

SYBASE®

システム管理ガイド：第 1 巻

Sybase IQ

15.2

ドキュメント ID : DC00284-01-1520-01

改訂 : 2010 年 4 月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいエディションまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は、予告なく変更されることがありますが、このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイベース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時にのみ提供されます。このマニュアルの内容を Sybase, Inc. の書面による事前の許可なく複製、転載、翻訳することは、電子的、機械的、手作業、光学的、その他、形態や手段を問わず禁じられています。

Sybase の商標は、**Sybase trademarks ページ** (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase および表記されている商標は、Sybase, Inc の商標です。® は、米国で登録されていることを示します。

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

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

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

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

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

目次

はじめに	xvii
------------	------

第 1 章	Sybase IQ システム管理の概要.....	1
	Sybase IQ の概要.....	1
	システム管理のツール.....	3
	データベース・サーバ.....	4
	設定可能なテーブル領域.....	4
	DB 領域、dbfile、ストア.....	4
	SQL Anywhere と Sybase IQ.....	5
	セキュリティの概要.....	5
	ユーザのタイプ.....	5
	パーミッションの付与.....	6
	同時処理.....	6
	マルチプレックス機能.....	6
	ストアド・プロシージャ.....	7
	Sybase IQ ストアド・プロシージャ.....	7
	カタログ・ストアド・プロシージャ.....	7
	システム・テーブルとシステム・ビュー.....	8
	コマンドと関数.....	8
	SQL 文のタイプ.....	9
	関数.....	9
	メッセージ・ロギング.....	9
	メッセージ・ログ内のバージョン情報.....	11
	メッセージ・ログ内の照合情報.....	11
	メッセージ・ログ内の接続情報.....	11
	メッセージ・ログ内の挿入通知.....	12
	メッセージ・ログ・ファイルの管理.....	12
	ユーティリティ・データベース.....	16
	大容量データベースの管理.....	17
	メモリ使用の管理.....	17
	データ・ロードの管理.....	17
	処理スレッドの管理.....	18
	ディスク領域の管理.....	18
	中間バージョン.....	19
	データベースの作成.....	19
	インデックスの作成.....	19

	クエリの最適化	20
	スキーマ設計	20
	UNION ALL ビュー	20
第 2 章	Sybase IQ の実行	21
	データベース・サーバの起動.....	21
	起動ユーティリティによるサーバの起動.....	22
	Sybase Central からのサーバの起動.....	24
	Windows の [スタート] メニューからのサーバの起動	26
	Windows ホストが再起動する場合のサーバの再起動.....	27
	Windows サービスとしてのサーバの実行	27
	コマンド・ライン・スイッチの使用.....	27
	サーバとデータベースの命名.....	29
	コマンド・ラインからのパフォーマンスとメモリの制御.....	31
	コマンド・ラインからのパーミッションの制御.....	35
	最大カタログ・ページ・サイズの設定	37
	クライアント／サーバ環境の設定.....	37
	強制リカバリ・モードでサーバを起動する	39
	dbisql からのサーバの起動.....	40
	同じマシンでの複数のサーバまたはクライアントの起動.....	40
	サーバ・アクティビティのモニタ	41
	データベース・サーバの停止.....	44
	サーバを停止および再起動する状況	44
	サーバを停止する方法.....	45
	サーバを停止できるユーザ	47
	オペレーティング・システム・セッションの停止	48
	データベースの開始と停止	48
	データベースの起動に関するガイドライン	49
	データベースの停止	50
	iqdemo データベースの起動	50
	Sybase Central の開始と停止	50
第 3 章	Sybase IQ の接続	51
	接続の概要	52
	接続の確立方法	52
	接続パラメータの構造.....	54
	接続文字列として渡される接続パラメータ	54
	ODBC データ・ソースへの接続パラメータの保存.....	55
	Sybase Central または Interactive SQL からの接続	55
	[接続] ダイアログの表示.....	56
	接続用のドライバの指定	57
	[接続] ダイアログ・ボックスの使用.....	58
	Sybase Central での接続を簡略化する方法.....	59
	サーバ・オブジェクトの作成.....	59
	接続プロファイルの作成	59

簡単な接続の例	60
Sybase Central または Interactive SQL から	
デモ・データベースへの接続	60
Sybase Central または Sybase IQ から	
同じマシン上のデータベースに接続する	61
Interactive SQL から他のデータベースへの接続	62
コマンド・ライン・ユーティリティを使用して接続する	63
埋め込みデータベースへの接続	65
データ・ソースを使用した接続	67
デフォルト接続パラメータの使用	68
Sybase IQ ユーティリティからの接続	69
ODBC データ・ソース	69
データ・ソースの保存場所	70
ODBC データ・ソースの作成と編集	70
ODBC データ・ソースの設定	73
ファイル・データ・ソースの使用	78
UNIX での ODBC データ・ソースの使用	79
iAnywhere Solutions Oracle ドライバの使用	80
Windows の設定	82
UNIX 設定	82
IQDSN 設定	83
OLE DB を使用してデータベースに接続する	83
OLE DB プロバイダ	84
ADO からの接続	85
接続パラメータ	86
接続パラメータについてのヒント	86
Sybase IQ の接続の確立方法	87
接続を確立する手順	87
インタフェース・ライブラリの検出	88
接続パラメータ・リストのアセンブル	89
サーバの検出	90
データベースの検出	92
Interactive SQL 接続	93
他のデータベースからの接続	93
UNIX システムでのポート番号の競合防止	94
サーバを検出できるかどうかのテスト	94
統合化ログインの使用	95
統合化ログインの使用方法	96
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	99
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	100
ネットワークから見た統合化ログイン	101
デフォルトの統合化ログイン・ユーザの作成	101
接続の切断と削除	102
接続ログ	102
起動のトラブルシューティング、シャットダウン、接続	103
Sybase IQ を起動できない	103

	データベースに接続できない.....	104
	緊急時のデータベース・サーバの停止 (UNIX)	106
	UNIX kill コマンドによる Interactive SQL の停止	106
第 4 章	接続パラメータと通信パラメータ	109
	接続パラメータ	109
	AppInfo 接続パラメータ (App)	110
	AutoPreCommit 接続パラメータ (AutoPreCommit)	112
	AutoStart 接続パラメータ (Astart).....	113
	AutoStop 接続パラメータ (Astop).....	113
	CharSet 接続パラメータ (CS).....	114
	CompressionThreshold 接続パラメータ (COMPTH)	114
	CommBufferSize 接続パラメータ (CBSize).....	114
	CommLinks 接続パラメータ (Links).....	115
	ConnectionName 接続パラメータ (CON)	117
	DatabaseFile 接続パラメータ (DBF).....	117
	DatabaseName 接続パラメータ (DBN)	118
	DatabaseSwitches 接続パラメータ (DBS).....	119
	DataSourceName 接続パラメータ (DSN).....	120
	DBKEY 接続パラメータ (DBKEY).....	120
	DisableMultiRowFetch 接続パラメータ (DMRF).....	121
	EngineName 接続パラメータ (ENG)	121
	EncryptedPassword 接続パラメータ (ENP).....	122
	Encryption 接続パラメータ (ENC)	122
	FileDataSourceName 接続パラメータ (FileDSN)	124
	Idle 接続パラメータ (IDLE)	125
	Integrated 接続パラメータ (INT).....	126
	Language 接続パラメータ (LANG)	126
	LazyClose 接続パラメータ (LCLOSE).....	127
	LivenessTimeout 接続パラメータ (LTO).....	127
	LogFile 接続パラメータ (LOG).....	128
	NewPassword 接続パラメータ (NEWPWD)	129
	Password 接続パラメータ (PWD).....	129
	PrefetchBuffer 接続パラメータ (PBUF)	129
	PrefetchRows 接続パラメータ (PROWS).....	130
	Prompt 接続パラメータ (PROMPT).....	131
	RetryConnectionTimeout 接続パラメータ (RetryConnTO).....	131
	ServerName 接続パラメータ (ENG)	131
	StartLine 接続パラメータ (START).....	131
	Unconditional 接続パラメータ (UNC)	132
	Userid 接続パラメータ (UID)	132
	ネットワーク通信のパラメータ	132
	Broadcast 通信パラメータ (BCAST).....	133
	BroadcastListener 通信パラメータ (BLISTENER)	134
	Certificate 通信パラメータ	134
	Certificate_Password 通信パラメータ	135

ClientPort パラメータ (CPort).....	135
DatabaseName 通信パラメータ (DBN).....	136
DoBroadcast パラメータ (DBROAD).....	137
Host パラメータ (IP).....	138
LDAP 通信パラメータ (LDAP).....	139
LocalOnly 通信パラメータ (LOCAL).....	139
Logfile 接続パラメータ (LOG).....	140
LogFormat 通信パラメータ (LF).....	140
LogMaxSize 通信パラメータ (LSIZE).....	141
LogOptions 通信パラメータ (LOPT).....	141
MaxConnections 通信パラメータ (MAXCONN).....	142
MaxRequestSize 通信パラメータ (MAXSIZE).....	143
MyIP 通信パラメータ (ME).....	143
PreFetchOnOpen 通信パラメータ.....	144
ReceiveBufferSize 通信パラメータ (RCVBUFSZ).....	144
SendBufferSize 通信パラメータ (SNDBUFSZ).....	145
ServerPort パラメータ (PORT).....	145
Sessions パラメータ.....	146
TDS パラメータ.....	147
Timeout パラメータ (TO).....	147
VerifyServerName 通信パラメータ (Verify).....	148
第 5 章	
データベース・オブジェクトの使用.....	151
Sybase IQ データベースの作成.....	151
データベースの設計.....	151
データベース・オブジェクトの編集ツール.....	152
データベース設定手順の概要.....	153
データ定義権限の拡張.....	155
デバイス・タイプの選択.....	155
データベース用の領域の割り付け.....	156
範囲分割.....	164
データベース・オブジェクトの使用.....	168
SQL によるデータベースの作成.....	168
IQ メイン・ストアと IQ テンポラリ・ストアの領域管理.....	178
データベース・オプションの設定.....	178
データベース内のシステム・オブジェクトの表示.....	179
データベースからの切断.....	180
データベースの削除.....	181
DB 領域の使用.....	181
dbfile の属性と操作.....	181
DB 領域の属性と操作.....	182
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.....	182
DB 領域の命名.....	188
DB 領域の追加.....	188
DB 領域の削除.....	190
DB 領域管理の例.....	191

テーブルの編集	196
テーブルの作成	197
テーブルの変更	202
テーブルの削除	203
プライマリ・キーと外部キーの作成	204
システム・ビューに含まれるテーブル情報	205
ビューの編集	205
ビューの作成	206
ビューの使用	207
ビューの修正	208
ビューに対するパーミッション	208
ビューの削除	209
システム・ビューに含まれるビュー情報	209
インデックスの編集	209
インデックスの概要	210
インデックスの作成	210
システム・ビューに含まれるインデックス情報	211
インデックスの検証	211
インデックスの名前変更	212
インデックスの削除	212
第 6 章	
Sybase IQ インデックスの使用	213
インデックスの概要	213
Sybase IQ インデックス・タイプ	214
従来のインデックスに勝る利点	217
Sybase IQ インデックスの作成	218
CREATE INDEX 文	218
Sybase Central を使用したインデックスの作成	219
TEXT インデックスの作成	219
複数のインデックスの同時作成	219
インデックス・タイプの選択	219
インデックス内のユニークな値の数	220
クエリのタイプ	220
インデックス条件：ディスク領域の使用率	222
インデックス内のデータ型	223
インデックス・タイプの組み合わせ	224
Sybase IQ インデックス・タイプ	225
高速射影 (FP) デフォルト・インデックス・タイプ	225
Low_Fast (LF) インデックス・タイプ	229
High_Group (HG) インデックス・タイプ	230
High_Non_Group (HNG) インデックス・タイプ	232
比較 (CMP) インデックス・タイプ	234
包含 (WD) インデックス・タイプ	235
日付 (DATE)、時刻 (TIME)、日時 (DTTM) インデックス・タイプ	237
TEXT インデックス	242
アドホック・ジョインのパフォーマンスの最適化	242
インデックスの選択	243

データ挿入後のカラム・インデックスの追加	244
ジョイン・インデックスの使用	245
ジョイン・インデックスによって向上するクエリのパフォーマンス	245
ジョイン・インデックスのロード時の考慮事項	245
クエリでのジョイン・インデックスの使用法	245
ジョイン・インデックス間の関係	245
ジョインがアドホックになるとき	246
ジョイン階層の概要	246
ジョイン・インデックス内のカラム	247
クエリ解析におけるジョイン階層	247
複数テーブルのジョインとパフォーマンス	249
ジョイン・インデックスの作成手順	250
ジョイン・インデックスの同期	252
テーブル間のジョイン関係の定義	252
CREATE JOIN INDEX 文の実行	255
Sybas Central でのジョイン・インデックスの作成	256
ジョイン階層のタイプ	256
ジョイン・インデックスに含まれるテーブルの修正	260
ジョイン・インデックス内のテーブルでの挿入と削除	261
テーブルのバージョン管理によるジョイン・インデックスへの アクセス制御	261
ジョイン・インデックスのサイズとメリットの見積もり	262
sp_iqestjoin によるジョイン・インデックス・サイズの見積もり	262
ロー数を比較したジョイン・インデックスのメリットの見積もり	263
第 7 章 データベースへのデータの入出力	265
インポートとエクスポートの概要	265
インポートとエクスポートの方法	266
データの入出力フォーマット	267
データ修正のパーミッション	268
データベース更新のスケジュール	268
データベースからのデータのエクスポート	269
出力先リダイレクションの使用	269
データ抽出オプション	270
LOAD TABLE 文を使ったデータのバルク・ロード	279
クライアントからの直接のデータ・ロード	286
分割されたテーブルのロード	286
メッセージ・ロギングの制御	287
整合性制約違反のロギング	288
BINARY ロード形式の使用	291
IQ の BINARY ロード形式とロード効率	292
オペレーティング・システムのネイティブ・データ型	292
DATE	293
TIME	293
TIMESTAMP	293
NUMERIC と DECIMAL	294
NULL の挿入	295

INSERT 文の使用	298
指定した値のロー単位の挿入	298
データベースからの指定したローの挿入	299
別のデータベースからの挿入	300
データの対話的インポート	302
エンディアン・フォーマットが異なるシステムへのデータの移動	303
ジョイン・インデックスのテーブルへの挿入	305
プライマリ・キー・カラムと外部キー・カラム	306
ラージ・オブジェクト・データの抽出	306
挿入時のデータ変換	306
15 より前のバージョンの Sybase IQ からのデータの挿入	308
ロード変換オプション	308
データ表示に関連する他の項目	321
Adaptive Server Enterprise のデータ型の対応	321
データのバルク・ロードのチューニング	326
UPDATE によるデータの更新	330
データベースの削除	331
第 8 章 ユーザ ID とパーミッションの管理	333
データベースのパーミッションと権限の概要	333
権限の継承	334
パーミッションの継承	335
権限の概要	335
所有権パーミッションの概要	339
DB 領域管理パーミッション	340
テーブルとビューのパーミッションの概要	340
グループ・パーミッションの概要	340
サーバのコマンド・ラインのパーミッション・オプション	341
ログイン管理	341
ユーザ・アカウントと接続の管理	342
ログイン要求失敗後の接続の阻止	344
ユーザのロックアウト	344
ユーザのロックアウト解除	344
ユーティリティ・データベース・サーバのセキュリティ	345
ユーティリティ・データベースのパスワードの定義	346
ファイル管理文を実行するためのパーミッション	346
個別のユーザ ID とパーミッションの管理	347
新規ユーザの作成	348
パスワードの変更	349
権限の付与	350
テーブルおよびビューに対するパーミッションの付与	351
パーミッションを付与する権利をユーザに付与する	353
プロシージャに対するパーミッションの付与	354
ユーザ・パーミッションの取り消し	355

グループの管理.....	356
グループの作成.....	356
ユーザにグループのメンバシップを付与する.....	357
グループのパーミッション.....	358
グループが所有するテーブルの参照.....	358
パスワードのないグループ.....	359
特殊なグループ.....	359
データベース・オブジェクトの名前とプレフィクス.....	360
高度なセキュリティを実現するためのビューとプロシージャの使い方.....	362
ビューを使用したセキュリティの調整.....	362
プロシージャを使用したセキュリティの調整.....	364
プロシージャを使用して接続を無効にする.....	366
ユーザ・パーミッションの評価方法.....	366
リソース接続使用の管理.....	367
ユーザ・リソースを制御するデータベース・オプション.....	367
ユーザ・リソースに影響するその他の設定.....	368
システム・テーブルのユーザとパーミッション.....	368
トランスポート・レイヤ・セキュリティ.....	370
IPv6 のサポート.....	370
第 9 章	データ整合性の確保..... 371
データ整合性の概要.....	371
データが無効になるとき.....	371
データの有効性の確保.....	372
データベース内容の変更.....	373
データ整合性ツール.....	373
整合性制約を実装するための SQL 文.....	375
カラムのデフォルトの使用.....	375
カラムのデフォルトの作成.....	376
カラムのデフォルトの変更と削除.....	377
カラムのデフォルト値の操作.....	377
Sybase Central におけるカラムのデフォルトの編集.....	379
日付、時刻、タイムスタンプのデフォルト.....	379
USER デフォルト.....	380
IDENTITY または AUTOINCREMENT のデフォルト.....	381
NEWID デフォルト.....	381
NULL デフォルト.....	382
文字列と数値のデフォルト.....	382
定数式のデフォルト.....	382
テーブル制約とカラム制約の使用.....	383
カラムまたはテーブルに対する UNIQUE 制約の使用.....	383
カラムに対する IQ UNIQUE 制約の使用.....	383
カラムに対する CHECK 条件の使用.....	384
ユーザ定義データ型の CHECK 条件の定義.....	385
Sybase Central におけるカラム制約の編集.....	385
テーブルに対する CHECK 条件の使用.....	386
CHECK 条件の修正と削除.....	386

エンティティ整合性と参照整合性の宣言.....	387
エンティティ整合性の宣言.....	387
エンティティ整合性の確保.....	388
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.....	388
参照整合性の宣言.....	389
参照整合性の損失.....	392
同時処理の制御.....	393
参照整合性チェックの無効化.....	395
システム・テーブルの整合性ルール.....	396
第 10 章	
トランザクションとバージョン管理.....	397
トランザクションとバージョン管理の概要.....	397
トランザクションの概要.....	397
同時実行性の概要.....	400
バージョン管理の概要.....	401
バージョン管理による矛盾の回避.....	408
ロックの仕組み.....	408
DML 操作のロック.....	409
DDL 操作のロック.....	410
プライマリ・キーとロック.....	411
ロックの管理.....	412
アクティブなロックの表示.....	412
ロック競合の管理.....	412
独立性レベル.....	413
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....	415
チェックポイント.....	415
トランザクション内のセーブポイント.....	416
トランザクションのロールバック.....	418
システム・リカバリ.....	418
リカバリにおけるトランザクション情報の役割.....	419
パフォーマンスの影響.....	420
重複したバージョンと削除.....	421
トランザクション内のカーソル.....	422
カーソルとバージョン管理.....	422
カーソルの依存性.....	423
カーソルのスクロール.....	423
ホールド・カーソル.....	423
位置付け操作.....	424
カーソル・コマンド構文と例.....	424
カーソルのログを取るメッセージの制御.....	424

第 11 章	各国語と文字セット	425
	各国語と文字セットの概要	425
	デフォルト照合の使用	427
	ソフトウェアの文字セットについて	427
	Windows でのコード・ページ	427
	照合を使用した文字のソート	428
	ロケールについて	428
	照合について	429
	照合の表示	429
	ANSI か OEM の選択	429
	ANSI 照合についての注意	429
	マルチバイト照合の使用	430
	文字セット変換について	431
	データベース・メッセージの文字変換	431
	接続文字列と文字セット	432
	文字セット変換の回避	433
	各国語と文字セットに関する作業	434
	デフォルトの照合の確認	434
	文字セットの環境設定	434
	ロケール情報の決定	435
	ロケールの設定	435
	名前付き照合を使用するデータベースの作成	437
	文字セット変換を使用するデータベース・サーバの起動	437
	ある照合から別の照合へのデータベースの変更	437
	互換性の問題	437
	パフォーマンスの問題	438
第 12 章	データのバックアップ、リカバリ、アーカイブ	439
	データの保護	439
	データベースのバックアップ	440
	データ・ストアの種類	440
	バックアップの種類	441
	アーカイブ・デバイスの選択	444
	バックアップの準備	446
	バックアップの実行	448
	オペレータの存在についての指定	448
	バックアップの種類指定	449
	仮想バックアップの指定	450
	アーカイブ・デバイスの指定	450
	その他のバックアップ・オプション	453
	テープ・デバイスの待機	454
	読み取り専用のハードウェアを使用したバックアップとリストア	454
	バックアップの例	455
	バックアップ中のエラーからのリカバリ	455
	バックアップ完了後の作業	456
	Sybase 以外の製品によるバックアップの実行	457

仮想バックアップ.....	458
仮想バックアップの種類.....	458
SAN スナップショットまたはシャドウ・ハードウェアを 使用した仮想バックアップ.....	460
システム・レベルのバックアップの実行.....	460
データベースの停止.....	461
適切なファイルのバックアップ.....	461
システム・レベルのバックアップからのリストア.....	463
データベースの検証.....	463
データベースのリストア.....	464
リストア前の作業.....	465
RESTORE 文.....	468
正しい順序のリストア.....	473
リストア後の再接続.....	475
リストア後にトランザクション・ログ名を変更する.....	475
リストア後のデータベースの検証.....	476
排他的な書き込みアクセスが必要なリストア.....	476
ヘッダ情報の表示.....	477
リストア中のエラーからのリカバリ.....	477
データベースのバックアップの検証.....	478
バックアップとシンボリック・リンク (UNIX のみ).....	480
アテンドなしバックアップ.....	481
バックアップとリストアに関する情報の取得.....	482
バックアップ・ログの配置.....	482
バックアップ・ログの内容.....	483
バックアップ・ログの管理.....	483
DB 領域名の記録.....	484
データ・バックアップ方式とリカバリ方式の決定.....	485
日常のバックアップの計画.....	485
バックアップとリストアの責任者の指定.....	486
バックアップとリストアのパフォーマンスの向上.....	487
読み取り専用ハードウェアによるデータのアーカイブ.....	488
読み取り専用ハードウェアの使用.....	489
第 13 章 システムのリカバリとデータベースの修復.....	493
リカバリと修復の概要.....	493
通常のリカバリ.....	494
データベースの検証.....	494
sp_iqcheckdb ストアド・プロシージャ.....	495
sp_iqcheckdb の出力.....	498
sp_iqcheckdb 実行時のリソースの問題.....	501
データベースの修復.....	502
インデックス・エラーの分析.....	502
インデックス・エラーの修復.....	505
割り付けの問題の分析.....	505
割り付けの問題の修復.....	507

強制リカバリ・モード	509
強制リカバリ・モードでのサーバの起動	510
リーク領域のリカバリ	512
マルチプレックス・データベースのリカバリ	513
トランザクション・ログを使わない緊急リカバリ	514
-fリカバリ・オプション	514
DBCC によってレポートされた問題の処理	515
修復できないインデックスの問題	515
矛盾するインデックス、テーブル、またはカラムの削除	516
DBCC エラー・メッセージ	518
第 14 章	
トラブルシューティングのヒント	521
状況別の解決策	521
サーバのリカバリとデータベースの修復	522
サーバの運用上の問題	523
データベース接続の問題	533
Interactive SQL (dbisql) の問題	535
リソースの問題	537
処理の問題	545
パフォーマンスの問題	551
Sybase Central の問題	551
ネットワーク通信のトラブルシューティング	552
互換性のあるプロトコルを使用していることの確認	552
最新のドライバを使用していることの確認	552
リポート時にコンピュータの電源を切る	553
レイヤごとのプロトコル・スタックの診断	553
TCP/IP プロトコル・スタックのテスト	553
配線の問題の診断	555
一般的なネットワーク通信の問題の確認	555
診断ツール	556
sp_iqstatus ストアド・プロシージャ	556
通知メッセージの解釈	558
sp_iqcheckdb ストアド・プロシージャ	562
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	562
現在実行中の文の検索	563
サーバ要求のロギング	563
診断情報を収集するための接続	566
通信の問題の診断	566
サポート・センタへの問題の報告	567
getiqinfo を使った診断情報の収集	567
getiqinfo によって収集される情報	569
接続情報の関連付け	570
その他の情報源	571
チェックリスト：サポート・センタに提出する情報	572
索引	573

はじめに

対象読者

このマニュアルでは、Sybase® IQ の管理の概念と管理に必要な手順について説明します。このマニュアルを使用して、データベースの設定、接続、ロード、セキュリティ、バックアップ、トラブルシューティングについて理解してください。

関連 Sybase IQ マニュアル

Sybase IQ 15.2 のマニュアル・セットの構成は次のとおりです。

- 使用しているプラットフォームの『リリース・ノート』 - マニュアルには記載できなかった最新の情報が記載されています。

リリース・ノートの最新バージョンを利用できる場合があります。製品の CD がリリースされた後で、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの『インストールおよび設定ガイド』 - Sybase IQ のインストール、アップグレード、およびいくつかの設定の手順について説明しています。
- 『新機能の概要 Sybase IQ 15.2』 - 最新バージョンの新機能と動作変更がまとめられています。
- 『Sybase IQ による高度なセキュリティ』 - Sybase IQ データ・レポジトリ内でのユーザによるカラムの暗号化の使用について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
- 『エラー・メッセージ』 - Sybase エラー・コード、SQLCode、SQLState で参照される Sybase IQ エラー・メッセージ、および SQL プリプロセッサのエラーと警告を示します。
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』 - IMSL C 統計ライブラリの時系列 C 関数の簡潔な説明が記載されています。このマニュアルは、RAP - The Trading Edition™ Enterprise のユーザのみが入手できます。
- 『Sybase IQ の概要』 - Sybase IQ や Sybase Central™ データベース管理ツールの操作に慣れていない場合に参照してください。操作の練習ができます。
- 『パフォーマンス&チューニング・ガイド』 - 巨大なデータベースのクエリ最適化、設計、チューニングについて説明しています。

-
- 『クイック・スタート』－ Sybase IQ のソフトウェア・インストールを確定するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う方法が記載されています。デモ・データベースのマルチプレックスへの変換についても説明しています。
 - 『リファレンス・マニュアル』－ Sybase IQ のリファレンス・ガイドです。
 - 『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』－ Sybase IQ でサポートされる SQL、ストアド・プロシージャ、データ型、システム・テーブルについて説明しています。
 - 『リファレンス：文とオプション』－ Sybase IQ でサポートされる SQL 文およびオプションについて説明しています。
 - 『システム管理ガイド』－ 以下で構成されています。
 - 『システム管理ガイド第 1 巻』－ 起動、接続、データベース作成、自動入力とインデックス作成、バージョン管理、照合、システムのバックアップとリカバリ、トラブルシューティング、データベースの修復方法について説明しています。
 - 『システム管理ガイド第 2 巻』－ プロシージャとバッチの作成および実行、OLAP でのプログラミング、リモート・データへのアクセス、Open Server としての IQ の設定について説明しています。このマニュアルでは、スケジューリングとイベント処理、XML でのプログラミング、デバッグについても説明しています。
 - 『時系列ガイド』－ 時系列の予測と分析に使用する SQL 関数について説明しています。この製品オプションを使用するには、RAP - The Trading Edition™ Enterprise が必要です。
 - 『Sybase IQ の非構造化データ分析の概要』－ Sybase IQ データベースでの非構造化データの格納と取得について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
 - 『ユーザ定義関数ガイド』－ ユーザ定義関数、パラメータ、考えられる使用事例が記載されています。
 - 『Sybase IQ マルチプレックスの使用』－ 複数のノードにまたがって発生する大きなクエリの負荷を管理するマルチプレックス機能の使用方法について説明しています。
 - 『ユーティリティ・ガイド』－ Sybase IQ ユーティリティ・プログラムのリファレンス項目 (使用可能な構文、パラメータ、オプションなど) について説明しています。

Sybase IQ 15.2 のマニュアル・セットは、Product Manuals からオンラインで利用できます。(<http://sybooks.sybase.com>)

**関連 SQL Anywhere
マニュアル**

Sybase IQ は SQL Anywhere® パッケージのコンポーネントである SQL Anywhere Server と多くのコンポーネントを共有しているため、Sybase IQ は SQL Anywhere Server と同じ機能を数多くサポートしています。IQ のマニュアル・セットは、SQL Anywhere のマニュアルの該当する箇所を参照しています。

SQL Anywhere には、次のマニュアルがあります。

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

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



Sybase IQ システム管理の概要

この章について

この章では、Sybase IQ について簡単に説明し、IQ システム管理の概要を示します。

内容

トピック名	ページ
Sybase IQ の概要	1
システム管理のツール	3
データベース・サーバ	4
SQL Anywhere と Sybase IQ	5
セキュリティの概要	5
同時処理	6
マルチプレックス機能	6
ストアド・プロシージャ	7
システム・テーブルとシステム・ビュー	8
コマンドと関数	8
メッセージ・ロギング	9
ユーティリティ・データベース	16
大容量データベースの管理	17

Sybase IQ の概要

Sybase IQ は、データ・ウェアハウジング専用設計された、高いパフォーマンスの意思決定支援サーバです。プラットフォームを問わないこの製品は、多くの一般的な Unix、Linux、Windows プラットフォーム上で動作します。

Sybase IQ は、Adaptive Server® ファミリに属します。Adaptive Server ファミリにはこのほかに、エンタープライズ・トランザクションおよび負荷が一様でない環境のための **Adaptive Server Enterprise** と、頻繁には接続を行わないモバイル・コンピューティングの場合によく使用される、小型の Adaptive Server である **SQL Anywhere** があります。

Sybase のデータベース・アーキテクチャ

Sybase のデータベース・アーキテクチャによって、Sybase IQ と SQL Anywhere には共通コード・ベースが提供され、データ・ストアの負荷が最適化されます。データ・ウェアハウジングには、IQ ストアを使用します。これらの製品はコマンド構文とユーザ・インタフェースを共通で使用するため、アプリケーション開発およびユーザのアクセスが容易になります。

複数のデータ・ソースへの高速アクセス

Sybase IQ では、IQ データベースだけでなく、Adaptive Server ファミリの他のデータベース、Sybase 以外のデータベース、フラット・ファイルなど、さまざまなソースのデータを統合できます。このデータを IQ データベースにインポートして、IQ の高速アクセス機能を利用できます。Sybase IQ のリモート・データ・アクセス機能を使用すると、他のデータベースも直接検索できます。

データ・ウェアハウジングと Sybase IQ

「データ・ウェアハウス」とは、ビジネス・アナリストが情報を分析するとき使用するデータの集まりです。通常は、運用データベースとは別のデータベースを使用して、日常の操作が中断しないようにします。データ・ウェアハウスは、意思決定支援システム (DSS) を構築するデータ・ストアとしてよく使われます。「意思決定支援システム」とは、ビジネス上の意思決定を支援するため、企業がデータを分析するとき使用するソフトウェア・アプリケーションです。

Sybase IQ のすべての機能は、DSS アプリケーションを強化するために設計されています。ユニーク・インデックス・システムによって、データ分析が高速化されます。また、クエリの最適化によって、結果に何千または何百万ものデータ・ローが含まれる場合でも、高速な応答が可能になります。複数のクエリ・ユーザのための同時データ・アクセスと、クエリ処理を中断しないでデータベースを更新する機能によって、毎日 24 時間いつでもアクセスできます。

Sybase IQ の「マルチプレックス」は、スケーラビリティの高い共有ディスク・グリッド・テクノロジーです。マルチプレックスを使用すると、共有データ・ソースに接続された独立のデータ処理ノードを介して、データのロードとクエリを同時に実行できます。詳細と構文については、『Sybase IQ マルチプレックスの使用』を参照してください。

Sybase IQ をさらに理解する

このマニュアルでは、Sybase IQ システムの管理方法について説明します。対象読者は、データベース管理者と、データベースの作成、ロード、操作の問題を理解する必要のあるユーザです。「はじめに」には、管理者にとって役に立つ他のマニュアルが記載されています。

システム管理のツール

データベース管理を支援するために、Sybase IQ では主なツールを 2 つ用意しています。

- **Sybase Central**。Sybase データベースを管理するためのアプリケーションです。このアプリケーションが支援するのは、データベース・オブジェクトを管理する場合や、データベースの作成、データベースのバックアップ、ユーザの追加、テーブルとインデックスの追加、データベースのマルチプレックス機能の有効化、データベース・パフォーマンスのモニタリングなどの一般的な管理タスクを実行する場合です。Sybase Central には、Java ベースのグラフィカル・ユーザ・インタフェースがあり、グラフィック・ツールを使用できるオペレーティング・システムで使用できます。
- **dbisql**。Interactive SQL とも呼ばれます。このアプリケーションを使用すると、SQL 文を対話的に入力し、データベースに送信できます。dbisql では、すべてのプラットフォームでウィンドウのようなユーザ・インタフェースを使用します。

『Sybase IQ の概要』では、Sybase Central と dbisql を使用して簡単な管理タスクを実行する方法について説明しています。Sybase Central と DBISQL を使い慣れていない場合は、『Sybase IQ の概要』を参照してこれらのツールについて理解した上で、このマニュアルのチュートリアルを使用してください。

Sybase IQ ではこれらのツールのほかに、システム管理機能を実行するストアド・プロシージャが多数用意されています。詳細については、「[ストアド・プロシージャ](#)」(7 ページ)を参照してください。ユーザ独自のプロシージャとバッチを作成することもできます。IQ のイベント処理機能を利用して、独自のシステム管理ツールを開発することもできます。詳細については、『システム管理ガイド第 2 巻』の「[第 6 章 スケジューリングとイベント処理によるタスクの自動化](#)」を参照してください。

Sybase IQ には、ストアド・プロシージャに加えて、バッファ・キャッシュのパフォーマンスをモニタするツールも用意されています。このモニタは、バッファ・キャッシュ、メモリ、Sybase IQ 内で行われた I/O 関数の統計情報を収集し、それらをログ・ファイルに保管します。詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

照合など、コマンドライン・ユーティリティに依存する管理タスクもあります。コマンドライン・ユーティリティについては、このマニュアルの他の章、および『ユーティリティ・ガイド』を参照してください。

データベース・サーバ

「データベース・サーバ」は、Sybase IQ システムの「頭脳」です。ユーザはデータベース・サーバを介してデータにアクセスし、直接はアクセスしません。データベースの情報を要求すると、要求はデータベース・サーバに送られ、そこで命令が実行されます。

設定可能なテーブル領域

「テーブルスペース」は、記憶領域全体のサブセットとして管理できるデータベース内の記憶領域の単位です。個々のオブジェクトやサブオブジェクトは、個別のテーブル領域に割り付けることができます。各データベースは複数のテーブル領域を含みます。

DB 領域、dbfile、ストア

Sybase IQ 15.2 では、**DB 領域**は、1 つまたは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テーブルスペースです。

用語の意味は、使用している製品のバージョンによって変わります。Sybase IQ 12.7 では、DB 領域とデータベース・ファイルは 1 対 1 のマッピングが実装されていました。DEFAULT_DISK_STRIPING オプションが「ON」の場合、Sybase IQ ではすべての利用可能な DB 領域にデータを自動的に分散し、パフォーマンスを最適化して管理を容易にしていました。テーブルまたはインデックスを作成する DB 領域をユーザが指定することはできませんでした。

対応する論理ファイル名と物理ファイル・パスを持つ「**dbfile**」という用語は、各オペレーティング・システム・ファイルを指します。

各物理ファイルのパスはユニークでなければなりません。dbfile 名は、DB 領域名と同じものを使用できます。

「ストア」は、特別な目的のための永続的なデータまたはテンポラリ・データを保存する 1 つまたは複数の DB 領域です。Sybase IQ には、次の 3 つのストアがあります。

- **カタログ・ストア**。システム DB 領域と、そのほかに最大 12 個のカタログ DB 領域が含まれます。
- **IQ メイン・ストア**。IQ_SYSTEM_MAIN DB 領域と他のユーザ定義の DB 領域が含まれます。
- **IQ テンポラリ・ストア**。IQ_SYSTEM_TEMP DB 領域が含まれます。

これらの DB 領域の詳細については、「[DB 領域のタイプ](#)」(156 ページ)を参照してください。

SQL Anywhere と Sybase IQ

カタログ・ストアは、SQL Anywhere ストアによく似ています。SQL Anywhere は、IQ があってもなくても存在できるリレーショナル・データベースです。カタログ・ストア内で SQL Anywhere スタイルのテーブルを IQ のテーブルと共存させたり、SQL Anywhere のデータベースを別に作成したりできます。

SQL Anywhere のテーブルと IQ のテーブルではフォーマットが異なります。SQL Anywhere データベース内でオブジェクトを作成するために使用するコマンドは IQ ストアのものと同じ場合がありますが、コマンドに指定できる機能が少し異なります。IQ ストアでの操作には、このマニュアルまたは『リファレンス：文とオプション』にあるコマンド構文を必ず使用してください。

このマニュアルでは、IQ ストアとそれに関連付けられているカタログ・ストアの管理方法について説明します。SQL Anywhere データベースがある場合、または、SQL Anywhere スタイルのテーブルをカタログ・ストア内に持つ場合、それらの作成、管理、使用については SQL Anywhere のマニュアルを参照してください。

Sybase IQ と SQL Anywhere の相違の概要については、『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「[付録 A 他の Sybase データベースとの互換性](#)」を参照してください。

セキュリティの概要

データベース管理者 (DBA) の役割は、データベースのセキュリティを管理することです。Sybase IQ では、管理者がユーザに権限を割り当てることによってセキュリティを管理します。

ユーザのタイプ

Sybase IQ では、各 IQ データベースに対して 3 つのカテゴリのユーザが認識されます。

- データベース管理者 (DBA)。データベース上ですべての操作を実行する完全な権限があります。このマニュアルは、通常ほとんどの管理タスクを実行する DBA を主に対象としています。
- 所有者。特定のデータベース・オブジェクトを作成したユーザです。そのオブジェクトに対してあらゆる操作を実行できます。
- *public* ユーザ。他のすべてのユーザです。あるオブジェクトの所有者は、他のユーザが所有するオブジェクトの *public* ユーザと見なされます。

パーミッションの付与

DBA はあらゆるタスクを実行できますが、DBA 以外のユーザが特定のタスクを実行するには権限の付与が必要です。たとえば、次のタスクを実行するには、適切な権限が必要です。

- データベースへの接続
- データベース、テーブル、インデックス、外部キーなどのデータベース・オブジェクトの作成
- 特定の DB 領域へのオブジェクトの配置
- データベース・オブジェクトの構造の変更
- データの挿入、更新、または削除
- データの選択 (表示)
- プロシージャの実行

DBA は、あらゆるタイプの権限を任意のユーザに付与できます。同様に、他のユーザが権限を付与できる場合もあります。ユーザが実行できるタスクと DBA がユーザを管理する方法の詳細については、「[第 8 章 ユーザ ID とパーミッションの管理](#)」を参照してください。

同時処理

Sybase IQ では、あるユーザがデータの挿入や削除またはデータベースのバックアップを行っているときに、別の複数のユーザが同時にデータベースに対してクエリを実行できます。テーブルの作成、削除、変更といったデータベースの構造の変更を行う場合、他のユーザは一時的にそのテーブルから排除されますが、データベース内の他のテーブルにアクセスするだけのクエリは、続行できます。

Sybase IQ では、複数バージョンのテーブル・データを管理することによって、そうした同時処理の間、データベースの整合性を保ちます。この方法を理解するには、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

マルチプレックス機能

マルチプレックスは、クラスタ化されたシステム内の複数のマシンを使用してアプリケーションをスケールし、共有ディスク記憶領域にある 1 つの IQ データベースをマウントして開くことができる Sybase IQ の強力な機能です。詳細については、『[Sybase IQ マルチプレックスの使用](#)』を参照してください。

ストアド・プロシージャ

Sybase IQ の「ストアド・プロシージャ」は、システム管理に役立ちます。ストアド・プロシージャによって、データベースとユーザの情報がわかり、データベース上でさまざまな操作が実行されます。この項では、ストアド・プロシージャについて簡単に説明します。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、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」を参照してください。

通常、ストアド・プロシージャは、そのプロシージャを実行したデータベース上で動作します。たとえば、ストアド・プロシージャ `sp_addlogin` を `iqdemo` データベースで実行すると、`iqdemo` にユーザが追加されます。

ユーザ独自のストアド・プロシージャも作成できます。詳細については、『システム管理ガイド第 2 巻』の「[第 1 章 プロシージャとバッチの使用](#)」を参照してください。

注意 例に示されているほとんどの文で、`iqdemo` データベースを使用しています。これは、Sybase IQ とともにインストールされるスクリプトを使用して作成できるデモ・データベースです。

Sybase IQ ストアド・プロシージャ

Sybase IQ には、情報の表示や管理を行う多数のストアド・プロシージャが用意されています。たとえば、`sp_iqstatus` は一般的なステータス情報を表示し、`sp_iqcheckdb` は現在のデータベースの妥当性を検査して割り付けの問題を修復できます。

IQ ストアド・プロシージャの詳細については、『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「[第 7 章 システム・プロシージャ](#)」を参照してください。

各種のマルチプレックス・ストアド・プロシージャも用意されています。詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。

カタログ・ストアド・プロシージャ

Adaptive Server Enterprise のカタログ・ストアド・プロシージャ以外にも、システム・プロシージャやカタログ・ストアド・プロシージャがあります。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「[第 7 章 システム・プロシージャ](#)」を参照してください。

Sybase IQ では、Adaptive Server Enterprise のカタログ・ストアド・プロシージャの `sp_column_privileges`、`sp_databases`、`sp_datatype_info`、`sp_server_info` をサポートしていません。

システム・テーブルとシステム・ビュー

Sybase IQ システム・テーブルには、データベース・サーバが Sybase IQ システムを管理するために必要なすべての情報が入っています。システム・テーブルはカタログ・ストア内にあり、カタログ・テーブルとも呼ばれます。システム・テーブルは、システム・ビューからのみアクセスできます。SYS ユーザ ID によって、システム・テーブルを所有します。

システム・テーブルの情報は次のとおりです。

- データベースの特性
- テーブルの特性。テーブル定義、各テーブルのサイズやロケーションについての情報など
- インデックスについての情報
- データベースと `dbisql` オプションの現在の設定

システム・ビューは、対応するシステム・テーブルの情報をより読みやすい形式で表示します。

システム・テーブル、システム・ビュー、およびそれらの内容の詳細については、『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「[第 8 章 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。

コマンドと関数

Sybase IQ コマンドはすべて SQL 文です。SQL とは、構造化問い合わせ言語のことで、データベース・アプリケーションで通常使用される言語です。Sybase IQ の SQL では、SQL Anywhere の SQL と同じ構文を使用します。異なる点は、Sybase IQ または SQL Anywhere のどちらかの製品だけでサポートされている特定の機能だけです。また、Sybase IQ の SQL では、Adaptive Server Enterprise が使用する SQL 言語である Transact-SQL との高度な互換性がサポートされています。

この項では、使用できる文と関数のタイプについて説明します。このマニュアルの他の章では、さまざまな管理タスクを実行するための文について説明します。サポートされている関数と文の詳細については、『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「[第 4 章 SQL 関数](#)」と『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」を参照してください。

SQL 文のタイプ

基本的に使用する SQL 文は次の 3 種類です。

- データ定義言語 (DDL: Data Definition Language) 文。データベース・スキーマ、テーブル、インデックスの定義と修正を行うことができます。DDL 文の例には、CREATE TABLE、CREATE INDEX、ALTER TABLE、DROP などがあります。
- データ操作言語 (DML: Data Manipulation Language) 文。データのクエリや、データベースのデータの入出力を実行できます。DML 文の例には、SELECT、SET、INSERT があります。
- プログラム制御文。プログラム実行フローを制御します。プログラム制御文は、Sybase IQ テーブル上で直接処理しません。プログラム制御文の例には、IF、CALL、ROLLBACK があります。

関数

関数はデータベースから情報を返します。式が使用できる場所でならどこでも使えます。Sybase IQ では次の関数が用意されています。

- データを集約させる関数 (AVG、COUNT、MAX、MIN、SUM、STDDEV、VARIANCE など)
- 数値データを操作する関数 (ABS、CEILING、SQRT、TRUNCATE など)
- 文字列データを操作する関数 (LENGTH、SOUNDEX、UCASE など)
- 日付と時刻のデータを操作する関数 (TODAY、DATEDIFF、DATEPART、MINUTES など)
- 取得したデータを別のフォーマットに変換する関数 (CAST、CONVERT)
- 分析データを操作する関数 (DENSE_RANK、NTILE、PERCENT_RANK、PERCENTILE_CONT、PERCENTILE_DISC、RANK など)

メッセージ・ロギング

各データベース用の IQ メッセージ・ログ・ファイルがあります。このログ・ファイルのデフォルト名は *dbname.iqmsg* です。IQ メッセージ・ログ・ファイルは、新しく作成したデータベースの起動時にカタログ・ストアと同じディレクトリに作成されます。データベースの作成者は別のロケーションや別のファイル名を指定することもできます。

デフォルトでは、Sybase IQ は、次のタイプのメッセージをメッセージ・ログ・ファイルに記録します。

- エラー・メッセージ
- ステータス・メッセージ
- 挿入通知メッセージ
- クエリ・プラン

メッセージ・ログ・ファイルは、他のテキスト・ファイルと同じように確認できます。

ファイルの冒頭部分とデータベースの開始時の出力は次のとおりです。

```
I. 05/06 17:07:25.0000000000 OpenDatabase Completed
I. 05/06 17:07:25.0000000000 IQ cmd line srv opts:
I. 05/06 17:07:25.0000000000 IQ full cmd line: -c 48m -gc 20 -gd all
-gl all -gm 10 -gp 4096 -ti 4400 -n sunopt_demo -x tcpip{port=1870}
iqdemo.db -gn 25 -o /sunopt/users/user1/sybase/IQ-15_2/logfiles/
sunopteng_iqdemo.0007.srvlog -hn 7
I. 05/06 17:07:25.0000000000 DB:r/w, Main Buffs=127, Temp Buffs=95,
Pgsz=131072/8192blksz/16bpc
I. 05/06 17:07:25.0000000000 DB: Frmt#:23F/2T/1P (FF: 03/18/1999)
I. 05/06 17:07:25.0000000000 DB: Versn:
15.2.0.5071/090501/P/Mainline/Sun_Opteron/OS 5.10/64bit/2009-05-01 01:21:40
I. 05/06 17:07:25.0000000000 DB: Name: /sunopt/users/user1/iqdemo.db
I. 05/06 17:07:25.0000000000 DB: Txn ID Seq: 1
I. 05/06 17:07:25.0000000000 DB: DBID Blk: 7730
I. 05/06 17:07:25.0000000000 DB: IQ Server sunopt_demo, PID 24485, LOGIN user1
I. 05/06 17:07:25.0000000000 DB: Database encryption is OFF.
I. 05/06 17:07:25. 0000000000
Mem:44mb/M44
Main      Blks:U7841/20%, Buffers:U6/L1
Temporary Blks:U65/0%, Buffers: U4/L0
Main      I:L24/P4 O:C2/D2/P0 D:0 C:100.0
Temporary I:L34/P0 O: C4/D4/P0 D:0 C: 0.0
I. 05/06 17:07:25.0000000000 Collation ISO_BINENG, Case Respect,
Blank Padding On, Comparisons are Binary
I. 05/06 17:07:26.0000000000 RcvyCmpl
I. 05/06 17:07:26.0000000000 Chk
I. 05/06 17:07:26.0000000000 ChkDone [NumTxnCP: 0]
I. 05/06 17:07:26.0000000000 PostChk
I. 05/06 17:07:26.0000000000 CloseDatabase
```

メッセージ・ログ内のバージョン情報

ファイルの先頭近くには、次のようなバージョン情報が表示されます。

```
Versn:
15.2.0.5071/100301/P/GA/Sun_Opteron/OS 5.10/64bit/2010-05-01 01:21:40
```

表 1-1: バージョン文字列の要素

バージョン文字列の要素	例	コメント
メジャー・リリース	15.2	先頭の 2 つのセグメントがメジャー・リリースを示す。
マイナー・リリース	15.2. <i>n</i>	このセグメントはマイナー・リリースを示す。
内部ビルド番号	5071	内部用のユニークな番号
内部ビルド日付	100301	内部用の日付識別子
内部ビルド・タイプ	P	P = 運用ビルド
リリース・タイプ	GA	GA = General Availability ESD = Engineering Software Distribution (メンテナンス)
ハードウェア・プラットフォーム	Sun_Opteron	ハードウェア・プロバイダの識別子
オペレーティング・システム・バージョン	OS 5.10	OS バージョンとビット・モードは、ソフトウェアが現在稼動しているシステムではなく、ソフトウェアがビルドされたシステムを表す。
ビット・モード	64bit	
ビルドされた日付と時刻	2010-03-01 01:21:40	YYYY-MM-DD hh:mm:ss (ISO datetime フォーマット) で表示される。 YYYY 西暦年 4 桁 MM 月 2 桁 (01 ~ 12) DD 日 2 桁 (01 ~ 31) hh 24 時間表記の時 2 桁 (00 ~ 23) mm 分 2 桁 (00 ~ 59) ss 秒 2 桁 (00 ~ 59)

メッセージ・ログ内の照合情報

データベースを起動すると、IQ メッセージ・ログ内のエント리는、照合名、大文字と小文字の区別、ブランク埋め込みの状態、文字変換テーブル要件を示します。“Comparisons are conditioned” は、文字データを比較するときに変換テーブルが必要ないことを意味します。

メッセージ・ログ内の接続情報

接続ハンドルを示す接続線とデータベース・ユーザ名は、最初のトランザクションが開始した後、および次のトランザクションが開始する前に出力されます。この情報は、データベースの接続ごとに 1 回だけ出力されます。例を示します。

```
I. 04/23 13:02:37. 0000000003
Connect: SA connHandle: 1 SA connID: 3
IQ connID: 0000000003 User: DBA
```

この接続情報を使用すると、*.iqmsg* ファイル内のクエリ・プランを、クエリ・テキストに一致させることができます。特定の時点で、各接続が実行している文を判断するには、*sp_iqcontext* ストアド・プロシージャを実行します。

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場合は、「[接続情報の関連付け](#)」(570 ページ)を参照してください。

メッセージ・ログ内の挿入通知

IQ メッセージ・ファイルには、挿入操作とロード操作が記録されます。詳細については、「[通知メッセージの解釈](#)」(558 ページ)を参照してください。

LOAD および INSERT 文のパラメータを使用して、通知メッセージを OFF に設定できます。

メッセージ・ログ・ファイルの管理

デフォルトでは、メッセージ・ログ・ファイルのサイズは無制限に増大し、ファイルはデータベースを削除しないかぎり存在します。メッセージ・ログ・ファイルのサイズを制御するには、ファイルのサイズの制限を設定し、メッセージ・ログ・ラッピングまたはログのアーカイブ操作を有効にします。

メッセージ・ログ・ファイルは、データベースを停止した後ならいつでも削除、名前の変更、コピーを行うことができます。

メッセージ・ログの管理は、サーバ・プロパティの *IQMsgMaxSize* と *IQMsgNumFiles* か、またはサーバ起動スイッチの *-iqmsgsz* と *-iqmsgnum* によって制御されます。

- *IQMsgMaxSize* または *-iqmsgsz* によって、アクティブなメッセージ・ログのサイズの上限をメガバイト (MB) 単位で設定します。使用できる値は 0 ~ 2047 (0 と 2047 を含む) の整数です。デフォルトは 0 で、メッセージ・ログ・ファイルのサイズの制限がないことを示します。
- *IQMsgNumFiles* または *-iqmsgnum* によって、メッセージ・ログのアーカイブの数を設定します。使用できる値は 0 ~ 64 (0 と 64 を含む) です。デフォルトは 0 で、メッセージはメインのメッセージ・ログ・ファイル内でラッピングされ、アーカイブされないことを示します。

サーバ・プロパティの値は、対応するサーバ起動スイッチよりも優先されず、サーバを起動すると、*-iqmsgsz* と *-iqmsgnum* サーバ起動スイッチの値がサーバ・ログ・ファイルに書き込まれます。

IQMsgMaxSize サーバ・プロパティと IQMsgNumFiles サーバ・プロパティの設定の詳細については、『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「第 7 章 システム・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。

-iqmsgsz サーバ起動スイッチと -iqmsgnum サーバ起動スイッチの設定の詳細については、『ユーティリティ・ガイド』の「第 1 章 データベース・サーバの実行」の「データベース・サーバの起動」を参照してください。

注意 IQMSG_LENGTH_MB データベース・オプションは廃止されました。既存のスクリプトやコードから削除してください。IQMSG_LENGTH_MB を使用しようとする、エラーが返されます。

メッセージ・ログ・ラッピング

メッセージ・ログ・ラッピングを有効にした場合、ログ・ファイルが IQMsgMaxSize サーバ・プロパティまたは -iqmsgsz サーバ起動スイッチで指定した最大サイズに達すると、新しいメッセージはファイルの先頭から書き込まれます。既存のメッセージは、行単位で上書きされます。

メッセージ・ログ・ラッピングは、IQMsgMaxSize または -iqmsgsz をメッセージ・ログ・ファイルのサイズに制限がないことを示すゼロより大きい値に設定し、IQMsgNumFiles サーバ・プロパティまたは -iqmsgnum サーバ起動スイッチをゼロ (デフォルト) に設定すると有効になります。

ラッピングを有効にしたときは、新しいメッセージが挿入された場所をメッセージ・ログ・ファイル内の <next msg insertion place> タグで知ることができます。ログ・ラッピングが有効になっており、ファイル内の最後のメッセージが最新のメッセージとはかぎらないことに注意を促すために、ファイルの先頭と末尾に追加のタグが表示されます。

メッセージ・ログのアーカイブ操作

メッセージ・ログのアーカイブ処理を有効にすると、メッセージ・ログ・ファイル・サイズを無限に大きくできるようにしなくてもファイルに書き込まれるすべての情報を維持できます。メッセージ・ログ・アーカイブは、アクティブな .iqmsg メッセージ・ログ・ファイルの内容が保存されるファイルです。

メッセージ・ログのアーカイブ処理は、sa_server_option システム・ストア・プロシージャを使用して IQMsgMaxSize サーバ・プロパティまたは -iqmsgsz サーバ起動スイッチをゼロより大きい値に設定し、IQMsgNumFiles サーバ・プロパティまたは -iqmsgnum サーバ起動スイッチをメッセージ・ログ・アーカイブの数に設定します。サーバ・プロパティの値は、対応するサーバ起動スイッチの値よりも優先されます。

メッセージ・ログ・
アーカイブの名前

.iqmsg メッセージ・ログ・アーカイブの名前は、*logname.iqmsg.n* のパターンに従います。ここで、*n* は 1 以上の整数、*logname* は CREATE DATABASE 文で指定されたメッセージ・パスの名前、または *.iqmsg* というサフィックスを持つデータベース名です。アーカイブは、メッセージ・ログ・ファイルと同じディレクトリに読み込み専用ファイルとして作成されます。

メッセージ・ログ・ファイルの *logname.iqmsg* が満杯になり、メッセージ・ログ・アーカイブの数が *IQMsgNumFiles* または *-iqmsgnum* で指定された数より小さい場合は、サーバは現在のメッセージ・ログの名前を変更して新しいアーカイブを作成します。新しいメッセージ・ログは、*logname.iqmsg* という名前で作成されます。

たとえば、*-iqmsgnum* が 5 の場合、メッセージ・ログ・アーカイブは、*logname.iqmsg.1*、*logname.iqmsg.2*、*logname.iqmsg.3*、*logname.iqmsg.4*、*logname.iqmsg.5* の順序で作成されます。

メッセージ・ログ・ファイルの *logname.iqmsg* が満杯になり、メッセージ・ログ・アーカイブの最大数がすでに存在する場合、ファイル・エクステンションが *.1* の最も古いアーカイブが削除されてから現在のメッセージ・ログがアーカイブ処理されます。

たとえば、*-iqmsgnum* が 5 で *logname.iqmsg.5* が存在する場合、次のように処理されます。

- ファイル *logname.iqmsg.1* が削除されます。
- ファイル *logname.iqmsg.2* ~ *logname.iqmsg.5* は *logname.iqmsg.1* ~ *logname.iqmsg.4* に、それぞれ名前が変更されます。
- アクティブなメッセージ・ログ・ファイルの名前が *logname.iqmsg.5* に変更されます。
- 新しいメッセージ・ログ・ファイル *logname.iqmsg* が作成されます。

IQMsgNumFiles (または *-iqmsgnum*) の値がゼロより大きい場合、サーバはこの方法を使用して常に最新のメッセージ・ログを維持します。

メッセージ・ログ管理の
エラー

メッセージ・ロギング中にディスクが満杯になると、メッセージ・ロギングは停止します。"Disk Full!!! Message logging stopped." というメッセージがサーバのログに書き込まれます。

メッセージ・ログの管理時に次のエラーが発生すると、メッセージ・ロギングは停止します。

- ファイルの名前変更エラー。サーバは現在アクティブなログをアーカイブ処理できないか、既存のログ・アーカイブの名前を変更できない。
- ファイル作成エラー。サーバは新しいメッセージ・ログを作成できない。
- ファイル削除エラー。サーバは古いアーカイブを削除できない。
- ディスク I/O エラー。

エラー状態が解決すると、サーバは自動的にメッセージ・ロギングを再開します。メッセージは、メッセージ・ロギングの停止時と再開時にサーバ・ログに書き込まれます。

たとえば、ファイルの名前の変更に失敗した場合は、"Renaming of <filename> failed. Message log could not be archived. Message logging stopped." というメッセージがサーバのログに書き込まれます。ロギングが再開されると、"Message logging resumed." のメッセージがサーバ・ログに書き込まれます。

ファイルが存在しないためにファイルの削除や名前の変更に失敗した場合は、エラーは無視され、ログ管理処理は続行されます。

iqmsg ファイルとログ・アーカイブのバックアップ

dbname.iqmsg メッセージ・ログ・ファイルとメッセージ・ログ・アーカイブはリストアに必要なファイルではありませんが、これらのファイルをバックアップしておくことをおすすめします。

リストア中に問題が発生した場合、データベースを停止してからバックアップを開始したかどうかを確認できる情報が *.iqmsg* ファイルに入ります。メッセージ・ログ・ファイルは、リカバリ中に発生したデータベース障害の原因を診断するときに役立つ場合があります。あとで分析に使用するため、リストア前にコピーを作成してください。

IQ メッセージ・ログ・ラッピングを有効にしている (IQMsgMaxSize サーバ・オプションまたは *-iqmsgsz* サーバ起動スイッチがゼロ以外で、IQMsgNumFiles サーバ・オプションまたは *-iqmsgnum* サーバ起動スイッチがゼロ) の場合は、*.iqmsg* ファイルをバックアップしておき、必要な場合にすべてのメッセージにアクセスして診断を下せるようにすることをおすすめします。

メッセージ・ログのアーカイブ処理を有効にしている (IQMsgMaxSize サーバ・オプションまたは *-iqmsgsz* サーバ起動スイッチがゼロ以外で IQMsgNumFiles または *-iqmsgnum* サーバ起動スイッチがゼロ以外) の場合は、メッセージ・ログ・アーカイブは自動的にバックアップされます。アーカイブされるメッセージ・ログの最大サイズは 128GB で、ほとんどの場合、このサイズでバックアップに対応できます。

注意 メッセージ・ログ・アーカイブのバックアップは、サーバを再起動する前に行う必要があります。サーバが再起動した後、*dbname.iqmsg* ファイルが一杯になった場合、既存のログ・アーカイブは無視され、新しいアーカイブが作成されます。古いアーカイブ・ログを保持するには、サーバの再起動前にファイルをバックアップします。

メッセージ・ログと SQL ログ・ファイルの同期

夏時間を使用する国で Sybase IQ を実行する場合、標準時に戻された後に Sybase IQ サーバを再起動する必要があります。再起動により、Sybase IQ メッセージ・ログと、サーバ起動時に `-zr` スイッチを指定することで作成される SQL ログの時差が修正されます。Sybase IQ サーバは、夏時間に変更された後にも再起動する必要があります。

参照

- サーバ起動スイッチの設定の詳細については、『ユーティリティ・ガイド』の「データベース・サーバの実行」の章の「データベース・サーバの起動」を参照してください。
- サーバ・プロパティの設定の詳細については、『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「[第7章 システム・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。

ユーティリティ・データベース

ユーティリティ・データベースは、基本的にはデータを持たないデータベースです。これは、接続先データベースを必要とするときに、実際のデータベースが存在しないかデータベースを実行できない場合に、データベース・サーバが使用するものです。Sybase IQ をインストールすると、ユーティリティ・データベースが自動的に作成されます。

このデータベースを削除しないでください。次の操作では、このデータベースが必要になります。

- `START ENGINE` コマンドでデータベースを指定しないで、データベース・サーバを起動する場合
- 接続先データベースがないときにデータベースを作成または削除する場合
- メディア障害などが原因で他のデータベースが使用できないときに、データベース・サーバを起動する、またはデータベースに接続する場合
- データベースをリストアする場合

デフォルトでは、ユーティリティ・データベースのユーザ ID は `DBA` であり、パスワードは `sql` です。このユーザ ID やパスワードは、インストール時に変更したり、後から実行ディレクトリにある `util_db.ini` ファイルの接続パラメータを編集して変更したりできます。

ユーティリティ・データベースの詳細については、「[ユーティリティ・データベース・サーバのセキュリティ](#)」(345 ページ)を参照してください。

大容量データベースの管理

Sybase IQ 独自の設計機能により、テラバイト単位のデータを格納できるほど、データベースを拡張できます。その構造はインデックベースであるため、入力ロー・データのサイズに比べて、データの格納領域は格段に小さくて済み、従来のリレーショナル・データベースに比べて、アクセスも高速に行われます。これらの機能のため、Sybase IQ は、大容量データベース (VLDB) を格納およびアクセスするのに最適です。

データベース管理者は、パフォーマンスに影響するオプションと機能について理解し、記載されているガイドラインに従う必要があります。デフォルト設定の多くは、そのまま使用しても最大の効率が自動的に得られるようになっていますが、特定のオプション設定については、最も高速な結果が得られるように、各自の設定、ロードの要件、クエリに基づいて、場合によっては設定を変更する必要があります。これらのオプションを適切に設定することは、すべての Sybase IQ データベースにおいて、最高のパフォーマンスを実現するために必須ですが、データベースがテラバイト単位で増大する場合には特に重要です。

この項では、大容量データベースの管理を支援する Sybase IQ の機能について説明し、詳細な説明と推奨事項の参照先を示します。

メモリ使用の管理

メモリの適切な割り当ては、すべての IQ データベースのパフォーマンスにおける主要な要因です。Sybase IQ では、そのバッファ・キャッシュ内のメモリをロードとクエリで使用します。また、接続、トランザクション、バッファ、データベース・オブジェクトの管理でも、メモリを使用します。

Sybase IQ には、メイン・ストア用とテンポラリ・ストア用の 2 つのバッファ・キャッシュがあります。これらのキャッシュのデフォルト・サイズは、データ・ウェアハウスの運用にとっては十分ではありません。データベースとテーブル、ロードとクエリの組み合わせ、オペレーティング・システムと使用可能なメモリ量に影響するその他のアプリケーションなどの要因を反映するように、サイズを調整する必要があります。

Sybase IQ のメモリ使用、IQ キャッシュ・サイズを決定するサーバとデータベースのオプション、プラットフォーム上で使用可能な総メモリに影響するその他のオプションの詳細については、『パフォーマンス&チューニング・ガイド』の「システム・リソースの管理」を参照してください。

データ・ロードの管理

データベースが大きくなるほど、データのロードを適切に管理することが非常に重要になってきます。ニーズを満たすようにロードを調整する方法には、次の方法があります。

- バッファ・マネージャを分割してロック競合を回避する方法。CPU の数に基づくバッファ分割はデフォルトで有効であり、サーバまたはデータベース・オプションを設定することによって調整できます。詳細については、「[ロック競合の管理](#)」(412 ページ) を参照してください。
- ロード用に十分なメモリを割り当てる方法。ただし、システム上で使用可能なメモリ以上のメモリは割り当てません。
- セーブポイントの解放、コミット、チェックポイント操作時に使用されるデータ構造体用に領域を確保する方法。詳細については、『リファレンス：文とオプション』の「[MAIN_RESERVED_DBSPACE_MB オプション](#)」と「[データベース・オプション](#)」を参照してください。

ロード・パフォーマンスの管理を促進するその他の機能のリストについては、「[データのバルク・ロードのチューニング](#)」(326 ページ) を参照してください。

処理スレッドの管理

Sybase IQ では、オペレーティング・システムのスレッドを使用して、クエリとロードを処理します。スレッドの使用を制御するオプションのデフォルト設定は、通常は、十分に適切なパフォーマンスを実現できる値です。ただし、場合によっては、これらの設定を変更する必要があります。Sybase IQ におけるスレッドの使用方法については、『パフォーマンス&チューニング・ガイド』の「[プロセス・スレッド・モデル](#)」を参照してください。スレッドの使用を制御するサーバ・オプションの設定については、『パフォーマンス&チューニング・ガイド』の「[データベース・サーバ](#)」を参照してください。

ディスク領域の管理

IQ システムのディスク I/O を管理する上で最も重要な要因は、次のとおりです。

- クエリとロードを行うのに十分なディスク領域を確保する
- そのディスク領域を効率的に使用して、高性能なマルチ CPU システムの処理速度をサポートできるだけの最高速の I/O を実現する

`sp_iqstatus` ストアド・プロシージャは、IQ メイン・ストアとテンポラリー・ストアで使用されている領域の割合を示します。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ーまたはメイン 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。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すると、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できます。詳細については、「[DB 領域管理の例](#)」(191 ページ) を参照してください。

ディスク・ストライピングは、最大の I/O パフォーマンスを得るための重要な方法です。ディスク・ストライピングでは、複数のディスク・ドライブにわたってデータをランダムに分散します。Sybase IQ 内部のディスク・ストライピング機能に加えて、オペレーティング・システムや、ディスク管理ソフトウェアとハードウェアのディスク・ストライピング機能も利用できます。Sybase IQ のディスク・ストライピングは、デフォルトで有効になっています。これらのディスク管理方法については、『パフォーマンス&チューニング・ガイド』の「[I/O の分散](#)」を参照してください。

中間バージョン

大容量データベース内のロードとクエリを管理する上での重要な要因は、Sybase IQ のトランザクションレベルでのバージョン管理です。特に、Sybase IQ は、中間のセーブポイントにトランザクションをロールバックする機能を備えており、長いトランザクションが完了できなかった場合でも、ロード全体を繰り返す必要はありません。この機能を最大限に利用する方法については、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

データベースの作成

Sybase IQ データベースを作成する場合は、正しい IQ ページ・サイズを選択することが非常に重要です。大容量データベースの場合は、128KB 以上の IQ ページ・サイズが必要です。詳細については、「[IQ ページ・サイズの選択](#)」(174 ページ)を参照してください。

インデックスの作成

Sybase IQ のカラムベースのインデックス構造によって、興味ある属性の選択や計算の実行が最適化されます。最適なパフォーマンスを実現するには、データとクエリの正しいインデックスのセットを持つ必要があります。データベースは、パフォーマンスに影響するすべてのカラムについて、インデックスを設定する必要があります。正しいインデックスの選択の詳細については、「[第 6 章 Sybase IQ インデックスの使用](#)」を参照してください。

クエリの最適化

Sybase IQ のクエリ・オプティマイザは、すべてのクエリを評価して、各種処理オプションを選択し、最適なパフォーマンスを実現するクエリ・プランを出力します。オプティマイザは、Sybase IQ の各リリース用に調整されており、ほとんどのクエリと、最大のデータベースを含めたほとんどのデータベースに対して最高のプランを選択します。

クエリ・プランを確認するオプションとクエリ・プランに影響するオプション、および最適なパフォーマンスを実現するクエリの構造に関する推奨事項については、『パフォーマンス&チューニング・ガイド』の「[第3章 クエリと削除の最適化](#)」を参照してください。

スキーマ設計

多くの場合、Sybase IQ は、データ・ウェアハウス設計で一般的な非正規化スキーマで適切に動作します。従来のリレーショナル・データベースでは、正規化によって冗長性を除去し、一貫性を向上することで、トランザクションの処理効率が向上します。データ・ウェアハウスでは、大規模な場合は特に、非正規化によって大量のデータに対するクエリ処理のパフォーマンスが向上します。詳細については、『パフォーマンス&チューニング・ガイド』の「[パフォーマンス向上のための非正規化](#)」を参照してください。

UNION ALL ビュー

多数のローから成るテーブルは、ロードに非常に時間がかかります。UNION ALL ビューは、この問題に対応する1つの方法です。Sybase IQ では、データを複数の別々のベース・テーブルに分けることによって(たとえば日付で分ける)、テーブルを分割できます。それらを、UNION ALL ビューという方法で結合し、論理的に1つの形に戻します。

UNION ALL ビューは、管理が容易です。たとえば、データを月ごとに分割する場合、テーブルを削除し、UNION ALL ビューの定義を適切に更新することで、月全体のデータを削除できます。日付の範囲述部を追加することなく、年、四半期などに対応する多くのビュー定義を作成できます。

UNION ALL ビューの確立と、これらのビューを参照するクエリの最適化に関する推奨事項については、『パフォーマンス&チューニング・ガイド』の「[ロードを高速化するための UNION ALL ビューの使用](#)」を参照してください。

この章について

この章では、Sybase IQ を使用してデータベース・サーバを起動し、データベースを起動し、データベースに接続する方法について説明します。

内容

トピック名	ページ
データベース・サーバの起動	21
Windows サービスとしてのサーバの実行	27
コマンド・ライン・スイッチの使用	27
サーバ・アクティビティのモニタ	41
データベース・サーバの停止	44
データベースの開始と停止	48
iqdemo データベースの起動	50
Sybase Central の開始と停止	50

データベース・サーバの起動

Sybase IQ では、この 3 つの手順を柔軟に実行できます。この章では、手順 1 と手順 2 を実行するためのオプションについて説明し、ユーザの条件に従って選択する項目を示します。手順 3 を実行するためのオプションについては、次の章で説明します。

Sybase IQ を実行するには、まずデータベース・サーバを起動します。

データベース・サーバは次の方法で起動できます。

- Sybase のユーティリティ `start_iq` を使用してサーバを起動します。詳細については、「[起動ユーティリティによるサーバの起動](#)」(22 ページ) を参照してください。
- Sybase Central で Start Database Server ウィザードを実行します。詳細については、「[Sybase Central からのサーバの起動](#)」(24 ページ) を参照してください。対話的にマルチプレックス・サーバを起動および停止するには、常に Sybase Central を使用します。
- Windows の [スタート] メニューからサーバを起動します。詳細については、「[Windows の \[スタート \] メニューからのサーバの起動](#)」(26 ページ) を参照してください。

- Sybase から提供される設定ファイルを使用して、サーバとデモ・データベースを起動します。詳細については、「[「iqdemo データベースの起動」\(50 ページ\)](#)を参照してください。
- 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します。

注意 システムの起動時に IQ サーバが自動的に開始するように、Windows システムを設定することもできます。詳細については、『インストールおよび設定ガイド』の「Installing Sybase IQ as a Service」を参照してください。

- ODBC データ・ソースにサーバ起動行を指定します。詳細については、「[「ODBC データ・ソースの作成と編集」\(70 ページ\)](#)を参照してください。
- ユーティリティ・コマンドにサーバ起動行を指定します。
- Interactive SQL から SQL コマンドを発行して、追加のサーバを起動します。詳細については、「[「dbisql からのサーバの起動」\(40 ページ\)](#)を参照してください。

注意 リモート・アクセス機能を使用して、他のデータベースからデータを挿入したり、他のデータベースにクエリを発行したりする場合は、『システム管理ガイド第2巻』の「[第4章 リモート・データへのアクセス](#)」と「[第5章 リモート・データ・アクセス用のサーバ・クラス](#)」を参照してください。

起動ユーティリティによるサーバの起動

サーバを起動するには、起動ユーティリティ `start_iq` または Sybase Central の使用をおすすめします。このコマンド・ライン・ユーティリティはすべてのプラットフォームで動作し、この章で後述する特別な場合を除いて、必須のすべてのパラメータが適切に設定されます。

起動ユーティリティは、通常次のような形式になっています。

```
start_iq [ server-options ] [ database-file  
[ database-options ], ... ]
```

次に、このコマンドの各要素について説明します。

- `server-options` には、データベース・サーバ名など、サーバで実行されるすべてのデータベースに対してサーバの動作を制御するオプションを指定します。

- *database-file* は、カタログ・ストアのファイル名です。コマンド・ラインに、1つまたは複数のデータベース・ファイル名を入力するか、このオプションを省略できます。これらのデータベースはそれぞれロードされ、アプリケーションで使用できます。データベースを起動したディレクトリにデータベース・ファイルが含まれている場合、パスを指定する必要はありません。データベース・ファイルが含まれていない場合は、パスを指定してください。*.db ファイル拡張子を指定する必要はありません。
- *database-options* は、開始するデータベース・ファイルごとに指定できるオプションであり、データベース・ファイルの動作の特定の部分を制御します。

この章には、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まれるように、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。

設定ファイルとよく使用されるオプションの詳細については、「[コマンド・ライン・スイッチの使用](#)」(27 ページ)を参照してください。

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、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。

注意 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。サーバ名とデータベース名の詳細については、「[サーバとデータベースの命名](#)」(29 ページ)を参照してください。

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

- 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。

`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。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。バージョン文字列とその他の起動メッセージの例については、「[メッセージ・ロギング](#)」(9 ページ)を参照してください。

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、`%IQLOGDIR15%` は、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 `%IQLOGDIR15%\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。

Sybase Central からのサーバの起動

Sybase Central は、IQ がサポートするすべてのプラットフォーム上で動作する、グラフィカル管理ツールです。

Sybase Central を実行するには、『[Sybase IQ Multiplex の使用](#)』の説明に従って、Sybase IQ Agent を設定および実行する必要があります。

データベース・サーバ起動ウィザードの実行

❖ 非マルチプレックス・サーバの起動

- 1 DBA 権限を持つアカウントを使用してログインします。
- 2 Sybase Central を起動します。

UNIX 上で Sybase Central を起動するには、次のコマンドを入力します。

```
% scjview
```

Windows 上で Sybase Central を起動するには、[プログラム]メニューから [Sybase] - [Sybase IQ 15.2] - [Sybase Central] を実行します。

- 3 Sybase Central の左側のウィンドウ枠で、Sybase IQ 15 を右クリックし、ドロップダウン・メニューから [サーバの起動] を選択します。
- 4 [次へ] をクリックします。シングル・サーバの起動がデフォルトです。
- 5 [接続パラメータ] 画面で指定する情報は、データベースを起動するために使用されます。

接続プロファイルには、簡単に接続できるようにパラメータが格納されています。接続プロファイルがある場合は、それをドロップダウンから選択し、[次へ] をクリックします。

サーバの接続プロファイルがない場合は、適切なテキスト・ボックスに接続パラメータを入力します (ウィザードの [概要] 画面で要求すると、これらのパラメータから接続プロファイルが作成されます)。マルチプレックス・サーバの起動には接続プロファイルを使用しないでください。

デフォルトの情報を変更するには、それを選択して書き換えるだけです。一般に、デフォルトのユーザ ID とパスワードで十分です。ユーザ ID を DBA から別のユーザに変更すると、機能が制限されます。

各サーバのデフォルトのポート番号は、使用されていない別の番号に必ず変更します。

Start Server Wizard

Connection Parameters

Specify Connection Information

Specify connection information for the server to be started.

Connection Profile: <Other>

Server Name: pcxp4338_iqdemo

Host Machine: pcxp4338

Port Number: 7777

Agent Port: 1870

Database Name: iqdemo

What user account (with DBA authority) do you want to use?

Username: DBA

Password: ●●●

< Back Next > Finish Cancel

- 6 パラメータを入力したら、[次へ]をクリックします。

[Database Path] 画面で、データベース名とパスを入力します。または [参照] ボタンを使用して、ローカル・ホスト上のデータベースを選択します。



データベースが暗号化されている場合は、ボックスに暗号鍵を入力します。

- 7 [次へ] をクリックします。[Summary] 画面には、指定した設定が表示されます。線の下にリストされているサーバ・オプションは、選択されている場合は強調表示され、選択されていない場合はグレー表示されます。以降の接続に備えて、[概要] 内のパラメータを指定する接続プロファイルを作成するには、チェックボックスをオンにします。
- 8 指定したオプションに問題がなければ、[完了] をクリックします。問題がある場合は、[戻る] ボタンをクリックして情報を変更します。

Windows の [スタート] メニューからのサーバの起動

この項では、Windows システム対応のデータベース・サーバを起動する方法について説明します。この章に記載した一般的な方法のいずれも使用できます。

[スタート] メニューを使用することによって、Windows でサーバを簡単に起動できます。[プログラム]-[Sybase]-[Sybase IQ 15.2] を選択します。

ここから、Sybase IQ デモ・データベース、Sybase Central、Interactive SQL を起動できます。

独自のデータベースを [プログラム] グループに登録できます。

Windows ホストが再起動する場合のサーバの再起動

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

Windows サービスとしてのサーバの実行

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。

コマンド・ライン・スイッチの使用

コマンド・ライン・スイッチを使用して、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「[第 1 章 データベース・サーバの実行](#)」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。詳細については、『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」および「[SET OPTION 文](#)」を参照してください。

コマンド・ライン・オプションの表示

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
```

大文字と小文字の区別

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。

設定ファイルの使用

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。

Sybase IQ では、次の設定ファイルが提供されています。

表 2-1: 設定ファイル

ファイル名	Location	用途
<i>default.cfg</i>	<i>\$IQDIR15/scripts</i> (UNIX), <i>%IQDIR15%\scripts</i> (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ます。Sybase IQ では、 <i>default.cfg</i> を新しい各データベース・ディレクトリにコピーし、 <i>params.cfg</i> という名前に変更します。 <i>default.cfg</i> (<i>scripts</i> 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ます。
<i>params.cfg</i>	データベース・ディレクトリ	Sybase Central からユーザがデータベースを作成した際に、作成された設定ファイル。このファイルへの変更は、この特定のファイルを使用するサーバにのみ影響します。
<i>iqdemo.cfg</i>	<i>\$IQDIR15/data/demo</i> (UNIX)、 <i>%IQDIR15%\data\demo</i> (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します。

これらの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、および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
```

これらのコマンド・ライン・オプションを次のように使用できます。

```
start_iq @mydb.cfg
```

複数のコマンド・ライン・スイッチを使用する例では、わかりやすくするために、設定ファイルに記述するときのようにそれぞれのスイッチを別の行で示しています。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。

注意 DBSTOP コマンドを使用してサーバを停止する場合は、サーバの起動時と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、パラメータが必要なときにそれらを確認を見つけることができます。

必須のコマンド・ライン・オプション

『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」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、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 32 ビット・プラットフォームでは、`-c 32M` が推奨されます。すべての 64 ビット・プラットフォームでは、`-c 48M` が推奨されます。

デフォルトの設定ファイル

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 (`default.cfg`) には、必要なすべてのスイッチが記述されています。このファイルは、Windows サービスと Sybase Central によってサーバを起動する場合に使用され、かつ、UNIX の `start_iq` コマンドで使用される `params.cfg` ファイルのソースです。設定ファイル内のスイッチは、`-n servername` スイッチを除き、`start_iq` コマンド・ラインで新しいスイッチを指定することによって上書きできます。

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。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「[第 2 章 SQL 言語の要素](#)」の「[予約語](#)」を参照してください。

サーバとデータベースの命名

サーバ・スイッチ (サーバに名前を付ける) 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にクライアント・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、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。

デフォルトの名前

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。

データベースの命名

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、覚えにくいファイル名の代わりにニックネームを使用できます。

サーバの命名

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。それにより、違うサーバに誤って接続することが防止されます。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、これは特に重要です。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 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。文字セットの詳細については、「[第 11 章 各国語と文字セット](#)」を参照してください。

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してください。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。

実行中のサーバでデータベースを起動する方法については、「[データベースの開始と停止](#)」(48 ページ)を参照してください。

大文字と小文字の区別および命名規則

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くことをおすすめします。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。

コマンド・ラインからのパフォーマンスとメモリの制御

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。この項に記載するオプションの多くは、IQ ストアの操作の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・ストアを操作するときを使用することのできるリソースにのみ関連するオプションは、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。カタログ・ストアにのみ影響を与えるオプションを指定する必要がある場合は、『リファレンス:文とオプション』の「[第 2 章 データベース・オプション](#)」を参照してください。

このマニュアルには、パフォーマンスのチューニングのためのヒントが示されています。Sybase IQ がメモリ、ディスク、プロセッサをどのように使用するか、リソース使用に対するユーザ接続の影響、リソース使用を制御するために設定できる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』を参照してください。

メモリ・オプションの設定

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

下に示すオプションと、サーバを実行中に設定できる他のオプションは、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のに使用可能なメモリ量を決定するために使用します。

IQ バッファ・キャッシュ・サイズ

メイン・キャッシュのデフォルトの IQ バッファ・サイズは 16MB で、テンポラリー・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ー・ストアのバッファ・キャッシュ・サイズは、次のいずれかの方法で設定する必要があります。

- 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、サーバ起動オプションの `-iqmc` (メイン・キャッシュ・サイズ) と `-iqtc` (テンポラリー・キャッシュ・サイズ) を指定します。次の方法が推奨されます。構文については、『ユーティリティ・ガイド』の「第 1 章 データベース・サーバの実行」の「データベース・サーバの起動」を参照してください。
- 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` または `temp_cache_memory_mb` パラメータとともに、`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、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` および `-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、`-iqmc` と `-iqtc` を使用してください。`-iqmc` および `-iqtc` オプションは、`sa_server_option` プロシージャによる設定を上書きしません。

`-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 つのテンポラリー・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。

同時ユーザ

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。ただし、`-gm` スイッチも設定する必要があります。このスイッチは、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。

$$gn_value = gm_value + 5$$

最小でも 25 を指定します。

UNIX プラットフォームの場合、詳細については、使用しているプラットフォームの『インストールおよび設定ガイド』を参照してください。

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して、特定の操作のパフォーマンスをチューニングすることもできます。または、`-iqgovern` スイッチを使用して、クエリの使用を制御できます。「[同時クエリ](#)」(33 ページ)を参照してください。

同時クエリ

`-iqgovern` スイッチは、サーバの同時クエリ数を指定します。これは、ライセンスによって規制される接続数とは異なります。`-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みてください。

連結メモリ

`-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。詳細については、『パフォーマンス&チューニング・ガイド』の「[プラットフォーム固有のメモリ・オプション](#)」を参照してください。

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のみ使用します。メモリが十分に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

`-iqmt` スイッチを使用して、Sybase IQ が使用できる処理スレッド数を設定します。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。

プロセッサの数

マルチプロセッサ・マシンで実行している場合は、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 `-gt` オプションを使用して指定できます。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。

**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` コマンドと、`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。次のいずれかの方法を使用することをおすすめします。

`start_iq`、`iqdemo.cfg`、または `default.cfg` を使用せずにサーバを起動する場合、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。

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 *nK* または *nM* として指定できます。

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、両方を実行する必要はありません。

- `-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。

注意 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。詳細については、「[IQ バッファ・キャッシュ・サイズ](#)」(32 ページ)を参照してください。

カタログ・キャッシュ・サイズの設定の詳細については、『ユーティリティ・ガイド』の「[データベース・サーバの起動](#)」に記載されている `-c`、`-ca`、`-ch`、`-cl` の各サーバ・オプションを参照してください。

CPU 数の設定

Sybase IQ 起動コマンドの `-iqnumbercpus` スイッチを使用すると、IQ で使用できる CPU の数を指定できます。この値は、リソース計画を目的として CPU の物理的な数を上書きします。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 の範囲です。

注意 `-iqnumbercpus` は、以下の場合にのみ使用することをおすすめします。

- Intel® CPU が搭載され、ハイパースレッディングが有効なマシンで、`-iqnumbercpus` を、使用可能な CPU 数に設定する場合。
- オペレーティング・システムのユーティリティを使って、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の設定

チェックポイント間隔

Sybase IQ はチェックポイントを使用して、参照ポイントや、データベースをリカバリするのに必要な他の情報を作成します。`-gc` スイッチを使用して、チェックポイントを使用しないでデータベース・サーバを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、この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 分が使用されます。

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、このスイッチによって上書きされないかぎり、最初に起動したデータベースによって指定されたりカバリ時間が使用されます。

その他のパフォーマンス関連のオプション

いくつかの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できます。このようなオプションには、`-gb` (Windows のデータベース・プロセス優先度)、`-p` (最大パケット・サイズ) などがあります。

コマンド・ラインからのパーミッションの制御

特定のグローバル・オペレーションを実行するのに必要なパーミッションは、いくつかのコマンド・ライン・オプションによって制御されます。

データベースの開始と停止

`-gd` オプションは、実行中のサーバでデータベースを起動できるのは、すでに接続しているデータベースに特定のレベルのパーミッションを持つユーザのみに制限します。

- **DBA** – DBA 権限を持つユーザだけが追加のデータベースを起動できます。
- **ALL** (`start_iq` と `default.cfg` のデフォルト) – すべてのユーザがデータベースを起動および停止できます。つまり、DBA が `START DATABASE` コマンドを発行する必要はありません (ユーザが特定の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には、依然としてパーミッションが必要であることを注意してください)。
- **NONE** – 稼働中のサーバでは、Interactive SQL からデータベースを起動または停止できません。

注意 サーバを起動するときに `-gd ALL` を設定しない場合、そのサーバでは DBA のみが追加のデータベースを起動できます。このため、サーバと同時に、またはサーバ起動以降に DBA によって起動されていないデータベースには、接続できません。ただし、DBA 以外のユーザもデータベースを停止できます。このため、一部のサイトでは、運用データベースにおけるこの設定を DBA に変更することが望まれます。

データベースの作成と削除

`-gu` オプションは、データベースを作成および削除できるユーザを、接続先のデータベースに対して特定のレベルのパーミッションを持つユーザに制限します。

- DBA – DBA 権限を持つユーザのみがデータベースを作成および削除できます。
- ALL (デフォルト) – すべてのユーザがデータベースを作成および削除できます。
- NONE – ユーザはデータベースを作成または削除できない。
- 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を作成および削除できます。詳細については、「[ユーティリティ・データベース](#)」(16 ページ)を参照してください。

サーバの停止

`-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) – DBA 権限を持つユーザだけ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

`-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、データベースに対して特定のレベルのパーミッションを持つユーザに制限します。

- DBA – DBA 権限を持つユーザのみ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 – データをロードできません。

最大カタログ・ページ・サイズの設定

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・ページに適合してい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、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、サーバ・ページ・サイズは、コマンド・ラインのデータベースの最大ページ・サイズと同じ大きさに設定されます。`-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` を指定しない場合は、コマンド・ラインで起動され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・ファイルを、ロードすることはできません。

大きなページ・サイズを使用する場合は、キャッシュ・サイズを増やしてください。キャッシュ・サイズが同じ場合は、領域を調整する柔軟性に欠け、大きなページの一部のみが保管されます。

注意 `-gp` オプションと次に示すページ・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。詳細については、「[IQ ページ・サイズの選択](#)」(174 ページ)を参照してください。

クライアント／サーバ環境の設定

3つの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。

- `-x` は、通信プロトコルのオプションを指定します。
- `-tl` はネットワーク接続タイムアウトを設定します。
- `-ti` はクライアント接続タイムアウトを設定します。

詳細については、以降の項を参照してください。

通信プロトコルの選択

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。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推奨されません。名前付きパイプは、ネットワーク通信では使用しません。

プロトコルの指定

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・サーバで利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数多くの同じオプションを制御できます。

次のコマンドによって、TCP/IP プロトコルを使用するサーバを起動します。

```
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あ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 スイッチの一部として機能するネットワーク通信パラメータの詳細については、「[ネットワーク通信のパラメータ](#)」(132 ページ)を参照してください。

非アクティブ接続の制限

Sybase IQ では、-tl と -ti の 2 つのパラメータを使用してユーザ接続を終了するタイミングを指定します。

デフォルト・ネットワーク・タイムアウトの設定

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 `-ti` スイッチを指定すると、接続時に `-ti` 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) 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 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、次の項で説明する `-ti` を推奨値に設定してください。このスイッチを次のように設定します。

```
-ti 300
```

この値でうまくいかない場合は、`-ti 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・タイムアウトの設定

`-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、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、このオプションは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) であり、たとえば、週末の休みに長期の実行を開始し、中間結果がロールバックされないようにできます。

詳細については、『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」の `-ti` サーバ・コマンド・ライン・オプションを参照してください。

強制リカバリ・モードでサーバを起動する

障害発生後にサーバを再起動する必要がある場合、通常と同じ起動オプションを使用して再起動できます。

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。サーバ・オプションについては、『ユーティリティ・ガイド』を参照してください。

dbisql 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、dbisql から新しいサーバを起動できます。START ENGINE コマンドを使用して、dbisql から指定されたサーバを起動します。

注意 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、STARTLINE 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。

例 1 行に入力された次の dbisql 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'
```

同じマシンでの複数のサーバまたはクライアントの起動

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。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、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 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。

共有メモリを使用するときの競合を回避するには、次の操作を行います。

- 各サーバに専用のテンポラリー・ディレクトリを作成します。両方のシステムに明示的に IQTMP15 環境変数を設定して、各クライアントがサーバと同じテンポラリー・ディレクトリを使用するようにします。環境変数の設定の詳細については、『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。
- 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。詳細については、『UNIX での ODBC データ・ソースの使用』(79 ページ)を参照してください。

- デフォルトでなく、明示的にパラメータを指定する接続文字列を使用します。
- 次のコマンドを発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、次の操作を行う必要があります。

- 起動時の `-n` パラメータで、各サーバにユニークな名前を指定します。
- `-x` パラメータで、各サーバにユニークなポート番号を指定します。

これらのパラメータの使用例については、『ユーティリティ・ガイド』を参照してください。

サーバ・アクティビティのモニタ

新しいユーザの場合は、サーバ・アクティビティをモニタすることをおすすめします。プラットフォームに適したコマンドを使用して、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

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。次は、AIX の場合に表示される例です。

```
Starting server myserver_iqdemo on myserver at port 2638 (05/22 16:18:58)
Run Directory      : /myserver/users/sybase/iq152/IQ-15_2/demo
Server Executable  : /myserver/users/sybase/iq152/IQ-15_2/bin64/iqsrv15
Server Output Log  : /myserver/users/sybase/iq152/IQ
15_2/logfiles/myserver_iqdemo.0001.srvlog
Server Version     : 15.2.0.5027/GA
Open Client Version : 15.0/P-EBF16070 ESD #15
User Parameters    : '@iqdemo.cfg' 'iqdemo.db'
Default Parameters : -ti 4400 -gn 25
```

```
I. 05/22 16:19:05.      Sybase IQ
I. 05/22 16:19:05.      Version 15.2
I. 05/22 16:19:05.      (64 ビット・モード)
I. 05/22 16:19:05. Copyright 1992-2010 by Sybase, Inc. All rights reserved
I. 05/22 16:19:05.
I. 05/22 16:19:05.4 physical processor(s) detected.
I. 05/22 16:19:05. Maximum number of physical processors the
server will use: 4
I. 05/22 16:19:05. Running AIX 5 3 on PPC
I. 05/22 16:19:05. Server built for PPC processor architecture
I. 05/22 16:19:05. 49152K of memory used for caching
```

```

I. 05/22 16:19:05.Minimum cache size: 49152K, maximum cache size:262144K
I. 05/22 16:19:05.Using a maximum page size of 4096 bytes
I. 05/22 16:19:05.Starting database "iqdemo"
(/myserver/users/sybase/iq151/IQ-15_2/demo/iqdemo.db)
at Fri May 22 2009 16:19
=====
IQ server starting with:
  10 connections      (      -gm )
  18 cmd resources   ( -iqgovern )
  267 threads        (      -iqmt )
  512 Kb thread stack size ( -iqtss )
 136704 Kb thread memory size ( -iqmt * -iqtss )
   4 IQ number of cpus ( -iqnumbercpus )
   0 MB maximum size of IQMSG file ( -iqmsgsz )
   0 copies of IQMSG file archives ( -iqmsgnum )

=====
I. 05/22 16:19:07.Transaction log:iqdemo.log
I. 05/22 16:19:08.Starting checkpoint of "iqdemo" (iqdemo.db) at Fri May 22 2009 16:19
I. 05/22 16:19:08.Finished checkpoint of "iqdemo" (iqdemo.db) at Fri May
22 2009 16:19
=====
IQ server starting with:
  10 connections      (      -gm )
  18 cmd resources   ( -iqgovern )
  267 threads        (      -iqmt )
  512 Kb thread stack size ( -iqtss )
 136704 Kb thread memory size ( -iqmt * -iqtss )
   4 IQ number of cpus ( -iqnumbercpus )
   0 MB maximum size of IQMSG file ( -iqmsgsz )
  0 copies of IQMSG file archives ( -iqmsgnum )

=====
I. 05/22 16:19:07.Transaction log:iqdemo.log
I. 05/22 16:19:08.Starting checkpoint of "iqdemo" (iqdemo.db)
at Fri May 22 2009 16:19
I. 05/22 16:19:08.Finished checkpoint of "iqdemo" (iqdemo.db)
at Fri May 22 2009 16:19I. 05/22 16:19:10.
Database "iqdemo" (iqdemo.db) started at Fri May 22 2009 16:19
I. 05/22 16:19:10.IQ Server myserver_iqdemo.
I. 05/22 16:19:10.Database server started at Fri May 22 2009 16:19
I. 05/22 16:19:10.Trying to start SharedMemory link ...
I. 05/22 16:19:10. SharedMemory link started successfully
I. 05/22 16:19:10.Trying to start TCPIP link ...
I. 05/22 16:19:15. TCPIP link started successfully
I. 05/22 16:19:15.Now accepting requests
New process id is 397436
Server started successfully

```

start_iq ログ・ファイル

start_iq ユーティリティを使用してサーバを起動すると、`$IQLOGDIR15` で定義されるディレクトリ内の ASCII テキスト・ファイルに、サーバ・アクティビティが記録されます。このファイルには、サーバからの標準出力とサーバのステータスが格納されます。

ログ・ファイル名のフォーマットは次のとおりです。

```
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、対象のサーバに対し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。例を示します。

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

`$IQLOGDIR15` ディレクトリを定義しないと、UNIX の場合、ログは `$IQDIR15/logfiles/` ディレクトリに書き込まれ、Windows の場合は Sybase IQ のインストーラで定義された `$IQLOGDIR15` 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て、次のコマンドを入力します。

```
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
```

例を示します。

```
jones 422034      1   0 17:47:36      -  0:04
/ibm64srv/users/sybase/iq152/IQ-15_2/bin64/
iqsrv15
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o
/ibm64srv/users/sybase/iq152/IQ-
15_2/logfiles/ibm64srv_iqdemo.0003.srvlog -hn 7
```

- 実行中のすべての Sybase IQ プロセスを表示する **stop_iq** ユーティリティを使用します。このユーティリティについては、以降の項で説明します。

Windows システムでは、システム・トレイに 1 つ以上の 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

`start_iq` 起動コマンドの `-o` パラメータを使用すると、`server.nnnn.srvlog` のデフォルト名を使用する代わりに、サーバ・ログ・ファイルの名前を指定できます。たとえば、サーバが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
```

このオプションを使用して、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、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

この項では、IQ データベース・サーバを停止する必要がある状況、停止方法、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について説明します。

サーバを停止および再起動する状況

限られたいくつかの状況では、IQ サーバを停止し、再起動する必要があります。

- Sybase IQ の新バージョンをインストールする場合。
- サーバ・コマンド・ライン・オプションをリセットする場合。
- 少数のサーバワイドなデータベース・オプションを反映する場合。完全なリストについては、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」の「[データベース・オプションのスコープと継続期間](#)」を参照してください。
- オペレーティング・システム・セッションを終了する前。

サーバを停止する方法

データベース・サーバを停止するのに望ましい方法は次のとおりです。

- Sybase Central (UNIX または Windows) では、サーバ名を右クリックし、[サーバの停止] を選択します。

IQ マルチプレックス内のサーバを停止するには、Multiplex フォルダを開き、停止する必要があるサーバを右クリックして、[コントロール]-[停止] を選択します。

- UNIX では、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ます。詳細については、「例 - `stop_iq` を使用したサーバの停止」(46 ページ) を参照してください。

`stop_iq` を実行すると、次のメッセージが表示されます。

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

- Windows の場合は、データベース・サーバ画面の [終了] をクリックするか、またはシステム・トレイの IQ アイコンを右クリックし、[終了] をクリックします。
- Windows でサーバがサービスとして実行している場合は、コントロール・パネルの中でサービス・マネージャを開きます。サービスを選択し、[停止] をクリックします。

通常、サーバが1つまたは複数のクライアントに接続している間はサーバを停止しないでください。この状態でサーバを停止しようとする、コミットされなかったトランザクションが失われることを知らせる警告が表示されます。すべてのクライアントを切断するか閉じ、再度実行してください。

データベース・サーバは次の方法で停止できます。

- オペレーティング・システムのコマンド・ラインで適切なパラメータを指定して `DBSTOP` コマンドを入力します。サーバを起動するときと同じパラメータを使用します。正しい接続パラメータを指定しないと、`DBSTOP` はサーバに接続してシャットダウンできません。`DBSTOP` の使用方法の詳細については、『ユーティリティ・ガイド』の「第1章 データベース・サーバの実行」を参照してください。
- `dbisql` ウィンドウまたはコマンド・ファイルで、`STOP ENGINE` コマンドを発行し、指定されたデータベース・サーバを停止します。
- UNIX の `cron` または `at` ジョブでは、適切な `-stop` オプションを指定して `stop_iq` を使用します。指定されたパラメータに基づいて、`cron` または `at` ジョブを起動するユーザに関連する、1台または複数のサーバが停止されません。ユーザは、サーバを起動したユーザと同じでなければなりません。オペレータに対してメッセージは表示されず、オペレータの操作は不要です。

このようなジョブで `stop_iq` を使用するには、適切な `-stop` オプションを使用してユーティリティを指定します。

```
stop_iq -stop one
```

`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。

```
stop_iq -stop all
```

`-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 によって起動されているすべてのサーバが停止します。

注意 `cron` 文では `stop_iq` 実行プログラムのフル・パス名を指定する必要があります。

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、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、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、ユーザ自身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。例を示します。

```
% stop_iq
Checking system for IQ 15 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-
15_0/logfiles/surya_ibm2.001.srvlog -hn 8

      pamela 409802 18:05:02 0:05 SVR:ibm1_iqdemo2 DB:iqdemo
PORT:2678/ibm1/users/sybase/iq151/IQ-15_2/bin64/iqsrv15 @iqdemo.cfg
iqdemo.db -ti 4400 -gn 25 -o /ibm1/users/sybase/iq151/IQ-
15_2/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/iq151/IQ-15_2/bin64/iqsrv15
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o /myserver/users/sybase/iq151/IQ-
15_2/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、IQ はサーバを停止しません。

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いない場合は、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 - dbisql からのサーバの停止

次の例では、dbisql 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、dbisql からサーバを停止できます。

サーバを停止できるユーザ

サーバを起動する場合、**-gk** オプションを使用して、**DBSTOP** または **STOP ENGINE** でサーバを停止するのに必要なパーミッション・レベルを設定できます。デフォルトのパーミッション・レベルは **DBA** ですが、**ALL** または **NONE** にも設定できます。**NONE** に設定した場合は、**DBA** でも **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設定することをおすすめします。

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、前の項で説明した UNIX コマンド以外のオペレーティング・システ
ム・コマンド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。詳細については、「[第 13 章 システムのリカバリとデータベースの
修復](#)」を参照してください。

注意 サーバを正しく停止し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。サーバを起動するときにデータベースを起動する場合は、「[データベース・
サーバの起動](#)」(21 ページ) を参照してください。

特に運用環境では、1 台のサーバで 1 つのデータベースのみ実行することをお
すすめします。

実行中のサーバ上での データベースの起動

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。詳細については、『リファレンス：
文とオプション』の「[STOP DATABASE 文 \[DBISQL\]](#)」を参照してください。
- Interactive SQL または Sybase Central 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。詳細については、「[ODBC データ・ソース](#)」(69 ページ) を参照してく
ださい。
- システム・コマンド・プロンプトから Interactive SQL を起動するときにデー
タベースを起動し、接続するには、接続パラメータに "DBF=*db-file*" パラ
メータを設定します。詳細については、「[Sybase Central または Interactive
SQL からデモ・データベースへの接続](#)」(60 ページ) を参照してください。
- Sybase Central からデータベースを起動する場合は、『Sybase IQ の概要』の
「[第 4 章 Managing Databases](#)」を参照してください。
- サーバに接続されているときに、DBF パラメータを使用してデータベース
に接続し、埋め込みデータベースを起動します。このパラメータは、新し
く接続するデータベース・ファイルを指定します。このデータベース・
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

次のような問題がデータベースの起動に影響を与えます。

ファイルのアクセス

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する可能性があります。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、Sybase IQ は、エラーをログに出力し、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)。IQ_SYSTEM_TEMP にオープンできるファイルが存在しない場合は、-iqnotemp 起動パラメータを使用しないかぎり、データベースは起動しません。

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することができます。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 を除く) 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、データベース管理者権限を持つユーザのみが IQ データベースを起動できます。ただし、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、DBA 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

データベースは次の方法で停止できます。

- 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。

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

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、データベースを簡単に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。デモ・データベースを作成して起動する方法については、『クイック・スタート』の「[IQ デモ・データベースの作成と使用](#)」を参照してください。

Sybase Central の開始と停止

システムがグラフィカル・ユーザ・インタフェースをサポートしている場合、Sybase Central を使用して多くの管理タスクを実行できます。『Sybase IQ の概要』の「[第 3 章 サーバの起動と接続](#)」の「[Sybase Central の開始と停止](#)」、またはオンライン・ヘルプを参照してください。

この章について

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・サーバにも接続できます。接続オプションを選択するときは、このような要素を考慮する必要があります。

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 Sybase IQ に接続できます。

内容

トピック	ページ
接続の概要	52
Sybase Central または Interactive SQL からの接続	55
Sybase Central での接続を簡略化する方法	59
簡単な接続の例	60
ODBC データ・ソース	69
ODBC データ・ソースの作成と編集	70
ファイル・データ・ソースの使用	78
UNIX での ODBC データ・ソースの使用	79
OLE DB を使用してデータベースに接続する	83
接続パラメータ	86
Sybase IQ の接続の確立方法	87
他のデータベースからの接続	93
サーバを検出できるかどうかのテスト	94
統合化ログインの使用	95
接続の切断と削除	102
接続ログ	102
起動のトラブルシューティング、シャットダウン、接続	103

接続の概要

この章では、クライアント・アプリケーションからデータベースに接続する方法について説明します。ここでは、ODBC、OLE DB、Embedded SQL アプリケーションからデータベースへの接続について解説します。また、Sybase Central および Interactive SQL からの接続についても説明します。

Sybase Open Client™ アプリケーションからデータベースへの接続の詳細については、『システム管理ガイド第 2 巻』の「[第 3 章 データ・サーバとしての Sybase IQ](#)」を参照してください。

JDBC 経由での接続 (Sybase Central または Interactive SQL を使用しない場合) の詳細については、『SQL Anywhere サーバ プログラミング』を参照してください。

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。

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 ])
```

接続の確立方法

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。Sybase IQ でサポートされているインタフェースは次のとおりです。

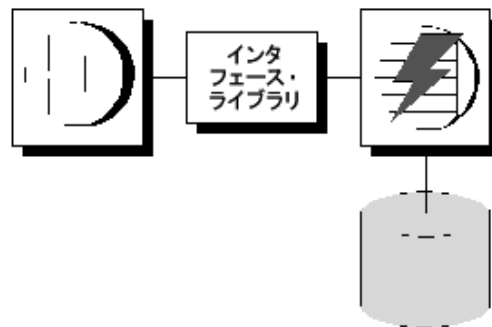
- *ODBC* – ODBC 接続については、この章で説明します。
- *OLE DB* – OLE DB 接続については、この章で説明します。
- *Embedded SQL* – Embedded SQL 接続については、この章で説明します。
- *Sybase Open Client* – Open Client 接続については、この章では説明しません。Open Client アプリケーションから IQ への接続については、『システム管理ガイド第 2 巻』の「[第 3 章 データ・サーバとしての Sybase IQ](#)」を参照してください。

- *JDBC* – *JDBC* 接続については、この章で説明します。*JDBC* 経由での接続については、『Sybase IQ の概要』の「[第4章 Managing Databases](#)」を参照してください。*JDBC* データ・ソースの作成については、『SQL Anywhere サーバ プログラミング』を参照してください。

注意 *JDBC* は、Java オブジェクトの実行とデータベース操作の間のリンクを提供します。Sybase IQ での Java サポートについては、「[データベースでの Java の有効化](#)」(176 ページ)を参照してください。

インタフェースは、クライアント・アプリケーションからの呼び出しに含まれている接続情報を、ファイル・データ・ソース内のディスクに格納されている接続情報と併せて使用して、必要なデータベースを実行しているサーバを見つけて接続します。次の図に、関連する各部分を簡単に示します。

図 3-1: アプリケーションをサーバに接続するインタフェース・ライブラリ



接続に対する理解

目的	参照箇所
Sybase Central または Interactive SQL からの接続の概要 (関連ドライバの説明を含む)	「Sybase Central または Interactive SQL からの接続」 (55 ページ)
すばやく起動する例を知りたい	「簡単な接続の例」 (60 ページ)
概要	「接続パラメータの構造」 (54 ページ)
使用可能な接続パラメータを知りたい	「第4章 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」 (69 ページ)
接続の確立方法について詳しい説明を読みたい	「ODBC データ・ソース」 (69 ページ)
ユーザの追加およびパーミッションの付与	「Sybase IQ の接続の確立方法」 (87 ページ)
ネットワーク特有の接続問題の診断	「ネットワーク通信のトラブルシューティング」 (552 ページ)
接続に関連する文字セットの問題	「接続文字列と文字セット」 (432 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)

接続文字列の表現

この章では、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。

接続文字列として渡される接続パラメータ

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。

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・ソースへの接続パラメータの保存

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、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。

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 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 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。

ODBC データ・ソースの詳細については、「[ODBC データ・ソース](#)」(69 ページ)を参照してください。

Sybase Central または Interactive SQL からの接続

Sybase Central または 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 Sybase Central または Interactive SQL に指示します。

具体的な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、Sybase Central または 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、Sybase Central または 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、Sybase Central および 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。

接続の例 (Sybase Central および Interactive SQL の例を含む) については、「[簡単な接続の例](#)」(60 ページ) を参照してください。

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 `dbisql` 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) で指定できます。完全なリストについては、「[第4章 接続パラメータと通信パラメータ](#)」を参照してください。

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
```

[接続] ダイアログの表示

Sybase Central と Interactive SQL では、データベースに接続するときに共通の [接続] ダイアログ・ボックスを使用します。

Sybase Central を起動したときは、このダイアログ・ボックスを手動で表示する必要があります。Interactive SQL を起動したときは、このダイアログ・ボックスが自動的に表示されます。新しい接続を開始するときは、[File] - [接続] を選択してこのダイアログ・ボックスを開くこともできます。

❖ [接続] ダイアログ・ボックスを開く (Sybase Central)

- Sybase Central で、[接続] - [Sybase IQ 15 に接続] を選択します。

Sybase Central プラグインが複数インストールされている場合は、リストから Sybase IQ を選択します。

メイン・ツールバーの [接続] ボタンをクリックするか、[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。

ヒント

「接続プロファイル」を作成しておくこと、特定のデータベースに簡単にすばやく接続できます。

❖ [接続] ダイアログ・ボックスを開く (Interactive SQL)

- Interactive SQL では、[Window] - [New Window] 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、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] タブでユーザ ID DBA とパスワード `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。

Sybase Central プラグインが複数インストールされている場合は、リストから Sybase IQ を選択します。

接続用のドライバの指定

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。

JDBC ドライバ

Interactive SQL および Sybase Central は、Sybase jConnect™ および iAnywhere JDBC ドライバの 2 種類のドライバをサポートしています。どちらも Sybase IQ に含まれています。

デフォルトでは、Sybase Central IQ プラグインおよび Interactive SQL では、iAnywhere JDBC ドライバが使用されます。jConnect 5.5 ドライバではサポートされていない JDBC 2.0 と完全スクロール可能なカーソルがサポートされています。jConnect は、Web ページのようなゼロ・フットプリントのアプリケーションでも有効です。

JDBC ドライバと必須ソフトウェアの詳細については、『SQL Anywhere サーバプログラミング』の「JDBC ドライバの選択」と「jConnect JDBC ドライバの使用」、および「[ODBC データ・ソース](#)」(69 ページ)を参照してください。

[接続] ダイアログ・ボックスの使用

[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。Sybase Central でも Interactive SQL でも同じダイアログ・ボックスを使用します。

[接続] ダイアログ・ボックスには次のタブがあります。

- [Identification] 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。
- [Advanced] 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の下に表示されます)。データベース名の後ろの角カッコには、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。

Sybase Central での接続を簡略化する方法

この項で説明する手順に従うと、Sybase Central での接続を簡略化できます。

サーバ・オブジェクトの作成

Sybase IQ には、接続情報をすばやく使用するための方法が用意されています。

- IQISQL を使用して接続する
- データベースの起動を単純化する
- Sybase Central からの接続を単純化する

Adaptive Server Enterprise データベースから Sybase IQ データベースに挿入するには、それぞれのサーバの `interfaces` ファイル内に「サーバ・オブジェクト」と呼ばれるエントリが必要です。`interfaces` ファイル内にエントリを作成するには、`IQDSEEDIT` (ディレクトリ・サービス・エディタ) を使用します。`IQDSEEDIT` を実行するには、Sybase ホーム・ディレクトリ (`%SYBASE%`) の所有者であるか、書き込みパーミッションを持っていることが必要です。

このファイルにサーバを追加すると、Sybase Central が接続情報を要求する際、[サーバ名] ドロップダウン・ボックスが有効になります。このドロップダウン・ボックスにタブしてスペース・バーを押すと、`IQDSEEDIT` で作成したエントリがすべてリストされます。使用するサーバの名前をリストから選択するか、サーバ名の 1 文字目のキーを押して選択します。同じ文字のキーを繰り返し押すと、その文字で始まる値が順に選択されます。

接続プロファイルの作成

接続プロファイルを使用すると、個別のユーザがシステムを起動したときに自動的にデータベースに接続したり、接続パラメータを入力しないで接続したりすることができます。接続プロファイルにはユーザ名とパスワードが含まれているため、インストール環境全体で使用するよりも、個人で使用するのに適しています。Sybase Central では、既存のプロファイルのリストから目的のプロファイルを選択したり、プロファイルを作成、編集、削除したり、Sybase Central の起動時にプロファイルを自動的に選択したりできます。

❖ 接続プロファイルの作成

- 1 Sybase Central のメニューから [接続] - [接続プロファイル] を選択するか、[F9] キーを押します。
- 2 [接続プロファイル] ダイアログで [新規作成] をクリックし、プロファイルを作成します。
- 3 [OK] をクリックします。

コンピュータを起動するたびに Sybase Central をこの接続プロファイルで接続する場合は、[接続プロファイル] ウィンドウで [スタートアップの設定] をクリックします (または [Alt] キーを押しながら [S] キーを押します)。

起動時に自動的に接続しない場合は、Sybase Central から手動で接続します。そのためには、[接続] - [接続プロファイル] を選択して [接続] をクリックします。

簡単な接続の例

Sybase IQ の接続モデルは設定によって複雑にもできますが、多くの場合、データベースへの接続は非常に簡単です。

この項では、アプリケーションを Sybase IQ データベースに接続する簡単な事例について説明します。Sybase IQ を使い始めたばかりであれば (ローカル・サーバで `iqdemo` データベースを実行しており、まだネットワークに接続していない場合など)、この項の説明だけで十分と考えられます。ただし、ほとんどの IQ 環境では、接続や切断を正しく行うには、接続パラメータの完全なセットが必須となります。

Sybase Central を使用してデータベースに接続する手順については、『Sybase IQ の概要』を参照してください。使用できる接続パラメータとその使用方法の詳細については、『[接続パラメータ](#)』(86 ページ) を参照してください。

Sybase Central または 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しています。

❖ デモ・データベースへの接続 (Sybase Central)

- 1 システムに適した方法で Sybase Central を起動します。

UNIX の場合、Sybase Central や IQ Agent などのユーティリティを呼び出す前に、`IQ-15_2.csh` (または `.sh`) スクリプトを用意します。

マルチプレックス環境では、IQ Agent が起動しない場合は次のように入力します。

```
%S99SybaseIQAgent15
```

Sybase Central を起動するには、次のように入力します。

```
% scjview
```

注意 『インストールおよび設定ガイド』の説明どおりに環境変数を設定した場合は、どのディレクトリからでも `scjview` コマンドを発行できます。

Windows の場合、Sybase Central を起動するには、[プログラム] - [Sybase] - [Sybase IQ 15.2] - [Sybase Central] を選択します。

- 2 [Sybase IQ] を選択します。
複数のタブを含んだパネルが右側に表示されます。
- 3 [ユーティリティ] タブで [Interactive SQL を開く] をダブルクリックします。
- 4 [ID] タブで、[ユーザ] と [パスワード] に DBA と sql を入力します。
- 5 [データベース] タブで、[検索] を選択します。
- 6 [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。

❖ デモ・データベースへの接続 (Interactive SQL)

- 1 [プログラム] - [Sybase] - [Sybase IQ 15.2] - [Interactive SQL] を選択します。
- 2 前の手順の手順4～6に従います。

実行中のどのデータベースにも同じ方法で接続できます。デフォルト以外の文字セットと言語も指定できます。

dbisql の使用方法の詳細については、『ユーティリティ・ガイド』の「[第2章 Interactive SQL \(dbisql\) の使用](#)」を参照してください。

Sybase Central または Sybase IQ から同じマシン上のデータベースに接続する

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るか。実行中の場合は、[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済みます。実行されていない場合は、そのデータベースを Sybase Central または Interactive SQL に起動させるために、データベース・ファイルを指定する必要があります。
- 同じマシン上で複数のデータベースを実行しているか。複数のデータベースが実行中の場合は、Sybase Central または Interactive SQL の接続先となるデータベースを指定する必要があります。データベースが1つだけの場合は、そのデータベースに接続するものと自動的に判断されるため、[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がありません。

実際の手順は、上記の質問に対する答えによって変わってきます。

❖ 実行中のローカル・サーバ上のデータベースへの接続

- 1 Sybase Central または 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 このダイアログ・ボックスの [ID] タブで、ユーザ ID とパスワードを入力します。

- 3 次のいずれかを実行します。
 - サーバ上のデータベースが 1 つだけの場合は、[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、このダイアログ・ボックスの [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。

❖ **まだ実行されていないデータベースへの接続**

- 1 Sybase Central または 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 このダイアログ・ボックスの [ID] タブを表示し、ユーザ ID とパスワードを入力します。
- 3 このダイアログ・ボックスの [データベース] タブをクリックします。
- 4 [Database File] フィールドにファイルを指定します (フル・パス、名前、拡張子を含めます)。[Browse] をクリックしてファイルを検索することもできます。
- 5 以降の接続で、このファイル名とは異なる名前をデータベース名として使用したい場合は、[Database Name] フィールドに名前を指定します (パスまたは拡張子を含めません)。

ヒント

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・ファイルを指定する必要はありません。

次の項も参照してください。

- [「\[接続\] ダイアログの表示」 \(56 ページ\)](#)
- [「簡単な接続の例」 \(60 ページ\)](#)

Interactive SQL から他のデータベースへの接続

dbisql から実行中のデータベースに接続するには、次の手順に従います。

❖ **UNIX 上の Interactive SQL からデータベースへの接続**

- 1 システム・コマンド・プロンプトに対して次のように入力し、サーバとデータベースを起動します。

```
start_iq dbname
```

- システム・コマンド・プロンプトで次のように入力して、Interactive SQL を起動します。

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

-c パラメータは、接続パラメータを指定します。接続パラメータの詳細については、「[接続パラメータ](#)」を参照してください。

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 $IQDIR15/demo/iqdemo.db
```

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。

コマンド・ライン・ユーティリティを使用して接続する

UNIX システム上のコマンド・ラインから実行中のデータベースに接続するには、次の手順に従います。

❖ UNIX システムからの接続

- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第1章 ファイル・ロケーションとインストール設定](#)」の説明に従って、PATH やその他の環境変数が正しく設定されていることを確認します。
- 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
```

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。

```
cd $IQDIR15/demo
start_iq @iqdemo.cfg iqdemo.db
```

- 次のように入力して、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。

-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・ソースに指定できます。

注意 パラメータ `links=tcPIP` (または `CommLinks=tcPIP`) 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(host=SERV2;port=1234) "
```

次のように別の `links` 句を使用しても同じ結果を得ることができます。

```
"links=tcPIP(host=SERV2:1234) "
```

❖ Windows システムからの接続

- 1 [プログラム]メニューから [Sybase] - [Sybase IQ 15.2] - [Interactive SQL] を選択します。または、Windows のコマンド・プロンプトで次のように入力します。

```
dbisql
```

UNIX への接続手順ですでに説明したように、`dbisql` コマンドで `-c` パラメータを使用すると接続パラメータを指定できます。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` (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。このポート番号を調べるには、[Sybase IQ 15.2] - [ODBC Administrator 32 ビット版] または [Sybase IQ 15.2] - [ODBC Administrator 64 ビット版] を実行します。[ユーザー DSN] タブで [ユーザー・データ・ソース] を選択し、[設定] をクリックします。ポート番号を調べるには、コマンド・プロンプトに 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、[検索] 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。

情報不足のために [接続] ダイアログ・ボックスまたはエラー・メッセージが表示された場合は、-host と -port を入力するか、[詳細] タブで不足情報を入力します。目的のデータベースがリモート・サーバ上にある場合は、次のように -host パラメータと -port パラメータをそれぞれ別々の行に入力します。

```
-host fiona  
-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。Interactive SQL ウィンドウでは、タイトル・バーにその接続のデータベース名、ユーザ ID、サーバ名が表示されます。

dbisqlc 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埋め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。この場合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) パラメータにデータベースのファイルを指定します。

DBF パラメータの使用

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断すると)、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は、デモ・データベースを埋め込みデータベースとしてロードする方法を示します。

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

path は、Sybase IQ インストール・ディレクトリの名前です。

StartLine (Start) パラメータの使用

次の接続パラメータは、埋め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)**

- 1 データベースを実行していない状態で Interactive SQL 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 15.2] - [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。次に、[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-15_2 の場合、次のように入力します。

```
c:¥sybase¥IQ-15_2¥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。Sybase IQ が起動してデモ・データベースをロードし、Interactive SQL がデータベースに接続します。

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。dbisqlc などの Embedded SQL アプリケーションと同様に、ODBC および JDBC は、iAnywhere JDBC ドライバを使用してデータ・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。詳細については、「[ODBC データ・ソースの作成と編集](#)」(70 ページ)を参照してください。

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。詳細については、「[接続用のドライバの指定](#)」(57 ページ)を参照してください。

❖ Sybase Central または Interactive SQL からデータ・ソースを使用して接続する

- 1 Sybase Central または 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] タブでユーザ ID とパスワードを入力します (たとえば DBA と sql)。
- 3 [ID] タブの下半分で、次のいずれかを実行します。
 - [ODBC Data Source Name] 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)。[Browse] 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)。[検索] をクリックしてファイルを検索することもできます。

Sybase IQ デモ・データ・ソースには、データベース・ファイルや 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。

注意 次のように dbisql コマンドで 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"
```

iqdemo データ・ソース

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。

デフォルト接続パラメータの使用

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 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場合

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
```

デフォルト動作の詳細については、「[Sybase IQ の接続の確立方法](#)」(87 ページ)を参照してください。

Sybase IQ 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 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。このようなデータベース・ユーティリティは、データベースに接続するときに、「[Sybase IQ の接続の確立方法](#)」(87 ページ)で示されている手順に従います。

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。

- 1 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 およびパスワード 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とります。

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

- 2 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』の「[第1章 ファイル・ロケーションとインストール設定](#)」を参照してください。
- 3 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』の「[第4章 データベース管理ユーティリティ](#)」を参照してください。

ODBC データ・ソース

一連の Sybase IQ 接続パラメータを、データ・ソースという形でシステム・レジストリまたはファイルに保存できます。データ・ソースは、ODBC (Open Database Connectivity) インタフェースを使用して接続するアプリケーションを使用するのに必要です。

Microsoft によって定義された ODBC インタフェースは、Windows 環境でクライアント・アプリケーションをデータベース管理システムに接続するための標準インタフェースです。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、ODBC インタフェースを使用して、さまざまなデータベース・システムにアクセスします。

データ・ソースの保存場所

データ・ソースを Windows または UNIX/Linux ベース・システムで使用して、接続を簡単に確立できるようにします。

ODBC を使用してデータベースに接続する場合、ODBC データ・ソースを使用します。データ・ソースには、一連の接続パラメータが格納されています。クライアント・コンピュータ上には、すべての接続先データベースに関する ODBC データ・ソースが必要です。

データ・ソースがある場合、接続文字列には目的のデータ・ソースを指定するだけで済みます。

- **データ・ソース** ユーザ・データ・ソースまたはシステム・データ・ソースを参照するには、DSN 接続パラメータを使用します。

```
DSN=my data source
```

Windows では、ユーザ・データ・ソースとシステム・データ・ソースはレジストリと *odbc.ini* ファイルに保存されます。UNIX プラットフォームでは、ユーザ・データ・ソースは *.odbc.ini* ファイルに保存されます。

- **ファイル・データ・ソース** ファイル内に保存されているデータ・ソースを参照するには、FileDSN 接続パラメータを使用します。

```
FileDSN=mysource.dsn
```

暗号化されたパスワードは FileDSN のみの機能ですが、その他は、DSN と FileDSN にまったく同じ接続情報を保存できます。

ODBC データ・ソースを使用すると、さまざまなクライアント接続が可能です。

- ODBC インタフェースを使用する Windows、Linux、および Unix 上のアプリケーション
- iAnywhere JDBC ドライバを使用する Java アプリケーション

ODBC データ・ソースの作成と編集

Windows では、ODBC アドミニストレータを使用して、ODBC データ・ソースを集中的に作成、管理できます。

注意 32 ビット ODBC ドライバを使用する場合は 32 ビット ODBC アドミニストレータを、64 ビット ODBC ドライバを使用する場合は 64 ビット ODBC アドミニストレータを使用してデータソースを管理します。

次の手順では、ODBC アドミニストレータを使用して、既存の *odbc.ini* に新しいデータ・ソースを追加したり、必要に応じて新しいファイルを作成したりします。

Sybase IQ には、データ・ソースを作成するための *iqdsn* というクロスプラットフォームのコマンド・ライン・ユーティリティが含まれています。

UNIX システムでの ODBC データ・ソースの管理方法については、「UNIX での ODBC データ・ソースの使用」(79 ページ) を参照してください。

作業を始める前に

この項では、ODBC データ・ソースの作成方法を説明します。データ・ソースを作成する前に、どの接続パラメータをデータ・ソースに含めるかを把握する必要があります。

詳細については、「簡単な接続の例」(60 ページ) と「接続パラメータ・リストのアセンブル」(89 ページ) を参照してください。

ODBC アドミニストレータでは、ユーザ・データ・ソース、ファイル・データ・ソース、システム・データ・ソースを扱うことができます。

ODBC アドミニストレータからのデータ・ソースの作成

❖ ODBC データ・ソースの作成 (ODBC アドミニストレータ)

- 1 次の方法で ODBC アドミニストレータを起動します。

[設定] - [コントロール パネル] - [管理ツール] - [データ ソース (ODBC)] を選択します。

または

[Sybase IQ 15.2] - [ODBC Administrator 32 ビット版] を選択します。

[ODBC データ ソース アドミニストレータ] が表示されます。

- 2 [追加] をクリックします。

[データ ソースの新規作成] ウィザードが表示されます。

- 3 ドライバのリストから [Sybase IQ] を選択し、[完了] をクリックします。

- 4 [ODBC 設定] ウィンドウで [データ・ソース名] を入力します。

- 5 ここで、[Login] タブをクリックします。データベースのユーザ ID とパスワードを入力します。たとえば、DBA と sql を使用します。

- 6 [データベース] タブをクリックします。データ・ソースがローカル・マシン上にある場合は、サーバ名、開始行、データベース・ファイル (パスを含む) を入力します。

- 7 データ・ソースがリモート・システム上にある場合は、[ネットワーク] タブをクリックします。適切なプロトコルのチェック・ボックスをクリックし、その横のオプションを指定します。たとえば、TCP/IP プロトコルとポート 1870 を使用してシステム PUSHKIN 上のサーバに接続するには、[TCP/IP] をクリックして次のように入力します。

```
host=pushkin:1870
```

ホストのネットワーク・アドレスを使用することもできます。例を示します。

```
host=157.133.66.75:1870
```

- 8 データ・ソースの定義が完了したら、[OK] をクリックします。

[ODBC データ ソース アドミニストレータ] で [ユーザー DSN] タブに戻ります。

[ODBC 設定] ダイアログ・ボックスとタブの詳細については、「[ODBC データ・ソースの設定](#)」(73 ページ) を参照してください。

注意 ネットワーク接続の指定では、データベース・サーバごとに異なる *systemname:port#* の組み合わせが必要です。ポート番号は、サーバの起動時に使用した番号と一致している必要があります。

コマンド・ラインからの ODBC データ・ソースの 作成

ユーザ・データ・ソースおよびシステム・データ・ソースは、*iqdsn* コマンド・ライン・ユーティリティを使用して作成できます。*iqdsn* ではファイル・データ・ソースを作成できません。ユーザ・データ・ソース、システム・データ・ソース、ファイル・データ・ソースを作成するには ODBC アドミニストレータも使用できます。

❖ ODBC データ・ソースの作成 (コマンド・ライン)

- 1 コマンド・プロンプトを開きます。
- 2 *iqdsn* コマンドを入力し、使用する接続パラメータを指定します。たとえば、次のコマンドでは Sybase IQ デモ・データベース用のデータ・ソースを作成します。このコマンドは 1 行に入力します。

```
iqdsn -w "My DSN"  
"uid=DBA;pwd=sql;dbf=c:\Program Files\Sybase\IQ-  
15_2\demo\iqdemo.db"
```

iqdsn の出力には、次の行が含まれます。

```
User Data Source "My DSN" written to registry.
```

iqdsn ユーティリティは、Windows コマンド・ラインで作成した Sybase IQ のデータ・ソースをリストします。

iqdsn ユーティリティの詳細については、『ユーティリティ・ガイド』の「[データ・ソース・ユーティリティ \(iqdsn\)](#)」を参照してください。

❖ ODBC データ・ソースのテスト

- 1 データベースを起動します。たとえば、デモ・データベースを起動するには、[プログラム] - [Sybase] - [Sybase IQ 15.2] - [Sybase IQ デモ・データベースの起動] を選択します。
- 2 [ODBC データ ソース アドミニストレータ] で、[ユーザー DSN] のリストから新しいデータ・ソースを選択します。

- 3 [設定] をクリックします。
- 4 [ODBC Configuration] ダイアログ・ボックスで、[Test Connection] をクリックします。

データ・ソースにアクセスできない場合は、それぞれのタブに正しいファイル名とパス名が指定されていることを確認します。

データ・ソースを編集するには、ODBC アドミニストレータのウィンドウのリストからデータ・ソースを選択し、[設定] をクリックします。

ODBC データ・ソースを作成するためにネットワーク経由で Windows にアクセスする場合は、『インストールおよび設定ガイド』を参照してください。

ODBC データ・ソースの設定

この項では、[ODBC 設定] ダイアログ・ボックスで各タブにまとめられているオプションの機能について説明します。

[ODBC] タブ

[データ・ソース名] ODBC データ・ソースを識別するためのデータ・ソース名です。データ・ソースには説明的な名前 (スペースも使用可能) を使用できますが、接続文字列で指定することを考慮して短い名前にすることをおすすめします。

詳細については、「[DataSourceName 接続パラメータ \(DSN\)](#) (120 ページ) を参照してください。

[説明] データ・ソースの説明を詳細に入力できます。これは省略可能です。

[トランスレータ] データベースで OEM コード・ページを使用している場合は、[MS Code Page Translator] を選択します。データベースで、デフォルトの ANSI コード・ページを使用している場合は、選択しないでください。

[独立性レベル] Sybase IQ データ・ソースの独立性レベルは、実際は常に 3 です。ただし、デフォルトのカatalog・ストアの独立性レベルは 0 です。通常は、これを空白のままにします。

詳細については、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

[Microsoft アプリケーション (SQL Statistics のキー)] SQL 統計で外部キーを返す場合は、このボックスをオンにします。ODBC 仕様では、プライマリ・キーと外部キーを SQL 統計で返さないように規定しています。ただし、Microsoft の一部のアプリケーション (Visual Basic や Access など) では、プライマリ・キーと外部キーを SQLStatistics で返します。

[Delphi アプリケーション] Borland Delphi アプリケーションのパフォーマンスを向上させるには、このボックスをオンにします。このオプションをオンにすると、各ローにブックマーク値が1つ割り当てられます。オンにしない場合は、2つのブックマーク値が割り当てられます(1つは前方フェッチ用、もう1つは後方フェッチ用です)。

Delphi では、1つのローに割り当てられた複数のブックマーク値を処理できません。このオプションをオフにすると、スクロール可能なカーソルのパフォーマンスが低下します。これは、正しいブックマーク値を取得するために、通常、カーソルの先頭から要求対象のローまでスクロールが行われるためです。

[フェッチ警告を表示しない] このボックスをオンにすると、フェッチ時にデータベース・サーバから返される警告メッセージが表示されなくなります。

[ドライバに起因するエラーを回避] Sybase IQ 15 ODBC ドライバは修飾子をサポートしないため、Driver not capable というエラーを返します。一部の ODBC アプリケーションは、このエラーを適切に処理しません。このエラーを無効にしてアプリケーションを動作させるには、このボックスをオンにします。

[文が完結するまでオート・コミットしない] Sybase IQ 15 ODBC ドライバを使用して、文が完結するまでコミット操作を遅延させる場合は、このボックスをオンにします。

[カーソル動作の記述] 1つのプロシージャを実行または再開する場合にカーソルを記述し直す頻度を選択します。

[テスト接続] 入力された情報で正しく接続されるかどうかをテストします。テストを実行するには、ユーザ ID とパスワードを入力する必要があります。

[Login] タブ

統合化ログインの使用 統合化ログインを使用した接続。ユーザ ID とパスワードを指定する必要はありません。その代わりに、オペレーティング・システムのユーザ ID とパスワードを Sybase IQ 統合化ログイン・メカニズムに提供します。このログインを使用するには、ユーザに統合化ログイン・パーミッションが付与されている必要があります。接続先のデータベースは、統合化ログインを受け入れるように設定されている必要があります。DBA アクセス権を所有しているユーザのみが統合化ログイン・パーミッションを管理できます。

詳細については、「[統合化ログインの使用方法](#)」(96 ページ)を参照してください。

[ユーザ ID] 接続に使用するユーザ ID を入力します。

詳細については、「[Userid 接続パラメータ \(UID\)](#)」(132 ページ)を参照してください。

[パスワード] 接続に使用するパスワードを入力します。

詳細については、「[Password 接続パラメータ \(PWD\)](#)」(129 ページ)を参照してください。

[パスワードの暗号化] このボックスをオンにすると、パスワードが暗号化された形式でプロファイルに保存されます。

詳細については、「[EncryptedPassword 接続パラメータ \(ENP\)](#)」(122 ページ)を参照してください。

[Database] タブ

[サーバ名] IQ サーバ名を入力する場所です。

詳細については、「[EngineName 接続パラメータ \(ENG\)](#)」(121 ページ)を参照してください。

[開始行] 起動するサーバを入力します。接続先のデータベース・サーバが現在実行されていないときにのみ、[開始行]パラメータを指定します。例を示します。

```
C:\Program Files\Sybase\IQ-15_2\win32\start_iq.exe
-gm 10 -gp 4096 -c 32M
dbf=path\iqdemo.db
uid=DBA pwd=sql
```

詳細については、「[StartLine 接続パラメータ \(START\)](#)」(131 ページ)を参照してください。

[データベース名] 接続先の Sybase IQ データベースの名前を入力する場所です。

詳細については、「[DatabaseName 接続パラメータ \(DBN\)](#)」(118 ページ)を参照してください。

[データベース・ファイル] サーバ・マシン上の Sybase IQ データベース・ファイルのフル・パスと名前を入力する場所です。[Browse] をクリックしてファイルを検索できます。例を示します。

```
C:\Program Files\Sybase\IQ-15_2\demo\iqdemo.db
```

詳細については、「[DatabaseFile 接続パラメータ \(DBF\)](#)」(117 ページ)を参照してください。

[暗号化キー] データベースが強力な方法で暗号化されている場合は、そのデータベースにアクセスするには正しい暗号化キーを指定する必要があります。

詳細については、「[Encryption 接続パラメータ \(ENC\)](#)」(122 ページ)を参照してください。

[データベースを自動起動] これを選択すると、新しいセッションの開始時に、目的のデータベースがまだ実行されていない場合は自動的に開始します。データベース・ファイルまたは開始行を指定した場合は、このオプションを有効にします。

[最終切断後にデータベースを停止] これを選択すると、最後のユーザの切断後にサーバが自動的に停止します。

[ネットワーク] タブ

[ネットワーク プロトコルおよびオプションの選択] これらのチェック・ボックスでは、ODBC DSN がネットワーク・データベース・サーバにアクセスするとき使用するプロトコルを指定します。隣のボックスには、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チューニングするための通信パラメータを入力できます。

- **[TCP/IP]** ネットワーク・パケットに ECC_TLS (Certicom) または RSA_TLS 暗号化を使用したい場合は、ネットワーク・データベース・サーバへのアクセスに TCP/IP プロトコルを選択します。

たとえば、TCP/IP 接続に関して `HOST=my_server;PORT=4563` という通信パラメータを入力できます。

- **[共有メモリ]** 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。

TCP/IP の例については、「[ODBC データ・ソースの作成 \(ODBC アドミニストレータ\)](#)」(71 ページ)を参照してください。

詳細については、「[第4章 接続パラメータと通信パラメータ](#)」の「CommLinks 接続パラメータ」と「ネットワーク通信のパラメータ」を参照してください。

[活性タイムアウト] 活性パケットは、接続が維持されていることを確認するためにクライアント／サーバを介して送信されます。クライアントが活性パケットを検出しない時間が一定の活性タイムアウト時間を超えると、通信が切断されます。このパラメータは、ネットワーク・サーバと TCP/IP 通信プロトコルを使用した場合のみ作用します。デフォルトは 120 秒です。

詳細については、「[LivenessTimeout 接続パラメータ \(LTO\)](#)」(127 ページ)を参照してください。

[アイドル・タイムアウト] 接続を終了させるまでのクライアントのアイドル時間の長さを設定します。クライアントが要求を送信しない時間が一定のアイドル・タイムアウト時間を超えると、接続が切断されます。

デフォルトのクライアント・アイドル時間は `-ti` サーバ・オプションで設定します。`start_iq` でのデフォルト設定は 4400 分です。

詳細については、「[Idle 接続パラメータ \(IDLE\)](#)」(125 ページ)を参照してください。

[バッファ・サイズ] 通信パケットの最大サイズをバイトで設定します。デフォルトのバッファ・サイズは 1460 です。

詳細については、「[CommBufferSize 接続パラメータ \(CBSIZE\)](#) (114 ページ) を参照してください。

[ネットワーク・パケットの暗号化の方法の選択] クライアント・マシンからネットワークに送信されるパケットを暗号化します。

- **[なし]** クライアントから送信される通信パケットを暗号化しません。デフォルトの設定です。
- **[単純]** クライアントから送信される通信パケットを単純暗号で暗号化します。単純暗号はすべてのプラットフォームでサポートされ、以前のバージョンの Sybase IQ でもサポートされています。
- **[TLS]** クライアントから送信される通信パケットを ECC_TLS (以前の Certicom) 暗号で暗号化します。これは強力な暗号の一種です。TCP/IP プロトコルでのみ使用できます。

[編集] をクリックし、[TLS 暗号化オプション] 画面を完了します。

詳細については、「[Encryption 接続パラメータ \(ENC\)](#) (122 ページ) を参照してください。

[詳細] タブ

このタブには、特殊な目的に使用される、大部分のユーザには必要ない高度なフィールドが含まれています。

[接続名] これから作成する接続の名前です。

[文字セット] 文字セットを指定します。文字セットとは、国または言語に固有の 256 個の文字、数字、記号のセットです。Microsoft Windows では ANSI 文字セットが使用されます。OEM 文字セットは、ANSI 文字セットを除く任意の文字セットです。

[複数のレコード・フェッチを許可] レコードを 1 つずつ取り出すのではなく、複数のレコードを一度に取り出せるようにします。デフォルトでは、複数のレコードのフェッチができます。

[ログ・ファイルにデバッグ情報を表示] デバッグ情報を保存するファイルの名前です。

[追加接続パラメータ] 追加のオプションを入力します。これらのパラメータよりも、このダイアログ・ボックスで設定する他のパラメータの方が優先されます。

ファイル・データ・ソースの使用

Windows オペレーティング・システムでは、ODBC データ・ソースは通常はシステム・レジストリに保存されます。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。

Windows のファイル・データ・ソースは、通常は *.dsn* という拡張子を持ちます。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、暗号化する場合を除いてパスワードを 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す。

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 ファイル・データ・ソースの作成 (ODBC アドミニストレータ)

- 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 [Sybase IQ 12] を選択し、[Next] 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。

テキスト・エディタでのファイル・データ・ソースの作成

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
```

『SQL Anywhere サーバ - データベース管理』>「データベースの管理」>「データベース管理ユーティリティ」>「データ・ソース・ユーティリティ (dbdsn)」を参照してください。

UNIX での ODBC データ・ソースの使用

UNIX オペレーティング・システムでは、ODBC データ・ソースは `.odbc.ini` というファイルに保存されます。UNIX システムで `.odbc.ini` ファイルを作成する場合は、次のように各識別子の長い形式を使用してください。

```
[My Data Source]
EngineName=myserver
CommLinks=tcpip(port=1870)
UserId=DBA
Password=sql
```

`.odbc.ini` ファイルには、任意の接続パラメータを入力できます。ネットワーク通信パラメータは `CommLinks (LINKS)` パラメータの一部として追加します。接続パラメータの完全なリストについては、「[ネットワーク通信のパラメータ \(132 ページ\)](#)」を参照してください。

UNIX 上で ODBC データ・ソースを作成および管理するには、`iqdsn` コマンド・ライン・ユーティリティを使用します。詳細については、「[コマンド・ラインからの ODBC データ・ソースの作成 \(72 ページ\)](#)」を参照してください。

ファイル・ロケーション

ODBC 関数への参照は、実行時に解決されます。

ODBC データ・ソースに接続するには、`.odbc.ini` ファイルのロケーションが次の変数のいずれかで参照される必要があります。Sybase IQ は、変数で指定されるディレクトリを次の順序で検索します。

- 1 `$ODBCINI` - `.odbc.ini` ファイルの正確なフル・パス名を格納している必要があります。
- 2 `$HOME`
- 3 現在のディレクトリ
- 4 `$PATH`

Sybase IQ クライアントは、`.odbc.ini` の検索時に次の変数を無視します。

- 1 `$ODBC_HOME`
- 2 `$ODBC_INI`

テキスト・エディタで `.odbc.ini` ファイルを編集し、データ・ソースのエントリを追加する必要があります。

UNIX システムでは、Sybase IQ は ODBC ドライバだけをインストールし、ドライバ・マネージャはインストールしません。ドライバ・ファイル名には、オペレーティング・システム固有の拡張子が付いています。たとえば、Solaris システムの場合は *so* です。*libodbc.so* (*libodbc.so.1*) または *libodbcinst.so* (*libodbcinst.so.1*) を使用する ODBC アプリケーションを使用している場合は、それらに対して、*\$\$SYBASE/IQ-15_2/lib/libdbodbc11.so.1* を指すシンボリック・リンクを作成します。カスタム ODBC アプリケーションを作成する場合は、*dbodbc11.so* に直接リンクできます。

ODBC ドライバ・マネージャがない場合、IQ ODBC ドライバ (シンボリック・リンク経由で検出) は、*.odbc.ini* を使用してデータ・ソース情報を取得します。

プラットフォームに適したドライバを使用しているかどうかを確認するには、『SQL Anywhere サーバ プログラミング』の「第 7 章 ODBC プログラミング」を参照してください。

ODBC トレース出力の有効化

ODBC トレース・ファイルを作成するには、『SQL Anywhere サーバ データベース管理』>「SQL Anywhere データ・アクセス API」>「SQL Anywhere ODBC API」>「ODBC アプリケーションの構築」>「UNIX での SQL Anywhere ODBC ドライバ・マネージャの使用」を参照してください。

UNIX システムおよび Linux システムでは、*libdbodbc11.so* ドライバを使用すると、このドライバによってマルチスレッド・ドライバまたは非スレッド・ドライバが選択されます。トレース機能は、個々のドライバ (*libdbodbc11_n.so* または *libdbodbc11_r.so*) ではなくスイッチ (*libdbodbc11.so*) に存在します。*_r* version を示すドライバを変更する場合、呼び出しシーケンスからスイッチを削除して、トレースを禁止します。

iAnywhere Solutions Oracle ドライバの使用

iAnywhere Solutions 11 - Oracle ODBC ドライバは、Sybase IQ または iAnywhere 専用に作成されたドライバです。このドライバは、サード・パーティ製ソフトウェアでは動作しません。

OMNI で Oracle を使用する場合は、この Oracle ドライバと同じコンピュータに Oracle クライアントをインストールする必要があります。

Oracle ドライバを設定するには、ODBC Administrator、*.odbc.ini* ファイル (UNIX の場合)、または *iqdsn* ユーティリティを使用します。

表 3-1 は、Oracle ドライバの設定オプションを示します。

表 3-1: *iAnywhere Oracle* ドライバの設定オプション

Windows ODBC Administrator	iqdsn コマンドまたは .odbc.ini ファイルの設定	説明
[Data Source Name]	iqdsn では、-w オプションを使用します。	データ・ソースを識別する名前。
[ユーザ ID]	iqdsn では、接続文字列で UserID オプションを設定します。	アプリケーションが Oracle データベースへの接続に使用するデフォルトのログイン ID。このフィールドをブランクのままにした場合は、接続時に情報を入力する必要があります。
[パスワード]	iqdsn では、接続文字列でこのオプションを設定します。	アプリケーションが Oracle データベースへの接続に使用するパスワード。このフィールドをブランクのままにした場合は、接続時に情報を入力する必要があります。
[SID]	SID	Oracle インストール・ディレクトリの <i>network/admin/tnsnames.ora</i> に保存されている TNS サービス名。
[Enable Microsoft Distributed Transactions]	iqdsn では、接続文字列で enableMSDTC オプションを使用します。 .odbc.ini ではサポートされません。	Microsoft 分散トランザクション・コーディネータにトランザクションを登録する場合は、このオプションを選択します。このオプションを選択した場合、Oracle ODBC ドライバには Oracle バイナリ・ファイル (Oracle 9i クライアントでは <i>oramts.dll</i> 、Oracle 10g クライアントでは <i>oramts10.dll</i>) が必要です。
[パスワードの暗号化]	iqdsn では、-pe オプションを使用します。 .odbc.ini ではサポートされません。	データ・ソースに暗号化された形式でパスワードを保存する場合は、このオプションを選択します。

Windows ODBC Administrator	iqdsn コマンドまたは .odbc.ini ファイルの設定	説明
[Procedures Return Results]	iqdsn では、接続文字列で ProcResults オプションを設定します。	ストアド・プロシージャが結果を返すことができる場合は、このオプションを選択します。デフォルトでは、プロシージャは結果を返しません (このオプションは選択されていません)。download_cursor スクリプトまたは download_delete_cursor スクリプトがストアド・プロシージャ呼び出しの場合は、このオプションを yes に設定します。
[Array Size]	iqdsn では、接続文字列で ArraySize オプションを設定します。	ローのプリフェッチに使用するバイト配列のサイズ (バイト単位)。文ごとに指定します。デフォルトは 60000 です。この値を増やすと、フェッチのパフォーマンスを大幅に向上させることができますが、追加のメモリを割り付ける必要があります。

Windows の設定

❖ Windows での Oracle ドライバ用 DSN の作成

- 1 [スタート]-[プログラム]-[Sybase]-[Sybase IQ 15.2]-[ODBC Administrator 32 ビット版] を選択します。
- 2 [追加] をクリックします。
- 3 [iAnywhere Solutions 11 - Oracle] を選択し、[完了] をクリックします。
- 4 設定オプションを指定します。
- 5 [テスト接続] をクリックし、[OK] をクリックします。

UNIX 設定

UNIX では、ODBC システム情報ファイル (通常は `.odbc.ini`) でドライバを設定する場合、このドライバのセクションは次のように表示されます (各フィールドには適切な値が設定されます)。

```
[sample_dsn_using_the_ias_odbc_driver_for_oracle]
Driver=full-path/libdboraodbc10_r.so
UserID=user-id
Password=password
SID=TNS-service-name
ProcResults=[yes|no]
ArraySize=bytes
```

各フィールドについては、[表 3-1](#) を参照してください。

IQDSN 設定

iqdsn ユーティリティを使用して Oracle DSN を作成するには、次の構文を使用します。

```
iqdsn -w data-source-name -or -c configuration-options
```

設定オプションについては、[表 3-1](#) を参照してください。

例を示します。

```
iqdsn -w MyOracleDSN -or -pe -c
Userid=dba;Password=sql;SID=abcd;ArraySize=100000;ProcResults
=y;enableMSDIC=n
```

OLE DB を使用してデータベースに接続する

OLE DB は、COM (Component Object Model) を使用することで、アプリケーションがさまざまなソースのデータを利用できるようにします。リレーショナル・データベースは、OLE DB を通じてアクセスできるデータ・ソースの 1 つです。

この項では、次の環境で OLE DB を使用して Sybase IQ データベースに接続する方法を説明します。

- Sybase PowerBuilder® が OLE DB データ・ソースにアクセスでき、Sybase IQ を PowerBuilder OLE DB データベース・プロファイルとして使用できる環境。
- Microsoft ADO (ActiveX Data Object) が OLE DB データ・ソース用のプログラミング・インタフェースを提供している環境。Microsoft Visual Basic などのプログラミング・ツールから Sybase IQ にアクセス可能。

OLE DB には Windows クライアントが必要です。ただし、OLE DB を使用すれば Windows サーバと UNIX サーバの両方にアクセスできます。

この項では、Sybase PowerBuilder 環境と Microsoft ADO 環境 (Visual Basic など) から OLE DB を使用する方法の概要を示します。ここでは、ADO または OLE DB を使用したプログラミングについて完全に網羅しているわけではありません。開発トピックの詳細については、開発ツールのマニュアルを参照してください。

OLE DB については、『SQL Anywhere サーバ - プログラミング』を参照してください。

注意 Sybase IQ では、OLE DB で使用される一部機能に対するサポートが、SQL Anywhere のサポートとは異なっています。

- Sybase IQ では、Windows CE はサポートされません。
 - Sybase IQ では、カーソルを通じたりモート・アップデートはサポートされません。
 - Sybase IQ では、動的 (動的スクロール) カーソル、静的 (無反応) カーソル、前方スクロールのみ (スクロールなし) カーソルがサポートされますが、キーセット (スクロール) カーソルはサポートされません。
 - Sybase IQ では、独立性レベルは何を指定しても必ず 3 になります。
-

OLE DB プロバイダ

アクセスするデータ・ソースの種類ごとに、「OLE DB プロバイダ」が必要です。各プロバイダはダイナミック・リンク・ライブラリです。Sybase IQ にアクセスするには、次の 2 つの OLE DB プロバイダを使用できます。

- **Sybase SQL Anywhere OLE DB プロバイダ** SQL Anywhere OLE DB Provider は、ODBC コンポーネントを使用せずに、OLE DB データ・ソースとして Sybase IQ へのアクセスを実現します。このプロバイダの短い名前は SAOLEDB です。

SAOLEDB プロバイダは、インストール時に自身を登録します。この登録プロセスでは、レジストリの COM セクションにレジストリ・エントリを作成し、SAOLEDB プロバイダが呼び出されたときに ADO がこの DLL を見つけられるようにします。DLL のロケーションを変更した場合は、そのロケーションを再登録する必要があります。

SQL Anywhere OLE DB プロバイダを使用する場合は、配備時に ODBC は必要ありません。

OLE DB プロバイダの詳細については、『SQL Anywhere サーバ - プログラミング』を参照してください。

- **Microsoft OLE DB Provider for ODBC** Microsoft が提供する OLE DB プロバイダです。短い名前は MSDASQL です。

MSDASQL プロバイダは、ODBC データ・ソースを OLE DB データ・ソースのように見せます。このプロバイダには SQL Anywhere ODBC ドライバが必要です。

ADO からの接続

ADO はオブジェクト指向のプログラミング・インタフェースです。ADO では、データ・ソースとのユニーク・セッションを **Connection** オブジェクトで表します。

接続を開始するには、**Connection** オブジェクトの次の機能を使用します。

- プロバイダの名前を表す **Provider** プロパティ。プロバイダ名を指定しない場合は、MSDASQL プロバイダが使用されます。
- 接続文字列を格納する **ConnectionString** プロパティ。このプロパティは Sybase IQ 接続文字列を格納します。この接続文字列は ODBC ドライバと同じ方法で使用されます。ODBC データ・ソース名を指定することも、他の接続文字列と同様に、UserID、Password、DatabaseName などのパラメータを明示的に指定することもできます。

接続パラメータのリストについては、「[接続パラメータ](#)」(86 ページ) を参照してください。

- 接続を開始する **Open** メソッド。

データベースに接続するためのサンプル・コードなど、ADO の詳細については、『SQL Anywhere サーバー プログラミング』を参照してください。

注意 『SQL Anywhere サーバー プログラミング』の各例を実行するには、データ・ソース「Sybase IQ デモ」とサンプル・ディレクトリ `%ALLUSERSPROFILE%\All Users\SybaseIQ\samples\SQLAnywhere\VB\Sampler\vb_sampler.sln` で代用します。Sybase IQ をインストールすると、適切なドライバが自動的に登録されますが、64 ビット Windows マシンで OLE DB に基づくアプリケーションを開発し配備するには、32 ビットと 64 ビットの両方のドライバを登録する必要があります。regsvr32 を実行するときは、`dboledb11.dll` および `dboledb11.dll` へのフル・パスを指定してください。

接続パラメータ

「第4章 接続パラメータと通信パラメータ」では、Sybase IQ 接続パラメータについて説明します。

接続パラメータについてのヒント

接続パラメータでは、1つの作業を実行するのに複数の方法を使用できることがよくあります。これは特に、埋め込みデータベースでデータベース・サーバを接続文字列によって起動する場合に当てはまります。たとえば、接続時にデータベースを起動する場合は、データベース名を DatabaseName (DBN) 接続パラメータまたは DatabaseSwitch (DBS) パラメータで指定できます。

次に、接続パラメータが競合した場合の対処方法をいくつか示します。

- **データベース・ファイルを DBF で指定する** データベース・ファイルは StartLine (START) パラメータでも DatabaseFile (DBF) 接続パラメータでも指定できますが、後者を使用することをおすすめします。
- **データベース名を DBN で指定する** データベース名は StartLine (START) パラメータ、DatabaseSwitch (DBS) 接続パラメータ、または DatabaseName (DBN) 接続パラメータで指定できますが、DBN を使用することをおすすめします。
- **キャッシュ・サイズを START パラメータで指定する** DatabaseFile (DBF) 接続パラメータを使用してデータベース・ファイルを指定する場合でも、開始方法の調整が必要となることがあります。これには、StartLine (START) 接続パラメータを使用します。推奨パラメータについては、「[必須のコマンド・ライン・オプション](#)」(29 ページ)を参照してください。

たとえば、StartLine (START) 接続パラメータでカタログ・キャッシュ・メモリの拡張を指定したい場合は、次に示す埋め込みデータベース接続パラメータのようにします。この例では、Windows のデフォルトのカタログ・キャッシュ・サイズ (32M) を 48M に増やしています。

```
DBF=path\iqdemo.db
DBN=Sybase IQ Demo
ENG=Sample Server
UID=DBA
PWD=sql
Start=start_iq -c 48M -gd all -gp 4096
```

Sybase IQ の接続の確立方法

この項では、インタフェース・ライブラリが接続を確立する方法について説明します。

この項の対象読者

多くの場合、この章の前の項で説明されている情報を使用して、簡単にデータベースに接続できます。しかし、サーバに正しく接続できない場合は、問題を解決するために、Sybase IQ が接続を確立するしくみを理解する必要があります。接続の問題を解決する方法の詳細については、「[付録 トラブルシューティングのヒント](#)」を参照してください。

注意 データベースに問題なく接続できる場合は、この項を読む必要はありません。

Sybase IQ は、次に示す種類のクライアント・アプリケーションに対してまったく同じ手順を実行します。

- **SQLDriverConnect** 関数を使用する ODBC アプリケーション。この関数は、ODBC アプリケーションが接続を確立する一般的な方法です。Sybase PowerBuilder や Power++ など、多くのアプリケーション開発システムは、この種類に属しています。
- **Embedded SQL™** を使用するクライアント・アプリケーションと、データベースに接続するための推奨関数 (`db_string_connect`) を使用するクライアント・アプリケーション。

Embedded SQL アプリケーションと Interactive SQL では、SQL CONNECT 文を使用できます。SQL CONNECT 文には、**CONNECT AS...** と **CONNECT USING...** の2種類の形式があります。ユーティリティや Interactive SQL などのデータベース管理ツールはすべて `db_string_connect` を使用します。

接続を確立する手順

Sybase IQ との接続を確立するため、クライアント・アプリケーションは次の手順を実行します。

- 1 インタフェース・ライブラリを検出します。クライアント・アプリケーションは ODBC ドライバまたは Embedded SQL インタフェース・ライブラリを検索する必要があります。
- 2 接続パラメータ・リストをアセンブルします。接続パラメータはさまざまな場所 (データ・ソース、アプリケーションによってアセンブルされた接続文字列、環境変数など) で指定されていることがあるため、Sybase IQ はこれらのパラメータを1つのリストにまとめます。

- 3 サーバを検出します。Sybase IQ は接続パラメータに基づいて、同じマシン上またはネットワーク上のデータベース・サーバを探します。
- 4 データベースを検出します。Sybase IQ は、検出したサーバ上で接続先のデータベースを探します。

次の項では、各手順の詳細について説明します。

インタフェース・ライブラリの検出

クライアント・アプリケーションは、Sybase IQ インタフェース・ライブラリの 1 つを呼び出します。通常は、この DLL または共有ライブラリの位置をユーザが意識する必要はありません。ここでは、問題が発生した場合にどのようにライブラリが検出されるかについて説明します。

ODBC ドライバの位置

ODBC では、インタフェース・ライブラリを ODBC ドライバとも呼びます。ODBC クライアント・アプリケーションは ODBC ドライバ・マネージャを呼び出し、このドライバ・マネージャが Sybase IQ のドライバを探します。

ODBC ドライバ・マネージャは、*odbc.ini* ファイルまたはレジストリ内で指定されたデータ・ソースに基づいて、ドライバを探します。ODBC アドミニストレータは、IQ クライアント・ソフトウェアのインストール時に設定されたドライバのレジストリ・エントリを基にして、ドライバの現在の場所を入力します。

Embedded SQL インタフェース・ライブラリの位置

Embedded SQL アプリケーションは、インタフェース・ライブラリを名前で指定して呼び出します。Sybase IQ Embedded SQL インタフェース・ライブラリの名前は、次のとおりです。

- Windows の場合：*dblib11.dll*
- UNIX の場合：*libdblib11* (オペレーティング・システム固有の拡張子が付く)

探索する場所は、オペレーティング・システムによって異なります。

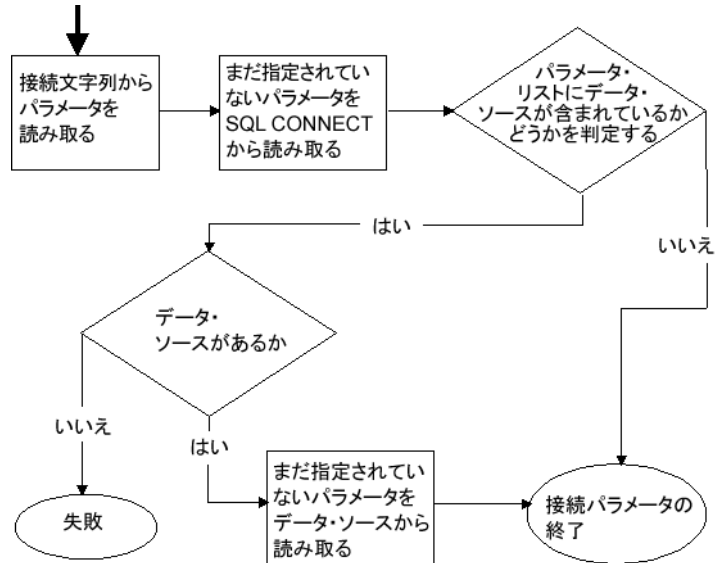
- Windows の場合、クライアント・アプリケーションは、現在のディレクトリ、システム・パス、*Windows* ディレクトリ、*Windows\System* ディレクトリからファイルを探します。
- UNIX の場合、クライアント・アプリケーションは、システム・パスとユーザ・ライブラリ・パスからファイルを探します。

ライブラリが検出されるタイミング

インタフェース・ライブラリが見つかると、クライアント・アプリケーションは、接続文字列をインタフェース・ライブラリに渡します。インタフェース・ライブラリは接続文字列を使用して接続パラメータのリストをアセンブルして、これをサーバとの接続を確立するときに使用します。

接続パラメータ・リストのアセンブル

次の図は、インタフェース・ライブラリが接続の確立に使用する接続パラメータのリストをアセンブルする方法を示しています。



注意事項

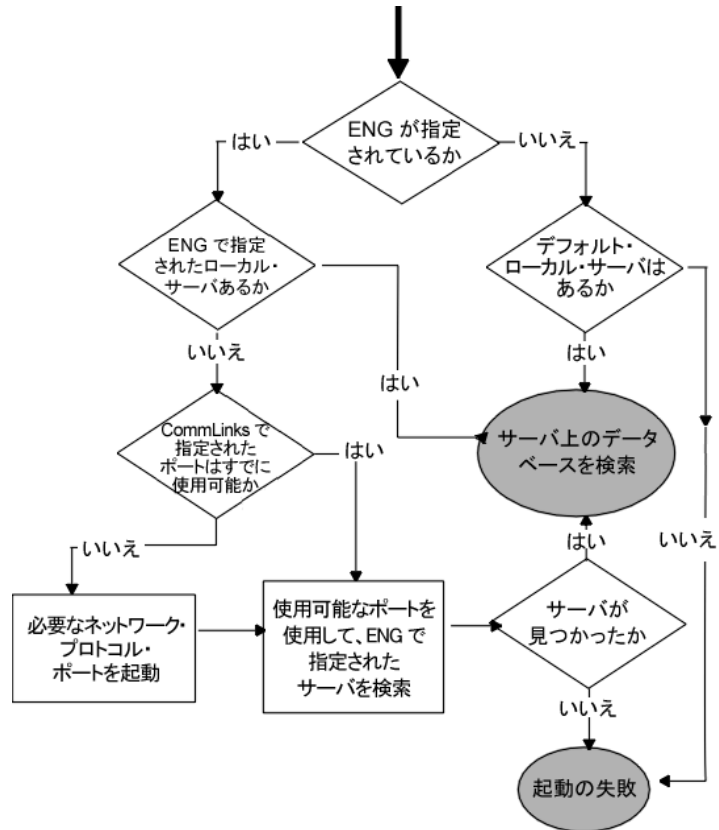
図の重要な点を次に示します。

- 優先度 - 複数の場所に格納されているパラメータは、次の優先順位に従います。
 - a 接続文字列
 - b SQLCONNECT
 - c データ・ソース
 つまり、パラメータがデータ・ソースと接続文字列の両方に指定された場合、データ・ソースの値より接続文字列の値が優先されます。
- 失敗 - この段階で失敗するのは、接続文字列または SQLCONNECT に、クライアントの接続ファイルに存在しないデータ・ソースが指定された場合のみです。
- 共通パラメータ - すでに他の接続が使用中である場合は、次の接続パラメータが無視されることがあります。
 - **AutoStop** データベースがすでにロードされている場合は、無視されます。
 - **DatabaseFile** DatabaseName が指定されていて、この名前を持つデータベースがすでに実行されている場合は、無視されます。

インタフェース・ライブラリは、接続しようとするときに接続パラメータの完全なリストを使用します。

サーバの検出

接続を確立するための次の手順では、Sybase IQ がサーバを検出します。接続パラメータ・リストにサーバ名 (EngineName (ENG) 接続パラメータ) が含まれている場合は、まずその名前のデータベース・サーバが検索され、次にネットワーク上で検索されます。ENG パラメータが指定されていない場合は、デフォルト・ローカル・サーバが検索されます。



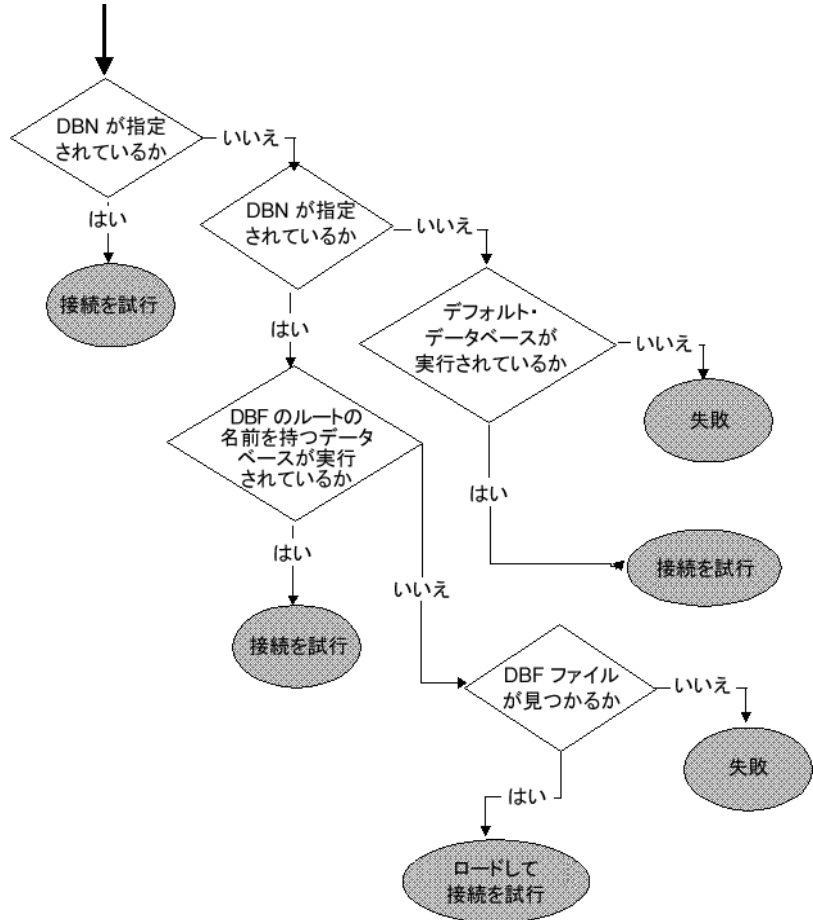
サーバが見つかった場合、Sybase IQ はそのサーバ上で目的のデータベースを見つけるかロードします。詳細については、「[データベースの検出](#)」(92 ページ)を参照してください。

注意

- ローカル接続では、サーバの検出は簡単です。ネットワーク接続を経由する場合は、CommLinks (LINKS) パラメータを使用し、ネットワーク通信パラメータを指定することで、検出方法を調整できます。
- ネットワーク検索では、Sybase IQ でサポートされる 1 つまたは複数のプロトコルが検索されます。ネットワーク・ライブラリは、各プロトコルに対して単一の「ポート」を開始します。この単一ポートは、そのプロトコルが使用されるすべての接続で同時に使用されます。
- CommLinks (LINKS) パラメータの引数では、それぞれのネットワーク・ポートに対して一連のネットワーク通信パラメータを指定できます。これらのパラメータは、ポートが最初に開始されるときにのみ使用されます。したがって、すでに開始しているポートに関しては、CommLinks (LINKS) で指定された通信パラメータは無視されます。
- ローカルおよび各ネットワーク・ポートに対してサーバ検出を行うときには、2 段階の手順を踏みます。まず、Sybase IQ がサーバ名キャッシュを調べ、その名前のサーバが使用できるかどうかを確認します。次に、使用可能な接続パラメータを使用して接続を試みます。
- マシンで最初に起動されるサーバは、デフォルト・サーバと呼ばれます。このサーバは、共有メモリ・ポートを使用します。

データベースの検出

サーバが見つかったら、次に、インタフェース・ライブラリはデータベースを検索します。例を示します。



注意

- DBF パラメータを使用する場合、Sybase IQ が指定のデータベースを探せるようにするには、DBF パスは絶対パス、または、サーバが起動された場所からの相対パスである必要があります。たとえば、`../foo/iqdemo` を指定した場合、まずサーバが置かれているディレクトリが参照され、次に `foo` が参照されます。
- デフォルトのデータベースは、サーバ上で起動しているものの中から 1 つ 選択されます。

Interactive SQL 接続

Interactive SQL (dbisql) ユーティリティは、データベースにすでに接続しているときに **CONNECT** 文が発行された場合に、Embedded SQL のデフォルトの動作とは異なる動作をします。**CONNECT** 文にデータベースやサーバが指定されていない場合、Interactive SQL は、デフォルト・データベースではなく現在のデータベースに接続します。この動作は、データベースを再起動するときに必要です。

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

他のデータベースからの接続

Adaptive Server Enterprise から、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、Sybase IQ テーブルを示す「プロキシ・テーブル」を Adaptive Server Enterprise 内に定義します。その後、Adaptive Server Enterprise からプロキシ・テーブルにクエリを行うことができます。関連情報については、[「Adaptive Server Enterprise のデータ型の対応」\(321 ページ\)](#) を参照してください。CIS の詳細については、『Adaptive Server Enterprise and OmniConnect 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 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。

- Sybase IQ から Adaptive Server Enterprise 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。詳細については、プラットフォームに対応した『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 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。
- 異なる Sybase IQ データベース内にあるテーブルのジョイン。

詳細については、『システム管理ガイド第2巻』の「[第4章 リモート・データへのアクセス](#)」を参照してください。

UNIX システムでのポート番号の競合防止

SQL Anywhere と Sybase IQ が同じシステムにインストールされている場合は、製品の競合を回避するために、*default.cfg* と他の設定ファイル (*iqdemo.cfg* など) 内で IQ ポート番号を変更する必要があります。両製品とも、デフォルト・ポート 2638 を使用しています。

❖ IQ ポート番号の変更

- 1 次の行を *\$IQDIR15/scripts/default.cfg* に追加します。この例では、未使用のポート番号は次の通りです。

```
-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 と Sybase IQ が同じサブネット上にある場合、サーバ名はユニークでなければなりません。

サーバを検出できるかどうかのテスト

`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
```

`dbping` の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。

接続試行中に詳細な情報を出力する方法については、「[LogFile 接続パラメータ \(LOG\)](#)」(128 ページ)を参照してください。この機能は、同じ章で説明されている Logfile 接続パラメータと組み合わせて使用すると特に有用です。

統合化ログインの使用

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。この項では、統合化ログイン機能について説明します。

統合化ログインの利点

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の両方が、単一のシステムによって確保されます。
- 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。

セキュリティと統合化ログインの詳細については、「[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](#)」(99 ページ)を参照してください。

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。

❖ 統合化ログインの使用

- 1 LOGIN_MODE データベース・オプションの値を、デフォルト値の **Standard** の代わりに **Mixed** または **Integrated** (このオプションは大文字と小文字を区別しません) のいずれかに設定することによって、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、DBA 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 (大文字と小文字を区別します)。

- **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** この設定では、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だけ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。ここでの唯一の例外は、DBA 権限のあるユーザ (完全な管理権を持つ) です。

例 次の SQL 文は、LOGIN_MODE データベース・オプションの値を **Mixed** に設定し、Standard と Integrated の両方のログイン接続を許可します。

```
SET OPTION "PUBLIC".LOGIN_MODE = Mixed
```

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 にのみマップできます。データベース・ユーザ ID がデータベースから削除されると、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・マッピングが自動的に削除されます。

ユーザ・プロファイルとデータベース・ユーザ ID を 1 対 1 の対応にする必要はありません。同じユーザ ID に対して複数のユーザ・プロファイルをマップできます。

DBA 権限を持つユーザのみ、統合化ログイン・マッピングを作成または削除できます。

統合化ログイン・マッピングは、Sybase Central でウィザードを使用するか、SQL 文を使用して作成できます。

❖ 統合化ログインのマッピング (Sybase Central)

- 1 DBA 権限を持つユーザとしてデータベースに接続します。
- 2 データベースの [Login Mappings] フォルダを右クリックし、[ログイン・マッピング] を選択します。[統合化ログイン] ウィザードが表示されます。
- 3 ウィザードの最初のページに、統合化ログインが作成されるシステム (コンピュータ) のユーザ名を入力します。名前は、リストから選択しても入力してもかまいません。

また、このユーザのマップ先のデータベース・ユーザ ID を選択します。ウィザードに、使用可能なデータベース・ユーザが表示されます。その中の 1 つを選択してください。新しいデータベース・ユーザ ID は追加できません。

- 4 ウィザードの残りの指示に従います。

❖ 統合化ログインのマッピング (SQL)

- 次の SQL 文を使用すると、fran_whitney と matthew_cobb という Windows ユーザが、DBA のユーザ ID またはパスワードを知らなくても (または指定しなくても)、DBA ユーザ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「[GRANT 文](#)」を参照してください。

統合化ログイン・パーミッションの取り消し

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。

❖ 統合化ログイン・パーミッションの取り消し (SQL)

- 1 DBA 権限を使用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。

例 次の SQL 文では、pchin という Windows ユーザから統合化ログイン・パーミッションを削除します。

```
REVOKE INTEGRATED LOGIN  
FROM pchin
```

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

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、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** を指定した場合は、統合化ログインが実行されます。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行います。

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。

Interactive SQL の例

たとえば、ユーザが、サーバのデフォルト・データベース内の統合化ログイン・マッピングと一致するユーザ・プロファイル名を使用してログオンした場合、次の Interactive SQL 文を使用した接続は成功します。

```
CONNECT USING 'INTEGRATED=yes'
```

次の Interactive SQL 文について考えます。

```
CONNECT
```

以下がすべてあてはまる場合、データベースに接続できます。

- サーバが現在動作中である
- 現在のサーバのデフォルト・データベースでは、統合化ログインの接続を受け入れることができる

- 現在のユーザのユーザ・プロファイル名と一致する統合化ログイン・マッピングが作成されている
- サーバから接続情報をさらに要求するダイアログ・ボックスが表示された場合に (dbisql ユーティリティの使用中等に発生)、ユーザが情報を追加せずに [OK] をクリックする

ODBC を介した統合化ログイン

ODBC を介してデータベースに接続するクライアント・アプリケーションは、データ・ソース設定のその他の属性に Integrated パラメータを含めることによって、統合化ログインを使用できます。

ODBC データ・ソースで属性 'Integrated=yes' を設定すると、統合化ログインを行うために DSN を使用してデータベース接続が実行されます。LOGIN_MODE データベース・オプションが Standard に設定されている場合は、ODBC ドライバがユーザに対してデータベース・ユーザ ID とパスワードを要求します。

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、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かぎり、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 を有効にしておくこと、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` 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、そのユーザは自動的に、完全な管理権限を持つ `DBA` としてデータベースに接続します。
- 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、どん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、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。これを行うには、Windows のユーザー・マネージャを使用します。

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` 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可能になります。

```
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
```

永久オプションが `Standard` の場合、データベースは停止時に元の値に戻ります。

一時的な `public` オプションの設定は、データベース・アクセスのためのセキュリティ対策として考えられます。これは、統合化ログインを有効にすることが、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頼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 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。

`SET OPTION` 文の使用法の詳細については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」を参照してください。

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・マシンおよびネットワーク・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 Domain 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、Domain Controller サーバのドメイン・パーミッションを持ち、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・ユーザの作成

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` に関する統合化ログイン・マッピングが存在しない場合、通常は、`JSMITH` のユーザ・プロファイルを使用した統合化ログインによる接続は失敗します。

しかし、`Guest` という名前のユーザ ID をデータベースに作成しておく、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は、統合化ログインが `Guest` ユーザ ID にマップするようになります。

データベースに `Guest` というユーザ ID が含まれている場合、デフォルトの統合化ログイン・ユーザは、統合化ログインを試みるすべてのユーザに対して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。

接続の切断と削除

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。
- Sybase Central では、ユーザが [接続] - [切断] を選択した場合。
- Interactive SQL や Sybase Central 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、DBA 権限を持つユーザは、次の方法で特定の接続を削除できます。

- Interactive SQL または Embedded SQL では、DROP CONNECTION 文を発行します。
- Sybase Central では、データベースの [ユーザとグループ] フォルダにアクセスします。

接続 ID とアクティブ接続の情報を表示する方法の詳細については、『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「[sp_iqconnection プロシージャ](#)」を参照してください。文の構文については、『リファレンス：文とオプション』の「[DISCONNECT 文 \[DBISQL\]](#)」と「[DROP CONNECTION 文](#)」を参照してください。Sybase Central については、『Sybase IQ の概要』の「[第 4 章 Managing Databases](#)」を参照してください。

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 `.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「[LOG_CONNECT オプション](#)」を参照してください。

起動のトラブルシューティング、シャットダウン、接続

データベース・サーバ、接続、Interactive SQL の問題を解決する方法については、以降の各項を参照してください。ネットワーク通信の問題など、その他のトラブルシューティングのヒントについては、「[付録 14 トラブルシューティングのヒント](#)」を参照してください。

Sybase IQ を起動できない

この項では、データベース・サーバの起動時によく発生する問題について説明します。

ファイルが有効なことを確認する

既存のトランザクション・ログが無効な場合、サーバは起動しません。このような問題が発生する状況としては、たとえば、開発中に、データベース・ファイルを新バージョンに置き換えるときに古いトランザクション・ログを削除しなかった場合などが考えられます。この場合、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ないため、無効になります。

IQ テンポラリ・ストア・ファイルの場合も同じです。

テンポラリ・ファイル用の十分なディスク領域があるかを確認する

Sybase IQ は、実行時の情報をテンポラリ・ファイルに保存します。このファイルは、TMP または TEMP 環境変数で示されたディレクトリ内、または UNIX の場合には SATMP 環境変数で示されたディレクトリ内に格納されています。Windows の場合、通常は `c:\temp` ディレクトリが使用されます。UNIX の場合、複数のデータベース・サーバが同じマシン上で実行されているときは、ユーザーごとに個別のテンポラリ・ディレクトリが必要です。このディレクトリは、通常、`/tmp/userid` に設定されます。

テンポラリ・ディレクトリが使用できる十分なディスク領域がない場合、サーバの起動に問題が生じることがあります。

ネットワーク通信ソフトウェアが実行されているかを確認する

データベース・サーバを実行する前に、適切なネットワーク通信ソフトウェアがインストールされ、実行されている必要があります。1つのネットワークだけがインストールされた状態で信頼性の高いネットワーク・ソフトウェアを実行している場合は、問題はありません。問題が発生した場合、標準以外のソフトウェアを実行している場合、または複数のネットワークを実行している場合は、『インストールおよび設定ガイド』のネットワーク通信の説明を参照してください。また、ネットワーク・ベンダのマニュアルも参照してください。

データベース・サーバを実行する前に、ネットワーク通信を必要とする他のソフトウェアが正しく動作していることを確認してください。

たとえば、TCP/IP プロトコルを使用している場合は、ping と telnet が正しく動作しているかどうかを確認します。ping および telnet アプリケーションには、さまざまな TCP/IP プロトコル・スタックが関係しています。

環境変数を確認する

サーバを起動するには、特定の環境変数を正しく設定してください。Windows の場合、必要な環境変数はインストール手順によって自動的に設定されます。UNIX の場合、自分で環境変数を設定する必要があります。こうした設定を自分で変更していない場合でも、サーバを起動できないときは、設定が正しいことを確認してください。環境変数の設定については、『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「[第 1 章 ファイル・ロケーションとインストール設定](#)」を参照してください。

ネットワーク通信の起動時の問題をデバッグする

ネットワークで接続を確立するときに問題が生じた場合は、クライアントとサーバの両方でデバッグ・オプションを使用し、問題点を診断できます。サーバ側では、`-z` コマンド・ライン・オプションを使用します。起動パラメータはサーバのウィンドウに表示されます (`start_iq` を使用してサーバを起動する場合はサーバ・ログに記録されます)。サーバを `start_iq` で起動する場合は、`-o filename` オプションを使用して、結果を出力ファイルに記録できます。

データベースに接続できない

Sybase IQ データベースに接続できない場合は、次の事項を確認してください。

- 入力したデータ・ソース名が正しいこと、または JDBC 接続で選択したサーバ名が正しいことを確認します。
- データ・ソース (DSN または FILEDSN) に、正しいサーバ名、データベース、ネットワーク・パラメータ、その他の必要となる接続情報が含まれていることを確認します。
 - Windows の場合は、[設定]-[コントロール パネル]-[データ ソース (ODBC)] を選択し、[ユーザー DSN] タブを選択して、目的のサーバのソース名を選択します。
 - UNIX の場合は、`.odbc.ini` ファイルを調べます。

- 目的のデータベース・サーバが実行中であることを確認します。
 - Windows の場合：サーバがローカル・ホスト上で実行されている場合は、画面下部のツールバーのアイコンを調べます。サーバがリモートの場合は、[設定]-[コントロールパネル]-[データソース (ODBC)]を選択すると、[ODBC データソース アドミニストレータ]が表示されます。ここで、[ユーザー DSN] タブを選択し、目的のサーバのソース名を選択し、[構成]-[テスト接続]を選択します。目的のサーバが実行中で、データ・ソースが正しく設定されている場合は、接続成功のメッセージが表示されます。
 - UNIX の場合：システム・プロンプトで次のように入力します。 *iqdemo* の部分は、使用しているデータベース・サーバ名に置き換えてください。

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

- コマンド・ラインで入力した接続パラメータが正しいことを確認します。
データベースにすでに接続しているときに CONNECT 文が発行された場合、Interactive SQL ユーティリティは Embedded SQL とは異なる動作をします。CONNECT 文にデータベースやサーバが指定されていないければ、Interactive SQL は、デフォルト・データベースではなく現在のデータベースに接続します。この動作は、データベースを再ロードするときが必要です。
サーバ名を指定せずに、データベース名を指定した場合、Interactive SQL は現在のサーバ上の指定したデータベースに接続しようとします(データベース・ファイル名ではなく、-n データベース・スイッチで定義されたデータベース名を指定する必要があります)。データベース名を指定せずにサーバ名を指定した場合、Interactive SQL は、指定したサーバ上のデフォルト・データベースに接続します。
- 実行中でないデータベースに接続する場合は、サーバを起動したときに -gd ALL スイッチを指定したかどうかを確認します。このスイッチを指定していない場合は、DBA だけがそのサーバ上のデータベースを起動できます (utility_db データベースに接続し、目的のデータベースに対して START DATABASE コマンドを発行します)。
- 要求された操作に対してデータベースを使用する権限があることを確認します。DBA またはデータベース所有者から、CONNECT 権限を付与してもらう必要があります。詳細については、「第8章 ユーザ ID とパーミッションの管理」を参照してください。
- ネットワークで接続を確立するときに問題が生じた場合は、クライアントとサーバの両方でデバッグ・オプションを使用し、問題点を診断できます。サーバ側では、-z コマンド・ライン・オプションを使用します。起動情報はサーバ・ウィンドウに表示されます。-o オプションを使用して、この結果を出力ファイルに記録できます。

- 要求されたデータベースに存在するすべてのファイルを確認します。少なくとも、IQ ストア (*dbname.iq*)、カタログ・ストア (*dbname.db*)、IQ テンポラリー・ストア (*dbname.iqtmp*)、トランザクション・ログ (*dbname.log*: 新しく作成され、まだ更新されていないデータベースにはない場合があります)、およびメッセージ・ファイル (*dbname.iqmsg*) が必要です。かっこで囲んだ名前はデフォルトのため、使用しているシステムによっては異なる場合があります。
- リストア操作が正常に完了したことを確認します。

[「Sybase IQ の接続の確立方法」\(87 ページ\)](#) も参照してください。

緊急時のデータベース・サーバの停止 (UNIX)

通常は、「[データベース・サーバの停止](#)」(44 ページ) で説明する方法を使用してサーバを停止します。この方法で停止できない場合や、データベース・サーバをバッチ・プロセスまたはバックグラウンド・プロセスとして起動した場合 (`start_iq` を使用した場合) は、次の手順に従います。

- 1 可能であれば、他のユーザがデータベースに接続していないことを確認します。
- 2 UNIX プロンプトで次のコマンドを入力します。

```
kill -hup pid
```

ここで、*pid* は停止するデータベース・サーバのプロセス ID です。

通常の方法では停止できないサーバを停止したり、プロセスの強制終了後にシステムをクリーンアップしたりするときの注意事項については、「[付録 14 トラブルシューティングのヒント](#)」を参照してください。

UNIX kill コマンドによる Interactive SQL の停止

通常、Interactive SQL を終了するには [ファイル]-[終了] コマンドを使用します。緊急時に UNIX コマンド・レベルから Interactive SQL を停止する必要がある場合は、正しいプロセスを停止するように注意してください。Interactive SQL を起動すると、2つのプロセスが起動します。`dbisql` 実行プログラムを実行するシェル・プロセスと、`$$SYBASE/IQ-15_2/jre/bin/native_threads` ディレクトリから実行される Java 実行プロセスです。このうち、Java 実行プロセスの方を強制終了します。

両方のプロセスを探すには、次のようなコマンドを使用します。

```
ps -ef | egrep "jre|java|dbisql"
ami 1712 21442 0 12:01:17 pts/11 0:00 /bin/sh /mysystem/ami/IQ-15_2/bin/dbisql
ami 1748 21894 0 12:10:13 pts/14 0:00 egrep jre|java|dbisql
ami 1715 1712 0 12:01:18 pts/11 0:07
/mysystem/ami/IQ-15_2/jre/bin/..bin/sparc/native_threa
```

その後、`kill -9`などのコマンドを使用して、Java プロセス (パス内に `jre` を含むもの) のプロセス ID を強制終了します。次に例を示します。

```
kill -9 1715
```


この章について

この章では、接続パラメータについて説明します。このパラメータを使って、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。

内容

トピック	ページ
「接続パラメータ」	109
「ネットワーク通信のパラメータ」	132

接続パラメータ

この項では、各接続パラメータについて説明します。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。

接続パラメータは、接続文字列に含まれます。次の場所に入力することもできます。

- 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

詳細については、「[第 3 章 Sybase IQ の接続](#)」を参照してください。

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 接続ダイアログは、共通の形式を持ちます。一部のパラメータは、これらのダイアログのチェックボックスやフィールドに対応していません。その他のパラメータは、[Advanced] 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のところで、パラメータが使用される状況を説明します。一般的に使用法にあげる項目は以下のとおりです。

- **埋め込みデータベース** Sybase IQ を埋め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、データベースがアンロードされ、サーバは停止します。埋め込みデータベースの詳細については、「[簡単な接続の例](#)」(60 ページ)を参照してください。

- **ネットワーク・サーバ** 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「[Ping ユーティリティ \(dbping\)](#)」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る接続パラメータは、次の場所から取得できます (優先度順)。

- 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。文字セットの詳細については、「[接続文字列と文字セット](#)」を参照してください。

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へ読み込まれます。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いて、かつ、データ・ソース内でいくつかのパラメータが明示的に設定されているときに、両者の間で競合が発生した場合は、明示的なパラメータの方が使用されます。

AppInfo 接続パラメータ (App)

機能	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	空の文字列
説明	この接続パラメータは、Embedded SQL、ODBC、または OLE DB クライアントからデータベース・サーバに送信されます。Open Client アプリケーションや jConnect アプリケーション (Interactive SQL (dbisql) や Sybase Central など) からは使用できません。

AppInfo 接続パラメータは、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 プロパティの文字列は、一連の **key=value** のペアをセミコロンで区切ったものです。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** 接続文字列内で AppInfo を指定した場合は、そこで入力した文字列
- **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・マシンの 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** 接続のローカルな時刻を表示するために万国標準時 (UTC) に加算する必要がある時間 (分単位)

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 (iAnywhere JDBC ドライバがデフォルトで使用されます)。

```
dbisql -c "uid=DBA;pwd=SQL;dbf=C:\Program Files\Sybase\IQ-15_2\demo\iqdemo.db"
```

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=10.25.99.116;
HOST=machine-name;
OSUSER=smith;
OS="Windows XP Build 2600 Service Pack 3";
EXE=C:\Program Files\Sybase\IQ-15_2\bin32\dbisql.exe;
```

```
PID=0x108c;
THREAD=0x124c;
VERSION=11.0.1.5538;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-300
```

- 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。

```
dbisql -c "uid=DBA;pwd=SQL;app=ISQL connection"
```

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=machine-name;
OS=Sun_Solaris 2.8;
PID=0x10e;
THREAD=0xe1;
VERSION=9.0.1.1549;
TIMEZONEADJUSTMENT=-300
APPINFO=ISQL connection
```

AutoPreCommit 接続パラメータ (AutoPreCommit)

機能 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 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 (Y) に設定すると実行前のコミットが有効になり、NO (N) 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-15_2\demo\iqdemo.db
AutoPreCommit=Y
UserID=DBA
Password=SQL
```

AutoStart 接続パラメータ (Astart)

機能	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	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、データベース・サーバは同一のマシンで起動されます。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	<ul style="list-style-type: none"> 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。 <pre>[My Demo Database] DatabaseFile=c:¥sybase¥IQ-15_2¥demo¥iqdemo.db Autostart=No UserID=DBA ENG=network_server</pre>

AutoStop 接続パラメータ (Astop)

機能	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	Yes
説明	<p>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。</p> <p>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。</p> <p>現在実行中でないデータベースに接続する場合にのみ、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。</p>
例	<p>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。</p> <pre>[Sample Embedded Database] DatabaseFile=c:¥sybase¥IQ-15_2¥demo¥iqdemo.db Autostop=No UserID=DBA</pre>

CharSet 接続パラメータ (CS)

機能	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	ロケール文字セット。これを調べる方法の詳細については、「 ロケール情報の決定 」(435 ページ)を参照してください。
説明	<p>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。</p> <p>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。有効な文字セット値については、「ロケールの設定」(435 ページ)を参照してください。</p> <p>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。ワイド (Unicode) 関数の使用時には、ODBC も Unicode クライアント API です。</p>
参照	「接続パラメータの構造」 (54 ページ)

CompressionThreshold 接続パラメータ (COMPTH)

説明	<p>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。</p> <p>CompressionThreshold パラメータは、SQL Anywhere プロトコル・オプションです。『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「接続パラメータとネットワーク・プロトコル・オプション」>「接続パラメータ」>「CompressionThreshold 接続パラメータ [COMPTH]」を参照してください。</p>
----	--

CommBufferSize 接続パラメータ (CBSize)

機能	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。
使用法	すべての場所
値	整数値 [k]
デフォルト	CommBufferSize 値を設定しないと、CommBufferSize はサーバ設定 (デフォルト値は 1460 バイト) を使って制御されます。
説明	CommBufferSize パラメータは、通信パケットのサイズをバイトで指定します。CommBufferSize の最小値は 300 で、最大値は 16000 です。

このプロトコル・スタックは、ネットワーク上のパケットの最大サイズを設定します。**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した場合は、最も大きいバッファがネットワーク・ソフトウェアによって分割されます。バッファ・サイズは、ネットワークで許可されているサイズよりいくらか小さく設定してください。これは、ネットワーク・ソフトウェアが、ネットワークに送信する前に各バッファに情報を追加することがあるためです。TCP/IP を使用しているときは、デフォルトの 1460 バイトでは、Ethernet パケットが完全にいっぱいになってしまいます。

パケット・サイズを大きくすると、マルチロー・フェッチと大きな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両方のメモリ使用量が増加します。

クライアント上で **CommBufferSize** が指定されていない場合は、その接続ではサーバのバッファ・サイズが使用されます。クライアント上で **CommBufferSize** が指定されている場合は、その接続では **CommBufferSize** の値が使用されます。

-p データベース・サーバ・オプションを使用して **CommBufferSize** を設定した場合は、**CommBufferSize** を独自に指定していないすべてのクライアントが、-p データベース・サーバ・オプションで指定されたサイズを使用するようになります。

例

バッファ・サイズを 400 バイトに設定します。

```
...
CommBufferSize=400
...
```

別の方法として、接続ウィンドウの [ネットワーク] タブの [バッファ・サイズ] 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)

機能	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	接続には共有メモリ通信リンクのみを使用します。
参照	「ネットワーク通信のパラメータ」(132 ページ) 『ユーティリティ・ガイド』の「表 1-1」の「-x list」
説明	CommLinks=ALL を指定した場合、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を与えるおそれがあるため、この設定は、どのプロトコルを使用すればいいのかわからないときのみ使用してください。

CommLinks (LINKS) 接続パラメータに複数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。デフォルトの設定です。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** まず共有メモリ・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。この設定は、どの通信プロトコルを使用すればよいかわからない場合に使用します。
- **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。パラメータのリストについては、「[ネットワーク通信のパラメータ](#)」(132 ページ)を参照してください。

次のような理由から、ALL とは反対に、特定のプロトコルを使用したい場合もあります。

- 不要なネットワーク・リンクが起動されていない場合は、ネットワーク・ライブラリの起動が少し速くなる。
- 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したい場合は、明示的にリンクを指定する必要があります。

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 コマンド・ライン・スイッチに対応します。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) , shm
```

ConnectionName 接続パラメータ (CON)

- 機能 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 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
```

DatabaseFile 接続パラメータ (DBF)

- 機能 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。
- Sybase IQ では、特別な状況における接続で、DBF パラメータとデータベース・ファイル名が必要な場合があります。詳細については、「[リストア後の再接続](#)」(475 ページ)を参照してください。
- 使用法 埋め込みデータベース
- 値 文字列
- デフォルト デフォルト設定なし
- 参照 『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」

- 説明 DatabaseFile (DBF) 接続パラメータは、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するために使用されます。
- 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は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す。
 - 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を基準とする相対パスです。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 ファイル名を使用することもできます。

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。

参照 [「DatabaseName 接続パラメータ \(DBN\)」](#)

例 デモ・データベース (Windows の場合は `C:\Program Files\Sybase\IQ-15_2\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-15_2\demo\iqdemo.db
```

DatabaseName 接続パラメータ (DBN)

- 機能 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ときに使用します。
- 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 文字列
- デフォルト デフォルト設定なし

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の名前と `DatabaseName (DBN)` パラメータで指定した名前が一致する場合にのみ、接続が確立されます。

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。

参照 HTTP 接続については、「[DatabaseName 通信パラメータ \(DBN\)](#)」(136 ページ) を参照してください。

例

- `cities.db` というデータベース・ファイルを起動し、このデータベース名を `Kitchener` にするには、次のコマンドを使用します。

```
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
```
- 別の方法として、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;DBF=cities.db
```
- しかし、次のコマンドを使用して `Kitchener` という名前のデータベースに接続しようとするると失敗します。

```
DBF=cities.db
```

DatabaseSwitches 接続パラメータ (DBS)

機能 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 スイッチなし

参照 『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」
「[StartLine 接続パラメータ \(START\)](#)」

説明	<p>現在実行中でないデータベースに接続する場合にのみ、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。</p> <p>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。</p>
例	<p>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 <i>/IQ-15_2/demo/iqdemo.db</i> (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。</p> <pre>dbcollat -c "uid=DBA;pwd=SQL;dbf=/IQ-15_2/demo/iqdemo.db; dbn=my_db;dbs=-n my_db" /tmp/temp.col</pre>

DataSourceName 接続パラメータ (DSN)

機能	ODBC データ・ソース情報を検索するときに、UNIX の場合は <i>.odbc.ini</i> ファイル、Windows の場合は <i>odbc.ini</i> 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	Anywhere
デフォルト	デフォルトのデータ・ソース名なし
参照	「FileDataSourceName 接続パラメータ (FileDSN)」 (124 ページ)
説明	ODBC アプリケーションでは、通常、データ・ソース名だけを ODBC に送信します。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。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す。
例	<p>次のパラメータは、データ・ソース名を使用します。</p> <pre>DSN=Dynamo Demo</pre>

DBKEY 接続パラメータ (DBKEY)

機能	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	デフォルトでは、データベースは暗号化されていません。
参照	『ユーティリティ・ガイド』の「 データベース・サーバの起動 」
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、暗号化されたキー “is!seCret” を使用して、現在実行中の、強い暗号で保護された *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)

機能 ネットワーク上での複数レコード・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 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、プリフェッチを回避します。

```
DMRF=Yes
```

EngineName 接続パラメータ (ENG)

機能 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* の同意語です。

使用法 ネットワーク・サーバ

値 文字列

デフォルト 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** を指定する必要があります。これは、[接続] ダイアログ・ボックスおよび ODBC アドミニストレータの [サーバ名] フィールドに相当します。

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。

プロトコル	トランケート後の長さ
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

参照 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第2章 SQL 言語の要素」の「識別子」

『ユーティリティ・ガイド』の「第1章 データベース・サーバの実行」の「データベース・サーバの命名制限 [-n]」

[「接続パラメータの構造」 \(54 ページ\)](#)

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)

機能 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 すべての場所 (DSN および FILEDSN 接続パラメータは冗長形のキーワードをサポートしていません)。

値 文字列

デフォルト 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 が優先されます。

参照 [「接続パラメータの構造」 \(54 ページ\)](#)

Encryption 接続パラメータ (ENC)

機能 クライアント・アプリケーションとサーバの間で送信されるパケットを暗号化します。

使用法 ECC_TLS (Certicom) と RSA_TLS は TCP/IP のみです。

NONE または SIMPLE は、すべての場所です。

値 文字列

デフォルト	NONE
説明	<p>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。</p> <p>ネットワーク・パケットのセキュリティが確実にない場合は、このパラメータを使用できます。暗号化はパフォーマンスにほとんど影響しません。</p> <p>Encryption (ENC) 接続パラメータは次の引数を受け入れます。</p> <ul style="list-style-type: none">• NONE 暗号化されていない通信パケットを受け入れます。この値は、以前のバージョンの Sybase IQ での NO に相当します。• 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 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。この値は、以前のバージョンの Sybase IQ での YES に相当します。• ECC_TLS (以前の Certicom) Certicom 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。この種の暗号化を使用するには、サーバとクライアントの両方が Solaris、Linux、およびサポートされるすべての Windows オペレーティング・システム上で稼働していて、その接続が TCP/IP ポートを使用している必要があります。UNIX プラットフォーム (Solaris および Linux を除く) は、クライアントまたはサーバの Certicom パラメータを認識しません。このサーバを認証するには、Certicom ソフトウェアが、次の引数で指定されるクライアントに関する値とサーバの証明書の値が一致しているかどうかを確認します。<ul style="list-style-type: none">• trusted_certificates 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は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は一致しなければなりません。• certificate_name 証明書の共通名を指定します。サーバの値とクライアントの値は一致しなければなりません。• RSA_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。この種の暗号化を使用するには、サーバとクライアントの両方が Solaris、Linux、およびサポートされるすべての Windows オペレーティング・システム上で稼働していて、その接続が TCP/IP ポートを使用している必要があります。UNIX プラットフォーム (Solaris および Linux を除く) は、クライアントまたはサーバの RSA_TLS パラメータを認識しません。このサーバを認証するには、Certicom ソフトウェアが、次の引数で指定されるクライアントに関する値とサーバの証明書の値が一致しているかどうかを確認します。<ul style="list-style-type: none">• trusted_certificates 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は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は一致しなければなりません。
- **certificate_name** 証明書の共通名を指定します。サーバの値とクライアントの値は一致しなければなりません。

警告！ サンプルの証明書は、テストのためだけに使用してください。サンプルの証明書からは、配備に耐えられる水準のセキュリティは提供されません。この証明書とそこで使用されるパスワードは、Sybase ソフトウェアに付属して広く配布されるからです。システムを保護するには、独自の証明書を作成する必要があります。

connection_property システム関数を使用すると、現在の接続の暗号化設定を取得できます。この関数は、使用される暗号化の種類に応じて none、simple、Certicom のいずれかの値を返します。

参照

『ユーティリティ・ガイド』の「データベース・サーバの起動」の「-ec サーバ・コマンド・ライン・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「[CONNECTION_PROPERTY 関数 \[システム \]](#)」

SQL Anywhere サーバ - データベース管理の「パート VI セキュリティ」の「トランスポート・レイヤ・セキュリティ」

例

- 次の接続文字列フラグメントでは、Certicom 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=ECC_TLS
(trusted_certificates=sample.crt)"
```

- 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=RSA_TLS
(trusted_certificates=sample.crt)"
```

FileDataSourceName 接続パラメータ (FileDSN)

機能

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

文字列

デフォルト	デフォルト名なし
説明	<p>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。</p> <p>ODBC アプリケーションと Embedded SQL アプリケーションは、どちらもファイル・データ・ソースを使用できます。</p>
例	<p>次に示すのは、ファイル・データ・ソース内にあるデータ・ソース記述です。</p> <pre>[Sample File Data Source] ENG=iqdemo DBA=DBA PWD=SQL</pre>
参照	「DataSourceName 接続パラメータ (DSN)」 (120 ページ)

Idle 接続パラメータ (IDLE)

機能	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 Sybase IQ Idle (IDLE) 接続パラメータを無視します。
値	整数値
デフォルト	-ti の値
説明	<p>Idle (IDLE) 接続パラメータは現在の接続にのみ適用されます。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。</p> <p>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 (<code>start_iq</code> 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。</p> <p>必要であれば、Idle および 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 <code>-ti</code> と <code>-tl</code>) を含めることができます。</p>
参照	『ユーティリティ・ガイド』の「 第1章 データベース・サーバの実行 」の「 <code>-ti</code> サーバ・コマンド・ライン・オプション」
例	<ul style="list-style-type: none"> 次の接続文字列フラグメントでは、この接続のタイムアウト値を 10 分に設定しています。 <pre>"ENG=myeng;LINKS=tcpip;IDLE=10"</pre>

Integrated 接続パラメータ (INT)

機能	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	なし
参照	『リファレンス：文とオプション』の「 第2章 データベース・オプション 」の「 LOGIN_MODE オプション 」
説明	<p>Integrated パラメータには次の設定があります。</p> <ul style="list-style-type: none"> • Yes 統合化ログインを試行します。接続の試行が失敗し、LOGIN_MODE オプションが Mixed に設定されている場合は、標準ログインを試行します。 • NO 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。 <p>統合化ログインを使用するクライアント・アプリケーションでは、login_mode データベース・オプションを Mixed か Integrated に設定してサーバを起動してください。</p>
例	<p>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。</p> <pre>INT=yes</pre>

Language 接続パラメータ (LANG)

機能	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 のように設定すると、デフォルトの言語をドイツ語に設定します。
デフォルト	SALANG 環境変数またはインストーラによって指定された言語 (SALANG 環境変数の方が優先)。
説明	<p>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。</p> <p>言語が指定されていない場合は、デフォルトの言語が使用されます。デフォルトの言語は、SALANG 環境変数またはインストーラによって指定されます (SALANG 変数の方が優先)。</p> <p>言語コードの詳細については、「ロケールの設定 (435 ページ)」を参照してください。</p> <p>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 のさまざまなツールとユーティリティから返されるメッセージにはデフォルト言語が使用されますが、サーバから返されるメッセージには接続の言語が使用されます。</p>
参照	「 接続パラメータの構造 」 (54 ページ)

LazyClose 接続パラメータ (LCLOSE)

機能	CLOSE <i>cursor-name</i> 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	なし
説明	<p>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 <i>cursor-name</i> データベース要求がキュー内にある間は、引き続き、独立性レベル 1 カーソルの安定性ロックがカーソルに適用されます。</p> <p>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。</p> <ul style="list-style-type: none">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 <p>めったにない状況ですが、CLOSE <i>cursor-name</i> データベース要求の後に次の要求をキャンセルした場合は、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、同じ名前での別のカーソルをオープンしようとすると失敗します。要求を頻繁にキャンセルするアプリケーションでは、LazyClose を使用しないことをお勧めします。</p>
参照	「接続パラメータの構造」(54 ページ)

LivenessTimeout 接続パラメータ (LTO)

機能	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	整数 (秒)
デフォルト	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。

ある接続上で `LivenessTimeout` 値の 3 分の 1 から 3 分の 2 の期間パケットが送信されないと、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。

サーバに対して 200 を超える接続が確立されているときは、サーバは自動的に、指定の `LivenessTimeout` 値に基づいてもっと大きな `LivenessTimeout` 値を計算します。その結果、より多くの接続がより効率よくサーバで処理できるようになります。

別の方法として、[ODBC Configuration] 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 秒に設定します。

```
LTO=60
```

LogFile 接続パラメータ (LOG)

機能 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 文字列

参照 [「接続パラメータの構造」 \(54 ページ\)](#)

説明 クライアント・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場合は、`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・メッセージをクライアントの現在の作業ディレクトリ内の `error.log` ファイルに送信することを指定しています。

```
...  
LogFile=error.log  
...
```

NewPassword 接続パラメータ (NEWPWD)

ファンクション	パスワードの有効期限が切れていても、ユーザは DBA の介入なしにパスワードを変更できます。
参照	『SQL Anywhere サーバ - データベース管理』

Password 接続パラメータ (PWD)

機能	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	パスワードの指定なし
参照	「EncryptedPassword 接続パラメータ (ENP)」 (122 ページ)
説明	<p>すべてのデータベース・ユーザにはパスワードがあります。ユーザがデータベースに接続できるようにするには、ユーザにパスワードを提供する必要があります。デフォルトでは、パスワードの大文字と小文字の区別はデータと同じです。IQ データベースは、デフォルトでは大文字と小文字を区別します。</p> <p>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。Sybase Central と Sybase IQ ODBC 設定ツールは、どちらも暗号化したパラメータを使用します。</p> <p>Password と EncryptedPassword の両方を指定した場合は、Password が優先されます。</p>
例	<p>次の接続文字列フラグメントでは、ユーザ ID に DBA、パスワードに SQL を指定します。</p>

```
uid=DBA;pwd=SQL
```

別の方法として、接続ウィンドウの [User ID] および [Password] 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)

ファンクション	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	整数値 [k m]
デフォルト	512 (KB)
参照	「PrefetchRows 接続パラメータ (PROWS)」

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 および PrefetchBuffer 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) 接続パラメータの値を増やすと、GET DATA 要求をバッファするために使用するメモリの量が増加します。GET DATA (SQLGetData) 要求を処理するアプリケーションの場合は、これによって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)

機能 データベースへの問い合わせ時にプリフェッチするローの最大数を設定します。

使用法 すべての場所

デフォルト 10 (.NET Data Provider を使用している場合は 200)

参照 [「PrefetchBuffer 接続パラメータ \(PBUF\)」](#)

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数回の絶対フェッチで数多くのロー (数百ロー以上) 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/サーバ通信がダイヤルアップ・リンクやワイド・エリア・ネットワークなどの低速ネットワークで行われている場合

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;...
```

Prompt 接続パラメータ (PROMPT)

有効期限が切れたパスワードを持つユーザがログインを試行すると、ユーザに新しいパスワードを表示し、そのパスワードで再接続します。

詳細については、『SQL Anywhere サーバ - データベース管理』を参照してください。

RetryConnectionTimeout 接続パラメータ (RetryConnTO)

サーバが見つからない場合は、指定した期間、接続をリトライし続けるようクライアント・ライブラリ (dblib、ODBC、ADO など) に指示します。

詳細については、『SQL Anywhere サーバ - データベース管理』を参照してください。

ServerName 接続パラメータ (ENG)

[「EngineName 接続パラメータ \(ENG\)」](#) と同意語です。

詳細については、『[「EngineName 接続パラメータ \(ENG\)」 \(121 ページ\)](#)』を参照してください。

StartLine 接続パラメータ (START)

機能

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

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。

使用できるコマンド・ライン・スイッチの詳細については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」を参照してください。

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)

機能	サーバへの接続があるときでも、 <i>dbstop</i> 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	なし
参照	『ユーティリティ・ガイド』の「 dbstop の構文 」
説明	<i>dbstop</i> 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。接続文字列中に Unconditional=Yes と指定すると、アクティブな接続があるときでもサーバは停止します。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	<ul style="list-style-type: none"> 次のコマンド・ラインでは、無条件にサーバを停止します。 <pre>dbstop -c "uid=DBA;pwd=SQL;eng=server-name;unc=yes"</pre>

Userid 接続パラメータ (UID)

機能	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	すべての場所 (DSN および FILEDSN 接続パラメータは冗長形のキーワードをサポートしていません)。
デフォルト	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では、ユーザ ID に DBA、パスワードに SQL を指定します。 <pre>uid=DBA;pwd=SQL</pre>

ネットワーク通信のパラメータ

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;. . .),...
```

クライアント側からは、通信パラメータは CommLinks 通信パラメータとして入力されます。

```
CommLinks=tcpip(PARM1=value1;PARM2=value2;. . .),...
```


パラメータがスペースを含む場合は、ネットワーク通信パラメータを二重引用符で囲んで、システム・コマンド・インタプリタによって正しく解析されるようにします。

```
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、二重引用符が必要です。

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。

上記の例では、1 行にすべてのコマンドを入力しています。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 および HTTPS
Broadcast (BCAST)	Certificate
BroadcastListener (BLISTENER)	Certificate_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)

機能	ブロードキャスト・アドレスを指定します。
使用法	TCP/IP
値	文字列 (IP アドレス形式)
デフォルト	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	<p>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 アドレスの中でネットワークを表す部分とホストを表す部分を示します。</p> <p>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。</p> <p>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。</p> <p>『SQL Anywhere サーバー データベース管理』>「データベースの起動とデータベースへの接続」>「クライアント/サーバ通信」>「TCP/IP プロトコルの使用」>「SQL Anywhere での IPv6 サポート」を参照してください。Sybase IQ では、Mac OS X プラットフォームはサポートされていません。</p>
例	<p>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。</p> <pre>LINKS=tcPIP (BROADCAST=ff02::1%2)</pre>

BroadcastListener 通信パラメータ (BLISTENER)

機能	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	YES
説明	値 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 を使用すると、TCP/IP で BroadcastListener=NO を指定するのと同じことになります。
参照	『ユーティリティ・ガイド』の「 データベース・サーバの起動 」
例	<ul style="list-style-type: none"> BroadcastListener=NO の TCP/IP 接続を受け入れるサーバを起動します。 <pre>start_iq -x tcPIP (BroadcastListener=NO)</pre>

Certificate 通信パラメータ

使用法	HTTP、HTTPS
値	文字列
デフォルト	デフォルトの証明書名なし
説明	このオプションでは、暗号化証明書の名前を指定できます。この証明書のパスワードは Certificate_Password パラメータで指定します。

- 例
- 特定の暗号化証明書を使用した Web 接続を必要とするサーバを起動します。

```
start_iq -xs
http(Certificate=cert.file;Certificate_Password=secret)
...
```

参照

[「Host パラメータ \(IP\)」 \(138 ページ\)](#)

[「DoBroadcast パラメータ \(DBROAD\)」 \(137 ページ\)](#)

[「ServerPort パラメータ \(PORT\)」 \(145 ページ\)](#)

Certificate_Password 通信パラメータ

使用法	HTTP、HTTPS
値	文字列
デフォルト	デフォルトの証明書パスワードなし
説明	このオプションでは、Certificate パラメータで指定された暗号化証明書に対応するパスワードを指定できます。

- 例
- 特定の暗号化証明書を使用した Web 接続を必要とするサーバを起動します。

```
start_iq -xs
http(Certificate=cert.file;Certificate_Password=secret)
...
```

ClientPort パラメータ (CPort)

使用法	TCP/IP。クライアント側のみ。
デフォルト	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を適用しない場合は、このパラメータを使用しないことをおすすめします。
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上、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します。

ClientPort オプションは、クライアント・アプリケーションが TCP/IP を使って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。

与えられたデータ・ソースまたは接続文字列を使用して複数の接続を確立するときの最善策は、ポート番号リストまたはポート番号の範囲を指定することです。ポート番号を1つだけ指定すると、アプリケーションは一度に1つの接続しか維持できません。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。ポート番号リストかポート番号の範囲またはその両方を指定すると、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、指定されたポート番号のすべてに対して試行を続けます。

- 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)

使用法 HTTP、HTTPS

値 AUTO、REQUIRED、*database-name*

デフォルト AUTO

説明 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。

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すが、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、Web 要求を処理するときにサーバ上のデフォルト・データベースが使用されます。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
- 次のコマンドでは、2つのデータベースを開始しますが、そのうち1つのデータベースだけは HTTP 経由でアクセスできるようにします。

```
start_iq -xs http(dbn=web) iqdemo.db web.db
```

DoBroadcast パラメータ (DBROAD)

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	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、サーバを検索するときにブロードキャストが実行されます。このブロードキャストは、最初にローカル・サブネットに到達します。HOST=を指定した場合は、指定した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) 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・サブネットに対してブロードキャストが実行されなくなります。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) 通信パラメータが必要です。

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。これはある特定の状況下では役に立ちますが、一般的には推奨できません。

例

- 次のコマンドを使ってクライアントを起動すると、データベース・サーバを検索するためのブロードキャストは実行されません。silver という名前のコンピュータ上でのみ、サーバが検索されます。

```
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 パラメータ (IP)

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
参照	「ClientPort パラメータ (CPort)」 (135 ページ)
場所	サーバ側とクライアント側
説明	<p>HOST は、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。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。</p> <p>TCP/IP の場合は、<i>hostname</i> 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。</p> <p>-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。</p> <p>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、<i>Host=nnn.nm.nnn.nnn;ServerPort=pppp</i> という形式で、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。</p> <p>TCP/IP を使用しているときは、IP と HOST は同じ意味になります。</p> <p>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。</p> <p>『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「クライアント/サーバ通信」>「TCP/IP プロトコルの使用」>「SQL Anywhere での IPv6 サポート」を参照してください。Sybase IQ では、Mac OS X プラットフォームはサポートされていません。</p>
値	文字列
デフォルト	追加のマシンなし
例	<ul style="list-style-type: none"> • 次の接続文字列フラグメントは、“kangaroo” と 197.75.209.222 (ポート 2369) というマシンを検索して、<i>iqdemo</i> というデータベース・サーバを見つけるようにクライアントに命令します。 <pre>...ENG=iqdemo CommLinks=tcPIP(IP=kangaroo;IP=197.75.209.222:2369)</pre> <ul style="list-style-type: none"> • 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。 <pre>dbisql -x "tcPIP(HOST=kangaroo;HOST=197.75.209.222)" iqdemo</pre> <ul style="list-style-type: none"> • 次の接続文字列フラグメントは、<i>my_server</i> と <i>kangaroo</i> 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。 <pre>dbisql -c "UID=DBA;PWD=sql;LINKS=tcPIP(HOST=my_server,kangaroo;PORT=2639)"</pre>

LDAP 通信パラメータ (LDAP)

使用法	TCP/IP (サーバ側のみ)
値	YES、NO、または <i>filename</i>
デフォルト	ON デフォルトの <i>filename</i> は <i>asaldap.ini</i> 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 [<i>dblocate</i>]) が LDAP サーバにクエリを発行できるようにします。これにより、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、Locate ユーティリティ (<i>dblocate</i>) もこのようなサーバを検索できるようになります。 LDAP= <i>filename</i> を指定すると、LDAP サポートが有効になり、指定のファイルが設定ファイルとして使用されます。LDAP=YES を指定すると、LDAP サポートが有効になり、 <i>saldap.ini</i> が設定ファイルとして使用されます。 LDAP は TCP/IP でのみ使用され、ネットワーク・サーバ上にかざられています。
参照	詳細については、SQL Anywhere サーバ - データベース管理の「LDAP サーバを使用した接続」を参照してください。

LocalOnly 通信パラメータ (LOCAL)

使用法	TCP/IP、HTTP、HTTPS
値	YES、NO
デフォルト	なし
説明	LocalOnly (LOCAL) 通信パラメータを指定すると、ローカル・マシン上にサーバがある場合は、クライアントがそのサーバにのみ接続します。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 [<i>dblocate</i>] 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。
参照	「Broadcast 通信パラメータ (BCAST)」 (133 ページ)

Logfile 接続パラメータ (LOG)

使用法	HTTP、HTTPS
値	ファイル名
デフォルト	なし
説明	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。
参照	「LogFormat 通信パラメータ (LF)」 (140 ページ) 「LogMaxSize 通信パラメータ (LSIZE)」 (141 ページ) 「LogOptions 通信パラメータ (LOPT)」 (141 ページ)

LogFormat 通信パラメータ (LF)

使用法	HTTP、HTTPS
値	フォーマット文字列
デフォルト	@T - @W - @I - @P - "@M @U @V" - @R - @L - @E
説明	<p>このパラメータは、ログ・ファイルに書き込むメッセージのフォーマットと、どのフィールドを書き込むかを制御します。次に示すフィールド・コードがフォーマット文字列に含まれている場合は、メッセージを書き込むときに、コード部分が現在の値に置き換えられます。</p> <ul style="list-style-type: none">• @@ @ 文字• @B 要求の処理を開始した日付と時刻 (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)• @C クライアントが接続した日付と時刻• @D 要求に関するデータベースの名前• @E エラー・メッセージのテキスト (エラーが発生した場合)• @F 要求の処理が完了した日付と時刻• @I クライアントの IP アドレス• @L 応答の長さ (バイト単位、ヘッダと本体を含む)• @M HTTP 要求メソッド• @P 要求に関するリスナ・ポート• @Q 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)• @R HTTP 応答のステータス・コードと説明• @S HTTP ステータス・コード

- **@T** 現在のログ・エントリが書き込まれた日付と時刻
- **@U** 要求された URI
- **@V** 要求された HTTP のバージョン
- **@W** 要求の処理にかかる時間 (@F ~ @B)。エラーのために要求が処理されなかった場合は 0.000

参照

[「Logfile 接続パラメータ \(LOG\)」 \(140 ページ\)](#)

[「LogMaxSize 通信パラメータ \(LSIZE\)」 \(141 ページ\)](#)

[「LogOptions 通信パラメータ \(LOPT\)」 \(141 ページ\)](#)

LogMaxSize 通信パラメータ (LSIZE)

説明

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。

LogMaxSize パラメータは、SQL Anywhere プロトコル・オプションです。『SQL Anywhere サーバ - データベース管理』 > 「データベースの起動とデータベースへの接続」 > 「接続パラメータとネットワーク・プロトコル・オプション」 > 「ネットワーク・プロトコル・オプション」 > 「LogMaxSize プロトコル・オプション [LSIZE]」を参照してください。

LogOptions 通信パラメータ (LOPT)

使用法

HTTP, HTTPS

値

NONE, OK, INFO, ERRORS, ALL, *status-codes*, REQHDRS, RESHDRS, HEADERS

デフォルト

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。

- **NONE** 何もログに記録しません。
- **OK** 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)。
- **INFO** 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)。
- **ERRORS** 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)。
- **ALL** すべての要求をログに記録します。

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。

- **C200** OK
- **C400** Bad request
- **C401** Unauthorized
- **C403** Forbidden
- **C404** Not found
- **C408** Request timeout
- **C501** Not implemented
- **C505** Service unavailable

さらに、次のキーワードを指定すると、メッセージに関する詳細情報も記録できます。

- **REQHDRS** 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。
- **RESHDRS** 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。
- **HEADERS** 要求を記録するときに、要求ヘッダと応答ヘッダもログ・ファイルに書き込みます (REQHDRS と RESHDRS の両方を指定した場合と同じ)。

参照

[「Logfile 接続パラメータ \(LOG\)」 \(140 ページ\)](#)

[「LogFormat 通信パラメータ \(LF\)」 \(140 ページ\)](#)

[「LogMaxSize 通信パラメータ \(LSIZE\)」 \(141 ページ\)](#)

MaxConnections 通信パラメータ (MAXCONN)

使用法

HTTP、HTTPS

値

数値

デフォルト

ライセンスのある接続の数

説明

サーバが受け付ける同時接続の最大数を指定します。0 を指定すると、無制限になります。

参照

[「MaxRequestSize 通信パラメータ \(MAXSIZE\)」 \(143 ページ\)](#)

MaxRequestSize 通信パラメータ (MAXSIZE)

説明 データベース・サーバが受け入れることのできる要求の最大サイズを指定します。

MaxRequestSize パラメータは、SQL Anywhere プロトコル・オプションです。『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「接続パラメータとネットワーク・プロトコル・オプション」>「ネットワーク・プロトコル・オプション」>「MaxRequestSize プロトコル・オプション [MAXSIZE]」を参照してください。

MyIP 通信パラメータ (ME)

使用法 TCP/IP、HTTP、HTTPS

値 文字列

説明 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。

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、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 番号としてキーワード NONE を指定した場合は、アドレス情報の特定処理が行われません。NONE キーワードは、この処理を行うとコストがかかるオペレーティング・システム上のクライアント (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) を主な対象としています。サーバで使用するための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。

『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「クライアント/サーバ通信」>「TCP/IP プロトコルの使用」>「SQL Anywhere での IPv6 サポート」を参照してください。

- 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-15_2¥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-15_2¥demo¥iqdemo.db"
```

- 次の接続文字列フラグメントによって、クライアントがアドレス情報を決定しないようにします。

```
...CommLinks= tcpip(MyIP=NONE)...
```

PreFetchOnOpen 通信パラメータ

使用法 ODBC

値 YES、NO

デフォルト なし

説明 このオプションを有効にすると、プリフェッチ要求とカーソル・オープン要求と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。オープン時にプリフェッチを行うためには、カラムがすでにバインドされている必要があります。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

ReceiveBufferSize 通信パラメータ (RCVBUFSZ)

使用法 TCP/IP

値 整数値 [k|m|g]

デフォルト マシンによって異なる。

説明 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。k、m、または g を使用して、それぞれキロバイト、メガバイト、またはギガバイトの単位を指定します。

SendBufferSize 通信パラメータ (SNDBUFSZ)

使用法 TCP/IP

値 整数値 [k|m|g]

デフォルト コンピュータによって異なる。

説明 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。デフォルト値はバイト単位ですが、k、m、または g を使用して、それぞれキロバイト単位、メガバイト単位、ギガバイト単位を指定できます。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。

参照 『SQL Anywhere サーバ - データベース管理』>「データベースの起動とデータベースへの接続」>「クライアント/サーバ通信」>「TCP/IP プロトコルの使用」を参照してください。サーバに RSA 暗号化機能を提供する RSA セキュリティ・モジュールは、Sybase IQ に無料で提供されていますが、Kerberos、ECC、FIPS の各セキュリティ・モジュールについては別途ライセンスが必要です。ライセンスの詳細については、『インストールおよび設定ガイド』を参照してください。

ServerPort パラメータ (PORT)

使用法 TCP/IP (すべてのプラットフォーム)、HTTP、HTTPS

値 整数値

デフォルト 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、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用する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。

このデータベース・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。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。

IQ データベース・サーバは常にポート 2638 で受信します (ネットワーク通信パラメータを使用して別のポートを指定した場合も、ポート 2638 を使用します)。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、IQ データベース・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 および 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 IQ ネットワーク・サーバを起動します。

```
start_iq -x tcpip c:¥sybase¥IQ-15_2¥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-15_2¥demo¥iqdemo2.db
```

これは、ポートが現在割り当て済みであるため、Server failed to start. Possible cause: - port number invalid or already in use というエラー・メッセージが表示されて失敗します。

-z スイッチを使用している場合にも、Couldn't bind to specified address というエラー・メッセージが表示されます。

- 3 別のデータベース・サーバを起動して、異なるポート番号を割り当てます。

```
start_iq -x "tcpip(ServerPort=2639)"  
c:¥sybase¥IQ-15_2¥demo¥iqdemo2.db
```

2639 が予約ポートでなく、他のアプリケーションが割り当てられていないかぎり、正常に処理されます。

- 4 マシン上の別の Web サーバがすでにポート 80 を使用している場合、またはこの下位ポート番号でサーバを起動するパーミッションを持っていない場合は、サーバの開始時に別の受信ポート (たとえば 8080 など) を指定できます。

```
start_iq -xs http{port=8080} -n server3 web.db
```

Sessions パラメータ

使用法

NetBIOS。サーバ側のみ。

説明

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。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。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 オペレーティング・システムによって異なります。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 パラメータ

使用法 TCP/IP、NamedPipes。サーバ側のみ。

値 YES、NO

デフォルト YES

説明 データベース・サーバへの TDS 通信を許可しないようにするには、TDS を NO に設定します。暗号化された接続だけをサーバに許可したい場合は、これらのポート・オプションで TDS 接続を拒否する必要があります (これが唯一の方法です)。

例

- 次のコマンドは、Open Client または jConnect アプリケーションからの接続は許可しないで、TCP/IP プロトコルを使ってデータベース・サーバを起動します。

```
start_iq -x tcpip(TDS=NO) ...
```

Timeout パラメータ (TO)

使用法 TCP/IP (すべてのプラットフォーム)、HTTP、HTTPS

値 整数 (秒)

デフォルト 5

説明 TIMEOUT は、通信確立時および通信切断時の応答待ち時間を秒単位で設定します。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、要求を受け取る時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)

使用法 TCP/IP (クライアント側のみ)

値 YES、NO

デフォルト YES

説明 この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。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。

DoBroadcast=NONE パラメータを使用して TCP で接続するときは、クライアントは TCP 接続を確立した後に、見つかったサーバの名前が目的のサーバと同じかどうかを検証します。VerifyServerName=NO が設定されている場合は、この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 IQ サーバの IP アドレス/ポートしか知らない場合でも、そのポートに接続できます。

接続文字列にはサーバ名が指定されているはずですが、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

クライアント・マシン上でこのオプションを使用するには、ODBC Administrator内で新しい 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、すべてのクエリ・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=ip_address:port#;DOBROADCAST=NONE;VERIFY=NO
```

次に例を示します。

```
host=123.456.77.888:2222;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。

この章について

この章では、データベースと、テーブル、ビュー、インデックスなどのデータベース・オブジェクトを作成、変更、および削除する方法について説明します。これ以外にサポートされているデータベース・オブジェクトには、プロシージャ(『システム管理ガイド第2巻』を参照)とログイン・ポリシー(「第8章 ユーザ ID とパーミッションの管理」を参照)の2つのタイプがあります。

注意 Sybase IQ はカタログ・ストアと IQ ストアの両方から構成されます。この章では、両方のストアと、IQ ストア内のオブジェクトの作成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 のテーブル特性を持っています。カタログ・ストア内にテーブルを作成する場合は、SQL Anywhere のマニュアルを参照する必要があります。

内容

トピック	ページ
Sybase IQ データベースの作成	151
データベース・オブジェクトの使用	168
DB 領域の使用	181
DB 領域管理の例	191
テーブルの編集	196
ビューの編集	205
インデックスの編集	209

Sybase IQ データベースの作成

この項では、データベースを作成するための手順とツールについて紹介します。データの格納先、必要な領域サイズ、データベース・オブジェクトの定義や修正の権限者など、決定する必要のある事項についても説明します。

データベースの設計

データベースを作成する前に、まずしっかりした設計をすることが重要です。データベース設計が正しいと、データの有用性と検索速度が大きく異なります。

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、IQ データベースの基本となる概念を理解する必要もあります。

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ことを目的として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。この章で説明する手順に従って IQ データベースを作成する前に、『Sybase IQ の概要』で説明されているリレーショナル・データベースとデータ・ウェアハウスの概念を理解しておいてください。

データベース・オブジェクトの編集ツール

Sybase IQ には、データベース・オブジェクトを編集するユーティリティとして、Sybase Central と dbisql の 2 つが含まれています。また、PowerDesigner もデータ・ウェアハウス全体の設計や作成に使用できます。

Sybase Central を使用したデータベース・オブジェクトの編集

Sybase Central は、ウィンドウ機能を持つシステム上でデータベース・オブジェクトの編集を行うための主要ツールです。Sybase Central では、サーバ、データベース、dbspace を管理できます。テーブル、プロシージャ、ビュー、インデックス、ユーザ、グループなど、あらゆる種類のデータベース・オブジェクトの作成、修正、削除を行うことができます。Sybase Central をマルチプレックス・サーバで使用する方法については、『Sybase IQ Multiplex の使用』を参照してください。

この章では、データベース・オブジェクトを編集するための SQL 文について説明します。Sybase Central を使用している場合、SQL 文は自動的に作成されます。Sybase Central に関する主な情報源は、Sybase Central のオンライン・ヘルプです。この章では、Sybase Central を使用して実行できるタスクについての説明は最小限にとどめます。

Sybase Central の使用方法の概要については、『Sybase IQ の概要』の「[第 4 章 Managing Databases](#)」を参照してください。

Interactive SQL を使用したデータベース・オブジェクトの編集

Interactive SQL (`dbisql`) は、SQL 文を入力するためのユーティリティです。Interactive SQL を使用してデータベース・スキーマを編集する場合は、SQL 文を 1 つずつ実行するのではなく、`dbisql` コマンド・ファイル内にコマンドのセットを作成します。その後、`dbisql` でこのファイルを実行して、データベースを作成できます。

データベース・オブジェクトの定義はデータベース・スキーマを形成します。スキーマは、空のデータベースと考えることができます。スキーマの作成と修正を行う SQL 文は、「データ定義言語」(DDL) と呼ばれます。

注意 1 つのテーブルに対しては一度に 1 人のユーザしか DDL 文を実行できません。テーブルに対して DDL 操作が行われている間は、そのテーブルがロックされます。ただし、同じデータベース内の別のオブジェクトに対する DDL 文ならば同時に実行できます。詳細については、「[ロックの仕組み](#)」(408 ページ) を参照してください。

`dbisql` 以外のツールを使用する場合も、この章の SQL 文に関する説明が適用されます。

`dbisql` コマンド・ファイル

`dbisql` コマンド・ファイルは、次に示すようにコマンドの末尾にセミコロンが付いたテキスト・ファイルです。

```
CREATE TABLE t1 ( .. );
CREATE TABLE t2 ( .. );
CREATE LF INDEX i2 ON t2 ( .. );
..
```

通常、`dbisql` コマンド・ファイルには、拡張子 `.sql` が付きます。コマンド・ファイルを実行するには、ファイルの内容を `dbisql` コマンド・ウィンドウに貼り付けるか (500 行以内の場合)、ファイルをコマンド・ウィンドウに読み込むためのコマンドを入力します。たとえば、次のコマンドは、

```
read makedb
```

`makedb.sql` ファイル内の `dbisql` コマンドを読み込みます。

コマンド・ウィンドウにファイルを読み込む方法の詳細については、『リファレンス：文とオプション』の「[READ 文 \[DBISQL\]](#)」を参照してください。

データベース設定手順の概要

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。この項では、IQ データベースとそのオブジェクトを設定する手順の概要を示します。

❖ IQ データベースの設定

- 1 Sybase IQ をインストールし、設定します。

この手順で、クライアントおよびサーバ環境と `iqdemo` データベースをインストールします。詳細については、プラットフォームに対応した『インストールおよび設定ガイド』を参照してください。

- 2 IQ データベースを作成します。

この手順で、IQ ストアとカタログ・ストアの両方が作成されます。Sybase Central、CREATE DATABASE SQL 文、または `iqinit` ユーティリティを使用できます。

マルチプレックス機能を簡単に使用できる Sybase Central の詳細については、「[Sybase Central を使用した IQ データベースの作成](#)」(171 ページ)を参照してください。

CREATE DATABASE の手順については、「[SQL によるデータベースの作成](#)」(168 ページ)を参照してください。

`iqinit` については、『ユーティリティ・ガイド』の「[第 5 章 データベース・オブジェクトの使用](#)」の「[初期化ユーティリティ \(iqinit\)](#)」を参照してください。

- 3 IQ データベースにテーブルを作成します。

CREATE TABLE 文、または Sybase Central のテーブル作成ウィザードを使用します。詳細については、「[テーブルの編集](#)」(196 ページ)を参照してください。

- 4 テーブルのインデックスを作成します。

CREATE INDEX 文、または Sybase Central 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。詳細については、「[第 6 章 Sybase IQ インデックスの使用](#)」を参照してください。

- 5 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。詳細については、「[第 7 章 データベースへのデータの入出力](#)」を参照してください。

データ定義タスクのスケジュール

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に、さらなるデータ定義操作(テーブルやインデックスの追加/修正など)が必要になったときには、次のガイドラインに従います。

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、短時間であっても、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。テーブルがジョイン・インデックスの一部の場合、データ定義操作が完了するまでは、ユーザはジョイン・インデックスのどのテーブルに対しても読み込み／書き込みを実行できません。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。

パフォーマンス・モニタリング用のダミー・テーブルの作成

IQ バッファ・キャッシュ・モニタを開始するには、永久テーブルまたはテンポラリ・テーブルを指定する必要があります。できれば、モニタリングのためだけに使用するダミー・テーブルを指定したほうがよいでしょう。パフォーマンス・モニタの詳細については、『パフォーマンス&チューニング・ガイド』の「[第5章 パフォーマンスのモニタリングとチューニング](#)」を参照してください。

データ定義権限の拡張

データ定義タスクを実行するには、適切な権限が必要です。

- DBA 権限があれば、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。これには、DBA 権限を他のユーザに付与できる権限も含まれます。
- 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 RESOURCE 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。

DBA 権限または PERMS ADMIN 権限を持つユーザとオブジェクト所有者は、個々のユーザやユーザ・グループに権限を付与できます。詳細については、「[第8章 ユーザ ID とパーミッションの管理](#)」を参照してください。-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・タイプの選択

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、これらのデバイスとしてオペレーティング・システム・ファイルを使用できます。また、ロー・パーティションと呼ばれるディスク部分も使用できます。データベースの作成時に、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決定します。

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。一方、開発環境では、デバイスの管理が容易になるためファイル・システムのほうが望ましい場合があります。

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。

データベース用の領域の割り付け

すべての 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。DB 領域には、次の3つのタイプがあり、それぞれ特定のタイプの Sybase IQ データを格納するよう設計されています。

- IQ メイン・ストア
- IQ テンポラリ・ストア
- IQ カタログ・ストア

DB 領域のタイプ

DB 領域には、次の6つのタイプがあり、それぞれ特定のタイプの Sybase IQ データを格納するよう設計されています。

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
システム 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

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_MAIN	IQ データベースは、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 IQ 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 メタデータなどで構築されています。IQ ユーザ・オブジェクトは、ここに格納できますが、他のメイン DB 領域に格納することをおすすめします。	1 個または複数	1 個または複数
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、ジョイン・インデックス、テーブル・メタデータなどの IQ オブジェクト	1 個または複数	1 個または複数
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 dbfile のセット	1 個または複数	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

データベースを開始するには、特定のファイルが使用可能でなくてはなりません。詳細については、「[データベースの起動に関するガイドライン](#)」(49 ページ)を参照してください。

テーブルまたはジョイン・インデックスの 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・テーブルとジョイン・インデックスの場合は、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* 句または CREATE JOIN INDEX 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 領域に作成されますが、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。

カタログ・ストア

これらのテーブルは、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、IQ 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。

システム DB 領域 システムという名前の IQ カatalog DB 領域は、IQ データベース用のメタデータを含んでおり、テーブルと同じフォーマットで SQL Anywhere リレーショナル・データベース・システムに格納されています。SQL Anywhere は、IQ の有無にかかわらず存在できるリレーショナル・データベース・システムです。カタログ・ストアには、IQ テーブルとともに SQL Anywhere 形式のテーブルを持つことができます。つまり、別の SQL Anywhere データベースを持つことができます。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。

他のカタログ DB 領域 システム DB 領域とは別の DB 領域に SQL Anywhere テーブルを作成できます。

IQ_SYSTEM_MAIN DB 領域

IQ_SYSTEM_MAIN DB 領域は、データベースの作成時や古い IQ データベースの Sybase IQ 15.2 へのアップグレード時に作成されます。IQ_SYSTEM_MAIN は、データベースがオープンするために必要な構造体 (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 IQ 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、データベース領域および identity メタデータ) を格納する特別な DB 領域です。IQ_SYSTEM_MAIN は常に、データベースが開いているときはオンラインです。

IQ_SYSTEM_MAIN のサイズ調整のガイドラインについては、[表 5-1 \(161 ページ\)](#) を参照してください。

他のユーザ・メイン DB 領域

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。管理者は、特に、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、IQ_SYSTEM_MAIN にユーザ・テーブルを作成することを許可できます。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。

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';
```

IQ テンポラリ DB 領域

IQ テンポラリ・ストア用の1つのDB領域であるIQ_SYSTEM_TEMPは、データベースの作成時や古いIQデータベースをSybase IQ 15.2にアップグレードしたときに作成されます。

各IQ DB領域には、ファイルをいくつでも含めることができます。IQファイルの合計数だけが16384個に制限されています。

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dbspaceを作成できます。

テーブルを作成しロードすると、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分配します。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。dbspaceのサイズは、予約した最大領域まで拡大できます。また、dbspaceを小さくすることもできます。その場合は、dbspace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必要があります。個々のデータベース・オブジェクトは、必要に応じて特定のdbspaceから移動させることができます。

すべてのディスク領域をIQ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Sybase IQで適切に処理するために必要です。

古い DB 領域が満杯となったら DB 領域を徐々に追加するのではなく、可能であれば、データベースの作成時にすべての DB 領域を作成します。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。

一般には、デバッグ用の独立したデータベースを作成することをおすすめします。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るようにします。

IQ ストアとテンポラリー・ストアに必要な領域

IQ 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、IQ 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。詳細については、「[ディスク領域の不足](#)」(537 ページ)を参照してください。

Sybase IQ は、明示的に定義したテンポラリー・テーブルに加えて、テンポラリー・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)。テンポラリー・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ー・ストアのサイズ調整ガイドライン

Sybase IQ 15.2 アーキテクチャでは、データ記憶領域に影響を与える次のような変更が加えられています。

- 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 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、ノードがバックアップしたときにバージョンと同期させるため、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。
- Sybase IQ 15.2 では、より多くの処理を並列に実行するため、以前のリリースよりも多くのテンポラリー領域を使用します。

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変化
- 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ための容量

Sybase では一般的なガイドラインを提供できますが、これらの要因が組み合わさると各データベースの要件が異なってきます。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。運用データベースのサイズのガイドラインについては、表 5-1 を参照してください。

表 5-1: 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	注意事項
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 単位です。 iqunload -n を使用するには、サーバは 12.7 ESD #5 以上である必要があります。
新しい運用データベースの作成	<ul style="list-style-type: none"> データベースをマイグレートする場合は、現在の IQ_SYSTEM_MAIN にロー・デバイスを使用し、標準サイズの新しい未使用のロー・デバイスを割り付けます。 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ります。 IQ_SYSTEM_MAIN にロー・ディスクを使用する場合は、可能なかぎりマルチプル・ロー・ディスクを使用します。マルチプル・ロー・ディスクを使用すると、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します。 ファイル・システムが、フォールト・トレラントで、高いパフォーマンスの冗長ディスク・アレイ (RAID 5 など) で実装されている場合、運用では IQ DB 領域にファイル・システムのファイルのみを使用してください。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ますが、マルチプレックス・システムでは、理想的に Storage Area Network デバイス上でクラスタ・ファイル・システムが必要です。 	ロー・デバイスを指定する場合は、ms_size を省略します。 メイン用の予約は常に IQ_SYSTEM_MAIN サイズの 20 パーセントに設定します。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 を使用します。
マルチプレックス用のメイン・ストアの作成	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。	

例

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操作、コミット操作、チェックポイント操作で使用される小さいが重要なデータ構造体用に 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。Sybase では、MAIN_RESERVED_DBSPACE_MB オプションの値を IQ_SYSTEM_MAIN SIZE の 20 パーセントに設定することをおすすめします。MAIN_RESERVED_DBSPACE_MB の詳細については、「[IQ メイン・ストアと IQ テンポラリ・ストアの領域管理](#)」(178 ページ)を参照してください。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・コマンドで指定できます。-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。移行の詳細については、『インストールおよび設定ガイド』を参照してください。

たとえば、次の文はロー・デバイス上に IQ_SYSTEM_MAIN を作成します。

```
iqunload -au -ms_filename /dev/rdsk/clt0d1 -c
"UID=DBA;PWD=SQL;DBF=latest.db"
```

たとえば、次の文は、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"
```

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 dbspace が満杯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、dbspace の要件を見積もるようにします。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、Sybase IQ のストアド・プロシージャを実行します。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「[第 7 章 システム・プロシージャ](#)」を参照してください。

ストアド・プロシージャを次の順序で実行すると、オブジェクトの領域不足を回避するのに役立ちます。

- 1 ストアド・プロシージャ `sp_iquestspace` を実行して、データベースの作成に必要な領域の量を、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見積もります。作成する各テーブルに対して 1 回ずつ、このプロシージャを実行します。構文は次のとおりです。

```
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 “RAW DATA index_size” の合計を足すと、すべてのテーブルの総合計になります。
- 3 ストアド・プロシージャ `sp_iqestjoin` を実行して、頻繁にジョインするテーブル上にジョイン・インデックスを作成するために必要な追加領域の量を見積もります。各テーブルの組み合わせに対して1回ずつ、このプロシージャを実行します。構文は次のとおりです。

```
sp_iqestjoin table1, table1rows, table2, table2rows
    [,relation] [,iqpagesize] ...
```

`sp_iqestjoin` が示すインデックス・サイズは、クエリによって異なります。

`sp_iqestjoin` を実行するたびに、提示されたインデックス・サイズの1つを選択します。テーブルを常に1対1でジョインすることがわかっている場合は、“Min Case index_size”を使用します。1対多のジョインをときどき行うことが予想される場合は、“Avg Case index_size”を使用します。1対多のジョインを多数使用することが予想される場合は、“Max Case index_size”を使用します。

- 4 すべてのテーブルの組み合わせに対して選択した `index_size` を合計します。
- 5 手順4のジョイン領域の合計を、手順2のテーブル領域の合計に足します。最小ジョイン領域と最大ジョイン領域に分けて計算します。
- 6 ストアド・プロシージャ `sp_iqestdbspaces` を実行して、指定領域から作成するDB領域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最大合計サイズと最小合計サイズを、このプロシージャの `minsize` および `maxsize` パラメータとして次のように使用します。

```
sp_iqestdbspaces (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。これらのストアド・プロシージャの詳細については、『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「[第7章 システム・プロシージャ](#)」を参照してください。

dbfile と出力ファイル

SYSDBFILE ビューには、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 領域とテンポラリ DB 領域にある dbfile、トランザクション・ログ・ファイル、および SA テンポラリ・ファイルを含む、使用しているデータベースにあるすべての dbfile が表示されます。

dbfile 以外のファイルは、SYSDBFILE 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 の切り替えで指定されます) 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 \(529 ページ\)](#) を参照)。

範囲分割

「分割」は、レンジ・オブジェクトをサブオブジェクトに分けるスキームです。次に例を示します。

- 記憶領域は、テーブル領域に分割されます。
- テーブルは、テーブル・パーティションに分割できます。

「テーブル・オブジェクト」は、テーブル・パーティション、カラム、インデックス、IQ ベース・テーブル、ジョイン・インデックス、一意性制約、プライマリ・キー、または外部キーです。

「テーブル・パーティション」はローの集まりであり、ユーザが作成したテーブルのサブセットです。特定のローは、2つの異なるパーティションに配置できません。各パーティションは、独自の DB 領域に配置して、個々に管理できます。パーティションは、親テーブルの次のような論理属性を共有します。

- 整合性制約とデフォルトが同じであるカラム定義
- 同じ参照整合性制約
- 同じ一意性制約とプライマリ・キー制約
- 同じ検査制約

テーブル作成者は、テーブルを分割するかどうか、分割する方法、および分割する数を選択します。テーブル作成者は、パーティション・キー、つまりテーブルを分割する方法を決定するテーブル・カラムを定義します。構文については、『リファレンス：文とオプション』の「[CREATE TABLE 文](#)」と「[ALTER TABLE 文](#)」を参照してください。

分割は、サブオブジェクトを読み取り専用にして管理することが基本的な目的です。サブオブジェクトを読み取り専用を設定して確定し、バックアップすると、メンテナンスは最小限ですみます。

DB 領域管理とテーブル分割の利点

分割と DB 領域管理を行うと、テーブル・レベルまたはデータベース・レベルよりも細かく管理操作（データ配置、dbcc、バックアップ、リストア）を実行できます。

Sybase IQ 15.2 では、DB 領域管理と範囲テーブル・パーティションで次の機能を使用できます。

- データ配置機能
- 階層状の記憶領域の管理（安価な記憶領域への重要性の低いデータの移動をサポート）

DB 領域管理と範囲テーブル・パーティションでは、次の機能によって保全性と有用性が向上します。

- DB 領域の読み取り専用 (RO) 状態と読み込み／書き込み (RW) 状態のサポート
- DB 領域のオンライン状態とオフライン状態のサポート

- 1つ以上の RO DB 領域またはファイル、またはすべての RW ファイルをバックアップまたはリストアすることによるバックアップ/リストアの時間短縮
- DB 領域またはテーブル・パーティション・ターゲットでのデータ検証のサポート (『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第7章 システム・プロシージャ」の「[sp_iqcheckdb プロシージャ](#)」を参照)
- Sybase IQ サーバが使用できない DB 領域、または「オフライン」とマークされて使用不可能な DB 領域で起動できる機能。ただし、カタログ・ストアと Sybase IQ システム DB 領域は除きます。

分割を行うと、記憶領域とデータがビジネス要件に応じて編成されるため、特にバックアップ、リストア、データベース検証などの管理およびランタイム操作が向上します。

Sybase IQ 15.2 の「範囲分割」では、1つのテーブル・カラムの値に基づいて、テーブルが論理パーティションに分割されます。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、ユーザは各パーティションに名前を付ける必要があります。

Sybase IQ では、最大 1024 個のパーティションに範囲分割できます。

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。たとえば、`June_2009`、`July_2009` のようにします。範囲テーブル・パーティションは、DB 領域に割り付けることができます。

パーティションでの DDL 操作

テーブル・パーティションの削除、名前の変更、分割、分割の解除、マージ、分割には、次の制限事項があります。

操作	制限事項
削除	パーティション・キー・カラムまたは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ません。
名前の変更	N/A
分割されていないテーブルの分割	N/A
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ります。データを移動する必要はありません。

操作	制限事項
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どちらかに属する必要があります。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ありません。
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・ページに移動されます。新しいDB領域でCREATEパーティションが必要です。

分割されたテーブルをジョイン・インデックスの一部にすることはできません。

パーティションでのDML操作

範囲分割では、テーブル・パーティションのロード、挿入、削除、トランケートなどのDML操作を実行できます。パーティション・キー・カラム以外の更新はサポートされています。

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DML操作を行うとエラーが返されます。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、文の挿入やロード、更新可能カーソルによる挿入を行うと、エラーが生成され、操作がロールバックされます。

分割されたテーブルでロードする場合、START ROW ID オプションを使用することはできません。分割されたテーブルでAPPEND_LOAD オプションを使用してロードすると、新しいローは適切なパーティションの末尾に追加されます。

考慮事項の詳細については、「[分割されたテーブルのロード](#)」(286 ページ)を参照してください。

分割されていないテーブルのオブジェクト配置

オブジェクトの作成時やオブジェクトの移動時に、テーブル・オブジェクト(データ型のカラム、インデックス、プライマリ・キー、外部キー、一意性制約、ジョイン・インデックス、分割されていないテーブルなど)に対してDB領域を指定できます。DB領域にCREATE権限が必要です。例を示します。

```
CREATE TABLE tabl (
    col1 INT IN dsp1,
    col2 VARCHAR(20),
    col3 CLOB IN dsp2,
    col4 DATE,
    col5 BIGINT,
    PRIMARY KEY (col5) IN dsp4) IN dsp3;
CREATE DATE INDEX col4_date ON tabl(col4) IN dsp5;
```

データ配置の結果は次のようになります。

DB 領域名	データ
dsp1	col1 の FP インデックス
dsp2	col3 の FP インデックス
dsp3	col2、col4、および col5 の FP インデックス
dsp4	プライマリ・キー (col5 の HG)
dsp5	DATE インデックス col4_date

分割されたテーブルのオブジェクト配置

分割されたテーブルの場合は、個々の DB 領域に各テーブル・パーティションを配置できます。また、個々の DB 領域にテーブル・パーティションの各カラムを配置することもできます。ただし、一般的に、個々の DB 領域は、BLOB または CLOB カラム、または 255 バイトよりも大きい CHAR、VARCHAR、VARBINARY のカラムのみに推奨されます。例を示します。

```
CREATE TABLE tab2(
  col1 INT IQ UNIQUE(65500),
  col2 VARCHAR(20),
  col3 CLOB PARTITION (p1 IN dsp11, p2 IN dsp12,
    p3 IN dsp13),
  col4 DATE,
  col5 BIGINT,
  col6 VARCHAR(500) PARTITION (p1 IN dsp21,
    p2 IN dsp22),
  PRIMARY KEY (col5) IN dsp2) IN dsp1
PARTITION BY RANGE (col4)
(p1 VALUES <= ('2006/03/31') IN dsp31,
 p2 VALUES <= ('2006/06/30') IN dsp32,
 p3 VALUES <= ('2006/09/30') IN dsp33
);
```

```
CREATE DATE INDEX c4_date ON tab2(col4) IN dsp3;
```

データ配置の結果は次のようになります。

パーティション	DB 領域	データ
p1	dsp11	col3 の FP インデックス (CLOB データ)
	dsp21	col6 の FP インデックス (VARCHAR(500) データ)
	dsp31	col1、col2、col4、および col5 の FP インデックス
p2	dsp12	col3 の FP (CLOB データ)
	dsp22	col6 の FP (VARCHAR(500) データ)
	dsp32	col1、col2、col4、および col5 の FP インデックス

パーティション	DB 領域	データ
p3	dsp13	col3 の FP インデックス (CLOB データ)
	dsp33	col1、col2、col4、col5、および col6 の FP インデックス (varchar(500) データ)
分割なし	dsp1	col1 のルックアップ・ストアと他の共有データ (すべてのパーティションの場合)
分割なし	dsp2	col5 のプライマリ・キー HG (すべてのパーティションの場合)
分割なし	dsp3	DATE インデックス col4_date (すべてのパーティションの場合)

データベース・オブジェクトの使用

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。

この章では、データベース・オブジェクトを定義する SQL 文について説明します。dbisql などの対話型 SQL ツールからデータベースを構築する場合は、SQL 文を直接使用できます。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。

さらに高度な用途のために、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。

SQL によるデータベースの作成

データベースの作成時に、データベース・サーバは次の 4 つの DB 領域を作成します。

dbspace 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.msg</i>
IQ_SYSTEM_TEMP	テンポラリ IQ ストア・ファイル	<i>dbname.iqtmp</i>
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。このファイルには、サーバが生成した名前に拡張子 *.tmp* が付きます。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仮データベースです。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。

- 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。
- 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。

```
dbisqlc -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)

- Sybase Central のデータベース作成ウィザードで、[リモート・ホストで IQ エージェントを使用してユーティリティ・サーバを起動する] を選択します。

ユーティリティ・データベースとそのセキュリティの詳細については、「[ユーティリティ・データベース・サーバのセキュリティ](#)」(345 ページ) を参照してください。

IQ データベースを初めて作成する場合は、『Sybase IQ の概要』の手順を参照してください。

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。

データベース・ファイルの配置と移動

データベースを作成する場合、そのロケーションを指定します。指定を行う前に、そのデータベースを移動する必要があるかどうか検討します。

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` コマンドを使用してフル・バックアップを行い、`RENAME` オプションを指定した `RESTORE` コマンドを使用してバックアップをリストアする必要があります。詳細については、「[第 13 章 システムのリカバリとデータベースの修復](#)」を参照してください。

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 `dbspace` 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。

警告! システム、IQ メイン、IQ テンポラリの DB 領域にファイル・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。こうすると I/O パフォーマンスが低下し、ローカル・エリア・ネットワークの負荷が大きくなり、`dbspace` ファイルに問題が起きるおそれがあります。UNIX 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。

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、Sybase IQ のマルチプレックス機能の使用を検討します。概要については、「[マルチプレックス機能](#)」(6 ページ) を参照してください。

データベース・ファイルの互換性	Sybase IQ サーバでは、Sybase IQ 12.6 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。
Sybase Central を使用した IQ データベースの作成	Sybase Central で IQ データベースを作成する方法については、『Sybase IQ の概要』の「 第4章 Managing Databases 」の「 Creating databases 」を参照してください。マルチプレックス・データベースを作成する方法については、『Sybase IQ マルチプレックスの使用』を参照してください。
CREATE DATABASE 文の使用	<p>CREATE DATABASE 文を使用して、IQ データベースを作成できます。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。</p> <ul style="list-style-type: none">• 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 NOT EQUAL 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ブランク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 Enterprise との互換性はありません。• 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。「相対パス名の使用」も参照してください。• ロー・デバイスの場合、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。

注意 新しいデータベースにおけるパスワードの大文字小文字の区別の詳細については、『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「[付録 A 他の Sybase データベースとの互換性](#)」の「[ユーザ ID とパスワード](#)」を参照してください。

全パラメータの詳細については、『リファレンス:文とオプション』の「[CREATE DATABASE 文](#)」を参照してください。次に IQ データベースの作成例をいくつか示します。

相対パス名の使用

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することをおすすめします。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。詳細については、『リファレンス:文とオプション』の「[CREATE DATABASE 文](#)」を参照してください。

ディレクトリ・パスを省略すると、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 の指定

必須の IQ PATH パラメータは、Anywhere データベースではなく IQ データベースを作成することを 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。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。

「[DB 領域の追加](#)」(188 ページ) で説明したように、別のドライブ上に領域を追加することはできませんが、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「第 6 章 物理的制限」を参照してください。

表 5-2: UNIX 上でのロー・デバイス名

UNIX プラットフォーム	例
AIX	/dev/rraw121v
HP-UX	/dev/vg03/rrchee12g
Sun Solaris	/dev/rsd0c
Sun AMD	/dev/rdisk/c5t0d0s1

表 5-3: Windows 上でのロー・デバイス名

デバイス・タイプ	要求される名前のフォーマット	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	Sybase Central では <code>%%C:</code> 、SQL では <code>%%C:</code>
パーティションなし	<code>PhysicalDriveN</code> : <code>N</code> 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。	Sybase Central では <code>%%PhysicalDrive32</code> 、SQL では <code>%%PhysicalDrive32</code>

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 文で指定するときは、円記号を必ず二重にする必要があります。詳細については、例 4 を参照してください。

例 1

次の文は `company.db` という名前の IQ 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 (サーバが起動したディレクトリ (この場合は `c:%%company`) 内の `company.db`)
- IQ ストア (`c:%%company%%iqdata%%company.iq`)
- テンポラリ ストア (`c:%%company%%company.iqtmp`)
- IQ メッセージ・ログ・ファイル (`c:%%company%%company.iqmsg`)

```
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 IQ 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 (サーバが起動したディレクトリ (この場合は `/disk1/company`) 内の `company.db`)
- IQ ストア (`/disk1/company/iqdata/company.iq`)

- テンポラリ・ストア (*/disk1/company/iqdata/company.iqtmp*)
- IQ メッセージ・ログ・ファイル (*/disk1/company/iqdata/company.iqmsg*)

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

例 3 次を示す 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 IQ データベースを作成しています。

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

例 4 次を示す 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 IQ データベースを作成しています。

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

IQ ページ・サイズの選択

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。このオプションで、メモリとディスクの使用量が決定されます。IQ PAGE SIZE には 65536 ~ 524288 バイトの 2 の累乗の値を指定します。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。

最高のパフォーマンスを得るためには、次の最小 IQ ページ・サイズの使用をおすすめします。

- 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。
- 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 億より多く 40 億より少ないデータベース、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、128KB (IQ PAGE SIZE 131072)。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。

Sybase IQ はデータを圧縮済みの形でディスクに格納します。データを処理するときは、データを圧縮解除して、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「第4章 システム・リソースの管理」を参照してください。

ワイド・データ用の IQ ページ・サイズの設定

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。

データベースのサイズの指定

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・ストアの IQ SIZE および IQ RESERVE パラメータと、テンポラリ・ストアの TEMPORARY SIZE および TEMPORARY RESERVE パラメータで定義されます。

- ロー・パーティションの場合は、IQ SIZE または TEMPORARY SIZE を指定する必要はありません。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、下記のデフォルトを使用するか、データのサイズに基づいて、下記の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 RESERVE および 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 5-4: 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	16KB	8MB
524288	32768000	16384000	32MB	16KB

カタログ・ページ・サイズを選択

CREATE DATABASE PAGE SIZE オプションを指定して、カタログ・ストアのページ・サイズを選択できます。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 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 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
```

データベースのブロック・サイズを選択

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。

データベースでの Java の有効化

デフォルトでは、IQ データベースで Java サポートが ON になっています。これは、JAVA OFF オプションで OFF にできます。Java が ON の場合、次のことが可能です。

- カatalog・ストアまたは IQ ストア内のテーブルにアクセスする Java プロシージャを記述できます。これらのクエリは他のクエリと同じように処理されます。
- Java データは IQ テーブルまたはカタログ・ストア・テーブルには格納できません。Java タイプの IQ カラムを作成しようとするエラーになります。
- Java アプリケーション・プログラミング・インタフェース (API: Application Programming Interface) をストアド・プロシージャで使用できます。

- Java データを IQ データベース内のテーブルに格納することはできないため、JDBC インタフェースを使用すると SQL データのみにアクセスできます。
- Sybase IQ では、ネットワーク・クライアント CD に収録されている複合 Java / ストアド・プロシージャ・デバッグの動作確認が行われています。

次の Java 機能は、カタログ・ストアのみでサポートしています。

- Java ベースのユーザ定義関数は IQ テーブルへのクエリ内では使用できませんが、カタログ・ストア・テーブル上では使用できます。
- IQ テーブルでは Java クラスをデータ型として使用できませんが、カタログ・ストア・テーブルでは使用できます。
- SQL Anywhere でサポートされている Java API 句は、IQ カタログ・ストアでもサポートされています。

Sybase IQ では、プロキシ・テーブルを利用することで、IQ から SQL Anywhere テーブルへのアクセスと、SQL Anywhere から IQ テーブルへのアクセスをサポートしています。リモート・データ・アクセス機能を使用して Anywhere から IQ テーブルに、または IQ から Anywhere テーブルにアクセスするときには、追加の Java 機能を利用できます。

- SQL Anywhere データベース内のテーブルに対するクエリでも、SQL Anywhere データベースから IQ テーブルへのクエリでも、Java ベースのユーザ定義関数を使用できます。リモート・データ・アクセス機能の使用方法の詳細については、『システム管理ガイド第2巻』の「[第4章 リモート・データへのアクセス](#)」を参照してください。
- SQL 文で Java 演算子を使用できます。
- SQL 文で Java API 句を使用できます。
- Java API クラスを、SQL で提供される使用可能な組み込み関数の拡張機能として扱うことができます。

Sybase IQ での Java サポートの詳細については、次を参照してください。

- Sybooks Online Help の SQL Anywhere Studio のマニュアルにある『SQL Anywhere サーバ - プログラミング』の「データベースにおける Java の概要」(<http://infocenter.sybase.com/help/index.jsp>)
- 『システム管理ガイド第2巻』の「[付録 B データベースでのロジックのデバッグ](#)」

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 dbfile の予約領域の割り付けサイズを決定します。領域の使用状況のモニタリングと領域不足の対処法については、「[リソースの問題](#)」(537 ページ) を参照してください。

DBA 権限を持つユーザは、接続ごとに使用される領域量を制限できます。また、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、現在の文が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。

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・オプションがデフォルト値の 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・オプションの詳細については、『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」を参照してください。

データベース・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。Sybase Central では、これらのオプションがすべて [データベース・オプション] ダイアログにまとめられています。Interactive SQL では、これらのオプションを SET OPTION 文で指定できます。

❖ データベース・オプションの設定 (Sybase Central)

- 1 サーバを開きます。
- 2 データベースを右クリックし、[オプション]を選択します。
- 3 値を編集します。

❖ データベース・オプションの設定 (SQL)

- SET OPTION 文の中でプロパティを指定します。

ヒント

[データベース・オプション]ダイアログでは、特定のユーザおよびグループに関するデータベース・オプションも設定できます。

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 グループのオプションを設定することになります。すべてのユーザとグループは PUBLIC のオプション設定を継承するからです。

詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」を参照してください。

データベース内のシステム・オブジェクトの表示

データベース内のテーブル、ビュー、ストアド・プロシージャ、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。

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。

❖ データベース内のシステム・オブジェクトの表示 (Sybase Central)

- 1 サーバを開きます。
- 2 接続されているデータベースを右クリックし、[所有者/コンテナのフィルタリングの設定]を選択します。
- 3 SYS と dbo を有効にし、[OK] をクリックします。

システム・テーブル、システム・ビュー、システム・プロシージャ、システム・ドメインが、それぞれのフォルダに表示されます (たとえば、システム・テーブルは [テーブル] フォルダ内に通常のテーブルと一緒に表示されます)。システム・ビューは、SYS ユーザ ID によって所有されています。

❖ システム・ビューのクエリ (SQL)

- 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。

例 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] ペインに表示します。

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

データベースからの切断

データベースに対する作業が終わったら、データベースから切断できます。Sybase IQ では、他のユーザを特定のデータベースから切断することもできます。これを Sybase Central で行う方法については、オンライン・ヘルプを参照してください。

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' )
```

❖ データベースから切断する (Sybase Central)

- 1 目的のサーバを開きます。
- 2 目的のデータベースを選択します。
- 3 ツールバーの [切断] ボタンをクリックします。

❖ データベースから切断する (SQL)

- DISCONNECT 文を実行します。

例 1 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
```

例 2 次の文は、Embedded SQL 内での DISCONNECT の使用法を示します。

```
EXEC SQL DISCONNECT :conn_name
```

❖ データベースから他のユーザを切断する (SQL)

- 1 DBA 権限を使用して既存のデータベースに接続します。
- 2 DROP CONNECTION 文を実行します。

例 3 次の文は、ID 番号 4 の接続を削除します。

```
DROP CONNECTION 4
```

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

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 `dbspace` 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なります。

```
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。

DB 領域の使用

DBA は、どのテーブル、インデックス、およびジョイン・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 領域の SQL 文とストアド・プロシージャを使用して、領域予約済みの DB 領域の作成、DB 領域のサイズ変更、DB 領域に関する情報の表示、オブジェクトの移動、空の DB 領域の削除を行う場合の例については、[「DB 領域管理の例」\(191 ページ\)](#) を参照してください。

dbfile の属性と操作

`dbfile` 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す。`dbfile` は、`dbfile` 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 `dbfile` 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、`dbfile` 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、`dbfile` 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。

`dbfile` 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。詳細については、『リファレンス:文とオプション』の[「ALTER DBSPACE 文」](#) 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 dbfile をオープンできない場合、データベースは起動しますが、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。

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 領域では、追加、削除、名前の変更などの操作を実行できます。『リファレンス：文とオプション』の「[第1章 SQL 文](#)」の「[ALTER DBSPACE 文](#)」、[CREATE DBSPACE 文](#)」、および「[DROP 文](#)」を参照してください。マルチプレックス DB 領域については、『[Sybase IQ マルチプレックスの使用](#)』の「[第3章 マルチプレックス・トランザクションの実行](#)」の「[マルチプレックス内の DB 領域の更新](#)」を参照してください。

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

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 dbfile 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 領域にある DB 領域または dbfile の名前変更

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、関連ファイルのカタログ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、ファイルのカタログ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。DB 領域でのオブジェクトの作成と移動に必要なパーミッションについては、「[DB 領域管理パーミッション](#)」(340 ページ)を参照してください。

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	状態	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		

テーブルまたはジョイン・インデックスは、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)。Sybase IQ では、読み取り専用テーブルや読み取り専用テーブル・パーティションで DML 操作が実行されるとエラーが返されます。

ジョイン・テーブルのいずれかが読み取り専用である場合、ジョイン・インデックスの作成、削除、同期など、ジョイン・インデックスに対する操作は失敗します。

表 5-5 は、変更可能な DB 領域の設定を示します。

表 5-5: 変更可能な DB 領域の設定

状態	変更の種類	ユーザ・メインでの変更	IQ_SYSTEM_MAIN または IQ_SYSTEM_TEMP での変更
オンラインの DB 領域			
	ALTER DBSPACE OFFLINE	RO の場合は可	不可
	ALTER DBSPACE ONLINE	不可	IQ_SYSTEM_MAIN の場合は可、 IQ_SYSTEM_TEMP の場合は可
	ALTER DBSPACE READONLY	RW の場合は可	不可
	ALTER DBSPACE READWRITE	RO の場合は可	不可
	ALTER STRIPING または STRIPESIZEKB	可	可
	RENAME DBSPACE	可	不可
	ADD FILE	可	可
	DROP FILE	空の場合は可	空の場合は可
	ALTER FILE READONLY	RW の場合は可	RW で、かつ最後の RW dbfile でない場合は可
	ALTER FILE READWRITE	RO の場合は可	RO の場合は可
	ALTER FILE SIZE	RW (DB 領域と dbfile) の場合は可	RW ファイルの場合は可
	ALTER FILE RENAME LOGICAL NAME	可	可
	ALTER FILE RENAME PATH	不可	IQ_SYSTEM_MAIN の場合は不可、 IQ_SYSTEM_TEMP の場合は可
オフラインの DB 領域			
	ALTER DBSPACE OFFLINE	不可	N/A
	ALTER DBSPACE ONLINE	可	N/A
	ALTER DBSPACE READONLY	不可	N/A
	ALTER DBSPACE READWRITE	不可	N/A

状態	変更の種類	ユーザ・メインでの変更	IQ_SYSTEM_MAIN または IQ_SYSTEM_TEMP での変更
	ALTER STRIPING または STRIPESIZEKB	可	N/A
	RENAME DBSPACE	可	N/A
	ADD FILE	不可	N/A
	DROP FILE	空の場合は可	N/A
	ALTER FILE RO	可	N/A
	ALTER FILE RW	可	N/A
	ALTER FILE SIZE	不可	N/A
	ALTER FILE RENAME LOGICAL NAME	可	N/A
	ALTER FILE RENAME PATH	可	N/A
動的にオフラインの DB 領域			
	ALTER DBSPACE OFFLINE	RO の場合は可	N/A
	ALTER DBSPACE ONLINE	不可	N/A
	ALTER DBSPACE READONLY	不可	N/A
	ALTER DBSPACE READWRITE	不可	N/A
	ALTER STRIPING または STRIPESIZEKB	可	N/A
	RENAME DBSPACE	可	N/A
	ADD FILE	不可	N/A
	DROP FILE	空の場合は可	N/A
	ALTER FILE READONLY	RW ファイルの場合は可	N/A
	ALTER FILE READWRITE	RO ファイルの場合は可	N/A
	ALTER FILE SIZE	不可	N/A
	ALTER FILE RENAME LOGICAL NAME	可	N/A
	ALTER FILE RENAME PATH	不可	N/A
読み取り専用 DB 領域			
	ALTER DBSPACE OFFLINE	オンラインの場合は可	N/A

状態	変更の種類	ユーザ・メインでの変更	IQ_SYSTEM_MAIN または IQ_SYSTEM_TEMP での変更
	ALTER DBSPACE ONLINE	オフラインの場合は可	N/A
	ALTER DBSPACE READONLY	不可	N/A
	ALTER DBSPACE READWRITE	オンラインの場合は可	N/A
	ALTER STRIPING および STRIPESIZEKB	可	N/A
	RENAME DBSPACE	可	N/A
	ADD FILE	可	N/A
	DROP FILE	空の場合は可	N/A
	ALTER FILE READONLY	RW の場合は可	N/A
	ALTER FILE READWRITE	RO の場合は可	N/A
	ALTER FILE SIZE	不可	N/A
	ALTER FILE RENAME LOGICAL NAME	可	N/A
	ALTER FILE RENAME PATH	オフラインの場合は可	N/A
読み込み/書き込み DB 領域			
	ALTER DBSPACE OFFLINE	不可	不可
	ALTER DBSPACE ONLINE	動的にオフラインの場合は可	可
	ALTER DBSPACE READONLY	RW の場合は可	不可
	ALTER DBSPACE READWRITE	RO の場合は可	不可
	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	空の場合は可
	ALTER FILE READONLY	RW ファイルの場合は可	RW であり、最後の dbfile でない場合は可
	ALTER FILE READWRITE	RO ファイルの場合は可	RO ファイルの場合は可
	ALTER FILE SIZE	RW ファイルの場合は可	RW ファイルの場合は可
	ALTER FILE RENAME LOGICAL NAME	可	可

状態	変更の種類	ユーザ・メインでの変更	IQ_SYSTEM_MAIN または IQ_SYSTEM_TEMP での変更
	ALTER FILE RENAME PATH	不可	IQ_SYSTEM_MAIN の場合は不可、 IQ_SYSTEM_TEMP の場合は不可
読み取り専用ファイル			
	ALTER FILE READONLY	不可	不可
	ALTER FILE READWRITE	可	可
	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	不可
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	不可

注意事項

- 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 IQ_SYSTEM_MAIN dbfile 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。

DB 領域の命名

DB 領域または dbfile 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、および 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。

DB 領域の追加

DB 領域は、CREATE DBSPACE 文または Sybase Central の DB 領域作成ウィザードを使用して作成します。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。DBA 権限または SPACE ADMIN 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「第 6 章 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。

Sybase では、ロー・デバイス上にメイン・ストアを作成することを推奨しています。

新しい DB 領域に対してロー・デバイスを指定すると、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、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 IQ 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 および 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「[第6章 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・システム上にストライピングしているオペレーティング・システムまたはハードウェアを使用する場合、ストライプ・サイズは最小 1MB、または可能な限り最大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 IQ を設定することもできます。

ディスク・ストライピングと複数の DB 領域の使用法の詳細については、『パフォーマンス&チューニング・ガイド』の「[I/O の分散](#)」を参照してください。

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。詳細については、「[必要な領域と DB 領域の見積もり](#)」(162 ページ)を参照してください。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、IQ_SYSTEM_MAIN DB 領域と同じディレクトリにある *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 領域を作成します。

```
CREATE DBSPACE library
USING FILE library
'library.iq' SIZE 100 MB IQ STORE
```

Sybase Central で DB 領域を作成するには、オンライン・ヘルプまたは『Sybase IQ の概要』の「[第6章 DB 領域の管理](#)」の「[DB 領域の作成](#)」を参照してください。

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

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。

- DBA 権限または SPACE ADMIN 権限を持っていること。
- ユーザ・テーブルまたはジョイン・インデックスのデータが含まれていないこと。Sybase IQ では、空でない DB 領域は削除できません。
- 必須の DB 領域 (**SYSTEM**、**IQ_SYSTEM_MAIN**、**IQ_SYSTEM_TEMP**、または **IQ_SYSTEM_MSG**) ではないこと。これらの DB 領域は削除できませんが、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は、IQ メイン・ストアまたはカタログ・ストアから他の DB 領域を削除できます。

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。

Sybase IQ では、DB 領域へのデータの格納方法の都合上、テーブルやジョイン・インデックスを明示的に移動しただけで DB 領域が空になることはまずありません (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。詳細については、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

DBA は、ストアド・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、ジョイン・インデックス、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。

`sp_iqdbspace` の出力フィールドの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「第7章 システム・プロシージャ」の「`sp_iqdbspace` 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  
  JAVA OFF  
  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 ずつ拡張できます。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` パラメータで設定)。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

使用法	TotalSize	予約	NumFiles	NumRWFiles
25	20M	90M	1	1
7	10M	10M	1	1
1	10N	20M	1	1
1	10M	40M	1	1

Stripingon	StripeSize	BlkTypes	OK ToDrop
T	1K	1H, 1248F, 32 D, 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 領域を拡張することはできません。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。

```
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	10N	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。

`ALTER DBSPACE` コマンドの完全な構文と動作については、『リファレンス：文とオプション』の「第 1 章 SQL 文」を参照してください。

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 t1clhg ON t1(c1);
CREATE hng INDEX t2clhng ON t2(c1);
INSERT t1 VALUES(1);
INSERT t2 VALUES(2);
COMMIT;
```

`dbspace` 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	Total Size	Reserve
IQ_SYSTEM_MAIN	MAIN	T	T	25	50M	100M
IQ_SYSTEM_TEMP	TEMPORARY	T	T	7	10M	10M
dbspacedb2	MAIN	T	T	1	10N	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、この領域はユーザ独自のデータベースには存在しない場合があります。

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	所有者	object_name	object_id	id
iq_main	table	DBA	empl	3,813	743
iq_main	table	DBA	iq_dummy	3,801	742
iq_main	table	DBA	sale	3,822	744
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	インデックス	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

columns	インデックス	metadata	primary_key
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
0B	0B	Y
0B	48K	Y
0B	0B	Y
0B	48K	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・ストアド・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_IDX_T733_C10_FP	iq_main	440K	1	1,046,689	1,047,147
GROUPO.Customers.ASIQ_IDX_T733_C1_FP	iq_main	440K	1	1,046,641	1,047,213
GROUPO.Customers.ASIQ_IDX_T733_C2_FP	iq_main	440K	1	1,046,961	1,047,203
GROUPO.Customers.ASIQ_IDX_T733_C3_FP	iq_main	440K	1	1,046,833	1,047,196

Object	DBSpaceName	ObjSize	DBSpPct	MinBlk	MaxBlk
GROUPO.Custom ers.ASIQ_IDX_ T733_C4_FP	iq_main	440K	1	1,046,737	1,047,189
GROUPO.Custom ers.ASIQ_IDX_ T733_C5_FP	iq_main	440K	1	1,046,929	1,047,182
GROUPO.Custom ers.ASIQ_IDX_ T733_C6_FP	iq_main	440K	1	1,047,009	1,047,175
GROUPO.Custom ers.ASIQ_IDX_ T733_C7_FP	iq_main	440K	1	1,046,945	1,047,168
GROUPO.Custom ers.ASIQ_IDX_ T733_C8_FP	iq_main	440K	1	1,046,785	1,047,161
GROUPO.Custom ers.ASIQ_IDX_ T733_C9_FP	iq_main	440K	1	1,046,881	1,047,154
GROUPO.Custom ers.ASIQ_IDX_ T733_I11_HG	iq_main	152K	1	1,047,121	1,047,206
GROUPO.Custom ers.IX_custom er_name	iq_main	304K	1	1,050,995	1,051,038

`sp_iqdbspace`、`sp_iqdbspaceinfo`、および `sp_iqindexinfo` システム・ストアド・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』の「[第7章 システム・プロシージャ](#)」を参照してください。

テーブルの編集

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。

この項では、データベースでのテーブルの作成、変更、削除の方法について説明します。例は `dbisql` 内で実行できますが、SQL 文は使用している管理ツールに依存しません。

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、標準化した方法でのテーブルの作成も可能となり、テーブルのコピーや修正ができます。

テーブルの作成

Sybase Central でのテーブル作成

Sybase Central を使用してテーブルを作成する方法については、『Sybase IQ の概要』の「[Managing tables](#)」を参照してください。

テーブルを作成する SQL 文

テーブルを作成する SQL 文は CREATE TABLE 文です。

この項では、CREATE TABLE 文の使い方を説明します。この項の例には、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。

dbisql からデモ・データベースに接続する方法については、「[Sybase Central または Interactive SQL からデモ・データベースへの接続](#)」を参照してください。

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。

SQL でのテーブルの作成

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（たとえば、technical や administrative など）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。プライマリ・キーの作成方法については、「[プライマリ・キーと外部キーの作成](#)」(204 ページ) を参照してください。

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。

CREATE TABLE 文の詳細については、『リファレンス：文とオプション』の「[CREATE TABLE 文](#)」を参照してください。CREATE TABLE 文を使用してテーブル定義に制約を組み込む方法については、「[第 9 章 データ整合性の確保](#)」を参照してください。

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がある場合があります。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』の「[第 3 章 SQL データ型](#)」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

Sybase IQ では、次の 4 つのテーブル・タイプを認識します。

- 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。

グローバル・テンポラリ・テーブルは、CREATE TABLE の GLOBAL TEMPORARY オプションを使用するか、Sybase Central のグローバル・テンポラリ・テーブル作成ウィザードを使用して作成します。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。

データベースには、ベース・テーブルの場合と同様に、グローバル・テンポラリ・テーブルの定義は1つしか含まれません。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。詳細については、[Sybase IQ マルチプレックスの使用](#)を参照してください。

テンポラリ・テーブルに作成するには、次の構文を使用します。

```
SELECT * INTO #TableTemp FROM lineitem
WHERE l_discount < 0.5
```

注意 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。

ローカル・テンポラリ・テーブルのバージョン管理の詳細については、「[テンポラリ・テーブルのバージョン管理](#)」を参照してください。

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** 文か Sybase Central 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

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。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 5-6: 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

ジョイン仮想テーブル

ジョイン仮想テーブルとは、通常のテーブルと同じように見える非正規化テーブルであり、名前、カラム、ロー、インデックスを持ちます。ジョイン仮想テーブルは、**CREATE JOIN INDEX** の結果として Sybase IQ によって内部処理用に作成され、**DROP JOIN INDEX** コマンドによって削除されます。ジョイン仮想テーブルを明示的に作成、修正、削除することはできません。それらを使用または修正しようとするとエラー・メッセージが表示されることがあります。ジョイン仮想テーブルはすべて無視してください。

マルチプレックス内で動作中のサーバは、ジョイン・インデックスを作成または削除できません。詳細については、[Sybase IQ Multiplex の使用](#) を参照してください。

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、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。

IQ インデックス作成の詳細については、[「第 6 章 Sybase IQ インデックスの使用」](#) を参照してください。

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 IQ データベース内に作成すると、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。

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)
- MINIMIZE_STORAGE オプション (SET OPTION またはプラグインの [データベース・オプション] ダイアログ)
- カラムのデータ型と幅
- IQ PAGE SIZE オプション (CREATE DATABASE またはプラグインのデータベース作成ウィザード)

次の表に、IQ UNIQUE オプションの意味を示します。

表 5-7: IQ UNIQUE の影響

IQ UNIQUE が 256 以下の場合	IQ UNIQUE が 65536 以下の場合	IQ UNIQUE が 16777216 以下の場合	IQ UNIQUE が未指定または 16777216 を超える場合
記憶領域が少数のユニークな値用に最適化される	記憶領域が適度な数のユニークな値用に最適化される	記憶領域が 3 バイトの FP インデックス用に最適化される	記憶領域が多数のユニークな値用に最適化される
クエリ・パフォーマンスは高速になり、メイン IQ ストアの必要領域量は少なくなる	クエリ・パフォーマンスは高速になり、メイン IQ ストアの必要領域量は少なくなる	クエリ・パフォーマンスは高速になり、メイン IQ ストアの必要領域量は少なくなる	クエリが遅くなる場合がある
IQ テンポラリ・ストアにロード用の小さな追加キャッシュが必要	IQ テンポラリ・ストアにロード用の追加キャッシュが必要 (必要な量は、ユニークな値の数とデータ型によって異なる)	IQ テンポラリ・ストアにロード用の大きな追加キャッシュが必要 (必要な量は、ユニークな値の数とデータ型によって異なる)	ロード用の追加キャッシュは不要
IQ UNIQUE <256 のカラムが多数あるときは、ロードが遅くなる場合がある	ロードが遅くなる場合がある	ロードが遅くなる場合がある	ロードが高速になる

MINIMIZE_STORAGE オプションの影響

MINIMIZE_STORAGE を ON にすると、すべての新規カラムに IQ UNIQUE 255 を指定したのと同じことになります。MINIMIZE_STORAGE のデフォルトは OFF です。詳細については、『リファレンス：文とオプション』の「MINIMIZE_STORAGE オプション」を参照してください。

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ときは、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。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「[第 7 章 システム・プロシージャ](#)」を参照してください。

UNIQUE と IQ UNIQUE
の相違点

IQ UNIQUE (*count*) は、特定のカラム内に入れることのできる別個の値の概数を指定します。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` カラムの各値、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 制約も持つ必要があります。

テーブルの変更

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 次のコマンドは、スキルの説明 (オプション) を書き込むためのカラムを `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ド・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。

詳細については、『リファレンス：文とオプション』の「ALTER TABLE 文」を参照してください。ALTER TABLE を使用してテーブル定義に制約を組み込む方法については、「第9章 データ整合性の確保」を参照してください。

Sybase Central でのテーブル変更

Sybase Central でテーブル定義を変更するには、Sybase Central のオンライン・ヘルプを参照してください。

ジョイン・インデックスでのテーブル変更

ジョイン・インデックスのジョイン条件に使用されているベース・テーブルのカラムに対しては、ADD、DROP、MODIFY を実行できません。ジョインしたカラムを変更するには、ジョイン・インデックスを削除し、そのテーブルを変更してからジョイン・インデックスを再作成する必要があります。ジョイン・インデックスの詳細については、「ジョイン・インデックスの使用」を参照してください。

テーブルの削除

次に示す DROP TABLE 文は、skill テーブルから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 文と同様に、DROP 文はテーブルの削除前後に COMMIT 文を自動的に実行します。これで、最後に COMMIT または ROLLBACK を実行した後にデータベースに加えた変更がすべて確定されます。

DROP 文は、テーブル内のカラムのいずれかがジョイン・インデックスに使用されている場合を除き、テーブルのすべてのインデックスを一緒に削除します。

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んでいる最中のテーブルをトランケートすると、通常のテーブルのバージョン管理規則が適用され、そのユーザのトランザクションが完了するまで古いバージョンのテーブルが残されます。詳細については、「第10章 トランザクションとバージョン管理」を参照してください。

DROP TABLE 文と 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。

詳細については、『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。

❖ **Sybase Central でのテーブルの削除**

- 1 データベースに接続します。
- 2 データベースの [テーブル] フォルダをクリックします。
- 3 削除するテーブルを右クリックして、ポップアップ・メニューから [削除] を選択します。

プライマリ・キーと外部キーの作成

CREATE TABLE 文と ALTER 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。この項では、プライマリ・キーと外部キーを例に、テーブル属性の設定方法について説明します。

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。つまり今回の例では、特定の skill_id を持つローが複数存在してはなりません。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。

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、従業員のスキルごとのスキル・レベルを含む emp_skill テーブルを、次のように作成します。

```
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。これらの外部キー・エントリは2つのことを表しています。第一に、`emp_id` カラムには、`employee` テーブルに含まれている有効な従業員番号 (`employee` テーブルのプライマリ・キー) を格納する必要があります。第二に、`skill_id` カラムには、`skill` テーブルに含まれている有効なスキル ID (`skill` テーブルのプライマリ・キー) を格納する必要があります。

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ー・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「[第2章 SQL 言語の要素](#)」を参照してください。

プライマリ・キーと外部キーの使い方の詳細については、「[第9章 データ整合性の確保](#)」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。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「[第8章 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.SYSTAB` ビューのすべてのカラムが表示されます。

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

Sybase Central でシステム・ビューを表示するには、「[データベース内のシステム・オブジェクトの表示 \(Sybase Central\)](#)」(179 ページ) の手順に従います。

ビューの編集

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** クエリが実行可能
- 複数のビューに対して **INSERT** 操作と **DELETE** 操作が実行可能
- 他のビューに基づいたビューを作成可能

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は作成不可
- 特定のビュー上では、**INSERT**、**DELETE**、**UPDATE** 操作のみ実行可能
- ビューには整合性の制約とキーは設定不可
- ビューはベース・テーブル内の情報を参照するだけで、その情報を自分で持つてはいません。ビューは呼び出されるたびに再計算されます。

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。

- **セキュリティの向上** 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する
- **利便性の向上** ユーザとアプリケーション開発者に、ベース・テーブルよりも見やすい形でデータを提供する
- **一貫性の向上** 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する

ビューの作成

SELECT 文は 1 つ以上のテーブルを処理し、結果セットを生成します。結果セットはテーブルでもあります。ベース・テーブルと同様に、**SELECT** クエリの結果セットにはカラムとローがあります。ビューが作成するクエリには名前が与えられ、データベースのシステム・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す。

```
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正しい情報を集めるのに時間がかかります。

Sybase Central でビューを作成する方法については、『Sybase IQ の概要』の「[第8章 ビューの使用](#)」の「[ビューの定義](#)」を参照してください。

ビューの使用

ビューを使用するときは、ビュー作成に使用する **SELECT** 文にも、ビューに対する挿入、削除、更新の機能にも、一定の制限があることに注意してください。

SELECT 文に対する制限

ビューを作成する **SELECT** 文にはいくつかの制限があります。特に、**SELECT** クエリでは **ORDER BY** 句は使用できません。カラムとローの並び順が重要でないことがリレーショナル・テーブルの特性であり、**ORDER BY** 句を使用すると、ビュー内のローの順序が決まってしまうます。ビューの定義内では、**GROUP BY** 句、サブクエリ、ジョインを使用できます。

ビューを作成するには、まず、必要な結果が必要なフォーマットで入手できるまで **SELECT** クエリを調整します。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。例を示します。

```
CREATE VIEW viewname AS
```

ビューでの挿入と削除

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。

次のビューに対しては更新、挿入、削除を実行できません。

- **COUNT(*)** などの集約関数を含むビュー
- **SELECT** 文内に **GROUP BY** 句を含むビュー
- **UNION** 操作を含むビュー

いずれの場合でも、ビューに対する **UPDATE**、**INSERT**、**DELETE** 操作を基本となるテーブルへのアクションに変換することはできません。

警告！ **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。

ビューの修正

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わけではなく、ビューの定義を新しい定義に置き換えます。

ALTER VIEW 文は、ビューのパーミッションを維持します。

例 前述の **DepartmentSize** ビューのカラム名をよりわかりやすい名前に変更する場合は、次の文を使用します。

```
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す。

- 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。
- ユーザがすべてのベース・テーブルに対するその操作の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
 - 非 DBA ユーザだが、ビューが参照するすべてのベース・テーブルを所有しているユーザ
 - 非 DBA ユーザで、ビューが参照する一部または全部のベース・テーブルを所有していないが、自分が所有していないベース・テーブルに対する **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** を持っているユーザ。

所有者がベース・テーブルに対するパーミッションの **WITH GRANT OPTION** を持つ代わりに、パーミッションが **PUBLIC** に付与されている場合もあります。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。

ビューの削除

Interactive SQL で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
```

Sybase Central での
ビューの削除

Sybase Central でビューを削除するには、対象のビューを右クリックし、ポップアップ・メニューから [Delete] を選択します。

詳細については、Sybase Central のオンライン・ヘルプを参照してください。

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。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「第8章 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
```

インデックスの編集

データベースの設計と作成を行う上で、パフォーマンスは重要な考慮事項です。Sybase IQ インデックスを使用すると、従来のリレーショナル・データベースの検索に比べて、検索のパフォーマンスが画期的に向上します。ただし、Sybase IQ においても、最高のパフォーマンスを達成し、メモリ、ディスク、CPU サイクルを最も効率よく使用するためには、データに合った正しいインデックスの選択が重要です。

インデックスの概要

データのあるすべての IQ データベース・カラムにはインデックスが必要です。IQ ストア内にデータベースを作成する場合、デフォルト・インデックスが各テーブルの各カラムに自動的に作成されます。次のいくつかのインデックス・タイプからも選択できます。

- インデックス化されたカラムに対する特定のタイプのクエリを最適化する、4つのカラム・インデックス・タイプ
- 2つ以上のテーブルのカラムを関連付けるクエリを最適化する、ジョイン・インデックス

たいいていの場合、デフォルトのインデックスだけではなく、データベース内のカラムの大半に対して1つ以上のインデックスを選択します。ユーザ・クエリ内で一定の方法でジョインされるテーブル・カラムに対して、ジョイン・インデックスの定義を行うこともできます。データベースのサイズ、利用可能なディスク領域、ユーザが送信するクエリに基づいてインデックスを選択します。

インデックスは、指定したテーブル、またはジョイン・インデックス用のテーブルのセットに対して作成されます。ビューにはインデックスを作成できません。

インデックスの作成

カラム・インデックスの作成には、次の3つの方法があります。

- CREATE INDEX コマンド
- Sybase Central の [インデックス作成] ウィザード
- CREATE TABLE の UNIQUE または PRIMARY KEY カラム制約 (これによりユニーク・インデックスが自動的に作成される)

ジョイン・インデックスの作成には、次の2つの方法があります。

- CREATE JOIN INDEX 文
- Sybase Central のジョイン・インデックス作成ウィザード

インデックスの選択や作成の詳細については、「[第 6 章 Sybase IQ インデックスの使用](#)」を参照してください。コマンドの構文については、『リファレンス：文とオプション』の「[CREATE INDEX 文](#)」と「[CREATE JOIN INDEX 文](#)」を参照してください。Sybase Central の使用方法については、『[Sybase IQ の概要](#)」の「[第 5 章 インデックス機能とデータのロード](#)」を参照してください。

システム・ビューに含まれるインデックス情報

インデックスの情報はシステム・ビュー SYSINDEX、SYSIQINDEX、SYSIXCOL に、ジョイン・インデックスの情報は SYSIQJOINIXTABLE 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「第8章 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。

ストアド・プロシージャを使用したインデックスの表示

ストアド・プロシージャ `sp_iqindex` を使用して、インデックスのリストやインデックスの情報を表示できます。たとえば、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'
```

次の情報が表示されます。出力は、2つの部分に分けて読みやすくしてあります。

table_name	table_owner	column_name	index_type
Departments	GROUPO	DepartmentHeadID	FP
Departments	GROUPO	DepartmentHeadID	HG
Departments	GROUPO	DepartmentID	FP
Departments	GROUPO	DepartmentID	HG
Departments	GROUPO	DepartmentName	FP

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)

このコマンドでテーブル名を省略すると、`sp_iqindex` はデータベース内の全テーブルに関する情報を表示します。

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。外部キー・インデックスの場合は、正当性の検証により、対応するローがプライマリ・テーブルに存在することと、それらのハッシュ値が一致することも確認できます。この検証は、SQL Anywhere の VALIDATE TABLE 文による正当性の検証と同じです。

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 (コマンドは 1 行に入力します)。

```
dbvalid -c "uid=dba;pwd=sql;eng=myserver"  
-i EmployeeIndex
```

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。

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変更と外部キーの役割名の変更の詳細については、『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。キー制約を適用するために作成されたインデックスの名前は変更できません。

インデックスの削除

カラム・インデックスまたはジョイン・インデックスが不要になったときは、DROP 文を使用してデータベースから削除できます。Sybase Central でインデックスを削除するには、テーブル名をクリックし、右クリックでオプションを表示して [削除] オプションをクリックします。ジョイン・インデックスを削除する前に、「[ジョイン・インデックスに含まれるテーブルの修正](#)」に書かれている特別な要件を参照してください。

この章について

この章では、Sybase IQ のインデックス・タイプ、インデックスの作成方法、データベース内のデータにもっとも適合するインデックス・タイプを判定する方法、そしてインデックスに関連するパフォーマンスとリソースの問題について説明します。

内容

トピック名	ページ
インデックスの概要	213
Sybase IQ インデックスの作成	218
インデックス・タイプの選択	219
Sybase IQ インデックス・タイプ	225
データ挿入後のカラム・インデックスの追加	244
ジョイン・インデックスの使用	245
ジョイン・インデックスのサイズとメリットの見積もり	262

インデックスの概要

インデックスは、データ検索のパフォーマンスを改善するために使用します。従来のインデックスでは、ほとんどの場合データ・レコードを示すのに B-tree という方法を使用します。B-tree が役に立つのは、一意のデータ値を数多く使用してフィルタリングし、非常に少数のレコードのセットまで絞り込む場合に限られます。たとえば、トランザクション処理システムの受注番号や顧客名のカラムなどで使われます。

Sybase IQ のインデックスは実際に、データをさまざまなクエリの高速化に使用できるように表現し、格納します。これは、データ・ウェアハウス環境、つまり一意の値が比較的少ない膨大なレコード群をクエリで検索し、集約関数の結果を求めるような環境に合わせて設計された方式です。

インデックスの負荷のモニタリングと解析の詳細については、『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと削除の最適化](#)」の「[負荷のモニタリング](#)」を参照してください。

Sybase IQ インデックス・タイプ

テーブルにデータをロードすると、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保管します。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。論理的に、データはこれまでどおりローごとにアクセスでき、従来のローベースの SQL データベースと同様です。またカラムで保管することにより、追跡しようと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、処理の対象となるのは数千から数百万ローに及ぶデータの特定の属性であって、従来のトランザクション処理で扱うような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、必要な属性の選択や計算を実行する機能が最適化されます。

Sybase IQ がカラムごとに作成するデフォルトのカラム保管構造は、実際に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。データベースのサイズ、利用可能な空きディスク容量およびユーザが送信するクエリのタイプによっては、このデフォルトのインデックスに Sybase IQ のビット処理インデックス・タイプを 1 つまたはそれ以上割り当てたいと考えることはかなり多いと思われます。カラム・インデックス・タイプは、いくつかの選択肢の中から選択できます。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。ジョイン・インデックスは注意して作成してください。ジョイン・インデックスにより大幅な負荷と更新が追加され、コストが削減されます。

カラム・インデックスの他に、Sybase IQ ではジョイン・インデックスも定義できます。ジョイン・インデックスは、関連テーブルのジョインに最適化されています。管理者は、ユーザがクエリを解析するためによくジョインする任意のテーブル・セットに対してジョイン・インデックスを作成できます。カラム・インデックスは、そのようなカラムを含むすべてのジョイン・インデックスの基礎となります。

この章の前半では、カラム・インデックスについて解説します。この章の後半では、ジョイン・インデックスについて解説します。詳細については、「[ジョイン・インデックスの使用](#)」(245 ページ)を参照してください。

Sybase IQ では、射影を最適化する「デフォルト・インデックス」がすべてのカラムに対して作成されます。

ユニークな値の数が 1,600 万個未満のカラムは、最適化されたデフォルト・インデックスに格納されるため、必要記憶領域が大幅に削減されます。この形式は IQ オプティマイザによるパフォーマンス向上をサポートし、SUM、SUM DISTINCT、MAX、MIN、および COUNT DISTINCT などの集約関数もサポートしています。この形式は次のカラムで使用できます。

- IQ UNIQUE() が指定されているすべてのカラム
- MINIMIZE_STORAGE データベース・オプションが ON のときに作成されたすべてのカラム

ただし、クエリのパフォーマンスを最大限に引き出すには、ほとんどのカラムについて、カラムのデータのカーディナリティと利用率をもっともよく表す表 6-1 に示されているインデックス・タイプを、以下のうちから追加で 1 つ以上選択しなくてはなりません。

表 6-1: Sybase IQ インデックス・タイプ

インデックス・タイプ	説明
比較または CMP	同じデータ型、精度、位取りの 2 つの異なるカラムのバイナリ比較 (<, >, または =) を格納します。
DATE	DATE データ型のカラムのインデックス。日付量を扱うクエリを処理するときに使用されます。
Datetime または DTTM	DATETIME または TIMESTAMP データ型のカラムのインデックス。日時量を扱うクエリを処理するときに使用されます。
High_Group または HG	高カーディナリティ・データに対する操作によって等価およびグループを処理するための拡張 B-tree インデックス (独自の値が 1,000 個以上の場合に推奨)。
High_Non_Group または HNG	範囲や集約関数を含む高カーディナリティの意思決定支援オペレーションに最適な、値ベースでないビットマップ・インデックス。
Low_Fast または LF	低カーディナリティ・データ上の処理クエリに関する値ベースのビットマップ・インデックス (一意の値が 1,000 までの場合に推奨、ただし一意の値が 10,000 まで対応可能)。
TIME	TIME データ型のカラムのインデックス。時間量を扱うクエリを処理するときに使用されます。
WD	CHAR、VARCHAR、または LONG VARCHAR カラムの内容を区切られたリストとして扱うことで、キーワードにインデックスを付けます。
TEXT	単語 (ワード) とその位置のインデックスを作成するときに使用されます。指定の距離、特定の順序、およびこれらの条件の組み合わせで、個々の単語、フレーズ、単語のペアを検索できます。

カラム内のデータの型や、そのカラム・データで行おうとしている操作に応じて、適切なカラム・インデックスを選択してください。一般的には、すべてのカラムに、任意のインデックスまたはインデックスの組み合わせを使用できません。ただし、いくつか例外があります。

DATE データ型を使用してテーブルを作成すると、DATE フィールドに最適化された 2 バイトの FP インデックスが作成されます。このインデックスは、データベース・オプション `MINIMIZE_STORAGE` の設定に依存しません。

DATE フィールドに 3 バイトの FP インデックスまたはフラットスタイルの FP インデックスを作成する場合は、テーブルの作成時に次の `IQ UNIQUE` 値を使用します。

- 3 バイトの FP の場合 - `IQ UNIQUE` を 65537 ~ 16777216 の値にする必要があります。
- フラット・スタイルの FP の場合 - `IQ UNIQUE` を 16777216 より大きい値にする必要があります。

非整数の数値データを含むカラムで High_Non_Group インデックス・タイプを利用するには、254 桁までをサポートする NUMERIC または DECIMAL データ型を使用します。

インデックス型の中には、不必要なインデックスを作成してディスク領域を消費する重複した機能を持っているものがあります。インデックスの選択方法の詳細については、以下の項で解説します。

カラムを FOREIGN KEY、PRIMARY KEY、または UNIQUE として指定したときは、Sybase IQ によって自動的に High_Group インデックスが作成されます。それぞれの外部キーには、非ユニークな High_Group インデックスが作成されます。

注意 カラム・セットに対して明示的に High_Group インデックスを作成することもできます。詳細については、『リファレンス:文とオプション』の「[CREATE INDEX 文](#)」を参照してください。

Sybase IQ でのインデックスの使われ方

最高のパフォーマンスを得るために、カラムに追加のインデックスを定義することもできます。Sybase IQ では、現在のクエリやジョイン述部に利用可能な最も高速なインデックスが使用されます。カラムに対して正しいタイプのインデックスを作成しなかった場合でも、Sybase IQ はそのカラムを含むクエリを解析できます。ただし、インデックス・タイプが正しい場合よりもレスポンスが遅くなります。

特定のカラムに複数のインデックスが定義されている場合、Sybase IQ はそのカラムのすべてのインデックスを同じ入力データから構築します。

オプティマイザからのアドバイス

データベースで INDEX_ADVISOR オプションを設定している場合、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。このメッセージは次の領域に注目します。

- 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
- 相関サブクエリ・カラム
- グループ化カラム

詳細については、『リファレンス:文とオプション』の「[第2章 データベース・オプション](#)」の「[INDEX_ADVISOR オプション](#)」を参照してください。

推奨に従う場合には、インデックスを自分で作成します。

インデックスの追加と削除

後でインデックスを追加する必要が生じてインデックスは簡単に追加できますが、データを挿入する前に必要なインデックスをすべて作成する方が時間を短縮できます。

所有者タイプが 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にかぎり、変更または名前の変更ができます。インデックスの名前変更と外部キーの役割名の変更の詳細については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「ALTER INDEX 文」を参照してください。

必要ないと判断したオプションのインデックスは、削除が可能です。インデックスの削除の詳細については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「DROP 文」に記載されている DROP INDEX コマンドに関する記述を参照してください。

注意 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 CONSTRAINT では、自動生成された非ユニークな HG インデックスは削除されません。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 コマンドの発行後に明示的にインデックスを削除します。

従来のインデックスに勝る利点

Sybase IQ インデックスには、従来のインデックス技術に勝る、以下のような利点があります。

- 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析されます。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、したがって反復する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れば済みます。

Sybase IQ インデックスの作成

CREATE INDEX 文または Sybase Central を使用すると、明示的にカラム・インデックスを作成できます。これら 2 つの方法は、以下の項で解説します。

CREATE INDEX 文

Sybase IQ のカラム・インデックスを作成するには、次の構文を使用します。

```
CREATE [ UNIQUE ] [ index-type ] INDEX 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 を指定しなかった場合は、HG インデックスが作成されます。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。

例 最初の例では、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)
```

この 2 つ目の例では、SalesOrder テーブルの Region カラムに、SalesOrderRegionIX という Low_Fast (LF) インデックスを作成します。

```
CREATE LF INDEX SalesOrderRegionIx
ON dbo.SalesOrder (Region)
```

CMP インデックスの作成方法の例については、「[比較 \(CMP\) インデックス・タイプ](#)」(234 ページ) を参照してください。

デフォルトでは、100,000 レコードが挿入されてインデックスにロードされるごとに、処理状況のメッセージが表示されます。レコード数を変更するには、CREATE INDEX または NOTIFY_MODULUS の NOTIFY オプションを指定します。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。

複数の CREATE INDEX 文をグループとして同時に実行したいときは、キーワード BEGIN PARALLEL IQ と END PARALLEL IQ で区切ります。これらのキーワードが使用できるのは IQ テーブル上でインデックスを作成するときのみで、テンポラリ・テーブルや SQL Anywhere テーブル上で作成するときには使用できません。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す。詳細については、リファレンス：文とオプションを参照してください。

Sybase Central を使用したインデックスの作成

Sybase Central を使用してカラム・インデックスを作成するには、『Sybase IQ の概要』の「第 5 章 インデックス機能とデータのロード」の「カラム・インデックスの作成」を参照してください。

TEXT インデックスの作成

詳細については、『Sybase IQ の非構造化データ分析の概要』の「第 2 章 TEXT インデックスとテキスト設定オブジェクト」の「TEXT インデックスの作成」を参照してください。

複数のインデックスの同時作成

場合により、同時に 2 つ以上のカラム・インデックスを作成できます。

- CREATE INDEX 文では、それぞれ 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 つの接続が CREATE INDEX 文を発行した場合は、先に発行した文が処理されます。後に発行した文に対しては、書き込みは 1 人しか許可されないというエラー・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。
- 2 つの接続がそれぞれ異なるテーブルに対して CREATE INDEX 文を発行したが、両方のテーブルが同じジョイン・インデックスに入っている場合は、一方の CREATE INDEX のみが処理され、もう一方に対しては書き込みは 1 人しか許可されないというエラー・メッセージが返されます。

インデックス・タイプの選択

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。インデックスを選択するときには、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、高速射影が簡単に行え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。ユニークな値と合計値の数、ユーザが期待するクエリ結果、およびそのデータがアドホック・ジョインとジョイン・インデックスのどちらで使用されるかを予測してください。

インデックス・タイプの詳細と、正しいタイプを選択するための条件については、以下の項で解説します。

インデックス内のユニークな値の数

Sybase IQ 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ています。この数が特定のレベルに達したら、表 6-2 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 6-2: 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 25,000 未満の場合は HG)
1,000 以上	HG か HNG または両方

MINIMIZE_STORAGE オプションを ON にして作成したカラム、または IQ UNIQUE 65536 以下を指定したカラムは、記憶領域の削減と、ある種のクエリのパフォーマンス向上のために最適化されたデフォルト・インデックスの形式で自動的に配置されます。

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 個以上 65536 個未満となります。
- 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

カラム内のデータが一般的にどのように問い合わせられるかを、あらかじめ理解しておく必要があります。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、またはこれらのうち複数当てはまるのか？
- 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 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析できますが、効率は下がります。

次の表は、推奨されるインデックス・タイプをクエリのタイプ別に示します。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。また、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。この項のもう1つの表を参照してください。

表 6-3: クエリのタイプ/インデックス

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	HNG、LF、HG、デフォルト
MIN/MAX 引数として	LF、HG、HNG
COUNT 引数として	デフォルト
COUNT DISTINCT、SELECT DISTINCT または GROUP BY 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、ALF、
ジョイン・インデックスで使用されるカラム	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、CMP
WHERE 句の範囲述部 (>、<、>=、<=、BETWEEN DATEPART 等号述部、範囲述部、および IN リスト述部内	LF、HG、または HNG。CMP、DATE、TIME、DTTM
CONTAINS 述部	WD、TEXT

HNG が推奨されていますが、場合によっては LF または HG の方が速く、HNG の代わりに使用されることもたびたびあります。HNG はたいていの場合一貫したパフォーマンスを示しますが、範囲を含む LF や HG のパフォーマンスは、選択した範囲のサイズによって異なります。

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。

これらの推定値は一般的には有効ですが、以下のように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、ユニークな値の数の方が重要な要素となります。
- COUNT、COUNT DISTINCT、SUM、MIN、MAX、AVG などの set 関数で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またはジョイン・インデックスを使用して解決できなくてはなりません。
- BIT データは、デフォルト・インデックスでのみ使用できます。255 バイトよりも大きい VARBINARY データは、デフォルト、TEXT、CMP の各インデックス・タイプでのみ使用できます。255 バイトよりも大きい CHAR および VARCHAR データは、デフォルト、CMP、TEXT、WD の各インデックス・タイプでのみ使用できます。LONG VARCHAR データは、デフォルト、TEXT、WD の各インデックス・タイプでのみ使用できます。DATE データは、DATE インデックス・タイプでのみ使用できます。TIME データは、TIME インデックス・タイプでのみ使用できます。DATETIME および TIMESTAMP データは、DTTM インデックス・タイプでのみ使用できます。

インデックス条件：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 6-4: 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・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 インデックスでは、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、インデックスが使用する領域は増加します。

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

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。各データ型でサポートされるその他のインデックスについては、次の表を参照してください。

表 6-5: 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

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
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
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
binary	CMP、HG、LF、TEXT	HNG、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ときには、適切なカラム・インデックス・タイプが複数ある場合があります。

表 6-6 は、インデックス・タイプの有効な組み合わせを示します。

表 6-6: インデックスの組み合わせ

既存のインデックス	追加するインデックス					
	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 = 妥当な組み合わせ

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ます。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 つは持っています。

Sybase IQ インデックス・タイプ

この項では、カラム・インデックス・タイプを使用する理由をそれぞれ詳しく解説します。

高速射影 (FP) デフォルト・インデックス・タイプ

トークン化は、64K 以上の個別値があるカラムに適用できます。

Sybase IQ データベースに永久テーブルを作成すると、IQ によってすべてのカラム値がデフォルト・インデックスに保存されます。FP (高速射影) インデックスと呼ばれるこのデフォルト・インデックスによって、射影が最適化され、特定の種類の検索条件を評価できるようになります。

カラムごとに 1 つの FP インデックスがあり、それぞれの FP は n 個の固定長エントリの配列になっています。ここで n は、テーブル内のローの数です。各カラム値は RecordID の昇順で保存されます。

状態、日付、月のフィールドなど、少数の個別値やユニークな値を使用すると、必要なディスク・ページの数を低減する最適化された形式の FP を作成できるため、射影のためにカラムに必要な記憶領域と I/O コストが大幅に削減されます。

最適化されたこれらの FP インデックスは、2 つの部分で構成されます。カラムの各個別値が正確に 1 度だけ出現する検索テーブルと、論理配列の各要素が検索テーブル内でセル値が保存されるロケーションを指すキーになっている値の論理配列です。

`sp_iqindexmetadata` ストアド・プロシージャは、指定の所有者またはテーブルに属する指定のインデックスを記述したレポートを生成します。

`sp_iqindexmetadata` の出力には、特定のインデックスが 1 バイト、2 バイト、3 バイト、およびフラット・スタイルの FP インデックスの中のどれに該当するかが示されます。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、およびプロシージャ』の「[sp_iqindexmetadata プロシージャ](#)」を参照してください。

FP(1) インデックス

カラム値の数が 256 未満の場合は、1 バイト FP インデックスが作成されます。各ローの実際のキー値は検索テーブルに保存され、FP エントリ (検索キー) には検索テーブルへのインデックスが含まれます。1 バイト FP インデックスでは、FP エントリごとに 1 バイトが占有されます。

メイン・キャッシュがデフォルトの 32MB に設定され、`FP_LOOKUP_SIZE_PPM` がデフォルトの 2500 に設定されている場合、`LOAD`、`INSERT`、または `UPDATE` を使用して、`char(2)`、`binary(2)`、または小さい整数の FP(1) カラムに新しい個別値を追加すると、FP(1) からフラット FP に変換されます。

FP(2) インデックス

カラム値の数が 256 より多く 65536 以下の場合、同じ検索テーブルが拡張されます。各 FP エントリ (検索キー) には検索テーブルへのインデックスが含まれますが、1 バイトではなく 2 バイトが占有されます。

1 バイトのエントリが 2 バイトのエントリに変換される際のコストを抑えるには、テーブルの作成時に **IQ UNIQUE** 値を 256 よりも大きく 65536 以下になるように指定できます。

FP(3) インデックス

カラム値の個別カウントが 65536 を超えると、3 バイトの FP インデックスを作成できます。FP(3) インデックスは、構造上 FP(1) および FP(2) インデックスと類似していますが、次の点が異なります。

- FP(3) 検索テーブルの最大サイズは、65536 ではなく 16777216 です。
- FP(3) インデックス・バッファ・ストレージには、それぞれ 3 バイトの検索キーが含まれます。

3 バイトのインデックスでは、個別カウントが 16777216 を超えない場合はカラムに値が格納されます (カラム・データ)。カラム・データのサイズが 3 バイトを超える場合のみ、カラムに 3 バイトのインデックスを作成できます。したがって、データ型 **BIT**、**TINYINT**、**SMALLINT**、**CHAR(<=3)**、**VARCHAR(<=3)**、**BINARY(<=3)**、および **VARBINARY(<=3)** のカラムで FP(3) インデックスを作成することはできません。また Sybase IQ では、**LONG VARCHAR** および **LONG VARBINARY** データ型の FP(3) インデックスもサポートしていません。

FP(3) インデックスを作成するには、**MINIMIZE_STORAGE** を ON に設定するか、65537 ~ 16777216 の **IQ UNIQUE** 制約値を使用してカラムを作成しておく必要があります。ユニーク・カウントが 65536 を超えたら、FP(2) インデックスから FP(3) を作成することもできます。ただし、表 6-7 に示すデータ・サイズにのみ対応し、**BIT**、**TINYINT**、**SMALLINT**、**CHAR(<=3)**、**VARCHAR(<=3)**、**BINARY(<=3)**、および **VARBINARY(<=3)** の各データ型は除外されます。詳細については、『Sybase IQ システム管理ガイド』の「第 9 章 データ整合性の確保」の「カラムに対する **IQ UNIQUE** 制約の使用」、および『Sybase IQ リファレンス・マニュアル』の「第 2 章 データベース・オプション」の「**MINIMIZE_STORAGE** オプション」を参照してください。

動作変更

FP インデックスの動作には、15.0 より前のバージョンの Sybase IQ とは異なる点がいくつかあります。

表 6-7: Sybase IQ 高速射影 (FP) インデックス

個別カウント	カラム・データ・サイズ = 1 バイト	カラム・データ・サイズ = 2 バイト	カラム・データ・サイズ = 3 バイト	カラム・データ・サイズ = 3 バイト未満
<257	FP(1)	FP(1)	FP(1)	FP(1)
257 - 65536	—	FP(2)	FP(2)	FP(2)
65537 - 16777216	—	—	フラット	FP(3)
>16777216	—	—	—	フラット

FP(1) および FP(2) インデックスと同様に、データ型が 255 バイトよりも広く 4 バイト幅よりも狭いカラムでは FP(3) インデックスはサポートされません。

FP(3) インデックスの作成、オーバーフロー、または FP(3) インデックスへの強制的な移行は、検索テーブルで使用される領域が FP_LOOKUP_SIZE オプションの現在の値よりも小さく、FP_LOOKUP_SIZE_PPM の現在の設定で指定されているメイン・キャッシュ部分よりも少ない場合にのみ許可されます。

Sybase IQ で使用される検索ページの最大数は、FP_LOOKUP_SIZE オプションと FP_LOOKUP_SIZE_PPM オプションのうち、値が小さい方のオプションによって制御されます。『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」の「[FP_LOOKUP_SIZE オプション](#)」および「[FP_LOOKUP_SIZE_PPM オプション](#)」を参照してください。

表 6-8 では、次の式に基づいて FP(3) インデックスでサポート可能な一意の値の最大数が計算されます。

$$\text{FP_LOOKUP_SIZE} / (\text{Column-Data size} + \text{Cardinality size})$$

カーディナリティ・サイズは、検索ストアにすべての個別データのカーディナリティを保存するために予約されている領域です。カーディナリティ・サイズには 4 または 8 バイトのいずれかの値を指定できます。次の例では、8 バイトの最大値が指定されます。

表 6-8: FP(3) における最大ユニーク値

FP_LOOKUP_SIZE (MB)	カラム・データ型の幅 (バイト)					
	4	8	32	64	128	255
1MB	87381	65536	26214	14563	7710	3986
4MB	349525	262144	104857	58254	30840	15947
8MB	699050	524288	209715	116508	61680	31895
16MB	1398101	1048576	419430	233016	123361	63791
32MB	2796202	2097152	838860	466033	246723	127583
64MB	5592405	4194304	1677721	932067	493447	255166
128MB	11184810	8388608	3355443	1864135	986895	510333
256MB	16777216	16777216	6710886	3728270	1973790	1020667

注意事項

表 6-8 に示されている値は、オプション FP_LOOKUP_SIZE が特定の値のときのカラム内のユニーク値の数についての見積もりです。実際の値とは異なる可能性があります。このような差は、カウントが 4 バイトまたは 8 バイトで保存されることに原因があります。

表 6-8 は、FP_LOOKUP_SIZE の値は FP_LOOKUP_SIZE_PPM の値以下であるという条件に基づいています。

FP(3) インデックスの設定

3 バイトのインデックスを設定する際には、テンポラリ・キャッシュ・サイズを調整しなければならない場合があります。次のようにサーバー起動コマンド・ライン・パラメータ `-iqtc` か `sa_server_option` システム・プロシージャ `temp_cache_memory_mb` オプションを使用して、値を設定できます。

```
CALL sa_server_option('temp_cache_memory_mb', value)
```

列挙型 FP インデックスは、ハッシュ・オブジェクトを使用してカラムに表される値を管理します。3 バイトの FP で使用されるハッシュ・オブジェクトのサイズは、個別値の数とカラムの幅に応じて大きくすることができます。テンポラリ・キャッシュに十分な割り付けが行われていると、`HASH_PINNABLE_CACHE_PERCENT` オプションの値がデフォルト値の 20 パーセント以上増加しても、ハッシュ・オブジェクト全体をキャッシュ内に保持できるため、パフォーマンスを向上させることができます。

キャッシュの使用率

FP(3) インデックスを最大限に使用するには、`FP_LOOKUP_SIZE` オプションをデフォルトの 16MB よりも大きい値に設定します。FP(3) インデックスのカラムで許容される個別カウントの最大数については、表 6-8 を参照してください。表 6-8 には、値の数が 16777216 未満の場合の例も示されています。この場合、フラット FP へのロールオーバーは、想定された 16777216 よりも小さいユニーク・カウントで発生します。

ロード

3 バイトのインデックスがあるカラムには、データをロードするための追加のキャッシュが必要です。3 バイトのインデックスがあるカラムをロードする前に、`FP_LOOKUP_SIZE` を適切な値に設定します。

固定バッファの不足が発生すると、Sybase IQ は `.iqmsg` ファイルに警告を返します。このファイルには、予想されるスラッシングの通知も含まれています。

```
Warning: Hash Insert forced buffer unpinning detected for FP
Index
Warning: Hash Insert thrashing detected for FP Index
```

フラット FP インデックス

一意の値の数が 16777216 を超えると、検索テーブルは作成されません。各 FP エントリは、実際のカラム・セル値を含みます。

MINIMIZE_STORAGE が ON の場合、検索 FP エントリをフラット・スタイルに変換するオーバーヘッドを回避できます。特定のフィールドの個別ロー・カウントが 16777216 よりも増えると、FP(3) インデックスがフラット・スタイルの FP インデックスに自動的に変換されます。フラット・スタイルの FP を作成するには、テーブル作成時に IQ UNIQUE 値が 16777216 よりも大きくなるように指定します。

注意 DATE データ型を使用してテーブルを作成すると、DATE フィールドに 2 バイトの FP インデックスが作成されます。このインデックスは、データベース・オプション MINIMIZE_STORAGE の設定に依存しません。

DATE フィールドに 3 バイトの FP インデックスまたはフラット・スタイルの FP インデックスを作成するには、テーブルの作成時に次の値を使用します。

- 3 バイトの FP インデックスの場合 – 65537 ~ 16777216 の IQ UNIQUE 値
- フラット・スタイルの FP インデックスの場合 – 16777216 より大きい IQ UNIQUE 値

Low_Fast (LF) インデックス・タイプ

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) を含むカラムで使用するには理想的です。LF は Sybase IQ でもっとも高速なインデックスです。

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 インデックスは、以下の場合に使用します。

- カラムに含まれるユニークな値の数が 1,000 未満である。
- 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ている。

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

次の表は、Low_Fast インデックスの長所と短所の一覧です。

表 6-9: LF の長所／短所

長所	短所
高速。特に単一テーブルの SUM、AVG、COUNT、COUNT DISTINCT、MIN、および MAX 操作では高速である。	カラムに含まれるユニークな値が 10,000 個を超える場合は使用不可。 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

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・タイプも使用できます。

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、ユニークな値の数が多の場合にだけ、その他のインデックス (HG および HNG) を検討してください。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ます。

High_Group (HG) インデックス・タイプ

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。

推奨される使用法

HG インデックスは、以下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。この変更をユーザが意識することはありませんが、クエリ・パフォーマンスが向上します。例については、『パフォーマンス&チューニング・ガイド』の「[第 3 章 クエリと削除の最適化](#)」にある「[ORDER BY クエリ・パフォーマンスの強化](#)」を参照してください。

注意 外部キー・カラムには、それぞれ個別の HG インデックスが必要です。ただし、ジョイン・インデックスがある場合は、明示的に作成された HG インデックスと外部キー制約の両方を同じカラムに持たせることはできません。

High_Group の長所と短所

次の表は、High_Group インデックスの長所と短所の一覧です。

表 6-10: HG の長所／短所

長所	短所
GROUP BY を使用するクエリを高速で処理できる。 このインデックスを使用するとジョイン・インデックス処理が簡単になる。このインデックスは、ジョイン関係で使用されるカラムの推奨インデックスのうちの 1 つであり、もう 1 つは LF である。	HNG インデックスに比べて多くのディスク空き容量が必要 (生データの約 3 倍の領域を使用する場合がある)。 このインデックス・タイプでは、データの読み込みや削除にもっとも長い時間がかかる。削除のパフォーマンスを最適化する方法については、『パフォーマンス&チューニング・ガイド』の「 削除オペレーションの最適化 」を参照。 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。 このインデックスは、FLOAT、REAL、および DOUBLE 型のデータには推奨されない。

HG インデックスのロード・パフォーマンスの改善については、『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」の「[大きい単一 \(ファクト\) テーブルのロードの改善](#)」を参照してください。

他のインデックスとの比較

LF 決定要因となるのは、ユニークな値の数です。カラムに入る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多い場合には 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・カラムであるかと、そのカラムについての **GROUP BY** が処理されるかです。このいずれかが真であれば **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。両方とも偽の場合は、ディスク領域を節約するために **High_Non_Group** を使用します。

追加インデックス

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

インデックス・タイプを指定せずに **CREATE INDEX** 文を発行すると、Sybase IQ では常にデフォルトで **High_Group** インデックスが作成されます。

Sybase IQ では、任意の **UNIQUE**、**FOREIGN KEY**、または **PRIMARY KEY** 制約に対して自動的に **High_Group** インデックスが作成されます。単一カラム外部キーに対しては、非ユニークなシングルカラム **High_Group** インデックスが作成されます。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。

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）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。詳細については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「**CREATE INDEX 文**」にある「**注意事項**」で説明されている「マルチカラム・インデックス」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・タイプ

範囲検索を行う場合は、**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、グループ・オペレーション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。

逆に、HG インデックスのほうが効率的に処理できるとわかっているクエリを行う場合、またはカラムがジョインの一部であるか一意性を確保したい場合は、HG インデックスを使用してください。

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。

推奨される使用法

HNG インデックスは、以下の場合に使用します。

- ユニークな値の数が多 (1,000 超)
- そのカラムに対して GROUP BY を行う必要がない

High_Non_Group の長所と短所

次の表では、High_Non_Group 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 6-11: 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る。</p> <p>カラムにユニークな値が多数ある場合は、これがもっとも高速なインデックスとなるが、次に示すような若干の例外がある。</p>	<p>GROUP BY クエリでは、このインデックスは推奨されない。</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 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、このインデックスを使用できない。</p>

他のインデックスとの比較

- 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 の間で選択する場合、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。

追加インデックス

High_Group インデックスは、HNG カラムにも使用できます。

比較 (CMP) インデックス・タイプ

比較 (CMP) インデックスは、2つのカラム間の関係に対するインデックスです。比較インデックスは、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。

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。精度と位取りが等しい場合には、これらのカラムに対して CMP インデックスを作成できます。CHAR、VARCHAR、BINARY、および VARBINARY カラムでは、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、テーブルを作成した後に適切な比較インデックスを作成する方法を示しています。

```
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、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)
```

制限

CMP には、次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 インデックスは、基本となるジョイン・インデックスには複製されません。
- 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。
- ユーザは、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、および REAL データ型をサポートしません。

包含 (WD) インデックス・タイプ

包含 (WD) インデックスは、CHAR、VARCHAR、および LONG VARCHAR データ型のカラム文字列に含まれる単語を格納します。

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。非構造化データ分析オプションの詳細については、『Sybase IQ の非構造化データ分析の概要』を参照してください。

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を含んでいるカラムへの高速アクセスを実現したいときに使用します (たとえば図書目録レコードや Web ページなど)。

WD には、次の制限が適用されます。

- UNIQUE 属性を指定できません。
- WD インデックスは CONTAINS または LIKE 述部でのみ使用されます。
- カラム名には、ベース・テーブルの CHAR、VARCHAR、または LONG VARCHAR 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)。
- 区切り文字のリストは一重引用符で囲む必要があります。Sybase Central のインデックス作成ウィザードでは、区切り文字の入力を求めるプロンプトが表示されたときにこのことは指示されませんが、一重引用符を省略するとエラーが返されます。
- DELIMITED BY 句を省略した場合、または指定した *separators-string* 値が空の文字列 (1 対の単一引用符) である場合は、デフォルトの区切り文字セットが使用されます。デフォルトの区切り文字セットには、すべての 7 ビットの ASCII 文字のうち、7 ビットの ASCII 英数字ではないものと、ハイフンおよび単一引用符 (これらはデフォルトで単語の一部とみなされます) を除いたものが含まれます。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。
- 複数の DELIMITED BY 句と LIMIT 句が指定されると、エラーは返されませんが、各タイプで最後の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 clwd on foo(cl)
DELIMITED BY 'f' LIMIT 40 LIMIT 99 DELIMITED BY 'g'
DELIMITED BY 'h';
```

文 2:

```
CREATE WD INDEX clwd on foo(cl)
DELIMITED BY 'h' LIMIT 99;
```

- **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、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。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することはできません。

CHAR カラムを使用する場合は、少なくとも空白を区切り文字の 1 つに指定するか、デフォルトの区切り文字セットを使用することを推奨します。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が不正な結果を返すことがあります。たとえば `company_name` というカラムに、区切り文字で区切られた 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空白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 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。

- `sp_iqcheckdb` (DBCC 一貫性チェッカ) の `allocation`、`check`、`verify`、および `repair` モードは、**CHAR**、**VARCHAR**、および **LONG VARCHAR** カラムの **WD** インデックスをサポートしています。

WD の長所と短所

次の表では、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 6-12: WD の長所／短所

長所	短所
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。	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。
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。	一意性を確保すると、インデックスが使用不可になる。
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。	カラム内のデータが CHAR、VARCHAR、または LONG VARCHAR の場合のみこのインデックスを使用できる。
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。	

WD インデックスのロード・パフォーマンスの改善については、『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」の「[大きい単一\(ファクト\)テーブルのロードの改善](#)」を参照してください。

日付 (DATE)、時刻 (TIME)、日時 (DTTM) インデックス・タイプ

日付、時刻、日時の量を扱うクエリを処理するときには、次の 3 種類のインデックス・タイプが使用されます。

- DATE インデックスは、日付を扱うクエリを処理するとき、データ型 DATE のカラムに対して使用されます。
- TIME インデックスは、時間を扱うクエリを処理するとき、データ型 TIME のカラムに対して使用されます。
- DTTM インデックスは、日時を扱うクエリを処理するとき、データ型 DATETIME または TIMESTAMP のカラムに対して使用されます。

推奨される使用法

日時関連の関数や操作を含むクエリで DATE、TIME、DATETIME、または TIMESTAMP カラムが使用されているときは、次のような場合に DATE、TIME、または DTTM インデックスを使用します。

- 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 DATE、TIME、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 および 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 インデックスが使用できない場合は、DATE、TIME、または DTTM インデックスで結果を取得します。

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 DATE、TIME、DATETIME、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。DATE、TIME、DATETIME、TIMESTAMP カラムで推奨されるインデックスについては、「[追加インデックス](#)」(242 ページ) を参照してください。

この項の例で使用する **tab** テーブルには、次の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す。

```
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、DTTM インデックスを使用して処理を行うときには、DATEPART 範囲述部や IN リスト述部に対して次のような条件が求められます。

- 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。さらに、比較のもう一方が定数式です (整数または整数型のホスト変数など)。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。さらに、AND の両側が定数式です (整数または整数型のホスト変数など)。例を示します。

```
DATEPART(YEAR, col1) BETWEEN host-var1 AND host-var2
```

- IN 条件

IN 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。さらに、IN リスト内の値が定数式です。例を示します。

```
DATEPART(MONTH, col1) IN (1999, 2001, 2003)
```

注意 DATE、TIME、および DTTM インデックスは、一部の日付要素 (Calyearofweek、Calweekofyear、Caldayofweek、Dayofyear、Millisecond) をサポートしません。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

次のような範囲述部を含むクエリでは、DATE、TIME、または DTTM インデックスを使用して処理が行われます。

- 比較条件

```
SELECT * FROM tab WHERE col1 < '2002/10/09';
```

```
SELECT * FROM tab WHERE col2 >= '2002/01/01 09:12:04.006';
```

比較の一方がカラム名で、もう一方が定数式 (定数値またはホスト変数) 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;
```

以上のクエリでは、通常は HNG インデックスよりも DATE、TIME、または DTTM インデックスを使用した方が高速です。

次の3つの特殊な状況では、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。

- 述部の範囲がちょうど年単位です (実開始日が年始で、実終了日が年末です)。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でちょうど月単位です (実開始日が月始で、実終了日が月の月末です)。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';
```

- 述部の範囲がちょうど1日です。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つの状況では、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 および HG インデックスの方が高速です。

DATE/TIME/DTTM の長所と短所

次の表では、DATE、TIME、または DTTM 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 6-13: DATE/TIME/DTTM の長所／短所

長所	短所
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。	HNG インデックスよりも多くのディスク領域を使用する。
DATE、TIME、DTTM インデックスは作成と削除ができる。	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 DATE、TIME、DATETIME、または TIMESTAMP データ型の場合のみ使用できる。

DATE/TIME/DTTM インデックスの制限

DATE、TIME、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、および DTTM インデックスのパフォーマンスは、HNG インデックスに関係しています。HNG に比べると、サポートされる状況では、通常は DATE、TIME、および DTTM インデックスの方が高速です。「推奨される使用法」で説明した特殊な状況では、DATE、TIME、および DTTM インデックスはさらに高速に処理されます。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。

追加インデックス

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 DATE、TIME、または DTTM インデックスを必ず使用することをおすすめします。さらに、DATE、TIME、DATETIME、または TIMESTAMP 型のカラムに対して HG または LF インデックスを作成した方がよいこともあります (特に、カラムに対する等号述部を評価する場合)。また、そのカラムを 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 インデックスを推奨します。

TEXT インデックス

注意 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。詳細については、『Sybase IQ の非構造化データ分析の概要』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包含 (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するクエリは、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。

アドホック・ジョインのパフォーマンスの最適化

アドホック・ジョインの処理をもっとも高速にするには、以下で参照されるすべてのカラムに Low_Fast または High_Group インデックスを作成します。

- アドホック・ジョイン・クエリの WHERE 句
- 集約関数以外のアドホック・ジョイン・クエリの HAVING 句の条件

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は、LF または HG インデックスを持っている必要があります。

インデックスの選択

インデックス・タイプの選択方法を要約したクイックチャートを以下に示します。

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 の選択リスト内の 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 インデックスを持ってい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

識別の条件	選択するインデックス
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 データ型のカラムを識別する。	DATE、TIME、または DTTM
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作成され、このインデックスは個々のテーブルの一部およびそのカラムを含むすべてのジョイン・インデックスの一部となります。CMP およびマルチカラム HG のインデックスはこの規則の例外です。

個々のテーブル内の既存のカラム・インデックスにすでにデータが入っている場合は、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。ジョイン・インデックスの一部として前にロードされたテーブルにインデックスを追加すると、自動的にデータも挿入および「同期」されます。同期の詳細については、「[ジョイン・インデックスの同期](#)」を参照してください。

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だとわかった場合に便利です。これを使用すると、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、それから 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したほうが、かかる時間は短くなります。

ジョイン・インデックスの使用

同じデータベース内の特定のテーブルが通常は一定の方法でジョインされることがわかっている場合は、それらのテーブルのジョイン・インデックスを作成できます。

ジョイン・インデックスを作成すると、テーブル・カラムに関連する内部構造が新しく作成されます。この内部構造は、内部、左外部、および右外部ローを含む 2 つまたはそれ以上のテーブルを表します。

ジョイン・インデックスによって向上するクエリのパフォーマンス

ジョイン・インデックスを使用すると、クエリ時に初めてジョインを定義する方法 (アドホック・ジョイン) よりも、通常はクエリのパフォーマンスが向上します。ただし多くの場合は、ジョイン・インデックスを作成しなくてもジョイン・カラムで最適なパフォーマンスを実現できます。

ジョイン・インデックスのロード時の考慮事項

ジョイン・インデックスは、他の IQ インデックスよりも多くの領域とロード時間を必要とします。ジョイン・インデックスをロードするには、まず基本となるテーブルをロードしてから、ジョイン・インデックスをロードする必要があります。

クエリでのジョイン・インデックスの使用法

作成したジョイン・インデックスの用途は、SELECT 文の条件によって決まります。FROM 句内のテーブルを WHERE 句で指定した関係によってジョインするジョイン・インデックスや、ナチュラル・ジョインまたはキー・ジョインの ANSI ジョイン構文に基づくジョイン・インデックスがある場合、これらのインデックスはクエリ処理を速めるのに使用されます。それ以外の場合は、個々のテーブルのインデックス間のアドホック・ジョインがクエリ時に実行されます。SELECT 内のテーブルのサブセットに対するジョイン・インデックスがある場合、Sybase IQ はそのジョイン・インデックスを使用してアドホック・ジョインの結果生成を速めます。

ジョイン・インデックス間の関係

Sybase IQ のジョイン・インデックスは、1 対多のジョイン関係をサポートしています。1 対多の関係の簡単な例は、営業担当者と顧客との関係です。営業担当者には 2 人以上の顧客がいますが、顧客には営業担当者は 1 人しかいません。

このような関係には、複数のレベルがある場合があります。しかし管理者は常に、2つのテーブルの間、または1つのテーブルと、下のレベルのジョインとの間でジョイン関係を指定します。関係の「多」側を表すテーブルを「トップ・テーブル」といいます。詳細については、次の「[ジョイン階層の概要](#)」を参照してください。

ジョインがアドホックになるとき

1つのクエリに含まれるすべての参照テーブルを処理するジョイン・インデックスがない場合、クエリはアドホック・ジョインによって解析されます。多対多のジョイン関係を表すジョイン・インデックスを作成することは不可能なため、そのような関係に対してはアドホック・クエリを発行することしかできません。アドホック・クエリは、パフォーマンスを犠牲にして柔軟性を実現するものです。ジョイン・インデックスのための十分な領域があり、多対多の関係またはマルチレベルのスター・ジョイン・インデックスが必要ない場合は、パフォーマンスが重視される部分にジョイン・インデックスを作成すると役に立ちます。

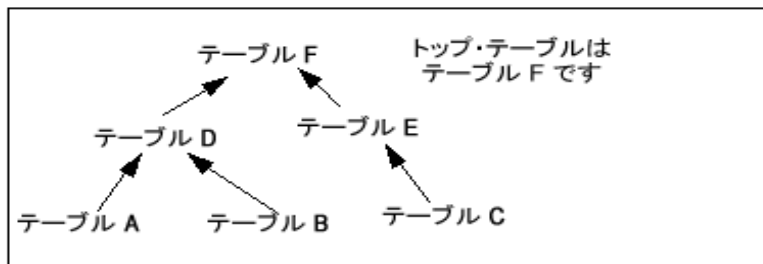
ジョイン階層の概要

Sybase IQ でサポートされるジョイン関係にはすべて階層があります。ジョイン階層は、ジョイン内のすべてのテーブルのつながり方を示すツリーと考えるください。

Sybase IQ のジョイン階層では、このツリーの最上部つまりジョインの終端にテーブルが1つあります。このテーブルは「トップ・テーブル」と呼ばれ、他のテーブルはここにつながりますが、このテーブルからは他のテーブルにつながっていません。トップ・テーブルは常に、1対多の関係の「多」側です。

ジョインの複雑さにより、ツリーの最下部とジョインの最初に向けてテーブルが一直線につながっている場合もあれば、ツリーを下へたどるにつれて横にたくさんの枝が出ている場合もあります。次の図では、枝が2つあるジョイン階層を示します。

図 6-1: ジョイン関係の階層



ジョイン階層は、以下のようになっています。

- 1つのテーブルは一度しか出てこない。
- 1つのテーブルから他のテーブルへは一度しかつながらない (出る矢印は1本のみ)。
- すべてのテーブルがつながっている。

ジョイン・インデックス内のコラム

テーブル A から E までを、ABCDE というジョイン・インデックスにジョインしたとします。各テーブルに 2 コラムのデータが入っている場合、このジョイン・インデックスには合計 14 コラムが入ることになります。Sybase IQ では、ジョインされたすべてのテーブル (トップ・テーブルを除く) について、追加の ROWID コラムがそれぞれ作成されます。この例では、コラムが 10 個 (5 つのテーブルそれぞれから 2 つずつ) と、ROWID コラムが 4 つになります。

LOAD TABLE 文または INSERT 文の NOTIFY オプションを使用すると、コラム・インデックスにデータを挿入したときに通知メッセージを受け取ることができます。この通知メッセージには、ジョイン・インデックス内の各コラム (ROWID を含む) が示されます。

メッセージの表示頻度は NOTIFY_MODULUS オプションで設定でき、このオプションの値は CREATE DATABASE または LOAD TABLE コマンドで変更できます。これらのメッセージの例については、「[通知メッセージの解釈](#)」(558 ページ) を参照してください。

ジョイン・インデックスのコラムのデータ型、精度、位取りは同じであることが必要です。

クエリ解析におけるジョイン階層

Sybase IQ では、ジョイン・インデックスで指定された完全なジョイン関係を扱うクエリを解析するときも、そのジョイン関係の連続サブセットを扱うクエリを解析するときも、同じジョイン・インデックスが使用されます。このようなサブセット関係のために別個のジョイン・インデックスを作成する必要はありません。

たとえば、[図 6-1 \(246 ページ\)](#) で示したテーブルをジョインするジョイン・インデックス ABCDEF があるとします。Sybase IQ では、ジョイン・インデックス ABCDEF を使用して、次の関係を扱うクエリを解析できます。

- 関係全体
- テーブル A 対テーブル D
- テーブル A 対テーブル D 対テーブル F
- テーブル B 対テーブル D

- テーブル B 対テーブル D 対テーブル F
- テーブル D 対テーブル F
- テーブル C 対テーブル E
- テーブル E 対テーブル F
- テーブル C 対テーブル E 対テーブル F

ただし、たとえばテーブル E 対テーブル D に対するクエリを解析するときジョイン・インデックス ABCDEF を使用することはできません。

1 対多の関係

1 対多のジョイン関係では、あるテーブルの 1 つのローは別のテーブルの 1 つまたは複数のローと一致することがありますが、最初のテーブルのローには、2 つめのテーブルの同じローと一致するものは 1 つしかありません。この状態を真にするには、最初のテーブルのジョイン・カラムに入る値をユニークにしなくてはなりません。

いずれかのテーブルに、もう一方のテーブルと一致するローがない場合もあります。この場合は外部ジョインを構築します。Sybase IQ では外部ジョインが完全にサポートされています。詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

ジョイン・カラムが複数のカラムで構成されている場合、その値の組み合わせは、1 対多の「1」側でユニークでなければなりません。たとえば、iqdemo データベースでは、Customers テーブルの ID と SalesOrders テーブルの CustomerID にそれぞれ顧客 ID が含まれています。Customers テーブルには顧客ごとに 1 つのローがあるため、各ローの ID カラムにはユニークな値が含まれます。SalesOrders テーブルには、顧客が行う取引ごとに 1 つのローが含まれます。それぞれの顧客は多くの取引を行うことが考えられるため、SalesOrders テーブルには、CustomerID カラムに同じ値を持つローが複数あります。

Customers.ID を SalesOrders.CustomerID にジョインすると、ジョイン関係は 1 対多になります。次の例に示すように、Customers の各ローに対して、SalesOrders のローが数多く対応する可能性があります (出力例のローは 15 に制限されています)。

```
SELECT SalesOrders.ID, SalesOrders.CustomerID,
Customers.GivenName
from SalesOrders, Customers
where SalesOrders.CustomerID = Customers.ID
```

ID	CustomerID	GivenName
2001	101	Michaels
2005	101	Michaels
2125	101	Michaels
2206	101	Michaels

2279	101	Michaels
2295	101	Michaels
2337	101	Michaels
2389	101	Michaels
2447	101	Michaels
2560	101	Michaels
2583	101	Michaels
2002	101	Beth
2142	101	Beth
2318	101	Beth
2338	101	Beth

警告！ 1 対多の関係が不正な場合、「1」側のテーブルから余分なローを削除しないと、ジョインを同期することはできません。同期しようとする、ローの重複によるエラーが発生し、トランザクションがロールバックします。

ジョイン・インデックスを作成するときには、ANSI FULL OUTER ジョイン構文を使用します。Sybase IQ では、このインデックスを完全な外部ジョインとして保存します。その後は、ジョイン・インデックス内のカラムに対してクエリを発行するときに、完全な外部ジョインや、内部、左外部、右外部のジョイン関係を指定できます。Sybase IQ は、ジョイン・インデックスのうち、所定のクエリに必要な部分だけを使用します。

複数テーブルのジョインとパフォーマンス

複数テーブルのジョインに関するルールを次に示します。

- テーブルを 1 対多の関係の「1」側にできるのは 1 回だけです。たとえば、ジョイン・インデックスまたはジョイン・クエリの中で、テーブル A をテーブル B に 1 対多の関係でジョインし、かつテーブル A をテーブル C に 1 対多の関係でジョインすることはできません。これらの関係では、それぞれ別個のジョイン・インデックスを作成する必要があります。
- 1 つのテーブルは関係階層に一度だけ定義できます。そのため、たとえば、テーブル A がテーブル B にジョインされ、テーブル B がテーブル C にジョインされ、テーブル C がテーブル A にジョインされるようなジョイン関係を事前に定義することはできません。定義済みのジョインを使用すると、テーブル A 対テーブル B の関係とテーブル C 対テーブル A の関係を別々にクエリすることができます。これを行うには、これらの関係それぞれに別個のジョイン・インデックスを作成してください。
- テーブルは、別のテーブルやジョイン定義にジョインすることができます。たとえば、テーブル A をテーブル B にジョインするジョイン・インデックスや、テーブル C をテーブル A と B のジョインにジョインするジョイン・インデックスを作成できます。

- 階層のトップ・テーブルは、他の階層との1対多の関係における「多」側です。
- もっとも有用なジョイン・インデックスは、通常は2つのテーブル間のジョインです。

環境によっては、ジョイン関係のサブセットに別個のジョイン・インデックスを作成できます。ジョイン・インデックスのサブセットのトップ・テーブルに、完全なジョイン・インデックスのトップ・テーブルよりも極端に少ない数のローしかない場合、サブセットのクエリは、サブセットのテーブルだけをクエリで使用した場合の完全なジョイン・インデックスについて同じクエリを行うよりも高速になります。

もちろんこのアプローチでは、追加のジョイン・インデックスを構築するのに、より大きなディスク領域とインデックス構築時間（メンテナンスの増加は言うに及ばず）が必要となります。サブセット・ジョイン・インデックスの場合、追加ジョイン・インデックスは、完全なジョイン・インデックスにすでに含まれている情報のサブセットを反復しています。このような特定のジョイン関係においては、クエリの速度とアプリケーションのディスク使用率のどちらがより重要かを決定しなくてはなりません。また、現在のバージョンの Sybase IQ では、ジョイン・インデックスを使用しても以前のリリースと同様のパフォーマンス向上を実現できないことがあるという点に注意してください（特に関係階層に複数のレベルが含まれている場合）。

ジョイン・インデックスの作成手順

ジョイン・インデックスを作成するには、以下の手順をすべて実行します。

- 1 CREATE TABLE コマンドまたは Sybase Central を使用して、ジョイン・インデックスに含まれるテーブルを作成します。これは永久テーブルでなければなりません。テンポラリー・テーブルはジョイン・インデックスの作成に使用できません。
- 2 任意のジョインに含まれる基本となるテーブル内で、カラムの特定のペアに関連するジョイン条件を特定します。

ジョインの条件やデータ構造に関するその他の仮定を明確にするために、データベースのスキーマを定義することが重要です。このスキーマは、外部キーとプライマリ・キーの関係を表し、スキーマ設計のその他のベスト・プラクティスに従う必要があります。外部キーに関連付けられるカラムは、データ型、精度、位取りが等しくなければなりません。

この関係がキー・ジョインに基づいている場合は、手順1の CREATE TABLE コマンドまたは ALTER TABLE コマンドで、ジョイン条件を参照整合性制約（プライマリ・キー宣言と外部キー宣言）として定義する必要があります。

- 3 ジョインに含まれる各カラムに対してプライマリ・キーを作成します。

- 4 ジョインされるテーブルのカラム・インデックスを作成します。

Sybase IQ でテーブル間にジョイン・インデックスを作成すると、そのジョイン・インデックス内では、それぞれのテーブルですでに定義されている IQ カラムのインデックス・タイプとデータ型が使用されます。ベース・テーブルのマルチカラム・インデックスは、そのベース・テーブルを使用して作成されたジョイン・インデックスに複写されません。

- 5 `LOAD TABLE` コマンドを使用して、テーブルにデータをロードします。また、`INSERT INTO` コマンドを使用して、既存のテーブルにデータを追加することもできます。

注意 ジョイン・インデックスそのものに挿入するのではなく、ジョイン・インデックスの各テーブルに単一テーブルの挿入として挿入してください。この方法は、インデックス付きデータに関する ANSI 規定に準拠しています。

- 6 `CREATE JOIN INDEX` コマンドを発行するか、Sybase Central のジョイン・インデックス作成ウィザードを使用して、ジョイン・インデックスを作成します。「[ジョイン階層の概要](#)」で説明したとおり、ジョイン階層はこの手順の一部として指定してください。
- 7 手順を実行する順序によっては、次で示すようにジョイン・インデックス内でテーブルを同期しなくてはならない場合があります。ジョイン・テーブルにデータがあると、自動的に同期が起こります。

手順がすべて完了するまで、インデックスは使用できません。ただし、必要に応じて手順の順序を一部調整することはできます。

- テーブルの作成時に関係を定義すると、手順 1 と 2 を一度に済ませることができません。
- データのロードは、ジョイン・インデックスの作成前と作成後のどちらでもかまいません。ジョイン・インデックスの作成後に、基本となるカラム・インデックスへデータをロードする場合は、同期のステップを実行してください。

ジョイン・インデックスの作成に必要な権限

あるテーブルを含むジョイン・インデックスを作成、変更、または同期できるのは、そのテーブルの所有者と DBA のみです。DBA でない場合は、そのテーブルの所有者であり、かつ `RESOURCE` 権限を持っていないかぎり、ジョイン・インデックスを作成できません。

データの挿入と削除の詳細については、「[第 7 章 データベースへのデータの出入力](#)」を参照してください。完全な構文については、『リファレンス：文とオプション』の「[CREATE TABLE 文](#)」、「[ALTER TABLE 文](#)」、「[LOAD TABLE 文](#)」、「[INSERT 文](#)」、「[SYNCHRONIZE JOIN INDEX 文](#)」を参照してください。以下の項では、ジョイン・インデックスの作成について詳しく説明します。

ジョイン・インデックスの同期

ジョイン・インデックスを使用する前には、ジョイン・インデックス・テーブルのデータを同期してください。同期を行うことで、ジョインに適した正しい順序でデータがロードされるようになります。

ジョイン・インデックスを作成すると、自動的に同期が実行されます。データのロードや挿入を伴うトランザクションの完了前に同期を行うと、そのテーブルは即座に、誰でも読み込める状態になります。データをロードしたらジョイン・インデックスを明示的に同期しなくてはなりません。ただし例外として、ジョイン階層のトップ・テーブルに変更を加えた場合のみ、ジョイン・インデックスは自動的に同期されます。

明示的に同期を行うには、以下のコマンドを実行します。

```
SYNCHRONIZE JOIN INDEX [join-index-name [, join-index-name]
```

インデックス名を省略すると、すべてのジョイン・インデックスが同期されます。

同期のパフォーマンスについてのヒント

同期には時間がかかります。パフォーマンスを改善するには、以下のことを試してみてください。

- オフピークの時間帯に同期の予定を組む。
- すべてのジョイン・インデックスを一度に同期するのではなく、1つずつ行う。
- 挿入と削除の作業をすべて完了した後に、同期を行う。挿入や削除を1つ行うごとに同期を行うのは得策ではありません。それは、ジョイン・インデックスの更新にかかる時間が、テーブルを更新する順序によって大きく異なるためです。一連の更新内容を同期した場合には、テーブルの変更をジョイン・インデックスに適用するための最適な順序が Sybase IQ によって選択されます。

テーブル間のジョイン関係の定義

ジョイン・インデックスを作成するときは、ジョイン内の関連するペアの関係を定義してください。関連するペアは常に2つのテーブルですが、あるテーブルを別のジョイン関係に関連させることによって関係を指定することもできます。

関係によって、指定する回数が1回のとくと2回のとことがあります。

- 「キー・ジョイン」は、あるテーブルのプライマリ・キーを別のテーブルの外部キーに関連付けます。キー・ジョインでは、CREATE TABLE または ALTER TABLE コマンドを使用して基本となるテーブルを作成または変更するときに PRIMARY KEY と FOREIGN KEY を指定する必要があります。

- すべてのジョインについて、**CREATE JOIN INDEX** コマンドを使用してジョイン・インデックスを作成するときに、関係を指定します。ジョインは、テーブルをリストする順序、指定するカラム、およびジョインの型 (キー・ジョイン、ナチュラル・ジョイン、または ON 句のジョイン) によって定義されます。

ジョイン関係のルールは、以下のとおりです。

- ジョイン関係内のテーブルのペアには、それぞれ最低 1 つのジョイン・カラムが入っていないてはなりません。
- いずれのテーブルにもジョイン・カラムがなくてはなりません。
- 2 つのテーブルのカラム数が同じで、指定された時点でジョイン・カラムが各テーブル・リスト内の同じ位置にあれば、その 2 つのテーブルは複数のジョイン・カラムを持つことができます。2 つのテーブルのリストの順序によって、カラムがどのように対応するかが決まります。

外部参照の使用

Sybase IQ では外部キーを使用して、ジョイン・インデックス内で使用するカラムの関係性を定義したり、クエリを最適化したりします。

ある種のジョイン・インデックスには外部キーに基づくキー・ジョインが必要とされることに注意してください。

Sybase IQ は、マルチカラム外部キーに基づくキー・ジョイン・インデックスをサポートしていません。

テーブル定義でのジョイン関係の例

次の例では、プライマリ・キーと外部キーを使用したジョイン関係の指定方法を示します。この例では 1 人の顧客が多数の注文を持つことができるため、**Customers** テーブルの **ID** カラム (プライマリ・キー) と **SalesOrders** テーブルの **CustomerID** カラムの関係は、1 対多です。そのため、**SalesOrders** の **CustomerID** を、**Customers** テーブルの **ID** カラムを参照する **FOREIGN KEY** として指定します。

最初の例では、**ID** カラムをプライマリ・キーとする **Customers** テーブルを作成します。簡単にするため、他のカラムは省略記号 (...) で表します。

```
CREATE TABLE GROUP0.Customers
( ID INTEGER NOT NULL,
...
PRIMARY KEY (ID),)
```

次に、カラムが 6 つある **SalesOrders** テーブルを作成し、**CustomerID** というカラムをプライマリ・キーに指定します。また、**SalesOrders** テーブルの **CustomerID** カラムを **Customers** テーブルの **ID** カラムに関連付ける外部キーを追加する必要があります。

外部キーは、テーブル作成時でもその後でも追加できます。この例では、REFERENCES 句を CREATE TABLE 文のカラム制約として含めることで、外部キーを追加します。

```
CREATE TABLE GROUPO.MySalesOrders
  (ID INTEGER NOT NULL,
  CustomerID INTEGER
  REFERENCES GROUPO.Customers (ID),
  OrderDate DATE NOT NULL,
  FinancialCodesID CHAR(2),
  Region CHAR(7),
  SalesRep INTEGER NOT NULL,
  PRIMARY KEY (ID),)
```

別の方法として、次の ALTER TABLE 文で行っているように、REFERENCES 句なしでテーブルを作成し、後から外部キーを追加することもできます。どちらかの文を実行できますが、両方を実行できません。

```
ALTER TABLE GROUPO.MySalesOrders
  ADD FOREIGN KEY ky_so_customer (CustomerID)
  REFERENCES GROUPO.Customers (ID)
```

スター・ジョイン・インデックスの場合は、テーブル作成に特殊な要件があります。例については、「[スター・ジョイン](#)」(257 ページ)を参照してください。

ジョイン・インデックス作成時のジョイン・タイプの指定

ジョイン・タイプは常に FULL OUTER で、キーワード OUTER はオプションです。また、以下のいずれかを実行してください。

- 2 つのテーブルから同じ名前を持つ対等なカラムをジョインする場合は、ジョイン・タイプに NATURAL JOIN を指定します。
- キーに基づいてカラムをジョインする場合は、基本となるテーブルの関係を、PRIMARY KEY を参照する FOREIGN KEY として指定する必要もあります。
- 2 つのテーブルのカラムにある等価の値をジョインする (等価ジョイン) 場合は、ON 句を指定します。

これらのルールは、ANSI 構文の要件に準拠しています。

ジョイン・インデックス作成時の関係の指定

非キー・ジョインでは、ジョイン・インデックスを作成するときにテーブルを指定する順序によって、テーブル間のジョイン関係の階層が決定されます。CREATE JOIN INDEX 文では、以下の 2 つの方法によってジョイン階層を指定できます。

- 各テーブルを階層の一番下から順にリストして、テーブルの各ペアの間のジョイン関係を書き出します。リストの最後にくるテーブルが、その階層のトップ・テーブルとなります。たとえば図 6-1 (246 ページ) では、F がトップ・テーブルで E がその下、C が階層の最下層になります。これら 3 つのテーブルのジョイン階層を指定するには、次のようにします。

```
C FULL OUTER JOIN E FULL OUTER JOIN F
```

- ジョイン関係の評価順を制御するには、カッコを使用します。数学の計算順序と同様、カッコによって評価順が決まります。つまり、もっとも内側のペアが最初に評価されます。この方法では、まず一番外側のカッコで囲まれたトップ・テーブルから始めて、次のレベルに進み、一番内側のカッコにはもっとも下の 2 つのレベルを記述します。この方法で前述の 3 つのテーブルのジョインを指定すると、次のようになります。

```
(F FULL OUTER JOIN (C FULL OUTER JOIN E))
```

1 つめの方法と同様、一番下のレベルのテーブルをもっとも内側のカッコの中に記述することに注意してください。

注意 この 3 つのテーブルはここで解説した方法でジョインできますが、図 6-1 に示す完全な階層を作成するにはキー・ジョインを使用する必要があります。詳細については、「[ジョイン階層のタイプ](#)」を参照してください。

ジョイン・インデックスを作成すると、ログのメッセージによってジョインのトップ・テーブルが識別されます。例を示します。

```
[20691]: Join Index 'join_on_tabletable' created from the following join relations:
[20694]:      Table Name      Relationship
[20697]: -----
[20696]: 1. join_on_table_a joined to 'join_on_table_b' One >> Many
[20692]: The ultimate/top table is join_on_table_b
[20697]: -----
```

CREATE JOIN INDEX 文の実行

ジョイン・インデックスを作成するための構文については、『SQL Anywhere サーバ - SQL リファレンス』の「[第 1 章 SQL 文](#)」の「[CREATE JOIN INDEX 文](#)」を参照してください。

例 1: キー・ジョイン

この例では、SalesOrders テーブルと Customers テーブルの間にキー・ジョインのジョイン・インデックスを作成します。このキー・ジョインは、SalesOrders テーブルの CustomerID カラムを Customers テーブルのプライマリ・キー ID に関連付ける外部キー ky_so_customer に基づいています。インデックスには、自由に名前をつけることができます。この例では、キー・ジョインの基礎となる外部キーを識別するために、ky_so_customer_join という名前をつけます。

```
CREATE JOIN INDEX ky_so_customer_join
FOR GROUPO.Customers FULL OUTER JOIN GROUPO.SalesOrders
```

例 2：ON 句のジョイン

この例では、ON 句を使用して同じ 2 つのテーブルのジョイン・インデックスを作成します。この構文は、外部キーの有無にかかわらず使用可能です。

```
CREATE JOIN INDEX customer_sales_order_join
FOR GROUPO.Customers FULL OUTER JOIN GROUPO.SalesOrders
ON Customers.ID=SalesOrders.CustomerID
```

例 3：ナチュラル・ジョイン

ナチュラル・ジョインを作成するには、ジョインされたカラムの名前が同じでなくてはなりません。前の例のテーブルにナチュラル・ジョインを作成していると、まったく見当違いの結果になります。次のコマンドは、SalesOrders テーブルの CustomerID カラムに Customers テーブルの ID カラムをジョインするのではなく、2 つのテーブル内の異なる ID カラムをジョインしますが、認められません。

```
CREATE JOIN INDEX customers_sales_order_join
FOR GROUPO.Customers NATURAL FULL OUTER JOIN
GROUPO.SalesOrders
```

SalesOrders テーブルと SalesOrderItems テーブルそれぞれの id カラム間にナチュラル・ジョインを作成したほうが合理的です。この場合、同じ名前のカラムには、対応する値が入っていません。これら 2 つのテーブルの間のナチュラル・ジョインに基づいてジョイン・インデックスを作成するコマンドは、次のとおりです。

```
CREATE JOIN INDEX sales_order_so_items_join
FOR GROUPO.SalesOrders NATURAL FULL OUTER JOIN
GROUPO.SalesOrderItems
```

Sybas Central でのジョイン・インデックスの作成

Sybase Central を使用してジョイン・インデックスを作成する方法については、『Sybase IQ の概要』の「第 5 章 インデックス機能とデータのロード」の「ジョイン・インデックスの作成」を参照してください。

ジョイン階層のタイプ

Sybase IQ では、2 つの異なるタイプのジョイン階層をサポートしています。

- リニア・ジョイン
- スター・ジョイン

アドホック・ジョインは、リニア・ジョインとスター・ジョインのどちらにも作成できます。ジョイン・インデックスはリニア・ジョインで使用するために設計されたものです。

リニア・ジョイン

リニア・ジョインは、枝分かれしていないツリーと考えられます。階層の各テーブルはその上のテーブルに関連付けられており、最終的にトップ・テーブルに到達します。図 6-1 (246 ページ) を参照してください。この図では、テーブル A、D、F は、リニア・ジョイン階層を構成しています。テーブル C、E および F も、もう 1 つのリニア・ジョイン階層を形成します。

リニア・ジョインでは、テーブルの各ペアは 1 対多の関係を表し、ペアの下のテーブルは「1」側、上のテーブルは「多」側となります。リニア・ジョイン階層は、基礎となるジョイン条件 (キー・ジョイン、ナチュラル・ジョインまたは ON 句ジョイン) のいずれにも依存できます。

スター・ジョイン

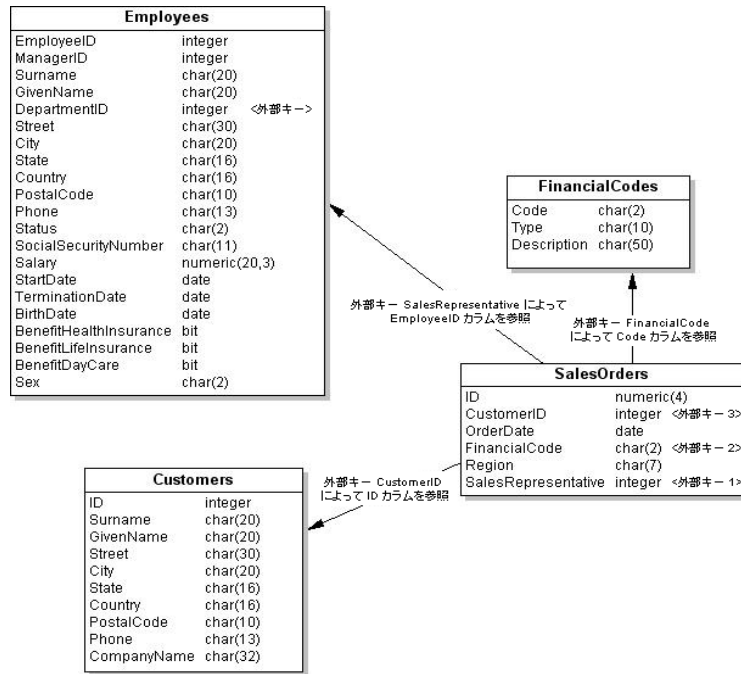
スター・ジョインは、多くの分岐から成り、それぞれの分岐が中央の 1 つのテーブルに直接関連している構造です。図 6-1 では、テーブル D、F、E が非常に単純なスター・ジョインとなっています。一般的には、テーブル F が多数のテーブルの中心にあり、それぞれのテーブルがテーブル F にジョインしています。

スター・ジョインでは、ジョインの中心にある 1 つのテーブルに複数のテーブルが関連して、1 対多の関係になります。ジョインの中心にあるテーブルは関係の「多」側、その周囲の各テーブルは関係の「1」側です。「1」側の各テーブルには、独自のユニークなプライマリ・キーを持つ値セットが含まれます。このテーブルは、関係の「多」側のテーブル内の外部キーによって、関係の「1」側のテーブルのプライマリ・キーに関連付けられます。

スターの中央にある「多」側のテーブルは、「ファクト・テーブル」と呼ばれることがあります。それに関連する「1」側のテーブルは、「次元テーブル」と呼ばれます。

例

次の例では、SalesOrders テーブルに3つの外部キーが含まれており、それぞれが別のテーブルのプライマリ・キーに関連付けられています。



このテーブルを作成するには、次のコマンドを使用します。

```
CREATE TABLE GROUPO.SalesOrders (
    ID NUMERIC (4) NOT NULL IQ UNIQUE (648),
    CustomerID INTEGER NOT NULL IQ UNIQUE (109),
    OrderDate date NOT NULL IQ UNIQUE (376),
    FinancialCode CHAR (2) NULL IQ UNIQUE (1),
    Region CHAR (7) NULL IQ UNIQUE (5),
    SalesRepresentative INTEGER
    NOT NULL IQ UNIQUE (11)
);

COMMENT ON TABLE GROUPO.SalesOrders is
'sales orders that customers have submitted
to the sporting goods company';

ALTER TABLE GROUPO.SalesOrders
ADD FOREIGN KEY FK_CustomerID_ID (CustomerID)
REFERENCES GROUPO.Customers (ID)
ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE GROUPO.SalesOrders
```

```
ADD FOREIGN KEY FK_FinancialCode_Code (FinancialCode)
REFERENCES GROUP0.FinancialCodes (Code)
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE GROUP0.SalesOrders
ADD FOREIGN KEY
FK_SalesRepresentative_EmployeeID
(SalesRepresentative)
REFERENCES GROUP0.Employees (EmployeeID)
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

上の図からわかるように、このスター・ジョインの中心となっているのは **Salesorders** テーブルです。このテーブルの外部キー・カラムそれぞれには、そのカラムが参照するプライマリ・キーのインスタンスが多数含まれています。たとえば、次のように入力します。

```
SELECT SalesRepresentative FROM SalesOrders
WHERE SalesRepresentative = 299
```

その結果、**SalesRepresentative** カラムの値が 299 であるローが、114 個表示されます。

しかし、次のように入力すると結果が異なります。

```
SELECT EmployeeID FROM Employees
WHERE EmployeeID = 299
```

この場合は、**EmployeeID** カラムの値が 299 であるローが 1 つだけ表示されます。

注意 どのジョインでも、クエリの最適化は基礎となるプライマリ・キーに大いに依存します。このときに外部キーは必要ありません。しかし、外部キーを使用するとメリットがあります。Sybase IQ は、プライマリ・キーと外部キーの関係を確認するためにロードを設定している場合は外部キーを使用します。

Sybase IQ は、特定のジョインに対する複数のジョイン・キー・カラムを使用したスター・ジョイン方式のジョイン・インデックスをサポートしていません。

真のスター・ジョイン（つまり他の次元とジョイン・キーを共有する次元が存在しない場合）では、IQ クエリ・オブティマイザは 1 つの句の中で最大 24 の次元テーブルを許可します。しかし、次元の数が増えるとクエリ処理にかかる時間が急激に増加するため、この上限に近づくにつれてパフォーマンスが低下します。

外部キーを作成するには、「[プライマリ・キーと外部キーの作成](#)」(204 ページ)を参照してください。外部キーの詳細については、「[エンティティ整合性と参照整合性の宣言](#)」(387 ページ)を参照してください。

ジョイン・インデックスに含まれるテーブルの修正

いったんジョイン・インデックスを作成すると、ジョイン・インデックスおよび基礎となるテーブルやインデックスに対して実行できる変更の種類が制限されます。

ジョイン・インデックスの一部となっているテーブルを削除することはできません。また、**ALTER TABLE** を使用して、ジョイン・インデックスの一部となっているカラムを追加、削除、修正することはできません。いずれの場合も、最初にジョイン・インデックスを削除する必要があります。そうすると、テーブルの削除や、ジョイン・インデックスを形成しているカラムの修正が可能になります。

ジョイン・インデックスを形成しているテーブルにカラムを追加することは可能です。しかし、次の項で説明するとおり、これらのカラムへのデータの挿入には制限があります。

ジョイン関係に含まれないカラムのインデックスを削除することや、ジョイン・インデックス内のテーブルの非ジョイン・カラムを修正することは可能です。しかし、ジョイン・カラムのインデックスやジョイン・カラムそのものを削除することはできません。定義済みジョイン関係にあるカラムには、最低 1 つのインデックスが必要です。ジョイン・インデックスを形成するすべてのカラムに **HG** インデックスと **LF** インデックスのどちらかが入っていることが、非常に望ましいとされます。

Sybase IQ は、ベース・テーブルを変更するときに、その変更内容をジョイン・インデックスにも自動的に適用します。非ジョイン・カラムに対して **ALTER TABLE** を実行した後に、ジョイン・インデックスを同期する必要はありません。

ジョイン・インデックスの **ALTER TABLE** に関する制限には、その他に以下のようなものがあります。

- **NATURAL** ジョイン条件に出し入れするカラムの名前を変更することはできません。
- 前に指定した **NATURAL** ジョインを形成するカラムを追加することはできません。
- ジョイン・インデックスで使用中のジョイン条件と **PRIMARY KEY/FOREIGN KEY** 関係が一致する場合、この関係を削除することはできません。
- ジョイン条件の一部となっているカラムから **NOT NULL** 制約を削除することはできません。
- ジョイン条件の一部となっているカラムのデータ型を修正することはできません。

ジョイン・インデックス内のテーブルでの挿入と削除

ジョイン・インデックスそのものではなく、基礎となるテーブルで挿入、ロード、または削除を行うことはいつでも可能です。最初にジョイン・インデックスを作成すると、以前にデータをテーブルにロードしていてもいなくても、ジョインされたテーブルを Sybase IQ が自動的に同期します。

既存のジョイン・インデックスを形成しているテーブルで挿入や削除を行う場合は、ジョイン階層のトップ・テーブルを更新しないかぎり、ジョイン・インデックスを明示的に同期する必要があります。ローを挿入し、その後同期を実行する前にそのローを削除すると、Sybase IQ は同期を最適化して挿入を省略します。

ジョイン・テーブル内のデータを操作するのに必要な権限

ジョイン・インデックスを作成する管理者は、そのジョインに対する操作を実行するのに必要な権限を持っています。ただし、グループの別のユーザがジョイン内のテーブルを操作できるようにするには、基本となる「ジョイン仮想テーブル」に対するパーミッションをそのユーザに明示的に付与する必要があります。ジョインに含まれるテーブルに対する適切な権限だけでなく、ジョイン仮想テーブルに対する権限も付与しなければなりません。

ジョイン仮想テーブルに対する権限を付与する前に、そのテーブルがシステム・テーブルにどのような名前で格納されているかを調べる必要があります。ジョイン・インデックスの名前が `emp_location` の場合は、次のクエリを使用すると、ジョイン仮想テーブルの名前が返されます。

```
select table_name from sys.systable
  where table_id in (select jvt_id from
                    sys.sysiqjoinindex
                    where joinindex_name='emp_location')
```

このクエリで返されたジョイン仮想テーブルの名前 `jvt_name` を使用して、このジョイン仮想テーブルに対するパーミッションを付与します。

```
grant all on jvt_name to user_names
```

基盤となるジョイン仮想テーブルに対する必要な権限を付与した後は、グループの他のユーザが、パーミッション・エラーを受け取らずにジョイン・インデックス内のテーブルに対して操作を実行できます。

テーブルのバージョン管理によるジョイン・インデックスへのアクセス制御

いずれのテーブルも、複数ユーザが同時に書き込むことはできません。これは、ジョイン・インデックスにおいて、あるユーザがジョイン・インデックス内のいずれかのテーブルを更新しているときには、他のユーザはそのインデックス内のテーブルを一切更新できないことを意味しています。最初のユーザのトランザクションがコミットされ、`SYNCHRONIZE` コマンドでテーブルが同期されるまでは、ジョインされたテーブルはすべて使用不可のままになります。

ジョイン・インデックスのテーブルの使用時、他のユーザには次のエラーが表示されます。

```
Cannot write to this table in current transaction.  
Another user has write mode access.
```

現在のトランザクションでは、そのジョイン・インデックスのテーブルに書き込むことはできません。書き込みを行うには、新しいトランザクションを開始する必要があります。

バージョン管理の詳細については、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

ジョイン・インデックスのサイズとメリットの見積もり

ジョイン・インデックスを作成する前に、そのサイズと潜在的なメリットを見積もる必要があります。

sp_iqestjoin によるジョイン・インデックス・サイズの見積もり

Sybase IQ では、ジョイン・インデックスのサイズを見積もるためのストアード・プロシージャ `sp_iqestjoin` が使用できます。

ジョインされるテーブルの各ペアについて、このプロシージャを実行してください。プロシージャを実行するときは、そのつど次のパラメータを入力します。

- 最初にジョインするテーブルの名前
- 最初のテーブルのローの数
- 2 つめにジョインするテーブルの名前
- 2 つめのテーブルのローの数
- 関係 (デフォルトは 1 対多)
- IQ ページ・サイズ (デフォルトは 131072 バイトまたは 128KB)

ジョイン・インデックスのサイズに影響する要因は数多くありますが、特にジョイン・インデックスに含まれる外部ジョインの数が大きく影響します。そのため、プロシージャから出される結果には 3 つの種類があります。テーブルを常に 1 対 1 でジョインすることがわかっている場合は、“Min Case index_size”を使用します。1 対多のジョインをときどき使用することが予想される場合は、“Avg Case index_size”を使用します。1 対多のジョインを多数使用することが予想される場合は、“Max Case index_size”を使用します。

これらの計算で示されるのは、ジョイン・インデックスに必要となるディスク領域の目安です。計算結果にはセグメントのバイト数と、ブロックの数が含まれます。このプロシージャでは、ジョイン・インデックスの作成にかかる時間もわかります。

既存のジョイン・インデックスの正確なサイズを知りたい場合は、別のストア・プロシージャ `sp_iqjoinindexsize` を使用します。

すべてのストア・プロシージャの構文については、『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』の「[第 7 章 システム・プロシージャ](#)」を参照してください。

ロー数を比較したジョイン・インデックスのメリットの見積もり

ジョイン・インデックスの作成を検討するときは、トップ・テーブル内のロー数と関連テーブル内のロー数を比較します。一般的には、トップ・テーブル内のロー数と関連テーブル内のロー数の比率が 10 対 1 未満のクエリでは、ジョイン・インデックスによってパフォーマンスが向上します。この比率が 100 対 1 と大きくても、ジョイン・インデックスのメリットが得られる場合もあります。逆に、比率が 10 対 1 と小さくても、クエリ・パフォーマンスが低下する場合があります。大きな比率で使用することを検討している場合は、テストを行って、クエリの効率化に役立つかどうかを確認してください。

この章について

この章では、データベースにデータを入出力する各種の方法、および各方法をいつ使用するとよいかについて説明します。また、他のタイプのデータベースから挿入されたデータを変換するときの問題点についても説明します。

内容

トピック名	ページ
インポートとエクスポートの概要	265
データベースからのデータのエクスポート	269
LOAD TABLE 文を使ったデータのバルク・ロード	279
BINARY ロード形式の使用	291
INSERT 文の使用	298
指定した値のロー単位の挿入	298
データベースからの指定したローの挿入	299
データの対話的インポート	302
エンディアン・フォーマットが異なるシステムへのデータの移動	303
ジョイン・インデックスのテーブルへの挿入	305
プライマリ・キー・カラムと外部キー・カラム	306
ラージ・オブジェクト・データの抽出	306
挿入時のデータ変換	306
データ表示に関連する他の項目	321
Adaptive Server Enterprise のデータ型の対応	321
データのバルク・ロードのチューニング	326
UPDATE によるデータの更新	330
データベースの削除	331

インポートとエクスポートの概要

Sybase IQ では、フラット・ファイルまたは直接データベース・テーブルからデータをインポートできます。データベースに指定した値を直接入力できます。他のフォーマットへのデータのエクスポートは、Interactive SQL ユーティリティと IQ データ抽出機能から実行できます。

Sybase IQ テーブルは論理テーブルで、データは格納されていません。データなど、クエリを解決するために必要なすべての情報は、Sybase IQ のインデックスに格納されています。データを IQ テーブルのカラムに挿入しても、実際はデータはテーブルのカラムではなく、カラムのインデックスに追加されません。インデックスを作成するには、テーブル単位でデータを挿入します。

インポートとエクスポートの方法

Sybase IQ では、データを追加、変更、または削除する方法を選択できます。

- フラット・ファイルからテーブルを効率的にバルク・ロードするには、SQL 文 `LOAD TABLE` を使用します。
- 指定した値をテーブルにロー単位で挿入するには、`VALUES` オプションを指定した SQL 文 `INSERT` を使用します。
- テーブルから選択したローを挿入するには、`SELECT` 句を指定した SQL 文 `INSERT` を使用します。
- 他のデータベース内にあるテーブルからローを挿入するには、`LOCATION` 句を指定した SQL 文 `INSERT` を使用します。
- テーブルから特定のローを削除するには、`DELETE` 文を使用します。`TRUNCATE` 文を使用すると、テーブルが 0 ローに初期化されます。
- テーブルにある既存のローを変更するには、`UPDATE` 文を使用します。

IQ データ抽出機能では、データがバイナリまたは ASCII フォーマットでエクスポートされ、それを他のデータベースにロードできます。この機能は、大量のデータを移動する場合や、ロードに使用できる出力ファイルが必要な場合に使用します。この機能の詳細とメリットについては、「[データ抽出オプション](#)」(270 ページ) を参照してください。

Interactive SQL からは、データをさまざまなフォーマットで別のデータベースにエクスポートしたり、テキスト・ファイルを出力したりできます。フォーマットとその選択方法については、次の項を参照してください。コマンドの出力のリダイレクトもできます。

注意 Sybase IQ では、LOAD TABLE FORMAT BCP オプションによって BCP をサポートしています。また、LOAD TABLE USING CLIENT FILE オプションを使用したりモート・データのバルク・ロードもサポートしています。詳細については、「[クライアントからの直接のデータ・ロード](#)」(286 ページ)、および『リファレンス：文とオプション』の「[第1章 SQL 文](#)」の「[LOAD TABLE 文](#)」を参照してください。

SQL Anywhere テーブルへの BCP を実行してから、内容を Sybase IQ に転送できますが、ローの転送は一度に 1 ローずつ行われます。Sybase IQ では、Open Client の Bulk-Library (bklbib) ルーチンをサポートしていません。そのため、Open Client の Bulk-Library を使用する BCP は、バルク・ロード・モードでは機能しません (Bulk-Library/C は、Client-Library アプリケーションと Server-Library アプリケーションが、高速データ転送用に Adaptive Server のバルク・コピー・インタフェースを使用できるようにするルーチンを提供します)。Sybase IQ と Adaptive Server の BCP フォーマットでは、日付に含まれる 1 桁の値の前にある先行ブランクをサポートしています。

データの入出力フォーマット

LOAD TABLE 文では、ファイルからロー単位でデータをインポートします。固定長フィールド、または可変長フィールドの ASCII 入力ファイルとバイナリ入力ファイルの両方がサポートされます。可変長フィールドは、フィールドの末尾がデリミタによって指定されます。

INSERT 文は、指定した一連の値から、または直接テーブルから、データを Sybase IQ テーブルに移動します。

Interactive SQL は次の出力ファイル・フォーマットをサポートします。

ファイル・フォーマット	説明
ASCII	1 行に 1 ローを持ち、値がデリミタで区切られたテキスト・ファイル。文字列値は状況によっては一重引用符で囲まれる。これは、LOAD TABLE で使われるフォーマットと同じ
DBASEII	DBASE II フォーマット
DBASEIII	DBASE III フォーマット
EXCEL	Excel フォーマット
FIXED	各カラム幅がカラム・タイプで定義した幅か、パラメータで指定された幅で、固定フォーマットのデータ・レコード
FOXPRO	FoxPro フォーマット

ファイル・フォーマット	説明
HTML	HTML フォーマット
LOTUS	Lotus ワークシート・フォーマット
SQL	テーブル内の情報を再作成するのに必要な Interactive SQL INPUT 文
XML	UTF-8 でコード化され、埋め込み DTD を含む XML フォーマット

IQ データ抽出機能では、データはバイナリまたは ASCII フォーマットでエクスポートされます。データ抽出の出力フォーマットの詳細については、「[データ抽出オプション](#)」(270 ページ)を参照してください。

❖ Interactive SQL の出力フォーマットの指定

- 1 Interactive SQL のメニュー・バーから [コマンド] - [オプション] を選択します。
- 2 [出力フォーマット] を選択します。
- 3 ここで選択した設定をデフォルトの出力フォーマットにするには、[恒久的な設定を行う] をクリックします。

データ修正のパーミッション

修正するデータベース・テーブルに適切なパーミッションがある場合のみ、データの修正文を実行できます。データベースの管理者とデータベース・オブジェクトの所有者は、grant 文と revoke 文を使用して、だれがどのデータ修正機能にアクセスするかを決定します。

データを挿入するには、テーブルまたはビューの INSERT パーミッションが必要となります。データを削除するには、テーブルまたはビューの DELETE パーミッションが必要です。データを更新するには、UPDATE パーミッションが必要です。DBA は、テーブルへのデータの挿入や削除を行うことができます。テーブルの所有者には、該当するテーブルの INSERT、DELETE、UPDATE のパーミッションが付与されています。

パーミッションを個々のユーザ、グループ、または public グループに付与したり、取り消したりできます。パーミッションの詳細については、「[第 8 章 ユーザ ID とパーミッションの管理](#)」を参照してください。

データベース更新のスケジュール

1 人のユーザがテーブルにデータを挿入している間も、複数のユーザがデータベース・テーブルに問い合わせできます。複数のユーザが異なるデータベースに対してデータの挿入や削除を行っている場合は、データベースを同時に更新できます。

データベースの更新中の同時使用が許可されている場合は、パフォーマンスとディスクの使用率が低下します。Sybase IQ での同時実行性に関する問題の処理方法については、「[第10章 トランザクションとバージョン管理](#)」を参照してください。ロード・パフォーマンスを改善するその他の方法については、「[データのバルク・ロードのチューニング](#)」(326 ページ)を参照してください。

データベースからのデータのエクスポート

この項では、Sybase IQ データベースからデータをエクスポートする方法について説明します。

注意 現在のバージョンの Sybase IQ のデータベースから IQ データをエクスポートする場合は、この章で説明する方法を使用することをおすすめします。独自の、またはサード・パーティ製のフロント・エンド・ツールを使用して、効率よく IQ データベースに問い合わせ希望どおりにフォーマットしたデータをエクスポートすることもできます。

システム・テーブル以外のテーブルをカタログ・ストアからエクスポートする必要がある場合には、この章で説明する方法に従ってください。データのアンロード方法については、『SQL Anywhere サーバー SQL リファレンス』を参照してください。

出力先リダイレクションの使用

出力リダイレクションを使用して、クエリ結果をエクスポートできます。

リダイレクション記号の `>#` とともに出力先を指定してコマンドを実行することにより、コマンドの出力をファイルまたはデバイスにリダイレクトできます。リダイレクション記号の後にはファイル名が続きます。コマンド・ファイル内では、ファイル名の後に置かれたセミコロンが文のターミネータとして使用されます。ファイルは、Interactive SQL が起動したディレクトリからの相対位置に配置されます。

この例では、出力はファイル *empfile* にリダイレクトされます。

```
SELECT *  
FROM Employees  
># empfile
```

ファイル名を引用符で囲まないようにします。

出力リダイレクションは、SELECT 文の中で特に有効です。

ファイルの内容を置き換える代わりに、出力を指定したファイルに付加するには、1つのリダイレクション記号に > を1つではなく2つ使用します (>># など)。指定したファイルの最初から出力を開始する場合、見出しをサポートする出力フォーマットでは、SELECT 文の結果に見出しが付加されます。

出力とメッセージのリダイレクト

リダイレクション記号の >& を使うと、エラー・メッセージと、エラーが発生したコマンドに関する統計を含むすべての出力がリダイレクトされます。例を示します。

```
SELECT *
FROM Employees
>& empfile
```

ファイル名を引用符で囲まないようにします。

この例は、ファイル *empfile* に SELECT 文を出力し、さらにその SELECT 文からの出力と関係する統計を出力します。

>& を使ったリダイレクションは、READ コマンドの動作やエラーのログを作成するのに役立ちます。各コマンドの統計値とエラーは、出力ファイルの中でコマンドに続いて書き出されます。

NULL 値の出力

データを書き出す理由のうち、最も多いのは他のソフトウェアで使うためです。ただし、他のソフトウェアでは NULL 値を適切に処理できない場合があります。

dbisql オプションの NULLS を使用すると、NULL 値を出力する方法を選択できます。または、IFNULL 関数を使用して、NULL 値が出てくるたびに指定した値を出力できます。

dbisql オプションの設定については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」を参照してください。

データ抽出オプション

データ抽出機能により、サイズの大きい結果セットを生成するクエリのパフォーマンスが大幅に向上します。現在この機能は、SET OPTION コマンドを使用して設定する一連のデータベース・オプションから構成されています。他のデータベース・オプションと同様に、データ抽出オプションはテンポラリー・オプションまたは永久オプションとして設定できます。これらのオプションは通常はテンポラリーとして設定されます。抽出オプションは接続に対して設定されます。

抽出オプションを使用すると、SELECT 文の出力先を標準インタフェースからリダイレクトし、1つまたは複数のディスク・ファイルまたは名前付きパイプに直接出力できます。抽出オプションを使用する利点として、次の2点が挙げられます。

- ・ バイナリ・フォーマットがサポートされています。したがって、出力データを同一または異なる IQ データベースにロードできます。

- SELECT 文の出力が大きい場合に、ASCII 出力では最高 4 倍、バイナリ出力では最高 9 倍速く処理できます。

抽出オプション

抽出処理を制御するオプションは 27 種類あります。次の表は、各オプションの指定可能な値とデフォルト値です。

オプション名	指定できる値	デフォルト値
Temp_Extract_Append	ON、OFF	OFF
Temp_Extract_Binary	ON、OFF	OFF
Temp_Extract_Column_Delimiter	文字列	' '
Temp_Extract_Directory	文字列	"
Temp_Extract_Name1	文字列	"
Temp_Extract_Name2	文字列	"
Temp_Extract_Name3	文字列	"
Temp_Extract_Name4	文字列	"
Temp_Extract_Name5	文字列	"
Temp_Extract_Name6	文字列	"
Temp_Extract_Name7	文字列	"
Temp_Extract_Name8	文字列	"
Temp_Extract_Null_As_Empty	ON、OFF	OFF
Temp_Extract_Null_As_Zero	ON、OFF	OFF
Temp_Extract_Quote	文字列	"
Temp_Extract_Quotes	ON、OFF	OFF
Temp_Extract_Quotes_All	ON、OFF	OFF
Temp_Extract_Row_Delimiter	文字列	"
Temp_Extract_Size1	プラットフォーム固有 *	0
Temp_Extract_Size2	プラットフォーム固有 *	0
Temp_Extract_Size3	プラットフォーム固有 *	0
Temp_Extract_Size4	プラットフォーム固有 *	0
Temp_Extract_Size5	プラットフォーム固有 *	0
Temp_Extract_Size6	プラットフォーム固有 *	0
Temp_Extract_Size7	プラットフォーム固有 *	0
Temp_Extract_Size8	プラットフォーム固有 *	0
Temp_Extract_Swap	ON、OFF	OFF

*Temp_Extract_Sizen オプションのデフォルト値はプラットフォーム固有です。これらの値は次のとおりです。

- AIX と HP-UX : 0 ~ 64GB
- Sun Solaris : 0 ~ 512GB
- Windows の場合 : 0 ~ 128GB
- Linux : 0 ~ 512GB

JFS2 など、大規模なファイル・システムでデフォルト値よりも大きなファイル・サイズをサポートしている場合、TEMP_EXTRACT_SIZE_n をファイル・システムが許可する値に設定します。たとえば、ITB をサポートするには、次のように設定します。

```
TEMP_EXTRACT_SIZE1 = 1073741824 KB
```

注意 整数値を指定できるデータベース・オプションでは、小数の *option-value* の設定が常に整数値にトランケートされます。たとえば、値 3.8 は 3 にトランケートされます。

この中で最も重要なオプションは TEMP_EXTRACT_NAME1 です。TEMP_EXTRACT_NAME1 がデフォルト値 (空の文字列) に設定されている場合、抽出処理は使用できず、リダイレクトされる出力もありません。抽出処理を使用可能にするには、Temp_Extract_Name1 を有効なパス名に設定します。すると、そのパス名のファイルに対して抽出が始まります。他で使用されていないファイルのパス名を選択してください。存在しないファイルを指定した場合、データ抽出機能がそのファイルを作成します。

指定されたファイルが置かれたディレクトリまたはフォルダと、指定されたファイルの両方に対する書き込みのパーミッションを、IQ の開始に使用しているユーザ (たとえば **sybase**) が持っていないてはなりません。追加モードの場合は、データ抽出機能は、抽出された行をファイルの末尾に追加します。ファイルに既に書き込まれているデータへは上書きしません。ファイルが存在しない場合は、データ抽出機能がファイルを作成します。

警告! 既存のファイルのパス名を指定し、TEMP_EXTRACT_APPEND オプションが OFF (デフォルト) に設定されている場合は、ファイルの内容が上書きされます。これは、ウィークリー・レポートなどのファイルであれば問題ありませんが、データベース・ファイルに対しては行わないでください。

TEMP_EXTRACT_NAME2 ~ TEMP_EXTRACT_NAME8 の各オプションは、TEMP_EXTRACT_NAME1 に追加して、複数の出力ファイルを指定するために使用されます。これらのオプションは順番に使用する必要があります。たとえば、TEMP_EXTRACT_NAME1 および TEMP_EXTRACT_NAME2 がすでに設定されていないかぎり、TEMP_EXTRACT_NAME3 を設定しても無効となります。

TEMP_EXTRACT_SIZE1 ~ TEMP_EXTRACT_SIZE8 の各オプションには、それぞれに対応する出力ファイルの最大サイズを指定します。TEMP_EXTRACT_SIZE1 は TEMP_EXTRACT_NAME1 で指定された出力ファイルの最大サイズ、TEMP_EXTRACT_SIZE2 は TEMP_EXTRACT_NAME2 で指定された出力ファイルの最大サイズ、というように指定します。

なお、データ抽出サイズのオプションのデフォルト値は 0 です。IQ はこのデフォルト値を、次の値に変換します。

デバイス・タイプ	Size
ディスク・ファイル	AIX と HP-UX : 0 ~ 64GB Sun Solaris と Linux : 0 ~ 512GB Windows の場合 : 0 ~ 128GB
テープ *	524288KB (0.5GB)
その他	制限なし

* テープ・デバイスは現時点ではサポートされていません。

TEMP_EXTRACT_APPEND は、TEMP_EXTRACT_SIZE_n オプションと互換性がありません。抽出追加出力ファイルのサイズを制限しようとすると、Sybase IQ からエラーが表示されます。

抽出先のディスク・ファイルまたは名前付きパイプが1つだけの場合は、TEMP_EXTRACT_NAME₂ ~ TEMP_EXTRACT_NAME₈ と TEMP_EXTRACT_SIZE₁ ~ TEMP_EXTRACT_SIZE₈ の各オプションはデフォルト値のままにします。

注意 SELECT がローを返さず、リダイレクトする出力がない場合は、長さゼロの空のファイルが作成されます。複数の抽出ファイルを指定したが、それらすべてを使用するほどデータがない、という場合でも、すべてのファイルが作成されます。

アクセスの制御

TEMP_EXTRACT_DIRECTORY オプションは、ユーザがデータ抽出機能を使用できるかどうかを制御します。また、一時抽出ファイルが配置されるディレクトリを制御し、TEMP_EXTRACT_NAME_n オプションで指定されたディレクトリ・パスは上書きされます。

ユーザの TEMP_EXTRACT_DIRECTORY オプションが FORBIDDEN (大文字と小文字を区別しない) という文字列に設定されている場合、そのユーザはデータ抽出を実行できません。このユーザがデータ抽出機能を使用しようとすると次のようなエラーになります。“You do not have permission to perform Extracts”。

PUBLIC グループの TEMP_EXTRACT_DIRECTORY が FORBIDDEN に設定されている場合は、どのユーザもデータ抽出を実行できません。

TEMP_EXTRACT_DIRECTORY が有効なディレクトリ・パスに設定されている場合、一時抽出ファイルはそのディレクトリに配置され、TEMP_EXTRACT_NAME_n オプションで指定されたパスは上書きされます。

TEMP_EXTRACT_DIRECTORY が無効なディレクトリ・パスに設定されている場合は、次のようなエラーが発生します。“Files does not exist File:<invalid path>”。

TEMP_EXTRACT_DIRECTORY が空白の場合、一時抽出ファイルは、TEMP_EXTRACT_NAME_n で指定されたディレクトリに配置されます。TEMP_EXTRACT_NAME_n でパスが指定されていない場合、抽出ファイルはデフォルトでサーバ起動ディレクトリに配置されます。

TEMP_EXTRACT_DIRECTORY オプションは、大きなデータ抽出ファイルの作成を書き込みアクセス権のあるディレクトリに制限することで、セキュリティを高め、ディスク管理を容易にします。このオプションを設定するには DBA 権限が必要です。

抽出の種類

データ抽出には、次の 3 種類があります。

- バイナリ
- バイナリ/スワップ
- ASCII

バイナリ抽出を実行すると、LOAD TABLE 文を使ってロードできるファイルが生成されます。この文では、全体的には「バイナリ」フォーマット、各カラムには「null バイト付きバイナリ」フォーマットが指定されています。

バイナリ/スワップ抽出は、逆のエンディアン別のマシンにロードするように設計されている点以外は、バイナリ抽出と同じ働きをします。

ASCII 抽出を実行するとテキスト・ファイルが生成されます。

2つのオプション Temp_Extract_Binary と Temp_Extract_Swap によって、3種類の抽出処理のうちどの抽出処理が実行されるかが決まります。

種類	Temp_Extract_Binary	Temp_Extract_Swap
バイナリ	ON	OFF
バイナリ/スワップ	ON	ON
ASCII	OFF	OFF

抽出の種類のデフォルトは ASCII です。

データ抽出機能を使用してデータをアンロードするときに、TEMP_EXTRACT_BINARY オプションを ON に設定した場合は、バイナリ・データをロードするときに、LOAD TABLE 文の BINARY WITH NULL BYTE パラメータを各カラムに対して必ず指定しなくてはなりません。

カラムとローのデリミタ

ASCII 抽出の場合、デフォルトでは、カラム値はカンマで区切られ、ローの末尾は、UNIX プラットフォームでは改行文字、Windows プラットフォームでは復帰/改行文字となっています。文字列には引用符は付きません。このデフォルト設定が適切でない場合は、次のオプションを使用してデリミタを変更してください。

- Temp_Extract_Column_Delimiter
- Temp_Extract_Row_Delimiter
- Temp_Extract_Quote

- Temp_Extract_Quotes
- Temp_Extract_Quotes_All

デリミタには1バイトから最大4バイト必要で、マルチバイト照合順を使用している場合はその照合順でデリミタが有効である必要があります。デリミタとして使用する文字列は、データ出力文字列内で使われないものにしてください。

なお、Temp_Extract_Row_Delimiter オプションのデフォルトは空文字列です。IQ は、このオプションのデフォルトの空文字列を、UNIX プラットフォームでは改行文字、Windows プラットフォームでは復帰/改行文字に変換します。

Temp_Extract_Column_Delimiter オプションは、カラム間のデリミタを制御します。ASCII 抽出で、このオプションが空文字列 " に設定されている場合は、抽出されたデータは、カラム・デリミタなしの固定幅の ASCII で書き込まれます。数値とバイナリ・データ型は、 n ブランクのフィールドに右揃えされます。ここで n は、その型の値を表すのに必要な最大バイト数を表します。文字データ型は、 n ブランクのフィールドで左揃えされます。

注意 固定幅 ASCII 抽出での最小のカラム幅は4バイトです。NULL 値に対して文字列 "NULL" を設定できるようにするためです。たとえば、抽出されたカラムが CHAR(2) で、Temp_Extract_Column_Delimiter が空の文字列 " に設定されている場合、抽出されたデータの後に2つのスペースが置かれます。

ASCII 抽出の間、引用符の使用は次のオプションで制御されます。

オプション	ASCII 抽出での動作
Temp_Extract_Quotes	文字列フィールドを引用符で囲む
Temp_Extract_Quotes_All	すべてのフィールドを引用符で囲む
Temp_Extract_Quote	引用符として使用する文字列を指定する

Temp_Extract_Quote オプションで指定する引用符文字列には、デリミタと同じ制限があります。このオプションのデフォルトは空文字列で、IQ はこれを一重引用符に変換します。

null 値の表現

Temp_Extract_Null_As_Zero オプションと Temp_Extract_Null_As_Empty オプションは、ASCII 抽出での NULL 値の表現を制御します。Temp_Extract_Null_As_Zero オプションが ON に設定されている場合、NULL 値は次のように表現されます。

- 算術型には '0'
- CHAR 文字型と VARCHAR 文字型には " (空文字列)
- 日付には " (空文字列)
- 時刻には " (空文字列)
- タイムスタンプには " (空文字列)

Temp_Extract_Null_As_Empty オプションが ON に設定されている場合は、すべてのデータ型に対して、null 値は "(空文字列)" で表現されます。

上記の引用符は、抽出出力ファイルには含まれないため注意してください。Temp_Extract_Null_As_Zero オプションと Temp_Extract_Null_As_Empty オプションが OFF に設定されている場合は、すべての状況において、NULL 値の表現には文字列 'NULL' が使用されます。デフォルトは OFF です。

Temp_Extract_Null_As_Zero が ON に設定されている場合は、ASCII 抽出で CHAR または VARCHAR カラムに対してファイルに書き込まれる文字数は、そのカラムの文字数に等しくなります。4 文字未満のカラムの場合でも同じです。以前のリリースでは、文字列 'NULL' を格納できるよう、Sybase IQ は最低 4 文字を必ず返していました。

メッセージ・ロギング

Query_Plan オプションが ON の場合、抽出されたカラムのタイムスタンプ付きのリストが IQ メッセージ・ログに出力されます

データ抽出機能の有効化

データ抽出オプションの使用には注意が必要です。

警告！ 抽出オプションを設定して SELECT 文を 2 回実行する場合、2 回目のときに抽出ファイル名を変更しないと、1 回目の実行時に出力したファイルが 2 回目の実行時に上書きされてしまいます。たとえ 1 秒後であっても 1 週間後であっても、SELECT 文を再度実行するたびに、抽出は先頭からやり直されます。ただし、Temp_Extract_Append オプションを ON に設定している場合は除きます。

また、抽出オプションは接続に対して設定される点にも注意してください。抽出オプションを設定した場合に、他のユーザが同じ接続を使用してデータベースに接続すると、そのユーザに対してもデータ抽出機能が有効化されてしまいます。自分が抽出した出力が、同じ接続を使用する他のユーザによって上書きされる可能性があります。

また、別のユーザが同じユーザ ID でログインした場合、オプションを無効にするまでは、このユーザが実行したクエリの出力は、同じ抽出ファイルに書き込まれます。抽出機能を使用する場合は、ユニークなユーザ ID から要求を実行してください。

❖ データ抽出オプションの有効化

- 1 前回の抽出出力ファイルを残しておきたい場合は、そのファイルに別の名前を付けて保存します。
- 2 以前に使用した抽出出力ファイルを削除します。
- 3 必要な抽出オプションを設定します。Temp_Extract_Name1 に出力先ファイル・パスが設定されているか確認してください。
`SET [TEMPORARY] OPTION option-name = option-value`
- 4 SELECT 文を発行して必要なデータを抽出します。

- 5 抽出処理が終了したら、Temp_Extract_Name1 を空の文字列にリセットするか、一時的な設定の場合は接続を切断します。

例

単一のディスク・ファイルへの抽出 次の文は、単一のディスク・ファイル *daily_report.txt* への抽出を実行します。

```
SET TEMPORARY OPTION Temp_Extract_Name1 = 'daily_report.txt';
SET TEMPORARY OPTION Temp_Extract_Name2 = '';
SELECT ....;
SET TEMPORARY OPTION Temp_Extract_Name1 = '';
```

ここでは、**SELECT** 文の実行前に Temp_Extract_Name2 を空の文字列に設定しています。1つのファイルのみに出力されるようにするためです。

また、**SELECT** 文の後で Temp_Extract_Name1 を空の文字列に設定しています。抽出を無効にするためです。抽出を無効にしないと、次の **SELECT** 文の実行時に *daily_report.txt* ファイルが上書きされます。

追加モードでの抽出 この例では、ディスク出力ファイル *hourly_report.txt* はすでに作成されており、ユーザ **sybase** に対して書き込みパーミッションが設定されています。次の文は、*hourly_report.txt* への抽出を実行し、各 **SELECT** 文の出力をファイルの末尾に追加していきます。

```
SET TEMPORARY OPTION Temp_Extract_Append = ON;
SET TEMPORARY OPTION Temp_Extract_Name1 = 'hourly_report.txt';
SET TEMPORARY OPTION Temp_Extract_Name2 = '';
SELECT ....;
SELECT ....;
SELECT ....;
SET TEMPORARY OPTION Temp_Extract_Name1 = '';
```

3つの **SELECT** 文の出力はいずれも *hourly_report.txt* ファイルに書き込まれます。最後の **SELECT** 文の後で Temp_Extract_Name1 を空の文字列に設定しています。抽出を無効にするためです。抽出を無効にしないと、次に実行する **SELECT** 文の出力も、*hourly_report.txt* ファイルの末尾に追加されます。

複数のディスク・ファイルへの抽出 次の文は、*file1.out*、*file2.out*、*file3.out* の各ディスク・ファイルへの抽出を実行します。

まず、ファイル名のオプションを設定します。

```
SET TEMPORARY OPTION Temp_Extract_Name1 = 'file1.out';
SET TEMPORARY OPTION Temp_Extract_Name2 = 'file2.out';
SET TEMPORARY OPTION Temp_Extract_Name3 = 'file3.out';
SET TEMPORARY OPTION Temp_Extract_Name4 = '';
```

次にファイルのサイズを、たとえば各 1MB に制限します。それには、対応する抽出サイズ・オプションを設定します。

```
SET TEMPORARY OPTION Temp_Extract_Size1 = '1024';
SET TEMPORARY OPTION Temp_Extract_Size2 = '1024';
SET TEMPORARY OPTION Temp_Extract_Size3 = '1024';
```

サイズ・オプションは KB (1024 バイト) 単位です。

これらの設定に基づいて、抽出の出力はまず *file1.out* に書き込まれます。そして、次のローを書き込むと *file1.out* のファイル・サイズが 1MB を超えるという時点で、出力は *file2.out* にリダイレクトされます。そして、*file2.out* も満杯 (次のローを書き込むと *file2.out* のファイル・サイズが 1MB を超える) という時点で、出力は *file3.out* にリダイレクトされます。IQ がまだ抽出していないローが残っている段階で、*file3.out* のサイズが 1MB を超えた場合、エラーが通知されます。

抽出処理の制約事項

データ抽出機能には、次の制限事項が適用されます。

- 抽出は、IQ ストアに格納されたデータに対してのみ実行できます。
- 抽出は、システム・テーブルやデータベース間ジョインに対しては実行できません。
- 抽出は、ユーザ定義関数やシステム関数を使用するクエリに対しては実行できません。ただし、システム関数 `suser_id()` と `suser_name()` は例外です。
- バイナリ LOAD TABLE は、常に VARCHAR データから空白を削除します。VARCHAR データに後続ブランクがある場合は、バイナリ・ロードによる挿入処理によってそのブランクがなくなります。
- VARBINARY データについては、抽出時に後続のゼロが埋め込まれます。たとえば、`varbinary(6)` として宣言されたフィールドに `0x1234` というデータが入っている場合、抽出時にゼロが埋め込まれ、抽出後には `0x123400` としてロードされます。
- バイナリ・フォーマットは今後のリリースで変更される予定です。
- 浮動小数点データを正確に再現する必要がある場合は、バイナリ・オプションを使用してください。
- 現在、テープ・デバイスはサポートしていません。
- `-q` (クワイエット・モード) オプションを指定して `dbisql` (Interactive SQL) を実行するとき、データ抽出コマンドがコマンド・ファイル内にある場合は、`dbisql` の [複数の結果セットを表示] オプションを設定し、これを永続的なものにしておく必要があります。このオプションが設定されていない場合、出力ファイルは作成されません。

[複数の結果セットを表示] オプションを設定するには、`dbisql` ウィンドウの [ツール]-[オプション] を選択し、[複数の結果セットを表示] ボックスをクリックして [永続化する] をクリックします。

また、`Temp_Extract_Name1` を設定した場合には、次の操作は実行できない点を留意しておいてください。

- ジョインのトップ・テーブルに対する LOAD、DELETE、INSERT または INSERT...LOCATION
- SYNCHRONIZE JOIN INDEX (明示的に発行した場合、または CREATE JOIN INDEX の処理過程で実行された場合)
- INSERT...SELECT

抽出とイベント

イベントは、結果セットを返す文の実行をサポートしません。サーバ・ログは、次のようなエラーを返します。

```
Handler for event 'test_ev' caused SQLSTATE '09W03'
Result set not permitted in 'test_ev'
```

イベントによってクエリを実行するには、ストアド・プロシージャを呼び出すイベントを作成し、そのストアド・プロシージャの結果をテンポラリ・テーブルに挿入します。抽出を使用した場合、テンポラリ・テーブルは常に空となり、オーバヘッドはわずかしか必要ありません。

次に例を示します。

```
CREATE PROCEDURE procl()
BEGIN
    SET TEMPORARY OPTION temp_extract_name1 = 'testproc.out';
    SELECT * FROM iq_table;
END;

CREATE EVENT "test_ev" ENABLE HANDLER
BEGIN
    SELECT * INTO #tmp FROM procl();
END;

TRIGGER EVENT test_ev;
```

名前付きパイプによる抽出

Sybase IQ では、ノンブロッキング・モードで名前付きパイプを開きます。つまり、名前付きパイプから抽出出力を読み取るプロセスは、パイプからデータがなくなるまで読み取ると、パイプを閉じます。Sybase IQ は、データがなくなるまで名前付きパイプに出力データを送信します。UNIX システムおよび Linux システムの場合、Sybase IQ はそれ以上出力するデータがなくなると、名前付きパイプを閉じます。Windows システムの場合、Sybase IQ は出力データをすべて送信し、データを読み取るプロセスが読み取りを終了するのを待ってから、名前付きパイプを閉じます。これらの名前付きパイプを読み取るために使用するアプリケーションを適宜設計する必要があります。

LOAD TABLE 文を使ったデータのバルク・ロード

LOAD TABLE 文を使用すると、テキスト・ファイルまたはバイナリ・ファイルから既存のデータベース・テーブルに効率よくデータをインポートできます。この文では、自動的に作成されたかまたはユーザによって定義されたカラム・インデックスにデータをロードします。

LOAD TABLE 文の実行に必要なパーミッションは、サーバのコマンド・ラインで `-gl` オプションを使用して設定します。`-gl all` 設定の使用をおすすめします。この設定では、任意のユーザがテーブルをロードおよびアンロードできます。これは、`start iq` により設定されるデフォルトの設定です。`-gl all` を設定した場合、LOAD TABLE 文を使用するためには、テーブルの所有者であるか、DBA 権限または ALTER パーミッションを持っていないことはありません。また、テーブルに対する INSERT パーミッションも必要です。

ロード中の入力ファイル・エラー処理については、『リファレンス：文とオプション』の「LOAD TABLE 文」の ON FILE ERROR ロード・オプションの説明を参照してください。

LOAD TABLE は、ラージ・オブジェクト (LOB) データのロードをサポートしています。詳細については、Sybase IQ の非構造化データ分析の概要を参照してください。

データのロードのための
コマンド・ファイル
の使用

大量のデータをロードする場合、通常はコマンド・ファイルを作成します。コマンド・ファイルを作成するには、『ユーティリティ・ガイド』の「[第 2 章 Interactive SQL \(dbisql\) の使用](#)」の順に従います。

トランザクション処理と
LOAD TABLE

IQ テーブルに対して LOAD TABLE 文を発行すると、データがロードされる前に自動的にセーブポイントが設定されます。ロードの完了に成功すると、Sybase IQ はセーブポイントを解放します。ロードに失敗した場合は、セーブポイントまでトランザクションがロールバックします。この方法によって、トランザクションを柔軟にコミットできます。たとえば、2 つの LOAD TABLE コマンドを発行する場合は、両方のコマンドをコミットしたり、どちらもコミットしないようにできます。

カタログ・ストア・テーブルに対して LOAD TABLE を発行する場合は、自動セーブポイントは設定されません。ロードに成功すると、自動的にコミットします。ロードに失敗すると、ロールバックします。ロードに成功したカタログ・ストア・テーブルはロールバックできません。

トランザクション処理の詳細については、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

整合性制約と LOAD
TABLE

LOAD TABLE では、整合性制約違反があった場合のロード動作を制御したり、違反についての情報をログに適宜記録したりすることができます。ロード時に発生する UNIQUE、NULL、DATA VALUE、FOREIGN KEY の制約違反を無視するかどうかや、違反が最大何件に達するまではロールバックを開始せずに無視するかを指定できます。また、特定の種類の整合性制約違反についての情報を、違反単位とロー単位の両方でロギングすることもできます。メッセージ・ログでは違反単位、ロー・ログではロー単位です。

メッセージ・ログとロー・ログの内容とフォーマットについては、「[整合性制約違反のロギング](#)」(288 ページ)を参照してください。

LOAD TABLE の簡単な例

次の文は、ファイル *dept.txt* から **Departments** テーブルのすべてのカラムにデータをロードします。この例では、明示的にデータを変換する必要はなく、入力カラムの幅が **Departments** テーブルのカラムの幅と一致することが前提となります。指定された場所にフラット・ファイル *dept.txt* が存在している必要があります。

```
LOAD TABLE Departments
  ( DepartmentID, DepartmentName, DepartmentHeadID )
FROM '/d1/MILL1/dept.txt'
```

ロードするファイルの指定

データのロード元のファイルを1つ以上指定します。**FROM** 句で、各 *filename-string* を指定し、複数の文字列をカンマで区切ります。

ファイルは1つずつ読み込まれ、**FROM** 句で指定された順に左から右へ処理されます。**SKIP** または **LIMIT** の値は、ロードの開始時に適用され、各ファイルには適用されません。

メモリが不足しているなどの理由でロードを完了できない場合は、ロード・トランザクション全体がロールバックします。

filename-string *filename-string* は、文字列としてサーバに渡されます。したがって、文字列は他の SQL 文字列と同じフォーマット要件に従います。特に、次のような場合です。

- 円記号 (¥) が文字 *n*、*x*、または *¥* の前に付いている場合は、エスケープ文字とみなされます。このため、Windows システムでディレクトリ・パスを指定する場合に、円記号の次に *n*、*x*、または *¥* が続く場合は、円記号を2つ続けて、1つの円記号を表す必要があります。たえず円記号を2つ続けると安全です。したがって、ファイル *c:¥newinput.dat* から **employees** テーブルにデータをロードする文は、次のようになります。

```
LOAD TABLE employees
FROM 'c:¥¥newinput.dat' ...
```

- サーバ側のロード (**LOAD TABLE ... USING FILE**) の場合、パス名はクライアント・アプリケーションではなく、データベース・サーバを起点とします。別のコンピュータのデータベース・サーバ上で文を実行している場合、ディレクトリ名はそのサーバ・マシンのディレクトリを参照し、クライアント・マシンのディレクトリは参照しません。ロード用の入力ファイルはサーバ・マシン上になければなりません。
- クライアント側のデータのロード (**LOAD TABLE ... USING CLIENT FILE**) の場合、パス名はクライアント・アプリケーションを起点とする必要があります。ディレクトリ名は、クライアント・マシンのディレクトリを参照します。

名前付きパイプ

ファイルは名前付きパイプを使用して指定できます。Windows で名前付きパイプ (または FIFO) を使用してロードする場合は、パイプを記述したプログラムで特殊な方法を使用して、パイプを閉じます。パイプ・ライタでは、FlushFileBuffers() を呼び出し、その後に DisconnectNamedPipe() を呼び出す必要があります (プログラムによって呼び出されない場合、Sybase IQ は hos_io::Read() から例外を報告します)。これによって PIPE_NOT_CONNECTED エラーが発生します。このエラーによって、パイプが停止したのは手順どおりの方法によってであり、制御不可能な接続解除状態になったためではないことが Sybase IQ に通知されます。呼び出しの詳細については、Microsoft のマニュアルを参照してください。

テーブル全体のフォーマット・オプションの指定

入力データのフォーマットを記述するオプションを複数指定できます。

FORMAT オプション *column-spec* を省略した場合に適用される、テーブルのカラム用のデフォルト・フォーマットを指定できます。*column-spec* に記述できるフォーマットと同じフォーマットをここに記述できます。FORMAT ロード・オプションも省略した場合は、ファイルはバイナリと解釈されます。

FORMAT BINARY および BINARY カラム指定句を使用した LOAD TABLE 文で読み込むことができるデータ・ファイルを生成するために、Sybase IQ で使用されるバイナリ・フォーマットの詳細については、「[BINARY ロード形式の使用](#)」(291 ページ) を参照してください。

DELIMITED BY オプション *column-spec* 定義でカラムのデリミタを省略した場合は、デフォルトのカラム・デリミタ文字であるカンマが使用されます。別のカラム・デリミタを指定するには、単一の ASCII 文字または 16 進の文字表現を指定します。特に、タブデリミタ値を指定するには、次のように、タブ文字 (9) の 16 進 ASCII コードを使用します。

```
...DELIMITED BY '¥x09' ...
```

デリミタとして改行文字を使用する場合は、特殊文字の組み合わせである '¥n' またはその ASCII 値である '¥x0a' を指定できます。

注意 *column-spec* の *delimiter-string* には最大 4 文字を指定できますが、DELIMITED BY オプションでは、ASCII 文字またはその 16 進表現を 1 つのみ指定できます。

STRIP STRIP 句では、引用符で囲まれていない値を挿入する前に、後続ブランクを削除するかどうかを指定します。LOAD TABLE コマンドでは、次の STRIP キーワードを指定できます。

- **STRIP OFF** 後続ブランクを削除しません。
- **STRIP RTRIM** 後続ブランクを削除します。
- **STRIP ON** 廃止されました。STRIP RTRIM に相当します。

STRIP を ON (デフォルト) に指定すると、後続ブランクを削除してから値を挿入します。これは **VARCHAR** データの場合にのみ効果的です。**STRIP** オプションをオフにするには、次の句を使用します。

```
... STRIP OFF ...
```

後続ブランクが削除されるのは、引用符で囲まれていない文字列だけです。引用符で囲まれた文字列は、後続ブランクを保持します。ブランクを区別する必要がない場合は、後続スペースをすべて削除する代わりに、**FILLER** オプションを使用して、削除するバイト数をより詳細に指定できます。Sybase IQ の場合、このオプションをオフにしたほうが効率的です。後続ブランクの処理は ANSI 規格に準拠します。**CHAR** データでは、常にブランクが埋め込まれます。したがって、この **STRIP** オプションが有効なのは、**VARCHAR** データの場合だけです。

STRIP オプションは、可変長の非バイナリ・データだけに適用され、ASCII 固定幅の挿入には適用されません。たとえば、次のようなスキーマを想定できます。

```
CREATE TABLE t( c1 VARCHAR(3) );
LOAD TABLE t( c1 ',' ) ..... STRIP RTRIM // trailing blanks trimmed

LOAD TABLE t( c1 ',' ) ..... STRIP OFF // trailing blanks not trimmed

LOAD TABLE t( c1 ASCII(3) ) ... STRIP RTRIM // trailing blanks not trimmed
LOAD TABLE t( c1 ASCII(3) ) ... STRIP OFF // trailing blanks trimmed

LOAD TABLE t( c1 BINARY ) ..... STRIP RTRIM // trailing blanks trimmed
LOAD TABLE t( c1 BINARY ) ..... STRIP OFF // trailing blanks trimmed
```

バイナリ・データの後続ブランクは常に削除されます。

QUOTES オプション **QUOTES** パラメータは省略可能であり、デフォルトは **ON** です。**QUOTES** が **ON** の場合、**LOAD TABLE** は入力文字列が引用符文字に囲まれていることを想定します。引用符文字はアポストロフィ (一重引用符) または二重引用符のいずれかです。文字列の中で最初に出てくるこのような文字は、文字列の引用符文字として処理されます。文字列データは、対応する引用符文字で終わっている必要があります。

QUOTES ON の場合、**カラム・デリミタ文字** または **ロー・デリミタ文字** を **カラム値の一部** とすることができます。先行する引用符文字と終端の引用符文字は、値の一部とはみなされず、ロードされるデータ値から取り除かれます。

QUOTES OFF の場合、Sybase IQ は、アポストロフィ (一重引用符) または二重引用符を削除しません。入力ファイル内で引用符が検出されると、データの一部として処理されます。**QUOTES OFF** の場合、**カラム・デリミタ文字** を **カラム値の一部** とすることができません。

構文と使用方法の詳細については、『リファレンス：文とオプション』の「[第1章 SQL 文](#)」の「[LOAD TABLE 文](#)」を参照してください。

LOAD TABLE QUOTES オプションの例 次のように定義されたテーブルを想定します。

```
CREATE TABLE t1 (c1 INT, c2 VARCHAR(20), c3 VARCHAR(20))
```

入力データは次のとおりです。

```
1, apple , fruit1 ,
2, "banana" , "fruit2",
3, " pear ", " fruit3 ",
```

次のクエリを実行すると、このデータのロード結果が表示されます。

```
SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1
```

次の表に、LOAD TABLE コマンドで使用される QUOTES および STRIP オプションの値と、各値を指定した場合の上記のクエリの結果を示します。各結果は '<' と '>' で囲まれています。

LOAD TABLE オプション		SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1 の結果				
QUOTES	STRIP	c1	c2	c3	length(c2)	length(c3)
ON	RTRIM	<1>	<apple>	<fruit1>	<5>	<6>
		<2>	<banana>	<fruit2>	<6>	<6>
		<3>	< pear >	< fruit3 >	<6>	<8>
ON	OFF	<1>	<apple >	<fruit1 >	<6>	<7>
		<2>	<banana>	<fruit2>	<6>	<6>
		<3>	< pear >	< fruit3 >	<6>	<8>
OFF	RTRIM	<1>	< apple>	< fruit1>	<6>	<7>
		<2>	< "banana">	< "fruit2">	<9>	<9>
		<3>	< "pear">	< "fruit3">	<9>	<11>
OFF	OFF	<1>	< apple >	< fruit1 >	<7>	<8>
		<2>	< "banana" >	< "fruit2">	<10>	<9>
		<3>	< "pear">	< "fruit3">	<9>	<11>

結果についての注意

- QUOTES ON で STRIP RTRIM の場合、囲まれていないフィールド c2 のロー 1 は、先行スペースと後続スペースの両方がトリムされます。
- QUOTES ON で STRIP OFF の場合、囲まれていないフィールド c2 のロー 1 は、先行スペースだけがトリムされます。
- QUOTES OFF で STRIP RTRIM の場合、囲まれていないフィールド c2 のロー 1 は、後続スペースだけがトリムされます。

- QUOTES OFF で STRIP OFF の場合、囲まれていないフィールド c2 のロー 1 は、先行スペースも後続スペースもトリムされません。
- QUOTES ON で STRIP RTRIM の場合、囲まれているフィールド c2 および c3 のロー 3 は、引用符内の先行スペースと後続スペースの両方ともトリムされません。

ロード・オプションの指定

Sybase IQ では、さまざまなロード・オプションを指定して、入力ファイルの解釈方法と処理方法、エラーが発生した場合の処理内容を指定できます。

ロード・オプションは、どのような順序で指定してもかまいません。すべてのオプションの詳細については、『リファレンス：文とオプション』の「[LOAD TABLE 文](#)」を参照してください。

FILLER オプションの例 Windows の例を次に示します。

```
LOAD TABLE nn
    (l_orderkey,
     l_quantity ASCII(PREFIX 2),
     FILLER(2),
 FROM 'C:\¥iq¥archive¥¥mill.txt'
 BYTE ORDER LOW
```

LIMIT オプションの例 次に示す Windows の例では、ローは省略されずに、最大 1,000,000 のローが挿入されます。

```
LOAD TABLE lineitem
    (l_shipmode ASCII(15),
     l_quantity ASCII(8),
     FILLER(30))
 FROM 'C:\¥iq¥archive¥¥mill.txt'
 PREVIEW ON
 LIMIT 1000000
```

ROW DELIMITED BY オプションの例 次の Windows の例では、l_orderkey カラムのカラム・デリミタをタブに設定し、ロー・デリミタを改行 (¥x0a) とそれに続く復帰 (¥x0d) の組に設定します。

```
LOAD TABLE mm
    (l_orderkey '¥x09',
     l_quantity ASCII(4),
     FILLER(6),
     l_shipdate DATE('YYYY/MM/DD'))
 FROM 'C:\¥iq¥archive¥¥mill.txt'
 ROW DELIMITED BY '¥x0a¥x0d'
```

SKIP オプションの例 この UNIX の例では、Sybase IQ は入力ファイルから 9,000 個のローを読み込み、最初の 5,000 個を省略して次の 4,000 個のローをロードします。入力ファイルに 8,000 個のローしかない場合は、3,000 個だけロードされます。

```
LOAD TABLE lineitem(
  l_shipmode ASCII(15),
  l_quantity ASCII(8),
  FILLER(30))
FROM '/d1/MILL1/tt.t'
LIMIT 4000
SKIP 5000
PREVIEW ON
```

LOAD TABLE でのローの追加

LOAD TABLE 文は、ファイルの内容をテーブルの既存のローに追加します。既存のローを置き換えるわけではありません。既存のローを置き換えるには、START ROW ID ロード・オプションを指定します。

既存のテーブルを空にするには、TRUNCATE TABLE 文を使用してすべてのローを削除します。

クライアントからの直接のデータ・ロード

Sybase IQ では、LOAD TABLE USING CLIENT FILE 文を使用したりリモート・データのバルク・ロードをサポートしています。LOAD TABLE USING FILE は、ローカル・サーバ上のデータをロードします。これは、廃止された iq_bcp ユティリティに代わるものです。

サーバとクライアントの両方が、Sybase IQ バージョン 15.0 以降である必要があります。サーバのリリースが 15.0 以降で、クライアントが 12.7 の場合は、ファイル I/O エラーが返されます。

構文については、『リファレンス：文とオプション』の「LOAD TABLE 文」の「USING 句」を参照してください。

分割されたテーブルのロード

分割されたテーブルをロードする場合は、いくつかの点に特に注意する必要があります。

- 分割されたテーブルを修正するときは、分割カラムがコマンドのカラム・リストの最初のカラムである場合に最高のパフォーマンスが得られます。

INSERT...LOCATION 文の SELECT 句で、ラージ・オブジェクト (LOB) カラムの前に分割カラムをリストし、プライマリ・ファイルからデータをロードしてください。可能であれば、プリロード・プロセスを使用して、プライマリ・ファイル内のデータを並べ替えてください。

- 分割されたテーブルでは、LOAD TABLE 文および INSERT 文の START ROW ID 句を使用できません。次のエラーが報告され、ロード操作がロールバックされます。

```
"Option START ROW ID not allowed on a partitioned table."
(SQLCODE -1009416L, SQLSTATE QCB14, Sybase error code
21054)
```

- 分割されたテーブルと分割されていないテーブルでは、APPEND_LOAD データベース・オプションの動作が異なります。各パーティションにロー ID の範囲が割り当てられます。

分割されたテーブルで APPEND_LOAD が ON の場合は、適切なパーティションの最後に新しいローが追加されます。APPEND_LOAD が OFF の場合は、適切なパーティションの削除されたローから最初に使用可能なロー ID および領域が再利用されます。

分割されていないテーブルで APPEND_LOAD が ON の場合は、テーブルのローの最後にある最大ロー ID の後に、新しいローが追加されます。APPEND_LOAD が OFF の場合は、削除されたロー ID が再利用されます。LOAD または INSERT の START ROW ID 句を使用して挿入を開始するローを指定することもできます。

- 分割カラムの内容を更新しようとすると、次のエラーが発生します。

```
"Updating partition key column on a partitioned table is
not allowed."
(SQLCODE -1009417L, SQLSTATE QCB15, Sybase error code
21055)
```

テーブルの分割は、個別にライセンスが必要な Sybase IQ VLDB Management オプションの一部です。

メッセージ・ロギングの制御

挿入およびロード操作中に表示されるメッセージの詳細については、「[通知メッセージの解釈](#)」(558 ページ) を参照してください。NOTIFY_MODULUS データベース・オプションは、ロード中に通知メッセージを表示する頻度のデフォルト値を調節したり、このメッセージを省略したりします。LOAD コマンドで NOTIFY オプションを指定すると、NOTIFY_MODULUS 設定が上書きされます。詳細については、『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」を参照してください。

IQMsgMaxSize サーバ・プロパティと -iqmsgsz サーバ起動スイッチによって、メッセージ・ログ・ラッピングとメッセージ・ログ・ファイルのサイズを制御します。詳細については、「[第 1 章 Sybase IQ システム管理の概要](#)」の「[メッセージ・ログ・ラッピング](#)」を参照してください。

整合性制約違反のロギング

LOAD TABLE では、整合性制約違反があった場合のロード動作を制御したり、違反についての情報をログに適宜記録したりすることができます。MESSAGE LOG ... ROW LOG オプションで ONLY LOG 句を指定すると、ロード時に特定の種類の整合性制約違反の情報をロギングできます。メッセージ・ログ・ファイルでは違反ごと、ロー・ログ・ファイルではローごとに記録されます。ONLY LOG 句を指定しなかった場合、これらのファイルには、ロードの開始と完了を示すタイムスタンプのみがロギングされます。

なお、整合性制約違反が記録されるメッセージ・ログ・ファイルとロー・ファイルは、IQ メッセージ・ログ・ファイル (.iqmsg) とは別物です。

MESSAGE LOG の内容とフォーマット

MESSAGE LOG ファイルには、それぞれの整合性制約違反のローとカラムの情報がロギングされています。各ロードについて、ロギングされるメッセージは3種類あります。タイムスタンプが付いたヘッダ、ロー情報、タイムスタンプが付いたトレーラです。ヘッダは各ロードにつき1回記録されます。トレーラは、文が正常に実行された場合に1回記録されます。ロー情報は、ロギングされた整合性制約違反それぞれに対して1回記録されます。

ヘッダ・メッセージのフォーマットは次のとおりです。

```
<datetime load started> Load Table <table-name>: Integrity Constraint Violations
```

例を示します。

```
2009-05-24 23:04:31 Load Table Customers: Integrity Constraint Violations
```

ロー情報メッセージは次の3つの部分で構成されます。

- **rowid** 整合性制約違反がなかったとしたらこのローがロードされていた、テーブル内でのロー番号です。
- **タイプ** 検出された整合性制約違反の種類です。
- **カラム番号** スキーマで指定されたカラム番号です。

例を示します。

```
1267 DATA VALUE 4
3216 UNIQUE 1
3216 NULL 3
3216 NULL 6
9677 NULL 1
```

トレーラ・メッセージのフォーマットは次のとおりです。

```
<datetime load completed> Load Table <table-name> Completed
```

例を示します。

```
2009-05-24 23:05:43 LOAD TABLE Customers: Completed
```

注意 MESSAGE LOG ファイルのローの数 (報告されるエラーの数) は、IGNORE CONSTRAINT オプションの制限を超えることがあります。並行して動作する複数のスレッドによってロードが実行されるからです。制約違反の数が指定された制限値を超えたということが、複数のスレッドによって報告される場合があります。

ROW LOG の内容とフォーマット

ROW LOG ファイルには、整合性制約違反が発生した各ローの rowid とデータ値がロギングされています。ローのデータは、各ローにつき 1 回のみ記録されます。そのローで整合性制約違反が何件発生したかは関係ありません。各ロードについて、ロギングされるメッセージは 3 種類あります。タイムスタンプが付いたヘッダ、ロー・データ、タイムスタンプが付いたトレーラです。ヘッダは各ロードにつき 1 回記録されます。トレーラは、文が正常に実行された場合に 1 回記録されます。

ヘッダ・メッセージのフォーマットは次のとおりです。

```
<datetime load started> Load Table <table-name>: Integrity Constraint Violations
<formatting information>
```

ここで、<formatting information> は、ロー・データのフォーマットに使用されている、日付、時刻、日付時刻フォーマットを表します。例を示します。

```
2009-05-24 23:04:31 Load Table Customers: Integrity Constraint Violations
Date Format: yyyy/mm/dd
Time Format: hh:mm:ss
Datetime format: yyyy/mm/dd hh:mm:ss
```

ロー・データ・メッセージは次の 2 つの部分で構成されます。

- **rowid** 整合性制約違反がなかったとしたらこのローがロードされていた、テーブル内でのロー番号です。
- **データ値** ロー内のデータ値です。カンマか、またはユーザ指定の LOG DELIMITED BY セパレータで区切られています。

例を示します。

```
3216 #Jones John#NULL#NULL#S#1945/01/12#NULL#
```

ロー・データ・メッセージでのデータ値のフォーマットは、次の規則に応じて決まります。

- データ型が **VARBINARY** または **BINARY** の場合は、データは ASCII の 16 進文字で表されます。
- **DATE** 値は、**DATE_FORMAT** データベース・オプションで指定されたフォーマットで表されます。デフォルト・フォーマットは **YYYY-MM-DD** です。

- DATETIME 値と TIMESTAMP 値は、TIMESTAMP_FORMAT データベース・オプションで指定されたフォーマットで表されます。デフォルトは YYYY-MM-DD HH:NN:SS.SSS です。
- TIME 値は、TIME_FORMAT データベース・オプションで指定されたフォーマットで表されます。デフォルトは HH:NN:SS.SSS です。
- NULL 値は、トークン NULL で表されます。

注意 ロー・データ・メッセージには、フィラー・フィールドは含まれません。

トレーラ・メッセージのフォーマットは次のとおりです。

```
<datetime load completed> Load Table <table-name>: Completed
```

例を示します。

```
2009-05-24 23:05:43 Load Table Customers: Completed
```

注意 MESSAGE LOG ファイルに記録された個々のエラーの数と、ROW LOG ファイルに記録されたローの数は一致しない場合があります。両者のローの数の違いは、複数のスレッドによって実行される、ロードの並行処理によるものです。制約違反の数が指定された制限値を超えたということが、複数のスレッドによって報告される場合があります。

MESSAGE LOG と ROW LOG の例

この例では、MESSAGE LOG ファイルと ROW LOG ファイルの内容とフォーマットを示します。

次の CREATE TABLE 文は、テーブルを作成します。このテーブルは、LOAD TABLE 文を使用してロードするためのものです。

```
CREATE TABLE Customers(name VARCHAR(80) NOT NULL,  
age TINYINT NULL,  
sex CHAR(1) NOT NULL,  
marital_status CHAR(1) NULL,  
birthdate DATE NOT NULL,  
credit_card VARCHAR(20) NOT NULL)
```

次の LOAD TABLE 文は、Customers テーブルにデータをロードします。

```
LOAD TABLE Customers ...  
IGNORE CONSTRAINT UNIQUE 200  
MESSAGE LOG 'msg.log' ROW LOG 'row.log'  
ONLY LOG UNIQUE, NULL, DATA VALUE  
LOG DELIMITED BY '#'
```

次に示す生データは、上記の LOAD TABLE 文でディスク・ファイルからロードされるデータです。

```
Jones John, 19, M, S, 06/19/83, CC
Cleven Bill, 56, M, OSIDJFJ, 02/23/43, CC
Jones John, 339, M, NULL, 01/12/45, NULL
NULL, 55, F, M, 10/02/37, ST
```

LOAD TABLE が完了した時点で、MESSAGE LOG ファイル *msg.log* には次の情報が記録されています。

```
2009-05-24 23:04:31 LOAD TABLE Customers: Integrity Constraint Violations
1267 DATA VALUE 4
3216 UNIQUE 1
3216 NULL 6
9677 NULL 1
2009-05-24 23:05:43 LOAD TABLE Customers Completed
```

LOAD TABLE が完了した時点で、ROW LOG ファイル *row.log* には次の情報が記録されています。

```
2009-05-24 23:04:31 LOAD TABLE Customers Integrity Constraint Violations
Date Format: yyyy/mm/dd
Time Format: hh:mm:ss
Datetime format: yyyy/mm/dd hh:mm:ss

1137 #Jones John#19#M#S#1983/06/19#CC#
1267 #Cleven Bill#56#M#OSIDJFJ#1943/02/23#CC#
3216 #Jones John#NULL#NULL#S#1945/01/12#NULL#
9677 #NULL#55#F#M#1937/10/02#ST#

2009-05-24 23:05:43 LOAD TABLE Customers Completed
```

BINARY ロード形式の使用

Sybase IQ へのデータのロードを高速化するには、Sybase IQ のバイナリ・フォーマットでデータ・ファイルを作成し、LOAD TABLE の FORMAT BINARY および BINARY カラム指定句を使用して、このデータを Sybase IQ にロードします。

LOAD TABLE の使用方法については、『リファレンス：文とオプション』を参照してください。

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

- DATE
- TIME
- DATETIME
- NUMERIC

IQ の BINARY ロード形式とロード効率

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

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

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

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

オペレーティング・システムのネイティブ・データ型

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

- BIT (1 バイト)
- TINYINT (1 バイト)
- SMALLINT (2 バイト)
- INT/UNSIGNED INT (4 バイト)
- BIGINT/UNSIGNED BIGINT (8 バイト)
- FLOAT (4 バイト)
- DOUBLE (8 バイト)
- CHAR/VARCHAR (文字データ)
- BINARY/VARBINARY (バイナリ・データ)

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

DATE

DATE カラム・データは、0000-01-01 以降の日数を表す 4 バイト (32 ビットの符号なし整数) として Sybase IQ に格納されます。

ある年、月、日に対応する暦日を Sybase IQ のバイナリ・フォーマットに変換するには、次の式を使用します。

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

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

TIME

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

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

TIMESTAMP

TIMESTAMP データは、マイクロ秒数を表す 64 ビットの符号なし整数として格納されます。ある年、月、日、時間、分、秒、マイクロ秒に対応するバイナリ TIMESTAMP 値は、次のように計算できます。

前述の日付の *binaryDateValue* を計算します。

前述の時間の *binaryTimeValue* を計算します。

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

NUMERIC と DECIMAL

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

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

値に 0 を埋め込み、小数点を削除すると、次の規則が当てはまります。

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

```
struct {
    unsigned char sign; // sign 1 for +, 0 for -
    unsigned char ndig; // # digits
    unsigned char exp; // exponent
    unsigned char erracc; // should be 0
    unsigned short digits[80];
};
```

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

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

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

```
lower order digit = digit[i] & 0x00FF
high order digit = digit[i] & 0xFF00
```

たとえば、NUMERIC(20) として表現される値 100 を考えてみます。この値のバイナリ・レイアウトは次のようになります。

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

Sign = 0x01
Number digits = 0x01
Exponent = 0x50
Erracc = 0x00
Digits = 0x0064
```

別の例として、値 32769 を考えてみます。

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

Sign = 0x01
Number digits = 0x02
Exponent = 0x50
Erracc = 0x00
Digits = 0x0ad1 0x0003
```

これらの数を基数 10 に変換すると、次のようになります。

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

NULL の挿入

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

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

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

```
000b32cb01000b32cc00
```

別の例として、値 32769 を NUMERIC(20) カラムにロードする場合、入力ファイルの内容は次のようになります。

```
0x0102 0x5000 0x0ad1 0x0003 0x0000 0x00
```

これには NULL BYTE が含まれています。

NUMERIC(19,2) と定義されたカラムに 23456789012345678.12 をロードする場合、ロード入力ファイルの内容は次のようになります。

```
0x0106 0x4f00 0x04b0 0x162e 0x04d2 0x1ed2 0x0d80 0x0002 0x0000
0x00
```

数字の後には、NULL BYTE (0x00) が続きます。

この数字を表す構造の桁配列には、7 つ (0 ~ 6) の符号なし短整数型が含まれています。“digits[0]” には、最下位桁が含まれます。

```
digits[0] = 0x04b0 (decimal 120)
digits[1] = 0x162e (decimal 5678)
digits[2] = 0x04d2 (decimal 1234)
digits[3] = 0x1ed2 (decimal 7890)
digits[4] = 0x0d80 (decimal 3456)
digits[5] = 0x0002 (decimal 2)
digits[6] = 0x0000
```

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

ZEROS は次のように解釈されます。

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

ロード：

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

ビッグ・エンディアン・バイト順を使用する入力データ・ファイルの表示：

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

実行：

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

結果：

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

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

ロード：

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

ビッグ・エンディアン・バイト順を使用する入力データ・ファイルの表示：

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

実行：

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

結果：

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

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

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

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

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

ビッグ・エンディアン・バイト順を使用する入力データ・ファイルの表示：

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

INSERT 文の使用

INSERT 文を使用すると、フラット・ファイルに最初にデータを記述しなくても、データを挿入できます。このコマンドを使用して、次の処理を実行できます。

- 指定した一連の値をロー単位で挿入する。
- データベース・テーブルからデータを直接挿入する。

これらの2つのコマンド形式については、この後の項を参照してください。

指定した値のロー単位の挿入

指定した値をテーブルにロー単位で追加するには、『リファレンス：文とオプション』の「INSERT 文」の構文 1 を使用します。

Sybase IQ は、指定された最初の値を指定された最初のカラムに挿入し、指定された 2 番目の値を 2 番目のカラムに挿入し、以下同様に処理します。カラム名のリストを指定しない場合は、カラムの作成順 (SELECT * を使って取り出すときと同じ順序) にテーブル・カラムの中に値を挿入します。Sybase IQ は、テーブルに空きがある場合には必ずローを挿入します。

値には、NULL、任意の正数と負数、またはリテラルを使用できます。

- CHAR、VARCHAR、DATE、TIME、TIMESTAMP または DATETIME カラムの値を一重引用符や二重引用符で囲みます。引用符を含む値を指定するには、“Smith's”のように、その引用符の外側を別の引用符の組で囲みます。

- DATE、TIME、TIMESTAMP または DATETIME カラムの場合は、特定のフォーマットを使用します。データ型の変換については、「[挿入時のデータ変換](#)」を参照してください。Sybase IQ の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』の「[第3章 SQL データ型](#)」を参照してください。

注意 TIMESTAMP と DATETIME のデータ型は同一です。

NULL 値の許可

カラムに NULL が許可されている場合に、ローのカラムの一部にのみ値を指定すると、値が指定されなかったカラムには NULL が挿入されます。NULL 値を指定する場合は、ロード先のカラムで NULL が許可されている必要があります。許可されていない場合は、INSERT が拒否されて、メッセージ・ログにエラー・メッセージが出力されます。Sybase IQ のカラムでは、デフォルトで NULL が許可されていますが、CREATE TABLE 文のカラム定義で NOT NULL を指定するか、プライマリ・キーなどの別の方法を使用して、この設定を変更できます。

例

次の例では、lineitem テーブルの l_shipdate カラムに 1995-06-09 が、l_orderkey カラムに 123 が、それぞれ追加されます。

```
INSERT INTO lineitem
    (l_shipdate, l_orderkey)
VALUES ('1995-06-09', 123)
```

データ・ローを多数挿入する場合は、次の項で説明するように、選択したローをデータベースから直接挿入するか、または LOAD TABLE 文を使用してフラット・ファイルからデータをロードする方が、値をロー単位で挿入するよりも効率的です。少数のローに値を挿入するのではなく、サーバへのアクセス回数が 1 回で済むように、少数の union を使用して SELECT 文を使用するようにしてください。

データベースからの指定したローの挿入

現在のデータベースの他のテーブル、または特殊データ・ストアとして定義されたデータベースから Sybase IQ にデータを挿入する場合は、『リファレンス：文とオプション』の「[INSERT 文](#)」の構文 2 を使用します。

この形式の INSERT 文を使用すると、一般的な SELECT 文の結果に基づいてデータのローをいくつでも挿入できます。

効率を最大化するには、1 つの INSERT 文でできるだけ多くのローを挿入します。最初の挿入後にさらにローを挿入するには、別の INSERT 文を使用します。

他の SQL データベースと同様に、Sybase IQ は、ロード先カラム・リスト内で指定されたカラムの順番と選択リストを照合して、データを挿入します。つまり、選択リスト内の最初のカラムのデータは最初のロード先カラムに挿入され、以下同じように処理されます。INSERT SELECT と INSERT VALUES でロード先カラム名を省略した場合は、Sybase IQ がデータをカラムに挿入する順番は、カラムが作成された順番と同じになります。

挿入先のテーブルは、現在接続しているデータベース内にある必要があります。Sybase IQ はデータをロード先カラムのすべてのインデックスに挿入します。

選択リスト内にあるテーブルのカラムのデータ型は同じか、または互換性がある必要があります。つまり、選択した値はテーブルのカラムのデータ型の値であるか、またはそのデータ型に変換可能でなければなりません。データ型と変換オプションの詳細については、「[挿入時のデータ変換](#)」を参照してください。

この形式の INSERT 文では、いずれかの insert-load-options を指定できます。

これらの全オプションの詳細については、「[ロード・オプションの指定](#)」を参照してください。

例 この例では、テーブル `partsupp` から同じデータベース内の別のテーブル `lineitem` ヘデータを挿入する場合を示します。ソース・カラム `l_quantity` のデータはロード先カラム `ps_availqty` に挿入されます。

```
INSERT INTO partsupp (ps_availqty)
SELECT l_quantity FROM lineitem
```

別のデータベースからの挿入

以下のアクセス可能なデータベースのテーブルからデータを挿入できます。

- 現在接続しているデータベースの IQ ストアまたはカタログ・ストアのテーブル。
- Adaptive Server Enterprise データベース内のテーブル。
- リモート・サーバ上のデータベースのテーブルに対応する、現在接続しているデータベースのプロキシ・テーブル。詳細については、『システム管理ガイド第2巻』の「[第4章 リモート・データへのアクセス](#)」と「[第5章 リモート・データ・アクセス用のサーバ・クラス](#)」を参照してください。

Adaptive Server Enterprise データベースからの直接挿入

Adaptive Server Enterprise や SQL Server データベースのデータは、INSERT 文の LOCATION 構文を使用して簡単に挿入できます。この方法で、選択したカラムを2つの Sybase IQ データベース間で移動することもできます。

この機能を使用するには、次のすべての条件が整っている必要があります。

- Sybase のコネクティビティ・ライブラリをシステムにインストールして、プラットフォームのロード・ライブラリ・パス環境変数をインストール先ディレクトリに設定する。

- 接続中の Adaptive Server Enterprise サーバがローカル・マシン上の *interfaces* ファイルに存在している。
- ソース・データベースである ASE または Sybase IQ データベースの読み込みパーミッション、およびターゲット・データベースである Sybase IQ データベースの INSERT パーミッションがある。

❖ Adaptive Server Enterprise からの直接のデータ挿入

- 1 同じユーザ ID とパスワードを使用して、Adaptive Server Enterprise と Sybase IQ データベースの両方に接続します。

- 2 Sybase IQ データベースで、次の構文を使用する文を発行します。

```
INSERT INTO iq_table
LOCATION 'ase_servername.ase_dbname'
{ SELECT col1, col2, col3,...
FROM owner.ase_table }
```

- 3 COMMIT を発行して、挿入を実行します。

Sybase IQ がリモート・サーバに接続するとき、CREATE EXTERNLOGIN でリモート・ログインが作成されており、CREATE SERVER 文でリモート・サーバが定義されている場合は、INSERT...LOCATION は現在の接続のユーザ ID にリモート・ログインを使用できます。詳細、完全な構文、および例については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」の「[INSERT 文](#)」を参照してください。

ASE の text および
image データのロード

Sybase IQ は Adaptive Server Enterprise のデータ型である TEXT をサポートしていませんが、長さが 255 バイトを超える IQ の CHAR または VARCHAR カラム、または LONG VARCHAR カラムから INSERT...LOCATION (構文 3) を実行できます。また、ASE データベース・カラムのデータ型である TEXT から実行可能です。ASE の TEXT および IMAGE カラムは、Sybase IQ が内部変換をサポートしていれば、他の Sybase IQ データ型のカラムに挿入できます。また、INSERT...LOCATION では、SELECT 文での変数の使用はサポートされていない点にも注意してください。デフォルトでは、リモート・データ・カラムの内容が 2GB を超えている場合、Sybase IQ は暗黙的にカラム値を 2GB にトランケートします。

非構造化データ分析機能を使用するには、正規のライセンスを取得しておく必要があります。詳細については、『Sybase IQ の非構造化データ分析の概要』を参照してください。

SELECT 文を区切る一重引用符の代わりに中カッコ {} を使用することもできます。なお、中カッコは、ODBC 標準ではエスケープ・シーケンスの開始と終了を表すため、ODBC のコンテキストではエラーが発生する可能性があります。

さらに大きなデータをロードする必要がある場合は、「[LOAD TABLE 文を使ってデータのバルク・ロード](#)」を参照してください。

INSERT 文の構文の詳細については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」を参照してください。

例 次のコマンドを実行すると、**detroit** サーバの Sybase IQ データベース **iq11db.dba** にある **lineitem** テーブルの **l_shipdate** と **l_orderkey** カラムから、現在のデータベースの **lineitem** テーブルの対応するカラムにデータが挿入されます。

```
INSERT INTO lineitem
  (l_shipdate, l_orderkey)
  LOCATION 'detroit.iq11db'
  { SELECT l_shipdate, l_orderkey
    FROM lineitem }
```

- ロード先カラムの名前とソース・カラムの名前は違ってよい。
- 指定された最初のソース・カラムが、指定された最初のターゲット・カラムに挿入され、以下同じように処理されるため、カラムを指定する順序は重要である。
- **INSERT** コマンド内で **SELECT** 文の述部を使用して、テーブル内の特定のローからのみデータを挿入できる。

例 この例では、挿入されるカラムは前の例と同じですが、**l_orderkey** が 1 のローのみが挿入されます。また、この例では、TDS パケット・サイズを 512 バイトと指定しています。

```
INSERT INTO lineitem
  (l_shipdate, l_orderkey)
  LOCATION 'detroit.iqdb'
  PACKETSIZE 512
  { SELECT l_shipdate, l_orderkey
    FROM lineitem
    WHERE l_orderkey = 1 }
```

データの対話的インポート

少量のデータを挿入する場合は、Interactive SQL で **INSERT** 文を使用して対話的に値を入力してもかまいません。

たとえば、次のコマンドを使用すると、1 つのローに対して、リストされた値を一度に挿入できます。

```
INSERT INTO T1
VALUES ( ... )
```

INSERT コマンドの詳細については、「[INSERT 文の使用](#)」を参照してください。

注意 Interactive SQL の [データ] メニューの [インポート] オプションは使用しないでください。このオプションは、Sybase IQ データベースの使用時にはサポートされません。

エンディアン・フォーマットが異なるシステムへのデータの移動

この項では、ビッグ・エンディアン・フォーマットのデータベースのデータをリトル・エンディアン・フォーマットのデータベースに移動する方法について説明します。この手順では、テーブル定義を移動しますが、再作成が必要なストアド・プロシージャやイベントなどのデータベース・オブジェクトの移行は行いません。

たとえば、Sun64 SPARC システムで作成された Sybase IQ データベースは、ビッグ・エンディアン (最上位バイトを先頭に配置) フォーマットのバイナリ・データを格納します。Sun Solaris x64 はリトル・エンディアン・システムであるため、Sun64 SPARC で作成された Sybase IQ データベースを ALTER DATABASE UPGRADE を使用してアップグレードし、Sun Solaris x64 システムで実行することはできません。

エンディアン構造が異なるハードウェア・プラットフォーム間でデータベースのデータを移動するには、次の手順に従う必要があります。

- ソース・プラットフォームからデータベース・スキーマ (テーブル、インデックスなど) をコピーします。
- ターゲット・プラットフォームで新しいデータベースを作成します。
- ソース・データベースからバイナリ・データのダンプを実行します。
- 新しいターゲット・データベースにデータをロードします。

❖ ビッグ・エンディアン・システムからリトル・エンディアン・システムへ、またはリトル・エンディアン・システムからビッグ・エンディアン・システムへのデータの移動

注意 作業を始める前に、データベース・スキーマとテーブル・スキーマを取得する手段があることを確認してください。

次の例では、`lineitem` という名前のテーブルをロードし、UFS (ファイル・システム) の `lineitem_binary.inp` という名前の抽出ファイルを特定します。

システムの最大ファイル・サイズについては、オペレーティング・システムのマニュアルで確認してください。たとえば、Sun Solaris x64 の抽出ファイルの最大サイズは、512GB です。

- 1 抽出オプションを有効にします。

```
SET TEMPORARY OPTION Temp_Extract_Name1 =
'lineitem_binary.inp'
SET TEMPORARY OPTION Temp_Extract_Name2 = ''
```

- 2 `lineitem` テーブルのバイナリ抽出ファイルを設定します。

```
SET TEMPORARY OPTION Temp_Extract_Binary = 'on'
SET TEMPORARY OPTION Temp_Extract_Swap = 'off'
```

- 3 出力を *lineitem_binary.inp* ファイルに格納します。

```
SELECT * FROM lineitem
```

- 4 抽出オプションを無効にします。

```
SET TEMPORARY OPTION Temp_Extract_Name1 = ''
```

- 5 ターゲット・システムにデータベースのコピーを作成します。

- 6 *lineitem* テーブルが次のように定義されている場合、次の手順で *lineitem* テーブルをロードします。

```
LOAD TABLE lineitem
( l_orderkey      BINARY WITH NULL BYTE,
  l_partkey       BINARY WITH NULL BYTE,
  l_suppkey       BINARY WITH NULL BYTE,
  l_linenumbers   BINARY WITH NULL BYTE,
  l_quantity      BINARY WITH NULL BYTE,
  l_extendedprice BINARY WITH NULL BYTE,
  l_discount      BINARY WITH NULL BYTE,
  l_tax           BINARY WITH NULL BYTE,
  l_returnflag    BINARY WITH NULL BYTE,
  l_linestatus    BINARY WITH NULL BYTE,
  l_shipdate      BINARY WITH NULL BYTE,
  l_commitdate    BINARY WITH NULL BYTE,
  l_receiptdate   BINARY WITH NULL BYTE,
  l_shipinstruct  BINARY WITH NULL BYTE,
  l_shipmode      BINARY WITH NULL BYTE,
  l_comment       BINARY WITH NULL BYTE )
FROM 'C:\¥¥mydata¥¥lineitem_binary.inp'
FORMAT BINARY
STRIP OFF
QUOTES OFF
ESCAPES OFF
PREVIEW ON
BYTE ORDER HIGH;
COMMIT
```

特に次の2つの句に注意してください。

- BINARY WITH NULL BYTE はバイナリ・ファイルのロード時に必要になります。
- BYTE ORDER HIGH は、元のデータが格納されたシステムからのバイト順序を指定します。この例のソース・データベースは、ビッグ・エンディアン・プラットフォームであるため、このデータにはバイト順序 HIGH が必要になります (リトル・エンディアン・データベースにはバイト順序 LOW が必要になります)。

マルチプレックス・データベースをロードする場合、ファイル名に絶対パス (完全に修飾されたパス) を使用します。相対パス名は使用しないでください。

ジョイン・インデックスのテーブルへの挿入

他のインデックスと同様にジョイン・インデックスの基本となるテーブルにデータをロードまたは挿入できます。唯一の違いは、ジョイン・インデックスを使用してクエリを解析する前に、ジョイン・インデックスのデータの同期を取る必要があることです。

注意 ジョイン・インデックスの一部となっているベース・テーブルを更新することはできません。挿入、ロード、または削除のみ可能です。

ジョイン・インデックスを初めて作成する場合、Sybase IQ によって自動的にジョイン・インデックスの同期が取られます。ジョイン・インデックスは、ロードの前後を問わず作成できます。ロードとジョイン・インデックス作成の順序は、ロードまたは同期のパフォーマンスに影響はありません。

ただし、ジョイン・インデックスを作成すると、ジョイン階層の一番上のテーブル以外の基本となるテーブルにデータを挿入またはロードする場合、明示的に同期を取る必要があります。これを行うには、『リファレンス：文とオブション』の「[ジョイン・インデックスの同期](#)」または「[SYNCHRONIZE JOIN INDEX 文](#)」を参照してください。

1人のユーザがジョイン・インデックスのテーブルを更新すると、ジョイン・インデックスの同期を取るまで、他のユーザはそのジョイン・インデックスの基本となるテーブルを更新できません。

エラーの原因となる異なる接続からの更新

複数のユーザが同じジョイン・インデックスに関連する別々のテーブルにデータを挿入するか、または削除する場合、2人目のユーザがトランザクションを開始する前に同期を行わないと、2人目のユーザは更新できません。次のいずれかの場合に、更新は失敗します。

- 最初のユーザのトランザクションが実行される前に 2 人目のユーザのトランザクションが開始される場合
- 最初のユーザのトランザクションが実行された後、ジョイン・インデックスの同期を取る前に 2 人目のユーザが更新しようとする場合

この問題が生じる原因は、基本となるテーブルが更新されたときに Sybase IQ が新しいバージョンのジョイン・インデックスを作成することにあります。すでに開始されている他のトランザクションは新しいバージョンのジョイン・インデックスを認識できません。新しいバージョンのテーブルは元の変更を行ったユーザに見えるため、1人のユーザがすべての変更を行う場合は問題はありません。

たとえば、テーブル A、B、C が同じジョイン・インデックスのメンバーであると仮定します。ユーザ 2 がトランザクションを開始し、ジョインに関連のない他のテーブルに書き込みます。次に、ユーザ 1 がテーブル B にデータを挿入します。この操作によって新しいバージョンのテーブル B と新しいバージョンのジョイン・インデックスが作成されます。ユーザ 2 は次にテーブル C に書き込みをしようとしています。他のユーザがテーブル C を変更していない場合でも、テーブル C はジョイン・インデックスのメンバーなので、ジョイン・インデックスの同期を取るまで更新できません。

ジョイン・インデックスの詳細については、「[第 6 章 Sybase IQ インデックスの使用](#)」を参照してください。トランザクション処理の詳細については、「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

プライマリ・キー・カラムと外部キー・カラム

他のカラムと同様にプライマリ・キー・カラムと外部キー・カラムにデータをロードまたは挿入します。

プライマリ・キーに挿入すると、Sybase IQ は、各値がユニークかどうかをチェックします。各値がユニークではない場合は、エラーが発生します。

ラージ・オブジェクト・データの抽出

ラージ・オブジェクト・データをロードまたは抽出する方法については、『Sybase IQ の非構造化データ分析の概要』の「[第 6 章 ラージ・オブジェクト・データのロードとアンロード](#)」を参照してください。非構造化データ分析機能を使用するには、正規のライセンスを取得しておく必要があります。

挿入時のデータ変換

Sybase IQ データベースに入力するデータは、さまざまなソースから取り込まれます。Sybase IQ のデータ型と正確に一致しないデータもあります。このため、データの変換が必要になることがあります。データは、明示的または暗黙的に変換されます。たとえば、INT データを CHAR カラムに挿入する場合は、明示的に変換します。

暗黙的な変換は次の場合に行われます。

- 同じデータベースの別のカラムから選択したデータを挿入する場合
- 他のデータベースから選択されたデータを挿入する場合
- フラット・ファイルからデータをロードする場合

明示的な変換が必要な場合、変換を指定する方法は、フラット・ファイルからロードするか、選択されたローを挿入するかによって異なります。

- **LOAD TABLE** 文で、*column-spec* にフォーマットを指定することによって、明示的にデータを変換する。
- **INSERT** 文で、**SELECT** 文または **VALUES** リストにデータ変換関数 **CAST**、**CONVERT**、**DATEPART** を使用することによって、明示的にデータを変換する。

Sybase IQ のデータ型の暗黙的および明示的な変換については、「[IQ におけるデータ変換](#)」(309 ページ) の表を参照してください。

プロキシ・テーブルから挿入する場合の変換については、『システム管理ガイド第2巻』の「[第4章 リモート・データへのアクセス](#)」を参照してください。

Sybase IQ のデータ型の多くは、SQL Anywhere および Adaptive Server Enterprise の同名のデータ型と完全な互換性がありますが、いくつかの相違点もあります。互換性の詳細については、「[Adaptive Server Enterprise のデータ型の対応](#)」(321 ページ) を参照してください。

バージョン間の互換性を確保するために、いくつかのデータ型は他のデータ型の同意語として定義されています。

- **DECIMAL** は **NUMERIC** の同意語です。
- **INTEGER** は **INT** の同意語です。
- **DATETIME** は **TIMESTAMP** の同意語です。
- **FLOAT** (*precision*) は、*precision* の値によって、**REAL** か **DOUBLE** のいずれかの同意語です。Adaptive Server Enterprise では、*precision* が 15 以下の場合には **REAL** が使用され、*precision* が 15 よりも大きい場合には **DOUBLE** が使用されます。Sybase IQ および SQL Anywhere の場合、カットオフ値はプラットフォームによって異なりますが、すべてのプラットフォームでカットオフ値は 22 より大きい値になります。
- **MONEY** は、Adaptive Server Enterprise の **NUMERIC(19,4)** と互換性のある同意語で、**NULL** を許容します。
- **SMALLMONEY** は、Adaptive Server Enterprise の **NUMERIC(10,4)** と互換性のある同意語で、**NULL** を許容します。

標準データ型と同じように同意語を使用できます。同意語が存在する場合、データは内部で標準データ型として格納されます。エラー・メッセージには同意語ではなく標準名が表示されます。

注意 デフォルトでは、Sybase IQ は入力データをバイナリ (数値データ) とみなし、バイナリとして挿入します。ただし、入力カラム長 (バイト) は宛先カラム長 (バイト) と一致する必要があります。一致しない場合、挿入は失敗するか、または予期しない結果が生じます。たとえば、4 バイトの整数データの入力カラムを SMALLINT 宛先カラムに挿入する場合、Sybase IQ はその入力カラムの最初の 2 バイトのみをロードします。

15 より前のバージョンの Sybase IQ からのデータの挿入

以前のバージョンから Sybase IQ バージョン 15.x にデータを移動する場合、挿入またはロードする前に特定のデータ型を変換する必要があります。詳細については、『インストールおよび設定ガイド』の「第 6 章 データの移行」の「以前のバージョンからのアップグレード」を参照してください。

ロード変換オプション

次の表では、LOAD TABLE 文の変換オプションをアルファベット順に示し、各オプションの機能について簡単に説明します。各オプションの詳細については、以降の項を参照してください。LOAD TABLE 文にこれらのオプションを使用する場合は、「[ロード・オプションの指定](#)」を参照してください。

表 7-1: フラット・ファイルからロードするための変換オプション

オプション	Sybase IQ のデータ型	対処法
ASCII	TINYINT、SMALLINT、INT (または INTEGER)、UNSIGNED INT、BIGINT、UNSIGNED BIGINT、NUMERIC (または DECIMAL)、REAL、DOUBLE、BIT、DATE、TIME、TIMESTAMP (または DATETIME)	デフォルトでは、Sybase IQ は、入力データは当該データ型に適切な幅のバイナリとみなします。ASCII を使用すると、データが文字フォーマットであることを Sybase IQ に伝え、幅を指定できます。このオプションでは、REAL データの E 表記ができませんが、パフォーマンスが低下することがあります。
ASCII	CHAR、VARCHAR	デフォルトでは、Sybase IQ は、ソース・カラムと宛先カラムのカラム幅は同じとみなすため、入力ファイルが正しく読み込めない場合があります。このオプションを使用して異なる入力カラム幅を指定できます。
DATE	DATE	固定フォーマットの ASCII 日付入力をバイナリに変換します。
DATETIME	TIMESTAMP (または DATETIME) または TIME	固定フォーマットの ASCII 時刻または日付/時刻入力をバイナリに変換します。入力指定は 12 時間制または 24 時間制に基づきます。
TIME	TIME	固定フォーマットの ASCII 時刻入力をバイナリに変換します。
NULL	all	挿入時に NULL に変換する入力データ値を指定します。

注意 フラット・ファイルからロードするときに、バイナリ・データまたは文字データを選択できる場合、バイナリ・データを使用します。バイナリ入力を使用すると、変換が不要なため、パフォーマンスが向上します。

IQ におけるデータ変換

INSERT 文を使用してフラット・ファイルではなくデータベースから直接データを挿入する場合、ロード変換オプションを使用できません。データを明示的に変換する必要がある場合、挿入するデータを指定する SELECT 文または VALUES リストで、変換関数 CAST または CONVERT を使用します。データを暗黙的に変換する場合は、Sybase IQ によって自動的に変換されます。

SELECT 文でデータ型を一致させる必要があるのに一致しない場合、暗黙的または明示的な変換が必要です。あるデータ型から他のデータ型への INSERT SELECT を実行したり、データ型が異なる値を比較または計算したりする場合に、SELECT 文でデータ型が一致しなくなります。

以下の各表に、次の項目を示します。

- Sybase IQ が暗黙的に行う変換 (I)
- 明示的に行う必要がある変換 (E)
- サポートされていない変換 (U)

これらの変換は、Sybase IQ データベース、SQL Anywhere データベース、または特殊データ・ストアとして接続された他のデータベースのデータに適用されます。

最初の表は、SELECT 文に WHERE 句が指定されていない場合、または WHERE 句が比較演算 (=、>、または <) に基づく場合の、暗黙的 (I)、明示的 (E)、未サポート (U) の変換を示します。

表 7-2: 比較演算のための IQ 変換

From:	To:																
	ti	si	in	ui	bi	ub	nu	rl	dl	bt	dt	tm	ts	ch	vc	bn	vb
tinyint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
smallint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
int	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
unsigned int	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
bigint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
unsigned bigint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
numeric	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	U	U
real	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	U	U
double	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	U	U

	To:																
From:	ti	si	in	ui	bi	ub	nu	rl	dl	bt	dt	tm	ts	ch	vc	bn	vb
bit	I	I	I	I	I	I	I	I	I	I	U	U	U	I	I	I	I
date	E	E	E	E	E	E	E	E	E	U	I	U	I	E	E	U	U
time	E	E	E	E	E	E	E	E	E	U	U	I	E	E	E	U	U
time-stamp	E	E	E	E	E	E	E	E	E	U	E	I	I	E	E	U	U
char	E	E	E	E	E	E	E	E	E	I	E	E	E	I	I	I	I
varchar	E	E	E	E	E	E	E	E	E	I	E	E	E	I	I	I	I
binary	I	I	I	I	I	I	U	U	U	U	U	U	U	I	I	I	I
varbinary	I	I	I	I	I	I	U	U	U	U	U	U	U	I	I	I	I

次のリストは、表 7-2、表 7-3、表 7-4 で使用されているコードの意味を示します。

コード	データ型	コード	データ型	コード	データ型
ti	tinyint	nu	numeric	ts	timestamp
si	smallint	rl	real	ch	char
in	int	dl	double	vc	varchar
ui	unsigned int	bt	bit	bn	binary
bi	bigint	dt	date	vb	varbinary
ub	unsigned bigint	tm	time		

2 番目の表は、SELECT 文の WHERE 句が算術演算 (+、-、など) に基づく場合の暗黙的 (I)、明示的 (E)、未サポート (U) の変換を示します。

表 7-3: 算術演算のための IQ 変換

	To:																
From:	ti	si	in	ui	bi	ub	nu	rl	dl	bt	dt	tm	ts	ch	vc	bn	vb
tinyint	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	I	I
smallint	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	I	I
int	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	I	I
unsigned int	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	I	I
bigint	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	I	I
unsigned bigint	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	I	I
numeric	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	U	U
real	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	U	U
double	I	I	I	I	I	I	I	I	I	I	U	U	U	E	E	U	U
bit	I	I	I	I	I	I	I	I	I	I	U	U	U	I	I	I	I
date	U	U	U	U	U	U	U	U	U	U	U	I	U	U	U	U	U
time	U	U	U	U	U	U	U	U	U	U	I	U	U	U	U	U	U
timestamp	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
char	E	E	E	E	E	E	E	E	E	I	U	U	U	I	I	I	I

	To:																
From:	ti	si	in	ui	bi	ub	nu	rl	dl	bt	dt	tm	ts	ch	vc	bn	vb
varchar	E	E	E	E	E	E	E	E	E	I	U	U	U	I	I	I	I
binary	I	I	I	I	I	I	U	U	U	U	U	U	U	I	I	I	I
varbinary	I	I	I	I	I	I	U	U	U	U	U	U	U	I	I	I	I

注意 算術演算では、bit データは暗黙的に tinyint に変換されます。

3 番目の表は、INSERT 文と UPDATE 文の暗黙的 (I)、明示的 (E)、未サポート (U) の変換を示します。

表 7-4: INSERT と UPDATE のための IQ 変換

	To:																
From:	ti	si	in	ui	bi	ub	nu	rl	dl	bt	dt	tm	ts	ch	vc	bn	vb
tinyint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
smallint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
int	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
unsigned int	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
bigint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
unsigned bigint	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	I	I
numeric	I	I	I	I	I	I	I	I	I	E	E	E	E	E	E	U	U
real	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	U	U
double	I	I	I	I	I	I	I	I	I	I	E	E	E	E	E	U	U
bit	I	I	I	I	I	I	I	I	I	I	U	U	U	I	I	I	I
date	E	E	E	E	E	E	E	E	E	E	I	U	I	E	E	U	U
time	E	E	E	E	E	E	E	E	E	E	U	I	E	E	E	U	U
time-stamp	E	E	E	E	E	E	E	E	E	E	E	I	I	E	E	U	U
char	I	I	I	I	I	I	I	I	I	I	E	E	E	I	I	I	I
varchar	I	I	I	I	I	I	I	I	I	I	E	E	E	I	I	I	I
binary	I	I	I	I	I	I	U	U	U	I	U	U	U	I	I	I	I
varbinary	I	I	I	I	I	I	U	U	U	I	U	U	U	I	I	I	I

カラム幅の問題

Sybase IQ は、入力データの幅が宛先カラムの幅と同じとみなして、入力ファイルを読み込みます。幅が同じでない場合、Sybase IQ が入力ファイルの当該カラムから読み込むバイト数が少なすぎたり多すぎたりすることがあります。この結果、カラムのデータを正しく読み込むことができません。また、これに続くカラムの読み込み開始位置がずれてしまうため、以降のカラムについても、データを正しく読み込むことができません。

たとえば、`input_column1` の幅が 15 バイト、`destination_column1` の幅が 10 バイトで、ASCII 変換オプションを指定しない場合、Sybase IQ は、入力カラムの幅が 10 バイトであるとみなします。入力データはいずれにせよ 10 バイトにトランケートされるため、`destination_column1` については問題ありません。しかし、入力ファイルの次のカラムの正しい開始位置はバイト 16 ですが、Sybase IQ は、最初のカラムの途中のバイト 11 から始まるとみなします。

逆に、`input_column1` の幅が 10 バイト、`destination_column1` の幅が 15 バイトで、ASCII 変換オプションを指定しない場合、Sybase IQ は入力カラム幅は 15 バイトとみなします。Sybase IQ は、入力ファイルから `input_column1` のすべてと次のカラムの 5 バイトを読み込み、その値を `destination_column1` に挿入します。このため、`destination_column1` に挿入された値と、それ以降のすべてのカラムの値は、いずれも正しくありません。

このような問題を避けるには、ASCII 変換オプションを使用します。このオプションによって、Sybase IQ は、入力カラムの固定または可変幅を指定するさまざまな方法を提供します。入力データには、特定のサイズ(バイト単位)の固定幅入力カラム、カラム・デリミタを含む可変幅入力カラム、バイナリ・プレフィクス・バイトで定義される可変幅入力カラムを含めることができます。

日付と時刻のロードの最適化

Sybase IQ には、日付、時刻、日付時刻データのロード時に ASCII からバイナリへの変換を行うためのパフォーマンスの最適化が組み込まれています。ロードする生データがこれらのフォーマットのいずれかにぴったり合致する場合は、適切なフォーマットを使用することによってロード時間を大幅に短縮できます。認識されるフォーマットは次のとおりです。

- “YYYY-MM-DD”
- “YYYY/MM/DD”
- “YYYY.MM.DD”
- “YYYYMMDD”
- “MM-DD-YYYY”
- “MM/DD/YYYY”
- “DD-MM-YYYY”
- “DD/MM/YYYY”

- “DD.MM.YYYY”
- “HH:NN:SS”
- “HHNNSS”
- “HH:NN:SS.S”
- “HH:NN:SS.SS”
- “HH:NN:SS.SSS”
- “HH:NN:SS.SSSS”
- “HH:NN:SS.SSSSS”
- “HH:NN:SS.SSSSSS”
- “YYYY-MM-DD HH:NN:SS”
- “YYYYMMDD HHNNSS”
- “YYYY-MM-DD HH:NN:SS.S”
- “YYYY-MM-DD HH:NN:SS.SS”
- “YYYY-MM-DD HH:NN:SS.SSS”
- “YYYY-MM-DD HH:NN:SS.SSSS”
- “YYYY-MM-DD HH:NN:SS.SSSSS”
- “YYYY-MM-DD HH:NN:SS.SSSSSS”

日付、時刻、または日付時刻カラムが1つまたは複数あるテーブルをロードする場合で、入力フォーマットが前述のいずれかのフォーマットのときには、ロード文で適切なフォーマットを明示的に指定すれば、ロード処理は非常に高速になります。そうしないと、ロードは非常に遅くなることがあります。

次のように作成した、日付カラムを持つテーブルがあるとします。

```
CREATE TABLE table1 (c1 DATE);
```

このテーブルをロードするには、次のような文を使用します。

```
LOAD TABLE table1 (c1 ASCII(10)) FROM ...
```

生データ・フォーマットが、最適化されたフォーマット(YYYY-MM-DDなど)となっている場合の方が、ロードははるかに速くなります。

以下の項では、変換オプションについてさらに詳しく説明します。

LOAD TABLE の ASCII 変換オプションの使用

ASCII 変換オプションによって次の操作を行います。

- ASCII 入力データをバイナリに変換し、カラムのデータが正しく読み取れるように入力カラムの幅を指定します。
- 入力カラムの幅が宛先カラムの幅と異なる場合、ASCII データを ASCII データ型のカラムに挿入します。このオプションは、各カラムで読み込む入力データのサイズを指定できます。

1、2、または 4 バイトのプレフィックスおよびカラム・デリミタによって、すべての Sybase IQ データ型にこのオプションを使用できます。

VARCHAR および CHAR
カラムのデータのトラン
ケート

Sybase IQ は、入力カラムの幅が宛先カラムの幅より大きい場合、挿入時にデータをトランケートします。入力データの幅が宛先カラムの幅より小さい場合、CHAR または VARCHAR データ型では、Sybase IQ は挿入時にデータにスペースを埋め込みます。

VARCHAR カラムへの可変幅挿入では、後続ブランクは削除されませんが、VARCHAR カラムへの固定幅挿入では削除されます。たとえば、テーブル `variable` のカラム `varcolumn` に挿入する場合を考えてみます。次の例は固定幅挿入で、2 つのブランク (_ で示す) が明示的に指定されているため、値は削除されません。

```
INSERT INTO vartable VALUES ('box__')
```

区切られた入力を使用してフラット・ファイルから同じ値を入力する場合は、可変幅挿入となり、後続ブランクは削除されます。

次の表は、Sybase IQ のデータ型に対する ASCII 変換オプションの動作を示します。この例では、フラット ASCII ファイル `shipinfo.t` から Sybase IQ テーブル `lineitem` にデータを挿入し、入力データとテーブルの内容とフォーマットの要約を示します。

表 7-5: 入力ファイル変換例

ファイル <code>shipinfo.t</code>			テーブル <code>lineitem</code>		
カラム	フォーマット	幅	カラム	データ型	幅
<code>l_shipmode</code>	CHAR	15	<code>l_shipmode</code>	VARCHAR	30
<code>l_quantity</code>	ASCII	8	<code>l_quantity</code>	INT	4

`l_shipmode` カラムの場合、ASCII データを ASCII カラム (VARCHAR データ型) に挿入します。2 つのカラムの幅が異なることに注意してください。このカラムと次の `l_quantity` カラムから正しく挿入するために、`l_shipmode` カラムの幅を指定して、正しいサイズの入力データを正しい位置で読み込めるようにします。

`l_quantity` カラムでは、ASCII データをバイナリ・カラム (INT データ型) に挿入します。このカラムから正しく挿入するため、入力データをバイナリに変換し、入力カラムの幅を指定する必要があります。

次に示す UNIX の例は変換を指定するコマンドです。

```
LOAD TABLE lineitem(  
    l_shipmode ASCII(15),  
    l_quantity ASCII(8),  
    FILLER(1))  
FROM '/d1/MILL1/shipinfo.t'  
PREVIEW ON
```

NULL またはブランク文字の置換

Sybase IQ では、長さ 0 の VARCHAR データがサポートされます。VARCHAR のセルの長さが 0 で、そのセルが NULL でない場合、長さ 0 のセルが作成されます。

その他のデータ型では、セルの長さがゼロの場合、Sybase IQ によって NULL が挿入されます。

このような長さ 0 の文字データの処理は、ANSI 規格に準拠しています。ANSI 以外の処理を行う場合は、『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」の「NON_ANSI_NULL_VARCHAR オプション」を参照してください。

DATE オプション

DATE 変換オプションによって、固定フォーマットで格納された ASCII データを DATE カラムに挿入します。このオプションは、ASCII データ入力をバイナリに変換し、入力データのフォーマットを指定します。DATE フォーマットは入力を解釈するために内部で使用され、データの格納または出力フォーマットに影響はありません。詳細については、ASCII 変換フォーマットを参照してください。

例

この Windows の例では、l_shipdate カラムのデータは、指定フォーマットからバイナリに変換されます。1 バイトの FILLER は入力ファイル内の改行を省略します。

```
LOAD TABLE lineitem(  
    l_orderkey NULLS (ZEROS) ASCII(4),  
    l_partkey ASCII(3),  
    l_shipdate DATE('MM/DD/YY'),  
    l_suppkey ASCII(5),  
    FILLER(1))  
FROM 'C:¥¥MILL1¥¥shipinfo.t'  
PREVIEW ON
```

DATE フォーマットの指定

y または Y (年)、m または M (月)、d または D (日)、j または J (ユリウス日付) を使用して入力データのフォーマットを指定します。フォーマット文字列の長さは入力カラム幅です。表 7-6 は、フォーマット・オプションを示します。

表 7-6: 日付のフォーマット

オプション	意味
yyyy または YYYY yy または YY	年を表す。デフォルトは 1900 年。
mm または MM	月を表す。1 桁の月は、月の先頭に 0 を付ける。たとえば、5 月は '05' となる。DATE 値で月を省略すると、日付はユリウス日付として扱われる。たとえば、'03' のように月のみを入力した場合、Sybase IQ はデフォルトの年と日を適用し、'1900-03-01' に変換する。
dd または DD jjj または JJJ	日付を表す。デフォルトの日付は 01。1 桁の日付は、日付の先頭に 0 を付ける。たとえば、最初の日は '01' となる。J または j は年のユリウス日付 (1 ~ 366) を示す。

入力時にはフォーマット・コードの大文字と小文字は無視されます。

出力時にはフォーマット・コードの大文字と小文字は次のように処理されます。

- 大文字と小文字が混在する場合 (たとえば、“Dd”)、ゼロを補わない。
- 大文字または小文字のみの場合 (たとえば、“DD”)、足りない桁の分だけゼロを補う。

たとえば、デフォルトの時刻フォーマットを使用する場合、時刻は 17:23:03.774 で、'HH:NN:Ss.SSS' を使用する場合は、17:23:3.774 となります。

次の表には、日付入力データと DATE 変換オプションによるフォーマットの指定方法が示されています。表の次に、日付を指定するための一般的な規則を示します。

表 7-7: DATE フォーマット・オプションの例

入力データ	フォーマット指定
12/31/09	DATE ('MM/DD/YY')
12-31-09	DATE ('MM-DD-YY')
20091231	DATE ('YYYYMMDD')
12/09	DATE ('MM/YY')
2009/123	DATE ('YYYY/JJJ')

- DATE 指定は、カッコと一重引用符または二重引用符で囲みます。
- Sybase IQ は、年、月、日の値のみを格納し、入力データの他の文字は格納しません。ただし、入力データに、スラッシュ (/)、ダッシュ (-)、ブランクなどの月、日、年の区切り文字が含まれる場合は、DATE フォーマットにその文字の位置を示し、無視できるようにします。
- Y、M、J、D 以外の文字を使用して Sybase IQ が省略する区切り文字を指定します。ブランクも使用できます。

- DATE フォーマットに年と、その年の日数のみを指定すると、Sybase IQ は日付をユリウス日付とみなします。たとえば、2009-33 は 2009 年の 33 日目、つまり、2009 年 2 月 2 日です。
- “5/27/32” などのように年を 2 桁のみで指定する場合、Sybase IQ は年および NEAREST_CENTURY オプションの設定に応じて 19yy または 20yy に変換します。

NEAREST_CENTURY 設定	指定された年	想定される年
デフォルト (50)	00-49 50-99	2000-2049 1950-1999
0	任意	1900 年代
100	任意	2000 年代

詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」を参照してください。

DATETIME 変換オプション

固定フォーマットで格納された ASCII データを、TIME、TIMESTAMP、または DATETIME カラムに挿入するには、DATETIME 変換オプションを使用します。このオプションは、ASCII データ入力をバイナリに変換し、入力データのフォーマットを指定します。DATETIME フォーマットは入力を解釈するために内部で使用され、データの格納または出力フォーマットに影響はありません。詳細については、ASCII 変換フォーマットを参照してください。

注意 以前のリリースとの互換性を保つため、カラムに DATETIME データが含まれるよう指定できません。ただし、DATETIME データは内部では同じフォーマット TIMESTAMP として格納されます。

次の構文を使用します。

```
DATETIME ('input-datetime-format')
```

この UNIX の例では、スラッシュは入力データの日付部分の区切り文字、コロンは時刻部分の区切り文字です。

```
LOAD TABLE lineitem(
  l_quantity ASCII(4),
  l_shipdate DATETIME('MM/DD/YY hh:mm:ss'),
  FILLER(1))
FROM '/dl/MILL1/tt.t'
PREVIEW ON
```

この UNIX の例では、FILLER(1) 句があるため、Sybase IQ は DATETIME カラムの次のカラム (VWAP) に NULL を挿入しません。

```
LOAD TABLE snapquote_stats_base
SYMBOL '¥x09',
snaptime DATETIME('MM/DD/YY hh:mm:ss'),
FILLER(1)
VWAP '¥x09',
RS_DAY '¥x09',
FROM '/d1/MILL1/tt.t'
PREVIEW ON
```

この UNIX の例では、宛先カラムに TIME データが含まれていますが、入力データは DATETIME です。DATETIME 変換オプションと FILLER を使用して日付部分をスキップします。

```
LOAD TABLE Customers (
    open_time DATETIME('hh:mmaa'),
    close_time DATETIME('hh:mmaa'),
    FILLER(9))
FROM '/d1/MILL1/tt.t'
PREVIEW ON
```

DATETIME 変換フォーマットの指定

DATETIME 入力データのフォーマットを次のように指定します。

- 年は Y または y
- 月は M または m
- 日は D または d
- 時刻は H または h
- 分を示す N または n (コロンを区切り文字として使用する場合、mm も使用できる)
- 秒またはコンマ以下の秒数を表す S または s

フォーマット文字列の長さは、入力カラム幅です。表 7-6 は、日付フォーマット・オプションを示します。次の表では、時間のフォーマット・オプションについて説明します。

表 7-8: 時間のフォーマット

オプション	意味
hh HH	時間を示す。時間は 24 時間表記に基づく。1 桁の時間は時間の前に 0 を付ける。たとえば、午前 1 時は '01' となる。午前 12 時を表す場合、'00' も有効な値である。
nn	分を示す。1 桁の分は、分の前に 0 を付ける。たとえば、8 分は '08' となる。
ss[.sssss]	秒数とコンマ以下の秒数を示す。
aa	午前または午後を示す。
pp	必要な場合にのみ、午後を示す。このオプションはリリース 12.0 より前の Sybase IQ とは互換性がない。以前は、pp は aa と同義だった。
hh	Sybase IQ は、分と秒をゼロとみなす。たとえば、入力した DATETIME 値が '03' の場合、Sybase IQ は '03:00:00.0000' に変換する。
hh:nn または hh:mm	Sybase IQ は秒をゼロとみなす。たとえば、入力した時刻値が '03:25' の場合、Sybase IQ は '03:25:00.0000' に変換する。

表 7-9 は、時刻入力データと DATETIME オプションのフォーマットの指定方法の例を示します。表の次に、時刻を指定する場合の一般的な規則を示します。

表 7-9: DATETIME フォーマット・オプション

入力データ	フォーマット指定
12/31/00 14:01:50	DATETIME ('MM/DD/YY hh:nn:ss')
123100140150	DATETIME ('MDDYYhhnnss')
14:01:50 12-31-00	DATETIME ('hh:mm:ss MM-DD-YY')
12/31/00 14:01:12.456	DATETIME ('MM/DD/YY hh:nn:sssss')
12/31/00 14:01.:123456	DATETIME ('MM/DD/YY hh:mm:sssss')
12/31/00 02:01:50AM	DATETIME ('MM/DD/YY hh:mm:ssaa')
12/31/00 02:01:50pm	DATETIME ('MM/DD/YY hh:mm:sspp')

- 時刻要素の指定文字は、かっこ一重引用符または二重引用符で囲みます。
- 浮動小数点を含め、9 桁までの秒数を入力できるため、コンマ以下の秒数を指定できます。入力とクエリでは、小数点の位置が変わるため、6 つまでの小数点位置を指定できます。ただし、Sybase IQ では、常に 2 桁の秒数と 6 桁の小数点以下の数値のみが格納されます (ss.ssssss)。それ以上の小数点以下の数値は指定できません。
- 区切り文字は、時間要素を区切るために使用します。ブランクを含むすべての文字は、区切り文字として使用できます。例ではコロン (:) を使用しています。
- Sybase IQ は、時間、分、秒の数値のみを格納し、入力データの他の文字は格納しません。ただし、時間、分、秒を区切るためにコロン (:) またはブランクなどの他の文字がデータに含まれる場合は、フォーマット指定の時刻部分にこれらの文字の位置を指定し、Sybase IQ がスキップできるようにします。

- 特定の値が午前または午後であることを示す場合は、入力データの特定の位置に大文字または小文字の 'a' または 'p' を指定してください。Sybase IQ が午前または午後の指定を検索する場所を示すため、フォーマット指定の適切な場所に小文字で 'aa' または 'pp' を指定します。'aa' は常に午前 / 午後を示すことを指定し、'pp' は必要な場合にのみ午後であることを示すことを指定します。
- フォーマット指定には、入力のすべての文字と一致する文字を使用する必要があります。'm' は分を表すために使用されているため、入力の 'm' と一致させる目的では使用できません。
- 時刻セクションで時間、分、または秒を指定しない場合、Sybase IQ はそれぞれの値を 0 と見なします。

NULL の処理

Sybase IQ カラム・インデックスに挿入するときに、入力データの特定の値を NULL に変換するには、NULL 変換オプションを使用します。このオプションは、すべてのカラムに使用できますが、カラムは NULL を許容する必要があります。すべての Sybase IQ のデータ型に、この変換オプションを指定できます。

次の構文を使用します。

```
NULLS ({BLANKS | ZEROS | literal' ['literal']...})
```

各パラメータの意味は、次のとおりです。

- **BLANKS** は、ブランクを NULL に変換することを示します。
- **ZEROS** は、バイナリ・ゼロを NULL に変換することを示します。
- **literal** は、指定したリテラルをすべて NULL に変換することを示します。指定したリテラルが入力ファイルの値と一致していると Sybase IQ に認識させるためには、先行ブランクおよび後続ブランクも含めて、両者が正確に一致している必要があります。20 までのリテラル値をリストできます。

同じカラムに追加の変換オプションを指定することが必要な場合があります。たとえば、ASCII データを INT カラムに挿入し、バイナリ・フォーマットで格納し、入力データのブランクを挿入時に NULL に変換する場合、ASCII 変換オプションを使用して入力をバイナリに変換し、さらに NULL 変換オプションを使用してブランクを NULL に変換します。

Windows の場合の例を次に示します。

```
LOAD TABLE lineitem(
  l_orderkey NULLS (ZEROS) ASCII (4),
  l_partkey ASCII (3),
  l_shipdate date ('MM/DD/YY'),
  l_suppkey ascii (5),
  FILLER (1))
FROM 'C:¥¥MILLl¥¥tt.t'
PREVIEW ON
```

データ表示に関連する他の項目

Sybase IQ では、クエリまたは挿入時にデータ型を明示的または暗黙的に変換する必要がある場合、結果は常にトランケートされます。以下に、このような場合について説明します。

- データを明示的に低い位取りに変換する場合、Sybase IQ は結果の値をトランケートします。たとえば、位取り 4 に格納されているクエリのカラム値を位取り 2 に CAST する場合は、2.4561 の値は 2.45 となります。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、およびプロシージャ』の「第4章 SQL 関数」を参照してください。
- Sybase IQ は、挿入時に低い位取りに暗黙的に変換する場合は、データをテーブルに挿入する前に値をトランケートします。たとえば、データ型 NUMERIC(7,3) のテーブルからデータ型 DECIMAL(12,2) のテーブルへ挿入する場合、2.456 の値は 2.45 となります。
- 算術演算の結果、規定の位取りより高くなった場合、Sybase IQ では、『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「第3章 SQL データ型」の「数値データ型」に定められている規則に従って位取りが決定された後に、その位取りに合わせて結果をトランケートします。

結果の値をトランケートではなく、丸める必要がある場合は、コマンドに ROUND 関数を使用します。ただし、挿入の場合、ROUND 関数をクエリ式の一部にのみ適用できます。

数値データの最大精度は 126 です。

Adaptive Server Enterprise のデータ型の対応

次の表に Sybase IQ データ型と Adaptive Server Enterprise データ型との互換性を示します。

一般的な規則は次のとおりです。

- Sybase IQ 文字列型は、いかなる Adaptive Server Enterprise 文字列型でも受け入れます。
- Sybase IQ 真数値型は、いかなる Adaptive Server Enterprise 数値型でも受け入れます。ただし、Sybase IQ データ型が、Adaptive Server Enterprise データ型よりも少ないデータしか格納できない場合は、値は NULL に変換されます (たとえば、基本のデータベースからデータをテーブルに挿入する場合)。
- Sybase IQ の日付/時刻型は、Adaptive Server Enterprise のすべての日付/時刻型を受け入れます。

サポートされていない Adaptive Server Enterprise のデータ型

次に示す Adaptive Server Enterprise のデータ型は、このバージョンの Sybase IQ ではサポートされていません。

- nchar、nvarchar
- unichar、univarchar、unitext
- text
- image
- unsigned smallint
- ネイティブな Java データ型
- XML データ型

次の点に注意してください。

- Sybase IQ は、バイナリ・ラージ・オブジェクト (BLOB) およびキャラクタ・ラージ・オブジェクト (CLOB) データ型を通じて Adaptive Server Enterprise の text および image データ型をサポートします。詳細については、『Sybase IQ の非構造化データ分析の概要』を参照してください。
- Sybase Central では、SYSTEM DB 領域 (カタログ・ストア) でのみ、特定のデータ型のカラムを作成できます。テーブル作成ウィザードで、[システム・ストアにテーブルを作成] を選択します。
- Sybase IQ は、Adaptive Server Enterprise の DATE、TEXT、UNSIGNED SMALLINT、NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR、または UNITEXT データ型をサポートしていません。ただし、INSERT 文の LOCATION 構文を使用して、Adaptive Server Enterprise データベースの DATE、TEXT、UNSIGNED SMALLINT、NCHAR、NVARCHAR、UNICHAR、UNIVARCHAR、または UNITEXT データ型のカラムからデータを挿入できます。

Adaptive Server Enterprise データ型の等価性

次の表は、Adaptive Server Enterprise の真数値型と、それに対応する Sybase IQ のデータ型を示します。

表 7-10: integer データ型

Adaptive Server Enterprise のデータ型	Sybase IQ のデータ型	注意事項
int	INT、BIGINT、UNSIGNED INT、UNSIGNED BIGINT、または NUMERIC	Sybase IQ では、INT(7,3) などの位取りされた整数は使用できません。INT(<i>precision,scale</i>) 形式のデータは NUMERIC(<i>precision,scale</i>) に変換されます。12.0 より前のバージョンの Sybase IQ および Adaptive Server Enterprise では、int データ型は -2,147,483,648 ~ 2,147,483,647 の値となり、これと異なります。 大きな整数値を扱う場合は、BIGINT、符号なし整数 (UNSIGNED INT)、または UNSIGNED BIGINT データ型を使用できます。UNSIGNED INT の場合、最後のビットは値の一部として使用されます。正または負の指定はありません。すべての数値は正とみなされるため、値は最大で 4,294,967,295 とできます。
numeric	適切な精度の DECIMAL または NUMERIC	定義する Sybase IQ のデータ型の精度が低すぎるために Adaptive Server Enterprise の値を格納できない場合、その値は NULL に変換されます。
decimal	適切な精度の DECIMAL または NUMERIC	上記を参照してください。
smallint	SMALLINT または NUMERIC	Sybase IQ の SMALLINT では、精度と位取りは使用できません。Adaptive Server Enterprise の smallint(<i>precision,scale</i>) は NUMERIC(<i>precision,scale</i>) に変換されます。上記の INT を参照してください。
tinyint	TINYINT	Sybase IQ の TINYINT カラムでは、精度と位取りは使用できません。Adaptive Server Enterprise の tinyint(<i>precision,scale</i>) は NUMERIC(<i>precision,scale</i>) に変換されます。上記の INT を参照してください。
bit	BIT	
unsigned smallint	サポートされていない	Sybase IQ は、Adaptive Server Enterprise の unsigned smallint データ型をサポートしていませんが、INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの unsigned smallint データ型のカラムからデータを挿入できます。

次の表は、Adaptive Server Enterprise の概数値データ型と、それに対応する Sybase IQ のデータ型を示します。

表 7-11: 概数値データ型

Adaptive Server Enterprise のデータ型	Sybase IQ のデータ型	注意事項
float (precision)	FLOAT (precision)	IQ は FLOAT のより高い精度をサポートしています。 HNG インデックスには FLOAT、REAL、または DOUBLE データは使用できません。
double precision	DOUBLE	
real	REAL	

次の表は、Adaptive Server Enterprise の文字データ型と、それに対応する Sybase IQ のデータ型を示します。

表 7-12: 文字データ型

Adaptive Server Enterprise のデータ型	Sybase IQ のデータ型	注意事項
char	CHAR	Sybase IQ と Adaptive Server Enterprise の文字データ型 (char または CHAR) はほぼ同じです。唯一の違いは、Sybase IQ では NULL を扱える点です。Sybase IQ の CHAR カラムを、Adaptive Server Enterprise の char カラムと完全に一致させるには、Sybase IQ のカラムに NOT NULL と指定します。Sybase IQ では、デフォルトで NULL が許容されています。Adaptive Server Enterprise の NULL を許容する char カラムは、内部で varchar に変換されます。
varchar	VARCHAR	上記の char の注意を参照してください。
nchar	サポートされていない	Sybase IQ は、Adaptive Server Enterprise の nchar データ型をサポートしていませんが、INSERT...LOCATION を使用して、ASE データベースの nchar データ型のカラムからデータを挿入できます。
nvarchar	サポートされていない	Sybase IQ では、Adaptive Server Enterprise の nvarchar データ型をサポートしていませんが、INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの nvarchar データ型のカラムからデータを挿入できます。
text	サポートされていない	Sybase IQ は、Adaptive Server Enterprise の text データ型をサポートしていませんが、INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの text データ型のカラムからデータを挿入できます。
unichar	サポートされていない	Sybase IQ は、Adaptive Server Enterprise の unichar データ型をサポートしていませんが、INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの unichar データ型のカラムからデータを挿入できます。
univarchar	サポートされていない	Sybase IQ は、Adaptive Server Enterprise の univarchar データ型をサポートしていませんが、INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの univarchar データ型のカラムからデータを挿入できます。
unitext	サポートされていない	Sybase IQ は、Adaptive Server Enterprise の unitext データ型をサポートしていませんが、INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの unitext データ型のカラムからデータを挿入できます。

次の表は、Adaptive Server Enterprise の money データ型と、それに対応する Sybase IQ のデータ型を示します。

表 7-13: money データ型

Adaptive Server Enterprise のデータ型	Sybase IQ のデータ型	注意事項
money	NUMERIC(19,4)	money データは、暗黙的に NUMERIC(19,4) に変換されます。
smallmoney	NUMERIC(10,4)	

次の表は、Adaptive Server Enterprise の DATE/TIME データ型と、それに対応する Sybase IQ のデータ型を示します。

表 7-14: 日付および時刻データ型

Adaptive Server Enterprise のデータ型	Sybase IQ のデータ型	注意事項
datetime	TIMESTAMP、DATE、または TIME	Adaptive Server Enterprise の datetime カラムでは、仮想的な日付である 0/0/0000 を基準日として、その前後の日数を 4 バイトの数値、時刻を 8 バイトの数値として保持し、1/1,000,000 秒の精度で日付を指定できます。Sybase IQ の TIMESTAMP (または DATETIME) カラムは、24 時間制に基づき 1/10,000 秒の精度で、1/1/0 以降の日数と時刻の 2 つをそれぞれ 4 バイト整数値として保持します。Sybase IQ はこの変換を自動的に行います。 Sybase IQ には、 DATE データ型も別にあります。これは単一の 4 バイト整数です。SQL Server または Adaptive Server Enterprise の datetime カラムから日付だけを抽出する場合は、Sybase IQ の DATE データ型を使用して抽出できます。それには、Sybase IQ の DATE カラムを、Adaptive Server Enterprise の datetime カラムと同じ名前前で定義します。Sybase IQ は datetime 値の該当部分を自動的に選択します。
smalldatetime	TIMESTAMP、DATETIME、DATE、または TIME	Adaptive Server Enterprise の smalldatetime カラムを、Sybase IQ では TIMESTAMP (または DATETIME) データ型として定義します。Sybase IQ は変換を正しく行います。通常の datetime と同様に、Adaptive Server Enterprise の smalldatetime カラムから日付のみを抽出する場合は、Sybase IQ の DATE データ型を使用します。
date	date	INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの date データ型のカラムからデータを挿入できます。
time	time	Sybase IQ のデータ型は、時、分、秒、秒以下で構成される時間です。秒の小数点以下は 6 桁まで格納されます。 time 値には 8 バイトの格納領域が必要です。 Adaptive Server Enterprise の time データ型は 00:00:00.000 から 23:59:59.999 の間です。24 時間制、または正午を 12AM、真夜中を 12PM とする 12 時間制のどちらでも使用できます。 time 値には、コロンか AM/PM 指示子を含めてください。AM または PM は、大文字でも小文字でもかまいません。 time 値には 4 バイトの格納領域が必要です。 INSERT...LOCATION を使用して、Adaptive Server Enterprise データベースの time データ型のカラムからデータを挿入できます。

次の表は、Adaptive Server Enterprise の binary データ型と、それに対応する Sybase IQ のデータ型を示します。

表 7-15: バイナリ・データ型

Adaptive Server Enterprise のデータ型	Sybase IQ のデータ型	注意事項
binary	BINARY	Sybase IQ は、すべての BINARY カラムに対し、後続のゼロを埋め込みます。BINARY カラムの長さは、必ず偶数の文字数とします。 HNG インデックスでは、BINARY データは使用できません。
varbinary	VARBINARY	Sybase IQ は、VARBINARY カラムについては、後続のゼロの埋め込みやトランケートは行いません。VARBINARY カラムの長さは、必ず偶数の文字数とします。 HNG インデックスでは、VARBINARY データは使用できません。

次に示す Adaptive Server Enterprise のデータ型はサポートされていないため、これらのデータ型のカラムは省略する必要があります。

- nchar、nvarchar
- univar、univarchar
- unsigned smallint
- ネイティブな Java データ型

これは、Adaptive Server Enterprise のカスタム・データ型にも適用されます。

データのインポート時における変換エラーの処理

外部ソースからデータをロードするときには、エラーが発生する可能性があります。たとえば、日付カラムに日付としては有効でないデータや、数字カラムに有効でない数字が入力されている可能性があります。

CONVERSION_ERROR データベース・オプションを使用すると、変換エラーが NULL 値に変換されるため、それらのエラーを無視できます。

dbisql データベース・オプションの設定の詳細については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「SET OPTION 文」を参照してください。

データのバルク・ロードのチューニング

データベースに大量のデータをロードする場合、時間がかかり、多くのディスク領域が使用されます。時間を節約するには、次の方法があります。

データベースを定義するときのロード・パフォーマンスを向上させる

データベース、テーブル、インデックスを定義する方法によっては、ロード・パフォーマンスに大きな影響を及ぼします。

特殊値の数を最適化する

Sybase IQ は、大量または少量の別個の値に対して、データのロードを最適化します。これは、`MINIMIZE_STORAGE` データベース・オプションの設定と、データベースおよびテーブルの作成時に指定するパラメータに基づいて行われます。ロードの最適化に関するパラメータには、次のものがあります。

- `UNIQUE` オプションと `IQ UNIQUE` オプション、およびカラムのデータ型と幅。これらはすべて、`CREATE TABLE` または `ALTER TABLE` コマンドで指定します。
- `IQ PAGE SIZE`。これは `CREATE DATABASE` コマンドで指定します。

ロードに関連するパラメータとその指定方法については、「[テーブルの作成 \(196 ページ\)](#)」および「[IQ ページ・サイズの選択 \(174 ページ\)](#)」を参照してください。

インデックスの作成

システム・リソースを最大限に活用するには、データをロードする前に必要なすべてのインデックスを作成します。データをロードした後で新しいインデックスを追加できますが、すべてのインデックスを一度にロードした方がロード操作がすばやく完了します。

DB 領域の追加

領域の使用状況のモニタリングおよび DB 領域の追加の詳細については、「[ディスク領域の不足 \(537 ページ\)](#)」および「[ディスク領域の使用状況のモニタリング \(541 ページ\)](#)」を参照してください。

サーバ起動オプションの設定

プラットフォームによっては、コマンドライン・オプションを設定し使用可能なメモリ量を調整できます。メモリを増やすとロード・パフォーマンスを改善できます。パフォーマンスに影響するコマンドライン・オプションについては、「[第4章 接続パラメータと通信パラメータ](#)」を参照してください。

ロード時の環境の調整

データをロードするときいくつかの要素を調整すると、ロード・パフォーマンスが向上します。

- ASCII またはバイナリ・フォーマットで生データにアクセスし、特に数百以上のローをロードするような場合、**LOAD TABLE** コマンドを使用します。**LOAD TABLE** コマンドを使用すると、最も速く挿入できます。
- フラット・ファイルからロードするときに、バイナリ・データまたは文字データを選択できる場合、バイナリ・データを使用します。バイナリ・データを使用すると、変換処理が行われず、I/O が少なくなるためパフォーマンスが向上します。
- 「**LOAD TABLE 文を使ったデータのバルク・ロード**」の説明に従って、**LOAD TABLE** コマンド・オプションを適切に設定します。

無視された整合性制約違反をロギングする場合は、**LOAD TABLE** の **IGNORE CONSTRAINT** オプションの **limit** はゼロ以外の値に設定することを強くおすすめします。ロギングする違反の数が多すぎると、ロードのパフォーマンスに影響します。

- データ・ファイルとデータベース・ファイルを物理的に別のディスク・ドライブに置くと、ロード中のディスク・ヘッドの動きを削減できます。
- データベース・キャッシュ・サイズを大きくします。ロードのために十分なメモリを確保することは、パフォーマンスを向上させるための重要な要素です。コマンド・ライン・オプション **iqmc** と **iqtc** を使用して、キャッシュ・サイズを増やします。詳細については、『ユーティリティ・ガイド』の第 1 章の「**start iq の server options**」を参照してください。これらのオプションを有効にするには、サーバを再起動してください。
- ロック競合を避けるために、データベースまたはサーバのバッファ・パーティションの程度を調整します。CPU 数に基づくバッファ・パーティションはデフォルトで有効になっており、**-iqpartition** サーバ・コマンド・ライン・オプションか **Cache_Partitions** データベース・オプションを設定することで調整できます。詳細については、「**ロック競合の管理**」(412 ページ)を参照してください。
- 一度に 1 人のユーザのみがデータベースを更新するようにしてください。複数のユーザが同時に異なるテーブルにデータを挿入できますが、テーブルの更新が同時に行われると、サーバのパフォーマンスは低下します。
- 混雑していない時間帯に、主要な更新をスケジュールしてください。更新中に数多くのユーザがテーブルを照会できますが、このためには CPU サイクル、ディスク領域、メモリが必要となります。これらのリソースを確保することによって、さらに速く挿入を行うことができます。
- **INSERT** 文を使用している場合は、可能であれば、サーバと同じマシン上で **Interactive SQL** またはクライアント・アプリケーションを実行します。ネットワークを介してデータをロードすると、通信のために余分な時間がかかります。新しいデータをロードする場合は、混雑している時間帯を避けてください。

INSERT...LOCATION を使用して、リモートの Adaptive Server Enterprise データベースから大量のテキストまたはバルク・データをネットワーク経由でロードしている場合は、LOCATION 句の PACKETSIZ パラメータを使用して TDS パケット・サイズを大きくします。この変更により、ロード・パフォーマンスが大きく向上することがあります。INSERT 文の構文の詳細については、『リファレンス：文とオプション』の「第1章 SQL 文」を参照してください。

インクリメンタル・ロード時の IQ メイン・ストア領域使用の低減

インクリメンタル・ロードでは、ロードするテーブル内の多数のページが変更されることがあります。このため、トランザクションがコミットされ、チェックポイントによって以前のバージョンが解放されるまでメイン・データベース領域内でページが一時的にバージョン化されます。同じテーブルから削除した後インクリメンタル・ロードを行う場合に、このバージョン化がよく行われます。これは、デフォルトでは、Sybase IQ は削除されたレコードのロー ID を再利用するためです。

バージョン管理されたページからの領域使用を減らすには、APPEND_LOAD オプションを ON に設定し、IQ がテーブルの最後に新しいデータを付加できるようにします。APPEND_LOAD はデフォルトでは OFF です。

APPEND_LOAD オプションは、LOAD、INSERT...SELECT、INSERT...VALUES の各文に適用されます。

バージョン管理の詳細については、「第10章 トランザクションとバージョン管理」を参照してください。

ロード時のスレッド使用についての理解

Sybase IQ は、可能なときにはマルチスレッドを使用してロード・パフォーマンスを向上させます。

固定幅のロード、およびロー・デリミタのある完全幅のロード (size と limit が 0) は、リソース (メモリかスレッドまたはその両方) が十分にある場合には、完全にマルチスレッドで動作します。リソースが十分でない場合には、ロードはシングル・スレッドで動作し、.iqmsg ファイルに次のメッセージが出力されます。

```
The insert to the table will be single threaded.
```

ロー・デリミタのないデータの可変幅ロードは、これを実行できるだけの十分なリソースがある場合でも、部分的なマルチスレッド動作にとどまります。部分的なマルチスレッドのロードでは、.iqmsg ファイルに次のメッセージが出力されます。

```
Portions of the insert/load will be multithreaded.
```

UPDATE によるデータの更新

UPDATE 文を使用して、その後にテーブル名やビュー名を続けると、テーブルにある単一のロー、ローのグループ、またはテーブル内のすべてのローを変更できます。すべてのデータ修正文と同様に、一度に 1 つのテーブルまたはビュー内のデータしか更新できません。

UPDATE 文によって、変更するローと新しいデータを指定します。新しいデータは、指定する定数か式、または他のテーブルから引き出したデータです。

UPDATE 文が整合性制約に違反すると、更新は行われず、エラー・メッセージが表示されます。たとえば、追加された値の 1 つが誤ったデータ型であったり、カラムやデータ型のいずれかに定義された制約に違反した場合には、更新は実行されません。UPDATE 文のデータ型変換の詳細については、「[IQ におけるデータ変換](#)」の表 7-4 (311 ページ) を参照してください。

UPDATE 構文

完全な構文については、『リファレンス：文とオプション』の「[UPDATE 文](#)」を参照してください。UPDATE 構文の簡略バージョンは次のとおりです。

```
UPDATE table-name
SET column_name = expression
WHERE search-condition
```

デモ・データベースの Customers テーブルの Newton Ent. 社が Einstein, Inc. 社に吸収される場合は、次のような文を使用して会社の名前を更新できます。

```
UPDATE Customers
SET company_name = 'Einstein, Inc.'
WHERE company_name = 'Newton Ent.'
```

WHERE 句で条件を使用できます。入力された正式会社名が不明の場合、次のような文を使用して会社名に Newton と付くデータに更新を試みます。

```
UPDATE Customers
SET company_name = 'Einstein, Inc.'
WHERE company_name LIKE 'Newton%'
```

探索条件は更新されるカラムを参照する必要はありません。Newton Entertainments の会社 ID は 109 です。ID 値はテーブルのプライマリ・キーなので、次の文を使用して正確なローを確実に更新できます。

```
UPDATE Customers
SET company_name = 'Einstein, Inc.'
WHERE id = 109
```

SET 句

SET 句は更新されるカラムとその新しい値を指定します。WHERE 句は更新する必要があるローを決定します。WHERE 句がない場合、すべてのローの指定されたカラムが SET 句による値で更新されます。

SET 句には正しいデータ型の式をすべて使用できます。

WHERE 句 WHERE 句は更新されるローを指定します。たとえば、次の文は One Size Fits All Tee Shirt を Extra Large Tee Shirt に置換します。

```
UPDATE Products
SET size = 'Extra Large'
WHERE name = 'Tee Shirt'
AND size = 'One Size Fits All'
```

FROM 句 FROM 句を使用して、1 つ以上のテーブルから更新するテーブルにデータを引き出すことができます。また、FROM 句は、更新するローを制御する選択条件を他方のテーブルに対して適用するために使用することもできます。

データベースの削除

データベースからデータを削除するには、次のいずれかを行います。

- 指定した基準を満たすすべてのローをテーブルから削除するには、DELETE 文を使用する。
- すべてのデータのローを含むテーブル全体を削除するには、DROP TABLE 文を使用する。
- テーブル定義は削除しないでテーブルからすべてのローを削除するには、TRUNCATE TABLE 文を使用する。

これらの文の構文については、『リファレンス：文とオプション』の「[第1章 SQL 文](#)」を参照してください。

削除用の領域

DELETE または TRUNCATE TABLE 文を使用する場合、Sybase IQ がデータ・ページのバージョンを格納する方法によっては、データベースに領域を追加することが必要な場合があります。詳細については、「[重複したバージョンと削除](#)」を参照してください。

DROP TABLE を使用する場合は、追加のバージョン・ページは必要ないため、領域を追加する必要はありません。

この章について

データベースの各ユーザには、ユニークなユーザ ID を割り当てる必要があります。ユーザ ID とは、データベースに接続するときに入力する名前です。この章では、ユーザ ID を管理する方法について説明します。

内容

トピック名	ページ
データベースのパーミッションと権限の概要	333
ログイン管理	341
ユーザ・アカウントと接続の管理	342
ユーティリティ・データベース・サーバのセキュリティ	345
個別のユーザ ID とパーミッションの管理	347
グループの管理	356
データベース・オブジェクトの名前とプレフィクス	360
高度なセキュリティを実現するためのビューとプロシージャの使い方	362
ユーザ・パーミッションの評価方法	366
リソース接続使用の管理	367
システム・テーブルのユーザとパーミッション	368
トランスポート・レイヤ・セキュリティ	370

データベースのパーミッションと権限の概要

ユーザ ID とパーミッションを正しく管理することは、データ・ウェアハウスではとても重要です。これによりユーザは、データベース内の情報のセキュリティとプライバシーを管理しながら作業を効率的に処理できます。

SQL 文を使用して、データベースの新規ユーザに対するユーザ ID の割り当て、パーミッションの付与と取り消し、ユーザの現在のパーミッションの表示を行います。

「パーミッション」は、データベース・オブジェクト (テーブル、ビュー、ユーザなど) の作成、修正、クエリ、使用、または削除を許可するものです。「権限」は、データベース・レベルの作業 (データベースのバックアップなど) の実行を許可するものです。

パーミッションはユーザ ID に対して割り当てられます。この章では、「ユーザ」とユーザ ID は同義語として扱います。ただし、パーミッションの付与および取り消しは、常に各ユーザ ID に対して行います。

個別のユーザ ID の設定

マルチユーザ・データベースにおいて、セキュリティが問題とならない場合でも、各ユーザに対して個別のユーザ ID の設定が必要な場合もあります。グループを作成して適切なパーミッションを付与すれば、各ユーザ ID の管理作業にかかるオーバーヘッドが非常に少なくなります。ユーザのグループについては後ほど説明します。

個別にユーザ ID を設定する理由を次に示します。

- ネットワーク・サーバ画面と Sybase Central の接続リストでは、個別のユーザ ID があればどの接続がどのユーザであるかを知ることができます。
- バックアップ・ログでバックアップを作成したユーザ ID を識別します。
- メッセージ・ログには、各データベース接続のユーザ ID が表示されます。詳細については、『システム管理ガイド第 1 巻』の「第 1 章 Sybase IQ システム管理の概要」の「メッセージ・ロギング」を参照してください。

パーミッションは (ユーザが属するグループから) すべて継承できますが、権限は一部しか継承できません。

完全な管理権限を持つ DBA を除き、各権限には特定の種類の作業を実行するパーミッションがあります。詳細については、「[プロセスを使用したセキュリティの調整](#)」(364 ページ) を参照してください。

権限の継承

表 8-1 は、ユーザに割り当てることができる権限と、各権限がグループ・メンバシップを通じて継承可能かどうかを示します。

表 8-1: 権限

権限	グループ・メンバシップを通じた継承
BACKUP	不可
DBA	不可
MULTIPLEX ADMIN	可
OPERATOR	可
PERMS ADMIN	可
PROFILE	可
READCLIENTFILE	可
READFILE	可
REMOTE DBA	不可
RESOURCE	不可
SPACE ADMIN	可
USER ADMIN	可
VALIDATE	不可
WRITECLIENTFILE	可

MULTIPLY ADMIN 権限については、『Sybase IQ マルチプレックスの使用』の「第 2 章 マルチプレックス・サーバの管理」の「MULTIPLY ADMIN 権限の概要」を参照してください。

Sybase IQ が提供する権限の詳細については、「権限の概要」(335 ページ)を参照してください。

表 8-1 の一覧にある権限は SQL Anywhere によって提供されています。『SQL Anywhere サーバ - データベース管理』>「データベースの設定」>「ユーザ ID、権限、パーミッションの管理」>「データベースのパーミッションと権限の概要」を参照してください。

Sybase IQ 15.2 より前のバージョンで作成されたデータベースに対する権限 MULTIPLY ADMIN、OPERATOR、PERMS ADMIN、SPACE ADMIN、USER ADMIN の付与または取り消しを行うと、次のエラー・コード 1347 が返されます。

```
Use of authority '%1' is not supported on this current
database.
```

パーミッションの継承

表 8-2 は、ユーザに割り当てることができるパーミッションと、各パーミッションがグループ・メンバシップを通じて継承可能かどうかを示します。

表 8-2: パーミッション

パーミッション	グループ・メンバシップを通じた継承
ALL	可
ALTER	可
CONNECT	不可
DELETE	可
INSERT	可
INTEGRATED LOGIN	不可
KERBEROS LOGIN	不可
REFERENCES	可
SELECT	可
UPDATE	可

権限の概要

以下の各項では、権限について詳しく説明します。

DBA 権限の概要

データベースを作成すると、ユーザ ID が 1 つ作成されます。この最初のユーザ ID は DBA で、初期パスワードはデフォルトで `sql` に設定されます。デフォルトのユーザ名またはパスワードを上書きするには、`DBA USER` 句または `DBA PASSWORD` 句を含む `CREATE DATABASE` 文を使用します。ユーザ ID の DBA には、自動的に DBA パーミッション (DBA 権限ともいう) が付与されます。このレベルのパーミッションでは、DBA ユーザ ID がデータベースでのすべての作業 (テーブルの作成、テーブル構造の変更、新規ユーザ ID の作成、ユーザのパーミッションの取り消しなど) を実行できます。

注意 データベースのセキュリティを確保するために、DBA はパスワードをデフォルトの `sql` から別の値に変更する必要があります。

DBA 権限を持つユーザ

DBA 権限のあるユーザは、「データベース管理者」または「データベース所有者」と呼ばれます。この章では、データベース管理者、つまり *DBA* について頻繁に記述されています。これは、DBA 権限を持つ 1 人または複数のユーザという意味になります。

DBA 権限は他のユーザ ID に付与したり引き継いだりできますが、この章では DBA ユーザ ID はデータベース管理者であることを前提としています。また、省略形の *DBA* は、DBA ユーザ ID と DBA 権限を付与されたユーザ ID の両方を意味するものとして使用されています。

警告! マルチプレックス・データベースの DBA ユーザを削除しないでください。これを行うと、データベースが無効になります。

例

次の例は、DBA 権限が必要なコマンドを実行するパーミッションを非 DBA ユーザに付与する方法を示します。この例では、非 DBA ユーザ (`user1`) にバックアップの実行を許可するポリシーを作成します。

```
CREATE PROCEDURE "DBA".do_backup()
BEGIN
    BACKUP DATABASE
        CRC ON
        ATTENDED OFF
        BLOCK FACTOR 4
        FULL
        TO 'fileA' SIZE 2000
        TO 'fileB' SIZE 2000
        TO 'fileC' SIZE 2000
;
END;
GRANT EXECUTE ON "DBA".do_backup TO user1;
```

新しいユーザの追加

DBA は、データベースに新しいユーザを追加する権限を持っています。追加されたユーザは、データベース上でタスクを実行するためのパーミッションを付与されます。この中には、SQL クエリを使ってデータベース中の情報を参照するだけのユーザもいますし、データベースに情報を追加するユーザもいます。また、データベースの構造そのものを変更するユーザもいます。DBA の責任を他のユーザ ID に分散することは多少できますが、データベース全体の管理は DBA 権限を持つ DBA の責任です。

DBA はデータベース・オブジェクトを作成して、他のユーザ ID に所有権を割り当てることができます。

データベース・オブジェクトを作成するコマンドの構文については、『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』の「[第 2 章 SQL 言語の要素](#)」を参照してください。

大文字と小文字を区別するデータベースでの DBA ユーザ ID

ユーザ ID とパスワードは、実際にはデータベース内のオブジェクトです。パスワードの大文字と小文字の区別の詳細については、『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「[付録 A 他の Sybase データベースとの互換性](#)」の「[ユーザ ID とパスワード](#)」を参照してください。

OPERATOR 権限の概要

OPERATOR 権限には、データベースのチェックポイントの設定、接続の削除 (DBA 権限を持つユーザの接続を含む)、データベースのバックアップ、およびシステムのモニタを行うパーミッションがあります。

OPERATOR 権限を他のユーザに付与できるのは、DBA 権限または PERMS ADMIN 権限を持つユーザだけです。DBA 権限のないユーザが BACKUP、CHECKPOINT、または DROP CONNECTION コマンドを実行するには、OPERATOR 権限が必要です。

PERMS ADMIN 権限の概要

PERMS ADMIN 権限は、データのパーミッション、グループ、権限、パスワードを管理するパーミッションです。

次のことを行うには、PERMS ADMIN 権限または DBA 権限が必要です。

- 該当するユーザ・オブジェクト (テーブル、ビュー、カラムなど) に対する SELECT、INSERT、UPDATE、DELETE、ALTER、または REFERENCES パーミッションの付与または取り消し。
- 該当するユーザ・オブジェクト (プロシージャや関数など) に対する EXECUTE パーミッションの付与または取り消し。
- グループの作成と削除、およびグループ・メンバシップの付与または取り消し。

- DBA と REMOTE DBA を除く、既存および新規のすべての権限の付与または取り消し。

アカウントのパスワードの変更。

PERMS ADMIN 権限を他のユーザに付与できるのは、DBA 権限または PERMS ADMIN 権限を持つユーザだけです。

RESOURCE 権限の概要

RESOURCE 権限とは、テーブル、ビュー、ストアド・プロシージャなどのデータベース・オブジェクトを作成するパーミッションのことです。RESOURCE 権限を他のユーザに付与できるのは、DBA 権限または PERMS ADMIN 権限を持つユーザだけです。

SPACE ADMIN 権限の概要

SPACE ADMIN 権限は、DB 領域を管理するパーミッションです。

次のことを行うには、SPACE ADMIN 権限が必要です。

- DB 領域の作成、変更、削除、コメント追加
- DB 領域に対する CREATE パーミッションの付与または取り消し
- DB 領域へのデータおよびメタデータの移動
- 読み取り専用の選択的リストア操作の実行 ([「リストア前の作業」\(465 ページ\)](#) を参照)

注意 SPACE ADMIN 権限では DB 領域を作成できます。ただし、DB 領域にオブジェクトを作成するには、その DB 領域に対する CREATE パーミッションをユーザに付与する必要があります。ユーザが SPACE ADMIN 権限と RESOURCE 権限の両方を持っている場合でも、DB 領域にオブジェクトを作成するには、その DB 領域に対する CREATE 権限を付与する必要があります。CREATE 権限がない状態でオブジェクトを作成しようとすると、Permission Denied エラーが返されます。CREATE 権限を追加するには、GRANT CREATE ON を使用します。

SPACE ADMIN 権限を他のユーザに付与できるのは、DBA 権限または PERMS ADMIN 権限を持つユーザだけです。

USER ADMIN 権限の概要

USER ADMIN 権限は、ユーザ、外部ログイン、ログイン・ポリシーを管理するパーミッションです。

次のことを行うには、USER ADMIN 権限が必要です。

- データベース・ユーザの作成と削除、および初期パスワードの割り当て
- ログイン・ポリシーの作成、変更、または削除、およびユーザへのログイン・ポリシーの割り当て
- ユーザ認証メカニズム (Kerberos や統合化ログインなど) の定義
- 外部ログインの作成または削除
- ユーザに対する次回ログイン時のパスワード変更の強制
- ユーザ・ログイン・ポリシーのリセット

USER ADMIN 権限を他のユーザに付与できるのは、DBA 権限または PERMS ADMIN 権限を持つユーザだけです。

所有権パーミッションの概要

データベース・オブジェクトの作成者は、そのオブジェクトの所有者になります。データベース・オブジェクトの所有権は、すなわちそのオブジェクトに対してアクションを実行するパーミッションになります。これらのパーミッションは、この章に出てくる他のパーミッションのようにユーザに割り当てられることはありません。

所有者

データベース内に新しくオブジェクトを作成したユーザは、そのオブジェクトの「所有者」と呼ばれます。所有者には、そのオブジェクトに対してすべての操作を行うパーミッションが自動的に付与されます。たとえば、テーブルの所有者はそのテーブルの構造を変更したり、他のユーザにテーブルのデータを更新するパーミッションを付与したりできます。

注意 テーブルの所有者がテーブルをロードできるのは、所有者が DBA である場合か、コマンド・ラインまたは設定ファイルで `-gl all` スイッチを指定してサーバを起動した場合だけです。所有権と RESOURCE 権限だけでは、LOAD TABLE は使用できません。LOAD TABLE 文を使用するためには、テーブルに対する INSERT パーミッションも必要です。

DBA は、データベース内のすべてのコンポーネントを修正するパーミッションを持っています。したがって、たとえば他のユーザが作成したテーブルを削除することもできます。DBA は、各オブジェクトの所有者が持つデータベース・オブジェクトに関するすべてのパーミッションを持ちます。

また DBA は、他のユーザのためにデータベース・オブジェクトを作成できます。この場合、オブジェクトの所有者は CREATE 文を実行したユーザ ID ではありません。この機能の使い方については、「[パスワードのないグループ](#)」(359 ページ) を参照してください。ここではデータベース・オブジェクトの所有者と作成者を同様に扱います。

DB 領域管理パーミッション

DB 領域でテーブル・オブジェクトまたはジョイン・インデックスを作成したり、新しい DB 領域にテーブル・オブジェクトまたはジョイン・インデックスを移動したりするには、DB 領域の CREATE パーミッションが必要です。DB 領域の CREATE パーミッションは、ユーザまたはグループ間で付与したり取り消したりできます。グループのメンバは、そのグループの CREATE パーミッションを継承します。デフォルトでは、IQ_SYSTEM_MAIN、IQ_SYSTEM_TEMP、および SYSTEM の CREATE パーミッションは PUBLIC に付与されます。他の IQ メイン DB 領域では、システム管理者がグループ／ユーザに DB 領域の CREATE パーミッションを明示的に付与してから、グループ／ユーザがその DB 領域にオブジェクトを作成または移動する必要があります。たとえば、テーブルを新しい IQ メイン DB 領域に配置する場合、ユーザはその DB 領域の CREATE パーミッションを持っている必要があります。また、オブジェクトを作成するには RESOURCE パーミッションも必要です。

テーブルとビューのパーミッションの概要

テーブルおよびビューに関するユーザ ID に付与されるパーミッションを次に示します。

パーミッション	説明
ALTER	テーブルの構造を変更できます。
DELETE	テーブルまたはビューからローを削除できます。
INSERT	テーブルまたはビューにローを追加できます。
REFERENCES	テーブルにインデックスを作成し、そのテーブルを参照する外部キーを作成できます。
SELECT	テーブルまたはビューの情報を参照できます。
UPDATE	テーブルまたはビューのローを更新できます。テーブル内のあるカラムのセットに関してのみ、このパーミッションを付与することもできます。
ALL	上記のすべてを実行可能。

マルチプレックスでは、書き込みサーバが所有するテーブルのテーブル・パーミッションを変更できるのは書き込みサーバだけです。

グループ・パーミッションの概要

各ユーザに対して個別にパーミッションを設定するのは時間がかかり、またエラーの原因になります。ほとんどのデータベースでは、グループ単位のパーミッション管理の方が、ユーザ ID 単位の管理よりはるかに効率的です。

グループに対してパーミッションを割り当てる方法は、個別のユーザに対する方法とまったく同じです。各新規ユーザに対してそのグループのメンバシップを割り当てれば、メンバシップに関連する一連のパーミッションを付与できます。

例 たとえば、さまざまな部署のグループ (sales、marketing など) を社内データベースに作成し、これらのグループにパーミッションを割り当てることができます。各営業部員は sales グループのメンバになり、自動的にデータベースの適切な領域にアクセスできるようになります。

ユーザ ID は複数のグループのメンバとなることができ、各グループのパーミッションをすべて継承します。

サーバのコマンド・ラインのパーミッション・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、データベースとサーバのいくつかの機能のパーミッション・レベルを設定するオプションがあります。表 8-3 にオプションのリストを示します。

表 8-3: パーミッションに影響を与える起動オプション

オプション	説明	指定できる値	デフォルト
<code>-gd level</code>	データベースを起動するためのパーミッションを設定する。	DBA、ALL、NONE	DBA
<code>-gk level</code>	サーバを停止するためのパーミッションを設定する。	DBA、ALL、NONE	DBA
<code>-gl level</code>	データをロードするためのパーミッションを設定する。	DBA、ALL、NONE	<code>start_iq</code> で起動されたサーバは ALL、他のサーバは DBA
<code>-gu level</code>	CREATE DATABASE や DROP DATABASE など、ユーティリティ・コマンドを実行するためのパーミッションを設定する。	DBA、ALL、NONE、UTILITY_DB	ALL

これらのオプション、およびパーミッション・レベルの値とデフォルトの詳細については、『システム管理ガイド第 1 巻』の「第 2 章 Sybase IQ の実行」の「コマンド・ラインからのパーミッションの制御」、および『ユーティリティ・ガイド』の「第 1 章 データベース・サーバの実行」を参照してください。

ログイン管理

Sybase IQ では、データベース・オブジェクト内のユーザのデータベース接続を確立するときに従う必要のある規則が定義されています。これをログイン・ポリシーと呼びます。ログイン・ポリシーは、オプションのセットから構成されるデータベース内の名前付きオブジェクトです。各ログイン・ポリシーは、ログイン・ポリシー・オプションと呼ばれるオプションのセットに関連付けられています。詳細については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の表 1-5 を参照してください。

新しいログイン・ポリシーを作成したり、既存のログイン・ポリシーをユーザに割り当てたりするには、DBA 権限または USER ADMIN 権限が必要です。ログイン・ポリシーはユーザ・グループの階層で継承することはできません。ポリシーを管理する SQL コマンドの構文については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「ALTER LOGIN POLICY 文」、「ALTER USER 文」、「CREATE LOGIN POLICY 文」、「DROP LOGIN POLICY 文」を参照してください。

新しい各データベースは、root ポリシーと呼ばれるデフォルトのログイン・ポリシーで作成されます。root ログイン・ポリシーのオプション値は変更できませんが、ポリシーは削除できません。ログイン・ポリシーを指定しないでユーザ・アカウントを作成した場合、ユーザは root ログイン・ポリシーの一部になります。ログイン・ポリシーを作成したときに明示的に設定されなかったオプションは、root ポリシーからその値を継承します。

『SQL Anywhere サーバー データベース管理』>「データベースの起動とデータベースへの接続」>「接続パラメータとネットワーク・プロトコル・オプション」>「ユーザ ID、権限、パーミッションの管理」>「ログイン・ポリシーの管理の概要」を参照してください。

ログイン管理コマンドは、どのマルチプレックス・サーバ上でも実行できます。コマンドは、マルチプレックス内のすべてのサーバに自動的に伝達されず、最高のパフォーマンスを実現するために、これらのコマンドまたは DDL をコーディネータで実行することをおすすめします。詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。

12.7 から Sybase IQ 15.2 にデータベースを移行すると、既存のログイン管理設定が削除されます。移行後にログイン管理設定を再作成するには、次の項の SQL 構文を使用します。

ユーザ・アカウントと接続の管理

Sybase IQ ログイン管理機能は、ユーザとデータベース接続を管理するのに役立ちます。

DBA 権限または USER ADMIN 権限を持つユーザは、次の方法でユーザを追加または削除したり、接続を制御したりできます。

- 単一ユーザのアクティブなログインの数を制限する。

ログインの数を制限するには、ユーザを **max_connections** ログイン・ポリシー・オプションを指定したログイン・ポリシーに割り当てます。

- ユーザをロックアウトする。

ロックアウトするには、ユーザを **locked** オプションを ON に設定したログイン・ポリシーに割り当てます。

- ユーザ・パスワードの有効期間を設定する。

有効期間を設定するには、ログイン・ポリシーに `max_failed_login_attempts` および `max_days_since_login` を指定し、このポリシーにユーザを割り当てます。また、SQL 文 `CREATE/ALTER USER` の `FORCE PASSWORD CHANGE` 句を使用して、ユーザ・パスワードの有効期限を明示的に設定することもできます。

ポリシーを管理する SQL コマンドの構文については、『リファレンス：文とオプション』の「`CREATE LOGIN POLICY`」、「`ALTER LOGIN POLICY`」、「`DROP LOGIN POLICY`」を参照してください。

新しい各データベースには、`root` という名前のログイン・ポリシーがあります。`root` ログイン・ポリシーのオプション値は変更できますが、ポリシーは削除できません。ログイン・ポリシーを指定しないでユーザ・アカウントを作成した場合、ユーザは `root` ログイン・ポリシーの一部になります。

表 8-4 は、Sybase IQ ログイン管理の各機能を実行するときに呼び出すプロシージャを示します。

表 8-4: ログイン管理用のストアド・プロシージャ

ストアド・プロシージャ	目的	必要な権限
<code>sa_get_user_status</code>	既存のユーザすべての現在のステータスを取得する。	DBA または USER ADMIN
<code>sp_expireallpasswords</code>	すべてのユーザ・パスワードをすぐに期限切れにする。	DBA または USER ADMIN
<code>sp_iqaddlogin</code>	次回ログイン時に、ユーザの追加、ユーザ・パスワードの定義、ログイン・ポリシーとパスワードの有効期限の指定を行う。	DBA
<code>sp_iqcopyloginpolicy</code>	既存のログイン・ポリシーをコピーして、新しいログイン・ポリシーを作成する。	DBA
<code>sp_iqdroplogin</code>	指定されたユーザを削除する。	DBA
<code>sp_iqmodify</code>	指定されたユーザにログイン・ポリシーを割り当てる。	DBA
<code>sp_iqmodifyadmin</code>	名前付きログイン・ポリシーのオプションを所定の値に設定する。	DBA
<code>sp_iqpassword</code>	ユーザのパスワードを変更する。	他のユーザのパスワードを変更する場合は DBA。 すべてのユーザが <code>sp_iqpassword</code> を実行して各自のパスワードを変更できる。

Sybase IQ ログイン管理プロシージャの詳細については、『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「[第 7 章 システム・プロシージャ](#)」を参照してください。

Sybase IQ ログイン管理で使用されるシステム・テーブルについては、『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「[第 8 章 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。

ログイン要求失敗後の接続の阻止

次の例は、ユーザが5回ログイン要求に失敗したら、それ以降は接続できないようにする方法を示します。

まず、ログイン・ポリシー・オプション `max_failed_login_attempts` を値 5 に設定したログイン・ポリシー `lp` を作成します。

```
CREATE LOGIN POLICY lp max_failed_login_attempts=5;
```

ログイン・ポリシー `lp` に属するユーザ `John` を作成します。

```
CREATE USER john IDENTIFIED BY j345 LOGIN POLICY lp;
```

`John` は `max_failed_login_attempts=5` の `lp` ログイン・ポリシーに属するため、このユーザは `max_failed_login_attempts` の制限 5 回を超えると、IQ サーバにログインできなくなります。

ユーザのロックアウト

ロックアウトするユーザのログイン・ポリシーを明確にすることで、ユーザをロックアウトできます。次のように、ログイン・ポリシー `locked=ON` であるログイン・ポリシーを作成します。

```
CREATE LOGIN POLICY locked_users locked=ON
```

ロックアウトするユーザを `locked_users` ポリシーに割り当てます。次に例を示します。

```
ALTER USER john LOGIN POLICY locked_users
```

ユーザのロックアウト解除

前のセクションでは、`locked=ON` のログイン・ポリシーに属するユーザがロックアウトされました。このようなユーザのロックアウトを解除するには、そのユーザをオプション `locked=OFF` のログイン・ポリシーに割り当てます。

また、ユーザは `max_failed_login_attempts` または `max_days_since_login` を超えた場合にもロックアウトされます。このようなユーザのロックアウトを解除するには、次のような文を使用します。

```
ALTER USER john RESET LOGIN POLICY
```

この文は、`ALTER USER userid LOGIN POLICY current-policy-for-user` とセマンティック的に等価です。

ユーティリティ・データベース・サーバのセキュリティ

Sybase IQ には、「ユーティリティ・データベース」というものがあります。これは、物理的な存在を持たない、幻データベースです。このデータベースには、データベース・ファイルはなく、データの格納もできません。ユーティリティ・データベースは、任意の Sybase IQ サーバで実行できます。

ユーティリティ・データベースは、限られた機能に特化されています。これは、CREATE DATABASE や DROP DATABASE などのデータベース・ファイル操作文を、物理データベースに接続せずに実行できるようにするために用意されているものです。また、ユーティリティ・データベースからは、データベース・プロパティと接続プロパティも取得できます。これらのプロパティは、ユーティリティ・データベースに接続しているときに作成するデータベースに適用されます。Sybase Central では、ユーティリティ・データベースのサーバはユーティリティ・サーバといいます。

ユーティリティ・データベースを起動するには、接続時にデータベース名として `utility_db` を指定します (“utility_db” をデータベース・ファイルとして指定しないでください。ユーティリティ・データベースにはデータベース・ファイルは関連付けられていないからです)。例を示します。

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

注意 ユーティリティ・データベースに接続して Windows のロー・パーティションを持つ IQ データベースを作成するときには、IQ PATH の構文の違いに注意してください。たとえば、デバイス I の Windows ロウ・パーティションを指定する場合、ユーティリティ・データベースでは “`¥¥¥¥I:`” と指定します。他の IQ データベースでは、円記号を二重にする必要があります。つまり、この同じデバイスを “`¥¥¥¥.¥¥¥¥I:`” と指定します。円記号は、IQ データベースではエスケープ文字として扱われるのに対し、ユーティリティ・データベースでは通常の文字として扱われます。

設定作業の 1 つとして、ユーティリティ・データベースとそのサーバのセキュリティの設定があります。ユーティリティ・データベース・サーバのセキュリティには次の 2 つの側面があります。

- ユーティリティ・データベースに接続できるのはどのユーザか
- ファイル管理文を実行できるのはどのユーザか

この項では、これらの内容について説明します。

ユーティリティ・データベースのパスワードの定義

ユーティリティ・データベースを使用するには、ユーザ ID DBA を指定する必要があります。ユーティリティ・データベースのパスワードは、`util_db.ini` というファイルに保持されています。このファイルは、サーバの実行ファイルと同じディレクトリにあります。このディレクトリはサーバ上にあるので、このファイルへのアクセスは制御できます。つまり、どのユーザがパスワードにアクセスできるかを制御できます。

`util_db.ini` ファイルには次の内容が記述されています。

```
[UTILITY_DB]
PWD=password
```

`utility_db` セキュリティ・レベルの使用は、データベース・サーバをホストするコンピュータの物理的なセキュリティに依存します。`util_db.ini` ファイルはテキスト・エディタで簡単に参照できるからです。

ファイル管理文を実行するためのパーミッション

データベースのセキュリティを高めるために、データベースの作成と削除は別個のレベルのセキュリティで制御されています。データベース・サーバのコマンドライン・オプションである `-gu` を使用して、どのユーザがファイル管理文を実行できるかを制御できます。

ファイル管理文の使用には、4つのレベルのパーミッションがあります。具体的には、`all`、`none`、`DBA`、`utility_db` の4つです。`utility_db` レベルでは、ユーティリティ・データベースに接続できるユーザだけがファイル管理文を使用できます。

表 8-5: ファイル管理のパーミッション

-gu スイッチの値	結果	適用対象
<code>all</code>	誰でもファイル管理文を実行できる	ユーティリティ・データベースを含む任意のデータベース
<code>none</code>	誰もファイル管理文を実行できない	ユーティリティ・データベースを含む任意のデータベース
<code>DBA</code>	DBA 権限を持つユーザだけがファイル管理文を実行できる	ユーティリティ・データベースを含む任意のデータベース
<code>utility_db</code>	ユーティリティ・データベースに接続できるユーザだけがファイル管理文を実行できる	ユーティリティ・データベースのみ

例

Sun、HP、Linux、Windows の各プラットフォームで、ユーティリティ・データベースへの接続とデータベースの作成または削除を許可する対象を、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限定するには、コマンド・ラインから次のコマンドでサーバを起動します。

```
start_iq -n testsrv -gu utility_db
```

AIX で、ユーティリティ・データベースへの接続とデータベースの作成または削除を許可する対象を、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限定するには、コマンド・ラインから次のコマンドでサーバを起動します。

```
start_iq -n testsrv -gu utility_db -iqmt 256
```

ユーティリティ・データベースのパスワードが **IQ&Mine49** へのインストール時に設定されていたものとする、次のコマンドにより、Interactive SQL ユーティリティがクライアント・アプリケーションとして起動し、**testsrv** というサーバに接続し、ユーティリティ・データベースをロードして、ユーザを接続します。

```
dbisql -c "uid=DBA;pwd=IQ&Mine49;dbn=utility_db;eng=testsrv"
```

この文を実行すると、ユーティリティ・データベースに正常に接続されます。これで、データベースの作成と削除ができるようになります。

注意 データベース名、ユーザ ID、パスワードは、大文字と小文字が区別されます。**dbisql** コマンドと *util_db.ini* ファイルで、大文字と小文字を一致させてください。

個別のユーザ ID とパーミッションの管理

この項では、ユーザにパーミッションを付与する方法、および Interactive SQL と Sybase Central を使用して新しいユーザを作成する方法について説明します。ほとんどのデータベースでは、パーミッション管理の大半を個別のユーザにパーミッションを割り当てて行うのではなく、グループで行います。ただし、グループは単に特別なプロパティが付加されたユーザ ID にすぎないので、グループ管理に関する説明に移る前に、ここでの説明をお読みください。

IQ スタアド・プロシ
ージャを使用したユーザ
管理

新しいユーザの作成には、IQ システム・プロシージャも使用できます。ユーザを Sybase IQ ログイン管理機能で管理するには、これらのプロシージャを使用してユーザの追加、パスワードやその他のログイン機能の修正を行う必要があります。

Sybase IQ ログイン管理でユーザを追加または変更するには、「[ユーザ・アカウントと接続の管理](#)」(342 ページ) で説明したシステム・プロシージャを使用します。

ASE ストアド・プロシージャを使用したユーザ管理

なお、データベース・オブジェクトに対するパーミッションをユーザに付与するには、この項で説明するコマンドとプロシージャを使用する必要があります。

この章では、Interactive SQL と Sybase Central を使用してユーザとグループを管理する方法について説明します。また、Adaptive Server Enterprise 互換のストアド・プロシージャを使用して、同様なタスクを数多く実行できます。Adaptive Server Enterprise またはバージョン 12.0 より前の Sybase IQ をすでに使用している場合は、そのストアド・プロシージャも使用できます。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、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」を参照してください。ASE ストアド・プロシージャの場合は、Sybase IQ ログイン管理機能を使用して接続を制限することはできません。

新規ユーザの作成

DBA または USER ADMIN 権限を持つユーザは、CREATE USER 文を使用して新しいユーザを作成できます。例を示します。

❖ データベースに新規ユーザを追加するには

この例では、ユーザ ID が M_Haneef でパスワードが *welcome* の新規ユーザをデータベースに追加します。

- 1 Interactive SQL から、DBA 権限または USER ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を発行します。

```
CREATE USER M_Haneef  
IDENTIFIED BY welcome
```

新規ユーザに対する初期パーミッション

デフォルトでは、新規ユーザにはデータベースに接続してシステム・テーブルを参照するためのパーミッションしか割り当てられていません。データベース内のテーブルにアクセスするには、パーミッションを割り当てる必要があります。

DBA は、特殊なグループである PUBLIC ユーザ・グループに対してパーミッションを割り当てることで、新規ユーザに自動的に付与されるパーミッションを設定できます。この方法については、「[特殊なグループ](#)」(359 ページ)を参照してください。

dbisql コマンド・ファイルを使用した新規ユーザの設定

新規ユーザを設定するコマンドを dbisql コマンド・ファイルに入力できます。コマンド・ファイルによって、繰り返し実行する処理を標準化できます。コマンド・ファイルの使用の詳細については、『ユーティリティ・ガイド』の「[第 2 章 Interactive SQL \(dbisql\) の使用](#)」を参照してください。

Sybase Central でユーザを作成する

❖ Sybase Central でユーザを作成するには

- 1 データベースに接続します。
- 2 [ユーザとグループ] フォルダを右クリックします。
- 3 [新規]-[ユーザ] を選択します。ユーザ作成ウィザードに従って操作を行います。

詳細については、IQ プラグインに関する Sybase Central オンライン・ヘルプを参照してください。

パスワードの変更

ユーザ・パスワードの変更

DBA 権限または PERMS ADMIN 権限がある場合は、次のコマンドを使用して既存のユーザのパスワードを変更できます。

```
ALTER USER userid IDENTIFIED BY password
```

新しいユーザを追加するときに既存のユーザのユーザ ID を誤って入力すると、その既存ユーザのパスワードを変更することになります。これは正常な動作とみなされるため、警告は発生しません。この動作は、バージョン 12 より前の Sybase IQ とは異なっています。

このような状況を回避するには、`sp_iqaddlogin` システム・プロシージャまたは `CREATE USER` を使用してユーザを追加します。`sp_addlogin` および `sp_adduser` システム・プロシージャを使用することもできます。Adaptive Server Enterprise やバージョン 12 より前の Sybase IQ の場合と同様に、これらのプロシージャでは、既存のユーザ ID を追加しようとするとエラーが発生します。

パスワード・ルールの実装

パスワード・ルールを設定し、割り当てられる新しいパスワードがそれらのルールに適合しているかどうかを確認できます。たとえば、パスワードに数字を 1 つ含めることや、ユーザ ID を使用してはならないことをルールに指定できます。詳細については、『リファレンス：文とオプション』の「[VERIFY_PASSWORD_FUNCTION オプション](#)」を参照してください。

最短のパスワード長を設定する方法については、『リファレンス：文とオプション』の「[MIN_PASSWORD_LENGTH オプション](#)」を参照してください。

DBA パスワードの変更

ユーザ ID DBA は、すべての管理権限とリソース作成権限を持ちます。ユーザ ID DBA のデフォルト・パスワードは、すべてのデータベースにおいて `sql` です。データベースに対する不正なアクセスを防ぐために、このパスワードを変更してください。次にユーザ ID DBA のパスワードを `new_password` に変更するコマンドを示します。

```
ALTER USER DBA
IDENTIFIED BY new_password
```

DBA パスワードを変更するには、DBA 権限または PERMS ADMIN 権限が必要です。

dbisql を使用している場合は、付与されたパーミッションを参照できるようにコマンド・ファイルに入力することをおすすめします。これにより、パーミッションを再作成する必要性が生じた場合に、そのファイルを修正し再度実行できます。

権限の付与

ほとんどの権限は同じ方法で付与されます。DBA 権限はすべての権限を付与できます。PERMS ADMIN 権限は、DBA と REMOTE DBA 以外のすべての権限を付与できます。

❖ ユーザ ID に RESOURCE パーミッションを付与するには

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT RESOURCE TO userid
```

❖ ユーザ ID に DBA パーミッションを付与するには

- 1 DBA 権限のあるユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT DBA TO userid
```

❖ ユーザ ID に OPERATOR パーミッションを付与するには

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT OPERATOR TO userid
```

❖ ユーザ ID に PERMS ADMIN パーミッションを付与するには

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT PERMS ADMIN TO userid
```

❖ ユーザ ID に RESOURCE パーミッションを付与するには

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT RESOURCE TO userid
```

❖ ユーザ ID に SPACE ADMIN パーミッションを付与するには

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT SPACE ADMIN TO userid
```

❖ ユーザ ID に USER ADMIN パーミッションを付与するには

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT USER ADMIN TO userid
```

注意事項

- DBA 権限をデータベース・ユーザに付与できるのは DBA だけです。
- DBA 権限は非常に強力です。データベースに対するすべてのアクションの実行と、データベース中のすべての情報へのアクセスを付与する権限を持っています。DBA 権限はごく少数のユーザだけに持たせるように制限してください。
- DBA 権限を持つユーザには、DBA 権限があるユーザ ID と権限がないユーザ ID の 2 つのユーザ ID を提供してください。これにより、ユーザは必要な場合にのみ DBA として接続できます。
- RESOURCE 権限は、テーブル、ビュー、インデックス、プロシージャなどの新規データベース・オブジェクトの作成を、ユーザに許可します。

テーブルおよびビューに対するパーミッションの付与

個々のテーブルやビューにパーミッションのセットを割り当てることができません。ユーザにこれらのパーミッションの組み合わせを付与して、テーブルまたはビューへのアクセスを定義します。

パーミッションの組み合わせ

- ALTER パーミッション (テーブル構造を変更できるパーミッション) と REFERENCES パーミッション (インデックスおよび外部キーを作成できるパーミッション) は、データベース・スキーマを修正できる権限を付与します。したがって、この組み合わせを割り当てるユーザを制限してください。これらのパーミッションはビューには適用されません。

- DELETE、INSERT、UPDATE パーミッションは、テーブルまたはビューのデータを変更する権限を与えます。この中で UPDATE パーミッションは、対象をテーブルまたはビュー内のカラムのセットに限定できます。
- SELECT パーミッションは、テーブルまたはビューのデータを参照する権限を与えますが、変更はできません。
- ALL パーミッションは、上記のすべてのパーミッションを付与します。

例 テーブルとビューに対するパーミッションはすべて、非常に似た方法で付与されます。たとえば、`sample_table` というテーブルからローを削除するパーミッションを `M_Haneef` に付与するには、次の手順に従います。

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザか、`sample_table` の所有者としてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT DELETE
ON sample_table
TO M_Haneef
```

`sample_table` というテーブルのカラム `column_1` と `column_2` だけを更新するパーミッションを `M_Haneef` に付与するには、次の手順に従います。

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザか、`sample_table` の所有者としてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT UPDATE (column_1, column_2)
ON sample_table
TO M_Haneef
```

テーブルとビューのパーミッションのうち UPDATE パーミッション以外には、テーブルまたはビューのすべてのデータに適用されるという制約があります。より細かいパーミッションを設定するには、テーブルに対してアクションを実行するプロシージャを作成し、そのプロシージャを実行するパーミッションをユーザに付与します。

Sybase Central でテーブルに対するパーミッションをユーザに付与する

Sybase Central で、プロシージャに対してユーザ・パーミッションを付与するには、次の手順に従います。

❖ Sybase Central でテーブルに対するパーミッションをユーザに付与する

- 1 データベースに接続します。
- 2 そのデータベースの [テーブル] フォルダをダブルクリックして、左パネルにテーブルを表示します。
- 3 テーブルをマウスの右ボタンでクリックし、ポップアップ・メニューから

- 4 [プロパティ] ダイアログの [パーミッション] タブで、テーブルのパーミッションを次のように設定します。
 - [Grant] をクリックして、完全なパーミッションを付与するユーザまたはグループを選択します。
 - ユーザまたはグループの横のフィールドをクリックし、特定のパーミッションを設定します。パーミッションはチェック・マークで示され、付与オプションは 2 つの '+' 記号の付いたチェック・マークで示されます。
 - ユーザを選択し、[参照]、[選択]、または[更新]の横のボタンをクリックして、その種類のパーミッションを個々のカラムに対して設定します。
 - リストのユーザまたはグループを選択し、[Revoke] をクリックして、すべてのパーミッションを取り消します。

また、[ユーザ & グループ] プロパティ・シートからもパーミッションを割り当てることができます。多数のユーザとグループに対してパーミッションをまとめて割り当てる場合は、テーブルのプロパティ・シートを使用します。多数のテーブルに対してパーミッションをまとめて割り当てる場合は、ユーザのプロパティ・シートを使用します。

パーミッションを付与する権利をユーザに付与する

「[テーブルおよびビューに対するパーミッションの付与](#) (351 ページ) で説明したテーブルとビューに対する各パーミッションには、WITH GRANT OPTION を割り当てることができます。このオプションはパーミッションを他のユーザに引き継ぐ権利を与えます。グループでのこの機能については、「[グループのパーミッション](#)」 (358 ページ) を参照してください。

例

sample_table というテーブルからローを削除するパーミッションと、このパーミッションを他のユーザに引き継ぐ権利を M_Haneef に付与するには、次の手順に従います。

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザか、sample_table の所有者としてデータベースに接続します。
- 2 次の SQL 文を入力して実行します。

```
GRANT DELETE ON sample_table  
TO M_Haneef  
WITH GRANT OPTION
```

プロシージャに対するパーミッションの付与

プロシージャに関して付与できるパーミッションは、プロシージャを実行する (呼び出す) ための EXECUTE パーミッションだけです。

ストアド・プロシージャを実行するパーミッションは、DBA または PERMS ADMIN か、プロシージャの所有者 (そのプロシージャを作成したユーザ ID) が付与できます。

プロシージャを実行するパーミッションを与える方法は、「[テーブルおよびビューに対するパーミッションの付与](#)」(351 ページ) で説明したテーブルおよびビューに対するパーミッションを付与する方法と似ています。

例 my_procedure というプロシージャを実行するパーミッションを M_Haneef に付与するには、次の手順に従います。

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザか、my_procedure プロシージャの所有者としてデータベースに接続します。
- 2 次の SQL 文を実行します。

```
GRANT EXECUTE
ON my_procedure
TO M_Haneef
```

プロシージャの実行パー
ミッション

プロシージャは所有者のパーミッションのもとで実行されます。テーブルのデータを更新するプロシージャは、そのプロシージャの所有者が対象となるテーブルの UPDATE パーミッションを持っていないと実行できません。

プロシージャの所有者が適切なパーミッションを持っているれば、そのプロシージャ実行のパーミッションを持ったユーザは、基本となるテーブルに対するパーミッションの有無にかかわらず、そのプロシージャを実行できます。プロシージャは、テーブルに対するパーミッションなしで、ユーザがテーブルに一定の作業を行うことを許可するのに使えます。

Sybase Central でプロ
シージャに対するパー
ミッションをユーザに付
与する

Sybase Central で、プロシージャに対してユーザ・パーミッションを付与するには、次の手順に従います。

❖ Sybase Central でプロシージャに対するパーミッションをユーザに付与する

- 1 データベースに接続します。
- 2 [ユーザ & グループ] フォルダをクリックして、パーミッションを付与するユーザを検索します。
- 3 ユーザをマウスの右ボタンでクリックし、ポップアップ・メニューから [コピー] を選択します。
- 4 [ストアド・プロシージャ] フォルダからユーザに実行を許可するプロシージャを検索します。
- 5 プロシージャをクリックし、メイン・メニューの [編集]-[貼り付け] を選択して、パーミッションを付与します。

詳細については、Sybase Central のオンライン・ヘルプを参照してください。

ユーザ・パーミッションの取り消し

ユーザのパーミッションは、実際には付与されたパーミッションと取り消されたパーミッションの組み合わせです。パーミッションの付与と取り消しを使って、データベースのユーザ・パーミッションを管理できます。

REVOKE 文は、GRANT 文とはまったく逆の処理を行います。M_Haneef が my_procedure を実行するのを禁止するには、次のコマンドを使用します。

```
REVOKE EXECUTE ON my_procedure FROM M_Haneef
```

このコマンドは、DBA 権限または PERMS ADMIN 権限によって発行するか、プロシージャの所有者が発行する必要があります。

sample_table からローを削除するパーミッションを取り消すには、次のコマンドを発行します。

```
REVOKE DELETE ON sample_table FROM M_Haneef
```

警告！ 権限の取り消しまたはユーザの削除を行う前に、次の制限に注意してください。

- REVOKE CONNECT または sp_dropuser を発行する前に、そのユーザが所有するオブジェクト (テーブルなど) をすべて削除してください。ユーザが所有するデータベース・オブジェクトが残っている状態で、そのユーザの接続権限を取り消したり、ストアド・プロシージャ sp_dropuser を使用したりすると、エラーが通知されます。
- sp_dropuser などのプロシージャには、Adaptive Server Enterprise のストアド・プロシージャとの互換性はほとんどありません。Adaptive Server Enterprise (または Sybase IQ 11.x) のストアド・プロシージャを使い慣れている場合は、それらのテキストと Sybase IQ 15.2 のプロシージャとを比較してから、dbisql でプロシージャを使用してください。比較には、次のコマンドを使用します。

```
sp_helptext 'owner.procedure_name'
```

Sybase が提供するすべてのシステム・ストアド・プロシージャでは、所有者は dbo です。別のユーザが所有する同じ名前前のストアド・プロシージャのテキストを参照するには、そのユーザを指定する必要があります。次に示すのはその例です。

```
sp_helptext 'myname.myprocedure'
```

グループの管理

DBA、RESOURCE、および GROUP パーミッション

前述の項で説明した個別ユーザのパーミッション管理を理解すれば、グループの管理は簡単です。グループも個別ユーザと同様にユーザ ID によって識別されますが、このユーザ ID には「メンバ」を持つというパーミッションが与えられています。

テーブル、ビュー、プロシージャに対するパーミッションがグループに対して付与されるか取り消されると、そのグループのメンバ全員が変更を継承します。しかし、DBA、RESOURCE、GROUP パーミッションは継承されません。これらのパーミッションは、ユーザ ID ごとに個別に割り当てる必要があります。

グループは、単に特別なパーミッションを持つユーザ ID にすぎません。グループに対するパーミッションの付与と取り消しは、「[個別のユーザ ID とパーミッションの管理](#)」(347 ページ) で説明したコマンドを使って、通常のユーザとまったく同じ方法で行います。

グループは、他のグループのメンバになることもできます。グループの階層が構築され、各グループはその親グループのパーミッションを継承します。

ユーザ ID には複数のグループのメンバシップを付与できます。したがって、ユーザとグループの関係は多対多になります。

パスワードなしのグループも作成できます。これは、部外者がグループ・ユーザ ID を使ってデータベースに侵入するのを防ぐ役割を持ちます。このセキュリティ機能については、「[パスワードのないグループ](#)」(359 ページ) を参照してください。

グループの作成

❖ 名前とパスワードのあるグループを作成するには

- 1 DBA 権限を持つユーザ、または USER ADMIN と PERMS ADMIN の両方の権限を持つユーザとしてデータベースに接続します。CREATE USER には USER ADMIN、GRANT GROUP には PERMS ADMIN がそれぞれ必要です。

- 2 次に示す SQL 文を使って、グループのユーザ ID を作成します。

```
CREATE USER personnel
IDENTIFIED BY group_password
```

- 3 personnel ユーザ ID に次の SQL 文を使って、メンバを持つパーミッションを付与します。

```
GRANT GROUP TO personnel
```

ユーザ ID にメンバを持てるようにする GROUP パーミッションは、グループのメンバには受け継がれません。そうしないと、ユーザ ID のそれぞれが、特殊な PUBLIC グループのメンバシップの結果として、自動的にグループになってしまいます。

Sybase Central でグループを作成する

❖ Sybase Central でグループを作成するには

- 1 データベースに接続します。
- 2 [ユーザとグループ]フォルダを右クリックします。
- 3 [新規] - [グループ]を選択します。ウィザードに従って処理を行います。詳細については、Sybase Central のオンライン・ヘルプを参照してください。

ユーザにグループのメンバシップを付与する

ユーザをグループのメンバにするには GRANT 文を使います。グループのメンバシップは、DBA 権限または PERMS ADMIN 権限によって付与できます。

たとえば、次にユーザ M_Haneef にグループ personnel のメンバシップを付与する手順を示します。

- 1 DBA 権限または PERMS ADMIN 権限を持つユーザとしてデータベースに接続します。
- 2 次の SQL 文を使用して、M_Haneef にグループのメンバシップを付与します。

```
GRANT MEMBERSHIP
IN GROUP personnel
TO M_Haneef
```

グループのメンバシップを割り当てられたユーザは、そのグループのテーブル、ビュー、プロシージャに関するパーミッションをすべて継承します。特定のユーザが特定のテーブル、ビュー、またはプロシージャにアクセスできないようにするには、そのオブジェクトのパーミッションを持つグループのメンバからそのユーザを除外します。

グループ内の特定のユーザに対してパーミッションを取り消すことはできません。

Sybase Central でユーザをグループに追加する

❖ Sybase Central でユーザをグループに追加するには

- 1 データベースに接続します。
- 2 そのデータベースの [ユーザ & グループ] フォルダをダブルクリックして開きます。グループは左パネルに表示され、ユーザとグループの両方が右パネルに表示されます。
- 3 右パネルでグループに追加するユーザを選択し、目的のグループまでドラッグします。

詳細については、Sybase Central のオンライン・ヘルプを参照してください。

グループのパーミッション

グループへのパーミッションは、通常のユーザ ID とまったく同じように付与できます。テーブル、ビュー、プロシージャに関するパーミッションは、他のグループとそのメンバも含めて、そのグループのメンバに継承されます。グループのパーミッションについては、データベース管理者が注意する点はいくつかあります。

注意

DBA、RESOURCE、GROUP パーミッションは、グループのメンバには継承されません。つまり、**personnel** ユーザ ID に RESOURCE パーミッションが付与されていても、**personnel** のメンバは RESOURCE パーミッションを持ちません。

データベース・オブジェクトの所有権は単一のユーザ ID に属し、グループ・メンバには継承されません。ユーザ ID **personnel** がテーブルを作成した場合、ユーザ ID **personnel** がそのテーブルの所有者であり、テーブルに変更を加える権限だけでなく、テーブルに関する権限を他のユーザに付与する権限も持ちます。**personnel** のメンバである他のユーザ ID は、このテーブルの所有者ではなく、これらの権利を持っていません。ただし、DBA またはユーザ ID **personnel** 自身が SELECT 権限をユーザ ID **personnel** に明示的に付与した場合は、すべてのグループ・メンバがそのテーブルに対する SELECT アクセス権を持つこととなります。言い換えれば、付与されたパーミッションだけが継承されます。

グループが所有するテーブルの参照

グループを使用して、データベース内のテーブルとプロシージャを検索できます。次のクエリを例にとります。

```
SELECT * FROM SYSGROUPS
```

このクエリは SYSGROUPS テーブルを常に検索します。これは、すべてのユーザが PUBLIC グループに属し、PUBLIC は SYSGROUPS テーブルの所有者である SYS グループに属するからです (SYSGROUPS テーブルには、データベース内のグループ・メンバシップを示す *group_name* と *member_name* の組み合わせのリストが入っています)。

テーブル **Employees** の所有者が **personnel** ユーザ ID で、**M_Haneef** が **Personnel** グループのメンバであれば、**M_Haneef** はテーブル **Employees** を SQL 文で単に **Employees** として参照できます。**Personnel** グループのメンバではないユーザは、修飾名 **Personnel.Employees** を使用する必要があります。

テーブルを所有するグループの作成

テーブルを所有することだけが目的のグループの作成をおすすめします。このグループには何のパーミッションも与えませんが、すべてのユーザをこのグループのメンバにします。こうすれば、すべてのユーザがテーブルを短い名前でも参照できます。次にパーミッション・グループを作成し、ユーザを適宜それらのパーミッション・グループのメンバにします。例については、「[データベース・オブジェクトの名前とプレフィクス](#)」(360 ページ)を参照してください。

パスワードのないグループ

グループのユーザ ID に属するユーザには、何らかのパーミッションがあります。またこのユーザは、グループのユーザ ID の名のもとに作成されたデータベース中のテーブルに対して、所有者のパーミッションを持ちます。

他のユーザ ID にグループ・メンバシップの変更を許可するのではなく、DBA、または PERMS ADMIN 権限を持つ指定のユーザだけがグループとそのデータベース・オブジェクトのすべての処理を実行するようにデータベースを設定できます。

これを行うには、グループを作成するときに、そのグループのユーザ ID で接続できないように指定します。そのためには、GRANT CONNECT 文をパスワードなしで入力します。例を示します。

```
GRANT CONNECT
TO personnel
```

これで、ユーザ ID `personnel` が作成されます。このユーザ ID にはグループ・パーミッションが付与できます。他のユーザ ID にもグループのメンバシップを付与でき、`personnel` に付与されたパーミッションを継承できます。しかし、`personnel` というユーザ ID を使用してデータベースに接続できるユーザはいません。これは、このユーザ ID には有効なパスワードがないためです。

ユーザ ID `personnel` を使用してデータベースに接続できるユーザがない場合でも、このユーザ ID はデータベース・オブジェクトの所有者になることができます。CREATE TABLE 文、CREATE PROCEDURE 文、CREATE VIEW 文のすべてで、文の実行者以外のユーザをオブジェクトの所有者として指定できます。このように所有権の割り当てを実行できるのは、DBA だけです。

特殊なグループ

データベースを作成するとき、2つのグループが自動的に作成されます。SYS と PUBLIC です。どちらのグループもパスワードを持たないため、これらの名前でデータベースに接続することはできません。この2つのグループはデータベース内で重要な働きをします。

SYS グループ

SYS グループはデータベース・オブジェクトとユーザ ID のすべてを含む、データベース構造を記述するシステム・テーブルとビューの所有者です。

システム・テーブルとビューの詳細、およびテーブルへのアクセスの詳細については、『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「第 8 章 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。

PUBLIC グループ

PUBLIC グループはデータベースの作成時に自動的に作成され、データベースに対する CONNECT パーミッションとシステム・テーブルに対する SELECT パーミッションが付与されます。

PUBLIC グループは SYS グループのメンバで、いくつかのシステム・テーブルとビューへのアクセスを継承しています。したがって、データベースのユーザは誰でもデータベース・スキーマについての情報を取得できます。このアクセスを制限するには、PUBLIC の SYS グループにおけるメンバシップを取り消します。

新しいユーザ ID は自動的に PUBLIC グループのメンバとなり、DBA により明確にそのグループに付与されたパーミッションをすべて継承します。必要に応じて、ユーザごとに PUBLIC グループのメンバシップを取り消すこともできます。

データベース・オブジェクトの名前とプレフィクス

データベース・オブジェクトの名前は識別子である必要があります。有効な識別子の規則については、『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「第 2 章 SQL 言語の要素」の「識別子」を参照してください。

このマニュアルのクエリとサンプル SQL 文では、デモ・データベースのデータベース・オブジェクトを簡略名を使用して参照します。例を示します。

```
SELECT *  
FROM Employees
```

テーブル、プロシージャ、ビューにはすべて所有者がいます。デモ・データベースのテーブルの所有者はユーザ ID DBA です。場合によっては、プレフィクスとして所有者のユーザ ID をオブジェクト名に付ける必要があります。次に例を示します。

```
SELECT *  
FROM "DBA".Employees
```

この `Employees` テーブルは「修飾された」といいます (この場合、所有者名 DBA は SQL キーワードなので二重引用符で囲みます)。単にオブジェクト名だけを示すだけでよい場合もあります。この項では、どのような場合にテーブル、ビュー、プロシージャに所有者名をプレフィクスとして付ける必要があるかを説明します。

データベース・オブジェクトを参照するとき、プレフィクスが必要ないのは次の場合です。

- 自分がデータベース・オブジェクトの所有者である場合
- 自分がデータベース・オブジェクトを所有するグループのメンバである場合

例

企業のデータベースを例に説明します。テーブルはすべてユーザ ID `company` によって作成されます。このユーザ ID はデータベース管理者が使うので、DBA 権限が付与されています。

```
GRANT CONNECT TO company
IDENTIFIED BY secret;
GRANT DBA TO company;
```

データベースのテーブルはユーザ ID `company` によって作成されます。

```
CONNECT USER company IDENTIFIED BY secret;
CREATE TABLE company.Customers ( ... );
CREATE TABLE company.Products ( ... );
CREATE TABLE company.Orders ( ... );
CREATE TABLE company.Invoices ( ... );
CREATE TABLE company.Employees ( ... );
CREATE TABLE company.Salaries ( ... );
```

会社の従業員全員がすべての情報にアクセスできるようにはしません。営業部の 2 人のユーザ ID、`Joe` と `Sally` にテーブル `Customers`、`Products`、`Orders` に対するアクセス権を持たせるために、`Sales` グループを作成します。

```
GRANT CONNECT TO Sally IDENTIFIED BY xxxxxx;
GRANT CONNECT TO Joe IDENTIFIED BY xxxxxx;
GRANT CONNECT TO Sales IDENTIFIED BY xxxxxx;
GRANT GROUP TO Sales;
GRANT ALL ON Customers TO Sales;
GRANT ALL ON Orders TO Sales;
GRANT SELECT ON Products TO Sales;
GRANT MEMBERSHIP IN GROUP Sales TO Sally;
GRANT MEMBERSHIP IN GROUP Sales TO Joe;
```

これで、`Joe` と `Sally` にこれらのテーブルを使用するパーミッションが付与されました。しかし、テーブルの所有者は `company` であり、`Sally` と `Joe` は `company` グループのメンバではないため、彼らがテーブルを参照するときには修飾子を付ける必要があります。

```
SELECT *
FROM company.customers
```

この状況を修正するために、`Sales` グループを `company` グループのメンバにします。

```
GRANT GROUP TO company;
GRANT MEMBERSHIP IN GROUP company TO Sales;
```

これで、`Sales` グループのメンバである `Joe` と `Sally` は、間接的に `company` グループのメンバになるため、修飾子なしでテーブルを参照できます。したがって、次のコマンドを使えるようになります。

```
SELECT *
FROM Customers
```

注意

`company` グループのメンバシップにより、Joe と Sally には上記以外のパーミッションはありません。`company` グループに対して他のテーブル・パーミッションを明示的に付与していないためです (`company` ユーザ ID は `Salaries` などのテーブルを作成して DBA 権限を持っているので、それらのテーブルを参照するパーミッションを暗黙的に持っています)。したがって、Joe と Sally は次のコマンドを実行するとエラーになります。

```
SELECT *
FROM Salaries;
SELECT *
FROM company.Salaries
```

どちらの場合も、Joe と Sally は `Salaries` テーブルを参照するパーミッションを持っていません。

高度なセキュリティを実現するためのビューとプロシージャの使い方

上位レベルのセキュリティが必要なデータベースでは、テーブルに対して直接パーミッションを定義するには限界があります。ユーザに付与したテーブルに関するパーミッションは、テーブル全体に対して適用されます。テーブルごとでなく、ユーザのパーミッションをより厳密に定義する必要がある場合は数多くあります。例を示します。

- `employee` テーブルにアクセスする必要があるユーザに対して、テーブルにある個人的な情報にまでアクセスを許可することは望ましくない。
- 営業担当者に訪問販売の詳細を含むテーブルの更新を許可したいが、そのパーミッションを担当者自身の部分だけに制限したい。

これらのケースでは、会社のニーズに応じてパーミッションを調整するためにビューとストアド・プロシージャを使うことができます。この項では、パーミッション管理でのビューとストアド・プロシージャの使い方について説明します。

ビューの作成方法については、『システム管理ガイド第1巻』の「[第5章 データベース・オブジェクトの使用](#)」の「[ビューの編集](#)」を参照してください。

ビューを使用したセキュリティの調整

ビューはベース・テーブルから選択したローとカラムを含む、計算されたテーブルです。ビューはユーザにテーブルの一部だけへのアクセスを与える場合に便利です。その部分として、ローまたはカラムを定義できます。たとえば、ユーザのグループが `Employees` テーブルの `Salary` カラムを参照できないようにしたり、ユーザ自身が作成したローしか参照できないようにしたりできます。

例 1

販売管理者は、自分の部署の従業員に関するデータベースの情報にアクセスする必要があります。しかし、この管理者が他部署の従業員のデータにアクセスする理由はありません。

次に、販売管理者のユーザ ID を作成し、管理者が必要な情報を得るためのビューを作成して、販売管理者のユーザ ID に適切なパーミッションを付与する手順を示します。

- 1 DBA 権限を持つユーザ ID から GRANT 文を使って新しいユーザ ID を作成します。次のように入力します。

```
CONNECT "DBA"  
IDENTIFIED by sql;  
GRANT CONNECT  
TO SalesManager  
IDENTIFIED BY sales
```

SELECT や FROM の場合と同様に、DBA は SQL キーワードなので引用符で囲む必要があります。

- 2 営業部の従業員だけを参照するビューを次のように定義します。

```
CREATE VIEW emp_sales AS  
SELECT EmployeeID, GivenName, Surname  
FROM "DBA".Employees  
WHERE DepartmentID = 200
```

ユーザ ID SalesManager がビューを使用できるように、テーブルの所有者を明示的に指定して、テーブルを "DBA".Employees と指定します。このようにしないと、SalesManager がビューを使用するとき、SELECT 文はこのユーザ ID が認識しないテーブルを参照することになります。

- 3 次のように、SalesManager にビューを参照するパーミッションを付与します。

```
GRANT SELECT  
ON emp_sales  
TO SalesManager
```

テーブルに対するパーミッションを付与するときと同じコマンドを使用して、ビューに対するパーミッションを付与します。

例 2

次の例は、販売管理者が注文のまとめを参照できるビューを作成します。このビューは、複数のテーブルからの情報を必要とします。

- 1 ビューを作成します。

```
CREATE VIEW order_summary AS  
SELECT OrderDate, Region, SalesRepresentative  
FROM "GROUPO".SalesOrders  
KEY JOIN "GROUPO".Customers
```

- 2 SalesManager にこのビューを参照するパーミッションを付与します。

```
GRANT SELECT
ON order_summary
TO SalesManager
```

- 3 成功したかどうかを確認するために、SalesManager ユーザ ID に接続して作成したビューを参照します。

```
CONNECT SalesManager IDENTIFIED BY sales ;
SELECT * FROM "GROUPO".emp_sales ;
SELECT * FROM "GROUPO".order_summary ;
```

販売管理者には基本となるテーブルを参照するパーミッションは付与されていません。したがって、次のコマンドはパーミッション・エラーを起こします。

```
SELECT * FROM "DBA".Employees ;
SELECT * FROM "DBA".SalesOrders;
```

ビューに対するその他の
パーミッション

前述の例では、ビューを使用して SELECT パーミッションを調整する方法について説明しました。同じ方法で、ビューに関する INSERT、DELETE、UPDATE パーミッションを付与できます。

ビューでのデータの修正を許可する方法については、『システム管理ガイド第1巻』の「第5章 データベース・オブジェクトの使用」の「ビューの使用」を参照してください。

プロシージャを使用したセキュリティの調整

ビューはデータへのアクセスを制限しますが、プロシージャはユーザの行動を制限します。「[プロシージャに対するパーミッションの付与](#)」(354 ページ)で説明したように、ユーザは、プロシージャの対象となるテーブルに対するパーミッションがなくても、プロシージャの EXECUTE パーミッションを持つことができます。

デフォルトでは、プロシージャはそのプロシージャの所有者のパーミッションで実行されます。テーブルを更新するプロシージャの場合、プロシージャの所有者がそのテーブルの UPDATE パーミッションを持っていれば、ユーザはプロシージャを実行できます。プロシージャの所有者は、CREATE/ALTER PROCEDURE 文に対して SQL SECURITY INVOKER を指定することによって、プロシージャを実行するユーザのパーミッションを使用してプロシージャを実行するよう制限できます。

厳密なセキュリティ

セキュリティを完全にするには、基本となるテーブルへのアクセスをすべて禁止し、ユーザまたはユーザのグループには、特定のストアド・プロシージャを実行するパーミッションだけを付与します。この方法であれば、データベースのデータを変更する方法を厳密に定義できます。

特定の権限を持つユーザが、IQ システム・プロシージャを使用して特定のタスクを管理できるようにするには、次の手順に従います。

- 対象の各権限用にグループを作成します。
- 指定したグループに権限を付与します。
- グループに、その権限のタスクを実行する IQ プロシージャの EXECUTE パーミッションを付与します。

権限を付与する新しいユーザを作成したら、その権限用に作成したグループにユーザのメンバシップを付与します。ほとんどの権限はグループ・メンバシップを通じて継承されるため、ユーザは権限と IQ プロシージャの実行パーミッションをグループから継承します。

❖ 関連ストアド・プロシージャを実行するためのパーミッションをユーザに付与する

次の手順には、DBA 権限、または PERMS ADMIN 権限と USER ADMIN 権限が必要です。

USER ADMIN 権限と、ユーザ管理に関連するプロシージャの実行パーミッションをユーザ user1 に付与するには、次の手順に従います。CREATE USER USERADMIN_GRP 文には USER ADMIN 権限が必要です。

- 1 グループ USER ADMIN_GRP を作成します。

```
CREATE USER USERADMIN_GRP
GRANT GROUP TO USERADMIN_GRP
```

- 2 USER ADMIN 権限を USERADMIN_GRP に付与します。

```
GRANT USER ADMIN TO USERADMIN_GRP
```

- 3 ユーザ管理用の Sybase IQ ストアド・プロシージャの EXECUTE パーミッションを USERADMIN_GRP に付与します。

```
GRANT EXECUTE on sp_iqaddlogin
to USERADMIN_GRP
GRANT EXECUTE on sp_iqcopyloginpolicy
to USERADMIN_GRP
GRANT EXECUTE on sp_iqdroplogin
to USERADMIN_GRP
GRANT EXECUTE on sp_iqmodifyadmin
to USERADMIN_GRP
GRANT EXECUTE on sp_iqmodifylogin
to USERADMIN_GRP
```

- 4 USERADMIN_GRP グループのメンバシップを user1 に付与します。ユーザ user1 は、USER ADMIN 権限と、割り当てられている IQ プロシージャの実行パーミッションを USERADMIN_GRP グループのメンバシップを通じて継承します。

```
GRANT MEMBERSHIP IN GROUP USERADMIN_GRP TO user1
```

同じ手順を使用して、他の権限用の同様のグループを作成できます。次に例を示します。

グループ名	権限の付与	ストアド・プロシージャの EXECUTE パーミッションの付与
OPERATOR_GRP	OPERATOR	sp_iqbackupdetails、sp_iqbackupsummary、sp_iqconnection、sp_iqsysmon
SPACEADMIN_GRP	SPACE ADMIN	sp_iqdbspace、sp_iqdbspaceinfo、sp_iqdbspaceobjectinfo、sp_iqemptyfile、sp_iqestdbspaces、sp_iqfile、sp_iqobjectinfo、sp_iqspaceused

マルチプレックスの例については、『Sybase IQ マルチプレックスの使用』を参照してください。

プロシージャを使用して接続を無効にする

ストアド・プロシージャ `sp_iqmodifylogin` を使用して、接続を無効にできません。例については、『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」の「[sp_iqmodifylogin プロシージャ](#)」を参照してください。

ユーザ・パーミッションの評価方法

グループのためにユーザのパーミッションは複雑になります。ユーザ `M_Haneef` はあるテーブルに対する `SELECT` パーミッションと `UPDATE` パーミッションを個人的に付与されていますが、このテーブルへのアクセス権がまったくないグループと、`SELECT` アクセス権だけを持つグループのメンバーでもあるとします。この場合、このユーザのパーミッションは一体どうなるのでしょうか。

Sybase IQ は、ユーザ ID が特定のアクションを実行するパーミッションを持っているかどうかを、次のように判断します。

- 1 ユーザ ID が DBA パーミッションを持っている場合、そのユーザはデータベース内ですべての作業を行うことができます。ユーザが「[権限](#) (334 ページ)」で説明した権限を持っている場合、ユーザはその権限に関連付けられたパーミッションを持っています。
- 2 DBA パーミッションがない場合、パーミッションは、個別のユーザに割り当てられたパーミッションによって異なります。そのアクションを実行するパーミッションが付与されていれば、アクションを処理できます。
- 3 そのユーザに個別の設定が行われていない場合は、そのユーザがメンバーであるグループごとのパーミッションを見ます。いずれかのグループがそのアクションを実行するパーミッションを持っていれば、そのユーザ ID もメンバーとしてパーミッションを持っており、アクションを処理できます。

特定のユーザが特定のテーブル、ビュー、またはプロシージャにアクセスできないようにするには、そのオブジェクトのパーミッションを持つグループのメンバからそのユーザを除外します。

このようにして、パーミッションが設定されている順序に関する問題を最小限に抑えます。

リソース接続使用の管理

ユーザとグループのセットを構築すると、データベースのパーミッションを管理できます。またデータベースのセキュリティと管理によって、個々のユーザが使用できるリソースを制限することもできます。

たとえば、データベースの他のユーザの接続速度を遅らせないために、単一の接続がメモリや CPU を大量に使用するのを防止できます。

ユーザ・リソースを制御するデータベース・オプション

Sybase IQ には、DBA がリソースの制御に使用できるデータベース・オプションのセットが用意されています。これらのオプションは「リソース・ガバナー」と呼ばれます。

オプションの設定

SET OPTION 文を使用してデータベース・オプションを設定できます。構文については、『リファレンス：文とオプション』の「[SET OPTION 文](#)」を参照してください。

オプションに関するリファレンス情報については、『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」を参照してください。

管理できるリソース

リソースを管理するには次のデータベース・オプションを使用できます。これらのオプションの詳細については、『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」、または『リファレンス：文とオプション』を参照してください。

- **CURSOR_WINDOW_ROWS** バッファへのカーソル・ローの数を定義します。
- **MAX_CARTESIAN_RESULT** 直積ジョインを含むクエリの結果ローの数を制限します。
- **MAX_IQ_THREADS_PER_CONNECTION** IQ の操作で使用する接続に対し有効な処理スレッドの数を設定します。
- **TEMP_CACHE_MEMORY_MB** IQ テンポラリ・ストアのキャッシュ・サイズを設定します (テンポラリ・キャッシュ・サイズの設定には、サーバ・オプション `-iqtc` の使用をおすすめします)。

- **QUERY_TEMP_SPACE_LIMIT** 1つのクエリが使用できるテンポラリ dbspace の量を制限します。
- **QUERY_ROWS_RETURNED_LIMIT** 1クエリ・オプティマイザに対し、リソース消費が多すぎるクエリを拒否するよう指定します。オプティマイザの見積もりで、クエリの結果セットがこのオプションの値を超えると判断された場合、オプティマイザはクエリを拒否し、エラー・メッセージを返します。

次のデータベース・オプションはエンジンに影響を与えますが、Sybase IQ にはほとんど影響しません。

- **JAVA_HEAP_SIZE** 接続ベースで Java アプリケーションに割り付けるメモリの一部の最大サイズ (バイト単位) を設定します。
- **MAX_CURSOR_COUNT** 1 接続に対するカーソル数を制限します。
- **MAX_STATEMENT_COUNT** 1 接続に対する準備文の数を制限します。

データベース・オプション設定は、グループ構造内で継承されません。

ユーザ・リソースに影響するその他の設定

Sybase IQ ログイン管理で、ユーザが利用できるリソースを制御できます。これにより、全ユーザまたは個々のユーザのデータベース接続数を制限したり、パスワードの有効期間を設定したり、個々のユーザをロックアウトしてデータベースに接続できないようにすることができます。詳細については、「[ユーザ・アカウントと接続の管理](#)」(342 ページ) を参照してください。

メモリに影響する設定と、ユーザ接続で利用できる他のリソースに影響する設定の詳細については、『パフォーマンス&チューニング・ガイド』の「[第4章 システム・リソースの管理](#)」を参照してください。

システム・テーブルのユーザとパーミッション

現在のデータベースのユーザとそのパーミッションに関する情報は、システム・テーブルとシステム・ビューに入っています。

これらの各テーブルの詳細については、『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「[第8章 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。

ほとんどのシステム・テーブルは、特殊なユーザ ID である SYS が所有しています。SYS ユーザ ID には接続できません。

DBA は他のテーブルと同様にすべてのシステム・テーブルに対して SELECT アクセス権を持ちます。他のユーザのシステム・テーブルへのアクセスは制限されています。たとえば、データベース・ユーザのパーミッションに関するすべての情報と、各ユーザ ID のパスワードに関する情報が格納されている SYS.SYSUSERPERM テーブルには、DBA だけがアクセスできます。しかし、パスワード以外の SYS.SYSUSERPERM のすべての情報が表示される SYS.SYSUSERPERMS ビューには、デフォルトですべてのユーザが SELECT アクセス権を持っています。新しいデータベース中で SYS、PUBLIC、DBA に対して設定されたすべてのパーミッションとグループ・メンバシップは自由に変更できます。

次に示すテーブルは、ユーザ ID、グループ、パーミッションに関する情報を含むシステム・テーブルとビューの要約です。すべてのテーブルとビューはユーザ ID SYS によって所有されており、修飾名は SYS.SYSUSERPERM などになります。

これらのテーブルに対して適切な SELECT クエリを使えば、すべてのユーザ ID とパーミッションの情報を得られます。

テーブル	デフォルト	内容
SYSUSERPERM	DBA のみ	データベース・レベルのパーミッションと各ユーザ ID のパスワード
SYSGROUP	PUBLIC	各グループのメンバごとに 1 ロール
SYSTABLEPERM	PUBLIC	GRANT コマンドで与えられるテーブルに関するすべてのパーミッション
SYSCOLPERM	PUBLIC	GRANT コマンドで与えられる UPDATE パーミッションを持つすべてのカラム
SYSROCPERMM	PUBLIC	各ロールには、1 つのプロシージャを使うパーミッションを付与された 1 人のユーザが含まれる

次に、ユーザ ID、グループ、パーミッションに関する情報を含むシステム・ビューの要約を示します。

ビュー	デフォルト	内容
SYSUSERAUTH	DBA のみ	ユーザ ID 以外のすべての SYSUSERPERM の情報
SYSUSERPERMS	PUBLIC	パスワード以外のすべての SYSUSERPERM の情報
SYSUSERLIST	PUBLIC	パスワード以外のすべての SYSUSERAUTH の情報
SYSGROUPS	PUBLIC	SYSGROUP の情報を読みやすくしたもの

ビュー	デフォルト	内容
SYSTABAUTH	PUBLIC	SYSTABLEPERM の情報を読みやすくしたもの
SYSCOLAUTH	PUBLIC	SYSCOLPERM の情報を読みやすくしたもの
SYSPROCAUTH	PUBLIC	SYSPROCPERM の情報を読みやすくしたもの

これらのテーブルとビュー以外に、データベースの各オブジェクトの情報を含むテーブルとビューがあります。

トランスポート・レイヤ・セキュリティ

トランスポート・レイヤ・セキュリティ (TLS: Transport-Layer Security) を使用して、クライアントと IQ サーバ間や IQ クライアントとデータベース・サーバ間の通信を保護できます。

Sybase IQ データベースのデータベース・ファイル暗号化は、SQL Anywhere データベースのものと同様です (Sybase Product Manuals Web サイトの『SQL Anywhere サーバ データベース管理』を参照)。また、Sybase IQ では、カラムを暗号化することもできます (『Sybase IQ による高度なセキュリティ』を参照)。

FIPS 暗号化、Kerberos 認証、およびカラムの暗号化のサポートは、別途ライセンスが必要な Sybase IQ Advanced Security オプションに含まれています。このオプションについては、『Sybase IQ による高度なセキュリティ』を参照してください。

IPv6 のサポート

Sybase IQ ではインターネット・プロトコル・バージョン 6 (IPv6) をサポートしています。これには、インターネット上でパケットをルーティングするためのアドレス情報と制御情報が含まれています。IPv6 では 2^{128} 個のユニークな IP アドレスをサポートします。この数は、先行の IPv4 がサポートしていたアドレス数に比べ大幅に増加しています。Sybase IQ では、クライアントまたはサーバで IP アドレスを指定できる IPv4 および IPv6 のアドレスをサポートします。

ODBC クラスでは、リモート・データ・アクセスでの IPv6 アドレスの使用をサポートします。JDBC クラスでは、リモート・データ・アクセスでの IPv6 アドレスの使用をサポートしていません。

IPv6 サポートの詳細については、『SQL Anywhere サーバ データベース管理』の「TCP/IP プロトコルの使用」を参照してください。

この章について

この章では、データベースのデータの有効性と信頼性を確保するための機能について説明します。これらの機能には、テーブルおよびカラムに対する制約や、適切なデータ型の選択が含まれます。

この章では、SQL 文の CREATE TABLE 文と ALTER TABLE 文について説明します。これらの基本的な使い方については、「第 5 章 データベース・オブジェクトの使用」を参照してください。

内容

トピック名	ページ
データ整合性の概要	371
カラムのデフォルトの使用	375
テーブル制約とカラム制約の使用	383
エンティティ整合性と参照整合性の宣言	387
システム・テーブルの整合性ルール	396

データ整合性の概要

データに整合性があるとは、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。

Sybase IQ では、データベースへのデータ入力を詳細に制御するためのストアド・プロシージャと JDBC を使用できます。プロシージャについては、『システム管理ガイド第 2 巻』を参照してください。

JDBC の詳細については、Sybooks Online Help の SQL Anywhere Studio のマニュアルにある『SQL Anywhere サーバ プログラミング』の「第 5 章 JDBC プログラミング」を参照してください。
(<http://infocenter.sybase.com/help/index.jsp>)

データが無効になるとき

適切な検査が行われない場合にデータベースのデータが無効になる例をいくつか次に示します。どの場合についても、この章で説明する機能を使用して、データが無効にならないようにできます。

- customers テーブルに存在しない customer_id に対応する注文を orders テーブルに入力した場合
- 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
- dept_id が 200 の新しい部署が設置され、これを会社のデータベースの department テーブルに追加する必要があるが、この情報を 2 人で別々に入力した場合

データの有効性の確保

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ための検査と、データが従うべきルールが必要です。こうしたルールは、ビジネス・ルールと呼ばれることがよくあります。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。

制約は組み込みが必要

データベースそのものに組み込まれた制約のほうが、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からです。

データベースに一度制約を組み込めば、データ・ソースにかかわらず、それ以降のデータベースに対するすべての動作に対して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、アプリケーションが変更されるたびに無効になります。場合によっては、制約を複数のアプリケーションに組み込むか、1 つのアプリケーション内で複数の箇所に組み込む必要があります。

IQ のデータは、通常、少数のユーザによって入力され、他のデータベースから直接ロードされることもしばしばです。このため、使用する抽出処理、変換処理、ロード処理によっては、IQ データベースの方が、OLTP データベースに比べて、データが無効になるようなエラーが発生する頻度は少なくなることがあります。

Sybase IQ によって自動的に適用されるかどうかにかかわらず、適用する制約を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。

制約による IQ の最適化の支援

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。この最適化には、制約は適用されません。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んでください。

次に、最適化をいくつか示します。この最適化は、データベースに組みこむ制約とその他の機能に依存します。

- ジョイン・インデックスは、異なるカラムからデータをジョインするクエリを最適化します。多くの場合、1つのジョイン・インデックスのジョイン関係は、ジョインされるテーブルに対して指定した外部キー制約に依存します。
- FOREIGN KEY、PRIMARY KEY、UNIQUE の各カラム制約と IQ UNIQUE パラメータを使用すると、ロードとクエリのパフォーマンスを向上できます。

制約が最適化に与える影響の詳細については、「[テーブルの作成](#)」(197 ページ)を参照してください。ジョイン・インデックスと外部キーの詳細については、「[ジョイン・インデックスの使用](#)」(245 ページ)を参照してください。

ロード中に制約を検査

Sybase IQ は、特定の制約が守られているかどうかを、ロード・オペレーション中に検査します。

- Sybase IQ 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します。
- 外部キー - プライマリ・キー関係に依存するジョイン・インデックスがある場合、Sybase IQ は、ジョイン・インデックスの同期時に、基本となるテーブル内のデータがジョイン対象のカラム間で想定される 1 対多の関係を保っているかを確認します。

データベース内容の変更

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 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

データ整合性の確保を支援するために、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できます。

制約

個々のカラムまたはテーブルごとに、データに対して何種類かの制約を使用できます。次に例を示します。

- NOT NULL 制約は、カラムに null 値が入力されるのを防ぎます。Sybase IQ がこの制約を適用します。

エンティティおよび参照整合性

- カラムに CHECK 条件を付けて、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、入力範囲を指定して、salary カラムに入力させることができます。
- 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設定できます。

ここにあげた制約とその他のテーブルおよびカラムの制約については、「[テーブル制約とカラム制約の使用](#)」(383 ページ) を参照してください。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。

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(「foreign」テーブル)の情報をもう一方のテーブル(「referenced」または「primary」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(またはカラムのセット)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(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**参照キー**と呼ばれます。

候補キーには次の制限があります。

- ジョイン・インデックスが存在する場合は、外部キーは候補キーにはなりません。
- 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。

- **エンティティ整合性** プライマリ・キーを追跡します。これで、null ではないことが確かなプライマリ・キーによって、
- **参照整合性** テーブル間の関係を定義する外部キーを追跡します。これで、すべての外部キーの値が、対応するプライマリ・キーの値に一致するか、NULL 値が許可されている場合は NULL になっていることが保証されます。

参照整合性の詳細については、「[エンティティ整合性と参照整合性の宣言](#)」(387 ページ) を参照してください。

整合性制約を実装するための SQL 文

整合性制約を実装するための SQL 文を次に示します。

- **CREATE TABLE 文** この文は、データベースの作成時に整合性制約を実装します。
- **ALTER TABLE 文** この文は、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。

これらの文の構文の詳細については、『リファレンス：文とオプション』の「[第1章 SQL 文](#)」を参照してください。

カラムのデフォルトの使用

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、指定された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。割り当てられる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

Sybase IQ は、カラムについて次のデフォルト値をサポートします。

- 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ャ・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

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、および GLOBAL AUTOINCREMENT という特別値を使用する値
- 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

Sybase IQ は、DEFAULT_TIMESTAMP_INCREMENT データベース・オプションの設定を無視します。

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のデフォルトを作成するか、または ALTER TABLE 文を使用して、カラムのデフォルトを後で追加できます。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 によって、すべてのテーブルのすべてのカラムに関する情報が返されます。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。

例 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)
```

次の文は、sales_order テーブルの id という既存のカラムに条件を追加します。これによって、クライアント・アプリケーションが値を指定しないかぎり、このカラムの値は自動的に増分されます。

```
ALTER TABLE sales_order MODIFY id DEFAULT AUTOINCREMENT
```

次の文は、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
```

他のデフォルト値は同様のやり方で指定します。詳細については、『リファレンス: 文とオプション』の「第 1 章 SQL 文」の「ALTER TABLE 文」、「CREATE TABLE 文」、および「CREATE DOMAIN 文」を参照してください。

カラムのデフォルトの変更と削除

デフォルトを作成するときと同じ形式の ALTER TABLE 文を使用して、カラムのデフォルトを変更または削除できます。次の文は、`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` に変更します。

```
ALTER TABLE sales_order
MODIFY order_date DEFAULT CURRENT DATE
```

カラムのデフォルトを `NULL` に変更すれば、そのデフォルトを削除できます。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値の操作

Sybase IQ は、次の文によってカラムのデフォルト値のロードと挿入をサポートします。

- `INSERT...VALUES`
- `INSERT...SELECT`
- `INSERT...LOCATION`
- `LOAD TABLE`
- `UPDATE`
- `SELECT...FROM...FOR UPDATE`

Sybase IQ は、次の要件に従ってカラムのデフォルト値の定義と挿入を処理します。

- Sybase IQ では、Sybase IQ が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` オペレーションを実行すると、エラーが報告されます。
- 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、Sybase IQ が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。サポートされている暗黙のデータ型変換については、[表 7-4 \(311 ページ\)](#) を参照してください。
- 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、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、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** 型のカラムのデフォルト値がマルチバイト文字の一部を含む文字列の場合、**TRIM_PARTIAL_MBC** データベース・オプションの設定に応じて、Sybase IQ は値を挿入する前にその部分をトリムすることがあります。
- カラムの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、Sybase IQ はエラー・メッセージを生成します。
- **LOAD TABLE** オペレーションの最中にデフォルト値の挿入によって発生するすべての制約違反は、ユーザが指定した **IGNORE CONSTRAINT** および **MESSAGE LOG/ROW LOG** オプションに適用されます。
- Sybase IQ では、**UTC TIMESTAMP** および **CURRENT UTC TIMESTMAP** のカラムのデフォルト値はサポートされていません。このタイプのカラムのデフォルト値を挿入または更新しようとするたびに、エラーが報告されます。
- 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。
- ジョイン・インデックスに関与するテーブルにはカラムのデフォルト値が許可されません。そうしたテーブルにデフォルト値を定義しようとすると、Sybase IQ はエラーを生成します。このルールは、**AUTOINCREMENT** デフォルト値のサポートに類似しています。
- 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れられます。
- 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** オプションを **ON** に設定してください。**DEFAULTS** オプションが **OFF** だと、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必要があ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、データベースと同じ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準拠する必要があります。`LOAD TABLE DEFAULT` オプションでは、ロード・デフォルト値として `AUTOINCREMENT`、`IDENTITY`、`GLOBAL AUTOINCREMENT` をサポートしていません。
- `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。

カラムのデフォルト値の定義と挿入の詳細については、それぞれのデフォルト値の型について説明している後出の各項を参照してください。デフォルトのカラム値の式で使用できる特別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「第2章 SQL 言語の要素」の「特別値」も参照してください。

Sybase Central におけるカラムのデフォルトの編集

Sybase Central で、カラム・プロパティ・シートの [値] タブを使用して、カラムのデフォルトを追加、変更、削除できます。詳細については、「[Sybase Central におけるカラムのデフォルトの編集](#)」(379 ページ) を参照してください。

注意 新しいカラムを作成する場合、[データ型] または [値] を選択してから省略記号をクリックするまで、一部の属性は表示されません。

日付、時刻、タイムスタンプのデフォルト

`DATE`、`TIME`、または `TIMESTAMP` データ型のカラムの場合、デフォルトとして特別値の `CURRENT DATE`、`CURRENT TIME`、`TIMESTAMP`、または `CURRENT TIMESTAMP` を使用できます。カラムのデータ型と互換性のあるデフォルトを選択する必要があります。

`CURRENT DATE` デフォルトの例

`CURRENT DATE` デフォルトは記録に便利です。

- 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、こちらのほうが高精度です。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日のうちに同じ顧客と何度も連絡をとることがあります。`CURRENT TIMESTAMP` 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 では日付と時刻がマイクロ秒単位の精度で記録されるため、データベースの中でイベントのシーケンスが重要な意味を持つときにも **CURRENT TIMESTAMP** が便利です。

TIMESTAMP デフォルト

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

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

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

タイムスタンプと時刻と日付の詳細については、『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「[第 3 章 SQL データ型](#)」を参照してください。

USER デフォルト

カラムに **DEFAULT USER** を割り当てるのは、データベース内にエントリを作成したユーザを識別するための簡単で確実な方法です。たとえば、営業部員が歩合制で働いている場合は、この情報が必須になることもあります。

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。こうしたユーザは、自分の作業に関連のあるテーブルのコピーをポータブル・コンピュータ上に作成し、マルチユーザ・データベースに接続していないときに変更を行い、その後、帰社後にトランザクション・ログをサーバに適用できます。

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** では、**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** および **LOAD** では、この定数は **CURRENT USER** と同じ効果があります。**UPDATE** では、**LAST USER** のデフォルト値を持つカラムが明示的に変更されない場合、現在のユーザの名前に変更されます。

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

IDENTITY または AUTOINCREMENT のデフォルト

IDENTITY/AUTOINCREMENT のデフォルトは、数値データ・フィールドの数値自体が意味を持たない場合に有用です。この機能では、新しい各ローの当該カラムに対して、それまでの最大値よりも1大きい値が代入されます。IDENTITY/AUTOINCREMENT カラムは、購入注文番号の記録や、カスタマ・サービスの通話の識別など、番号の識別が必要な場合に使用できます。

AUTOINCREMENT 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あるのが普通です (『リファレンス：文とオプション』の「第1章 SQL 文」の「CREATE TABLE 文」を参照してください)。たとえば、オートインクリメントのデフォルトは、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。サーバはインデックスまたはキー定義を使用して最大値を検出するからです。

オートインクリメントの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。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「第2章 SQL 言語の要素」を参照してください。

Sybase IQ は、特別値の GLOBAL AUTOINCREMENT をサポートしません。

オートインクリメントと負の数

IDENTITY/AUTOINCREMENT は、正の数に対して使用するためのものです。テーブルの作成時には、IDENTITY/AUTOINCREMENT の初期値は0に設定されます。

オートインクリメントと IDENTITY カラム

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。Sybase IQ は、互換性を維持するために、両方のキーワードをサポートします。

NEWID デフォルト

GUID (Globally Unique Identifiers) と呼ばれることもある UUID (Universally Unique Identifiers) を使用すると、テーブル内のローを一意に識別できます。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、およびプロシージャ』の「第4章 SQL 関数」にある「NEWID 関数 [その他]」か「第3章 SQL データ型」の「バイナリ・データ型」にある「UNIQUEIDENTIFIER データ型」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の場合、NULL デフォルトを指定することは、デフォルトをまったく指定しないことと同じです。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「第 2 章 SQL 言語の要素」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がある場合に有用です。たとえば、2つのオフィスを持つ組織があり、本部が city_1 に、支部が city_2 に置かれている場合、ロケーション・カラムのデフォルト・エントリを city_1 に設定するとデータ入力容易になる可能性があります。

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、定数式を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なる値でない場合は、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() ) )
```

テーブル制約とカラム制約の使用

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・ユーザに支障がある場合があります。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 の実行中、文の処理は行われますが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。

この項では、制約を使用してテーブルに正しいデータを入力したり、Sybase IQ のパフォーマンスが向上するように情報を与えたりする方法について説明します。

カラムまたはテーブルに対する UNIQUE 制約の使用

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 制約を適用すると、Sybase IQ がこの条件を実行します。

UNIQUE は、基本的には PRIMARY KEY 制約と同じです。違うのは、UNIQUE 制約は 1 つのテーブル内で複数指定できる点です。UNIQUE と PRIMARY KEY のいずれでも、カラムは NULL 値を持つことはできません。

例 1 次の例では、`ss_number` というカラムを `employee` テーブルに追加し、テーブル全体でカラムの各値をユニークにします。

```
ALTER TABLE employee
ADD ss_number char(11) UNIQUE
```

例 2 この例では、ユニーク・エントリの作成に 3 つのカラムが必要です。

```
ALTER TABLE product
ADD UNIQUE (name, size, color)
```

カラムに対する IQ UNIQUE 制約の使用

IQ UNIQUE 制約は、カラム内の個別の値の予測数を指定します。IQ UNIQUE 制約は、テーブル内のすべてのカラムに適用できます。この制約を使用して、インデックスのロードを最適化できます。

Sybase Central の IQ プラグインでは、カラム・プロパティ・ページで **IQ UNIQUE** 制約を追加できます。詳細については、オンライン・ヘルプを参照してください。

たとえば、**employee** テーブルの **state** カラムに **IQ UNIQUE(50)** を指定して、可能な値が 50 個 (アメリカの州の数) だけであることを示します。可能な値は、それぞれ何度も使用できます。

MINIMIZE_STORAGE データベース・オプションが ON の場合、新しいカラムすべてに対して **IQ UNIQUE(255)** を指定するのと同じ意味になります。バージョン 12.6 現在、このオプションはデフォルトで OFF になっています。

カラムに対する CHECK 条件の使用

CHECK 条件を使用して、カラム内の値がある一定の基準を満たすように指定できます。

1つのカラム内の値に CHECK 条件を適用して、値が従うべきルールを指定できます。こうしたルールを指定することによって、不正なデータが入力されないようにします。企業内部の規定や手順を反映する場合は、さらに厳密なルールにします。

カラム内の値を一定の範囲に収めるときに、個々のカラム値に対して CHECK 条件を使用すると便利です。次にその例を示します。

例 1

決められた値のいずれかが入力されるように設定できます。たとえば、**city** カラムに事業所の所在地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います。

```
ALTER TABLE office
MODIFY city
CHECK ( city IN ( 'city_1', 'city_2', 'city_3' ) )
```

デフォルトでは、文字列の比較の場合、**CASE RESPECT** オプションを使用して特に指定しないかぎり、大文字と小文字が区別されません。詳細については、『リファレンス：文とオプション』の「**CREATE DATABASE 文**」の CASE 句を参照してください。

例 2

日付や数字が一定の範囲内に収まるように設定できます。たとえば、**employee** テーブルの **start_date** カラム (従業員の入社日) 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。この例で使用した **YYYY/MM/DD** フォーマットには、現在のオプション設定に関係なく、常に認識されるという利点があります。

ユーザ定義データ型の CHECK 条件の定義

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別の CHECK 条件を設定すると、この新しい条件は、先に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す。

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 )
```

変数名の先頭に @ が付いていれば、@col の代わりに使用できます。posint データ型を使用して定義されたカラムは、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。

ユーザ定義の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』の「第3章 SQL データ型」の「ドメイン」を参照してください。

Sybase Central におけるカラム制約の編集

Sybase Central でカラム制約を追加、修正、削除するには、カラム・プロパティ・シートの [制約] タブを使います。

❖ カラムのプロパティ・シートの表示

- 1 データベースに接続します。
- 2 目的のデータベースの [テーブル] フォルダをクリックし、変更するカラムの入ったテーブルをクリックします。
- 3 カラムを選択し、[ファイル] - [プロパティ] をクリックします。

詳細については、Sybase Central のオンライン・ヘルプを参照してください。

テーブルに対する CHECK 条件の使用

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、CHECK 条件を適用できます。通常、このような CHECK 条件を使って、ローで入力または変更された2つの値が互いに正しい関係を持っているかをチェックします。さまざまなカラム CHECK 条件を個別にシステム・テーブルに格納して、それぞれの条件を個別に置き換えたり削除したりできます。この方法は柔軟性が高いので、可能であれば CHECK 条件をカラムごとに設定することをおすすめします。

たとえば、図書館データベースでは、返却日 `date_returned` は貸出日 `date_borrowed` よりも必ず後（または同じ日）になる必要があります。

```
ALTER TABLE loan
ADD CHECK(date_returned >= date_borrowed)
```

CHECK 条件の修正と削除

テーブルで既存の CHECK 条件を変更するには、いくつかの方法があります。

- 前述のように、テーブルまたはカラムに新しい CHECK 条件を追加できます。
- カラムの CHECK 条件を NULL に設定して削除できます。次に示す文では、`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。次に示す文では、`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' ) )
```

カラムに定義されている CHECK 条件に対し、テーブルに定義されている CHECK 条件を修正するには、2通りの方法があります。

- `ALTER TABLE` と `ADD` テーブル制約句を使って、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 条件を作成します。

テーブル上のすべてのカラムの CHECK 条件とユーザ定義データ型から継承した CHECK 条件を含む、テーブル上のすべての CHECK 条件は、ALTER TABLE 文とともに DELETE CHECK 句を使用して削除します。次に例文を示します。

```
ALTER TABLE table_name
DELETE CHECK
```

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、column not found というエラー・メッセージが返されます。

エンティティ整合性と参照整合性の宣言

データベースの関係構造によって、データベース・サーバはデータベース内の情報を識別できます。また、Sybase IQ では、テーブル間のプライマリ・キー-外部キー関係が、その関係に依存しているあらゆるジョイン・インデックス内のすべてのローによって適切に維持されます。

エンティティ整合性の宣言

各テーブルのプライマリ・キーを指定したら、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は、エンティティ整合性を管理するための追加的な操作は必要ありません。

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 句か 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。詳細については、『リファレンス：文とオプション』の「ALTER TABLE 文」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している場合は、CREATE TABLE コマンドや ALTER TABLE コマンドを明示的に入力する必要はありません。指定した情報に基づいて、アプリケーションが文自体を生成してくれます。

プライマリ・キーの作成については、「[プライマリ・キーと外部キーの作成](#) (204 ページ)」を参照してください。CREATE TABLE 文の構文の詳細については、「[CREATE TABLE 文](#)」を参照してください。テーブル構造の変更については、「[ALTER TABLE 文](#)」を参照してください。どちらの文についても、『リファレンス：文とオプション』の「[第1章 SQL 文](#)」に記載されています。

エンティティ整合性の確保

テーブルのローが挿入または更新されると、データベース・サーバは、そのテーブルのプライマリ・キーの有効性が確実に保たれる (テーブル内の各ローがプライマリ・キーによってユニークに識別される) ようにします。

例 1 デモ・データベースの `employee` 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、新しい `employee ID` の値がユニーク、かつ `NULL` でないことを、`IQ` はチェックします。デモ・データベースの構造図については、『`Sybase IQ` の概要』を参照してください。

例 2 デモ・データベースの `sales_order_items` 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。

このテーブルは注文されたアイテムの情報を格納します。カラムには注文の番号が入っています。1 つの注文にはいくつかのアイテムが含まれているため、このカラムだけではプライマリ・キーにはなりません。追加の `line_id` カラムはアイテムに対応する行を識別します。`id` カラムと `line_id` カラムの 2 つを組み合わせると、アイテムをユニークに指定できるため、プライマリ・キーを作成できます。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。

`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` または `DELETE` の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。また、`Sybase IQ` では、ユーザ指定のデータ整合性に違反するローも拒否させることができます。

参照整合性の宣言

外部キーの関係を有効にするには、外部キーのエントリを参照先テーブルのローのプライマリ・キー値に対応させてください。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、**候補キー**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**参照キー**と呼ばれます。

外部キーの定義

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。

プライマリ・キーの作成については、「[プライマリ・キーと外部キーの作成](#)」(204 ページ)を参照してください。

例

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を想定できます。

```
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・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わからなくなります。これは必須の外部キーです。

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、Sybase IQ は、プライマリ・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)。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。詳細については、「[参照整合性チェックの無効化](#)」(395 ページ)を参照してください。

❖ **強制力のない既存の外部キーによる参照整合性の適用**

- 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る場合は、ALTER TABLE 文を使用して削除します。例を示します。

```
ALTER TABLE DEPT DROP FOREIGN KEY MGR_EMPNO;
ALTER TABLE EMPLOYEE DROP FOREIGN KEY DEPT;
```

前述のスキーマでは、強制力のない外部キー制約である、DEPT(DeptNo)を参照する MGR_EMPNO と 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。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) ;
```

例3 新しいテーブルを作成する場合は、参照整合性は次のように適用します。

❖ **新しいテーブルでの参照整合性の適用**

- 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役割名は 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);
```

次に示すのは、外部テーブルの作り方の別の例です。この文では、同じ外部キーに対するユーザ指定の役割名は EMP_DEPT です。

```
CREATE TABLE EMPLOYEE(EmpNo int primary key,  
DeptNo int,  
LastName varchar(20),  
FirstName varchar(20),  
Salary int,  
FOREIGN KEY EMP_DEPT(DeptNo) REFERENCES DEPT(DeptNo));
```

- 3 外部キー制約を追加します。例を示します。

```
ALTER TABLE DEPT ADD FOREIGN KEY MGR_EMPNO(Mgr) REFERENCES  
EMPLOYEE (EmpNo);
```

例4 外部キー制約を削除する方法は次のとおりです。

- 1つ目の CREATE TABLE の例のように、役割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役割名は DEPT です。

```
ALTER TABLE EMPLOYEE DROP FOREIGN KEY DEPT;
```

外部キーが複数あり、役割名が不明の場合は、`sp_iqconstraint` プロシージャを使用して役割名を表示できます。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、およびプロシージャ』の「[第7章 システム・プロシージャ](#)」を参照してください。

- 2つ目の CREATE TABLE の例では、役割名 EMP_DEPT を割り当てたので、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;
```

- 例 5 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
```

- 例 6 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;
```

- 例 7 このデータベースに、事業所をリストする office テーブルも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。

参照整合性の損失

データベースに対して、次のいずれかが行われた場合、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。

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、Sybase IQ は、UPDATE や DELETE を許可しません。

同時処理の制御

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 9-1: IQ エラーが返る同時処理

最初の要求	並行するトランザクションの要求
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/ ALTER TABLE DROP 外部テーブルへの外部キー	関連付けられたプライマリ・テーブルの削除可能なローに対する 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

Sybase IQ は、古いバージョンの外部テーブルかプライマリ・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、別のトランザクションが ALTER TABLE ADD 外部キーまたは DROP 外部キーを要求した場合にも、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、Sybase IQ では以下が実行できます。

- 1つまたは複数の外部テーブルと共有プライマリ・テーブルに対する LOAD/INSERT の同時実行。
- 外部テーブルに対する LOAD/INSERT と、1つまたは複数の別の外部テーブルに対する DELETE/UPDATE/TRUNCATE TABLE の同時実行。
- 2つ以上の外部テーブルに対する DELETE/UPDATE/TRUNCATE TABLE 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 DELETE/TRUNCATE TABLE と、共有プライマリ・テーブルに対する DELETE/UPDATE/TRUNCATE TABLE 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 処理によって削除される場合)。

外部テーブルとプライマリ・テーブルに対する同時処理

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 LOAD/INSERT/UPDATE 処理と、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、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 DELETE とプライマリ・テーブルに対する DELETE/UPDATE 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す。

外部テーブルとプライマリ・テーブルに対する同時処理を理解するために、次のような例を考えてみましょう。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・キー一意性制約のいずれかです。

表 9-2 は、外部テーブルとプライマリ・テーブルの両方に対して、許可される処理とエラーを返す処理を示します。表 9-2 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 9-2: 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	DELETE/ TRUNCATE TABLE ftab1	UPDATE ftab1 (fk1,fk2)	新しいイン デックス非 FK ftab1 の挿入 (fk1,fk2)	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 ftab2 TRUNCATE TABLE ftab2	可能	可能	可能	可能	可能	可能
DELETE ptab TRUNCATE TABLE ptab	Error	可能	Error	可能	Error	Error
UPDATE ftab2(fk1,fk2)	可能	可能	可能	可能	可能	可能
UPDATE ptab (pk1,pk2)	Error	可能	Error	可能	Error	Error
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	Error	Error
古いバージョンの ftab2 は使用されて いない	なし	なし	なし	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、同時実行の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

Sybase IQ の `DISABLE_RI_CHECK` オプションを使用すれば、必要に応じて参照整合性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することをおすすめします。詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」の「[DISABLE_RI_CHECK オプション](#)」を参照してください。

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のすべては、次に示すシステム・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	検査制約は、SYS.SYSTABLE の view_def カラムに入っている。view_def には、ビューを作成する CREATE VIEW コマンドが入っている。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・キーである。

各システム・テーブルの内容の詳細については、『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「[第 8 章 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。Sybase Central または dbisql 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。

この章について

この章では、スナップショット・バージョン管理と呼ばれる、Sybase IQ におけるトランザクション処理の方法と、この方法がパフォーマンスとデータベース管理の他の面に及ぼす影響について説明します。

内容

トピック名	ページ
トランザクションとバージョン管理の概要	397
バージョン管理による矛盾の回避	408
ロックの仕組み	408
ロックの管理	412
独立性レベル	413
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	415
パフォーマンスの影響	420
トランザクション内のカーソル	422

トランザクションとバージョン管理の概要

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できます。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。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。

トランザクションの概要

トランザクションとは単に、複数の SQL 文をまとめたものです。各トランザクションは、ある一貫した状態から別の状態へとデータベースを変更するタスクを実行します。これらのトランザクション単位は、メディア障害やシステム障害からデータベースを保護し、データの一貫性を保持する上で重要な役割を果たします。

作業の論理単位としてのトランザクション

トランザクションは、作業の論理単位です。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。つまり、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、それまで行つた変更を取り消す、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、各ユーザの作業を小さいブロックに分割します。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。

注意 Sybase IQ でのトランザクションの処理方法は、SQL Anywhere が IQ を使用しない場合に行う処理方法とはまったく異なります。この章では、Sybase IQ でのトランザクションの処理方法について説明します。Anywhere のみのデータベースで作業を行っている場合、トランザクションとロックの詳細について、『SQL Anywhere サーバ - SQL の使用法』を参照してください。

トランザクションの使用

Sybase IQ では、コマンドをトランザクションにグループ化できます。多くの場合、IQ 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。

トランザクションの開始

トランザクションは次のいずれかのイベントで自動的に開始します。

- データベースへの接続に続く最初の文
- 前のトランザクションの終りに続く最初の文

Sybase IQ は、Adaptive Server Enterprise との互換性のために、BEGIN TRANSACTION などの Transact-SQL コマンドもサポートします。IQ では、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。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「付録 A 他の Sybase データベースとの互換性」を参照してください。

トランザクションの完了

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 COMMIT 文で確定する
- トランザクションで行われたすべての変更を ROLLBACK 文で取り消す
- 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
- オートコミットする文を実行する

ALTER、CREATE、DROP などのデータベース定義コマンドは、すべてオートコミットを実行します。2 つの dbisql オプションを指定して、コミットを自動的に実行させることもできます。

トランザクションをコミットして、データをディスクへ書き込む

書き込み操作を実行する場合、Sybase IQ はデータをディスクにすぐには書き込みません。その代わりに、データ・キャッシュにデータを書き込みます。データ・キャッシュは、データベースから読み込んだ使用中のページを格納するメモリ領域です。キャッシュに対して読み込みと書き込みを行うことにより、Sybase IQ によるディスクへのアクセス回数を減らすことができます。これは、IQ の高速パフォーマンスにおける重要な部分です。

最終的に、IQ はダーティ・ページ (更新されたページ) をディスクに書き込む必要があります。Sybase IQ は、トランザクションがコミットするたびにダーティ・ページをディスクに書き込みます。この方法は、IQ がトランザクション・ログにデータ挿入のログを取る必要がないため、IQ ユーザにとって大変便利です。IQ で典型的である非常に大きな挿入のログを取らないことにより、ユーザはディスクとパフォーマンス・コストを大幅に節約できます。

トランザクションをさらに分割

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、セーブポイントを使用して他のアクションを発生させたりできます。セーブポイントの詳細については、「[トランザクション内のセーブポイント](#)」(416 ページ) を参照してください。

トランザクションに関する情報の表示

sp_iqtransaction ストアド・プロシージャは、トランザクション処理のスナップショットを表示します。たとえば、作成および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です。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。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「[第 7 章 システム・プロシージャ](#)」の「[sp_iqtransaction プロシージャ](#)」を参照してください。

同時実行性の概要

Sybase IQ は同時に複数のトランザクションを実行できます。この機能を、同時実行性といいます。データベース・サーバ内の特殊なメカニズムにより、IQ の複数のトランザクションを相互に干渉することなく同時に実行できます。

IQ での同時実行性の仕組み

1 つのトランザクションを構成する SQL 文の実行中に、データベース・サーバは他のトランザクションの文のいくつかまたはすべてを実行できます。同時に処理されるトランザクションは同時的であると言います。

Sybase IQ の同時実行性の方法は、特にデータ・ウェアハウス用に設計されています。通常データ・ウェアハウス環境では、多くのユーザがデータベースから読み込みを行います。データベースの更新を行う必要があるのは DBA のみです。ただし、他のユーザがクエリ結果の要求や受信を続行中に、これらの更新の実行が可能でなければならない場合はよくあります。

Sybase IQ では、1 つのデータベースに対して、複数のユーザが同時に多数接続できます。接続している複数のユーザまたはアプリケーションからのトランザクションを、同時に処理することもできます。

Sybase IQ では、すべてのデータベース操作が 1 つのトランザクション内で発生し、しかもこれらの操作が相互に干渉しないことを保証します。これは、テーブル・レベルでのアクセス制限の設定と、スナップショット・バージョン管理（「[バージョン管理の概要](#)」を参照）と呼ばれる技術の使用によって実現されます。1 つのテーブルでは、IQ は複数の読み取りトランザクションの同時処理を行うことができますが、書き込みトランザクションは 1 つしか実行できません。この方法によりデータベースの内部的な一貫性が保持されます。

バックアップの同時実行性

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。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。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。バックアップ操作とリストア操作の同時実行性の問題の詳細については、「[第 12 章 データのバックアップ、リカバリ、アーカイブ](#)」を参照してください。

同時実行性の利点

データ・ウェアハウスは、多数のユーザが共有する情報の共通レポジトリです。これらのユーザは情報に頻繁にアクセスします。ユーザの作業を妨げないように、データベース・サーバは多数のトランザクションを同時処理できる必要があります。

さらに、多くのサイトでデータベースへの頻繁な更新も必要となります。稼働率の高いサイトでは、DBA は排他的アクセスが可能になる時点まで挿入や削除を延期することはできません。同様に、他のユーザの作業を中断させることなくデータベースを定期的にバックアップできることが重要です。

Sybase IQ の同時実行性の方法により、クエリを行うユーザは情報に直ちにアクセスでき、受信する情報の安全性と精度が保証されます。

バージョン管理の概要

Sybase IQ は、トランザクションを同時に処理できるようにするために、「スナップショット・バージョン管理」を使用します。

スナップショット・バージョン管理をカメラで撮るスナップショットに例えることができます。対象物のスナップショットを撮った場合、その対象物のある瞬間のイメージを手に入れることができます。同様に、IQ がデータベース内のオブジェクトのスナップショットを撮る場合も、そのオブジェクトのある瞬間のイメージを保持します。

ただし、カメラと違い、IQ はイメージが変更されるたびにオブジェクト全体のコピーを作成する必要がありません。代わりに、イメージ (データベース・ページ) の変更されている部分のみをコピーします。変更がないデータベース・ページはデータベース内のすべてのアクティブなバージョンで共有されます。

IQ は、接続、コミット、またはロールバックの後で最初のコマンドが実行されたときに、スナップショットを撮ります。ユーザは、**BEGIN TRANSACTION** コマンドを明示的に実行すれば、スナップショットを早目に撮らせることができます。そのトランザクションを通して、オブジェクトから読み取りを行うユーザは未変更のイメージ、つまりスナップショット・バージョンを参照します。

テーブル・レベルのバージョン管理

Sybase IQ では、ユーザが参照可能なレベルでのバージョン管理の単位は、テーブルです。テーブル・レベルのバージョン管理が Sybase IQ で有効なのは、次の理由のためです。

- IQ データ構造はテーブル・レベルでカラムのデータを集約する
- ほとんどの IQ 挿入と削除はテーブル全体でデータを書き込む

テーブル・レベルのバージョン管理により、Sybase IQ は、書き込み操作が発生するレベル、およびクエリの結果がフォーカスされるレベルで、データへのアクセスを制御できます。

ただし、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。この方法によりシステム・リソースを節約できます。

IQ テーブルのなかには、数百万ページのデータから構成されるものもあります。そのテーブルを更新する場合、ほんの数パーセントのページに書き込むだけで済みます。テーブル全体の各バージョンの完全なコピーを保持するには、膨大なディスク領域が必要となります。Sybase IQ は、テーブル・バージョンで未更新のページを共有することにより、ディスク領域を節約します。

Sybase IQ のテーブル・レベルのバージョン管理は、マルチプレックス・データベースにまで拡張されています。書き込みサーバに対して、新しいバージョンのテーブルを作成するトランザクションがコミットされると、新しいバージョンが使用できることを示す制御情報がセカンダリ・サーバに直ちに送信されます。セカンダリ・サーバ上で開始する新しいトランザクションでは、書き込みサーバ上の新しいトランザクションと同様に、自動的に新しいバージョンのテーブルを参照します。

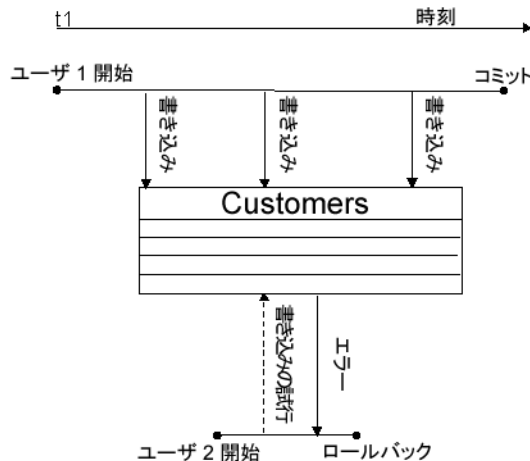
テーブル・レベルでの 1 人の書き込みユーザと複数の読み取りユーザ

IQ では 1 つのテーブルに対して、1 人のユーザが挿入や削除の書き込みアクセスを行い、複数の読み込みユーザがクエリを同時に発行できます。

図 10-1 に示すような状況を仮定します。まず、ユーザ 1 がトランザクションを開始し、customer テーブルにデータの挿入を開始します。ユーザ 1 のトランザクションが開いている間は、他のユーザは customer テーブルに書き込みできません。customer テーブルに書き込もうとするトランザクションは、ユーザ 1 のトランザクションがコミットするまでの間エラーを受信します。

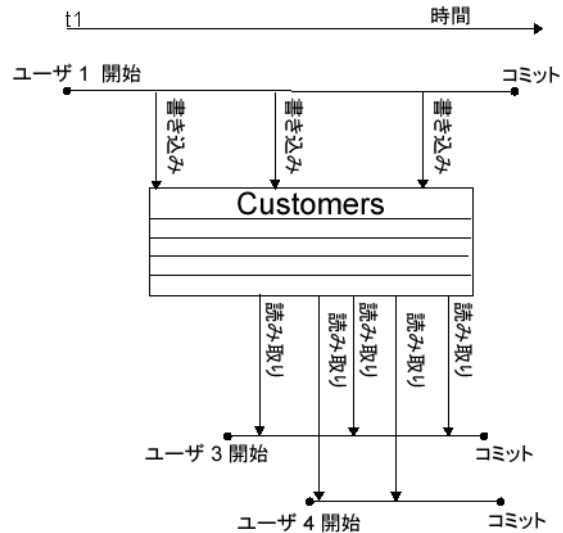
図 10-1 では、ユーザ 2 はユーザ 1 のトランザクションがコミットする前に書き込もうとしてエラーを受信します。ユーザ 2 のアプリケーションは、トランザクションをロールバックするか、別のテーブルに書き込みを行うかを決定します。ただし、ユーザ 2 は、同じトランザクション内で customer テーブルへの書き込みをもう一度行うことはできません。

図 10-1: 一度に 1 人の書き込みユーザ



一方、他のユーザは **Customers** テーブルからいつでも読み取りを行うことができます。このように、データベース管理者がテーブル・データの挿入や削除を行っている間もクエリ処理は可能です。図 10-2 では、ユーザ 3 とユーザ 4 は、ユーザ 1 の書き込みトランザクションが開いている間に、**Customers** テーブルに対してクエリを実行できます。

図 10-2: 1 人の書き込みユーザ、複数の読み取りユーザ

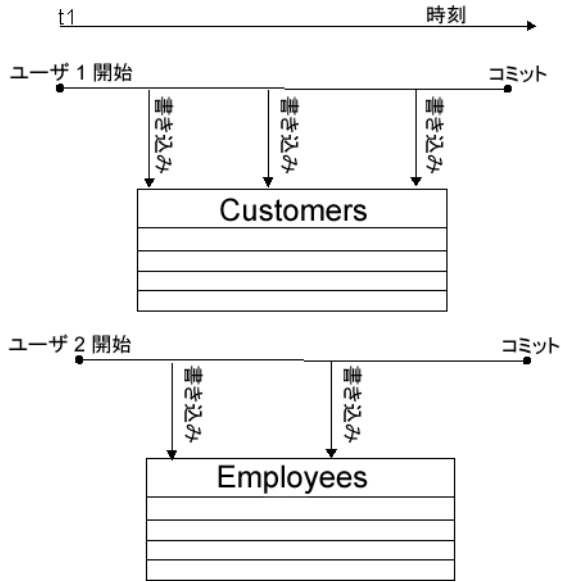


データベースでの複数の書き込みユーザと読み込みユーザ

IQ データベース内では、各書き込みユーザがデータの挿入や削除を行っているテーブルが異なっていれば、複数の読み込み専用ユーザと読み込み／書き込みユーザが同時に操作を行うことができます。したがって、たとえば、図 10-3 に示すように、ユーザ 1 のトランザクションが **Customers** テーブルにデータの挿入や削除を行っている間、ユーザ 2 は **Employees** テーブルにデータをロードするトランザクションを開始できます。同時に、他のユーザは、これらのテーブルの両方、またはデータベース内の他のテーブルにクエリを発行するトランザクションを実行できます。

一般に、読み取り専用ユーザは任意のセカンダリ・サーバに接続し、読み取り／書き込みユーザは書き込みサーバに接続します。読み取り／書き込みユーザはクエリ・サーバにも接続できますが、変更できるのは、グローバル・テーブルまたはテンポラリ・テーブルと SQL Anywhere ベース・テーブル内のローカル・データだけです。

図 10-3: 異なるテーブルへの同時挿入



単一のテーブルに対するデータ定義操作は、すべての他の読み取りユーザや書き込みユーザをそのテーブルからロック・アウトします。詳細については、「DDL 操作のロック」(410 ページ)を参照してください。

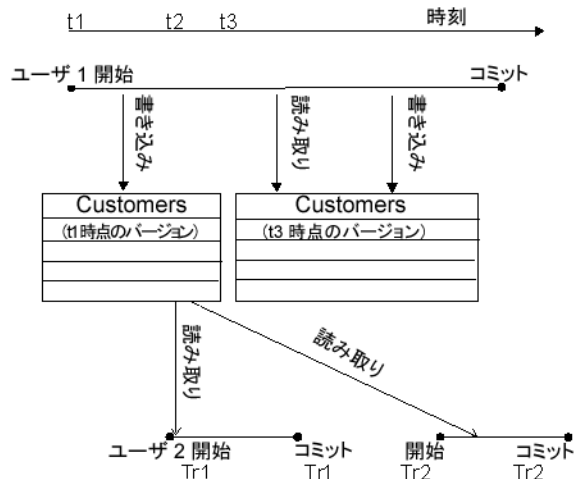
トランザクションによるコミットされたデータの使用

書き込みトランザクションがコミットすると、コミットされたデータが結果として生じます。各トランザクションはトランザクションが開始した時点で、データベースの最新のコミットされたバージョンを使用します。トランザクションがコミットするまでそのバージョンを使用します。

トランザクションの開始時刻は、開始タイムスタンプと呼ばれます。開始タイムスタンプは、トランザクションの最初の読み込み前の任意の時刻です。トランザクションが行う挿入や削除はスナップショットに反映されます。したがって、トランザクションを実行中のユーザにとっては、そのトランザクションがデータをテーブルに書き込んだ後もう一度読み込むときは常にスナップショットのイメージが変更します。それ以外のすべてのユーザにとっては、イメージはトランザクションをコミットするまで変わりません。

つまり、各トランザクションは信頼性のある状態のデータのスナップショットで開始します。クエリの発行時に参照するデータのスナップショットは、読み込み中のテーブルを別のユーザが更新している場合でも変更されません。たとえば、図 10-4 では、ユーザ 1 の書き込みトランザクションが開始するときは、最新のコミットされた customer テーブル・バージョンを使用します。ユーザ 2 のトランザクションは、ユーザ 1 の書き込みの開始後、ユーザ 1 のコミット前に、開始します。したがって、ユーザ 2 の最初のトランザクション (Tr1) はユーザ 1 が行う更新はまったく参照しません。ユーザ 2 の 2 番目のトランザクションは、ユーザ 1 がコミットした後に開始するので、ユーザ 1 が行った変更をすべて参照します。

図 10-4: トランザクションによるコミットされたデータの使用



書き込みユーザが参照できるデータは、その書き込みユーザが行った変更分だけが変更されます。ある書き込みユーザが参照できるデータは、その書き込みユーザのトランザクションがコミットするまで他のトランザクションによって変更できません。たとえば、図 10-4 では、ユーザ 1 はデータの挿入、クエリの発行、データの削除の順に実行します。このクエリ結果には、ユーザ 1 が直前に行った挿入が反映されます。

ユーザ 1 のトランザクションの開始後、コミットする前に開始する他のトランザクションは、ユーザ 1 のトランザクションの開始時刻後のデータ・バージョンを参照できます。最新の変更は、まだ変更が未コミットのため、参照できません。ユーザ 1 のトランザクションがコミットすると直ちに、新しいトランザクションはユーザ 1 の変更を参照できます。

読み込みトランザクションのコミットのタイミングがバージョンに及ぼす影響

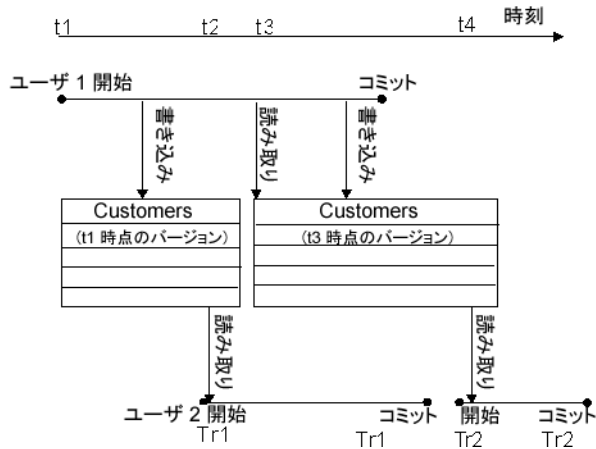
読み取りトランザクションは既存の書き込みトランザクションが参照するデータに影響を与えないのに対し、読み取りトランザクションをコミットすると他のトランザクションに影響があります。

- あるユーザの読み込みトランザクションのコミットが、同時実行の書き込みトランザクションがコミットしそのユーザが新しい読み込みトランザクションを開始する前に行われる場合、バージョンは同じままです。
- 読み込みトランザクションのコミットが、同時実行の書き込みトランザクションのコミットの後に行われる場合、新しいトランザクションは読み込み専用の場合も読み込み／書き込みの場合も、新しいバージョンを使用します。

図 10-4 (405 ページ) は、1 番目の例を示します。ユーザ 2 の両方のトランザクションは、ユーザ 1 のトランザクションの開始時に使用したバージョンと同じバージョンを使用します。データのコミットされた最新のバージョンであるからです。

図 10-5 は、2 番目の例を示します。今度は、ユーザ 2 の最初の読み込みトランザクション (Tr1) のコミットは、ユーザ 1 の書き込みトランザクションの後に行われます。ユーザ 2 の 2 番目のトランザクション (Tr2) が開始する場合、ユーザ 1 からのコミットされたデータが反映される新しいバージョンを使用します。

図 10-5: 読み込みトランザクションのコミットの影響



複数のトランザクションにまたがるホールド・カーソル

トランザクションが常にコミットされた最新のバージョンを使用するという規則の唯一の例外は、ホールド・カーソルを使用するトランザクションの場合です。ホールド・カーソルは、複数のトランザクションにまたがるのが可能なため、処理方法が異なります。詳細については、「トランザクション内のカーソル」を参照してください。

Sybase IQ のバージョン追跡方法

Sybase IQ は、メタデータに存在する各データベース・オブジェクトにバージョン識別子を割り当てます。バージョン識別子は単一のコマンドより長い寿命を持っています。IQ ではこれらのバージョン識別子を使用して、任意のデータベース・オブジェクトへの書き込みが常にオブジェクトの最新のバージョンに基づいて行われるようにします。データベースの各アクティブ・バージョンをディスクに保管します。

古いバージョンがアクティブ・トランザクションで不要になると、Sybase IQ はそれをキャッシュから削除します。バージョンは、それを使用するトランザクションが次のいずれかを実行するまで必要です。

- コミット
- ロールバック
- そのバージョンを解放する **RELEASE SAVEPOINT** コマンドの発行

加えて、マルチプレックスでないデータベースについては、Sybase IQ は、他のトランザクションが特定のバージョンのテーブルを使用できないことを認識し、最も古いアクティブ・トランザクションのコミットやロールバックを待たずにその領域を直ちに解放します。この機能は、小さな挿入、削除、更新を大量に実行する場合に特に有用です。

セーブポイントに関する定義、解放、ロールバックの詳細については、「[トランザクション内のセーブポイント](#)」(416 ページ) を参照してください。

テンポラリー・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ー・テーブルは、「グローバル・テンポラリー・テーブル」と呼ばれます。グローバル・テンポラリー・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・テンポラリー・テーブルのローは、**COMMIT** の実行時に削除されます。テンポラリー・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。

「ローカル・テンポラリー・テーブル」は、データベースで作成されるのではなく、宣言されます。1 人のユーザのみが、ローカル・テンポラリー・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・テンポラリー・テーブルを宣言すると、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・テンポラリー・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、次のエラー・メッセージが表示される場合があります。“インデックスを作成するには、ローカル・テンポラリー・テーブル <テーブル名> をコミットしなければなりません。”

バージョン管理の目的から、Sybase IQ はベース・テーブル (メイン・データベース・テーブル) とグローバル・テンポラリー・テーブルを区別しません。これは、テンポラリー・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 つのテンポラリー・テーブルに対して開くことはないからです。

バージョン管理による矛盾の回避

バージョン管理を行わないと、同時実行の読み込みや書き込み操作によりデータベース内で矛盾が発生する可能性があります。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。

Sybase IQ 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。この方法により、IQ は矛盾を回避します。

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、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。IQ はトランザクションの開始や終了を自動的にを行い、読み込み操作と書き取り操作が相互に干渉しないようにします。

ロックの仕組み

Sybase IQ では、ユーザが要求する操作タイプに基づいて、すべてのロックが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は、ロックを保持していると言います。

あるテーブルが Sybase IQ でロックされる場合、他のトランザクションはそのテーブルに書き込みアクセスを行うことができませんが、読み込みアクセスは可能です。データ定義操作は、この一般的な読み込みアクセスの例外となります。詳細については、後述の説明を参照してください。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。

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。

WRITE ロックは、LOCK TABLE 文を使用して、新しいトランザクション内のテーブルのセットで予約できます。LOCK TABLE によって、現在のトランザクションがコミットされ、ロックが使用可能になるまでトランザクションをキューに入れておくことができます。構文については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「LOCK TABLE」を参照してください。

DML 操作のロック

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、Sybase IQ 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込みユーザを許可します。この規則には次のような意味があります。

- 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、1 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。

Sybase IQ では、SHARE、WRITE、および EXCLUSIVE の各ロックをキューに入れることができるので、テーブルを指定した期間ロックすることが可能です。WRITE ロックは、一度に複数のテーブルに対して使用できます。

その後で使用する DML 文でバージョン・エラーが返されるのを防ぐには、WRITE ロックを変更予定の 1 つまたは複数のテーブルに対して予約します。詳細と構文については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「LOCK TABLE 文」を参照してください。

デッドロックの場合、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は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。SYNCHRONIZE JOIN INDEX の実行が“テーブル x には、その他のテーブルにジョインするデータがありません。”というエラーと共に失敗した場合、ジョイン・インデックスとジョイン仮想テーブルに関与しているすべてのデータベース・テーブルは、明示的に接続を切断するか、COMMIT 文または ROLLBACK 文を発行しないかぎり、WRITE モードでロックされたままとなります。また、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、Sybase IQ によって SQL Anywhere エラー -210 が返されます。

DDL 操作のロック

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 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、起動できません。

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。

- これらのコマンドには、完了時の自動的な COMMIT が含まれます。
- 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、エラーを受信する。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 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。

複数の DDL コマンドを同時に実行しようとする、ユーザは次のエラー・メッセージを受信する場合があります。

```
Cannot perform DDL command now on table <tablename> as a DDL
command is already in progress on that table.
```

CREATE DBSPACE または DROP DBSPACE コマンドの処理中に、ユーザが明示的に CHECKPOINT コマンドを発行すると、チェックポイントは失敗し、次のメッセージが発行されます。

```
Run time SQL Error
```

CHECKPOINT コマンドが処理中の場合、CREATE DBSPACE または DROP DBSPACE コマンドを発行したユーザは次のメッセージを受信します。

```
Cannot perform requested command as there is a
CHECKPOINT command in progress.
```

削除中に `CREATE DBSPACE` を発行したユーザは次のメッセージを受信します。

```
Cannot perform requested command as there is a  
DROP DBSPACE command in progress.
```

作成中に `DROP DBSPACE` を発行したユーザは次のメッセージを受信します。

```
Cannot perform requested command as there is a  
CREATE DBSPACE command in progress.
```

テンポラリ・テーブルに関する特別な規則については、「[テンポラリ・テーブルのバージョン管理](#)」(407 ページ)を参照してください。

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、エラーを受信します。

このエラーが発生すると、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。

あるトランザクションがジョイン・インデックスの一部を構成するテーブルの定義を変更する場合、そのインデックスにジョインされたカラムを含むテーブルがすべてロックされます。オリジナルの書き込みトランザクション内の特定のカラムがジョインされているかどうかに関係なく、このような結果が発生します。

インデックス作成コマンドの場合、これらの規則に例外が 1 つあります。`CREATE INDEX` と `CREATE JOIN INDEX` は、インデックスの作成によって影響を受けるテーブルに対して `SELECT` と同時に実行できます。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またはジョイン・インデックスを使用できません。

`GRANT`、`REVOKE`、`SET OPTION` 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。オプション設定の詳細については、『リファレンス：文とオプション』を参照してください。

プライマリ・キーとロック

あるテーブルを更新できるユーザは 1 人のため、プライマリ・キーの生成を行っても、同時実行による矛盾は発生しません。

ロックの管理

ロックとロック解除は自動的に行われますが、Sybase IQ では、ストアド・プロシージャ、IQ モニタ、データベースとサーバのオプションを使用してロックを管理できます。

アクティブなロックの表示

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ために書き込みが失敗した場合、次のような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.
```

そのテーブルをロックしたユーザを確認するには、`sp_iqtransaction` プロシージャを使用します。`sp_iqtransaction` の出力から TxnID2 を見つけ出し、同じローの出力からユーザ名を探します。

`sp_iqlocks` プロシージャは、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。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

構文の詳細と出力例については、『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「[第 7 章 システム・プロシージャ](#)」の「[sp_iqlocks プロシージャ](#)」を参照してください。

`sp_iqtransaction` プロシージャでは、トランザクションについてさらに詳細な情報が得られます。

ロック競合の管理

ロードまたはクエリのパフォーマンス問題が、ロック競合によって引き起こされる場合があります。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、IQ またはオペレーティング・システムが持つ機能を使用します。

- `-contention` オプションを指定して 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。

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。

Sybase IQ では、IQ メイン・キャッシュとテンポラリ・キャッシュを分割することでロック競合を抑えています。デフォルトの分割レベルは IQ サーバの CPU の数に基づいて決まり、ほとんどの状況ではそのレベルで十分です。ロック競合が疑われる場合は、分割レベルを直接制御すると役立つ場合があります。それには、サーバ起動オプションの `-iqpartition` か、データベース設定オプションの `Cache_Partitions` を設定します。これらのオプションの設定については、『リファレンス：文とオプション』の「[コマンド・ライン・スイッチの使用](#)」(27 ページ) と「[CACHE_PARTITIONS オプション](#)」を参照してください。

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。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。

テーブルに対する書き込みロック競合の管理の例については、「[付録 14 トラブルシューティングのヒント](#)」を参照してください。

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。IQ 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。

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の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、常に読み込みが可能
- 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

Sybase IQ が提供する独立性レベルにより、いくつかの種類の矛盾を回避できます。最も頻繁に発生する矛盾を次に示します。

- **ダーティ・リード** トランザクション A がオブジェクトを変更しましたが、変更をコミットまたはロールバックしていません。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取ります。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。

Sybase IQ 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。また、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。こうすることにより、Adaptive Server IQ は上記の矛盾をすべて回避します。

IQ では、**SET OPTION ISOLATION_LEVEL** を使用して独立性レベルを 0、1、2、3 (ANSI レベル 1、2、3、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。

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、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。データベース・リカバリでは常に、すべてのコミットされたトランザクションがリカバリされます。データベース障害の時点でコミットされていないトランザクションは、リカバリされません。

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。これらのコマンドは、チェックポイントの設定、セーブポイントの設定と解放、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、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。IQ は、データベースのリカバリが必要な場合にこの情報を使用します。

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。その代わりに、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・ローだけを書き込みます。

「第 1 章 Sybase IQ システム管理の概要」で説明したように、Sybase IQ は OLAP データベースです。単一の 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、Sybase IQ 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。各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・ページをディスクに書き込みます。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ほうがずっと効率的です。

リカバリ時のチェックポイントの役割

システム障害やメディア障害からリカバリするためには、Sybase IQ は内部的に一貫性のあるポイントにデータベースをリストアできる必要があります。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。各チェックポイントで IQ がディスクに書き込む情報は、リカバリ処理に必要なものです。

チェックポイントの発生時点

Sybase IQ 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を設定することもできますが、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 分に設定されています。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。構文については、『リファレンス：文とオプション』の「SET OPTION 文」を参照してください。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。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、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。

リカバリ時のチェックポイントの詳細については、「[リカバリにおけるトランザクション情報の役割](#)」を参照してください。

トランザクション内のセーブポイント

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、トランザクション内の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、コミットしてディスク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。

ROLLBACK TO SAVEPOINT 文を使用すると、セーブポイント以降の変更をすべて取り消せます。セーブポイントとロールバックの詳細については、「[セーブポイントの名前付けとネスト](#)」を参照してください。

セーブポイントの解放

RELEASE SAVEPOINT 文の実行後、またはトランザクションの終了後は、セーブポイントは使えなくなります。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。データは内部ではページ・レベルでバージョン付けされています。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、セーブポイントに関連付けられているページが解放されて、ディスク領域を効率よく使用できます。

セーブポイント n を解放すると、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 コマンドでは、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 を発行すると、セーブポイント以降の変更をすべて取り消せます。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 SAVEPOINT 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。次に、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。

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す。

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場合、名前を付けた自分用のセーブポイントを設定する必要があります。

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ができます。名前を付けてネストしたセーブポイントを使用して、トランザクション内に多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 の間の変更も、その前のセーブポイントにロールバックするか、そのトランザクション自体をロールバックすれば取り消せます。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、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できます。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、そのトランザクション内の全操作が取り消されます。データベースを以前の状態に戻すことになるため、データベースをロールバックすると言います。

ロールバックの発生理由

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、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ー・ストアの両方が以前の状態に戻されます。また、次のようにロックが解放されます。

- 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
- 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。

- 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。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は関係ありません。
- 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。

カーソルの詳細については、「[トランザクション内のカーソル](#)」を参照してください。セーブポイントへのロールバックの詳細については、「[セーブポイントへのロールバック](#)」を参照してください。

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、Sybase IQ は自動的にリカバリしようとしてします。

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、各永続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、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 サーバを再起動する必要があります。テンポラリー・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。サーバまたはデータベースを起動できない場合やユーザが接続できない場合は、「第 13 章 システムのリカバリとデータベースの修復」を参照してください。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` を実行することをおすすめします。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。詳細については、「第 13 章 システムのリカバリとデータベースの修復」を参照してください。

リカバリにおけるトランザクション情報の役割

Sybase IQ のリカバリのメカニズムは、データ・ウェアハウス用に設計されたものです。この環境では通常、発生するトランザクションの数は少ないですが、各トランザクションの処理に非常に時間がかかることがあります。

このモデルを最適化するため、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。次のような情報も記録します。

- 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。この情報は *transaction log* に格納します。
- バージョン・ページとディスク上の空き領域のロケーション。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、Sybase IQ は、トランザクションをすべて長時間かけて繰り返すのではなく、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、トランザクション・ログのサイズは増加し続けます。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、時間の経過とともにログが非常に大きくなる可能性があります。そのため、ログを定期的にトランケートする必要があります。稼働中のデータベース、マルチプレックス・データベース、非マルチプレックス・データベースのトランザクション・ログ・ファイルをトランケートする方法については、『パフォーマンス&チューニング・ガイド』の「[第4章 システム・リソースの管理](#)」の「[トランザクション・ログ・ファイル](#)」を参照してください。

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。

データベースのバックアップとリストアに関連するその他の同時実行性の詳細については、「[バックアップの実行](#)」(448 ページ) を参照してください。

パフォーマンスの影響

スナップショット・バージョン管理のパフォーマンスに対する影響は、最小限におさえる必要があります。他のユーザの読み込み中にデータベースを更新できることには、悪影響もありますが、柔軟な利用が可能となるということの方が、はるかに重要です。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。ディスク領域に関する項目も参照してください。
- 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、スワイパ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できるのは、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 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。

この項で説明した Sybase IQ でのリソースの使用方法の詳細については、『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」を参照してください。

重複したバージョンと削除

データを削除するには、IQ ストアに `dbspace` 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。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 ストア内に必要となります。

たとえば、各ページが 100 ローを保持しているデータベースから 10 ローを削除するとします。10 ローのそれぞれが別々のデータ・ページ上にある場合、IQ 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 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。インデックス・タイプのサイズの詳細については、「[インデックス条件：ディスク領域の使用率](#)」を参照してください。

データの削除中に領域が不足すると、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、`dbspace` 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。新しいデータベース領域の使用を必要とするデータを挿入する前に、ディスク領域の確保を必ず行ってください。

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 dbspace 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。

トランザクション内のカーソル

カーソルを使用すると、SELECT の結果をカーソルというデータ型の形式で返すことができます。カーソルはテーブルと似ていますが、1つのローを現在のローとして識別するプロパティを持っています。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、プロシージャをプログラミングする場合や、データベースへのアクセス用アプリケーションを Embedded SQL で記述する場合に非常に便利です。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。dbisql を対話的に使用している場合には使用できません。

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、その背後にあ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 コマンドはローを閲覧しますが、その順序はランダムで一貫性がありません。ORDER BY フレーズを SELECT 文に追加すると、順番を設定できます。

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。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「[第7章 システム・プロシージャ](#)」の「[sp_iqcursorinfo プロシージャ](#)」を参照してください。

カーソルとバージョン管理

カーソルを使用する場合、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。その後、時刻 y にそのテーブルを更新します。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する必要があります。

データベースのロールバック時のカーソルの動作については、「[ロールバックの影響](#)」(418 ページ)を参照してください。

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。以降の項では、トランザクション処理に重要な影響がある特定のカーソルの特性について説明します。

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」といわれます。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。不定依存カーソルでは、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。Sybase IQ は、不定依存の更新可能カーソルをサポートします。

カーソルのスクロール

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` 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いう場合は、`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、カーソルが開かれ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。

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可能です。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、それ以降の更新ではエラーが生成されます。

HOLD オプションは通常、トランザクションの時間が長い 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。

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す。

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、読み込みまたは書き込み操作の開始場所が決定します。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。Sybase IQ は位置付け更新と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、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・コマンド構文と例

カーソルの使用例など、プロシージャでのカーソルの使用方法の詳細については、『システム管理ガイド第 2 巻』を参照してください。カーソル関連のコマンドの構文については、『リファレンス:文とオプション』を参照してください。

カーソルのログを取るメッセージの制御

デフォルトでは、IQ メッセージ・ファイルにはカーソル操作のログは取られません。問題の原因を判断するためにカーソル操作の追跡が必要な場合は、LOG_CURSOR_OPERATIONS オプションをオンにすると、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。LOG_CURSOR_OPERATIONS オプションの詳細については、『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「[第 2 章 データベース・オプション](#)」を参照してください。

この章について

この章では、Sybase IQ インストール環境で各国語を扱うための設定方法、および Sybase IQ と SQL Anywhere の異なる点について説明します。

内容

トピック	ページ
各国語と文字セットの概要	425
ソフトウェアの文字セットについて	427
ロケールについて	428
照合について	429
文字セット変換について	431
各国語と文字セットに関する作業	434
互換性の問題	437
パフォーマンスの問題	438

各国語と文字セットの概要

複数の文字セットを使用する環境で作業を行う場合、または英語以外の言語を使用する場合に発生する可能性のある問題については、『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」>「SQL Anywhere のローカライズ版」の「SQL Anywhere の国際化機能」を参照してください。

データベースを作成するときは、照合順（「照合」とも呼ぶ）を指定します。照合とは、「文字セット」と、データベース内の文字の「ソート順」を組み合わせたものです。

データベース照合は、オブジェクト名（テーブル名やカラム名など）をはじめとする、データベース内のすべての文字データ型をソートし、比較するとき使用されます。

Sybase IQ によるデータベース照合のサポートでは、SQL Anywhere Collation Algorithm の容量効率および速度を利用しています。

- データベース・オプション `SORT_COLLATION` を使用すると、`ORDER BY` 式で `SORTKEY` 関数を暗黙的に使用できます。このオプションの値が有効な照合名または照合 ID に設定されている場合、`ORDER BY` 句の文字式列は、`SORTKEY` 関数が呼び出された場合と同じように取り扱われます。

詳細については、『リファレンス：文とオプション』の「第2章 データベース・オプション」の「[SORT_COLLATION オプション](#)」を参照してください。

- **SORTKEY** 関数では、Sybase Unicode Infrastructure Library (Unilib®) ではなく、International Components for Unicode (ICU) ライブラリを使用します。15.0 より前のバージョンの Sybase IQ を使用して作成されたソート・キー値には、15.0 以降のバージョンを使用して作成された値と同じ値は含まれていません。Sybase IQ の 15.0 より前のバージョンを使用して生成されたデータベースのソート・キー値を再生成してください。

詳細および構文については、『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』の「第4章 SQL 関数」の「[SORTKEY 関数 \[文字列 \]](#)」を参照してください。

- **CREATE DATABASE** パラメータ **COLLATION** では、データベースに対する照合の指定がサポートされています。

詳細については、『リファレンス：文とオプション』の「第1章 SQL 文」の「[CREATE DATABASE 文](#)」を参照してください。

データベースの照合は、オペレーティング・システムで使用される照合と一致する必要があります。Sybase Central のデータベース作成ウィザードでは、デフォルトの文字セットは ISO_BINENG です。必要に応じて、オペレーティング・システムで使用される文字セットに合わせて変更してください。

- **CP874toUTF8** ユーティリティは、CP874 文字セットのデータを UTF8 照合に変換します。これは、タイ語のために Sybase IQ でサポートされる照合です。**CP874toUTF8** ユーティリティは、ICU ライブラリを呼び出してデータ変換を実行します。このユーティリティを使用して、データを UTF8 に変換せずに CP874 文字セットのデータをロードすることもできます。

詳細については、『ユーティリティ・ガイド』の「第4章 データベース管理ユーティリティ」の「[CP874toUTF8 ユーティリティ](#)」を参照してください。

Sybase IQ では、カスタム照合はサポートされなくなっています。カスタム照合を使用するデータベースを1ステップで再構築する場合、照合は保持されます。データベースをアンロードし、作成したデータベースにスキーマおよびデータをロードする場合、用意されている照合のうちのいずれかを使用する必要があります。

データベース照合の変更点と、Sybase IQ 15.0 で廃止された照合のリストについては、『Sybase IQ 15.0 の新機能』の「第2章 Sybase IQ 15.0 の動作変更」を参照してください。廃止された照合を使用して作成された既存の 12.6 または 12.7 データベースから Sybase IQ 15.2 に移行するには、**iqunload** ユーティリティを使用します。**iqunload** の詳細については、『インストールおよび設定ガイド』を参照してください。

デフォルト照合の使用

デフォルトの操作で Sybase IQ データベースを作成すると、そのデータベースの照合は ISO_BINENG になります。この照合を使用すると、IQ データベースで最適なパフォーマンスが発揮されますが、ソート順は必ずしも適切ではありません。詳細については、「[パフォーマンスの問題](#)」(438 ページ) を参照してください。

これは SQL Anywhere とは異なり、データベースを作成するオペレーティング・システムの文字セットに応じて、新しいデータベースのデフォルトの照合が類推されるわけではありません。

このデフォルトの方法でシステムを設定できない場合は、データベースで使用する照合を決定し、データベース・システム間で整合性を保ったままデータを交換できるように文字セット変換を使用するかどうかを判断してください。

ソフトウェアの文字セットについて

シングルバイトおよびマルチバイトの文字セットなど、各国の言語と文字セットに関するソフトウェア問題については、『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」の「文字セットの知識」を参照してください。

Windows でのコード・ページ

多くの言語では、文字の数はシングルバイト文字セットで扱える程度です。シングルバイト文字セットでは、各文字が 1 「バイト」(2 桁の 16 進数) で表されます。

シングルバイト文字セットで表現できるのは、最大 256 文字です。アクセント記号の付いた文字を含め、世界で使用されているすべての文字を 1 つのシングルバイト文字セットで表現することはできません。しかしこの問題は、1 つまたは複数の言語に適する文字セットをそれぞれ記述した「コード・ページ」の開発により解決されました。たとえば、Code Page 869 はギリシャ語の文字セット、Code Page 850 は各国語でさまざまな文字を表現するのに適した国際的な文字セットになっています。

Windows での ANSI および OEM コード・ページの詳細については、『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」>「文字セットの知識」の「Windows 環境の ANSI コード・ページと OEM コード・ページ」を参照してください。

サポートされるコード・ページのリストについては、『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」>「文字セットと照合の参考情報」の「サポートされている照合と代替照合」を参照してください。

照合を使用した文字のソート

データベースの照合順は、文字のアルファベット順という概念をもとに、その範囲を広げて数字やスペースを含めた文字セットのすべての文字に対応したものです。

各ソート位置に複数の文字を関連付ける

各ソート位置には複数の文字を割り当てることができます。これは、たとえばアクセント付きの文字をアクセントなしの文字と同じに扱いたいときなどに便利です。

ソート位置が同じ 2 つの文字はデータベースではまったく同じものとして扱われます。したがって、照合で *a* と *e* が同じソート位置に割り当てられている場合、次のような結果になります。

```
WHERE col1 = 'want'
```

上のような探索条件を持つクエリでは、`col1` のエントリが `went` であるローも条件を満たします。

ソート位置ごとに大文字と小文字を指定できます。大文字と小文字を区別するデータベース (Sybase IQ データベースではこれがデフォルト) では、大文字と小文字が別のもので扱われます。区別されないデータベースでは、同じものとして扱われます。

ヒント

ドイツ語のシステムのデフォルト照合を選択するコードでは、1252DEU ではなく 1252LATIN1 を選択してください。1252DEU ではウムラウトのある文字とない文字が区別されるのに対し、1252LATIN1 では区別されません。

1252LATIN1 では Muller と M 罅 ler は同じとみなされますが、1252DEU では違うとみなされます。1252DEU ではウムラウトが付いた文字は別の文字と判断されるため、ob、öa、というアルファベット順になります。

マルチバイト文字セットの最初のバイトによる照合順

マルチバイト文字のソート順は、最初のバイトにのみ設定できます。最初のバイトの値が同じ文字は、それ以降のバイトの 16 進値によってソートされます。

ロケールについて

データベース・サーバおよびクライアント・ライブラリは、「ロケール定義」を使用して、言語と文字セットの環境を認識します。詳細については、『SQL Anywhere サーバ - データベース管理』の「データベースの設定」>「国際言語と文字セット」の「ロケールの知識」を参照してください。

照合について

用意されている照合、および特定の環境下で使用する照合に関する推奨事項については、『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」の「照合の知識」と「文字セットと照合の参考情報」を参照してください。

照合の表示

現在使用されている照合を取得するには、コマンド・プロンプトで次のコマンドを入力します。

```
SELECT * FROM SYS.SYSCOLLATION
```

サポートされる照合、代替の照合、および推奨される照合については、『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」>「文字セットと照合の参考情報」の「サポートされている照合と代替照合」と「推奨文字セットと照合」を参照してください。

ANSI か OEM の選択

Sybase IQ の照合は、ANSI または OEM として指定されたコード・ページに基づいています。ほとんどの場合、ANSI コード・ページの使用が推奨されます。OEM を使用する場合は、ユーザのクライアント・マシンの OEM コード・ページに合うコード・ページを選択します。

いかなる場合でも、別の変換ドライバは使用しないでください。変換ドライバは、サーバの文字セット変換と競合します。別の変換ドライバを使用すると、データの破壊につながる可能性が大了。

Interactive SQL と Sybase Central では、iAnywhere JDBC ドライバが文字セット変換を取り扱います。

ANSI 照合についての注意

ISO_1 照合

ISO_1 は、Adaptive Server Enterprise のデフォルトである ISO_1 照合と互換性があります。相違点は次のとおりです。

- β 、小文字の sharp s ($\%xDF$) は、Sybase IQ と SQL Anywhere では小文字の **s** の後にソートされますが、Adaptive Server Enterprise では **ss** の後にソートされます。
- Sybase IQ と SQL Anywhere では、 \mathcal{A} と \mathcal{a} ($\%xC6$ と $\%xE6$) に対応する合字は、それぞれ **A** と **a** の後にソートされます (**A**、**a**、 \mathcal{A} 、 \mathcal{a})。Adaptive Server Enterprise でのソート順は、**A**、**a**、 \mathcal{A} 、 \mathcal{a} となります。

- 1252LATIN1 照合** この照合には、ユーロ通貨記号といくつかの他の文字 (Z-with-caron および z-with-caron) が含まれています。シングルバイトの Windows オペレーティング・システムでは、ほとんどの場合はこの照合が推奨されます。英語または西欧言語を使用している Windows ユーザには、この照合をおすすめします。
- ISO1LATIN1 照合** この照合は ISO_1 と同じですが、A0 ~ BF の範囲にある値のソート方法が異なります。Adaptive Server Enterprise との互換性を保つために、ISO_1 照合では 0xA0 ~ 0xBF の範囲に文字が定義されていません。ただし、この照合のベースとなった ISO Latin 1 文字セットには、この範囲にも文字が割り当てられています。ISO1LATIN1 照合では、この範囲の文字が反映されます。
- Adaptive Server Enterprise と互換性を保つ必要がない場合、通常は、ISO_1 ではなく ISO1LATIN1 を使用することをおすすめします。
- 英語または西欧言語を使用している UNIX ユーザで、デフォルト照合である ISO_BINENG の最適なパフォーマンスが多少損なわれてもよい場合は、ISO1LATIN1 をおすすめします。
- ISO9LATIN1 照合** この照合は ISO1LATIN1 と同じですが、1252 LATIN1 照合に含まれる、ユーロ通貨記号などの新しい文字を含んでいます。
- お使いのコンピュータで ISO Latin 9 文字セットを使用していて、ISO_BINENG の最適なパフォーマンスが多少損なわれてもよい場合は、この照合を使用してください。

マルチバイト照合の使用

この項では、マルチバイト文字セットがどのように処理され、サポートされている照合に適用されるかを説明します。

Sybase IQ には、いくつかのマルチバイト文字セットを使用した照合が用意されています。

Sybase IQ は可変幅文字セットをサポートしています。可変幅の文字セットには、1 バイトで表される文字と、複数バイト (最大 4 バイト) で表される文字が混在しています。文字の最初のバイトは、その文字に使用するバイト数と、その文字の種類 (スペース、数字、またはアルファベット) を示します。

UTF8 照合については、UTF-8 文字は 1 ~ 4 バイトで表されます。他のマルチバイト照合については、1 または 2 バイトが使用されます。用意されているすべてのマルチバイト照合についても、2 バイト以上で構成される文字は「アルファベット」とみなされます。したがって、二重引用符なしで識別子に使用できません。

Sybase IQ は、UTF-16 や UTF-32 など、16 ビットおよび 32 ビットの文字セットをサポートしません。

Embedded SQL 以外のすべてのクライアント・ライブラリは Unicode に対応しており、UTF-16 エンコーディングが使用できます。クライアントとサーバの間で変換が実行されます。

日本語のサポート 日本語の Windows アプリケーションには、照合 932JPN を使用することをおすすめします。照合 932JPN は、SJIS や SJIS2 にはロードできない 32 ビット・マルチバイト文字のロードをサポートしています。SJIS および SJIS2 は古い照合です。SJIS は代替照合として利用できます。SJIS2 は現在ではサポートされていません。Unix アプリケーションについては、EUC_JAPAN を使用してください。

タイ語のサポート Sybase IQ では、CP874toUTF8 ユーティリティを使用して、CP874 形式のデータ・ファイルを UTF8 に変換します。これは、タイ語のために Sybase IQ でサポートされる照合です。構文については、『ユーティリティ・ガイド』を参照してください。このユーティリティを使用して、データを UTF8 に変換せずに CP874 文字セットのデータをロードすることもできます。

SORTKEY() 関数は、ソート順 thaidict (タイ語辞書)、および UTF8 形式のタイ語の文字セットで値を返します。これは次の文と同じ結果です。

```
SELECT c1, SORTKEY(c1) from T1 where rid=3
SELECT c1, SORTKEY(c1, 'thaidict') from T1 where rid=3
SELECT
  '¥340¥270¥201¥340¥271¥207', SORTKEY('¥340¥279¥201¥340¥271¥207'
) from T1 where rid=3
```

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「第 4 章 SQL 関数」の「SORTKEY 関数 [文字列]」を参照してください。

文字セット変換について

Sybase IQ では、同じ文字が文字セットまたはコード・ページ内の異なる位置に定義されている場合に、文字セット間の変換を行うことができます。これを可能にするには、文字セット間にある程度の互換性が必要になります。たとえば、EUC-JIS と cp932 の文字セット間では変換できますが、EUC-JIS と cp1252 の間ではできません。

この項では、Sybase IQ がどのようにして文字セット変換を実行するかを説明します。この説明は、複数の文字セットを使用する環境でアプリケーションやデータベースの配備などを行う上級ユーザ向けです。

データベース・メッセージの文字変換

データベース・ソフトウェアから出力されるエラーなどのメッセージは、「言語リソース・ライブラリ」に保持されています。ローカライズ版 Sybase IQ は、このライブラリのローカライズ版を備えています。

クライアント・アプリケーションのユーザ側では、データベース内のデータや、データベースからのメッセージが表示されます。言語ライブラリ内の文字列である一部のデータベース・メッセージには、データベースからの文字を入れるプレースホルダが含まれています。たとえば、存在しないカラムを指定してクエリを実行すると、次のエラー・メッセージが返されます。

```
Column column-name not found
```

column-name に入る文字列は、データベースから取り込まれます。

データベースが言語ライブラリと異なる文字セットを使用している場合にも、このような情報を常にクライアント・アプリケーションに提供するために、データベース・サーバは、データベース照合で使用されている文字に合わせてメッセージの文字を自動的に変換します。

❖ データベース・メッセージの文字変換の使用

- データベースで使用する照合が、クライアント・マシンで使用されている文字セットや、Sybase IQ の言語リソース・ライブラリで使用されている文字セットと互換性があることを確認してください。ローカライズの異なる Sybase IQ では、言語リソース・ライブラリも異なります。

使用したい文字が各文字セットに入っているか確認する必要があります。

メッセージは、文字セット変換がオンになっているかオフになっているかわからず、常にデータベース照合の文字セットに変換されます。

データベース・サーバの文字セット変換がオン (デフォルト) になっていて、かつクライアントの文字セットがデータベース照合で使用されているものと異なる場合は、文字セット変換がさらに実行されます。

接続文字列と文字セット

接続文字列の文字変換は、特殊なケースです。クライアント・ライブラリは、データベース・サーバを検出または起動するために、接続文字列を解析します。解析はサーバの文字セットや言語を認識せずに実行されます。

インタフェース・ライブラリでは、接続文字列は以下のように解析されます。

- 1 `keyword = value` コンポーネントに分解されます。これは、CommLinks パラメータを大カッコ { } で囲んでいないかぎり、文字セットの種類に関係なく実行されます。大カッコの代わりにカッコ () を使用することをおすすめします。一部のマルチバイト文字セットでは、フォロー・バイト (先頭バイト以外のバイト) として大カッコを使用できます。
- 2 サーバが検出されます。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。Windows の場合は、ANSI 文字セットが使用されます。クライアントとサーバのマシン間で文字セット変換の問題が起きなければ、拡張文字も使用できます。

マシン間での互換性を最大限に高めるためには、サーバ名にはアルファベットの ASCII 文字 1 ~ 127 (または 33 ~ 126) およびアンダースコアのみを使い、記号を使わないようにしてください。サーバ名は 40 文字にトラケートされます。

- 3 DatabaseName (DBN) または DatabaseFile (DBF) パラメータは、データベース・サーバの文字セットで解釈されます。
- 4 データベースが検出されると、他の接続パラメータがその文字セットに従って解釈されます。

文字セット変換の回避

文字セット変換にはパフォーマンス・コストがかかります。文字セット変換の必要がない環境を設定できる場合は、このコストがかからず、管理も簡単です。

シングルバイト文字セット (SBCS) で作業をしていて、7 ビット ASCII 文字 (値 1 ~ 127) だけが使用されている場合、文字セット変換は必要ありません。コード・ページがデータベース内やクライアントのオペレーティング・システムで異なっている場合でも、この範囲では互換性があります。英語のインストール環境の多くは、この条件に当てはまります。このバージョンでは、文字セット変換はデフォルトでオンになっています。これをオフにするには、`-ct` オプションを使用します。

詳細については、『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」を参照してください。

拡張文字を使用する必要がある場合は、次の手順に従います。

- クライアント・マシンのオペレーティング・システムのコード・ページが、データベース内で使用されているものと一致する場合は、データベース内のデータには文字セット変換は必要ありません。
- 自分の国の言語にローカライズされた Sybase IQ が使用可能で、オペレーティング・システムでコード・ページを使用している場合は、データベース・メッセージの文字セット変換は必要ありません。Sybase IQ のメッセージ文字列で使われる文字セットは、以下のとおりです。

言語	【文字セット】
英語	cp1252
フランス語	cp1252
ドイツ語	cp1252
日本語	cp932 (シフト JIS)

また、前述のように、クライアント / サーバの文字セット変換はデフォルトで実行されます。接続文字列で `CharSet=none` を指定した場合、文字セット変換は無効になります。

各国語と文字セットに関する作業

この項では、各国語と文字セットに関する作業をまとめて説明します。

デフォルトの照合の確認

データベース作成時に照合を明示的に指定しなかった場合は、デフォルトの照合が使用されます。Sybase IQ データベースでは、デフォルトの照合は常に ISO_BINENG です。

文字セットの環境設定

この項では、文字セットの問題が正しく処理されるようにコンピューティング環境を設定する方法を説明します。ロケールの環境を適切に設定すると、クライアント/サーバ間の文字セット変換を有効にする必要はなくなります。

❖ 文字セットの環境設定

- 1 その環境で使用する各コンピューティング・プラットフォームについて、デフォルト・ロケールを決定します。デフォルト・ロケールは、各コンピュータの文字セットと言語です。Windows では、文字セットは ANSI コード・ページです。

ロケール情報を確認する方法については、「[ロケール情報の決定](#)」(435 ページ)を参照してください。

- 2 ご自分の環境にそのロケール設定が適しているかどうか確認します。
詳細については、「[照合について](#)」(429 ページ)を参照してください。
- 3 デフォルトの設定が不適切な場合は、データに合った文字セット、言語、データベース照合を決定し、文字セット変換が行われないようにしてください。

詳細については、「[文字セット変換の回避](#)」(433 ページ)を参照してください。

- 4 環境内の各コンピュータのロケールを、決定した値に設定します。
詳細については、「[ロケールの設定](#)」(435 ページ)を参照してください。
- 5 デフォルトの照合を使用するデータベースを作成します。デフォルトの照合が適切でない場合は、名前付きの照合を使ってデータベースを作成します。
詳細については、「[名前付き照合を使用するデータベースの作成](#)」(437 ページ)を参照してください。

データベースで使用する照合を選択するときは、以下のことに注意してください。

- データベース内のデータに適した文字セットとソート順を使用している照合を選択します。多くの場合、この条件に合う照合は OEM 照合や ANSI 照合を含めていくつかあります。
- 文字セット変換を使用すると、パフォーマンス・コストがかかり、システムの設定も複雑になります。文字セット変換の必要がない照合を選択してください。

クライアント・マシンのオペレーティング・システムで使用している文字セットに合った照合順をデータベースで使用すると、文字セット変換が不要になります。クライアント・マシンで Windows を使用している場合は、ANSI 文字セットを選択してください。Sybase IQ データベース・サーバのバージョン 15.2 以降では、文字セット変換はデフォルトで有効になっています。文字セット変換を無効にするには、コマンド・ラインで `CharSet=none` を使用します。

詳細については、「[文字セット変換の回避](#)」(433 ページ) を参照してください。

文字セットの詳細については、『SQL Anywhere サーバ - データベース管理』の「データベースの設定」>「国際言語と文字セット」>「文字セットと照合の参考情報」の「サポートされている文字セット」を参照してください。

ロケール情報の決定

ロケール情報は、システム関数を使用して決定できます。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「[第 4 章 SQL 関数](#)」の「[システム関数](#)」を参照してください。これらの関数を使用してクライアント接続、データベース、およびデータベース・サーバのロケール情報を返す方法については、『SQL Anywhere サーバ - データベース管理』の「データベースの設定」>「国際言語と文字セット」>「国際言語と文字セットのタスク」の「[ロケール情報の確認](#)」を参照してください。

ロケールの設定

オペレーティング・システムのデフォルト・ロケールを使用することも、コンピュータ上の Sybase IQ コンポーネントで使用するロケールを明示的に指定することもできます。

❖ コンピュータ上での Sybase IQ ロケールの設定

- 1 デフォルト・ロケールが必要事項を満たしている場合は、何も設定する必要はありません。

オペレーティング・システムのデフォルト・ロケールを調べるには、「[ロケール情報の決定](#)」(435 ページ) を参照してください。

- 2 ロケールの変更が必要な場合は、SALANG 環境変数か SACHARSET 環境変数、またはその両方を次のように設定します。

```
SACHARSET=charset;SALANG=language_code
```

ここで、*charset* は有効な文字セット・ラベル、*language_code* は『SQL Anywhere サーバー データベース管理』の「データベースの設定」>「国際言語と文字セット」>「ロケールの知識」の「ロケール言語の知識」にある言語ラベル値のリストの言語コードです。

さまざまなオペレーティング・システムで環境変数を設定するには、『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「第 1 章 ファイル・ロケーションとインストール設定」を参照してください。

❖ INSERT...LOCATION 文のロケールの設定

データベースが使用するロケールがプラットフォームのデフォルトと異なる場合は、ローカル・クライアントの環境変数を設定する必要があります。言語、照合順、文字セット、日付/時刻フォーマットの正しい情報を Sybase IQ がロードできるようにするためです。

Sybase IQ は、ロケール名を決定するときに、まず LC_ALL 環境変数の値をチェックします。LC_ALL が設定されていない場合、Sybase IQ は LANG 環境変数の値を使用します。どちらの環境変数も設定されていない場合、Sybase IQ はロケール・ファイルにある「デフォルト」のエントリを使用します。

- 1 `$$SYBASE/locales/locales.dat` ファイルをテキスト・エディタで開きます。例を示します。

```
locale = default, us_english, roman8
locale = C, us_english, roman8
locale = american, us_english, roman8
locale = english.iso88591, us_english, iso_1
```

- 2 LC_ALL または LANG 環境変数を正しい値に設定します。手順 1 のプラットフォームで、データベースの照合が iso_1、使用する言語が英語の場合は、環境変数 LC_ALL または LANG の値を“american.iso88591”に設定する必要があります。設定しない場合、Sybase IQ はロケール名“default”を使用します。その照合は“roman8”です。

たとえば、sh シェルまたは ksh シェルの場合、次のようになります。

```
LC_ALL= american.iso88591;export LC_ALL
```

csh シェルまたは tsch シェルの場合、次のようになります。

```
setenv LC_ALL american.iso88591
```

Sybase IQ は、INSERT...LOCATION 文の実行時にローカライゼーション情報をロードします。

名前付き照合を使用するデータベースの作成

IQ データベースのデフォルト照合は、常に ISO_BINENG です。データベースを作成するときは、それぞれに異なる照合を指定できます。名前付き照合を使用したデータベースの作成については、『SQL Anywhere サーバ - データベース管理』の「データベースの設定」>「国際言語と文字セット」>「国際言語と文字セットのタスク」の「名前を付けた照合を使用してデータベースを作成する」を参照してください。

文字セット変換を使用するデータベース・サーバの起動

文字セット変換は、クライアントとサーバのロケールが異なる場合に実行されます。Sybase IQ では、文字セット変換はデフォルトで有効になっています。データベース・サーバのコマンド・ラインを使用すれば、文字セット変換のオンとオフを明示的に切り替えることができます。

❖ データベース・サーバで文字セット変換を無効にする

- 接続文字列の CharSet=none パラメータを使用して接続します。例を示します。

```
CharSet=none
```

ある照合から別の照合へのデータベースの変更

データベースを他の照合に変更するには、データベースの再構築が必要です。照合はデータベース作成時に選択され、変更できません。データベースの再構築による照合の変更については、『SQL Anywhere サーバ - データベース管理』の「データベースの設定」>「国際言語と文字セット」>「国際言語と文字セットのタスク」の「データベースの照合を変更する」を参照してください。

互換性の問題

バージョン 12.0 より前の Sybase IQ では、常に ASCII のソート順が使用され、大文字は小文字より前に配置されていました。12.4.2 以降のバージョンのデフォルトでは、IQ データベースは、バージョン 12 より前の Sybase IQ と同じ順序でデータをソートします。デフォルトでは、以下の CREATE DATABASE オプションが適用されます。

```
CREATE DATABASE dbname  
COLLATION 'ISO_BINENG'  
BLANK PADDING ON  
CASE RESPECT
```

上記のオプションを指定すると、照合順の中で、大文字はすべての小文字より前になります。たとえば、このオプションでは 'XYZ' は 'abc' より前になります。これは以前のバージョンの Sybase IQ と同じ順序です。

パフォーマンスの問題

文字セットと照合順がバイナリの方が、バイナリでない場合より文字データのパフォーマンスが向上します。

最高のパフォーマンスを発揮するためには、データベースはこのデフォルトのオプション設定で作成してください。

```
CREATE DATABASE dbname
COLLATION 'ISO_BINENG'
CASE RESPECT
```

上記のオプションを指定すると、文字セットと照合順がバイナリになります。この2つのオプションを上記以外の値に設定すると、文字セットも照合順もバイナリになりません。

この設定の短所は、大文字が常に小文字の前に置かれることです。たとえば、“apple” よりも “BANANA” の方が前になります。より自然なソート順にしながら、大文字と小文字を区別するデータベースが必要な場合、パフォーマンスが多少損なわれてもよければ、デフォルトの ISO_BINENG ではなく ISO_1 の照合を使用してください。

注意 大文字と小文字を区別するパスワードの詳細については、『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「[付録 A 他の Sybase データベースとの互換性](#)」の「[ユーザ ID とパスワード](#)」を参照してください。

データのバックアップ、リカバリ、アーカイブ

この章について

この章では、データベースのバックアップおよびリカバリの方法と、読み取り専用のハードウェアを使用して、変更できないデータをアーカイブし、簡単にアクセスできるようにする方法について説明します。また、定期的なバックアップが重要である理由を明らかにし、バックアップのスケジューリングにおいて考慮すべき点を示します。

内容

トピック名	ページ
データの保護	439
データベースのバックアップ	440
仮想バックアップ	458
システム・レベルのバックアップの実行	460
データベースの検証	463
データベースのリストア	464
アテンドなしバックアップ	481
バックアップとリストアに関する情報の取得	482
データ・バックアップ方式とリカバリ方式の決定	485
読み取り専用ハードウェアによるデータのアーカイブ	488

データの保護

Sybase IQ には、2 種類のコンピュータ障害とデータベースの不整合を防ぐための機能が数多く用意されています。

- システム障害は、トランザクションが部分的にしか終了していないのにコンピュータまたはオペレーティング・システムが停止してしまうと発生します。これはコンピュータを不適切にオフしたり再起動したりしたとき、または他のアプリケーションがオペレーティング・システムをクラッシュさせた場合や停電の場合にも起こります。
- メディア障害は、データベース・ファイル、ファイル・システム、またはデータベース・ファイルを保管するデバイスが故障すると発生します。

システム障害が起こると通常は自動的にリカバリされるので、ユーザがデータベースをリストアする必要はありません。システム障害からのリカバリについては、「[第 13 章 システムのリカバリとデータベースの修復](#)」を参照してください。

メディア障害が起こったり、使用するデータベースのデータに何らかの理由で不整合が生じたりすると、ユーザはデータベースをリストアする必要があります。あらゆる状況からデータを保護するために、データベースを定期的にバックアップします。特に、新しく大量のデータをデータベースに挿入したあとは、必ずデータベースをバックアップしてください。

障害が発生すると、Sybase IQ のリカバリ・メカニズムは、トランザクションをアトミックな単位で処理します。未完了のトランザクションはすべてロールバックされ、コミットされたトランザクションはすべて保存されます。このようにして、障害が発生してもデータベースのデータの一貫性が保持されます。

データベースのバックアップ

IQ データベースをバックアップするには、`BACKUP` コマンドを使用します。バックアップには、Sybase IQ データ (IQ ストア) と基本となる SQL Anywhere データベース (カタログ・ストア) の両方が含まれます。

バックアップは、データベースにおける読み取りや書き込みの操作と同時に実行されます。これに対してリストア中は、同じデータベースで他の操作を行うことはできません。

データベースをバックアップするには、そのデータベースに接続する必要があります。`BACKUP` コマンドでは別のデータベースを指定することはできません。

マルチプレックス・データベースのバックアップ方法については、『[Sybase IQ マルチプレックスの使用](#)』の「[第 5 章 マルチプレックス環境でのデータのバックアップとリストア](#)」を参照してください。

データ・ストアの種類

Sybase IQ データ・ストアは 1 つまたは複数のファイルで構成されます。これらのファイルには、ユーザ・データと、起動、リカバリ、バックアップ、トランザクションの管理に使用される内部データベース構造の両方が含まれます。通常、IQ データベースには次のストアがあります。

- `db-name.db` は、データベースや追加した標準 SQL Anywhere データベース・オブジェクトを記述するシステム・テーブルやストアド・プロシージャを含むカタログ DB 領域です。これはカタログ・ストアと呼ばれ、`dbspace-name` は `SYSTEM` です。カタログ・ストアでは、追加の DB 領域を作成できます。

- *db-name.iq* は、IQ テーブル・データとインデックスを含んでいるメイン・データ DB 領域です。これは IQ ストアと呼ばれ、dbspace-name は `IQ_SYSTEM_MAIN` です。dbfile 名は、dbspace-name (`IQ_SYSTEM_MAIN`) と一致します。IQ ストアでは複数の DB 領域を作成でき、各 DB 領域は、`IQ_SYSTEM_MAIN` を含む複数の dbfile を保持できます。
- *db-name.iqtmp* は、特定のクエリで生成されたテンポラリー・テーブルを含んでいるイニシャル・テンポラリー DB 領域です。これは IQ テンポラリー・ストアと呼ばれ、dbspace-name は `IQ_SYSTEM_TEMP` です。dbfile は、IQ テンポラリー・ストアに追加できます。

ログ・ファイルを含め、これらのどのストアも障害発生領域となり得ます。

バックアップの種類

Sybase IQ データをバックアップするには次の 4 種類の方法があります。

- データベースのバックアップ
- オペレーティング・システム・レベルのバックアップ
- 仮想バックアップ
- アーカイブのバックアップ (ログ・ファイル用)

Sybase IQ には、4 種類のデータベース・バックアップ方法があります。

- フル・バックアップ。データベースの完全なコピーを作成します。
- 仮想バックアップ。IQ ストアのテーブル・データ以外のすべてのデータベースをコピーします。
- インクリメンタル・バックアップ。最後にいずれかのバックアップが実行されてからのトランザクションをすべてコピーします。
- フル・バックアップ以降のインクリメンタル・バックアップ。前回のフル・バックアップが実行されてからのトランザクションをすべてコピーします。

どの種類のバックアップでも、カタログ・ストアは完全にバックアップされます。カタログ・ストアのサイズは、IQ ストアよりもかなり小さい場合がほとんどです。ただし、カタログ・ストアが IQ ストアよりも大きい (またはほぼ同じサイズである) 場合、IQ のインクリメンタル・バックアップを実行すると、サイズが予想以上に大きくなります。

インクリメンタル仮想バックアップは、`BACKUP` 文によってサポートされます。

テンポラリー・ストアのデータはバックアップされません。ただし、テンポラリー・ストア構造の再作成に必要なメタデータとその他の情報はバックアップされます。

❖ IQ ストアとカタログ・ストアのバックアップ

次にバックアップ手順の概要を示します。詳細についてはこの章の後半で説明します。バックアップは詳細を理解したうえで実行してください。

- 1 DBA 権限付きのアカウントを使用してサーバに接続します。マルチプレックス・データベースの場合は、コーディネータに接続する必要があります。
- 2 BACKUP コマンドを実行します。完全な構文については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「BACKUP 文」を参照してください。

次のファイルのバックアップを作成します。

- カタログ・ストア (SYSTEM DB 領域ファイル。通常は *dbname.db* という名前)
 - IQ ストアのすべての DB 領域ファイル
- 3 各サーバの *params.cfg* ファイルのコピーを作成します。このファイルは、BACKUP ではバックアップできません。
 - 4 次のファイルのサイズを保存します。
 - コーディネータ上のすべての DB 領域ファイル
 - IQ テンポラリ・ストア

バックアップ対象データ

BACKUP コマンドでは、コミットされたデータだけがバックアップされます。バックアップは、コミットおよび自動チェックポイントで開始します。このチェックポイントでは、バックアップ・プログラムによってバックアップ対象データが決定されます。この時点で、データベースの現在のスナップショットのバージョンがバックアップされます。チェックポイントが発生した時点でコミットされていないデータは、バックアップされません。

バックアップの最後に 2 番目の自動チェックポイントが発生します。バックアップの実行中にコミットされたすべてのデータが、引き続きバックアップされます。

Sybase IQ は、バックアップ時に実際に使用されているリカバリ可能データベース・ブロックだけをバックアップします。空きブロックはバックアップしません。

また Sybase IQ は、データベース・ファイルと、接続している IQ データベースに関連したカタログ情報をバックアップします。このとき、トランザクション・ログ・ファイルはバックアップしません。データベースのリストアにトランザクション・ログを使用しないためです。

トランザクションのすべてのコマンドが何らかの理由で正しく機能しなかった場合、またはデータベースに欠落しているファイルがある場合、バックアップは失敗します。

バックアップ、リストア、リカバリ中のトランザクション・ログ

Sybase IQ は、システム障害からのリカバリでトランザクション・ログ・ファイルを使用します。IQ データベースのリストア、コミットされた IQ トランザクションのリカバリ、または Sybase IQ データベースのカatalog・ストアのリストアには、トランザクション・ログは使用されません。トランザクション・ログが存在すると、フル・リストアを実行できません。フル・リストアを実行するには、このファイルを削除する必要があります。

注意 SQL Anywhere データベースでは、トランザクション・ログとその他のログを異なる方法で使用します。SQL Anywhere データベースをリカバリする場合は、トランザクション・ログ・ファイルが必要であり、BACKUP コマンドがこのファイルを保持します。詳細については、『SQL Anywhere サーバ - SQL の使用法』を参照してください。また、カatalog・ストアにシステム・テーブル以外のデータがある場合、リカバリできるのは、そのデータのトランザクションが障害発生前にディスクに書き込まれたときのみです。

トランザクション・ログのライブ・バックアップ

dbbackup ユーティリティの `-l` オプションを使用すると、トランザクション・ログのライブ・バックアップを作成できます。

❖ トランザクション・ログのライブ・バックアップの作成

- 1 オンラインのマシンで障害が発生した場合にデータベースを起動するためのセカンダリ・マシンをセットアップします (セカンダリ・マシンで。
- 2 セカンダリ・マシンのフル・バックアップを定期的に作成します。
- 3 次のコマンドを実行して、セカンダリ・マシンに対しトランザクション・ログのライブ・バックアップを実行します。

```
dbbackup -l path¥filename.log -c "connection_string"
```

通常は、dbbackup ユーティリティはセカンダリ・マシンから実行してください。プライマリ・マシンが使用できなくなったときに、セカンダリ・マシンを使用してデータベースを再起動できます。再起動に必要な情報は、データベース・ファイルとトランザクション・ログに保持されています。

バックアップ・データの分散

BACKUP コマンドを実行すると、1 番目のアーカイブ・デバイスでカタログ・ストアのフル・バックアップが常に実行されます。次に、指定したすべてのデバイスで並行して、IQ ストアのデータがバックアップされます。ブロックは、アーカイブ・メディア全体にわたって均等に分散されるわけではありません。各スレッドの処理速度に応じて、特定デバイスのブロックを増やすことができます。

注意 一連のファイルはバックアップされた順にリストアする必要があるため、バックアップ・データの分散は重要です。詳細については、「[正しい順序のリストア](#)」(473 ページ) を参照してください。

データベースの一貫性の確保

BACKUP コマンドを実行しても、データベースをバックアップする前に必要なファイルがすべて存在するかどうかはチェックされますが、内部の一貫性はチェックされません。詳細なチェックが必要な場合は、バックアップ前にストア・プロシージャ `sp_iqcheckdb` を実行します。詳細については、「[データベースの検証](#)」(463 ページ) を参照してください。

アーカイブ・デバイスの選択

任意の IQ データベースを WORM デバイスなどの磁気テープまたはディスクにバックアップできます (WORM デバイスの詳細については、「[読み取り専用ハードウェアによるデータのアーカイブ](#)」(488 ページ) を参照してください)。Sybase IQ では、複数のテープ・ドライブを使用してデバイスに近い速度でバックアップとリストアを実行できます。また、ディスク・ストライピングがオンの場合は、複数のディスクに対しても実行できます。バックアップ・デバイスの名前は、BACKUP コマンドの `archive_device` パラメータに指定します。

ディスク・バックアップの要件

ディスク・バックアップでは、ファイル・システムにアクセスする必要があります。この場合、ロー・ディスクはバックアップ・メディアとして使用できません。RAID (redundant array of independent devices) デバイスの全ディスクが 1 つのデバイスとして扱われます。

テープ・バックアップの要件

定期的な大規模なデータベースのバックアップを行う場合は、お使いのプラットフォームで DLT デバイスをサポートしていれば、これを使用する必要があります。いずれの場合でも、複数のテープ・ドライブを使用することをおすすめします。

Sybase IQ の BACKUP では、次のテープ・ドライブをサポートしています。

- DLT (Digital Linear Tape)。UNIX システムで使用します。
- 4 mm Digital Data Storage (DDS)
- 8 mm

Sybase IQ では、複数のテープとともにスタッカ・ドライブを使用することもできます。

Sybase IQ の BACKUP は、ジュークボックスやロボット・ローダをサポートしていません。これらが必要な場合は、サード・パーティ製メディア・マネージャを使用してください。

Sybase IQ の BACKUP は、UNIX システムで使用している QIC (Quarter Inch Cartridge) デバイスなど固定長のテープ・デバイスをサポートしていません。

各プラットフォーム固有のバックアップ要件を以下に示します。

- AIX システムでは、テープ・デバイスのブロック・モードを固定長、可変長のどちらにも設定できます。ブロック・モードの表示方法および変更方法の詳細については、『インストールおよび設定ガイド』を参照してください。Sybase IQ の BACKUP では、固定長のブロック・モードはサポートされていません。
- IBM Linux on POWER では、IQ データベースを SCSI テープにバックアップするためには、可変長データ転送に対応できるようにデバイスのブロック・サイズを設定する必要があります。IQ バックアップを実行する前に、SCSI テープ・デバイスのデフォルトのブロック・サイズを設定します。スーパーユーザとしてログインし、Linux シェル・コマンド `mt` を次のように実行します。

```
mt -f /dev/st0 defblksize 0
```

バックアップ・デバイス数の制限

Sybase IQ の BACKUP コマンドを使用するときに、複数の TO 句を指定すると、複数のデバイスの操作を並行処理できます。BACKUP 文では、デバイス数の上限は 36 になっています。この制限を超えると、エラーが発生します。

この制限は、すべてのバージョンの Sybase Risk Analytics Platform と Sybase RAP - The Trading Edition™ に適用されます。

将来のバックアップに対する修正アクション

BACKUP コマンドで 36 以下で TO 句を指定し、将来のバックアップを作成する必要があります。このとき作成されたバックアップの数が Sybase IQ の今後のバージョンに適用される制限数となります。

バックアップ・コマンドは少なくします。I/O およびハードウェアとデバイスの競合が増加するため、デバイス数を極端に増加しないでください。現実的には、マシンのコアごとにほぼ 1 つのデバイスを使用すると、CPU 使用率が飽和状態になります。より高速なシステムではコアごとに最大 2 つのデバイスを使用します。

バックアップの準備

BACKUP を実行するには、次の項に示す条件を満たす必要があります。

DBA 権限の取得

BACKUP または RESTORE を実行するには、データベースに対する DBA 権限が必要です。DBA ユーザとしてログオンするか、「[権限の付与](#)」(350 ページ)の説明に従って DBA から DBA 権限を付与してもらう必要があります。

テープのリワインディング

Sybase IQ は、テープを使用する前にリワインドしません。必ずバックアップやリストアの操作で使用したテープの開始ポイントが正しいことを確認してから、テープをデバイスに挿入してください。

リワインディング・デバイスを使用している場合は、バックアップ後にテープがリワインドされます。自動的にテープをリワインドするテープ・デバイスを使用する場合は、情報を上書きしないように注意してください。

古いディスク・バックアップの保持

BACKUP を実行すると、同じ名前の既存のディスク・ファイルが上書きされます。既存のバックアップを保持する必要がある場合は、アーカイブ・デバイスに異なるファイル名またはパス名を使用して新しいバックアップを作成します。または、既存のバックアップを別のロケーションに移動させてからバックアップを開始します。

2 種類の BACKUP 実行方法

BACKUP の実行方法には、次の 2 種類があります。

- アテンド付き。アテンド付きのモードではオペレータがその場にいることを想定しているため、必要に応じてアーカイブ・メディアをマウントするように要求されます。この方法を使用する場合は、コマンド・ラインで BACKUP を対話的に実行する必要があります。

- アテンダなし。アテンダなしのモードではオペレータがその場にはいないことを想定しているため、メッセージは表示されません。代わりに、必要な領域を正しく見積もり、それに応じて使用するデバイスを設定する必要があります。この場合エラーは、すべて致命的なエラーとみなされます。

サード・パーティ製ソフトウェアを使用して、バックアップを作成できる場合もあります。このような製品は、アテンダなしバックアップを行うとき特に役立ちます。オペレータ不在時のバックアップに関する詳細については、「[アテンダなしバックアップ](#)」(481 ページ)を参照してください。

注意 BACKUP は、Interactive SQL だけでなく、バッチ・スクリプトやプロシージャからも実行できます。イベント・ハンドラを使用してバックアップを自動化させることもできます。詳細については、『システム管理ガイド第 2 巻』の「[第 6 章 スケジューリングとイベント処理によるタスクの自動化](#)」を参照してください。

メディア容量の見積もり

バックアップを行う前に、使用するアーカイブ・メディアに十分な空き領域があることを確認します。ディスクやテープの使用可能領域を見積もる場合は、次の規則に注意してください。

- IQ ストアのフル・バックアップまたはインクリメンタル・バックアップと同様に、カタログ・ストアのフル・バックアップには十分な空き領域が必要です。Sybase IQ システム・テーブルの他に SQL Anywhere のデータもカタログ・ストアに含まれている場合、このデータのバックアップ領域も必要となります。
- トランザクション・ログはバックアップされないため、このための領域は必要ありません。
- テープ・バックアップの場合、指定する最初のテープ・セットは、カタログ・ストアのフル・バックアップ(カタログ・ストアの IQ 以外のデータを含む)を保持できる必要があります(テープ・セットは、所定のアーカイブ・デバイスで作成される 1 つまたは複数のバックアップ・テープで構成されます)。
- 複数のテープ・ドライブを搭載したスタッカ・デバイスでは、指定したデバイスで使用するテープのサイズがすべて同一でなければなりません。

バックアップのたびに、常に新しいテープを使用することをおすすめします。

バックアップを開始する前に、Sybase IQ はまずバックアップ用の十分なディスク・ファイル領域があるかどうかテストします。オペレータがいるアテンダ付きバックアップの場合は、領域が不足していると、データを書き込む前にディスクからファイルをいくつか移動するよう指示されます。ディスク領域が確保できるまで、バックアップは開始されません。

同様に、アテンド付きディスク・バックアップの最中に領域がなくなると、**BACKUP** は開いているバックアップ・ファイルをすべて閉じ、領域の一部がクリアされるまで待機します。その後、新しいバックアップ・ファイルから再び処理を開始します。必要に応じてバックアップを停止することもできます。

デフォルトでは、8KB 以上のディスク空き領域を確保してから、バックアップを再開しなければなりません。

アテンドなしバックアップでは、領域の解放を指示するメッセージが表示されません。したがって、アテンドなしバックアップは、領域が足りなくなると、失敗します。アテンドなしバックアップでは、**BACKUP** によるサイズの見積もり方法が異なります。詳細については、「[アテンドなしバックアップ](#)」(481 ページ) を参照してください。

テープを使用してオペレータがいるアテンド付きバックアップを行う場合は、すぐにバックアップが開始されます。領域がなくなったら、追加テープをマウントしてください。

バックアップの実行

構文については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」の「[BACKUP 文](#)」を参照してください。

同時実行性とバックアップ

次の点を除き、バックアップは他のほとんどのデータベース操作と同時に実行できます。

- カタログ・ストアのバックアップ中、メタデータは変更できません。
- チェックポイントや DBCC コマンドは実行できません。

ただし、バックアップの開始時にコミットされていないトランザクションはバックアップされません。バックアップ中にシステム障害またはメディア障害が発生しても、コミットされていないトランザクションはリストアできません。

バックアップを開始すると、**CHECKPOINT** コマンドは実行できません。

オペレータの存在についての指定

ATTENDED ON または **OFF** は、新しいテープやディスク・ファイルが必要なときにユーザが介入するかどうかを制御します。デフォルトは **ON** です。

ディスクを使用してアテンドなしバックアップを行う場合は、ディスク領域の追加を指示されることはありません。領域がなくなると、エラーが発生して **BACKUP** が停止します。

テープを使用してアテンションなしバックアップを行う場合は、新しいロード用テープの挿入を指示されることはありません。SIZE オプションと STACKER オプションで、領域がなくなった場合の対処方法を指定できます。これらのオプションの詳細については、「[アーカイブ・デバイスの指定](#) (450 ページ) を参照してください。

バックアップの種類の設定

FULL | INCREMENTAL | INCREMENTAL SINCE FULL では、バックアップの種類を指定します。次のいずれかを選択します。

- FULL は、カタログ・ストアと IQ ストアのフル・バックアップを行います。FULL がデフォルトです。

仮想バックアップの場合、BACKUP 文の VIRTUAL DECOUPLED | VIRTUAL ENCAPSULATED オプションを使用できます。詳細については、「[仮想バックアップ](#)」(458 ページ) を参照してください。

- INCREMENTAL は、カタログ・ストアのフル・バックアップを実行したあと、前回バックアップが実行されてからの IQ ストアの変更内容をすべてバックアップします。
- INCREMENTAL SINCE FULL は、カタログ・ストアのフル・バックアップを実行したあと、前回フル IQ バックアップが実行されてからの IQ ストアの変更内容をすべてバックアップします。

INCREMENTAL および INCREMENTAL SINCE FULL の仮想バックアップは、BACKUP 文の VIRTUAL DECOUPLED オプションおよび VIRTUAL ENCAPSULATED オプションによりサポートされます。

データベースの一連の読み取り／書き込みファイルのフル・バックアップ、フル・バックアップ以降のインクリメンタル・バックアップ、またはインクリメンタル・バックアップを制限するには、READWRITE FILES ONLY キーワードを使用します。バックアップされた読み取り／書き込み DB 領域またはファイルは、IQ メイン・ストアに属します。バックアップされたファイルは、バックアップ・コマンドによってカタログの読み取り／書き込みステータスがチェックされると、選択されます。READWRITE FILES ONLY および READONLY FILES の使用の制限については、『リファレンス：文とオプション』の「SQL 文」の「BACKUP 文」を参照してください。

IQ バックアップを実行すると、一連の読み取り専用 DB 領域または読み取り専用ファイルもバックアップされます。読み取り専用の DB 領域またはファイルは、IQ メイン・ストアに属します。バックアップされたファイルは、ユーザが選択できます。

バックアップの種類を選択方法については、「[日常のバックアップの計画](#)」(485 ページ) を参照してください。

仮想バックアップの指定

VIRTUAL DECOUPLED | VIRTUAL ENCAPSULATED '*shell-command*' オプションは、仮想バックアップの種類を指定します。VIRTUAL ENCAPSULATED パラメータの *shell-command* 変数は、バックアップ操作の一部として、シェル・コマンドでシステム・レベル・バックアップを実行することを許可します。仮想バックアップの詳細については、「[仮想バックアップ](#)」(458 ページ) を参照してください。

アーカイブ・デバイスの指定

TO *archive_device* 句では、バックアップ先のディスク・ファイルまたはシステム・テープ・ドライブを指定し、アーカイブ・デバイスの数を制御します。詳細については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」を参照してください。

バックアップ・ファイル名 (ディスクを使用する場合)

BACKUP を実行すると常に、指定した *archive_device* の名前にサフィックスが追加され、ディスク・バックアップ・ファイルにファイル名が割り当てられます。サフィックスは “.” とそれに続く数字で構成されます。この数字は新しいファイルが増えるたびに 1 ずつ増加します。たとえば、*archive_device* として */iqback/mondayinc* を指定した場合、バックアップ・ファイル名は */iqback/mondayinc.1*、*/iqback/mondayinc.2* といった名前になります。この規則により、そのファイル・サイズが上限を超えないように制御しながら、必要なサイズのバックアップを格納できます。詳細については、[SIZE](#) オプションを参照してください。この規則に対応するため、使用するファイル・システムが長いファイル名をサポートしている必要があります。

また、*archive_device* に指定するディレクトリ名が存在していることを確認する必要があります。BACKUP では、存在していないディレクトリは作成されません。存在していないディレクトリでバックアップを開始しようとすると、バックアップに失敗します。

ディスク・ファイルのロケーションを指定するときに、相対パス名を使用しないでください。BACKUP は、サーバが起動されたロケーションを起点としてこのパス名を解釈するため、バックアップ時にロケーションを正確に特定できないことがあります。また、パスで指定されたディレクトリ以外にデータが存在すると、バックアップ用の領域が足りなくなる場合があります。

テープ・デバイスの開始位置の調整

BACKUP を実行しても、テープの開始位置は調整されません。バックアップを開始する前にテープの開始位置を正しく調整し、リワインドされたテープ・デバイスを使用する場合は、すでにあるバックアップを上書きしないように注意してください。このため、リワインドされていないテープ・デバイスの使用をおすすめします。正しい命名規則については、オペレーティング・システムの各プラットフォーム用のマニュアルを参照してください。

テープ・デバイスの指定 (UNIX の場合)

次に、UNIX プラットフォームでリワインドされていないテープ・デバイスを指定する方法の例を示します。

- Sun Solaris プラットフォームの場合は、'/dev/rmt/0n' のように、デバイス名の後ろに「リワインドしていない」ことを示す文字 n を挿入する。
- IBM AIX プラットフォームの場合は、'/dev/rmt0.1' のように、適切な圧縮設定とリワインド設定を指定する小数点とそれに続く数字を挿入する。
- HP-UX プラットフォームの場合は、'/dev/rmt/0mn' のように、デフォルトのテープ・メカニズムを指定する '0m' と「リワインドしていない」ことを示す 'n' を指定する。

警告！ テープ・デバイス名のスペルを間違えて、システムで使用できないテープ・デバイス名を書き込むと、BACKUP はそのテープ・デバイスをディスク・ファイルとみなします。

テープ・デバイスの指定
(Windows の場合)

Windows システムではデバイスのリワインディング状態を指定できません。サポートされているのは、テープ・デバイスに対する固定長の I/O 操作だけです。一方、Sybase IQ がサポートしているのは可変長デバイスです。Windows システムで固定長テープ I/O に対応するには、追加処理を行う必要があります。

Windows はテープ・パーティションをサポートしていますが、Sybase IQ はテープ・パーティションを使用しません。このため、Sybase IQ のバックアップやリストアに使用するテープをフォーマットする場合、別のアプリケーションを使用しないでください。Windows では、1 番目のテープ・デバイスは '~~¥¥¥¥~~tape0'、2 番目は '~~¥¥¥¥~~tape1' という具合に命名されます。

警告！ バックアップ (およびその他のほとんどの状況) では、文字列中出现する ¥n、¥x または ¥¥ のうちの最初の ¥ は、Sybase IQ によりエスケープ文字として処理されます。このためバックアップ用のテープ・デバイスを指定する場合は、Windows の命名規則に従って必要な場所に ¥ を 2 つずつ使用しなければなりません。たとえば、バックアップに使用する 1 番目の Windows テープ・デバイスを '~~¥¥¥¥~~tape0'、2 番目を '~~¥¥¥¥~~tape1' とします。追加すべき ¥ を省略したり、テープ・デバイス名の入力を間違えたり、システム上にないテープ・デバイス名を入力したりすると、Sybase IQ はそれをディスク・ファイル名と解釈します。

テープ・バックアップの
サイズ指定

Windows の固定長 I/O の詳細については、『パフォーマンス&チューニング・ガイド』の「バックアップ操作のチューニング」を参照してください。

TO 句の SIZE オプションは、ストライプ上のバックアップ・データの最大サイズ (KB) を示します。

サード・パーティ製バックアップ製品ではなく Sybase のバックアップ機能を使用する場合、テープの終了マーカを正確に検出できないプラットフォームでアテンドなしテープ・バックアップを実行するときは、**SIZE** を指定してください。なお、**SIZE** の値は、出力デバイスごとに指定します。対応するデバイスで使用されるボリュームを、この値より小さくすることはできません。アテンド付きテープ・バックアップの場合は **SIZE** を指定する必要はありませんが、サイズを正確に指定するにはこれを指定するのがもっとも良い方法です。

SIZE を指定していない場合、バックアップ中にテープの領域がなくなるとエラーが発生します。指定されたサイズになる前にテープの領域がなくなると、すぐにエラーが発生するのではなく次の処理が行われます。

- アテンド付きバックアップで **SIZE** と **STACKER** を指定している場合は、次のテープを開こうとします。
- アテンド付きバックアップで **SIZE** は指定しているが、**STACKER** を指定していない場合は、新しいテープを挿入するよう指示されます。
- アテンドなしバックアップで **SIZE** と **STACKER** を指定している場合は、次のテープを開こうとします。使用可能なボリュームがない場合や、**STACKER** を指定しない場合は、エラーが発生します。

追加されたテープはどれもリストアに必要なヘッダ情報が含まれていないので、リストアするときはテープのマウント順に注意してください。順番を間違えるとデータベースの一貫性が失われる可能性があります。

Windows では、テープ・デバイスで **SIZE** オプションを使用する場合、次の条件を満たす必要があります。

- **SIZE** の値は 64 の倍数でなければならない。それ以外の値は、64 の倍数に切り捨てられる。
- **SIZE** を明示的に指定しないと、自動で 1.5GB に設定される。

ディスク・バックアップのサイズ指定

TO 句の **SIZE** オプションは、ストライプ上のバックアップ・データの最大サイズ (KB) を示します。なお、**SIZE** の値は、出力デバイスごとに指定します。

Sybase のバックアップ機能を使用する場合、*archive_device* と名前を付けたディスク・ファイルがデフォルトの 2GB (UNIX) または 1.5GB (Windows) より大きければ、アテンド付きでもアテンドなしでも **SIZE** を指定してください。

バックアップ中、指定した *archive_device* に書き込まれる情報の量が **SIZE** に達すると、現在のファイルが閉じられ、同じ名前 (ただし、次の昇順番号が付加された名前) のファイルが別に作成されます。

たとえば、*archive_device* の 1 つに *janfull* というディスク・ファイルを指定し、最大ファイル・サイズが 200MB になるよう **SIZE 200000** を指定したとします。このときバックアップに 2GB が必要になると、*janfull.1*、*janfull.2* ~ *janfull.10* という具合に、200MB のファイルが 10 個作成されます。バックアップを実行するにあたっては、それだけの量のデータを書き込むだけの余裕がディスクにあるかどうかを確認する必要があります。

スタッカ・デバイスの指定

TO 句の STACKER オプションは、自動的にロードされたマルチテープ・スタッカ・デバイスにバックアップしていることを示し、デバイスのテープの数を指定します。ATTENDED が ON であり、STACKER が指定されている場合、BACKUP は次のテープがロードされるまで無期限に待機します。指定したスタッカ・デバイスで使用するテープのサイズは、すべて同一でなければなりません。

サード・パーティ製バックアップでのデバイス指定

注意 サード・パーティ製バックアップ製品を使用する場合は、SIZE や STACKER を指定しないでください。サイズ情報は *vendor specific information* 文字列に保持されます。この文字列を指定する場合、「[Sybase 以外の製品によるバックアップの実行](#)」(457 ページ) を参照してください。

その他のバックアップ・オプション

ブロック係数の指定

BLOCK FACTOR では、同時にアーカイブ・デバイスに書き込む IQ ブロックの数を指定します。値は 0 より大きくしてください。それ以外の場合、BACKUP からエラー・メッセージが生成されます。UNIX プラットフォームでは、BLOCK FACTOR のデフォルト値は 25 です。Windows では、BLOCK FACTOR のデフォルト値はデータベースのブロック・サイズによって異なります。たとえば、ブロック・サイズが 512 バイトの場合は、BLOCK FACTOR が 120 ブロックになります。また、ブロック・サイズが 32KB の場合は、BLOCK FACTOR が 1 ブロックになります。

このパラメータは、バックアップ中にバッファとして使用されるメモリ量も制御するため、バックアップのパフォーマンスに直接影響を及ぼします。ブロック係数の効果は、ディスク・サブシステム、テープおよびプロセッサの速度によって異なります。小さなブロック係数でバックアップのパフォーマンスが向上するシステムもあれば、大きなブロック係数でパフォーマンスが向上するシステムもあります。プラットフォームの最適な I/O サイズとブロック係数については、プラットフォームで使用するオペレーティング・システムのマニュアルを参照してください。

エラー・チェック

CRC ON または OFF は、32 ビット周期の冗長性検査をアクティブにしたり、非アクティブにしたりします (BACKUP では、ハードウェアで使用可能なエラー検出もすべて使用されます)。CRC ON の場合、以降の RESTORE コマンドの処理中に、バックアップで計算されたチェックサムが検証されます。デフォルトは CRC ON です。

コメントの追加

WITH COMMENT では、バックアップ・アーカイブのヘッダ情報の一部として使用される文字列 (最長 32KB) を指定します。このオプションを省略すると NULL が入力されます。RESTORE DATABASE FROM CATALOG ONLY を実行するか、Sybase IQ のバックアップ・ログ *backup.syb* を表示すると、コメント文字列を参照できます。

SQL Anywhere 専用のデータベースをバックアップする場合、その他の BACKUP オプションについては、『SQL Anywhere サーバー SQL リファレンス』を参照してください。

テープ・デバイスの待機

バックアップやリストアの操作中にアーカイブ・デバイスをオープンできない場合 (メディアのロードが必要な場合など)、Sybase IQ は 10 秒間待機してから再試行します。デバイスが正常に挿入されるか、[Ctrl + C] キーを押して操作を終了しないかぎり、サーバは無制限に再試行を行います。メッセージはサーバの `.stderr` ファイルに書き込まれます。アーカイブ・デバイスを開けないことを示すコンソール通知はありません。

読み取り専用のハードウェアを使用したバックアップとリストア

Sybase IQ では、バックアップとリストアの両方の操作で、読み取り専用ハードウェアがサポートされています。次のような規則に従います。

- Sybase IQ は、リストア時に読み取り専用デバイスへの書き込みを行いません。デバイスがハードウェア・レベルで読み取り専用モードにフリーズされている可能性があるからです。
- 仮想バックアップでは、read-only の dbspace のヘッダ・ブロックやその他のブロックは、バックアップやリストアの対象外です。read-only の dbspace は変更されないことが確定しているため、仮想バックアップとリストアで read-only の dbspace のリストアが必要なのは、read-only の dbspace でメディア障害が発生したときのみです。
- 非仮想フル・バックアップでは、モードに関係なくすべての dbspace がバックアップされます。
- 非仮想インクリメンタル・バックアップでは、次のような読み取り専用の DB 領域はバックアップされません。
 - インクリメンタル・バックアップの基となる前回のバックアップの時点で読み取り専用だったもの
および
 - それ以降変更されていないもの

このような DB 領域の内容は、前回の依存バックアップに完全に含まれています。読み取り専用 DB 領域で、依存バックアップ以降に変更されたものはバックアップされます。

詳細については、「[読み取り専用ハードウェアによるデータのアーカイブ](#)」(488 ページ) を参照してください。

バックアップの例

例 1 – フル・バックアップ

UNIX システムで、データベース `iquser` を 2 つのテープ・デバイスにバックアップする例を示します。これはアテンド付きのフル・バックアップです。このバックアップを実行する前に、テープのバックアップ・ファイルの書き込み開始位置を調整し、`iquser` へ接続しておく必要があります。そして次のコマンドを発行します。

```
BACKUP DATABASE
TO '/dev/rmt/0n'
TO '/dev/rmt/1n'
WITH COMMENT 'Jan 18 full backup of iquser'
```

まず、カタログ・ストアが `/dev/rmt/0n` にバックアップされます。次に、IQ ストアが両方のテープにバックアップされます。

例 2 – インクリメンタル・バックアップ

例 1 と同じデータベースのインクリメンタル・バックアップの例を示します。ここではテープ・デバイスを 1 つだけ使用し、次のコマンドを発行します。

```
BACKUP DATABASE
INCREMENTAL
TO '/dev/rmt/0n' SIZE 150
WITH COMMENT 'Jan 30 incremental backup of iquser'
```

その他の例

これら 2 種類のバックアップからデータベースをリストアする方法の例については、この章の後半で説明します。

読み取り専用ファイルおよび DB 領域を指定するバックアップの例については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「BACKUP 文」を参照してください。

バックアップ中のエラーからのリカバリ

バックアップに失敗する原因のほとんどは、領域不足またはハードウェア障害のいずれかです。また、サード・パーティ製ソフトウェアを使用して障害が発生する場合があります。

バックアップ領域の確認

BACKUP では、STACKER オプションと SIZE パラメータを使用してバックアップ用の領域が十分であるかどうかを判断します。

- ディスク・バックアップの場合、領域が不足していると判断されると、実際にデータを書き込む前にバックアップに失敗します。

- バックアップを開始するのに十分な領域があると判断されても、バックアップの完了前に領域が足りなくなった場合(見積もりが間違っている場合や、バックアップ中にユーザが別アプリケーションで大量のディスク領域を使用している場合など)、アテンド付きバックアップでは新しいテープのロードまたはディスク領域の解放が要求されます。アテンドなしバックアップの場合は、領域がなくなるとバックアップに失敗します。
- STACKER も SIZE も指定しないと、完了するかテープやディスクが一杯になるまでバックアップ処理は続行します。領域がなくなると、アテンド付きバックアップでは新しいテープのロードまたはディスク領域の解放が要求され、アテンドなしのバックアップではバックアップに失敗します。

リカバリの試行

バックアップに失敗すると、バックアップ・プログラムは次のようにリカバリを試みます。

- バックアップ開始時または終了時のチェックポイントでバックアップに失敗すると、通常のチェックポイント・リカバリを実行する。
- チェックポイント間でバックアップに失敗すると、バックアップをロール・バックする。
- 開始チェックポイントから終了チェックポイントまでの間にシステム障害が発生し、データベースをリストアしなければならない場合は、以前のバックアップ・テープまたはディスク・ファイルのセットを使用してリストアする必要がある。
- FULL バックアップ後の終了チェックポイントでシステム障害が発生した場合は、作成したばかりのバックアップ・テープまたはファイルからリストアできる。

バックアップ完了後の作業

データベースまたはその dbspace の 1 つを移動させなければならない場合、バックアップで作られたデータベース内の各 dbspace の名前が必要になります。バックアップ完了後にこの情報を記録する方法の詳細については、「[DB 領域名の記録](#)」(484 ページ)を参照してください。

Sybase IQ には、RESTORE SQL 文の VERIFY 句を使用して、Sybase IQ バージョン 12.6 以降の既存のデータベースのバックアップを検証するメカニズムがあります。詳細については、「[データベースのバックアップの検証](#)」(478 ページ)を参照してください。

Sybase 以外の製品によるバックアップの実行

Sybase IQ では、さまざまなサード・パーティ製品を使用してバックアップやリストアを実行できます。使用するパッケージは、Adaptive Server Enterprise バックアップ・インタフェースに準拠していなければなりません。また、Sybase データベースをサポートしているかどうか、使用する製品のマニュアルを確認してください。

サード・パーティ製品を使用してバックアップやリストアを実行するには、Sybase IQ を使用して実行するのと同じように **BACKUP** 文または **RESTORE** 文を発行します。ただし、次の場合は除きます。

- `archive_device` ごとに、実際のデバイス名ではなく次のフォーマットで文字列を指定する場合。

`dll_name::vendor_specific_information`

- **STACKER** パラメータまたは **SIZE** パラメータを指定しない場合。

`dll_name` は、実行時にロードされるダイナミック・リンク・ライブラリに対応しています。これには、1 から 30 バイトまでの長さで、英数字およびアンダースコア文字だけを使用できます。`dll_name` は、各 `archive_device` と一致しなければなりません。

`vendor_specific_information` の内容は製品によって異なります。また、`archive_device` ごとに異なる場合もあります。`dll_name::` とベンダ情報を含む文字列全体の長さは、255 バイト以内です。

バックアップ・プログラムは、サード・パーティ製プログラムにベンダ情報を自動的に渡します。サード・パーティ製プログラムによるバックアップを要求すると、この情報がバックアップ・ヘッダ・ファイルに配置され、実際に指定した各 `archive_device` 用に作成された最初のテープまたはディスク・ファイルで、このヘッダ・ファイルが書き込まれます。

注意 この構文を Sybase IQ で使用して動作確認されているサード・パーティ製品は、ごく一部に限られます。使用上のその他の指示や制約については、『リリース・ノート』を参照してください。サード・パーティ製品をこの方法で使用して IQ データベースをバックアップする場合は、その製品が動作確認されたものであるかどうかを事前に確認してください。Sybase IQ 製品の Sybase Certification Report については、Technical Documents を参照してください。(http://www.sybase.com/support/techdocs/)

仮想バックアップ

仮想バックアップ (NULL バックアップ) では、IQ ストアのテーブル・データを除くすべての IQ データベースがバックアップされます。IQ ストアに関しては別途、オペレーティング・システム・レベルでコピーを作成する必要があります。仮想バックアップからリストアを実行するには、まず、オペレーティング・システム・レベルで作成した IQ ストアのコピーをリストアし、次に仮想バックアップの IQ フル・リストアに取りかかります。

仮想バックアップでは、次のものがバックアップされます。

- すべての IQ カタログ・データ
- すべての IQ メタデータ
- 個々のテーブルには依存しない IQ ストアのすべてのメタデータ (フリーリスト、バックアップ、チェックポイントの情報を含む)

仮想バックアップでは、上記以外のテーブルのデータやメタデータはバックアップされません。

仮想バックアップを作成するには、BACKUP コマンドに VIRTUAL DECOUPLED パラメータか VIRTUAL ENCAPSULATED パラメータのどちらかを指定して、フル IQ バックアップを実行します。VIRTUAL パラメータを指定した場合、IQ ストアのテーブル・データとメタデータはバックアップ・ファイルにはコピーされません。

仮想バックアップの種類

仮想バックアップには次の 2 種類があります。

- **カプセル化仮想バックアップ** – システム・レベル・バックアップのリストアに続いて IQ 仮想バックアップのリストアを実行し、データベースのフル・リストアに相当する結果を得ます。
- **分離仮想バックアップ** – システム・レベル・バックアップのリストア、IQ 仮想バックアップのリストア、フル・リストア以降のインクリメンタル・リストアの順に実行し、データベースのフル・リストアに相当する結果を得ます。

カプセル化仮想バックアップ

手順を増やすことなく、テーブル・データのシステム・レベル・バックアップと仮想バックアップの一貫性を維持するには、バックアップ・コマンドの実行中にバックアップ・トランザクションごとにシステム・レベル・バックアップを作成する必要があります。VIRTUAL ENCAPSULATED 'shell-command' パラメータを指定すると、バックアップ操作の一部として任意のシェル・コマンドを実行して、これらのセマンティックを保証できます。シェル・コマンドが 0 以外のステータスを返した場合、バックアップ操作はエラーを返します。ユーザは、シェル・コマンドがシステム・レベル・バックアップを正しく実行していることを確認する必要があります。

❖ カプセル化仮想バックアップの実行

- 次のような SQL 文を使用します。

```
BACKUP DATABASE FULL VIRTUAL ENCAPSULATED  
'dd if=iqdemo.iq of=iqdemo.iq.copy'  
TO 'iqdemo.full'
```

❖ カプセル化仮想バックアップからのリストア

- 1 IQ ストアのシステム・レベル・コピーをリストアします。
- 2 バックアップ・ファイルからフル IQ リストアを実行します。
- 3 IQ データベースを起動します。

分離仮想バックアップ

システム・レベル・バックアップがバックアップ・トランザクション外で実行されると、IQ ストア・バックアップは IQ バックアップ・ファイルとの一貫性を失います。しかし、非仮想 IQ インクリメンタル・バックアップは仮想フル・バックアップと共に、一貫性のあるデータベースを表します。これは、IQ インクリメンタル・バックアップでは、仮想フル・バックアップ中またはそれ以降に変更された IQ ストアのすべてのデータとメタデータがコピーされるからです。バックアップ・コマンドで発生する自動コミットや自動チェックポイントによっても IQ ストアに変更が加えられ、独立したシステム・レベル・バックアップの一貫性が失われるということに注意してください。インクリメンタル・リストアを適用せずにデータベースを使用すると、予期しない結果が得られます。

❖ 分離仮想バックアップの実行

- 1 次のような SQL 文を使用してフル IQ バックアップを実行します。

```
BACKUP DATABASE FULL VIRTUAL DECOUPLED  
TO 'iqdemo.full'
```

- 2 シェル・コマンドを使用して、IQ ストアのシステム・レベル・バックアップを実行します。

```
dd if=iqdemo.iq of=iqdemo.iq.copy
```

- 3 非仮想インクリメンタル IQ バックアップを実行します。

```
BACKUP DATABASE INCREMENTAL SINCE FULL  
TO 'iqdemo.isf'
```

❖ 分離仮想バックアップからのリストア

- 1 IQ ストアのシステム・レベル・コピーをリストアします。次に例を示します。

```
dd if=iqdemo.copy of=iqdemo.iq
```

- 2 IQ フル・バックアップ・ファイルからリストアします。

```
RESTORE DATABASE iqdemo.db FROM 'iqdemo.full'
```

- 3 IQ インクリメンタル・バックアップ・ファイルからリストアします。

```
RESTORE DATABASE iqdemo.db FROM 'iqdemo.isf'
```

- 4 IQ データベースを起動します。

SAN スナップショットまたはシャドウ・ハードウェアを使用した仮想バックアップ

Storage Area Network (SAN) スナップショットまたはシャドウ・ハードウェアは、メイン・データベースではなくシャドウ・コピー上でシステム・レベル・バックアップを実行できるようにして、バックアップ・プロセスの柔軟性を高めます。仮想バックアップの一部であるシステム・レベル・バックアップの代わりに、シャドウを分離できます。次に、IQ ストアのシャドウ・コピーに対して、システム・レベル・バックアップを実行できます。これにより、フル・バックアップを短時間でできるようになります。

システム・レベルのバックアップの実行

IQ データをバックアップする最も信頼性の高い方法は、BACKUP コマンドによるものです。ただし、この項の手順に注意深く従えば、IQ データベースに対してシステム・レベルのバックアップを実行できます。

IQ データベースのバックアップにシステム・レベルのバックアップを使用するには、これらの手順に従う必要があります。適切な防護策をとることなくシステム・レベルのバックアップから IQ データベース・ファイルのリストアを試みると、データが損失したり一貫性が失われたりする場合があります。その原因は、システム・レベルのバックアップ中のデータベースにおけるアクティビティか、失われたファイルのいずれかです。

データベースの停止

システム・レベルのバックアップの実行中、IQ データベースは停止させておく必要があります。マルチプレックス・システムでのシステム・レベルのバックアップの実行については、『Sybase IQ マルチプレックスの使用』を参照してください。

システム・レベルのバックアップを開始するには、IQ データベースをシャットダウンする必要があります。また、システム・レベルのバックアップが完了するまで、どのユーザも IQ データベースを開始しないようにしてください。

データベース停止の確認

.db ファイルを保護するために、データベースが正常に停止するとファイルは読み取り専用になり、データベースの使用時に読み書きができるように設定されます。バックアップを実行するスクリプトを書き込んでいる場合は、ファイルのアクセス・モードを確認するスクリプトを書き込んで、データベースが停止しているかどうか確認することをおすすめします。

データベースが停止したままであることを確認するために、スクリプトによって、.iqmsg ファイルのサイズがスクリプトの開始時と終了時で同じであることをチェックできます。スクリプトの実行中にデータベースが起動していた場合は、.iqmsg ファイルのサイズが大きくなります。

適切なファイルのバックアップ

必要なファイル

次のファイルをバックアップする必要があります。

- すべての SYSTEM DB 領域ファイル (通常は *dbname.db* という名前)

注意 カタログ・ストアに追加の DB 領域が存在する場合、SYSDBSPPACES にリストされます。

- システム・リカバリに必要なトランザクション・ログ・ファイル (通常は *dbname.log* という名前)
- IQ_SYSTEM_MAIN DB 領域ファイル (通常は *dbname.iq* という名前)
- IQ メイン・ストアに追加された DB 領域用のファイル

次のファイルのサイズを保存します。

- IQ_SYSTEM_TEMP DB 領域ファイル (通常は *dbname.iqtmp* という名前)
- IQ_SYSTEM_TEMP に追加されたファイル

テンポラリ DB 領域をバックアップする必要はありません。IQ は、データベース起動時に現在のサイズを示すファイルを認識してさえいれば、どのテンポラリ DB 領域でも再構築できます。したがって、テンポラリ DB 領域を保持するために使用されたファイルやロー・デバイスのレコードは簡単に残せます。

オプション・ファイル

`dbname.iqmsg`、`$IQDIR15/logfiles/*.srvlog`、`$IQDIR15/logfiles/*.stderr` ファイルなどの ASCII メッセージ・ファイルがリストアに必要ではない場合でも、これらのファイルをバックアップすることをおすすめします。リストア中に問題が発生した場合、データベースを停止してからバックアップを開始したかどうかを確認できる情報が `.iqmsg` ファイルに入ります。

これらのファイルは、発生したデータベース障害の原因を診断するときに役立つ場合があります。あとで分析に使用するため、リストア前にコピーを作成してください。

IQ メッセージ・ログの循環を有効にしている場合は、`.iqmsg` ファイルをバックアップしておく、診断のために必要になった場合に、すべてのメッセージにアクセスできます。詳細については、「[メッセージ・ロギング](#)」(9 ページ)を参照してください。

メッセージ・ログ・アーカイブが有効な場合 (IQMsgMaxSize サーバ・オプションまたは `-iqmsgsz` サーバ起動スイッチが 0 以外で、IQMsgNumFiles サーバ・オプションまたは `-iqmsgnum` サーバ起動スイッチが 0 以外の場合)、メッセージ・ログのアーカイブは自動的にバックアップされます。アーカイブされるメッセージ・ログの最大サイズは 128GB で、ほとんどの場合、このサイズでバックアップに対応できます。

注意 メッセージ・ログ・アーカイブのバックアップは、サーバを再起動する前に行う必要があります。サーバが再起動した後、`dbname.iqmsg` ファイルが一杯になった場合、既存のログ・アーカイブは無視され、新しいアーカイブが作成されます。古いアーカイブ・ログを保持するには、サーバの再起動前にファイルをバックアップします。

バックアップ・リストの更新

SYSTEM、IQ_SYSTEM_MAIN、または IQ_SYSTEM_TEMP に関らずデータベースに追加されたどの DB 領域でも、使用するシステム・バックアップの仕様に追加することが非常に重要です。今後数ヶ月の間や組織編成が変更されたあとで `dbspace` が追加されると、この手順に失敗する場合があります。

必要なファイルのすべてをバックアップしているかどうかを確認するには、システム・レベルのバックアップ用のスクリプトを使用します。バックアップを開始する前に、システム・バックアップの仕様の中に収められている DB 領域のリストと、`SYSDFILE` (システム・データベース用) や `SYSIQFILE` (IQ データベース用) から選択したものをこのスクリプトで比較します。

ロー・デバイスとシンボリック・リンク

使用するデータベース・ファイルがロー・デバイス上にある場合は、システム・バックアップによって `/dev/*` 内のデバイス名だけでなく、ロー・デバイスの内容もバックアップされていることを確認します。

また、推奨方法に従ってロー・デバイスの名前にシンボリック・リンクが使用されている場合、シンボリック・リンクからシステム・バックアップ・ユーティリティにアクセスし、デバイスがバックアップされることを確認してください。

システム・レベルのバックアップからのリストア

システム・レベルのバックアップからリストアする必要がある場合は、バックアップ時と同様にデータベース・サーバが停止していることを確認してください。詳細については、「[データベースの停止](#)」(461 ページ) を参照してください。マルチプレックス・データベースをリストアするときは、すべてのクエリ・サーバと書き込みサーバを停止する必要があります。

全ファイルの有無の確認

リストアする前にバックアップ内容のテーブルを確認し、IQ に必要なファイルがすべて存在していることを確かめます。ファイルのリストは使用するアプリケーションによって異なります。必要なファイルとオプション・ファイルについては、「[適切なファイルのバックアップ](#)」(461 ページ) を参照してください。

テンポラリ DB 領域ファイルに関しては、ファイルまたはロー・デバイスのファイル名 (またはシンボリック・リンク) とサイズが正しいかどうかを確認します。テンポラリ `dbspace` ファイルの内容は、データベースを再起動するまで関連性はありません。

所有権とパーミッションの確認

システム・レベルのリストア中に、所有権のレベルとパーミッションのレベルが変更されないことを確認してください。

データベースの検証

データベースのバックアップは、データベース内部の一貫性が維持されている場合にのみ有効です。バックアップでは必ず、データベースが利用可能な状況にあるかどうか事前に確認されます。しかし、バックアップを実行する前に、リストアするデータベースを検証し、その状態が安定しているかどうかを確認することをお勧めします。リストア・プログラムでは、データベースが存在していない可能性があるため、リストア・データの一貫性が維持されているかどうかのチェックは行いません。

使用するデータベースを検証するには、次のコマンドを発行します。

```
sp_iqcheckdb 'check database'
```

`sp_iqcheckdb` ストアド・プロシージャとサーバ起動スイッチは、IQ データベース一貫性チェッカ (DBCC) のインタフェースです。

DBCC には、大量の一貫性チェックを実行するための検証モードがいくつか用意されています。データベースの一貫性をチェックするモードは 3 種類あり、アロケーション・マップをリセットするモードは 1 つです。`sp_iqcheckdb` コマンド文字列にターゲットとして `'database'` を指定した場合、どのモードでもすべてのデータベース・オブジェクトがチェックされます。このコマンド文字列には、個別のテーブルとインデックスも指定できます。個別のテーブル名を指定した場合、それらのテーブルに作成されているすべてのインデックスもチェックされます。

`sp_iqcheckdb` の 3 つの検証モードのアクション、出力結果、処理速度の一覧については、『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「[第 7 章 システム・プロシージャ](#)」の表 7-6 を参照してください。

`DBCC_LOG_PROGRESS` データベース・オプションは、`sp_iqcheckdb` に対して、進行メッセージを IQ メッセージ・ファイルに出力するよう指示します。これらのメッセージにより、`sp_iqcheckdb` プロシージャの処理状況を追うことができます。

`sp_iqcheckdb` はバックアップの前または後と、データベースに問題発生が疑われる場合に実行してください。`sp_iqcheckdb` の使用方法と `sp_iqcheckdb` 出力の意味については、「[第 13 章 システムのリカバリとデータベースの修復](#)」を参照してください。`sp_iqcheckdb` の構文の詳細については、『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「[第 7 章 システム・プロシージャ](#)」を参照してください。

マルチプレックス・データベースの検証

`sp_iqcheckdb` は、IQ マルチプレックスの書き込みサーバでのみ実行してください。`sp_iqcheckdb` をマルチプレックスのセカンダリ・サーバで実行すると、エラーが返されます。

`sp_iqcheckdb` の同時実行性の問題

`sp_iqcheckdb` をデータベース全体で実行すると、`sp_iqcheckdb` は使用中のすべてのデータベース・ページを読み取ります。I/O をできるだけ効率的に実行するため、このプロシージャはデータベース・サーバの処理能力のほとんどを消費します。このためシステムで同時に実行している他のアクティビティは、通常より処理が遅くなります。DBCC の CPU 使用率を低めに抑えるには、`sp_iqcheckdb` のパラメータ `resources resource-percent` を指定して、CPU とスレッドの数の比を制御します。`resources` パラメータの使用方法については、「[第 13 章 システムのリカバリとデータベースの修復](#)」の「[sp_iqcheckdb 実行時のリソースの問題](#)」を参照してください。

`sp_iqcheckdb` の実行中に他のユーザがアクティブである場合、トランザクションの表示だけが結果に反映されます。

データベースのリストア

データベースを作成してフル・バックアップを行うと、必要に応じてそのデータベースをリストアできるようになります。Sybase IQ では、バックアップ開始時の自動 CHECKPOINT の状態にデータベースをリストアします。

リストア前の作業

データベースをリストアする前に、次の条件を満たしていることを確認してください。

- DBA 権限が必要です。
- アーカイブ・バックアップから読み取り専用ファイルまたは DB 領域をリストアする場合、RESTORE 文の発行時に、データベースが実行中であっても構いません。また、管理者がデータベースに接続することもできます。バックアップに含まれている名前は、データベース・システムのテーブル情報と一致していれば、読み取り専用ファイルのパス名と一致している必要はありません。

データベースは、READWRITE FILES ONLY またはすべてのファイルのバックアップの FULL、INCREMENTAL SINCE FULL、または INCREMENTAL リストア目的では、実行しないでください。

データベースは、読み取り専用ファイルのバックアップをリストアするために実行しているかまたは実行していないかも知れません。読み取り専用 DB 領域の特定のファイルをリストアする場合または IQ_SYSTEM_MAIN 以外の読み取り／書き込み DB 領域の読み取り専用ファイルをリストアする場合、DB 領域はオフラインである必要があります。この制限は、IQ_SYSTEM_MAIN には適用されません。読み取り専用 DB 領域は、読み取り専用状態が同じ場合は、選択的リストアを使用してリストアできます。

- データベース内のすべてのファイルをリストアするか、バックアップのみの目的で読み取り／書き込みファイルからすべてのファイルをリストアする場合、utility_db データベースに接続する必要があります。utility_db データベースの詳細とこれを使用するために必要な権限の設定方法については、プラットフォームに対応した『インストールおよび設定ガイド』を参照してください。utility_db の起動方法については、「ユーティリティ・データベース・サーバのセキュリティ」(345 ページ) を参照してください。
- データベース内のすべてのファイルをリストアするか、バックアップのみの目的で読み取り／書き込みファイルからすべてのファイルをリストアする場合、どのユーザもリストアするデータベースには接続できません。指定したデータベースにアクティブな読み取り専用ユーザまたは読み取り／書き込みユーザがいると、エラーが発生して RESTORE が終了します。

次の 2 つの起動スイッチを使用して接続を制限することをおすすめします。

- -gd DBA を使用する。稼働中のサーバのデータベースの起動と停止を、DBA 権限を持つユーザのみが行えるようにするため (データベースを起動または停止するには、クライアントはすでにサーバに接続している必要があるため、このスイッチは接続を妨げません)。
- -gm 1 を使用する。非常時に DBA がサーバに接続して、他の接続を停止できるよう、接続の制限に加えて単一の接続を使用できるようにするため。

接続を制限するための別の方法としては、アップグレードを実行しようとするサーバの接続に対して次のように指定し、

```
sa_server_option('disable_connections', 'ON')
```

を実行し、リストア後に同じ接続で

```
sa_server_option('disable_connections', 'OFF')
```

を実行するという方法があります。この方法の欠点は、別の DBA 接続からの緊急アクセスが不可能なことです。

- データベースを適切なサーバにリストアする必要があります。また、サーバには必要なアーカイブ・デバイスがなければなりません。Sybase に備わっているリストア機能を実行する場合、バックアップ作成時と同じ数のアーカイブ・デバイス(ディスク・ファイルまたはテープ・ドライブ)が必要です。
- フル・リストアの場合は、カタログ・ストア(デフォルトでは .db ファイル)とトランザクション・ログ(デフォルトでは .log ファイル)がリストア先のロケーションに存在しないようにしてください。これらのファイルのいずれかが存在するときは、それを削除するか別のディレクトリに移動させてから、フル・リストアを実行する必要があります。

フル・リストアを開始すると、古いデータベースは一度すべて破壊されてから再作成されます。誤ってフル・リストアが実行されないように、カタログ・ストアとトランザクション・ログのファイルを手動で削除する必要があります。

- インクリメンタル・リストアの場合は、カタログ・ストア(.db ファイル)が必要です。このファイルがリストア先以外のロケーションに存在する場合は、「データベース・ファイルの移動」(470 ページ)に示す手順に従ってください。このファイルが存在しない場合、実行できるのはフル・リストアのみです(インクリメンタル・リストアの前にフル・リストアを実行すると、正しいファイルが正しい場所に配置されます)。
- どのインクリメンタル・リストアの場合も、最後のリストアを実行したあとにデータベースを変更しないでください。

リストアには、データベースとサーバへの排他的アクセスが必要です。データベースが不用意にオープンされないよう DBA による制御を強化するには、**-gd DBA** オプション・セットを使用してデータベース・サーバを起動してください。ただし、リストアするデータベースは起動しません。別のユーザが接続できないようにするため、RESTORE によって自動的にこのデータベースが起動されます。

フル・バックアップおよびすべての後続のインクリメンタル・バックアップを含むバックアップ全体、またはバックアップのセットをリストアする必要があります。バックアップ・アーカイブからの個々のファイルのリストアは、読み取り専用 DB 領域およびファイルでのみサポートされています。ただし、RESTORE コマンドの RENAME 句を使用して、データベース・ファイルを新しいロケーションに移動できます。

データベースをリストアする前に、RESTORE SQL 文の VERIFY 句を使用して、データベースのバックアップ (Sybase IQ バージョン 12.6 以降の既存のデータベースのバックアップ) を検証できます。詳細については、「[データベースのバックアップの検証](#)」(478 ページ) を参照してください。

DB 領域の変更に対応したリストア

一連のインクリメンタル・リストア中は、実行されるオペレーションに応じて必要な DB 領域が作成、削除されます。たとえば、データベースのフル・バックアップを行い、次に DB 領域をデータベースに追加してからインクリメンタル・バックアップを行うとします。これらのバックアップ・データからリストアを行うと、インクリメンタル・リストアを開始するときに、RESTORE によって新しい DB 領域用のファイルが作成されます。同様に dbspace を削除するとリストア中にこの dbspace が削除されますが、実際のファイルは削除されません。

SYSTEM DB 領域の SYSFILE システム・テーブルの file_name カラムは、リストア時に更新されません。SYSTEM dbspace では、データベース作成時の名前が file_name カラムに常に反映されます。SYSTEM DB 領域のファイル名が、データベース・ファイルの名前になります。

ディスク・バックアップ・ファイルのリストア

ディスク・バックアップを行ってからテープにそのファイルに移す場合は、バックアップ作成時と同じ名前を使用してそれらのファイルをディスクに戻す必要があります。Sybase IQ では、テープ・ディレクトリに移したディスク・ファイルを、テープから直接リストアすることはできません。

Sybase に備わっているバックアップとリストアの機能を使ってリストアする場合は、リストア用のアーカイブ・デバイス (ディスク・ファイル) をバックアップ作成時と同じ数だけ指定する必要があります。

テープ・バックアップ・ファイルのリストア

テープからリストアする場合は、ユーザはテープの IQ データの開始位置を調整する必要があります。RESTORE ではテープの開始位置は調整されません。

Sybase に備わっているバックアップとリストアの機能を使ってリストアする場合は、リストア用のテープ・デバイスをバックアップ作成時と同じ数だけ使用する必要があります。

インクリメンタル・リストア用のファイル指定

インクリメンタル・リストアの場合は、IQ ストアおよびカタログ・ストアのどちらでも、リストアするファイルが置きかえるファイルの数やサイズと一致しなければなりません。

リストア中のデータベース変更を防ぐ

一連のインクリメンタル・リストアの実行中、全セットのリストアが完了する前にユーザがデータベースを変更すると、残りのインクリメンタル・リストアは実行されません。たとえば、1つのフル・リストアと2つのインクリメンタル・リストアで構成されたセットがある場合、フル・リストアが完了してから2番目または3番目の **RESTORE** コマンドが発行されるまでの間にユーザが書き込んだトランザクションがコミットされると、インクリメンタル・リストアを実行できなくなります。この場合は、フル・バックアップをリストアし、インクリメンタル・リストアをもう一度行う必要があります。

最後のリストアからデータベースが変更された場合にインクリメンタル・リストアを実行しようとする、次のエラーが発生します。

```
Database has changed since the last restore
```

注意 Sybase IQ では、以前のリストアからデータベースが変更された場合にインクリメンタル・リストアを実行することはできません。また、ユーザによるデータベースの変更を防止することはできません。リストアの処理がすべて完了するまでデータベースに変更が加えられないようにすることは、DBA またはシステム・オペレータの役割です。

互換バックアップからのリストア

RESTORE では、Sybase IQ 15.0 以降のデータベース・ファイルをリストアできます。データベースのフォーマットが変更されたため、バージョン 12.x で作成されたバックアップからのリストアは実行できません。

Sybase IQ 12.x データベースのデータを Sybase IQ に移動するには、次の手順に従います。

- 1 バージョン 12.6 の『インストールおよび設定ガイド』に記載された移行手順に従って、バージョン 12.6 ESD #11 以降にアップグレードします。
- 2 15.2 の『インストールおよび設定ガイド』に記載された移行手順に従います。

RESTORE では、Sybase IQ のバックアップを SQL Anywhere データベースにリストアすることはできません。

RESTORE 文

データベースをリストアするときは、『リファレンス：文とオプション』の「第1章 SQL 文」の「**RESTORE 文**」にある構文を参照してください。この文には、読み取り専用および読み取り読み取り／書き込みファイルと DB 領域の要件についても記述されています。

この文を発行するには、DBA として `utility_db` データベースに接続する必要があります。

db_file と 1 つ以上の *archive_device* を指定する必要があります。

db_file 用に、デフォルトで *.db* サフィックスを付けて作成されたデータベースのカタログ・ストア・ファイルのロケーションを指定します。フル・パス名またはデータベースが作成されたディレクトリを起点としたパス名を指定できます。新しいパス名を指定すると、カタログ・ストアとそれに基づいて作成されたすべてのファイルが該当のロケーションに移動されます。ただし、**RENAME** 句に含めたファイルは移動されません。

バックアップの場合と同じく、各 *archive_device* に API (サード・パーティ製) を指定します。Sybase API の場合、リストアする物理的なテープ・デバイスまたはディスク・ファイルの名前を指定します。サード・パーティ製 API の場合、*archive_device* 文字列の内容はベンダによって異なります。ロー・ディスク・デバイスをアーカイブ・デバイスにしないでください。Sybase API を使用してディスク・ファイルからリストアする場合は、バックアップ作成時と同じ数のアーカイブ・デバイスを指定しなければなりません。

警告! テープ・デバイス名のスペルを間違え、システムで使用できないテープ・デバイス名を書き込むと、**RESTORE** はそのテープ・デバイスをディスク・ファイルとみなし、そこから読み取ろうとします。

デバイス指定の詳細については、「[アーカイブ・デバイスの指定](#)」(450 ページ) を参照してください。

注意 Windows でテープ・デバイスからリストアを行う場合は、リストア用のテープ・デバイスを指定するとき、バックアップ時のように円記号 (¥) を二重にする必要はありません。

例 1 – 同じロケーションへのリストア

この Windows の例では、データベースを *iquser.db* にリストアします。データベースは 2 つのディスク・ファイルからリストアされます。また、すべてのデータベース・ファイルは元のロケーションへリストアされます。

```
RESTORE DATABASE 'iquser.db'  
FROM 'c:¥¥iq¥¥backup1'  
FROM 'c:¥¥iq¥¥backup2'
```

読み取り／書き込みファイルおよび読み取り専用ファイルのリストアの例については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」の「[RESTORE 文](#)」を参照してください。

データベース・ファイルの移動

データベース・ファイルを新しいロケーションに移動する必要がある場合 (ディスク・ドライブの 1 つに障害が発生した場合など) は、次のいずれかの方法を使用します。

- カタログ・ストア (デフォルトでは `.db` ファイル) の入ったデータベース・ファイルを移動するため、`db_file` として新しい名前を指定する。
- トランザクション・ログ・ファイルの移動または名前変更のため、トランザクション・ログ・ユーティリティ (dblog) を使用する。構文と詳細については、『ユーティリティ・ガイド』の「[トランザクション・ログ・ユーティリティ \(dblog\)](#)」を参照してください。
- その他のデータベース・ファイルを移動するため、RENAME オプションを使用する。

新しいデバイスへのリストア

ロー・デバイスにリストアするときは、そのデバイスのサイズが、リストアする DB 領域を十分に保持できるサイズかどうかを確認します。IQ RESTORE はロー・デバイスのサイズをチェックし、DB 領域をリストアするために十分なサイズがなければエラーを返します。

オペレーティング・システムはロー・デバイスのわずかな領域を使用し、残りの領域を IQ データベースが占めます。DB 領域をリストアするときは、ロー・パーティションには、IQ DB 領域とオペレーティング・システム用に確保されている領域の両方が必要です。

IQ のメイン DB 領域またはテンポラリ DB 領域をロー・パーティションにリストアするには、次に示すように、システム・テーブルから各 IQ DB 領域に必要なロー・デバイスのサイズを調べます。

```
SELECT segment_type, file_name, block_count,
       data_offset, block_size,
       (block_count * block_size) + data_offset AS raw_size
FROM SYS.SYSIQFILE, SYS.SYSIQINFO
where segment_type != 'Msg' ORDER BY 1,2
```

`segment_type` と `file_name` は参考情報です。‘Main’ または ‘Temp’ タイプのセグメントがロー・パーティションに格納されることはあっても、メッセージ・ファイル (‘Msg’ タイプ) が格納されることはありません。 `file_name` は DB 領域の名前です。

`block_count` は、IQ によって使用されるブロックの数を表す整数型カラムです。

`data_offset` は、オペレーティング・システム用に確保されているバイト数を表す整数型カラムです。

`block_size` は、IQ ブロックあたりのバイト数を表す整数型カラムです。

`raw_size` は、この DB 領域をリストアするのに必要なロー・デバイスの最小サイズ (バイト単位) を表す整数型カラムです。元のロー・デバイスより 10MB 以上大きいローデバイスにリストアすることをおすすめします。

例 2 – カタログ・ストアの移動

この例では、例 1 と同じデータベースをリストアします。ただし例 2 では、カタログ・ストア・ファイルとそれに基づいて作成されたすべてのデータベース・ファイルを移動します。これには、次のように新しいロケーション `c:\$newdir` を使用して元のファイル名を置き換えます。

```
RESTORE DATABASE 'c:\$newdir\$iqnew.db'
FROM 'c:\$iq\$backup1'
FROM 'c:\$iq\$backup2'
```

Sybase IQ は、次のようにカタログ・ストア以外のデータベース・ファイルを移動します。

- **RENAME** 句を指定すると、ファイルはそのロケーションに移動します。
- **RENAME** 句を指定せずに、相対パス名を使用してファイルを作成した場合、このファイルはデータベース・ファイルの新しいロケーションに基づいてリストアされます。つまり、カタログ・ストア・ファイルを保持する **SYSTEM DB** 領域に基づいて作成された元のファイルは、カタログ・ストア・ファイルに基づいてリストアされます。カタログ・ストアに基づいて作成された元のファイルは、カタログ・ストアに基づいてリストアされます。
- **RENAME** 句を指定せずに、絶対パス名を使用してファイルを作成した場合、ファイルは元のロケーションにリストアされます。

つまり、データベース全体を移動する場合は、**RENAME** 句でデータベース内のすべての **IQ DB** 領域 (必須 **DB** 領域、テンポラリ **DB** 領域、ユーザ定義 **DB** 領域) の新しいロケーションを指定します。**SYSTEM DB** 領域のみ、**RENAME** 句に含めません。

一部のファイルだけを移動する場合、元のファイルを上書きすることに問題はなりません。実際に移動させるファイルの名前だけを変更してください。

SYSIQDBFILE テーブルの表示と一致するように、それぞれの `dbfile name` を指定します。また、`new dbspace path` を新しいロー・パーティションとして指定するか、その **DB** 領域の新しいフル・パス名または相対パス名として指定します。

RENAME オプションを使用して、部分リストアを指定することはできません。

RENAME 句の相対パス名は、データベースまたは **DB** 領域を作成したときと同じように機能します。メイン **IQ** ストアの **DB** 領域、テンポラリ・ストアの **DB** 領域、およびメッセージ・ログは、`db_file` (カタログ・ストア) のロケーションに基づいてリストアされます。またユーザが作成した **IQ** ストアの **DB** 領域は、カタログ・ストアが入っているディレクトリに基づいてリストアされます。

フル・バックアップとインクリメンタル・バックアップの両方のリストア中にファイル名を変更するときは、リストアのセット内で一貫した **DB** 領域名とパスを使用してください。これは、ファイルの名前を正しく変更するもっとも安全な方法です。

フル・バックアップとインクリメンタル・バックアップの間で DB 領域を追加し、データベース・ファイルの名前を変更する場合は、フル・リストアだけでなくインクリメンタル・リストア用の **RENAME** 句も必要です。同様に、バックアップ中に DB 領域が 1 つ削除されると、DB 領域の削除後に発生したバックアップから、リストア用の **RENAME** 句を 1 つ減らす必要があります。

使用するデータベースの DB 領域名のリストを取得する方法については、「[DB 領域名の記録](#)」(484 ページ)を参照してください。**RENAME** 句に含める正しい名前について理解できます。

例 3 – ユーザ DB 領域の移動

この例では、この章で先に示した例のフル・バックアップやインクリメンタル・バックアップを行う方法について示します。この場合にメディア障害が発生すると、UNIX のロー・パーティションを使用できなくなります。このロー・パーティション上のユーザ定義 dbfile である **IQ_USER** を、新しいロー・パーティション `/dev/rdsk/c1t5d2s1` へ移動させる必要があります。その他のデータベース・ファイルに影響はありません。

はじめに、**utility_db** データベースに接続します。次に 2 つのテープ・デバイスにあるフル・バックアップをリストアします。この場合に使用するテープ・デバイスは、バックアップの作成時と同じものです。ただし、同じ数のアーカイブ・デバイスや同じメディア・タイプ(テープまたはディスク)を使用して、正しい順序で同じテープ・セットを使用している場合にかぎり、デバイスが異なることもあります。詳細については、「[正しい順序のリストア](#)」(473 ページ)を参照してください。

最初の **RESTORE** コマンドは次のとおりです。

```
RESTORE DATABASE 'iquser'  
FROM '/dev/rmt/0n'  
FROM '/dev/rmt/1n'  
RENAME IQ_SYSTEM_MAIN TO '/dev/rdsk/c2t0d1s1'  
RENAME IQ_SYSTEM_TEMP TO '/dev/rdsk/c2t1d1s1'  
RENAME IQ_SYSTEM_MSG TO 'iquser.iqmsg'  
RENAME IQ_USER TO '/dev/rdsk/c1t5d2s1'
```

インクリメンタル・バックアップをリストアする、2 番目の **RESTORE** コマンドは次のとおりです。

```
RESTORE DATABASE 'iquser'  
FROM '/dev/rmt/0n'  
RENAME IQ_SYSTEM_MAIN TO '/dev/rdsk/c2t0d1s1'  
RENAME IQ_SYSTEM_TEMP TO '/dev/rdsk/c2t1d1s1'  
RENAME IQ_SYSTEM_MSG TO 'iquser.iqmsg'  
RENAME IQ_USER TO '/dev/rdsk/c1t5d2s1'
```

注意 DB 領域が 1 つだけ新しいロケーションへリストアされるので、最後の **RENAME** 句だけを使用してこれらのコマンドを発行することもできます。ここで示したようにファイルまたはロー・パーティションのすべてをリストアすると、リストアされる各ロケーションを正確に把握できます。

CATALOG ONLY オプションを使用してヘッダ情報を表示する

CATALOG ONLY オプションを使用すると、データベースのヘッダ情報が表示され、`.backup.syb` ファイルに書き込まれます。このオプションでは、カタログ・ストアまたは IQ ストアのいずれのデータもリストアされません。表示される情報のリストについては、「[バックアップ・ログの内容](#)」(483 ページ) を参照してください。

CATALOG ONLY を指定するときは、FROM `archive_device` 句を含める必要があります。ただし、RENAME 句は省略します。

データ・ソースと設定ファイルの調整

データベースを移動するとき、データベースの新しいロケーションに反映させるためデータ・ソース、設定ファイル、統合化ログインを修正する必要があります。

正しい順序のリストア

フル・バックアップからリストアを行うと、バックアップ作成時に使用された各ブロックがディスクに書き込まれます。インクリメンタル・バックアップからリストアを行うと、以前のバックアップ (または以前のフル・バックアップ) と今回のバックアップの間に変更されたブロックだけがディスクに書き込まれます。

リストアするバックアップごとに個別の RESTORE コマンドを使用し、正しい順序でフル・バックアップとインクリメンタル・バックアップをリストアする必要があります。RESTORE によってバックアップが順番にリストアされ、この順番が誤りであると判断された場合は、次のエラーが表示されます。

```
SQL Code: -1012009
SQL State: QUA09
This restore cannot immediately follow the previous restore.
```

正しい順番を判断するには、バックアップ・ログに格納されているバックアップ・ファイルの情報が必要です。このファイルの内容とロケーションについては、「[バックアップとリストアに関する情報の取得](#)」(482 ページ) を参照してください。

次のようにバックアップをリストアします。

- データベースの一貫性が失われている場合や、任意のファイルを新しいロケーションに移動する場合は、FULL バックアップをリストアする必要があります。
- 最後に行ったバックアップが FULL バックアップの場合や、データベースを以前の状態にリストアしてからインクリメンタル・バックアップを行う必要がある場合は、フル・バックアップだけをリストアします。

- INCREMENTAL_SINCE_FULL バックアップ後にデータベース障害が発生した場合は、まず最新の FULL バックアップをリストアしてから、INCREMENTAL_SINCE_FULL バックアップをリストアします。
- INCREMENTAL_SINCE_FULL バックアップではなく、最後に FULL バックアップを行ってから INCREMENTAL バックアップを 1 回以上実行した場合は、まず FULL バックアップをリストアしてから、作成された順に INCREMENTAL バックアップをリストアします。

通知ストア・プロシージャ `sp_iqrestoreaction` を使用して、安定したデータベース・セットを作成するのに必要な一連のリストア・アクションを示すこともできます。示された手順と上記のルールを常に比較して確認するようにしてください。また、ストア・プロシージャは、データベース・ファイルの移動には含まれません。

任意のバックアップ内で、テープをリストアする順番も重要です。特に、バックアップ・セット (あるアーカイブ・デバイス上の任意のバックアップによって作成されたテープのセット) ごとにテープの順番を次のように追跡する必要があります。

- 最初のカatalog・ストアのバックアップを含むテープ・セットを、最初のアーカイブ・デバイス上にリストアする。
- 各セットの中で、作成された順にテープをリストアする。
- セットをインタリーブすることはできないので、あるセットをリストアしてから別のセットをリストアする。
- 各セットにおける順番が正しいかぎり、最初のセットの次にリストアするセットの順番は関係ない。

バックアップ作成時と同じ数のドライブを使用してリストアし、誤って異なるセットのテープをインタリーブしないようにしてください。

例

3つのアーカイブ・デバイスを使用してフル・バックアップをリストアし、3種類のテープ・セット (A、B、C) が作成されたとします。各セットの内容とリストアする順番は次のとおりです。

セット A テープ A_1 、 A_2 、 A_3 。テープ A_1 と A_2 にはカatalog・ストアが含まれています。まず、このセットを最初のデバイスでリストアする必要があります。

セット B テープ B_1 と B_2 。セット A のあとで、このセットをリストアします。セット C の先でも後でも問題ありません。2番めのデバイスと3番めのデバイスのいずれかでリストアできます。

セット C テープ C_1 、 C_2 、 C_3 。セット A のあとで、このセットをリストアします。セット B の先でも後でも問題ありません。2番めのデバイスと3番めのデバイスのいずれかでリストアできます。

リストア・プログラムは、各セットのテープが単一のデバイス上に正しい順番で存在しているかどうかチェックします。この条件が満たされていない場合はエラーが発生し、正しいテープを供給するまでリストアは続行されません。カタログ・ストアが含まれたセット以外は、任意のデバイスにどのセットを挿入しても問題ありません。

注意 必ずカタログ・ストアのテープ・セットを最初にリストアしてください。リストア・プログラムはこれをチェックしません。

これらの規則はディスク・ファイルの場合にも当てはまりますが、特定のディスク・デバイスで複数のファイルをバックアップすることはありません。

リストア後の再接続

Sybase IQ では、特定の状況下で、データベースへの接続に DBF パラメータとデータベース・ファイル名が必要になります。具体的には、Interactive SQL を使用していて、*utility_db* への接続中にバックアップからデータベースをリストアした場合に必要です。

たとえば、次のように DBF パラメータを指定します。

```
CONNECT USING 'uid=DBA;pwd=sql;dbf=nodel/users/fiona/mydb.db;
links=tcPIP{host=serv1;port=1234};eng=serv1_iqdemo'
```

Sybase IQ 12.6 ESD5 までは、リストアしたデータベースに次の構文で接続できました。

```
CONNECT DATABASE mydb USER DBA IDENTIFIED BY SQL
```

現在は、上記のコマンドを実行すると、specified database not found というエラーが返されます。

このエラーを防ぐもう 1 つの方法は、*utility_db* への接続中に **START DATABASE** コマンドを入力する方法です。たとえば次のように入力します。

```
START DATABASE mydb
```

Interactive SQL で接続しているときには、この方法を使用します。

リストア後にトランザクション・ログ名を変更する

データベース内にあるその他すべてのファイルについて名前を変更したり、移動したりするときも、ログ・ファイルの場合と同じ作業を行います。ログ・ファイルの移動または名前変更には、トランザクション・ログ・ユーティリティ (dblog) を使用します。このユーティリティは、次のときに実行します。

- 新しいデータベース名を使用して **RESTORE** を使用した後
- **RENAME** オプションを指定して **RESTORE** を使用した後

注意 トランザクション・ログ・ファイル名を変更するときは、データベース上でデータベース・サーバを実行しないでください。

何らかの制約があってデータベースをリストアしていない場合であっても、**dblog** を使用してトランザクション・ログの名前を変更することはできます。トランザクション・ログ・ユーティリティは、**dblog** コマンドライン・ユーティリティを使用してシステム・コマンド・ラインからアクセスできます。詳細については、『ユーティリティ・ガイド』の「[トランザクション・ログ・ユーティリティ \(dblog\)](#)」を参照してください。

リストア後のデータベースの検証

テープが正しい順番でリストアされたことを確認するには、データベースのリストアの完了後にストアド・プロシージャ **sp_iqcheckdb** を実行します。インクリメンタル・バックアップのセットをリストアしている場合は、各バックアップのリストアが完了してから **sp_iqcheckdb** を実行するのがもっとも安全な方法です。ただし、時間を節約したいのであれば、必ず最新のインクリメンタル・バックアップをリストアしたあとで、**sp_iqcheckdb** を実行するのもよいでしょう。詳細については、「[データベースの検証](#)」(463 ページ) を参照してください。

排他的な書き込みアクセスが必要なリストア

RESTORE を開始すると、その他のユーザは指定されたデータベースにアクセスできなくなります。フル・バックアップからリストアを実行してから、1つ以上のインクリメンタル・バックアップからリストアを実行する場合は、この2種類のリストア間にユーザがデータベースを修正しないようにしてください。修正はできますが、インクリメンタル・リストアを実行できなくなります。この場合は、リストア処理を最初からもう一度実行しなければなりません。

この制限は、リカバリ中にシステムがクラッシュすると必要になるすべてのインクリメンタル・リストアにも適用されます。リストア中に発生したシステム障害またはメディア障害からリカバリする場合は、次のいずれかを実行する必要があります。

- 元の順番でフル・リストアとインクリメンタル・リストアの操作を続ける。
- フル・リストアを実行してから、データベースを完全にリカバリするために必要なインクリメンタル・リストアを実行する。

デフォルトのデータベース・サーバ起動設定 `-gd DBA` は、データベースを起動するときに必要な DBA 権限を与えます。DBA が `RESTORE` を実行すると、データベースが自動的に起動し、リストアに必要な情報を取得してから停止します。リストアの最後に、このコマンドはデータベースを起動してチェックポイントを発行し、再びデータベースを停止します。このプロシージャによって、リストア中の DBA による排他的な書き込みアクセスが確保されます。

インクリメンタル・リストアがすべて完了すると、DBA によって `START DATABASE` コマンドが再び発行され、その他のユーザがデータベースにアクセスできるようになります。

マルチプレックス・データベースをリストアする場合は、『Sybase IQ マルチプレックスの使用』を参照してください。

ヘッダ情報の表示

`RESTORE` 文に `CATALOG ONLY` オプションを指定して (`FILE` 句は指定せずに) 実行すると、ヘッダ・ファイルの内容を表示できます。

ヘッダ・ファイルに表示される情報の例については、「[バックアップ・ログの内容](#)」(483 ページ) に示されたサンプル・バックアップ・ログの `RESTORE` 行を参照してください。`RESTORE` に `CATALOG ONLY` を指定して実行すると、実際の `RESTORE` に対するバックアップ・ログ・エントリと同じフォーマットで情報が生成されます。

バックアップ・アーカイブの詳細な情報を入手するには、最初のバックアップ・アーカイブに対応するファイル・パスを受け入れるコマンド・ライン・ユーティリティ `db_backupheader` を使用します。このユーティリティによって、バックアップ・アーカイブ・ファイルが読み込まれます。データベースには接続しません。

バックアップ・アーカイブの情報は次の内容で構成されています。

- バックアップ情報
- バックアップ時のデータベース情報
- データベース内の DB 領域ごとの DB 領域情報
- DB 領域内の dbfile ごとの dbfile 情報

リストア中のエラーからのリカバリ

操作の初期段階でインクリメンタル・リストアに失敗した場合は、そのデータベースをそのまま使用できます (データベースは存在し、リストア開始前に一貫性は維持されていたとみなされます)。フル・リストアに失敗した場合、そのデータベースを使用できなくなります。

操作の特定の段階を過ぎてから障害が発生すると、リストア・プログラムは、一貫性が失われたことを示すマークをデータベースに付けます。この場合は、FULL RESTORE によるリカバリしか実行できません。FULL RESTORE の実行中に障害が発生した場合は、以前の FULL BACKUP に戻る必要があります。

データベースのバックアップの検証

Sybase IQ には、RESTORE SQL 文の VERIFY 句を使用して、Sybase IQ バージョン 12.6 以降の既存のデータベースのバックアップを検証するメカニズムがあります。検証プロセスでは、フル・バックアップ、インクリメンタル・バックアップ、フル・バックアップ以降のインクリメンタル・バックアップ、または仮想バックアップの指定された Sybase IQ データベース・バックアップ・アーカイブを検証し、リストア・プロセスでチェックされても書き込み操作が実行されないエラーと同じエラーがないかどうか、指定されたアーカイブをチェックするようサーバに指示します。ステータスとエラー・メッセージは、すべてサーバ・ログ・ファイルに書き込まれます。

バックアップ検証プロセスは、データベースのホストとは異なるホストで実行できます。RESTORE VERIFY を実行するには、DBA、BACKUP、または OPERATOR 権限が必要です。

RESTORE VERIFY 句の構文と使用方法については、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「RESTORE 文」を参照してください。

注意 バックアップ・アーカイブの検証は、データベース一貫性チェッカ (DBCC: Database Consistency Checker) の verify モード (sp_iqcheckdb 'verify...') とは異なります。RESTORE VERIFY では、バックアップ・アーカイブの一貫性を検証して、そのアーカイブをリストアできるかどうかを確認するのに対し、DBCC ではデータベースのデータの一貫性を検証します。

sp_iqcheckdb 'verify...' を実行してから、バックアップを実行してください。一貫性のないデータベースをバックアップし、同じバックアップ・アーカイブからリストアした場合、RESTORE VERIFY で検証の成功が報告された場合でも、データは一貫性のない状態のままになっています。

インクリメンタル・バックアップの検証

既存のデータベース・ファイルとインクリメンタル・アーカイブの互換性をチェックするには、RESTORE... VERIFY COMPATIBLE 句を使用します。RESTORE...VERIFY COMPATIBLE を呼び出したシステム上にデータベース・ファイルが存在しない場合は、エラーが返されます。フル・バックアップの検証時に COMPATIBLE を指定した場合、このキーワードは無視されます。フル・バックアップのリストア中に、互換性チェックを実行する必要はありません。

インクリメンタル・リストアで COMPATIBLE を含めずに RESTORE VERIFY を指定した場合、Sybase IQ は DB 領域を検索せず、互換性チェックを実行しません。ファイルが存在しない場合でも、警告は表示されません。互換性チェックが実行されるのは、COMPATIBLE 句を含めた場合だけです。

インクリメンタル・リストアで RESTORE VERIFY COMPATIBLE を指定しても、IQ カタログ・ストアまたは Sybase IQ DB 領域が存在しない場合は、互換性チェックを実行することはできません。この場合、エラーが報告され、操作は失敗します。

インクリメンタル・バックアップの検証中、RESTORE VERIFY COMPATIBLE プロセスでは Sybase IQ DB 領域を読み取り専用モードでオープンして一貫性検査を実行します。DB 領域が修正されることはありません。

インクリメンタル・リストアで、データベースが修正されていたり、特定のインクリメンタル・アーカイブがデータベースの正しいアーカイブではない場合、RESTORE VERIFY COMPATIBLE によって、エラー Database has changed since last restore. (SQLCODE -1012008, SQLSTATE QUA08) または This restore cannot immediately follow the previous restore. (SQLCODE -1012009, SQLSTATE QUA09) が報告されます。

検証進行状況レポート

RESTORE VERIFY プロセスでは、コマンドで指定されたすべてのストライプが検証されます。検証プロセスでストライプおよびそれに対応するファイルがチェックされると、検証済みの IQ ブロック単位で進行状況が報告されます。IQ ブロックが 5000 ブロック検証されるたびに、次のようなメッセージがサーバ・ログ・ファイルに書き込まれます。

```
5000/100000 (5%) Blocks verified
```

すべての IQ ブロックが検証された時点で、最後のメッセージがサーバ・ログ・ファイルに書き込まれます。

検証アクションが開始されると、サーバ・ログに RESTORE VERIFY Started というメッセージが書き込まれ、検証済みの IQ ブロックの数が書き込まれます。検証アクションが完了すると、RESTORE VERIFY successfully completed というメッセージが書き込まれます。例を示します。

```
I. 11/17 06:45:24. VERIFY RESTORE Started
I. 11/17 06:45:24. Total number of IQ blocks to be verified:
764
I. 11/17 06:45:24. Total number of IQ blocks verified: 764/764
( 100 % )
I. 11/17 06:45:24. VERIFY RESTORE Successfully Complete
```

検証エラー・レポート

検証プロセスでエラーが検出された後もプロセスを続行できる場合は、アーカイブが引き続きチェックされ、検出されたエラーの情報がログに記録されます。検証を続行できるエラーは次のとおりです。

```
Header of block to be restored appears to be corrupted.
(SQLCODE -10120111, SQLSTATE QUA11)

Media data appears corrupted (bad checksum).
(SQLCODE -1012012, SQLSTATE QUA12)
```

```
Media meta data appears corrupted (boundary record).
(SQLCODE -1012013, SQLSTATE QUA13)
```

```
Media meta data appear corrupted (multiple begin boundary
records). (SQLCODE -1012014, SQLSTATE QUA14)
```

上記のいずれかのエラーが検出され、アーカイブの末尾まで検証プロセスを続行できる場合、Sybase IQ は次のエラーを報告します。

```
The verification of the provided archive has failed.Please
check the server log for details of the errors thrown during
verify.
```

RESTORE に関する上記以外エラーが検出された場合は、発生したエラーが報告され、検証プロセスは中止されます。

バックアップとシンボリック・リンク (UNIX のみ)

バックアップにシンボリック・リンクが含まれる場合、Sybase IQ によって、意図せぬディレクトリに DB 領域が作成される場合があります。たとえば、次のファイルに dbspace を作成したとします。

```
-rw-r--r-- 1 fiona sybase 122880000 Feb 26 18:27 iqdemo.db
-rw-r--r-- 1 fiona sybase 122880000 Feb 26 18:27 iqdemo.iq1
-rw-r--r-- 1 fiona sybase 122880000 Feb 26 18:27 iqdemo.iq2
-rw-r--r-- 1 fiona sybase 122880000 Feb 26 18:27 iqdemo.iq3
-rw-r--r-- 1 fiona sybase 122880000 Feb 26 18:27 iqdemo.iqtmp
-rw-r--r-- 1 fiona sybase 122880000 Feb 26 18:27 iqdemo.iqmsg
```

事前に次のリンクを作成しておくと、そのリンクが示すディレクトリ (またはロー・パーティション) で DB 領域が作成されます。

```
lrwxrwxrwx 1 fiona sybase 14 Feb 26 17:48 iqdemo.iq1 ->
LINKS/iqdemo.iq1
lrwxrwxrwx 1 fiona sybase 14 Feb 26 17:48 iqdemo.iq2 ->
LINKS/iqdemo.iq2
lrwxrwxrwx 1 fiona sybase 18 Feb 26 17:48 iqdemo.iq3 ->
/dev/rdisk/c2t6d0s0
```

これらのファイルを **CATALOG ONLY** オプションを使用してバックアップしリストアする場合、この情報は実際には保存されていないため、これらのファイルがリンクしていたことはどこにも表示されません。

Sybase IQ は、シンボリック・リンクが置かれたディレクトリに実際に存在していたかのようにファイルを保存します。そしてリストアするときに、ディレクトリまたはデータベース名が付けられたロー・パーティションにファイルが再作成されます。リストア時に存在するかどうかに関らず、リンクが再び使用されることはありません。データベースは元のロケーションにリストアされます。

アテンドなしバックアップ

ATTENDED OFF オプションを使用すると、オペレータなしでバックアップを実行することを指定できます。Sybase IQ では、現在 2 種類のアテンドなしバックアップ機能をサポートしています。

- バックアップ中、オペレータが指示に応答する必要はありません。
- アーカイブ・ドライブがスタッカ・ドライブの場合、一連のテープを単一のドライブに自動的にロードします。スタッカ・ドライブは、アテンド付き、アテンドなし両方のバックアップで使用できます。

アテンドなしバックアップでは、テープ・メディア障害を除いた予期されるエラーの原因がすべて検出され、バックアップを開始する前にすべてレポートされます。これには、ディスクやテープの使用可能な領域や、一貫したサイズおよびブロック係数などが含まれます。

ディスクを使用してアテンドなしバックアップを行う場合、Sybase IQ はまずディスクの空き領域がバックアップ用に十分であるかどうかテストします。ただし、領域を確保するために、バックアップ ファイル用領域を事前に割り付けることはしません。別のユーザがそのディスクにデータを書き込んでバックアップ用の領域が足りなくなると、ディスク領域がなくなった時点でバックアップが失敗します。

テープを使用してバックアップを行う場合は、各テープに格納するデータ量を見積もり、BACKUP コマンドの *TO archive_device* パラメータに見積もったキロバイト数を指定する必要があります。バックアップ・プログラムは、データベースのバックアップに必要な容量を判断するため、内部に格納された情報をチェックします。テープに十分な空き領域があると判断されると、バックアップが続行されます。ただし、テープの使用可能な領域を過剰に見積もったために領域が足りなくなると、その時点でバックアップに失敗します。

アテンドなしバックアップで *SIZE* パラメータを省略する場合、バックアップ全体を 1 つのテープに収める必要があります。

サード・パーティ製バックアップ製品を使用する場合は、デバイスの仕様、ファイルのサイズ、スタッカ・デバイスなどのバックアップに必要な情報を、ベンダ情報の文字列に含めなければなりません。詳細については、各ベンダのマニュアルを参照してください。

注意 Sybase IQ では、アテンドなしリストアを実行できません。

バックアップとリストアに関する情報の取得

Sybase IQ では、バックアップ・メディアの管理に役立つバックアップ・ログ (*backup.syb*) を生成します。このログはバックアップの作成やデータベースのリストアには使用されませんが、バックアップやリストアの最中に関連情報がこのファイルに記録されます。

注意 特定のバックアップに関する情報だけを表示するには、**CATALOG ONLY** オプションを使用して **RESTORE** を実行します。このオプションを使用すると、バックアップのヘッダ・ファイルがファイルからではなくメディアから表示されるため、DBA はテープまたはファイル上の情報を識別できます。詳細については、「[ヘッダ情報の表示](#)」(477 ページ) を参照してください。

バックアップ・ログの配置

backup.syb は、ASCII テキスト・フォーマットのファイルです。このファイルのロケーションは、次のようにサーバ起動時の環境変数の設定によって異なります。

- UNIX では、サーバがこのファイルを次の順番で各ロケーションに配置しようとします。
 - IQLOGDIR15 環境変数で指定されたディレクトリ
 - HOME 環境変数で指定されたディレクトリ
 - アカウント情報から取得されたホーム・ディレクトリ
 - 現在のディレクトリ (サーバを起動したディレクトリ)

ファイルが home ディレクトリに配置された場合は、隠しファイルにするためにファイル名の先頭に “.” が付けられます。このファイルが現在のディレクトリに置かれている場合、プレフィックスは付きません。

- Windows では、サーバがこのファイルを次の順番で各ロケーションに配置しようとします。
 - IQLOGDIR15 環境変数で指定されたディレクトリ
 - サーバの実行ファイルが収められているディレクトリ

バックアップ・ログの内容

バックアップやリストアを実行するたび、次のフィールドがカンマで区切られてバックアップ・ログに含まれます。

- 操作 (Backup または Restore)
- バージョン
- データベース名
- データベースの種類 (Sybase IQ または SQL Anywhere)
- バックアップまたはリストアの実行日時
- 作成者のユーザ ID
- バックアップ／リストアの種類。Full、Incremental、Incremental_since_full または Database File Only (SQL Anywhere データベースのみ)。
- メソッド：Archive (IQ または Anywhere データベース用) または Image (Anywhere データベース専用)
- ロケーション
- 一重引用符で囲まれたコメント (BACKUP コマンドに入力した場合)。コメントに引用符が含まれる場合は、一重引用符が 2 つ連続して表示されます。

バックアップ・ログの例を示します。

```
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 16:25:00.000', DBA, Full, Arch, TED_FULL00, ''
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 16:53:00.000', DBA, Incr, Arch,
TED_X_bkup_inc, ''
RESTORE, 2.0, all_types.db, ASIQ, '2009-01-31 16:25:00.000', DBA, Full, Arch, TED_FULL00, ''
RESTORE, 2.0, all_types.db, ASIQ, '2009-01-31 16:53:00.000', DBA, Incr, Arch,
TED_X_bkup_inc, ''
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 20:07:00.000', DBA, InSF, Arch,
A_partial2_yes_sf, ''
BACKUP, 2.0, all_types.db, ASIQ, '2009-01-31 20:07:00.000', DBA, InSF, Arch,
A_partial2_yes_sf, ''
```

バックアップ・ログの管理

バックアップ・メディアのパージ後に、バックアップ・ログをクリーンアップすることをおすすめします。クリーンアップにはテキスト・エディタを使用します。BACKUP または RESTORE によってこのファイルに情報が記録されたあとは、この情報の精度はチェックできないので、編集には注意が必要です。

1つのサーバにはバックアップ・ログが1つだけ存在します。サーバはこのファイルに対して読み込み／書き込みが可能でなければなりません。システム管理者は、その他のユーザによるこのファイルへのアクセスを制限できます。1つのシステム上で複数のデータベース・サーバを実行している場合は、各サーバごとに異なる IQLOGDIR15 環境変数を設定し、バックアップ・ログを個別に生成してください。

警告！ バックアップまたはリストアの実行中、バックアップ・ログを編集しないでください。BACKUP または RESTORE がファイルにデータを書き込み中、そのファイルを修正するとファイル内の情報が無効となります。

DB 領域名の記録

RESTORE の RENAME オプションを使用してデータベースまたはその DB 領域の1つを移動する必要がある場合、データベース内の DB 領域名をすべて把握する必要があります。DB 領域名は各データベースの SYSFILE テーブルに示されていますが、リストア時にはこのテーブルを使用できません。この情報を参照するには、最初のバックアップ・アーカイブ・ファイルで db_backupheader を実行します。または、データベースをバックアップするたびに、sp_iqdbspace および sp_iqfile ストアド・プロシージャを実行するか、次の文を発行します。

```
SELECT dbf.dbfile_name, f.*
FROM SYSFILE f, SYSDBFILE dbf
WHERE f.file_id=dbf.dbfile_id
```

データベースが置かれているディスク以外の場所にこのクエリの結果を保持して、必要なときに DB 領域名の完全なリストを参照できるようにしてください。

Interactive SQL で次のスクリプトを実行することもできます。このスクリプトでは、実際にどのファイルのロケーションも変更しない場合に、使用する RENAME 句のセットを含んだ出力ファイルが生成されます。代わりに新しいファイル・ロケーションを指定し、このスクリプトによって作成されたファイルを RESTORE 文に使用できます。

注意 リストアする必要があるときにデータベースが存在しない場合があるので、データベースのバックアップが完了してからこのスクリプトを実行します。

```
-- Get dbspace and IQ file names and add
-- rename syntax including quotation marks

select 'rename' as 'restore ... rename' ,
dbf.dbfile_name as 'IQ file' , 'to' as 'to' ,
'''' + f.file_name + '''' as 'file_path'
from SYSFILE f, SYSDBFILE dbf
where f.store_type=2 and f.file_id=dbf.dbfile_id
```



```
-- Send output to a file in proper format
-- without delimiters or extra quotation marks

output to restore.tst delimited by ' ' quote '';

-- This produces a restore.tst file like the following:
-- rename IQ_SYSTEM_MAIN to '/dev/rdsk/c2t0d1s7'
-- rename IQ_SYSTEM_TEMP to '/dev/rdsk/c2t1d1s7'
-- rename IQ_SYSTEM_MSG to 'all_types.iqmsg'
```

データ・バックアップ方式とリカバリ方式の決定

効率的なシステム・バックアップ方式を開発するには、フル・バックアップ、インクリメンタル・バックアップ、フル・バックアップ以降のインクリメンタル・バックアップの最適な組み合わせを決定する必要があります。その次にバックアップの実行スケジュールを設定します。さまざまなバックアップ・オプションの機能と、それらがデータベース障害の発生時にリストア速度へ及ぼす影響を理解してください。

日常のバックアップの計画

データベースの作成直後にそのフル・バックアップを実行し、ベース・ポイントを決定します。その後、定められたスケジュールに基づいてフル・バックアップとインクリメンタル・バックアップを実行してください。データベースを大きく変更したあとのバックアップは特に重要です。

バックアップ・プランは、次の条件によって異なります。

- 使用するシステムへのロード
- 使用するデータベースのサイズ
- データに加えられた変更の数
- バックアップ時間の短縮とリカバリ時間の短縮における相対的な重要性

バックアップの種類決定

フル・バックアップ、インクリメンタル・バックアップ、またはフル・バックアップ以降のインクリメンタル・バックアップのいずれを実行するか決定する場合は、バックアップの作成にかかる時間とそのリストアにかかる時間のバランスをとる必要があります。また、必要なメディア容量も考慮してください。インクリメンタル・バックアップは、実行時間が比較的短く、テープまたはディスクの使用領域も小さくなります。フル・バックアップは、実行時間が比較的長く、大量の領域が必要です。

フル・バックアップ以降のインクリメンタル・バックアップはその中間に位置します。このバックアップは、インクリメンタル・バックアップと同じ実行時間およびメディア容量から開始されます。ただし、データベースの変更量とフル・バックアップ後のバックアップ数が増加すると、実行時間や必要なメディア容量がフル・バックアップと同じになったり、さらに大きくなったりする可能性があります。

一般的に、リストア操作はこの逆になります。たとえば、非常に古いフル・バックアップや大量のインクリメンタル・バックアップからのリストアにはかなり時間がかかり、新しいフル・バックアップよりも大量の領域が必要です。

インクリメンタル・バックアップの利点は、最後のバックアップまたはフル・バックアップから変更されたデータだけをバックアップするので、データベース全体をバックアップする場合よりも実行時間が短縮され、使用する領域が小さくなることです。その反面、インクリメンタル・バックアップに頼りすぎると、最終的にリストアする場合の実行時間が長くなります。

たとえば、データベースのフル・バックアップを1度行えば、理論的にはそれ以降インクリメンタル・バックアップだけを実行すればよいはずですが、将来的にリカバリ作業にかかる時間が非常に長くなり、フル・バックアップを定期的に行う場合よりもテープまたはディスク領域が大量に必要であるため、この方法は実用的ではありません。バックアップの実行中、その他のユーザも読み込み/書き込みアクセスができることに注意してください。ただしリストア中は、誰もそのデータベースを使用することはできません。大量のインクリメンタル・バックアップをリストアすると、リストアが終了するまでユーザはシステムを使用できないことに注意してください。

より優れた方法は、インクリメンタル・バックアップとフル・バックアップを組み合わせることです。

データベースをたくさん変更すればするほど、バックアップの重要性が高まり、インクリメンタル・バックアップの利点が少なくなります。たとえば、毎夜データの10パーセントかそれ以上に影響を及ぼす変更内容をデータベースで更新する場合は、そのたびにフル・バックアップ以降のインクリメンタル・バックアップを実行し、週1回フル・バックアップを行います。これに対して、データベースの変更が少ない場合は、毎月1回フル・バックアップを行って、その間はインクリメンタル・バックアップを行う方法が適しています。

バックアップとリストアの責任者の指定

多くの組織で、バックアップとリカバリの操作のすべてを実行するオペレータが存在します。Sybase IQ データベースのバックアップまたはリストアの担当者には、そのデータベースのDBA 権限が必要です。

バックアップとリストアのパフォーマンスの向上

データベースのバックアップまたはリストアにかかる合計時間は、フル・リストアとインクリメンタル・リストアの組み合わせ方法によって大きく異なります。バックアップおよびリストア操作の処理速度に影響を及ぼす要因は、その他にもあります。この要因には、アーカイブ・デバイスの数、データ検証、バックアップに使用できるメモリ容量、IQ ストアやカタログ・ストアのサイズなどがあります。

アーカイブ・デバイス数の増加

BACKUP 文の TO 句では、アーカイブ・デバイスの数を制御します。詳細については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」を参照してください。

データ検証の省略

BACKUP コマンドで CRC OFF を設定し、バックアップ時間とリストア時間を短縮することもできます。これによって周期的な冗長性検査が非アクティブに設定されます。CRC ON を設定すると、その後のリストア操作中にバックアップで算出された数値が検証され、両方のコマンドのパフォーマンスに影響を及ぼします。デフォルトは CRC ON です。この検査をオフにすると、パフォーマンスが高速になる代わりに、データ精度が低下する可能性があるので注意してください。

バックアップ・データのスパール

ディスクを使用してバックアップを作成してから、それらを長期保管するためテープにスパールすると、処理時間を短縮して効率的に作業できます。この方法を選択した場合は、データをディスクに戻してからリストアする必要があります。

バックアップ中の使用メモリの増加

バックアップ中バッファに使用されるメモリの量は、バックアップ時間に直接影響を及ぼします（主にテープ・バックアップの場合）。メモリの使用量は、BACKUP コマンドの BLOCK FACTOR パラメータによって制御されます。バックアップに時間がかかる場合は、BLOCK FACTOR の値を大きくしてこの時間を短縮できます。

BLOCK FACTOR の効果は、使用するオペレーティング・システムと、データベースの作成時に指定したブロック・サイズによって異なります。新しく作成されたデータベースの IQ ページ・サイズがデフォルトの 128KB であれば、ブロック・サイズはデフォルトの 8192 バイトになります。

UNIX の場合、デフォルトの **BLOCK FACTOR** 値は 25 です。**BLOCK FACTOR** 値は最低でも 25 に設定することをおすすめします。この値を指定して **BACKUP** を使用すると、ほとんどの UNIX テープ・ドライブで理想的にデータをバッファすることができます。これは、バックアップ中ドライブをビジー状態に保つのに十分なデータがメモリ内に保持されるためです。

Windows の場合、デフォルトの **BLOCK FACTOR** 値はデータベースのブロック・サイズに基づいて算出されます。通常、Windows の最大スループット値が設定されます。Windows のテープ・デバイスの処理方法が原因で、**BLOCK FACTOR** 値を大きくしてもバックアップ時間を短縮できない場合もあります。

システム負荷の分散

Sybase IQ では、データベース構造に影響を及ぼす操作を除き、その他のあらゆる読み取り／書き込み操作と同時にバックアップを実行できます。ただし、システム・リソース (ディスク、メモリ、CPU サイクル) を最大限に利用するために、システム使用率が低い時間帯にバックアップをスケジュールすることをおすすめします。

カタログ・ストアのサイズ制御

IQ データベースは、IQ ストアと基本となるカタログ・ストアで構成されています。

フル・バックアップとインクリメンタル・バックアップのどちらの場合も、**BACKUP** では最初にカタログ・ストアのフル・バックアップが行われます。通常、カタログ・ストアのサイズは非常に小さく、システム・テーブル、メタデータ、その他、データベース管理に必要な情報だけが含まれています。ただし、カタログ・ストア内に IQ 以外のテーブルを作成できます。IQ 以外のデータを、カタログ・ストアではなく SQL Anywhere 専用のデータベースへ個別に保存すると、IQ バックアップのパフォーマンスを向上させることができます。

バックアップでは、コミットされたデータベースの最新バージョンだけがコピーされます。トランザクションを開くときに使用されるその他のバージョンのページは、バックアップされません。

読み取り専用ハードウェアによるデータのアーカイブ

Health Insurance Portability and Accountability Act (HIPAA) やサーベンス・オクスリー法など、最近の法令では、データの保存と法規遵守に関するルールが厳格に定められており、変更が不可能で簡単にアクセスできる形式でデータをアーカイブしておく必要があります。データの量は数テラバイトに達することがあり、データの保存期間も数年から数十年に及びます。

これらの要件に対処できるようになったのは、WORM ディスク記憶領域ソリューションの進歩によるものです。WORM とは、Write Once, Read Many (書き込みは 1 回、読み込みは多数回) の略です。WORM 記憶領域の先駆けは、記憶領域内の各ロケーションに対して 1 回のみ永続的に書き込める光ディスク・テクノロジーです。Sybase IQ では、WORM ディスク・アレイのことを「読み取り専用ハードウェア」と呼びます。読み込み専用ハードウェアの機能は、WORM 保護レイヤを追加した低コストなディスク・アレイ・ハードウェアで実現されます。保護レイヤがあるため、データを「フリーズ」するまでは、ディスクを通常の読み書きに使用できます。

ユーザは、データをフリーズするときに、無期限または一定の保持期間を指定します。ディスクのフリーズは、ボリューム・レベルまたはファイル・レベルで可能です。フリーズされたデータは変更できません。また、保持期間は、延長は可能ですが、短縮はできません。

読み取り専用ハードウェアの機能は、WORM ディスク・アレイ・ハードウェアに限ったものではありません。DB 領域を読み取り専用に変更した後で、ロー・デバイスまたはファイル・システム・ファイルから書き込み権限を削除することもできます。

読み取り専用ハードウェアの使用

この項では、読み取り専用ハードウェアの一般的な操作について説明します。

注意 WORM ディスク・アレイ・ハードウェアがなくても、読み込み専用ハードウェアの機能は使用できます。DB 領域を読み取り専用に変更した後で、ロー・デバイスまたはファイル・システム・ファイルから書き込みの権限を削除することもできます。

❖ アーカイブの作成

db.db という単一のカタログ・ストア DB 領域と、*A*、*B*、*C* という 3 つのメイン IQ DB 領域で構成される IQ データベースがあるとします。

- 1 *t0* の時点で、3 つのメイン DB 領域すべてを読み取り専用に変更します。
- 2 データベースを停止して *db.db* をコピーするか、*dbbackup* を使用して、データベースを実行したままでコピーを作成し、*db.db* を *db.db0* にコピーします。
- 3 DB 領域 *A*、*B*、および *C* をハードウェア・レベルでフリーズします。*db.db0* を変更不可能な形式で格納します。たとえば、WORM デバイスのファイル・システム・ファイルに格納し、フリーズします。

これで、*t0* の時点のデータベースが変更不可能な形式でアーカイブされました。

❖ 新しい DB 領域の作成

- 1 *D* と *E* という、2つの新しいメイン DB 領域を作成します。
- 2 データベース *db.db* を運用データベースとして引き続き使用します。

t0 の時点で存在したデータベース・オブジェクト (テーブルやインデックスなど) が変更され、*db.db* と *db.db0* が同じではなくなっている場合があります。*t0* の時点で存在したテーブルが存在し、*t0* の時点で存在したデータのローが変更されずに含まれているかぎり、*db.db* は引き続き *A*、*B*、および *C* の各 DB 領域からデータを読み取ります。これらの条件が成り立たなくなった場合でも、*A*、*B*、および *C* が *db.db* から削除されていないかぎり、これらは引き続きオープンされます (これらの *db.db* を削除できるのは、*db.db* から見てこれらが空の場合に限られます)。

❖ アーカイブされたデータの調査

t0 の時点でアーカイブされたデータベースを調べる必要があるとします。

- 1 アーカイブされた読み取り専用の *db.db0* を、読み書き可能なファイル *db.db0.working* にコピーします。
- 2 *db.db0.working* を起動します。サーバ名 *db.db0.working* が運用システム *db.db* と競合しないかぎり、運用システムを停止する必要はありません。*db.db0.working* は、*A*、*B*、*C*、および *D* を読み取り専用モードで開きます。このとき、UNIX では、*db.db* がこれらのファイルを使用していることとの競合は生じませんが、Windows では共有違反が発生します。

カタログ・ファイル *db.db0.working* は読み取り／書き込みモードでオープンされます。

- 3 アーカイブされたデータベースを調べるためのユーザ *inv* を作成します。
- 4 *inv* に対し、ビュー、ストアド・プロシージャ、グローバルまたはローカル・テンポラリ・テーブルなど、調査に必要な構造体を作成するための RESOURCE パーミッションを付与します。

db.db0 および *A*、*B*、*C* は元のまま変更されません。

❖ 作業用アーカイブの更新

t0 から長い期間が経過した場合、ALTER DATABASE UPGRADE で IQ メイン・ストアのオブジェクトが変更されないときには、*db.db0.working* をアップグレードできます。

- 1 *t0* の時点で存在したテンポラリ DB 領域は、*db.db0.working* を起動するうえで必要ありません。サーバ起動スイッチ *-iqnotemp* を使用して *db.db0.working* を起動します。
- 2 テンポラリ DB 領域を削除して新しいものを作成するか、または *-iqnotemp* パラメータで作成されたテンポラリ領域を使用します。

❖ **新しいアーカイブの作成**

t1 時点の新しいアーカイブは次のように作成します。

- 1 *D* と *E* の両 DB 領域を読み取り専用にします。
- 2 *db.db* を *db.db1* にコピーします。
- 3 *D* と *E* をフリーズします。
- 4 *db.db1* を変更不可能な形式で保存します。
- 5 *F* や *G* など、新しいメイン DB 領域を作成します。
- 6 運用システム *db.db* を引き続き使用します。

アーカイブしたデータベース *db.db0* または *db.db1* はいつでも使用でき、両方を同時に使用することも可能です。これらを使用するには、*db.db0* または *db.db1* を作業ファイルにコピーし、サーバを起動するだけです。

「[アーカイブの作成](#)」(489 ページ) の手順を実行した後でこの項の手順を実行すれば、*db.db* のアーカイブ・バージョンをいくつでも作成できます。

システムのリカバリとデータベースの修復

この章について

データベース・サーバを再起動すると、Sybase IQ は自動的にリカバリを試みます。サーバをリカバリできずに再起動する場合、特にシステム障害や停電の後は、データベースの一貫性が失われることがあります。この章では、通常のリカバリ時の処理内容、データベースの一貫性を検証する方法、データベースの矛盾を修復する方法、特殊なりカバリ・モードについて説明します。

内容

トピック名	ページ
リカバリと修復の概要	493
通常のリカバリ	494
データベースの検証	494
データベースの修復	502
強制リカバリ・モード	509
トランザクション・ログを使わない緊急リカバリ	514
DBCC によってレポートされた問題の処理	515
DBCC エラー・メッセージ	518

リカバリと修復の概要

Sybase IQ サーバまたはデータベースの再起動中に問題が発生した場合は、この章の情報を利用してデータベース起動時の問題を診断し、データベースの一貫性を検証し、データベースを修復できます。障害発生後にサーバを再起動できた場合は、なるべくユーザーに接続を許可する前にデータベースを検証することをおすすめします。データベースの検証には、この章で説明する `sp_iqcheckdb` ストアド・プロシージャを使用します。

サーバまたはデータベースを起動できない場合、データベースは起動するがユーザーから接続できない場合、またはデータベースの検証中に問題が検出された場合は、強制リカバリを実行するか、データベースをリストアする必要があります。

サーバ・ログと IQ メッセージ・ログの調査

この章では、これらの各機能を実行する必要があるときに、それを判断する方法について説明します。また、データベースの検証、強制リカバリ、リーク領域リカバリ、インデックスの修復の詳細について説明します。データベースのリストアの詳細については、『システム管理ガイド第 1 巻』の「[第 12 章 データのバックアップ、リカバリ、アーカイブ](#)」を参照してください。

どのタイプのリカバリや修復が必要かを判断するには、サーバ・ログ (`servername.nnnn.srvlog`) および IQ メッセージ・ログ (`dbname.iqmsg`) の情報が必要となります。必要に応じて Sybase 製品の保守契約を結んでいるサポート・センタに提供できるように、情報を保持しておいてください。

たとえば、データの矛盾が検出された場合は、詳細な診断情報が `dbname.iqmsg` ファイルに含まれていることがあります。

通常のリカバリ

システムのリカバリ時には、コミットされていないすべて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 (コミットされなかったトランザクションが使用していたデータベース・ページのスナップショット) に使用されていたすべてのディスク領域が使用可能な領域のプールに返されます。マルチプレックス・データベースでないかぎり、このときのデータベースには、各永久テーブルの最後にコミットされたバージョンだけが格納されています。マルチプレックス・データベースには、セカンダリ・サーバからアクセスできるすべてのバージョンが格納されています。バージョン管理の詳細については、『システム管理ガイド第 1 巻』の「[第 10 章 トランザクションとバージョン管理](#)」を参照してください。

システム障害や通常のリカバリ時のシステム停止からのリカバリ時に、Sybase IQ はアクティブだったすべての接続を再オープンします。ユーザ接続数を設定する `-gm` オプションが障害発生時に有効だった場合は、少なくともサーバの停止時に実際に使用していたのと同じ接続数で IQ サーバを再起動する必要があります。

データベースの検証

停電などの異常終了によってサーバを再起動した後は、できるだけ早くデータベースの一貫性を検査してください。データベースの一貫性検査は、データベースのバックアップを実行する前にも行います。どちらの場合も、`sp_iqcheckdb` ストアド・プロシージャを使用して、データベースの一貫性の問題を検出および修復できます。

この項では、データベースの検証に `sp_iqcheckdb` を使用する方法について説明します。`sp_iqcheckdb` を使用して、検出された一貫性の問題を修復する方法の詳細については、「[データベースの修復](#)」の項を参照してください。

sp_iqcheckdb ストアド・プロシージャ

IQ データベース一貫性チェッカ (DBCC) は、データベースの検証を行います。sp_iqcheckdb ストアド・プロシージャは、サーバ起動オプションと組み合わせ、DBCC のインタフェースとして使用されます。sp_iqcheckdb コマンド文字列を指定して、さまざまなモードの検査と修復を選択します。コマンド文字列で特に指定しなければ、sp_iqcheckdb はすべてのデータベース・ページを読み込んで、データベースの一貫性を検査します。

注意 セカンダリ・サーバでは、sp_iqcheckdb によるフリー・リストの検査は行われません。それ以外の検査はすべて行われます。

DBCC には、実行する一貫性検査の量が異なる 3 種類のモードと、アロケーション・マップをリセットするためのモードがあります。sp_iqcheckdb コマンド文字列で個別の DB 領域、テーブル、パーティション、インデックス、またはインデックス・タイプを指定しないかぎり、どのモードでもすべてのデータベース・オブジェクトが検査されます。個別のテーブル名を指定した場合、それらのテーブルに作成されているすべてのインデックスもチェックされます。

注意 sp_iqcheckdb ストアド・プロシージャは、参照整合性をチェックせず、参照整合性に違反があった場合も修復しません。

sp_iqcheckdb の構文

sp_iqcheckdb 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』の「[第 7 章 システム・プロシージャ](#)」の「[sp_iqcheckdb プロシージャ](#)」を参照してください。

DBCC のパフォーマンス

DBCC の実行時間は、データベース検査全体のデータベースのサイズ、指定するテーブルやインデックスの数、マシンのサイズによって異なります。データベースの一部、つまり指定したテーブル、インデックス、またはインデックス・タイプだけを検査する場合は、データベース全体を検査する場合より時間がかかりません。sp_iqcheckdb の各モードの処理時間については、『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「[第 7 章 システム・プロシージャ](#)」の表 7-6 を参照してください。

DBCC のパフォーマンスを最大限にするには、sp_iqcheckdb コマンド文字列でパラメータをできるだけ具体的に指定します。可能であれば「allocation」または「check」検証モードを使用し、検査が必要なデータベース・オブジェクトが正確にわかっている場合は、テーブルまたはインデックスの名前を指定します。

**sp_iqcheckdb check
モード**

check モードでは、**sp_iqcheckdb** はすべての IQ インデックスの内部一貫性検査を実行し、各データベース・ブロックが正しく割り付けられているかどうかを検査します。出力可能なすべてのデータベース統計がレポートされます。このモードでは、すべてのデータ・ページが読み込まれ、すべてのタイプの割り付けの問題と、ほとんどのタイプのインデックスの矛盾が検出されます。多くのデータベースでは、check モードの方が verify モードよりかなり高速に実行されます。

check モードで実行するのは次のような場合です。

- クエリを実行したときに、メタデータ、NULL カウント、または個別カウント・エラーが返された場合

次の表は、check モードの例を示します。

表 13-1: sp_iqcheckdb check モードの例

コマンド	説明
sp_iqcheckdb 'check database'	データベース内のすべてのテーブルとインデックスの内部検査
sp_iqcheckdb 'check table t1'	テーブル t1 内のすべてのインデックスのデフォルト検査
sp_iqcheckdb 'check index t1c1hg'	インデックス t1c1hg の内部検査
sp_iqcheckdb 'check indextype FP database'	データベース内にある FP タイプのすべてのインデックスの検査

**sp_iqcheckdb verify
モード**

verify モードでは、**sp_iqcheckdb** は内部インデックスの一貫性と割り付けの検査に加えて、インデックス間の一貫性検査も実行します。出力可能なすべてのデータベース統計がレポートされます。各非 FP インデックスの内容が、対応する FP インデックスと照合されます。verify モードでは、すべてのデータ・ページが読み込まれ、すべてのタイプの割り付けの問題と、すべてのタイプのインデックスの矛盾が検出されます。

verify モードで実行するのは次のような場合です。

- クエリを実行したときに、メタデータ、NULL カウント、または個別カウント・エラーが返された場合

次の表は、verify モードの例を示します。

表 13-2: sp_iqcheckdb verify モードの例

コマンド	説明
sp_iqcheckdb 'verify database'	データベース内のすべてのインデックスの内容を検証する
sp_iqcheckdb 'verify table t1'	テーブル t1 内のすべてのインデックスの内容を検証する
sp_iqcheckdb 'verify index t1c1hg'	インデックス t1c1hg の内容を検証する
sp_iqcheckdb 'verify indextype HG table t1'	テーブル t1 内のすべての HG インデックスの内容を検証する

注意 個別の非 FP インデックスを check モードで検査すると、対応する FP インデックスが内部一貫性検査で自動的に検証され、DBCC の結果に表示されます。

sp_iqcheckdb allocation モード

allocation モードでは、sp_iqcheckdb は、各データベース・ブロックが内部物理ページ・マッピング構造 (ブロックマップ) に従って正しく割り付けられているかどうかを検査します。割り付けに関するデータベース統計もレポートされます。このモードは非常に高速に実行されます。ただし、allocation モードではインデックスの一貫性は検査されず、すべてのタイプの割り付けの問題を検出することはできません。

allocation モードで実行するのは、次のような場合です。

- 複合的に所有されたブロックによるリークしたブロックまたは矛盾するインデックスを検査する
- 強制リカバリ後に、sp_iqcheckdb を dropleaks モードで実行し、アロケーション・マップをリセットする (データベースをターゲットとして使用する)
- 重複ブロックや未所有のブロックを検査する (データベースまたは特定のテーブルやインデックスをターゲットとして使用する)
- ページ・ヘッダ・エラーが発生した場合

次の表は、allocation モードの例を示します。

表 13-3: sp_iqcheckdb allocation モードの例

コマンド	説明
sp_iqcheckdb 'allocation database'	データベース全体の割り付けの検査
sp_iqcheckdb 'allocation database dumpleaks'	データベース全体の割り付けの検査と、リークしたブロックに対するブロック番号の IQ メッセージ・ファイルへの出力
sp_iqcheckdb 'allocation table t1'	テーブル t1 の割り付けの検査
sp_iqcheckdb 'allocation index t1c1hg'	インデックス t1c1hg の割り付けの検査
sp_iqcheckdb 'allocation indextype LF table t2'	テーブル t2 内のすべての LF インデックスの割り付けの検査

テーブルの一部のパーティションがオフラインになっている場合は、パーティション・ターゲットを指定してテーブルの一部のみを検査できます。

すべてのモードを組み合わせて、1つのセッションで複数のチェックをデータベースに対して行うことができます。次の例では、sp_iqcheckdb は、CPU の半分を使って、テーブル t2 のパーティション p1 に対しては簡単なチェックを、インデックス i1 に対しては詳細なチェックを、データベース全体に対しては割り付けチェックを行います。

```
sp_iqcheckdb 'check table t2 partition p1
verify index i1
allocation database resources 50'
```

allocation モード・オプションは、DBCC コマンド 'allocation database' でのみ使用できます。

次の allocation モード・オプションを指定すると、影響を受けるデータベース・ブロックのブロック番号が、IQ メッセージ・ファイルに出力されます。

- `dumpleaks` – リーク・ブロック
- `dumpdups` – 重複ブロック
- `dumpunallocs` – 割り付けられていないブロック

`resetclocks` オプションは、内部データベースのバージョン・クロックが遅れている場合に、クロックの値を修正します。Sybase IQ の保守契約を結んでいるサポート・センタに連絡した場合を除き、`resetclocks` オプションはその他の目的には使用しません。

`resetclocks` オプションは、シングル・ユーザ・モードで実行する必要があり、DBCC コマンド 'allocation database' でのみ使用できます。`resetclocks` コマンドの構文は次のとおりです。

```
sp_iqcheckdb 'allocation database resetclocks'
```

`sp_iqcheckdb dropleaks`
モード

Sybase IQ サーバがシングルノード・モードで実行されている場合は、データベースまたは DB 領域のターゲットで `dropleaks` モードを使用して、データベース・ターゲット全体または指定された DB 領域ターゲットのアロケーション・マップをリセットできます。ターゲットが DB 領域の場合の `dropleaks` 操作は、その DB 領域での読み取り／書き込み操作も防ぐものでなければなりません。指定されたデータベースまたは DB 領域内のすべての DB 領域がオンラインになっている必要があります。

次の例では、最初の文によってデータベース全体のアロケーション・マップがリセットされ、2 番目の文によって DB 領域 `dbbsp1` のアロケーション・マップがリセットされます。

```
sp_iqcheckdb 'dropleaks database'
sp_iqcheckdb 'dropleaks dbspace dbbsp1'
```

注意 インデックス・エラーの修復には、`sp_iqrebuildindex` を使用してください。現在のところ、ジョイン・インデックスの修復はサポートされていません。

sp_iqcheckdb の出力

`sp_iqcheckdb` の出力は、DBCC によってレポートされた統計とエラーの詳細なリストで構成されます。0 以外の値だけが表示されます。エラーを含む行には、アスタリスク (*****) でフラグが付けられます。エラーが発生すると、DBCC がレポートする統計が不正確になる場合があることに注意してください。

DBCC エラー・メッセージの全リストについては、「[DBCC エラー・メッセージ](#)」(518 ページ) の項を参照してください。

`sp_iqcheckdb` の出力は、常に IQ メッセージ・ファイル (`iqmsg`) にコピーされます。`sp_iqcheckdb` の出力を別のファイルにリダイレクトするには、次のコマンドを入力します。

```
sp_iqcheckdb ># file_name
```

このコマンドで、`file_name` は出力を受け取るファイルの名前です。

`DBCC_LOG_PROGRESS` オプションが ON の場合、`sp_iqcheckdb` は進行メッセージを IQ メッセージ・ファイルに送ります。これらのメッセージから、ユーザは `sp_iqcheckdb` プロシージャの実行時の進行状況を追うことができます。

このコマンドの進捗ログの出力サンプルを次に示します。

```
sp_iqcheckdb 'check database'
```

```
IQ Utility Check Database
Start CHECK STATISTICS table: tloansf
Start CHECK STATISTICS for field: aqsn_dt
Start CHECK STATISTICS processing index:
ASIQ_IDX_T444_C1_FP
Start CHECK STATISTICS processing index:
tloansf_aqsn_dt_HNG
Done CHECK STATISTICS field: aqsn_dt
```

Future Version Errors

“DBCC Future Version Errors” というメッセージが表示された場合は、DBCC トランザクションが開始されてから DDL 操作が実行されています。DBCC は残りのテーブルの処理を続行しますが、リーク・ブロックの検査は実行されず、省略されたテーブルは統計には含まれません。

DBCC Future Version エラーを回避するには、`sp_iqcheckdb` を実行する前に `COMMIT` コマンドを実行します。

次の DBCC 出力は、Future Version エラーを示します。

```
=====|=====|=====
DBCC Verify Mode Report          |          |
=====|=====|=====
** DBCC Future Version Errors    |1        |*****
```

有効なデータベースの出力例

次は、`sp_iqcheckdb` を verify モードで実行したときの例です。エラーは検出されていません。リーク領域はなく、データベースの割り付けは一貫しており、すべてのインデックスで一貫性が保たれています。

この例で実行したコマンド・ラインは、`sp_iqcheckdb 'verify database'` です。DBCC はすべてのインデックスを検証しますが、ここに示すインデックス検証の出力は省略されています。

一貫性が保たれていると DBCC が判断した各インデックスには、結果セットで `verified` というマークが付けられます。

データベースの検証

Stat	Value	Flags
===== ===== =====		
DBCC Verify Mode Report		
===== ===== =====		
DBCC Status	No Errors Detected	
DBCC Work units Dispatched	75	
DBCC Work units Completed	75	
===== ===== =====		
Index Summary		
===== ===== =====		
Verified Index Count	86	
===== ===== =====		
Allocation Summary		
===== ===== =====		
Blocks Total	8192	
Blocks in Current Version	4855	
Blocks in All Versions	4855	
Blocks in Use	4855	
% Blocks in Use	59	
===== ===== =====		
Allocation Statistics		
===== ===== =====		
DB Extent Count	1	
Blocks Created in Current TXN	211	
Blocks To Drop in Current TXN	212	
Marked Logical Blocks	8240	
Marked Physical Blocks	4855	
Marked Pages	515	
Blocks in Freelist	126422	
Imaginary Blocks	121567	
Highest PBN in Use	5473	
Total Free Blocks	3337	
Usable Free Blocks	3223	
% Total Space Fragmented	1	
% Free Space Fragmented	3	
Max Blocks Per Page	16	
1 Block Page Count	104	
3 Block Page Count	153	
...		
16 Block Hole Count	199	
===== ===== =====		
Index Statistics		
===== ===== =====		
...		
Verified Index	fin_data.DBA.ASIQ_IDX_T209_C3_HG	
Verified Index	fin_data.DBA.ASIQ_IDX_T209_C4_FP	
Verified Index	product.DBA.ASIQ_IDX_T210_C1_FP	
...		
Verified Index	employee.DBA.ASIQ_IDX_T212_C20_FP	


```

Verified Index          |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP |
FP Indexes Checked     |68                                |
HNG Indexes Checked    |1                                  |
HG Indexes Checked     |17                                |
=====|=====
...

```

DBCC 出力には、Container Statistics、Buffer Manager Statistics、Catalog Statistics、Connection Statistics、Compression Statistics などの見出しで分類された詳細な統計情報も含まれます。Sybase IQ デモ用データベース `iqdemo` に接続した後で、`sp_iqcheckdb 'verify database'` コマンドを実行すると、出力可能な統計の例を参照できます。

sp_iqcheckdb 実行時のリソースの問題

`sp_iqcheckdb` の実行中にリソースの問題が発生した場合、`sp_iqcheckdb` の出力または `.iqmsg` ファイルに次のいずれかのメッセージが表示されることがあります。

- `Out of memory` および `DBCC Out of Memory Errors` この操作を実行するための十分なメモリがありません。他の IQ の操作や他のアプリケーションが `sp_iqcheckdb` ストアド・プロシージャと同時に実行されないようにする必要があります。
- `No buffers available` および `DBCC Out of Buffers Errors` DBA はバッファ・キャッシュ・サイズを増やす必要があります。

バッファ・キャッシュ・サイズを永続的に設定するには、データベース・オプション `TEMP_CACHE_MEMORY_MB` を使用します。サーバ起動スイッチ `-iqmc` および `-iqtc` を使用すると、データベース・オプションで設定したバッファ・キャッシュ・サイズの値を上書きできます。データベース・オプションとサーバ起動スイッチの両方を使用してバッファ・キャッシュ・サイズを設定する方法については、『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」の「[バッファ・キャッシュ・サイズの設定](#)」を参照してください。

DBCC は 1 つのインスタンスを実行するように最適化されているので、複数のデータベース一貫性検査を同時に実行しないでください。

DBCC の CPU 使用率を低めに抑えるには、`sp_iqcheckdb` のパラメータ `resources resource-percent` を指定して、CPU とスレッドの数の比を制御します。`resource-percent` のデフォルト値は 100 で、1 CPU あたり 1 つのスレッドが作成されます。この値は、ほとんどのマシンの負荷容量に適合します。DBCC をバックグラウンド・プロセスとして実行している場合は、`resource-percent` の値を 100 未満に設定してスレッドの数を減らします。スレッドの最小数は 1 です。

`resource-percent` が 100 よりも大きい場合、CPU の数以上のスレッドがあり、一部のマシン構成ではパフォーマンスが向上することがあります。

データベース・オプション `DBCC_PINNABLE_CACHE_PERCENT` を使用すると、`DBCC` のバッファ使用量を調整できます。

`DBCC_PINNABLE_CACHE_PERCENT` のデフォルトでは、キャッシュの 50% を使用します。詳細については、『リファレンス：文とオプション』の「第 2 章 データベース・オプション」の

「[DBCC_PINNABLE_CACHE_PERCENT オプション](#)」を参照してください。

データベースの修復

`sp_iqcheckdb` を `dropleaks` モードで実行すると、割り付けの問題を修復できます。`DBCC` が割り付けの問題を修復している最中にインデックスの矛盾が検出されると、エラーが生成され、割り付けの問題は修復されません。

リーク・ブロック (割り付けられているが使用されていないブロック) のリカバリの詳細については、「[リーク領域のリカバリ](#)」(512 ページ) の項を参照してください。

インデックス・エラーの分析

この項では、`sp_iqcheckdb` を使用してインデックスの矛盾を分析する方法について説明し、インデックスの問題が検出されたときの `DBCC` 出力を示し、インデックスの問題に関係する `DBCC` エラーについて説明します。

矛盾するインデックスがある場合の出力例

次は、`sp_iqcheckdb` を実行し、インデックスの矛盾が検出されたときの出力例です。`DBCC` は、検査したインデックスについての概要と詳細の両方を表示します。レポート上部の `Index Summary` セクションは、矛盾するインデックスが検出されたかどうかを示します。矛盾するインデックスの名前と問題のタイプが `Index Statistics` セクションに示されています。アスタリスク (*****) が付いた行には、矛盾するインデックスについての情報が含まれています。

レポートされるエラーのタイプで最も多いのは、`RID` が余分にあるエラー、見つからないエラー、または重複しているエラーです。これらのエラーは、インデックスがデータを正しく表しておらず、誤った結果を生じたり、その他の誤りの原因になったりする可能性があることを示しています。これらのエラーは一般に、矛盾の詳細を示す他のエラーを伴います。

この例では、`DBCC` は `HNG` インデックスの矛盾をレポートしています。対応する `FP` インデックスの検査結果は良好なので、損傷した `HNG` インデックスの修復に `FP` インデックスを `sp_iqrebuildindex` とともに使用できます。

この例で実行したコマンド・ラインは `sp_iqcheckdb 'verify database'` です。

`DBCC` は詳細なレポートを生成しますが、この例では出力の一部の行が省略されています。

```

          Stat                               Value                               Flags
=====|=====|=====
DBCC Verify Mode Report                    |                               |
=====|=====|=====
** DBCC Status                            |Errors Detected                    |*****
    DBCC Work units Dispatched            |75                                |
    DBCC Work units Completed             |75                                |
=====|=====|=====
Index Summary                              |                               |
=====|=====|=====
** Inconsistent Index Count                |1                                 |*****
    Verified Index Count                  |85                                |
=====|=====|=====
Index Statistics                           |                               |
=====|=====|=====
** Inconsistent Index                      |contact.DBA.idx01_HNG            |*****
...
    Verified Index                        |fin_data.DBA.ASIQ_IDX_T209_C3_HG |
    Verified Index                        |fin_data.DBA.ASIQ_IDX_T209_C4_FP |
...
    Verified Index                        |employee.DBA.ASIQ_IDX_T212_C19_FP |
    Verified Index                        |employee.DBA.ASIQ_IDX_T212_C20_FP |
    Verified Index                        |iq_dummy.DBA.ASIQ_IDX_T213_C1_FP |
** Extra Index RIDs                       |5                                 |*****
    FP Indexes Checked                    |68                                |
    HNG Indexes Checked                   |1                                 |
    HG Indexes Checked                    |17                                |
=====|=====|=====

```

sp_iqcheckdb によって検出された矛盾するインデックスは **contact.DBA.idx01_HNG** です。

sp_iqcheckdb を再度実行し、矛盾するインデックスだけを検査すると、次の DBCC 出力が生成されます。この例で実行したコマンド・ラインは **sp_iqcheckdb 'verify index DBA.contact.idx01_HNG'** です。

```

          Stat                               Value                               Flags
=====|=====|=====
DBCC Verify Mode Report                    |                               |
=====|=====|=====
** DBCC Status                            |Errors Detected                    |*****
    DBCC Work units Dispatched            |1                                 |
    DBCC Work units Completed             |1                                 |
=====|=====|=====
Index Summary                              |                               |
=====|=====|=====
** Inconsistent Index Count                |1                                 |*****
    Verified Index Count                  |1                                 |
=====|=====|=====

```

```

Index Statistics
=====
** Inconsistent Index      | contact.DBA.idx01_HNG      | *****
   Verified Index         | contact.DBA.ASIQ_IDX_T206_C1_FP |
** Extra Index RIDs       | 5                           | *****
   FP Indexes Checked     | 1                           |
   HNG Indexes Checked    | 1                           |
=====

```

DBCC インデックス・エラー 次の表は、インデックスの問題に関係する DBCC 出力のメッセージを示します。より詳細な DBCC メッセージのリストについては、「[DBCC エラー・メッセージ](#)」(518 ページ)の項を参照してください。

表 13-4: DBCC インデックス・エラー

DBCC メッセージ	説明／対処方法
Inconsistent Index Count	DBCC が検出したインデックスの数に矛盾がある。
Inconsistent Index	DBCC が矛盾を検出したインデックスの名前。
Extra Index RIDs Missing Index RIDs Duplicate Index RIDs	矛盾するすべてのインデックスで、矛盾しているローの合計数。
Bitmap Verify Errors	すべてのデータベース・オブジェクト内の矛盾するビットマップの合計数。
FP Lookup Table Inconsistencies	修復不可能なエラー。1 バイトまたは 2 バイトの FP が内部的に矛盾している。
Non-Completed Index Count	検査中に例外が発生したために検証できなかったインデックスの数。
Non-Completed Index	検査中に例外が発生したために検証されなかったインデックスの名前。例外が Future Version、メモリ不足、またはバッファ不足エラーの場合は、DBCC 接続をコミットして DBCC を再実行する。
VDO Incorrect First Available Fields VDO Incorrect Next Available Fields VDO Incorrect Used Count Fields VDO Incorrect In-use Bitvec VDO Incorrect In-use Bitmap VDO Incorrect Partial Bitmap VDO Incorrect Deleted Bitmaps	修復不可能なエラー。テーブル全体がアクセスできなくなる可能性がある。エラーを解決するには、矛盾するテーブルを強制削除する必要がある。
HG Missing Groups HG Extra Groups HG Extra Keys HG Missing Keys B-Tree Invalid Item Count B-Tree Invalid Item Count G-Array Empty Page Errors G-Array Bad Group Type Errors G-Array Out of Order Group Errors	High Group インデックス固有のエラー。

インデックス・エラーの修復

`sp_iqcheckdb` ストアド・プロシージャで行えるのは、インデックスに矛盾があることを示すことです。インデックスを修復するには、`sp_rebuildindex` プロシージャを使用します。その後で、`sp_iqcheckdb` を `verify` モードで実行して、矛盾の有無をチェックします。それでもインデックスが矛盾している場合は、「[矛盾するインデックス、テーブル、またはカラムの削除](#)」(516 ページ)の説明に従ってインデックスを削除および再作成した後、インデックスを再構築します。インデックスの修復の例については、『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「[第 7 章 システム・プロシージャ](#)」の「[sp_iqrebuildindex プロシージャ](#)」を参照してください。

注意 Sybase IQ には FP インデックスを修復する機能が用意されていないため、`sp_iqrebuildindex` プロシージャでは FP インデックスを修復できません。

割り付けの問題の分析

この項では、`sp_iqcheckdb` を使用して割り付けの問題を分析する方法について説明し、割り付けの問題が検出されたときの DBCC 出力を示し、割り付けの問題に関係する DBCC エラーについて説明します。

データベースは、データベース・オブジェクトが使用しているブロックを追跡するアロケーション・マップ (フリー・リスト) を保持しています。DBCC は、次の 3 つのタイプの割り付けの問題を検出します。

リーク・ブロック リーク・ブロックは、データベース・アロケーション・マップによると割り付けられているが、実際はどのデータベース・オブジェクトにも属していないブロックです。リーク・ブロックは DBCC でリカバリできます。

割り付けられていないブロック 割り付けられていないブロックは、データベース・アロケーション・マップによると割り付けられていないが、実際はデータベース・オブジェクトに使用されているブロックです。割り付けられていないブロックは DBCC でリカバリできます。

複合的に所有されたブロック 複合的に所有されたブロックは、複数のデータベース・オブジェクトに使用されているブロックです。関係する構造の少なくとも 1 つに、矛盾するデータが含まれています。DBCC では、このタイプの割り付けの問題は修復できません。このタイプのエラーが発生した場合は、ブロックを共有しているインデックスが判明するまで、インデックスのリストを指定して DBCC を再度実行します。次に、それらのインデックスをすべて削除し、複合的に所有されたブロックをなくします。矛盾するインデックスを削除する方法の詳細については、「[矛盾するインデックス、テーブル、またはカラムの削除](#)」(516 ページ)の項を参照してください。

リーク領域の出力例

`sp_iqcheckdb` を実行し、リークした領域が見つかった場合に表示される出力の例を次に示します。アスタリスク (*****) が付いた行には、割り付けの問題についての情報が含まれています。この例では、DBCC は 16 のリーク・ブロックをレポートしています。

この例で実行したコマンド・ラインは `sp_iqcheckdb 'allocation database'` です。

```

          Stat                               Value                               Flags
=====|=====|=====
DBCC Allocation Mode Report | | |
=====|=====|=====
** DBCC Status                |Errors Detected                |*****
  DBCC Work units Dispatched |164                             |
  DBCC Work units Completed  |164                             |
=====|=====|=====
Allocation Summary           | | |
=====|=====|=====
  Blocks Total                |8192                            |
  Blocks in Current Version   |4785                            |
  Blocks in All Versions      |4785                            |
  Blocks in Use               |4801                            |
  % Blocks in Use             |58                              |
** Blocks Leaked              |16                              |*****
=====|=====|=====
Allocation Statistics       | | |
=====|=====|=====
  ...
** 1st Unowned PBN           |1994                            |*****
  ...
=====|=====|=====

```

オフラインになっている DB 領域が 1 つ以上ある場合に、特定の DB 領域に割り付けの問題があるかどうかを調べるには、次の構文を使用します。

```
sp_iqcheckdb 'allocation dbspace dbspace-name'
```

DBCC 割り付けエラー

`sp_iqcheckdb` を `allocation` モードまたは検証モードで実行すると、DBCC が生成する出力で割り付けの問題がレポートされます。Allocation Summary セクションに、“** Blocks Leaked” や “** Blocks with Multiple Owners” など、アスタリスクでフラグが付けられた値がある場合は、割り付けの問題が発生しています。

次の表は、割り付けの問題に関する DBCC 出力のメッセージを示します。より詳細な DBCC メッセージのリストについては、「[DBCC エラー・メッセージ \(518 ページ\)](#)」の項を参照してください。

表 13-5: DBCC 割り付けエラー

DBCC メッセージ	説明／対処方法
Block Count Mismatch	この数は常に他の割り付けエラーを伴います。
Blocks Leaked 1st Unowned PBN	どのデータベース・オブジェクトにも使用されていないことが判明したブロック。エラーの修復には、 <code>sp_iqcheckdb dropleaks</code> モードを使用してください。
Blocks with Multiple Owners 1st Multiple Owner PBN	複数のデータベース・オブジェクトに使用されているブロック。矛盾しているとレポートされたオブジェクトを削除します。
Unallocated Blocks in Use 1st Unallocated PBN	データベース・オブジェクトに使用されているが、使用中のマークが付いていないブロック。エラーの修復には、 <code>sp_iqcheckdb dropleaks</code> モードを使用してください。

Allocation Summary の行が問題を示していない場合でも、Index Summary セクションの “Inconsistent Index Count” で値がレポートされている場合は、1 つまたは複数のインデックスが矛盾しています。インデックスの修復については、「[インデックス・エラーの修復](#)」(505 ページ) の項を参照してください。

割り付けの問題の修復

次の手順では、`sp_iqcheckdb dropleaks` を使用してデータベースの割り付けの問題を修復します。

注意 次の手順では、`-gd` スイッチと `-gm` スイッチを使用してデータベース・アクセスを制限します。さらに厳密な方法については、「[リカバリ中のデータベース・アクセスの制限](#)」(510 ページ) を参照してください。

❖ DBCC を使用して割り付けの問題を修復する

- 1 サーバを起動します。例を示します。

```
start_iq -n my_db_server -x 'tcpip(port=7934)'
-gd dba -gm 1 /work/database/my_db.db
```

注意 “.DB” ではなく “.db” 拡張子でデータベースを起動する必要があります。

2 つのサーバ起動スイッチを使用してアクセスを制限することをおすすめします。

- `-gd DBA` を使用して、DBA 権限のあるユーザだけがデータベースを起動および停止できるようにします (データベースを起動または停止するには、クライアントはすでにサーバに接続している必要があるため、このスイッチは接続を妨げません)。
- `-gm 1` を使用して、1 つの接続に加えて制限より 1 つ多い DBA 接続を許可し、非常時に DBA が接続して他の接続を停止できるようにします。

接続の制限の詳細については、「[リカバリ中のデータベース・アクセスの制限](#)」(510 ページ) を参照してください。

- 2 ストアド・プロシージャ `sp_iqcheckdb` を `dropleaks` モードで実行します。

```
sp_iqcheckdb 'dropleaks database'
```

オフラインになっている DB 領域が 1 つ以上ある場合は、次のコマンドを実行することで、特定の DB 領域に関連した割り付けの問題を修復できます。

```
sp_iqcheckdb 'dropleaks dbspace dbspace-name'
```

割り付けの修復が成功すると、`sp_iqcheckdb` は “Freelist Updated” というメッセージを表示します。エラーが検出された場合、`sp_iqcheckdb` は “Freelist Not Updated” および “Errors Detected” というメッセージを返します。

- 3 `sp_iqcheckdb` の終了後にサーバを停止します。サーバを停止するには、プラットフォームの種類を問わず `stop_iq` を使用するか、Windows ではコンソール・ウィンドウの [シャットダウン] ボタンを使用します。

割り付けの問題が修復された後は、アロケーション統計が DBCC 出力にエラーなしで表示されます。

DBCC 割り付け修復の出力

DBCC のレポート上部の Allocation Summary セクションには、割り付けの使用状況についての情報が表示されます。Allocation Statistics セクションには、ブロックについてのより詳細な情報が表示されます。DBCC 出力には、リカバリされたリーク・ブロックの修復メッセージは含まれません。

次に例を示します。

```
sp_iqcheckdb 'dropleaks';
checkpoint;
```

`sp_iqcheckdb` の出力はエラーを示していないので、`checkpoint` が実行されます。

この出力例では、DBCC がレポートする一部の統計が省略されています。

Stat	Value	Flags
DBCC Allocation Mode Report		
DBCC Status	Freelist Updated	
DBCC Status	No Errors Detected	
DBCC Work units Dispatched	75	
DBCC Work units Completed	75	


```

Allocation Summary
=====
  Blocks Total          | 8192
  Blocks in Current Version | 4594
  Blocks in All Versions | 4594
  Blocks in Use        | 4610
  % Blocks in Use      | 56
=====
Allocation Statistics
=====
  DB Extent Count      | 1
  Marked Logical Blocks | 8176
  Marked Physical Blocks | 4594
  Marked Pages         | 511
  Blocks in Freelist   | 126177
  Imaginary Blocks     | 121567
  Highest PBN in Use   | 5425
  Total Free Blocks     | 3582
  Usable Free Blocks    | 3507
  % Free Space Fragmented | 2
  Max Blocks Per Page  | 16
  1 Block Page Count   | 103
  3 Block Page Count   | 153
  ...
  16 Block Hole Count  | 213
=====

```

リカバリ後のデータベースの起動

強制リカバリまたはリーク・ブロック・リカバリを実行するときは、“DB”ではなく“db”拡張子でデータベースを起動する必要があります。例を示します。

```

start_iq -n my_db_server -x 'tcpip{port=7934}'
-gd dba my_db /work/database/my_db.db

```

強制リカバリ・モード

強制データベース・リカバリと通常のデータベース・リカバリでは、次の点が異なります。

- 強制リカバリでは、データベース内のすべての記憶領域に使用中のマークが付けられます。** 矛盾している可能性のあるアロケーション・マップをリカバリするために、データベース内のすべての記憶領域に使用中のマークが付けられます。sp_iqcheckdb を dropleaks モードで使用して、アロケーション・マップを正しい状態にリセットします。

- **インクリメンタル・バックアップは無効になります。** 強制リカバリ・モードでデータベースを開いた後は、インクリメンタル・バックアップが無効になります。次のバックアップでは、フル・バックアップを行う必要があります。フル・バックアップを行うと、インクリメンタル・バックアップが再び有効になります。
- **強制リカバリ・パラメータは、サーバが動作している間に開くすべてのデータベースに適用されます。** したがって、データベースを開いた後で、DBA はサーバをいったん停止し、強制リカバリ・フラグを指定せずにサーバを再起動して、それ以降は通常モードでデータベースが開くようにします。強制リカバリを有効にしたままデータベースを繰り返し開いてもデータベースに害はありませんが、DBA が混乱する可能性があります。データベースを強制リカバリ・モードで開くたびに、データベース内のすべての記憶領域に使用中のマークが付けられます。

これ以降の項では、以下の手順について説明します。

- 「強制リカバリ・モードでのサーバの起動」
- 「リーク領域のリカバリ」
- 「マルチプレックス・データベースのリカバリ」

強制リカバリ・モードでのサーバの起動

アロケーション・マップが矛盾している場合は、強制リカバリによってサーバを起動できます。このモードでは、オプションは矛盾に関する情報を表示します。このような矛盾を修復するオプションを指定することもできます。

この手順に従うのは、障害リカバリ中に `s_buf` またはフリー・リストのエラーが発生した場合だけです。SQL Anywhere のリカバリ・エラーが発生した場合は、`-iqfrec` で問題は解決できません。

リカバリ中のデータベース・アクセスの制限

DBA は、アクセスを制限することで、強制リカバリ中にデータベースが不用意に開かれないように、より詳細な制御を行うことができます。2つのサーバ起動スイッチを使用してアクセスを制限することをおすすめします。

- `-gd DBA` を使用して、DBA 権限のあるユーザだけが、実行中のサーバでデータベースを起動および停止できるようにします(データベースを起動または停止するにはクライアントはすでにサーバに接続している必要があるため、このスイッチによって接続が妨げられることはありません)。
- `-gm 1` を使用して、1つの接続に加えて制限より1つ多い DBA 接続を許可し、非常時に DBA が接続して他の接続を停止できるようにします。

接続を制限するための別の方法としては、強制リカバリを実行しようとしているサーバの接続を開始した直後に、次のように指定します。

```
sa_server_option('disable_connections', 'ON')
```

さらに、リカバリ後に同じ接続に対して次のように指定します。

```
sa_server_option('disable_connections', 'OFF')
```

この方法の欠点は、別の DBA 接続からの緊急アクセスが不可能なことです。

❖ 強制リカバリ・モードでサーバを起動する

- 1 `start_iq` コマンドで `-iqfreq` サーバ起動オプションを使用して、サーバを強制リカバリ・モードで起動し、すべてのページに使用中のマークを付けます。例を示します。

```
start_iq -n my_server -x 'tcpip(port=7934}'  
-gd dba -gm 1 -iqfreq my_db /database/my_db.db
```

強制リカバリがシングルノード・モードでサーバを起動します。最初にすべてのセカンダリ・サーバを停止します。

警告！ 障害が発生した後でマルチプレックス書き込みサーバを起動するときは、上書きの起動スイッチ (`-iqmpx_ov 1`) を指定し、シングルノード・モード (`-iqmpx_sn 1`) で起動します。リカバリにマルチプレックス・モード (デフォルト) は使用しないでください。

データベース名は 2 回指定します。1 つは強制リカバリにかけるデータベースを指定します。もう 1 つは起動するデータベースを指定します。`-iqfreq` サーバ起動オプションには、データベース名を指定する必要があります。これは物理データベース名で、大文字と小文字が区別されることに注意してください。データベース名の確認に `select_dbname` を使用しないでください。このコマンドは `-n` 起動オプションで割り当てられた論理名を返します。

- 2 必要であれば、`sp_iqcheckdb` を実行してリークしたブロックを検査できます。詳細については、「[割り付けの問題の分析](#)」(505 ページ) を参照してください。
- 3 サーバが正常に起動したら、サーバを停止します。サーバを停止するには、UNIX では `stop_iq` を使用し、Windows ではコンソール・ウィンドウの [シャットダウン] ボタンを使用します。
- 4 `-iqfreq` オプションを指定せずに、通常の方法でサーバを再起動します。

強制リカバリ・モードでサーバを起動できない場合は、Sybase 製品の保守契約を結んでいるサポート・センタにお問い合わせください。

sp_iqcheckdb を実行しない強制リカバリの使用

強制リカバリを実行すると、データベースは有効な、ただし完全に割り付けられたモードで起動します。言い換えると、すべての操作を実行できますが、永久メイン dbspace は残りません。何か他の操作を行う前に、sp_iqcheckdb を dropleaks モードで実行して失われた DB 領域をリカバリするか、新しい DB 領域を追加する必要があります。クエリには追加の永久 DB 領域が必要ないので、クエリも正常に実行できます。ただし、データのロード、挿入、削除はできません。

警告！ データベースを検証せずにクエリを実行しても、データに矛盾が生じることはありません。ただし、サーバの障害の原因となったデータに問題がある場合は、サーバで再び障害が起きるか、誤った結果が生じる可能性があります。

sp_iqcheckdb を使用して、失われた領域やリーク領域を再利用する方法の詳細については、「[リーク領域のリカバリ](#)」の項を参照してください。

リーク領域のリカバリ

アロケーション・マップは、ページが IQ 内で使用中かそうでないかをサーバが判断するために使われます。システム障害が起きた場合や、データベースを強制リカバリで開いた場合には、データベースのアロケーション・マップが実際の使用状況の割り付けを反映していないことがあります。このような場合、「データベースに「リークした」記憶領域または「リーク・ブロック」がある」といいます。一般に、少数のリーク・ブロックであれば気にする必要はありません。リーク・ブロックが何メガバイト分もある場合は、その領域をリカバリすることを検討してください。

sp_iqcheckdb スタアド・プロシージャを dropleaks モードで使用すると、指定したデータベース内でリークした記憶領域をリカバリできます。

リークした記憶領域をリカバリしているときは、アロケーション・マップを変更する他のトランザクションは遮断されます。たとえば、データベースを変更するチェックポイントやコマンドなどの操作があります。

リークした記憶領域のリカバリと強制リカバリは、同時に実行することも別々に実行することもできます。強制リカバリを行わずにデータベース内のリーク領域をリカバリする場合は、「[割り付けの問題の修復](#)」(507 ページ) の項の手順に従います。強制リカバリを行った後でデータベース内のリーク領域をリカバリする場合は、次の「[強制リカバリを使用したリーク領域のリカバリ](#)」の項の手順に従います。

強制リカバリを使用した
リーク領域のリカバリ

「[割り付けの問題の修復](#)」(507 ページ) の手順に従ってもリークした記憶領域をリカバリできなかった場合は、次の手順で行います。

注意 次の手順では、`-gd` スイッチと `-gm` スイッチを使用してデータベース・アクセスを制限します。さらに厳密な方法については、「[リカバリ中のデータベース・アクセスの制限](#)」(510 ページ) を参照してください。

❖ 強制リカバリを使用してリーク領域をリカバリする

- 1 `start_iq` コマンドで `-iqfrec` オプションを指定してサーバを起動します。例を示します。

```
start_iq -n my_db_server -x 'tcpip(port=7934)'
-gd dba -gm 1
-iqfrec my_db /work/database/my_db.db
```

データベース名は 2 回続けて指定します。1 つは起動するデータベースを指定します。もう 1 つは強制リカバリにかけるデータベースを指定します。`-iqfrec` オプションには、データベース名を指定する必要があります。

- 2 リカバリするデータベースに接続します。
- 3 ストアド・プロシージャ `sp_iqcheckdb` を `dropleaks` モードで実行します。

```
sp_iqcheckdb 'dropleaks database'
```

エラーがなく、`sp_iqcheckdb` が “Freelist Updated” というメッセージを表示した場合は、リーク領域のリカバリと強制リカバリが完了しています。次の手順に進みます。

矛盾が検出された場合は、「[矛盾するインデックス、テーブル、またはカラムの削除](#)」の項の手順に従って、矛盾するオブジェクトを削除します。次に、`sp_iqcheckdb` を再度実行してリーク領域をリカバリします。

- 4 チェックポイントを発行します。
- 5 通常の方法でサーバを停止します。
- 6 通常の方法でサーバを再起動し、通常の処理を再開します。

マルチプレックス・データベースのリカバリ

マルチプレックス・データベースに関するリカバリの問題のトラブルシューティングを行う前に、『[Sybase IQ Multiplex の使用](#)』を参照してください。

トランザクション・ログを使わない緊急リカバリ

通常は、この章のこれまでの項で説明してきたリカバリ手順に従ってください。

まれに、次のすべての条件が揃った場合は、緊急リカバリ手順を使用することがあります。

- バックアップが存在しない
 - トランザクション・ログが紛失している
 - ミラー・ログがない
 - Sybase Engineering と協力して代替オプションを策定する十分な時間がない
- これらの条件のもとでは、`-f` オプションを指定してサーバを再起動できます。

`-f` リカバリ・オプション

機能 トランザクション・ログが失われた場合に、データベース・サーバを強制的に起動します。

Syntax `start_iq -n server-name [other-server-options] -f`

説明 データベースと同じディレクトリにトランザクション・ログがある場合、データベース・サーバはカタログのチェックポイント・リカバリを実行し、トランザクション・ログを使用してリカバリを実行した後に終了します。つまり、サーバはここで動作を停止します。その後は、`-f` オプションを指定せずにデータベース・サーバを再起動し、通常の作業を再開できます。

トランザクション・ログがない場合、データベース・サーバはデータベースのチェックポイント・リカバリを実行した後に終了します。つまり、サーバはここで動作を停止します。その後は、`-f` オプションを指定せずにデータベース・サーバを再起動し、通常の作業を再開できます。

警告！ `-f` オプションを使用すると、普通はサーバを復旧できますが、トランザクションのリプレイをバイパスするため、データベースを破損することも少なくありません。こうした破損は後になるまで気付かれなことがあり、通常は修復できません。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。強制リカバリ (`-iqfreq`) を行ってデータベースを開き直すことも必要になる場合があります。

例

```
start_iq -n bad_server -x 'tcpip(port=7934)'  
-gd dba -gm 1 -f
```

DBCC によってレポートされた問題の処理

次の表は、問題があることを表す DBCC 出力のメッセージを示します。より詳細な DBCC メッセージのリストについては、「[DBCC エラー・メッセージ](#)」(518 ページ)の項を参照してください。

表 13-6: DBCC が修復できない問題を示すメッセージ

DBCC メッセージ	説明/対処方法
FP Lookup Table Inconsistencies	修復不可能なエラー。1 バイトまたは 2 バイトの FP が内部的に矛盾している。
VDO Incorrect First Available Fields	修復不可能なエラー。テーブル全体がアクセスできなくなる可能性がある。エラーを解決するには、矛盾するテーブルを強制削除する必要がある。
VDO Incorrect Next Available Fields	
VDO Incorrect Used Count Fields	
VDO Incorrect In-use Bitvec	
VDO Incorrect In-use Bitmap	
VDO Incorrect Partial Bitmap	
VDO Incorrect Deleted Bitmaps	
Blocks with Multiple Owners 1st Multiple Owner PBN	複数のデータベース・オブジェクトに使用されているブロック。矛盾しているとレポートされたオブジェクトを削除する。
DBCC Meta-data Errors Blockmap Invalid Chunksize Error Count Blockmap Compression Bit Error Count Blockmap Invalid Block Number Error Count	内部ページ・マッピング構造が矛盾しており、オブジェクトを削除する必要がある。
DBCC Inconsistent Disk Block Headers DBCC Decompress Errors	オブジェクトの記憶領域が矛盾しており、オブジェクトを削除する必要がある。

これらの修復不可能な問題を解決する方法については、これ以降の項を参照してください。

修復できないインデックスの問題

DBCC がインデックスの問題を検出すると、そのインデックスの名前が問題の種類とともにレポートされます。非 FP インデックスの修復には、`sp_iqrebuildindex` を使用します。FP インデックスは修復できません。`sp_iqcheckdb` をデフォルトまたは `check` モードで実行し、インデックスが “Inconsistent Index” とレポートされた場合に従う手順については、「[インデックス・エラーの分析](#)」(502 ページ)の項を参照してください。

問題の種類に応じて、`DROP INDEX`、`ALTER TABLE DROP COLUMN`、`DROP TABLE`、または `FORCE_DROP` オプションを使用して問題を解決します。「[矛盾するインデックス、テーブル、またはカラムの削除](#)」(516 ページ)の項を参照してください。

矛盾するインデックスやテーブルを修復するための最良の方法を判断するときは、Sybase 製品の保守契約を結んでいるサポート・センタに相談することをおすすめします。

矛盾するインデックス、テーブル、またはカラムの削除

sp_iqcheckdb が修復不可能なインデックス、カラム、またはテーブルをレポートした場合は、対応する DROP INDEX、ALTER TABLE DROP COLUMN、または DROP TABLE 文を使用して、これらのオブジェクトを削除します。

注意 Sybase 製品の保守契約を結んでいるサポート・センタからの指示がないかぎり、オブジェクトを強制削除しないでください。

矛盾するオブジェクトを削除できない場合は、FORCE_DROP テンポラリ・オプションを設定します。FORCE_DROP オプションを設定すると、IQ サーバは削除したオブジェクトのディスク上の記憶領域を再利用する代わりに、暗黙的にリークします。このリーク領域は、後から DBCC を使用してリカバリできます。オブジェクトの記憶領域に関する唯一の情報はオブジェクト自体の中にあり、この情報がオブジェクトの矛盾を招いている可能性があるため、このリークは矛盾するオブジェクトにとって必要なことです。

FORCE_DROP データベース・オプションは、セカンダリ・ノードでは使用できません。セカンダリ・ノードで強制削除が試行されると、エラーが返されます。FORCE_DROP はテンポラリ・オプションであるため、このオプションの値が同期時にセカンダリ・ノードに送信されることはありません。

注意 オブジェクトを強制削除するときは、DBA 以外にデータベースに接続しているユーザがないことを確認してください。強制削除の直後にサーバを再起動します。

次の手順では、-gd スイッチと -gm スイッチを使用してデータベース・アクセスを制限します。-gd スイッチは、稼働中のサーバのデータベースを起動または停止できるユーザだけを制限します。さらに厳密な方法については、「[リカバリ中のデータベース・アクセスの制限](#)」(510 ページ)を参照してください。

❖ 矛盾するオブジェクトを削除する

- 1 サーバを再起動します。

```
start_iq -n bad_db_server -x 'tcpip{port=7934}'  
-gm 1 -gd dba bad_db.db
```

オブジェクトを強制削除するときは、他のユーザに接続を許可しないようにします。

2 つのサーバ起動スイッチを使用してアクセスを制限することをおすすめします。

- `-gd DBA` を使用して、DBA 権限のあるユーザだけがデータベースを起動および停止できるようにします (データベースを起動または停止するには、クライアントはすでにサーバに接続している必要があるため、このスイッチは接続を妨げません)。
- `-gm 1` を使用して、1 つの接続に加えて制限より 1 つ多い DBA 接続を許可し、非常時に DBA が接続して他の接続を停止できるようにします。

接続の制限の詳細については、インストールおよび設定ガイドを参照してください。

- 2 テンポラリ・オプション `FORCE_DROP` を ON に設定します。

```
set temporary option FORCE_DROP = 'ON'
```

- 3 矛盾するすべてのオブジェクトを削除します。

必要に応じて、`DROP INDEX`、`ALTER TABLE DROP COLUMN`、`DROP TABLE` の各コマンドを使用します。サーバを再起動するまでは、他の `DDL` コマンドや `DML` コマンドを入力しないでください。

- 4 サーバを再起動します。

リーク領域をリカバリし、アロケーション・マップを正常な状態に更新するには、サーバを起動します。

```
start_iq -n bad_db_server -x 'tcpip{port=7934}'  
-gm 1 -gd dba bad_db.db
```

- 5 `sp_iqcheckdb` を実行します。

```
sp_iqcheckdb 'dropleaks database';
```

この手順により、データベース・アロケーション・マップが計算されたアロケーション・マップにリセットされます。

詳細については、「[リーク領域のリカバリ](#)」(512 ページ) および「[データベースの検証](#)」(494 ページ) の項を参照してください。

DBCC エラー・メッセージ

次の表は、DBCC 出力で最も重要なメッセージを示します。

表 13-7: DBCC エラー・メッセージ

DBCC メッセージ	説明／対処方法
Inconsistent Index Count	DBCC が検出したインデックスの数に矛盾があります。
Inconsistent Index	DBCC が矛盾を検出したインデックスの名前。
Extra Index RIDs Missing Index RIDs Duplicate Index RIDs	矛盾するすべてのインデックスで、矛盾しているローの合計数。
Bitmap Verify Errors	すべてのデータベース・オブジェクト内の矛盾するビットマップの合計数。
FP Lookup Table Inconsistencies	修復不可能なエラー。1 バイトまたは 2 バイトの FP が内部的に矛盾しています。
Non-Completed Index Count	検査中に例外が発生したために検証できなかったインデックスの数。
Non-Completed Index	検査中に例外が発生したために検証されなかったインデックスの名前。例外が Future Version、メモリ不足、またはバッファ不足エラーの場合は、DBCC 接続をコミットして DBCC を再実行してください。
HG Missing Groups HG Extra Groups HG Extra Keys HG Missing Keys B-Tree Invalid Item Count B-Tree Invalid Item Count G-Array Empty Page Errors G-Array Bad Group Type Errors G-Array Out of Order Group Errors	High Group インデックス固有のエラー。
VDO Incorrect First Available Fields VDO Incorrect Next Available Fields VDO Incorrect Used Count Fields VDO Incorrect In-use Bitvec VDO Incorrect In-use Bitmap VDO Incorrect Partial Bitmap VDO Incorrect Deleted Bitmaps	修復不可能なエラー。テーブル全体がアクセスできなくなる可能性がある。エラーを解決するには、矛盾するテーブルを強制削除する必要があります。
Block Count Mismatch	この数は他の割り付けエラーを伴います。
Blocks Leaked 1st Unowned PBN	どのデータベース・オブジェクトにも使用されていないことが判明したブロック。エラーの修復には、dropleaks モードを使用してください。
Blocks with Multiple Owners 1st Multiple Owner PBN	複数のデータベース・オブジェクトに使用されているブロック。矛盾しているとレポートされたオブジェクトを削除してください。
Unallocated Blocks in Use 1st Unallocated PBN	データベース・オブジェクトに使用されているが、使用中のマークが付いていないブロック。エラーの修復には、dropleaks モードを使用してください。

DBCC メッセージ	説明／対処方法
Freelist Updated	割り付けの修復が成功したことを示します。
Freelist Not Updated	割り付けの修復時にエラーが検出され、割り付けの修復が成功しなかったことを示します。
Invalid Blockmap Unique ID Generator Blockmap Unique ID Generator Updated Invalid Transaction ID Counter Transaction ID Generator Updated	DBCC <code>resetclocks</code> オプションに固有のエラーと修復メッセージ。
DBCC Future Version Errors	DDL が実行されているために、DBCC がテーブルを開くことができなかった。DBCC 接続をコミットし、DBCC を再実行してください。
DBCC Locked Table Access Conflict	DBCC が、別の接続によってロックされているテーブルを開こうとした。DBCC 処理を確実に完了するために、他のユーザがデータベース内のテーブルをロックしていないことを確認してください。
DBCC Out of Buffers Errors	IQ メイン・キャッシュのサイズが小さすぎる。メイン・キャッシュ・サイズを増やすか、個別のオブジェクトに対して DBCC を実行してください。
DBCC Out of Memory Errors	システム・メモリが不足しているために DBCC 操作を完了できない。
DBCC Meta-data Errors Blockmap Invalid Chunksize Error Count Blockmap Compression Bit Error Count Blockmap Invalid Block Number Error Count	内部ページ・マッピング構造が矛盾しており、オブジェクトを削除する必要があります。
DBCC Page Read Errors	オブジェクトを読み込もうとしている最中に I/O エラーが発生した。ハードウェアの診断を実行します。
DBCC Inconsistent Disk Block Headers DBCC Decompress Errors	オブジェクトの記憶領域が矛盾しており、オブジェクトを削除する必要があります。
DBCC Unknown Exceptions	DBCC に認識されないタイプの例外が発生した。IQ メッセージ・ファイルで詳細を確認してください。
Unowned LVC cells Duplicate LVC cell rows Unallocated LVC cell rows	<p>VARCHAR カラムまたは CLOB カラムに矛盾があることを示すメッセージ。未所有の LVC セルは、小容量の使用不可ディスク領域を表し、無視しても問題ありません。重複した LVC セルと未割り付けの LVC セルは、ダメージを受けたカラムを削除しないと解決できない重大なエラーです。</p> <p>ダメージを受けたカラムを削除するには、古いカラムのコピーから新しいカラムを作成した後で、元のカラムを削除し、新しいカラムの名前を古いカラムの名前に変更します。</p> <p>LVC は、幅が 255 より大きい VARCHAR カラムです。LVC は CLOB によっても使用されます。</p>

この章について

この章では、Sybase IQ の実行中にときおり発生する可能性があるさまざまな問題を解決するための推奨事項について説明します。

内容

トピック名	ページ
状況別の解決策	521
ネットワーク通信のトラブルシューティング	552
診断ツール	556
サポート・センタへの問題の報告	567
チェックリスト：サポート・センタに提出する情報	572

Sybase IQ マルチプレックス・サーバに特に関係する問題の解決方法については、『Sybase IQ Multiplex の使用』を参照してください。

ここで説明する方法を使用しても問題を解決できない場合は、Sybase オンライン・サポート Web サイトの MySybase で有益な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 ([Technical Support] を開く前に収集する情報のリストについては、「サポート・センタへの問題の報告」(567 ページ) の項を参照してください)。

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 MySybase を指定します。その他の有益な 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。
(<http://www.sybase.com/mysybase>)

状況別の解決策

この項では、発生する状況の種類、問題を診断するための詳細な情報の取得方法、問題を解決するための対処法について説明します。この項で説明する問題は、次のカテゴリに分類されています。

- 「サーバのリカバリとデータベースの修復」
- 「サーバの運用上の問題」
- 「データベース接続の問題」

- [「Interactive SQL \(dbisql\) の問題」](#)
- [「リソースの問題」](#)
- [「処理の問題」](#)
- [「パフォーマンスの問題」](#)
- [「Sybase Central の問題」](#)

これ以降の項で説明するものを含めて、さまざまな状況を診断するときに役立つ情報の取得方法については、[「診断ツール」\(556 ページ\)](#)の項を参照してください。

サーバのリカバリとデータベースの修復

サーバまたはデータベースを起動できない場合、データベースは起動するが接続できない場合、またはデータベースの検証中に問題が検出された場合は、この項の説明を参照し、問題を解決するための対処法を判断してください。

❖ 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

1 サーバは起動するか？

サーバが起動する場合は手順 2 に進みます。

サーバが起動しない場合は、[「サーバの運用上の問題」\(523 ページ\)](#)の項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」\(510 ページ\)](#)の項を参照し、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。データベースをバックアップからリストアすることが必要になる可能性があります。

2 データベースに接続できるか？

データベースに接続できない場合は、[「データベース接続の問題」\(533 ページ\)](#)の項で、トラブルシューティングの推奨事項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」\(505 ページ\)](#)の項で、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」\(494 ページ\)](#)の項で、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](#)」(494 ページ) の項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](#)」(505 ページ) 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「[割り付けの問題の修復](#)」(507 ページ) を参照してください。

サーバの運用上の問題

この項では、起動、停止、無応答、異常終了など、サーバの運用上の問題について説明します。

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。

- `$IQDIR15/logfiles/<servername>.nnnn.stderr`
 - `$IQDIR15/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
- 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。
 - サーバがトランザクション・ログを見つけれられない。
 - オペレーティング・システムが適切なパッチ・レベルになっていない。
 - ネットワーク接続が機能していない。
 - サーバ名がネットワーク上で重複している。
 - サーバのポート番号がマシン上で重複している。

- サーバがすでに Windows サービスとして実行されている (Windows システムのみ)。
- メモリが不足している。
- 環境変数が正しく設定されていない。
- `start_iq` を実行できない。

対処法

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`)、および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](#)」(514 ページ) で説明されている緊急リカバリの方法を使用します。

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・ファイルに次の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、「[トランザクション・ログを使わない緊急リカバリ](#)」(514 ページ) で説明している方法でサーバを起動できる可能性があります。必要に応じて、Sybase 製品の保守契約を結んでいるサポート・センタにお問い合わせください。

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 サーバ名がネットワーク上で重複している場合、つまり複数のシステムで同じ名前のサーバが動作している場合は、`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、`iqsrv15` コマンドを使用してサーバを起動してみます。構文と前提条件については、「[start_iq を実行できない](#)」(526 ページ) を参照してください。`iqsrv15` 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。

サーバのポート番号がマシン上で重複している 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 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` コマンドを実行し、マシンですでに実行されている 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、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
```

Windows でメモリが不足している 32 ビット Windows システムで Sybase IQ サーバが起動しない場合は、使用中の Windows サーバのバージョンに応じて、Microsoft 4GT RAM チューニング機能が有効になっているかどうかを確認します。4GT オプションはブート時に、Windows オペレーティング・システムで最大 3GB の動的メモリをユーザ・プロセスに割り付けられるように設定します。サポートされる Windows プラットフォームのリストと、4GT 機能を有効にする方法の詳細については、『インストールおよび設定ガイド Windows 版』の「Sybase IQ のインストール」の章の「システムの稼働条件」を参照してください。

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。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「[第 1 章 ファイル・ロケーションとインストール設定](#)」の「[環境変数の設定](#)」を参照してください。

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` コマンド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。「[データベース接続の問題](#)」(533 ページ) の項を参照してください。

それでも `start_iq` コマンドを実行できない場合は、Sybase Central の [ツール] - [Sybase IQ 15] - [サーバの起動] を使用するか、`iqsrv15` コマンドを使用します。

注意 `iqsrv15` 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。Sybase IQ サーバを起動するときには、必ず `start_iq` を使用してください。

iqsrv15 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 unlimited 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。適切なオプションと、設定ファイルでオプションを設定する方法の詳細については、『インストールおよび設定ガイド』を参照してください。
- iqsrv15 を呼び出す前に、環境にパス \$IQDIR/lib64/ocs 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 "${LIBPATH}:${IQDIR15}/lib64/ocs"
```

他の UNIX/LINUX プラットフォームの場合：

```
% setenv LD_LIBRARY_PATH
"${LD_LIBRARY_PATH}:${IQDIR15}/lib64/ocs"
```

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、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ディレクトリを既知のディレクトリに変更する必要があります。

iqsrv15 の構文は次のとおりです。

```
iqsrv15 -n server-name -gm number
[ other-server-switches ] [ database-file [ database-switches ] ]
```

注意 `iqsrv15` コマンド・ラインでは、最後に指定するオプションが優先します。したがって、設定ファイルの値に代わる値を使用する場合は、変更するオプションを設定ファイル名の後ろに示します。例を示します。

```
iqsrv15 @iqdemo.cfg -x 'tcpip(port=1870)' iqdemo
```

この例では、`-x` パラメータが `iqdemo.cfg` ファイルの接続情報を上書きします。

`iqsrv15` 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 filename` 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。

`iqsrv15` コマンドを実行してもサーバが起動しない場合は、`iqsrv15` 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。例を示します。

```
iqsrv15 -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。

参照

サーバの起動の詳細については、「[第 2 章 Sybase IQ の実行](#)」と「[第 3 章 Sybase IQ の接続](#)」、および「[起動のトラブルシューティング、シャットダウン、接続](#)」(103 ページ)を参照してください。

Sybase IQ が処理または応答を停止する

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 2 つです。これらは Sybase IQ メッセージ・ファイルを調べるとわかります。

- ディスク領域の不足。対処法については、「[ディスク領域の不足](#)」(537 ページ)の項を参照してください。
- 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。『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」の「[バッファ・キャッシュの管理](#)」を参照してください。

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 Sybase IQ データベース・オプション `QUERY_PLAN='ON'` を使用して、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。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。

- Sybase IQ メッセージ・ファイルとサーバ・ログ・ファイルに「領域が不足しています ...」というメッセージが記録されてい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。領域不足の問題を解決する方法の詳細については、「[ディスク領域の不足](#)」(537 ページ) の項を参照してください。

データベース・オプション `MAIN_RESERVED_DBSPACE_MB` および `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。詳細については、「[IQ メイン・ストアと IQ テンポラリ・ストアの領域管理](#)」(178 ページ) を参照してください。

- 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、Sybase IQ サーバ・プロセス (iqsrv15) 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、Sybase IQ サーバ・プロセスは正常に処理を実行しています。

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。この情報を取得し、サーバ要求をログに記録する方法の詳細については、「[現在実行中の文の検索](#)」(563 ページ) および「[サーバ要求のロギング](#)」(563 ページ) の項を参照してください。

- 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 `dbisql` を使用して、表 14-1 に示す情報を指定の順序で収集します。

表 14-1: 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
<code>SELECT db_name()</code>	データベース名
<code>CHECKPOINT</code>	チェックポイントを実行できる
<code>sa_conn_properties ># sa_conn_properties.out</code>	コネクション情報
<code>sa_conn_info ># sa_conn_info.out</code>	コネクション情報
<code>sa_db_properties ># sa_db_properties.out</code>	データベースのプロパティ情報
<code>sa_eng_properties ># sa_eng_properties.out</code>	サーバのプロパティ情報
<code>sp_iqstatus ># sp_iqstatus.out</code>	データベースのステータス情報
<code>sp_iqconnection ># sp_iqconnection.out</code>	コネクション情報
<code>sp_iqtransaction ># sp_iqtransaction.out</code>	トランザクション情報

この問題を解決できない場合は、Sybase 製品の保守契約を結んでいるサポート・センタに問い合わせてください。サポート・センタ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。詳細については、「[サポート・センタへの問題の報告](#)」(567 ページ) を参照してください。

- 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 *\$IQDIR15/logfiles* ディレクトリ (Windows 32 および 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* フォルダ) に作成することで、Sybase IQ スレッドごとにスタック・トレースを生成できます。

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 Sybase IQ を起動すると、*IQDIR15* 変数が自動的に設定されます。*ASDIR* 変数が設定されていない場合は、*iqsrv15* を起動したディレクトリに *DumpAllThreads* ファイルを作成します。

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、Sybase 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ときに役立ちます。詳細については、「[サポート・センタへの問題の報告](#)」(567 ページ) を参照してください。

- 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。

a データベースに接続するか、既存の接続を使用します。

b `CREATE TABLE #dummy_monitor(c1 INT);`

c `IQ UTILITIES MAIN INTO #dummy_monitor START MONITOR '-append -debug -interval 10 -file_suffix iqdbgmon';`

d `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。

e `IQ UTILITIES MAIN INTO #dummy_monitor STOP MONITOR;`

f `IQ UTILITIES PRIVATE INTO #dummy_monitor STOP MONITOR;`

バッファ・キャッシュのモニタの詳細については、『パフォーマンス & チューニング・ガイド』の「[第 5 章 パフォーマンスのモニタリングとチューニング](#)」の「[バッファ・キャッシュのモニタリング](#)」を参照してください。

- 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 + C] 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 つまたは複数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。

サーバを停止できない

通常、サーバを停止するには、`dbstop` ユ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、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。このいずれの方法を使用してもサーバが停止しない場合は、下の「対処法」の項を参照してください。

考えられる原因

さまざまな原因が考えられます。

対処法

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 Sybase 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|iqrsv15"
/usr/ucb/ps -awwwlx|egrep "PPID|iqrsv15"
```

- 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、システムをリブートします。
- 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 Sybase 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 [iqsrv15.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。

Sybase 製品の保守契約を結んでいるサポート・センタに提出する情報の全リストについては、「[サポート・センタへの問題の報告](#)」(567 ページ) の項を参照してください。

システム障害 /Sybase IQ の障害

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、「[サポート・センタへの問題の報告](#)」(567 ページ) の項でリストされている追加情報とともに 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、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。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・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。

参照

- 「第 13 章 システムのリカバリとデータベースの修復」
- 『システム管理ガイド第 1 巻』の「第 10 章 トランザクションとバージョン管理」の「システム・リカバリ」

データベース接続の問題

この項では、データベースへの接続時に発生する可能性のある問題について説明します。

データベースに接続できない

考えられる原因

- データ・ソースが定義されていない。または、データ・ソースを誤って入力したか、誤って定義した。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 および 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。この場合は次のエラー・メッセージが返されます。

```
Unable to connect
```

または

```
Could not connect to the database.
```

その後ろに次のメッセージが続きます。

```
Invalid user ID or password.
```

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。

- 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。この場合は次のエラー・メッセージが返されます。

```
Unable to connect
```

または

```
Could not connect to the database.
```

その後ろに次のメッセージが続きます。

```
Specified database not found.
```

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 パラメータおよび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。詳細については、「[リストア後の再接続](#)」(475 ページ) を参照してください。

- 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。この場合は次のエラー・メッセージが返されます。

```
Unable to connect
Database server connection limit exceeded.
```

- ディスク領域が不足している。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。この場合は次のエラー・メッセージが返されます。

```
Connection failed.
Database server not running.
```

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ります。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` コマンド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、次のメッセージが表示される。

```
Unable to start - server not found
```

または

```
Database server not running.
```

これはクライアントがネットワーク上のデータベース・サーバを検出できないことを示します。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。要求された文字セットをサーバがサポートしていない場合は、接続が失敗します。

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。

注意 データベースに接続できないことと、Sybase IQ がデータベースを開こうとしたときに起きる Sybase IQ サーバ・レベルのエラーを混同しないでください。

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。”というメッセージが表示されている場合、データベースは正常に開かれていますので、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。

参照

- データ・ソースを作成および編集する方法、Sybase IQ が接続を確立するしくみ、ポート番号を指定する方法、データベース接続の問題をトラブルシューティングする方法の詳細については、『システム管理ガイド第 1 巻』の「[第 3 章 Sybase IQ の接続](#)」
- データベース・パーミッションについては、「[第 8 章 ユーザ ID とパーミッションの管理](#)」
- 『リファレンス：文とオプション』の「[LOGIN_PROCEDURE オプション](#)」
- 「[ディスク領域の不足](#)」(537 ページ)
- 「[第 13 章 システムのリカバリとデータベースの修復](#)」

Interactive SQL (dbisql) の問題

この項では、dbisql の操作に関連する問題のトラブルシューティングについて説明します。

データ・トランケーションまたはデータ変換エラー

考えられる原因

あるプロシージャが別のプロシージャを動的結果セットで呼び出し、以下のすべてが成り立っていると、データ・トランケーション・エラーまたはデータ変換エラーが発生します。

- Sybase IQ サーバーのバージョンが 12.5 である。
- dbisql が iAnywhere JDBC ドライバを介して接続している。
- dbisql のバージョンが 7.04 より上である。

dbisql が ODBC ドライバを介して接続している場合、または Sybase IQ 12.6 が dbisql 9.0.1 とともに使用されている場合は、この問題は発生しません。

- 使用中の端末と Sybase IQ が予期している端末とで、表示特性が異なっている。
- 使用中の端末と Sybase IQ が予期している端末とで、サポートするファンクション・キーが異なっている。

対処法

この問題を回避する方法はいくつかあります。

- ODBC ドライバを介して dbisql を接続します。
- 次のいずれかを使用できます。
 - Sybase IQ 15.1 または 15.2 と dbisql バージョン 11.0.1 の組み合わせ
 - Sybase IQ 12.7 と dbisql バージョン 9.0.2 の組み合わせ
 - Sybase IQ 12.6 と dbisql バージョン 9.0.1 の組み合わせ
 - Sybase IQ 12.5 と dbisql バージョン 7.0.4 の組み合わせ
- 次のような文をプロシージャの先頭に追加し、サーバが結果セットを追加しないようにします。

```
IF 1 = 0 THEN
SELECT 1 AS a FROM nosuchtable;
END IF;
```

dbisql を終了した後もディレクトリが残る

注意 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

IQTMP15 環境変数がローカル・ディレクトリを指すように設定されていない。クライアント接続が行われるたびに、テンポラリ・ディレクトリにいくつかのディレクトリとファイルが作成されます。これらのファイルは、接続が終了するときに Sybase IQ によって削除されます。IQTMP15 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 IQTMP15 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

この項では、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、リソースの問題のトラブルシューティングについて説明します。

ディスク領域の不足

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代わりに、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 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 dbfile 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、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か、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。

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する例については、「[ディスク領域の使用状況のモニタリング](#)」(541 ページ) の項を参照してください。

対処法

- 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” というメッセージが表示されている場合は、別の DB 領域を追加する必要があります。メッセージ・ファイルの Sybase IQ 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。

Sybase IQ メッセージ・ログに次のメッセージが残されてい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
[MSG_OUT_OF_DBSPACE: SQL Code -1009170L,
SQL State QSB66, Sybase Error Code 20223]
```

%2 は DB 領域の名前です。

このメッセージは、次のエラー・メッセージに置き換わります。
データベース <dbname> 内の { IQ STORE | IQ TEMPORARY STORE } DB 領域が不足しています。別のセッションで CREATE DBSPACE ... { IQ STORE | IQ TEMPORARY STORE } コマンドを発行して、少なくとも **nn** MB の DB 領域を追加してください。

- 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ることがわかります。**sp_iqstatus** コマンドを実行し、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 使用率をモニタして、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、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、Sybase IQ は稼働しています。
- sp_iqstatus** 出力で、次の 2 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。

- 領域不足の状況が発生している場合、またはマルチプレックス・サーバで `sp_iqstatus` によるメイン・ブロックの使用が高いパーセンテージを示している場合は、`sp_iqversionuse` を実行して、使用されているバージョン、およびそのバージョンでリカバリできる領域の量を確認します。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「第 7 章 システム・プロシージャ」の「`sp_iqversionuse` プロシージャ」を参照してください。

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。詳細については、「強制リカバリ・モードでのサーバの起動」(510 ページ)を参照してください。マルチプレックス・サーバについては、『Sybase IQ マルチプレックスの使用』を参照してください。

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 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、次の手順に従います。

- 次のいずれかの方法でデータベース・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、Sybase IQ ウィンドウを表示します。次に、[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。

サーバが停止しない場合は、「サーバを停止できない」を参照してください。

- `start_iq` コマンドを使用してエンジンを再起動します。
- データベースに接続します。
- `CREATE DBSPACE` コマンドを使用して領域を追加します。
- 領域の不足によって失敗した操作を再度実行します。

dbspace 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 文で IN SYSTEM を使用しない。
- システム・ストアド・プロシージャを実行した後で COMMIT 文を発行します。
- 長時間実行されるトランザクションの後で COMMIT 文を発行します。

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、Sybase IQ は「DB 領域が最大ファイル・サイズに達しています。」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、.db、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、メイン Sybase IQ データベース・ディレクトリにあります。.tmp ファイルは、\$IQTMP15/<servername>/tmp の下か、\$IQTMP15 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、誤って 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

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、たとえば sp_iqstatus や .iqmsg ファイルによってリストされるメモリ使用率で、使用したメイン IQ ブロックが 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。

ディスク領域の使用状況のモニタリング

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。「[ディスク領域の使用状況のモニタリング](#)」(541 ページ) の項で、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、領域が残り少なくなったときに通知させる例を参照してください。

将来のための領域の確保

Sybase IQ は最低 200MB、最後の DB 領域のサイズの 50% を予約します。

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

MAIN_RESERVED_DBSPACE_MB および

TEMP_RESERVED_DBSPACE_MB を設定します。COMMIT または

CHECKPOINT の実行中の領域不足に対処できるように、これらのオプションを十分なサイズに設定します。詳細については、『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」を参照してください。

デ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。

次にイベント・ハンドラのサンプル・コードを示します。この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 iqsrv15 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_dbpace 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_dbpace 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、特権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;
```

イベントの使用方法の詳細については、『システム管理ガイド第2巻』の「[第6章 スケジューリングとイベント処理によるタスクの自動化](#)」を参照してください。イベントを作成、修正、制御するSQL文の詳細については、『リファレンス: 文とオプション』の「[第1章 SQL文](#)」の「[CREATE EVENT 文](#)」、「[ALTER EVENT 文](#)」、「[TRIGGER EVENT 文](#)」を参照してください。

スレッドの不足

考えられる原因

クライアントがこのクエリで利用可能なサーバ・スレッドが不十分です。[-1010011] ['QXA11'] のようなメッセージを受け取った場合、クエリは IQ ストア用により多くのカーネル・スレッドを必要としている。

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。
- システム・ストアド・プロシージャ `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。スレッドの中には、割り当てられてもアイドル状態になっているものがあることに注意してください。
- 状況が持続する場合は、サーバを再起動し、より多くの IQ スレッドを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、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、 $60 * 4 + 50 * (\text{numCPUs} - 4) + \text{numConnections} + 3 = 653$ となります。最小値は、 $\text{numConnections} + 3$ です。サーバ・スレッドの総数が、64 ビット・プラットフォームでは 4096、32 ビット・プラットフォームでは 2048 を超えることはできません。『ユーティリティ・ガイド』の「第 1 章 データベース・サーバの実行」の「データベース・サーバの起動」を参照してください。
- 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、Sybase IQ は「スレッドが不足しています。その他の追加スレッドでサーバを起動してください。」(SQLCODE -1012024) というエラーを返すことがあります。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ています。Sybase IQ では、各バックアップ・デバイスに少なくとも 1 つのスレッド、各 CPU に 2 つのスレッド、またリストア用の「チーム」に 1 つのスレッドを割り付けようとしています。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。『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」の「`MAX_IQ_THREADS_PER_CONNECTION` オプション」および「`MAX_IQ_THREADS_PER_TEAM` オプション」を参照してください。

スタック・オーバーフロー

考えられる原因 スタック・トレース・ファイル (*stktrc-YYYYMMDD-HHNNSS_#.iq*) にエラー `AbortIfEndofStack` が記録されている場合は、スレッド・スタックがオーバーフローしています。

対処法

- 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 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、Sybase 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

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 `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。Sybase IQ のすべてのインスタンスが (Sybase が推奨するように) 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 `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
```

不十分なプロシージャ識別子

Sybase IQ は内部カタログ `proc_id` をプロシージャに対して順番に割り当て、未使用 `proc_id` は再使用されません。プロシージャが削除されて作成されると、Sybase IQ 12.6 より前に作成されたデータベースは、最終的に `proc_id` の上限である 32767 に達し、`CREATE PROCEDURE` が Sybase IQ 12.6 で “Item already exists” エラーを返す場合があります。

Sybase IQ 12.6 GA より前のバージョンで作成されたデータベースの場合、データベースが Sybase IQ 12.6 以降にアップグレードされていたとしても、プロシージャに対する `proc_id` の上限は 32767 です。Sybase IQ 12.6 以降で作成されたデータベースには、この制限は適用されません。

`proc_id` カラムのデータ型が `SMALLINT` の場合は、`proc_id` の最大値 32767 が適用されます。使用しているデータベースに対する `proc_id` の現在の最大値を判別するには、次のクエリを実行します。

```
SELECT MAX (proc_id) FROM sys.sysprocedure
```

Sybase IQ 12.6 ESD7 以降では、Sybase IQ 12.6 より前に作成されたデータベースに対し、最大の `proc_id` が `ALTER DATABASE UPGRADE` を実行できるレベルであることを保証します。最大の `proc_id` がこれより大きいと、`ALTER DATABASE UPGRADE` は失敗し、メッセージ “Database upgrade not possible” を返します。

Sybase IQ 12.6 より前のバージョンで作成されたデータベースに対するこの問題を解決するため、`ALTER DATABASE UPGRADE` では、すべてのストアード・プロシージャを作成し直すことで `proc_id` を圧縮する、12.6 ESD7 以降の `PROCEDURE ON` 句をサポートします。構文は `ALTER DATABASE UPGRADE PROCEDURE ON` です。12.6 以降で作成されたデータベースの場合は、`PROCEDURE ON` 句は無視されます。

`ALTER DATABASE UPGRADE PROCEDURE ON` は、すべてのプロシージャをコメントなしで作成し直します。コマンドを実行した後でプロシージャにコメントを戻したい場合は、コメントを含むプロシージャに対するソース・コードで `ALTER PROCEDURE <procedure_name>` を実行します。`ALTER DATABASE UPGRADE PROCEDURE ON` を実行する前に、`sp_helptext <owner>.<procname>` コマンドを使用して、コメント付きのプロシージャのテキストを保存できます。

`ALTER DATABASE UPGRADE PROCEDURE ON` を実行する直前に、バックアップとして、データベースの `.db` ファイルと `.log` ファイルをコピーします。`ALTER DATABASE UPGRADE` コマンドで変更されるのはカタログだけなので、フル・バックアップは必要ありません。

処理の問題

この項では、ロード、クエリ、インデックス、テーブル・アクセスに関連する処理の問題のトラブルシューティングについて説明します。

ロード時のディスク領域の使用状況のモニタリングについては、「[ディスク領域の使用状況のモニタリング](#)」(541 ページ) を参照してください。

テーブルのインデックスが多すぎる

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。
参照	『Sybase IQ の概要』の「 第 8 章 ビューの使用 」

ロードまたはクエリに予想以上の時間がかかる

考えられる原因	<ul style="list-style-type: none"> 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、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。このため、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 または LF インデックスがない。
対処法	<p>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。</p> <ul style="list-style-type: none"> 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。IQ バッファ・キャッシュ・モニタについては、『パフォーマンス&チューニング・ガイド』の「第 5 章 パフォーマンスのモニタリングとチューニング」の「バッファ・キャッシュのモニタリング」を参照してください。 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 <code>vmstat</code> 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 パフォーマンス モニタを使用します。 <p>必要に応じてバッファ・サイズをリセットします。『パフォーマンス&チューニング・ガイド』の「第 5 章 パフォーマンスのモニタリングとチューニング」の「バッファ・キャッシュのモニタリング」を参照してください。</p> <p>ページングをモニタし、スラッシングが問題と判断した場合は、ハッシュ・アルゴリズムを伴うクエリが含まれる文の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。</p>

HASH_THRASHING_PERCENT のデフォルト値は 10% です。HASH_THRASHING_PERCENT の値を増やすと、ロールバックが起きるまでに許容するディスクへのページングが増え、HASH_THRASHING_PERCENT の値を減らすと、ロールバックが起きるまでに許容するページングが減ります。

以前のバージョンの 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、デフォルトの HASH_THRASHING_PERCENT の制限に達するとロールバックされるようになります。エラー Hash insert thrashing detected. または Hash find thrashing detected. (SQLState QFA43、SQLCode -1001047) がレポートされます。実行に必要なリソースをクエリに割り当てるには、次の 1 つ以上の対応策を講じてください。

- 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・キャッシュのサイズが減ることに注意してください。
- 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 の値を減らします。

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。

参照

- 『パフォーマンス&チューニング・ガイド』の「[第 4 章 システム・リソースの管理](#)」
- 『パフォーマンス&チューニング・ガイド』の「[第 5 章 パフォーマンスのモニタリングとチューニング](#)」
- 『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」の「[HASH_THRASHING_PERCENT オプション](#)」および「[HASH_PINNABLE_CACHE_PERCENT オプション](#)」

ユニークな値の数でロードが失敗する

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。

```
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。
これを行うには、**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。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。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

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。

参照 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」および「[テーブルでの書き込みロック競合の管理](#)」(548 ページ)

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。この項のサンプル・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ッ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。

次のストアド・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
```

ストアド・プロシージャの使用方法の詳細については、『システム管理ガイド 第2巻』の「[第1章 プロシージャとバッチの使用](#)」を参照してください。

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 WRITE ロックを予約します。たとえば、次の例では `SalesOrders`、`Customers`、および `SalesOrderItems` の各テーブルで WRITE 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。

```

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;

```

参照 ロックおよびロックの管理の詳細については、「[ロックの仕組み](#)」(408 ページ)を参照してください。

チェックポイントのヒント

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。

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、変更する必要はありません。`-gc` と `-gr` の値を変更するよう指示された場合は、これらのサーバ・スイッチの設定方法の詳細について、『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」を参照してください。

パフォーマンスの問題

この項では、パフォーマンスに影響を与える可能性があるいくつかの設定について説明します。パフォーマンスの問題を診断および解決する方法の詳細については、『パフォーマンス&チューニング・ガイド』の以下の章を参照してください。

- Sybase IQ メモリ、ディスク、その他のリソースの使用とパフォーマンスへの影響について理解するには、「[第 4 章 システム・リソースの管理](#)」を参照してください。
- IQ バッファ・キャッシュ・モニタを使用する方法については、「[第 5 章 パフォーマンスのモニタリングとチューニング](#)」を参照してください。

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

考えられる原因

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 Sybase IQ がアクセスできないマシンでは、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 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。詳細については、『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」の -iqnumbercpus サーバ・オプションを参照してください。

Sybase Central の問題

この項では、Sybase Central の操作に関連する問題のトラブルシューティングについて説明します。

Sybase Central の一部のフィールドが表示されない

考えられる原因

システムで暗い背景色と白色のテキストを使用している。

対処法

ウィンドウ・システムに備わっている機能を使い、Sybase Central の表示色を白色または明るい色の背景色と暗い色のテキストに変更します。

ネットワーク通信のトラブルシューティング

これ以降の項では、主に Windows および Windows ベースのクライアントで発生する通信の問題のトラブルシューティングについて説明します。

ネットワーク・ソフトウェアは、さまざまな種類のコンポーネントで構成されるため、問題が起きる可能性が高くなります。ここでは、ネットワークのトラブルシューティングに関するいくつかのヒントを紹介しますが、ネットワークの問題を解決するときは、まずネットワーク通信ソフトウェアのベンダが提供しているマニュアルやテクニカル・サポートを活用してください。

また、これ以降の項で説明するものを含めて、さまざまな状況を診断するとき役に立つ情報の取得方法については、「[診断ツール](#)」(556 ページ)の項を参照してください。

互換性のあるプロトコルを使用していることの確認

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。

`-x` スwitchの詳細については、『[ユーティリティ・ガイド](#)』の「[第1章 データベース・サーバの実行](#)」を参照してください。

最新のドライバを使用していることの確認

古いネットワーク・アダプタ・ドライバが通信の問題の原因となることがよくあります。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。

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。

リブート時にコンピュータの電源を切る

ネットワーク・アダプタ・カードの中には、コンピュータをリブート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てリブートします。

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ときは、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する方法が役立ちます。

Sybase IQ と同じ上位レイヤ・プロトコルを使った接続でなくても、何らかの方法でサーバ・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 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正しく機能す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。多くの TCP/IP パッケージに付属する ping ユーティリティは、IP ネットワーク・レイヤのテストに役立ちます。

ping を使った 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、使用中のコンピュータの IP アドレスが 191.72.109.12 の場合、コマンド・ライン・プロンプトで次のように入力します。

```
ping 191.72.109.12
```

ping を送信したら、パケットがルーティングされているかどうかを確認します。ルーティングされている場合は、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、コンピュータは自身にパケットをルーティングできています。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。

この先の作業に進む前に、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。次の手順では、telnet コマンドを使用して TCP/IP スタックをテストする方法を示します。

- 1 1 台のマシンで Telnet サーバ・プロセス (デーモン) を起動します。この方法については、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。コマンド・ライン・プログラムでは、一般に次の命令を入力します。

```
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 接続が確立されれば、プロトコル・スタックは安定しており、クライアントとサーバは2つのコンピュータ間で TCP/IP リンクを使って通信できるはずですが、Telnet 接続を確立できない場合は、問題があります。この先の作業に進む前に、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。1つのマシンでのみ問題が起きる場合は、配線の問題またはハードウェアの問題である可能性があります。

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

ネットワーク通信パラメータについては、『システム管理ガイド第1巻』の「[第4章 接続パラメータと通信パラメータ](#)」の「[ネットワーク通信のパラメータ](#)」を参照してください。

ここでは、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示します。

「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、次のメッセージが表示されたとします。

```
Unable to start - server not found
```

これはクライアントがネットワーク上のデータベース・サーバを検出できないことを示します。次の問題がないかどうかを確認してください。

- 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る。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 98、Windows NT、Windows 2000 では、[コントロールパネル]のネットワークの設定を通じて設定にアクセスします。

- 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・ドライバが正しくインストールされていないか、ネットワーク配線が正しく設置されていない。
- 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、Sybase IQ マルチユーザ・サポートに適合していない。

「Unable to initialize any communication links」メッセージ

クライアントを起動しようとしたときに、次のメッセージが表示されたとします。

```
Unable to initialize any communication links
```

これはリンクを確立できないことを示します。ネットワーク・ドライバがインストールされていない可能性があります。-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

この項では、前の項で説明したものを含めて、さまざまな状況を診断するときに役立つ情報の取得方法について説明します。

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 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。

```

Sybase IQ (TM)                Copyright (c) 1992-2010 by Sybase, Inc.
                                All rights reserved.
Version:                        15.2.0.5027/0490416/P/GA/MS/
                                Windows 2000/32bit/2010-04-16 02:11:41
Time Now:                       2010-04-27 14:09:00.648
Build Time:                     2010-04-16 09:54:19
File Format:                    23 on 03/18/1999
Server mode:                   IQ Server
Catalog Format:                 2
Stored Procedure Revision:     1
Page Size:                     131072/8192blksz/16bpp
Number of Main DB Files:      2
Main Store Out of Space:      N
Number of Temp Files:         1
DB Blocks: 1-3200              IQ_SYSTEM_MAIN
DB Blocks: 1045440-1055039    iq_main
Temp Blocks: 1-1600           IQ_SYSTEM_TEMP
Create Time:                   2010-04-03 14:14:06.124
Update Time:                   2010-04-25 14:14:26.687
Main IQ Buffers:              255, 32Mb
Temporary IQ Buffers:         191, 24Mb
Main IQ Blocks Used:          5915 of 11200, 52%=46Mb,
                                Max Block#: 1051278
Temporary IQ Blocks Used:     81 of 800, 10%=0Mb, Max Block#: 161
Main Reserved Blocks Available: 1600 of 1600, 100%=12Mb
Temporary Reserved Blocks Available: 800 of 800, 100%=6Mb
IQ Dynamic Memory:            Current: 69mb, Max: 70mb
Main IQ Buffers:              Used: 6, Locked: 0
Temporary IQ Buffers:        Used: 5, Locked: 0
Main IQ I/O:                  I: L899/P3 O: C3/D91/P89 D:0 C:100.0
Temporary IQ I/O:            I: L4043/P0 O:C674/D718/P47 D:669 C:100.0
Other Txn Versions:          0 = 0Mb
Active Txn Versions:         0 = C:0Mb/D:0Mb
Last Full Backup ID:         0
Last Full Backup Time:
Last Backup ID:               0
Last Backup Type:            None
Last Backup Time:
DB Updated:                   1
Blocks in next ISF Backup:    0 Blocks: =0Mb
Blocks in next ISI Backup:    0 Blocks: =0Mb
File Encryption Status:      OFF

```

メイン IQ I/O とテンポラリ IQ I/O の出力コードの意味は次のとおりです。

- I: 入力
- L: 読み込まれた論理ページ数 (“Finds”)
- P: 読み込まれた物理ページ数

- O: 出力
- C: 作成されたページ数
- D: ダーティ・ページの数
- P: 物理的書き込み
- D: 破損したページ数
- C: 圧縮率

次の情報を確認します。

- 使用したメイン IQ ブロックおよび使用したテンポラリ IQ ブロックは、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、dbspace を追加する必要があります。
- メイン IQ バッファおよびテンポラリ IQ バッファは、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。
- 他のバージョンは、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。
- アクティブなトランザクション・バージョンは、アクティブな書き込みトランザクションの数と、それらが作成および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。
- 使用できる予約済みメイン・ブロックおよび使用できる予約済みテンポラリ・ブロックは、使用可能な予約済み領域の量を示します。
- メイン IQ I/O およびテンポラリ IQ I/O 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。例については、「[メイン・バッファ・キャッシュ・アクティビティ・メッセージ](#)」(561 ページ) を参照してください。

通知メッセージの解釈

デフォルトでは、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。この項では、各通知メッセージについて説明します。

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挿入されるたびに、Sybase IQ 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 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。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

このメッセージにはメモリ使用に関する情報が表示されます。

```
Mem: 469mb/M470
```

表 14-2: メモリ・メッセージ

項目	説明
Mem:# mb	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	IQ サーバの起動後に使用された最大のメガバイト数

メイン IQ ストアによるメッセージのブロック

この行は永続的な (メイン) IQ ストアを示します。

Main Blks: U63137/6%, Buffers: U12578/L7

表 14-3: メイン・ブロック

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。これは 100% であるのが普通です。バッファ・マネージャは、他のデータに対してバッファを使用する必要が生じるまでは、バッファをメモリに残しておくからです。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す。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からです。</p> <p>注：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ります。バッファが割り当てられるたびにこの数字は増えますが、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りません。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しますが、メイン・キャッシュについては、IQ がバッファを破棄する場合としない場合があります。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ますからです。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ありません。</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とは、現在使用中でキャッシュから取り除くことができないバッファのことです。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します。また、ソート・オブジェク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、バッファをロックします。</p> <p>この数字はバッファを要求するたびに増えます。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ります。</p> <p>注意 バッファのロックではメモリは消費されません。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ます。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します。</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](#)」(541 ページ)を参照してください。

IQ テンポラリ・ストアによるメッセージのブロック

このメッセージは前述のメイン IQ ストア・ブロック・メッセージに似た情報を提供します。

Temporary Blks: U273/0%, Buffers: U1987/L1960

メイン・バッファ・キャッシュ・アクティビティ・メッセージ

この行にはメイン・バッファ・キャッシュに関する情報が表示されます。

Main I: L331224/P22 O: D25967/P7805 C:D0

表 14-4: メイン IQ ストア・ファイル・メッセージ

項目	説明
Main: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
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。この値には、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場合があります。

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。前述の値になるような場合のバッファ・キャッシュ・サイズの設定または IQ パフォーマンス・モニタの使用の詳細については、『パフォーマンス&チューニング・ガイド』を参照してください。

テンポラリー・バッファ・キャッシュ・メッセージ

この行にはテンポラリー・バッファ・キャッシュに関する情報が表示されます。

Temporary I: L25240/P8 O: D4749/P0 C:D0

前述のメイン・バッファ・キャッシュ・メッセージについての説明を参照してください。

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)、ユーザ名がロギングされます。これらは、`.iqmsg` ファイルに対して、1 つのデータベース接続について 1 回のみ記録されます。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、「[接続情報の関連付け](#)」(570 ページ) を参照してください。

```
2010-05-12 09:34:42 0000000002 Txn 173
2010-05-12 09:34:42 0000000002 Connect: 1550990889. SA connID: 1. User: DBA.
```

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` ストアド・プロシージャは、最後に実行された `sp_iqcheckdb` プロシージャによって収集されたデータベース統計情報を表示します。

`sp_iqcheckdb` および `sp_iqdbstatistics` ストアド・プロシージャの実行と使用の詳細については、「[第 13 章 システムのリカバリとデータベースの修復](#)」の「[データベースの検証](#)」を参照してください。

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。

`sp_iqcheckoptions` を実行すると、DBA にはすべてのグループとユーザに永続的に設定されているすべてのオプションと、DBA に設定されているテンポラリ・オプションが表示されます。DBA 以外のユーザには、そのユーザ自身の temporary オプションが表示されます。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、次のコマンドを実行します。

```
sp_iqcheckoptions
```

システム・テーブル DBO.SYSOPTIONDEFAULTS には、Sybase IQ および ASA 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。すべてのオプションのデフォルト値を参照する必要がある場合は、このテーブルにクエリを実行します。

詳細については、『リファレンス：文とオプション』の「[第 7 章 システム・プロシージャ](#)」の「[sp_iqcheckoptions プロシージャ](#)」、および「[第 2 章 データベース・オプション](#)」の「[オプション設定の検索](#)」を参照してください。

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、プロシージャの実行時にシステムでどの文が実行されていたかを示し、その文を発行したユーザと接続を識別します。sp_iqconnection、.iqmsg ログ、-zr サーバ要求ログ (.srvlog)、およびスタック・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。

詳細および出力例については、『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「[第 7 章 システム・プロシージャ](#)」の「[sp_iqcontext プロシージャ](#)」を参照してください。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、「[接続情報の関連付け](#)」(570 ページ)を参照してください。

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、サーバ要求のロギングが役に立ちます。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 コマンド・ライン・オプションの現在の設定を上書きします。

サーバ要求は、サーバ・ログ・ファイル *.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 (ALL、NONE、または SQL) を設定します。-zo オプションは、要求レベル・ログ情報を通常のログ・ファイルとは別のファイルにリダイレクトし、-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。『ユーティリティ・ガイド』の「[第 1 章 データベース・サーバの実行](#)」の「[データベース・サーバの起動](#)」を参照してください。

`sa_server_option` ストアド・プロシージャを使用すると、Sybase IQ サーバを再起動せずに要求レベル・ログの有効と無効を切り替えることができます。次のコマンドは、限られた一連の要求の要求レベル・ログを有効にし、出力を `sqllog.txt` ファイルにリダイレクトします。

```
call sa_server_option('request_level_logging','SQL');
call sa_server_option('request_level_log_file',
                      'sqllog.txt');
```

次のコマンドは、要求レベル・ログを無効にします。

```
call sa_server_option('request_level_log_file','');
```

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、「[接続情報の関連付け](#)」(570 ページ) を参照してください。

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ます。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。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、UPDATE、および 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。

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に、要求ログから抜粋を示します。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” の行以降のための、オフセットは前の行からあります。この場合 “+13” は、最終行から 13 ミリ秒が経過したことを意味します。場合によっては、“=” は最終行から 0 ミリ秒が経過したことを意味します。

次に、要求ログの要約を示します。


```

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

要求ログ・ファイルへの出力フォーマット (-zr サーバ起動スイッチを設定して生成) は、Sybase IQ 15.1 で変更されました。ストアド・プロシージャ `sa_get_request_profile` および `sa_get_request_times` を使用して、-zr ログ・ファイルを読み込み、結果を要約できます。

`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・テンポラリ・テーブル `satmp_request_profile` に結果を要約します。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・テンポラリ・テーブル `satmp_request_time` に結果を要約します。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「[第1章 データベース・サーバの実行](#)」の「[start iq の server options](#)」にある `-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「[第7章 システム・プロシージャ](#)」の「[sa_server_option システム・プロシージャ](#)」、『SQL Anywhere サーバ - SQL の使用法』の「データベース・パフォーマンスのモニタリングと改善」>「データベース・パフォーマンスの改善」>「その他の診断ツールと方法」にある「要求ロギング」を参照してください。

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。詳細については、『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

Sybase ソフトウェアのインストール環境ごとに、Sybase 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者がいます。マニュアルだけ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 Sybase のサポート・センタまでご連絡ください。

サポート・センタでは問題を解決するために、ご使用の Sybase IQ 環境についての情報が必要となります。この項では、問題の解決に必要な情報、自動化された `getiqinfo` ツールを使用してこの情報を収集する方法、さまざまな Sybase IQ ユーティリティとログ・ファイルの情報を関連付ける方法について説明します。

getiqinfo を使った診断情報の収集

Sybase IQ には、Sybase 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、Sybase IQ 環境、ログ・ファイルについての情報を収集します。

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、「サポート・センタへの問題の報告」(567 ページ)で説明するすべての情報に加えて、問題の解決に必要なその他の情報も自動的に収集します。

`getiqinfo` スクリプトは、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、Sybase IQ 環境が正しくセットアップされており、サーバが実行されている場合のみ正常に実行されます。

`getiqinfo` を実行する前に

スクリプトを実行する前に、次の情報を用意しておきます。

- 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
- 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、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、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、Sybase IQ サーバの起動に使用されたの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っ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-15_2¥win32` ディレクトリにあるバッチ・スクリプトです。

❖ `getiqinfo` を実行する

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-15_2/bin32` ディレクトリ (32 ビット・プラットフォームの場合) または `IQ-15_2/bin64` ディレクトリ (64 ビット・プラットフォームの場合) に移動し、次のように入力します。

```
getiqinfo.sh
```

- Windows で、[スタート] - [ファイル名を指定して実行] - [`<install_path>¥IQ-15_2¥bin32¥getiqinfo.bat`] (32 ビット・プラットフォームの場合) または [`<install_path>¥IQ-15_2¥bin64¥getiqinfo.bat`] (64 ビット・プラットフォームの場合) を選択します。

2 プログラムのプロンプトが表示されたら、次の情報を入力します。

- 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、および UNIX では `stktre*.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 Sybase IQ エンジン名 (サーバ名) とポート番号。
- このデータベースの DBA 権限を持つユーザ ID とパスワード。
- 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 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 スクリプトは、次のすべての情報を収集します。

- 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
- 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 *stktre-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 の場合：*IQ-15_2/logfiles/<servername>.000n.stderr* および *IQ-15_2/logfiles/<servername>.000n.srvlog*
 - 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*) の起動および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

万が一この情報を手動で収集することになった場合のために、サポート・センタで必要となる情報を記録するためのチェックリストをこの章の末尾に掲載します。

注意 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度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。

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。

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。次の手順は、`.srvlog` ファイルと `.iqmsg` ファイル内の接続識別子を関連付けて、関連情報を見つける方法を示しています。

❖ `.srvlog` ファイルと `.iqmsg` ファイル間での接続情報の関連付け

1 `.iqmsg` ファイルで、目的の接続を探します。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。

```
16:14:59. [,1000000061,sp_iqdbspace,48,select
str_replace(dbspaceName,'" ',null) into dbspaceName_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_dbspace_temp dbspace info
' || dbspaceName
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 ROWS

03/17 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。

その他の情報源

問題を解決できない場合は、Sybase 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** を指定し、[MySybase] をクリックします。
(<http://www.sybase.com/support/techdocs/>)

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、この情報の大部分を収集できます。

要求される情報	値
Sybase IQ のバージョン (15.2 GA や ESD 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 および server.nnnn.stderr)	
スタック・トレース・ファイル (stktre-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・スナップショット (可能な場合)	

索引

数字

- 2000 年
- 変換オプション 317

A

- Access
 - ODBC 設定 73
- Adaptive Server Enterprise
 - text と image データの挿入 301
 - unichar データ型 324
 - unitext データ型 324
 - univarchar データ型 324
 - データの挿入 59, 300
- Adaptive Server Enterprise の unichar データ型 324
- Adaptive Server Enterprise の unitext データ型 324
- Adaptive Server Enterprise の univarchar データ型 324
- Advanced Security オプション 370
- ALL パーミッション 351
- allocation
 - DBCC 修復の出力 508
- ALTER DBSPACE 文
 - ADD パラメータ 191
 - SIZE パラメータ 192
 - 例 191
- ALTER TABLE 文
 - 外部キー 389
- ALTER パーミッション 351
- ALTER 文
 - オートコミット 399
- ANSI コード・ページ
 - 選択 429
- APPEND_LOAD オプション
 - 分割されたテーブル 286
- AppInfo 接続パラメータ 110
- ASCII
 - 挿入時の変換 314
 - データ抽出 274
 - データ・フォーマット 267
 - 変換オプション 308

- 変換のパフォーマンス 312
- 文字セット 427
- AStart 接続パラメータ 113
- ASTMP 環境変数
 - ディスク容量 103
- AStop 接続パラメータ 113
- AUTOINCREMENT
 - default 381
 - Ultra Light アプリケーション 381
 - 符号付きデータ型 381
 - 負の数 381
- AutoPreCommit 接続パラメータ 112
- AutoStart 接続パラメータ 113
- AutoStop 接続パラメータ 113
- AVG 関数 229

B

- BACKUP 文 448
- BCAST 通信パラメータ
 - 説明 133
- bcp サポート 267
- binary
 - データ抽出 274
- BINARY ロード形式
 - LOAD TABLE 291
 - データ・ファイル 291
- BIT データ
 - 使用可能なインデックス 223
 - 変換 311
- BLISTENER 通信パラメータ
 - 説明 134
- BLOB データ 301
- BLOCK FACTOR
 - BACKUP 文のオプション 453
- BroadcastListener 通信パラメータ
 - 説明 134
- Broadcast 通信パラメータ
 - 説明 133

C

CATALOG ONLY
 RESTORE オプション 477
 CBSize 接続パラメータ 114
 Certificate_Password 通信パラメータ
 説明 135
 Certificate 通信パラメータ
 説明 134
 CharSet 接続パラメータ 114
 CHAR データ
 長さ 0 のセル 315
 CHECK 条件
 カラム 384
 削除 386
 テーブル 386
 変更 386
 ユーザ定義データ型 385
 ClientPort 通信パラメータ 135
 CLOB データ 301
 CMP インデックス 234
 推奨される使用法 234
 制限 234
 CommBufferSize 接続パラメータ 114
 CommLinks 接続パラメータ 115
 connect
 パーミッション 348
 ConnectFailed イベント・ハンドラ 344
 ConnectionName 接続パラメータ 117
 CONNECT 文
 デフォルト・データベース 93
 CONVERSION_ERROR データベース オプション 326
 COUNT DISTINCT
 インデックス選択への影響 220
 COUNT 関数 229
 CREATE DATABASE 文
 IQ RESERVE パラメータ 191
 IQ SIZE パラメータ 191
 TEMPORARY SIZE パラメータ 191
 TEMPORARY RESERVE パラメータ 191
 照合 426
 CREATE DBSPACE 文
 SIZE パラメータ 191
 RESERVE パラメータ 191
 CREATE INDEX 文 218
 CREATE JOIN INDEX 文 200, 255
 CREATE TABLE 文
 コマンド・ファイル 153
 の例 197

CREATE パーミッション 340
 CREATE 文
 オートコミット 399
 同時実行性の規則 410
 CS 接続パラメータ 114
 CS_TEXT_TYPE 301
 -c スイッチ 33

D

DatabaseFile 接続パラメータ 117
 DatabaseName 接続パラメータ 118
 DatabaseName 通信パラメータ
 説明 136
 DatabaseSwitches 接続パラメータ 119
 DataSourceName 接続パラメータ 120
 DATE インデックス
 推奨される使用法 237
 DATEPART
 クエリ 238
 DATETIME
 変換オプション 308
 ロード変換オプション 317
 DATETIME データ型 318
 変換フォーマット 318
 ロードの最適化 312
 DATE インデックス 237
 他のインデックスとの比較 242
 短所 241
 長所 241
 追加インデックス 242
 DATE オプション 308, 315
 DATE データ型
 変換フォーマットの指定 315
 ロードの最適化 312
 DATE フォーマット
 2 桁の日付の変換 317
 DBA (データベース管理者)
 定義 336
 DBASE フォーマット 267
 DBA 権限
 継承不可能 356
 説明 336
 付与 336, 350
 DBA (データベース管理者)
 役割 5

DBCC

- sp_iqcheckdb インタフェース 495
- インデックスと割り付けの検査 496
- インデックスの検証と修復 502
- インデックスの修復 502
- インデックスの問題の検出 515
- インデックスの問題の分析 502
- 実行時間 495
- 出力 498
- 出力メッセージ 518
- 出力例 498
- データベースの検証 494
- 内部インデックスの検査 496
- パフォーマンス 495
- 割り付けエラーの検出 506
- 割り付けの検査 497
- 割り付けの検証と修復 505
- 割り付けの修復 505, 507
- 割り付けの問題の分析 505
- DBCC_LOG_PROGRESS オプション 464, 498
- DBF 接続パラメータ
 - 埋め込みデータベース 65
- DBISQL
 - 概要 153
- dbisql
 - コマンド・ライン・パラメータ 63
 - 出力フォーマットの指定 268
 - 挿入、データ、対話的 302
 - トラブルシューティング 535
 - ログオン・ウィンドウ 64
 - 「Interactive SQL」参照 153
- dbisql Java 536
- DBKEY 接続パラメータ 120
- DBLOCATE ユーティリティ 105
- dblog ユーティリティ 475
- DBN 接続パラメータ 118
- DBN 通信パラメータ
 - 説明 136
- dbo ユーザ ID
 - 所有されるビュー 207
- dbspace
 - 作成の例 191
 - 使用状況の表示 193
 - 定義 156
 - ロー・デバイスへのリストア 470
- DBSpaceLogger イベント 542
- DBS 接続パラメータ 119

DB 領域

- DB 領域の不足状況 178, 537
- IQ_SYSTEM_MSG 159
- IQ_SYSTEM_TEMP 156
- SYSTEM 156
- インデックス情報の表示 193
- オフライン 506
- 管理の例 191
- サイズの変更 191
- 削除 190
- 作成 188
- 作成時のファイルのロケーション 170
- パーミッション 338
- 命名 188
- 読み込み専用 183
- 領域の使用状況のモニタリング 542
- 領域不足のエラー・メッセージ 537
- DDL
 - 説明 151
- DDL (データ定義言語) 9
- DDL ロック 410
- DEFAULT_DISK_STRIPING オプション 4
- DELETE パーミッション 351
- DELIMITED BY オプション 282
- Delphi
 - ODBC 設定 74
- DisableMultiRowFetch 接続パラメータ 121
- DML 166
- DML (データ操作言語) 9
- DMRF 接続パラメータ 121
- DOBROADCAST 通信パラメータ 137
- DROP JOIN INDEX 文 200
- DROP TABLE 文
 - の例 203
- DROP VIEW 文
 - 制限 207
 - 例 209
- DROP 文
 - オートコミット 399
 - 同時実行性の規則 410
- DSN 接続パラメータ 120
- 説明 69
- DSS (意志決定支援システム) 2
- DTTM インデックス
 - 推奨される使用法 237
 - 短所 241
 - 長所 241
 - 追加インデックス 242
 - 他のインデックスとの比較 242

索引

DumpAllThreads ファイル 530
dumpdups
 sp_iqcheckdb オプション 498
dumpleaks
 sp_iqcheckdb オプション 498
dumpunallocs
 sp_iqcheckdb オプション 498

E

EncryptedPassword 接続パラメータ 122
Encryption 接続パラメータ
 説明 122
ENC 接続パラメータ
 説明 122
EngineName 接続パラメータ 121
ENG 接続パラメータ 121
ENP 接続パラメータ 122
Excel フォーマット 267

F

FileDataSourceName 接続パラメータ 124
FileDSN
 作成 78
 接続パラメータ 69
 分散 78
FIPS サポート 370
FIXED フォーマット 267
FORCE_DROP オプション 516
FoxPro フォーマット 267
FP(1) インデックス 225
FP(2) インデックス 225
FP(3) インデックス 226
FROM 句
 UPDATE 文 331
 ジョイン・インデックス 245

G

getinfo スクリプト 567
gm スイッチ 32
 リカバリの影響 419, 494
GRANT 文
 DBA 権限 350
 RESOURCE 権限 350

WITH GRANT OPTION 353
 グループの作成 356
 グループ・メンバシップ 357
 新規ユーザ 348
 パーミッション 351
 パスワード 349
 パスワードなし 359
 プロシージャ 354

GROUP BY 句
 インデックス選択への影響 220
GROUP パーミッション
 継承不可能 356

H

HASH_THRASHING_PERCENT オプション 546
Health Insurance Portability and Accountability Act 488
HG インデックス
 クエリのパフォーマンス 232
 自動作成 232
 短所 231
 長所 231
 追加インデックス 232
 NULL 値 232
 NULL 値のマルチコラム 232
 外部キー制約 230
 推奨される使用法 230
 他のインデックスとの比較 231
High_Group インデックス
 「HG インデックス」参照
High_Non_Group インデックス
 「HNG インデックス」参照
HIPAA 488
HNG インデックス
 推奨される使用法 233
 他のインデックスとの比較 233
 短所 233
 長所 233
 追加インデックス 233
HOST 通信パラメータ 138

I

IANA
 ポート番号 145

- iAnywhere JDBC ドライバ 536
 - ASA データベースへの接続 57
 - ICU ライブラリ 426
 - IDLE 接続パラメータ 125
 - INSERT LOCATION 文 300
 - INSERT パーミッション 351
 - INSERT 文 244
 - VALUES オプション 298
 - インクリメンタル 299
 - 整合性 373
 - 説明 298
 - パフォーマンス 299
 - 分割されたテーブル 286
 - INT 接続パラメータ 126
 - integer データ型
 - Adaptive Server Enterprise と Sybase IQ の対応 322
 - Interactive SQL
 - 「dbisql」参照 153
 - Interfaces ファイル
 - エントリの追加 59
 - IPv6 アドレス 143
 - IPv6 のサポート 370
 - IPX
 - サーバ設定 132
 - IP アドレス 110
 - ping 553
 - IP 通信パラメータ 138
 - IQ PAGE SIZE 174
 - IQ UNIQUE
 - 値の変更 202
 - パフォーマンスの影響 201
 - IQ UNIQUE テーブル・オプション 201
 - IQ UNIQUE 制約 383
 - IQ UTILITIES
 - バッファ・キャッシュのモニタリング 530
 - iq_bcp
 - LOAD TABLE USING FILE に置換 286
 - 廃止 286
 - IQ_SYSTEM_MAIN
 - CREATE パーミッション 340
 - サイズのガイドライン 161
 - IQ_SYSTEM_MSG DB 領域 159
 - IQ_SYSTEM_TEMP
 - CREATE パーミッション 340
 - iqdemo データベース 7
 - IQDSEdit 59
 - iqdsn コマンド 72
 - iqgovern スイッチ 33
 - IQMsgMaxSize サーバ・プロパティ 12
 - IQMsgNumFiles サーバ・プロパティ 12
 - iqmsgnum サーバ起動スイッチ 12
 - iqmsgsz サーバ起動スイッチ 12
 - iqmsg メッセージ・ログ 9, 12
 - アーカイブ処理 13
 - 最大サイズの設定 13
 - ラッピング 13
 - iqmsg メッセージ・ログのアーカイブ処理 13
 - iqmsg ログ 12
 - アーカイブ処理 13
 - 最大サイズの設定 13
 - ラッピング 13
 - iqnumbercpus
 - サーバ・スイッチ 34
 - iqwmem スイッチ 33
 - ISO_1 照合
 - 説明 429
 - ISO_BINENG
 - デフォルトの照合 427, 437, 438
 - ISO_BINENG 照合 427
 - ISQL
 - 接続 59
- ## J
- Java
 - Sybase IQ での使用 53, 176
 - メモリ要件 33
 - jConnect ドライバ
 - IQ データベースへの接続 57
 - JDBC
 - 接続 52
 - 接続の設定 59
 - JDBC 接続
 - 説明 57
- ## K
- Kerberos
 - パーミッション 338
 - Kerberos 認証 370
- ## L
- LANalyzer 555
 - LANG 環境変数 436

索引

- LANG 接続パラメータ
 - 説明 126
 - LazyClose 接続パラメータ
 - 説明 127
 - LC_ALL 環境変数 436
 - LCLOSE 接続パラメータ
 - 説明 127
 - LDAP サーバ
 - LDAP 通信パラメータ 139
 - LDAP 通信パラメータ
 - 説明 139
 - LF インデックス 229
 - 推奨される使用方法 229
 - 他のインデックスとの比較 230
 - 短所 230
 - 長所 230
 - 追加インデックス 230
 - ユニークな値の最大数を越える 547
 - LF 通信パラメータ
 - 説明 140
 - Links 接続パラメータ 115
 - LivenessTimeout 接続パラメータ 127
 - LOAD TABLE
 - BINARY 291
 - LOAD TABLE 内 318
 - LOAD TABLE 文 244
 - QUOTES オプション 283
 - QUOTES オプションの例 284
 - QUOTES キーワード 284
 - STRIP ON 句 282
 - STRIP キーワード 284
 - USING CLIENT FILE 286
 - USING FILE 句 286
 - 整合性制約 280
 - 説明 279
 - バイナリ・フォーマット 291
 - バイナリ・フォーマット・データ・ファイル 291
 - パフォーマンス 312
 - 分割されたテーブル 286
 - LOB 301
 - LocalOnly 通信パラメータ
 - 説明 139
 - LOCAL 通信パラメータ
 - 説明 139
 - LogFile
 - 接続パラメータ 128
 - 通信パラメータ 140
 - LogFormat 通信パラメータ
 - 説明 140
 - LOGIN_MODE データベース・オプション
 - 統合化ログイン 96
 - LogMaxSize 通信パラメータ
 - 説明 114, 141
 - LogOptions 通信パラメータ
 - 説明 141
 - LOG 通信パラメータ
 - 説明 140
 - LOPT 通信パラメータ
 - 説明 141
 - Lotus フォーマット 267
 - Low_Fast インデックス
 - 「LF インデックス」参照
 - LSIZE 通信パラメータ
 - 説明 114, 141
 - LTO 接続パラメータ 127
- ## M
- max_days_since_login
 - 超過 344
 - max_failed_login_attempts
 - 超過 344
 - MAX_TEMP_SPACE_PER_CONNECTION オプション 178
 - MaxConnections 通信パラメータ
 - 説明 142
 - MAXCONN 通信パラメータ
 - 説明 142
 - MaxRequestSize 通信パラメータ
 - 説明 143
 - MAXSIZE 通信パラメータ
 - 説明 143
 - MESSAGE LOG
 - 内容 288
 - 例 290
 - metadata
 - カタログ・ストア内 158
 - Microsoft Access 546
 - ODBC 設定 73
 - Microsoft Visual Basic
 - ODBC 設定 73
 - Money データ型 324, 326
 - MYIP 通信パラメータ 143
 - MySybase
 - アクセス 571
 - オンライン・サポート 571

N

- National Language Support
 - 概要 425, 427
 - 照合 434
 - 説明 425
 - マルチバイト文字セット 430
- NDIS
 - ドライバ 552
- NEAREST_CENTURY オプション 317
- net.cfg ファイル 555
- NetBIOS
 - サーバ設定 132
- NetWare
 - ネットワーク・アダプタの設定 555
- NEWID
 - default 381
- NEWPWD 接続パラメータ 129
- NOT NULL 制約 373
- NOTIFY オプション
 - INSERT 文 247
- Novell クライアント・ソフトウェア 552
- NULL 318
 - default 382
 - 挿入 298
 - バックアップ 458
 - 変換 320
 - 変換オプション 308, 320
 - マルチカラム HG インデックス 232
- NULLS オプション
 - dbisql 270
- NULL 値
 - 出力 270
 - マルチカラム HG インデックス 232

O

- ODBC
 - UNIX 用の初期化ファイル 79
 - UNIX サポート 79
 - 接続パラメータ 109
 - データ・ソース 69
 - データ・ソースの設定 73
 - ドライバの位置 88
- ODBC 接続
 - 説明 57
- ODBC データ・ソース
 - jConnect の使用 57

- ODBC 変換ドライバ
 - ODBC 設定 73
- ODI ドライバ 552
- OEM コード・ページ
 - 選択 429
- ON 句のジョイン 256
- Open Database Connectivity。「ODBC」を参照
- OPERATOR 権限
 - 説明 337
 - 付与 350

P

- params.cfg ファイル 27
- Password 接続パラメータ 129
- PERMS ADMIN 権限
 - 説明 337
 - 付与 350
- PF(1) インデックス 225
- ping
 - TCP/IP 553
- PIPE_NOT_CONNECTED エラー 282
- PORT 通信パラメータ 145
- PreFetchOnOpen 通信パラメータ
 - 説明 144
- PROMPT 接続パラメータ 131
- PUBLIC グループ 359
- PWD 接続パラメータ 129

Q

- QUERY_TEMP_SPACE_LIMIT オプション 178
- QUOTES
 - LOAD TABLE キーワード 284

R

- RCVBUFSZ 通信パラメータ
 - 説明 144
- REAL データ型
 - Adaptive Server Enterprise と Sybase IQ の
データの対応 323
- ReceiveBufferSize 通信パラメータ
 - 説明 144
- REFERENCES パーミッション 351
- RELEASE SAVEPOINT 文 416

索引

resetclocks
 sp_iqcheckdb オプション 498
RESOURCE 権限
 継承不可能 356
 説明 338
 付与 350, 351
RESTORE 文
 COMPATIBLE 句 478
 VERIFY 句 456, 467, 478
 説明 468
 バックアップの検証 456, 467, 478
RetryConnectionTimeout 接続パラメータ 131
REVOKE 文
 説明 355
ROLLBACK TO SAVEPOINT 文 416
ROLLBACK 文 416
ROW LOG
 内容 289
 の例 290
RSA サポート 370

S

sa_get_request_profile
 要求ログ・ファイルの分析 566
sa_get_request_times
 要求ログ・ファイルの分析 566
SACHARSET 環境変数 435
SALANG 環境変数 435
SAVEPOINT 文
 トランザクション 416
SCSI テープ・バックアップ 445
SELECT DISTINCT 射影 220
SELECT パーミッション 351
SELECT 文
 INSERT 文内 302
 ジョイン・インデックス 245
 デリミタ 301
 ビュー作成の制限 207
SendBufferSize 通信パラメータ
 説明 145
ServerName 接続パラメータ 121
ServerPort 通信パラメータ 145
Service Manager
 サーバの起動 27
SESSIONS 通信パラメータ 146
SET 句
 UPDATE 文 330
SNDBUFSZ 通信パラメータ
 説明 145
SORT_COLLATION オプション 425
SORTKEY 関数 425, 426
sp_iqcheckdb
 allocation モード 497
 check モード 496
 DBCC_LOG_PROGRESS 464
 DBCC_LOG_PROGRESS オプション 498
 DBCC の機能 495
 dropleaks モード 498
 dumpdups オプション 498
 dumpleaks オプション 498
 dumpunallocs オプション 498
 resetclocks オプション 498
 verify モード 496
 アロケーション・マップのリセット 498
 インデックスと割り付けの検査 496
 インデックスの検証と修復 502
 インデックスの修復 502
 インデックスの問題の分析 502
 構文 495
 実行時間 495
 出力 498
 出力の解釈 506
 出力メッセージ 518
 出力例 498
 データベースの一貫性のチェック 463
 データベースの検証 494
 内部インデックスの検査 496
 パフォーマンス 495
 リソースの問題 501
 割り付けの検査 497
 割り付けの検証と修復 505
 割り付けの修復 505, 507
 割り付けの問題の分析 505
sp_iqcheckoptions ストアド・プロシージャ 178
sp_iqdbspace
 DB 領域の使用状況 193
sp_iqdbspaceinfo
 DB 領域の使用状況 194
sp_iquestdbspaces
 必要な DB 領域の見積もり 162
sp_iquestjoin 262
 ジョイン・インデックス領域の必要量の
 見積もり 162

- sp_iqestspace
 - データベース領域の必要量の見積もり 162
- sp_iqindexinfo
 - インデックス情報の表示 195
- sp_iqrebuildindex 505
- sp_iqstatus
 - 出力例 557
 - トラブルシューティングでの使用 556
- sp_iqtransaction
 - ブロックしている書き込みの確認 548
- SPACE ADMIN 権限
 - 説明 338
 - 付与 351
- SQLCONNECT 環境変数
 - 接続 69
- SQL ログ
 - 夏時間 16
- srvlog
 - 接続情報の関連付け 570
- srvlog ファイル
 - 接続情報 570
- START ROW ID オプション
 - INSERT 文 300
- start_iq
 - コマンドが実行されない 526
 - トラブルシューティング 526
 - パラメータ 526
- start_iq ユーティリティ 22
- StartLine 接続パラメータ 131
- Start パラメータ
 - 埋め込みデータベース 66
- STRIP
 - LOAD TABLE キーワード 284
- STRIP ON 句
 - 廃止 282
- STRIP オプション 282
- SUM 関数 229
- Sybase Central 50
 - DB 領域の作成 189
 - 概要 152
 - カラム制約 385
 - カラムのデフォルト 379
 - 起動 24, 60
 - グループの作成 357
 - グループへのユーザの追加 357
 - テーブルの作成 197
 - テーブルの変更 203
 - パーミッション 352, 354
 - ビューの削除 209
 - 表示の問題 544, 551
 - ユーザの作成 349
 - Sybase Central におけるカラムのデフォルトの編集 379
 - Sybase IQ
 - Adaptive Server Enterprise とデータ型を対応させる 321
 - Sybase IQ 12.5 サーバ 536
 - SYSCOLAUTH ビュー
 - パーミッション 369
 - SYSCOLUMN テーブル
 - 整合性 396
 - SYSDUMMY テーブル
 - パーミッション 369
 - SYSDUMMYPERM テーブル
 - リストア後の file_name 467
 - SYSPROCAUTH ビュー
 - パーミッション 369
 - SYSPROCPERM テーブル
 - パーミッション 369
 - SYSTABAUTH ビュー
 - パーミッション 369
 - SYSTABLE テーブル
 - 整合性 396
 - SYSTABLEPERM テーブル
 - パーミッション 369
 - SYSSUSERAUTH ビュー
 - パーミッション 369
 - SYSSUSERLIST ビュー
 - パーミッション 369
 - SYSSUSERPERMS ビュー
 - パーミッション 369
 - SYSSUSERPERM テーブル
 - パーミッション 369
 - SYSSVIEW ビュー
 - ビュー情報 209
 - SYS グループ 359

索引

T

TCP/IP

- BroadcastListener (BLISTENER) 通信パラメータ 134
- LDAP 通信パラメータ 139
- サーバ設定 132
- サーバのポート番号 145
- トラブルシューティング 553
- ファイアウォール経由の接続 135
- テスト 554

TDS 通信パラメータ 147

Telnet

- TCP/IP のテスト 554

TEMP 環境変数

- ディスク容量 103

TEMP_EXTRACT_NULL_AS_EMPTY オプション 275

TEMP_EXTRACT_NULL_AS_ZERO オプション 275

TEXT インデックス 242

- 作成 219

TIME インデックス 237

- 他のインデックスとの比較 242
- 短所 241
- 長所 241
- 追加インデックス 242

TIME データ型

- ロードの最適化 312

TIMEOUT 通信パラメータ 147

TIME インデックス

- 推奨される使用法 237

TMP 環境変数

- ディスク容量 103

TO 通信パラメータ 147

U

UID 接続パラメータ 132

Unconditional 接続パラメータ 132

UNC 接続パラメータ 132

Unicode

- 文字セット 426, 430

Unilib ライブラリ 426

UNIQUE 制約 383

UNIX

- ODBC サポート 79

UPDATE 文

- ジョイン操作の使用 331
- 使用 330

USER ADMIN 権限

- 説明 338

USER 権限

- 付与 351

Userid 接続パラメータ 132

USER 特別値

- default 380

util_db.ini ファイル 346

V

VALUES オプション

- INSERT 文 298

VARCHAR データ

- 長さ 0 のセル 315

VERIFY 通信パラメータ 148

Visual Basic

- ODBC 設定 73

VLDB

- 管理 17

W

WarehouseArchitect

- 説明 152

WD インデックス 235

- 推奨される使用法 235

- 短所 237

- 長所 237

WHERE 句

- ジョイン・インデックス 245

- UPDATE 文 331

- インデックス選択への影響 221

Windows サービス 22

WITH GRANT OPTION 句 353

WORD インデックス 235

WORM 記憶領域 489

X

XML フォーマット 267

Z

- zr ログ・ファイル 566
- Z オプション
 - データベース・サーバ 104

あ

- アドホック・ジョイン
 - パフォーマンス 242
- アロケーション・マップ
 - エラーの修復 506
 - 矛盾 510
 - リカバリ 509
 - 割り付けの検査 497
- 暗号化
 - Encryption (ENC) 接続パラメータ 122
 - FIPS 370
 - RSA 370
 - カラム 370
 - 強力な暗号 122
 - 通信 147
 - データベース 370
- 暗号化パスワード
 - ODBC 設定 75

い

- イーサネット 555
- 移行
 - 照合 426
 - ソート・キー 426
 - 廃止された照合 426
- 依存性
 - カーソル動作 423
- 一貫性の検査
 - マルチプレックス 495
- イベント
 - DBSpaceLogger 542
 - ディスク領域の使用状況のモニタリング 541
 - 領域の使用状況のモニタリング 542
- イベント・ハンドラ
 - ConnectFailed 344
- インタフェース・ライブラリ
 - 接続 52
- インデックス
 - default 225, 226
 - FP(1) 225

- FP(1) および FP(2) 225
- FP(2) 226
- FP(3) 226
- sp_iqcheckdb エラー 507, 515
- Sybase Central で作成 219
- TEXT インデックス 242
- TEXT インデックスの作成 219
- インデックス・タイプの選択 243
- 概要 209
- 検索 225, 226
- 検証 211
- 検証と修復 502
- 高速射影 225
- 再構築 202
- 削除 212
- 作成 196, 218
- システム・テーブル 211
- 自動作成 200
- 修復 505
- 修復不可能なエラー 515
- 説明 213
- 追加と削除 216
- ディスク領域の使用率 222
- テーブル上に多すぎる 546
- テーブルのロード後の追加 244
- 名前の変更 212
- 破損の削除 516
- フラット FP 228
- 並行作成 218, 219
- マルチカラム HG と NULL 232
- 矛盾 505
- ユニークな値の最大数 547
- 論理的な問題の検出 515
- インデックス・タイプ
 - LF 229
 - 説明 214
 - 選択 243
- インデックス・タイプ
 - 推奨 221
 - 選択の条件 219
- インポート、データ
 - LOAD TABLE 文 279
 - バージョン 12 より前の IQ データベース 300
 - Adaptive Server Enterprise から 300
 - 変換エラー 326
- 引用符
 - SQL 内 360

う

- 埋め込みデータベース 109, 113
 - Java 66
 - 起動 65
 - 接続 65
 - 接続パラメータ 86

え

- エラー
 - DB 領域の不足状況 537
 - 修復不可能 515
 - 挿入と削除 409
 - データ変換 326
 - トランザクション処理 409
- エラー・メッセージ
 - PIPE_NOT_CONNECTED 282
 - ファイルへのリダイレクト 270
 - 文字セット変換 431
- 円記号 (¥)
 - Windows ロー・デバイス 173
- 演算子
 - 作業 486
- エンティティ整合性
 - 確保 387
 - 説明 374

お

- オートコミット・モード
 - ODBC 74
- 大文字と小文字の区別
 - コマンド・ライン 27
 - 接続パラメータ 109
 - データベース名とサーバ名 31
 - パスワード 171, 438
- オブジェクト
 - 修飾名 360
- オプション
 - DBCC_LOG_PROGRESS 464, 498
 - MAX_TEMP_SPACE_PER_CONNECTION 178
 - QUERY_TEMP_SPACE_LIMIT 178
 - 設定 367
- オプション値
 - トランケーション 271, 272

か

- カーソル
 - ODBC 設定 74
 - 依存性 423
 - 位置付け操作 424
 - コマンドの構文 424
 - スクロール 423
 - 接続の制限 367
 - トランザクション 422
 - ホールド 406, 423
 - メッセージ・ロギング 424
 - 例 424
- 開始行
 - ODBC 設定 75
- 外部キー
 - オプション 392
 - 強制力のない 204
 - 強制力のない既存の外部キー 392
 - 作成 204
 - 制限 374
 - データの挿入 306
- 書き込みサーバ
 - 置き換え 498
- カスタム照合 426
- 仮想バックアップ
 - カプセル化仮想バックアップ 459
 - 分離仮想バックアップ 459
- カタログ・ストア
 - キャッシュ・サイズの設定 33
 - 説明 158
- カタログ・ファイル
 - 増加 539
- 活性
 - ODBC 設定 76
- 可変長データ転送 445
- カラム
 - 削除 202
 - 修復不可能なエラー 515
 - 追加 202
 - デフォルト 375
 - パーミッション 337
 - プロパティ 379
 - 変更 202
- カラム・デリミタ
 - ロード・フォーマット・オプション 282
- カラムのデフォルトの使用 375
- カラムのデフォルトの変更と削除 377

カラム幅
 挿入の問題 312
 カラム、ロード時に設定される 318
 環境変数
 LANG 436
 LC_ALL 436
 SACHARSET 435
 SALANG 435
 SQLCONNECT 69
 関数
 BFILE 301
 BLOB 用 301
 CLOB 用 301
 タイプ 9
 パーミッション 337

き
 キー・ジョイン 252, 255
 起動 50
 障害の解決 510
 チェックポイント・エラー 510
 トラブルシューティング 103
 トラブルシューティングのヒント 523
 割り付けエラー 510
 起動スクリプト 22
 起動パラメータ 27
 起動、マルチプレックス 24
 起動ユーティリティ 22
 キャッシュ
 書き込み 399
 キャッシュ・サイズ
 カタログ・ストアの設定 33
 強制リカバリ 510
 書き込みサーバの置き換え 498
 サーバの起動に失敗 510
 重複ブロックの検出 498
 手順 513
 複合的に所有されたブロックの検出 498
 割り付けられていないブロックの検出 498
 強制力のない外部キー 392
 強力な暗号
 Encryption (ENC) 接続パラメータ 122

く

クエリ
 DATEPART を使用 238
 スラッシング 546
 同時実行性の規則 409
 同時ユーザの制限 33
 パフォーマンス 232
 パフォーマンスの問題 546
 範囲述部 239
 クエリ・サーバ
 書き込みサーバの置き換え 498
 クエリのタイプ
 インデックス・タイプ 220
 クライアント
 直接のデータ・ロード 286
 クライアント側
 Encryption (ENC) 接続パラメータ 122
 クライアント・プロセス情報 110
 グループ
 PUBLIC 359
 Sybase Central 357
 SYS 359
 管理 356
 作成 356
 パーミッション 337, 340, 358
 パスワードなし 359
 メンバシップ 357
 グローバル・テンポラリ・テーブル
 説明 198

け

権限
 挿入と削除 268
 データベース・オブジェクトの定義 155
 言語サポート
 概要 425
 照合 434
 説明 425
 マルチバイト文字セット 430
 現在の日付と時刻のデフォルト 379
 検索インデックス 225, 226

索引

ク

- 高速射影インデックス 225
- 後続ブランク
 - 削除 278, 282
- 後続ブランクの削除 278, 282
- 候補キー
 - 制限 374
- コード・ページ
 - サポート対象 427
 - 説明 427
- コマンド・ファイル
 - dbisql 153
 - データベース・オブジェクトの作成 153
- コマンド・ライン・スイッチ 27
 - 必須 29
 - 表示 27

ク

- サーバ
 - CPU 使用率 529
 - Windows での起動 526
 - 応答なし 528, 529, 530
 - 起動時の自動開始 22
 - 起動に失敗 510
 - 再利用 44
 - 処理の停止 528
 - 接続 57
 - 停止の問題 531
 - 「データベース・サーバ」参照
 - デッドロック 531
 - トランザクション・ログ 524
 - 命名 29
 - ユニークな名前 525
 - ユニークなポート番号 525
 - リカバリ 493, 522
 - 領域の不足 529
- サーバ・オブジェクト
 - 追加 59
- サーバ起動スイッチ
 - iqmsgnum 12
 - iqmsgsz 12
- サーバの再利用 44
- サーバ・プロパティ
 - IQMsgMaxSize 12
 - IQMsgNumFiles 12

- サーバ名
 - ODBC 設定 75
- サーバ・ログ・ファイル 43
- サーベンス・オクスリー法 488
- 削除
 - カラムのデフォルト 377
 - ビュー 207
- 作成
 - カラムのデフォルト 376
- サブトランザクション
 - セーブポイント 416
- サポート・センタ
 - MySybase 571
 - オンライン・ヘルプ 571
 - チェックリスト 572
 - 問題の報告 567
- サポート・センタ用のチェックリスト 572
- 参照整合性
 - 確保 387
 - カラムのデフォルト 375
 - 強制力のない既存の外部キーによる適用 392
 - 宣言 389
 - パーミッション 268

シ

- 時刻データ型
 - Adaptive Server Enterprise と Sybase IQ のデータ対応 325
- システム DB 領域 156
- システム障害
 - リカバリ 532
- システム・テーブル 8
 - SYSDATE 467
 - インデックス 211
 - 説明 205
 - パーミッション 368
 - 表示 179
 - ユーザとグループ 368
- システムの無応答 528
- システム・ビュー
 - 整合性 396
 - パーミッション 368
- システム・レベルのバックアップ 460
- 指定
 - ドライバ 57
- 射影
 - 最適化 225

- シャットダウン
 - データベース 50
 - トラブルシューティング 103, 531
- 集合 229
- 修飾されたオブジェクト名 360
- 修復
 - allocation 505
 - インデックス 516
 - データベース 493, 522
 - テーブル 516
- 出力の分析 566
- 出力フォーマット
 - dbisql 268
- 出力リダイレクション 269, 270
- ジョイン
 - 更新 331
 - スター 257
 - パフォーマンスの影響 242
 - 複数テーブル 249
 - リニア 257
- ジョイン・インデックス
 - Sybase Central で作成 256
 - カラムの変更 203
 - 基礎となるテーブルの修正 261
 - 候補キー 374
 - サイズの見積もり 262
 - 作成 196, 250
 - ジョイン階層 246
 - ジョイン関係 252
 - 説明 245
 - 挿入 305
 - テーブル内のカラム 247
 - 同期 252
 - トップ・テーブルと関連テーブルの
 - ロー数の比率 263
 - ベース・テーブル 305
 - マルチプレックス 200
 - メリットの見積もり 263
- ジョイン階層 246
- ジョイン仮想テーブル 200
- ジョイン・カラム 252
- ジョイン関係
 - 指定 254
 - 定義 252
- ジョイン・タイプ 256
- 障害
 - システム 439
 - メディア 439
- 照合
 - ANSI 429
 - CP874toUTF8 ユーティリティ 426
 - CREATE DATABASE 文 426
 - default 427, 434, 437, 438
 - ISO_1 429
 - SORT_COLLATION オプション 425
 - SORTKEY 関数 425, 426
 - 移行 426
 - カスタム 426
 - 互換性の問題 437
 - このバージョンでの変更点 426
 - サポート対象 429
 - 推奨 429
 - 説明 425, 428, 429
 - 選択 434
 - ソート 428
 - 代替 429
 - データベースの作成 437, 438
 - 日本語での推奨 431
 - 廃止 426
 - 廃止された照合の移行 426
 - パフォーマンスの問題 438
 - 表示 429
 - 変更 437
 - マルチバイト 430
- 所有者
 - 説明 339
 - 役割 5
- シングルノード・モード 200
- シングルバイト文字セット
 - 説明 427
- 診断ツール 556
 - sa_server_option 563
 - sp_iqcheckdb 562
 - sp_iqcheckoptions 562
 - sp_iqconnection 563
 - sp_iqcontext 563
 - sp_iqdbstatistics 562
 - sp_iqstatus 556
 - サーバ起動オプションの確認 562
 - サーバ要求のロギング 563
 - 通信の問題 566
 - データベース・オプションの確認 562

す

- スキーマのロード
 - 推奨するデータベース・サイズ 161
- スキーマ
 - 作成 196
 - 変更 196
- スクロール
 - カーソル 423
- スター・ジョイン 257
- スタック・トレース
 - スレッドに対する生成 530
 - ロケーション 532
- ストアド・プロシージャ
 - Sybase IQ 7
 - 実行パーミッションの付与 365
 - 説明 7
- スナップショット・バージョン管理
 - 「バージョン管理」参照 397
- スラッシング
 - HASH_THRASHING_PERCENT オプション 546
- スレッド
 - スタック・トレースの生成 530
 - 不足 543
 - ロード時の使用 329

せ

- 整合性
 - 概要 371
 - カラムのデフォルト 375
 - 制約 373, 375
- 整合性制約違反
 - ログの例 288
- 製品サポート 567
- 制約
 - パフォーマンスへの影響 372
- 制約違反
 - ログの例 288
- セーブポイント
 - トランザクション内 416
- セキュリティ
 - Advanced Security オプション 370
 - Encryption (ENC) 接続パラメータ 122
 - FIPS サポート 370
 - IPv6 のサポート 370
 - Kerberos 認証 370

- RSA サポート 370
- カラムの暗号化 370
- 説明 333
- データベースの暗号化 370
- 統合化ログイン 99, 100
- ビュー 362
- プロシージャ 354, 362
- ログインの失敗 344

接続

- BroadcastListener (BLISTENER) 通信パラメータ 134
- iAnywhere JDBC ドライバ 57
- Interactive SQL 60, 93
- jConnect 57
- JDBC 52
- LDAP 通信パラメータ 139
- 埋め込みデータベース 65
- 外部ホスト上のデータベースへの接続 64
- 概要 52
- 確立 54
- 管理 342
- サーバによる確立方法 87
- 最大値 341
- 削除 102
- 削除に必要なパーミッション 337
- 制限 466, 510
- データ・ソースの使用 67
- デフォルト・データベース 93
- 同時ユーザの制限 32
- トラブルシューティング 103
- ファイアウォール 135
- 文字セット 432
- リモート 57
- 例 60
- ローカル・データベース 60
- ロギング 102
- 接続情報
 - IQ メッセージ・ファイル 570
 - srvlog ファイル 570
 - 要求ログ 570
- 接続の削除 102
- 接続パラメータ
 - CharSet 114
 - Encryption (ENC) 122
 - IDLE 125
 - LANG 126
 - LazyClose [LCLOSE] 127
 - 位置 89

- 埋め込みデータベース 86
- 大文字と小文字を区別しない 109
- 接続文字列 54
 - 説明 109
- データ・ソース 69
- デフォルト 68
- 矛盾 86
- 優先度 110
- 接続ハンドル 11, 562
- 接続プロファイル 59
- 接続名
 - ODBC 設定 77
- 接続文字列
 - 表現 54
 - 文字セット 432
- 切断 102
- 設定
 - ODBC データ・ソース 73
- 設定パラメータ
 - 上書き 528
- 設定ファイル
 - 使用 27
- ゼロ
 - NULL への変換 320
- 選択的リストア操作 338
- 選択、ドライバ
 - iAnywhere JDBC ドライバの使用 57
 - jConnect JDBC ドライバの使用 57

そ

- 挿入
 - Adaptive Server Enterprise データベースからの挿入 300
 - 概要 265
 - カラムのデフォルト 377
 - カラム幅の問題 312
 - 旧バージョンからの挿入 308
 - 指定したロー 299
 - ジョイン・インデックス・テーブル 305
 - 対話的 302
 - パフォーマンス 326
 - プライマリ・キー・カラムと外部キー・カラム 306
 - 別のデータベースからの挿入 300
 - 「データのロード」参照 308
 - 挿入変換オプション 308

- ソート・キー
 - 移行 426
- ソート
 - 照合 428
- ソート順
 - 照合 425
- ソフトウェアのリリース番号 11

ち

- チェックポイント
 - 間隔の調整 550
 - システム・リカバリ 419
 - 自動、明示的 415
 - 説明 415
 - パーミッション 337
 - リカバリ 415
- 中カッコ 301

つ

- 通信
 - Encryption (ENC) 接続パラメータ 122
 - トラブルシューティング 552
 - パラメータ 132
- 通信パラメータ
 - Broadcast (BCAST) 133
 - BroadcastListener 134
 - Certificate 134
 - Certificate_Password 135
 - ClientPort 135
 - DatabaseName (DBN) 136
 - DOBROADCAST 137
 - HOST 138
 - LDAP (LDAP) 139
 - LocalOnly (LOCAL) 139
 - LogMaxSize (LSIZE) 114, 141
 - LogOptions (LOPT) 141
 - MaxConnections 142
 - MYIP 143
 - PreFetchOnOpen 144
 - ReceiveBufferSize (RCVBUFSZ) 144
 - SendBufferSize (SNDBUFSZ) 145
 - ServerPort 145
 - SESSIONS 146
 - TDS 147
 - TIMEOUT 147, 148
 - 説明 109, 143

索引

通知カウント 218
通知メッセージ 558

て

定数式のデフォルト 382
ディスク
領域の使用状況のモニタリング 540, 541
領域の不足 528, 529, 537
ディスク・アレイ
WORM 489
ディスク容量
インデックス 222
節約 423
割り付け 188
ディスク領域の不足
対処法 528, 529, 537
領域の使用状況のモニタリング 540, 541
ディレクトリ・サービス・エディタ
「IQDSEEDIT」を参照
データ
インポート 265
エクスポート 265, 269
クライアント 286
削除 331
重複 371
抽出 270
重複 372
トランザクション 404
入出力フォーマット 267
無効 371
ロード 265
データ型
Adaptive Server Enterprise と Sybase IQ データ型の変換 322
Adaptive Server Enterprise の unichar 324
Adaptive Server Enterprise の unitext 324
Adaptive Server Enterprise の univarchar 324
FLOAT 323
integer 322
money 324, 326
REAL 323
Sybase IQ と Adaptive Server Enterprise の対応 321
テーブル作成での指定 198
変換 306
文字 324
ロード時の変換 308
データ整合性
概要 371
カラムのデフォルト 375
システム・テーブル内のルール 396
制約 375
データ操作言語 166
データ・ソース
jConnect の使用 57
ODBC 69
UNIX 79
コマンド・ラインから作成 72
接続 67
設定 73
説明 69
データ・ソースの説明
ODBC 73
データ・ソース名
ODBC 73
データ抽出
ASCII 274
binary 274
アクセスの制御 273
オプション 270
オプション・リスト 271
説明 270
バイナリ/スワップ 274
データ定義
作成 196
データ定義言語
説明 151
同時実行性の規則 410
データ・トランケーション・エラー 536
データのエクスポート
概要 265
説明 269
データの修正
パーミッション 268
データの抽出
オプション 270
オプション・リスト 271
説明 270
データのバルク・ロード
クライアント・データ 286
データのロード
Adaptive Server Enterprise のデータ 301
ASCII 変換オプション 314
エラー 547
概要 265

- カラムのデフォルト 377
- クライアント・データ 286
- 整合性制約違反 288
- 制約違反のロギング 288
- 通知メッセージ 558
- 同時実行性の規則 409
- 名前付きパイプ 282
- パフォーマンス 326, 546
- 必要な権限 268
- ファイルの指定 281
- フォーマット・オプション 282
- 変換エラー 326
- 変換オプション 308
- ラージ・オブジェクト 301
- 領域の使用状況のモニタリング 541
- 「挿入」参照 308
- データベース
 - Adaptive Server Anywhere 158
 - Database Consistency Checker 463
 - Interactive SQL による管理 153
 - アンロード 48
 - 一貫性のチェック 463
 - 移動 170
 - 埋め込み 109, 113
 - オブジェクトの作成 196
 - オブジェクトの使用 151
 - 起動 48
 - 起動時の命名 30
 - 起動パーミッション 49
 - 検証 463
 - サイズ 174
 - 削除 181
 - 作成 168
 - 作成と削除のためのパーミッション 346
 - 作成に必要な権限 155
 - 修復 493, 522
 - 初期化 168
 - 初期サイズ 161
 - 所有者の役割 5
 - スキーマの作成 196
 - セキュリティの概要 5
 - 設計 151
 - 接続 52, 57, 87
 - 設定の概要 153
 - 選択的リストア 338
 - 相対パス名 172
 - 大容量 17
 - 停止 50
 - デフォルトの特性 171
 - パーミッション 6, 35, 333
 - ファイルの移動 470
 - ファイル・ロケーション 170
 - ブロック・サイズ 176
 - ページ・サイズ 174
 - 命名 29
 - 文字セット 431
 - ユーティリティ 16
 - ユーティリティ・データベースを使用した作成 345
 - 読み取り専用のリストア操作 338
 - 領域の事前割り付け 156
 - ロケーションの選択 172
- データベース管理者
 - 定義 336
 - 「DBA」参照 336
- データベース・サーバ
 - [スタート]メニューからの起動 26
 - Windows サービス 27
 - Windows 上での起動 26
 - 起動 21
 - 起動時の命名 30
 - 起動の回避 113
 - 緊急時の停止 106
 - コマンド・ライン・スイッチ 27
 - 実行、コマンド・プロンプト 22
 - 接続 57
 - 説明 4
 - 停止 44, 48, 132
 - 名前スイッチ 29
 - リモート 57
- データベースのリストア
 - バックアップの検証 456, 467, 478
 - ファイル名の変更 470
- データベース・ファイル
 - ODBC 設定 75
- データベース・ファイル名の変更 470
- データベース名
 - ODBC 設定 75
- データベース・ユーティリティ 69
- データ変換エラー 536
- データ・リンク・レイヤ
 - トラブルシューティング 553
- テープ・デバイス
 - バックアップ用 450

索引

- テーブル
 - 新しい DB 領域への移動 340
 - キーの追加 204
 - グループ所有者 358
 - 削除 203
 - 作成 196, 197
 - 修飾名 358, 360
 - 修復不可能なエラー 515
 - ジョイン関係 252
 - 所有者 339
 - パーミッション 337, 338
 - 破損 516
 - 複数テーブルのジョイン 249
 - ブロックされたアクセス 548
 - ブロックされたアクセスの管理 548
 - 変更 202
 - 読み込み専用 183
 - ロード 279
 - ロック 408
 - 「システム・テーブル」の情報参照 8
 - テーブル・パーティション 164
 - テーブル領域
 - allocation 4
 - テーブル・レベルのバージョン管理
 - 説明 401
 - 「バージョン管理」参照 401
 - デッドロック
 - 解決 531
 - 検出 531
 - デバイス・タイプ
 - データベース 155
 - デフォルト
 - AUTOINCREMENT 381
 - NEWID 381
 - NULL 382
 - Sybase Central で作成 379
 - USER 特別値 380
 - カラム 375
 - 現在の日付と時刻 379
 - 作成 376
 - 接続パラメータ 68
 - 挿入 377
 - 定数式 382
 - 文字列と数値 382
 - ロード 377
 - デフォルト・インデックス
 - es 225
 - デフォルトの設定ファイル 27
 - デフォルトのログイン・ポリシー 342, 343
 - デモ・データベース 7
 - デリミタ
 - SELECT 文 301
 - テンポラリ記憶領域
 - 領域を節約するオプション 423
 - テンポラリ・ストア
 - 領域管理 178
 - テンポラリ・テーブル
 - 説明 198
 - バージョン管理 407
 - ロード 199
- ## と
- 同期
 - 説明 252
 - マルチプレックス 200
 - 統合化ログイン
 - ODBC 設定 74
 - オペレーティング・システム 95
 - 使用 98
 - デフォルト・ユーザ 101
 - ネットワーク面 101
 - パーミッション 338
 - 同時実行性
 - Sybase IQ 400
 - 挿入、削除、クエリ 409
 - データ定義 410
 - バックアップ 400, 448
 - 読み込みと書き込み 402
 - 動的結果セット 536
 - 独立性レベル 413
 - ODBC 設定 73
 - トップ・テーブル
 - サイズとパフォーマンス 250
 - ドライバ
 - iAnywhere JDBC ドライバ 57
 - jConnect JDBC ドライバ 57
 - ドライバに起因するエラー
 - ODBC 設定 74
 - トラブルシューティング 521
 - dbisql 535
 - Sybase Central 551
 - 一般的な問題 555
 - 起動、シャットダウン、接続 103

- サーバの運用 523
- サーバの起動 104
- 処理の問題 545
- データベース接続 87, 533
- 配線の問題 555
- プロトコル 552
- リソースの問題 537
- トランザクション
 - カーソル 422
 - 起動 398
 - サブトランザクションとセーブポイント 416
 - 終了 399
 - セーブポイント 416
 - 説明 397
 - 定義 397
 - リカバリ 418, 494
 - ロールバック 418
- トランザクション処理
 - 説明 397
- トランザクションのコミット
 - 読み込みトランザクションのタイミングの影響 405
- トランザクション・ログ
 - システム・リカバリ 419
 - 使用しないリカバリ 514
 - 名前の変更 475
- トレース
 - スレッドに対する生成 530

な

- 内部ビルド番号 11
- ナチュラル・ジョイン 256
- 夏時間
 - SQL ログ・ファイル 16
 - メッセージ・ログ・ファイル 16
- 名前付きパイプ 282
 - 抽出 279

に

- 日時インデックス
 - 「DTM インデックス」参照

ね

- ネットワーク・アダプタ
 - ドライバ 552
- ネットワーク通信
 - 起動のトラブルシューティング 103, 104
 - コマンド・ライン・スイッチ 132
- ネットワーク番号
 - IPX アドレス 137
- ネットワーク・プロトコル
 - ODBC 設定 76
 - トラブルシューティング 552

は

- バージョン管理
 - カーソル 422
 - システム・リカバリ 419
 - 説明 397, 401
 - テーブル・レベル 401
 - テンポラリ・テーブル 407
 - 独立性レベル 413
 - パフォーマンスの影響 420
 - リカバリ 494
- バージョン文字列 11
- パーティション
 - DDL 操作 165
 - 最大数 165
 - 読み込み専用 183
- パーティションあり 265
- パーティション・キー 164
- パーミッション
 - connect 348
 - DBA 権限 336
 - DB 領域管理 340
 - OPERATOR 権限 337
 - PERMS ADMIN 権限 337
 - RESOURCE 権限 338, 350
 - SPACE ADMIN 権限 338
 - Sybase Central 352, 354
 - USER ADMIN 権限 338
 - WITH GRANT OPTION 353
- 概要 333
- 管理 333
- グループ 340, 356, 358
- グループ・メンバシップ 357
- 継承 353, 356

索引

- 個別 347
- コマンド・ライン・スイッチ 35
- 種類 6
- テーブル 340, 351
- パスワード 349
- パスワードの付与 348
- ビュー 208, 351
- ビューでの INSERT と DELETE 364
- 付与権 353
- プロシージャ 354
- 矛盾 366
- リスト 368
- 配線
 - トラブルシューティング 555
- ハイパースレッディング
 - サーバ・スイッチ 34
- バグ
 - レポート 567
- バス名
 - データベース 172
- パスワード
 - ODBC 設定 75
 - 大文字と小文字の区別 171, 438
 - 確認 349
 - デフォルト 336
 - 変更 349
 - 有効期間 341
 - 有効期間の設定 342
 - ユーティリティ・データベース 346
 - ルール 349
 - 割り当て 338
 - 最小の長さ 349
- パスワードの確認 349
- バックアップ
 - .iqmsg ファイル 462
 - iqmsg ファイル 15
 - Linux 445
 - NULL 458
 - SAN を使用した仮想バックアップ 460
 - アテンド付き 446
 - アテンドなし 446, 481
 - インクリメンタル 485
 - インクリメンタル・バックアップの検証 478
 - エラーからのリカバリ 455
 - 確認 456, 467, 478
 - 仮想 450, 458
 - 高速バックアップ 460
 - サード・パーティ 457
 - システム・レベル 460
 - 使用メモリの増加 487
 - スケジューリング 485
 - 責任 486
 - 説明 439
 - 待機時間 454
 - 対象データ 442
 - テープ・デバイスの指定 (NT の場合) 451
 - デバイス 444, 450
 - 同時実行性 400
 - 同時実行性の問題 448
 - パーミッション 337
 - パフォーマンスについて 487
 - 必要な権限 446
 - ヘッダ・ファイルの表示 477
 - マルチプレックス 440
 - 満杯 485
 - メッセージ・ログ 15, 462
 - メッセージ・ログ・アーカイブ 15, 462
 - 読み取り専用ハードウェア 454
 - バックアップの検証 456, 467, 478
 - インクリメンタル 478
 - エラー・レポート 479
 - 進行状況レポート 479
- バックアップ・ログ
 - 説明 482
 - ロケーション 482
- バッファ・キャッシュ
 - IQ UTILITIES コマンド 530
 - モニタ 530
 - 領域の不足 528
- バッファ・サイズ
 - ODBC 設定 77
- パフォーマンス
 - アドホック・ジョイン 242
 - インデックス 209
 - クエリとロード 546
 - 制約の影響 372
 - 挿入 299
 - データのロード 326
 - バージョン管理の影響 420
 - フラット・ファイルからのロード 309
- バルク・コピー 267
- 範囲分割 165
- 定義 164

ひ

- 比較インデックス
 - 「CMP インデックス」参照
- 非構造化データ分析オプション 301
- 日付データ型
 - Adaptive Server Enterprise と Sybase IQ の
 - データの対応 325
- ビュー
 - SELECT 文の制限 207
 - 永久テーブルとの相違点 206
 - 削除 207, 209
 - 作成 206
 - 使用 207
 - 所有者 339
 - セキュリティ 362
 - 挿入と削除 207
 - パーミッション 208, 337, 338, 340
 - 変更 208
 - 編集 205
- ビルド番号 11

ふ

- ファイアウォール
 - BroadcastListener (BLISTENER) 通信パラメータ
 - 134
 - LDAP 通信パラメータ 139
 - 接続 135
- ファイル
 - 出力のリダイレクト 269, 270
- ファイル・サイズ
 - 制御 539
- フェッチ・オペレーション
 - 警告の抑制 74
- 複数のユーザ
 - Sybase Central で作成 349
 - グループに追加 357
 - 個別に作成 347
 - パーミッション 338
 - ログインの失敗 344
 - ロック 342
 - ロックアウト 344
 - ロック解除 344
- 複数のレコード・フェッチ
 - ODBC 設定 77
- 物理レイヤ
 - トラブルシューティング 555
- プライマリ・キー
 - AUTOINCREMENT 381
 - エンティティ整合性 387
 - 作成 204
 - データの挿入 306
 - マルチカラム 388
- フラット FP インデックス 228
- フラット・ファイル
 - ロード変換オプション 308
 - ロード元 279
- ブランク
 - NULL への変換 320
 - 後続ブランクの削除 278, 282
- フレーム・タイプ 555
- プロシージャ
 - 作成パーミッション 337, 338
 - システム 7
 - 所有者 339
 - セキュリティ 362
 - 動的結果セット 536
 - パーミッション 354
 - 「ストアド・プロシージャ」参照 7
- ブロック・サイズ 176
- ブロックされた書き込みアクセス
 - 競合の管理 548
 - ブロックしている書き込みの確認 548
- ブロックマップ 497
- ブロック・モード 445
- プロトコル
 - スイッチ 38
 - トラブルシューティング 552
- プロファイル
 - 接続 59
- 分割
 - 定義 164
- 分割されたテーブル
 - APPEND_LOAD オプション 286
 - INSERT 286
 - LOAD TABLE 286
- 分析
 - 非構造化データ分析オプション 301

索引

へ

- 並列 CREATE INDEX 218
- ページ・サイズ 174
 - カタログ 37
 - スイッチ 37
- ベース・テーブル 198
 - ジョイン・インデックス内 305
- 変換
 - Adaptive Server Enterprise と Sybase IQ 間 322
 - インポート時のエラー 326
 - 挿入オプション 308
 - 挿入時 306
- 変換オプション
 - DATE 315
 - DATETIME 317, 318
 - DATE フォーマットの指定 315
 - 長さ 0 のセルの置換 315
 - パフォーマンス 312
 - フラット・ファイルのロード 308
- 変換ドライバ
 - ODBC 設定 73
- 変更
 - カラムのデフォルト 377
 - 照合 437

ほ

- 包含インデックス
 - 「WD インデックス」参照
- ポート番号
 - UNIX での指定 64
 - データベース・サーバ 145
 - デフォルト 64
- ホールド・カーソル 406, 423

ま

- マニュアル
 - SQL Anywhere xix
 - Sybase IQ xvii
- マルチスレッド
 - ロード時 329
- マルチバイト文字セット
 - 使用 430

- マルチプレックス
 - 一貫性の検査 495
 - ジョイン・インデックスの作成 200
- マルチプレックス・サーバ
 - 停止 45
 - ロード・バランス 148
- マルチプレックス・データベース
 - 検証 464
 - バックアップ 440
 - リストア 463
- マルチプロセッサ・マシン
 - スイッチ 31

む

- 矛盾した状態 510
- 矛盾するインデックス
 - 修復 505

め

- メイン・ストア
 - 領域管理 178
- メッセージ
 - DB 領域の不足状況 537
 - ファイルへのリダイレクト 270
 - メッセージ・ログに記録 9
 - メモリの通知 559
- メッセージ・ファイル
 - 接続情報 570
- メッセージ・ロギング 9, 12
- メッセージ・ログ 9
 - IQ_SYSTEM_MSG DB 領域 159
 - アーカイブのバックアップ 15, 462
 - 夏時間 16
 - バックアップ 15, 462
- メッセージ・ログ管理 9, 12
- メモリ
 - カタログ・ストア・キャッシュ 33
 - 接続の制限 367
 - 連続メモリのプールの作成 33
- メモリ・メッセージ
 - ロード通知メッセージ 559

も

- 文字セット
 - ISO_BINENG 427
 - Unicode 426, 430
 - UTF8 427
 - コード化 425, 427
 - シングルバイト 427
 - 接続パラメータ 114
 - 説明 425
 - 選択 434
 - 変換 437
 - 変換の回避 433
 - マルチバイト 430
- 文字セット変換
 - エラー・メッセージ 431
 - オンとオフ 437
 - サーバの起動 437
 - 無効化 437
- 文字データ型
 - Adaptive Server Enterprise と Sybase IQ のデータの対応 324
 - Adaptive Server Enterprise の unichar 324
 - Adaptive Server Enterprise の unitext 324
 - Adaptive Server Enterprise の univarchar 324
- 文字列と数値のデフォルト 382
- 問題
 - レポート 567

ゆ

- ユーザ 342
- ユーザ ID
 - ODBC 設定 74
 - 管理 333
 - 削除 355
 - 作成 348
 - デフォルト 336
 - メッセージ・ログ 11
 - リスト 368
- ユーザ定義データ型
 - CHECK 条件 385
- ユーザ認証
 - パーミッション 338
- ユーザ名
- ユーティリティ
 - CP874toUTF8 426
 - トランザクション・ログ 475

- ユーティリティ・データベース
 - 起動 345
 - セキュリティ 345
 - 接続 347
 - 説明 16
 - データベース作成のためのパスワード 346
 - パスワードの設定 346
- ユーティリティ・プログラム
 - 廃止された iq_bcp 286
- ユーロ記号
 - 1252 LATIN1 照合 430

よ

- 要求ログ・ファイル 566
 - sa_get_request_profile の使用 566
 - sa_get_request_times の使用 566
 - 接続情報 570
- 要求ログ・レベル 570
- 読み取り専用ハードウェアの例 489
 - バックアップ 454
- 読み取り専用 DB 領域 182

ら

- ラージ・オブジェクト・データ 301

り

- リーク領域リカバリ 512
- リカバリ
 - 書き込みサーバの置き換え 498
 - 強制 510
 - サーバ 493, 522
 - システム 418, 493, 494
 - システム障害 532
 - 通常 494
 - データベースの検証 495
 - データベースの修復 495
 - 特殊なモード 509
 - トランザクション 418, 494
 - トランザクション・ログ 419
 - トランザクション・ログの紛失 514
 - バージョン管理 494

索引

- バックアップの紛失 514
- リーク領域 512
- リストア操作
 - エラーからのリカバリ 477
 - 説明 464
 - 正しい順序の確認 473
 - 他のユーザの除外 476
 - バックアップの検証 456, 467, 478
 - パフォーマンスについて 487
 - ヘッダ・ファイルの表示 477
 - 読み取り専用の選択的リストア 338
 - リストア後の SYSFILE 467
 - ロー・デバイスへのリストア操作 470
- リソース計画
 - iqnumbercpus スイッチ 34
- リダイレクト
 - ファイルへの出力 269, 270
- リモート・データ
 - バルク・ロード 286
 - ロード 286
- リモート・データ・アクセス
 - プロキシ・テーブル 300
- 領域管理
 - DB 領域の不足状況 178, 537
 - IQ テンポラリ・ストア 178
 - IQ メイン・ストア 178
 - 領域待ちの状況 178, 537

る

- ルータ
 - ブロードキャスト 137

れ

- 連結メモリ
 - iqwmem スイッチの設定 33

ろ

- ローカル・テンポラリ・テーブル
 - 説明 198
- ロー・デバイス 155
 - Windows 上での名前の指定 173
 - ユーティリティ・データベース 345
 - ロー・デバイスへのリストア 470

- ロード
 - スレッド使用 329
- ロード・オプション 285
- ロードの最適化 312
- ロード・パフォーマンス
 - ロック競合 412
- ロード変換
 - 「変換オプション」参照
- ロールバック
 - DB 領域の不足状況 178, 537
- ロギング、接続 102
- ロゲイン
 - 制限 342
 - 統合化 95, 96
- ロゲイン管理
 - プロシージャのリスト 343
- ロゲイン試行
 - 制限の超過 344
- ロゲインの失敗 344
- ロゲイン・ポリシー 341
 - パーミッション 338
 - リセット 344
 - ロックのオプション 344
- ロゲイン・ポリシーのリセット 344
- ログ・ファイル
 - サーバ 43
 - 接続情報の関連付け 570
- ロケール
 - INSERT...LOCATION 436
 - 設定 435
 - 説明 425, 428
 - 判断 435
 - 文字セット 431
- ロック
 - DDL 操作 410
 - 競合の管理 548
 - テーブル 408
- ロックアウト
 - 自動 344
- ロック競合
 - 管理 412

わ

- 割り付け
 - 検証と修復 505