



管理：データベース

SAP Sybase IQ 16.0 SP03

ドキュメント ID：DC02007-01-1603-01

改訂：2013 年 12 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

データベース管理ワークフロー	1
データ定義タスクをスケジュールする場合のガイド ライン	1
データ定義のシステム権限	2
デバイス選択	2
データベースサーバの実行	3
サーバの起動	3
サーバ起動ユーティリティ	4
Windows サービスとしてのサーバ	6
コマンドラインスイッチ	6
サーバアクティビティログ	36
サーバログファイルの命名	39
UNIX ログファイル	39
データベースサーバの停止	39
サーバを停止または再起動する状況	39
サーバの停止	40
サーバを停止するために必要なパーミッショ ン	43
オペレーティングシステムセッションの停止	43
データベースの開始と停止	43
データベースの起動に関するガイドライン	44
データベースの停止	45
iqdemo データベースの起動	45
サーバおよびデータベースへの接続	47
接続ロードマップ	47
接続方法	48
Interactive SQL からデモデータベースへの接 続	49

データベースステータスによるローカル接続 への影響	49
ローカルサーバ上で実行しているデータベー スへの接続	49
UNIX 上の Interactive SQL からデータベースへ の接続	50
UNIX システムからの接続	50
Windows システムからの接続	51
埋め込みデータベースへの接続	52
TCP/IP プロトコル	54
データソースを使用した接続	57
デフォルト接続パラメータ	58
SAP Sybase IQ のユーティリティからの接続	59
サポートされている接続インタフェース	60
接続ステータス	61
接続パラメータの構造	62
接続文字列の形式	63
アプリケーションが接続パラメータを渡す方 法	63
ODBC データソースの接続パラメータ	63
Interactive SQL 接続	65
[接続] ダイアログ ボックス	66
[接続] ダイアログボックスを開く (Interactive SQL)	66
接続用のドライバ	67
ファイルデータソース	67
ODBC アドミニストレータでのファイルデー タソースの作成	68
ファイルデータソースとテキストエディタ	68
他のデータベースからの接続	69
UNIX 上でのポート番号の競合防止	70
接続をテストする方法	70

統合化ログイン	71
統合化ログインの使用方法	73
セキュリティについての考慮事項： 無制限	
データベースアクセス	75
一時的に public オプションを設定してセキュ	
リティを追加する	76
ネットワークから見た統合化ログイン	77
デフォルトの統合化ログインユーザ	77
接続プール	77
テンポラリ接続	80
論理サーバの接続	81
シンプレックスでの接続	81
マルチプレックスでの接続	82
接続を終了する方法	82
接続ログ	83
データベースの作成と管理	85
データベースの作成	85
SQL でのデータベース作成	85
IQ メインストアと IQ テンポラリストアの領域	
管理	95
データベースの削除	96
データベースから他のユーザを切断する	96
Embedded SQL でデータベースから切断する	96
Interactive SQL でデータベースからすべての接続を	
切断する	97
Interactive SQL でのシステムオブジェクトの表示	97
Interactive SQL でのデータベースオプションの設定	97
データ記憶領域の管理	99
領域の割り付け	99
DB 領域のタイプ	100
カタログストア	102
IQ_SYSTEM_MAIN DB 領域	102

他のユーザメイン DB 領域	103
IQ テンポラリ DB 領域	103
IQ 共有テンポラリ DB 領域	103
IQ メッセージファイル DB 領域	105
データベースの領域	105
IQ ストアとテンポラリストアに必要な領域	106
メインストアとテンポラリストアのサイズ調整ガイ ドライン	106
ローデバイスへの Windows アクセスの設定	109
ローデバイスに対するアクセスパーミッションの表 示	110
ローデバイスへのパーミッションの設定	111
Windows でのローデバイス用シンボリックリンクの 設定	113
必要な領域と DB 領域の見積もり	113
DB 領域管理の例	114
データの格納	120
DB 領域ファイルの属性と操作	120
DB 領域の属性と操作	123
読み取り専用および読み込み／書き込み DB 領 域とファイル	126
直接接続ストレージのキャッシュ DB 領域	135
キャッシュ DB 領域の作成	136
キャッシュ DB 領域の削除	137
キャッシュ DB 領域へのストレージの追加	137
キャッシュ DB 領域からのストレージの削除	138
キャッシュ DB 領域の無効化と再有効化	139
キャッシュ DB 領域の DB ファイルの読み込み 専用／読み書きステータスの変更	139
キャッシュ DB 領域のモニタリング	140
データベースオブジェクトの管理	145
SAP Sybase IQ データベース設計	145

ビューの管理	146
ビューの作成	146
ビューの使用におけるガイドライン	147
ビューの使用に関するガイドライン	148
ビューに対するパーミッション	149
ビューを削除する方法	149
システムビューに含まれるビュー情報	149
テーブルの管理	150
テーブル作成のガイドライン	150
テーブル変更のガイドライン	156
テーブル削除のガイドライン	156
プライマリキーの作成	157
外部キーの作成	157
システムビューに含まれるテーブル情報	158
テーブルパーティション	158
制限事項	159
範囲パーティション	161
ハッシュパーティション	162
ハッシュ範囲パーティション	163
インデックスデータ	165
インデックスの概要	165
インデックスタイプ比較	166
包含 (WD) インデックス	167
Compare (CMP) インデックス	169
日付 (DATE) インデックスおよび日時 (DTTM) インデックス	170
高速射影 (FP) インデックス	174
High_Group (HG) インデックス	176
High_Non_Group (HNG) インデックス	178
Low_Fast (LF) インデックス	179
TEXT インデックス	181
時刻 (TIME) インデックス	181

インデックスの選択条件	182
インデックス内のユニークな値の数	183
クエリのタイプ	184
クエリタイプにより推奨されるインデックス ...	185
カラム値により推奨されるインデックス	186
ディスク領域の使用率	187
インデックス内のデータ型	188
インデックスタイプの組み合わせ	189
インデックスの作成	190
カラムインデックスの同時作成	191
インデックスのロードに関するステータス メッセージ	191
データ挿入後のカラムインデックスの追加	191
CREATE INDEX 文のグループの実行	192
インデックスアドバイザの実行	192
インデックスの名前変更	193
インデックスの表示	193
システムビューに含まれるインデックス情報 ...	194
インデックスの削除	194
外部キー制約削除時のインデックスの保持	194
ジョインのパフォーマンスの最適化	194
データ整合性の確保	197
データ整合性の概要	197
データが有効でない場合	197
有効なデータに関するルールと検査	197
データベースの内容を変更する文	198
データ整合性ツール	198
整合性制約を実装する文	199
カラムのデフォルトを使用したデータ整合性の向上	200
サポートされるデフォルト値	200
デフォルト値の制限	200

カラムのデフォルトの作成	201
カラムのデフォルトの変更	201
カラムのデフォルトの削除	201
カラムのデフォルトとしてサポートされる値 ..	202
日付、時刻、タイムスタンプのデフォルト	203
USER デフォルト	204
IDENTITY または AUTOINCREMENT のデフォ ルト	205
NEWID デフォルト	205
NULL デフォルト	206
文字列と数値のデフォルト	206
定数式のデフォルト	206
テーブルとカラムの制約	207
カラムまたはテーブルに対する UNIQUE 制約 ..	207
カラムに対する IQ UNIQUE 制約	208
カラムに対する CHECK 条件	208
ユーザ定義データ型に対する CHECK 条件	209
カラムに対する CHECK 条件	210
テーブルに対する CHECK 条件	211
テーブルの CHECK 条件の削除	211
エンティティと参照整合性	211
クライアントアプリケーションがエンティ ティ整合性に違反する場合	212
参照整合性	213
同時処理	217
参照整合性チェックの無効化	219
システムテーブルの整合性ルール	220
カタログストアインデックスの検証	220
トランザクションとバージョンの管理	221
トランザクション	221
トランザクションアクティビティの表示	222
独立性レベル	227

トランザクションのブロック	228
接続ブロッキングの有効化	229
接続ブロッキングの無効化	230
BLOCKING オプション	230
ブロッキングタイムアウトスレシヨルドの 設定	231
BLOCKING_TIMEOUT オプション	231
トランザクションのブロックによるデッド ロック	232
バージョン	239
テーブルレベルのスナップショットバージョ ン管理	239
ローレベルのスナップショットバージョン管 理	246
トランザクションのバージョン管理タイプの 表示	253
テンポラリテーブルのバージョン管理	253
ロック競合がパフォーマンスに与える影響の調査	254
(廃止予定)-contention	255
チェックポイント、セーブポイント、トランザク ションのロールバック	256
チェックポイント	257
トランザクション内のセーブポイント	257
トランザクションのロールバック	259
システムリカバリ	260
リカバリにおけるトランザクション情報の役 割	260
バックアップの同時実行性	261
トランザクション内のカーソル	261
カーソルとバージョン管理	262
カーソルの依存性	262
カーソルのスクロール	262

ホールドカーソル	263
位置付け操作	263
カーソルのメッセージロギング	264
リモートトランザクション	264
リモートトランザクションの制限	264
プロシージャとバッチの作成	265
プロシージャ	265
プロシージャの作成	265
プロシージャの変更	266
プロシージャの呼び出し	266
プロシージャの削除	266
ストアードプロシージャを実行する権限	266
プロシージャの結果をパラメータとして返す ..	267
プロシージャの結果を結果セットとして返す ..	267
プロシージャ情報の表示	267
プロシージャパラメータ情報の表示	268
プロシージャでのカーソル	268
IQ UTILITIES を使用したストアードプロシ ージャの作成	272
ユーザ定義関数	274
ユーザ定義関数の作成	274
ユーザ定義関数の呼び出し	281
ユーザ定義関数の削除 (SQL の場合)	282
ユーザ定義関数を実行するためのパーミッ ション	282
ユーザ定義関数実行機能の付与 (SQL の場合) ..	283
バッチ	283
制御文	287
複合文	287
複合文での宣言	288
アトミックな複合文	288
プロシージャの構造	289

プロシージャで使用可能な SQL 文	289
プロシージャのパラメータ宣言	289
プロシージャにパラメータを渡す方法	290
関数にパラメータを渡す方法	291
プロシージャの結果	292
RETURN 文を使って値を返す	292
結果をプロシージャのパラメータとして返す 方法	293
プロシージャからの結果セットで返される情 報	295
複数の結果セットを返す	297
プロシージャの変数結果セット	297
エラーと警告の処理	298
デフォルトのエラー処理	299
ON EXCEPTION RESUME を使ったエラー処 理	300
デフォルトの警告処理	302
例外ハンドラ	303
ネストされた複合文と例外処理	305
プロシージャでのトランザクションとセーブポイン ト	307
プロシージャ、ファンクション、トリガ、イベン ト、またはビューの内容を隠す	308
プロシージャ、トリガ、イベント、バッチで使用で きる文	309
バッチで使用される SELECT 文	309
プロシージャ、トリガ、ユーザ定義関数、 バッチで使用される EXECUTE IMMEDIATE	310
スケジュールとイベントの使用によるタスクの自動化	315
スケジュールとイベントの使用によるタスクの自動 化	315

イベント	315
スケジュール	316
システムイベント	317
イベントハンドラ	320
イベントハンドラ	323
イベントやスケジュールに関する情報の取得 ...	324
データベースイベントの監査	327
dbtran データベース管理ユーティリティ	327
AUDITING オプション [データベース]	331
トラブルシューティングのヒント	333
オンラインサポートのソース	333
状況別の解決策	333
サーバのリカバリとデータベースの修復のため の判断フロー	333
サーバの運用上の問題	334
データベース接続の問題	344
Interactive SQL (dbisql) の問題	345
リソースの問題	346
処理の問題	354
パフォーマンスの問題	358
ネットワーク通信のトラブルシューティング	359
互換性のあるプロトコルの使用	359
最新のドライバの使用	359
再起動時のコンピュータの電源切断	360
レイヤごとのプロトコルスタックの診断	360
TCP/IP プロトコルスタックのテスト	360
配線の問題の診断	362
一般的なネットワーク通信の問題の確認	362
診断ツール	363
新しいテンポラリファイルトポロジへのリス トア	363
sp_iqstatus ストアドプロシージャ	363

通知メッセージの解釈	366
sp_iqcheckdb ストアドプロシージャ	371
データベースオプションとサーバ起動オプションの値の確認	372
現在実行中の文の検索	372
サーバ要求のロギング	372
診断情報を収集するための接続	377
通信の問題の診断	377
サポートセンタへの問題の報告	378
getiqinfo を使った診断情報の収集	378
getiqinfo によって収集される情報	380
.srvlog ファイルと .iqmsg ファイル間での接続情報の関連付け	381
サポート Web サイト	382
チェックリスト：サポートセンタに提出する情報	383
付録：接続パラメータと通信パラメータリファレンス	385
接続パラメータ	385
AppInfo 接続パラメータ [Appinfo]	386
AutoPreCommit 接続パラメータ [AutoPreCommit]	388
AutoStart 接続パラメータ [Astart]	388
AutoStop 接続パラメータ [Astop]	389
CharSet 接続パラメータ [CS]	390
CompressionThreshold (COMPTH) 接続パラメータ	390
CommBufferSize 接続パラメータ [CBSize]	391
CommBufferSpace 接続パラメータ [CBSpace]	392
CommLinks 接続パラメータ [Links]	393
ConnectionName 接続パラメータ [CON]	395
ConnectionPool 接続パラメータ [CPOOL]	395
DatabaseFile 接続パラメータ [DBF]	396

DatabaseName 接続パラメータ [DBN]	397
DatabaseSwitches 接続パラメータ [DBS]	398
DataSourceName 接続パラメータ [DSN]	399
DBKEY 接続パラメータ [DBKEY]	399
DisableMultiRowFetch 接続パラメータ [DMRF]	400
EngineName 接続パラメータ [ENG]	400
EncryptedPassword 接続パラメータ [ENP]	401
Encryption 接続パラメータ [ENC]	402
Escape 接続パラメータ [ESCAPE]	403
FileDataSourceName 接続パラメータ [FileDSN]	403
Idle 接続パラメータ [IDLE]	404
Integrated 接続パラメータ [INT]	405
Language 接続パラメータ [LANG]	406
LazyClose 接続パラメータ [LCLOSE]	406
LivenessTimeout 接続パラメータ [LTO]	407
LogFile 接続パラメータ [LOG]	408
LogicalServer 接続パラメータ [LS]	409
LoginRedirection 接続パラメータ [REDIRECT]	409
NewPassword (NEWPWD) 接続パラメータ	410
NodeType 接続パラメータ	411
Password 接続パラメータ [PWD]	412
PrefetchBuffer 接続パラメータ [PBUF]	413
PrefetchRows 接続パラメータ [PROWS]	414
RetryConnectionTimeout (RetryConnTO) 接続 パラメータ	414
ServerName 接続パラメータ [ENG]	415
StartLine 接続パラメータ [START]	415
Unconditional 接続パラメータ [UNC]	416
Userid 接続パラメータ [UID]	416

ネットワーク通信パラメータ	417
Broadcast 通信パラメータ [BCAST]	418
BroadcastListener 通信パラメータ [BLISTENER]	419
ClientPort 通信パラメータ [CPort]	419
DatabaseName 通信パラメータ [DBN]	420
DoBroadcast 通信パラメータ [DBROAD]	421
Host 通信パラメータ [IP]	422
Identity 通信パラメータ	423
Identity_Password 通信パラメータ	424
LDAP 通信パラメータ [LDAP]	425
LocalOnly 通信パラメータ [LOCAL]	425
Logfile 通信パラメータ [LOG]	426
LogFormat 通信パラメータ [LF]	426
LogMaxSize (LSIZE) プロトコルオプション	427
LogOptions 通信パラメータ [LOPT]	428
MaxConnections 通信パラメータ [MAXCONN]	429
MaxRequestSize (MAXSIZE) プロトコルオプ ション	429
MyIP 通信パラメータ [ME]	430
PreFetchOnOpen 通信パラメータ	431
ReceiveBufferSize 通信パラメータ [RCVBUFSZ]	432
SendBufferSize 通信パラメータ [SNDBUFSZ]	432
ServerPort 通信パラメータ [PORT]	433
Sessions 通信パラメータ	434
TDS 通信パラメータ	435
Timeout 通信パラメータ [TO]	435
VerifyServerName 通信パラメータ [Verify]	436
付録：SQL 文とオプションのリファレンス	439

データベースオプション	439
データベースオプションのスコープと継続期 間	439
DEDICATED_TASK オプション	440
LOG_CONNECT オプション	440
MIN_PASSWORD_LENGTH オプション	441
VERIFY_PASSWORD_FUNCTION オプション	442
SQL 文	444
ALTER DBSPACE 文	444
ALTER INDEX 文	449
ALTER LOGICAL SERVER 文	453
ALTER LS POLICY 文	454
ALTER TABLE 文	459
CHECKPOINT 文	477
COMMIT 文	478
CREATE DATABASE 文	480
CREATE DOMAIN 文	493
CREATE INDEX 文	495
CREATE LOGICAL SERVER 文	504
CREATE LS POLICY 文	507
CREATE TABLE 文	511
DROP 文	531
DROP LOGICAL SERVER 文	535
GRANT INTEGRATED LOGIN 文	536
IQ UTILITIES 文	537
LOCK TABLE 文	541
ROLLBACK 文	545
SAVEPOINT 文	546
SET OPTION 文	547
STOP DATABASE 文 [Interactive SQL]	549
索引	553

目次

管理ワークフロー

SAP® Sybase® IQ データベースの作成は、インストールしてからユーザがデータベースを利用できるようになるまでの大きな設定プロセスの一部です。

1. SAP Sybase IQ をインストールし、設定します。

クライアントとサーバ両方の環境をインストールします。『インストールおよび設定ガイド』を参照してください。

2. データベースサーバを起動して接続します。

データベースがデータベースへの通信チャンネルを制御し、開けるようにします。このためには、**start_iq** ユーティリティ、SAP Control Center、Windows の [スタート] メニュー、または設定ファイルを使用します。

3. データベースを作成します。

IQ ストアとカタログストアの両方が作成されます。SAP Control Center、SQL 文 **CREATE DATABASE**、または **iqinit** ユーティリティを使用できます。

4. データベースにテーブルを作成します。

CREATE TABLE 文または SAP Control Center のテーブル作成ウィザードを使用します。

5. テーブルのインデックスを作成します。

CREATE INDEX 文または SAP Control Center のインデックス作成ウィザードを使用します。テーブルの作成時に、特定のインデックスの自動作成もできます。

6. テーブルにデータをロードします。

ファイルからデータのバルクロードを行うには、**LOAD TABLE** 文を使用します。既存のデータベースからデータのローを抽出するには、**INSERT** 文を使用します。『管理：ロード管理』を参照してください。

データ定義タスクをスケジュールする場合のガイドライン

データベースが作成され、他のユーザからアクセスできる状態になった後は、テーブルやインデックスの追加/修正など、さらなるデータ定義操作を実行する必要があります。

データベースの使用が少ない時間にデータ定義操作をスケジュールします。テーブルの作成中や変更中は、短時間ですが、他のすべてのユーザはそのテーブルへの読み込み/書き込みをブロックされます。

データ定義中の同時実行性規則の詳細については、「DDL 操作のロック」を参照してください。

データ定義のシステム権限

データ定義タスクを実行するには、適切なシステム権限が必要です。

- **SERVER OPERATOR** システム権限を持つユーザは、すべてのデータ定義タスクを実行できます。他のユーザにこのシステム権限を付与して特定のタスクを実行させることもできます。たとえば、**SERVER OPERATOR** 権限をほかのユーザに付与するタスクです。
- データベースオブジェクトを作成するには、作成するオブジェクトタイプに対応した **CREATE ANY OBJECT** システム権限が必要です。
- オブジェクトの作成者は、そのオブジェクトの所有者となります。オブジェクトの所有者は自動的にそのオブジェクトに対するすべての操作を実行するシステム権限を持ち、他のユーザにテーブル内の情報を更新する権限を付与できます。

必要なシステム権限の管理権限を持つユーザおよびオブジェクト所有者は、必要なシステム権限を個々のユーザに付与できます。**-gu** コマンドラインオプションを使用して、データベースの作成や削除に必要なパーミッションレベルの設定を行うこともできます。

デバイス選択

データベースやデータベースオブジェクトはデバイス上に格納します。すべてのプラットフォームで、これらのデバイスはオペレーティングシステムファイル、またはローパーティションと呼ばれるディスクの一部です。データベースの作成時に、SAP Sybase IQ はローパーティションかディスクファイルかを自動的に判別します。

運用環境では、ローパーティションへインストールすると、処理パフォーマンスとリカバリ機能が向上します。また、ファイルシステムにインストールするとデバイスを容易に管理できるため、開発環境にはこちらをおすすめします。

注意： カタログストアとトランザクションログはローパーティション上に格納できません。

SAP Sybase IQ データベースサーバの実行

各 SAP Sybase IQ データベースはサーバ上で実行します。使用しているプラットフォームと管理ツールに応じて、SAP Sybase IQ では、サーバの起動と停止を行うためのさまざまな方法が用意されています。

サーバの起動

さまざまな方法でサーバを起動できます。

任意のプラットフォームでのサーバの起動方法

さまざまな方法でデータベースサーバを起動できます。

サーバの起動方法	操作
サーバ起動ユーティリティ	<p>コマンドラインに start_iq コマンドを入力する。</p> <p>構文：</p> <pre>start_iq [options]</pre> <p>複数のコマンドラインオプションを入力する場合、すべてを 1 行で入力する必要がある (改行なし)。</p> <p>便利に使うには、サーバの起動時にデータベース名を指定する。これにより、データベースとサーバを同時に起動できる。</p> <p>データベースを起動しないでサーバを起動する場合はデータベース名を省略する。データベース名を省略する場合は、-n サーバスイッチを使用してサーバ名を明示的に指定する必要がある。データベースを作成またはリストアする場合、または、データベースの使用はクライアントソフトウェアに任せて、サーバの起動と停止のみを制御する場合に、この方法を使用する。</p>
設定ファイル	SAP Sybase から提供される設定ファイルを使用して、サーバと iqdemo データベースを起動する。
サーバ起動コマンド	ショートカットまたはデスクトップアイコンにサーバ起動コマンドを配置する。
ODBC データソース	データソースにサーバ起動行を指定する。

サーバの起動方法	操作
ユーティリティコマンド	ユーティリティコマンドにサーバ起動行を指定する。
Interactive SQL	Interactive SQL から SQL コマンドを発行して、追加のサーバを起動する。
SAP Control Center	SAP Control Center で起動する方法については、SCC またはサイト http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680 の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

Windows サーバの起動方法

さまざまな方法で Windows サーバを起動できます。

サーバの起動方法	操作
コマンドライン	stop_iq コマンドまたは dbstop コマンドを実行する。
デスクトップ	システムトレイの[サーバアイコン]を右クリックして、 Start server name を選択する。
Windows の [スタート] メニュー	[スタート] > [すべてのプログラム] > [Sybase] > [Sybase IQ 16.0] > [Sybase IQ デモ データベースの起動]の順にクリックする。
サービスマネージャ	サーバを Windows サービスとして起動した場合は、次の手順に従う。 <ol style="list-style-type: none"> 1. [コントロールパネル]で、[管理ツール] > [サービス]を選択する。 2. SAP Sybase IQ サービスを選択して、Start the service をクリックする。

サーバ起動ユーティリティ

起動ユーティリティ **start_iq** はすべてのプラットフォームで動作し、一部の文書化されている例外を除いて、必須のパラメータが適切に設定されます。

コマンドラインオプションの例がいくつか示されています。設定ファイルにそのまま書き込めるように、それぞれのオプションを別々の行に分けて記述してあります。これらのオプションを直接コマンドラインに入力する場合は、改行しないで1行に入力してください。

各種のコマンドラインスイッチを選択して、データベースの起動やサーバの停止に必要なパーミッションや使用するネットワークプロトコルなどの機能を指定で

きます。コマンドラインオプションは、SAP Sybase IQ の動作とパフォーマンスをチューニングする方法の1つです。

データベースサーバとデータベースを起動する

サーバを起動するときにデータベース名を指定して、データベースとサーバを同時に起動すると便利です。

デフォルトでは、データベース名がサーバ名として使用されますが、異なる名前をサーバに指定できます。

データベースを起動せずに、データベースサーバだけを起動する

データベースを起動しないでサーバのみを起動する場合は、**start_iq** コマンドにデータベースファイルを指定しないでサーバ名を指定します。

データベース名を省略する場合は、**-n** サーバスイッチを使用してサーバ名を明示的に指定する必要があります。データベースを作成またはリストアする場合、または、データベースの使用はクライアントソフトウェアに任せて、サーバの起動と停止のみを制御する場合に、この方法を使用します。

起動ユーティリティによるサーバの起動

start_iq コマンドは、指定されたサーバをバックグラウンドプロセスとして起動し、データベースが指定されている場合は、そのデータベースを起動します。さらに、必要なすべての起動オプションを設定します。

1. 書き込み可能なディレクトリに変更します。
2. システムプロンプトで次のように入力します。

```
start_iq servername [ database ]
```

データベースを指定しない場合は、**-n <server name>** を使用する必要があります。これを使用しないと、サーバは起動しません。上記の例では、サーバはデフォルトのポート 2638 で起動します。

サーバログ

サーバが起動すると、サーバを起動したユーザのウィンドウまたはコンソールに、サーバが実行されていることを示すメッセージが表示されます。

サーバ環境に関するその他の情報や、起動に失敗した場合は発生した可能性がある問題を示すメッセージも表示されます。

サーバメッセージはすべてサーバログに書き込まれます。デフォルトでは、%IQLLOGDIR16% が Windows プラットフォームへのインストールによって設定され、サーバログは %IQLLOGDIR16%\servername.nnnn.srvlog に保存されます。ここで、*nnnn* は、サーバが起動された回数を示します。**-o** 起動オプションを使用して、サーバログに名前を指定することもできます。

Windows サービスとしてのサーバ

Windows では、サーバをサービスとして実行できます。これによって、マシンをログオフしてもサーバを実行したままにできます。

この機能とその他の Windows 固有の機能については、『インストールおよび設定ガイド』を参照してください。

自動サーバ起動

SAP Sybase IQ Service Manager を使用して、SAP Sybase IQ サーバを起動するサービスを定義します。

次に、ホストが起動したらサーバを自動的に起動するサービスを設定できます。サービスは、非マルチプレックスサーバまたはマルチプレックスサーバのどちらかを起動します。

コマンドラインスイッチ

コマンドラインスイッチを使用して、SAP Sybase IQ 環境を定義します。

コマンドラインスイッチの完全なリストとその詳細なリファレンス情報については、『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」を参照してください。

コマンドラインオプションを使用して設定できる値のいくつかは、**SET OPTION** コマンドを使用して変更できます。

コマンドラインオプションの表示

コマンドラインスイッチでは大文字と小文字が区別されます。使用可能なコマンドラインスイッチをすべて表示するには、オペレーティングシステムのプロンプトで次のコマンドを入力します。

```
start_iq -h
```

設定ファイル

複数セットのコマンドラインスイッチを設定ファイルに格納できます。

コマンドラインスイッチの拡張セットを使用すると、コマンドラインスイッチを設定ファイルに保存し、サーバコマンドラインでそのファイルを呼び出すことができます。コマンドラインの場合と同様に設定ファイルでスイッチを指定しますが、複数行にスイッチを入力できる点が異なります。

設定ファイルリスト

SAP Sybase IQ には、テンプレートとして使用できる設定ファイルが用意されています。

表 1：設定ファイル

ファイル名	ロケーション	使用法
default.cfg	\$IQDIR16/scripts (UNIX)、%IQDIR16% ¥scripts (Windows)	汎用設定ファイル。このファイルは、 start_iq とマルチプレックス起動のデフォルトオプションとして使用される。SAP Sybase IQ では、default.cfg を新しい各データベースディレクトリにコピーし、params.cfg という名前に変更する。default.cfg (scripts ディレクトリ内) への変更は、ファイルの変更後に作成されたすべてのデータベースに継承される。
iqdemo.cfg	\$IQDIR16/data/demo (UNIX)、%IQDIR16%¥ data¥demo (Windows)	デモデータベースを起動し、起動スイッチを推奨デフォルト値に設定する。

設定ファイルの例

事前定義された設定ファイルをテンプレートとして使用して、独自のファイルを作成できます。

たとえば、次の設定ファイルは、Elora という名前のデータベースサーバ上で、データベース **mydb.db** を起動します。キャッシュは 32MB で、チェックポイント間隔は 20 分です。すべてのユーザが、データベースの起動または停止、データのロードを実行できます。ユーザ接続は 10 に制限されていて、カタログページサイズは 4096 バイト、デフォルトのクライアント接続タイムアウトは 72 時間、ネットワークプロトコルは TCP/IP で、指定のポート番号は 1870 です。

```
-n Elora -c 32m -gc 20
-gd all -gl all -gm 10 -gp 4096 -ti 4400 -x tcpip(port=1870) path
¥mydb.db
```

以下を使用して、これらすべてのコマンドラインオプションを実行できます。

```
start_iq @mydb.cfg
```

注意：DBSTOP コマンドでサーバを停止する場合は、サーバの起動時に指定した同じパラメータを指定する必要があります。設定ファイルを使用してサーバを起動すると、サーバを停止するのにパラメータが必要なときにそれらを確認することができます。

SAP Sybase IQ マニュアルにおけるコマンドライン例の書式

SAP Sybase IQ マニュアルのすべての例では、わかりやすくするために、複数のコマンドラインスイッチが複数の行で記載されています。これは、設定ファイルに記述するのと同じ書式です。

これらのスイッチを直接コマンドラインに入力する場合は、1行に入力してください。

必須のコマンドラインオプション

すべてのコマンドラインスイッチがオプションであるとはかぎりません。

『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」で説明されているコマンドラインスイッチの多くはオプションですが、SAP Sybase IQ を効果的に実行するには、**-n** スイッチを指定する必要があります。

注意：すべての64ビットプラットフォームでは、**-c 48M** を指定することをおすすめします。

TCP/IP を使用してサーバに接続する場合は、ネットワーク接続パラメータも指定する必要があります。パラメータ **-x 'tcpip(port=nnnn)'** を設定せずにサーバを起動すると、サーバはデフォルトのTCP/IPポート番号2638を使用します。すでに使用されているポート番号を指定すると、サーバは起動に失敗します。

デフォルトの設定ファイル

デフォルトの設定ファイル (default.cfg) には、必要なすべての起動スイッチが記述されています。

default.cfg ファイル:

- UNIX **start_iq** コマンドで使用される params.cfg ファイルのソースを提供します。
- Windows サービス用のサーバを起動します。

設定ファイル内の **-n servername** スイッチ以外のスイッチを上書きするには、**start_iq** コマンドラインで新しいスイッチを指定します。

デモデータベース用の設定ファイル

デモデータベースの起動に使用する iqdemo.cfg ファイルは、起動スイッチを推奨デフォルト値に設定します。

SAP Sybase IQ で提供しているスクリプトを使用してデモデータベースを作成するときに、このファイルを自動的に作成できます。

命名の制限

引用符で囲む場合でも、データベース名、ユーザ識別子、サーバ名には、ハイフンを含む名前や予約語を使用しないでください。

たとえば、次の名前は許可されていません。

grant

june-1999-prospects

"foreign"

予約語 (キーワード) の完全なリストについては、『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』の「SQL 言語要素」の「キーワード」の「予約語」を参照してください。

サーバ名とデータベース名

サーバに名前を付けるサーバスイッチとして **-n** コマンドラインスイッチを使用する必要があります。これによって、間違ったサーバに誤って接続することが回避されます。

サーバ名とデータベース名は、データベースに接続するときクライアントアプリケーションが使用する接続パラメータに含まれています。Windows では、サーバ名はデスクトップアイコンおよびサーバウィンドウのタイトルバーに表示されます。

注意： 複数のデータベースを起動できますが、SAP Sybase では、SAP Sybase IQ サーバ上では 1 つのデータベースのみ実行することを強くおすすめします。複数のデータベースを実行する必要がある場合は、別個の IQ データベースサーバ上で、異なるポートを使用して、各データベースを起動します。

デフォルトのサーバ名

サーバ名を指定しないと、デフォルトのサーバ名は最初に起動されるデータベースの名前となります。

大文字と小文字の区別と命名規則

サーバ名とデータベース名は、Windows では大文字と小文字が区別されませんが、UNIX では大文字と小文字が区別されます。

サーバとデータベース、他のすべてのデータベースオブジェクトに関して、大文字と小文字の区別などの命名規則を定義しておきます。命名規則に従うことによって問題を防止できる場合があります。

データベース名

データベースファイルの後に `-n` スイッチを指定すると、データベースに名前を付けることができます。

たとえば、次のコマンドラインは、データベースを起動し、データベースに名前を付けます。

```
start_iq -n MyServer mydb.db -n MyDB
```

データベースに名前を付けることによって、ニックネームを使用できます。

サーバ名

最初のデータベースファイルの前に `-n` スイッチを指定して、データベースに名前を付けます。残りのパラメータは `default.cfg` ファイルから追加されます。

たとえば、次のコマンドラインは、`Cambridge_iqdemo` という名前のサーバを起動し、そのサーバで `iqdemo` データベースを起動します。

```
start_iq -n Cambridge_iqdemo iqdemo.db
```

各サーバ名は、ローカルエリアネットワーク (ドメイン) 全体でユニークである必要があります。これによって、間違ったサーバに誤って接続することが回避されます。ホスト名とポート番号の組み合わせでは、サーバをユニークに識別できません。サーバ名にユニークな識別子を追加するという規則を設けるのが有効です。ローカルデータベース接続に共有メモリを使用するマルチユーザネットワーク環境では、ユニークなサーバ名を付けることは特に重要です。この規則によって、他のホストシステムで同じ名前の別のデータベースが起動されている場合でも、すべてのユーザは正しいデータベースに接続できます。

使用されている文字セットにかかわらず SAP Sybase IQ がサーバを検出できるように、サーバ名には 7 ビットの ASCII (下位ページ) 文字のみを使用してください。

サーバ名を指定することによって、データベースがロードされていないデータベースサーバを起動できます。次のコマンドは、データベースがロードされていない `Galt` という名前のサーバを起動します。

```
start_iq -n Galt -gm 10 -gp 4096
```

注意： デフォルトのサーバ名を使用してサーバを起動することはできますが、サーバ名とデータベース名の両方を指定し、それぞれ異なる名前を指定することをおすすめします。このようにすると、サーバとそのサーバ上で実行されているデータベースを区別できます。データベースを起動しないでサーバを起動する場合は、サーバ名を指定する必要があります。

パフォーマンスを制御するコマンドラインオプション

いくつかのコマンドラインオプションは、データベースサーバのパフォーマンスに影響を与えることがあります。

パフォーマンスコマンドラインオプションの多くは、IQ ストアを操作するためのリソースを制御し、パフォーマンスに大きな影響を与える場合があります。カタログストアを操作するためのリソースのみを制御するオプションは、パフォーマンス全体にはあまり影響を与えない場合があります。

参照：

- データベースオプション (439 ページ)

メモリオプション

SAP Sybase IQ では、メモリは次のようなさまざまな目的に使用されています。

- クエリの解析用にディスクから読み込むデータのバッファ
- フラットファイルからロードするときにディスクから読み込むデータのためのバッファ
- 接続、トランザクション、バッファ、データベースオブジェクトを管理するためのオーバヘッド

IQ バッファキャッシュサイズ制御

IQ バッファキャッシュサイズオプションと、サーバを実行中に設定できる他のオプションは、使用可能なメモリ量を決定するために使用します。

メインバッファキャッシュのデフォルトの IQ バッファサイズは 16MB で、テンポラリキャッシュのデフォルトサイズは 8MB ですが、実際のデータベースの使用には小さすぎます。

IQ メインストアとテンポラリストアのバッファキャッシュサイズを、次のいずれかの方法で設定します。

- (推奨) 現在のサーバセッションに対して、サーバワイドなバッファキャッシュサイズを設定するには、データベース起動ユーティリティの **start_iq** オプションの **-iqmc** (メインバッファキャッシュサイズ) と **-iqtc** (テンポラリキャッシュサイズ) を指定します。
- データベースのキャッシュサイズを設定するには、**main_cache_memory_mb** パラメータまたは **temp_cache_memory_mb** パラメータを指定して **sa_server_option** ストアドプロシージャを使用します。

IQ バッファキャッシュサイズをシステムが許容するサイズより大きく設定しても、SAP Sybase IQ はデータベースを開くことができません。

サーバオプション (**-iqmc** と **-iqtc**) では、マシンの物理メモリ量を唯一の制限として、システムで有効なメモリをすべて使用することもできます。このため、64

ビットシステムでは、**sa_server_option** による設定を上書きしない **-iqmc** と **-iqtc** を使用してください。

-iqmc と **-iqtc** で設定されるキャッシュサイズは、起動しているすべてのデータベースに対して、サーバが停止するまで適用されます。そのため、たとえば、**-iqmc** と **-iqtc** を 500 (MB) に設定し、サーバ起動時に 1 つのデータベースを起動し、同じサーバ上で後から別のデータベースを起動する場合、2 つのメインキャッシュと 2 つのテンポラリキャッシュ用に少なくとも 2GB を用意する必要があります。

-iqmc iqsrv16 サーバオプション

メイン IQ ストアのバッファキャッシュサイズを MB 単位で指定します。

構文

-iqmc size

デフォルト

64MB

備考

デフォルト値 64 MB を上書きするこのスイッチは、サーバが起動されてから停止されるまでの間に起動されたすべてのデータベースに適用されます。つまり、サーバの起動時にデータベースを 1 つ起動し、その後別のデータベースを 1 つ起動した場合、メインバッファキャッシュに **-iqmc** の 2 倍のサイズのキャッシュが必要になることとなります。

ラージメモリ所要量は、利用可能な総物理メモリの 3 分の 1 を表します。メインストアに十分なメモリを確保するために、**-iqmc** 起動パラメータの値を利用可能な総物理メモリの 3 分の 1 に設定してください。

サイズ値は、常に単位なしで指定します。たとえば、**-iqmc 32MB** ではなく、**-iqmc 32** と指定します。単位が必要な SQL Anywhere とは異なり、**start_iq** では、単位を指定するとこのスイッチは無視されます。

注意： SAP Sybase IQ サーバでは複数のデータベースを実行しないでください。

-iqtc iqsrv16 サーバオプション

IQ テンポラリストアのキャッシュサイズを MB で指定します。

構文

-iqtc size

デフォルト
64MB

備考

デフォルト値 64 MB を上書きするこのスイッチは、SAP Sybase IQ サーバが起動されてから停止されるまでの間に起動されたすべてのデータベースに適用されます。つまり、サーバの起動時にデータベースを 1 つ起動し、その後別のデータベースを 1 つ起動した場合、テンポラリキャッシュに **-iqtc** の 2 倍のサイズのキャッシュが必要になることとなります。通常、1 つの SAP Sybase IQ サーバで複数のデータベースを実行することは避けてください。

ラージメモリ所要量は、利用可能な総物理メモリの 3 分の 1 を表します。IQ テンポラリストアキャッシュに十分なメモリを確保するために、**-iqtc** 起動パラメータの値を利用可能な総物理メモリの 3 分の 1 に設定してください。

サイズ値は、常に単位なしで指定します。たとえば、**-iqtc 32MB** ではなく、**-iqtc 32** と指定します。単位が必要な SQL Anywhere とは異なり、**start_iq** では、単位を指定するとこのスイッチは無視されます。

IQ_SYSTEM_TEMP DB 領域は、必ず **-iqtc** 以上に設定してください。

IQ バッファ分割制御

デフォルトでは、CPU 数に基づいたバッファ分割は有効です。ロードまたはクエリのパフォーマンスが向上するように分割を調整します。

-iqpartition iqsrv16 サーバオプション

IQ メインバッファキャッシュと一時バッファキャッシュのパーティション数を指定します。

構文

-iqpartition num

備考

IQ メインバッファキャッシュと一時バッファキャッシュのパーティション数を指定します。2 の累乗の値を指定する必要があります。デフォルトでは、指定可能な値は、0 (デフォルト)、1、2、4、8、16、32、64、128、256 です。デフォルトでは、SAP Sybase IQ はパーティション数を *number_of_cpus/8* として自動的に計算し、結果を 2 の累乗に丸めます。最大数は 64 です。キャッシュパーティションの数を調整することによって、パフォーマンスが向上することがあります。-

iqpartition スイッチは、この値を SAP Sybase IQ サーバに設定し、Cache_Partitions データベースオプションで設定された値を上書きします。

- jConnect™ for JDBC™ システムオブジェクトをデータから除外する。jConnect JDBC ドライバを使用してシステムカタログ情報にアクセスするには、jConnect カタログサポートが必要 (デフォルトでインストールされる)。このオプションを指定した場合でも、システム情報にアクセスしないかぎり、JDBC を使用できる。jConnect サポートは、**ALTER DATABASE** 文を使用して後から追加できる。

CACHE_PARTITIONS オプション

メインおよびテンポラリバッファキャッシュに使用するパーティションの数を設定します。

指定できる値

0、1、2、4、8、16、32、64

- **0** – (デフォルト) SAP Sybase IQ は、パーティション数を `number_of_cpus/8` として自動的に計算し、結果を 2 の累乗に丸めます。最大数は 64 です。
- **1** – パーティションは 1 つのみ。この値では分割が無効化されます。
- **2 ~ 64** – パーティションの数。2 の累乗の値を指定する必要があります。

デフォルト

0

スコープ

オプションは、データベース (PUBLIC) レベルでのみ設定できます。

このオプションを設定するには、**SET ANY SYSTEM OPTION** システム権限が必要です。次回、データベースサーバを起動したときに現在のデータベースで有効になります。

備考

バッファキャッシュの分割は、複数の CPU を搭載したシステムでロック競合を減らし、パフォーマンスを向上させることがあります。通常は、システム上の CPU の数を基に SAP Sybase IQ が自動計算した値を使用します。ただし、マルチ CPU 環境でクエリまたはロードのパフォーマンスが予想より悪い場合は、**CACHE_PARTITIONS** の値を変更するとパフォーマンスが向上することがあります。

理想的なパーティションの数は、CPU の数とプラットフォームの両方で決まりません。環境に最も適した設定を判断するには、いくつかの値で試してみる必要があります。

CACHE_PARTITIONS に設定した値は、メインとテンポラリの両方のバッファキャッシュに適用されます。各バッファキャッシュにおいて、パーティションの絶対最大数は 64 です。

-iqpartition start_iq サーバオプションでは、サーバレベルでパーティションの制限を設定します。サーバの起動時に **-iqpartition** を指定すると、この設定が **CACHE_PARTITIONS** の設定よりも優先されます。

パーティションの数は、その他のバッファキャッシュの設定には影響しません。また、IQ モニタが収集する統計情報にも影響しません。すべてのパーティションの統計情報は、合計され、1つの値として報告されます。

例

100 の CPU が搭載されたシステムで、**CACHE_PARTITIONS** を設定しなかった場合、SAP Sybase IQ はパーティションの数を自動的に 16 に設定します。

$100 \text{ cpus} / 8 = 12$ なので、丸めると 16 になります。

この設定では、メインバッファキャッシュのパーティション数とテンポラリキャッシュのパーティション数がそれぞれ 16 になります。

100 の CPU を搭載した同じシステムで、パーティション数を明示的に 8 に設定するには、次のように指定します。

```
SET OPTION "PUBLIC".CACHE_PARTITIONS=8
```

同時ユーザスイッチ

ライセンスによって絶対的な同時ユーザ数が設定されます。この **-gm** スイッチは、特定のサーバの同時ユーザ接続数を制限します。

-gn スイッチによって、カタログストアで使用される実行スレッドの数と複数のユーザが使用する場合のコネクティビティを設定します。このコネクティビティは、すべてのオペレーティングシステムとサーバに適用されます。

Windows では、**start_iq** が、次の式を使用してこのパラメータの値を計算し、設定します。

```
gn_value >= gm_value * 1.5
```

-gn は、**-gm** の値の少なくとも 1.5 倍の値に設定します。最小値は 25 です。スレッドの総数は、プラットフォーム固有の最大スレッド数を超えないようにします。詳細については、「**-iqmt num**」を参照してください。

-gm iqsrv16 データベースサーバオプション

データベースサーバに対する同時接続の数を制限します。

構文

```
iqsrv16 -gm integer ...
```

デフォルト

パーソナルサーバのデフォルト値は 10 です。ネットワークデータベースサーバのデフォルト値は 32766 ですが、この数値は、操作時にサーバで使用される内部テンポラリー接続によって削減されます。

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

サーバの接続制限を定義します。ライセンス契約に許可されている数より大きい値、もしくはメモリ制約を超えた値をここで設定した場合、その値は無効です。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。

DROP CONNECTION システム権限を持つユーザがデータベースサーバに接続して他の接続を削除できるように、データサーバは、接続制限より 1 つ余分に DBA 接続を許可します。

-gn iqsrv16 サーバオプション

複数のユーザが使用する際の接続の数と、カタログストアで使用される実行スレッドの数を設定します。

構文

-gn integer

備考

このパラメータは、すべてのオペレーティングシステムとサーバに適用されます。各接続では、1 つの要求に対して 1 つのスレッドを使用します。要求を処理すると、スレッドは他の接続が使用できるようプールに戻ります。接続は同時に複数の要求を処理できないので、接続が同時に複数のスレッドを使用することはありません。

この規則の例外となるのは、Java アプリケーションがスレッドを使用する場合です。Java アプリケーションでは、各スレッドがデータベースサーバで使われる実行スレッドとなります。

Windows を使用している場合は、**start_iq** でこのパラメータを指定します。パラメータの値を計算するには、次の式を使用します。

```
gn_value >= gm_value * 1.5
```

-gn は、**-gm** の値の少なくとも 1.5 倍の値に設定します。最小値は 25 です。スレッドの総数は、プラットフォーム固有の最大スレッド数を超えないようにします。詳細については、「**-iqmt num**」を参照してください。

同時クエリスイッチ

ユーザ接続数をライセンスで許可されたユーザ数より少ない数に制限するには、**-iqgovern** スイッチを使用して、クエリの使用を制御します。

-iqgovern スイッチは、特定のサーバでの同時クエリ数を制御します。

-iqgovern スイッチを指定することによって、IQ はディスクへのバッファデータのページングを最適化し、メモリの過剰使用を防止できます。**-iqgovern** のデフォルト値は $(2 \times \text{CPU 数}) + 10$ です。場合によっては、いろいろな値を試して最適な値を見つける必要があります。アクティブな接続が多数あるサイトの場合は、**-iqgovern** を多少低めに設定してください。

-iqgovern iqsrv16 サーバオプション

サーバで同時に実行できるクエリの数を設定します。

構文

-iqgovern num

デフォルト

RECOVERY_TIME データベースオプションの設定 (デフォルトでは 2 分)

備考

同時に実行できるクエリ数は、接続の数とは異なります。1 つの接続で複数のオープンカーソルを持つことができます。**-iqgovern** を指定することによって、SAP Sybase IQ でのディスクへのバッファデータのページングを最適化し、メモリの過剰使用を防止できます。このスイッチのデフォルト値は、マシンの CPU 数の 2 倍に 10 を加算した数です。場合によっては別の値、たとえば CPU 数の 2 倍に 4 を加算した数などを使うと、より高いスループットが得られることがあります。接続するユーザが多い場合、特にこの傾向があります。

連結メモリスイッチ

-iqwmem スイッチは、特定の UNIX プラットフォームでのみ「連結メモリ」のプールを作成します。

警告！ このスイッチは、連結メモリを追加する余裕がメモリにある場合にのみ使用します。メモリが十分でないときにこのスイッチを使用すると、パフォーマンスが著しく低下することがあります。

-iqwmem iqsrv16 サーバオプション

HP および Sun の UNIX システムで使用される「連結」メモリプールを作成します。

構文

-iqwmem *size*

備考

このメモリはロックされているため、オペレーティングシステムによるページングはできません。このスイッチは、連結メモリを追加するためのメモリを確保できる場合にのみ使用してください。メモリが十分でないときにこのスイッチを使用すると、パフォーマンスが著しく低下することがあります。

処理スレッド数

SAP Sybase IQ は、プロセスで実行される処理の種類、使用可能な合計スレッド数、さまざまなオプションの設定に基づいて、それぞれ異なるカーネルスレッド数を各ユーザ接続に割り当てます。スレッド数を増やすとパフォーマンスが向上します。

-iqmt スwitchを使用して、SAP Sybase IQ が使用できる処理スレッド数を設定します。

-iqmt iqsrv16 サーバオプション

作成する SAP Sybase IQ スレッドの数を指定します。

構文

-iqmt *num*

備考

デフォルトは、最初の 4 CPU に対して CPU ごとに 60 スレッド、残りの CPU に対して CPU ごとに 50 スレッド、それにシステムで使用するための 3 スレッドとデータベース接続およびバックグラウンドタスクに必要なスレッドを加えた数です。たとえば、CPU が 12、接続スレッドが 10 のシステムの場合は、 $60 * 4 + 50 * (\text{numCPUs} - 4) + \text{numConnections} + 6 = 656$ と計算されます。

num の最小値は *num_conn* + 3 です。

サーバスレッドの総数は、4096 を超えてはいけません。

プロセッサの数

マルチプロセッサマシンでは、**-gt** オプションを使用して、データベースサーバでカタログストアの操作に使用されるプロセッサの数を設定します。

デフォルトでは、使用可能なすべてのプロセッサが使用されます。

-gt iqsrv16 データベースサーバオプション

使用できる物理プロセッサの最大数を設定します (ライセンスされたプロセッサの数を上限とする)。このオプションは、マルチプロセッサシステムでのみ役立ちます。

構文

```
iqsrv16 -gt num-processors ...
```

指定可能な値

- **num-processors** - 1 と、次に示す項目の最小値との間の整数で値を設定できます。

コンピュータ上の物理プロセッサの数

サーバがライセンスを受けている CPU の最大数 (CPU がライセンスされている場合)

-gt オプションに範囲外の値を指定すると、下限または上限の値が設定されます。

備考

パーシートライセンスでは、ネットワークデータベースサーバは、コンピュータで使用可能なすべての CPU を使用します。CPU ベースのライセンスの場合、ネットワークデータベースサーバはライセンスを受けたプロセッサ数のみ使用可能です。ネットワークデータベースサーバが使用できる CPU 数は、SAP Sybase IQ のエディションによって制限される場合もあります。

-gt オプションの値を指定すると、データベースサーバはアフィニティマスクを調整し (ハードウェアプラットフォームでサポートされている場合)、指定した物理プロセッサの数のみを使用して実行するようにデータベースサーバを制限します。データベースサーバが n 個のプロセッサのライセンスを受けている場合、デフォルトでは、サーバは n 個の物理プロセッサにおいて、すべての論理プロセッサ (ハイパースレッドとコア) 上で実行します。**-gtc** オプションを使用すると、この動作にさらに制限を加えることができます。

-gt オプションは、**-gta** オプションとは一緒に使用できません。

カタログストアのキャッシュサイズ

-c スイッチを使用して、カタログストアのキャッシュのメモリ量を設定します。

start_iq コマンド、`iqdemo.cfg` および `default.cfg` 設定ファイルは、-c パラメータを、64 ビットシステムでは 48MB、32 ビットシステムでは 32MB に設定します。

10000 より小さいキャッシュサイズは、K バイト単位とみなします (1K = 1024 バイト)。10000 以上のキャッシュサイズはバイト単位とみなします。キャッシュサイズは *mK* または *mM* として指定できます。

start_iq、`iqdemo.cfg`、または `default.cfg` を使用せずにサーバを起動すると、デフォルトの初期キャッシュサイズは、物理メモリ量、オペレーティングシステム、データベースファイルのサイズに基づいて計算されます。使用可能なキャッシュが足りない場合、データベースサーバは、カタログ用の追加のキャッシュを確保します。

警告！ カタログストアのキャッシュサイズを制御するには、設定ファイル (`.cfg`) またはサーバ起動時の UNIX コマンドラインで、次のいずれかを実行する必要があります。ただし、両方は実行しないでください。

- -c パラメータを設定する
- -cl パラメータと -ch パラメータを使用して、カタログストアのキャッシュサイズの特定の上限と下限を設定する

これらのパラメータの別の組み合わせは指定しないでください。

IQ ストアのキャッシュサイズは、カタログキャッシュサイズには依存しません。

-c iqsrv16 サーバオプション

カタログストアページとその他のサーバ情報のキャッシュに使うために予約する初期メモリを設定します。

構文

-c cache-size [k|m|g|p]

備考

データベースページをキャッシュに入れるためのメモリがキャッシュ内に別途確保されている場合は、そのメモリがデータベースサーバで使用されます。10000 より小さいキャッシュサイズは KB 単位とみなします (1K = 1024 バイト)。10000 以上のキャッシュサイズはバイト単位とみなします。キャッシュサイズ *nK*、*nM*、または *nP* (1M = 1024 KB) も設定できます。ここで、P は物理システムメモリの割合です。

default.cfg ファイルの `-c` と `start_iq` のデフォルト値は、Windows プラットフォームでは 32MB (`-c 32M`)、UNIX プラットフォームでは 48MB (`-c 48M`) です。このデフォルトを使用するか、またはより大きな値に `-c` を設定します。

P の代わりに % も使用できますが、UNIX 以外のオペレーティングシステムでは、ほとんどの場合 % を環境変数のエスケープ文字として使用するため、% 文字をエスケープする必要があります。たとえば、システムの物理メモリの 20 パーセントを使用するには、次のように指定します。

```
start_iq -c 20%% ...
```

`-c` を、`-ch` または `-cl` と同じ設定ファイルまたはコマンドラインで使用しないでください。関連情報については、`-ch cache-size` オプションおよび `-ca 0` オプションを参照してください。

start_iq サーバオプション `-c` が指定されていない場合のカタログストアの初期キャッシュサイズ

start_iq サーバオプション `-c` スイッチの値を指定しない場合 (指定するには、コマンドラインを使用するか **start_iq** のデフォルトを適用)、カタログストアの初期キャッシュ割り付けがデータベースサーバで計算されます。

サーバでは、以下のようにキャッシュ割り付けが計算されます。

- データベースサーバは、デフォルトの最小キャッシュサイズとして 32MB を使用します。
- データベースサーバは、ランタイムのデフォルト最小キャッシュサイズを計算します。これは、次のうちの小さい方の数値になります。
 - マシンの物理メモリの 25%。
 - コマンドラインで指定されたメインデータベースファイルの合計サイズ。メインデータベースファイル以外の `dbspace` ファイルは、この計算の対象とはなりません。ファイルを指定しないと、この値は 0 になります。
- データベースサーバは、計算された 2 つの値のうち、大きい方のサイズを割り付けます。

CPU 数のスイッチ

SAP Sybase IQ 起動コマンドの `-iqnumbercpus` スイッチは、リソースを計画するために、IQ で使用可能な CPU の数を指定して物理的な CPU の数を上書きします。

このパラメータのデフォルト値は CPU の総数ですが、指定できる値は 1 ~ 512 です。

注意： `-iqnumbercpus` は、次のマシンのみで使用してください。

- Intel® の CPU を搭載し、ハイパースレッディングが有効になっているマシン。 `-iqnumbercpus` には、使用可能な CPU の数を指定します。

- オペレーティングシステムのユーティリティを使って、SAP Sybase IQ で使用可能な CPU がマシンにある CPU の一部に制限されているマシン。

-iqnumbercpus を、使用可能な CPU 数より大きな値に設定すると、パフォーマンスに影響する可能性があります。

タイミングに影響を与えるオプション

いくつかのコマンドラインオプションは、特定のデータベースサーバイベントが発生するタイミングを制御します。

チェックポイント間隔

SAP Sybase IQ はチェックポイントを使用し、データベースのリカバリに必要な参照ポイントなどの情報を生成します。

-gc スイッチを使用して、データベースサーバがチェックポイントを行わずに実行する最長時間を分単位で設定します。

データベースサーバが複数のデータベースで実行されている場合、**-gc** スイッチによって上書きされないかぎり、最初に起動するデータベースによって指定されたチェックポイント時間が使用されます。値 0 を入力すると、デフォルト値の 20 が使用されます。

リカバリ時間

-gr パラメータを使用して、データベースサーバがシステム障害からリカバリするのにかかる最長時間を分単位で設定します。

データベースサーバが複数のデータベースで実行されている場合、**-gr** スイッチによって上書きされないかぎり、最初に起動するデータベースによって指定されたリカバリ時間が使用されます。

-gc iqsrv16 データベースサーバオプション
チェックポイント間の最大間隔を設定します。

構文

```
iqsrv16 -gc minutes ...
```

デフォルト

60 分

指定可能な値

- **minutes** – デフォルト値は、`checkpoint_time` データベースオプションの設定値 (デフォルトは 60 分) です。値 0 を入力すると、デフォルト値の 60 が使用されます。

適用対象

すべてのオペレーティングシステムとデータベースサーバ

備考

このオプションは、各データベースでチェックポイントを行わずに、データベースサーバを実行する期待最大時間を分数で設定するために使用します。

通常、チェックポイントは、指定する時間より頻繁に発生します。

ネットワークパフォーマンスオプション

-gb (Windows のデータベースプロセス優先度) と **-p** (最大パケットサイズ) オプションを使用して、ネットワークパフォーマンスをチューニングします。

-gb iqsrv16 データベースサーバオプション

サーバプロセスの優先度クラスを設定します。

Windows 構文

```
iqsrv16 -gb { idle | normal | high | maximum } ...
```

UNIX 構文

```
iqsrv16 -gb level ...
```

指定可能な値

- **UNIX** – UNIX の場合、*level* は -20 ~ 19 の整数です。UNIX のデフォルト値は、親プロセスの *nice* 値と同じです。*level* 値を下げると、より適切なスケジューリング優先度になります。**-gb** オプションには、*nice* 値の設定に関するすべての制限が適用されます。たとえば、ほとんどの UNIX プラットフォームの場合、プロセスの優先度レベルを下げる (たとえば、0 から -1 に変更する) ことができるのは root ユーザだけです。
- **Windows** – Windows の場合、一般的な設定として使用されるのは、*normal* または *high* です。値 *idle* は万全を期すために提供されています。値 *maximum* はコンピュータの実行を妨げる場合があります。

適用対象

Windows、UNIX

-p iqsrv16 データベースサーバオプション

通信パケットの最大サイズを設定します。

構文

```
iqsrv16 -p integer ...
```

デフォルト
7300 バイト

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

最小値は 500 バイトで、最大値は 65535 バイトです。

接続の通信バッファサイズを変更するには、CommBufferSize (CBSIZE) 接続パラメータを設定します。

権限を制御するコマンドラインオプション

データベースサーバの起動コマンド **start iq** には、一部のデータベースおよびサーバ機能権限レベルを設定するオプションが含まれます。

データベースの起動および停止に関連するスイッチ

-gd オプションを使用すると、実行中のサーバ上でデータベースを起動または停止できるユーザを、すでに接続しているデータベースに対して一定のレベルの権限が付与されたユーザに制限することができます。

- **DBA** – (デフォルト値) **SERVER OPERATOR** システム権限を持つユーザのみが追加のデータベースを起動できます。
- **ALL** – (**start iq** および **default.cfg** のデフォルト) すべてのユーザがデータベースの起動と停止を実行できます。この設定は、**DBA** が **START DATABASE** コマンドを発行する必要がないことを意味します。ただし、ユーザがデータベースを起動した後、そのデータベースにアクセスする権限がユーザに付与される必要があります。
- **NONE** – 実行中のサーバ上で **Interactive SQL** からデータベースを起動または停止できるユーザはいません。

注意：サーバを起動したときに **-gd ALL** が設定されていない場合、**SERVER OPERATOR** システム権限が付与されたユーザのみがそのサーバ上で追加のデータベースを起動できます。つまり、サーバと同時に起動されていないデータベース、またはサーバ起動後に **SERVER OPERATOR** システム権限を持つユーザによって起動されていないデータベースには、ユーザは接続できません。ただし、**SERVER OPERATOR** システム権限がないユーザもデータベースを停止することはできます。このため、運用データベースではこの設定を **DBA** に変更することをおすすめします。

データベースの作成と削除に関連するスイッチ

-gu オプションは、データベースを作成または削除できるユーザを、接続先のデータベースの一定のレベルの権限を持つユーザに制限します。

- **DBA** – **SERVER OPERATOR** システム権限を持つユーザのみがデータベースの作成および削除を実行できます。
- **ALL** (デフォルト) – すべてのユーザがデータベースの作成および削除を実行できます。
- **NONE** – ユーザはデータベースの作成や削除を実行できません。
- **UTILITY_DB** – `utility_db` データベースに接続できるユーザのみがデータベースの作成および削除を実行できます。

サーバ停止に関連するスイッチ

-gk オプションは、**dbstop** ユーティリティまたは **STOP ENGINE** コマンドを使用してサーバを停止できるユーザを制限します。

- **DBA** (デフォルト) – **SERVER OPERATOR** システム権限を持つユーザのみがサーバを停止できます。
- **ALL** – すべてのユーザがサーバを停止できます。
- **NONE** – ユーザは、**dbstop** ユーティリティまたは **STOP ENGINE** コマンドを使用してサーバを停止できません。

データベースのロードとアンロードに関連するスイッチ

-gl オプションは、**LOAD TABLE** を使用してデータをロードできるユーザを、データベースに対する一定の権限を持つユーザに制限します。

- **DBA** – **LOAD ANY TABLE**、**ALTER ANY TABLE**、または **ALTER ANY OBJECT** システム権限を持つすべてのユーザがデータをロードできます。
- **ALL** (`start_iq` と `default.cfg` のデフォルト) – すべてのユーザがデータをロードできます。
- **NONE** – データのロードはできません。

-gd iqsrv16 データベースサーバオプション

実行中のデータベースサーバ上でデータベースを起動または停止するために必要な権限を設定します。

構文

```
iqsrv16 -gd { DBA | all | none } ...
```

指定可能な値

- **DBA** – **SERVER OPERATOR** システム権限を持つユーザのみがデータベースを起動または停止できます。
- **all** – すべてのユーザがデータベースを起動または停止できます。リモートクライアントからアクセスできるネットワークサーバへの設定はおすすめしません。

- **none** – データベースサーバ自体が起動しているか停止している場合を除き、データベースの開始と停止は許可されません。

デフォルト

デフォルトの設定は、ネットワークデータベースサーバの DBA です。大文字と小文字の両方の構文を使用できます。

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

このオプションは、ユーザが新しいデータベースファイルをデータベースサーバにロードするために、または実行中のデータベースサーバでデータベースを停止するために必要な権限のレベルを指定します。

オプションを DBA に設定した場合、クライアントアプリケーションは、データベースを開始または停止させるために同一のサーバ上で動作している別のデータベースへの既存の接続を使用する必要があります。DatabaseFile 接続パラメータを使用して、まだ実行中でないデータベースを起動することはできません。

次のように StartDBPermission サーバプロパティを使用して -gd オプションの設定を取得できます。

```
SELECT PROPERTY ( 'StartDBPermission' );
```

データベースサーバを停止するための権限は、-gk オプションで指定します。

例

ネットワークデータベースサーバで -gd オプションを使用する手順は、次のとおりです。

1. ネットワークデータベースサーバを起動します。

```
iqsrv16 -su mypwd -gd DBA -n my_server
```

2. Interactive SQL からユーティリティデータベースに接続します。

```
dbisql -c "UID=DBA;PWD=mypwd;DBN=utility_db"
```

3. データベースを起動します。

```
START DATABASE 'demo.db';
```

4. データベースに接続します。

```
CONNECT USING 'DBN=demo;UID=DBA;PWD=sql';
```

-gk iqsrv16 データベースサーバオプション

データベースサーバの停止に必要な権限を設定します。

構文

```
iqsrv16 -gk { DBA | all | none } ...
```

指定可能な値

- **DBA** – **SERVER OPERATOR** システム権限を持つユーザのみがデータベースサーバを停止できます。これはネットワークサーバのデフォルトです。
- **all** – データベースサーバを停止するのに権限が必要ありません。
- **none** – データベースサーバを停止できません。

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

-gd データベースサーバオプションは、**dbstop** ユーティリティに適用されるほか、次の文にも適用されます。

- **ALTER DATABASE *dbname* FORCE START** 文
- **STOP DATABASE** 文

-gl iqsrv16 サーバオプション

LOAD TABLE を使用してデータをロードするためのパーミッションを設定します。

構文

-gl level

備考

LOAD TABLE 文はデータベースサーバマシンからファイルを読み取ります。このような文を使用したファイルシステムアクセスを制御するために、**-gl** コマンドラインスイッチを使用して、このような文の使用に必要なデータベースパーミッションレベルを調整できます。*level* は次のいずれかです。

- **DBA** - **LOAD ANY TABLE** システム権限、**ALTER ANY TABLE** システム権限、または **ALTER ANY OBJECT** システム権限を持つユーザのみがデータをロードできます。
- **ALL** - すべてのユーザがデータをロードできます。
- **NONE** - データのロードはできません。

オプションには大文字の構文も小文字の構文も使用できます。

デフォルト設定は、**start_iq** を使用して起動されたサーバの場合は **all** で、その他の場合は **dba** です。以前のバージョンとの一貫性を維持するために、すべてのシステムで **all** 値を使用してください。iqdemo.cfg および default.cfg 設定ファイルでは、**all** 設定が使用されています。

-gu iqsrv16 データベースサーバオプション

データベースファイル管理文(データベースの作成や削除などの文)の実行に必要な権限を設定します。

構文

```
iqsrv16 -gu { all | none | DBA | utility_db } ...
```

指定可能な値

-gu オプション	効果	適用対象
all	このオプションは推奨されなくなりました。すべてのユーザがファイル管理文を実行できます。	ユーティリティデータベースを含むすべてのデータベース
none	ファイル管理文の実行は許可されません。	ユーティリティデータベースを含むすべてのデータベース
DBA	SERVER OPERATOR システム権限を持つユーザのみがファイル管理文を実行できます。	ユーティリティデータベースを含むすべてのデータベース
utility_db	ユーティリティデータベースに接続できるユーザのみがファイル管理文を実行できます。	ユーティリティデータベースのみ

デフォルト

DBA

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

次のデータベースファイル管理文を実行できるユーザが制限を受けます。

- ALTER DATABASE dbfile ALTER TRANSACTION LOG
- CREATE DATABASE 文
- CREATE DECRYPTED DATABASE 文
- CREATE DECRYPTED FILE 文
- CREATE ENCRYPTED DATABASE 文
- CREATE ENCRYPTED FILE 文
- DROP DATABASE 文

- **RESTORE DATABASE 文**

utility_db を指定した場合、これらの文はユーティリティデータベースからのみ実行できます。DBA を指定した場合、これらの文は SERVER OPERATOR システム権限を持つユーザのみ実行できます。none を指定した場合は、どのユーザもこれらの文を実行できません。

例

ファイル管理文の使用を防ぐため、-gu オプションの none 権限レベルを使用してデータベースサーバを起動します。次のコマンドは、データベースサーバを起動し、TestSrv という名前を付けます。このコマンドによって mytestdb.db データベースがロードされますが、どのユーザも、そのサーバを使用してデータベースを作成または削除したり、他のファイル管理文を実行したりすることはできません。これは、ユーザのリソース作成権の有無や、ユーティリティデータベースをロードして接続できるかどうかには関係ありません。

```
iqsrv16 -n TestSrv -gu none c:¥mytestdb.db
```

ユーティリティデータベースのパスワードを知っているユーザだけにファイル管理文の実行を許可するには、次のコマンドを実行してサーバを起動します。

```
iqsrv16 -n TestSrv -su secret -gu utility_db
```

次のコマンドは、Interactive SQL をクライアントアプリケーションとして起動し、TestSrv という名前のサーバに接続し、ユーティリティデータベースをロードして、ユーザを接続させます。

```
dbisql -c
"UID=DBA;PWD=secret;DBN=utility_db;Host=host1;Server=TestSrv"
```

上記のコマンドが正常に実行されると、ユーザがユーティリティデータベースに接続し、ファイル管理文を実行できます。

最大カタログページサイズ

データベースサーバキャッシュは、固定サイズのメモリ領域であるページに調整されます。サーバは、停止するまでカタログストアに対して1つのキャッシュを使用するため、カタログページのサイズはすべて同じにする必要があります。

カタログファイルも、4096、8192、16384、または 32768 バイトのページに調整されます。どのデータベースページも、キャッシュページに収まらなければなりません。

-gp オプションを使用して、カタログページサイズを明示的に設定します。-gp を最大サイズ 32768 に設定することによって、SAP Sybase IQ でサポートされるテーブルごとのカラム数を最大にできます。

デフォルトでは、サーバページサイズは、コマンドラインで指定されたデータベースの最大ページサイズと同じ大きさです。-gp オプションは、このデフォルトを無効にします。サーバがいったん起動すると、サーバより大きいカタログペー

ジサイズを持つデータベースをロードできません。-gp を指定しないかぎり、コマンドラインで起動したデータベースより大きいカタログページサイズのデータベースファイルはロードできません。

より大きいページサイズを使用する場合は、必ずキャッシュサイズを増やしてください。キャッシュサイズを変更しないと、大きなページの一部だけが保管され、領域を柔軟に調整できなくなります。

注意： -gp オプションとそこで指定したページサイズはカタログストアにのみ適用されます。IQ ストアのページサイズは、**CREATE DATABASE** コマンドの **IQ PAGE SIZE** パラメータで設定します。

-gp iqsrv16 データベースサーバオプション

許可される最大データベースページサイズを設定します。

構文

```
iqsrv16 -gp { 2048 | 4096 | 8192 | 16384 | 32768 } ...
```

デフォルト

4096 (データベースをロードせずにデータベースサーバを起動した場合)

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

サーバのページサイズよりも大きいページサイズのデータベースファイルはロードできません。このオプションは、サーバのページサイズをバイト数で明示的に設定します。

デフォルトでは、サーバページサイズは、コマンドラインで指定されたデータベースの最大ページサイズと同じ大きさです。

いずれのプラットフォームでも、このオプションを指定せずにデータベースがロードされていない状態でサーバを起動した場合、デフォルト値は 4096 になります。

クライアント/サーバ環境のオプション

-x (通信プロトコル)、-tl (ネットワーク接続タイムアウト)、および -ti (クライアント接続タイムアウト) オプションを使用して、クライアント/サーバ環境を設定できます。

通信プロトコル

クライアントアプリケーションとデータベースサーバ間の通信には、通信プロトコルが必要です。SAP Sybase IQ では、ネットワーク通信用と同一マシン通信用の通信プロトコルセットをサポートしています。

データベースサーバは、次のプロトコルをサポートします。

- 「共有メモリ」は、同一マシン通信で使用され、デフォルトでロードされます。
- **TCP/IP**は、すべてのプラットフォームでサポートされます。
- **名前付きパイプ**は、Windows 2000/2003/XP のみでサポートされます。名前付きパイプは、ODBC または Embedded SQL を使用する Windows クライアントアプリケーション間の同一マシン通信用に用意されていますが、この目的で使用することは一般にはおすすしめしません。名前付きパイプは、ネットワーク通信では使用しません。

サーバプロトコルスイッチ

デフォルトでは、データベースサーバは、使用可能なプロトコルをすべて有効にします。**-x** コマンドラインスイッチによって、データベースサーバで使用できるプロトコルを制限できます。クライアント側は、**CommLinks** 接続パラメータを使用してこれらのオプションの多くを制御できます。

TCP/IP プロトコルを使用してサーバを起動するには、次のように指定します。

```
start_iq -x "tcpip" -n myserver
```

この例で引用符は必須ではありませんが、**-x** の引数にスペースが含まれている場合は必要となります。このスイッチを使用せずに TCP/IP を使用したり、またはポート番号を指定しない場合は、デフォルトのポート 2638 が使用されます。

パラメータを追加して、プロトコルごとにサーバの動作をチューニングできます。たとえば、次のコマンドラインは、2つのネットワークカード(そのうち1つは指定したポート番号を持つ)を使用するようにサーバに指示します。ここでは複数行に分けてありますが、このコマンドはすべて1行に入力してください。

```
start_iq -x "tcpip(MyIP=192.75.209.12:2367,192.75.209.32)" path
¥iqdemo.db
```

-x iqsrv16 サーバオプション

サーバ側のネットワーク通信プロトコルを指定します。

構文

-x list

備考

list は、**tcpip** 設定または **namedpipes** 設定のカンマ区切りのリストです。次はその例です。

```
-x tcpip,ipx
```

この場合は、TCP/IP および IPX 通信のみを使用できます。

デフォルトでは、使用しているオペレーティングシステム上で動作するデータベースサーバによってサポートされている、すべての設定が試行されます。

一部のプロトコルについては、次の形式の追加のパラメータを指定できます。

```
-x tcpip (PARM1=value1;PARM2=value2;...)
```

UNIX では、複数のパラメータを指定する場合に二重引用符が必要です。

```
-x "tcpip (PARM1=value1;PARM2=value2;...)"
```

非アクティブ接続を制限するスイッチ

サーバオプション `-tl` および `-ti` により、いつ SAP Sybase IQ のユーザ接続を閉じるかを指定します。

デフォルトネットワークタイムアウトスイッチ

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 通信プロトコルを介して、定期的に活性パケットが送信されます。活性パケットを検出しないで、指定した活性タイムアウト時間 (デフォルトは 2 秒) にわたってサーバが実行される場合、通信は切断されます。このとき、サーバによって、対象のクライアントと関連するすべての接続が切断されます。警告はありません。開いているトランザクション内のすべてのアクティビティがロールバックされます。

サーバで `-tl` スイッチを指定すると、接続時に `-tl` スイッチを指定していないすべてのクライアントに対して活性タイムアウト (秒単位) を設定できます。活性パケットは、活性タイムアウト値の 4 分の 1 の間隔で送信されます。

サーバレベルでこのスイッチに大きい値を設定できます。多くのユーザ、特に以前のバージョンの SAP Sybase IQ を使用していたユーザは、2 分間操作をしないと切断されるとは予期していません。

活性タイムアウトを 300 に設定し、`-ti` の推奨値を設定してください。このスイッチを次のように設定します。

```
-tl 300
```

この値が適切でない場合は、`-tl 1200` を指定して、活性タイムアウトを 20 分間に設定してください。

注意: クライアントと同じマシンでサーバを実行している場合は、活性タイムアウトは発生しません。

デフォルトクライアントタイムアウトスイッチ

-ti スイッチで指定される時間 (分) 要求を送信しなかったクライアント接続は、SAP Sybase IQ によって切断されます。非アクティブな接続を切断することによって、接続が保持しているロックを解放します。**start_iq** のデフォルト値は 4400 (約 72 時間) です。この設定では、たとえば、週末の開始時に長期の実行を開始しても、中間結果がロールバックされません。

-ti iqsrv16 データベースサーバオプション

非アクティブ接続を切断します。

構文

```
iqsrv16 -ti minutes ...
```

デフォルト

240 (4 時間)

適用対象

すべてのオペレーティングシステムとデータベースサーバ。

備考

minutes で指定された分数の間、要求を送信しなかった接続を切断します。最大値は 32767 です。データベーストランザクション処理中のクライアントコンピュータは、トランザクションが終了するか、接続が切断されるまでロックされます。-ti オプションを指定すると、非アクティブ接続が切断され、ロックが解除されません。

ほとんどの接続はネットワークリンク (TCP) 経由で行われているため、-ti オプションは iqsrv16 と一緒に使用すると非常に便利です。

値を 0 に設定すると、非アクティブ接続は検査されず、接続が切断されません。Idle 接続パラメータが使用されていない場合、TCP/IP 接続のアイドルタイムアウト値は -ti データベースサーバオプションによって制御されます。両方の -ti データベースサーバオプションと Idle 接続パラメータが指定された場合、アイドルタイムアウト値は接続パラメータによって制御されます。

-tl iqsrv16 データベースサーバオプション

活性パケットを送信する期間を設定します。

構文

```
iqsrv16 -tl seconds ...
```

適用対象

TCP/IP を使用するすべてのデータベースサーバ

備考

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 通信プロトコルを介して、定期的に活性パケットが送信されます。接続で活性パケットを検出することなく、指定した LivenessTimeout 時間 (デフォルトは 2 分) にわたってサーバが実行されていると、通信は切断され、サーバはそのクライアントに関連付けられている接続を削除します。非スレッドの UNIX クライアントと TDS 接続では、活性パケットによる確認は行われません。

サーバで `-tl` オプションを指定すると、活性期間が指定されていないすべてのクライアントに対して LivenessTimeout 値を設定できます。

LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で接続がパケットを送信しない場合に、活性パケットが送信されます。

接続数が 200 を超えると、サーバは指定されている LivenessTimeout 値に基づいて、それより大きい LivenessTimeout 値を自動的に計算するため、多数の接続を効率的に処理できます。活性パケットは、各アイドル接続において、LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で送信されます。大量の活性パケットが同時に送信されることはありません。送信に時間がかかる活性パケットは、LivenessTimeout 値の 3 分の 2 の期間が経過した後で送信できます。活性パケットの送信に時間がかかる場合、データベースサーバメッセージログに警告が表示されます。この警告が発生したら、LivenessTimeout 値の増加を検討してください。

これは一般的にはおすすめしませんが、次のように指定して活性タイムアウトを無効にできます。

```
iqsrv16 -tl 0 -n my_server
```

LivenessTimeout オプションを無効にせずに、次のように値を 1 時間に増やすことを検討してください。

```
iqsrv16 -tl 3600 -n my_server
```

強制リカバリモードオプション

障害の後は、通常の起動オプションでサーバを再起動します。リカバリを強制的に行うか、リークした記憶領域をリカバリするために起動オプションを指定することが必要な場合もごくまれにあります。

-iqfrec iqsrv16 サーバオプション

強制リカバリモードでデータベースを開きます。

構文

-iqfrec dbname

Interactive SQL からのサーバの起動

実行中のデータベースサーバにすでに接続している場合は、Interactive SQL から新しいサーバを起動できます。

1 行に入力された次の Interactive SQL コマンドは、データベースサーバを起動し、そのサーバに jill_newserv という名前を付け、ネットワーク接続、接続数、カタログページサイズを指定します。

```
START ENGINE AS jill_newserv
STARTLINE 'start_iq -x tcpip(port=5678) -gm 10 -gp 4096'
```

START ENGINE コマンドを使用して、Interactive SQL から指定したサーバを起動します。

通常は、この方法を使用しないことをおすすめします。使用する場合は、正しいシステムでサーバを起動し、コマンドに適切なサーバパラメータを指定し、サーバを起動するシステムで環境変数を正しく設定してください。

共有メモリの競合

実際の運用環境では、通常、同じシステム上で複数のサーバを実行しません。SAP Sybase は、同じシステム上で複数のサーバを実行しないことを、強くおすすめします。ただし、開発環境では複数のサーバを実行する場合があります。

同じ UNIX マシンで複数のサーバまたはクライアントを実行して、共有メモリを有効にする場合、間違ったサーバに接続しないように注意してください。

サーバを起動しようとする、次のメッセージが表示される場合があります。

```
DBSPAWN ERROR -96 -- database engine already running
```

このエラーは、起動プロセスが、以前に起動されたサーバの共有メモリセグメントを見つけ、共有メモリセグメントを作成できないことを意味します。このエラーは、SAP Sybase IQ または SQL Anywhere サーバが実行されている場合に発生する可能性があります (Interactive SQL は、後から起動されたサーバに接続するときでも、共有メモリポートを参照できる場合は以前のサーバに接続します)。システムごとに SAP Sybase IQ または SQL Anywhere のいずれか 1 つのサーバのみを実行することで、このエラーを回避できます。

共有メモリ競合を回避する方法

共有メモリ使用時に競合を回避する方法はいくつかあります。

- 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。各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、両方のシステムに明示的に IQTMP16 環境変数を設定します。
- 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
```

1つのシステムで複数のサーバを実行している場合は、サーバごとに次のユニークな値を設定します。

- 名前。起動時に **-n** パラメータで指定します。
- ポート番号。 **-x** パラメータで指定します。

サーバアクティビティログ

プラットフォームに応じたコマンドを使用して、SAP Sybase IQ がサーバのアクティビティをログファイルに取り込むように設定できます。

サーバの起動メッセージ

SAP Sybase IQ サーバを起動すると、一連のメッセージがサーバログウィンドウに表示されます。実際に表示されるメッセージは、使用しているプラットフォームとライセンスが供与されているオプションによって異なります。AIX システムの例を次に示します。

```
Starting server myserver_iqdemo on myserver at port 3658 (09/06  
17:25:23)
```

```
Run Directory      : /myserver/users/sybase/iqdemo_160_sep05  
Server Executable  : /myserver/users/sybase/160_sep05/IQ-16_0/  
bin64/iqsrv16  
Server Output Log  : /myserver/users/sybase/160_sep05/IQ-16_0/  
logfiles/iqdemo_3658.0001.srvlog  
Server Version     : 16.0.0.6556/Mainline  
Open Client Version : N/A  
User Parameters    : '@iqdemo.cfg' 'iqdemo.db'  
Default Parameters : -ti 4400 -gn 25
```

```
I. 09/06 17:25:26. Sybase IQ  
I. 09/06 17:25:26. Version 16.0  
I. 09/06 17:25:26. (64bit mode)
```

```

I. 09/06 17:25:26. Copyright 1992-2012 by Sybase, Inc. All rights reserved
I. 09/06 17:25:26. Copyright (c) 2001-2012, Sybase, Inc.
I. 09/06 17:25:26. Portions copyright (c) 1988-2011, iAnywhere Solutions, Inc. All rights reserved.
I. 09/06 17:25:26. Use of this software is governed by the Sybase License Agreement.
I. 09/06 17:25:26. Refer to http://www.sybase.com/softwarelicenses.
I. 09/06 17:25:26.
I. 09/06 17:25:26. Processors detected: 4 (containing 16 logical processors)
I. 09/06 17:25:26. Maximum number of physical processors the server will use: 4
I. 09/06 17:25:26. Running SunOS 5.10 Generic_144489-12 on X86_64
I. 09/06 17:25:26. Server built for X86_64 processor architecture
I. 09/06 17:25:26. 49152K of memory used for caching
I. 09/06 17:25:26. Minimum cache size: 49152K, maximum cache size: 262144K
I. 09/06 17:25:26. Using a maximum page size of 4096 bytes
I. 09/06 17:25:27. Starting database "iqdemo" (/myserver/users/sybase/iqdemo_160_sep05/iqdemo.db) at Tue Sep 06 2012 17:25
I. 09/06 17:25:27. Transaction log: iqdemo.log
I. 09/06 17:25:27. Starting checkpoint of "iqdemo" (iqdemo.db) at Tue Sep 06 2012 17:25
I. 09/06 17:25:27. Finished checkpoint of "iqdemo" (iqdemo.db) at Tue Sep 06 2012 17:25
I. 09/06 17:25:28. Database "iqdemo" (iqdemo.db) started at Tue Sep 06 2012 17:25
I. 09/06 17:25:28. IQ Server iqdemo_3658.
I. 09/06 17:25:28. Database server started at Tue Sep 06 2012 17:25
I. 09/06 17:25:28. Trying to start SharedMemory link ...
I. 09/06 17:25:28. SharedMemory link started successfully
I. 09/06 17:25:28. Trying to start TCPIP link ...
I. 09/06 17:25:28. Starting on port 3658
I. 09/06 17:25:33. TCPIP link started successfully
I. 09/06 17:25:33. Now accepting requests
I. 09/06 17:25:45. Database server shutdown due to HUP signal
I. 09/06 17:25:45. TCPIP listener on IP address (::):3658 is exiting
I. 09/06 17:25:45. TCPIP listener on IP address 0.0.0.0:3658 is exiting
I. 09/06 17:25:45. Starting checkpoint of "iqdemo" (iqdemo.db) at Tue Sep 06 2012 17:25
I. 09/06 17:25:45. Finished checkpoint of "iqdemo" (iqdemo.db) at Tue Sep 06 2012 17:25
I. 09/06 17:25:46. Database server stopped at Tue Sep 06 2012 17:25

```

start_iq ログファイル

start_iq ユーティリティでサーバを起動すると、ASCII テキストファイルにサーバアクティビティのログが記録されます。このファイルは、`$IQLOGDIR16` で定義されたディレクトリに置かれており、サーバからの標準的な出力とサーバステータスを含みます。

ログファイル名のフォーマットは次のとおりです。

SAP Sybase IQ データベースサーバの実行

```
your_server_name.nnnn.srvlog
```

サーバを起動するたびに、数値が1ずつ増加します。たとえば、ディレクトリ内は次のようになります。

```
demo.0001.srvlog demo.0002.srvlog
testdemo.0001.srvlog
```

最新のセッションの詳細を参照するには、対象のサーバについて、数値が最も大きいログを選択します。ログの内容を表示するには、**tail -f** コマンドを発行します。次に例を示します。

```
% tail -f demo.0002.srvlog
```

\$IQLOGDIR16 ディレクトリを定義しないと、UNIX の場合、ログは \$IQDIR16/logfiles/ ディレクトリに書き込まれ、Windows の場合は SAP Sybase IQ のインストールで定義された \$IQLOGDIR16 ディレクトリに書き込まれます。

start_iq を実行する場合、**-z** オプションを指定して、追加の接続情報を含むようにログファイルを拡張できます。これは、新規ユーザや接続に関するトラブルシューティングを行うユーザに役立ちます。

UNIX システムでは、サーバが実行されているかどうかを確認する方法が2つあります。

- サーバが起動されたマシンにログインし、次のコマンドを発行します。

```
ps -eaf | grep iqsrv
```

この出力は、UNIX プラットフォーム間では若干異なります。IBM AIX の場合、カラムは次のようになります。

UID	PID	PPID	C	STIME	TTY	TIME	CMD
-----	-----	------	---	-------	-----	------	-----

次に例を示します。

```
jones 422034 1 0 17:47:36 - 0:04
/ibm64srv/users/sybase/iq160/IQ-16_0/bin64/
iqsrv16
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o
/ibm64srv/users/sybase/iq160/IQ-
16_0/logfiles/ibm64srv_iqdemo.0003.srvlog -hn 7
```

- 次の項で説明する **stop_iq** ユーティリティを使用して、実行中のすべての SAP Sybase IQ プロセスを表示します。

Windows システムでは、システムトレイに1つ以上の SAP Sybase IQ アイコンがあるかどうかをチェックします。カーソルを各アイコン上に置くと、サーバ名が表示されます。

サーバログファイルの命名

サーバログファイルには、デフォルトで `server.nnnn.srvlog` の名前が割り当てられます。

サーバログファイルの名前を変更するには、`start_iq` コマンドの `-o` スイッチを使用します。

たとえば、サーバを起動したディレクトリの `results` という名前のファイルに出力を保存するには、次のようにサーバを起動します。

```
start_iq -n imyserver -o results
```

`-o` スイッチを使用して、ログファイルのフルパスを指定することもできます。

UNIX ログファイル

UNIX プラットフォームでは、`stdout` や `stderr` の出力をはじめとするオペレーティングシステムの出力は、別のログファイルに取得されます。

ファイル名のフォーマットは次のとおりです。

```
your_server_name.####.stderr
```

予期しない例外が発生した場合、SAP Sybase IQ はこれをスタックトレースファイルに書き込みます。UNIX システムでは、スタックトレース情報が格納されるファイル名のフォーマットは、次のとおりです。

```
stktrc-YYYYMMDD-HHNNSS_#.iq
```

データベースサーバの停止

システム管理者は、SAP Sybase IQ データベースサーバを停止する方法だけでなく、停止する必要がある状況、停止できるユーザの制御、オペレーティングシステム停止時のサーバの停止方法を理解しておく必要があります。

サーバを停止または再起動する状況

いくつかの状況では、サーバを停止して、再起動する必要があります。

次に例を示します。

- SAP Sybase IQ の新しいバージョンをインストールするとき。
- サーバコマンドラインオプションをリセットするとき。
- サーバ全体を対象とする少数のデータベースオプションを有効にするとき。
- オペレーティングシステムセッションを終了する前。

参照：

- データベースオプションの範囲と継続期間 (439 ページ)

サーバの停止

サーバを停止するためのさまざまな方法

サーバの停止に必要なデフォルトのパーミッションレベルは **SERVER OPERATOR** システム権限ですが、**-gk** 起動オプションを使用して、デフォルトを **ALL** または **NONE** に変更できます。パーミッションレベルを **ALL** に設定すると、すべてのユーザがサーバを停止できます。**NONE** に設定すると、誰もサーバを停止できなくなります。運用環境では、サーバオペレータのみがデータベースサーバを停止できるようにしてください。

クライアントに接続中のサーバを停止しないでください。これを行うと、コミットされていないトランザクションが失われます。停止前にすべてのクライアントを閉じるか、すべてのクライアントと切断してください。

UNIX 対応サーバ停止コマンド

複数のコマンドによってデータベースサーバを停止できます。

コマンド	説明
stop_iq	ユーザ、接続、ロード処理のステータスを考慮しないで、サーバを完全に停止する。 構文： <code>stop_iq [options]</code>
dbstop	サーバまたはデータベースを停止する。アクティブな接続がある場合でも、追加オプションによってサーバを停止できる。 構文： <code>dbstop [options] server-name</code> dbstop を使用してサーバを停止する場合は、サーバの起動時に指定したすべての接続パラメータに加え、 server-name も指定する必要がある。
iqsrv16	入力を別のデバイスにリダイレクトしていない場合で、サーバを起動したウィンドウで入力するとサーバを停止する。 q
STOP ENGINE	STOP ENGINE は、データベースサーバを停止する SQL 文である。 構文： <code>STOP ENGINE engine-name [UNCONDITIONALLY]</code> UNCONDITIONALLY 引数は、サーバへのクライアント接続がある場合でもデータベースサーバを停止する。

Windows でのサーバの停止方法

複数の方法でサーバをシャットダウンまたは停止できます。

サーバの停止方法	操作
コマンドライン	stop_iq コマンドまたは dbstop コマンドを実行する。
デスクトップ	システムトレイの[サーバアイコン]を右クリックして、 Shutdown server name を選択する。
サービスマネージャ	サーバを Windows サービスとして起動した場合は、次の手順に従う。 <ol style="list-style-type: none"> [コントロールパネル]で、[管理ツール]>[サービス]を選択する。 SAP Sybase IQ サービスを選択して、Stop the service をクリックする。
Interactive SQL	stop engine コマンドを実行して、指定したデータベースサーバを停止する。

UNIX での cron または at ジョブでのサーバの停止

stop_iq を **cron** または **at** ジョブで使用するには、**stop_iq** 実行プログラムのフルパス名と適切な **-stop** オプションを指定します。

cron または **at** ジョブを開始するユーザ ID によって起動されているサーバが 1 台しかない場合、**-stop one** を設定すると、1 台のサーバが停止します。これによって、複数のサーバが実行されている場合、対象と異なるサーバを誤って停止することがなくなります。

```
stop_iq -stop one
```

-stop all を設定すると、**cron** または **at** ジョブを起動するユーザ ID によって起動されたすべてのサーバが停止されます。

```
stop_iq -stop all
```

例 — stop_iq を使用したサーバの停止

次の例では、UNIX オペレーティングシステムのコマンドラインで **stop_iq** ユーティリティを使用して SAP Sybase IQ サーバを停止し、そのサーバへのすべてのユーザ接続を閉じます。

stop_iq コマンドを発行すると、SAP Sybase IQ によって、自分以外のユーザが所有しているすべてのサーバがリストされ、次に、自分が所有しているサーバがリストされます。サーバを停止するかどうかを尋ねるメッセージが表示されます。次に例を示します。

SAP Sybase IQ データベースサーバの実行

```
% stop_iq

Checking system for IQ 16 Servers ...
The following 2 server(s) are owned by other users.
##      Owner      PID      Started      CPU_Time      Additional Information
--      -
          handari  19895   15:43:44   183:38
start_iq @iqdemo.cfg iqdemo.db -gn 105 -o /server1/users/surya/
IQ-16_0/logfiles/surya_ibm2.001.srvlog -hn 8          pamel  409802
18:05:02      0:05 SVR:ibm1_iqdemo2 DB:iqdemo
PORT:2678/ibm1/users/sybase/iql60/IQ-16_0/bin64/iqsrv16 @iqdemo.cfg
iqdemo.db -ti 4400 -gn 25 -o /ibm1/users/sybase/iql60/IQ
16_0/logfiles/ibm64qa_iq
The following 1 server(s) are owned by 'kermit'
##      Owner      PID      Started      CPU_Time      Additional Information
--      -
1:      kermit    422034  15:11:37      0:07      SVR:myserver_iqdemo
DB:iqdemo PORT:2638 /myserver/users/sybase/iql60/IQ-16_0/bin64/
iqsrv16
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o /myserver/users/sybase/
iql60/IQ-
16_0/logfiles/myserver iq
start_iq -c 32m -gd all -gm 10 -gn 25 -gp 4096 -ti 4400 -tl 300
@iqdemo.cfg

--

Please note that 'stop_iq' will shut down a server completely
without regard for users connections or load processes status.
For more control, use the 'dbstop' utility, which has options
that control stopping servers based on active connections.

Do you want to stop the server displayed above <Y/N>?
```

サーバを停止するには、 Y(yes)と入力します。次のようなメッセージが表示されます。

```
Shutting down server (422034) ...
Checkpointing server (422034) ...
Server shutdown.
```

サーバの実行を続けるには、 N(no)と入力します。Nを入力すると、システムプロンプトに戻り、サーバは停止しません。

実行中のサーバの中に、自分自身のユーザ ID を使用して起動したものがない場合は、SAP Sybase IQ は、他のユーザによって実行されているサーバの情報を表示した後に、次のようなメッセージを表示します。

```
There are no servers owned by 'kermit'
```

例 — Interactive SQL からのサーバの停止

次の例では、Interactive SQL からサーバを停止します。

```
STOP ENGINE Ottawa UNCONDITIONALLY
```

オプションの **UNCONDITIONALLY** キーワードを指定すると、データベースとの接続がある場合でも、そのデータベースサーバを停止します。

注意：サーバで実行されているデータベース (utility_db データベースを含む) の 1 つに DBA として接続している場合、または -gk ALL オプションを指定してサーバを起動した場合は、Interactive SQL からそのサーバを停止できます。

サーバを停止するために必要なパーミッション

サーバの起動時に **-gk** オプションを使用すると、ユーザが **DBSTOP** または **STOP ENGINE** を使用してサーバを停止するために必要なパーミッションレベルを設定できます。

必要なパーミッションレベルのデフォルトは、SERVER OPERATOR システム権限を必要とする **DBA** ですが、この値を **ALL** または **NONE** に設定することもできます。**NONE** に設定すると、SERVER OPERATOR システム権限を持つユーザも **STOP ENGINE** を実行できません。運用環境では、DBA のみがデータベースサーバを停止できるようにしてください。

UNIX コマンドラインで **stop_iq** を実行するか、Windows プラットフォームで [シャットダウン] を選択すると、サーバが起動されたマシン上のサーバとデータベースも停止できます。

オペレーティングシステムセッションの停止

データベースサーバを明示的に停止してから、オペレーティングシステムのセッションを閉じてください。

データベースサーバが実行中のオペレーティングシステムセッションを閉じたり、オペレーティングシステムコマンド (**stop_iq** 以外) を使用してデータベースサーバを停止すると、サーバは正しく停止しません。次にデータベースをロードするときに、自動的にリカバリが行われます。

注意：サーバが正しく停止されないコマンド例として、Windows タスク マネージャの [プロセス] ウィンドウでプロセスを停止する操作が挙げられます。

データベースの開始と停止

サーバを起動するとき、またはサーバの起動後にデータベースを起動できます。

特に運用環境では、1 つのサーバで 1 つのデータベースのみ実行します。

実行中のサーバ上でデータベースを起動する方法は、いくつかあります。

- Interactive SQL または Embedded SQL からデータベースを起動するには、**START DATABASE** 文を使用します。
- Interactive SQL からデータベースを起動し、接続するには、データベースファイルを指定するデータソースを使用します。
- システムコマンドプロンプトから Interactive SQL を起動するときデータベースを起動し、接続するには、“**DBF=db-file**” 接続パラメータを含めます。
- サーバに接続中の組み込みデータベースを起動するには、DBF パラメータを使用してデータベースファイルを新しい接続として指定して、データベースに接続します。データベースファイルは現在のサーバにロードされます。

データベースの起動に関するガイドライン

データベースを起動するときは、次の問題を考慮してください。

ファイルアクセス

データベースが起動するには、IQ_SYSTEM_MAIN のすべてのファイル、IQ_SYSTEM_TEMP のすべてのファイル、およびカタログファイル SYSTEM が使用可能である必要があります。データベースは、完全にオープンにできない DB 領域をスキップして起動できます。IQ_SYSTEM_MAIN 以外の IQ メインストア DB 領域のいずれかの書き込み可能なファイル、または SYSTEM 以外のいずれかのカタログ DB 領域ファイルを、サーバの起動時にオープンできない場合、SAP Sybase IQ は、エラーをログに出力し、その DB 領域を動的にオフラインとしてマークします(カタログ内でオフラインとマークするのではなく、メモリ内でオフラインとマークします)。オープンできない IQ_SYSTEM_TEMP ファイルがある場合、**-iqnotemp** 起動パラメータを指定しないかぎり、データベースは起動しません。

SAP Sybase IQ は、各 DB 領域ファイルヘッダ内の `commit_id` と、システムテーブル `ISYSDBFILE` および `ISYSIQDBSPACE` 内の値の一貫性を確認し、一致しないファイルまたは DB 領域がある場合、これらを前述のようにオフラインとマークします。

起動時にオフラインとマークされた DB 領域は、問題が修正され、DB 領域をオープンできるという前提で、**ALTER DBSPACE ONLINE** 文を通じてオンラインにできます。パスの問題を修正するには、**ALTER DBSPACE *dbspace name* ALTER FILE *logical filename* RENAME PATH *new pathname*** を使用して DB 領域ファイルのパスを修正できます。

オフラインの DB 領域に格納されているテーブルオブジェクトは使用できません。オフラインの DB 領域内のテーブルオブジェクトに対する DDL または DML 要求 (**ALTER DBSPACE ONLINE** 以外) は、エラーを生成します。DB 領域をオフラインにした後でも、バッファキャッシュ内にデータページが存在する可能性があることに注意してください。テーブルのスペースが非常に小さい場合は、DB 領域がオフ

ラインの場合でも、テーブル全体がバッファキャッシュ内のメモリに存在し、一時的に使用可能な場合があります。

ページサイズ制限

サーバは、固定サイズのページを使用して、メモリにデータベース情報を保持します。サーバがいったん起動すると、サーバより大きいカタログページサイズまたは IQ ページサイズのデータベースはロードできません。このため、**-gp** スイッチを使用してカタログページサイズを最大値 32768 バイトに設定します。

パーミッション制限

データベースを起動するのに必要なパーミッションレベルは、**-gd** サーバコマンドラインオプションによって決定されます。デフォルトでは、このオプションは **DBA** に設定され、**SERVER OPERATOR** システム権限を持つユーザのみが SAP Sybase IQ データベースを起動できます。このオプションを **ALL** または **NONE** にも設定できます。**ALL** は、すべてのユーザがデータベースを起動できることを意味します。**NONE** は、**SERVER OPERATOR** システム権限を持つユーザを含め、ユーザがデータベースを起動できないことを意味します。

データベースの停止

データベースを停止するには、次の方法があります。

- 接続文字列で起動したデータベースとの接続を切断する。**AUTOSTOP** 接続パラメータを **NO** に明示的に設定しない場合、最後のユーザが接続を切断すると、自動的にデータベースは停止します。
- Interactive SQL または Embedded SQL から、**STOP DATABASE** 文を使用する。

『リファレンス：文とオプション』の「**STOP DATABASE** 文 [Interactive SQL]」を参照してください。

参照：

- **STOP DATABASE** 文 [Interactive SQL] (549 ページ)

iqdemo データベースの起動

インストール時に提供されたスクリプトを使用して、iqdemo データベースと、データベースを起動するための設定ファイルを作成します。

設定ファイル iqdemo.cfg には、デモデータベースを起動するのに必要なすべてのパラメータが含まれています。『クイックスタート』を参照してください。

SAP Sybase IQ サーバおよびデータベースへの接続

SAP Sybase IQ はクライアント／サーバ環境で実行され、ネットワークを通じて多数のユーザがデータベースサーバに接続できます。

複数のデータベースサーバに接続する必要がある場合は、サーバを明確に識別する接続オプションを指定します。

注意： Windows または Linux クライアント上の Sybase Central または Interactive SQL から、UNIX サーバ上の SAP Sybase IQ に接続できます。

クライアントアプリケーションは、ODBC、OLE DB、Embedded SQL の各アプリケーション、Sybase Central、Interactive SQL からデータベースに接続できます。

データベースを使用するクライアントアプリケーションは、まず、データベースへの接続を確立する必要があります。接続は、クライアントアプリケーションのすべてのアクティビティを実行するチャンネルとなります。たとえば、ユーザがデータベース上で実行できる操作はユーザ ID によって決定されますが、このユーザ ID は接続確立要求の一部として送信され、データベースサーバに渡されます。

接続ロードマップ

ロードマップは、特定の接続要求を解決するのに最適なトピックを示しています。

接続要求	関連する最適なトピックのリンク
クライアントアプリケーションを使用した接続	「サポートされている接続インタフェース」
接続文字列のカスタマイズ	「接続パラメータと通信パラメータ」
データソースの作成	「ODBC データソース」
ネットワーク接続問題の診断	「ネットワーク通信のトラブルシューティング」

参照：

- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)

- [Interactive SQL 接続 \(65 ページ\)](#)
- [ファイルデータソース \(67 ページ\)](#)
- [他のデータベースからの接続 \(69 ページ\)](#)
- [接続をテストする方法 \(70 ページ\)](#)
- [統合化ログイン \(71 ページ\)](#)
- [接続プール \(77 ページ\)](#)
- [テンポラリ接続 \(80 ページ\)](#)
- [論理サーバの接続 \(81 ページ\)](#)
- [接続を終了する方法 \(82 ページ\)](#)
- [接続ログ \(83 ページ\)](#)
- [付録：接続パラメータと通信パラメータリファレンス \(385 ページ\)](#)
- [ネットワーク通信のトラブルシューティング \(359 ページ\)](#)

接続方法

アプリケーションから SAP Sybase IQ に接続するには、さまざまな方法があります。

ローカルサーバ上の `iqdemo` データベースを問い合わせるために必要な接続パラメータは、次に記載するもののみです。

多くの IQ アプリケーション環境では、より複雑な接続パラメータが必要です。

参照：

- [接続ロードマップ \(47 ページ\)](#)
- [サポートされている接続インタフェース \(60 ページ\)](#)
- [接続ステータス \(61 ページ\)](#)
- [接続パラメータの構造 \(62 ページ\)](#)
- [ODBC データソースの接続パラメータ \(63 ページ\)](#)
- [Interactive SQL 接続 \(65 ページ\)](#)
- [ファイルデータソース \(67 ページ\)](#)
- [他のデータベースからの接続 \(69 ページ\)](#)
- [接続をテストする方法 \(70 ページ\)](#)
- [統合化ログイン \(71 ページ\)](#)
- [接続プール \(77 ページ\)](#)
- [テンポラリ接続 \(80 ページ\)](#)
- [論理サーバの接続 \(81 ページ\)](#)

- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

Interactive SQL からデモデータベースへの接続

このマニュアルで紹介している多くの例では、**dbisql** と呼ばれる Interactive SQL からデモデータベースに接続することから始まります。

1. [プログラム] > [Sybase] > [Sybase IQ 16.0] > [Interactive SQL] を選択します。
2. [ID] タブで、[ユーザ] と [パスワード] にそれぞれ DBA と sql を入力します。
3. [データベース] タブで、[検索] を選択します。
4. [サーバの検索] 画面から iqdemo サーバを選択し、[OK] をクリックします。

同じ手順を使用して、すでに実行中であるデータベースサーバに接続できます。デフォルト以外の文字セットと言語も指定できます。

データベースステータスによるローカル接続への影響

最も単純な接続シナリオは、接続先のデータベースが同じマシン上にある場合です。

その場合は、まず次の質問に答えてください。

- 接続先のデータベースがすでにサーバ上で実行されていますか。実行されている場合は、[接続] ダイアログボックスで指定するパラメータの数が少なくても済みます。実行されていない場合は、Interactive SQL が起動できるよう、データベースファイルの識別情報を入力してください。
- 同じマシン上で複数のデータベースが実行されていますか。実行されている場合は、Interactive SQL から接続するデータベースの識別情報を入力してください。1つしかデータベースがない場合は、Interactive SQL はそのデータベースに接続します。[接続] ダイアログボックスでデータベースを指定する必要はありません。

ローカルサーバ上で実行しているデータベースへの接続

データベースがローカルサーバ上ですでに実行している場合は、[接続] ダイアログボックスで指定するパラメータの数が通常よりも少なくても済むことがあります。

1. Interactive SQL を起動し、[接続] ダイアログボックスを開きます (自動的に開かない場合)。
2. [ID] タブで、ユーザ ID とパスワードを入力します。
3. 次のいずれかを行います。

- サーバに含まれているデータベースが1つしかない場合は、[OK]をクリックしてそのデータベースに接続します。
- サーバ上に複数のデータベースがある場合は、[データベース] タブをクリックし、データベース名を指定します。このとき、通常はデータベースファイルの名前をパスまたは拡張子なしで指定します。

注意： データベースがすでにサーバ上にロードされて開始されている場合は、データベース名を指定するだけで接続を確立できます。データベースファイルは不要です。

UNIX 上の Interactive SQL からデータベースへの接続

ローカルマシン上のデータベースに接続する場合は、ホストとポートを指定する必要はありません。

1. システムコマンドプロンプトでサーバとデータベースを起動します。

```
start_iq dbname
```

2. Interactive SQL を起動します。

```
dbisql -c "uid=userID;pwd=password" -host hostname -port portnum -n servername dbname.db
```

-c パラメータは、接続パラメータを指定します。

たとえば、リモートホスト fiona 上のデモデータベースに接続するには、次のように入力します。

```
dbisql -c "uid=DBA;pwd=sql" -host fiona -port 1870 -n fiona_iqdemo $IQDIR16/demo/iqdemo.db
```

UNIX システムからの接続

UNIX システムのコマンドラインから、実行中のデータベースに接続します。

前提条件

PATH などの環境変数が正しく設定されていることを確認してください。

手順

1. UNIX プロンプトで次のように入力して、デモデータベースが実行中のサーバにロードされていることを確認します。

```
ps -eaf | grep iqdemo
```

```
cd $IQDIR16/demo
```

2. デモデータベースを起動するには、次のように入力します。

```
start_iq @iqdemo.cfg iqdemo.db
```

3. Interactive SQL を起動します。

```
dbisql -c "uid=DBA;pwd=sql;eng=servername;links=tcPIP"
```

servername の部分には、サーバを起動するときに **start_iq** コマンドに指定したものと同一サーバ名を指定します。

注意： Java ベースのバージョンではなく、より古いユーティリティの Interactive SQL Classic を使用する場合は、**dbisqlc** の代わりに **dbisql** を入力します。ただし、**dbisqlc** には **dbisql** の機能の一部が含まれていません。

-c パラメータは、接続パラメータを指定します。これらのパラメータは、この章の後半で説明するように、データソースにも指定できます。

注意： **links=tcPIP** (または **CommLinks=tcPIP**) パラメータは、TCP/IP を使用してデータベースに接続する場合にのみ必要となります。共有メモリポートを使用してローカルデータベースに接続する場合は、**links** パラメータを省略できます。ただし、完全なネットワークパラメータを指定する方が安全です。プラットフォームによっては、完全なネットワークパラメータの指定が必要となります。

リモートホスト上のデータベースに接続するには、ホスト名とポート番号を追加します。次に例を示します。

```
dbisql -c "uid=DBA;pwd=sql;eng=SERV1_iqdemo;links=tcPIP(port=1234)"
```

Windows システムからの接続

[スタート] メニューまたはコマンドプロンプトから、実行中のデータベースに接続します。

1. [プログラム] > [Sybase] > [Sybase IQ 16.0] > [Interactive SQL] を選択するか、Windows のコマンドプロンプトで次のコマンドを入力します。

```
dbisql
```

接続パラメータを指定するには、**dbisql** コマンドに **-c** パラメータを含めます。これらのパラメータを省略すると、Interactive SQL 接続ダイアログボックスが表示されます。

2. [接続] ダイアログボックスで、ユーザ名とパスワードを入力します。
たとえば *iqdemo* データベースの場合は、*DBA* と *sql* (SAP Sybase IQ データベースを作成したときのデフォルトのユーザとパスワードの組み合わせ) を入力します。
3. [データベース] タブをクリックし、サーバの起動時に使用したサーバ名 (たとえば *iqdemo* データベースの場合は *hostname_iqdemo*) を入力します。この名前は、ローカルエリアネットワーク上でユニークである必要があります。

リモートサーバの場合は、[ネットワーク] タブで *host name* と *port number* という形式でサーバを指定します。

デフォルトのポート番号は 2638 ですが、サーバが別の番号を使って開始されている場合は、代わりにその番号を使用します。ポート番号を確認するには、[Sybase IQ 16.0]>[ODBC アドミニストレータ 32 ビット版]または[Sybase IQ 16.0]>[ODBC アドミニストレータ 64 ビット版]を実行します。[ユーザー DSN] タブで[ユーザデータソース]を選択し、[設定] をクリックします。コマンドプロンプトで dblocate と入力します。

この手順では、サーバ上で最初に開始されたデータベースに接続します。複数のデータベースが実行されている場合は、[参照] をクリックして目的のデータベースを選択する必要があります。

4. [OK] をクリックしてデータベースに接続します。

[接続] ダイアログボックス、または情報が不足しているというエラーメッセージが表示されたら、[詳細] タブに **-host** と **-port**、または不足している情報を入力します。データベースがリモートサーバ上にある場合は、**-host** および **-port** パラメータを別の行に入力します。次に例を示します。

```
-host filona  
-port 1870
```

5. データベースに接続すると、Interactive SQL ウィンドウが表示されます。接続のデータベース名、ユーザ ID、およびサーバ名がタイトルバーに表示されます。

Interactive SQL Classic を使用して接続した場合は、[統計情報] ウィンドウに「データベースに接続しました。」というメッセージと、そのデータベースで使用されている照合ラベルについてのメッセージが表示されます。

埋め込みデータベースへの接続

「埋め込みデータベース」は、単一アプリケーションで使用するために設計されており、アプリケーションと同じマシン上で実行され、通常、アプリケーションユーザに対して非表示になっています。

アプリケーションが組み込みデータベースを使用する場合、通常は、アプリケーションの接続時にはデータベースは動作していません。データベースを起動するには、接続文字列を使用し、接続文字列の DatabaseFile (DBF) パラメータでデータベースファイルを指定します。

データベースファイルパラメータ

DBF パラメータは、使用するデータベースファイルを指定します。データベースファイルは自動的にデフォルトサーバにロードされます。実行中のサーバがない場合は、サーバを起動します。

データベースへの接続がなくなると (通常は、その接続を開始したアプリケーションが切斷すると)、データベースはアンロードされます。接続によってサーバが起動された場合は、データベースがアンロードされるとサーバは停止します。

次の接続パラメータを使用して、デモデータベースを組み込みデータベースとしてロードします。

```
dbf=path¥iqdemo.db
uid=DBA
pwd=sql
```

path は、SAP Sybase IQ インストールディレクトリの名前です。

Start パラメータ

次の接続パラメータは、組み込みデータベースとしてのデモデータベースの起動をカスタマイズする方法を示します。これは、キャッシュサイズなどのコマンドラインオプションを使用する場合に便利です。

```
Start=start_iq -gd all
-gl all -gm 10 -gn 25 -gp 4096 -c 32M
-ti 4400 -tl 300
dbf=path¥iqdemo.db
uid=DBA
pwd=sql
```

例：Interactive SQL からの接続

この例では、デモデータベースを Interactive SQL 内の埋め込みデータベースとして使用しています。

Interactive SQL から埋め込みデータベースへの接続 (Windows)

埋め込みデータベースに接続するには、次を行います。

1. データベースを実行していない状態で SAP Sybase IQ を起動します。次のいずれかの方法を使用できます。
 - Windows の [プログラム] メニューから [Sybase] > [Sybase IQ 16.0] > [Interactive SQL] を選択します。
 - システムのコマンドプロンプトで `dbisql` と入力します。

Interactive SQL は、起動時にはどのデータベースにも接続されていません。

2. コマンドウィンドウに `CONNECT` と入力し、[F9] キーを押してコマンドを実行します。接続ダイアログが表示されます。
3. データベース用の ODBC データソースがある場合は、そのデータソースを選択します。
4. ユーザ ID として `DBA` を、パスワードとして `sql` を入力します。[データベース] タブをクリックします。[データベースファイル] フィールドにデモデータ

ベースのフルパスを入力します。たとえば、インストールディレクトリが c:\¥sybase¥IQ-16_0 の場合、次のように入力します。

```
c:\¥sybase¥IQ-16_0¥iqdemo.db
```

5. その他のフィールドには何も入力しないで、[OK] をクリックします。SAP Sybase IQ が起動し、デモデータベースがロードされます。SAP Sybase IQ はデモデータベースに接続します。

TCP/IP プロトコル

SAP Sybase IQ は、異なるコンピュータ上で実行中のデータベースにクライアントを接続するために TCP/IP を使用します。

TCP/IP プロトコルを使用するときは、トランスポートレイヤセキュリティと RSA 暗号化テクノロジーを使用して、クライアント/サーバ通信を安全化できます。

UDP

UDP は、トランスポートレイヤのプロトコルです。これは IP の最上部のプロトコルです。SAP Sybase IQ は、初期のサーバ名解決を行うのに UDP を使用し、その後、TCP を使用して接続と通信を行います。

クライアントのブロードキャストへの応答としてデータベースサーバから送信される UDP パケットには機密情報が含まれません。これらのパケットに含まれるデータは次のとおりです。

- データベースサーバ名
- ポート番号
- データベースサーバのバージョン
- データベースサーバで実行されているデータベースの名前

-dh データベースオプションを使用すると、UDP ブロードキャスト応答に対してデータベース名を難読化することができます。

-sb サーバオプションを指定すると、UDP リスナを無効にできます。

Windows での TCP/IP の使用

すべての Windows プラットフォーム上のデータベースサーバの TCP/IP 実装では、Winsock 2.2 を使用します。Windows Mobile 上のクライアントは、Winsock 1.1 standard を使用します。

TCP/IP を使用したクライアント/サーバ通信の暗号化

デフォルトでは通信パケットが暗号化されないため、セキュリティに関して潜在的な危険があります。単純暗号化またはトランスポートレイヤセキュリティを使用して、TCP/IP を介したクライアントアプリケーションとデータベースサーバ間の通信を安全化できます。トランスポートレイヤセキュリティにより、サーバ認

証、RSA 暗号化テクノロジーを使用した強力な暗号化、およびデータ整合性を保護するその他の機能が提供されます。

SAP Sybase IQ での IPv6 サポート

IPv6 対応のコンピュータでは、ネットワークデータベースサーバは、デフォルトで IPv4 と IPv6 のすべてのアドレスを受信します。IPv6 は、Windows CE ではサポートされません。

通常、IPv6 を使用するようにデータベースサーバの開始行を変更する必要はありません。IP アドレスの指定が必要な場合は、データベースサーバライブラリとクライアントライブラリは、どちらも IPv4 アドレスと IPv6 アドレスを受け付けます。たとえば、コンピュータで複数のネットワークカードを使用できる場合、2つの IPv4 アドレスと2つの IPv6 アドレスが存在することがあります。IPv6 アドレスの場合は、アドレスを角カッコまたはカッコで囲む必要があります。データベースサーバが受信する IPv6 アドレスを1つに制限する場合は、アドレスを次のフォーマットで指定できます。

```
iqsrv16 -x tcpip(MyIP=fd77:55f:5a64:52a:202:5445:5245:444f) ...
```

同様に、クライアントアプリケーションでサーバの IP アドレスを指定する必要がある場合は、接続文字列または ODBC データソースに次のフォーマットでアドレスを指定できます。

```
...HOST=fd77:55f:5a64::444f;...
```

各インタフェースにインタフェース識別子が与えられており、IPv6 アドレスの末尾に示されます。たとえば、ipconfig.exe にアドレス fd77:55f:5a64::444f がリストされた場合、インタフェース識別子は 7 です。IPv6 アドレスを Windows プラットフォームに指定する場合は、インタフェース識別子を使用する必要があります。UNIX では、インタフェース識別子とインタフェース名のどちらでも指定できます (インタフェース名は、ifconfig によってレポートされるインタフェース識別子の名前です)。たとえば、IPv6 アドレスが fd77:55f:5a64::444f; である場合、インタフェース名は eth1 です。Linux (カーネル 2.6.13 以降) で IPv6 アドレスを指定する場合は、インタフェース識別子が必要です。この要件は次によって指定された値に影響します。

例

ipconfig.exe に2つのインタフェースがリストされており、片方の識別子が1でもう片方の識別子が2だとします。インタフェース番号2によって使用されているネットワーク上のデータベースサーバを検索している場合、クライアントライブラリに対してそのインタフェースにだけブロードキャストするよう指示できます。

```
LINKS=tcpip (BROADCAST=ff02::1%2)
```

IPv6 リンクローカルのマルチキャストアドレスは `ff02::1` です。

ファイアウォール接続

クライアントアプリケーションがファイアウォールの片側にあり、データベースサーバがもう一方の側にある場合は、接続が制限されます。ファイアウォールソフトウェアは、ネットワークポートに従って、ネットワークパケットをフィルタリングする可能性があります。また通常は、UDP パケットはファイアウォールを通過できません。

通常、接続は、Host 接続パラメータを使用して適切に設定されたファイアウォールを経由し、データベースサーバのアドレスとポートを提供することで実行できます。データベースサーバがデフォルトポート 2638 を使用している場合、ポートは必要ありません。

次の接続文字列は、ポート 2020 を使用するアドレス `serverhost` のコンピュータ上で実行中の `myserver` という名前のデータベースサーバに接続します。Host 接続パラメータが TCP/IP アドレスとポートを指定しているので、UDP パケットは使用されません。

```
Server=myserver;Host=serverhost:2020
```

特定のクライアントポートのみを許可するファイアウォール

特定のクライアントポートのみを許可するようにファイアウォールを設定する必要がある場合、Host 接続パラメータではなく、CommLinks(LINKS) 接続パラメータを使用する必要があります。次の TCP/IP プロトコルオプションは、CommLinks 接続パラメータを使用するときが必要です。

- **Host** – このプロトコルオプションには、データベースサーバを実行しているホスト名を設定します。IP と短縮してもかまいません。
- **ServerPort** – データベースサーバがデフォルトポート 2638 を使用していない場合、使用しているポートを指定します。Port と短縮してもかまいません。
- **ClientPort** – このプロトコルオプションには、使用するクライアントアプリケーションで有効な範囲の値を設定します。CPORT と短縮してもかまいません。
- **DoBroadcast=NONE** – サーバに接続するときに UDP が使用されないようにするには、このプロトコルオプションを設定します。DOBROAD と短縮してもかまいません。

ファイアウォールは、SAP Sybase IQ データベースサーバのアドレスとすべての SAP Sybase IQ クライアントのアドレスとの間の TCP/IP トラフィックを許可するように設定します。SAP Sybase IQ データベースサーバのアドレスは、SAP Sybase IQ サーバを実行中のコンピュータの IP アドレス (HOST プロトコルオプション) と SAP Sybase IQ データベースサーバの IP ポート番号 (ServerPort プロトコルオプション、デフォルトは 2638) です。

クライアントポートが再び利用されるまでに数分のタイムアウトがあるため、各クライアントコンピュータからの同時接続の最大数よりも多いクライアントポート数の範囲を使用します。ClientPort プロトコルオプションで指定されるクライアントの範囲は、ファイアウォールで許可される範囲と一致する必要があります。

例

次の接続文字列は、クライアントアプリケーションをポート 5050 ~ 5060 に制限します。また、サーバポート 2020 を使用するアドレス myhost のコンピュータで実行されているデータベースサーバ myeng に接続します。DoBroadcast プロトコルオプションを NONE に設定しているため、UDP ブロードキャストは実行されません。

```
Server=myeng;LINKS=tcpip(ClientPort=5050-5060;HOST=myhost;PORT=2020;DoBroadcast=NONE)
```

ダイヤルアップネットワーク接続 (CommLinks 接続パラメータ)

ダイヤルアップネットワーク接続経由で接続するには、CommLinks 接続パラメータを使用する必要があります。

クライアント側では、以下のプロトコルオプションを指定してください。

- **Host – Host (IP)** プロトコルオプションを使用して、データベースサーバのホスト名または IP アドレスを指定します。
- **DoBroadcast – Host (IP)** プロトコルオプションを指定した場合は、データベースサーバでブロードキャスト検索を行う必要はありません。このため、ダイレクトブロードキャストを使用してください。
- **MyIP** – クライアント側では、MyIP=NONE に設定してください。
- **TIMEOUT** – サーバを検索する間のクライアントの待機時間を長くするには、TIMEOUT (TO) プロトコルオプションを設定します。

例

次に示すのは、ダイヤルアップネットワーク接続経由で接続するために使用する一般的な CommLinks (LINKS) 接続パラメータの例です。

```
LINKS=tcpip(MyIP=NONE;DoBroadcast=DIRECT;HOST=server-ip);  
TIMEOUT=15)
```

データソースを使用した接続

一連の接続パラメータをデータソースに保存できます。Interactive SQL Classic などの Embedded SQL アプリケーションと同様に、iAnywhere JDBC ドライバを使用する ODBC と JDBC は、データソースを使用します。

データソースは ODBC アドミニストレータから作成できます。

すべてのアプリケーションに対して、データソースの使用が有効です。

iqdemo データソースには、データベースファイルや、デモデータベースを起動するための Start パラメータなど、一連の接続パラメータが格納されています。データソースのサーバ名は "hostname_iqdemo" となり、hostname は使用しているシステム名です。

ODBC データソースを使用した接続

Interactive SQL から ODBC データソースを使用して接続します。

1. Interactive SQL を起動し、[接続] ダイアログボックスを表示します (自動的に表示されない場合)。
2. [ID] タブ (Interactive SQL Classic の場合は [ログイン] タブ) で、ユーザ ID とパスワードを、たとえば DBA と sql と入力します。
3. [ID] タブの下部で、次のいずれかを指定します。
 - [ODBC データソース名] オプションを選択し、データソース名を指定します (これは、レジストリ内のデータソースを参照する DSN 接続パラメータに相当します)。[参照] をクリックすると、データソースのリストが表示されます。
 - [ODBC データソースファイル] オプションを選択し、データソースファイルを指定します (これは、ファイル内のデータソースを参照する FileDSN 接続パラメータに相当します)。[参照] をクリックしてファイルを検索することもできます。

SAP Sybase IQ デモデータソースには、データベースファイルや起動パラメータなど、一連の接続パラメータが格納されています。

注意： Interactive SQL 起動時に次のように **dsn** 接続パラメータを使用すると、データソース名も指定できます。

```
dbisql -c "dsn=Sybase IQ Demo"
```

デフォルト接続パラメータ

接続時にデフォルト動作を使用する場合は、多くの接続パラメータの指定を省略できます。

注意： ただし、運用環境でデフォルト動作を使用するときには注意する必要があります。特に、他の SAP Sybase IQ アプリケーションや SQL Anywhere アプリケーションをマシンにインストールする可能性のある顧客にアプリケーションを配布する場合は、注意が必要です。

デフォルトデータベースサーバ

ローカルサーバにデータベースが接続され、そのサーバで複数のデータベースが開始されている場合、接続先のデータベースを指定する必要がありますが、サーバをデフォルトのままにすることもできます。

```
dbn=db_name  
uid=user_id  
pwd=password
```

注意：複数のローカルサーバが実行されている場合は、このパラメータを使用しないでください。正しいサーバに接続できないことがあります。

デフォルトデータベース

複数のサーバが動作している場合、接続先を指定してください。サーバで起動されているデータベースが1つのみの場合は、データベース名を指定する必要はありません。次の接続文字列は、デフォルトデータベースを使用して、指定されたサーバに接続します。

```
eng=server_name  
uid=user_id  
pwd=password
```

デフォルトを使用しない接続

次の接続文字列は、指定されたデータベースを使用して、指定されたサーバに接続します。

```
eng=server_name  
dbn=db_name  
uid=user_id  
pwd=password
```

SAP Sybase IQ のユーティリティからの接続

データベースファイルを直接処理するのではなくサーバと通信する SAP Sybase IQ データベースユーティリティは、サーバとの通信を Embedded SQL によって実現します。

データベースユーティリティが接続パラメータの値を取得する方法

ほとんどの管理ユーティリティでは、次のようにして接続パラメータ値を取得します。

コマンドラインで値を指定した場合は、その値を使用します。たとえば、以下のコマンドは、ユーザ ID DBA とパスワード sql を使用して、demo データベースでカタログストアのバックアップを取得します。

```
dbbackup -y -x -c  
'uid=DBA;pwd=sql;eng=iqdemo;dbn=iqdemo.db;links=tcPIP{port=2638}' -  
d '/mydir'
```

1. コマンドラインの値がない場合は、SQLCONNECT 環境変数設定を使用します。この変数は、SAP Sybase IQ データベースユーティリティで自動的に設定されません。この方法を使用すると、他の方法よりもパスワードのセキュリティが向上します。SQLCONNECT 環境変数の詳細については、『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』>「ファイルロケーションとインストール設定」>「環境変数」を参照してください。
2. コマンドラインまたは SQLCONNECT 環境変数でパラメータが設定されていない場合、デフォルトサーバのデフォルトデータベースに接続するためのユーザ ID とパスワードが要求されます。

各データベースユーティリティのコマンドラインオプションの詳細については、『ユーティリティガイド』を参照してください。

サポートされている接続インタフェース

接続を確立するために、クライアントアプリケーションはサポートされているインタフェースのいずれかで関数を呼び出します。

SAP Sybase IQ では次のインタフェースがサポートされています。

- ODBC
- OLE DB
- Embedded SQL
- SAP Sybase Open Client — 『プログラミング』の「クライアントアプリケーションのデータサーバとしての SAP Sybase IQ」を参照してください。
- JDBC — 『プログラミング』の「JDBC CLI」を参照してください。Interactive SQL は、iAnywhere JDBC ドライバをサポートしています。これは、SAP Sybase IQ に搭載されています。

注意： iAnywhere JDBC ドライバは非推奨になりました。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)

- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

接続ステータス

クライアントツールによっては接続状態がはっきり表示されず、コマンドが失敗したときに初めて接続されていないことがわかる場合があります。

接続を迅速に確認する方法は、データベース名をクエリすることです。

現在のデータベースを表示するには、次の構文を使用します。

```
select db_name()
```

別のデータベースを指定するには、次の構文を使用します。

```
select db_name([ database_id ])
```

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)

- 接続ログ (83 ページ)

接続パラメータの構造

アプリケーションからデータベースに接続する場合は、一連の「*接続パラメータ*」を使用して接続を定義します。

接続パラメータには、サーバ名、データベース名、ユーザ ID などの情報が含まれています。

各接続パラメータは、キーワードと値の組み合わせとして、*parameter=value* という形式で指定されます。たとえば、デフォルトパスワードのパスワード接続パラメータは、次のように指定します。

```
Password=sql
```

接続パラメータを組み合わせると接続文字列を作ります。接続文字列では、各接続パラメータを、次に示すようにセミコロンで区切ります。

```
ServerName=host_iqdemo;DatabaseName=iqdemo
```

一部の接続パラメータはサーバの起動方法に影響を与えます。StartLine (START) 接続パラメータ内でのサーバオプションの指定ではなく、次の接続パラメータの使用をおすすめします。

- EngineName (ENG)
- DatabaseFile (DBF)
- DatabaseName (DBN)

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)

- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

接続文字列の形式

このマニュアルでは、さまざまな接続文字列の例を次の形式で示します。

```
parameter1=value1  
parameter2=value2  
...
```

これは、次の接続文字列に相当します。

```
parameter1=value1;parameter2=value2
```

接続文字列は 1 行に入力し、各パラメータ設定をセミコロンで区切ります。

アプリケーションが接続パラメータを渡す方法

接続パラメータは、「*接続文字列*」としてインタフェースライブラリに渡されます。

この文字列は、セミコロンで区切った複数のパラメータから構成されます。

通常、アプリケーションによって構築され、インタフェースライブラリに渡される接続文字列は、ユーザが情報を入力する方法と直接対応していません。代わりに、ユーザがダイアログボックスに入力するか、アプリケーションが初期化ファイルから接続情報を読み取ることができます。

SAP Sybase IQ のユーティリティの中には、接続文字列を **-c** コマンドラインオプションとして受け入れ、その接続文字列を変更しないでインタフェースライブラリに渡すものもあります。たとえば、サーバ `myserver` 上の `iqdemo` という名前のデータベースを停止するには、次のコマンドを入力します。

```
dbstop -c "uid=DBA;pwd=sql;eng=myserver;dbn=iqdemo"
```

注意： Interactive SQL は接続文字列を内部で処理します。DBISQL は、単に接続パラメータをインタフェースライブラリに渡すわけではありません。コマンドプロンプトからの接続文字列のテストには、Interactive SQL を使用しないでください。

ODBC データソースの接続パラメータ

アプリケーション開発システムなど、多くのクライアントアプリケーションは、SAP Sybase IQ へのアクセスに ODBC インタフェースを使用します。

ODBC アプリケーションがデータベースに接続するときには、通常は ODBC データソースを使用します。ODBC データソースは接続パラメータの集まりであり、レジストリまたはファイルに保存されます。

SAP Sybase IQ の場合は、Windows 上の ODBC アプリケーションだけでなく、次のアプリケーションも ODBC データソースを使用できます。

- UNIX 上の SAP Sybase IQ クライアントアプリケーションは、Windows オペレーティングシステム上のクライアントアプリケーションと同様に ODBC データソースを使用できます。UNIX 上では、データソースはファイルとして保存されます。
- ODBC アプリケーションに加えて、OLE DB インタフェースまたは Embedded SQL インタフェースを使用する SAP Sybase IQ クライアントアプリケーションも、ODBC データソースを使用できます。
- Interactive SQL は ODBC データソースを使用できます。
- iAnywhere JDBC ドライバを使用した JDBC 接続では、ODBC データソースを使用できます。

注意： iAnywhere JDBC ドライバは非推奨になりました。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

Interactive SQL 接続

Interactive SQL を使用してデータベースを管理するには、まずデータベースに接続する必要があります。

[接続] ダイアログボックスで、接続先のデータベースの名前、場所、接続方法を指定します。

接続プロセスは状況によって異なります。たとえば、同じマシン上ですでにサーバを実行していて、このサーバにデータベースが 1 つしかない場合は、[接続] ダイアログボックスにユーザ ID とパスワードを入力するだけで済みます。これだけで、Interactive SQL は実行中のサーバ上のデータベースに接続します。

実行中のサーバ上に複数のデータベースがロードされている場合、またはサーバを実行していない場合、またはサーバを別のマシン上で実行している場合は、Interactive SQL が適切なデータベースに接続できるように、[接続] ダイアログボックスにもっと詳しい情報を入力する必要があります。

この項では、Interactive SQL の [接続] ダイアログボックスの使用方法を説明します。

注意： わかりやすくするために、デフォルトのパラメータではなく Interactive SQL の接続パラメータを指定します。接続パラメータは、コマンドライン、設定ファイル、または初期化ファイル (.odbc.ini や odbc.ini など) で指定できません。

たとえば、1 つのサーバ上で複数のデータベースを実行している場合は、データベース名の指定が必要です。サブネットが設定されたネットワークの場合は、ホスト番号などのプロトコルオプションを持つ **CommLinks** パラメータを指定します。

.odbc.ini ファイルでは、各パラメータを長い形式で指定します。たとえば、DBF ではなく DatabaseFile を使用します。パラメータが不完全、または正しくない場合は、次のようなエラーが表示されることがあります。

Database name required to start engine

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)

- ODBC データソースの接続パラメータ (63 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

[接続] ダイアログ ボックス

[接続] ダイアログボックスでは、サーバまたはデータベースに接続するためのパラメータを定義します。

[接続] ダイアログボックスには次のタブがあります。

- [ID] タブでは、データベースに対する ID 情報を指定し、データソースを指定します。
- [データベース] タブでは、接続先のサーバかデータベースまたはその両方を指定します。
- [ネットワーク] タブでは、共有メモリまたは TCP/IP 接続プロトコルの指定、セキュリティオプションの選択、暗号化パラメータの指定を行います。
- [詳細] タブでは、接続パラメータを追加し、接続用のドライバを指定します。

接続が確立すると、メインウィンドウの左側のウィンドウ枠に、接続したデータベースの名前が表示されます(そのデータベースを実行しているサーバ名の下に表示されます)。データベース名の後ろにこの接続のユーザ ID が表示されます。

Interactive SQL で接続すると、その接続情報 (データベース名、ユーザ ID、データベースサーバを含む) が [SQL 文] ペインの上のタイトルバーに表示されます。

[接続] ダイアログボックスを開く (Interactive SQL)

Interactive SQL の [接続] ダイアログボックスを使用することにより、データベースに接続できます。

Interactive SQL では、[ファイル]>[新規ウィンドウ] または [SQL]>[接続] を選びます。

[F11] キーを押して [接続] ダイアログボックスを表示することもできます。

[接続] ダイアログボックスが表示されたら、接続に必要な接続パラメータを指定します。たとえば、SAP Sybase IQ デモデータベースに接続するには、[データベー

ス] タブの [参照] ボタンを使用してデータベースファイルとして `iqdemo.db` を指定し、[ID] タブで `User ID DBA` と `Password sql` を入力して、[OK] をクリックします。

サーバがリモートの場合は、[データベース] タブの [ネットワーク上でデータベースサーバを検索] が選択されていることを確認します。

注意： ユーザによって作成されたデータベースに接続するときは、[データベースファイル] フィールドと [データベース名] フィールドの両方を指定する必要があります。完全なパス名を指定してください。

接続用のドライバ

データベースを使用する際、要求やコマンドはすべてドライバを通じてデータベースに送信されます。

サポートされているドライバとプログラミングインタフェースについては、『プログラミング』で説明します。

ファイルデータソース

Windows オペレーティングシステムでは、ODBC データソースを通常システムレジストリに格納します。ファイルデータソースは、ファイルとして保存されるデータソースです。ファイルデータソースは Windows システムと UNIX システムの両方でサポートされています。

通常、Windows のファイルデータソースの拡張子は `.dsn` です。これは複数のセクションで構成され、各セクションは角カッコで囲まれた名前で始まります。DSN ファイルのレイアウトは、初期化ファイルに非常に似ています。

ファイルデータソースを使用して接続するには、**FileDSN** 接続パラメータを使用します。同じ接続の中で **DSN** と **FileDSN** を併用することはできません。

分散可能なファイルデータソース

ファイルデータソースの利点の 1 つは、ファイルをユーザに配布できることです。これにより、接続情報を各マシン上で作成し直す必要はなくなります。ファイルがファイルデータソースのデフォルト位置にある場合、ODBC によって自動的に選択されます。このように、多くのユーザの接続の管理が簡単になります。

注意： DSN は Windows のレジストリに保存されるため、パブリックな情報となります。したがって、暗号化していないパスワードを DSN に保存しないでください。データソースにパスワードを保存する場合は、FileDSN を使用します。

Embedded SQL アプリケーションでも ODBC ファイルデータソースを使用できません。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

ODBC アドミニストレータでのファイルデータソースの作成

ODBC アドミニストレータツールを使用して、ファイルデータソースを作成できます。

1. ODBC アドミニストレータを起動し、[File DSN] タブを表示して [Add] をクリックします。
2. ドライバのリストから SAP Sybase IQ を選択し、[次へ] をクリックします。
3. データソースの作成手順に従います。

ファイルデータソースとテキストエディタ

ファイルデータソースはテキストファイルであるため、どのテキストエディタでも編集できます。テキストエディタを使用する場合の制限事項の 1 つは、このファイルに暗号化パスワードを保存できないことです。

ファイルデータソースの例

```
[Sample File Data Source]
ENG = iqdemo
```

```
DBA = DBA  
PWD = sql
```

『ユーティリティガイド』の「iqdsn データベース管理ユーティリティ」を参照してください。

他のデータベースからの接続

SAP Adaptive Server® Enterprise から、SAP Sybase IQ テーブル内のデータを外部データソースとして使用できます。この機能の利点は、コンポーネント統合サービス(CIS)インタフェースを使用して、分散している異種ソースのデータをクライアントから使用できることです。

CIS を使用して、SAP Sybase IQ テーブルを示す「プロキシテーブル」を Adaptive Server 内に定義します。その後、Adaptive Server からプロキシテーブルにクエリを実行することができます。ASE の『コンポーネント統合サービス・ユーザーズ・ガイド』を参照してください。

CIS と SAP Sybase IQ を使用することによって、いろいろな方法で他のデータベースに接続したり、データを共有したりできます。たとえば、ユーザアプリケーションが共通のインタフェースを通じてデータウェアハウス全体にアクセスすることもできます。CIS を使用すると、次のことができます。

- SAP Sybase IQ から Adaptive Server データベース内のデータへのアクセス。この機能は、特定のプラットフォームでのみサポートされています。使用しているプラットフォームについては、『インストールおよび設定ガイド』を参照してください。
- 他のデータベースサーバ上にある SAP Sybase IQ および SQL Anywhere データベース内のデータへのアクセス。
- 外部データソース (他のベンダのリレーショナルデータベース、Excel スプレッドシートデータ、テキストファイルなど) へのアクセス。
- 別の SAP Sybase IQ データベースからのテーブルのジョイン。

『プログラミング』の「リモートデータへのアクセス」を参照してください。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)

- ファイルデータソース (67 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

UNIX 上でのポート番号の競合防止

設定ファイルを更新して、ポート番号の競合を防止します。

1. 次の行を未使用のポート番号とともに \$IQDIR16/scripts/default.cfg に追加します。次に例を示します。

```
-x tcpip{port=4444}
```

2. 各設定ファイル内でポート番号の定義を検索します。たとえば、/usr/summers/mydemo/iqdemo.cfg には次の行が指定されています。

```
-x tcpip{port=2638}
```

3. 行を編集して、デフォルトのポート番号を新しい番号に置換します。以下に例を示します。

```
-x tcpip{port=4444}
```

4. 作業が終了したら各ファイルを保存します。

SQL Anywhere が SAP Sybase IQ と同じサブネット上にインストールされている場合は、サーバに一意の名前を指定する必要があります。

接続をテストする方法

dbping コマンドラインユーティリティは、接続のトラブルシューティングに役立ちます。これを使用すると、ネットワーク上で特定の名前のサーバが使用可能かどうかをテストできます。

dbping ユーティリティは接続文字列をコマンドラインオプションとして受け取りますが、デフォルトでは、サーバを検出するために必要な部分だけが使用されます。このユーティリティでは、サーバを起動しません。

次の行は、Ciaran という名前のサーバを TCP/IP 接続上で使用できるかどうかをテストします。

```
dbping -c "eng=Ciaran;CommLinks=tcpip"
```


次のコマンドでは、現在のマシン上のデフォルトサーバが使用できるかどうかをテストします。

```
dbping
```

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

統合化ログイン

統合化ログイン機能では、データベース接続とオペレーティングシステム/ネットワークログインの両方に対して、単一のユーザ ID とパスワードを使用できます。

サポートされているオペレーティングシステム

統合化ログイン機能は、Windows サーバでのみ利用できます。サポートされる Windows プラットフォーム上のクライアントは、統合化ログインを使用して、Windows 上のネットワークサーバに接続できます。

統合化ログインの利点

統合化ログインは、1つまたは複数の Windows ユーザプロファイルをデータベース内の既存のユーザにマッピングする機能です。ユーザプロファイルのセキュリティをナビゲートし、マシンへのログインに成功したユーザは、他のユーザ ID またはパスワードを指定しないでデータベースに接続できます。

そのためには、データベースでの統合化ログインの使用を有効にし、マシンやネットワークへのログインに使用するユーザプロファイルとデータベースユーザの間をマッピングする必要があります。

統合化ログインを使用すると、ユーザにとって便利なだけでなく、データベースとネットワークの両方のセキュリティを1つのセキュリティシステムで維持できます。次のような利点があります。

- 統合化ログインを使用してデータベースに接続する場合、ユーザはユーザ ID またはパスワードを入力する必要がありません。
- 統合化ログインを使用する場合、ユーザの認証はデータベースではなくオペレーティングシステムによって行われます。データベースのセキュリティと、マシンやネットワークのセキュリティには、単一のシステムが使用されます。
- 複数のユーザプロファイルを1つのデータベースユーザ ID にマッピングできます。
- Windows マシンへのログインに使用する名前とパスワードは、データベースユーザの ID とパスワードに一致している必要はありません。

警告！ 統合化ログインによって提供される単一のセキュリティシステムは便利ですが、データベース管理者が熟知しておく必要のある重要なセキュリティ上の考慮事項があります。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

統合化ログインの使用方法

統合化ログインを使用して正常に接続するには、いくつかの手順を実行する必要があります。

1. LOGIN_MODE データベースオプションの値を、デフォルト値である **Standard** の代わりに **Mixed** または **Integrated** (大文字と小文字の区別はなし) に設定して、データベース内で統合化ログイン機能を有効にします。この手順には、SET ANY SECURITY OPTION システム権限が必要です。
2. ユーザプロファイルと既存のデータベースユーザの間に、統合化ログインマッピングを作成します。これには、SQL 文を使用します。
3. 統合化ログイン機能がトリガされる方法で、クライアントアプリケーションから接続します。

以上の各手順については、この後の項で説明します。

統合化ログイン機能の有効化

login_mode データベースオプションは、統合化ログイン機能が有効かどうかを判断します。データベースオプションが適用されるのは、それが指定されているデータベースに対してだけであるため、同じサーバ内にロードされて動作している場合でも、データベースが異なれば統合化ログインの設定も異なります。

LOGIN_MODE データベースオプションには、次の3つの値のいずれかが設定されます (大文字と小文字の区別なし)。

- Standard – 統合化ログインは使用できません。これがデフォルト設定です。統合化ログインを使用して接続しようとする、エラーが発生します。
- Mixed – 統合化ログインと標準ログインの両方を使用できます。
- Integrated – この設定では、データベースへのすべてのログインを統合化ログインで行う必要があります。

警告！ LOGIN_MODE データベースオプションを **Integrated** に設定すると、統合化ログインマッピングを付与されているユーザのみに接続が制限されます。ユーザ ID とパスワードを使用して接続しようとする、エラーが生成されます。ただし、MANAGE ANY USER システム権限を持つユーザは例外です。

SQL での統合化ログインの作成

ユーザプロファイルは、既存のデータベースユーザ ID に対してのみマッピングできます。データベースからデータベースユーザ ID が削除されると、そのデータ

ベースユーザ ID に基づくすべての統合化ログインマッピングも、自動的に削除されます。

次の SQL 文を使用すると、Windows ユーザの `fran_whitney` と `matthew_cobb` は、DBA ユーザ ID またはパスワードを知らない、または指定しなかった場合でも、ユーザ DBA としてデータベースにログインできます。

```
GRANT INTEGRATED LOGIN  
TO fran_whitney, matthew_cobb  
AS USER DBA
```

詳細については、『リファレンス：文とオプション』の「SQL 文」>「GRANT 文」を参照してください。

参照：

- GRANT INTEGRATED LOGIN 文 (536 ページ)

統合化ログインパーミッションの取り消し (SQL)

Interactive SQL を使用して、統合化ログインマッピングを削除できます。

1. `MANAGE ANY USER` システム権限に関する管理権限を持つユーザとしてデータベースに接続します。
2. `REVOKE INTEGRATED LOGIN FROM` 文を実行します。

クライアントアプリケーションからの接続

クライアントアプリケーションは、次のいずれかの方法で統合化ログインを使用して、データベースに接続できます。

- 接続パラメータのリストで `INTEGRATED` パラメータを **yes** に設定します。
- 接続文字列または接続ダイアログボックスで、ユーザ ID とパスワードをどちらも指定しません。この方法を使用できるのは、SAP Sybase IQ 管理ユーティリティなどの Embedded SQL アプリケーションのみです。

接続文字列で `INTEGRATED=yes` が指定されている場合は、統合化ログインが試行されます。接続が失敗し、`LOGIN_MODE` データベースオプションが **Mixed** に設定されている場合は、サーバは標準ログインを試行します。

ユーザ ID やパスワードを指定しないでデータベースへ接続した場合、統合化ログインが行われます。現在のユーザプロファイル名がデータベース内の統合化ログインマッピングに一致するかどうかによって、接続の正否が決まります。

セキュリティについての考慮事項：無制限データベースアクセス

統合化ログイン機能の基盤になっているのは、SAP Sybase IQ がデータベースへのアクセスの制御に使用するシステムではなく、Windows のログイン制御システムです。

データベースのホストであるマシンにログインでき、この章で説明した他の条件を満たしている場合は、データベースセキュリティを通過します。

ユーザが "dsmith" として Windows サーバに正常にログインした場合、そのユーザは、統合化ログインマッピングかデフォルトの統合化ログインユーザ ID のいずれかがある場合にすぎ、それ以上の資格情報を提示しなくてもデータベースに接続できます。

統合化ログインを使用する場合、データベース管理者は、データベースへの不正なアクセスを制限するために、Windows のログインセキュリティの実施方法に特に注意する必要があります。

特に、Windows のワークステーションやサーバをインストールすると、デフォルトで "Guest" ユーザプロファイルが作成されて、有効になることに注意してください。

警告！ ユーザプロファイル Guest を有効にしておくと、そのサーバがホストとなるデータベースへの無制限なアクセスが許可されます。

Guest ユーザプロファイルが有効で、かつパスワードがブランクの場合、そのサーバに対するログインはすべて成功します。ユーザプロファイルがそのサーバに存在することや、指定されたログイン ID にドメインログインパーミッションがあることは要求されません。そのサーバには、文字どおり、どのユーザがどのログイン ID やパスワードを使用してもログインできてしまいます。このようなユーザは、デフォルトでは Guest ユーザプロファイルにログインします。

統合化ログイン機能を有効にしてデータベースに接続する場合は、このことに注意する必要があります。

次のシナリオを考えてみてください。ここでは、データベースをホストする Windows サーバ上に、ブランクのパスワードで有効になる "Guest" ユーザプロファイルがあるものとします。

- ユーザ dsmith とデータベースユーザ ID DBA との間に、統合化ログインマッピングが存在します。ユーザ dsmith が正しいログイン ID とパスワードを使用してサーバに接続すると、完全な管理権限を持つユーザ DBA としてデータベースに接続されます。
- しかし実際には、他のユーザが "dsmith" としてサーバに接続しようとした場合、どのようなパスワードを指定しても、この接続は成功します。これは、Windows がデフォルトで "Guest" ユーザプロファイルに接続しようとするため

です。いったん "dsmith" のログイン ID を使用してサーバへのログインに成功すると、権限のないユーザでも、統合化ログインマッピングを使用して、DBA としてデータベースに接続できます。

注意：セキュリティのため、"Guest" ユーザプロファイルは必ず無効にしてください。最も安全な統合化ログインポリシーは、SAP Sybase IQ データベースをホストするすべての Windows マシン上で "Guest" を無効にすることです。これは、Windows のユーザマネージャユーティリティを使用して実行できます。

一時的に **public** オプションを設定してセキュリティを追加する

次の SQL 文を使用してデータベースの LOGIN_MODE オプションの値を **Mixed** (非推奨) または **Integrated** に設定すると、そのデータベースの統合化ログインが永続的に有効になります。

```
SET OPTION Public.LOGIN_MODE = Mixed
```

データベースが停止されて再起動した場合でも、このオプションの値は変わらず、統合化ログインも有効のままです。

login_mode オプションを一時的に変更した場合でも、ユーザは統合化ログインを介してアクセスできます。次の文は、オプションの値を一時的に変更します。

```
SET TEMPORARY OPTION "Public".LOGIN_MODE = Mixed
```

永久オプション値が **Standard** の場合、データベースは停止時にその値に戻ります。

データベースアクセスの追加セキュリティ対策として、**public** オプションを一時的に設定することが考えられます。これは、統合化ログインを有効にするということは、データベースが動作しているオペレーティングシステムのセキュリティに依存することを意味するためです。データベースが停止されて別のマシン (ユーザのマシンなど) にコピーされた場合、そのデータベースへのアクセスは、コピー先マシンのオペレーティングシステムのセキュリティモデルではなく SAP Sybase IQ のセキュリティモデルによって制御されるようになります。

『リファレンス：文とオプション』の「SQL 文」>「SET OPTION 文」を参照してください。

参照：

- SET OPTION 文 (547 ページ)

ネットワークから見た統合化ログイン

データベースがネットワークサーバにある場合、統合化ログインを使用するには、次の2つの条件のいずれかが満たされている必要があります。

- 統合化ログイン接続に使用するユーザプロファイルが、ローカルマシンとサーバの両方に存在する必要があります。ユーザプロファイル名、ユーザプロファイルのパスワードが両方のマシンで同じである必要があります。
たとえば、ユーザ **jsmith** が、ネットワークサーバ上にロードされているデータベースに統合化ログインを使用して接続しようとした場合は、ローカルマシンとそのデータベースのホストとなるアプリケーションサーバの両方に、同じユーザプロファイルの名前とパスワードが存在している必要があります。**jsmith** は、ローカルマシンとネットワークサーバのホストとなるサーバの両方に対して、ログインを許可されている必要があります。
- ネットワークアクセスが Microsoft ドメインによって制御されている場合、統合化ログインを試みるユーザは、ドメインコントローラサーバのドメインパーミッションを持っていて、ネットワークにログインしている必要があります。ローカルマシンのユーザプロファイルとネットワークサーバのユーザプロファイルとは、一致している必要はありません。

デフォルトの統合化ログインユーザ

デフォルトの統合化ログインユーザ ID を作成すると、現在使用中のユーザプロファイルに関する統合化ログインマッピングが存在しなくても、統合化ログインを使用した接続を成功させることができます。

たとえば、ユーザプロファイル名 **jsmith** に統合化ログインマッピングが存在しない場合、使用されているユーザプロファイルが **jsmith** であれば、統合化ログイン接続をしようとしても通常は失敗します。

ただし、**Guest** という名前のユーザ ID をデータベースに作成すると、ユーザプロファイル **jsmith** を明示的に識別する統合化ログインマッピングがない場合でも、統合化ログインは **Guest** ユーザ ID に正常にマッピングします。

デフォルトの統合化ログインユーザは、データベースに **Guest** というユーザ ID が含まれている場合は、統合化ログインを試みるすべてのユーザのデータベースへの接続を許可します。新しく接続したユーザに付与されるパーミッションと権限は、**Guest** ユーザ ID に付与されている権限によって決まります。

接続プール

接続プールによって、データベースサーバに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。接続プールが接続に対して有効になっていると、接続は切断されたときに自動的にキャッシュされ、

アプリケーションが再接続された時点で再利用できます。接続プールは、ConnectionPool (CPOOL) 接続パラメータで制御します。同じ接続文字列を使用してアプリケーションで指定回数の接続を行うと、接続がプールされます。

接続をキャッシュするには、同じ接続文字列を使用してアプリケーションで最低でも 5 回の接続を行う必要があります。キャッシュされた接続を再利用するには、接続名はそのつど異なってもかまいませんが、他のすべての接続パラメータは同じにする必要があります。

アプリケーションで接続を再度処理するときに、同じ接続文字列に対して使用できる接続がキャッシュされている場合は、キャッシュされた接続が再利用されず。接続は、ConnectionPool (CPOOL) 接続パラメータで指定された期間中、キャッシュされた状態のままになります (デフォルトでは 60 秒)。

アプリケーションの動作に変更があると、キャッシュされた接続は再利用されません。たとえば、キャッシュされた接続は、次の場合に再利用されません。

- 接続が存在しないときに自動的に停止するデータベースの場合。
- 接続が無効になっている場合。
- データベースサーバが接続制限値に達した場合。
- パスワードが変更された場合。
- login_mode オプションが設定されている場合。

アプリケーションに対して接続プーリングを透過的にするため、接続のキャッシュ時に障害が発生した場合には接続が切断されます。キャッシュされた接続を再利用しようとするときに障害が発生すると、データベースサーバで通常の接続が行われます。

接続が切断され、CPOOL 接続パラメータで指定された最大接続数に達していない場合は、接続がキャッシュされ、接続が再初期化されます。接続はアプリケーションによって切断されますが、キャッシュされた接続はデータベースサーバに接続したままになります。接続のクリーンアップと再初期化には、次の処理が含まれます。

- 未処理のすべてのトランザクションのロールバック。
- テンポラリテーブル、テンポラリ関数、変数の削除。
- 接続オプションと接続カウンターのリセット。
- データベースサーバの接続カウントの増分と減分。キャッシュされた接続しかないデータベースサーバが停止した場合、アクティブな接続があることは通知されません。
- 定義済みのすべての切断イベントと接続イベントの実行。
- login_procedure データベースオプションの実行とログインポリシーの検証。
- 接続 ID のリセット。

他の接続プール製品での接続プールの使用

接続プールをサポートしている製品または API を使用している場合、その製品または API の接続プールは SAP Sybase IQ の接続プールよりも優先されます。どちらのタイプの接続プールも、同時にアクティブにすることができます。

使用中の製品または API の接続プールの動作は、SAP Sybase IQ の接続プールと大幅に異なる場合があります。使用中の製品または API の接続プールの動作がアプリケーションに適していない場合は、SAP Sybase IQ の接続プールを使用して、アプリケーションのパフォーマンスを向上させることができます。

接続プールと読み込み専用スケールアウト

接続に対して NodeType (NODE) 接続パラメータも指定されている場合、通常、アプリケーションはプライマリサーバに接続し、プライマリサーバによって負荷が最も低いコピーノードが決まります。その時点で、そのノードに接続がリダイレクトされます。このような接続と切断をアプリケーションで短時間に何度も行うと、接続がプールされ、どのコピーノードを使用するかがプライマリサーバに問われなくなります。この動作により、プライマリサーバでの負荷は削減されますが、予期する動作にならない可能性があります。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

テンポラリ接続

テンポラリ接続は、バックアップの実行やデータベースの初期化といった操作を実行します。

sa_conn_info システムプロシージャまたは **sa_conn_list** システムプロシージャを使用して、テンポラリ接続に関する情報を取得できます。ParentConnection プロパティでは、テンポラリ接続を作成した接続の接続 ID が返されます。

テンポラリ接続には、10 億(1000000000)より大きい接続 ID と接続の機能を説明する名前があります。

次の例では、**sa_conn_info** システムプロシージャを使用して、テンポラリ接続を作成した接続を示す結果セットが返されます。

```
SELECT Number, Name, ParentConnection FROM sa_conn_info();
```

接続 8 では、**CREATE DATABASE** 文を実行するテンポラリ接続が作成されます。

Number	Name	ParentConnection
1000000048	INT: CreateDB	8
9	SQL_DBC_14675af8	(NULL)
8	SQL_DBA_152d5ac0	(NULL)

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- 論理サーバの接続 (81 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

論理サーバの接続

マルチプレックス環境では、論理サーバを使用してのみサーバにアクセスできません。

障害またはマルチプレックス環境からのメンバノードの除外などの理由により、メンバノードのすべてが常にその論理サーバで利用できるわけではありません。特定のタイミングでは、有効な論理サーバ設定は、論理サーバで実際に利用できるすべてのメンバノードで構成される論理サーバの現在の動的な構成を表します。有効な論理サーバ設定は、実質的に静的論理サーバ設定の機能で、マルチプレックス環境の動的な状態を表します。

注意： マルチプレックス設定のみが、論理サーバをサポートします。組み込みの論理サーバと論理サーバポリシーについての情報は、シンプレックス環境のカタログに保持できますが、使用されません。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 接続を終了する方法 (82 ページ)
- 接続ログ (83 ページ)

シンプレックスでの接続

シンプレックス環境では、接続は、論理サーバ割り当てのログインポリシー設定に影響されません。

接続には、論理サーバコンテキストはありません。ログインポリシーオプション 'locked' の基本設定が、サーバが接続を受け付ける前に適用されます。

マルチプレックスでの接続

マルチプレックスでは、ログインポリシーによって論理サーバへのアクセスが管理されます。1つのログインポリシーのすべてのユーザは、割り当てられた論理サーバの有効なメンバであるマルチプレックスサーバのみにアクセスできます。

マルチプレックスサーバに接続すると、接続の論理サーバコンテキストが、ユーザのログインポリシーの有効な論理サーバ割り当てとカレントノードに基づいて決定されます。

接続の論理サーバコンテキスト

ユーザ接続が確立すると、ユーザのログインポリシーと現在のノードによって、接続の論理サーバコンテキストが決定されます。

- ユーザのログインポリシーの有効な論理サーバ割り当てが1つの以上の論理サーバの場合、接続の論理サーバコンテキストは、指定された論理サーバのいずれかにおける、現在のノードの明確に定義されたメンバシップに基づいて決定されます。

注意： 指定したログインポリシーがアクセスを許可する論理サーバには、重複するメンバシップはありません。

- 現在のノードがユーザのログインポリシーに割り当てられたいずれの論理サーバのメンバでない場合、接続は失敗します。
- ユーザのログインポリシーに対する有効な論理サーバ割り当てがNONEに設定されている場合も、接続は失敗します。

接続を終了する方法

特定の状況下では、接続を切断する必要があります。

接続は次の場合に切断されます。

- Interactive SQL または Embedded SQL では、ユーザまたはアプリケーションが、現在の接続、指定の接続、またはアプリケーションのすべての接続に対して明示的な DISCONNECT 文を発行した場合。
- Interactive SQL では、ユーザが [SQL] - [切断] を選択した場合。
- Interactive SQL など、アクティブな接続を持つアプリケーションが停止した場合。

また、Interactive SQL または Embedded SQL では、DROP CONNECTION システム権限のあるユーザが DROP CONNECTION 文を発行して特定の接続を停止することもできます。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)
- Interactive SQL 接続 (65 ページ)
- ファイルデータソース (67 ページ)
- 他のデータベースからの接続 (69 ページ)
- 接続をテストする方法 (70 ページ)
- 統合化ログイン (71 ページ)
- 接続プール (77 ページ)
- テンポラリ接続 (80 ページ)
- 論理サーバの接続 (81 ページ)
- 接続ログ (83 ページ)

接続ログ

デフォルトでは、ユーザがデータベースへの接続を確立または切断するたびに、その操作が <dbname>.iqmsg によってログに記録されます。

LOG_CONNECT データベースオプションを使用すると、ユーザの接続と切断に関するロギングを制御できます。ユーザの接続時には接続ログが無効で、ユーザの切断時には接続ログが有効に設定されていた場合は、そのユーザが接続していないのに切断を行ったことがメッセージログに記録されます。

詳細については、『リファレンス：文とオプション』の「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のリスト」>「LOG_CONNECT オプション」を参照してください。

参照：

- 接続ロードマップ (47 ページ)
- 接続方法 (48 ページ)
- サポートされている接続インタフェース (60 ページ)
- 接続ステータス (61 ページ)
- 接続パラメータの構造 (62 ページ)
- ODBC データソースの接続パラメータ (63 ページ)

- [Interactive SQL 接続 \(65 ページ\)](#)
- [ファイルデータソース \(67 ページ\)](#)
- [他のデータベースからの接続 \(69 ページ\)](#)
- [接続をテストする方法 \(70 ページ\)](#)
- [統合化ログイン \(71 ページ\)](#)
- [接続プール \(77 ページ\)](#)
- [テンポラリ接続 \(80 ページ\)](#)
- [論理サーバの接続 \(81 ページ\)](#)
- [接続を終了する方法 \(82 ページ\)](#)
- [LOG_CONNECT オプション \(440 ページ\)](#)

SAP Sybase IQ データベースの作成と管理

SAP Sybase IQ データベースを作成および管理するには、Interactive SQL または SAP Control Center を使用します。

SAP Sybase IQ データベースの作成

データベースを定義するには、SQL 文またはデータベース設計ツールを使用します。

アプリケーション設計システムには、Sybase PowerDesigner® のようにデータベースオブジェクトの作成機能を備えたものがあります。これらのツールは SQL 文を作成し、それを通常 ODBC インタフェースを通してサーバに送ります。このようなツールを使えば、ユーザがテーブルの作成やパーミッションの割り当てなどを行う SQL 文を作成する必要はありません。

Sybase PowerDesigner などのデータベース設計ツールは、すぐれた設計のデータベースを開発するためのより堅牢で信頼性の高い方法を提供しています。

SAP Control Center でのデータベース作成

SAP Control Center で SAP Sybase IQ を作成するには、次の手順に従います。SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

SQL でのデータベース作成

dbisql などの対話型 SQL ツールからデータベースを作成する場合、SQL 文を使用してデータベースオブジェクトを直接に定義できます。

アプリケーション設計ツールを使っている場合でも、設計ツールでサポートされない機能をデータベースに追加するには SQL 文を使うことになります。

データベースが作成されたら、それに接続して、データベースに必要なテーブルやその他のオブジェクトを作成できます。

データベースを作成する前に

SQL 文を使用してデータベースを作成する前に、前提条件のアクションを実行します。

SQL 文を使用してデータベースを作成するには、次の操作を行います。

- データベースサーバの起動
- **dbisql** の起動

dbisql でデータベースを作成するには、既存のデータベースに接続するか、ユーティリティデータベースを起動します。ユーティリティデータベースは、データベースファイルもデータも持たない幻データベースです。

注意：サーバの起動時に **-m** サーバオプションが設定されている場合は、データベースを作成できません。

ユーティリティデータベースの起動

まだデータベースが作成されていない場合は、新しいデータベースを作成する前にユーティリティデータベースを起動します。

ユーティリティデータベースは次の方法で起動できます。

1. 起動コマンドに **-n enginename** だけを指定して、データベースのないデータベースサーバを起動します。
2. コマンドラインから **dbisql** を起動します。このとき、次のように接続文字列内で Database Name を **utility_db** に設定します。

```
dbisql -c "uid=dba;pwd=sql;eng=myserver;dbn=utility_db;..."
```

(utility_db にはデータベースファイルがないため、これを Database File で指定してはいけません)

自動的に作成される DB 領域

データベースの作成時に、データベースサーバは特定の DB 領域を自動作成します。

DB 領域名	目的	デフォルトのオペレーティングシステムファイル名
IQ_SYSTEM_MAIN	メイン (永久) IQ ストアファイル	<i>dbname.iq</i>
IQ_SYSTEM_MSG	メッセージログファイル	<i>dbname.iqmsg</i>
IQ_SYSTEM_TEMP	テンポラリ IQ ストアファイル	<i>dbname.iqtmp</i>
IQ_SHARED_TEMP	テンポラリ IQ ストア	最初、この DB 領域にはファイルが含まれていません。
SYSTEM	カタログストアファイル	<i>dbname.db</i>

SYSTEM DB 領域には、データベース作成時のスキーマ定義を保持するシステムテーブルが格納されています。また、カタログストア用に、個別のチェックポイ

ントログ、ロールバックログ、およびオプションとしてライトファイル、トランザクションログ、トランザクションログミラーも保持します。

注意：これらのデータベースファイルに加えて、データベースサーバは、セッション時に必要な情報をテンポラリファイルに保持します。このテンポラリファイルは、IQ テンポラリストアとは別のものであり、データベースサーバが停止すると不要になります。サーバが生成した名前に拡張子 `.tmp` を付けたものが、テンポラリファイル名です。テンポラリファイルのロケーションは `TEMP` 環境変数 (UNIX の場合は、コーディネータ環境変数) により決定されます。

データベースファイルの配置

データベースを作成するときに、そのデータベースを移動する必要が出てくるかどうか検討します。DB 領域のファイルのロケーションは、パフォーマンスに影響することがあります。

IQ カタログファイル (`.db`) とトランザクションログファイル (`.log`) は安全に移動できます。実行中のデータベースをコピーすることは絶対に避けてください。相対パス名を使用してデータベースを作成した場合は、サーバを停止してオペレーティングシステムのファイルコピーコマンドを使用することで、これらのファイルを移動できます。絶対パス名 (完全修飾パス名) を使用してデータベースを作成した場合、ファイルを移動するには、**BACKUP DATABASE** コマンドを使用してフルバックアップを行った後、**BACKUP DATABASE** コマンドで **RENAME** オプションを指定してバックアップをリストアする必要があります。

データベースが停止しているときは、ローパーティション上の IQ DB 領域を別のパーティションに移動できます。移動先のパーティションは、少なくとも現在の DB 領域と同じサイズでなければなりません。また、移動先のパーティションは、DB 領域を開始するために同じパスを持っていないければなりません。

dd などのバイトコピーユーティリティを使用できるのは、ローデータをファイルシステムデバイスまたは別のローデバイスにコピーする場合のみです。dd を使用して DB ファイルをシステムからローデバイスに移動しないでください。

警告！ DB 領域 (システム、IQ メイン、IQ テンポラリ) 用のシステムファイルを割り付けるときは、そのファイルを、ローカルエリアネットワーク上で共有されているファイルシステムに配置しないでください。これを行うと、I/O パフォーマンスが低下することがあります。また、ローカルエリアネットワークが過負荷になったり、DB 領域ファイルに問題が発生したりするなど、他の問題が起きるおそれがあります。UNIX プラットフォームと Linux プラットフォームでは、Network File System (NFS) にマウントされたファイルシステムを避けてください。Windows プラットフォームでは、別のノードが所有するネットワークドライブに DB 領域のファイルを配置しないでください。ファイルの配置に関するこれらの推奨事項は、ログファイルにも適用されます。

競合を避けるため、1つの接続上の1人のデータベース管理者がすべてのDB領域を管理するようにしてください。

システムDB領域ファイル、IQメインDB領域ファイル、IQテンポラリDB領域ファイルなどランダムアクセスファイル専用のディスクドライブ数を増やすことによって、ランダムアクセスファイルに関連するパフォーマンスを向上させることができます。トランザクションログファイルやメッセージログファイルなど順次アクセスファイルに関連するパフォーマンスは、専用ディスクドライブに格納することによって向上させることができます。

ファイル配置がパフォーマンスに及ぼす影響を抑制するために、次のことをおすすめます。

- ランダムディスク I/O を順次ディスク I/O から分離する。
- データベースファイル、テンポラリ DB 領域、トランザクションログファイルをデータベースサーバと同じ物理マシン上に配置する。
- Adaptive Server などの他のデータベースのプロキシテーブルの I/O から SAP Sybase IQ データベース I/O を分離する。
- トランザクションログとメッセージログを、IQ ストア、カタログストア、テンポラリストア、Adaptive Server などのプロキシデータベースとは別のディスクに配置する。

IQ 要件が大きくて複雑なため複数の物理システムが必要な場合は、SAP Sybase IQ のマルチプレックス機能の使用を検討します。

ローデバイスのパーミッション

データベースを作成したり、DB領域を追加したりする前に、すべてのローデバイスが読み取りと書き込みのパーミッションを持っており、`/dev/rawctl` が読み取りのパーミッションを持っていることを確認してください。Linux のローデバイス名には、`/dev/raw/rdevname` のフォーマットを使用します。たとえば `/dev/raw/raw10` のようになります。

データベースファイルの互換性

SAP Sybase IQ サーバでは、SAP Sybase IQ 12.7 より前のバージョンで作成されたデータベースを管理できません。同様に、古いサーバでは新しいデータベースを管理できません。

CREATE DATABASE 文のデフォルト

CREATE DATABASE 文には、2つの必須パラメータと複数のオプションパラメータがあります。

カタログストアのファイル名と **IQ PATH** を指定する必要があります。他のパラメータはすべてオプションです。

すべてデフォルトを使用すると、データベースの特性は次のようになります。

- 大文字と小文字が区別されます (**CASE RESPECT**)。つまり、"ABC" と "abc" の比較は **NOTEQUAL** になります。デフォルトのログインのユーザ ID は DBA (大文字) で、パスワードは sql (小文字) です。デフォルトでは、パスワードは大文字と小文字を区別します。ユーザ名は常に、大文字と小文字を区別しません。
- カタログページサイズは 4096 バイトです (**PAGE SIZE 4096**)。
- 長さの異なる 2 つの文字列を比較する場合、IQ は短い方の文字列を長い方の文字列と同じ長さになるまで空白を埋め込んだものとして扱います。つまり 'abc' と 'abc' は等しいとされます (**BLANK PADDING ON**)。
- Adaptive Server と互換性がありません。
- IQ ページサイズは 128KB です (**IQ PAGE SIZE 131072**)。
- IQ メッセージファイルと IQ テンポラリストアは、カタログストアと同じディレクトリにあります。
- ローデバースの場合、**IQ SIZE** と **TEMPORARY SIZE** はローパーティションの最大サイズになります。オペレーティングシステムファイルの場合は、後述のこのパラメータの説明を参照してください。
- IQ テンポラリストアサイズは IQ サイズの半分になります。
- jConnect JDBC ドライバが有効になります (**JCONNECT ON**)。
- 照合 ISO_BINENG が使用されます。照合順序は ASCII 文字セットの文字順序と同じです。大文字と小文字を区別するデータベースでは、すべての大文字がすべての小文字に先行します (たとえば、'A' も 'B' も 'a' に先行します)。
- **IQ RESERVE** と **TEMPORARY RESERVE** は 0 になります。
- **SYSTEM PROCEDURE AS DEFINER** が OFF です。

パスワードのセキュリティの強化

パスワードは、データベースのセキュリティシステムの重要な部分です。パスワードのセキュリティを強化するには、いくつかのオプションがあります。

- **ログインポリシーの実装** – パスワードの変更頻度を制御し、アカウントがロックされるまでに実行可能なログイン試行回数を指定するか、パスワードの有効期限切れを強制します。「ログインポリシー」を参照してください。
- **パスワードの最小長の実装** – デフォルトでは、パスワードは任意の長さで指定できます。セキュリティを強化するために、すべての新しいパスワードに必要な最小長を設定して、短い(したがって、推測しやすい)パスワードを禁止することができます。推奨される最小長は 6 です。「MIN_PASSWORD_LENGTH」を参照してください。
- **パスワードルールの実装** – パスワードにおける特定の種類の文字の要求、パスワードの再使用の禁止、パスワードの有効期限の適用などを含む詳細なパスワードルールを実装します。ルールの検証は、新しいユーザ ID の作成時、またはパスワードの変更時に行われます。「VERIFY_PASSWORD_FUNCTION」を参照してください。

参照：

- MIN_PASSWORD_LENGTH オプション (441 ページ)
- VERIFY_PASSWORD_FUNCTION オプション (442 ページ)

相対パス名

データベースの各ファイルに相対パス名または完全修飾パス名を使用して、データベースを作成できます。

相対パス名を使用してデータベースを作成してください。絶対パス名を指定した場合は、データベースをバックアップしてリストアしないかぎり、ファイルを別のパス名に移動できなくなります。

データベースが UNIX 上にある場合は、各パス名のシンボリックリンクを定義できます。

ディレクトリパスを省略すると、SAP Sybase IQ では、次のようにファイルが配置されます。

- カタログストアは、サーバの作業ディレクトリを基準とする相対位置に作成されます。
- IQ ストア、テンポラリストア、メッセージログのファイルは、カタログストアと同じディレクトリ内に作成されるか、カタログストアを基準とする相対位置に作成されます。
- トランザクションログは、カタログストアと同じディレクトリ内に作成されます (ファイル名を指定しない場合にも作成されます)。ただし、同じ物理マシン上の、カタログストアと IQ ストアとは別の物理デバイス上に置く必要があります。

注意： 相対パス名を使用してデータベースを作成した場合、データベースが配置されているディレクトリからデータベースサーバを起動する必要があります。設定ファイルを使用してサーバを起動すると、一貫したロケーションからサーバを起動できます。

IQ PATH パラメータのガイドライン

IQ PATH パラメータは必須です。このパラメータは、SQL Anywhere データベースではなく、SAP Sybase IQ データベースを作成することを SAP Sybase IQ に指示します。

このパラメータ内に IQ ストアのロケーションを指定します。

データベースのロケーションは慎重に選択してください。SAP Sybase IQ データベースやそのファイルを別のロケーションに移動することはできますが、そのためにはデータベースを停止し、場合によってはバックアップとリストアを実行する必要があります。

別のドライブ上に領域を追加することはできますが、この追加領域には新しいデータしか格納できません。特定のインデックス、テーブル、データのローを、あるロケーションから別のロケーションへと簡単には移動できません。

各オペレーティングシステムには、ローデバイス名に関する独自のフォーマットがあります。Sun Solaris でのローデバイスの初期化に関する重要な注意事項については、『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』の「物理的制限」を参照してください。

表 2 : UNIX 上でのローデバイス名

	例
AIX	/dev/rraw121v
HP-UX	/dev/vg03/rrchee12g
Sun Solaris	/dev/rsd0c
Sun AMD	/dev/rdsk/c5t0d0s1

表 3 : Windows 上でのローデバイス名

デバイスタイプ	要求される名前フォーマット	SQL の例
パーティションあり	そのパーティションに割り当てられた文字	¥¥¥¥.¥¥C:
パーティションなし	<i>PhysicalDriveN</i> : <i>N</i> は0から始まる数値で、必要なだけ大きくできる。物理ドライブ番号を調べるには、[管理ツール]の[ディスク アドミニストレータ]を実行する。	¥¥¥¥.¥¥ PhysicalDrive32

Windows システムでは、円記号を含むデバイス名を指定するときに、システムが円記号と文字の組み合わせをタブや改行コマンドなどのエスケープシーケンスと間違えるのを防ぐために、円記号を二重にする必要があります。

Windows で、ローデバイス名を SQL 文で指定するときには、円記号を必ず二重にする必要があります。

例 1

次の文は company.db という名前のデータベースを作成します。このデータベースは、次の 4 つの Windows ファイルから構成されます。

- カタログストアは、company.db にあります。これは、サーバが起動したディレクトリにあります (この場合は c:¥company)。

- IQ ストアは、`c:¥company¥iqdata¥company.iq` にあります。
- テンポラリストアは、`c:¥company¥company.iqtmp` にあります。
- IQ メッセージログファイルは、`c:¥company¥company.iqmsg` にあります。

```
CREATE DATABASE 'company.db'  
IQ SIZE 200  
IQ PATH 'c:¥¥company¥¥iqdata¥¥company.iq'
```

例 2

次の文は `company.db` という名前のデータベースを作成します。このデータベースは次の4つの UNIX ファイルから構成されます。

- カタログストアは、`company.db` にあります。これは、サーバが起動したディレクトリにあります (この場合は `/disk1/company`)。
- IQ ストアは、`/disk1/company/iqdata/company.iq` にあります。
- テンポラリストアは、`/disk1/company/company.iqtmp` にあります。
- IQ メッセージログファイルは、`/disk1/company/company.iqmsg` にあります。

```
CREATE DATABASE 'company.db'  
IQ SIZE 2000  
IQ PATH '/disk1/company/iqdata/company.iq'
```

例 3

次に示す UNIX の例では、IQ PATH で指定したローパーティションに `company` という名前のデータベースを作成しています。

```
CREATE DATABASE 'company'  
IQ PATH '/dev/rdisk/c0t0d0s0'
```

例 4

次に示す Windows の例では、IQ PATH で指定したローパーティションに `company` という名前のデータベースを作成しています。

```
CREATE DATABASE 'company'  
IQ PATH '¥¥¥¥.¥¥D:'
```

IQ PAGE SIZE パラメータのガイドライン

IQ PAGE SIZE パラメータは、メモリとディスクの使用量を決定します。

IQ ストアのページサイズを設定するには、**IQ PAGE SIZE** オプションを使用します。**IQ PAGE SIZE** は 65536 ~ 524288 バイトの 2 の累乗です。IQ ストア内のすべての DB 領域で、IQ ページサイズは同じとなります。

パフォーマンスを最大化するには、次の最小 IQ ページサイズを使用してください。

- 最も大きいテーブルのロー数が 10 億以下のデータベース、または合計サイズが 8TB 未満のデータベースの場合は、64KB (**IQ PAGE SIZE 65536**)。これは新規データベースの最小サイズです。32 ビットプラットフォームでは、IQ ページサイズを 64KB にすると最高のパフォーマンスが得られます。
- 64 ビットプラットフォームで、最も大きいテーブルのロー数が 10 億～40 億のデータベース、または合計サイズが 8TB 以上になることが予想されるデータベースの場合は、128KB (**IQ PAGE SIZE 131072**) です。128KB はデフォルトの IQ ページサイズです。
- 64 ビットプラットフォームで、最も大きいテーブルのロー数が 40 億より多いデータベース、または合計サイズが 8TB 以上になることが予想されるデータベースの場合は、256KB (**IQ PAGE SIZE 262144**)。

マルチユーザ環境とメモリ制約のあるシステムでは、IQ ページサイズを最小の 64KB にすると、ページングを最小減に抑えることができます。

SAP Sybase IQ はデータを圧縮してディスクに格納します。データを処理するときは、データを解凍してデータページをメモリに移動します。IQ ページサイズから、IQ ストアのディスク圧縮量とデフォルト I/O 転送ブロックサイズが決まります。ほとんどのアプリケーションでは、デフォルト値を使用するのが最適です。これらの設定、およびリソース使用量とパフォーマンスに影響を与えるその他のオプションについては、『パフォーマンス&チューニングガイド』の「システムリソースの管理」を参照してください。

ワイドデータのページサイズ

データベース内にワイドテーブルが含まれている場合は、現在のロー数に適したものよりも 1 つ上のランクの IQ ページサイズを使用するとパフォーマンスが良くなります。たとえば、CHAR または VARCHAR のワイドカラム (255～32,767 バイトのカラム) を複数含むテーブルでは、通常よりも大きな IQ ページサイズが必要になります。

IQ ではデータをカラムに格納するため、ローの最大長という概念はありません。しかし、実質的な限界は IQ ページサイズの半分になります。これが、クエリがクライアントに返すことのできる最も広い結果セットだからです。少なくとも、最も広いテーブルの幅の 2 倍ある IQ ページサイズを選択してください。

データベースサイズのガイドライン

データベースの作成時に、初期 IQ データベースファイル (IQ_SYSTEM_MAIN DB 領域) のサイズと予約サイズを MB 単位で設定します。

これらの値は、メインストアについては **IQ SIZE** および **IQ RESERVE** パラメータによって、テナポラリストアについては **TEMPORARY SIZE** および **TEMPORARY RESERVE** パラメータによって決まります。

- ローパーティションについては、**IQ SIZE** と **TEMPORARY SIZE** を指定する必要はありません。SAP Sybase IQ がローデバイスのサイズを決定し、**IQ SIZE** と **TEMPORARY SIZE** を自動的に設定します。サイズを明示的に指定する場合は、実際のローパーティションサイズよりも大きな値は指定できません。
- オペレーティングシステムファイルの場合は、デフォルトを使用するか、データのサイズに基づいて、必要最小値から最大 4TB までの値を 1MB 刻みで指定します。

IQ ストアとテンポラリストアのデフォルトと最小サイズ

IQ RESERVE パラメータと **TEMPORARY RESERVE** パラメータは一連のブロックを予約するため、DB 領域のサイズを後から変更できます。ただし、**IQ RESERVE** を必要以上に大きくすると追加のディスク領域を消費することになります。

表 4 : IQ ストアとテンポラリストアのデフォルトと最小サイズ

IQ ページサイズ	IQ ストアのデフォルトサイズ	テンポラリストアのデフォルトサイズ	明示的に指定する場合の IQ ストアの最小サイズ	明示的に指定する場合のテンポラリストアの最小サイズ
65536	4096000	2048000	4MB	2MB
131072	8192000	4096000	8MB	4MB
262144	16384000	8192000	16MB	8MB
524288	32768000	16384000	32MB	16MB

PAGE SIZE パラメータのガイドライン

カタログストアのページサイズは、**CREATE DATABASE PAGE SIZE** オプションで設定します。このオプションのデフォルト値かつ最小値は 4096 (4KB) です。

例

次の文では、**PAGE SIZE** が 4096 バイト (4KB) のカタログを備えたデータベースを作成します。このデータベースの IQ ストアは UNIX ローパーティション上にあり、**IQ PAGE SIZE** は 131072 バイト (128KB) です。デフォルトでは、IQ ストアサイズはローパーティションのサイズに等しくなり、テンポラリストアはその半分になります。ここではテンポラリストアのパスを指定していないため、テンポラリストアはカタログストアと同じディレクトリに作成されます。

```
CREATE DATABASE 'company'
IQ PATH '/dev/rdisk/c2t6d0s3'
PAGE SIZE 4096
IQ PAGE SIZE 131072
```


ブロックサイズのガイドライン

ほとんどの場合は、IQ ページサイズに基づくデフォルトのブロックサイズをそのまま使用します。

IQ メインストアと IQ テンポラリストアの領域管理

MAIN_RESERVED_DBSPACE_MB オプションと **TEMP_RESERVED_DBSPACE_MB** オプションを設定すると、チェックポイント操作、コミット操作、セーブポイントの解放操作に余裕ができます。

これらのオプションでは、**IQ_SYSTEM_MAIN** または **IQ_SYSTEM_TEMP** のそれぞれで、最後の読み込み／書き込み DB 領域ファイルの予約領域の割り付けサイズを決定します。

SET ANY PUBLIC OPTION システム権限を持つユーザは、接続ごとに使用される領域の量を制限できます。また、SAP Sybase IQ の IQ メインストアまたは IQ テンポラリストアで領域が不足した場合は、新しい領域が追加されるまで、サーバが領域不足のトランザクションをサスペンドすることはなくなりました。IQ メインストアまたは IQ テンポラリストアで領域が不足しているトランザクションは失敗し、ロールバックされます。

データベースオプション **MAX_TEMP_SPACE_PER_CONNECTION** は、接続ごとに使用される IQ テンポラリストアの領域量を制限し、クエリに加え、すべてのデータ操作言語 (DML) 文のテンポラリストアの使用率も追跡します。

MAX_TEMP_SPACE_PER_CONNECTION は、文によるテンポラリストアの実際の実行時の使用率をモニタして制限します。接続が **MAX_TEMP_SPACE_PER_CONNECTION** オプションで設定された割り当てを超えた場合は、エラーが返され、現在の文はロールバックされます。

QUERY_TEMP_SPACE_LIMIT データベースオプションのデフォルト値は 0 です。これは、クエリによるテンポラリストアの使用率が制限されていないことを意味します。接続ごとのテンポラリストアの使用率を制限するには、DBA は、クエリを含むすべての DML 文に **MAX_TEMP_SPACE_PER_CONNECTION** オプションを設定します。

SAP Sybase IQ データベースをバージョン 15.0 より前のリリースからアップグレードすると、**MAX_TEMP_SPACE_PER_CONNECTION** データベースオプションはデフォルト値の 0 に設定されます。**sp_iqcheckoptions** を使用すると、アップグレードの前と後のオプションのデフォルト値と現在の値を見つけることができるため、新しいオプション設定がアップグレードしたデータベースに適しているかどうかを判断できます。

データベースの削除

データベースを削除すると、データベースへの変更を記録するトランザクションログを含むすべてのテーブルとデータが、ディスクから削除されます。そのデータベースに関連付けられたすべての DB 領域も削除されます。

データベースを削除するには、次の SQL 文を使用します。

```
DROP DATABASE dbname
```

データベースの作成時に指定したものとまったく同じデータベース名とパス名を指定する必要があります。

たとえば、Windows システムでは次のように指定します。

```
DROP DATABASE 'c:¥sybase¥data¥mydb.db'
```

データベースは、停止してから削除してください。接続パラメータ **AUTOSTOP=no** が使用されている場合、**STOP DATABASE** 文の実行が必要になる場合があります。

データベースから他のユーザを切断する

SAP Sybase IQ では、指定したデータベースから他のユーザを切断できます。

connection_property 関数を使用して接続番号を要求すると、ユーザの *connection-id* を取得できます。次の文は、現在の接続の接続 ID を返します。

```
SELECT connection_property( 'number' )
```

1. **DROP CONNECTION** システム権限に関する管理権限を持つユーザとして既存のデータベースに接続します。
2. Interactive SQL を使用して、**DROP CONNECTION** 文を実行します。

次の文は、ID 番号 4 の接続を削除します。

```
DROP CONNECTION 4
```

Embedded SQL でデータベースから切断する

データベースでの操作が完了したら、指定した接続またはすべての接続を切断できます。

EXEC SQL DISCONNECT 文を実行します。

次の文は、Embedded SQL 内での DISCONNECT の使用法を示します。

```
EXEC SQL DISCONNECT :conn_name
```

Interactive SQL でデータベースからすべての接続を切断する

データベースでの操作が完了したら、指定した接続またはすべての接続を切断できます。

DISCONNECT 文を実行します。

次の文は、Interactive SQL から **DISCONNECT** を使用し、すべての接続を切断する方法を示します。

```
DISCONNECT ALL
```

Interactive SQL でのシステムオブジェクトの表示

Interactive SQL では、システムテーブルにクエリを実行することはできませんが、システムビューの内容は参照できます。

ほとんどのシステムテーブルには、クエリを実行できる同じシステムビューがあります。

データベース内のテーブル、ビュー、ストアードプロシージャ、またはドメインは、システムオブジェクトです。システムテーブルはデータベースそのものに関する情報を格納し、システムプロシージャとシステムドメインは Sybase Transact-SQL 互換性をほぼサポートしています。

1. Interactive SQL を使用してデータベースに接続します。
2. 検索するテーブルのシステムビューを指定して、**SELECT** 文を実行します。

ISYSTAB システムテーブルを検索するには、ビュー SYS.SYSTAB の内容を [結果] ウィンドウ枠に表示します。

```
SELECT *  
FROM SYS.SYSTAB
```

Interactive SQL でのデータベースオプションの設定

データベースオプションとは、データベースの動作方法や実行方法を変更するための設定オプションです。

SET OPTION 文の中でプロパティを指定します。

注意： データベースそのもののオプションを設定すると、実際には、そのデータベース内の PUBLIC ロールのオプションを設定することになります。すべてのユーザとロールは PUBLIC のオプション設定を継承するからです。

データ記憶領域の管理

SAP Sybase IQ がディスク領域を割り付ける方法を指定し、最適なパフォーマンスを得られるようにデータをディスク上に編成します。

領域の割り付け

すべての SAP Sybase IQ データベースは、ファイルシステムとローパーティションのどちらに格納されている場合でも、事前に割り付けられます。

各データベースは複数のテーブル領域を含みます。テーブル領域とは、記憶領域全体の論理的なサブセットとして管理できるデータベース内の記憶単位です。個々のオブジェクトとサブオブジェクトを個々のテーブル領域に割り付けることができます。

「DB 領域」は、1 つ以上のオペレーティングシステムファイルで構成されるテーブル領域です。

ストアは、永続的データまたはテンポラリデータを特別な目的のために格納する 1 つまたは複数の DB 領域です。SAP Sybase IQ には、次の 5 個のストアがあります。

- カタログストア。システム DB 領域と、そのほかにユーザ定義のカタログ DB 領域が含まれます。
- IQ メインストア。IQ_SYSTEM_MAIN DB 領域とその他のユーザ DB 領域があります。
- IQ テンポラリストア。IQ_SYSTEM_TEMP DB 領域があります。
- IQ 共有テンポラリストア。IQ_SHARED_TEMP DB 領域があります。
- ローレベルのバージョン管理 (RLV) ストア。高パフォーマンスのローレベル更新のためのインメモリストアです。

DB 領域のタイプ

各タイプの DB 領域には、特定のタイプの SAP Sybase IQ データが格納されます。

DB 領域のタイプ	格納されるデータ	DB 領域に含まれるファイル	DB 領域の数
SYSTEM DB 領域	システムテーブル、ビュー、ストアドプロシージャ、SQL Anywhere テーブル、および関数の定義。	1 個	1 個または複数
他のカタログ DB 領域	SQL Anywhere テーブル。	1 個	1 個または複数
IQ_SYSTEM_MAIN	コミットした各トランザクションとチェックポイントしたアクティブな各トランザクションのロールフォワード/ロールバックデータ、インクリメンタルバックアップメタデータ、データベース領域、identity メタデータなどのデータベース構造体。ユーザオブジェクトをここに格納することもできるが、他のメイン DB 領域に格納することが推奨される。	1 個または複数	1 個
他のメイン DB 領域 (ユーザ DB 領域とも呼ばれます)	テーブル、インデックス、テーブルメタデータなどのデータベースオブジェクト。	1 個または複数	1 個または複数。複数のユーザ DB 領域を作成するには、IQ_VLDBMGMT オプションのライセンスが必要。
RLV_STORE	ローレベルバージョン管理 (RLV) に対応したテーブルからのリアルタイムインメモリデータ。	1 個	1 個

DB 領域のタイプ	格納されるデータ	DB 領域に含まれるファイル	DB 領域の数
IQ_SYSTEM_TEMP	スタンドアロンデータベースまたはマルチプレックスノードに対して1つのテンポラリ DB 領域を定義する 1～n 個のテンポラリ DB 領域ファイルのセット。詳細については、次の表を参照。	1 個または複数	1 個
IQ_SHARED_TEMP	すべてのマルチプレックスノードで共有される 1つのテンポラリ DB 領域を定義する 1～n 個のテンポラリ DB 領域ファイルのセット。詳細については、次の表を参照。	1 個または複数 (最初は空)	1 個
IQ_SYSTEM_MSG	データベースアクティビティに関するメッセージをログに記録する外部ファイル。	マルチプレックスノードごとに 1 個	1 個

テーブルの DB 領域は、明示的または暗黙的に指定されています。ベーステーブルの場合は、**DEFAULT_DBSPACE** オプションの値によって DB 領域のロケーションが暗黙的に決定されます。または、ロケーションは、**CREATE TABLE IN *dbspace_name*** 句を使用して明示的に指定できます。ベーステーブルは通常、IQ メインストアの DB 領域に作成しますが、IQ インデックスを使用せずに、カタログストアの DB 領域に作成することもできます。

グローバルテンポラリテーブルの場合は、**IN SYSTEM** 句を指定して SA グローバルテンポラリテーブルを明示的に作成します。IQ テンポラリテーブルは、デフォルトで **IQ_SYSTEM_TEMP** に作成されます。

常に IQ_SYSTEM_TEMP に格納	常に IQ_SHARED_TEMP に格納	IQ_SHARED_TEMP または IQ_SYSTEM_TEMP に格納可能
サーバ障害からのリカバリのためのスクラッチ領域	分散クエリ処理のための分散テンポラリオブジェクト	IQ テンポラリユーザオブジェクト (IQ テンポラリテーブルおよびインデックスなど)
共有記憶領域管理のためのローカルビットマップ		バージョン管理されていないローカルなテンポラリオブジェクトのためのスクラッチ領域
テンポラリ記憶領域のトランザクションステータス (ロールフォワードおよびロールバックのためのセーブポイント)		

カタログストア

カタログストアは、SAP Sybase IQ データベース用のメタデータを含んでいます。

メタデータは、テーブル、カラム、インデックスのレイアウトを記述します。カタログストアは、単にカタログと呼ばれることもあります。

- **SYSTEM DB 領域 - SYSTEM** という名前のカタログ DB 領域は、データベース用のメタデータを含んでおり、SQL Anywhere リレーショナルデータベースシステムのテーブルと同じフォーマットで格納されています。SQL Anywhere は、SAP Sybase IQ の有無にかかわらず存在できます。カタログストア内で SQL Anywhere スタイルのテーブルを SAP Sybase IQ テーブルと共存させたり、SQL Anywhere データベースを別に作成したりできます。各カタログ DB 領域には、必ず 1 つのファイルが含まれています。
- その他のカタログ DB 領域 - **SYSTEM DB 領域** とは別の DB 領域に SQL Anywhere テーブルを作成できます。

カタログストアに領域を事前に割り付けるには、**ALTER DBSPACE** 文を実行します。

IQ_SYSTEM_MAIN DB 領域

IQ_SYSTEM_MAIN は特別な DB 領域で、データベースをオープンするために必要な構造体が格納されます (チェックポイントログ、コミットされた各トランザクションとチェックポイントが設定されたアクティブな各トランザクションのロールフォワード/ロールバックデータ、インクリメンタルバックアップメタデータ、データベース領域と identity メタデータ)。

IQ_SYSTEM_MAIN DB 領域は、データベースの作成時や古いデータベースの SAP Sybase IQ 16.0 へのアップグレード時に作成されます。IQ_SYSTEM_MAIN は、データベースが開いているときは常にオンラインです。

他のユーザメイン DB 領域

ユーザメイン DB 領域を作成して、IQ_SYSTEM_MAIN にユーザがユーザテーブルやインデックスを配置しないようにします。

IQ_SYSTEM_MAIN には、ユーザテーブルやインデックスを配置しないようにしてください。これらのテーブルが小さく、非常に重要なテーブルである場合には、IQ_SYSTEM_MAIN にユーザテーブルを作成することを管理者が許可できます。ただし、データベースを作成したらすぐに、管理者は 2 つ目のメイン DB 領域 (ユーザメイン DB 領域) を作成し、PUBLIC の DB 領域 IQ_SYSTEM_MAIN における CREATE 権限を取り消し、選択したユーザまたは PUBLIC に新しいメイン DB 領域の CREATE 権限を付与して、PUBLIC.DEFAULT_DBSPACE を新しいユーザメイン DB 領域に設定することをおすすめします。

次に例を示します。

```
CREATE DBSPACE user_main USING FILE user_main 'user_main1' SIZE
10000; GRANT CREATE ON user_main TO PUBLIC; REVOKE CREATE ON
IQ_SYSTEM_MAIN FROM PUBLIC; SET OPTION PUBLIC.DEFAULT_DBSPACE =
'user_main';
```

注意：複数のユーザ DB 領域を作成するには、IQ_VLDBMGMT オプションのライセンスが必要です。

IQ テンポラリ DB 領域

IQ テンポラリストア用の唯一の DB 領域である IQ_SYSTEM_TEMP は、データベースの作成時やアップグレード時に作成されます。

各 DB 領域には、任意の数のファイルを格納できます。IQ ファイルの合計数だけが 16383 個に制限されています。

IQ_SYSTEM_TEMP DB 領域の大きさは、少なくとも **start_iq** ユーティリティの **-iqt** サーバオプションで指定した値にしてください。

IQ 共有テンポラリ DB 領域

データベースを作成するか、データベースを SAP Sybase IQ 16.0 にアップグレードすると、IQ 共有テンポラリストア IQ_SHARED_TEMP 用の DB 領域が 1 つ作成されます。この DB 領域には、分散クエリ処理に参加するノード間で共有されるテンポラリ構造体が格納されます。

この DB 領域には、最初何も含まれていません。ファイルを追加するには、**ALTER DBSPACE ADD FILE** を使用します。この DB 領域へのファイルの割り付けはオプションで、マルチプレックスサーバでの分散クエリ処理でのみ必要になります。

TEMP_DATA_IN_SHARED_TEMP 論理サーバポリシーオプションを ON に設定すると、共有テンポラリストアが空でないかぎり、すべてのテンポラリテーブルデータと対象スクラッチデータが共有テンポラリストアに書き込まれます。このオプションを設定した後、または読み書きファイルを共有テンポラリストアに追加した後、セカンダリノードを再起動する必要があります。共有テンポラリストアに読み書きファイルが含まれていない場合、またはセカンダリノードを再起動しない場合、データは IQ_SYSTEM_TEMP に書き込まれます。OFF の場合、すべてのテンポラリテーブルデータとスクラッチデータはローカルのテンポラリストアに書き込まれます。

共有システムテンポラリストア

共有テンポラリストレージを使用するマルチプレックス設定では、各セカンダリサーバに対してローカルストアが個別に必要なのではなく、IQ_SHARED_TEMP DB 領域を共有システムテンポラリストアとして使用できます。共有システムテンポラリストアは、マルチプレックス設定を簡素化し、パフォーマンスを改善し、分散クエリ処理をサポートします。

マルチプレックスシステム:

- 論理サーバポリシーオプション TEMP_DATA_IN_SHARED_TEMP を ON に設定すると、SAP Sybase IQ はすべてのテンポラリオブジェクトを IQ_SHARED_TEMP DB 領域に作成します。このオプションを設定した後、または読み書きファイルを共有テンポラリストアに追加した後、セカンダリノードを再起動する必要があります。(共有テンポラリストアに読み書きファイルが含まれていない場合、またはセカンダリノードを再起動しない場合、データは IQ_SYSTEM_TEMP に書き込まれます。)
- **IN IQ_SYSTEM_TEMP** 句を使用して作成するテンポラリユーザオブジェクト(テーブルやテーブルインデックスなど)は、論理サーバオプション TEMP_DATA_IN_SHARED_TEMP の値に応じて IQ_SYSTEM_TEMP または IQ_SHARED_TEMP のいずれかに書き込まれます。
 - TEMP_DATA_IN_SHARED_TEMP が OFF である場合、オブジェクトは IQ_SYSTEM_TEMP に書き込まれます。
 - TEMP_DATA_IN_SHARED_TEMP を ON に設定した場合、オブジェクトは IQ_SHARED_TEMP に書き込まれます。

SAP Sybase IQ は、**IN IQ_SHARED_TEMP** 句を使用したテンポラリユーザオブジェクトの作成をサポートしません。

- **WITH STOP SERVER** 句は、論理サーバ内のすべてのサーバを自動的に停止します。以下の文は **WITH STOP SERVER** をサポートします。

- **ALTER LOGICAL SERVER**
- **ALTER LS POLICY**
- **CREATE LOGICAL SERVER**
- **DROP LOGICAL SERVER**
- **ALTER LS POLICY ... WITH STOP SERVER** を使用して TEMP_DATA_IN_SHARED_TEMP オプションを ON または OFF に変更すると、その論理サーバ内のすべてのサーバが自動的に停止します。論理サーバが TEMP_DATA_IN_SHARED_TEMP オプションにより指定されているストア内にテンポラリデータを配置するようにするには、これらのサーバを再起動する必要があります。
- **ALTER LS POLICY** を使用して TEMP_DATA_IN_SHARED_TEMP を OFF に設定した場合、次回、通常のサーバ起動後、論理サーバはテンポラリデータの SYSTEM テンポラリ領域への配置を開始します。
- また、**CREATE LOGICAL SERVER** 文、**ALTER LOGICAL SERVER** 文、または **DROP LOGICAL SERVER** 文で **WITH STOP SERVER** 句を使用して、間接的に TEMP_DATA_IN_SHARED_TEMP 値を変更することもできます。

参照：

- ALTER LS POLICY 文 (454 ページ)

IQ メッセージファイル DB 領域

IQ_SYSTEM_MSG は、データベース IQ メッセージログファイルのファイルパスを指すシステム DB 領域です。

IQ_SYSTEM_MSG はデータを格納しないため、ストアとはみなされません。IQ_SYSTEM_MSG DB 領域には、マルチプレックスノードごとに 1 つのファイルがあります。デフォルトでは、シンプレックスサーバまたはマルチプレックスのコーディネータ上にあるメッセージファイルの物理ファイル名は、<dbname>.iqmsg です。マルチプレックスのセカンダリノード上にある IQ メッセージファイルの物理ファイル名は、<servername>.iqmsg です。

IQ_SYSTEM_MSG は IQ ストアの DB 領域ではないため、**READONLY** や **OFFLINE** などの **ALTER** コマンドは IQ_SYSTEM_MSG に適用されません。

データベースの領域

各ストアの最初の DB 領域は、そのデータベースの作成時に自動的に作成されます。必要に応じて追加の DB 領域を作成できます。

テーブルを作成しロードすると、SAP Sybase IQ はそのストア内の、空き領域のある既存の DB 領域すべてにデータを配布します。DB 領域の作成時に、成長を見込

んで領域を多めに予約しておくことができます。DB 領域のサイズは、予約した最大領域まで拡大できます。また、DB 領域を小さくすることもできます。その場合は、DB 領域のトランケートされる部分からすべてのデータを移動しておくことが必要です。必要に応じて特定の DB 領域から個々のデータベースオブジェクトを移動させることができます。

すべてのディスク領域をデータベースに割り付けることは避けてください。ディスク領域を 10 パーセントは残すようにします。この領域は、領域不足の状態を SAP Sybase IQ で適切に処理するために必要です。

古い DB 領域がいっぱいになったら DB 領域を徐々に追加するのではなく、データベースの作成時にすべての DB 領域を作成してください。この方法であれば、DB 領域をより均等に埋めることができるため、ディスク I/O が向上します。

デバッグ用の独立したデータベースを作成します。開発作業中はサーバに障害が起きる可能性が高いため、運用データベース上での開発作業は避けてください。

IQ ストアとテンポラリストアに必要な領域

データベースに必要な領域の大きさは、データ量と、作成するインデックスの数と種類によって決まります。

データベースへのロードまたは挿入時に領域が不足すると、サーバはトランザクション全体をロールバックするか、セーブポイントにロールバックします。

SAP Sybase IQ は、明示的に定義したテンポラリテーブルに加えて、テンポラリストアを一時的な結果領域として使用します(ロード中または削除中のソート、ハッシュ、ビットマップの一時的な結果をここに格納します)。テンポラリストアに必要な領域量は、発行されたクエリのタイプ、同時使用の程度、データのサイズによって決まります。

メインストアとテンポラリストアのサイズ調整ガイドライン

データ記憶領域のガイドラインは、SAP Sybase IQ アーキテクチャの影響を受けません。

- `IQ_SYSTEM_MAIN` DB 領域は、IQ テーブルメタデータ以外のすべてのデータベースメタデータを保持します。IQ テーブルメタデータは、テーブルの DB 領域とテーブルバージョン (TLV) ログに格納されています。ノードが停止した場合、マルチプレックスは、ノードが復旧したときに同期させるため、バージョンを格納する必要があります。これらのバージョンは、大きな領域を使用する場合があります。

- IQ_SYSTEM_MAIN DB 領域の約 20 パーセントが事前割り付けされたフリーリスト領域で使用されており、ユーザデータに使用できません。
- SAP Sybase IQ のこのバージョンは、より多くの処理を並列で実行するため、以前のリリースよりも多くのテンポラリ領域を使用します。

IQ_SYSTEM_MAIN ストアに必要な領域に影響を与えるのは、次の 3 つの要因です。

- バージョン管理 — 管理されているバージョンの違い。
- データとインデックスの性質。
- データの動的性質 — 常により多くのデータをロードする能力。

マニュアルでは一般的なガイドラインを提供していますが、各データベースに必要な領域は、これらの要因の組み合わせによってさまざまです。合計サイズが 500GB より小さい開発サーバまたはレポートサーバの場合、IQ_SYSTEM_MAIN ファイルは 10 ～ 20GB で十分です。

表 5 : 運用データベースにおける IQ_SYSTEM_MAIN と IQ_SYSTEM_TEMP のサイズのガイドライン

タスク	ガイドライン	説明
空のスキーマの iqunload -n 出力からのロード、または小さいテストデータベースへのロード	メイン用 10GB、テンポラリ用 5GB	CREATE DATABASE のサイズは MB 単位である。

タスク	ガイドライン	説明
新しい運用データベースの作成	<ul style="list-style-type: none"> データベースを移行して、現在の IQ_SYSTEM_MAIN にローデバイスを使用する場合は、標準サイズの新しい未使用のローデバイスを割り付ける。 IQ_SYSTEM_MAIN の合計サイズは、データベースの合計サイズの 100 分の 1 以上にし、少なくともメイン用に 100GB、予約用に 100GB を用意する必要がある。 IQ_SYSTEM_MAIN にローディスクを使用する場合は、可能なかぎり複数のローディスクを使用する。複数のローディスクを使用すると、SAP Sybase IQ でデバイス間のデータをストライピングできるため、パフォーマンスが向上する。 運用の IQ DB 領域には、高いパフォーマンスの冗長ディスクアレイ (RAID 5 など) で実装されているフォールトトレラントなファイルシステムを使用する。シングルサーバシステムの場合、ローカルファイルシステムを使用できるが、マルチプレックスシステムではローデバイスが必要で、Storage Area Network デバイス上にあるのが理想的である。 	<p>ローデバイスを指定する場合は、ms_size を省略する。</p> <p>Windows システムでは、管理者権限を持つユーザアカウントのみがローデバイスにアクセスできる。rawaccredit ユーティリティは、現在のセッションに対するデバイスへのパーミッションを設定する。</p> <p>メイン用の予約は常に IQ_SYSTEM_MAIN サイズの 20 パーセントに設定する。メイン用の予約を設定するには、データベースオプション MAIN_RESERVED_DBSPACE_MB を使用する。</p>
マルチプレックス用のメインストアの作成	<p>シンプレックスデータベースに推奨されている領域の 2 倍、または少なくともメイン DB 領域用に 200GB、予約 DB 領域用に 200GB。</p>	

例 1

CREATE DATABASE 構文では、デフォルトのサイズ単位は GB ではなく MB です。

次の文は、100GB の IQ_SYSTEM_MAIN と 100GB の予約領域 (将来的な拡張のため) を持つデータベースを作成します。

```
CREATE DATABASE 'test.db'
IQ_PATH 'test.iq'
IQ_SIZE 100000
IQ_RESERVE 100000
TEMPORARY_PATH 'test.iqtmp'
TEMPORARY_SIZE 5000
```

例 2

MAIN_RESERVED_DBSPACE_MB を使用すると、セーブポイントの解放、コミット、およびチェックポイント操作で使用される、小さいが重要な特定のデータ構造体のために SAP Sybase IQ が IQ メインストア内に確保する領域の量を制御できます。

MAIN_RESERVED_DBSPACE_MB オプション値を IQ_SYSTEM_MAIN SIZE の 20 パーセントに設定します。たとえば、IQ_SYSTEM_MAIN が 100GB の場合は、次のように 20GB に設定します。

```
SET OPTION PUBLIC.MAIN_RESERVED_DBSPACE_MB = 20000
```

例 3

IQ_SYSTEM_MAIN のサイズは、データベースのマイグレーションコマンドで指定します。

-ms_size パラメータには、GB ではなく MB 単位で指定した値が必要です。ローデバイスを指定する場合は、ms_size を省略します。ローデバイスの場合、未使用のローパーティションを指定してください。

UNIX 上のローデバイス上に IQ_SYSTEM_MAIN を作成します。

```
iqunload -au -ms_filename /dev/rdisk/c1t0d1 -c
"UID=DBA;PWD=SQL;DBF=latest.db"
```

Windows 上のローデバイス上に IQ_SYSTEM_MAIN を作成します。

```
iqunload -au -ms_filename ¥¥¥¥.¥¥PhysicalDrive1 -c
"UID=DBA;PWD=SQL;DBF=latest.db"
```

ローデバイスへの Windows アクセスの設定

Windows システムは、ローデバイスへのアクセスを、管理者権限を持つユーザアカウントのみに制限します。

管理者権限を持たないアカウントを使用して SAP Sybase IQ サーバを実行するには、システムの再起動ごとに、そのアカウントに対する新しいデバイスアクセスパーミッションを有効にする必要があります。

rawaccedit ユーティリティは、現在のセッションに対するデバイスへのパーミッションを設定します。

書き込みサーバに対するクエリサーバへの読み書きアクセス、およびクエリサーバに対する読み込みアクセスを設定します。

1. コマンドプロンプトで次のように入力します。

rawaccedit

2. [IQ ローデバイスアクセス] ウィンドウで、アクセス権を提供するユーザとデバイスの名前を入力します。

[ユーザの名前] ボックスに移動するには Alt+N を使用し、[ローデバイス名] ボックスに移動するには Alt+D を使用できます

指定するデバイス	入力内容
分割されていないローデバイス	物理デバイス番号を入力する。 分割されていないデバイスは ¥¥.¥¥PhysicalDriveN という名前になり、N は 0 で始まる数値である。物理デバイス番号を確認するには、[アクセサリ]>[システムツール]>[システム情報]の順に選択する。
分割ローデバイス	そのパーティションに割り当てる文字を入力する。

3. [追加]をクリックします。
4. 一番上のパネルに表示されるユーザ名とデバイス名のエラーを訂正し、[ACL を更新して終了]をクリックします。

デバイスアクセスパーミッションは Windows を再起動するまで変更されません。

ローデバイスに対するアクセスパーミッションの表示

Windows ローデバイスへの現在のアクセスパーミッションを表示します。

rawaccess ユーティリティは admin ユーザとして実行する必要があります。これは、管理権限を持っていることと同じではありません。これは、[コマンドプロンプト]を右クリックして、[管理者として実行]を選択することによって、実行できます。管理者として実行している場合は、ウィンドウのタイトルバーに "Administrator:" と表示されます。

permission_types は Windows 対応の Microsoft Access Control Entry (ACE) に準拠します。詳細については、[http://msdn.microsoft.com/en-us/library/windows/desktop/aa374899\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374899(v=vs.85).aspx) を参照してください。

1. 管理者としてコマンドプロンプトウィンドウを開きます。
2. **rawaccess**、続いてローデバイス文字を入力します。

rawaccess filename

各パラメータの意味は次のとおりです。

- **filename** – ローデバイスへの Windows 汎用名前空間ファイル名パス。次の2つの形式のうちの1つ。
 - `¥¥.¥logical_drive_name`
 - `¥¥.¥physical_device_id`
- **logical_drive_name** – ディスクパーティションを表すのに使用するパーティション文字 (¥¥.¥D: など)。
- **physical_device_id** – Window デバイスマネージャにより割り当てられた名前。これらの名前は、コントロールパネルのディスクマネージャを使用して検索できます (¥¥.¥PhysicalDisk1 など)。

```

1. Allow Exec to ¥Everyone (Well Known Group)
2. Allow All to NT AUTHORITY¥SYSTEM (Well Known Group)
3. Allow All to BUILTIN¥Administrators (Alias)
4. Allow Exec to NT AUTHORITY¥RESTRICTED (Well Known Group)

```

ローデバイスへのパーミッションの設定

Windows 上でローデバイスを使用するには、事前にそのローデバイスにアクセスパーミッションを設定しておくことが必要です。

SAP Sybase IQ でローデバイスを使用する場合、SAP Sybase IQ を実行するアカウントは UAC (ユーザアカウント制御) を使用できません。

rawaccess ユーティリティは admin ユーザとして実行する必要があります。これは、管理権限を持っていることと同じではありません。これは、[コマンドプロンプト]を右クリックして、[管理者として実行]を選択することによって、実行できます。管理者として実行している場合は、ウィンドウのタイトルバーに "Administrator:" と表示されます。

デフォルトでは、Windows はローデバイスへの直接書き込みアクセスを許可しません。オペレーティングシステムによって設定される他のデフォルトエントリでマージされるので /GRANT オプションがアクセスを追加する最も簡単な方法です。

permission_types は Windows 対応の Microsoft Access Control Entry (ACE) に準拠します。 [http://msdn.microsoft.com/en-us/library/windows/desktop/aa374899\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374899(v=vs.85).aspx) を参照してください。

rawaccess は、Win32 デバイス名前空間の標準 Microsoft 汎用名前付け規則を使用します。 [http://msdn.microsoft.com/en-ca/library/windows/desktop/aa365247\(v=vs.85\).aspx](http://msdn.microsoft.com/en-ca/library/windows/desktop/aa365247(v=vs.85).aspx) を参照してください。

1. 管理者としてコマンドプロンプトウィンドウを開きます。

2. 以下を入力し、パーミッションを付与するユーザを指定します。

```
rawaccess filename [{/permission_type} trustee]
```

各パラメータの意味は次のとおりです。

- **filename** – ローデバイスへの Windows 汎用名前空間ファイル名パス。次の2つの形式のうちの1つ。
 - ¥¥.¥logical_drive_name
 - ¥¥.¥physical_device_id
 - **logical_drive_name** – ディスクパーティションを表すのに使用するパーティション文字 (¥¥.¥D: など)。
 - **physical_device_id** – Window デバイスマネージャにより割り当てられた名前。これらの名前は、コントロールパネルのディスクマネージャを使用して検索できます (¥¥.¥PhysicalDisk1 など)。
 - **permission_type** – オプションのアクセス動作のタイプ。指定しないと、現在のアクセスリストが表示されます。
 - DENY – 指定のトラスティに対して指定のデバイスへのアクセス拒否エントリを追加します。
 - GRANT – 指定のトラスティに対して指定のデバイスへのアクセス許可エントリを追加します。アクセス許可エントリは、既存の継承アクセス許可権限とマージされます。
 - REVOKE – 指定のトラスティに対する指定のデバイスへのアクセス許可エントリを削除します。
 - SET – 指定のトラスティに対する指定のデバイスへのアクセス許可エントリを設定します。
 - **trustee** – ユーザアカウントの有効なログイン ID。
3. パーミッションが正しく設定されていることを検証するには、アクセスパーミッションを表示します。

例:

ローデバイス E: へのアクセスパーミッションを user1 に付与します。

```
rawaccess ¥¥.¥E: /Grant user1@company.com
```

ローデバイスのアクセスパーミッションは次のように表示されます。

```
1. Allow All to COMPANY¥user1 (User)
2. Allow Exec to ¥Everyone (Well Known Group)
3. Allow All to NT AUTHORITY¥SYSTEM (Well Known Group)
4. Allow All to BUILTIN¥Administrators (Alias)
5. Allow Exec to NT AUTHORITY¥RESTRICTED (Well Known Group)
```

Windows でのローデバイス用シンボリックリンクの設定

シンボリックリンクを使用すると、Windows システムでのローデバイスアクセスが簡素化されます。シンボリックリンクは再起動されるまで保持されます。

前提条件

- Windows Vista 以降の Windows オペレーティングシステム。
- Windows Vista で管理者権限を持っているか、[ローカルセキュリティポリシー] 管理コンソールで動作を変更する必要があります。

手順

1. T などのデバイス文字により指定される NTFS ファイルシステムとしてシンボリックリンクのマウントポイントを作成します。
2. Windows コマンドプロンプトで、**mklink** コマンドを使用して、次のようにシンボリックリンクを作成します。

```
mklink T:¥IQ_SYSTEM_MAIN ¥¥.¥Volume{GUID}
```

データベースを作成したり領域を追加するスクリプトは、¥¥.

¥PhysicalDeviceN 形式ではなく、T:¥IQ_SYSTEM_MAIN を参照できます。

ストレージがオフラインになっても、システムがオンラインに戻ると、シンボリックリンクは維持されています。

必要な領域と DB 領域の見積もり

データベースまたは特定の DB 領域がいっぱいになった場合に生じる問題を避けるため、データベースとそのオブジェクトを作成する前に、DB 領域の要件を見積もります。

データベースに必要な領域の量と DB 領域の数を見積もるには、SAP Sybase IQ のストアドプロシージャを実行します。

1. **sp_iquestspace** を実行して、基本となるデータベーステーブル内のロー数に基づいて、データベースの作成に必要な領域の量を見積もります。作成する各テーブルに対して、次のプロシージャを実行します。 **sp_iquestspace** *table_name, rows[, iqpagesize]*.

各テーブルに必要な領域の容量は、"RAW DATA index_size" として返されます。

2. **sp_iquestdbspaces** を実行して、指定領域から作成する DB 領域ファイルの数とサイズを決定します。手順 5 で計算したインデックスの合計サイズを、*minsize* パラメータと *maxsize* パラメータとして次のように使用します。

```
sp_iquestdbspaces (dbsize [,iqpagesize]
[,minsize] [,maxsize] ...
```

これらの計算はすべて見積もりです。実際の結果は、データベースに作成したカラムとインデックスによって異なります。各プロシージャの構文と使用上の注意については、『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』の「システムプロシージャ」を参照してください。

DB 領域管理の例

この項では、新しいデータベースの作成から、データベースへのオブジェクト／データの追加、オブジェクトの移動、空の DB 領域の削除にいたるまで、一連の DB 領域管理プロセスを具体的に見ていきます。この例では、サンプル SQL コードと、関連するシステムストアプロシージャの出力も併せて紹介します。

データベースオブジェクトの作成

次の文を使用して、`dbspacedb` という小さなデータベースを作成します。

```
CREATE DATABASE 'D:¥IQ¥dbspacedb'
  IQ PATH 'D:¥IQ¥dbspacedb.iq'
  IQ SIZE 10
  IQ RESERVE 100
  TEMPORARY SIZE 10
  TEMPORARY RESERVE 10
  JCONNECT OFF;
```

`dbspacedb` データベースに接続します。

```
CONNECT DATABASE dbspacedb
  user DBA identified by sql;
```

`dbspacedb` データベースに 2 つの DB 領域を追加します。

```
CREATE DBSPACE dbspacedb2
  USING FILE dbspace2 'D:¥IQ¥dbspacedb.iq2'
  SIZE 10 RESERVE 20MB;CREATE DBSPACE dbspacedb3
  USING FILE dbspace3 'D:¥IQ¥dbspacedb.iq3'
  SIZE 10 RESERVE 40MB;
```

DB 領域のサイズの変更

次の例でテーブルを追加するには、前の例のメインストアは小さすぎます。ここでは、**ALTER DBSPACE** コマンドを使用して DB 領域のサイズを変更します。

データベース `dbspacedb` では、**CREATE DATABASE** 文の **IQ RESERVE** パラメータを使用して、IQ メインストア用の予約サイズを 100MB に設定しています。この IQ メインストア (IQ_SYSTEM_MAIN DB 領域) は 100MB 拡張できます。元の IQ_SYSTEM_MAIN は、10MB のサイズで作成されています (**CREATE DATABASE** の **IQ SIZE** パラメータ)。次の **ALTER DBSPACE** コマンドでは、**ADD** パラメータを使用して IQ_SYSTEM_MAIN DB 領域に 10MB を追加し、合計 20MB にしています。

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD 10mb;
sp_iqdbspace;
```

DBSpaceName	DBSpaceType	Writable	Online
IQ_SYSTEM_MAIN	MAIN	T	T
IQ_SYSTEM_TEMP	TEMPORARY	T	T
dbspacedb2	MAIN	T	T
dbspacedb3	MAIN	T	T

Usage	TotalSize	Reserve	NumFiles	NumRWFiles
25	20M	90M	1	1
7	10M	10M	1	1
1	10M	20M	1	1
1	10M	40M	1	1

Stripingon	StripeSize	BlkTypes	OK ToDrop
T	1K	1H、1248F、32D、177A、128M	N
T	1K	1H、64F、16A	N
T	1K	1H	Y
T	1K	1H	Y

IQ RESERVE 値を指定しないで `dbspacedb` データベースを作成した場合、DB 領域を拡張できません。ただし、DB 領域を小さくし、DB 領域からトランケートしたサイズを予約領域に追加することはできます。

現在の IQ_SYSTEM_MAIN DB 領域のサイズは 20MB です。DB 領域のサイズを 15MB に変更します。

データ記憶領域の管理

```
ALTER DBSPACE IQ_SYSTEM_MAIN SIZE 15mb;
sp_iqdbspace;
```

DBSpaceName	DBSpaceType	Writable	Online
IQ_SYSTEM_MAIN	MAIN	T	T
IQ_SYSTEM_TEMP	TEMPORARY	T	T
dbspacedb2	MAIN	T	T
dbspacedb3	MAIN	T	T

Usage	TotalSize	Reserve	NumFiles	NumRWFiles
25	15M	95M	1	1
7	10M	10M	1	1
1	10M	20M	1	1
1	10M	40M	1	1

Stripingon	StripeSize	BlkTypes	OK ToDrop
T	1K	1H、1248F、32D、177A、128M	N
T	1K	1H、64F、16A	N
T	1K	1H	Y
T	1K	1H	Y

DB 領域のサイズを減らすことができるのは、トランケートする部分が未使用である場合だけです。DB 領域上でオブジェクトに使用されているブロックを判別するには、**sp_iqdbspaceinfo** を使用します。

データベースオブジェクトの追加

dbspacedb データベースに 2 つのテーブルを作成し、インデックスを作成して、データを追加します。

```
CREATE TABLE t1(c1 int);
CREATE TABLE t2(c1 int);
CREATE hg INDEX t1c1hg ON t1(c1);
CREATE hng INDEX t2c1hng ON t2(c1);
INSERT t1 VALUES (1);
INSERT t2 VALUES (2);
COMMIT;
```

DB 領域に関する情報の表示

dbspacedb データベース内のすべての DB 領域に関する情報を表示します。次の例は、iqdemo データベース内の DB 領域を示しています。出力は 2 つの部分に分けて読みやすくしてあります。

```
sp_iqdbspace;
```

DBSpaceName	DBSpaceType	Writable	Online	Usage	TotalSize	Reserve
IQ_SYSTEM_MAIN	MAIN	T	T	25	10M	100M
IQ_SYSTEM_TEMP	TEMPORARY	T	T	7	10M	10M
dbspacedb2	MAIN	T	T	1	10M	20M
dbspacedb3	MAIN	T	T	1	10M	40M

NumFiles	NumRWFiles	Stripingon	StripeSize	BlkTypes	OK ToDrop
1	1	T	1K	1H、1248F、32D、177A、128M	N
1	1	T	1K	1H、64F、16A	N
1	1	T	1K	1H	Y
1	1	T	1K	1H	Y

特定の DB 領域のオブジェクトの配置と領域の使用状況に関する情報を表示します。

注意： 出力内容をわかりやすくするため、次の 2 つの例は 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、この領域はお使いのデータベースには存在しない場合があります。

次の出力は、複数の部分に分けて読みやすくしてあります。

```
sp_iqdbspaceinfo;
```

dbspace_name	object_type	owner	object_name	object_id	id
iq_main	table	DBA	emp1	3,813	743
iq_main	table	DBA	iq_dummy	3,801	742
iq_main	table	DBA	sale	3,822	744

データ記憶領域の管理

dbspace_name	object_type	owner	object_name	object_id	id
iq_main	table	GROUPO	Contacts	3,662	734
iq_main	table	GROUPO	Customers	3,639	733
iq_main	table	GROUPO	Departments	3,756	740
iq_main	table	GROUPO	Employees	3,765	741
iq_main	table	GROUPO	FinancialCodes	3,736	738
iq_main	table	GROUPO	FinancialData	3,745	739
iq_main	table	GROUPO	Products	3,717	737
iq_main	table	GROUPO	SalesOrderItems	3,704	736
iq_main	table	GROUPO	SalesOrders	3,689	735

columns	indexes	metadata	primary_key
96K	0B	1.37M	0B
24K	0B	464K	0B
96K	0B	1.22M	0B
288K	0B	5.45M	24K
240K	48K	4.63M	24K
72K	0B	1.78M	24K
408K	0B	8.03M	24K
72K	0B	1.53M	24K
96K	0B	2.19M	24K
272K	192K	4.67M	24K
120K	0B	2.7M	24K
144K	0B	3.35M	24K

unique_constraint	foreign_key	dbspace_online
0B	0B	Y
0B	0B	Y

unique_constraint	foreign_key	dbspace_online
0B	0B	Y
0B	48K	Y
0B	0B	Y
0B	48K	Y
0B	48K	Y
0B	0B	Y
0B	48K	Y
0B	0B	Y
0B	104K	Y
0B	144K	Y

sp_iqindexinfo システムストアプロシージャを使用して、特定のテーブルまたはインデックスのオブジェクトの配置と領域の使用状況を表示します。次の情報は、iqdemo データベースからの情報です。

```
sp_iqindexinfo 'table GROUPO.Customers';
```

Object	DBSpaceName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Customers	iq_main	200K	1	1,045,460	1,051,032
GROUPO.Customers.ASIQ_INDEX_T733_C10_FP	iq_main	440K	1	1,046,689	1,047,147
GROUPO.Customers.ASIQ_INDEX_T733_C1_FP	iq_main	440K	1	1,046,641	1,047,213
GROUPO.Customers.ASIQ_INDEX_T733_C2_FP	iq_main	440K	1	1,046,961	1,047,203
GROUPO.Customers.ASIQ_INDEX_T733_C3_FP	iq_main	440K	1	1,046,833	1,047,196
GROUPO.Customers.ASIQ_INDEX_T733_C4_FP	iq_main	440K	1	1,046,737	1,047,189
GROUPO.Customers.ASIQ_INDEX_T733_C5_FP	iq_main	440K	1	1,046,929	1,047,182

Object	DBSpaceName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Customers.ASIQ_INDEX_T733_C6_FP	iq_main	440K	1	1,047,009	1,047,175
GROUPO.Customers.ASIQ_INDEX_T733_C7_FP	iq_main	440K	1	1,046,945	1,047,168
GROUPO.Customers.ASIQ_INDEX_T733_C8_FP	iq_main	440K	1	1,046,785	1,047,161
GROUPO.Customers.ASIQ_INDEX_T733_C9_FP	iq_main	440K	1	1,046,881	1,047,154
GROUPO.Customers.ASIQ_INDEX_T733_I11_HG	iq_main	152K	1	1,047,121	1,047,206
GROUPO.Customers.IX_customer_name	iq_main	304K	1	1,050,995	1,051,038

sp_iqdbspace、**sp_iqdbspaceinfo**、および **sp_iqindexinfo** システムストアプロシージャの完全な構文については、『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』を参照してください。

データの格納

DBA は、どのテーブルおよびインデックスをどの DB 領域に入れるかを決定でき、オブジェクトを別の DB 領域に移動させたり、DB 領域内のデータを空にしてからその DB 領域を削除したりできます。

また、各 DB 領域に何回書き込みを行ったらディスクストライピングアルゴリズムが別のストライプに移動するかを定義することもできます。

DB 領域ファイルの属性と操作

DB 領域ファイルには、読み込み／書き込み状態または読み取り専用状態がありません。

DB 領域ファイルは、DB 領域ファイルを追加した時は読み込み／書き込み状態ですが、その実行時の読み込み／書き込み状態は、DB 領域と DB 領域ファイルの両方の読み込み／書き込み状態によって異なります。管理者は、DB 領域ファイルの読み取り／書き込み状態と読み取り専用状態を変更できますが、DB 領域ファイルのオンライン／オフライン状態を変更することはできません。

DB 領域ファイルでは、追加、削除、論理名の変更、ファイルパス名の変更などの操作を実行できます。

参照：

- ALTER DBSPACE 文 (444 ページ)

RLV DB 領域へのファイルの追加

RLV DB 領域にファイルを追加して、RLV トランザクションログの保管の容量を追加する必要があります。

前提条件

- SAP Sybase IQ サーバにシンプレックスデータベースがある。
- データベースに、RLV DB 領域が 1 つ存在し、オンラインである。
- DB 領域が読み書きである場合、RLV 対応オブジェクトが存在しない。

手順

文 **ALTER DBSPACE** *dbspace name* **ADD FILE** *filename* を使用します。

```
ALTER DBSPACE d1 ADD FILE 'rlv2.iq'
```

インメモリ RLV ストレージの性質上、ファイルを READONLY として指定することはできません。

RLV DB 領域からのファイルの削除

唯一のファイルではなく、使用中でもないファイルは RLV DB 領域から削除することができます。

前提条件

- SAP Sybase IQ サーバにシンプレックスデータベースがある。
- データベースに、RLV DB 領域が 1 つ存在する。
- RLV DB 領域は読み込み専用である。また DB 領域が読み書きの場合はファイルが使用中でない。

手順

文 **ALTER DBSPACE** *<dbspace name>* **DROP FILE** *<dbspace filename>* を使用します。

```
ALTER DBSPACE d1 DROP FILE rlv2
```

IQ_SYSTEM_TEMP からの DB ファイルの削除

DB ファイルを削除するには、**DROP DBSPACE** コマンドを発行します。

前提条件

DB 領域を削除するには、次の条件を満たす必要があります。

- `MANAGE ANY DBSPACE` システム権限が必要です。
- DB ファイルにはユーザテーブル内のデータを含めることはできません。SAP Sybase IQ では、そのファイルが空でないかぎり、DB ファイルを削除できません。
- DB 領域に少なくとも 1 つの読み書き可能な DB ファイルがある必要があります。
- DB ファイル上で旧バージョンのテーブルを使用するすべてのトランザクションをコミットまたはロールバックしておく必要があります。

手順

1. システムテーブルを使用するグローバルテナンティテーブルスキーマを探します。DROP TABLE コマンドを実行して、DB ファイルのすべてのオブジェクトを移動します。

```
SELECT * FROM systable WHERE table_type = 'GBL TEMP'
```

2. DB ファイルのすべての領域が空であることを確認します。

```
sp_iqfile IQ_SYSTEM_TEMP
```

OkToDrop カラムに "Y" が含まれている場合、DB ファイルを削除できます。

3. DB ファイルを削除します。マルチプレックスデータベースに対しては、絶対パスを使用します。

```
ALTER DBSPACE IQ_SYSTEM_TEMP DROP FILE FileHist3
```

DB 領域ファイルと出力ファイル

`SYSDBFIL` ビューには、使用しているデータベースにあるすべての DB 領域ファイルが表示されます。表示される DB 領域ファイルには、カタログ DB 領域ファイル、IQ メッセージファイル、IQ のメイン DB 領域とテナンティ DB 領域にある DB 領域ファイル、トランザクションログファイル、SA テンポラリファイルなどがあります。

DB 領域ファイル以外のファイルは、`SYSDBFIL` ビューには表示されません。これらのファイルには、コンソールログ (`-o` スイッチで指定されます) や SQL ログ (`-zo` で指定されます) など、サーバ起動時に生成されるファイルが含まれます。これらのログファイルは、データベースプロパティまたはサーバプロパティとして表示され、`sa_db_properties()` などのストアードプロシージャまたはシステム関数 `db_property()` によって検査される場合があります (構文については、表 14-1 (ページ 628) を参照してください)。

DB 領域の属性と操作

DB 領域には、オンライン、オフライン、動的にオフラインの3つの状態があります。

動的にオフラインとは、DB 領域がカタログでオフラインとマークされるのではなく、メモリでオフラインとマークされることを意味します。データベースの起動時に1つ以上のDB 領域ファイルをオープンできない場合、データベースは起動しますが、そのようなDB 領域は動的にオフラインとマークされます。管理者は問題を解決した後に **ALTER DBSPACE ONLINE** を使用してDB 領域をオンラインに戻すことができますが、この場合、DB 領域のメモリ内状態だけが変更されます。

注意： インデックス、データ、またはパーティションがオフラインのDB 領域にある場合には、テーブルデータにアクセスできません。

DB 領域には、オンライン、オフライン、動的にオフライン状態以外に、読み込み／書き込み状態または読み取り専用状態もあります。作成時は、DB 領域はオンラインかつ読み込み／書き込み状態です。

DB 領域には、ストライピング属性もあります。管理者は、ストライピングをオンにするかどうかとストライプサイズを指定できます。

DB 領域は、作成、変更、削除できます。

マルチプレックス DB 領域については、『管理：マルチプレックス』を参照してください。SAP Control Center のDB 領域の状態を変更する方法については、SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

DB 領域の名前変更ガイドライン

DB 領域またはDB 領域ファイルの名前は変更できますが、カタログDB 領域 SYSTEM、IQ メインDB 領域 IQ_SYSTEM_MAIN、IQ テンポラリDB 領域 IQ_SYSTEM_TEMP、共有テンポラリDB 領域 IQ_SHARED_TEMP、IQ メッセージDB 領域 IQ_SYSTEM_MSG の名前変更や削除はできません。

IQ_SYSTEM_MAIN や IQ_SYSTEM_TEMP にあるファイルの論理名と IQ_SYSTEM_MSG ファイルの論理名は変更できますが、SYSTEM にあるファイルの論理名は変更できません。ALTER DBSPACE RENAME TO を使用して、DB 領域 IQ_SYSTEM_MAIN、IQ_SYSTEM_TEMP、IQ_SYSTEM_MSG、または SYSTEM の名前を変更することはできません。

追加の DB 領域

CREATE DBSPACE 文を使用して DB 領域を作成します。

新しい DB 領域は既存の DB 領域と同じディスクドライブに作成しても、違うディスクドライブに作成してもかまいません。MANAGE ANY DBSPACE システム権限が必要です。

ローデバイスおよびオペレーティングシステムファイルにおける DB 領域の最大サイズについては、『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』の「物理的制限」を参照してください。プラットフォームによっては、ファイルシステムのファイルを、この最大サイズまで大きくできるようにする必要があります。

SIZE と **RESERVE** は、IQ ストアと IQ テンポラリストアにのみ指定できます。カタログストアには指定できません。

メインストアをローデバイス上に作成します。

新しい DB 領域に対してローデバイスを指定すると、SAP Sybase IQ はそのファイルサイズを自動的に決定し、IQ ストアとして使用できるようにデバイス全体を割り付けます。ファイルデバイスの場合、これは予測不可能な結果をもたらす可能性があります。

デバイスがローデバイスでないことを示すと、ファイルサイズを指定できるようになります。ウィザードでは、指定されたパスが存在するかどうかを確認されます。パスが存在しない場合、SAP Sybase IQ はエラーを返します。

リソース使用やパフォーマンスへの DB 領域数の影響

1 データベース当たりの DB 領域の最大数はオペレーティングシステムによって制限され、この制限は調整可能です。具体的な最大数は、1つのデータベースにつき 2,047 個の DB 領域に、カタログストア用の最大 12 個の DB 領域を足したものです。DB 領域の数を増やしても、メモリ使用状況やパフォーマンスに実際的な影響はありません。ただし、最大数に達するような状況にならないようにしてください。

注意： HP と AIX プラットフォームでは、より多くの DB 領域にデータを分割すると、重複 I/O のパフォーマンスが向上します。

データをローパーティション上に格納している場合は、ドライブごとに 1つの DB 領域を持つことができます。DB 領域のサイズ制限については、『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』の「物理的制限」を参照してください。

データをファイルシステムに格納している場合は、記憶領域システム内でストライプを利用できます。マルチユーザシステム上でオペレーティングシステムスト

ライピングまたはハードウェアストライピングを使用する場合、ストライプサイズは最小 1MB、または可能なかぎり大きいサイズである必要があります。ストライプサイズは、IQ ページサイズの数倍であることが必要です。また、ソフトウェアストライピングを実行するよう SAP Sybase IQ を設定することもできます。

DB 領域を追加する前に、領域の必要量を見積もることができます。リソースを最も効率的に使用するためには、DB 領域をバックアップメディアに入るくらいに小さく、そしてディスクを十分に埋めるくらいの大きさにします。

例

次のコマンドは、library.iq というオペレーティングシステムファイルを指す library という新しい DB 領域を IQ_SYSTEM_MAIN DB 領域と同じディレクトリに作成します。

```
CREATE DBSPACE library
USING FILE library
'library.iq' SIZE 100 MB IQ STORE
```

SAP Control Center に DB 領域を作成する方法については、SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

DB 領域を追加または削除したら、**CHECKPOINT** を発行します。システムリカバリは、最新のチェックポイントの後から開始されます。

DB 領域の削除に関するガイドライン

データベースファイルを削除するには、**DROP DBSPACE** コマンドを発行します。

DB 領域を削除するには、次の条件を満たす必要があります。

- **MANAGE ANY DBSPACE** システム権限が必要です。
- ユーザテーブルからのデータが含まれていてはなりません。SAP Sybase IQ では、そのファイルが空でないかぎり、DB 領域を削除できません。
- 必須の DB 領域であってはなりません。SYSTEM、IQ_SYSTEM_MAIN、IQ_SYSTEM_TEMP、または IQ_SYSTEM_MSG。これらの DB 領域は削除できませんが、他の DB 領域は、IQ メインストアまたはカタログストアから削除できます。ただし、DB 領域にユーザ作成のオブジェクトが含まれていない場合にかぎります。

DB 領域を空にするには、次の作業を行う必要があります。

- DB 領域のすべてのオブジェクトを移動または削除します。
- 古いバージョンのテーブルを使用しているトランザクションのみをコミットまたはロールバックします。

SAP Sybase IQ では、DB 領域へのデータの格納方法の都合上、テーブルを明示的に移動しただけで DB 領域が空になることはありません (特にディスクストライピ

ングを使用している場合)。通常は、空のテーブルも多少の領域を占めているため、DB 領域内のテーブルをトランケートしても DB 領域を空にできません。テーブルを移動するには、**ALTER TABLE MOVE** を使用します。

他のユーザが読み込み中のテーブルを移動すると、テーブルの通常のバージョン管理規則が適用されます。つまり、そのユーザのトランザクションが完了するまで、古いテーブルバージョンが保持されます。

DBA は、ストアードプロシージャ **sp_iqspaceinfo**、**sp_iqdbspaceinfo**、**sp_iqindexinfo** を実行することで、テーブルやインデックスが配置されている DB 領域を特定できます。これらのプロシージャは、各 DB 領域内でそれぞれのテーブルおよびインデックスによって使用されているブロックの数を示します。

特定の DB 領域を削除できるかどうかを調べるには、**sp_iqdbspace** を実行します。Block Types (Blk Types) カラムを調べると、各 DB 領域の内容がわかります。このカラムにブロックタイプ "H"、"F"、"R"、"B"、"X"、"C" が含まれ、他のブロックタイプが含まれていない場合は、DB 領域を削除できます。

ブロックタイプ "A" は、アクティブなテーブルバージョンのデータを表します。移動するテーブルを特定するには、**sp_iqdbspaceinfo** を使用します。

ブロックタイプ "O" は、まだ使用されている可能性がある古いバージョンを表します。アクティブな接続をロールバックまたはコミットして、この領域を解放する必要があります。ブロックタイプ "M" は、マルチプレックスを表します。

SAP Control Center を使用して DB 領域を削除する方法については、SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

読み取り専用および読み込み／書き込み DB 領域とファイル

ファイルは、ファイルの状態が読み取り専用または読み取り／書き込みのいずれかである場合は、読み取り専用ですが、所有する DB 領域の状態は読み取り専用です。

DB 領域を読み取り専用に変更しても、関連ファイルのカatalogの状態は読み取り専用に変更されませんが、オペレーティングシステムレベルで関連ファイルが読み取り専用になります。つまり、ファイルのカatalogの読み取り専用状態または読み込み／書き込み状態は維持されますが、ファイル内のデータは修正できません。

読み取り専用 DB 領域では、管理者は次の操作を実行できます。

- ファイルの追加
- DB 領域にある DB 領域ファイルのファイルパス名の変更 (メイン DB 領域をオフラインにする必要があります)

- 空のファイルの削除
- DB 領域、または DB 領域内の DB 領域ファイルの名前変更

DB 領域と関連ファイルの状態

DB 領域とその関連ファイルは、個々に読み取り専用 (RO) 状態または読み込み／書き込み (RW) 状態を持つことができます。

オブジェクト	ステータス	有効な状態	テーブルの DB 領域	テーブルの状態
dbspace1	RW	RW	dbspace1	RW
- file1	RO	RO		
- file2	RW	RW		
dbspace2	RO	RO	dbspace2	RO
- file1	RO	RO		
- file2	RW	RO		
dbspace3	RW	RO	dbspace3	RO
- file1	RO	RO		
- file2	RO	RO		

テーブルは、読み取り専用 DB 領域に割り当てると読み取り専用になります。テーブルパーティションも、読み取り専用 DB 領域に割り当てると読み取り専用になります。読み取り専用テーブルまたは読み取り専用テーブルパーティションでは、テーブルの入力、削除、更新、ロード、トランケート、更新可能カーソルを使用した挿入／削除／更新などのデータ修正を行うことはできません。読み取り専用テーブルまたは読み取り専用テーブルパーティションでは、ALTER TABLE によるカラムの追加／削除、インデックスの作成／削除などの DDL 操作は実行できません。

読み取り専用 DB 領域に書き込もうとすると、修正されたページがディスクにフラッシュされる際に検出されます。INSERT...VALUES 文の実行時に修正されたページは、INSERT...VALUES 文ではない次のコマンドが実行されるまで、データベースには書き込まれません (このような動作をするコマンドは INSERT...VALUES だけです)。SAP Sybase IQ では、読み取り専用テーブルまたは読み取り専用テーブルパーティションで DDL 操作を行うとエラーが返されます。

変更可能な DB 領域

変更可能な DB 領域の設定は、変更操作の種類と DB 領域の特定の属性によって異なります。変更操作は、DB 領域の状態 (オンラインまたはオフライン)、DB 領域の読み込み／書き込み状態、DB 領域のタイプによって制御されます。

表 6 : 変更可能な DB 領域の設定

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、IQ_SYSTEM_MAIN、または IQ_SYSTEM_TEMP での変更
オンラインの DB 領域			
	ALTER DBSPACE OFF-LINE	RO の場合は可	不可
	ALTER DBSPACE ON-LINE	不可	IQ_SYSTEM_MAIN の場合は可、IQ_SHARED_TEMP と IQ_SYSTEM_TEMP の場合は不可
	ALTER DBSPACE RE-ADONLY	RW の場合は可	不可
	ALTER DBSPACE READ-WRITE	RO の場合は可	不可
	ALTER STRIPING または STRIPESIZEKB	可	シンプレックスとマルチプレックスコーディネータ上では可
	RENAME DBSPACE	可	不可
	ADD FILE	可	可

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、IQ_SYSTEM_MAIN、またはIQ_SYSTEM_TEMP での変更
	DROP FILE	空かつ RO の場合は可	空かつ RO の場合は可。 IQ_SHARED_TEMP のファイルを削除する操作は、シングルノードモードで起動されたコーディネータ上でのみ実行される必要がある。シンプレックス内の IQ_SHARED_TEMP のファイルを削除する操作もサポートされる。 IQ_SHARED_TEMP で最初に RW になったファイルが、最後に削除される必要がある。
	ALTER FILE READONLY	RW の場合は可	RW であり、最後の RW DB 領域ファイルでない場合は可。IQ_SHARED_TEMP のファイルは、読み取り専用に変更できない。
	ALTER FILE READWRITE	RO の場合は可	RO の場合は可
	ALTER FILE SIZE	RW DB 領域と DB 領域ファイルの場合は可	RW の場合は可
	ALTER FILE RENAME LOGICAL NAME	可	可
	ALTER FILE RENAME PATH	不可	IQ_SHARED_TEMP と IQ_SYSTEM_TEMP の場合は可。変更はデータベースの再起動後に有効になる。IQ_SYSTEM_MAIN の場合は不可。
オフラインの DB 領域			
	ALTER DBSPACE OFF-LINE	不可	該当なし

データ記憶領域の管理

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、IQ_SYSTEM_MAIN、またはIQ_SYSTEM_TEMP での変更
	ALTER DBSPACE ONLINE	可	該当なし
	ALTER DBSPACE READONLY	不可	該当なし
	ALTER DBSPACE READWRITE	不可	該当なし
	ALTER STRIPING または STRIPESIZEKb	可	該当なし
	RENAME DBSPACE	可	該当なし
	ADD FILE	不可	該当なし
	DROP FILE	空の場合は可	該当なし
	ALTER FILE RO	可	該当なし
	ALTER FILE RW	可	該当なし
	ALTER FILE SIZE	不可	該当なし
	ALTER FILE RENAME LOGICAL NAME	可	該当なし
	ALTER FILE RENAME PATH	可	なし
動的にオフラインのDB領域			
	ALTER DBSPACE OFFLINE	RO の場合は可	該当なし
	ALTER DBSPACE ONLINE	可	IQ_SYSTEM_MAIN と IQ_SYSTEM_TEMP の場合は N/A、IQ_SHARED_TEMP の場合は可

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、 IQ_SYSTEM_MAIN、または IQ_SYSTEM_TEMP での変更
	ALTER DBSPACE RE-ADONLY	RW の場合は可	該当なし
	ALTER DBSPACE READ-WRITE	不可	該当なし
	ALTER STRIPING または STRIPESIZEKKB	可	該当なし
	RENAME DBSPACE	可	該当なし
	ADD FILE	不可	該当なし
	DROP FILE	空の場合は可	該当なし
	ALTER FILE READONLY	不可	該当なし
	ALTER FILE READWRITE	不可	該当なし
	ALTER FILE SIZE	不可	該当なし
	ALTER FILE RENAME LOGICAL NAME	可	該当なし
	ALTER FILE RENAME PATH	不可	該当なし
読み取り専用 DB 領域			
	ALTER DBSPACE OFF-LINE	オンラインの場合は可	該当なし
	ALTER DBSPACE ON-LINE	オフラインの場合は可	該当なし
	ALTER DBSPACE RE-ADONLY	不可	該当なし
	ALTER DBSPACE READ-WRITE	オンラインの場合は可	該当なし

データ記憶領域の管理

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、IQ_SYSTEM_MAIN、またはIQ_SYSTEM_TEMP での変更
	ALTER STRIPING および STRIPESIZEKB	可	該当なし
	RENAME DBSPACE	可	該当なし
	ADD FILE	可	該当なし
	DROP FILE	空の場合は可	該当なし
	ALTER FILE READONLY	RW の場合は可	該当なし
	ALTER FILE READWRITE	RO の場合は可	該当なし
	ALTER FILE SIZE	不可	該当なし
	ALTER FILE RENAME LOGICAL NAME	可	該当なし
	ALTER FILE RENAME PATH	オフラインの場合は可	該当なし
読み込み/書き込み DB 領域			
	ALTER DBSPACE OFF-LINE	不可	不可
	ALTER DBSPACE ON-LINE	動的にオフラインの場合は可	IQ_SYSTEM_MAIN と IQ_SYSTEM_TEMP の場合は不可、IQ_SHARED_TEMP で動的にオフラインの場合は可
	ALTER DBSPACE RE-ADONLY	可	不可
	ALTER DBSPACE READ-WRITE	不可	不可

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、IQ_SYSTEM_MAIN、またはIQ_SYSTEM_TEMP での変更
	ALTER STRIPING および STRIPESIZEKB	可	可
	RENAME DBSPACE	可	不可
	ADD FILE	可	可
	DROP FILE	空の場合は可	空かつ RO の場合は可。 IQ_SHARED_TEMP のファイルを削除する操作は、シングルノードモードで起動されたコーディネータ上でのみ実行される必要がある。シンプレックス内の IQ_SHARED_TEMP のファイルを削除する操作もサポートされる。 IQ_SHARED_TEMP で最初に RW になったファイルが、最後に削除される必要がある。
	ALTER FILE READONLY	RW の場合は可	RW の場合は可。IQ_SHARED_TEMP のファイルは、読み取り専用に変更できない。
	ALTER FILE READWRITE	RO の場合は可	RO の場合は可
	ALTER FILE SIZE	RW の場合は可	RW の場合は可
	ALTER FILE RENAME LOGICAL NAME	可	可
	ALTER FILE RENAME PATH	不可	可。変更はデータベースの再起動後に有効になる。
読み取り専用ファイル			
	ALTER FILE READONLY	不可	不可
	ALTER FILE READWRITE	可	可

ステータス	変更の種類	ユーザメインでの変更	IQ_SHARED_TEMP、IQ_SYSTEM_MAIN、またはIQ_SYSTEM_TEMP での変更
	ALTER FILE SIZE	不可	不可
	ALTER FILE RENAME LOGICAL NAME	可	可
	ALTER FILE RENAME PATH	オフラインの場合は可	可。変更はデータベースの再起動後に有効になる。
読み込み／書き込みファイル			
	ALTER FILE READONLY	可	可
	ALTER FILE READWRITE	不可	不可
	ALTER FILE SIZE	DB 領域が RW かつオンラインの場合は可	可
	ALTER FILE RENAME LOGICAL NAME	可	可
	ALTER FILE RENAME PATH	不可	可。変更はデータベースの再起動後に有効になる。

注意：

- 動的にオフラインとは、DB 領域がカタログでオフラインとマークされるのではなく、メモリ内でオフラインとマークされることを意味します。
- 読み取り専用 (RO) IQ_SYSTEM_MAIN DB 領域ファイルは、動的にオフラインにできます。
- IQ_SYSTEM_MSG の場合、許可されている修正は、パス名の変更だけです。この変更は、コマンド ALTER DBSPACE IQ_SYSTEM_MSG RENAME 'filepath' を使用して実行されます。

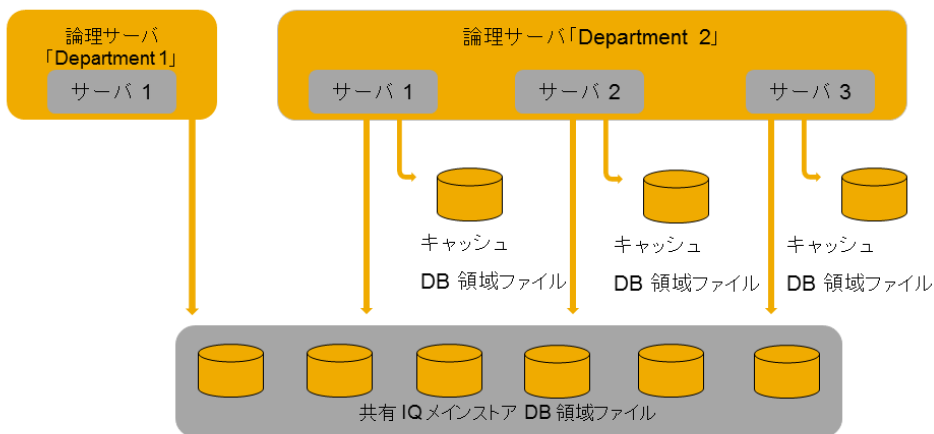
直接接続ストレージのキャッシュ DB 領域

各ノードにローカルのキャッシュ DB 領域を設定することにより、ソリッドステートディスク (SSD) などの直接接続ストレージデバイスをマルチプレックスまたはシンプレックスサーバに追加します。ローカル直接接続ストレージは、データベースに SSD バッファプール拡張を提供します。これにより、一般的には I/O パフォーマンスが、特に共有 SAN デバイスのパフォーマンスが向上します。直接接続ストレージは、マルチプレックス構成のときにもっとも大きなメリットがありますが、シンプレックスサーバでもメリットはあります。

キャッシュ DB 領域は、IQ メインストア (IQ_SYSTEM_MAIN)、IQ テンポラリストア (IQ_SYSTEM_TEMP)、RLV (IQ RLV STORE) DB 領域のように独立した DB 領域です。すべての DB 領域のように、キャッシュ DB 領域も、単一のファイルやデバイス、または複数のファイルやデバイスを含むことができます。1 シンプレックスおよびマルチプレックスノード当たり最大で1つのキャッシュ DB 領域を設定できます。

注意： キャッシュ DB 領域を IQ メインストアのメインバッファキャッシュと混同しないでください。メインバッファキャッシュサイズは、データベース起動ユーティリティ `start iq` オプション `-iqmc` (メインバッファキャッシュサイズ) を使用して設定します。『ユーティリティガイド』を参照してください。

キャッシュ DB 領域を作成および設定すると、サーバは、ローカルに使用される読み込み専用ページ (アフィニティ化 (優先) ページと非アフィニティ化ページの両方) をキャッシュ DB 領域にキャッシュします。IQ メインストア DB 領域ファイルからの以降の読み込みでは、サーバはまずキャッシュ DB 領域をチェックして、ページが存在しているかどうかを確認します。存在している場合、サーバは共有 SAN IQ メインストア DB 領域ではなく、高速 SSD キャッシュ DB 領域からページを読み込むので、パフォーマンスが向上します。



キャッシュ DB 領域に、ノードのページのワーキングセット全体を格納できるだけの容量がある場合、パフォーマンスが最大になります。キャッシュ DB 容量がページのワーキングセットより小さい場合、パフォーマンス上のメリットも小さくなります。

キャッシュ DB 領域はデータのアフィニティ機能と連動してパフォーマンスを向上させます。データアフィニティは、キャッシュされたデータにアクセスすることによって、マルチプレックスサーバのクエリパフォーマンスを向上させます。アフィニティは、マルチプレックスデータベースに対しては自動的に有効になります。アフィニティは、ノードに割り当てられているパーティションのページをそのノードに対するメインバッファキャッシュメモリに格納しようとしています。ただし、アプリケーションのページのワーキングセットが、ノードのメインバッファキャッシュメモリを超えている場合、直接接続ストレージをセカンダリ高速アクセスレイヤとして追加します。

キャッシュ DB 領域の作成

キャッシュ DB 領域を作成して、直接接続ストレージ (DAS) デバイスをシングルプレックスノードまたはマルチプレックスノードに追加します。キャッシュ DB 領域は、ストレージエリアネットワークからの読み込みを減らすことにより、パ

パフォーマンスが向上します。各マルチプレックスノードとシンプレックスノードに1つのキャッシュ DB 領域を設定できます。

前提条件

MANAGE ANY DBSPACE システム権限が必要です。

手順

1. シンプレックスノードまたはマルチプレックスノードに接続します。
2. キャッシュ DB 領域と DB ファイルを作成します。CREATE DBSPACE 文で IQ CACHE STORE キーワードを指定します。

```
CREATE DBSPACE myDAS
USING FILE iqdas1
'/dev/rdisk/ssd104' IQ CACHE STORE
```

キャッシュ DB 領域の削除

シンプレックスまたはマルチプレックスノードからキャッシュ DB 領域を削除します。他の DB 領域とは異なり、キャッシュ DB 領域は削除前に空にする必要はありません。

前提条件

- シンプレックスノードまたはマルチプレックスノード上のキャッシュ DB 領域。
- MANAGE ANY DBSPACE システム権限が必要です。

手順

1. シンプレックスノードまたはマルチプレックスノードに接続します。
2. キャッシュ DB 領域を削除します。

```
DROP DBSPACE myDAS
```

キャッシュ DB 領域へのストレージの追加

ファイルまたはデバイスパスを指定して、直接接続ストレージをキャッシュ DB 領域に追加します。

前提条件

- シンプレックスノードまたはマルチプレックスノード上のキャッシュ DB 領域。
- MANAGE ANY DBSPACE システム権限が必要です。

手順

1. シンプレックスノードまたはマルチプレックスノードに接続します。
2. ファイルまたはデバイスパスを指定します。

- ファイルを指定します。

```
ALTER DBSPACE myDAS
ADD FILE iqdas2
'sampled.iqcache' SIZE 1024
```

- デバイスパスを指定します。

```
ALTER DBSPACE myDAS
ADD FILE iqdas2
'/dev/rdisk/clt2d3s4'
```

キャッシュ DB 領域からのストレージの削除

ファイルまたはローデバイスを削除することにより、キャッシュ DB 領域から直接接続ストレージを削除します。

前提条件

- シンプレックスノードまたはマルチプレックスノード上のキャッシュ DB 領域。
- `MANAGE ANY DBSPACE` システム権限が必要です。

手順

DB 領域が読み書きステータスである場合でも、キャッシュ DB 領域から DB ファイルを削除できます。

1. シンプレックスノードまたはマルチプレックスノードに接続します。
2. `ALTER DBSPACE` 文を使用して単一のファイルまたはローデバイスを削除します。

```
ALTER DBSPACE myDAS DROP FILE iqdas2
```

注意： `DROP FILE` 句の複数使用は許可されません。

キャッシュ DB 領域の無効化と再有効化

キャッシュ DB 領域のオンライン/オフラインのステータスを変更することにより、キャッシュ DB 領域を削除することなく無効にしたり、無効になったキャッシュ DB 領域を再度有効にしたりできます。

前提条件

- シンプレックスノードまたはマルチプレックスノード上のキャッシュ DB 領域。
- MANAGE ANY DBSPACE システム権限が必要です。

手順

1. シンプレックスノードまたはマルチプレックスノードに接続します。
2. ALTER DBSPACE 文を使用して、キャッシュ DB 領域を無効にしたり、無効にしたキャッシュ DB 領域を再有効化します。

- キャッシュ DB 領域

```
ALTER DBSPACE myDAS OFFLINE
```

を無効にします。

- 無効のキャッシュ DB 領域を再有効化します。

```
ALTER DBSPACE myDAS ONLINE
```

キャッシュ DB 領域の DB ファイルの読み込み専用/読み書きステータスの変更

キャッシュ DB 領域内の単一の DB ファイルを読み込み専用ステータスまたは読み書きステータスに変更します。

前提条件

- シンプレックスノードまたはマルチプレックスノード上のキャッシュ DB 領域。
- MANAGE ANY DBSPACE システム権限が必要です。

手順

1. シンプレックスノードまたはマルチプレックスノードに接続します。
2. ALTER DBSPACE 文を使用して、DB ファイルを読み込み専用または読み書きステータスに変更します。

- キャッシュ DB 領域の DB ファイルを読み込み専用に変更します。
`ALTER DBSPACE myDAS ALTER FILE iqdas2 READONLY`
- キャッシュ DB 領域の DB ファイルを読み書きステータスに変更します。
`ALTER DBSPACE myDAS ALTER FILE iqdas2 READWRITE`

キャッシュ DB 領域のモニタリング

`sp_iqstatus`、`sp_iqdbspace`、`sp_iqfile`、および `sp_iqcheckdb` ストアドプロシージャを使用して、キャッシュ DB 領域とその DB ファイルの主要パフォーマンス指標をモニタします。

前提条件

- `sp_iqstatus` は、ALTER DATABASE、MANAGE ANY DBSPACE、MONITOR、または SERVER OPERATOR システム権限を必要とします。これらのシステム権限のいずれも保持していないユーザは、EXECUTE 権限が付与されている必要があります。
- `sp_iqdbspace` は MANAGE ANY DBSPACE システム権限を必要とします。MANAGE ANY DBSPACE システム権限を持たないユーザがこのストアドプロシージャを実行するには、EXECUTE 権限が付与されている必要があります。
- `sp_iqfile` は MANAGE ANY DBSPACE システム権限を必要とします。MANAGE ANY DBSPACE システム権限がないユーザは、EXECUTE 権限が付与されている必要があります。

手順

1. `sp_iqstatus` を実行して、キャッシュ DB 領域とその DB ファイルに関するメモリ使用率情報を表示します。

```
...
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
...
    
```

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。
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。

詳細については、『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』の「sp_iqstatus プロシージャ」を参照してください。

2. **sp_iqdbspace** を実行して、読み込み／書き込みステータスやオンライン／オフ
ラインステータスなどのキャッシュ DB 領域についての詳細情報を表示しま
す。CREATE DBSPACE 文に指定したキャッシュ DB 領域名 (この例では
myDAS) のローを表示します。キャッシュ DB 領域では、DBSpaceType 列が
CACHE になっています。

```

IQ_SYSTEM_MAIN MAIN          T T 21 100M 0B 1 1 T 1K 1H,5088F,32D,128M
N
IQ_SYSTEM_TEMP TEMPORARY    T T 1  100M 0B 1 1 T 1K 1H,64F,16A N
myDAS CACHE                  T T 1  100M 0B 5 5 T 1K 5H,192F Y
MainUser MAIN                T T 1  100M 0B 1 1 T 1K 1H Y
    
```

カラム	説明
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)。

カラム	説明
Reserve	DB 領域のすべてのファイルに追加できる予約領域の合計。
NumFiles	DB 領域内のファイルの数。
NumRWFiles	DB 領域内の読み込み/書き込みファイルの数。
StripingOn	T (オン) または F (オフ)。
StripeSize	ディスクストライピングの有効化時は、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"。

3. `sp_iqfile` を実行して、キャッシュ DB 領域内の各 DB ファイルについて詳細情報を表示します。 **CREATE DBSPACE** 文に指定したキャッシュ DB 領域名 (この例では `myDAS`) のローを表示します。

```

IQ_SYSTEM_MAIN IQ_SYSTEM_MAIN exampleDB.iq MAIN RW T 21 100M 0B 1K 1H,5088F,32D,128M 1 25600
MainUser MainUser exampleDB.iq2 MAIN RW T 1 100M 0B 1K 1H 522208 547807
myDAS ssd_dev_1 /dev/raw/ssd_dev_1 CACHE RW T 2 20M 0B 1K 1H,64F 1 5120
myDAS ssd_dev_2 /dev/raw/ssd_dev_2 CACHE RW T 1 20M 0B 1K 1H,32F 522208 527327
myDAS ssd_dev_3 /dev/raw/ssd_dev_3 CACHE RW T 1 20M 0B 1K 1H,32F 1044416 1049535
myDAS ssd_dev_4 /dev/raw/ssd_dev_4 CACHE RW T 1 20M 0B 1K 1H,32F 1566624 1571743
myDAS ssd_dev_5 /dev/raw/ssd_dev_5 CACHE RW T 1 20M 0B 1K 1H,32F 2088832 2093951
IQ_SYSTEM_TEMP IQ_SYSTEM_TEMP exampleDB.iqtmp TEMPORARY RW T 1 100M 0B 1K 1H,64F,16A 1 25600
    
```

カラム	説明
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	CACHE
RWMode	DB 領域のモード。読み込み/書き込み (RW) または読み込み専用 (RO)。
Online	T (オンライン) または F (オフライン)。

カラム	説明
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"。

詳細については、『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』の「sp_iqstatus プロシージャ」を参照してください。

次のステップ

sp_iqcheckdb ストアドプロシージャを使用して、キャッシュ DB 領域に対するデータベース検証を実行します。『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「sp_iqcheckdb プロシージャ」を参照してください。

SAP Sybase IQ データベースオブジェクトの管理

SAP Sybase IQ では、テーブルやビューなどのデータベースオブジェクトを作成、変更、削除できます。

注意： SAP Sybase IQ は、カタログストアと IQ ストアで構成されています。この項では、IQ ストアにオブジェクトを作成する方法について説明します。カタログストアに作成されたテーブルは、SQL Anywhere テーブルの特性を持っています。カタログストアにテーブルを作成する場合には、SQL Anywhere のマニュアルを参照してください。

SAP Sybase IQ データベース設計

データベースを正しく設計することにより、データの操作性とデータ取得の速度が向上します。

データベースを作成する前に、データの格納場所、データベースで必要な領域の量、データベースを定義または変更できるユーザを決定する必要があります。SAP Sybase IQ は、この作業を支援するツールとプロセスを備えています。

Sybase PowerDesigner® はデータベースの設計を支援するツールであり、概念的、物理的、またはオブジェクト指向のデータモデルを作成してから、そのモデルに基づいてデータベースを生成します。また、リバースエンジニアリングを行って、既存のデータベースからモデルを作成することもできます。

どの設計ツールを使用するにしても、通常は、データベースを設計して内容を定義するのはデータベース管理者 (DBA) の仕事です。効率的な設計を作成するには、DBA は組織内の人間それぞれに対して調査を行い、データの使われ方を把握する必要があります。DBA はまた、SAP Sybase IQ データベースの基本となる概念を理解する必要があります。

SAP Sybase IQ データベースは、データウェアハウスとして使用するために最適化されたリレーショナルデータベースです。リレーショナルデータベースとしては、データを編成する関連テーブルのセットを持ちます。またデータウェアハウスとしては、非常に大きなデータのセットへの、インデックスを使用した効率的なアクセスを提供します。

データベースを作成するときには、テーブルの構造、テーブル内で許可されるデータ型、テーブル間の関係、テーブルデータを格納するインデックス、データ

へのアクセスを制御するビューを指定します。データベースを作成する前に、『SAP Sybase IQ の概要』を参照して概念を理解しておいてください。

ビューの管理

ビューは計算されたテーブルです。

ビューを使うと、データベースのユーザが必要とする情報を、制御可能なフォーマットで表示できます。

ビューとベーステーブルの類似点

ビューは、次のような点でデータベースの永久テーブル(「ベーステーブル」ともいいます)に似ています。

- ベーステーブルと同様、ビューにもアクセスパーミッションを設定可能
- ビュー上でも **SELECT** クエリの実行が可能
- 一部のビューに対しては、**INSERT** 操作と **DELETE** 操作の実行が可能
- 他のビューに基づくビューの作成が可能

ビューと永久テーブルの相違点

ビューと永久テーブルの相違点を次に示します。

- ビュー上ではインデックスの作成は不可
- 特定のビュー上では、**INSERT**、**DELETE**、**UPDATE** 操作のみ実行可能
- ビューに対する整合性制約とキーの設定は不可
- ビューはベーステーブル内の情報を参照するだけで、その情報のコピーは保持しない。ビューは呼び出されるたびに再計算される。

データアクセスを調整する利点

ビューを使用して、データベース内のデータへのアクセスを調整します。アクセスの調整にはいくつか目的があります。

- ユーザのアクションに関係のないデータへのアクセスを制限します。
- ベーステーブルよりも見やすい形でユーザとアプリケーション開発者にデータを提供します。
- よく使われるクエリの定義をデータベース内で集中管理します。

ビューの作成

SELECT 文は 1 つ以上のテーブルで動作して、テーブルでもある結果セットを作成します。

ベーステーブルと同様に、**SELECT** クエリの結果セットにもカラムとローがあります。ビューを作成するクエリには名前が与えられ、システムテーブルに定義が格納されます。

例

各部署の従業員数を頻繁にリストするとします。このリストは次の文で取得できません。

```
SELECT DepartmentID, COUNT(*)
FROM Employees
GROUP BY DepartmentID
```

このクエリの結果を含むビューを作成するには次の文を使います。

```
CREATE VIEW DepartmentSize AS
SELECT DepartmentID, COUNT(*)
FROM Employees
GROUP BY DepartmentID
```

ビュー内の情報は、独立した形でデータベースに格納されているわけではありません。ビューを参照するたびに関連する **SELECT** 文が実行され、適切なデータが取り出されます。

この方法の長所は、Employees テーブルに変更があると、DepartmentSize ビューの情報も自動的に更新されることです。ただし、**SELECT** 文が複雑な場合は、ビューを使用するたびに SQL で正確な情報を収集する時間が長くなります。

ビューの使用におけるガイドライン

ビュー作成に使用する **SELECT** 文にも、ビューの挿入、削除、更新の機能にも、一定の制限があります。

SELECT 文に対する制限

ORDER BY 句を **SELECT** クエリで使用することはできません。リレーショナルテーブルでは、ローやカラムの並び順には意味がありませんが、**ORDER BY** 句を使用すると、ビューのローの順序が規定されるからです。**GROUP BY** 句、サブクエリ、ジョインは、ビューの定義で使用できます。

スカラー値サブクエリは、最上位レベルの **SELECT** リスト内でのみサポートされます (ビュー、派生テーブル、サブクエリではサポートされません)。最上位レベルの **SELECT** の **FROM** 句で使用するビューまたは派生テーブルが単純なため、最上位レベルの **SELECT** にフラット化できる場合もあります。結果として、前述のルールが実際に適用されるのは、サブクエリ、フラット化されていないビュー、およびフラット化されていない派生テーブルに対してのみです。次に例を示します。

```
CREATE VIEW test_view AS SELECT testkey, (SELECT COUNT(*) FROM
tagtests WHERE tagtests.testkey = testtrd.testkey ) FROM
testtrd
```

```
SELECT * FROM test_view
Msg 21, Level 14, State 0:
```

```
SQL Anywhere Error -1005004: Subqueries are allowed only as arguments of
comparisons, IN, and EXISTS,
-- (opt_Select.cxx 2101)
```

ビューを作成するには、必要とする正確な結果が必要なフォーマットで得られるまで **SELECT** クエリを編集します。思いどおりの **SELECT** クエリが完成したら、そのクエリの先頭に次のフレーズを追加してビューを作成します。次に例を示します。

```
CREATE VIEW viewname AS
```

ビューの挿入と削除に関するガイドライン

ビューに関連付けられている **SELECT** 文によっては、そのビューに対する **UPDATE**、**INSERT**、**DELETE** 文を実行できないことがあります。

以下を含むビューでは更新、挿入、または削除を行うことができません。

- **COUNT(*)** などの集合関数
- **SELECT** 文の **GROUP BY** 句
- **UNION** 操作

これらの場合には、**UPDATE**、**INSERT**、または **DELETE** コマンドから基本となるテーブルに対する操作を処理できないためです。

警告！ dbo ユーザ ID はシステムオブジェクトを所有するため、dbo ユーザ ID が所有するビューを削除しないでください。このようなビューを削除したり、テーブルに変更したりすると、予期しない問題が発生する可能性があります。

ビューの使用に関するガイドライン

ビューを修正するには、**ALTER VIEW** 文を使用します。

ALTER VIEW 文は、既存のビューの定義を修正するのではなく、ビューの定義を新しい定義に置き換えます。

ALTER VIEW 文では、ビューのパーミッションが保持されます。

例

たとえば、前述の DepartmentSize ビューのカラム名をよりわかりやすい名前に置き換える場合は、次の文を使用できます。

```
ALTER VIEW DepartmentSize
  (DepartmentID, NumEmployees)
AS
  SELECT DepartmentID, COUNT(*)
  FROM Employees
  GROUP BY DepartmentID
```

ビューに対するパーミッション

次に示す条件が 1 つ以上満たされている場合、ビューを通して操作を実行できません。

- ビューに対するその操作の適切なパーミッションが、DBA によってユーザに付与されている。
- ユーザがすべてのオブジェクト (ベーステーブル、ビュー、プロシージャなど) に対し、その操作を行う適切なパーミッションを持っている。
- ビューに対するその操作の適切なパーミッションが、非 DBA ユーザによってユーザに付与されている。この非 DBA ユーザはビューの所有者であるか、ビューに対する適切なパーミッションの WITH GRANT OPTION を持っている必要がある。ビューの所有者は次のいずれかである。

- DBA

- DBA ではないが、ビューによって参照されるすべてのオブジェクトの所有者でもある。

- DBA ではなく、ビューが参照する一部またはすべてのオブジェクトの所有者ではないが、ビューの所有者が自分の所有していないオブジェクトに対する SELECT パーミッションの WITH GRANT OPTION と、操作に必要なその他のパーミッションの WITH GRANT OPTION を持っている。

所有者がオブジェクトに対するパーミッション (WITH GRANT OPTION) を保持する代わりに、PUBLIC にパーミッションが付与される場合もあります。これには、システムテーブルに対する SELECT パーミッションが含まれます。

UPDATE パーミッションはビュー全体に対してのみ付与できます。テーブルの場合と異なり、ビュー内の各カラムに対する UPDATE パーミッションは付与できません。

ビューを削除する方法

Interactive SQL のデータベースからビューを削除するには、**DROP** 文を使用します。次に DepartmentSize ビューを削除する文を示します。

```
DROP VIEW DepartmentSize
```

システムビューに含まれるビュー情報

データベースのビューに関する情報は、SYS.SYSVIEW システムビューにあります。

『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「システムテーブルとシステムビュー」を参照してください。

このビューの情報を参照するには、Interactive SQL を使用します。**dbisql** コマンドウィンドウに次の文を入力すると、SYS.SYSVIEW ビューのすべてのカラムが表示されます。

```
SELECT *  
FROM SYS.SYSVIEW
```

特定のビューの定義を含むテキストファイルを抽出する文の例を、次に示します。

```
SELECT view_def FROM SYS.SYSVIEW  
WHERE view_object_id = 1583;  
OUTPUT TO viewtext.sql  
FORMAT ASCII
```

テーブルの管理

データベースを作成したときにデータベース内にあるテーブルは、データベーススキーマを保持する「システムテーブル」だけです。

データベースでのテーブルの作成、変更、削除が必要な場合があります。**dbisql** を使用してマニュアルに掲載されている例を実行できますが、SQL 文は使用する管理ツールには依存しません。

データベースのテーブルを定義してソースコードコントロールシステムに格納する、**CREATE TABLE** 文と **ALTER TABLE** 文を含んだコマンドファイルを作成できます。このコマンドファイルで、必要に応じてデータベースを再作成できます。また、標準化した方法でテーブルを作成し、それをコピーしたり、修正したりすることができます。

テーブル作成のガイドライン

Interactive SQL を使用してテーブルを作成します。

Interactive SQL でのテーブル作成

テーブルを作成する SQL 文は **CREATE TABLE** です。

この項では、**CREATE TABLE** 文を使用する方法について説明します。この項の例では、サンプルデータベースを使用します。例を試してみる場合は、**dbisql** を実行し、ユーザ ID DBA とパスワード sql を指定してデモデータベースに接続します。

Interactive SQL 以外のツールでもテーブルを作成できます。次に説明する SQL 文は、使用するツールに依存しません。

例

次の文は、会社内の従業員のスキルレベルを記述する永久的な IQ テーブルを新しく作成します。テーブルには、各スキルの ID 番号、名前、種別 (technical または administrative) のカラムが作成されます。

```
CREATE TABLE skill (  
  skill_id INTEGER NOT NULL,  
  skill_name CHAR( 20 ) NOT NULL,  
  skill_type CHAR( 20 ) NOT NULL  
)
```

このコマンドを実行するには、**dbisql** コマンドウィンドウに入力して、実行キー ([F9] キー) を押します。

- それぞれのカラムにはデータ型があります。skill_id カラムは integer (101 などの整数)、skill_name カラムは最大 20 字の固定幅の CHARACTER 文字列です。
- データ型の後に指定されているフレーズ NOT NULL は、この例のすべてのカラムに値を入れなければならないことを示します。
- 通常、プライマリキーを持たないテーブルは作成しません。

SAP Sybase IQ はテーブル作成前に **COMMIT** 文を内部的に実行し、データベースに対するそれまでの変更すべてを永久的なものにします。テーブル作成後も **COMMIT** 文が実行されます。

警告！ グローバルテーブルまたはベーステーブルを変更または作成する際、データベースの他のユーザに支障をきたす場合があります。大きなテーブルの場合、**ALTER** または **CREATE TABLE** の実行に時間がかかる可能性があります。**CREATE TABLE** の処理によって、文が完了するまで、他の IQ プロセスの実行が遅れます。**ALTER TABLE** 文は、他の接続がアクティブなときでも実行できますが、変更するテーブルを他の接続が使用しているときには実行できません。**ALTER TABLE** の処理によって、文の処理中に提供されるテーブルを参照している他の要求が除外されます。

データ型の指定

テーブルの作成時に、各カラムのデータ型を指定します。

カスタマイズしたデータ型も定義できます。サポートされているデータ型のリストについては、『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』を参照してください。カスタマイズしたデータ型の作成方法の詳細については、『リファレンス：文とオプション』の「**CREATE DOMAIN** 文」を参照してください。

テーブルタイプ

SAP Sybase IQ では、次の 4 つのテーブルタイプが認識されます。

- ベーステーブル
- ローカルテンポラリテーブル
- グローバルテンポラリテーブル
- ジョイン仮想テーブル

永久的なベーステーブル

ベーステーブルは、明示的に削除されるまでデータベース内に永久的に維持されるため、メインテーブル、永続テーブル、永久テーブルとも呼ばれます。

ユーザの切断、サーバの再起動、リカバリを行っても、ベーステーブルは維持されます。ベーステーブルとその中のデータには、適切なパーミッションを持つすべてのユーザがアクセスできます。前の例で示した **CREATE TABLE** 文はベーステーブルを作成します。

テンポラリテーブルの作成

テンポラリテーブルには、グローバルとローカルの 2 種類があります。

CREATE TABLE に **GLOBAL TEMPORARY** オプションを使用して、グローバルテンポラリテーブルを作成します。

作成したグローバルテンポラリテーブルは、**DROP TABLE** 文で明示的に削除するまでデータベースに存在します。

ベーステーブルと同様、データベースに含まれるグローバルテンポラリテーブルの定義は 1 つのみです。ただし、各ユーザはグローバルテンポラリテーブル内に個々にデータのインスタンスを持っています。これらのローは、そのローを挿入した接続だけが参照できます。接続の終了時またはコミット時には削除されます。それぞれの接続は、グローバルテンポラリテーブルのスキーマを、そのテーブルを最初に参照したときの状態のまま継承します。マルチプレックスサーバ上に作成されたグローバルテンポラリテーブルは、他のマルチプレックスサーバ上でも作成されます。

テンポラリテーブルに選択するには、次の構文を使用します。

```
SELECT * INTO #TableTemp FROM lineitem
WHERE l_discount < 0.5
```

注意： SAP Sybase は、テンポラリテーブルに対して **SELECT INTO** を実行するスクリプトを作成するときは、ベースカラムではない select リスト項目を **CAST** 式にラップすることを強くおすすめします。これにより、テンポラリテーブルのカラムのデータ型が目的のデータ型になることが保証されます。

自分の接続専用のローカルテンポラリテーブルを宣言するには、**DECLARE LOCAL TEMPORARY TABLE** 文を使用します。ローカルテンポラリテーブルは、接続の終

了時またはコミット時まで、または宣言した複合文内で存在します。他のユーザはこのテーブルとそのデータにまったくアクセスできません。

その接続に同じ名前のローカルテンポラリテーブルがある場合、ベーステーブルまたはグローバルテンポラリテーブルを作成しようとしても失敗します。これは、新しいテーブルを *owner.table* がユニークに識別できないからです。

ただし、既存のベーステーブルまたはグローバルテンポラリテーブルとしてなら、ローカルテンポラリテーブルを同じ名前で作成できます。テーブル名への参照は、ローカルテンポラリテーブルにアクセスします。ローカルテンポラリテーブルが最初に解決されるからです。

次のシーケンス例を見てみましょう。

```
CREATE TABLE t1 (c1 INT);
INSERT t1 VALUES (9);

DECLARE LOCAL TEMPORARY TABLE t1 (c1 INT);
INSERT t1 VALUES (8);

SELECT * FROM t1;
```

返される結果は 8 です。ローカルテンポラリテーブルが接続により削除されるまで、t1 に対する参照はいずれも、ローカルテンポラリテーブル t1 を参照します。

グローバルテンポラリテーブルの削除と変更

グローバルテンポラリテーブルを削除するには、ベーステーブルの場合と同じように、**DROP TABLE** 文を使用します。

他の接続で使用されている場合は、グローバルテンポラリテーブルの削除や変更はできません。

テーブルの配置

SAP Sybase IQ では、テーブルは現在のデータベース内に作成されます。

SAP Sybase IQ データベースに接続している場合、テーブルは次のように配置されます。

表 7：テーブルの配置

テーブルタイプ	許可されている配置	デフォルトの配置
永久	IQ ストア、カタログストア	IQ ストア
グローバルテンポラリ	IQ テンポラリストア、カタログストア	IQ テンポラリストア
ローカルテンポラリ	IQ テンポラリストアまたはカタログストア。作成ユーザのみが参照可能。	IQ テンポラリストア

IQ テーブルのインデックスの自動作成

特定のカラムについてのインデックス作成を自動化するには、テーブル作成時に **PRIMARY KEY** または **UNIQUE** 制約を使用します。

これらのオプションを指定すると、SAP Sybase IQ はそのカラムの一意性を確保するための **HG** インデックスを作成します。

ALTER TABLE コマンドを使用して、既存のテーブルに **UNIQUE** カラムを追加するか、既存のカラムを **UNIQUE** として指定すると、**HG** インデックスが自動的に作成されます。

記憶領域とクエリパフォーマンスの最適化

永久テーブルを SAP Sybase IQ データベースに作成すると、SAP Sybase IQ は自動的にその永久テーブルを、射影と呼ばれるタイプのクエリを簡単に実行できるようにするデフォルトインデックスに格納します。

SAP Sybase IQ は、クエリパフォーマンスの向上と必要な記憶領域量の削減のために、次の要素に基づいてこの構造を最適化します。

- **IQ UNIQUE** オプション (**CREATE TABLE** またはプラグインの [カラムのプロパティ] ページ)
- カラムのデータ型と幅
- **IQ PAGE SIZE** オプション (**CREATE DATABASE** または SAP Control Center データベース作成ウィザード)

IQ UNIQUE の影響

IQ UNIQUE は、カラムのカーディナリティを定義し、カラムを Flat FP と NBit のどちらとしてロードするかを決定するオプションのカラム制約です。

IQ UNIQUE は、カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(*n*) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT オプションおよび FP_NBIT_LOOKUP_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

- FP_NBIT_AUTOSIZE_LIMIT は、NBit としてロードする、重複しない値の数を制限します。
- FP_NBIT_LOOKUP_MB は、NBit ディクショナリの合計サイズのスレッシュホールドを設定します。
- FP_NBIT_ROLLOVER_MAX_MB は、NBit からフラット FP への暗黙的な NBit ロールオーバーで使用するディクショナリサイズを設定します。

- `FP_NBIT_ENFORCE_LIMITS` は、NBit ディクショナリのサイズ制限を強制します。このオプションはデフォルトで `OFF` になっています。

`IQ UNIQUE` を `FP_NBIT_AUTOSIZE_LIMIT` 未満の n 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が `FP_NBIT_AUTOSIZE_LIMIT` を超えるときにカラムを NBit としてロードする場合は、`IQ UNIQUE` を使用します。

インデックスと `IQ UNIQUE`

`IQ UNIQUE` の見積もりが間違っている場合でも、オプティマイザは次に大きいインデックスを使用するため、ロードに不利益が出ることはありません。

クエリの場合、`IQ UNIQUE` の見積もりを間違えて `HG`、`LF`、または記憶領域を最適化したデフォルトインデックスを作成した場合、オプティマイザは `IQ UNIQUE` の値を無視し、実際のインデックス数を使用します。これらのインデックスを作成せず、さらに見積もりが大幅に間違っている場合は (たとえば、ユニーク値の実際の数が 1200 万個のときに `IQ UNIQUE 1000000` と指定した場合など)、クエリのパフォーマンスが低下することがあります。

既存のインデックスの `IQ UNIQUE` 値を変更するには、`sp_iqrebuildindex` プロシージャを実行します。

`UNIQUE` と `IQ UNIQUE` の相違点

`IQ UNIQUE (n)` は、特定のカラム内に格納できる別個の値の概数を指定します。`IQ UNIQUE (n)` 値を 0 に設定すると、カラムは Flat FP としてロードされます。`IQ UNIQUE` が指定されていないカラム、または `FP_NBIT_AUTOSIZE_LIMIT` および `FP_NBIT_LOOKUP_MB` オプションで定義された限度以下の `IQ UNIQUE (n)` 値が指定されているカラムは、暗黙的に NBit としてロードされます。別個の値はそれぞれ何度でも使用できます。たとえば、`employee` テーブルの `state` カラムで使用できる値は限られていますが、それぞれの値は複数のローで使用できます。

一方、`UNIQUE` または `PRIMARY KEY` を指定した場合は、そのカラム内ではそれぞれの値を一度しか使用できません。たとえば、`employee` テーブルの `ss_number` の各値、つまり従業員の社会保障番号は、そのカラム全体で一度しか使用できません。この一意性は `NULL` 値にも適用されます。したがって、`UNIQUE` と指定したカラムは、`NOT NULL` 制約を持つ必要もあります。

テーブル変更のガイドライン

この項では、**ALTER TABLE** 文を使用してテーブル構造を変更する方法について説明します。

例 1

次に示すコマンドは、スキルの説明を自由に書き込むカラムを skill テーブルに追加します。

```
ALTER TABLE skill
ADD skill_description CHAR( 254 )
```

例 2

次の文は、skill_type カラムの名前を classification に変更します。

```
ALTER TABLE skill
RENAME skill_type TO classification
```

例 3

次の文は、classification カラムを削除します。

```
ALTER TABLE skill
DELETE classification
```

例 4

次の文はテーブル全体の名前を変更します。

```
ALTER TABLE skill
RENAME qualification
```

これらの例は、データベースの構造を変更する方法を示しています。**ALTER TABLE** 文では、外部キーの追加や削除など、多くのテーブル特性を変更できます。ただし、**MODIFY** を使用してテーブルやカラムの制約を変更することはできません。変更するのではなく、**DELETE** で古い制約を削除し、**ADD** で新しい制約を追加する必要があります。いずれの場合も一度変更を行うと、このカラムを参照しているストアプロシージャ、ビュー、およびその他の項目は機能しなくなります。

テーブル削除のガイドライン

テーブルを削除すると、テーブル内のすべてのレコードとテーブル定義が削除されます。

次に示す **DROP TABLE** 文は、skill テーブルにあるすべてのレコードを削除し、データベースから skill テーブルの定義を削除します。

```
DROP TABLE skill
```

CREATE 文と同様、**DROP** 文はテーブルを削除する前と後に **COMMIT** を自動的に実行します。したがって、最後に **COMMIT** または **ROLLBACK** を実行した後の変更がすべて確定されます。

DROP 文では、テーブル上のインデックスもすべて削除されます。

データローだけを削除し、テーブル自体は残す場合は、**TRUNCATE TABLE** 文を使用します。他のユーザが読み込み中のテーブルをトランケートすると、テーブルの通常のバージョン管理ルールが適用されます。つまり、そのユーザのトランザクションが完了するまで、古いテーブルバージョンが保持されます。

DROP TABLE と **TRUNCATE TABLE** 文の実行には数秒しかかかりません。データサイズは操作の速度に影響しません。

プライマリキーの作成

テーブル内の各ローは、このプライマリキーによってユニークに識別されます。

CREATE TABLE 文と **ALTER TABLE** 文では、カラムの制約と検査を含め、さまざまなテーブル属性を設定できます。

プライマリキーの作成

次の文は、前の例と同じ skill テーブルを作成します。ただし、今回はプライマリキーが追加されます。

```
CREATE TABLE skill (  
    skill_id INTEGER NOT NULL,  
    skill_name CHAR( 20 ) NOT NULL,  
    skill_type CHAR( 20 ) NOT NULL,  
    primary key( skill_id )  
)
```

プライマリキーの値は、テーブル内の各ローでユニークでなければなりません。このため今回の例では、特定の skill_id を持つローが複数存在してはなりません。

プライマリキーのカラムに NULL 値を入れることはできません。このため、プライマリキーのカラムには、**NOT NULL** を指定する必要があります。

外部キーの作成

それぞれの外部キー関係では、あるカラム内の候補キー(プライマリキーと一意性制約)を別のテーブル内のカラムに関連付けます(これが外部キーになります)。

たとえば、次のように、従業員のスキルごとのスキルレベルの説明を含む emp_skill という名前のテーブルを作成します。

```
CREATE TABLE emp_skill(  
    emp_id INTEGER NOT NULL,  
    skill_id INTEGER NOT NULL,
```

```
"skill level" INTEGER NOT NULL,  
PRIMARY KEY( emp_id, skill_id ),  
FOREIGN KEY REFERENCES employee,  
FOREIGN KEY REFERENCES skill  
)
```

この emp_skill テーブル定義には、emp_id と skill_id の2つのカラムで構成されるプライマリキーがあります。1人の従業員が複数のスキルを持っている場合は、複数のローに表示されます。また、複数の従業員が特定のスキルを持っている場合は、同じ skill_id が何度か表示されることがあります。

emp_skill テーブルには、2つの外部キーがあります。外部キーエントリは、emp_id カラムに employee テーブルから employee テーブルのプライマリキーである有効な従業員番号が格納されている必要があること、および、skill_id に skill テーブルから skill テーブルのプライマリキーである有効なエントリが格納されている必要があることを示します。

テーブルにはプライマリキーは1つしか定義できませんが、外部キーはいくつでも定義できます。

ローカル、グローバル、自動のどの種類のテンポラリテーブルであっても、外部キー制約を作成することはできません。

有効な文字列と識別子の詳細については、『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』の「SQL 言語の要素」を参照してください。

システムビューに含まれるテーブル情報

データベースのテーブルに関する情報は、SYS.SYSTAB システムビューにあります。

このビューの情報を参照するには、SAP Control Center の [Execute SQL] ウィンドウまたは Interactive SQL を使用します。コマンドウィンドウに次の文を入力すると、SYS.SYSTAB ビューのすべてのカラムが表示されます。

```
SELECT *  
FROM SYS.SYSTAB
```

テーブルパーティション

テーブル分割では、大きなテーブルを、より小さく管理しやすいストレージオブジェクトに分割することによって、パフォーマンスを向上させることができます。

各パーティションは親テーブルと同じ論理属性を共有しますが、別々の DB 領域に配置して個別に管理できます。

注意： 分割されたデータがオフラインの DB 領域にある場合には、テーブルデータにアクセスできません。すべての DB 領域がオンラインであることを確認してください。

参照：

- データ記憶領域の管理 (99 ページ)

制限事項

テーブルパーティションにはいくつかの制限事項があります。

- 範囲パーティションと複合分割スキームは、ハッシュ範囲パーティションと同様、個別にライセンスが必要な VLDB Management オプションを必要とします。
- 分割できるのはベーステーブルのみです。グローバルテンポラリテーブルや宣言したローカルテンポラリテーブルは分割できません。テーブルパーティションのすべてのローは、物理的に同じ場所に配置されます。
- 範囲パーティションまたは範囲サブパーティションは別々の DB 領域に配置できますが、個々の DB 領域は、BLOB カラムまたは CLOB カラム、あるいは 255 バイトよりも大きな CHAR カラム、VARCHAR カラム、または VARBINARY カラムのみを配置することをお奨めします。

パーティションキー

パーティションキーに以下のものを含めることはできません。

- LOB(BLOB カラムまたは CLOB) カラム
- BINARY カラム、または VARBINARY カラム
- 長さが 255 バイトを超える CHAR カラムまたは VARCHAR カラム
- BIT カラム
- FLOAT/DOUBLE/REAL カラム

パーティションキーカラムには、NULL 値と DEFAULT 値が含まれます。NULL 値はすべて、同じパーティションにマッピングされます。

ハッシュパーティションにはパーティションキーカラムを 8 つまで、範囲パーティションまたは範囲サブパーティションにはキーカラムを 1 つのみ含めることができます。

範囲パーティションまたは範囲サブパーティションは、1024 個まで定義できません。

DML 操作

LOAD、INSERT、DELETE、TRUNCATE、TRUNCATE TABLE PARTITION などの DML 操作を実行できます。パーティションキーカラムの UPDATE はサポートされていないため、エラーが発生します。分割されたテーブルの他のすべてのカラムの UPDATE を実行できます。

SAP Sybase IQ は、READ-ONLY テーブルまたは READ-ONLY テーブルパーティションでの DML 操作に対して例外を生成します。特定のローが指定したパーティションの範囲に適合しない場合、INSERT 文と LOAD 文または更新可能カーソルによる INSERT はエラーを生成し、操作はロールバックされます。

DDL 操作

DROP、RENAME、PARTITION、UNPARTITION、MERGE、SPLIT、および MOVE のパーティションには、いくつかの制限事項があります。

表 8 : パーティションでの DDL 操作の制限事項

操作	制限事項
削除	パーティションキーまたは範囲サブパーティションキーからカラムを削除することはできない。分割されたテーブルの最後のパーティションは削除できない。
分割されていないテーブルの分割	既存のテーブルではハッシュ分割できない。
隣接する 2 つのパーティションのマージ	両方のパーティションが同じ DB 領域に存在している必要がある。データを移動する必要はない。
パーティションの分割	分割後、すべてのローが 2 つのパーティションの最初のパーティションに属する必要がある。分割されたパーティションは、元の DB 領域と同じ DB 領域にあるため、データを移動する必要はない。 範囲サブパーティションはデータを移動しない場合にかぎり、分割できる。分割されるサブパーティションのすべての既存のローは、分割後に最初のサブパーティションに残す必要がある。
新しい DB 領域へのパーティションの移動	パーティションのすべてのローが新しい DB 領域のデータページに移動される。新しい DB 領域で CREATE パーミッションが必要である。

参照 :

- 範囲パーティション (161 ページ)

- ハッシュパーティション (162 ページ)
- ハッシュ範囲パーティション (163 ページ)
- ALTER TABLE 文 (459 ページ)
- CREATE TABLE 文 (511 ページ)

範囲パーティション

範囲分割では、各パーティションに対して構築している *partition-key* 値の範囲に基づいて大きなテーブルを分割します。

情報ライフサイクル管理戦略の一環として、範囲分割はバックアップ時間やリストア時間を短縮し、より細かなレベルで詳細なデータ検証を行い、多層ストレージをサポートできます。

range-partitioning-scheme の *partition-key* は、ローが属するパーティションを指定する値を含むカラムです。

```
range-partitioning-scheme:
    PARTITION BY RANGE ( partition-key ) ( range-partition-decl [ ,
    range-partition-decl ... ] )
```

range-partition-declaration は、指定されたパーティションを DB 領域に配置する方法を指定します。

```
range-partition-declaration:
    range-partition-name VALUES <= ( {constant | MAX } ) [ IN
    dbspace-name ]
```

VALUES 句は、各パーティションの上限を (昇順に) 指定します。各範囲パーティションは、独自の DB 領域に配置して、個々に管理できます。パーティション名は、テーブル上にあるパーティションセット内でユニークである必要があります。

制限事項

範囲分割は、単一の分割キーカラムおよび最大 1024 パーティションまでに制限されています。

例

範囲分割では、日付ごとにテーブルを分割するのが最も一般的な形式です。この例では、6つのカラムと3つのパーティションを含む、範囲分割されたテーブル *bar* を作成し、データを日付に基づいたパーティションにマッピングします。

```
CREATE TABLE bar (
    c1 INT IQ UNIQUE(65500),
    c2 VARCHAR(20),
    c3 CLOB PARTITION (P1 IN Dsp11, P2 IN Dsp12,
        P3 IN Dsp13),
    c4 DATE,
    c5 BIGINT,
    c6 VARCHAR(500) PARTITION (P1 IN Dsp21,
```

```
P2 IN Dsp22),
PRIMARY KEY (c5) IN Dsp2) IN Dsp1
PARTITION BY RANGE (c4)
(P1 VALUES <= ('2006/03/31') IN Dsp31,
P2 VALUES <= ('2006/06/30') IN Dsp32,
P3 VALUES <= ('2006/09/30') IN Dsp33);
```

参照：

- 制限事項 (159 ページ)
- ハッシュパーティション (162 ページ)
- ハッシュ範囲パーティション (163 ページ)
- ALTER TABLE 文 (459 ページ)
- CREATE TABLE 文 (511 ページ)

ハッシュパーティション

ハッシュ分割では、内部ハッシュ関数によって処理された *partition-key* の値に基づいて、データをパーティションにマップします。

ハッシュ分割では、データが論理パーティションに分散されて、並列実行の対象となります。これによって、大きなテーブルおよび分散されたクエリ (DQP) のジョインパフォーマンスが向上します。

hash-partitioning-scheme 宣言内の *partition-key* は、1つのカラムまたはカラムのグループです。その複合値によってデータの各ローが格納されるパーティションが決まります。

hash-partitioning-scheme:

```
PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
```

ハッシュパーティションキーは最大 8 カラムで、組み合わせた宣言カラム幅が 5300 バイト以下に制限されています。ハッシュパーティションの場合、テーブル作成者はパーティションキーカラムのみを決定します。パーティションの数と位置は内部的に決定されます。

制限事項

- ハッシュパーティションの追加、削除、マージ、分割はできません。
- カラムをハッシュ分割キーから追加または削除することはできません。

例

この例では、テーブル `tbl142` には、PRIMARY KEY (カラム `c1`) と HASH PARTITION KEY (カラム `c4` および `c3`) が含まれます。

```
CREATE TABLE tbl142 (
  c1 BIGINT NOT NULL,
  c2 CHAR(2) IQ UNIQUE(50),
  c3 DATE IQ UNIQUE(36524),
```

```
c4 VARCHAR(200),
PRIMARY KEY (c1)
PARTITION BY HASH ( c4, c3 )
)
```

ここでは、プライマリキーとハッシュパーティションキーの両方がジョインキーである一般的な例を示します。

```
CREATE TABLE bar (
  c1 BIGINT NOT NULL,
  c2 CHAR(2) IQ UNIQUE(50),
  c3 DATE IQ UNIQUE(36524),
  c4 VARCHAR(200),
  c5 CLOB,
  PRIMARY KEY ( c1 )
  PARTITION BY HASH ( c1 )
)
```

参照：

- 制限事項 (159 ページ)
- 範囲パーティション (161 ページ)
- ハッシュ範囲パーティション (163 ページ)
- ALTER TABLE 文 (459 ページ)
- CREATE TABLE 文 (511 ページ)

ハッシュ範囲パーティション

ハッシュ範囲分割は、ハッシュ分割されたテーブルを、範囲に基づいてさらに分割する複合分割スキーマです。

ハッシュ範囲分割には、ハッシュ分割と範囲分割の両方の利点があります。ハッシュ分割では、分散クエリのパフォーマンスが最適なものとなり、範囲サブ分割では、情報ライフサイクル管理戦略の一環として管理タスクが拡張されます。

hash-range-partitioning-scheme 宣言内の **SUBPARTITION BY RANGE** 句は、新しい範囲サブパーティションを既存のハッシュ範囲分割テーブルに追加します。

hash-range-partitioning-scheme:

```
PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
[ SUBPARTITION BY RANGE
  ( range-partition-decl [ , range-partition-decl ... ] ) ]
```

ハッシュパーティションはデータの論理的な配分および配置方法を指定するのに対して、範囲サブパーティションはデータの物理的な配置方法を指定します。新しい範囲サブパーティションは、既存のハッシュ範囲分割テーブルと同じハッシュ分割キーを持つハッシュによって論理的に分割されます。範囲サブパーティションキーは1つのカラムに制限されています。

例

この例 (テーブル tbl142) には、PRIMARY KEY (カラム c1) とハッシュパーティションキー (カラム c4 および c2)、および範囲サブパーティションキー (カラム c3) が含まれています。

```
CREATE TABLE tbl142 (  
  c1 bigint not null,  
  c2 char(2) iq unique(50),  
  c3 date  
  c4 varchar(200),  
  PRIMARY KEY (c1)) IN Dsp1  
  PARTITION BY HASH (c4, c2)  
  SUBPARTITION BY RANGE (c3)  
  (P1 VALUES <= ('2011/03/31') IN Dsp31,  
  P2 VALUES <= ('2011/06/30') IN Dsp32,  
  P3 VALUES <= ('2011/09/30') IN Dsp33);
```

この例では、テーブル tbl143 には、カラム c4 の RANGE サブパーティションの記憶領域として異なる DB 領域があります。この構文は、RANGE パーティションの既存の構文に似ています。カラム c4 の SUBPARTITION の使用に注意してください。

```
CREATE TABLE tbl143 (  
  c1 bigint not null,  
  c2 char(2) iq unique(50),  
  c3 date  
  c4 varchar(200)  
  SUBPARTITION (P1 in Dsp331, P2 in Dsp332, P3 in Dsp333),  
  PRIMARY KEY (c1)) IN Dsp1  
  PARTITION BY HASH (c4, c2)  
  SUBPARTITION BY RANGE (c3)  
  (P1 VALUES <= ('2011/03/31') IN Dsp31,  
  P2 VALUES <= ('2011/06/30') IN Dsp32,  
  P3 VALUES <= ('2011/09/30') IN Dsp33) ;
```

参照：

- 制限事項 (159 ページ)
- 範囲パーティション (161 ページ)
- ハッシュパーティション (162 ページ)
- ALTER TABLE 文 (459 ページ)
- CREATE TABLE 文 (511 ページ)

SAP Sybase IQ のインデックスデータ

SAP Sybase IQ インデックスは、特定のタイプのクエリに対してメモリ、ディスクおよび CPU サイクルを最大限活用するよう設計されています。

- 通常、インデックスのサイズを小さく維持できます。データベース全体が完全にインデックス化され、アドホッククエリで使用可能になります。格納に要する領域は、生データの場合と変わりません。従来のデータベースでは、たいてい約 3 倍の領域が必要になります。
- クエリは、関連するカラム上だけでインデックスを効率的に組み合わせて操作することによって解決されます。これにより、時間のかかるテーブルスキャンを行う必要がなくなります。
- I/O が最小化され、潜在的なボトルネックがなくなります。
- インデックスが小型なので、その後のクエリのためにより多くのデータをメモリに保管でき、したがって反復的な分析のスループットが向上します。
- チューニングはデータに依存するため、アドホッククエリを何度行う場合でも、一度データを最適化するだけで済みます。

インデックスの概要

インデックスによってデータ取得のパフォーマンスが向上します。SAP Sybase IQ のインデックスは、従来のインデックスとは異なり、データウェアハウス環境におけるクエリを高速化するように設計されています。

テーブルにデータをロードすると、SAP Sybase IQ はそのデータをテーブル内の各カラムにローではなくカラム単位で物理的に格納します。カラム単位にすることで、IQ インデックスには従来のローベースのインデックスにはない大きな利点が生まれています。

SAP Sybase IQ がカラムごとに作成するデフォルトのカラム格納構造は、データの保管および射影のために最適化されたインデックスです。定義したカラムインデックスは、個々のテーブルの一部として作成されます。

論理的には、従来のローベースの SQL データベースと同様にデータにアクセスできます。カラムで保管することにより、追跡する属性に従ってデータが構築されます。データウェアハウス環境では通常、数千から数百万ローに及ぶデータから処理の対象となるのは特定の属性であって、完全なロー単位のデータではありません。そのため、カラム単位で保管することにより、特定の属性の選択や計算を実行する機能が最適化されます。

インデックスタイプ比較

タスクに適したカラムインデックスタイプを選択します。

表 9：インデックスタイプ比較

インデックスタイプ	目的
Compare (CMP) インデックス (169 ページ)	同じデータ型、精度、位取りの2つの異なるカラム間の関係をインデックス化する。
包含 (WD) インデックス (167 ページ)	CHAR データ、VARCHAR データ、または LONG VARCHAR データのカラム文字列から単語を格納する。
日付 (DATE) インデックスおよび日時 (DTTM) インデックス (170 ページ)	日付量または日時量を扱うクエリを処理する。
高速射影 (FP) インデックスデフォルトでは、データベースエンジンによって自動的にすべてのカラムに高速射影 (FP) インデックスが作成されます。FP インデックスは射影を最適化し、データベースエンジンが特定の種類の検索条件を評価できるようにします。	射影を最適化し、特定の種類の検索条件を評価可能とする。デフォルトでは永久テーブルに割り当てられる。
High_Group (HG) インデックス (176 ページ)	integer データ型のジョインカラムで使用する。GROUP BY と等号演算を効率的に処理する。独自の値が 1,000 個以上のカラムに対して推奨される。
High_Non_Group (HNG) インデックス (178 ページ)	範囲や集約関数を含む高カーディナリティの演算に対して使用される。
Low_Fast (LF) インデックス (179 ページ)	Yes/No、扶養家族の人数などユニークな値が 1,000 未満のカラム、または SUM、AVG、および COUNT のカラムに対して理想的である。SAP Sybase IQ では最速のインデックスタイプである。
TEXT インデックス (181 ページ)	非構造化データ分析機能に対して使用される。SAP Sybase IQ ライセンスオプション。

インデックスタイプ	目的
時刻 (TIME) インデックス (181 ページ)	時間を扱うクエリを処理する。

包含 (WD) インデックス

包含 (WD) インデックスを使用すると、CHAR、VARCHAR、LONG VARCHAR データ型のカラム文字列に含まれる単語を格納できます。

注意： LONG VARCHAR カラムを作成するには、非構造化データ分析機能を使用する正規のライセンスを取得している必要があります。『ユーザ定義関数』を参照してください。

WD の推奨される使用法

WD インデックスは、キーワードのリストが格納されたカラムへの高速アクセスが必要な場合に使用します (たとえば図書目録レコードや Web ページなど)。

WD インデックスには、次の制限が適用されます。

- UNIQUE 属性を指定できません。
- CONTAINS または LIKE 述部でのみ使用できます。
- カラム名には、ベーステーブルの CHAR、VARCHAR、または LONG VARCHAR のカラムを指定する必要があります。
- 最小許容カラム幅は 3 バイトで、最大許容カラム幅は LOB カラムの最大幅です (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)。
- デリミタのリストは一重引用符で囲む必要があります。
- DELIMITED BY 句を省略するか、*separators-string* 値を空 (一重引用符) で指定すると、SAP Sybase IQ ではデフォルトのセパレータセットが使用されます。デフォルトの文字セットには、7 ビットの ASCII 文字のすべて (7 ビットの ASCII 英数字ではありません) が含まれ、デフォルトで単語の一部とみなされるハイフンと一重引用符は含まれません。デフォルトのセパレータのセットには 64 文字が含まれます。
- DELIMITED BY 句と LIMIT 句を複数指定した場合は、エラーにはなりませんが、各タイプの最後に指定した句だけが使用されます。たとえば、次の 2 つの文は同じ結果を返します。

文 1：

```
CREATE WD INDEX c1wd on foo(c1) DELIMITED BY 'f' LIMIT 40 LIMIT 99
DELIMITED BY 'g' DELIMITED BY 'h';
```

文 2：

```
CREATE WD INDEX c1wd on foo(c1) DELIMITED BY 'h' LIMIT 99;
```

- WD** インデックスを作成すると、そのカラムへ挿入されるデータはすべて、セパレータと最大ワードサイズで解析されます。このセパレータと最大ワードサイズは、インデックスの作成後に変更できません。

CHAR カラムの場合は、少なくともスペースをセパレータの 1 つに指定するか、デフォルトのセパレータセットを使用します。SAP Sybase IQ では、CHAR カラムに最大カラム幅になるまで空白が埋め込まれます。カラム内に文字データの他に空白が含まれていると、**WD** インデックス付きのデータに対して発行したクエリから返される結果が不正確になることがあります。たとえば、`company_name` というカラムにセパレータで区切られた 2 つの単語が含まれているとします。ただし、2 つ目の単語には次のようにブランクが埋め込まれています。

```
'Concord' 'Farms'
```

このとき、ユーザが次のクエリを入力したとします。

```
SELECT COUNT(*) FROM Customers WHERE CompanyName contains ('Farms')
```

パーサはこの文字列を次のように解釈します。

```
'Farms'
```

次の解釈はされません。

```
'Farms'
```

返される値は 1 です。この問題を防ぐには、CHAR カラムではなく VARCHAR カラムを使用します。

- sp_iqcheckdb** (DBCC 一貫性チェック) の `allocation`、`check`、`verify`、`repair` の各モードは、CHAR、VARCHAR、LONG VARCHAR のカラムの **WD** インデックスをサポートします。

WD の長所と短所

次の表に、**WD** インデックスを使用する上での長所と短所を示します。

表 10 : **WD** の長所と短所

長所	短所
<p>大きなロードでは大幅なパフォーマンス向上が期待できる。</p> <p>このインデックスを使用すると一部の LIKE 述部を高速に実行できる。</p> <p>このインデックスを使用する CONTAINS 述部は LIKE 述部よりも優先される。</p> <p>キーワードや URL の一部にインデックスを付ける最善の方法。</p>	<p>必要なディスク領域の量が非常に大きくなる可能性がある。</p> <p>一意性を確保すると、インデックスが使用不可になる。</p> <p>カラム内のデータが CHAR、VARCHAR、または LONG VARCHAR の場合のみ使用可能。</p>

Compare (CMP) インデックス

Compare (CMP) インデックスは、2つのカラム間の関係です。Compare インデックスは、同じデータ型、精度、位取りを持つ2つの異なるカラムに対して作成できます。**CMP** インデックスには2つのカラムのバイナリ比較(<、>、または=)が格納されます。

CMP の推奨される使用法

CMP インデックスは、NULL、NOT NULL、または混合のカラムに対して作成できます。**CMP** インデックスはユニークにできません。

数値データ型と10進数データ型は同一とみなされます。**CMP** インデックスは、精度と位取りが等しいカラムに対して作成できます。CHAR、VARCHAR、BINARY、VARBINARY カラムにおける精度とは、カラム幅が等しいことを指します。

たとえば、次のコマンドはテーブルを作成し、次に適切な **CMP** インデックスを作成します。

```
CREATE TABLE f(c1 INT NOT NULL, c2 INT NULL, c3 CHAR(5), c4 CHAR(5))
CREATE CMP INDEX c1c2cmp ON f(c1, c2)
```

次のインデックスは、インデックス対象のカラムのデータ型、精度、位取りが等しくないので無効になります。

```
CREATE CMP INDEX c1c3cmp ON f(c1, c3)
```

CMP インデックスの制限

CMP には、以下の制限が適用されます。

- **CMP** インデックスは削除できます。
- **CMP** インデックスはユニークにできません。
- **CMP** インデックス内に定義されているカラムを変更または削除しようとすると、例外が発生します。
- **CMP** インデックス内に定義されている既存のカラムに **ALTER TABLE MODIFY** を発行することはできません。
- **CMP** インデックスは、BIT、FLOAT、DOUBLE、REAL データ型をサポートしません。

日付 (DATE) インデックスおよび日時 (DTTM) インデックス

DATE および **DTTM** インデックスタイプは、日付または日時の数量が関係するクエリを処理します。

- **DATE** インデックスは、データ型 **DATE** のカラムで、日付の数量が関係するクエリの処理に使用されます。
- **DTTM** インデックスは、データ型 **DATETIME** または **TIMESTAMP** のカラムで日時の数量が関係するクエリの処理に使用されます。

DATE および DTTM インデックスタイプの推奨される使用法

DATE インデックス、または **DTTM** インデックスは、**DATE**、**DATETIME**、または **TIMESTAMP** のカラムが、日付および時刻の関数と操作を含むクエリで使用されるときに使用されます。

- **DATEPART** 等号述部 (=、!=)、**DATEPART** 範囲述部 (>、<、>=、<=、!>、!<、**BETWEEN**)、**DATEPART IN** リスト述部を含むクエリ
- 範囲述部 (>、<、>=、<=、**BETWEEN**) を含むクエリ

注意： **DATE**、**DATETIME**、または **TIMESTAMP** カラムを使用する単純な等号述部 (**DATEPART** を含まない) の場合は、**LF** と **HG** インデックスを使用すると最高のパフォーマンスが得られます。**LF** または **HG** インデックスを使用できない場合は、**DATE** または **DTTM** インデックスを使用して結果を取得します。

DATE、**DATETIME**、または **TIMESTAMP** カラムが、等号 (ジョイン条件を含む) または **IN** 述部の **GROUP BY** 句か **WHERE/HAVING** 句で使用されている場合は、カラムに **LF** または **HG** のインデックスが必要です。これは、これらのインデックスのみが高速等号探索を実行できるためです。

この項の例で使用する **tab** テーブルには、次のように定義されたカラムが含まれています。

```
CREATE TABLE tab
(col1 DATE,
 col2 DATETIME,
 col3 TIME);
```

DATEPART 等号述部、範囲述部、IN リスト述部を含むクエリ

等号述部 (= または !=) を含むクエリでは、比較の一方が **DATEPART** 式またはその他の日時関数 (たとえば **YEAR**、**QUARTER**、**DAY**、**MINUTE**) で、もう一方が定数式 (定数値またはホスト変数を含む) である場合は、**DATE**、**TIME**、または **DTTM** インデックスを使用して結果セットを取得します (インデックスが使用可能な場合)。

たとえば次のクエリでは **DATE**、**TIME**、または **DTTM** インデックスが使用されません。

```
SELECT * FROM tab WHERE DATEPART(YEAR, col1) = 2002;
SELECT * FROM tab WHERE DATEPART(HOUR, col2) = 20;
SELECT * FROM tab WHERE MINUTE (col3) != 30;
SELECT * FROM tab WHERE DATEPART(MONTH, col2) = @tmon;
```

上記の @tmon は INTEGER 型のホスト変数です。

DATE、TIME、および DTTM インデックスを使用する処理に適した DATEPART 範囲と IN リスト述部条件は次のとおりです。

- 比較条件 >, <, >=, <=, !=, !<

比較の一方が、テーブルカラムまたはビューカラムをパラメータとする日時関数または DATEPART 関数です。比較のもう一方は定数式です (整数または整数型のホスト変数など)。次に例を示します。

```
DATEPART(WEEK, col1) != 23
```

```
DATEPART(YEAR, col1) = 2001
```

```
HOUR(col3) >= 1
```

- BETWEEN ... AND 条件

BETWEEN の左側が、テーブルカラムまたはビューカラムをパラメータとする日時関数または DATEPART 関数です。AND の両側は定数式です (整数または整数型のホスト変数など)。次に例を示します。

```
DATEPART(YEAR, col1) BETWEEN host-var1 AND host-var2
```

- IN 条件

IN の左側は、テーブルカラムまたはビューカラムをパラメータとする日時関数または DATEPART 関数です。IN リスト内の値は定数式です。次に例を示します。

```
DATEPART(MONTH, col1) IN (1999, 2001, 2003)
```

注意：DATE、TIME、DTTM インデックスは、一部の日付単位 (Calyearofweek、Calweekofyear、Caldayofweek、Dayofyear、Millisecond) をサポートしていません。次に例を示します。

```
SELECT * FROM tab WHERE DATEPART(MILLISECOND, col3)
= 100;
```

```
SELECT * FROM tab WHERE DATEPART(DAYOFYEAR, col1) <= 89;
```

この場合は、クエリオプティマイザは他のインデックスを使用して結果を取得します。

範囲述部を含むクエリ

範囲述部を含むクエリで、述部カラムのインデックスがどのように使用されるかを確認します。

次のような範囲述部を含む場合、**DATE**、**TIME**、または **DTTM** インデックスがクエリの処理に選択されます。

- 比較条件:

```
SELECT * FROM tab WHERE col1 < '2002/10/09';
SELECT * FROM tab WHERE col2 >= '2002/01/01 09:12:04.006';
```

比較の一方は、カラム名で、もう一方は定数式(定数値とホスト変数)です。

- **BETWEEN** 条件:

```
SELECT * FROM tab WHERE col3 BETWEEN '09:12:04.006' AND
'20:12:04.006';
SELECT * FROM tab WHERE col2 BETWEEN tmp_datetime1 AND
tmp_datetime2;
```

これらのクエリの場合は、通常、**DATE**、**TIME**、または **DTTM** インデックスが **HNG** インデックスよりも高速です。

次の3つの特殊な例では、**DATE** または **DTTM** インデックスを使用すると大幅なパフォーマンス向上が期待できます。

- 述部の範囲が完全な年単位です(実開始日が年始で、実終了日が年末です)。次に例を示します。

```
SELECT * FROM tab WHERE col1 BETWEEN '1993-01-01' AND
'1996-12-31';

SELECT * FROM tab WHERE col1 >= '1993-01-01' AND
col1 < '1997-01-01';

SELECT * FROM tab WHERE col2 BETWEEN '1993-01-01 00:00:00.000000'
AND '1996-12-31 23:59:59.999999';
```

- 述部の範囲が同一年内の完全な月単位です(実開始日が月始で、実終了日が月の月末です)。次に例を示します。

```
SELECT * FROM tab WHERE col1 > '1993-01-31' AND
col1 <= '1993-06-31';

SELECT * FROM tab WHERE col2 >= '1993-01-01 00:00:00.000000' AND
col1 < '1993-06-01 00:00:00.000000';
```

- 述部の範囲が完全な1日です。次に例を示します。

```
SELECT * FROM tab WHERE col2 >= '1993-01-31 00:00:00.000000' AND
col2 <= '1993-01-31 23:59:59.999999';
```

注意: 上記の3つの例では、年の範囲、月の範囲、1日の範囲という概念に注意する必要があります。たとえば、**DTTM** インデックスは次の4パターンを年の範囲として認識します。

```
col2 > 'year1/12/31 23:59:59.999999' and
col2 < 'year2/01/01 00:00:00.000000'

col2 >= 'year1/01/01 00:00:00.000000' and
col2 < 'year2/01/01 00:00:00.000000'

col2 > 'year1/12/31 23:59:59.999999' and
col2 <= 'year2/12/31 23:59:59.999999'

col2 >= 'year1/01/01 00:00:00.000000' and
col2 <= 'year2/12/31 23:59:59.999999'
```

次のような範囲は年の範囲に一致しません。

```
col2 > 'year1/12/31 23:59:59.999999' and
col2 <= 'year2/01/01 00:00:00.000000'

col2 > 'year1/01/01 00:00:00.000000' and
col2 < 'year2/01/01 00:00:00.000000'
```

1つめの範囲は、正しい年の範囲に 'year2/01/01 00:00:00.000000' の値が加わっているため一致しません。2つめの範囲では、'year1/01/01 00:00:00.000000' の値が足りません。

月の範囲と1日の範囲に関しても、**DTTM** インデックスと **DATE** インデックスの両方に対して同様の規則が適用されます。

上記3つの特殊な状況に該当しない小さな日付範囲 (値の数が 60 未満) の場合は、**DATE** インデックスよりも **LF** と **HG** インデックスの方が高速です。

DATE/DTTM の長所と短所

次の表に、**DATE** または **DTTM** インデックス使用の長所と短所を示します。

表 11 : DATE/TIME/DTTM の長所/短所

長所	短所
<p>日付、時刻、日時を扱うクエリを他のインデックスタイプより高速に処理できる。</p> <p>DATE または DTTM インデックスは作成したり、削除したりできる。</p>	<p>HNG インデックスよりも多くのディスク領域を使用する。</p> <p>高速な等号探索を行うには LF または HG インデックスが必要である。</p> <p>カラム内のデータが DATE、DATETIME、または TIMESTAMP データ型の場合のみこれらのインデックスを使用できる。</p>

DATE/DTTM インデックスの制限

DATE および **DTTM** インデックスには次の制限が適用されます。

- **UNIQUE** キーワードを使用できない。
- 単一カラムに対してのみ作成できる。
- **Calyearofweek**、**Calweekofyear**、**Caldayofweek**、**Dayofyear**、**Millisecond** の日付要素をサポートしていない。

他のインデックスとの比較

DATE、**TIME**、**DTTM** インデックスのパフォーマンスは、**HNG** インデックスのパフォーマンスと同等です。

HNG と比較して、サポートされる状況では一般に **DATE**、**TIME**、および **DTTM** インデックスは、**HNG** よりも最大で 2 倍高速です。したがって、**DATE**、**TIME**、**DATETIME**、または **TIMESTAMP** データ型のカラムでは、**DATE**、**TIME**、または **DTTM** インデックスに加えて、**HNG** インデックスを作成する必要はありません。

DATE/DTTM カラムのその他のインデックス

DATE、**TIME**、**DATETIME**、または **TIMESTAMP** データ型のカラムを **WHERE** 句、**ON** 条件、**GROUP BY** 句で参照する場合には、それらのカラムで必ず **DATE**、**TIME**、または **DTTM** インデックスを使用します。

HG または **LF** インデックスは、**DATE**、**TIME**、**DATETIME**、または **TIMESTAMP** カラムで、特にカラムに対する等号述部を評価する場合にも適しています。また、カラムを **GROUP BY** 句で頻繁に使用し、かつ、そのカラムのユニークな値の数が 1000 未満である場合 (つまり 3 年分の日数より少ない場合) には、**LF** インデックスもおすすめします。

高速射影 (FP) インデックス

デフォルトでは、データベースエンジンによって自動的にすべてのカラムに高速射影 (**FP**) インデックスが作成されます。**FP** インデックスは射影を最適化し、データベースエンジンが特定の種類の検索条件を評価できるようにします。

FP インデックスは n 個の固定長エントリの配列です。 n はテーブル内のローの数です。カラムに適用される **IQ UNIQUE** 記憶域ディレクティブは、カラムをフラット **FP** と **NBit** のどちらとしてロードするかを決定します。**Flat FP** インデックスには実際のカラムのセルの値が含まれます。**NBit** は圧縮スキームで、データが格納されている辞書のインデックス作成に n ビットを使用します。**LOB** データ型 (文字型とバイナリの両方) と **BIT** データ型を除くすべてのデータ型は **NBit** カラムにすることができます。

IQ UNIQUE

IQ UNIQUE は、カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(*n*) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT オプションおよび FP_NBIT_LOOKUP_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

- FP_NBIT_AUTOSIZE_LIMIT は、NBit としてロードする、重複しない値の数を制限します。
- FP_NBIT_LOOKUP_MB は、NBit デクシヨナリの合計サイズのスレッシュホールドを設定します。
- FP_NBIT_ROLLOVER_MAX_MB は、NBit からフラット FP への暗黙的な NBit ロールオーバーで使用するデクシヨナリサイズを設定します。
- FP_NBIT_ENFORCE_LIMITS は、NBit デクシヨナリのサイズ制限を強制します。このオプションはデフォルトで OFF になっています。

IQ UNIQUE を FP_NBIT_AUTOSIZE_LIMIT 未満の *n* 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が FP_NBIT_AUTOSIZE_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

FP カラムインデックスの再構築

sp_iqindexmetadata は、FP インデックスタイプ (Flat FP または NBIT)、個別カウント、IQ UNIQUE *n* 値、デクシヨナリサイズなど、カラムインデックスに関する詳細を表示します。

sp_iqrebuildindex は Flat FP を NBIT として、または NBIT を Flat FP として再構築します。index_clause は、IQ UNIQUE *n* を '0' (NBIT カラムを Flat FP に再キャスト) から FP_NBIT_AUTOSIZE_LIMIT および FP_NBIT_LOOKUP_MB オプションで定義されている制限値までの間の明示的な値にリセットできます。カウントが *n* の値を超え、FP_NBIT_ENFORCE_LIMITS=ON である場合、操作はロールバックされます。FP_NBIT_ENFORCE_LIMITS=OFF (デフォルト) である場合、NBIT デクシヨナリは増え続けます。

その他の情報

- 『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」 > 「FP_NBIT_AUTO_LIMIT」

- 『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」 > 「FP_NBIT_LOOKUP_MB」
- 『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」 > 「FP_NBIT_ROLLOVER_MAX_MB」
- 『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」 > 「FP_NBIT_ENFORCE_LIMIT」
- 『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』 > 「システムプロシージャ」 > 「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」 > 「sp_iqrebuildindex」
- 『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』 > 「システムプロシージャ」 > 「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」 > 「sp_iqindexmetadata」

High_Group (HG) インデックス

High_Group インデックスは一般的に、integer データ型のジョインカラムで使用します。また High_Group インデックスは **GROUP BY** を効率的に処理するため、High_Non_Group よりも一般的に使用されます。

HG の推奨される使用法

HG インデックスとマルチカラム HG インデックスは、異なる機能を提供します。どちらを使用するか選択するときに、以下の点を考慮してください。

HG インデックスは次の場合に使用します。

- カラムをジョイン述部で使用する。
- カラムに含まれるユニークな値の数が 1000 を超える。

マルチカラム **HG** インデックスを使用して、複数のカラムへの参照がある **ORDER BY** クエリのパフォーマンスを強化します。この変更により、ユーザが意識することなく、クエリパフォーマンスが向上します。

注意： 外部キーカラムには、独自の個別 **HG** インデックスが必要です。

High_Group の長所と短所

High_Group インデックスには、次のような長所と短所があります。

表 12 : HG の長所と短所

長所	短所
GROUP BY を使用するクエリを高速で処理。	<p>HNG インデックスに比べて多くのディスク容量が必要 (未加工データの約 3 倍の領域を使用する場合がある)。</p> <p>データの格納や削除に要する期間が最長。</p> <p>カラム内のデータが BIT、255 バイト超の VARBINARY、255 バイト超の CHAR、または 255 バイト超の VARCHAR の場合は使用不可。</p> <p>FLOAT、REAL、DOUBLE のデータにこのインデックスは推奨されない。</p>

他のインデックスとの比較

HG インデックスと LF および HNG インデックスの比較

LF

決定要因となるのは、ユニークな値の数です。カラムのユニークな値の数が多い場合は、High_Group を使用します。ユニークな値の数が少ない場合には Low_Fast を使用します。

HNG

決定要因となるのは、カラムがジョインカラムであるか、カラムに対して **GROUP BY** が処理されるかどうかです。このいずれかに該当する場合は、High_Group を使用します。単体でも、High_Non_Group と組み合わせてもかまいません。いずれにも該当しない場合は、High_Non_Group を使用してディスク領域を節約します。

HG カラムのその他のインデックス

場合によっては、High_Group インデックスの条件に適合するカラムが、別のインデックスタイプの方が速いクエリの中で使用されることもあります。この場合は、そのカラムに追加インデックスを作成します。

High_Group インデックスの自動作成

SAP Sybase IQ では、インデックスタイプを指定せずに **CREATE INDEX** 文を発行すると、常にデフォルトで High_Group インデックスが作成されます。

SAP Sybase IQ では、**UNIQUE**、**FOREIGN KEY**、または **PRIMARY KEY** 制約に対して High_Group インデックスが自動で作成されます。単一カラム外部キーに対して、SAP Sybase IQ は非ユニークなシングルカラム High_Group インデックスを作成します。マルチカラム外部キーに対しては、非ユニークな複合 High_Group インデック

スが暗黙的に作成されます。非ユニークな HG インデックスでは重複する値が許可され、オプションで NULL も許可されます。このインデックスは参照整合性の基盤となり、クエリパフォーマンスの向上のために使用されます。

SAP Sybase IQ では、カラム定義で NULL 値の使用が許可され、制約 (プライマリキーまたは一意性) が適用されていない場合に、ユーザが作成したユニークなマルチカラム HG インデックスのデータ値で NULL を使用できます。マルチカラムインデックスの詳細については、『リファレンス：文とオプション』の「CREATE INDEX 文」を参照してください。

複数のカラムに対するジョインまたはマルチカラムの GROUP BY 句を含むクエリは、このインデックスによってパフォーマンスが向上します。これは、ユニークでない複合 High Group インデックスが、ジョインおよび結果サイズのカーディナリティをより正確に見積もるからです。さらに、このインデックスはプッシュダウンとサブクエリも最適化します。

High_Non_Group (HNG) インデックス

範囲検索には、HNG インデックスを追加します。

HNG インデックスで必要となるディスク領域は、HG インデックスで必要なディスク領域の約 1/3 です。この点だけを考えても、group 操作が必要でない場合は、HG ではなく HNG インデックスを使用してください。

一方、HG インデックスのほうが効率的に処理できると判明しているクエリを実行する場合、またはカラムがジョインの一部であるか一意性を確保する必要がある場合は、HG インデックスを使用します。

注意： HG インデックスの代わりに HNG インデックスを使用すると、4 つ以上のテーブルを結合する複雑なアドホッククエリのパフォーマンスが大幅に低下する場合があります。そのようなクエリのパフォーマンスが重要となるアプリケーションでは、HG と HNG の両方を選択します。

HNG の推奨される使用法

HNG インデックスは、GROUP BY では推奨されません。

次の場合に、HNG インデックスを使用します。

- ユニークな値の数が多 (1,000 超)。
- カラムで GROUP BY を実行する必要がない。

High_Non_Group の長所と短所

High_Non_Group インデックスには、次のような長所と短所があります。

表 13 : HNG の長所／短所

長所	短所
<p>圧縮アルゴリズムを使用しているため、パフォーマンスを犠牲にせずに必要なディスク領域を小さくすることができます。</p> <p>一部の例外を除き、カラムにユニークな値が多数ある場合に最高速のインデックス。</p>	<p>GROUP BY クエリでは、このインデックスタイプは推奨されない。</p> <p>一意性が適用される場合は使用不可。</p> <p>カラム内のデータが FLOAT、REAL、DOUBLE、BIT、BINARY、VARBINARY、255 バイト超の CHAR、または 255 バイト超の VARCHAR の場合は使用不可。</p>

他のインデックスとの比較

HNG インデックスを類似したインデックスと比較します。

- **HNG** インデックスは、必要なディスク領域は **HG** よりも小さくなりますが、**GROUP BY** を効率的に実行できません。
- **LF** と **HNG** のいずれかを選択する場合、決定要因となるのはユニークな値の数です。ユニークな値の数が 1,000 を超える場合は、**HNG** を使用します。

HNG カラムのその他のインデックス

High_Group インデックスは、**HNG** カラムにも使用できます。

Low_Fast (LF) インデックス

このインデックスは、性別や Yes/No、True/False、扶養家族の人数、収入レベルなど、非常に少数のユニークな値 (1,000 未満) が格納されるカラムでの使用に最適です。**LF** は、SAP Sybase IQ で最速のインデックスです。

等号のテストを行うときは、ルックアップを 1 回行うだけで結果セットをすばやく取得できます。不等号のテストを行うには、あと 2～3 回のルックアップを行う必要があります。**SUM**、**AVG**、**COUNT** などの計算も、このインデックスを使用すると非常に高速に行うことができます。

カラム内のユニークな値の数が増加すると、挿入やいくつかのクエリでパフォーマンスが低下し、必要なメモリとディスクの容量が増加し始めます。それでも、等号のテストを行うときには、多数のユニークな値を含むカラムでも、これがもっとも高速なインデックスとなります。

推奨される使用法

LF インデックスを使用するのは、カラムに含まれるユニークな値の数が 1,000 未満である場合、またはカラムに含まれるユニークな値の数が 1,000 未満で、そのカラムがジョイン述部で使用される場合です。

10,000 個以上のユニークな値を含むカラムには、**LF** インデックスを使用しないでください。テーブル内のローの数が 25,000 未満の場合は、**HG** インデックスを使用してください。これにより、同じ操作で必要とされるディスク I/O 操作が減少します。

Low_Fast の長所と短所

Low_Fast インデックスには、次のような長所と短所があります。

長所	短所
高速。特に単一テーブルの SUM 、 AVG 、 COUNT 、 COUNT DISTINCT 、 MIN 、 MAX 操作では高速である。	ユニークな値が最大 10,000 個の場合に使用されるインデックスタイプ。 カラム内のデータが BIT、255 バイト超の VARBINARY、255 バイト超の CHAR、または 255 バイト超の VARCHAR の場合は使用不可。

他のインデックスとの比較

LF インデックスを他のインデックスと比較する方法について理解を深めてください。

HNG/HG

考慮すべき主要な要素は、カラム内のユニークな値の数です。数が小さい場合は **LF** を使用します。

追加インデックス

Low_Fast カラムには、High_Non_Group インデックスタイプも適しています。

注意：ユニークな値の数が少ない場合 (1,000 未満の場合) は、通常、**LF** インデックスを使用するのが最適です。**WHERE** 句に指定されるカラムの場合は、まずこのインデックスを検討してください。その他のインデックスタイプ (**HG** および **HNG**) は、ユニークな値の数が多の場合にのみ検討してください。ユニークな値の数が多範囲クエリでは、**HNG** インデックスの使用も検討してください。

TEXT インデックス

TEXT インデックスを使用するには、非構造化データ分析機能を使用する正規のライセンスを取得しておく必要があります。

注意：『非構造化データ分析』を参照してください。

カラム文字列でキーワードを使用する Containment(WD) インデックスとは異なり、**TEXT** インデックスには、インデックスカラム内の単語の位置情報が格納されます。**TEXT** インデックスを使用したクエリは、テーブル内のすべての値をスキャンする必要があるクエリよりも高速になる可能性があります。

TEXT インデックスの作成 (Interactive SQL)

全文検索を実行するには、検索するカラムに **TEXT** インデックスを作成する必要があります。

TEXT インデックスには、インデックスカラム内の単語の位置情報が格納されません。

1. **CREATE ANY INDEX** または **CREATE ANY OBJECT** システム権限を持つユーザとして、あるいは基本となるテーブルの所有者としてデータベースに接続します。
2. **CREATE TEXT INDEX** 文を実行します。

次の例は、**TEXT** インデックス myTxtIdx を、iqdemo データベースの Customers テーブルの CompanyName カラムに作成します。default_char テキスト設定オブジェクトが使用されます。

```
CREATE TEXT INDEX myTxtIdx ON Customers
( CompanyName ) CONFIGURATION default_char
```

時刻 (TIME) インデックス

TIME インデックスは、時間量を扱うクエリを処理するときに使用します。

TIME インデックスは、データ型 **TIME** のカラムに対して使用されます。

TIME インデックスタイプの推奨される使用法

時刻関連の関数や演算を含むクエリで **TIME** カラムが使用されているときに **TIME** を使用します。

- DATEPART 等号述部 (=、!=)、DATEPART 範囲述部 (>、<、>=、<=、!>、!<、**BETWEEN**)、DATEPART IN リスト述部を含むクエリ
- 範囲述部 (>、<、>=、<=、**BETWEEN**) を含むクエリ

注意： **TIME** カラムを使用する単純な等号述部 (DATEPART を含まない) の場合は、**LF** と **HG** インデックスを使用すると最高のパフォーマンスが得られます。**LF** また

は **HG** インデックスを使用できない場合は、**TIME** インデックスを使用して結果を取得します。

等号述部 (ジョイン条件を含む) または **IN** 述部のための **GROUP BY** 句または **WHERE/HAVING** 句で、**TIME** または **TIMESTAMP** カラムが使用されている場合は、これらのカラムに **LF** または **HG** インデックスを用意する必要があります。高速な等号探索を実行できるのはこれらのインデックスだけだからです。

TIME の長所と短所

TIME インデックスには、次のような長所と短所があります。

長所	短所
時刻を扱うクエリを他のインデックスタイプより高速に処理できる。 TIME インデックスは作成したり、削除したりできる。	HNG インデックスよりも多くのディスク領域を使用する。 高速な等号探索を行うには LF または HG インデックスが必要である。 カラム内のデータが TIME データ型の場合のみこれらのインデックスを使用できる。

TIME インデックスの制限

TIME インデックスには、次の制限が適用されます。

- **UNIQUE** キーワードを使用できない。
- 単一カラムに対してのみ作成できる。
- 日付の単位 **Millisecond** をサポートしない。

インデックスの選択条件

ある所定のカラムにインデックスセットを定義すると、クエリの処理速度を大幅に向上させることができます。

インデックスの選択については、4つの主要な条件があります。

- ユニークな値の数
- クエリのタイプ
- ディスク領域の使用率
- データ型

個々の条件ではなくすべての条件を組み合わせ、推奨されるものを使用してください。また、すべてのカラムは自動的に、高速射影が簡単に実行できる方法で保管されます。追加インデックスの作成を判断するには、各カラムのデータを詳しく見ます。クエリの結果で想定されるユニークな値の数と合計値を予測します。

インデックス内のユニークな値の数

インデックスは、その中に含まれるユニークな値 (別個の値) の数に従って最適化されます。

IQ UNIQUE は、カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(n) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT オプションおよび FP_NBIT_LOOKUP_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

- FP_NBIT_AUTOSIZE_LIMIT は、NBit としてロードする、重複しない値の数を制限します。
- FP_NBIT_LOOKUP_MB は、NBit ディクショナリの合計サイズのスレッシュホールドを設定します。
- FP_NBIT_ROLLOVER_MAX_MB は、NBit からフラット FP への暗黙的な NBit ロールオーバーで使用するディクショナリサイズを設定します。
- FP_NBIT_ENFORCE_LIMITS は、NBit ディクショナリのサイズ制限を強制します。このオプションはデフォルトで OFF になっています。

IQ UNIQUE を FP_NBIT_AUTOSIZE_LIMIT 未満の n 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が FP_NBIT_AUTOSIZE_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

個別値の数が特定のレベルに達したら、次の表の推奨事項に従ってインデックスを選択してください。

表 14 : 考慮する順序

ユニークな値の数	推奨されるインデックスタイプ
1,000 未満	LF (テーブルのロー数が 100,000 未満の場合は HG)
1,000 以上	HG and/or HNG

以下に、さまざまな数のユニークな値を含むカラムの例を示します。

- 結婚状況のカラム。このカラムに入るユニークな値は非常に少数です (独身、既婚、NULL)。
- 州名や県名のカラム。このカラムに入るユニークな値は、100 個未満です。

- 日付データが入るカラム。このカラムに入るユニークな値は、100 個より多く、65536 個未満になります。
- 口座番号や社会保険番号が入るカラム。このカラムに入るユニークな値は、数千個または数百万個にも及びます。

クエリのタイプ

一部のインデックスタイプは、特定のタイプのクエリを最適化します。

一般にカラム内のデータに対してどのようなクエリが実行されるかを、あらかじめ調べておきます。次に例を示します。

- そのカラムはジョイン述部の一部になるか。
- カラム内にユニークな値が多数ある場合、そのカラムは **GROUP BY** 句で使用されるのか、**COUNT DISTINCT** の引数になるのか、**SELECT DISTINCT** 射影に含まれるのか。
- そのカラムは、同じデータ型、精度、位取りを持つ他のカラムと頻繁に比較されるか。

多くの場合、カラムのデータ型によって、そのカラムの用途を判断できます。たとえば、日付カラムは **WHERE** 句で範囲検索に使用されることが多く、価格や売上高のカラムは射影の中で集約関数 (**SUM**、**AVG** など) の引数として使用されるのが一般的です。

注意： SAP Sybase IQ は不適切なインデックスタイプのカラムを含むクエリでも解決できますが、効率は下がります。

IQ ではオプティマイザで利用できる統計がインデックスしかないため、クエリパフォーマンスを最適化するには、ジョイン述部、サブクエリの述部、**GROUP BY** 句、および **DISTINCT** 句で使用されるカラムが **HG** インデックスか **LF** インデックスを持っている必要があります。高カーディナリティカラムでは **HG**、低カーディナリティカラムでは **LF** を使用しますが、ロー数が 100,000 未満のテーブルでは **HG** を使用してください。

一般に次のような予測が有効とされますが、他の要素が優先される場合もあります。

- 範囲述部に関しては、ユニークな値の数が最も重要な要素となる。
- set 関数 **COUNT**、**COUNT DISTINCT**、**SUM**、**MIN**、**MAX**、および **AVG** を使用する場合、デフォルト以外のインデックスを使用するには、クエリ全体を単一のテーブルを使用して解決する必要がある。
- **BIT** データは、デフォルトのインデックスのみで使用可。255 バイトより大きい **VARBINARY** データはデフォルト、**TEXT**、および **CMP** インデックスタイプでのみ使用可。255 バイトより大きい **CHAR** および **VARCHAR** データは、デフォルト、**CMP**、**TEXT**、および **WD** インデックスタイプでのみ使用可。**LONG VARCHAR** データはデフォルト、**TEXT**、および **WD** インデックスタイプでのみ

使用可。DATE データのみ、**DATE** インデックスタイプで使用可。TIME データのみ、**TIME** インデックスタイプで使用可。DATETIME および TIMESTAMP データのみ、**DTTM** インデックスタイプで使用可。

クエリタイプにより推奨されるインデックス

クエリに対して推奨されるインデックスタイプを使用します。

各クエリでの動作が通常速いものから遅いものへと順に記載します。これらの推奨インデックスのみが、インデックスタイプを選択する条件というわけではありません。ユニークな値の数やディスク空き容量も考慮に入れる必要があります。

クエリ用途のタイプ	推奨されるインデックスタイプ
SELECT 射影のリスト内	デフォルト
SUM(A+B) などの計算式内	デフォルト
AVG/SUM 引数として	LF, HG、デフォルト
MIN/MAX 引数として	LF, HG
COUNT 引数として	デフォルト
COUNT DISTINCT, SELECT DISTINCT または GROUP BY の引数として	LF、HG、デフォルト
分析関数の引数として	LF、デフォルト
フィールドが重複不可の場合	HG
アドホックジョイン条件で使用されるカラム	デフォルト、HG, LF、
WHERE 句の LIKE 引数として	デフォルト
IN 引数として	HG, LF
等号または不等号 (=、!=) 内	HG, LF、または CMP
WHERE 句の範囲述部 (>、<、>=、<=、BETWEEN) 内	LF、HG、または CMP、DATE、TIME、DTTM
DATEPART 等号述部、範囲述部、および IN リスト述部内	DATE、TIME、DTTM
CONTAINS 述部内	WD, TEXT

カラム値により推奨されるインデックス

カラム内のユニークな値の数のような基準は、データにとって適切なインデックスタイプを識別するうえで役立ちます。

識別の条件	選択するインデックス
すべてのカラムで自動的に作成されたインデックスを調べる。	デフォルトインデックス
UNIQUE または PRIMARY KEY 制約のあるカラムで自動的に作成されたインデックスを調べる。	UNIQUE によって作成された HG
ジョイン述部で使用されているすべてのカラムを識別し、ユニークな値の数に従ってインデックスタイプを選択する。	HG または LF
含まれているユニークな値が少なく、複数のインデックスをまだ使用していないカラムを識別する。	LF
含まれるユニークな値の数が多く、 SELECT DISTINCT または DISTINCT COUNT の select リスト内の GROUP BY 句の一部になっているカラムを識別する。	HG
アドホックジョインクエリの WHERE 句で使用される可能性のあるカラムの中で、まだ HG または LF インデックスがないカラムを識別する。	HG または LF
含まれるユニークな値の数が多く、 GROUP BY 、 SELECT DISTINCT 、または DISTINCT COUNT では使用されないカラムを識別する。	HNG
データ型、精度、位取りが同じで、頻繁に比較される可能性がある一対のカラムを識別する。	CMP
キーワードまたは URL のリストを含むカラムを識別する。	WD
ユニークな値が多数含まれ、 GROUP BY 、 SELECT DISTINCT 、または DISTINCT COUNT では使用されない、 DATE 、 TIME 、 DATETIME 、または TIMESTAMP のカラムを識別する。	DATE 、 TIME 、 または DTTM

識別の条件	選択するインデックス
残りのすべてのカラムを見て、ユニークな値の数、クエリのタイプおよびディスク容量に基づいて追加インデックスを作成するかを決定する。またすべてのカラムについて、選択したインデックスタイプでそのカラムのデータ型が使用可能であることを確認する。	

ディスク領域の使用率

次の表では、各インデックスが使用する領域量の推定値と、ソースデータベースまたはフラットファイルのカラムデータの量との比較を示します。

表 15 : インデックスによるディスク領域の使用率

インデックスのタイプ	生データと比較した領域の概算	コメント
Default	生データより小さいか、同じ	独自の値の数が 255 未満の場合、インデックスが使用する領域は生データよりも大幅に小さくなる。
High_Group	2 倍未満	独自の値の数が減少する (つまりグループごとのエントリの数が増加する) につれて、使用領域は生データのサイズに比例して減少する。
High_Non_Group	生データより小さいか、同じ	ほとんどの場合は生データよりも小さくなる。
Low_Fast	2 倍未満	High_Group と同じ。
Date	生データより小さいか、同じ	High_Non_Group より大きい。
Time	生データより小さいか、同じ	High_Non_Group より大きい。
Datetime	生データより小さいか、同じ	High_Non_Group より大きい。

LF および **HG** インデックスの場合、インデックスのサイズはユニークな値の数によって異なります。ユニークな値の数が多いほど、インデックスに必要な領域は増加します。

CMP インデックスは常に追加のインデックスになるので、ディスク領域の節約にはなりません。

インデックス内のデータ型

カラムにインデックスを作成するときに、カラムのデータ型に対してサポートされているインデックスタイプのみを選択します。

デフォルトインデックスには任意のデータ型を使用できます。

表 16 : 各データ型でサポートされるインデックス

データ型	サポートされるインデックス	サポートされないインデックス
tinyint	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
smallint	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
int	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
unsigned int	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
bigint	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
unsigned bigint	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
numeric、decimal	CMP、HG、HNG、LF	WD、DATE、TIME、DTTM、TEXT
double	LF (HG も許容、ただし非推奨)	CMP、HNG、WD、DATE、TIME、DTTM、TEXT
float	LF (HG も許容、ただし非推奨)	CMP、HNG、WD、DATE、TIME、DTTM、TEXT
real	LF (HG も許容、ただし非推奨)	CMP、HNG、WD、DATE、TIME、DTTM、TEXT
bit	(デフォルトインデックスのみ)	CMP、HG、HNG、LF、WD、DATE、TIME、DTTM、TEXT
date	CMP、HG、HNG、LF、DATE	WD、TIME、DTTM、TEXT
time	CMP、HG、HNG、LF、TIME	WD、DATE、DTTM、TEXT

データ型	サポートされるインデックス	サポートされないインデックス
datetime、timestamp	CMP、HG、HNG、LF、DTTM	WD、DATE、TIME、TEXT
char (255 バイト以下)、character	CMP、HG、HNG、LF、WD、TEXT	DATE、TIME、DTTM
char (255 バイト超)	CMP、WD、TEXT	HG、HNG、LF、DATE、TIME、DTTM
varchar (255 バイト以下)	CMP、HG、HNG、LF、WD、TEXT	DATE、TIME、DTTM
varchar (255 バイト超)	CMP、WD、TEXT	HG、HNG、LF、DATE、TIME、DTTM
long varchar	WD、TEXT	CMP、HG、HNG、LF、DATE、TIME、DTTM
バイナリ (255 バイト以下)	CMP、HG、LF、TEXT	HNG、WD、DATE、TIME、DTTM
バイナリ (255 バイト超)	CMP、TEXT	HG、HNG、LF、WD、DATE、TIME、DTTM
varbinary (255 バイト以下)	CMP、HG、LF、TEXT	HNG、WD、DATE、TIME、DTTM
varbinary (255 バイト超)	CMP、TEXT	HG、HNG、LF、WD、DATE、TIME、DTTM

インデックスタイプの組み合わせ

カラムを複数のタイプのクエリで使用する場合、複数のインデックスタイプを割り当てることが必要な場合もあります。

表 17: インデックスの組み合わせ

	追加するインデックス					
既存のインデックス	HG	HNG	LF	CMP ^a	WD	DATE、TIME、またはDTTM
HG	-	1	2	1	1	1
HNG	1	-	1	1	2	2
LF	2	1	-	1	2	1

1 = 推奨される組み合わせ

2 = 推奨されない組み合わせ

a. CMP インデックスは 1 対のカラムに適用される。それぞれのカラムは、CMP 以外のインデックスを少なくとも 1 つは持っている。

インデックスの作成

IQ ストア内にデータベースを作成すると、SAP Sybase IQ はデフォルトインデックスを各テーブルカラムに自動的に作成します。最適なパフォーマンスのためには、追加インデックスを作成してください。

前提条件

分割データにインデックスを作成する場合、すべての DB 領域がオンラインである必要があります。

手順

1. データに適したインデックスタイプを選択します。選択するインデックスタイプが、そのカラムのデータ型に対応していることを確認します。
インデックスタイプを指定しない場合、SAP Sybase IQ は **HG** インデックスを作成します。フロントエンドツールの中には、この理由で自動的に **HG** インデックスを作成するものがいくつかあります。
2. 次のような方法でインデックスを作成します。
 - **CREATE INDEX** 文を入力します。
たとえば、SalesOrderItems テーブルの ShipDate カラムに、ShipIx という High_Non_Group (**HNG**) インデックスを作成します。

```
CREATE HNG INDEX ShipIx
ON dbo.SalesOrderItems (ShipDate)
```
 - SAP Control Center のテーブル作成ウィザードを使用します。
 - **CREATE TABLE** の PRIMARY KEY または UNIQUE カラム制約を使用して、**HG** インデックスを自動的に作成します。

参照：

- CREATE INDEX 文 (495 ページ)
- CREATE TABLE 文 (511 ページ)

カラムインデックスの同時作成

場合によっては、同時に複数のカラムインデックスを作成できます。

- それぞれの **CREATE INDEX** 文では、1つのインデックスしか作成できません。
- 同じテーブルに対して2つの接続から **CREATE INDEX** 文が発行された場合は、先に発行された文が処理されます。2番目の文に対しては、同時に実行可能なライタが1つのみであることを示すエラーメッセージが返されます。
- 2つの接続がそれぞれ異なるテーブルに対して **CREATE INDEX** 文を発行した場合は、両方の文が並行して処理されます。

インデックスのロードに関するステータスメッセージ

進行状況メッセージが有効な場合は、100,000レコードが挿入されるたびにステータスメッセージが表示されます。

メッセージを有効にするには、**CREATE INDEX** の **NOTIFY** オプションをゼロ以外の値に設定します。

レコード数を変更するには、**CREATE INDEX** の **NOTIFY** オプション、または **NOTIFY_MODULUS** オプションを使用します。これらのメッセージを表示しないようにするには、**NOTIFY 0** と指定します。

データ挿入後のカラムインデックスの追加

追加のカラムインデックスを作成すると、**CREATE INDEX** コマンドによって新規インデックスが個々のテーブルの一部として作成されます。

個々のテーブルの既存のカラムインデックスにすでにデータが格納されている場合は、**CREATE INDEX** 文によって、既存のインデックスから新規インデックスにデータが挿入されます。これによって、個々のテーブルにあるカラムのすべてのカラムインデックス間で、データの整合性が確保されます。

この機能は、すでにカラムにデータを挿入した後で追加インデックスが必要であると判明した場合に便利です。この方法を使用すると、最初からやりなおさなくてもインデックスを追加できます。

注意： 既存のインデックスからのデータの挿入には、時間がかかる場合があります。データを挿入する前に適切なインデックスをすべて作成してから、**LOAD TABLE** 文または **INSERT** 文を使用してすべてのインデックスに一度にデータを挿入すると、所要時間を短縮できます。

CREATE INDEX 文のグループの実行

キーワードの **BEGIN PARALLEL IQ** と **END PARALLEL IQ** を使用して、グループとして同時に実行したい複数の **CREATE INDEX** 文を区切ります。

これらのキーワードは、テンポラリテーブルやカタログストアテーブルではなく IQ メインストア内のベーステーブルにインデックスを作成する場合にのみ使用します。

これらの **CREATE INDEX** 文が 1 つでも失敗すると、すべてがロールバックされません。

```
BEGIN PARALLEL IQ
  CREATE HG INDEX c1_HG on table1 (col1);
  CREATE HNG INDEX c12_HNG on table1 (col12);
  CREATE LF INDEX c1_LF on table1 (col1);
  CREATE HNG INDEX c2_HNG on table1 (col2);
END PARALLEL IQ
```

インデックスアドバイザーの実行

データベースで **INDEX_ADVISOR** オプションを設定している場合、SAP Sybase IQ は、パフォーマンス向上のためにインデックスを追加することを推奨するメッセージをメッセージログまたはクエリプランに発行します。

1. Interactive SQL で **INDEX_ADVISOR** オプションを設定します。

```
SET OPTION index_advisor = 'ON'
```

2. メッセージログファイルを調べます。デフォルトでは、メッセージログは dbname.iqmsg という名前のファイルとして、カタログストアと同じディレクトリ内に作成されます。

次の領域に注目するメッセージを探します。

- ローカル述部カラム
- シングルカラムジョインキーカラム
- 関連サブクエリカラム
- グループ化カラム

3. インデックスアドバイザーメッセージの推奨に従う場合には、インデックスを自分で作成します。

```
CREATE HG INDEX id_hg
ON SalesOrderItems
( ID ) IN Dsp5
```

インデックスの名前変更

所有者のタイプが **USER** の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インデックスは名前を変更できます。

インデックスの名前を変更するには、**ALTER INDEX** 文を使用します。

注意： キー制約を適用するために作成したインデックスの名前は変更できないという点に注意してください。

インデックスの表示

インデックスに関する情報を表示するには、ストアドプロシージャ **sp_iqindex** を使用します。

Departments テーブルのインデックスをリストするには、次のコマンドを発行します。

- `sp_iqindex 'Departments'`

出力は、2つの部分に分けて読みやすくしてあります。

table_name	table_owner	column_name	index_type
Departments	GRUPO	DepartmentHeadID	FP
Departments	GRUPO	DepartmentHeadID	HG
Departments	GRUPO	DepartmentID	FP
Departments	GRUPO	DepartmentID	HG
Departments	GRUPO	DepartmentName	FP

- すべてのテーブルに関する情報を表示するには、**sp_iqindex** からテーブル名を省略します。

index_name	unique_index	dbspace_id	remarks
ASIQ_IDX_T740_C3_FP	N	16,387	(NULL)
ASIQ_IDX_T740_C3_HG	N	16,387	(NULL)
ASIQ_IDX_T740_C1_FP	U	16,387	(NULL)
ASIQ_IDX_T740_I4_HG	Y	16,387	(NULL)
ASIQ_IDX_T740_C2_FP	N	16,387	(NULL)

システムビューに含まれるインデックス情報

インデックスに関する情報は、統合ビュー SYSINDEXES とシステムビュー SYSIQIDX にあります。

これらのビューの詳細については、『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』の「システムテーブルとシステムビュー」を参照してください。

インデックスの削除

必要なくなったカラムインデックスを削除します。
次のいずれかを実行します。

- **DROP** 文を入力します。『リファレンス：文とオプション』の「DROP 文」を参照してください。
- SAP Sybase IQ でインデックスを削除します。

外部キー制約削除時のインデックスの保持

外部キー制約を削除するときは、基本となる **HG** インデックスを残してください。非ユニークな **HG** インデックスはクエリのパフォーマンスを向上させますが、作成にコストがかかることがあります。

ALTER TABLE DROP FOREIGN KEY 文を入力します。

リファレンス：文とオプションの **ALTER INDEX** 文を参照してください。

注意： **ALTER TABLE DROP FOREIGN KEY CONSTRAINT** は、自動的に作成されたユニークでない **HG** インデックスを削除しません。関連する外部キーが残っている場合は、プライマリキーを削除できません。このようなインデックスを削除するには、**ALTER TABLE DROP FOREIGN KEY** を発行した後に明示的に削除する必要があります。

ジョインのパフォーマンスの最適化

インデックスを作成して、ジョインパフォーマンスを最適化することができます。

ジョインの処理を最高速にするには、以下で参照可能なすべてのカラムに Low_Fast (**LF**) または High_Group (**HG**) インデックスを作成します。

- ジョインクエリの **WHERE** 句

- 集約関数以外のジョインクエリの **HAVING** 句の条件

次に例を示します。

```
SELECT n_name, sum(l_extendedprice*(1-l_discount)) AS revenue
FROM customer, orders, lineitem, supplier, nation, region
WHERE c_custkey = o_custkey AND o_orderkey =
l_orderkey AND l_suppkey = s_suppkey AND
c_nationkey = s_nationkey AND s_nationkey =
n_nationkey AND n_regionkey = r_regionkey AND
r_name = 'ASIA' AND o_orderdate >= '1994-01-01'
AND o_orderdate < '1995-01-01' GROUP BY n_name HAVING n_name LIKE
"I%" AND SUM(l_extendedprice*(1-l_discount)) > 0.50 ORDER BY 2
DESC
```

このクエリで参照されるカラムのうち、`l_extendedprice` と `l_discount` を除くすべてのカラムに、**LF** または **HG** インデックスが必要です。

データ整合性の確保

テーブルとカラムの制約とデータ型を適切に選択することによって、データベース内のデータの有効性と信頼性が確保されます。

データ整合性の概要

データの整合性の確保とは、データが有効、つまり適切かつ正確で、データベースの関係構造が保たれていることを意味します。

データベースの関係構造は、**参照整合性制約** (テーブル間のデータの一貫性を保つビジネスルール) によって記述されます。

SAP Sybase IQ では、データベースへのデータ入力方法の詳細な制御を可能にするストアードプロシージャと JDBC をサポートしています。JDBC の詳細については、SAP Sybase IQ の『プログラミング』のガイドを参照してください。

データが有効でない場合

適切な検査が行われないと、データが無効になる場合があります。

次に例を示します。

- customer テーブルに存在しない **customer_id** で orders テーブルに注文を入力した場合
- 数値データを入力するべき場所にテキストを入力した場合
- カラム長より長い数値データを入力した場合
- 複数のユーザが **dept_id** 200 で新しい部署に関する同じ情報を department テーブルに入力した場合

有効なデータに関するルールと検査

データベースのデータの有効性を確保するには、有効なデータと無効なデータを定義する検査を作成し、データが従うべきルールを設計します。

多くの場合こうしたルールは、ビジネスルールと呼ばれます。検査とルールをまとめて制約と呼びます。特定のカラムのデータ整合性を管理するルールはカラム制約です。特定のテーブルの 1 つまたは複数のカラムの整合性を管理するルールはテーブル制約です。テーブル制約とカラム制約はいずれも、テーブル内の単一のカラムに適用できます。また、テーブル制約では、テーブル内の一連のカラムに対するルールも設定できます。

データベース自体に組み込まれた制約は、クライアントアプリケーションに組み込まれたり、データベースユーザへの説明として書き出された制約よりも、本質

的に信頼できます。データベースに組み込まれた制約はデータベースの定義の一部であり、すべてのアプリケーションに対して常に適用されるためです。

データベース内に制約が設定されると、それ以降に発生するすべてのデータベースとの対話にその制約が適用されます。これに対して、クライアントアプリケーションに組み込まれた制約は、ソフトウェアが変更されるたびに影響を受けます。複数のアプリケーションに制約を組み込んだり、1つのアプリケーション内の複数の場所に組み込むことが必要になります。

SAP Sybase IQ が適用するかどうかにかかわらず、適用する制約はすべて宣言してください。制約を宣言することによって、データ要件を確認し、組織のビジネスルールに合ったデータベースを設計できます。

SAP Sybase IQ は、指定された制約に基づいて、何種類かの最適化を実行します。この最適化は、制約の適用には依存しません。クエリとロードオペレーションで最高のパフォーマンスを得るには、データベースにすべての制約を組み込みます。**FOREIGN KEY**、**PRIMARY KEY**、**UNIQUE**、および **IQ UNIQUE** カラム制約はロードとクエリのパフォーマンスを向上できます。

SAP Sybase IQ は、ロード処理時に特定の制約が守られているかどうかを検査します。たとえば、SAP Sybase IQ では、ロードされているデータのデータ型と長さが適切かどうかを確認されます。

データベースの内容を変更する文

クライアントアプリケーションは、SQL 文を発行することによってデータベーステーブル内の情報を変更します。

データベースの情報を実際に変更する SQL 文は、それほど多くはありません。

- テーブルの既存のローを削除するには、**DELETE** 文を使用します。
- テーブルに新しいローを挿入するには、**INSERT** 文または **LOAD TABLE** 文を使用します。
- セルの値を変更するには、**UPDATE** 文を使用します。

データ整合性ツール

データの整合性を維持するには、データ制約やデータベースの参照構造を指定する制約を使用します。

制約

個々のカラムまたはテーブルのデータに対して使用可能な制約タイプもあります。

- **NOT NULL** 制約は、カラムに null エントリが格納されないようにします。SAP Sybase IQ ではこの制約が適用されます。

- カラムに CHECK 条件を割り当て、そのカラムのすべてのローが特定の条件に従うように指定できます。たとえば、salary カラムのエントリが必ず指定範囲内になるようにすることができます。
- 複数のカラム間で関連する値に CHECK 条件を付けることができます。たとえば、図書館データベースで、date_returned は date_borrowed よりも後のエントリであるという条件を指定できます。

カラムの制約はユーザ定義のデータ型から継承できます。

エンティティと参照整合性

リレーショナルデータベーステーブル内のデータは、テーブル間の関係によって互いに関連付けられています。この関係は、データベース設計に組み込まれた候補キーと外部キーによって定義されます。

外部キーは、1つのカラムまたは複数のカラムの組み合わせによって構成されます。各外部キーは、一方のテーブル(外部テーブル)の情報をもう一方のテーブル(参照テーブルまたはプライマリテーブル)の情報と関連付けます。外部テーブル内の特定のカラムまたはカラムの結合がプライマリテーブルに対する外部キーとして指定されます。

一意性制約を持つプライマリキーまたはカラム(単一または複数)は、候補キーと呼ばれます。参照されるカラムおよびカラムのセットは候補キーでなくてはなりません。このカラムは参照キーと呼ばれます。外部キーでもある候補キーに対しては、外部キー制約を指定できません。

次に示す参照整合性ルールによって、データベースの構造が定義されます。

- プライマリキーを追跡することで、null ではないことを保証するプライマリキーによって、特定のテーブルのすべてのローをユニークに特定できることを保証します。
- テーブル間の関係を定義する外部キーを追跡します。外部キー値はすべて、対応するプライマリキーの値と一致しているか、NULL が許容される定義の場合には NULL 値を格納している必要があります。

整合性制約を実装する文

SQL 文を発行することによって整合性制約を実装します。

- **CREATE DATABASE** 文は、データベースの作成時に整合性制約を実装します。
- **ALTER DATABASE** 文は、既存のデータベースに整合性制約を追加したり、制約を削除したりします。

カラムのデフォルトを使用したデータ整合性の向上

データベーステーブルに新しいローが入力されるたびに、カラムのデフォルトによって、特定の値が単数または複数のカラムに自動的に割り当てられます。

カラムのデフォルト値について、クライアントアプリケーション側では何もする必要はありません。ただし、クライアントアプリケーションでカラムの値を指定すれば、その新しい値によってカラムのデフォルト値が上書きされます。

カラムのデフォルトを使用すると、ローが挿入された日付や時刻、テーブル内のローを最初に変更したユーザのユーザ ID といった情報を、カラムにすばやく自動的に入力できます。カラムのデフォルトを使用すると、データの整合性が高まりますが、データの整合性が強制的に適用されるわけではありません。クライアントアプリケーションは常にデフォルトに上書きできます。

サポートされるデフォルト値

SAP Sybase IQ では、カラムのデフォルト値がサポートされています。

- **CREATE TABLE** 文または **ALTER TABLE** 文で指定された文字列
- **CREATE TABLE** 文または **ALTER TABLE** 文で指定された数値
- 自動的に増分された数値 (カラムの前の最大値より 1 つ大きい値)
- **NEWID** 関数によって生成される UUID (Universally Unique Identifier) 値
- 現在の日付、時刻、またはタイムスタンプ
- 現在のデータベースの名前
- データベースユーザの現在のユーザ ID およびローを最後に変更したユーザの名前
- SQL Remote アプリケーション用データベースのパブリッシュユーザ ID
- NULL 値
- データベースオブジェクトを参照していない定数式
- **CREATE DOMAIN** 文を使用してユーザ定義のドメイン (データ型) で指定された、サポートされるデフォルト値

デフォルト値の制限

一部のカラムデフォルト値はサポートされていません。

SAP Sybase IQ は、カラムのデフォルトとして次の値をサポートしません。

- **UTC TIMESTAMP**、**CURRENT UTC TIMESTAMP**、**GLOBAL AUTOINCREMENT** という特別な値を使用する値
- カラムのデータ型と互換性のないデフォルト値

- テーブルまたはカラムの検査制約に違反するデフォルト値
- データベースオブジェクトを参照する定数式

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。

カラムのデフォルトの作成

CREATE TABLE 文を使用してテーブルの作成時にカラムデフォルトを作成するか、後で **ALTER TABLE** 文を使って追加します。

CREATE DOMAIN 文を使用して、ユーザ定義のドメイン (データ型) の作成時にデフォルト値を指定することもできます。

ストアドプロシージャ `sp_iqcolumn` は、すべてのテーブルのすべてのカラムに関する情報を返します。`sp_iqcolumn` の結果セットによって返されるカラムの 1 つは「デフォルト」と呼ばれ、そのカラムの特定のデフォルト値を示します。

1. 次の文は、`CHARACTER` 型のカラム `c1` に対して指定されたデフォルトの特別値 **LAST USER** を持つ `tab1` というテーブルを作成します。

```
CREATE TABLE tab1 (c1 CHAR(20) DEFAULT LAST USER)
```

2. `sales_order` テーブルの `id` という既存カラムに条件を追加して、クライアントアプリケーションが値を指定しない場合にカラムの値が自動的に増加するようにします。

```
ALTER TABLE sales_order MODIFY id DEFAULT AUTOINCREMENT
```

3. データ型が `INTEGER` でデフォルト値が 45 の、`dom1` というドメインを定義します。

```
CREATE DOMAIN dom1 INTEGER DEFAULT 45
```

カラムのデフォルトの変更

カラムのデフォルトを変更するには、デフォルトを作成するときと同じ形式の **ALTER TABLE** 文を使用します。

たとえば、`order_date` というカラムのデフォルト値を現在の設定から **CURRENT DATE** に変更するには、次の文を使用します。

```
ALTER TABLE sales_order
MODIFY order_date DEFAULT CURRENT DATE
```

カラムのデフォルトの削除

カラムのデフォルトを削除するには、デフォルトを `NULL` に変更します。

次の文は、`order_date` カラムからデフォルトを削除します。

```
ALTER TABLE sales_order
MODIFY order_date DEFAULT NULL
```

カラムのデフォルトとしてサポートされる値

SAP Sybase IQ では、カラムのデフォルト値をロードしたり、挿入したりできません。

次の文を使用します。

- **INSERT...VALUES**
- **INSERT...SELECT**
- **INSERT...LOCATION**
- **LOAD TABLE**
- **UPDATE**
- **SELECT...FROM...FOR UPDATE**

カラムのデフォルト値を定義して挿入する場合

- SAP Sybase IQ で評価できないデフォルト値を指定できます。サポートされていないデフォルト値を持つテーブルに **INSERT**、**LOAD**、または **ALTER ADD** 操作を実行すると、エラーがレポートされます。
- カラムのデータ型と互換性のないデフォルト値をサーバが挿入しようとするとき、SAP Sybase IQ はエラーまたは警告を生成します。たとえば、整数カラムに対して 'N/A' というデフォルト式を定義し、カラム値を指定せずに挿入またはロードを行うと、**CONVERSION_ERROR** データベースオプションの設定に応じてエラーまたは警告が生成されます。
- デフォルト値が **CHARACTER** 型のカラムには長すぎる場合、**STRING_RTRUNCATION** データベースオプションの設定に応じて、SAP Sybase IQ は文字列をトランケートするか例外を生成します。
- **VARCHAR** または **LONG VARCHAR** 型のカラムのデフォルト値が長さゼロの文字列の場合、**NON_ANSI_NULL_VARCHAR** データベースオプションの設定に応じて、SAP Sybase IQ は **NULL** または長さゼロの文字列を挿入します。
- **VARCHAR**、**CHAR**、または **LONG VARCHAR** カラムのデフォルト値に部分的なマルチバイト文字が含まれている場合、SAP Sybase IQ は **TRIM_PARTIAL_MBC** データベースオプションの設定に従って、値を挿入する前にその文字を削除することがあります。
- デフォルト値がテーブルまたはカラムの検査制約に違反している場合、サーバがそのデフォルト値を挿入しようとするたびに、SAP Sybase IQ からエラーメッセージが生成されます。
- **LOAD TABLE** 操作中にデフォルト値の挿入によって発生する制約違反は、ユーザが指定した **IGNORE CONSTRAINT** および **MESSAGE LOG/ROW LOG** オプションに関係なく、無視されません。

- **UTC TIMESTAMP** および **CURRENT UTC TIMESTMAP** のカラムデフォルト値は SAP Sybase IQ ではサポートされていません。
- ベーステーブルに定義されたカラムのデフォルト値は、それらのテーブルが関与するジョインに伝達されません。
- デフォルト値が定義されているカラムがテーブルに追加されると、その新しいカラムのすべてのローにそのデフォルト値が入力されます。
- テーブルの既存のカラムのデフォルト値を変更しても、テーブルの既存の値は変更されません。
- **LOAD TABLE** 文の **DEFAULT** オプションで指定されたデフォルト値を使用するには、**LOAD TABLE DEFAULTS** オプションが On でなければなりません。**DEFAULTS** オプションが OFF の場合、指定されたロードデフォルト値は使用されず、代わりに NULL 値がカラムに挿入されます。
- **LOAD TABLE DEFAULT** の指定には、**LOAD TABLE** コマンドで指定されたファイルからロードされるカラムを少なくとも 1 つ以上含めます。
- **LOAD TABLE DEFAULT** *default-value* の文字セットは、データベースのものと同一である必要があります。また、カラムとデフォルト値に関する制限でサポートされているデフォルト値に従っている必要があります。**LOAD TABLE DEFAULT** オプションでは、デフォルトのロード値として **AUTOINCREMENT**、**IDENTITY**、および **GLOBAL AUTOINCREMENT** はサポートされません。
- **LOAD TABLE DEFAULT** 句で指定されたロードデフォルト値では、デフォルト値の暗号化はサポートされていません。

日付、時刻、タイムスタンプのデフォルト

DATE、TIME、または TIMESTAMP データ型のカラムでは、**CURRENT DATE**、**CURRENT TIME**、**TIMESTAMP**、または **CURRENT TIMESTAMP** の特別値をデフォルトとして使用できます。選択するデフォルト値はカラムのデータ型との互換性がなければなりません。

CURRENT DATE デフォルト

CURRENT DATE デフォルトは、次のような日付を記録するのに便利です。

- 顧客連絡先データベースにおける通話の日付
- 販売エントリデータベースにおける注文の日付
- 図書館データベースにおける本の貸し出し日付

CURRENT TIMESTAMP デフォルト

CURRENT TIMESTAMP は **CURRENT DATE** に似ていますが、より高い精度の値を提供します。たとえば、顧客連絡先管理アプリケーションのユーザが、1日のうちと同じ顧客と何度も連絡をとることがあります。**CURRENT TIMESTAMP** のデフォルトは、これらの連絡を区別するのに便利です。

CURRENT TIMESTAMP は日付と時間を 100 万分の 1 秒単位で記録するので、**CURRENT TIMESTAMP** はデータベースに記録されている各イベントの順序が重要である場合に便利です。

TIMESTAMP デフォルト

DEFAULT TIMESTAMP を使用してカラムを宣言した場合、挿入操作とロード操作のデフォルト値が提供されます。この値は、ローが更新されたときに常に、最新の日時に更新されます。

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

SAP Sybase IQ は、**UTC TIMESTAMP** や **CURRENT UTC TIMESTAMP** のデフォルト値をサポートしていません。また、SAP Sybase IQ は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。SAP Sybase IQ は、**UTC TIMESTAMP** または **CURRENT UTC TIMESTAMP** タイプのカラムでデフォルト値の挿入または更新が試行されるたびにエラーを生成します。

USER デフォルト

DEFAULT USER をカラムに割り当てることで、データベースのエントリを入力したユーザを識別できます。たとえば、売り上げ歩合制の営業員がデータベースを使用する場合、このような情報が必要になります。

ユーザ ID をテーブルのプライマリキーに組み込むのは、たまに接続するユーザに有効な方法であり、情報更新中に矛盾が起きるのを防ぐのに役立ちます。このようなユーザは、データベースから必要な部分を携帯端末にコピーして、マルチユーザデータベースに接続していない状態でデータを修正し、後でサーバにアクセスしてトランザクションログを適用することができます。

USER デフォルト

特別値の **USER** と **CURRENT USER** は、現在の接続のユーザ ID が含まれている文字列を返し、文字データ型のカラムのデフォルト値として使用できます。**UPDATE** 操作時には、**USER** または **CURRENT USER** のデフォルト値を持つカラムは変更されません。

LAST USER デフォルト

特別値の **LAST USER** は、ローを最後に変更したユーザの名前を返し、文字データ型のカラムのデフォルト値として使用できます。**INSERT** 操作と **LOAD** 操作では、この定数は、**CURRENT USER** と同じ効果があります。**UPDATE** 操作では、**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ければ、現在のユーザ名に変更されます。

LAST USER を **DEFAULT TIMESTAMP** と組み合わせて使用すると、デフォルト値を使用して、ローを最後に変更したユーザと変更日時の両方を記録できます (ただし、別々のカラムに記録されます)。

IDENTITY または AUTOINCREMENT のデフォルト

IDENTITY/AUTOINCREMENT デフォルトは、値それ自体には意味がない数値データフィールドで役立ちます。

この機能では、新しい各ローの当該カラムに対して、現在の最大値よりも 1 大きい値が割り当てられます。**IDENTITY/AUTOINCREMENT** カラムは、注文伝票番号を記録したり、顧客からの問い合わせを識別したりする場合など識別番号が必要なときに使用できます。

通常、オートインクリメントカラムは、プライマリキーカラムであるか、またはユニークな値を持つように制約が付けられたカラムです (『リファレンス：文とオプション』の「**CREATE TABLE 文**」を参照してください)。サーバではインデックスまたはキー定義を使用して最大値が検出されるため、オートインクリメントのデフォルトは、たとえば、当該カラムがインデックスの先頭カラムの場合に有用です。

AUTOINCREMENT カラムに挿入された最新の値は、`@@identity` グローバル変数を使用して取得できる場合もあります。

SAP Sybase IQ は、特別値 **GLOBAL AUTOINCREMENT** はサポートしていません。

テーブル作成時の **IDENTITY/AUTOINCREMENT** の初期値は 0 です。正の整数でのみ機能します。

AUTOINCREMENT のデフォルトが設定されたカラムは、Transact-SQL アプリケーションでは **IDENTITY** カラムと呼ばれます。SAP Sybase IQ はどちらのキーワードもサポートしています。

NEWID デフォルト

テーブル内のローをユニークに識別するには、GUID (Globally Unique Identifiers) と呼ばれることもある UUID (Universally Unique Identifiers) を使用します。

この値は、あるコンピュータで生成される値が、別のコンピュータで生成される値と一致しないような方法で生成されます。このため、UUID は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。

詳細については、『リファレンス：ビルディングブロック、テーブル、およびプロセス』の「**NEWID**」を参照してください。

NULL デフォルト

NULL 値を許容するカラムに NULL デフォルトを指定しても、デフォルトを指定しない場合と同じになります。

ローを挿入するクライアントが値を明示的に割り当てなければ、ローは自動的に NULL 値を受け取ります。

NULL デフォルトを使用できるのは、一部のカラムの情報がオプションの場合か必ず使用できるとはかぎらない場合です。

NULL 値の詳細については、『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』の「NULL 値」を参照してください。

文字列と数値のデフォルト

カラムが文字列データ型または数値データ型を保持するものであるかぎり、特定の文字列または数値をデフォルト値として指定できます。

カラムのデータ型に変換できるデフォルト値を指定する必要があります。

デフォルトの文字列と数値は、当該のカラムに典型的なエン트리がある場合に有効です。たとえば、ある会社には本社 **city_1** と小規模の事業所 **city_2** の2つのオフィスがある場合、所在地カラムのデフォルトを **city_1** にしておけば入力が簡単になります。

定数式のデフォルト

データベースオブジェクトを参照していないかぎり、定数式はデフォルト値として使用できます。

GETDATE や **DATEADD** などの関数を定数式のデフォルト値の中で使用できます。デフォルト定数式が関数や単一値でない場合は、その式をカッコで囲む必要があります。

定数式では、カラムのデフォルトに、たとえば今日から 15 日目の日付を表すエントリを含めることができます。

```
... DEFAULT ( DATEADD( DAY, 15, GETDATE() ) )
```


テーブルとカラムの制約

制約を使用すると、テーブルに入力されたデータが正しいことを保証したり、パフォーマンスを向上させる情報を SAP Sybase IQ に提供することができます。

CREATE TABLE 文と **ALTER TABLE** 文を使って、テーブルにさまざまな属性を設定できます。基本的なテーブル構造 (カラム番号、カラム名、データ型、テーブルの名前と保存場所) 以外にも、データの整合性を保つための設定ができます。

警告！ テーブルを変更または作成すると、他のデータベースユーザに悪影響を及ぼす場合があります。大きなテーブルの場合、**ALTER TABLE** または **CREATE TABLE** の実行に時間がかかる場合があります。**CREATE TABLE** の処理によって、文が完了するまで、他のプロセスの実行が遅延します。**ALTER TABLE** 文は、他の接続がアクティブなときでも実行できますが、変更するテーブルを他の接続が使用しているときには実行できません。**ALTER TABLE** の実行中には、変更するテーブルを参照するその他のリクエストは受け入れられません。

カラムまたはテーブルに対する **UNIQUE** 制約

UNIQUE 制約では、1つまたは複数のカラムによってテーブル内の各ローがユニークに識別されるように指定します。

UNIQUE 制約は、基本的には **PRIMARY KEY** 制約と同じです。違いは、**UNIQUE** 制約は1つのテーブル内で複数指定できることです。**UNIQUE** と **PRIMARY KEY** のいずれについても、カラムに **NULL** 値を格納することはできません。

例 1

次の例では、カラム `ss_number` を `employee` テーブルに追加し、テーブル全体でこのカラム内の各値をユニークにします。

```
ALTER TABLE employee
ADD ss_number char(11) UNIQUE
```

例 2

次の例では、3つのカラムからユニークエントリを作成します。

```
ALTER TABLE product
ADD UNIQUE (name, size, color)
```

カラムに対する IQ UNIQUE 制約

IQ UNIQUE は、カラムの予期されるカーディナリティを定義し、カラムを Flat FP または NBit のどちらとしてロードするかを決定します。

IQ UNIQUE は、カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(*n*) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT オプションおよび FP_NBIT_LOOKUP_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

- FP_NBIT_AUTOSIZE_LIMIT は、NBit としてロードする、重複しない値の数を制限します。
- FP_NBIT_LOOKUP_MB は、NBit ディクショナリの合計サイズのスレッシュホールドを設定します。
- FP_NBIT_ROLLOVER_MAX_MB は、NBit からフラット FP への暗黙的な NBit ロールオーバーで使用するディクショナリサイズを設定します。
- FP_NBIT_ENFORCE_LIMITS は、NBit ディクショナリのサイズ制限を強制します。このオプションはデフォルトで OFF になっています。

IQ UNIQUE を FP_NBIT_AUTOSIZE_LIMIT 未満の *n* 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が FP_NBIT_AUTOSIZE_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

カラムに対する CHECK 条件

CHECK 条件を使用して、カラム内の値の基準を指定します。

この条件では、妥当性保持のためにデータが満たすべきルール、または組織のポリシーや手順を反映したルールを指定することもできます。

個々のカラム値に対する CHECK 条件は、カラム内の値を一定の範囲に収める必要がある場合に便利です。

例 1

入力は、決められた値のいずれか 1 つと一致しなければなりません。たとえば、city カラムに事業所が所在する市区町村など、一定数の許容市区町村以外は入力できないようにするには、次に示すような制約を使用します。

```
ALTER TABLE office
MODIFY city
CHECK ( city IN ( 'city_1', 'city_2', 'city_3' ) )
```

デフォルトでは、**CASE RESPECT** オプションを使用し、明示的に大文字と小文字を区別するものとして作成したデータベースでないかぎり、文字列比較で大文字と小文字は区別されません。

例2

日付または数字は、特定の範囲内に収まる必要があります。たとえば、employee テーブルの **start_date** カラムの日付が、会社創立日から今日の日付までの範囲内となるようにする必要がある場合は次のように指定できます。

```
ALTER TABLE employee
MODIFY start_date
CHECK ( start_date BETWEEN '1983/06/27'
AND CURRENT DATE )
```

使用できる日付フォーマットはいくつかあります。YYYY/MM/DD 形式は、現在のオプション設定に関係なく必ず認識されます。

ユーザ定義データ型に対する CHECK 条件

ユーザ定義のデータ型に CHECK 条件を付加できます。そのデータ型で定義されたカラムは CHECK 条件を継承します。そのカラムに対して明示的に設定された CHECK 条件は、ユーザ定義のデータ型で設定した CHECK 条件よりも優先されません。

ユーザ定義データ型に CHECK 条件を設定すると、CHECK 条件が評価されるときに、先頭に @ の付いた変数はカラムの名前と置き換えられます。たとえば、次に示すユーザ定義データ型は正の整数しか受け付けません。

```
CREATE DATATYPE posint INT
CHECK ( @col > 0 )
```

@col ではなく @ のプレフィックスを付けた任意の変数名を使用できます。posint データ型を使用して定義されたカラムは、別の CHECK 条件が明示的に指定されていないかぎり、正の整数だけを受け付けます。

ALTER TABLE 文とともに **DELETE CHECK** 句を使用して、ユーザ定義のデータ型から継承した CHECK 条件をはじめとするすべての CHECK 条件を、テーブル定義から削除できます。

ユーザ定義データ型の詳細については、『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』を参照してください。

カラムに対する CHECK 条件

CHECK 条件は、1つのカラムに対する制約ではなくテーブル全体への制約として適用できます。

通常、このような CHECK 条件を使って、入力または変更された1つのロー内の2つの値が相互に適切な関係を持っているかをチェックします。カラム CHECK 条件はシステムテーブルに個別に格納されます。それぞれの条件は、個別に置換したり削除することができます。可能なかぎり、個々のカラムに対して CHECK 条件を使用することをおすすめします。

CHECK 条件の追加

テーブルまたは個々のカラムに新しい CHECK 条件を追加できます。

たとえば、図書館データベースでは、特定のエントリの返却日 `date_returned` は貸出日 `date_borrowed` 以降である必要があります。

```
ALTER TABLE loan
ADD CHECK (date_returned >= date_borrowed)
```

CHECK 条件の削除

個々のカラムから CHECK 条件を削除できます。

カラムの CHECK 条件を削除するには、CHECK 条件に NULL に設定します。

次の文では、**customer** テーブルの **phone** カラムから CHECK 条件が削除されます。

```
ALTER TABLE customer MODIFY phone
CHECK NULL
```

CHECK 条件の置換

個々のカラムから CHECK 条件を削除できます。

CHECK 条件を追加するときと同じ方法で、CHECK 条件を置き換えることができます。

次の文では、**office** テーブルの **city** カラムに対して、CHECK 条件の追加または置き換えが実行されます。

```
ALTER TABLE office
MODIFY city
CHECK ( city IN ( 'city_1', 'city_2', 'city_3' ) )
```

テーブルに対する CHECK 条件

テーブルに定義されている CHECK 条件を修正するには、2とおりの方法があります。

- **ALTER TABLE** と **ADD** テーブル制約句を使用して、新しい CHECK 条件を追加します。
- **ALTER TABLE DELETE CHECK** を使って、カラムの CHECK 条件を含むすべての CHECK 条件を削除し、新しく CHECK 条件を追加します。

テーブルからカラムを削除しても、そのテーブルの制約に含まれ、そのカラムに関係する CHECK 条件は削除されません。制約が削除されていない場合、テーブルのデータに対してクエリを実行しようとするとき、「column not found」というエラーメッセージが返されます。

テーブルの CHECK 条件の削除

DELETE CHECK 句を指定して **ALTER TABLE** 文を使用することにより、テーブルに対するすべての CHECK 条件を削除します。削除される CHECK 条件には、すべてのカラムに対する CHECK 条件とユーザ定義データ型から継承された CHECK 条件が含まれます。

例：

```
ALTER TABLE table_name
DELETE CHECK
```

エンティティと参照整合性

各テーブルにプライマリキーを指定すると、クライアントアプリケーションの開発者やデータベース管理者が、エンティティ整合性を管理するために実行しなければならない操作はありません。

テーブル所有者は、テーブルの作成時にそのプライマリキーを定義します。テーブルの構造を後日変更する場合は、**ALTER TABLE** 文の **DELETE PRIMARY KEY** 句または **ADD PRIMARY KEY** 句を使用して、プライマリキーの再定義もできます。『リファレンス：文とオプション』を参照してください。

アプリケーション開発システムやデータベース設計ツールには、データベーステーブルを作成したり修正したりできるものがあります。そのようなシステムを使用する場合、**CREATE TABLE** コマンドまたは **ALTER TABLE** コマンドを明示的に入力する必要はないことがあります。指定した情報に基づいて、アプリケーションによって文が生成されます。

テーブルローを挿入または更新すると、データベースサーバによってそのテーブルのプライマリキーの有効性が確実に保持されます。つまり、テーブル内の各ローがプライマリキーによりユニークに識別されます。

例 1

デモデータベースの **Employees** テーブルでは、employee ID がプライマリキーとして使用されます。新しい従業員がこのテーブルに追加されると、IQ によって新しい employee ID の値がユニーク、かつ NULL でないことがチェックされます。デモデータベースのテーブルのリストについては、『SAP Sybase IQ の概要』の「テーブル名とテーブル所有者」を参照してください。

例 2

デモデータベースの **SalesOrderItems** テーブルでは、2つのカラムを使用して1つのプライマリキーを定義します。

このテーブルには注文可能な製品に関する情報が格納されています。一方のカラムには、注文を特定する **id** が格納されていますが、1件の注文に複数の製品が存在することもあるため、このカラム自体をプライマリキーにすることはできません。追加の **line_id** カラムで、製品に対応する明細が特定されます。カラム **id** と **line_id** を結合することで製品がユニークに指定され、プライマリキーが構成されます。これはマルチカラムプライマリキーと呼ばれます。

クライアントアプリケーションがエンティティ整合性に違反する場合

エンティティ整合性では、プライマリキーまたは一意性制約の値がテーブル内でユニークであること、かつキーに NULL 値を含まないことが必要です。

クライアントアプリケーションがプライマリキー値を挿入または更新するときに値がユニークでない場合は、エンティティ整合性違反になります。

エンティティ整合性違反があると、新しい情報はデータベースに追加されず、クライアントアプリケーションにエラーが返ります。

このエラーをユーザに知らせる方法や、ユーザに適切な処置を実施させる方法は、アプリケーションプログラマが判断してください。ここでの適切な処置とは、プライマリキーに対してユニークな値を指定するだけです。

SAP Sybase IQ は、外部キーまたは候補キーに基づく各 **UPDATE**、候補キーに基づく各 **DELETE**、外部キーに基づく各 **LOAD/INSERT** について、参照整合性をチェックします。参照整合性違反が発生すると、**UPDATE** 要求または **DELETE** 要求は直ちに拒否されてロールバックされます。参照整合性に違反する **LOAD/INSERT** 要求も、拒否またはロールバックされます。SAP Sybase IQ では、ユーザ指定のデータ整合性に違反するローを拒否するようにオプションで指定できます。

参照整合性

外部キーのエントリは、参照先のテーブルのローのプライマリキー値に対応していなければなりません。

場合によっては、プライマリキー以外のカラムの組み合わせが参照先になります。一意性制約を持つプライマリキーまたはカラム (単一または複数) は、「**candidate key**」と呼ばれます。参照されるカラム (単一または複数) は候補キーでなくてはなりません。このカラムは「**referenced key**」と呼ばれます。

参照整合性の損失

SAP Sybase IQ には、参照整合性損失に対する保護機能が備わっています。

次のような場合、データベースの参照整合性が失われる可能性があります。

- プライマリキー値が更新または削除され、それに対応する外部キー値がある場合。そのプライマリキーを参照するすべての外部キーが参照整合性違反となります。
- 外部テーブルに新しいローを追加し、外部キーに入力した値に対して、対応する候補キー値がない場合。データベースは参照整合性違反となります。

SAP Sybase IQ には、参照整合性が失われるこれら 2 つの事態の両方に対する保護が備わっています。

参照されている候補キーが更新または削除されるときには、SAP Sybase IQ は、**UPDATE** や **DELETE** を許可しません。

外部キー

外部キーの作成には、プライマリキーと同様に、**CREATE TABLE** 文または **ALTER TABLE** 文を使用します。

注意： ローカルテンポラリテーブルに対しては外部キー制約を作成できません。グローバルテンポラリテーブルは、**ON COMMIT PRESERVE ROWS** で作成してください。

デモデータベースには、employee テーブルと department テーブルが含まれています。employee テーブルのプライマリキーは employee ID、department テーブルのプライマリキーは department ID です。

たとえば、次のようなスキーマがあるとします。

```
DEPT table
{ DeptNo int primary key
  DeptName varchar(20),
  Mgr int,
  foreign key MGR_EMPNO (Mgr) references EMPLOYEE(EmpNo) on update
  restrict }
```

```
EMPLOYEE table
{ EmpNo int primary key,
  DeptNo int references DEPT(DeptNo) on delete restrict,
  LastName varchar(20),
  FirstName varchar(20),
  Salary int }
```

この employee テーブルで、department ID は、department テーブルに対する外部キーです。employee テーブルの各 department ID は、department テーブルの department ID と厳密に一致しています。

外部キーの関係は、多対1の関係で、この例では必須です。employee テーブルの中には、同じ department ID エントリを含む複数のエントリがありますが、department ID は department テーブルのプライマリーキーであり、ユニークです。重複したエントリを含む department テーブルのカラムを外部キーが参照できるとすると、department テーブルのどのローが正しい参照先になるのかがわからなくなります。これは必須の外部キーです。

参照整合性違反

SAP Sybase IQ は、RESTRICT アクション付きの参照整合性 (ANSI デフォルト) を文レベルでサポートしています。

つまり、SAP Sybase IQ は、プライマリーキーや一意性制約のあるカラムに対する更新と削除の要求が、対応する外部キーが依存する値を削除するようなものである場合、その要求を拒否します (削除と更新を要求する順番には注意してください)。SAP Sybase IQ はエラーメッセージを返し、参照整合性に違反するロード処理をロールバックしますが、特定のローを無視するように指定することは可能です。

強制力のない既存の外部キーによる参照整合性の適用

強制力のない外部キーを使用して参照整合性を確保できます。

次のようなスキーマがあるとします。

```
DEPT table
{ DeptNo int primary key
  DeptName varchar(20),
  Mgr int,
  foreign key MGR_EMPNO (Mgr) references EMPLOYEE(EmpNo) on update
  restrict }
```

```
EMPLOYEE table
{ EmpNo int primary key,
  DeptNo int references DEPT(DeptNo) on delete restrict,
  LastName varchar(20),
  FirstName varchar(20),
  Salary int }
```

1. 候補キーと外部キーの関係を特定します。

前述のスキーマでは、この関係には次の2つがあります。

- 外部キー (EMPLOYEE.DeptNo) から候補キー (DEPT.DeptNo)
 - 外部キー (DEPT.Mgr) から候補キー (EMPLOYEE.EMPNo)
2. プライマリキー、または候補キーの一意性制約がない場合は、**ALTER TABLE** 文を使用して追加します(前述の例では、プライマリキーが既に存在します)。候補キーの値はすべて、ユニークでかつ null 以外でなくてはなりません。
 3. 強制力のない外部キー制約があれば削除します。次に例を示します。

```
ALTER TABLE DEPT DROP FOREIGN KEY MGR_EMPNO;
ALTER TABLE EMPLOYEE DROP FOREIGN KEY DEPT;
```

このスキーマでは、DEPT(DeptNo) を参照する、強制力のない外部キー制約 MGR_EMPNO および EMPLOYEE(DeptNo) を削除する必要があります。EMPLOYEE(DeptNo) から DEPT(DeptNo) に対し、ユーザ指定の役割名がない場合は、デフォルトの役割名はプライマリテーブルと同じ(つまり DEPT)です。

4. 外部キー制約を追加します。次に例を示します。

```
ALTER TABLE DEPT ADD FOREIGN KEY MGR_EMPNO (Mgr) REFERENCES
EMPLOYEE (EmpNo);
ALTER TABLE EMPLOYEE ADD FOREIGN KEY EMP_DEPT (DeptNo) REFERENCES
DEPT (DeptNo);
```

新しいテーブルでの参照アクティビティの適用

新しいテーブルに参照整合性を適用します。

1. プライマリテーブルを作成します。次に例を示します。

```
CREATE TABLE DEPT (DeptNo int primary key,
DeptName varchar(20),
Mgr int );
```

2. 外部テーブルを作成します。たとえば、次の文では、指定した外部キーのデフォルトのロール名は DEPT です。

```
CREATE TABLE EMPLOYEE (EmpNo int primary key,
DeptNo int REFERENCES DEPT (DeptNo)
ON DELETE RESTRICT,
LastName varchar(20),
FirstName varchar(20),
Salary int);
```

3. 外部キー制約を指定します。次に例を示します。

```
ALTER TABLE DEPT ADD FOREIGN KEY EMP_DEPT (DeptNo)
REFERENCES DEPT (DeptNo);
```

この文では、同じ外部キーに対するユーザ指定のロール名は EMP_DEPT です。

外部キーの削除

テーブルから外部キーを削除します。

1. プライマリテーブルと外部テーブルを作成します。次に例を示します。

データ整合性の確保

```
CREATE TABLE DEPT(DeptNo int primary key,  
DeptName varchar(20),  
Mgr int );
```

```
CREATE TABLE EMPLOYEE(EmpNo int primary key,  
DeptNo int references DEPT(DeptNo)  
on delete restrict,  
LastName varchar(20),  
FirstName varchar(20),  
Salary int);
```

2. ロール名が割り当てられていない場合は、指定した外部キーのデフォルトのロール名は DEPT です。

```
ALTER TABLE EMPLOYEE DROP FOREIGN KEY DEPT;
```

3. 外部キーが複数あり、ロール名が不明の場合は、**sp_iqconstraint** プロシージャを使用してロール名を表示できます。『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』を参照してください。

4. ロール名としてたとえば EMP_DEPT を割り当てた場合は、キーの削除時にその名前を次のように指定します。

```
ALTER TABLE EMPLOYEE DROP FOREIGN KEY EMP_DEPT;
```

これらの文では、暗黙的に作成される、EMPLOYEE(DeptNo) の非ユニークな HG インデックスは削除されません。これを削除するには、次のように、**sp_iqindex** を使用して HighGroup インデックス名を見つけ、**DROP INDEX** 文を使用します。

```
sp_iqindex('EMPLOYEE');
```

```
EMPLOYEE DBA DeptNO FP ASIQ_IDX_T27_C2_FP N  
EMPLOYEE DBA DeptNO HG ASIQ_IDX_T27_C2_HG N  
EMPLOYEE DBA EmpNO FP ASIQ_IDX_T27_C1_FP N  
EMPLOYEE DBA EmpNO HG ASIQ_IDX_T27_I11_HG N  
EMPLOYEE DBA FirstName FP ASIQ_IDX_T27_C4_FP N  
EMPLOYEE DBA LastName FP ASIQ_IDX_T27_C3_FP N  
EMPLOYEE DBA Salary FP ASIQ_IDX_T27_C5_FP N
```

```
DROP INDEX ASIQ_IDX_T27_C2_HG
```

テーブルを削除するには、関連付けられている外部キー制約をすべて削除する必要があります。外部キー制約とテーブルはこの順序で削除します。

```
ALTER TABLE DROP FOREIGN KEY MGR_EMPNO;  
DROP TABLE EMPLOYEE;  
DROP TABLE DEPT;
```

これらのテーブルを削除するもう 1 つの方法としては、次の 2 つの **ALTER TABLE** 文を任意の順序で実行してから、**DROP TABLE** 文を任意の順序で実行するやり方もあります。

```
ALTER TABLE DEPT DROP FOREIGN KEY MGR_EMPNO;  
ALTER TABLE EMPLOYEE DROP FOREIGN KEY EMP_DEPT;
```

データベースに、事業所の所在地をリストする office テーブルが含まれているとします。employee テーブルでは、office テーブルに対する外部キーを作成し、それ

それぞれの従業員が勤務する事業所を示すこともできます。ただし、データベース設計者は、従業員が採用された時点でまだ勤務地が決定していない場合を想定する必要があります。この場合、勤務地が不明または未決定の場合に備えて、外部キーに NULL 値を指定できるように設定してください。

同時処理

SAP Sybase IQ の参照整合性では、外部テーブルに対するロードまたは挿入の間は、プライマリテーブルに対する更新または削除の同時処理が制限されます。

表 18 : エラーが返される同時処理

最初の要求	並行するトランザクションの要求
1つのトランザクションによる次のいずれかの要求: LOAD/INSERT/UPDATE/ ALTER TABLE ADD 外部キー/外部 テーブルへの ALTER TABLE DROP 外部キー	削除可能なローを含む、関連付けられた プライマリテーブルに対する DELETE
	関連付けられたプライマリテーブルに 対する UPDATE
	関連付けられたプライマリテーブルに 対する TRUNCATE

SAP Sybase IQ は、古いバージョンの外部テーブル、プライマリテーブル、またはその両方を使用しているトランザクションがあるときに、他のトランザクションが外部キーの追加や外部キーの削除によるテーブルの変更を要求した場合も、エラーを生成します。

強制力のある、または強制力のない外部キーとプライマリキーの両方について、SAP Sybase IQ では以下が実行できます。

- 1つ以上の外部テーブルと共有プライマリテーブルに対するロードまたは挿入の同時実行。
- 外部テーブルに対するロードまたは挿入と、別の1つ以上の外部テーブルでの削除、更新、またはトランケートの同時実行。
- 複数の外部テーブルに対する削除、更新、またはトランケートの同時実行 (同じプライマリテーブルを共有している場合でも可能)。
- 外部テーブルに対する削除、更新、またはトランケートと、共有されているプライマリテーブルに対する削除、更新、またはトランケートの同時実行。
- **ALTER TABLE ADD** 外部キーまたは **DROP** 外部キー (古いバージョンの外部テーブル/プライマリテーブルを使用しているトランザクションがなく、それらの古いテーブルが外部キーの **ADD/DROP** 処理の一部として削除される場合)。

外部テーブルとプライマリテーブルに対する同時操作

SAP Sybase IQ では、テーブルレベルのバージョン管理により、一貫した参照整合性チェックを実現しつつ、外部テーブルに対するロード、挿入または更新処理と、

データ整合性の確保

プライマリテーブルに対する **LOAD/INSERT** 処理の同時実行が可能となっています。

また、SAP Sybase IQ では、**DELETE** または **UPDATE** を要求するトランザクションの開始時に、削除された古い値が外部テーブルに存在しないことが確認されます。これにより、外部テーブルに対する削除処理とプライマリテーブルに対する削除または更新処理の同時実行のときに、一貫した参照整合性チェックが実現されません。

ftab1 と ftab2 という 2 つの外部テーブルと、ptab という 1 つのプライマリテーブルの間に、2 つの外部キー制約があるとします。外部キー ftab1 (fk1, fk2) は候補キー ptab (pk1, pk2) を参照します。外部キー ftab2 (fk1, fk2) も同じ候補キーを参照します。候補キー ptab (pk1, pk2) はプライマリキーまたは一意性制約のいずれかです。

次の表は、外部テーブルとプライマリテーブルの両方に対して、許可される操作とエラーを返す操作を示します。この表の情報は、強制力のある外部キーと候補キーにのみ当てはまります。

表 19 : 外部テーブルとプライマリテーブルに対する同時 DML

	LOAD/ INSERT ftab1	DE- LETE/ TRUN- CATE TABLE ftab1	UPDATE ftab1 (fk1,fk2)	新しいイン デックス非 FK ftab1 (fk1,fk2) の 挿入	ADD FK ftab1 (fk1, fk2)	DROP FK ftab1 (fk2, fk2)
LOAD ftab2	可能	可能	可能	可能	可能	可能
LOAD ptab	可能	可能	可能	可能	可能	可能
INSERT ftab2	可能	可能	可能	可能	可能	可能
INSERT ptab	可能	可能	可能	可能	可能	可能
DELETE/ TRUN- CATE TABLE ftab2	可能	可能	可能	可能	可能	可能
DELETE ptab TRUNCATE TABLE ptab	エラー	可能	エラー	可能	エラー	エラー
UPDATE ftab2(fk1,fk2)	可能	可能	可能	可能	可能	可能
UPDATE ptab (pk1,pk2)	エラー	可能	エラー	可能	エラー	エラー

	LOAD/ INSERT ftab1	DE- LETE/ TRUN- CATE TABLE ftab1	UPDATE ftab1 (fk1,fk2)	新しいイン デックス非 FK ftab1 (fk1,fk2) の 挿入	ADD FK ftab1 (fk1, fk2)	DROP FK ftab1 (fk2, fk2)
新しいインデック スの挿入	可能	可能	可能	可能	可能	可能
QUERY ((fk1,fk2) の有無にかかわら ず使用されている、 古いバージョンの ftab1/ptab)	可能	可能	可能	可能	エラー	エラー
古いバージョンの ftab2 は使用されて いない	該当なし	該当なし	該当なし	該当なし	可能(使 用されて いない古 いバー ジョンの ftab1 は すべて削 除)	可能(使用 されてい ない古い バージョ ンの ftab1 はすべて 削除)

同時実行の競合が発生するのは、あるトランザクションが外部キーカラムへのロードを実行するのと並行して、別のトランザクションがそれに関連付けられた候補キーカラムを更新する場合です。あるトランザクションが外部キーカラムへのロードを実行するのと並行して、別のトランザクションが、それに関連付けられた候補テーブルのうちで、それとは関連付けられていない候補キーカラムを更新する場合には、競合は発生しません。

注意： union all ビューに対するクエリでは、パフォーマンスの効率化のために、ジョインキーやグループ化カラムとして使用されているカラムが参照するテーブルがオープンされます。トランザクションがコミットされ、テーブルの読み込みロックが解放されるまでは、外部キーがジョイン条件またはグループ化カラムとして使用されているテーブルの変更や削除はできません。一方、これらのテーブルに対するロード、挿入、削除、更新は、クエリの実行中でも可能です。

参照整合性チェックの無効化

SAP Sybase IQ のオプション `DISABLE_RI_CHECK` を使用することで、参照性合成チェックを回避できます。

参照整合性チェックを回避すると、この機能の存在意義がなくなるため、このオプションは慎重に使用してください。

システムテーブルの整合性ルール

データベースにおける整合性検査とルールに関する情報はすべて、システムテーブルとシステムビューに格納されています。

システムテーブル	説明
SYS.SYSTABLE	CHECK 制約は、SYS.SYSTABLE の view_def カラムに保持されている。ビューに関しては、view_def に、ビューを作成する CREATE VIEW コマンドが保持されている。table_type カラムを調べることで、特定のテーブルがベーステーブルとビューのどちらであるかを確認できる。値が BASE の場合はベーステーブル、 VIEW の場合はビューである。
SYS.SYSFOREIGNKEYS	このビューは、SYS.SYSFOREIGNKEY テーブルと SYS.SYSFKCOL テーブルの外部キー情報をより見やすい形式で表示する。
SYS.SYSCOLUMNS	このビューは、SYS.SYSCOLUMN テーブルの情報をより見やすい形式で表示する。このビューで表示されるカラムの情報は、デフォルト設定とプライマリーキーである。

Interactive SQL を使用して、これらのテーブルやビューをブラウズできます。

カタログストアインデックスの検証

カタログストアでは、SQL Anywhere のテーブルのインデックスを検証して、インデックスで参照されているすべてのローがテーブル内で実際に機能しているかどうかを確認できます。

インデックスを検証するには、コマンドプロンプトを開いて **dbvalid** ユーティリティを実行します。

たとえば次の文 (1 行で入力する必要があります) は、EmployeeIndex というインデックスを検証します。-i スイッチは、指定したオブジェクト名がインデックスであることを表します。

```
dbvalid -c "uid=dba;pwd=sql;eng=myserver"
-i EmployeeIndex
```

次のステップ

『ユーティリティガイド』の「dbvalid データベース管理ユーティリティ」を参照してください。

トランザクションとバージョンの管理

トランザクション処理では、論理的に関連した複数のコマンドを1つの単位として実行する必要があります。SAP Sybase IQ は、更新中のデータベースから多数のユーザが読み取りを実行できるように、トランザクション処理を使用します。トランザクションは、データの精度の管理、システム障害時のデータリカバリに必要なものです。

トランザクション処理の重要な面として、あるユーザのトランザクションの影響が他のユーザに及ばないようにする機能があります。SAP Sybase IQ のトランザクション処理の方法は、スナップショットバージョン管理と呼ばれ、ISO によって承認された最高位の独立性レベルをサポートします。スナップショットバージョン管理は、各データベースユーザにデータベースのスナップショットを提供します。データベースへの変更は、トランザクションをコミットするまで他のデータベースユーザには表示されません。

参照：

- テーブルレベルのスナップショットバージョン管理 (239 ページ)
- ローレベルのスナップショットバージョン管理 (246 ページ)

トランザクション

各トランザクションは論理的に関連した一連のコマンドで、1つのタスクを実行し、データベースをある一貫した状態から別の状態へ変換します。

トランザクションはアトミックです。SAP Sybase IQ は、トランザクション内のすべての文を1単位として実行します。各トランザクションの最後で、変更をコミットして確定します。トランザクション内のコマンドのうち、正しく処理されないものがあつた場合は、それまで行った変更を取り消す、つまりロールバックできます。ユーザアプリケーションでは、変更のコミットまたはロールバックを行う条件を制御します。

トランザクションは小さなブロックで構成されています。各ブロックが完了すると、情報が一貫しているポイントにマーク付けされます。トランザクション処理は、データベースに正しい情報が入るようにするために必要なものです。

ほとんどの場合、SAP Sybase IQ のトランザクションは、発行されたコマンドとオプションセットに基づいて、自動的に開始/終了します。明示的にコマンドを発行してトランザクションを開始/終了することもできます。

トランザクションは、次のいずれかに続いて最初の文から自動的に開始します。

トランザクションとバージョンの管理

- データベースへの接続
- 前のトランザクションの終了

SAP Sybase IQ では Adaptive Server との互換性のため、**BEGIN TRANSACTION** など、SAP Sybase IQ の Transact-SQL コマンドもサポートしています。**BEGIN TRANSACTION** コマンドを使用してトランザクションを明示的に開始できます。

トランザクションは次のいずれかのイベントで完了します。

- データベースの変更を確定する **COMMIT** 文
- トランザクションで行われたすべての変更を取り消す **ROLLBACK** 文
- データベースの接続解除時、ロールバック (デフォルト) またはコミットが暗黙的に行われる
- オートコミットが実行される文の実行でデータベースへの変更が行われる

データベース定義コマンドの **ALTER**、**CREATE**、**DROP** などとはすべて自動的にコミットを実行します。

セーブポイントを使用して、トランザクション内の重要な状態を識別し、その状態に選択して戻ったり、他のアクションを発生させたりできます。

参照：

- CHECKPOINT 文 (477 ページ)
- SAVEPOINT 文 (546 ページ)

トランザクションアクティビティの表示

sp_iqtransaction ストアドプロシージャを使用して、作成されて使用されているメイン領域とテンポラリ領域、オープンカーソル、セーブポイントなどのトランザクションアクティビティのスナップショットを表示します。

sp_iqtransaction を実行します。

IQ トランザクションマネージャの各トランザクション制御ブロックについて 1 つのローを返します。

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_iqllocks** 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。
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>

カラム名	説明
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。
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)。

カラム名	説明
TempWorkSpaceKB	<p>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。</p> <p>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。</p>
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。
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。

トランザクションとバージョンの管理

カラム名	説明
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      MainTableKBCr      TempTableKBCr      TempTableKBCr
=====  =====  =====  =====
              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			

独立性レベル

トランザクション処理の重要な一面は、操作を隔離するデータベースサーバの機能です。ANSI 規格では、4つの独立性レベルを定義しています。レベルが高いほど、トランザクションの他のトランザクションからの独立性は高くなり、データベースの内部的な一貫性が保持される保証が高くなります。

あるトランザクション内の操作やデータが、他の同時実行のトランザクション内でどの程度参照できるかを、独立性レベルで制御します。テーブルレベルのスナップショットバージョン管理では最高レベルの独立性をサポートしています。このレベルでは、すべてのスケジュールを直列化できます。

テーブルレベルのスナップショットバージョン管理は、次の規則に従って同時実行のトランザクション間で高レベルの独立性を保持します。

- トランザクション管理は、各トランザクションの開始時にコミットされたデータのスナップショットを保持する。
- トランザクションは、使用するスナップショットのバージョンが保持されているかぎり、常に読み取りが可能。
- トランザクションの書き込みは、そのトランザクションが参照するスナップショットに反映される。
- トランザクションが開始すると、そのトランザクションは他のトランザクションが行った更新を参照できない。

テーブルレベルのスナップショットのバージョン管理のこの独立性レベルでは、数種類の矛盾が回避されます。最も頻繁に発生する矛盾を以下に示します。

- **ダーティリード** – トランザクション A がオブジェクトを修正し、変更のコミットもロールバックもしないとします。トランザクション B がその変更したオブジェクトを読み込みます。次に、トランザクション A がさらにそのオブジェクトを変更してから、**COMMIT** を実行します。この状況では、トランザクション B が参照したオブジェクトは、コミットされていない状態のものとなります。
- **繰り返し不可能読み出し** – トランザクション A がオブジェクトを読み込みます。次にトランザクション B がそのオブジェクトを変更または削除して、**COMMIT** を実行します。トランザクション A がもう一度そのオブジェクトを読み込もうとしたときには、オブジェクトは変更されているか、削除されてしまっています。
- **幻データ要素** – トランザクション A が、一定の条件を満たすデータのセットを読み込みます。トランザクション B が次に **INSERT**、**COMMIT** の順に実行します。この新しいコミットされたデータは、先程の条件を満たします。ここでトランザクション A がもう一度データを読み込むと、取得するデータセットは最初のものとは異なります。

- **更新内容の消失** – カーソルを使用するアプリケーションで、トランザクション A がデータセットに対する変更を書き込みます。トランザクション B が次に以前のデータを基に行った更新を保存します。トランザクション A の変更は完全に消失します。

テーブルレベルのスナップショットバージョン管理では、ある時点でテーブルを変更できるのは 1 人のユーザだけです。また、変更が完了するまでの間、その変更を他のユーザは参照できません。いつの時点でも使用中のデータオブジェクトのタイムスタンプのあるスナップショットを保持します。

IQ では独立性レベルを 0、1、2、または 3 (ANSI レベル 1、2、3、または 4 に相当) に設定できますが、実際に設定する必要はありません。4 以外のレベルに設定したとしても、すべてのユーザは独立性レベル 4 で実行します。低い独立性レベルを設定してもパフォーマンス上の利点はありません。

同時トランザクションによるテーブルへのアクセスまたは変更を防ぐ方法の詳細については、『リファレンス：文とオプション』の「**LOCK TABLE 文**」を参照してください。

トランザクションのブロック

BLOCKING オプションを On に設定すると、あるトランザクションが取得しようとした書き込みロックが既存の書き込みロックと競合している場合、競合するロックが解放されるか、**BLOCKING_TIMEOUT** スレッショルドに達するまで、そのトランザクションは待機状態となります。**BLOCKING** のデフォルトは OFF です。

接続ブロックは、ロックを保持する接続がロックを解放する (テーブルレベルまたはローレベルのロックをブロック解除する) まで、要求トランザクションをスリープ状態にします (テーブルレベルまたはローレベルのロックをブロック)。

BLOCKING_TIMEOUT 値をミリ秒単位で設定し、ブロック期間を制御します。

BLOCKING_TIMEOUT が 0 (デフォルト) の場合、接続内でブロックされているトランザクションはすべて、要求されたロックを接続が獲得するまで無限に待機します。**BLOCKING_TIMEOUT** が設定され、ロックが指定期間内に解放されなかった場合、待機しているトランザクションはエラーメッセージを受信します。

ブロッキングは、テーブルレベルまたはローレベルのロックを構築した後にトランザクションマネージャがトランザクションスナップショットバージョンを作成するという遅延トランザクションバージョン管理を利用しています。

次に、テーブルレベルロックのブロック例を示します。

- ユーザ A が接続する。
- ユーザ A が `INSERT INTO iq_table1 VALUES (3, 300);` を実行する。
- ユーザ A が接続する。

- ユーザ B が `INSERT INTO iq_table1 VALUES (4, 400);` を実行する。

接続ブロックが無効であれば、ユーザ B のトランザクションはロールバックされ、別のユーザが `iq_table1` をロックしていることを示すエラーがユーザ B に届きます。

接続ブロックが有効である (および `BLOCKING_TIMEOUT` が 0 に設定されている) 場合、ユーザ B の `INSERT` 文は、ユーザ A の `INSERT` 文がコミットされ、書き込みロックが解放されるまでスリープ状態になります。

接続ブロックが有効で `BLOCKING_TIMEOUT` が 200 ミリ秒に設定されている場合、ユーザ A のトランザクションがブロックタイムアウトスレッシュولدである 200 ミリ秒以内にコミットされなければ、ユーザ B のトランザクションはロールバックされ、別のユーザが `iq_table1` をロックしていることを示すエラーがユーザ B に届きます。

ローレベルロックの例については、『管理：インメモリ、ローレベルのバージョン管理』の「RLV ストアのブロッキング管理」を参照してください。

接続ブロッキングの有効化

接続ブロッキングを有効化すると、別のトランザクションの既存のロックと競合するロックを獲得しようとするトランザクションを、すべての競合ロックが解放されるか、`BLOCKING_TIMEOUT` スレッシュولدに達するまで強制的に待機させます。

前提条件

- SAP Sybase IQ サーバにシンプレックスデータベースがある。

手順

BLOCKING データベースオプションを ON に設定します。

```
set temporary option blocking = 'On';
```

注意：ブロッキングオプションは、接続レベルまたは PUBLIC レベルで設定できません。

接続ブロッキングの無効化

接続ブロッキングを無効にすると、別のトランザクションの既存のロックと競合するロックを獲得しようとするトランザクションに対してトランザクションのロールバックとエラーの表示が強制されます。

前提条件

- SAP Sybase IQ サーバにシンプレックスデータベースがある。

手順

BLOCKING データベースオプションを OFF に設定します。

```
set temporary option blocking = 'Off';
```

注意：ブロッキングオプションは、接続レベルまたは PUBLIC レベルで設定できません。

BLOCKING オプション

ロック競合に対する動作を制御します。BLOCKING は、マルチプレックスのセカンダリノードではサポートされません。

指定できる値

ON、OFF

デフォルト値

OFF

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

BLOCKING が OFF の場合、トランザクションが書き込み処理を試みて、それが別のトランザクションの読み込みロックによってブロックされた場合に、トランザクションにエラーが返されます。

BLOCKING が ON の場合、別のトランザクションが保持する既存のロックと競合するロックを取得しようとした場合、競合するすべてのロックが解放されるか、`blocking_timeout` に達するまで、トランザクションは待機します。`blocking_timeout` ミリ秒以内にロックが解放されない場合は、待機しているトランザクションにエラーが返されます。

ブロッキングタイムアウトスレッシュホールドの設定

スレッシュホールドを使用して、トランザクションがロックを獲得するまでの待ち時間をミリ秒単位で設定します。トランザクションが別のトランザクションの既存ロックと競合するロックを獲得しようとした場合は、**BLOCKING_TIMEOUT** オプションのスレッシュホールドに達するまで待機します。競合が引き続き存在する場合は、トランザクションがロールバックし、エラーが表示されます。

前提条件

SAP Sybase IQ サーバにシンプレックスデータベースがある。

手順

注意： デフォルト値は 0 で、ブロックされたトランザクションは時間制限なく競合トランザクションがロックを解放するまで待機する必要があることが指定されます。

BLOCKING_TIMEOUT データベースオプション値をミリ秒単位の値で設定して、競合トランザクションがロックを解放するまでのトランザクション待機時間を指定します。

```
set temporary option blocking_timeout = '400';
```

注意： ブロッキングオプションは、接続レベルまたは PUBLIC レベルで設定できません。

BLOCKING_TIMEOUT オプション

トランザクションがロックを獲得するまでの待機時間を制御します。**BLOCKING_TIMEOUT** は、マルチプレックスのセカンダリノードではサポートされません。

指定可能な値

整数値 (ミリ秒単位)

デフォルト

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せん。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

ブロッキングオプションを ON に設定すると、トランザクションが取得しようとしたロックが既存のロックと競合している場合は、設定された時間(ミリ秒単位)が経過するまで、競合するロックが解放されるのを待ちます。blocking_timeout ミリ秒以内にロックが解放されない場合は、待機しているトランザクションにエラーが返されます。

このオプションを 0 に設定すると、ロックを獲得しようとしているトランザクションはすべて、競合するトランザクションがロックを解放するまで待機します。

トランザクションのブロックによるデッドロック

トランザクションのブロックによってデッドロックが起こる可能性があります。デッドロックとは、トランザクションのまとまりで、そのどれもが処理を進行できない状態をいいます。

デッドロックが発生する理由は次の 2 つです。

- **環状ブロックの競合** – トランザクション A がトランザクション B にブロックされ、トランザクション B がトランザクション A にブロックされている状態。この状態から脱け出すには、どちらかのトランザクションをキャンセルします。同様の状況は 3 つ以上のトランザクションが環状にブロックされた場合にも発生します。

トランザクションのデッドロックを排除するために、データベースサーバはデッドロックに関わっている接続を選択し、その接続でアクティブなトランザクションの変更をロールバックして、エラーを返します。データベースサーバは内部のヒューリスティックを使用してロールバックする接続を選択し、**BLOCKING_TIMEOUT** オプションによって決められた残りのブロックの待機時間が最も短い接続を優先します。すべての接続が永久に待機するように設定されている場合は、サーバによってデッドロックが検出された接続を、犠牲にする接続として選択します。

- **すべてのワーカのブロック** – トランザクションがブロックされても、ワーカは放棄されたわけではありません。たとえば、データベースサーバには 3 つのワーカが設定されています。トランザクション A、B、C が、現在要求を実行していないトランザクション D によってブロックされます。これ以上利用でき

るワーカがないため、デッドロック状態が発生します。この状況は、スレッドデッドロックと呼ばれます。

データベースサーバに n ワーカを設定したと想定します。 $n-1$ の数のワーカがブロックされているときに最後のワーカをブロックしようとする、スレッドデッドロックが発生します。データベースサーバのカーネルは、最後のワーカをブロックすることを許可できません。ブロックすることによって、すべてのワーカがブロックされ、データベースサーバが応答を停止するためです。代わりに、データベースサーバは最後のワーカをブロックしようとしたタスクを終了し、その接続でアクティブなトランザクションの変更をロールバックして、エラーを返します。

数十または数百の接続のあるデータベースサーバでは、データベースのサイズまたはブロックが原因で、実行時間の長い要求でスレッドデッドロックが発生することがあります。この場合、`start_iq` ユーティリティの `-gn` サーバオプションの値を増やすことができます。

SAP Control Center のロックまたはデッドロックを表示するには、SAP Control Center のオンラインヘルプを参照してください。

Interactive SQL でのデッドロックレポートイベントの作成

デッドロックに関する情報を取得するための、テーブルおよびシステムイベントを作成します。

前提条件

SAP Sybase IQ サーバにシンプレックスデータベースがある。

手順

1. `sa_report_deadlocks` システムプロシージャが返すデータを格納するテーブルを作成します。

```
CREATE TABLE DeadlockDetails (
  deadlockId INT PRIMARY KEY DEFAULT AUTOINCREMENT,
  snapshotId BIGINT,
  snapshotAt TIMESTAMP,
  waiter INTEGER,
  who VARCHAR(128),
  what LONG VARCHAR,
  object_id UNSIGNED BIGINT,
  record_id BIGINT,
  owner INTEGER,
  is_victim BIT,
  rollback_operation_count UNSIGNED INTEGER );
```

2. デッドロックが発生したときに、電子メール通知を送信するイベントを作成します。

```
CREATE EVENT DeadlockNotification
```

```

TYPE Deadlock
HANDLER
BEGIN
  INSERT INTO DeadlockDetails WITH AUTO NAME
  SELECT snapshotId, snapshotAt, waiter, who, what, object_id,
  record_id,
  owner, is_victim, rollback_operation_count
  FROM sa_report_deadlocks ();
  COMMIT;
  CALL xp_startmail ( mail_user = 'John Smith',
  mail_password = 'mypwd' );
  CALL xp_sendmail( recipient='DBAdmin',
  subject='Deadlock details added to the
  DeadlockDetails table.' );
  CALL xp_stopmail ( );
END;

```

このイベントにより、**sa_report_deadlocks** システムプロシージャの結果がテーブルにコピーされ、管理者にデッドロックが通知されます。

3. **log_deadlocks** オプションを On に設定します。

```
SET OPTION PUBLIC.log_deadlocks = 'On';
```

4. 最後に実行された文のロギングを有効にします。

```
CALL sa_server_option( 'RememberLastStatement', 'YES' );
```

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。
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。

カラム名	データ型	説明
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 -zl 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。

LOG DEADLOCKS オプション

デッドロックレポートのオン/オフを制御します。

指定できる値
On、Off

デフォルト
Off

スコープ
オプションは、データベース (PUBLIC) レベルでのみ設定できます。

このオプションを設定するには、SET ANY SYSTEM OPTION システム権限が必要です。すぐに有効になります。

備考

このオプションを On に設定した場合、データベースサーバは、内部バッファ内のデッドロックに関する情報をログに記録します。バッファのサイズは、10000 バイトに固定されています。デッドロック情報は sa_report_deadlocks ストアドプロシージャを使用すると表示できます。このオプションを Off に設定すると、バッファの内容は保持されます。

デッドロックが発生すると、そのデッドロックに関係する接続のみの情報がレポートされます。接続のレポート順序は、どの接続がどの行を待っているかに応じて決定されます。スレッドデッドロックの場合、すべての接続の情報がレポートされます。

デッドロックレポートがオンになっているときは、デッドロックの発生時に Deadlock システムイベントを使用してアクションを実行することもできます。

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。

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。
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。
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。

カラム名	データ型	説明
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)

バージョン

データベースサーバは、トランザクションが作成され、トランザクションがテーブルまたはテーブルローに対する必要な書き込みロックを取得した後で、スナップショットバージョンをバインドします。テーブルレベルのバージョン管理は、単一ライターアクセスおよびテーブルレベルのロックを有効にします。ローレベルのバージョン管理(RLV)では、RLV 対応テーブルの同時ライターアクセスおよびローレベルのロックを有効にします。

テーブルレベルのスナップショットバージョン管理

デフォルトのトランザクション処理であるテーブルレベルのスナップショットバージョン管理では、複数の読み込みユーザがテーブルのオリジナルコピーに対して作業を続行している間に、複数の書き込みユーザが1つのテーブルを順番に修正できます。つまり、修正は1人ずつ行います。一度に複数の書き込みユーザが修正を行うことはできません。

テーブルレベルのスナップショットバージョン管理を使用しなければ、同時実行の読み込みや書き込み操作により、IQ メインストア内で矛盾が発生する可能性があります。SAP Sybase IQ が提供するテーブルレベルのバージョン管理では、トランザクションを直列化し、テーブルのバージョンレベルを作成することによって、矛盾を回避します。

どのようなトランザクション処理システムでもデータベースの一貫性を保証するように設計されていますが、SAP Sybase IQ の方法では、適切なトランザクション中でのクエリや更新の配置について悩む必要がなくなります。SAP Sybase IQ はトランザクションの開始や終了を自動的に行い、読み込み操作と書き込み操作が相互に干渉しないようにします。

テーブルレベルのスナップショットバージョン管理が有効で、テーブルが RLV ストレージ用に構成されている場合、同時実行の読み込みおよび書き込み操作は無効なままです。テーブルレベルのスナップショットバージョン管理の設定は、テーブルの RLV 設定より優先されます。

テーブルレベルのスナップショットバージョン管理の有効化

デフォルトでは、テーブルレベルのスナップショットバージョン管理が有効になっています。デフォルトのバージョン管理方法が変わったら、**ALLOW_SNAPSHOT_VERSIONING** データベースオプションを使用して、その方法を有効にしてください。

以下を実行することにより、テーブルレベルのバージョン管理を一時的に設定します。

```
SET TEMPORARY OPTION Snapshot_Versioning = 'Table-level'
```

テーブルレベルのロック

テーブルレベルのロックはすべて、ユーザが要求する操作タイプに基づいて自動的に発生します。

ロックを明示的に要求する必要はありません。テーブルへのアクセス権を持つトランザクションがロックを保持します。

SAP Sybase IQ でテーブルがロックされた場合、データ定義操作以外の他のトランザクションが書き込みアクセスを行うことはできません。書き込みロックされているテーブルにアクセスしようとする他の書き込みトランザクションは、エラーを受信します。読み込みアクセスはどのトランザクションでも行うことができます。

ロックは他のトランザクションが同時にアクセスすることを防止し、データベースに格納されている情報の信頼性を維持します。データベースサーバでは、あるトランザクションが取得したすべてのロックを、コミットまたはロールバックでそのトランザクションが終了するまで、保持します。

LOCK TABLE 文を使用することにより、新しいトランザクション内でテーブルのセットに対して書き込みロックを予約できます。**LOCK TABLE** 文は現在のトラン

ザクションをコミットし、ロックが使用できるようになるまで他のトランザクションをキューに入れておくことを可能にします。

DML 操作のテーブルロック

データ操作言語 (DML: Data Manipulation Language) 操作には、挿入、削除、更新、クエリなどがあります。いずれの操作の場合も、テーブルレベルのスナップショットバージョン管理は任意のテーブルに対して 1 人の書き込みユーザと複数の読み取りユーザを許可します。

テーブルレベルのスナップショットバージョン管理では、次のような動作になります。

- 読み込みトランザクションは書き込みトランザクションをブロックしない
- 書き込みトランザクションは読み込みトランザクションをブロックしない
- 1 人の更新ユーザと複数の読み込み専用ユーザが同時に同じテーブルにアクセスできる
- テーブルのデータを更新できるのは、一度に 1 人のユーザのみ

テーブルを最初に書き込みモードで開くトランザクションが、そのテーブルへのアクセス権を取得します。そのテーブルを書き込みモードで開こうとする 2 番目のトランザクションは、エラーを受信します。現在のトランザクション内のそのテーブルにさらに書き込もうとすると、失敗します。トランザクションの続行は可能ですが、読み取り操作か、他のテーブルへの書き込み操作しかできません。

SAP Sybase IQ では、共有、書き込み、および排他の各ロックをキューに入れることができるので、指定した期間テーブルをロックすることが可能です。複数のテーブルに同時に書き込みロックをかけることができます。

後で使用する DML 文でバージョンエラーが返されるのを防ぐには、**LOCK TABLE** 文を使用して、変更予定の 1 つまたは複数のテーブルに対して書き込みロックを予約します。

デッドロックの場合、通常、ブロック状態になった最後の **LOCK TABLE** 文がロールバックされ、発生したデッドロックの形式に関するエラーがそのトランザクションに返されます。

場合によっては、**COMMIT** 文または **ROLLBACK** 文を発行する必要があります。たとえば、整合性制約などの理由で DML 文が失敗した場合も、ロックを解放する際に明示的な **COMMIT** または **ROLLBACK** が必要となります。

参照先テーブルで参照整合性の違反があったため、または別のテーブルのロックが利用できないために DML 文が失敗した場合は、「SQL Anywhere Error -210」が返されます。

DDL 操作のテーブルロック

データ定義言語 (DDL: Data Definition Language) 操作には、**CREATE**、**DROP**、**ALTER** などがあります。

特定のテーブルまたはインデックスに対する DDL 操作は、すべての変更中のテーブルから他のすべての読み取りユーザや書き込みユーザをロックアウトします。この方法はクエリ結果の精度にとって重要です。この方法だと、たとえば、あるテーブルカラムが、そのカラムからのデータの選択中に、データベースから消失しないようにできます。

CREATE、**DROP**、**ALTER** の各コマンドには、次のような特徴があります。

- これらのコマンドは、他のトランザクションが、変更の対象となるテーブルまたはインデックスを使用している間は、実行できません。
たとえば、ユーザがテーブルに対して **SELECT** を発行した場合、そのテーブルはロックされ、当該ユーザがログアウトするか、他のテーブルに対して **SELECT** を発行するか、**ROLLBACK** を発行するまでは変更できません。
- これらのコマンドでは、完了時に **COMMIT** が自動的に発行されます。
- 変更中のテーブルを使用しようとする既存のトランザクションは、エラーを受信します。つまり、ユーザがあるテーブルにアクセスしている場合に、DDL コマンドがそのテーブルを変更すると、ユーザのコマンドは失敗します。
- **CREATE DBSPACE**、**DROP DBSPACE**、**CHECKPOINT** のコマンドは、1つのデータベースに対して一度に実行できるのはいずれか 1つのみです。

DDL ロックエラー

DDL コマンドの処理中に別の DDL コマンドを発行すると、エラーが発生することがあります。

次のようなエラーメッセージが表示されることがあります。

```
Cannot perform DDL command now on table <tablename> as a DDL command is already in progress on that table.
```

CREATE DBSPACE コマンドまたは **DROP DBSPACE** コマンドの処理中に、明示的に **CHECKPOINT** コマンドを発行すると、チェックポイントは失敗し、次のメッセージが返されます。

```
Run time SQL Error
```

CHECKPOINT コマンドが処理中の場合、**CREATE DBSPACE** コマンドまたは **DROP DBSPACE** コマンドを発行すると次のメッセージが返されます。

```
Cannot perform requested command as there is a CHECKPOINT command in progress.
```

削除中に **CREATE DBSPACE** を発行すると、次のメッセージが返されます。

```
Cannot perform requested command as there is a
DROP DBSPACE command in progress.
```

作成中に **DROP DBSPACE** を発行すると、次のメッセージが返されます。

```
Cannot perform requested command as there is a
CREATE DBSPACE command in progress.
```

あるトランザクションが特定のテーブルまたはインデックスに対して DDL コマンドを発行した場合、その DDL トランザクションがコミットする前に開始したトランザクションがそのテーブルにアクセスしようとする、エラーが返されます。このエラーが発生した場合、現在のトランザクション内でそのテーブルに対してさらに読み取りや書き込みを行おうとしても、すべて失敗します。

これらのインデックス作成コマンドの規則には、例外が 1 つあります。インデックス作成の影響を受けるテーブルに対して **CREATE INDEX** コマンドと **SELECT** 文を使用する場合、SAP Sybase IQ では、インデックスを作成しているトランザクションがコミットするまで、新しいインデックスは使用できません。

GRANT、**REVOKE**、**SET OPTION** の各コマンドも DDL 操作とみなされますが、同時実行による矛盾が発生しないため、制限されません。**GRANT** と **REVOKE** では常にオートコミットが発生します。**SET OPTION** では、**TEMPORARY** と指定された場合を除いて、オートコミットが発生します。**GRANT** と **REVOKE** は、データベースに現在接続中のユーザに対しては許可されません。**SET OPTION** は、データベースサーバに送信される後続のすべての SQL 文に影響します。ただし、データベースサーバの再起動後に有効となるオプションの場合は除きます。

プライマリキーとロック

テーブルレベルのスナップショットのバージョン管理が有効な場合、あるテーブルを更新できるユーザは 1 人のみであるため、プライマリキーを生成しても、同時実行による矛盾は発生しません。

アクティブなテーブルロックの表示

テーブルをロックしているユーザを確認します。テーブルへ書き込みを行おうとしたときに、他のトランザクションがそのテーブルをロックしていると、書き込みは失敗します。

1. **sp_iqtransaction** を実行します。
2. 出力から、トランザクション識別子とユーザ名を見つけます。
3. **sp_iqlocks** プロシージャを実行します。

現在のデータベースのカatalogストアと IQ ストアの各ロックについて、**sp_iqlocks** で次のことがわかります。

- ロックを保持している接続とユーザ ID
- ロックが保持されているテーブル

- ロックの種類と、ロックを識別するための名前

たとえば、別のトランザクションがロックしている場合、次のメッセージが表示されます。

```
Cannot open the requested object for write in the current transaction (TxnID1). Another user has write access in transaction TxnID2.
```

`sp_iqtransaction` の出力から TxnID2 を探し、同じローでユーザ名を調べます。

テーブルレベルのスナップショットのバージョン管理がパフォーマンスに与える影響

通常、テーブルレベルのスナップショットのバージョン管理は、パフォーマンスにほとんど影響しません。

ただし、次に示すリソース上の問題に注意してください。

- 複数のユーザが同じデータベースページの異なるバージョンを同時に使用している場合、バッファの消費量が少し増加する可能性がある。
- バージョン管理には多少オーバーヘッドがかかるが、パフォーマンスにはほとんど影響しない。
- スレッド制御はユーザが取得する処理用リソース数を決定し、スイープ制御は少数のスレッドを使用してダーティデータページをディスクへ一掃する。どちらの制御も、パフォーマンスへの影響はあまりありません。
- ディスク領域が問題となる場合がある。重複するバージョンを格納すると、同時に使用されるバージョンの数やサイズによっては大量のディスク領域が使用される可能性があります。メタデータとデータベースページバージョンは、**RELEASE SAVEPOINT** の実行時か、または特定のバージョンを参照可能な最後のトランザクションのコミットまたはロールバック時に削除されるまで、保持されます。削除後の領域は再利用されます。

ロックによる遅延はほとんどありません。個々のコミット、ロールバック、チェックポイントが他の読み込みまたは書き取りトランザクションをブロックすることがありますが、非常に短い時間だけです。

これらすべてのパフォーマンスとディスク使用要因がシステムにどの程度影響するかは、ユーザが SAP Sybase IQ の読み込みと書き込みの同時実行機能を利用する程度に応じて異なります。特に、コミット前にかかった書き込みトランザクションの時間、書き込みトランザクション中に実行された読み込みトランザクションの数、これらのトランザクションが処理するロー数、中間セーブポイントでデータページの解放を行うかどうかによって、ディスク領域の必要量は大幅に異なります。

重複したバージョンと削除

テーブルレベルのバージョン管理を使用しているときにデータを削除するには、IQ メインストアに DB 領域を追加して、ディスク領域を増加しなければならない場合があります。

削除に必要な領域量は、データページ上でのデータの分散状態によって異なります。削除されるローのサイズや数は関係ありません。SAP Sybase IQ は、削除が開始してからトランザクションがコミットされるまでの間、削除するデータが入っている各ページのバージョンを保持する必要があります。削除中のローが多数のデータページに分散している場合、それらの余分のデータページを保持するための領域が IQ メインストア内に必要となります。

たとえば、各ページが 100 個のローを保持しているデータベースから 10 個のローを削除するとします。10 個のローのそれぞれが別々のデータページ上にある場合、IQ メインストアには 10 ページ分のバージョンページ用の領域が必要となります。各領域は、100 個のローを保持できる大きさでなくてはなりません。このように分散されることはあまりありませんが、可能性はあります。

データの削除に必要な領域はインデックスタイプによって異なります。データを削除するインデックスのサイズに比例し、最悪の場合、インデックスと同じサイズの領域が必要となります。

データの削除中に領域が不足すると、SAP Sybase IQ はその削除を中断し、通知ログに次のメッセージを表示します。

```
Out of disk space
```

領域を追加すると、削除が再開します。削除トランザクションがコミットすると、その領域は他の削除または挿入用に使用できるようになります。通常データベース内にそれほど領域を必要としない場合、DB 領域を削除して他の目的に使用する余分のディスク領域を確保できます。新しい DB 領域が必要となる可能性のあるデータを挿入する前に、ディスク領域を確保しておいてください。

削除中の領域不足によって、他のクエリユーザに影響をおよぼさないようにします。

領域が不足し、別の DB 領域を追加するだけの十分なディスク領域がない場合は、データベースエンジンを停止して再起動してください。データベースがロールバックされます。これで、ローの削除を、より小さなトランザクションに分けて実行できます。

注意： **DROP TABLE** と **DROP DATABASE** は、バージョンページを作成しないで、テーブルまたはデータベースとそこにある全データを削除するため、これらのコマンドを使用するために領域を追加する必要はありません。

ローレベルのスナップショットバージョン管理

ローレベルのスナップショットバージョン管理は、インメモリ RLV ストレージが有効なテーブルのみに適用されます。ローレベルのスナップショットバージョン管理では、複数のライタが、テーブルに対して DML 変更を同時に実行できますが、同じローに対しては実行できません。

ローレベルのスナップショットバージョン管理では、ローロックを使用してローレベルでテーブルをロックします。ローロックでは、テーブルローに対して書き込みロックを行います。つまり、**BLOCKING** および **BLOCKING_TIMEOUT** オプション設定に応じて、トランザクションがブロックされるか失敗します。**BLOCKING** が ON の場合は、トランザクションはブロックされます。**BLOCKING** が OFF の場合は、ALREADY LOCKED SQL 例外が発生しトランザクションは直ちに失敗します。

トランザクションのブロックにより、ローレベルのスナップショットバージョン管理は同じテーブルの異なるローに同時に書き込みを行うことができます。

BLOCKING および **BLOCKING_TIMEOUT** オプションの設定によっては、ローロックの競合でエラーが発生するか、指定タイムアウト時間内にロックが解放された場合はそのロックの獲得が試みられます。テーブルレベルのバージョン管理が設定されたトランザクションが、ローレベルのバージョン管理が設定されたトランザクションによってロックされたローを含むテーブルに書き込みを行おうとすると、テーブルレベルのトランザクションはエラーで失敗するか、指定タイムアウト時間内にロックが解放された場合はブロックされ再試行されます。

テーブル (**CREATE**、**DROP**、および **ALTER**) に対して DDL 変更を実行します。ただしテーブルレベルでテーブルをロックします。

スナップショットバージョン管理の指定

SNAPSHOT_VERSIONING オプションを使用して、スナップショットバージョン管理タイプを Row-level または Table-Level のいずれかに設定します。オプションは、データベース (PUBLIC) レベル、接続レベル (TEMPORARY)、またはユーザレベルで設定できます。インメモリ RLV ストアを使用するには、トランザクションでローレベルのスナップショットバージョン管理を有効にします。同じテーブルの異なるローを同時更新するには、各トランザクションまたは接続でローレベルのスナップショットバージョン管理を有効にする必要があります。

前提条件

- Row-level に設定されている場合は、RLV ストアの DB 領域には 1 つ以上の DB ファイルが存在する。
- Row-level に設定されている場合は、テーブルが RLV 対応である。

- PUBLIC または他のユーザ/ロールに対して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。

手順

トランザクションに対してスナップショットバージョン管理プロパティが設定されると、トランザクションがコミットするまで、そのプロパティは変わりません。

1. **SET OPTION** コマンドの範囲を決定し、データベース全体のオプション、接続レベルのオプション、またはユーザレベルのオプションとしてオプションを設定します。
 - **SET OPTION public.SNAPSHOT_VERSIONING...**
 - **SET TEMPORARY OPTION SNAPSHOT_VERSIONING...**
 - **SET OPTION *username*.SNAPSHOT_VERSIONING...**
2. スナップショットバージョン管理タイプを指定します。

レベル	オプション
Row-level	ローレベルのスナップショットバージョン管理。インメモリ RLV ストレージが必要。ローレベルのスナップショットバージョン管理では、複数のライタが、テーブルに対して DML 変更を同時に実行できるが、同じローに対しては実行できない。
Table-level	従来(下位互換)の SAP Sybase IQ のバージョン管理の動作。テーブルレベルでスナップショットを作成する。複数のライタがテーブルに対する DML 変更を同時に実行することはできない。

```
SET TEMPORARY OPTION Snapshot_Versioning = 'Row-level';

CREATE TABLE rv_locks(c1 int, c2 int, c3 int);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;
```

チュートリアル：書き込みを意図したロックのモニタ

このチュートリアルでは、RLV 対応テーブルを作成し、トランザクションを実行します。また、**sp_iqlocks** ストアドプロシージャを使用して、データベース内のスキーマレベルロックと書き込みを意図したロックに関するレポートを実行します。その後で、**sp_iqconnection** と **sa_conn_info** のストアドプロシージャを使用して、書き込みを意図したロックを制御する内部接続を表示します。

前提条件

- SAP Sybase IQ サーバにシンプレックスデータベースがある。
- RLV ストレージが設定されている。

手順

ヒント： ロックは、SAP Control Center を使用してモニタすることができます。詳細については、SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照を参照してください。

1. RLV 対応テーブル `rv_locks` および `rv_locks2` を作成して、テーブルレベルのスナップショットバージョン管理を設定します。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'Table-level';

CREATE TABLE rv_locks(c1 INT, c2 INT, c3 INT);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;

CREATE TABLE rv_locks2(c1 int, c2 int, c3 int);

INSERT INTO rv_locks2 VALUES (1,1,1);
INSERT INTO rv_locks2 VALUES (2,2,2);
INSERT INTO rv_locks2 VALUES (3,3,3);
INSERT INTO rv_locks2 VALUES (4,4,4);
COMMIT;

ALTER TABLE rv_locks ENABLE RLV STORE;
ALTER TABLE rv_locks2 ENABLE RLV STORE;
```

2. 接続ブロッキングを有効化して、ブロッキングタイムアウトのスレッシュホールドを設定します。

```
SET TEMPORARY OPTION BLOCKING = 'ON';
SET TEMPORARY OPTION BLOCKING_TIMEOUT = '0';
```

3. `sp_iqlocks` ストアドプロシージャを使用して、データベースロックの現在の設定を表示します。この時点で返されるロックはありません。

```
sp_iqlocks
```

RLV 対応テーブルに書き込みを意図したロックが存在しないため、テーブルのインメモリ RLV 部分をさらに作成する必要があります。

4. 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プロパティを `row-level` に設定します。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'row-level';
```

5. テーブルに書き込みます。

```
INSERT INTO rv_locks VALUES (5,5,5);
```

RLV 対応テーブルに対する書き込みまたはクエリを実行すると、要求に応じてメモリ内にテーブルの RLV 対応部分が作成されます。

6. `sp_iqlocks` を再実行します。

```
sp_iqlocks
```

今回は、プロシージャから書き込みを意図したロックが返されます。

```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifer,row_range
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks',,'Schema','Transaction','Shared',
,
'RLV_CONN_T775',
1000000407,,'BASE','DBA','rv_locks',,'Table','Transaction','Int
ent',,
```

接続 ID 100000407 には `rv_locks` テーブルに対する書き込みを意図したロックがあります。このロックタイプは `Intent` に設定されており、書き込みを意図したロックであることが示されます。

注意：接続 ID 番号 (100000407) は、サーバ自体内の内部接続を表すため大きいです。この内部接続は、RLV 対応テーブルのロックの管理に使用されます。

接続 ID 3 には、テーブルに対するスキーマロックがあります。このロックタイプは `Shared` に設定されており、共有スキーマロックであることが示されます。

7. 挿入を実行し、コミットされていないトランザクションに戻り、これをコミットします。

```
Commit
```

コミット時に、トランザクションが保持していたロックがデータベースによって解放されます。このチュートリアルでは、共有スキーマロックのみが解放されます。RLV 対応テーブルは、コミット済データとともにメモリ内に存在しています。したがってこの時点では、テーブルの RLV 対応部分が保持している書き込みを意図したロックのみが存在することになります。

8. `sp_iqlocks` を再実行します。

```
sp_iqlocks
```

スキーマロックはなくなりましたが、書き込みを意図したロックは残っています。

```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifer,row_range
'RVL_CONN_T775',
```


Userid "INT: RLVLockConn" は、内部接続を示します。この接続は RLV 対応テーブルで書き込みを意図したロックの制御に使用されます。

ローロック

ローロックはテーブルローの書き込みロックです。ロックされたローのカラムへの書き込みを、ロックを所有するトランザクションに許可します。このロックには 1 度に複数の所有者が存在することはできません。書き込みを意図したロックが前提条件になります。ロックマネージャがローロックを許可するには、トランザクションが書き込みを意図したロックを保持している必要があります。

テーブルローの書き込みロックは、ロックされたローのカラムへの書き込みを、ロックを保持するトランザクションに許可します。ロックを要求するトランザクションが書き込みを意図したロックを保持していなければ、このロックは許可されません。ローの書き込みロックは排他ロックです。1 つのローに対していつでも書き込みロックを保持できるのは、1 つのトランザクションのみです。トランザクションが書き込みロックを取得すると、他のトランザクションによるローのロック要求は拒否されます。

ローロックはローの削除中にのみ存在します。RLV ストアは追加専用のストアです。つまり書き込みアクションごとに新しいローがストアに追加されます。

UPDATE 文のように、**INSERT** 文によって新しいローがストアに追加されます。

RLV ストアでは、**UPDATE** を、**INSERT** が後に続く **DELETE** であると見なします。ローが削除される前に、**DELETE** 文または **UPDATE** 文において、データベースはローレベルのロックを取得します。

Interactive SQL での書き込みを意図したロックのモニタ

sa_locks システムプロシージャを使用して、書き込みを意図したロックの詳細を表示できます。

前提条件

- RLV ストレージが設定されている。
- テーブルが RLV ストレージに登録されている。
- データベースオプション **SNAPSHOT_VERSIONING** が 'ローレベル' に設定されている。

手順

1. **sa_locks** システムプロシージャを実行します。
2. テーブルおよびローに対する意図的のロックを表示する **lock_class** カラムを表示します。
3. **sp_iqlocks** ストアドプロシージャを使用して、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特定します。このプロシージャは、ロ

クを保持している接続とユーザ ID、ロックが保持されているテーブル、ロックのタイプ、ロックを識別する名前を含めて、データベースで現在保持されているロックについての情報を表示します。

書き込みを意図したロック

書き込みを意図したロックは、将来のテーブルローへの書き込み権限をトランザクションに付与するテーブルの書き込みロックです。書き込みを意図したロックは、複数の要求接続が保持できます。

テーブルの RLV 対応部分がメモリに存在していると、常に書き込みを意図したロックが存在します。**sp_iqlocks** ストアドプロシージャを使用して、書き込みを意図したロックの詳細を表示できます。

書き込みを意図したロックは、テーブルの書き込みロックおよびテーブルの排他ロックと競合します。この競合により、ロックマネージャがテーブルに対する書き込みを意図したすべてのロックを解放するまで、テーブルレベルのスナップショットバージョン管理トランザクションはテーブルへの書き込みや DDL 操作を実行できなくなります。テーブルレベルのスナップショットバージョン管理トランザクションとローレベルのスナップショットバージョン管理トランザクションの接続がテーブルに書き込みを行う場合は、書き込みを意図したロックにより同期が行われます。次のシナリオを考えてみます。

接続	アクション
ローレベルのスナップショットバージョン管理トランザクション A	<ul style="list-style-type: none"> • table_1 の複数のローへの書き込みを行うクエリを実行する。 • ロックマネージャは、table_1 への書き込みを意図したロックを作成する。 • ロックマネージャは、ローレベルの DML 更新に対する書き込みを意図した複数のローカルなロックを作成する。ロックマネージャはローレベルのロックを作成する。
テーブルレベルのスナップショットバージョン管理トランザクション B	table_1 への書き込みを試行する。トランザクション B は、書き込みを意図したロックによってブロックされる。
ローレベルのスナップショットバージョン管理トランザクション A	トランザクション A をコミットする。RLV ストアのテーブルの変更は IQ メインストアにマージされる。書き込みを意図したロックは解放される。
テーブルレベルのスナップショットバージョン管理トランザクション B	table_1 への書き込みを続ける。

ローレベルの DDL ロックに関する考慮事項

RLV 対応テーブルに対するデータ定義言語 (DDL) (たとえば、**CREATE INDEX**、**DROP INDEX**、**ALTER TABLE ADD**、**ALTER**、**DROP** など) の変更には、テーブルレベルの排他ロックが必要です。DDL イベントの場合、RLV 対応テーブルに対するロックの動作は IQ メインストアテーブルの場合と同じで、書き込み接続がテーブルに対する排他ロックを保持します。**BLOCKING** が ON に設定されている場合、テーブルに対して競合する DML および DDL トランザクションはすべて、DDL の変更がコミットされるまでブロックされます。**BLOCKING** が OFF に設定されている場合、競合トランザクションのロック要求は即座に失敗します。

トランザクションのバージョン管理タイプの表示

アクティブな各トランザクションのバージョン管理タイプ (テーブルレベルかローレベル) を表示するには、**sp_iqtransaction** ストアドプロシージャを使用します。

1. **sp_iqtransaction** を実行します。
2. VersioningType カラムを調べます。

VersioningType	説明
Row-level	トランザクションはローレベルバージョン管理を使用する。ローレベルのバージョン管理は、RLV 対応テーブルの同時ライタアクセスおよびローレベルのロックを有効にする。
Table-level	トランザクションはテーブルレベルバージョン管理を使用する。テーブルレベルのバージョン管理は、単一ライタアクセスおよびテーブルレベルのロックを有効にする。

参照：

- **sp_iqtransaction** プロシージャ (222 ページ)

テンポラリテーブルのバージョン管理

データベースで作成したテンポラリテーブルは、「グローバルテンポラリテーブル」と呼ばれます。データベースで作成されたのではなく、宣言されたテンポラリテーブルは、「ローカルテンポラリテーブル」と呼ばれます。

グローバルテンポラリテーブルには、適切なパーミッションを持つすべてのユーザがアクセスできます。各ユーザはそのテーブルの自分用のインスタンスを持ちます。ただし、特定のローセットを参照できるのは、1 人のユーザのみです。デフォルトで、グローバルテンポラリテーブルのローは、**COMMIT** の実行時に削除

されます。テンポラリテーブルの作成時に **ON COMMIT PRESERVE ROWS** を指定すると、このデフォルトの動作を変更できます。

1人のユーザのみが、ローカルテンポラリテーブル内の任意のローを参照できます。このテーブルはユーザが接続を解除すると削除されます。ローカルテンポラリテーブルを宣言すると、SAP Sybase IQ は、トランザクションをオートコミットしないでセーブポイントを発行します。他のタイプのテーブルに対するデータ定義操作の場合と同様です。ローカルテンポラリテーブルのデータをコミットしてからインデックスを作成してください。コミットされていないデータを使用してインデックスを作成しようとした場合、「インデックスを作成するには、ローカルテンポラリテーブル <テーブル名> をコミットしなければなりません。」と表示される場合があります。

SAP Sybase IQ は、ベーステーブル (IQ メインストアのデータベーステーブル) のバージョン管理とグローバルテンポラリテーブルのバージョン管理を区別しません。これは、テンポラリテーブル内のデータには1人のユーザしかアクセスできないため、複数の書き込みトランザクションが1つのテンポラリテーブルに対して開くことはないからです。

グローバルテンポラリテーブルまたはローカルテンポラリテーブルで RLV ストレージを有効化することは可能ですが、テンポラリテーブルで RLV の複数ユーザの同時書き込み機能を利用することはできません。テンポラリテーブルで RLV ストレージを有効化すると、RLV ストアの遅延時間の短い DML を利用できます。

ロック競合がパフォーマンスに与える影響の調査

ロードまたはクエリのパフォーマンスの問題の一部は、ロック競合に起因します。カーネルのシステム時間が 10% を超える場合は、ロック競合が発生している可能性があります。

1. ロック競合がシステムのパフォーマンスに影響を与えているかどうかを確認するには、次のいずれかを実行します。
 - **-contention** オプションを指定して SAP Sybase IQ モニタを実行します。
 - UNIX プラットフォームでは、**sar** または **vmstat** ユーティリティを実行します。
 - Windows プラットフォームでは、タスクマネージャで CPU 使用率を確認します。
2. ロック競合が疑われる場合は、以下を設定して、分割レベルを直接制御すると役立つ場合があります。
 - **-iqpartition** サーバ起動オプション
 - **cache_partitions** データベースオプション

注意： カーネルのシステム時間が通常より高い場合には、カーネルが適切にチューニングされていない可能性もあります。その場合は、カーネルパラメータの調整が必要です。SAP Sybase IQ の設定を変更しても、カーネルのチューニングの不備は改善できません。

参照：

- CACHE_PARTITIONS オプション (14 ページ)

(廃止予定)-contention

多くの重要なバッファキャッシュとメモリマネージャロックを表示します。これらのロックカウンタとミューテックスカウンタは、バッファキャッシュおよびヒープメモリ内のアクティビティと、これらのロックがどれだけ迅速に解消されたかを示します。タイムアウト数が 20% を超えている場合は、問題の発生を示しています。

使用法

```
monitor_options -contention
```

出力

表 20 : -contention の出力フィールド

出力フィールド	説明
<i>AU</i>	現在アクティブなユーザ数。
<i>LRULks</i>	LRU がロックされた (テンポラリキャッシュ用に繰り返された) 回数。
<i>woTO</i>	タイムアウトなしにロックが付与された (テンポラリキャッシュ用に繰り返された) 回数。
<i>Loops</i>	ロックが付与される前に SAP Sybase IQ がリトライした (テンポラリキャッシュ用に繰り返された) 回数。
<i>TOs</i>	SAP Sybase IQ がタイムアウトして、ロックのために待機する必要があった (テンポラリキャッシュ用に繰り返された) 回数。
<i>BWaits</i>	キャッシュ内のバッファに対する Busy Waits の (テンポラリキャッシュ用に繰り返された) 回数。
<i>IOLock</i>	SAP Sybase IQ が圧縮化 I/O プールをロックした (テンポラリキャッシュ用に繰り返された) 回数。無視してかまわない。

出力フィールド	説明
<i>IOWait</i>	圧縮化 I/O プール上のロックのために SAP Sybase IQ が待機する必要があった (テンポラリキャッシュ用に繰り返された) 回数。無視してかまわない。
<i>HTLock</i>	SAP Sybase IQ がブロックマップハッシュテーブルをロックした (テンポラリキャッシュ用に繰り返された) 回数。
<i>HTWait</i>	ブロックマップハッシュテーブルのために SAP Sybase IQ が待機する必要があった (テンポラリキャッシュ用に繰り返された) 回数。HTLock と HTWait は、使用中のブロックマップ数を示す。
<i>FLLock</i>	SAP Sybase IQ がフリーリストをロックする必要があった (テンポラリキャッシュ用に繰り返された) 回数。
<i>FLWait</i>	フリーリスト上のロックのために SAP Sybase IQ が待機する必要があった (テンポラリキャッシュ用に繰り返された) 回数。
<i>MemLks</i>	SAP Sybase IQ がメモリマネージャ (ヒープ) をロックした回数。
<i>MemWts</i>	メモリマネージャロックのために SAP Sybase IQ が待機する必要があった回数。

注意： SAP Sybase IQ ではスピンロックを使用しなくなりました。このため、[woTO]、[Loops]、[TOs] の統計はめったに使用されません。

チェックポイント、セーブポイント、トランザクションのロールバック

同時実行性の実現に加えて、スナップショットバージョン管理のトランザクション処理はデータリカバリにおいて重要な役割を果たします。

SAP Sybase IQ では、システム障害やメディア障害が発生すると、安定したデータセットをリカバリするトランザクション関連の 3 つのコマンドを使用します。

CHECKPOINT コマンドはチェックポイントの設定、**SAVEPOINT** コマンドはセーブポイントの設定、**RELEASE SAVEPOINT** はセーブポイントの解放、**ROLLBACK TRANSACTION** はトランザクションのロールバックを行います。

チェックポイント

チェックポイントは、SAP Sybase IQ が内部的に追跡した情報をディスクに書き込むときに、トランザクション内の有効なポイントにマーク付けします。SAP Sybase IQ はデータベースのリカバリでこの情報を使用します。

SAP Sybase IQ のチェックポイントの使用方法は、SQL Anywhere などの OLTP データベースとは異なります。多くの場合、OLTP データベースでは、少数のローにしかな影響しない短いトランザクションが実行されます。このため、ページ全体をディスクに書き込むのはコストがかかります。このため、一般に OLTP データベースは、チェックポイントでディスクに書き込みを行い、変更されたデータローだけを書き込みます。

SAP Sybase IQ は OLAP データベースです。1つの OLAP トランザクションで数千、数百万のデータローを変更することがあります。このため、データベースサーバはチェックポイントの発生を待たずに物理的な書き込みを実行します。トランザクションがコミットするたびに更新されたデータページがディスクに書き込まれます。OLAP データベースの場合、任意のチェックポイントで少量のデータを書き込むより、データページ全体を書き込むことで効率が大幅に向上します。

チェックポイントのほとんどは自動的に発生します。明示的なチェックポイントも設定できますが、必要ではありません。

チェックポイントは次のような時点で発生します。

- トランザクションが **CHECKPOINT** コマンドを発行した時点
- **CHECKPOINT_TIME** の値を超えた時点
- バックアップ処理の開始/終了時点
- データベースサーバが停止した時点

CHECKPOINT_TIME はチェックポイントの間隔の最大時間です。デフォルトでは 60 分です。チェックポイントの間隔を調整するには、**SET OPTION** 文を使用します。チェックポイント時間の調整や明示的なチェックポイントの発行は、ほとんど必要ありません。SAP Sybase IQ は各トランザクションのコミット後に実際のデータページを書き込むため、SAP Sybase IQ でのチェックポイントの制御は OLTP データベース製品の場合ほど重要ではありません。

トランザクション内のセーブポイント

SAP Sybase IQ は、トランザクション内のセーブポイントをサポートします。

SAVEPOINT 文は、トランザクションの中に中間ポイントを定義します。単一の IQ トランザクションで数百万のデータローを書き込む場合があるため、コミットして IQ メインストアに書き込むデータの量を、トランザクションを完全に実行した場合より少ない量に制限できます。セーブポイントを設定すると、トランザクションをさらに分割できます。

セーブポイントの解放

セーブポイントを解放すると、そのセーブポイントまで使用されていたバージョンページが解放されます。

内部的にはデータはページレベルでバージョン付けされています。SAP Sybase IQ は更新したばかりのページを別にコピーして保持し、それ以外のページは以前のバージョンと共有します。

セーブポイントを解放すると、ディスク領域を効率よく使用できます。

この Interactive SQL コマンドはセーブポイント *n* を解放します。

```
RELEASE SAVEPOINT n
```

そのセーブポイント以降のすべてのリソースが解放され、中間セーブポイントへのロールバックは不可能となります。

RELEASE SAVEPOINT では、ロックは解放されません。

セーブポイントへのロールバック

ROLLBACK TO SAVEPOINT コマンドは、セーブポイント以降のすべての変更を取り消します。

このコマンドは、指定したセーブポイント、または指定しなかった場合は最新のセーブポイントまでロールバックします。セーブポイント *n* へロールバックすると、*n* 以上のすべてのセーブポイントに対する全アクションが取り消されます。

通常、ロックはトランザクションの終了時にのみ解放されます。ただし、次の条件を満たす場合、**ROLLBACK TO SAVEPOINT** でロックが解放されます。

あるトランザクション内に一連のセーブポイントがあるとします。まず、書き込み操作を実行します。そのトランザクションを以前のセーブポイントまでロールバックします。ロールバックで、そのセーブポイント以降のアクションがすべて取り消されます。そのアクションにはセーブポイント以降に行われた書き込み操作と、取得されたロックが含まれます。

SAP Sybase IQ は、更新可能カーソルに対するセーブポイント操作をサポートしません。

自動セーブポイントとユーザ定義セーブポイント

IQ は、各 DML コマンドの前後に暗黙的なセーブポイントを設定します。

これらのセーブポイントに関連付けられたデータページバージョンは、コマンドの完了時に解放されます。ある DML コマンドの終了後もデータページバージョンを保持するには、名前を付けた自分用のセーブポイントを設定します。

セーブポイントの名前付けとネスト

セーブポイントに名前を付けてネストすることにより、1つのトランザクション内に複数のアクティブなセーブポイントを設定できます。

SAVEPOINT と **RELEASE SAVEPOINT** の間の更新も、その前のセーブポイントにロールバックするか、トランザクション全体をロールバックすれば取り消すことができます。トランザクション内の変更は、トランザクションがコミットされるまで確定していません。トランザクションが終了すると、セーブポイントはすべて解放されます。

セーブポイントにより、SAP Sybase IQ は使用可能なディスク領域のロケーションに関して保持している情報を更新します。この情報はトランザクションのロールバック時に使用されます。

未解放のセーブポイントが古い中間バージョンをアクティブにして余分なディスク領域を消費する場合がありますが、セーブポイントを使用してもオーバヘッドが増えることはありません。

トランザクションのロールバック

トランザクションをロールバックすると、自動的に、またはユーザの明示的な要求により、そのトランザクション内の全操作が取り消されます。

データベースをロールバックするとは、データベースを以前の状態に戻すことを意味します。

ロールバックの原因

ロールバックは、明示的に要求することもできますし、自動的にも発生します。

ROLLBACK 文を使用すると、最後の **COMMIT** または **ROLLBACK** 以降のデータベースへの変更が取り消されます。

ROLLBACK TO SAVEPOINT 文を使用すると、指定した **SAVEPOINT** 以降、または最後の **SAVEPOINT** 以降のデータベースへの変更が取り消されます。

トランザクション内のユーザがコミットしないでログアウトまたは接続を解除した場合、SAP Sybase IQ はデータベースを自動的にロールバックします。ロールバックは、最新のコミットまたはロールバックのポイントまで行われます。

ロールバックの影響

ロールバックを行うと、メインストアとテンポラリストアの両方が以前の状態に戻され、ロックも解放されます。

- トランザクションロールバックでは、トランザクションが保持するすべてのロックが解放される。

トランザクションとバージョンの管理

- セーブポイントへのロールバックでは、そのセーブポイント以降に取得されたすべてのロックが解放される。

オープンカーソルのロールバックでは、すべてのカーソル情報が削除され、ホールドカーソルと非ホールドカーソルの両方が閉じられます。

- カーソルが、ロールバック中のトランザクション内で、または以前のトランザクション内で開いていたかどうかに関係なく、トランザクションのロールバックではすべてのカーソルが閉じられる。
- セーブポイントへのロールバックでは、そのセーブポイント以降に開かれたすべてのカーソルが閉じられる。

システムリカバリ

システム障害や停電の発生時、または停止していたデータベースサーバを再起動する場合、SAP Sybase IQ は自動的にリカバリしようとしてします。

SAP Sybase IQ データベースのリカバリ時に、未コミットのトランザクションがロールバックされ、古いバージョンが使用していたディスク領域は使用可能な領域のプールに戻されます。この時点で、データベースには、各永久テーブルの最新のコミットされたバージョンしかありません。

システム障害からのリカバリ時に、SAP Sybase IQ は障害発生時にアクティブだったすべての接続を再オープンします。ユーザの接続数を設定する `-gm` パラメータが障害発生時に有効であった場合は、少なくとも障害の発生時に使用していた接続数で IQ サーバを再起動します。テンポラリテーブルの内容はリカバリできません。

障害が発生したら、データベースサーバとデータベースを再起動してみます。リカバリするには、サーバログと IQ メッセージログの情報が必要です。

システム障害の発生後、できればユーザに接続を許可する前に、ストアドプロシージャ `sp_iqcheckdb` を実行します。このプロシージャはデータベース内の各ブロックを検査して統計情報を生成します。この統計情報で、データベースの一貫性や整合性が検査できます。

リカバリにおけるトランザクション情報の役割

SAP Sybase IQ のリカバリのメカニズムは、通常、トランザクションの数が少なく、またトランザクションの処理にかかるデータウェアハウス環境向けに設計されたものです。

SAP Sybase IQ は、データベース更新を実際のデータベースページのコピー上で行い、書き込みトランザクションがコミットするたびにそのデータをディスクに書き込みます。SAP Sybase IQ は以下も記録します。

- 各トランザクションで変更されたデータのロケーションと量は、トランザクションログに格納されます。
- バージョンページと空き領域のロケーションは、ディスクに格納されます。SAP Sybase IQ は、この情報を使用して、バージョンが不要になったときに領域を解放します。書き込みトランザクション中に作成されたすべてのバージョンは、その書き込み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不要になります。セーブポイントで個々のバージョンを解放することもできます。
- トランザクション中に発生したチェックポイントに関する追加情報。

データベースのリカバリが必要になった場合、SAP Sybase IQ はトランザクションログの情報とチェックポイント情報を使用して素早くリストアします。バージョンと空き領域に関する情報を使用して、トランザクションのロールバックと、古いバージョンが占領していたディスク領域の解放を行います。

トランザクションログは領域をほとんど必要としません。コミットされた各トランザクションあたり 128 バイト程度です。チェックポイントと使用可能なディスク領域に関する情報に必要な領域も非常に少量です。ただし、データを変更するトランザクションが多いシステムではログが非常に大きくなる可能性があるため、ログを定期的にトランケートする必要があります。

チェックポイント情報は次のチェックポイントで削除されます。特定のセーブポイントに関連する情報は、そのセーブポイントの解放またはロールバック時に削除されます。

バックアップの同時実行性

読み込み操作や書き込み操作と同時にバックアップを実行することもできます。

バックアップは DML 操作です。バックアップでは、バックアップコマンドの開始時点(チェックポイント)のデータがバックアップされます。ただし、リストア操作は、データベースへの書き込みを行うため排他的アクセスが必要です。

トランザクション内のカーソル

SELECT の結果を、カーソルというデータ型の形式で返します。

カーソルはテーブルと似ていますが、1つのローは現在のローとして識別されません。各種のコマンドを使用して、カーソルをロー間でナビゲーションできます。たとえば、**FETCH** コマンドはカーソルからローを取り出し、それを現在のローとして識別します。このコマンドを繰り返し呼び出すと、カーソル内のすべてのローを閲覧できます。

カーソルは、プロシージャを記述する場合や、データベースへのアクセス用アプリケーションを Embedded SQL で記述する場合に便利です。また、カーソルは多

くのフロントエンドクエリツールでも使用されています。Interactive SQL を使用する場合は利用できません。

SAP Sybase IQ のカーソルは更新可能です。データベースでのカーソルの処理中に、基本となるデータを修正できます。

カーソル内のローには、テーブル内のローと同様、順番がありません。**FETCH** コマンドはローを順に処理していきますが、その順序はランダムで一貫性がない可能性があります。**ORDER BY** フレーズを **SELECT** 文に付加することによって、順番を設定します。

sp_iqcursorinfo ストアドプロシージャは、サーバ上で現在開いているカーソルに関する情報を表示します。

カーソルとバージョン管理

カーソルを使用する場合、SAP Sybase IQ は、単一のトランザクション内で複数のバージョンを管理する必要があります。

たとえば、時刻 x に `customer` テーブルを使用する `cust_cursor` というカーソルを開くとします。そのテーブルを時刻 y に更新します。SAP Sybase IQ は、時刻 x から `cust_cursor` の使用を終了するまでの間、`customer` テーブルのバージョンを保持します。

SAP Sybase IQ では、DSS アプリケーションでの使用に向くようにカーソルがサポートされています。

カーソルの依存性

カーソルが開いた時点から閉じるまでの間に、カーソルのメンバシップ (カーソルが返すデータロー) が変わることが可能な場合、そのカーソルは依存型と呼ばれます。非依存カーソルでは、開いた時にそのメンバシップが固定します。

`insensitive` カーソルでは、その結果セットのメンバシップと値は、変更に関しては不定です。値依存カーソルでは、そのメンバシップに関しては非依存的、結果セットの順序と値については依存的です。SAP Sybase IQ では、更新可能な `insensitive` カーソルをサポートしています。

カーソルのスクロール

SAP Sybase IQ カーソルは、スクロールとすることも、非スクロールとすることもできます。

非スクロールカーソルの場合、**FETCH NEXT** と **FETCH RELATIVE 0** のコマンド形式でしか、データの検索と取り出しができません。どのローがフェッチされたかは、追跡しません。**DYNAMIC SCROLL** として宣言されたカーソルは、**SCROLL** として宣言されたカーソルと同じです。

数百万もの非常に多くのローを取り出す場合にテンポラリ記憶領域の必要量を節約するには、**FORCE_NO_SCROLL_CURSORS** オプションを ON に設定します。ただし、フロントエンドアプリケーションで後方スクロールカーソルオペレーションが頻繁に使用される場合は、このオプションを OFF に設定した方が、クエリの応答が速くなります。

フロントエンドアプリケーションで後方スクロールがほとんど行われなない場合は、**FORCE_NO_SCROLL_CURSORS= 'ON'** を永続的な **PUBLIC** オプションに設定します。メモリの節約になるため、クエリのパフォーマンスが向上します。

ホールドカーソル

HOLD オプションを指定してカーソルを開くと、**COMMIT** でトランザクションが終了した場合に、終了後もそのカーソルが開いた状態のままにできます。

ホールドカーソルは、カーソルを開いたトランザクションが **ROLLBACK** された場合は、開いたままとはなりません。

SAP Sybase IQ では、ホールドカーソルはコミットされるまでは更新できます。コミット後のホールドカーソルは、読み取り専用として内部的にマーク付けされ、それ以降に位置付け更新しようとするエラーが生成されます。

HOLD オプションは通常、DSS 環境では使用されませんが、このオプションが役に立つ場合もあります。たとえば、多くの既存のアプリケーションではホールドカーソルの使用を推奨しており、いくつかの ODBC ドライバはデフォルトでホールドカーソルを使用します。

SAP Sybase IQ は、ホールドカーソル用に必要なバージョン管理を提供していません。

ホールドカーソルはパフォーマンスに影響します。メモリ、ディスク領域、処理スレッドなど、カーソルが使用するすべてのリソースは、カーソルを閉じるまで保持されます。

位置付け操作

位置付け操作では、カーソルの現在のロケーションによって、読み取りまたは書き込み操作の開始場所が決定します。

SAP Sybase IQ は位置付けフェッチをサポートしています。これは、長いクエリトランザクションで役立ちます。SAP Sybase IQ は位置付け更新と位置付け削除の操作もサポートしています。これらは、短い挿入や削除を目的としたものです。ほとんどの場合、SAP Sybase IQ データベースへの更新には大量のデータが伴います。再位置付けは、そのような書き込み操作では非常に小さな部分です。

位置付け更新と削除は、個別のステートメントとしてではなく、カーソルに対する操作として(したがってトランザクションの一部として)処理されます。カーソ

ルを開いた後でエラーが発生すると、このカーソルをとおして実行されたすべての処理がロールバックされます。

カーソルのメッセージロギング

デフォルトでは、カーソル操作のログは SAP Sybase IQ メッセージファイルに書き込まれません。

問題の原因を判断するためにカーソル操作を追跡するには、`LOG_CURSOR_OPERATIONS` オプションをオンにして、カーソルが開いたり閉じたりするたびに、メッセージが生成されるようにします。更新可能なカーソルを通じて行われたデータ変更も、IQ メッセージファイルにログが記録されます。

リモートトランザクション

リモートサーバが関与するトランザクションの管理には、2 フェーズコミットプロトコルを使用します。

SAP Sybase IQ は、ほとんどの場合においてトランザクションの整合性を保証します。

リモートトランザクションの制限

リモートトランザクション管理では、セーブポイントおよびネストされた文に制限があります。

トランザクション管理には、次のような制限があります。

- セーブポイントは、リモートサーバには伝達されません。
- リモートサーバに関わるトランザクションで、**BEGIN TRANSACTION** 文と **COMMIT TRANSACTION** 文の組がネストされている場合は、最も外側の組のみが処理されます。矛盾する **BEGIN TRANSACTION** 文と **COMMIT TRANSACTION** 文の組は、リモートサーバには送信されません。

プロシージャとバッチの作成

プロシージャとバッチにより SAP Sybase IQ データベースのセキュリティ、効率、標準化を高めることができます。

プロシージャは、すべてのアプリケーションで使えるように、SQL 文をデータベースに格納します。これによりデータベースのセキュリティ、効率、標準化を高めることができます。ユーザ定義関数は、クエリやその他の SQL 文で使うための結果を返すプロシージャの一種です。

さまざまな用途で、サーバ側の JDBC は、SQL ストアドプロシージャよりも柔軟にデータベースにロジックを構築します。

バッチは、データベースサーバにグループとして送られる SQL 文のセットです。制御文などのプロシージャで使用できる機能の多くは、バッチ内でも使用できます。

注意： ソースコードに対する変更と、ソース (ストアドプロシージャを含む) から作成されデータベースに展開されるオブジェクトに対する変更を追跡する場合は、ソース制御を使用します。

プロシージャ

プロシージャの作成

SAP Sybase IQ プロシージャを作成するには、SQL 文 **CREATE PROCEDURE** を使用します。

前提条件

CREATE PROCEDURE または CREATE ANY PROCEDURE システム権限が必要です。

手順

CREATE PROCEDURE 文を入力します。

たとえば、SAP Sybase IQ デモデータベース `iqdemo.db` を使用します。

```
CREATE PROCEDURE new_dept (IN id INT,
                           IN name CHAR(35),
                           IN head_id INT)
BEGIN
    INSERT
        INTO GROUP0.departments (DepartmentID,
```

プロシージャとバッチの作成

```
DepartmentName,  
DepartmentHeadID)  
values (id, name, head_id);  
END
```

注意： `CREATE PROCEDURE` の `AT location-string` SQL 構文を使用して、プロキストアドプロシージャを作成します。

プロシージャの変更

SAP Control Center または Interactive SQL を使って既存のプロシージャを変更できます。プロシージャの所有者であるか、`ALTER ANY PROCEDURE` システム権限を保持していることが必要です。

`ALTER PROCEDURE` 文を使用して、プロシージャを変更することもできます。

プロシージャの呼び出し

プロシージャの呼び出しには `CALL` 文を使用します。プロシージャは、アプリケーションプログラムから呼び出すことも、他のプロシージャから呼び出すこともできます。

プロシージャの削除

いったん作成したプロシージャは、誰かが明示的に削除するまではデータベースに残っています。プロシージャの所有者か `DROP ANY PROCEDURE` システム権限を持つユーザだけがプロシージャをデータベースから削除できます。

データベースからプロシージャを削除する `DROP PROCEDURE` 文を実行します。

ストアドプロシージャを実行する権限

プロシージャの所有者はそれを作成したユーザです。所有者は権限なしでそのプロシージャを実行できます。

注意： この情報はシステムプロシージャには当てはまりません。システムプロシージャを実行する権限の付与の詳細については、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 > システムプロシージャを参照してください。

プロシージャを実行するパーミッションを他のユーザに付与するには、`GRANT EXECUTE` コマンドを使用します。たとえば、プロシージャ `new_dept` の所有者は、次の文を使用してプロシージャの実行パーミッションを `another_user` に付与できます。

```
GRANT EXECUTE ON new_dept TO another_user
```

プロシージャを実行する権限を取り消すには、次の文を実行します。

```
REVOKE EXECUTE ON new_dept FROM another_user
```

プロシージャの結果をパラメータとして返す

プロシージャは、複数の方法で呼び出し元の環境に結果を返します。

- OUT または INOUT パラメータとして個々の値を返す。
- 結果セットとして。
- **RETURN** 文を使用して結果を 1 つ返す。

注意：たとえば、SAP Sybase IQ デモデータベース iqdemo.db を使用します。

```
CREATE PROCEDURE SalaryList (IN department_id INT)
RESULT ( "Employee ID" INT, "Salary" NUMERIC(20,3) )
BEGIN
    SELECT EmployeeID, Salary
    FROM Employees
    WHERE Employees.DepartmentID = department_id;
END
```

プロシージャの結果を結果セットとして返す

プロシージャは、個別のパラメータとして呼び出しを行った環境に結果を返すだけでなく、結果セットとして情報を返すこともできます。通常、結果セットになるのはクエリの結果です。

ストアードプロシージャ内でテンポラリテーブルを動的に作成した後、そのテーブルに対する **SELECT** 文を実行する場合は、**EXECUTE IMMEDIATE WITH RESULT SET ON** 構文を使用して、Column not found というエラーが起きないようにします。

次に例を示します。

```
CREATE PROCEDURE p1 (IN @t varchar(30))
BEGIN
    EXECUTE IMMEDIATE
    'SELECT * INTO #resultSet FROM ' || @t;
    EXECUTE IMMEDIATE WITH RESULT SET ON
    'SELECT * FROM #resultSet';
END
```

参照：

- プロシージャの結果 (292 ページ)

プロシージャ情報の表示

データベース内のシステムおよびユーザ定義プロシージャに関する情報を表示するには、**sp_iqprocedure** を使用します。

Interactive SQL で、**sp_iqprocedure** を実行します。

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。

プロシージャとバッチの作成

```
sp_iqprocedure sp_test

proc_name      proc_owner    proc_defn      replicate      srvid          r
emarks
sp_test        DBA            create procedure N            (NULL)        (NULL)
               DBA.sp_test(in n1
               integer)
               begin message `sp_test`end
```

プロシージャパラメータ情報の表示

ストアードプロシージャパラメータに関する情報を表示するには、**sp_procparm** を使用します。

Interactive SQL で、**sp_iqprocparm** を実行します。

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)
```

プロシージャでのカーソル

カーソルは、結果セットに複数のローがあるクエリまたはストアードプロシージャからローを1つずつ取り出します。

「カーソル」は、クエリまたはプロシージャに対するハンドルまたは識別子で、結果セットの中の現在の位置を示します。

カーソル管理

カーソル管理はファイル管理に似ています。カーソル管理については、次の手順に従います。

1. **DECLARE** 文を使って、**SELECT** 文またはプロシージャにカーソルを宣言します。
2. **OPEN** 文を使って、カーソルを開きます。

3. FETCH 文を使って、カーソルから結果をローごとに取り出します。
4. ローが見つかりませんという警告が、結果セットの最後に表示されます。
5. CLOSE 文を使ってカーソルを閉じます。

デフォルトでは、カーソルはトランザクションの最後に、COMMIT または ROLLBACK によって自動的に閉じられます。WITH HOLD 句を使って開いたカーソルは、明示的に閉じるまで、トランザクション内で開いた状態になります。

カーソル位置

カーソルの位置は柔軟に指定できます。カーソルを開くと最初のローの前に置かれます。カーソル位置は、クエリ結果の最初か最後を基準とした絶対位置、または現在のカーソル位置を基準とした相対位置に移動できます。カーソル位置の変更方法とカーソルで可能な操作は、プログラミングインタフェースが制御します。

カーソルでフェッチできるローの位置番号は、integer 型のサイズによって管理されます。integer に格納できる値より 1 小さい 2147483646 までの番号が付けられたローをフェッチできます。ローの位置番号に、クエリ結果の最後を基準として負の数を使用している場合、integer に格納できる負の最大値より 1 大きい数までの番号のローをフェッチできます。

現在のカーソル位置でローを更新または削除するには、位置付け更新と位置付け削除という特別な操作を使用できます。先頭のローの前か、末尾のローの後にカーソルがある場合、対応するカーソルローがないことを示すエラーが返されません。

注意： asensitive カーソルに挿入や更新をいくつか行くと、カーソル位置に問題が生じます。SELECT 文に ORDER BY 句を指定しないかぎり、SAP Sybase IQ はカーソル内の予測可能な位置にはローを挿入しません。場合によっては、カーソルを閉じてもう一度開かないと、挿入したローがまったく表示されないことがあります。SAP Sybase IQ では、カーソルを開くためにワークテーブルを作成する必要がある場合にこうしたことが起こります。

UPDATE 文によって、カーソル中のローが移動することがあります。これは、既存のインデックスを使用する ORDER BY 句がカーソルに指定されている場合に発生します (ワークテーブルは作成されません)。STATIC SCROLL カーソルを使うとこの問題はなくなりますが、よりメモリを消費します。

注意： SAP Sybase IQ は、FIRST、LAST、ABSOLUTE の各オプションを結果セットの先頭から開始するものとして扱います。負数のローカウントの RELATIVE は、現在の位置から開始するものとして扱います。

SELECT 文のカーソル

次に、SELECT 文でカーソルを使用するプロシージャの例を示します。この例では、ListCustomerValue プロシージャに使用するのと同じクエリをベースとして、ストアプロシージャ言語の特徴を示します。

```
CREATE PROCEDURE TopCustomerValue(
    OUT TopCompany CHAR(36),
    OUT TopValue INT )
BEGIN
-- 1. Declare the "row not found" exception
DECLARE err_notfound
    EXCEPTION FOR SQLSTATE '02000';
-- 2. Declare variables to hold
--     each company name and its value
DECLARE ThisName CHAR(36);
DECLARE ThisValue INT;
-- 3. Declare the cursor ThisCompany
--     for the query
DECLARE ThisCompany CURSOR FOR
SELECT CompanyName,
    CAST( sum( SalesOrderItems.Quantity *
        Products.UnitPrice ) AS INTEGER )
    AS value
FROM Customers
    INNER JOIN SalesOrders
    INNER JOIN SalesOrderItems
    INNER JOIN Products
GROUP BY CompanyName;
-- 4. Initialize the values of TopValue
SET TopValue = 0;
-- 5. Open the cursor
OPEN ThisCompany;
-- 6. Loop over the rows of the query
CompanyLoop:
LOOP
    FETCH NEXT ThisCompany
        INTO ThisName, ThisValue;
    IF SQLSTATE = err_notfound THEN
        LEAVE CompanyLoop;
    END IF;
    IF ThisValue > TopValue THEN
        SET TopCompany = ThisName;
        SET TopValue = ThisValue;
    END IF;
END LOOP CompanyLoop;
-- 7. Close the cursor
CLOSE ThisCompany;
END;
```

注意

この TopCustomerValue プロシージャには、次の特徴があります。

- 例外が宣言されます。この例外は、プロシージャの後にクエリの結果のループが完了するときに通知されます。
- クエリの各ローの結果を入れる2つのローカル変数 `ThisName` と `ThisValue` が宣言されます。
- カーソル `ThisCompany` が宣言されます。`SELECT` 文は会社名とその会社からの注文の合計額のリストを作成します。
- ループで使うため、`TopValue` の初期値は0に設定されています。
- `ThisCompany` カーソルが開きます。
- `LOOP` 文はクエリの各ローをループして、各会社の名前を変数 `ThisName` と `ThisValue` に入れます。`ThisValue` が現在の最大値よりも大きければ、`TopCompany` と `TopValue` は `ThisName` と `ThisValue` の値にリセットされます。
- プロシージャの最後にカーソルは閉じられます。
- `SELECT` 文に `ORDER BY` 値 `DESC` 句を追加して、ループを使わずにこのプロシージャを作成することもできます。その場合、カーソルの最初のローのみをフェッチする必要があります。

この `TopCompanyValue` プロシージャでの `LOOP` 文は標準的な形式で、最後のローを処理して終了します。`FOR` ループを使うと、このプロシージャはさらに簡潔になります。`FOR` 文は1つの文に、上記のプロシージャのいくつかの要素を組み込みます。

```
CREATE PROCEDURE TopCustomerValue2(
    OUT TopCompany CHAR(36),
    OUT TopValue INT )
BEGIN
    -- 1. Initialize the TopValue variable
    SET TopValue = 0;
    -- 2. Do the For Loop
    FOR CompanyFor AS ThisCompany
    CURSOR FOR
        SELECT CompanyName AS ThisName,
            CAST( sum( SalesOrderItems.Quantity *
                Products.UnitPrice ) AS INTEGER )
            AS ThisValue
        FROM Customers
            INNER JOIN SalesOrders
            INNER JOIN SalesOrderItems
            INNER JOIN Products
        GROUP BY ThisName
    DO
        IF ThisValue > TopValue THEN
            SET TopCompany = ThisName;
            SET TopValue = ThisValue;
        END IF;
    END FOR;
END;
```

IQ UTILITIES を使用したストアードプロシージャの作成

SAP Sybase IQ に用意されているシステムストアードプロシージャは、SQL で実装されています。

ローカルテンポラリテーブルと **IQ UTILITIES** 文は、システムストアードプロシージャとまったく同じように使用します。

プロシージャのすべての SQL コードは暗号化され、共有ライブラリ (UNIX では `libiqscripts16_r.so` ファイル、Windows では `iqscripts16.dll` ファイル) に集約されます。

警告！ 文は正しく使用しないと、IQ サーバやデータベースに重大な問題が生じる可能性があります。

ストアードプロシージャコードを表示するには、Interactive SQL で `sp_helptext 'owner.procname'` を入力します。

IQ UTILITIES の構文は次のとおりです。

IQ UTILITIES MAIN INTO *local-temp-table-name arguments*

プロシージャから派生した独自のプロシージャを作成できます。次に例を示します。

- システムストアードプロシージャを呼び出すプロシージャを作成します。
- システムストアードプロシージャとは別個に、同様の機能を実行するプロシージャを作成します。
- システムストアードプロシージャと同じ構造を使用し、かつ独自の機能を追加したプロシージャを作成します。たとえば、フロントエンドツールやブラウザで、テキストではなくグラフィカル形式でプロシージャの結果を表示するなどです。
- 前記の2つのオプションのいずれかを選択する場合、**IQ UTILITIES** 文と、その使用方法の厳密な規則を理解する必要があります。

IQ UTILITIES コマンド

IQ UTILITIES は、ほとんどの IQ のシステムプロシージャで、実行時に内部で動作する文です。多くの場合、**IQ UTILITIES** が動作していることは、ユーザの関知するところではありません。**IQ UTILITIES** をユーザが直接実行するのは、IQ バッファ キャッシュモニタを起動するときのみです。

IQ UTILITIES を使用することで、IQ のシステムテーブルに保持されている情報を体系的に収集およびレポートできます。一般的なユーザインタフェースはありません。既存のシステムプロシージャと同様に、**IQ UTILITIES** のみで使用できます。

システムプロシージャは、情報を格納するローカルテンポラリテーブルを宣言します。システムプロシージャは、**IQ UTILITIES** を実行してシステムテーブルから情

報を取得し、それをローカルテンポラリテーブルに格納します。システムプロシージャは、ローカルテンポラリテーブルから情報をレポートするだけのこともあれば、追加的な処理を実行することもあります。

システムプロシージャの中には、事前に定義された番号を **IQ UTILITIES** 文の引数の 1 つとして指定するものもあります。この番号に応じて特定の機能 (たとえば、システムテーブルの情報から値を得るなど) が実行されます。

呼び出すプロシージャの選択

データベースの情報をレポートする、作成済みのシステムプロシージャについては、**IQ UTILITIES** を使用して独自バージョンを作成するのが安全です。

たとえば、**sp_iqspaceused** は、IQ メインストアと IQ テンポラリストアの使用済み領域と空き領域についての情報を表示します。システムストアプロシージャから作成するプロシージャの所有者が正しいことを確認します。

IQ の処理を制御するシステムプロシージャについては、独自バージョンを作成しないでください。IQ の処理を制御するプロシージャを修正すると、重大な問題につながる可能性があります。

IQ UTILITIES で使用される番号

番号は **IQ UTILITIES** コマンドで引数として使用されます。各番号は、システムプロシージャと組み合わせて使用されます。

表 21 : システムプロシージャで使用される **IQ UTILITIES** の番号

番号	プロシージャ	コメント
10000	sp_iqtransaction	
20000	sp_iqconnection および sp_iqmpxcountdbremote	
30000	sp_iqspaceused	
40000	sp_iqspaceinfo	
50000	sp_iqlocks	
60000	sp_iqmpxversionfetch	使用できない。
70000	sp_iqmpxdumpltivlog	
80000	sp_iqcontext	
100000	sp_iqindexfragmentation	
110000	sp_iqrowdensity	

プロシージャのテスト

IQ サーバとデータベースの安定性を確保するには、運用環境で実行する前に開発環境でプロシージャをテストします。

ユーザ定義関数

ユーザ定義関数はプロシージャの集まりで、呼び出しを行った環境に単一の値を返します。

ユーザ定義関数の作成

ユーザ定義関数を作成するには、**CREATE FUNCTION** 文を使用します。

前提条件

- ユーザ本人が所有する関数を作成するには、**CREATE PROCEDURE** システム権限が必要です。
- 他のユーザが所有する関数を作成するには、**CREATE ANY PROCEDURE** または **CREATE ANY OBJECT** のシステム権限が必要です。
- プロシージャに外部参照が含まれる場合は、前述のシステム権限に加えて **CREATE EXTERNAL REFERENCE** システム権限も必要です。

SAP Control Center でユーザ定義関数を作成するには、SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

CREATE FUNCTION 文

データベースにユーザ定義関数を作成します。所有者名を指定することにより、別のユーザが使用する関数を作成できます。パーミッションがあれば、ユーザ定義関数も他の非集合関数とまったく同じように使用できます。

クイックリンク：

「パラメータ」 (275 ページ)

「例」 (279 ページ)

「使用法」 (280 ページ)

「標準」 (281 ページ)

「パーミッション」 (281 ページ)

構文

構文 1

```
CREATE [ OR REPLACE ] [ TEMPORARY ] FUNCTION [ owner.]function-name
( [ parameter, ... ] )
[ SQL SECURITY { INVOKER | DEFINER } ]
RETURNS data-type ON EXCEPTION RESUME
| [ NOT ] DETERMINISTIC
{ compound-statement | AS tsq-compound-statement
| EXTERNAL NAME library-call
| EXTERNAL NAME java-call LANGUAGE JAVA }
```

構文 2

```
CREATE FUNCTION [ owner.]function-name ( [ parameter, ... ] )
RETURNS data-type
URL url-string
[ HEADER header-string ]
[ SOAPHEADER soap-header-string ]
[ TYPE { 'HTTP[:{ GET | POST }]' | 'SOAP[:{ RPC | DOC }]' } ]
[ NAMESPACE namespace-string ]
[ CERTIFICATE certificate-string ]
[ CLIENTPORT clientport-string ]
[ PROXY proxy-string ]
```

parameter - (back to Syntax 1) or (back to Syntax 2)
IN parameter-name data-type [**DEFAULT** expression]

tsq-compound-statement - (back to Syntax 1)
 sql-statement
 sql-statement ...

library-call - (back to Syntax 1)
 '[**operating-system:**]function-name@library; ...'

operating-system - (back to library-call)
 UNIX

java-call - (back to Syntax 1)
 '[package-name.]class-name.method-name **method-signature**'

method-signature - (back to java-call)
 ([field-descriptor, ...]) **return-descriptor**

field-descriptor and **return-descriptor** - (back to method-signature)
Z | **B** | **S** | **I** | **J** | **F** | **D** | **C** | **V** | [descriptor | **L** class-name;

url-string - (back to Syntax 2)
 '{ HTTP | HTTPS | HTTPS_FIPS }://[user:password@]hostname[:port] [/path]'

パラメータ

(先頭に戻る) (274 ページ)

- **CREATE [OR REPLACE]** – パラメータ名は、データベース識別子に関するルールに従って付ける必要があります。また、有効な SQL データ型でなければなりません。さらに、この引数が関数に値を渡す式であることを示すキーワード **IN** を先頭に付ける必要があります。

CREATE 句を指定すると新しい関数が作成され、**OR REPLACE** 句を使用すると既存の関数が同じ名前でも置き換えられます。関数が置き換えられた場合、関数の定義は変更されますが、既存のパーミッションは保持されます。**OR REPLACE** 句をテンポラリ関数で使用することはできません。

- **TEMPORARY** – 作成した接続でのみ参照できる関数になり、接続を削除すると関数も自動的に削除されます。テンポラリ関数を明示的に削除することもできます。テンポラリ関数に対して **ALTER**、**GRANT**、および **REVOKE** 操作を実行することはできません。また他の関数とは異なり、テンポラリ関数はカタログやトランザクションログに記録されません。

テンポラリ関数は、作成者 (現在のユーザ) のパーミッションで実行されます。また作成者のみが所有できます。そのため、テンポラリ関数を作成するときは所有者を指定しないでください。読み込み専用のデータベースに接続するときに、テンポラリ関数の作成と削除を行うことができます。

- **SQL SECURITY – INVOKER** (関数を呼び出しているユーザ) または **DEFINER** (関数を所有しているユーザ) のどちらとして関数が実行されるかを定義します。デフォルトは **DEFINER** です。

INVOKER を指定すると、プロシージャを呼び出す各ユーザに対して注釈を付ける必要があるため、より多くのメモリが使用されます。また、名前解決も **INVOKER** として実行されます。そのため、適切な所有者で、すべてのオブジェクト名 (テーブル、プロシージャなど) を修飾するように注意してください。

- **data-type** – **LONG BINARY** と **LONG VARCHAR** を戻り値のデータ型として使用することはできません。
- **compound-statement** – **BEGIN** と **END** で囲まれ、セミコロンで区切られた SQL 文のセットです。「**BEGIN ... END** 文」を参照してください。
- **tsql-compound-statement** – Transact-SQL 文のバッチです。
- **external-name** – 外部ライブラリにある関数への呼び出しのラッパーです。**RETURNS** 句の後に他の句を持つことはできません。ライブラリ名にはファイル拡張子が付く場合があります。この拡張子は通常、Windows では **.dll**、UNIX では **.so** です。拡張子がない場合、ライブラリに対するプラットフォーム固有のデフォルトのファイル拡張子が追加されます。

external-name 句は、テンポラリ関数ではサポートされていません。

- **LANGUAGE JAVA** – Java メソッドのラッパーです。Java プロシージャの呼び出しについては、「CREATE PROCEDURE 文」を参照してください。
- **ON EXCEPTION RESUME** – Transact-SQL のようなエラー処理を使用します。「CREATE PROCEDURE 文」を参照してください。
- **[NOT] DETERMINISTIC** – クエリで呼び出されるたびに、関数が再評価されます。このように指定されていない関数の結果は、パフォーマンスを向上させるためにキャッシュされ、クエリの評価中に同じパラメータで関数が呼び出されるたびに再利用されます。

基本となるデータを修正するなどの関連動作を伴う関数は、**NOT DETERMINISTIC** として宣言する必要があります。たとえば、プライマリキー値を生成し、**INSERT ... SELECT** 文で使用される関数は、次のように **NOT DETERMINISTIC** として宣言してください。

```
CREATE FUNCTION keygen( increment INTEGER )
RETURNS INTEGER
NOT DETERMINISTIC
BEGIN
    DECLARE keyval INTEGER;
    UPDATE counter SET x = x + increment;
    SELECT counter.x INTO keyval FROM counter;
    RETURN keyval
END
INSERT INTO new_table
SELECT keygen(1), ...
FROM old_table
```

特定の入力パラメータに対して常に同じ値を返す関数は、**DETERMINISTIC** として宣言できます。**NOT DETERMINISTIC** として宣言しないかぎり、ユーザ定義関数はすべて **DETERMINISTIC** 関数として処理されます。**DETERMINISTIC** 関数は同じパラメータに対して一貫した結果を返し、副次的影響はありません。つまり、データベースサーバは、同じ関数を同じパラメータで2回連続して呼び出した場合は同じ結果が返され、クエリのセマンティックに不要な副次的影響が生じることはない想定します。

- **URL** – HTTP または SOAP Web サービスクライアント関数を定義する場合のみ使用します。Web サービスの URL を指定します。オプションのユーザ名とパスワードのパラメータは、HTTP 基本認証に必要なクレデンシャルとして機能します。HTTP 基本認証は、ユーザとパスワードの情報を base-64 でエンコードし、HTTP 要求の "Authentication" ヘッダに渡します。

Web サービスクライアント関数の場合、SOAP 関数と HTTP 関数の戻り型は、**VARCHAR** などの文字データ型のいずれかでなければなりません。返される値は、HTTP 応答の本文です。HTTP ヘッダ情報は含まれません。ステータス情

報などの詳細情報が必要な場合は、関数の代わりにプロシージャを使用します。

パラメータ値は要求の一部として渡されます。使用される構文は、要求のタイプによって決まります。HTTP:GET の場合、パラメータは URL の一部として渡されます。HTTP:POST 要求の場合、値は要求の本文に置かれます。SOAP 要求へのパラメータは、常に要求本文にバンドルされます。

- **HEADER** – HTTP Web サービスクライアント関数を作成する場合は、この句を使用して HTTP 要求ヘッダのエントリを追加または変更します。HTTP ヘッダに指定できるのは印字可能な ASCII 文字のみで、大文字と小文字は区別されません。この句の使用法の詳細については、「CREATE PROCEDURE 文」の HEADER 句を参照してください。
- **SOAPHEADER** – SOAP Web サービスを関数として宣言する場合は、この句を使用して 1 つ以上の SOAP 要求ヘッダのエントリを指定します。SOAP ヘッダは、静的定数として宣言したり、代入パラメータメカニズムを使用して動的に設定したりできます (hd1、hd2 などに IN、OUT、または INOUT パラメータを宣言)。Web サービス関数では、1 つ以上の IN モード代入パラメータを定義できますが、INOUT または OUT 代入パラメータは定義できません。
- **TYPE** – Web サービス要求を行う場合に使用するフォーマットを指定します。SOAP が指定されている場合、または type 句が含まれていない場合は、デフォルトのタイプである SOAP:RPC が使用されます。HTTP は HTTP:POST を暗黙的に指定します。SOAP 要求は常に XML 文書として送信されるため、SOAP 要求の送信には常に HTTP:POST が使用されます。
- **NAMESPACE** – SOAP クライアント関数のみに適用されます。この句は、SOAP:RPC 要求と SOAP:DOC 要求の両方に通常必要なメソッドネームスペースを示します。要求を処理する SOAP サーバは、このネームスペースを使用して、SOAP 要求メッセージ本文内のエンティティの名前を解釈します。ネームスペースは、Web サービスサーバから使用できる SOAP サービスの WSDL 記述から取得できます。デフォルト値は、プロシージャの URL のオプションのパスコンポーネントの直前までです。
- **CERTIFICATE** – 安全な (HTTPS) 要求を行うには、HTTPS サーバで使用される証明書にクライアントがアクセスできる必要があります。必要な情報は、セミコロンで区切られたキー／値のペアの文字列で指定されます。証明書はファイルに置かれ、file キーを使用して提供されるファイルの名前、または証明書全体を文字列に配置できますが、両方は配置できません。次のキーを使用できます。

キー	省略形	説明
file		証明書のファイル名
certificate	cert	証明書自体
company	co	証明書で指定された会社
unit		証明書で指定された会社の部門
name		証明書で指定された共通名

証明書は、HTTPS サーバに対する直接の要求、または安全でないサーバから安全なサーバにリダイレクトされる可能性がある要求に対してのみ必要です。

- **CLIENTPORT** – HTTP クライアントプロシージャが TCP/IP を使用して通信するポート番号を示します。この句は、ファイアウォールを介する通信のためのものであり、このような通信にのみおすすめます。ファイアウォールは TCP/UDP ポートに従ってフィルタします。単一のポート番号、ポート番号の範囲、または両方の組み合わせを指定できます。たとえば、CLIENTPORT '85,90-97' を指定できます。
- **PROXY** – プロキシサーバの URI を指定します。クライアントがプロキシを介してネットワークにアクセスする場合に使用します。プロシージャがプロキシサーバに接続し、そのプロキシサーバを介して Web サービスに要求を送信することを示します。

例

(先頭に戻る) (274 ページ)

- **例 1** – 文字列 `firstname` と文字列 `lastname` を連結します。

```
CREATE FUNCTION fullname (
    firstname CHAR(30),
    lastname CHAR(30) )
RETURNS CHAR(61)
BEGIN
    DECLARE name CHAR(61);
    SET name = firstname || ' ' || lastname;
    RETURN (name);
END
```

fullname 関数は、次の例のように使用します。

- 2つの提供された文字列からフルネームを戻します。

```
SELECT fullname ('joe','smith')
```

fullname('joe', 'smith')
joe smith

- 全従業員の名前をリストします。

```
SELECT fullname (givenname, surname)
FROM Employees
```

fullname (givenname, surname)
Fran Whitney
Matthew Cobb
Philip Chin
Julie Jordan
Robert Breault
...

- **例 2** – Transact-SQL 構文を使用しています。

```
CREATE FUNCTION DoubleIt ( @Input INT )
RETURNS INT
AS
DECLARE @Result INT
SELECT @Result = @Input * 2
RETURN @Result
```

文 `SELECT DoubleIt(5)` は、10 の値を返します。

- **例 3** – Java で記述された外部関数を作成します。

```
CREATE FUNCTION dba.encrypt( IN name char(254) )
RETURNS VARCHAR
EXTERNAL NAME
'Scramble.encrypt (Ljava/lang/String;)Ljava/lang/String;'
LANGUAGE JAVA
```

使用法

(先頭に戻る) (274 ページ)

ユーザ定義関数を修正したり、定義にスクランブルをかけて関数の内容を隠したりするには、**ALTER FUNCTION** 文を使用します。

関数を実行するときに、必ずしもすべてのパラメータを指定する必要はありません。**CREATE FUNCTION** 文の中にデフォルト値がある場合、パラメータが指定されていないと、デフォルト値が割り当てられます。呼び出し元によって引数が指定されておらず、デフォルトも設定されていない場合は、エラーが発生します。

関連する動作：

- オートコミット

標準

(先頭に戻る) (274 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。

パーミッション

(先頭に戻る) (274 ページ)

自分が所有する関数の場合、CREATE PROCEDURE システム権限が必要です。

任意のユーザが所有する関数の場合、次のいずれかが必要です。

- CREATE ANY PROCEDURE システム権限
- CREATE ANY OBJECT システム権限

外部参照を含む関数を作成するには、関数の所有者であるかどうかに関係なく、CREATE EXTERNAL REFERENCE システム権限も必要です。

ユーザ定義関数の呼び出し

ユーザ定義関数は、パーミッションがあれば、集合関数以外の組み込み関数を使用できる同じ場所で使用できます。

次の Interactive SQL 文は、姓と名前の入った 2 つのカラムから氏名を返します。

```
SELECT fullname (GivenName, LastName)
FROM Employees;
```

fullname (Employees.GivenName,Employees.LastName)
Fran Whitney
Matthew Cobb
Philip Chin
...

次の文は、文中に提供された姓と名前から氏名を表示します。

```
SELECT fullname ('Jane', 'Smith');
```

fullname ('Jane','Smith')
Jane Smith

関数に対する `Execute` パーミッションを付与されたユーザは `fullname` 関数を使用できます。

ユーザ定義関数の削除 (SQL の場合)

ユーザ定義関数は、明示的に削除されるまでデータベースに残ります。

前提条件

ユーザ定義関数の所有者であるか、または次のいずれかのシステム権限を持っている必要があります。

```
DROP ANY PROCEDURE  
DROP ANY OBJECT
```

手順

1. データベースに接続します。
2. 次のような `DROP FUNCTION` 文を実行します。

```
DROP FUNCTION function-name;
```

ユーザ定義関数が削除されます。

ユーザ定義関数を実行するためのパーミッション

ユーザ定義関数の所有者はそれを作成したユーザです。所有者は権限なしでそれを実行できます。

ユーザ定義関数の所有者は、**GRANT EXECUTE** コマンドを使用して、他のユーザに権限を付与できます。

たとえば、関数 `fullname` の作成者は、次の文を使用して `fullname` を使用するパーミッションを `another_user` に付与できます。

```
GRANT EXECUTE ON fullname TO another_user
```

この文は、関数を使用する権限を取り消します。

```
REVOKE EXECUTE ON fullname FROM another_user
```

『管理：ユーザ管理とセキュリティ』 > 「セキュリティ管理」 > 「権限」 > 「オブジェクトレベル権限」 > 「オブジェクトレベル権限の付与と取り消し」 > 「関数とプロシージャに対する EXECUTE 権限の付与」を参照してください。

ユーザ定義関数実行機能の付与 (SQL の場合)

EXECUTE オブジェクトレベル権限を付与して、ユーザ定義関数を実行する機能を付与します。

前提条件

ユーザ定義関数の所有者であるか、またはその関数の EXECUTE 権限に対する管理権限を持っていることが必要です。

ユーザ定義関数の所有権はその関数の作成者に属しており、そのユーザが関数を実行する場合には権限は必要ありません。

手順

関数を作成して、他のユーザがその関数を実行できるようにします。

1. データベースに接続します。
2. 次のような GRANT EXECUTE 文を実行します。

```
GRANT EXECUTE ON function-name TO user-id;
```

被付与者がプロシージャを実行できるようになりました。

次のステップ

なし。

バッチ

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。プロシージャ (CASE、IF、LOOP など) で使われる制御文はバッチでも使えます。バッチが BEGIN/END で囲まれた複合文で構成される場合、ホスト変数、変数のローカル宣言、カーソル、テンポラリテーブル、例外を含めることもできます。ホスト変数参照は、次の制限付きでバッチ内で使用できます。

- ホスト変数を参照できるのはバッチ内の 1 文だけです。
- ホスト変数を使用する文の前に、結果セットを返す文を入れることはできません。

バッチの使用を明確に示すために、BEGIN/END を使うことをおすすめします。

バッチ内の文はセミコロンで区切ることができます。その場合、バッチは Watcom-SQL 構文に準拠しています。文を区切るためにセミコロンを使用しない複数文のバッチは、Transact-SQL 構文に準拠します。バッチの構文によって、バッチ内で使用できる文とバッチ内でのエラーの処理方法が決まります。

プロシージャとバッチの作成

多くの点で、バッチはストアドプロシージャに似ていますが、いくつかの違いがあります。

- バッチには名前がありません。
- バッチにはパラメータを使用できません。
- バッチは永続的にデータベースに保存されません。
- バッチは異なる接続で共有できません。

簡単なバッチは、デリミタのない SQL 文のセットで、次の行に go という単語が続きます。次の例では、Eastern Sales という部署を作成し、Massachusetts のすべての営業担当者を Eastern Sales に転送します。これは Transact-SQL バッチの例です。

```
INSERT
INTO Departments ( DepartmentID, DepartmentName )
VALUES ( 220, 'Eastern Sales' )

UPDATE Employees
SET DepartmentID = 220
WHERE DepartmentID = 200
AND State = 'MA'

COMMIT
go
```

go という単語は Interactive SQL によって認識され、前の文は 1 つのバッチとしてサーバに送信されます。

次の例は見た目は似ていますが、Interactive SQL での処理はまったく異なります。この例では、Transact-SQL 構文を使用しません。各文はセミコロンで区切られています。Interactive SQL はセミコロンで区切られた各文を個別にサーバに送信します。この場合は、バッチとしては処理されません。

```
INSERT
INTO Departments ( DepartmentID, DepartmentName )
VALUES ( 220, 'Eastern Sales' );

UPDATE Employees
SET DepartmentID = 220
WHERE DepartmentID = 200
AND State = 'MA';

COMMIT;
```

Interactive SQL でバッチとして処理するには、BEGIN ... END を使用して、複合文に変更します。次の構文は、前の例を修正したものです。複合文に含まれる 3 つの文は、バッチとしてサーバに送信されます。

```
BEGIN
  INSERT
  INTO Departments ( DepartmentID, DepartmentName )
  VALUES ( 220, 'Eastern Sales' );

  UPDATE Employees
```

```
SET DepartmentID = 220
WHERE DepartmentID = 200
AND State = 'MA';

COMMIT;
END
```

この例の場合、サーバがバッチと個別のどちらで文を実行しても結果は同じになります。ただし、結果が異なる場合もあります。次の例を考えます。

```
DECLARE @CurrentID INTEGER;
SET @CurrentID = 207;
SELECT Surname FROM Employees
WHERE EmployeeID=@CurrentID;
```

Interactive SQL を使用してこの例を実行すると、データベースサーバが変数が見つからないことを示すエラーを返します。このエラーは、Interactive SQL が3つの文を個別にサーバに送信するために発生します。これらの文はバッチとしては実行されません。このようなエラーに対処するには、複合文を使用して Interactive SQL が強制的に3つの文をバッチとしてサーバに送信するようにします。次の例では、複合文を使用しています。

```
BEGIN
  DECLARE @CurrentID INTEGER;
  SET @CurrentID = 207;
  SELECT Surname FROM Employees
    WHERE EmployeeID=@CurrentID;
END
```

一連の文を BEGIN と END で囲んだ場合、Interactive SQL は強制的に文をバッチとして処理します。

IF 文は複合文の一例です。Interactive SQL は、次の文を1つのバッチとしてサーバに送信します。

```
IF EXISTS ( SELECT *
            FROM SYSTAB
            WHERE table_name='Employees' )
THEN
  SELECT Surname AS LastName,
         GivenName AS FirstName
  FROM Employees;
  SELECT Surname, GivenName
  FROM Customers;
  SELECT Surname, GivenName
  FROM Contacts;
ELSE
  MESSAGE 'The Employees table does not exist'
  TO CLIENT;
END IF
```

別の方法で SQL 文を作成して実行した場合、この例は適用されません。たとえば、ODBC を使用するアプリケーションでは、セミコロンで区切られた文をバッチとして作成および実行できます。

プロシージャとバッチの作成

Interactive SQL の文とサーバ向けの SQL 文が混在している場合は、注意が必要です。次の例は、Interactive SQL の文と SQL 文を混合する場合の問題を示します。この例では、Interactive SQL の OUTPUT 文が複合文に組み込まれているので、その他のすべての文と一緒にバッチとしてサーバに送信され、構文エラーが発生します。

```
IF EXISTS ( SELECT *
            FROM SYSTAB
            WHERE table_name='Employees' )
THEN
    SELECT Surname AS LastName,
           GivenName AS FirstName
    FROM Employees;
    SELECT Surname, GivenName
    FROM Customers;
    SELECT Surname, GivenName
    FROM Contacts;
    OUTPUT TO 'c:YYtempYYquery.txt';
ELSE
    MESSAGE 'The Employees table does not exist'
    TO CLIENT;
END IF
```

正しい OUTPUT 文の例は、次のとおりです。

```
IF EXISTS ( SELECT *
            FROM SYSTAB
            WHERE table_name='Employees' )
THEN
    SELECT Surname AS LastName,
           GivenName AS FirstName
    FROM Employees;
    SELECT Surname, GivenName
    FROM Customers;
    SELECT Surname, GivenName
    FROM Contacts;
ELSE
    MESSAGE 'The Employees table does not exist'
    TO CLIENT;
END IF;
OUTPUT TO 'c:YYtempYYquery.txt';
```


制御文

プロシージャ、トリガ、またはユーザ定義関数の本文、またはバッチの中には、論理フローや意思決定のための制御文が複数あります。使用可能な制御文は、次のとおりです。

制御文	構文
複合文	<pre>BEGIN [ATOMIC] Statement-list END</pre>
条件実行：IF	<pre>IF condition THEN Statement-list ELSEIF condition THEN Statement-list ELSE Statement-list END IF</pre>
条件実行：CASE	<pre>CASE expression WHEN value THEN Statement-list WHEN value THEN Statement-list ELSE Statement-list END CASE</pre>
繰り返し：WHILE、LOOP	<pre>WHILE condition LOOP Statement-list END LOOP</pre>
繰り返し：FOR カーソルループ	<pre>FOR loop-name AS cursor-name CURSOR FOR select-statement DO Statement-list END FOR</pre>
中断：LEAVE	<pre>LEAVE label</pre>
CALL	<pre>CALL procname(arg, ...)</pre>

複合文

複合文はキーワード `BEGIN` で始まり、キーワード `END` で終わります。プロシージャまたはトリガの本文は**複合文**です。また、バッチでも使うことができます。

プロシージャとバッチの作成

複合文はネストでき、他の制御文とともにプロシージャ、トリガ、またはバッチの実行フローを定義します。

複合文は、SQL 文のセットをまとめて1つの単位として扱えるようにします。複合文の中の SQL 文はセミコロンで区切ります。

複合文での宣言

複合文中のローカル宣言は、キーワード **BEGIN** のすぐ後に続きます。このローカル宣言は複合文中にのみ存在します。複合文に次のものを宣言できます。

- 変数
- カーソル
- テンポラリテーブル
- 例外処理 (エラー識別子)

ローカル宣言は、複合文またはその中でネストされる複合文の中のどの文からでも参照できます。ローカル宣言は、複合文中から呼び出された他のプロシージャからは見えません。

アトミックな複合文

アトミックな文とは、完全に実行されるか、まったく実行されない文のことです。たとえば、何千ものローを更新する **UPDATE** 文では、たくさんのローを更新した後にエラーが発生することがあります。文が完了しないと、変更されたすべてのローが元の状態に戻ります。したがって、**UPDATE** 文はアトミックです。

複合文でないすべての SQL 文はアトミックです。**BEGIN** キーワードの後にキーワード **ATOMIC** を追加して、複合文をアトミックにすることができます。

```
BEGIN ATOMIC
  UPDATE Employees
  SET ManagerID = 501
  WHERE EmployeeID = 467;
  UPDATE Employees
  SET BirthDate = 'bad_data';
END
```

この例の2つの **UPDATE** 文は、アトミックな複合文の一部です。これら2つの文は、1つの文として更新を完了するか、両方ともエラーになります。最初の **UPDATE** 文はエラーなしで完了するとします。次の **UPDATE** 文は **BirthDate** カラムに割り当てた値を日付に変換できないため、エラーになります。

このアトミックな複合文はエラーになり、**UPDATE** 文の結果は両方とも取り消されます。現在実行中のトランザクションがコミットされても、この複合文中の文は両方ともその効果をもたらしません。

アトミックな複合文が成功すると、現在実行中のトランザクションがコミットされた場合のみ、複合文中で実行された変更は有効になります。アトミックな複合文が成功しても、その文で発生したトランザクションがロールバックされた場合

は、アトミックな複合文もロールバックされます。アトミックな複合文の開始時に、セーブポイントが設定されます。文でエラーが発生すると、そのセーブポイントにロールバックされます。

アトミックな複合文がオートコミット (非連鎖) モードで実行されると、文の実行が完了するまでコミットモードが手動 (連鎖) に変更されます。手動モードでは、アトミックな複合文内で DML 文を実行しても、即座に COMMIT や ROLLBACK は実行されません。アトミックな複合文が正常に完了すると、COMMIT 文が実行されます。正常に完了しない場合は ROLLBACK 文が実行されます。

COMMIT 文と ROLLBACK 文、および一部の ROLLBACK TO SAVEPOINT 文は、アトミックな複合文内で使用できません。

プロシージャの構造

プロシージャの本文は複合文で構成されます。

複合文は、一連の SQL 文を囲む **BEGIN** と **END** で構成されています。各文はセミコロンで区切られています。

プロシージャで使用可能な SQL 文

プロシージャでは、ほぼすべての SQL 文を使用できます。

- **SELECT**、**UPDATE**、**DELETE**、**INSERT**、および **SET VARIABLE**
- 他のプロシージャを実行する **CALL** 文
- 制御文
- **CURSOR** 文
- 例外処理文
- **EXECUTE IMMEDIATE** 文

一部の SQL 文はプロシージャ内で使用できません。

- **CONNECT** 文
- **DISCONNECT** 文

プロシージャ内で **COMMIT** 文、**ROLLBACK** 文、**SAVEPOINT** 文を使用できますが、特定の制限が適用されます。

リファレンス：文とオプション > SQL 文の各文については、「使用法」を参照してください。

プロシージャのパラメータ宣言

プロシージャパラメータは、CREATE PROCEDURE 文にリストとして表示されません。パラメータ名は、カラム名など他のデータベース識別子に対するルールに

プロシージャとバッチの作成

従って付けてください。パラメータは有効なデータ型で、キーワード IN、OUT、INOUT のいずれかのプレフィクスが付いています。デフォルトでは、パラメータは INOUT パラメータです。これらのキーワードには、次のような意味がありません。

- **IN** - 引数はプロシージャに値を提供する式です。
- **OUT** - 引数はプロシージャから値を与えられる変数です。
- **INOUT** - 引数はプロシージャに値を提供する変数で、プロシージャから新しい値を与えられることもあります。

CREATE PROCEDURE 文中のプロシージャパラメータにはデフォルト値を設定できます。デフォルト値は定数で、NULL でもかまいません。たとえば、次に示すプロシージャは、IN パラメータのデフォルトとして NULL を指定しています。これは意味のないクエリを実行するのを避けるためです。

```
CREATE PROCEDURE CustomerProducts (
    IN customer_ID
        INTEGER DEFAULT NULL )
RESULT ( product_ID INTEGER,
        quantity_ordered INTEGER )
BEGIN
    IF customer_ID IS NULL THEN
        RETURN;
    ELSE
        SELECT      Products.ID,
                   sum( SalesOrderItems.Quantity )
        FROM        Products,
                   SalesOrderItems,
                   SalesOrders
        WHERE SalesOrders.CustomerID = customer_ID
        AND SalesOrders.ID = SalesOrderItems.ID
        AND SalesOrderItems.ProductID = Products.ID
        GROUP BY Products.ID;
    END IF;
END;
```

次に示す文は DEFAULT NULL を割り当て、プロシージャはクエリを実行しないで戻ります。

```
CALL CustomerProducts ();
```

プロシージャにパラメータを渡す方法

ストアドプロシージャパラメータのデフォルト値は、CALL 文の2通りの形式のどちらでも使用できます。

CREATE PROCEDURE 文の引数リストの末尾にオプションのパラメータがある場合、これらは CALL 文で省略できます。次に示すのは、INOUT パラメータを3つ持つプロシージャの例です。

```
CREATE PROCEDURE SampleProcedure (
    INOUT var1 INT DEFAULT 1,
```

```

        INOUT var2 int DEFAULT 2,
        INOUT var3 int DEFAULT 3 )
...

```

この例では、プロシージャを呼び出す環境で、プロシージャに渡す数値を格納するための変数を3つ設定してあるものと想定しています。

```

CREATE VARIABLE V1 INT;
CREATE VARIABLE V2 INT;
CREATE VARIABLE V3 INT;

```

次に示すように、SampleProcedure プロシージャは最初のパラメータを指定するだけで呼び出せます。この場合、var2 と var3 にはデフォルト値が使用されます。

```
CALL SampleProcedure( V1 );
```

また、次のように、プロシージャは最初のパラメータには DEFAULT 値を使用して、2番目のパラメータのみを指定することによっても呼び出せます。

```
CALL SampleProcedure( DEFAULT, V2 );
```

オプションの引数を使ってプロシージャを呼び出すよりも柔軟な方法は、パラメータに名前を付けて渡すという方法です。このとき、SampleProcedure プロシージャは次のように呼び出すことができます。

```
CALL SampleProcedure( var1 = V1, var3 = V3 );
```

または次のようになります。

```
CALL SampleProcedure( var3 = V3, var1 = V1 );
```

関数にパラメータを渡す方法

ユーザ定義関数は CALL 文で呼び出すのではなく、組み込み関数と同じように使用できます。たとえば、次の文は FullName 関数を使用して従業員の名前を取り出します。

注意

- デフォルトパラメータは呼び出し関数でも使用できます。ただしパラメータは、名前を付けて関数に渡すことはできません。
- パラメータは参照ではなく、値で渡されます。関数とそのパラメータの値を変更しても、その変更は関数を呼び出した環境には戻されません。
- ユーザ定義関数では出力パラメータは使用できません。
- ユーザ定義関数は結果セットを返すことはできません。

例：全従業員の名前のリスト

Interactive SQL で次のクエリを実行します。

```

SELECT FullName( GivenName, Surname ) AS Name
FROM Employees;

```

次の結果が表示されます。

Name
Fran Whitney
Matthew Cobb
Philip Chin
Julie Jordan
...

プロシージャの結果

プロシージャは、1つまたは複数のデータローを返します。

単一のローのデータからなる結果の場合は、プロシージャへの引数で返すことができます。複数のローのデータからなる結果の場合は、結果セットで返します。また、プロシージャは **RETURN** 文で1つの値を返すこともできます。

RETURN 文を使って値を返す

RETURN 文は、呼び出しを行った環境に1つの整数値を返した後、すぐにプロシージャを終了します。

前提条件

この作業を実行するための前提条件はありません。

手順

1. 次の文を実行します。

```
RETURN expression
```

2. 式の値が、呼び出しを行った環境に返されます。返ってきた値を変数に保存するには、CALL 文の拡張機能を使います。

```
CREATE VARIABLE returnval INTEGER;  
returnval = CALL variable/procedure-name? myproc();
```

値が返され、変数として保存されます。

結果をプロシージャのパラメータとして返す方法

プロシージャは、プロシージャのパラメータで呼び出しを行った環境に結果を返すことができます。次の文を使用して、プロシージャ内でパラメータと変数に値を割り当てることができます。

- SET 文

次に示すプロシージャは、SET 文を使用して OUT パラメータに値を割り当てて返します。次の文を実行するには、CREATE PROCEDURE システム権限を持っている必要があります。

```
CREATE PROCEDURE greater(
    IN a INT,
    IN b INT,
    OUT c INT )
BEGIN
    IF a > b THEN
        SET c = a;
    ELSE
        SET c = b;
    END IF ;
END;
```

- INTO 句を持つ SELECT 文

「シングルロクエリ」がデータベースから取り出すローの数は多くても 1 つだけです。このタイプのクエリは SELECT 文に INTO 句を組み合わせて作成します。INTO 句は SELECT リストの後に続き、FROM 句より前に指定します。SELECT リストの各項目の値を受け取るための変数のリストが含まれます。変数は、SELECT リストの項目数と同じ数だけ用意します。

SELECT 文が実行されると、データベースサーバは SELECT 文の結果を取り出して、変数に入れます。クエリの結果、複数のローが取り出されると、データベースサーバはエラーを返します。複数のローを返すクエリにはカーソルを使用します。

クエリの結果、選択されたローが存在しない場合、変数は更新されず、警告が返されます。

SELECT 文を実行するには、オブジェクトに対する適切な SELECT 権限を持っている必要があります。

例 1：プロシージャを作成し、SELECT...INTO 文を使用してその結果を選択します。

1. Interactive SQL を起動して、SAP Sybase IQ サンプルデータベースに接続します。CREATE PROCEDURE システム権限、および Employee テーブルに対する SELECT 権限または SELECT ANY TABLE システム権限のいずれかを持っている必要があります。
2. [SQL 文] ウィンドウ枠で、次の文を実行し、従業員の平均給与を OUT パラメータとして返すプロシージャ (AverageSalary) を作成します。

プロシージャとバッチの作成

```
CREATE PROCEDURE AverageSalary( OUT average_salary NUMERIC(20,3) )
BEGIN
    SELECT AVG( Salary )
        INTO average_salary
        FROM GROUPO.Employees;
END;
```

3. プロシージャの結果を格納する変数を作成します。この場合、出力変数は小数点以下3桁の数値となります。

```
CREATE VARIABLE Average NUMERIC(20,3);
```

4. 作成した変数を使ってプロシージャを呼び出します。

```
CALL AverageSalary( Average );
```

5. プロシージャが正しく作成され、実行された場合、Interactive SQL の [メッセージ] タブにエラーは表示されません。
6. 次の文を実行して変数の値を検査します。

```
SELECT Average;
```

7. 出力変数 Average の値を見ます。[結果] ウィンドウ枠の [結果] タブに、この変数の値 49988.623 が表示されます。これが従業員の給与の平均値です。

例 2 : シングルロー SELECT 文の結果を返します。

1. Interactive SQL を起動して、SAP Sybase IQ サンプルデータベースに接続します。CREATE PROCEDURE システム権限、および Customers テーブルに対する SELECT 権限または SELECT ANY TABLE システム権限のいずれかを持っていることが必要です。
2. 指定した顧客によって行われた発注の数を返すには、次の文を実行します。

```
CREATE PROCEDURE OrderCount (
    IN customer_ID INT,
    OUT Orders INT )
BEGIN
    SELECT COUNT(SalesOrders.ID)
        INTO Orders
    FROM GROUPO.Customers
        KEY LEFT OUTER JOIN SalesOrders
        WHERE Customers.ID = customer_ID;
END;
```

3. このプロシージャは、Interactive SQL で次の文を使ってテストします。次の文は ID が 102 の顧客からの注文の回数を返します。

```
CREATE VARIABLE orders INT;
CALL OrderCount ( 102, orders );
SELECT orders;
```

例 2 の注意 –

- customer_ID パラメータは IN パラメータとして宣言されます。このパラメータは顧客の ID をプロシージャに渡します。

- Orders パラメータは OUT パラメータとして宣言されます。これは変数 orders の値を呼び出し元の環境に返します。
- 変数 Orders はプロシージャの引数リストで宣言されているので、DECLARE 文は必要ありません。
- SELECT 文は 1 つのローを返して、変数 Orders に入れます。

プロシージャからの結果セットで返される情報

プロシージャは、個別のパラメータとして呼び出しを行った環境に結果を返すだけでなく、結果セットとして情報を返すこともできます。通常、結果セットになるのはクエリの結果です。

RESULT 句の変数の数は、SELECT 文のリスト内の変数の数に一致しなければなりません。データ型が一致しない場合は、可能であれば自動的にデータ型の変換が行われます。

RESULT 句は CREATE PROCEDURE 文の一部であり、文デリミタは付きません。

SELECT 文のリスト内の変数の名前は、RESULT 句の変数の名前と一致する必要はありません。

ビューでプロシージャの結果セットを変更するには、基本となるテーブルに対する適切な権限がユーザに必要です。

ストアドプロシージャまたはユーザ定義関数が結果を返す場合、出力パラメータや戻り値もサポートされません。

デフォルトでは Interactive SQL は最初の結果セットのみを返します。Interactive SQL でプロシージャが結果の複数のローを返せるようにするには、[オプション] ウィンドウの [結果] タブで [複数の結果セットを表示] オプションを設定します。

例 1

次に示すプロシージャは、注文した顧客のリストと、注文の合計額を返します。

Interactive SQL で次の文を実行します。

```
CREATE PROCEDURE ListCustomerValue ()
RESULT ( "Company" CHAR(36), "Value" INT )
BEGIN
    SELECT CompanyName,
           CAST( SUM( SalesOrderItems.Quantity *
                    Products.UnitPrice )
                AS INTEGER ) AS value
    FROM Customers
       INNER JOIN SalesOrders
       INNER JOIN SalesOrderItems
       INNER JOIN Products
    GROUP BY CompanyName
    ORDER BY value DESC;
END;
```

プロシージャとバッチの作成

CALL ListCustomerValue (); を実行すると、次の結果セットが返されます。

Company	Value
The Hat Company	5016
The Igloo	3564
The Ultimate	3348
North Land Trading	3144
Molly's	2808
...	...

例 2

次に示すプロシージャは、ある部署の従業員 1 人 1 人の給与をセットにして返します。Interactive SQL で次の文を実行します。

```
CREATE PROCEDURE SalaryList( IN department_id INT )
RESULT ( "Employee ID" INT, Salary NUMERIC(20,3) )
BEGIN
    SELECT EmployeeID, Salary
    FROM Employees
    WHERE Employees.DepartmentID = department_id;
END;
```

RESULT 句での名前はクエリの結果と対応付けられ、表示される結果のカラムの見出しに使われます。

研究開発部 (部署 ID 100) の従業員の給与を表示するには、次の文を実行します。

```
CALL SalaryList( 100 );
```

[結果] ウィンドウ枠に次の結果セットが表示されます。

Employee ID	Salary
102	45700.000
105	62000.000
160	57490.000
243	72995.000
...	...

複数の結果セットを返す

Interactive SQL を使用して、プロシージャから複数の結果セットを返すことができます。

前提条件

この作業を実行するための前提条件はありません。

手順

デフォルトでは、Interactive SQL では、複数の結果セットは表示されません。

1. Interactive SQL で、データベースに接続します。
2. [ツール] » [オプション] をクリックします。
3. [SAP Sybase IQ] をクリックします。
4. [結果] タブで、[すべての結果セットを表示] をクリックします。
5. [OK] をクリックします。

このオプションを有効にすると、Interactive SQL には複数の結果セットが表示されます。設定内容はすぐに反映され、無効にされるまで後続のセッションすべてについて有効になります。

次のステップ

プロシージャ定義に含まれる RESULT 句を使う場合、結果セットはそれに合わせなければなりません。つまり、結果セットは SELECT 文のリストと同じ数の項目を持ち、データ型は RESULT 句にリストされたデータ型に自動的に変換されます。

RESULT 句を省略した場合は、プロシージャは返されるカラムの数や型がさまざまに異なる結果セットを返すことができます。

プロシージャの変数結果セット

RESULT 句は、プロシージャでは省略可能です。RESULT 句を省略すると、実行方法に応じて、さまざまなカラム数またはカラム型の、異なる結果セットを返すプロシージャを記述できます。

変数結果セット機能を使用しない場合は、性能を高めるために RESULT 句を使用してください。

たとえば、次のプロシージャは、変数として Y を入力した場合は 2 カラムを、それ以外の場合は 1 カラムを返します。

```
CREATE PROCEDURE Names( IN formal char(1) )  
BEGIN
```

プロシージャとバッチの作成

```
IF formal = 'y' THEN
  SELECT Surname, GivenName
  FROM Employees
ELSE
  SELECT GivenName
  FROM Employees
END IF
END;
```

クライアントアプリケーションで使用しているインタフェースによっては、プロシージャでの変数結果セットの使用に制限があります。

- **Embedded SQL** – 正しい形式の結果セットを取得するには、結果セットのカーソルが開かれてからローが返されるまでの間に、プロシージャコールを記述 (DESCRIBE) します。

RESULT 句を使用しないでプロシージャを作成し、そのプロシージャが変数結果セットを返す場合には、プロシージャを参照する SELECT 文の DESCRIBE が失敗することがあります。DESCRIBE の失敗を防ぐには、SELECT 文の FROM 句に WITH 句を含めることをおすすめします。または、DESCRIBE 文で WITH VARIABLE RESULT 句を使用できます。WITH VARIABLE RESULT 句は、各 OPEN 文の後にプロシージャ呼び出しを記述すべきかどうかを判断するのに使用できます。

- **ODBC** – 変数結果セットプロシージャは ODBC アプリケーションで使用できます。SAP Sybase IQ ODBC ドライバは、変数結果セットを正しく記述します。
- **Open Client アプリケーション** – Open Client アプリケーションは、変数結果セットプロシージャを使用できます。SAP Sybase IQ は、変数結果セットを正しく記述します。

エラーと警告の処理

アプリケーションが SQL 文を実行した後、ステータスコードをチェックできます。ステータスコード (リターンコード) は文が正しく実行されたかどうかを表示して、エラーの場合はその理由を提示します。プロシージャを呼び出す CALL 文にも同じメカニズムが使われます。

エラーのレポートには、SQLCODE か SQLSTATE のどちらかのステータス表示を使用します。

SQL 文が実行されると、SQLCODE と SQLSTATE と呼ばれる特別なプロシージャ変数に値が入ります。この特別値は、文の実行中に変わった状況が発生したかどうかを示します。SQLCODE と SQLSTATE の値は、IF 文を SQL 文の後に置いてチェックできます。その結果によって適切な動作が行われます。

たとえば、SQLSTATE 変数はローが正しくフェッチされたかどうかを示すのに使用できます。TopCustomerValue プロシージャには、SELECT 文中のすべてのローが処理されたかどうかを検知するために SQLSTATE テストが使われています。

デフォルトのエラー処理

この項では、プロシージャ内にエラー処理を指定しなかった場合に、SAP Sybase IQ がエラーを処理する方法を説明します。

さまざまな動作に例外ハンドラを使用できます。

警告の処理はエラーの処理とは少し異なります。

エラーを処理するには、特に指定しないかぎり次の 2 通りの方法があります。

- **デフォルトのエラー処理** – プロシージャかトリガがエラーを起こしたときに、呼び出しを行った環境にエラーコードが返されます。
- **ON EXCEPTION RESUME – CREATE PROCEDURE** 文に ON EXCEPTION RESUME 句が含まれていれば、プロシージャはエラーを起こした箇所の次の文から実行を再開します。

ON EXCEPTION RESUME を使用するプロシージャの正確な動作は、on_tsq_error オプション設定によって指定します。

デフォルトのエラー処理

通常、プロシージャまたはトリガの SQL 文がエラーを起こすと、そのプロシージャまたはトリガは実行を停止し、SQLCODE と SQLSTATE に適切な値が入った状態でアプリケーションに制御が戻されます。これは最初の文がエラーを起こしたときも同じです。また、トリガの場合、トリガを起動した操作も取り消され、エラーがアプリケーションに返されます。

次の例のプロシージャは、アプリケーションからプロシージャ OuterProc を呼び出し、OuterProc が InnerProc を呼び出して、そこでエラーが発生した場合の処理を示します。

```
CREATE PROCEDURE OuterProc()
BEGIN
    MESSAGE 'Hello from OuterProc.' TO CLIENT;
    CALL InnerProc();
    MESSAGE 'SQLSTATE set to ',
        SQLSTATE, ' in OuterProc.' TO CLIENT
END;
CREATE PROCEDURE InnerProc()
BEGIN
    DECLARE column_not_found
        EXCEPTION FOR SQLSTATE '52003';
    MESSAGE 'Hello from InnerProc.' TO CLIENT;
    SIGNAL column_not_found;
    MESSAGE 'SQLSTATE set to ',
        SQLSTATE, ' in InnerProc.' TO CLIENT;
```

プロシージャとバッチの作成

```
END;  
CALL OuterProc();
```

Interactive SQL の [メッセージ] タブに次のメッセージが表示されます。

```
Hello from OuterProc.  
Hello from InnerProc.
```

InnerProc 内の DECLARE 文は、サーバが認識しているエラー条件に関連して事前に定義された SQLSTATE 値のうち、1 つの記号名を宣言します。

MESSAGE 文は Interactive SQL の [メッセージ] タブにメッセージを送ります。

SIGNAL 文は InnerProc プロシージャ内から、エラーであることを外部に知らせる役割を持ちます。

InnerProc の SIGNAL 文の後の文は実行されず、InnerProc はすぐに呼び出しを行った環境 (この場合はプロシージャ OuterProc) に制御を戻します。OuterProc の CALL 文の後に続く文は実行されません。エラーは呼び出しを行った環境に戻され、処理されます。たとえば、Interactive SQL はエラーメッセージをメッセージウィンドウに表示してエラーの処理を行います。

TRACEBACK 関数はエラーが起きたときに実行していた文をリストします。

Interactive SQL から TRACEBACK を使うには、次の文を入力します。

```
SELECT TRACEBACK();
```

ON EXCEPTION RESUME を使ったエラー処理

ON EXCEPTION RESUME 文が CREATE PROCEDURE 文に含まれていた場合、エラーが起きると、次の文が検査されます。その文がエラーを処理する場合、プロシージャの実行が継続され、エラーが発生した文の次の文を実行します。エラーが発生したとき、呼び出しを行った環境に制御を戻しません。

on_tsq_error オプション設定を使用して、ON EXCEPTION RESUME を使用するプロシージャの動作を変更できます。

エラー処理文には、次のようなものがあります。

```
IF  
SELECT @variable =  
CASE  
LOOP  
LEAVE  
CONTINUE  
CALL  
EXECUTE  
SIGNAL
```

```
RESIGNAL
DECLARE
SET VARIABLE
```

次に示すプロシージャは、アプリケーションからプロシージャ OuterProc を呼び出し、OuterProc が InnerProc を呼び出して、そこでエラーが発生した場合の処理を示します。例文は、この項の最初で使用したプロシージャを基にしています。

```
DROP PROCEDURE OuterProc;
DROP PROCEDURE InnerProc;

CREATE PROCEDURE OuterProc()
ON EXCEPTION RESUME
BEGIN
    DECLARE res CHAR(5);
    MESSAGE 'Hello from OuterProc.' TO CLIENT;
    CALL InnerProc();
    SET res=SQLSTATE;
    IF res='52003' THEN
        MESSAGE 'SQLSTATE set to ',
            res, ' in OuterProc.' TO CLIENT;
    END IF
END;

CREATE PROCEDURE InnerProc()
ON EXCEPTION RESUME
BEGIN
    DECLARE column_not_found
        EXCEPTION FOR SQLSTATE '52003';
    MESSAGE 'Hello from InnerProc.' TO CLIENT;
    SIGNAL column_not_found;
    MESSAGE 'SQLSTATE set to ',
        SQLSTATE, ' in InnerProc.' TO CLIENT;
END;

CALL OuterProc();
```

Interactive SQL の [メッセージ] タブに、次のメッセージが表示されます。

```
Hello from OuterProc.
Hello from InnerProc.
SQLSTATE set to 52003 in OuterProc.
```

実行パスを次に示します。

1. OuterProc は InnerProc を実行して呼び出します。
2. InnerProc では、SIGNAL 文がエラーを通知します。
3. MESSAGE 文はエラー処理文ではないので、制御は OuterProc に返され、メッセージは表示されません。
4. OuterProc では、エラーに続く文が SQLSTATE の値を res という変数に割り当てます。これはエラー処理文なので、実行は継続され、OuterProc メッセージが表示されます。

デフォルトの警告処理

エラーと警告の処理方法は異なります。デフォルトのエラー処理は、SQLSTATE と SQLCODE に値を入れてエラー発生時の呼び出しを行った環境に制御を戻しますが、警告処理のデフォルトは、SQLSTATE と SQLCODE に値を入れてプロシージャの実行を続けます。

次に示す例は、デフォルトの警告処理を示します。

この場合、ローが見つからないことを示す条件が SIGNAL 文によって生成されます。これはエラーではなく警告です。

```
DROP PROCEDURE OuterProc;
DROP PROCEDURE InnerProc;

CREATE PROCEDURE OuterProc()
BEGIN
    MESSAGE 'Hello from OuterProc.' TO CLIENT;
    CALL InnerProc();
    MESSAGE 'SQLSTATE set to ',
        SQLSTATE, ' in OuterProc.' TO CLIENT;
END;
CREATE PROCEDURE InnerProc()
BEGIN
    DECLARE row_not_found
        EXCEPTION FOR SQLSTATE '02000';
    MESSAGE 'Hello from InnerProc.' TO CLIENT;
    SIGNAL row_not_found;
    MESSAGE 'SQLSTATE set to ',
        SQLSTATE, ' in InnerProc.' TO CLIENT;
END;

CALL OuterProc();
```

Interactive SQL の [メッセージ] タブに、次のメッセージが表示されます。

```
Hello from OuterProc.
Hello from InnerProc.
SQLSTATE set to 02000 in InnerProc.
SQLSTATE set to 00000 in OuterProc.
```

両方のプロシージャとも、警告によって SQLSTATE に値 (02000) が設定された後も実行を続けました。

InnerProc で 2 番目の MESSAGE 文を実行すると、警告がリセットされます。SQL 文は、SQLSTATE を 00000、SQLCODE を 0 にリセットします。プロシージャがエラー状態を保存する必要がある場合、エラーまたは警告の原因となった文の実行直後に値を割り当てる必要があります。

例外ハンドラ

エラーは呼び出しを行った環境へ戻すよりも、プロシージャまたはトリガの内部で捕捉して処理した方が良い場合があります。これは**例外ハンドラ**を使用しています。

例外ハンドラは、複合文の EXCEPTION 部分で定義します。

複合文でエラーが起きた場合、例外ハンドラが実行されます。警告では、例外ハンドラは実行されません。ネストされた複合文の中でエラーが起きた場合、また、複合文の中から起動されたプロシージャやトリガの中でエラーが起きた場合は、例外処理コードが実行されます。

中断エラー SQL_INTERRUPT、SQLSTATE 57014 の例外ハンドラには、ROLLBACK や ROLLBACK TO SAVEPOINT などの中断のできない文だけを含めません。例外ハンドラに、接続の中断時に呼び出される中断可能な文を含めると、データベースサーバは最初の中断可能な文で例外ハンドラを停止し、中断エラーを返します。

例外ハンドラでは、文が失敗した原因を判断するために、SQLSTATE または SQLCODE 特別値を使用できます。代わりに、ERRORMSG 関数を引数なしで使用すると、SQLSTATE に関連付けられたエラー条件が返されます。この情報を指定できるのは、各 WHEN 句の最初の文のみであり、文は複合文にはできません。

この例では、InnerProc プロシージャで見つからないカラムに関するエラーを処理するために、コードが追加されています。

```
DROP PROCEDURE OuterProc;
DROP PROCEDURE InnerProc;

CREATE PROCEDURE OuterProc()
BEGIN
    MESSAGE 'Hello from OuterProc.' TO CLIENT;
    CALL InnerProc();
    MESSAGE 'SQLSTATE set to ',
        SQLSTATE, ' in OuterProc.' TO CLIENT
END;
CREATE PROCEDURE InnerProc()
BEGIN
    DECLARE column_not_found
    EXCEPTION FOR SQLSTATE '52003';
    MESSAGE 'Hello from InnerProc.' TO CLIENT;
    SIGNAL column_not_found;
    MESSAGE 'Line following SIGNAL.' TO CLIENT;
    EXCEPTION
    WHEN column_not_found THEN
        MESSAGE 'Column not found handling.' TO CLIENT;
    WHEN OTHERS THEN
        RESIGNAL ;
END;
```

```
CALL OuterProc();
```

Interactive SQL の [メッセージ] タブに、次のメッセージが表示されます。

```
Hello from OuterProc.  
Hello from InnerProc.  
Column not found handling.  
SQLSTATE set to 00000 in OuterProc.
```

EXCEPTION 句は例外ハンドラを宣言します。これ以降の文はエラーが起きないかぎり実行されません。WHEN 句は例外名 (DECLARE 文で宣言) と、その例外が起こったときに実行する文を定義します。WHEN OTHERS THEN 句はその前の WHEN 句以外で例外が起こったときに実行する文を定義します。

上の例では、RESIGNAL は例外を上位レベルの例外ハンドラに渡します。WHEN OTHERS THEN が例外ハンドラ中に定義されていない場合は、RESIGNAL がデフォルト処理になります。

追加の注意事項

- InnerProc の SIGNAL 文に続く行ではなく、EXCEPTION ハンドラが実行されます。
- 見つからないカラムに関するエラーが発生したため、エラー処理のための MESSAGE 文が実行され、SQLSTATE は 0 にリセットされます (エラーがないことを示します)。
- 例外処理コードが実行された後、制御は OuterProc に戻され、OuterProc はエラーがなかったかのように前へ進みます。
- ON EXCEPTION RESUME は指定した例外処理とは一緒に使えません。ON EXCEPTION RESUME が含まれていると、例外処理コードは実行されません。
- エラーに対するエラー処理コードが RESIGNAL 文である場合、制御は OuterProc プロシージャに戻され、SQLSTATE の値は 52003 に設定されたままになります。これは、InnerProc にはエラー処理コードがないのと同じです。OuterProc にはこれ以外のエラー処理コードはないため、プロシージャはエラーになります。

例外処理とアトミックな複合文

アトミック複合内でエラーが発生し、その文にエラー処理用の例外ハンドラが実装されている場合は、複合文はアクティブな例外なしで完了し、例外より前の変更は取り消されません。例外ハンドラが発生したエラーを処理しない場合、または別のエラー (RESIGNAL によるエラーを含む) を発生させた場合は、そのアトミックな複合文で行われた変更は取り消されます。

ネストされた複合文と例外処理

エラーを引き起こした文に続くコードは、プロシージャ定義に ON EXCEPTION RESUME 句が含まれる場合のみ、実行されます。

ネストされた複合文を使用すると、エラーの後にどの文が実行され、どの文が実行されないのかを制御できます。

次の例は、ネストされた複合文をどのように使用してフローを制御するかを示します。

```
DROP PROCEDURE OuterProc;
DROP PROCEDURE InnerProc;

CREATE PROCEDURE InnerProc()
BEGIN
  BEGIN
    DECLARE column_not_found
    EXCEPTION FOR SQLSTATE VALUE '52003';
    MESSAGE 'Hello from InnerProc' TO CLIENT;
    SIGNAL column_not_found;
    MESSAGE 'Line following SIGNAL' TO CLIENT
    EXCEPTION
      WHEN column_not_found THEN
        MESSAGE 'Column not found handling' TO
        CLIENT;
      WHEN OTHERS THEN
        RESIGNAL;
  END;
  MESSAGE 'Outer compound statement' TO CLIENT;
END;

CALL InnerProc();
```

Interactive SQL の [メッセージ] タブに、次のメッセージが表示されます。

```
Hello from InnerProc
Column not found handling
Outer compound statement
```

エラーを引き起こした SIGNAL 文が検出されると、制御は複合文の例外ハンドラに渡されて、「カラムが見つかりません。」というメッセージが出力されます。次に制御は外部複合文に渡され、「Outer compound statement」メッセージが出力されます。

内部複合文でカラムが見つかりません。(SQLSTATE) 以外のエラーが検出されると、例外ハンドラは RESIGNAL 文を実行します。RESIGNAL 文は、呼び出しを行った環境に制御を直接戻します。外部複合文の残りの文は実行されません。

例

この例は、RESIGNAL による sa_error_stack_trace システムプロシージャの出力を示しています。

```
CREATE PROCEDURE DBA.error_reporting_procedure()
BEGIN
    SELECT *
    FROM sa_error_stack_trace();
END;

CREATE PROCEDURE DBA.proc1()
BEGIN TRY
    BEGIN TRY
        DECLARE v INTEGER = 0;
        SET v = 1 / v;
    END TRY
    BEGIN CATCH
        CALL DBA.proc2();
    END CATCH
END TRY
BEGIN CATCH
    CALL DBA.error_reporting_procedure();
END CATCH;

CREATE PROCEDURE DBA.proc2()
BEGIN
    CALL DBA.proc3();
END;

CREATE PROCEDURE DBA.proc3()
BEGIN
    RESIGNAL;
END;
```

CALL proc1() を使用して上記のプロシージャが呼び出されると、次の結果セットが作成されます。

StackLevel	UserName	ProcName	LineNumber	IsResignal
1	DBA	proc1	8	0
2	DBA	proc2	3	0
3	DBA	proc3	3	1
4	DBA	proc1	5	0

この例は、RESIGNAL と BEGIN 文による sa_error_stack_trace システムプロシージャの出力を示しています。

```
CREATE PROCEDURE DBA.error_reporting_procedure()
BEGIN
    SELECT *
    FROM sa_error_stack_trace();
```

```

END;

CREATE PROCEDURE DBA.proc1()
BEGIN
  BEGIN
    DECLARE v INTEGER = 0;
    SET v = 1 / v;
    EXCEPTION WHEN OTHERS THEN
      CALL DBA.proc2();
  END
END
EXCEPTION WHEN OTHERS THEN
  CALL DBA.error_reporting_procedure();
END;

CREATE PROCEDURE DBA.proc2()
BEGIN
  CALL DBA.proc3();
END;

CREATE PROCEDURE DBA.proc3()
BEGIN
  RESIGNAL;
END;

```

CALL proc1() を使用して上記のプロシージャが呼び出されると、次の結果セットが作成されます。

StackLevel	UserName	ProcName	LineNumber	IsResignal
1	DBA	proc1	8	0
2	DBA	proc2	3	0
3	DBA	proc3	3	1
4	DBA	proc1	5	0

プロシージャでのトランザクションとセーブポイント

プロシージャ内の SQL 文は現在のトランザクションの一部です。1つのトランザクション内で複数のプロシージャを呼び出すことや、1つのプロシージャ内に複数のトランザクションを持つことができます。

アトミックな文中では COMMIT と ROLLBACK は許可されません。

プロシージャではセーブポイントを使用できますが、ROLLBACK TO SAVEPOINT 文はアトミックオペレーションが開始される以前のセーブポイントを参照することはできません。また、アトミックオペレーション内のすべてのセーブポイントは、その操作が終了したときに解除されます。

プロシージャ、ファンクション、トリガ、イベント、またはビューの内容を隠す

プロシージャ、ファンクション、トリガ、およびビュー内に含まれるロジックを公開しないで、アプリケーションとデータベースを配布する場合は、ALTER PROCEDURE、ALTER FUNCTION、ALTER TRIGGER、ALTER EVENT、ALTER VIEW 文の SET HIDDEN 句を使用して、これらのオブジェクトの内容を隠すことができます。

前提条件

オブジェクトの所有者であるか、ALTER ANY OBJECT システム権限を持っているか、次の権限の 1 つを持っていることが必要です。

プロシージャとファンクション – ALTER ANY PROCEDURE システム権限

ビュー – ALTER ANY VIEW システム権限

イベント – MANAGE ANY EVENT システム権限

トリガ –

ALTER ANY TRIGGER システム権限

基礎となるテーブルに対する ALTER 権限と CREATE ANY OBJECT システム権限

ビューに対するトリガについては、ALTER ANY TRIGGER と ALTER ANY VIEW のシステム権限を持っていることが必要です。

手順

SET HIDDEN 句は、関連オブジェクトを使用可能な状態に保ちながら、その内容を難読化して読み取れないようにします。また、アンロードして、別のデータベースに再ロードすることもできます。

修正を元に戻すことはできません。修正すると、オブジェクトの元のテキストが削除されます。オブジェクトの元のソースをデータベースの外部に保存しておく必要があります。

注意： preserve_source_format データベースオプションを On に設定すると、データベースサーバは、プロシージャ、ビュー、トリガ、イベントの CREATE 文と ALTER 文によってフォーマットされたソースを保存し、それを適切なシステムビューのソースカラムに配置します。この場合、オブジェクト定義とソース定義の両方が隠されます。

ただし、`preserve_source_format` データベースオプションを On に設定しても、`SET HIDDEN` 句でオブジェクトの元のソース定義を削除できます。

`SET HIDDEN` 句を持つ適切な `ALTER` 文を使用します。

オプション	アクション
個別のオブジェクトを隠す	1つのプロシージャ、ファンクション、トリガ、イベント、またはビューを隠すには、 <code>SET HIDDEN</code> 句を持つ適切な <code>ALTER</code> 文を実行します。
特定のタイプのすべてのオブジェクトを隠す	すべてのプロシージャ、ファンクション、トリガ、イベント、またはビューを隠すには、 <code>SET HIDDEN</code> 句を持つ適切な <code>ALTER</code> 文をループで実行します。

オートコミットが実行されます。オブジェクト定義は表示されません。ただし、オブジェクトは直接参照でき、クエリ処理中に使用できることは変わりません。

プロシージャ、トリガ、イベント、バッチで使用できる文

バッチにはほとんどの SQL 文を使用できますが、次の文は使用できません。

```
ALTER DATABASE (構文 3 および 4)
CONNECT
CREATE DATABASE
CREATE DECRYPTED FILE
CREATE ENCRYPTED FILE
DISCONNECT
DROP CONNECTION
DROP DATABASE
FORWARD TO
INPUT、OUTPUT などの Interactive SQL 文
PREPARE TO COMMIT
STOP SERVER
```

`COMMIT`、`ROLLBACK`、`SAVEPOINT` 文はプロシージャ、トリガ、バッチで使用できますが、若干の制限があります。

バッチで使用される **SELECT** 文

バッチには 1 つまたは複数の `SELECT` 文を含めることができます。次に例を示します。

```
IF EXISTS ( SELECT *
            FROM SYSTAB
```

プロシージャとバッチの作成

```
WHERE table_name='Employees' )
THEN
  SELECT    Surname AS LastName,
           GivenName AS FirstName
  FROM Employees;
  SELECT Surname, GivenName
  FROM Customers;
  SELECT Surname, GivenName
  FROM Contacts;
END IF;
```

結果セットのエイリアスは、最初の SELECT 文でのみ必要です。サーバはバッチ中の最初の SELECT 文を結果セットの記述に使用するからです。

各クエリの後には、次の結果セットを取り出すための RESUME 文が必要です。

プロシージャ、トリガ、ユーザ定義関数、バッチで使用される EXECUTE IMMEDIATE

EXECUTE IMMEDIATE 文を使うと、文字列(引用符で囲む)と変数を使って文を組み立てることができます。次に示すのは、テーブルを作成する EXECUTE IMMEDIATE 文を含むプロシージャの例です。

```
CREATE PROCEDURE CreateTableProcedure(
  IN tablename CHAR(128) )
BEGIN
  EXECUTE IMMEDIATE 'CREATE TABLE '
  || tablename
  || '( column1 INT PRIMARY KEY )'
END;
```

EXECUTE IMMEDIATE 文は、結果セットを返すクエリで使用できます。文が結果セットを返すように指定するために、EXECUTE IMMEDIATE 文で WITH RESULT SET ON 句を使用します。デフォルトの動作では、文は結果セットを返しません。WITH RESULT SET ON または WITH RESULT SET OFF を指定することは、プロシージャが作成されるときだけではなく、プロシージャが実行されるときの動作についても影響をもたらすことになります。

次のプロシージャを考えてみます。

```
CREATE OR REPLACE PROCEDURE test_result_clause()
BEGIN
  EXECUTE IMMEDIATE WITH RESULT SET OFF 'SELECT 1';
END;
```

プロシージャ定義に RESULT SET 句が含まれていないと、データベースサーバ側ではプロシージャによって結果セットが生成されるのかどうかを判断しようとしてしまいます。ここでは、EXECUTE IMMEDIATE 文では、結果セットを生成しないように設定されています。そのため、データベースサーバ側では、プロシージャには結果セットのカラムがないと判断されるため、このプロシージャの SYSPROCPARM システムビューにはカラムが表示されません。このプロシージャの呼び出しに

DESCRIBE を指定しても、結果カラムは返されません。カーソルを開いたり、文を実行するかどうかを決定するために、Embedded SQL アプリケーションによってこの情報が使用された場合には、文が実行されて、エラーが返されます。

2 番目の例では、上記のプロシージャの修正バージョンを考えてみます。

```
CREATE OR REPLACE PROCEDURE test_result_clause()
BEGIN
    EXECUTE IMMEDIATE WITH RESULT SET ON 'SELECT 1';
END;
```

ここでは、WITH RESULT SET ON 句が指定されているため、SYSPROCPARM システムビューにはこのプロシージャのローが存在しています。プロシージャが EXECUTE IMMEDIATE を使用しているため、データベースサーバ側では、実際に結果セットがどのようなようになるかについては把握できません。ただし、結果セットが存在することは予測できるため、データベースサーバは SYSPROCPARM に、expression という名前の SMALLINT 型のダミーの結果セットのカラムを定義することによって、結果セットに対応していることを示します。作成される結果セット用ダミーカラムが 1 つだけです。サーバ側では、EXECUTE IMMEDIATE 文が使用されているときに、各結果セットのカラム数や型については判断できません。そのため、さらに少し修正したバージョンの例を考えてみます。

```
CREATE OR REPLACE PROCEDURE test_result_clause()
BEGIN
    EXECUTE IMMEDIATE WITH RESULT SET ON 'SELECT 1, 2, 3';
END;
```

ここでは、SELECT 文は 3 つのカラムの結果セットを返しますが、サーバの SYSPROCPARM システムビューには、1 つのローがあるだけです。そこで、次のクエリを見てください。

```
SELECT * FROM test_result_clause();
```

このクエリは、SQLCODE -866 エラーになります。これは、実行時の結果セットの定義が SYSPROCPARM に事前に用意された定義と一致しないためです。

上記のクエリを実行するには、次に示すように、結果セットのカラムの名前と型を明示的に指定します。

```
SELECT * FROM test_result_clause() WITH (x INTEGER, y INTEGER, z
INTEGER);
```

WITH RESULT SET ON が指定されている場合は、実行時に、データベースサーバは EXECUTE IMMEDIATE 文を処理し、結果セットを返します。ただし、WITH RESULT SET OFF が指定されているか、または句が省略されている場合でも、データベースサーバは解析された文字列引数の最初の文の型については、同じように認識することができます。その文が SELECT 文であれば、結果セットを返します。それでは、前述した 2 番目の例を見てください。

```
CREATE OR REPLACE PROCEDURE test_result_clause()
BEGIN
```

プロシージャとバッチの作成

```
EXECUTE IMMEDIATE WITH RESULT SET OFF 'SELECT 1';  
END;
```

Interactive SQL からこのプロシージャを呼び出すとエラーにはなりません。ここで、1つの SELECT 文ではなく、バッチが含まれたプロシージャに変更するとします。

```
CREATE OR REPLACE PROCEDURE test_result_clause()  
BEGIN  
    EXECUTE IMMEDIATE WITH RESULT SET OFF  
        'begin declare v int; set v=1; select v; end';  
END;
```

この場合には、test_result_clause プロシージャを呼び出すと、エラー (SQLCODE -946, SQLSTATE 09W03) が発生します。

次の最後の例では、プロシージャ内で SELECT 文を EXECUTE IMMEDIATE 文の引数として構築して、プロシージャが結果セットを返すようにする方法を示しています。

```
CREATE PROCEDURE DynamicResult (  
    IN Columns LONG VARCHAR,  
    IN TableName CHAR(128),  
    IN Restriction LONG VARCHAR DEFAULT NULL )  
BEGIN  
    DECLARE Command LONG VARCHAR;  
    SET Command = 'SELECT ' || Columns || ' FROM ' || TableName;  
    IF ISNULL( Restriction, '' ) <> '' THEN  
        SET Command = Command || ' WHERE ' || Restriction;  
    END IF;  
    EXECUTE IMMEDIATE WITH RESULT SET ON Command;  
END;
```

このプロシージャは次のように呼び出されます。

```
CALL DynamicResult(  
    'table_id,table_name',  
    'SYSTAB',  
    'table_id <= 10');
```

これは次のような結果になります。

table_id	table_name
1	ISYSTAB
2	ISYSTABCOL
3	ISYSIDX
...	...

上記の CALL では、プロシージャで EXECUTE IMMEDIATE が使用されていても、正しく結果セットが返されます。ODBC などの一部のサーバ API では、PREPARE-DESCRIBE-EXECUTE-OR-OPEN を利用して要求が結合されており、結果セットを

返すかどうかによって、文を実行するか、または開くかが選択されます。文が開かれる必要がある場合には、API またはアプリケーションは DESCRIBE CURSOR を続けて発行することによって、プロシージャが作成されたときに構築された SYSPROCPARM システムビューの内容に依存するのではなく、実際の結果セットを把握できるようにします。この方法は、DBISQL でも DBISQLC でも使用されています。この場合、上記のプロシージャの CALL はエラーにならずに実行されます。ただし、文の DESCRIBE の結果に依存するアプリケーションインタフェースの場合は、どの文でも処理できるというわけではありません。

アトミックな複合文中では、COMMIT を行う EXECUTE IMMEDIATE 文は使えません。COMMIT 文はこのコンテキストでは許可されていません。

スケジュールとイベントの使用によるタスクの自動化

スケジューリング機能とイベント処理機能を使用して、データベース管理やその他のタスクを自動化します。

スケジュールとイベントの使用によるタスクの自動化

データベース管理タスクの多くは、体系的に実行すると効果的です。たとえば、定期的なバックアップ手順はデータベース管理手順の重要な部分です。

データベースにイベントを追加し、イベントのスケジュールを設定することによって SAP Sybase IQ のルーチンタスクを自動化できます。スケジュールに設定されている時刻になると、いつでも「イベントハンドラ」と呼ばれる一連のアクションがデータベースサーバによって実行されます。

また、データベース管理では、ある状態が発生したときにアクションを実行することも必要です。たとえば、トランザクションログが格納されているディスクの空き領域が少なくなってきたときには、適切な処置を行うよう、システム管理者に電子メールで通知することが考えられます。これらのタスクも各システムイベントに対してイベントハンドラを定義することによって自動化できます。

イベント

データベースにイベントを追加し、イベントのスケジュールを設定することによって SAP Sybase IQ のルーチンタスクを自動化できます。SAP Sybase IQ では、次のタイプのイベントがサポートされています。

- **スケジュールされたイベント** – スケジュールに関連付けられており、指定の時間に実行されます。
- **システムイベント** – データベースサーバによって追跡される特定のタイプの条件に関連付けられています。
- **手動イベント** – TRIGGER EVENT 文を使用して明示的に起動されます。
- **ユーザトレースイベント** – アプリケーションに関する情報をイベントトレースセッションに記録するのに使用されます。これらのイベントは、データベースへの接続のすべてから参照できます。

各イベントハンドラの実行後、エラーが発生しない場合、COMMIT が発生します。エラーが発生した場合は、ROLLBACK が発生します。

スケジュール

アクティビティをスケジュールすると、事前に設定した時刻に一連のアクションを確実に実行できます。スケジュール情報とイベントハンドラはいずれもデータベース自体に格納されます。

通常は必要ありませんが、1つの名前付きイベントに2つ以上のスケジュールを関連付けることによって複雑なスケジュールを定義できます。たとえば、営業時間が曜日によって変わるような販売店で、営業時間中1時間に1回イベントを発生させることができます。それぞれ別のスケジュールを持つ複数のイベントを定義して、共通のストアードプロシージャを呼び出すことで同じような効果が得られません。

イベントをスケジュールするとき、英語の曜日をフルネーム (Monday、Tuesday など) で使用することも、省略形 (Mon、Tue など) で使用することもできます。英語以外の言語で稼働するサーバで曜日名を認識する必要がある場合は、フルネームの英語の曜日を使用する必要があります。

次に便利なスケジュールの例を示します。

例

終業時に受注の要約を作成します。この例では `iqdemo.db` を使用しています。

```
CREATE TABLE OrderSummary(c1 date, c2 int);
CREATE EVENT SummarizeSchedule
START TIME '6:00 pm' on ('Mon', 'Tue', 'Wed', 'Thu', 'Fri')
HANDLER
BEGIN
    INSERT INTO DBA.OrderSummary
        SELECT MAX(OrderDate),
        COUNT(*)
    FROM GROUP0.SalesOrders
    WHERE OrderDate = current date
END;
```

スケジュール定義

柔軟に設定を可能にするために、スケジュール定義にはいくつかの構成要素が用意されています。

- **名前** – 各スケジュール定義には名前があります。2つ以上のスケジュールを1つのイベントに割り当てることができます。これは複雑なスケジュールを設計するのに便利です。
- **起動時刻** – イベントの起動時刻を定義できます。これは、イベントの実行が開始する時刻です。

- **範囲** – 起動時刻の代わりとして、イベントがアクティブになる時刻の範囲を指定できます。イベントは、指定した起動時刻と終了時刻の間に発生します。頻度は指定した周期で決定します。
- **周期** – 各スケジュールには周期を定義できます。イベントは、何日または何曜日ごとの何時何分何秒ごとという形で指定できる頻度でトリガされます。反復イベントには、EVERY または ON 句が含まれています。

システムイベント

SAP Sybase IQ は、いくつかのシステムイベントを追跡します。各システムイベントが提供するフックに一連のアクションをハングすることができます。データベースサーバはイベントを追跡し、システムイベントが定義された**トリガ条件**を満たしたときに (イベントハンドラに定義された) アクションを実行します。

イベントハンドラを定義して、選択したシステムイベントタイプが発生し、定義したトリガ条件を満たしたときに実行されるようにします。このようにしておくことで、データのセキュリティが向上し、管理が容易になります。イベントハンドラの動作は、実行中にエラーが検出されなければコミットされ、エラーが検出された場合はロールバックされます。

システムイベントタイプ

- **BackupEnd** – BackupEnd イベントタイプを使用すると、バックアップ終了時にアクションを実行できます。
- **接続イベント** – 接続が確立されたとき (Connect) または接続できなかったとき (ConnectFailed)。これらのイベントはセキュリティの目的で使用できます。イベントハンドラに接続する代わりに、ログインプロシージャを使用することもできます。
- **DatabaseStart** – DatabaseStart イベントタイプを使用すると、データベース起動時にアクションを実行できます。
- **Deadlock** – Deadlock イベントを使用すると、デッドロック発生時にアクションを実行できます。イベントハンドラでは、sa_report_deadlocks プロシージャを使用して、デッドロックが発生するに至った状況に関する情報を取得できます。Deadlock イベントを使用するときは、データベースサーバがデッドロック情報を取得するように設定する必要があります。これを行うには、log_deadlocks オプションを On にし、sa_server_option または -zl サーバオプションを使用して RememberLastStatement 機能を有効にします。

Deadlock イベントは、接続デッドロックとスレッドデッドロックの発生時に起動します。デッドロックイベントは、sa_report_deadlocks システムプロシージャによって取得できるもの以上の情報は提供しません。しかし、このイベントを使用すると、デッドロックにすぐに対処できます。データベースサーバに

保持されるデッドロック関連の情報は量が限られているため、迅速な対応が重要なこともあります。

- **Disconnect** – Disconnect イベントを使用すると、ユーザまたはアプリケーションの切断時にアクションを実行できます。
- **ディスクの空き領域** – データベースファイル (DBDiskSpace)、ログファイル (LogDiskSpace)、テンポラリファイル (TempDiskSpace) を格納しているデバイスの使用可能なディスク領域を追跡します。このシステムイベントは、Windows Mobile では使用できません。

ディスク領域イベントを使用すると、ディスク領域が不足したときに管理者に警告することができます。

データベースサーバの起動時に -fc オプションを指定して、データベースサーバでファイルシステムがいっぱいになった場合のコールバック関数を実装できます。

- **ファイルサイズ** – ファイルが指定したサイズに達したとき。これはデータベースファイル (GrowDB)、トランザクションログ (GrowLog)、テンポラリファイル (GrowTemp) に使用できます。

ファイルサイズイベントを使用すると、データベース上での異常なアクションを追跡したり、バルクオペレーションをモニタすることができます。

- **GlobalAutoincrement** – GLOBAL AUTOINCREMENT で定義されたカラムの残りの値がこの範囲の 1% を下回ると、GlobalAutoincrement イベントが起動します。これは、このイベント用のパラメータとして指定された残りの値のテーブルと数字に基づいて、global_database_id オプション用の新しい値を要求するのに使用できます。このイベントにおけるテーブルの残りの値を取得するには、EVENT_PARAMETER 関数を使用し、RemainingValues パラメータと TableName パラメータを指定します。RemainingValues は、そのカラム用に生成できる残りの値の数を返します。TableName は、範囲の終わりに近づいている GLOBAL AUTOINCREMENT カラムがあるテーブルを返します。
- **RAISERROR エラー** – RAISERROR イベントタイプを使用すると、RAISERROR 文が実行されたときにアクションを実行できます。RAISERROR 文で使用するエラー番号は、EVENT_CONDITION 関数 (たとえば、EVENT_CONDITION ('ErrorNumber')) を使用してイベントハンドラ内に定義できます。
- **アイドル時間** – データベースサーバが指定した時間アイドル状態にあったとき (ServerIdle)。このイベントタイプを使用すると、定型の管理操作をアクセスの少ない時間に行えます。

イベントのトリガ条件

各イベント定義には対応するシステムイベントがあります。また、イベント定義は1つまたは複数のトリガ条件を持ちます。システムイベントに対するトリガ条件が満たされるとイベントハンドラがトリガされます。

トリガ条件は CREATE EVENT 文の WHERE 句に含まれていて、AND キーワードを使用して結合できます。各トリガ条件は次のフォームで定義します。

```
event_condition( condition-name ) comparison-operator value
```

condition-name 引数は、さまざまなイベントタイプに対応できるようにあらかじめ設定されている文字列から1つを選択します。たとえば、DBSize(メガバイト単位のデータベースファイルサイズ)を使用して GrowDB システムイベントに適したトリガ条件を構築することができます。データベースサーバは、条件名とイベントタイプの対応をチェックしません。イベントタイプのコンテキストで条件に意味があるかどうかを確認する必要があります。

例

- データベースに侵入しようとする者を発見すると、管理者に通知します。この例では iqdemo.db を使用しています。

```
create event SecurityCheck
type ConnectFailed
handler
begin
declare num_failures int;
declare mins int;
insert into FailedConnections( log_time )
values ( current timestamp );
select count( * ) into num_failures
from FailedConnections
where log_time >= dateadd(minute, -5, current timestamp );
if( num_failures >= 3 ) then
  select datediff( minute, last_notification, current timestamp )
into mins from Notification;
  if( mins > 30 ) then
    update Notification
    set last_notification = current timestamp;
  end if
end if
end;
```

イベントハンドラ

イベントハンドラは、イベントをトリガするアクションとは別の接続上で実行されます。そのため、クライアントアプリケーションに影響することはありません。イベントハンドラは、イベントの作成者の権限で実行されます。

イベントハンドラは、スケジュールされたイベント用か、システムイベント処理用かにかかわらず、複合文を含んでいて、多くの点でストアドプロシージャに似ています。ループや条件付き実行などを追加することができます。

各イベントハンドラの実行後、エラーが発生しない場合、COMMIT が発生します。エラーが発生した場合は、ROLLBACK が発生します。

イベントハンドラのためのコンテキスト情報

ストアドプロシージャとは異なり、イベントハンドラには引数がありません。EVENT_PARAMETER 関数を使用して、イベントがトリガされたコンテキストに関する情報にアクセスできます。返される情報には、イベントがトリガされた接続 ID とユーザ ID、イベント名、実行回数が含まれます。

イベントハンドラのテスト

開発中は、好きなときにイベントハンドラをトリガできた方が便利です。TRIGGER EVENT 文を使うと、トリガ条件やスケジュールした時刻に関係なく、明示的にイベントを実行できます。ただし、無効なイベントハンドラを TRIGGER EVENT によって実行することはできません。

運用データベース上でイベントハンドラを開発するのはよいことではありませんが、明示的に ALTER EVENT 文を使ってイベントハンドラを無効にすることができます。

コードの共有

複数のイベントを処理するアクションを1つにまとめておくと便利です。たとえば、データベースファイルまたはログファイルを格納しているデバイスのディスク領域が少なくなってきたときに、通知アクションを実行することができます。これを実行するには、ストアドプロシージャを作成し、各イベントハンドラの本文から呼び出します。このとき、必要なコンテキスト情報をパラメータとしてプロシージャに渡します。

イベントハンドラのデバッグ

イベントハンドラのデバッグは、ストアドプロシージャのデバッグによく似ています。イベントハンドラは、イベントリストに表示されます。

イベントハンドラを隠す

ALTER EVENT 文と SET HIDDEN 句を使用して、イベントハンドラの定義を隠すことができます。SET HIDDEN 句を指定すると、ISYSEVENT システムテーブルの

action カラムに格納されているイベントハンドラの定義が永続的に難読化されません。

アクティブなイベントの制限

また、EVENT_PARAMETER 関数と NumActive コンテキスト名を使用して、現在アクティブになっている特定のイベントハンドラのインスタンスの数を判断できます。この関数は、一定時間に1つのイベントハンドラで1つのインスタンスだけを実行させるように制限する場合に利用できます。

データベースサーバによるシステムイベントのチェック

システムイベントは、CREATE EVENT 文の中で指定されるイベントタイプによって分類されます。イベントタイプには2つの種類があります。

- **アクティブイベントタイプ** – イベントタイプには、データベースサーバ自体のアクションの結果であるものがあります。こうしたアクティブなイベントタイプには、データベースファイルサイズ、さまざまなデータベースアクションの開始時、終了時 (BackupEnd など)、RAISERROR などが含まれます。

データベースサーバは、アクションを実行するときに、WHERE 句に定義されたトリガ条件が満たされているかどうかをチェックし、条件が満たされていればイベントタイプに対して定義されたイベントをトリガします。

- **ポーリングイベントタイプ** – ディスクの空き領域 (DBDiskSpace など) や IdleTime などのイベントタイプは、データベースのアクションだけでトリガされません。

このタイプのイベントに対して、データベースサーバは 30 秒ごとにポーリングします。ポーリングはデータベースの開始後、約 30 秒後から開始されます。

IdleTime イベントタイプの場合、データベースサーバはサーバが 30 秒間アイドル状態にあったかどうかをチェックします。その間まったく要求が開始されず、現在アクティブな要求もなければ、秒単位のアイドルチェック間隔時間をアイドル時間の合計に追加します。そうでない場合はアイドル時間の合計が 0 にリセットされます。したがって、IdleTime の値は、常に 30 秒の倍数になります。IdleTime がトリガ条件に指定した間隔より長くなると、IdleTime に関連付けられたイベントハンドラが起動します。

データベースサーバによるスケジュールされたイベントのチェック

イベントのスケジュール時刻の計算は、データベースサーバの起動時と、スケジュールされた各イベントハンドラの完了時に行われます。

次のスケジュール時刻の計算は、スケジュール定義に指定された増分に基づいて、増分を前回の起動時刻に追加することで行われます。指定した増分よりイベントハンドラの実行時間が長くなり、現在の処理が終わらないうちに次のスケジュール時刻が来る場合、データベースサーバは、現在の処理の後に次のスケジュール時刻がくるように増分します。

スケジュールとイベントの使用によるタスクの自動化

たとえば、実行に 65 分かかるイベントハンドラが 9 時 ~ 5 時の間の 1 時間ごとに起動するように要求された場合、実際には 9 時、11 時、1 時と 2 時間ごとに実行されます。

次回の実行まで待機時間を設ける処理を 9 時 ~ 5 時の間で実行するには、各実行の合間に `WAITFOR` 文を使って、指定した完了時間が経過するまでループするようにハンドラを定義できます。

データベースサーバを断続的に実行していて、スケジュール時刻にデータベースサーバが実行中でない場合、イベントハンドラが起動時に実行されることはありません。その代わりに、次のスケジュール時刻は起動時に計算されます。たとえば、毎晩 1 時にバックアップを実行するようにスケジュールしていても、終業時にはいつもデータベースサーバを停止している場合、バックアップが実行されることはありません。

次にスケジュールされているイベント実行が 1 時間以上後の場合、データベースサーバは時間単位で次のスケジュール時間を計算します。これにより、夏時間の開始または終了のためにシステムクロックが調整されたときに、イベントが予定どおりに起動されます。

イベントハンドラの実行

イベントハンドラがトリガされると、一時的に内部接続が確立され、その上でイベントハンドラが実行されます。ハンドラは、そのハンドラがトリガされるに至った接続で実行されるわけではありません。このため、クライアントアプリケーションとの対話で使用される `MESSAGE...TO CLIENT` などの文は、イベントハンドラ内では意味を持ちません。同様に、結果セットを返す文は使用できません。

ハンドラが実行される一時的な接続は、ライセンス契約の接続制限には数えられません。`login_procedure` に指定したプロシージャはイベント接続では実行されません。

イベントハンドラは、イベント所有者の権限に基づいて別の接続上で実行されます。また、イベントハンドラからプロシージャを呼び出すこともできます。その場合、プロシージャは、その所有者の権限を使用して実行されます。イベントハンドラが実行される接続は、パーソナルデータベースサーバの最大接続数である 10 には含められません。

イベントエラーが発生すると、データベースサーバメッセージログに記録されません。

注意： イベントハンドラ内のトランザクションは、実行中にエラーが検出されなかった場合はコミットされ、エラーが検出された場合はロールバックされます。

イベントハンドラを隠す

セキュリティ向上のために、ALTER EVENT 文を使用してイベントハンドラの定義を隠すことができます。

前提条件

MANAGE ANY EVENT または ALTER ANY OBJECT のどちらかのシステム権限が必要です。

手順

1. データベースに接続します。
2. SET HIDDEN 句を使用して ALTER EVENT 文を実行します。

```
ALTER EVENT event-name SET HIDDEN
```

イベントハンドラが、ISYSEVENT システムテーブルのアクションカラムに格納されているイベントハンドラ定義の中で永続的に難読化されます。

イベントハンドラ

イベントハンドラは、イベントをトリガするアクションとは別の接続上で実行されます。そのため、クライアントアプリケーションに影響することはありません。イベントハンドラは、イベントの作成者の権限で実行されます。

イベントハンドラは、スケジュールされたイベント用か、システムイベント処理用かにかかわらず、複合文を含んでいて、多くの点でストアドプロシージャに似ています。ループや条件付き実行などを追加することができます。

各イベントハンドラの実行後、エラーが発生しない場合、COMMIT が発生します。エラーが発生した場合は、ROLLBACK が発生します。

イベントハンドラのためのコンテキスト情報

ストアドプロシージャとは異なり、イベントハンドラには引数がありません。EVENT_PARAMETER 関数を使用して、イベントがトリガされたコンテキストに関する情報にアクセスできます。返される情報には、イベントがトリガされた接続 ID とユーザ ID、イベント名、実行回数が含まれます。

イベントハンドラのテスト

開発中は、好きなときにイベントハンドラをトリガできた方が便利です。TRIGGER EVENT 文を使うと、トリガ条件やスケジュールした時刻に関係なく、明示的にイベントを実行できます。ただし、無効なイベントハンドラを TRIGGER EVENT によって実行することはできません。

スケジュールとイベントの使用によるタスクの自動化

運用データベース上でイベントハンドラを開発するのはよいことではありませんが、明示的に ALTER EVENT 文を使ってイベントハンドラを無効にすることができます。

コードの共有

複数のイベントを処理するアクションを1つにまとめておくと便利です。たとえば、データベースファイルまたはログファイルを格納しているデバイスのディスク領域が少なくなってきたときに、通知アクションを実行することができます。これを実行するには、ストアプロシージャを作成し、各イベントハンドラの本文から呼び出します。このとき、必要なコンテキスト情報をパラメータとしてプロシージャに渡します。

イベントハンドラのデバッグ

イベントハンドラのデバッグは、ストアプロシージャのデバッグによく似ています。イベントハンドラは、イベントリストに表示されます。

イベントハンドラを隠す

ALTER EVENT 文と SET HIDDEN 句を使用して、イベントハンドラの定義を隠すことができます。SET HIDDEN 句を指定すると、ISYSEVENT システムテーブルの action カラムに格納されているイベントハンドラの定義が永続的に難読化されます。

アクティブなイベントの制限

また、EVENT_PARAMETER 関数と NumActive コンテキスト名を使用して、現在アクティブになっている特定のイベントハンドラのインスタンスの数を判断できます。この関数は、一定時間に1つのイベントハンドラで1つのインスタンスだけを実行させるように制限する場合に利用できます。

イベントやスケジュールに関する情報の取得

SAP Sybase IQ は、イベント、システムイベント、スケジュールに関する情報を、SYSEVENT、SYSEVENTTYPE、SYSSCHEDULE の各システムテーブルに保存しています。

ALTER EVENT 文を使用してイベントを変更するとき、イベント名のほかに、オプションでスケジュール名を指定します。TRIGGER EVENT 文を使用してイベントをトリガするには、イベント名を指定します。

イベント名は、SYSEVENT システムテーブルをクエリして、一覧表示できます。

```
SELECT event_id, event_name FROM SYSEVENT
```

スケジュール名は、SYSSCHEDULE システムテーブルをクエリして、一覧表示できます。

```
SELECT event_id, sched_name FROM SYSSCHEDULE
```

スケジュールとイベントの使用によるタスクの自動化

イベントには固有のイベント ID が付いています。イベントと関連するスケジュールの対応付けには、SYSEVENT と SYSSCHEDULE の `event_id` カラムを使用します。

データベースイベントの監査

監査により、データベースイベントがトランザクションログに記録されます。

dbtran データベース管理ユーティリティ

ログ変換ユーティリティ **dbtran** を使用すると、トランザクションログを `.sql` コマンドファイルに変換できます。

構文

データベースサーバに対して処理を実行する場合

```
dbtran [ options ] -c { connection-string } -n SQL-file
```

トランザクションログに対して処理を実行する場合

```
dbtran [ options ] [ transaction-log ] [ SQL-file ]
```

パラメータ

オプション	説明
@data	指定した環境変数または設定ファイルからオプションを読み込む。
-a	コミットされていないトランザクションをトランザクションログに表示するかどうかを制御する。トランザクションログには、トランザクションによって実行された最新の COMMT の前に加えられた変更が記録される。-a を指定しないと、出力ファイルにはコミットされたトランザクションのみが表示される。-a を指定した場合は、トランザクションログで検出された、コミットされていないトランザクションも表示される。
-c "keyword=val-ue; ..."	このユーティリティをデータベースサーバに対して実行する場合に、接続文字列を指定する。
-d	トランザクションを古いものから新しいものへと順に書き出すことを指定する。この機能はデータベースアクティビティの監査目的で使用される。dbtran の出力をデータベースに対して適用しないこと。

オプション	説明
-ek <i>key</i>	<p>強力に暗号化されているデータベースの暗号化キーを指定する。強い暗号で保護されたデータベースがある場合は、データベースまたはトランザクションログを使用するときに暗号化キーを指定する必要がある。-ek または -ep を指定すること。ただし、両方を指定することはできない。正しい暗号化キーを指定しないとコマンドが失敗する。データベースサーバに対して -c オプションを使用して dbtran を実行している場合は、-ek オプションではなく、接続パラメータを使用してキーを指定すること。たとえば、次のコマンドでは、データベースサーバサンプルからデータベース enc.db に関するトランザクションログが取得され、その出力が log.sql に保存される。</p> <pre data-bbox="417 574 1170 626">dbtran -n log.sql -c "ENG=sample;DBF=enc.db;UID=DBA;PWD=sql;DBKEY=mykey"</pre>
-ep	<p>暗号化キーを入力するよう要求する。このオプションを指定すると、暗号化キーを入力するためのウィンドウが表示される。クリアテキストでは暗号化キーを見ることができないようにすることで、高いセキュリティが得られる。-ek または -ep のいずれかを指定する。両方を指定することはできない。正しい暗号化キーを指定しないとコマンドが失敗する。-c オプションを使用して dbtran をデータベースサーバに対して実行する場合は、-ep オプションではなく、接続パラメータを使用してキーを指定する。たとえば、次のコマンドでは、データベースサーバサンプルからデータベース enc.db に関するトランザクションログが取得され、その出力が log.sql に保存される。</p> <pre data-bbox="417 1013 1170 1065">dbtran -n log.sql -c "ENG=sample;DBF=enc.db;UID=DBA;PWD=sql;DBKEY=mykey"</pre>
-f	<p>最後のチェックポイント以降に完了したトランザクションのみを出力する。</p>
-g	<p>auditing データベースオプションがオンの場合に、監査情報をトランザクションログに追加する。</p>
-ir <i>offset1,offset2</i>	<p>2つの指定オフセット間の部分的トランザクションログを出力する。</p>

オプション	説明
-is <i>source</i> ,...	次の1つ以上のソース(カンマ区切りのリストで指定)の操作によって変更された、ローに対する操作を出力する。 <ul style="list-style-type: none"> • All – すべてのロー。デフォルト設定。 • SQLRemote – SQL Remote を使用して変更されたローだけを含む。省略形の "SR" も使用可能。 • RepServer – Replication Agent (LTM) と Replication Server を使用して変更されたローだけを含む。省略形の "RS" も使用可能。 • Local – レプリケートされないローだけを含む。
-it <i>owner.table</i> ,...	カンマ区切りのリストに指定したテーブルに対する操作を出力する。各テーブルを <i>owner.table</i> として指定する。
-j <i>date/time</i>	最新のチェックポイントから指定の日付または時刻までの間に発生したトランザクションのみを変換する。ユーザは、日付、時刻、または日時のいずれかを引用符で囲んだものを引数として指定する。時刻を省略した場合、デフォルトは 00:00。日付を省略した場合、デフォルトは現在の日付になる。日付と時刻の有効なフォーマットは、"YYYY/MMM/DD HH:NN" である。
-k	エラーが検出されたときに部分的な .sql ファイルが消去されないようにする。 dbtran の実行中にエラーが検出された場合、その時点まで生成された .sql ファイルは、部分的なファイルが使用されないように、通常は消去される。このオプションを指定すると、破損したトランザクションログからトランザクションをサルベージしようとするときに役立つ。
-m	トランザクションログを格納するディレクトリを指定する。このオプションは -n オプションとともに使用する。
-n <i>filename</i>	dbtran をデータベースサーバに対して実行するときに SQL 文を格納する出力ファイルを指定する。
-o <i>filename</i>	指定したファイルに、出力メッセージを書き込む。
-r	コミットされていないトランザクションを削除する。これはデフォルトの動作である。
-rsu <i>username</i> ,...	ユーザ名のカンマで区切られたリストを指定し、デフォルトの Replication Server ユーザ名を上書きする。デフォルトでは、-is オプションは、デフォルトの Replication Server ユーザ名 dbmaint および sa を想定する。

オプション	説明
-s	UPDATE 文の生成方法を制御する。プライマリーまたはユニークインデックスがテーブルにない場合にこのオプションを使用しないと、 dbtran は、ローが重複しているときに非標準の FIRST キーワードの付いた UPDATE 文を生成する。このオプションを使用した場合は、 FIRST キーワードが省かれ、 SQL 標準との互換性が維持される。
-sr	SQL Remote によるリモートサイトへの操作の分散方法について記述するコメントを生成し、出力ファイルに挿入する。
-t	コマンドファイルにトリガが含まれるかどうかを制御する。デフォルトでは、トリガによって実行されたアクションはコマンドファイルに含まれない。コマンドファイルをデータベースに対して実行したときに、対応するトリガがデータベース内にある場合、トリガはアクションを自動的に実行する。コマンドファイルを実行するデータベース内に、対応するトリガが存在しない場合は、トリガアクションを含める必要がある。
-u <i>userid</i> ,...	指定したユーザのトランザクションログだけが出力されるように制限する。
-x <i>userid</i> ,...	指定したユーザ以外のトランザクションログが出力されるように制限する。
-y	確認メッセージを表示することなく、既存のコマンドファイルを置き換える。-q を指定する場合、-y も指定しないと操作は失敗する。
<i>transaction-log</i>	変換するログファイルを指定する。-c または -m オプションとは一緒に使用できない。
<i>SQL-file</i>	変換した情報を含む出力ファイルを指定する。 <i>transaction-log</i> のみで使用する。

使用法

dbtran は、次のように実行できます。

- データベースサーバに対して実行する — -c オプションに続けて指定した接続文字列によってデータベースサーバに接続し、-n オプションで指定したファイルに出力を格納します。この方法での実行には、**BACKUP DATABASE** システム権限が必要です。たとえば、次のコマンドでは、**iqdemo** というサーバのログ情報が変換され、**iqdemo.sql** という名前のファイルに出力が格納されます。

```
dbtran -c "eng=iqdemo;dbn=iqdemo;dbf=iqdemo.db;uid=DBA;pwd=sql" -n iqdemo.sql
```

- トランザクションログファイルに対して実行する — トランザクションログファイルに対して直接作用します。ユーザがこの文を実行しないように、トランザクションログファイルを一般のアクセスから保護してください。

```
dbtran iqdemo.log iqdemo.sql
```

dbtran では、トランザクションログの開始時のログオフセットが表示されます。これを使用して、生成された複数のログファイルの生成順序を判断できます。

dbtran-c は、オンライントランザクションログファイル、およびそれと同じディレクトリにあるすべてのオフライントランザクションログファイルの変換を試みます。ディレクトリに複数のデータベースのトランザクションログファイルが格納されている場合は、エラーが発生する場合があります。このエラーを防ぐには、必ず各ディレクトリに1つのデータベースのみのトランザクションログファイルが格納されるようにします。

1つのトランザクションが複数のトランザクションログにまたがる場合があります。トランザクションログファイルに複数のログにまたがるトランザクションが含まれている場合、1つのトランザクションログファイルの変換を実行すると (**dbtran demo.log** など)、複数のログにまたがるトランザクションが失われることがあります。**dbtran** で完全なトランザクションを生成するためには、ディレクトリ内のトランザクションログファイルで **-c** オプションまたは **-m** オプションを使用してください。

終了コードは、0 (正常) または 0 以外 (エラー) です。

このユーティリティは、**@filename** パラメータを受け付けます。

AUDITING オプション [データベース]

データベース内の監査を有効または無効にします。

指定できる値

ON、OFF

デフォルト値

OFF

スコープ

オプションは、データベース (PUBLIC) レベルでのみ設定できます。

このオプションを設定するには、SET ANY SECURITY OPTION システム権限が必要です。すぐに有効になります。

備考

監査とは、データベース内の多くのイベントに関する詳細をトランザクションログに記録することです。監査は、パフォーマンスの多少の低下と引き換えに、いくつかのセキュリティ機能を実現します。データベースの監査をオンにすると、トランザクションログの使用は停止できません。トランザクションログをオフにする前に、監査をオフにする必要があります。監査がオンの場合、データベースは読み取り専用モードで起動できません。

AUDITING オプションを機能させるには、監査オプションを ON に設定し、**sa_enable_auditing_type** システムプロシージャを使用して、監査対象の情報の種類を指定する必要があります。これにはパーミッションチェックの組み合わせ、接続試行回数、DDL 文、パブリックオプション、トリガなどが含まれます。次に該当する場合は監査が行われません。

- **AUDITING** オプションが OFF に設定されている場合
- 監査オプションが無効の場合

AUDITING オプションを ON に設定し、監査オプションを指定しない場合、すべての種類の監査情報が記録されます。

トラブルシューティングのヒント

SAP Sybase IQ には、問題に対応するためのリソースが数多く用意されています。

オンラインサポートのソース

ドキュメントを使用して問題を解決できない場合は、SAP Sybase IQ オンラインサポート Web サイトの MySybase を参照してください。

MySybase では、必要に応じてカスタマイズしたビューを使用して、最新の詳細なサポートケース、ソフトウェア情報、解決済みや既知の問題などを検索できます。オンラインで [Technical Support] を開くこともできます。

MySybase は、ほとんどのインターネットブラウザから使用することができます。この無料サービスに登録し、サービスを利用する方法を参照するには、Web ブラウザで MySybase を指定します。その他の有益な SAP Sybase Web サイトについては、『リリースノート』を参照してください。

状況別の解決策

特定の問題を診断、解決するために、より多くの情報が必要な場合もあります。診断ツールを使用すると、さまざまな状態を診断できます。

サーバのリカバリとデータベースの修復のための判断フロー

サーバやデータベースの起動、データベースへの接続、またはデータベースの検証で問題が発生することがあります。

1. サーバは起動するか？

当てはまる場合は、手順 2 に進みます。

当てはまらない場合は、「サーバの運用上の問題」を参照してください。この項の推奨手順に従ってもサーバを起動できない場合は、「強制リカバリモードでのサーバの起動」を参照し、サーバを強制リカバリモードで起動します。

強制リカバリモードでサーバが起動しない場合は、Sybase 製品の保守契約を結んでいるサポートセンタに連絡してください。バックアップからデータベースをリストアする必要がある場合があります。

2. データベースに接続できるか？

トラブルシューティングのヒント

データベースに接続できない場合は、トラブルシューティングの推奨事項について「データベース接続の問題」を参照してください。

データベースに接続できており、前に強制リカバリモードでサーバを起動した場合は、「割り付けの問題の分析」で、データベース割り付けの検証とリークブロックのリカバリについての説明を参照してください。

データベースに接続できるが、データベースの一貫性が失われている可能性がある場合は、「データベースの検証」で、データベースの一貫性検査についての説明を参照してください。

3. サーバは動作していて接続できるが、データベースの一貫性を検証したい。

前にサーバを強制リカバリモードで起動したか、データベースの一貫性が失われている可能性がある場合は、DBCC の検査を実行してデータベースを検証してください。インデックスの一貫性とデータベースの割り付けを検査する方法については、「データベースの検証」を参照してください。

4. サーバは動作しており、接続できる。DBCC の検査を実行した。DBCC によって検出されたインデックスの矛盾または割り付けの問題を修復する必要がある。

sp_iqcheckdb を実行し、結果の Index Summary セクションと Index Statistics セクションにエラーが表示された場合は、DBCC を使用してインデックスの問題を修復する手順について「インデックスエラーの修復」を参照してください。

sp_iqcheckdb を実行し、結果の Allocation Summary セクションと Allocation Statistics セクションにエラーが表示された場合は、DBCC を使用して割り付けの問題を修復する手順について「DBCC を使用した割り付けの問題の修復」を参照してください。

サーバの運用上の問題

起動、停止、無応答、異常終了など、サーバの運用に影響を及ぼす可能性のある問題です。

SAP Sybase IQ が起動しない

サーバの起動で問題がある場合、**start_iq** は 0 以外の値を返します。

起動時に **-o** スイッチの後ろにログファイルを指定しなかった場合、SAP Sybase IQ では以下の中で最初に定義されているものにエラーが書き込まれます。

- \$IQDIR16/logfiles/<servername>.nnnn.stderr
- \$IQDIR16/logfiles/<servername>.nnnn.srvlog
- システムアプリケーションログファイル

考えられる原因はいくつかあります。

トランザクションログファイルがデータベースと一致していない

サーバログファイル (.srvlog) とサーバを起動しているウィンドウに次のメッセージが表示されます。

```
Starting database "dbname" (/dbdir/dbname.db)
at Fri Apr 27 2009 10:53 Transaction log: dbname.log
Error: Cannot open transaction log file
-- Can't use log file "dbname.log" since the database
file has been used more recently
Cannot open transaction log file
-- Can't use log file "dbname.log" since the database
file has been used more recently
Database server stopped at Fri Apr 27 2009 10:53
```

サーバの起動時にこれらのエラーが表示された場合は、サーバが正しいトランザクションログファイルを使用しているかどうかを確認します。適切なトランザクションログファイルが見つからない場合、この状況からリカバリする最も安全な方法は、最新の有効なバックアップからリストアすることです。

適切なトランザクションログファイルが見つからず、バックアップからのリストアもできない場合は、トランザクションログを使わない緊急リカバリの方法を実施します。

サーバがトランザクションログを見つけられない

サーバがトランザクションログを見つけられずに起動できない場合は、サーバログファイルにメッセージが表示されます。

```
Transaction log: /dbdir/dbname.log...
Error: Cannot open transaction log file
-- No such file or directory
Cannot open transaction log file
-- No such file or directory
```

サーバの起動時にこのエラーが表示された場合は、トランザクションログファイルを見つけて、データベース .db ファイルと同じディレクトリにコピーします。適切なトランザクションログファイルが見つからない場合は、最新の有効なバックアップからリストアします。

他にサーバを起動する方法がない場合は、緊急リカバリの `-f` オプションを使用してサーバを起動できる可能性があります。必要に応じて、SAP Sybase 製品の保守契約を結んでいるサポートセンタに問い合わせてください。

警告！ この手順はきわめて危険性が高いので、非常時以外は使用しないでください。

サーバ名がネットワーク上で重複している

複数のサーバが同じ名前のサーバを持つ場合、**start_iq** コマンドでサーバを起動しようとしたときに、サーバログファイル(*.srvlogまたは**-o**起動オプションで指定された名前のファイル)に次のメッセージが表示されます。

```
DBSPAWN ERROR: -85  
Communication error
```

サーバログファイルにこれらのエラーが表示されてサーバが起動しない場合は、**iqsrv16** コマンドを使用してサーバを起動してみます。**iqsrv16** コマンドを使用すると、次のようなより具体的なエラーメッセージが返されます。

```
A database server with that name has already started
```

ネットワーク上でサーバ名が重複していることが問題の原因と確認したら、すでに実行中のサーバ名と異なる名前を使ってサーバを起動します。

ログファイル名が不正である

異なる要求レベルのログファイルを指定しているが、ファイル名が不正な識別子であると、サーバの起動時にエラーが発生します。

```
Naming conflict: "iqdemo" --  
aborting
```

```
Database naming conflict --  
aborting startup
```

これらのエラーは、**-zo** オプションで指定したファイルパスにスペースが含まれていることを示している場合があります。

スペースを含むファイル名を引用符で囲んで**-zo** オプションを再度指定します。

サーバのポート番号がマシン上で重複している

SAP Sybase IQ サーバを実行中に、同じマシン上で同じポート番号を使用して別の SAP Sybase IQ サーバを起動しようとする、サーバログファイル(*.srvlog)に次のメッセージが表示されます。

```
Trying to start TCPIP link ...  
TCPIP communication link not started  
Unable to initialize requested communication links  
...  
DBSPAWN ERROR: -85  
Communication error
```

```
Server failed to start
```

サーバログファイルにこれらのメッセージが表示されてサーバが起動しない場合は、**stop_iq** コマンド (UNIX) を実行し、マシンですでに実行されている SAP

Sybase IQ サーバの名前とポート番号を表示します。次に、未使用のポート番号を指定するか、ポート番号を指定せずにサーバを起動してみます。ポート番号を指定せずにサーバを起動した場合 (そしてデフォルトのポート番号がすでに使用されている場合) は、SAP Sybase IQ が使用可能なポート番号を生成します。

ポート番号を指定せずにサーバを起動すると、サーバログファイルに次のようなメッセージが表示されます。

```
Trying to start TCPIP link ...
Unable to start on default port; starting on port
49152 instead
TCPIP link started successfully
Now accepting requests
...
Server started successfully
```

サーバが間違っただパスで起動された

新しいマルチプレックスサーバを起動するときに、データベースファイルパスは、マルチプレックスサーバの作成時に指定したデータベースファイルパスと一致している必要があります。

間違っただパスを使用した場合、サーバの起動が失敗し、サーバログファイル (*.srvlog) に次のメッセージが表示されます。

```
E. 08/18 07:22:19. MPX: server myserver
has been started with an incorrect catalog path
(expected path: /work/IQ-16_0/demo/mympx/iqdemo.db) .
-- (st_database.cxx 7883)
I. 08/18 07:22:19. Database server shutdown due
to startup error
DBSPAWN ERROR: -82
Unable to start specified database: autostarting
database failed
```

これらのメッセージが表示される場合、適切なパスを指定してサーバを再起動します。サーバパスに関して UNIX のソフト (シンボリック) リンクの使用を計画する場合、ソフトリンクを作成してから **CREATE MULTIPLEX SERVER** を実行する必要があります。

環境変数が正しく設定されていない

データベース設定ファイルのパラメータが、**start_iq** で使用されたパラメータと異なる場合は、正しいパラメータを使用してサーバが起動されるようにします。

start_iq を実行できない

start_iq コマンドを実行できず、通常は設定ファイルまたは他のコマンドラインスイッチを使用している場合は、**start_iq** にサーバ名とデータベース名だけを指定してサーバを起動してみます。

この単純なコマンドでサーバが起動する場合は、コマンドラインまたは設定ファイルで指定した1つ以上のスイッチやパラメータが問題の原因と考えられます。サーバの起動を妨げているパラメータやスイッチを切り分けてみます。

最も基本的な **start_iq** コマンドを使用してもサーバが起動しない場合は、通常の設定ファイルとコマンドラインスイッチを使用して **iqdemo** デモデータベースを起動してみます。iqdemo データベースを使用するとサーバが起動する場合は、使用中のデータベースに問題がある可能性があります。

それでも **start_iq** コマンドを実行できない場合は、**iqsrv16** コマンドを使用します。

注意： **iqsrv16** は、サーバ起動エラーのトラブルシューティングを行う場合にのみ使用します。SAP Sybase IQ サーバの起動には、必ず **start_iq** を使用します。

iqsrv16 コマンドを実行する前に、次の作業を実行してください (通常は **start_iq** によって実行される作業です)。

- すべての制限を削除してから、スタックサイズおよび記述子を制限します。Cシェルに移動し、次のコマンドを入力します。

```
% unlimited
% limit stacksize 8192
% limit descriptors 4096
```

注意： **unlimit** はソフト制限にのみ影響します。ハード制限がある場合は、カーネルパラメータを設定して、それらを変更してください。

- 使用しているプラットフォームに合わせて、すべてのサーバオプションを適切に設定します。『インストールおよび設定ガイド』を参照してください。
- **iqsrv16** を呼び出す前に、環境にパス **\$SYBASE/OCS-15_0/lib** を追加して、エンジンと必要なライブラリをロードします。このパスを環境に設定するのはテストの間だけです。次のように設定します。

AIX の場合：

```
% setenv LIBPATH "${LIBPATH}:{SYBASE}/OCS-15_0/lib"
```

他の UNIX/LINUX プラットフォームの場合：

```
% setenv LD_LIBRARY_PATH "${LD_LIBRARY_PATH}:${SYBASE}/OCS-15_0/lib"
```

データベースが相対パス名を使用して作成されている場合は、データベースが配置されたディレクトリからデータベースサーバを起動する必要があります。

サーバを起動する場合、現在のディレクトリを確認してください。相対パス名を使用して作成される新しいデータベースファイルのロケーションは、サーバ起動

ディレクトリによって決まります。異なるディレクトリからサーバを起動すると、SAP Sybase IQ はデータベースファイルを検出できません。

サーバ起動スクリプトでは、サーバ起動コマンドを発行する前に既知の場所に変更する必要があります。

iqsrv16 の構文は次のとおりです。

```
iqsrv16 -n server-name -gm number
[ other-server-switches ] [ database-file [ database-switches ] ]
```

注意： **iqsrv16** コマンドラインでは、最後に指定されたオプションが優先されます。したがって、設定ファイルを上書きする場合は、変更するオプションを設定ファイル名の後にリストします。次に例を示します。

```
iqsrv16 @iqdemo.cfg -x 'tcpip{port=1870}' iqdemo
```

この例では、**-x** パラメータによって、`iqdemo.cfg` ファイルの接続情報が上書きされます。

iqsrv16 コマンドを実行してもサーバが起動しない場合は、**iqsrv16** ユーティリティを最小限のスイッチとパラメータとともに使用して、再度サーバを起動してみます。次に例を示します。

```
iqsrv16 -n <servername> <dbname>.db -c 32m
-gd all -gl all
```

最小限のパラメータとスイッチを使用するとサーバが起動する場合は、通常サーバの起動に使用しているいずれかのパラメータやスイッチが問題の原因となっている可能性があります。サーバの起動を妨げているパラメータやスイッチを切り分けてみます。

iqsrv16 コマンドを使用してサーバを起動する場合は、サーバはバックグラウンドで実行されず、メッセージは自動的にサーバログに保存されません。ただし、**-o file name** サーバスイッチを指定すると、メッセージはサーバウィンドウだけでなく、指定されたファイルにも送信されます。

SAP Sybase IQ が処理または応答を停止する

サーバが応答しない問題の原因は SAP Sybase IQ メッセージファイルを調べるとわかります。

考えられる原因

サーバが応答しない場合に最もよくある原因は、次のとおりです。

- ディスク領域の不足
- メインバッファキャッシュまたはテンポラリバッファキャッシュの領域の不足

対処法

処理中または停止の最中にサーバが応答しなくなる傾向がある場合は、**start_iq** コマンドラインオプション **-z** と SAP Sybase IQ データベースオプション **QUERY_PLAN = 'ON'** を使用して、SAP Sybase IQ メッセージファイル (.iqmsg) とサーバログファイル (.srvlog) に情報を記録します。

この情報を記録することに加えて、問題の原因を突き止めるためのいくつかの手順があります。

- SAP Sybase IQ メッセージファイルとサーバログファイルの両方に You have run out of space... メッセージがあるかどうかを確認します。IQ メインストアまたは IQ テンポラリストアの領域が不足している場合は、**CREATE DBSPACE** コマンドで適切な DB 領域を追加します。

データベースオプションの **MAIN_RESERVED_DBSpace_MB** と **TEMP_RESERVED_DB_SPACE_MB** を、**DDL COMMIT** または **CHECKPOINT** 時の領域不足に対処できるだけの十分な値に設定することも重要です。数百 MB あれば十分ですが、大きなデータベースではさらに高い値に設定できます。

- CPU 使用率をオペレーティングシステムレベルで数分間モニタし、SAP Sybase IQ サーバプロセス (iqsrv16) が CPU サイクルを消費しているかどうかを確認します。この情報を記録します。CPU 使用率が変化している場合は、SAP Sybase IQ サーバプロセスは正常に処理を実行しています。

SAP Sybase IQ サーバの CPU 使用率が正常な場合は、サーバが何をしているか、つまりサーバが現在どの文を実行しているかを調べることができます。

- 領域の不足を示す兆候がない場合は、新しい接続または既存の接続で Interactive SQL を使用して、次の情報を指定の順序で収集します。

表 22 : サーバが応答しない場合に収集する情報

コマンド	収集する情報
SELECT db_name()	データベース名
CHECKPOINT	チェックポイントを実行できる
sa_conn_properties ># sa_conn_properties.out	接続情報
sa_conn_info ># sa_conn_info.out	接続情報
sa_db_properties ># sa_db_properties.out	データベースのプロパティ情報
sa_eng_properties ># sa_eng_properties.out	サーバのプロパティ情報
sp_iqstatus ># sp_iqstatus.out	データベースのステータス情報

コマンド	収集する情報
sp_iqconnection ># sp_iqconnection.out	接続情報
sp_iqtransaction ># sp_iqtransaction.out	トランザクション情報

この問題を解決できない場合は、保守契約を結んでいる SAP Sybase サポートセンターにお問い合わせください。サポートセンターが問題を診断するときに、ここで収集した情報を使用します。

- サーバの応答がない場合は、DumpAllThreads または dumpallthreads というファイルを \$IQDIR16/logfiles ディレクトリ (Windows 64 プラットフォームでは %ALLUSERSPROFILE%\¥SybaseIQ¥logfiles フォルダ、Vista 64 では C:\¥ProgramData¥SybaseIQ¥logfiles) に作成することで、SAP Sybase IQ スレッドごとにスタックトレースを生成します。推奨手順に従って、プログラム マネージャまたは **start_iq** コマンドを使用して SAP Sybase IQ を起動すると、*IQDIR16* 変数が自動的に設定されます。*IQDIR16* 変数が設定されていない場合は、iqsrv16 を起動したディレクトリに DumpAllThreads ファイルを作成します。

SAP Sybase IQ サーバは DumpAllThreads ファイルの存在を検出し、スタックトレースファイル stktrc-YYYYMMDD-HHNNSS_#.iq に各 IQ スレッドのスタックトレースを書き込みます。スタックトレースがスタックトレースファイルに書き込まれた後、DumpAllThreads ファイルは削除されます。

このスタックトレース情報は、SAP Sybase 製品の保守契約を結んでいるサポートセンターが問題を診断するときに役立ちます。

- データベースに接続できる場合は、メインバッファキャッシュとテンポラリ(プライベート)バッファキャッシュに対して **IQ UTILITIES** バッファキャッシュモニタを次のように 10 秒間隔で 10 分間実行します。
 1. データベースに接続するか、既存の接続を使用します。
 2. CREATE TABLE #dummy_monitor(c1 INT);
 3. IQ UTILITIES MAIN INTO #dummy_monitor START MONITOR '-append -debug -interval 10 -file_suffix iqdbgmon';
 4. IQ UTILITIES PRIVATE INTO #dummy_monitor START MONITOR '-append -debug -interval 10 -file_suffix iqdbgmon';
プロセスを 10 分間実行した後、バッファキャッシュモニタを停止します。
 5. IQ UTILITIES MAIN INTO #dummy_monitor STOP MONITOR;
 6. IQ UTILITIES PRIVATE INTO #dummy_monitor STOP MONITOR;
- SAP Sybase IQ メッセージファイルの終わり近くに Resource count 0 というメッセージがあるかどうかを確認します。このメッセージの後ろに Open Cursor というメッセージが続く場合もあります。これらのメッセージはリソースの枯渇を示しており、デッドロックが発生する可能性があります。この問題を直ちに解決するには、[CTRL] キーを押しながら [C] キーを押すか、また

は **DROP CONNECTION** コマンドを使用して、アクティブな接続数を減らします。

リソースの枯渇によるデッドロックを回避するための長期的な解決策は、次の1つ以上の方法を使用することです。

- **-gm** サーバ起動オプションの値を減らして、サーバ上のユーザ数を減らす
- 別のセカンダリサーバをマルチプレックスに追加する
- CPU を追加してハードウェアの処理能力を上げる

システム障害/SAP Sybase IQ の障害

システム/SAP Sybase IQ の障害の原因は、SAP Sybase IQ メッセージファイルを調べるとわかります。

考えられる原因

さまざまな原因が考えられます。

対処法

- データベースを再起動する前に、メッセージログファイル (dbname.iqmsg) をコピーするか、ファイル名を変更します。こうすることで、ファイル内の有用な情報が失われないようにします。
- UNIX の場合は、スタックトレースのコピーを、SAP Sybase 製品の保守契約を結んでいるサポートセンタに送ります。スタックトレースは、データベースサーバを起動したディレクトリにある `stktrc-YYYYMMDD-HHNNSS_#.iq` という名前のファイルに記録されています。障害が発生したときにデータベースを開いていた場合は、SAP Sybase IQ メッセージログ (デフォルト名は `dbname.iqmsg`) にもスタックトレースが記録されています。SAP Sybase サポートセンタが障害の原因を突き止めるときに、この情報が役に立ちます。
- **start_iq** コマンドを使用してサーバを再起動します。データベースが再起動すると、リカバリが自動的に行われます。
- データベースを起動せずにサーバを起動してみます。サーバは起動できるが、データベースを起動できない場合は、開始行や接続プロファイルでデータベースのパラメータが正しく指定されているかどうかを確認します。
- カタログストアテーブルのクエリを広範囲に実行する場合は、サーバを再起動し、**TEMP_SPACE_LIMIT_CHECK** オプションが ON になっていることを確認します。このオプションを設定すると、接続がカタログストアのテンポラリファイル領域のクォータを超えた場合に、致命的でないエラーが返されます。

サーバを停止できない

サーバを停止するには、**dbstop** コーティリティまたは **stop_iq** を実行するか、UNIX のサーバウィンドウで **q** と入力するか、または Windows のサーバウィンドウで [シャットダウン] をクリックします。

考えられる原因

さまざまな原因が考えられます。

対処法

サーバを停止できない場合は、以下の対処法を実行します。

UNIX システムでは、次の手順に従います。

1. **ps** オペレーティングシステムユーティリティの出力を取得し、この出力を製品の保守契約を結んでいるサポートセンタに提出できるようにします。Sun Solaris では、2 種類の **ps** オプションを使用できます。両方を使用してください。

```
ps -aAdeflclj|egrep "PPID|iqrsv16"
```

```
/usr/ucb/ps -awwwlx|egrep "PPID|iqrsv16"
```

2. プロセスをオペレーティングシステムレベルで強制終了し、コアダンプを生成します。

```
kill -6 pid
```

start_iq を実行したディレクトリに小さなコアファイルが作成されます。この方法でサーバプロセスを強制終了できた場合は、手順 5 に進みます。

3. これでもサーバプロセスが終了しない場合は、手順 1 と同様に **ps** の出力を取得します。 **ps** を実行する両方の回 (プロセスの強制終了を試みる前と後) の出力を保持します。次に、より強力なシグナルを使ってプロセスを強制終了します。

```
kill -9 pid
```

4. この方法を使用してもプロセスが終了しない場合は、もう一度 **ps** の出力を取得した後、システムを再起動します。

5. すべての **ps** 出力、コアファイル (手順 2 で生成した場合)、**stktrc-YYYYMMDD-HHNNSS_#.iq** のスタックトレースを製品の保守契約を結んでいるサポートセンタに提出します。

Windows システムでは、次の手順に従います。

1. タスク バーを右クリックし、[タスク マネージャ] をクリックしてタスク マネージャを起動します。
2. [プロセス] タブで [iqrsv16.exe] を選択し、[プロセスの終了] ボタンをクリックしてデータベースサーバを停止します。

3. 必要に応じて Windows を再起動します。

データベース接続の問題

データベースへの接続時に問題が発生することがあります。

データベースに接続できない

データベースの接続の問題が発生することがあります。

考えられる原因

- データソースが定義されていないか、誤って定義した。正しいユーザ ID とパスワードを使用して接続し直します。
データソースは一連の接続パラメータで、レジストリ (Windows の場合) またはファイル (Windows と UNIX) に格納されます。
- 誤ったユーザ名またはパスワードを指定した。
- ユーザにデータベースを使用するパーミッションが与えられていない。
- TDS を使用して (jConnect を使用するなど) 接続するときに、ユーザ ID またはパスワードが 30 バイトを超えている。次のようなメッセージが表示されます。

```
Invalid user ID or password
CT-LIBRARY error:
ct_connect(): protocol specific layer:
external error: The attempt to connect to the server failed.
```

- 誤ったデータベースファイル名を指定した。正しいデータベースファイル名を使用して接続し直します。
Interactive SQL を使用し、utility_db への接続時にバックアップからデータベースをリストアしたときは、**DBF** パラメータと接続するデータベースファイル名を指定する必要があります。
- データベースファイルが見つからない。dbname.db、dbname.iq、dbname.iqmsg (dbname はデータベース名) の各ファイルがすべて存在している必要があります。
- 接続数の制限、または DBA が定義したその他のログインの制限を超えた。
- ディスク領域が不足している。SAP Sybase IQ メッセージファイルでディスク領域関連のメッセージを確認します。
- 指定したサーバ名が誤っている。サーバの名前を確認し、正しいサーバ名を使用して接続し直します。
- サーバのマシン名またはアドレスが変更された。
- クライアントから初めて接続するときに、サーバ名を指定せず、誤ったポート番号を指定すると、データベースに接続できないことがある。この場合は次のエラーメッセージが返されます。

```
Could not connect to the database.
Database server not found.
```

Interactive SQL から接続する場合、[サーバ名] フィールドの名前のスペルが正しく、[ネットワーク] タブで適切なネットワークが選択されており、データ

ベースサーバが起動していることを確認します。接続するときにはサーバ名を指定するか、正しいポート番号を使用します。サーバ名と、サーバが受信しているポート番号を確認するには、**stop_iq** (UNIX) を実行して情報を表示します。

- ポート番号が適切な範囲を超えているか、別のプロセスに使用されている。
- クライアントを起動しようとしたときに

```
Unable to start - server not found
```

と

```
Database server not running.
```

のいずれかのメッセージを受信した場合、クライアントはネットワーク上でデータベースサーバを見つけることができない。接続文字列が誤っているか、サーバ名キャッシュに誤った接続情報や古い接続情報が含まれている可能性があります。たとえば、別のポート番号を使用してサーバを起動した場合は、クライアントアプリケーションで接続時に新しいポート番号を指定しても、接続情報はサーバ名キャッシュから取得されます。

- CharSet 接続パラメータで文字セットを指定し、その文字セットをサポートしていないサーバに接続しようとした。

CharSet を指定せずに接続し直してみます。クライアントのローカルの文字セットをサーバがサポートしていない場合、接続は成功しますが、文字セットがサポートされていないという警告が表示されます。

対処法

データベースに問題があるために接続できないと思われる場合は、`dbname.iqmsg` ファイルを調べて、問題が発生した箇所を突き止めることができます。

「データベースは正常に開かれています。」というメッセージは、データベースが正常に開かれていることを示しているため、問題はクライアント接続に関係しています。メッセージが表示されていない場合は、データベースが開くときまたはリカバリするときに障害が起きた可能性があります。

Interactive SQL (dbisql) の問題

`dbisql` を使用中に問題が発生することがあります。

dbisql を終了した後もディレクトリが残る

この問題は、NFS ファイルシステムのユーザにのみ影響します。

考えられる原因

IQTMP16 環境変数がローカルディレクトリを指していない。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。これらのファイルは、接続が終了するときに SAP Sybase IQ によって削除

されます。IQTMP16がローカルディレクトリを指していない場合、NFSが作成する .nfs* ファイルを見つけることができません。

対処法

IQTMP16をローカルディレクトリに設定し、サーバを再起動します。

リソースの問題

リソースの問題には、ディスク領域の不足、スレッド数の不足、スレッドスタックオーバーフロー、未使用のシステムリソースなどがあります。

ディスク領域の不足

SAP Sybase IQ サーバは、DB 領域が不足したときに、領域が追加されるのを待つのではなく、トランザクション全体をロールバックするか、セーブポイントまでロールバックします。

バッファまたは DB 領域割り付け要求に利用できるテンポラリまたはメイン DB 領域が不足している場合、要求を行う文がロールバックします。

この時点で、DBA は **ALTER DBSPACE** コマンドまたは **ALTER FILE** コマンドを使用して DB 領域に領域を追加できます(DB 領域の代わりにファイルを追加することもできます。1つの DB 領域に複数の DB 領域ファイルを含めることができます)。

警告！ ディスク領域が不足したときに、SAP Sybase IQ が特定のシステムロックを保持しているか、チェックポイントを実行していると、ディスク領域を追加できないことがあります。したがって、ディスク領域を使い果たす前に、ディスク領域が残り少ないことを認識し、新しい DB 領域を追加することが大切です。

対処法

- SAP Sybase IQ メッセージログ (dbname.iqmsg) で最近のメッセージを確認します。「out of space」というメッセージは、別の DB 領域を追加する必要があることを示しています。SAP Sybase IQ メッセージファイルのメッセージには、領域が不足している DB 領域が示されます。データの挿入時に問題が起きる場合は、IQ メインストアの領域を増やす必要があると考えられます。大量のソートマージジョインを行うクエリ時に問題が起きる場合は、IQ テンポラリストアの領域を増やす必要があると考えられます。

SAP Sybase IQ のメッセージログに以下のメッセージがないかどうかを確認します。

- DB 領域が不足しているときにバッファまたは DB 領域の割り付け要求が失敗した場合は、次のエラーメッセージが dbname.iqmsg メッセージファイルに記録されます。

```
You have run out of space in %2 DBSpace. %1
```

```
[EMSG_OUT_OF_DBSPACE: SQL Code -1009170L,
SQL State QSB66, Sybase Error Code 20223]
```

%2 は DB 領域の名前です。

- DB 領域が不足しているときにトランザクション全体がロールバックされた場合、次のエラーメッセージが報告されます。

```
%1 -- Transaction rolled back"
```

```
[IQ_TRANSACTION_ROLLBACK: SQL Code -1285L,
SQL State 40W09, Sybase Error Code 2973]
```

%1 は、重要な操作中にサーバによって検出された、トランザクションをロールバックさせる原因になったエラーです。

- バッファの割り付け要求がダーティなバッファを検出したが、バッファマネージャが領域不足のためにバッファをフラッシュできない場合、次のエラーメッセージが返されて、現在の文がロールバックします。

```
%2: All buffer cache pages are in use, ask your
DBA to increase the size of the buffer cache. %1
```

```
[EMSG_BUFMAN_ALLSLOTSLOCKED: SQL Code -1009031L,
SQL State QSA31, Sybase Error Code 20052]
```

%2 は例外となる特別なバッファキャッシュです。

- 新しい接続を使用してデータベースに接続してみます。正常に接続できる場合、クエリは待機していても、データベースサーバは動作しています。**sp_iqstatus** を実行し、詳細な情報を取得します。
- データベースに接続できない場合は、そのプロセッサの CPU 使用率をモニターして、SAP Sybase IQ が使用できない状態かどうかを確認します。CPU 使用率が短い時間間隔で変化しない場合、SAP Sybase IQ は停止していると考えられます。CPU 使用率が変化している場合、SAP Sybase IQ は稼働しています。
- sp_iqstatus** 出力で、次の行を確認します。

```
Main IQ Blocks Used:,10188 of 12288,
82%, Max Block#: 134840
```

```
Temporary IQ Blocks Used:,163 of 6144,
2%, Max Block#: 97
```

使用ブロックの割合が 90% 台の場合は、**CREATE DBSPACE** コマンドを使用してディスク領域を増やす必要があります。この例では、メイン IQ ブロックの 82% とテンポラリ IQ ブロックの 2% が使用されているので、IQ メインストアでは間もなく領域の追加が必要となります。

- 領域不足条件が発生している場合、またはマルチプレックスサーバで **sp_iqstatus** によるメインブロックの使用が高いパーセンテージを示している場合は、**sp_iqversionuse** を実行して、使用されているバージョン、およびそのバージョンでリカバリできる領域の量を確認します。

チェックポイント中の領域不足

強制リカバリモードで起動し、できるかぎり早く領域を追加します。

新しいチェックポイントを実行する前に、DB 領域を追加する必要があります。

ディスク領域が不足している状況でのチェックポイントの影響

チェックポイントが要求されたときに、すでに SAP Sybase IQ でディスク領域が不足していると、次のエラーが表示されて **checkpoint** コマンドが失敗します。

```
You have run out of space during the CHECKPOINT operation.
```

```
[EMSG_IQSTORE_OUTOFSPACE_CHECKPOINT:'QSB33', 1009133].
```

新しいチェックポイントを実行する前に、DB 領域を追加する必要があります。

サーバに接続できない場合の領域の追加

操作中に領域を使い果たし、サーバに接続できないために領域を追加できない場合は、**CREATE DBSPACE** コマンドを使用して領域を追加します。

1. 次のいずれかの方法でサーバを停止します。
 - プラットフォームの種類を問わず、**dbstop** を実行します。
 - Windows の場合は、Windows タスク バーで目的のサーバアイコンをクリックし、SAP Sybase IQ ウィンドウを表示し、[停止]ボタンをクリックします。
 - UNIX の場合は、**stop_iq** を実行するか、サーバを起動したウィンドウで `q` と入力します。
2. **start_iq** コマンドを使用してエンジンを再起動します。
3. データベースに接続します。
4. **CREATE DBSPACE** コマンドを使用して領域を追加します。
5. 領域の不足によって失敗した操作を再度実行します。

DB 領域管理のサイズの管理

カタログファイルのサイズが増加するのは正常なことで、その割合はアプリケーションとカタログの内容によって異なります。 .db ファイルのサイズがパフォーマンスに影響を与えることはなく、.db ファイル内の空きページが必要に応じて再利用されます。

カタログファイルの増大を最小限に抑えるには、次の方法を使用します。

- **CREATE TABLE** 文に対しては、**IN SYSTEM** を使用しないようにする
- システムストアプロシージャを実行した後で **COMMIT** 文を発行する
- 長時間実行されるトランザクションの後、**COMMIT** 文を発行する

カタログストアのいずれかのファイル (.tmp、.db、または .iqmsg) のサイズを拡大できない場合、SAP Sybase IQ は「A dbspace has reached its maximum

file size」というエラーを返します。この問題を回避するには、次のようにします。

- 領域の使用状況を定期的にモニタします。
- .tmp ファイル、.db ファイル、または .iqmsg ファイルがある場所に、オペレーティングシステムのファイルサイズの制限 (Sun Solaris の **ulimit** など) が設定されていないことを確認します。.db ファイルと .tmp ファイルは通常、SAP Sybase IQ のメインのデータベースディレクトリにあります。.tmp ファイルは、\$IQTMP16/<servername>/tmp の下か、\$IQTMP16 が設定されていない場合は /tmp/.SQLAnywhere/<servername>/tmp の下にあります。

誤ったタイプの領域の追加

テンポラリ DB 領域の領域が不足している場合に、**TEMPORARY** キーワードを指定せずに **CREATE DBSPACE** コマンドを実行すると、テンポラリ DB 領域を作成できません。

代わりに、IQ_SYSTEM_TEMP として存在している DB 領域にファイルを追加します。

断片化

SAP Sybase IQ では、最小の未使用領域まで利用して、断片化を制御しています。

それでも、断片化はやはり発生します。データベースの領域が不足している場合は、**sp_iqstatus** ファイルまたは .iqmsg ファイルによってリストされるメモリ使用率で、使用中のメイン IQ ブロックの値が 100% 未満でも、通常はデータベースが断片化していることを示します。

領域の解放

接続の領域が不足しているときに、別の接続でテーブルやインデックスを削除して領域を解放することはできません。領域が不足したトランザクションは、これらのオブジェクトをスナップショットバージョンで参照するからです。

将来のための領域の確保

SAP Sybase IQ は最低 200MB、最後の DB 領域のサイズの 50% を自動的に予約します。

将来領域が不足した場合に備えて、新しい DB 領域を追加できるだけの十分な領域を確保するには、データベースオプションの **MAIN_RESERVED_DBSPACE_MB** と **TEMP_RESERVED_DBSPACE_MB** を、**COMMIT** または **CHECKPOINT** の実行時の領域不足に十分対処できるだけの値に設定します。

ディスク領域の使用状況のモニタリング

イベントハンドラを使用して、ディスク領域の使用状況をモニタし、利用可能な領域が残り少なくなったら通知させることができます。

この項で示す最初の例は、ロード時の領域をモニタするときに特に役立ちます。ロードを開始する前にイベントハンドラを有効にし、ロードが完了した後でイベントハンドラを無効にできます。

このイベントハンドラのサンプルコードを変更すれば、他の種類のモニタリングも実行できます。

```
-- This event handler sends email to the database
-- administrator whenever the IQ main DBSpace is more than
-- 95 percent full.

-- This event handler runs every minute. The event handler uses
-- sp_iqspaceused to sample the space usage. If the space is
-- more than 95 percent full, a file that contains the date and
-- time is created in the directory where iqsrv16 is
-- running. The file contents are then mailed to the database
-- administrator and the file is removed.
-- This event can be enabled before a load and be used
-- to monitor disk space usage during loading. The event can
-- then be disabled after the load.

create event out_of_space
schedule
start time '1:00AM' every 1 minutes
handler

begin
declare mt unsigned bigint;
declare mu unsigned bigint;
declare tt unsigned bigint;
declare tu unsigned bigint;

call sp_iqspaceused(mt, mu, tt, tu);

if mu*100/mt > 95 then
  call xp_cmdshell('date > ./temp_m_file');
  call xp_cmdshell('mailx -s add_main_dbSPACE iqdba@iqdemo.com
  < ./temp_m_file');
  call xp_cmdshell('/bin/rm -rf ./temp_m_file');
end if;

if tu*100/tt > 95 then
  call xp_cmdshell('date > ./temp_file');
  call xp_cmdshell('mailx -s add_temp_dbSPACE iqdba@iqdemo.com
  < ./temp_file');
  call xp_cmdshell('/bin/rm -rf ./temp_file');
end if;

end
```


次のコードは領域の使用状況をモニタするタイマベースのイベントを作成し、権限を付与されていない操作時の領域不足状態のときに発生する可能性がある予期しないロールバックを防止します。DBSpaceLogger イベントは、サンプル iqdemo データベースに作成されます。

```
CREATE EVENT DBSpaceLogger
SCHEDULE START TIME '00:00:01' EVERY 300 SECONDS
HANDLER
BEGIN
DECLARE DBSpaceName VARCHAR(128);
DECLARE Usage SMALLINT;
DECLARE cursor_1 CURSOR FOR
SELECT DBSpaceName, Usage
FROM sp_iqdbspace()
WHERE Usage > 0
ORDER BY Usage
FOR READ ONLY;

OPEN cursor_1;
idx1: LOOP
FETCH cursor_1 INTO DBSpaceName, Usage;
IF SQLCODE <> 0 THEN LEAVE idx1 END IF;
IF Usage >= 70 AND Usage < 80 THEN
call dbo.sp_iqlogtoiqmsg('Information: DBSpace' +
DBSpaceName + ''s usage is more than 70%');
ELSEIF Usage >= 80 AND Usage < 90 THEN
call dbo.sp_iqlogtoiqmsg('Warning: DBSpace ' +
DBSpaceName + ''s usage is more than 80%');
ELSEIF Usage >= 90 AND Usage < 100 THEN
call dbo.sp_iqlogtoiqmsg('Critical Warning: DBSpace
' + DBSpaceName + ''s usage is more than 90%');
END IF;
END LOOP;
CLOSE cursor_1;
END;
```

スレッドの不足

クエリに必要な数のサーバスレッドが利用できないことがあります。

考えられる原因

このクエリで使用可能な十分なサーバスレッドがありません。[-1010011] ['QXA11'] というクライアントメッセージは、IQ ストア用に追加のカーネルスレッドをクエリが必要としていることを示します。

対処法

- 別のクエリが終了し、そのクエリが使用しているスレッドが解放されるのを待ちます。その後でクエリを再発行します。
- **sp_iqconnection** を実行します。カラム IQThreads に、現在接続に割り当てられている IQ スレッド数が表示されます。このカラムから、どの接続がリソー

スを最も多く使用しているかを判断できます。割り当て済みのスレッドでも、アイドルである可能性があります。

- 状況が持続する場合は、サーバを再起動し、IQ スレッドを追加指定する必要があります。-iqmt サーバ起動スイッチを使用して、SAP Sybase IQ が使用できる処理スレッド数を増やします。

デフォルトは、最初の4 CPU に対して CPU ごとに 60 スレッド、残りの CPU に対して CPU ごとに 50 スレッド、それにシステムで使用するための 3 スレッドとデータベース接続およびバックグラウンドタスクに必要なスレッドを加えた数です。たとえば、CPU が 12、接続スレッドが 10 のシステムの場合は、 $60 * 4 + 50 * (\text{numCPUs} - 4) + \text{numConnections} + 6 = 656$ となります。最小値は、 $\text{numConnections} + 3$ です。サーバスレッドの総数は、10000 を超えてはいけません。

- サーバのスレッドが不足している場合、またはリストア中に接続で十分なスレッドを使用できない場合には、エラー Ran out of threads. Start up server with more threads. (SQLCODE -1012024) が表示されることがあります。RESTORE コマンドは、スレッドの「チーム」を割り付けてリストア操作を実行しようとします。SAP Sybase IQ では、バックアップデバイスごとに少なくとも 1 つのスレッド、CPU ごとに 2 つのスレッド、またチームに 1 つのスレッドを割り付けようとします。各接続や各チームだけでなく、サーバにも十分なスレッドが割り付けられていることを確認してください。データベースオプション MAX_IQ_THREADS_PER_CONNECTION と MAX_IQ_THREADS_PER_TEAM を使用します。

スタックオーバーフロー

スレッドスタックがオーバーフローする場合に問題が発生することがあります。

スタックトレースファイル (stktrc-YYYYMMDD-HHNNSS_#.iq) の

AbortIfEndofStack は、スレッドスタックがオーバーフローしていることを示します。

考えられる原因

- この問題を回避するには、サーバパラメータ -iqtss を 300 (32 ビットオペレーティングシステムの場合) または 500 (64 ビットオペレーティングシステムの場合) に設定して SAP Sybase IQ を再起動します。サーバ起動スイッチ -iqtss は、スレッドスタックサイズを KB 単位で指定します。それでも不十分な場合は、-iqtss の値を問題が解決するまで 72 ずつ上げていきます。
- 可能であれば、エラーの原因となったコマンドを特定し、製品の保守契約を結んでいるサポートセンタに送付します。

異常終了後に残される未使用のセマフォと共有メモリ

異常終了した場合に、未使用のセマフォと共有メモリが残ることがあります。

AIX、HP-UX、およびLinuxのプラットフォームでは、同一コンピュータ上でのクライアントとサーバ間の通信にセマフォを使用します。各クライアントおよび各サーバは、1つのセマフォを割り付けます。クライアントは読み込み対象のサーバにパケットを送信するときにサーバのセマフォに信号を送ります。その逆も同様です。特定のシステムに必要なセマフォの数は、共有メモリ経由でローカルサーバに接続するローカルクライアントアプリケーションの数によって異なります。クライアントは、1つ以上のサーバへのマルチプレックス接続用に複数のセマフォの割り付けが必要な場合は、すべてのセマフォをセマフォグループに割り付けようとしています。

考えられる原因

UNIXシステムでプロセスを強制終了すると、セマフォや共有メモリが自動的にクリーンアップされずに、残されたままになることがあります。不必要なセマフォを除去するには、定期的にUNIXの**ipcs**コマンドを実行し、セマフォと共有メモリのステータスを確認します。

ipcs -a コマンドは、セマフォと共有メモリセグメントのID番号、所有者、作成時刻をリストします。SAP Sybase IQのすべてのインスタンスが(推奨されているように)同じユーザによって起動された場合は、OWNERカラムでそのユーザ名を検索できます。使用されていない共有メモリセグメントとセマフォを特定します。

対処法

それらの共有メモリセグメントとセマフォを使用していないことを所有者に確認した後、UNIXの**ipcrm**コマンドを実行して削除します。メモリセグメントIDの指定には**-m**パラメータを、セマフォID番号の指定には**-s**コマンドを次の形式で使用します。

```
ipcrm -m mid1 -m mid2 ... -s sid1 -s sid2 ...
```

次に例を示します。

```
% ipcrm -m 40965 -s 5130 -s36682
```

バッファの不足

リソースマネージャが、キャッシュが足りなくて操作を実行できないと判断した場合、操作は起動せず、Insufficient buffers (バッファの不足) が戻されます。

これらの提案の1つを使用して問題を解決してください。

- バッファキャッシュのサイズを増やして、操作を再度実行する。

トラブルシューティングのヒント

- サーバがそれほどビジーではなく、より多くのバッファキャッシュが利用できる時間帯に操作をスケジュールし直す。
- マルチプレックス環境において、負荷を別のノードに移行する。

処理の問題

処理の問題が、ロード、クエリ、インデックス、テーブルアクセスに関連していることがあります。

テーブルのインデックスが多すぎる

テーブルのインデックスが多すぎる場合、問題が発生することがあります。

考えられる原因

Microsoft Access ユーザが、32 を超えるインデックスを持つテーブルにリンクしようとしている。

対処法

テーブル内のすべてのカラムを選択するビューを作成し、ベーステーブルの代わりにそのビューにリンクします。

ロードまたはクエリに予想以上の時間がかかる

ロードまたはクエリに長い時間がかかることによって、問題が発生する場合があります。

考えられる原因

- IQ バッファキャッシュが大きすぎるために、オペレーティングシステムがスラッシングしている。
- IQ バッファキャッシュが小さすぎるために、SAP Sybase IQ がクエリデータをキャッシュに格納しきれず、スラッシングしている。
- システムのメモリ要件の合計がシステムメモリの合計を超えるように IQ バッファキャッシュサイズを設定しようとした。そのため、バッファキャッシュが自動的にデフォルトのサイズに縮小された。
- ユーザ定義の関数またはデータベース間のジョインが CIS の介在を必要としている。
- **WHERE** 句と **GROUP BY** 句で使用されているカラムに HG インデックスまたは LF インデックスがない。

対処法

スラッシングが問題かどうかを判断するために、ページングをモニタします。

- IQ のページングをモニタするには、IQ バッファキャッシュモニタを実行します。

- オペレーティングシステムのページングをモニタするには、UNIX の **vmstat** ユーティリティや他のプラットフォーム固有のツール、または Windows のパフォーマンス モニタを使用します。

必要に応じてバッファサイズをリセットします。

ハッシュアルゴリズムを伴うクエリ実行時のスラッシングの量を制限することもできます。HASH_THRASHING_PERCENT データベースオプションを調整し、許容するハードディスク I/O の割合を制御します。この割合を超えると、文がロールバックされてエラーが返されます。

HASH_THRASHING_PERCENT のデフォルト値は 10% です。この値を大きくするとロールバックするまでのディスクへのページング量が増え、この値を小さくするとページング量が減ります。

以前のバージョンの SAP Sybase IQ では実行されていた、ハッシュアルゴリズムを伴うクエリが、HASH_THRASHING_PERCENT のデフォルト制限に達するとロールバックされるようになり、以下のメッセージのいずれかが表示されます。

- Hash insert thrashing detected.
- Hash find thrashing detected.(SQLState QFA43, SQLCode -1001047)

実行に必要なリソースをクエリに割り当てるには、以下の対応策を 1 つ以上講じてください。

- HASH_THRASHING_PERCENT の値を増やし、ページングの制限を緩和する。
- テンポラリキャッシュのサイズを増やす (DBA のみ)。テンポラリキャッシュのサイズを増やすと、メインバッファキャッシュのサイズが減る。
- SAP Sybase IQ がこの文の 1 つ以上のハッシュサイズの見積もりを誤っている原因を突き止めて改善する。
- データベースオプション HASH_PINNABLE_CACHE_PERCENT の値を減らす。

クエリで起きている可能性のある問題を特定するには、テンポラリデータベースオプション QUERY_PLAN='ON' と QUERY_DETAIL='ON' を指定してクエリを実行し、クエリプランを生成します。そして、クエリプランの見積もりを調査します。QUERY_PLAN_AFTER_RUN='ON' を指定すると、クエリの実行が終了した後でクエリプランが出力されるので、追加情報が得られます。生成されたクエリプランはメッセージログファイルにあります。

ユニークな値の数でロードが失敗する

クエリ内のユニークな値の数が原因で問題が発生することがあります。

考えられる原因

ログファイルの次のメッセージは、LF インデックスを持つカラムに 10000 を超えるユニークな値があることを示します。

トラブルシューティングのヒント

```
1009103: Number of unique values exceeded for index.  
index_name_LF 10000
```

Low_Fast インデックスは 1000 のユニークな値用に最適化されていますが、10000 が上限となっています。

対処法

LF インデックスを **HG** インデックスで置き換えます。**DROP INDEX** 文を発行し、エラーメッセージに示された **LF** インデックスを削除します。次に例を示します。

```
DROP INDEX DBA.employee.emp_lname_LF
```

次に、**CREATE INDEX** 文を発行し、新しい **HG** インデックスを作成します。次に例を示します。

```
CREATE HG INDEX ON DBA.employee (emp_lname)
```

ロックテーブルに書き込めない

ロックテーブルが原因で問題が発生することがあります。

考えられる原因

他のユーザがすでに書き込みアクセスを行っているオブジェクトに対して書き込もうとすると、次のようなエラーメッセージが表示されます。Cannot open the requested object for write in the current transaction (TxnID1).Another user has write access in transaction TxnID2.

対処法

sp_iqlocks ストアドプロシージャを使用して、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特定します。このプロシージャは、ロックを保持している接続とユーザ ID、ロックが保持されているテーブル、ロックのタイプ、ロックを識別する名前を含めて、データベースで現在保持されているロックについての情報を表示します。

エラーメッセージには、書き込もうとしているユーザのトランザクション ID (TxnID1) と、現在書き込んでいるユーザのトランザクション ID (TxnID2) も含まれています。テーブルをロックしているトランザクションについての詳細な情報が必要な場合は、**sp_iqtransaction** ストアドプロシージャを実行します。

テーブルでの書き込みロック競合の管理

トランザクションの大部分がロックを取得できる場合、複数のユーザが使用するテーブルで書き込みロックに対する多くの競合が発生すると、処理に影響を与えることがあります。

このサンプルストアドプロシージャは、テーブルでの書き込みロックに対する競合を管理する方法の一例です。このプロシージャは、テーブルでの書き込みロ

ク競合を解消はしませんが、競合を管理して、トランザクションが書き込みロックを取得できるようにします。

次のサンプルストアードプロシージャコードは、イベントの記録に使われる `dbo.event` という名前のテーブルでのロック競合を管理します。プロシージャは呼び出し元に `event_id` を返します。このテーブルでは、書き込みロックに対する多くの競合が発生します。ストアードプロシージャ `dbo.log_event` は、テーブル `dbo.event` に情報を記録します。アクセスエラーが発生すると、エラーが取得され、プロシージャは 5 秒間スリープした後、再びテーブルへの書き込みを試みます。5 秒間の再試行間隔は、通常は競合が解決されるのに十分な長さなので、`dbo.event` テーブルに対する書き込みロックを取得できるようになります。

このコードを変更すれば、他の同様のタスクも実行できます。

```
if exists (select 1
           from   sys.sysprocedure a
           join   sys.sysuserperm b on a.creator = b.user_id
           where  a.proc_name = 'log_event' and b.user_name = 'dbo')
then
    drop procedure dbo.log_event;
end if;

create procedure dbo.log_event(in @event varchar(255))
on exception resume
begin
    declare @event_id   bigint;
    declare @res        char(5);
    set @event_id=0;
    loop1: loop
        commit work;
        select  max(event_id)+1
                into    @event_id
                from    dbo.event;
        insert  dbo.event
                values (@event_id,@event,current timestamp,null,null);
        set @res=sqlstate;
        if @res = ' ' or(@res <> 'QDA29' and @res <> 'QDA11') then
            leave loop1
        end if;
        call dbo.sleep(5);
    end loop loop1;
    commit work;
    return @event_id
end
```

重要な更新操作のエラーを防ぐには、事前に必要なすべてのテーブルで書き込みロックを予約します。たとえば、次の例では `SalesOrders`、`Customers`、`SalesOrderItems` の各テーブルで書き込みロックを予約しますが、これらは仮想更新をする際に必要です。

トラブルシューティングのヒント

```
BEGIN
WHILE TRUE LOOP
    LOCK TABLE SalesOrders, SalesOrderItems, Customers IN WRITE MODE
WAIT '30:00:00';
    If SQLCODE indicates that lock could not be acquired
    then
        SET status_msg = 'lock for required tables
        not yet acquired - retrying';
        Message to client status_msg;
    ELSE
        BREAK;
    ENDIF;
END LOOP; // Locks on SalesOrders, SalesOrderItems, Customers are
acquired
Update table SalesOrders ...;
INSERT INTO SalesOrderItems ...;
LOAD INTO Customers ...;
COMMIT;
END;
```

チェックポイントのヒント

チェックポイント時間とリカバリ時間はデフォルト値で十分なため、ほとんどの場合、変更する必要はありません。

チェックポイントの間隔は、デフォルトで 60 分に設定されています。

サーバを起動するときにチェックポイントの間隔を調整するには、**start_iq** コマンドまたは `dbname.cfg` 設定ファイルで **-gc** オプションと **-gr** オプションを変更します。**-gc** スイッチでは、チェックポイントタイムアウト時間を分単位で指定します。**-gr** スイッチでは、最大リカバリ時間を分単位で指定します。データベースエンジンは、両方のスイッチを使用してチェックポイント時間を計算します。

start_iq データベースオプションの詳細については、『ユーティリティガイド』を参照してください。

パフォーマンスの問題

パフォーマンスに影響を及ぼす可能性のある設定があります。

マルチ CPU またはハイパースレッディング対応マシンでの低いパフォーマンス

マルチ CPU またはハイパースレッディング対応マシンで問題が発生することがあります。

考えられる原因

SAP Sybase IQ は、使用可能な物理 CPU の数を把握しているときに最も効率的に動作します。ハイパースレッディングが有効になっているマシンや、使用可能なすべての CPU に SAP Sybase IQ がアクセスできないマシンでは、SAP Sybase IQ はスレッドを作成しすぎて、本来のパフォーマンスを発揮できません。

対処法

-iqnumbercpus を SAP Sybase IQ で使用可能な CPU 数に設定してサーバを起動し、物理的な CPU 数を上書きします。

start_iq データベースオプションの詳細については、『ユーティリティガイド』を参照してください。

ネットワーク通信のトラブルシューティング

ネットワークソフトウェアにはさまざまなコンポーネントが含まれているため、トラブルシューティングが必要な問題が発生しやすくなります。

ネットワークのトラブルシューティングを支援する一番の情報源は、ネットワーク通信ソフトウェアベンダが提供しているネットワーク通信ソフトウェアのマニュアルやテクニカルサポートです。ただし、ベストプラクティスを実行したり、診断ツールを使用したりして、さまざまな状況に関する情報を取得することができます。

互換性のあるプロトコルの使用

クライアントまたはサーバコンピュータに複数のプロトコルスタックがインストールされている場合は、クライアントとデータベースサーバが同じプロトコルを使用していることを確認してください。

サーバで使用するプロトコルのリストはサーバの **-x** コマンドラインスイッチで選択し、クライアントアプリケーションで使用するプロトコルのリストは **CommLinks** 接続パラメータで選択します。

これらのオプションを使用して、各アプリケーションが同じプロトコルを使用していることを確認します。

デフォルトでは、データベースサーバもクライアントライブラリも、使用可能なすべてのプロトコルスタックを使用します。サーバはすべてのアクティブなプロトコルでクライアント要求をサポートし、クライアントはすべてのアクティブなプロトコルでサーバを検索します。

start_iq データベース起動ユーティリティの **-x** スwitchの詳細については、『ユーティリティガイド』を参照してください。

最新のドライバの使用

必要に応じて、ネットワークアダプタの最新バージョンの NDIS または ODI ドライバを使用しているかどうかを確認してください。

最新のネットワークアダプタドライバは、アダプタカードの製造元または供給元から入手できます。

トラブルシューティングのヒント

ネットワークアダプタの製造元や供給元では、カードの最新バージョンのドライバを提供しています。カードの製造元の多くは Web サイトを公開しているので、そこから最新バージョンの NDIS および ODI ドライバをダウンロードできます。

ネットワークソフトウェアの供給元からも、最新のネットワークアダプタドライバを入手できる場合があります。

Novell クライアントソフトウェアをダウンロードすると、すべてのネットワークアダプタで使用される Novell ソフトウェアに加えて、いくつかのネットワークアダプタ用の ODI ドライバが含まれています。

再起動時のコンピュータの電源切断

ネットワークアダプタボードの中には、コンピュータを再起動したときに完全にリセットされないものがあります。トラブルシューティングを行うときは、コンピュータの電源を切り、数秒間待ってから電源を入れます。

レイヤごとのプロトコルスタックの診断

クライアントアプリケーションとデータベースサーバの通信に問題がある場合は、クライアントとデータベースサーバが互換性のあるプロトコルスタックを使用しているかどうかを確認します。

ネットワーク通信の問題を切り分ける方法は、プロトコルスタックを下から順に調べて、各レベルの通信が正常に機能しているかどうかをテストします。

SAP Sybase IQ と同じ上位レイヤプロトコルを使用した接続でなくても、サーバコンピュータに接続できるのであれば、データリンクレイヤは機能しています。

たとえば、クライアントアプリケーションを実行しているコンピュータから、データベースサーバを実行しているコンピュータのディスクドライブに接続してみます。

データリンクレイヤが機能していることを確認したら、次は SAP Sybase IQ と同じネットワークレイヤとトランスポートレイヤを使用する他のアプリケーションが機能しているかどうかを確認します。

TCP/IP プロトコルスタックのテスト

TCP/IP を実行している場合は、クライアントコンピュータとサーバコンピュータの TCP/IP プロトコルスタックの互換性テストに使用できるいくつかのアプリケーションがあります。

ping を使用した IP レイヤのテスト

各 IP レイヤには、ピリオドで区切られた 4 つの整数 (191.72.109.12 など) から成るアドレスが関連付けられています。**ping** は IP アドレスの引数を取り、指定された IP プロトコルスタックに 1 つのパケットを送信します。

最初に、自分自身のコンピュータに ping を送信し、プロトコルスタックが正しく機能しているかどうかを確認します。たとえば、IP アドレスが 191.72.109.12 の場合、コマンドプロンプトで次のコマンドを入力します。

```
ping 191.72.109.12
```

パケットがルート指定されているかどうかを確認します。ルート指定されている場合は、次のような出力が表示されます。

```
c:> ping 191.72.109.12
Pinging 191.72.109.12 with 32 bytes of data:
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
...
```

ping が成功した場合は、コンピュータが自身にパケットをルーティングできていることを示します。これにより、IP レイヤは正しく設定されていると考えることができます。TCP/IP を実行している他のユーザに IP アドレスを聞いて、そのユーザのコンピュータに ping を送信してみます。

他の診断に進む前に、クライアントコンピュータからデータベースサーバを実行しているコンピュータに ping を送信できることを確認してください。

Telnet を使った TCP/IP スタックのテスト

TCP/IP スタックをさらにテストするために、1 台のコンピュータでサーバアプリケーションを起動し、別のコンピュータでクライアントプログラムを起動して、正しく通信できるかどうかをテストします。

TCP/IP の実装に通常付属するいくつかのアプリケーションをこの目的に使用できます。telnet コマンドを使用して TCP/IP スタックをテストする手順は次のとおりです。

1. 1 台のマシンで Telnet サーバプロセス (または *daemon*) を起動します。方法については、TCP/IP ソフトウェアのマニュアルを参照してください。通常のコマンドライン Telnet プログラムでは、コマンドプロンプトで次の命令を入力します。

```
telnetd
```

2. 別のマシンで Telnet クライアントプロセスを起動し、接続できるかどうかを確認します。この方法についても、TCP/IP ソフトウェアのマニュアルを参照してください。通常、次のような命令を入力します。

```
telnet server_name
```

この命令で、*server_name* は Telnet サーバプロセスを実行しているコンピュータの名前または IP アドレスです。

2 つのマシンの間で Telnet 接続が確立されるということは、プロトコルスタックが安定し、クライアントとサーバは TCP/IP リンクを使用して通信できることを示し

ています。Telnet 接続を確立できない場合は、問題があります。他の診断に進む前に、TCP/IP プロトコルスタックを正しく機能させる必要があります。

配線の問題の診断

ネットワーク配線やコネクタの不良が原因で起きる問題は、切り分けが難しいことがあります。

同じ設定の同じようなマシンで問題を再現してみます。特定のマシンでのみ問題が発生する場合は、配線の問題かハードウェアの問題です。

NetWare で配線の問題を検出する方法については、Novell NetWare のマニュアルを参照してください。イーサネットまたはトークンリングネットワークの配線の問題を診断する場合は、Novell LANalyzer プログラムが役に立ちます。NetWare 認定リセラーに、配線の問題の診断および解決を支援できる NetWare 認定技術者の名前を教えてください。

一般的なネットワーク通信の問題の確認

いくつかの一般的なネットワーク通信の問題とその解決策を把握しておいてください。

"Unable to start — server not found" メッセージ

クライアントを起動しようとしたときに "Unable to start - server not found" というメッセージを受信した場合は、クライアントがネットワークでデータベースサーバを検索できないことを示します。

- クライアントマシンのネットワークドライバのネットワーク設定パラメータが、サーバマシンの設定パラメータと異なっています。たとえば、2つのイーサネットアダプタカードが共通のフレームタイプを使用している必要があります。Novell NetWare では、フレームタイプは net.cfg ファイルで設定されます。Windows では、[コントロールパネル] のネットワーク設定でフレームタイプ設定を検索します。
- TCP/IP プロトコルでは、クライアントは要求をブロードキャストしてデータベースサーバを検索します。こうしたブロードキャストは、通常はゲートウェイを通過しないので、別の(サブ)ネットワークにあるマシンのデータベースサーバは検出されません。この場合は、サーバが実行されているマシンのホスト名を -x サーバ起動コマンドラインオプションで指定する必要があります。TCP 経由で NetWare サーバに接続するときは、この指定が必須です。
- ネットワークドライバまたはネットワーク配線が正しくインストールされていません。
- ネットワークドライバのネットワーク設定パラメータが、SAP Sybase IQ マルチユーザサポートに適合していない可能性があります。

"Unable to initialize any communication links" メッセージ

"Unable to initialize any communication links" というメッセージは、いずれのリンクも確立できないことを示します。

原因として、ネットワークドライバがインストールされていないことが考えられます。-x サーバ起動オプションで特に指定しないかぎり、サーバとクライアントは使用可能なすべてのプロトコルを使用して通信リンクを開始しようとします。必要なドライバをインストールする方法については、ネットワークのマニュアルを参照してください。

診断ツール

いくつかのツールは、さまざまな状況を診断するのに役立ちます。

新しいテナポラリファイルトポロジへのリストア

テナポラリ DB 領域ファイルを開くことができないか、またはテナポラリ DB 領域ファイルが破損している場合は、データベースを別のテナポラリファイルトポロジにリストアできます。

1. バックアップされたデータベースのすべてのテナポラリ IQ ファイル定義をリストア時に無視するようにしてユーティリティサーバを起動します。

```
start_iq -n utility_startup_svr -c 32m
-x 'tcpip{port=1234}' -iqnotemp
```

2. データベースをリストアします。

```
RESTORE DATABASE 'iqdemo'
FROM '/system1/IQ16/IQ-16_0/demo/backup/iqmain'
```

3. **-iqnotemp** フラグを使用して、リストアしたデータベースを再起動します。
4. **IQ_SYSTEM_TEMP** 内のすべてのファイルを削除します。

```
ALTER DBSPACE IQ_SYSTEM_TEMP DROP FILE ALL
```

5. **-iqnotemp** フラグを指定せずにサーバを再起動します。
6. 新しいテナポラリ DB ファイルを **IQ_SYSTEM_TEMP** に追加します。

sp_iqstatus ストアドプロシージャ

sp_iqstatus ストアドプロシージャは、さまざまな SAP Sybase IQ ステータス情報を提供します。

注意： 次の例は、iqdemo サンプルデータベースの出力を示しています。サンプルのユーザ DB 領域 iq_main は、ユーザが作成した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。

トラブルシューティングのヒント

Sybase IQ (TM)	Copyright (c) 1992-2013 by Sybase, Inc. All rights reserved.
Version:	16.0.0.6552/110812/P/GA/Sun_Sparc/OS 5.10/64bit/2012-08-12 03:08:39
Time Now:	2012-09-13 10:33:19.979
Build Time:	2012-08-13 03:08:39
File Format:	23 on 03/18/1999
Server Mode:	IQ Multiplex Coordinator Server
Catalog Format:	2
Stored Procedure Revision:	1
Page Size:	131072/8192blksz/16bpp
Number of Main DB Files:	2
Main Store Out Of Space:	N
Number of 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: 1-12800	IQ_SYSTEM_MAIN
DB Blocks: 1045440-1058239	iq_main
Local Temp Blocks: 1-3200	IQ_SYSTEM_TEMP
Create Time:	2013-08-17 11:31:03.313
Update Time:	2013-09-12 10:32:00.077
Main IQ Buffers:	510, 64Mb
Temporary IQ Buffers:	510, 64Mb
Main IQ Blocks Used:	8076 of 19200, 42%=63Mb, Max Block#: 1051107
Shared Temporary IQ Blocks Used:	0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used:	113 of 1600, 7%=0Mb, Max Block#: 834
Main Reserved Blocks Available:	6400 of 6400, 100%=50Mb
Shared Temporary Reserved Blocks Available:	0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available:	1600 of 1600, 100%=12Mb

IQ Dynamic Memory:	Current: 150mb, Max: 150mb
Main IQ Buffers:'	Used:509, Locked: 0
Temporary IQ Buffers:'	Used:8, Locked: 0
Main IQ I/O:'	I: L184357/P71 O: C18370/D25255/P20297 D: 5613 C:51.8
Temporary IQ I/O:'	I:L248471/P0 O: C22502/D25269/P4896 D:22494 C:59.3
Other Versions:'	2 = 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:	1
Blocks in next ISF Backup:	0 Blocks:=0Mb
Blocks in next ISI Backup:	0 Blocks:=0Mb
Main Tlvlog Size:	Pages: 2, Recs: 413, Replays: 0/0
DB File Encryption Status:	OFF

出力コード Main IQ I/O と Temporary IQ I/O 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

次の情報を確認します。

トラブルシューティングのヒント

- Main IQ Blocks Usedと Temporary IQ Blocks Usedの行は、DB領域の使用中の部分を示します。使用中のブロックの割合(行の中央の統計)が90%台後半の場合は、DB領域を追加します。
- 予約済みブロックはユーザ操作に使用できないので、Main IQ Blocks Usedと Temporary IQ Blocks Usedは、行 DB Blocks (メイン IQ ブロックの総数)-Main Reserved Blocks Availableと行 Temp Blocks (テンポラリ IQ ブロックの総数)-Temporary Reserved Blocks Availableに基づいて計算されます。
- Main IQ Buffersと Temporary IQ Buffersの行はそれぞれ、メインバッファキャッシュとテンポラリバッファキャッシュの現在のサイズを示します。
- Other Versionsは、他のdbのバージョンと消費された合計領域を示します。これらのバージョンは、アクティブなトランザクションから参照されなくなったり、参照できなくなったりすると、最終的に削除されます。
- Active Txn Versionsは、アクティブな書き込みトランザクションの数と、それらが作成、破棄したデータの量を示します。これらのトランザクションがコミットされると、「破棄された」データは古いバージョンとなり、最終的に削除されます。トランザクションがロールバックされると、「作成された」データは解放されます。
- Main Reserved Blocks Availableと Temporary Reserved Blocks Availableは、使用可能な予約領域の量を示します。
- Main IQ I/Oと Temporary IQ I/Oの行には、I/OステータスがIQメッセージログと同じフォーマットで表示されます。

通知メッセージの解釈

デフォルトでは、SAP Sybase IQは挿入とロード処理中のデータベースの情報をIQメッセージログ(.iqmsg ファイル)に出力します。

これらのメッセージの統計情報によって、DB領域の追加などの保守タスクと最適化タスクをいつ実行する必要があるかが示されます。メッセージはロードの進行状況についても報告します。

挿入開始時には、次のような処理の説明が通知されます。

```
In table 'tab2', the full width insert
of 2 columns will begin at record 1.
I. 02/11 13:28:14. 0000000002 Insert Started:
I. 02/11 13:28:14. 0000000002 tab2
I. 02/11 13:28:14. 0000000227 [20895]: Insert Pass 1
completed in 0 seconds.
I. 02/11 13:28:14. 0000000227 [20895]: Insert Pass 2
completed in 0 seconds.
I. 02/11 13:28:14. 0000000227 [20834]:
  1 records were inserted into 'tab2'.
```


NOTIFY ロードオプションで指定したレコード数が SAP Sybase IQ によって挿入されるたびに、サーバから次のようなメッセージが送信されます。

```
2010-05-27 13:03:49 0000000002
[20897]: 100000 Records, 2 Seconds
```

先頭行は、直前の通知メッセージ以降に SAP Sybase IQ が読み込んだロー数と、その読み込み処理に要した秒数を示します。SAP Sybase IQ が毎回同じ数のローを読み込む場合でも、データ読み込みの内容 (必要なデータ変換回数など) に応じて所要時間は変化します。報告された時間間隔が 1 秒よりも小さい場合は、通常「0 秒」として報告されます。

メモリメッセージ

メモリメッセージには、SAP Sybase IQ サーバのメモリ使用状況に関する情報が表示されます。

IQ メッセージログ (.iqmsg ファイル) のこの行には、次のようなメモリの使用状況に関する情報が表示されます。

```
Mem: 469mb/M470
```

表 23 : メモリ使用状況に関する情報

項目	説明
Mem:# mb	この SAP Sybase IQ サーバで使用中の現在のメモリ (メガバイト単位)。
M# mb	この SAP Sybase IQ サーバの起動後に使用された最大メガバイト数。

IQ メインストアブロックメッセージ

IQ メインストアブロックメッセージには、IQ メインストアのブロックとバッファの使用状況に関する情報が表示されます。

IQ メッセージログ (.iqmsg ファイル) のこの行は、永続的な IQ メインストアを示します。

```
Main Blks: U63137/6%, Buffers: U12578/L7
```

表 24 : IQ メインストアブロックメッセージ

項目	説明
U#	使用中のブロックの数
#%	データベースにデータが格納されているパーセンテージ

項目	説明
Buffers:U#	<p>使用中のバッファの数。バッファマネージャは、他のデータに対してバッファが必要になるまでは、バッファをメモリに残しておくため、これは通常100%である。SAP Sybase IQ は、バッファをできるだけ積極的かつ効率的に使用するため、一般には、使用されているバッファとロックされているバッファの値は無意味である。</p> <p>この値は、メインバッファキャッシュに適合するバッファの最大数まで大きくなる。バッファが割り当てられるたびにこの数字は増えるが、使用中のバッファの数が減るのはバッファが破損したときのみで、バッファのロックが解除されるか、またはフラッシュしたときは減らない。テンポラリキャッシュのオブジェクトは終了時にバッファを解放するが、メインキャッシュについては、SAP Sybase IQ がバッファを破棄する場合としない場合がある。バッファは、ロックが解除されているかぎりは再利用できるためである。そのバッファが空であるか、データが含まれているか、それとも破棄されたデータが含まれているかは関係ない。</p>
L#	<p>ロックされているバッファの数。ロックされているバッファは、現在使用中であり、キャッシュから取り除くことはできない。SAP Sybase IQ は、ハッシュオブジェクトなど一部のオブジェクトについては、メモリに保持しておくためにバッファをロックする。また、ソートなど、その他のオブジェクトについては、負荷の状況や、そのオブジェクトにとって適切かどうかという判断に基づいて、バッファをロックする。</p> <p>この数字はバッファを要求するたびに増える。スクリプトの実行中に最大値を超えると、最大値を超えたコマンドは失敗し、後続のコマンドは不正に終了する可能性がある。</p> <p>バッファのロックではメモリは使用されない。ロックされたバッファは、メモリ内の構造体にフラグが設定される。このフラグは、バッファがロックされているかどうかにかかわらず存在する。</p>

ディスク領域を使い果たす前に、サーバのディスク領域が残り少ないことを認識し、新しい DB 領域を追加することが大切です。イベントハンドラを使用してディスク領域の使用状況をモニタし、ロード時に使用可能な領域が少なくなった場合に通知する方法の例については、「ディスク領域の使用状況のモニタリング」を参照してください。

IQ テンポラリストアブロックメッセージ

IQ テンポラリストアブロックメッセージには、IQ テンポラリストアのブロックとバッファの使用状況に関する情報が表示されます。

IQ メッセージログ (.iqmsg ファイル) の次の行は、テンポラリ IQ ストアを示します。

```
Temporary Blks: U273/0%, Buffers: U1987/L1960
```

表 25 : IQ テンポラリストアブロックメッセージ

項目	説明
U#	使用中のブロックの数
#%	データベースにデータが格納されているパーセンテージ
Buffers:U#	<p>使用中のバッファの数。バッファマネージャは、他のデータに対してバッファが必要になるまでは、バッファをメモリに残しておくため、これは通常 100% である。IQ は、バッファをできるだけ積極的かつ効率的に使用するため、一般には、使用されているバッファとロックされているバッファの値は無意味である。</p> <p>テンポラリキャッシュのオブジェクトは終了時にバッファを解放する。</p>
L#	<p>ロックされているバッファの数。ロックされているバッファは、使用中であり、キャッシュから取り除くことはできない。IQ は、ハッシュオブジェクトなど一部のオブジェクトについては、メモリに保持しておくためにバッファをロックする。また、ソートなど、その他のオブジェクトについては、負荷の状況や、そのオブジェクトにとって適切かどうかという判断に基づいて、バッファをロックする。</p> <p>この数字はバッファを要求するときに増える。スクリプトの実行中に最大値を超えると、最大値を超えたコマンドは失敗し、後続のコマンドは不正に終了する可能性がある。</p> <p>バッファのロックではメモリは使用されない。ロックされたバッファは、メモリ内の構造体にフラグが設定される。このフラグは、バッファがロックされているかどうかにかかわらず存在する。</p>

ディスク領域を使い果たす前に、サーバのディスク領域が残り少ないことを認識し、新しい DB 領域を追加することが大切です。イベントハンドラを使用してディスク領域の使用状況をモニタし、ロード時に使用可能な領域が少なくなった場合に通知する方法の例については、「ディスク領域の使用状況のモニタリング」を参照してください。

メインバッファキャッシュアクティビティメッセージ

メインバッファキャッシュアクティビティメッセージには、IQ メインストアバッファキャッシュに関する情報が表示されます。

IQ メッセージログ (.iqmsg ファイル) の次の行は、IQ メインストアバッファキャッシュに関する情報を示します。

```
Main      I: L331224/P22 O: D25967/P7805 C:D0
```

表 26 : IQ メインストアバッファキャッシュメッセージ

項目	説明
Main:I:L#	論理ファイル読み込みの数

項目	説明
P#	物理ファイル読み込みの数
O:D#	バッファが破損した回数
P#	物理的な書き込み数
C:D#	バッファマネージャデータ圧縮比率。この値は、圧縮に必要な合計バイト数から圧縮後に使用されるバイト数を引き、圧縮に必要な合計バイト数で割って 100 倍したものである。つまり、この値はデータがどれだけ圧縮されているか (圧縮前のサイズに対する圧縮率) を示す。この値が大きいほど、圧縮率は高くなる。圧縮に適しているのは特定のデータブロックのみである。圧縮に適しているブロックは、インデックス、(データベースの 90 ~ 95%)、ソートセットなどである。このソートには、バッファマネージャで 사용되는データ圧縮技術のみが影響する。データがバッファマネージャに到達する前にその他のデータ圧縮が行われる可能性があるため、合計のデータ圧縮比はこれより大きくなる可能性がある。

一般に、バッファキャッシュが満杯であると仮定した場合に、物理的な読み込みあたりの論理読み込み回数は 10 ~ 1000 となります。値が 10 より小さい場合は、バッファマネージャで過度なスラッシングが発生しています。値が 1000 より大きい場合は、バッファキャッシュに割り付けるメモリが多すぎる可能性があります。

テンポラリバッファキャッシュメッセージ

テンポラリバッファキャッシュアクティビティメッセージには、IQ テンポラリストアバッファキャッシュに関する情報が表示されます。

IQ メッセージログ (.iqmsg ファイル) の次の行は、IQ テンポラリストアバッファキャッシュに関する情報を示します。

```
Temporary I: L25240/P8 O: D4749/P0 C:D0
```

表 27: テンポラリ IQ ストアバッファキャッシュメッセージ

項目	説明
Temporary:I:L#	論理ファイル読み込みの数
P#	物理ファイル読み込みの数
O:D#	バッファが破損した回数
P#	物理的な書き込み数

項目	説明
C:D#	バッファマネージャデータ圧縮比率。この値は、圧縮に必要な合計バイト数から圧縮後に使用されるバイト数を引き、圧縮に必要な合計バイト数で割って 100 倍したものです。つまり、この値はデータがどれだけ圧縮されているか (圧縮前のサイズに対する圧縮率) を示します。この値が大きいほど、圧縮率は高くなります。圧縮に適しているのは特定のデータブロックのみです。圧縮に適しているブロックは、インデックス、(データベースの 90 ~ 95%)、ソートセットなどです。この比率には、バッファマネージャで使用されるデータ圧縮技術のみが影響します。データがバッファマネージャに到達する前にその他のデータ圧縮が行われる可能性があるため、合計のデータ圧縮比はこれより大きくなる可能性があります。

一般に、バッファキャッシュが満杯であると仮定した場合に、物理的な読み込みあたりの論理読み込み回数は 10 ~ 1000 となります。値が 10 より小さい場合は、バッファマネージャで過度なスラッシングが発生しています。値が 1000 より大きい場合は、バッファキャッシュに割り付けるメモリが多すぎる可能性があります。

ユーザ名、接続ハンドル、接続 ID

テンポラリバッファキャッシュメッセージの後で、接続ハンドル、接続 ID (SA connID)、ユーザ名がデータベース接続ごとに 1 回 .iqmsg ファイルにロギングされます。

IQ メッセージログ (.iqmsg ファイル) の次の行は、接続情報を示します。

```
2010-05-12 09:34:42 0000000002 Txn 173
2010-05-12 09:34:42 0000000002 Connect: 1550990889. SA connID: 1.
User: DBA.
```

接続ハンドルは、**sa_conn_info** ストアドプロシージャで表示される値です。

注意： **-zr** ログファイル内の接続情報を .iqmsg ファイル内の情報に関連付けるには、「.srvlog ファイルと .iqmsg ファイル間での接続情報の関連付け」を参照してください。

sp_iqcheckdb ストアドプロシージャ

データベースに問題があると思われる場合は、**sp_iqcheckdb** ストアドプロシージャを実行してみます。

このプロシージャは、すべてのデータベースページをディスクからメモリに読み込み、さまざまな一貫性検査を行います。ただし、データベースのサイズによっては、検査の実行に時間がかかることがあります。

sp_iqdbstatistics には、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が表示されます。

データベースオプションとサーバ起動オプションの値の確認

サーバの起動、リソース、または処理の問題を診断するときに、データベースオプションとサーバ起動オプションの現在の値を確認することが必要になる場合があります。

接続済みのユーザの場合は、**sp_iqcheckoptions** ストアドプロシージャを実行すると、デフォルトから変更されているデータベースオプションの現在の値とデフォルト値のリストが表示されます。**sp_iqcheckoptions** は、デフォルト値から変更されているサーバ起動オプションもリストします。

DBA が **sp_iqcheckoptions** を実行すると、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。また、すべてのユーザにデフォルト以外のサーバ起動オプションが表示されます。

sp_iqcheckoptions ストアドプロシージャには、パラメータを指定する必要はありません。Interactive SQL で次のストアドプロシージャを実行します。

```
sp_iqcheckoptions
```

システムテーブル DBO.SYSOPTIONDEFAULTS には、SAP Sybase IQ と SQL Anywhere オプションのすべての名前とデフォルト値が格納されています。このテーブルにクエリを実行すると、すべてのオプションのデフォルト値を参照できます。

現在実行中の文の検索

問題を診断するときに、問題が発生したときにどの文が実行されていたかを知りたいかもしれません。

sp_iqcontext ストアドプロシージャは、現在実行中の文を一覧表示し、その文を発行したユーザと接続を特定します。**sp_iqconnection**、**.iqmsg** ログ、**-zr** サーバ要求ログ (**.srvlog**)、スタックトレースから取得した情報をこのユーティリティとともに使用すると、問題発生時の状況を判断できます。

接続情報を使用して **.iqmsg** ログと **-zr** サーバ要求ログのエントリを照合する場合は、**.srvlog** ファイルと **.iqmsg** ファイル間で接続情報を関連付けます。

サーバ要求のロギング

特定の種類の問題、特にクエリの問題を切り分ける場合は、サーバ要求を記録します。

要求レベルログを有効にするには、次の2つの方法があります。

- サーバを起動するときに **-zr** コマンドラインオプションを設定する。
- **sa_server_option** ストアドプロシージャを呼び出す。このプロシージャは、**-zr** の現在の設定より優先されます。

サーバ要求は、*.srvlog に記録されます。**-zr** サーバ起動オプションは、操作の要求レベルログを有効にし、ログに記録する要求のタイプ (SQL | HOSTVARS | PLAN | PROCEDURES | TRIGGERS | OTHER | BLOCKS | REPLACE | ALL | NONE) を設定します。**-zo** オプションは、通常のログファイルとは別のファイルに、要求レベルログ情報をリダイレクトします。**-zs** は、このファイルのサイズを制限します。

注意： ログに書き込まれているクエリテキストのサイズが指定制限値を超えた場合、そのクエリテキストはトランケートされず、テキスト全体がログに記録されます。

データベースサーバを起動した後で、**sa_server_option** を使用し、要求ログ設定を調整してロギングの対象とする情報を増減できます。次のコマンドは、限られた一連の要求の要求ロギングを有効にし、出力を sqllog.txt ファイルにリダイレクトします。

```
call sa_server_option('RequestLogging','SQL');
      call sa_server_option('RequestLogFile',
                           'sqllog.txt');
```

要求レベルログを無効にするには、次のストアドプロシージャを使用します。

```
call sa_server_option('RequestLogging','NO');
```

SQL ログファイルとログレベルの現在の設定を参照するには、次の文を実行します。

```
select property('RequestLogFile'), property('RequestLogging');
```

接続情報を使用して .iqmsg ログと **-zr** サーバ要求ログ (.srvlog) のエントリを照合する場合は、.srvlog ファイルと .iqmsg ファイル間で接続情報を関連付けます。

注意： SAP Sybase IQ バージョン 15.1 で、要求ログが修正されました。固定フォーマット行プレフィックスの代わりに、一般的な情報はカンマ区切りテキストで記録されることになりました。可能な場合、時間は "=" (前の行と同じ) または +nnn (前の行の nnn ミリ秒後) として記録されます。そのため、SAP Sybase IQ 15.1 より前のバージョンに比べ、要求ログはかなり小さくなっています。

さらに、追加情報も要求ログに記録されます。クエリでは、独立性レベル、フェッチされたロー数、およびカーソルタイプが情報として記録されます。**INSERT** 文、**UPDATE** 文、**DELETE** 文では、影響を受けたロー数と起動されたトリガ数が情報として記録されます。

トラブルシューティングのヒント

場合によっては、プロシージャおよびトリガ内で実行される文をログに記録することもできます。

クエリプランの省略形を要求ログに記録できます。プロシージャのロギングが有効になっている場合は、プロシージャ文のプランも記録されます。

次の出力は、**-zr all** オプションを指定してサーバを起動したときの要求ログから抜粋したものです。この例では、ユーザは `iqdemo` データベースに接続し、`sp_iqstatus` コマンドを実行しています。

各行にはカンマで区切られたいくつかのフィールドがあり、最初のフィールドは時間を示します。定期的に、タイムスタンプが次の形式で出力されます。

```
MMdd hhmmss.sss
0523 095954.807,[,1000000001,sp_iq_mpx_init,16,iq utilities status 1
```

例として `+13,C,1,UID=DBA` の行以降では、オフセットは前の行を起点とします。この場合 `+13` は、直前の行から 13 ミリ秒が経過したことを意味します。場合によっては、`=` は、直前の行から 0 ミリ秒が経過したことを意味します。

次に、要求ログの一部を示します。

```
0523 095954.807,[,1000000001,sp_iq_mpx_init,16,iq
utilities status 1
+2,[,1000000001,sp_iq_mpx_init,16
+1,[,1000000001,sp_iq_mpx_init,62,message STRING('IQ
Server ',@@servername,') to console
+2,[,1000000001,sp_iq_mpx_init,62
taj% pg iqdemo.sqllog
0523 095954.807,[,1000000001,sp_iq_mpx_init,16,iq
utilities status 1
+2,[,1000000001,sp_iq_mpx_init,16
+1,[,1000000001,sp_iq_mpx_init,62,message STRING('IQ
Server ',@@servername,') to console
+2,[,1000000001,sp_iq_mpx_init,62
0523 100510.344,<,1,CONNECT
+13,C,1,UID=DBA
+83,>,1,CONNECT,1
+1,<,1,PREPARE,SELECT @@version, if 'A'<>'a' then 1
else 0 endif, isnull(property('IsIQ'),'NO'),
isnull(connection_property('odbc_distinguish_char_and_
varchar'),'Off'),
isnull(connection_property('odbc_describe_binary_as_va
rbinary'),'Off'), connection_property('charset'),
db_property('charset')
+1,>,1,PREPARE,65536
=,<,1,EXEC,65536
+79,P,1,[S]DUMMY<seq>
=,>,1,EXEC
+1,<,1,DROP_STMT,65536
=,>,1,DROP_STMT
=,<,1,PREPARE,SET TEMPORARY OPTION time_format =
'hh:nn:ss',SET TEMPORARY OPTION timestamp_format =
```



```
'yyyy-mm-dd hh:nn:ss.ssssss';SET TEMPORARY OPTION
date_format = 'yyyy-mm-dd';SET TEMPORARY OPTION
date_order = 'ymd';SET TEMPORARY OPTION isolation_level
= 0;
+1,>,1,PREPARE,65537
+1,<,1,EXEC,65537
=,[1,*batch*,1,set temporary option time_format =
'hh:nn:ss'
+11,],1,*batch*,1
=,[1,*batch*,1,set temporary option timestamp_format =
'yyyy-mm-dd hh:nn:ss.ssssss'
+11,],1,*batch*,1
+1,[1,*batch*,1,set temporary option date_format =
'yyyy-mm-dd'
+11,],1,*batch*,1
=,[1,*batch*,1,set temporary option date_order = 'ymd'+11,],
1,*batch*,1
=,[1,*batch*,1,set temporary option isolation_level = 0
+11,],1,*batch*,1
=,>,1,EXEC
```

要求ログファイルの分析

ストアドプロシージャ **sa_get_request_profile** と **sa_get_request_times** を使用して、**-zr** ログファイルを読み込み、結果を要約します。

sa_get_request_profile は要求ログを分析して、類似する文の実行時間を特定し、グローバルテナンソラリテーブル **satmp_request_profile** に結果を要約します。次に例を示します。

```
call sa_get_request_profile('/sys1/users/jones/iqreqs1_zr.log');
select * from satmp_request_profile;
```

sa_get_request_times も要求ログを分析して、文の実行時間を特定し、グローバルテナンソラリテーブル **satmp_request_time** に結果を要約します。次に例を示します。

```
call sa_get_request_times('/sys1/users/jones/iqreqs1_zr.log');
select * from satmp_request_time;
```

要求レベルログの詳細については、『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」の「start_iq-zo スイッチ」、『リファレンス：ビルディングブロック、テーブル、およびプロシージャ』の「**sa_server_option** システムプロシージャ」を参照してください。

要求ロギング

要求ロギングは、アプリケーションから受け取った要求と、アプリケーションに送られた応答のログを個別に記録します。データベースサーバがアプリケーションに何を要求されているかを特定したい場合に最も役立ちます。

特定のアプリケーションのパフォーマンスを分析するときに、データベースサーバとクライアントのどちらに原因があるか不明な場合は、要求ロギングから始め

ることもおすすめします。要求ロギングは、問題の原因となっている可能性の高い、データベースサーバに対する特定の要求を特定するためにも使用できます。

注意： 要求ロギング機能で提供されるすべての機能とデータは、診断トレーシングを使用した場合も利用できます。診断トレーシングでは、それ以外の機能やデータも提供されます。

ログに取られる情報には、タイムスタンプ、接続 ID、要求タイプなどがあります。クエリについては、独立性レベル、フェッチされたローの数、カーソルタイプもあります。INSERT 文、UPDATE 文、DELETE 文については、対象のロー数と実行されたトリガ数もあります。

注意： 要求ログには、難読化された機密情報を持つすべての文が含まれます。機密情報が難読化されていない唯一のケースは、文に解析エラーがある場合です。

-zr サーバオプションを使用すると、データベースサーバの起動時に要求ロギングをオンにできます。-zo サーバオプションを使用すると、出力を要求ログファイルにリダイレクトして、さらに分析を進めることができます。-zn オプションと -zs オプションを使用すると、保存する要求ログファイルの数と要求ログファイルの最大サイズを指定できます。

sa_get_request_times システムプロシージャは、要求ログを読み込み、ログの文とその実行時間をグローバルテナポラリテーブル (satmp_request_time) に格納します。INSERT、UPDATE、DELETE の各文の場合は、記録される時間は、文が実行された時間です。クエリの場合、記録された時間は、PREPARE から DROP (DESCRIBE/OPEN/FETCH/CLOSE) までの合計所要時間です。したがって、オープンカーソルには注意する必要があります。

改善候補の文について satmp_request_time を分析します。低コストでも頻繁に実行される文が、パフォーマンスの問題を引き起こしている可能性があります。

sa_get_request_profile を使用して sa_get_request_times を呼び出し、satmp_request_time を satmp_request_profile という別のグローバルテナポラリテーブルにまとめることができます。また、このプロシージャは、文をグループ化し、呼び出しの回数、実行時間などの情報を提供します。

警告！

Perl スクリプト tracetime.pl を使用してログを分析する場合は、要求ログがキャプチャされる間、max_client_statements_cached オプションを 0 に設定してクライアント文のキャッシュを無効にしておく必要があります。

例

要求ログへの出力は、sa_server_option システムプロシージャを使用してフィルタして、特定の接続やデータベースからの要求のみを含めるようにできます。これ

により、アクティブな接続の多いデータベースサーバや複数のデータベースのあるデータベースサーバをモニタリングするときのログサイズを縮小できます。

- **接続に基づくフィルタ** – 次の構文を使用します。

```
CALL sa_server_option( 'RequestFilterConn' , connection-id );
```

connection-id は `CALL sa_conn_info()` を実行して取得できます。

- **データベースに基づくフィルタ** – 次の構文を使用します。

```
CALL sa_server_option( 'RequestFilterDB' , database-id );
```

database-id は、データベースに接続している場合に `SELECT`

`CONNECTION_PROPERTY('DBNumber')` を実行して入手できます。フィルタは、明示的にリセットするまで、またはデータベースサーバが停止されるまで有効です。

- **フィルタのリセット** – 次の2つの文のいずれかを使用して、接続またはデータベースごとにフィルタをリセットします。

```
CALL sa_server_option( 'RequestFilterConn' , -1 );
```

```
CALL sa_server_option( 'RequestFilterDB' , -1 );
```

- **要求ログへのホスト変数の出力** – ホスト変数の値を要求ログに含めるには、次の操作を行います。

- `-zr` サーバオプションに値 `hostvars` を指定します。

- 次の文を実行します。

```
CALL sa_server_option( 'RequestLogging' , 'hostvars' );
```

要求ログ分析プロシージャ `sa_get_request_times` は、ログ中のホスト変数を認識し、それらをグローバルテンポラリテーブル `satmp_request_hostvar` に追加します。

診断情報を収集するための接続

データベースオプション `DEDICATED_TASK` を使用すると、DBA は要求処理タスクが1つの接続からの要求だけを処理するように指定できます。

この事前に確立された接続を使用して、他の接続に応答しなくなったデータベースサーバの状態についての情報を収集できます。『リファレンス：文とオプション』の「**DEDICATED_TASK** オプション」を参照してください。

通信の問題の診断

サーバで起動時に通信エラーが返される場合は、サーバを起動するときに `-z` コマンドラインオプションを設定します。

このスイッチは、サーバ起動時の通信リンクについての診断情報を提供します。情報は、サーバが起動した標準出力と `srvlog` ファイルに記録されます。

サポートセンタへの問題の報告

マニュアルまたはオンラインヘルプでは解決できない問題があった場合には、担当の方を通して SAP のサポートセンタまでご連絡ください。

SAP ソフトウェアがインストールされているサイトには、SAP 製品の保守契約を結んでいるサポートセンタとの連絡担当の方 (コンタクトパーソン) を決めてあります。

サポートセンタでは問題を解決するために、ご使用の SAP Sybase IQ 環境についての情報が必要となります。

SAP Control Center に関して問題がある場合は、SCC またはサイト <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680> の SAP Control Center for SAP Sybase IQ のオンラインヘルプを参照。

getiqinfo を使った診断情報の収集

SAP Sybase IQ には、SAP 製品の保守契約を結んでいるサポートセンタが問題を診断するために必要な情報を収集するスクリプトが用意されています。

getiqinfo スクリプトは、オペレーティングシステム環境、SAP Sybase IQ 環境、ログファイルについての情報を収集します。

SAP サポートセンタに問題を報告する前に、このスクリプトを実行してください。それにより、サポートセンタでは問題をより迅速に解決できるようになり、ユーザ側の手間も少なくなります。

getiqinfo スクリプトは、SAP Sybase IQ インストール環境のトラブルシューティング用には設計されておらず、オンサイトのトラブルシューティング機能は備えていません。このスクリプトは、SAP Sybase IQ 環境が正しく設定されており、サーバが実行されている場合にのみ正常に実行されます。

getiqinfo を実行する前に

getiqinfo スクリプトを実行する前に、情報を収集します。

スクリプトを実行する前に、次の内容を確認します。

- データベースファイルのロケーション
- サーバの起動時に設定ファイルを使用している場合は、設定ファイルのフルパス
- SAP Sybase IQ メッセージファイルの名前を変更している場合は、.iqmsg ファイルのフルパス

可能であれば、SAP Sybase IQ サーバを実行したままにするか、**getiqinfo** を実行する前にサーバを起動します。これにより、SAP Sybase IQ の実行中にしか取得でき

ないデータベースの内部データをスクリプトで収集できるようになります。スクリプトがサーバを自動的に起動することはありません。

スクリプトは、SAP Sybase IQ サーバの起動に使用される設定と同じ環境設定で実行されます。**getiqinfo** は、いくつかの IQ 固有の環境変数を使用してファイルを検索します。

スクリプトが収集したデータは、現在のディレクトリ (プログラムを起動したディレクトリ) に保存されます。このディレクトリに十分な領域があることを確認してください。スクリプトはロケーション変更用のプロンプトを表示しませんが、スクリプトを変更して変数 *DEST_DIR* を再設定すれば出力ロケーションを変更できます。

getiqinfo スクリプトの実行

UNIX プラットフォームでは、**getiqinfo** はシェルスクリプトです。Windows プラットフォームでは、`getiqinfo.bat` は `IQ-16_0¥bin64` ディレクトリにあるバッチスクリプトです。**getiqinfo** スクリプトは IQ サーバとともにインストールされますが、ネットワーククライアントでは使用できません。

手順は UNIX プラットフォームと Windows プラットフォームで異なります。

1. プラットフォームに応じた方法でスクリプトを起動します。
 - UNIX コマンドプロンプトで、`IQ-16_0/bin64` ディレクトリに次のように入力します。


```
getiqinfo.sh
```
 - Windows では、[スタート]>[実行]を選択し、`<install_path>¥IQ-16_0¥bin64¥getiqinfo.bat` と入力します。
2. 入力するように求められたら、次の内容を入力します。
 - データベースファイルのディレクトリ。これは、`.iqmsg` ファイルと、UNIX では `stktrc*.iq` ファイルのデフォルトのロケーションでもあります。
 - データベースファイルのベース名 (`.db` サフィックスが付かないファイル名)。これは `.iqmsg` ファイルのデフォルトのベース名でもあります。
 - これらのファイルを検索するその他のディレクトリ。
 - このデータベースサーバの SAP Sybase IQ エンジン名 (サーバ名) とポート番号。
 - 次のいずれかが付与されたユーザのユーザ ID とパスワード。
 - DROP CONNECTION システム権限
 - MONITOR システム権限
 - SERVER OPERATOR システム権限

トラブルシューティングのヒント

- SAP Sybase IQ サーバの起動時に設定ファイルを使用した場合は、設定ファイルのフルパス。
- **-zo** サーバオプションを指定した場合は、出力ファイルのフルパス。

リストされたファイルを SAP Sybase 製品の保守契約を結んでいるサポートセンタに送付するように指示するメッセージも表示されます。

getiqinfo によって収集される情報

getiqinfo スクリプトによって情報が収集されます。

- ハードウェアの種類、メモリ容量、CPU タイプ、速度、CPU の数
- オペレーティングシステム (Sun Solaris 2.10 など)
- スワップ領域サイズ
- SAP Sybase IQ のバージョンと EBF レベル、Anywhere のバージョン
- 問題が発生した日付と時刻のスタックトレースファイル。データベースサーバを起動したディレクトリに `stktrc-YYYYMMDD-HHMMSS_#.iq` という名前で保存される (UNIX および Linux プラットフォームのみ)。
- エラーが発生したコマンドまたはクエリ
- メッセージログファイル。 `dbname.iqmsg` という名前で、デフォルトではデータベースサーバを起動したディレクトリに保存される。
- クエリプラン (`.iqmsg` ファイルに記録される。この後の「注意」を参照)
- サーバログ
 - UNIX では、 `IQ-16_0/logfiles/<servername>.000n.stderr` および `IQ-16_0/logfiles/<servername>.000n.srvlog` である。
 - Windows プラットフォームでは、必要に応じてサーバを再起動し、コンソールウィンドウのコピーを手動で収集する必要がある。
- 設定ファイル (デフォルトでは `dbname.cfg`) の起動オプションと接続オプションの設定
- データベースオプションの設定と **sa_conn_properties** からの出力 (サーバがまだ動作している場合)

次の情報は、**getiqinfo** では収集されませんが、サポートセンタから求められることがあります。

- 使用している接続プロトコル (ODBC、JDBC、TDS など)
- Open Client のバージョン
- 設定タイプ (シングルユーザかマルチユーザか)
- 使用しているフロントエンドツール (Brio Query など)
- データベースのスキーマとインデックス
- **sp_iqcheckdb** プロシージャからの出力

以下のオプションが設定されている場合は、**getiqinfo** によってクエリプランの詳細が自動的に収集されます。オプションを設定し、エラーが発生したコマンドを再実行することによって、この情報を手動で収集することもできます。

```
SET TEMPORARY OPTION QUERY_PLAN = 'ON'
```

```
SET TEMPORARY OPTION QUERY_DETAIL = 'ON'
```

クエリプランはメッセージログファイルにあります。これらのオプションのデフォルト値は `QUERY_PLAN = ON` および `QUERY_DETAIL = OFF` です。

パフォーマンスの問題がある場合は、次のオプションを設定してください。

```
SET TEMPORARY OPTION QUERY_PLAN_AFTER_RUN = 'ON'
```

このオプションを設定すると、Sybase 製品の保守契約を結んでいるサポートセンタでは、クエリ処理のどのステップに時間がかかっているのか判断できます。

.srvlog ファイルと .iqmsg ファイル間での接続情報の関連付け

サポートセンタから、設定ファイルで `start_iq` コマンドの `-zr` オプションを設定するように求められることがあります。

このサーバ起動オプションは、サーバに送信される文を追跡するように要求ログレベルを設定します。パラメータは `ALL`、`NONE`、または `SQL` です。このオプションを設定すると、サーバ名に `.srvlog` というサフィックスが付いたログファイルが生成されます。

SAP Sybase IQ のメッセージファイル `.iqmsg` では、サーバへの各接続は接続ハンドルによって識別されます。`.iqmsg` メッセージファイルには、エラー、警告、トレース情報が接続ごとに記録されます。

`.srvlog` ファイルと `.iqmsg` ファイル内の接続識別子を相互に関連付けて、関連情報を見つける手順は次のとおりです。

1. `.iqmsg` ファイルで、目的の接続を探します。次に例を示します。

```
Connect: SA connHandle: 1000000061
```

次の行は、この接続の `iqmsg` ログファイルの内容を示しています。

```
16:14:59. 0000000062 Connect: SA connHandle: 1000000061
SA connID: 31 IQ connID: 0000000062 User: DBA
03/17 16:15:00. 0000000062 Cmt 12064
03/17 16:15:00. 0000000062 PostCmt 0
03/17 16:15:00. 0000000000 Disconnect: SA connHandle: 1000000061
SA connID: 31 IQ connID: 0000000062 User: DBA
```

2. `.srvlog` ファイルで、`.iqmsg` ファイルの "SA connHandle" の後にある番号を検索して、この接続のすべての行を抽出します。

たとえば、`.srvlog` ファイルで "1000000061" を検索します。

トラブルシューティングのヒント

```
16:14:59. [,1000000061,sp_iqdbspace,48,select
str_replace(dbpaceName,' ','') into dbpaceName_literal
03/17 16:14:59. P,1000000061,[S][0]DUMMY<seq>
03/17 16:14:59. ],1000000061,sp_iqdbspace,48
03/17 16:14:59. P,1000000061,[1]ISYSIQDBFILE<seq> JNL
dbf<ISYSDBFILE>
JNL ISYSDBSPACE<ISYSDBSPACE>
03/17 16:14:59. [,1000000061,sp_iqdbspace,58,execute immediate
with
quotes on 'iq utilities main into iq_dbpace_temp dbpace info
' || dbpaceName
03/17 16:14:59. P,1000000061,[S]INSERT ROWS
03/17 16:14:59. P,1000000061,[S]INSERT ROWS
03/17 16:14:59. P,1000000061,[S]INSERT ROWS
03/17 16:14:59. P,1000000061,[S]INSERT ROWS03/17 16:14:59. P,
1000000061,[S]INSERT ROWS
03/17 16:14:59. ],1000000061,sp_iqdbspace,58
03/17 16:14:59. [,1000000061,sp_iqdbspace,60,select
d.dbpace_name as DBSpaceName, min(SegType) as DBSpaceType,...
03/17 16:15:00. ],1000000061,sp_iqdbspace,60
03/17 16:15:00. P,1000000061,Work[ Sort[ GrByH[ dbf<seq> JNL
ISYSIQDBSPACE<ISYSIQDBSPACE> JNL ISYSDBSPACE<ISYSDBSPACE> JH*
iq_dbpace_temp<seq> ] ] ] : ISYSIQPARTITIONCOLUMN<seq> :
idx<seq> : tab<seq>
03/17 16:15:00. [,1000000061,sp_iqdbspace,105,drop table
dbo.iq_dbpace_temp
03/17 16:15:00. ],1000000061,sp_iqdbspace,105
03/17 16:15:00. P,1000000061,[1]Work[ Sort[ sp_iqdbpace<call> ] ]
```

この例の接続ハンドルは 1000000061 です。

サポート Web サイト

問題を解決できない場合は、SAP Sybase IQ オンラインサポート Web サイトの MySybase で追加の情報を得られることがあります。

MySybase では、必要に応じてカスタマイズしたビューを使用して、詳細なサポートケース、最新のソフトウェア情報、解決済みや既知の問題などを検索できます。オンラインで [Technical Support] を開くこともできます。

MySybase は、ほとんどのインターネットブラウザで表示できます。この無料サービスに登録し、サービスを利用する方法を参照するには、Web ブラウザで Technical Documents at <http://www.sybase.com/support/techdocs/> を指定し、[MySybase] をクリックします。

チェックリスト： サポートセンタに提出する情報

getiqinfo スクリプトを実行すると、情報を収集できます。

要求される情報	値
SAP Sybase IQ のバージョン (16.0 GA や SP 番号など)	
sp_iqlmconfig の出力	
ハードウェアの種類	
メモリ容量	
CPU の数	
オペレーティングシステム名とバージョン (Microsoft Windows 2008 Service Pack 1 など)	
オペレーティングシステムのパッチレベル	
使用しているフロントエンドツール (Business Objects Crystal Reports など)	
使用している接続プロトコル (ODBC、JDBC、TDS など)	
Open Client のバージョン	
設定タイプ (シングルノードかマルチプレックスか)	
メッセージログファイル (dbname.iqmsg)	
サーバログファイル (server.nnnn.srvlog と server.nnnn.stderr)	
スタックトレースファイル (stktrc-YYYYMMDD-HHNNSS_#.iq)	
エラーが発生したコマンドまたはクエリ	
起動オプション設定	
接続オプション設定	
データベースオプション設定	
データベースのスキーマとインデックス	
sp_iqstatus の出力	
クエリプラン： オプション (Query_Plan、Query_Detail、Query_Plan_After_Run、Query_Plan_As_Html、Query_Plan_As_Html_Directory、Query_Timing) を設定し、コマンドまたはクエリを再実行	
問題のスクリーンショット (可能な場合)	

トラブルシューティングのヒント

付録：接続パラメータと通信パラメータリファレンス

接続パラメータは、クライアントアプリケーションからデータベースへの接続を確立したり、記述したりします。

接続パラメータ

接続パラメータは、接続文字列に含めます。

接続パラメータは、次に含めることができます。

- アプリケーションの接続文字列
- ODBC データソース
- [Sybase IQ 接続] ダイアログボックス

パラメータ名の後に、短い形式の名前をカッコで囲んで示します。短い形式の名前は、接続コマンド内で省略形として使用できます。

ODBC 設定ダイアログと Windows オペレーティングシステム用の Sybase IQ 接続ダイアログの書式は共通です。一部のパラメータは、これらのウィンドウのチェックボックスやフィールドに対応しています。その他のパラメータは、[詳細] タブにあるテキストボックスに入力できます。

使用法についての注釈

接続パラメータでは、大文字と小文字を区別しません。

各接続パラメータの使用法は、パラメータが使用される状況を記述します。一般的な使用法は次のとおりです。

- **組み込みデータベース** – SAP Sybase IQ を組み込みデータベースとして使用するときは、接続時にサーバが起動され、データベースがロードされます。アプリケーションがデータベースから切断されると、データベースはアンロードされ、サーバが停止します。
- **ネットワークサーバ** – SAP Sybase IQ をネットワークサーバとして使用するときは、ネットワーク上で既に動作中のサーバをクライアントアプリケーションが検索し、データベースに接続します。

dbping ユーティリティを使用して、接続文字列をテストできます。例については、『ユーティリティガイド』の「dbping データベース管理ユーティリティ」を参照してください。

ブール (true または false) 引数は、true の場合は YES、ON、1、TRUE のいずれか、false の場合は NO、OFF、0、FALSE のいずれかになります。

インタフェースライブラリで使用されている接続パラメータは、次の場所から取得できます (優先度の高い順)。

- 接続文字列
- SQLCONNECT 環境変数
- データソース

サーバ名は、ASCII 文字セットの 1～127 の範囲の文字で構成してください。他のパラメータには、このような制限はありません。

次の規則によって、パラメータの優先度が決まります。

- 接続文字列のエントリは、左から右に読み込まれます。同じパラメータを複数指定すると、文字列の最後に指定したパラメータが適用されます。
- 文字列に DSN や FILEDSN のエントリが含まれている場合は、プロファイルが設定ファイルから読み込まれ、設定ファイル内のエントリが使用されます (そのエントリがまだ設定されていない場合)。たとえば、接続文字列にデータソース名が含まれており、データソースに含まれるいくつかのパラメータを接続文字列が明示的に設定する場合、競合が発生すると、明示的なパラメータが使用されます。

AppInfo 接続パラメータ [Appinfo]

管理者が、データベースサーバから特定のクライアント接続のオリジンを識別するのに役立ちます。

使用法

すべての場所

デフォルト値

空の文字列

説明

この接続パラメータは、Embedded SQL クライアント、ODBC クライアント、OLE DB クライアント、UNIX 上の **DBISQLC** からデータベースサーバに送信されます。Open Client からは提供されません。

このパラメータは、クライアントマシンの IP アドレス、オペレーティングシステムなど、クライアントプロセスに関する情報を含む、生成された文字列で構成されます。この文字列は、その接続のデータベースサーバに関連付けられ、次の文を使用して取得できます。

```
SELECT connection_property( 'AppInfo' )
```

クライアントは、この生成される文字列に付加する形で、独自の文字列を指定できます。AppInfo プロパティ文字列は、セミコロンで区切られた **key=value** ペアのシーケンスです。有効なキーは次のとおりです。

- **API** – DBLIB、ODBC、OLEDB、または ADO.NET (iAnywhere JDBC ドライバでは ODBC が戻される)
- **APPINFO** – 接続文字列に AppInfo を指定したときに入力される文字列
- **EXE** – クライアントの実行ファイルの名前 (Windows のみ)
- **HOST** – クライアントマシンのホスト名
- **IP** – クライアントの IP アドレス
- **OS** – オペレーティングシステムの名前とバージョン番号 (例：Sun Solaris 2.9)
- **PID** – クライアントのプロセス ID
- **THREAD** – クライアントのスレッド ID
- **VERSION** – 使用する接続プロトコルのバージョン (メジャー番号、マイナー番号、ビルド番号を含む) (例：9.0.1.1549)
- **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分数

クライアント接続パラメータでデバッグログファイルを指定した場合は、APPINFO の文字列がデバッグログファイルに追加されます。

例

- Interactive SQL からデモデータベースに接続します (デフォルトで iAnywhere JDBC ドライバを使用します)。

```
dbisql nogui -c "uid=DBA;pwd=sql" -host MachineName -port 1234
```

次の方法で、現在のアプリケーション情報を表示します。

```
SELECT connection_property('AppInfo')
```

結果は次のようになります (1 つの文字列です)。

```
IP=12.345.67.899;
HOST=MachineName;
OSUSER=sybase;
OS='SunOS 5.10 Generic_144489_04';
EXE='/Sybase/IQ16/shared/JRE-7_0_1/bin/amd64/java';
PID=0x52af;
THREAD=0x14;
VERSION=16.0.0.290;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-240
```

- Interactive SQL から、AppInfo プロパティに独自の情報を追加して、デフォルトデータベースに接続します。

```
isql -U DBA -P sql -S MachineName:1234 --appname MyISQL
```

付録：接続パラメータと通信パラメータリファレンス

次の方法で、現在のアプリケーション情報を表示します。

```
SELECT connection_property('AppInfo')
```

結果は次のようになります (1 つの文字列です)。

```
HOST=MachineName;  
PID=21155;  
EXE=MyISQL
```

AutoPreCommit 接続パラメータ [AutoPreCommit]

実行前に各文を強制的にコミットします。

使用法
ODBC

デフォルト値
No

説明
デフォルトでは、各文は実行後にコミットを発行します。AutoPreCommit が Yes の場合、各 select 文の前に commit 文が発行されるため、ユーザは常にすべてのデータベースオブジェクトの最新バージョンを参照できます。

例
AutoPreCommit オプションを Yes に設定すると実行前のコミットが有効になり、No に設定すると無効になります。このオプションは、.odbc.ini ファイル、または [接続] ダイアログの [詳細] タブで設定します。

次の例では、各文を実行前にコミットします。

```
[Sample DSN]  
DatabaseFile=c:\Program Files\Sybase\IQ-16_0\demo\iqdemo.db  
AutoPreCommit=Y  
UserID=DBA  
Password=SQL
```

AutoStart 接続パラメータ [Astart]

接続が検出されない場合は、データベースサーバを起動しないようにします。

使用法
すべての場所

デフォルト値
Yes

説明

デフォルトでは、接続中にサーバが検出されなくてデータベースファイルが指定されている場合、同一のマシンで何らかのデータベースサーバが起動されます。この動作が起こらないようにするには、接続文字列の `AutoStart` パラメータを `Off` に設定します。

例

次のデータソースフラグメントは、ネットワークサーバがないときにデータベースサーバを起動しないようにします。

```
[My Demo Database]
DatabaseFile=c:\¥sybase¥IQ-16_0¥demo¥iqdemo.db
Autostart=No
UserID=DBA
ENG=network_server
```

AutoStop 接続パラメータ [Astop]

オープン接続がなくなった場合でも、すぐにデータベースをアンロードしないようにします。

使用法

埋め込みデータベース

デフォルト値

Yes

説明

デフォルトでは、接続文字列で起動したすべてのサーバは、接続がなくなると停止します。また、接続文字列からロードしたすべてのデータベースも、接続がなくなるとすぐにアンロードされます。この動作は、`Autostop=Yes` と設定した場合と同じです。

`Autostop=No` に設定すると、その接続で起動したすべてのデータベースは、接続がなくなってもアンロードされません。したがって、データベースサーバも停止しません。

現在実行していないデータベースに接続するときのみ、`AutoStop` パラメータを使用します。データベースがすでにロードされている場合は、パラメータは無視されます。

例

次の Windows の接続プロファイルは、接続が削除されたときにデータベースをアンロードしないようにします。

```
[Sample Embedded Database]
DatabaseFile=c:\sybase\IQ-16_0\demo\iqdemo.db
Autostop=No
UserID=DBA
```

CharSet 接続パラメータ [CS]

この接続で使用する文字セットを指定します。

使用法

すべての場所

デフォルト値

ロケール文字セット

説明

CharSet に値を指定すると、指定された文字セットが現在の接続に使用されます。CharSet=none に設定すると、接続の文字セット変換が無効になります。

データのアンロード時に、CharSet 接続パラメータを使用して、文字セットを指定できます。

非可逆文字セット変換を回避するために、Unicode クライアント API の使用時に CharSet 接続パラメータを設定しないことをおすすめします。Unicode クライアント API には、ADO.NET、OLE DB、および iAnywhere JDBC ドライバがあります。ODBC も、ワイド (Unicode) 関数が使用されている場合は、Unicode クライアント API です。

CompressionThreshold (COMPTH) 接続パラメータ

パケットの圧縮が適用される最小パケットサイズを増減します。

構文

```
{ CompressionThreshold | COMPTH }=size[ k ]
```

使用法

TDS 以外。他には特に制限なし。圧縮接続にのみ適用。

指定可能な値

- **size** – この整数は、パケットの圧縮が適用される最小パケットサイズを指定します。デフォルト値はバイト単位ですが、k を使用してキロバイトの単位を指定できます。クライアントとデータベースサーバで圧縮スレッシュホールドの設定が異なる場合は、クライアントの設定が適用されます。サポートされている最

小値は 1 バイト、最大値は 32767 バイトです。80 バイト未満の値はおすすめしません。

デフォルト
120

CompressionThreshold 値が設定されていない場合、圧縮スレッシュホールド値は、サーバ側の設定によって制御されます。デフォルトは 120 バイトです。

備考

圧縮した場合に転送速度が速くなるパケットのみを圧縮するように圧縮スレッシュホールドを変更して、圧縮接続のパフォーマンスを向上できます。

圧縮が有効な場合、パケットは、各々のサイズに応じて圧縮するかどうかを決定します。たとえば、SAP Sybase IQ では、圧縮のスレッシュホールドよりも小さいパケットは、通信の圧縮が有効な場合でも圧縮されません。同様に、小さなパケット (100 バイト未満) は、通常はまったく圧縮されません。パケットの圧縮には CPU 時間が必要なので、小さなパケットを圧縮しようとする、実際にパフォーマンスが低下することがあります。

圧縮スレッシュホールドの値を小さくすると、非常に低速なネットワークではパフォーマンスが向上し、値を大きくすると CPU 使用率の減少によってパフォーマンスが向上する場合があります。ただし、圧縮のスレッシュホールド値を小さくするとクライアントとサーバの両方で CPU 使用率が増加するので、パフォーマンス分析を行って、圧縮のスレッシュホールドを変更することでパフォーマンスが向上するかどうかを判断してください。

例

圧縮スレッシュホールド値を 100 バイトに設定して接続します。

```
CompressionThreshold=100
```

CommBufferSize 接続パラメータ [CBSIZE]

通信パケットの最大サイズをバイト単位で設定します。キロバイト単位で指定するには K を使用します。

構文

```
{ CommBufferSize | CBSIZE }=size[ k ]
```

使用法

すべての場所

指定可能な値

- **size** – 通信パケットの最大サイズを指定します。デフォルト値はバイト単位ですが、k を使用してキロバイトの単位を指定できます。CommBufferSize の最小値は 500 バイトで、最大値は 65535 バイトです。

デフォルト値

CommBufferSize 値を設定しないと、サーバ側の設定 (デフォルトは 7300 バイト) によって CommBufferSize が制御されます。

備考

ネットワーク上のパケットの最大サイズは、プロトコルスタックによって設定されます。CommBufferSize をネットワークで許可されているサイズより大きく設定すると、通信パケットがネットワークソフトウェアによって分割されます。デフォルトのサイズは、標準 Ethernet TCP/IP の最大パケットサイズ (1460 バイト) の倍数です。

パケットサイズを大きくすると、複数のローのフェッチと長いローのフェッチのパフォーマンスが向上しますが、クライアントとサーバのメモリ使用量が増加します。

クライアント側で CommBufferSize の指定がないと、接続ではサーバのバッファサイズが使用されます。クライアント側で CommBufferSize の指定がある場合、接続ではクライアント側の CommBufferSize 値が使用されます。

-p データベースサーバオプションを使用して CommBufferSize を設定すると、CommBufferSize を指定していないすべてのクライアントで -p データベースサーバオプションで指定されたサイズが使用されます。

例

次の文では、バッファサイズを 1460 バイトに設定します。

```
...  
CommBufferSize=1460  
...
```

別の方法として、接続ウィンドウの [ネットワーク] タブの [Buffer size] テキストボックスに値を入力して、このパラメータを設定することもできます。

CommBufferSpace 接続パラメータ [CBSpace]

起動時にネットワークバッファに割り付ける領域の量を、キロバイト単位で指定します。

使用法

すべての場所

値
整数

デフォルト値
10

説明

CommBufferSpace はグローバル設定で、すべての接続に適用されます。

例

次のプロファイルフラグメントは、起動時にネットワークバッファとして 200KB を割り付けるようにネットワークライブラリに指示します。

```
...  
CBSpace=200  
...
```

接続ウィンドウの[ネットワーク]タブの[Buffer space]テキストボックスに値を入力して、このパラメータを設定することもできます。

CommLinks 接続パラメータ [Links]

クライアント側のネットワーク通信リンクを指定します。

使用法

すべての場所

値

文字列

デフォルト値

接続には共有メモリ通信リンクのみを使用します。

説明

CommLinks=ALL と指定すると、クライアントは、使用可能なすべての通信プロトコルを使用してサーバを検索します。CommLinks=ALL を指定するとパフォーマンスに影響する場合がありますため、この設定は使用するプロトコルが不明なときにのみ使用してください。

CommLinks (LINKS) 接続パラメータに 1 つ以上のプロトコルを指定した場合は、クライアントはそれらの通信プロトコルを指定された順に使用してネットワークデータベースサーバを検索します。指定されたプロトコルを使用した接続試行が失敗した場合は、他に試すべきプロトコルが残っている場合でも、接続エラーが発生し、接続試行がアボートします。

CommLinks (LINKS) 接続パラメータを指定しない場合、クライアントは現在のマシン上で、共有メモリ接続のみを使用してサーバを検索します。これはデフォルトサーバであり、CommLinks=ShMem と指定した場合と同じになります。共有メモリプロトコルは、同じマシン上の同じオペレーティングシステムで実行されているクライアントとサーバの通信に使用されます。

次に、CommLinks パラメータの使用可能な値を説明します。

- **SharedMemory (ShMem)** – 同一マシン通信のための共有メモリプロトコルを起動します。これがデフォルト設定です。クライアントは、プロトコルリストに共有メモリプロトコルがある場合は、リスト内のプロトコルの順序に関係なく、まず共有メモリプロトコルを試します。
- **ALL** – 最初に共有メモリプロトコルを使用して接続を試行した後に、残りすべての使用可能な通信プロトコルを使用します。使用する通信プロトコルが不明の場合は、この設定を使用してください。
- **TCP/IP** – TCP/IP 通信リンクを起動します。TCP/IP は、すべてのオペレーティングシステムでサポートされています。

これらの値には、それぞれ追加のネットワーク通信パラメータを指定できます。

次のような場合には、**ALL** の代わりに特定のプロトコルを使用できます。

- 不要なネットワークリンクを起動しなければ、ネットワークライブラリの起動を少し高速化できる。
- データベースへの接続を高速化できる。
- 追加のネットワーク通信パラメータを指定して特定のプロトコルのブロードキャスト動作を調整するために、明示的にリンクを指定する必要がある。

追加のネットワーク通信パラメータを各リンクに指定すると、リンクのブロードキャスト動作を調整できます。

CommLinks パラメータは、データベースサーバの `-x iqsrv16` サーバオプションに対応します。デフォルトでは、ネットワークサーバは使用可能なすべてのプロトコルを起動します (`-x ALL` を指定した場合と同じ)。

例

- 次の接続文字列フラグメントでは、TCP/IP プロトコルのみを起動します。

```
CommLinks=tcPIP
```
- 次の接続文字列フラグメントは、共有メモリプロトコルを起動し、共有メモリプロトコルを通じてデータベースサーバを検索します。検索が失敗した場合は、TCP/IP ポートを起動し、直接接続されている TCP/IP ネットワーク上でサーバとホスト **kangaroo** を検索します。

```
CommLinks=tcPIP (HOST=kangaroo) , shmem
```

ConnectionName 接続パラメータ [CON]

接続に名前を付け、マルチ接続アプリケーションで簡単に切り替えができるようにします。

使用法

ODBC には使用不可

デフォルト値

接続名なし

説明

確立中の特定の接続に名前を付けるオプションのパラメータです。複数の接続を確立しても切り替えを行わないときは、このパラメータを指定する必要はありません。

接続名はデータソース名とは異なります。

例

FirstCon という名前の接続に接続します。

```
CON=FirstCon
```

ConnectionPool 接続パラメータ [CPOOL]

クライアントの接続プールの動作を制御します。

構文

```
ConnectionPool = { NO | YES  
  [ ( [ Timeout=timeout-sec; ] [ MaxCached=max-cached-conn ] ) ] }
```

使用法

非スレッド化 UNIX クライアントを除くすべてのプラットフォーム。

許容値

- **timeout-sec** – 接続プールのアイドルタイムアウト時間 (秒単位)。デフォルト値は 60 秒です。キャッシュされた接続は、timeout-sec で指定された時間内に再利用しないと、再利用できなくなります。
- **max-cached-conn** – 各アプリケーションのキャッシュされた接続の最大数。デフォルトの接続数は 5 です。接続が切断され、max-cached-conn で指定された最大接続数に達していない場合は、接続がキャッシュされます。接続が再初期化されます。接続はアプリケーションによって切断されますが、キャッシュされた接続はデータベースサーバに接続したままになります。

デフォルト

YES

備考

接続プールによって、データベースサーバに対して複数の短時間接続を行うアプリケーションのパフォーマンスが向上する可能性があります。接続が切断されると自動的にキャッシュされ、アプリケーションが再接続した時点で再利用できます。接続をプールするには、接続名は異なってもかまいませんが、他のすべての接続パラメータは同じである必要があります。

DatabaseFile 接続パラメータ [DBF]

データベースの起動時に、ロードまたは接続に使用するターゲットデータベースファイルを指定します。

すでに実行中のデータベースに接続する場合は、**DatabaseName(DSN)** パラメータを使用します。

特別な状況における接続では、SAP Sybase IQ は、DBF パラメータとデータベースファイル名を必要とする場合があります。

使用法

埋め込みデータベース

値

String

デフォルト値

なし

説明

DBF は、データベースサーバ上でまだ実行されていない特定のデータベースファイルをロードし、そのデータベースに接続します。

- DatabaseFile パラメータと同じ名前であっても .db 拡張子のない名前でもデータベースがロードされた場合は、データベースファイルではなく、そのデータベースに対して接続が確立されます。
- ファイル名に拡張子が含まれていない場合は、.db のファイルが検索されません。
- ファイルのパスは、データベースサーバの作業ディレクトリの相対パスです。サーバをコマンドプロンプトから起動すると、コマンドを入力するときの(現在の)ディレクトリが作業ディレクトリになります。サーバをアイコンかショートカットから起動すると、アイコンかショートカットで指定されたディ

レクトリが作業ディレクトリになります。完全なパスとファイル名を指定することをおすすめします。

- データベースファイルとデータベース名の両方を指定した場合、データベースファイルは無視され、実行中のデータベースへの接続には使用されません。

UNC ファイル名も使用できます。

警告！ データベースファイルは、データベースサーバと同じマシン上にある必要があります。ネットワークドライブにあるデータベースファイルを操作すると、ファイルが破損することがあります。

例

デモデータベース (Windows の場合は C:\Program Files\Sybase\IQ-16_0\demo ディレクトリにインストールされています) をロードして接続するには、次の DBF パラメータを使用します。

```
DBF=C:\Program Files\Sybase\IQ-16_0\demo\iqdemo.db
```

DatabaseName 接続パラメータ [DBN]

接続先となるロード済みデータベースを指定します。すでに実行されているデータベースに接続する場合に使用します。

まだ実行されていないデータベースに接続する場合は、**DatabaseFile (DBF)** パラメータを使用します。

使用法

実行中のネットワークサーバ

値

String

デフォルト値

なし

説明

サーバ上でデータベースが起動している場合は、データベース名が割り当てられています (管理者が `-n` オプションを使用して割り当てるか、サーバが拡張子とパスを削除したファイルのベース名を使用して割り当てます)。

接続先のデータベースがすでに実行されている場合は、データベースファイルではなくデータベース名を指定してください。

実行中のデータベース名と DatabaseName (DBN) パラメータで指定した名前が一致した場合にのみ、接続が行われます。

注意： データベース名とデータベースファイルの両方を指定した場合、データベースファイルは無視され、実行中のデータベースへの接続には使用されません。

例

- cities.db という名前のデータベースファイルを起動して、その名前を Kitchener に変更するには、次のコマンドを使用できます。

```
start iq cities.db -n Kitchener
```
- 上記のコマンドを実行した場合は、次のコマンドを使用して、Kitchener という名前の実行中のデータベースに接続できます。

```
DBN=Kitchener
```
- あるいは、次のコマンドを使用することもできます。

```
DBN=Kitchener;DBF=cities.db
```
- しかし、次のコマンドでは、Kitchener という名前のデータベースに接続できません。

```
DBF=cities.db
```

DatabaseSwitches 接続パラメータ [DBS]

データベースの起動時に、データベース固有のスイッチを提供します。

使用法

データベースがロードされていないときに、サーバに接続します。

デフォルト値

スイッチなし

説明

現在実行していないデータベースに接続するときのみ、**DatabaseSwitches** を使用してください。**DatabaseFile** で指定したデータベースをサーバが起動するとき、サーバは提供された **DatabaseSwitches** をコマンドラインオプションとして使用して、データベースの起動オプションを決定します。

このパラメータを使用して提供できるのは、データベーススイッチだけです。サーバスイッチは **START 接続パラメータ** を使用して提供してください。

例

次の UNIX コマンドを 1 行で入力します。このコマンドでは、デフォルトのデータベースサーバに接続し、データベースファイル `IQ-16_0/demo/iqdemo.db` (DBF パラメータ) をロードします。さらに、このファイルに `my_db` (DBS パラメータ) という名前を付け、この名前のデータベース (DBN パラメータ) に接続します。


```
dbcollat -c "uid=DBA;pwd=SQL;dbf=/IQ-16_0/demo/iqdemo.db;  
dbn=my_db;dbns=-n my_db" /tmp/temp.col
```

DataSourceName 接続パラメータ [DSN]

ODBC データソース情報を検索するとき、UNIX の場合は `.odbc.ini` ファイル、Windows の場合は `odbc.ini` ファイルまたはレジストリのどこを探すかを ODBC ドライバマネージャまたは Embedded SQL ライブラリに指示します。

使用法
特に制限なし

デフォルト値
なし

説明
データソース名だけを ODBC に送信するのは、ODBC アプリケーションの一般的な手法です。ODBC ドライバマネージャと ODBC ドライバは、接続パラメータの残りの部分を含んだデータソースを探します。SAP Sybase IQ では、Embedded SQL アプリケーションも ODBC データソースを使用して接続パラメータを保存できません。

例
次のパラメータは、データソース名を使用します。

```
DSN=Dynamo Demo
```

DBKEY 接続パラメータ [DBKEY]

暗号化されたデータベースを接続要求で起動します。

使用法
データベースの起動。実行中のデータベースに接続するときには使用しません。

デフォルト値
デフォルトでは、データベースは暗号化されていません。

説明
このパラメータは、暗号化されたデータベースを接続要求で起動するときに指定します。

例
次のプロファイルフラグメントは、暗号化キー "is!seCret" を使用して、強い暗号で保護された、現在実行中の `marvin.db` という名前のデータベースに接続することを示しています。

```
...
UID=dba;PWD=sql;DBF=marvin.db;DBKEY='is!seCret'
...
```

DisableMultiRowFetch 接続パラメータ [DMRF]

ネットワーク上での複数レコードのフェッチをオフにします。

使用法

すべての場所

デフォルト値

No

デフォルトでは、データベースサーバが単純なフェッチ要求を受信すると、アプリケーションは追加のローを要求します。このパラメータを ON に設定すると、この動作を無効にできます。

DisableMultiRowFetch パラメータを ON に設定した場合の動作は、**PREFETCH** オプションを OFF に設定した場合と同じです。

例

次の接続文字列フラグメントは、プリフェッチを無効にします。

```
DMRF=Yes
```

EngineName 接続パラメータ [ENG]

接続先となる実行中のデータベースサーバを指定します。これは **ServerName** と同じものを表します。

使用法

ネットワークサーバ

値

String

デフォルト値

デフォルトのローカルデータベースサーバ

説明

ネットワークサーバに接続するには、**EngineName** を指定します。[接続] ダイアログや ODBC アドミニストレータでは、[**Server Name**] フィールドになります。

サーバ名は、クライアントマシンの文字セットに従って解釈されます。マルチバイト文字をサーバ名に使用することはおすすめできません。

サーバ名は有効な識別子である必要があります。長いサーバ名は、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の命名制限については、『ユーティリティガイド』を参照してください。

プロトコル	トランケーションの長さ
UNIX 共有メモリ	31 バイト
非 UNIX 共有メモリ	40 バイト
TCP/IP	40 バイト

例

Guelph という名前のサーバに接続します。

```
ENG=Guelph
```

EncryptedPassword 接続パラメータ [ENP]

パスワードを指定し、データソースの暗号化形式で保管します。

使用法

(DSN および FILEDSN 接続パラメータは長いキーワードをサポートしていません)

値

String

デフォルト値

なし

説明

データソースは、ファイルとしてディスク上に、またはレジストリ内に保管されます。パスワードをディスクに保管する場合は、セキュリティ上の問題が生じることがあります。そのため、パスワードをデータソースに入力すると、パスワードは暗号化形式で保管されます。

Password と **EncryptedPassword** の両方を指定した場合は、**Password** が優先されません。

Encryption 接続パラメータ [ENC]

トランスポートレイヤセキュリティまたは単純暗号化を使用してクライアントアプリケーションとデータベースサーバ間で送信されるパケットを暗号化します。

使用法

TLS の場合は、TCP/IP のみ

NONE または **SIMPLE** の場合は、特に制限なし

値

String

デフォルト値

NONE

Encryption の値を設定しない場合は、暗号化はサーバの設定 (デフォルトは暗号化なし) によって制御されます。

説明

ネットワークパケットのセキュリティが確実でない場合は、このパラメータを使用できます。暗号化がパフォーマンスに及ぼす影響はごくわずかです。Encryption (ENC) 接続パラメータには次の引数を指定できます。

- **None**：暗号化されていない通信パケットを受け入れます。
- **Simple**：すべてのプラットフォームと 12.6 より前のバージョンの SAP Sybase IQ でサポートされている単純暗号を使って暗号化された通信パケットを受け入れます。単純暗号化では、サーバ認証、RSA 暗号化、またはその他のトランスポートレイヤセキュリティ機能は提供されません。
- **TLS**：RSA 暗号化技術を使用して暗号化された通信パケットを受け入れます。FIPS 認定の RSA 暗号化の場合は、FIPS=Y を指定します。RSA FIPS は別の認定ライブラリを使用しますが、RSA を指定している SAP Sybase IQ サーバと互換性があります。クライアントとサーバ間の TLS 接続は、サポートされているすべてのプラットフォームでサポートされますが、FIPS=Y は LinuxAMD と Windows (32 および 64 ビット) でのみサポートされます。サーバを認証するために、サーバの証明書の値が、次の引数を使用して指定した値に一致することがソフトウェアによって確認されます。
 - **trusted_certificate**：クライアントがサーバを認証するために使用する証明書ファイルを指定します。
 - **certificate_company**：組織フィールドの値を指定します。サーバの値とクライアントの値が一致しなければなりません。

- **certificate_unit**：組織単位フィールドの値を指定します。サーバの値とクライアントの値が一致しなければなりません。
- **certificate_name**：証明書の通称を指定します。サーバの値とクライアントの値が一致しなければなりません。

警告！ サンプルの証明書はテスト目的でのみ使用してください。サンプルの証明書とそこで使用されるパスワードは SAP Sybase IQ ソフトウェアに付属して広く配布されているため、展開後の運用環境でこの証明書を使用した場合、セキュリティは確保されません。独自の証明書を作成して、システムを保護してください。

CONNECTION_PROPERTY システム関数を使用することによって、現在の接続の暗号化設定を取得できます。

例

- 次の接続文字列フラグメントでは、RSA 暗号化と信頼された証明書のサンプルを使用して、TCP/IP リンクを通じてデータベースサーバ myeng に接続します。

```
"ENG=myeng; LINKS=tcPIP;  
Encryption=(FIPS=N;TRUSTED_CERTIFICATE=iq-16_0/samples/  
certificates/rsaroot.crt)"
```

Escape 接続パラメータ [ESCAPE]

テーブルまたはカラムのリストを返すときに、ODBC ドライバで生成される SQL 文の LIKE 句で使用されるエスケープ文字を指定することができます。

説明

Escape パラメータをサポートしているのは SAP Sybase IQ のみです。

デフォルトでは、ODBC ドライバでチルダ (~) がエスケープ文字として使用されますが、一部のアプリケーションではエスケープ文字に円記号 (¥) が使用されることもあります。16 進のコードおよび記号として格納されている文字に使用するデフォルトのエスケープ文字は、円記号 (¥) です。たとえば、¥x0A は改行文字です。

例

感嘆符 (!) をエスケープ文字として使用するには、次のように入力します。

```
ESCAPE='!'
```

FileDataSourceName 接続パラメータ [FileDSN]

接続先のデータベースに関する情報を含んだ ODBC ファイルデータソースがあることをクライアントライブラリに通知します。

使用法

すべての場所

値
String

デフォルト値
None

説明

ファイルデータソースは、レジストリに保管される ODBC データソースと同じ情報を持ちます。ファイルデータソースは、簡単にエンドユーザに配布できるため、接続情報を各マシン上で再構成する必要はありません。

ODBC と Embedded SQL アプリケーションのいずれもファイルデータソースを使用できます。

例

次に示すのは、ファイルデータソース内にあるデータソースの記述です。

```
[Sample File Data Source]
ENG=iqdemo
DBA=DBA
PWD=SQL
```

Idle 接続パラメータ [IDLE]

Idle 接続パラメータには、接続がタイムアウトするまでの時間を指定します。

関数

接続のアイドルタイムアウト期間を指定します。

使用法

TDS (Tabular Data Stream™) 接続と共有メモリ接続を除くすべての場所。共有メモリ接続と TDS 接続 (jConnect を含む) は SAP Sybase IQ **Idle (IDLE)** 接続パラメータを無視します。

値
Integer

デフォルト値
-ti の値

説明

Idle (IDLE) 接続パラメータは、現在の接続に対してのみ適用されます。同じサーバ上の複数の接続に対してそれぞれ異なるタイムアウト値を設定できます。

接続アイドルタイムアウト値が設定されていない場合、アイドルタイムアウト値はサーバの設定 (**start_iq** で設定されているときのデフォルトは 4400 分) によって制御されます。タイムアウト値が競合する場合は、指定されているかどうかに関係なく、常に接続のタイムアウト値の方がサーバのタイムアウト値よりも優先されます。

必要であれば、Idle と Liveness Timeout の両方に、関連するサーバコマンドラインパラメータ (それぞれ **-ti** と **-tl**) を含めることができます。

参照

-ti iqsrv16 サーバオプションについては、『ユーティリティガイド』の「start_iq データベースサーバ起動ユーティリティ」を参照してください。

例

次の接続文字列フラグメントは、この接続のタイムアウト値を 10 分に設定します。

```
"ENG=myeng;LINKS=tcpip;IDLE=10"
```

Integrated 接続パラメータ [INT]

統合化ログイン機能を使用します。

使用法

すべての場所

値

YES, NO

デフォルト値

NO

説明

Integrated パラメータには次の設定があります。

- 統合化ログインを行います。接続の試行が失敗し、**LOGIN_MODE** オプションが **Mixed** (廃止予定) に設定されている場合は、標準ログインを試行します。
- デフォルトの設定です。統合化ログインは試行されません。

クライアントアプリケーションで統合化ログインを使用するには、**LOGIN_MODE** データベースオプションを **Mixed** (廃止予定) か **Integrated** に設定してサーバを起動してください。

例

次のデータソースフラグメントは、統合化ログインを使用します。

```
INT=yes
```

Language 接続パラメータ [LANG]

接続の言語を指定します。

使用法

すべての場所

値

言語を表す 2 文字の組み合わせ。たとえば、**LANG=DE** の場合、デフォルトの言語がドイツ語に設定されます。

デフォルト値

IQLANG 環境変数またはインストーラによって指定された言語 (IQLANG 環境変数の方が優先)

説明

この接続パラメータは接続の言語を設定します。サーバが指定の言語をサポートしている場合には、サーバからのエラーまたは警告はこの言語で返されます。

言語を指定しない場合は、デフォルトの言語が使用されます。

この接続パラメータは接続に対してのみ作用します。SQL Anywhere ツールとユーティリティから返されるメッセージはデフォルトの言語で表示されますが、サーバから戻されるメッセージは接続の言語で表示されます。

LazyClose 接続パラメータ [LCLOSE]

CLOSE *cursor-name* データベース要求がキューに入れられ、次のデータベース要求とともにサーバに送信されます。これにより、カーソルをクローズするたびにネットワーク要求を送信する必要がなくなります。

使用法

すべての場所

値

YES, NO

デフォルト値

NO

説明

このパラメータを有効にしたときは、次のデータベース要求が来るまで、カーソルは実際にはクローズされません。CLOSE *cursor-name* データベース要求がキュー

イングされている間は、独立性レベル 1 のすべてのカーソル安定性ロックがカーソルに適用されます。

このオプションを有効にすると、次の場合にパフォーマンスが向上します。

- ネットワークの遅延時間が長い場合
- アプリケーションがカーソルのオープン要求とクローズ要求を多数送信する場合

非常にまれな状況で、CLOSE *cursor-name* データベース要求の後に次の要求がキャンセルされた場合に、クライアント側ではカーソルがクローズされたものと認識しているのに、サーバ側では実際にはカーソルがクローズされていないことがあります。その後に、同じ名前別のカーソルをオープンしようとするとう失敗します。要求を頻繁にキャンセルするアプリケーションでは、LazyClose を使用しないでください。

LivenessTimeout 接続パラメータ [LTO]

接続が利用されなくなった場合の終了方法を制御します。

使用法

TCP/IP 通信プロトコル上のネットワークサーバ

すべてのプラットフォーム (非スレッド UNIX アプリケーションを除く)

値

Integer (秒単位)

デフォルト値

120

LivenessTimeout 値を設定しない場合、活性タイムアウトはサーバ設定 (デフォルト値は 120 秒) を使って制御されます。

説明

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 通信プロトコルを介して、定期的に活性パケットが送信されます。クライアントが活性要求パケットまたは活性応答パケットを検出しない時間が一定の活性タイムアウト期間を超えると、通信が切断されます。

LivenessTimeout 値の 3 分の 1 から 3 分の 2 の期間で接続がパケットを送信しない場合に、活性パケットが送信されます。

通信が切断されると、クライアントマシンは接続先のサーバのアドレスを忘れます。クライアントマシンは、次にサーバに接続するときにアドレスを検索し、そのサーバに対するすべての現在の接続を削除します。

サーバへの接続数が 200 を超えると、サーバは指定された **LivenessTimeout** 値に基づいて、それより大きい **LivenessTimeout** 値を自動的に算出します。それにより、サーバでの大量の接続の処理が効率化されます。

別の方法として、[ODBC 設定] ダイアログボックスの [ネットワーク] タブの [LivenessTimeout] テキストボックスに値を入力して、このパラメータを設定することもできます。

例

次の例は、Liveness Timeout 値を 60 秒に設定しています。

```
LTO=60
```

LogFile 接続パラメータ [LOG]

クライアントのエラーメッセージおよびデバッグメッセージをファイルに送信します。

使用法

すべての場所

値

String

説明

クライアントのエラーメッセージとデバッグメッセージをファイルに保存するには、**LogFile (LOG)** パラメータを使用します。

ファイル名にパスが含まれる場合、そのパスはクライアントアプリケーションの現在の作業ディレクトリを基準とした相対パスになります。

LogFile (LOG) 接続パラメータは接続に対して固有なので、1つのアプリケーションからさまざまな接続に対して異なる LogFile 引数を設定できます。

例

次のコマンドラインフラグメントでは、この接続に関するクライアントメッセージを、そのクライアントの現在の作業ディレクトリ内にある `error.log` ファイルに送信することを指定しています。

```
...  
LogFile=error.log  
...
```

LogicalServer 接続パラメータ [LS]

接続対象の論理サーバを指定します。

使用法
特に制限なし

値
String

説明

このパラメータを使用して、対象論理サーバを明示的に指定します。これにより、SAP Sybase IQ は別のマルチプレックスノードに接続をリダイレクトできます。

接続文字列に LogicalServer が指定されていない場合、ユーザのログインポリシー内のデフォルト論理サーバの設定が論理サーバコンテキストとして使用されます。

LoginRedirection 接続パラメータ [REDIRECT]

接続レベルのログインリダイレクションを制御します。

使用法
このパラメータを有効にするには、論理サーバレベルのログインリダイレクションが有効である必要があります。

値
[ON | OFF]

デフォルト
ON

説明

特定のアプリケーションが論理サーバ内の特定の対象ノードに接続することが必要になります。

REDIRECT=OFF と指定すると、その接続に対してログインリダイレクションは発生しません。初期接続のノードが対象論理サーバの接続要件と要求されたサーバロールを満たしていない場合、接続は失敗します。

たとえば、ノード間で共有されない SQL Anywhere テーブルにデータを格納している論理サーバ内の対象ノードに対しては、接続レベルでのログインリダイレクションは無効にするなどできます。

NewPassword (NEWPWD) 接続パラメータ

パスワードの有効期限が切れている場合でも、DBA の介入を必要とせずユーザが各自でパスワードを変更できます。

構文

```
{ NewPassword | NEWPWD }={ password-string | * }
```

使用法

新しいパスワードの入力を要求するクライアントライブラリは、Microsoft Windows でのみサポートされます。

指定可能な値

- **password-string** – ユーザが新しいパスワードを入力すると、データベースサーバはユーザ ID とパスワードの認証を実行し、login_procedure オプションが呼び出される前にパスワードを変更しようとします。このプロセスにより、ユーザは DBA の介入なく有効期限の切れたパスワードを変更できるようになります。verify_password_function オプションを設定した場合は、新しいパスワードの検証が行われます。統合化ログインや Kerberos ログインで認証を実行する場合、元のパスワードは検証されず、データベースサーバは新しいパスワード値を無視し、パスワードは変更されません。
- ***** – Microsoft Windows の場合、特別値 * を使用すると、既存のパスワードの有効期限が切れている場合にかぎり、クライアントライブラリは新しいパスワードの入力を要求します。ユーザは既存のパスワードと新しいパスワードを入力し、さらに確認のために新しいパスワードをもう一度入力する必要があります。ユーザがフィールドに値を入力して [OK] をクリックすると、古いパスワードが認証され、データベースサーバはパスワードを変更しようとします。verify_password_function オプションを設定した場合は、新しいパスワードの検証が行われます。ユーザのパスワードの有効期限切れを確認し、新しいパスワードの入力を要求し、パスワードの認証と変更を行うプロセスは、クライアントライブラリに対する 1 回の接続呼び出しで実行されます。

デフォルト

パスワードは変更されず、クライアントライブラリは新しいパスワードを要求しません。

備考

この接続パラメータは、password_life_time オプションや password_expiry_on_next_login オプションを使用してログインポリシーを実装する場合に非常に便利です。また、login_procedure で「パスワードの有効期限が切れ

ています。」というエラーを伝えることで、パスワードの有効期限ポリシーを実装することもできます。

動作環境でパスワードの入力要求がサポートされていない場合、ユーザは「パスワードの有効期限が切れています。」というエラーを受信します。Microsoft Windows 環境の場合、呼び出し元のアプリケーションに最上位レベルのウィンドウが複数あったり、最小化されている場合、プロンプトウィンドウでは呼び出し元のアプリケーションウィンドウとの対話を正しく防げない(モーダルでなくなったり、正しい親ウィンドウがない)ことがあります。

Windows 環境では、ODBC SQLDriverConnect 関数を使用していて DriverCompletion 引数が SQL_DRIVER_NOPROMPT 以外に設定されている場合、パスワードの有効期限が切れていると、接続時に新しいパスワードの入力を要求されます。DBPROP_INIT_PROMPT プロパティが DBPROMPT_NOPROMPT 以外に設定されている場合、OLE DB で接続時に新しいパスワードの入力を要求されることがあります。いずれの場合も、接続パラメータ NewPassword=* が指定されているように動作します。

- ユーザ ID には、次に該当する値を指定できません。

空白スペース、一重引用符、または二重引用符で始まる値
空白スペースで終わる値
セミコロンを含む値

- パスワードは大文字と小文字が区別され、次に該当する値は指定できません。

空白スペース、一重引用符、または二重引用符で始まる値
空白スペースで終わる値
セミコロンを含む値
255 バイト長を超える値

例

次の接続文字列では、接続時にユーザ Test1 のパスワードを変更します。

```
"UID=Test1;PWD=welcome;NEWPWD=hello"
```

Windows 環境の場合、次の接続文字列では、既存のパスワードの有効期限が切れている場合に、ユーザ Test1 に新しいパスワードの入力を要求します。

```
"UID=Test1;PWD=welcome;NEWPWD=*"
```

NodeType 接続パラメータ

論理サーバのメンバーノードに対してサーバロールを指定します。

使用法
特に制限なし

値

{ **READER** | **WRITER** | **ANY** }

デフォルト

ANY

説明

論理サーバは、リーダノードとライタノードの両方をメンバーとして含むことができます。このパラメータにより、アプリケーションは特定ロールを持つメンバーノードに接続することができます。目的のロールは **NODETYPE = {READER | WRITER | ANY}** として指定できます。

- **READER** – ユーザ定義関数を実行するアプリケーションが、リーダロールを持つメンバーロールに接続することが必要になります。
- **WRITER** – 文 (**INSERT**, **UPDATE**, **DELETE** など) を実行するアプリケーションが、ライタロールを持つメンバーノードに接続することが必要になります。
- **ANY** – 大部分の読み込み専用 (UDF を実行しない) アプリケーションが、論理サーバのどのメンバーノードにでも接続できます。

SAP Sybase IQ は、ユーザが **SERVER** 論理サーバコンテキストでログインする際、接続パラメータ **NodeType** を無視します。

Password 接続パラメータ [PWD]

接続時のパスワードを指定します。

使用法

すべての場所

デフォルト値

パスワードの指定なし

説明

すべてのデータベースユーザにはパスワードが設定されています。ユーザがデータベースに接続するためには、パスワードを提供する必要があります。デフォルトでは、パスワードで大文字と小文字が区別されるかどうかはデータと同じです。また、SAP Sybase IQ は、デフォルトで大文字と小文字を区別します。

パスワードパラメータは暗号化されていません。パスワードを接続プロファイルに保存する場合は、**EncryptedPassword** パラメータを使用してください。SAP Sybase IQ ODBC 設定ツールは、暗号化したパスワードを使用します。

Password と **EncryptedPassword** の両方を指定した場合は、**Password** が優先されません。

例

次の接続文字列フラグメントは、ユーザ ID DBA とパスワード SQL を指定します。

```
uid=DBA;pwd=SQL
```

別の方法として、接続ウィンドウの [ユーザ ID] テキストボックスと [パスワード] テキストボックスを使って、これらのパラメータを設定できます。

PrefetchBuffer 接続パラメータ [PBUF]

ユーザが最大メモリ量を設定できるようにします。

ローバッファリングのための最大メモリ量をバイト単位で設定します。

使用法

すべての場所

値

Integer { **k** | **m** }

デフォルト値

512 (KB)

説明

PrefetchBuffer 接続パラメータでは、プリフェッチされたローを格納するためにクライアントに割り付けられるメモリの量を制御します。デフォルト値はバイト単位ですが、**k** または **m** を使用して、キロバイト単位またはメガバイト単位を指定できます。この接続パラメータは、64KB ~ 8MB の値を受け入れます。場合によっては、クライアントがデータベースサーバからプリフェッチするローの数を増やすと、クエリのパフォーマンスが向上します。**PrefetchRows** と **PrefetchBuffer** の接続パラメータを使用すると、プリフェッチするローの数を増やすことができます。

PrefetchBuffer (PBUF) 接続パラメータを増やすと、GET DATA 要求のバッファに使用できるメモリ容量が増えます。多数の GET DATA (SQLGetData) 要求を処理するアプリケーションでは、このように設定するとパフォーマンスが向上します。

例

プリフェッチされるローの数が PrefetchBuffer メモリの制限によって減少するかどうかを確認するには、次の接続文字列フラグメントを使用します。

```
...prefetchrows=100;logfile=c:¥ client.txt
```

メモリ制限を 256KB に増やすには、次のフラグメントを使用します。

```
...prefetchrows=100;prefetchbuffer=256
```

PrefetchRows 接続パラメータ [PROWS]

PrefetchRows 接続パラメータを使うと、データベースのクエリ時にプリフェッチされるローの最大数を指定できます。

使用法

すべての場所

デフォルト値

10 (.NET Data Provider を使用している場合は 200)

説明

クライアントがデータベースサーバからプリフェッチするローの数を増やすと、単一ローフェッチまたはワイドフェッチで RELATIVE 0 または 1 のフェッチのみを行うカーソルのパフォーマンスが向上します。ワイドフェッチには、Embedded SQL 配列フェッチと ODBC ブロックフェッチが含まれます。

特に次の状況ではパフォーマンスが向上します。

- アプリケーションが非常に少ない絶対フェッチで多数の (何百もの) ローをフェッチする場合。
- アプリケーションがローを大量にフェッチし、かつ、クライアントとサーバが同じマシン上にあるか高速ネットワークで接続されている場合
- クライアント/サーバ通信にダイヤルアップリンクや広域ネットワークなどの伝送速度の遅いネットワークを使用している場合。

プリフェッチされるローの数は、PrefetchRows 接続パラメータと PrefetchBuffer 接続パラメータの両方によって制限されます (PrefetchBuffer 接続パラメータは、プリフェッチされるローを格納するためのメモリを制限します)。

プリフェッチできるローの最大数は 1000 です。

例

次の接続文字列フラグメントでは、プリフェッチされるローの数を 100 に設定しています。

```
...prefetchrows=100;...
```

RetryConnectionTimeout (RetryConnTO) 接続パラメータ

クライアントライブラリ (DBLIB、ODBC、ADO など) に、サーバが見つからないうちは指定した時間が経過するまで接続を試行し続けるよう指示します。

構文

```
{ RetryConnectionTimeout | RetryConnTO }=timeout-value
```


使用法

特に制限なし

指定可能な値

- **timeout-value** – この接続パラメータで指定される値は、秒単位のタイムアウトです。接続試行のリトライ回数を示すカウンタ値ではありません。デフォルト値の 0 は、接続を 1 度だけ試行することを示します。

デフォルト

0

備考

間隔は 0.5 秒で、接続しようとしてデータベースサーバが見つからなかった場合にだけリトライされます。それ以外のエラーは、すぐに返されます。データベースサーバが見つからなかった場合、少なくとも `RetryConnectionTimeout` 接続パラメータに指定されている期間、接続が試行されます。

デフォルトの TCP タイムアウトは 5 秒です。接続文字列に 5 秒未満に指定された `RetryConnTO` が含まれている場合でも (`Host=host-name;RetryConnTO=3` など)、5 秒間は接続が試行されます。

例

次の接続文字列フラグメントは、接続試行を最低 5 秒間リトライし続けるようクライアントライブラリに指示します。

```
...RetryConnTO=5;...
```

ServerName 接続パラメータ [ENG]

EngineName 接続パラメータ [ENG] の同意語です。

StartLine 接続パラメータ [START]

アプリケーションからデータベースサーバを起動します。

使用法

埋め込みデータベース

デフォルト値

StartLine パラメータなし

説明

現在実行中でないデータベースサーバに接続するときにかぎり、StartLine パラメータを指定します。StartLine パラメータは、サーバを起動するコマンドラインです。

例

次のデータソースフラグメントでは、32MB のキャッシュを使ってデータベースサーバを起動します。

```
StartLine=dbeng6 -c 32M iqdemo.db
```

Unconditional 接続パラメータ [UNC]

サーバへの接続があるときでも、**dbstop** によってサーバを停止します。

使用法

すべての場所

デフォルト値

No

説明

dbstop コマンドラインユーティリティはデータベースサーバを停止します。接続文字列で **Unconditional=Yes** を指定すると、データベースサーバはアクティブな接続があるときでも停止します。**Unconditional** が **Yes** に設定されていない場合は、アクティブな接続がないときにかぎりサーバは停止します。

例

次のコマンドラインでは、無条件にサーバを停止します。

```
dbstop -c "uid=DBA;pwd=SQL;eng=server-name;unc=yes"
```

参照

『ユーティリティガイド』の「c」

Userid 接続パラメータ [UID]

データベースへのログオン時に使用するユーザ ID を指定します。

使用法

(DSN および FILEDSN 接続パラメータは長いキーワードをサポートしていません)

デフォルト値

なし

説明

データベースに接続するときは、必ずユーザ ID を指定してください。ユーザ ID では大文字と小文字を区別しません。また、ユーザ ID は Case Respect データベースプロパティの設定の影響を受けません。

例

次の接続文字列フラグメントは、ユーザ ID DBA とパスワード SQL を指定します。

```
uid=DBA;pwd=SQL
```

ネットワーク通信パラメータ

クライアント／サーバのネットワーク通信に問題がある場合は、クライアントとサーバに対して、多数のコマンドラインパラメータを設定できます。このパラメータを使って、ネットワークプロトコルの問題を回避します。

次の例のように、サーバまたはクライアントのコマンドラインにネットワーク通信パラメータを指定します。

```
start_iq -x tcpip(PARM1=value1;PARM2=value2;. . .),...
```

クライアント側で、次の例のように通信パラメータを入力します。

```
CommLinks=tcpip(PARM1=value1;PARM2=value2;. . .),...
```

スペースが含まれるネットワーク通信パラメータは、適切に解析されるよう、引用符で囲みます。

```
start_iq -x "tcpip(PARM1=value 1;PARM2=value 2;...),..."
start_iq -x "tcpip(PARM1=value1;PARM2=value2;...)"
```

UNIX では、セミコロンがコマンドの区切り文字として解釈されるため、複数のパラメータを指定する場合は、引用符が必要です。

ブールパラメータは、YES、ON、TRUE、1 のいずれかによってオンになり、NO、OFF、FALSE、0 のいずれかによってオフになります。ブールパラメータは大文字と小文字を区別しません。

通信パラメータを設定するコマンドを 1 行に入力します。また、コマンドを設定ファイルに記述し、サーバまたはクライアントのコマンドラインの @ スイッチを使って設定ファイルを呼び出すこともできます。

TCP/IP、HTTP、HTTPS の通信パラメータ

TCP/IP、HTTP、HTTPS で現在使用できるパラメータは次のとおりです。

TCP/IP	HTTP と HTTPS
Broadcast (BCAST)	ID
BroadcastListener (BLISTENER)	Identity_Password
ClientPort (CPORT)	DatabaseName (DBN)
DLL	LocalOnly (LOCAL)
DoBroadcast (DOBROAD)	LogFile (LOG)
Host (IP)	LogMaxSize (LSize)
LocalOnly (LOCAL)	LogOptions (LOpt)
LDAP (LDAP)	LogFormat (LF)
MyIP (ME)	MaxConnections (MaxConn)
ReceiveBufferSize (RCVBUFSZ)	MaxRequestSize (MaxSize)
SendBufferSize (SNDBUFSZ)	MyIP (ME)
ServerPort (PORT)	ServerPort (PORT)
TDS	Timeout (TO)
Timeout (TO)	
VerifyServerName (VERIFY)	

Broadcast 通信パラメータ [BCAST]

ブロードキャストアドレスを指定します。

使用法
TCP/IP

値
String (IP アドレス形式)

デフォルト値
同じサブネット上の全アドレスへのブロードキャスト

説明
デフォルトのブロードキャストアドレスは、ローカル IP アドレスとサブネットマスクを使用して作成されます。サブネットマスクは、IP アドレスのどの部分がネットワークを指定し、どの部分がホストを指定するかを示します。

サブネットが 10.24.98.x でマスクが 255.255.255.0 の場合、デフォルトのブロードキャストアドレスは 10.24.98.255 になります。

Windows プラットフォームで IPv6 アドレスを指定する場合は、インタフェース識別子を使用します。UNIX プラットフォームでは、IPv6 アドレスのインタフェース識別子とインタフェース名の両方がサポートされています。Linux (カーネル 2.6.13 以降) では、インタフェース識別子が必須です。

例

IPv6 の使用時に、インタフェース番号 2 でのみブロードキャストするようクライアントに指示するには、次のようにします。

```
LINKS=tcpip (BROADCAST=ff02::1%2)
```

BroadcastListener 通信パラメータ [BLISTENER]

このポートのブロードキャスト受信を制御します。

使用法

TCP/IP、サーバ側

値

YES, NO

デフォルト値

YES

説明

NO を指定すると、このポートの受信が無効になります。**-sb 0** を指定することは、TCP/IP で BroadcastListener=NO に設定するのと同じことです。

例

BroadcastListener=NO の TCP/IP 接続を受け入れるサーバを起動します。

```
start_iq -x tcpip (BroadcastListener=NO)
```

ClientPort 通信パラメータ [CPort]

TCP/IP を使用してクライアントアプリケーションが通信するポート番号を指定します。1 つのポート番号、またはポート番号の組み合わせを指定したり、ポート番号の範囲を指定したりできます。

使用法

TCP/IP (クライアント側のみ)

デフォルト値

ネットワークの実装によって、接続ごとに動的に割り当てられます。ファイアウォールによる制限がない場合は、このパラメータを使用しないでください。

説明

このオプションは、ファイアウォールを介する通信のためのものです。ファイアウォールソフトウェアは、TCP/UDP ポートに従ってフィルタします。ファイアウォールの都合により必要な場合以外は、このパラメータを使用しないことをおすすめます。

ポート番号のリストまたは範囲を指定して、指定されたデータソースまたは接続文字列を使用する複数の接続を確立します。ポート番号を1つだけ指定すると、アプリケーションが維持できるのは、一度に1つの接続のみとなります。また、1つの接続を閉じた後は、短いタイムアウト時間が生じます。その間、指定されたポートを使って新しい接続を作成できません。ポート番号のリストや範囲を指定すると、アプリケーションは、いずれかのポート番号との接続が確立するまで、試行を続けます。

例

- 次の例は、ポート 6000 を使用するアプリケーションから、ポート 5000 を使用する my_server という名前のサーバへの接続を確立します。

```
CommLinks=tcPIP{ClientPort=6000;ServerPort=5000};  
ServerName=my_server
```

- 次の例は、ポート 5050 から 5060、および 5040 と 5070 を使用できるアプリケーションから、デフォルトのサーバポートを使用する my_server という名前のサーバへ通信する接続を確立します。

```
CommLinks=tcPIP{ClientPort=5040,5050-5060,5070};ServerName=my_server
```

DatabaseName 通信パラメータ [DBN]

Web 要求を処理するとき使用するデータベースの名前を指定します。URI の一部としてデータベース名を要求するかどうかを指定する場合は、REQUIRED または AUTO を使用します。

使用法

HTTP、HTTPS

値

AUTO、REQUIRED、*database-name*

*デフォルト値***AUTO***説明*

このパラメータを **REQUIRED** に設定した場合は、URI でデータベース名を指定する必要があります。

このパラメータを **AUTO** に設定した場合は、URI でデータベース名を指定できませんが、必須ではありません。URI にデータベース名が含まれていない場合は、サーバ上のデフォルトデータベースで Web 要求が処理されます。**AUTO** に設定すると、URI にデータベース名が含まれているかどうかをサーバが推測しなければならないので、Web サイトの設計時にはあいまいさを避けるようにする必要があります。

このパラメータにデータベース名を指定した場合は、そのデータベースを使ってすべての Web 要求が処理されます。この場合、URI にデータベース名を含めてはなりません。

例

次のコマンドは2つのデータベースを起動しますが、HTTP 経由でのアクセスを許可されるのはそのうちの1つだけです。

```
start_iq -xs http(dbn=web) iqdemo.db web.db
```

DoBroadcast 通信パラメータ [DBROAD]

サーバを検索するためのブロードキャストを行います。

使用法

TCP/IP (すべてのプラットフォーム)

値

ALL、**NONE**、**DIRECT** (クライアント側)

YES、**NO** (サーバ側)

デフォルト値

ALL

説明

- クライアントでの使用 - DoBroadcast=ALL (以前の DoBroadcast=YES) に設定した場合は、データベースサーバを検索するときに、ローカルサブネットへのブロードキャストが実行されます。最初は、ローカルサブネットにブロードキャストされます。HOST= を指定した場合、各ホストにもブロードキャスト

パケットが送信されます。TCP の場合は、すべてのブロードキャストパケットが UDP パケットになります。

- DoBroadcast=DIRECT (以前の DoBroadcast=NO) を設定した場合、データベースサーバを検索するときに、ローカルサブネットへのブロードキャストは実行されません。ブロードキャストパケットは、HOST (IP) 通信パラメータで指定したホストにのみ送信されます。DoBroadcast=DIRECT に設定した場合は、HOST (IP) 通信パラメータは必須です。
- DoBroadcast=NONE に設定した場合は、UDP ブロードキャストが使用されなくなります。指定の HOST/PORT に対して TCP/IP 接続が直接確立され、サーバ名が検証されます。TCP/IP では、VerifyServerName (VERIFY) 通信パラメータを **NO** に設定することで、サーバ名の検証を回避できます。HOST (IP) 通信パラメータは必須のパラメータですが、ServerPort (PORT) 通信パラメータはオプションです。
- DIRECT または NONE の場合は、HOST オプションを使ってサーバホストを指定します。
- サーバでの使用 - DoBroadcast=NO に設定した場合は、データベースサーバが同じ名前を持つ他のサーバを探すときに、ブロードキャストを実行しなくなります。この設定が役に立つ場合もまれにありますが、通常はおすすめしません。

例

- 次のコマンドは、ブロードキャストを実行してデータベースサーバを検索することなく、クライアントを起動します。サーバは **silver** という名前のコンピュータ上でのみ検索されます。

```
dbisql -x tcpip(DOBROADCAST=DIRECT;HOST=silver) iqdemo
```

- UNIX でこのオプションを使用する場合は、二重引用符で囲んでください。

```
dbisql -x "tcpip(DOBROADCAST=DIRECT;HOST=silver)" iqdemo
```

Host 通信パラメータ [IP]

クライアントライブラリの検索対象となる、直接接続されているネットワークの外部にある追加マシンを指定します。

使用法

TCP/IP (全プラットフォーム)、サーバ側およびクライアント側

説明

サーバ上では、重複する名前のサーバを起動しないようにして検索されます。

TCP/IP の場合は、*hostname* を指定するか、ピリオドで区切った IP アドレスを指定します。

-z スイッチを使用した場合は、サーバの起動時にこのアドレス情報が表示されま
す。また、**LogFile** が指定されている場合 (**Debug** が TRUE に設定されている場合)
は、この情報がログファイルに書き込まれます。

セミコロンで区切ったアドレスのリストを使って、複数のマシンを検索すること
もできます。また、コロンを区切り文字として使って、ポート番号を IP アドレス
に追加できます。別の方法として、*Host=nnn.nn.nnn.nnn;ServerPort=pppp* のように、
ホストとサーバポートを明示的に指定することもできます。

TCP/IP を使用する場合、**IP** と **HOST** は同意語です。

Windows プラットフォームで IPv6 アドレスを指定する場合は、インタフェース識
別子を使用します。UNIX プラットフォームでは、IPv6 アドレスのインタフェー
ス識別子とインタフェース名の両方がサポートされています。Linux (カーネル
2.6.13 以降) では、インタフェース識別子が必要です。

値
String

デフォルト値
追加のマシンなし

例

- 次の接続文字列フラグメントは、"kangaroo" という名前のマシンと
197.75.209.222 (ポート 2369) のマシンを検索し、**iqdemo** という名前のデー
タベースサーバを見つけるようにクライアントに指示します。

```
...ENG=iqdemo CommLinks=tcPIP(IP=kangaroo;IP=197.75.209.222:2369)
```
- UNIX では、TCP/IP オプションの前後を二重引用符で囲む必要があります。

```
dbisql -x "tcPIP(HOST=kangaroo;HOST=197.75.209.222)" iqdemo
```
- 次の接続文字列フラグメントは、**my_server** と **kangaroo** というマシンを検索し
て、データベースサーバを見つけるようにクライアントに命令します。クライ
アントは、最初に応答を返したホストに対して接続を確立しようとします。

```
dbisql -c  
"UID=DBA;PWD=sql;LINKS=tcPIP(HOST=my_server,kangaroo;PORT=2639)"
```

Identity 通信パラメータ

ID ファイルの名前を指定します。

構文
Identity=identity-file

使用可能なプロトコル
HTTPS

許容値

- **identity-file** – この文字列は、ID ファイルの名前を指定します。

デフォルト
なし

備考

トランスポートレイヤセキュリティを使用している場合、ID ファイルには、パブリック証明書とプライベートキーが含まれており、自己署名されない証明書の場合は、さらに署名を行うすべての証明書も含まれています。これには暗号化証明書も含まれます。この証明書のパスワードは、Identity_Password パラメータで指定します。

例

特定の暗号化された証明書を使用するために、Web 接続が必要なデータベースサーバを起動します。

```
iqsrv16 -xs https(Identity=cert.file;Identity_Password=secret) ...
```

Identity_Password 通信パラメータ

暗号化証明書のパスワードを指定します。

構文

Identity_Password=password

使用可能なプロトコル
HTTPS

許容値

- **password** – この文字列は、暗号化証明書のパスワードを指定します。

デフォルト
なし

備考

トランスポートレイヤセキュリティを使用している場合、このオプションにより、Identity 通信パラメータで指定した暗号化証明書に対応するパスワードを指定します。

例

特定の暗号化された証明書を使用するために、Web 接続が必要なデータベースサーバを起動します。

```
iqsrv16 -xs https (Identity=cert.file;Identity_Password=secret) ...
```

LDAP 通信パラメータ [LDAP]

このパラメータを使用すると、WAN またはファイアウォールを介して接続するクライアントが、IP アドレスを指定しなくてもサーバを検索できるようになります。

使用法

このパラメータを使用すると、WAN またはファイアウォールを介して接続するクライアントが、IP アドレスを指定しなくてもサーバを検索できるようになります。また、検出ユーティリティ (**dblocate**) もそのようなサーバを検索できるようになります。

TCP/IP (サーバ側のみ)

値

YES、**NO**、または *filename* のいずれか

デフォルト値

ON

デフォルトの *filename* は *asaldap.ini* です。

説明

データベースサーバを LDAP サーバに登録し、クライアント (および Locate ユーティリティ [**dblocate**]) が LDAP サーバにクエリを発行できるようにします。

LDAP=filename と指定すると、LDAP のサポートがオンになり、指定したファイルが設定ファイルとして使用されます。**LDAP=YES** と指定すると、LDAP のサポートがオンになり、*saldap.ini* が設定ファイルとして使用されます。

LDAP は TCP/IP でネットワークサーバ上でのみ使用されます。

LocalOnly 通信パラメータ [LOCAL]

クライアントがローカルマシン上のサーバ (存在する場合) のみに接続できるようにします。

使用法

TCP/IP、HTTP、HTTPS

値

YES、**NO**

デフォルト値

NO

説明

ローカルマシン上に指定のサーバ名に該当するサーバがない場合は、サーバを自動起動しません。

LocalOnly (LOCAL) 通信パラメータが役に立つのは、DoBroadcast=ALL (デフォルト) の場合のみです。

LocalOnly=YES の場合は、別のマシン上のサーバからのブロードキャスト応答が無視されるという点以外は、通常のブロードキャストが実行されます。

サーバに対して LocalOnly (LOCAL) 通信パラメータを使用すると、ローカルマシンからのみ接続可能になります。リモートマシンからの接続ではこのサーバを検出できず、Locate [dblocate] ユーティリティでもこのサーバを検出できません。

LocalOnly (LOCAL) 通信パラメータを YES に設定してサーバを実行すると、そのネットワークサーバを接続や CPU の制限がないパーソナルサーバとして実行できます。

Logfile 通信パラメータ [LOG]

データベースサーバが Web 要求に関する情報を書き込むファイルの名前を指定します。

使用法

HTTP、HTTPS

値

Filename

デフォルト値

なし

LogFormat 通信パラメータ [LF]

ログファイルに書き込まれるメッセージのフォーマットと、表示されるフィールドを制御します。

使用法

HTTP、HTTPS

値

Format-string

デフォルト値

@T - @W - @I - @P - "@M @U @V" - @R - @L - @E

説明

次に示すコードのいずれかがメッセージ文字列に含まれている場合は、各メッセージの書き込み時に、コードが現在の値に置き換えられます。

- @ 文字
- 要求の処理が開始された日付／時刻 (エラーにより要求をキューに入れられない場合を除く)
- クライアントが接続した日付／時刻
- 要求に関連するデータベース名
- エラーが発生した場合の、エラーメッセージテキスト
- 要求の処理が終了した日付／時刻
- クライアントの IP アドレス
- 応答の長さ (バイト単位、ヘッダと本文を含む)
- HTTP 要求メソッド
- 要求に関連するリスナポート
- 要求が処理用のキューに入れられた日付と時刻 (エラーのために要求がキューに入れられなかった場合を除く)
- HTTP 応答のステータスコードおよび説明
- HTTP ステータスコード
- 現在のログエントリが書き込まれた日付／時刻
- 要求された URI
- 要求された HTTP のバージョン
- 要求の処理にかかった時間 (@F - @B)、またはエラーにより要求が処理されなかった場合は 0.000

LogMaxSize (LSIZE) プロトコルオプション

データベースサーバが Web 要求に関する情報を書き込むログファイルの最大サイズを制御します。

構文

```
{ LogMaxSize | LSIZE }=size[ k | m | g ]
```

使用可能なプロトコル

HTTP、HTTPS

指定可能な値

- **size** – この整数は、Web 要求情報が書き込まれるファイルの最大サイズを指定します。デフォルト値はバイト単位ですが、k、m、g のいずれかを使用してキロバイト、メガバイト、またはギガバイトの単位を指定できます。LogMaxSize が 0 の場合、ログファイルのサイズは無制限です。

デフォルト

0

備考

ログファイルが指定したサイズに達すると、名前が変更されて、別のログファイルが作成されます。

LogOptions 通信パラメータ [LOPT]

ログに記録されるメッセージの種類を制御します。

使用法

HTTP、HTTPS

値

NONE、OK、INFO、ERRORS、ALL、*status-codes*、REQHDRS、RESHDRS、HEADERS

デフォルト値

ALL

説明

メッセージの種類を表すキーワードと HTTP ステータスコードを指定できます。複数の値を指定したい場合は、カンマ区切りで指定します。

次のキーワードを使用して、ログに記録するメッセージの種類を制御できます。

- 何もログに記録しません。
- 正常に完了した要求をログに記録します (20x HTTP ステータスコード)。
- リダイレクトまたは無修正のステータスコードを返す要求をログに記録します (30x HTTP ステータスコード)。
- すべてのエラーをログに記録します (40x および 50x HTTP ステータスコード)。
- すべての要求をログ出力します。

次の一般的な HTTP ステータスコードも指定できます。ステータスコードを指定すると、そのステータスコードを返す要求をログに記録できます。

- OK
- Bad request
- Unauthorized
- Forbidden
- Not found
- Request timeout
- Not implemented
- Service unavailable

さらに、次のキーワードを指定すると、ログメッセージに関する詳細情報も記録できます。

- 要求を記録するときに、要求ヘッダもログファイルに書き込みます。
- 要求を記録するときに、応答ヘッダもログファイルに書き込みます。
- 要求のロギング時に、ログファイルに要求ヘッダと応答ヘッダの両方を書き込みます (REQHDRS、RESHDRS と同様)。

MaxConnections 通信パラメータ [MAXCONN]

サーバで許可される同時接続の数を指定します。

使用法

HTTP、HTTPS

値

Number

デフォルト値

ライセンスのある接続の数

説明

0を指定すると、無制限になります。

MaxRequestSize (MAXSIZE) プロトコルオプション

データベースサーバで許可できる最大要求サイズを指定します。

構文

```
{ MaxRequestSize | MAXSIZE }=size[ k | m | g ]
```

使用可能なプロトコル

HTTP、HTTPS

指定可能な値

- **size** – この整数は、データベースサーバで許可できる最大要求サイズを指定します。デフォルト値はバイト単位ですが、k、m、g のいずれかを使用してキロバイト、メガバイト、またはギガバイトの単位を指定できます。値 0 はこの制限を無効にしますが、細心の注意を払って使用してください。この制限がないと、悪質なクライアントがデータベースサーバに過負荷をかけたり、メモリ不足を引き起こしたりする可能性があります。

デフォルト

100k

備考

要求のサイズがこの制限を超えると、接続は閉じられ、クライアントは要求が大きすぎることを示す応答を返します。この値は要求サイズのみを制限するもので、応答サイズは制限しません。

例

次のコマンドは、150000 バイトまでのサイズの要求を受け入れるようサーバに指示します。

```
iqsrv16 -xs http(MaxRequestSize=150000)
```

MyIP 通信パラメータ [ME]

クライアントがアドレス情報の特定を試みるかどうかを制御します。

使用法

TCP/IP、HTTP、HTTPS

値

String

説明

MyIP (ME) パラメータは、複数のネットワークアダプタを使用するコンピュータに対して指定します。

各アダプタには IP アドレスがあります。デフォルトでは、SAP Sybase IQ は最初に見つかったネットワークカードを使用します。データベースサーバで複数のネットワークカードを使用する場合は、各カードのアドレスを MyIP (ME) パラメータで指定します。

IP アドレスとしてキーワード NONE が指定されている場合は、アドレス情報の決定は行われません。NONE キーワードは主に、この処理を行うと負荷がかかるオペレーティングシステム上のクライアント向けです (たとえば、複数のネットワー

クカードを使用しているマシンや、リモートアクセス (RAS) ソフトウェアとネットワークカードを使用しているマシンなど)。サーバで使用するものではありません。

Windows プラットフォームでは、このオプションを、複数の IP アドレスを持つマシンに対して何度でも使用できます。

複数の IP アドレスを指定するときはカンマで区切ります。オプションとして、ポート番号をコロンで区切って IP アドレスに追加できます。

Windows プラットフォームで IPv6 アドレスを指定する場合は、インタフェース識別子を使用します。UNIX プラットフォームでは、IPv6 アドレスのインタフェース識別子とインタフェース名の両方がサポートされています。Linux (カーネル 2.6.13 以降) では、インタフェース識別子が必須です。

例

- 次の Windows コマンドライン (すべて 1 行に入力) では、2 つのネットワークカードを使用し、そのうち 1 つには指定のポート番号を使用するようサーバに指示しています。

```
start_iq -x tcpip(MyIP=192.75.209.12:2367,192.75.209.32)
c:¥sybase¥IQ-16_0¥demo¥iqdemo.db
```

- 次のコマンドライン (すべて 1 行に入力) では、IPv6 ネットワークカードを使用するようサーバに指示しています。

```
start_iq -x tcpip(MyIP=fe80::5445:5245:444f)
"c:¥sybase¥IQ-16_0¥demo¥iqdemo.db"
```

- 次の接続文字列フラグメントは、クライアントがアドレス情報を決定しないように指定します。

```
...CommLinks= tcpip(MyIP=NONE)...
```

PreFetchOnOpen 通信パラメータ

このパラメータを有効にすると、プリフェッチ要求とカーソルオープン要求が一緒に送信されます。これにより、カーソルをオープンするたびにローフェッチのネットワーク要求を送信せずに済みます。

使用法
ODBC

値
YES, NO

デフォルト値
NO

説明

オープン時にプリフェッチを実行するには、カラムをバインドしておきます。PreFetchOnOpen を使用しているときに、カーソルのオープンから最初のフェッチまでの間にカラムを再バインドすると、パフォーマンスが低下します。

結果セットを返すクエリまたはストアードプロシージャのときに ODBC の SQLExecute または SQLExecDirect を呼び出すと、カーソルがオープンします。

このオプションを有効にすると、次の場合にパフォーマンスが向上します。

- ネットワークの遅延時間が長い
- アプリケーションがカーソルを開く要求と閉じる要求を多数送信する

ReceiveBufferSize 通信パラメータ [RCVBUFSZ]

TCP/IP プロトコルスタックによって使用されるバッファのサイズを設定します。ネットワーク上の LOB のパフォーマンスが重要である場合には、この値を増やします。

使用法

TCP/IP

値

Integer [**k** | **m** | **g**]

v

許可される最大値

1048576 バイト (1MB)

デフォルト値

マシンによって異なる。

説明

単位をキロバイト、メガバイト、またはギガバイトで指定するには、それぞれ **k**、**m**、**g** のいずれかを使用してください。

SendBufferSize 通信パラメータ [SNDBUFSZ]

TCP/IP プロトコルスタックによって使用されるバッファのサイズを設定します。

使用法

TCP/IP

値

Integer [**k** | **m** | **g**]

許可される最大値

1048576 バイト (1MB)

デフォルト値

コンピュータによって異なる。

説明

デフォルト値はバイト単位ですが、**k**、**m**、**g** のいずれかを使用してキロバイト、メガバイト、またはギガバイトを指定できます。ネットワーク上の LOB のパフォーマンスが重要である場合には、この値を増やします。

ServerPort 通信パラメータ [PORT]

データベースサーバの場合、**ServerPort** オプションでは、TCP/IP 通信のポート番号を指定します。データソースでは、**ServerPort** オプションは、データベースサーバが TCP/IP 通信を受信するときに使用する 1 つまたは複数のポートをクライアントに指示します。

使用法

TCP/IP (すべてのプラットフォーム)、HTTP、HTTPS

値

Integer

デフォルト値

TCP/IP のデフォルト値は **2638** です。HTTP のデフォルト値は 80 です。HTTPS のデフォルト値は 443 です。

説明

Internet Assigned Numbers Authority は、SAP Sybase IQ データベースサーバの TCP/IP 通信に対してポート番号 2638 を割り当てています。しかし、この予約ポートをアプリケーションで使うことが禁止されているわけではないため、データベースサーバと別のアプリケーションの間でアドレスが競合する可能性があります。

クライアントは、**ServerPort** パラメータで指定されたすべてのポートにブロードキャストしてサーバを検索します。**ServerPort** を指定しない場合は、デフォルトのポートである 2638 が使用されます。

データベースサーバは、**ServerPort** で指定されたポート番号を TCP/IP 接続に使用します。**ServerPort** を指定しない場合は、デフォルトのポートである 2638 が使用

されます。そのため、アプリケーションはポート番号を指定しないで IQ データベースサーバに接続できます。ただし HP-UX オペレーティングシステムでは、サーバが別のポートで開始されている場合には、ポート 2638 が受信に使われなくなります。

デフォルトでは、データベースサーバは HTTP と HTTPS の標準ポートである 80 と 443 で受信を行います。

例

1. Windows では、次のように、SAP Sybase IQ ネットワークサーバを起動します。

```
start_iq -x tcPIP c:¥sybase¥IQ-16_0¥demo¥iqdemo.db
```

ポート番号 2638 が取得されます。

2. 別のデータベースサーバを起動しようとします。

```
start_iq -x tcPIP c:¥sybase¥IQ-16_0¥demo¥iqdemo2.db
```

デフォルトのポートは現在割り付けられているので、サーバは別のポートで起動します。

3. ポート 2639 で別のデータベースサーバを起動します。

```
start_iq -x "tcPIP(ServerPort=2639)" c:¥sybase¥IQ-16_0¥demo¥iqdemo3.db
```

2639 が予約ポートでなく、他のアプリケーションに割り当てられていないかぎり、正常に処理されます。

4. すでにマシン上の別の Web サーバがポート 80 を使用しているか、またはこのような若いポート番号でサーバを起動するためのパーミッションがない場合は、8080 などの代替ポートを受信するサーバを起動します。

```
start_iq -xs http{port=8080} -n server3 web.db
```

Sessions 通信パラメータ

1 つの LAN アダプタからサーバが一度に通信できるクライアントの最大数を設定します。

使用法

NetBIOS。サーバ側のみ。

説明

デフォルト設定は、オペレーティングシステムによって異なります。値は整数で、最大値は 254 です。

NetBIOS ネットワークソフトウェアでは、使用できるコマンドの数がマシンごとに制限されています。SAP Sybase IQ は NetBIOS コマンドを使用しており、コマン

ドの数が上限に達した場合には、その数が Sessions パラメータの値より少ない場合でも、以降の接続が拒否されます。

デフォルト値

OS 固有。Windows では、デフォルトは 16 です。

例

次の文は、iqdemo というデータベースのサーバを起動し、最大 200 の NetBIOS 接続を可能にします。

```
start_iq -x netbios(sessions=200) iqdemo.db
```

TDS 通信パラメータ

サーバが TDS 接続を許可するかどうかを制御します。

使用法

TCP/IP、NamedPipes (サーバ側のみ)

値

YES, NO

デフォルト値

YES

説明

データベースサーバへの TDS 接続を許可しないようにするには、TDS を NO に設定します。サーバに対して暗号化された接続だけが行われるように保証するには、これらのポートオプションを使用して TDS 接続を禁止することが唯一の方法です。

例

次のコマンドは、TCP/IP プロトコルを使用するデータベースサーバを起動しますが、Open Client または jConnect アプリケーションからの接続を許可しません。

```
start_iq -x tcpip(TDS=NO) ...
```

Timeout 通信パラメータ [TO]

通信確立時と通信切断時の応答待ち時間を秒単位で指定します。

使用法

TCP/IP (すべてのプラットフォーム)、HTTP、HTTPS

値

Integer (秒単位)

許可される最大値

3600 秒

デフォルト値

5

説明

TCP/IP 通信の確立が困難である場合は、さらに長い時間を設定する必要があります。

HTTP または HTTPS アプリケーションでは、このパラメータを使用して、要求を受け取るときに許容される最大のアイドル時間を指定します。この限界に達した場合は接続がクローズされ、クライアント側に 408 REQUEST TIMEOUT が返されます。0 を指定するとアイドルタイムアウトが無効になりますが、その場合は十分な注意が必要です。この制限がないと、悪意のあるクライアントがサーバのリソースを消費し、他のクライアントからの接続を妨害するおそれがあります。

例

次のデータソースフラグメントでは、タイムアウト時間を 20 秒にして、TCP/IP 通信リンクのみを起動します。

```
...  
CommLinks=tcPIP (TO=20)  
...
```

VerifyServerName 通信パラメータ [Verify]

VerifyServerName 通信パラメータは、ホストへの接続時にサーバ名を検証するかどうかを指定します。

使用法

TCP/IP (クライアント側のみ)

値

YES, NO

デフォルト値

YES

説明

通常、このオプションは設定しません。このオプションは、マルチプレックスセカンダリサーバ間でクエリの負荷を分散させる必要がある場合に、これらのサーバに接続するときのみ使用します。

TCP で接続する場合、DoBroadcast=NONE パラメータを指定すると、クライアントによって TCP 接続が行われ、検出されたサーバと検索対象サーバの名前が一致しているかどうかを検証されます。VerifyServerName=NO を指定すると、サーバ名の検証が省略されます。これにより、IQ クライアントが SAP Sybase IQ サーバの IP アドレス/ポートしか知らない場合でも、そのサーバに接続できます。

接続文字列にはサーバ名の指定が必要ですが、この名前は無視されます。VerifyServerName (VERIFY) 通信パラメータが使用されるのは、DoBroadcast=NONE の場合にかぎられます。

下記の例に示すとおり、このオプションを NO に設定すると、特定の IP アドレスやポート番号に接続するよう指定できます。この IP アドレスとポート番号は、IQ クライアントと IQ サーバのゲートウェイとして機能するロードバランスマシンのものです。

例

このオプションを使用するには、クライアントマシンの ODBC Administrator で新しい ODBC DSN を作成し、次のようにパラメータを指定します。

- [データベース] タブで、すべてのセカンダリサーバへの接続で使用される汎用的なサーバ名を指定します (例: qserv)。サーバ名は必須ですが、[ネットワーク] タブのパラメータのために無視されます。
- [ネットワーク] タブで [TCP/IP] チェックボックスをオンにし、テキストボックスに次の情報を入力します。

```
host=hostname;port=port#;DOBROADCAST=NONE;VERIFY=NO
```

次に例を示します。

```
host=hostname;port=2222;DOBROADCAST=NONE;VERIFY=NO
```

IQ クライアントがこの DSN に接続するときには、ロードバランサが特定のセカンダリサーバへの接続をそのマシンの負荷に応じてディスパッチします。

付録：SQL 文とオプションのリファレンス

特定の SQL 文とデータベースオプションには、SAP Sybase IQ データベース管理をサポートするための特別な構文があります。

データベースオプション

データベースオプションは、データベースの動作を変更します。

データベースオプションの範囲と継続期間

オプションは、パブリック、ユーザ、テンポラリの 3 つの範囲レベルで設定できます。

temporary オプションは、user と public より優先度が高くなっています。ユーザレベルのオプションは public より優先度が高くなっています。ユーザレベルのオプションを現在のユーザに対して設定すると、対応する temporary オプションも同じように設定されます。

COMMIT の動作などの一部のオプションは、データベース全体を範囲とします。これらのオプションを設定するには、DBA パーミッションが必要です。ISOLATION_LEVEL などの他のオプションは、現在の接続のみに適用でき、特別なパーミッションは必要ありません。

オプション設定への変更が有効になる時点は、オプションによって異なります。RECOVERY_TIME などのグローバルオプションの変更は、サーバの次回起動時に有効になります。次のリストは、サーバの再起動時に有効化されるオプションの一部を示しています。

サーバの再起動を必要とするデータベースオプション
CACHE_PARTITIONS
CHECKPOINT_TIME
OS_FILE_CACHE_BUFFERING
OS_FILE_CACHE_BUFFERING_TEMPDB
PREFETCH_BUFFER_LIMIT
PREFETCH_BUFFER_PERCENT

サーバの再起動を必要とするデータベースオプション
RECOVERY_TIME
SWEEPER_THREADS_PERCENT
WASH_AREA_BUFFERS_PERCENT

一般に、現在の接続にのみ影響を与えるオプションはすぐに有効になります。たとえば、トランザクションの途中でもオプション設定を変更できます。

警告！ カーソルが開いている間にオプションを変更すると、結果の信頼性が損なわれる場合があります。たとえば、カーソルが開いているときに DATE_FORMAT を変更しても、次のローのフォーマットは変わりません。カーソルの取得方法によっては、ユーザが指定した変更が有効になるまでにいくつかのローが処理されてしまう場合があります。

DEDICATED_TASK オプション

要求処理のタスクが、1つの接続からの要求だけを処理するようにします。

指定できる値

ON、OFF

デフォルト値

OFF

スコープ

オプションは、現在の接続の存続期間において、個々の接続または PUBLIC ロールにテンポラリオプションとしてのみ設定できます。

このオプションを設定するには、SET ANY SYSTEM OPTION システム権限が必要です。すぐに有効になります。

備考

DEDICATED_TASK 接続オプションを ON に設定すると、要求処理タスクは、その接続の要求処理専用となります。このオプションを有効にして接続を事前に確立しておくことで、データベースサーバが応答しなくなった場合にデータベースサーバの状態に関する情報を収集できます。

LOG_CONNECT オプション

ユーザ接続のロギングを制御します。

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) レベルでのみ設定できます。

このオプションを設定するには、SET ANY SECURITY OPTION システム権限が必要です。すぐに有効になります。

備考

このオプションが ON の場合、ユーザが SAP Sybase IQ データベースに接続するたび、または接続を切断するたびに、IQ メッセージログ (.iqmsg ファイル) にメッセージが書き込まれます。

注意： ユーザの接続時にこのオプションが OFF (接続のロギングが無効) に設定されており、そのユーザが切断する前に ON にされた場合、メッセージログにはユーザの接続は記録されず、切断だけが記録されます。

MIN_PASSWORD_LENGTH オプション

データベースの新しいパスワードの長さの最小値を設定します。

指定できる値

0 以上の整数

値はバイト単位です。シングルバイト文字セットの場合、これは文字数と同じになります。

デフォルト値

3 文字

スコープ

オプションは、データベース (PUBLIC) レベルでのみ設定できます。

このオプションを設定するには、SET ANY SECURITY OPTION システム権限が必要です。すぐに有効になります。

備考

このオプションを使用すると、すべての新しいパスワードの長さに最小値が設定され、セキュリティが強化されます。既存のパスワードには影響しません。

例

新しいパスワードの最小長を 6 バイトに設定します。

```
SET OPTION PUBLIC.MIN_PASSWORD_LENGTH = 6
```

VERIFY_PASSWORD_FUNCTION オプション

パスワードルールの実装に使用できる、ユーザ指定の認証の関数を指定します。

指定できる値
文字列

デフォルト
" (空の文字列)。 (パスワードが設定されている場合、関数は呼び出されません)

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 SECURITY 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
VERIFY_PASSWORD_FUNCTION オプションの値が有効な文字列に設定されている場合、**GRANT CONNECT TO** *userid* **IDENTIFIED BY** *password* 文によってオプション値で指定された関数が呼び出されます。

ユーザが関数を上書きするのを防ぐために、オプション値は、*owner.function_name* の形式で指定する必要があります。

関数は、次の 2 つのパラメータを取ります。

- *user_name* VARCHAR(128)
- *new_pwd* VARCHAR(255)

戻り値のデータ型は、VARCHAR(255) です。

VERIFY_PASSWORD_FUNCTION が設定されている場合、**GRANT CONNECT** 文で複数のユーザ ID とパスワードを指定することはできません。

例
次のサンプルコードは、テーブルと関数を定義し、いくつかのログインポリシーオプションを設定します。同時に、パスワードにおける特定の種類の文字の要求、パスワードの再使用の禁止、パスワードの有効期限の適用などを含む詳細なパスワード規則も実装します。ユーザ ID の作成時、またはパスワードの変更時には、

verify_password_function オプションを使用してデータベースサーバによってこの関数が呼び出されます。アプリケーションでは post_login_procedure オプションによって指定されるプロシージャが呼び出され、有効期限前にパスワードの変更が必要であることを通知することができます。

```
-- only DBA should have privileges on this table
CREATE TABLE DBA.t_pwd_history(
    pk          INT          DEFAULT AUTOINCREMENT PRIMARY KEY,
    user_name   CHAR(128),   -- the user whose password is set
    pwd_hash    CHAR(32) );  -- hash of password value to detect
                                -- duplicate passwords

-- called whenever a non-NULL password is set
-- to verify the password conforms to password rules
CREATE FUNCTION DBA.f_verify_pwd( uid      VARCHAR(128),
                                new_pwd   VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
    -- enforce password rules
    -- enforce minimum length (can also be done with
    -- min_password_length option)
    IF length( new_pwd ) < 6 THEN
        RETURN 'password must be at least 6 characters long';
    END IF;

    -- number of lowercase characters IN new_pwd
    SELECT count(*) INTO num_lower_chars
        FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';

    -- enforce rules based on characters contained in new_pwd
    IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
        < 1 THEN
        RETURN 'password must contain at least one numeric digit';
    ELSEIF length( pwd_alpha_only ) < 2 THEN
        RETURN 'password must contain at least two letters';
    ELSEIF num_lower_chars = 0
        OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
        RETURN 'password must contain both upper- and lowercase
characters';
    END IF;

    -- not the same as any user name
    -- (this could be modified to check against a disallowed words
table)
    IF EXISTS( SELECT * FROM SYS.SYSUSER
                WHERE lower( user_name ) IN
( lower( pwd_alpha_only ),
                                lower( new_pwd ) ) ) THEN
        RETURN 'password or only alphabetic characters in password '
||
        'must not match any user name';
    END IF;

    -- not the same as any previous password for this user
    IF EXISTS( SELECT * FROM t_pwd_history
```

```
WHERE user_name = uid
      AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
RETURN 'previous passwords cannot be reused';
END IF;

-- save the new password
INSERT INTO t_pwd_history( user_name, pwd_hash )
      VALUES( uid, hash( uid || new_pwd, 'md5' ) );

RETURN( NULL );
END;

ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';

-- All passwords expire in 180 days. Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
ALTER LOGIN POLICY DEFAULT password_life_time = 180;

-- If an application calls the procedure specified by the
-- post_login_procedure option, then the procedure can be used to
-- warn the user that their password is about to expire. In
particular,
-- Interactive SQL calls the post_login_procedure.
ALTER LOGIN POLICY DEFAULT password_grace_time = 30;
```

オプションをオフにするには、空の文字列を指定します。

```
SET OPTION PUBLIC.VERIFY_PASSWORD_FUNCTION = ''
```

SQL 文

SQL 文はデータ定義タスクとデータ操作タスクを実行します。

ALTER DBSPACE 文

既存の DB 領域に対し、読み書きのモードの変更、サイズの変更、または領域の拡張を行います。

クイックリンク：

「パラメータ」 (445 ページ)

「例」 (448 ページ)

「使用法」 (448 ページ)

「標準」 (449 ページ)

「パーミッション」 (449 ページ)

構文

```

ALTER DBSPACE dbspace-name
  { ADD new-file-spec [, new-file-spec ... ]
  | DROP FILE logical-file-name [, FILE logical-file-name ... ]
  | RENAME TO newname | RENAME 'new-file-pathname'
  | READONLY | READWRITE
  | ONLINE | OFFLINE
  | STRIPING{ ON | OFF }
  | STRIPESIZEKB size-in-KB
ALTER FILE file-name
  { READONLY | [ FORCE ] READWRITE }
  | SIZE file-size [ KB | MB | GB | TB ]
  | ADD file-size [ KB | MB | GB | TB | PAGES ] }
RENAME PATH 'new-file-pathname'
RENAME TO newname

new-file-spec - (back to Syntax)
  FILE logical-file-name 'file-path' iq-file-opts

iq-file-opts - (back to new-file-spec)
  [ [ SIZE ] file-size ]
  ...[ KB | MB | GB | TB ] ]
  [ RESERVE reserve-size [ KB | MB | GB | TB ] ]

```

パラメータ

(先頭に戻る) (444 ページ)

- **ADD** – 指定した DB 領域に 1 つまたは複数のファイルを追加します。各ファイルに対して、DB ファイル名と物理ファイルパスは必要かつユニークでなければなりません。ファイルは IQ メイン、IQ 共有テンポラリ、IQ テンポラリ、またはキャッシュの各 DB 領域に追加できます。ファイルを読み込み専用の DB 領域に追加できますが、DB 領域は読み込み専用のままになります。ファイルをマルチプレックス共有テンポラリ DB 領域に追加できますが、読み込み専用モードの場合のみです (ADD FILE のデフォルト)。

カタログ DB 領域には 1 つのファイルしか含むことができないため、ADD FILE をカタログ DB 領域で使用できない場合があります。

RLV DB 領域の場合は、シンプレックスサーバのみで ADD FILE を使用します。マルチプレックス RLV DB 領域にはファイルを追加できません。

キャッシュ DB 領域の場合は、マルチプレックスサーバまたはシンプレックスサーバで ADD FILE を使用します。

ALTER FILE 句で使用する場合、ファイルのサイズを、ページ、キロバイト (KB)、メガバイト (MB)、ギガバイト (GB)、テラバイト (TB) の単位で拡張します。デフォルトは MB です。ファイルのサイズは、フリーリスト (アロケー

ションマップ)に十分な余裕があり、DB 領域に十分な領域が確保されていなければ追加できません。

- **DROP FILE** – 指定したファイルを IQ DB 領域から削除します。ファイルは空である必要があります。最後のファイルを指定の DB 領域から削除することはできません。DB 領域に 1 つのファイルしか含まれていない場合は、**DROP DBSPACE** を代用します。
- **RENAME TO** – **DROP FILE** 句とともに使用すると、単一ファイルを含む DB 領域のパス名を変更します。**RENAME PATH** 句と意味的に同じです。DB 領域に複数のファイルが含まれている場合はエラーが返されます。
IQ_SYSTEM_MAIN、IQ_SYSTEM_MSG、IQ_SYSTEM_TEMP、
IQ_SHARED_TEMP、または SYSTEM の名前は変更できません。

ALTER FILE 句とともに使用すると、指定されたファイルの論理名を新しい名前に変更します。新しい名前はデータベース内でユニークにします。

- **READONLY** – **DROP** 句とともに使用すると、IQ_SYSTEM_MAIN、IQ_SYSTEM_TEMP、IQ_SYSTEM_MSG、IQ_SHARED_TEMP、および SYSTEM 以外の DB 領域を読み込み専用に変更します。DB 領域に現在割り当てられたオブジェクトへの DML 変更を禁止します。キャッシュ DB 領域および IQ メインストア内の DB 領域に対してのみ使用できます。

ALTER FILE 句とともに使用すると、指定したファイルを読み込み専用に変更します。このファイルは、IQ メイン DB 領域に関連付けられている必要があります。IQ_SHARED_TEMP のファイルは **READONLY** ステータスに変更することはできません。

- **READWRITE** – **DROP FILE** 句とともに使用すると、DB 領域を読み書き用に変更します。DB 領域はオンラインである必要があります。キャッシュ DB 領域および IQ メインストア内の DB 領域に対してのみ使用できます。

ALTER FILE 句とともに使用すると、指定したキャッシュ DB 領域、IQ メインまたはテンポラリストアの DB 領域を読み書き用に変更します。このファイルは、キャッシュ DB 領域、IQ メインまたはテンポラリの DB 領域に関連付けられている必要があります。

- **ONLINE** – オフライン DB 領域と関連するすべてのファイルをオンラインにします。キャッシュ DB 領域内および IQ メインストア内の DB 領域に対してのみ使用できます。
- **OFFLINE** – オンライン読み込み専用 DB 領域と関連するすべてのファイルをオフラインにします (DB 領域が読み書き用である、すでにオフラインになっている、あるいはキャッシュ DB 領域または IQ メインストア内にはない場合は、エ

ラーが返されます)。キャッシュ DB 領域内または IQ メインストア内の DB 領域に対してのみ使用できます。

- **STRIPING** – DB 領域のディスクストライピングを指定どおりに変更します。ディスクストライピングがオンに設定されている場合、データは DB 領域内の各ファイルからラウンドロビン方式で割り付けられます。たとえば、最初に書き込みがあったデータベースページが最初のファイルへ、2 番目に書き込みがあったページが指定の DB 領域内の次のファイルへ、というようになります。読み込み専用の DB 領域はスキップされます。
- **STRIPESIZEKB** – ディスクストライピングアルゴリズムが指定した DB 領域の次のストライプに移動する前に、各ファイルに書き込むデータ量をキロバイト (KB) で指定します。
- **FORCE READWRITE** – ALTER FILE 句とともに使用すると、セカンダリノードに既知のファイルステータスの問題がある場合でも、指定した共有テンポラリストアの DB ファイルのステータスを読み書き用に変更します。ファイルは、IQ メイン、共有テンポラリ、またはテンポラリの DB 領域に関連付けられている可能性があります。IQ_SYSTEM_MAIN とユーザメインの新しい DB ファイルは読み書き用に作成されるので、この句は、共有テンポラリ DB 領域にのみ影響します。
- **SIZE** – ファイルの新しいサイズを、キロバイト (KB)、メガバイト (MB)、ギガバイト (GB)、テラバイト (TB) の単位で指定します。デフォルトはメガバイトです。DB 領域のサイズは、フリーリスト (アロケーションマップ) に十分な余裕があり、DB 領域に十分な領域が確保されていなければ増加できません。DB 領域のサイズを減少できるのは、切り取られる部分が未使用である場合だけです。
- **RENAME PATH** – ALTER FILE 句とともに使用すると、指定したファイルに関連付けられたファイルのパス名を変更します。この句はファイルを古いパスの代わりに新しいファイルパスに関連付けるだけであり、オペレーティングシステムのファイル名を実際に変更するわけではありません。ファイル名を変更するには、オペレーティングシステム経由で実行する必要があります。ファイルのパス名を変更するには、DB 領域がオフラインである必要があります。新しいパスは、DB 領域をオンラインで変更するか、データベースを再起動する場合に使用されます。

新しいパスにアクセスできない場合は、データベースを起動できないため、IQ_SYSTEM_MAIN 内でファイルのパス名を変更する必要はありません。

IQ_SYSTEM_MAIN 内のファイルのパス名を変更する必要がある場合は、ファイルを読み込み専用を設定し、ファイルを空にして削除してから、ファイルに新しいパス名を付けて追加し直します。DB ファイルへの物理ファイルパスは、一重引用符で囲みます。

例

(先頭に戻る) (444 ページ)

- **例 1** – DspHist という名前の DB 領域のモードを、READONLY に変更します。

```
ALTER DBSPACE DspHist READONLY
```

- **例 2** – サイズが 500MB のファイル FileHist3 を追加して、DB 領域 DspHist に 500MB を追加します。

```
ALTER DBSPACE DspHist  
ALTER FILE FileHist3 ADD 500MB
```

- **例 3** – UNIX システムで、2つの 500MB ファイルを DB 領域 DspHist に追加します。

```
ALTER DBSPACE DspHist ADD  
FILE FileHist3 '/History1/data/file3' SIZE 500MB,  
FILE FileHist4 '/History1/data/file4' SIZE 500
```

- **例 4** – DB 領域 IQ_SYSTEM_TEMP のサイズを、2GB だけ増加します。

```
ALTER DBSPACE IQ_SYSTEM_TEMP ADD 2 GB
```

- **例 5** – 2つのファイルを DB 領域 DspHist から削除します。ファイルはどちらも空である必要があります。

```
ALTER DBSPACE DspHist  
DROP FILE FileHist2, FILE FileHist4
```

- **例 6** – DB 領域 IQ_SYSTEM_MAIN のサイズを、1000 ページだけ増加します (ADD 句のデフォルトはページです)。

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD 1000
```

- **例 7** – ファイルをキャッシュ DB 領域 myDAS に追加します。

```
ALTER DBSPACE myDAS ADD FILE iqdas2 'sampledb.iqcache' size 1024
```

- **例 8** – DB ファイル iqdas2 をキャッシュ DB 領域 myDAS から削除します。

```
ALTER DBSPACE myDAS DROP FILE iqdas2
```

- **例 9** – キャッシュ DB 領域 myDAS を無効にします。

```
ALTER DBSPACE myDAS OFFLINE
```

- **例 10** – myDAS キャッシュ DB 領域の DB ファイル iqdas2 を読み込み専用にします。

```
ALTER DBSPACE myDAS ALTER FILE iqdas2 READONLY
```

使用法

(先頭に戻る) (444 ページ)

ALTER DBSPACE は、読み書きのモードの変更、オンライン/オフライン状態の変更、ファイルサイズの変更、DB 領域名の変更、ファイルの論理名またはファイルパスの変更、または DB 領域ストライピングパラメータの設定を行います。既存の DB 領域の詳細については、**sp_iqdbspace** プロシージャ、**sp_iqdbspaceinfo** プロシージャ、**sp_iqfile** プロシージャ、**sp_iqdbspaceobjectinfo**、および **sp_iqobjectinfo** を実行してください。DB 領域名と DB ファイル名では、大文字と小文字は常に区別されません。CASE RESPECT を指定してデータベースが作成され、大文字と小文字が区別されるファイルがオペレーティングシステムでサポートされている場合、物理ファイルパスの大文字と小文字は区別されます。そうでない場合、ファイルパスの大文字と小文字は区別されません。

DB 領域と DB ファイルの名前は、引用符で囲まないか、二重引用符で囲みます。

Windows でパスを指定する場合、円記号 (¥) の後に n または x がある場合は円記号を 2 つ重ねます。こうすることで、SQL の文字列の規則に従って、改行文字 (¥n) または 16 進数字 (¥x) として解釈されるのを回避できます。円記号は常にエスケープした方が安全です。

関連する動作：

- オートコミット
- 自動チェックポイント
- モードを READONLY に変更すると、DB 領域上のデータベースの内部構造が読み書き用であるいずれかの DB 領域へ直ちに移動されます。

標準

(先頭に戻る) (444 ページ)

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

パーミッション

(先頭に戻る) (444 ページ)

MANAGE ANY DBSPACE システム権限が必要です。

ALTER INDEX 文

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インデックスの名前、およびインデックスと外部キーがユーザによって明示的に作成された場合の外部キーのロール名を変更したり、カタログ保存テーブル上でインデックスのクラスタ化された内容を変更したりします。キー制約を適用するために作成したインデックスの名前は変更できないという点に注意してください。

クイックリンク：

「パラメータ」 (450 ページ)

「例」 (450 ページ)

「使用法」 (451 ページ)

「標準」 (451 ページ)

「パーミッション」 (452 ページ)

構文

```
ALTER { INDEX index-name
      | [ INDEX ] FOREIGN KEY role-name
      | [ INDEX ] PRIMARY KEY
      | ON [owner.]table-name { rename-clause | move-clause | cluster-clause }
```

rename-clause - (構文に戻る)
RENAME TO | **AS** *new-name*

move-clause - (構文に戻る)
MOVE TO *dbspace-name*

cluster-clause - (構文に戻る)
CLUSTERED | **NONCLUSTERED**

パラメータ

(先頭に戻る) (449 ページ)

- **ON** - 名前を変更するインデックスや外部キーを含むテーブルの名前を指定します。
- **RENAME** - インデックスまたは外部キーロールの新しい名前を指定します。
- **MOVE** - 指定したインデックス、一意性制約、外部キー、またはプライマリキーを指定した DB 領域に移動します。一意性制約キーまたは外部キーでは、一意なインデックス名を指定する必要があります。
- **cluster-clause** - インデックスを **CLUSTERED** と **NONCLUSTERED** のどちらに変更するかを指定します。カタログストアテーブルのみに適用され、特定のテーブル上で 1 つのインデックスのみ、クラスタ化できます。

例

(先頭に戻る) (449 ページ)

- **例 1** - プライマリキー (c5 の HG) を DB 領域 Dsp4 から Dsp8 に移動します。

```
CREATE TABLE foo (  
  c1 INT IN Dsp1,
```

```

c2 VARCHAR(20),
c3 CLOB IN Dsp2,
c4 DATE,
c5 BIGINT,
PRIMARY KEY (c5) IN Dsp4) IN Dsp3);
CREATE DATE INDEX c4_date ON foo(c4) IN Dsp5;
ALTER INDEX PRIMARY KEY ON foo MOVE TO Dsp8;

```

- **例 2** – DATE インデックスを Dsp5 から Dsp9 に移動します。

```
ALTER INDEX c4_date ON foo MOVE TO Dsp9
```

- **例 3** – テーブル `jal.mytable` のインデックス `COL1_HG_OLD` の名前を `COL1_HG_NEW` に変更します。

```
ALTER INDEX COL1_HG_OLD ON jal.mytable
RENAME AS COL1_HG_NEW
```

- **例 4** – テーブル `dba.Employees` の外部キーのロール名 `ky_dept_id` を `emp_dept_id` に変更します。

```
ALTER INDEX FOREIGN KEY ky_dept_id
ON dba.Employees
RENAME TO emp_dept_id
```

使用法

(先頭に戻る) (449 ページ)

新しい DB 領域に対する CREATE 権限があり、かつテーブル所有者であるか、MANAGE ANY DBSPACE システム権限がある必要があります。

注意： ローカルテンポラリテーブルのインデックスを変更しようとする時、「インデックスが見つかりません。」というエラーが返されます。デフォルトインデックス (FP) などユーザ作成でないインデックスを変更しようとする時、「インデックスを変更できません。所有者のタイプが USER の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インデックスのみを変更できます。」というエラーが返されます。

関連する動作：

- オートコミット。Interactive SQL の、[結果] ウィンドウ枠の [結果] タブがクリアされます。現在接続しているすべてのカーソルを閉じます。

標準

(先頭に戻る) (449 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。

パーミッション

(先頭に戻る) (449 ページ)

move-clause マテリアライズドビューの場合は、次のいずれかが必要です。

- **MANAGE ANY DBSPACE** システム権限
- **ALTER ANY INDEX** システム権限
- **ALTER ANY OBJECT** システム権限
- マテリアライズドビューを所有し、さらに次のいずれかがある。
 - **CREATE ANY OBJECT** システム権限
 - ターゲット DB 領域に対する **CREATE** 権限

move-clause その他のインデックスの場合は、次のいずれかが必要です。

- **MANAGE ANY DBSPACE** システム権限
- **ALTER ANY INDEX** システム権限
- **ALTER ANY OBJECT** システム権限
- 基本となるテーブルを所有しているか、次のいずれかとともに **REFERENCES** 権限がある
 - **CREATE ANY OBJECT** システム権限
 - ターゲット DB 領域に対する **CREATE** 権限

マテリアライズドビューの **cluster-clause** は以下のいずれかを必要とします。

- **ALTER ANY INDEX** システム権限
- **ALTER ANY OBJECT** システム権限
- マテリアライズドビューの所有者である。

他のすべてのインデックスの **cluster-clause** は以下のいずれかを必要とします。

- **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
- テーブルの所有者である。

それ以外のすべての句は次のいずれかを必要とします。

- **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
- 基本となるテーブルの所有者である。

ALTER LOGICAL SERVER 文

データベース内の既存のユーザ定義による論理サーバの設定を変更します。次の文では、論理サーバで共有される物理ノード間で、一貫した共有システムテンポラリストア設定が強制的に適用されます。

クイックリンク：

「パラメータ」 (453 ページ)

「例」 (453 ページ)

「使用法」 (454 ページ)

「パーミッション」 (454 ページ)

構文

```
ALTER LOGICAL SERVER logical-server-name
    { alter-ls-clause } [ WITH STOP SERVER ]

alter-ls-clause - (back to Syntax)
    { ADD MEMBERSHIP '(' { ls-member, ... } ') '
    | DROP MEMBERSHIP '(' { ls-member, ... } ') '
    | POLICY policy-name }

ls-member - (back to alter-ls-clause)
    FOR LOGICAL COORDINATOR | mpx-server-name
```

パラメータ

(先頭に戻る) (453 ページ)

- **logical-server-name** – 既存のユーザ定義論理サーバの名前を指します。
- **WITH STOP SERVER – TEMP_DATA_IN_SHARED_TEMP** データベースオプションが直接的または間接的に変更された場合に、論理サーバ内のすべてのサーバを自動的に停止します。

適用対象

マルチプレックスのみ。

例

(先頭に戻る) (453 ページ)

- **例 1** – マルチプレックスノード *n1* と *n2* を論理サーバ *ls1* に追加して、ユーザ定義の論理サーバを変更します。

```
ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (n1, n2)
```

- **例 2** – コーディネータの論理メンバーシップを追加し、現在のコーディネータノード n1 の名前付きメンバーシップを論理サーバ ls1 から削除します。

```
ALTER LOGICAL SERVER ls1 ADD MEMBERSHIP (FOR LOGICAL COORDINATOR)  
ALTER LOGICAL SERVER ls1 DROP MEMBERSHIP (n1)
```

- **例 3** – 論理サーバ ls2 の論理サーバポリシーを、ポリシー lsp1 に変更します。

```
ALTER LOGICAL SERVER ls2 POLICY lsp1
```

使用法

(先頭に戻る) (453 ページ)

SYS.ISYSIQLSMEMBER システムテーブルには、論理サーバメンバーシップの定義が格納されています。

論理サーバに追加、または論理サーバから削除されたメンバーノードは、そのノードで **ALTER LOGICAL SERVER** に対応する TLV ログが適用されるまで、論理サーバ接続の受け入れ、または拒否をしません。論理サーバの既存の接続は、ノードが論理サーバから削除されてもそのノードで稼動し続けますが、これらの接続の分散処理は停止します。

次の場合には、エラーが返されます。

- **ADD MEMBERSHIP** 句で指定された ls-member のいずれかが、すでに論理サーバのメンバーである場合。
- **DROP MEMBERSHIP** 句で指定された ls-member のいずれかが、論理サーバの既存のメンバーでない場合。
- 論理サーバのメンバーシップ変更により、あるノードが単一のログインポリシーに割り当てられた複数の論理サーバに属することになる場合。1つのログインポリシー内で論理サーバのメンバーシップの重複は許可されません。

パーミッション

(先頭に戻る) (453 ページ)

MANAGE MULTIPLEX システム権限が必要です。

ALTER LS POLICY 文

ルート論理サーバポリシーまたはユーザが作成した論理サーバポリシーのオプション値の一部またはすべてを変更します。次の文では、論理サーバで共有される物理ノード間で、一貫した共有システムテナンタリストア設定が強制的に適用されます。

クイックリンク：

「パラメータ」 (455 ページ)

「例」 (455 ページ)

「使用法」 (456 ページ)

「パーミッション」 (457 ページ)

構文

```
ALTER LS POLICY ls-policy-name ls-option-value-list
  [ WITH STOP SERVER ]
```

ls-option-value-list - (back to Syntax)
 { **ls-option-name** = *ls-policy-option-value* } ...

ls-option-name - (back to ls-option-value-list)

```
ALLOW_COORDINATOR_AS_MEMBER
| DQP_ENABLED
| ENABLE_AUTOMATIC_FAILOVER
| LOGIN_REDIRECTION
| REDIRECTION_WAITERS_THRESHOLD
| TEMP_DATA_IN_SHARED_TEMP
```

パラメータ

(先頭に戻る) (454 ページ)

- **ls-policy-name** – 論理サーバポリシーの名前。ルートを指定してルート論理サーバポリシーを変更します。
- **ls-option-value-list** – 論理サーバポリシーオプションの名前。オプションのリストについては、「使用法 (456 ページ)」を参照してください。
- **ls-policy-option-value** – 指定されていないオプションの値はすべて、ルート論理サーバポリシーから継承されます。「使用法 (456 ページ)」を参照してください。
- **WITH STOP SERVER** – **TEMP_DATA_IN_SHARED_TEMP** オプションが直接的または間接的に変更された場合、論理サーバ内のすべてのサーバを自動的に停止します。

適用対象

マルチプレックスのみ。

例

(先頭に戻る) (454 ページ)

- **例 1** – 論理サーバポリシーを変更します。

```
ALTER LS POLICY root
ALLOW_COORDINATOR_AS_MEMBER=ON
```

- **例 2** – 論理サーバポリシーを変更し、オプション値が変更された場合にサーバが自動的に停止するようにします。

```
ALTER LS POLICY root
TEMP_DATA_IN_SHARED_TEMP=ON WITH STOP SERVER
```

使用法

(先頭に戻る) (454 ページ)

より小さな IQ_SYSTEM_TEMP DB 領域が必要な場合は、TEMP_DATA_IN_SHARED_TEMP を ON に設定します。こうすることによって、テンポラリデータが IQ_SYSTEM_TEMP ではなく、IQ_SHARED_TEMP に書き込まれます。ただし、分散クエリ処理環境では、DQP_ENABLED と TEMP_DATA_IN_SHARED_TEMP の両方を ON に設定すると、IQ_SHARED_TEMP 内の追加データで SAN が飽和する可能性があります。これは、IQ_SHARED_TEMP に対する追加の I/O 処理が DQP のパフォーマンスに悪影響を及ぼす可能性があるためです。

オプション	説明
ALLOW_COORDINATOR_AS_MEMBER	<p>ROOT 論理サーバポリシーに対してのみ設定可能。ON (デフォルト) の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーにできる。OFF の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーとして使用できない。</p> <ul style="list-style-type: none"> • 値 – ON、OFF • デフォルト – ON
DQP_ENABLED	<p>0 に設定すると、クエリ処理が分散されない。1 (デフォルト) に設定した場合、書き込み可能な共有テンポラリファイルが存在するかぎり、クエリ処理が分散される。2 に設定した場合、クエリ処理がネットワーク上で分散され、共有テンポラリストアは使用されない。</p> <ul style="list-style-type: none"> • 値 – 0、1、2 • デフォルト – 1

オプション	説明
ENABLE_AUTO- MATIC_FAILOVER	<p>ROOT 論理サーバポリシーに対してのみ設定可能。ON は、指定のログインポリシーが適用される論理サーバの自動フェールオーバーを有効化。OFF (デフォルト) は、論理サーバレベルの自動フェールオーバーを無効にし、手動フェールオーバーを可能にする。デフォルト値の設定に戻すには DEFAULT を指定する。</p> <ul style="list-style-type: none"> • 値 – ON、OFF、DEFAULT • デフォルト – OFF
LOGIN_REDIR- ECTION	<p>ON は、指定のログインポリシーが適用される論理サーバのログインリダイレクションを有効化。OFF (デフォルト) は、論理サーバレベルのログインリダイレクションを無効にし、外部接続管理を可能にする。</p> <ul style="list-style-type: none"> • 値 – ON、OFF • デフォルト – OFF
REDIRECTION_ WAITERS_ THRESHOLD	<p>SAP Sybase IQ がこの論理サーバへの接続を別のサーバにリダイレクトする前に、キューに入れることが可能な接続数を指定する。任意の整数値に設定可能。デフォルトは 5。</p> <ul style="list-style-type: none"> • 値 – 整数 • デフォルト – 5
TEMP_DATA_IN_ SHARED_TEMP	<p>ON の場合、共有テンポラリストアに 1 つ以上の読み書きファイルが追加されていると、すべてのテンポラリテーブルデータと対象スクラッチデータが共有テンポラリストアに書き込まれる。このオプションを設定した後、または、読み書きファイルを共有テンポラリストアに追加した後は、すべてのマルチプレックスノードの再起動が必要 (共有テンポラリストアに読み書きファイルがない場合、または、ノードを再起動しなかった場合は、データが IQ_SYSTEM_TEMP に書き込まれる)。OFF (デフォルト) の場合は、すべてのテンポラリテーブルデータとスクラッチデータがローカルテンポラリストアに書き込まれる。</p> <ul style="list-style-type: none"> • 値 – ON、OFF • デフォルト – OFF

パーミッション

(先頭に戻る) (454 ページ)

MANAGE MULTIPLEX システム権限が必要です。

LS ポリシーオプション

ルート LS ポリシーとユーザ定義 LS ポリシーで使用可能なオプションを次に示します。

オプション	説明
ALLOW_COORDINATOR_AS_MEMBER	<p>ROOT 論理サーバポリシーに対してしか設定できない。ON (デフォルト) の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーにすることができる。OFF の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーとして使用できない。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - ON
DQP_ENABLED	<p>0 に設定した場合、クエリ処理が分散されない。1 (デフォルト) に設定した場合、書き込み可能な共有テンポラリファイルが存在するかぎり、クエリ処理は分散される。2 に設定した場合、クエリ処理がネットワーク上で分散され、共有テンポラリストアは使用されない。</p> <ul style="list-style-type: none"> • 値 - 0, 1, 2 • デフォルト - 1
LOGIN_REDIRECTION	<p>ON の場合、指定のログインポリシーが適用される論理サーバのログインリダイレクションが有効になる。OFF (デフォルト) の場合、論理サーバレベルのログインリダイレクションが無効になり、外部接続管理が可能になる。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - OFF
REDIRECTION_WAITERS_THRESHOLD	<p>SAP Sybase IQ がこの論理サーバへの接続を別のサーバにリダイレクトする前に、キューに入れることが可能な接続数を指定する。任意の整数値に設定できる。デフォルトは 5。</p> <ul style="list-style-type: none"> • 値 - 整数 • デフォルト - 5

オプション	説明
TEMP_DATA_IN_SHARED_TEMP	<p>ON の場合、共有テンポラリストアに少なくとも 1 つの読み書きファイルが追加されているかぎり、すべてのテンポラリテーブルデータと対象スクラッチデータが共有テンポラリストアに書き込まれる。このオプションを設定した後、または、読み書きファイルを共有テンポラリストアに追加した後に、すべてのマルチプレックスノードを再起動する必要がある (共有テンポラリストアに読み書きファイルが含まれていない場合、または、ノードを再起動しなかった場合は、代わりに、データが IQ_SYSTEM_TEMP に書き込まれる)。OFF (デフォルト) の場合は、すべてのテンポラリテーブルデータとスクラッチデータがローカルテンポラリストアに書き込まれる。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - OFF

ALTER TABLE 文

テーブルの定義を変更します。

クイックリンク：

「パラメータ」 (462 ページ)

「例」 (470 ページ)

「使用法」 (473 ページ)

「標準」 (473 ページ)

「パーミッション」 (474 ページ)

構文

構文 1 - Alter Owner

```
ALTER TABLE table_name ALTER OWNER TO new_owner
  [ { PRESERVE | DROP } PERMISSIONS ]
  [ { PRESERVE | DROP } FOREIGN KEYS ]
```

構文 2

```
ALTER TABLE [ owner. ] table-name
  | { ENABLE | DISABLE } RLV STORE
  { alter-clause, ... }
```

alter-clause - (back to Syntax 2)

```
  ADD create-clause
    | ALTER column-name column-alteration
    | ALTER [ CONSTRAINT constraint-name ] CHECK ( condition )
    | DROP drop-object
```

```

| RENAME rename-object
| move-clause
| SPLIT PARTITION range-partition-name
  INTO ( range-partition-decl, range-partition-decl )
| MERGE PARTITION partition-name-1 INTO partition-name-2
| UNPARTITION
| PARTITION BY
      range-partitioning-scheme
      | hash-partitioning-scheme
      | composite-partitioning-schemecomposite-partitioning-scheme

create-clause - (back to alter-clause)
  column-name column-definition [ column-constraint ]
  | table-constraint
  | [ PARTITION BY ] range-partitioning-scheme

column definition - (back to create-clause)
  column-name data-type [ NOT NULL | NULL ]
  [ IN dbspace-name ]
  [ DEFAULT default-value | IDENTITY ]

column-constraint - (back to create-clause)
  [ CONSTRAINT constraint-name ]
  { UNIQUE
  | PRIMARY KEY
  | REFERENCES table-name [ ( column-name ) ] [ actions ]
  | CHECK ( condition )
  | IQ UNIQUE ( integer )
  }

table-constraint - (back to create-clause)
  [ CONSTRAINT constraint-name ]
  { UNIQUE ( column-name [ , ... ] )
  | PRIMARY KEY ( column-name [ , ... ] )
  | foreign-key-constraint
  | CHECK ( condition )
  }

foreign-key-constraint - (back to table-constraint)
  FOREIGN KEY [ role-name ] [ ( column-name [ , ... ] ) ]
  ... REFERENCES table-name [ ( column-name [ , ... ] ) ]
  ... [ actions ]

actions - (back to foreign-key-constraint)
  [ ON { UPDATE | DELETE } { RESTRICT } ]

column-alteration - (back to alter-clause)
  { column-data-type | alterable-column-attribute } [ alterable-column-attribute ... ]

  | ADD [ constraint-name ] CHECK ( condition )
  | DROP { DEFAULT | CHECK | CONSTRAINT constraint-name }

alterable-column-attribute - (back to column-alteration)
  [ NOT ] NULL
  | DEFAULT default-value
  | [ CONSTRAINT constraint-name ] CHECK { NULL | ( condition )

```

```

    }

default-value - (back to alterable-column-attribute)
    CURRENT { DATABASE | DATE | REMOTE USER | TIME | TIMESTAMP | USER |
    PUBLISHER }
    | string
    | global variable
    | [ - ] number
    | ( constant-expression )
    | built-in-function ( constant-expression )
    | AUTOINCREMENT
    | NULL
    | TIMESTAMP
    | LAST USER
    | USER

drop-object - (back to alter-clause)
    { column-name
    | CHECK constraint-name
    | CONSTRAINT
    | UNIQUE ( index-columns-list )
    | PRIMARY KEY
    | FOREIGN KEY fkey-name
    | [ PARTITION ] range-partition-name
    }

rename-object - (back to alter-clause)
    new-table-name
    | column-name TO new-column-name
    | CONSTRAINT constraint-name TO new-constraint-name
    | [ PARTITION ] range-partition-name TO new-range-partition-name

move-clause - (back to alter-clause)
    { ALTER column-name
    MOVE
    { PARTITION ( range-partition-name TO new-dbspace-name)
    | TO new-dbspace-name }
    }
    | MOVE PARTITION range-partition-name TO new-dbspace-name
    | MOVE TO new-dbspace-name
    | MOVE TABLE METADATA TO new-dbspace-name
    }

range-partitioning-scheme - (back to alter-clause)
    RANGE( partition-key )
    ( range-partition-decl [, range-partition-decl ... ] )

partition-key - (back to range-partitioning-scheme)
    column-name

range-partition-decl - (back to alter-clause) or (back to range-
partitioning-scheme)
    range-partition-name VALUES <= ( { constant | MAX } ) [ IN dbspace-
name ]

```

```

hash-partitioning-scheme - (back to alter-clause) or (back to composite-
partitioning-scheme)
    HASH ( partition-key, ... ] )

composite-partitioning-scheme - (back to alter-clause)
    hash-partitioning-scheme SUBPARTITION range-partitioning-scheme

```

パラメータ

(先頭に戻る) (459 ページ)

- **{ ENABLE | DISABLE } RLV STORE** - リアルタイムのインメモリ更新用に、このテーブルを RLV ストアに登録します。IQ テンポラリテーブルまたはマルチプレックス環境ではサポートされていません。この値は、データベースオプション **BASE_TABLES_IN_RLV** の値よりも優先されます。
- **ADD column-definition [column-constraint]** - テーブルに新しいカラムを追加します。

NOT NULL を指定するには、テーブルが空であることが必要です。IDENTITY カラムまたは DEFAULT AUTOINCREMENT 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。カラムにデフォルトの IDENTITY 値が指定されていれば、新しいカラムのすべてのローに連続する値が入力されます。また、1つのカラムキーに、FOREIGN 制約をカラム制約として追加できます。IDENTITY/DEFAULT AUTOINCREMENT カラムの値は、テーブル内の各ローをユニークに識別します。

IDENTITY/DEFAULT AUTOINCREMENT カラムには、挿入や更新の際に自動的に生成される連続した数値が格納されます。DEFAULT AUTOINCREMENT カラムは、IDENTITY カラムとも呼ばれます。IDENTITY/DEFAULT AUTOINCREMENT を使用するカラムは、整数データ型のいずれか、または位取りが 0 の真数値型であることが必要です。カラム制約と IDENTITY/DEFAULT AUTOINCREMENT カラムの詳細については、「CREATE TABLE 文」を参照してください。

IQ UNIQUE 制約 - カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(*n*) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT、FP_NBIT_LOOKUP_MB、および FP_NBIT_ROLLOVER_MAX_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

IQ UNIQUE を FP_NBIT_AUTOSIZE_LIMIT 未満の *n* 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP として

ロードする場合や、重複しない値の数が FP_NBIT_AUTOSIZE_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

注意：

- 高い IQ UNIQUE 値を指定する際はメモリ使用率を考慮します。マシンリソースに制限がある場合、FP_NBIT_ENFORCE_LIMITS='OFF' (デフォルト) でロードしないでください。
SAP Sybase IQ 16.0 以前は、16777216 を超える IQ UNIQUE *n* 値はフラット FP にロールオーバーされていました。16.0 では、より大きい IQ UNIQUE 値がトークン化でサポートされていますが、カーディナリティとカラム幅に応じて、重大なメモリリソース要件が発生する場合があります。
- BIT、BLOB、および CLOB の各データ型は NBit ディクショナリ圧縮をサポートしません。FP_NBIT_IQ15_COMPATIBILITY='OFF' である場合、これらのデータ型を含む CREATE TABLE 文または ALTER TABLE 文でゼロ以外の IQ UNIQUE カラムを指定すると、エラーが返されます。

-
- **ALTER *column-name* *column-alteration*** – カラム定義を次のように変更します。
 - **SET DEFAULT *default-value*** - テーブルの既存のカラムのデフォルト値を変更します。この作業では MODIFY 句も使用できますが、ALTER は ISO/ANSI SQL に準拠しているのに対して MODIFY は準拠していません。デフォルト値を変更しても、テーブルの既存の値は変更されません。
 - **DROP DEFAULT** - テーブルの既存のカラムのデフォルト値を削除します。この作業では MODIFY 句も使用できますが、ALTER は ISO/ANSI SQL に準拠しているのに対して MODIFY は準拠していません。デフォルトを削除しても、テーブルの既存の値は変更されません。
 - **ADD** - 名前付き制約または CHECK 条件をカラムに追加します。新しい制約または条件は、それを定義した後でテーブルに対して実行される処理のみに適用されます。テーブルの既存の値は、新しい制約や条件を満たすかどうかの検証を受けません。
 - **CONSTRAINT *column-constraint-name*** - オプションのカラム制約名を指定すると、後で、カラム制約全体を修正するのではなく、制約を個別に修正または削除できます。
 - **[CONSTRAINT *constraint-name*] CHECK (*condition*)** - この句はカラムに検査制約を追加するときに使用します。
 - **SET COMPUTE (*expression*)** - 計算カラムに関連付けられた式を変更します。この文を実行すると、カラムの値が再計算されます。新しい式が無効な場合、この文は失敗します。
 - **DROP COMPUTE** - 計算カラムから非計算カラムに変更します。この文はテーブル内の既存の値を変更しません。

- **ADD table-constraint** – テーブルに制約を追加します。

シングルカラムまたはマルチカラムのキーに、外部キー制約をテーブル制約として追加することもできます。PRIMARY KEY を指定する場合、テーブルには CREATE TABLE 文または別の ALTER TABLE 文で作成したプライマリキーがあってはなりません。テーブル制約の詳細については、「CREATE TABLE 文」を参照してください。

注意： テーブルまたはカラムの制約は MODIFY (変更) できません。制約を変更するには、古い制約を DELETE (削除) し、新しい制約を ADD (追加) します。

- **DROP drop-object** – テーブルオブジェクトを削除します。
 - **DROP column-name** - カラムをテーブルから削除します。カラムがマルチカラムインデックス、一意性制約、外部キー、またはプライマリキーに含まれている場合は、インデックス、制約またはキーを削除してからカラムを削除してください。このようにするとカラムを参照する検査制約は削除されません。IDENTITY/DEFAULT AUTOINCREMENT カラムを削除できるのは、テーブルの IDENTITY_INSERT が OFF に設定され、かつテーブルがローカルテンポラリテーブルでない場合だけです。
 - **DROP CHECK** - テーブルのすべての検査制約を削除します。テーブル検査制約とカラム検査制約の両方が対象となります。
 - **DROP CONSTRAINT constraint-name** - テーブルまたは指定したカラムの名前付き制約を削除します。
 - **DROP UNIQUE (column-name, ...)** - 指定したカラムの一意性制約を削除します。一意性制約を参照する外部キー (プライマリキーではなく) も削除します。関連する外部キー制約がある場合は、エラーが報告されます。ALTER TABLE を使用して、プライマリキーを参照するすべての外部キーを削除した後でなければ、プライマリキー制約を削除することはできません。
 - **DROP PRIMARY KEY** - プライマリキーを削除します。このテーブルのプライマリキーを参照するすべての外部キーも削除します。関連する外部キー制約がある場合は、エラーが報告されます。プライマリキーに強制力がない場合、そのプライマリキーに強制力のない外部キー制約が存在すると、DELETE はエラーを返します。
 - **DROP FOREIGN KEY role-name** - 特定のロール名を持つ、該当テーブルの外部キー制約を削除します。その外部キー制約に対して自動的に作成された、ユニークでない HG インデックスは削除されません。HG インデックスは、DROP INDEX 文を使用して明示的に削除してください。
 - **DROP [PARTITION]** - 指定したパーティションを削除します。パーティション P1 内のローとパーティション定義が削除されます。最後のパーティションは削除できません。これは、分割されたテーブルが非分割テーブルに変換されるためです (分割されたテーブルをマージするには、UNPARTITION 句を代わりに使用します)。次に例を示します。

```
CREATE TABLE foo (c1 INT, c2 INT)
PARTITION BY RANGE (c1)
(P1 VALUES <= (100) IN dbbsp1,
 P2 VALUES <= (200) IN dbbsp2,
 P3 VALUES <= (MAX) IN dbbsp3
 ) IN dbbsp4);
LOAD TABLE ...
ALTER TABLE DROP PARTITION P1;
```

- **RENAME *rename-object*** – テーブル内のオブジェクトの名前を変更します。
- **RENAME *new-table-name*** – テーブルの名前を *new-table-name* に変更します。古いテーブル名を使用しているアプリケーションは、修正する必要があります。また、古いテーブル名と同じ名前が自動的に割り当てられた外部キーの名前は、変更しません。
- **RENAME *column-name* TO *new-column-name*** – カラムの名前を *new-column-name* に変更します。古いカラム名を使用しているアプリケーションがある場合は、修正が必要になります。
- **RENAME [PARTITION]** – 既存のパーティション名を変更します。
- **RENAME *constraint-name* TO *new-constraint-name*** – 制約の名前を *new-constraint-name* に変更します。古い制約名を使用しているアプリケーションがある場合は、修正が必要になります。
- **MOVE clause** – テーブルオブジェクトを移動します。各テーブルオブジェクトは1つのDB領域にのみ置くことができます。どのタイプのALTER MOVEも、移動中はテーブルへの変更をすべてブロックします。

注意： オブジェクトをキャッシュDB領域に移動することはできません。

- **MOVE TO** - テーブルが新しいDB領域にマッピングされると、そのテーブルと同じDB領域に存在するカラム、インデックス、一意性制約、プライマリキー、外部キー、メタデータなどのすべてのテーブルオブジェクトを新しいDB領域に移動します。ALTER Column MOVE TO句は、分割テーブルに対しては要求できません。
BITデータ型のカラムはDB領域に明示的に配置することはできません。以下はBITデータ型に対してサポートされていません。
ALTER TABLE t2 alter c1_bit MOVE TO iq_main;
- **MOVE TABLE METADATA** - テーブルのメタデータを新しいDB領域に移動します。分割されたテーブルでは、MOVE TABLE METADATAはパーティション間で共有されるメタデータも移動します。
- **MOVE PARTITION** - 指定したパーティションを新しいDB領域に移動します。
- **PARTITION BY** – 大きなテーブルを、より小さく管理しやすいストレージオブジェクトに分割します。各パーティションは親テーブルと同じ論理属性を共有しますが、別々のDB領域に配置して個別に管理できます。SAP Sybase IQは、次のような複数のテーブル分割スキームをサポートしています。

- ハッシュパーティション
- 範囲パーティション
- 複合パーティション

partition-key は、テーブル分割キーが格納されている 1 つまたは複数のカラムです。分割キーには、NULL 値 および DEFAULT 値を含めることができますが、次のカラムを含めることはできません。

- LOB (BLOB または CLOB) カラム
- BINARY または VARBINARY カラム
- 長さが 255 バイトを超える CHAR または VARCHAR カラム
- BIT カラム
- FLOAT/DOUBLE/REAL カラム
- **PARTITION BY RANGE** – 分割カラム内の値の範囲によってローを分割します。範囲分割は、単一の分割キーカラムおよび最大 1024 パーティションまでに制限されています。range-partitioning-scheme 内の partition-key は、テーブル分割キーが格納されているカラムです。

```
range-partition-decl:
  partition-name VALUES <= ( {constant-expr | MAX } [ ,
  { constant-expr | MAX } ]... )
  [ IN dbspace-name ]
```

partition-name は、テーブルローが格納される新しいパーティションの名前です。パーティション名は、テーブル上にあるパーティションセット内でユニークである必要があります。パーティション名は必須です。

- **VALUE** – 各パーティションの包括的な上限を (昇順に) 指定します。ユーザは、各ローが 1 つのパーティションのみに分配されるように、各範囲分割の分割基準を指定する必要があります。NULL は分割カラムに使用でき、NULL を分割キー値に含んだローは最初のテーブル分割に属します。ただし、NULL をバインド値に指定することはできません。

最初のパーティションには、下限 (MIN 値) は設定されていません。分割キーの最初のカラムにある NULL セルのローは、最初のパーティションに移動します。最後のパーティションでは、包括的な上限または MAX を指定できます。最後のパーティションの上限値が MAX でない場合は、最後のパーティションの上限値よりも大きい分割キーの値を含んだローをロードまたは挿入すると、エラーが生成されます。

- **MAX** – 無制限の上限を示し、最後のパーティションに対してのみ指定できます。
- **IN** – partition-decl でパーティションのローが存在する DB 領域を指定します。

次の制限を設定すると、範囲分割されたテーブルの分割キーとバインド値がその影響を受けます。

- 未分割のテーブルを範囲分割できるのは、すべての既存のローが最初のパーティションに属する場合のみです。
- パーティションバインドは定数式でなく、定数として指定する必要があります。
- パーティションバインドは、パーティションの作成順に応じて、昇順で指定する必要があります。つまり、2 番目のパーティションの上限は最初のパーティションよりも高く指定する必要がある、というようになります。さらに、パーティションバインドの値は、対応する分割キーカラムのデータ型と互換性がなければなりません。たとえば、VARCHAR は CHAR と互換性があります。
- バインド値に対応する分割キーのカラムとは異なるデータ型が指定されていると、SAP Sybase IQ はバインド値を分割キーのカラムのデータ型に変換します。ただし、次の場合は例外となります。
- 明示的な変換は使用できません。この例では、INT から VARCHAR に明示的に変換しようとしてエラーが生成されます。

```
CREATE TABLE Employees(emp_name VARCHAR(20))
PARTITION BY RANGE(emp_name)
(p1 VALUES <=(CAST (1 AS VARCHAR(20))),
p2 VALUES <= (CAST (10 AS VARCHAR(20)))
```

- データロスにつながる暗黙的な変換は使用できません。この例では、パーティションバインドは分割キー型と互換性がありません。丸めを前提で処理を行うとデータロスにつながる可能性があり、エラーが生成されます。
- この例では、パーティションバインドと分割キーのデータ型の間には互換性があります。バインド値は FLOAT 値に直接変換されます。丸め処理は必要なく、変換はサポートされています。

```
CREATE TABLE emp_id (id INT) PARTITION BY RANGE(id) (p1 VALUES
<= (10.5), p2 VALUES <= (100.5))
```

- この例では、パーティションバインドと分割キーのデータ型の間には互換性があります。バインド値は FLOAT 値に直接変換されます。丸め処理は必要なく、変換はサポートされています。
- 非バイナリデータ型からバイナリデータ型に変換することはできません。たとえば、次の変換は実行できず、エラーが返されます。

```
CREATE TABLE newemp (name BINARY)
PARTITION BY RANGE(name)
(p1 VALUES <= ("Maarten"),
p2 VALUES <= ("Zymlerman"))
```

- NULL を範囲分割テーブルで境界として使用することはできません。
- 分割キーの最初のカラムのセル値が NULL と評価された場合、ローは最初のパーティションに挿入されます。SAP Sybase IQ は、1 つのカラムの分割キーのみをサポートしているため、分割キー内に NULL が含まれていると、ローは最初のパーティションに分配されます。
- **PARTITION BY HASH** – 内部ハッシュ関数によって処理された分割キーの値に基づいて、データをパーティションにマップします。ハッシュ分割キーは最大

8 カラムで、組み合わせた宣言カラム幅が 5300 バイト以下に制限されています。ハッシュパーティションの場合、テーブル作成者は分割キーカラムのみを決定します。パーティションの数と位置は内部的に決定されます。

hash-partitioning 宣言内の partition-key は、1 つのカラムまたはカラムのグループです。その複合値によってデータの各ローが格納されるパーティションが決まります。

```
hash-partitioning-scheme:
  HASH ( partition-key [ , partition-key, ... ] )
```

• **制限事項** –

- ハッシュ分割できるのはベーステーブルのみです。グローバルテンポラリテーブルやローカルテンポラリテーブルを分割しようとすると、エラーが発生します。
- 未分割のテーブルをハッシュ分割できるのは、そのテーブルが空の場合のみです。
- ハッシュパーティションの追加、削除、マージ、分割はできません。
- カラムをハッシュ分割キーから追加または削除することはできません。
- **PARTITION BY HASH RANGE** – ハッシュ分割されたテーブルを範囲によってさらに分割します。hash-range-partitioning-scheme 宣言内の SUBPARTITION BY RANGE 句は、新しい範囲サブパーティションを既存のハッシュ範囲分割テーブルに追加します。

```
hash-range-partitioning-scheme:
  PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
    [ SUBPARTITION BY RANGE ( range-partition-decl [ , range-
      partition-decl ... ] ) ]
```

ハッシュパーティションはデータの論理的な配分および配置方法を指定するのに対して、範囲サブパーティションはデータの物理的な配置方法を指定します。新しい範囲サブパーティションは、既存のハッシュ範囲分割テーブルと同じハッシュ分割キーを持つハッシュによって論理的に分割されます。範囲サブパーティションは 1 つのカラムに制限されています。

• **制限事項** –

- ハッシュ分割できるのはベーステーブルのみです。グローバルテンポラリテーブルやローカルテンポラリテーブルを分割しようとすると、エラーが発生します。
- ハッシュ分割されたテーブルを範囲によってさらに分割できるのは、そのテーブルが空の場合のみです。
- ハッシュパーティションの追加、削除、マージ、分割はできません。
- カラムをハッシュ分割キーから追加または削除することはできません。

注意： 範囲パーティションと複合分割スキームは、ハッシュ範囲パーティションと同様、個別にライセンスが必要な VLDB Management オプションを必要とします。

- **MERGE PARTITION** – *partition-name-1* を *partition-name-2* にマージします。2つのパーティションが隣接しており、データが同じ DB 領域にある場合は、パーティションをマージできます。低いパーティションの値を持つパーティションを高いパーティションの値を持つ隣接パーティションにマージする場合にのみ、パーティションをマージできます。サーバでは、パーティションのマージ先の DB 領域に対する CREATE 権限がチェックされないことに注意してください。隣接するパーティションを作成する方法の例については、「CREATE TABLE 文」の例を参照してください。
- **RENAME PARTITION** – 既存の PARTITION の名前を変更します。
- **UNPARTITION** – 分割されたテーブルからパーティションを削除します。各カラムは1つの DB 領域に配置されます。サーバでは、すべてのパーティションのデータの移動先となる DB 領域に対する CREATE 権限がチェックされないことに注意してください。ALTER TABLE UNPARTITION は、データベースのアクティビティをすべてブロックします。
- **ALTER OWNER** – テーブルの所有者を変更します。ALTER OWNER 句を ALTER TABLE 文の他の [alter-clause] 句と組み合わせて使用することはできません。
 - **[PRESERVE | DROP] PERMISSIONS** — 新しい所有者に古い所有者と同じ権限を付与しない場合は、DROP privileges 句 (デフォルト) を指定して、明示的に付与されていたテーブルへのユーザアクセスを許可する権限をすべて取り消すことができます。そのテーブルの所有者の権限が暗黙的に与えられたものである場合、その権限が新しい所有者に付与され、前の所有者から削除されます。
 - **[PRESERVE | DROP] FOREIGN KEYS** - 新しい所有者が参照先テーブル内のデータにアクセスできないようにするには、DROP FOREIGN KEYS 句 (デフォルト) を使用して、テーブル内のすべての外部キーと、そのテーブルを参照しているすべての外部キーを削除します。新しい所有者がすべての参照元テーブルを所有していないかぎり、PRESERVE FOREIGN KEYS 句と DROP PERMISSIONS 句を組み合わせて使用すると失敗します。

ALTER TABLE ALTER OWNER 文は次の場合に失敗します。

- 元のテーブルと同じ名前を持つ別のテーブルが存在し、新しいユーザがそのテーブルを所有している場合。
- PRESERVE FOREIGN KEYS 句と PRESERVE PERMISSIONS 句の両方が指定され、新しいテーブル所有者以外のユーザが所有する外部キーが存在し、その外部キーが暗黙的に付与された権限 (テーブルの所有者に与えられた権限など) に依存するテーブルを参照している場合。この失敗を回避するに

は、参照元テーブルの元の所有者に対して SELECT 権限を明示的に付与するか、該当する外部キーを削除します。

- PRESERVE FOREIGN KEYS が指定され、PRESERVE PERMISSIONS が指定されていないときに、新しいテーブル所有者以外のユーザが所有する外部キーが存在し、その外部キーがテーブルを参照している場合。この失敗を回避するには、該当する外部キーを削除します。
- PRESERVE FOREIGN KEYS 句が指定され、暗黙的に付与された権限 (テーブルの所有者に与えられた権限など) に依存する外部キーがテーブルに含まれている場合。この失敗を回避するには、新しい所有者に対して参照先テーブルの SELECT 権限を明示的に付与するか、該当する外部キーを削除します。
- テーブル内にシーケンスを参照するデフォルト値を持つカラムがあり、シーケンスジェネレータの USAGE 権限が暗黙的に付与された権限 (シーケンスの所有者に与えられた権限など) に依存する場合。この失敗を回避するには、テーブルの新しい所有者に対してシーケンスジェネレータの USAGE 権限を明示的に付与します。
- 元のテーブルに依存する有効化されたマテリアライズドビューが存在する場合。

例

(先頭に戻る) (459 ページ)

- **例 1** – 従業員の勤務先の事務所を示す新しいカラムを、Employees テーブルに追加します。

```
ALTER TABLE Employees
ADD office CHAR(20)
```

- **例 2** – office カラムを Employees テーブルから削除します。

```
ALTER TABLE Employees
DROP office
```

- **例 3** – Customers テーブルにカラムを追加して、各顧客の販売担当を割り当てます。

```
ALTER TABLE Customers
ADD SalesContact INTEGER
REFERENCES Employees (EmployeeID)
```

- **例 4** – 新しいカラム CustomerNum を Customers テーブルに追加して、デフォルト値の 88 を割り当てます。

```
ALTER TABLE Customers
ADD CustomerNum INTEGER DEFAULT 88
```

- **例 5** – c2、c4、および c5 の **FP** インデックスを DB 領域 Dsp3 から Dsp6 に移動します。c1 の **FP** インデックスは Dsp1 に残ります。c3 の **FP** インデックス

は Dsp2 に残ります。c5 のプライマリーキーは Dsp4 に残ります。DATE インデックス c4_date は Dsp5 に残ります。

```
CREATE TABLE foo (
    c1 INT IN Dsp1,
    c2 VARCHAR(20),
    c3 CLOB IN Dsp2,
    c4 DATE,
    c5 BIGINT,
    PRIMARY KEY (c5) IN Dsp4) IN Dsp3);

CREATE DATE INDEX c4_date ON foo(c4) IN Dsp5;
ALTER TABLE foo
    MOVE TO Dsp6;
```

- **例 6** – FP インデックス c1 のみを DB 領域 Dsp1 から Dsp7 に移動します。

```
ALTER TABLE foo ALTER c1 MOVE TO Dsp7
```

- **例 7** – 多数の **ALTER TABLE** 句を使用して、パーティションの移動、分割、名前変更、およびマージを行います。

分割されたテーブルを作成します。

```
CREATE TABLE bar (
    c1 INT,
    c2 DATE,
    c3 VARCHAR(10))
    PARTITION BY RANGE(c2)
    (p1 VALUES <= ('2005-12-31') IN dbsp1,
    p2 VALUES <= ('2006-12-31') IN dbsp2,
    p3 VALUES <= ('2007-12-31') IN dbsp3,
    p4 VALUES <= ('2008-12-31') IN dbsp4);
INSERT INTO bar VALUES(3, '2007-01-01', 'banana nut');
INSERT INTO BAR VALUES(4, '2007-09-09', 'grape jam');
INSERT INTO BAR VALUES(5, '2008-05-05', 'apple cake');
```

パーティション p2 を dbsp5 に移動します。

```
ALTER TABLE bar MOVE PARTITION p2 TO DBSP5;
```

パーティション p4 を 2 つのパーティションに分割します。

```
ALTER TABLE bar SPLIT PARTITION p4 INTO
    (p41 VALUES <= ('2008-06-30') IN dbsp4,
    p42 VALUES <= ('2008-12-31') IN dbsp4);
```

次の **SPLIT PARTITION** では、データを移動する必要があるため、エラーが報告されます。既存のローが分割後にすべて同じパーティションにあるとはかぎりません。

```
ALTER TABLE bar SPLIT PARTITION p3 INTO
    (p31 VALUES <= ('2007-06-30') IN dbsp3,
    p32 VALUES <= ('2007-12-31') IN dbsp3);
```

次のエラーが報告されます。

```
No data move is allowed, cannot split partition p3.
```

次の **SPLIT PARTITION** では、パーティションの境界値が変更されるため、エラーが報告されます。

```
ALTER TABLE bar SPLIT PARTITION p2 INTO  
  (p21 VALUES <= ('2006-06-30') IN dbsp2,  
   P22 VALUES <= ('2006-12-01') IN dbsp2);
```

次のエラーが報告されます。

```
Boundary value for the partition p2 cannot be changed.
```

パーティション p3 を p2 にマージします。高い境界値から低い境界値のパーティションへのマージは使用できないため、エラーが報告されます。

```
ALTER TABLE bar MERGE PARTITION p3 into p2;
```

次のエラーが報告されます。

```
Partition 'p2' is not adjacent to or before partition 'p3'.
```

パーティション p2 を p3 にマージします。

```
ALTER TABLE bar MERGE PARTITION p2 INTO P3;
```

パーティション p1 の名前を p1_new に変更します。

```
ALTER TABLE bar RENAME PARTITION p1 TO p1_new;
```

テーブル bar の分割を解除します。

```
ALTER TABLE bar UNPARTITION;
```

テーブル bar を分割します。このコマンドでは、すべてのローが最初のパーティションに含まれている必要があるため、エラーが報告されます。

```
ALTER TABLE bar PARTITION BY RANGE (c2)  
  (p1 VALUES <= ('2005-12-31') IN dbsp1,  
   P2 VALUES <= ('2006-12-31') IN DBSP2,  
   P3 VALUES <= ('2007-12-31') IN dbsp3,  
   P4 VALUES <= ('2008-12-31') IN dbsp4);
```

次のエラーが報告されます。

```
All rows must be in the first partition.
```

テーブル bar を分割します。

```
ALTER TABLE bar PARTITION BY RANGE (c2)  
  (p1 VALUES <= ('2008-12-31') IN dbsp1,  
   P2 VALUES <= ('2009-12-31') IN dbsp2,  
   P3 VALUES <= ('2010-12-31') IN dbsp3,  
   P4 VALUES <= ('2011-12-31') IN dbsp4);
```

- **例 8** – テーブル tab1 を変更して、インメモリリアルタイム更新の対象として RLV ストアに登録されないようにします。

```
ALTER TABLE tabl DISABLE RLV STORE
```

使用法

(先頭に戻る) (459 ページ)

ALTER TABLE 文は、以前作成したテーブルのテーブル属性(カラム定義、制約)を変更します。構文では複数の ALTER 句を使用できますが、1つの ALTER TABLE 文の中では1つの table-constraint または column-constraint しか追加、修正、削除できないことに注意してください。ALTER TABLE 文は、他の接続で現在使用中のテーブルに影響を及ぼす場合は実行できません。ALTER TABLE は処理に時間がかかり、この文の処理中は、同じテーブルを参照する要求がサーバで処理されません。

注意： ローカルテナポラリテーブルは変更できませんが、グローバルテナポラリテーブルは、テーブルを使用する接続が1つだけの場合には変更できます。

SAP Sybase IQ は、REFERENCES 制約と CHECK 制約を適用します。ALTER TABLE 文で追加されるテーブルまたはカラムの検査制約は、追加される新しいカラムのいずれかに対して定義された場合にのみ、ALTER TABLE 操作の中で評価されます。CHECK 制約の詳細については、「CREATE TABLE 文」を参照してください。

ビュー定義に **SELECT *** を使用し、その **SELECT *** で参照されるテーブルを変更する場合は、**ALTER VIEW <viewname> RECOMPILE** を実行してビュー定義を訂正し、ビューをクエリしたときに予期しない結果が返されるのを防ぐ必要があります。

関連する動作:

- オートコミット。ALTER と DROP オプションは現在の接続に対するすべてのカーソルをクローズします。Interactive SQL のデータウィンドウもクリアされます。
- ALTER TABLE 操作の開示時にチェックポイントを実行します。
- カラムまたはテーブルを変更すると、その変更したカラムを参照するストアドプロシージャ、ビューなどは機能しなくなります。

標準

(先頭に戻る) (459 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - 一部の句は SAP Adaptive Server® Enterprise でサポートされています。

パーミッション

(先頭に戻る) (459 ページ)

構文 1

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こと

構文 2

構文 1 に必要なシステム権限は、使用する句によって異なります。

句	必要な権限
ADD	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> • ALTER ANY TABLE システム権限 • ALTER ANY OBJECT システム権限 • 基礎となるテーブルに対する ALTER 権限 • 基本となるテーブルを所有していること <p>UNIQUE、PRIMARY KEY、FOREIGN KEY、または IQ UNIQUE カラム制約には、上記とともに基本となるテーブルの REFERENCES 権限が必要。</p> <p>FOREIGN KEY テーブル制約には、上記とともに次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE ANY INDEX システム権限 • CREATE ANY OBJECT システム権限 • ベーステーブルに対する REFERENCES 権限 <p>PARTITION BY RANGE には、上記とともに次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE ANY OBJECT システム権限 • パーティションが作成される DB 領域に対する CREATE 権限

句	必要な権限
ALTER	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> • 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こと <p>プライマリキーまたは一意性制約を変更するには、そのテーブルに対する REFERENCES 権限も必要。</p>
DROP	<p>制約を持たないカラムを削除するには、次のいずれかが必要。</p> <ul style="list-style-type: none"> • ALTER ANY OBJECT システム権限 • ALTER ANY TABLE システム権限 • 基礎となるテーブルに対する ALTER 権限 • 基本となるテーブルを所有していること <p>ALTER 権限を使用する場合、制約を持つカラムまたはテーブルを削除するには、上記とともに REFERENCES 権限が必要。</p> <p>自分が所有するテーブルのパーティションを削除する場合は、何も必要ない。</p> <p>他のユーザが所有するテーブルのパーティションを削除する場合は、次のいずれかが必要。</p> <ul style="list-style-type: none"> • 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
RENAME	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> • 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こと

句	必要な権限
MOVE	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> • ALTER ANY TABLE システム権限 • ALTER ANY OBJECT システム権限 • システム権限 • 基礎となるテーブルに対する ALTER 権限 • 基本となるテーブルを所有していること <p>次のいずれかも必要。</p> <ul style="list-style-type: none"> • CREATE ANY OBJECT システム権限 • パーティションの移動先となる DB 領域に対する CREATE 権限
SPLIT PARTITION	<p>自分が所有するテーブルのパーティションの場合は、何も必要ない。 他のユーザが所有するテーブルのパーティションの場合は、次のいずれかが必要。</p> <ul style="list-style-type: none"> • SELECT ANY TABLE システム権限 • テーブルに対する SELECT 権限 <p>次のいずれかも必要。</p> <ul style="list-style-type: none"> • 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
MERGE PARTITION、UNPARTITION	<p>自分が所有するテーブルの場合は、何も必要ない。 他のユーザが所有するテーブルの場合は、次のいずれかが必要。</p> <ul style="list-style-type: none"> • 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
PARTITION BY	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE ANY OBJECT システム権限 • パーティションが作成される DB 領域に対する CREATE 権限 <p>次のいずれかも必要。</p> <ul style="list-style-type: none"> • 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こと

句	必要な権限
/DISABLE RLV STORE	次のいずれかが必要。 <ul style="list-style-type: none"> ALTER ANY TABLE システム権限 ALTER ANY OBJECT システム権限

参照：

- 制限事項 (159 ページ)
- 範囲パーティション (161 ページ)
- ハッシュパーティション (162 ページ)
- ハッシュ範囲パーティション (163 ページ)

CHECKPOINT 文

データベースのチェックポイントを実行します。

クイックリンク：

「使用法」 (477 ページ)

「標準」 (478 ページ)

「パーミッション」 (478 ページ)

構文**CHECKPOINT****使用法**

(先頭に戻る) (477 ページ)

CHECKPOINT は、データベースサーバにチェックポイントの実行を強制します。またチェックポイントは、内部アルゴリズムに従ってデータベースサーバによっても自動的に実行されます。アプリケーションは通常、**CHECKPOINT** を発行する必要がありません。

SAP Sybase IQ のチェックポイントの使用方法は、SQL Anywhere などの OLTP データベースとは異なります。多くの場合、OLTP データベースでは、少数のローにしか影響しない短いトランザクションが実行されます。このため、ページ全体をディスクに書き込むのはコストがかかります。代わりに、OLTP データベースは通常、チェックポイントでディスクに書き込みを行い、変更されたデータローのみを書き込みます。SAP Sybase IQ は OLAP データベースです。1 つの OLAP トランザクションで数千、数百万のデータローを変更することがあります。このため、データベースサーバはチェックポイントの発生を待たずに物理的な書き込みを実行します。トランザクションがコミットするたびに更新されたデータページが

ディスクに書き込まれます。OLAP データベースの場合、任意のチェックポイントで少量のデータを書き込むより、データページ全体を書き込むことで効率が大幅に向上します。

チェックポイント時間の調整や明示的なチェックポイントの発行は、ほとんど必要ありません。SAP Sybase IQ は各トランザクションのコミット後に実際のデータページを書き込むため、SAP Sybase IQ でのチェックポイントの制御は OLTP データベース製品の場合ほど重要ではありません。

標準

(先頭に戻る) (477 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - Adaptive Server でサポートされています。

パーミッション

(先頭に戻る) (477 ページ)

CHECKPOINT システム権限が必要です。

COMMIT 文

データベースへの変更を保存します。またはユーザ定義のトランザクションを終了します。

クイックリンク：

「例」 (478 ページ)

「使用法」 (479 ページ)

「標準」 (480 ページ)

「パーミッション」 (480 ページ)

構文

構文 1 – トランザクションを終了し、すべての変更内容を永続化します。

```
COMMIT [ WORK ]
```

構文 2 – ネストされたトランザクションを作成します

```
COMMIT TRAN[SACTION ] [ transaction-name ]
```

例

(先頭に戻る) (478 ページ)

- **例 1** – 現在のトランザクションをコミットします。

```
COMMIT
```

- **例 2** – 次の Transact-SQL バッチは、@@trancount の連続した値を、0、1、2、1、0 としてレポートします。

```
PRINT @@trancount
BEGIN TRANSACTION
PRINT @@trancount
BEGIN TRANSACTION
PRINT @@trancount
COMMIT TRANSACTION
PRINT @@trancount
COMMIT TRANSACTION
PRINT @@trancount
go
```

使用法

(先頭に戻る) (478 ページ)

- **構文 1** – データ定義文は、自動的にコミットします。詳細については、各 SQL 文の「関連する動作」を参照してください。

データベースサーバが無効な外部キーを検出すると、**COMMIT** 文は失敗します。その場合、無効な外部キーがあることによってトランザクションを終了できなくなります。通常、外部キー整合性はそれぞれのデータ操作オペレーションごとにチェックされます。ただし、データベースオプション **WAIT_FOR_COMMIT** が ON に設定されているか、または特殊な外部キーが **CHECK ON COMMIT** 句によって定義されている場合、データベースサーバは **COMMIT** 文が実行されるまで整合性のチェックを遅延します。

- **構文 2** – ネストされたトランザクションはセーブポイントに似ています。ネストされたトランザクションの一番外側で実行された場合は、この文によってデータベースに対する変更が永続化されます。トランザクション内で実行した場合、**COMMIT TRANSACTION** 文はトランザクションのネストレベルを 1 つ減らします。トランザクションがネストされているときは、最も外側の **COMMIT** だけがデータベースに対する変更を永続化します。

オプションのパラメータ *transaction-name* はこのトランザクションに割り当てられた名前です。有効な識別子にする必要があります。トランザクション名は、ネストされた **BEGIN/COMMIT** 文または **BEGIN/ROLLBACK** 文の最も外側のペアでのみ使用してください。

さまざまなオプションを使用して、**COMMIT** 文の動作を詳細に制御できます。「**COOPERATIVE_COMMIT_TIMEOUT** オプション」、**「COOPERATIVE_COMMITS** オプション」、**「DELAYED_COMMITS** オプション」、および **「DELAYED_COMMIT_TIMEOUT** オプション」を参照してください。

付録：SQL 文とオプションのリファレンス

さい。**Commit** 接続プロパティを使用して、現在の接続のコミット数を取得できます。

関連する動作：

- WITH HOLD によって開かれたカーソルを除き、すべてのカーソルを閉じます。
- ON COMMIT PRESERVE ROWS を使用して宣言されたものを除き、この接続で宣言されたテンポラリテーブルのすべてのローを削除します。

標準

(先頭に戻る) (478 ページ)

- SQL - ISO/ANSI SQL 準拠。
- SAP Sybase Database 製品 - Adaptive Server でサポートされています。構文 2 は、ISO/ANSI SQL 文法の Transact-SQL 拡張です。

パーミッション

(先頭に戻る) (478 ページ)

データベースに接続しておく必要があります。

CREATE DATABASE 文

複数のオペレーティングシステムファイルから構成されるデータベースを作成します。

クイックリンク：

「パラメータ」 (481 ページ)

「例」 (487 ページ)

「使用法」 (488 ページ)

「標準」 (492 ページ)

「パーミッション」 (492 ページ)

構文

```
CREATE DATABASE db-name
... [ [ TRANSACTION ] { LOG ON [ log-file-name ]
      [ MIRROR mirror-file-name ] } ]
... [ CASE { RESPECT | IGNORE } ]
... [ PAGE SIZE catalog-page-size ]
... [ COLLATION collation-label[( collation-tailoring-string ) ] ]
... [ ENCRYPTED { algorithm-key-spec | OFF } ]
... [ BLANK PADDING ON ]
```

```

... [ JCONNECT { ON | OFF } ]
... [ IQ PATH iq-file-name ]
... [ IQ SIZE iq-file-size ]
... [ IQ PAGE SIZE iq-page-size ]
... [ BLOCK SIZE block-size ]
... [ IQ RESERVE sizeMB ]
... [ TEMPORARY RESERVE sizeMB ]
... [ MESSAGE PATH message-file-name ]
... [ TEMPORARY PATH temp-file-name ]
... [ TEMPORARY SIZE temp-db-size ]
... [ DBA USER userid ]
... [ DBA PASSWORD password ]
... [ SYSTEM PROCEDURE AS DEFINER {ON | OFF} ]

```

catalog-page-size (bytes) - (構文に戻る)
{ 4096 | 8192 | 16384 | 32768 }

collation-label - (構文に戻る)
string

collation-tailoring-string - (構文に戻る)
keyword=value

algorithm-key-spec - (構文に戻る)
ON
| [ON] KEY *key* [ALGORITHM AES-*algorithm*]
| [ON] ALGORITHM AES-*algorithm* KEY *key*
| [ON] ALGORITHM 'SIMPLE'

AES-*algorithm* - (back to algorithm-key-spec)
'AES' | 'AES256' | 'AES_FIPS' | 'AES256_FIPS'

key: - (back to algorithm-key-spec)
quoted string

iq-page-size (bytes) - (構文に戻る)
{ 65536 | 131072 | 262144 | 524288 }

block-size (bytes) - (構文に戻る)
{ 4096 | 8192 | 16384 | 32768 }

パラメータ

(先頭に戻る) (480 ページ)

- **TRANSACTION LOG** - データベースサーバがデータベースに対して加えられたすべての変更を記録するファイル。トランザクションログはシステムリカバリで重要な役割を果たします。TRANSACTION LOG 句を指定しない場合、またはファイル名のパスを省略した場合、.db ファイルと同じディレクトリに格納されます。ただし、.db と .iq は別の物理デバイス上に格納する必要があります。ローパーティション上では作成できません。

- **MIRROR** – トランザクションログと同一のコピーで、通常はデータ保護を強化するため別のデバイスで管理されます。デフォルトでは、SAP Sybase IQ はミラー化されたトランザクションログを使用しません。トランザクションログミラーを使用する場合、ファイル名を指定する必要があります。相対パスを使用すると、トランザクションログミラーはカタログストア (db-name.db) のディレクトリを基準にした相対位置に作成されます。ヒント：トランザクションログのミラーコピーは必ず作成してください。
- **CASE** – CASE RESPECT を使用して作成したデータベースでは、比較と文字列操作で、対象となるすべての値の大文字と小文字が区別されます。カラム、プロシージャ、ユーザ ID などのデータベースオブジェクトの名前は対象外です。CASE 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ます。デフォルト (RESPECT) では、すべての比較において大文字と小文字が区別されます。CASE RESPECT を使用すれば、CASE IGNORE よりもパフォーマンスが高くなります。
- **PAGE SIZE** – カタログテーブルを含むデータベースの SQL Anywhere セグメントのページサイズは、4096、8192、16384、または 32768 バイトです。通常、デフォルトの 4096 (4KB) を使用します。データベースが大きい場合は、このデフォルトよりも大きいページサイズが必要になることがあり、その結果、パフォーマンスが高くなる場合があります。それより小さい値ではデータベースがサポートできるカラム数が制限される可能性があります。4096 よりも小さいページサイズを指定しても、SAP Sybase IQ によってページサイズは 4096 になります。
- **COLLATION** – データベース内の文字データ型のソートと比較で使用される照合順。照合によって、使用するエンコード (文字セット) に文字の比較と順序付けに関する情報が提供されます。COLLATION 句を指定していない場合、SAP Sybase IQ はオペレーティングシステムの言語とエンコードに基づいて照合を選択します。ほとんどのオペレーティングシステムでは、デフォルトの照合順である ISO_BINENG を使用して、最適なパフォーマンスを発揮します。ISO_BINENG では、照合順序は ASCII 文字セットの文字順序と同じです。すべての大文字がすべての小文字に先行します (たとえば、'A' も 'B' も 'a' に先行します)。

照合は、サポートされる照合リストから選択できます。SAP Sybase IQ サーバで作成された SQL Anywhere データベースでは、照合に Unicode Collation Algorithm (UCA) を使用することもできます。UCA を指定する場合、ENCODING 句も指定してください。SAP Sybase IQ では SAP Sybase IQ データベースの UCA ベースの照合はサポートされていません。UCA ベースの照合がデータベースの **CREATE DATABASE** 文で指定されている場合は、サーバからエラー「UCA collation is not supported」が返され、データベースを作成できません。照合順はデータベースの作成後は変更できません。

必要に応じて、文字列のソートや比較を詳細に制御することために、照合の調整オプション (*collation-tailoring-string*) を指定することもできます。これらのオプションは、keyword=value ペアの形式で指定し、カッコで囲んで、照合名に続けて指定します。

SAP Sybase IQ で使用する照合の調整オプションには、SAP Sybase IQ データベースの照合の調整オプション (*collation-tailoring-string*) で指定できるキーワード、別の形式、値が含まれています。

表 28 : SAP Sybase IQ で使用する照合の調整オプション

キーワード	照合	別の形式	指定可能な値
CaseSensitivity	サポートされているすべての照合	CaseSensitive、Case	<ul style="list-style-type: none"> • respect – 文字の大文字小文字の違いが考慮される。UCA 照合の場合、UpperFirst の指定と等しくなる。その他の照合では、respect の値は照合自体によって異なる。 • ignore – 文字の大文字小文字の違いは無視される。 • UpperFirst – 常に、大文字を先にソートする (Aa)。 • LowerFirst – 常に、小文字を先にソートする (aA)。

注意： 照合の適合化オプションの中には、SAP Sybase IQ サーバで作成された SQL Anywhere データベース用の UCA 照合を指定する場合にサポートされているものもあります。その他の照合と SAP Sybase IQ の場合は、大文字と小文字の区別の適合化オプションのみがサポートされます。また、照合の適合化オプションで作成したデータベースは、15.0 より前のデータベースサーバでは起動できません。

- **ENCRYPTED** – 物理データベースファイルに格納されたデータを判読不能にします。データベース全体を暗号化するには、CREATE DATABASE ENCRYPTED キーワード (TABLE キーワードなし) を使用します。ENCRYPTED TABLE 句は、SQL Anywhere テーブルのテーブル暗号化のみを有効にする場合に使用します。テーブルレベルの暗号化は、SAP Sybase IQ テーブルではサポートされていません。テーブルの暗号化を有効化すると、以降に ENCRYPTED 句を使用して作成されるテーブルや変更されるテーブルがデータベースの作成時に指定した設定を使用して暗号化されることとなります。

データベース暗号化には、単純と強力という 2 つのレベルがあります。

- 単純暗号化は、難読化と同じです。データは判読できませんが、暗号に関する専門知識を持つユーザはデータを解読できます。単純暗号化の場合

は、**CREATE DATABASE** 句の **ENCRYPTED ON ALGORITHM 'SIMPLE'** または **ENCRYPTED ALGORITHM 'SIMPLE'** を指定するか、アルゴリズムやキーを指定せずに **ENCRYPTED ON** 句を指定します。

- 強力な暗号化は、128 ビットのアルゴリズムとセキュリティキーを使用して処理されます。データは判読不能で、キーがなければ事実上解読できません。強力な暗号化では、128 ビットまたは 256 ビットの AES アルゴリズムを使用する **CREATE DATABASE** 句の **ENCRYPTED ON ALGORITHM** を指定し、**KEY** 句を使用して暗号化キーを指定します。キーの値には、少なくとも 16 文字の長さを持ち、大文字と小文字を含み、数字、文字、特殊文字を使用したものを選ぶことをおすすめします。
この暗号化キーは、データベースを起動するたびに入力する必要があります。

暗号化は、データベースの作成時にのみ指定できます。暗号化を既存のデータベースに導入するには、完全にアンロードした後でデータベースを再作成し、すべてのデータを再ロードする必要があります。アルゴリズムを指定せずに **ENCRYPTED** 句を使用すると、デフォルトの AES が使用されます。デフォルトでは、暗号化は OFF です。

警告！ 暗号化キーは保護する必要があります。キーのコピーを安全な場所に保管してください。キーを失うと、データベースにまったくアクセスできなくなります。そのうえ、修復する方法もありません。

- BLANK PADDING** – 比較の際に後続ブランクは無視されます (**BLANK PADDING ON**)。Embedded SQL プログラムは文字配列にフェッチされた文字列を埋め込みます。このオプションは、ISO/ANSI SQL 標準との互換性を保つために用意されています。**CREATE DATABASE** では、**BLANK PADDING OFF** がサポートされなくなりました。
- JCONNECT** – SAP Sybase jConnect for JDBC ドライバを使用してシステムカタログ情報にアクセスするには、jConnect サポートをインストールします。**JCONNECT** を OFF に設定すると、jConnect システムオブジェクトが除外されます (デフォルトは ON)。その場合でも、システム情報にアクセスしないかぎり、JDBC を使用できます。
- IQ PATH** – SAP Sybase IQ データを含むメインセグメントファイルのパス名。オペレーティングシステムファイル、または I/O デバイスのローパーティションを指定できます (ローパーティションを指定する場合のフォーマットについては、各プラットフォーム用の『インストールおよび設定ガイド』を参照してください)。SAP Sybase IQ は指定したパス名に基づいて型を自動的に検出します。相対パスを使用すると、ファイルはカタログストア (.db ファイル) のディレクトリに相対して作成されます。

IQ PATH 句を省略して、次のいずれかのオプションを指定するとエラーが生成されます。IQ SIZE、IQ PAGE SIZE、BLOCK SIZE、MESSAGE PATH、TEMPORARY PATH、および TEMPORARY SIZE。

- **IQ SIZE** – IQ PATH 句で指定するローパーティションまたはオペレーティングシステムファイルのいずれかのサイズ (MB 単位)。ローパーティションの場合は、IQ SIZE を指定せずに、常にデフォルトを使用してください。これにより、SAP Sybase IQ がローパーティション全体を使用できます。IQ SIZE を指定する場合は、IQ SIZE の値を I/O デバイスのサイズに一致させる必要があります。そうしないと、SAP Sybase IQ からエラーが返されます。オペレーティングシステムファイルの場合は、次の表の最小値から最大値 (4TB) までの間で指定できます。

オペレーティングシステムファイルのデフォルトサイズは IQ PAGE SIZE に応じて異なります。

表 29 : IQ ストアファイルとテンポラリストアファイルのデフォルトと最小サイズ

IQ ページサイズ	IQ SIZE のデフォルト	TEMPORARY SIZE のデフォルト	明示的な IQ SIZE の最小値	明示的な TEMPORARY SIZE の最小値
65536	4096000	2048000	4MB	2MB
131072	8192000	4096000	8MB	4MB
262144	16384000	8192000	16MB	8MB
524288	32768000	16384000	32MB	16MB

- **IQ PAGE SIZE** – (IQ のテーブルとインデックスが格納された) データベースの SAP Sybase IQ セグメントのページサイズ (バイト単位)。値は 65536 から 524288 バイトまでの 2 の累乗です。デフォルトは 131072 (128KB) です。サイズにこれ以外の値を指定すると、その値より大きい次の値に変更されます。IQ ページサイズは、データベースのデフォルト I/O 転送ブロックサイズと最大データ圧縮比を決定します。

最良のパフォーマンスを得るには、次の最小ページサイズを使用します。

- 最も大きいテーブルのロー数が 10 億以下のデータベース、または合計サイズが 8TB 未満のデータベースの場合は、64KB (IQ PAGE SIZE 65536)。これは新規データベースの最小サイズです。32 ビットプラットフォームでは、IQ ページサイズを 64KB にすると最高のパフォーマンスが得られます。
- 最も大きいテーブルに含まれるローの数が 10 億より多く 40 億未満であるか、合計サイズが 8TB 以上になる可能性がある、64 ビットプラットフォーム

ム上のデータベースの場合は、128KB (IQ PAGE SIZE 131072)。128KB はデフォルトの IQ ページサイズです。

- 最も大きいテーブルに含まれるローの数が 40 億を超え、合計サイズが 8TB 以上になる可能性がある、64 ビットプラットフォーム上のデータベースの場合は、256KB (IQ PAGE SIZE 262144)。
- **BLOCK SIZE** – データベースの SAP Sybase IQ セグメントの I/O 転送ブロックサイズ (バイト単位)。IQ PAGE SIZE よりも小さい、4096 から 32768 までの 2 の累乗の値を指定してください。それ以外の値をサイズに指定すると、その値より大きい次のサイズに変更されます。デフォルト値は IQ PAGE SIZE 句の値に応じて異なります。ほとんどのアプリケーションでは、デフォルト値を使用するのが最適です。
- **IQ RESERVE** – 将来の DB ファイルのサイズ拡大に備えてメイン IQ ストア (IQ_SYSTEM_MAIN DB 領域) 用に確保する領域のサイズ (メガバイト単位)。sizeMB パラメータには、0 よりも大きい任意の数値を指定できます。DB 領域の作成後は、この予約サイズを変更できなくなります。IQ RESERVE を指定すると、データベースの内部 (フリーリスト) 構造により多くの領域が使用されるようになります。予約サイズが大きすぎると、内部構造体で必要とする領域が、指定されたサイズよりも大きくなり、エラーになります。
- **TEMPORARY RESERVE** – 将来の DB ファイルのサイズ拡大に備えてテンポラリ IQ ストア (IQ_SYSTEM_TEMP DB 領域) 用に確保する領域のサイズ (メガバイト単位)。sizeMB パラメータには、0 よりも大きい任意の数値を指定できます。DB 領域の作成後は、この予約サイズを変更できなくなります。TEMPORARY RESERVE を指定すると、データベースの内部 (フリーリスト) 構造により多くの領域が使用されるようになります。予約サイズが大きすぎると、内部構造体で必要とする領域が、指定されたサイズよりも大きくなり、エラーになります。

注意： データベースがバックアップからリストアされると、テンポラリ DB 領域用に確保された領域とモードは破棄されます。

- **MESSAGE PATH** – SAP Sybase IQ メッセージトレースファイルを格納するセグメントのパス名。オペレーティングシステムファイルを指定する必要があります。メッセージファイルはローパーティション上には作成できません。相対パスを指定するか、パスを省略した場合、メッセージファイルは .db ファイルのディレクトリに相対して作成されます。
- **TEMPORARY SIZE** – TEMPORARY PATH 句で指定するローパーティションまたはオペレーティングシステムファイルのいずれかのサイズ (メガバイト単位)。ローパーティションの場合、TEMPORARY SIZE を指定せずに常にデフォルトを使用してください。これによって SAP Sybase IQ はローパーティション全体を使用できるようになります。オペレーティングシステムファイルのデフォルトは、常に IQ SIZE 値の 2 分の 1 です。IQ ストアがローパーティション

にあり、テナンタリストアがオペレーティングシステムファイルである場合、デフォルトの TEMPORARY SIZE は IQ スタアのローパーティションの半分のサイズになります。

- **DBA USER – SYS_AUTH_DBA_ROLE** システムロールが付与されたデフォルトのユーザアカウントのユーザ名。この句を指定しない場合、SAP Sybase IQ によってデフォルトの DBA ユーザ ID が作成されます。
- **DBA PASSWORD – SYS_AUTH_DBA_ROLE** システムロールが付与されたデフォルトのユーザアカウントのパスワード。
- **SYSTEM PROCEDURE AS DEFINER** – 権限付きシステムプロシージャが invoker (プロシージャの実行者) と definer (プロシージャの所有者) のどちらの権限で実行されるかを定義します。OFF (デフォルト) の場合、つまり指定しないと、すべての権限付きシステムプロシージャが、invoker の権限で実行されます。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。

ON の場合、16.0 より前の権限付きシステムプロシージャは、definer の権限で実行されます。16.0 以降の権限付きシステムプロシージャは、invoker の権限で実行されます。

例

(先頭に戻る) (480 ページ)

- **例 1** – 次の Windows の例は、mydb という名前の SAP Sybase IQ データベースを、関連する mydb.db、mydb.iq、mydb.iqtmp、mydb.iqmsg の各ファイルと共に C:\¥s1¥data ディレクトリに作成します。

```
CREATE DATABASE 'C:\¥¥s1¥¥data¥¥mydb'
BLANK PADDING ON
IQ PATH 'C:\¥¥s1¥¥data'
IQ SIZE 2000
IQ PAGE SIZE 131072
```

- **例 2** – 次の UNIX コマンドは、IQ PATH と TEMPORARY PATH 用のローデバイスで SAP Sybase IQ データベースを作成します。IQ のデフォルトのページサイズである 128KB が適用されています。

```
CREATE DATABASE '/s1/data/bigdb'
IQ PATH '/dev/md/rdisk/bigdb'
MESSAGE PATH '/s1/data/bigdb.iqmsg'
TEMPORARY PATH '/dev/md/rdisk/bigtmp'
```

- **例 3** – 次の Windows コマンドは、IQ PATH 用のローデバイスで SAP Sybase IQ データベースを作成します。ローデバイス名には円記号を 2 つ続けて指定します (Windows の仕様)。

```
CREATE DATABASE 'company'
IQ PATH 'YYYY.YYE:'
JCONNECT OFF
IQ SIZE 40
```

- **例 4** – 次の UNIX の例は、AES 暗号化アルゴリズムとキー "is!seCret" を使用して、強力な暗号で保護された SAP Sybase IQ データベースを作成します。

```
CREATE DATABASE 'marvin.db'
BLANK PADDING ON
CASE RESPECT
COLLATION 'ISO_BINENG'
IQ PATH '/filesystem/marvin.main1'
IQ SIZE 6400
IQ PAGE SIZE 262144
TEMPORARY PATH '/filesystem/marvin.temp1'
TEMPORARY SIZE 3200
ENCRYPTED ON KEY 'is!seCret' ALGORITHM 'AES'
```

使用法

(先頭に戻る) (480 ページ)

指定された名前と属性でデータベースを作成します。SAP Sybase IQ データベースの作成には IQ PATH 句が必須です。この句が指定されていない場合は、標準の SQL Anywhere データベースが作成されます。

SAP Sybase IQ は、データベース作成時に、データベースを構成するさまざまなタイプのデータを格納するための 4 つのデータベースファイルを自動生成します。各ファイルは DB 領域に対応し、その論理名を使って SAP Sybase IQ はデータベースファイルを識別します。

- *db-name.db* は、カタログ DB 領域 SYSTEM を保持するファイルです。システムテーブルのほか、データベースとユーザが追加するすべての標準 SQL Anywhere データベースオブジェクトについて記述するストアプロシージャが格納されます。*.db* 拡張子を付けずに指定すると、SAP Sybase IQ で自動的に追加されます。この最初の DB 領域には、カタログストアが格納されており、後で、DB 領域を追加してカタログストアのサイズを大きくすることが可能です。ローパーティション上では作成できません。
- *db-name.iq* は、IQ テーブルとインデックスを格納するメインデータ DB 領域 IQ_SYSTEM_MAIN を保持するファイルのデフォルト名です。IQ PATH 句を使用して別のファイル名を指定できます。この最初の DB 領域には、IQ ストアが格納されています。

警告！ IQ_SYSTEM_MAIN は特殊な DB 領域で、IQ db_identity ブロック、IQ チェックポイントログ、コミットされた各トランザクションおよびチェックポイントが設定されたアクティブな各トランザクションの IQ ロールフォワード / ロールバックビットマップ、インクリメンタルバックアップビットマップ、

およびフリーリストルートページといった、データベースを開く際に必要なすべての構造体が格納されます。IQ_SYSTEM_MAIN は、データベースが開いているときは常にオンラインです。

管理者は、特に、テーブルが小さく、重要なテーブルである場合に、IQ_SYSTEM_MAIN でのユーザテーブルの作成を許可できます。ただし、一般的な方法では、データベースの作成後直ちに、管理者は、2つ目のメイン DB 領域を作成して、DB 領域 IQ_SYSTEM_MAIN における CREATE 権限をすべてのユーザから取り消します。その後、新しいメイン DB 領域に対する CREATE 権限を選択したユーザに付与して、PUBLIC.default_dbSPACE を新しいメイン DB 領域に設定します。

- *db-name.iqtmp* は、初期テンポラリ DB 領域 IQ_SYSTEM_TEMP を保持するファイルのデフォルト名です。特定のクエリによって生成されたテンポラリテーブルを格納します。このファイルの必要サイズはクエリのタイプとデータ量に応じて異なります。TEMPORARY PATH 句を使用すると別のファイル名を指定できます。この最初の DB 領域には、テンポラリストアが格納されています。
- *db-name.iqmsg* は、メッセージトレース DB 領域 IQ_SYSTEM_MSG を格納するファイルのデフォルト名です。MESSAGE PATH 句を使用すると別のファイル名を指定できます。

これらのファイルのほかに、データベースにはトランザクションログファイル (*db-name.log*) が存在します。また、トランザクションログミラーファイルが存在する場合があります。

ファイル名 (*db-name*、*log-file-name*、*mirror-file-name*、*iq-file-name*、*message-file-name*、*temp-file-name*) は、オペレーティングシステムファイル名を含む文字列です。リテラル文字列として、一重引用符で囲んでください。

Windows でパスを指定する場合、円記号 (¥) の後に n または x がある場合は円記号を2つ重ねます。こうすることで、SQL の文字列の規則に従って、改行文字 (¥n) または 16 進数字 (¥x) として解釈されるのを回避できます。円記号は常にエスケープした方が安全です。次に例を示します。

```
CREATE DATABASE 'c:¥¥sybase¥¥mydb.db'
LOG ON 'e:¥¥logdrive¥¥mydb.log'
JCONNECT OFF
IQ PATH 'c:¥¥sybase¥¥mydb'
IQ SIZE 40
```

パスを指定しない場合、または相対パスを指定する場合は、以下のことに注意してください。

- カタログストアファイル (*db-name.db*) は、サーバの作業ディレクトリを基準とする相対位置に作成されます。

- IQ ストア、テンポラリストア、メッセージログのファイルは、カタログストアと同じディレクトリ内に作成されるか、カタログストアを基準とする相対位置に作成されます。

相対パス名の使用をおすすめします。

警告！ データベースファイル、テンポラリ DB 領域ファイル、トランザクションログファイルは、データベースサーバと同じ物理マシンに配置する必要があります。データベースファイルとトランザクションログファイルをネットワークドライブに置かないでください。ただし、トランザクションログは、そのミラーファイルとは別のデバイスに置く必要があります。

UNIX 系オペレーティングシステムでは、指定するファイルのパス名を含む間接的なポインタであるシンボリックリンクを作成できます。シンボリックリンクを相対パス名として使用できます。データベースファイル名のシンボリックリンクを作成することには、以下の利点があります。

- 実際のデバイス名は一般に意味を持たないが、ローデバイスへのシンボリックリンクには、意味のある名前を付けることができる。
- シンボリック名を使用すると、バックアップ以降に別のディレクトリへ移動されたデータベースファイルをリストアする際の問題を回避できる。

シンボリックリンクを作成するには、**ln -s** コマンドを使用します。次に例を示します。

```
ln -s /disk1/company/iqdata/company.iq company_iq_store
```

このリンクを作成しておけば、完全修飾パス名を使用する代わりに **CREATE DATABASE** または **RESTORE DATABASE** などのコマンドでシンボリックリンクを指定できます。

データベースや DB 領域を作成する際、パスはすべての DB 領域ファイルでユニークである必要があります。**CREATE DATABASE** コマンドで、これら 2つのストアに対して同じパスとファイル名を指定すると、エラーが返されます。

次のいずれかの方法で、ユニークなパスを作成できます。

- ファイルごとに異なる拡張子を指定する (mydb.iq、mydb.iqtmp など)。
- 異なるファイル名を指定する (mydb.iq および mytmp.iq など)。
- 異なるパス名 (/iqfiles/main/iq、/iqfiles/temp/iq など) または異なるローパーティションを指定する。
- データベースの作成時に、**TEMPORARY PATH** を省略する。この場合、テンポラリストアは、デフォルトの名前と拡張子 `dbname.iqtmp` でカタログストアと同じパスに作成されます。*dbname* は、データベース名です。

警告！ UNIX 系オペレーティングシステムでデータベースの一貫性を保つためには、別のファイルへのリンクであるファイルの名前を指定する必要があります。

SAP Sybase IQ では、リンクされたファイルが指すターゲットを検出できません。コマンド内でファイル名が異なっても、同じオペレーティングシステムファイルポイントを指していないか確認してください。

テーブルに挿入される文字列は、データベースで大文字と小文字が区別されるかどうかに関係なく、常に入力された大文字と小文字がそのまま格納されます。文字列 Value は、文字データ型カラムに挿入される場合、常に大文字の V と残りの文字が小文字でデータベースに格納されます。**SELECT** 文は、Value として文字列を返します。ただし、データベースで大文字と小文字が区別されない場合は、すべての比較で 'value' は 'value'、'VALUE' などと同じとみなされます。SAP Sybase IQ サーバは大文字と小文字を任意に組み合わせた結果を返すので、大文字と小文字を区別しない (CASE IGNORE) データベースで、大文字と小文字を区別する結果は期待できません。

たとえば、次のテーブルとデータがあるとします。

```
CREATE TABLE tb (id int NOT NULL,  
                 string VARCHAR(30) NOT NULL);  
INSERT INTO tb VALUES (1, 'ONE');  
SELECT * FROM tb WHERE string = 'oNe';
```

SELECT の結果は "oNe" (WHERE 句で指定) となることがあり、必ずしも "ONE" (データベースに格納される形) になるとはかぎりません。

同じように、次の結果は

```
SELECT * FROM tb WHERE string = 'One';
```

"One" となることがあり、次の結果は

```
SELECT * FROM tb WHERE string = 'ONE';
```

"ONE" となることがあります。

すべてのデータベースは、次のように 1 つ以上のユーザ ID と、

```
DBA
```

パスワードを指定して作成されます。

```
sql
```

新しいデータベースでは、データベース内の設定に関係なく、すべてのパスワードの大文字と小文字が区別されます。ユーザ ID は CASE RESPECT 設定に影響されません。

データベースの起動時に、ページサイズを現在のサーバのページサイズより大きくすることはできません。サーバのページサイズは、最初に起動されたデータベースによって決まるか、またはサーバのコマンドラインで **-gp** コマンドラインオプションを指定して設定します。

すべての文で、コマンドラインの長さはカタログページサイズに制限されます。デフォルトの 4KB で十分なことがほとんどですが、多数の DB 領域を参照する **RESTORE DATABASE** コマンドなど、非常に長いコマンドに対応できるように、大きな PAGE SIZE 値が必要となるケースもまれにあります。多数のテーブルまたはビューを伴うクエリの実行でも、ページサイズを大きくする必要が生じる場合があります。

デフォルトのカタログページサイズは 4KB なので、ページサイズが 1024 である `utility_db` などのデータベースに接続する場合にのみ問題が発生します。この制限によって、多くの DB 領域を参照する **RESTORE DATABASE** コマンドが失敗する可能性があります。この問題を回避するには、SQL コマンドラインの長さがカタログページサイズ以上にならないようにします。

または、**-gp 32768** を指定してエンジンを開始して、カタログページサイズを増加させます。

関連する動作：

- オートコミット

標準

(先頭に戻る) (480 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 — Adaptive Server は、**CREATE DATABASE** 文を用意していますが、オプションは異なります。

パーミッション

(先頭に戻る) (480 ページ)

この文を実行するのに必要なパーミッションは、以下のように、**-gu** サーバコマンドラインオプションを使用して設定します。

- **NONE** – この文を発行できるユーザはいません。
- **DBA – SERVER OPERATOR** システム権限が必要です。
- **UTILITY_DB** – `utility_db` データベースに接続できるユーザのみがこの文を発行できます。

サーバを実行中のアカウントには、ファイルが作成されたディレクトリの書き込みパーミッションが必要です。

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」 (493 ページ)

「例」 (494 ページ)

「使用法」 (494 ページ)

「標準」 (495 ページ)

「パーミッション」 (495 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) (493 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) (493 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) (493 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 OBJECTT** 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) (493 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) (493 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

CREATE INDEX 文

指定したテーブル、またはテーブルのペアにインデックスを作成します。一度インデックスを作成すると、**DROP INDEX** 文を使用して削除するとき以外は、SQL 文中で再び参照されることはありません。

クイックリンク：

「パラメータ」 (496 ページ)

「例」 (499 ページ)

「使用法」 (500 ページ)

「標準」 (504 ページ)

「パーミッション」 (504 ページ)

構文

```
CREATE [ UNIQUE ] [ index-type ] INDEX [ IF NOT EXISTS ] index-name
...ON [ owner.]table-name
... ( column-name [ , column-name ] ...)
...[ { IN | ON } dbspace-name ]
```

```
...[ NOTIFY integer ]
...[ DELIMITED BY 'separators-string' ]
...[ LIMIT maxwordsize-integer ]
```

index-type

```
{ CMP | HG | HNG | LF | WD | DATE | TIME | DTTM }
```

パラメータ

(先頭に戻る) (495 ページ)

- **index-type** – SAP Sybase IQ テーブルのカラムの場合、HG (High_Group)、HNG (High_Non_Group)、LF (Low_Fast)、WD (Word)、DATE、TIME、または DTTM (Datetime) を index-type に指定できます。index-type を指定しない場合は、HG インデックスがデフォルトで作成されます。

IQ メインストアテーブルの2つのカラム間の関係にインデックスを作成するには、index-type に CMP (Compare) を指定します。2つのカラムのデータ型、精度、位取りは同じであることが必要です。CHAR、VARCHAR、BINARY、または VARBINARY のカラムの場合、精度は2つのカラムの幅が同じであることを意味します。

クエリ速度をできるだけ高速にするために、どのインデックスタイプを選択すべきかは、次の条件に応じて異なります。

- カラム内のユニークな値の数
- クエリ内での当該カラムの使用方法
- 使用可能なディスク領域量

IQ メインストアテーブルの1つのカラムに複数のインデックスを指定できますが、各インデックスは異なるインデックスタイプでなければなりません。

CREATE INDEX を使用して、重複するインデックスタイプを追加することはできません。現在のクエリまたはその一部において利用可能なインデックスの中で最速のものが SAP Sybase IQ によって選択されます。ただし、追加する各インデックスタイプによって、そのテーブルの必要領域量が大幅に増加することがあります。

- **column-name** – インデックスを作成するカラムの名前を指定します。カラム名は、オプションの相関名の後に続く識別子です (通常、相関名はテーブル名です。相関名の詳細については、「FROM 句」を参照してください)。カラム名に英字、数字、アンダースコア以外の文字が使用されている場合は、二重引用符 ("") で囲んでください。

UNIQUE 句を指定すると、指定可能なインデックスは HG と LF だけになります。外部キーにはユニークでないインデックスが必要で、複合外部キーにはユニークでない複合 HG インデックスが必要です。マルチカラムの複合キーの場合、ユニークでもユニークでなくても、HG インデックスの幅の最大値は 5300

バイトです。CHAR または VARCHAR のデータは、複合キーまたはシングルカラムの HG、LF、HNG、DATE、TIME、または DTTM のインデックスの一部である場合は、255 バイト以内にする必要があります。

- **UNIQUE** – インデックスの全カラムで、テーブル内に同じ値を持つローが 2 つ存在しないことを保証します。各インデックスキーはユニークであるか、少なくとも 1 つのカラムで NULL を格納している必要があります。複数のカラムを持つユニークな HG インデックスは作成できますが、それ以外のインデックスタイプを使用してマルチカラムインデックスを作成することはできません。UNIQUE は、CMP、HNG、WD、DATE、TIME、または DTTM のインデックスタイプを使用して指定することはできません。

SAP Sybase IQ では、カラム定義で NULL 値の使用が許可され、制約 (プライマリーキーまたは一意性) も適用されていない場合に、ユーザが作成したユニークなマルチカラム HG インデックスのデータ値で NULL を使用できます。詳細については、「注意事項」の「マルチカラムインデックス」を参照してください。

- **IF NOT EXISTS** – 指定したオブジェクトがすでに存在する場合、変更は行われず、エラーは返されません。
- **IN** – インデックスの配置を指定します。IN 句を省略した場合は、テーブルが作成される DB 領域にインデックスが作成されます。インデックスは常に、そのインデックスのテーブルと同じ型の DB 領域 (IQ ストアまたはテンポラリストア) に配置されます。インデックスをロードすると、そのタイプのデータベースファイルで、利用可能な領域のあるものすべてにデータが分配されます。SAP Sybase IQ は指定された *dbspace-name* がそのインデックスに適合することが保証されます。テンポラリテーブル上のインデックスに対して IQ_SYSTEM_MAIN または他のメイン DB 領域を指定した場合や、その逆の場合には、エラーが発生します。**CREATE DATABASE... CASE IGNORE** または **CASE RESPECT** の指定に関係なく、DB 領域名では常に大文字と小文字が区別されません。
- **DELIMITED BY** – カラムの文字列を複数のワードに解析して、そのカラムの WD インデックスに格納するためのセパレータを指定します。この句が省略されるか、値に空の文字列が指定された場合、SAP Sybase IQ はデフォルトのセパレータのセットを使用します。デフォルトのセパレータのセットは、デフォルトの照合順序 (ISO-BINENG) にあわせて作成されています。7 ビットの ASCII 文字 (7 ビットの ASCII 英数字ではありません) で構成され、ハイフンと一重引用符は含まれません。ハイフンと一重引用符は、デフォルトではワードの一部となります。デフォルトのセパレータのセットには 64 文字が含まれます。たとえば、カラムの値が次の文字列だとします。

```
The cat is on the mat
```

また、データベースが CASE IGNORE の設定でデフォルトのセパレータを使用して作成されたとすると、この文字列から WD インデックスに次のワードが格納されます。

```
cat is mat on the
```

DELIMITED BY 句と LIMIT 句を複数指定した場合は、エラーにはなりません
が、各タイプの最後に指定した句だけが使用されます。

- separators-string** – データベースの作成時に使用した照合順序に含まれる文字を 0 個以上並べて指定する必要があります。separators-string の文字は、それぞれセパレータとして扱われます。separators-string に文字が指定されなければ、デフォルトのセパレータのセットが使用されます (セパレータには使用する照合順序に含まれる単一の文字を指定してください)。separators-string に 256 を超える文字 (セパレータ) を指定することはできません。

タブをデリミタに指定するには、separators-string に <TAB> の文字を入力するか、タブ文字を表す 16 進の ASCII コード (¥x09) を使用します。"¥" と入力すると、¥ と t が指定されます。改行をデリミタに指定するには、<RETURN> の文字か、16 進の ASCII コード (¥x0a) を入力します。

たとえば、DELIMITED BY ' :;.¥/t' 句では、次の 7 つのセパレータが指定されます。空白 : ; . ¥ / t

表 30：タブと改行をデリミタに指定

デリミタ	DELIMITED BY 句のセパレータ文字列
タブ	' ' (<TAB> を入力) または '¥x09'
改行文字	' ' (<RETURN> を入力) または '¥x0a'

- LIMIT** – WD インデックスの作成時にのみ使用できます。WD インデックスで許可される、ワードの最大長を指定します。解析中にこれよりも長いワードが検出されるとエラーが発生します。デフォルトは 255 バイトです。指定可能な値の最小値は 1 で、最大値は 255 です。**CREATE INDEX** 文に指定された、またはデフォルトで定義された最大ワード長がカラム幅を超える場合は、使用された最大ワード長がカラム幅まで自動的に削減されます。最大ワード長を短く設定すると、挿入、削除、更新に必要な領域と時間が節約できます。空のワード (セパレータを 2 つ続けて入力) は自動的に無視されます。WD インデックスを作成すると、そのカラムに挿入されるデータはすべて、作成時に定義したセパレータと最大ワードサイズで解析されるようになります。セパレータや最大ワードサイズを、インデックスが作成された後で変更することはできません。

- **NOTIFY** - n 個のレコードが正常にインデックスに追加された後で、通知メッセージを表示します。メッセージは標準出力デバイスに送信されます。メッセージには、メモリの使用状況、データベース領域、使用中のバッファ数に関する情報が含まれます。デフォルトは 100,000 レコードです。NOTIFY をオフにするには、NOTIFY を 0 に設定します。

例

(先頭に戻る) (495 ページ)

- **例 1** - projected_earnings カラムと current_earnings カラムに比較インデックスを作成します。この 2 つのカラムは、精度と位取りが同じである 10 進のカラムです。

```
CREATE CMP INDEX proj_curr_cmp
ON sales_data
( projected_earnings, current_earnings )
```

- **例 2** - SalesOrderItems テーブルの ID カラムに High_Group インデックスを作成します。このインデックスのデータページは、DB 領域 Dsp5 から割り付けられます。

```
CREATE HG INDEX id_hg
ON SalesOrderItems
( ID ) IN Dsp5
```

- **例 3** - SalesOrderItems テーブルの ProductID カラムに High_Group インデックスを作成します。

```
CREATE HG INDEX item_prod_hg
ON Sales_OrderItems
( ProductID)
```

- **例 4** - 通知メッセージを生成しないで、SalesOrderItems テーブルの同じ ProductID カラムに Low_Fast インデックスを作成します。

```
CREATE LF INDEX item_prod
ON SalesOrderItems
( ProductID)
NOTIFY 0
```

- **例 5** - earnings_report テーブルで **WD** インデックスを作成します。文字列のデリミタに、スペース、コロン、セミコロン、ピリオドを指定しています。文字列の長さは 25 に制限されます。

```
CREATE WD INDEX earnings_wd
ON earnings_report_table(varchar)
DELIMITED BY ' : ; . '
LIMIT 25
```

- **例 6** - SalesOrders テーブルの OrderDate カラムに **DTTM** インデックスを作成します。

```
CREATE DTTM INDEX order_dttm
ON SalesOrders
( OrderDate )
```

使用法

(先頭に戻る) (495 ページ)

- **CREATE INDEX** 文ではインデックスの所有者を指定できません。インデックスが定義されているテーブルの所有者が、自動的にそのインデックスの所有者になります。各所有者に対して、インデックス名はユニークである必要があります。
- ビューに対してインデックスを作成することはできません。各インデックス名は特定のテーブルに対してユニークである必要があります。
- **CREATE INDEX** 文は、他の接続で現在変更中のテーブルに影響を及ぼす場合は処理されません。ただし、インデックスを追加しているテーブルに対して、クエリを実行することは可能です。
- **WD** インデックスを作成すると、そのカラムへ挿入されるデータはすべて、セパレータで解析されます。最大ワードサイズはインデックスの作成後に変更できません。CHAR カラムを使用する場合は、少なくともスペースをセパレータの1つに指定するか、デフォルトのセパレータセットを使用します。SAP Sybase IQ では、最大カラム幅になるまで CHAR カラムに自動的にスペースが埋め込まれます。カラムに文字データのほかにブランクが含まれる場合は、**WD** インデックスが作成されたデータに対するクエリで、誤った結果が返されることがあります。たとえば、CompanyName というカラムにセパレータで区切られた2つの単語が含まれているとします。ただし、2つ目の単語には次のようにブランクが埋め込まれています。

```
'Concord' 'Farms'
```

このとき、ユーザが次のクエリを入力したとします。

```
SELECT COUNT(*) FROM Customers WHERE CompanyName contains ('Farms')
```

パーサはこの文字列を次のように解釈します。

```
'Farms'
```

次の文字列に一致するとは解釈しないため、

```
'Farms'
```

1 ではなく 0 を返します。この問題を防ぐには、CHAR カラムではなく VARCHAR カラムを使用します。

- データ型:
 - **CREATE INDEX** を使用して、BIT データを格納するカラムにインデックスを作成することはできません。

- 255 バイトを超える CHAR および VARCHAR データに対して作成できるのは、デフォルトインデックス、**CMP** インデックス、および **WD** インデックスのみです。
- LONG VARCHAR データに対して作成できるのは、デフォルトのインデックスタイプと **WD** インデックスタイプのみです。
- 255 バイトよりも大きい BINARY と VARBINARY のデータに作成可能なインデックスタイプは、デフォルトインデックス、**CMP** インデックス、および **TEXT** インデックスのみです。
- FLOAT、REAL、または DOUBLE のデータを格納するカラムに **HNG** インデックスおよび **CMP** インデックスを作成することはできません。
- **TIME** インデックスを作成できるのは、TIME データ型のカラムだけです。
- **DATE** インデックスを作成できるのは、DATE データ型のカラムだけです。
- **DTTM** インデックスを作成できるのは、DATETIME データ型または TIMESTAMP データ型のカラムだけです。
- 複数のカラムから、ユニークなまたはユニークでない **HG** インデックスを作成できます。SAP Sybase IQ では、外部キーを構成するカラムセットに対して、ユニークでない **HG** インデックスが自動的に作成されます。

複数のカラムに対応できるインデックスタイプは、**HG** と **CMP** だけです。複数のカラムから、ユニークな **HNG** または **LF** のインデックス、**DATE**、**TIME**、または **DTTM** のインデックスは作成できません。

マルチカラムを連結したキーの最大幅は 5KB (5300 バイト) です。連結可能なカラムの数は、5KB に収まるカラムの数によって決まります。CHAR または VARCHAR のデータは、それがシングルカラムの **HG**、**LF**、**HNG**、**DATE**、**TIME**、または **DTTM** のインデックスの複合キーの一部である場合は、255 バイトを超えることはできません。

マルチカラムインデックスに対する **INSERT** には、インデックスのすべてのカラムを含める必要があります。

ORDER BY 句にカラムが 1 つだけある場合は、マルチカラム **HG** インデックスを使用するとクエリが速く実行されます。次に例を示します。

```
SELECT abs (x) from t1
ORDER BY x
```

上記の例では、**HG** インデックスはソート順に x を縦方向に射影します。

クエリパフォーマンスを高めるには、マルチカラム **HG** インデックスを使用して、次の状況にある **SELECT** または **ORDER BY** 句にある複数のカラム (**ROWID** を含めることも可能) に対して、**ORDER BY** 操作を実行します。

- すべての射影されたカラムとすべての順序カラム (**ROWID** を除く) がインデックス内に存在する。

- 順序キーが先頭の **HG** カラムに順に一致する。

複数のマルチカラム **HG** インデックスが上記の条件を満たしている場合は、個別カウント数が最も低いインデックスが使用されます。

クエリに **ORDER BY** 句が含まれており、**ORDER BY** カラムリストが (**SELECT** リストに参照されるすべてのカラムがマルチカラムインデックス内にある) マルチカラムインデックスのプレフィクスである場合、マルチカラムインデックスは次のように縦方向の射影を使用します。

```
SELECT x, z, y FROM T
ORDER BY x, y
```

SELECT リストのベースカラム上に式が存在し、すべての式で参照された全カラムがマルチカラムインデックスにある場合、クエリは次のようにマルチカラムインデックスを使用します。

```
SELECT power(x,2), x+y, sin(z) FROM T
ORDER BY x, y
```

上記の2つの例以外にも、**ROWID()** 関数が **SELECT** リストの式にある場合は、マルチカラムインデックスが使用されます。次に例を示します。

```
SELECT rowid()+x, z FROM T
ORDER BY x, y, z
```

上記の3つの例以外にも、**ROWID()** が **ORDER BY** リストの終わりにあり、そのリストのカラム (**ROWID()** のリストを除く) が正確な順序でマルチカラムインデックスを使用している場合は、マルチカラムインデックスがクエリに対して使用されます。次に例を示します。

```
SELECT z, y FROM T
ORDER BY x, y, z, ROWID()
```

SAP Sybase IQ では、カラム定義で **NULL** 値の使用が許可され、制約 (プライマリーキーまたは一意性) も適用されていない場合に、ユーザが作成したユニークなマルチカラム **HG** インデックスのデータ値の中で **NULL** を使用できます。この機能のルールを次に示します。

- **NULL** は未定義の値として扱われる。
- ユニークなインデックスカラムで、複数のローに **NULL** 値を設定できる。
 1. シングルカラムインデックスで、インデックスカラムの複数のローに **NULL** 値を設定できる。
 2. マルチカラムインデックスで、1つまたは複数のインデックスカラムの複数のローに **NULL** 値を設定できる。ただし、インデックスのユニーク性が確保できるように他のカラムが非 **NULL** 値の場合に限る。
 3. マルチカラムインデックスで、インデックスに関与しているすべてのカラムで複数のローに **NULL** 値を設定できる。

次にこれらの規則の例を示します。次のようにして作成されたテーブル `table1` があるとします。


```
CREATE TABLE table1
(c1 INT NULL, c2 INT NULL, c3 INT NOT NULL);
```

NULL 値が使用できるカラムに、ユニークなシングルカラムの **HG** インデックスを作成します。

```
CREATE UNIQUE HG INDEX c1_hg1 ON table1 (c1);
```

上記の規則 1 に従って、複数のローのインデックスカラムに NULL 値を挿入します。

```
INSERT INTO table1(c1,c2,c3) VALUES (NULL,1,1);
INSERT INTO table1(c1,c2,c3) VALUES (NULL,2,2);
```

NULL 値が使用できるカラムに、ユニークなマルチカラムの **HG** インデックスを作成します。

```
CREATE UNIQUE HG INDEX c1c2_hg2 ON table1(c1,c2);
```

上記の規則 2 に従って、インデックスのユニークさを確保する必要があります。次の **INSERT** は成功しません。これは、ロー 1 とロー 3 のマルチカラムインデックス `c1c2_hg2` に同じ値が入るためです。

```
INSERT INTO table1(c1,c2,c3) VALUES (NULL,1,3);
```

しかし、次の **INSERT** 操作は、規則 1 と規則 3 に従って成功します。

```
INSERT INTO table1(c1,c2,c3) VALUES (NULL,NULL,3);
INSERT INTO table1(c1,c2,c3) VALUES (NULL,NULL,4);
```

マルチカラムインデックスでのユニークさが保持されています。

次の **UPDATE** 操作は成功します。これは、規則 3 によって、複数のローでマルチカラムインデックスのすべてのカラムに NULL 値を設定できるためです。

```
UPDATE table1 SET c2=NULL WHERE c3=1
```

マルチカラム **HG** インデックスが一意性制約に制御されている場合は、インデックスに関与しているカラムには NULL 値を設定できません。

- **BEGIN PARALLEL IQ ... END PARALLEL IQ** 文を使用して、複数の IQ メインストアテーブルに対する **CREATE INDEX** 文をグループにまとめることで、それらの文を 1 つの DDL 文であるかのように実行できます。詳細については、「**BEGIN PARALLEL IQ ... END PARALLEL IQ** 文」を参照してください。

警告！ コミットされていないデータを含むローカルテンポラリテーブルで **CREATE INDEX** コマンドを使用するとエラーになり、「インデックスを作成するためにローカルテンポラリテーブル <tablename> をコミットする必要があります。」というエラーメッセージが表示されます。ローカルテンポラリテーブルのデータをコミットしてからインデックスを作成してください。

関連する動作：

- オートコミット

標準

(先頭に戻る) (495 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - Adaptive Server には、SAP Sybase IQ よりも複雑な **CREATE INDEX** 文があります。Adaptive Server 構文は SAP Sybase IQ で使用できませんが、一部の句やキーワードは無視されます。Adaptive Server **CREATE INDEX** 文の完全な構文については、『Adaptive Server リファレンスマニュアル 第2巻：コマンド』を参照してください。

Adaptive Server のインデックスは、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のどちらかになります。クラスタードインデックスを使用すると、ほとんどの場合ノンクラスタードインデックスよりも速くデータを検索できます。テーブルごとに1つだけクラスタードインデックスが許可されます。

SAP Sybase IQ は、クラスタードインデックスをサポートしていません。CLUSTERED と NONCLUSTERED のキーワードは SQL Anywhere で許可されますが、SAP Sybase IQ では無視されます。*index-type* が指定されない場合、SAP Sybase IQ は指定のカラムに対して HG インデックスを作成します。

SAP Sybase IQ では、DESC キーワードを指定できません。

インデックス名は、SAP Sybase IQ でも Adaptive Server でも、指定されたテーブル内でユニークである必要があります。

パーミッション

(先頭に戻る) (495 ページ)

インデックスが作成される DB 領域に対する CREATE 権限が必要です。次のいずれかも必要です。

- CREATE ANY INDEX システム権限
- CREATE ANY OBJECT システム権限
- インデックス付きのテーブルに対する REFERENCES 権限
- 基本となるテーブルを所有していること

CREATE LOGICAL SERVER 文

ユーザ定義の論理サーバを作成します。次の文では、論理サーバで共有される物理ノード間で、一貫した共有システムテンポラリストア設定が強制的に適用されます。

クイックリンク：

「パラメータ」 (505 ページ)

「例」 (506 ページ)

「パーミッション」 (507 ページ)

構文

```
CREATE LOGICAL SERVER logical-server-name [
    { Is-create-clause, ... } ] [ WITH STOP SERVER ]

Is-create-clause - (back to Syntax)
    { MEMBERSHIP ( { Is-member, ... } ) | POLICY ls-policy-name }

Is-member - (back to Is-create-clause)
    FOR LOGICAL COORDINATOR | mpx-server-name
```

パラメータ

(先頭に戻る) (504 ページ)

- **logical-server-name** – 次を除く任意のユーザ定義の識別子
 - ALL
 - AUTO
 - COORDINATOR
 - DEFAULT
 - NONE
 - OPEN
 - SERVER
- **MEMBERSHIP** – コーディネータに対する論理的なメンバーシップを定義するには、MEMBERSHIP 句に FOR LOGICAL COORDINATOR を含めます。

論理サーバの作成時にメンバーを指定しないと、論理サーバは空で作成されます。

注意： OPEN 論理サーバや SERVER 論理サーバなどに対する暗黙的な論理サーバメンバーシップ定義が格納されることはありません。

SYS.ISYSLOGICALMEMBER システムテーブルには、論理サーバメンバーシップの定義が格納されています。

ルート論理サーバポリシーの **ALLOW_COORDINATOR_AS_MEMBER** オプションを ON から OFF に変更しても、カタログに格納されているメンバーシップ情報に影響はありません。論理サーバ設定の有効性のみに影響します。

ALLOW_COORDINATOR_AS_MEMBER オプションが OFF に設定されている場合でも、マルチプレックスサーバ名を指定するか、FOR LOGICAL COORDINATOR 句を使用して、現在のコーディネータに対する論理サーバのメンバーシップを定義できます。メンバーシップ定義はカタログに格納されま

すが、マルチプレックスサーバがコーディネータとして動作している間は非アクティブです。

カタログには、論理サーバとそのメンバーシップの定義が格納されています。

- **POLICY** – 論理サーバとユーザ定義の論理サーバポリシーを関連付けます。**POLICY** 句が指定されていない場合、論理サーバはルートポリシーと関連付けられます。

`SYS.ISYSIQLOGICALSERVER` システムテーブルには、対応する論理サーバの論理サーバポリシーについての情報が格納されています。

- **ls-policy-name** – `ROOT` を除く任意のユーザ定義の識別子です。
- **WITH STOP SERVER** – `TEMP_DATA_IN_SHARED_TEMP` オプションが直接的または間接的に変更された場合、論理サーバ内のすべてのサーバを自動的に停止します。

適用対象

マルチプレックスのみ。

例

(先頭に戻る) (504 ページ)

- **例 1** – 3つのマルチプレックスノードを含む、ユーザ定義の論理サーバ *ls1* をメンバーとして作成します。

```
CREATE LOGICAL SERVER ls1 MEMBERSHIP ( n1, n2, n3 )
```

- **例 2** – 3つのメンバーノードを含む、ユーザ定義の論理サーバ *ls1* を作成し、論理サーバポリシー名 *lsp1* を定義します。

```
CREATE LOGICAL SERVER ls1 MEMBERSHIP ( w1_svr, w2_svr, r2_svr )  
POLICY lsp1
```

- **例 3** – 例 2 同様にサーバを作成します。ただし、`TEMP_DATA_IN_SHARED_TEMP` オプションが直接的または間接的に変更された場合、**WITH STOP SERVER** は論理サーバ内の全サーバを自動的に停止します。

```
CREATE LOGICAL SERVER ls1 MEMBERSHIP ( w1_svr, w2_svr, r2_svr )  
POLICY lsp1 WITH STOP SERVER
```

- **例 4** – 論理サーバポリシー *lspolicy1* を指定して、メンバーノードが存在しないユーザ定義の論理サーバ *ls1* を作成します。

```
CREATE LOGICAL SERVER ls1 POLICY lspolicy1
```

- **例 5** – ここで、*n1* は現在のコーディネータです。マルチプレックスノードの名前付きメンバーシップ *n2* および *n3* と、コーディネータの論理メンバーシップ

を指定して論理サーバ *ls2* を作成します。また、*ls2* の論理サーバポリシーを *lspolicy2* に設定します。

```
CREATE LOGICAL SERVER ls2 POLICY
MEMBERSHIP FOR LOGICAL COORDINATOR
lspolicy1, n1, n2, n3 POLICY lspolicy2
```

パーミッション

(先頭に戻る) (504 ページ)

MANAGE MULTIPLEX システム権限が必要です。

CREATE LS POLICY 文

ユーザ定義の論理サーバのポリシーを作成します。次の文では、論理サーバで共有される物理ノード間で、一貫した共有システムテナンタリストア設定が強制的に適用されます。

クイックリンク：

「パラメータ」 (507 ページ)

「例」 (508 ページ)

「使用法」 (508 ページ)

「標準」 (510 ページ)

「パーミッション」 (510 ページ)

構文

```
CREATE LS POLICY ls-policy-name ls-option-value-list [ WITH STOP SERVER ]
```

```
ls-option-value-list - (back to Syntax)
  { ls-option-name = ls-policy-option-value } ...
```

```
ls-option-name - (back to ls-option-value-list)
  ALLOW_COORDINATOR_AS_MEMBER
  | DQP_ENABLED
  | ENABLE_AUTOMATIC_FAILOVER
  | LOGIN_REDIRECTION
  | REDIRECTION_WAITERS_THRESHOLD
  | TEMP_DATA_IN_SHARED_TEMP
```

パラメータ

(先頭に戻る) (507 ページ)

- **ls-policy-name** – 論理サーバポリシーの名前。ポリシー名には root を除く任意の識別子が指定できます。

- **ls-option-value-list** – 論理サーバポリシーオプションの名前。「使用法 (508 ページ)」を参照してください。
- **ls-policy-option-value** – 指定されていないオプションの値はすべて、ルート論理サーバポリシーから継承されます。「使用法 (508 ページ)」を参照してください。
- **WITH STOP SERVER** – TEMP_DATA_IN_SHARED_TEMP オプションが直接的または間接的に変更された場合、論理サーバ内のすべてのサーバを自動的に停止します。

適用対象

マルチプレックスのみ。

例

(先頭に戻る) (507 ページ)

- **例 1** – *lspolicy1* という名前のユーザ定義の論理サーバポリシーを作成します。

```
CREATE LS POLICY lspolicy1
ALLOW_COORDINATOR_AS_MEMBER=ON;
```

使用法

(先頭に戻る) (507 ページ)

より小さな IQ_SYSTEM_TEMP DB 領域が必要な場合は、TEMP_DATA_IN_SHARED_TEMP を ON に設定します。こうすることによって、テンポラリデータが IQ_SYSTEM_TEMP ではなく、IQ_SHARED_TEMP に書き込まれます。ただし、分散クエリ処理環境では、DQP_ENABLED と TEMP_DATA_IN_SHARED_TEMP の両方を ON に設定すると、IQ_SHARED_TEMP 内の追加データで SAN が飽和する可能性があります。これは、IQ_SHARED_TEMP に対する追加の I/O 処理が DQP のパフォーマンスに悪影響を及ぼす可能性があるためです。

オプション	説明
ALLOW_COORDINATOR_AS_MEMBER	<p>ROOT 論理サーバポリシーに対してのみ設定可能。ON (デフォルト) の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーにできる。OFF の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーとして使用できない。</p> <ul style="list-style-type: none"> • 値 – ON、OFF • デフォルト – ON

オプション	説明
DQP_ENABLED	<p>0 に設定すると、クエリ処理が分散されない。1 (デフォルト) に設定した場合、書き込み可能な共有テンポラリファイルが存在するかぎり、クエリ処理が分散される。2 に設定した場合、クエリ処理がネットワーク上で分散され、共有テンポラリストアは使用されない。</p> <ul style="list-style-type: none"> • 値 - 0、1、2 • デフォルト - 1
ENABLE_AUTO-MATIC_FAILOVER	<p>ROOT 論理サーバポリシーに対してのみ設定可能。ON は、指定のログインポリシーが適用される論理サーバの自動フェールオーバーを有効化。OFF (デフォルト) は、論理サーバレベルの自動フェールオーバーを無効にし、手動フェールオーバーを可能にする。デフォルト値の設定に戻すには DEFAULT を指定する。</p> <ul style="list-style-type: none"> • 値 - ON、OFF、DEFAULT • デフォルト - OFF
LOGIN_REDIREC-TION	<p>ON は、指定のログインポリシーが適用される論理サーバのログインリダイレクションを有効化。OFF (デフォルト) は、論理サーバレベルのログインリダイレクションを無効にし、外部接続管理を可能にする。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - OFF
REDIRECTION_WAITERS_THRESHOLD	<p>SAP Sybase IQ がこの論理サーバへの接続を別のサーバにリダイレクトする前に、キューに入れることが可能な接続数を指定する。任意の整数値に設定可能。デフォルトは 5。</p> <ul style="list-style-type: none"> • 値 - 整数 • デフォルト - 5

オプション	説明
TEMP_DATA_IN_SHARED_TEMP	<p>ON の場合、共有テンポラリストアに 1 つ以上の読み書きファイルが追加されていると、すべてのテンポラリテーブルデータと対象スクラッチデータが共有テンポラリストアに書き込まれる。このオプションを設定した後、または、読み書きファイルを共有テンポラリストアに追加した後は、すべてのマルチプレックスノードの再起動が必要 (共有テンポラリストアに読み書きファイルがない場合、または、ノードを再起動しなかった場合は、データが IQ_SYSTEM_TEMP に書き込まれる)。OFF (デフォルト) の場合は、すべてのテンポラリテーブルデータとスクラッチデータがローカルテンポラリストアに書き込まれる。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - OFF

標準

(先頭に戻る) (507 ページ)

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

パーミッション

(先頭に戻る) (507 ページ)

MANAGE MULTIPLEX システム権限が必要です。

LS ポリシーオプション

ルート LS ポリシーとユーザ定義 LS ポリシーで使用可能なオプションを次に示します。

オプション	説明
ALLOW_COORDINATOR_AS_MEMBER	<p>ROOT 論理サーバポリシーに対してしか設定できない。ON (デフォルト) の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーにすることができる。OFF の場合は、コーディネータを任意のユーザ定義論理サーバのメンバーとして使用できない。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - ON

オプション	説明
DQP_ENABLED	<p>0 に設定した場合、クエリ処理が分散されない。1 (デフォルト) に設定した場合、書き込み可能な共有テンポラリファイルが存在するかぎり、クエリ処理は分散される。2 に設定した場合、クエリ処理がネットワーク上で分散され、共有テンポラリストアは使用されない。</p> <ul style="list-style-type: none"> • 値 - 0, 1, 2 • デフォルト - 1
LOGIN_REDIRECTION	<p>ON の場合、指定のログインポリシーが適用される論理サーバのログインリダイレクションが有効になる。OFF (デフォルト) の場合、論理サーバレベルのログインリダイレクションが無効になり、外部接続管理が可能になる。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - OFF
REDIRECTION_WAITERS_THRESHOLD	<p>SAP Sybase IQ がこの論理サーバへの接続を別のサーバにリダイレクトする前に、キューに入れることが可能な接続数を指定する。任意の整数値に設定できる。デフォルトは 5。</p> <ul style="list-style-type: none"> • 値 - 整数 • デフォルト - 5
TEMP_DATA_IN_SHARED_TEMP	<p>ON の場合、共有テンポラリストアに少なくとも 1 つの読み書きファイルが追加されているかぎり、すべてのテンポラリテーブルデータと対象スクラッチデータが共有テンポラリストアに書き込まれる。このオプションを設定した後、または、読み書きファイルを共有テンポラリストアに追加した後、すべてのマルチプレックスノードを再起動する必要がある (共有テンポラリストアに読み書きファイルが含まれていない場合、または、ノードを再起動しなかった場合は、代わりに、データが IQ_SYSTEM_TEMP に書き込まれる)。OFF (デフォルト) の場合は、すべてのテンポラリテーブルデータとスクラッチデータがローカルテンポラリストアに書き込まれる。</p> <ul style="list-style-type: none"> • 値 - ON、OFF • デフォルト - OFF

CREATE TABLE 文

データベースまたはリモートサーバに新しいテーブルを作成します。

クイックリンク：

「パラメータ」 (514 ページ)

「例」 (524 ページ)

「使用法」 (528 ページ)

「標準」 (529 ページ)

「パーミッション」 (530 ページ)

構文

```
CREATE [ { GLOBAL | LOCAL } TEMPORARY ] TABLE
  [ IF NOT EXISTS ] [ owner. ] table-name
  ... ( column-definition [ column-constraint ] ...
  [ , column-definition [ column-constraint ] ... ]
  [ , table-constraint ] ... )
  | { ENABLE | DISABLE } RLV STORE

... [ IN dbspace-name ]
... [ ON COMMIT { DELETE | PRESERVE } ROWS ]
[ AT location-string ]
[ PARTITION BY
  range-partitioning-scheme
  | hash-partitioning-scheme
  | composite-partitioning-scheme ]

column-definition - (構文に戻る)
  column-name data-type
  [ [ NOT ] NULL ]
  [ DEFAULT default-value | IDENTITY ]
  [ PARTITION | SUBPARTITION ( partition-name IN dbspace-name
  [ , ... ] ) ]

default-value - (back to column-definition)
  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
```

column-constraint - (構文に戻る)

```
[ CONSTRAINT constraint-name ] {
  { UNIQUE
    | PRIMARY KEY
    | REFERENCES table-name [ ( column-name ) ] [ action ]
  }
  [ IN dbspace-name ]
  | CHECK ( condition )
  | IQ UNIQUE ( integer )
}
```

table-constraint - (構文に戻る)

```
[ CONSTRAINT constraint-name ]
{ { UNIQUE ( column-name [ , column-name ] ... )
  | PRIMARY KEY ( column-name [ , column-name ] ... )
}
[ IN dbspace-name ]
| foreign-key-constraint
| CHECK ( condition )
| IQ UNIQUE ( integer )
}
```

foreign-key-constraint - (back to table-constraint)

```
FOREIGN KEY [ role-name ] [ ( column-name [ , column-name ] ... ) ]
...REFERENCES table-name [ ( column-name [ , column-name ] ... ) ]
...[ actions ] [ IN dbspace-name ]
```

actions - (back to foreign-key-constraint)

```
[ ON { UPDATE | DELETE } RESTRICT ]
```

location-string - (構文に戻る) or (back to composite-partitioning-scheme)

```
{ remote-server-name. [ db-name ].[ owner ].object-name
  | remote-server-name; [ db-name ]; [ owner ];object-name }
```

range-partitioning-scheme - (構文に戻る)

```
RANGE ( partition-key ) ( range-partition-decl [ , range-partition-decl ... ] )
```

partition-key - (back to range-partitioning-scheme) or (back to hash-partitioning-scheme)

```
column-name
```

range-partition-decl - (back to range-partitioning-scheme)

```
VALUES <= ( { constant-expr
  | MAX } [ , { constant-expr
  | MAX } ] ... )
[ IN dbspace-name ]
```

hash-partitioning-scheme - (構文に戻る) or (back to composite-partitioning-scheme)

```
HASH ( partition-key [ , partition-key, ... ] )
```

composite-partitioning-scheme - (構文に戻る)

```
hash-partitioning-scheme SUBPARTITION range-partitioning-scheme
```

パラメータ

(先頭に戻る) (511 ページ)

- **IN** – column-definition 句、column-constraint 句、table-constraint 句、foreign-key 句、および partition-decl 句で、オブジェクトが作成される DB 領域を指定するために使用します。IN 句を省略した場合、SAP Sybase IQ はテーブルが割り当てられている DB 領域にオブジェクトを作成します。

この句で SYSTEM を指定し、永久テーブルまたはテンポラリテーブルをカタログストアに置くことができます。IQ_SYSTEM_TEMP を指定すると、テンポラリなユーザオブジェクト (テーブル、パーティション、またはテーブルインデックス) を IQ_SYSTEM_TEMP に格納できます。または、

TEMP_DATA_IN_SHARED_TEMP オプションが 'ON' に設定されており、IQ_SHARED_TEMP DB 領域に RW ファイルが含まれている場合は、IQ_SHARED_TEMP に格納できます (IN 句を IQ_SHARED_TEMP とともに指定することはできません)。これ以外の IN 句の使用はすべて無視されます。デフォルトでは、すべての永久テーブルはメイン IQ ストアに、すべてのテンポラリテーブルはテンポラリ IQ ストアに配置されます。グローバルテンポラリテーブルとローカルテンポラリテーブルを IQ ストアに置くことは絶対にできません。

以下の構文はサポートされていません。

```
CREATE LOCAL TEMPORARY TABLE tab1(c1 int) IN IQ_SHARED_TEMP
```

BIT データ型のカラムは DB 領域に明示的に配置することはできません。以下は BIT データ型に対してサポートされていません。

```
CREATE TABLE t1(c1_bit bit IN iq_main);
```

- **ON COMMIT** – テンポラリテーブルに対してのみ使用できます。デフォルトで、テンポラリテーブルのローは COMMIT のときに削除されます。
- **AT** – location-string 句で指定されたりリモートロケーションにマップするプロキシテーブルを作成します。プロキシテーブル名は最大で 30 文字までです。AT 句は、デリミタとしてセミコロン (;) をサポートします。セミコロンが location-string 句のどこかにある場合、そのセミコロンはフィールドデリミタです。セミコロンがない場合は、ピリオドがフィールドデリミタです。これにより、データベースフィールドと所有者フィールドでファイル名と拡張子を使用できます。

セミコロンのフィールドデリミタは、現在サポートされていないサーバクラスで主に使用されていますが、ピリオドもフィールドデリミタとして機能する状況ではセミコロンも使用できます。たとえば、次の文は、テーブル proxy_a をリモートサーバ myasa の SQL Anywhere データベース mydb にマッピングします。

```
CREATE TABLE proxy_al
AT 'myasa;mydb;;al'
```

外部キー定義は、リモートテーブルでは無視されます。リモートテーブルを参照するローカルテーブルの外部キー定義も無視されます。プライマリーキー定義は、サーバがプライマリーキーをサポートする場合、リモートサーバに送信されます。

シンプレックス環境では、同じノード上でリモートテーブルを参照するプロキシテーブルを作成することはできません。マルチプレックス環境では、マルチプレックス内で定義されたリモートテーブルを参照するプロキシテーブルを作成することはできません。

- **IF NOT EXISTS** – 指定したオブジェクトがすでに存在する場合、変更は行われず、エラーは返されません。
- **{ ENABLE | DISABLE } RLV STORE** – このテーブルをインメモリアルタイム更新の対象として RLV ストアに登録します。IQ テンポラリーテーブルはサポートされていません。この値は、データベースオプション **BASE_TABLES_IN_RLV** の値よりも優先されます。この値を ENABLE に設定するには、CREATE TABLE システム権限と、RLV ストアの DB 領域に対する CREATE パーミッションが必要です。
- **column-definition** – テーブルカラムを定義します。使用可能なデータ型については、『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』の「SQL データ型」を参照してください。同じテーブル内の 2 つのカラムが同じ名前を持つことはできません。最大 45,000 のカラムを作成可能ですが、1 つのテーブルに 10,000 を超えるカラムを作成すると、パフォーマンスの低下を招くおそれがあります。
 - **[NOT] NULL** – NULL 値を含めるか、除外するかを設定します。NOT NULL を指定した場合や、カラムが UNIQUE 制約または PRIMARY KEY 制約を受ける場合は、カラムに NULL 値を含めることができません。NULL を許可するカラム数のテーブルごとの制限は、約 $8 * (\text{database-page-size} - 30)$ です。
 - **DEFAULT default-value** – CREATE TABLE (および ALTER TABLE) 文の DEFAULT キーワードでカラムのデフォルト値を指定します。DEFAULT 値は、カラムの値を指定しない INSERT (または LOAD) 文でカラムの値として使用されます。
 - **DEFAULT AUTOINCREMENT** – DEFAULT AUTOINCREMENT カラムの値は、テーブル内の各ローをユニークに識別します。この種のカラムは、Adaptive Server との互換性を考慮して IDENTITY カラムとも呼ばれます。IDENTITY/DEFAULT AUTOINCREMENT カラムには、挿入や更新の際に自動的に生成される連続した数値が格納されます。IDENTITY/DEFAULT AUTOINCREMENT を使用する場合、カラムは整数データ型のいずれか、ま

たは真数型で、位取りを 0 にする必要があります。カラムの値は NULL でもかまいません。テーブル名は、所有者の名前で修飾して指定する必要があります。

ON でテーブルへの挿入を行います。IDENTITY/DEFAULT

AUTOINCREMENT カラムへの値が指定されなければ、カラム内のどの値よりも大きいユニークな値が生成されます。カラムへ格納する値を INSERT に指定すれば、その値が使用されます。指定した値がそのカラムの現在の最大値よりも小さい場合、指定した値は後続の挿入に対して生成する値の起点として使用されます。

ローを削除しても IDENTITY/AUTOINCREMENT カウンタはデクリメントされません。ローの削除によって作成されたギャップは、挿入を行うときに明示的に割り当てることによってのみ埋めることができます。

IDENTITY/AUTOINCREMENT カラムへの挿入を実行するには、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。

以下に、IDENTITY カラムを持つテーブルを作成し、データを明示的に追加する例を示します。

```
CREATE TABLE mytable(c1 INT IDENTITY);
SET TEMPORARY OPTION IDENTITY_INSERT = "DBA".mytable;
INSERT INTO mytable VALUES (5);
```

最大値よりも小さいロー番号を明示的に挿入すると、後続のローで明示的に割り当てなくても、その最大値より 1 大きい値に自動的にインクリメントされます。

カラムに直前に挿入された値は、グローバル変数 @@identity を調べることによって確認できます。

- **IDENTITY – AUTOINCREMENT** デフォルトを使用する代わりに Transact-SQL® 互換の代替手段です。SAP Sybase IQ では、IDENTITY 句または DEFAULT AUTOINCREMENT 句のどちらかを使用して IDENTITY カラムを作成できます。
- **table-constraint** – データベース内のデータの整合性を保証します。整合性制約には次の 4 つのタイプがあります。
 - **UNIQUE** – 1 つまたは複数のカラムによってテーブル内の各ローがユニークに識別されるように指定します。テーブル内の 2 つのローは、指定されたすべてのカラム中に同じ値を持つことはできません。1 つのテーブルに複数の一意性制約が存在することがあります。
 - **PRIMARY KEY** – UNIQUE 制約と同じですが、プライマリキー制約はテーブルに 1 つしか作成できない点が違います。プライマリキー制約と一意性制約を同じカラムに指定することはできません。プライマリキーは通常、

特定のローにとって最適な識別子を識別します。たとえば、顧客番号は顧客テーブルのプライマリキーです。

- **FOREIGN KEY** – 一方のカラムセットに対する値を制限し、他方のテーブルのプライマリキーまたは一意性制約の値と一致させます。たとえば、外部キー制約を使用して、請求書テーブルの顧客番号が顧客テーブルの顧客番号と確実に一致するようにできます。

ローカルテンポラリテーブルに対しては外部キー制約を作成できません。グローバルテンポラリテーブルは ON COMMIT PRESERVE ROWS で作成してください。

- **CHECK** – 任意の条件を検証できます。たとえば、検査制約を使用して Gender カラムに Male と Female の値しか含まれないようにすることができます。テーブル内のどのローも、制約に違反することは許されません。**INSERT** 文または **UPDATE** 文によってローが制約に違反する場合、操作は許可されず、この文の結果は取り消されます。

カラムの検査制約に記述され、先頭に '@' の記号が付くカラム識別子は、実際のカラム名のプレースホルダです。したがって、次のように記述された文は、

```
CREATE TABLE t1(c1 INTEGER CHECK (@foo < 5))
```

次の文とまったく同じになります。

```
CREATE TABLE t1(c1 INTEGER CHECK (c1 < 5))
```

テーブルの検査制約に記述され、先頭に '@' の記号が付くカラム識別子はプレースホルダではありません。

ある文によって整合性制約に違反するデータベースへの変更が生じた場合、その文は事実上実行されず、エラーがレポートされます「事実上」とは、エラーが検出されるより前にこの文が行った変更がすべて取り消されることを示します。

SAP Sybase IQ はそのカラムの HG インデックスを作成することにより、シングルカラム UNIQUE 制約を強制します。

注意： BIT データ型のカラムに、UNIQUE 制約または PRIMARY KEY 制約を定義することはできません。また、BIT データ型のカラムのデフォルトでは NULL 値を使用できませんが、カラムを明示的に定義して、NULL 値を使用できるように変更できます。

- **column-constraint** – カラムに格納可能な値を制限します。カラム制約とテーブル制約によってデータベース内のデータの整合性が保証されます。整合性制約の違反を起こす文は、実行が完了しません。このような文がエラー検出の前に行った変更は取り消され、エラーがレポートされます。カラム制約は、対応するテーブル制約の省略形です。たとえば、次の文は同じです。

```
CREATE TABLE Products (
  product_num integer UNIQUE
)
CREATE TABLE Products (
  product_num integer,
  UNIQUE ( product_num )
)
```

通常、カラム制約を使用するのは、制約がテーブル内で複数のカラムを参照しない場合です。複数のカラムを参照する場合は、テーブル制約を使用する必要があります。

- IQ UNIQUE** – カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(n) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP_NBIT_AUTOSIZE_LIMIT、FP_NBIT_LOOKUP_MB、FP_NBIT_ROLLOVER_MAX_MB の各オプションで定義された上限まで暗黙的に NBit としてロードされます。
 - FP_NBIT_AUTOSIZE_LIMIT は、NBit としてロードする、重複しない値の数を制限します。
 - FP_NBIT_LOOKUP_MB は、NBit ディクショナリの合計サイズのスレッショルドを設定します。
 - FP_NBIT_ROLLOVER_MAX_MB は、NBit からフラット FP への暗黙的な NBit ロールオーバーで使用するディクショナリサイズを設定します。
 - FP_NBIT_ENFORCE_LIMITS は、NBit ディクショナリのサイズ制限を強制します。このオプションはデフォルトで OFF になっています。

IQ UNIQUE を FP_NBIT_AUTOSIZE_LIMIT 未満の n 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が FP_NBIT_AUTOSIZE_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

注意：

- 高い IQ UNIQUE 値を指定する際はメモリ使用率を考慮します。マシンリソースに制限がある場合、FP_NBIT_ENFORCE_LIMITS='OFF' (デフォルト) でロードしないでください。SAP Sybase IQ 16.0 以前は、16777216 を超える IQ UNIQUE n 値はフラット FP にロールオーバーされていました。16.0 では、より大きい IQ UNIQUE 値がトークン化でサポートされていますが、カーディナリティとカラム幅によっては、大量のメモリリソースが必要になる場合があります。

- BIT、BLOB、および CLOB の各データ型は NBit ディクショナリ圧縮をサポートしません。FP_NBIT_IQ15_COMPATIBILITY='OFF' である場合、これらのデータ型を含む CREATE TABLE 文または ALTER TABLE 文でゼロ以外の IQ UNIQUE カラムを指定すると、エラーが返されます。

-
- **column-constraint 句と table-constraint 句** – カラム制約とテーブル制約によってデータベース内のデータの整合性が保証されます。

- **PRIMARY KEY または PRIMARY KEY (column-name, ...)** – テーブルのプライマリキーは、リストしたカラムで構成されます。プライマリキーに指定されたどのカラムにも NULL 値を格納することはできません。SAP Sybase IQ では、テーブル内の各ローが必ず、ユニークなプライマリキー値を持ちます。1 つのテーブルが持てる PRIMARY KEY は 1 つだけです。

2 番目の形式 (PRIMARY KEY の後にカラムリストが続く形式) を使用する場合、カラムのリスト順ではなく定義順にカラムを含めるプライマリキーが作成されます。

カラムを PRIMARY KEY、FOREIGN KEY、または UNIQUE として指定すると、SAP Sybase IQ によってそのカラムに自動的に High_Group インデックスが作成されます。マルチカラムプライマリキーの場合、このインデックスは個々のカラムではなく、プライマリキーに対して作成されます。パフォーマンスを高めるため、各カラムに HG または LF インデックスを個別に作成してください。

- **REFERENCES primary-table-name [(primary-column-name)]** – カラムを、プライマリテーブルのプライマリキーまたは一意性制約に対する外部キーとして定義します。通常、外部キーは一意性制約のためのものではなく、プライマリキーのためのものです。プライマリカラム名を指定する場合、この名前は一意性制約またはプライマリキーの制約を受けるプライマリテーブルのカラム名と一致する必要があります。また、この制約は、その 1 カラムだけで構成される必要があります。それ以外の場合、外部キーは第 2 のテーブルのプライマリキーを参照します。プライマリキーと外部キーは、データ型と精度、位取り、符号が同じであることが必要です。シングルカラムの外部キーには、ユニークでないシングルカラムの HG インデックスだけが作成されます。マルチカラム外部キーに対しては、SAP Sybase IQ はユニークでない複合 HG インデックスを作成します。ユニークまたはユニークでない HG インデックスのマルチカラム複合キーの最大幅は 1KB です。

テンポラリテーブルは、ベーステーブルを参照する外部キーを持つことはできません。また、ベーステーブルも、テンポラリテーブルを参照する外部キーを持つことはできません。ローカルテンポラリテーブルは、外部キーを持つことも、外部キーによって参照されることもできません。

- **FOREIGN KEY [role-name] [(...)] REFERENCES primary-table-name [(...)]** – 別のテーブルのプライマリキーまたは一意性制約を参照する外部キーを定義します。通常、外部キーは一意性制約のためのものではなく、プライマリキーのためのものです (この説明では、この他方のテーブルをプライマリテーブルと呼びます)。

プライマリテーブルのカラム名が指定されていない場合、プライマリテーブルのカラムは、テーブルのプライマリキーの中のカラムになります。外部キーカラム名が指定されていない場合、外部キーカラムは、プライマリテーブルの中のカラムと同じ名前になります。外部キーカラム名が指定されている場合は、プライマリキーのカラム名を指定する必要があります。これらのカラム名は、リスト内の位置に応じて一対一になります。

プライマリテーブルと外部キーテーブルが同じでない場合、参照されるキーには一意性制約またはプライマリキー制約を定義する必要があります。参照されるキーと外部キーは、カラムの数、データ型、符号、精度、位取りが同じでなくてはなりません。

ローの外部キーの値は、外部キー内の 1 つまたは複数のカラムで、null を許可する外部キーカラムに null が格納されている場合を除き、プライマリテーブルのいずれかのローに格納された候補キー値として出現する必要があります。

明示的に定義されない外部キーのカラムは、プライマリテーブルの対応するカラムと同じデータ型で自動的に作成されます。これらの自動的に作成されたカラムは外部テーブルのプライマリキーの一部にはなりません。したがって、プライマリキーと外部キーの両方の中で使われるカラムは、明示的に作成する必要があります。

role-name は外部キーの名前です。*role-name* の主な機能は、同じテーブルに対する 2 つの外部キーを区別することです。*role-name* が指定されていない場合、*role-name* は次のように割り当てられます。

1. テーブル名と同じ *role-name* を含む外部キーが存在しない場合、テーブル名が *role-name* として割り当てられます。
2. テーブル名がすでに使用されている場合、*role-name* は、0 が埋め込まれた、テーブルに固有の 3 桁の数字と結合されたテーブル名になります。

参照整合性アクションは、データベース内で外部キー関係を維持するために取られるアクションを定義します。データベーステーブルからプライマリキー値が変更されたり削除されると、それに対応して何らかの修正が必要となる外部キー値が他のテーブルに存在する可能性があります。ON DELETE 句の後に続けて RESTRICT 句を指定できます。

- **RESTRICT** – データベースのどこかに対応する外部キーがあるにもかかわらず、プライマリキー値を更新または削除しようとすると、エラーになります。外部キーを更新して、候補キーに一致しない新しい値を作成しよう

とすると、エラーになります。この動作はデフォルトです。ただし、オプションで参照整合性に違反するローを拒否するように LOAD を指定した場合を除きます。これにより、文レベルでの参照整合性が確保されます。

アクションを何も指定しないで CHECK ON COMMIT を使用すると、RESTRICT は DELETE のアクションに対して適用されます。SAP Sybase IQ は CHECK ON COMMIT をサポートしません。

グローバルテンポラリテーブルは、ベーステーブルを参照する外部キーを持つことはできません。また、ベーステーブルも、グローバルテンポラリテーブルを参照する外部キーを持つことはできません。ローカルテンポラリテーブルは、外部キーを持つことも、外部キーによって参照されることもできません。

- **CHECK (条件)** – 条件に合わないローは許可されません。INSERT 文によってローがこの条件を満たさなくなる場合、操作は許可されず、この文の効果は取り消されます。

変更は、条件が FALSE の場合にのみ拒否されます。条件が UNKNOWN の場合、変更は許可されます。CHECK 条件は SAP Sybase IQ では強制されません。

注意： 孤立した外部キーがないと確実にわかっている場合を除き、できる限り、SAP Sybase IQ に参照整合性(すなわち、外部キーとプライマリキーの関係)を定義しないでください。

- **リモートテーブル** – 外部キー定義は、リモートテーブルでは無視されます。リモートテーブルを参照するローカルテーブルの外部キー定義も無視されます。プライマリキー定義は、サーバがプライマリキーをサポートする場合、リモートサーバに送信されます。
- **PARTITION BY** – 大きなテーブルを、より小さく管理しやすいストレージオブジェクトに分割します。各パーティションは親テーブルと同じ論理属性を共有しますが、別々の DB 領域に配置して個別に管理できます。SAP Sybase IQ は、次のような複数のテーブル分割スキームをサポートしています。
 - ハッシュパーティション
 - 範囲パーティション
 - 複合パーティション

partition-key は、テーブル分割キーが格納されている 1 つまたは複数のカラムです。分割キーには、NULL 値 および DEFAULT 値を含めることができますが、次のカラムを含めることはできません。

- LOB (BLOB または CLOB) カラム
- BINARY または VARBINARY カラム

- 長さが 255 バイトを超える CHAR または VARCHAR カラム
- BIT カラム
- FLOAT/DOUBLE/REAL カラム
- **PARTITION BY RANGE** – 分割カラム内の値の範囲によってローを分割します。範囲分割は、単一の分割キーカラムおよび最大 1024 パーティションまでに制限されています。range-partitioning-scheme 内の partition-key は、テーブル分割キーが格納されているカラムです。

```
range-partition-decl:  
  partition-name VALUES <= ( { constant-expr | MAX } [ ,  
  { constant-expr | MAX } ]... )  
  [ IN dbspace-name ]
```

partition-name は、テーブルローが格納される新しいパーティションの名前です。パーティション名は、テーブル上にあるパーティションセット内でユニークである必要があります。パーティション名は必須です。

- **VALUE** – 各パーティションの包括的な上限を (昇順に) 指定します。ユーザは、各ローが 1 つのパーティションのみに分配されるように、各範囲分割の分割基準を指定する必要があります。NULL は分割カラムに使用でき、NULL を分割キー値に含んだローは最初のテーブル分割に属します。ただし、NULL をバインド値に指定することはできません。

最初のパーティションには、下限 (MIN 値) は設定されていません。分割キーの最初のカラムにある NULL セルのローは、最初のパーティションに移動します。最後のパーティションでは、包括的な上限または MAX を指定できます。最後のパーティションの上限値が MAX でない場合は、最後のパーティションの上限値よりも大きい分割キーの値を含んだローをロードまたは挿入すると、エラーが生成されます。

- **MAX** – 無制限の上限を示し、最後のパーティションに対してのみ指定できます。
- **IN-partition-decl** でパーティションのローが存在する DB 領域を指定します。

次の制限を設定すると、範囲分割されたテーブルの分割キーとバインド値がその影響を受けます。

- パーティションバインドは定数式でなく、定数として指定する必要があります。
- パーティションバインドは、パーティションの作成順に応じて、昇順で指定する必要があります。つまり、2 番目のパーティションの上限は最初のパーティションよりも高く指定する必要があります、というようになります。さらに、パーティションバインドの値は、対応する分割キーカラムのデータ型と互換性がなければなりません。たとえば、VARCHAR は CHAR と互換性があります。

- バインド値に対応する分割キーのカラムとは異なるデータ型が指定されていると、SAP Sybase IQ はバインド値を分割キーのカラムのデータ型に変換します。ただし、次の場合は例外となります。
- 明示的な変換は使用できません。この例では、INT から VARCHAR に明示的に変換しようとしてエラーが生成されます。

```
CREATE TABLE Employees(emp_name VARCHAR(20))
PARTITION BY RANGE(emp_name)
(p1 VALUES <=(CAST (1 AS VARCHAR(20))),
p2 VALUES <=(CAST (10 AS VARCHAR(20)))
```

- データロスにつながる暗黙的な変換は使用できません。この例では、パーティションバインドは分割キー型と互換性がありません。丸めを前提で処理を行うとデータロスにつながる可能性があり、エラーが生成されます。

```
CREATE TABLE emp_id (id INT) PARTITION BY RANGE(id) (p1 VALUES
<= (10.5), p2 VALUES <= (100.5))
```

- この例では、パーティションバインドと分割キーのデータ型の間には互換性があります。バインド値は FLOAT 値に直接変換されます。丸め処理は必要なく、変換はサポートされています。

```
CREATE TABLE id_emp (id FLOAT)
PARTITION BY RANGE(id) (p1 VALUES <= (10),
p2 VALUES <= (100))
```

- 非バイナリデータ型からバイナリデータ型に変換することはできません。たとえば、次の変換は実行できずに、エラーが返されます。

```
CREATE TABLE newemp (name BINARY)
PARTITION BY RANGE(name)
(p1 VALUES <= ("Maarten"),
p2 VALUES <= ("Zymlerman"))
```

- NULL を範囲分割テーブルで境界として使用することはできません。
- 分割キーの最初のカラムのセル値が NULL と評価された場合、ローは最初のパーティションに挿入されます。SAP Sybase IQ は、1つのカラムの分割キーのみをサポートしているため、分割キー内に NULL が含まれていると、ローは最初のパーティションに分配されます。
- **PARTITION BY HASH** - 内部ハッシュ関数によって処理された分割キーの値に基づいて、データをパーティションにマップします。ハッシュ分割キーは最大 8 カラムで、組み合わせた宣言カラム幅が 5300 バイト以下に制限されています。ハッシュパーティションの場合、テーブル作成者は分割キーカラムのみを決定します。パーティションの数と位置は内部的に決定されます。

hash-partitioning 宣言内の partition-key は、1つのカラムまたはカラムのグループです。その複合値によってデータの各ローが格納されるパーティションが決まります。

```
hash-partitioning-scheme:
HASH ( partition-key [ , partition-key, ... ] )
```

- **制限事項** -

- ハッシュ分割できるのはベーステーブルのみです。グローバルテンポラリテーブルやローカルテンポラリテーブルを分割しようとすると、エラーが発生します。
 - ハッシュパーティションの追加、削除、マージ、分割はできません。
 - カラムをハッシュ分割キーから追加または削除することはできません。
- PARTITION BY HASH RANGE** – ハッシュ分割されたテーブルを範囲によってさらに分割します。hash-range-partitioning-scheme 宣言内の SUBPARTITION BY RANGE 句は、新しい範囲サブパーティションを既存のハッシュ範囲分割テーブルに追加します。

```
hash-range-partitioning-scheme:
PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
    [ SUBPARTITION BY RANGE ( range-partition-decl [ , range-
partition-decl ... ] ) ]
```

ハッシュパーティションはデータの論理的な配分および配置方法を指定するのに対して、範囲サブパーティションはデータの物理的な配置方法を指定します。新しい範囲サブパーティションは、既存のハッシュ範囲分割テーブルと同じハッシュ分割キーを持つハッシュによって論理的に分割されます。範囲サブパーティションは1つのカラムに制限されています。

• **制限事項 –**

- ハッシュ分割できるのはベーステーブルのみです。グローバルテンポラリテーブルやローカルテンポラリテーブルを分割しようとすると、エラーが発生します。
- ハッシュパーティションの追加、削除、マージ、分割はできません。
- カラムをハッシュ分割キーから追加または削除することはできません。

注意： 範囲パーティションと複合分割スキームは、ハッシュ範囲パーティションと同様、個別にライセンスが必要な VLDB Management オプションを必要とします。

例

(先頭に戻る) (511 ページ)

- 例 1** – 5つのカラムを持つ SalesOrders2 という名前のテーブルを作成します。カラム FinancialCode、OrderDate、ID のデータページは DB 領域 Dsp3 にあります。整数カラム CustomerID のデータページは DB 領域 Dsp1 にあります。CLOB カラム History のデータページは DB 領域 Dsp2 にあります。プライマリキー (ID の HG) のデータページは DB 領域 Dsp4 にあります。

```
CREATE TABLE SalesOrders2 (
    FinancialCode CHAR(2),
    CustomerID int IN Dsp1,
    History CLOB IN Dsp2,
```

```
OrderDate TIMESTAMP,
ID BIGINT,
PRIMARY KEY(ID) IN Dsp4
) IN Dsp3
```

- 例 2** - 4つのカラムを持つ `fin_code2` というテーブルを作成します。カラム `code`、`type`、`id` のデータページはデフォルトの DB 領域にあり、データベースオプション `DEFAULT_DBSpace` の値によって決まります。CLOB カラム `description` のデータページは DB 領域 `Dsp2` にあります。外部キー `fk1` (`c1` の HG) のデータページは DB 領域 `Dsp4` にあります。

```
CREATE TABLE fin_code2 (
code INT,
type CHAR(10),
description CLOB IN Dsp2,
id BIGINT,
FOREIGN KEY fk1(id) REFERENCES SalesOrders(ID) IN Dsp4
)
```

- 例 3** - テーブル `t1` を作成します。ここで、パーティション `p1` はパーティション `p2` に、パーティション `p2` はパーティション `p3` に隣接します。

```
CREATE TABLE t1 (c1 INT, c2 INT)
PARTITION BY RANGE(c1)
(p1 VALUES <= (0), p2 VALUES <= (10), p3 VALUES <= (100))
```

- 例 4** - 6つのカラムと3つのパーティションを持つ、RANGE 分割されたテーブル `bar` を作成し、データを日付に基づいたパーティションにマッピングします。

```
CREATE TABLE bar (
c1 INT IQ UNIQUE(65500),
c2 VARCHAR(20),
c3 CLOB PARTITION (P1 IN Dsp11, P2 IN Dsp12,
P3 IN Dsp13),
c4 DATE,
c5 BIGINT,
c6 VARCHAR(500) PARTITION (P1 IN Dsp21,
P2 IN Dsp22),
PRIMARY KEY (c5) IN Dsp1
PARTITION BY RANGE (c4)
(P1 VALUES <= ('2006/03/31') IN Dsp31,
P2 VALUES <= ('2006/06/30') IN Dsp32,
P3 VALUES <= ('2006/09/30') IN Dsp33
) ;
```

各パーティションのデータページ割り付け：

パーティション	DB 領域	カラム
P1	Dsp31	c1、c2、c4、c5
P1	Dsp11	c3

パーティション	DB 領域	カラム
P1	Dsp21	c6
P2	Dsp32	c1、c2、c4、c5
P2	Dsp12	c3
P2	Dsp22	c6
P3	Dsp33	c1、c2、c4、c5、c6
P3	Dsp13	c3
P1、P2、P3	Dsp1	c1 および他の共有データのルックアップストア
P1、P2、P3	Dsp2	プライマリキー (c5 の HG)

- 例 5** – HASH 分割された (table tbl42) を作成します。このテーブルは、PRIMARY KEY (カラム c1) と HASH PARTITION KEY (カラム c4 および c3) を含みます。

```
CREATE TABLE tbl42 (
  c1 BIGINT NOT NULL,
  c2 CHAR(2) IQ UNIQUE(50),
  c3 DATE IQ UNIQUE(36524),
  c4 VARCHAR(200),
  PRIMARY KEY (c1)
)
PARTITION BY HASH ( c4, c3 )
```

- 例 6** – PRIMARY KEY (カラム c1)、ハッシュ分割キー (カラム c4 および c2)、および範囲サブ分割キー (カラム c3) を指定して、ハッシュ範囲分割されたテーブルを作成します。

```
CREATE TABLE tbl42 (
  c1 BIGINT NOT NULL,
  c2 CHAR(2) IQ UNIQUE(50),
  c3 DATE,
  c4 VARCHAR(200),
  PRIMARY KEY (c1)) IN Dsp1

PARTITION BY HASH ( c4, c2 )
SUBPARTITION BY RANGE ( c3 )
( P1 VALUES <= (2011/03/31) IN Dsp31,
  P2 VALUES <= (2011/06/30) IN Dsp32,
  P3 VALUES <= (2011/09/30) IN Dsp33) ;
```

- 例 7** – 図書データベース用にテーブルを作成し、貸出された図書の情報を保持します。


```
CREATE TABLE borrowed_book (
  date_borrowed DATE NOT NULL,
  date_returned DATE,
  book          CHAR(20)
                REFERENCES library_books (isbn),
  CHECK( date_returned >= date_borrowed )
)
```

- **例 8** – リモートサーバ `SERVER_A` にテーブル `t1` を作成し、リモートテーブルにマッピングされるプロキシテーブル `t1` を作成します。

```
CREATE TABLE t1
( a INT,
  b CHAR(10))
AT 'SERVER_A.dbl.joe.t1'
```

- **例 9** – カラム `c1` に特殊定数 `LAST USER` のデフォルト値を含むテーブル `tab1` を作成します。

```
CREATE TABLE tab1(c1 CHAR(20) DEFAULT LAST USER)
```

- **例 10** – カラム `c1` を持つローカルテンポラリテーブル `tab1` を作成します。

```
CREATE LOCAL TEMPORARY TABLE tab1(c1 int) IN IQ_SYSTEM_TEMP
```

この例では、次の場合、`tab1` が `IQ_SYSTEM_TEMP` DB 領域に作成されます。

- `DQP_ENABLED` 論理サーバポリシーオプションが `ON` に設定されているが、読み取り／書き込み可能なファイルが `IQ_SHARED_TEMP` に存在しない場合
- `DQP_ENABLED` オプションが `OFF`、`TEMP_DATA_IN_SHARED_TEMP` 論理サーバポリシーオプションが `ON` に設定されているが、読み取り／書き込み可能なファイルが `IQ_SHARED_TEMP` に存在しない場合
- `DQP_ENABLED` オプションと `TEMP_DATA_IN_SHARED_TEMP` オプションが両方とも `OFF` に設定されている場合

この例では、次の場合、同じテーブル `tab1` が `IQ_SHARED_TEMP` DB 領域に作成されます。

- `DQP_ENABLED` が `ON` で、読み取り／書き込み可能なファイルが `IQ_SHARED_TEMP` に存在する場合
- `DQP_ENABLED` が `OFF`、`TEMP_DATA_IN_SHARED_TEMP` が `ON` で、読み取り／書き込み可能なファイルが `IQ_SHARED_TEMP` に存在する場合
- **例 11** – インメモリ `RLV` ストアでローレベルのバージョン管理とリアルタイムストレージを使用できるようにしたテーブル `tab1` を作成します。

```
CREATE TABLE tab1 ( c1 INT, c2 CHAR(25) ) ENABLE RLV STORE
```

使用法

(先頭に戻る) (511 ページ)

所有者名を指定することにより、別のユーザが使用するテーブルを作成できます。GLOBAL TEMPORARY または LOCAL TEMPORARY が指定されていない場合、テーブルはベーステーブルと呼ばれます。指定すると、テーブルはテンポラリテーブルとなります。

作成されたグローバルテンポラリテーブルは、ベーステーブルと同様にデータベース内に存在し、DROP TABLE 文によって明示的に削除されるまでデータベース内に残ります。テンポラリテーブル内のローは、ローを挿入した接続だけが参照できます。同じ、または異なるアプリケーションからの複数の接続が、同じテンポラリテーブルを同時に使用することもできます。このとき、それぞれの接続で参照できるのは自身のローだけです。この接続は、最初にグローバルテンポラリテーブルを参照し、存在していれば、そのテーブルのスキーマを継承します。接続が終了すると、テンポラリテーブルのローは削除されます。

ローカルテンポラリテーブルを作成するときは、所有者を指定しないでください。たとえば、テンポラリテーブルの作成時に、CREATE TABLE dbo.#temp(col1 int) のように所有者を指定すると、ベーステーブルが間違っって作成されます。

その接続に同じ名前のローカルテンポラリテーブルがある場合、ベーステーブルまたはグローバルテンポラリテーブルを作成しようとしても失敗します。これは、新しいテーブルを owner.table がユニークに識別できないからです。

ただし、既存のベーステーブルまたはグローバルテンポラリテーブルとしてなら、ローカルテンポラリテーブルを同じ名前で作成できます。テーブル名への参照は、ローカルテンポラリテーブルにアクセスします。ローカルテンポラリテーブルが最初に解決されるからです。

次のシーケンス例を見てみましょう。

```
CREATE TABLE t1 (c1 int);
INSERT t1 VALUES (9);

CREATE LOCAL TEMPORARY TABLE t1 (c1 int);
INSERT t1 VALUES (8);

SELECT * FROM t1;
```

返される結果は 8 です。ローカルテンポラリテーブルが接続により削除されるまで、t1 に対する参照はいずれも、ローカルテンポラリテーブル t1 を参照します。

プロシージャの完了後も保持されるテーブルを作成する場合、DECLARE LOCAL TEMPORARY TABLE 文ではなく CREATE LOCAL TEMPORARY TABLE 文をプロシージャに使用します。CREATE LOCAL TEMPORARY TABLE 文を使用して作成されたローカルテンポラリテーブルは、明示的に削除するか接続が終了するまで保持されます。

CREATE LOCAL TEMPORARY TABLE を使用して IF 文で作成されたローカルテンポラリテーブルは、IF 文が完了した後も保持されます。

SAP Sybase IQ では、SAP Sybase IQ テーブルのテーブルレベルの暗号化の CREATE TABLE ENCRYPTED 句はサポートされていません。ただし、CREATE TABLE ENCRYPTED 句は、SAP Sybase IQ データベースの SQL Anywhere テーブルではサポートされています。

関連する動作：

- オートコミット

標準

(先頭に戻る) (511 ページ)

- SQL – ISO/ANSI SQL 文法のベンダ拡張。
次にベンダ拡張を示します。
 - { **IN | ON** } *dbspace-name* 句
 - **ON COMMIT** 句
 - いくつかのデフォルト値
- SAP Sybase Database 製品 - Adaptive Server でサポートされていますが、いくつか相違があります。
 - **テンポラリテーブル** – ポンド記号 (#) を持つ **CREATE TABLE** 文でテーブル名に先行して、テンポラリテーブルを作成することができます。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。宣言されたテンポラリテーブルの詳細については、「**DECLARE LOCAL TEMPORARY TABLE** 文」を参照してください。
 - **物理的配置** – テーブルの物理的配置は SAP Sybase IQ と Adaptive Server では方法が異なります。Adaptive Server によってサポートされている **ON segment-name** 句は SAP Sybase IQ でもサポートされていますが、*segment-name* は IQ の DB 領域を参照します。
 - **制約** – SAP Sybase IQ は名前付き制約と名前付きデフォルトをサポートしませんが、制約とデフォルトの定義をデータ型定義にカプセル化できるユーザ定義データ型はサポートします。また、**CREATE TABLE** 文の明示的なデフォルトと **CHECK** 条件もサポートします。
 - **NULL** – (デフォルト) デフォルトで、Adaptive Server のカラムは、NOT NULL にデフォルト設定されていますが、SAP Sybase IQ でのデフォルト設定は NULL であり、NULL 値が許可されています。この設定は、**ALLOW_NULLS_BY_DEFAULT** オプションを使用して制御できます。「**ALLOW_NULLS_BY_DEFAULT** オプション [TSQL]」を参照してください。

データ定義文を転送可能にするには、NULL または NOT NULL を明示的に指定します。

パーミッション

(先頭に戻る) (511 ページ)

テーブルタイプ	必要な権限
IQ メインストア内のベーステーブル	<p>自分が所有するテーブル – テ이블が作成される DB 領域に対する CREATE 権限が必要。次のいずれかも必要。</p> <ul style="list-style-type: none"> • CREATE TABLE システム権限 • CREATE ANY OBJECT システム権限 <p>任意のユーザが所有するテーブル – テ이블が作成される DB 領域に対する CREATE 権限が必要。次のいずれかも必要。</p> <ul style="list-style-type: none"> • CREATE ANY TABLE システム権限 • CREATE ANY OBJECT システム権限
グローバルテンポラリテーブル	<p>自分が所有するテーブル – 次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE TABLE システム権限 • CREATE ANY OBJECT システム権限 <p>任意のユーザが所有するテーブル – 次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE ANY TABLE システム権限 • CREATE ANY OBJECT システム権限
プロキシテーブル	<p>自分が所有するテーブル – 次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE PROXY TABLE システム権限 • CREATE ANY TABLE システム権限 • CREATE ANY OBJECT システム権限 <p>任意のユーザが所有するテーブル – 次のいずれかが必要。</p> <ul style="list-style-type: none"> • CREATE ANY TABLE システム権限 • CREATE ANY OBJECT システム権限

参照：

- 制限事項 (159 ページ)
- 範囲パーティション (161 ページ)
- ハッシュパーティション (162 ページ)

- ハッシュ範囲パーティション (163 ページ)

DROP 文

データベースからオブジェクトを削除します。

クイックリンク：

「パラメータ」 (531 ページ)

「例」 (533 ページ)

「使用法」 (533 ページ)

「標準」 (533 ページ)

「パーミッション」 (533 ページ)

構文

DROP

```
{ DBSPACE dbspace-name
| { DATATYPE [ IF EXISTS ]
| DOMAIN } datatype-name
| EVENT [ IF EXISTS ] event-name
| INDEX [ IF EXISTS ] [ [ owner ]. table-name . ] index-name
| MESSAGE message-number
| TABLE [ IF EXISTS ] [ owner . ] table-name
| VIEW [ IF EXISTS ] [ owner . ] view-name
| MATERIALIZED VIEW [ IF EXISTS ] [ owner . ] view-name
| PROCEDURE [ IF EXISTS ] [ owner . ] procedure-name
| FUNCTION [ IF EXISTS ] [ owner . ] function-name }
```

パラメータ

(先頭に戻る) (531 ページ)

- **IF EXISTS** – この **DROP** 文が存在しないデータベースオブジェクトを削除しようとしたときに、エラーを返さないようにする場合に使用します。
- **INDEX** – 明示的に作成されたインデックスがあれば削除します。一意性制約または外部キー制約、または関連付けられたプライマリキーがない場合は、暗黙的に作成されたインデックスだけが削除されます。

DROP INDEX ユニークでない HG インデックスに、強制力のない外部キーが関連付けられているとエラーになります。

警告！ DBO ユーザが所有するビューを削除しないでください。このようなビューを削除したり、テーブルに変更したりすると、問題が起きる可能性があります。

DROP INDEX は、文が他の接続で現在使用中のテーブルに影響するときには処理されません。

- **TABLE – DROP TABLE** は、プライマリテーブルに外部キー制約 (強制力のない外部キー制約を含む) が関連付けられている場合は処理されません。

テーブルに **IDENTITY** カラムがあって **IDENTITY_INSERT** がこのテーブルに設定されている場合も、**DROP TABLE** は処理されません。そのテーブルを削除するには、**IDENTITY_INSERT** をクリア、すなわち空の文字列 ('') に設定するか、別のテーブル名に設定します。

外部キーには、ユニークでないシングルまたはマルチカラムの **HG** インデックスを設定できます。プライマリキーは、ユニークなシングルまたはマルチカラムの **HG** インデックスを設定できます。既存の外部キー制約、プライマリキー制約、一意性制約に対して暗黙的に作成された **HG** インデックスを削除することはできません。

初期の **DB** 領域は **SYSTEM**、**IQ_SYSTEM_MAIN**、**IQ_SYSTEM_TEMP**、**IQ_SYSTEM_MSG** の 4 つです。これらの初期の **DB** 領域を削除することはできませんが、複数の **DB** 領域を含んだ **IQ** メインストアまたはカタログストアからは **DB** 領域を削除できます。ただし、少なくとも 1 つの **DB** 領域が読み書きモードのままである必要があります。

DB 領域内のテーブルを削除しなければ、**DB** 領域を削除することはできません。**DB** 領域にユーザデータが残っていればエラーが返されます。これ以外の構造は、**DB** 領域が削除されるとき自動的に移動されます。**DB** 領域を削除できるのは、**DB** 領域を読み込み専用指定した後だけです。

注意： コマンドによって使用された後の **DB** 領域には、コマンド使用後のどの時点のデータも含まれている可能性があるため、その **DB** 領域に対する **DROP DBSPACE** は処理されません。

DROP TABLE は、文が他の接続で現在使用中のテーブルに影響するときには処理されません。

- **PROCEDURE – DROP PROCEDURE** はプロシージャが別の接続で使用中の場合は処理されません。
- **DATATYPE – DROP DATATYPE** はデータ型がテーブルで使用されている場合、処理されません。データ型を削除するには、ユーザ定義データ型を持つすべてのカラムのデータ型を変更する必要があります。**DROP DATATYPE** ではなく **DROP DOMAIN** の使用をおすすめします。これは、**DROP DOMAIN** が **ANSI/ISO SQL3** のドラフトで使用されている構文であるためです。

例

(先頭に戻る) (531 ページ)

- **例 1** - データベースから Departments テーブルを削除します。

```
DROP TABLE Departments
```

- **例 2** - データベースから emp_dept ビューを削除します。

```
DROP VIEW emp_dept
```

- **例 3** - 接続されているシンプレックスノードまたはマルチプレックスノードから myDAS メインキャッシュを削除します。

```
DROP DBSPACE myDAS
```

使用法

(先頭に戻る) (531 ページ)

DROP は指定したデータベース構造の定義を削除します。構造が DB 領域である場合、DB 領域を削除する前に、DB 領域内のデータを含むすべてのテーブルを削除または移動する必要があります。これ以外の構造の場合は自動的に移動されます。構造がテーブルである場合、テーブル内のデータは削除プロセスの一部として自動的に削除されます。同様に、テーブルのすべてのインデックスとキーも、**DROP TABLE** によって削除されます。

文が別の接続で現在使用されているテーブルに影響する場合、**DROP DBSPACE** は処理されません。別の接続で使用中のプロシージャも、**DROP PROCEDURE** では処理されません。

関連する動作

- オートコミット。dbisql の [データ] ウィンドウは閉じられます。**DROP TABLE** および **DROP INDEX** では、現在の接続のすべてのカーソルが閉じられます。
- ローカルテンポラリテーブルは例外です。削除されると、コミットは実行されません。

標準

(先頭に戻る) (531 ページ)

- SQL - ISO/ANSI SQL 準拠。
- SAP Sybase Database 製品 - Adaptive Server でサポートされています。

パーミッション

(先頭に戻る) (531 ページ)

DBSPACE 句 - DROP ANY OBJECT システム権限が必要です。ユーザが当該データベースに対して排他的な接続を保持している必要があります。

DOMAIN 句 - 次のいずれかが必要です。

- DROP DATATYPE システム権限
- DROP ANY OBJECT システム権限
- オブジェクトを所有している

FUNCTION 句 - 次のいずれかが必要です。

- DROP ANY PROCEDURE システム権限
- DROP ANY OBJECT システム権限
- 関数を所有している

INDEX 句 - 次のいずれかが必要です。

- DROP ANY INDEX システム権限
- DROP ANY OBJECT システム権限
- インデックス付きのテーブルに対する REFERENCES 権限
- インデックスの基本となるテーブルを所有している

DBA または適切な権限を持つユーザは、完全に修飾された名前を使用しなくても、別のユーザが所有するテーブルのインデックスを削除できます。DBA が所有するベーステーブルのインデックスを削除する場合、他のユーザはすべて完全に修飾されたインデックス名を指定する必要があります。

MATERIALIZED VIEW 句 - 次のいずれかが必要です。

- DROP ANY MATERIALIZED VIEW システム権限
- DROP ANY OBJECT システム権限
- マテリアライズドビューを所有している

PROCEDURE 句 - 次のいずれかが必要です。

- DROP ANY PROCEDURE システム権限
- DROP ANY OBJECT システム権限
- プロシージャを所有している

TABLES 句 - 次のいずれかが必要です。

- DROP ANY TABLE システム権限
- DROP ANY OBJECT システム権限
- テーブルを所有している

このテンポラリテーブルを参照したすべてのユーザが切断されるまで、グローバルテンポラリテーブルは削除されません。

VIEW 句 - 次のいずれかが必要です。

- DROP ANY VIEW システム権限
- DROP ANY OBJECT システム権限
- ビューを所有している

それ以外のすべての句 - 次のいずれかが必要です。

- DROP ANY OBJECT システム権限
- オブジェクトを所有している

DROP LOGICAL SERVER 文

ユーザ定義の論理サーバを削除します。次の文では、論理サーバで共有される物理ノード間で、一貫した共有システムテナポラリストア設定が強制的に適用されます。

クイックリンク：

「パラメータ」 (535 ページ)

「例」 (535 ページ)

「使用法」 (536 ページ)

「パーミッション」 (536 ページ)

構文

```
DROP LOGICAL SERVER logical-server-name
[ WITH STOP SERVER ]
```

パラメータ

(先頭に戻る) (535 ページ)

- **WITH STOP SERVER** - TEMP_DATA_IN_SHARED_TEMP オプションが直接的または間接的に変更された場合、論理サーバ内のすべてのサーバを自動的に停止します。

適用対象

マルチプレックスのみ。

例

(先頭に戻る) (535 ページ)

- **例 1** - ユーザ定義の論理サーバ `ls1` を削除します。

```
DROP LOGICAL SERVER ls1
```

使用法

(先頭に戻る) (535 ページ)

SAP Sybase IQ は、論理サーバの削除時に、次のカタログ変更を内部で実行します。

- 論理サーバのすべてのメンバーシップ定義を削除します。
- 削除対象の論理サーバへの明示的な割り当てが存在するログインポリシーのそれぞれから論理サーバの割り当てを削除します。削除対象の論理サーバがログインポリシーに割り当てられている唯一の論理サーバである場合、SAP Sybase IQ によってそのログインポリシーの論理サーバ割り当てが NONE に設定されます。
- 論理サーバエントリを ISYSIQ.LOGICALSERVER から削除します。

パーミッション

(先頭に戻る) (535 ページ)

MANAGE MULTIPLEX システム権限が必要です。

GRANT INTEGRATED LOGIN 文

1 つまたは複数の Windows ユーザプロファイルと既存のデータベースユーザ ID の間に、明示的な統合ログインマッピングを作成します。これによって、ローカルマシンにログインできたユーザは、ユーザ ID またはパスワードを入力しなくてもデータベースに接続できます。

クイックリンク：

「パラメータ」 (536 ページ)

「標準」 (537 ページ)

「パーミッション」 (537 ページ)

構文

GRANT INTEGRATED LOGIN

```
TO user_profile_name [, ...]  
AS USER userID [, ...]
```

パラメータ

(先頭に戻る) (536 ページ)

- **userID** – ログインパスワードを持つ既存のユーザまたはロールの名前になります。複数の userID はカンマで区切ります。

標準

(先頭に戻る) (536 ページ)

- SQL – その他の構文は、ISO/ANSI SQL 文法のベンダ拡張です。
- SAP Sybase Database 製品 - セキュリティモデルは Adaptive Server と SAP Sybase IQ では異なるため、他の構文も異なります。

パーミッション

(先頭に戻る) (536 ページ)

MANAGE ANY USER システム権限が必要です。

IQ UTILITIES 文

バッファキャッシュの統計を収集するキャッシュモニタを起動します。

クイックリンク：

「パラメータ」 (538 ページ)

「例」 (540 ページ)

「使用法」 (540 ページ)

「標準」 (541 ページ)

「パーミッション」 (541 ページ)

構文

```
IQ UTILITIES { MAIN | PRIVATE }
  [ INTO ] table-name
  { START MONITOR [ 'monitor-options' ]
  | STOP MONITOR }
```

monitor-options

```
{ -summary
| {-append | -truncate } -bufalloc
| -cache
| -cache_by_type
| -contention
| -debug
| -file_suffix suffix
| -io
| -interval seconds
| -threads }...
```

パラメータ

(先頭に戻る) (537 ページ)

- **START MONITOR** – IQ バッファキャッシュモニタを開始します。
- **MAIN** – IQ ストアのメインバッファキャッシュ内のテーブルをすべてモニタリングします。
- **PRIVATE** – テンポラリストアのテンポラリバッファキャッシュ内のテーブルをすべてモニタリングします。
- **dummy_table_name** – SAP Sybase IQ の任意のベーステーブルまたはテンポラリテーブルを指定します。他の **IQ UTILITIES** コマンドと構文上の互換性を持たせるために、テーブル名を指定する必要があります。最も望ましいのは、モニタリング専用のテーブルを作成することです。
- **monitor_options** – バッファキャッシュモニタ出力を制御します。複数回指定でき、それらを引用符で囲む必要があります。

オプション	説明
-summary	メインとテンポラリの両方のバッファキャッシュの要約情報を表示。モニタオプションを何も指定しない場合は、サマリレポートを表示。使用法: <code>monitor_options -summary</code>
-cache	メインバッファキャッシュまたはテンポラリバッファキャッシュのアクティビティの詳細を表示。重要なフィールドは Finds、HR%、BWait。使用法: <code>monitor_options -cache</code>
-cache_by_type	-cache の結果を IQ ページタイプごとに集計 ([Bwaits] カラムは例外で、合計だけを表示)。この形式は、保守契約を結んでいるサポートセンタに情報を送る場合に非常に有益。使用法: <code>monitor_options -cache_by_type</code>
-file_suffix	名前 <dbname>.<connid>-<main_or_temp>-<suffix> のモニタリング出力ファイルを作成する。オプションのファイル拡張子を指定しない場合は、デフォルトのファイル拡張子 .iqmon を使用。使用法: <code>monitor_options -file_suffix {extension}</code>

オプション	説明
-io	<p>指定した期間のメインまたはテンポラリ (プライベート) のバッファ キャッシュの I/O 率と圧縮比を表示。これらのカウンタで、サーバのすべてのアクティビティを表す。情報のデバイス別集計はなし。</p> <p>使用方法:</p> <pre>monitor_options -io</pre>
-bufalloc	<p>ソート、ハッシュ、ビットマップなどのオブジェクト用にバッファ キャッシュ内の領域を予約する、メインバッファアロケータまたはテンポラリバッファアロケータの情報を表示。使用方法:</p> <pre>monitor_options -bufalloc</pre>
-contention	<p>多くの重要なバッファキャッシュとメモリマネージャロックを表示。これらのロックカウンタとミューテックスカウンタは、バッファキャッシュおよびヒープメモリ内のアクティビティと、これらのロックがどれだけ迅速に解消されたかを示す。タイムアウト数が 20% を超えている場合は、問題の発生を示す。使用方法:</p> <pre>monitor_options -contention</pre>
-threads	<p>処理スレッドマネージャのカウンタを表示。値はサーバワイド (つまり、メインとプライベートのどちらでこのオプションを選択するかは無関係)。使用方法:</p> <pre>monitor_options -threads</pre>
-interval	<p>レポート間隔を秒単位で指定。デフォルトは 60 秒ごと。最小値は 2 秒ごと。</p> <p>通常、クエリの実行中やパフォーマンスに問題があるときに、モニタをデフォルトの間隔で実行すると、有益な結果の取得が可能。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る。ジョブ時間に見合った間隔を指定する。通常は 1 分で十分。使用方法:</p> <pre>monitor_options -interval</pre>
-append -truncate	<p>前者は既存の出力ファイルに追加し、後者は既存の出力ファイルをtruncateする。デフォルトは、truncate。使用方法:</p> <pre>monitor_options -append -truncate</pre>
-debug	<p>同じ情報を扱う標準表示モードの有無にかかわらず、パフォーマンスモニタで使用可能な情報がすべて表示される。-debug は、主に、保守契約を結んでいるサポートセンタへの情報提供に使用される。使用方法:</p> <pre>monitor_options -debug</pre>

- **STOP MONITOR** – **START MONITOR** とほぼ同じですが、オプションを指定する必要はありません。

注意：

- モニタを簡単に使用するには、ストアードプロシージャを作成してダミーテーブルを宣言し、出力ロケーションを指定して、モニタを起動します。
 - 表示する間隔は、ページ単位ではなく、出力行単位です。ただし、次の2つの場合は例外です。-cache_by_type 句と -debug 句では、表示ごとに新しいページが開始されます。
-

例

(先頭に戻る) (537 ページ)

- **例 1** – バッファキャッシュモニタを開始し、IQ テンポラリバッファキャッシュのアクティビティを記録します。

```
IQ UTILITIES PRIVATE INTO monitor START MONITOR '-cache -interval 20'
```

使用法

(先頭に戻る) (537 ページ)

各バッファキャッシュをモニタリングするには、別々のコマンドを発行します。モニタが結果を収集している間は、これらの各セッションを開いておきます。接続を閉じると、モニタは実行を停止します。1つの接続で最大で2つのモニタの実行に対応できます。1つはメインバッファキャッシュ用で、もう1つはテンポラリバッファキャッシュ用です。

モニタリング出力ファイルのディレクトリ配置を制御するには、**MONITOR_OUTPUT_DIRECTORY** オプションを設定します。このオプションを設定しない場合は、データベースと同じディレクトリに結果が出力されます。モニタを実行している間、すべてのモニタリング出力ファイルが使用されます。モニタの実行が停止した後も、ファイルはそのまま残ります。

マルチプレックスクエリサーバを作成する前に、モニタリングで使用するテンポラリテーブルを宣言するか、新しいデータベースの作成時に永続的なダミーテーブルを作成してください。これによって DDL の変更を回避し、実際の運用稼働時にデータがクエリサーバに保持されるようにします。

UNIX のようなオペレーティングシステムでは、クエリの実行中にモニタリング出力を確認できます。次に例を示します。

次のコマンドでモニタを起動します。

```
iq utilities main into monitor_tab  
start monitor "-cache -interval 2 -file_suffix iqmon"
```

出力が `dbname.conn#-[main|temp]-iqmon` という名前の ASCII ファイルに送信されます。したがって、`iqdemo` データベースでは、バッファモニタから `iqdemo.2-main-iqmon` に結果が送信されます。

バッファキャッシュモニタは、各実行の結果をこれらのログに書き込みます。

- `dbname.connection#-main-iqmon //for main buffer cache results`
- `dbname.connection#-temp-iqmon //for temp buffer cache results`

プレフィクス `dbname.connection#` は、データベース名と接続番号を示します。複数の接続番号が存在して、自分のものがどれであるかが不明の場合は、カタログストアプロシージャ `sa_conn_info` を実行します。このプロシージャを実行すると、アクティブなデータベース接続のそれぞれについて、接続番号、ユーザ ID などの情報が表示されます。サフィックス `iqmon` を任意のサフィックスに変更するには、`-file_suffic` 句を使用します。テキストエディタを使用して、ファイルを表示または印刷します。同じデータベースと同じ接続番号からモニタを再実行すると、前回の結果が上書きされます。モニタの実行結果を保存する場合は、ファイルを別の場所にコピーするか、`-append` オプションを使用してください。

標準

(先頭に戻る) (537 ページ)

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

パーミッション

(先頭に戻る) (537 ページ)

なし

LOCK TABLE 文

他の同時トランザクションが指定の時間内にテーブルにアクセスしたり変更したりするのを防ぎます。

クイックリンク：

「パラメータ」 (542 ページ)

「例」 (543 ページ)

「使用法」 (543 ページ)

「標準」 (544 ページ)

「パーミッション」 (544 ページ)

構文

```
LOCK TABLE table-list [ WITH HOLD ]  
    IN { SHARE | WRITE | EXCLUSIVE } MODE [ WAIT time ]
```

table-list

```
[ owner. ] table-name [ , [ owner. ] table-name, ...]
```

パラメータ

(先頭に戻る) (541 ページ)

- **table-name** – ビューではなくベーステーブルである必要があります。WRITE モードが有効なのは、IQ ベーステーブルの場合だけです。**LOCK TABLE** はテーブルリストにあるすべてのテーブルをロックするか、1 つもロックしないかのどちらかです。テーブルは、ローレベルバージョン管理 (RLV) に対応してはいけません。SQL Anywhere テーブルのロックを取得する場合、または SHARE か EXCLUSIVE のどちらかのロックを取得するときに指定できるのは、1 つのテーブルだけです。*table-name* の解析には、標準の SAP Sybase IQ オブジェクト修飾ルールが使用されます。
- **WITH HOLD** – ロックは、接続の終わりまで保持されます。この句を指定しない場合は、現在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た時点でロックは解放されます。**WRITE MODE** を持つ同じ文での **WITH HOLD** 句の使用はサポートされていなく、エラー SQLCODE=-131, ODBC 3 State="42000" が返されます。
- **SHARE** – 他の同時トランザクションによるテーブルの変更はできませんが、読み込みアクセスは許可します。このモードでは、他のトランザクションが間接的に、または **LOCK TABLE** を使用して明示的に修正中のローをロックしていない限り、テーブルのデータを修正できます。
- **WRITE** – 他のトランザクションによるテーブルのリストの変更はできません。接続の一番外側のトランザクションを無条件にコミットします。トランザクションが使用するスナップショットバージョンは、**LOCK TABLE IN WRITE MODE** 文ではなく、SAP Sybase IQ が処理する次のコマンドの実行により確立されます。
WRITE モードロックは、トランザクションがコミットされるかロールバックされる時、または接続が切断されたときに解放されます。
- **EXCLUSIVE** – 他のトランザクションによるテーブルへのアクセスはできません。このモードでは、他のトランザクションはテーブルに対してクエリの発行、あらゆる修正、その他のあらゆる操作を実行できません。

- **WAIT time** – 全種類のロックの最長ブロック時間を指定します。この句はロックモードが **WRITE** の場合に必須です。time 引数が指定された場合、サーバは指定時間内に使用できる場合に限り、指定されたテーブルをロックします。この time 引数は、hh:nn:ss:sss の形式で指定できます。日付部分が指定されても、サーバはそれを無視し、その引数をタイムスタンプに変換します。time 引数が指定されない場合は、**WRITE** ロックが使用できるようになるか、または割り込みが発生するまで、サーバはいつまでも待ちます。

例

(先頭に戻る) (541 ページ)

- **例 1** – 次の文は、Customers と Employees の各テーブルが 5 分 3 秒以内に使用できるようになる場合、それぞれのテーブルの **WRITE** ロックを取得します。

```
LOCK TABLE Customers, Employees IN WRITE MODE WAIT
'00:05:03'
```

- **例 2** – Customers と Employees の各テーブルの **WRITE** ロックが使用できるようになるか、または割り込みが発生するまで、いつまでも待ちます。

```
LOCK TABLE Customers, Employees IN WRITE MODE WAIT
```

使用法

(先頭に戻る) (541 ページ)

コーディネータ上の IQ メインストアのテーブルに対して **LOCK TABLE** 文を実行しても、セカンダリサーバ上の接続からこれらのテーブルへのアクセスは影響を受けません。次に例を示します。

コーディネータ接続では、次のコマンドを発行します。

```
LOCK TABLE coord1 WITH HOLD IN EXCLUSIVE MODE
```

コーディネータ上で **sp_iqlocks** を実行すると、テーブル coord1 に排他 (E) ロックが存在することを確認できます。

セカンダリサーバ上の接続に対して **sp_iqlocks** を実行した結果には、テーブル coord1 に対する排他ロックが示されません。この接続上のユーザは、コーディネータ上のテーブル coord1 に対する更新を表示できます。

コーディネータ上の他の接続からは、coord1 に対する排他ロックを表示できません。コーディネータ上の別の接続からテーブル coord1 を選択しようとする、と、「User DBA has the row in coord1 locked」が返されます。

ビューでの **LOCK TABLE** はサポートされていません。ビューをロックしようすると、コマンドで指定されたモードにかかわらず、共有のスキーマロックが取得

されます。共有のスキーマロックでは、他のトランザクションによるテーブルスキーマの変更はできません。

Transact-SQL (T-SQL) ストアドプロシージャ言語は、**LOCK TABLE** をサポートしません。たとえば、次の文は、「Syntax error near LOCK」を返します。

```
CREATE PROCEDURE tproc()  
AS  
BEGIN  
COMMIT;  
LOCK TABLE t1 IN SHARE MODE  
INSERT INTO t1 VALUES(30)  
END
```

Watcom-SQL ストアドプロシージャ言語は、**LOCK TABLE** をサポートします。デフォルトでは、コマンドのデリミタはセミコロン (;) です。次に例を示します。

```
CREATE PROCEDURE tproc()  
AS  
BEGIN  
COMMIT;  
LOCK TABLE t1 IN SHARE MODE  
INSERT INTO t1 VALUES(30)  
END
```

標準

(先頭に戻る) (541 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - Adaptive Server でサポートされています。WITH HOLD 句は、Adaptive Server ではサポートされていません。Adaptive Server には WAIT 句がありますが、これは SQL Anywhere ではサポートされていません。

パーミッション

(先頭に戻る) (541 ページ)

SHARE モードでテーブルをロックするには、SELECT 権限が必要です。

EXCLUSIVE モードでテーブルをロックするには、そのテーブルの所有者であるか以下のシステム権限のいずれかが必要です。

- ALTER ANY OBJECT
- INSERT ANY TABLE
- UPDATE ANY TABLE
- DELETE ANY TABLE
- ALTER ANY TABLE
- LOAD ANY TABLE

- TRUNCATE ANY TABLE

ROLLBACK 文

最後の **COMMIT** または **ROLLBACK** の後に加えられた変更を取り消します。

クイックリンク：

「使用法」 (545 ページ)

「標準」 (545 ページ)

「許可」 (545 ページ)

構文

ROLLBACK [**WORK**]

使用法

(先頭に戻る) (545 ページ)

ROLLBACK は、作業 (トランザクション) の論理単位を終了し、このトランザクションの間にデータベースに加えられたすべての変更を取り消します。トランザクションは、1つのデータベース接続上で **COMMIT** 文または **ROLLBACK** 文の間で行われるデータベース作業です。

関連する動作：

- **WITH HOLD** 句を指定してオープンしなかったカーソルをすべてクローズします。
- **ROLLBACK** を実行して、トランザクションによって保持されているロックを開放します。

標準

(先頭に戻る) (545 ページ)

- SQL - ISO/ANSI SQL 準拠。
- SAP Sybase Database 製品 - Adaptive Server でサポートされています。

パーミッション

(先頭に戻る) (545 ページ)

なし。ただし、ユーザはデータベースに接続する必要があります。

SAVEPOINT 文

現在のデータベース内でセーブポイントを確立します。

クイックリンク：

「パラメータ」 (546 ページ)

「使用法」 (546 ページ)

「標準」 (546 ページ)

「許可」 (546 ページ)

構文

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

パラメータ

(先頭に戻る) (546 ページ)

- **savepoint-name** – **RELEASE SAVEPOINT** 文または **ROLLBACK TO SAVEPOINT** 文で使用できる識別子です。

使用法

(先頭に戻る) (546 ページ)

トランザクションが終了すると、すべてのセーブポイントは自動的に解放されません。

トリガまたはアトミックな複合文の実行中に確立されたセーブポイントは、アトミック操作が終了すると自動的に解放されます。

標準

(先頭に戻る) (546 ページ)

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - Adaptive Server ではサポートされていません。Adaptive Server と互換性のある方法で同様の機能を実装するには、トランザクションのネストを使用することができます。

パーミッション

(先頭に戻る) (546 ページ)

なし

SET OPTION 文

データベースの動作および Transact-SQL との互換性に影響を及ぼすオプションを変更します。オプションの値を設定すると、すべてのユーザまたは個別のユーザの動作を一時的または永続的なスコープで変更できます。

クイックリンク：

「パラメータ」 (547 ページ)

「例」 (548 ページ)

「使用法」 (548 ページ)

「標準」 (549 ページ)

「パーミッション」 (549 ページ)

構文

```
SET [ EXISTING ] [ TEMPORARY ] OPTION
... [ userid. | PUBLIC. ] option-name = [ option-value ]
```

パラメータ

(先頭に戻る) (547 ページ)

- **option-value** – ホスト変数 (インジケータ使用可)、文字列、識別子、または数値。文字列に設定する場合、*option-value* の最大長は 127 バイトです。

option-value を省略すると、指定されたオプション設定がデータベースから削除されます。これがユーザ個人のオプション設定の場合は、値は PUBLIC 設定に戻ります。

注意： 整数値の指定が可能なデータベースオプションの場合、SAP Sybase IQ によって *option-value* の小数設定がすべて整数値にトランケートされます。たとえば、3.8 という値は 3 にトランケートされます。

- **EXISTING** – そのオプションに PUBLIC のユーザ ID 設定がすでに存在している場合を除き、個別のユーザ ID に対するオプション値は設定できません。
- **TEMPORARY** – 変更の有効期間を変更します。TEMPORARY 句がない場合、オプションの変更は永続的です。**SET OPTION** を使用して明示的に変更されるまで、変更されません。

個別のユーザ ID を使用して TEMPORARY 句が適用された場合、そのユーザがデータベースにログインしている間だけ、新しいオプション値が有効になります。

PUBLIC ユーザ ID とともに TEMPORARY 句が使用された場合、データベースの実行中はその変更が継続されます。データベースが停止されると、PUBLIC ユーザ ID の TEMPORARY オプションは永続値に戻ります。

TEMPORARY オプションが削除されると、オプション設定は永続的な設定に戻ります。

例

(先頭に戻る) (547 ページ)

- 例 1 – DATE_FORMAT オプションを設定します。

```
SET OPTION public.date_format = 'Mmm dd yyyy'
```

- 例 2 – WAIT_FOR_COMMIT オプションを ON に設定します。

```
SET OPTION wait_for_commit = 'on'
```

- 例 3 – embedded SQL の例を示します。

```
EXEC SQL SET OPTION :user.:option_name = :value;  
EXEC SQL SET TEMPORARY OPTION Date_format = 'mm/dd/yyyy';
```

使用法

(先頭に戻る) (547 ページ)

オプションのクラスは次のとおりです。

- 一般的なデータベースオプション
- Transact-SQL 互換性データベースオプション

ユーザ ID または PUBLIC ユーザ ID を指定することで、個別ユーザ、*userid* で示されるロール、PUBLIC ユーザ ID (全ユーザがメンバーであるロール) のいずれに対してオプションが設定されるかが決まります。オプションがロール ID に適用される場合、ロールのメンバーによるオプション設定の継承は行われません。その変更はロール ID のみに適用されます。ロールが指定されない場合、そのオプション変更は現在ログオン中で **SET OPTION** 文を発行したユーザ ID に適用されます。たとえば、次の文では、オプション変更が PUBLIC ユーザ ID に適用されます。

```
SET OPTION Public.login_mode = standard
```

Embedded SQL では、データベースオプションが設定できるのは、一時的にすぎません。

PUBLIC ユーザ ID のオプション値を変更すると、独自の値を設定していないすべてのユーザにそのオプション値が設定されます。そのオプションに PUBLIC ユーザ ID 設定がすでに存在している場合を除き、個別のユーザ ID に対するオプション値は設定できません。

オプションの値を永続に設定するのではなく、PUBLIC ユーザ ID に対するオプションを一時的に設定すると、セキュリティが向上します。たとえば、**LOGIN_MODE** オプションが有効な場合、データベースは、そのデータベースを実行しているシステムのログインセキュリティに依存します。このオプションを一時的に有効にすると、Windows ドメインのセキュリティに依存しているデータベースは、データベースが停止し、ローカルマシンにコピーされた場合でも、セキュリティが損なわれることはありません。この場合、一時的に有効化された **LOGIN_MODE** は、永続値、つまり統合化ログインが許可されない Standard などに戻ります。

警告！ カーソルからローをフェッチしている際のオプション設定の変更は、予期しない動作を招く可能性があるためサポートされていません。たとえば、カーソルからのフェッチ中に **DATE_FORMAT** 設定を変更すると、結果セットで返されるローの日付フォーマットが統一されないことになります。ローをフェッチしている間にオプション設定を変更しないでください。

標準

(先頭に戻る) (547 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - Adaptive Server ではサポートされていません。SAP Sybase IQ では、**SET** 文の使用時に一部の Adaptive Server オプションをサポートしています。

パーミッション

(先頭に戻る) (547 ページ)

各自のオプションを設定する際に特に要求されるシステム権限はありません。

別のユーザのデータベース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。

PUBLIC ユーザ ID の **SYSTEM** オプションを設定するには、**SET ANY SYSTEM OPTION** システム権限が必要です。

PUBLIC ユーザ ID の **SECURITY** オプションを設定するには、**SET ANY SECURITY OPTION** システム権限が必要です。

STOP DATABASE 文 [Interactive SQL]

指定したデータベースサーバ上のデータベースを停止します。

クイックリンク：

「パラメータ」 (550 ページ)

「例」 (550 ページ)

「標準」 (550 ページ)

「パーミッション」 (550 ページ)

構文

```
STOP DATABASE database-name  
... [ ON engine-name ]  
... [ UNCONDITIONALLY ]
```

パラメータ

(先頭に戻る) (549 ページ)

- **database-name** – データベース起動時の `-n` パラメータ、または DBN (DatabaseName) 接続パラメータで指定した名前。通常、この名前は、カタログストアを格納するデータベースファイルのファイル名から `.db` 拡張子を除いたものですが、ユーザ定義の名前も使用できます。
- **engine-name** – 指定しないと、作動中のすべてのエンジンで指定した名前のデータベースが検索されます。
- **UNCONDITIONALLY** – 指定すると、データベースはそのデータベースへの接続がある場合でも停止します。指定しないと、接続がある場合には、データベースは停止しません。

例

(先頭に戻る) (549 ページ)

- **例 1** – デフォルトサーバ上のデータベース `sample` を停止します。

```
STOP DATABASE sample
```

標準

(先頭に戻る) (549 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - なし。

パーミッション

(先頭に戻る) (549 ページ)

SERVER OPERATOR システム権限が必要です。

STOP ENGINE 文 [Interactive SQL]

データベースサーバを停止します。

クイックリンク：

「パラメータ」 (551 ページ)

「例」 (551 ページ)

「標準」 (551 ページ)

「パーミッション」 (551 ページ)

構文

```
STOP ENGINE engine-name [ UNCONDITIONALLY ]
```

パラメータ

(先頭に戻る) (551 ページ)

- **UNCONDITIONALLY** - 指定すると、データベースサーバはそのサーバへの接続がある場合でも停止します。指定しないと、接続がある場合には、データベースサーバは停止しません。

例

(先頭に戻る) (551 ページ)

- **例 1** - `sample` という名前のデータベースを停止します。

```
STOP ENGINE sample
```

標準

(先頭に戻る) (551 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - なし。

パーミッション

(先頭に戻る) (551 ページ)

なし

索引

記号

-c スイッチ 20
 -contention 255
 -iqnotemp 363

A

AES 暗号化アルゴリズム
 CREATE DATABASE 文 480
 ALLOW_SNAPSHOT_VERSIONING 240
 ALTER DBSPACE 文
 ADD パラメータ 114
 SIZE パラメータ 115
 例 114
 構文 444
 ALTER INDEX 文
 エラー 449
 ALTER LOGICAL SERVER 文
 構文 453
 ALTER LS POLICY 文
 構文 454
 ALTER TABLE 文
 外部キー 213
 構文 459
 ALTER VIEW 文
 RECOMPILE 459
 ALTER 文
 オートコミット 221
 AppInfo 接続パラメータ 386
 AStart 接続パラメータ 388
 AStop 接続パラメータ 389
 AUDITING オプション 331
 AUTOINCREMENT
 デフォルト 205
 負の数 205
 AUTOINCREMENT カラムのデフォルト 511
 AutoPreCommit 接続パラメータ 388
 AutoStart 接続パラメータ 388
 AutoStop 接続パラメータ 389
 AVG 関数 179

B

BCAST 通信パラメータ
 説明 418
 BEGIN TRANSACTION 文
 リモートデータアクセス 264
 BIT データ
 使用可能なインデックス 188
 BLISTENER 通信パラメータ
 説明 419
 BLOCKING オプション 229, 230
 BLOCKING_TIMEOUT オプション 231
 Broadcast 通信パラメータ
 説明 418
 BroadcastListene 通信パラメータ
 説明 419

C

CACHE_PARTITIONS オプション 14
 CALL 文
 例 266
 CBSize 接続パラメータ 391
 CBSpace 接続パラメータ 392
 CharSet 接続パラメータ 390
 CHECK ON COMMIT 句
 参照整合性 511
 CHECK 条件
 カラム 208
 説明 511
 テーブル 210
 ユーザ定義データ型 209
 削除 211
 変更 211
 CHECKPOINT 文
 構文 477
 ClientPort 通信パラメータ 419
 CMP インデックス 169
 推奨される使用法 169
 制限 169
 CommBufferSize 接続パラメータ 391

索引

CommBufferSpace 接続パラメータ 392

COMMIT TRANSACTION 文

Transact-SQL 478

COMMIT 文

リモートデータアクセス 264

構文 478

CommLinks 接続パラメータ 393

ConnectionName 接続パラメータ 395

ConnectionPool 接続パラメータ 395

COUNT DISTINCT

インデックス選択への影響 184

COUNT 関数 179

CREATE DATABASE 文 88

IQ RESERVE パラメータ 114

IQ SIZE パラメータ 114

RESERVE パラメータ 114

SIZE パラメータ 114

TEMPORARY RESERVE パラメータ 114

TEMPORARY SIZE パラメータ 114

ローデバイス 87

構文 480

CREATE DBSPACE 文 124

CREATE DOMAIN 文

構文 493

CREATE FUNCTION 文

構文 274

CREATE INDEX 文

テーブルでの使用 495

構文 495

CREATE LOGICAL SERVER 文 504

CREATE LS POLICY 文

構文 507

CREATE TABLE 文

例 150

構文 511

CREATE TEXT INDEX 181

CREATE 文

オートコミット 221

同時実行性の規則 242

CS 接続パラメータ 390

D

DatabaseFile 接続パラメータ 396

DatabaseName 接続パラメータ 397

DatabaseName 通信パラメータ

説明 420

DatabaseSwitches 接続パラメータ 398

DataSourceName 接続パラメータ 399

DATE インデックス 170

制限 174

推奨される使用法 170

他のインデックスとの比較 174

短所 173

長所 173

追加インデックス 174

DATEPART

クエリ 170

Datetime インデックス

次を参照： DTTM インデックス

DB 領域

DB 領域の不足状況 95, 346

IQ_SHARED_TEMP 100, 103

IQ_SYSTEM_MSG 105

IQ_SYSTEM_TEMP 100

インデックス情報の表示 114

オフライン 99

オフラインに設定 444

サイズの変更 114

作成時のファイルロケーション 87

作成の例 114

システム 100

自動作成 86

使用状況情報の表示 114

ファイルロケーション 87

領域の使用状況のモニタリング 350

領域不足のエラーメッセージ 346

ローデバイスの追加 87

管理の例 114

作成 124

削除 125, 531

定義 99

読み取り専用 127

変更 444

命名 123

DB 領域をオンラインに設定 444

DBF 接続パラメータ 396

組み込みデータベース 52

- dbisql
 - コマンドラインパラメータ 50
 - 接続 50
 - 接続パラメータの例 48
 - トラブルシューティング 345
 - ポート番号 50
 - ログオンウィンドウ 51
 - DBKEY 接続パラメータ 399
 - DBN 接続パラメータ 397
 - DBN 通信パラメータ
 - 説明 420
 - dbo ユーザ ID
 - ビューの所有者 147, 531
 - DBS 接続パラメータ 398
 - DBSpaceLogger イベント 350
 - dbtran ユーティリティ 327
 - 構文 327
 - 終了コード 327
 - DDL 253
 - 説明 145
 - DDL ロック 242
 - DEDICATED_TASK オプション
 - 説明 440
 - DisableMultiRowFetch 接続パラメータ 400
 - DMRF 接続パラメータ 400
 - DOBROADCAST 通信パラメータ 421
 - DQP
 - パフォーマンス 454, 507
 - DROP DATATYPE 文
 - 構文 531
 - DROP DBSPACE 文
 - 構文 531
 - DROP DOMAIN 文
 - 構文 531
 - DROP FUNCTION 文
 - 構文 531
 - DROP INDEX 文
 - 構文 531
 - DROP LOGICAL SERVER 文 535
 - DROP PROCEDURE 文
 - 構文 531
 - DROP TABLE
 - IDENTITY_INSERT オプション 531
 - DROP TABLE 文
 - 構文 531
 - 例 156
 - DROP VIEW 文
 - 制限 147, 531
 - 構文 531
 - 例 149
 - DROP イベント
 - 構文 531
 - DROP メッセージ
 - 構文 531
 - DSN 接続パラメータ 399
 - DTTM インデックス 170
 - 推奨される使用法 170
 - 他のインデックスとの比較 174
 - 短所 173
 - 長所 173
 - 追加インデックス 174
 - DumpAllThreads ファイル 339
- ## E
- ENC 接続パラメータ
 - 説明 402
 - EncryptedPassword 接続パラメータ 401
 - Encryption 接続パラメータ
 - 説明 402
 - ENG 接続パラメータ 400
 - EngineName 接続パラメータ 400
 - ENP 接続パラメータ 401
 - Escape 接続パラメータ 403
 - Escape 403
- ## F
- FileDataSourceName 接続パラメータ 403
 - FileDSN
 - 配布 67
 - 作成 67
- ## G
- getiqinfo スクリプト 378
 - gm スイッチ 15
 - リカバリの影響 260

索引

GROUP BY 句

インデックス選択への影響 184

gt スイッチ 19

H

HASH_THRASHING_PERCENT オプション
354

HG インデックス

NULL 値 177, 495

NULL 値のマルチカラム 177, 495

外部キー制約削除時のインデックスの保持
194

クエリパフォーマンス 177

外部キー制約 176

自動作成 177

推奨される使用法 176

他のインデックスとの比較 177

短所 177

長所 177

追加インデックス 177

High_Group インデックス

次を参照：HG インデックス

High_Non_Group インデックス

次を参照：HNG インデックス

HNG インデックス 178

推奨される使用法 178

他のインデックスとの比較 179

短所 179

長所 179

追加インデックス 179

HOST 通信パラメータ 422

I

IANA

ポート番号 433

IDENTITY カラム

および DROP TABLE 531

IDENTITY_INSERT オプション

テーブルの削除 531

IDLE 接続パラメータ 404

INSERT 文 191

整合性 198

INT 接続パラメータ 405

INTEGRATED LOGIN 文

付与 536

IP アドレス 386

ping 360

IP 通信パラメータ 422

IPv6 アドレス 431

IPX

サーバ設定 417

IQ PAGE SIZE 92

IQ UNIQUE

値の変更 155

IQ UNIQUE テーブルオプション 154

IQ UNIQUE 制約 208

IQ UTILITIES

バッファキャッシュモニタ 339

IQ UTILITIES 文

構文 537

IQ ストア

ローデバイスアクセス 109

ローデバイスのパーミッション 110, 111

IQ_SYSTEM_MAIN

サイズのガイドライン 106

IQ_SYSTEM_MSG DB 領域 105

IQ_SYSTEM_TEMP 363

iqgovern スイッチ 17

iqmc スイッチ 11

iqmt スイッチ 18

iqnumbercpus

サーバスイッチ 21

iqpartition スイッチ 11

iqpartition 起動スイッチ 13

iqwmem スイッチ 17

J

Java

SAP Sybase IQ での使用 60

メモリ要件 20

jConnect ドライバ

データベースへの接続 67

JDBC 接続

説明 67

L

LANalyzer 362

LANG 接続パラメータ
 説明 406

LazyClose 接続パラメータ
 説明 406

LCLOSE 接続パラメータ
 説明 406

LDAP サーバ
 LDAP 通信パラメータ 425

LDAP 通信パラメータ
 説明 425

LF インデックス 179
 ユニークな値の最大数を超える 355
 推奨される使用法 180
 他のインデックスとの比較 180
 短所 180
 長所 180
 追加インデックス 180

LF 通信パラメータ
 説明 426

Links 接続パラメータ 393

LivenessTimeout 接続パラメータ 407

LOAD TABLE 文 191

LOCAL 通信パラメータ
 説明 425

LocalOnly 通信パラメータ
 説明 425

LOCK TABLE
 構文 541

LOG 通信パラメータ
 説明 426

LOG_CONNECT データベースオプション 440

LogFile
 接続パラメータ 408
 通信パラメータ 426

LogFormat 通信パラメータ
 説明 426

LogicalServer
 接続パラメータ 409

LOGIN_MODE データベースオプション
 統合化ログイン 73

LogOptions 通信パラメータ
 説明 428

LOPT 通信パラメータ
 説明 428

Low_Fast インデックス
 次を参照：LF インデックス

LS ポリシー 458, 510

LTO 接続パラメータ 407

M

MAX_TEMP_SPACE_PER_CONNECTION オプション 95

MAXCONN 通信パラメータ
 説明 429

MaxConnections 通信パラメータ
 説明 429

MDSR 暗号化アルゴリズム
 CREATE DATABASE 文 480

Microsoft Access 354

MIN_PASSWORD_LENGTH オプション 441

mklink ユーティリティ 113

MYIP 通信パラメータ 430

MySybase
 アクセス 382
 オンラインサポート 382

N

NBit 174

NDIS
 ドライバ 359

net.cfg ファイル 362

NetBIOS
 サーバ設定 417

NetWare
 ネットワークアダプタの設定 362

NEWID
 デフォルト 205

NodeType
 接続パラメータ 411

NOT NULL 値制約 198

Novell クライアントソフトウェア 359

NULL
 デフォルト 206
 マルチカラム HG インデックス 177, 495

NULL 値
 マルチカラムの HG インデックス 177,
 495

O

ODBC

接続パラメータ 385

ODBC データソース

jConnect の使用 67

ODBC 接続

説明 67

ODI ドライバ 359

P

Password 接続パラメータ 412

ping

TCP/IP 360

PORT 通信パラメータ 433

PreFetchOnOpen 通信パラメータ

説明 431

PREPARE 文

リモートデータアクセス 264

PWD 接続パラメータ 412

Q

QUERY_TEMP_SPACE_LIMIT オプション 95

R

rawaccredit ユーティリティ 109

rawaccess ユーティリティ 110, 111

RCVBUFSZ 通信パラメータ

説明 432

ReceiveBufferSize 通信パラメータ

説明 432

REFERENCES 句 459

RELEASE SAVEPOINT 文 257

RESTRICT アクション 511

Rigndael 暗号化アルゴリズム

CREATE DATABASE 文 480

RLV DB 領域

ファイル, 削除 121

ファイル, 追加 121

ROLLBACK TO SAVEPOINT 文 257

ROLLBACK 文 257

構文 545

S

sa_conn_info 247

sa_get_request_profile

要求ログファイルの分析 375

sa_get_request_times

要求ログファイルの分析 375

sa_report_deadlocks システムプロシージャ 233

SAP Sybase IQ

停止 40

SAVEPOINT 文

トランザクション 257

構文 546

SELECT * 459

SELECT DISTINCT 射影 184

SELECT 文

ビュー作成の制限 147

SELECT 文の制限 147

SendBufferSize 通信パラメータ

説明 432

ServerName 接続パラメータ 400

ServerPort 通信パラメータ 433

Service Manager

サービスの起動 6

SESSIONS 通信パラメータ 434

SET OPTION 文

構文 547

SET TEMPORARY OPTION 文

構文 547

SNAPSHOT_VERSIONING 246

SNDBUFSZ 通信パラメータ

説明 432

sp_iqcheckoptions ストアドプロシージャ 95

sp_iqconnection 247

sp_iqdbspace

DB 領域の使用状況情報 114

sp_iqdbspaceinfo

DB 領域の使用状況情報 114

sp_iqestdbspaces

必要な DB 領域の見積もり 113

sp_iqestspace

必要な DB 領域の見積もり 113

sp_iqindex ストアドプロシージャ 193

sp_iqindexinfo

インデックス情報の表示 114

sp_iqlocks 247
 sp_iqprocedure
 プロシージャに関する情報 267
 sp_iqprocparm
 プロシージャパラメータ 268
 sp_iqstatus
 出力例 363
 トラブルシューティングでの使用 363
 sp_iqtransaction 253
 ブロックしている書き込みの確認 356
 sp_iqtransaction システムプロシージャ 222
 SQL Anywhere JDBC ドライバ
 SAP Sybase IQ および SQL Anywhere データベースへの接続 67
 SQLCONNECT 環境変数
 接続 59
 srvlog
 接続情報の関連付け 381
 srvlog ファイル
 接続情報 381
 SSD 135
 start_iq
 コマンドが実行されない 338
 トラブルシューティング 338
 パラメータ 337
 start_iq ユーティリティ 4
 StartLine 接続パラメータ 415
 STOP DATABASE 文
 構文 549
 STOP ENGINE 文
 構文 551
 SUM 関数 179
 Sybase Control Center
 DB 領域の作成 124
 SYSCOLUMN テーブル
 整合性 220
 SYSFOREIGNKEY テーブル
 整合性 220
 SYSTABLE テーブル
 整合性 220
 SYSVIEW ビュー
 ビュー情報 149

T

TCP/IP
 BroadcastListener [BLISTENER] 通信パラメータ 419
 LDAP 通信パラメータ 425
 サーバのポート番号 433
 サーバ設定 417
 テスト 361
 トラブルシューティング 360
 ファイアウォール経由の接続 419
 TDS 通信パラメータ 435
 Telnet
 TCP/IP のテスト 361
 TEMP_DATA_IN_SHARED_TEMP
 論理サーバポリシーのオプション 454
 TEXT インデックス 181
 作成 181
 TIME インデックス 181
 推奨される使用法 181
 制限 182
 短所 182
 長所 182
 他のインデックスとの比較 174
 追加インデックス 174
 TIMEOUT 通信パラメータ 435
 TLV 239
 TO 通信パラメータ 435
 Transact-SQL
 COMMIT TRANSACTION 478

U

UID 接続パラメータ 416
 UNC 接続パラメータ 416
 Unconditional 接続パラメータ 416
 UNIQUE 制約 207
 USER 特別値
 デフォルト 204
 Userid 接続パラメータ 416
 Utilities 文 537

V

VERIFY 通信パラメータ 436

索引

VERIFY_PASSWORD_FUNCTION オプション
442

W

WarehouseArchitect

説明 145

WD インデックス 167

CHAR カラム 495

デリミタ 495

推奨される使用法 167

短所 168

長所 168

WHERE 句

インデックス選択への影響 184

WORD インデックス 167

Z

zr ログファイル 375

あ

暗号化アルゴリズム

CREATE DATABASE 文 480

い

イーサネット 362

一意性

制約 511

イベント

DBSpaceLogger 350

イベント名の取得 324

スケジュール名の取得 324

ディスク領域の使用状況のモニタリング
350

領域の使用状況のモニタリング 350

削除 531

インタフェースライブラリ

接続 61

インデックス

NBit 174

TEXT インデックス 181

インデックスタイプの選択 186

作成 150, 190, 265, 495

システムビュー 194

情報の表示 193

所有者 495

推奨される組み合わせ 189

ディスク領域の使用率 187

テーブルでの使用 495

テーブルのロード後の追加 191

テーブル上の多すぎるインデックス 354

デフォルト 174

トークン化 FP (NBit) 174

名前変更 193

フラット FP 174

並列作成 192

マルチカラム 495

マルチカラム HG および NULL 値 177,
495

命名 495

メッセージログ内のアドバイス 192

ユニーク 495

ユニークな値の最大数 355

リスト表示 193

高速射影 174

再構築 155

削除 194, 531

自動作成 154

説明 165

並行作成 191

インデックスアドバイス 192

インデックスタイプ

LF 179

推奨 166, 185

選択 186

選択条件 182

え

エラー

DB 領域が不足した状況 346

エラー 241

トランザクション処理 241

円記号

Windows のローデバイス 90

エンティティ整合性

確保 211

お

- オプション
 - DEDICATED_TASK 440
 - MAX_TEMP_SPACE_PER_CONNECTION 95
 - QUERY_TEMP_SPACE_LIMIT 95
 - カーソル 439
 - スコープ 439
 - 継続期間 439
 - 設定 547
 - 優先度 439
- オプション値
 - トランケーション 547
- オフライン
 - DB 領域 444
- オンライン
 - DB 領域 444

か

- カーソル
 - スクロール 262
 - データベースオプション 439
 - トランザクション 261
 - プロシージャ 268
 - ホールド 263
 - メッセージロギング 264
 - 位置付け操作 263
 - 依存性 262
- 改行
 - WD インデックスのデリミタ 495
- 外部キー
 - オプション 215
 - 強制力のない既存の 215
 - 整合性制約 511
 - 名前のない 511
- 外部キー制約
 - 削除 194
- 書き込みを意図したロック 252
 - 表示 247, 251
- カタログストア
 - インデックスの検証 220
 - キャッシュサイズの設定 20
 - 説明 102
 - 領域の事前割り付け 102

- カタログファイル
 - サイズの増加 348

- カラム
 - 削除 156
 - 制約 511
 - 追加 156
 - デフォルト 200
 - 変更 156, 459
- カラムのデフォルトの使用 200
- カラムのデフォルトの変更と削除 201

き

- キャッシュ DB 領域 135
 - DB ファイルの読み書きステータスへの変更 139
 - DB ファイルの読み込み専用への変更 139
 - 再有効化 139
 - 削除 137
 - 作成 136
 - ストレージの削除 138
 - ストレージの追加 137
 - 無効 139
 - モニタリング 140
- キャッシュサイズ
 - カタログストアの設定 20
 - カタログストア用の設定 20
- 強制力のない外部キー 215
- 強力な暗号化
 - CREATE DATABASE 文 480

く

- クエリ
 - DATEPART を使用 170
 - スラッシング 354
 - 同時ユーザの制限 17
 - パフォーマンス 177
 - パフォーマンスの問題 354
 - 同時実行性の規則 241
 - 範囲述部 172
- クエリタイプ
 - インデックスタイプ 184
- クエリプラン
 - インデックスの推奨事項 192

索引

組み込みデータベース 385
 Java 52
 起動 52
クライアントプロセス情報 386
クライアント側
 Encryption [ENC] 接続パラメータ 402
グローバルテンポラリテーブル 253
 説明 152

こ

コマンドラインスイッチ 6
 必須 8
 表示 6

さ

サーバ
 CPU 使用率 339
 応答しない 339
 停止 40
 デッドロック 339
 トランザクションログ 335
 ユニークなポート番号 336, 337
 ユニークな名前 336
 リカバリ 333
 領域不足 339
 再利用 39
 処理の停止 339
 接続 67
 停止の問題 343
 無応答 339
 名前の競合 336
 命名 9
 論理サーバの作成 504
 論理サーバの削除 535
 次も参照： データベースサーバ
サーバの再利用 39
サーバログファイル 36
削除
 カラムのデフォルト 201
 ビュー 147, 531
サブクエリ
 スカラ値 147
サブトランザクション
 セーブポイント 257

サポートセンタ
 MySybase 382
 オンラインヘルプ 382
 チェックリスト 383
 問題の報告 378
サポートセンタ用のチェックリスト 383

し

システム DB 領域 100
システムテーブル
 説明 158
システムの無応答 339
システムビュー
 整合性 220
システムプロシージャ
 sp_iqtransaction 222
システム障害
 障害からのリカバリ 342
ジョイン
 パフォーマンス 194
 パフォーマンスの影響 194
シンボリックリンク 480

す

スカラ値サブクエリ 147
スキーマ
 作成 150
 変更 150
スクロール
 カーソル 262
スケジュール 315
スタックトレース
 スレッドに対する生成 339
 ロケーション 342
ストアドプロシージャ
 情報の表示 267
スナップショットバージョン管理 221
 テーブルレベル 239, 240
 ローレベル 246
スラッシング
 HASH_THRASHING_PERCENT オプション 354
スレッド
 スタックトレースの生成 339

増加 18
不足 351

せ

制約

パフォーマンスへの影響 198

セーブポイント

トランザクション内 257

セキュリティ

Encryption [ENC] 接続パラメータ 402

パスワードの最小の長さ 441

監査 331

統合化ログイン 75, 76

接続

BroadcastListener [BLISTENER] 通信パラメータ 419

DEDICATED_TASK オプション 440

iAnywhere JDBC ドライバ 67

Interactive SQL 49

jConnect 67

JDBC 60

JDBC SAP Sybase Open Client 60

LDAP 通信パラメータ 425

概要 61

データソースの使用 57

テンポラリ 80

同時ユーザの制限 15

ファイアウォール 419

リモート 67, 264

ローカルデータベース 49

ロードマップ 47

ロギング 83, 440

確立 62

埋め込みデータベース 52

例 48

設定パラメータ

上書き 338

セパレータ

WD インデックス 495

セマフォ 353

そ

相対パス 480

た

タブ

WD インデックスのデリミタ 495

ち

チェックポイント

システムリカバリ 260

自動、明示的 257

間隔の調整 358

説明 257

直接接続ストレージ 135

て

停止

サーバ 40

データベース 45

トラブルシューティング 343

ディスク

領域不足 339

領域の使用状況のモニタリング 350

領域の不足 346

ディスク領域不足

推奨される対応策 339

ディスク領域

インデックス 187

割り付け 124

節約 262

ディスク領域の不足

対処法 346

領域の使用状況のモニタリング 350

データ

重複 197

無効 197

データ型

テーブル作成での指定 150

ユーザ定義の削除 531

作成 493

データソース

jConnect の使用 67

接続 57

データのロード

エラー 355

カラムのデフォルト 202

パフォーマンス 354

通知メッセージ 366

同時実行性の規則 241

領域の使用状況のモニタリング 350

索引

- データベース
 - SQL Anywhere 102
 - アンロード 43
 - オブジェクトの作成 150
 - オブジェクトの使用 145
 - 組み込み 385
 - 権限 24
 - サイズ 92
 - 初期サイズ 106
 - スキーマ作成 150
 - デフォルトの特性 88
 - ファイルロケーション 87
 - ブロックサイズ 95
 - ページサイズ 92
 - 領域の事前割り付け 99
 - ロケーションの選択 90
 - 移動 87
 - 起動 43
 - 起動パーミッション 45
 - 起動時の命名 10
 - 作成 85, 480
 - 作成に必要な権限 2
 - 削除 96
 - 修復 333
 - 初期化 85
 - 接続 61, 67
 - 設計 145
 - 設定の概要 1
 - 相対パス名 90
 - 停止 45, 549
 - 埋め込み 389
 - 名前の競合 336
 - 命名 9
- データベースオプション
 - DEDICATED_TASK 440
 - カーソル 439
 - 最大文字列長 547
 - 継続期間 439
- データベースサーバ
 - Windows サービス 6
 - 起動 3
 - コマンドプロンプトからの起動 4
 - コマンドラインスイッチ 6
 - リモート 67
 - 起動の回避 388
 - 起動時の命名 10
 - 接続 67
 - 停止 39, 43, 416, 551
 - 名前スイッチ 9
- データベースの停止 549
- データベースファイル
 - 変更 444
- データベースユーティリティ 59
 - ログの変換 327
- データリンクレイヤ
 - トラブルシューティング 360
- データ整合性
 - カラムのデフォルト 200
 - システムテーブル内のルール 220
 - 概要 197
 - 制約 199
- データ定義
 - 作成 150
- データ定義言語
 - 説明 145
 - 同時実行性の規則 242
- テーブル
 - GLOBAL TEMPORARY 511
 - キーの追加 157
 - 定義の変更 459
 - テンポラリ 511
 - ブロックされたアクセス 356
 - ブロックされたアクセスの管理 356
 - ロック 240, 541
 - 作成 150, 511
 - 削除 156, 531
 - 読み取り専用 127
 - 変更 156, 459
- テーブル制約 511
- テーブルパーティション
 - DDL 操作 159
 - DML 操作 159
 - hash-partition-key 162
 - hash-partitioning-scheme 162
 - パーティションキー 159
 - ハッシュパーティション 162
 - ハッシュ範囲パーティション 163
 - 範囲パーティション 161
 - 定義 158

- テーブルレベルのスナップショットバージョン管理 239
 - テーブルロック
 - 表示 247
 - テーブルロック競合
 - 管理 254
 - デッドロック 232
 - 解決 339
 - 検出 339
 - レポート 233
 - デバイスタイプ
 - データベース 2
 - デフォルト
 - AUTOINCREMENT 205
 - NEWID 205
 - NULL 206
 - USER 特別値 204
 - カラム 200
 - ロード 202
 - 現在の日付と時刻 203
 - 作成 201
 - 接続パラメータ 58
 - 挿入 202
 - 定数式 206
 - 文字列と数値 206
 - デフォルトインデックス
 - NBit 174
 - トークン化 FP (NBit) 174
 - フラット FP 174
 - デフォルトの設定ファイル 7
 - デリミタ
 - 例 495
 - テンポラリ DB 領域ファイル 363
 - テンポラリストア
 - 領域管理 95
 - テンポラリテーブル 511
 - 説明 152
 - バージョン管理 253
 - ロード 152
 - 作成 511
 - テンポラリ記憶領域
 - 領域を節約するオプション 262
- と**
- トークン化 FP (NBit) 174
- ドメイン 493
 - ドライバ
 - jConnect JDBC ドライバ 67
 - JDBC 60
 - SQL Anywhere JDBC ドライバ 67
 - ドライバの選択
 - iAnywhere JDBC ドライバの使用 67
 - jConnect JDBC ドライバの使用 67
 - トラブルシューティング 333
 - dbisql 345
 - サーバの運用 334
 - データベース接続 344
 - プロトコル 359
 - リソースの問題 346
 - 一般的な問題 362
 - 処理の問題 354
 - 配線の問題 362
 - トランザクション 221
 - ROLLBACK 文 545
 - SAVEPOINT 文 546
 - カーソル 261
 - 開始 221
 - コミット 478
 - サブトランザクションとセーブポイント 257
 - セーブポイント 257
 - 説明 221
 - ブロッキング 232
 - リカバリ 260
 - リモートデータアクセス 264
 - ロールバックス 259
 - 管理 264
 - トランザクション処理
 - 説明 221
 - トランザクションログ
 - システムリカバリ 260
 - トランザクション管理 264, 478
 - Transact-SQL 478
 - トレース
 - スレッドに対する生成 339
- ね**
- ネットワークアダプタ
 - ドライバ 359

索引

- ネットワークプロトコル
 - トラブルシューティング 359
- ネットワーク通信
 - コマンドラインスイッチ 417
- ネットワーク番号
 - IPX アドレス 421

は

- バージョン
 - 独立性レベル 227
- バージョン管理 221, 239, 246
 - カーソル 262
 - システムリカバリ 260
 - テンポラリテーブル 253
 - パフォーマンスの影響 244
- パーティション
 - 削除 459
 - 読み取り専用 127
- パーティションの削除 459
- パーティションの制限 14
- パーミッション
 - ビュー 149
 - ローデバイス 87
- ハイパースレディング
 - サーバスイッチ 21
- バグ
 - レポート 378
- パス
 - 相対 480
- パスワード
 - 最小の長さ 441
- パスワードのセキュリティ 89
- パス名
 - データベース 90
- バックアップ
 - 同時実行性 261
- ハッシュパーティション
 - hash-partition-key 162
 - hash-partitioning-scheme 162
 - 制限事項 162
 - 例 162
- ハッシュ範囲パーティション
 - hash-range-partitioning-scheme 163
 - SUBPARTITION BY RANGE 163
 - 例 163

- バッチ
 - 説明 265
- バッファキャッシュ
 - IQ UTILITIES コマンド 339
 - パーティショニング 13, 14
 - モニタ 339
 - 領域不足 339, 353
- バッファの不足
 - バッファキャッシュ 353
- バッファ領域
 - 接続パラメータ 392
- パフォーマンス
 - クエリとロード 354
 - ジョイン 194
 - 制約の影響 198
 - テーブルレベルのバージョン管理が与える影響 244
- 範囲パーティション
 - partition-key 161
 - range-partition-decl 161
 - range-partitioning-scheme 161
 - VALUES 句 161
 - 制限事項 161
 - 範囲パーティション 161
 - 例 161

ひ

- ビュー 147
 - インデックス 495
 - 永久テーブルとの相違点 146
 - 削除 147, 149, 531
 - 挿入と削除 147
 - パーミッション 149
 - 変更されたテーブル 459
 - 作成 146
 - 使用 147
 - 修正 148
 - 編集 146

ふ

- ファイアウォール
 - BroadcastListener [BLISTENER] 通信パラメータ 419
 - LDAP 通信パラメータ 425

- 接続 419
 - ファイル
 - DB 領域 444
 - オフラインに設定 444
 - オンラインに設定 444
 - ファイルサイズ
 - 制御 348
 - プライマリキー
 - AUTOINCREMENT 205
 - マルチカラム 212
 - 作成 157
 - フラット FP 174
 - フレームタイプ 362
 - プロシージャ
 - カーソル 268
 - 実行権限 266
 - 所有者 267, 268
 - パラメータ 268
 - 結果セット 267
 - 結果を返す 267, 292
 - 呼び出し 266
 - 構造 289
 - 削除 266, 531
 - 使用可能な SQL 文 289
 - 説明 265
 - ブロッキング 232
 - ブロックサイズ 95
 - ブロックされた書き込みアクセス
 - ブロックしている書き込みの確認 356
 - 競合の管理 356
 - プロトコル
 - スイッチ 31
 - トラブルシューティング 359
 - 分割
 - DDL 操作 159
 - DML 操作 159
 - hash-partition-key 162
 - hash-partitioning-scheme 162
 - hash-range-partitioning-scheme 163
 - partition-key 161
 - range-partitioning-scheme 161
 - SUBPARTITION BY RANGE 163
 - VALUES 句 (範囲パーティション) 161
 - 制限事項 161, 162
 - テーブル 158
 - パーティションキー 159
 - ハッシュパーティション 162
 - ハッシュ範囲パーティション 163
 - 範囲パーティション 161
 - 例 161-163
 - 分散クエリ処理
 - パフォーマンス 454, 507
- へ
- 並列 CREATE INDEX 192
 - ページサイズ 92
 - カタログ 29
 - スイッチ 29
 - ベーステーブル 152
- ほ
- ポート番号
 - データベースサーバ 433
 - ホールドカーソル 263
- ま
- マテリアライズドビュー
 - 削除 531
 - マルチカラムインデックス 495
 - マルチプレックスサーバ
 - ロードバランス 436
 - マルチプレックスデータベース
 - 作成 480
 - マルチプロセッサマシン
 - スイッチ 11
- め
- メインキャッシュ
 - 削除 531
 - メインストア
 - 領域管理 95
 - メタデータ
 - カタログストア内 102
 - メッセージ
 - DB 領域が不足した状況 346
 - メモリの通知 367
 - 削除 531

索引

メッセージファイル

接続情報 381

メッセージログ

IQ_SYSTEM_MSG DB 領域 105

インデックスの推奨事項 192

メモリ

カタログストアのキャッシュ用 20

連結メモリのプールの作成 17

メモリメッセージ

ロード通知メッセージ 367

も

文字列

データベースオプションの長さ 547

モニタ

IQ UTILITIES 文でのモニタ 537

開始と停止 537

モニタ出力オプション

-contention 255

ゆ

ユーザ定義データ型

CHECK 条件 209

CREATE DOMAIN 文 493

削除 531

ユーザ定義関数

実行権限 282

呼び出し 281

作成 274

使用 274

ユニークインデックス 495

り

リカバリ

サーバ 333

システム 260

システム障害 342

トランザクション 260

トランザクションログ 260

リソース計画

iqnumbercpus スイッチ 21

リダイレクト 409

リモートサーバ

トランザクション管理 264

リモートデータアクセス 449

リンク

シンボリック 480

る

ルータ

ブロードキャスト 421

ルート論理サーバポリシー 454

れ

連結メモリ

iqwmem スイッチの設定 17

ろ

ローカルテンポラリテーブル 253

説明 152

ローカルな書き込みを意図 252

ローデバイス 2

DB 領域の追加 87

Windows でのアクセスの設定 109

Windows でのシンボリックリンク 113

Windows でのパーミッションの設定 111

Windows でのパーミッションの表示 110

Windows での命名規則 90

データベース作成 87

パーミッション 87

命名 480

ロード

スケラビリティ 14

ロードスキーマ

推奨されるデータベースサイズ 106

ロードパフォーマンス

iqpartition サーバオプション 13

テーブルロック競合 254

ロールバック 221

DB 領域の不足状況 95, 346

ローレベルのスナップショットバージョン管理 246

ローロック 251

ログイン

統合化 71, 73

次も参照: 接続

ログインリダイレクション 454

ログファイル
サーバ 36
接続情報の関連付け 381
ロック
DDL 操作 242

ROLLBACK による解放 545
テーブル 240, 541
競合の管理 356
ロック競合
iqpartition サーバオプション 13

