

## システム管理ガイド：第 1 巻

### **Sybase IQ**

15.1

ドキュメント ID : DC00284-01-1510-01

改訂 : 7 月 2009

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

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

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

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

Sybase の商標は、**Sybase の商標リスト** (<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その他の社名および製品名は、当該各社の商標または登録商標である可能性があります。

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

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

# 目次

|           |      |
|-----------|------|
| はじめに..... | xvii |
|-----------|------|

|              |                                  |          |
|--------------|----------------------------------|----------|
| <b>第 1 章</b> | <b>Sybase IQ システム管理の概要 .....</b> | <b>1</b> |
|              | Sybase IQ の概要.....               | 2        |
|              | 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 .....           | 5        |
|              | SQL Anywhere と Sybase IQ.....    | 5        |
|              | セキュリティの概要 .....                  | 6        |
|              | ユーザのタイプ .....                    | 6        |
|              | パーミッションの付与 .....                 | 7        |
|              | 同時処理.....                        | 7        |
|              | マルチプレックス機能 .....                 | 8        |
|              | ストアド・プロシージャ .....                | 8        |
|              | Sybase IQ ストアド・プロシージャ .....      | 8        |
|              | カタログ・ストアド・プロシージャ .....           | 9        |
|              | システム・テーブルとシステム・ビュー .....         | 9        |
|              | コマンドと関数 .....                    | 10       |
|              | SQL 文のタイプ .....                  | 10       |
|              | 関数 .....                         | 11       |
|              | メッセージ・ロギング .....                 | 11       |
|              | メッセージ・ログ内のバージョン情報 .....          | 12       |
|              | メッセージ・ログ内の照合情報 .....             | 13       |
|              | メッセージ・ログ内の接続情報 .....             | 13       |
|              | メッセージ・ログ内の挿入通知 .....             | 14       |
|              | メッセージ・ログ・ファイルの管理 .....           | 14       |
|              | ユーティリティ・データベース .....             | 19       |
|              | 大容量データベースの管理 .....               | 19       |
|              | メモリ使用の管理 .....                   | 20       |
|              | データ・ロードの管理 .....                 | 20       |
|              | 処理スレッドの管理 .....                  | 21       |
|              | ディスク領域の管理 .....                  | 21       |
|              | 中間バージョン管理 .....                  | 22       |

|              |                                      |           |
|--------------|--------------------------------------|-----------|
|              | データベースの作成 .....                      | 22        |
|              | インデックスの作成 .....                      | 23        |
|              | クエリの最適化 .....                        | 23        |
|              | スキーマ設計 .....                         | 23        |
|              | UNION ALL ビュー .....                  | 24        |
| <b>第 2 章</b> | <b>Sybase IQ の実行 .....</b>           | <b>25</b> |
|              | データベース・サーバの起動 .....                  | 25        |
|              | 起動ユーティリティによるサーバの起動 .....             | 26        |
|              | Sybase Central からのサーバの起動 .....       | 29        |
|              | Windows の [スタート] メニューからのサーバの起動 ..... | 32        |
|              | Windows ホストが再起動する場合のサーバの再起動 .....    | 32        |
|              | Windows サービスとしてのサーバの実行 .....         | 32        |
|              | コマンド・ライン・スイッチの使用 .....               | 32        |
|              | サーバとデータベースの命名 .....                  | 35        |
|              | コマンド・ラインからのパフォーマンスとメモリの制御 .....      | 37        |
|              | コマンド・ラインからのパーミッションの制御 .....          | 41        |
|              | 最大カタログ・ページ・サイズの設定 .....              | 43        |
|              | クライアント／サーバ環境の設定 .....                | 44        |
|              | 強制リカバリ・モードでサーバを起動する .....            | 46        |
|              | DBISQL からのサーバの起動 .....               | 46        |
|              | 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 .....      | 47        |
|              | サーバ・アクティビティのモニタ .....                | 48        |
|              | データベース・サーバの停止 .....                  | 51        |
|              | サーバを停止および再起動する状況 .....               | 51        |
|              | サーバを停止する方法 .....                     | 52        |
|              | サーバを停止できるユーザ .....                   | 55        |
|              | オペレーティング・システム・セッションの停止 .....         | 55        |
|              | データベースの開始と停止 .....                   | 56        |
|              | データベースの起動に関するガイドライン .....            | 57        |
|              | データベースの停止 .....                      | 58        |
|              | iqdemo データベースの起動 .....               | 58        |
|              | Sybase Central の開始と停止 .....          | 59        |
| <b>第 3 章</b> | <b>Sybase IQ の接続 .....</b>           | <b>61</b> |
|              | 接続の概要 .....                          | 62        |
|              | 接続の確立方法 .....                        | 62        |
|              | 接続パラメータの構造 .....                     | 64        |
|              | 接続文字列として渡される接続パラメータ .....            | 65        |
|              | ODBC データ・ソースへの接続パラメータの保存 .....       | 65        |

|                                                         |     |
|---------------------------------------------------------|-----|
| Sybase Central または Interactive SQL からの接続 .....          | 66  |
| [接続] ダイアログの表示 .....                                     | 67  |
| 接続用のドライバの指定 .....                                       | 69  |
| [接続] ダイアログ・ボックスの使用 .....                                | 69  |
| Sybase Central での接続を簡略化する方法 .....                       | 70  |
| サーバ・オブジェクトの作成 .....                                     | 70  |
| 接続プロファイルの作成 .....                                       | 70  |
| 簡単な接続の例 .....                                           | 71  |
| Sybase Central または DBISQL からサンプル・データベースに接続する .....      | 71  |
| 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 ..... | 73  |
| DBISQL から他のデータベースに接続する .....                            | 74  |
| コマンド・ライン・ユーティリティを使用して接続する .....                         | 75  |
| 埋め込みデータベースへの接続 .....                                    | 78  |
| データ・ソースを使用した接続 .....                                    | 79  |
| デフォルト接続パラメータの使用 .....                                   | 80  |
| Sybase IQ ユーティリティからの接続 .....                            | 81  |
| ODBC データ・ソースの使用 .....                                   | 82  |
| データ・ソースの保存場所 .....                                      | 82  |
| ODBC データ・ソースの作成と編集 .....                                | 83  |
| ODBC データ・ソースの設定 .....                                   | 86  |
| ファイル・データ・ソースの使用 .....                                   | 92  |
| UNIX での ODBC データ・ソースの使用 .....                           | 93  |
| OLE DB を使用してデータベースに接続する .....                           | 95  |
| OLE DB プロバイダ .....                                      | 96  |
| ADO からの接続 .....                                         | 96  |
| 接続パラメータ .....                                           | 97  |
| 接続パラメータについてのヒント .....                                   | 97  |
| Sybase IQ の接続の確立方法 .....                                | 98  |
| 接続を確立する手順 .....                                         | 99  |
| インタフェース・ライブラリの検出 .....                                  | 100 |
| 接続パラメータ・リストのアセンブル .....                                 | 101 |
| サーバの検出 .....                                            | 103 |
| データベースの検出 .....                                         | 105 |
| Interactive SQL 接続 .....                                | 106 |
| 他のデータベースからの接続 .....                                     | 106 |
| UNIX システムでのポート番号の競合防止 .....                             | 107 |
| サーバを検出できるかどうかのテスト .....                                 | 108 |
| 統合化ログインの使用 .....                                        | 108 |
| 統合化ログインの使用法 .....                                       | 109 |
| 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 .....                    | 113 |

|                                             |     |
|---------------------------------------------|-----|
| 一時的に public オプションを設定してセキュリティを<br>追加する ..... | 115 |
| ネットワークから見た統合化ログイン .....                     | 115 |
| デフォルトの統合化ログイン・ユーザの作成 .....                  | 116 |
| 接続の切断と削除 .....                              | 117 |
| 接続ログ .....                                  | 117 |
| 起動のトラブルシューティング、シャットダウン、接続 .....             | 118 |
| Sybase IQ を起動できない .....                     | 118 |
| データベースに接続できない .....                         | 120 |
| 緊急時のデータベース・サーバの停止 (UNIX) .....              | 122 |
| UNIX kill コマンドによる DBISQL の停止 .....          | 122 |
| UNIX での DBISQL ウィンドウの問題の解決 .....            | 123 |

|              |                                             |            |
|--------------|---------------------------------------------|------------|
| <b>第 4 章</b> | <b>接続パラメータと通信パラメータ .....</b>                | <b>125</b> |
|              | 接続パラメータ .....                               | 125        |
|              | AppInfo 接続パラメータ (App) .....                 | 127        |
|              | AutoPreCommit 接続パラメータ (AutoPreCommit) ..... | 129        |
|              | AutoStart 接続パラメータ (Astart) .....            | 129        |
|              | AutoStop 接続パラメータ (Astop) .....              | 130        |
|              | CharSet 接続パラメータ (CS) .....                  | 130        |
|              | CommBufferSize 接続パラメータ (CBSIZE) .....       | 131        |
|              | CommLinks 接続パラメータ (Links) .....             | 132        |
|              | ConnectionName 接続パラメータ (CON) .....          | 134        |
|              | DatabaseFile 接続パラメータ (DBF) .....            | 134        |
|              | DatabaseName 接続パラメータ (DBN) .....            | 136        |
|              | DatabaseSwitches 接続パラメータ (DBS) .....        | 137        |
|              | DataSourceName 接続パラメータ (DSN) .....          | 137        |
|              | DBKEY 接続パラメータ (DBKEY) .....                 | 138        |
|              | DisableMultiRowFetch 接続パラメータ (DMRF) .....   | 138        |
|              | EngineName 接続パラメータ (ENG) .....              | 139        |
|              | EncryptedPassword 接続パラメータ (ENP) .....       | 140        |
|              | Encryption 接続パラメータ (ENC) .....              | 140        |
|              | FileDataSourceName 接続パラメータ (FileDSN) .....  | 142        |
|              | Idle 接続パラメータ (IDLE) .....                   | 143        |
|              | Integrated 接続パラメータ (INT) .....              | 144        |
|              | Language 接続パラメータ (LANG) .....               | 144        |
|              | LazyClose 接続パラメータ (LCLOSE) .....            | 145        |
|              | LivenessTimeout 接続パラメータ (LTO) .....         | 146        |
|              | LogFile 接続パラメータ (LOG) .....                 | 147        |
|              | NewPassword 接続パラメータ (NEWPWD) .....          | 147        |
|              | Password 接続パラメータ (PWD) .....                | 148        |
|              | PrefetchBuffer 接続パラメータ (PBUF) .....         | 148        |
|              | PrefetchRows 接続パラメータ (PROWS) .....          | 149        |
|              | Prompt 接続パラメータ (PROMPT) .....               | 150        |

|                                            |     |
|--------------------------------------------|-----|
| ServerName 接続パラメータ (ENG).....              | 150 |
| StartLine 接続パラメータ (START).....             | 150 |
| Unconditional 接続パラメータ (UNC) .....          | 151 |
| Userid 接続パラメータ (UID) .....                 | 151 |
| ネットワーク通信のパラメータ .....                       | 152 |
| Broadcast 通信パラメータ (BCAST).....             | 153 |
| BroadcastListener 通信パラメータ (BLISTENER)..... | 153 |
| Certificate 通信パラメータ .....                  | 154 |
| Certificate_Password 通信パラメータ .....         | 154 |
| ClientPort パラメータ (CPort).....              | 155 |
| DatabaseName 通信パラメータ (DBN) .....           | 156 |
| DoBroadcast パラメータ (DBROAD).....            | 156 |
| DLL パラメータ .....                            | 158 |
| Host パラメータ (IP).....                       | 158 |
| LDAP 通信パラメータ (LDAP).....                   | 159 |
| LocalOnly 通信パラメータ (LOCAL).....             | 160 |
| Logfile 接続パラメータ (LOG).....                 | 160 |
| LogFormat 通信パラメータ (LF) .....               | 161 |
| LogMaxSize 通信パラメータ (LSIZE) .....           | 162 |
| LogOptions 通信パラメータ (LOPT) .....            | 162 |
| MaxConnections 通信パラメータ (MAXCONN).....      | 163 |
| MaxRequestSize 通信パラメータ (MAXSIZE).....      | 164 |
| MyIP 通信パラメータ (ME) .....                    | 164 |
| PreFetchOnOpen 通信パラメータ .....               | 165 |
| ReceiveBufferSize 通信パラメータ (RCVBUFSZ).....  | 166 |
| SendBufferSize 通信パラメータ (SNDBUFSZ) .....    | 166 |
| ServerPort パラメータ (PORT).....               | 166 |
| Sessions パラメータ .....                       | 168 |
| TDS パラメータ .....                            | 168 |
| Timeout パラメータ (TO).....                    | 168 |
| VerifyServerName 通信パラメータ (Verify) .....    | 169 |

|              |                                     |            |
|--------------|-------------------------------------|------------|
| <b>第 5 章</b> | <b>データベース・オブジェクトの使用 .....</b>       | <b>171</b> |
|              | Sybase IQ データベースの作成 .....           | 172        |
|              | データベースの設計 .....                     | 172        |
|              | データベース・オブジェクトの編集ツール .....           | 173        |
|              | データベース設定手順の概要 .....                 | 174        |
|              | データ定義権限の拡張 .....                    | 176        |
|              | デバイス・タイプの選択 .....                   | 177        |
|              | データベース用の領域の割り付け .....               | 177        |
|              | 範囲分割 .....                          | 186        |
|              | データベース・オブジェクトの使用 .....              | 191        |
|              | SQL によるデータベースの作成 .....              | 192        |
|              | IQ メイン・ストアと IQ テンポラリ・ストアの領域管理 ..... | 202        |

|                                         |     |
|-----------------------------------------|-----|
| データベース・オプションの設定 .....                   | 203 |
| データベース内のシステム・オブジェクトの表示 .....            | 204 |
| データベースからの切断 .....                       | 205 |
| データベースの削除 .....                         | 206 |
| DB 領域の使用 .....                          | 206 |
| dbfile の属性と操作 .....                     | 207 |
| DB 領域の属性と操作 .....                       | 207 |
| 読み取り専用および読み込み／書き込み DB 領域と<br>ファイル ..... | 207 |
| DB 領域の命名 .....                          | 212 |
| DB 領域の追加 .....                          | 213 |
| DB 領域の削除 .....                          | 215 |
| DB 領域管理の例 .....                         | 216 |
| テーブルの編集 .....                           | 221 |
| テーブルの作成 .....                           | 221 |
| テーブルの変更 .....                           | 228 |
| テーブルの削除 .....                           | 229 |
| プライマリ・キーと外部キーの作成 .....                  | 229 |
| システム・ビューに含まれるテーブル情報 .....               | 231 |
| ビューの編集 .....                            | 231 |
| ビューの作成 .....                            | 232 |
| ビューの使用 .....                            | 233 |
| ビューの修正 .....                            | 234 |
| ビューに対するパーミッション .....                    | 234 |
| ビューの削除 .....                            | 235 |
| システム・ビューに含まれるビュー情報 .....                | 235 |
| インデックスの編集 .....                         | 236 |
| インデックスの概要 .....                         | 236 |
| インデックスの作成 .....                         | 237 |
| システム・ビューに含まれるインデックス情報 .....             | 237 |
| インデックスの検証 .....                         | 238 |
| インデックスの名前変更 .....                       | 239 |
| インデックスの削除 .....                         | 239 |

## 第 6 章

|                                     |     |
|-------------------------------------|-----|
| Sybase IQ インデックスの使用 .....           | 241 |
| インデックスの概要 .....                     | 241 |
| Sybase IQ インデックス・タイプ .....          | 242 |
| 従来のインデックスに勝る利点 .....                | 246 |
| Sybase IQ インデックスの作成 .....           | 246 |
| CREATE INDEX 文 .....                | 246 |
| Sybase Central を使用したインデックスの作成 ..... | 247 |
| 複数のインデックスの同時作成 .....                | 247 |



|                                                    |     |
|----------------------------------------------------|-----|
| インデックス・タイプの選択 .....                                | 248 |
| インデックス内のユニークな値の数 .....                             | 249 |
| クエリのタイプ .....                                      | 249 |
| インデックス条件：ディスク領域の使用率 .....                          | 251 |
| インデックス内のデータ型 .....                                 | 252 |
| インデックス・タイプの組み合わせ .....                             | 253 |
| Sybase IQ インデックス・タイプ .....                         | 254 |
| 高速射影 (FP) デフォルト・インデックス・タイプ .....                   | 254 |
| Low_Fast (LF) インデックス・タイプ .....                     | 259 |
| High_Group (HG) インデックス・タイプ .....                   | 261 |
| High_Non_Group (HNG) インデックス・タイプ .....              | 263 |
| 比較 (CMP) インデックス・タイプ .....                          | 265 |
| 包含 (WD) インデックス・タイプ .....                           | 266 |
| 日付 (DATE)、時刻 (TIME)、日時 (DTTM) インデックス・<br>タイプ ..... | 269 |
| アドホック・ジョインのパフォーマンスの最適化 .....                       | 274 |
| インデックスの選択 .....                                    | 275 |
| データ挿入後のカラム・インデックスの追加 .....                         | 276 |
| ジョイン・インデックスの使用 .....                               | 277 |
| ジョイン・インデックスによって向上するクエリの<br>パフォーマンス .....           | 277 |
| ジョイン・インデックスのロード時の考慮事項 .....                        | 277 |
| クエリでのジョイン・インデックスの使用法 .....                         | 277 |
| ジョイン・インデックス間の関係 .....                              | 278 |
| ジョインがアドホックになるとき .....                              | 278 |
| ジョイン階層の概要 .....                                    | 278 |
| ジョイン・インデックス内のカラム .....                             | 279 |
| クエリ解析におけるジョイン階層 .....                              | 280 |
| 複数テーブルのジョインとパフォーマンス .....                          | 282 |
| ジョイン・インデックスの作成手順 .....                             | 283 |
| ジョイン・インデックスの同期 .....                               | 285 |
| テーブル間のジョイン関係の定義 .....                              | 286 |
| CREATE JOIN INDEX 文の実行 .....                       | 289 |
| Sybas Central でのジョイン・インデックスの作成 .....               | 290 |
| ジョイン階層のタイプ .....                                   | 290 |
| ジョイン・インデックスに含まれるテーブルの修正 .....                      | 294 |
| ジョイン・インデックス内のテーブルでの挿入と削除 .....                     | 295 |
| 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<br>のアクセス制御 .....       | 296 |
| ジョイン・インデックスのサイズとメリットの見積もり .....                    | 297 |
| sp_iqestjoin によるジョイン・インデックス・サイズの<br>見積もり .....     | 297 |
| ロー数を比較したジョイン・インデックスのメリットの<br>見積もり .....            | 298 |

|              |                                             |            |
|--------------|---------------------------------------------|------------|
| <b>第 7 章</b> | <b>データベースへのデータの入出力</b>                      | <b>299</b> |
|              | インポートとエクスポートの概要                             | 300        |
|              | インポートとエクスポートの方法                             | 300        |
|              | データの入出力フォーマット                               | 301        |
|              | データ修正のパーミッション                               | 302        |
|              | データベース更新のスケジュール                             | 303        |
|              | データベースからのデータのエクスポート                         | 303        |
|              | 出力先リダイレクションの使用                              | 304        |
|              | データ抽出オプション                                  | 305        |
|              | LOAD TABLE 文を使ったデータのバルク・ロード                 | 316        |
|              | クライアントからの直接のデータ・ロード                         | 323        |
|              | 分割されたテーブルのロード                               | 323        |
|              | メッセージ・ロギングの制御                               | 325        |
|              | 整合性制約違反のロギング                                | 326        |
|              | INSERT 文の使用                                 | 330        |
|              | 指定した値のロー単位の挿入                               | 330        |
|              | データベースからの指定したローの挿入                          | 331        |
|              | 別のデータベースからの挿入                               | 332        |
|              | データの対話的インポート                                | 335        |
|              | 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               | 336        |
|              | ジョイン・インデックスのテーブルへの挿入                        | 338        |
|              | プライマリ・キー・カラムと外部キー・カラムへの挿入                   | 340        |
|              | 部分幅挿入                                       | 340        |
|              | 部分幅挿入の規則                                    | 342        |
|              | 挿入時のデータ変換                                   | 346        |
|              | 15 より前のバージョンの Sybase IQ からのデータの挿入           | 348        |
|              | ロード変換オプション                                  | 348        |
|              | カラム幅の問題                                     | 352        |
|              | 日付と時刻のロードの最適化                               | 353        |
|              | LOAD TABLE の ASCII 変換オプションの使用               | 354        |
|              | DATE オプション                                  | 356        |
|              | DATETIME 変換オプション                            | 358        |
|              | NULL の処理                                    | 361        |
|              | データ表示に関連する他の項目                              | 362        |
|              | Adaptive Server Enterprise とのデータ型の対応        | 363        |
|              | サポートされていない Adaptive Server Enterprise のデータ型 | 363        |
|              | Adaptive Server Enterprise データ型の等価性         | 364        |
|              | データ・インポート時の変換エラーの処理                         | 369        |
|              | データのバルク・ロードのチューニング                          | 369        |
|              | 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             | 369        |
|              | サーバ起動オプションの設定                               | 370        |

|                                           |            |
|-------------------------------------------|------------|
| ロード時の環境の調整 .....                          | 371        |
| インクリメンタル・ロード時のメイン IQ ストア領域使用<br>の低減 ..... | 373        |
| ロード時のスレッド使用についての理解 .....                  | 373        |
| UPDATE によるデータの更新 .....                    | 374        |
| データの削除 .....                              | 375        |
| <b>第 8 章 ユーザ ID とパーミッションの管理 .....</b>     | <b>377</b> |
| データベース・パーミッションの概要 .....                   | 377        |
| DBA 権限の概要 .....                           | 378        |
| RESOURCE 権限の概要 .....                      | 380        |
| 所有権パーミッションの概要 .....                       | 380        |
| DB 領域管理パーミッション .....                      | 381        |
| テーブルとビューのパーミッションの概要 .....                 | 381        |
| グループ・パーミッションの概要 .....                     | 382        |
| サーバのコマンド・ラインのパーミッション・オプション .....          | 382        |
| ログイン管理 .....                              | 383        |
| IQ のユーザ・アカウントと接続の管理 .....                 | 384        |
| ログイン要求失敗後の接続の阻止 .....                     | 385        |
| ユーザのロックアウト .....                          | 386        |
| ユーザのロックアウト解除 .....                        | 386        |
| ユーティリティ・データベース・サーバのセキュリティ .....           | 387        |
| ユーティリティ・データベースのパスワードの定義 .....             | 388        |
| ファイル管理文を実行するためのパーミッション .....              | 388        |
| 個別のユーザ ID とパーミッションの管理 .....               | 390        |
| 新規ユーザの作成 .....                            | 390        |
| パスワードの変更 .....                            | 391        |
| DBA および RESOURCE 権限の付与 .....              | 392        |
| テーブルおよびビューに対するパーミッションの付与 .....            | 393        |
| パーミッションを付与する権利をユーザに付与する .....             | 396        |
| プロシージャに対するパーミッションの付与 .....                | 396        |
| ユーザ・パーミッションの取り消し .....                    | 397        |
| グループの管理 .....                             | 398        |
| グループの作成 .....                             | 399        |
| ユーザにグループのメンバシップを付与する .....                | 400        |
| グループのパーミッション .....                        | 401        |
| グループが所有するテーブルの参照 .....                    | 402        |
| パスワードのないグループ .....                        | 402        |
| 特殊なグループ .....                             | 403        |
| データベース・オブジェクトの名前とプレフィクス .....             | 404        |
| 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<br>使い方 ..... | 406        |
| セキュリティを調整するためにビューを使用する .....              | 407        |
| プロシージャを使用したセキュリティの調整 .....                | 408        |

|                                            |            |
|--------------------------------------------|------------|
| プロシージャを使用して接続を無効にする .....                  | 409        |
| ユーザ・パーミッションの評価方法 .....                     | 409        |
| リソース接続使用の管理 .....                          | 410        |
| ユーザ・リソースを制御するデータベース・オプション ....             | 410        |
| ユーザ・リソースに影響するその他の設定 .....                  | 411        |
| システム・テーブルのユーザとパーミッション .....                | 412        |
| トランスポート・レイヤ・セキュリティ .....                   | 413        |
| IPv6 のサポート .....                           | 414        |
| <br>                                       |            |
| <b>第 9 章</b>                               |            |
| <b>データ整合性の確保 .....</b>                     | <b>415</b> |
| データ整合性の概要 .....                            | 415        |
| データが無効になるとき .....                          | 416        |
| データの有効性の確保 .....                           | 416        |
| データベース内容の変更 .....                          | 418        |
| データ整合性ツール .....                            | 418        |
| 整合性制約を実装するための SQL 文 .....                  | 419        |
| カラムのデフォルトの使用 .....                         | 420        |
| カラムのデフォルトの作成 .....                         | 421        |
| カラムのデフォルトの変更と削除 .....                      | 422        |
| カラムのデフォルト値の操作 .....                        | 422        |
| Sybase Central におけるカラムのデフォルトの編集 .....      | 424        |
| 日付、時刻、タイムスタンプのデフォルト .....                  | 425        |
| USER デフォルト .....                           | 426        |
| IDENTITY または AUTOINCREMENT のデフォルト .....    | 426        |
| NEWID デフォルト .....                          | 427        |
| NULL デフォルト .....                           | 428        |
| 文字列と数値のデフォルト .....                         | 428        |
| 定数式のデフォルト .....                            | 428        |
| テーブル制約とカラム制約の使用 .....                      | 429        |
| カラムまたはテーブルに対する UNIQUE 制約の使用 .....          | 429        |
| カラムに対する IQ UNIQUE 制約の使用 .....              | 430        |
| カラムに対する CHECK 条件の使用 .....                  | 430        |
| ユーザ定義データ型の CHECK 条件の定義 .....               | 431        |
| Sybase Central におけるカラム制約の編集 .....          | 432        |
| テーブルに対する CHECK 条件の使用 .....                 | 432        |
| CHECK 条件の修正と削除 .....                       | 432        |
| エンティティ整合性と参照整合性の宣言 .....                   | 433        |
| エンティティ整合性の宣言 .....                         | 434        |
| エンティティ整合性の確保 .....                         | 434        |
| 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<br>違反する場合 ..... | 435        |
| 参照整合性の宣言 .....                             | 435        |
| 参照整合性の損失 .....                             | 439        |
| 同時処理の制御 .....                              | 440        |

|               |                                            |            |
|---------------|--------------------------------------------|------------|
|               | 参照整合性チェックの無効化.....                         | 443        |
|               | システム・テーブルの整合性ルール.....                      | 443        |
| <b>第 10 章</b> | <b>トランザクションとバージョン管理 .....</b>              | <b>445</b> |
|               | トランザクションとバージョン管理の概要.....                   | 445        |
|               | トランザクションの概要.....                           | 446        |
|               | 同時実行性の概要.....                              | 448        |
|               | バージョン管理の概要 .....                           | 450        |
|               | バージョン管理による矛盾の回避 .....                      | 458        |
|               | ロックの仕組み .....                              | 459        |
|               | DML 操作のロック .....                           | 459        |
|               | DDL 操作のロック .....                           | 461        |
|               | プライマリ・キーとロック .....                         | 463        |
|               | ロックの管理.....                                | 463        |
|               | アクティブなロックの表示 .....                         | 463        |
|               | ロック競合の管理.....                              | 464        |
|               | 独立性レベル.....                                | 465        |
|               | チェックポイント、セーブポイント、トランザクションの<br>ロールバック ..... | 466        |
|               | チェックポイント.....                              | 467        |
|               | トランザクション内のセーブポイント .....                    | 468        |
|               | トランザクションのロールバック .....                      | 470        |
|               | システム・リカバリ .....                            | 471        |
|               | リカバリにおけるトランザクション情報の役割.....                 | 472        |
|               | パフォーマンスの影響 .....                           | 474        |
|               | 重複したバージョンと削除 .....                         | 475        |
|               | トランザクション内のカーソル .....                       | 476        |
|               | カーソルとバージョン管理 .....                         | 477        |
|               | カーソルの依存性.....                              | 477        |
|               | カーソルのスクロール .....                           | 477        |
|               | ホールド・カーソル .....                            | 478        |
|               | 位置付け操作 .....                               | 478        |
|               | カーソル・コマンド構文と例.....                         | 479        |
|               | カーソルのログを取るメッセージの制御.....                    | 479        |
| <b>第 11 章</b> | <b>各国語と文字セット .....</b>                     | <b>481</b> |
|               | 各国語と文字セットの概要.....                          | 481        |
|               | デフォルト照合の使用 .....                           | 483        |
|               | ソフトウェアの文字セットについて.....                      | 483        |
|               | Windows でのコード・ページ .....                    | 483        |
|               | 照合を使用した文字のソート.....                         | 484        |
|               | ロケールについて .....                             | 485        |
|               | 照合について .....                               | 485        |

|                                |     |
|--------------------------------|-----|
| 照合の表示 .....                    | 485 |
| ANSI か OEM の選択 .....           | 486 |
| ANSI 照合についての注意 .....           | 486 |
| マルチバイト照合の使用 .....              | 487 |
| 文字セット変換について .....              | 488 |
| データベース・メッセージの文字変換 .....        | 488 |
| 接続文字列と文字セット .....              | 489 |
| 文字セット変換の回避 .....               | 490 |
| 各国語と文字セットに関する作業 .....          | 491 |
| デフォルトの照合の確認 .....              | 491 |
| 文字セットの環境設定 .....               | 492 |
| ロケール情報の決定 .....                | 493 |
| ロケールの設定 .....                  | 493 |
| 名前付き照合を使用するデータベースの作成 .....     | 495 |
| 文字セット変換を使用するデータベース・サーバの起動 .... | 495 |
| ある照合から別の照合へのデータベースの変更 .....    | 495 |
| 互換性の問題 .....                   | 496 |
| パフォーマンスの問題 .....               | 496 |

## 第 12 章

|                                                   |            |
|---------------------------------------------------|------------|
| <b>データのバックアップ、リカバリ、アーカイブ .....</b>                | <b>497</b> |
| データの保護 .....                                      | 497        |
| データベースのバックアップ .....                               | 498        |
| データ・ストアの種類 .....                                  | 499        |
| バックアップの種類 .....                                   | 499        |
| アーカイブ・デバイスの選択 .....                               | 503        |
| バックアップの準備 .....                                   | 504        |
| バックアップの実行 .....                                   | 507        |
| オペレータの存在についての指定 .....                             | 508        |
| バックアップの種類の指定 .....                                | 508        |
| 仮想バックアップの指定 .....                                 | 509        |
| アーカイブ・デバイスの指定 .....                               | 509        |
| その他のバックアップ・オプション .....                            | 513        |
| テープ・デバイスの待機 .....                                 | 514        |
| 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<br>リストア .....           | 514        |
| バックアップの例 .....                                    | 515        |
| バックアップ中のエラーからのリカバリ .....                          | 516        |
| バックアップ完了後の作業 .....                                | 517        |
| Sybase 以外の製品によるバックアップの実行 .....                    | 517        |
| 仮想バックアップ .....                                    | 518        |
| 仮想バックアップの種類 .....                                 | 519        |
| SAN スナップショットまたはシャドウ・ハードウェアを<br>使用した仮想バックアップ ..... | 521        |

|                                   |     |
|-----------------------------------|-----|
| システム・レベルのバックアップの実行 .....          | 521 |
| データベースの停止 .....                   | 522 |
| 適切なファイルのバックアップ .....              | 522 |
| システム・レベルのバックアップからのリストア .....      | 524 |
| データベースの検証 .....                   | 525 |
| データベースのリストア .....                 | 526 |
| リストア前の作業 .....                    | 526 |
| RESTORE 文 .....                   | 531 |
| 正しい順序のリストア .....                  | 537 |
| リストア後の再接続 .....                   | 539 |
| リストア後にトランザクション・ログ名を変更する .....     | 540 |
| リストア後のデータベースの検証 .....             | 540 |
| 排他的な書き込みアクセスが必要なリストア .....        | 541 |
| ヘッダ情報の表示 .....                    | 541 |
| リストア中のエラーからのリカバリ .....            | 542 |
| バックアップとシンボリック・リンク (UNIX のみ) ..... | 542 |
| アテンンドなしバックアップ .....               | 543 |
| バックアップとリストアに関する情報の取得 .....        | 544 |
| バックアップ・ログの配置 .....                | 545 |
| バックアップ・ログの内容 .....                | 545 |
| バックアップ・ログの管理 .....                | 546 |
| DB 領域名の記録 .....                   | 547 |
| データ・バックアップ方式とリカバリ方式の決定 .....      | 548 |
| 日常のバックアップの計画 .....                | 548 |
| バックアップとリストアの責任者の指定 .....          | 550 |
| バックアップとリストアのパフォーマンスの向上 .....      | 550 |
| 読み取り専用ハードウェアによるデータのアーカイブ .....    | 552 |
| 読み取り専用ハードウェアの使用 .....             | 553 |

## 第 13 章

|                                |     |
|--------------------------------|-----|
| システムのリカバリとデータベースの修復 .....      | 557 |
| リカバリと修復の概要 .....               | 557 |
| 通常のリカバリ .....                  | 558 |
| データベースの検証 .....                | 559 |
| sp_iqcheckdb ストアド・プロシージャ ..... | 559 |
| sp_iqcheckdb の出力 .....         | 563 |
| sp_iqcheckdb 実行時のリソースの問題 ..... | 566 |
| データベースの修復 .....                | 567 |
| インデックス・エラーの分析 .....            | 567 |
| インデックス・エラーの修復 .....            | 571 |
| 割り付けの問題の分析 .....               | 571 |
| 割り付けの問題の修復 .....               | 574 |
| 強制リカバリ・モード .....               | 576 |
| 強制リカバリ・モードでのサーバの起動 .....       | 577 |
| リーク領域のリカバリ .....               | 579 |

|                                            |            |
|--------------------------------------------|------------|
| マルチプレックス・データベースのリカバリ .....                 | 581        |
| トランザクション・ログを使わない緊急リカバリ .....               | 581        |
| -f リカバリ・オプション .....                        | 581        |
| DBCC によってレポートされた問題の処理 .....                | 582        |
| 修復できないインデックスの問題 .....                      | 583        |
| 矛盾するインデックス、テーブル、またはカラムの削除 ....             | 584        |
| DBCC エラー・メッセージ.....                        | 586        |
| <br>                                       |            |
| <b>第 14 章</b>                              |            |
| <b>トラブルシューティングのヒント .....</b>               | <b>589</b> |
| 状況別の解決策 .....                              | 590        |
| サーバのリカバリとデータベースの修復.....                    | 590        |
| サーバの運用上の問題 .....                           | 592        |
| データベース接続の問題.....                           | 602        |
| Interactive SQL (dbisql/dbisqlc) の問題 ..... | 605        |
| リソースの問題 .....                              | 607        |
| 処理の問題 .....                                | 618        |
| パフォーマンスの問題 .....                           | 623        |
| Sybase Central の問題 .....                   | 624        |
| ネットワーク通信のトラブルシューティング .....                 | 624        |
| 互換性のあるプロトコルを使用していることの確認.....               | 625        |
| 最新のドライバを使用していることの確認 .....                  | 625        |
| リブート時にコンピュータの電源を切る.....                    | 626        |
| レイヤごとのプロトコル・スタックの診断 .....                  | 626        |
| TCP/IP プロトコル・スタックのテスト .....                | 627        |
| 配線の問題の診断.....                              | 628        |
| 一般的なネットワーク通信の問題の確認.....                    | 628        |
| 診断ツール .....                                | 630        |
| sp_iqstatus ストアド・プロシージャ.....               | 630        |
| 通知メッセージの解釈 .....                           | 632        |
| sp_iqcheckdb ストアド・プロシージャ.....              | 636        |
| 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<br>確認 .....     | 636        |
| 現在実行中の文の検索 .....                           | 637        |
| サーバ要求のロギング .....                           | 637        |
| 診断情報を収集するための接続 .....                       | 641        |
| 通信の問題の診断.....                              | 641        |
| サポート・センタへの問題の報告 .....                      | 641        |
| getiqinfo を使った診断情報の収集 .....                | 642        |
| getiqinfo によって収集される情報 .....                | 644        |
| 接続情報の関連付け .....                            | 646        |
| その他の情報源 .....                              | 647        |
| チェックリスト：サポート・センタに提出する情報 .....              | 647        |
| <br>                                       |            |
| <b>索引 .....</b>                            | <b>649</b> |



# はじめに

## このマニュアルの内容

Sybase® IQ は、データ・ウェアハウスやデータ・マート用に特化された高性能の意思決定支援サーバです。この『システム管理ガイド：第 1 巻』では Sybase IQ における管理の概念とその手順について説明しています。

## 対象読者

このマニュアルはシステム管理者とデータベース管理者、または Sybase IQ の設定や管理を行う必要のあるユーザを対象としています。リレーショナル・データベース・システムの基礎知識と、Sybase IQ のユーザ・レベルの基本的な経験があることを前提にしています。このマニュアルは、他のマニュアルと併用するように構成されています。

## このマニュアルの使用方法

次の表は、目的に応じた参照先の章を示します。

**表 1：このマニュアルの使用方法**

| 目的                              | 参照先                                                           |
|---------------------------------|---------------------------------------------------------------|
| Sybase IQ 管理者の役割                | <a href="#">第 1 章 「Sybase IQ システム管理の概要」</a>                   |
| IQ データベース・サーバの起動と停止、ユーザ接続の設定    | <a href="#">第 2 章 「Sybase IQ の実行」</a>                         |
| ユーザ接続の設定                        | <a href="#">第 3 章 「Sybase IQ の接続」、第 4 章 「接続パラメータと通信パラメータ」</a> |
| Sybase IQ データベースの作成             | <a href="#">第 5 章 「データベース・オブジェクトの使用」</a>                      |
| Sybase IQ インデックスの選択             | <a href="#">第 6 章 「Sybase IQ インデックスの使用」</a>                   |
| データベースへのデータのロード                 | <a href="#">第 7 章 「データベースへのデータの入出力」</a>                       |
| ユーザの追加とユーザ権限の割り当て               | <a href="#">第 8 章 「ユーザ ID とパーミッションの管理」</a>                    |
| テーブル内のデータに対する制約の指定              | <a href="#">第 9 章 「データ整合性の確保」</a>                             |
| トランザクションの仕組み                    | <a href="#">第 10 章 「トランザクションとバージョン管理」</a>                     |
| 使用する言語向けのデータベース設定               | <a href="#">第 11 章 「各国語と文字セット」</a>                            |
| データベースのバックアップとリストア、およびデータのアーカイブ | <a href="#">第 12 章 「データのバックアップ、リカバリ、アーカイブ」</a>                |
| サーバのバックアップとリカバリ、およびデータベースの修復    | <a href="#">第 13 章 「システムのリカバリとデータベースの修復」</a>                  |
| Sybase IQ 実行時に発生した問題の解決         | <a href="#">第 14 章 「トラブルシューティングのヒント」</a>                      |
| プロシージャとバッチの作成                   | <a href="#">第 2 巻第 1 章 「プロシージャとバッチの使用」</a>                    |
| OLAP 関数の使用                      | <a href="#">第 2 巻第 2 章 「OLAP の使用」</a>                         |

| 目的                                           | 参照先                                                                                                |
|----------------------------------------------|----------------------------------------------------------------------------------------------------|
| IQ を Open Server として使用するための設定                | 第 2 巻第 3 章「データ・サーバとしての Sybase IQ」                                                                  |
| リモート・データにアクセスするための Sybase IQ の設定             | 第 2 巻第 4 章「リモート・データへのアクセス」<br>第 2 巻第 5 章「リモート・データ・アクセス用のサーバ・クラス」                                   |
| スケジューリングとイベント処理を使用してのデータベース管理タスクの自動化         | 第 2 巻第 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」                                                              |
| Sybase IQ 内の XML ドキュメントへの Java ツールを使用してのアクセス | 付録 A「データベースでのロジックのデバッグ」                                                                            |
| Sybase デバッガの使用                               | 付録 A「データベースでのロジックのデバッグ」                                                                            |
| JDBC を使用したデータへのアクセス                          | 『SQL Anywhere® サーバー プログラミング』                                                                       |
| クライアントと Sybase IQ サーバ間で渡されるデータの機密性と整合性の保護    | 『SQL Anywhere Studio セキュリティ・ガイド』および『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」 |

## 関連マニュアル

Sybase IQ 15.1 マニュアル・セットの構成は次のとおりです。

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

- 『Sybase IQ によるラージ・オブジェクト管理』では、Sybase IQ データ・レポジトリ内での BLOB (Binary Large Object) および CLOB (Character Large Object) の格納と取得について説明しています。このオプションの製品をインストールするには、別のライセンスが必要です。
- 『Sybase IQ 15.0 の新機能』には、バージョン 15.0 の新機能と動作変更が記載されています。
- 『新機能の概要 Sybase IQ 15.1』には、最新バージョンの新機能と動作変更の概要がまとめられています。
- 『パフォーマンス&チューニング・ガイド』では、巨大なデータベースのクエリ最適化、設計、チューニングについて説明しています。
- 『クイック・スタート』には、Sybase IQ ソフトウェアのインストールを検証するために Sybase IQ に付属しているデモ・データベースの構築とクエリを行う手順が記載されています。デモ・データベースをマルチプレックスに変換するための情報も記載されています。
- 『リファレンス・マニュアル』— Sybase IQ の 2 冊のリファレンス・ガイドで構成されています。
  - 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』では、Sybase IQ でサポートされる SQL、ストアド・プロシージャ、データ型、およびシステム・テーブルについて説明しています。
  - 『リファレンス：文とオプション』では、Sybase IQ でサポートされる SQL 文およびオプションについて説明しています。
- 『システム管理ガイド』— 2 巻構成です。
  - 『システム管理ガイド：第 1 巻』では、起動、接続、データベース作成、自動入力とインデックス作成、マルチプレックス管理、データの整合性、トランザクション、バージョン設定、照合について説明しています。
  - 『システム管理ガイド：第 2 巻』には、セキュリティ、バックアップとアーカイブ、リモート・データ・アクセス、およびイベントについての説明と、問題の解決方法、システムのリカバリ方法、データベースの修復方法が記載されています。
- 『ユーザ定義関数ガイド』には、ユーザ定義関数、パラメータ、および考えられる使用事例が記載されています。

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

---

### Sybase IQ および SQL Anywhere

Sybase IQ は SQL Anywhere Server を拡張したものであり、SQL Anywhere<sup>®</sup> パッケージの 1 コンポーネントです。そのため、Sybase IQ は SQL Anywhere Server の多くの同じ機能をサポートします。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 Server — データベース管理ガイド』では、SQL Anywhere データベースの実行、管理、構成方法について説明します。データベース接続、データベース・サーバ、データベース・ファイル、バックアップ手順、セキュリティ、高可用性、Replication Server での複製、管理ユーティリティおよびオプションについても説明します。
- 『SQL Anywhere Server — プログラミング』では、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET プログラミング言語を使用したデータベース・アプリケーションの構築および展開方法について説明します。このマニュアルでは、ADO.NET や ODBC などの各種プログラミング・インタフェースについても説明します。
- 『SQL Anywhere Server — SQL リファレンス』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 言語の SQL Anywhere の実装 (検索条件、構文、データ型、関数) についても説明します。
- 『SQL Anywhere Server — SQL の使用法』では、データベースの設計／作成方法、データのインポート／エクスポート／変更方法、データの検索方法、およびストアド・プロシージャとトリガの作成方法について説明します。

Product Manuals (<http://www.sybase.com/support/manuals/>) および DocCommentXchange ([http://dcx.sybase.com/dcx\\_home.php](http://dcx.sybase.com/dcx_home.php)) の SQL Anywhere 11.0.1 コレクションの SQL Anywhere マニュアルも参照してください。

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

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

#### その他の情報

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

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

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

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

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

Sybase Product Manuals Web サイトにアクセスするには、[Product Manuals \(http://sybooks.sybase.com\)](http://sybooks.sybase.com) にアクセスしてください。

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ **製品動作確認の最新情報にアクセスする**

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

❖ **コンポーネント動作確認の最新情報にアクセスする**

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

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

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

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

❖ **EBF とソフトウェア・メンテナンスの最新情報にアクセスする**

- 1 Web ブラウザで [Sybase Support Page \(http://www.sybase.com/support\)](http://www.sybase.com/support) (<http://www.sybase.com/support>) を指定します。

- 2 [EBFs/Maintenance] を選択します。ユーザ名とパスワードの入力が求められたら、MySybase のユーザ名とパスワードを入力します。
- 3 製品を選択します。
- 4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

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

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

## 構文の表記規則

このマニュアルでは、構文の説明に次の表記規則を使用します。

- **キーワード** SQL キーワードは大文字で示します。ただし、SQL キーワードは大文字と小文字の区別がないので、入力するときはどちらで入力してもかまいません。たとえば、SELECT は Select でも select でも同じです。
- **プレースホルダ** 適切な識別子または式で置き換えられる項目は、斜体で表記します。
- **継続** 省略記号 (...) で始まる行は、前の行から文が続いていることを表します。
- **繰り返し項目** 繰り返し項目のリストは、リストの要素の後ろに省略記号 (ピリオド3 つ ...) を付けて表します。1 つまたは複数の要素を指定できます。複数の要素を指定する場合は、各要素間はカンマで区切る必要があります。
- **オプション指定部分** 文のオプション指定部分は、角カッコで囲みます。次に例を示します。

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

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

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

```
[ ASC | DESC ]
```

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

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

QUOTES { ON | OFF }

この例では、ON、OFF のどちらかを必ず入力しなければなりません。中カッコ自体は入力しないでください。

## 書体の表記規則

表 2 に、このマニュアルで使用している書体の表記規則を示します。

表 2：書体の表記規則

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

## デモ・データベース

Sybase IQ には、デモ・データベースを作成するスクリプトが用意されています。

デモ・データベースは小規模企業の例を示しています。このデータベースは、企業の内部情報 (従業員、部署、財務データ) に加えて、製品情報 (製品) と販売情報 (注文、顧客、担当者) で構成されています。

デモ・データベースを作成するには、UNIX ではファイル

`$IQDIR15/demo/mkiqdemo.sh` を、Windows では

`%ALLUSERSPROFILE%¥SybaseIQ¥demo¥mkiqdemo.bat` を実行します。

デモ・データベースは、`iqdemo.db` という名前のファイルに作成されます。

## アクセシビリティ機能

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



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

---

### アクセシビリティ・ツールの設定

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

---

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

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

### 不明な点があるときは

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



# 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                                     | ページ |
|------------------------------------------|-----|
| <a href="#">Sybase IQ の概要</a>            | 2   |
| <a href="#">システム管理のツール</a>               | 3   |
| <a href="#">データベース・サーバ</a>               | 4   |
| <a href="#">SQL Anywhere と Sybase IQ</a> | 5   |
| <a href="#">セキュリティの概要</a>                | 6   |
| <a href="#">同時処理</a>                     | 7   |
| <a href="#">マルチプレックス機能</a>               | 8   |
| <a href="#">ストアド・プロシージャ</a>              | 8   |
| <a href="#">システム・テーブルとシステム・ビュー</a>       | 9   |
| <a href="#">コマンドと関数</a>                  | 10  |
| <a href="#">メッセージ・ロギング</a>               | 11  |
| <a href="#">ユーティリティ・データベース</a>           | 19  |
| <a href="#">大容量データベースの管理</a>             | 19  |

## Sybase IQ の概要

Sybase IQ は、データ・ウェアハウジング専用に設計された、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<sup>®</sup> 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樣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、さまざまなソースを使用して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に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用されます。「**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に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 マルチプレックスは、高度にスケーラブルな共有ディスク・グリッド・テクノロジーで、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

通常、データベース管理者 (DBA) は、表 1 (xvii ページ) に示すタスクについて責任を負います。

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 ともいいます。これを使用すると、対話的に SQL 文を入力し、それを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 では、Java ベースの DBISQL と、以前のリリースで提供されていた C ベースの DBISQL の両方をサポートします。

『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](#)」(8 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

「**テーブル領域**」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 15.1 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領域です。

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す。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のパスがある「**dbfile**」という用語は、オペレーティング・システムの各ファイルのことを指します。

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](#)」(177 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)。データベース上で実行するすべての操作についての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するユーザで、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。その他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リングし、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 「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** ほとんどの例文では、`iqdemo` データベースを使用します。これは、Sybase IQ と共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サンプル・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 9 章 システム・テーブル](#)」と「[第 8 章 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。

- DDL (データ定義言語) 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。
- DML (データ操作言語) 文。データの検索や、データベースからのデータの入出力ができます。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
-gi 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_1/logfiles/
sunopteng_iqdemo.0007.srvlog -hn 7
I. 05/06 17:07:25. 0000000000 DB: r/w, Main Buffs=127, Temp Buffs=95,
Pgpsz=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:
15.1.0.5071/090501/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:
15.1.0.5071/090501/P/GA/Sun_Opteron/OS 5.10/64bit/2009-05-01 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   | 15.1           | 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   | 15.1. <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     | 090501         | 内部用の日付識別子。                  |
| 内部ビルド・タイプ   | P              | P = 運用ビルド                   |

| バージョン文字列の要素         | 例                      | コメント                                                                                                                                                                                                                              |
|---------------------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| リリース・タイプ            | GA                     | GA = General Availability<br>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         | 2009-05-01<br>01:21:40 | YYYY-MM-DD hh:mm:ss (ISO datetime フォーマット) で表示される。<br><b>YYYY</b> 西暦年 4 桁<br><b>MM</b> 月 2 桁 (01 ~ 12)<br><b>DD</b> 日 2 桁 (01 ~ 31)<br><b>hh</b> 24 時間表記の時 2 桁 (00 ~ 23)<br><b>mm</b> 分 2 桁 (00 ~ 59)<br><b>ss</b> 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・ログ内のエントリは、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。次に例を示します。

```
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には、「[接続情報の関連付け](#)」(646 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](#)」(632 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) 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ます。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・オプションは今後廃止されることになりました。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と、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・ストアド・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* のパターンになります。ここで、*n* は 1 以上の整数で、*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* の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。

## メッセージ・ログ管理 のエラー

- 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." というエラー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のエラー。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、「<ファイル名> の名前の変更に失敗しました。メッセージ・ログはアーカイブされませんでした。メッセージ・ロギングは停止しました。」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、「メッセージ・ロギングが再開しました。」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勧めします。

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・ログの最大量は 128GB で、ほとんどの場合に十分な量です。

---

**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](#)」(387 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することができます。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、運用データ・ウェアハウス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。デフォルトでは CPU の数に基づくバッファ・パーティションが有効になっており、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](#)」(464 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。『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](#)」(369 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、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できます。[「DB 領域管理の例」\(216 ページ\)](#)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。

## 中間バージョン管理

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」\(198 ページ\)](#)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やクエリ・プランに影響するオプションと、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

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多くの場合に適切に動作するように設計されています。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。『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。

# Sybase IQ の実行

## 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                                   | ページ |
|----------------------------------------|-----|
| <a href="#">データベース・サーバの起動</a>          | 25  |
| <a href="#">Windows サービスとしてのサーバの実行</a> | 32  |
| <a href="#">コマンド・ライン・スイッチの使用</a>       | 32  |
| <a href="#">サーバ・アクティビティのモニタ</a>        | 48  |
| <a href="#">データベース・サーバの停止</a>          | 51  |
| <a href="#">データベースの開始と停止</a>           | 56  |
| <a href="#">iqdemo データベースの起動</a>       | 58  |
| <a href="#">Sybase Central の開始と停止</a>  | 59  |

## 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](#)」(26 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](#)」(29 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](#)」(32 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サンプル・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](#)」(58 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](#)」(83 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。「[DBISQL からのサーバの起動](#)」(46 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](#)」(32 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](#)」(35 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、あるいは、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。この例で、サーバはデフォルトのポート 2638 で起動します。

このコマンドは、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たり、指定されたデータベースを起動したり、すべての必要な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](#)」(11 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、%IQLOGDIR15% は、Windows プラットフォーム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 マルチプレックスの使用』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 15.1] - [Sybase Central Java Edition] を選択します。

- 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びに使用されます。

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。

サーバ・ウィザードの起動

接続パラメータ

接続情報の指定

起動するサーバの接続情報を指定してください。

接続プロファイル(F): その他

サーバ名(S): pcxp4338\_iqdemo

ホストマシン(H): pcxp4338

ポート番号(P): 7777

エージェント・ポート(Q): 1870

データベース(D) 名前: iqdemo

どのユーザ・アカウント (DBA 権限を所有) を使用しますか？

ユーザ名(U): DBA

パスワード(A): ●●●●

< 戻る(B) 次へ(N) > 完了(F) キャンセル

- 6 パラメータを入力したら、[次へ] をクリックします。



[データベース・パス] 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、ボックスに暗号化キーを入力します。

- 7 [次へ] をクリックします。[概要] 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 15.1] の順に選択します。

ここから、Sybase IQ Demo データベース、Sybase Central、Interactive SQL Classic、Interactive SQL Java 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              | ロケーション                                                                     | 用途                                                                                                                                                                                                                                              |
|--------------------|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>default.cfg</i> | <i>\$IQDIR15/scripts</i> (UNIX)、<br><i>%IQDIR15%\scripts</i> (Windows)     | 汎用設定ファイル。このファイルは、 <b>start_iq</b> 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する。                                                                                                                                                            |
| <i>iqdemo.cfg</i>  | <i>\$IQDIR15/data/demo</i> (UNIX)、<br><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する。                                                                                                                                                                                                             |

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  
tcipip(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実に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。

---

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 'tcipip(port=nnnn)'** を設定せずにサーバを起動すると、サーバはデフォルトの TCP/IP ポート番号 2638 を使用します。すでに使用されているポート番号を指定すると、サーバは起動に失敗します。

### デフォルトの設定ファイル

デフォルトの設定ファイル (*default.cfg*) には、必要なすべての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せんが、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](#)」([56 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に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ーヘッド

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・キャッシュのデフォルト・サイズは 8MB ですが、実際のデータベースの使用には小さすぎます。IQ メイン・ストアとテンポラリ・ストアのバッファ・キャッシュ・サイズは、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・キャッシュ・サイズ) を指定します。この方法が推奨されます。

- データベースのキャッシュ・サイズを設定するには、  
`main_cache_memory_mb` 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` と `-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・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」\(39 ページ\)](#) 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にのみ使用します。メモリが十分でないときにこのスイッチを使用すると、パフォーマンスが著しく低下することがあります。

---

## 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](#)」(37 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`、および `-cl` の各サーバ・オプションの説明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 ～ 128 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リ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 権限を持つユーザのみが追加のデータベースを起動できる。

- **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る。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に注意してください)。
- **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](#)」(19 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** 権限を持つユーザのみがサーバを停止できます。
- **ALL** — すべてのユーザがサーバを停止できます。
- **NONE** — ユーザは **DBSTOP** ユーティリティまたは **STOP ENGINE** コマンドを使用してサーバを停止できません。

### データベースのロードとアンロード

**-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」\(198 ページ\)](#)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・サーバ間の通信では、通信プロトコルが必要です。Sybase IQ 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・サーバで使用できるプロトコルを制限できます。クライアント側は、CommLinks 接続パラメータを使用して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](#)」(152 ページ)を参照してください。

## 非アクティブ接続の制限

### デフォルト・ネットワーク・タイムアウトの設定

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。

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 -tl スイッチを指定していないすべてのクライアントに対して活性タイムアウト(秒)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。

```
-tl 300
```

この値でうまくいかない場合は、-tl 1200 を指定して、活性タイムアウトを 20 分間に設定してください。

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

#### デフォルト・クライアント・タイムアウトの設定

-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 コマンドを実行すると、このコマンドはデータベース・サーバを起動し、それに 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・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・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」\(93 ページ\)](#)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/iql51/IQ-15_1/demo
Server Executable   : /myserver/users/sybase/iql51/IQ-15_1/bin64/iqsrv15
Server Output Log   : /myserver/users/sybase/iql51/IQ
15_1/logfiles/myserver_iqdemo.0001.srvlog
Server Version      : 15.1.0.5027/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 15.1
I. 05/22 16:19:05.      (64bit mode)
I. 05/22 16:19:05. Copyright 1992-2009 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/iql51/IQ-15_1/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
新しいプロセス ID は 397436 です。
サーバは正常に開始しました。

```

## 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。次に例を示します。

```
% 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 Sybase IQ のインストールで定義された **\$IQDIR15/logfiles/** ディレクトリに書き込まれ、Windows の場合は **\$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
```

次に例を示します。

```
jones  422034      1    0 17:47:36      -   0:04
/ibm64srv/users/sybase/iq151/IQ-15_1/bin64/
iqsrv15
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o
/ibm64srv/users/sybase/iq151/IQ-
15_1/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・オプションの範囲と継続期間](#)」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、ドロップダウン・メニューから [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、[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](#)」(54 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と、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** では、データベース・サーバを起動したウィンドウで、次のように入力します。

q

異なるデバイスに入力をリダイレクトした場合、または **start\_iq** を使用してサーバを起動した場合、このコマンドは機能しません。**iqsrv15** を使用した場合にのみ機能します。

- **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す。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** 実行ファイルのフル・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。次に例を示します。

```
% 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
iqsrv15 @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/iql151/IQ-15_1/bin64/iqsrv15 @iqdemo.cfg
iqdemo.db -ti 4400 -gn 25 -o /ibm1/users/sybase/iql151/IQ
15_1/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/iql151/IQ-15_1/bin64/iqsrv15
@iqdemo.cfg iqdemo.db -ti 4400 -gn 25 -o /myserver/users/sybase/iql151/IQ-
15_1/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か、-gk ALL オプションを指定してサーバを起動した場合、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。システム・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。

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](#)」(25 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。

- DBISQL 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。
- DBISQL 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・ソースの使用](#)」(82 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 DBISQL 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*” を設定します。「[Sybase Central または DBISQL からサンプル・データベースに接続する](#)」(71 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 データベースの管理](#)」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限り、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。
- **DBISQL** 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すると、起動を容易にする **iqdemo** データベースと設定ファイルを作成できます。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 の起動と停止](#)」またはオンライン・ヘルプを参照してください。



# Sybase IQ の接続

## 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** UNIX サーバ上の Sybase IQ には、Sybase Central か、Windows または Linux クライアント上の dbisql から接続できます。

---

## 内容

| トピック                                                     | ページ |
|----------------------------------------------------------|-----|
| <a href="#">接続の概要</a>                                    | 62  |
| <a href="#">Sybase Central または Interactive SQL からの接続</a> | 66  |
| <a href="#">Sybase Central での接続を簡略化する方法</a>              | 70  |
| <a href="#">簡単な接続の例</a>                                  | 71  |
| <a href="#">ODBC データ・ソースの使用</a>                          | 82  |
| <a href="#">ODBC データ・ソースの作成と編集</a>                       | 83  |
| <a href="#">ファイル・データ・ソースの使用</a>                          | 92  |
| <a href="#">UNIX での ODBC データ・ソースの使用</a>                  | 93  |
| <a href="#">OLE DB を使用してデータベースに接続する</a>                  | 95  |
| <a href="#">接続パラメータ</a>                                  | 97  |
| <a href="#">Sybase IQ の接続の確立方法</a>                       | 98  |
| <a href="#">他のデータベースからの接続</a>                            | 106 |
| <a href="#">サーバを検出できるかどうかのテスト</a>                        | 108 |
| <a href="#">統合化ログインの使用</a>                               | 108 |
| <a href="#">接続の切断と削除</a>                                 | 117 |
| <a href="#">接続ログ</a>                                     | 117 |
| <a href="#">起動のトラブルシューティング、シャットダウン、接続</a>                | 118 |

## 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 Server — 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、すべてクライアント・アプリケーションからのアクティビティにより形成されます。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。

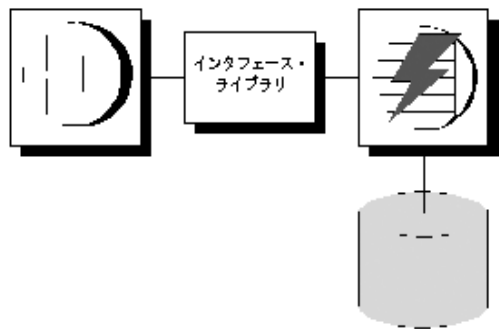


- *Embedded SQL* — Embedded SQL 接続については、この章で説明します。
- *Sybase Open Client* — Open Client 接続については、この章では説明しません。Open Client アプリケーションから IQ への接続については、『システム管理ガイド：第 2 巻』の「[第 3 章 データ・サーバとしての Sybase IQ](#)」を参照してください。
- *JDBC* — JDBC 接続については、この章で説明します。JDBC 経由での接続については、『Sybase IQ の概要』の「[第 4 章 データベースの管理](#)」を参照してください。JDBC データ・ソースの作成については、『SQL Anywhere Server — 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](#)」([201 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



接続について

| 目的                                                         | 参照箇所                                                                 |
|------------------------------------------------------------|----------------------------------------------------------------------|
| Sybase Central または Interactive SQL からの接続の概要 (関連ドレイバの説明を含む) | 「 <a href="#">Sybase Central または Interactive SQL からの接続</a> 」(66 ページ) |
| すばやく起動する例                                                  | 「 <a href="#">簡単な接続の例</a> 」(71 ページ)                                  |

| 目的                  | 参照箇所                                              |
|---------------------|---------------------------------------------------|
| 概要                  | 「 <a href="#">接続パラメータの構造</a> 」(64 ページ)            |
| 使用可能な接続パラメータ        | 「 <a href="#">第 4 章 接続パラメータと通信パラメータ</a> 」         |
| データ・ソースの作成          | 「 <a href="#">ODBC データ・ソースの使用</a> 」(82 ページ)       |
| 接続の確立方法についての詳しい説明   | 「 <a href="#">ODBC データ・ソースの使用</a> 」(82 ページ)       |
| ユーザの追加およびパーミッションの付与 | 「 <a href="#">Sybase IQ の接続の確立方法</a> 」(98 ページ)    |
| ネットワーク特有の接続問題の診断    | 「 <a href="#">ネットワーク通信のトラブルシューティング</a> 」(624 ページ) |
| 接続に関連する文字セットの問題     | 「 <a href="#">接続文字列と文字セット</a> 」(489 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で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なかには、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、次の例は、Windows システムの一般的な照合ユーティリティ (dbcollat) のコマンド・ラインです。すべて 1 行に入力してください。

```
dbcollat -c "uid=DBA;pwd=sql;dbn=iqdemo"  
c:¥temp¥iqdemo.col
```

**注意** DBISQL 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 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・ソースの使用](#)」(82 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](#)」(71 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、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。

エンジンを起動するためにデータベース名が必要です。

---

## [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、[ファイル] - [新しいウィンドウ] 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 では、[ツール]-[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、[ファイル]-[新しいウィンドウ] または [SQL]-[接続] を選択します。

[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** サンプル・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 *iqdemo.db* を指定し、[ID] タブでユーザ ID として DBA と入力し、パスワードとして sql と入力して、[OK] をクリックします。

サーバがリモートの場合、[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] ボックスがオンになっ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します。[データベース名] フィールドには完全なパス名を指定します。

---

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。

## 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。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 (Java) 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については、『SQL Anywhere Server — プログラミング』の「JDBC ドライバの選択」および「jConnect JDBC ドライバの使用」と「[ODBC データ・ソースの使用](#)」(82 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。

- [ID] タブでは、データベースに対する ID 情報を指定し、データ・ソースを指定します。
- [データベース] タブでは、接続先のサーバかデータベースまたはその両方を指定します。
- [詳細] タブでは、接続パラメータを追加し、接続用のドライバを指定します。

接続が確立すると、メイン・ウィンドウの左側のウィンドウ枠に、接続したデータベースの名前が表示されます (そのデータベースを実行しているサーバ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、`IQDSEDIT` (ディレクトリ・サービス・エディタ) を使用します。`IQDSEDIT` 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、`IQDSEDIT` 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なりますが、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。この項は、ネットワーク接続に慣れていない方、たとえばローカル・サーバで `iqdemo` サンプル・データベースを実行していてまだネットワークに接続していない方などを対象にしています。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要となります。

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](#)」(97 ページ) を参照してください。

## Sybase Central または DBISQL からサンプル・データベースに接続する

このマニュアルで紹介している例の多くは、DBISQL と呼ばれる Interactive SQL からサンプル・データベースに接続しています。

❖ サンプル・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 の場合は、Sybase Central または IQ Agent のようなユーティリティを呼び出す前に *IQ-15\_1.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 の場合は、Sybase Central を起動するには [ プログラム ] - [ Sybase ] - [ Sybase IQ 15.1 ] - [ Sybase Central Java Edition ] 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。

❖ サンプル・データベースへの接続 (Interactive SQL)

- 1 [ スタート ] メニューから Interactive SQL を起動するには、[ プログラム ] - [ Sybase ] - [ Sybase IQ 15.1 ] - [ Interactive SQL Java ] を選択します。
- 2 前の手順の手順 4 から 6 を実行してください。

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、まず次の質問に教えてください。

- 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 [データベース・ファイル] 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)。[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。
- 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、[データベース名] 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」 \(67 ページ\)](#)
- [「簡単な接続の例」 \(71 ページ\)](#)

## DBISQL から他のデータベースに接続する

実行中のデータベースに DBISQL から接続するには、次の手順に従います。

### ❖ UNIX 上の DBISQL 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
```

- 2 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、DBISQL 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* 上のサンプル・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

- 1 『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、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。
- 2 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
```

サンプル 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
```

- 3 次のコマンドを入力して、DBISQL を起動します。

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

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

---

**注意** Java ベースのバージョンではなく、古いユーティリティの Interactive SQL Classic を使用する場合は、`dbisql` の代わりに `dbisqlc` と入力します。`dbisqlc` はサポートされていますが、`dbisqlc` には `dbisql` のすべての機能が含まれていないことに注意してください。

---

-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、プラットフォームによっては、安全なネットワーク・パラメータの設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 15.1] - [Interactive SQL Java] 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
```

---

**注意** C バージョンの Interactive SQL を使用したい場合は、[プログラム] メニューから [Sybase] - [Sybase IQ 15.1] - [Interactive SQL Classic] を選択するか、Windows のコマンド・プロンプトで次のように入力します。

```
dbisqlc
```

---

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、DBISQL 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、*host name: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 15.1] - [ODBC Administrator 32 ビット版] または [Sybase IQ 15.1] - [ODBC Administrator 64 ビット版] を実行します。[ユーザー DSN] タブで [ユーザ・データ・ソース] を選択し、[設定] をクリックします。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、DBISQL ウィンドウが表示されます。DBISQL ウィンドウでは、タイトル・バー上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、サンプル・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サンプル・データベースの起動をカスタマイズする方法を示しています。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 : DBISQL からの接続

この例では、サンプル・データベースを DBISQL 内の埋め込みデータベースとして使用しています。

#### ❖ DBISQL 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 DBISQL を開始します。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 15.1] - [Interactive SQL Java] を選びます。
- システムのコマンド・プロンプトで `dbisql` と入力します。

DBISQL 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] フィールドにサンプル・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_1` の場合、次のように入力します。

```
c:\sybase\iq-15_1\sqdemo.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 が起動してサンプル・データベースをロードし、DBISQL 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 のような Embedded SQL アプリケーションと同じように、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](#)」(83 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](#)」(69 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 (Interactive SQL Classic では [ログイン] 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 データ・ソース名] オプションを選択し、データ・ソース名を指定します (これは、レジストリ内のデータ・ソースを参照する DSN 接続パラメータに相当します)。[検索] 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、サンプル・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。

---

**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、サンプル・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](#)」(98 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](#)」(98 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=tcp  
ip{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 3 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・ソースの使用

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」\(93 ページ\)](#) 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」\(71 ページ\)](#) と [「接続パラメータ・リストのアセンブル」\(101 ページ\)](#) 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 15.1] > [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、[ログイン] 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。次に例を示します。

```
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](#)」(86 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、使用したい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 サンプル・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_1¥demo¥iqdemo.db"
```

iqdsn の出力には、次の行が含まれます。

レジストリに書き込まれたユーザ・データ・ソース「My DSN」。

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、サンプル・データベースを起動するには [ プログラム ] メニューから [Sybase] - [Sybase IQ 15.1] - [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 設定] ダイアログ・ボックスで、[接続テスト] 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確なファイル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 にアクセスする必要がある場合は、『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 \(DSN\)](#)」(137 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、「ドライバが動作しません」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。

## [ログイン] 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](#)」(109 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 \(UID\)](#)」(151 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 \(PWD\)](#)」(148 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 \(ENP\)](#)」(140 ページ) を参照してください。

## [データベース] 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 \(ENG\)](#)」(139 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。次に例を示します。

```
C:¥Program Files¥Sybase¥IQ-15_1¥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 \(START\)](#)」(150 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 \(DBN\)](#)」(136 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。[検索] をクリックしてファイルを検索できます。次に例を示します。

```
C:¥Program Files¥Sybase¥IQ-15_1¥demo¥iqdemo.db
```

詳細については、「[DatabaseFile 接続パラメータ \(DBF\)](#)」(134 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 \(ENC\)](#)」(140 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に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 暗号化を使用する場合は、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 \(ODBC アドミニストレータ\)](#)」(84 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 \(LTO\)](#)」(146 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 \(IDLE\)](#)」(143 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 \(CBSize\)](#)」(131 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 \(ENC\)](#)」(140 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が始まります。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 [追加] 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、[次へ] をクリックします。
- 3 データ・ソースの作成手順に従います。

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
```

## 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](#)」(152 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](#)」(85 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 `$ODBCINI` または `$ODBCHOME` 変数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に従って検索します。

- 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 ODBC\_HOME — *.odbc.ini* ファイルを格納しているディレクトリに設定されている必要があります。

UNIX システムでは、Sybase IQ は ODBC ドライバをインストールするだけで、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。Sun Solaris システムの場合、*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ときは、それらに対して、*\$SYBASE/IQ-15\_1/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 Server — 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、次のように環境変数 *ASA\_ODBCTRACE\_VERBOSE* および *ASA\_ODBCTRACE\_FILE* を設定する必要があります。

```
$ setenv ASA_ODBCTRACE_VERBOSE ALL
$ setenv ASA_ODBCTRACE_FILE
/bluesun/fiona/odbctrace.out
$ dbisql -c "dsn=J123456" -nogui
```

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。



## 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 にアクセス可能です。

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 Server — 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 がサポートされません。
  - Sybase IQ では、カーソルを通じたリモート・アップデートが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、個別の **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 ASA OLE DB Provider** 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 **ASAProv** です。

ASAProv 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、ASAProv 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 の場所を変更した場合は、その場所を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 プロバイダについては、『SQL Anywhere Server — 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 Sybase IQ 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](#)」(97 ページ)を参照してください。

- 接続を開始する `Open` メソッド。

ADO の詳細については、『SQL Anywhere Server — プログラミング』を参照してください。

例

次の Visual Basic コードは、Sybase IQ への OLE DB 接続を開始します。

```
' Declare the connection object
Dim myConn as New ADODB.Connection
myConn.Provider = "ASAProv"
myConn.ConnectionString = "Data Source=Sybase IQ Demo"
myConn.Open
```

## 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](#)」(34 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、「[第14章 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...** という 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ので、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 : *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。

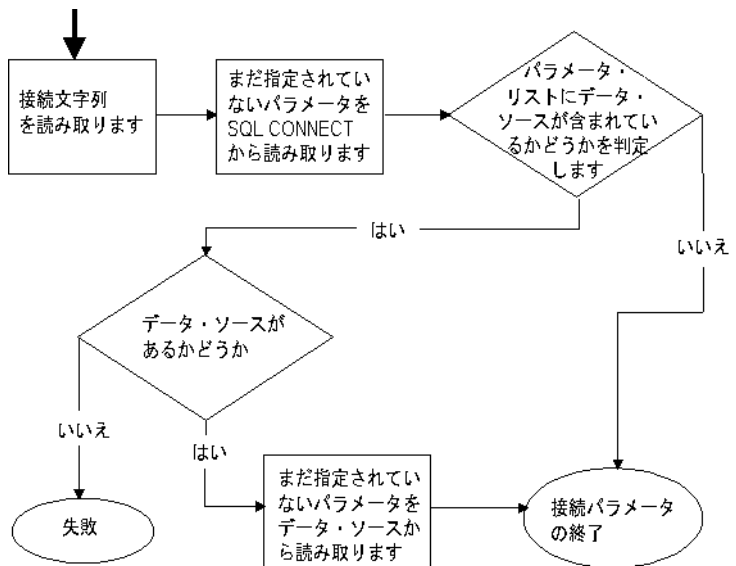
- Windows の場合、クライアント・アプリケーションは、現在のディレクトリ、システム・パス、*Windows* ディレクトリ、*Windows¥system* ディレクトリからファイルを探します。
- UNIX の場合、クライアント・アプリケーションは、システム・パスとユーザ・ライブラリ・パスからファイルを探します。

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ったら、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

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。

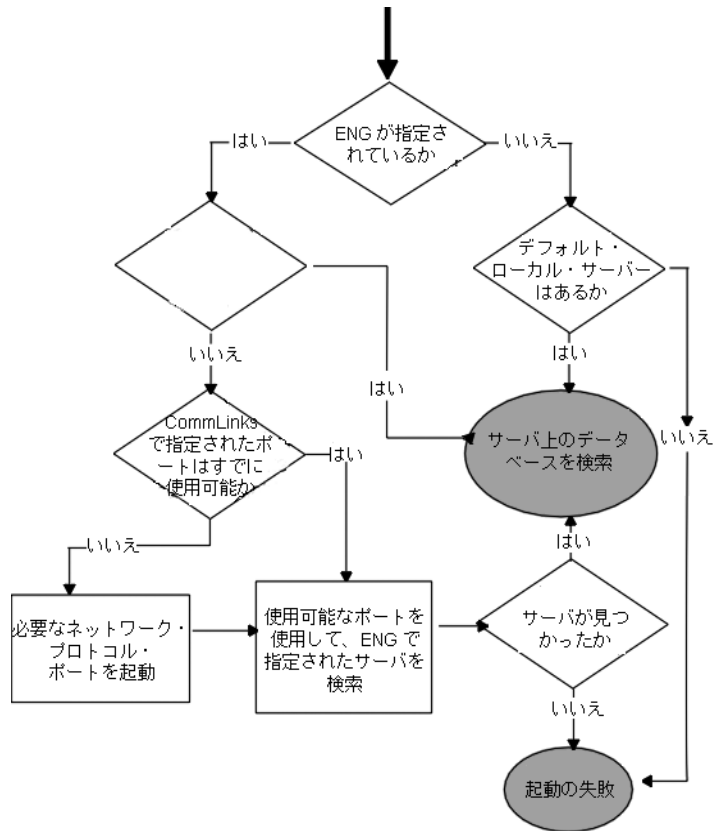
- 失敗 — この段階で失敗するのは、接続文字列または SQLCONNECT に、クライアントの接続ファイルに存在しないデータ・ソースが指定された場合のみです。
- 共通パラメータ — すでに他の接続が使用中である場合は、次の接続パラメータが無視されることがあります。
  - **AutoStop** データベースがすでにロードされている場合は、無視されます。
  - **DatabaseFile** DatabaseName が指定されていて、この名前を持つデータベースがすでに実行されている場合は、無視されます。

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・サーバが検索されます。



サーバが検出された場合、Sybase IQ はそのサーバ上で必要なデータベースの検出またはロードを実行します。詳細については、「[データベースの検出](#)」(105 ページ) を参照してください。

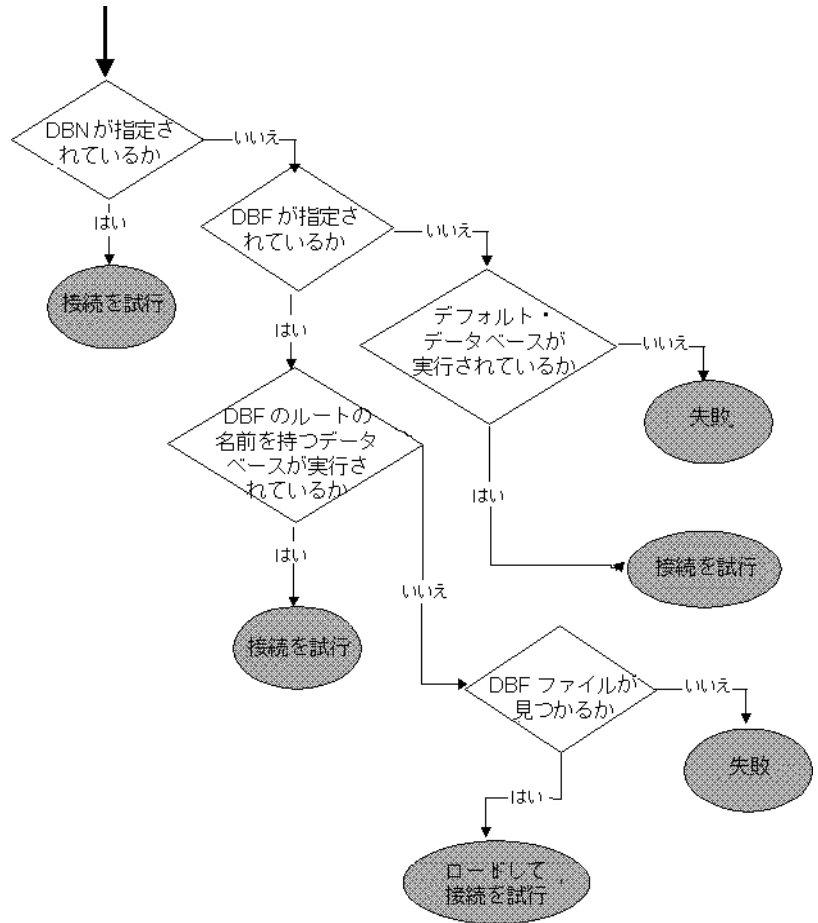
### 注意

- 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。次に例を示します。



### 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・ソースとして使用することができます。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 とのデータ型の対応 \(363 ページ\)](#)」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。

リモート・データベースで Sybase IQ を使用方法については、『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。次に例を示します。

```
-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、この情報はサーバを検出するためにのみ使用されます。このユーティリティでは、サーバを起動しません。

次の例では、**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=tcipip"
```

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 \(LOG\)」\(147 ページ\)](#)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効です。

## 統合化ログインの使用

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。

サポートされているオペレーティング・システム

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

### 統合化ログインの利点

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

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

統合化ログインを使用すると、ユーザにとって便利なだけでなく、データベースとネットワークの両方のセキュリティを 1 つのセキュリティ・システムで維持することができます。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 ID またはパスワードを入力する必要はありません。
- 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 とパスワードと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](#)」(113 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と、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と、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・マッピングは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 そのデータベースの [統合化ログイン] フォルダを開き、[ログイン・マッピング] をダブルクリックします。[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・マッピングを削除することができます。

### ❖ 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することができます。

- 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 DBISQL 文

```
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
- サーバからさらに接続情報を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 サーバに正しくログインした場合、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、データベースが置かれている Windows サーバ上に、ブランクのパスワードの “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することができます。

---

**注意** セキュリティ用の “Guest” ユーザ・プロファイルの無効化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](#)」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と、ユーザ・プロファイル **JSMITH** を明示的に識別する統合化ログイン・マッピングがない場合は、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 Java では、ユーザが [SQL] - [切断] (Interactive SQL Classic の場合は [コマンド] - [切断]) を選択した場合。
- Sybase Central では、ユーザが [ツール] - [切断] を選択した場合。
- DBISQL 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。

- DBISQL 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 [接続しているユーザ] 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。文構造については、『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 データベースの管理](#)」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グ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、DBISQL 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、「[第 14 章 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や、標準以外のソフトウェアを使用している場合、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、問題点を診断することができます。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、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に必要です。

サーバ名を指定せずに、データベース名を指定した場合、DBISQL は現在のサーバ上の指定したデータベースに継続しようとします (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、DBISQL 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、問題点を診断することができます。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することができます。
- 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 store (*dbname.iq*)、catalog store (*dbname.db*)、IQ temporary store (*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。かつてで困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。
- リストアが成功したことを確認します。

詳細については、「[Sybase IQ の接続の確立方法](#)」(98 ページ) を参照してください。

## 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」(51 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発行します。

```
kill -hup pid
```

`pid` は、停止させるデータベース・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、「[第 14 章 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 DBISQL の停止

通常は、DBISQL を終了させるには [ファイル] - [終了] を選びます。緊急時に UNIX コマンド・レベルから DBISQL を停止するときは、正しいプロセスを停止するように注意してください。DBISQL を起動すると、2 つのプロセスが起動します。1 つは `dbisql` 実行ファイルを実行するシェル・プロセスで、もう 1 つは `$$SYBASE/IQ-15_1/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。

まず、次のようなコマンドを使用して、両方のプロセスを探します。

```
ps -ef | egrep "jre|java|dbisql"
ami 1712 21442 0 12:01:17 pts/11 0:00 /bin/sh /mysystem/ami/IQ-15_1/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_1/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
```

## UNIX での DBISQL ウィンドウの問題の解決

UNIX での対話型 DBISQL ユーティリティでは、文字ベースのウィンドウを使用します。このウィンドウでは、ASCII 文字セットの一部である線描画文字と他の文字セットを使用します。DBISQL がこのウィンドウを正しく表示する機能は、使用する端末の種類とオペレーティング・システムが使用する文字セット変換によって異なります。ウィンドウが線描画文字ではなく強調文字で表示される場合は、このマニュアルの説明どおりにコマンドを入力したり、出力を確認したりできます。

DBISQL では、ファンクション・キーをさまざまな操作に使用します。特別な調節をしないかぎり、一部の UNIX のウィンドウ環境では、ファンクション・キーがサポートされていません。

DBISQL ウィンドウの外観を向上させたい場合や、DBISQL 内でファンクション・キーが使用できない場合の対処方法については、『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。



# 接続パラメータと通信パラメータ

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        | 125 |
| ネットワーク通信のパラメータ | 152 |

## 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す。その他のパラメータは、[詳細] タブにあるテキスト・ボックスに入力できます。

使用法についての注釈

接続パラメータでは、大文字と小文字を区別しません。

各接続パラメータの使用法のところで、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](#)」(71 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、OLE クライアントからデータベース・サーバに送信されるだけでなく、Interactive SQL (Windows 上の Classic および UNIX 上の DBISQLC) から送信されます。Open Client アプリケーションや jConnect アプリケーション (Java バージョンの 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することができます。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 (UNIX のみ)
- **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 からサンプル・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_1¥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)。

```
HOST=machine-name;
OS=Windows XP Build 2600 Service Pack 1;
PID=0x724;
THREAD=0x6bc;
EXE=C:¥Program Files¥Sybase¥IQ-
15_1¥win32¥dbisqlg.exe;
VERSION=9.0.2.1021;
API=ODBC;
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。<br>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_1\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"> <li>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。</li> </ul> <pre>[My Sample Database] DatabaseFile=c:\sybase\IQ-15_1\demo\iqdemo.db Autostart=No</pre> |

```
UserID=DBA
ENG=network_server
```

### 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_1¥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、「 <a href="#">ロケール情報の決定</a> 」(493 ページ)を参照してください。      |
| 説明    | CharSet に値を指定すると、指定された文字セットが現在の接続に使用されます。CharSet=none に設定すると、接続の文字セット変換が無効になります。 |

データのアンロード時に、CharSet 接続パラメータを使用して、文字セットを指定できます。有効な文字セット値については、「[ロケールの設定](#)」(493 ページ)を参照してください。

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 接続パラメータを使用しないことをお勧めします。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。

参照

[「接続パラメータの構造」](#) (64 ページ)

## CommBufferSize 接続パラメータ (CBufferSize)

**機能** 通信パケットの最大サイズをバイトで設定します。

**使用法** すべての場所

**値** 整数値

**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・サイド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」\(152 ページ\)](#)

『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に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。
- **TCPIP** 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](#)」(152 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る。

追加のネットワーク通信パラメータを各リンクに指定すると、リンクのブロードキャスト動作を調整できます。

CommLinks パラメータは、データベース・サーバの `-x` 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),shmem`

## 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** まだ実行されていないデータベースを起動するときに使用します。DatabaseFile 接続パラメータには、ロードして接続するデータベース・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、「[リスト後の再接続](#)」(539 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「 <a href="#">第 1 章 データベース・サーバの実行</a> 」                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 説明    | <p>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。</p> <ul style="list-style-type: none"><li>• DatabaseFile パラメータと同じ名前 (ただし .db 拡張子なし) のデータベースがロードされている場合は、そのデータベースに対して接続が確立されます。</li><li>• ファイル名に拡張子を指定しなかった場合は、.db のファイルが検索されます。</li><li>• 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。</li><li>• 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。</li></ul> <p>UNC ファイル名を使用することもできます。</p> |

---

**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 | 「 <a href="#">DatabaseName 接続パラメータ (DBN)</a> 」                                                                            |
| 例  | デモ・データベース (Windows の場合は <i>C:\Program Files\Sybase\IQ-15_1\demo</i> 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_1¥demo¥iqdemo.db
```

## DatabaseName 接続パラメータ (DBN)

|       |                                                                                                                                                                                                                                                                                |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | <p>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。接続先となるロード済みデータベースを指定します。</p> <p>すでに実行中のデータベースに接続する場合は、<b>DatabaseFile (DBF)</b> パラメータを使用します。</p>                                                                                                                                              |
| 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    | <p>サーバ上でデータベースが起動されている場合は、必ずデータベース名が割り当てられています (管理者が <b>-n</b> オプションを使用して割り当てるか、サーバがファイル名のベースに拡張子を付けてパスを削除したものを割り当てます)。</p> <p>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。</p> <p>実行中のデータベースの名前と <b>DatabaseName (DBN)</b> 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。</p> |

---

**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 \(DBN\)](#)」(156 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することができます。

```
DBN=Kitchener;DBF=cities.db
```
  - ただし、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    | データベース起動時に、データベースに指定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「 <a href="#">第 1 章 データベース・サーバの実行</a> 」<br>「 <a href="#">StartLine 接続パラメータ (START)</a> 」                                                                                                                                                                               |
| 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、<b>DatabaseSwitches</b> を使用してください。<b>DatabaseFile</b> 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 <b>DatabaseSwitches</b> 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 <b>START 接続パラメータ</b> 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。このコマンドを使うと、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_1/demo/iqdemo.db</i> (DBF パラメータ) をロードし、そのファイルに <b>my_db</b> (DBS パラメータ) という名前を付け、その名前のデータベース (DBN パラメータ) に接続します。</p>                                                                                |

```
dbcollat -c
"uid=DBA;pwd=SQL;dbf=/IQ-15_1/demo/iqdemo.db;
dbn=my_db;dbs=-n my_db" /tmp/temp.col
```

## 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 NT 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   | すべての場所                                                                                                                                                   |
| デフォルト | デフォルトのデータ・ソース名なし                                                                                                                                         |
| 参照    | 「 <a href="#">FileDataSourceName 接続パラメータ (FileDSN)</a> 」(142 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  | 次のパラメータは、データ・ソース名を使用します。<br><br>DSN=Dynamo Demo                                                                                                                           |

## 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「 <a href="#">データベース・サーバの起動</a> 」。                                                                                                                                    |
| 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 <i>marvin.db</i> 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。<br><br>...<br>UID=dba;PWD=sql;DBF=marvin.db;DBKEY='is!seCret'<br><br>... |

## 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<br><br>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 ServerName と同意語です。接続する、実行中のデータベース・サーバ名です。

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・オプション・パラメータ」。

「接続パラメータの構造」(64 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    | <p>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。</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> |
| 参照    | <a href="#">「接続パラメータの構造」 (64 ページ)</a> 。                                                                                                                                                                |

## 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   | <p>ECC_TLS (Certicom) と RSA_TLS は TCP/IP のみです。</p> <p>NONE または SIMPLE は、すべての場所です。</p>                                                                                                                                                                                                                                                                                                                                                                          |
| 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でない場合は、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。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"> <li>• <b>none</b> 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。</li> <li>• <b>simple</b> 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。</li> </ul> |

- **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。
  - **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。
  - **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 Server — データベース管理ガイド』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=tcip; 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=tcip; 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 | <p><a href="#">「DataSourceName 接続パラメータ (DSN)」 (137 ページ)</a>。</p>                                                                                                                         |

## 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   | すべての場所 (TDS 接続および共有メモリ接続を除く) 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することができます。</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    | <p>『ユーティリティ・ガイド』の「<a href="#">第 1 章 データベース・サーバの実行</a>」の <code>-ti</code> サーバ・コマンド・ライン・オプション</p>                                                                                                                                                                                                                                                                                                          |

- 例
- 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。

```
"ENG=myeng;LINKS=tcip;IDLE=10"
```

## 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 | NO                                                                                                                                                                                                                                                                                                                         |
| 参照    | 『リファレンス：文とオプション』の「 <a href="#">第 2 章 データベース・オプション</a> 」の「 <a href="#">LOGIN_MODE オプション</a> 」。                                                                                                                                                                                                                              |
| 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"><li><b>Yes</b> 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。</li><li><b>No</b> 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。</li></ul> <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、「<a href="#">ロケールの設定</a>」(493 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 | <p><a href="#">「接続パラメータの構造」</a> (64 ページ)。</p>                                                                                                                                                                                                                                                                                                                              |

## LazyClose 接続パラメータ (LCLOSE)

|       |                                                                                                                                                                                                                                                                                                           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | <p>このオプションを有効にすると、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一緒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。</p>                                                                                                                                                                     |
| 使用法   | <p>すべての場所</p>                                                                                                                                                                                                                                                                                             |
| 値     | <p>YES、NO</p>                                                                                                                                                                                                                                                                                             |
| デフォルト | <p>NO</p>                                                                                                                                                                                                                                                                                                 |
| 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"><li>・ ネットワークの遅延時間が長い場合</li><li>・ 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</li></ul> |

めったにない状況ですが、CLOSE *cursor-name* 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。その後に、同じ名前でのカーソルをオープンしようとするとう失敗します。要求を頻繁にキャンセルするアプリケーションでは、LazyClose を使用しないことをお勧めします。

参照 [「接続パラメータの構造」 \(64 ページ\)](#)。

### LivenessTimeout 接続パラメータ (LTO)

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能    | 使用されていない接続の終了を制御します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 使用法   | TCP/IP 通信プロトコル上のネットワーク・サーバ<br>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<br><br>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    | <p>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。</p> <p>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。</p> <p>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。</p> <p>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。</p> |

別の方法として、[ODBC 設定] ダイアログ・ボックスの [ネットワーク] タブの [LivenessTimeout] テキスト・ボックスに値を入力して、このパラメータを設定することもできます。

例 次の例は、Liveness Timeout 値を 60 秒に設定します。

```
LTO=60
```

## LogFile 接続パラメータ (LOG)

|     |                                                                                                                                                                                                                                                  |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 機能  | 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送信します。                                                                                                                                                                                                          |
| 使用法 | すべての場所                                                                                                                                                                                                                                           |
| 値   | 文字列                                                                                                                                                                                                                                              |
| 参照  | <a href="#">「接続パラメータの構造」 (64 ページ)</a> 。                                                                                                                                                                                                          |
| 説明  | <p>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。</p> <p>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。</p> <p>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 引数を設定することができます。</p> |
| 例   | <p>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 <code>error.log</code> ファイルに送信することを指定しています。</p> <pre>... LogFile=error.log ...</pre>                                                                                            |

## NewPassword 接続パラメータ (NEWPWD)

|    |                                                       |
|----|-------------------------------------------------------|
| 機能 | 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。        |
|    | 詳細については、『SQL Anywhere Server — データベース管理ガイド』を参照してください。 |

## 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    | <a href="#">「EncryptedPassword 接続パラメータ (ENP)」 (140 ページ)</a> 。                                                                                                                                                                                                                                                                                                           |
| 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> <pre>uid=DBA;pwd=SQL</pre> <p>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。</p>                                                                                                                                                                                                      |

## PrefetchBuffer 接続パラメータ (PBUF)

|       |                                                |
|-------|------------------------------------------------|
| 機能    | ロー・バッファリングのための最大メモリ量をキロバイト単位で設定します。            |
| 使用法   | すべての場所                                         |
| 値     | 整数値                                            |
| デフォルト | 64 (KB)                                        |
| 参照    | <a href="#">PrefetchRows 接続パラメータ (PROWS)</a> 。 |

**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当てられるメモリの量を制御します。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の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** 次の接続文字列フラグメントを使用すると、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できます。

```
...prefetchrows=100;logfile=c:¥ client.txt
```

次の文字列では、メモリの上限を 256KB に増やします。

```
...prefetchrows=100;prefetchbuffer=256
```

## PrefetchRows 接続パラメータ (PROWS)

**機能** データベースのクエリ時にプリフェッチされるローの最大数を設定します。

**使用法** すべての場所

**デフォルト** 10

**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)。

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 Server — データベース管理ガイド』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 \(ENG\)](#)」(139 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、*dbstop* によってサーバを停止します。
- 使用法                    すべての場所
- デフォルト                No
- 参照                      『ユーティリティ・ガイド』の「[dbstop](#)」
- 説明                      *dbstop* コマンドライン・ユーティリティはデータベース・サーバを停止します。接続文字列中に **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
- 次のコマンド・ラインでは、無条件にサーバを停止します。
- ```
dbstop -c "uid=DBA;pwd=SQL;eng=server-name;unc=yes"
```

## Userid 接続パラメータ (UID)

- 機能                      データベースにログオンするユーザ ID です。
- 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。
- ```
uid=DBA;pwd=SQL
```

## 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)   |

| TCP/IP                       | HTTP および HTTPS           |
|------------------------------|--------------------------|
| 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)

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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>BROADCAST は、TCP/IP プロトコルの実装でブロードキャスト・メッセージを識別するために使われる IP アドレスを指定します。</p> <p>ブロードキャスト・アドレスは、ネットワーク IP アドレス部分に加え、残りの整数部分をすべて 255 にしたものです。次に例を示します。</p> <ul style="list-style-type: none"> <li>ネットワーク部分が 95 (クラス A) の場合、ブロードキャスト・アドレスは 95.255.255.255 になります。</li> <li>ネットワーク部分が 132.10 (クラス B) の場合、ブロードキャスト・アドレスは 132.10.255.255 になります。</li> <li>ネットワーク部分が 197.31.175 (クラス C) の場合、ブロードキャスト・アドレスは 197.31.175.255 になります。</li> </ul> <p>ブロードキャスト IP アドレスが 255.255.255.255 の場合は、すべてのサブネットにブロードキャストを行います。</p> |

## BroadcastListener 通信パラメータ (BLISTENER)

|     |             |
|-----|-------------|
| 使用法 | TCP/IP、サーバ側 |
| 値   | YES、NO      |

|       |                                                                                                                                                     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| デフォルト | YES                                                                                                                                                 |
| 説明    | <p>このオプションでは、このポートのブロードキャスト受信を無効にできます。</p> <p>-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。</p>                                      |
| 参照    | 『ユーティリティ・ガイド』の「 <a href="#">データベース・サーバの起動</a> 」                                                                                                     |
| 例     | <ul style="list-style-type: none"><li>BroadcastListener=NO の TCP/IP 接続を受け入れるサーバを起動します。</li></ul> <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     | <ul style="list-style-type: none"><li>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。</li></ul> <pre>start_iq -xs<br/>http(Certificate=cert.file;Certificate_Password=secret) ...</pre> |

|    |                                                                                                                                                              |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 参照 | <a href="#">「Host パラメータ (IP)」 (158 ページ)</a><br><a href="#">「DoBroadcast パラメータ (DBROAD)」 (156 ページ)</a><br><a href="#">「ServerPort パラメータ (PORT)」 (166 ページ)</a> |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 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=tcip{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, <i>database-name</i>                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| デフォルト | AUTO                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 説明    | <p>Web 要求を処理するときに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。</p> <p>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。</p> <p>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。</p> <p>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。</p> |
| 例     | <ul style="list-style-type: none"><li>• 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。</li></ul>                                                                                                                                                                                                                                                                                                                                                             |

```
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) |

## デフォルト

YES、NO (サーバ側)

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
```

## DLL パラメータ

|       |                                                                                                                                             |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 使用法   | TCP/IP (Windows)                                                                                                                            |
| 説明    | 必要なネットワーク・インタフェース機能がデフォルトのプロトコル・スタックとは異なる DLL 内にある場合に、テストされていない TCP/IP プロトコル・スタックをサポートするために使われます。クライアントまたはサーバは、必要な機能を指定の DLL 内で検索するようになります。 |
| 値     | 文字列                                                                                                                                         |
| デフォルト | サポートされているすべての Windows プラットフォームでは、デフォルトは <code>ws2_32.dll</code> (Winsock 2.0) になります。                                                        |
| 例     | 次のコマンドは、 <code>abc.dll</code> 内にあるプロトコル・インタフェース機能を使用してサーバを起動します。<br><pre>iqsrv15 -x tcpip(dll=abc.dll) iqdemo</pre>                         |

## 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  | <a href="#">「ClientPort パラメータ (CPort)」 (155 ページ)</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 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.nn.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> |
| 値   | 文字列                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |



|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| デフォルト | 追加のマシンなし                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 例     | <ul style="list-style-type: none"> <li>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、iqdemo 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。</li> </ul> <pre>...ENG=iqdemo CommLinks=tcpip(IP=kangaroo;IP=197.75.209.222:2369)</pre> <ul style="list-style-type: none"> <li>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。</li> </ul> <pre>dbisql -x "tcpip(HOST=kangaroo;HOST=197.75.209.222)" iqdemo</pre> <ul style="list-style-type: none"> <li>次の接続文字列フラグメントは、my_server と kangaroo というマシンを検索して、データベース・サーバを見つけるようにクライアントに命令します。クライアントは、最初に応答を返したホストに対して接続を確立しようとします。</li> </ul> <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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 説明    | <p>デフォルトの <i>filename</i> は <i>asaldap.ini</i> です。</p> <p>データベース・サーバを LDAP サーバに登録し、クライアント (および Locate ユーティリティ [dblocate]) が LDAP サーバにクエリを発行できるようにします。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 (dblocate) もこのようなサーバを検索できるようになります。</p> <p>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>asaldap.ini</i> が設定ファイルとして使用されます。</p> <p>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。</p> |

参照 詳細については、『SQL Anywhere Server — データベース管理ガイド』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 NO

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 [dblocate] ユーティリティでもこのサーバを検出できません。LocalOnly (LOCAL) 通信パラメータを YES に設定してサーバを実行すると、そのネットワーク・サーバを接続や CPU の制限がないパーソナル・サーバとして実行できます。

参照 [「Broadcast 通信パラメータ \(BCAST\)」 \(153 ページ\)](#)

### 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 \(LF\)」 \(161 ページ\)](#)

[「LogMaxSize 通信パラメータ \(LSIZE\)」 \(162 ページ\)](#)

[「LogOptions 通信パラメータ \(LOPT\)」 \(162 ページ\)](#)

## 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 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。

- @@ @ 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 \(LOG\)」 \(160 ページ\)](#)
  - [「LogMaxSize 通信パラメータ \(LSIZE\)」 \(162 ページ\)](#)
  - [「LogOptions 通信パラメータ \(LOPT\)」 \(162 ページ\)](#)

### LogMaxSize 通信パラメータ (LSIZE)

- 使用法 HTTP、HTTPS
- 値 サイズ
- デフォルト 0
- 説明 ログ・ファイルが指定のサイズに達すると、ファイル名が変更され、別のログ・ファイルが作成されます。LogMaxSize を 0 に設定した場合は、ログ・ファイルのサイズが無制限になります。
- 参照
- [「Logfile 接続パラメータ \(LOG\)」 \(160 ページ\)](#)
  - [「LogFormat 通信パラメータ \(LF\)」 \(161 ページ\)](#)
  - [「LogOptions 通信パラメータ \(LOPT\)」 \(162 ページ\)](#)

### 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 \(LOG\)」 \(160 ページ\)](#)

[「LogFormat 通信パラメータ \(LF\)」 \(161 ページ\)](#)

[「LogMaxSize 通信パラメータ \(LSIZE\)」 \(162 ページ\)](#)

## MaxConnections 通信パラメータ (MAXCONN)

|       |              |
|-------|--------------|
| 使用法   | HTTP、HTTPS   |
| 値     | 番号           |
| 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 \(MAXSIZE\)」 \(164 ページ\)](#)

### MaxRequestSize 通信パラメータ (MAXSIZE)

**使用法** HTTP、HTTPS

**値** サイズ

**デフォルト** 100KB

**説明** サーバが受け付ける要求の最大サイズを指定します。要求のサイズがこの制限を超えている場合は、接続がクローズされ、クライアント側に 413 ENTITY TOO LARGE 応答が返されます。この値は要求のサイズのみを制限します (応答のサイズは制限しません)。0 を指定すると無制限になりますが、その場合は十分な注意が必要です。要求のサイズを制限しないと、悪意のあるクライアントがサーバに過大な負荷をかけたり、サーバのメモリを使い果たしたりするおそれがあります。

**参照** [「MaxConnections 通信パラメータ \(MAXCONN\)」 \(163 ページ\)](#)。

### MyIP 通信パラメータ (ME)

**使用法** TCP/IP、HTTP、HTTPS

**値** 文字列

**説明** MyIP パラメータは、複数のネットワーク・アダプタを持つマシン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、もともとは、この処理を行うと大きな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・カードを持つマシンや、リモート・アクセス (RAS) ソフトウェアとネットワーク・カードを持つマシンなど) のために用意されたものです。そのため、サーバの使用は想定されていません。

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。

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。

```
iqsrv15 -x
tcpip (MyIP=192.75.209.12:2367,192.75.209.32)
c:¥sybase¥IQ-15_1¥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 NO

**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・オープン要求が一緒に送信されます。これにより、カーソルをオープンするたびにロー・フェッチのネットワーク要求を送信せずに済みます。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                                                                                 |
| 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。 |

## SendBufferSize 通信パラメータ (SNDBUFSZ)

|       |                                                                                     |
|-------|-------------------------------------------------------------------------------------|
| 使用法   | TCP/IP                                                                              |
| 値     | 整数値                                                                                 |
| 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。 |

## 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 <b>2638</b> です。HTTP のデフォルト値は <b>80</b> です。HTTPS のデフォルト値は <b>443</b> です。                                                                                                                                                                    |
| 説明    | <p>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 <b>2638</b> を割り当てています。しかし、この予約ポートをアプリケーションで使うことが禁止されているわけではないため、データベース・サーバと別のアプリケーションの間でアドレスが競合する可能性があります。</p> <p>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。</p> |



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 が受信に使用されなくなります。

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_1¥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_1¥demo¥iqdemo2.db
```

これは、ポートが現在割り当てられているため、"Server failed to start. Possible cause: - port number invalid or already in use," というエラー・メッセージが発生して失敗します。

-z スイッチを使用している場合は、「指定されたアドレスにバインドできませんでした。」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_1¥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    | <p>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。</p> <p>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。</p> |
| 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     | <p>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。</p> <pre>iqsrv15 -x netbios(sesions=200) iqdemo.db</pre>                                                                                                                                          |

## 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    | <p>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)。</p>                                                    |
| 例     | <ul style="list-style-type: none"> <li>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。</li> </ul> <pre>start_iq -x tcpip(TDS=NO) ...</pre> |

## 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    | <p>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。</p> <p>HTTP または HTTPS アプリケーションでは、このパラメータを使用して、要求を受け取るときに許容される最大のアイドル時間を指定します。この限界に達した場合は接続がクローズされ、クライアント側に 408 REQUEST TIMEOUT が返されます。0 を指定するとアイドル・タイムアウトが無効になりますが、その場合は十分な注意が必要です。この制限がないと、悪意のあるクライアントがサーバのリソースを消費し、他のクライアントからの接続を妨害するおそれがあります。</p> |
| 例     | <p>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。</p>                                                                                                                                                                                                                                                                           |

```
...
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    | <p>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、マルチブレックス・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。</p> <p>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、そのポートに接続することができます。</p> <p>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。</p> |

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 アドミニストレータ内で新しい 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す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・バランス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 |
|-------------------------------------|-----|
| <a href="#">Sybase IQ データベースの作成</a> | 172 |
| <a href="#">データベース・オブジェクトの使用</a>    | 191 |
| <a href="#">DB 領域の使用</a>            | 206 |
| <a href="#">DB 領域管理の例</a>           | 216 |
| <a href="#">テーブルの編集</a>             | 221 |
| <a href="#">ビューの編集</a>              | 231 |
| <a href="#">インデックスの編集</a>           | 236 |

## 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<sup>®</sup> 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、DB 領域を管理できます。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、削除ができます。Sybase Central をマルチプレックス・サーバで使用するには、『Sybase IQ マルチプレックスの使用』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 データベースの管理](#)」を参照してください。

### DBISQL 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。DBISQL を使ってデータベース・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。それから DBISQL でこのファイルを実行して、データベースを作成します。

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、空のデータベースであると考えられます。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](#)」(459 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 ( .. );
..
```

DBISQL コマンド・ファイルには通常、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・ウィンドウへのファイル読み込みに関する詳細については、『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 または `CREATE DATABASE` 文を使用できます。

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](#)」(194 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](#)」(192 ページ)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・エディタを使用します。「[テーブルの編集](#)」(221 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] 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。「[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 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、これらのデバイスでオペレーティング・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 テンポラリー・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 |
| 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      |

| 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_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](#)」(57 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 15.1 へのアップグレード時に作成されます。IQ\_SYSTEM\_MAIN は、特別な DB 領域で、データベースが IQ チェックポイント・ログ、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、データベース領域および identity メタデータを開くために必要な構造を含んでいます。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 \(183 ページ\)](#)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。

次に例を示します。

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

### 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 データベースのアップグレード時に作成されます。

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 DB 領域を作成できます。

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。DB 領域のサイズは、予約した最大領域まで拡大できます。また、DB 領域を小さくすることもできます。その場合は、DB 領域のトランケートされる部分からすべてのデータを移動しておく必要があります。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 DB 領域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。

### DB 領域の作成時期

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 領域がより均等に埋められ、ディスク I/O が向上します。

### IQ 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](#)」(607 ページ)を参照してください。

### テンポラリ・ストアに必要な領域

Sybase IQ は、明示的に定義したテンポラリ・テーブルに加えて、テンポラリ・ストアを一時的な結果領域として使用します(ロード中または削除中のソート、ハッシュ、ビットマップの一時的な結果をここに格納します)。テンポラリ・ストアに必要な領域量は、発行されたクエリのタイプ、同時使用の程度、データのサイズによって決まります。

### 開発データベースの作成

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。

## メイン・ストアとテンポラリ・ストアのサイズ調整ガイドライン

Sybase IQ 15.1 アーキテクチャでは、次のようなデータ格納領域に影響を与える変更があります。

- **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・リスト領域で使用中であり、ユーザ・データに使用できません。
- Sybase IQ 15.1 では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                                                                                                                                                                                  |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 空のスキーマの<br>iqunload -n 出力からの<br>ロード、または小さい<br>テスト・データベース<br>へのロード | メイン用 10GB、テンポラリ用 5GB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | CREATE DATABASE の<br>サイズは MB 単位です。<br>iqunload -n を使用するに<br>は、サーバは 12.7 ESD #5<br>以上である必要があります。                                                                                     |
| 新しい運用データベー<br>スの作成                                                | <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>IQ_SYSTEM_MAIN にロー・ディスクを使用する場合は、可能な限りマルチプル・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。</li> <li>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。</li> </ul> | ロー・デバイスを指定する<br>場合は、ms_size を省略<br>します。<br><br>メイン用の予約は常に<br>IQ_SYSTEM_MAIN サイズ<br>の 20 パーセントに設<br>定します。メイン用の予<br>約を設定するには、デー<br>タベース・オプション<br>MAIN_RESERVED_DBSP<br>ACE_MB を使用します。 |
| マルチプレックス用の<br>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 では、セーブポイントの解放、コミット、およびチェックポイント操作で使用する、小さいが重要なデータ構造体のために、IQ メイン・ストア内で予約される領域の量を制御します。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](#)」(202 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 単位ではなく、MB 単位の値が必要です。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。マイグレーションの詳細については、『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/rdisk/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 DB 領域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、DB 領域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や使用方法については、『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" で返されます。

- 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_iquestjoin` 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_iquestjoin table1, table1rows, table2, table2rows  
[, relation] [, iqpagesize] ...
```

`sp_iquestjoin` 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_iquestjoin` を実行するごとに、示されたインデックス・サイズから 1 つ選択します。常にテーブルを 1 対 1 でジョインする場合は、"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可能性がある場合は、"Avg Case index\_size" を使用します。1 対多のジョインを数多く使用する可能性がある場合は、"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_iquestdbspaces` 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 で計算した最小と最大のインデックス合計サイズを、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_iquestdbspaces (dbsize [,iqpagesize]
                  [,minsize] [,maxsize] ...
```

これらの計算はすべて見積もりです。実際の結果は、データベースに作成したカラムとインデックスによって異なります。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

`SYSDATABASE` 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 メインおよびテンポラリ・データベースの `dbfile`、トランザクション・ログ・ファイル、`SA` テンポラリ・ファイルなど、データベース内のすべての `dbfile` が表示されます。

`dbfile` 以外のファイルは、`SYSDATABASE` 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・プロパティとしては表示されず、`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()` によって判断される場合があります (構文の詳細については、[表 14-1 \(598 ページ\)](#) を参照してください)。

## 範囲分割

「分割」は、ラージ・オブジェクトをサブオブジェクトに分けるスキーマです。たとえば、次のようになります。

- 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、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、次のような親テーブルの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に設定して確定し、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 15.1 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、あるいはすべての 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 15.1「[範囲分割](#)」では、テーブルが1つのテーブル・カラムの値に基づいて、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 パーミッションが必要です。                        |

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は、**START ROW ID** オプションを使用してロードできません。また、部分幅ロードおよび挿入は、分割されたテーブルではサポートされていません。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](#)」(323 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。次に例を示します。

```
CREATE TABLE tab1 (
    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 tab1(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。次に例を示します。

```
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                         |
| 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 のような対話型の SQL ツールからデータベースを構築する場合は、SQL 文を直接使うことができます。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。

| DB 領域名         | 目的                   | デフォルトの<br>オペレーティング・<br>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>                  |

システム DB 領域にあるシステム・テーブルは、データベース作成時のスキーマ定義を保持します。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、データベース・ファイルもデータも持たないデータベースである「ユーティリティ・データベース」を起動します。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、データベースを作成するエンジンとして [ユーティリティ・サーバ] 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](#)」(387 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は、ファイルを移動するときに、BACKUP コマンドを使用してフル・バックアップを取り、そのバックアップを RENAME オプション付きの RESTORE コマンドでリストアします。「[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 DB 領域と同じサイズでなければなりません。また、移動先のパーティションは、DB 領域を開始するために同じパスを持っていなければなりません。

---

**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、DB 領域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](#)」(8 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には、『Sybase IQ の概要』の「[第 4 章 データベースの管理](#)」の「[データベースの作成](#)」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には、『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

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。

- 大文字と小文字を区別します (CASE RESPECT)。つまり、‘ABC’ と ‘abc’ の比較は NOT EQUAL になります。デフォルト・ログインのユーザ ID が DBA (大文字) になり、パスワードが sql (小文字) になることに注意してください。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、短いほうの文字列を長いほう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 テンポラリー・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 テンポラリー・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に修飾されたパス名を使用して、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作成したデータベースは、データベースが配置されたディレクトリから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後から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て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](#)」(213 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すが、この追加領域には新しいデータしか格納できません。特定のインデックス、テーブル、データのローを、あるロケーションから別のロケーションへと簡単には移動できません。

各オペレーティング・システムには、ロー・デバイス名に関する独自のフォーマットがあります。Sun Solaris でのロー・デバイスの初期化に関する重要な注意事項については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[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・デバイスの名前

| UNIX プラットフォーム | 例                   |
|---------------|---------------------|
| AIX           | /dev/rraw12lv       |
| HP-UX         | /dev/vg03/rrchee12g |
| Sun Solaris   | /dev/rsd0c          |
| Sun AMD       | /dev/rdisk/c5t0d0s1 |

表 5-3：Windows 上でのロー・デバイスの名前

| 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 <del>¥¥¥C:</del> 、SQL では <del>¥¥¥¥¥¥C:</del>                                              |
| パーティションなし | <i>PhysicalDriveN</i> : <i>N</i> は 0 から始まる数値で、必要なだけ大きくできる。物理ドライブ番号を調べるには、[管理ツール] の [ディスク アドミニストレータ] を実行する。 | Sybase Central では <del>¥¥¥</del><br><i>PhysicalDrive32</i> 、SQL では <del>¥¥¥¥¥¥</del> <i>PhysicalDrive32</i> |

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・デバイスの名前を SQL 文で指定するときには、円記号を必ず二重にする必要があります。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 次に示す UNIX の例では、IQ PATH で指定したロー・パーティションに **company** という名前の 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 次に示す Windows の例では、IQ PATH で指定したロー・パーティションに **company** という名前の 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・サイズを選択

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 より小さい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・プラットフォーム上のデータベースで、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。これらの設定と、リソースの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、通常の場合よりも大きな 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 パラメータはいくつかのブロックを範囲として予約するため、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                     | 16MB                       |

カタログ・ページ・サイズの選択

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/rdsk/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。これを、**JAVA OFF** オプションで OFF にすることができます。Java が ON の場合、次のことが可能です。

- カタログ・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とエラーになります。
- Java の API (Application Programmer's 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 オンライン・ヘルプ \(http://info.center.sybase.com/help/index.jsp\)](http://info.center.sybase.com/help/index.jsp) の SQL Anywhere Studio のマニュアルにある『SQL Anywhere Server — プログラミング』の「データベースにおける Java の概要」
- 『システム管理ガイド：第 2 巻』の「[付録 A 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、コミット、セーブポイントの解放といった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](#)」(607 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切であ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 目的のサーバを開きます。
- 2 目的のデータベースを右クリックし、ポップアップ・メニューから [オプション] を選択します。
- 3 目的の値を編集します。

### ❖ データベース・オプションの設定 (SQL)

- SET OPTION 文の中で、目的の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等のシステム・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 目的のサーバを開きます。
- 2 接続されている目的のデータベースを右クリックし、[所有者／コンテンツ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

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、『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。

## データベースの削除

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

データベースを削除するには、次の SQL 文を使用します。

```
DROP DATABASE dbname
```

データベース名とそのパス名は、データベースの作成時に指定したものをまったく同じように指定します。

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](#)」(216 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。『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 つのタイプのオンライン状態があります。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。データベースが起動し、1 つまたは複数の dbfile を開くことができない場合は、データベースは起動しますが、DB 領域が動的にオフラインとマークされます。管理者は `ALTER DBSPACE ONLINE` を使用して、問題を解決してから DB 領域をオンラインに戻すことができますが、これは DB 領域のメモリ内の状態のみを変更します。

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 領域の場合は、『[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 領域の場合、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、関連するファイルのカタログ状態は読み取り専用になりませんが、オペレーティング・システム・レベルでは、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読み取り専用または読み取り／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」(381 ページ) を参照してください。

DB 領域とその関連するファイルでは、個々の RO 状態または 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、RO DB 領域への割り付け時は読み取り専用です。テーブル・パーティションは、パーティションが RO DB 領域に割り付けられるときは読み取り専用です。RO テーブルまたは RO 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はできません。RO テーブルまたは RO 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・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、いずれかのジョイン・テーブルが RO である場合は失敗します。

次の表は、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            | 可能       | 可能                                   |
|-----------------|------------------|----------|--------------------------------------|
|                 |                  | ユーザ・メイン  | IQ_SYSTEM_MAIN、<br>IQ_SYSTEM_TEMP    |
| オンラインの<br>DB 領域 |                  |          |                                      |
|                 | DB 領域のオフラインの変更   | 可        | 不可                                   |
|                 | DB 領域のオンラインの変更   | 不可       | テンポラリの場合は不可、<br>IQ_SYSTEM_MAIN の場合は可 |
|                 | DB 領域の RO の変更    | RW の場合は可 | 不可                                   |
|                 | DB 領域の RW の変更    | RO の場合は可 | 不可                                   |
|                 | ストライピング・パラメータの変更 | 可        | シングル・ノードまたは<br>マルチプレックス・コーディネータの場合は可 |
|                 | DB 領域名の変更        | 可        | 不可                                   |
|                 | ファイルの追加          | 可        | 可                                    |
|                 | ファイルの削除          | 空の場合は可   | 空および RO の場合は可                        |
|                 | ファイルの RO の変更     | RW の場合は可 | RW で、かつ最後の RW<br>dbfile でない場合は可      |
|                 | ファイルの RW の変更     | RO の場合は可 | RO の場合は可                             |
|                 | ファイル・サイズの変更      | 可        | 可                                    |
|                 | ファイルの論理名の変更      | 可        | 可                                    |
|                 | ファイルのパス名の変更      | 不可       | 不可                                   |
|                 |                  |          |                                      |

| 状態              | 変更の種類            | 可能       | 可能 |
|-----------------|------------------|----------|----|
| オフライン DB 領域     |                  |          |    |
|                 | DB 領域のオフラインの変更   | 不可       | NA |
|                 | DB 領域のオンラインの変更   | 可        | NA |
|                 | DB 領域の RO の変更    | 不可       | NA |
|                 | DB 領域の RW の変更    | 不可       | NA |
|                 | ストライピング・パラメータの変更 | 可        | NA |
|                 | DB 領域名の変更        | 可        | NA |
|                 | ファイルの追加          | 不可       | NA |
|                 | ファイルの削除          | 空の場合は可   | NA |
|                 | ファイルの RO の変更     | 可        | NA |
|                 | ファイルの RW の変更     | 可        | NA |
|                 | ファイル・サイズの変更      | 不可       | NA |
|                 | ファイルの論理名の変更      | 可        | NA |
|                 | ファイルのパス名の変更      | 可        | NA |
| 動的にオフラインの DB 領域 |                  |          |    |
|                 | DB 領域のオフラインの変更   | RO の場合は可 | NA |
|                 | DB 領域のオンラインの変更   | 可        | NA |
|                 | DB 領域の RO の変更    | RW の場合は可 | NA |
|                 | DB 領域の RW の変更    | 不可       | NA |
|                 | ストライピング・パラメータの変更 | 可        | NA |
|                 | DB 領域名の変更        | 可        | NA |
|                 | ファイルの追加          | 不可       | NA |
|                 | ファイルの削除          | 空の場合は可   | NA |
|                 | ファイルの RO の変更     | 可        | NA |
|                 | ファイルの RW の変更     | 可        | NA |
|                 | ファイル・サイズの変更      | 不可       | NA |
|                 | ファイルの論理名の変更      | 可        | NA |
|                 | ファイルのパス名の変更      | 不可       | NA |

| 状態                 | 変更の種類            | 可能            | 可能            |
|--------------------|------------------|---------------|---------------|
| 読み取り専用<br>DB 領域    |                  |               |               |
|                    | DB 領域のオフラインの変更   | オンラインの場合は可    | NA            |
|                    | DB 領域のオンラインの変更   | オフラインの場合は可    | NA            |
|                    | DB 領域の RO の変更    | 不可            | NA            |
|                    | DB 領域の RW の変更    | オンラインの場合は可    | NA            |
|                    | ストライピング・パラメータの変更 | 可             | NA            |
|                    | DB 領域名の変更        | 可             | NA            |
|                    | ファイルの追加          | 可             | NA            |
|                    | ファイルの削除          | 空の場合は可        | NA            |
|                    | ファイルの RO の変更     | RW の場合は可      | NA            |
|                    | ファイルの RW の変更     | RO の場合は可      | NA            |
|                    | ファイル・サイズの変更      | 不可            | NA            |
|                    | ファイルの論理名の変更      | 可             | NA            |
|                    | ファイルのパス名の変更      | オフラインの場合は可    | NA            |
|                    |                  |               |               |
| 読み取り／書き込み<br>DB 領域 |                  |               |               |
|                    | DB 領域のオフラインの変更   | 不可            | 不可            |
|                    | DB 領域のオンラインの変更   | 動的にオフラインの場合は可 | 不可            |
|                    | DB 領域の RO の変更    | 可             | 不可            |
|                    | DB 領域の RW の変更    | 不可            | 不可            |
|                    | ストライピング・パラメータの変更 | 可             | 可             |
|                    | DB 領域名の変更        | 可             | 不可            |
|                    | ファイルの追加          | 可             | 可             |
|                    | ファイルの削除          | 空の場合は可        | 空および RO の場合は可 |
|                    | ファイルの RO の変更     | RW の場合は可      | RW の場合は可      |
|                    | ファイルの RW の変更     | RO の場合は可      | RO の場合は可      |
|                    | ファイル・サイズの変更      | RW の場合は可      | RW の場合は可      |

| 状態            | 変更の種類        | 可能         | 可能 |
|---------------|--------------|------------|----|
|               | ファイルの論理名の変更  | 可          | 可  |
|               | ファイルのパス名の変更  | 不可         | 不可 |
|               |              |            |    |
| 読み取り専用ファイル    |              |            |    |
|               | ファイルの RO の変更 | 不可         | 不可 |
|               | ファイルの RW の変更 | 可          | 可  |
|               | ファイル・サイズの変更  | 不可         | 不可 |
|               | ファイルの論理名の変更  | 可          | 可  |
|               | ファイルのパス名の変更  | オフラインの場合は可 | 不可 |
|               |              |            |    |
| 読み取り／書き込みファイル |              |            |    |
|               | ファイルの RO の変更 | 可          | 可  |
|               | ファイルの RW の変更 | 不可         | 不可 |
|               | ファイル・サイズの変更  | 可          | 可  |
|               | ファイルの論理名の変更  | 可          | 可  |
|               | ファイルのパス名の変更  | 不可         | 不可 |

注意

- 動的にオフラインとは、DB 領域がカタログでオフラインとマークされるのではなく、メモリ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

新しいデータベース・ファイル (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。新しい DB 領域を作成するには、DBA 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 によって、そのファイル・サイズが自動的に指定され、使用するデバイス全体が IQ ストアとして割り当てられます。デバイスが実際はファイル・デバイスである場合、期待どおりの結果が得られない可能性があるため、Sybase ではこの方法を推奨しません。

デバイスがロー・デバイスでないことを指定すると、ウィザードでファイル・サイズを指定できる [ファイル・サイズ] フィールドが有効になります。また、ウィザードでは、次のページに移動する前に、指定されたパスが存在することも確認されます。

### 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。ただし、最大数近くに達するような状況にならないようにしてください。DB 領域の数を増加しても、メモリ使用やパフォーマンスに実際的な影響はありません。

---

**注意** HP や AIX プラットフォーム上では、より多くの DB 領域にデータを分割すると、重複した I/O の使用が向上します。

---

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。いずれの場合でも、ストライプ・サイズは IQ ページ・サイズの数倍でなければなりません。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。領域量の見積もり方法については、「[必要な領域と DB 領域の見積もり](#)」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ファイル *library.iq* 内に、新しい DB 領域 *library* を作成します。

```
CREATE DBSPACE library
USING FILE library
'library.iq' SIZE 100 MB IQ STORE
```

### Sybase Central での DB 領域の作成

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。



よりクリーンなリカバリのためのチェックポイントの発行

DB 領域を追加または削除した後は、CHECKPOINT の発行をおすすめします。システム・リカバリが必要な場合、リカバリが最新のチェックポイントの後から開始します。

## DB 領域の削除

DROP DBSPACE コマンドを発行して、データベース・ファイルを削除できます。DB 領域を削除するには、次の条件が満たされている必要があります。

- ・ 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 領域を削除できません。
- ・ SYSTEM、IQ\_SYSTEM\_MAIN、IQ\_SYSTEM\_TEMP、IQ\_SYSTEM\_MSG は必須の DB 領域ではないこと。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、まず次の操作を行います。

- ・ 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 コマンドを使用する必要があります。

他のユーザが読み込んでいる最中のテーブルを移動すると、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。

DBA は、ストアド・プロシージャ `sp_iqspaceinfo`、`sp_iqdbspaceinfo`、および `sp_iqindexinfo` を実行することで、DB 領域テーブルおよびインデックスの場所を特定できます。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、ストアド・プロシージャ `sp_iqdbspace` を実行します。Block Types カラムには、各 DB 領域の内容が示されます。ブロック・タイプが 'H'、'F'、'R'、'B'、'X'、'C' のみの場合は、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からのデータを表します。どのテーブルを移動しなければならないかを判断するには、`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 CREATE DATABASE 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;
```

dbspacedb データベース内に 2 つのテーブルを作成し、インデックスを作成し、データを追加します。

```
CREATE TABLE t1(c1 int);
CREATE TABLE t2(c1 int);
CREATE hg INDEX t1c1hg ON t1(c1);
CREATE hng INDEX t2c1hng ON t2(c1);
INSERT t1 VALUES (1);
INSERT t2 VALUES (2);
COMMIT;
```

DB 領域に関する情報の表示

sp\_iqdbspace システム・ストアド・プロシージャを使用して、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。次の出力は、2 つの部分に分けて読みやすくしてあります。

| DBSpaceName    | DBSpaceType | Writable | Online | Usage | TotalSize | Reserve | NumFiles | NumRWFiles |
|----------------|-------------|----------|--------|-------|-----------|---------|----------|------------|
| IQ_SYSTEM_MAIN | MAIN        | T        | T      | 25    | 50M       | 100M    | 1        | 1          |
| IQ_SYSTEM_TEMP | TEMPORARY   | T        | T      | 7     | 10M       | 10M     | 1        | 1          |
| dbspacedb2     | MAIN        | T        | T      | 1     | 10N       | 20M     | 1        | 1          |
| dbspacedb3     | MAIN        | T        | T      | 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         |

sp\_iqdbspaceinfo システム・ストアド・プロシージャを使用して、特定の DB 領域のオブジェクト配置や領域の使用状況に関する情報を表示します。次の情報は、iqdemo DB 領域からの情報です。

```
sp_iqdbspaceinfo;
```

| dbspace_name | object_type | owner  | 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 | indexes | metadata | primary_key | unique_constraint | foreign_key |
|---------|---------|----------|-------------|-------------------|-------------|
| 96K     | 0B      | 1.37M    | 0B          | 0B                | 0B          |
| 24K     | 0B      | 464K     | 0B          | 0B                | 0B          |
| 96K     | 0B      | 1.22M    | 0B          | 0B                | 0B          |
| 288K    | 0B      | 5.45M    | 24K         | 0B                | 48K         |
| 240K    | 48K     | 4.63M    | 24K         | 0B                | 0B          |
| 72K     | 0B      | 1.78M    | 24K         | 0B                | 48K         |
| 408K    | 0B      | 8.03M    | 24K         | 0B                | 48K         |
| 72K     | 0B      | 1.53M    | 24K         | 0B                | 0B          |
| 96K     | 0B      | 2.19M    | 24K         | 0B                | 48K         |
| 272K    | 192K    | 4.67M    | 24K         | 0B                | 0B          |
| 120K    | 0B      | 2.7M     | 24K         | 0B                | 104K        |
| 144K    | 0B      | 3.35M    | 24K         | 0B                | 144K        |

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配置や領域の使用状況に関する情報を表示します。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 |
| GROUPO.Customers.ASIQ_IDX_T733_C4_FP  | iq_main     | 440K    | 1       | 1,046,737 | 1,047,189 |
| GROUPO.Customers.ASIQ_IDX_T733_C5_FP  | iq_main     | 440K    | 1       | 1,046,929 | 1,047,182 |
| GROUPO.Customers.ASIQ_IDX_T733_C6_FP  | iq_main     | 440K    | 1       | 1,047,009 | 1,047,175 |
| GROUPO.Customers.ASIQ_IDX_T733_C7_FP  | iq_main     | 440K    | 1       | 1,046,945 | 1,047,168 |
| GROUPO.Customers.ASIQ_IDX_T733_C8_FP  | iq_main     | 440K    | 1       | 1,046,785 | 1,047,161 |
| GROUPO.Customers.ASIQ_IDX_T733_C9_FP  | iq_main     | 440K    | 1       | 1,046,881 | 1,047,154 |
| GROUPO.Customers.ASIQ_IDX_T733_I11_HG | iq_main     | 152K    | 1       | 1,047,121 | 1,047,206 |
| GROUPO.Customers.IX_customer_name     | iq_main     | 304K    | 1       | 1,050,995 | 1,051,038 |

sp\_iqdbspace、sp\_iqdbspaceinfo、および sp\_iqindexinfo システム・ストアド・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。

#### DB 領域のサイズの変更

この項で紹介する ALTER DBSPACE コマンドは、DB 領域のサイズを必要に応じて変更する方法を示しています。

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 | Usage | TotalSize | Reserve | NumFiles | NumRWFiles |
|----------------|-------------|----------|--------|-------|-----------|---------|----------|------------|
| IQ_SYSTEM_MAIN | MAIN        | T        | T      | 25    | 20M       | 90M     | 1        | 1          |
| IQ_SYSTEM_TEMP | TEMPORARY   | T        | T      | 7     | 10M       | 10M     | 1        | 1          |
| dbspacedb2     | MAIN        | T        | T      | 1     | 10N       | 20M     | 1        | 1          |
| dbspacedb3     | MAIN        | T        | T      | 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         |

`dbspacedb` データベースが `IQ RESERVE` 値を指定しないで作成されていた場合は、DB 領域を拡張できません。ただし、DB 領域を小さくし、その DB 領域からトランケートした分の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。`ALTER DBSPACE` コマンドと `SIZE` パラメータを使用して、この DB 領域のサイズを `15MB` に変更できます。

```
ALTER DBSPACE IQ_SYSTEM_MAIN SIZE 15mb;

sp_iqdbspace;
```

| DBSpaceName    | DBSpaceType | Writable | Online | Usage | TotalSize | Reserve | NumFiles | NumRWFiles |
|----------------|-------------|----------|--------|-------|-----------|---------|----------|------------|
| IQ_SYSTEM_MAIN | MAIN        | T        | T      | 25    | 15M       | 95M     | 1        | 1          |
| IQ_SYSTEM_TEMP | TEMPORARY   | T        | T      | 7     | 10M       | 10M     | 1        | 1          |
| dbspacedb2     | MAIN        | T        | T      | 1     | 10N       | 20M     | 1        | 1          |
| dbspacedb3     | MAIN        | T        | T      | 1     | 10M       | 40M     | 1        | 1          |

| Stripingon | StripeSize | BlkTypes                   | OK ToDrop |
|------------|------------|----------------------------|-----------|
| T          | 1K         | 1H,1248F,32D,177A,<br>128M | N         |
| T          | 1K         | 1H,64F,16A                 | N         |
| T          | 1K         | 1H                         | Y         |
| T          | 1K         | 1H                         | Y         |

DB 領域のサイズを縮小できるのは、トランケートする部分が使用中でない場合にかぎられます。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。

## 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「[テーブルの管理](#)」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。

## SQL でのテーブルの作成

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` を指定してサンプル・データベースに接続します。

**DBISQL** からサンプル・データベースへ接続する方法の詳細については、「[Sybase Central または DBISQL からサンプル・データベースに接続する](#)」を参照してください。

**DBISQL** 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。

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](#)」(229 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 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
```

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。

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、その接続に名前が同じローカル・テンポラリ・テーブルがある場合に失敗します。これは、この新しいテーブルを *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 マルチプレックスの使用』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 UNIQUE が<br>256 以下の場合                   | IQ UNIQUE が<br>65536 以下の場合                                | IQ UNIQUE が<br>16777216 以下の場合                                | IQ UNIQUE が未指定または<br>16777216 を超える場合 |
|--------------------------------------------|-----------------------------------------------------------|--------------------------------------------------------------|--------------------------------------|
| 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、かつ、見積もりが大幅に間違っている場合は (たとえば実際のユニーク値の数が 1200 万個のときに IQ UNIQUE 1000000 と指定した場合など)、クエリのパフォーマンスが低下することがあります。

既存のインデックスの IQ UNIQUE 値を変更するには、sp\_iqrebuildindex プロシージャを実行します。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

次のコマンドは、前の例と同様に **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つのカラムから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** テーブルにはまた、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。



テンポラリ・テーブルには、ローカル、グローバル、自動のどの種類のテンポラリ・テーブルであっても外部キー制約を作成できません。

有効な文字列や識別子の詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「SQL 言語の要素」の章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

データベース内のテーブルに関する情報は、システム・ビュー SYS.SYSTAB 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・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 \(Sybase Central\)](#)」(204 ページ)の手順に従います。

## ビューの編集

### ビューとベース・ テーブルの類似点

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

### ビューと永久テーブルの相違点

- 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 つまたは複数のテーブルを操作し、それ自体がテーブルになっている結果セットを生成します。ベース・テーブルと同様に、SELECT 文の結果セットにも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
```

この文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には、『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。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。次に例を示します。

```
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 に所有されているビューは削除しないでください。これらのビューを削除したり、テーブルに変更したりすると、予想外の問題が起き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 次を示す例では、前述の **DepartmentSize** ビューのカラム名をもっとわかりやすい名前に変更します。

```
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

次を示す条件が 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作用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作用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作用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 [削除] 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

データベース内のテーブルに関する情報は、システム・ビュー SYS.SYSVIEW 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・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のタイプ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・インデックスの作成] 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・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。



# Sybase IQ インデックスの使用

## 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                                      | ページ |
|-------------------------------------------|-----|
| <a href="#">インデックスの概要</a>                 | 241 |
| <a href="#">Sybase IQ インデックスの作成</a>       | 246 |
| <a href="#">インデックス・タイプの選択</a>             | 248 |
| <a href="#">Sybase IQ インデックス・タイプ</a>      | 254 |
| <a href="#">データ挿入後のカラム・インデックスの追加</a>      | 276 |
| <a href="#">ジョイン・インデックスの使用</a>            | 277 |
| <a href="#">ジョイン・インデックスのサイズとメリットの見積もり</a> | 297 |

## 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 のインデックスはこのデータを表し、実際に格納して、さまざまなクエリ的高速化に使用できるようにします。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することができ、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](#)」(277 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。                                             |
| 日時または 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<br>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<br>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<br>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。                  |

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することができます。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 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 は 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 は 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
- 関連サブクエリ・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も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の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ることで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することができます。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)
```

2 つ目の例では、SalesOrder テーブルの Region カラムに、SalesOrderRegionIX という Low\_Fast 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](#)」(265 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します。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。

## 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することができます。

- `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、クエリの実行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わ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。この数があるレベルに達したら、表 6-2 の推奨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 65,536 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。次に例を示します。

- そのカラムはジョイン述部の一部になるか？
- カラム内にユニークな値が多数ある場合、そのカラムは GROUP BY 句で使用されるのか、COUNT DISTINCT の引数になるのか、SELECT DISTINCT 射影に含まれるのか、あるいはこれらのうち複数が当てはまるのか？
- 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。

次の表では、推奨されるインデックス・タイプをクエリの種類別に示します。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するカラム                              | デフォルト、HG、LF、                     |
| ジョイン・インデックスで使用するカラム                               | 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                   | LF、HG、または HNG。CMP、DATE、TIME、DTTM |
| DATEPART 等号述部、範囲述部、および IN                         | DATE、TIME、DTTM                   |
| リスト述部内                                            |                                  |
| CONTAINS 述部                                       | WD                               |

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 옵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の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、デフォルトのインデックス・タイプと CMP 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、デフォルトのインデックス・タイプ、CMP インデックス・タイプ、および WD インデックス・タイプでのみ使用できます。LONG VARCHAR データは、デフォルトのインデックス・タイプと WD インデックス・タイプで使用できます。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                                                           |
|----------------|----------------|----------------------------------------------------------------|
| デフォルト          | 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 |
| smallint     | CMP、HG、HNG、LF | WD、DATE、TIME、DTTM |
| int          | CMP、HG、HNG、LF | WD、DATE、TIME、DTTM |
| unsigned int | CMP、HG、HNG、LF | WD、DATE、TIME、DTTM |

| 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                  |
|--------------------------------|--------------------|---------------------------------|
| bigint                         | CMP、HG、HNG、LF      | WD、DATE、TIME、DTTM               |
| unsigned bigint                | CMP、HG、HNG、LF      | WD、DATE、TIME、DTTM               |
| numeric、decimal                | CMP、HG、HNG、LF      | WD、DATE、TIME、DTTM               |
| double                         | LF (HG も許容、ただし非推奨) | CMP、HNG、WD、DATE、TIME、DTTM       |
| float                          | LF (HG も許容、ただし非推奨) | CMP、HNG、WD、DATE、TIME、DTTM       |
| real                           | LF (HG も許容、ただし非推奨) | CMP、HNG、WD、DATE、TIME、DTTM       |
| bit                            | (デフォルト・インデックスのみ)   | CMP、HG、HNG、LF、WD、DATE、TIME、DTTM |
| date                           | CMP、HG、HNG、LF、DATE | WD、TIME、DTTM                    |
| time                           | CMP、HG、HNG、LF、TIME | WD、DATE、DTTM                    |
| datetime、timestamp             | CMP、HG、HNG、LF、DTTM | WD、DATE、TIME                    |
| char (255 バイト以下)、<br>character | CMP、HG、HNG、LF、WD   | DATE、TIME、DTTM                  |
| char (255 バイト超)                | CMP、WD             | HG、HNG、LF、DATE、TIME、DTTM        |
| varchar (255 バイト以下)            | CMP、HG、HNG、LF、WD   | DATE、TIME、DTTM                  |
| varchar (255 バイト超)             | CMP、WD             | HG、HNG、LF、DATE、TIME、DTTM        |
| long varchar                   | WD                 | CMP、HG、HNG、LF、DATE、TIME、DTTM    |
| binary                         | CMP、HG、LF          | HNG、WD、DATE、TIME、DTTM           |
| varbinary (255 バイト<br>以下)      | CMP、HG、LF          | HNG、WD、DATE、TIME、DTTM           |
| varbinary (255 バイト超)           | CMP                | HG、HNG、LF、WD、DATE、TIME、DTTM     |

## インデックス・タイプの組み合わせ

複数のタイプのクエリで使用されるカラムの場合は、1 つ以上のカラム・インデックス・タイプを使用することをおすすめします。表 6-6 に、有効な組み合わせのインデックス・タイプを示します。

表 6-6 : 有効な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 <sup>a</sup> | 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 = ありえない組み合わせ

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。

**注意** Sybase IQ Version 12 の High\_Group インデックスは、これまでのバージョンのものとは異なります。一部のカラムで、High\_Group と High\_Non\_Group の両方を使用できるようになっています。これまで、両方あっても意味をなしませんでした。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 以上の一意の値があるカラムに適用されます。

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が評価されるようになります。

各カラムには 1 つの FP インデックスがあり、それぞれの FP は *n* 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、または月のフィールドなど、重複しない値やユニークな値を使用すると、必要なディスク容量が減少して最適な FP 形式が作成され、射影のためのカラムおよび I/O コストに要する格納領域を画期的に削減することができます。

最適化されたこれらの FP インデックスは、2 つの部分から構成されています。1. カラム内にある重複しない各値が正確に 1 度だけ表示される検索テーブル。2. 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場所を示すキーになっている値の論理配列。

`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。この出力により、指定されたインデックスが 1 バイト、2 バイト、3 バイト、またはフラット・スタイルの FP インデックスであるかどうかを簡単に確認できます。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「[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

値の数が 256 以内の場合は、1 バイト FP インデックスが作成されます。FP インデックスはこの形式から始まります。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。

32MB のデフォルトのメイン・キャッシュの設定と、2500 のデフォルトの `FP_LOOKUP_SIZE_PPM` の設定では、LOAD、INSERT、または UPDATE から新しい一意の値が `char(2)`、`binary(2)`、または小さい整数の FP(1) カラムに追加され、FP(1) が FP に変換されます。

## FP(2) インデックス

値の数が 256 よりも大きく 65536 以下の場合は、同じ検索テーブルが生成されます。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

一意のカウントが 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。

- 検索テーブルの最大サイズは 16777216 です。65536 ではありません。
- 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。FP(3) は一意のカウントが 65536 を超えると FP(2) インデックスからのロールオーバー後に作成することもできます。しかし、表 6-7 に示されているデータ・サイズのみに対応し、データ型 BIT、TINYINT、SMALLINT、CHAR(<=3)、VARCHAR(<=3)、BINARY(<=3)、および VARBINARY(<=3) は除外されます。『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」と『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」を参照してください。

動作変更

旧リリースと比べて FP インデックスの動作に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・<br>サイズ = 1 バイト | カラム・データ・<br>サイズ = 2 バイト | カラム・データ・<br>サイズ = 3 バイト | カラム・データ・<br>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) インデックスのように、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ていません。

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 オプションのいずれか小さい方の値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。

$$\text{FP\_LOOKUP\_SIZE} / (\text{カラム・データ・サイズ} + \text{カーディナリティ・サイズ})$$

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**

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、テンポラリー・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、個別の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。テンポラリー・キャッシュに十分な割り当てが行われていると、オプション `HASH_PINNABLE_CACHE_PERCENT` 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留めることによってパフォーマンスを向上させることができます。

### 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一意の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 にも 16777216 を下回る値の場合の例が示されています。この場合、フラット `FP` へのロールオーバーは、見込まれている 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的なキャッシュ要件があります。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 エントリがフラットスタイルに変改される際のコストを抑えることができます。特定のフィールドの一意のロー数が 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 `IQ UNIQUE` 値を使用します。

- 3 バイトの FP インデックスの場合 — `IQ UNIQUE` は 65537 ～ 16777216 の値にする必要があります。
  - フラット・スタイルの FP インデックスの場合 — `IQ UNIQUE` は 16777216 より大きい値にする必要があります。
- 

## 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、一度ルックアップを行うだけで素早く結果セットが出ます。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。<br>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い場合にだけ、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くの範囲クエリでは、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](#)」の「[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。<br>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)。<br>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には、『パフォーマンス & チューニング・ガイド』の「 <a href="#">削除オペレーションの最適化</a> 」を参照してください。<br>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。<br>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大きい場合は、High\_Group を使用します。ユニークな値の数が小さい場合には 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、かつ制約 (プライマリ・キーまたは一意性) が設定されていない場合は、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」にある CREATE INDEX 文の「注意」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、非ユニークな複合 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                                                                                                          |
|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 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。   | GROUP BY クエリでは、このインデックスは推奨されない。                                                                             |
| 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。 | 一意性を確保すると、インデックスが使用不可になる。                                                                                   |
|                                                          | 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。 |

## 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)
```

## 制限事項

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。
- NULL として定義されたカラムに対して CMP インデックスを作成することはできませんが、テーブルへの部分挿入を行うときに、その挿入データに CMP インデックスのすべてのカラムが含まれていない場合は、操作が失敗します。
- 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、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、ラージ・オブジェクト管理機能を使用する正規のライセンスを取得する必要があります。ラージ・オブジェクト管理オプションの詳細については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。

---

## 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 Add Index ウィザードでは、区切り文字を指定するときにこのことが指示されませんが、一重引用符を省略するとエラーになります。
- 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(c1)
DELIMITED BY 'f' LIMIT 40 LIMIT 99 DELIMITED BY 'g'
DELIMITED BY 'h';
```

文 2:

```
CREATE WD INDEX clwd on foo(c1)
DELIMITED BY 'h' LIMIT 99;
```

- WD インデックスの作成後は、そのカラムに対するすべての挿入が指定の区切り文字に基づいて解析されます。また、インデックス作成後に最大ワード・サイズを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ので、

```
'Farms'
```

1 ではなく 0 を返します。この問題を防ぐには、CHAR カラムではなく VARCHAR カラムを使用します。

- 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](#)」(274 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 関数である。さらに、比較のもう一方が定数式である (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 関数である。さらに、AND の両側が定数式である (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 関数である。さらに、IN リスト内の値が定数式である。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) である。

- 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ある (実開始日が年始で、実終了日が年末である)。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ある (実開始日が月始で、実終了日が月の月末である)。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日である。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 つ目の範囲は、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 つ目の範囲では、'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 データ型のカラムに対しては、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 句や ON 条件、GROUP BY 句で参照する場合には、それらのカラムに対して必ず DATE、TIME、または DTTM インデックスを作成することを推奨します。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。

## 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
- 次に例を示します。

```
SELECT n_name, sum(l_extendedprice*(1-l_discount))
  AS revenue
FROM customer, orders, lineitem, supplier,
     nation, region
WHERE c_custkey      = o_custkey
   AND o_orderkey    = l_orderkey
   AND l_suppkey      = s_suppkey
   AND c_nationkey    = s_nationkey
   AND s_nationkey    = n_nationkey
```

```

AND n_regionkey      = r_regionkey
AND r_name            = 'ASIA'
AND o_orderdate       >= '1994-01-01'
AND o_orderdate       < '1995-01-01'

GROUP BY n_name
HAVING n_name LIKE "I%"
      AND SUM(l_extendedprice*(1-l_discount)) > 0.50
ORDER BY 2 DESC

```

このクエリで参照されるカラムのうち、*l\_extendedprice* と *l\_discount* を除くすべてのカラムは、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<br>または<br>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<br>または<br>LF     |
| 含まれるユニークな値の数が多く、GROUP BY、SELECT DISTINCT、または DISTINCT COUNT では使用されないカラムを識別する。            | HNG                 |
| データ型、精度、位取りが同じで、頻繁に比較される可能性がある一対のカラムを識別する。                                               | CMP                 |

| 識別の条件                                                                                                                   | 選択するインデックス         |
|-------------------------------------------------------------------------------------------------------------------------|--------------------|
| 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                 |
| 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、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することができます。

**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、それらのテーブルにジョイン・インデックスを作成します。

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 人以上の顧客がありますが、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「多数」側を表すテーブルは、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

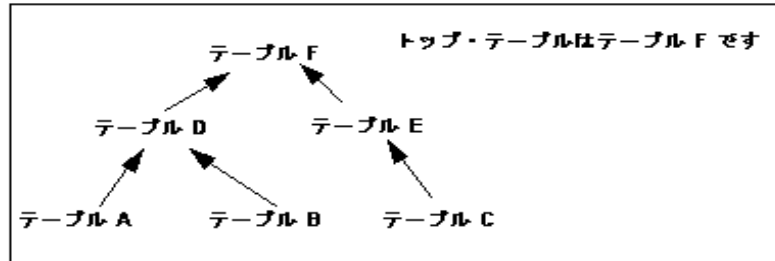
Sybase IQ でサポートされるジョイン関係にはすべて階層があります。ジョイン階層は、ジョイン内のすべてのテーブルのつながり方を示すツリーと考えてください。

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繋がりますが、このテーブルからは他のテーブルに繋がっていません。トップ・テーブルは常に、1 対多の関係の「多数」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・インデックス内のカラム

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 カ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 カ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 カラムがそれぞれ作成されます。この例では、カラムが 10 個 (5 つのテーブルそれぞれから 2 つずつ) と、ROWID カラムが 4 つになります。

LOAD TABLE または INSERT 文の NOTIFY オプションを使用すると、カラム・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カラム (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](#)」(632 ページ) を参照してください。

ジョイン・インデックスのカラムのデータ型、精度、位取りは同じであることが必要です。

### 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 \(279 ページ\)](#) 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にジョイン・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 つ目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」の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 が入っています。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、顧客が行う取り引きごとに 1 つのローが含まれます。それぞれの顧客は多くの取り引き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。

したがって、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      |

---

**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ければ、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することができます。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。

- あるテーブルを 1 対多の関係で「1」の側に定義できるのは、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」の側です。

- 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することができます。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・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、特定のカラムのペアに関連するジョインの条件を見つけます。

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、ジョインの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・インデックスの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す。
- 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入出力](#)」を参照してください。CREATE TABLE 文、ALTER TABLE 文、LOAD TABLE 文、INSERT 文、および SYNCHRONIZE JOIN INDEX 文の完全な構文については、『リファレンス：文とオプション』を参照してください。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、指定する回数は1回のときと2回のときがあります。

- 「**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間の関係を定義したり、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 GROUPO.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), )
```

あるいは、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て、後から外部キーを追加することもできます。次のいずれかまたはその他の文を実行できますが、2 つの文を実行することはできません。

```
ALTER TABLE GROUP0.MySalesOrders
ADD FOREIGN KEY ky_so_customer (CustomerID)
REFERENCES GROUP0.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**」(291 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内にある対等な値をジョインする (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 (279 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 つ目の方法と同様、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](#) の階層を完全に作成するにはキー・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]: 以下のジョイン関係から作成されたジョイン・インデックス 'join_on_tabletable':
[20694]:      テーブル名                関係
[20697]: -----
[20696]: 1. join_on_table_a が 'join_on_table_b' にジョインしました 1 つ >> 多数
[20692]: 基本ノットップ・テーブル: 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 Server — 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ないてはなりません。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「[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 \(279 ページ\)](#) 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) のいずれにも依存することができます。

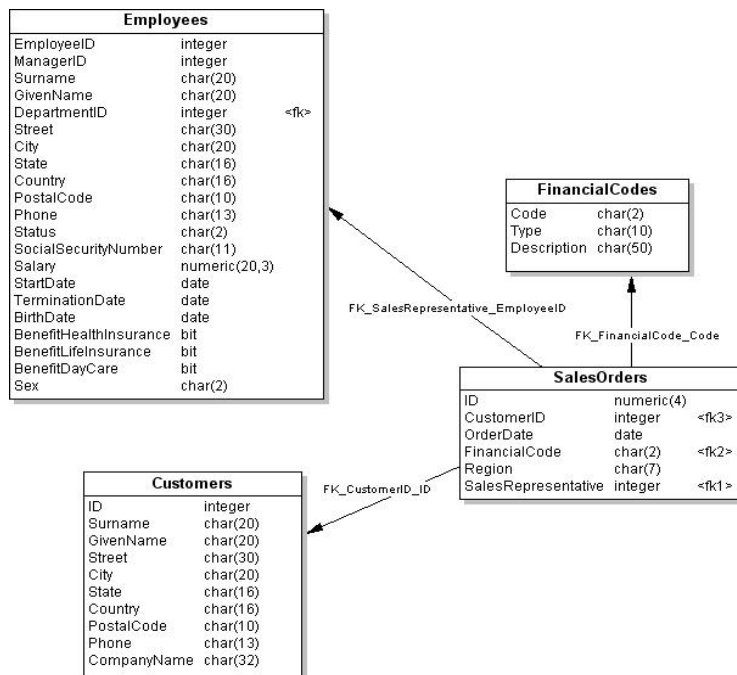
## スター・ジョイン

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、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 GROUPO.FinancialCodes (Code)
ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE GROUPO.SalesOrders
ADD FOREIGN KEY
FK_SalesRepresentative_EmployeeID
(SalesRepresentative)
REFERENCES GROUPO.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、次元の数が増えた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](#)」(229 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](#)」(433 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・インデックスを形成するテーブルへの部分幅挿入は実行できません。ジョイン・インデックス内のテーブルにカラムを追加する必要がある場合は、以下のうちのいずれかを行ってください。

- ジョイン・インデックスを削除し、部分幅挿入を行って、ジョイン・インデックスを作成し直す。
- テーブルのすべてのカラムにロードまたは挿入する。

## 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。しかし、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、基盤となる「ジョイン仮想テーブル」に対する権限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することができます。

## 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、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\_iquestjoin 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ド・プロシージャ `sp_iquestjoin` 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 つ目にジョインするテーブルの名前
- 2 つ目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。常にテーブルを 1 対 1 でジョインする場合は、“Min Case index\_size” を使用します。1 対多のジョインを時折行う可能性がある場合は、“Avg Case index\_size” を使用します。1 対多のジョインを数多く使用する可能性がある場合は、“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、別のストアド・プロシージャ `sp_iqjoinindexsize` を使用します。

すべてのストアド・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もあります。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。

# データベースへのデータの入出力

## 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                                                 | ページ |
|------------------------------------------------------|-----|
| <a href="#">インポートとエクスポートの概要</a>                      | 300 |
| <a href="#">データベースからのデータのエクスポート</a>                  | 303 |
| <a href="#">LOAD TABLE 文を使ったデータのバルク・ロード</a>          | 316 |
| <a href="#">INSERT 文の使用</a>                          | 330 |
| <a href="#">指定した値のロー単位の挿入</a>                        | 330 |
| <a href="#">データベースからの指定したローの挿入</a>                   | 331 |
| <a href="#">データの対話的インポート</a>                         | 335 |
| <a href="#">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</a>        | 336 |
| <a href="#">ジョイン・インデックスのテーブルへの挿入</a>                 | 338 |
| <a href="#">プライマリ・キー・カラムと外部キー・カラムへの挿入</a>            | 340 |
| <a href="#">部分幅挿入</a>                                | 340 |
| <a href="#">挿入時のデータ変換</a>                            | 346 |
| <a href="#">データ表示に関連する他の項目</a>                       | 362 |
| <a href="#">Adaptive Server Enterprise とのデータ型の対応</a> | 363 |
| <a href="#">データのバルク・ロードのチューニング</a>                   | 369 |
| <a href="#">UPDATE によるデータの更新</a>                     | 374 |
| <a href="#">データの削除</a>                               | 375 |

## 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、DBISQL 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 テーブルの列に挿入しても、実際はデータはテーブルの列ではなく、列のインデックスに追加されます。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](#)」(305 ページ)を参照してください。

DBISQL 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 を使用したリモート・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](#)」(323 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 に転送することができますが、SQL Anywhere から Sybase IQ へのローの転送は一度に 1 ローずつ行われます。Sybase IQ では、BLKLIB がサポートされていません。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。Sybase IQ と Adaptive Server Enterprise の BCP 形式は、日付に 1 桁の値が入っている場合に、空白をサポートします。

## データの入出力フォーマット

LOAD TABLE 文は、1 行に 1 つのローを持つテキスト・ファイル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                                                                             |

| ファイル・フォーマット | 説明                                                   |
|-------------|------------------------------------------------------|
| 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                                        |
| 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 でエンコードされた、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](#)」(305 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 DBISQL メニュー・バーから [コマンド] - [オプション] を選択します。
- 2 ドロップダウン・リストから [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。

## データ修正のパフォーマンス

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、パフォーマンスが低下しディスクの使用率が増加します。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](#)」(369 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 Server — 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、DBISQL 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。次に例を示します。

```
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・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    | "      |

| オプション名                     | 指定できる値      | デフォルト値 |
|----------------------------|-------------|--------|
| 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\_Size*n* 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 : 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` に有効なパス名を設定してください。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。パス名には、他で使用されていないファイル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。

`TEMP_EXTRACT_NAME1` は、`TEMP_EXTRACT_APPEND` オプションが ON に設定されている場合には、出力ファイル名の指定にも使用されます。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**) が持っていないわけではありません。追加モードの場合は、抽出されたロー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！** 既存のファイルへのパス名を指定し、`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、ファイルの中身が上書きされます。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  | サイズ                             |
|-----------|---------------------------------|
| ディスク・ファイル | AIX と HP-UX : 0 ～ 64GB          |
|           | Sun Solaris と Linux : 0 ～ 512GB |
|           | Windows : 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<sub>n</sub> 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<sub>n</sub> 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すると、「データ抽出機能を使用するためのパーミッションがありません。」というエラーが返されます。

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<sub>n</sub> 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、「ファイルが存在しません。ファイル：<invalid path>」というエラーが返されます。

TEMP\_EXTRACT\_DIRECTORY が空白の場合、一時抽出ファイルは、TEMP\_EXTRACT\_NAME<sub>n</sub> で指定されたディレクトリに配置されます。TEMP\_EXTRACT\_NAME<sub>n</sub> 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 抽出の場合は、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の文字列です。IQ は、このオプションのデフォルトの空の文字列を、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の文字列 " 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の文字列で、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 " (空の文字列)
- 日付には " (空の文字列)
- 時刻には " (空の文字列)
- タイムスタンプには " (空の文字列)

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 " (空の文字列) 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。

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。

イベント 'test\_ev' のハンドラで SQLSTATE '09W03' が発生しました。  
'test\_ev' では結果セットは許可されません。

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、オーバーヘッドはわずかしかなりません。

次に例を示します。

```
CREATE PROCEDURE proc1()
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 proc1();
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、パイプから読み取る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は、それに応じて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てはなりません。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。

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ため、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](#)」の手順に従います。

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](#)」(326 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、パス名はデータベース・サーバと相対的です。クライアント・アプリケーションとは相対的ではありません。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・アプリケーションと相対的である必要があります。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。パイプを記述するときには、FlushFileBuffers() と DisconnectNamedPipe() を呼び出します。プログラムによって呼び出されない場合は、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、『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「[Sybase IQ BINARY ロード形式](#)」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 オプション** 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      // 後続ブランクは削除される

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

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

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

バイナリ・データの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<br>オプション |       | SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1<br>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>        |

| LOAD TABLE<br>オプション |       | SELECT c1, c2, c3, LENGTH(c2), LENGTH(c3) FROM t1<br>の結果 |              |               |            |            |
|---------------------|-------|----------------------------------------------------------|--------------|---------------|------------|------------|
| QUOTES              | STRIP | c1                                                       | c2           | c3            | length(c2) | length(c3) |
| 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。

**BLOCK FACTOR オプションの例** 次の UNIX の例では、BLOCK FACTOR に 50,000 のレコードを指定して PREVIEW オプションを指定します。

```
LOAD TABLE lineitem
  (l_shipmode ASCII(15),
   l_quantity ASCII(8),
   FILLER(30))
FROM '/d1/MILL1/tt.t'
BLOCK FACTOR 50000 PREVIEW ON
```

**BLOCK SIZE オプションの例** 次の UNIX の例では、BLOCK SIZE に 200,000 バイトを指定します。

```
LOAD TABLE mm
  (l_orderkey '¥x09',
   l_quantity '¥x09',
   l_shipdate DATE('YYYY/MM/DD'))
FROM '/d1/MILL1/tt.t'
BLOCK SIZE 200000
```

**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'
BLOCK FACTOR 1000
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 9,000 のローを入力ファイルから読み込み、先頭の 5,000 個のローを省略し、次の 4,000 個のローをロードします。入力ファイル内に 8,000 個のローしかない場合は、3,000 個のローのみがロードされます。

```
LOAD TABLE lineitem(
    l_shipmode ASCII(15),
    l_quantity ASCII(8),
    FILLER(30))
FROM '/d1/MILL1/tt.t'
BLOCK FACTOR 1000
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。このオプションを使用して既存のローにデータを挿入する方法の例については、「[部分幅挿入](#)」(340 ページ)を参照してください。

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` ユー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 リリース 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変更する場合は、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ときに、パフォーマンスが最適化されます。

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、プライマリ・ファイル内のデータは、プリロード・プロセスを使用して並べ替える必要があります。

- 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。

「オプション START ROW ID は分割されたテーブルで使用できません。」

(SQLCODE -1009416L, SQLSTATE QCB14, Sybase エラー・コード 21054)

- 分割されたテーブルでは INSERT 文の START ROW ID 句がサポートされていないため、部分幅挿入はおすすめしません。

部分幅挿入を実行する必要がある場合は、ロード操作のカラム・リストにテーブルの分割カラムのパーティション・キー・カラムを含め、指定されていないすべてのカラムは NULL のままにしてください。カラム・リストでパーティション・キー・カラムが省略された場合は、次のエラーが報告されます。

「操作を行うことができません - パーティション・キー・カラム %2 が指定されていません。」

(SQLCODE -1009418L, SQLSTATE QCB16, Sybase エラー・コード 21056)

ここで、%2 はパーティション・キーの名前です。

- 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。分割されたテーブルの各パーティションに、ロー ID の範囲が割当てられます。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。分割されていないテーブルでは、LOAD 文または INSERT 文の 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により、ローの挿入場所を制御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、次のエラーが発生します。

「分割されたテーブル上のパーティション・キー・カラムを更新することはできません。」

(SQLCODE -1009417L、SQLSTATE QCB15、Sybase エラー・コード 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](#)」(632 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：

```
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。

- **ロー ID** 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：

```
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：

```
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：

```
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
```

## 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。START ROW ID オプションを指定すると、部分幅挿入を実行できます。このオプションを指定する前に、「[部分幅挿入](#)」を参照してください。部分幅挿入の詳細、および分割されたテーブルでの START ROW ID オプションの使用方法については、「[分割されたテーブルのロード](#)」(323 ページ)を参照してください。

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 へ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。

この機能を利用するには、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に、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、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、あるいは 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。

ラージ・オブジェクト管理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。ラージ・オブジェクト管理オプションの詳細については、『Sybase IQ によるラージ・オブジェクト管理』を参照してください。

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。

もっと大きなデータのロードが必要な場合には、「[LOAD TABLE 文を使ったデータのバルク・ロード](#)」(316 ページ)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 }
```

**注意** **START ROW ID** を使用して、ロード先テーブル内のカラムよりも少ないカラムを選択すると、NULL 値が許可されている場合は、ロード先テーブルの残りのローのカラムは NULL になります。詳細については、「[部分幅挿入](#)」(340 ページ)を参照してください。

12 より前のバージョンの Sybase IQ からデータのインポート

12.0 以前のバージョンの Sybase IQ データベースからデータをインポートするには、次のいずれかの方法を使用します。

- **UNLOAD FORMAT** オプションを指定した **LOAD TABLE** コマンド
- **INSERT...LOCATION** 構文

**INSERT** コマンドのその他の構文は使用できません。

古いバージョンからのロードの詳細については、『インストールおよび設定ガイド』を参照してください。

## データの対話的インポート

少量のデータを挿入する場合、DBISQL 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** DBISQL 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手順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 でアップグレードして 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。

以降では、この手順の詳細について説明します。

❖ 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。ジョイン・インデックスと他のインデックスには、2つの相違点があります。

- 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。

- ジョイン・インデックスに関連するテーブルに部分幅挿入を行うことはできない。

**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、**SYNCHRONIZE** コマンドを使用します。こ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』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・カラムへの挿入

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。

## 部分幅挿入

デフォルトでは、インデックスに領域がある場合、新しいローが挿入され、各 LOAD TABLE または INSERT 文によって新しいローが開始されます。この方法は、挿入するデータが新しいローである場合にかぎり有効です。Sybase IQ では、rowid を指定すると、既存のローへの個別のカラムの挿入もできます。

部分幅挿入は、垂直挿入とも呼ばれ、テーブルのカラムのサブセットへの挿入を意味します。2 つ以上の部分幅挿入を使用して、テーブルの全カラムにデータを挿入できます。

部分幅挿入では、次のことを行うことができます。

- 一度に数行ずつデータを挿入します。この方法はメモリに制約がある場合に役立ちます。

たとえば、各グループのインデックスに別の **LOAD TABLE** 文または **INSERT** 文を使用し、**START ROW ID** オプションを使用して **ROW ID** の一貫性を保ち、メモリ要件を低く抑えながら、数個のカラムに対して一度にデータを挿入できます。この方法は、大きなテーブルにデータを挿入するときにすべてのインデックスを一度に移植するのに十分なメモリがない場合に便利です。

- 複数のフラット・ファイルなどの異なるデータ・ソースを使用して、テーブルの異なるグループのカラムにデータを挿入します。
- テーブルのカラムにデータを挿入してから、新しいカラムと対応するインデックスをそのテーブルに追加します。詳細については、**ALTER INDEX** コマンドを参照してください。

---

**警告！** これは高度な操作です。部分幅挿入のすべての手順が正しく行われない場合は、データが正しく挿入されないことがあります。Sybase IQ に関して経験があり、データに精通している場合にのみ、このタイプの挿入を実行してください。テーブルのすべてのカラム・インデックスに同時にデータを挿入する全体幅挿入は、ロー・レベルの整合性を保証し、エラーを防止します。

---

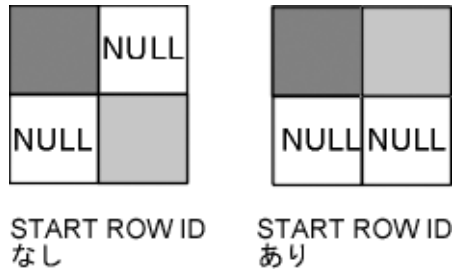
**START ROW ID** を使用して、挿入を開始するローを指定します。その後で、1 つの部分幅 **INSERT** 文または **LOAD TABLE** 文を使用して、ローのいくつかのカラムにデータを挿入し、追加の **INSERT** 文または **LOAD TABLE** 文を使用して、同じローの他のカラムにデータを挿入できます。

**LOAD TABLE** および **INSERT** コマンドの **START ROW ID** 句は分割されたテーブルでサポートされていないため、部分幅挿入はおすすめしません。

すでにデータが含まれているカラムに挿入しようすると、エラーになります。

各挿入を開始するローを制御する必要があります。**START ROW ID** を使用しない場合、次のローから挿入が開始され、現在のローの残りのカラムには **NULL** が挿入されます。図 7-1 はこれを表した図です。2 つの影付きパターンは、2 回の挿入操作によってカラムに挿入されるデータを示します。

図 7-1 : 部分幅挿入のための START ROW ID の使用



**注意** INSERT VALUES コマンド・フォーマットを使用して部分幅挿入を行わないでください。START ROW ID は、このフォーマットによって指定できないため、上の図に示す問題が発生します。

部分幅挿入の規則

初期の部分幅挿入に含まれず、データが入力されていないカラム・インデックスは、NULL を許容する必要があります。Sybase IQ は、このようなカラム・インデックスに NULL を挿入します。カラム・インデックスで NULL が許容されていない場合、挿入は失敗します。

部分幅挿入を行う場合、次の手順に従います。

- 1 ローの各セットに対する最初の部分挿入の場合、START ROW ID を指定しないでください。次に挿入可能なローは Sybase IQ が自動的に判断します。
- 2 同じセットのローに対する 2 番目以降の部分幅挿入の場合は、START ROW ID オプションを使用して挿入を開始するローを指定します。この値は、この例に示すように挿入メッセージ・ログの先頭のレコード番号です。

```
In table 'Departments', the full width insert of 3
columns
will begin at record 1.
```

ROWID 関数を使用して以下のクエリのようにロー ID の表示もできます。

```
SELECT *, ROWID(table_name) FROM table_name
```



## 例 1

次の UNIX の例は、`lineitem` テーブルのインデックスにファイル `tt.t` から 4 つのカラムが正しく挿入されない場合を示しています。最初に 2 つのカラムを 1 つの `LOAD TABLE` 文で挿入し、次にもう 2 つのカラムを別の `LOAD TABLE` 文で挿入していますが、追加するカラムを揃えるための `START ROW ID` オプションを使用していません。

```
LOAD TABLE lineitem
  (l_partkey ASCII(4),
   l_suppkey ASCII(4),
   FILLER(13))
FROM '/d1/MILL1/tt.t'
PREVIEW ON
NOTIFY 1000
```

```
LOAD TABLE lineitem
  (FILLER(8),
   l_quantity ASCII(6),
   l_orderkey ASCII(6),
   FILLER(1))
FROM '/d1/MILL1/tt.t'
PREVIEW ON
NOTIFY 1000
```

以下の `SELECT` 文の結果は、5 つのローではなく、10 のローが誤って格納されたことを示します。

```
SELECT *, rowid(lineitem) FROM lineitem
```

| <code>l_orderkey</code> | <code>l_partkey</code> | <code>l_suppkey</code> | <code>l_quantity</code> | <code>rowid(lineitem)</code> |
|-------------------------|------------------------|------------------------|-------------------------|------------------------------|
| -----                   | -----                  | -----                  | -----                   | -----                        |
| NULL                    | 1                      | 12                     | NULL                    | 1                            |
| NULL                    | 2                      | 37                     | NULL                    | 2                            |
| NULL                    | 3                      | 28                     | NULL                    | 3                            |
| NULL                    | 4                      | 13                     | NULL                    | 4                            |
| NULL                    | 5                      | 9                      | NULL                    | 5                            |
| 190                     | NULL                   | NULL                   | 19                      | 6                            |
| 215                     | NULL                   | NULL                   | 2127                    | 7                            |
| 29                      | NULL                   | NULL                   | 1376                    | 8                            |
| 200                     | NULL                   | NULL                   | 119                     | 9                            |
| 59                      | NULL                   | NULL                   | 4                       | 10                           |

(10 rows affected)

### 例 2

次の例は、この操作の正しい処理方法を示します。2 番目の **LOAD TABLE** 文で **START ROW ID** オプションを指定している点に注目してください。

```
LOAD TABLE lineitem
  (l_partkey ASCII(4),
   l_suppkey ASCII(4),
   FILLER(13))
FROM '/d1/MILL1/tt.t'
PREVIEW ON
NOTIFY 1000
```

| SELECT *, rowid(lineitem) FROM lineitem |           |           |            |                 |
|-----------------------------------------|-----------|-----------|------------|-----------------|
| l_orderkey                              | l_partkey | l_suppkey | l_quantity | rowid(lineitem) |
| NULL                                    | 1         | 12        | NULL       | 1               |
| NULL                                    | 2         | 37        | NULL       | 2               |
| NULL                                    | 3         | 28        | NULL       | 3               |
| NULL                                    | 4         | 13        | NULL       | 4               |
| NULL                                    | 5         | 9         | NULL       | 5               |

```
(5 rows affected)
```

```
LOAD TABLE lineitem
  (FILLER(8),
   l_quantity ASCII(6),
   l_orderkey ASCII(6),
   FILLER(1))
FROM '/d1/MILL1/tt.t'
PREVIEW ON
NOTIFY 1000
START ROW ID 1
```

| SELECT *, rowid(lineitem) FROM lineitem |           |           |            |                 |
|-----------------------------------------|-----------|-----------|------------|-----------------|
| l_orderkey                              | l_partkey | l_suppkey | l_quantity | rowid(lineitem) |
| 190                                     | 1         | 12        | 19         | 1               |
| 215                                     | 2         | 37        | 2127       | 2               |
| 29                                      | 3         | 28        | 1376       | 3               |
| 200                                     | 4         | 13        | 119        | 4               |
| 59                                      | 5         | 9         | 4          | 5               |

```
(5 rows affected)
```

最初の 2 つのカラムと同じローに 2 番目の 2 つのカラムのデータを挿入するために、2 つのカラムを追加する **INSERT** コマンドの **START ROW ID** オプションにロー番号を指定する必要があります。

## FILLER オプションの使用

FILLER オプションは、Sybase IQ がスキップする入力ファイルのカラムを指定します。2 番目の 2 つのカラムはスキップされるため、この LOAD TABLE 文ではこれらの 2 つのカラムに NULL が挿入されます。この文を実行するには、これらのカラムで NULL を許容する必要があります。

## 例 3

この Windows の例では、partsupp テーブルに ps\_partkey と ps\_availqty という 2 つのカラムがあり、partsupp はジョイン・インデックスの一部でないと仮定します。

ps\_value のデータは ps\_availqty で計算されるため、ps\_availqty カラムにすでにデータが入力されている必要があります。このため、partsupp テーブルにデータを挿入するには、ps\_availqty と ps\_partkey に挿入してから、ps\_value に挿入します。

はじめに、ASCII ファイル tt.t から直接 partsupp のデータを挿入します。

```
LOAD TABLE partsupp
    (ps_partkey ASCII(6),
     ps_availqty ASCII(6),
     FILLER(2))
FROM 'C:¥¥iq¥¥archive¥¥mill1.txt'
```

```
SELECT *, rowid(partsupp) FROM partsupp
```

| ps_partkey | ps_suppkey | ps_availqty | ps_value | rowid(partsupp) |
|------------|------------|-------------|----------|-----------------|
| 213        | NULL       | 190         | NULL     | 1               |
| 24         | NULL       | 215         | NULL     | 2               |

(2 rows affected)

次に ps\_availqty を選択し、その 80% の値を計算してテーブルに挿入します。この場合、INSERT コマンドを使用して SELECT 文の結果を挿入してください。

```
INSERT INTO partsupp(ps_value)
START ROW ID 1
SELECT ps_availqty * 0.80 FROM partsupp
```

```
SELECT *, rowid(partsupp) FROM partsupp
```

| ps_partkey | ps_suppkey | ps_availqty | ps_value | rowid(partsupp) |
|------------|------------|-------------|----------|-----------------|
| 213        | NULL       | 190         | 152.00   | 1               |
| 24         | NULL       | 215         | 172.00   | 2               |

(2 rows affected)

他のファイルから `ps_partkey` と `ps_availqty` に後でデータをロードする場合、以下のように次のローから正しく挿入が開始されます。

```
LOAD TABLE partsupp
    (ps_partkey ASCII(6),
     ps_availqty ASCII(6),
     FILLER(2))
FROM 'C:¥¥iq¥¥archive¥¥mill2.txt'
```

```
SELECT *, rowid(partsupp) FROM partsupp
```

| ps_partkey | ps_suppkey | ps_availqty | ps_value | rowid(partsupp) |
|------------|------------|-------------|----------|-----------------|
| 213        | NULL       | 190         | 152.00   | 1               |
| 24         | NULL       | 215         | 172.00   | 2               |
| 28         | NULL       | 490         | NULL     | 3               |
| 211        | NULL       | 15          | NULL     | 4               |

(4 rows affected)

`ps_value` の値を計算し、挿入するには、この例に示す `INSERT` 文を繰り返し、`START ROW ID` 値を新しいロー番号 3 に変更します。

部分幅挿入のプレビュー

部分幅挿入を正しく行わない場合にはエラーが発生する可能性があるため、部分幅挿入をプレビューします。`PREVIEW` ロード・オプションを使用すると、ロード先テーブルの入力のレイアウトが表示されます。このオプションは、`LOAD TABLE` で使用可能ですが、`INSERT` コマンドでは使用できません。

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](#)」(349 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 とのデータ型の対応](#)」(363 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 以下の場合は **REAL** が使用され、*precision* が 15 よりも大きい場合は **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「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                                                                                                                                 | 動作                                                                                                                                                          |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 表記ができますが、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。                                                           |

| オプション    | Sybase IQ のデータ型                   | 動作                                                                    |
|----------|-----------------------------------|-----------------------------------------------------------------------|
| 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・ファイルからデータをロードするときに、バイナリ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 : IQ conversions for comparison operations

|                    | 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<br>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<br>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  |
| 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 : IQ conversions for arithmetic operations

|                 | 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  |
| 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 : IQ conversions for INSERT and UPDATE

|                    | 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<br>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<br>bigint | I   | I  | I  | I  | I  | I  | I  | I  | I  | I  | E  | E  | E  | E  | E  | I  | I  |

|            | To: |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|------------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| From:      | ti  | si | in | ui | bi | ub | nu | rl | dl | bt | dt | tm | ts | ch | vc | bn | vb |
| 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) 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・プレフィックス・バイトで定義される可変幅入力カラムを含むことができます。

## 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 `varitable` 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 varitable 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 shipinfo.t         |        |    | テーブル lineitem           |         |    |
|-------------------------|--------|----|-------------------------|---------|----|
| 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<br>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<br>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場合 (たとえば、“DD”、“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従って 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  | 2000-2049 |
|                    | 50-99  | 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す。ただし、DATETIME データは内部では同等なフォーマット 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 '/d1/MILL1/tt.t'
BLOCK FACTOR 1000
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'
BLOCK FACTOR 1000
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'
BLOCK FACTOR 1000
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
- 分は 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は 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<br>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 時の場合、'00' も使用できる。 |
| 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[.ssssss]        | 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<br>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 ('MMDDYYhhnnss')          |
| 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:ssssss') |
| 12/31/00 14:01:123456 | DATETIME ('MM/DD/YY hh:mm:ssssss') |
| 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、カッコと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 桁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:¥¥MILL1¥¥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](#)」に定義する規則に従って位取りが決定された後に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 とのデータ型の対応

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 日付／時刻型は、いかなる Adaptive Server Enterprise 日付／時刻型でも受け入れます。

## 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 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 によるラージ・オブジェクト管理』を参照してください。
- Sybase IQ 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、ASE 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                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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</i> , <i>scale</i> ) 形式のデータは NUMERIC( <i>precision</i> , <i>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。<br><br>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。                                                                                                                                                                                                                                                                                                                                                          |

| Adaptive Server Enterprise のデータ型 | Sybase IQ のデータ型            | 注意                                                                                                                                                      |
|----------------------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 SMALLINT では精度と位取りは使用できません。Adaptive Server Enterprise smallint(precision,scale) は NUMERIC(precision,scale) に変換されます。上記の INT を参照してください。          |
| tinyint                          | TINYINT                    | Sybase IQ TINYINT カラムでは精度と位取りは使用できません。Adaptive Server Enterprise tinyint(precision,scale) は NUMERIC(precision,scale) に変換されます。詳細については、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、ASE 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                                                                           |
|----------------------------------|-------------------|------------------------------------------------------------------------------|
| 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                                                                                                                                                                                                                                                                                                                        |
|----------------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 の方は 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。NULL を許容する Adaptive Server Enterprise の 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 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、ASE 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、ASE 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、ASE 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、ASE 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、ASE 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                                     |
|----------------------------------|-----------------|----------------------------------------|
| 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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| datetime                         | TIMESTAMP または 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 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。<br><br>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 カラムと同じ名前で作成します。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、ASE 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。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Adaptive Server Enterprise のデータ型 | Sybase IQ のデータ型 | 注意                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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。 <b>time</b> 値には 8 バイトの格納領域が必要です。<br>Adaptive Server Enterprise の <b>time</b> 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。 <b>time</b> 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。 <b>time</b> 値には 4 バイトの格納領域が必要です。<br>INSERT...LOCATION を使用して、ASE データベースの <b>time</b> 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                                                                                                                                                                                                                                                                                                    |
|----------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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。<br>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。<br>HNG インデックスでは、VARBINARY データは使用できません。<br>VARBINARY カラムから選択されたデータの挿入に INSERT ... LOCATION を使用する場合は、ローカル・データベースでは、挿入で使用するメモリを制限するよう、LOAD_MEMORY_MB オプションを設定し、またリモート・データベースでは、ASE_BINARY_DISPLAY を 'OFF' に設定します。 |

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・インポート時の変換エラーの処理

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](#)」(221 ページ) および「[IQ ページ・サイズの選択](#)」(198 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](#)」(607 ページ) および「[ディスク領域の使用状況のモニタリング](#)」(612 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に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**」に説明するように **LOAD TABLE** コマンド・オプションを正しく設定します。特に、十分なメモリがある場合や、ロード時に他のユーザがアクセスしていない場合には、**BLOCK FACTOR** の値を増やします。

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 **サーバのコマンド・ライン・スイッチ**」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。

- **SET OPTION** コマンドを使用して **LOAD\_MEMORY\_MB** オプションを変更することによって、ロード処理に使用されるヒープ・メモリ量を調整します。**LOAD\_MEMORY\_MB** をデフォルト (0) に設定すると、Sybase IQ は十分なヒープ・メモリを使って、パフォーマンスを最適にします。システムで仮想メモリが不足する場合、2000 未満の値を指定して、ロードができるようになるまで値を小さくします。大きなテーブルに挿入する場合、**LOAD\_MEMORY\_MB** を小さい値 (100 ~ 200 MB) に設定することが必要な場合があります。ただし、値が小さすぎると、物理的にデータをロードできなくなることもあります。なお、このオプションは **INSERT**、**UPDATE**、**SYNCHRONIZE**、**DELETE** の処理にも影響することに注意してください。
- 幅が非常に大きい **varchar** データをロードする場合は、**LOAD\_MEMORY\_MB** データベース・オプションと、**LOAD TABLE** コマンドの **BLOCK FACTOR** オプションの値を小さくします。パフォーマンス・チューニングでは常に言えるように、これらの値の調整には試行錯誤が必要な場合があります。
- 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](#)」(464 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、もし可能であれば、サーバと同じマシン上で **DBISQL** 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 PACKETSIZE 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・ロード時のメイン IQ 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 DB 領域内でページが一時的にバージョン管理されます。同じテーブルから削除した後にインクリメンタル・ロードを行う場合に、このバージョン管理がよく行われます。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 (メモリカ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。

テーブルへの挿入は単一のスレッド構成になります。

ロー・デリミタのないデータの可変幅ロード、および部分幅の可変幅ロードについては、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。

挿入／ロードの部分は単一のスレッドになります。

## 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 (351 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
```

サンプル・データベースの Customers テーブル内の 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することができます。

```
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%'
```

検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、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                                            | ページ                 |
|-------------------------------------------------|---------------------|
| <a href="#">データベース・パーミッションの概要</a>               | <a href="#">377</a> |
| <a href="#">ログイン管理</a>                          | <a href="#">383</a> |
| <a href="#">IQ のユーザ・アカウントと接続の管理</a>             | <a href="#">384</a> |
| <a href="#">ユーティリティ・データベース・サーバのセキュリティ</a>       | <a href="#">387</a> |
| <a href="#">個別のユーザ ID とパーミッションの管理</a>           | <a href="#">390</a> |
| <a href="#">グループの管理</a>                         | <a href="#">398</a> |
| <a href="#">データベース・オブジェクトの名前とプレフィクス</a>         | <a href="#">404</a> |
| <a href="#">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</a> | <a href="#">406</a> |
| <a href="#">ユーザ・パーミッションの評価方法</a>                | <a href="#">409</a> |
| <a href="#">リソース接続使用の管理</a>                     | <a href="#">410</a> |
| <a href="#">システム・テーブルのユーザとパーミッション</a>           | <a href="#">412</a> |
| <a href="#">トランスポート・レイヤ・セキュリティ</a>              | <a href="#">413</a> |

## データベース・パーミッションの概要

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。

### 個別のユーザ ID の設定

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 は同意語として扱います。ただし、パーミッションの付与と取り消しは、常に各ユーザ ID に対して行います。

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があります。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。

## DBA 権限の概要

データベースを作成すると、使用可能なユーザ ID が 1 つ作成されます。この最初のユーザ ID は DBA で、パスワードは sql と設定されます。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、データベースに関係する作業をすべて行うことができます。これにはテーブルの作成、テーブル構造の変更、新規ユーザ ID の作成、ユーザからのパーミッションの取り消しなどが含まれます。

---

**注意** データベースのセキュリティを確立するために、DBA のパスワードをデフォルトの sql から別のもの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または単に DBA として頻繁に表現されます。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 権限を他のユーザ ID に付与することはできますが、この章では、DBA ユーザ ID はデータベース管理者のことを指します。DBA はユーザ ID の DBA と、DBA 権限を付与されたユーザ ID 両方の意味に使われます。

**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ありますし、データベースに情報を追加するユーザもあります。また、データベースの構造そのものを変更するユーザもあります。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 とパスワードはデータベースのオブジェクトです。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「[ユーザ ID とパスワード](#)」を参照してください。

## 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。DBA だけが RESOURCE 権限を他のユーザに付与できます。

## 所有者パーミッションの概要

### 所有者

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。

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、DBA である場合か、サーバの起動時にコマンド・ラインまたは設定ファイルで `-gl all` スイッチが設定されていた場合に限られます。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することができます。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」(402 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る                                                    |
| DELETE     | テーブルまたはビューからローを削除できる                                             |
| INSERT     | テーブルまたはビューにローを追加できる                                              |
| 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る                            |
| SELECT     | テーブルまたはビューの情報を参照できる                                              |
| UPDATE     | テーブルまたはビューのローを更新できる。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、社内の部署ごとに **sales**、**marketing** などのグループを社内データベースに作成し、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へ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 8-1 にオプションのリストを示します。

表 8-1：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                              |
|-----------|----------------------------------------------------------------------|-------------------------|------------------------------------|
| -gd level | 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                                |
| -gk level | 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                                |
| -gl level | 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            | start_iq で起動されたサーバは ALL、他のサーバは DBA |
| -gu level | 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と、パーミッション・レベルの値および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](#)」と、『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 1-7 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るには、DBA 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](#)」、「[CREATE LOGIN POLICY 文](#)」、および「[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。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。ログイン・ポリシーの管理の詳細については、『SQL Anywhere Server — SQL リファレンス』の「[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マルチプレックス・サーバで実行でき、これらのコマンドはマルチプレックスのすべてのサーバに自動的に伝達されます。DDL でパフォーマンス上の理由で推奨されているとおり、これらのコマンドはコーディネータで実行する必要があります。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。

データベースを Sybase IQ 15.1 にマイグレートすると、既存のログイン管理設定が削除され、所定のストアド・プロシージャおよびシステム・テーブルは「Sybase IQ 15.0 の新機能」にリストされた新しいものに置換されます。マイグレーション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。

## IQ のユーザ・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 がユーザの追加または削除と接続の制御を行うには、次の方法があります。

- 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`」、および「`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 8-2 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。すべてのプロシージャを実行するには DBA 権限が必要です。ただし、sp\_iqpassword は除きます。すべてのユーザが sp\_iqpassword を実行して各自のパスワードを変更できます。

**表 8-2 : ログイン管理用のストアド・プロシージャ**

| ストアド・プロシージャ名          | 行うタスク                                                    |
|-----------------------|----------------------------------------------------------|
| sa_get_user_status    | 既存のユーザすべての現在のステータスを取得する。                                 |
| sp_expireallpasswords | すべてのユーザ・パスワードをすぐに期限切れにする。                                |
| sp_iqaddlogin         | 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。 |
| sp_iqcopyloginpolicy  | 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。                    |
| sp_iqdroplogin        | 指定されたユーザを削除する。                                           |
| sp_iqmodify           | 指定されたユーザにログイン・ポリシーを割り当てる。                                |
| sp_iqmodifyadmin      | 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。                           |
| sp_iqpassword         | ユーザのパスワードを変更する。                                          |

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するシステム・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 9 章 システム・テーブル](#)」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 サーバで稼働できます。

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”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）。次に例を示します。

```
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 ロー・パーティションを指定するには、ユーティリティ・データベースで“`\\Y.YI:`”と指定します。一方、他の IQ データベースでは、円記号を二重に指定する必要があります。つまり、上記と同じデバイスなら“`\\\\Y.YYI:`”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。

設定作業として行うべきタスクの 1 つに、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` という 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 8-3：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                       |
|------------|------------------------------|----------------------------|
| all        | 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 |
| none       | 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 |
| DBA        | 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 |

| -gu スイッチの値 | 結果                                      | 適用対象             |
|------------|-----------------------------------------|------------------|
| utility_db | 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と、DBISQL と Sybase Central で新しいユーザを作成する方法を説明します。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 ストアド・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、「[IQ のユーザ・アカウントと接続の管理](#)」(384 ページ) で説明したシステム・プロシージャを使用します。

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。

### ASE ストアド・プロシージャを使用したユーザ管理

この章では、DBISQL 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

新規ユーザは、DBA が GRANT CONNECT 文を使ってデータベースに追加します。次に例を示します。

### ❖ 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 DBISQL から、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 次の SQL 文を発行します。

```
CREATE USER M_Haneef
IDENTIFIED BY welcome
```

データベースに新規ユーザを追加できるのは DBA だけです。

#### 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](#)」(403 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 & グループ] フォルダを開きます。
- 3 [新しいユーザ] アイコンをダブルクリックするか、[ファイル]-[新しいユーザ] 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。

こうした操作を実施しないようにするには、システム・プロシージャ `sp_addlogin` と `sp_adduser` を使用してユーザを追加します。これらのプロシージャは、Adaptive Server Enterprise やバージョン 12 より前の Sybase IQ と同じく、既存のユーザ ID を追加しようとする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これらのルールに適合しているかどうかを確認することができます。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には、『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 権限が必要です。

---

**警告！** マルチブレックス・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、データベースを使用できなくなります。

---

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することができます。

## DBA および RESOURCE 権限の付与

DBA と RESOURCE 権限はまったく同じ方法で付与できます。

## ❖ ユーザ ID に RESOURCE 権限を付与するには

- 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 RESOURCE TO userid
```

DBA 権限の場合は、次の SQL 文を使います。

```
GRANT DBA TO userid
```

## 注意

- DBA または RESOURCE 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 つを与えてください。これにより、ユーザは必要ときにだけ 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す。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することができます。

- **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、変更する権限は与えません。
- **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、**M\_Haneef** に **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与えるには、次の手順に従います。

- 1 **DBA** 権限を持ったユーザとして、または **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
```

**M\_Haneef** に **sample\_table** というテーブル上で **column\_1** と **column\_2** のカラムだけを更新するパーミッションを付与するには、次の手順に従います。

- 1 **DBA** 権限を持ったユーザとして、または **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 [プロパティ] を選択します。
- 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。
  - [付与] 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、[取り消し] 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。

[パーミッション] タブのカラムの凡例は次のとおりです。

- A=Alter
- D=Delete
- I=Insert
- R=Reference
- S=Select
- U=Update

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」(393 ページ)で説明したテーブルとビューに対するパーミッションには、WITH GRANT OPTION を付けることができます。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」(401 ページ)を参照してください。

例 M\_Haneef に 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与え、このパーミッションを他のユーザに付与する、という権利を付与するには、次の手順に従います。

- 1 DBA 権限を持ったユーザとして、または 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) によって付与されます。

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」(393 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 M\_Haneef に my\_procedure というプロシージャを実行するパーミッションを与えるには、次の手順に従います。

- 1 データベースに DBA 権限を持ったユーザとして、または my\_procedure の所有者として接続します。
- 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 から my\_procedure を実行するパーミッションを取り消すには、次のコマンドを使います。

```
REVOKE EXECUTE ON my_procedure FROM M_Haneef
```

このコマンドを発行するのは、DBA かプロシージャの所有者である必要があります。

`sample_table` からローを削除するパーミッションは、次のコマンドを発行することによって取り消されます。

```
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 15.1 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'
```

- 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、データベースを使用できなくなります。

---

## グループの管理

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。



## DBA、RESOURCE、および GROUP パーミッション

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。これらは個別ユーザ ID ごと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](#)」(390 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](#)」(402 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 権限のあるユーザとしてデータベースに接続します。
- 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。

```
GRANT CONNECT
TO 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 そのデータベースの [ユーザ & グループ] フォルダをクリックします。
- 3 [グループの追加] をダブルクリックします。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 またはグループのユーザ ID によって付与できます。

例として、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 権限を持ったユーザまたはグループ・ユーザ ID の personnel としてデータベースに接続します。
- 2 次の SQL 文を使って、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がいくつかあります。

**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、ユーザ ID **personnel** が RESOURCE パーミッションを付与されていても、**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** がテーブルを作成すれば、**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利だけでなく、テーブルに関する権利を他のユーザに与える権利を持ちます。**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** に明示的に付与すれば、すべてのグループ・メンバはそのテーブルに対して SELECT 権限を持ちます。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ないユーザは Personnel.Employees と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](#)」(404 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。このユーザ ID には、グループのメンバシップを付与したり取り消したりできます。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。

グループとグループのデータベース・オブジェクトのすべての処理を DBA が行い、他のユーザ ID にはグループ・メンバシップの変更を許可しない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。こうするには、次に示すように GRANT CONNECT 文をパスワードなしで入力します。次のようになります。

```
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はいませんが、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 9 章 システム・テーブル](#)」および「[第 8 章 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。

このマニュアルでは、クエリとサンプル SQL 文で使用するサンプル・データベースのオブジェクトには、簡略名を使用します。次に例を示します。

```
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。サンプル・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ーである場合

**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、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。テーブルの所有者は **company** で、Sally と Joe は **company** グループのメンバではないからです。

```
SELECT *
FROM company.customers
```

この状況を整えるために、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、Joe と Sally は、Sales グループのメンバであり、間接的に **company** グループのメンバともなり、テーブルを修飾子なしで参照できます。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

Joe と Sally は **company** グループのメンバであり、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。次に例を示します。

- **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。



## セキュリティを調整するためにビューを使用する

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分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
```

(DBA は SELECT や FROM と同様に 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、テーブルは所有権を明確にして **"DBA".Employees** と指定する必要があります。そうしないと、**SalesManager** がビューを使うとき、SELECT 文はユーザ ID が認識しないテーブルを参照してしまいます。

- 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
```

テーブルに対するのとまったく同じコマンドを使ってビューの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す。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。

ビューでデータを修正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](#)」([396 ページ](#)) で説明したように、ユーザはプロシージャの対象になるテーブルのパーミッションなしに、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。

**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。

**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す。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` は特定のテーブルに対して個人的に `SELECT` パーミッションと `UPDATE` パーミッションを付与されていますが、このテーブルに対してアクセス権のないグループのメンバでもあり、`SELECT` アクセス権だけを保有しているグループのメンバでもあります。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、次の方法で判断します。

- 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。
- 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 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を提供します。これらのオプションは「リソース・ガバナンス」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。
- **LOAD\_MEMORY\_MB** 連続したロード処理で利用できるヒープ・メモリ割り付け量の上限を設定します。

- **MAX\_CARTESIAN\_RESULT** 直積ジョインを含むクエリから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 DB 領域の量を制限します。
- **QUERY\_ROWS\_RETURNED\_LIMIT** クエリ・オブティマイザに対し、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。オブ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、オブ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 への影響は制限されます。

- **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、「[IQ のユーザ・アカウントと接続の管理](#)」(384 ページ)を参照してください。

メモリに影響する設定や、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 9 章 システム・テーブル](#)」を参照してください。

大半のシステム・テーブルの所有者は、特別なユーザ ID **SYS** です。**SYS** ユーザ ID に接続することはできません。

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 のパスワードを持つ **SYS.SYSUSERPERM** テーブルには、DBA だけがアクセスできます。しかし、パスワードだけを除くすべての情報があるビューの **SYS.SYSUSERPERMS** 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・メンバー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 ロウ                         |
| 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 |

| テーブル        | デフォルト  | 内容                                         |
|-------------|--------|--------------------------------------------|
| SYSPROCPERM | PUBLIC | 各ローには、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         |
| 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、Sybase Product Manuals Web サイトの『SQL Anywhere Server — データベース管理ガイド』で説明されているように SQL Anywhere データベースの暗号化に似ています。また、Sybase IQ を使用すると、『Sybase IQ による高度なセキュリティ』で説明されているように、カラムを暗号化できます。

FIPS 暗号化、Kerberos 認証、およびカラムの暗号化のサポートは個別にライセンス供与された Sybase IQ Advanced Security オプションに含まれています。これについては『Sybase IQ による高度なセキュリティ』で説明しています。

## 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 Server — データベース管理ガイド』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                               | ページ |
|------------------------------------|-----|
| <a href="#">データ整合性の概要</a>          | 415 |
| <a href="#">カラムのデフォルトの使用</a>       | 420 |
| <a href="#">テーブル制約とカラム制約の使用</a>    | 429 |
| <a href="#">エンティティ整合性と参照整合性の宣言</a> | 433 |
| <a href="#">システム・テーブルの整合性ルール</a>   | 443 |

## 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 オンライン・ヘルプ](#) (<http://infocenter.sybase.com/help/index.jsp>) の SQL Anywhere Studio のマニュアルにある『SQL Anywhere Server — プログラミング』の「第 5 章 JDBC Programming」を参照してください。

## 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。

- |                  |                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 整合性のないスキーマ       | <ul style="list-style-type: none"><li>• <code>customers</code> テーブルに存在しない <code>customer_id</code> に対応する注文を <code>orders</code> テーブルに入力した場合</li></ul>              |
| 不正なフォーマットの<br>情報 | <ul style="list-style-type: none"><li>• 数値データを入力すべき場所にテキストを入力した場合</li><li>• カラム長より長い数値データを入力した場合</li></ul>                                                         |
| 重複データ            | <ul style="list-style-type: none"><li>• <code>dept_id</code> 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 <code>department</code> 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</li></ul> |

## 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 | <p>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。</p> |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|

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込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](#)」(221 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](#)」(277 ページ)を参照してください。

#### ロード中の制約の検査

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。
- 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](#)」(429 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。

### エンティティおよび参照整合性

リレーショナル・データベース・テーブル内のデータは、テーブル間の関係によって互いに関連付けられています。この関係は、データベース設計に組み込まれた候補キーと外部キーによって定義されます。

外部キーは、1 つのカラムまたは複数のカラムの組み合わせによって構成されます。各外部キーは、一方のテーブル (「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れません。
- 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、特定のテーブルの各ローが固有に識別されることが保証されます。
- **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](#)」(433 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することができます。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 **DEFAULT** 値を指定できません。サポートされていない **DEFAULT** 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 **DEFAULT** 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 \(351 ページ\)](#) を参照してください。
- **DEFAULT** 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 **DEFAULT** 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 **DEFAULT** 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 **DEFAULT** 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 **DEFAULT** 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 **DEFAULT** 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 **DEFAULT** 値が許可されません。そうしたテーブルに **DEFAULT** 値を定義しようとする、**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利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](#)」([424 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 DEFAULT 値をサポートしません。また IQ も、データベース・オプション DEFAULT\_TIMESTAMP\_INCREMENT をサポートしません。UTC TIMESTAMP または CURRENT UTC TIMESTAMP タイプのカラムで DEFAULT 値の挿入または更新が試みられるたびに、Sybase IQ はエラーを生成します。

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することができます。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が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。ここで使用した `YYYY/MM/DD` のフォーマットには、現在のオプション設定にかかわらず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。制約を削除しない場合は、テーブルのデータに対するクエリは「カラムが見つかりません」というエラー・メッセージを返します。

## 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、明示的に **CREATE TABLE** コマンドや **ALTER TABLE** コマンドを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](#)」(229 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 サンプル・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。サンプル・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 サンプル・データベースの **sales\_order\_items** テーブルでは、2 つのカラムを使って 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](#)」(229 ページ)を参照してください。

例 サンプル・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考えてください。

```
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が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](#)」(443 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。次に例を示します。

```
ALTER TABLE DEPT DROP FOREIGN KEY MGR_EMPNO;
ALTER TABLE EMPLOYEE DROP FOREIGN KEY DEPT;
```

外述のスキーマでは、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。次に例を示します。

```
ALTER TABLE DEPT ADD FOREIGN KEY MGR_EMPNO (Mgr)
REFERENCES EMPLOYEE (EmpNo);
ALTER TABLE EMPLOYEE ADD FOREIGN KEY
EMP_DEPT (DeptNo) REFERENCES DEPT (DeptNo);
```

#### 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。たとえば次のとおりです。

```
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。次に例を示します。

```
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・キーか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、どの処理が認められ、どの処理がエラーを返すかを示します。表 9-2 は、強制力のある外部キーと候補キーに対して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/<br>INSERT<br>ftab1 | DELETE/<br>TRUNCATE<br>TABLE<br>ftab1 | UPDATE<br>ftab1<br>(fk1,fk2) | 新しいイン<br>デックス非 FK<br>ftab1 の挿入<br>(fk1,fk2) | ADD FK<br>ftab1<br>(fk1 fk2) | DROP<br>FK ftab1<br>(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<br>TRUNCATE TABLE ftab2 | 可能                       | 可能                                    | 可能                           | 可能                                          | 可能                           | 可能                             |
| DELETE ptab<br>TRUNCATE TABLE ptab   | エラー                      | 可能                                    | エラー                          | 可能                                          | エラー                          | エラー                            |

|                                                                    | LOAD/<br>INSERT<br>ftab1 | DELETE/<br>TRUNCATE<br>TABLE<br>ftab1 | UPDATE<br>ftab1<br>(fk1,fk2) | 新しいイン<br>デックス非FK<br>ftab1 の挿入<br>(fk1,fk2) | ADD FK<br>ftab1<br>(fk1 fk2)                      | DROP<br>FK ftab1<br>(fk2, fk2)                    |
|--------------------------------------------------------------------|--------------------------|---------------------------------------|------------------------------|--------------------------------------------|---------------------------------------------------|---------------------------------------------------|
| UPDATE ftab2(fk1,fk2)                                              | 可能                       | 可能                                    | 可能                           | 可能                                         | 可能                                                | 可能                                                |
| UPDATE ptab (pk1,pk2)                                              | エラー                      | 可能                                    | エラー                          | 可能                                         | エラー                                               | エラー                                               |
| 新しいインデックスの<br>挿入                                                   | 可能                       | 可能                                    | 可能                           | 可能                                         | 可能                                                | 可能                                                |
| QUERY ((fk1,fk2) の有無<br>にかかわらず使用され<br>ている、古いバージョン<br>の ftab1/ptab) | 可能                       | 可能                                    | 可能                           | 可能                                         | エラー                                               | エラー                                               |
| 古いバージョンの ftab2<br>は使用されていない                                        | なし                       | なし                                    | なし                           | なし                                         | 可能 (使用<br>されていない古<br>いバージョンの<br>ftab1 はすべて<br>削除) | 可能 (使用<br>されていない古<br>いバージョンの<br>ftab1 はすべて<br>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                                                                                                                                                                                                                                                                                 |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>SYS.SYSTABLE</code>       | 検査制約は、 <code>SYS.SYSTABLE</code> の <code>view_def</code> カラムに入っている。 <code>view_def</code> には、ビューを作成する <code>CREATE VIEW</code> コマンドが入っている。あるテーブルについてそれがベース・テーブルなのかビューなのかを確認するとき、 <code>table_type</code> カラムに <code>BASE</code> が記述されている場合はベース・テーブル、 <code>VIEW</code> の場合はビューである。 |
| <code>SYS.SYSFOREIGNKEYS</code> | このビューは、 <code>SYS.SYSFOREIGNKEY</code> テーブルと <code>SYS.SYSFKCOL</code> テーブルからの外部キー情報を、より見やすい形で表示する。                                                                                                                                                                                |
| <code>SYS.SYSCOLUMNS</code>     | このビューは、 <code>SYS.SYSCOLUMN</code> テーブルからの情報を、より見やすい形で表示する。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9章 システム・テーブル](#)」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                                             | ページ                 |
|--------------------------------------------------|---------------------|
| <a href="#">トランザクションとバージョン管理の概要</a>              | <a href="#">445</a> |
| <a href="#">バージョン管理による矛盾の回避</a>                  | <a href="#">458</a> |
| <a href="#">ロックの仕組み</a>                          | <a href="#">459</a> |
| <a href="#">ロックの管理</a>                           | <a href="#">463</a> |
| <a href="#">独立性レベル</a>                           | <a href="#">465</a> |
| <a href="#">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</a> | <a href="#">466</a> |
| <a href="#">パフォーマンスの影響</a>                       | <a href="#">474</a> |
| <a href="#">トランザクション内のカーソル</a>                   | <a href="#">476</a> |

## 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、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 Server — 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 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、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取り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。

最終的には、IQ はダーティ・ページ、つまり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](#)」(468 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えます。

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取りを行います。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。読み取り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。その代わり、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、複数の読み取りユーザがクエリを同時に発行できます。

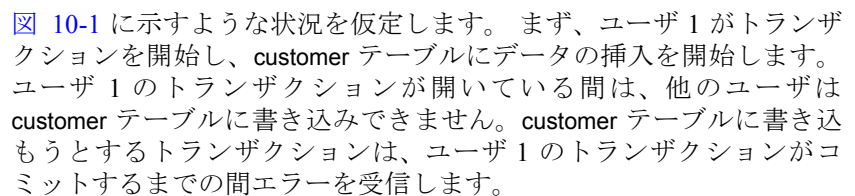
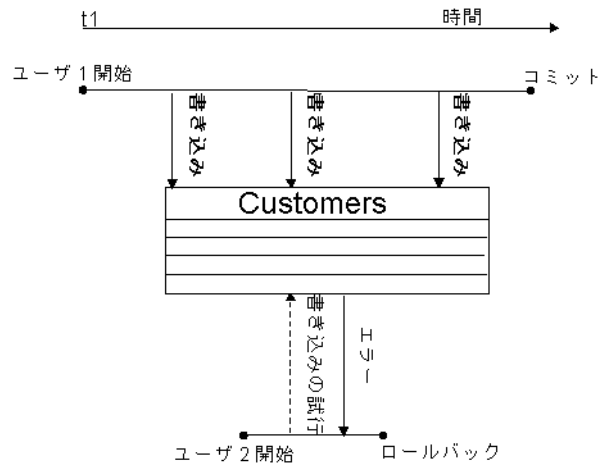
 **図 10-1** に示すような状況を仮定します。まず、ユーザ 1 がトランザクションを開始し、**customer** テーブルにデータの挿入を開始します。ユーザ 1 のトランザクションが開いている間は、他のユーザは **customer** テーブルに書き込みできません。**customer** テーブルに書き込みもうとするトランザクションは、ユーザ 1 のトランザクションがコミットするまでの間エラーを受信します。

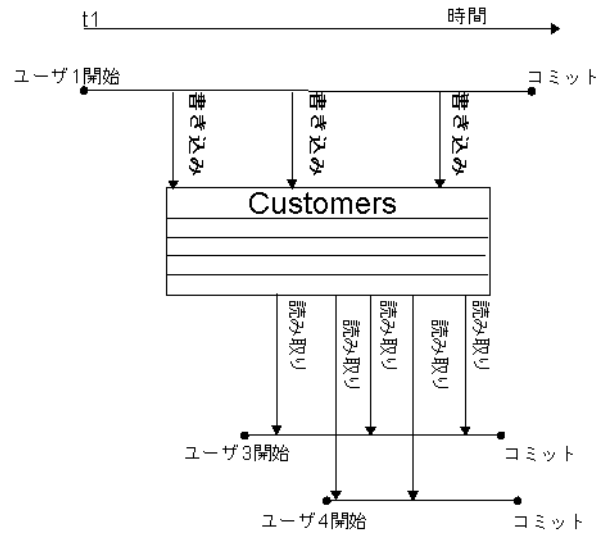
**図 10-1** では、ユーザ 2 はユーザ 1 のトランザクションがコミットする前に書き込みもうとしてエラーを受信します。ユーザ 2 のアプリケーションは、トランザクションをロールバックするか、または異なるテーブルへ書き込みを行うかを決定します。ただし、ユーザ 2 は、同じトランザクション内で **customer** テーブルへの書き込みをもう一度行うことはできません。

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

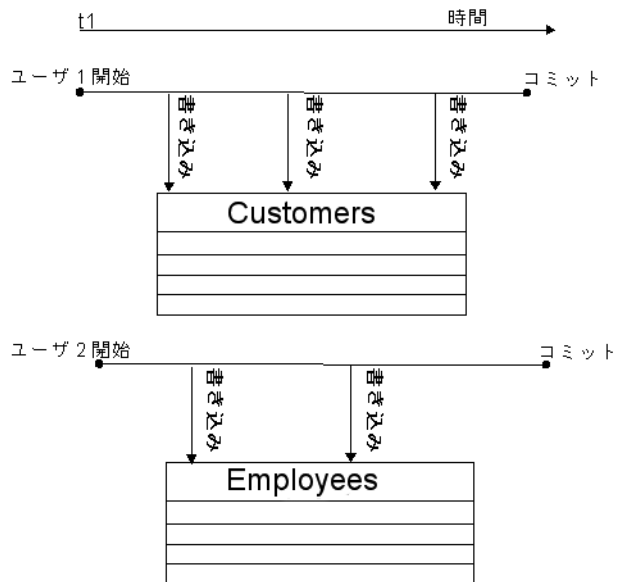


## データベースでの複数の書き込みユーザと読み取りユーザ

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、複数の読み取り専用ユーザと読み取り／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](#)」(461 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

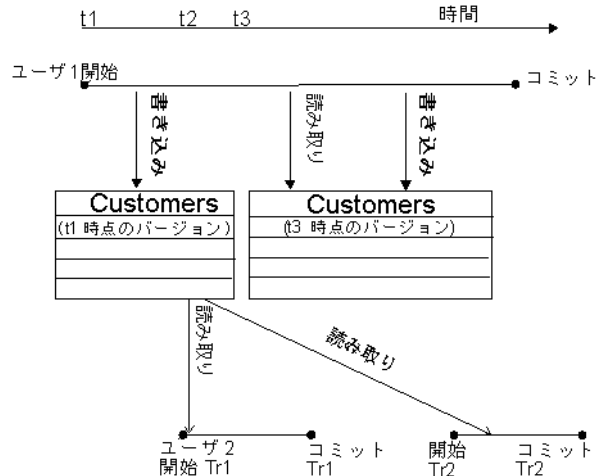
書き込みトランザクションがコミットすると、コミットされたデータが結果として生じます。各トランザクションはトランザクションが開始した時点で、データベースの最新のコミットされたバージョンを使用します。トランザクションがコミットするまでそのバージョンを使用します。

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取り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取る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、読み取り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



書き込みユーザーが参照できるデータは、その書き込みユーザー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ーが参照できるデータは、その書き込みユーザー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、ユーザー 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、ユーザー 1 が直前に行った挿入が反映されます。

ユーザー 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、ユーザー 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。ユーザー 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ー 1 の変更を参照できます。

### 読み取り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。

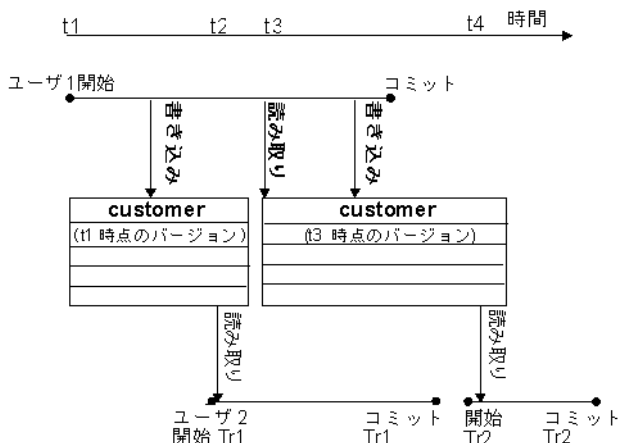
- あるユーザーの読み取りトランザクションのコミットが、同時実行の書き込みトランザクションがコミットしそのユーザーが新しい読み取りトランザクションを開始する前に行われる場合、バージョンは同じままです。

- 読み取り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取り専用の場合も読み取り／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 (455 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 の最初の読み取りトランザクション (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 : 読み取り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ときを認識し、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](#)」(468 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、このデフォルト値を上書きできます。

「ローカル・テンポラリ・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・テーブル内の任意のローを参照できます。このテーブルはユーザが接続を解除すると削除されます。ローカル・テンポラリ・テーブルを宣言すると、Sybase IQ は、トランザクションをオートコミットしないでセーブポイントを発行します。他のタイプのテーブルに対するデータ定義操作の場合と同様です。ローカル・テンポラリ・テーブルのデータをコミットしてからインデックスを作成してください。コミットされていないデータを使用してインデックスを作成しようとした場合、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・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・テーブルを区別しません。これは、テンポラリ・テーブル内のデータには1人のユーザしかアクセスできないため、複数の書き込みトランザクションが1つのテンポラリ・テーブルに対して開くことはないからです。

## 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、同時実行の読み取り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、複数の読み取り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変更できます。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行い、読み取り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、読み取りアクセスは可能です。データ定義操作は、この一般的な読み取り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) 操作には、挿入、削除、更新、クエリが含まれます。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取り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。

- 読み取り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取り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取り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 文を発行しない限り、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) 操作には、CREATE、DROP、ALTER が含まれます。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すると、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。

テーブル <テーブル名> で DDL コマンドがすでに実行中なので、このテーブルで DDL コマンドを実行できません。

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。

実行時 SQL エラー

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。

CHECKPOINT コマンドが実行中なので、要求されたコマンドを実行できません。

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。

**DROP DBSPACE コマンドが実行中なので、要求されたコマンドを実行できません。**

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。

**CREATE DBSPACE コマンドが実行中なので、要求されたコマンドを実行できません。**

テンポラリー・テーブルに関する特別な規則については、「[テンポラリー・テーブルのバージョン管理](#)」(457 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同時実行性の規則

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、インデックス作成により影響を受けるテーブルに対する **SELECT** と同時実行が可能です。**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。

### 制限されない **GRANT**、**REVOKE**、 **SET OPTION**

**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・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。

現在のトランザクション (TxnID1) で書き込むために要求されたオブジェクトを開けません。別のユーザがトランザクション (TxnID2) で書き込みアクセスを行っています。

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](#)」(32 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、「[第 14 章 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、常に読み取りが可能
- ・ トランザクションの書き込みは、そのトランザクションが参照するスナップショットに反映される
- ・ トランザクションが開始すると、そのトランザクションは他のトランザクションが行った更新を参照できない

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取ります。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取ります。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取ると、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。OLTP データベースでのトランザクションは短く、少数のローにしか影響しないことが多いです。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、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。ただし、チェックポイント時間を調整したり、明示的にチェックポイントを発行したりする必要はほとんどありません。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ー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・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、各永久テーブルの最新のコミットされたバージョンしかありません。

システム障害からのリカバリ時に、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度開きます。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・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が起動できない場合や、ユーザが接続できない場合の対処方法については、「[第 13 章 システムのリカバリとデータベースの修復](#)」を参照してください。リカバリには、サーバ・ログと IQ メッセージ・ログの情報が必要です。

システム障害の発生後、できればユーザに接続を許可する前に、ストア・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・ログが必要な領域量は非常に少なく、各コミットされたトランザクション当り 128 バイトです。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も非常に少量です。

トランザクション・ログが削除されるのは、次の場合です。

- 完全バックアップ後 (常時)
- インクリメンタル・バックアップ後 (オプション)
- バックアップ・ファイルがメディア障害後にリストアされ、新しいログが開始した後 (常時)

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](#)」(507 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。他のユーザの読み取り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、スリー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一掃する。どちらの制御も、パフォーマンスへの影響はあまりありません。
- ディスク領域が問題となる場合がある。重複するバージョンを格納すると、同時に使用されるバージョンの数やサイズによっては大量のディスク領域が使用される可能性があります。メタデータとデータベース・ページ・バージョンは、**RELEASE SAVEPOINT**の実行時か、または特定のバージョンを参照可能な最後のトランザクションのコミットまたはロールバック時に削除されるまで、保持されます。削除後の領域は再利用されます。

ロックによる遅延はほとんどありません。個々のコミット、ロールバック、チェックポイントが、他の読み取り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** の読み取り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取り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 DB 領域を追加して、ディスク領域を増加しなければならない場合があります。削除に必要な領域量は、データ・ページ上でのデータの分散状態によって異なります。削除されるローのサイズや数は関係ありません。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。削除中のインデックスのサイズに比例するか、最悪の場合、等しくなります。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。

### ディスク領域の不足

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おいてください。

削除中の領域不足によって、他のクエリ・ユーザに影響をおよぼさないようにします。

領域が不足し、別の DB 領域を追加するだけの十分なディスク領域がない場合は、データベース・エンジンを停止して再起動してください。データベースがロールバックされます。これで、ローの削除を、より小さなトランザクションに分けて実行できます。

---

**注意** DROP TABLE と DROP DATABASE は、バージョン・ページを作成しないで、テーブルまたはデータベースとそこにある全データを削除するため、これらのコマンドを使用するために領域を追加する必要はありません。

---

## 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](#)」(471 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

カーソルは、そのメンバシップ、つまりカーソルが返すデータ・ローが、カーソルが開いた時から閉じるまでの間に変わることが可能な場合、依存적であるといいます。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적、結果セットの順序と値については依存적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。

オプション **FORCE\_NO\_SCROLL\_CURSORS** を **ON** に設定すると、すべてのカーソルを非スクロールにできます。数百万といった大量のローを取り出すような場合、このオプションを使用して、テンポラリ記憶領域の必要量を節約できます。

## 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、読み取り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、個別の文としてではなく、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、プロシージャ内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 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 |
|----------------------------------|-----|
| <a href="#">各国語と文字セットの概要</a>     | 481 |
| <a href="#">ソフトウェアの文字セットについて</a> | 483 |
| <a href="#">ロケールについて</a>         | 485 |
| <a href="#">照合について</a>           | 485 |
| <a href="#">文字セット変換について</a>      | 488 |
| <a href="#">各国語と文字セットに関する作業</a>  | 491 |
| <a href="#">互換性の問題</a>           | 496 |
| <a href="#">パフォーマンスの問題</a>       | 496 |

## 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 Server — データベース管理ガイド> 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。

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** 関数は、Sybase Unicode Infrastructure Library (Unilib<sup>®</sup>) の代わりに、International Components for Unicode (ICU) ライブラリを使用します。Sybase IQ の 15.0 より前のバージョン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・キー値を再生成する必要があります。

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。

- **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、International Components for Unicode (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 3 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。

カスタム照合の作成は、Sybase IQ ではサポートされません。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 Behavior Changes in Sybase IQ 15.0](#)」を参照してください。iqunload ユーティリティを使用して、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 15.1 に移行できます。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](#)」(496 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の設定を行えない場合は、データベースで使用する照合を自分で決定し、さらに、データベース・システム間で整合性を保ったままデータ交換を行えるように文字セット変換を行うか決定する必要があります。

## ソフトウェアの文字セットについて

シングルバイト文字とマルチバイト文字など、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 Server — データベース管理ガイド> 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 Server — データベース管理ガイド>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 Server — データベース管理ガイド>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。

### 各ソート位置への複数の文字の関連付け

各ソート位置には複数の文字を関連付けることができます。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等に扱う場合に便利です。

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 coll = '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として扱われます。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 Müller は同じとみなされますが、1252DEU では違くとみなされます。1252DEU ではウムラウトが付いた文字は別物と判断されるため、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。ロケールの詳細については、『SQL Anywhere Server — データベース管理ガイド>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

用意された照合および特定の環境下で使用する照合の推奨については、『SQL Anywhere Server — データベース管理ガイド>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」および「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、推奨される照合の詳細については、『SQL Anywhere Server — データベース管理ガイド>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」および「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 照合は、ANSI または OEM のどちらかを指定されたコード・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です。

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。

- 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、と (¥xC6 and ¥xE6) に対応する合字は、A と a の後にソートされます (A、a、æ、Æ)。Adaptive Server Enterprise でのソート順は、A、a、Æ、æ 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す。

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。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ブ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。

カラムカラム名が見つかりません。

カラム名に入る文字列は、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。

使用する文字が各文字セットに入っているかを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にかかわらず、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) になっている、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

接続文字列の文字セット変換は、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 文字にトランケートされます。
- 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。データベース・サーバの起動時に `-ct-` コマンドライン・スイッチを使用した場合は、文字セット変換は無効になっています。

## 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](#)」(493 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](#)」(485 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](#)」(490 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](#)」(493 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](#)」(495 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 15.1 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、「[文字セット変換の回避](#)」(490 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 Server — データベース管理ガイド> 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、システム機能を使って決定できます。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 Server — データベース管理ガイド> 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 デフォルトの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](#)」(493 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・ラベルで、*language\_code* は『SQL Anywhere Server — データベース管理ガイド>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からの言語コードです。

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。次に例を示します。

```
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。

次に示すのは、sh または ksh シェルの場合の例です。

```
LC_ALL= american.iso88591;export LC_ALL
```



次に示すのは、`csh` または `tsch` シェルの場合の例です。

```
setenv LC_ALL american.iso88591
```

Sybase IQ は、`INSERT...LOCATION` 文の実行時にロー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作成の詳細については、『SQL Anywhere Server — データベース管理ガイド> 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。次に例を示します。

```
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の詳細については、『SQL Anywhere Server — データベース管理ガイド> 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' よりも前になります。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場合の方が、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高くなります。

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 BANANA の方が前になります。ソート順をより適切にして、大文字小文字を区別するデータベースを作成する際、パフォーマンスが多少損なわれてもよい場合は、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** 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。

---

# 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                                     | ページ |
|------------------------------------------|-----|
| <a href="#">データの保護</a>                   | 497 |
| <a href="#">データベースのバックアップ</a>            | 498 |
| <a href="#">仮想バックアップ</a>                 | 518 |
| <a href="#">システム・レベルのバックアップの実行</a>       | 521 |
| <a href="#">データベースの検証</a>                | 525 |
| <a href="#">データベースのリストア</a>              | 526 |
| <a href="#">アテンドなしバックアップ</a>             | 543 |
| <a href="#">バックアップとリストアに関する情報の取得</a>     | 544 |
| <a href="#">データ・バックアップ方式とリカバリ方式の決定</a>   | 548 |
| <a href="#">読み取り専用ハードウェアによるデータのアーカイブ</a> | 552 |

## 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にもかかわらず、コンピュータまたはオペレーティング・システムが停止すると発生します。これはコンピュータを不適切にオフにしたり再起動したとき、または他のアプリケーションによってオペレーティング・システムがクラッシュし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するようにしてください。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、Adaptive Server 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・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・テーブルを含んでいるイニシャル・テンポラリ DB 領域です。これは IQ テンポラリ・ストアと呼ばれ、*dbspace-name* は **IQ\_SYSTEM\_TEMP** です。*dbfile* は、IQ テンポラリ・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。データベースを完全にコピーします。
- 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atalog・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 テンポラリー・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。

バックアップの終了時に、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されません。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、および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時にトランザクション・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 データベースのカタログ・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。このデータベースをリカバリするにはトランザクション・ログ・ファイルが必要で、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 Server — SQL の使用法』を参照してください。また、カタログ・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・マシンに Sybase IQ をインストールして設定します)。
- 2 セカンダリ・マシンのフル・バックアップを定期的に行います。
- 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・メディアで均等に分散されません。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ります。そのため、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」 \(537 ページ\)](#) 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ド・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」 \(525 ページ\)](#) 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](#)」(552 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、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・ドライブも使用できます。

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際、複数の TO 句を指定して、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。

この制限数は、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、「[DBA および RESOURCE 権限の付与](#)」(392 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。

- アテンンド付き。アテン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。
- アテンンドなし。アテンンドなしのモードではオペレータがその場にいら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、それに従っ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、アテンションレス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、「[アテンションレスバックアップ](#)」(543 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ので、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) を保持する必要があります (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ド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、バックアップを再開するには、8KB 以上のディスク空き領域を確保する必要があります。

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」(543 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

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せん。例外を次に示します。

- 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。
- チェックポイントまたは DBCC を発行するコマンドはありません。

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ダ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ダ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](#)」(509 ページ)を参照してください。

### バックアップの種類の指定

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](#)」(518 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の選択方法については、「[日常のバックアップの計画](#)」(548 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 ENCAPSUAUED '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](#)」(518 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` 句には、バックアップの送信先となるディスク・ファイルやシステム・テープ・ドライブを指定します。`TO archive_device` 句は、送信先のファイルまたはデバイスごとに 1 つ指定します。この句は、少なくとも 1 つ必要です。`BACKUP` を実行すると、指定した全デバイスに対し、並行して (同時に) 出力が分散します。

Sybase IQ は、最大 32 のハードウェア・デバイスのバックアップをサポートしています。バックアップ時間を短縮するには、コアごとに 1 つまたは 2 つのデバイスを指定して、ハードウェアおよび IO の競合を回避します。バックアップ・デバイスごとに複数のファイルが作成されないように **BACKUP** コマンドの **SIZE** パラメータを設定し、**BACKUP** コマンドの **BLOCK FACTOR** 句で使用される値を検査します。『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。

**BACKUP** は情報を複数のデバイスに等分に分散させようとしませんが、実際にそうなる保証はありません。

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することができます。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、このパス名をサーバが開始されたロケーションに基づくパス名と解釈し、バックアップ時に正確なロケーションを認識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、文字列中に出現する ~~¥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。

---

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。

### テープ・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。

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](#)」(517 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。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必要がある場合は、その他の **BACKUP** オプションについて、『SQL Anywhere Server — 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。デバイスのオープンに成功するか、[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* 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、読み取り専用の DB 領域のヘッダ・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。読み取り専用の DB 領域は変更されないことが確定しているので、仮想バックアップとリストアで読み取り専用の DB 領域のリストアが必要なのは、読み取り専用の DB 領域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 DB 領域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](#)」(552 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 iuser'
```

### 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 DB 領域の 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 DB 領域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](#)」(547 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を使用して 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 \(http://www.sybase.com/support/techdocs/\)](http://www.sybase.com/support/techdocs/) を参照してください。

---

## 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・コマンドによって 0 以外のステータスが返された場合、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、予想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、これを読み取り専用にしてデータベースを正しく停止し、データベースを使用するときに読み書きができるように設定します。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かめるため、.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、スクリプトによってチェックすることができます。スクリプト実行中にデータベースが起動された場合、.iqmsg ファイルの方が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、SYSDBSACES 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・ファイルは、リストアに必要なファイルではありませんが、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 `.iqmsg` ファイルに含まれている情報で確認できます。

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、必要に応じてすべてのメッセージにアクセスして診断できます。詳細については、「[メッセージ・ロギング](#)」(11 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 DB 領域が追加されると、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、*SYSFILE* (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](#)」(522 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](#)」(522 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 DB 領域ファイルの内容は、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を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すすめします。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。

次の表に、**sp\_iqcheckdb** のこの 3 つの検証モードのアクション、出力結果、処理速度を示します。

| モード        | 検出されるエラー                   | 出力          | 処理速度         |
|------------|----------------------------|-------------|--------------|
| allocation | 割り付けエラー                    | アロケーション統計のみ | 1 時間あたり 4TB  |
| check      | 割り付けエラー<br>ほとんどのインデックス・エラー | 利用可能なすべての統計 | 1 時間あたり 60GB |
| verify     | 割り付けエラー<br>ほとんどのインデックス・エラー | 利用可能なすべての統計 | 1 時間あたり 15GB |

**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。

### マルチプレックス・データベースの検証

`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。

### `sp_iqcheckdb` の同時実行性の問題

`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` 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。



- アーカイブの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か、管理者がデータベースに接続しても構いません。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のリストア目的では、実行できます。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、各自の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](#)」(387 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。稼動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、リストアしている場所で接続を開始した直後に

```
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 接続から非常時のアクセスができないことです。

- 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 (デフォルトの `.db` ファイル) とトランザクション・ログ (デフォルトの `.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」\(532 ページ\)](#) 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ら、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。

## 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 DB 領域を削除するとリストア中にこの DB 領域が削除されますが、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 DB 領域では、データベース作成時の名前は `file_name` カラムに常に反映します。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を移す場合は、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、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 15.1 に移動するには、次の作業を行う必要があります。

- 1 バージョン 12.6 の『インストールおよび設定ガイド』に説明されているマイグレーション・プロシージャを使用して、バージョン 12.6 ESD 1 以上にアップグレードします。
- 2 『15.1 インストールおよび設定ガイド』に説明されているマイグレーション・プロシージャに従います。

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

データベースのリストアには、『リファレンス：文とオプション』の「第 1 章 SQL 文」の「RESTORE 文」の構文を利用します。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しなければならないことに注意してください。

`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に基づく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そ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、バックアップ作成時と同じ数のアーカイブ・デバイスを指定する必要があります。

---

**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](#)」(509 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

ディスク・ドライブの 1 つに障害が発生した場合など、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、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

新しいデバイスへ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 領域を書き込むのに十分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、IQ DB 領域が残りの領域を使用します。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 (type ‘Msg’) がロー・パーティションに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は、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は、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させる場合は、**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ありませぬ。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 領域を作成した時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」(547 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限り、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」(537 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すると、リストアされる各ロケーションを正確に把握することができます。

---

## 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](#)」(545 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
```

このリストアを以前のリストアの直後に実行できません。

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](#)」(544 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・テープのセット (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限り、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 の後で、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 の後で、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 IQSybase 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、DBISQLC または DBISQL 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=node1/users/fiona/mydb.db;
links=tcip{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
```

現在では、上のコマンドを実行すると、「指定されたデータベースが見つかりません。」というエラーが返ります。

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
```

DBISQL (Java) 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

テーブル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](#)」(525 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、この 2 種類のリストア間は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、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・データベースを作成する場合は、『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](#)」(545 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。

## 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 DB 領域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、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。

## アテンダなしバックアップ

**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 種類のアテンダ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・デバイスがスタッカ・ドライブの場合、一連のテープは単一のドライブに自動的にロードされます。スタッカ・ドライブは、アテンダ付き、アテンダなし両方のバックアップで使用できます。

アテンダ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ダなしバックアップを行う場合は、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。

アテンダ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、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、アテンダ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、使用するバックアップ・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](#)」(541 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ると、`."` というプレフィックスが付けられてこのファイルが隠しファイルになります。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に、次のフィールドがカンマで区切られてバックアップ・ログに含められます。

- 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取り／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することができます。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。

DBISQL 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/rdisk/c2t0d1s7'
```

```
-- rename IQ_SYSTEM_TEMP to '/dev/rdsd/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おく必要があります。

## 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の決定

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する必要があります。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・バックアップと同等の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、しかし、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取り／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。バックアップとリストアの操作のパフォーマンス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 と RESTORE によって、指定したすべてのアーカイブ・デバイスからの IQ データの取得とそれらへの書き込みが並行して行われます。その後、カタログ・ストアが最初のデバイスに書き込まれます。バックアップ時間とリストア時間を短縮するには、優れた並行処理を行う必要があります。

Sybase IQ は、最大 32 のハードウェア・デバイスのバックアップをサポートしています。バックアップ時間を短縮するには、コアごとに 1 つまたは 2 つのデバイスを指定して、ハードウェアおよび IO の競合を回避します。バックアップ・デバイスごとに複数のファイルが作成されないように BACKUP コマンドの SIZE パラメータを設定し、BACKUP コマンドの BLOCK FACTOR 句で使用する値を検討します。『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・データのスプール

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プールすると、処理時間を短縮して効率的に作業することができます。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することが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、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することができます。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、読み取りは多数回) 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。読み取り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、読み取り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。

- 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**

- D* と *E* という、2 つの新しいメイン DB 領域を作成します。
- 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限り、*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限り、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。

- 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。
- 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* と競合しない限り、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。

- 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。
- 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* を起動するうえでは必要ありません。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](#)」(553 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、特殊なリカバリ・モードについて説明します。

## 内容

| トピック                                   | ページ |
|----------------------------------------|-----|
| <a href="#">リカバリと修復の概要</a>             | 557 |
| <a href="#">通常のリカバリ</a>                | 558 |
| <a href="#">データベースの検証</a>              | 559 |
| <a href="#">データベースの修復</a>              | 567 |
| <a href="#">強制リカバリ・モード</a>             | 576 |
| <a href="#">トランザクション・ログを使わない緊急リカバリ</a> | 581 |
| <a href="#">DBCC によってレポートされた問題の処理</a>  | 582 |
| <a href="#">DBCC エラー・メッセージ</a>         | 586 |

## 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することができます。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。

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。

### サーバ・ログと IQ メッセージ・ログ の調査

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て、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](#)」の「[sp\\_iqcheckdb プロシージャ](#)」の項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 7-4 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 <b>t1</b> の割り付けの検査                                  |
| sp_iqcheckdb 'allocation index t1c1hg'          | インデックス <b>t1c1hg</b> の割り付けの検査                            |
| sp_iqcheckdb 'allocation indextype LF table t2' | テーブル <b>t2</b> 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、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 — リークしたブロック
- 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・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ます。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](#)」(586 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。

| 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](#)」の項を参照してください。

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 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。`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](#)」(579 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](#)」(586 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<br>Missing Index RIDs<br>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 <b>Future Version</b> 、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。 |
| VDO Incorrect First Available Fields<br>VDO Incorrect Next Available Fields<br>VDO Incorrect Used Count Fields<br>VDO Incorrect In-use Bitvec<br>VDO Incorrect In-use Bitmap<br>VDO Incorrect Partial Bitmap<br>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。                                              |
| HG Missing Groups<br>HG Extra Groups<br>HG Extra Keys<br>HG Missing Keys<br>B-Tree Invalid Item Count<br>B-Tree Invalid Item Count<br>G-Array Empty Page Errors<br>G-Array Bad Group Type Errors<br>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だけです。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](#)」(584 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](#)」(584 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](#)」(586 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う。                                                                                   |
| Blocks Leaked<br>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<br>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<br>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](#)」(571 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](#)」(577 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。次に例を示します。

```
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ので、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](#)」(577 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は、`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` を使用するが、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。次に例を示します。

```
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

Sybase IQ には、たとえリカバリ情報に矛盾があってもサーバを起動できるように、特殊な強制リカバリ・モードが用意されています。強制リカバリ・モードを使用すると、破損情報を表示した後、データベースを修復できます。

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。

## 強制リカバリ・モードでのサーバの起動

データベースを開くときの例外またはアサートでサーバが起動しない場合は、サーバを強制リカバリで起動します。アロケーション・マップまたはチェックポイント情報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。

---

**注意** 強制リカバリは、通常のデータベース・リカバリではデータベースを稼働状態にリストアできない場合にのみ使用してください。

---

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 `-iqfrec` 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。次に例を示します。

```
start_iq -n my_server -x 'tcpip(port=7934)'
-gd dba -gm 1 -iqfrec 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。`-iqfrec` 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](#)」([571 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 `-iqfrec` 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・センタに問い合わせてください。

`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 DB 領域は残りません。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。このような場合に、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」があると云います。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` ストアド・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](#)」(574 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。

「[割り付けの問題の修復](#)」(574 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。

強制リカバリを使用したリーク領域のリカバリ

**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](#)」(577 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。次に例を示します。

```
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 マルチプレックスの使用』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。

構文  
説明

`start_iq -n server-name [ other-server-options ] -f`

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で終了します。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で終了します。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いことがあり、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 (`-iqfrec`) 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](#)」(586 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。 |



| DBCC メッセージ                                                                                                                                                                                                                                    | 説明／対処方法                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| VDO Incorrect First Available Fields<br>VDO Incorrect Next Available Fields<br>VDO Incorrect Used Count Fields<br>VDO Incorrect In-use Bitvec<br>VDO Incorrect In-use Bitmap<br>VDO Incorrect Partial Bitmap<br>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。 |
| Blocks with Multiple Owners<br>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。          |
| DBCC Meta-data Errors<br>Blockmap Invalid Chunksize Error Count<br>Blockmap Compression Bit Error Count<br>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<br>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](#)」(567 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](#)」(584 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 テンポラリー・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 はテンポラリー・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](#)」(577 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 'tcip{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 テンポラリー・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 'tcipip{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](#)」(579 ページ) および「[データベースの検証](#)」(559 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る。                                                                                         |
| Inconsistent Index                                                                                                                                                                                                                            | DBCC が矛盾を検出したインデックスの名前。                                                                                           |
| Extra Index RIDs<br>Missing Index RIDs<br>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 <b>Future Version</b> 、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<br>HG Extra Groups<br>HG Extra Keys<br>HG Missing Keys<br>B-Tree Invalid Item Count<br>B-Tree Invalid Item Count<br>G-Array Empty Page Errors<br>G-Array Bad Group Type Errors<br>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<br>VDO Incorrect Next Available Fields<br>VDO Incorrect Used Count Fields<br>VDO Incorrect In-use Bitvec<br>VDO Incorrect In-use Bitmap<br>VDO Incorrect Partial Bitmap<br>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う。                                                                                                 |
| Blocks Leaked<br>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、 <b>dropleaks</b> モードを使用してください。                                     |

| DBCC メッセージ                                                                                                                                           | 説明／対処方法                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Blocks with Multiple Owners<br>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<br>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、 <b>dropleaks</b> モードを使用してください。            |
| Freelist Updated                                                                                                                                     | 割り付けの修復が成功したことを示す。                                                                            |
| 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す。                                                       |
| Invalid Blockmap Unique ID Generator<br>Blockmap Unique ID Generator Updated<br>Invalid Transaction ID Counter<br>Transaction ID Generator Updated   | DBCC <b>resetclocks</b> 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<br>Blockmap Invalid Chunksize Error Count<br>Blockmap Compression Bit Error Count<br>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 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てください。                                         |
| DBCC Inconsistent Disk Block Headers<br>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。                                                          |
| 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。                                            |

| DBCC メッセージ                                                                | 説明／対処方法                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unowned LVC cells<br>Duplicate LVC cell rows<br>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。<br/>LVC は CLOB によっても使用されます。</p> |

# トラブルシューティングのヒント

## 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                                    | ページ                 |
|-----------------------------------------|---------------------|
| <a href="#">状況別の解決策</a>                 | <a href="#">590</a> |
| <a href="#">ネットワーク通信のトラブルシューティング</a>    | <a href="#">624</a> |
| <a href="#">診断ツール</a>                   | <a href="#">630</a> |
| <a href="#">サポート・センタへの問題の報告</a>         | <a href="#">641</a> |
| <a href="#">チェックリスト：サポート・センタに提出する情報</a> | <a href="#">647</a> |

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 マルチプレックスの使用』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」([641 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 \(http://www.sybase.com/mysybase\)](http://www.sybase.com/mysybase) を指定します。追加の有益な Sybase Web サイトについては、「[Sybase EBF とソフトウェア・メンテナンス](#)」([xxii ページ](#))の項を参照してください。

## 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 \(dbisql/dbisqlc\) 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](#)」(630 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](#)」(592 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](#)」(577 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](#)」(602 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](#)」(571 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](#)」(559 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](#)」(559 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、「[インデックス・エラーの修復](#)」(571 ページ) の項で、DBCC を使用してインデックスの問題を修復する手順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、「[割り付けの問題の修復](#)」(574 ページ) の項で、DBCC を使用して割り付けの問題を修復する手順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。

```
データベース "dbname" (/dbdir/dbname.db) を起動しています。
Fri Apr 27 2009 10:53 トランザクション・ログ : dbname.log
エラー : トランザクション・ログ・ファイルを開けません
-- ログ・ファイル "dbname.log" を使用できません。データベース・
ファイルがログ・ファイルより後に使用されています。トランザクション・
ログ・ファイルを開けません。
```

-- ログ・ファイル "dbname.log" を使用できません。データベース・ファイルがログ・ファイルより後に使用されています。

Fri Apr 27 2009 10:53 でデータベース・サーバが停止しました。

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](#)」(581 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。

トランザクション・ログ : /dbdir/dbname.log...

エラー : トランザクション・ログ・ファイルを開けません。

-- そのようなファイルまたはディレクトリは存在しません。

トランザクション・ログ・ファイルを開けません。

-- そのようなファイルまたはディレクトリは存在しません。

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](#)」(581 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・センタに問い合わせてください。

---

**警告！** この手順はきわめて危険性が高いので、非常時以外は使用しないでください。

---

**サーバ名がネットワーク上で重複している** 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 エラー : -85

通信エラーが発生しました。

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。`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。

その名前のデータベース・サーバはすでに起動しています。

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と、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。

```
TCPIP リンクを開始しています ...
TCPIP 通信リンクを起動しませんでした。
要求された通信リンクを初期化できません。
...
DBSPAWN エラー : -85
通信エラーが発生しました。
```

サーバの起動に失敗しました。

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。

```
TCPIP リンクを開始しています ...
デフォルト・ポートで起動できません。ポート 49152 で起動します。
TCPIP リンクは正常に開始しました。
要求を受信中です。
...
サーバは正常に開始しました。
```

**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」の項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。『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](#)」(602 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 Start Database Server ユーティリティ、または iqsrv15 コマンドを使用します。

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** `unlimit` は、ソフト制限にのみ影響します。ハード制限がある場合は、カーネル・パラメータを設定して、それらを変更してください。

---

- 使用中の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する。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。

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・ファイルの格納場所は、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。次に例を示します。

```
iqsrv15 @iqdemo.cfg -x 'tcpip(port=1870)' iqdemo
```

この例では、`nx` 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、メッセージは指定されたファイルに保存され、サーバ・ウィンドウに表示されます。

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。次に例を示します。

```
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、「[起動のトラブルシューティング、シャットダウン、接続](#)」(118 ページ) の項を含む、「[第 2 章 Sybase IQ の実行](#)」および「[第 3 章 Sybase IQ の接続](#)」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](#)」(607 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](#)」の項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](#)」(607 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](#)」(202 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](#)」(637 ページ) および「[サーバ要求のロギング](#)」(637 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、次の表に示す情報を (この順序で) 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 &gt;# sa_conn_properties.out</code> | 接続情報           |
| <code>sa_conn_info &gt;# sa_conn_info.out</code>             | 接続情報           |
| <code>sa_db_properties &gt;# sa_db_properties.out</code>     | データベースのプロパティ情報 |
| <code>sa_eng_properties &gt;# sa_eng_properties.out</code>   | サーバのプロパティ情報    |
| <code>sp_iqstatus &gt;# sp_iqstatus.out</code>               | データベースのステータス情報 |
| <code>sp_iqconnection &gt;# sp_iqconnection.out</code>       | 接続情報           |
| <code>sp_iqtransaction &gt;# 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が役立つことがあります。「[サポート・センタへの問題の報告](#)」(641 ページ)の項を参照してください。



- サーバの応答がないとき、ファイル *DumpAllThreads* または *dumpallthreads* をディレクトリ *\$IQDIR15/logfiles* (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。「[サポート・センタへの問題の報告](#)」(641 ページ) の項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](#)」の項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、サーバ上のユーザー数を減らす
- 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 コマンドを実行するか、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|iqlsrv15"

/usr/ucb/ps -awwwlx | egrep "PPID|iqlsrv15"
```

- 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」\(641 ページ\)](#)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](#)」(641 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 *stktre-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。

接続できません。

または

データベースに接続できませんでした。

その後ろに次のメッセージが続きます。

ユーザ ID またはパスワードが無効です。

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。
- 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。

接続できません。

または

データベースに接続できませんでした。

その後ろに次のメッセージが続きます。

指定されたデータベースが見つかりません。

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。

dbisqlc または dbisql を使用し、*utility db* に接続する際にバックアップからデータベースをリストアしたときは、DBF パラメータと接続するデータベース・ファイル名を指定する必要があります。詳細については、「[リストア後の再接続](#)」(539 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する必要があります。
- 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。

接続できません。

データベース・サーバに接続できる限界数を超えています。

- 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。

接続に失敗しました。  
データベース・サーバは実行されていません。

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。

データベースに接続できませんでした。  
データベース・サーバが見つかりません。

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・サーバが起動され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たとします。

起動できません。サーバが見つかりませんでした。

または

データベース・サーバは実行されていません。

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](#)」(607 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/dbisqlc) の問題

この項では、dbisql および dbisqlc 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 Java が iAnywhere JDBC ドライバを介して接続している
- dbisql Java のバージョンが 7.04 より上である

dbisql Java 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 と共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。

- dbisql Java を ODBC ドライバを介して接続します。
- 次のいずれかを使用できます。
  - Sybase IQ 15.1 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;
```

## dbisqlc ウィンドウが UNIX で動作しない

### 考えられる原因

- 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** Sybase IQ に付属する端末情報拡張ファイル (.tix) をインストールし、必要に応じて編集します。このファイルには、ファンクション・キーと特殊キー・シーケンスの定義が格納されています。端末情報拡張ファイルのインストールの詳細については、マニュアル『Sybase IQ の概要』の「[第 3 章 サーバの起動と接続](#)」の「[Interactive SQL を使用したデータベースへの接続](#)」の項を参照してください。

## 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と、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と、ディスク領域を追加できないことがあります。したがって、ディスク領域を使い果たす前に、ディスク領域が残り少ないことを認識し、新しい DB 領域を追加することが大切です。

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](#)」(612 ページ) の項を参照してください。

## 対処法

表 tSybase 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、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。

```
%2 DB 領域内の領域が不足しています。%1
[EMSG_OUT_OF_DBSPACE: SQL Code -1009170L,
SQL State QSB66, Sybase Error Code 20223]
```

%2 は DB 領域の名前です。

このメッセージは、次のエラー・メッセージに置き換わりします。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 -- トランザクションはロールバックされました。"
```

```
[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: すべてのバッファ・キャッシュ・ページが使用中です。
バッファ・キャッシュのサイズを増やすよう DBA に依頼し
てください。%1
[MSG_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。

```
使用したメイン IQ ブロック : ,10188 of 12288,
82%, Max Block#: 134840
使用したテンポラリ IQ ブロック : ,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、できる限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](#)」(577 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。

```
CHECKPOINT 操作中に領域が不足しました。
[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することができない場合は、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。

## DB 領域のサイズの管理

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、既存のテンポラリ DB 領域に IQ\_SYSTEM\_TEMP という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](#)」(612 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_dbSPACE iqdba@iqdemo.com
    < ./temp_m_file');
    call xp_cmdshell('/bin/rm -rf ./temp_m_file');
end if;

if tu*100/tt > 95 then
    call xp_cmdshell('date > ./temp_file');
    call xp_cmdshell('mailx -s add_temp_dbSPACE iqdba@iqdemo.com
    < ./temp_file');
    call xp_cmdshell('/bin/rm -rf ./temp_file');
end if;

end

```

次のコードはタイマベースのイベントを作成します。イベントは領域の使用状況をモニタし、権限のない操作で領域が不足した場合に発生する可能性がある予期しないロールバックを回避します。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的詳細については、『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、変更、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。-iqmt のデフォルト値は  $(60 * \text{numCPU}) + (2 * \text{num\_conn}) + 5$ 。スレッドの総数 (-iqmt と -gn の合計) が、64 ビット・プラットフォームでは 4096、IBM AIX 32 ビット・サーバでは 1000、それ以外の 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。IBM AIX プラットフォームでは、このオプションが start\_iq 起動ユーティリティによって自動的に 450 に設定されます。
- リストア中に、サーバのスレッドが不足したか、接続に十分なスレッドが使用できない場合、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) の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」と「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 ( <i>stktrc-YYYYMMDD-HHNNSS_#.iq</i> ) 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     | <ul style="list-style-type: none"> <li>• この問題を回避するには、32 ビット・オペレーティング・システムではサーバ・パラメータ -iqtss を 300 に、64 ビット・オペレーティング・システムでは 500 に設定して Sybase IQ を再起動します。32 ビット・システムでは、-iqtss を増やすために、LOAD_MEMORY_MB を減らすことが必要な場合があります。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。</li> <li>• 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。</li> </ul> |

## ヒープ/ロード・メモリの不足

- |         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 考えられる原因 | エラー「利用可能なすべての仮想メモリは使用されています...」が表示された場合は、仮想メモリが不足しています。                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 対処法     | <p>前のエラーが発生した場合は、いくつかの対応が可能です。</p> <ul style="list-style-type: none"> <li>• <code>LOAD_MEMORY_MB</code> をゼロ以外の値に設定して、<code>LOAD</code> コマンドが使用できる仮想メモリ容量の上限を設定できます。指定可能な最大値は 2000MB です。</li> <li>• また、<code>BLOCK FACTOR</code> または <code>BLOCK SIZE LOAD</code> の各コマンド・オプションを調整することもできます。これらのコマンド・オプションのデフォルトはそれぞれ 10000 と 500000 ですが、任意の値に設定することが可能です。デフォルトより小さい値に設定すると、ロードによって使用される仮想メモリの量が少なくなります。</li> <li>• 部分幅ロードと呼ばれる方法で、一度にカラムのサブセットだけをロードすることもできます。</li> </ul> |

## 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 | <p>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 <code>ipcs</code> 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。</p> <p><code>ipcs -a</code> 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、<code>OWNER</code> 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。</p> |
| 対処法     | <p>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 <code>ipcrm</code> 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 <code>-m</code> パラメータを、セマフォ ID 番号の指定には <code>-s</code> コマンドを次の形式で使します。</p>                                                                                                                                                                                 |

```
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 以降は、Sybase IQ 12.6 より前で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「データベースのアップグレードができません。」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](#)」(612 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「 <a href="#">第 8 章 ビューの使用</a> 」        |

### 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"><li>• 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。</li><li>• 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。</li><li>• 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。</li><li>• 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 の介入を必要としている。</li><li>• 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。</li></ul> |
| 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"><li>• 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&amp;チューニング・ガイド』の「<a href="#">第 5 章 パフォーマンスのモニタリングとチューニング</a>」の「<a href="#">バッファ・キャッシュのモニタリング</a>」の項を参照してください。</li></ul>                                                                                                                          |

- 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 `vmstat` ユーティリティや他のプラットフォーム固有のツール、または Windows パフォーマンス モニタを使用します。

必要に応じてバッファ・サイズをリセットします。『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](#)」の項を参照してください。

ページングをモニタリングして、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。

`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: 一意の値の数がインデックスを超えています。
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。次に例を示します。

```
DROP INDEX DBA.employee.emp_lname_LF
```

次に、`CREATE INDEX` 文を発行し、新しい HG インデックスを作成します。次に例を示します。

```
CREATE HG INDEX ON DBA.employee (emp_lname)
```

## ロック・テーブルに書き込めない

#### 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。

現在のトランザクション (TxnID) で書き込むために要求されたオブ

ジェクトを開けません。別のユーザがトランザクション (TxnID2) で書き込みアクセスを行っています。

#### 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](#)」(621 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 ñ 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](#)」(459 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` および `-gr` の値を変更するよう指示された場合は、『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](#)」で、これらのサーバ・スイッチの設定方法の詳細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・モニタを使用するには、「[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     | <code>-iqnumbercpus</code> 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「 <a href="#">第 1 章 データベース・サーバの実行</a> 」で、 <code>-iqnumbercpus</code> 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に役立つ情報の取得方法については、「[診断ツール](#)」(630 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。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覚えてもらうこともできます。

## 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の説明については、『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](#)」の項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。

### “起動できません。サーバが見つかりませんでした。” 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。

起動できません。サーバが見つかりませんでした。

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。

表t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。

### “通信リンクを初期化できません。” 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。

通信リンクを初期化できません。

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。

sp\_iqstatus ストアド・プロシージャの出力を次に示します。

|                           |                                                                      |
|---------------------------|----------------------------------------------------------------------|
| Sybase IQ (TM)            | Copyright (c) 1992-2009 by Sybase, Inc.<br>All rights reserved.      |
| バージョン :                   | 15.1.0.5027/0490416/P/GA/MS/<br>Windows 2000/32bit/16.04.09 02:11:41 |
| 現在の時刻 :                   | 27.04.09 14:09:00.648                                                |
| ビルド時間 :                   | 16.04.09 09:54:19                                                    |
| ファイル形式 :                  | 23 on 03/18/1999                                                     |
| サーバ・モード :                 | IQ Server                                                            |
| カタログ形式 :                  | 2                                                                    |
| ストアド・プロシージャ・リビジョン :       | 1                                                                    |
| ページ・サイズ :                 | 131072/8192blks/16bpp                                                |
| メイン DB ファイル数 :            | 2                                                                    |
| 領域外のメイン・ストア :             | N                                                                    |
| テンポラリ・ファイル数 :             | 1                                                                    |
| DB ブロック : 1-3200          | IQ_SYSTEM_MAIN                                                       |
| DB ブロック : 1045440-1055039 | iq_main                                                              |
| テンポラリ・ブロック : 1-1600       | IQ_SYSTEM_TEMP                                                       |
| 作成時間 :                    | 03.04.09 14:14:06.124                                                |
| 更新時間 :                    | 25.04.09 14:14:26.687                                                |
| メイン IQ バッファ :             | 255, 32Mb                                                            |
| テンポラリ IQ バッファ :           | 191, 24Mb                                                            |
| 使用したメイン IQ ブロック :         | 5915 of 11200, 52%=46Mb,                                             |



|                        |                                           |
|------------------------|-------------------------------------------|
|                        | 最大ブロック数 : 1051278                         |
| 使用したテンポラリ IQ ブロック :    | 81 of 800, 10%=0Mb, 最大ブロック数 : 161         |
| 使用できる予約済みメイン・ブロック :    | 1600 of 1600, 100%=12Mb                   |
| 使用できる予約済みテンポラリ・ブロック :  | 800 of 800, 100%=6Mb                      |
| IQ 動的メモリ :             | 現在 : 69mb, 最大 : 70mb                      |
| メイン IQ バッファ :          | 使用済み : 6, ロック済み : 0                       |
| テンポラリ IQ バッファ :        | 使用済み : 5, ロック済み : 0                       |
| メイン IQ I/O :           | I: L899/P3 O: C3/D91/P89 D:0 C:100.0      |
| テンポラリ IQ I/O :         | I: L4043/P0 O:C674/D718/P47 D:669 C:100.0 |
| 他のトランザクション・バージョン :     | 0 = 0Mb                                   |
| アクティブなトランザクション・バージョン : | 0 = C:0Mb/D:0Mb                           |
| 前回の完全バックアップ ID :       | 0                                         |
| 前回の完全バックアップ時間 :        |                                           |
| 前回のバックアップ ID :         | 0                                         |
| 前回のバックアップの種類 :         | なし                                        |
| 前回のバックアップ時間 :          |                                           |
| DB 更新 :                | 1                                         |
| 次の ISF バックアップ内のブロック :  | 0 ブロック : =0Mb                             |
| 次の ISI バックアップ内のブロック :  | 0 ブロック : =0Mb                             |
| ファイル暗号化ステータス :         | 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 ブロック行および使用したテンポラリ IQ ブロック行は、DB 領域の使用中の部分を示します。使用中のブロックの割合 (行の中央の統計) が 90% 台後半の場合は、DB 領域を追加する必要があります。

- **メイン IQ バッファ行およびテンポラリ IQ バッファ行**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。
- **他のバージョン行**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・バージョン行**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・ブロック行および使用できる予約済みテンポラリ・ブロック行**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行およびテンポラリ IQ I/O行**には、I/O ステータスが IQ メッセージ・ログと同じフォーマットで表示されます。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](#)」(634 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。

```
テーブル 'partsupp', で、5 カラムの全幅挿入はレコード
1 で開始します。
2009-05-27 13:03:47 00000000002 挿入が開始しました :
2009-05-27 13:03:47 00000000002 partsupp
```

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。

```
2009-05-27 13:03:49 00000000002
[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 が毎回同じ数のメッセージ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 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) |
| 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><b>注意:</b> 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には再利用できるからです。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> |

| 項目 | 説明                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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> <hr/> <p><b>注意</b> 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」\(612 ページ\)](#)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は、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・モニタの使用法について、『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・バッファ・キャッシュ・メッセージ

この行にはテンポラリ・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](#)」(646 ページ) を参照してください。

---

```
2009-05-12 09:34:42 0000000002 Txn 173
2009-05-12 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 `DBA.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ことがあります。`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 SQL` ログ、さらにスタック・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 SQL` ログのエントリを照合する場合は、「[接続情報の関連付け](#)」(646 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。

**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** SQL ログのエントリを照合する場合は、[「接続情報の関連付け」 \(646 ページ\)](#) を参照してください。

要求ログ・フォーマットはこのバージョンの Sybase IQ で変更され、より簡潔になり追加の情報を記録することができます。固定フォーマット行プレフィックスの代わりに、一般的な情報はカンマ区切りテキストで記録されます。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。最終的な要求ログは、以前のバージョンで作成されたログ・サイズの約 1/3 になります。

この Sybase IQ のバージョンでは、追加の情報は要求ログに記録されません。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。



プロシージャおよびトリガ内で実行される文は、オプションで記録できます。

クエリ・プランの省略形は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、プロシージャ・プラン文も記録されます。

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記録される限られた要求を示します。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
```

この行の後続行、“+13,C,1,UID=DBA”などの場合、オフセットは前の行から行います。この場合、“+13”は最終行以降 13 ミリ秒が経過したことを意味します。場合によっては、“=”は最終行以降 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。

プロシージャ **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。次に例を示します。

```

call sa_get_request_profile('/sys1/users/jones/iqreqs1_zr.log');
select * from satmp_request_profile;

```

プロシージャ **sa\_get\_request\_times** は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。次に例を示します。

```
call sa_get_request_times('/sys1/users/jones/iqreqsl_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「[サーバのコマンド・ライン・スイッチ](#)」の項、『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 Server — SQL の使用法』>「データベース・パフォーマンスのモニタリングと改善」>「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

サポート契約を購入済みの Sybase 製品のインストールには、定められた 1 人以上のユーザに対して、Sybase 製品の保守契約を結んでいるサポート・センタを利用する権利が付属します。マニュアル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](#)」(641 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に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_1¥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_1/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_1/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 メニューで、[スタート]-[ファイル名を指定して実行]  
-[<install\_path>¥IQ-15\_1¥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\_1¥bin64¥getiqinfo.bat] (64 ビット・プラットフォームの場合) を入力します。

##### 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。
- Sybase IQ このデータベース・サーバのエンジン名 (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 *stktrc-YYYYMMDD-HHMMSS\_#.iq* という名前で保存される (UNIX および Linux プラットフォームのみ)
- エラーが発生したコマンドまたはクエリ
- メッセージ・ログ・ファイル。 *dbname.iqmsg* という名前で、デフォルトではデータベース・サーバを起動したディレクトリに保存される
- クエリ・プラン (*.iqmsg* ファイルに記録される。この後の「注意」を参照)
- サーバ・ログ
  - UNIX の場合 : *IQ-15\_1/logfiles/<servername>.000n.stderr* および *IQ-15\_1/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。オプションを設定すると、データベース名に `.zr` 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。このログ・ファイルでは、サーバに対する個々の接続が接続ハンドルによって識別されます。

接続ハンドルはユニークではないため、Sybase IQ が独自の Sybase IQ 接続 ID を割り当て、この ID が Sybase IQ メッセージ・ファイルに表示されます。Sybase IQ 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。2 つのファイルは接続に別々の識別子を使用するため、`.zr` 出力を `.iqmsg` ファイルと照合し、特定の接続の情報を簡単に見つけることができません。

次の手順は、2 つのファイルの識別子を関連付けて、関連情報を見つける方法を示します。この例では、`.zr` 出力ファイルを `example.zr`、Sybase IQ メッセージ・ファイルを `example.iqmsg` とします。

### ❖ `.zr` ファイルと `.iqmsg` ファイルの接続情報を関連付けるには

- 1 `.zr` ファイルで、目的の接続を検索します。たとえば、次の接続を検索するとします。

```
.conn: 240215640
```

UNIX システムでは、次のコマンドを使用します。

```
grep 240215640 example.zr | grep CONNECT
04/19 06:42:060.690 ** REQUEST conn: 240215640 CONNECT
04/19 06:42:070.204 ** DONE      conn: 240215640 CONNECT
      Conn=569851433
04/19 06:46:17.646 ** REQUEST conn: 240215640 DISCONNECT
04/19 06:46:17.670 ** DONE      conn: 240215640 DISCONNECT
```

- 2 同じ行で、`Conn=` の後ろに続く番号を見つけます。この例では次の番号になります。

```
Conn=569851433
```

- 3 `.iqmsg` ファイルで、“Connection handle is” の後ろにこの番号が続いている行を検索します。次に例を示します。

```
grep 569851433 example.iqmsg
```



```
2009-04-19 07:46:57 0000000002 Connection handle is : 569851433. SA
connID: 2.User Name is : DBA.
```

この例の Sybase IQ 接続ハンドルは 000000002 です。

- 4 この接続を含むすべての行を .iqmsg ファイルから抽出します。

```
grep ' 0000000002 ' example.iqmsg
```

## 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](http://www.sybase.com/support/techdocs/) (<http://www.sybase.com/support/techdocs/>) を指定し、[MySybase] をクリックします。

## チェックリスト：サポート・センタに提出する情報

getiqinfo スクリプトを実行すると、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 (15.1 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<br>(Microsoft Windows 2008 Service Pack 1 など) |   |

## チェックリスト：サポート・センタに提出する情報

| 要求される情報                                                                                                                                                  | 値 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 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<br>(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<br>(server.nnnn.svrlog および server.nnnn.stderr)                                                                                               |   |
| スタック・トレース・ファイル<br>(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：オプションを設定し (Query_Plan、<br>Query_Detail、Query_Plan_After_Run、<br>Query_Plan_As_Html、Query_Plan_As_Html_Directory、<br>Query_Timing)、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 358

## A

Access

ODBC 設定 87

Adaptive Server Enterprise

text と image データの挿入 334

unichar データ型 365

unitext データ型 365

univarchar データ型 365

データの挿入 70, 333

Advanced Security オプション 414

ALL パーミッション 393

ALTER DBSPACE 文

ADD パラメータ 220

SIZE パラメータ 220

例 219

ALTER INDEX 文 341

ALTER TABLE 文

外部キー 436

ALTER パーミッション 393

ALTER 文

オートコミット 447

ANSI コード・ページ

選択 486

APPEND\_LOAD オプション

分割されたテーブル 323

AppInfo 接続パラメータ 127

ASCII

挿入時の変換 354

データ抽出 309

データ・フォーマット 301

変換オプション 348

変換のパフォーマンス 353

文字セット 483

AStart 接続パラメータ 129

ASTMP 環境変数

ディスク容量 118

AStop 接続パラメータ 130

AUTOINCREMENT

Ultra Light アプリケーション 427

デフォルト 426

符号付きデータ型 427

負の数 427

AutoPreCommit 接続パラメータ 129

AutoStart 接続パラメータ 129

AutoStop 接続パラメータ 130

AVG 関数 260

## B

BACKUP 文 507

BCAST 153

BCAST 通信パラメータ

説明 153

bcp サポート 301

BIT データ

使用可能なインデックス 252

変換 351

BLISTENER 通信パラメータ 153

説明 153

BLOB 334

BLOCK FACTOR

BACKUP 文のオプション 513

BLISTENER 153

BroadcastListener 通信パラメータ

説明 153

Broadcast 通信パラメータ

説明 153

## C

CATALOG ONLY  
 RESTORE オプション 541  
 CBufferSize 接続パラメータ 131  
 Certificate\_Password 通信パラメータ  
 説明 154  
 Certificate 通信パラメータ  
 説明 154  
 CharSet 接続パラメータ 130  
 CHAR データ  
 長さ 0 のセル 356  
 CHECK 条件  
 カラム 430  
 削除 432  
 テーブル 432  
 変更 432  
 ユーザ定義データ型 431  
 ClientPort 通信パラメータ 155  
 CLOB 334  
 CMP インデックス 265  
 推奨される使用法 265  
 制限 266  
 CommBufferSize 接続パラメータ 131  
 CommLinks 接続パラメータ 132  
 connect  
 パーミッション 390  
 ConnectFailed イベント・ハンドラ 385  
 ConnectionName 接続パラメータ 134  
 CONNECT 文  
 デフォルト・データベース 106  
 CONVERSION\_ERROR データベース オプショ  
 ン 369  
 COUNT DISTINCT  
 インデックス選択への影響 249  
 COUNT 関数 260  
 CREATE DATABASE 文 194  
 IQ RESERVE パラメータ 216  
 IQ SIZE パラメータ 216  
 TEMPORARY RESERVE パラメータ 216  
 TEMPORARY SIZE パラメータ 216  
 照合 482  
 CREATE DBSPACE 文 213  
 RESERVE パラメータ 217

SIZE パラメータ 217  
 CREATE INDEX 文 246  
 CREATE JOIN INDEX 文 225, 289  
 CREATE TABLE 文  
 コマンド・ファイル 174  
 例 221  
 CREATE パーミッション 381  
 CREATE 文  
 オートコミット 447  
 同時実行性の規則 461  
 CS\_TEXT\_TYPE 334  
 CS 接続パラメータ 130  
 -c スイッチ 39

## D

DatabaseFile 接続パラメータ 134  
 DatabaseName 接続パラメータ 136  
 DatabaseName 通信パラメータ  
 説明 156  
 DatabaseSwitches 接続パラメータ 137  
 DataSourceName 接続パラメータ 137  
 DATEPART  
 クエリ 270  
 DATETIME  
 変換オプション 348  
 ロード変換オプション 358  
 DATETIME データ型 359  
 変換フォーマット 359  
 ロードの最適化 353  
 DATE インデックス 269  
 推奨される使用法 269  
 他のインデックスとの比較 274  
 短所 273  
 長所 273  
 追加インデックス 274  
 DATE オプション 348, 356  
 DATE データ型  
 変換フォーマットの指定 356  
 ロードの最適化 353  
 DATE フォーマット  
 2 桁の日付の変換 358  
 DBASE フォーマット 301

- DBA 権限
  - 継承不可能 399
  - 説明 378
  - 付与 379, 392
- DBA (データベース管理者)
  - 責任 3
  - 定義 378
  - 役割 6
- DBA ユーザ 392
  - 削除 392
- DBCC
  - sp\_iqcheckdb インタフェース 559
  - インデックスと割り付けの検査 560, 561
  - インデックスの検証と修復 567
  - インデックスの修復 567
  - インデックスの問題の検出 583
  - インデックスの問題の分析 567
  - 実行時間 560
  - 出力 563
  - 出力メッセージ 586
  - 出力例 563
  - データベースの検証 559
  - 内部インデックスの検査 560
  - パフォーマンス 560
  - 割り付けエラーの検出 573
  - 割り付けの検査 561
  - 割り付けの検証と修復 571
  - 割り付けの修復 571, 574
  - 割り付けの問題の分析 571
- DBCC\_LOG\_PROGRESS オプション 525, 563
- DBF 接続パラメータ 134
  - 埋め込みデータベース 78
- DBISQL
  - 「Interactive SQL」参照 173
  - 概要 173
  - コマンド・ライン・パラメータ 75
  - 出力フォーマットの指定 302
  - 挿入、データ、対話的 335
  - トラブルシューティング 605
  - ログオン・ウィンドウ 76
- DBISQL Java 605
- DBKEY 接続パラメータ 138
- DBLOCATE ユーティリティ 120
- dblog ユーティリティ 540
- DBN 156
- DBN 接続パラメータ 136
- DBN 通信パラメータ
  - 説明 156
- DBO ユーザ ID
  - 所有されるビュー 234
- DBSpaceLogger イベント 613
- DBS 接続パラメータ 137
- DB 領域
  - DB 領域の不足状況 202, 607
  - IQ\_SYSTEM\_MSG 181
  - IQ\_SYSTEM\_TEMP 177
  - インデックス情報の表示 217
  - オフライン 573
  - 管理の例 216
  - サイズの変更 219
  - 削除 215
  - 作成 213
  - 作成時のファイルのロケーション 194
  - 作成の例 217
  - システム 177
  - 使用状況の表示 217
  - 定義 177
  - 命名 212
  - 読み込み専用 208
  - 領域の使用状況のモニタリング 613
  - 領域不足のエラー・メッセージ 608
  - ロー・デバイスへのリストア 533
- DDL
  - 説明 171
- DDL (データ定義言語) 10
- DDL ロック 461
- DEFAULT\_DISK\_STRIPING オプション 4
- DELETE パーミッション 393
- DELIMITED BY オプション 318
- Delphi
  - ODBC 設定 87
- DisableMultiRowFetch 接続パラメータ 138
- DLL 通信パラメータ 158
- DML 189
- DML (データ操作言語) 10
- DMRF 接続パラメータ 138
- DOBROADCAST 通信パラメータ 156

## 索引

DROP JOIN INDEX 文 225  
DROP TABLE 文  
    例 229  
DROP VIEW 文  
    制限 234  
    例 235  
DROP 文  
    オートコミット 447  
    同時実行性の規則 461  
DSN 接続パラメータ 137  
    説明 82  
DSS (意志決定支援システム) 2  
DTTM インデックス 269  
    推奨される使用法 269  
    他のインデックスとの比較 274  
    短所 273  
    長所 273  
    追加インデックス 274  
DumpAllThreads ファイル 599  
dumpdups  
    sp\_iqcheckdb オプション 562  
dumpleaks  
    sp\_iqcheckdb オプション 562  
dumpunallocs  
    sp\_iqcheckdb オプション 562

## E

ENC 140  
EncryptedPassword 接続パラメータ 140  
Encryption 接続パラメータ  
    説明 140  
ENC 接続パラメータ 140  
    説明 140  
EngineName 接続パラメータ 139  
ENG 接続パラメータ 139  
ENP 接続パラメータ 140  
Excel フォーマット 301

## F

FileDataSourceName 接続パラメータ 142

FileDSN  
    作成 92  
    接続パラメータ 82  
    分散 92  
FILLER オプション 345  
FIPS サポート 414  
FIXED フォーマット 301  
FORCE\_DROP オプション 584  
FoxPro フォーマット 301  
FP(1) インデックス 254  
FP(2) インデックス 254  
FP(3) インデックス 256  
FROM 句  
    UPDATE 文 375  
    ジョイン・インデックス 277

## G

getiqinfo スクリプト 642  
Getting Started CD xxi  
gm スイッチ 38  
    リカバリの影響 472, 558  
GRANT 文  
    DBA 権限 392  
    RESOURCE 権限 392  
    WITH GRANT OPTION 396  
    グループの作成 399  
    グループ・メンバシップ 400  
    新規ユーザ 390  
    パーミッション 393  
    パスワード 391  
    パスワードなし 402  
    プロシージャ 396  
GROUP BY 句  
    インデックス選択への影響 249  
GROUP パーミッション  
    継承不可能 399

## H

HASH\_THRASHING\_PERCENT オプション 619

Health Insurance Portability and Accountability  
Act 552

HG インデックス

NULL 値 263

NULL 値のマルチカラム 263

外部キー制約 261

クエリのパフォーマンス 263

自動作成 263

推奨される使用法 261

他のインデックスとの比較 262

短所 262

長所 262

追加インデックス 263

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 552

HNG インデックス 263

推奨される使用法 264

他のインデックスとの比較 265

短所 264

長所 264

追加インデックス 265

HOST 通信パラメータ 158

## I

IANA

ポート番号 166

iAnywhere JDBC ドライバ 605

ASA データベースへの接続 69

ICU ライブラリ 482

IDLE 143

Idle 接続パラメータ

説明 143

INSERT

分割されたテーブル 323

INSERT LOCATION 文 333

INSERT パーミッション 393

INSERT 文 276

VALUES オプション 330

インクリメンタル 331

整合性 418

説明 330

パフォーマンス 331

部分幅挿入 341

integer データ型

Adaptive Server Enterprise と Sybase IQ の

対応 364

Integrated 接続パラメータ 144

Interactive SQL

「DBISQL」参照 173

ウィンドウの問題 123

Interfaces ファイル

エントリの追加 70

IPv6 のサポート 414

IPX

サーバ設定 152

IP アドレス 127

ping 627

IP 通信パラメータ 158

IQ PAGE SIZE 198

IQ UNIQUE

値の変更 227

パフォーマンスの影響 226

IQ UNIQUE 制約 430

IQ UNIQUE テーブル・オプション 226

IQ UTILITIES

バッファ・キャッシュのモニタリング 599

iq\_bcp

LOAD TABLE USING FILE に置換 323

廃止 323

IQ\_SYSTEM\_MAIN

CREATE パーミッション 381

サイズのガイドライン 183

IQ\_SYSTEM\_MSG DB 領域 181

IQ\_SYSTEM\_TEMP

CREATE パーミッション 381

iqdemo データベース 8

IQDSEdit 70

iqdsn コマンド 85

iqgovern スイッチ 39

IQMsgMaxSize サーバ・プロパティ 14

IQMsgNumFiles サーバ・プロパティ 14

iqmsgnum サーバ起動スイッチ 14

iqmsgsz サーバ起動スイッチ 14

## 索引

iqmsg メッセージ・ログ 11, 14  
    アーカイブ処理 16  
    最大サイズの設定 15  
    ラッピング 15  
iqmsg メッセージ・ログのアーカイブ処理 16  
iqmsg ログ 14  
    アーカイブ処理 16  
    最大サイズの設定 15  
    ラッピング 15  
iqnumbercpus  
    サーバ・スイッチ 40  
iqwmem スイッチ 39  
ISO\_1 照合  
    説明 486  
ISO\_BINENG  
    デフォルトの照合 483, 495, 496  
ISQL  
    接続 70

## J

Java  
    Sybase IQ での使用 63, 201  
    メモリ要件 39  
jConnect ドライバ  
    IQ データベースへの接続 69  
JDBC  
    接続 63  
    接続の設定 70  
JDBC 接続  
    説明 69

## K

Kerberos 認証 414

## L

LANalyzer 628  
LANG 144  
Language 接続パラメータ  
    説明 144

LANG 環境変数 494  
LANG 接続パラメータ  
    説明 144  
LazyClose 接続パラメータ  
    説明 145  
LC\_ALL 環境変数 494  
LCLOSE 145  
LCLOSE 接続パラメータ  
    説明 145  
LDAP 159  
LDAP サーバ  
    LDAP 通信パラメータ 159  
LDAP 通信パラメータ  
    説明 159  
LF インデックス 259  
    推奨される使用法 260  
    他のインデックスとの比較 260  
    短所 260  
    長所 260  
    追加インデックス 261  
    ユニークな値の最大数を超える 620  
LF 通信パラメータ  
    説明 161  
Links 接続パラメータ 132  
LivenessTimeout 接続パラメータ 146  
LOAD 354  
LOAD TABLE 内 359  
LOAD TABLE 文 276  
    FILLER オプション 345  
    QUOTES オプション 320  
    QUOTES オプションの例 320  
    QUOTES キーワード 320  
    STRIP ON 句 319  
    STRIP キーワード 320  
    USING CLIENT FILE 323  
    USING FILE 句 323  
    整合性制約 317  
    説明 316  
    パフォーマンス 353  
    部分幅挿入 341  
    分割されたテーブル 323  
LOB 334  
LOCAL 160



LocalOnly 通信パラメータ  
 説明 160

LOCAL 通信パラメータ  
 説明 160

LogFile 接続パラメータ 147

LogFile 通信パラメータ  
 説明 160

LogFormat 通信パラメータ  
 説明 161

LOGIN\_MODE データベース・オプション  
 統合化ログイン 110

LogMaxSize 通信パラメータ  
 説明 162

LogOptions 通信パラメータ  
 説明 162

LOG 通信パラメータ  
 説明 160

LOPT 162

LOPT 通信パラメータ  
 説明 162

Lotus フォーマット 301

Low\_Fast インデックス  
 「LF インデックス」参照

LSIZE 162

LSIZE 通信パラメータ  
 説明 162

LTO 接続パラメータ 146

**M**

max\_days\_since\_login  
 超過 386

max\_failed\_login\_attempts  
 超過 386

MAX\_TEMP\_SPACE\_PER\_CONNECTION オプ  
 ション 203

MaxConnections 通信パラメータ  
 説明 163

MAXCONN 通信パラメータ  
 説明 163

MaxRequestSize 通信パラメータ  
 説明 164

MAXSIZE 通信パラメータ  
 説明 164

MESSAGE LOG  
 内容 326  
 例 328

Microsoft Access 618  
 ODBC 設定 87

Microsoft Visual Basic  
 ODBC 設定 87

money データ型 366, 368

MYIP 通信パラメータ 164

MySybase  
 アクセス 647  
 オンライン・サポート 647

## N

National Language Support  
 概要 481, 483  
 照合 492  
 説明 481  
 マルチバイト文字セット 487

NDIS  
 ドライバ 625

NEAREST\_CENTURY オプション 358

net.cfg ファイル 629

NetBIOS  
 サーバ設定 152

NetWare  
 ネットワーク・アダプタの設定 629

NEWID  
 デフォルト 427

NEWPWD 接続パラメータ 147

NOT NULL 制約 418

NOTIFY オプション  
 INSERT 文 279

Novell クライアント・ソフトウェア 625

NULL 359  
 挿入 330  
 デフォルト 428  
 部分幅挿入の結果 335  
 変換 361  
 変換オプション 348, 361  
 マルチカラム HG インデックス 263

## 索引

NULLS オプション  
DBISQL 304

NULL 値  
出力 304  
マルチカラム HG インデックス 263  
NULL バックアップ 518

## O

ODBC  
UNIX サポート 93  
UNIX 用の初期化ファイル 93  
接続パラメータ 125  
データ・ソース 82  
データ・ソースの設定 86  
ドライバの位置 100  
ODBC 接続  
説明 69  
ODBC データ・ソース  
jConnect の使用 69  
ODBC 変換ドライバ  
ODBC 設定 87  
ODI ドライバ 625  
OEM コード・ページ  
選択 486  
ON 句のジョイン 290  
Open Database Connectivity。「ODBC」を参照

## P

params.cfg ファイル 33  
Password 接続パラメータ 148  
ping  
TCP/IP 627  
PIPE\_NOT\_CONNECTED エラー 318  
PORT 通信パラメータ 166  
PreFetchOnOpen 通信パラメータ  
説明 165  
Prompt 接続パラメータ 150  
PUBLIC グループ 403  
PWD 接続パラメータ 148

## Q

QUERY\_TEMP\_SPACE\_LIMIT オプション 203  
QUOTES  
LOAD TABLE キーワード 320

## R

RCVBUFSZ 166  
RCVBUFSZ 通信パラメータ  
説明 166  
REAL データ型  
Adaptive Server Enterprise と Sybase IQ のデータ  
型の対応 365  
ReceiveBufferSize 通信パラメータ  
説明 166  
REFERENCES パーミッション 393  
RELEASE SAVEPOINT 文 468  
resetclocks  
sp\_iqcheckdb オプション 563  
RESOURCE 権限  
継承不可能 399  
説明 380  
付与 392  
RESTORE 文  
説明 531  
REVOKE 文  
説明 397  
ROLLBACK TO SAVEPOINT 文 468  
ROLLBACK 文 468  
ROW LOG  
内容 327  
例 328  
RSA サポート 414

## S

sa\_get\_request\_profile  
要求ログ・ファイルの分析 640  
sa\_get\_request\_times  
要求ログ・ファイルの分析 640  
SACHARSET 環境変数 493  
SALANG 環境変数 493

- SAVEPOINT 文
  - トランザクション 468
- SCSI テープ・バックアップ 504
- SELECT DISTINCT 射影 249
- SELECT パーミッション 393
- SELECT 文
  - INSERT 文内 335
  - ジョイン・インデックス 277
  - デリミタ 334
  - ビュー作成の制限 233
- SendBufferSize 通信パラメータ
  - 説明 166
- ServerName 接続パラメータ 139
- ServerPort 通信パラメータ 166
- Service Manager
  - サーバの起動 32
- SESSIONS 通信パラメータ 168
- SET 句
  - UPDATE 文 375
- SNDBUFSZ 166
- SNDBUFSZ 通信パラメータ
  - 説明 166
- SORT\_COLLATION オプション 481
- SORTKEY 関数 481, 482
- sp\_iq\_post\_process\_login 383
- sp\_iq\_process\_login 383
- sp\_iqcheckdb
  - allocation モード 561
  - check モード 560
  - DBCC\_LOG\_PROGRESS 525
  - DBCC\_LOG\_PROGRESS オプション 563
  - DBCC の機能 559
  - dropleaks モード 563
  - dumpdups オプション 562
  - dumpleaks オプション 562
  - dumpunallocs オプション 562
  - resetclocks オプション 563
  - verify モード 561
  - アロケーション・マップのリセット 563
  - インデックスと割り付けの検査 560, 561
  - インデックスの検証と修復 567
  - インデックスの修復 567
  - インデックスの問題の分析 567
  - 構文 559
  - 実行時間 560
  - 出力 563
  - 出力の解釈 573
  - 出力メッセージ 586
  - 出力例 563
  - データベースの一貫性のチェック 525
  - データベースの検証 559
  - 内部インデックスの検査 560
  - パフォーマンス 560
  - リソースの問題 566
  - 割り付けの検査 561
  - 割り付けの検証と修復 571
  - 割り付けの修復 571, 574
  - 割り付けの問題の分析 571
- sp\_iqcheckoptions ストアド・プロシージャ 203
- sp\_iqdbspace
  - DB 領域の使用状況 217
- sp\_iqdbspaceinfo
  - DB 領域の使用状況 218
- sp\_iquestdbspaces
  - 必要な DB 領域の見積もり 185
- sp\_iquestjoin 297
  - ジョイン・インデックス領域の必要量の  
見積もり 185
- sp\_iquestspace
  - データベース領域の必要量の見積もり 185
- sp\_iqindexinfo
  - インデックス情報の表示 218
- sp\_iqlocklogin 383
- sp\_iqrebuildindex 571
- sp\_iqstatus
  - 出力例 630
  - トラブルシューティングでの使用 630
- sp\_iqtransaction
  - ブロックしている書き込みの確認 620
- SQLCONNECT 環境変数
  - 接続 82
- SQL ログ
  - 夏時間 18
- START ROW ID オプション 341
  - INSERT 文 332
  - 説明 341, 344
  - 部分幅挿入 335, 342

start\_iq  
   コマンドが実行されない 595  
   トラブルシューティング 595  
   パラメータ 595  
 start\_iq ユーティリティ 26  
 StartLine 接続パラメータ 150  
 Start パラメータ  
   埋め込みデータベース 78  
 STRIP  
   LOAD TABLE キーワード 320  
 STRIP ON 句  
   廃止 319  
 STRIP オプション 319  
 SUM 関数 260  
 Sybase Central 59  
   DB 領域の作成 214  
   概要 173  
   カラム制約 432  
   カラムのデフォルト 424  
   起動 29, 72  
   グループの作成 400  
   グループへのユーザの追加 401  
   テーブルの作成 221  
   テーブルの変更 228  
   パーミッション 394, 397  
   ビューの削除 235  
   表示の問題 616, 624  
   ユーザの作成 391  
 Sybase Central におけるカラムのデフォルトの  
   編集 424  
 Sybase IQ  
   Adaptive Server Enterprise とのデータ型の  
   対応 363  
 Sybase IQ 12.5 サーバ 605  
 Sybase IQ ユーザ管理  
   概要 384  
   プロシージャのリスト 385  
 SyBooks CD xxi  
 SYSCOLAUTH ビュー  
   パーミッション 413  
 SYSCOLUMN テーブル  
   整合性 443

SYSDUMMY テーブル  
   パーミッション 412  
 SYSFILE テーブル  
   リストア後の file\_name 529  
 SYSFORIGNKEY テーブル  
   整合性 443  
 SYSGROUPS ビュー  
   パーミッション 413  
 SYSGROUP テーブル  
   パーミッション 412  
 SYSPROCAUTH ビュー  
   パーミッション 413  
 SYSPROCPERM テーブル  
   パーミッション 412  
 SYSTABAUTH ビュー  
   パーミッション 413  
 SYSTABLEPERM テーブル  
   パーミッション 412  
 SYSTABLE テーブル  
   整合性 443  
 SYSUSERAUTH ビュー  
   パーミッション 413  
 SYSUSERLIST ビュー  
   パーミッション 413  
 SYSUSERPERMS ビュー  
   パーミッション 413  
 SYSUSERPERM テーブル  
   パーミッション 412  
 SYSVIEW ビュー  
   ビュー情報 235  
 SYS グループ 403

## T

TCP/IP  
   BroadcastListener 153  
   LDAP 通信パラメータ 159  
   サーバ設定 152  
   サーバのポート番号 166  
   テスト 627  
   トラブルシューティング 627  
   ファイアウォール経由の接続 155  
 TDS 通信パラメータ 168

## Telnet

TCP/IP のテスト 627

TEMP\_EXTRACT\_NULL\_AS\_EMPTY  
オプション 311TEMP\_EXTRACT\_NULL\_AS\_ZERO  
オプション 311TEMP 環境変数  
ディスク容量 118

TIMEOUT 通信パラメータ 168

TIME インデックス 269  
推奨される使用法 269  
他のインデックスとの比較 274  
短所 273  
長所 273  
追加インデックス 274TIME データ型  
ロードの最適化 353TMP 環境変数  
ディスク容量 118

TO 通信パラメータ 168

## U

UID 接続パラメータ 151  
Unconditional 接続パラメータ 151

UNC 接続パラメータ 151

unichar ASE データ型 365

Unicode  
文字セット 482Unicode 文字セット  
説明 487

Unilib ライブラリ 482

UNIQUE 制約 429

unitext ASE データ型 365

univarchar ASE データ型 365

UNIX 93  
ODBC サポート 93UPDATE 文  
ジョイン操作の使用 375  
使用 374

Userid 接続パラメータ 151

USER 特別値  
デフォルト 426

util\_db.ini ファイル 388

## V

VALUES オプション  
INSERT 文 330VARCHAR データ  
長さ 0 のセル 356

VERIFY 通信パラメータ 169

Visual Basic  
ODBC 設定 87VLDB  
管理 19

## W

WarehouseArchitect  
説明 172WD インデックス 266  
推奨される使用法 266

短所 268

長所 268

WHERE 句  
UPDATE 文 375  
インデックス選択への影響 250  
ジョイン・インデックス 277

Windows サービス 26

WITH GRANT OPTION 句 396

WORD インデックス 266

WORM 記憶領域 553

## X

XML フォーマット 301

## Z

zr ログ・ファイル 640

-Z オプション  
データベース・サーバ 119

## あ

- アクセシビリティ
  - マニュアル xxiv
- アドホック・ジョイン
  - パフォーマンス 274
- アロケーション・マップ
  - エラーの修復 573
  - 矛盾 577
  - リカバリ 576
  - 割り付けの検査 561
- 暗号化
  - Encryption 140
  - FIPS 414
  - RSA 414
  - カラム 413
  - 強力な暗号 140
  - 通信 168
  - データベース 413
- 暗号化パスワード
  - ODBC 設定 88

## い

- イーサネット 629
- 移行
  - 照合 482
  - ソート・キー 482
  - 廃止された照合 482
- 依存性
  - カーソル動作 477
- 一貫性の検査
  - マルチプレックス 559
- イベント
  - DBSpaceLogger 613
  - ディスク領域の使用状況のモニタリング 612
  - 領域の使用状況のモニタリング 613
- イベント・ハンドラ
  - ConnectFailed 385
- インタフェース・ライブラリ
  - 接続 62
- インデックス
  - FP(1) および FP(2) 254, 255
  - FP(1) インデックス 255
  - FP(2) インデックス 255
  - FP(3) インデックス 256
  - sp\_iqcheckdb エラー 573, 583
  - Sybase Central で作成 247
  - インデックス・タイプの選択 275
  - 概要 236
  - 検索 255
  - 検証 238
  - 検証と修復 567
  - 再構築 227
  - 削除 239
  - 作成 221, 246
  - システム・テーブル 237
  - 自動作成 226
  - 射影可能 FP 254
  - 修復 571
  - 修復不可能なエラー 582
  - 説明 241
  - 追加と削除 245
  - ディスク領域の使用率 251
  - テーブル上に多すぎる 618
  - テーブルのロード後の追加 276
  - デフォルト 254, 255
  - 名前の変更 239
  - 破損の削除 584
  - フラット FP 259
  - 並行作成 247
  - マルチカラム HG と NULL 263
  - 矛盾 571
  - ユニークな値の最大数 620
  - 論理的な問題の検出 583
- インデックス・タイプ
  - LF 259
  - 推奨 250
  - 説明 242
  - 選択 275
  - 選択の条件 248
- インポート、データ
  - Adaptive Server Enterprise からの挿入 332
  - LOAD TABLE 文 316
  - バージョン 12 より前の IQ データベース 333
  - 変換エラー 369

引用符  
SQL 内 404

## う

埋め込みデータベース 126, 130  
Java 78  
起動 78  
接続 78  
接続パラメータ 97

## え

エラー  
DB 領域の不足状況 608  
修復不可能 582  
挿入と削除 459  
データ変換 369  
トランザクション処理 459  
エラー・メッセージ  
PIPE\_NOT\_CONNECTED 318  
ファイルへのリダイレクト 304  
文字セット変換 488  
¥ 円記号  
Windows ロー・デバイス 197  
演算子  
作業 550  
エンティティ整合性  
確保 433  
説明 419

## お

オートコミット・モード  
ODBC 88  
大文字と小文字の区別  
コマンド・ライン 33  
接続パラメータ 125  
データベース名とサーバ名 36  
パスワード 195, 496

オブジェクト  
修飾名 404  
オプション  
DBCC\_LOG\_PROGRESS 525, 563  
MAX\_TEMP\_SPACE\_PER\_CONNECTION 203  
QUERY\_TEMP\_SPACE\_LIMIT 203  
設定 410  
オプション値  
トランケーション 306

## か

カーソル  
ODBC 設定 88  
依存性 477  
位置付け操作 478  
コマンドの構文 479  
スクロール 477  
接続の制限 410  
トランザクション 476  
ホールド 456, 478  
メッセージ・ロギング 479  
例 479  
開始 59  
開始行  
ODBC 設定 89  
外部キー  
オプション 439  
強制力のない 230  
強制力のない既存の外部キー 439  
作成 230  
制限 419  
データの挿入 340  
書き込みサーバ  
置き換え 563  
カスタム照合 482  
仮想バックアップ  
カプセル化仮想バックアップ 519  
分離仮想バックアップ 520  
カタログ・ストア  
キャッシュ・サイズの設定 39  
説明 179

## 索引

カタログ・ファイル  
  増加 611  
活性  
  ODBC 設定 90  
可変長データ転送 504  
カラム  
  削除 228  
  修復不可能なエラー 582  
  追加 228  
  デフォルト 420  
  プロパティ 424  
  変更 228  
カラム、ロード時に設定される 359  
カラム・デリミタ  
  ロード形式オプション 318  
カラムのデフォルトの使用 420  
カラムのデフォルトの変更と削除 422  
カラム幅  
  挿入の問題 352  
環境変数  
  LANG 494  
  LC\_ALL 494  
  SACHARSET 493  
  SALANG 493  
  SQLCONNECT 81  
関数  
  BFILE 334  
  BLOB 用 334  
  CLOB 用 334  
  タイプ 11  
  
**き**  
キー・ジョイン 286, 289  
起動  
  障害の解決 577  
  チェックポイント・エラー 577  
  トラブルシューティング 118  
  トラブルシューティングのヒント 592  
  割り付けエラー 577  
起動スクリプト 26  
起動パラメータ 32  
起動、マルチプレックス 29

起動ユーティリティ 26  
キャッシュ  
  書き込み 447  
キャッシュ・サイズ  
  カタログ・ストアの設定 39  
強制リカバリ 577  
  書き込みサーバの置き換え 563  
  サーバの起動に失敗 577  
  重複ブロックの検出 562  
  手順 580  
  複合的に所有されたブロックの検出 562  
  割り付けられていないブロックの検出 562  
強制力のない外部キー 439  
強力な暗号  
  Encryption 140

## <

クエリ  
  DATEPART を使用 270  
  スラッシング 618  
  同時実行性の規則 459  
  同時ユーザの制限 39  
  パフォーマンス 263  
  パフォーマンスの問題 618  
  範囲述部 271  
クエリ・サーバ  
  書き込みサーバの置き換え 563  
クエリのタイプ  
  インデックス・タイプ 249  
クライアント  
  直接のデータ・ロード 323  
クライアント側  
  Encryption 140  
クライアント・データバルク・ロード 323  
クライアント・プロセス情報 127  
グループ  
  PUBLIC 403  
  Sybase Central 400  
  SYS 403  
  管理 398  
  作成 399  
  パーミッション 382, 401



パスワードなし 402  
 メンバシップ 400  
 グローバル・テンポラリ・テーブル  
 説明 223

## け

権限  
 挿入と削除 302  
 データベース・オブジェクトの定義 176  
 言語サポート  
 概要 481  
 照合 492  
 説明 481  
 マルチバイト文字セット 487  
 現在の日付と時刻のデフォルト 425  
 検索インデックス 255

## こ

後続ブランク  
 削除 314, 319  
 後続ブランクの削除 314, 319  
 候補キー  
 制限 419  
 コード・ページ  
 サポート対象 484  
 説明 483  
 コマンド・ファイル  
 DBISQL 173  
 データベース・オブジェクトの作成 174  
 コマンド・ライン・スイッチ 32  
 必須 34  
 表示 33

## さ

サーバ  
 CPU 使用率 598  
 Windows での起動 595  
 応答なし 597, 598, 599  
 起動時の自動開始 26  
 起動に失敗 577  
 再利用 51  
 処理の停止 597  
 接続 68  
 停止の問題 600  
 「データベース・サーバ」参照  
 デッドロック 600  
 トランザクション・ログ 592, 593  
 命名 35  
 ユニークな名前 593  
 ユニークなポート番号 594  
 リカバリ 557, 590  
 領域の不足 597  
 サーバ・オブジェクト  
 追加 70  
 サーバ起動スイッチ  
 -iqmsgnum 14  
 -iqmsgsz 14  
 サーバの再利用 51  
 サーバ・プロパティ  
 IQMsgMaxSize 14  
 IQMsgNumFiles 14  
 サーバ名  
 ODBC 設定 88  
 サーバ・ログ・ファイル 50  
 サーベンス・オクスリー法 552  
 削除 392  
 カラムのデフォルト 422  
 ビュー 234  
 作成  
 カラムのデフォルト 421  
 サブトランザクション  
 セーブポイント 468  
 サポート・センタ  
 MySybase 647  
 オンライン・ヘルプ 647  
 チェックリスト 647  
 問題の報告 641  
 サポート・センタ用のチェックリスト 647  
 参照整合性  
 確保 433  
 カラムのデフォルト 420  
 強制力のない既存の外部キーによる適用 439

宣言 435  
 パーミッション 303  
 サンプル・データベース xxiv, 8

## し

## 時刻データ型

Adaptive Server Enterprise と Sybase IQ のデータの  
 対応 367  
 システム DB 領域 177  
 システム障害  
 リカバリ 601  
 システム・テーブル 9  
 SYSFILE 529  
 インデックス 237  
 説明 231  
 パーミッション 412  
 表示 204  
 ユーザとグループ 412  
 システムの無応答 597  
 システム・ビュー  
 整合性 443  
 パーミッション 412  
 システム・レベルのバックアップ 521  
 指定  
 ドライバ 69  
 射影可能 FP インデックス 254  
 シャットダウン  
 データベース 58  
 トラブルシューティング 118  
 修飾されたオブジェクト名 404  
 修復  
 インデックス 584  
 データベース 557, 590  
 テーブル 584  
 割り付け 571  
 集約 260  
 出力の分析 640  
 出力フォーマット  
 DBISQL 302  
 出力リダイレクション 304, 305  
 ジョイン  
 更新 375

スター 291  
 パフォーマンスの影響 274  
 複数テーブル 282  
 リニア 291  
 ジョイン・インデックス  
 Sybase Central で作成 290  
 カラムの変更 228  
 基礎となるテーブルの修正 295  
 候補キー 419  
 サイズの見積もり 297  
 作成 221, 283  
 ジョイン階層 278  
 ジョイン関係 286  
 説明 277  
 挿入 338  
 テーブル内のカラム 279  
 同期 285  
 トップ・テーブルと関連テーブルのロー数の  
 比率 298  
 ベース・テーブル 339  
 マルチプレックス 225  
 メリットの見積もり 298  
 ジョイン階層 278  
 ジョイン仮想テーブル 225  
 ジョイン・カラム 286  
 ジョイン関係  
 指定 288  
 定義 286  
 ジョイン・タイプ 290  
 障害  
 システム 497  
 メディア 497  
 照合  
 ANSI 486  
 CP874toUTF8 ユーティリティ 482  
 CREATE DATABASE 文 482  
 ISO\_1 486  
 SORT\_COLLATION オプション 481  
 SORTKEY 関数 481, 482  
 移行 482  
 カスタム 482  
 互換性の問題 496  
 このバージョンでの変更点 482

サポート対象 485  
 推奨 485  
 説明 481, 484, 485  
 選択 492  
 ソート 484  
 代替 485  
 データベースの作成 495, 496  
 デフォルト 483, 491, 495, 496  
 日本語での推奨 487  
 廃止 482  
 廃止された照合の移行 482  
 パフォーマンスの問題 496  
 表示 485  
 変更 495  
 マルチバイト 487  
 所有者  
   説明 380  
   役割 6  
 シングルノード・モード 225  
 シングルバイト文字セット  
   説明 483  
 診断ツール 630  
   sa\_server\_option 637  
   sp\_iqcheckdb 636  
   sp\_iqcheckoptions 636  
   sp\_iqconnection 637  
   sp\_iqcontext 637  
   sp\_iqdbstatistics 636  
   sp\_iqstatus 630  
   サーバ起動オプションの確認 636  
   サーバ要求のロギング 637  
   通信の問題 641  
   データベース・オプションの確認 636

## す

垂直挿入  
   説明 340  
 スキーマ  
   作成 221  
   変更 221

スキーマのロード  
   推奨するデータベース・サイズ 183  
 スクロール  
   カーソル 477  
 スター・ジョイン 291  
 スタック・トレース  
   スレッドに対する生成 599  
   ロケーション 601  
 ストアド・プロシージャ  
   Sybase IQ 8  
   説明 8  
 スナップショット・バージョン管理  
   「バージョン管理」参照 445  
 スラッシング  
   HASH\_THRASHING\_PERCENT  
     オプション 619  
 スレッド  
   スタック・トレースの生成 599  
   不足 614  
   ロード時の使用 373

## せ

整合性  
   概要 415  
   カラムのデフォルト 420  
   制約 418, 419  
 整合性制約違反  
   ログの例 326  
 製品サポート 641  
 製品マニュアル xxi  
 制約  
   パフォーマンスへの影響 417  
 制約違反  
   ログの例 326  
 セーブポイント  
   トランザクション内 468  
 セキュリティ  
   Advanced Security オプション 414  
   Encryption 140  
   FIPS サポート 414  
   IPv6 のサポート 414  
   Kerberos 認証 414

- RSA サポート 414
  - カラムの暗号化 413
  - 説明 377
  - データベースの暗号化 413
  - 統合化ログイン 113, 115
  - ビュー 406
  - プロシージャ 396, 406
  - ログインの失敗 385
  - 接続
    - BroadcastListener 153
    - iAnywhere JDBC ドライバ 69
    - Interactive SQL 71, 106
    - jConnect 69
    - JDBC 63
    - LDAP 通信パラメータ 159
    - 埋め込みデータベース 78
    - 外部ホスト上のデータベースへの接続 76
    - 概要 62
    - 確立 64
    - 管理 384
    - サーバによる確立方法 98
    - 最大値 383
    - 削除 117
    - 制限 528, 578
    - データ・ソースの使用 79
    - デフォルト・データベース 106
    - 同時ユーザの制限 38
    - トラブルシューティング 118
    - ファイアウォール 155
    - 文字セット 489
    - リモート 68
    - 例 71
    - ローカル・データベース 71
    - ロギング 117
  - 接続情報
    - IQ メッセージ・ファイル 646
  - 接続の削除 117
  - 接続のロギング 117
  - 接続パラメータ
    - CharSet 130
    - Encryption 140
    - Idle 143
    - Language 144
    - LazyClose 145
    - 位置 101
    - 埋め込みデータベース 97
    - 大文字と小文字を区別しない 125
    - 接続文字列 64
    - 説明 125
    - データ・ソース 82
    - デフォルト 80
    - 矛盾 97
    - 優先度 126
  - 接続ハンドル 13, 636
  - 接続プロファイル 70
  - 接続名
    - ODBC 設定 91
  - 接続文字列
    - 表現 64
    - 文字セット 489
  - 切断 117
  - 設定
    - ODBC データ・ソース 86
  - 設定パラメータ
    - 上書き 596
  - 設定ファイル
    - 使用 33
  - ゼロ
    - NULL への変換 361
- ## そ
- 挿入
    - Adaptive Server Enterprise データベースからの挿入 333
    - 概要 300
    - カラムのデフォルト 422
    - カラム幅の問題 352
    - 旧バージョンからの挿入 348
    - 指定したロー 331
    - ジョイン・インデックス・テーブル 338
    - 対話的 335
    - 「データのロード」参照 348
    - パフォーマンス 369
    - 部分幅挿入 340

- プライマリ・キー・カラムと外部キー・カラム 340
- 別のデータベースからの挿入 332
- 挿入変換オプション 348
- ソート
  - 照合 484
- ソート・キー
  - 移行 482
- ソート順
  - 照合 481
- ソフトウェアのリリース番号 12

## た

- 第 508 条
  - 法令遵守 xxiv

## ち

- チェックポイント
  - 間隔の調整 623
  - システム・リカバリ 472
  - 自動、明示的 468
  - 説明 467
  - リカバリ 467
- 中カッコ 334

## つ

- 通信
  - Encryption 140
  - トラブルシューティング 624
  - パラメータ 152
- 通信パラメータ
  - Broadcast 153
  - BroadcastListener 153
  - Certificate 154
  - Certificate\_Password 154
  - ClientPort 155
  - DatabaseName 156
  - DLL 158
  - DOBROADCAST 156

- HOST 158
- LDAP 159
- LocalOnly 160
- LogMaxSize 162
- LogOptions 162
- MaxConnections 163
- MYIP 164
- PreFetchOnOpen 165
- ReceiveBufferSize 166
- SendBufferSize 166
- ServerPort 166
- SESSIONS 168
- TDS 168
- TIMEOUT 168, 169
- 説明 125, 164
- 通知カウント 247
- 通知メッセージ 632

## て

- 停止
  - トラブルシューティング 600
- 定数式のデフォルト 428
- ディスク
  - 領域の使用状況のモニタリング 612
  - 領域の不足 597, 607
- ディスク・アレイ
  - WORM 553
- ディスク容量
  - インデックス 251
  - 節約 477
  - 割り付け 213
- ディスク領域の不足
  - 対処法 597, 607
  - 領域の使用状況のモニタリング 612
- 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 300
  - エクスポート 300, 303
  - クライアント 323
  - 削除 375
  - 重複 416
  - 抽出 305

- トランザクション 454
- 入出力フォーマット 301
- 無効 416
- ロード 300
- データ型
  - Adaptive Server Enterprise と Sybase IQ データ型の  
変換 363
  - Adaptive Server Enterprise の unichar 365
  - Adaptive Server Enterprise の unitext 365
  - Adaptive Server Enterprise の univarchar 365
  - FLOAT 365
  - integer 364
  - money 366, 368
  - REAL 365
  - Sybase IQ と Adaptive Server Enterprise の対応 363
  - テーブル作成での指定 223
  - 変換 346
  - 文字 365
  - ロード時の変換 348
- データ整合性
  - 概要 415
  - カラムのデフォルト 420
  - システム・テーブル内のルール 443
  - 制約 419
- データ操作言語 189
- データ・ソース
  - jConnect の使用 69
  - ODBC 82
  - UNIX 93
  - コマンド・ラインから作成 85
  - 接続 79
  - 設定 86
  - 説明 82
- データ・ソースの説明
  - ODBC 87
- データ・ソース名
  - ODBC 86
- データ抽出
  - ASCII 309
  - アクセスの制御 308
  - オプション 305
  - オプション・リスト 305
  - 説明 305
- バイナリ 309
- バイナリ／スワップ 309
- データ定義
  - 作成 221
  - 同時実行性の規則 461
- データ定義言語
  - 説明 171
- データ・トランケーション・エラー 605
- データのエクスポート
  - 概要 300
  - 説明 303
- データの修正
  - パーミッション 302
- データの抽出
  - オプション 305
  - オプション・リスト 305
  - 説明 305
- データのロード
  - ASCII 変換オプション 354
  - ASE データ 334
  - エラー 620
  - 概要 300
  - 形式オプション 318
  - 整合性制約違反 326
  - 制約違反のロギング 326
  - 「挿入」参照 348
  - 通知メッセージ 632
  - 同時実行性の規則 459
  - 名前付きパイプ 318
  - パフォーマンス 369, 618
  - 必要な権限 302
  - ファイルの指定 317
  - 変換エラー 369
  - 変換オプション 348
  - ラージ・オブジェクト 334
  - 領域の使用状況のモニタリング 612
- データベース
  - Adaptive Server Anywhere 179
  - Interactive SQL による管理 173
  - アンロード 56
  - 一貫性のチェック 525
  - 移動 193
  - 埋め込み 126, 130

- オブジェクトの作成 221
- オブジェクトの使用 171
- 開始 56
- 管理タスク 3
- 起動時の命名 35
- 起動パーミッション 58
- 検証 525
- サイズ 198
- 削除 206
- 作成 191
- 作成と削除のためのパーミッション 388
- 作成に必要な権限 176
- サンプル xxiv
- 修復 557, 590
- 初期化 192
- 初期サイズ 183
- 所有者の役割 6
- スキーマの作成 221
- セキュリティの概要 6
- 設計 172
- 接続 62, 68, 98
- 設定の概要 174
- 相対パス名 196
- 大容量 19
- 停止 58
- データベース一貫性チェッカ (DBCC) 525
- デフォルトの特性 194
- デモ xxiv
- パーミッション 7, 41, 377
- ファイルの移動 532
- ファイル・ロケーション 194
- ブロック・サイズ 201
- ページ・サイズ 198
- 命名 35
- 文字セット 488
- ユーティリティ 19
- ユーティリティ・データベースを使用した  
作成 387
- 領域の事前割り付け 177
- ロケーションの選択 196
- データベース管理者  
「DBA」参照 378
- 定義 378
- データベース・サーバ
  - Windows サービス 32
  - Windows 上での起動 32
  - 起動 25
  - 起動時の命名 36
  - 起動の回避 129
  - 緊急時の停止 122
  - コマンド・ライン・スイッチ 32
  - 実行、コマンド・プロンプト 26
  - [スタート]メニューからの起動 32
  - 接続 68
  - 説明 4
  - 停止 51, 55, 151
  - 名前スイッチ 35
  - リモート 68
- データベースのリストア
  - ファイル名の変更 532
- データベース・ファイル
  - ODBC 設定 89
- データベース・ファイル名の変更 532
- データベース名
  - ODBC 設定 89
- データ変換エラー 605
- データ・リンク・レイヤ
  - トラブルシューティング 626
- テープ・デバイス
  - バックアップ用 510
- テーブル
  - 新しい DB 領域への移動 381
  - キーの追加 229
  - グループ所有者 402
  - 削除 229
  - 作成 221
  - 「システム・テーブル」の情報参照 9
  - 修飾名 402, 404
  - 修復不可能なエラー 582
  - ジョイン関係 286
  - 所有者 380
  - パーミッション 380
  - 破損 584
  - 複数テーブルのジョイン 282
  - ブロックされたアクセス 620

- ブロックされたアクセスの管理 621
- 変更 228
- 読み込み専用 208
- ロード 316
- ロック 459
- テーブル・パーティション 187
- テーブル領域
  - 割り付け 4
- テーブル・レベルのバージョン管理
  - 説明 450
  - 「バージョン管理」参照 450
- デッドロック
  - 解決 600
  - 検出 600
- デバイス・タイプ
  - データベース 177
- デフォルト
  - AUTOINCREMENT 426
  - NEWID 427
  - NULL 428
  - Sybase Central で作成 424
  - USER 特別値 426
  - カラム 420
  - 現在の日付と時刻 425
  - 作成 421
  - 接続パラメータ 80
  - 挿入 422
  - 定数式 428
  - 文字列と数値 428
  - ロード 422
- デフォルト・インデックス
  - 説明 254
- デフォルトの設定ファイル 33
- デフォルトのログイン・ポリシー 383, 384
- デモ・データベース xxiv
- デリミタ
  - SELECT 文 334
- テンポラリ記憶領域
  - 領域を節約するオプション 477
- テンポラリ・ストア
  - 領域管理 202
- テンポラリ・テーブル
  - 説明 223

- バージョン管理 457
- ロード 224

## と

- 同期
  - 説明 285
- 統合化ログイン
  - ODBC 設定 88
  - オペレーティング・システム 108
  - 使用 112
  - デフォルト・ユーザ 116
  - ネットワーク面 115
- 同時実行性
  - Sybase IQ 448
  - 挿入、削除、クエリ 459
  - データ定義 461
  - バックアップ 449, 507
  - 読み取りと書き込み 451
- 動的結果セット 605
- 独立性レベル 465
  - ODBC 設定 87
- トップ・テーブル
  - サイズとパフォーマンス 283
- ドライバ
  - iAnywhere JDBC ドライバ 69
  - jConnect JDBC ドライバ 69
- ドライバに起因するエラー
  - ODBC 設定 87
- ドライバの選択
  - iAnywhere JDBC ドライバの使用 69
  - jConnect JDBC ドライバの使用 69
- トラブルシューティング 589
  - DBISQL 605
  - DBISQLC 605
  - Sybase Central 624
  - 一般的な問題 628
  - 起動、シャットダウン、接続 118
  - サーバの運用 592
  - サーバの起動 119
  - 処理の問題 618
  - データベース接続 98, 602



- 配線の問題 628
- プロトコル 624
- リソースの問題 607
- トランザクション
  - カーソル 476
  - 開始 447
  - サブトランザクションとセーブポイント 468
  - 終了 447
  - セーブポイント 468
  - 説明 445
  - 定義 446
  - リカバリ 471, 558
  - ロールバック 470
- トランザクション処理
  - 説明 445
- トランザクションのコミット
  - 読み取りトランザクションのタイミングの影響 455
- トランザクション・ログ
  - システム・リカバリ 472
  - 使用しないリカバリ 581
  - 名前の変更 540
- トレース
  - スレッドに対する生成 599

## な

- 内部ビルド番号 12
- ナチュラル・ジョイン 290
- 夏時間
  - SQL ログ・ファイル 18
  - メッセージ・ログ・ファイル 18
- 名前付きパイプ 318
  - 抽出 315

## に

- 日時インデックス
  - 「DTTM インデックス」参照

## ね

- ネットワーク・アダプタ
  - ドライバ 625
- ネットワーク通信
  - 起動のトラブルシューティング 119
  - コマンド・ライン・スイッチ 152
- ネットワーク番号
  - IPX アドレス 156
- ネットワーク・プロトコル
  - ODBC 設定 90
  - トラブルシューティング 624

## は

- バージョン管理
  - カーソル 477
  - システム・リカバリ 472
  - 説明 445, 450
  - テーブル・レベル 450
  - テンポラリ・テーブル 457
  - 独立性レベル 465
  - パフォーマンスの影響 474
  - リカバリ 558
- バージョン文字列 12
- パーティション
  - DDL 操作 188
  - 最大数 188
  - 読み込み専用 208
- パーティション・キー 187
- パーミッション
  - connect 390
  - DBA 権限 378
  - DB 領域管理 381
  - RESOURCE 権限 380, 392
  - Sybase Central 394, 397
  - WITH GRANT OPTION 396
  - 概要 377
  - 管理 377
  - グループ 382, 398, 401
  - グループ・メンバシップ 400
  - 継承 396, 398
  - 個別 390

- コマンド・ライン・スイッチ 41
- タイプ 7
- テーブル 381, 393
- パスワード 391
- パスワードの付与 390
- ビュー 234, 393
- ビューでの INSERT と DELETE 408
- 付与権 396
- プロシージャ 396
- 矛盾 409
- リスト 412
- 配線
  - トラブルシューティング 628
- バイナリ
  - データ抽出 309
- ハイパースレッディング
  - サーバ・スイッチ 40
- パイプ
  - 名前付き 315
- バグ
  - レポート 641
- パス名
  - データベース 196
- パスワード
  - ODBC 設定 88
  - 大文字と小文字の区別 195, 496
  - 確認 392
  - デフォルト 378
  - 変更 391
  - 有効期間 383
  - 有効期間の設定 384
  - ユーティリティ・データベース 388
  - ルール 392
  - 最小の長さ 392
- パスワードの確認 392
- バックアップ
  - .iqmsg ファイル 523
  - iqmsg ファイル 17
  - Linux 504
  - NULL 518
  - SAN を使用した仮想バックアップ 521
  - アテンド付き 505
  - アテンドなし 505, 543
  - インクリメンタル 548
  - エラーからのリカバリ 516
  - 仮想 509, 518
  - 高速 521
  - サード・パーティ 517
  - システム・レベル 521
  - 使用メモリの増加 551
  - スケジューリング 548
  - 責任 550
  - 説明 497
  - 待機時間 514
  - 対象データ 501
  - テープ・デバイスの指定 (NT の場合) 511
  - デバイス 503, 509
  - 同時実行性 449
  - 同時実行性の問題 507
  - パフォーマンスの問題 550
  - 必要な権限 504
  - フル 548
  - ヘッダ・ファイルの表示 541
  - マルチプレックス 498
  - メッセージ・ログ 17, 523
  - メッセージ・ログ・アーカイブ 17, 523
  - 読み取り専用ハードウェア 514
- バックアップ・ログ
  - 説明 544
  - ロケーション 545
- バッファ・キャッシュ
  - IQ UTILITIES コマンド 599
  - モニタ 599
  - 領域の不足 597
- バッファ・サイズ
  - ODBC 設定 90
- パフォーマンス
  - アドホック・ジョイン 274
  - インデックス 236
  - クエリとロード 618
  - 制約の影響 417
  - 挿入 331
  - データのロード 369
  - バージョン管理の影響 474
  - フラット・ファイルからのロード 349

バルク・コピー 301  
 バルク・ロード 323  
 範囲分割 188  
   定義 186

## ひ

比較インデックス  
   「CMP インデックス」参照  
 日付データ型  
   Adaptive Server Enterprise と Sybase IQ のデータ  
   の対応 367  
 ビュー  
   SELECT 文の制限 233  
   永久テーブルとの相違点 232  
   削除 234, 235  
   作成 232  
   使用 233  
   所有者 380  
   セキュリティ 406  
   挿入と削除 233  
   パーミッション 234, 380, 381  
   変更 234  
   編集 231  
 表記規則  
   構文 xxiii  
   書体 xxiv  
   マニュアル xxiii, xxiv  
 標準  
   第 508 条への準拠 xxiv  
 ビルド番号 12

## ふ

ファイアウォール  
   BroadcastListener 153  
   LDAP 通信パラメータ 159  
   接続 155  
 ファイル  
   出力のリダイレクト 304, 305  
 ファイル・サイズ  
   制御 611

フェッチ操作  
   警告の非表示 87  
 複数のユーザ  
   Sybase Central で作成 391  
   グループに追加 401  
   個別に作成 390  
   ログインの失敗 385  
   ロック 384  
   ロックアウト 386  
   ロック解除 386  
 複数レコードのフェッチ  
   ODBC 設定 91  
 物理レイヤ  
   トラブルシューティング 628  
 部分幅挿入  
   START ROW ID オプション 335  
   規則 342  
   説明 340  
   例 343  
 プライマリ・キー  
   AUTOINCREMENT 426  
   エンティティ整合性 434  
   作成 229  
   データの挿入 340  
   マルチカラム 435  
 フラット FP インデックス 259  
 フラット・ファイル  
   ロード変換オプション 348  
   ロード元 316  
 ブランク  
   NULL への変換 361  
   後続ブランクの削除 314, 319  
 フレーム・タイプ 629  
 プロシージャ  
   作成パーミッション 380  
   システム 8  
   所有者 380  
   「ストアド・プロシージャ」参照 8  
   セキュリティ 406  
   動的結果セット 605  
   パーミッション 396  
 ブロック・サイズ 201

## 索引

ブロックされた書き込みアクセス  
  競合の管理 621  
  ブロックしている書き込みの確認 620  
ブロックマップ 561  
ブロック・モード 504  
プロトコル  
  スイッチ 44  
  トラブルシューティング 624  
プロファイル  
  接続 70  
分割  
  定義 186  
分割されたテーブル  
  APPEND\_LOAD オプション 323  
  INSERT 323  
  LOAD TABLE 323

## へ

並列 CREATE INDEX 247  
ページ・サイズ 198  
  カタログ 43  
  スイッチ 43  
ベース・テーブル 223  
  ジョイン・インデックス内 339  
変換  
  Adaptive Server Enterprise と Sybase IQ 間 363  
  インポート時のエラー 369  
  挿入オプション 348  
  挿入時 346  
変換オプション  
  DATE 356  
  DATETIME 358, 359  
  DATE フォーマットの指定 356  
  長さ 0 のセルの置換 356  
  パフォーマンス 353  
  フラット・ファイルのロード 348  
変換ドライバ  
  ODBC 設定 87  
変更  
  カラムのデフォルト 422  
  照合 495

## ほ

包含インデックス  
  「WD インデックス」参照  
法令遵守  
  第 508 条 xxiv  
ポート番号  
  UNIX での指定 76  
  データベース・サーバ 166  
  デフォルト 76  
ホールド・カーソル 456, 478

## ま

マニュアル  
  CD xxi  
  SQL Anywhere xx  
  Sybase IQ xviii  
  アクセシビリティ機能 xxiv  
  オンライン xxi  
  表記規則 xxiii, xxiv  
マルチスレッド  
  ロード時 373  
マルチバイト文字セット  
  使用 487  
マルチプレックス  
  一貫性の検査 559  
  ジョイン・インデックスの作成 225  
マルチプレックス・サーバ  
  停止 52  
  ロード・バランス 169  
マルチプレックス・データベース  
  検証 526  
  バックアップ 498  
  リストア 524  
マルチプレックスの同期 225  
マルチプロセッサ・マシン  
  スイッチ 37

## む

矛盾した状態 577

矛盾するインデックス  
修復 571

## め

メイン・ストア  
領域管理 202  
メタデータ  
カタログ・ストア内 179  
メッセージ  
DB 領域の不足状況 608  
ファイルへのリダイレクト 304  
メッセージ・ログに記録 11  
メモリの通知 633  
メッセージ・ファイル  
接続情報 646  
メッセージ・ロギング 11, 14  
メッセージ・ログ 11  
IQ\_SYSTEM\_MSG DB 領域 181  
アーカイブのバックアップ 17, 523  
夏時間 18  
バックアップ 17, 523  
メッセージ・ログ管理 11, 14  
メモリ  
カタログ・ストア・キャッシュ 39  
接続の制限 410  
連結メモリのプールの作成 39  
メモリ・メッセージ  
ロード通知メッセージ 633

## も

文字セット  
Unicode 482, 487  
コード化 481, 483  
シングルバイト 483  
接続パラメータ 130  
説明 481  
選択 492  
変換 495  
変換の回避 490  
マルチバイト 487

文字セット変換  
エラー・メッセージ 488  
オンとオフ 495  
サーバの起動 495  
無効化 495  
文字データ型  
Adaptive Server Enterprise と Sybase IQ のデータ  
の対応 365  
Adaptive Server Enterprise の unichar 365  
Adaptive Server Enterprise の unitext 365  
Adaptive Server Enterprise の univarchar 365  
文字列と数値のデフォルト 428  
問題  
レポート 641

## ゆ

ユーザ 377  
ユーザ ID  
ODBC 設定 88  
管理 377  
削除 397  
作成 390  
デフォルト 378  
メッセージ・ログ 13  
リスト 412  
ユーザ管理  
「Sybase IQ ユーザ管理」参照  
ユーザ定義データ型  
CHECK 条件 431  
ユーザ名  
ユーティリティ  
CP874toUTF8 482  
トランザクション・ログ 540  
ユーティリティ・データベース  
起動 387  
セキュリティ 387  
接続 389  
説明 19  
データベース作成のためのパスワード 388  
パスワードの設定 388

## 索引

ユーティリティ・プログラム  
  廃止された iq\_bcp 323  
ユーロ記号  
  1252LATIN1 照合 486, 487

## よ

要求ログ・ファイル 640  
  sa\_get\_request\_profile の使用 640  
  sa\_get\_request\_times の使用 640  
要求ログ・レベル 646  
読み取り専用ハードウェア  
  例 553  
読み取り専用 DB 領域 207  
読み取り専用ハードウェア  
  バックアップ 514

## ら

ラージ・オブジェクト管理 334

## り

リーク領域リカバリ 579  
リカバリ  
  書き込みサーバの置き換え 563  
  強制 577  
  サーバ 557, 590  
  システム 471, 557, 558  
  システム障害 601  
  通常 558  
  データベースの検証 559  
  データベースの修復 559  
  特殊なモード 576  
  トランザクション 471, 558  
  トランザクション・ログ 472  
  トランザクション・ログの紛失 581  
  バージョン管理 558  
  バックアップの紛失 581  
  リーク領域 579  
リストア操作

エラーからのリカバリ 542  
説明 526  
正しい順序の確認 537  
他のユーザの除外 541  
パフォーマンスの問題 550  
ヘッダ・ファイルの表示 541  
リストア後の SYSFILE 529  
ロー・デバイスへ 533  
リソース計画  
  iqnumbercpus スイッチ 40  
リダイレクト  
  ファイルへの出力 304, 305  
リモート・データ  
  バルク・ロード 323  
  ロード 323  
リモート・データ・アクセス  
  プロキシ・テーブル 333  
領域管理  
  DB 領域の不足状況 202, 607  
  IQ テンポラリ・ストア 202  
  IQ メイン・ストア 202  
  領域待ちの状況 202, 607

## る

ルータ  
  ブロードキャスト 156

## れ

連結メモリ  
  iqwmem スイッチの設定 39  
連邦リハビリテーション法  
  第 508 条 xxiv

## ろ

ロー ID  
  通知メッセージ・ログ 342  
  表示 342

- ローカル・テンポラリ・テーブル  
説明 223
  - ロー・デバイス 177
    - Windows 上での名前の指定 197
    - ユーティリティ・データベース 387
    - リストア 533
  - ロード
    - カラムのデフォルト 422
    - クライアント・データ 323
    - スレッド使用 373
  - ロード・オプション 321
  - ロードの最適化 353
  - ロード・パフォーマンス
    - ロック競合 464
  - 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 202, 607
  - ログイン
    - 制限 384
    - 統合化 108, 109
  - ログイン試行
    - 制限の超過 386
  - ログインの失敗 385
  - ログイン・ポリシー 383
    - リセット 386
  - ログイン・ポリシー・オプション
    - ロック 386
  - ログイン・ポリシーのリセット 386
  - ログ・ファイル
    - 接続情報の関連付け 646
    - サーバ 50
  - ロケール
    - INSERT...LOCATION 494
    - 決定 493
    - 設定 493
    - 説明 481, 485
    - 文字セット 488
  - ロック
    - DDL 操作 461
    - 競合の管理 621
    - テーブル 459
  - ロックアウト
    - 自動 385
  - ロック競合
    - 管理 464
- わ
- 割り付け
    - DBCC 修復の出力 575
    - 検証と修復 571

