULL,  
picJPG LONG BINARY NOT NULL);
```

LOB カラム picJPG で圧縮を無効にするには、**sp_iqsetcompression** を呼び出します。

```
CALL sp_iqsetcompression('USR', 'pixTable', 'picJPG',  
'OFF');
```

このコマンドはローを返しません。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqsharedtempdistrib プロシージャ

共有テンポラリ領域の使用分散について、現在の状況を示します。

sp_iqsharedtempdistrib をコーディネータで実行すると、すべてのノードの共有テンポラリ領域の分散状況が表示されます。セカンダリノードで実行すると、そのノードの共有テンポラリ領域の使用状況が表示されます。

共有テンポラリ領域は、マルチプレックスを構成する各ノードがオンデマンドで使用できるように予約されています。領域はアロケーションユニットを単位として各ノードに予約されます。ノードは、動的な領域需要に基づいて、複数のアロケーションユニットを予約できます。アロケーションユニットは、必要に応じてより多くの領域を使用できるようにノードに貸し出され、不要になるとグローバルプールに戻されます。領域の使用状況が減少し、かつ貸し出し時間が終了するか、またはサーバが停止すると、アロケーションユニットの有効期間は終了します。

構文

```
sp_iqsharedtempdistrib
```

適用対象

マルチプレックスのみ。

権限

MANAGE ANY DBSPACE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

カラム	データ型	説明
Server_id	unsigned bigint	SYSIQMPXINFO からの、マルチプレックスサーバのサーバ ID。
DBSpace_name	char(128)	領域が予約される DB 領域の名前。

カラム	データ型	説明
Unit_type	char(10)	アロケーションユニットのタイプ。有効な値： <ul style="list-style-type: none"> アクティブ – 現在予約されており、ノードによって使用されている。 有効期間切れ – ノード用に予約されているが、グローバル領域プールへの返還処理中。 隔離中 – ノード用に予約されているが、ノード障害により隔離されている。
VersionID	unsigned bigint	ユニットのバージョン ID。アクティブなユニットでは、ユニットがノード用に予約された時点のバージョン。有効期間切れのユニットでは、ユニットが有効期間切れとなった時点のバージョン。隔離中のユニットでは、ユニットが隔離された時点のバージョン。
NBlocks	unsigned bigint	ユニット内の未使用ブロックの数。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqshowcompression プロシージャ

LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムの圧縮設定を表示します。

構文

```
sp_iqshowcompression ( owner, table, column )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- ALTER ANY TABLE
- ALTER ANY OBJECT

備考

カラム名と圧縮設定を返します。圧縮設定値は 'ON' (圧縮は有効) および 'OFF' (圧縮は無効) です。

表 110 : `sp_iqshowcompression` パラメータ

名前	説明
<i>owner</i>	圧縮を設定するテーブルの所有者
<i>table</i>	圧縮を設定するテーブル
<i>column</i>	圧縮を設定するカラム

例

次のテーブル定義を想定します。

```
CREATE TABLE USR.pixTable (picID INT NOT NULL,
picJPG LONG BINARY NOT NULL);
```

`pixTable` テーブルのカラムの圧縮ステータスをチェックするには、**`sp_iqshowcompression`** を呼び出します。

```
CALL sp_iqshowcompression('USR', 'pixTable',
'picJPG');
```

このコマンドは 1 つのローを返します。

```
'picJPG', 'ON'
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqshowpsexex プロシージャ

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_iqshowpsexex [ connection-id ]
```

パラメータ

- connection-id** – 接続 ID を表す整数。

`connection-id` を指定した場合、**`sp_iqshowpsexex`** は指定した接続に関する情報のみを返します。`connection-id` を指定しない場合、**`sp_iqshowpsexex`** 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_iqshowpsexex`** はローを返しません。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

備考

sp_iqshowpsex 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 111 : sp_iqshowpsex 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む。HOST：クライアントマシン EXE のホスト名クライアントの実行可能な APPINFO の名前 (Windows のみ)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、このオプションは値 2 (MEDIUM) を持つ。値 1、2、および 3 はそれぞれ、HIGH、MEDIUM、および LOW として表示される。
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 0 を持つ。
query_row_limit	結果セットの予測サイズにもとづいてクエリを拒否するためのローレッシュヨルドを設定する、データベースオプション QUERY_ROWS_RETURNED_LIMIT の値。デフォルトは 0 で、制限がないことを示す。
query_temp_space_limit	ユーザクエリによってテンポラリ 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 2,000 MB。
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。値 0 は、制限がないことを示す。

カラム名	説明
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。値 0 は、制限がないことを示す。

注意： **AppInfo** プロパティは、Interactive SQL 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。**AppInfo** 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-16_0\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：

- CONNECTION_PROPERTY 関数 [システム] (177 ページ)
- sp_iqcontext プロシージャ (466 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqspaceinfo プロシージャ

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] [...] \]
```

適用対象

シンプレックスとマルチプレックス。

権限

MANAGE ANY DBSPACE システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

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。

マルチプレックスデータベース上で実行する場合のデフォルトのパラメータは main で、共有 IQ ストアのサイズが返されます。

パラメータを指定しない場合、結果を受け取るには、テーブルなどのユーザ作成オブジェクトが 1 つ以上必要です。

例

この出力は、iqdemo データベースで実行される **データベース上で実行される sp_iqspaceinfo** ストアドプロシージャから返されたものです。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 (516 ページ)
- sp_iqdbspace プロシージャ (477 ページ)
- sp_iqdbspaceinfo プロシージャ (481 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqspaceused プロシージャ

空き領域と IQ ストア、IQ テンポラリストア、RLV ストア、および IQ グローバルと IQ ローカルの共有テンポラリストアの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,
out shTempTotalKB      unsigned bigint,
out shTempTotalKBUsed  unsigned bigint,
out shTempLocalKB      unsigned bigint,
out shTempLocalKBUsed  unsigned bigint,
out rlvLogKB           unsigned bigint,
out rlvLogKBUsed       unsigned bigint)

```

適用対象

シンプレックスとマルチプレックス。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR

備考

sp_iqspaceused は、unsigned bigint の out パラメータとしていくつかの値を返します。このシステムストアプロシージャをユーザ定義のストアプロシージャから呼び出すと、メイン、テンポラリ、および RLV ストアの領域の使用量を確認できます。

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。

マルチプレックスデータベースで実行すると、このプロシージャは、プロシージャを実行しているサーバに適用されます。また、IQ_SHARED_TEMP で使用される領域も返します。

カラム名	説明
mainKB	IQ メインストアの領域の合計 (KB 単位)。
mainKBUsed	データベースが使用している IQ メインストアの領域 (KB 単位)。セカンダリマルチプレックスノードは 'Null' を返す。
tempKB	IQ テンポラリストアの領域の合計 (KB 単位)。
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の合計 (KB 単位)。
shTempTotalKB	IQ グローバル共有テンポラリストアの領域の合計 (KB 単位)。

カラム名	説明
shTempLocalKB	IQ ローカル共有テンポラリストアの領域の合計 (KB 単位)。
shTempLocalKBUsed	データベースが使用している IQ ローカル共有テンポラリストアの領域 (KB 単位)。
rlvLogKB	RLV ストアの領域の合計 (KB 単位)。
rlvLogKBUsed	データベースが使用している RLV ストアの領域 (KB 単位)。

例

sp_iqspaceused には 7 つの出力パラメータが必要です。7 つの出力パラメータを宣言してから **sp_iqspaceused** を呼び出す、ユーザ定義ストアプロシージャ **myspace** を作成します。

```

create or replace procedure dbo.myspace ()
begin
    declare mt unsigned bigint;
    declare mu unsigned bigint;
    declare tt unsigned bigint;
    declare tu unsigned bigint;
    declare gt unsigned bigint;
    declare gu unsigned bigint;
    declare lt unsigned bigint;
    declare lu unsigned bigint;
    declare tt_t unsigned bigint;
    declare mt_t unsigned bigint;
    declare gt_t unsigned bigint;
    declare lt_t unsigned bigint;
    call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu);
    if (tt = 0) then
        set tt_t = 0;
    else
        set tt_t = tu*100/tt;
    end if;
    if (mt = 0) then
        set mt_t = 0;
    else
        set mt_t = mu*100/mt;
    end if;
    if (gt = 0) then
        set gt_t = 0;
    else
        set gt_t = gu*100/gt;
    end if;
    if (lt = 0) then
        set lt_t = 0;
    else
        set lt_t = lu*100/lt;
    end if;
    select cast(mt/1024 as unsigned bigint) as mainMB,

```

```

        cast(mu/1024 as unsigned bigint) as mainusedMB, mt_t as
mainPerCent,
        cast(tt/1024 as unsigned bigint) as tempMB,
        cast(tu/1024 as unsigned bigint) as tempusedMB, tt_t as
tempPerCent,
        cast(gt/1024 as unsigned bigint) as shTempTotalKB,
        cast(gu/1024 as unsigned bigint) as shTempTotalKBUsed, gt_t
as globalshTempPerCent,
        cast(lt/1024 as unsigned bigint) as shTempLocalMB,
        cast(lu/1024 as unsigned bigint) as shTempLocalKBUsed, lt_t
as localshTempPerCent;
end

```

sp_iqspaceused の出力を表示するには、**myspace** を実行します。

```
myspace
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqstatistics プロシージャ

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。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが必要です。MANAGE ANY STATISTICS システム権限。

備考

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

カラム名	データ型	説明
stat_desc	VARCHAR(255)	統計の説明
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 SAP Sybase IQ サーバによって消費された合計 CPU 時間 (秒)	秒
1	CpuUserTime	最後にサーバを起動してから SAP Sybase IQ サーバによって消費された CPU ユーザ時間 (秒)	秒
2	CpuSystemTime	最後にサーバを起動してから SAP Sybase IQ サーバによって消費された CPU システム時間 (秒)	秒
3	ThreadsFree	未使用の SAP Sybase IQ スレッド数	なし
4	ThreadsInUse	使用中の SAP Sybase IQ スレッド数	なし
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	メインバッファキャッシュの現在のサイズ (メガバイト)	MB
8	MainCacheFinds	メインバッファキャッシュの合計ルックアップ要求数	なし
9	MainCacheHits	メインバッファキャッシュの合計ヒット数	なし
10	MainCachePagesPinned	メインバッファキャッシュの固定されたページ数	ページ
11	MainCachePagesPinned-Percentage	メインバッファキャッシュの固定されたページの割合	%

stat_num	stat_name	stat_desc	stat_unit
12	MainCachePagesDirtyPercentage	メインバッファキャッシュのダーティページの割合	%
13	MainCachePagesInUsePercentage	メインバッファキャッシュの使用ページ割合	%
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合計ルックアップ要求数	なし
16	TempCacheHits	テンポラリキャッシュの合計ヒット数	なし
17	TempCachePagesPinned	テンポラリキャッシュの固定されたページ数	ページ
18	TempCachePagesPinnedPercentage	テンポラリキャッシュの固定されたページの割合	%
19	TempCachePagesDirtyPercentage	テンポラリキャッシュのダーティページの割合	%
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	メインストアへの書き込み量 (KB)	KB
23	TempStoreDiskReads	メインストアからの読み取り量 (KB)	KB
24	TempStoreDiskWrites	メインストアへの書き込み量 (KB)	KB
25	ConnectionsTotalConnections	サーバが起動してからの合計接続数	なし
26	ConnectionsTotalDisconnections	サーバが起動してからの合計切断数	なし
27	ConnectionsActive	アクティブな接続数	なし
28	OperationsWaiting	SAP Sybase IQ リソースガバナを待機している処理の数	なし

stat_num	stat_name	stat_desc	stat_unit
29	OperationsActive	SAP Sybase IQ リソースガバナによって認識されたアクティブな同時操作の数	なし
30	OperationsActiveLoadTableStatements	アクティブな LOAD TABLE 文の数	なし

例

1つの統計(総 CPU 時間)を表示します。

```
sp_iqstatistics 'CpuTotalTime'
```

MainCache% のすべての統計を表示します。

```
SELECT * from sp_iqstatistics() WHERE stat_name LIKE 'MainCache%'
```

stat_num	stat_name	stat_desc	stat_value	stat_unit
7	MainCacheCurrentSize	キャッシュ DB 領域の現在のサイズ(メガバイト)	64	mb
8	MainCacheFinds	キャッシュ DB 領域の合計ルックアップ要求数	95303	
9	MainCacheHits	キャッシュ DB 領域の合計ヒット数	95283	
10	MainCachePagesPinned	キャッシュ DB 領域の固定されたページ数	0	ページ
11	MainCachePagesPinnedPercentage	キャッシュ DB 領域の固定されたページの割合	0	%
12	MainCachePagesDirtyPercentage	キャッシュ DB 領域のダーティページの割合	0.39	%
13	MainCachePagesInUsePercentage	使用中のキャッシュ DB 領域ページの割合	4.44	%

参照:

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqstatus プロシージャ

現在のデータベースについて、さまざまな SAP Sybase IQ ステータス情報を表示します。

構文

```
sp_iqstatus
```

適用対象

シンプレックスとマルチプレックス。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限:

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR
- SERVER OPERATOR

備考

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。いずれか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値に Y を表示します。

ローレベルのバージョン管理 (RLV) ストアによって使用されるメモリは、**sp_iqstatus** でモニタできます。**RLV memory limit** ローには、`-iqrlvmem` サーバオプションまたは `sa_server_option rlv_memory_mb` で指定されるメモリ制限が表示されます。RLV memory used ローには、RLV ストアによって使用されるメモリ量が表示されます。

直接接続されているストレージデバイスによって使用される、キャッシュ DB 領域内のメモリは、**sp_iqstatus** でモニタできます。

測定	説明
キャッシュ DB 領域ファイル数	データベース内のキャッシュ DB 領域の DB ファイルの数。

測定	説明
キャッシュ DB 領域ブロック	キャッシュ DB 領域ブロックと対応するストレージデバイスの DB ファイル名。
使用中のキャッシュ DB 領域 IQ ブロック	IQ ブロックの合計数に対する使用中の IQ ブロックの数。使用率はパーセントでも表示。パーセントが高い場合は、ストレージの追加を検討する。

sp_iqspaceused は、**sp_iqstatus** によって提供された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。

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加算します。

```
(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を省略します。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。

マルチプレックスの場合、このプロシージャは共有 IQ ストアと IQ テンポラリストアに関する情報もリストします。**sp_iqstatus** でマルチプレックスサーバのメインブロック使用率が高いことが示された場合は、**sp_iqversionuse** を実行して、使用されているバージョンと、バージョンをリリースすることでリカバリできる容量を確認します。

例

注意：この例には、iq_main というサンプルのユーザ DB 領域が含まれていますが、この領域はユーザ独自のデータベースには存在しない場合があります。

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) 1992-2013 by SAP AG or an SAP affiliate
company. All rights reserved.
Version: 16.0.0.562/130821/P/Mainline/Sun_x64/OS 5.10/64bit/
```

```
2013-08-21 06:15:41
Time Now: 2013-08-21 06:27:14.150
Build Time: 2013-08-21 06:15:41
File Format: 23 on 03/18/1999
Server mode: IQ Server
Catalog Format: 2
Stored Procedure Revision: 1
Page Size: 65536/4096blks/16bpp
Number of Main DB Files: 2
Main Store Out Of Space: N
Number of Cache Dbspace Files: 5
Number of Shared Temp DB Files: 0
Shared Temp Store Out Of Space: N
Number of Local Temp DB Files: 1
Local Temp Store Out Of Space: N
DB Blocks: 1-25600 IQ_SYSTEM_MAIN
DB Blocks: 522208-547807 MainUser
Cache Dbspace Blocks: 1-5120 ssd_dev_1
Cache Dbspace Blocks: 522208-527327 ssd_dev_2
Cache Dbspace Blocks: 1044416-1049535 ssd_dev_3
Cache Dbspace Blocks: 1566624-1571743 ssd_dev_4
Cache Dbspace Blocks: 2088832-2093951 ssd_dev_5
Local Temp Blocks: 1-25600 IQ_SYSTEM_TEMP
Create Time: 2013-08-21 06:27:05.444
Update Time: 2013-08-21 06:27:14.035
Main IQ Buffers: 1588, 100Mb
Temporary IQ Buffers: 1588, 100Mb
Main IQ Blocks Used: 5250 of 38400, 13%=20Mb, Max Block#: 5313
Cache Dbspace IQ Blocks Used: 197 of 25600, 0%=0Mb, Max Block#: 0
Shared Temporary IQ Blocks Used: 0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used: 65 of 12800, 0%=0Mb, Max Block#: 0
Main Reserved Blocks Available: 12800 of 12800, 100%=50Mb
Shared Temporary Reserved Blocks Available: 0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available: 12800 of 12800, 100%=50Mb
IQ Dynamic Memory: Current: 292mb, Max: 308mb
Main IQ Buffers: Used: 18, Locked: 0
Temporary IQ Buffers: Used: 4, Locked: 0
Main IQ I/O: I: L459/P9 O: C21/D33/P22 D:1 C:100.0
Temporary IQ I/O: I: L320/P0 O: C54/D59/P8 D:50 C:100.0
Other Versions: 0 = 0Mb
Active Txn Versions: 0 = C:0Mb/D:0Mb
Last Full Backup ID: 0
Last Full Backup Time:
Last Backup ID: 0
Last Backup Type: None
Last Backup Time:
DB Updated: 0
Blocks in next ISF Backup: 0 Blocks: =0Mb
Blocks in next ISI Backup: 0 Blocks: =0Mb
IQ large memory space: 2048Mb
IQ large memory flexible percentage: 50
IQ large memory flexible used: 0Mb
IQ large memory inflexible percentage: 90
IQ large memory inflexible used: 0Mb
IQ large memory anti-starvation percentage: 50
DB File Encryption Status: OFF
```

```
RLV memory limit: 2048Mb  
RLV memory used: 0Mb
```

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: 入力項目
- 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 (621 ページ)
- `sp_iqversionuse` プロシージャ (629 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqsysmon プロシージャ

SAP 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、バッファキャッシュ、メモリ、スレッド、ロック、入出力機能、および CPU 使用率の管理などが含まれます。

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
```

バッチモードでのパラメータ

- **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 形式で指定)。
- **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。

省略形の完全なリストについては、「備考(0 ページ)」の項を参照してください。

複数のセクションを指定する場合は、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みます。デフォルトでは、すべてのセクションを表示します。

IQ メイン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。プレフィ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る。'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る。

注意： SAP Sybase IQ コンポーネントのディスク 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パラメータ

- **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** – ファイルモードで **sp_iqsysmon** を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。出力ファイルのサフィックスを変更するには、**file_suffix** オプションを使用します。**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 1 つ以上のオプションを含めることができます。
 - **-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の間隔は 60 秒です。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。

最初の表示では、サーバの起動からのカウンタが示されます。それ以降の表示では、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、有意な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する必要があります。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で、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への追加、後者は既存の出力ファイルのトランケートを **sp_iqsysmon** に指示します。デフォルトは、トランケートです。両方のオプションを指定した場合、文字列内の後ろのほうで指定されたオプションが優先されます。
- **-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。

省略形の完全なリストについては、「備考 (0 ページ)」の項を参照してください。

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。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に警告が書き込まれます。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが必要です。MONITOR システム権限。

備考

レポート対象のレポートセクションまたは IQ コンポーネント	入力する省略形
バッファ割り付け	(メイン) – mbufalloc (テンポラリ) – tbufalloc
バッファマネージャ	(メイン) – mbufman (テンポラリ) – tbufman
バッファプール	(メイン) – mbufpool (テンポラリ) – tbufpool
カタログの統計	catalog

レポート対象のレポートセクションまたは IQ コンポーネント	入力する省略形
CPU の使用率	cpu
フリーリスト管理	(メイン) – mfreelist (テンポラリ) – tfreelist
メモリ管理	memory
プリフェッチ管理	(メイン) – mprefetch (テンポラリ) – tprefetch
IQ RLV インメモリストアの統計	rlv
LMA (Large Memory Allocator) の統計	lma
サーバコンテキスト統計	server
スレッド管理	threads
トランザクション管理	txn

sp_iqsysmon ストアドプロシージャは、SAP 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、バッファキャッシュ、メモリ、スレッド、ロック、入出力機能、および CPU 使用率の管理などが含まれます。

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 – 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に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、SAP 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 – 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。

バッチモードでの構文例

例 1 :

システムプロシージャ

バッチモードでモニタを開始し、メインストアおよびテナポラリストアのすべてのセクションを表示します。

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor
```

例 2：

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。

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor 'mbufman mbufpool'
```

例 3：

10 分後にモニタ情報を出力します。

```
sp_iqsysmon '00:10:00'
```

例 4：

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
```

例 5：

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 proc1
go
execute proc2
go
select sum(total_sales) from titles
go
sp_iqsysmon stop_monitor, bufman
go
```

例 6：

2 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'
```

例 7：

1 分後に、レポートの RLV セクションのみを出力します。

```
sp_iqsysmon '01:00:00', 'rlv'
```

例 8:

5 分後に、レポートの LMA セクションのみを出力します。

```
sp_iqsysmon '00:00:05', 'lma'
```

例 9:

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'
```

ファイルモードでの構文例

例 1:

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
```

例 2:

dbname.connid-testmon という名前の ASCII ファイルに、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出力のみを追加します。データベース iqdemo の場合は、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 3:

レポートの RLV セクションおよび LMA セクションのみを出力します。

```
sp_iqsysmon start_monitor, 'filemode', '-section rlv lma'
sp_iqsysmon stop_monitor
```

例 4:

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。

```
sp_iqsysmon start_monitor, 'filemode', '-interval 5 -section
mbufpool memory'
sp_iqsysmon stop_monitor
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqsysmon プロシージャ例

sp_iqsysmon 出力例です。

例 1：

20 分後にバッファ割り付け (メインおよびテンポラリ) の出力を表示します。

```
sp_iqsysmon '00:20:00', 'mbufalloc tbufalloc'

=====
Buffer Allocator (Main) "
=====

STATS-NAME                VALUE
NActiveCommands           2
BufAllocMaxBufs           2275 ( 81.6% )
BufAllocAvailBufs         2115 ( 93.0% )
BufAllocReserved          160 ( 7.0% )
BufAllocAvailPF           750 ( 33.0% )
BufAllocSlots             100
BufAllocNPinUsers         0
BufAllocNPFUsers          2
BufAllocNPostedUsrs      0
BufAllocNUnpostUsrs      0
BufAllocPinQuota          0
BufAllocNPostEst          0
BufAllocNUnPostEst        0
BufAllocMutexLocks        0
BufAllocMutexWaits        0 ( 0.0% )

STATS-NAME                VALUE
NActiveCommands           2
BufAllocMaxBufs           2275 ( 81.6% )
BufAllocAvailBufs         2115 ( 93.0% )
BufAllocReserved          160 ( 7.0% )
BufAllocAvailPF           750 ( 33.0% )
BufAllocSlots             100
BufAllocNPinUsers         0
BufAllocNPFUsers          2
BufAllocNPostedUsrs      0
BufAllocNUnpostUsrs      0
BufAllocPinQuota          0
BufAllocNPostEst          0
BufAllocNUnPostEst        0
BufAllocMutexLocks        0
BufAllocMutexWaits        0 ( 0.0% )

STATS-NAME                TOTAL  UNKNWN  HASH      CSORT      ROW
ROWCOL      FP  GARRAY  LOB  BTREE    BM      BV   STORE  TEST
NumClients      2    0    0    0    0    0    2
```

0	0	0	0	0	0	0	0	0	0	0
PinUserQuota				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PrefetchUserQuota				160	0	0	0	0	160	
0	0	0	0	0	0	0	0	0	0	0
PinUserRegisters				2	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PfUserRegisters				4697	0	0	0	0	382	
2621	377	182		0	2	0	0	0	0	0
ClientCountOfPinners				0	1	3		6	10	
33	66	100	333	666	1000	3333	6666	10000		
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Hash				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Sort				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Row				2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
RowColumn				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
FP				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Garray				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
LOB				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
BTree				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
BM				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
BV				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Store				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Test				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
DBCC				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Run				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
QCPRun				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
TextDoc				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
VDO				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

システムプロシージャ

Load	Pass	2	0	0	0	0	0
0	0	0	0	0	0	0	0
STATS-NAME (cont'd)		DBCC	BLKMAP	IQUTIL			
NumClients	0	0	0	0	0	0	0
PinUserQuota	0	0	0	0	0	0	0
PrefetchUserQuota	0	0	0	0	0	0	0
PinUserRegisters	0	0	0	0	0	0	0
PfUserRegisters	0	0	0	0	0	0	0
	0	0	1133	0			
ClientCountOfPinner	33333	66666	100000	4294967295			
Unknown	0	0	0	0			
Hash	0	0	0	0			
Sort	0	0	0	0			
Row	0	0	0	0			
RowColumn	0	0	0	0			
FP	0	0	0	0			
Garray	0	0	0	0			
LOB	0	0	0	0			
BTree	0	0	0	0			
BM	0	0	0	0			
BV	0	0	0	0			
Store	0	0	0	0			
Test	0	0	0	0			
DBCC	0	0	0	0			
Unknown	0	0	0	0			
Unknown	0	0	0	0			
Run	0	0	0	0			
QCPRun	0	0	0	0			
TextDoc	0	0	0	0			
Unknown	0	0	0	0			
Unknown	0	0	0	0			
VDO	0	0	0	0			
Load	0	0	0	0	0	0	0

=====
 Buffer Allocator (Temporary)
 =====

STATS-NAME	VALUE
NActiveCommands	2
BufAllocMaxBufs	2275 (81.6%)
BufAllocAvailBufs	2263 (99.5%)
BufAllocReserved	12 (0.5%)
BufAllocAvailPF	908 (39.9%)
BufAllocSlots	100
BufAllocNPinUsers	2
BufAllocNPFUsers	2
BufAllocNPostedUsrs	0
BufAllocNUnpostUsrs	0

BufAllocPinQuota												175
BufAllocNPostEst												2
BufAllocNUnPostEst												2
BufAllocMutexLocks												0
BufAllocMutexWaits												0(0.0%)
STATS-NAME				TOTAL	UNKNWN		HASH		CSORT		ROW	
ROWCOL	FP	GARRAY		LOB	BTREE		BM	BV	STORE		TEST	
NumClients				4	0		0		4		0	
0	0	0	0	0	0		0	0	0		0	
PinUserQuota				10	0		0		10		0	
0	0	0	0	0	0		0	0	0		0	
PrefetchUserQuota				2	0		0		2		0	
0	0	0	0	0	0		0	0	0		0	
PinUserRegisters				668	0		300		247		0	
0	0	0	0	0	0		0	0	0		0	0
PfUserRegisters				675	0		0		295		0	
0	0	0	0	0	0		0	0	1		0	
ClientCountOfPinner				0	1		3		6		10	
33	66	100	333	666	1000		3333		6666		10000	
Unknown				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Hash				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Sort				2	0		1		0		1	
0	0	0	0	0	0		0	0	0		0	
Row				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
RowColumn				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
FP				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Garray				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
LOB				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
BTree				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
BM				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
BV				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Store				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Test				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
DBCC				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Unknown				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Unknown				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
Run				0	0		0		0		0	
0	0	0	0	0	0		0	0	0		0	
QCPRun				0	0		0		0		0	

システムプロシージャ

0	0	0	0	0	0	0	0	0	0	0
TextDoc				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Unknown				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
VDO				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Load				Pass	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)				DBCC	BLKMAP	IQUTIL				
NumClients				0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
PinUserQuota				0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
PrefetchUserQuota				0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
PinUserRegisters				0	0	0	110	2		
0	0	0	0	9	0	0	0	0	0	
PfUserRegisters				0	0	0	378	0		
0	0	1	0	0	0	0	0	0	0	
ClientCountOfPinner				33333	66666	100000	4294967295			
Unknown				0	0	0	0	0	0	
Hash				0	0	0	0	0	0	
Sort				0	0	0	0	0	0	
Row				0	0	0	0	0	0	
RowColumn				0	0	0	0	0	0	
FP				0	0	0	0	0	0	
Garray				0	0	0	0	0	0	
LOB				0	0	0	0	0	0	
BTree				0	0	0	0	0	0	
BM				0	0	0	0	0	0	
BV				0	0	0	0	0	0	
Store				0	0	0	0	0	0	
Test				0	0	0	0	0	0	
DBCC				0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	
Run				0	0	0	0	0	0	
QCPRun				0	0	0	0	0	0	
TextDoc				0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	
Unknown				0	0	0	0	0	0	
VDO				0	0	0	0	0	0	
Load				0	0	0	0	0	0	
0										

例 2 :

20 分後にバッファマネージャ (メインおよびテンポラリ) の出力を表示します。

```
sp_iqsysmon '00:20:00', 'mbufman tbufman'
```


=====
Buffer Manager (Main)
=====

STATS-NAME			TOTAL	NONE	TXTPPOS	TXTDOC	CMPACT		
BTREEV	BTREEF	BV	VDO	DBEXT	DBID	SORT	STORE	GARRAY	
Finds			80137	0	0	0	0	9046	
3307	0	20829	0	0	0	0	275		
Hits			80090	0	0	0	0	9015	
3291	0	20829	0	0	0	0	275		
Hit%			99.9	0	0	0	0	99.7	
99.5	0	100	0	0	0	0	100		
FalseMiss			26469	0	0	0	0	0	
63	40	0	1097	0	0	0	0	0	0
UnOwnRR			48	0	0	0	0	31	
16	0	1	0	0	0	0	0		
Cloned			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
Creates			1557	0	0	0	0	60	
179	0	256	0	0	0	0	58		
Destroys			546	0	0	0	0	0	
12	21	0	6	0	0	0	0	29	
Dirtyies			7554	0	0	0	0	1578	
585	0	0	0	0	0	0	0		
RealDirtyies			2254	0	0	0	0	0	
117	180	0	542	0	0	0	0	58	
PrefetchReqs			80	0	0	0	0	0	
0	0	0	74	0	0	0	0	0	0
PrefetchNotInMem			1	0	0	0	0	0	
0	0	0	1	0	0	0	0	0	0
PrefetchInMem			1466	0	0	0	0	0	
0	0	0	1466	0	0	0	0	0	0
Reads			48	0	0	0	0	31	
16	0	1	0	0	0	0	0		
PReadBlks			114	0	0	0	0	0	
80	32	0	2	0	0	0	0	0	0
PReadKB			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
ReReads			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
Writes			2002	0	0	0	0	104	
163	0	538	0	0	0	0	29		
PWriteBlks			6506	0	0	0	0	210	
326	0	1115	0	0	0	0	58		
PWriteKB			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
GrabbedDirty			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO			0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)			BARRAY	BLKMAP	HASH	CKPT	BM		
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG		

システムプロシージャ

Finds				2681	8329	0	0	0	35670
0	0	0	0	0	0	0	0	0	0
Hits				2681	8329	0	0	0	35670
0	0	0	0	0	0	0	0	0	0
Hit%				100	100	0	0	0	100
0	0	0	0	0	0	0	0	0	0
FalseMiss				84	8329	0	0	0	16856
0	0	0	0	0	0	0	0	0	0
UnOwnRR				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Cloned				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Creates				108	358	0	0	0	538
0	0	0	0	0	0	0	0	0	0
Destroys				0	126	0	0	0	59
0	0	0	0	0	0	0	0	0	0
Dirtyies				512	235	0	0	0	4644
0	0	0	0	0	0	0	0	0	0
RealDirtyies				128	593	0	0	0	636
0	0	0	0	0	0	0	0	0	0
PrefetchReqs				6	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PrefetchNotInMem				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PrefetchInMem				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Reads				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PReadBlks				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PReadKB				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReReads				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Writes				128	466	0	0	0	574
0	0	0	0	0	0	0	0	0	0
PWriteBlks				239	3728	0	0	0	830
0	0	0	0	0	0	0	0	0	0
PWriteKB				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
GrabbedDirty				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
STATS-NAME				VALUE					
BusyWaits				98					
LRUNumLocks				401784					
LRUNumSpinsWoTO				0	0%				
LRUNumSpinLoops				4315					
LRUNumTimeOuts				4315	-1.10%				
BmapHTNumLocks				0					
BmapHTNumWaits				0	0%				
CacheTeamTimesWoken				182					

CacheTeamNumAsleep	10					
BmapHTMaxEntries	4096					
BmapHTNEntries	27					
BmapHTNInserts	31954					
BmapHTNCollisn	203					
BmapHTNFind	51419					
BmapHTNHits	19576					
BmapHTNHits1	19550					
BmapHTNHits2	26					
BmapHTNClears	31933					
BmapHTNLChain	1					
BmapHTNRehash	0					
BlockmapMutexsNLocks	0					
BlockmapMutexsNWait	0					
BlockmapUID	3659					
BlockmapUIDnallocs	3652					
BlockmapRegEver	31851					
BlockmapRegisters	31844					
BufHTNBuckets	4608					
BufHTNEntries	1208					
BufHTNw2orMore	158					
BufHTMaxBucketSize	19					
BufHTNFoiledOps	0					
IONumLocks	0					
IONumWait	0	0%				
=====						
Buffer Manager (Temporary)						
=====						
STATS-NAME	TOTAL	NONE	TXTTPOS	TXTDOC	CMPACKT	
BTREEV	BV	DBEXT	DBID	SORT	STORE	GARRAY
BTREEF						
Finds	31656	0	0	0	0	0
0	0	0	0	1022	0	0
Hits	31655	0	0	0	0	0
0	0	0	0	1022	0	0
Hit%	100	0	0	0	0	0
0	0	0	0	100	0	0
FalseMiss	23898	0	0	0	0	0
0	0	0	0	0	0	0
UnOwnRR	0	0	0	0	0	0
0	0	0	0	0	0	0
Cloned	0	0	0	0	0	0
0	0	0	0	0	0	0
Creates	5682	0	0	0	0	0
0	0	0	0	1048	716	0
Destroys	5670	0	0	0	0	0
0	0	0	0	821	17	0
Dirtyes	6702	0	0	0	0	0
0	0	0	0	379	0	0
RealDirtyes	5692	0	0	0	0	0
0	0	0	0	1048	716	0
PrefetchReqs	1	0	0	0	0	0
0	0	0	0	0	0	0
PrefetchNotInMem	1	0	0	0	0	0
0	0	0	0	0	0	0

システムプロシージャ

PrefetchInMem				446	0	0	0	0	0
0	0	0	0	0	0	0	446	0	0
Reads				2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PReadBlks				4096	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PReadKB				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReReads				2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Writes				10	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PWriteBlks				80	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PWriteKB				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
GrabbedDirty				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

STATS-NAME (cont'd)			BARRAY	BLKMAP	HASH	CKPT	BM
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG
Finds			0	8569	124	0	21939
0	0	0	0	0	2	0	0
Hits			0	8569	124	0	21939
0	0	0	0	0	1	0	0
Hit%			0	100	100	0	100
0	0	0	0	0	50	0	0
FalseMiss			0	8569	0	0	15328
0	0	0	0	0	1	0	0
UnOwnRR			0	0	0	0	0
0	0	0	0	0	0	0	0
Cloned			0	0	0	0	0
0	0	0	0	0	0	0	0
Creates			0	1440	777	0	1041
0	0	0	0	0	0	660	0
Destroys			0	1434	777	0	123
0	0	0	0	0	0	660	0
Dirtyes			0	0	0	0	6323
0	0	0	0	0	0	0	0
RealDirtyes			0	1440	777	0	1051
0	0	0	0	0	0	660	0
PrefetchReqs			0	0	0	0	0
0	0	0	0	0	1	0	0
PrefetchNotInMem			0	0	0	0	0
0	0	0	0	0	1	0	0
PrefetchInMem			0	0	0	0	0
0	0	0	0	0	0	0	0
Reads			0	0	0	0	0
0	0	0	0	2	0	0	0
PReadBlks			0	0	0	0	0
0	0	0	0	0	4096	0	0
PReadKB			0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
ReReads				0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0
Writes				0	0	0	0	0	10
0	0	0	0	0	0	0	0	0	0
PWriteBlks				0	0	0	0	0	80
0	0	0	0	0	0	0	0	0	0
PWriteKB				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
GrabbedDirty				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO				0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
STATS-NAME				VALUE					
BusyWaits				0					
LRUNumLocks				136253					
LRUNumSpinsWoTO				0	0%				
LRUNumSpinLoops				2780					
LRUNumTimeOuts				2780	-0.02%				
BmapHTNumLocks				0					
BmapHTNumWaits				0	0%				
CacheTeamTimesWoken				1					
CacheTeamNumAsleep				10					
BmapHTMaxEntries				4096					
BmapHTNEntries				17					
BmapHTNInserts				2334					
BmapHTNCollisn				0					
BmapHTNFind				183					
BmapHTNHits				0					
BmapHTNHits1				0					
BmapHTNHits2				0					
BmapHTNClears				2327					
BmapHTNLChain				0					
BmapHTNRehash				0					
BlockmapMutexsNLocks				0					
BlockmapMutexsNWaits				0					
BlockmapUID				2380					
BlockmapUIDnallocs				2335					
BlockmapRegEver				2344					
BlockmapRegisters				2334					
BufHTNBuckets				4608					
BufHTNEntries				24					
BufHTNw2orMore				0					
BufHTMaxBucketSize				3					
BufHTNFoiledOps				0					
IONumLocks				0					
IONumWaits				0	0%				

例 3 :

20 分後にバッファプール (メインおよびテンポラリ) の出力を表示します。

システムプロシージャ

```

sp_iqsysmon '00:20:00', 'mbufpool tbufpool'

=====
Buffer Pool (Main)
=====

STATS-NAME          TOTAL      NONE  TXTTPOS  TXTDOC  CMPACT
BTREEV  BTREEF      BV      VDO  DBEXT      DBID      SORT  STORE  GARRAY
MovedToMRU          68731      0      0      0      0      0      9094
2767      0      21083      0      0      0      0      303
MovedToWash          0      0      0      0      0      0      0
0      0      0      0      0      0      0      0
RemovedFromLRU      67564      0      0      0      0      0
9020      2597      0      20830      0      0      0      0      274
RemovedFromWash      11457      0      0      0      0      0
1559      356      0      2189      0      0      0      0      68
RemovedInScanMode      0      0      0      0      0      0
0      0      0      0      0      0      0      0
MovedToPSList          0      0      0      0      0      0
0      0      0      0      0      0      0      0
RemovedFromPSList      0      0      0      0      0      0
0      0      0      0      0      0      0      0

STATS-NAME (cont'd)  BARRAY  BLKMAP  HASH  CKPT  BM
TEST  CMID  RIDCA  LOB  LVCRID  FILE  RIDMAP  RVLOG
MovedToMRU          2169  8561      0      0      24754
0      0      0      0      0      0      0
MovedToWash          0      0      0      0      0
0      0      0      0      0      0
RemovedFromLRU      2065  8330      0      0      24448
0      0      0      0      0      0
RemovedFromWash      233  1437      0      0      5615
0      0      0      0      0      0
RemovedInScanMode      0      0      0      0      0
0      0      0      0      0      0
MovedToPSList          0      0      0      0      0
0      0      0      0      0      0
RemovedFromPSList      0      0      0      0      0
0      0      0      0      0      0

STATS-NAME          VALUE
Pages                2787
InUse                1208 ( 43.3% )
Dirty                 11 ( 0.4% )
Pinned                19 ( 0.7% )
Flushes                0
FlushedBufferCount    0
GetPageFrame          1605
GetPageFrameFailure    0
GotEmptyFrame         1605
Washed                 0
TimesSweepersWoken    0
PriorityWashed          0
NPrioritySweepersWoken 0
washTeamSize           10
WashMaxSize            455 ( 16.3% )

```

```

washNBuffers          455 ( 16.3% )
washNDirtyBuffers     0 ( 0.0% )
washSignalThreshold   46 ( 1.7% )
washNActiveSweepers  0
NPriorityWashBuffers  0
NActivePrioritySweepers 0
washIntensity         0
FlushAndEmpties      0
EmptiedBufferCount   0
EmptiedSkippedCount  0
EmptiedWriteCount    0
EmptiedErrorCount    0
nAffinityTotal        0 ( 0.0% )
nAffinityArea         0 ( 0.0% )

```

=====
Buffer Pool (Temporary)
=====

STATS-NAME	BV	TOTAL	NONE	TXTPPOS	TXTDOC	CMPACT
BTREEV BTREEF		VDO DBEXT	DBID	SORT	STORE	GARRAY
MovedToMRU		30514	0	0	0	0
0 0	0	0	0	1218	696	0
MovedToWash		258	0	0	0	0
0 0	0	0	0	0	256	0
RemovedFromLRU		30506	0	0	0	0
0 0	0	0	0	1218	694	0
RemovedFromWash		30503	0	0	0	0
0 0	0	0	0	1218	694	0
RemovedInScanMode		0	0	0	0	0
0 0	0	0	0	0	0	0
MovedToPSList		0	0	0	0	0
0 0	0	0	0	0	0	0
RemovedFromPSList		0	0	0	0	0
0 0	0	0	0	0	0	0

STATS-NAME (cont'd)	BARRAY	BLKMAP	HASH	CKPT	BM	
TEST CMID RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG	
MovedToMRU		0	8575	124	0	19898
0 0	0	0	3	0	0	0
MovedToWash		0	0	0	0	0
0 0	0	0	2	0	0	0
RemovedFromLRU		0	8569	124	0	19898
0 0	0	0	3	0	0	0
RemovedFromWash		0	8569	124	0	19898
0 0	0	0	0	0	0	0
RemovedInScanMode		0	0	0	0	0
0 0	0	0	0	0	0	0
MovedToPSList		0	0	0	0	0
0 0	0	0	0	0	0	0
RemovedFromPSList		0	0	0	0	0
0 0	0	0	0	0	0	0

```

STATS-NAME          VALUE
Pages                2787
InUse                24 ( 0.9% )

```

システムプロシージャ

Dirty	17	(0.6%)
Pinned	4	(0.1%)
Flushes	0	
FlushedBufferCount	0	
GetPageFrame	5684	
GetPageFrameFailure	0	
GotEmptyFrame	5684	
Washed	0	
TimesSweepersWoken	0	
PriorityWashed	0	
NPrioritySweepersWoken	0	
washTeamSize	10	
WashMaxSize	455	(16.3%)
washNBuffers	20	(0.7%)
washNDirtyBuffers	13	(0.5%)
washSignalThreshold	46	(1.7%)
washNActiveSweepers	0	
NPriorityWashBuffers	0	
NActivePrioritySweepers	0	
washIntensity	0	
FlushAndEmpties	0	
EmptiedBufferCount	0	
EmptiedSkippedCount	0	
EmptiedWriteCount	0	
EmptiedErrorCount	0	
nAffinityTotal	0	(0.0%)
nAffinityArea	0	(0.0%)

例 4 :

20 分後にプリフェッチマネージャ (メインおよびテンポラリ) の出力を表示します。

```
sp_iqsysmon '00:20:00', 'mprefetch tprefetch'
```

```
=====
Prefetch Manager (Main)
=====

STATS-NAME                VALUE
PFMgrNThreads             10
PFMgrNSubmitted           81
PFMgrNDropped             0
PFMgrNValid               0
PFMgrNRead                1
PFMgrNReading             0
PFMgrCondVar              Locks 0  Lock-Waits 0 ( 0.0% )  Signals 0
Broadcasts 2  Waits 2

=====
Prefetch Manager (Temporary)
=====

STATS-NAME                VALUE
PFMgrNThreads             10
PFMgrNSubmitted           1
```



```
PFMgrNDropped          0
PFMgrNValid            0
PFMgrNRead             1
PFMgrNReading          0
PFMgrCondVar           Locks 0 Lock-Waits 0 ( 0.0% ) Signals 0
Broadcasts 2 Waits 2
```

例 5 :

20 分後に IQ ストアフリーリスト (メインおよびテンポラリ) の出力を表示します。

```
sp_iqsysmon '00:20:00', 'mfreelist tfreelist'
```

```
=====
IQ Store (Main) Free List
=====

STATS-NAME              VALUE
FLBitCount              74036
FLIsOutOfSpace          NO
FLMutexLocks            0
FLMutexWaits            0 ( 0.0% )

=====
IQ Store (Temporary) Free List
=====

STATS-NAME              VALUE
FLBitCount              4784
FLIsOutOfSpace          NO
FLMutexLocks            0
FLMutexWaits            0 ( 0.0% )
```

例 6 :

20 分後にメモリマネージャ、スレッドマネージャ、CPU 使用率、トランザクションマネージャの出力を表示します。

```
sp_iqsysmon '00:20:00', 'memory threads cpu txn'
```

```
=====
Memory Manager
=====

STATS-NAME              VALUE
MemAllocated            67599536 ( 66015 KB )
MemAllocatedMax         160044816 ( 156293 KB )
MemAllocatedEver        1009672456 ( 986008 KB )
MemNAllocated           77309
MemNAllocatedEver       914028
MemNTimesLocked         0
MemNTimesWaited         0 ( 0.0 %)

=====
Thread Manager
=====
```

システムプロシージャ

```

STATS-NAME                               VALUE
ThrNumOfCpus                             4
ThreadLimit                              99
ThrNumThreads                             98      ( 99.0 %)
ThrReserved                              15      ( 15.2 %)
ThrNumFree                                55      ( 55.6 %)
NumThrUsed                                44      ( 44.4 %)
UsedPerActiveCmd                          22
ThrNTeamsInUse                            5
ThrMaxTeams                               7
NumTeamsAlloc                             238
TeamThrAlloc                              421
SingleThrAlloc                            492
ThrMutexLocks                             0
ThrMutexWaits                             0      ( 0.0 %)

```

===== CPU time statistics =====

```

STATS-NAME                               VALUE
Elapsed Seconds                           59.65   ( 25.0 %)
CPU User Seconds                           37.79   ( 15.8 %)
CPU Sys Seconds                             1.89    ( 0.8 %)
CPU Total Seconds                          39.68   ( 16.6 %)

```

===== Transaction Manager =====

```

STATS-NAME                               VALUE
TxnMgrNPending                            0
TxnMgrNBlocked                            2
TxnMgrNWaiting                            0
TxnMgrPCcondvar                            Locks    0      Lock-Wait 0 ( 0.0 %)
Signals 0 Broadcasts 2 Waits 2
TxnMgrTxnIDseq                            407
TxnMgrtxncblock                           Locks    0      Lock-Wait 0 ( 0.0 %)
TxnMgrVersionID                           0
TxnMgrOAVI                                0
TxnMgrVersionLock                         Locks    0      Lock-Wait 0 ( 0.0 %)
Signals 0 Broadcasts 0 Waits 0

```

例 7 :

20 分後にサーバコンテキストおよびカタログ統計の出力を表示します。

```
sp_iqsysmon '00:20:00', 'context catalog'
```

===== Context Server statistics =====

```

STATS-NAME                               VALUE
StCntxNumConns                            1

```

```

StCntxNResource          16
StCntxNOrigResource     18
StCntxNWaiting          0
StCntxNWaited           0
StCntxNAdmitted         1116
StCntxLock              Locks 0 Lock-Waits 0 ( 0.0 %)
StCntxCondVar           Locks 0 Lock-Waits 0 ( 0.0 %)

=====
Catalog, DB Log, and Repository statistics
=====

STATS-NAME                VALUE
CatalogLock              RdLocks 0   RdWaits 0 ( 0.0 %) RdTryFails
0 WrLocks 30037 WrWaits 0 ( 0.0 %) WrTryFail 0
DbLogMLock               Locks 0 Lock-Waits 0 ( 0.0 %)
DbLogSLock               Locks 0 Lock-Waits 0 ( 0.0 %)
RepositoryNList           0
RepositoryLock           Locks 1 SpinsWoTO 0 ( 0.0 %) Spins
0 TimeOuts 0 ( 0.0 %)

```

例 8 :

20 分後に、IQ RLV インメモリストアおよび LMA (Large Memory Allocator) の統計の出力を表示します。

```
sp_iqsysmon '00:20:00', 'rlv lma'
```

```
=====
IQ In-Memory Store
=====
```

```

STATS-NAME                VALUE
RLV Memory Limit          2048 MB
RLV Memory Used           0 MB
RLV Chunks Used           0

```

```
=====
Large Memory Allocator
=====
```

```

STATS-NAME                VALUE
Large Memory Space        2048 MB
Large Memory Max Fle      512 MB
Large Memory Num Fle      0
Large Memory Flexibl      0.5
Large Memory Flexibl      0 MB
Large Memory Inflexi      0.9
Large Memory Inflexi      0 MB
Large Memory Anti-St      0.5
Large Memory Num Con      0

```

sp_iqtable プロシージャ

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 ] )

table_type :
    TEMP
    | VIEW
    | ALL
    | any_other_value
```

構文 2

```
sp_iqtable [table_name='tablename'],
[table_owner='tableowner' ], [table_type='tabletype' ]
```

パラメータ

- **table_type** –
 - **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、グローバルテンポラリテーブル、ビュー
 - **any_other_value** – IQ テーブル

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

構文 1 では、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 および sp_iqtable NULL, dbo, SYSTEM とします。

注意： 構文 1 では、*table_type* の値である ALL と VIEW を一重引用符で囲む必要があります。

構文 2 では、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。

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 SAP 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。

カラム名	説明
table_name	テーブル名。
table_type	<ul style="list-style-type: none"> • BASE – ベーステーブル。 • MAT VIEW – マテリアライズドビュー (SA テーブルのみ)。 • GBL TEMP – グローバルテナポラリテーブル。 • 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、SAP Sybase IQ ユーザが使用することはできない)。 • VIEW – ビュー。
table_owner	テーブルの所有者
server_type	<p>IQ – IQ ストアで作成されたオブジェクト。</p> <p>SA – SA ストアで作成されたオブジェクト。</p> <p>ビューはすべて SA ストアで作成される。</p>
location	<p>TEMP – IQ テンポラリストア。</p> <p>MAIN – IQ ストア。</p> <p>SYSTEM – カタログストア。</p>
dbspace_id	DB 領域を識別する番号。
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。
PartitionType	<p>分割されている場合は、パーティションのタイプを示す。</p> <ul style="list-style-type: none"> • ハッシュ範囲 • 範囲 • ハッシュ • なし
isRLV	テーブルが RLV 対応であることを示す。

参照：

- sp_iqcolumn プロシージャ (457 ページ)
- sp_iqconstraint プロシージャ (465 ページ)
- sp_iqdatatype プロシージャ (473 ページ)
- sp_iqevent プロシージャ (496 ページ)
- sp_iqhelp プロシージャ (502 ページ)
- sp_iqindex および sp_iqindex_alt プロシージャ (509 ページ)
- sp_iqpkeys プロシージャ (553 ページ)
- sp_iqprocparm プロシージャ (558 ページ)
- sp_iq_reset_identity プロシージャ (570 ページ)
- sp_iqview プロシージャ (632 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqtable プロシージャ例

sp_iqtable 出力例。

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
Departments	BASE	GROUPO

Server_type	Location	dbspace_id
IQ	Main	16387

isPartitioned	備考	table_constraints
N	スポーツ用品会社のさまざまな部署の名前と代表を含む	(NULL)

PartitionType	isRlv
なし	F

次の構文は、いずれも GROUPO が所有するすべてのテーブルを返します。

```
sp_iqtable NULL, GROUPO
sp_iqtable table_owner='GROUPO'
```

Table_name	Table_type	Table_owner	Server_type	Location
Contacts	BASE	GROUPO	IQ	Main
Customers	BASE	GROUPO	IQ	Main
Departments	BASE	GROUPO	IQ	Main
Employees	BASE	GROUPO	IQ	Main
FinancialCodes	BASE	GROUPO	IQ	Main
FinancialData	BASE	GROUPO	IQ	Main
Products	BASE	GROUPO	IQ	Main
SalesOrders	BASE	GROUPO	IQ	Main
SalesOrderItems	BASE	GROUPO	IQ	Main

dbspace_id	isPartitioned	備考	table_constraints
16387	N	会社が連絡先情報を保有したいと考えるすべての人々の名前、住所、および電話番号	(NULL)
16387	N	スポーツ用品会社の顧客	(NULL)
16387	N	スポーツ用品会社のさまざまな部署の名前と代表を含む	(NULL)
16387	N	名前、給与、入社日、誕生日などの情報を含む	(NULL)
16387	N	スポーツ用品会社の収入と支出のタイプ	(NULL)
16387	N	スポーツ用品会社の収入と支出	(NULL)
16387	N	スポーツ用品会社で販売される製品	(NULL)
16387	N	注文に含まれる個々の項目	(NULL)
16387	N	顧客がスポーツ用品会社に発行した注文	(NULL)

PartitionType	isRlvd
なし	F
なし	F

PartitionType	isRlvd
なし	F
なし	F
なし	F
なし	F
なし	F
なし	F
なし	F

sp_iqtablesize プロシージャ

指定したテーブルのサイズを返します。

構文

```
sp_iqtablesize ( table_owner table_name )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- MANAGE ANY DBSPACE
- ALTER ANY TABLE
- テーブルを所有している

備考

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。

カラム名	説明
Ownername	所有者の名前
Tablename	テーブルの名前
Columns	テーブルのカラム数
KBytes	物理テーブルサイズ (KB)

カラム名	説明
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
RlvLogPages	ディスク上に RLV テーブルログ情報を保持するために必要な IQ ページの数
RlvLogKBytes	RLV テーブルログのサイズ (KB 単位)

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、128KB 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： SAP 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 を読み書きします。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となります。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、通常ほぼ 100% になります。SAP 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.t1')
```

Ownername	Tablename	Columns
DBA	t1	3

(続き) KBytes	Pages	CompressedPages
192	5	4

(続き) NBlocs	RlvLogPages	RlvLogBytes
24	96	12288

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqtableuse プロシージャ

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

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。 MONITOR システム権限。

備考

SYSTEM で作成されたテーブルはレポートされません。

表 112 : 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：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)
- sp_iqworkmon プロシージャ (637 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqtransaction プロシージャ

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
```

適用対象

シンプレックスとマルチプレックス。

権限

MONITOR システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqtransaction は、SAP 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。Name、Userid、ConnHandle の各カラムは、**Name**、**Userid**、**Number** の各接続プロパティにそれぞれ対応しています。ローは TxnID の順に並べられます。

sp_iqtransaction の出力には、進行中のトランザクションがない接続は含まれません。すべての接続を含めるには、**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_iqlocks** を使用することをおすすめします。

カラム名	説明
Name	アプリケーションの名前。
Userid	接続のユーザ ID。

カラム名	説明
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、 begin transaction の間に割り当てられる。トランザクション ID 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、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。
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の場合、CmtID は 0。
VersionID	<p>シンプレックスノードおよびマルチプレックスノードの場合、値 0 は、トランザクションがバージョン管理されておらず、VersionID が割り当てられていないことを示す。</p> <p>マルチプレックスコーディネータでは、トランザクションがテーブルロックを確立してから、VersionID が割り当てられる。マルチプレックスセカンダリサーバは、コーディネータから VersionID を受け取る。VersionID 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、SAP 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内部的に使用される。</p>
State	<p>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 SAP 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る。現行のトランザクションの状態は NONE、ACTIVE、ROLLING_BACK、ROLLED_BACK、COMMITTING、COMMITTED、および APPLIED。</p> <p>NONE、ROLLING_BACK、ROLLED_BACK、COMMITTING、および APPLIED は、非常に短い一時的な状態。</p> <p>ACTIVE は、トランザクションがアクティブであることを示す。</p> <p>COMMITTED は、トランザクションが完了し、APPLIED への遷移待ちである状態を示す。APPLIED の状態では、すべてのトランザクションに認識されないバージョンは、ガーベジコレクションの対象となる。</p> <p>トランザクションの状態が ROLLED_BACK、COMMITTED、または APPLIED になると、開いているカーソルで保持されているロック以外のロックは所有できなくなる。</p>
ConnHandle	接続の ID 番号。

カラム名	説明
IQConnID	.iqmsg ファイル内のすべてのメッセージの一部として含まれている 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ある。
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 (KB 単位)。
MainTableKBDr	このトランザクションによって削除済みの IQ ストアの領域のうち、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領域の容量 (KB 単位)。
TempTableKBCr	このトランザクションが IQ テンポラリテーブルのデータの格納用に作成した IQ テンポラリストアの領域 (KB 単位)。
TempTableKBDr	このトランザクションによって削除された IQ テンポラリテーブルの領域のうち、IQ カーソルで表示されているか、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 IQ テンポラリストアのディスク上に残っている領域の容量 (KB 単位)。
TempWorkSpaceKB	ステータスが ACTIVE であるトランザクションでは、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 (ソート、ハッシュ、テンポラリビットマップなど)。この数字は、 sp_iqtransaction を実行するタイミングによって変わる。たとえば、クエリエンジンがテンポラリキャッシュに 60MB を作成しても、クエリ処理の継続中にその大部分をすぐに解放することがある。したがって、クエリが完了した後に sp_iqtransaction を実行すると、このカラムに表示される数字が大幅に小さくなる。トランザクションがアクティブでなくなると、このカラムはゼロになる。 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。
TxnCreateTime	トランザクションの開始時刻。すべての SAP Sybase IQ トランザクションは、アクティブな接続が確立されるか、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開始される。

カラム名	説明
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 SAP Sybase IQ カーソルの数。トランザクションが ACTIVE である場合、そのトランザクションで作成されたオープンしているカーソルの数を示す。トランザクションのステータスが 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解放されることがある。したがって、この番号はトランザクション内でユーザが作成したセーブポイントの数を示すものではない。
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。これは実装の詳細であるため、ユーザ作成のセーブポイントの数が反映されていない場合がある。
MPXServerName	アクティブなトランザクションが、ノード間通信 (INC) 接続からのトランザクションであるかどうかを示す。INC 接続からのトランザクションの場合、この値は、そのトランザクションが開始されたマルチプレックスサーバの名前となる。INC 接続からのトランザクションでない場合、NULL となる。トランザクションがアクティブでない場合は、常に NULL となる。
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。関連付けられているトランザクションがない場合は 0 (ゼロ)。
VersioningType	トランザクションのスナップショットバージョン管理のタイプ。テーブルレベル (デフォルト) またはローレベルのいずれか。ローレベルのスナップショットバージョン管理 (RLV) は RLV 対応テーブルにのみ適用される。トランザクションが開始されると、この値は変更できない。
Blocking	接続ブロッキングが有効 (True) か無効 (False) かを示す。 BLOCKING データベースオプションを使用して接続ブロッキングを設定する。true の場合は、トランザクションがブロックされる。つまりトランザクションは、競合しているロックが解放されるまで待機してからロック要求を再試行する。
BlockingTimeout	ロックの競合が解消されるまでトランザクションが待機する時間 (ミリ秒) を示す。 BLOCKING_TIMEOUT データベースオプションを使用して、タイムアウトのスレッシュホールドを設定する。値 0 (デフォルト) は、トランザクションが無期限で待機することを示す。

例

sp_iqtransaction の出力例

```

Name      Userid  TxnID  CmtID  VersionID  State      ConnHandle  IQConnID
=====  =====  =====  =====  =====  =====  =====  =====
red2      DBA     10058   10700   10058     Active     419740283   14

MainTableKBCr      MainTableKBDr      TempTableKBCr  TempTableKBDr
=====  =====  =====  =====
              0              0              65824              0

TempWorkSpaceKB  TxnCreateTime      CursorCount  SpCount
SpNumber
=====
              0      2013-03-26 13:17:27.612              1      3      2

MPXServerName  GlobalTxnID  VersioningType  Blocking
BlockingTimeout
=====  =====  =====  =====
              (NULL)              0      Row-level      True
0

```

参照：

- sp_iqstatus プロシージャ (589 ページ)
- sp_iqversionuse プロシージャ (629 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqunusedcolumn プロシージャ

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
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR システム権限。

備考

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 113 : 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                  GROUPO
SalesOrders        CustomerID         GROUPO
SalesOrders        OrderDate          GROUPO
SalesOrders        FinancialCode      GROUPO
SalesOrders        Region             GROUPO
SalesOrders        SalesRepresentative GROUPO
SalesOrderItems    ID                  GROUPO
SalesOrderItems    LineID             GROUPO
SalesOrderItems    ProductID          GROUPO
SalesOrderItems    Quantity           GROUPO
SalesOrderItems    ShipDate           GROUPO
Contacts           ID                  GROUPO
Contacts           Surname            GROUPO
Contacts           GivenName          GROUPO ...

```

参照：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)
- sp_iqworkmon プロシージャ (637 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqunusedindex プロシージャ

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
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR システム権限。

備考

SYSTEM で作成されたテーブルまたはローカルテナ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 114 : 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        GROUPO HG
ASIQ_IDX_T450_C6_HG SalesOrders        GROUPO HG
ASIQ_IDX_T450_C4_HG SalesOrders        GROUPO HG
ASIQ_IDX_T450_C2_HG SalesOrders        GROUPO HG
ASIQ_IDX_T451_I6_HG SalesOrderItems    GROUPO HG
ASIQ_IDX_T451_C3_HG SalesOrderItems    GROUPO HG
ASIQ_IDX_T451_C1_HG SalesOrderItems    GROUPO HG
ASIQ_IDX_T452_I11_HG Contacts           GROUPO HG
ASIQ_IDX_T453_I10_HG Contacts           GROUPO HG
ASIQ_IDX_T454_I4_HG  FinancialCodes     GROUPO HG
ASIQ_IDX_T455_I5_HG  FinancialData      GROUPO HG
ASIQ_IDX_T455_C3_HG  FinancialData      GROUPO HG
ASIQ_IDX_T456_I8_HG  Products           GROUPO HG
ASIQ_IDX_T457_I4_HG  Departments        GROUPO HG

```

システムプロシージャ

```
ASIQ_IDX_T457_C3_HG Departments          GROUPO HG
ASIQ_IDX_T458_I21_HG Departments          GROUPO HG
ASIQ_IDX_T458_C5_HG  Departments          GROUPO HG
```

参照：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)
- sp_iqworkmon プロシージャ (637 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqunusedtable プロシージャ

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
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。 MONITOR システム権限。

備考

SYSTEM で作成されたテーブルとローカルテナポラリテーブルはレポートされません。

表 115 : 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          GROUPO
Contacts           GROUPO Employees            GROUPO
empl               DBA SalesOrders            GROUPO
FinancialData      GROUPO Departments          GROUPO
```

SalesOrderItems	GROUPO Products	GROUP
iq_dummy	DBA Customers	GROUPO
sale	DBA	

参照：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqworkmon プロシージャ (637 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqversionuse プロシージャ

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
```

適用対象

シンプレックスとマルチプレックス。

権限

MONITOR システム権限。そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。

備考

sp_iqversionuse システムストアプロシージャは、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している場合のトラブルシューティングに役立ちます。

領域不足状態が発生している場合、または **sp_iqstatus** で示されるマルチプレックスサーバ上のメインブロックの使用率が高い場合は、**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をリリースすることでリカバリできる領域の量を確認します。

このプロシージャは、特定のバージョンの各ユーザのローを生成します。まずコーディネータ上で **sp_iqversionuse** を実行し、解放するバージョンと、そのバージョンを使用しなくなった場合に解放される領域の大きさ (KB 単位) を調べます。接続 ID は、コーディネータに接続しているユーザの IQConn カラムに表示されま

す。セカンダリサーバで使用されているバージョンは、セカンダリサーバ名として接続 ID 0 とともに表示されます。

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 カラムは、マルチプレックス設定で使用されます。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。データベースサーバ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。

セカンダリサーバへの個別の接続を調べるには、マルチプレックスセカンダリサーバ上で **sp_iqversionuse** を実行します。他のサーバのユーザはセカンダリサーバ上に表示されません。

表 116 : sp_iqversionuse のカラム

カラム名	説明
VersionID	シンプレックスデータベースでは、VersionID としてゼロが表示される。マルチプレックスコーディネータでは、VersionID はアクティブなトランザクションの TxnID、コミット済みのトランザクションの CmtID と同じである。マルチプレックスセ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。VersionID は、マルチプレックスデータベース内のすべてのノードでデータベースバージョンをユニークに識別するために、SAP Sybase IQ のメモリ内カタログと SAP Sybase IQ トランザクションマネージャで内部的に使用される。
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

sp_iqversionuse システムプロシージャからの出力例を示します。

```

VersionID Server          IQConnID WasReported
=====
0 ab2ab_iqdemo          9          0

MinKBRelease MaxKBRelease
=====
0             0

```

次の例は、マルチプレックスの出力を示しています。最も古いバージョン 42648 がコーディネータ (*mpxw*) 上の接続 108 で使用されています。接続 108 のトランザクションがコミットまたはロールバックされると、7.9MB の領域が解放されます。コーディネータからの出力によれば、バージョン 42686 はセカンダリサーバ (*mpxq*) で使用されています。セカンダリサーバ出力では、実際の接続は接続 31 です。バージョン 42686 を解放して返される実際の領域の大きさは、最初に 42648 を解放するかどうかによって異なります。

コーディネータ上のバージョン 42715 および 42728 に対する WasReported は 0 です。これは、まだレプリケートされていない新しいバージョンであるためです。バージョン 42728 はセカンダリサーバ出力に含まれていないため、セカンダリサーバではまだ使用されていません。

sp_iqversionuse がコーディネータ *mpxw* 上で実行された場合、次の出力が返されます。

```
call dbo.sp_iqversionuse
```

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42648	'mpxw'	108	1	7920	7920
42686	'mpxq'	0	1	7920	304
42702	'mpxq'	0	1	0	688
42715	'mpxq'	0	0	0	688
42728	'mpxq'	0	0	0	688

sp_iqversionuse がセカンダリサーバ (*mpxq*) 上で実行された場合、次の出力が返されます。

```
call dbo.sp_iqversionuse
```

VersionID	Server	IQConn	WasReported	MinKBRelease	MaxKBRelease
42686	'mpxq'	31	1	0	0
42715	'mpxq'	00	1	0	0

参照：

- `sp_iqstatus` プロシージャ (589 ページ)
- `sp_iqtransaction` プロシージャ (621 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

sp_iqview プロシージャ

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])
```

view_type:

```
SYSTEM  
| ALL  
| any_other_value
```

構文 2

```
sp_iqview [view_name='viewname'],  
[view_owner='viewowner' ],[view_type='viewtype' ]
```

パラメータ

- **view_type** –
 - **SYSTEM** – システムビュー
 - **ALL** – ユーザビューとシステムビュー
 - **any_other_value** – ユーザビュー

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

構文 1 `sp_iqview NULL,NULL,SYSTEM` では、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'` とします。

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 では、パラメータは一重引用符で囲んで、どのような順番で指定しても構いません。

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 117 : `sp_iqview` のカラム

カラム名	説明
view_name	ビューの名前
view_owner	ビューの所有者
view_def	CREATE VIEW 文で指定されたビューの定義
remarks	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せん。

参照：

- `sp_iqcolumn` プロシージャ (457 ページ)
- `sp_iqconstraint` プロシージャ (465 ページ)
- `sp_iqdatatype` プロシージャ (473 ページ)
- `sp_iqevent` プロシージャ (496 ページ)
- `sp_iqhelp` プロシージャ (502 ページ)
- `sp_iqindex` および `sp_iqindex_alt` プロシージャ (509 ページ)
- `sp_iqpkeys` プロシージャ (553 ページ)
- `sp_iqprocparm` プロシージャ (558 ページ)
- `sp_iq_reset_identity` プロシージャ (570 ページ)
- `sp_iqtable` プロシージャ (614 ページ)
- データベースが使用するセキュリティモデルの特定 (432 ページ)

`sp_iqview` プロシージャ例

`sp_iqview` を使用する場合は、次の例を参照してください。

次の構文は、いずれもビュー `ViewSalesOrders` 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 `GROUP0` が所有するすべてのビューを返します。

```
sp_iqview NULL, GROUP0
```

```
sp_iqview view_owner='GROUP0'
```

view_name	view_owner	view_def	remarks
ViewSalesOrders	GROUPO	Create views GROUPO , ViewSalesOrders(ID, LineID, ProductID, Quantity, OrderDate, Ship-Date, Region, SalesRepresentativeName	(NULL)

sp_iqwho プロシージャ

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 ] ]
```

パラメータ

- **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** は指定された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** は指定された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** – オプションであり、最初のパラメータが指定されたときのみ指定できます。*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。“user” 以外の値が *arg-type* に指定された場合、**sp_iqwho** は、エラー “Invalid parameter” を返します。

arg-type 値を二重引用符で囲んでください。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

備考

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。

カラム名	説明
ConnHandle	SA 接続ハンドル
IQConnID	SAP 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。
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。
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。
IQCmdType	接続から発行された SAP Sybase IQ コマンドのタイプ。コマンドを発行しない場合は NONE。
QIdle	この接続を通じて最後の SAP 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 SAP 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る。
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る。
Q Cursors	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。
QThreads	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、QThreads の最小値は 1。
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。
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。

表 118 : 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。SAP Sybase IQ には適用されないため省略される。

sp_who カラム	sp_iqwho カラム
spid	ConnHandle、IQConnID
status	QIdle、SAIdle
loginame	Userid
origname	ユーザのエイリアス。SAP Sybase IQ には適用されないため省略される。
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ない。
blk_spid	BlockedOn
dbname	SAP 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、省略される。
cmd	ReqType、IQCmdType
block_xloid	BlockUserid

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* は排他的であり、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。

SAP 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。次に例を示します。

```
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 が開いた接続に関する情報を表示します。

構文	出力
sp_iqwho	すべてのアクティブな接続を表示する。
sp_iqwho 3	接続 3 に関する情報を表示する。
sp_iqwho “DBA”	ユーザ DBA が開いた接続を表示する。

構文	出力
sp_iqwho 3, "user"	3 をユーザ名として解釈し、ユーザ 3 が開いた接続を表示する。 ユーザ 3 が存在しない場合、エラー "User 3 does not exist" を返す。
sp_iqwho non-existing-user	エラー "User non-existing-user does not exist" を返す。
sp_iqwho 3, "xyz"	エラー "Invalid parameter: xyz" を返す。

sp_iqwho プロシージャの例

sp_iqwho を使用する場合は、この例を参照してください。

すべてのアクティブな接続を表示します。

ConnHandle	IQConnID	Userid	ReqType	IQCmdType	Blocked
On	BlockUserid	IQCursors			
12	118	DBA	CURSOR_OPEN	IQUTILITYOPENCURSOR	0
(NULL)	0				
13	119	shweta	DO_NOTHING	NONE	0
(NULL)	0				
IQThreads	IQIdle	SAIdle	TempTableSpaceKB	TempWorkSpaceKB	
1	1	0	0	0	
1	16238757	470	0	0	

sp_iqwho の Adaptive Server との互換性

SAP Sybase IQ sp_iqwho ストアドプロシージャは、Adaptive Server の sp_who プロシージャが表示したカラムに対応する SAP Sybase IQ を取り込みます。

一部の Adaptive Server カラムは、SAP Sybase IQ に適用できないため省略されます。

sp_iqworkmon プロシージャ

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。sp_iqworkmon は、すべての SQL 文の情報を収集します。

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つだけ指定する場合、指定できるのは *action* だけです。次に例を示します。

```
sp_iqworkmon 'stop'
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR システム権限。

備考

パラメータ	説明
action の値	適用する制御アクションを指定する。値 <i>start</i> を指定すると、指定したモードでモニタリングがすぐに開始する。値 <i>stop</i> を指定すると、モニタリングがすぐに停止する。値 <i>status</i> (デフォルト値) を指定すると、状態が変わることなく現在のステータスが表示される。 統計は、 <i>reset</i> 引数でクリアされるまで、またはサーバが再起動されるまで保持される。サーバの再起動後、統計の収集は自動的に再開しない。 <i>start</i> を使用して再開する必要がある。
mode	制御するモニタリングの種類を指定する。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する。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る。

sp_iqworkmon を実行すると、常に結果セットも表示されます。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。

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：

- sp_iqcolumnuse プロシージャ (460 ページ)
- sp_iqindexadvice プロシージャ (513 ページ)
- sp_iqindexuse プロシージャ (525 ページ)
- sp_iqtableuse プロシージャ (620 ページ)
- sp_iqunusedcolumn プロシージャ (625 ページ)
- sp_iqunusedindex プロシージャ (627 ページ)
- sp_iqunusedtable プロシージャ (628 ページ)

カタログストアードプロシージャのアルファベット順リスト

カタログストアードプロシージャは、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。

このプロシージャを所有するのは、dbo ユーザ ID です。このプロシージャに対する EXECUTE 権限は、PUBLIC ロールが持っています。

sa_ansi_standard_packages システムプロシージャ

SQL 文で使用されている、コア以外の SQL 拡張機能に関する情報を返します。

構文

```
sa_ansi_standard_packages (
  standard
, statement
)
```

引数

- **standard** – コア拡張機能で使用する標準を指定する LONG VARCHAR パラメータ。SQL:1999 または SQL:2003。
- **statement** – 評価する SQL 文を指定する LONG VARCHAR パラメータ。

結果セット

カラム名	データ型	説明
package_id	VARCHAR(10)	機能識別子。
package_name	LONG VARCHAR	機能名。

備考

文に使用される、コア以外の拡張機能がない場合、結果セットは空です。

権限
なし。

関連する動作
なし。

sa_audit_string システムプロシージャ

文字列をトランザクションログに追加します。

構文

```
sa_audit_string( string )
```

引数

- **string** – トランザクションログに追加する VARCHAR(128) 文字列。

備考

このシステムプロシージャでは、監査が有効な場合、トランザクションログに格納されている監査情報にコメントが追加されます。この文字列は最大 128 文字までです。

権限

MANAGE AUDITING システム権限が必要です。

関連する動作
なし。

例

次の例では、sa_audit_string を使用してコメントをトランザクションログに追加します。

```
CALL sa_audit_string( 'Auditing test' );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_char_terms システムプロシージャ

CHAR 文字列を単語に分割し、各単語をローとして、その位置とともに返します。

構文

```
sa_char_terms( 'char-string' [, 'text-config-name'  
[, 'owner' ] ] )
```

パラメータ

- **char-string** – 解析する CHAR 文字列。
- **text-config-name** – 文字列の処理時に適用するテキスト設定オブジェクト。デフォルト値は 'default_char' です。
- **owner** – 指定されたテキスト設定オブジェクトの所有者。デフォルト値は DBA です。

権限
なし

備考

sa_char_terms を使用して、テキスト設定オブジェクトの設定が適用されるときに、文字列がどのように解釈されるのかを確認できます。これは、インデックスの作成中またはクエリ文字列からどの単語が削除されるのかを知りたい場合に便利です。

例

CHAR 文字列 'the quick brown fox jumped over the fence' 内の単語が返されます。

```
CALL sa_char_terms
( 'the quick brown fox jumped over the fence' );
```

表 119 : CHAR 文字列の解釈

用語	位置
the	1
quick	2
brown	3
fox	4
jumped	5
over	6
the	7
fence	8

sa_checkpoint_execute システムプロシージャ

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実行される 1 つまたは複数のユーザコマンド。シェルコマンドはシステムシェルに固有です。コマンドはセミコロン (;) で区切られます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- CHECKPOINT
- MANAGE ANY MIRROR SERVER

備考

サーバがクワイース (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 を対話型コマンドと共に使用しないでください。使用した場合、サーバは対話型コマンドが強制終了されるまで待機することになります。上書きフラグを設定し、対話型になる可能性のあるシェルコマンドのプロンプトが表示されないようにしてください。具体的には、**COPY**、**MOVE**、**DELETE** などのコマンドでプロンプトが表示される可能性があります。

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ます。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/'
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_conn_activity システムプロシージャ

サーバ上に示されている指定したデータベース接続に関して、最後に作成された SQL 文を返します。

構文

```
sa_conn_activity( [ connidparm ] )
```

引数

- connidparm** – 接続 ID 番号を指定する任意の INTEGER パラメータ。デフォルトは NULL です。

結果セット

カラム名	データ型	説明
Number	INTEGER	現在の接続の接続 ID (数値) を返します。
Name	VARCHAR(255)	現在の接続の名前を返します。 テンポラリ接続名の場合、接続名の先頭に INT: が追加されます。
Userid	VARCHAR(255)	接続のユーザ ID を返します。
DBNumber	INTEGER	データベースの ID 番号を返します。

カラム名	データ型	説明
LastReqTime	VARCHAR(255)	指定された接続において最後の要求が開始された時刻を返します。このプロパティは、イベントなどの内部接続の場合は空の文字列を返すことがあります。
LastStatement	LONG VARCHAR	現在の接続で最後に準備された SQL 文を返します。

備考

connidparm が 0 未満の場合、現在の接続についての情報が返されます。 *connidparm* を指定しない場合または NULL の場合、データベースサーバ上で実行されているすべてのデータベースに対するすべての接続に関する情報が返されます。

sa_conn_activity システムプロシージャは、接続について最後に作成された SQL 文で構成される一連の結果を返します。 *sa_conn_activity* を呼び出す前に、データベースサーバで文の記録をオンにします。文の記録をオンにするには、データベースサーバの起動時に *-zl* オプションを指定するか、次の文を実行します。

```
CALL sa_server_option('RememberLastStatement','ON');
```

このプロシージャは、データベースサーバがビジー状態のときに、各接続について作成された最後の SQL 文の情報を取得するのに便利です。この機能は、ログイン要求の代わりとして使用できます。

権限

現在の接続 ID に対してこのシステムプロシージャを実行する場合、権限は必要ありません。他の接続に対してこのシステムプロシージャを実行する場合は、SERVER OPERATOR、MONITOR、または DROP CONNECTION のシステム権限が必要です。

関連する動作

なし。

例

次の文は、 *sa_conn_activity* システムプロシージャを使用して、各接続に関しても最後に作成された SQL 文を表示します。

```
CALL sa_conn_activity( );
```

Number	Name	Userid	DBNumber	...
1,949	SQL_DBC_117acc40	DBA	0	...
1,948	setup	User1	0	...
...

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_conn_info システムプロシージャ

接続プロパティ情報をレポートします。

構文

```
sa_conn_info( [ connidparm ] )
```

引数

- connidparm** – 接続 ID 番号を指定する任意の INTEGER パラメータ。デフォルトは NULL です。

結果セット

カラム名	データ型	説明
Number	INTEGER	現在の接続の接続 ID (数値) を返します。
Name	VARCHAR(255)	現在の接続の接続 ID (数値) を返します。 テンポラリ接続名の場合、接続名の先頭に INT: が追加されます。
Userid	VARCHAR(255)	接続のユーザ ID を返します。
DBNumber	INTEGER	データベースの ID 番号を返します。
LastReqTime	VARCHAR(255)	指定された接続において最後の要求が開始された時刻を返します。このプロパティは、イベントなどの内部接続の場合は空の文字列を返すことがあります。
ReqType	VARCHAR(255)	最後の要求のタイプを返します。接続が接続プールによってキャッシュされる場合、その ReqType 値は CONNECT_POOL_CACHE になります。
CommLink	VARCHAR(255)	接続の通信リンクを返します。これは SAP Sybase IQ がサポートするネットワークプロトコルであり、同一コンピュータ接続の場合は local となります。
NodeAddr	VARCHAR(255)	クライアント/サーバ接続のクライアントアドレスを返します。

カラム名	データ型	説明
ClientPort	INTEGER	クライアントのTCP/IPポート番号を返します。接続の種類がTCP/IPでない場合は、0を返します。
ServerPort	INTEGER	データベースサーバのTCP/IPポート番号または0を返します。
BlockedOn	INTEGER	現在の接続がブロックされていない場合は0を返し、ブロックされている場合はロック競合によってブロックされた接続の接続番号を返します。
LockRowID	UNSIGNED BIGINT	ロックされたローの識別子を返します。 ローに関連付けられているロックで接続が待機していない場合(つまり、ロックで待機していない場合、または関連付けられているローがないロックで待機している場合)、LockRowIDはNULLです。
LockIndexID	INTEGER	ロックされたインデックスの識別子を返します。 LockTableのテーブルのすべてのインデックスにロックが関連付けられている場合、LockIndexIDは-1です。インデックスに関連付けられているロックで接続が待機していない場合(つまり、ロックで待機していない場合、または関連付けられているインデックスがないロックで待機している場合)、LockIndexIDはNULLです。
LockTable	VARCHAR(255)	接続が現在ロックを待機している場合、そのロックに関連付けられているテーブルの名前を返します。それ以外の場合は、空の文字列を返します。
UncommitOps	INTEGER	コミットされていないオペレーションの数を返します。
ParentConnection	INTEGER	データベースオペレーション(データベースのバックアップや作成など)を実行するためにテンポラリ接続を作成した接続の接続IDを返します。その他のタイプの接続について、このプロパティはNULLを返します。

備考

connidparm が 0 未満の場合、現在の接続について、接続プロパティで構成される結果セットが返されます。*connidparm* を指定しない場合または NULL の場合、

データベースサーバ上で実行されているすべてのデータベースに対するすべての接続について、接続プロパティが返されます。

ブロックの場合には、このプロシージャが返す `BlockedOn` 値によって、どのユーザがどのユーザによってブロックされているかを調べることができます。sa_locks システムプロシージャを使用して、ブロックされた接続で保持されているロックを表示できます。

これらプロパティに関する詳細については、次のような例を実行できます。

```
SELECT *, DB_NAME( DBNumber ),
        CONNECTION_PROPERTY( 'LastStatement', Number )
FROM sa_conn_info( );
```

`LockRowID` の値は、sa_locks プロシージャの出力でロックを検索するときに使用できます。

`LockIndexID` の値は、sa_locks プロシージャの出力でロックを検索するときに使用できます。また、`LockIndexID` の値は、`SYSIDX` システムビューを使用して表示できる `ISYSIDX` システムテーブルのプライマリキーに対応します。

ロックにはそれぞれ関連付けられたテーブルがあるため、`LockTable` の値を使用して、ロックで接続が待機しているかどうかを明確に判断できます。

権限

現在の接続 ID に対してこのシステムプロシージャを実行する場合、権限は必要ありません。他の接続に対してこのシステムプロシージャを実行する場合は、`SERVER OPERATOR`、`MONITOR`、または `DROP CONNECTION` のシステム権限が必要です。

関連する動作

なし。

例

次の例では、sa_conn_info システムプロシージャを使用して、サーバに対する全接続の接続プロパティをまとめた結果セットを返します。

```
CALL sa_conn_info( );
```

Number	Name	Userid	DBNumber	...
79	SQL_DBC_10dcf810	DBA	0	...
46	setup	User1	0	...
...

次の例では、sa_conn_info システムプロシージャを使用して、どの接続がテンポラリ接続を作成したかを示す結果セットを返します。

```
SELECT Number, Name, ParentConnection FROM sa_conn_info();
```

接続 8 がテンポラリ接続を作成し、そのテンポラリ接続によって CREATE DATABASE 文が実行されました。

Number	Name	ParentConnection
1000000048	INT: CreateDB	8
9	SQL_DBC_14675af8	(NULL)
8	SQL_DBA_152d5ac0	(NULL)

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_conn_list システムプロシージャ

接続 ID を含む結果セットを返します。

構文

```
sa_conn_list ([ connidparm ] [ ,dbidparm ])
```

パラメータ

- connidparm** – 接続 ID 番号を指定する任意の INTEGER パラメータです。
- dbidparm** – データベース ID 番号を指定する任意の INTEGER パラメータです。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- SERVER OPERATOR
- MONITOR
- DROP CONNECTION

備考

カラム名	データ型	説明
Number	INTEGER	接続の ID 番号。

connidparm が 0 より大きい場合、指定した接続についての情報が返されます。*connidparm* が 0 未満の場合、現在の接続についての情報が返されます。*connidparm* と *dbidparm* を指定しない場合、またはこれらのパラメータが NULL の場合、データベースサーバ上で実行されているすべてのデータベースに対するすべての接続の接続 ID が返されます。

connidparm が NULL で *dbidparm* が 0 以上の場合、そのデータベースの接続 ID のみが返されます。*connidparm* が NULL で *dbidparm* が 0 未満の場合、現在のデータベースの接続 ID のみが返されます。

例

次の例は、**sa_conn_list** システムプロシージャを使用して接続 ID のリストを表示します。

```
CALL sa_conn_list( );
```

番号
1,949
1,948
...

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_conn_properties システムプロシージャ

接続プロパティ情報をレポートします。

構文

```
sa_conn_properties( [ connidparm ] )
```

引数

- connidparm** – 接続 ID 番号を指定する任意の INTEGER パラメータ。デフォルトは NULL です。

結果セット

カラム名	データ型	説明
Number	INTEGER	現在の接続の接続 ID (数値) を返します。
PropNum	INTEGER	接続プロパティの番号が返されます。
PropName	VARCHAR(255)	接続プロパティの名前が返されます。
PropDescription	VARCHAR(255)	接続プロパティの説明が返されます。
Value	LONG VARCHAR	接続プロパティの値が返されます。

備考

接続 ID を Number として返し、使用可能な各接続プロパティについて PropNum、PropName、PropDescription、Value を返します。すべての接続プロパティ、接続に関するデータベースオプション設定、接続に関連する統計情報の値が返されます。NULL 値を持つ有効なプロパティも返されます。

connidparm が 0 未満の場合、現在の接続のプロパティ値が返されます。connidparm が指定されていないか NULL の場合、現在のデータベースに対するすべての接続のプロパティ値が返されます。

権限

現在の接続 ID に対してこのシステムプロシージャを実行する場合、権限は必要ありません。他の接続に対してこのシステムプロシージャを実行する場合は、SERVER OPERATOR、MONITOR、または DROP CONNECTION のシステム権限が必要です。

関連する動作

なし。

例

次の例では、sa_conn_properties システムプロシージャを使用して、全接続の接続プロパティ情報をまとめた結果セットを返します。

```
CALL sa_conn_properties( );
```

Number	PropNum	PropName	...
79	37	ClientStmtCacheHits	...
79	38	ClientStmtCacheMisses	...
...

次の例では、sa_conn_properties システムプロシージャを使用して、CPU 時間の降順で、すべての接続のリストを返します*。

```
SELECT Number AS connection_number,
       CONNECTION_PROPERTY ( 'Name', Number ) AS connection_name,
       CONNECTION_PROPERTY ( 'Userid', Number ) AS user_id,
       CAST ( Value AS NUMERIC ( 30, 2 ) ) AS approx_cpu_time
FROM sa_conn_properties( )
WHERE PropName = 'ApproximateCPUTime'
ORDER BY approx_cpu_time DESC;
```

* この例の提供者は RisingRoad Professional Services (<http://www.risingroad.com>) の Breck Carter です。

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_db_info システムプロシージャ

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_info( [ dbidparm ] )
```

引数

- dbidparm** – データベース ID 番号を指定する任意の INTEGER パラメータ。デフォルトは NULL です。

結果セット

カラム名	データ型	説明
Number	INTEGER	現在の接続の接続 ID (数値) を返します。
Alias	VARCHAR(255)	データベース名を返します。
File	VARCHAR(255)	パスを含むデータベースルートファイル名を返します。
ConnCount	INTEGER	データベースとの接続の数を返します。プロパティ値には、内部処理に使用されている接続は含まれませんが、イベントと外部環境サポートに使用されている接続は含まれます。
PageSize	INTEGER	データベースのページサイズ (バイト単位) を返します。
LogName	VARCHAR(255)	パスを含むトランザクションログファイル名を返します。

備考

データベース ID を指定した場合、sa_db_info は、指定したデータベースの Number、Alias、File、ConnCount、PageSize、LogName を持つ 1 つのローを返します。

dbidparm が 0 より大きい場合、指定したデータベースのプロパティが返されます。dbidparm が 0 より小さい場合、現在のデータベースのプロパティが返されます。dbidparm を指定しない場合または NULL の場合、データベースサーバで実行されているすべてのデータベースのプロパティが返されます。

権限

現在のデータベースに対してこのシステムプロシージャを実行する場合、権限は必要ありません。他のデータベースに対してこのシステムプロシージャを実行する場合は、SERVER OPERATOR または MONITOR のシステム権限が必要です。

関連する動作

なし。

例

次の文は、サーバで実行されている各データベースのローを返します。

```
CALL sa_db_info( );
```

プロパティ	値
Number	0
Alias	iqdemo
File	C:\ProgramData\SybaseIQ\demo \iqdemo.db
ConnCount	3
PageSize	4096
LogName	C:\ProgramData\SybaseIQ\demo \iqdemo.log

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_db_properties システムプロシージャ

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( [ dbidparm ] )
```

引数

- dbidparm** – データベース ID 番号を指定する任意の INTEGER パラメータ。デフォルトは NULL です。

結果セット

カラム名	データ型	説明
Number	INTEGER	データベース ID 番号。
PropNum	INTEGER	データベースプロパティの番号。
PropName	VARCHAR(255)	データベースプロパティ名。
PropDescription	VARCHAR(255)	データベースプロパティの説明。
Value	LONG VARCHAR	データベースプロパティの値。

備考

データベース ID を指定した場合、sa_db_properties システムプロシージャはデータベース ID 番号と、使用可能な各データベースプロパティの PropNum、PropName、PropDescription、Value を返します。データベースに関するすべてのデータベースプロパティと統計情報の値が返されます。NULL 値を持つ有効なプロパティも返されます。

dbidparm が 0 より大きい場合、指定したデータベースのデータベースプロパティが返されます。dbidparm が 0 より小さい場合、現在のデータベースのデータベースプロパティが返されます。dbidparm を指定しない場合または NULL の場合、データベースサーバで実行されているすべてのデータベースのプロパティが返されます。

権限

現在のデータベースに対してこのシステムプロシージャを実行する場合、権限は必要ありません。他のデータベースに対してこのシステムプロシージャを実行する場合は、SERVER OPERATOR または MONITOR のシステム権限が必要です。

関連する動作

なし。

例

次の例では、sa_db_properties システムプロシージャを使用して、呼び出し側がシステム権限 SERVER OPERATOR または MONITOR を持っている場合のすべてのデータベースのデータベースプロパティをまとめた結果セットを返します。それ以外の場合は、現在のデータベースのデータベースプロパティが返されます。

```
CALL sa_db_properties( );
```

Number	PropNum	PropName	...
0	0	ConnCount	...

Number	PropNum	PropName	...
0	1	IdleCheck	...
0	2	IdleWrite	...
...

次の例では、`sa_db_properties()` システムプロシージャを使用して、2 番目のデータベースのデータベースプロパティをまとめた結果セットを返します。

```
CALL sa_db_properties( 1 );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_describe_shapefile システムプロシージャ

ESRI シェイプファイルに含まれるカラムの名前と型を記述します。このシステム機能は、空間データ機能とともに使用します。

構文

```
sa_describe_shapefile ( shp_filename , srid [, encoding ] )
```

パラメータ

- **shp_filename** – ESRI シェイプファイルのロケーションを識別する VARCHAR(512) パラメータ。ファイル名の拡張子は `.shp` にしてください。また、同じディレクトリにある同じベース名の `.dbf` ファイルを関連付けてください。パスは、クライアントアプリケーションではなく、データベースサーバを基準にした相対パスを指定します。
- **srid** – シェイプファイル内のジオメトリの SRID を識別する INTEGER パラメータ。カラムに複数の SRID を格納できるようにするには、NULL を指定します。NULL を指定すると、ジオメトリ値に対して実行できる操作が制限されます。
- **encoding** – シェイプファイルを読み込むときに使用するエンコードを識別する任意の VARCHAR(50) パラメータ。デフォルトは NULL です。エンコードが NULL の場合は、ISO-8859-1 文字セットが使用されます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。補足：

- `-gl` データベースオプションが DBA に設定されている場合は、次のシステム権限のいずれかが必要です。
 - ALTER ANY TABLE

- ALTER ANY OBJECT
- LOAD ANY TABLE
- READ FILE
- -gl データベースオプションが ALL に設定されている場合、追加のシステム権限は必要ありません。
- -gl データベースオプションが NONE に設定されている場合は、READ FILE システム権限が必要です。

備考

カラム名	データ型	説明
column_number	INTEGER	このローが記述するカラムの順序位置 (開始値は 1)。
name	VARCHAR(128)	カラム名。
domain_name_with_size	VARCHAR(160)	サイズと精度を含むデータ型名 (CREATE TABLE 関数または CAST 関数で使用)。

sa_describe_shapefile システムプロシージャは、ESRI シェイプファイル内のカラムの名前と型を記述するときに使用します。この情報を使用して、LOAD TABLE 文または INPUT 文によってシェイプファイルからデータをロードするテーブルを作成できます。また、このシステムプロシージャは、OPENSTRING...FORMAT SHAPEFILE の WITH 句を指定することで、シェイプファイルを読み込むときにも使用できます。

例

次の例は、シェイプファイルデータを格納するためのテーブルを作成するときに使用した文字列を表示します。

```
BEGIN
    DECLARE create_cmd LONG VARCHAR;
    SELECT 'create table if not exists esri_load( record_number int
primary key, ' ||
        (SELECT list( name || ' ' || domain_name_with_size, ', '
ORDER BY column_number )
FROM sa_describe_shapefile( 'c:\¥¥esri¥¥tgr36069trt00.shp',
1000004326 )
WHERE column_number > 1 ) || ' )'
INTO create_cmd;
SELECT create_cmd;
EXECUTE IMMEDIATE create_cmd;
END
```

LOAD ANY TABLE システム権限を持っており、-gl データベースオプションが NONE に設定されていない場合は、次の文を使用してシェイプファイルデータをテーブルにロードできます。

```
LOAD TABLE esri_load
USING FILE 'c:\¥¥esri¥¥tgr36069trt00.shp'
FORMAT SHAPEFILE;
```

sa_dependent_views システムプロシージャ

指定したテーブルまたはビューのすべての従属ビューリストを返します。

構文

```
sa_dependent_views (
[ tbl_name
[, owner_name ] ]
)
```

引数

- **tbl_name** – テーブルまたはビューの名前を指定する任意の CHAR(128) パラメータ。デフォルトは NULL です。
- **owner_name** – 任意の CHAR(128) パラメータを使用して、tbl_name の所有者を指定します。デフォルトは NULL です。

結果セット

カラム名	データ型	説明
table_id	UNSIGNED INTEGER	テーブルまたはビューのオブジェクト ID。
dep_view_id	UNSIGNED INTEGER	従属ビューのオブジェクト ID。

備考

このプロシージャを使用して、テーブルとその従属ビューの ID リストを取得します。

テーブル名と所有者名について指定された条件を満たす既存のテーブルがない場合、エラーは発生しません。この他にも次の制約が適用されます。

- **owner** と **tbl_name** がどちらも NULL の場合は、従属ビューを持つすべてのテーブルに関する情報が返されます。
- **tbl_name** が NULL で **owner** が指定されている場合は、指定された所有者が所有するすべてのテーブルに関する情報が返されます。
- **tbl_name** が指定されていて **owner** が NULL の場合は、指定された名前を持ついずれかのテーブルに関する情報が返されます。

権限

なし。

関連する動作

なし。

例

次の例では、sa_dependent_views システムプロシージャを使用して、SalesOrders テーブルに依存するビューの ID リストを取得します。このプロシージャは、SalesOrders に対して table_id を、従属ビュー ViewSalesOrders に対して dep_view_id を返します。

```
CALL sa_dependent_views( 'SalesOrders' );
```

次の例では、sa_dependent_views システムプロシージャを SELECT 文で使用して、SalesOrders テーブルに依存するビューの名前リストを取得します。このプロシージャは、ViewSalesOrders ビューを返します。

```
SELECT t.table_name FROM SYSTAB t,  
sa_dependent_views( 'SalesOrders' ) v  
WHERE t.table_id = v.dep_view_id;
```

sa_disable_auditing_type システムプロシージャ

特定のイベントの監査を無効にします。

構文

```
sa_disable_auditing_type( types )
```

引数

- **types** –カンマで区切られた文字列を指定する VARCHAR(128) パラメータ。この文字列には、次の 1 つ以上の値が含まれています。
 - **all** – すべてのタイプの監査を無効にします。
 - **connect** – 成功した接続と失敗した接続の両方の監査を無効にします。
 - **connectFailed** – 失敗した接続の監査を無効にします。
 - **DDL** – DDL 文の監査を無効にします。
 - **options** – パブリックオプションの監査を無効にします。
 - **permission** – パーミッションチェック、ユーザチェック、SETUSER 文の監査を無効にします。
 - **permissionDenied** – 失敗したパーミッションチェックと失敗したユーザチェックの監査を無効にします。
 - **triggers** – トリガイイベントに応じて監査を無効にします。

備考

sa_disable_auditing_type システムプロシージャを使用すると、1 つ以上の情報カテゴリの監査を無効にできます。

システムプロシージャ

このオプションを all に設定すると、すべての監査を無効にできます。監査を無効にするには、PUBLIC.auditing オプションを Off に設定する方法もあります。

権限

SET ANY SECURITY OPTION システム権限が必要です。

関連する動作

なし。

例

すべての監査を無効にするには、次の文を実行します。

```
CALL sa_disable_auditing_type( 'all' );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_disk_free_space システムプロシージャ

トランザクションログ、トランザクションログミラー、テンポラリファイルに使用可能な領域に関する情報をレポートします。

構文

```
sa_disk_free_space( [ p_dbpace_name ] )
```

引数

- p_dbpace_name** – トランザクションログファイル、トランザクションログミラーファイル、またはテンポラリファイルの名前を指定する VARCHAR(128) パラメータ。デフォルトは NULL です。

SYSTEM を指定すると、メインデータベースファイルに関する情報を取得できます。TEMPORARY または TEMP を指定するとテンポラリファイル、TRANSLOG を指定するとトランザクションログ、TRANSLOGMIRROR を指定するとトランザクションログミラーに関する情報を、それぞれ取得できます。

結果セット

カラム名	データ型	説明
dbspace_name	VARCHAR(128)	これは、トランザクションログファイル、トランザクションログミラーファイル、またはテンポラリファイルです。
free_space	UNSIGNED BIGINT	ボリューム上の空きバイト数。

カラム名	データ型	説明
total_space	UNSIGNED BIGINT	ドライブで使用可能な合計ディスク領域。

備考

p_dbSPACE_name パラメータが指定されていないか NULL の場合、存在する場合はトランザクションログ、トランザクションログミラー、テンポラリファイルごとに1つのローが結果セットに含まれます。*p_dbSPACE_name* が指定されている場合は、1つまたは0個のローが返ります (log または mirror が指定されていてログファイルまたはミラーファイルが存在しない場合は0です)。

権限

MANAGE ANY DBSPACE システム権限が必要です。

関連する動作

なし。

例

次の例では、sa_disk_free_space システムプロシージャを使用して、空き領域についての情報を格納した結果セットを返します。

```
CALL sa_disk_free_space( );
```

dbSPACE_name	free_space	total_space
system	10952101888	21410402304
translog	10952101888	21410402304
temporary	10952101888	21410402304

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_enable_auditing_type システムプロシージャ

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( types )
```

引数

- types* – カンマで区切られた文字列を指定する VARCHAR(128) パラメータ。この文字列には、次の1つ以上の値が含まれています。

- **all** – すべてのタイプの監査を有効にします。
- **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。
- **connectFailed** – 失敗した接続の監査を有効にします。
- **DDL** – DDL 文の監査を有効にします。
- **options** – パブリックオプションの監査を有効にします。
- **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。
- **permissionDenied** – 失敗したパーミッションチェックと失敗したユーザチェックの監査を有効にします。
- **triggers** – トリガイベント後の監査を有効にします。

備考

sa_enable_auditing_type は、PUBLIC.auditing オプションと一緒に使用して、特定のタイプの情報の監査を有効にします。

PUBLIC.auditing オプションを On に設定して、監査対象の情報タイプを指定しない場合は、デフォルト設定 (all) が有効になります。この場合、すべてのタイプの監査情報が記録されます。

PUBLIC.auditing オプションを On に設定して、sa_disable_auditing_type を使用してすべてのタイプ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 を使用して監査対象にする情報のタイプを指定します。

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。

権限

SET ANY SECURITY OPTION システム権限が必要です。

関連する動作

なし。

例

オプションの監査のみを有効にするには、次のようにします。

```
CALL sa_enable_auditing_type( 'options' );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_eng_properties システムプロシージャ

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 ( )
```

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名。
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。

備考

使用可能な各サーバプロパティの PropNum、PropName、PropDescription、Value を返します。データベースサーバに関するすべてのデータベースサーバプロパティと統計情報の値が返されます。

権限

なし。

関連する動作

なし。

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	...
...

sa_external_library_unload システムプロシージャ

外部ライブラリをアンロードします。

構文

```
sa_external_library_unload ( [ 'external-library' ] )
```

パラメータ

- **external-library** – 必要に応じて、この LONG VARCHAR パラメータを使用して、アンロードするライブラリの名前を指定します。ライブラリを指定しない場合は、使用中でない外部ライブラリがすべてアンロードされます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、次のものも必要です。MANAGE ANY EXTERNAL OBJECT システム権限

備考

指定した外部ライブラリが使用中であるか、ロードされていない場合は、エラーが返されます。パラメータを指定しない場合で、ロードされている外部ライブラリが見つからないときは、エラーが返されます。

例

myextlib.dll という名前の外部ライブラリをアンロードします。

```
CALL sa_external_library_unload( 'myextlib.dll' );
```

現在使用されていないすべてのライブラリをアンロードします。

```
CALL sa_external_library_unload();
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_flush_cache システムプロシージャ

データベースサーバキャッシュ内の現在のデータベースに対するすべてのページを空にします。

構文

```
sa_flush_cache ( )
```

備考

データベース管理者は、このプロシージャを使用して現在のデータベースのデータベースサーバキャッシュの内容を空にします。パフォーマンスの計測に使用し、同じ結果が繰り返し得られるようにします。

権限

SERVER OPERATOR システム権限が必要です。

関連する動作

なし。

例

次の例では、データベースサーバキャッシュ内の現在のデータベースに対するすべてのページを空にします。

```
CALL sa_flush_cache( );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_get_user_status システムプロシージャ

ユーザの現在のステータスを特定できます。

構文

```
sa_get_user_status( )
```

結果セット

カラム名	データ型	説明
user_id	UNSIGNED INTEGER	ユーザを識別するユニークな番号。
user_name	CHAR(128)	ユーザの名前。
connections	INTEGER	このユーザによる現在の接続数。
failed_logins	UNSIGNED INTEGER	ユーザがログインしようとして失敗した回数。
last_login_time	TIMESTAMP	ユーザが最後にログインした現地時刻。
locked	TINYINT	ユーザアカウントがロックされているかどうかを示すインジケータ。
reason_locked	LONG VARCHAR	アカウントがロックされた理由。
user_dn	CHAR(1024)	LDAP サーバに接続しているユーザ ID の識別名 (DN)。

カラム名	データ型	説明
user_dn_cached_at	TIMESTAMP	DN が保存されたローカル時刻。
password_change_state	BIT	二重パスワードの変更が進行中であるかどうかを示す値 (0 = いいえ、1 = はい) デフォルトは 0 です。
password_change_first_user	UNSIGNED INTEGER	二重パスワードの最初の部分を設定したユーザの user_id で、それ以外の場合は NULL。
password_change_second_user	UNSIGNED INTEGER	二重パスワードの 2 番目の部分を設定したユーザの user_id で、それ以外の場合は NULL。
user_dn	CHAR(1024)	ユーザの識別名を設定します。
user_dn_cached_at	TIMESTAMP	識別名が見つかった日付と時刻。

備考

このプロシージャは、ユーザの現在のステータスを示す結果セットを返します。基本的なユーザ情報に加えて、ユーザがロックアウトされているかどうかを示すカラムや、ロックアウトの理由が格納されたカラムが含まれています。ポリシーによるロック、パスワードの有効期限超過、失敗試行回数の超過を理由として、ユーザをロックアウトできます。

LDAP ユーザ認証を使用してユーザを認証した場合、出力にユーザの識別名と、識別名が見つかった日付と時刻が含まれます。

権限

自分自身についての情報を表示できます。権限は必要ありません。他のユーザの情報を表示する場合は、MANAGE ANY USER システム権限が必要です。

関連する動作

なし。

例

次の例は、sa_get_user_status システムプロシージャを使用して、データベースユーザのステータスを返します。

```
CALL sa_get_user_status;
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_get_ldapsrv_status システムプロシージャ

LDAP サーバ設定オブジェクトの現在のステータスを確認します。

構文

sa_get_ldapsrv_status()

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

カラム名	データ型	説明
ldsrv_id	UNSIGNED BIGINT	LDAP サーバ設定オブジェクトのユニークな識別子。プライマリーキーであり、ログインポリシーから LDAP サーバを参照する際に使用する。
ldsrv_name	CHAR(128)	LDAP サーバ設定オブジェクトに割り当てられている名前。
ldsrv_state	CHAR(9)	LDAP サーバの読み込み専用状態。 1 - RESET 2 - READY 3 - ACTIVE 4 - FAILED 5 - SUSPENDED 数値がシステムテーブルに格納され、対応するテキスト値がシステムビューに表示される。
ldsrv_last_state_change	TIMESTAMP	最終状態変更発生時刻を示す。LDAP サーバのローカルのタイムゾーンに関係なく、この値は協定世界時 (UTC) で格納される。

SYSLDAPSERVER のカラムの値は、チェックポイントが発生してメモリの内容がディスク上のカタログに書き込まれる前に確認します。カタログのカラム `ldsrv_state` と `ldsrv_last_state_change` の更新は、LDAP ディレクトリサーバの障害による接続の切断など、LDAP サーバオブジェクトの状態が変化するイベントが発生したために実行される LDAP サーバオブジェクトに対するチェックポイントの実行中に非同期に実行されます。LDAP サーバオブジェクトの状態には、LDAP ディレクトリサーバの状態が反映されます。

sa_list_external_library システムプロシージャ

現在サーバにロードされている外部ライブラリをリストします。

構文

```
sa_list_external_library ( )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、次のものも必要です。MANAGE ANY EXTERNAL OBJECT システム権限

備考

エンジンにロードされている外部ライブラリのリストを、参照カウントとともに返します。

参照カウントは、エンジン内のライブラリのインスタンス数です。

sa_external_library_unload プロシージャを実行して外部ライブラリをアンロードできるのは、その参照カウントが0の場合のみです。

例

外部ライブラリとその参照カウントを一覧表示します。

```
CALL sa_list_external_library()
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_locks システムプロシージャ

データベース内のすべてのロックを表示します。

構文

```
sa_locks (  
    [ connection  
    [, creator  
    [, table_name  
    [, max_locks ] ] ] )
```

パラメータ

- connection** – この INTEGER パラメータを使用して、接続 ID 番号を指定します。プロシージャは、指定された接続に関するロック情報のみを返します。デフォルト値は 0 (NULL) で、この場合は全接続の情報が返されます。
- creator** – この CHAR(128) パラメータを使用して、ユーザ ID を指定します。プロシージャは、指定したユーザが所有するテーブルに関する情報のみを返しま

す。creator パラメータのデフォルト値は NULL です。このパラメータが NULL に設定されている場合、sa_locks は次の情報を返します。

- table_name パラメータが指定されていない場合、データベースのすべてのテーブルに関するロック情報が返されます。
- table_name パラメータを指定すると、現在のユーザが作成した、指定した名前を持つテーブルのロック情報が返されます。
- **table_name** – この CHAR(128) パラメータを使用して、テーブル名を指定します。プロシージャは、指定したテーブルに関する情報のみを返します。デフォルト値は NULL で、この場合はすべてのテーブルの情報が返されます。
- **max_locks** – この INTEGER パラメータを使用して、情報が返されるロックの最大数を指定します。デフォルト値は 1000 です。値が -1 の場合は、すべてのロック情報が返されます。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR システム権限。

備考

カラム名	データ型	説明
conn_name	VARCHAR(128)	現在の接続の名前。
conn_id	INTEGER	接続 ID 番号。
user_id	CHAR(128)	接続のユーザ ID。
table_type	CHAR(6)	テーブルのタイプ。テーブルの場合は BASE、グローバルテンポラリテーブルの場合は GLBTMP、マテリアライズドビューの場合は MVIEW。
creator	VARCHAR(128)	テーブルの所有者。
table_name	VARCHAR(128)	ロックが保持されているテーブル。
index_id	INTEGER	インデックス ID または NULL。
lock_class	CHAR(8)	ロッククラス。Schema、Row、Table、Position のいずれか。
lock_duration	CHAR(11)	ロックの継続期間。Transaction、Position、Connection のいずれか。
lock_type	CHAR(9)	ロックタイプ (ロッククラスに依存する)。

システムプロシージャ

カラム名	データ型	説明
row_identifier	UNSIGNED BIGINT	ローの識別子。8 バイトのロー識別子または NULL。

sa_locks プロシージャは、データベース内の全ロックの情報で構成される結果セットを返します。lock_type カラムの値は、lock_class カラムのロック分類によって変わります。次の値が返されます。

ロッククラス	ロックタイプ	コメント
Schema	<ul style="list-style-type: none"> • Shared – 共有スキーマロック。 • Exclusive – (IQ カタログストア テーブルのみ) 排他スキーマロック。 	スキーマロックの場合、row_identifier とインデックス ID の値は NULL。

ロック クラス	ロックタイプ	コメント
Row	<ul style="list-style-type: none"> • Read – 読み込みロック。 • Intent – 意図的ロック。 • ReadPK – 読み込みロック。 • Write – 書き込みロック。 • WriteNoPK – 書き込みロック。 • Surrogate – 代理ロック。 	<p>ローの読み取りロックは、短期のロック (独立性レベル 1 でスキャン) にするか、高い独立性レベルの長期ロックにすることができる。lock_duration カラムは、読み取りロックがカーソル安定性 (Position) のために短期になるか、COMMIT/ROLLBACK (Transaction) まで保持される長期になるかを指定する。ローのロックは、常に特定のローに保持される。この 8 バイトのロー識別子は、row_identifier カラムの 64 ビット整数値としてレポートされる。</p> <p>代理ロックは、ローロックの特殊なケースである。代理ロックは代理エントリで保持される。代理エントリは、参照整合性のチェックが遅延するときに作成される。テーブルで作成されるすべての代理エントリについてユニークな代理ロックはない。ただし、代理ロックは、指定した接続で指定されるテーブルで作成される代理エントリのセットに対応する。row_identifier 値は、代理ロックに関連付けられているテーブルと接続についてユニークである。</p> <p>必要に応じて、ローのキーとキーではない部分を別々にロックできる。1 つの接続で共有 (読み取り) アクセス用にローのキー部分に対する読み取りロックを取得して、他の接続でローのキーではない他のカラムに対する書き込みロックを取得できるようにすることができる。ローのキーではないカラムを更新しても、そのローを参照する外部ローの挿入と削除は妨げられない。</p>

ロッククラス	ロックタイプ	コメント
Table	<ul style="list-style-type: none"> • Shared – 共有テーブルロック。 • Intent – 意図的にテーブルロックを更新。 • Exclusive – (IQ カタログストアテーブルのみ) 排他テーブルロック。 	なし
Position	<ul style="list-style-type: none"> • Phantom – (IQ カタログストアテーブルのみ) 幻ロック。 • Insert – 挿入ロック。 	通常、位置ロックも特定のローに保持され、そのローの 64 ビットロー識別子は結果セットの row_identifier カラムに表示される。ただし、row_identifier カラムが NULL の場合、位置ロックはスキャン (インデックススキャンまたは逐次スキャン) 全体に保持される。

位置ロックは、逐次テーブルスキャンまたはインデックススキャンと関連付けられます。index_id カラムは、位置ロックが逐次スキャンに関連付けられているかどうかを示します。逐次スキャンのために位置ロックが保持される場合、index_id カラムは NULL です。特定のインデックススキャンの結果として位置ロックが保持される場合、そのインデックスのインデックス識別子は index_id カラムにリストされます。インデックス識別子は、SYSIDX ビューを使用して表示できる ISYSIDX システムテーブルのプライマリキーに対応します。位置ロックがすべてのインデックスのスキャンで保持される場合、インデックス ID 値は -1 です。

例

次のクエリを実行して、ロックを識別できます。

```
CALL sa_locks( );
```

sa_make_object システムプロシージャ

ALTER 文を実行する前に、オブジェクトのスケルトンインスタンスが存在することを確認します。

構文

```
sa_make_object(
objtype
, objname
[, owner
```

```
[, tablename ] ]
)
```

```
objtype:
'procedure'
| 'function'
| 'view'
| 'trigger'
| 'service'
| 'event'
```

引数

- **objtype** – 作成されるオブジェクトのタイプを指定する CHAR(30) パラメータ。objtype が 'trigger' の場合、この引数はトリガが作成されるテーブルの所有者を示します。
- **objname** – 作成されるオブジェクトの名前を指定する CHAR(128) パラメータ。
- **owner** – 作成されるオブジェクトの所有者を指定する任意の CHAR(128) パラメータ。デフォルトは NULL です。
- **tablename** – この CHAR(128) パラメータは、objtype が 'trigger' である場合にのみ必要です。トリガを作成するテーブルの名前を指定するときに使用します。デフォルトは NULL です。

備考

このプロシージャは、データベーススキーマの作成または変更のために繰り返し実行されるスクリプトで使用すると便利です。このようなスクリプトに共通する問題は、最初の実行では CREATE 文を実行する必要がありますが、その後は 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使用すると、オブジェクトが存在するかどうかを確認するためにシステムビューを問い合わせる必要がなくなります。

プロシージャ、ファンクション、ビュー、トリガについては、このシステムプロシージャの代わりに OR REPLACE 句を使用できるようになりました。

使用するには、このプロシージャの後にオブジェクト定義全体を含む ALTER 文を実行します。

権限

次の権限が必要です。

- **呼び出し側が所有するプロシージャまたは関数** – CREATE PROCEDURE システム権限、CREATE ANY PROCEDURE システム権限、または CREATE ANY OBJECT システム権限
- **他のユーザが所有するプロシージャまたは関数** – CREATE ANY PROCEDURE システム権限または CREATE ANY OBJECT システム権限

- サービス – MANAGE ANY WEB SERVICE システム権限
- イベント – MANAGE ANY EVENT システム権限または CREATE ANY OBJECT システム権限
- 呼び出し側が所有するビュー – CREATE VIEW システム権限、CREATE ANY VIEW システム権限、または CREATE ANY OBJECT システム権限
- 他のユーザが所有するビュー – CREATE ANY VIEW システム権限または CREATE ANY OBJECT システム権限
- トリガ – 自分が所有するテーブルにトリガがある場合は、CREATE ANY TRIGGER システム権限または CREATE ANY OBJECT システム権限のいずれかが必要です。

他のユーザが所有するテーブルにトリガがある場合は、CREATE ANY TRIGGER システム権限または CREATE ANY OBJECT システム権限のいずれかが必要です。さらに、次のいずれかのシステム権限が必要です。

- ALTER ANY TABLE 権限
- ALTER ANY OBJECT システム権限
- トリガが作成されたテーブルの ALTER 権限。

関連する動作

オートコミット。

例

次の文は、スケルトンプロシージャ定義が作成されていることを確認し、プロシージャを定義し、プロシージャに対する権限を付与します。これらの命令が記述されたスクリプトファイルは、データベースに対して繰り返し実行でき、エラーは起こりません。

```
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' );
```

sa_nchar_terms システムプロシージャ

NCHAR 文字列を単語に分割し、各単語をローとして、その位置とともに返します。

構文

```
sa_nchar_terms ( 'char-string' [ , 'text-config-name' [ , 'owner' ] ] )
```

パラメータ

- **char-string** – 解析する NCHAR 文字列。
- **text-config-name** – 文字列の処理時に適用するテキスト設定オブジェクト。デフォルト値は 'default_nchar' です。
- **owner** – 指定されたテキスト設定オブジェクトの所有者。デフォルト値は DBA です。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

sa_nchar_terms を使用して、テキスト設定オブジェクトの設定が適用されるときに、文字列がどのように解釈されるのかを確認できます。インデックス処理で、またはクエリ文字列から、どの単語が削除されるかを確認する場合に便利です。

sa_nchar_terms の構文は、**sa_char_terms** システムプロシージャの構文と似ています。

注意： NCHAR データ型は、**IN SYSTEM** テーブルについてのみサポートされています。

sa_performance_diagnostics システムプロシージャ

データベースサーバが要求のタイミングの記録を有効にしたとき、すべての接続について要求のタイミング情報の概要を返します。

構文

```
sa_performance_diagnostics ( )
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものが 필요합니다。MONITOR システム権限。

備考

カラム名	データ型	説明
Number	INTEGER	現在の接続の接続 ID (数値) を返す。
Name	VARCHAR(255)	<p>現在の接続の名前を返す。</p> <p>ConnectionName (CON) 接続パラメータを使用して接続名を指定できる。</p> <p>データベースサーバによって作成されるテンポラリ接続には次の名前が使用される。</p> <ul style="list-style-type: none"> • INT:ApplyRecovery • INT:BackupDB • INT:Checkpoint • INT:Cleaner • INT:CloseDB • INT>CreateDB • INT>CreateMirror • INT:DelayedCommit • INT:DiagRcvr • INT:DropDB • INT:EncryptDB • INT:Exchange • INT:FlushMirrorLog • INT:FlushStats • INT:HTTPReq • INT:PromoteMirror • INT:PurgeSnapshot • INT:ReconnectMirror • INT:RecoverMirror • INT:RedoCheckpoint • INT:RefreshIndex • INT:ReloadTrigger • INT:RenameMirror • INT:RestoreDB • INT:StartDB • INT:VSS
Userid	VARCHAR(255)	接続のユーザ ID を返す。

カラム名	データ型	説明
DBNumber	INTEGER	データベースの ID 番号を返す。
LoginTime	TIMESTAMP	接続が確立された日付と時刻を返す。
TransactionStart-Time	TIMESTAMP	COMMIT または ROLLBACK の後にデータベースが最初に変更された時刻を含む文字列を返す。最後の COMMIT または ROLLBACK 以降にデータベースが変更されていない場合は、空の文字列を返す。
LastReqTime	TIMESTAMP	指定された接続において最後の要求が開始された時刻を返す。このプロパティは、イベントなどの内部接続の場合は空の文字列を返すことがある。
ReqType	VARCHAR(255)	最後の要求のタイプを返す。接続が接続プールによってキャッシュされている場合、その ReqType 値は CONNECT_POOL_CACHE になる。
ReqStatus	VARCHAR(255)	<p>要求のステータスを返す。値は次のいずれか。</p> <ul style="list-style-type: none"> • Idle – この接続では現在、要求を処理していない。 • Unscheduled – この接続では他の処理が実行されており、使用可能なデータベースサーバワーカを待っている。 • BlockedIO – この接続はブロックされ、I/O 処理の完了を待っている。 • BlockedContention – この接続はブロックされ、共有データベースサーバデータ構造体へのアクセスを待っている。 • BlockedLock – この接続はブロックされ、オブジェクトのロックの解放を待っている。 • Executing – この接続では現在、要求を実行している。 <p>アスタリスク (*) が付いている値が返されるのは、-zt サーバオプションによってデータベースサーバで要求タイミング情報のロギングがオンになっている場合に限られる。要求タイミング情報のロギングが実行されていない場合は (デフォルト)、値は Executing として返される。</p>

カラム名	データ型	説明
ReqTimeUnscheduled	DOUBLE	未スケジュール時間を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqTimeActive	DOUBLE	要求の処理にかかった時間 (秒単位) を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqTimeBlockIO	DOUBLE	I/O 処理の完了を待っていた時間 (秒単位) を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqTimeBlockLock	DOUBLE	ロックを待っていた時間 (秒単位) を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqTimeBlockContention	DOUBLE	アトミックアクセスを待っていた時間 (秒単位) を返す。RequestTiming サーバプロパティが Off に設定されている場合は、NULL を返す。
ReqCountUnscheduled	INTEGER	接続がスケジューリングを待った回数を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqCountActive	INTEGER	処理が完了した要求の数を返す。RequestTiming サーバプロパティが Off に設定されている場合は、NULL を返す。
ReqCountBlockIO	INTEGER	接続が I/O 処理の完了を待った回数を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqCountBlockLock	INTEGER	接続がロックを待った回数を返す。-zt オプションが指定されていない場合は、NULL を返す。
ReqCountBlockContention	INTEGER	接続がアトミックアクセスを待った回数を返す。-zt オプションが指定されていない場合は、NULL を返す。
LastIdle	INTEGER	要求間のチック数を返す。
BlockedOn	INTEGER	現在の接続がブロックされていない場合は 0 を返し、ブロックされている場合はロック競合によってブロックされた接続の接続番号を返す。

カラム名	データ型	説明
UncommitOp	INTEGER	コミットされていない操作の数を返す。
CurrentProcedure	VARCHAR(255)	接続で現在実行されているプロシージャの名前を返す。接続でネストされたプロシージャコールが実行されている場合は、現在のプロシージャの名前を返す。実行されているプロシージャがない場合は、空の文字列を返す。
EventName	VARCHAR(255)	接続でイベントハンドラが実行されている場合、関連付けられているイベントの名前を返す。それ以外の場合は、空の文字列を返す。
CurrentLineNumber	INTEGER	接続で実行されているプロシージャまたは複合文の現在の行番号を返す。実行されているプロシージャの名前は CurrentProcedure プロパティで取得できる。現在の行がクライアント側からの複合文の一部である場合は、空の文字列を返す。
LastStatement	LONG VARCHAR	<p>現在の接続で最後に準備された SQL 文を返す。</p> <p>LastStatement の値は、文が準備されると同時に設定され、文が削除されると同時にクリアされる。各接続につき 1 つの文の文字列のみが記憶される。</p> <p>ある接続について sa_conn_activity が空でない値を返した場合、その接続で現在実行されている文である可能性が高くなる。その文が完了している場合は、文がすでに削除され、このプロパティの値がクリアされている可能性がある。アプリケーションが複数の文を準備し、それらの文のステートメントハンドルを保持している場合、LastStatement の値は接続で現在実行されている処理を表さない。</p> <p>クライアントでの文のキャッシュが有効で、キャッシュされた文が再使用される場合、このプロパティは空の文字列を返す。</p>
LastPlanText	LONG VARCHAR	接続で最後に実行されたクエリの長いテキストプランを返す。最後のプランを記憶するかどうかは、sa_server_option システムプロシージャの RememberLastPlan オプションを設定するか、-zp サーバオプションを使用して制御する。

カラム名	データ型	説明
AppInfo	LONG VARCHAR	接続を確立したクライアントに関する情報を返す。HTTP 接続では、ブラウザの情報が含まれる。jConnect または Sybase Open Client の古いバージョンを使用した接続については、情報が不完全な場合がある。 API 値は、DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD、DBEXPRESS のいずれか。
LockCount	INTEGER	接続で保持されているロックの数を返す。
SnapshotCount	INTEGER	接続に関連付けられているスナップショットの数を返す。

sa_performance_diagnostics システムプロシージャは、要求のタイミングプロパティで構成される結果セットと統計情報 (統計情報を収集するようにサーバが指示されている場合) を返します。要求のタイミング情報の記録は、**sa_performance_diagnostics** の呼び出し前にデータベースサーバでオンにする必要があります。オンにするには、データベースサーバの起動時に `-zt` オプションを指定するか、次の文を実行します。

```
CALL sa_server_option( 'RequestTiming', 'ON' );
```

例

次のクエリを実行すると、データベースサーバ要求が完了するまでの待ち時間が長い接続を特定することができます。

```
SELECT Number, Name,
       CAST( DATEDIFF( second, LoginTime, CURRENT_TIMESTAMP ) AS
DOUBLE ) AS T,
       IF T <> 0 THEN (ReqTimeActive / T) ELSE NULL ENDIF AS
PercentActive
FROM sa_performance_diagnostics()
WHERE T > 0 AND PercentActive > 10.0
ORDER BY PercentActive DESC;
```

現在実行中で、60 秒よりも長く実行されているすべての要求を検索します。

```
SELECT Number, Name,
       CAST( DATEDIFF( second, LastReqTime, CURRENT_TIMESTAMP ) AS
DOUBLE ) AS ReqTime
FROM sa_performance_diagnostics()
WHERE ReqStatus <> 'IDLE' AND ReqTime > 60.0
ORDER BY ReqTime DESC;
```

sa_procedure_profile_summary システムプロシージャ

データベースで実行されたすべてのプロシージャ、関数、イベント、またはトリガの実行時間に関する概要情報をレポートします。

構文

```
sa_procedure_profile_summary (
    [ filename
    [, save_to_file ] ] )
```

パラメータ

- **filename** – プロファイル情報を保存するファイル、またはロードするファイルを指定する任意の LONG VARCHAR パラメータ。デフォルトは NULL です。プロファイル情報の保存とロードの詳細については、以下の備考部分を参照してください。
- **save-to-file** – 概要情報をファイルに保存するか、前に保存したファイルからロードするかを指定する任意の INTEGER パラメータ。デフォルトは 0 です。

結果セット

カラム名	データ型	説明
object_type	CHAR(1)	オブジェクトのタイプ。使用可能なオブジェクトタイプのリストについては、下記の「備考」を参照。
object_name	CHAR(128)	ストアドプロシージャ、ファンクション、イベント、またはトリガの名前。
owner_name	CHAR(128)	オブジェクトの所有者。
table_name	CHAR(128)	トリガに対応するテーブル (他のオブジェクトタイプの場合、値は NULL)。
executions	UNSIGNED INTEGER	各プロシージャの実行回数。
millisecs	UNSIGNED INTEGER	プロシージャの実行時間 (ミリ秒単位)。
foreign_owner	CHAR(128)	システムトリガの外部テーブルを所有するデータベースユーザ。
foreign_table	CHAR(128)	システムトリガの外部テーブルの名前。

備考

このプロシージャは、次の用途で使用できます。

- **現在の概要情報を返す** – この場合、引数指定することなく、プロシージャを呼び出すだけです。
- **現在の概要情報をファイルに保存** – この場合、*filename* 引数を含め、*save_to_file* 引数に 1 を指定します。
- **保存された概要情報をファイルからロード** – この場合、*filename* 引数を含め、*save_to_file* 引数に 0 を指定します。この方法でプロシージャを使用できるのは、ロードするファイルが、プロシージャを実行するデータベースと同じデータベースで作成されたものである場合です。それ以外の場合、結果は使用できない可能性があります。

このプロシージャは、ストアドプロシージャ、ファンクション、イベント、トリガの使用頻度や効率に関する情報を返すため、この情報を使用して、時間がかかるプロシージャを微調整してデータベースのパフォーマンスを改善できます。

プロファイリングを有効にしてから、データベースのプロファイルを作成します。

結果セットの *object_type* カラムは、次のようになります。

- **P** – ストアドプロシージャ
- **F** – ファンクション
- **E** – イベント
- **T** – トリガ
- **C** – ON UPDATE システムトリガ
- **D** – ON DELETE システムトリガ

各実行について、概要情報ではなく、行ごとの詳細が必要な場合、**sa_procedure_profile** プロシージャを使用します。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。MONITOR または MANAGE PROFILING システム権限。

次の権限も必要です。

- SELECT ANY TABLE (*filename* が NULL でなく、*save_to_file* が 1 の場合)
- LOAD ANY TABLE (*filename* が NULL でなく、*save_to_file* が 0 の場合)

例

次の文は、データベースで実行されたすべてのプロシージャ、関数、イベント、またはトリガについて実行時間を返します。

```
CALL sa_procedure_profile_summary( );
```

次の文は、前の例と同じ概要情報を返します。また、summaryinfo.txt というファイルに保存します。

```
CALL sa_procedure_profile_summary( 'summaryinfo.txt', 1 );
```

次の文はいずれも、summaryinfo.txt というファイルに保存されている概要情報をロードするときに使用できます。

```
CALL sa_procedure_profile_summary( 'summaryinfo'.txt, 0 );
CALL sa_procedure_profile_summary( 'summaryinfo.txt' );
```

sa_procedure_profile システムプロシージャ

データベースで実行されたプロシージャ、関数、イベント、またはトリガ内の各行について、実行時間に関する情報をレポートします。

構文

```
sa_procedure_profile (
    [ filename
    [, save_to_file ] ] )
```

パラメータ

- **filename** – プロファイル情報を保存するファイル、またはロードするファイルを指定する任意の LONG VARCHAR パラメータ。デフォルトは NULL です。プロファイル情報の保存とロードの詳細については、以下の備考部分を参照してください。
- **save_to_file** – プロファイル情報をファイルに保存するか、前に保存したファイルからロードするかを指定する任意の INTEGER パラメータ。デフォルトは 0 です。

結果セット

カラム名	データ型	説明
object_type	CHAR(1)	オブジェクトのタイプ。使用可能なオブジェクトタイプのリストについては、下記の「備考」を参照。
object_name	CHAR(128)	ストアドプロシージャ、ファンクション、イベント、またはトリガの名前。object_type が C または D の場合、これはシステムトリガが定義された外部キーの名前です。
owner_name	CHAR(128)	オブジェクトの所有者。
table_name	CHAR(128)	トリガに対応するテーブル (他のオブジェクトタイプの場合、値は NULL)。
line_num	UNSIGNED INTEGER	プロシージャ内の行番号。
executions	UNSIGNED INTEGER	行の実行回数。

カラム名	データ型	説明
millisecs	UNSIGNED INTEGER	行の実行時間 (ミリ秒単位)。
percentage	DOUBLE	全体的な実行時間に対する特定の行の実行時間の割合。
foreign_owner	CHAR(128)	システムトリガの外部テーブルを所有するデータベースユーザ。
foreign_table	CHAR(128)	システムトリガの外部テーブルの名前。

備考

このプロシージャは、次の用途で使用できます。

- **詳細なプロシージャのプロファイル情報を返す** – この場合、引数指定することなく、プロシージャを呼び出すだけです。
- **詳細なプロシージャのプロファイル情報をファイルに保存** – この場合、*filename* 引数を含め、*save_to_file* 引数に 1 を指定します。
- **詳細なプロシージャのプロファイル情報を前に保存したファイルからロード** – この場合、*filename* 引数を含め、*save_to_file* 引数に 0 を指定します。この方法でプロシージャを使用できるのは、ロードするファイルが、プロシージャを実行するデータベースと同じデータベースで作成されたものである場合です。それ以外の場合、結果は使用できない可能性があります。

結果セットには、プロシージャ、トリガ、関数、イベント内の個々の行に関する実行回数の情報と、行が使用するプロシージャの合計の実行回数の割合が含まれるため、このプロファイル情報を使用して、パフォーマンスを低下させる可能性がある遅いプロシージャを微調整できます。

プロファイリングを有効にしてから、データベースのプロファイルを作成します。

結果セットの *object_type* カラムは、次のようになります。

- **P** – ストアドプロシージャ
- **F** – 関数
- **E** – イベント
- **T** – トリガ
- **C** – ON UPDATE システムトリガ
- **D** – ON DELETE システムトリガ

各実行について、行ごとの詳細ではなく、概要情報が必要な場合、**sa_procedure_profile_summary procedure** プロシージャを使用します。

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。
MONITOR または MANAGE PROFILING システム権限。

次の権限も必要です。

- SELECT ANY TABLE (*filename* が NULL でなく、*save_to_file* が 1 の場合)
- LOAD ANY TABLE (*filename* が NULL でなく、*save_to_file* が 0 の場合)

例

次の文は、データベースで実行されたすべてのプロシージャ、関数、イベント、またはトリガの各行について実行時間を返します。

```
CALL sa_procedure_profile( );
```

次の文は、上記の例と同じ詳細なプロシージャのプロファイル情報を返します。
また、`detailedinfo.txt` というファイルに保存します。

```
CALL sa_procedure_profile( 'detailedinfo.txt', 1 );
```

次の文はいずれも、`detailedinfo.txt` というファイルから詳細なプロシージャのプロファイル情報をロードするときに使用できます。

```
CALL sa_procedure_profile( 'detailedinfo.txt', 0 );  
CALL sa_procedure_profile( 'detailedinfo.txt' );
```

sa_report_deadlocks システムプロシージャ

データベースサーバによって作成された内部バッファから、デッドロックに関する情報を取り出します。

構文

```
sa_report_deadlocks( )
```

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。
MONITOR システム権限。

備考

カラム名	データ型	説明
snapshotId	BIGINT	デッドロックインスタンス (特定のデッドロックに関するすべてのローが同じ ID を持つ)。
snapshotAt	TIMESTAMP	デッドロックが発生した時刻。
waiter	INT	待機している接続の接続ハンドル。

カラム名	データ型	説明
who	VARCHAR(128)	待機している接続に関連付けられているユーザ ID。
what	LONG VARCHAR	待機している接続によって実行されているコマンド。 この情報は、データベースサーバのコマンドラインの <code>-zl</code> オプションを指定して最後に作成された SQL 文の取得をオンにした場合にのみ使用できる。
object_id	UNSIGNED BIGINT	ローを含むテーブルのオブジェクト ID。
record_id	BIGINT	システムテーブルのロー ID。
owner	INT	待機しているロックを所有している接続の接続ハンドル。
is_victim	BIT	ロールバックされたトランザクションを識別。
rollback_operation_count	UNSIGNED INT	トランザクションをロールバックした場合に失われる可能性のある、コミットされていないオペレーションの数。
iq_rid	UNSIGNED BIGINT	IQ RLV 対応テーブルのロー ID。
iq_txn_id	UNSIGNED BIGINT	関連付けられているローのトランザクション ID。

log_deadlocks オプションが On に設定されている場合、データベースサーバは、デッドロックに関する情報を内部バッファに記録します。sa_report_deadlocks システムプロシージャを使用して、ログ内の情報を表示できます。

関連する動作
なし。

sa_rowgenerator システムプロシージャ

指定された開始値と終了値の間のローを格納した結果セットを返します。

構文

```
sa_rowgenerator (
  [ , rstart
  [ , rend
  [ , rstep ] ] ]
)
```

引数

- **rstart** – 開始値を指定する任意の INTEGER パラメータ。デフォルト値は 0 です。
- **rend** – **rstart** 以上の終了値を指定する任意の INTEGER パラメータ。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任意の INTEGER パラメータ。デフォルト値は 1 です。

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 句で使用して、番号のシーケンスを生成できます。RowGenerator システムテーブルを使用する代わりに、このプロシージャを使用できます。sa_rowgenerator は次のようなタスクに使用できます。

- 結果セット内の既知の数のローについてテストデータを生成する。
- あらゆる範囲の値に対するローを格納した結果セットを生成する。たとえば、1 か月の毎日についてのローを生成したり、郵便番号の範囲を生成したりできます。
- 結果セット内に指定した数のローを格納するクエリを生成する。これは、クエリのパフォーマンスのテストに役立ちます。

正しい開始値、終了値、正の 0 以外の増分値を指定しないと、ローは返されません。

次の文を使用して、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( 1, 255 );
```

権限

なし。

関連する動作

なし。

例

次のクエリは、現在の月の 1 日ごとに 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、郵便番号が(0-9999)、(10000-19999)、...、(90000-99999)の範囲の地域に居住している従業員数を示します。従業員のいない範囲もあり、その場合は警告が発生します。

sa_rowgenerator プロシージャを使用すると、ある範囲の郵便番号に従業員がいない場合でも、これらの範囲を生成できます。

```
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;
```

次の例は、データのローを 10 個生成し、それらを NewEmployees テーブルに挿入します。

```
INSERT INTO NewEmployees ( ID, Salary, Name )
SELECT row_num, CAST( RAND() * 1000 AS INTEGER ), 'Mary'
FROM sa_rowgenerator( 1, 10 );
```

次の例は、sa_rowgenerator システムプロシージャを使用して、すべての整数を含むビューを作成します。この例の値 2147483647 は、サポートされている最大の符号付き整数を表します。

```
CREATE VIEW Integers AS
SELECT row_num AS n
FROM sa_rowgenerator( 0, 2147483647, 1 );
```

次の例は、sa_rowgenerator システムプロシージャを使用して、0001-01-01 ~ 9999-12-31 の日付を含むビューを作成します。この例の値 3652058 は、0001-01-01 ~ 9999-12-31 (それぞれサポートされている最初の日付と最後の日付) の日数を表します。

```
CREATE VIEW Dates AS
SELECT DATEADD( day, row_num, '0001-01-01' ) AS d
FROM sa_rowgenerator( 0, 3652058, 1 );
```

次のクエリは、1900 年から 2058 年までの間の 54 週あるすべての年を返します。

```
SELECT DATEADD ( day, row_num, '1900-01-01' ) AS d, DATEPART ( week,
d ) w
FROM sa_rowgenerator ( 0, 63919, 1 )
WHERE w = 54;
```

sa_server_option システムプロシージャ

サーバの実行中にサーバオプションを上書きします。

構文

```
sa_server_option( opt , val )
```

パラメータ

- **opt** – サーバオプション名を指定する CHAR(128) パラメータ。
- **val** – サーバオプションの新しい値を指定する CHAR(128) パラメータ。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、次のものも必要です。アプリケーションプロファイリングまたは要求ロギングに関連する次のオプションを使用する場合は、MANAGE PROFILING システム権限が必要です。

- ProcedureProfiling
- ProfileFilterConn
- ProfileFilterUser
- RequestFilterConn
- RequestFilterDB
- RequestLogFile
- RequestLogging
- RequestLogMaxSize
- RequestLogNumFiles

備考

オプション名	値	その他の情報
AutoMultiProgrammingLevel	YES、NO	デフォルトは YES。 YES に設定すると、データベースサーバでマルチプログラミングレベルが自動的に調整される。マルチプログラミングレベルは、同時にアクティブにできるタスクの最大数を制御する。このオプションを NO に設定して、マルチプログラミングレベルを手動で制御するようにした場合でも、マルチプログラミングレベルの初期値、最小値、最大値は設定できる。

オプション名	値	その他の情報
AutoMultiProgrammingLevelStatistics	YES、NO	デフォルトは NO。 YES に設定すると、マルチプログラミングレベルの自動調整の統計がデータベースサーバのメッセージログに表示される。
CacheSizingStatistics	YES、NO	デフォルトは NO。 YES に設定すると、キャッシュサイズが変更されるたびに、データベースサーバメッセージウィンドウにキャッシュ情報が表示される。
CollectStatistics	YES、NO	デフォルトは YES。 YES に設定すると、データベースサーバによってパフォーマンスモニタの統計情報が収集される。
ConnsDisabled	YES、NO	デフォルトは NO。 YES に設定すると、データベースサーバ上のデータベースに対する他の接続は許可されない。
ConnsDisabled-ForDB	YES、NO	デフォルトは NO。 YES に設定すると、現在のデータベースに対するその他の接続は許可されない。
ConsoleLogFile	filename	データベースサーバメッセージログ情報の記録に使用されるファイル名。空の文字列を指定すると、ファイルへのロギングが停止する。この値は SQL 文字列であるため、パス内の円記号を二重にする。
ConsoleLogMax-Size	file-size (バイト)	データベースサーバメッセージログ情報の記録に使用されるファイルの最大サイズ(バイト単位)。データベースサーバメッセージログファイルが、このプロパティまたは -on サーバオプションで指定されたサイズに達すると、ファイルが拡張子 .old の付いた名前に変更される (既存のファイルが存在する場合は、同じ名前で置換される)。その後、データベースサーバメッセージログファイルが再開される。

オプション名	値	その他の情報
CurrentMultiProgrammingLevel	integer	デフォルトは 20。 データベースサーバのマルチプログラミングレベルを設定する。
DatabaseCleaner	ON、OFF	デフォルトは ON。 テクニカルサポートの指示による場合を除き、このオプションの設定は変更しないこと。
DeadlockLogging	ON、OFF、 RESET、CLEAR	デフォルトは OFF。 デッドロックのロギングを制御する。値 <code>deadlock_logging</code> もサポートされる。サポートされる値は次のとおり。 <ul style="list-style-type: none"> • ON – デッドロックロギングを有効にする。 • OFF – デッドロックロギングを無効にするが、デッドロックデータの表示は可能。 • RESET – デッドロックデータが存在する場合は、記録済みのデッドロックデータをクリアし、その後、デッドロックロギングを有効にする。 • CLEAR – デッドロックデータが存在する場合は、記録済みのデッドロックデータをクリアし、その後、デッドロックロギングを無効にする。 デッドロックロギングが有効になると、 <code>sa_report_deadlocks</code> システムプロシージャを使用してデータベースからデッドロック情報を取得できる。
DebuggingInformation	YES、NO	デフォルトは NO。 診断メッセージなどのメッセージを表示してトラブルシューティングに使用できるようにする。メッセージは、データベースサーバメッセージウィンドウに表示される。

オプション名	値	その他の情報
DiskSandbox	ON、OFF	<p>デフォルトは OFF。</p> <p>データベースサーバ上で起動されたデータベースのうちディスクサンドボックス設定が明示的に行われていないすべてのデータベースについて、デフォルトのディスクサンドボックス設定を行う。sa_server_option システムプロシージャを使用してディスクサンドボックス設定を変更しても、データベースサーバですでに実行されているデータベースには影響しない。sa_server_option システムプロシージャを使用してディスクのサンドボックス設定を変更する場合は、manage_disk_sandbox セキュリティ機能のセキュリティ機能キーを指定する必要がある。</p>
DropBadStatistics	YES、NO	<p>デフォルトは YES。</p> <p>自動統計管理で、不適切な推定値を返す統計をデータベースから削除できるようにする。</p>
DropUnusedStatistics	YES、NO	<p>デフォルトは YES。</p> <p>自動統計管理で、連続 90 日間使用されなかった統計をデータベースから削除できるようにする。</p>
IdleTimeout	整数 (分)	<p>デフォルトは 240。</p> <p>指定された時間 (分単位) にわたって、要求を送信しなかった TCP/IP 接続を切断する。こうすることで、アクティブではない接続がロックを無制限に保持することが回避される。</p>
IPAddressMonitor-Period	整数 (秒)	<p>最小値は 10、デフォルト値は 0。ポータブルデバイスのデフォルト値は 120。</p> <p>新しい IP アドレスのチェック間隔を秒単位で設定する。</p>

オプション名	値	その他の情報
LivenessTimeout	整数 (秒)	デフォルトは 120。 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 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る。ネットワークサーバが、活性パケットを検出することなく、LivenessTimeout 期間まで実行されると、通信が切断される。
MaxMultiProgrammingLevel	整数	デフォルトは CurrentMultiProgrammingLevel の値の 4 倍。 データベースサーバのマルチプログラミングレベルの最大値を設定する。
MessageCategory-Limit	整数	デフォルトは 400。 sa_server_messages システムプロシージャを使用して取得可能なメッセージの最小数を、重大度レベルとカテゴリごとに設定する。
MinMultiProgrammingLevel	整数	デフォルトは -gtc サーバオプション値とコンピュータの論理 CPU 数の最小値。
OptionWatchAction	MESSAGE、 ERROR	デフォルトは MESSAGE。 リストにオプションを設定しようとしたときにデータベースサーバが実行するアクションを指定する。OptionWatchAction を MESSAGE に設定し、OptionWatchList で指定されたオプションを設定した場合、データベースサーバメッセージウィンドウにメッセージが表示され、その設定したオプションがオプションウォッチリストに含まれていることが示される。 OptionWatchAction を ERROR に設定すると、そのオプションはオプションウォッチリストに含まれているため設定できないことを示すエラーが返される。 このプロパティの現在の設定は、次の実行によって表示できる。 <pre>SELECT DB_PROPERTY('OptionWatchAction');</pre>

オプション名	値	その他の情報
OptionWatchList	データベースオプションのカンマ区切りリスト	<p>設定されたデータベースオプションのうち、通知の対象としたり、データベースサーバからエラーが返されるようにするオプションのリストを、カンマで区切って指定する。この文字列の長さは 128 バイトに制限される。デフォルトは、空の文字列。たとえば、次のコマンドでは、automatic_timestamp、float_as_double、tsql_hex_constant の各オプションがウォッチするオプションのリストに追加される。</p> <pre>CALL sa_server_option('OptionWatchList','automatic_timestamp, float_as_double,tsql_hex_constant');</pre> <p>このプロパティの現在の設定は、次の実行によって表示できる。</p> <pre>SELECT DB_PROPERTY('OptionWatchList');</pre>
ProcedureProfiling	YES、NO、RESET、CLEAR	デフォルトは NO。
ProfileFilterConn	connection-id	<p>他の接続によるデータベースの使用を阻害することなく、特定の接続 ID に関するプロファイル情報を取得するように、データベースサーバに指示する。接続フィルタが有効な場合、SELECT PROPERTY('ProfileFilterConn') の戻り値はモニタ対象の接続の接続 ID。ID が指定されていない場合、または接続フィルタが無効な場合の戻り値は -1。</p>

オプション名	値	その他の情報
ProcessorAffinity	プロセッサの番号、範囲、またはその両方のカンマ区切りのリスト。デフォルトは、すべてのプロセッサを使用するか、または <code>-gta</code> オプションの設定。	<p>データベースサーバに対して、Windows 上または Linux 上で使用する論理プロセッサを指示する。プロセッサの番号、範囲、またはその両方のカンマ区切りのリストを指定する。範囲の下限ポイントを省略すると、ゼロと見なされる。範囲の上限ポイントを省略すると、オペレーティングシステムで認識される最上位の CPU であると思なされる。sa_cpu_topology システムプロシージャで返される in_use カラムには、データベースサーバの現在のプロセッサのアフィニティが格納される。in_use カラムは、データベースサーバがプロセッサを使用しているかどうかを示す。または、ProcessorAffinity データベースサーバプロパティの値はクエリすることも可能。</p> <p>次の場合、指定されたすべての論理プロセッサをデータベースサーバが使用するわけではないこともある。</p> <ul style="list-style-type: none"> 指定された論理プロセッサのうち、1 つ以上が存在しないか、オフラインである場合。 ライセンスで使用が許可されない場合。 <p>無効なプロセッサ ID を指定すると、sa_server_option からエラーが返される。</p>
ProfileFilterUser	user-id	データベースサーバに対して、特定のユーザ ID のプロファイリング情報を取得するよう指示する。
QuittingTime	有効な日付と時刻	データベースサーバに対して、指定された時間にサーバを停止するよう指示する。

オプション名	値	その他の情報
RememberLastPlan	YES、NO	<p>デフォルトは NO。</p> <p>データベースサーバに対して、その接続で最後に実行されたクエリの長いテキストプランをキャプチャするように指示する。この設定は、-zp サーバオプションでも制御される。</p> <p>RememberLastPlan が ON の場合は、LastPlanText 接続プロパティの値を問い合わせることで、この接続で最後に実行されたクエリの計画のテキスト表現を取得する。</p> <pre>SELECT CONNECTION_PROPERTY('Last-PlanText');</pre>

オプション名	値	その他の情報
RememberLast-Statement	YES、NO	<p>デフォルトは NO。</p> <p>データベースサーバに対して、サーバ上で実行されている各データベースに関して最後に作成された SQL 文を取得するように指示する。ストアードプロシージャコールの場合、プロシージャ内の文ではなく、最も外側のプロシージャコールのみが表示される。RememberLastStatement が ON の場合は、LastStatement 接続プロパティの値を問い合わせることで、特定の接続の LastStatement の現在の値を取得できる。</p> <pre data-bbox="655 586 1180 638">SELECT CONNECTION_PROPERTY('Last-Statement');</pre> <p>クライアントでの文のキャッシュが有効で、キャッシュされた文が再使用される場合、このプロパティは空の文字列を返す。</p> <p>RememberLastStatement が ON の場合、次の文によって指定された接続に対して最後に作成された文が返される。</p> <pre data-bbox="655 868 1180 920">SELECT CONNECTION_PROPERTY('Last-Statement', connection-id);</pre> <p>sa_conn_activity システムプロシージャは、すべての接続に対して同じ情報を返す。</p> <hr/> <p>注意： -zl が指定されている場合、または RememberLastStatement サーバ設定がオンになっている場合、すべてのユーザが sa_conn_activity システムプロシージャを呼び出すか、LastStatement 接続プロパティの値を取得して、他のユーザが最後に作成した SQL 文を見つけることができる。このオプションは注意して使用し、不要になったらオフにする。</p>

オプション名	値	その他の情報
RequestFilterConn	connection-id、 -1	<p>要求ログ情報をフィルタして、特定の接続の情報のみが記録されるようにする。このフィルタリングによって、多くのアクティブな接続や複数のデータベースがあるデータベースサーバを監視するときに、要求ログファイルのサイズを削減できる。次の文を実行して、接続 ID を取得できる。</p> <pre>CALL sa_conn_info();</pre> <p>接続 ID を取得した後で特定の接続をロギングするには、次の文を実行する。</p> <pre>CALL sa_server_option('RequestFilterConn', connection-id);</pre> <p>フィルタは、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なままになる。フィルタをリセットするには、次の文を使用します。</p> <pre>CALL sa_server_option('RequestFilterConn', -1);</pre>
RequestFilterDB	database-id、 -1	<p>要求ログ情報をフィルタして、特定のデータベースの情報のみが記録されるようにする。これによって、複数のデータベースのあるサーバを監視するときに、要求ログファイルのサイズを削減できる。目的のデータベースに接続しているときに次の文を実行して、データベース ID を取得できる。</p> <pre>SELECT CONNECTION_PROPERTY('DBNumber');</pre> <p>特定のデータベースの情報のみをロギングするには、次の文を実行する。</p> <pre>CALL sa_server_option('RequestFilterDB', database-id);</pre> <p>フィルタは、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なままになる。フィルタをリセットするには、次の文を使用する。</p> <pre>CALL sa_server_option('RequestFilterDB', -1);</pre>

オプション名	値	その他の情報
RequestLogFile	filename	<p>要求情報の記録に使用されるファイルの名前。空の文字列を指定すると、要求ログファイルへのログインが停止する。要求のログインが有効でも、要求のログファイルを指定しなかった場合、または空の文字列に設定されている場合、サーバは要求をデータベースサーバメッセージウィンドウにログインする。この値は SQL 文字列であるため、パス内の円記号を二重にする。</p> <p>クライアントでの文のキャッシュが有効であり、かつ、ログが tracetime.pl Perl スクリプトを使用して分析される場合は、要求ログが取得される間、max_client_statements_cached オプションを 0 に設定して、クライアントでの文のキャッシュを無効にする必要がある。</p>

オプション名	値	その他の情報
RequestLogging	SQL、 HOSTVARS、 PLAN、 PROCEDURES、 TRIGGERS、 OTHER、 BLOCKS、 REPLACE、 ALL、YES、 NONE、NO	<p>デフォルトは NONE。</p> <p>データベースサーバオプション <code>-zr</code> と <code>-zo</code> とともにトラブルシューティングで使用するためにデータベースサーバに送信される個々の SQL 文のロギングを有効にする。次の値をプラス記号 (+) またはカンマで区切って組み合わせた値。</p> <ul style="list-style-type: none"> • PLAN – 実行プランのロギングを有効にする (短いプラン)。プロシージャ (PROCEDURES) のロギングが有効な場合は、プロシージャの実行プランも記録される。 • HOSTVARS – ホスト変数の値のロギングを有効にする。HOSTVARS を指定した場合、SQL にリストされている情報もロギングされる。 • PROCEDURES – プロシージャ内から実行されている文のロギングを有効にする。 • TRIGGERS – トリガ内から実行されている文のロギングを有効にする。 • OTHER – SQL に含まれないその他の要求タイプのロギングを有効にする (FETCH や PREFETCH など)。ただし、OTHER を指定して SQL を指定しない場合、SQL+OTHER を指定した場合と同じ。OTHER を含めると、ログファイルが急速に拡大し、サーバのパフォーマンス低下につながる可能性がある。 • BLOCKS – 別の接続で接続がブロックされたときと、接続のブロックが解除されたときに表示する詳細のロギングを有効にする。 • REPLACE – ロギングの開始時に、既存の要求ログは同じ名前を持つ新規の (空の) ログで置換される。それ以外の場合、既存の要求ログが開き、新規エントリがファイルの末尾に追加されます。 • ALL – すべてのサポート情報をロギングする。この値は、SQL+PLAN+HOSTVARS+PROCEDURES+TRIGGERS+OTHER+BLOCKS を指定した場合と同じ。この設定では、ログファイルが急速に拡大し、サーバ

オプション名	値	その他の情報
		<p>のパフォーマンス低下につながる可能性がある。</p> <ul style="list-style-type: none"> • NO または NONE – 要求ログに対するロギングを無効にする。 <p>このプロパティの現在の設定は、次の実行によって表示できる。</p> <pre>SELECT PROPERTY('RequestLogging');</pre>
RequestLogMax-Size	file-size (バイト)	<p>要求ログ情報の記録に使用されるファイルのバイト単位での最大サイズ。0を指定した場合は、要求ロギングファイルの最大サイズは適用されず、ファイルの名前は変更されない。これはデフォルト値。要求ログファイルが、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、ファイルが拡張子 .old の付いた名前に変更される (既存のファイルが存在する場合は、同じ名前で置換される)。要求ログファイルが再起動する。</p>
RequestLogNum-Files	整数	<p>保持する要求ログファイルのコピーの数。要求ロギングが長期間にわたって有効になっていると、要求ログファイルが大きくなることもある。-zn オプションを使用すると、保持する要求ログファイルのコピー数を指定できる。</p>

オプション名	値	その他の情報
RequestTiming	YES、NO	<p>デフォルトは NO。</p> <p>データベースサーバに、新しい各接続のタイミング情報を保守するように指示する。この機能はデフォルトでオフになっている。オンにすると、データベースサーバはすべての新しい接続の累積タイマを管理する。このタイマは、接続が確立した状態でのサーバステータスごとの時間を示す。変更は新しい接続についてのみ有効で、各接続の期間持続する。</p> <p>sa_performance_diagnostics システムプロシージャを使用して、このタイミング情報の概要を取得できる。または、次の接続プロパティを調べて各値を取得できる。</p> <ul style="list-style-type: none"> • ReqCountUnscheduled • ReqTimeUnscheduled • ReqCountActive • ReqTimeActive • ReqCountBlockIO • ReqTimeBlockIO • ReqCountBlockLock • ReqTimeBlockLock • ReqCountBlockContention • ReqTimeBlockContention <p>RequestTiming サーバプロパティがオンであると、追加のカウンタを管理するため、要求ごとに多少のオーバーヘッドがかかる。</p>
rlv_auto_merge	ON、OFF	<p>デフォルトは ON。</p> <p>ローレベルのバージョン管理が有効なテーブルで、IQ メインストアへの RLV ストアの自動マージを有効または無効にする。</p> <p>rlv_auto_merge が OFF の場合は、RLV ストアと IQ メインストアの自動マージは行われない。つまり、rlv_memory_mb の上限スレッシュホールドに達する前に、RLV ストアが IQ メインストアに同期されるようにデータを手動でマージする必要がある。</p>

オプション名	値	その他の情報
rlv_memory_mb	最小値は 1 MB。 最大値は 2048。 それ以外の値を入力しても、メモリ量は 2048 MB に設定される。	メモリ (RLV ストア) の最大量を MB 単位で指定して、新しいレベルのバージョン用に予約する。デフォルト値は 2,048 MB。値がシステム仮想メモリの上限の 2/3 を超え、サーバはエラーを生成します の場合。

オプション名	値	その他の情報
SecureFeatures	feature-list	<p>すでに動作しているデータベースサーバのセキュリティ機能を管理できる。feature-list は、機能名または機能セットのカンマ区切りリスト。このリストに機能を追加することで、追加した機能の可用性を制限する。セキュリティ機能のリストから項目を削除するには、セキュリティ機能名の前にマイナス記号 (-) を指定する。</p> <p>sa_server_option('SecureFeatures',...) を呼び出す場合は、接続で ManageFeatures セキュリティ機能が有効になっている必要がある。-sf キー (システムセキュリティ機能キー) は、その他のすべての機能とともに ManageFeatures を有効にする。したがって、システムセキュリティ機能キーを使用する場合は、SecureFeatures セットの変更は接続で有効にならない。ただし、その他のキー (たとえば、create_secure_feature_key システムプロシージャを使用して作成されたキー) を使用する場合は、キーに含まれる他の機能によっては、接続で変更が直ちに有効になる。</p> <p>機能へのアクセスを許可または禁止する変更を行うと、データベースサーバで直ちに有効になる。sa_server_option システムプロシージャを実行する接続は、その接続が使用しているセキュリティ機能と、指定した機能へのアクセスが許可されているかどうかによって、変更が有効または無効になる。</p> <p>たとえば、2つの機能を保護するには、次の構文を使用する。</p> <pre>CALL sa_server_option('SecureFeatures', 'CONSOLE_LOG,WEBCLIENT_LOG');</pre> <p>この文を実行すると、変更内容に応じてセキュリティ機能のリストが設定される。</p>

オプション名	値	その他の情報
StatisticsCleaner	ON、OFF	<p>デフォルトは ON。</p> <p>統計クリーナーでは、テーブルに対してスキャンを実行して、不適切な推定値を示す統計を修正する。デフォルトでは、統計クリーナーはバックグラウンドで実行され、パフォーマンスへの影響は最小限に抑えられる。</p> <p>統計クリーナーをオフにしても統計ガバナンスは無効にならないが、統計クリーナーがオフのときには、統計はクエリの実行時にのみ作成または修正される。</p>
WebClientLogFile	filename	<p>Web サービスクライアントログファイルの名前。Web サービスクライアントログファイルは、-zoc サーバオプションや WebClientLogFile プロパティを使用して、ファイル名を設定またはリセットするたびにトランケートされる。この値は文字列なので、パス内の円記号を二重にする。</p>
WebClientLogging	ON、OFF	<p>デフォルトは OFF。</p> <p>このオプションは、Web サービスクライアントのロギングを有効または無効にする。ログに記録される情報には、HTTP の要求と応答のデータが含まれている。ON を指定すると Web サービスクライアントログファイルへのロギングが開始され、OFF を指定するとファイルへのロギングが中止される。</p>

関連する動作

なし。

例

次の文を使用した場合は、キャッシュサイズを変更するたびに、データベースサーバメッセージウィンドウにキャッシュ情報が表示されるようになります。

```
CALL sa_server_option( 'CacheSizingStatistics', 'YES' );
```

次の文は、現在のデータベースへの新しい接続を禁止します。

```
CALL sa_server_option( 'ConnsDisabledForDB', 'YES' );
```

次の文は、すべての SQL 文、プロシージャの呼び出し、プラン、イベントのブロックとブロック解除のロギングを有効にし、新しい要求ログを開始します。

```
CALL sa_server_option( 'RequestLogging', 'SQL+PROCEDURES+BLOCKS+PLAN+REPLACE' );
```

参照:

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_set_http_header システムプロシージャ

Web サービスによる HTTP 応答ヘッダの設定を許可します。

構文

```
sa_set_http_header(  
fldname  
, val  
)
```

引数

- fldname** – HTTP ヘッダフィールドのいずれかの名前を含む文字列を指定する CHAR(128) パラメータ。
- val** – 指定されたパラメータに設定する値を指定する LONG VARCHAR パラメータ。応答ヘッダを NULL に設定すると、そのヘッダは事実上削除されます。

備考

特別なヘッダフィールド `@HttpStatus` を設定すると、要求によってステータスコードが返されるように設定されます。ステータスコードは応答コードとも呼ばれます。たとえば、次のスクリプトはステータスコードを 404 Not Found に設定します。

```
CALL sa_set_http_header( '@HttpStatus', '404' );
```

3 桁のステータスコードを指定し、コロンで区切ったテキストメッセージを任意で追加することで、ユーザ定義のステータスメッセージを作成できます。たとえば、次のスクリプトは、メッセージ付きのステータスコード "999 User Code" を出力します。

```
CALL sa_set_http_header( '@HttpStatus', '999:User Code' );
```

注意: LogOptions プロトコルオプションを使用してログに記録している場合、ユーザ定義のステータステキストメッセージはデータベースの文字セットには変換されません。

エラーメッセージの本文は、自動的に挿入されます。有効な HTTP エラーコードだけが使用できます。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。

`sa_set_http_header` プロシージャは、呼び出されたときに、ヘッダフィールドの既存のヘッダ値を必ず上書きします。

データベースサーバによって自動的に生成された応答ヘッダは削除できます。たとえば、次のコマンドは Expires 応答ヘッダを削除します。

```
CALL sa_set_http_header( 'Expires', NULL );
```

権限
なし。

関連する動作
なし。

例

次の例は、Content-Type ヘッダフィールドを text/html に設定します。

```
CALL sa_set_http_header( 'Content-Type', 'text/html' );
```

sa_set_http_option システムプロシージャ

Web サービスによるプロセス制御用の HTTP オプションの設定を許可します。

構文

```
sa_set_http_option(  
  optname  
, val  
)
```

引数

- **optname** – HTTP オプションのいずれかの名前を含む文字列を指定する CHAR(128) パラメータ。

サポートされるオプションは次のとおりです。

- **CharsetConversion** – このオプションを使用して、結果セットをデータベースの文字セットエンコードからクライアントの文字セットエンコードに自動的に変換するかどうかを制御します。指定できる値は、ON と OFF だけです。デフォルト値は ON です。
- **AcceptCharset** – このオプションを使用して、応答の文字セットエンコードに関する Web サーバの優先度を指定します。1つ以上の文字セットエンコードを優先度順に指定できます。このオプションの構文は、RFC2616 Hypertext Transfer Protocol の HTTP Accept-Charset 要求ヘッダフィールドの仕様に使用する構文に準拠します。

Web ブラウザなどの HTTP クライアントでは、Accept-Charset 要求ヘッダを使用して、優先度順の文字セットエンコードのリストを指定できます。必要に応じて、各エンコードには、対応する品質値 ($q=qvalue$) を指定できます。この品質値は、そのエンコードに対するクライアントの優先度を表します。デフォルトでは、品質値は 1 ($q=1$) です。次に例を示します。

```
Accept-Charset: iso-8859-5, utf-8;q=0.8
```

AcceptCharset HTTP オプション値内のプラス記号 (+) は、現在のデータベースの文字セットエンコードを表すショートカットとして使用できます。また、プラス記号は、クライアントのリストでもエンコードが指定されている場合に、クライアントによって割り当てられている品質値に関係なく、データベースの文字セットエンコードを優先する必要があることも示します。

AcceptCharset HTTP オプション内のアスタリスク (*) は、クライアントとサーバのリストに共通部分がない場合に、クライアントで優先される文字セットエンコードがサーバでもサポートされていれば、Web サービスでその文字セットエンコードを使用する必要があることを示します。

応答の送信時には、クライアントと Web サービスの両方で優先される最初の文字セットエンコードが使用されます。クライアントの優先度順が最初に考慮されます。共通のエンコード優先度が存在しない場合は、Web サービスのリストにアスタリスク (*) がなければ、Web サービスで最も優先されるエンコードが使用されます。アスタリスクがある場合は、クライアントで最も優先されるエンコードが使用されます。

AcceptCharset HTTP オプションを使用しない場合は、クライアントで指定され、サーバでサポートされている最も優先度の高い文字セットエンコードが使用されます。クライアントで指定されたどのエンコードもサポートされていない場合 (またはクライアントが Accept-Charset 要求ヘッダを送信しない場合) は、データベースの文字セットエンコードが使用されます。

クライアントが Accept-Charset ヘッダを送信しない場合は、次のいずれかのアクションが実行されます。

- AcceptCharset HTTP オプションが指定されていない場合、Web サーバではデータベースの文字セットエンコードを使用します。
- AcceptCharset HTTP オプションが指定されている場合、Web サーバでは最も優先度の高い文字セットエンコードを使用します。

クライアントが Accept-Charset ヘッダを送信する場合は、次のいずれかのアクションが実行されます。

- AcceptCharset HTTP オプションが指定されていない場合、Web サーバではクライアントで優先される文字セットエンコードのいずれかを使用しようとする (最も優先度の高いエンコードから開始)。クライアントで優先されるどのエンコードも Web サーバでサポートされていない場合、Web サーバではデータベースの文字セットエンコードを使用します。
- AcceptCharset HTTP オプションが指定されている場合、Web サーバでは両方の優先度リストで共通する最初の文字セットエンコードを使用しようとする (クライアントで最も優先されるエンコードから開始)。たと

例えば、クライアントが送信する Accept-Charset ヘッダリストのエンコードの優先度順が iso-a、iso-b、iso-c であり、Web サーバでの優先度順が iso-b、iso-a、iso-c の場合、iso-a が選択されます。

```
Web client: iso-a, iso-b, iso-c
Web server: iso-b, iso-a, iso-c
```

2つのリストに共通部分がない場合、Web サーバで最初に優先される文字セットが使用されます。次の例では、エンコード iso-d が使用されません。

```
Web client: iso-a, iso-b, iso-c
Web server: iso-d, iso-e, iso-f
```

AcceptCharset HTTP オプションにアスタリスク (*) が含まれている場合は、クライアントのエンコードが優先されるため、iso-a が使用されます。基本的に、アスタリスクを使用すると、2つのリストに共通部分がないという状況がなくなります。

理想的な状況は、クライアントと Web サービスの両方でデータベースの文字セットエンコードが使用される場合です。このような場合、文字セットの変換の必要性がなくなり、Web サーバの応答時間が向上します。

CharsetConversion オプションを OFF に設定している場合、AcceptCharset の処理は実行されません。

- **SessionID** – このオプションを使用して、HTTP セッションの作成、削除、または名前の変更を行います。Web サービスでこのオプションを設定して HTTP セッションを作成するときにはデータベース接続は維持されますが、サーバの再起動時にはセッションは維持されません。すでにセッションコンテキスト内にある場合、この呼び出しによってセッションの名前が新しいセッション ID に変更されます。NULL 値を使用して呼び出した場合、Web サービスが終了するとセッションは削除されます。

生成されるセッションキーの長さは 128 文字に制限され、複数のデータベースがロードされている場合はデータベース間でユニークになります。

- **SessionTimeout** – このオプションを使用して、非アクティブ状態の HTTP セッションを維持する時間を分単位で指定します。このタイムアウト期間は、HTTP 要求で指定のセッションが使用されると常にリセットされます。SessionTimeout を過ぎると、セッションは自動的に削除されます。
- **val** – 指定されたオプションに設定する値を指定する LONG VARCHAR パラメータ。

備考

Web サービスを処理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て、オプションを設定します。

sa_set_http_option が Web サービスによって開始されたプロシージャ内から呼び出され、オプションまたはオプションの値のいずれかが無効な場合は、エラーが返ります。

権限
なし。

関連する動作
なし。

例

次の例は、sa_set_http_option を使用して、データベースの文字セットエンコードに関する Web サービスの優先度を指定する方法を示します。2 番目の選択肢として UTF-8 エンコードが指定されています。アスタリスク (*) は、クライアントで最も優先される文字セットエンコードが Web サーバでサポートされている場合に、Web サービスでその文字セットエンコードが使用されることを示しています。

```
CALL sa_set_http_option( 'AcceptCharset', '+,UTF-8,*');
```

次の例は、sa_set_http_option を使用して、Web サービスで使用されている文字エンコードを正しく識別する方法を示します。この例では、Web サーバは 1251CYR データベースに接続されており、キリル語のアルファベットを含む HTML ドキュメントを任意の Web ブラウザに提供するように準備されています。

```
CREATE OR REPLACE PROCEDURE cyrillic_html()
RESULT (html_doc XML)
BEGIN
    DECLARE pos INT;
    DECLARE charset VARCHAR(30);
    CALL sa_set_http_option( 'AcceptCharset', 'iso-8859-5, utf-8' );
    SET charset = CONNECTION_PROPERTY( 'CharSet' );
    -- Change any IANA labels like ISO_8859-5:1988
    -- to ISO_8859-5 for Firefox.
    SET pos = LOCATE( charset, ':' );
    IF pos > 0 THEN
        SET charset = LEFT( charset, pos - 1 );
    END IF;
    CALL sa_set_http_header( 'Content-Type', 'text/html; charset=' ||
        charset );
    SELECT '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">' ||
        XMLCONCAT(
            XMLELEMENT('HTML',
                XMLELEMENT('HEAD',
                    XMLELEMENT('TITLE', 'Cyrillic characters')
                ),
                XMLELEMENT('BODY',
                    XMLELEMENT('H1', 'First 5 lowercase Russian letters'),
                    XMLELEMENT('P', UNISTR('Ўu0430Ўu0431Ўu0432Ўu0433Ўu0434'))
                )
            )
    )
```

```
);
END;

CREATE SERVICE cyrillic
TYPE 'RAW'
AUTHORIZATION OFF
USER DBA
AS CALL cyrillic_html();
```

Firefox などの Web ブラウザから Web サービスに配信される、次の Accept-Charset ヘッダを例に、使用する正しい文字セットエンコードを確立するプロセスについて説明します。このヘッダは、ブラウザでは ISO-8859-1 と UTF-8 のエンコードが優先されるものの、他のエンコードも処理できることを示しています。

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

転送される Web ページにキリル語の文字が含まれているため、Web サービスでは ISO-8859-1 文字セットエンコードは受け入れられません。Web サービスでは、sa_set_http_option の呼び出しで指定されているように、ISO-8859-5 または UTF-8 エンコードが優先されます。この例では、UTF-8 エンコードが選択されます (両方で許容されるため)。データベース接続プロパティ CharSet は、どのエンコードが Web サービスで選択されたかを示します。sa_set_http_header プロシージャは、Web ブラウザに対して HTML ドキュメントのエンコードを示すために使用します。

```
Content-Type: text/html; charset=UTF-8
```

Web ブラウザで Accept-Charset が指定されていない場合、Web サービスではデフォルトで最初の優先エンコード ISO-8859-5 が使用されます。sa_set_http_header プロシージャは、HTML ドキュメントのエンコードを示すために使用します。

```
Content-Type: text/html; charset=ISO_8859-5
```

次の例では、ユニークな HTTP セッション識別子を設定します。

```
BEGIN
  DECLARE sessionid VARCHAR(30);
  DECLARE tm TIMESTAMP;
  SET tm = NOW(*);
  SET sessionid = 'MySessions_' ||
    CONVERT( VARCHAR, SECONDS(tm)*1000 + DATEPART(millisecond,tm));
  SELECT sessionid;
  CALL sa_set_http_option('SessionID', sessionid);
END;
```

次の例では、HTTP セッションのタイムアウトを 5 分に設定します。

```
CALL sa_set_http_option('SessionTimeout', '5');
```

sa_table_page_usage システムプロシージャ

データベーステーブルのページ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( )
```

結果セット

カラム名	データ型	説明
TableId	UNSIGNED INTEGER	テーブル ID。
TablePages	INTEGER	テーブルで使用されるテーブルページの数。
PctUsedT	INTEGER	使用されているテーブルページ領域の割合。
IndexPages	INTEGER	テーブルで使用されるインデックスページの数。
PctUsedI	INTEGER	使用されているインデックスページ領域の割合。
PctOfFile	INTEGER	テーブルが使用しているデータベースファイル総数の割合。
TableName	CHAR(128)	テーブル名。

備考

結果には、情報ユーティリティで提供される情報と同じ内容が含まれています。progress_messages データベースオプションが Raw または Formatted に設定されている場合は、sa_table_page_usage システムプロシージャの実行中にデータベースサーバからクライアントに進行メッセージが送信されます。

権限

MANAGE ANY DBSPACE システム権限が必要です。

関連する動作

なし。

例

次の例では、SalesOrderItems テーブルのページの使用状況に関する情報を取得します。

```
SELECT * FROM sa_table_page_usage( )
WHERE TableName = 'SalesOrderItems';
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_text_index_stats システムプロシージャ

データベース内の TEXT インデックスに関する統計情報を返します。

構文

```
sa_text_index_stats( )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- MANAGE ANY STATISTICS
- CREATE ANY INDEX
- ALTER ANY INDEX
- DROP ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- DROP ANY OBJECT

備考

sa_text_index_stats を使用して、データベース内の各 TEXT インデックスの統計情報を表示します。

表 120 : sa_text_index_stats によって返される TEXT インデックスの統計情報

カラム名	データ型	説明
owner_id	UNSIGNED INT	テーブルの所有者の ID
table_id	UNSIGNED INT	テーブルの ID
index_id	UNSIGNED INT	TEXT インデックスの ID
text_config_id	UNSIGNED BIGINT	TEXT インデックスによって参照されるテキスト設定の ID
owner_name	CHAR(128)	所有者の名前
table_name	CHAR(128)	テーブルの名前
index_name	CHAR(128)	TEXT インデックスの名前
text_config_name	CHAR(128)	テキスト設定オブジェクトの名前
doc_count	UNSIGNED BIGINT	TEXT インデックスに含まれるインデックス付けされたカラム値の総数

カラム名	データ型	説明
doc_length	UNSIGNED BIGINT	TEXT インデックス内のデータの長さの合計
pending_length	UNSIGNED BIGINT	保留中の変更の合計長
deleted_length	UNSIGNED BIGINT	保留中の削除の合計長
last_refresh	TIMESTAMP	最終更新日時

IMMEDIATE REFRESH TEXT インデックスの場合、pending_length、deleted_length、および last_refresh の値は NULL です。

例

データベース内の各 TEXT インデックスに関する統計情報を返します。

```
CALL sa_text_index_stats( );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_text_index_vocab システムプロシージャ

TEXT インデックスに含まれるすべての単語と、各単語が含まれるインデックス値の合計数のリストを返します。

構文

```
sa_text_index_vocab (
    'text-index-name',
    'table-name',
    'table-owner'
)
```

パラメータ

- **text-index-name** – この CHAR(128) パラメータを使用して、TEXT インデックスの名前を指定します。
- **table-name** – この CHAR(128) パラメータを使用して、TEXT インデックスが構築されているテーブルの名前を指定します。
- **table-owner** – この CHAR(128) パラメータを使用して、テーブルの所有者を指定します。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- SELECT ANY TABLE システム権限
- インデックステーブルに対する SELECT 権限

備考

sa_text_index_vocab は、TEXT インデックスに含まれるすべての単語と、各単語が含まれるインデックス値の合計数を返します (これは、単語が一部のインデックス値で複数回出現する場合、合計出現回数未満となります)。

パラメータ値はホスト変数またはホスト式にできません。引数の *text-index-name*、*table-name*、*table-owner* は、制約または変数である必要があります。

例

sa_text_index_vocab を実行して、GROUPO によって所有されるテーブル Customers の TEXT インデックス MyTextIndex に含まれるすべての単語を返します。

```
sa_text_index_vocab
('MyTextIndex', 'Customers', 'GROUPO');
```

表 121 : インデックス内の単語

用語	出現回数
a	1
Able	1
Acres	1
Active	5
Advertising	1
Again	1
...	...

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_validate システムプロシージャ

すべてまたは一部のデータベースでチェックサム検証を実行します。

構文

```
sa_validate(
[ tbl_name
```

```
[, owner_name ] ]
)
```

引数

- **tbl_name** – 検証するテーブルまたはマテリアライズドビューの名前を指定する任意の CHAR(128) パラメータ。デフォルトは NULL です。
- **owner_name** – 所有者を指定する任意の CHAR(128) パラメータ。所有者のみを指定すると、この所有者が所有するすべてのテーブルとマテリアライズドビューが検証されます。デフォルトは NULL です。

権限

VALIDATE ANY OBJECT システム権限が必要です。

関連する動作

なし。

備考

指定される引数	検証タイプ
なし	データベース内のすべてのテーブル、マテリアライズドビュー、インデックスが検証されます。チェックサムの検証を含め、データベース自体も検証されます。
<i>tbl_name</i>	現在のユーザが所有している指定されたテーブルまたはマテリアライズドビュー、およびそのすべてのインデックスが検証されます。
<i>owner_name</i>	指定されたユーザが所有するすべてのテーブル、マテリアライズドビュー、インデックスが検証されます。
<i>tbl_name</i> と <i>owner_name</i>	指定されたユーザが所有している指定されたテーブルまたはマテリアライズドビュー、およびそのすべてのインデックスが検証されます。

プロシージャは Messages という 1 つのカラムを返します。検証時にエラーが返されると、このカラムに表示されます。エラーなく検証が成功した場合は、カラムに「エラーは見つかりませんでした。」が格納されます。

警告！ 警告

テーブルまたはデータベース全体の検証は、データベースに変更を加えている接続がない場合に実行してください。そうしないと、実際に破損していなくても、何らかの形でデータベースが破損したことを示すエラーがレポートされます。

例

次の文は、ユーザ DBA が所有するテーブルとマテリアライズドビューの検証を実行します。

```
CALL sa_validate( owner_name = 'DBA' );
```

参照：

- データベースが使用するセキュリティモデルの特定 (432 ページ)

sa_verify_password システムプロシージャ

現在のユーザのパスワードを検証します。

構文

```
sa_verify_password( curr_pswd )
```

引数

- curr_pswd** – 現在のデータベースユーザのパスワードを指定する CHAR(128) パラメータ。

戻り値

INTEGER 値を返す関数。

備考

このプロシージャは sp_password で使用されます。パスワードが一致した場合は 0 が返され、エラーは発生しません。パスワードが一致しなかった場合、エラーと診断されます。パスワードが一致しない場合は、接続が終了しません。

権限

なし。

関連する動作

なし。

例

次の例では、現在のユーザが DBA または User1 の場合に現在の接続のパスワードの検証を試行します。現在のパスワードが一致しない場合は、エラーが発生します。

```
IF USER_NAME() = 'DBA' THEN
    SELECT sa_verify_password( 'sql' );
ELSEIF USER_NAME() = 'User1' THEN
    SELECT sa_verify_password( 'user' );
END IF;
```

sp_alter_secure_feature_key システムプロシージャ

認証キー、機能リスト、またはその両方を変更して、以前に定義されたセキュリティ機能キーを変更します。

構文

```
sp_alter_secure_feature_key (
    name,
    auth_key,
    features )
```

パラメータ

- **name** – 変更するセキュリティ機能キーの VARCHAR (128) 名。指定した名前のキーが既存である必要があります。
- **auth_key** – セキュリティ機能キーの CHAR (128) 認証キー。認証キーは、6 文字以上の空でない文字列か、または既存の認証キーが変更されないことを示す NULL である必要があります。
- **features** – キーで有効にできるセキュリティ機能のカンマ区切りのリスト (LONG VARCHAR)。feature_list には、既存の feature_list リストが変更されないことを示す NULL を指定できます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、データベースサーバ所有者であり、その接続に対する manage_keys 機能が有効である必要があります。

備考

このプロシージャを使用すると、既存のセキュリティ機能キーの認証キーまたは機能リストを変更できます。

sp_auth_sys_role_info システムプロシージャ

権限を対応するシステムロールおよびロール ID にマップするレポートを生成します。このプロシージャは、権限ごとに 1 つのローを返します。

構文

```
sp_auth_sys_role_info()
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

カラム名	データ型	説明
auth	varchar(20)	権限の名前。
role_name	char(128)	対応するシステムロールの名前。
role_id	unsigned int	システムロールの ID。

sp_create_secure_feature_key システムプロシージャ

新しいセキュリティ機能キーを作成します。

構文

```
sp_create_secure_feature_key (
    name,
    auth_key,
    features )
```

パラメータ

- **name** – 新しいセキュリティ機能キーの VARCHAR (128) 名。この引数は、NULL または空白文字列にできません。
- **auth_key** – セキュリティ機能キーの CHAR (128) 認証キー。認証キーは、6 文字以上の空でない文字列にする必要があります。
- **features** – 新しいキーで有効にできるセキュリティ機能のカンマ区切りのリスト (LONG VARCHAR)。機能の前に "-" を指定すると、このセキュリティ機能キーを設定してもその機能は再び有効にはならないことを意味します。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、データベースサーバ所有者であり、その接続に対する manage_keys 機能が有効である必要があります。

備考

このプロシージャでは、どのユーザにも設定可能な新しいセキュリティ機能が作成されます。システムセキュリティ機能キーは、-sk データベースサーバオプションを使用して作成されます。

sp_displayroles システムプロシージャ

ユーザ定義のロールまたはユーザに付与されているすべてのロールを表示するか、またはロールの階層ツリー全体を表示します。

構文

```
sp_displayroles (  
    [ user_role_name ],  
    [ display_mode ],  
    [ grant_type ] )
```

パラメータ

- **user_role_name** – 有効な値は次のとおりです。
 - 有効なシステム権限名またはシステム権限ロール名
 - 有効なユーザ定義ロール名
 - 有効なユーザ名

デフォルトでは、引数が指定されない場合、現在のログインユーザが使用されます。

- **display_mode** – 有効な値は次のとおりです。
 - **EXPAND_UP** – 入力されたロールまたはシステム権限が付与されているすべてのロールを表示します。これは、親レベルのロール階層ツリーです。
 - **EXPAND_DOWN** – 入力されたロールまたはユーザに付与されているすべてのロールまたはシステム権限を表示します。これは、子レベルのロール階層ツリーです。

引数が指定されない場合 (デフォルト)、直接付与されたロールまたはシステム権限のみが表示されます。

- **grant_type** – 有効な値は次のとおりです。
 - **ALL** – 付与されているすべてのロールまたはシステム権限を表示します。
 - **NO_ADMIN** – WITH NO ADMIN OPTION 句または WITH ADMIN OPTION 句を使用して付与されているすべてのロールまたはシステム権限を表示します。
 - **ADMIN** – WITH ADMIN OPTION 句または WITH ADMIN ONLY OPTION 句を使用して付与されているすべてのロールまたはシステム権限を表示します。

引数を指定しないと、[ALL] が使用されます。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。他のユーザに対してこのプロシージャを実行する場合は、MANAGE ROLES システム権限が必要

です。ロールまたはシステム権限に対して実行する場合は、そのロールの管理者であるか、システム権限に対する管理権限が必要です。

備考

カラム名	データ型	説明
role_name	char(128)	ロール/システム権限の名前をリストする。
parent_role_name	char(128)	親のロール名をリストする。
grant_type	char(10)	付与タイプをリストする。
role_level	smallint	Expand_down モードの場合、1 は直接付与されたロールを示し、2 は 1 つ下の階層を示し、以下同様。 Expand_up モードの場合、0 は指定されたロールが付与されているロールを示し、-1 は 1 つ上の階層を示し、以下同様。

名前としてシステム権限名が指定された場合、結果には、システム権限ロール名ではなくシステム権限名が表示されます。

モードとして Expand_down が指定された場合、レベル 1 の parent_role_name は NULL (直接付与されたロール) です。モードが指定されない場合 (デフォルト)、直接付与されたロールのみ表示されるので、role_level は 1、parent_role_name は NULL です。

名前としてユーザ名、モードとして expand_up が指定された場合、ユーザは任意のロール階層の最上位レベルに存在するので、結果は返されません。同様に、名前として不変のシステム権限名、モードとして Expand_down が指定された場合、不変のシステム権限は任意のロール階層の最下位レベルに存在するので、結果は返されません。

デフォルトモードの場合、parent_role_name カラムは NULL、role_level は 1 です。

例

次の例では、次の GRANT 文が実行されているとします。

```
GRANT SERVER OPERATOR TO r4;
GRANT BACKUP DATABASE TO r3 WITH ADMIN OPTION;
GRANT DROP CONNECTION TO r3 WITH ADMIN ONLY OPTION;
GRANT MONITOR TO r2;GRANT CHECKPOINT TO r1;
GRANT ROLE r2 TO r1 WITH ADMIN OPTION;
GRANT ROLE r3 TO r2 WITH NO ADMIN OPTION;
GRANT ROLE r4 TO r3 WITH ADMIN ONLY OPTION;
GRANT ROLE r1 TO user1;
GRANT ROLE r1 TO r7;
GRANT ROLE r7 TO user2 WITH ADMIN OPTION;
GRANT BACKUP DATABASE TO user2 WITH ADMIN ONLY OPTION;
```

sp_displayroles('user2', 'expand_down', 'ALL')を実行すると、次のような内容が出力されます。

role_name	parent_role_name	grant_type	role_level
r7	NULL	ADMIN	1
PUBLIC	NULL	NO ADMIN	1
BACKUP DATABASE	NULL	ADMIN ONLY	1
dbo	PUBLIC	NO ADMIN	2
r1	r7	NO ADMIN	2
r2	r1	ADMIN	3
CHECKPOINT	r1	NO ADMIN	3
r3	r2	NO ADMIN	4
MONITOR	r2	NO ADMIN	4
r4	r3	ADMIN ONLY	5
BACKUP DATABASE	r3	ADMIN	5
DROP CONNECTION	r3	ADMIN ONLY	5

sp_displayroles('user2', 'expand_down', 'NO_ADMIN')を実行すると、次のような内容が出力されます。

role_name	parent_role_name	grant_type	role_level
r7	NULL	ADMIN	1
PUBLIC	NULL	NO ADMIN	1
dbo	PUBLIC	NO ADMIN	2
r1	r7	NO ADMIN	2
r2	r1	ADMIN	3
CHECKPOINT	r1	NO ADMIN	3
r3	r2	NO ADMIN	4
MONITOR	r2	NO ADMIN	4
BACKUP DATABASE	r3	ADMIN	5

sp_displayroles('r3', 'expand_up', 'NO_ADMIN')を実行すると、次のような内容が出力されます。

role_name	parent_role_name	grant_type	role_level
r1	r7	NO ADMIN	-2
r2	r1	ADMIN	-1
r3	r2	NO ADMIN	0

`sp_displayroles('r1', 'NO_ADMIN', 'expand_up')` を実行すると、次のような内容が出力されます。

role_name	parent_role_name	grant_type	role_level
r1	r7	NO ADMIN	0

sp_drop_secure_feature_key システムプロシージャ

セキュリティ機能キーを削除します。

構文

```
sp_drop_secure_feature_key ( name )
```

パラメータ

- **name** – 削除するセキュリティ機能キーの VARCHAR (128) 名。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、データベースサーバ所有者であり、その接続に対する `manage_keys` 機能が有効である必要があります。

備考

指定したキーが存在しない場合は、エラーが返されます。指定したキーが存在する場合、それがセキュリティ機能とセキュリティ機能キーの管理が可能な最後のセキュリティ機能キーでないかぎり、削除されます。たとえば、システムセキュリティ機能キーは、`manage_features` と `manage_keys` の両方のセキュリティ機能が有効な別のキーが存在しないかぎり、削除できません。

sp_expireallpasswords システムプロシージャ

すべてのパスワードをただちに有効期限切れにします。

構文 1

```
call sp_expireallpasswords
```

構文 2

sp_expireallpasswords

権限

そのシステムプロシージャに対する EXECUTE 権限に加え、次のものがが必要です。
MANAGE ANY USER システム権限。

参照：

- sp_iqaddlogin プロシージャ (437 ページ)
- sp_iqcopyloginpolicy プロシージャ (469 ページ)
- sp_iqmodifylogin プロシージャ (535 ページ)
- sp_iqpassword プロシージャ (551 ページ)

sp_list_secure_feature_key システムプロシージャ

ディレクトリの内容に関する情報を返します。

構文

sp_list_secure_feature_keys ()

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、データベースサーバ所有者であり、その接続に対する manage_keys 機能が有効である必要があります。

備考

カラム名	データ型	説明
name	VARCHAR(128)	セキュリティ機能キーの名前。
features	LONG VARCHAR	セキュリティ機能キーによって有効化されるセキュリティ機能。

このプロシージャは、既存のセキュリティ機能キーの名前と、各キーによって有効にできるセキュリティ機能のセットを返します。

ユーザが manage_features と manage_keys のセキュリティ機能を有効にしている場合、このプロシージャはすべてのセキュリティ機能キーのリストを返します。

ユーザが manage_keys セキュリティ機能のみを有効にしている場合、このプロシージャは、現在のユーザが有効にしている機能と同じ機能またはそのサブセットを有効にするキーを返します。

sp_login_environment システムプロシージャ

ユーザがログインするときの接続オプションを設定します。

構文

```
sp_login_environment ( )
```

備考

sp_login_environment は、デフォルトで login_procedure データベースオプションによって呼び出されるプロシージャです。

このプロシージャを編集しないことをおすすめします。ログイン環境を変更するには、異なるプロシージャを指すよう login_procedure オプションを変更します。

権限

なし。

関連する動作

なし。

sp_objectpermission システムプロシージャ

指定されたロールまたはユーザ名に付与されているオブジェクト権限または指定されたオブジェクトまたは DB 領域に対して付与されているオブジェクト権限のレポートを生成します。

構文

```
sp_objectpermission ( [object_name], [object_owner], [object_type] )
```

パラメータ

- **object_name** – オブジェクト、DB 領域、ユーザ、またはロールの名前。指定しない場合、現在のユーザのオブジェクト権限がレポートされる。デフォルト値は NULL。
- **object_owner** – 指定されたオブジェクト名のオブジェクト所有者の名前。指定されたオブジェクト所有者が所有する指定されたオブジェクトのオブジェクト権限が表示される。別のユーザまたはロールが所有するオブジェクトのオブジェクト権限を取得するには、このパラメータを指定する必要がある。デフォルト値は NULL。
- **object_type** – 有効な値：
 - TABLE*
 - VIEW
 - MATERIALIZED VIEW

システムプロシージャ

- SEQUENCE
- PROCEDURE
- FUNCTION
- DBSPACE
- USER

注意： * カラムレベルのオブジェクト権限も表示される。

値が指定されていない場合、すべてのオブジェクトタイプの権限が返される。
デフォルト値は NULL。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。どのユーザも **sp_objectpermission** を実行して、自分自身に付与されているすべてのオブジェクト権限を取得することができます。また、オブジェクト所有者もこのプロシージャを実行して、自分の所有するオブジェクトのオブジェクト権限を取得することができます。次のオブジェクト権限を取得するには、追加のシステム権限が必要です。

- 他のユーザに付与されたオブジェクト権限や他のユーザが所有するオブジェクトに対して付与されたオブジェクト権限 – MANAGE ANY OBJECT PRIVILEGE システム権限も必要です。
- ロールが所有するオブジェクトに対して付与されたオブジェクト権限やロールに付与されたオブジェクト権限 – MANAGE ANY OBJECT PRIVILEGE システム権限を持っているか、そのロールのロール管理者であることも必要です。
- DB 領域のオブジェクト権限 – MANAGE ANY DBSPACE システム権限が必要です。

備考

カラム名	データ型	説明
grantor	char(128)	付与者のユーザ ID。
grantee	char(128)	被付与者のユーザ ID。
object_name	char(128)	オブジェクトの名前。
owner	char(128)	オブジェクト所有者の名前。
object_type	char(20)	オブジェクトのタイプ。
column_name	char(128)	カラム名。
permission	char(20)	権限の名前。
grantable	char(1)	権限が付与可能かどうか。

すべての引数が任意で、次のレポートを生成することができます。

- 入力オブジェクト (テーブル、ビュー、プロシージャ、関数、シーケンスなど) の場合、このプロシージャは、そのオブジェクトに対するさまざまなオブジェクト権限を持つすべてのロールとユーザのリストを表示します。
- 入力ロールまたはユーザの場合、このプロシージャは、ロールまたは入力に付与されているすべてのオブジェクト権限のリストを表示します。
sp_objectpermission を実行してユーザまたはロールのオブジェクト権限を表示する場合、ロールの付与によって継承されたオブジェクト権限も表示されます。
- 入力 DB 領域名の場合、このプロシージャは、指定された DB 領域に対する **CREATE** 権限を持つすべてのユーザまたはロールのリストを表示します。
- デフォルトでは、オブジェクトタイプは **NULL** であり、指定されたオブジェクト名に一致する既存のすべてのオブジェクトタイプに対するオブジェクト権限が表示されます。

例

次の **GRANT** 文が実行されているとします。

```
GRANT SERVER OPERATOR TO r4;
GRANT BACKUP DATABASE TO r3 WITH ADMIN OPTION;
GRANT DROP CONNECTION TO r3 WITH ADMIN ONLY OPTION;
GRANT MONITOR TO r2;GRANT CHECKPOINT TO r1;
GRANT ROLE r2 TO r1 WITH ADMIN OPTION;
GRANT ROLE r3 TO r2 WITH NO ADMIN OPTION;
GRANT ROLE r4 TO r3 WITH ADMIN ONLY OPTION;
```

次のオブジェクト権限を検討します。

- r5 はデータベース内のテーブル `test_tab` とプロシージャ `test_proc` を所有します。
- r5 に対する管理権限を持つ u5 が次の権限を付与します。
 - **GRANT SELECT ON r5.test_tab TO r2 WITH GRANT OPTION;**
 - **GRANT SELECT (c1), UPDATE (c1) ON r5.test_tab TO r6 WITH GRANT OPTION;**
 - **GRANT EXECUTE ON r5.test_proc TO r3;**
- r6 に対する管理権限を持つ u6 が次の権限を付与します。
 - **GRANT SELECT (c1), REFERENCES (c1) ON r5.test_tab TO r3;**

`sp_objectpermission('r1')` を実行すると、次のような内容が出力されません。

表 122 : sp_objectpermission('r1') の出力例

grantor	grantee	object_name
u5	r2	test_tab
u6	r3	test_tab
u6	r3	test_tab
u6	r3	test_proc

(続き) owner	object_type	grantor
r5	TABLE	u5
r5	COLUMN	u6
r5	COLUMN	u6
r5	PROCEDURE	u6

(続き) grantable	column_name	privilege
Y	NULL	SELECT
N	c1	SELECT
Y	c1	REFERENCES
N	NULL	EXECUTE

sp_objectpermission('test_tab', 'r5', 'table') を実行すると、次のような内容が出力されます。

表 123 : sp_objectpermission('test_tab', 'r5', 'table') の出力例

grantor	grantee	object_name
u5	r2	test_tab
u5	r6	test_tab
u5	r6	test_tab
u6	r3	test_tab

grantor	grantee	object_name
u6	r3	test_tab

(続き) owner	object_type	grantor
r5	TABLE	u5
r5	COLUMN	u5
r5	COLUMN	u5
r5	COLUMN	u6
r5	COLUMN	u6

(続き) column_name	privilege	grantable
NULL	SELECT	Y
c1	SELECT	Y
c1	UPDATE	Y
c1	SELECT	N
c1	REFERENCES	N

sp_proc_priv システムプロシージャ

ストアドプロシージャを実行したりプロシージャの権限チェックに合格するために最低限必要なシステム権限のレポートを生成します。

構文

```
sp_proc_priv ( [proc_name] )
```

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

カラム名	データ型	説明
proc_name	char(128)	ストアドプロシージャ名。

カラム名	データ型	説明
privilege	long varchar	権限チェックに合格するために必要な権限。

ストアードプロシージャに対して複数のシステム権限がカンマで区切って表示される場合は、そのいずれか 1 つがあればストアードプロシージャを実行できます。ストアードプロシージャに対して複数の行が表示される場合は、ストアードプロシージャを実行するには各行のシステム権限が必要になります。

このプロシージャは、ストアードプロシージャのシステム権限のうち、プロシージャの権限チェックに必ず合格するものだけをリストします。その他のシステム権限で、特定の条件のもとでは権限チェックに合格してプロシージャを実行できるものも場合によってはありますが、そのような権限はこのプロシージャではリストされません。

例

パラメータを指定せずに **sp_proc_priv** を呼び出すと、すべてのストアードプロシージャとその実行に必要なシステム権限が表示されます。実行にシステム権限を必要としないストアードプロシージャは表示されません。

sp_proc_priv () を実行すると、次のような内容が出力されます。

proc_name	privileges
sp_iqrowdensity	MONITOR, MANAGE ANY DBSPACE, CREATE ANY INDEX, ALTER ANY INDEX, CREATE ANY OBJECT, ALTER ANY OBJECT
sp_iqworkmon	MONITOR
sp_iqindexsize	MANAGE ANY DBSPACE, ALTER ANY INDEX, ALTER ANY OBJECT
sp_addlogin	MANAGE ANY USER
sp_iqemptyfile	BACKUP DATABASE, SERVER OPERATOR, ALTER DATABASE
sp_iqemptyfile	INSERT ANY TABLE, UPDATE ANY TABLE, DELETE ANY TABLE, ALTER ANY TABLE, LOAD ANY TABLE, TRUNCATE ANY TABLE, ALTER ANY OBJECT
...	...

プロシージャ名のパラメータを指定して **sp_proc_priv** を呼び出すと、そのプロシージャの実行に必要なシステム権限が返されます。システム権限が必要ない場合は、プロシージャに対して "No Privilege Required" (権限不要) と表示されます。

proc_name	privileges
sp_iqindexsize	MANAGE ANY DBSPACE, ALTER ANY INDEX

プロシージャが存在しない場合は、エラーメッセージが表示されます。

sp_remote_columns システムプロシージャ

リモートテーブルにあるカラムとそれらのデータ型の記述のリストを生成します。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 文で定義します。

構文

```
sp_remote_columns (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

引数

- **@server_name** – CREATE SERVER 文で指定されたサーバ名を含む文字列を指定する CHAR(128) パラメータ。
- **@table_name** – リモートテーブルの名前を指定する CHAR(128) パラメータ。
- **@table_owner** – @table_name の所有者を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@table_qualifier** – 任意の CHAR(128) パラメータを使用して、@table_name が格納されているデータベースの名前を指定します。デフォルトでは "%" です。

結果セット

カラム名	データ型	説明
database	CHAR(128)	データベース名。
owner	CHAR(128)	データベース所有者名。
table_name	CHAR(128)	テーブル名。
column_name	CHAR(128)	カラムの名前。
domain_id	SMALLINT	カラムのデータ型を示す INTEGER。
width	INTEGER	このカラムの意味は、データ型によって異なります。character 型の場合、width は文字数を表します。
scale	SMALLINT	このカラムの意味は、データ型によって異なります。NUMERIC データ型の場合、scale とは小数点以下の桁数です。

カラム名	データ型	説明
nullable	SMALLINT	NULL カラム値が許可される場合、値は1です。それ以外の場合、値は0です。
base_type_str	CHAR(4096)	カラムの物理的な型を表す注釈付きの型文字列。

備考

CREATE EXISTING TABLE 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ち場合があります。sp_remote_columns はリモートテーブルのカラムとそのデータ型の記述のリストを生成します。データベースを指定する場合は、所有者または値 NULL を指定してください。

権限

なし。

関連する動作

なし。

標準と互換性

- **Sybase – Open Client/Open Server** でサポートされています。

例

次の例は、リモート SAP Sybase IQ データベースサーバ RemoteSA の ULProduct テーブルのカラムに関する情報を返します。テーブルの所有者は DBA です。

```
CALL sp_remote_columns( 'RemoteSA', 'ULProduct', 'DBA', null );
```

次の例は、リモートサーバ RemoteASE を使用する Adaptive Server Enterprise データベースの SYSOBJECTS テーブルのカラムに関する情報を返します。テーブルの所有者は指定されていません。

```
CALL sp_remote_columns( 'RemoteASE', 'sysobjects', null, 'Production' );
```

次の例は、リモートサーバ MyAccessDB を使用する Microsoft Access データベース c:\¥users¥me¥documents¥MyAccessDB.accdb の Customers テーブルのカラムに関する情報を返します。Access データベースにはテーブル所有者がないため、NULL が指定されます。

```
CALL sp_remote_columns( 'MyAccessDB', 'Customers', null, 'c:\¥users¥me¥documents¥MyAccessDB.accdb' );
```


sp_remote_exported_keys システムプロシージャ

指定されたプライマリテーブルに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 文で定義します。

構文

```
sp_remote_exported_keys (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

引数

- **@server_name** – プライマリテーブルが格納されているサーバを指定する CHAR(128) パラメータ。
- **@table_name** – プライマリキーを格納するテーブルを指定する CHAR(128) パラメータ。
- **@table_owner** – プライマリテーブルの所有者を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@table_qualifier** – プライマリテーブルを格納するデータベースを指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。

結果セット

カラム名	データ型	説明
pk_database	CHAR(128)	プライマリキーテーブルがあるデータベース。
pk_owner	CHAR(128)	プライマリキーテーブルの所有者。
pk_table	CHAR(128)	プライマリキーテーブル。
pk_column	CHAR(128)	プライマリキーカラムの名前。
fk_database	CHAR(128)	外部キーテーブルがあるデータベース。
fk_owner	CHAR(128)	外部キーテーブルの所有者。
fk_table	CHAR(128)	外部キーテーブル。
fk_column	CHAR(128)	外部キーカラムの名前。
key_seq	SMALLINT	キーのシーケンス番号。

カラム名	データ型	説明
fk_name	CHAR(128)	外部キーの名前。
pk_name	CHAR(128)	プライマリキーの名前。

備考

このプロシージャは、特定のプライマリテーブルに外部キーを持つリモートテーブルに関する情報を提供します。sp_remote_exported_keys システムプロシージャの結果セットには、データベース、所有者、テーブル、カラム、プライマリキーと外部キーの両方の名前、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 と JDBC 呼び出しのために結果セットが変わる場合もありますが、外部キーのテーブルとカラムに関する情報は常に返されます。

権限

なし。

関連する動作

なし。

例

この例は、リモートサーバ RemoteSA 上の ULEmployee テーブルの外部キーの関係に関する情報を返します。

```
CALL sp_remote_exported_keys( 'RemoteSA', 'ULEmployee', 'DBA' );
```

sp_remote_imported_keys システムプロシージャ

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提供します。

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 文で定義します。

構文

```
sp_remote_imported_keys (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

引数

- **@server_name** – この CHAR(128) パラメータを使用して、外部キーテーブルが格納されているサーバを指定します。このパラメータの値は必須です。

- **@table_name** – 外部キーを格納するテーブルを指定する CHAR(128) パラメータ。このパラメータの値は必須です。
- **@table_owner** – 外部キーテーブルの所有者を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@table_qualifier** – 外部キーテーブルを含むデータベースを指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。

結果セット

カラム名	データ型	説明
pk_database	CHAR(128)	プライマリキーテーブルがあるデータベース。
pk_owner	CHAR(128)	プライマリキーテーブルの所有者。
pk_table	CHAR(128)	プライマリキーテーブル。
pk_column	CHAR(128)	プライマリキーカラムの名前。
fk_database	CHAR(128)	外部キーテーブルがあるデータベース。
fk_owner	CHAR(128)	外部キーテーブルの所有者。
fk_table	CHAR(128)	外部キーテーブル。
fk_column	CHAR(128)	外部キーカラムの名前。
key_seq	SMALLINT	キーのシーケンス番号。
fk_name	CHAR(128)	外部キーの名前。
pk_name	CHAR(128)	プライマリキーの名前。

備考

外部キーは、対応するプライマリキーを持つ別のテーブル内のローを参照します。このプロシージャを使用すると、特定の外部テーブルに対応するプライマリキーを持つ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、プライマリキーと外部キーの両方の名前、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 と JDBC 呼び出しのために結果セットが変わる場合もありますが、プライマリキーのテーブルとカラムに関する情報は常に返されます。

権限

なし。

関連する動作

なし。

例

次の例は、リモートサーバ RemoteSA 上の ULOrder テーブルの外部キーに対応するテーブルとプライマリキーを返します。

```
CALL sp_remote_imported_keys( 'RemoteSA', 'ULOrder', 'DBA' );
```

sp_remote_primary_keys システムプロシージャ

リモートデータアクセスを使用してリモートテーブルについてのプライマリキー情報を提供します。

構文

```
sp_remote_primary_keys (
  @server_name
  , @table_name
  [, @table_owner
  [, @table_qualifier ] ]
)
```

引数

- **@server_name** – この CHAR(128) パラメータを使用して、リモートサーバ名を指定します。
- **@table_name** – この CHAR(128) パラメータを使用して、リモートテーブルの名前を指定します。
- **@table_owner** – リモートテーブルの所有者を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@table_qualifier** – リモートデータベースの名前を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。

結果セット

カラム名	データ型	説明
database	CHAR(128)	リモートデータベースの名前。
owner	CHAR(128)	テーブルの所有者。
table_name	CHAR(128)	テーブル名。
column_name	CHAR(128)	プライマリキーカラムの名前。
key_seq	SMALLINT	プライマリキーのシーケンス番号。
pk_name	CHAR(128)	プライマリキーの名前。

備考

このシステムプロシージャは、リモートデータアクセスを使用してリモートテーブルについてのプライマリキー情報を提供します。

基本となる ODBC 呼び出しに違いがあるため、返される情報は、サーバに指定されたリモートデータアクセスクラスによって、カタログの値やデータベースの値と多少異なります。

権限

なし。

標準と互換性

- **Sybase – Open Client/Open Server** でサポートされています。

関連する動作

なし

例

次の例は、SAP Sybase IQ リモートサーバ RemoteSA の DBA が所有するテーブルのプライマリキーに関する情報を返します。

```
CALL sp_remote_primary_keys( 'RemoteSA', null, 'DBA' );
```

Adaptive Server Enterprise サーバ RemoteASE の production データベースにある Fred が所有するすべてのテーブルのプライマリキーのリストを取得します。

```
CALL sp_remote_primary_keys( 'RemoteASE', null, 'Fred',  
'production' );
```

sp_remote_tables システムプロシージャ

サーバ上のテーブルのリストを返します。

構文

```
sp_remote_tables(  
  @server_name  
  [, @table_name  
  [, @table_owner  
  [, @table_qualifier  
  [, @with_table_type ] ] ]  
)
```

引数

- **@server_name** – この CHAR(128) パラメータを使用して、リモートサーバ名を指定します。

- **@table_name** – リモートテーブルの名前を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@table_owner** – リモートテーブルの所有者を指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@table_qualifier** – **table_name** が格納されているデータベースを指定する任意の CHAR(128) パラメータ。デフォルトでは "%" です。
- **@with_table_type** – リモートテーブルのタイプを含めるよう指定する任意の BIT パラメータ。デフォルトは 0 です。結果セットに、テーブルのタイプを一覧するカラムを結果セットに含める場合は 1 を、そうでない場合は 0 を指定します。

結果セット

カラム名	データ型	説明
database	CHAR(128)	リモートデータベースの名前。
owner	CHAR(128)	テーブル所有者の名前。
table_name	CHAR(128)	テーブル名。
table_type	CHAR(128)	テーブルタイプを指定します。値は、リモートサーバのタイプによって異なります。たとえば、指定できる値として TABLE、VIEW、SYS、GBL TEMP があります。

備考

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 文で定義します。

データベースサーバを設定するときに、特定のサーバ上で使用可能なリモートテーブルのリストを取得しておく役立ち場合があります。このプロシージャは、サーバ上のテーブルのリストを返します。

このプロシージャには 5 つのパラメータを指定できます。テーブル、所有者、またはデータベース名を指定すると、テーブルのリストはその引数に当てはまるものだけに限定されます。

権限

なし。

関連する動作

なし。

標準と互換性

- **Sybase – Open Client/Open Server** でサポートされています。

例

次の例は、RemoteSA という名前の SAP Sybase IQ リモートサーバの DBA が所有するテーブルに関する情報を返します。

```
CALL sp_remote_tables( 'RemoteSA', null, 'DBA' );
```

Adaptive Server Enterprise サーバ RemoteASE の production データベースにある Fed が所有するすべてのテーブルのリストを取得します。

```
CALL sp_remote_tables( 'RemoteASE', null, 'Fred', 'production' );
```

サーバ RemoteExcel で参照されている 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します。

```
CALL sp_remote_tables( 'RemoteExcel' );
```

sp_servercaps システムプロシージャ

リモートサーバの機能についての情報を表示します。

構文

```
sp_servercaps( @server_name )
```

引数

- **@server_name** – CREATE SERVER 文で定義されたサーバを指定する CHAR(128) パラメータ。@server_name は、CREATE SERVER 文で使用されるサーバと同じサーバ名です。

結果

カラム	データ型	説明
capid	INTEGER	機能識別子。
capname	CHAR(128)	機能の名前。
capvalue	CHAR(128)	機能の設定。通常は T (true) または F (false)。

備考

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 文で定義します。

このプロシージャは、リモートサーバの機能についての情報を表示します。この機能の情報は、どのくらいの SQL 文をリモートサーバに転送できるかを判断するために使用されます。サーバの機能をリストする ISYSCAPABILITY システムテーブルは、最初のリモートサーバに接続するまでは設定されません。

標準と互換性

- **Sybase** – Open Client/Open Server でサポートされています。

権限

なし。

関連する動作

なし。

例

リモートサーバ RemoteSA に関する情報を表示します。

```
CALL sp_servercaps( 'RemoteSA' );
```

sp_sys_priv_role_info システム権限

システム権限を対応するシステムロールとマッピングするレポートを生成します。システム権限ごとに 1 つのローが返されます。

構文

sp_sys_priv_role_info()

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

カラム名	データ型	説明
sys_priv_name	char(128)	システム権限の名前。
sys_priv_role_name	char(128)	システム権限に対応するロール名。
sys_priv_id	unsigned int	システム権限の ID。

sp_tsql_environment システムプロシージャ

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の接続オプションを設定します。

構文

```
sp_tsql_environment( )
```

備考

sp_login_environment プロシージャは、login_procedure データベースオプションによって指定されるデフォルトのプロシージャです。新規接続ごとに、

login_procedure で指定されたプロシージャが呼び出されます。接続に TDS 通信プロトコルを使用する場合 (つまり、Open Client または jConnect 接続の場合)、今度 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 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。

新しいプロシージャを作成し login_procedure オプションを変更してそれらの新しいプロシージャをポイントして、デフォルトの動作を変更します。

次は、sp_tsql_environment プロシージャが設定したオプションのリストです。

```
if db_property( 'IQStore' ) = 'Off' then
    -- SAP Sybase IQ 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';
```

権限
なし。

関連する動作
なし。

例

次は、sp_tsql_environment プロシージャを呼び出す例です。

```
CALL sp_tsql_environment();
```

sp_use_secure_feature_key システムプロシージャ

既存のセキュリティ機能キーを有効にします。

構文

```
sp_use_secure_feature_key ( name, sfkey)
```

パラメータ

- **name** – 有効にするセキュリティ機能キーの VARCHAR (128) 名。
- **sfkey** – 有効にするセキュリティ機能キーの CHAR (128) 認証キー。認証キーは 6 文字以上である必要があります。

権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

備考

このプロシージャは、指定されたセキュリティ機能キーで有効になるセキュリティ機能を有効にします。

Adaptive Server のシステムプロシージャおよびカタログプロシージャ

Adaptive Server には、システムプロシージャとカタログプロシージャがあります。これらは、多くの管理用の関数を実行し、システム情報を取得します。SAP Sybase IQ は、これらのプロシージャのいくつかをサポートします。

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ードプロシージャです。カタログストアードプロシージャは、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 と同様の機能を実行しますが、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われている場合、動作をチェックしておいたほうがいいでしょう。ストアー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'
```

SAP Sybase が提供するすべてのシステムストアー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ー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 のシステムプロシージャ

SAP Sybase IQ で提供されている Adaptive Server のシステムプロシージャは次のとおりです。

表 124 : SAP Sybase IQ で提供されている ASE システムプロシージャ

システムプロシージャ	説明	パーミッション
sp_addlogin <i>userid, password[, defdb [, deflanguage [, fullname]]]</i>	データベースに新規ユーザアカウントを追加する。	MANAGE ANY USER システム権限が必要。
sp_addmessage <i>message-number, message_text [, language]</i>	ストアードプロシージャの PRINT 呼び出しと RAISERROR 呼び出しに使用するユーザ定義メッセージを SYSUSERMESSAGES に追加する。	CREATE MESSAGE システム権限または CREATE ANY OBJECT システム権限が必要。
sp_addtype <i>typename, datatype, [, "identity"/ nulltype]</i>	ユーザ定義データ型を作成する。SAP Sybase IQ では IDENTITY カラムはサポートしていない。	CREATE DATATYPE システム権限または CREATE ANY OBJECT システム権限が必要。
sp_adduser <i>userid [, name_in_db [, grpname]]</i>	データベースに新規ユーザを追加する。	新しいユーザを作成するには、MANAGE ANY USER システム権限が必要。新しいユーザを作成し、指定したロールにそのユーザを追加するには、MANAGE ANY USER システム権限と MANAGE ROLES システム権限が必要。
sp_dboption <i>[dbname, option-name, {true false}]</i>	データベースオプションを表示または変更する。	不要。
sp_droplogin <i>userid</i>	データベースからユーザを削除する。	MANAGE ANY LOGIN POLICY システム権限が必要。
sp_dropmessage <i>message-number [, language]</i>	ユーザ定義メッセージを削除する。	DROP MESSAGE システム権限が必要。
sp_droptype <i>typename</i>	ユーザ定義データ型を削除する。	DROP DATATYPE システム権限が必要。
sp_dropuser <i>userid</i>	データベースからユーザを削除する。	MANAGE ANY USER システム権限が必要。

システムプロシージャ	説明	パーミッション
<code>sp_getmessage</code> <i>message-num, @msg-var output [, language]</i>	PRINT 文と RAISERROR 文で使用するために、SYSUSERMESSAGES 内に格納されているメッセージ文字列を取得する。	不要。
<code>sp_helptext</code> 'owner.object-name'	システムプロシージャまたはビューのテキストを表示する。	不要。
<code>sp_password</code> <i>caller_passwd, new_passwd [, userid]</i>	ユーザ ID のパスワードを追加または変更する。	ユーザ所有のパスワードを変更するには、システム権限は不要。別のユーザのパスワードを変更するには、 CHANGE PASSWORD システム権限が必要。

注意： `sp_dropuser` などのプロシージャには、Adaptive Server のストアードプロシージャとの互換性はほとんどありません。Adaptive Server を使い慣れている場合は、それらのテキストと SAP Sybase IQ のプロシージャとを比較してから、Interactive SQL 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ードプロシージャの場合、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ー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'
```

参照：

- ユーザ、グループ/ロール、およびパーミッション (886 ページ)

Adaptive Server のカタログプロシージャ

SAP Sybase IQ には、`sp_column_privileges` プロシージャ以外の Adaptive Server のカタログプロシージャのほとんどが実装されています。

また、SAP 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ードプロシージャが用意されています。

表 125 : SAP 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
sp_columns <i>table-name</i> [, <i>table-owner</i>] [, <i>table-qualifier</i>] [, <i>column-name</i>]	指定したカラムのデータ型を返す。	
sp_fkeys <i>pktable_name</i> [, <i>pktable-owner</i>] [, <i>pktable-qualifier</i>] [, <i>fktable-name</i>] [, <i>fktable_owner</i>] [, <i>fktable-qualifier</i>]	指定したテーブルの外部キー情報を返す。	
sp_pkeys <i>table-name</i> [, <i>table_owner</i>] [, <i>table-qualifier</i>]	指定したテーブルのプライマリキー情報を返す。	sp_iqpkeys
sp_special_columns <i>table_name</i> [, <i>table-owner</i>] [, <i>table-qualifier</i>] [, <i>col-type</i>]	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す。	
sp_sproc_columns <i>proc-name</i> [, <i>proc_owner</i>] [, <i>proc-qualifier</i>] [, <i>column-name</i>]	ストアー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す。	sp_iqprocparm
sp_stored_procedures [<i>sp-name</i>] [, <i>sp-owner</i>] [, <i>sp-qualifier</i>]	1つ以上のストアードプロシージャの情報を返す。	sp_iqprocedure
sp_tables <i>table-name</i> [, <i>table-owner</i>] [, <i>table-qualifier</i>] [, <i>table-type</i>]	FROM 句に表示できるオブジェクトのリストを返す。	

次の Adaptive Server 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

システムテーブルとシステムビュー

SAP Sybase IQ は、システムテーブル、システムビュー、統合ビュー、互換ビュー、および Adaptive Server T-SQL 互換ビューをサポートします。

システムテーブル

すべての SAP Sybase IQ データベースの構造は、多くのシステムテーブルに記述されています。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。

DUMMY システムテーブルは、ユーザが直接アクセスできる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。

表 126 : システムテーブルのリスト

システムテーブル	内部使用のみ
DUMMY	いいえ
ISYSARTICLE	はい
ISYSARTICLECOL	はい
ISYSATTRIBUTE	はい
ISYSATTRIBUTENAME	はい
ISYSCAPABILITY	はい
ISYSCHECK	はい
ISYSCOLPERM	はい
ISYSCOLSTAT	はい
ISYSCONSTRAINT	はい
ISYSDBFILE	はい
ISYSDBSPACE	はい
ISYSDBSPACEPERM	はい
ISYSDEPENDENCY	はい
ISYSDOMAIN	はい
ISYSEVENT	はい

システムテーブルとシステムビュー

システムテーブル	内部使用のみ
ISYSEXTERNENV	はい
ISYSEXTERNENVOBJECT	はい
ISYSEXTERNLOGIN	はい
ISYSFKEY	はい
ISYSGROUP	はい
ISYSHISTORY	はい
ISYSIDX	はい
ISYSIDXCOL	はい
ISYSIQBACKUPHISTORY	はい
ISYSIQBACKUPHISTORYDETAIL	はい
ISYSIQDBFILE	はい
ISYSIQDBSPACE	はい
ISYSIQIDX	はい
ISYSIQINFO	はい
ISYSIQLOGICALSERVER	はい
ISYSIQLOGINPOLICYLSINFO	はい
ISYSIQLSLOGINPOLICYOPTION	はい
ISYSIQLSMEMBER	はい
ISYSIQLSPOLICY	はい
ISYSIQLSPOLICYOPTION	はい
ISYSIQMPXSERVER	はい
ISYSIQMPXSERVERAGENT	はい
ISYSIQPARTITIONCOLUMN	はい
ISYSIQTAB	はい
ISYSIQTABCOL	はい
ISYSJAR	はい
ISYSJARCOMPONENT	はい
ISYSJAVACLASS	はい

システムテーブル	内部使用のみ
ISYSLOGINMAP	はい
ISYSLOGINPOLICY	はい
ISYSLOGINPOLICYOPTION	はい
ISYSMVOPTION	はい
ISYSMVOPTIONNAME	はい
ISYSOBJECT	はい
ISYSOPTION	はい
ISYSOPTSTAT	はい
ISYSPARTITION	はい
ISYSPARTITIONKEY	はい
ISYSPARTITIONSCHEME	はい
ISYSPHYSIDX	はい
ISYSPROCEDURE	はい
ISYSPROCPARM	はい
ISYSPROCPERM	はい
ISYSPROXYTAB	はい
ISYSPUBLICATION	はい
ISYSREMARK	はい
ISYSREMOTEOPTION	はい
ISYSREMOTEOPTIONTYPE	はい
ISYSREMOTETYPE	はい
ISYSREMOTEUSER	はい
ISYSSCHEDULE	はい
ISYSSERVER	はい
ISYSSOURCE	はい
ISYSSQLSERVERTYPE	はい
ISYSSUBPARTITIONKEY	はい
ISYSSUBSCRIPTION	はい

システムテーブル	内部使用のみ
ISYSSYNC	はい
ISYSSYNCPROFILE	はい
ISYSSYNCSRIPT	はい
ISYSTAB	はい
ISYSTABCOL	はい
ISYSTABLEPERM	はい
ISYSTEEXTCONFIG	はい
ISYSTEEXTIDX	はい
ISYSTEEXTIDXTAB	はい
ISYSTRIGGER	はい
ISYSTYPEMAP	はい
ISYSUSER	はい
ISYSUSERAUTHORITY	はい
ISYSUSERMESSAGE	はい
ISYSUSERTYPE	はい
ISYSVIEW	はい
ISYSWEBSERVICE	はい

SYS.DUMMY テーブルと IQ_DUMMY テーブルの比較

DUMMY システムテーブルは、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 テーブルを使用したクエリを実行するのは、SAP Sybase IQ ではなく SQL Anywhere (カタログストア) です。SAP Sybase IQ データベース内で、ダミーテーブルを作成できます。次に例を示します。

```
CREATE TABLE iq_dummy (dummy_col INT NOT NULL);
```

さらに、このテーブルを明示的に使用します。

```
SELECT NOW() FROM iq_dummy;
```

DUMMY システムテーブル

カラム名	カラム型	カラム制約	テーブル制約
dummy_col	INTEGER	NOT NULL	

DUMMY テーブルは、常に 1 つだけのローを持つ、読み込み専用の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 IQ.DUMMY;
```

FROM 句での IQ.DUMMY の使用はオプションです。FROM 句でテーブルが指定されていない場合、テーブルは IQ.DUMMY テーブルと見なされます。上記の例は、次のように記述できます。

```
SELECT USER, today(*);
```

dummy_col – このカラムは使用されません。テーブルはカラムなしでは作成できないので、このカラムが存在します。

IQ.DUMMY テーブルからの読み取りコストは、同様のユーザ作成テーブルからの読み取りコストよりも低いです。これは、IQ.DUMMY のテーブルページにラッチがないためです。

アクセスプランは、IQ.DUMMY テーブルのスキャンによって構築されるわけではありません。そうではなく、IQ.DUMMY への参照がローコンストラクタアルゴリズムに置き換えられ、これがテーブル参照を仮想化します。これにより、IQ.DUMMY の使用に伴う競合を排除できます。DUMMY は、テーブル名か関連名またはその両方として、短いプラン、長いプラン、グラフィカルなプランに引き続き表示されます。

システムビュー

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。

統合ビュー

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

システムテーブルとシステムビュー

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。

互換ビュー

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 SAP Sybase IQ との互換性を維持する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してください。互換ビューは、SAP Sybase IQ の将来のバージョンではサポートされなくなる可能性があります。

システムビューのアルファベット順リスト

システムテーブルは非表示です。ただし、各テーブルにはシステムビューがあります。将来的な IQ メインストアとの互換性を保つために、アプリケーションではシステムビューを使用し、変更の可能性がある基本となるシステムテーブルは使用しないでください。

ASE T-SQL 互換ビュー

SAP 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 のシステムテーブルとビューに対応しています。

参照：

- Transact-SQL 互換ビュー (863 ページ)

ST_GEOMETRY_COLUMNS 統合ビュー

ST_GEOMETRY_COLUMNS システムビューの各ローは、データベースに定義されている空間カラムに関する記述です。

注意： 空間データ、空間参照系、および空間測定単位は、カタログストアでのみ使用できます。

カラム名	データ型	説明
table_catalog	VARCHAR(128)	内部用。
table_schema	CHAR(128)	空間カラムを含むテーブルが属するスキーマの名前。これはテーブル所有者と同じ。
table_name	CHAR(128)	空間カラムを含むテーブルの名前。
column_name	CHAR(128)	空間カラムの名前。

カラム名	データ型	説明
srs_name	CHAR(128)	空間カラムに関連付けられている SRS の名前。 SRS がカラムに関連付けられていない場合、 srs_name は NULL。
srs_id	INTEGER	空間カラムに関連付けられている SRS の SRID。
table_id	UNSIGNED INT	カラムが属するテーブルの数値識別子。
column_id	UNSIGNED INT	カラムの数値識別子。
geometry_type_name	VARCHAR(32767)	カラムに含まれているジオメトリの空間データ型 (ST_Point、ST_Geometry など)。

ST_SPATIAL_REFERENCE_SYSTEMS 統合ビュー

ST_SPATIAL_REFERENCE_SYSTEMS システムビューの各ローは、データベースに定義されている SRS に関する記述です。このビューには、SYSSPATIALREFERENCESYSTEM システムビューとは若干異なる量の情報があります。

注意： 空間データ、空間参照系、および空間測定単位は、カタログストアでのみ使用できます。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	システムでのみ使用。
owner	UNSIGNED INT	SRS の所有者。
srs_name	CHAR(128)	SRS の名前。
srs_id	INTEGER	空間参照系の数値識別子 (SRID)。

カラム名	データ型	説明
srs_type	CHAR(11)	<p>SQL/MM 標準による定義に従った SRS のタイプ。次のいずれかの値を取る。</p> <ul style="list-style-type: none"> • GEOGRAPHIC – 緯度と経度 (および標高) の軸を持つ、測地参照される座標系に基づく SRS 用。これらの SRS は、PLANAR タイプまたは ROUND EARTH タイプ。 • PROJECTED – 緯度と経度の軸を持たず、測地参照される座標系に基づく SRS 用。これらの SRS は PLANAR タイプ。 • ENGINEERING – 測地参照されない座標系に基づく SRS 用。これらの SRS は PLANAR タイプ。 • GEOCENTRIC – サポートされない。 • COMPOUND – サポートされない。 • VERTICAL – サポートされない。 <p>srs_type が空の場合、タイプは指定されない。</p>
round_earth	CHAR(1)	SRS タイプが ROUND EARTH (Y) または PLANAR (N) のいずれであるか。
axis_order	CHAR(12)	データベースサーバが緯度と経度に関連してポイントを解釈する方法を記述 (たとえば、ST_Lat メソッドと ST_Long メソッドを使用する場合)。非地理的空間参照系の場合、軸順序は x/y/z/m。地理的空間参照系の場合、デフォルトの軸順序は long/lat/z/m であり、lat/long/z/m もサポートされる。
snap_to_grid	DOUBLE	計算の実行時に使用されるグリッドのサイズを定義。
tolerance	DOUBLE	ポイントを比較するときに使用する精度を定義。
semi_major_axis	DOUBLE	ROUND EARTH SRS の楕円の中心から赤道までの距離。
semi_minor_axis	DOUBLE	ROUND EARTH SRS の楕円の中心から両極までの距離。

カラム名	データ型	説明
inv_flattening	DOUBLE	ROUND EARTH SRS の楕円に使用される逆扁平率。これは、次の式によって算出される比率となる。 $1/f = (\text{semi-major-axis}) / (\text{semi-major-axis} - \text{semi-minor-axis})$
min_x	DOUBLE	座標において許される x の最小値。
max_x	DOUBLE	座標において許される x の最大値。
min_y	DOUBLE	座標において許される y の最小値。
max_y	DOUBLE	座標において許される y の最大値。
min_z	DOUBLE	座標において許される z の最小値。
max_z	DOUBLE	座標において許される z の最大値。
min_m	DOUBLE	座標において許される m の最小値。
max_m	DOUBLE	座標において許される m の最大値。
min_lat	DOUBLE	座標において許される緯度の最小値。
max_lat	DOUBLE	座標において許される緯度の最大値。
min_long	DOUBLE	座標において許される経度の最小値。
max_long	DOUBLE	座標において許される経度の最大値。
organization	LONG VARCHAR	空間参照系によって使用される座標系を作成した組織の名前。
organization_coord- sys_id	INTEGER	座標系を作成した組織が座標系に対して指定した ID。
linear_unit_of_ measure	CHAR(128)	SRS によって使用される線形測定単位。
angular_unit_of_ measure	CHAR(128)	SRS によって使用される角度測定単位。
polygon_format	LONG VARCHAR	多角形のリングの方向。CounterClockwise、ClockWise、または EvenOdd のいずれか。
storage_format	LONG VARCHAR	データが正規化フォーマット (Internal)、非正規化フォーマット (Original)、または両方 (Mixed) のいずれかで格納されるか。

カラム名	データ型	説明
definition	LONG VARCHAR	追加の定義設定。
transform_definition	LONG VARCHAR	この SRS から別の SRS にデータを変換するときに使用される変換定義設定。
description	LONG VARCHAR	SRS の説明。

ST_UNITS_OF_MEASURE 統合ビュー

ST_UNITS_OF_MEASURE システムビューの各ローは、データベースに定義されている測定単位に関する記述です。このビューには、SYSUNITOFMEASURE システムビューよりも多い情報があります。

注意： 空間データ、空間参照系、および空間測定単位は、カタログストアでのみ使用できます。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	システムでのみ使用。
owner	UNSIGNED INT	測定単位の所有者。
unit_name	CHAR(128)	測定単位の名前。
unit_type	CHAR(7)	角度または線形。
conversion_factor	DOUBLE	測定単位の換算係数。
description	LONG VARCHAR	測定単位の説明。

SYSARTICLE システムビュー

SYSARTICLE システムの各ローは、パブリケーション内のアーティクルを示します。このビューの基本となるシステムテーブルは ISYSARTICLE です。

カラム名	データ型	説明
publication_id	UNSIGNED INT	このアーティクルが含まれるパブリケーション。
table_id	UNSIGNED INT	各アーティクルは、単一のテーブルのカラムとローから構成されます。このカラムには、このテーブルの ID が含まれます。
where_expr	LONG VARCHAR	WHERE 句で定義されたローのサブセットを含むアーティクルの場合、このカラムに探索条件が含まれます。

カラム名	データ型	説明
subscribe_by_expr	LONG VARCHAR	SUBSCRIBE BY 式で定義されたローのサブセットを含むアーティクルの場合、このカラムに式が含まれます。
query	CHAR(1)	データベースサーバにアーティクルタイプ情報を示します。
alias	VARCHAR(256)	アーティクルのエイリアス。
schema_change_active	BIT	テーブルとパブリケーションが同期スキーマ変更の一部である場合、1 です。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (publication_id, table_id)
```

```
FOREIGN KEY (publication_id) REFERENCES SYS.ISYSPUBLICATION (publication_id)
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

参照：

- SYSARTICLECOL システムビュー (755 ページ)
- SYSARTICLECOLS 統合ビュー (756 ページ)

SYSARTICLECOL システムビュー

SYSARTICLECOL システムビューの各ローは、アーティクル内のカラムを識別します。このビューの基本となるシステムテーブルは ISYSARTICLECOL です。

カラム名	データ型	説明
publication_id	UNSIGNED INT	カラムが含まれるパブリケーションのユニークな識別子。
table_id	UNSIGNED INT	カラムが属するテーブル。
column_id	UNSIGNED INT	SYSTABCOL システムビューのカラム識別子。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (publication_id, table_id, column_id)
```

```
FOREIGN KEY (publication_id, table_id) REFERENCES SYS.ISYSARTICLE (publication_id, table_id)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL (table_id, column_id)
```

SYSARTICLECOLS 統合ビュー

SYSARTICLECOLS ビューの各ローは、アーティクル内のカラムを識別します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSARTICLECOLS"
  as select p.publication_name,t.table_name,c.column_name
  from SYS.ISYSARTICLECOL as ac
  join SYS.ISYSPUBLICATION as p on p.publication_id =
ac.publication_id
  join SYS.ISYSTAB as t on t.table_id = ac.table_id
  join SYS.ISYSTABCOL as c on c.table_id = ac.table_id
  and c.column_id = ac.column_id
```

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アーティクルを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSARTICLES"
  as select u1.user_name as publication_owner,p.publication_name,
  u2.user_name as table_owner,t.table_name,
  a.where_expr,a.subscribe_by_expr,a.alias
  from SYS.ISYSARTICLE as a
  join SYS.ISYSPUBLICATION as p on(a.publication_id =
p.publication_id)
  join SYS.ISYSTAB as t on(a.table_id = t.table_id)
  join SYS.ISYSUSER as u1 on(p.creator = u1.user_id)
  join SYS.ISYSUSER as u2 on(t.creator = u2.user_id)
```

SYSCAPABILITIES 統合ビュー

SYSCAPABILITIES ビューの各ローは、リモートデータベースサーバの機能のステータスを示します。このビューは、ISYSCAPABILITY と ISYSCAPABILITYNAME の各システムテーブルからデータを取得します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSCAPABILITIES"
  as select
  ISYSCAPABILITY.capid, ISYSCAPABILITY.srvid,property('RemoteCapabilit
y', ISYSCAPABILITY.capid) as capname, ISYSCAPABILITY.capvalue
  from SYS.ISYSCAPABILITY
```

SYSCAPABILITY システムビュー

SYSCAPABILITY システムビューの各ローは、リモートデータベースサーバの機能のステータスを示します。このビューの基本となるシステムテーブルは ISYSCAPABILITY です。

カラム名	データ型	説明
capid	INTEGER	SYSCAPABILITYNAME システムビューに表示されている機能 ID。
srvid	UNSIGNED INT	SYSSERVER システムビューに表示されている、機能が適用されるサーバ。
capvalue	CHAR(128)	機能の値。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (catid, srvid)
```

```
FOREIGN KEY (srvid) REFERENCES SYS.ISYSSERVER (srvid)
```

SYSCAPABILITYNAME システムビュー

SYSCAPABILITYNAME システムビューの各ローには、SYSCAPABILITY システムビュー内の各機能 ID の名前があります。

カラム名	データ型	説明
catid	INTEGER	機能をユニークに識別する番号。
capname	VARCHAR(32000)	機能の名前。

備考

SYSCAPABILITYNAME システムビューは、sa_rowgenerator と次のサーバプロパティの組み合わせを使用して定義されます。

```
RemoteCapability
MaxRemoteCapability
```

SYSCATALOG 統合ビュー

SYSCATALOG ビューの各ローは、システムテーブルを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSCATALOG" ( creator,
    tname, dbspacename, tabletype, ncols, primary_key, "check",
    remarks )
as select u.user_name, tab.table_name, dbs.dbSPACE_name,
```

システムテーブルとシステムビュー

```
if tab.table_type_str = 'BASE' then 'TABLE' else
tab.table_type_str endif,
(select count() from SYS.ISYSTABCOL
 where ISYSTABCOL.table_id = tab.table_id),
if ix.index_id is null then 'N' else 'Y' endif,
null,
rmk.remarks
from SYS.SYSTAB as tab
 join SYS.ISYSDBSPACE as dbs on (tab.dbpace_id = dbs.dbpace_id)
 join SYS.ISYSUSER as u on u.user_id = tab.creator
 left outer join SYS.ISYSIDX as ix on (tab.table_id = ix.table_id
 and ix.index_id = 0)
 left outer join SYS.ISYSREMARK as rmk on (tab.object_id =
rmk.object_id)
```

SYSCERTIFICATE システムビュー

SYSCERTIFICATE システムビューの各ローには、テキスト PEM フォーマットの証明書が保管されています。このビューの基本となるシステムテーブルは ISYSCERTIFICATE です。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	証明書の ID。
cert_name	CHAR(128)	証明書の名前。
contents	LONG BINARY	圧縮形式の証明書の内容。
update_time	TIMESTAMP	最後に作成または置換を行ったローカル日時。
update_time_utc	TIMESTAMP WITH TIME ZONE	最後に作成または置換を行った UTC 日時。

基本となるシステムテーブルに関する制約

PRIMARY KEY (object_id)

UNIQUE INDEX (cert_name)

SYSCHECK システムビュー

SYSCHECK システムビューの各ローは、テーブル内の名前付き検査制約を定義します。このビューの基本となるシステムテーブルは ISYSCHECK です。

カラム名	データ型	説明
check_id	UNSIGNED INT	データベースの制約をユニークに識別する番号。
check_defn	LONG VARCHAR	CHECK 式。

基本となるシステムテーブルに関する制約

PRIMARY KEY (check_id)

```
FOREIGN KEY (check_id) REFERENCES SYS.ISYSCONSTRAINT (constraint_id)
```

SYSCOLAUTH 統合ビュー

SYSCOLAUTH ビューの各ローは、カラムに付与されている一連の権限 (UPDATE、SELECT、または REFERENCES) を示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW
"SYS"."SYSCOLAUTH" ( grantor,grantee,creator,tname,colname,
  privilege_type,is_grantable )
as select u1.user_name,u2.user_name,u3.user_name,tab.table_name,
  col.column_name,cp.privilege_type,cp.is_grantable
from SYS.ISYSCOLPERM as cp
  join SYS.ISYSUSER as u1 on u1.user_id = cp.grantor
  join SYS.ISYSUSER as u2 on u2.user_id = cp.grantee
  join SYS.ISYSTAB as tab on tab.table_id = cp.table_id
  join SYS.ISYSUSER as u3 on u3.user_id = tab.creator
  join SYS.ISYSTABCOL as col on col.table_id = cp.table_id
  and col.column_id = cp.column_id
```

SYSCOLLATION 互換ビュー (旧式)

SYSCOLLATION 互換ビューには、データベースの照合順情報が格納されます。組み込み関数経由で取得でき、カタログには保存されません。このビューの定義を次に示します。

```
ALTER VIEW "SYS"."SYSCOLLATION"
as select 1 as collation_id,
  DB_PROPERTY('Collation') as collation_label,
  DB_EXTENDED_PROPERTY('Collation','Description') as
collation_name,
  cast(DB_EXTENDED_PROPERTY('Collation','LegacyData') as
binary(1280)) as collation_order
```

SYSCOLLATIONMAPPINGS 互換ビュー (旧式)

SYSCOLLATIONMAPPINGS 互換ビューにはデータベース照合マッピングを持つローが 1 つだけ含まれています。組み込み関数経由で取得でき、カタログには保存されません。このビューの定義を次に示します。

```
ALTER VIEW "SYS"."SYSCOLLATIONMAPPINGS"
as select DB_PROPERTY('Collation') as collation_label,
  DB_EXTENDED_PROPERTY('Collation','Description') as
collation_name,
  DB_PROPERTY('Charset') as cs_label,
  DB_EXTENDED_PROPERTY('Collation','ASEnsensitiveSortOrder') as
so_case_label,
  DB_EXTENDED_PROPERTY('Collation','ASEnsensitiveSortOrder') as
so_caseless_label,
  DB_EXTENDED_PROPERTY('Charset','java') as jdk_label
```

SYSCOLPERM システムビュー

GRANT 文を使うと、テーブルの各カラムに UPDATE、SELECT、または REFERENCES 権限を与えることができます。UPDATE、SELECT、または REFERENCES 権限を持つ各カラムは、SYSCOLPERM システムビューの各ローに記録されます。このビューの基本となるシステムテーブルは ISYSCOLPERM です。

カラム名	データ型	説明
table_id	UNSIGNED INT	カラムが属するテーブルのテーブル番号。
grantee	UNSIGNED INT	カラムに権限を与えられたユーザ ID のユーザ番号。被付与者が PUBLIC ロールを持っている場合、権限はすべてのユーザ ID に付与されます。
grantor	UNSIGNED INT	権限を付与するユーザ ID のユーザ番号。
column_id	UNSIGNED INT	このカラム番号と table_id を使って、権限を付与されたカラムを識別します。
privilege_type	SMALLINT	このカラムの数値は、カラム権限の種類 (16=REFERENCES、1=SELECT、または 8=UPDATE) を示します。
is_grantable	CHAR(1)	カラムの権限が GRANT OPTION を使って付与されているかどうかを示します。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (table_id, grantee, grantor, column_id, privilege_type)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL (table_id, column_id)
```

```
FOREIGN KEY (grantor) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)
```

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ヒストグラムなどのカラムの統計情報が含まれます。このビューの内容を取り出すには、sa_get_histogram ストアドプロシージャまたはヒストグラムユーティリティを使用するのが最適です。このビューの基本となるシステムテーブルは ISYSCOLSTAT です。

カラム名	データ型	説明
table_id	UNSIGNED INT	このカラムが属するテーブルまたはマテリアライズドビューをユニークに識別する番号。
column_id	UNSIGNED INT	table_id とともに使用してカラムをユニークに識別する番号。
format_id	SMALLINT	システムでのみ使用。
update_time	TIMESTAMP	このカラムの統計情報が最後に更新されたローカル時刻。
density	FLOAT	カラムの単一の値の平均による選択性の推定。ローに格納されている大きい単一値の選択性は考慮されません。
max_steps	SMALLINT	システムでのみ使用。
actual_steps	SMALLINT	システムでのみ使用。
step_values	LONG BINARY	システムでのみ使用。
frequencies	LONG BINARY	システムでのみ使用。
update_time_utc	TIMESTAMP WITH TIME ZONE	このカラムの統計情報が最後に更新された UTC 時刻。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (table_id, column_id)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL  
(table_id, column_id)
```

SYSCOLSTATS 統合ビュー

SYSCOLSTATS ビューには、オプティマイザによって使用されるカラム統計が、ヒストグラムとして格納されます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSCOLSTATS" AS SELECT u.user_name, t.table_name,  
c.column_name, s.format_id,  
dateadd(mi, PROPERTY('TimeZoneAdjustment'), s.update_time) as  
update_time, s.density, s.max_steps, s.actual_steps,  
s.step_values, s.frequencies, TODATETIMEOFFSET( s.update_time,  
0 ) as update_time_utc  
FROM SYS.ISYSCOLSTAT s  
JOIN SYS.ISYSTABCOL c on (s.table_id = c.table_id and s.column_id =  
c.column_id)  
JOIN SYS.ISYSTAB t on (t.table_id = c.table_id)  
JOIN SYS.ISYSUSER u on (u.user_id = t.creator)
```

SYSCOLUMN 互換ビュー (旧式)

SYSCOLUMN ビューは、SYSCOLUMN システムテーブルを提供していた古いバージョンのソフトウェアとの互換性を保つために用意されています。ただし、以前の SYSCOLUMN テーブルは、SYSTABCOL システムビューに対応する ISYSTABCOL システムテーブルで置換されたため、ISYSTABCOL の使用をおすすめします。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSCOLUMN"
  as select b.table_id,
    b.column_id,
    if c.sequence is null then 'N' else 'Y' endif as pkey,
    b.domain_id,
    b.nulls,
    b.width,
    b.scale,
    b.object_id,
    b.max_identity,
    b.column_name,
    r.remarks,
    b."default",
    b.user_type,
    b.column_type
  from SYS.SYSTABCOL as b
    left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
    left outer join SYS.ISYSIDXCOL as c on(b.table_id = c.table_id
and b.column_id = c.column_id and c.index_id = 0)
```

SYSCOLUMNS 統合ビュー

SYSCOLUMNS ビューの各ローは、カタログ内の各テーブルとビューのカラム 1 つを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW
"SYS"."SYSCOLUMNS"( creator, cname, tname, coltype, nulls, length,
  syslength, in_primary_key, colno, default_value,
  column_kind, remarks )
  as select
u.user_name, col.column_name, tab.table_name, dom.domain_name,
  col.nulls, col.width, col.scale, if ixcol.sequence is null then 'N'
else 'Y' endif, col.column_id,
  col."default", col.column_type, rmk.remarks
  from SYS.SYSTABCOL as col
    left outer join SYS.ISYSIDXCOL as ixcol on(col.table_id =
```



```
ixcol.table_id and col.column_id = ixcol.column_id and
ixcol.index_id = 0)
    join SYS.ISYSTAB as tab on(tab.table_id = col.table_id)
    join SYS.ISYSDOMAIN as dom on(dom.domain_id = col.domain_id)
    join SYS.ISYSUSER as u on u.user_id = tab.creator
    left outer join SYS.ISYSREMARK as rmk on(col.object_id =
rmk.object_id)
```

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。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSCOMMENTS ASE 互換ビュー

syscomments テーブルには、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、関数ベースインデックスキー、その他の形式のコンパイル済みオブジェクトのそれぞれに対するエントリが含まれています。

このビューは、DBO が所有します。

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。それぞれのオブジェクトは最大 65,025 ローまで使用できます。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)

- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSCONSTRAINT システムビュー

SYSCONSTRAINT システムビューの各ローは、データベース内の名前付き制約を示します。このビューの基本となるシステムテーブルは ISYSCONSTRAINT です。

カラム名	データ型	説明
constraint_id	UNSIGNED INT	制約のユニークな ID。
constraint_type	CHAR(1)	制約の型。 <ul style="list-style-type: none"> • C - カラム検査制約 • T - テーブル制約 • P - プライマリキー • F - 外部キー • U - 一意性制約
ref_object_id	UNSIGNED BIGINT	制約の適用対象となるカラム、テーブル、インデックスのオブジェクト ID。
table_object_id	UNSIGNED BIGINT	制約の適用対象となるテーブルのオブジェクト ID。
constraint_name	CHAR(128)	制約名。

基本となるシステムテーブルに関する制約

PRIMARY KEY (constraint_id)

FOREIGN KEY (ref_object_id) REFERENCES SYS.ISYSOBJECT (object_id)

FOREIGN KEY (table_object_id) REFERENCES SYS.ISYSOBJECT (object_id)

UNIQUE CONSTRAINT (table_object_id, constraint_name)

SYSDBFILE システムビュー

SYSDBFILE システムビューの各ローは、DB 領域ファイルを示します。このビューの基本となるシステムテーブルは ISYSDBFILE です。

カラム名	データ型	説明
dbfile_id	SMALLINT	内部でのみ使用。
dbspace_id	SMALLINT	データベースの各 DB 領域ファイルには、ユニークな番号が割り当てられています。システム DB 領域には、すべてのシステムオブジェクトが含まれ、0 の dbspace_id があります。

カラム名	データ型	説明
dbfile_name	CHAR(128)	DB 領域のファイル名。
file_name	LONG VARCHAR	DB 領域のユニークなファイル名。これは CREATE TABLE コマンド中で使われます。
lob_map	LONG VARBIT	内部でのみ使用。

基本となるシステムテーブルに関する制約

PRIMARY KEY (dbfile_id)

FOREIGN KEY (dbspace_id) REFERENCES SYS.ISYSDBSPACE (dbspace_id)

UNIQUE index (file_name)

SYSDBSPACE システムビュー

SYSDBSPACE システムビューの各ローは、DB 領域ファイルを示します。このビューの基本となるシステムテーブルは ISYSDBSPACE です。

カラム名	データ型	説明
dbspace_id	SMALLINT	DB 領域を識別するユニークな番号。システム DB 領域には、すべてのシステムオブジェクトが含まれ、0 の dbspace_id があります。
object_id	UNSIGNED BIGINT	DB 領域のオブジェクト ID。
dbspace_name	CHAR(128)	DB 領域のユニークなファイル名。これは CREATE TABLE コマンド中で使われます。
store_type	TINYINT	内部でのみ使用。

基本となるシステムテーブルに関する制約

PRIMARY KEY (dbspace_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH
UNIQUE FULL

SYSDBSPACEPERM システムビュー

SYSDBSPACEPERM システムビューの各ローは、DB 領域ファイルの権限に関する記述です。このビューの基本となるシステムテーブルは ISYSDBSPACEPERM です。

カラム名	データ型	説明
dbspace_id	SMALLINT	DB 領域を識別するユニークな番号。システム DB 領域には、すべてのシステムオブジェクトが含まれ、0 の dbspace_id があります。
grantee	UNSIGNED INT	権限を取得するユーザのユーザ ID。
privilege_type	SMALLINT	grantee に付与される権限。たとえば、CREATE は DB 領域でオブジェクトを作成する権限を grantee に付与します。

基本となるシステムテーブルに関する制約

```
FOREIGN KEY (dbspace_id) REFERENCES SYS.ISYSDBSPACE (dbspace_id)
```

```
FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)
```

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間の依存性を示します。このビューの基本となるシステムテーブルは ISYSDEPENDENCY です。

あるオブジェクトが定義内の別のオブジェクトを参照する場合、この2つのデータベースオブジェクト間に依存性が存在します。たとえば、ビューのクエリ指定がテーブルを参照する場合、ビューはテーブルに依存しています。データベースサーバが、テーブル、ビュー、マテリアライズドビュー、カラムの依存性を追跡します。

カラム名	データ型	説明
ref_object_id	UNSIGNED BIGINT	参照オブジェクトのオブジェクト ID。
dep_object_id	UNSIGNED BIGINT	参照オブジェクトの ID。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (ref_object_id, dep_object_id)
```

```
FOREIGN KEY (ref_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (dep_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

SYSDOMAIN システムビュー

SYSDOMAIN システムビューは、組み込みデータ型に関する情報(ドメインとも呼びます)を記録します。通常の操作ではビューのコンテンツは変化しません。このビューの基本となるシステムテーブルは ISYSDOMAIN です。

カラム名	データ型	説明
domain_id	SMALLINT	各データ型に割り当てられたユニークな番号。この番号は変更できません。
domain_name	CHAR(128)	CREATE TABLE コマンド中に見られるデータ型の名前 (CHAR または INTEGER など)。
type_id	SMALLINT	ODBC データ型。この値は、Transact-SQL 互換の dbo.SYSTYPES テーブルにある data_type の値に対応します。
"precision"	SMALLINT	このデータ型を使って格納できる有効桁数。数値でないデータ型に対応するカラム値は NULL です。

基本となるシステムテーブルに関する制約

PRIMARY KEY (domain_id)

SYSEVENT システムビュー

SYSEVENT システムビューの各ローは、CREATE EVENT で作成されたイベントを示します。このビューの基本となるシステムテーブルは ISYSEVENT です。

カラム名	データ型	説明
event_id	UNSIGNED INT	各イベントに割り当てられたユニークな番号。
object_id	UNSIGNED BIGINT	データベースのイベントをユニークに識別するイベントの内部 ID。
creator	UNSIGNED INT	イベントの所有者のユーザ番号。ユーザ名は SYSUSER システムビューで確認できます。
event_name	VARCHAR(128)	イベント名。
enabled	CHAR(1)	イベントが起動できるかどうかを示します。

カラム名	データ型	説明
location	CHAR(1)	<p>イベントを起動するロケーション。</p> <p>Y = AT ALL 句と FOR PRIMARY 句が指定される E = AT CONSOLIDATED 句と FOR PRIMARY 句が指定される T = AT REMOTE 句と FOR PRIMARY 句が指定される P = (AT 句は指定されない) FOR PRIMARY 句が指定される B = AT ALL 句と FOR ALL 句が指定される D = AT CONSOLIDATED 句と FOR ALL 句が指定される S = AT REMOTE 句と FOR ALL 句が指定される M = (AT 句は指定されない) FOR ALL 句が指定される C = AT CONSOLIDATED (FOR 句は指定されない) R = AT REMOTE (FOR 句は指定されない) A = AT ALL 句 (FOR 句は指定されない)</p>
event_type_id	UNSIGNED INT	システムイベントの場合、イベント型は SYSEVENTTYPE システムビューにリストされます。
action	LONG VARCHAR	イベントハンドラ定義。難読化された値は、隠されたイベントを示します。
external_action	LONG VARCHAR	システムでのみ使用。
condition	LONG VARCHAR	イベントハンドラの起動を制御するのに使用される条件。
remarks	LONG VARCHAR	イベントの注釈。このカラムは ISYSREMARK に由来します。
source	LONG VARCHAR	イベントの元のソース。このカラムは ISYSSOURCE に由来します。

基本となるシステムテーブルに関する制約

PRIMARY KEY (event_id)

FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

UNIQUE INDEX (event_name)

SYSEVENTTYPE システムビュー

SYSEVENTTYPE システムビューは、CREATE EVENT で参照できるシステムのイベントタイプを定義します。

カラム名	データ型	説明
event_type_id	INT	各イベント型に割り当てられたユニークな番号。
name	VARCHAR(32000)	システムイベント型の名前。
description	VARCHAR(32000)	システムイベント型の説明。

備考

SYSEVENTTYPE システムビューは、sa_rowgenerator と次のサーバプロパティの組み合わせを使用して定義されます。

EventTypeName
 EventTypeDesc
 MaxEventType

SYSEXTERNENV システムビュー

C/C++ で記述された Embedded SQL と ODBC アプリケーション、Java、Perl、PHP、または Microsoft .NET Framework Common Language Runtime (CLR) に基づく C# や Visual Basic などの言語で記述されたアプリケーションを含む、多くの外部ランタイム環境がサポートされています。

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ために必要な情報を示します。このビューの基本となるシステムテーブルは ISYSEXTERNENV です。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	外部環境のユニークな識別子。
name	CHAR(128)	このカラムは、外部環境または言語の名前を識別します。これは、java、perl、php、clr、c_esql32、c_esql64、c_odbc32、c_odbc64 のいずれか 1 つです。

カラム名	データ型	説明
scope	CHAR(1)	このカラムは、CONNECTION の場合は C、DATABASE の場合は D のどちらかです。scope カラムは、外部環境が接続ごとに 1 つ起動されるのか、データベースごとに 1 つ起動されるのかを識別します。 接続ごとに 1 つ起動される外部環境 (PERL、PHP、C_ESQL32、C_ESQL64、C_ODBC32、C_ODBC64 など) では、外部環境を使用する接続ごとに外部環境のインスタンスが 1 つあります。接続ごとの場合、外部環境は接続が切断されると終了します。 データベースごとに 1 つ起動される外部環境 (JAVA や CLR など) では、外部環境を使用するデータベースごとに外部環境のインスタンスが 1 つあります。データベースごとに 1 つ起動される外部環境は、データベースが停止されると終了します。
support_result_sets	CHAR(1)	このカラムは、結果セットを返すことのできる外部環境を識別します。PERL と PHP 以外のすべての外部環境が結果セットを返すことができます。
location	LONG VARCHAR	このカラムは、外部環境の実行ファイル／バイナリファイルが置かれているデータベースサーバコンピュータのロケーションを識別します。この句には、実行プログラム／バイナリの名前を指定します。このパスは、完全修飾パスまたは相対パスのいずれかにできます。相対パスの場合、実行ファイル／バイナリファイルはデータベースサーバによって検索できるロケーションに置く必要があります。
options	LONG VARCHAR	このカラムは、外部環境に関連付けられている実行ファイルを起動するために、コマンドラインで指定する必要があるオプションを識別します。このカラムは変更しないでください。
user_id	UNSIGNED INT	外部環境が最初に起動すると、データベースへの接続を確立し、外部環境使用のための設定を実行します。デフォルトでは、この接続は DBA ユーザ ID を使用して作成されます。しかし、MANAGE ANY EXTERNAL OBJECT システム権限を持つ別のユーザ ID を外部環境で使用するのをデータベース管理者が望む場合は、user_id カラムに別のユーザ ID を指定します。通常、このカラムは NULL であるため、データベースサーバはデフォルトで DBA ユーザ ID を使用します。

基本となるシステムテーブルに関する制約

PRIMARY KEY (object_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)


```
UNIQUE INDEX (name)
```

SYSEXTERNENVOBJECT システムビュー

C/C++ で記述された Embedded SQL と ODBC アプリケーション、Java、Perl、PHP、または Microsoft .NET Framework Common Language Runtime (CLR) に基づく C# や Visual Basic などの言語で記述されたアプリケーションを含む、多くの外部ランタイム環境がサポートされています。

SYSEXTERNENVOBJECT システムビューの各ローは、インストールされた外部オブジェクトを示します。このビューの基本となるシステムテーブルは ISYSEXTERNENVOBJECT です。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	外部オブジェクトのユニークな識別子。
extenv_id	UNSIGNED BIGINT	外部環境のユニークな識別子 (SYSEXTERNENV.object_id)。
owner	UNSIGNED INT	このカラムは、外部オブジェクトの作成者/所有者を識別します。
name	LONG VARCHAR	このカラムは、INSTALL EXTERNAL OBJECT 文で指定されている外部オブジェクトの名前を識別します。
contents	LONG BINARY	外部オブジェクトの内容。
update_time	TIMESTAMP	このカラムは、オブジェクトが最後に変更 (またはインストール) されたローカル時刻を識別します。
update_time_utc	TIMESTAMP WITH TIME ZONE	このカラムは、オブジェクトが最後に変更 (またはインストール) された UTC 時刻を識別します。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (object_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (extenv_id) REFERENCES SYS.ISYSEXTERNENV (object_id)
```

```
FOREIGN KEY (owner) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (name)
```

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外部ログインを示します。このビューの基本となるシステムテーブルは ISYSEXTERNLOGIN です。

注意： SYSEXTERNLOGINS システムテーブルに含まれる前のカタログバージョン。このテーブル名は ISYSEXTERNLOGIN ('S' なし) に変更され、このビューの基本となるテーブルになります。

カラム名	データ型	説明
user_id	UNSIGNED INT	ローカルデータベースでのユーザ ID。
srvid	UNSIGNED INT	SYSSERVER システムビューにリストされているリモートサーバ。
remote_login	VARCHAR(128)	このユーザのリモートサーバのログイン名。
remote_password	VARBINARY(128)	このユーザのリモートサーバのパスワード。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (user_id, srvid)
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (srvid) REFERENCES SYS.ISYSSERVER (srvid)
```

SYSDATE 互換ビュー (旧式)

SYSDATE システムビューの各ローは、データベースの DB 領域を示します。各データベースは、1 つ以上の DB 領域で構成されます。各 DB 領域は、1 つのオペレーティングシステムファイルに対応します。

メインデータベースファイル、テンポラリファイル、トランザクションログファイル、トランザクションログミラーファイル用に、自動的に DB 領域が作成されます。トランザクションログ、トランザクションログミラーの DB 領域に関する情報は、SYSDATE システムビューには表示されません。

```
ALTER VIEW "SYS"."SYSDATE"
  as select b.dbfile_id as file_id,
    if b.dbpace_id = 0 and b.dbfile_id = 0 then
      db_property('File')
    else
      if b.dbpace_id = 15 and b.dbfile_id = 15 then
        db_property('TempFileName')
      else
        b.file_name
      endif
    endif as file_name,
  a.dbpace_name,
```

```

a.store_type,
b.lob_map,
b.dbspace_id
from SYS.ISYSDBSPACE as a
join SYS.ISYSDBFILE as b on(a.dbspace_id = b.dbspace_id)

```

SYSFKCOL 互換ビュー (旧式)

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関係を示します。このビューは使用されなくなりました。代わりに SYSIDX と SYSIDXCOL のシステムビューを使用してください。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```

ALTER VIEW "SYS"."SYSFKCOL"
as select a.table_id as foreign_table_id,
a.index_id as foreign_key_id,
a.column_id as foreign_column_id,
a.primary_column_id
from SYS.ISYSIDXCOL as a
,SYS.ISYSIDX as b
where a.table_id = b.table_id
and a.index_id = b.index_id
and b.index_category = 2

```

SYSFKEY システムビュー

SYSFKEY システムビューの各ローは、システム内の外部キー制約を示します。このビューの基本となるシステムテーブルは ISYSFKEY です。

カラム名	データ型	説明
foreign_table_id	UNSIGNED INT	外部テーブルのテーブル番号。
foreign_index_id	UNSIGNED INT	外部キーのインデックス番号。
primary_table_id	UNSIGNED INT	プライマリテーブルのテーブル番号。
primary_index_id	UNSIGNED INT	プライマリキーのインデックス番号。

カラム名	データ型	説明
match_type	TINYINT	制約と一致する型。一致する型には次の型があります。 <ul style="list-style-type: none"> • 0 – デフォルトのマッチングを使用 • 1 – SIMPLE • 2 – FULL • 129 – SIMPLE UNIQUE • 130 – FULL UNIQUE
check_on_commit	CHAR(1)	COMMIT で外部キーが有効であるかどうかを確認されるまで、INSERT と UPDATE 文を待たせるかどうかを示します。
nulls	CHAR(1)	外部キーのカラムが NULL 値を許容するかどうかを示します。この設定は外部キーのカラム中の nulls 設定とは独立しています。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (foreign_table_id, foreign_index_id)
```

```
FOREIGN KEY (foreign_table_id, foreign_index_id) REFERENCES  
SYS.ISYSIDX (table_id, index_id)
```

```
FOREIGN KEY (primary_table_id, primary_index_id) REFERENCES  
SYS.ISYSIDX (table_id, index_id)
```

SYSFOREIGNKEY 互換ビュー (旧式)

SYSFOREIGNKEY ビューは、SYSFOREIGNKEY システムテーブルを提供していた古いバージョンのソフトウェアとの互換性を保つために用意されています。ただし、以前の SYSFOREIGNKEY システムテーブルは、SYSFKEY システムビューに対応する ISYSFKEY システムテーブルで置換されたため、ISYSFKEY の使用をおすすめします。

外部キーは、外部テーブルとプライマリテーブルの2つのテーブル間の関係です。外部キーは、SYSFOREIGNKEY の1つのローと SYSFKCOL の1つまたは複数のローで定義されます。SYSFOREIGNKEY には、外部キーに関する一般的な情報が入っています。SYSFKCOL は外部キーのカラムを識別して、外部キーの各カラムとプライマリテーブルの中のプライマリキーのカラムを関連付けます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSFOREIGNKEY"  
as select b.foreign_table_id,  
b.foreign_index_id as foreign_key_id,
```

```

a.object_id,
b.primary_table_id,
p.root,
b.check_on_commit,
b.nulls,
a.index_name as role,
r.remarks,
b.primary_index_id,
a.not_enforced as fk_not_enforced,
10 as hash_limit
from(SYS.ISYSIDX as a left outer join SYS.ISYSPHYSIDX as p
on(a.table_id = p.table_id and a.phys_index_id = p.phys_index_id))
  left outer join SYS.ISYSREMARK as r on(a.object_id =
r.object_id)
  ,SYS.ISYSFKEY as b
where a.table_id = b.foreign_table_id
and a.index_id = b.foreign_index_id

```

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外部キー 1 つを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```

ALTER VIEW "SYS"."SYSFOREIGNKEYS"( foreign_creator,
foreign_tname,
primary_creator,primary_tname,role,columns )
as select fk_up.user_name,fk_tab.table_name,pk_up.user_name,
pk_tab.table_name,ix.index_name,
(select list(string(fk_col.column_name,' IS ',
pk_col.column_name)
order by fkc.table_id,fkc.index_id,fkc."sequence")
from SYS.ISYSIDXCOL as fkc
  join SYS.ISYSTABCOL as fk_col on(
fkc.table_id = fk_col.table_id
and fkc.column_id = fk_col.column_id)
  ,SYS.ISYSTABCOL as pk_col
where fkc.table_id = fk.foreign_table_id
and fkc.index_id = fk.foreign_index_id
and pk_col.table_id = fk.primary_table_id
and pk_col.column_id = fkc.primary_column_id)
from SYS.ISYSFKEY as fk
  join SYS.ISYSTAB as fk_tab on fk_tab.table_id =
fk.foreign_table_id
  join SYS.ISYSUSER as fk_up on fk_up.user_id = fk_tab.creator
  join SYS.ISYSTAB as pk_tab on pk_tab.table_id =
fk.primary_table_id
  join SYS.ISYSUSER as pk_up on pk_up.user_id = pk_tab.creator
  join SYS.ISYSIDX as ix on ix.table_id = fk.foreign_table_id and
ix.index_id = fk.foreign_index_id

```

SYSGROUP 互換ビュー

SYSGROUP システムビューには、各グループの各メンバーに1つのローがあります。このビューは、グループとメンバーの多対多の関係を示します。グループには複数のメンバーがあり、ユーザは複数のグループのメンバーになることができます。

カラム名	データ型	説明
group_id	UNSIGNED INT	グループのユーザ番号。
group_member	UNSIGNED INT	メンバーのユーザ番号。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (group_id, group_member)
```

```
FOREIGN KEY group_id (group_id) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY group_member (group_member) REFERENCES SYS.ISYSUSER (user_id)
```

SYSGROUPS 互換ビュー

SYSGROUPS ビューには、各グループの各メンバーに1つのローがあります。このビューは、グループとメンバーの多対多の関係を示します。グループには複数のメンバーがあり、ユーザは複数のグループのメンバーになることができます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSGROUPS"( group_name,
    member_name )
as select g.user_name,u.user_name
    from SYS.ISYSROLEGRANT,SYS.ISYSUSER as g,SYS.ISYSUSER as u
    where ISYSROLEGRANT.role_id = g.user_id and ISYSROLEGRANT.grantee
= u.user_id and(
    u.user_name in( 'SYS_SPATIAL_ADMIN_ROLE' )
    or u.user_id <= 2147483648) and(
    g.user_type = (0x02|0x04|0x08)
    or g.user_name in( 'SYS','PUBLIC','dbo','diagnostics',
    'rs_systabgroup','SA_DEBUG','SYS_SPATIAL_ADMIN_ROLE' ) )
```

SYSHISTORY システムビュー

SYSHISTORY システムビューの各ローは、データベースの開始、データベースの調整など、データベースに対するシステム操作を記録します。このビューの基本となるシステムテーブルは ISYSHISTORY です。

カラム名	データ型	説明
operation	CHAR(128)	<p>データベースファイルに対して実行されるオペレーションのタイプ。operation には次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> • INIT – データベースがいつ作成されたかに関する情報。 • UPGRADE – データベースがいつアップグレードされたかに関する情報。 • START – 特定のオペレーティングシステム上で特定のバージョンのデータベースサーバを使用してデータベースがいつ開始されたかに関する情報。 • LAST_START – データベースサーバが最後に開始された時刻に関する情報。LAST_START オペレーションは、LAST_START ローに現在格納されている値とは異なるバージョンのデータベースサーバか異なるオペレーティングシステム、またはその両方でデータベースが開始されたときに START オペレーションに変換されます。 • DTT – DB 領域で実行される 2 番目から最後のディスク転送時間 (DTT: Disk Transfer Time) の調整操作に関する情報。つまり、ALTER DATABASE CALIBRATE 文か ALTER DATABASE RESTORE DEFAULT CALIBRATION 文のどちらかの最後から 2 番目の実行に関する情報です。 • LAST_DTT – DB 領域で実行される <i>最新の</i> DTT 調整操作に関する情報。つまり、ALTER DATABASE CALIBRATE 文か ALTER DATABASE RESTORE DEFAULT CALIBRATION 文のどちらかの最新の実行に関する情報です。 • LAST_BACKUP – 最後のバックアップ (バックアップの日時を含みます)、バックアップの種類、バックアップするファイル、バックアップを実行したデータベースサーバのバージョンに関する情報。

カラム名	データ型	説明
object_id	UNSIGNED INT	DTT と LAST_DTT 以外の操作については、このカラムの値は 0 になります。DTT と LAST_DTT の操作の場合、SYSDBSPACE システムビューで定義されている DB 領域の dbspace_id です。
sub_operation	CHAR(128)	DTT と LAST_DTT 以外の操作については、このカラムの値は、空の単一引用符 ("") のセットになります。DTT と LAST_DTT の操作の場合、このカラムには、DB 領域で実行されるサブ操作の種類も含まれます。次のような値があります。 <ul style="list-style-type: none"> • DTT_SET - DB 領域の調整が設定されます。 • DTT_UNSET - DB 領域のデフォルト設定が復元されます。
version	CHAR(128)	オペレーションの実行に使用されるデータベースサーバのバージョンとビルド番号。
platform	CHAR(128)	オペレーションが実行されたオペレーティングシステム。
first_time	TIMESTAMP	特定のオペレーティングシステム上の特定のバージョンのソフトウェアでデータベースが最初に開始されたローカル日時。
last_time	TIMESTAMP	特定のオペレーティングシステム上の特定のバージョンのソフトウェアでデータベースが最後に開始されたローカル日時。
details	LONG VARCHAR	このカラムは、データベースサーバの起動に使用されたコマンドラインオプションやデータベースに対して有効になっている機能ビットなどの情報を格納します。この情報は、Sybase 製品の保守契約を結んでいるサポートセンタが使用します。
first_time_utc	TIMESTAMP WITH TIME ZONE	特定のオペレーティングシステム上の特定のバージョンのソフトウェアでデータベースが最初に開始された UTC 日時。
last_time_utc	TIMESTAMP WITH TIME ZONE	特定のオペレーティングシステム上の特定のバージョンのソフトウェアでデータベースが最後に開始された UTC 日時。

基本となるシステムテーブルに関する制約

PRIMARY KEY (operation, object_id, version, platform)

SYSIDX システムビュー

SYSIDX システムビューの各ローは、データベースの論理インデックスを定義します。このビューの基本となるシステムテーブルは ISYSIDX です。

カラム名	データ型	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ユニークに識別します。
index_id	UNSIGNED INT	テーブル内のインデックスを識別するユニークな番号。
object_id	UNSIGNED BIGINT	データベースのインデックスをユニークに識別するインデックスの内部 ID。
phys_index_id	UNSIGNED INT	論理インデックスの実装に使用する基本となる物理インデックスを識別します。この値は、テンポラリテーブルまたはリモートテーブル上のインデックスに対しては NULL となります。その他の場合、この値は SYSPHYSIDX システムビューの物理インデックスの object_id に対応します。
dbspace_id	SMALLINT	インデックスを含むファイルの ID。この値は SYSDBSpace システムビューのエントリに対応します。
index_category	TINYINT	インデックスの型。次のような値があります。 <ul style="list-style-type: none"> • 1 – プライマリキー • 2 – 外部キー • 3 – セカンダリインデックス (一意性制約を含む) • 4 – テキストインデックス
"unique"	TINYINT	インデックスがユニークインデックス (1) か、一意性制約 (2) か、予約済み (3) か、ユニークでないインデックス (4) か、またはユニークインデックス WITH NULLS NOT DISTINCT (1) かを示します。ユニークインデックスでは、インデックスカラムの 2 つのローは同じ値を持ってません。
index_name	CHAR(128)	インデックスの名前。
not_enforced	CHAR(1)	システムでのみ使用。
file_id	SMALLINT	廃止予定。このカラムは SYSVIEW に存在しますが、基本となるシステムテーブル ISYSIDX には存在しません。このカラムの内容は dbspace_id と同じであり、互換性のために提供されています。今後は dbspace_id を使用してください。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (table_id, index_id)

FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH
UNIQUE FULL

FOREIGN KEY (table_id, phys_index_id) REFERENCES SYS.ISYSPHYSIDX
(table_id, phys_index_id)

UNIQUE INDEX (index_name, table_id, index_category)
```

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インデックスのカラム1つを示します。このビューの基本となるシステムテーブルは ISYSIDXCOL です。

カラム名	データ型	説明
table_id	UNSIGNED INT	インデックスが適用されるテーブルを識別します。
index_id	UNSIGNED INT	カラムが適用されるインデックスを識別します。table_id と index_id は共同で SYSIDX システムビューにある1つのインデックスを識別します。
sequence	SMALLINT	インデックス内の各カラムには、0 から始まるユニークな番号が割り当てられます。この番号はインデックス内のカラムの相対的な重要度を示します。もっとも重要なカラムの sequence 番号は 0 になります。
column_id	UNSIGNED INT	インデックスを作成するテーブルのカラムを識別します。table_id と column_id は共同で、SYSCOLUMN システムビューで表される1つのカラムを識別します。
"order"	CHAR(1)	インデックス内のカラムが昇順 (A) か、降順 (D) かを示します。テキストインデックスの場合、この値は NULL です。
primary_column_id	UNSIGNED INT	この外部キーカラムに対応するプライマリキーの ID。非外部キーのカラムの場合、値は NULL です。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (table_id, index_id, column_id)

FOREIGN KEY (table_id, index_id) REFERENCES SYS.ISYSIDX (table_id,
index_id)
```

```
FOREIGN KEY (table_id, column_id) REFERENCES SYS.ISYSTABCOL
(table_id, column_id)
```

SYSINDEX 互換ビュー (旧式)

SYSINDEX ビューは、SYSINDEX システムテーブルを提供していた古いバージョンのソフトウェアとの互換性を保つために用意されています。ただし、以前のSYSINDEX テーブルは、SYSIDX システムビューに対応する ISYSIDX システムテーブルで置換されたため、ISYSIDX の使用をおすすめします。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSINDEX"
as select b.table_id,
       b.index_id,
       b.object_id,
       p.root,
       b.dbspace_id,
       case b."unique"
       when 1 then 'Y'
       when 2 then 'U'
       when 3 then 'M'
       when 4 then 'N'
       when 5 then 'Y'
       else 'I'
       end as "unique",
       t.creator,
       b.index_name,
       r.remarks,
       10 as hash_limit,
       b.dbspace_id as file_id
from(SYS.ISYSIDX as b left outer join SYS.ISYSINDEX as p
on(b.table_id = p.table_id and b.phys_index_id = p.phys_index_id))
left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
,SYS.ISYSTAB as t
where t.table_id = b.table_id
and b.index_category = 3
```

SYSINDEXES 統合ビュー

SYSINDEXES ビューの各ローは、データベース内のインデックス1つを示します。このビューの代わりに、SYSIDX と SYSIDXCOL のシステムビューを使用することもできます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSINDEXES"( icreator,
       iname, fname, creator, tname, indextype,
       colnames, interval, level_num )
```

```

as select u.user_name,idx.index_name,dbs.dbSPACE_name,u.user_name,
tab.table_name,
case idx.index_category
when 1 then 'Primary Key'
when 2 then 'Foreign Key'
when 3 then (
if idx."unique" = 4 then 'Non-unique'
else if idx."unique" = 2 then 'UNIQUE constraint'
else if idx."unique" = 5 then 'UNIQUE NULLS NOT DISTINCT'
else 'UNIQUE'
endif
endif
endif) when 4 then 'Text Index' end,(select
list(string(c.column_name,
if icx."order" = 'A' then ' ASC' else ' DESC' endif) order by
ixc.table_id asc,ixc.index_id asc,ixc.sequence asc)
from SYS.ISYSIDXCOL as icx
join SYS.ISYSTABCOL as c on(
c.table_id = icx.table_id
and c.column_id = icx.column_id)
where icx.index_id = idx.index_id
and icx.table_id = idx.table_id),
0,0
from SYS.ISYSTAB as tab
join SYS.ISYSDBSPACE as dbs on(tab.dbSPACE_id = dbs.dbSPACE_id)
join SYS.ISYSIDX as idx on(idx.table_id = tab.table_id)
join SYS.ISYSUSER as u on u.user_id = tab.creator

```

SYSINDEXES ASE 互換ビュー

sysindexes には、クラスタードインデックスごとに1つのロー、ノンクラスタードインデックスごとに1つのロー、クラスタードインデックスのないテーブルごとに1つのロー、text カラムや image カラムのあるテーブルごとに1つのローが含まれています。

このテーブルには、関数ベースインデックスまたは計算カラムに作成されたインデックスごとに1つのローも格納されています。

このビューは、DBO が所有します。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYS COMMENTS ASE 互換ビュー (763 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSIQVINDE X ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSINFO 互換ビュー (旧式)

SYSINFO ビューは、データベースが作成されたときに定義された、データベースの情報を示します。このテーブルは常に1つのローを持ちます。このビューは組み込み関数経由で取得でき、カタログには保存されません。次に SYSINFO ビューの定義を示します。

```
ALTER VIEW "SYS"."SYSINFO"( page_size,
    encryption,
    blank_padding,
    case_sensitivity,
    default_collation,
    database_version )
as select db_property('PageSize'),
    if db_property('Encryption') <> 'None' then 'Y' else 'N' endif,
    if db_property('BlankPadding') = 'On' then 'Y' else 'N' endif,
    if db_property('CaseSensitive') = 'On' then 'Y' else 'N' endif,
    db_property('Collation'),
    NULL
```

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
type	tinyint	NOT null	バックアップのタイプ： 0 = FULL 1 = INCREMENTAL 2 = INCREMENTAL SINCE FULL
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

カラム名	カラムタイプ	カラム制約	説明
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
version	unsigned int	NOT null	バックアップのバージョン

基本となるシステムテーブルでの制約

プライマリキー (bu_id)

参照：

- sp_iqbackupdetails プロシージャ (438 ページ)

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。

カラム名	カラムタイプ	説明
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。

基本となるシステムテーブルでの制約

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。

参照：

- SYSIQTABCOL システムビュー (798 ページ)

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

カラム名	カラムタイプ	説明
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	SAP 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
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	unsigned bigint	その dbfile ファイルを作成したトランザクション ID
alter_txn_id	unsigned bigint	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	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 DB 領域名

基本となるシステムテーブルでの制約

外部キー (server_id) が SYS.ISYSIQMPXSERVER を参照。

ユニーク (server_id, file_name)

参照：

- SYSIQFILE システムビュー (廃止) (787 ページ)

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	カラム型	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

カラム名	カラム型	説明
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
online	char(1)	'T' - オンライン、'F' - オフライン
create_txn_id	unsigned bigint	DB 領域を作成するトランザクション ID
alter_txn_id	unsigned bigint	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 DB ファイルに移動する前に、DB 領域の各ファイルに書き込まれるデータ量 (KB)
is_rlv_store	char(1)	'T' - DB 領域は RLV ストアの DB 領域、'F' - DB 領域は MIN、SHARED_TEMP、または TEMPORARY ストアの DB 領域

基本となるシステムテーブルでの制約

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。

参照：

- SYSIQDBFILE システムビュー (785 ページ)

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

カラム名	カラムタイプ	説明
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_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とき使用さ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

基本となるシステムテーブルでの制約

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 SAP 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。このテーブルは常に 1 つのローを持ちます。

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

カラム名	カラムタイプ	説明
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ー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	unsigned bigint	内部用
rd_commit_txn_id	unsigned bigint	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
last_multiplex_mode	TINYINT NULL	(SAP Sybase IQ 16.0 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

SYSIQLOGICALSERVER システムビュー

システムテーブル ISYSIQLOGICALSERVER を読みやすい形式で表示します。

ISYSIQLOGICALSERVER システムテーブルには、論理サーバの情報と、関連付けられた論理サーバポリシーの情報が格納されています。

カラム名	カラム型	説明
ls_id	UNSIGNED BIGINT NOT NULL	論理サーバの ID 番号。
ls_object_id	UNSIGNED BIGINT NOT NULL	論理サーバのオブジェクト ID 番号。

カラム名	カラム型	説明
ls_policy_id	UNSIGNED BIGINT NOT NULL	論理サーバポリシーの ID 番号。
ls_name	CHAR(128) NOT NULL UNIQUE	論理サーバ名。

基本となるシステムテーブルでの制約:

- プライマリキー (ls_id)
- object_id 外部キー (ISYSOBJECT)
- ls_policy_id 外部キー (ISYSIQLSPOLICY)

SYSIQLOGINPOLICYLSINFO システムビュー

テーブル ISYSIQLLOGINPOLICYLSINFO を読みやすい形式で表示します。

システムテーブル ISYSIQLLOGINPOLICYLSINFO には、ログインポリシーの論理サーバ割り当ての情報が格納されています。

カラム名	カラム型	説明
login_policy_id	UNSIGNED BIGINT NOT NULL	ログインポリシーの ID 番号。
ls_id	UNSIGNED BIGINT NOT NULL	論理サーバの ID 番号。

基本となるシステムテーブルでの制約:

- プライマリキー (login_policy_id、ls_id)
- login_policy_id 外部キー (ISYSLOGINPOLICY)
- ls_id 外部キー (ISYSIQLLOGICALSERVER)

SYSIQLSLOGINPOLICIES 統合ビュー

ログインポリシーのすべての論理サーバ割り当てを示します。

この統合システムビューは、SYSIQLLOGICALSERVER、ISYSIQLLOGINPOLICYLSINFO、ISYSLOGINPOLICY からの情報を表示します。

カラム名	カラム型	説明
ls_id	UNSIGNED BIGINT NOT NULL	論理サーバの ID 番号。
ls_name	CHAR(128)	論理サーバ名。
login_policy_id	UNSIGNED BIGINT NOT NULL	ログインポリシーの ID 番号。
login_policy_name	char(128)	ログインポリシーの名前。

SYSIQLSLOGINPOLICYOPTION システムビュー

テーブル ISYSIQLSLOGINPOLICYOPTION を読みやすい形式で表示します。

ISYSIQLSLOGINPOLICYOPTION テーブルには、ログインポリシーオプション値の論理サーバレベルの設定が格納されています。

カラム名	カラム型	説明
login_policy_id	UNSIGNED BIGINT NOT NULL	ログインポリシーの ID 番号。
ls_id	UNSIGNED BIGINT NOT NULL	論理サーバの ID 番号。
login_option_name	CHAR(128) NOT NULL	ログインポリシーオプションの名前。
login_option_value	LONG VARCHAR NOT NULL	ログインポリシーオプションの値。

基本となるシステムテーブルでの制約:

- プライマリキー (login_policy_id、ls_id、login_option_name)
- login_policy_id 外部キー (ISYSLOGINPOLICY)
- ls_id 外部キー (ISYSIQLOGICALSERVER)

SYSIQLSMEMBER システムビュー

ISYSIQLSMEMBER テーブルのグループ情報を表示します。このテーブルには、論理サーバのメンバーシップ情報が格納されています。

ISYSIQLSMEMBER には、論理サーバの情報と対応するマルチプレックスサーバの情報が格納されています。

カラム名	カラム型	説明
ls_id	UNSIGNED BIGINT NOT NULL	論理サーバの ID 番号。
logical_membership_type	TINYINT NOT NULL	論理メンバーシップのタイプ。
mpx_server_id	UNSIGNED INT NOT NULL	マルチプレックスサーバの ID 番号。
membership_info	UNSIGNED INT NOT NULL	メンバーシップ情報。

基本となるシステムテーブルでの制約:

- プライマリキー (ls_id、logical_membership_id、mpx_server_id)
- ls_id 外部キー (ISYSIQLOGICALSERVER)

システムテーブルとシステムビュー

論理サーバのメンバーシップがマルチプレックスサーバ名を使用して定義されている場合、logical_membership_type の値は 0 で、mpx_server_id はマルチプレックスサーバのサーバ ID です。

コーディネータの論理メンバーシップの場合、mpx_server_id は 0 で logical_membership_type は 1 です。

SYSIQLSMEMBERS 統合ビュー

ユーザ定義論理サーバのすべてのメンバーシップを示します。

カラム名	カラム型	説明
ls_id	UNSIGNED BIGINT NOT NULL	論理サーバの ID 番号。
ls_name	CHAR(128) NOT NULL	論理サーバ名。
server_id	UNSIGNED INT NOT NULL	サーバ名を使用して定義されているメンバーシップの場合はメンバのマルチプレックスサーバ ID。コーディネータの論理メンバーシップの場合は 0。
server_name	CHAR(128) NOT NULL	サーバ名を使用して定義されているメンバーシップの場合はメンバのマルチプレックスサーバ名。コーディネータの論理メンバーシップの場合は 'LOGICAL COORDINATOR'。
membership_type	TINYINT NOT NULL	サーバ名を使用して定義されているメンバーシップの場合は 0、コーディネータの論理メンバーシップの場合は 1。

SYSIQLSPOLICY システムビュー

テーブル ISYSIQLSPOLICY を読みやすい形式で表示します。

システムテーブル ISYSIQLSPOLICY には、論理サーバポリシーが格納されています。

カラム名	カラム型	説明
ls_policy_id	UNSIGNED BIGINT NOT NULL	論理サーバポリシーの ID 番号。
ls_policy_name	CHAR(128) NOT NULL UNIQUE	論理サーバポリシー名。

基本となるシステムテーブルでの制約:

- プライマリキー (ls_policy_id)
- object_id 外部キー (ISYSOBJECT)

SYSIQLSPOLICYOPTION システムビュー

テーブル ISYSIQLSPOLICYOPTION を読みやすい形式で表示します。

ISYSIQLSPOLICYOPTION テーブルには、論理サーバポリシーオプションが格納されています。

カラム名	カラム型	説明
ls_policy_id	UNSIGNED BIGINT NOT NULL	ログインポリシーの ID 番号。
ls_policy_option_name	CHAR(128) NOT NULL	論理サーバポリシーオプション名。
ls_policy_option_value	LONG VARCHAR NOT NULL	論理サーバポリシーオプション値。

基本となるシステムテーブルでの制約:

- プライマリキー (ls_policy_id、ls_policy_option_name)
- ls_policy_id 外部キー (ISYSIQLSPOLICY)

SYSIQMPXSERVER システムビュー

テーブル ISYSIQMPXSERVER を読みやすい形式で表示します。

ISYSIQMPXSERVER システムテーブルには、特定のマルチプレックスノードのメンバーシッププロパティおよびバージョンステータスデータが格納されています。

カラム名	カラムタイプ	説明
server_id	UNSIGNED INT NOT NULL	サーバの ID 番号。
server_name	CHAR(128) NOT NULL	サーバ名。大文字と小文字の区別がないユニークな名前でないといけない。
role	TINYINT NOT NULL	コーディネータ、リーダ、またはライター。
status	TINYINT NOT NULL	excluded または included。
current_version	UNSIGNED BIGINT NOT NULL	サーバの現在のバージョン ID。
active_version	LONG BINARY NULL	サーバ上でアクティブなバージョンのリスト (エンコード済み)。
connection_info	LONG VARCHAR NULL	セミコロンで区切られた、パブリックドメイン接続用のホスト名とポートのペアを含む文字列。

カラム名	カラムタイプ	説明
db_path	LONG VARCHAR NOT NULL	サーバのデータベースファイルへのフルパス。
private_connection_info	LONG VARCHAR NULL	セミコロンで区切られた、プライベートネットワーク接続用のホスト名とポートのペアを含む文字列。

基本となるシステムテーブルでの制約:

- プライマリキー (server_id)

SYSIQMPXSERVERAGENT システムビュー

テーブル ISYSIQMPXSERVERAGENT の読み込み可能バージョンを表示します。ISYSIQMPXSERVERAGENT システムテーブルには、指定されたマルチプレックスサーバのエージェント接続定義が格納されています。

カラム名	カラム型	説明
server_id	UNSIGNED INT NOT NULL	サーバの ID 番号。
agent_connection_info	LONG VARCHAR NOT NULL	セミコロンで区切られた、各マルチプレックスノード上の SCC IQ エージェント接続用のホスト名とポートのペアを含む文字列。
agent_user_name	LONG VARCHAR NOT NULL	SCC IQ エージェントのユーザ名を含む文字列。
agent_pwd	VARBINARY(1024) NOT NULL	SCC IQ エージェントの暗号化パスワードを含む文字列。

基本となるシステムテーブルでの制約:

- プライマリキー (server_id)
- 外部キー (server_id) が SYS.SYSIQMPXSERVER を参照

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、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。

参照:

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)

- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYSCOMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSIQPARTITIONCOLUMN システムビュー

ISYSIQPARTITIONCOLUMN からのグループ情報を読みやすい形式で表示します。

```
ALTER VIEW "SYS"."SYSIQPARTITIONCOLUMN"
as select * from SYS.ISYSIQPARTITIONCOLUMN
```

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する。
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。

基本となるシステムテーブルでの制約

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。

SYSIQRLVMERGEHISTORY システムビュー

ローレベルのバージョン管理 (RLV) ストアと IQ メインストアとのマージが開始するたびに RLV 対応テーブルごとにログエントリが追加されます。マージが完了すると、ログエントリが更新されます。

カラム名	カラムの型	説明
merge_id	unsigned bigint	ログエントリのユニークな識別子
table_id	unsigned int	sys.systable システムテーブルの外部キー
start_time	timestamp	マージが開始した時刻
end_time	timestamp	マージが終了した時刻
status	char (9)	STARTED COMPLETED FAILED
return_code	tinyint	マージの完了後の SQL コード
merge_type	char (9)	マージトリガの原因: AUTOMATIC DML DDL SHUTDOWN USER
merge_mode	char (12)	BLOCKING NON-BLOCKING
merge_detail	varchar (255)	エラー情報などの追加情報 (提供された場合)
rows_inserted	unsigned bigint	マージの結果として挿入されたローの数
rows_updated	unsigned bigint	マージの結果として更新されたローの数
rows_deleted	unsigned bigint	マージの結果として削除されたローの数
rows_forwarded	unsigned bigint	マージの時点でコミットされていなかったローの数

SYSIQRVLOG システムビュー

ISYSIQRVLOG からのグループ情報を読みやすい形式で表示します。

SYSIQRVLOG ビュー内の各ローは RLV 対応テーブルのログに対応します。

table_id 0 が含まれるローは、サーバ全体のコミットログを表します。

カラム名	カラム型	説明
stream_id	unsigned int	ログストリームの識別子。
table_id	unsigned int	ログストリームが所属するテーブルを示す。NULL は、コミットログストリームを示す。
partition_low	unsigned int	ログが最後にアクティブだったときに使用されていたパーティションマップに対応。

カラム名	カラム型	説明
partition_high	int	ログが最後にアクティブだったときに使用されていたパーティションマップに対応。
identity_location	unsigned bigint	ログストリームの ID ブロックの場所。

SYSIQTAB システムビュー

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。

```
ALTER VIEW "SYS"."SYSIQTAB"
as select * from SYS.ISYSIQTAB
```

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	カラム型	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) が割り当てられている。
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。
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	unsigned bigint	内部用。
txn_id	unsigned bigint	内部用。
update_time	timestamp	IQ テーブルの最終変更日時。
is_rlv	char(1)	'T' - RLV ストレージがテーブルで有効、'F' - RLV ストレージがテーブルで無効。

基本となるシステムテーブルでの制約

プライマリキー (table_id)

参照：

- SYSIQTABLE システムビュー (廃止) (798 ページ)

SYSIQTABCOL システムビュー

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。

```
ALTER VIEW "SYS"."SYSIQTABCOL"
as select * from SYS.ISYSIQTABCOL
```

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
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す (T/F)。
is_nbit	char(1)	カラムが NBit (T) であるかフラット FP (F) であるかを示す。

基本となるシステムテーブルでの制約

プライマリキー (table_id)

参照：

- SYSIQCOLUMN システムビュー (廃止) (785 ページ)

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。

参照：

- SYSIQTAB システムビュー (797 ページ)

SYSIQVINDEXT ASE 互換ビュー

sysiqvindex では、FP 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。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYSCOMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSIXCOL 互換ビュー (旧式)

SYSIXCOL ビューは、SYSIXCOL システムテーブルを提供していた古いバージョンのソフトウェアとの互換性を保つために用意されています。ただし、SYSIXCOL システムテーブルは ISYSIDXCOL システムテーブルで置換され、ISYSIDXCOL システムビューに対応しています。SYSIDXCOL システムビューの使用をおすすめします。

SYSIXCOL の各ローは、インデックスのカラムを示します。ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSIXCOL"
as select a.table_id,
       a.index_id,
       a.sequence,
       a.column_id,
       a."order"
from SYS.ISYSIDXCOL as a
     ,SYS.ISYSIDX as b
where a.table_id = b.table_id
     and a.index_id = b.index_id
     and b.index_category = 3
```

SYSJAR システムビュー

SYSJAR システムビューの各ローは、データベースに格納されている JAR ファイルを定義します。このビューの基本となるシステムテーブルは ISYSJAR です。

カラム名	データ型	説明
jar_id	INTEGER	JAR ファイルを識別するユニークな番号。
object_id	UNSIGNED BIGINT	データベースのインデックスをユニークに識別する JAR ファイルの内部 ID。
creator	UNSIGNED INT	JAR ファイルの作成者のユーザ番号。INSTALL JAVA 文の AS USER 句で設定できます。

カラム名	データ型	説明
jar_name	LONG VARCHAR	JAR ファイルの名前。
jar_file	LONG VARCHAR	このカラムは使用されなくなり、値は NULL になります。
update_time	TIMESTAMP	JAR ファイルが最後に更新されたローカル時刻。
update_time_utc	TIMESTAMP WITH TIME ZONE	JAR ファイルが最後に更新された UTC 時刻。

基本となるシステムテーブルに関する制約

PRIMARY KEY (jar_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

UNIQUE INDEX (jar_name)

SYSJARCOMPONENT システムビュー

SYSJAR システムビューの各ローは、JAR ファイルコンポーネントを定義します。このビューの基本となるシステムテーブルは ISYSJARCOMPONENT です。

カラム名	データ型	説明
component_id	INTEGER	コンポーネントの ID を含むプライマリキー。
jar_id	INTEGER	JAR の ID 番号を含むフィールド。
component_name	LONG VARCHAR	コンポーネント名。
component_type	CHAR(1)	このカラムは使用されなくなり、値は NULL になります。
contents	LONG BINARY	JAR ファイルのバイトコード。

基本となるシステムテーブルに関する制約

PRIMARY KEY (component_id)

FOREIGN KEY (jar_id) REFERENCES SYS.ISYSJAR (jar_id)

SYSJAVACLASS システムビュー

SYSJAVACLASS システムビューの各ローは、データベースに格納されている Java クラス 1 つを示します。このビューの基本となるシステムテーブルは ISYSJAVACLASS です。

カラム名	データ型	説明
class_id	INTEGER	Java クラスのユニークな番号。テーブルのプライマリキーでもあります。
object_id	UNSIGNED BIGINT	データベースのインデックスをユニークに識別する Java クラスの内部 ID。
creator	UNSIGNED INT	クラスの作成者のユーザ番号。INSTALL JAVA 文の AS USER 句で設定できます。
jar_id	INTEGER	クラスがある JAR ファイルの ID。
class_name	LONG VARCHAR	Java クラスの名前。
public	CHAR(1)	クラスがパブリック (Y) かプライベート (N) かを示します。
component_id	INTEGER	SYSJARCOMPONENT システムビューのコンポーネントの ID。
update_time	TIMESTAMP	クラスが最後に更新されたローカル時刻。
update_time_utc	TIMESTAMP WITH TIME ZONE	クラスが最後に更新された UTC 時刻。

基本となるシステムテーブルに関する制約

PRIMARY KEY (class_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)

FOREIGN KEY (component_id) REFERENCES SYS.ISYSJARCOMPONENT (component_id)

SYSLDAPSERVER システムビュー

ISYSLDAPSERVER システムテーブルに関する情報を読みやすい形式で表示します。

ISYSLDAPSERVER システムテーブルは、LDAP サーバの一連の属性を定義します。

カラム名	カラム型	説明
ldsrv_id	UNSIGNED BIGINT NOT NULL	LDAP サーバのユニークな識別子。プライマリキーであり、ログインポリシーから LDAP サーバを参照する際に使用する。
ldsrv_name	CHAR(128) NOT NULL	LDAP サーバに割り当てられた名前。
ldsrv_state	CHAR(9) NOT NULL	LDAP サーバの読み込み専用状態。 1 - RESET 2 - READY 3 - ACTIVE 4 - FAILED 5 - SUSPENDED 注意： 数値がシステムテーブルに格納され、対応するテキスト値がシステムビューに表示されます。
ldsrv_start_tls	TINYINT NOT NULL	LDAP サーバとの接続にトランスポートレイヤセキュリティ (TLS) を使用するかどうかを制御。これは TRUSTED_CERTIFICATE_FILE とともに使用され、LDAP サーバとの接続や検索に際して暗号化された通信を提供する。 有効な範囲：1 (オン) または 0 (オフ)。 デフォルト値は 0 (OFF)。
ldsrv_num_retries	TINYINT NOT NULL	LDAP サーバが失敗を返すかフェイルオーバを開始するまでに許可される認証試行回数を制御。 有効な範囲：1 ~ 60 デフォルト値は 3。
ldsrv_timeout	UNSIGNED INT NOT NULL	接続または検索のタイムアウト値 (単位はミリ秒) を制御。 有効な範囲：1 ~ 3600000 (1 時間) デフォルト値は 10000。
ldsrv_last_state_change	TIMESTAMP NOT NULL	最終状態変更発生時刻を示す。LDAP サーバのローカルのタイムゾーンに関係なく、この値は協定世界時 (UTC) で格納される。
ldsrv_search_url	CHAR(1024) NULL	ユーザ ID に基づいてユーザの識別名 (DN) を探すために使用される LDAP URL。

カラム名	カラム型	説明
ldsrv_auth_url	CHAR(1024) NULL	ユーザ ID が与えられたユーザの DN を探すために使用される LDAP 検索文字列。
ldsrv_access_dn	CHAR(1024) NULL	ユーザ ID の DN を検索して取得するために LDAP サーバにアクセスする際に使用される DN。
ldsrv_access_dn_pwd	VARBINARY(1024) NULL	アクセスアカウントのパスワード。このパスワードはディスクに保存されるときに対称的に暗号化される。

SYSLOGINMAP システムビュー

SYSLOGINMAP システムビューには、統合化ログインまたは Kerberos ログインを使用してデータベースに接続できる各ユーザに 1 つのローが含まれます。そのため、このビューへのアクセスは制限されています。このビューの基本となるシステムテーブルは ISYSLOGINMAP です。

カラム名	データ型	説明
login_mode	TINYINT	ログインの種類: 統合化ログインの場合は 1、Kerberos ログインの場合は 2。
login_id	VARCHAR(1024)	database_uid にマッピングする統合化ログインのユーザプロファイル名または Kerberos 原則。
object_id	UNSIGNED BIGINT	ユニークな識別子。ユーザ ID とデータベースのユーザ ID 間のマッピングごとに 1 つ。
database_uid	UNSIGNED INT	ログイン ID をマッピングするデータベースユーザ ID。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (login_mode, login_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (database_uid) REFERENCES SYS.ISYSUSER (user_id)
```

SYSLOGINPOLICY システムビュー

このビューの基本となるシステムテーブルは ISYSLOGINPOLICY です。

カラム名	データ型	説明
login_policy_id	UNSIGNED BIGINT	ログインポリシーのユニークな識別子。
login_policy_name	CHAR(128)	ログインポリシーの名前。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (login_policy_id)
FOREIGN KEY (login_policy_id) REFERENCES SYS.ISYSOBJECT (object_id)
UNIQUE INDEX (login_policy_name)
```

SYSLOGINPOLICYOPTION システムビュー

このビューの基本となるシステムテーブルは ISYSLOGINPOLICYOPTION です。

カラム名	データ型	説明
login_policy_id	UNSIGNED BIGINT	ログインポリシーのユニークな識別子。
login_option_name	CHAR(128)	ログインポリシーの名前。
login_option_value	LONG VARCHAR	ログインポリシーの作成時の値。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (login_policy_id, login_option_name)
FOREIGN KEY (login_policy_id) REFERENCES SYS.ISYSLOGINPOLICY
(login_policy_id)
```

SYSLOGINS ASE 互換ビュー

このビューは、DBO が所有します。SYSLOGINS の各ローは、有効な Adaptive Server の各ユーザアカウントです。

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またはテキストインデックスを作成した時点のオプション値に関する記述です。オプションの名前は、SYSMVOPTIONNAME システムビューにあります。このビューの基本となるシステムテーブルは ISYSMVOPTION です。

カラム名	データ型	説明
view_object_id	UNSIGNED BIGINT	マテリアライズドビューのオブジェクト ID。
option_id	UNSIGNED INT	データベースのオプションを識別するユニークな番号。オプション名を参照するには、SYSMVOPTIONNAME システムビューを参照してください。
option_value	LONG VARCHAR	マテリアライズドビューが作成された時点のオプション値。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (view_object_id, option_id)
```

```
FOREIGN KEY (view_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (option_id) REFERENCES SYS.ISYSMVOPTIONNAME (option_id)
```

SYSMVOPTIONNAME システムビュー

SYSMVOPTION システムビューの各ローには、マテリアライズドビューまたはテキストインデックスを作成した時点の名前オプション値があります。オプションの値は、SYSMVOPTION システムビューにあります。このビューの基本となるシステムテーブルは ISYSMVOPTIONNAME です。

カラム名	データ型	説明
option_id	UNSIGNED INT	データベースのオプションをユニークに識別する番号。
option_name	CHAR(128)	オプションの名前。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (option_id)
```

```
UNIQUE INDEX (option_name)
```

SYSOBJECT システムビュー

SYSOBJECT システムビューの各ローは、データベースオブジェクトを示します。このビューの基本となるシステムテーブルは ISYSOBJECT です。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	データベースのオブジェクトをユニークに識別するインデックスの内部 ID。
status	TINYINT	オブジェクトのステータス。次のような値があります。 <ul style="list-style-type: none"> • 1 (有効) – オブジェクトは、データベースサーバから使用できます。このステータスは、ENABLED と同等です。つまり、オブジェクトを ENABLE にすると、ステータスは VALID に変更されます。 • 2 (無効) – 内部操作後に、オブジェクトを再コンパイルする試みが失敗しました。たとえば、依存するオブジェクトを変更するスキーマの変更後などです。データベースサーバは、文で参照されるオブジェクトを再コンパイルする試みを継続します。 • 4 (無効化) – ユーザがオブジェクトを、明示的に無効化しました。たとえば、ALTER TABLE...DISABLE VIEW DEPENDENCIES 文を使用する場合などです。

カラム名	データ型	説明
object_type	TINYINT	オブジェクトの種類。
creation_time	TIMESTAMP	オブジェクトを作成したローカル日時。
object_type_str	CHAR(128)	オブジェクトの種類。
creation_time_UTC	TIMESTAMP WITH TIME ZONE	オブジェクトを作成した UTC 日時。

基本となるシステムテーブルに関する制約

PRIMARY KEY (object_id)

SYSOBJECTS ASE 互換ビュー

sysobjects には、テーブル、ビュー、ストアードプロシージャ、拡張ストアードプロシージャ、ログ、ルール、デフォルト、トリガ、検査制約、参照制約、計算カラム、関数ベースインデックスキー、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ごとに 1 つのローが含まれています。

このビューは、DBO が所有します。

さらに、オブジェクトタイプが N の場合は、パーティションの条件 ID ごとに 1 つのローが含まれます。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYS COMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSOPTION システムビュー

SYSOPTION システムビューには、データベースに格納されている各オプション設定のローのオプションが含まれます。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 ロールに対する設定は、自分の設定を持たないユーザが使うデフォルトの設定になります。このビューの基本となるシステムテーブルは ISYSOPTION です。

カラム名	データ型	説明
user_id	UNSIGNED INT	このオプション設定が適用されるユーザ番号。
"option"	CHAR(128)	オプションの名前。
"setting"	LONG VARCHAR	現在のオプションの設定。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (user_id, "option")
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

SYSOPTIONS 統合ビュー

SYSOPTIONS ビューの各ローは、SET コマンドを使用して作成されているオプション1つを示します。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、ユーザ ID PUBLIC に対する設定は、自分の設定を持たないユーザが使うデフォルトの設定になります。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSOPTIONS" ( user_name, "option", setting )
as select u.user_name, opt."option", opt.setting
from SYS.ISYSOPTION as opt
join SYS.ISYSUSER as u on opt.user_id = u.user_id
```

SYSOPTSTAT システムビュー

SYSOPTSTAT システムビューは、コストモデルの調整情報を、ALTER DATABASE CALIBRATE 文で計算して格納します。このビューのコンテンツは内部使用のみです。sa_get_dtt システムプロシージャ経由でアクセスすることをおすすめします。このビューの基本となるシステムテーブルは ISYSOPTSTAT です。

カラム名	データ型	説明
stat_id	UNSIGNED INT	システムでのみ使用。
group_id	UNSIGNED INT	システムでのみ使用。
format_id	SMALLINT	システムでのみ使用。
data	LONG BINARY	システムでのみ使用。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (stat_id, group_id, format_id)
```

SYSPARTITION システムビュー

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) を示します。このビューの基本となるシステムテーブルは ISYSPARTITION 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する。
partition_object_id	unsigned bigint	各テーブルパーティションはそれ自体が 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る。
partition_values	long varchar	範囲分割またはリスト分割の分割基準を含む。 範囲分割の場合、値にはこのパーティションの上限が含まれる。 リスト分割の場合、値には ';' で区切られた値リストが含まれる。位置はパーティションの序数。
position	unsigned int	パーティションの元の番号。 範囲パーティションの場合、2 以上の位置については、(位置 - 1) のパーティションに排他的下限が含まれる。
partition_name	char(128)	パーティションの名前

基本となるシステムテーブルでの制約

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

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。

```
ALTER VIEW "SYS"."SYSPARTITIONKEY"
as select * from SYS.ISYSPARTITIONKEY
```

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る。
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する。
position	smallint	パーティションキー内でのこのカラムの位置。0 から始まる。位置 0 はそのパーティションキーの 1 番目のカラム。

基本となるシステムテーブルでの制約

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。

SYSPARTITIONS システムビュー

ISYSPARTITIONS システムテーブルからのグループ情報を読みやすい形式で表示します。

SYSPARTITIONS 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) を示します。このビューの基本となるシステムテーブルは ISYSPARTITIONS です。

```
ALTER VIEW "SYS"."SYSPARTITIONS"
as select * from SYS.ISYSPARTITIONS
```

カラム名	データ型	説明
table_id	unsigned int	インデックスが対応するテーブルのオブジェクト 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する。
partition_object_id	unsigned bigint	各テーブルパーティションはそれ自体が 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る。
partition_dbpace_id	smallint	パーティションが配置されている DB 領域のオブジェクト ID。

カラム名	データ型	説明
partition_values	long varchar	この範囲パーティションの上限。
position	unsigned int	パーティションの序数。
partition_name	char(128)	パーティションの名前。

基本となるシステムテーブルでの制約:

プライマリキー (partitioned_object_id, partition_id)

外部キー (partitioned_object_id) が SYS.ISYSOBJECT を参照

外部キー (partition_object_id) が SYS.ISYSOBJECT を参照

SYSPARTITIONSCHEME システムビュー

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個々のオブジェクト (テーブルやインデックス) を示します。

```
ALTER VIEW "SYS"."SYSPARTITIONSCHEME"
as select * from SYS.ISYSPARTITIONSCHEME
```

カラム名	カラム型	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割り当てられている。
partition_method	tinyint	このテーブルの分割方法。有効な値：1 - 範囲分割、3 - ハッシュ分割 (2 は未使用)。
subpartition_method	tinyint	このテーブルのサブ分割方法。有効な値：NULL - サブ分割なし、1 - 範囲分割、3 - ハッシュ分割 (2 は未使用)。

基本となるシステムテーブルでの制約

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、データベースの物理インデックスを定義します。このビューの基本となるシステムテーブルは ISYSphysIDX です。

カラム名	データ型	説明
table_id	UNSIGNED INT	インデックスが対応するテーブルのオブジェクト ID。
phys_index_id	UNSIGNED INT	テーブル内の物理インデックスのユニークな番号。
root	INTEGER	データベースファイルにおける物理インデックスのルートページの位置を識別します。
key_value_count	UNSIGNED INT	インデックス内の個別のキー値の数。
leaf_page_count	UNSIGNED INT	リーフインデックスページの数。
depth	UNSIGNED SMALLINT	物理インデックスの深さ (レベル数)。
max_key_distance	UNSIGNED INT	システムでのみ使用。
seq_transitions	UNSIGNED INT	システムでのみ使用。
rand_transitions	UNSIGNED INT	システムでのみ使用。
rand_distance	UNSIGNED INT	システムでのみ使用。
allocation_bitmap	LONG VARBIT	システムでのみ使用。
long_value_bitmap	LONG VARBIT	システムでのみ使用。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (table_id, phys_index_id)
```

SYSprocAUTH 統合ビュー

SYSprocAUTH ビューの各ローは、プロシージャに付与されている一連の権限を示します。または、SYSprocPERM システムビューを使用することもできます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSprocAUTH" ( grantee,
  creator, procname )
  as select u1.user_name,u2.user_name,p.proc_name
  from SYS.ISYSPROCEDURE as p
  join SYS.ISYSprocPERM as pp on(p.proc_id = pp.proc_id)
  join SYS.ISYSUSER as u1 on u1.user_id = pp.grantee
  join SYS.ISYSUSER as u2 on u2.user_id = p.creator
```

SYSPROCEDURE システムビュー

SYSPROCEDURE システムビューの各ローは、データベース内のプロシージャを示します。このビューの基本となるシステムテーブルは ISYSPROCEDURE です。

カラム名	データ型	説明
proc_id	UNSIGNED INT	各プロシージャにはユニークな番号 (プロシージャ番号) が割り当てられます。
creator	UNSIGNED INT	プロシージャの所有者。
object_id	UNSIGNED BIGINT	データベースのプロシージャをユニークに識別するプロシージャの内部 ID。
proc_name	CHAR(128)	プロシージャ名。1 人の作成者が、同じ名前のプロシージャを 2 つ持つことはできません。
proc_defn	LONG VARCHAR	プロシージャの定義。
remarks	LONG VARCHAR	プロシージャに関する注記。ISYSREMARK システムテーブル内に格納されている値。
replicate	CHAR(1)	このプロパティは内部でのみ使用されます。
srvid	UNSIGNED INT	プロシージャがリモートデータベースサーバ上のプロシージャのプロキシである場合は、そのリモートサーバを表します。
source	LONG VARCHAR	回避されたプロシージャのソース。この値は、ISYSSOURCE システムテーブル内に格納されています。
avg_num_rows	FLOAT	プロシージャが FROM 句に表示されるときに、クエリの最適化に使用するために収集される情報。
avg_cost	FLOAT	プロシージャが FROM 句に表示されるときに、クエリの最適化に使用するために収集される情報。
stats	LONG BINARY	プロシージャが FROM 句に表示されるときに、クエリの最適化に使用するために収集される情報。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (proc_id)
```

```
FOREIGN KEY (srvid) REFERENCES SYS.ISYSSERVER (srvid)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (proc_name, creator)
```

SYSPROC システムビュー

SYSPROC システムビューの各ローは、データベース内のプロシージャに対するパラメータ 1 つを示します。このビューの基本となるシステムテーブルは **ISYSPROC** です。

カラム名	データ型	説明
proc_id	UNSIGNED INT	このパラメータが属するプロシージャをユニークに識別します。
parm_id	SMALLINT	各プロシージャは、パラメータに 1 から順に番号を付けます。パラメータ番号の順は、定義された順になっています。関数の場合、最初のパラメータの内容は関数名であり、その関数の戻り値を表します。
parm_type	SMALLINT	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"> • 0 - 標準のパラメータ (変数) • 1 - 結果変数 - 結果セットを返すプロシージャで使用 • 2 - SQLSTATE エラー値 • 3 - SQLCODE エラー値 • 4 - 関数からの戻り値
parm_mode_in	CHAR(1)	このパラメータが、プロシージャに値を提供するかどうかを示します (IN または INOUT パラメータ)。
parm_mode_out	CHAR(1)	このパラメータが、プロシージャからの値 (IN または INOUT パラメータ) と RESULT 句のカラムのどちらを返すかを示します。
domain_id	SMALLINT	SYSDOMAIN システムビューにリストされたデータ型番号から、パラメータのデータ型を識別します。
width	BIGINT	文字列パラメータでは長さ、数値パラメータでは精度、その他のデータ型では記憶領域のサイズをバイトで示します。
scale	SMALLINT	数値データ型の場合、小数点以下の桁数です。その他のデータ型の場合、このカラムの値は 1 です。
user_type	SMALLINT	パラメータのユーザ型 (適用できる場合)。
parm_name	CHAR(128)	プロシージャパラメータの名前。
"default"	LONG VARCHAR	パラメータのデフォルト値。参照情報としてのみ表示されます。

カラム名	データ型	説明
remarks	LONG VARCHAR	常に NULL を返します。ODBC ドライバの旧バージョンを新しいパーソナルデータベースサーバで使用できるようにするために用意されています。
base_type_str	VARCHAR(32767)	パラメータの物理的な型を表す注釈付きの型文字列。

基本となるシステムテーブルに関する制約

PRIMARY KEY (proc_id, parm_id)

FOREIGN KEY (proc_id) REFERENCES SYS.ISYSPROCEDURE (proc_id)

FOREIGN KEY (domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)

FOREIGN KEY (user_type) REFERENCES SYS.ISYSUSERTYPE (type_id)

SYSROCPARMS 統合ビュー

SYSROCPARMS ビューの各ローは、データベース内のプロシージャに対するパラメータを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSROCPARMS" ( creator,
    procname, parmname, parm_id, parmtype, parmmode, parmdomain,
    length, scale, "default", user_type )
as select
up.user_name, p.proc_name, pp.parm_name, pp.parm_id, pp.parm_type,
    if pp.parm_mode_in = 'Y' and pp.parm_mode_out = 'N' then 'IN'
    else if pp.parm_mode_in = 'N' and pp.parm_mode_out = 'Y' then
'OUT'
    else 'INOUT'
    endif
endif, dom.domain_name, pp.width, pp.scale, pp."default", ut.type_name
from SYS.SYSROCPARM as pp
    join SYS.ISYSPROCEDURE as p on p.proc_id = pp.proc_id
    join SYS.ISYSUSER as up on up.user_id = p.creator
    join SYS.ISYSDOMAIN as dom on dom.domain_id = pp.domain_id
left outer join SYS.ISYSUSERTYPE as ut on ut.type_id =
pp.user_type
```

SYSROCPERM システムビュー

SYSROCPERM システムビューの各ローは、プロシージャに対する EXECUTE 権限を付与されているユーザを表します。このビューの基本となるシステムテーブルは ISYSROCPERM です。

カラム名	データ型	説明
proc_id	UNSIGNED INT	プロシージャ番号は、EXECUTE 権限が付与されたプロシージャをユニークに特定します。
grantee	UNSIGNED INT	権限の被付与者のユーザ番号。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (proc_id, grantee)
```

```
FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (proc_id) REFERENCES SYS.ISYSPROCEDURE (proc_id)
```

SYSROCS 統合ビュー

SYSROCS ビューには、プロシージャまたはファンクションに記録されているプロシージャ名、ファンクション名、作成者名、コメントが表示されます。

ビューを構成するテーブルとカラムは、以下の ALTER VIEW 文で示されます。

```
ALTER VIEW "SYS"."SYSROCS"( creator,
    procname,remarks )
as select u.user_name,p.proc_name,r.remarks
    from SYS.ISYSPROCEDURE as p
        join SYS.ISYSUSER as u on u.user_id = p.creator
        left outer join SYS.ISYSREMARK as r on(p.object_id =
r.object_id)
```

SYSROXYTAB システムビュー

SYSROXYTAB システムビューの各ローは、プロキシテーブルのリモートパラメータを示します。このビューの基本となるシステムテーブルは ISYSROXYTAB です。

カラム名	データ型	説明
table_object_id	UNSIGNED BIGINT	プロキシテーブルのオブジェクト ID。
existing_obj	CHAR(1)	プロキシテーブルがリモートサーバに以前に存在していたかどうかを示します。
srvid	UNSIGNED INT	プロキシテーブルと関連付けられたリモートサーバのユニークな ID。

カラム名	データ型	説明
remote_location	LONG VARCHAR	リモートサーバに関するプロキシテーブルの位置。

基本となるシステムテーブルに関する制約

PRIMARY KEY (table_object_id)

FOREIGN KEY (table_object_id) REFERENCES ISYSOBJECT (object_id)
MATCH UNIQUE FULL

FOREIGN KEY (srvid) REFERENCES SYS.ISYSSERVER (srvid)

SYSPUBLICATION システムビュー

SYSPUBLICATION システムビューの各ローは、パブリケーションを示します。このビューの基本となるシステムテーブルは ISYSPUBLICATION です。

カラム名	データ型	説明
publication_id	UNSIGNED INT	パブリケーションをユニークに識別する番号。
object_id	UNSIGNED BIGINT	データベースのパブリケーションをユニークに識別するプロシージャの内部 ID。
creator	UNSIGNED INT	パブリケーションの所有者。
publication_name	CHAR(128)	パブリケーションの名前。
remarks	LONG VARCHAR	パブリケーションに関する注記。ISYSREMARK システムテーブル内に格納されている値。
type	CHAR(1)	このカラムは使用されません。
sync_type	UNSIGNED INT	<p>パブリケーションの同期の種類。次のような値があります。</p> <ul style="list-style-type: none"> • 0 (ログスキャン) – トランザクションログを使用して、最後のアップロード後に変更されたすべての関連データをアップロードする通常のパブリケーションです。 • 1 (スクリプト化されたアップロード) – このパブリケーションの場合、トランザクションログは無視され、アップロードは格納済みプロシージャを使用してユーザが定義します。ストアドプロシージャに関する情報は、ISYSSYNCSCRIPT システムテーブルに格納されます。 • 2 (ダウンロード専用) – これはダウンロードのみのパブリケーションです。データはアップロードされません。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (publication_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (publication_name, creator)
```

SYSPUBLICATIONS 統合ビュー

SYSPUBLICATIONS ビューの各ローは、パブリケーションを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSPUBLICATIONS"  
as select u.user_name as creator,  
p.publication_name,  
r.remarks,  
p.type,  
case p.sync_type  
when 0 then 'logscan'  
when 1 then 'scripted upload'  
when 2 then 'download only'  
else 'invalid'  
end as sync_type  
from SYS.ISYSPUBLICATION as p  
join SYS.ISYSUSER as u on u.user_id = p.creator  
left outer join SYS.ISYSREMARK as r on(p.object_id =  
r.object_id)
```

SYREMARK システムビュー

SYREMARK システムビューの各ローは、オブジェクトの注釈(コメント)を示します。このビューの基本となるシステムテーブルは ISYSREMARK です。

カラム	データ型	説明
object_id	UNSIGNED BIGINT	関連する注釈があるオブジェクトの内部 ID。
remarks	LONG VARCHAR	オブジェクトと関連付けられた注釈またはコメント。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (object_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

SYSREMOPTION システムビュー

SYSREMOPTION システムビューの各ローは、メッセージリンクパラメータの値を示します。このビューの基本となるシステムテーブルは ISYSREMOPTION です。

このビューの一部のカラムには、機密データが含まれている可能性があります。SYSREMOPTION2 ビューを使用すると、機密データを含む可能性のあるカラムを除いて、このビューのデータにパブリックアクセスできます。

カラム	データ型	説明
option_id	UNSIGNED INT	メッセージリンクパラメータの ID 番号。
user_id	UNSIGNED INT	パラメータが設定されているユーザ ID。
"setting"	VARCHAR(255)	メッセージリンクパラメータの値。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (option_id, user_id)
```

```
FOREIGN KEY (option_id) REFERENCES SYS.ISYSREMOPTIONTYPE (option_id)
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

SYSREMOPTION2 統合ビュー

SYSREMOPTION および SYSREMOPTIONTYPE システムビューのカラムを一緒に結合して、読みやすい形式で表します。

SELECT ANY TABLE システム権限のないユーザの場合、設定されているカラムの値は非表示になります。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSREMOPTION2"
as select ISYSREMOPTION.option_id,
ISYSREMOPTION.user_id,
SYS.HIDE_FROM_NON_DBA(ISYSREMOPTION.setting) as setting
from SYS.ISYSREMOPTION
```


SYSREMOPTIONS 統合ビュー

SYSREMOPTIONS ビューの各ローは、メッセージリンクパラメータの値を示します。

SELECT ANY TABLE システム権限のないユーザの場合、設定されているカラムの値は非表示になります。SYSREMOPTION2 ビューを使用すると、機密データ以外のデータにパブリックアクセスできます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSREMOPTIONS"
  as select srt.type_name,
    sup.user_name,
    srot."option",
    SYS.HIDE FROM NON_DBA(sro.setting) as setting
  from SYS.ISYSREMOPTIONTYPE as srt
    ,SYS.ISYSREMOPTIONTYPE as srot
    ,SYS.ISYSREMOPTION as sro
    ,SYS.ISYSUSER as sup
  where srt.type_id = srot.type_id
  and srot.option_id = sro.option_id
  and sro.user_id = sup.user_id
```

SYSREMOPTIONTYPE システムビュー

SYSREMOPTIONTYPE システムビューの各ローは、メッセージリンクパラメータ 1 つを示します。このビューの基本となるシステムテーブルは ISYSREMOPTIONTYPE です。

カラム	データ型	説明
option_id	UNSIGNED INT	メッセージリンクパラメータの ID 番号。
type_id	SMALLINT	このパラメータを使用するメッセージタイプの ID 番号。
option	VARCHAR(128)	メッセージリンクパラメータの名前。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (option_id)
```

```
FOREIGN KEY (type_id) REFERENCES SYS.ISYSREMOPTIONTYPE (type_id)
```

SYSREMOTETYPE システムビュー

このビューの基礎となるシステムテーブルは **ISYSREMOTETYPE** です。

カラム名	データ型	説明
type_id	SMALLINT	サポートされているメッセージシステムのうち、このユーザにメッセージを送信するために使用されているものを識別します。
object_id	UNSIGNED BIGINT	データベースのリモートタイプをユニークに識別するインデックスの内部 ID。
type_name	CHAR(128)	メッセージシステムの名前。
publisher_address	LONG VARCHAR	リモートデータベースパブリッシャのアドレス。
remarks	LONG VARCHAR	リモートタイプに関する注釈。ISYSREMARK システムテーブル内に格納されている値。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (type_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
UNIQUE INDEX (type_name)
```

SYSREMOTETYPES 統合ビュー

SYSREMOTETYPES ビューの各ローは、パブリッシャアドレスなど、メッセージタイプ 1 つを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSREMOTETYPES"  
as select rt.type_id,rt.type_name,rt.publisher_address,rm.remarks  
from SYS.ISYSREMOTETYPE as rt  
left outer join SYS.ISYSREMARK as rm on(rt.object_id =  
rm.object_id)
```

SYSREMOTEUSER システムビュー

SYSREMOTEUSER システムビューの各ローは、REMOTE システム権限を持つユーザ ID (サブスクライバ) を示します。そのユーザに送信されたメッセージと、そのユーザから送信されたメッセージのステータスも合わせて示します。このビューの基本となるシステムテーブルは **ISYSREMOTEUSER** です。

カラム名	データ型	説明
user_id	UNSIGNED INT	REMOTE 権限を持つユーザのユーザ番号。
consolidate	CHAR(1)	ユーザに CONSOLIDATE 権限 (Y) か REMOTE 権限 (N) が付与されていることを表します。
type_id	SMALLINT	メッセージシステムのうち、このユーザにメッセージを送信するために使用されているものを識別します。
address	LONG VARCHAR	メッセージが送信されるアドレス。アドレスは address_type に対して適切でなければいけません。
frequency	CHAR(1)	メッセージが送信される頻度。
send_time	TIME	次にメッセージがこのユーザへ送信される時刻。
log_send	UNSIGNED BIGINT	メッセージは、log_send が log_sent よりも大きいサブスクライバだけに送信されます。
time_sent	TIMESTAMP	このサブスクライバに、最後にメッセージが送信されたローカル時刻。
log_sent	UNSIGNED BIGINT	最後に送信されたオペレーションのログオフセット。
confirm_sent	UNSIGNED BIGINT	このサブスクライバから最後に確認されたオペレーションに対する、ログオフセット。
send_count	INTEGER	メッセージが送信された回数。
resend_count	INTEGER	メッセージがサブスクライバデータベースに、一度だけ適用されたことを確認するためのカウンタ。
time_received	TIMESTAMP	このサブスクライバから、最後にメッセージが受信されたローカル時刻。
log_received	UNSIGNED BIGINT	現在のデータベースで最後に受信されたオペレーションに対する、サブスクライバのデータベース内のログオフセット。
confirm_received	UNSIGNED BIGINT	最後に確認メッセージが送信されたオペレーションに対する、サブスクライバのデータベース内のログオフセット。
receive_count	INTEGER	メッセージが受信された回数。
rereceive_count	INTEGER	メッセージが現在のデータベースに、一度だけ適用されたことを確認するためのカウンタ。
time_sent_utc	TIMESTAMP WITH TIME ZONE	このサブスクライバに、最後にメッセージが送信された UTC 時刻。

カラム名	データ型	説明
time_received_utc	TIMESTAMP WITH TIME ZONE	このサブスクライバから、最後にメッセージが受信された UTC 時刻。

基本となるシステムテーブルに関する制約

PRIMARY KEY (user_id)

FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)

FOREIGN KEY (type_id) REFERENCES SYS.ISYSREMOTETYPE (type_id)

UNIQUE INDEX (type_id, address)

SYSREMOTEUSERS 統合ビュー

SYSREMOTEUSERS ビューの各ローは、REMOTE システム権限を持つユーザ ID (サブスクライバ) を示します。そのユーザに送信した、またはそのユーザから送信されたメッセージのステータスも合わせて示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSREMOTEUSERS" AS SELECT u.user_name,
r.consolidate, t.type_name, r.address, r.frequency, r.send_time,
(if r.frequency = 'A' then NULL
else if r.frequency = 'P' then
if r.time_sent IS NULL then CURRENT TIMESTAMP
else (select min( minutes( dateadd(mi,
PROPERTY('TimeZoneAdjustment'), a.time_sent),
60*hour(a.send_time) + minute( seconds( a.send_time, 59 ) ) ) )
FROM SYS.ISYSREMOTEUSER a WHERE a.frequency = 'P' AND a.send_time =
r.send_time ) endif
else if CURRENT DATE + r.send_time > coalesce( dateadd(mi,
PROPERTY('TimeZoneAdjustment'), r.time_sent), CURRENT TIMESTAMP)
then CURRENT DATE + r.send_time
else CURRENT DATE + r.send_time + 1
endif endif endif) as next_send, r.log_send ,
dateadd(mi, PROPERTY('TimeZoneAdjustment'), r.time_sent)
as time_sent , r.log_sent, r.confirm_sent, r.send_count,
r.resend_count,
dateadd(mi, PROPERTY('TimeZoneAdjustment'), r.time_received) as
time_received ,
r.log_received, r.confirm_received, r.receive_count,
r.rereceive_count ,
TODATETIMEOFFSET( r.time_sent, 0 ) as time_sent_utc ,
TODATETIMEOFFSET( r.time_received, 0 ) as time_received_utc
FROM SYS.ISYSREMOTEUSER r JOIN SYS.ISYSUSER u ON ( u.user_id =
r.user_id ) JOIN SYS.ISYSREMOTETYPE t ON ( t.type_id = r.type_id )
```

SYSROLEGRANTEXT システムビュー

SYSROLEGRANTEXT システムビューには、SET USER および CHANGE PASSWORD システム権限に関する構文拡張が格納されています。このシステムビューは SYSROLEGRANT システムビューに関連しています。

カラム名	データ型	説明
grant_id	UNSIGNED INT	各 GRANT 文の識別に使用する ID。
user_id	UNSIGNED INT	user-list または role-list で特定の拡張付与に基づいて指定された user_ids。

user-list オプションまたは ANY WITH ROLES role-list オプションを使用して、SET USER 権限や CHANGE PASSWORD 権限を付与したり取り消すときは、このビューが拡張構文の値によって更新されます。

基本となるシステムテーブルに関する制約:
プライマリキー (grant_id、user_id)

SYSROLEGRANT システムビュー

SYSSEQUENCE システムビューには、システム定義ロールまたはユーザ定義ロールの付与ごとに 1 つのローがあります。このビューの基礎となるシステムテーブルは、ISYSROLEGRANT です。

カラム名	データ型	説明
grant_id	UNSIGNED INT	各 GRANT 文の識別に使用する ID。
role_id	UNSIGNED INT	ISYSUSER に基づいて、付与されるロールの ID。
grantee	UNSIGNED INT	ISYSUSER に基づいて、ロールを付与されるユーザの ID。

カラム名	データ型	説明
grant_type	TINYINT	<p>3桁を使用して付与のタイプを表す。1桁目は、権限が付与されているかどうかを示す。2桁目は、管理権限が付与されているかどうかを示す。3桁目は、システム権限が継承可能かどうかを示す。</p> <ul style="list-style-type: none"> • 001 – 権限が付与されているが、継承可能ではなく、管理権限もない。SYS_AUTH_DBA_ROLE と SYS_AUTH_REMOVE_DBA_ROLE を除く、継承不可のレガシー権限にのみ適用。 • 101 – 権限が付与されており、継承可能であるが、管理権限はない。 • 110 – 管理権限のみが付与されている。 • 111 – 権限が付与されており、継承可能で、管理権限がある。 • 001 – 権限が付与されており、管理権限はあるが、継承可能ではない。レガシー権限の SYS_AUTH_DBA_ROLE と SYS_AUTH_REMOVE_DBA_ROLE にのみ適用。
grant_scope	TINYINT	<p>SET USER および CHANGE PASSWORD によって使用され、付与のスコップを設定する。値は、次のうちの1つまたは複数。</p> <ul style="list-style-type: none"> • 001 – ユーザリスト • 010 – ANY WITH ROLES • 110 – ANY
grantor	CHAR (128)	ロールの付与者のユニークな識別子。

基本となるシステムテーブルに関する制約:

プライマリキー (grant_ID)

ユニークインデックス (role_id、grantee、grant_scope)

SYSROLEGRANTS システムビュー

SYSROLEGRANTS システムビューは SYSROLEGRANT システムビューと同じですが、(ロール ID だけでなく) ロールの名前と (ユーザ ID だけでなく) 付与者の名前という 2 つの追加カラムが含まれています。

カラム名	データ型	説明
grant_id	UNSIGNED INT	発行された各 GRANT 文のユニークな識別子。
role_id	UNSIGNED INT	(ISYSUSER テーブルで定義されている) ユーザに付与されたロールのユニークな識別子。
role_name	CHAR(128)	role_id 値に対応するロールの名前。
grantee	UNSIGNED INT	ロールを付与されたユーザのユニークな識別子。
grantee_name	CHAR(128)	grantee 値に対応する被付与者の名前。
grant_type	TINYINT	<p>ロールとその基本となる権限がどのように付与されたかを示す。次のような値がある。</p> <ul style="list-style-type: none"> • 1 - 基本となる権限が管理権限なし、権限継承なしで付与されている。 <hr/> <p>注意： この値は、SYS_AUTH_DBA_ROLE と SYS_AUTH_REMOVE_DBA_ROLE を除く、継承不可のすべてのレガシーロールに適用されます。</p> <ul style="list-style-type: none"> • 3 - 基本となる権限が管理権限あり、権限継承なしで付与されている。 <hr/> <p>注意： この値は、継承不可のレガシーロール SYS_AUTH_DBA_ROLE と SYS_AUTH_REMOVE_DBA_ROLE にのみ適用されます。</p> <ul style="list-style-type: none"> • 5 - 基本となる権限が管理権限なし、権限継承ありで付与されている。 • 6 - 基本となる権限に対する管理権限のみが付与されている。 • 7 - 基本となる権限が管理権限あり、権限継承ありで付与されている。 •

カラム名	データ型	説明
grant_scope	TINYINT	付与が適用される範囲を定義する。次のような値がある。 <ul style="list-style-type: none"> 1 - ユーザリスト 2 - 指定したロールのメンバーシップを付与された任意のユーザ 3 - すべてのユーザ <hr/> 注意： この値は SET USER および CHANGE PASSWORD システム権限にのみ適用され、これらの値の有効な任意の組み合わせを格納できます。
grantor	CHAR (128)	ロールの付与者のユニークな識別子。

SYSSCHEDULE システムビュー

SYSSCHEDULE システムビューの各ローは、CREATE EVENT 文の SCHEDULE 句に指定されたイベントの起動時刻を示します。このビューの基本となるシステムテーブルは ISYSSCHEDULE です。

カラム名	データ型	説明
event_id	UNSIGNED INT	各イベントに割り当てられたユニークな番号。
sched_name	VARCHAR(128)	イベントのスケジュールと関連付けられた名前。
recurring	TINYINT	スケジュールを反復するかどうかを示します。
start_time	TIME	スケジュールの起動時刻。
stop_time	TIME	BETWEEN を指定した場合、スケジュールの停止時刻を示します。
start_date	DATE	イベントの実行がスケジュールされている最初の日付。
days_of_week	TINYINT	イベントがスケジュールされている曜日を示すビットマスク。 <ul style="list-style-type: none"> x01 = 日曜日 x02 = 月曜日 x04 = 火曜日 x08 = 水曜日 x10 = 木曜日 x20 = 金曜日 x40 = 土曜日

カラム名	データ型	説明
days_of_month	UNSIGNED INT	イベントがスケジュールされている日付を示すビットマスク。いくつかの例を示します。 x01 = 1 日 x02 = 2 日 x40000000 = 31 日 x80000000 = その月の最終日
interval_units	CHAR(10)	EVERY で指定される間隔単位。 HH = 時間 NN = 分 SS = 秒
interval_amt	INTEGER	EVERY で指定される期間。

基本となるシステムテーブルに関する制約

PRIMARY KEY (event_id, sched_name)

FOREIGN KEY (event_id) REFERENCES SYS.ISYSEVENT (event_id)

SYSSERVER システムビュー

SYSSERVER システムビューの各ローは、リモートサーバを示します。このビューの基本となるシステムテーブルは ISYSSERVER です。

注意： SYSSERVERS システムテーブルに含まれる前のカタログバージョン。このテーブル名は ISYSSERVER ('S' なし) に変更され、このビューの基本となるテーブルになります。

カラム名	データ型	説明
srvid	UNSIGNED INT	リモートサーバの識別子。
srvname	VARCHAR(128)	リモートサーバの名前。
srvclass	LONG VARCHAR	CREATE SERVER 文で指定されたサーバクラス。
srvinfo	LONG VARCHAR	サーバ情報。
srvreadonly	CHAR(1)	サーバが読み込み専用かどうか。

基本となるシステムテーブルに関する制約

PRIMARY KEY (srvid)

SYSSOURCE システムビュー

SYSSOURCE システムビューの各ローには、SYSOBJECT システムビューにリストされているオブジェクトのソースコード (適用できる場合) が含まれます。このビューの基本となるシステムテーブルは ISYSSOURCE です。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	ソースコードが定義されているオブジェクトの内部 ID。
source	LONG VARCHAR	オブジェクトを作成したときに preserve_source_format データベースオプションが On の場合、このカラムにはオブジェクトの元のソースコードが含まれます。

基本となるシステムテーブルに関する制約

PRIMARY KEY (object_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

SYSSPATIALREFERENCESYSTEM システムビュー

SYSSPATIALREFERENCESYSTEM システムビューの各ローは、データベースに定義されている SRS に関する記述です。このビューの基礎となるシステムテーブルは ISYSSPATIALREFERENCESYSTEM です。

このビューには、ST_SPATIAL_REFERENCE_SYSTEMS システムビューとは若干異なる量の情報があります。

注意： 空間データ、空間参照系、および空間測定単位は、カタログストアでのみ使用できます。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	システムでのみ使用。
owner	UNSIGNED INT	SRS の所有者。
srs_name	CHAR(128)	SRS の名前。
srs_id	INTEGER	空間参照系の数値識別子 (SRID)。
round_earth	CHAR(1)	SRS タイプが ROUND EARTH (Y) または PLANAR (N) のいずれであるか。

カラム名	データ型	説明
axis_order	CHAR(12)	データベースサーバが緯度と経度に関連してポイントを解釈する方法を記述(たとえば、ST_Lat メソッドと ST_Long メソッドを使用する場合)。非地理的空間参照系の場合、軸順序は x/y/z/m。地理的空間参照系の場合、デフォルトの軸順序は long/lat/z/m であり、lat/long/z/m もサポートされる。
snap_to_grid	DOUBLE	計算の実行時に使用されるグリッドのサイズを定義。
tolerance	DOUBLE	ポイントを比較するときに使用する精度を定義。
semi_major_axis	DOUBLE	ROUND EARTH SRS の楕円の中心から赤道までの距離。
semi_minor_axis	DOUBLE	ROUND EARTH SRS の楕円の中心から両極までの距離。
inv_flattening		ROUND EARTH SRS の楕円に使用される逆扁平率。 逆扁平率 (f) は、回転楕円体の極から赤道への扁平度を定義する算術値。値の範囲は、扁平なし(完全な円)から完全な扁平(直線)まで。逆扁平率は、次に示すように値 1/f。 $1/f = (\text{semi_major_axis}) / (\text{semi_major_axis} - \text{semi_minor_axis})$
min_x	DOUBLE	座標において許される x の最小値。
max_x	DOUBLE	座標において許される x の最大値。
min_y	DOUBLE	座標において許される y の最小値。
max_y	DOUBLE	座標において許される y の最大値。
min_z	DOUBLE	座標において許される z の最小値。
max_z	DOUBLE	座標において許される z の最大値。
min_m	DOUBLE	座標において許される m の最小値。
max_m	DOUBLE	座標において許される m の最大値。
organization	LONG VARCHAR	空間参照系によって使用される座標系を作成した組織の名前。

カラム名	データ型	説明
organization_coord_sys_id	INTEGER	座標系を作成した組織が座標系に対して指定した ID。
srs_type	CHAR(11)	SQL/MM 標準による定義に従った SRS のタイプ。次のいずれかの値を取る。 <ul style="list-style-type: none"> • GEOGRAPHIC – 緯度と経度 (および標高) の軸を持つ、測地参照される座標系に基づく SRS 用。これらの SRS は、PLANAR タイプまたは ROUND EARTH タイプ。 • PROJECTED – 緯度と経度の軸を持たず、測地参照される座標系に基づく SRS 用。これらの SRS は PLANAR タイプ。 • ENGINEERING – 測地参照されない座標系に基づく SRS 用。これらの SRS は PLANAR タイプ。 • GEOCENTRIC – サポートされない。 • COMPOUND – サポートされない。 • VERTICAL – サポートされない。 srs_type が空の場合、タイプは指定されない。
linear_unit_of_measure	UNSIGNED BIGINT	空間参照系によって使用される線形測定単位。
angular_unit_of_measure	UNSIGNED BIGINT	空間参照系によって使用される角度測定単位。
count_in_use	UNSIGNED BIGINT	内部でのみ使用。
polygon_format	LONG VARCHAR	多角形のリングの方向。CounterClockwise、ClockWise、または EvenOdd のいずれか。
storage_format	LONG VARCHAR	データが正規化フォーマット (Internal)、非正規化フォーマット (Original)、または両方 (Mixed) のいずれかで格納されるか。
definition	LONG VARCHAR	OGC 標準によって定義されているフォーマットでの、空間参照系の WKT 定義。
transform_definition	LONG VARCHAR	この SRS から別の SRS にデータを変換するときを使用される変換定義設定。

基本となるシステムテーブルに関する制約:

```

PRIMARY KEY (object_id)
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id)
FOREIGN KEY (linear_unit_of_measure) REFERENCES
SYS.ISYSUNITOFMEASURE (object_id)
FOREIGN KEY (angular_unit_of_measure) REFERENCES
SYS.ISYSUNITOFMEASURE (object_id)
FOREIGN KEY (owner) REFERENCES SYS.ISYSUSER (user_id)
UNIQUE CONSTRAINT (srs_name)
UNIQUE CONSTRAINT (srs_id)

```

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。このビューの基本となるシステムテーブルは ISYSSQLSERVERTYPE です。

カラム名	データ型	説明
ss_user_type	SMALLINT	Adaptive Server Enterprise ユーザ型。
ss_domain_id	SMALLINT	Adaptive Server Enterprise のドメイン ID。
ss_type_name	VARCHAR(30)	Adaptive Server Enterprise 型名。
primary_sa_domain_id	SMALLINT	対応する SAP Sybase IQ のプライマリドメイン ID。
primary_sa_user_type	SMALLINT	対応する SAP Sybase IQ のプライマリユーザタイプ。

基本となるシステムテーブルに関する制約

```

PRIMARY KEY (ss_user_type)

```

SYSSUBPARTITIONKEY システムビュー

ISYSSUBPARTITIONKEY システムテーブルからのグループ情報を読みやすい形式で表示します。

SYSSUBPARTITIONKEY システムビューには、ISYSPARTITIONSCHEME ビューで記述されている分割テーブル内の、ISYSPARTITION ビューで記述されているパーティションの各カラムのローが含まれます。

カラム名	データ型	説明
partitioned_object_id	unsigned bigint	分割されたオブジェクト (テーブルまたはインデックス) に割り当てられたユニークな ID。

カラム名	データ型	説明
column_id	unsigned int	パーティションキーの一部であるテーブルカラムを識別する。partitioned_object_id と column_id の両方で、SYSTABCOL システムビューで記述されている 1 つのカラムを識別する。
position	smallint	パーティションキー内でのカラムの位置。値 0 はパーティションキーの 1 番目のカラム。値 1 はパーティションキーの 2 番目のカラム。

基本となるシステムテーブルでの制約:

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

SYSSUBSCRIPTION システムビュー

SYSSUBSCRIPTION システムビューの各ローは、REMOTE システム権限を持つあるユーザ ID からの、あるパブリケーションに対するサブスクリプションに関する記述です。このビューの基本となるシステムテーブルは ISYSSUBSCRIPTION です。

カラム名	データ型	説明
publication_id	UNSIGNED INT	ユーザ ID のサブスクリプションが作成されているパブリケーションの識別子。
user_id	UNSIGNED INT	パブリケーションに対して送信されたユーザの ID。
subscribe_by	CHAR(128)	サブスクリプション用の SUBSCRIBE BY 式がある場合、その値。
created	UNSIGNED BIGINT	トランザクションログ内の、サブスクリプションが作成されたオフセット。
started	UNSIGNED BIGINT	トランザクションログ内の、サブスクリプションが開始されたオフセット。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (publication_id, user_id, subscribe_by)
```

```
FOREIGN KEY (publication_id) REFERENCES SYS.ISYSPUBLICATION (publication_id)
```

```
FOREIGN KEY (user_id) REFERENCES SYS.ISYSUSER (user_id)
```

SYSSUBSCRIPTIONS 統合ビュー

REMOTE システム権限を持つあるユーザ ID からの、あるパブリケーションに対するサブスクリプションについて、各ローで示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSSUBSCRIPTIONS"
as select p.publication_name,u.user_name,s.subscribe_by,s.created,
s.started
from SYS.ISYSSUBSCRIPTION as s
join SYS.ISYSPUBLICATION as p on(p.publication_id =
s.publication_id)
join SYS.ISYSUSER as u on u.user_id = s.user_id
```

SYSSYNC システムビュー

SYSSYNC システムビューには、同期に関する情報が含まれます。このビューの一部のカラムには、機密データが含まれている可能性があります。そのため、このビューへのアクセスは制限されています。SYSSYNC2 ビューを使用すると、機密データを含む可能性のあるカラムを除いて、このビューのデータにパブリックアクセスできます。このビューの基本となるシステムテーブルは ISYSSYNC です。

カラム名	データ型	説明
sync_id	UNSIGNED INT	ユニークにローを識別する番号。
type	CHAR(1)	値は常に D です。
publication_id	UNSIGNED INT	SYSPUBLICATION システムビューに入っている publication_id。
progress	UNSIGNED BIGINT	最後に成功したアップロードのログオフセット。
site_name	CHAR(128)	ユーザ名。
"option"	LONG VARCHAR	同期オプション。
server_connect	LONG VARCHAR	サーバのアドレスまたは URL。
server_conn_type	LONG VARCHAR	同期時に使用する、TCP/IP などの通信プロトコル。
last_download_time	TIMESTAMP	最後にサーバからダウンロードストリームを受信した時刻。
last_upload_time	TIMESTAMP	最後に情報のアップロードが成功した時刻 (サーバで測定)。デフォルトは jan-1-1900 です。

カラム名	データ型	説明
created	UNSIGNED BIGINT	サブスクリプションが作成されたログオフセット。
log_sent	UNSIGNED BIGINT	情報をどこまでアップロードしたかを示すログの進行状況。更新されるこのカラムのエントリに対するアップロード確認の受信は必要ありません。
generation_number	INTEGER	ファイルベースのダウンロードで、このサブスクリプションに対して最後に受信した世代番号。デフォルトは0。
extended_state	VARCHAR(1024)	内部でのみ使用。
script_version	CHAR(128)	CREATE SYNCHRONIZATION SUBSCRIPTION 文、ALTER SYNCHRONIZATION SUBSCRIPTION 文、および START SYNCHRONIZATION SCHEMA CHANGE 文によって使用されるスクリプトバージョンを示します。
subscription_name	CHAR(128)	サブスクリプションの名前。
server_protocol	UNSIGNED BIGINT	内部でのみ使用。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (sync_id)
```

```
FOREIGN KEY (publication_id) REFERENCES SYS.ISYSPUBLICATION  
(publication_id)
```

```
UNIQUE INDEX (publication_id, site_name)
```

```
UNIQUE INDEX (subscription_name)
```

SYSSYNC2 統合ビュー

SYSSYNC2 ビューから、機密情報を公開することなく、SYSSYNC システムビューで見つかったデータ (同期に関する情報) にパブリックアクセスできます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

SELECT ANY TABLE システム権限のないユーザには、server_connect と option のカラムの値は非表示になります。

```
ALTER VIEW "SYS"."SYSSYNC2"  
as select ISYSSYNC.sync_id,
```



```

ISYSSYNC.type,
ISYSSYNC.publication_id,
ISYSSYNC.progress,
ISYSSYNC.site_name,
SYS.HIDE_FROM_NON_DBA(ISYSSYNC."option") as "option",
SYS.HIDE_FROM_NON_DBA(ISYSSYNC.server_connect) as
server_connect,
ISYSSYNC.server_conn_type,
ISYSSYNC.last_download_time,
ISYSSYNC.last_upload_time,
ISYSSYNC.created,
ISYSSYNC.log_sent,
ISYSSYNC.generation_number,
ISYSSYNC.extended_state,
ISYSSYNC.script_version,
ISYSSYNC.subscription_name
from SYS.ISYSSYNC

```

SYSSYNCPUBLICATIONDEFAULTS 統合ビュー

SYSSYNCPUBLICATIONDEFAULTS は、同期に関連するパブリケーションに対応したデフォルトの同期設定のビューです。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

SELECT ANY TABLE システム権限のないユーザには、server_connect と option のカラムの値は非表示になります。

```

ALTER VIEW "SYS"."SYSSYNCPUBLICATIONDEFAULTS"
as select s.sync_id,
p.publication_name,
SYS.HIDE_FROM_NON_DBA(s."option") as "option",
SYS.HIDE_FROM_NON_DBA(s.server_connect) as server_connect,
s.server_conn_type
from SYS.ISYSSYNC as s join SYS.ISYSPUBLICATION as p
on(p.publication_id = s.publication_id) where
s.site_name is null

```

SYSSYNCS 統合ビュー

SYSSYNCS ビューには、同期に関する情報が入っています。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

SELECT ANY TABLE システム権限のないユーザには、server_connect と option のカラムの値は非表示になります。

```

ALTER VIEW "SYS"."SYSSYNCS"
as select p.publication_name,s.progress,s.site_name,
SYS.HIDE_FROM_NON_DBA(s."option") as "option",
SYS.HIDE_FROM_NON_DBA(s.server_connect) as server_connect,

```

```
s.server_conn_type,s.last_download_time,
s.last_upload_time,s.created,s.log_sent,s.generation_number,
s.extended_state
from SYS.ISYSSYNC as s
left outer join SYS.ISYSPUBLICATION as p
on p.publication_id = s.publication_id
```

SYSSYNCSCRIPT システムビュー

SYSSYNCSCRIPT システムビューの各ローは、スクリプト化されたアップロードに関するストアプロシージャを識別します。このビューは、SYSSYNCSCRIPTS ビューとほとんど同じですが、このビューの値は未加工形式である点が異なります。

このビューの基本となるシステムテーブルは ISYSSYNCSCRIPT です。

カラム名	データ型	説明
pub_object_id	UNSIGNED BIGINT	スクリプトが所属するパブリケーションのオブジェクト ID。
table_object_id	UNSIGNED BIGINT	スクリプトの適用対象となるテーブルのオブジェクト ID。
type	UNSIGNED INT	アップロードプロシージャのタイプ。
proc_object_id	UNSIGNED BIGINT	パブリケーションに使用するストアプロシージャのオブジェクト ID。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (pub_object_id, table_object_id, type)
FOREIGN KEY (pub_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
FOREIGN KEY (table_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
FOREIGN KEY (proc_object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

SYSSYNCSCRIPTS 統合ビュー

SYSSYNCSCRIPTS ビューの各ローは、スクリプト化されたアップロードに関するストアプロシージャを識別します。このビューは、SYSSYNCSCRIPT システムビューとほとんど同じですが、データが未加工形式ではなく、ユーザが読みやすい形式で表示される点が異なります。

```
ALTER VIEW "SYS"."SYSSYNCSCRIPTS"
as select p.publication_name,
t.table_name,
case s.type
when 0 then 'upload insert'
when 1 then 'upload delete'
when 2 then 'upload update'
else 'unknown'
```

```

end as type,
c.proc_name
from SYS.ISYSSYNCSRIPT as s
  join SYS.ISYSPUBLICATION as p on p.object_id = s.pub_object_id
  join SYS.ISYSTAB as t on t.object_id = s.table_object_id
  join SYS.ISYSPROCEDURE as c on c.object_id = s.proc_object_id

```

SYSSYNCSUBSCRIPTIONS 統合ビュー

SYSSYNCSUBSCRIPTIONS ビューには、同期のサブスクリプションと関連付けられた同期設定が含まれます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

SELECT ANY TABLE システム権限のないユーザには、server_connect と option のカラムの値は非表示になります。

```

ALTER VIEW "SYS"."SYSSYNCSUBSCRIPTIONS"
as select s.sync_id,
  p.publication_name,
  s.progress,
  s.site_name,
  SYS.HIDE_FROM_NON_DBA(s."option") as "option",
  SYS.HIDE_FROM_NON_DBA(s.server_connect) as server_connect,
  s.server_conn_type,
  s.last_download_time,
  s.last_upload_time,
  s.created,
  s.log_sent,
  s.generation_number,
  s.extended_state
from SYS.ISYSSYNC as s join SYS.ISYSPUBLICATION as p
on(p.publication_id = s.publication_id)
where s.publication_id is not null and
  s.site_name is not null and exists
(select 1 from SYS.SYSSYNCSUSERS as u
  where s.site_name = u.site_name)

```

SYSSYNCSUSERS 統合ビュー

同期ユーザに関連する同期設定のビューです。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

SELECT ANY TABLE システム権限のないユーザには、server_connect と option のカラムの値は非表示になります。

```

ALTER VIEW "SYS"."SYSSYNCSUSERS"
as select ISYSSYNC.sync_id,
  ISYSSYNC.site_name,
  SYS.HIDE_FROM_NON_DBA(ISYSSYNC."option") as "option",

```

```
SYS.HIDE_FROM_NON_DBA (ISYSSYNC.server_connect) as
server_connect,
ISYSSYNC.server_conn_type
from SYS.ISYSSYNC where
ISYSSYNC.publication_id is null
```

SYSTAB システムビュー

SYSTAB システムビューの各ローは、データベース内のテーブル 1 つを示します。ビューの追加情報が SYSVIEW システムビューにあります。このビューの基本となるシステムテーブルは ISYSTAB です。

カラム名	データ型	説明
table_id	UNSIGNED INT	各テーブルにはユニークな番号(テーブル番号)が割り当てられます。
dbspace_id	SMALLINT	テーブルを含む DB 領域を示す値。
count	UNSIGNED BIGINT	テーブルまたはマテリアライズドビュー内のロー数。この値は、各チェックポイントが正常処理されているときに更新されます。この数を使用してデータベースアクセスが最適化されます。非マテリアライズドビューの場合、またはリモートテーブルの場合、カウントは常に 0 です。
creator	UNSIGNED INT	このユーザ番号はテーブルまたはビューの所有者を示します。
table_page_count	INTEGER	基本となるテーブルで使用されるメインページの総数。
ext_page_count	INTEGER	このテーブルで使用される拡張ページの総数。
commit_action	INTEGER	グローバルテンポラリテーブルの場合、0 は、テーブルの作成時に ON COMMIT PRESERVE ROWS 句が指定されたことを示します。1 は、テーブルの作成時に ON COMMIT DELETE ROWS 句が指定されたことを示します(テンポラリテーブルのデフォルト動作)。3 は、テーブルの作成時に NOT TRANSACTIONAL 句が指定されたことを示します。非テンポラリテーブルの場合、commit_action は常に 0 です。
share_type	INTEGER	グローバルテンポラリテーブルの場合、4 は、テーブルの作成時に SHARE BY ALL 句が指定されたことを示します。5 は、テーブルの作成時に SHARE BY ALL 句が指定されなかったことを示します。非テンポラリテーブルの場合、share_type は常に 5 です。これは、非テンポラリテーブルの作成時に SHARE BY ALL 句は指定できないためです。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	テーブルのオブジェクト ID。
last_modified_at	TIMESTAMP	テーブルのデータが最後に修正されたローカル日時。このカラムは、チェックポイント時にものみ更新されません。
table_name	CHAR(128)	テーブルまたはビューの名前。1人の作成者が、同じ名前のテーブルまたはビューを2つ以上持つことはできません。
table_type	TINYINT	テーブルまたはビューのタイプ。次のような値があります。 <ul style="list-style-type: none"> • 1 - ベーステーブル • 2 - マテリアライズドビュー • 3 - グローバルテンポラリテーブル • 4 - ローカルテンポラリテーブル • 5 - テキストインデックスベーステーブル • 6 - テキストインデックスグローバルテンポラリテーブル • 21 - ビュー
replicate	CHAR(1)	この値は内部でのみ使用されます。
server_type	TINYINT	基本となるテーブルのデータの場所。次のような値があります。 <ul style="list-style-type: none"> • 1 - ローカルサーバ • 3 - リモートサーバ
tab_page_list	LONG VARBIT	内部でのみ使用。テーブルの情報を含むページセット (ビットマップで表示されます)。
ext_page_list	LONG VARBIT	内部でのみ使用。テーブルに関するローの拡張と大規模なオブジェクト (LOB) ページを含むページセット (ビットマップで表示されます)。
pct_free	UNSIGNED INT	指定されている場合、テーブルの PCT_FREE の指定、それ以外の場合は NULL。
clustered_index_id	UNSIGNED INT	テーブルのクラスタ化されたインデックスの ID。クラスタ化されたインデックスがない場合、このフィールドは NULL です。
encrypted	CHAR(1)	テーブルまたはマテリアライズドビューが暗号化されるかどうか。

カラム名	データ型	説明
last_modified_tsn	UNSIGNED BIGINT	テーブルを修正したトランザクションに割り当てられたシーケンス番号。このカラムは、チェックポイント時にのみ更新されます。
current_schema	UNSIGNED INT	テーブルの現在のスキーマバージョン。
file_id	SMALLINT	廃止予定。このカラムは SYSVIEW に存在しますが、基本となるシステムテーブル ISYSTAB には存在しません。このカラムの内容は dbspace_id と同じであり、互換性のために提供されています。今後は dbspace_id を使用してください。
table_type_str	CHAR(13)	table_type の読み取り可能な値。次のような値があります。 <ul style="list-style-type: none"> • 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 • VIEW – ビュー
last_modified_at_utc	TIMESTAMP WITH TIME ZONE	テーブルのデータが最後に修正された UTC 日時。このカラムは、チェックポイント時にのみ更新されます。

基本となるシステムテーブルに関する制約

```
FOREIGN KEY (dbspace_id) REFERENCES SYS.ISYSDBSPACE (dbspace_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
PRIMARY KEY (table_id)
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (table_name, creator)
```

SYSTABAUTH 統合ビュー

SYSTABAUTH ビューには、SYSTABLEPERM システムビューの情報が表示されますが、より読みやすい形式です。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSTABAUTH" ( grantor,
    grantee, screator, stname, tcreator, ttname,
    selectauth, insertauth, deleteauth,
    updateauth, updatecols, alterauth, referenceauth,
```

```

loadauth,truncateauth )
as select u1.user_name,u2.user_name,u3.user_name,tab1.table_name,
u4.user_name,tab2.table_name,tp.selectauth,tp.insertauth,
tp.deleteauth,tp.updateauth,tp.updatecols,tp.alterauth,
tp.referenceauth,tp.loadauth,tp.truncateauth
from SYS.ISYSTABLEPERM as tp
join SYS.ISYSUSER as u1 on u1.user_id = tp.grantor
join SYS.ISYSUSER as u2 on u2.user_id = tp.grantee
join SYS.ISYSTAB as tab1 on tab1.table_id = tp.stable_id
join SYS.ISYSUSER as u3 on u3.user_id = tab1.creator
join SYS.ISYSTAB as tab2 on tab2.table_id = tp.stable_id
join SYS.ISYSUSER as u4 on u4.user_id = tab2.creator

```

SYSTABCOL システムビュー

SYSTABCOL システムビューには、データベースの各テーブルとビューの各カラムのローが含まれます。このビューの基本となるシステムテーブルは ISYSTABCOL です。

カラム名	データ型	説明
table_id	UNSIGNED INT	カラムが属するテーブルまたはビューのオブジェクト ID。
column_id	UNSIGNED INT	カラムの ID。各テーブルについて、カラムの番号は 1 から開始されます。 column_id 値によって、SELECT * が使用された場合の結果セットのカラムの順序が決定されます。また、INSERT 文にカラム名のリストが指定されない場合も、この値によってカラム順が決定されます。
domain_id	SMALLINT	カラムのデータ型を、SYSDOMAIN システムビューにリストされているデータ型番号で示します。
nulls	CHAR(1)	NULL 値をカラムに許可するかどうかを指定します。
width	BIGINT	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をバイトで示します。
scale	SMALLINT	NUMERIC または DECIMAL データ型のカラムについて、小数点以下の桁数。文字列のカラムの場合、1 の値は文字長のセマンティックを指定します。0 は、バイト長のセマンティックを指定します。
object_id	UNSIGNED BIGINT	テーブルカラムのオブジェクト ID。

カラム名	データ型	説明
max_identity	BIGINT	AUTOINCREMENT、IDENTITY、または GLOBAL AUTOINCREMENT カラムの場合、カラムの最大値。
column_name	CHAR(128)	カラム名。
"default"	LONG VARCHAR	カラムのデフォルト値。この値を指定した場合、INSERT 文が値を指定しないときのみ使われます。
user_type	SMALLINT	ユーザ定義のデータ型を使用してカラムが定義される場合、データ型。
column_type	CHAR(1)	カラムのタイプ (C=計算済みカラム、R=他のカラム)。
compressed	TINYINT	このカラムを圧縮形式で格納するかどうか。
collect_stats	TINYINT	システムが自動的にカラムの統計情報を収集および更新するかどうか。
inline_max	SMALLINT	ローに格納する BLOB の最大バイト数。NULL 値は、デフォルトが適用されたこと、またはカラムは文字型またはバイナリ型ではないことを示します。NULL ではない inline_max 値は、CREATE TABLE 文または ALTER TABLE 文を使用してカラムに指定した INLINE 値に対応します。
inline_long	SMALLINT	BLOB サイズが inline_max 値を超えた場合、ローに格納されている BLOB のバイトを複製した数。NULL 値は、デフォルトが適用されたこと、またはカラムは文字型またはバイナリ型ではないことを示します。NULL ではない inline_long 値は、CREATE TABLE 文または ALTER TABLE 文を使用してカラムに指定した PREFIX 値に対応します。
lob_index	TINYINT	内部的なしきい値サイズ(約8データベースページ)を超過するカラムの BLOB 値に関するインデックスを構築するかどうか。NULL 値は、デフォルトを適用するか、カラムが BLOB タイプであることを示します。1 の値は、インデックスを構築することを示します。0 の値は、インデックスを構築しないことを示します。NULL ではない lob_index 値は、CREATE TABLE 文または ALTER TABLE 文を使用してカラムに指定した INDEX または NO INDEX 値に対応します。
base_type_str	VARCHAR(32,767)	カラムの物理的な型を表す注釈付きの型文字列。

カラム名	データ型	説明
nonmaterialized_value	LONG BINARY	内部使用のみ。
start_schema	UNSIGNED INT	このカラムが存在するテーブルスキーマの最初のバージョン。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (table_id, column_id)
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

```
FOREIGN KEY (domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (user_type) REFERENCES SYS.ISYSUSERTYPE (type_id)
```

SYSTABLE 互換ビュー (旧式)

SYSTABLE ビューは、SYSTABLE システムテーブルを提供していた古いバージョンのソフトウェアとの互換性を保つために用意されています。ただし、以前の SYSTABLE テーブルは、SYSTAB システムビューに対応する ISYSTAB システムテーブルで置換されたため、SYSTAB の使用をおすすめします。

SYSTABLE ビューの各ローは、データベース内のテーブル 1 つを示します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSTABLE"  
as select b.table_id,  
       b.file_id,  
       b.count,  
       0 as first_page,  
       b.commit_action as last_page,  
       COALESCE(ph.root,0) as primary_root,  
       b.creator,  
       0 as first_ext_page,  
       0 as last_ext_page,  
       b.table_page_count,  
       b.ext_page_count,  
       b.object_id,  
       b.table_name,  
       b.table_type_str as table_type,  
       v.view_def,  
       r.remarks,  
       b.replicate,  
       p.existing_obj,  
       p.remote_location,
```

```

'T' as remote_objtype,
p.srvid,
case b.server_type
when 1 then 'SA'
when 2 then 'IQ'
when 3 then 'OMNI'
else 'INVALID'
end as server_type,
10 as primary_hash_limit,
0 as page_map_start,
s.source,
b."encrypted"
from SYS.SYSTAB as b
left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
left outer join SYS.ISYSSOURCE as s on(b.object_id =
s.object_id)
left outer join SYS.ISYSVIEW as v on(b.object_id =
v.view_object_id)
left outer join SYS.ISYSPROXYTAB as p on(b.object_id =
p.table_object_id)
left outer join(SYS.ISYSIDX as i left outer join SYS.ISYSPHYSIDX
as ph on(i.table_id = ph.table_id
and i.phys_index_id = ph.phys_index_id)) on(b.table_id =
i.table_id and i.index_category = 1
and i.index_id = 0)

```

SYSTABLEPERM システムビュー

GRANT 文によってテーブルやビューに対して付与された権限は、SYSTABLEPERM システムビューに格納されます。このビューの各ローは1つのテーブル、権限を与えるユーザ ID (grantor)、そして権限を与えられるユーザ ID (grantee) に対応します。このビューの基本となるシステムテーブルは ISYSTABLEPERM です。

カラム名	データ型	説明
stable_id	UNSIGNED INT	権限が適用されるテーブルまたはビューのテーブル番号。
grantee	UNSIGNED INT	権限を受けるユーザ ID のユーザ番号。
grantor	UNSIGNED INT	権限を与えるユーザ ID のユーザ番号。
selectauth	CHAR(1)	SELECT 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
insertauth	CHAR(1)	INSERT 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。

カラム名	データ型	説明
deleteauth	CHAR(1)	DELETE 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
updateauth	CHAR(1)	UPDATE 権限が、テーブル内のすべてのカラムに付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
updatecols	CHAR(1)	UPDATE 権限が、基本となるテーブル内の一部のカラムだけに付与されたかどうかを示します。updatecols が "Y" であれば、カラムに UPDATE 権限を与える 1 つまたは複数のローが、SYSCOLPERM システムビューにあります。
alterauth	CHAR(1)	ALTER 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
referenceauth	CHAR(1)	REFERENCE 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
loadauth	CHAR(1)	LOAD 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
truncateauth	CHAR(1)	TRUNCATE 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
loadauth	CHAR(1)	LOAD 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。
truncateauth	CHAR(1)	TRUNCATE 権限が付与されたかどうかを示します。可能な値は Y、N、または G です。これらの値の詳細については、以下の備考部分を参照してください。

備考

与えられる権限には型がいくつかあります。それぞれの権限は、次の3つのいずれかの値を持ちます。

- **N** – No。grantee はこの権限を grantor から付与されていません。
- **Y** – Yes。grantee はこの権限を grantor から付与されています。

- **G** – grantee はこの権限を受けています。また、grantee は同じ権限を他のユーザーに付与できます。

注意： grantee は同じテーブルに関する権限を、他の grantor から受けることがあります。その場合、この情報は、SYSTABLEPERM システムビューの異なるローで見つかります。

基本となるシステムテーブルに関する制約

PRIMARY KEY (stable_id, grantee, grantor)

FOREIGN KEY (stable_id) REFERENCES SYS.ISYSTAB (table_id)

FOREIGN KEY (grantor) REFERENCES SYS.ISYSUSER (user_id)

FOREIGN KEY (grantee) REFERENCES SYS.ISYSUSER (user_id)

SYSTEXTCONFIG システムビュー

SYSTEXTCONFIG システムビューの各ローは、全文検索機能で使用するテキスト設定オブジェクト 1 つを示します。このビューの基本となるシステムテーブルは ISYSTEXTCONFIG です。

カラム名	データ型	説明
object_id	UNSIGNED BIGINT	テキスト設定オブジェクトのオブジェクト ID。
creator	UNSIGNED INT	テキスト設定オブジェクトの作成者。
term_breaker	TINYINT	文字列を単語に分割するために使用されるアルゴリズム。値は GENERIC の場合は 0、NGRAM の場合は 1 です。GENERIC の場合、英数字以外の文字で区切られた 1 つまたは複数の英数字の文字列は、単語として扱われます。NGRAM は近似一致または単語の区切りにホワイトスペースを使用しないドキュメントに使用します。
stemmer	TINYINT	内部でのみ使用。
min_term_length	TINYINT	1 つの単語に許容される最小文字数。min_term_length より短い単語は無視されます。 MINIMUM TERM LENGTH 設定は、GENERIC 単語区切りの場合のみ意味を持ちます。NGRAM テキストインデックスの場合、この設定は無視されます。

カラム名	データ型	説明
max_term_length	TINYINT	GENERIC テキストインデックスの場合は、1つの単語に許容される最大長(文字数)。max_term_lengthより長い単語は無視されます。 NGRAM テキストインデックスの場合は、単語が分解される N-gram の長さ。
collation	CHAR(128)	内部でのみ使用。
text_config_name	CHAR(128)	テキスト設定オブジェクトの名前。
prefilter	LONG VARCHAR	外部事前フィルタライブラリの関数名およびライブラリ名です。
postfilter	LONG VARCHAR	内部でのみ使用。
char_stoplist	LONG VARCHAR	CHAR カラムでの全文検索実行時に無視する単語。これらの単語は、テキストインデックスからも省略されます。このカラムは、テキスト設定オブジェクトが default_char から作成されるときに使用されません。
nchar_stoplist	LONG NVARCHAR	NCHAR カラムでの全文検索実行時に無視する単語。これらの単語は、テキストインデックスからも省略されます。このカラムは、テキスト設定オブジェクトが default_nchar から作成されるときに使用されません。
external_term_breaker	LONG VARCHAR	外部単語区切りライブラリの関数名およびライブラリ名です。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (object_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE INDEX (creator, text_config_name)
```

SYSTEXTIDX システムビュー

SYSTEXTIDX システムビューの各ローは、1つのテキストインデックスを示します。このビューの基本となるシステムテーブルは ISYSTEXTIDX です。

カラム名	データ型	説明
index_id	UNSIGNED BIGINT	SYSIDX のテキストインデックスのオブジェクト ID。
sequence	UNSIGNED INT	内部でのみ使用。
status	UNSIGNED INT	内部でのみ使用。
text_config	UNSIGNED BIGINT	SYSTEXTCONFIG のテキスト設定オブジェクトのオブジェクト ID。
next_handle	UNSIGNED INT	内部でのみ使用。
last_handle	UNSIGNED INT	内部でのみ使用。
deleted_length	UNSIGNED BIGINT	テキストインデックスの削除されたインデックス付きの値の合計サイズ。
pending_length	UNSIGNED BIGINT	次のリフレッシュ時にテキストインデックスに追加されるインデックス付きの値の合計サイズ。
refresh_type	TINYINT	再表示タイプ。次のいずれかになります。 <ul style="list-style-type: none"> • 1 – MANUAL • 2 – AUTO • 3 – IMMEDIATE
refresh_interval	UNSIGNED INT	自動リフレッシュの間隔 (分単位)。
last_refresh	TIMESTAMP	最後のリフレッシュのローカル時刻。
last_refresh_utc	TIMESTAMP WITH TIME ZONE	最後のリフレッシュの UTC 時刻。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (index_id, sequence)
```

```
FOREIGN KEY (index_id) REFERENCES SYS.ISYSOBJECT (object_id)
```

```
FOREIGN KEY (text_config) REFERENCES SYS.ISYSTEXTCONFIG (object_id)
```

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である生成されたテーブルを示します。このビューの基本となるシステムテーブルは ISYSTEXTIDXTAB です。

カラム名	データ型	説明
index_id	UNSIGNED BIGINT	内部でのみ使用。
sequence	UNSIGNED INT	内部でのみ使用。
table_type	UNSIGNED INT	内部でのみ使用。
table_id	UNSIGNED INT	内部でのみ使用。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (index_id, sequence, table_type)
```

```
FOREIGN KEY (index_id, sequence) REFERENCES SYS.ISYSTEXTIDX  
(index_id, sequence)
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

SYSTRIGGER システムビュー

SYSTRIGGER システムビューの各ローは、データベース内のトリガ 1 つを示します。このテーブルには、参照トリガアクションを持つ外部キー定義に、自動的に作成されるトリガも含まれます (たとえば、ON DELETE CASCADE)。このビューの基本となるシステムテーブルは ISYSTRIGGER です。

カラム名	データ型	説明
trigger_id	UNSIGNED INT	SYSTRIGGER ビュー内のトリガのユニークな番号です。
table_id	UNSIGNED INT	トリガが所属するテーブルのテーブル ID。
object_id	UNSIGNED BIGINT	データベース内のトリガのオブジェクト ID です。

カラム名	データ型	説明
event	CHAR(1)	トリガが起動されるきっかけとなる操作。 <ul style="list-style-type: none"> • A – INSERT、DELETE • B – INSERT、UPDATE • C – UPDATE COLUMNS • D – DELETE • E – DELETE、UPDATE • I – INSERT • M – INSERT、DELETE、UPDATE • U – UPDATE
trigger_time	CHAR(1)	イベントに関連するトリガが起動されるタイミング。 <ul style="list-style-type: none"> • A – AFTER (ローレベルのトリガ) • B – BEFORE (ローレベルのトリガ) • I – INSTEAD OF (ローレベルのトリガ) • K – INSTEAD OF (文レベルのトリガ) • R – RESOLVE • S – AFTER (文レベルのトリガ)
trigger_order	SMALLINT	同じタイプ (insert、update、または delete) の複数のトリガが、同じ時刻 (BEFORE または AFTER トリガのみが該当) に起動するように設定されている場合に、トリガが起動される順序。
foreign_table_id	UNSIGNED INT	参照トリガアクション (たとえば ON DELETE CASCADE) を持つ外部キー定義のあるテーブルの ID。foreign_table_id 値は、ISYSIDX.table_id の値を反映します。
foreign_key_id	UNSIGNED INT	foreign_table_id によって参照されるテーブルの外部キーの ID。foreign_key_id 値は、ISYSIDX.index_id の値を反映します。
referential_action	CHAR(1)	外部キーによって定義される動作。この 1 文字の値は、外部キーを作成したときに指定した動作に対応します。 <ul style="list-style-type: none"> • C – CASCADE • D – SET DEFAULT • N – SET NULL • R – RESTRICT
trigger_name	CHAR(128)	トリガ名。1 つのテーブルには、同じ名前のトリガは複数存在できません。

カラム名	データ型	説明
trigger_defn	LONG VARCHAR	トリガを作成するのに使ったコマンド。
remarks	LONG VARCHAR	トリガに関する注記。ISYSREMARK システムテーブル内に格納されている値。
source	LONG VARCHAR	トリガの SQL ソース。この値は、ISYSSOURCE システムテーブル内に格納されています。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (trigger_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
FOREIGN KEY (table_id) REFERENCES SYS.ISYSTAB (table_id)
```

```
FOREIGN KEY fkey_index (foreign_table_id, foreign_key_id) REFERENCES  
SYS.ISYSIDX (table_id, index_id)
```

```
UNIQUE INDEX (table_id, event, trigger_time, trigger_order)
```

```
UNIQUE INDEX (trigger_name, table_id)
```

```
UNIQUE INDEX (table_id, foreign_table_id, foreign_key_id, event)
```

SYSTRIGGERS 統合ビュー

SYSTRIGGERS ビューの各ローは、データベース内のトリガ 1 つを示します。このテーブルには、参照トリガアクションを持つ外部キー定義に、自動的に作成されるトリガも含まれます (たとえば、ON DELETE CASCADE)。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSTRIGGERS"( owner,  
    trigname,tname,event,trigtime,trigdefn )  
as select u.user_name,trig.trigger_name,tab.table_name,  
    if trig.event = 'I' then 'INSERT'  
    else if trig.event = 'U' then 'UPDATE'  
    else if trig.event = 'C' then 'UPDATE'  
    else if trig.event = 'D' then 'DELETE'  
    else if trig.event = 'A' then 'INSERT,DELETE'  
    else if trig.event = 'B' then 'INSERT,UPDATE'  
    else if trig.event = 'E' then 'DELETE,UPDATE'  
    else 'INSERT,DELETE,UPDATE'  
    endif  
    endif  
    endif  
    endif  
    endif
```

```

        endif
    endif,if trig.trigger_time = 'B' or trig.trigger_time = 'P' then
'BEFORE'
    else if trig.trigger_time = 'A' or trig.trigger_time = 'S' then
'AFTER'
        else if trig.trigger_time = 'R' then 'RESOLVE'
            else 'INSTEAD OF'
        endif
    endif
endif, trig.trigger_defn
from SYS.ISYSTRIGGER as trig
join SYS.ISYSTAB as tab on (tab.table_id = trig.table_id)
join SYS.ISYSUSER as u on u.user_id = tab.creator where
trig.foreign_table_id is null
    
```

SYSTYPEMAP システムビュー

SYSTYPEMAP システムビューには、SYSSQLSERVERTYPE システムビューの互換性マッピング値があります。このビューの基本となるシステムテーブルは ISYSTYPEMAP です。

カラム名	データ型	説明
ss_user_type	SMALLINT	Adaptive Server Enterprise ユーザ型を格納します。
sa_domain_id	SMALLINT	対応する SAP Sybase IQ domain_id を格納します。
sa_user_type	SMALLINT	対応する SAP Sybase IQ ユーザ型を格納します。
nullable	CHAR(1)	型が NULL 値を許容するかどうか。

基本となるシステムテーブルに関する制約

```
FOREIGN KEY (sa_domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)
```

SYSTYPES ASE 互換ビュー

systypes には、システムが提供するデータ型およびユーザ定義データ型ごとに1つのローが含まれています。ドメイン(ルールで定義される)とデフォルトが存在する場合は、それらも含まれています。

このビューは、DBO が所有します。システムが提供するデータ型を示すローは、変更できません。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYSCOMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)

- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

SYSUSER システムビュー

SYSUSER システムビューの各ローは、システム内のユーザを示します。スタンドアロンロールも同様にこのビューに格納されますが、これらのローにとっては user_id、object_id、user_name、user_type カラムのみ意味があります。このビューの基本となるシステムテーブルは ISYSUSER です。

カラム名	データ型	説明
user_id	UNSIGNED INT	ログインポリシーが割り当てられたユーザのユニークな識別子。
object_id	UNSIGNED BIGINT	データベースのユーザのユニークな識別子。
user_name	CHAR(128)	ユーザのログイン名。
password	BINARY(128)	ユーザのパスワード。セキュリティ上の理由から、このカラムのデータは SELECT ANY TABLE システム権限を持つユーザのみに表示されます。
login_policy_id	UNSIGNED BIGINT	ログインポリシーのユニークな識別子。
expired_password_on_login	TINYINT	次のログイン時にユーザのパスワードの有効期限が切れるかどうかを示す値。
password_creation_time	TIMESTAMP	ユーザのパスワードが作成されたローカル時刻。
failed_login_attempts	UNSIGNED INT	アカウントがロックされるまでにユーザがログインを失敗できる回数。
last_login_time	TIMESTAMP	ユーザが最後にログインした現地時刻。

カラム名	データ型	説明
user_type	TINYINT	ユーザが通常のユーザなのか、ロールなのか、ロールとして拡張されたユーザなのかを示す値。ユーザ、ロール、または拡張ロールが変更可能(可変)なのか、または削除可能なのかを示します。考えられる値は、次のとおりです。 <ul style="list-style-type: none"> • 1 - 不変のシステムロール。 • 5 - 可変のシステムロール。 • 9 - 不変および削除可能なシステムロール。 • 12 - 可変および削除可能なユーザ。 • 13 - 可変および削除可能なロール。 • 14 - 可変および削除可能な、ロールとして拡張されたユーザ。
user_dn	CHAR (1024)	ドメイン内および複数のドメインでユニークな、ユーザの LDAP 識別名 (DN) の ID。DN は LDAP サーバでの認証に使用されます。
user_dn_cached_at	TIMESTAMP	user_dn カラムが最後にキャッシュされた時刻。この値は古い DN をパージするかどうか決めるために使用されます。データベースサーバのローカルのタイムゾーンに関係なく、この値は協定世界時 (UTC: Coordinated Universal Time) で格納されます。
password_creation_time_utc	TIMESTAMP WITH TIME ZONE	ユーザのパスワードが作成された UTC 時刻。
last_login_time_utc	TIMESTAMP WITH TIME ZONE	ユーザが最後にログインした UTC 時刻。
dual_password	BINARY(128)	ユーザの二重パスワードの 1 番目か 2 番目、またはその両方の部分。セキュリティ上の理由から、このカラムのデータは SELECT ANY TABLE システム権限を持つユーザのみに表示されます。
lock_time	TIMESTAMP	ログイン試行の失敗のためユーザがロックされたタイムスタンプ。

基本となるシステムテーブルに関する制約

PRIMARY KEY (user_id)

FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH UNIQUE FULL

```
FOREIGN KEY (login_policy_id) REFERENCES SYS.ISYSLOGINPOLICY
(login_policy_id)
```

```
UNIQUE INDEX (user_name)
```

SYSUSERAUTH 互換ビュー (旧式)

SYSUSERAUTH ビューの各ローは、user_id を公開せずに、ユーザに関して記述します。各ユーザは、ユーザ名で識別されます。このビューはパスワードを表示するため、そのデータを表示するには SELECT ANY TABLE システム権限が必要です。

SYSUSERAUTH ビューは、古いバージョンのソフトウェアとの互換性を保つために用意されています。代わりに SYSROLEGRANTS 統合ビューを使用してください。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

注意： このビューのタイトルには auth (権限) という単語が含まれていますが、セキュリティモデルはロールと権限に基づきます。したがって、ビュー内のデータは、ビューの定義に記載されているテーブルとビューからのロール情報を使用してコンパイルされます。

```
ALTER VIEW "SYS"."SYSUSERAUTH" ( name,
    password, resourceauth, dbaauth, scheduleauth, user_group )
as select
SYSUSERPERM.user_name, SYSUSERPERM.password, SYSUSERPERM.resourceauth
, SYSUSERPERM.dbaauth, SYSUSERPERM.scheduleauth, SYSUSERPERM.user_group
P
from SYS.SYSUSERPERM
```

SYSUSERAUTHORITY 互換ビュー (旧式)

SYSUSERAUTHORITY ビューは、古いバージョンのソフトウェアとの互換性を保つために用意されています。代わりに SYSROLEGRANTS 統合ビューを使用してください。

SYSUSERAUTHORITY システムビューの各ローは、ユーザ ID に付与された権限を示します。

注意： このビューのタイトルには authority (権限) という単語が含まれていますが、セキュリティモデルはロールと権限に基づきます。したがって、ビュー内のデータは、ビューの定義に記載されているテーブルとビューからのロール情報を使用してコンパイルされます。

```
ALTER VIEW "SYS"."SYSUSERAUTHORITY" as
select ISYSROLEGRANT.grantee as user_id,
sp_auth_sys_role_info.auth
from SYS.ISYSROLEGRANT
```

```
natural join dbo.sp_auth_sys_role_info()
where ISYSROLEGRANT.grant_type <> (0x02|0x04) and
not ISYSROLEGRANT.grantee = any(select
sp_auth_sys_role_info.role_id from dbo.sp_auth_sys_role_info())
union
select ISYSUSER.user_id,
cast('GROU' as varchar(20)) as auth
from SYS.ISYSUSER
where ISYSUSER.user_name
in( 'SYS','PUBLIC','diagnostics','SYS_SPATIAL_ADMIN_ROLE','rs_systa
bgroup','SA_DEBUG','dbo' ) union
select ISYSUSER.user_id,
cast('GROU' as varchar(20)) as auth
from SYS.ISYSUSER
where ISYSUSER.user_type = (0x02|0x04|0x08) union
select cast(opt.setting as unsigned integer) as user_id,
cast('PUBLISH' as varchar(20)) as auth
from SYS.ISYSOPTION as opt
where opt."option" like '%db_publisher%' and opt.setting not like
'%-1%'
```

SYSUSERLIST 互換ビュー (旧式)

SYSUSERAUTH ビューは、古いバージョンのソフトウェアとの互換性を保つために用意されています。

SYSUSERLIST ビューの各ローは、user_id とパスワードを公開せずに、ユーザに関して記述します。各ユーザは、ユーザ名で識別されます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSUSERLIST" ( name,
resourceauth, dbaauth, scheduleauth, user_group )
as select
SYSUSERPERM.user_name, SYSUSERPERM.resourceauth, SYSUSERPERM.dbaauth,
SYSUSERPERM.scheduleauth, SYSUSERPERM.user_group
from SYS.SYSUSERPERM
```

SYSUSERMESSAGE システムビュー

SYSUSERMESSAGE システムビューの各ローには、エラーに対するユーザ定義メッセージが格納されます。このビューの基本となるシステムテーブルは ISYSUSERMESSAGE です。

注意： SYSUSERMESSAGES システムテーブルに含まれる前のカタログバージョン。このテーブル名は ISYSUSERMESSAGE ('S'なし)に変更され、このビューの基本となるテーブルになります。

カラム名	データ型	説明
error	INTEGER	エラー条件に対するユニークな識別番号。

カラム名	データ型	説明
uid	UNSIGNED INT	メッセージを定義するユーザ番号。
description	VARCHAR(255)	エラー条件に対応するメッセージ。
langid	SMALLINT	予約。

基本となるシステムテーブルに関する制約

```
FOREIGN KEY (uid) REFERENCES SYS.ISYSUSER (user_id)
```

```
UNIQUE CONSTRAINT (error, langid)
```

SYSUSEROPTIONS 統合ビュー

SYSUSEROPTIONS ビューには、各ユーザに有効なオプション設定が含まれます。ユーザにオプション設定がない場合、このビューには PUBLIC のオプション設定が表示されます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSUSEROPTIONS" ( user_name,
    "option", setting )
as select u.user_name,
    o."option",
    isnull((select s.setting
        from SYS.ISYSOPTION as s
        where s.user_id = u.user_id
        and s."option" = o."option"),
    o.setting)
from SYS.SYSOPTIONS as o,SYS.ISYSUSER as u
where o.user_name = 'PUBLIC'
```

SYSUSERPERM 互換ビュー (旧式)

SYSUSERPERM ビューの各ローは1つのユーザIDを表します。このビューのデータを表示するには、SELECT ANY TABLE システム権限が必要です。

前のバージョンで使用できる権限とパーミッションが表示されるだけなので、このビューは使用されなくなりました。SYSROLEGRANTS 統合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。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```
ALTER VIEW "SYS"."SYSUSERPERM"
as select b.user_id,
    b.object_id,
    b.user_name,
    b.password,
```

```

if AA.resourceauth is not null and AA.resourceauth > 0 then
  'Y' else 'N' endif as resourceauth,
if AA.dbauth is not null and AA.dbauth > 0 then
  'Y' else 'N' endif as dbauth,
  'N' as scheduleauth,
if exists(select * from SYS.ISYSOPTION as opt
  where opt."option" like '%db_publisher%' and opt.setting not
like '%-1'
  and b.user_id = cast(opt.setting as integer)) then
  'Y' else 'N' endif as publishauth,
if AA.remotedbauth is not null and AA.remotedbauth > 0 then
  'Y' else 'N' endif as remotedbauth,
if b.user_type = (0x02|0x04|0x08) or b.user_name
in( 'SYS','PUBLIC','diagnostics','SYS_SPATIAL_ADMIN_ROLE','rs_systa
bgroup','SA_DEBUG','dbo' ) then
  'Y' else 'N' endif as user_group,
  r.remarks
from SYS.ISYSUSER as b
  left outer join SYS.ISYSREMARK as r on(b.object_id =
r.object_id)
  left outer join(select sum(if sp_auth_sys_role_info.auth =
'RESOURCE' then 1 else 0 endif) as resourceauth,
  sum(if sp_auth_sys_role_info.auth = 'DBA' then 1 else 0 endif)
as dbauth,
  sum(if sp_auth_sys_role_info.auth = 'REMOTE DBA' then 1 else 0
endif) as remotedbauth,
  ISYSROLEGRANT.grantee
  from SYS.ISYSROLEGRANT natural join
dbo.sp_auth_sys_role_info()
  where ISYSROLEGRANT.grant_type <> (0x02|0x04)
  and sp_auth_sys_role_info.auth in( 'DBA','RESOURCE','REMOTE
DBA' )
  group by ISYSROLEGRANT.grantee) as AA
on(AA.grantee = b.user_id)

```

SYSUSERPERMS 互換ビュー (旧式)

前のバージョンで使用できる権限とパーミッションが表示されるだけなので、このビューは使用されなくなりました。SYSROLEGRANTS 統合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。

SYSUSERPERMS ビューの各ローは 1 つのユーザ ID を表します。ただし、パスワード情報は含まれません。すべてのユーザがこのビューを表示できます。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

```

ALTER VIEW "SYS"."SYSUSERPERMS"
  as select
SYSUSERPERM.user_id,SYSUSERPERM.user_name,SYSUSERPERM.resourceauth,
SYSUSERPERM.dbauth,

SYSUSERPERM.scheduleauth,SYSUSERPERM.user_group,SYSUSERPERM.publish

```



```
auth, SYSUSERPERM.remotedbaauth, SYSUSERPERM.remarks
from SYS.SYSUSERPERM
```

SYSUSERTYPE システムビュー

SYSUSERTYPE システムビューの各ローには、ユーザ定義のデータ型の説明が格納されます。このビューの基本となるシステムテーブルは ISYSUSERTYPE です。

カラム名	データ型	説明
type_id	SMALLINT	ユーザ定義データ型のユニークな識別番号。
creator	UNSIGNED INT	データ型の所有者のユーザ番号。
domain_id	SMALLINT	SYSDOMAIN システムビューにリストされたデータ型番号で示す、ユーザ定義データ型の元になるデータ型。
nulls	CHAR(1)	ユーザ定義データ型が NULL を許可するかどうか。可能な値は Y、N、または U です。値 U は、NULL 入力属性が指定されていないことを示します。
width	BIGINT	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記憶領域のサイズをバイト数で示します。
scale	SMALLINT	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 を示します。
type_name	CHAR(128)	データ型の名前。
"default"	LONG VARCHAR	データ型のデフォルト値。
"check"	LONG VARCHAR	データ型の CHECK 条件。
base_type_str	VARCHAR(32767)	ユーザタイプの物理的な型を表す注釈付きの型文字列。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (type_id)
```

```
FOREIGN KEY (creator) REFERENCES SYS.ISYSUSER (user_id)
```

```
FOREIGN KEY (domain_id) REFERENCES SYS.ISYSDOMAIN (domain_id)
```

```
UNIQUE CONSTRAINT (type_name)
```

SYSUSERS ASE 互換ビュー

sysusers には、データベースで許可されたユーザごとに 1 つのロー、およびグループまたは役割ごとに 1 つのローが含まれています。

このビューは、DBO が所有します。

参照：

- 各 Adaptive Server データベースに存在するテーブル (864 ページ)
- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYS COMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)

SYSVIEW システムビュー

SYSVIEW システムビューの各ローは、データベース内のビューを示します。

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。このビューの基本とな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。マテリアライズドビューは、IQ カタログストアの SQL Anywhere テーブルでのみサポートされません。

カラム名	データ型	説明
view_object_id	UNSIGNED BIGINT	ビューのオブジェクト ID。
view_def	LONG VARCHAR	ビューの定義 (クエリ仕様)。
mv_build_type	TINYINT	現在使われていない。
mv_refresh_type	TINYINT	ビューに定義された再表示タイプ。有効な値は IMMEDIATE (1) と MANUAL (2)。
mv_use_in_optimization	TINYINT	クエリの最適化時にマテリアライズドビューを使用できるかどうか (0 = 最適化で使用できない、1 = 最適化で使用できる)。
mv_last_refreshed_at	TIMESTAMP	マテリアライズドビューが最後にリフレッシュされたローカル日時を示す。

カラム名	データ型	説明
mv_known_stale_at	TIMESTAMP	マテリアライズドビューが古くなったローカル時刻。この値は、基本となるいずれかのベーステーブルの変更が検出された時刻に対応する。値0は、ビューが新しいか、古いけれどもデータベースサーバから古いというマークが付けられていないこと(古くなってからそのビューが使用されていない場合など)を示す。マテリアライズドビューのステータスを確認するには、sa_materialized_view_info システムプロシージャを使用する。
mv_last_refreshed_tsn	UNSIGNED BIGINT	マテリアライズドビューをリフレッシュしたトランザクションに割り当てられたシーケンス番号。
mv_last_refreshed_at_utc	TIMESTAMP WITH TIME ZONE	マテリアライズドビューが最後にリフレッシュされた UTC 日時を示す。
mv_known_stale_at_utc	TIMESTAMP WITH TIME ZONE	マテリアライズドビューが古くなった UTC 時刻。この値は、基本となるいずれかのベーステーブルの変更が検出された時刻に対応する。値0は、ビューが新しいか、古いけれどもデータベースサーバから古いというマークが付けられていないこと(古くなってからそのビューが使用されていない場合など)を示す。マテリアライズドビューのステータスを確認するには、sa_materialized_view_info システムプロシージャを使用する。このカラムには、mv_last_refreshed_at が0の場合は0が含まれ、mv_last_refreshed_at が NULL の場合は NULL が含まれる。

基本となるシステムテーブルに関する制約

PRIMARY KEY (view_object_id)

FOREIGN KEY (view_object_id) references SYS.ISYSOBJECT (object_id)
MATCH UNIQUE FULL

SYSVIEWS 統合ビュー

SYSVIEWS ビューの各ローには、ビュー定義など、ビューに関して記述します。

ビューを構成するテーブルとカラムは、以下の SQL 文で示されます。特定のテーブルまたはカラムの詳細については、以下のビュー定義にあるリンクを参照してください。

システムテーブルとシステムビュー

```
ALTER VIEW "SYS"."SYSVIEWS"( vcreator,
viewname,viewtext )
as select u.user_name,t.table_name,v.view_def
from SYS.ISYSTAB as t
join SYS.ISYSVIEW as v on(t.object_id = v.view_object_id)
join SYS.ISYSUSER as u on(u.user_id = t.creator)
```

SYSWEBSERVICE システムビュー

SYSWEBSERVICE システムビューの各ローには、Web サービスの説明が格納されます。このビューの基本となるシステムテーブルは ISYSWEBSERVICE です。

カラム名	データ型	説明
service_id	UNSIGNED INT	Web サービスをユニークに識別する番号。
object_id	UNSIGNED BIGINT	Web サービスの ID。
service_name	CHAR(128)	Web サービスに割り当てられた名前。
service_type	VARCHAR(40)	サービスのタイプには、RAW、HTTP、XML、SOAP、DISH などがあります。
auth_required	CHAR(1)	すべての要求に有効なユーザ名とパスワードが必要かどうか。
secure_required	CHAR(1)	HTTP などのセキュリティ保護されていない接続を受け入れるのか、または HTTPS などのセキュリティ保護された接続だけを受け入れるのか。
url_path	CHAR(1)	URL の解釈を制御します。
user_id	UNSIGNED INT	認証が有効の場合、サービス使用のパーミッションを持つユーザまたはユーザグループを識別します。認証が無効の場合、要求を処理するときに使用するアカウントを指定します。
parameter	LONG VARCHAR	DISH サービスにインクルードされる SOAP サービスを識別するプレフィクス。
statement	LONG VARCHAR	要求に応答して常に実行される SQL 文。NULL の場合、各要求に含まれる任意の文が代わりに実行されます。DISH 型のサービスでは無視されます。
remarks	LONG VARCHAR	Web サービスに関する注記。ISYSREMARK システムテーブル内に格納されている値。
enabled	CHAR(1)	Web サービスが、現在、有効または無効のいずれであるかを示します (CREATE SERVICE を参照してください)。

基本となるシステムテーブルに関する制約

```
PRIMARY KEY (service_id)
```

```
FOREIGN KEY (object_id) REFERENCES SYS.ISYSOBJECT (object_id) MATCH  
UNIQUE FULL
```

```
UNIQUE CONSTRAINT (service_name)
```

Transact-SQL 互換ビュー

Adaptive Server および SAP Sybase IQ では、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を用意しています。

Adaptive Server 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、サーバ内にあるすべ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 データベースとともに多数のデータベースが存在し、各データベースにはそれに関連するシステムテーブルがあります。

SAP Sybase IQ の場合、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としての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ながら、複数のデータベースを同時に実行できます。

Adaptive Server と SAP Sybase IQ のシステムカタログは異なります。Adaptive Server 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分散されています。一方、SAP 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ていません。

互換性のあるアプリケーションを作成できるように、SAP Sybase IQ には、特殊なユーザ dbo が所有する一連のビューがあります。これらは、Adaptive Server のシステムテーブルおよびビューに対応しています。構造上の違いのために、特定の Adaptive Server テーブルまたはビューの内容が SAP Sybase IQ のコンテキストで無意味になる場合、その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。

ここでは、Adaptive Server のシステムテーブルと、そのシステムテーブルの SAP 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。

参照：

- ASE T-SQL 互換ビュー (750 ページ)

各 Adaptive Server データベースに存在するテーブル

Adaptive Server のすべてのシステムテーブルが SAP Sybase IQ システムカタログに実装されているわけではありません。

表 127 : 各 Adaptive Server データベースに存在するテーブル

テーブル名	説明	データ?	SAP Sybase IQ によるサポート
sysalternates	データベースユーザにマップされたユーザごとのロー	いいえ	いいえ
syscolumns	テーブルまたはビューのカラムごと、およびプロシージャのパラメータごとに 1 ロー。SAP Sybase IQ では、クエリするときに所有者名 dbo (つまり、dbo.syscolumns) を使用する。	はい	はい
syscomments	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。	はい	はい
sysconstraints	テーブルまたはカラムに関連する参照制約および検査制約それぞれにつき 1 つのロー。	いいえ	いいえ
sysdepends	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。	いいえ	いいえ
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。SAP Sybase IQ では、クエリするときに所有者名 dbo (つまり、dbo.sysindexes) を使用する。	はい	はい
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、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。	はい	はい
sysiqindex	FP IQ 以外のインデックスごとに 1 ロー。	はい	はい
syskeys	ユーザが設定した (Adaptive Server 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。	いいえ	いいえ
syslogs	トランザクションログ。	いいえ	いいえ

テーブル名	説明	データ?	SAP Sybase IQ によるサ ポート
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。	はい
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。	いいえ	いいえ
sysprotects	ユーザのパーミッション情報。	いいえ	いいえ
sysreferences	テーブルまたはカラムに宣言された参照整合性制約につき 1 つのロー。	いいえ	いいえ
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。	いいえ	いいえ
syssegments	各セグメント (ディスクの一部分の集まりに名前を付けたもの) につき 1 つのロー。	いいえ	いいえ
systhresholds	データベースに定義されたスレッシュホールドにつき 1 つのロー。	いいえ	いいえ
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。	はい	はい
sysusers	データベースに許可されているユーザにつき 1 つのロー。	はい	はい

参照：

- SYSCOLUMNS ASE 互換ビュー (763 ページ)
- SYS COMMENTS ASE 互換ビュー (763 ページ)
- SYSINDEXES ASE 互換ビュー (782 ページ)
- SYSIQOBJECTS ASE 互換ビュー (794 ページ)
- SYSIQVINDEXT ASE 互換ビュー (798 ページ)
- SYSOBJECTS ASE 互換ビュー (806 ページ)
- SYSTYPES ASE 互換ビュー (852 ページ)
- SYSUSERS ASE 互換ビュー (859 ページ)

Adaptive Server の master データベースに存在するテーブル

Adaptive Server のすべての master データベーステーブルが SAP Sybase IQ システムカタログに実装されているわけではありません。

表 128 : ASE master データベーステーブル

テーブル名	説明	データ?	SAP Sybase 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。	いいえ	いいえ
sysconfigures	ユーザが設定できる設定パラメータごとに 1 ロー。	いいえ	いいえ
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。	いいえ	いいえ
sysdatabases	サーバ上のデータベースごとに 1 ロー。	いいえ	いいえ
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。	いいえ	いいえ
sysengines	現在オンラインのサーバごとに 1 ロー。	いいえ	いいえ
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。	いいえ	いいえ
syslocks	アクティブロックについての情報。	いいえ	いいえ
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。	いいえ	いいえ
syslogins	有効なユーザアカウントごとに 1 ロー。	はい	はい
sysmessages	システムエラーまたは警告ごとに 1 ロー。	いいえ	いいえ
sysprocesses	サーバプロセスについての情報。	いいえ	いいえ
sysremotelogins	リモートユーザごとに 1 ロー。	いいえ	いいえ
sysrvroles	サーバ全体に適用される役割ごとに 1 ロー。	いいえ	いいえ
syssservers	リモートサーバごとに 1 ロー。	いいえ	いいえ
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。	いいえ	いいえ

Adaptive Server の sybsecurity データベースに存在するテーブル

Adaptive Server のすべての sybsecurity データベーステーブルが SAP Sybase IQ システムカタログに実装されているわけではありません。

表 129 : ASE sybsecurity データベーステーブル

テーブル名	説明	データ?	SAP Sybase IQ によるサポート
sysaudits	監査レコードごとに 1 行。	いいえ	いいえ
sysauditoptions	グローバル監査オプションごとに 1 行。	いいえ	いいえ

他の Sybase データベースとの互換性

この項のトピックを参考にすると、他の SAP Sybase データベースから SAP Sybase IQ への移行が簡単になります。また、これらのトピックは、Adaptive Server または SQL Anywhere と互換性のある SAP Sybase IQ アプリケーションの作成ガイドとしても利用できます。

SAP 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。この項では、SAP Sybase IQ を Adaptive Server および SQL Anywhere と比較します。

SQL Anywhere について

SAP 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 SAP Sybase IQ でもサポートされているとはかぎりません。SAP Sybase IQ のマニュアルを使用してください。

Transact-SQL サポートの概要

SAP Sybase IQ は、SQL Anywhere と同様に、*Transact-SQL* (Sybase Adaptive Server がサポートする SQL のダイアレクト) の大部分のサブセットをサポートしています。

SAP 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ます。アプリケーション、ストアードプロシージャ、バッチファイルの多くは、Adaptive Server と SAP 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 と SAP 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、既存の Adaptive Server アプリケーションを SQL Anywhere データベースまたは SAP Sybase IQ データベースで実行するには、多少の変更が必要です。

次に、SAP 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、SAP Sybase IQ と Adaptive Server 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書かれた文では、以前のバージョンの SAP 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。このような文については、SQL Anywhere と SAP Sybase IQ では 2 種類の SQL ダイアレクトがサポートされています。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。

- 1 つのプロシージャまたはバッチは、Transact-SQL と Watcom-SQL のどちらかの言語のみで実行できます。バッチまたはプロシージャ中では、どちらか 1 つの言語だけで制御文を記述してください。たとえば、ダイアレクトごとに異なったフロー制御文があります。

SAP Sybase IQ は、既存の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ます。

さらに、SAP Sybase IQ では、Transact-SQL のストアー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構造と設定について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。ただし、SAP 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、SAP Sybase IQ にとって意味を持たないテーブルにはローがありません。また、SAP 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、SQL Anywhere、および SAP Sybase IQ のアーキテクチャ

Adaptive Server、SQL Anywhere、SAP Sybase IQ は、それぞれの明確な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。

SAP Sybase IQ は、データウェアハウジングと分析処理専用設計された、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 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に最適な製品です。

このセクションでは、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、互換性のあるデータベース管理のために SAP Sybase IQ と SQL Anywhere に含まれている、Adaptive Server 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 と SAP Sybase IQ や SQL Anywhere とでは異なります。

Adaptive Server 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持っている場合、サーバに接続できます。パーミッションがあれば、そのサーバ上のすべての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。

SAP Sybase IQ には Adaptive Server の master データベースに対応するものではありません。その代わりに、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ているシステム全体にわたる一連のシステムテーブルはありません。各 SAP Sybase IQ データベースサーバは、動的にデータベースを開始および停止できます。このデータベースに対する個別の接続は、ユーザが保持できます。1つのサーバで実行する SAP 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 SAP Sybase IQ は、Transact-SQL のサポートと Open Server のサポートによって、Adaptive Server 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。

領域の割り付けとデバイス管理

Adaptive Server、SQL Anywhere、および SAP Sybase IQ 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 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、スレッシュホールドを使って自動的に領域を追加できます。
- SAP Sybase IQ では、最初に **CREATE DATABASE** 文にロー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** を使用すると、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。SAP Sybase IQ では、ファイルシステム領域も利用できます。

SAP Sybase IQ は Transact-SQL **DISK** 文 (**DISK INIT**、**DISK MIRROR**、**DISK REFIT**、**DISK REINIT**、**DISK REMIRROR**、**DISK UNMIRROR** など) をサポートしません。

- SQL Anywhere は、最初の **CREATE DATABASE** 文がロー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、SAP Sybase IQ と同様です。SQL Anywhere では、dbinit という名前のコマンドユーティリティを使用してデータベースを初期化できます。SAP Sybase IQ には、このユーティリティの拡張バージョンとして、SAP Sybase IQ データベースを初期化する **iqinit** という名前のユーティリティが用意されています。

システムテーブル、カタログストア、IQ メインストア

SAP Sybase IQ データベースは、ジョイントデータストアです。

ジョイントストアの構成を次に示します。

- カタログストアは、システムテーブルとストアドプロシージャを含み、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 メインストアは、一連の SAP 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 SAP Sybase IQ は、カタログ (テーブル、カラムなど) に Adaptive Server とは異なるスキーマを使用します。
- SQL Anywhere と SAP Sybase IQ には、Adaptive Server 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。
- Adaptive Server 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を所有します。
- SQL Anywhere と SAP Sybase IQ では、システム所有者 (ユーザ ID SYS) がカタログオブジェクトを所有します。

注意： ユーザ ID dbo は、SAP Sybase IQ が提供する、Adaptive Server と互換性のあるシステムビューを所有します。

データ型

Adaptive Server、SQL Anywhere、SAP Sybase IQ では、データ型の扱いがそれぞれ異なります。

この項では、データ型の互換性情報について説明します。

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

Adaptive Server、SQL Anywhere、SAP Sybase IQ では、いずれも BIT データ型をサポートしていますが、それぞれに違いがあります。

相違点は次のとおりです。

- SQL Anywhere では 0 または 1 のみを許可します。
- Adaptive Server と SAP 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。

文字データ型の互換性

Adaptive Server、SQL Anywhere、SAP Sybase IQ では、いずれも CHAR データ型と VARCHAR データ型を許可していますが、これらのデータ型の取り扱い方法がそれぞれ異なります。

- SAP Sybase IQ 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 と SAP Sybase IQ は、CHAR (固定長) と VARCHAR (可変長) のデータを区別します。
Adaptive Server では、VARCHAR 値の後続ブランクが削除されます。SAP Sybase IQ では、VARCHAR 値の後続ブランク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。

CHAR または VARCHAR に挿入する場合、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 に挿入すること (暗黙の変換) を許可しています。
- Adaptive Server と SAP 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決まります。

- Adaptive Server 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、最大 2GB の LONG VARCHAR をサポートしています。

他の Sybase データベースとの互換性

- SQL Anywhere は、名前として LONG VARCHAR とその同意語 TEXT をサポートしていますが、Adaptive Server は名前として TEXT のみをサポートし、LONG VARCHAR をサポートしていません。
- SAP Sybase IQ は、最大 32K - 1 バイトの CHAR と VARCHAR をサポートしています。
SAP Sybase IQ はさらに、LONG で 512TB (SAP Sybase IQ ページサイズが 128KB の場合) および 2PB (SAP Sybase IQ ページサイズが 512KB の場合) までサポートしています。VARCHAR.SAP Sybase IQ の LONG VARCHAR データ型については、『非構造化データ分析』を参照してください。
- Adaptive Server は NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR の各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 SAP Sybase IQ は、Unicode を別のデータ型としてではなく、CHAR データ型と VARCHAR データ型の中でサポートしています。
- SAP Sybase IQ と Adaptive Server 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。

参照：

- 文字データ型 (389 ページ)

バイナリデータ型

バイナリデータ型のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

表 130 : バイナリデータ型の許容されるサイズ

データ型	Adaptive Server	SQL Anywhere	SAP 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*	未対応	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 の使用*

*SAP Sybase IQ の LONG BINARY データ型については、『非構造化データ分析』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 と SQL Anywhere では、バイナリデータの表示方法が異なります。

- SAP Sybase IQ は Adaptive Server と SQL Anywhere の両方の表示形式をサポートしています。
- BINARY フィールドに '123' と入力すると、SQL Anywhere の表示形式ではバイト単位の '123' になります。Adaptive Server の表示形式ではニブル単位の '0x616263' になります。

参照：

- バイナリデータ型 (398 ページ)
- NEWID 関数 [その他] (282 ページ)
- STRTOUUID 関数 [文字列] (360 ページ)
- UUIDTOSTR 関数 [文字列] (374 ページ)
- 文字データ型 (389 ページ)

date、time、datetime、timestamp データ型

Adaptive Server、SQL Anywhere、SAP Sybase IQ 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 SAP 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 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 SAP 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。SQL Anywhere/SAP Sybase IQ の datetime データ型のミリ秒単位の精度は、Adaptive Server 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 のデフォルトでは、日付を "MM-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できます。
- SQL Anywhere と SAP Sybase IQ のデフォルトは、ISO の "YYYY-MM-DD" 形式ですが、オプションを設定してこれを変更できます。

時刻の変換は次のとおりです。

- Adaptive Server 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、秒の小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。

他の Sybase データベースとの互換性

- SQL Anywhere と SAP Sybase IQ では、デリミタ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は、次のようになります。

- Adaptive Server と SAP 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 データベースから取得した TIME 値と DATETIME 値は、**INSERT...LOCATION** を使用して DATETIME カラムがある SAP Sybase IQ テーブルに挿入すると、変更されます。**INSERT...LOCATION** 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 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、SAP 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
```

Adaptive Server の日時値と時刻値との互換性

INSERT...LOCATION を使用して Adaptive Server 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 データベース内の DATETIME 値が '2012-11-08 10:37:22.823' であるとします。**INSERT...LOCATION** を使用してこの値を取得し、SAP Sybase IQ に格納すると、値は '2012-11-08 10:37:22.823333' になります。

BIGTIME と BIGDATETIME のサポート

SAP Sybase IQ は、コンポーネント統合サービス (CIS) と **INSERT...LOCATION** について Adaptive Server データ型の BIGTIME と BIGDATETIME をサポートしていません。

- Adaptive Server とのコンポーネント統合サービス — aseodbc サーバクラスプロキシテーブルは、データ型 BIGTIME と BIGDATETIME のカラムを含む Adaptive Server テーブルにマップされます。
Adaptive Server テーブルにマップされるプロキシテーブルの作成時に、マッピングが指定されなかった場合、BIGDATETIME カラムはデフォルトで TIMESTAMP カラムにマップされます。BIGTIME カラムはデフォルトで TIME カラムにマップされます。

asejdbc サーバクラスは、BIGTIME と BIGDATETIME のデータ型をサポートしていません。

- **INSERT...LOCATION — INSERT...LOCATION** コマンドは、データ型 BIGTIME と BIGDATETIME のカラムを含む Adaptive Server テーブルから SAP Sybase IQ テーブルにデータをロードします。

SAP Sybase IQ は、Adaptive Server のデータ型 BIGTIME を SAP Sybase IQ のデータ型 TIME に挿入します。

SAP Sybase IQ は、Adaptive Server のデータ型 BIGDATETIME を SAP Sybase IQ のデータ型 DATETIME、DATE、TIME、および TIMESTAMP に挿入します。

数値データ型

Adaptive Server、SQL Anywhere、SAP Sybase IQ では、デフォルトの精度と位取りがそれぞれ異なります。

- Adaptive Server 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。
- SAP 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、SAP 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。

text データ型

TEXT データのサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server は最大 2GB の LONG VARBINARY (SQL Anywhere の場合は LONG BINARY) と TEXT をサポートしています。SQL Anywhere は LONG VARBINARY をカラム型としてサポートしていませんが、同じ目的で LONG BINARY を使用します。SQL Anywhere は最大 2GB の LONG BINARY と TEXT をサポートしています。
- SAP Sybase IQ は最大 32KB - 1 の VARCHAR をサポートしています。SAP 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。SAP Sybase IQ の LONG VARCHAR データ型については、『非構造化データ分析』を参照してください。

image データ型

IMAGE データのサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server と SQL Anywhere は最大 2GB の IMAGE をサポートしています。
- SAP Sybase IQ は、LONG BINARY 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。SAP Sybase IQ の LONG BINARY データ型については、『非構造化データ分析』を参照してください。

Java データ型

Adaptive Server では、データベースで Java データ型を使用できます。SQL Anywhere と SAP Sybase IQ では使用できません。

データ定義言語

Adaptive Server、SQL Anywhere、SAP Sybase IQ では、データベースとデータベースオブジェクトの作成方法がそれぞれ異なります。

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

Interactive SQL を使用して、Transact-SQL と互換性のあるデータベースを作成します。

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とは、データ、識別子、パスワードに対する大文字と小文字の区別のことです。

データの大文字と小文字の区別

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。

SAP Sybase IQ のデータを比較する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。SAP 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 での大文字と小文字の区別は、Adaptive Server システムにインストールされているソート順によって異なります。シングルバイト文字セットを使用している場合は、Adaptive Server 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

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。

SAP 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 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 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。SAP Sybase IQ では、大文字と小文字を区別しません。

ユーザ ID とパスワードの大文字と小文字の区別

パスワードの大文字と小文字の区別は、他の識別子とは異なります。

SAP Sybase IQ と SAP Sybase SQL Anywhere® では、データベース自体の大文字小文字の区別の設定にかかわらず、新しく作成されたデータベース内のパスワードではすべて大文字と小文字が区別されます。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場合、SAP Sybase IQ および SQL Anywhere でのパスワードの大文字と小文字の区別は、次のように決まります。

- 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と小文字の区別が有効になります。

SAP 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の確保

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。

ネームスペース外では重複した名前も許可されます。データベースオブジェクトによっては、Adaptive Server と、SQL Anywhere および SAP Sybase IQ で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。

- SAP Sybase IQ と SQL Anywhere では、テーブル名は、特定の所有者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 の両方が employee という名前のテーブルを作成できます。この場合、完全修飾名 user1.employee と user2.employee を使用することによって、ユニーク性が確保されます。
- Adaptive Server では、テーブル名は、データベース内で、また所有者に対して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。ある1つのテーブル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、どの2つのテーブルも、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、所有者が同じかどうかに関係なく、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を変更する

SAP 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。SQL Anywhere では、**ALTER INDEX** 文を使用して、インデックス、外部キーロール名、および外部キーの名前を変更できます。Adaptive Server 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 文使用時の考慮事項

互換性のあるテーブルを作成するときは、NULL の取り扱い、検査制約、参照整合性、デフォルト値、カラムの識別、計算カラム、テンポラリテーブル、テーブルのロケーションの互換性を考慮してください。

カラム内の NULL

互換性のある NULL の取り扱い

- SQL Anywhere と SAP 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 になると想定します。この動作を変更するには、データベースオプション **ALLOW_NULLS_BY_DEFAULT** で、Transact-SQL との互換性を OFF に設定します。

- SQL Anywhere と SAP Sybase IQ では、BIT カラムが NULL になれないと想定します。
- Adaptive Server では、NULL が宣言されていない限り、カラムが NULL になれないと想定します。

検査制約

SAP 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。

SAP Sybase IQ では、ユーザ定義関数、プロキシテーブル、非 SAP 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は許可されません。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。SAP Sybase IQ では、次を含む検査制約は許可されません。

- 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 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。

SAP 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

整合性の適用について次のようなアクションがサポートされています。

- SQL Anywhere では、次のすべての ANSI アクションをサポートしています。SET NULL、CASCADE、DEFAULT、RESTRICT。
- Adaptive Server では、次の 2 つのアクションをサポートしています。SET NULL、DEFAULT。

注意： Adaptive Server 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。

- SAP Sybase IQ では RESTRICT アクションだけがサポートされています。
- SAP Sybase IQ では NOT NULL FOREIGN KEY はサポートされていません。
- SAP Sybase IQ には、カラムが候補キーと外部キーの両方に同時にな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 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。
- SAP 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、SAP 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。

- SAP Sybase IQ では、デフォルト値として、IDENTITY または DEFAULT AUTOINCREMENT をサポートしています。SAP Sybase IQ では、任意の精度の位取り 0 の数値型の identity カラムをサポートしています。カラムは NULL でもかまいません。SAP 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。SAP Sybase IQ 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。明示的な挿入や更新の場合には、データベースオプション `IDENTITY_INSERT` をテーブル名に設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` を使用して作成したテーブルには、Identity/Autoincrement カラムがありません。SAP Sybase IQ のビューに IDENTITY/DEFAULT AUTOINCREMENT カラムを含めることはできません。
- SQL Anywhere は、AUTOINCREMENT のデフォルト値をサポートします。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 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。Adaptive Server の identity カラムは、位取り 0、最大精度 38 の数値データ型に限られます。また、値は正の値である必要があり、データ型の範囲によって制限され、null であってはなりません。Adaptive Server 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。Adaptive Server 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。Adaptive Server のビューに `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、Adaptive Server 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、計算カラムへのインデックスの設定をサポートしています。
- Adaptive Server と SAP Sybase IQ 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、所有者を指定しないで、**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、SAP Sybase IQ が宣言するテンポラリテーブルであり、現在の接続でしか使用できません。

テーブルの配置

テーブルの物理的な配置は、Adaptive Server と SAP Sybase IQ では方法が異なります。SAP Sybase IQ は `ON segment-name` 句をサポートしますが、`segment-name` は SAP Sybase IQ の DB 領域を参照します。

`sp_iqstatus` プロシージャの出力

```
Sybase IQ (TM) Copyright (c) 1992-2013 by SAP AG or an SAP affiliate
company. All rights reserved.
Version: 16.0.0.562/130821/P/Mainline/Sun_x64/OS 5.10/64bit/
2013-08-21 06:15:41
Time Now: 2013-08-21 06:27:14.150
Build Time: 2013-08-21 06:15:41
File Format: 23 on 03/18/1999
Server mode: IQ Server
Catalog Format: 2
Stored Procedure Revision: 1
Page Size: 65536/4096blks/16bpp
Number of Main DB Files: 2
Main Store Out Of Space: N
Number of Cache Dbspace Files: 5
Number of Shared Temp DB Files: 0
Shared Temp Store Out Of Space: N
Number of Local Temp DB Files: 1
Local Temp Store Out Of Space: N
DB Blocks: 1-25600 IQ_SYSTEM_MAIN
DB Blocks: 522208-547807 MainUser
```

```
Cache Dbspace Blocks: 1-5120 ssd_dev_1
Cache Dbspace Blocks: 522208-527327 ssd_dev_2
Cache Dbspace Blocks: 1044416-1049535 ssd_dev_3
Cache Dbspace Blocks: 1566624-1571743 ssd_dev_4
Cache Dbspace Blocks: 2088832-2093951 ssd_dev_5
Local Temp Blocks: 1-25600 IQ_SYSTEM_TEMP
Create Time: 2013-08-21 06:27:05.444
Update Time: 2013-08-21 06:27:14.035
Main IQ Buffers: 1588, 100Mb
Temporary IQ Buffers: 1588, 100Mb
Main IQ Blocks Used: 5250 of 38400, 13%=20Mb, Max Block#: 5313
Cache Dbspace IQ Blocks Used: 197 of 25600, 0%=0Mb, Max Block#: 0
Shared Temporary IQ Blocks Used: 0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used: 65 of 12800, 0%=0Mb, Max Block#: 0
Main Reserved Blocks Available: 12800 of 12800, 100%=50Mb
Shared Temporary Reserved Blocks Available: 0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available: 12800 of 12800, 100%=50Mb
IQ Dynamic Memory: Current: 292mb, Max: 308mb
Main IQ Buffers: Used: 18, Locked: 0
Temporary IQ Buffers: Used: 4, Locked: 0
Main IQ I/O: I: L459/P9 O: C21/D33/P22 D:1 C:100.0
Temporary IQ I/O: I: L320/P0 O: C54/D59/P8 D:50 C:100.0
Other Versions: 0 = 0Mb
Active Txn Versions: 0 = C:0Mb/D:0Mb
Last Full Backup ID: 0
Last Full Backup Time:
Last Backup ID: 0
Last Backup Type: None
Last Backup Time:
DB Updated: 0
Blocks in next ISF Backup: 0 Blocks: =0Mb
Blocks in next ISI Backup: 0 Blocks: =0Mb
IQ large memory space: 2048Mb
IQ large memory flexible percentage: 50
IQ large memory flexible used: 0Mb
IQ large memory inflexible percentage: 90
IQ large memory inflexible used: 0Mb
IQ large memory anti-starvation percentage: 50
DB File Encryption Status: OFF
RLV memory limit: 2048Mb
RLV memory used: 0Mb
```

CREATE DEFAULT 文、CREATE RULE 文、CREATE DOMAIN 文使用時の考慮事項

SAP 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 は、名前付きデフォルトを作成するために Create Default 文および Create Rule 文をサポートしています。
- SQL Anywhere と SAP 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 文使用時の考慮事項

トリガのサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- 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 は、文レベルのトリガのみをサポートしています。
- SAP Sybase IQ はトリガをサポートしていません。

注意：トリガは、実質的にはストアー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書き込むのに利用することもできます。

CREATE INDEX 文使用時の考慮事項

CREATE INDEX の構文は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ多少異なります。

- Adaptive Server と SQL Anywhere 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 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 の **CREATE INDEX** 文は SQL Anywhere でも機能しますが、SQL Anywhere は **FILLFACTOR**、**IGNORE_DUP_KEY**、**SORTED_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** の構文は、インデックスコンサルタントで使用する **VIRTUAL** キーワードをサポートしていますが、実際のクエリの実行ではサポートしていません。
- SAP Sybase IQ は、特殊なインデックスタイプとして **LF**、**HG**、**HNG**、**DATE**、**TIME**、**DTTM**、**WD** の 7 つをサポートしています。SAP Sybase IQ は、同じデータ型、精度、位取りの 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。SAP Sybase IQ は、**CREATE INDEX** 文でインデックスタイプ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,...)
```

ユーザ、グループ／ロール、およびパーミッション

Adaptive Server、SQL Anywhere、SAP Sybase IQ では、ユーザおよびロール／グループのモデルにいくつかの相違点があります。

Adaptive Server では、ユーザがサーバに接続するため、各ユーザには、サーバのログイン ID とパスワードに加えて、そのサーバ上でアクセスする各データベースのユーザ ID が必要です。

SQL Anywhere と SAP Sybase IQ では、ユーザにサーバのログイン ID は必要ありません。SQL Anywhere と SAP Sybase IQ のすべてのユーザにデータベースのユーザ ID とパスワードが与えられます。

ユーザロール／グループ

多数のユーザに一度にパーミッションを付与できるように、SQL Anywhere と SAP Sybase IQ はユーザロールをサポートし、Adaptive Server はユーザグループをサポートしています。基本的にロールとグループは同じですが、動作にいくつかの相違点があります。

- Adaptive Server では、ユーザは 1 つのグループのメンバーにしかありません。
- SQL Anywhere と SAP Sybase IQ では、ユーザは複数のロールのメンバーになることができ、ロールの階層構造も許可されています。

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 ロールまたはグループがあります。すべてのユーザが自動的に public ロールまたはグループのメンバーになります。

データベースオブジェクトのパーミッション

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** 文と **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およびビューに設定でき、UPDATE パーミッションをデータベーステーブルの特定の列に設定できます。SQL Anywhere と SAP Sybase IQ では、LOAD および TRUNCATE パーミッションをデータベーステーブルおよびビューに設定することもできます。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 ロールまたはグループに付与します。

- すべての製品で、ストアードプロシージャに EXECUTE パーミッションを与えることができます。
- Adaptive Server は、次のアイテムの **GRANT** と **REVOKE** もサポートしています。
 - オブジェクト：テーブル内のカラム、ビュー内のカラム、ストアードプロシージャ
 - ユーザが可能な操作：**CREATE DATABASE, CREATE DEFAULT, CREATE PROCEDURE, CREATE RULE, CREATE TABLE, CREATE VIEW**
- SQL Anywhere と SAP Sybase IQ では、データベースオブジェクトのパーミッションを付与するには、MANAGE ANY OBJECT PRIVILEGE システム権限を持っている必要があります (これに近い Adaptive Server のパーミッションは、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。この句を使用すると、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ます。ただし、SAP Sybase IQ と SQL Anywhere では、**GRANT EXECUTE** 文で **WITH GRANT OPTION** を使用することはできません。

データベース全体に適用されるパーミッション

Adaptive Server 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が使用されます。

- SQL Anywhere と SAP Sybase IQ では、SYS_AUTH_DBA_ROLE 互換ロールを使用して、ユーザにデータベース内の完全なパーミッションを付与します。ただし、顧客のセキュリティ要件に合わせて SYS_AUTH_DBA_ROLE 互換ロールがロールの階層に移行されていないことを前提とします。
- Adaptive Server のシステム管理者は、このパーミッションをサーバ上のすべてのデータベースに対して使用できます。
- 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 の **SETUSER** 文を使用する必要があります。

ユーザの追加

Adaptive Server でユーザを追加するには、**sp_addlogin** と **sp_adduser** の2つのステップが必要です。

SQL Anywhere と SAP Sybase IQ では、1ステップでユーザを追加できます。

SAP Sybase IQ ログイン管理のストアードプロシージャを使用すると、ユーザを追加または削除する必要がなくても、該当するシステム権限を持つユーザが SAP Sybase IQ のユーザアカウントを追加または削除できます。SAP Sybase IQ ユーザ管理が有効になっていれば、これらの SAP Sybase IQ のユーザアカウントを使用して、ユーザ接続やパスワードの期限を管理することができます。

他の Sybase データベースとの互換性

SQL Anywhere と SAP Sybase IQ では、Adaptive Server のシステムプロシージャを使用してユーザおよびグループを管理できますが、これらのプロシージャの厳密な構文や機能は異なる場合があります。

参照：

- Adaptive Server のシステムプロシージャ (741 ページ)

ロード形式

ロード形式のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- SAP Sybase IQ は、ASCII、BINARY、および 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 は、BCP を通じて ASCII および BINARY ロード形式を扱います。

注意： SAP 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 互換のオプション

SAP Sybase IQ のデータベースオプションを設定するには、**SET OPTION** 文を使用します。

『リファレンス：文とオプション』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

クエリ要件は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

移植可能な SQL を記述するための一般的なガイドライン

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するのは間違っていることもあります。

次に、互換性のある SQL を記述するときの一般的なガイドラインを示します。

- 複数のデータベース管理システムで使用するために SQL を記述する場合は、SQL 文をできるだけ明示的にします。

- 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きます。
- 演算子の優先順位のデフォルトをそのまま適用する場合でも、カッコを使用して文中での実行順序を明示的に指定します。
- Adaptive Server 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 の規則に従います。
- プロシージャとバッチでは、**BEGIN** 文の直後に変数とカーソルを宣言します。SAP Sybase IQ ではこのようにする必要がありますが、Adaptive Server 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 および SAP Sybase IQ の予約語を識別子としてデータベースで使用しないでください。

互換性のあるクエリの記述方法の基準

SAP Sybase IQ と Adaptive Server の両方のデータベースで実行されるクエリの記述方法には、基準が2つあります。

- クエリ中のデータ型、式、探索条件が互換性を持つこと。
- **SELECT** 文そのものの構文が互換性を持つこと。

SAP Sybase IQ がサポートする Transact-SQL の **SELECT** 文の一部を次に示します。

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"
```

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。

参照：

- Transact-SQL プロシージャ内の変数 (901 ページ)

サブクエリのサポート

SAP Sybase IQ におけるサブクエリのサポートは、現時点では Adaptive Server および SQL Anywhere におけるサポートとは多少異なります。

Adaptive Server と SQL Anywhere は **ON** 句内のサブクエリをサポートしていますが、SAP Sybase IQ は現時点ではサポートしていません。

サブクエリ内の **UNION** のサポートは次のとおりです。

- SQL Anywhere は、相関サブクエリと非相関サブクエリの両方で **UNION** をサポートしています。
- SAP Sybase IQ は、非相関サブクエリのみで **UNION** をサポートしています。
- Adaptive Server は、いずれのサブクエリでも **UNION** をサポートしていません。

SQL Anywhere は、文法でスカラ値を使用できるその他の多くの場所でサブクエリをサポートしています。Adaptive Server と SAP Sybase IQ 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 句のサポート

GROUP BY ALL のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server は、**WHERE** 句や **HAVING** 句によって除外されたグループも含め、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** の Transact-SQL 拡張をサポートしていません。

GROUP BY 句の **ROLLUP** と **CUBE** は次のようにサポートされます。

- SAP Sybase IQ と SQL Anywhere は、**GROUP BY** 句内の **ROLLUP** および **CUBE** をサポートしています。
- Adaptive Server は、現時点では **ROLLUP** と **CUBE** をサポートしていません。

Adaptive Server は、**SELECT** 句内のグループ化されていないカラムの射影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。SAP 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。

COMPUTE 句のサポート

COMPUTE のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server は、**COMPUTE** 句内の Transact-SQL をサポートしています。
- Transact-SQL の **COMPUTE** 句は ANSI 標準になく、また、この機能はほとんどのサードパーティ製フロントエンドツールで提供されているため、SQL Anywhere と SAP Sybase IQ ではサポートしていません。

WHERE 句のサポート

WHERE 句での Contains() 述部のサポート、および Like() 述部での末尾の空白の処理は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- SAP 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)。SAP Sybase IQ では、可能な場合、WORD インデックスと TEXT インデックスを使用して検索を最適化します。
- Adaptive Server は **Contains()** をサポートしていません。

Transact-SQL 外部ジョインのサポート

サポートされている外部ジョイン構文は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server は、*= および =* を使用する Transact-SQL 外部ジョイン構文をサポートしています。
- SQL Anywhere と SAP Sybase IQ は Transact-SQL 外部ジョインをサポートしていますが、曖昧になる可能性がある一部の複雑な Transact-SQL 外部ジョインは拒否されます。
- SAP 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：SQL Anywhere と SAP Sybase IQ では、Transact-SQL 外部ジョイン構文は非推奨です。

代わりに使用する ANSI 構文など、Transact-SQL 外部ジョインの詳細については、SAP Community Network で閲覧可能なホワイトペーパー『Transact-SQL 外部ジョインのセマンティックと互換性』を参照してください。SQL Anywhere について記述

されたものですが、このホワイトペーパーの内容は SAP Sybase IQ にも当てはまりません。

ANSI ジョインのサポート

ANSI ジョイン構文のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- SAP Sybase IQ は、現時点では **ON** 句内のサブクエリをサポートしていません。
- Adaptive Server と SQL Anywhere は、**ON** 句内のサブクエリをサポートしています。
- ANSI ジョイン構文を使用するクエリでの **FROM** 句内の **CONTAINS** 条件はサポートされませんが、最良のパフォーマンスが得られない場合があります。**FROM** 句で **CONTAINS** に外部ジョインを使用するのは、各 **CONTAINS** 句からの "score" カラムが必要な場合のみにしてください。それ以外の場合は、**CONTAINS** を **ON** 条件または **WHERE** 句に移動してください。

FULL OUTER JOIN のサポートは次のとおりです。

- SQL Anywhere と SAP Sybase IQ は **FULL OUTER JOIN** をサポートしています。
- Adaptive Server は **FULL OUTER JOIN** をサポートしていません。

Null 比較のサポート

Adaptive Server 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 を意味します。

SQL Anywhere と SAP Sybase IQ では、ANSINULL オプションが OFF に設定されていないかぎり、NULL の比較に ANSI セマンティックが使用されます。このオプションが OFF に設定されている場合、このような比較は Adaptive Server 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 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のサポート

長さゼロの文字列の処理は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server では、長さゼロの文字列は NULL 値として処理されます。Adaptive Server 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 SAP Sybase IQ では、ANSI セマンティックに従って、長さゼロの文字列は NULL ではなく実際の値として処理されます。

HOLDLOCK、SHARED、FOR BROWSE のサポート

HOLDLOCK、SHARED、および FOR BROWSE 構文は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- Adaptive Server は HOLDLOCK、SHARED、および FOR BROWSE 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。
- SAP Sybase IQ はこれらのキーワードをサポートしていません。

SQL 関数のサポート

SAP Sybase IQ は、SQL Anywhere および Adaptive Server とほぼ同じ関数をサポートしていますが、次のような相違点があります。

- Adaptive Server は **PatIndex()** で **USING CHARACTERS | USING BYTES** 構文をサポートしていますが、SQL Anywhere と SAP Sybase IQ はサポートしていません。
- Adaptive Server は **Reverse()** 関数をサポートしていますが、SQL Anywhere と SAP Sybase IQ はサポートしていません。
- Adaptive Server は **Length()** の代わりに構文として **Len()** を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 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。
- SAP Sybase IQ は **Str_Replace()** をサポートしています。
- Adaptive Server と SQL Anywhere は、変更時刻の2つのタイムスタンプを比較する **TSEQUAL()** をサポートしていますが、SAP Sybase IQ は **TSEQUAL()** をサポートしていません (**TSEQUAL** は、SAP Sybase IQ のテーブルレベルのバージョン管理モデルには関係ありません)。
- SAP Sybase IQ は **ROWID()** をサポートしていますが、Adaptive Server と SQL Anywhere はサポートしていません。
- SQL Anywhere と SAP Sybase IQ は、データ型の変換のために、Adaptive Server の **Convert()** に加えて、**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 SAP Sybase IQ は、**Lower()** および **Upper()** の同意語として **Lcase()** および **Ucase()** をサポートしていますが、Adaptive Server はサポートしていません。

他の Sybase データベースとの互換性

- SQL Anywhere と SAP Sybase IQ は、**Locate()** 文字列関数をサポートしていますが、Adaptive Server はサポートしていません。
- SQL Anywhere は、文字列をそれぞれのデータ型に変換できるかどうかをテストする **IsDate()** および **IsNumeric()** 関数をサポートしていますが、Adaptive Server はサポートしていません。SAP Sybase IQ は **IsDate()** をサポートしています。SAP Sybase IQ では **IsNumeric** を使用できますが、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**、および **UIDTOSTR** 関数をサポートしていますが、Adaptive Server はサポートしていません。これらの関数は SAP 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、一部の日付関数など、SAP Sybase IQ と SQL Anywhere で動作が異なる SQL 関数があります。SAP Sybase IQ のデータベースオプション **ASE_FUNCTION_BEHAVIOR** は、SAP Sybase IQ の一部のデータ型変換関数 (**HEXTOINT** や **INTTOHEX** など) の出力が Adaptive Server の関数の出力と一致するように指定します。

OLAP 関数のサポート

現時点では、Adaptive Server は OLAP 関数をサポートしていません。SAP Sybase IQ と SQL Anywhere はサポートしています。

SAP 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、SAP Sybase IQ の OLAP 関数をすべてサポートしています。

現時点では、Adaptive Server 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。

システム関数のサポート

SAP Sybase IQ と SQL Anywhere は、一部の Adaptive Server システム関数をサポートしていません。

SQL Anywhere と SAP Sybase IQ でサポートされていない Adaptive Server システム関数は次のとおりです。

- **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。
- **host_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」の管理に使用する。
- **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 を返す。

他の Sybase データベースとの互換性

- `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す。

ユーザ定義関数のサポート

ユーザ定義関数 (UDF) のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

- SQL Anywhere は、SQL、Java、および C で記述された UDF をサポートしています。
- Adaptive Server は、Java で記述された UDF のみをサポートしています。
- SAP Sybase IQ は、ライセンスが必要なオプションとして外部 C/C++ および Java UDF をサポートしています。SAP Sybase IQ では、CIS クエリ分解を通じて Interactive SQL 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。

日付の算術式の異なる解釈

SQL Anywhere と SAP Sybase IQ は、日付に使用さ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 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、この 2 つからタイムスタンプを作成します。

SELECT INTO 文のサポート

SELECT INTO 文で使用可能なテーブルの種類は、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

SELECT INTO 文の例を見てみましょう。

```
select into table1 from table2
```

- Adaptive Server 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 は **SELECT INTO EXISTING TABLE** もサポートしています。
- SQL Anywhere と SAP Sybase IQ では、`table1` は永久テーブルまたはテンポラリテーブルです。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、指定したカラムの数にかかわらず、常にテンポラリテーブルが作成されます。カラムが 1 つしかないテーブルに対して **SELECT INTO** を実行すると、結果がホスト変数に格納されます。

更新可能なビューのサポート

Adaptive Server 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、より自由度の高いルールを設定するかを制御する **ANSI_UPDATE_CONSTRAINTS** オプションが用意されています。

SAP Sybase IQ では、フラット化できる単一テーブルのビューでのみ **UPDATE** が許可されています。SAP Sybase IQ は **WITH CHECK** をサポートしていません。

UPDATE と DELETE の FROM 句のサポート

Adaptive Server、SQL Anywhere、SAP Sybase IQ はいずれも、複数のテーブルが指定された FROM 句を UPDATE および DELETE でサポートしています。

Transact-SQL のプロシージャ言語の概要

ストアードプロシージャ言語は SQL の一部として、ストアードプロシージャとバッチで使用されます。

SQL Anywhere と SAP 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 のストアードプロシージャの概要

SQL Anywhere と SAP Sybase IQ のストアードプロシージャ言語は、ISO/ANSI ドラフト標準に基づいているため、Transact-SQL ダイアレクトとは多くの点で異なります。

多くの概念と機能は似ていますが、構文が異なります。概念が似ているため、SQL Anywhere と SAP Sybase IQ での Transact-SQL のサポートによって2つのダイアレクト間の自動変換を行うことができます。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、SQL Anywhere と SAP Sybase IQ での Transact-SQL ストアドプロシージャ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 では、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 および SAP Sybase IQ の ISQL ユーティリティと Adaptive Server の isql ユーティリティには、バッチを対話型で実行するためのよく似た機能があります。

プロシージャで使用される制御文はバッチでも使用できます。SQL Anywhere と SAP Sybase IQ は、バッチでの制御文の使用をサポートしています。また、Transact-SQL のように、一連の文を区切りなしで使用できます。この場合、バッチの終わりを示す **GO** 文で終了します。

コマンドファイルに保存されているバッチについては、SQL Anywhere と SAP 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 はパラメータをサポートしていません。

プロシージャとバッチ内の SQL 文

SAP Sybase IQ でサポートされている SQL 文の中には、一方のダイレクトに属し、他方には属さないものがあります。

そのため、2つの言語を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。これは次のことを意味します。

- Transact-SQL 専用の文は、両方の言語に属する文と共に、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 でサポートされていない文を、両方のサーバでサポートされている文と共に、バッチまたはプロシージャ内に含めることができます。
- Transact-SQL 専用の文と SAP Sybase IQ 専用の文を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。

Transact-SQL 互換性が改善され、以前は受け入れられていた誤った SQL 構文がエラーで失敗するようになりました。

IF 文内の式サブクエリ

Adaptive Server と SQL Anywhere は、式サブクエリによって返された変数とスカラー値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_il) from a) then set var4 = 100;
end if; end;
```

CASE 文のサポート

SAP 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 では、**CASE** 文はサポートされていません。CASE 式のみがサポートされています。

参照：

- 式 (29 ページ)

ローレベルのカーソル演算のサポート

Adaptive Server、SQL Anywhere、SAP Sybase IQ は、UPDATE および DELETE でのカーソルの使用をサポートしています。

次の例を考えます。

```
UPDATE WHERE CURRENT OF {cursor}
```

```
DELETE WHERE CURRENT OF {cursor}
```

SAP 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許可されていません。SAP 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 コマンドのサポート

PRINT のサポートは、Adaptive Server、SQL Anywhere、SAP Sybase IQ でそれぞれ異なります。

PRINT の実行結果はクライアントによって異なります。

- Adaptive Server 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 SAP Sybase IQ では、**PRINT** は Open Client および 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 のストアードプロシージャは、SAP Sybase IQ では Interactive SQL を使用して機能します。

注意： Interactive SQL は非推奨の iAnywhere JDBC ドライバを使用します。

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に加えて、SQL Anywhere と SAP Sybase IQ は、Watcom-SQL ダイアレクトと Transact-SQL ダイアレクト間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。

表 131 : 自動変換を有効にする関数

関数	説明
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)	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 プロシージャから返される結果セット

SQL Anywhere および SAP Sybase IQ のプロシージャと Transact-SQL プロシージャは、異なる結果セットを返します。

SQL Anywhere と SAP 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。

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
```

対応する SQL Anywhere または SAP Sybase IQ のプロシージャは次のとおりです。

```
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 = Employee.DepartmentID
END
```

クライアントツールが複数の結果をクライアントに表示する方法には、多少の違いがあります。

- **isql** は、1つのストリームにすべての結果を表示します。
- **Interactive SQL** 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、**Interactive SQL** を再起動するか再接続してください。

Transact-SQL プロシージャ内の変数

SQL Anywhere と SAP Sybase IQ は、Transact-SQL とは異なる方法でプロシージャ内の変数に値を割り当てます。

SQL Anywhere と SAP Sybase IQ は SET 文を使用して、プロシージャ内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** では、ストアー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 **SAP Sybase IQ** では、複合文の最初 (つまり、**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、複合文の実行中持続します。

参照：

- 互換性のあるクエリの記述方法の基準 (889 ページ)

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 言語と Transact-SQL 言語では異なります。

デフォルトでは、Watcom-SQL 言語のプロシージャはエラーが起こると終了し、呼び出した環境に SQLSTATE 値と SQLCODE 値を返します。

EXCEPTION 文を使って、Watcom-SQL スト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。文の後ろにあるこの変数をチェックして、プロシージャから強制的に返すことができます。たとえば、次の文は、エラーが起こると終了させます。

```
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、次のように指定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'
```

次のテーブルは、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 132 : 組み込みプロシージャの戻り値

値	意味
0	プロシージャが正常に実行された
-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

値	意味
-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
-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。

プロシージャ内での **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** 文は実行終了時にエラーを通知しません。代わりに、プロシージャが完了し、**RAISERROR** のステータスコードとメッセージを保存してから、最新の **RAISERROR** を返します。**RAISERROR** を返したプロシージャが他のプロシージャから呼び出された場合、最も外側のプロシージャが終了してから **RAISERROR** が返されます。

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** グローバル変数を検査して、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 に似たエラー処理

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、設定できます。

次のように、**ON EXCEPTION RESUME** 句を **CREATE PROCEDURE** 文に指定します。

```
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ません。このため、これらの2つの句は一緒に使用しないでください。

SQL Anywhere と SAP Sybase IQ の相違点および共通機能

SAP Sybase IQ と SQL Anywhere では、データベースおよびサーバの起動と管理、データベースオプションのサポート、DDL のサポート、DML のサポートが異なります。

詳細については、SAP 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、SAP Sybase IQ のマニュアルで SQL Anywhere の具体的な機能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。

SQL Anywhere サーバおよびデータベースの起動と管理

SAP Sybase IQ と SQL Anywhere では、データベースおよびサーバの起動と管理が異なります。

- SAP Sybase IQ では、SQL Anywhere のネットワークサーバ起動コマンドの代わりに、サーバ起動コマンド **start_iq** を使用します。
- SAP Sybase IQ はパーソナルサーバをサポートしていません。
- SAP Sybase IQ は、SQL Anywhere のサーバ用コマンドラインオプションを多数サポートしていますが、すべてをサポートしているわけではありません。SAP 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。
- SAP 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 (UNIX) が用意されています。
- SAP Sybase IQ と SQL Anywhere では、**BACKUP DATABASE** 文および **RESTORE DATABASE** 文で使用可能な句が異なります。

- SAP Sybase IQ では、マルチプレックス操作についてのみ SQL Remote がサポートされています。

SAP Sybase IQ は、SQL Anywhere のデータベース管理ユーティリティを多数サポートしていますが、すべてをサポートしているわけではありません。

- SAP Sybase IQ でサポートされていない SQL Anywhere のユーティリティは次のとおりです。
 - **backup**
 - **compression**
 - **console**
 - **initialization**
 - **license**
 - **log transfer**
 - **log translation**
 - **rebuild**
 - **spawn**
 - 一部の **transaction log** オプション (-g、-il、-ir、-n、-x、-z)
 - **uncompression**
 - **unload**
 - **upgrade**
 - **write file**
- SAP 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 メインストアを検証するには、**sp_iqcheckdb** を使用します。

SQL Anywhere データ定義言語 (DDL) の相違点

SQL Anywhere と SAP Sybase IQ では、DDL の動作が異なります。

- **ALTER TABLE** 文の **DELETE/DROP** 句または **PRIMARY KEY** 句では、SAP 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を報告します)。SQL Anywhere は常に **CASCADE** アクションを取ります。
- 同様に、SAP Sybase IQ では、関連付けられた外部キー制約がある場合、**DROP TABLE** 文がエラーを報告します。
- SAP Sybase IQ は、DDL 文 **CREATE COMPRESSED DATABASE**、**CREATE TRIGGER**、**SETUSER** をサポートしていません。
- SAP Sybase IQ は文レベルの参照整合性をサポートしていますが、SQL Anywhere は、**CREATE TABLE** 文の **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。

- SAP Sybase IQ テーブルは、SQL Anywhere (またはカタログ) 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、SAP 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。
- SAP Sybase IQ データベースでは、SQL Anywhere テーブルでしかパブリケーションを作成できません。
- **CREATE DATABASE** では、大文字と小文字の区別と照合のデフォルトが異なります。SAP Sybase IQ のデフォルトは **CASE RESPECT** と **ISO_BINENG** 照合です。SQL Anywhere のデフォルトは **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。
- SAP Sybase IQ は、SQL Anywhere でサポートされている **CREATE ENCRYPTED DATABASE** コマンドと **CREATE DECRYPTED DATABASE** コマンドをサポートしていません。『管理：ユーザ管理とセキュリティ』を参照してください。

SQL Anywhere データ操作言語 (DML) の相違点

SAP Sybase IQ では、SQL Anywhere のすべての DML オブジェクトおよび構文がサポートされているわけではありません。

- SAP Sybase IQ は次の DML 文およびプロシージャ文をサポートしていません。
 - **EXPLAIN**
 - **GET DATA**
 - **INPUT**
 - **PREPARE TO COMMIT**
 - **PUT**
 - **READTEXT**
 - **ROLLBACK TRIGGER**
 - **SYSTEM**
 - **UNLOAD TABLE**
 - **VALIDATE TABLE**

注意：一連の抽出オプションは、**UNLOAD TABLE** に似た役割を果たします。

- SAP Sybase IQ は **INSERT...LOCATION** 構文をサポートしていますが、SQL Anywhere はサポートしていません。
- SAP Sybase IQ と SQL Anywhere では、**LOAD TABLE** オプションが異なります。
- SAP Sybase IQ の **OPEN** 文では、**BLOCK** 句および **ISOLATION LEVEL** 句がサポートされていません。
- SAP Sybase IQ はトリガをサポートしていません。
- SAP Sybase IQ と SQL Anywhere では、カーソルのサポートと同様に、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** の使用方法が異なります。

- SAP Sybase IQ のストアードプロシージャから **SELECT** を実行する場合、CIS の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。
- SAP Sybase IQ は、**FROM** 句の `<database name>.<owner>.<table name>` など、Adaptive Server 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、SAP Sybase IQ は、クエリ `SELECT * FROM XXX..TEST` を `SELECT * FROM TEST` と解釈します。

Adaptive Server と SAP Sybase IQ の相違点および共通機能

SAP Sybase IQ と Adaptive Server では、ストアードプロシージャのサポートおよびビューのサポートが異なります。

詳細については、SAP 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 を使用している場合や、SAP Sybase IQ のマニュアルで Adaptive Server の具体的な機能を参照している場合は、Adaptive Server のマニュアルセットを参照してください。

Adaptive Server ストアドプロシージャ

SAP Sybase IQ では、一部のストアードプロシージャがサポートされていません。

SAP Sybase IQ では、次の Adaptive Server ストアドプロシージャがサポートされなくなりました。

- `sp_addserver`
- `sp_configure`
- `sp_estspace`
- `sp_help`
- `sp_helpuser`
- `sp_who`

SAP Sybase IQ では、次のカタログプロシージャがサポートされなくなりました。

- `sp_column_privileges`
- `sp_databases`
- `sp_datatype_info`
- `sp_server_info`

Adaptive Server システムビュー

SAP Sybase IQ では、一部のビューがサポートされていません。

SAP Sybase IQ では、次の Adaptive Server ビューがサポートされなくなりました。

他の Sybase データベースとの互換性

- sysalternates
- sysaudits
- sysauditoptions
- sysconstraints
- syscharsets
- sysconfigures
- syscurconfigs
- sysdatabases
- sysdepends
- sysdevices
- sysengines
- syskeys
- syslanguages
- syslocks
- syslogs
- sysloginroles
- sysmessages
- sysprocedures
- sysprocesses
- sysprotects
- sysreferences
- sysremotelogins
- sysroles
- syssegments
- sysservers
- sysrvroles
- systhresholds
- sysusages

カラム名の違い

Adaptive Server の SYSTYPES ビューで使用されるカラム名は "allownulls" です。
SAP Sybase IQ の SYSTYPES ビューで使用されるカラム名は "allowsnulls" です。

索引

数字

- 3 値的論理
 - NULL 値 82
 - 説明 46

A

- ABS 関数 157
- ACOS 関数 157
- Adaptive Server
 - 互換性 869, 876
- ALL
 - 条件 50
- AND 条件 52
- ANSI_SUBSTRING オプション 363
- ANY
 - 条件 50
- ARGN 関数 158
- ASCII 関数 159
- ASCII 値 159, 171
- ASIN 関数 160
- ATAN 関数 161
- ATAN2 関数 162
- AVG 関数 162

B

- BETWEEN 条件 51
- BFILE 関数
 - 説明 163
- BIGDATETIME データ型
 - 互換性 876
- BIGINTTOHEX 関数 164
- BIGTIME データ型
 - 互換性 876
- BINARY データ型 399
- BIT データ型
 - Transact-SQL 404
 - 互換性 873
- BIT_LENGTH 関数 165
- BLOB データ型
 - LIKE 条件 55

- BYTE_LENGTH 関数 166
- BYTE_LENGTH64 関数
 - 説明 167
- BYTE_SUBSTR 関数
 - 説明 167
- BYTE_SUBSTR64 関数
 - 説明 167

C

- CASE 式 38
 - NULLIF 関数 291
- CAST 関数 168, 415
- CEIL 関数 169
- CEILING 関数 170
- char
 - 単語への分割 640
- CHAR データ型
 - 説明 389
- CHAR 関数 171
- CHAR_LENGTH 関数 171
- CHAR_LENGTH64 関数
 - 説明 173
- CHARACTER VARYING データ型
 - 後続ブランクの削除 392, 873
 - 説明 389
- CHARACTER データ型
 - 説明 389
- CHARINDEX 関数 173
- CHECK 条件
 - Transact-SQL 881
- CHECKPOINT 文
 - チェックポイント中のバックアップ 642
- CLOB データ型 55
- COALESCE 関数 174
- COL_LENGTH 関数 175
- COL_NAME 関数 176
- COMPUTE 句
 - Transact-SQL 891
- CONNECTION_PROPERTY 関数 177

索引

CONTAINS 条件
 TEXT インデックス 53
 WD インデックス 52
CONVERT 関数 178, 415
 整数から日付への変換 181
 日付から整数への変換 181
 日付から文字列への変換 181
 文字列から日付への変換 181
CORR 関数 182
COS 関数 183
 cosine 183
COT 関数 184
COUNT 関数 187
COVAR_POP 関数 185
COVAR_SAMP 関数 186
CPU 使用率
 データベース一貫性チェック 443
CREATE DECRYPTED DATABASE 文 906
CREATE DEFAULT 文
 サポート対象外 884
CREATE DOMAIN 文
 Transact-SQL との互換性 884
 構文 412
 使用 410
CREATE ENCRYPTED DATABASE 文 906
CREATE INDEX 文
 SAP Sybase IQ 885
 Transact-SQL 885
CREATE RULE 文
 サポート対象外 884
CREATE TABLE 文
 Transact-SQL 880
CREATE TRIGGER
 サポートされていない 885
CUBE 演算
 GROUPING 関数 231
CUME_DIST 関数 188
CURRENT DATABASE
 特別値 69
CURRENT DATE
 デフォルト 69
 特別値 69
CURRENT PUBLISHER
 デフォルト 70

 特別値 70
CURRENT TIME
 デフォルト 70
 特別値 70
CURRENT TIMESTAMP
 デフォルト 70
 特別値 70
CURRENT USER
 デフォルト 71
 特別値 71

D

DATALENGTH 関数 189
DATE データ型 405
DATE 関数 191
DATE_ORDER オプション 409
DATEADD 関数 191
DATECEILING 関数 193
DATEDIFF 関数 196
DATEFLOOR 関数 198
DATEFORMAT 関数 201
DATENAME 関数 202
DATEPART 関数 203
DATEROUND 関数 205
DATETIME 関数 208
DAY 関数 208
DAYNAME 関数 209
DAYS 関数 209
DB 領域
 管理 871
DB_ID 関数 211
DB_NAME 関数 212
DB_PROPERTY 関数 213
dbcc
 スレッド使用 443
DBCC
 実行時間 443
 出力 443
 データベースの検証 443
 パフォーマンス 443
DBCC_LOG_PROGRESS オプション 443
dbinit ユーティリティ 871
dbspaces
 読み／書き操作の防止 443

- DDL
 SQL Anywhere 905
 DECIMAL data type 394
 DEGREES 関数 214
 DENSE_RANK 関数 214
 DIFFERENCE 関数 216
 DISK 文
 サポート対象外 871
 DML
 SQL Anywhere 906
 DOW 関数 217
 dropleaks モード 443
 DUMMY テーブル 748
- E**
- ELSE
 IF 式 38
 ENDIF
 IF 式 38
 ERRORMSG 関数
 SQL 構文 218
 EVENT_CONDITION 関数 219
 EVENT_CONDITION_NAME 関数 221
 EVENT_PARAMETER 関数 221
 EXISTS 条件 53
 EXP 関数 223
 EXP_WEIGHTED_AVG 関数 223
- F**
- FIRST_VALUE 関数 225
 FLOAT データ型 395
 FLOOR 関数 227
 FOR BROWSE 構文
 Transact-SQL 893
 FP インデックス
 検証 443
 FROM 句 99
 UPDATE と DELETE 897
- G**
- GETDATE 関数 228
 GRAPHICAL_PLAN 関数 228
 GROUP BY
 互換性 890
 GROUP_MEMBER 関数
 SQL 構文 232
 GROUPING 関数 231
 GUID
 NEWID 関数の SQL 構文 282
 STRTOUUID 関数の SQL 構文 360
 UUIDTOSTR 関数の SQL 構文 374
- H**
- HEXTOBIGINT 関数 232
 HEXTOINT 関数 234
 ASE_FUNCTION_BEHAVIOR オプション
 234
 HOLDLOCK 構文
 Transact-SQL 893
 HOUR 関数 235
 HOURS 関数 236
 HTML_PLAN 240
 HTML_PLAN 関数 240
 HTTP 関数 98
- I**
- identity カラム
 互換性 882
 デフォルト値としてサポートされている
 882
 IF 式 38
 IFNULL 関数 247
 IMAGE データ型 874
 互換性 878
 IN 条件 53
 INDEX_COL 関数 248
 INDEX_PREFERENCE オプション 63
 INSERTSTR 関数 249
 INTEGER データ型 395
 INTTOHEX 関数 250
 ASE_FUNCTION_BEHAVIOR オプション
 250
 IQ メインストア 872
 iq_dummy テーブル 85, 748
 一貫性のある結果を取得 95
 IQ_USE_DIRECTIO 環境変数 12
 IQCHARSET 環境変数 8

索引

iqinit コーティリティ 871
IQLANG 9
IQLANG 環境変数 9
IQLLOGDIR16 環境変数 11
IQTMP16 環境変数 11
IS NULL 条件 54
ISDATE 関数
 SQL 構文 251
ISNULL 関数 252
ISNUMERIC 関数
 SQL 構文 253

J

Java
 ユーザ定義関数 155
Java Runtime Environment
 設定 15
Java データ型
 互換性 878
JAVA_HOME 環境変数 13

L

language_code 9
LAST USER
 特別値 71
LAST_VALUE 関数 256
LCASE 関数 258
LD_LIBRARY_PATH 環境変数 13
LEFT 関数 261
LEN 関数
 SQL 構文 262
LENGTH 関数 263, 267
LIBPATH 環境変数 13
LIKE 条件 55
 LONG BINARY データ 55
 LONG VARCHAR データ 55
 ラジオオブジェクトデータ 55
LIST 関数 264
LOB データの圧縮 576
 設定の表示 578
 設定の変更 576
LOB データ型
 LIKE 条件 55

LOB のデータ圧縮 576
 設定の表示 578
 設定の変更 576
LOB 圧縮
 設定の表示 578
 設定の変更 576
 無効化 576
 有効化 576
LOCATE 関数 268
LOG 関数 270
LOG10 関数 270
LONG BINARY データ型 874, 878
 LIKE 条件 55
LONG VARCHAR データ型 55
LOWER 関数 271
LTRIM 関数 272
LVC セル 443

M

master データベース
 サポート対象外 871
MAX 関数 273
MEDIAN 関数 274
MIN 関数 276
MINUTE 関数 276
MINUTES 関数 277
MOD 関数 278
MONTH 関数 279
MONTHNAME 関数 280
MONTHS 関数 281
MPXServerName カラム 462

N

nchar
 単語への分割 673
NEWID 関数
 SQL 構文 282
NEXT_CONNECTION 関数 284
NEXT_DATABASE 関数 285
NOT 条件 59
NOW 関数 289
NTILE 関数 289

- NULL
 - Transact-SQL との互換性 880
- NULL 値
 - 説明 82
- null 比較
 - Transact-SQL 892
- NULLIF 関数 39, 291
- NUMBER 関数 292
- O**
- OBJECT_ID 関数 293
- OBJECT_NAME 関数 294
- OCTET_LENGTH 関数 295
- OLAP
 - DENSE_RANK 関数 214
 - GROUPING 関数 231
 - NTILE 関数 289
 - PERCENT_RANK 関数 298
 - PERCENTILE_CONT 関数 299
 - PERCENTILE_DISC 関数 302
 - RANK 関数 311
 - STDDEV 関数 352
 - ウィンドウ関数 89
 - ウィンドウ関数の種類 89
 - ウィンドウ指定 89
 - ウィンドウ名 89
 - ウィンドウ集合関数 87
 - ランク付け関数 88
 - ロー比較関数 88
 - 数値関数 88
 - 統計関数 88
 - 分散統計関数 88
- OLAP OVER 句 89
- OLAP 関数
 - Interrow 関数 92
 - 互換性 894
- ON EXCEPTION RESUME 句
 - Transact-SQL 904
- Open Client の設定値 16
- OR キーワード 52
- OVER 句 89
- P**
- PATH 環境変数 14
- PATINDEX 関数 296
- PERCENT_RANK 関数 298
- PERCENTILE_CONT 関数 299
- PERCENTILE_DISC 関数 302
- PI 関数 304
- POWER 関数 304
- PRINT コマンド
 - Transact-SQL 899
- PROPERTY 関数 305
- PROPERTY_DESCRIPTION 関数 306
- PROPERTY_NAME 関数 307
- PROPERTY_NUMBER 関数 307
- Q**
- QUARTER 関数 308
- QUOTED_IDENTIFIER オプション 41
- R**
- RADIANS 関数 309
- RAISERROR 文
 - ON EXCEPTION RESUME 904
 - Transact-SQL 903
- RAND 関数 310
- RANK 関数 311
- REGR_AVGX 関数 312
- REGR_AVGY 関数 313
- REGR_COUNT 関数 315
- REGR_INTERCEPT 関数 316
- REGR_R2 関数 317
- REGR_SLOPE 関数 318
- REGR_SXX 関数 320
- REGR_SXY 関数 321
- REGR_SYY 関数 322
- REMAINDER 関数 323
- REPEAT 関数 324
- REPLACE 関数 325
 - SELECT INTO 文 34, 180, 249, 258, 261, 271, 272, 324, 325, 327, 330, 335, 369, 371, 372
- REPLICATE 関数 327
- resetclocks
 - sp_iqcheckdb オプション 443

索引

REVERSE 関数
SQL 構文 329
RIGHT 関数 330
ROLLUP 演算
GROUPING 関数 231
ROUND 関数 331
ROWID 関数 334
RTRIM 関数 335

S

sa_char_terms ストアドプロシージャ 640
sa_checkpoint_execute システムプロシージャ
642
sa_conn_list システムプロシージャ 648
sa_describe_shapefile システムプロシージャ
654
sa_external_library_unload ストアドプロシージャ
662
sa_get_ldapservers_status システムプロシージャ
665
sa_get_table_definition システムプロシージャ
436
sa_list_external_library ストアドプロシージャ
666
sa_locks システムプロシージャ 666
sa_nchar_terms ストアドプロシージャ 673
sa_performance_diagnostics システムプロシージャ
673
sa_text_index_stats ストアドプロシージャ 711
sa_text_index_vocab ストアドプロシージャ 712
SECOND 関数 336
SECONDS 関数 337
SELECT INTO
REPLACE 関数の使用 34, 180, 249, 258, 261,
271, 272, 324, 325, 327, 330, 335,
369, 371, 372
Transact-SQL 896
SELECT 文
Transact-SQL 889
SET OPTION 文
Transact-SQL 888
SHARED 構文
Transact-SQL 893
SIGN 関数 338
SIGNAL 文
Transact-SQL 903

SIMILAR 関数 338
SIN 関数 339
SMALLDATETIME データ型 405
SMALLMONEY データ型 398
SOME 条件 50
SORTKEY 関数 340
SOUNDEX 関数 345
sp_auth_sys_role_info 716
sp_displayroles システムプロシージャ 718
sp_expireallpasswords システムプロシージャ
721
sp_has_role 関数 346
sp_iq_reset_identity システムプロシージャ 570
sp_iqaddlogin システムプロシージャ 437
sp_iqbackupdetails ストアドプロシージャ 438
sp_iqbackupsummary ストアドプロシージャ
441
sp_iqbrestoreaction ストアドプロシージャ 571
sp_iqcardinality_analysis システムプロシージャ
442
sp_iqcheckdb
allocation モード 443
check モード 443
DBCC_LOG_PROGRESS オプション 443
dropleaks モード 443
resetclocks オプション 443
verify モード 443
構文 443
サンプル出力 443
実行時間 443
出力 443
パフォーマンス 443
sp_iqcheckdb システムプロシージャ 443
sp_iqcheckoptions システムプロシージャ 453
sp_iqcolumn システムプロシージャ 457
sp_iqcolumnmetadata プロシージャ 459
sp_iqcolumnuse システムプロシージャ 460
sp_iqconnection システムプロシージャ 461
sp_iqcontext システムプロシージャ 466
sp_iqcopyloginpolicy システムプロシージャ 469,
534
sp_iqcursorsinfo システムプロシージャ 470
sp_iqdatatype システムプロシージャ 473
sp_iqdbsize システムプロシージャ 476

- sp_iqdbspac システムプロシージャ 477
- sp_iqdbspaceinfo システムプロシージャ 481
- sp_iqdbspaceobjectinfo システムプロシージャ 485
- sp_iqdbstatistics システムプロシージャ 489, 585
- sp_iqdroplogin システムプロシージャ 490
- sp_iqemptyfile システムプロシージャ 491
- sp_iqestdbspaces システムプロシージャ 492
- sp_iqestspace システムプロシージャ 495
- sp_iqevent システムプロシージャ 496
- sp_iqfile システムプロシージャ 499
- sp_iqhelp システムプロシージャ 502
- sp_iqindex システムプロシージャ 509
- sp_iqindex_alt システムプロシージャ 509
- sp_iqindexadvice システムプロシージャ 513
- sp_iqindexfragmentation システムプロシージャ 514
- sp_iqindexinfo
 - インデックス情報の表示 517
- sp_iqindexinfo システムプロシージャ 516
- sp_iqindexmetadata システムプロシージャ 518
- sp_iqindexrebuildwidedata システムプロシージャ 521
- sp_iqindexsize システムプロシージャ 522
- sp_iqindexuse システムプロシージャ 525
- sp_iqlocks システムプロシージャ 529
- sp_iqmergerlvstore システムプロシージャ 534
- sp_iqmodifylogin 535
- sp_iqmodifylogin システムプロシージャ 535
- sp_iqmpxdumptlvlog ストアドプロシージャ 538
- sp_iqmpxfilestatus システムプロシージャ 539
- sp_iqmpxinconnpoolinfo ストアドプロシージャ 540
- sp_iqmpxinheartbeatinfo ストアドプロシージャ 541
- sp_iqmpxincstatistics ストアドプロシージャ 542
- sp_iqmpxinfo ストアドプロシージャ 543
- sp_iqmpxsuspendedconninfo システムプロシージャ 545
- sp_iqmpxvalidate システムプロシージャ 546
- sp_iqmpxversioninfo システムプロシージャ 547
- sp_iqobjectinfo システムプロシージャ 548
- sp_iqpassword システムプロシージャ 551
- sp_iqkeys システムプロシージャ 553
- sp_iqprocedure システムプロシージャ 555
- sp_iqprocparm システムプロシージャ 558
- sp_iqrebuildindex システムプロシージャ 561, 574
- sp_iqrename システムプロシージャ 568
- sp_iqrlvmemory システムプロシージャ 573
- sp_iqsetcompression システムプロシージャ 429
- sp_iqsetcompression ストアドプロシージャ 576
- sp_iqsharedtempdistrib システムプロシージャ 577
- sp_iqshowcompression システムプロシージャ 429
- sp_iqshowcompression ストアドプロシージャ 578
- sp_iqshowpsexec システムプロシージャ 579
- sp_iqspaceinfo システムプロシージャ 581
 - 出力例 581
- sp_iqspaceused システムプロシージャ 582
- sp_iqstatus システムプロシージャ 589
 - 出力例 589
- sp_iqsysmon システムプロシージャ 592
- sp_iqtable システムプロシージャ 614
- sp_iqtablesize システムプロシージャ 618
- sp_iqtableuse システムプロシージャ 620
- sp_iqtransaction システムプロシージャ 621
- sp_iqunusedcolumn システムプロシージャ 625
- sp_iqunusedindex システムプロシージャ 627
- sp_iqunusedtable システムプロシージャ 628
- sp_iqversionuse システムプロシージャ 629
- sp_iqview システムプロシージャ 632
- sp_iqwho システムプロシージャ 634
- sp_iqworkmon システムプロシージャ 637
- sp_objectpermission システムプロシージャ 723
- sp_sys_priv_role_info 738
- SPACE 関数 349
- SQL
 - IQ 言語の違い 421
 - ユーザ定義関数 155
- SQL Anywhere 869
 - 参照整合性制約 881

索引

SQL 関数

ERRORMSG 関数の構文 218
GRAPHICAL_PLAN 関数の構文 228
GROUP_MEMBER 関数の構文 232
GROUPING 関数の構文 231
HTML_PLAN 関数の構文 240
ISDATE 関数の構文 251
ISNUMERIC 関数の構文 253
LEN 関数の構文 262
NEWID 関数の構文 282
REVERSE 関数の構文 329
STR_REPLACE 関数の構文 357
STRTOUUID 関数の構文 360
UUIDTOSTR 関数の構文 374
互換性 893

SQL 構文

CURRENT DATABASE 特別値 69
CURRENT PUBLISHER 特別値 70
CURRENT USER 特別値 71
LAST USER 特別値 71
TIMESTAMP 特別値 72
USER 特別値 73
識別子 26

SQL 文

CREATE DECRYPTED DATABASE 906
CREATE ENCRYPTED DATABASE 906

SQL 文字列の区切り 26

SQL92 準拠 421

SQLCODE

特別値 72

SQLCONNECT 環境変数 14

SQLFLAGGER 関数 349

SQLSTATE

特別値 72

SQRT 関数 351

SQUARE 関数 351

STDDEV 関数 352

STDDEV_POP 関数 354

STDDEV_SAMP 関数 355

STR 関数 356

STR_REPLACE 関数

SQL 構文 357

STRING 関数 359

STRTOUUID 関数

SQL 構文 360

STUFF 関数 361

SUBSTR 関数 362

SUBSTRING 関数 362

SUBSTRING64 関数

説明 365

SUM 関数 365

SUSER_ID 関数 366

SUSER_NAME 関数 367

SYBASE 環境変数 15

SYBASE_JRE 環境変数 15

SYBASE_OCS 環境変数 16

SYSIQBACKUPHISTORY システムビュー 783

SYSIQBACKUPHISTORYDETAIL システムビュー 784

SYSIQDBFILE システムビュー 785

SYSIQDBSPACE システムビュー 786

SYSIQIDX システムビュー 787

SYSIQPARTITIONCOLUMN システムビュー 795

SYSIQLVMERGEHISTORY システムビュー 796

SYSIQRVLOG システムビュー 796

SYSIQTAB システムビュー 797

SYSPARTITION システムビュー 808

SYSPARTITIONKEY システムビュー 808

SYSPARTITIONS システムビュー 809

SYSPARTITIONSCHEME システムビュー 810

SYSROLEGRANTTEXT システムビュー 823

SYSSEPARTITIONKEY システムビュー 831

T

TAN 関数 368

tangent 368

TEXT インデックス

統計 711, 712

TEXT データ型 390, 874

互換性 874, 877

THEN

IF 式 38

TIME データ型 405

TIMESTAMP

データ型 406

- データ型の互換性 875, 876
 - 式の変換 208
 - 特別値 72
 - TINYINT データ型 395
 - TLV ログ
 - 表示内容 538
 - TODAY 関数 368, 748
 - Transact-SQL
 - 参照整合性制約 881
 - システムカタログ 863
 - ジョイン 891
 - バッチ 898
 - ビット処理演算子 35
 - プロシージャ 897
 - プロシージャ言語の概要 897
 - ユーザ定義データ型 414
 - ローカル変数 74
 - 移植可能な SQL の記述 888
 - 外部ジョイン演算子 37
 - 概要 869
 - 結果セット 900
 - 互換性のあるデータベースの作成 878
 - 式 40
 - 説明 869
 - 定数 40
 - 比較条件 45
 - 文字列 41
 - 変数 901
 - Transact-SQL との互換性
 - データベース 878
 - TRIM 関数 369
 - TRUNCNUM 関数 370
- U**
- UCASE 関数 371
 - UNION
 - サブクエリ内 890
 - UNIQUEIDENTIFIER データ型 403
 - UNIQUEIDENTIFIERSTR データ型
 - 説明 389
 - UPPER 関数 372
 - USER
 - 特殊定数 748
 - 特別値 73
 - USER_ID 関数 373
 - USER_NAME 関数 373
 - UUID
 - NEWID 関数の SQL 構文 282
 - STRTOUUID 関数の SQL 構文 360
 - UIDTOSTR 関数の SQL 構文 374
 - UIDTOSTR 関数
 - SQL 構文 374
- V**
- VAR_POP 関数 375
 - VAR_SAMP 関数 377
 - VARBINARY データ型 399
 - VARCHAR データ型
 - 後続ブランクの削除 392, 873
 - 説明 389, 390
 - VARIANCE 関数 378
- W**
- WEEKS 関数 380
 - WEIGHTED_AVG 関数 381
 - WHERE 句
 - Transact-SQL 891
 - WIDTH_BUCKET 関数 383
 - WITHIN GROUP 句 91
- Y**
- YEAR 関数 385
 - YEARS 関数 386
 - YMD 関数 388
- あ**
- アーキテクチャ
 - Adaptive Server Enterprise 870
 - SQL Anywhere 870
 - アークコサイン 157
 - アークサイン 160
 - アークタンジェント 161
 - アークタンジェント率 162
 - アドバイス
 - クリア 513

索引

- ストア 513
 - 表示 513
- アポストロフィ
 - 文字列中 28
- アルファベット
 - 定義 26
- アロケーションマップ
 - リセット 443
- アロケーションユニット 577
- アンロード
 - 外部ライブラリ 662

い

- 一貫性検査
 - パーティション 443
- イベント
 - EVENT_CONDITION 関数 219
 - EVENT_CONDITION_NAME 関数 221
 - EVENT_PARAMETER 関数 221
 - 情報の表示 496, 502
- インスタンス
 - 外部ライブラリ 666
- インストールディレクトリ
 - 説明 3
- インデックス
 - Adaptive Server 885
 - SAP Sybase IQ 885
 - SQL Anywhere 885
 - Transact-SQL 879
- インデックス設定ヒント 62

う

- ウィンドウ関数
 - ウィンドウ関数の種類 89
 - ウィンドウパーティション 89
 - ウィンドウ名または指定 89
- ウィンドウ関数、定義 89
- ウィンドウ指定 89
- ウィンドウの種類 89
- ウィンドウの定義 89
- ウィンドウ名 89
- ウィンドウ集合関数 87

え

- エラー
 - Transact-SQL 902, 904

- エラーメッセージ
 - ERRORMSG 関数 218
 - テキストの取得 218

お

- オブジェクト
 - ID の調査 293
 - 情報の表示 502
 - 名前の調査 294
 - 名前の変更 568
- オブジェクト名の変更
 - sp_iqrename プロシージャ 568
- オプション
 - DBCC_LOG_PROGRESS 443
 - QUOTED_IDENTIFIER 41
 - ログインポリシー 793
 - 論理サーバのログインポリシー 791
- オブティマイザ
 - ユーザ定義の選択性 60
 - 見積もり 60

か

- カーソル
 - Transact-SQL 899
 - ローレベル SAP Sybase IQ 899
 - 情報の表示 470
- カタログ
 - Adaptive Server 互換性 872
 - システムテーブル 745
- カタログストア
 - SAP Sybase IQ 872
 - モニタリング 592
- カッコ
 - SQL 識別子 26
 - データベースオブジェクト 26
- カラム
 - ユーザ定義データ型 411
 - 命名 31
- カラムのデフォルト
 - サポートされていない 882
- カラム長 175
- カラム名 176
- カレントユーザ
 - 環境設定 16

環境変数 IQDIR16 環境変数 8

き

キー

検証 443
情報の表示 553

キーワード

SQL 19
一覧表 19

く

クエリ

Transact-SQL 889

グループ

Adaptive Server 886

グローバルユニーク識別子

NEWID 関数の SQL 構文 282

グローバル変数

リスト 76
互換性 77
説明 73, 75

け

計算カラム

サポートされていない 883

検査制約 881

Transact-SQL との互換性 881
適用 881

検証

インデックス 443
キー 443
分割されたテーブル 443

こ

コーディネータ

サスペンド中のトランザクション 545
共有テンポラリストアの使用状況 577

コードページ

データ記憶領域 391

互換性

Adaptive Server 869
参照整合性制約 881

Adaptive Server のデータ型 876

コメント

コメントインジケータ 80

さ

サーバ

プロパティ 105

サーバ管理

SQL Anywhere と IQ 904

サービス

レジストリエントリ 16

サスペンド中の接続 545

サブクエリ

Adaptive Server 890
SAP Sybase IQ 890
SQL Anywhere 890
SAP Sybase IQ 実装 423
検索条件 47
式内 32
分離 48

サブクエリ述部の分離 48

参照整合性制約

CASCADE がサポートされない 881
互換性 881

し

システムカタログ

Transact-SQL 863
Adaptive Server 互換性 872

システムテーブル

DUMMY 748
ISYSIQLOGICALSERVER 789
ISYSIQLOGINPOLICYLSINFO 790
ISYSIQLSLOGINPOLICIES 790
ISYSIQLSLOGINPOLICYOPTION 791
ISYSIQLSMEMBER 791
ISYSIQLSMEMBERS 792
ISYSIQLSPOLICY 792
ISYSIQLSPOLICYOPTION 793
Adaptive Server 互換性 872
情報の表示 502
説明 745

システムビュー

Adaptive Server 907
SYSCOLUMNS ASE 互換ビュー 763

- SYSINDEXES ASE 互換ビュー 782
- SYSIQBACKUPHISTORY 783
- SYSIQBACKUPHISTORYDETAIL 784
- SYSIQCOLUMN 785
- SYSIQDBFILE 785
- SYSIQDBSPACE 786
- SYSIQFILE 787
- SYSIQIDX 787
- SYSIQINFO 788
- SYSIQITAB 797
- SYSIQLOGICALSERVER 789
- SYSIQLOGINPOLICYLSINFO 790
- SYSIQLSLOGINPOLICIES 790
- SYSIQLSLOGINPOLICYOPTION 791
- SYSIQLSMEMBER 791
- SYSIQLSMEMBERS 792
- SYSIQLSPOLICY 792
- SYSIQLSPOLICYOPTION 793
- SYSIQMPXSERVER 793
- SYSIQMPXSERVERAGENT 794
- SYSIQOBJECTS ASE 互換ビュー 794
- SYSIQPARTITIONCOLUMN 795
- SYSIQLVMERGEHISTORY 796
- SYSIQRVLOG 796
- SYSIQTABCOL 798
- SYSIQTABLE システムビュー 798
- SYSIQVINDEXT ASE 互換ビュー 798
- SYSLOGINS ASE 互換ビュー 804
- SYSOBJECTS ASE 互換ビュー 806
- SYSPARTITION 808
- SYSPARTITIONKEY 808
- SYSPARTITIONS 809
- SYSPARTITIONSCHEME 810
- SYSSUBPARTITIONKEY 831
- SYSTYPES ASE 互換ビュー 852
- SYSUSERS ASE 互換ビュー 859
- SYSVIEW 860
- 統合 749
- システムプロシージャ
 - sa_checkpoint_execute 642
 - sa_conn_list 648
 - sa_describe_shapefile 654
 - sa_get_table_definition 436
 - sa_locks 666
 - sa_performance_diagnostics 673
 - sp_expireallpasswords 721
 - sp_iqaddlogin 437
 - sp_iqbackupdetails 438
 - sp_iqbackupsummary 441
 - sp_iqcardinality_analysis 442
 - sp_iqcheckdb 443
 - sp_iqcheckoptions 453
 - sp_iqclient_lookup 455
 - sp_iqcolumn 457
 - sp_iqcolumnuse 460
 - sp_iqconnection 461
 - sp_iqcontext 466
 - sp_iqcopyloginpolicy 469, 534
 - sp_iqcursorinfo 470
 - sp_iqdatatype 473
 - sp_iqdbsize 476
 - sp_iqdbspaceobjectinfo 485
 - sp_iqdbstatistics 489
 - sp_iqdroplogin 490
 - sp_iqemptyfile 491
 - sp_iqestdbspaces 492
 - sp_iqestspace 495
 - sp_iqevent 496
 - sp_iqfile 499
 - sp_iqhelp 502
 - sp_iqindex 509
 - sp_iqindex_alt 509
 - sp_iqindexadvice 513
 - sp_iqindexsize 522
 - sp_iqindexuse 525
 - sp_iqmodifylogin 535
 - sp_iqmpxfilestatus 539
 - sp_iqmpxinconnpoolinfo 540
 - sp_iqmpxinheartbeatinfo 541
 - sp_iqmpxinstatistics 542
 - sp_iqmpxinfo 543
 - sp_iqmpxsuspendedconninfo 545
 - sp_iqobjectinfo 548
 - sp_iqpassword 551
 - sp_iqpkeys 553
 - sp_iqprocedure 555
 - sp_iqprocparm 558
 - sp_iqrename 568
 - sp_iqrestoreaction 571
 - sp_iqsetcompression 429
 - sp_iqshowcompression 429
 - sp_iqshowpsex 579
 - sp_iqspaceinfo 581
 - sp_iqspaceused 582
 - sp_iqstatistics 585
 - sp_iqstatus 589
 - sp_iqsysmon 592

- sp_iqtable 614
 - sp_iqtablesize 618
 - sp_iqtableuse 620
 - sp_iqtransaction 621
 - sp_iqunusedcolumn 625
 - sp_iqunusedindex 627
 - sp_iqunusedtable 628
 - sp_iqversionuse 629
 - sp_iqview 632
 - sp_iqwho 634
 - sp_iqworkmon 637
 - 情報の表示 502
 - 説明 429
 - システムロール
 - 互換ロールの移行 430
 - システム関数 102
 - COL_LENGTH 175
 - COL_NAME 176
 - CONNECTION_PROPERTY 177
 - DATALENGTH 189
 - DB_ID 211
 - DB_NAME 212
 - DB_PROPERTY 213
 - EVENT_CONDITION 219
 - EVENT_CONDITION_NAME 221
 - EVENT_PARAMETER 221
 - INDEX_COL 248
 - NEXT_CONNECTION 284
 - NEXT_DATABASE 285
 - OBJECT_ID 293
 - OBJECT_NAME 294
 - PROPERTY 305
 - PROPERTY_DESCRIPTION 306
 - PROPERTY_NAME 307
 - PROPERTY_NUMBER 307
 - SUSER_ID 366
 - SUSER_NAME 367
 - Transact-SQL 895
 - USER_ID 373
 - USER_NAME 373
 - システム変数 75
 - ジョイン
 - Transact-SQL 891
 - 外部演算子 37
 - 自動 422
 - ジョイン演算子
 - ANSI 892
 - Transact-SQL 891
 - ジョイン等号条件 67
 - ジョイン等号条件に関するユーザ指定のヒント 67
- ## す
- ストアプロシージャ
 - Adaptive Server Enterprise 907
 - sa_char_terms 640
 - sa_external_library_unload 662
 - sa_list_external_library 666
 - sa_nchar_terms 673
 - sa_text_index_stats 711
 - sa_text_index_vocab 712
 - sp_iqbackupdetails 438
 - sp_iqbackupsummary 441
 - sp_iqclient_lookup 455
 - sp_iqmpxdumptlvlog 538
 - sp_iqmpxinconnpoolinfo 540
 - sp_iqmpxinheartbeatinfo 541
 - sp_iqmpxinstatistics 542
 - sp_iqmpxinfo 543
 - sp_iqrestoreaction 571
 - sp_iqsetcompression 576
 - sp_iqshowcompression 578
 - ストアプロシージャ言語
 - 概要 897
 - スレッド
 - dbcc 443
- ## せ
- セカンダリサーバ
 - 共有テンポラリストアの使用状況 577
 - セキュリティモデル 432
- ## そ
- その他の関数 156
 - ARGN 158
 - COALESCE 174
 - IFNULL 247
 - ISNULL 252
 - NULLIF 291
 - NUMBER 292
 - ROWID 334
 - SQLFLAGGER 349

索引

て

- ディレクトリ構造 3
 - データ
 - 大文字と小文字の区別 878
 - データベース
 - ID 番号の調査 211, 285
 - システムテーブル 745
 - システムプロシージャ 429
 - プロパティ 134
 - プロパティ値 213
 - 大文字と小文字の区別 878
 - 名前の調査 212
 - データベースオブジェクト
 - ID の調査 293
 - 識別 26
 - 名前の調査 294
 - データベースオプション
 - DATE_ORDER 409
 - QUOTED_IDENTIFIER 41
 - データ型
 - Adaptive Server 872
 - IMAGE 874, 878
 - LONG BINARY 874, 878
 - SAP Sybase IQ 872
 - SQL Anywhere 872
 - TEXT 390, 874, 877
 - UNIQUEIDENTIFIERSTR 389
 - バイナリ 398
 - ユーザ定義 410
 - 互換性 416
 - 作成 412
 - 情報の表示 473, 502
 - 数値 394
 - 日付と時刻 405
 - 文字 389
 - データ型の互換性
 - bit データ 873
 - IMAGE データ 878
 - Java データ 878
 - TEXT データ 877
 - バイナリデータ 874
 - 数値データ 877
 - 日時と時刻のデータ 876
 - 日付と時刻のデータ 875
 - 文字データ 873
 - データ型変換
 - BIT から CHAR 418
 - BIT から VARBINARY 417
 - BIT から VARCHAR 418
 - CHAR から BIT 418
 - VARCHAR から BIT 418
 - 関数 94
 - 説明 415, 417
 - データ型変換関数 94
 - BIGINTTOHEX 164
 - CAST 168
 - CONVERT 178
 - HEXTOBIGINT 232
 - HEXTOINT 234
 - INTTOHEX 250
 - テーブル
 - iq_dummy 85
 - Transact-SQL 880
 - 情報の表示 502
 - デバイス
 - 管理 871
 - デフォルト
 - CURRENT DATE 69
 - CURRENT PUBLISHER 70
 - CURRENT TIME 70
 - CURRENT TIMESTAMP 70
 - CURRENT USER 71
 - Transact-SQL 884
 - デフォルト値
 - CURRENT DATABASE 69
 - CURRENT PUBLISHER 70
 - CURRENT USER 71
 - LAST USER 71
 - TIMESTAMP 72
 - USER 73
 - サポートされていない 882
 - テンポラリテーブル
 - Transact-SQL 883
- ### と
- ドメイン 412
 - 説明 410
 - トランザクション
 - サスペンド 545
 - トランザクション管理
 - sp_iqsysmon を使用したモニタリング 592

トリガ
サポートされていない 885

ね

ネームスペース
インデックス 879
ネストした外部ジョイン 891

は

パーセンタイル
NTILE 関数による計算 289
パーティション
一貫性検査 443
パーミッション
Adaptive Server 886
バイト長 295
バイナリデータ
互換性 874
パスワード
期限切れ 721
大文字と小文字の区別 879
追加または変更 551
パターン一致
照合 57
制限 56
説明 55
バックアップ
アクティビティが活発でない間 642
チェックポイント中 642
バックアップ操作
概要 441
バックアップ履歴ファイル
場所 11
バッチ
Transact-SQL の概要 898
作成 898
バッファキャッシュ
sp_iqsysmon を使用したモニタリング 592
パフォーマンス
sp_iqshowpsexo 接続情報 579
sp_iqsysmon プロシージャ 592
モニタリング 592
パブリッシャ
SQL Remote 70

ひ

ビット処理演算子 35
ビット長 165
ビュー
更新可能 897
情報の表示 502
ヒント
インデックス設定 62
実行フェーズ 65, 66

ふ

ファイル
ロケーション 4
プライマリキー
UUID と GUID 282
UUID による一意な値の生成 282
一意な値の生成 282
情報の表示 553
ブランク
後続の削除 392, 873
プリフェッチ
sp_iqsysmon を使用したモニタリング 592
プロシージャ
Transact-SQL 900
Transact-SQL の概要 897
エラー処理 902, 904
パラメータ情報の表示 558
情報の表示 502, 555
変換 900
戻り値 902
プロシージャ言語
概要 897
プロパティ
ID の記述 306
サーバ 105
サーバレベル 305
データベース 134
接続 104
番号の調査 307
名前の調査 307
分割されたテーブル
検証 443

索引

ほ

ポリシー

論理サーバ 792

ま

マルチプレックス

システムプロシージャ 461

バージョン情報の表示 547

設定のチェック 546

論理サーバのメンバーシップ 791, 792

め

メインキャッシュ

検証 443

メモリ

sp_iqsysmon を使用したモニタリング 592

メンバーシップ

論理サーバ 791, 792

も

モニタ

sp_iqsysmon プロシージャ 592

ゆ

ユーザ

削除 490

修正 535

情報の表示 634

追加 437

ユーザ ID

Adaptive Server 886

ユーザ名からの調査 366, 373

大文字と小文字の区別 879

SAP Sybase IQ ユーザ管理

sp_iqdroplogin 490

ユーザ管理

次を参照：ログイン管理

ユーザ指定の条件

クエリ 60

ユーザ指定の条件の選択性 60

ユーザ指定の条件ヒント、ガイドラインと使用法 69

ユーザ指定の条件ヒント文字列 61

ユーザ定義データ型

CREATE DOMAIN 文 412

Transact-SQL 414

説明 410

大文字と小文字の区別 879

ユーザ定義関数 155

互換性 896

ユーザ名

ユーザ ID からの調査 367, 373

ユーティリティ

SQL Anywhere 904

ユニバーサルユニーク識別子

NEWID 関数の SQL 構文 282

ら

ラージオブジェクトデータ

LIKE 条件 55

ランク付け関数 88

り

リスト

外部ライブラリ 666

リストア操作

矛盾のない状態 571

リテラル

最大長 425

リテラル文字列 28, 31

る

ルール

Transact-SQL 884

れ

レジストリエントリ

説明 16

ろ

ロー

カウント 187

- ローカルマシン
 - 環境設定 16
- ローカル変数
 - 説明 73
- ロード形式
 - Transact-SQL Anywhere 888
- ログインポリシー
 - コピー 469, 534
 - ユーザの割り当て 535
 - 論理サーバの割り当て 790
- ログイン管理
 - sp_expireallpasswords 721
 - sp_iqaddlogin 437
 - sp_iqcopyloginpolicy 469, 534
- ロケール
 - 設定 9
- ロック
 - 表示 529
- 論理サーバポリシー
 - システムテーブル 789, 792

