



Sybase IQ の概要

---

## Sybase IQ 15.4

ドキュメント ID：DC37422-01-1540-01

改訂：2011年11月

Copyright © 2011 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、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告な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るものであり、無断で使用することはできません。

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 およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

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

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

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

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

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

# 目次

<b>Sybase IQ について .....</b>	<b>1</b>
関連製品 .....	3
マニュアル .....	4
アーキテクチャ .....	6
利点 .....	6
コンポーネント .....	7
ツールとユーティリティ .....	7
Windows オプション .....	8
データベース・サーバ .....	9
デモ・データベース .....	9
<b>概念 .....</b>	<b>13</b>
サーバ .....	13
データベース・コンピューティング .....	13
Sybase IQ データベース・ファイル .....	14
各要素の関係 .....	16
SQL とデータベース・コンピューティング .....	16
リレーショナル・データベース .....	16
データベース・テーブル .....	17
プライマリ・キーと外部キー .....	18
その他のデータベース・オブジェクト .....	19
クエリ .....	20
その他の SQL 文 .....	21
システム・テーブル .....	21
<b>サーバの起動と接続 .....</b>	<b>23</b>
Windows クライアントの接続 .....	23
UNIX クライアントの接続 .....	24
Sybase Central とデータベース管理 .....	25
Sybase Central インタフェース .....	25

Sybase IQ Agent の実行 .....	26
Sybase Central メイン・ウィンドウのナビゲーション .....	28
データベース・スキーマの表示 .....	28
その他のデータベース・オブジェクトの表示 .....	30
Interactive SQL を使用したデータベースへの接続 .....	32
Sybase Central から Interactive SQL を起動する .....	32
コマンド・ラインから Interactive SQL を起動する .....	32
Windows での Interactive SQL の起動 .....	33
データベースへのコマンドの送信 .....	34
データベース・サーバの停止 .....	34
<b>データの選択 .....</b>	<b>35</b>
テーブル情報の表示 .....	35
クエリ結果の順序付け .....	37
カラムとローの選択 .....	38
検索条件の使用 .....	39
クエリでの日付の比較 .....	40
WHERE 句での複合検索条件 .....	41
検索条件でのパターン・マッチング .....	41
発音によるローのマッチング .....	42
検索条件のショートカットの使用 .....	43
集約データの取得 .....	43
集合関数によるグループ化されたデータの取得 .....	44
グループの制限 .....	45
小計計算の活用 .....	46
ROLLUP .....	46
Cube .....	48
分析データの取得 .....	49
ランク付け統計関数 .....	49

逆分散統計関数 .....	50
重複したローの削除 .....	51
<b>テーブルのジョイン .....</b>	<b>53</b>
外積ジョイン .....	53
ジョインの制限 .....	53
テーブル関係 .....	55
ジョイン演算子 .....	55
キー・ジョイン .....	56
ナチュラル・ジョイン .....	57
アドホック・ジョインとジョイン・インデックスの 使用 .....	58
ジョインとデータ型 .....	59
ストア間またはデータベース間ジョインのサポート ...	60
リモート・データベースと異種データベースのクエ リ .....	61
リモート・データベースのジョイン .....	61
ジョインとサブクエリ .....	62
<b>データベースの管理 .....</b>	<b>65</b>
テーブルの管理 .....	65
Office テーブルの作成 .....	65
office テーブルのカラムの作成 .....	66
カラム・プロパティの編集 .....	67
Office テーブルのデモ・データベースからの削 除 .....	67
プロシージャの管理 .....	68
ストアド・プロシージャの表示 .....	68
ユーザとグループの管理 .....	69
新しいログイン・ポリシーの追加 .....	70
データベースへのグループの追加 .....	70
デモ・データベースへのユーザの追加 .....	70
デモ・データベースの元の状態へのリストア .....	71
ユーザ接続の管理 .....	71

インデックス機能とデータのロード .....	73
カラム・インデックスの作成 .....	73
カラム・インデックスのタイプ .....	74
カラムへの Low Fast インデックスの追加 .....	75
ジョイン・インデックスの作成 .....	76
SalesOrders と SalesOrderItems のジョイン・	
インデックスの作成 .....	77
Department と Employee のジョイン・イン	
デックスの作成 .....	77
ジョイン・インデックス・データの更新 .....	78
データのロード .....	78
<b>DB 領域の管理 .....</b>	<b>81</b>
DB 領域の作成 .....	81
DB 領域の変更 .....	83
<b>データベースの更新 .....</b>	<b>85</b>
テーブルへのローの追加 .....	85
永続的な変更 .....	86
データの一貫性の確保 .....	87
セーブポイント .....	87
データ一貫性のテスト .....	88
セーブポイントの設定 .....	88
複数のアカウントによる複数のテーブルの更新 .....	89
ローの削除 .....	90
ローの更新 .....	90
ジョイン・インデックスの同期 .....	91
DB 領域、テーブル、インデックスの削除 .....	91
<b>ビューの使用 .....</b>	<b>93</b>
ビューの定義 .....	93
ビューの作成 .....	94
ビュー・データの表示 .....	94
ビューの変更 .....	94
ビューの削除 .....	95

ビューの制限と高度な使用 .....	95
セキュリティを目的とするビューの使用 .....	96
データベース内の情報へのアクセスの制限 .....	98
<b>用語解説 .....</b>	<b>99</b>
<b>索引 .....</b>	<b>103</b>

# 目次

## Sybase IQ について

Sybase® IQ は、データ・ウェアハウス専用に設計された、高いパフォーマンスの意思決定支援サーバです。

Sybase IQ は、Sybase 製品ファミリに属します。Sybase 製品ファミリにはこの他に、Adaptive Server Enterprise と SQL Anywhere があります。Sybase IQ 内のコンポーネント統合サービスは、メインフレーム、UNIX、または Windows サーバ上のリレーショナル・データベースまたは非リレーショナル・データベースへの直接アクセスを提供します。

### 製品エディション

Sybase IQ には、以下の製品エディションが用意されています。

製品エディション	説明
Enterprise Edition	すべてのオプションと機能がサポートされます。 オプションの機能を使用するには、追加のライセンスを別途購入する必要があります。
Small Business Edition	Enterprise Edition で利用できるほとんどの機能がサポートされますが、マルチプレックスまたはオプション機能はサポートされません。
Single Application Server Edition	Small Business Edition と同じ機能を利用できますが、1 台または 2 台のマシン環境に制限されます。
Evaluation Edition	Enterprise Edition で利用できるすべてのオプションと機能が、30 日間の評価期間でサポートされます。
Express Edition	評価版のすべてのオプションと機能をサポートしています。 評価版とは異なり、Express Edition には有効期限がありませんが、IQ メイン DB 領域の合計サイズが制限されます。また、その他のライセンス・オプションを追加したり、運用環境に導入したりすることはできません。

### オプション機能

オプション機能によって、Enterprise Edition のユーティリティを拡張できます。

オプション	説明
非構造化データ分析	<p>バイナリ・ラージ・オブジェクト (BLOB: Binary Large Object) とキャラクタ・ラージ・オブジェクト (CLOB: Character Large Object) の格納と取得がサポートされます。</p> <p>バイナリ・ラージ・オブジェクト (BLOB: Binary Large Object) とキャラクタ・ラージ・オブジェクト (CLOB: Character Large Object) の格納と取得がサポートされます。</p>
Advanced Security	<p>次のオプションのセキュリティ機能がサポートされます。</p> <ul style="list-style-type: none"> <li>• カラムの暗号化</li> <li>• FIPS のネットワーク暗号化</li> <li>• Kerberos 接続認証</li> </ul>
マルチプレックス・グリッド	<p>マルチプレックス環境にノードを追加できます。</p> <p>セカンダリ・マルチプレックス・ノード (リーダー/ライター) を起動するには、Sybase IQ に追加のライセンスが必要です。</p>
Very Large Database Management	<p>複数のテーブル領域と DB 領域を追加し、管理しやすいサブセットにデータを論理的に分割できます。</p> <p>IQ サーバが 2 つ以上の IQ ユーザ DB 領域を作成する場合、または 2 つ以上の IQ ユーザ DB 領域から起動する場合、Sybase IQ には追加ライセンスが必要です。</p>
Express Edition	<p>評価版のすべてのオプションと機能をサポートしています。</p> <p>評価版とは異なり、Express Edition には有効期限がありませんが、IQ メイン DB 領域の合計サイズが制限されます。また、その他のライセンス・オプションを追加したり、運用環境に導入したりすることはできません。</p>

### ライセンス

Evaluation Edition 以外のすべての製品エディションにライセンスが必要です。オプション機能は別売です。また、ライセンスが別途必要です。

Evaluation Edition は、ライセンスされていないサーバです。Enterprise Edition のすべての機能とオプションにアクセスできます。ただし、30 日間の評価期間以降に Evaluation Edition を実行するには、適切なライセンスを購入してインストールする必要があります。

## 関連製品

---

Sybase IQ は、完全な分析インフラストラクチャを提供する、多様な Sybase 製品群の一部です。

- SQL Anywhere® – SQL Anywhere は、サーバ環境、デスクトップ環境、モバイル環境、リモート・オフィス環境用の包括的なデータ管理パッケージです。Sybase IQ は、SQL Anywhere の拡張版です。このため、Sybase IQ には、SQL Anywhere の機能が多く取り込まれています。  
詳細については、Sybase Web サイトの [製品] > [データベース管理] > [SQL Anywhere] を参照してください。
- Sybase RAP – The Trading Edition® は、投資管理部門用の分析プラットフォームです。Sybase RAP が基盤とする技術の一部は、Adaptive Server Enterprise と Sybase IQ によって実現されています。  
詳細については、Sybase Web サイトの [製品] > [分析 & ビジネス・インテリジェンス] > [RAP - The Trading Edition] を参照してください。
- Sybase PowerDesigner® – PowerDesigner は、データ・アーキテクチャ、情報アーキテクチャ、エンタープライズ・アーキテクチャに対応する、業界最高レベルのモデリングとメタデータ管理のソリューションです。  
詳細については、Sybase Web サイトの [製品] > [モデリング & 開発] > [PowerDesigner] を参照してください。
- Sybase IQ InfoPrimer – Sybase IQ InfoPrimer は、データの抽出、変換、ロードのためのプラットフォームです。迅速に効率よく、データを複数のソースから Sybase IQ のレポート環境に移動できます。  
詳細については、Sybase Web サイトの [製品] > [全製品リスト] > [Sybase ETL] を参照してください。
- Sybase WorkSpace – WorkSpace によって、最も重要な設計開発ツールが、使いやすいオープン・ソースのフレームワークに組み込まれます。  
詳細については、Sybase Web サイトの [製品] > [モデリング & 開発] > [WorkSpace] を参照してください。

## マニュアル

---

この概要を参照して、特定のトピックに関する情報を入手してください。

### 最新リリース情報

マニュアル名	コンテンツ
『インストールと設定』	インストールと設定の手順。
『リリース・ノート』	最新の製品情報。
『オプションライセンスガイド』	別途購入できる機能。
『新機能の概要』	リリース固有機能の概要。

### はじめに

マニュアル名	コンテンツ
『Sybase IQ の概要』	Sybase IQ の実習。
『Sybase IQ クイック・スタート』	IQ デモ・データベースの作成とクエリの手順。
『ユーティリティ・ガイド』	コマンド・ライン・ユーティリティのリファレンス。
『Sybase IQ Multiplex の使用』	マルチプレックスの設定と管理。
『パフォーマンス&チューニング・シリーズ』	データベース、システム、およびクエリのチューニング・オプション。

### 参照

マニュアル名	コンテンツ
『システム管理ガイド：第1巻』	データベースの設定と管理。
『システム管理ガイド：第2巻』	データベース・アプリケーションの開発。
『リファレンス：文とオプション』	SQL 構文、パラメータ、オプション。
『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』	SQL 文、関数、プロシージャ。
『エラー・メッセージ』	エラー・メッセージと警告メッセージ (Web のみ)。

## オプション機能

マニュアル名	コンテンツ
『非構造化データ分析』	バイナリ・ラージ・オブジェクト (BLOB: Binary Large Object) とキャラクタ・ラージ・オブジェクト (CLOB: Character Large Object) の格納と取得。
『ユーザ定義関数ガイド』	ユーザ定義関数の C/C++ インタフェース
『Sybase IQ Multiplex の使用』	マルチプレックスの設定と管理。
『高度なセキュリティ』	Sybase IQ の高度なセキュリティ・オプション。
『IMSL 数値関数ライブラリ・ユーザ・ガイド：第2巻 C 統計ライブラリ』	IMSL C 統計ライブラリ関数。
『時系列ガイド』	時系列予測／分析関数

## SQL Anywhere のリファレンス

マニュアル名	コンテンツ
『SQL Anywhere サーバ - データベース管理』	Sybase IQ のマニュアルで参照されている、その他の管理の詳細。
『SQL Anywhere サーバ - プログラミング』	API プログラミングのマニュアルと例。
『SQL Anywhere サーバ - SQL リファレンス』	Sybase IQ のマニュアルで参照されている、その他の SQL 機能。

## ライセンス

マニュアル名	コンテンツ
『Sybase ソフトウェア資産管理 (SySAM) 2』	ライセンスの生成、オプション、管理。
『SySAM 2 クイック・スタート・ガイド』	SPDC ライセンス生成。
『FLEXnet ライセンス・エンド・ユーザ・ガイド』	FLEXnet ライセンス・ユーティリティ。

## アーキテクチャ

---

Sybase IQ のアーキテクチャは、一般的なりレーショナル・データベースとは異なっています。Sybase IQ は、ライタではなくリーダを重視しています。このため、多くのユーザに高速なクエリ応答が提供されます。

- ローではなく、カラムへのデータの格納
- すべてのカラムにインデックスを指定することによるパフォーマンスの向上
- 大きなページ・サイズの設定によるパフォーマンスの向上
- 大きなテンポラリ・キャッシュの設定によるほとんどの操作でのパフォーマンスの向上
- データへのアクセスのテーブル・レベルでの発生
- ほとんどのクエリ結果は、テーブル・レベルでデータにフォーカスする
- ほとんどの挿入と削除での、単一のローではなくテーブル全体に対して行われるデータの書き込み

Sybase IQ の設定と最適なパフォーマンスについては、『システム管理ガイド：第1巻』、『システム管理ガイド：第2巻』、および『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。

## 利点

---

Sybase IQ は、ミッション・クリティカルなビジネス・ソリューションにおいて優れたパフォーマンスを発揮できるように最適化された、意思決定支援システムです。

- インデックスのみの実行プランを使用して、任意のタイプのクエリを処理するインテリジェントなクエリ処理。
- ユニプロセッサ・システムと並列システム上でのアドホック・クエリのパフォーマンス。
- マルチサーバ設定で、大量のクエリの負荷を管理するマルチプレックス機能。
- 非常に柔軟なスキーマ・サポート。
- ほとんどの環境において、クエリ固有の調整を必要としない効率的な実行が可能。
- 高速な初期およびインクリメンタル・ロード。
- 高速なデータの集約、カウント、比較。
- マルチユーザ環境用に最適化された並列処理。
- ストアド・プロシージャ。
- クエリ時間の短縮による生産性の向上。
- 生データよりも小さな領域にデータベース全体とインデックスを格納。

- 入出力 (I/O) の削減。

## コンポーネント

---

Sybase IQ には、サーバの操作に役立つツールとユーティリティがあります。

### ツールとユーティリティ

Sybase IQ には、日常の管理タスクの実行に役立つツールとユーティリティがあります。

表 1 : Sybase IQ ユーティリティ

ユーティリティ	説明
Sybase Control Center	Sybase Control Center (SCC) は、Sybase 製品の Web ベースの管理コンソールです。企業内に Sybase Control Center を導入するには、1 つ以上の SCC サーバが必要です。 詳細については、Control Center のマニュアルを参照してください。
Sybase Central	Sybase Central は、Sybase 製品の管理コンソールです。Sybase IQ のタスクを管理するには、Sybase IQ プラグインが必要です。 このマニュアルのチュートリアルと手順では、ほとんどの場合に Sybase Central を使用します。
Interactive SQL	Interactive SQL は、Sybase IQ に同梱されているツールです。このツールを使用すると、SQL 文の実行、スクリプトの作成、データベース・データの表示ができます。 Interactive SQL 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。
コマンド・ライン・ユーティリティ	コマンド・ライン・ユーティリティを使用して、データベース管理タスクを実行できます。 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。

**注意：** 利用可能なユーティリティは、使用するオペレーティング・システムとインストール・オプションによって異なります。

## Windows オプション

Windows では、Sybase IQ プログラム・グループを使用することで、共通ユーティリティにすばやくアクセスできます。Sybase IQ サーバと Sybase IQ クライアントでは、インストール時に利用できるオプションが異なります。

サーバのインストール

表 2 : Sybase IQ サーバ・オプション

オプション	説明
Interactive SQL	データベースへの SQL 文送信に使用する、Java ベースの Interactive SQL ユーティリティを起動します。
ODBC アドミニストレータ	32 ビットまたは 64 ビット・プラットフォーム用データベースへの ODBC 接続を管理します。
ロー・デバイス・アクセス	レジストリにユーザとロー・デバイスを追加します。または、レジストリからユーザとロー・デバイスを削除します。
サンプル・アプリケーションとサンプル・プロジェクト	サンプル・アプリケーションとプロジェクトのリストを表示します。  これらのサンプル実行の詳細については、 <a href="http://dcx.sybase.com/index.html#1201/en/dbprogramming/dbprogramming12.html">http://dcx.sybase.com/index.html#1201/en/dbprogramming/dbprogramming12.html</a> 『SQL Anywhere サーバ - プログラミング』を参照してください。
Sybase IQ デモ・データベースの起動	デモ・データベースを実行するデータベース・サーバを起動します。
Sybase Central	Sybase Central を起動します。
Sybase IQ ReadMe ファイル	マニュアルとソフトウェアに加えられた最新の変更内容、および重要事項が掲載されています。
Sybase サービス・マネージャ	Sybase IQ を Windows サービスとして設定、変更、または削除できます。
Web 上の Sybase	Sybase オンライン・リソースを紹介している Web ページを開きます。

## クライアント・インストール

表 3 : Sybase IQ クライアント・オプション

オプション	説明
サンプル・アプリケーションとサンプル・プロジェクト	サンプル・アプリケーションとプロジェクトのリストを表示します。 これらのサンプル実行の詳細については、 <a href="http://dcx.sybase.com/index.html#1201/en/dbprogramming/dbprogramming12.html">http://dcx.sybase.com/index.html#1201/en/dbprogramming/dbprogramming12.html</a> 『SQL Anywhere サーバ - プログラミング』を参照してください。
Web 上の Sybase	Sybase オンライン・リソースを紹介している Web ページを開きます。
ODBC アドミニストレータ	32 ビットまたは 64 ビット・プラットフォーム用データベースへの ODBC 接続を管理します。
Interactive SQL	データベースへの SQL 文送信に使用する、Java ベースの Interactive SQL ユーティリティを起動します。

データベース・サーバ

Sybase IQ ネットワーク・サーバは、ネットワークと複数のユーザ間のクライアント／サーバ通信をサポートします。

サーバを起動するには、オペレーティング・システムに適した起動ユーティリティを実行します。

- UNIX または Linux オペレーティング・システムでは、start\_iq を使用します。
- Windows では、start\_iq.exe を使用します。

デモ・データベース

このマニュアルでは、多くの例で Sybase IQ デモ・データベース (iqdemo.db) をデータ・ソースとして使用しています。デモ・データベースは Sybase IQ Server Suite の一部としてインストールされ、サーバに格納されます。クライアント・アクセスは、Sybase IQ Client Suite の一部としてインストールされるツールによって行うことができます。

### テーブル名とテーブル所有者

デモ・データベースには、データ・テーブルとユーティリティ・テーブルが用意されています。

デモ・データベースは、所有権によって区分されています。このマニュアル内にある SQL の例では、ほとんどの場合、GROUPO テーブルへのアクセスが最低条件として必須です。

#### GROUPO テーブル

GROUPO テーブルには、スポーツ衣料品を販売する架空の企業に関する内部情報が格納されています。サンプル・データには、この企業の内部情報 (従業員、部署、財務データ) とともに、製品情報 (製品) や販売情報 (受注、顧客、連絡先) が入っています。

表 4 : GroupO テーブル名

名前	説明
Contacts	顧客担当者および営業担当者。
Customers	顧客名および住所。
Departments	マネージャや名前などの部署情報。
Employees	名前、給与、勤務地などの従業員情報。
FinancialCodes	財務コードを持つ支出と収入の各項目。
FinancialData	会社の四半期ごとの財務情報。
Products	価格や在庫量などの製品情報。
SalesOrderItems	販売指示項目。各注文は、1 つ以上の項目で構成されます。注文項目に関する情報は、個別のテーブルに格納されます。
SalesOrders	Customer ID、OrderDate、FinancialCode、Region、SalesRepresentative のある個別の販売指示。

#### DBA テーブル

DBA が所有しているテーブルには、ユーティリティとサンプル・データ・テーブルが含まれています。DBA テーブルにアクセスするには、DBA 権限が必要です。

表 5 : DBA テーブル名

テーブル	説明
iq_dummy	<p>iq_dummy は、単一ロー、単一カラムのユーティリティ・テーブルです。このテーブルを使用してデータベースから情報を抽出します。たとえば、次のように <b>NOW()</b> 関数を iq_dummy に対して実行すると、現在の日付と時刻が返されます。</p> <pre>SELECT NOW() FROM iq_dummy</pre> <p>Sybase IQ の DUMMY システム・テーブルは、FROM 句がないすべてのクエリで使用されます。詳細については、『システム管理ガイド：第 1 巻』の「パフォーマンス・モニタリング用のダミー・テーブル」を参照してください。</p>
empl	<p>サンプルの従業員テーブルには、dept_id、start_date、name、salary のカラムがあります。</p>
sale	<p>サンプルの販売テーブルには、prod_id、month_num、rep_id、および sales のカラムがあります。</p>

#### SYSOPTION - DEFAULTS テーブル

SYSOPTIONDEFAULTS は、DBO が所有するユーティリティ・テーブルです。このテーブルには、Sybase IQ のすべてのオプション名と値が格納されています。このテーブルにクエリを実行すると、すべてのオプションのデフォルト値を参照できます。

**注意：** デモ・データベースは、大文字と小文字を区別しません。つまり、比較や文字列の操作で、大文字と小文字の差異は考慮されません。たとえば、デモ・データベースを使用する場合、ユーザ ID とパスワードの入力は、大文字でも小文字でもかまいません。ユーザが作成する Sybase IQ データベースは、デモ・データベースとは異なり、デフォルトで大文字と小文字を区別することに注意してください。



## 概念

いくつかの基本的な用語と概念を理解しておくこと、Sybase IQ を操作するときに役立ちます。

## サーバ

---

データベースには、データベース用の通信チャンネルと操作デバイスを提供するサーバを通じてアクセスしてください。

Sybase IQ サーバは、データベースを動的に起動および停止できます。さらに、同じマシン上で実行しているアプリケーションやユーザ、または別のコンピュータ上で実行しているアプリケーションやユーザからも 2 台のマシンをつなぐネットワークを介して、接続を受け入れることができます。ユーザは、サーバに対してではなく 1 つのデータベースに対する接続権限を持ちます。サーバごとに 1 つのデータベースのみを管理することを強くおすすめします。

複数の Sybase IQ サーバを使用して、Sybase IQ、SQL Anywhere、Enterprise の各データベースを組み合わせ使用できます。複数の Adaptive Server Enterprise サーバを使用して、Sybase IQ データベースと Adaptive Server Enterprise データベースを組み合わせ使用することもできます。

Sybase IQ マルチプレックスは、共有ディスク・アレイに分散された Sybase IQ データベースです。マルチプレックスの各サーバが Sybase IQ を実行します。1 つの Sybase IQ テンポラリ・ストアと 1 つのカatalog・ストアを組み合わせ 1 台のサーバを構成し、サーバ間で共通の Sybase IQ ストアを共有します。クエリの負荷が高い状態で共有ディスク・アレイを使用しているユーザは、Sybase IQ マルチプレックスを作成することによって、CPU 処理能力の向上、メモリ領域の増加を実現できます。1 つの Sybase IQ マルチプレックスで、共有データベースに対して複雑なクエリを実行している多くのユーザをサポートできます。

すべての Sybase IQ サーバを、Sybase Central で管理できます。

## データベース・コンピューティング

---

データベース・アプリケーションとデータベース・サーバが連携して、データベースを管理します。

すべての情報システムは、次の要素で構成されています。

## 概念

- データベース - データはデータベースに格納されます。
- データベース・サーバ - データベースを管理します。その他のアプリケーションがデータベース・ファイルを直接操作することはできません。すべてのアプリケーションはデータベース・サーバと通信します。
- 言語インタフェース - アプリケーションはインタフェースを使用してデータベース・サーバと通信します。ODBC、JDBC、Sybase Open Client、または Embedded SQL を使用できます。

言語インタフェースは、クライアント・アプリケーションがデータベースと通信するとき使用できる一連の関数呼び出しを提供します。ODBC および JDBC の場合、ライブラリは一般にドライバと呼ばれます。インタフェースは、通常、UNIX オペレーティング・システムでは共有ライブラリとして提供され、Windows オペレーティング・システムではダイナミック・リンク・ライブラリ (DLL) として提供されます。JDBC インタフェースは、コンパイル済みの Java クラスの zip ファイルである Sybase jConnect ドライバを使用します。

Sybase IQ ネットワーク・サーバを操作している場合は、言語インタフェースはクライアント・コンピュータ上にあります。

- クライアント・アプリケーション - 言語インタフェースの 1 つを使用して、データベース・サーバと通信します。  
アプリケーションの開発に使用する RAD (Rapid Application Development) ツール (Sybase Enterprise Application Studio™ ツールの 1 つなど) には、データベース・サーバと通信するための独自の方法があり、言語インタフェースの詳細は隠されている場合があります。その場合でも、すべてのアプリケーションはサポートされている言語インタフェースの 1 つを使用します。

## Sybase IQ データベース・ファイル

データベースを作成すると、Sybase IQ によって多くのファイルが作成されます。

表 6: データベース・ファイル

デフォルトの物理ファイル名	内部名	説明	作成される数
dbname.db	該当せず	SQL Anywhere DB 領域ファイル。カタログ・ストアの一部。	各サーバに 1 つ以上
dbname.log	該当せず	SQL Anywhere トランザクション・ログ。カタログ・ストアの一部。	サーバごとに 1 つ
dbname.iq	IQ_SYSTEM_MAIN	IQ メイン・ストア用の最初の dbfile(追加ファイルの名前はユーザが指定)。	データベースごとに 1 つ以上

デフォルトの物理ファイル名	内部名	説明	作成される数
dbname.iqtmp	IQ_SYSTEM_TEMP	最初の IQ テンポラリ dbfile。	データベースごとに1つ以上
dbname.iqmsg	IQ_SYSTEM_MSG	判読可能な、デバッグ出力の IQ トレース・ファイル。	データベースごとに1つ
dbname.lmp	該当せず	判読可能な IQ ライセンス・ファイル。	データベースごとに1つ

**注意：** DBA は、ここに示されているデフォルトのファイル名と拡張子を上書きできます。

データベースの作成時には、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できます。ディレクトリ・パスの指定を省略すると、Sybase IQ は次に示すようにファイルを作成します。

- カタログ・ストアが、サーバの作業ディレクトリを基準に作成されます。
- IQ ストアが、サーバの作業ディレクトリを基準に作成されます。
- テンポラリ・ストアが、カタログ・ストアと同じディレクトリに作成されます (ファイル名を指定しない場合にも作成されます)。
- メッセージ・ログが、カタログ・ストアと同じディレクトリに作成されます (ファイル名を指定しない場合にも作成されます)。
- トランザクション・ログが、カタログ・ストアと同じディレクトリに作成されます (ファイル名を指定しない場合にも作成されます)。

**注意：** 最適なパフォーマンスを得るには、トランザクション・ログを、カタログ・ストアと IQ ストアとは別の物理デバイスに配置してください。これは、カタログ・ストアと IQ ストアがランダム・アクセス・ファイルであるのに対して、トランザクション・ファイルはシーケンシャル・アクセス・ファイルであるためです。

メイン・ストアとテンポラリ・ストアには、ほとんどのテーブル・データが格納されています。各データベースには、テンポラリ・データ用の一意なファイルがあります。ファイルを指定しない場合は、一時記憶領域として使用されるファイルが Sybase IQ によって作成されます。

## 各要素の関係

クライアント/サーバ環境にある完全なアプリケーションは、データベース・サーバと1つまたは複数のクライアント・アプリケーションで構成されます。

データベース・サーバは、ネットワーク通信をサポートするネットワーク・サーバの場合があります。クライアント/サーバ環境で動作するために、クライアント・アプリケーションを変更する必要はありません。

## SQL とデータベース・コンピューティング

クライアント・アプリケーションは Structured Query Language (SQL) 文を使用して、情報の取得やテーブルへの行の挿入などのデータベース・タスクを実行します。

クライアント・アプリケーションの開発方法に応じて、SQL 文は、プログラミング言語からの関数呼び出しで指定したり、アプリケーション開発ツールで提供される特別なウィンドウ内で、グラフィックに作成したりできます。

SQL 文は、プログラミング・インタフェースによってデータベース・サーバに伝達され、データベース・サーバによって実行された後、アプリケーションに結果が返されます。

クライアント/サーバ通信プロトコルは、クライアント・アプリケーションとデータベース・サーバの間で情報の伝達を行います。プログラミング・インタフェースは、アプリケーションが情報を送信する方法を定義します。使用するインタフェースやネットワーク・プロトコルに関係なく、サーバに送信されるのは SQL 文であり、クライアント・アプリケーションに返されるのは SQL 文の結果です。

---

**注意：** Sybase IQ、SQL Anywhere、および Adaptive Server Enterprise によってサポートされている SQL (Structured Query Language) 構文は少しだけ異なります。SQL 互換性の概要については、『リファレンス』の「ビルディング・ブロック、テーブル、およびプロシージャ」の「他の Sybase データベースとの互換性」を参照してください。

---

## リレーショナル・データベース

リレーショナル・データベース管理システムは、テーブル形式のデータを格納および取得するシステムです。リレーショナル・データベースは、相互関係を持つデータを格納するテーブルの集まりで構成されています。

## データベース・テーブル

リレーショナル・データベースでは、すべてのデータは、ローとカラムで構成されるテーブルに格納されます。

各テーブルは1つ以上のカラムを持ち、各カラムには、整数、文字の並び(テキスト用)、日付など、特定のデータ型が割り当てられます。テーブル内の各ローは、カラムごとに最大で1つの値を持ちます。特定のローとカラムに値がない場合は、その値を NULL と呼ぶことがあります。これは、現在は不明、適用外、または使用不可と解釈されます。

従業員情報が格納されているテーブルの例を、次に示します。

emp_ID	emp_lname	emp_fname	emp_phone
10057	Huong	Zhang	1096
10693	Donaldson	Anne	7821

### リレーショナル・テーブルの特性

リレーショナル・データベースのテーブルには、いくつかの重要な特性があります。

- カラムまたはローの順序による論理的な影響はありません。ただし、一部の特殊なクエリはカラムの順序による影響を与えます。同様にローの順序も、ローを返すいくつかのクエリの順序に影響を与え、このためクエリの実行のパフォーマンスに影響することもあります。ローの順序が大事である場合は、ローを返す順序をクエリで指定してください。ある特定の順序の使用頻度が高く、それ以外の順序の使用頻度が低い場合は、ローをその特定の順序に保つことが最善の策です。
- 各カラムの各ローには、値が含まれていないか (NULL カラム)、または値を1つだけ含みます。
- 特定のカラムの各値はすべて同じ型です。

次の表は、テーブルとその内容を表す、いくつかの公式および略式のリレーショナル・データベース用語と、非リレーショナル・データベースでその用語に相当する用語を示します。このマニュアルでは、略式の用語を使用しています。

公式のリレーショナル用語	略式のリレーショナル用語	対応する非リレーショナル用語
リレーション	テーブル	ファイル
属性	カラム	フィールド
タプル	Row	レコード

### 各テーブルに格納する情報

データベースを設計する場合は、データベース内の各テーブルに、従業員、製品、顧客など、特定の項目に関する情報が格納してあることを確認してください。

リレーショナル・データベースは、関連しないテーブルの集まりではありません。プライマリ・キーと外部キーを使用して、異なるテーブル内の情報間の関係を記述できます。

## プライマリ・キーと外部キー

プライマリ・キーと外部キーによって、データベース・テーブル内の各ローを識別し、テーブル間の関係を定義できます。これらのキーは、データベースの関係構造を定義します。

### プライマリ・キー

リレーショナル・データベース内の各テーブルには、プライマリ・キーがある場合とない場合があります。プライマリ・キーは1つのカラムまたはカラムの集まりであり、テーブル内の各ローを一意に識別できます。2つのローが、同じ値のプライマリ・キーを持つことはできません。

使用可能なすべてのカラムから、プライマリ・キーを選択できます。各テーブルのプライマリ・キーの長さをできるだけ短くしておいてください。可能であれば、プライマリ・キーの値を、char や varchar ではなく符号なし整数型にしてください。

たとえば、デモ・データベースの SalesOrderItems テーブルには次のカラムがあります。

- 顧客の注文を識別する ID カラム。
- 販売指示の各アイテムの識別番号がある LineID カラム。
- 顧客が発注した製品を識別する ProductID カラム。
- 発注されたアイテム数を示す Quantity カラム。
- 注文が出荷された日付を識別する ShipDate カラム。

特定のアイテムを識別するには、ID と行の LineID の両方が必要です。プライマリ・キーは、この両方のカラムで構成されます。

### 外部キー

あるテーブル内の情報は、外部キーによって別のテーブル内の情報に関連付けられます。

たとえば、デモ・データベースには、従業員情報が格納してあるテーブルと、部署情報が格納してあるテーブルがあります。Departments テーブルには、次の3つのカラムがあります。

- DepartmentID – 部署を識別する ID 番号。これがテーブルのプライマリ・キーになります。
- DepartmentName – 部署の名前が格納されているカラム。
- DepartmentHeadID – 部長の従業員 ID。

特定の従業員の所属部署名を探せるように、その従業員の部署名を Employees テーブルに入力しておく必要はありません。代わりに、Employees テーブルには、従業員の部署 ID を格納するカラムがあります。これを、Departments テーブルへの外部キーと呼びます。外部キーは、対応するプライマリ・キーまたは一意性制約のある、テーブル内の特定のローを参照します。このように参照されるプライマリ・キーと一意性制約は、候補キーと呼ばれます。

この例の場合、Employee テーブル (関係の外部キーを格納) は、外部テーブルまたは参照元テーブルと呼ばれます。Department テーブル (参照されるプライマリ・キーを格納) は、プライマリ・テーブルまたは参照先テーブルと呼ばれます。

## その他のデータベース・オブジェクト

リレーショナル・データベースには、関連テーブルの集まりが複数格納されています。

リレーショナル・データベースを構成するオブジェクトには、次のものがあります。

オブジェクト	説明
インデックス	インデックスにより、情報をすばやく検索できます。データベースの場合、インデックスは、インデックスが付けられた各カラム値を、インデックス値を含むデータのローが格納される物理位置に関連付けます。 インデックスは、高いパフォーマンスを実現する上での重要な設計要素です。
Java オブジェクト	Java クラスをデータベースのカatalog・ストアにインストールできます。Java クラスによって、データベースに論理を構築する手段が提供されます。 Sybase IQ 内での Java データ型のサポートの詳細については、『システム管理ガイド：第 2 巻』を参照してください。

オブジェクト	説明
プロシージャとファンクション	<p>これらは、データベース自体に格納され、データベース内の情報を操作するルーチンです。</p> <p>独自のストアド・プロシージャを作成して名前を付けると、特定のデータベース・クエリを実行したり、他のデータベース・タスクを実行したりすることができます。ストアド・プロシージャにはパラメータを指定できます。たとえば、プロシージャ・コールの中で、ある金額をパラメータとして指定し、その金額を超える支払いをしたすべての顧客の名前を返すストアド・プロシージャを作成できます。</p>
ユーザとグループ	<p>データベースの各ユーザは、ユーザ ID とパスワードを持ちます。各ユーザにパーミッションを設定することで、機密情報を保護できます。パーミッションをより簡単に管理できるように、ユーザをグループに割り当てることができます。</p>
ビュー	<p>ビューは計算テーブル、または仮想テーブルです。クライアント・アプリケーションにとってはテーブルに似ていますが、データは格納されていません。代わりに、アクセスされるたびに、基本となるテーブルからビュー内の情報が計算されます。</p> <p>実際に情報を格納するテーブルを、ビューと区別してベース・テーブルと呼ぶことがあります。</p>

## クエリ

リレーショナル・システムにおける基本のクエリ演算は、選択、射影、ジョインです。SELECT 文でこのすべてを実現できます。

### 射影と制限

射影は、テーブルのカラムのサブセットです。制限(選択とも呼ばれる)は、いくつかの条件に基づいた、テーブル内のローのサブセットです。

たとえば、次の **SELECT** 文は、15 ドルを超えるすべての製品の名前と価格を取得します。

```
SELECT Name, UniPrice
FROM Products
WHERE UnitPrice > 15
```

このクエリは、制限 (**WHERE** unit\_price > 15) と射影 (**SELECT** name, unit\_price) の両方を使用します。

### ジョイン

ジョインは、キー・カラム内の値を比較して一致する値を持つローを返すことで、複数のテーブルのローをリンクします。たとえば、このクエリは、

SalesOrderItems、Products、Employees、SalesOrdersの各テーブルをジョインし、売上総数と収入を計算します。

```
SELECT Products.ID AS 'Product Code',
       Products.Name AS Item,
       Products.Description AS Style,
       SUM(SalesOrderItems.Quantity) AS Sales,
       Products.UnitPrice,
       SUM(SalesOrderItems.Quantity * Products.UnitPrice)
       AS Revenue FROM Employees
JOIN SalesOrders ON SalesOrders.SalesRepresentative =
  Employees.EmployeeID
JOIN SalesOrderItems ON SalesOrderItems.ID =
  SalesOrders.ID
JOIN Products ON Products.ID =
  SalesOrderItems.ProductID
GROUP BY Products.ID, Products.Description,
  Products.Name, Products.UnitPrice
ORDER BY Sales
```

## その他の SQL 文

SQLでは、クエリだけでなく、さまざまな操作を実行できます。SQLは、テーブル、ビュー、その他のデータベース・オブジェクトを作成する文を備えています。また、テーブルを修正する文(insert文とdelete文)や、このマニュアルで説明するその他の多くのデータベース・タスクを実行するコマンドも備えています。

## システム・テーブル

すべてのデータベースには、システムがデータとシステムを管理するために使用する特別なテーブルである、システム・テーブルの集まりが含まれています。これらのテーブルを、データ辞書またはシステム・カタログと呼ぶことがあります。Sybase IQでは、これらはカタログ・ストアに格納されています。

システム・テーブルには、データベースに関する情報が格納されています。システム・テーブル以外のテーブルは直接変更できますが、それと同じようにシステム・テーブルを変更しないでください。システム・テーブルには、データベース内のテーブル、データベースのユーザ、各テーブル内のカラムなどに関する情報が格納されます。この情報は、データに関するデータ、つまりメタデータです。データを挿入する前に、システム・テーブルを使用して各種カラムのサイズとデータ型を確認できます。

概念

## サーバの起動と接続

チュートリアルでの作業では、Sybase Central にアクセスし、デモ・データベースにクライアント接続する必要があります。クライアント接続の確立に Sybase Central を使用するかどうかは、お使いのクライアント・ツールが Windows 用か UNIX 用かによって決まります。

### Windows クライアントの接続

Windows 上でサンプル・クエリを実行するには、Sybase Central ネットワーク・クライアントを起動し、デモ・データベースに接続します。

#### 前提条件

DBA またはシステム管理者に、次のことを依頼してください。

- Sybase IQ Client Suite をワークステーションにインストールする。
- デモ・データベースを作成および起動する。デモ・データベースは、Sybase IQ Server Suite の一部としてインストールされます。iqdemo.db を起動する手順については、『インストールおよび設定ガイド』を参照してください。
- デモ・データベースにアクセスできるアカウントを作成する。サンプル・クエリを再作成するには、GROUPO テーブルへのアクセス権限を持つアカウントが必要です。

#### 手順

1. [スタート] > [プログラム] > [Sybase] > [Sybase Central] をクリックします。
2. 必要に応じて、[Tips] および [Welcome] ウィンドウを閉じます。
3. [Connections] > [Connect with Sybase IQ 15] をクリックします。
4. [Connection] ダイアログで次のように入力します。

フィールド	値
[Authentication]	[Database]
[User ID]	DBA または dba (大文字と小文字の区別なし)
[Password]	sql (大文字と小文字の区別あり)
[Action]	[Connect to a running database on another computer]

5. [Server Name] ボックスで、[Find] をクリックします。
6. [Servers] ダイアログから、[<localhost>\_iqdemo] を選択します。  
このマニュアルでは、プレースホルダ <localhost> を使用してホストを指定します。手順でプレースホルダが使用されているときは、代わりに該当するホスト名に置き換えてください。
7. [Connect] をクリックします。

## UNIX クライアントの接続

---

UNIX 系のオペレーティング・システム上でサンプル・クエリを実行するには、Sybase Central ネットワーク・クライアントを起動し、デモ・データベースに接続します。

### 前提条件

DBA またはシステム管理者に、次のことを依頼してください。

- Sybase IQ Client Suite をワークステーションにインストールする。環境設定が正しいことを確認してください。
- デモ・データベースを作成および起動する。デモ・データベースは、Sybase IQ Server Suite の一部としてインストールされます。iqdemo.db を起動する手順については、『インストールおよび設定ガイド』を参照してください。
- デモ・データベースにアクセスできるアカウントを作成する。サンプル・クエリを再作成するには、GROUPO テーブルへのアクセス権限を持つアカウントが必要です。

### 手順

1. コンソールから、\$IQDIR15/bin64 に移動し、scjview と入力します。
2. 必要に応じて、[Tips] および [Welcome] ウィンドウを閉じます。
3. [Connections] > [Connect with Sybase IQ 15] をクリックします。
4. [Connection] ダイアログで次のように入力します。

フィールド	値
[Authentication]	[Database]
[User ID]	DBA または dba (大文字と小文字の区別なし)
[Password]	sql (大文字と小文字の区別あり)

フィールド	値
[Action]	[Connect to a running database on another computer]

5. [Server Name] ボックスで、[Find] をクリックします。
6. [Servers] ダイアログから、[<localhost>\_iqdemo] を選択します。  
このマニュアルでは、プレースホルダ **<localhost>** を使用してホストを指定します。手順でプレースホルダが使用されているときは、代わりに該当するホスト名に置き換えてください。
7. [Connect] をクリックします。

## Sybase Central とデータベース管理

Sybase Central は、Sybase IQ と一緒にインストールされるデータベース管理ツールです。Sybase Central を使用して、標準のデータベース管理タスクをグラフィカルなユーザ・インタフェースから実行できます。

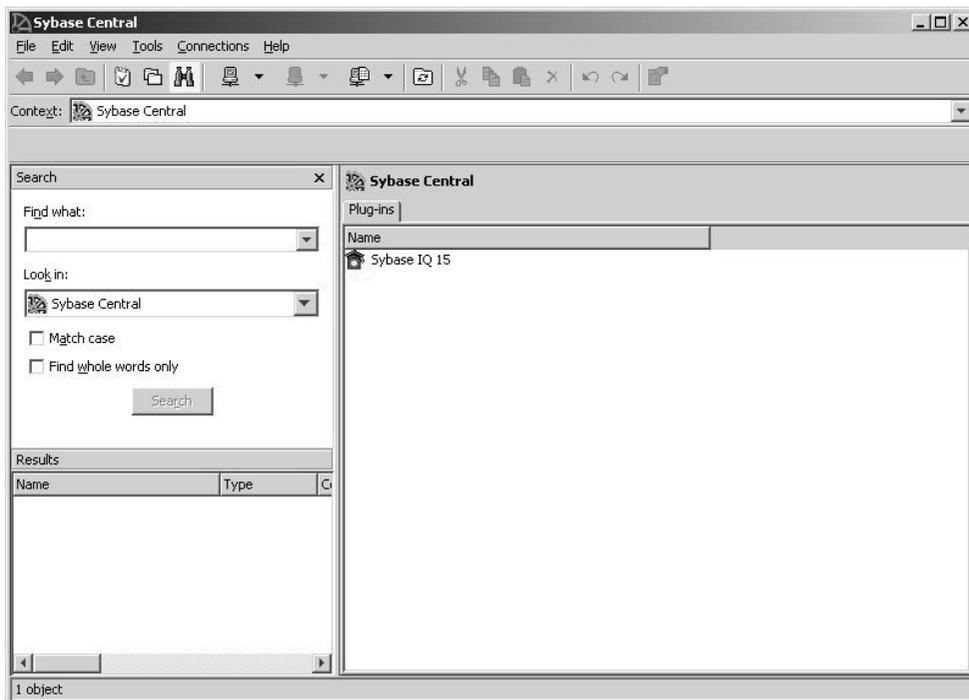
データベース管理タスクは、一般に、次の2つのカテゴリに分類されます。

- SQL 文をデータベース・エンジンに送信することで実行されるタスク。
- Sybase IQ ユーティリティによって実行されるタスク

Interactive SQL を使用すると、Sybase Central で実行できるタスクと同じデータ定義のタスクを、多数実行することもできます。ただし、マルチプレックス・データベースやクエリ・サーバの作成のように、Sybase Central と連携することで最も効果を発揮するタスクもあります。

## Sybase Central インタフェース

Sybase Central のメイン・ウィンドウは、左右に並んだ2つのウィンドウ枠に分割されています。左側のウィンドウ枠には、タスク、フォルダ、または検索オプションが表示されます。右側のウィンドウ枠には、オブジェクトに関連付けられている追加オプション、アクション、またはビュー、あるいは選択したタスクが表示されます。ツールバーまたは [ビュー] メニューのオプションを使用すると、左側のウィンドウ枠の表示を変更できます。



### プラグインのロード

Sybase 製品をインストールすると、そのプラグインもインストールされます。Sybase IQ のプラグインは自動的にロードされます。対象のプラグインが表示されない場合、プラグインを手動でロードできます。

1. [ツール] - [プラグイン] を選択します。
2. [Sybase IQ] を選択し、[Register] をクリックします。

Sybase IQ が一覧に表示されていない場合、[ロード]>[参照] をクリックし、*\$IQDIR15/java/iq.jpr* を選択します。

### Sybase Central の停止

Sybase Central を停止すると、メイン・ウィンドウが閉じます。Sybase Central を停止するには、[ファイル] をクリックし、[終了] を選択します。

## Sybase IQ Agent の実行

Sybase Central 上でのマルチプレックス操作(クエリ・サーバの作成など)の実行の多くは、Sybase IQ Agent が実行されている必要があります。Sybase IQ Agent は、通常、Sybase IQ がインストールされている Windows マシンをリブートすると起動

されます。マルチプレックス・データベースを管理する前に、Sybase IQ Agent が実行されていることを確認してください。

### Windows 上での Sybase IQ Agent の確認

サービス・コンソール上で、Sybase IQ Agent のステータスを確認します。

1. [コントロールパネル] で、[管理ツール]-[サービス] を順に選択します。
2. サービス・コンソール上で、Sybase IQ Agent のステータスを確認します。
3. [ステータス] が [起動済み] でない場合は、エージェントを右クリックし、[起動] を選択します。

### UNIX 上での Sybase IQ Agent の確認

`-agent` のオプションを使用して、Sybase IQ Agent のステータスを確認します。

1. `-agent` のオプションを指定して、`stop_iq` ユーティリティを実行します。

```
stop_iq -agent
```

2. 自分のユーザ名が所有者である実行中の Agent がない場合、`$IQDIR15/bin<platform>` に変更し、`S99SybaseIQAgent15` を実行します。

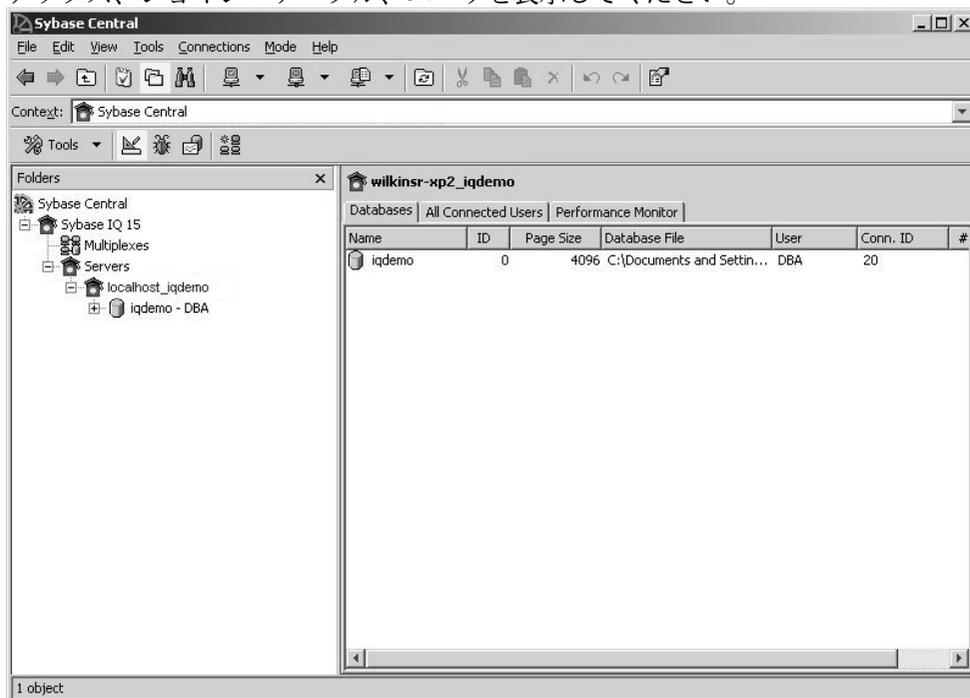
- `uname -n` によって返されるホスト名を使用してエージェントを起動するには、次のように、オプションの `-host` パラメータを使用します。

```
S99SybaseIQAgent15 -host
```

- ホストのエイリアスを使用して Agent を起動するには、次のように入力します。S99SybaseIQAgent15 -host <foo> この場合、`foo` は、`/etc/hosts` ファイル内のエイリアスです。

## Sybase Central メイン・ウィンドウのナビゲーション

このチュートリアルの手順に従って、データベース内のテーブル、カラム、インデックス、ジョイン・テーブル、ユーザを表示してください。



## データベース・スキーマの表示

データベース・スキーマは、データベース内のすべてのオブジェクトの集まりです。Sybase Central では、データベース・スキーマは、コンテナとその内容の階層として表示されます。

データベース・コンテナを展開し、コンテナの内容を表示するにはいくつかの方法があります。

- [タスク] ビューの右側のウィンドウ枠にある [コンテンツ] タブに、データベース・オブジェクトのフォルダ・ビューが表示されます。特定のタスクを実行するには、右側のウィンドウ枠でオブジェクトをダブルクリックすると、左側のウィンドウ枠にそのオブジェクトに関連する一連のタスクが表示されます。タスク・リストから、実行対象のタスクをクリックします。
- [Folders] ビューの左側のウィンドウ枠にある階層ツリーに、データベース・オブジェクトが表示されます。ツリーのブランチをクリックするとビューを展開または折りたたみ、右側のウィンドウ枠にそのオブジェクトの詳細が表示さ

れます。たとえば、左側のウィンドウ枠で Sybase IQ プラグインを選択すると、右側のウィンドウ枠に、Sybase IQ のサーバ、ユーティリティ、サービスが表示されます。

- [検索] ウィンドウ枠では、オブジェクト階層内の項目を検索できます。項目を検索するには、[検索文字列] ボックスにオブジェクトの名前を入力し、[場所] ドロップダウンでオブジェクト・クラスを選択してから [検索] をクリックします。[結果] ウィンドウ枠内のオブジェクトをクリックすると、右側のウィンドウ枠内にそのオブジェクトが表示されます。

---

**注意：** 特定のフィールドを表示できない場合は、ウィンドウ・システムの機能を使用して、白または明るい背景に暗い文字が表示されるように、Sybase Central の表示を変更します。

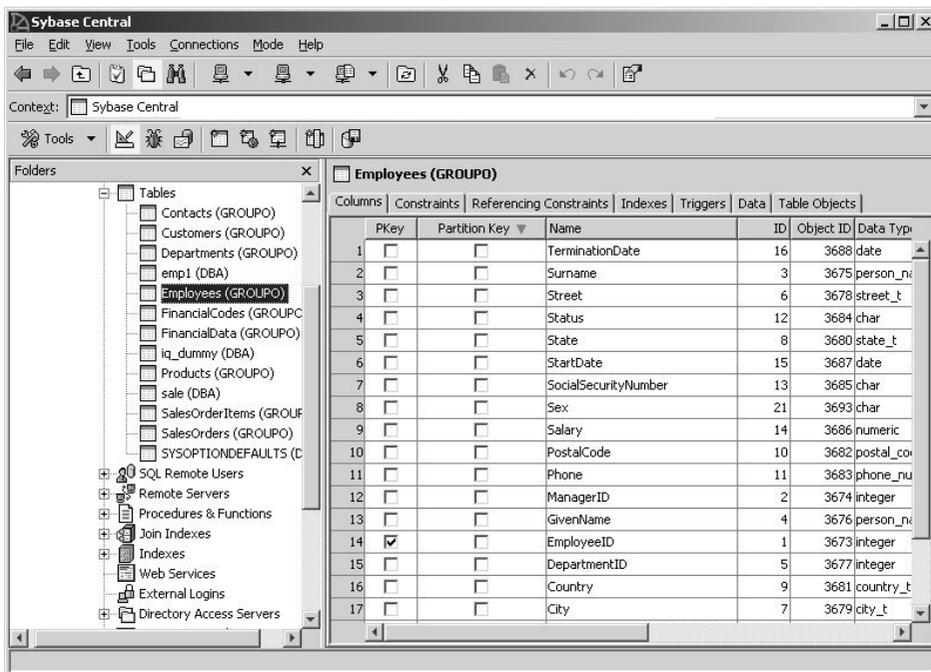
---

### データベース内のテーブルの表示

[テーブル] フォルダを使用すると、データベース・テーブルにアクセスできます。

1. データベースに接続し、[テーブル] フォルダをクリックしてから、右側のウィンドウ枠にカラムの定義を表示するテーブルをクリックします。  
右側のウィンドウ枠には、テーブルのカラム、制約、参照制約、インデックス、カラム用にフォルダを格納している各テーブル・オブジェクト、外部キー、関係、インデックスなどを表示するタブがあります。
2. 右側のウィンドウ枠で、タブをクリックして、テーブルのカラム、制約、参照制約、インデックス、トリガ、データ、およびテーブル・オブジェクトを表示します。

## サーバの起動と接続



## その他のデータベース・オブジェクトの表示

すべての Sybase IQ サーバには、データベース・フォルダがあります。

オブジェクト	説明
DB 領域	データベース内の dbfile の論理集合です。
ディレクトリ・アクセス・サーバ	データベース・サーバを実行しているコンピュータのローカル・ファイル・システムへのアクセスを提供します。ディレクトリ・アクセス・サーバのプロキシ・テーブルを作成すると、サーバ・コンピュータ上にあるファイルやディレクトリにアクセスできます。
ドメイン	ユーザ定義データ型と呼ばれることもあるドメインは、組み込みデータ型のエイリアスです。
イベント	特定の時刻に実行することがスケジュールされているストアド・プロシージャなどのように、データベース管理を自動化します。
外部環境	データベース・サーバ外のプログラミング言語、ユーティリティ、およびオブジェクトです。
外部ログイン	これを使用して、リモート・サーバまたはディレクトリ・アクセス・サーバ上のディレクトリにアクセスできます。

オブジェクト	説明
インデックス	クエリ処理のためにデータを格納および表現する構造体。インデックスにより、情報をすばやく検索できます。データベースの場合、インデックスは、インデックスが付けられた各カラム値を、インデックス値を含むデータのローが格納される物理位置に関連付けます。
ジョイン・インデックス	個々のカラムではなく、カラムのセットに対して作成されるジョインであり、2つ以上のテーブルの外部ジョイン全体を表します。
ログイン・マッピング	Windows ユーザ・プロファイルまたは Kerberos プリンシパルを既存のデータベース・ユーザにマップします。
ログイン・ポリシー	ユーザのデータベースへの接続時に適用される規則を定義する一連のオプションです。
プロシージャとファンクション	モジュールベースの言語の SQL プロシージャ。
パブリケーション	SQL Remote 内の複製データを識別します。Sybase IQ データベースでは、パブリケーションは、SQL Anywhere テーブル上でしか作成できません。
リモート・サーバ	これを使用して、別のデータベース・サーバ上にあるデータにアクセスできます。
SQL Remote サブスクリプション	パブリッシャからリモート・データベース (サブスクライバ) へのリンクを作成します。
SQL Remote ユーザ	SQL Remote セットアップでリモート・データベースを定義します。
システム・トリガ	システム・テーブルを修正するイベント。
テーブル	データベースに格納されるベース・テーブル。
トリガ	データが変更されたときに SQL 文またはプロシージャを自動的に実行します。
ユーザとグループ	データベースのユーザと権限を識別するメカニズム。
ビュー	データベース内にクエリとして格納され、アクセス時に評価される計算テーブル。
Web Services	Web サーバに対して送付された HTML 要求および XML 要求の処理方法を規定します。

### データベース・サーバからの切断

切断オプションを使用すると、データベース・サーバからログオフできます。複数のサーバに接続している場合は、指定したサーバとの接続を切断できます。接続されているデータベースが1つだけの場合は、そのデータベースが自動的に切断されます。

次のいずれかを行います。

- [接続] - [切断] を選択します。
- ツールバーにある [切断] ボタンをクリックします。
- データベース・コンテナを右クリックし、[切断] を選択します。

## Interactive SQL を使用したデータベースへの接続

Interactive SQL は、Sybase IQ に同梱されているツールです。このツールを使用すると、SQL 文の実行、スクリプトの作成、データベース・データの表示ができます。Interactive SQL を起動するには、コマンド・ラインまたは Windows のアプリケーションを使用します。このどちらを使用するかは、使用するオペレーティング・システムによって決まります。

### Sybase Central から Interactive SQL を起動する

Sybase Central から Interactive SQL を起動します。

1. 左ウィンドウ枠で、Sybase IQ プラグインを選択し、次のいずれかを実行します。
  - [ツール] > [Sybase IQ 15] > [Interactive SQL のオープン] を選択します。
  - データベースを右クリックし、[Interactive SQL を開く] を選択します。
  - ストアド・プロシージャを右クリックし、[Interactive SQL から実行] を選択します。
2. [接続] ウィンドウを開き、パラメータを指定します。

### コマンド・ラインから Interactive SQL を起動する

コマンド・ラインから Interactive SQL を起動し、データベースに接続できます。コンソールまたはコマンド・ラインから次のように入力します。

```
dbisql -c "uid=DBA;pwd=sql" -host <host name> -port 2638
```

-c パラメータは、接続パラメータを指定します。接続パラメータの完全なリストについては、『システム管理ガイド：第1巻』の「接続パラメータと通信パラメータ」を参照してください。

**注意：** iqdemo データベースとその他の Sybase IQ データベースのデフォルトのユーザ ID とパスワードは、DBA と sql です。データベースの接続に関して問題が

発生する場合は、ユーザ ID とパスワードの大文字と小文字を変更して指定してみてください。データベースの大文字と小文字の区別がオンに設定されている可能性があります。

---

## Windows での Interactive SQL の起動

Windows では、Interactive SQL を Sybase IQ プログラム・グループから起動できません。

1. [スタート]>[プログラム]>[Sybase]>[Sybase IQ]>[Interactive SQL] をクリックします。
2. [ID] タブの [ユーザ] ボックスに DBA を入力し、[パスワード] ボックスに sql を入力します。  
これは、Sybase IQ データベースのデフォルトのユーザ ID とパスワードです。DBA は、大文字でも小文字でもかまいませんが、sql は大文字と小文字を区別するため小文字で入力する必要があります。
3. [データベース] タブで、[サーバ名] ボックスからサーバを選択します。[検索] をクリックして実行中のサーバを検索します。  
サーバ名は、<hostname>\_dbname 形式で表示されます。ホスト名が localhost-xp で、サンプルの iqdemo データベースを実行中の場合、サーバ名は localhost-xp\_iqdemo になります。サーバがリモートの場合は、サーバ名の代わりに host:port を入力します。
4. [データベース名] ボックスでデータベースを選択します。  
実行中のデータベースが 1 つのみの場合は、このフィールドを空白のままにできます。
5. 次に行う手順は、次のように、サーバの場所によって異なります。
  - 同じマシンにインストールされたサーバに対して Interactive SQL を実行するには、適切なデータベース・ファイルを選択してください。
  - ネットワーク・サーバに対して Interactive SQL を実行するには、[ネットワーク] タブをクリックし、[TCP/IP] ボックスをクリックして、ホスト名とポート番号を指定します。
6. [OK] をクリックします。

他のアプリケーションも、ほとんど同様の方法でデータベース・サーバに接続します。接続するには、ユーザ ID とパスワードを送信する必要があります。

Interactive SQL の使用方法の詳細については、『ユーティリティ・ガイド』を参照してください。

## データベースへのコマンドの送信

データベースに接続したら、データベースにコマンドを送信できます。

たとえば、[SQL 文] ウィンドウに、次の行を入力します。

```
SELECT * FROM SalesOrders
```

ツールバーにある右向きの三角形のボタンをクリックするか、[F9] キーを押して、クエリを実行します。SQL 文に、SalesOrders テーブルのすべての(\*)カラムが表示されます。

## データベース・サーバの停止

Sybase Central または Interactive SQL を使用して、サーバを停止します。

次のいずれかを行います。

サーバの停止に使用するもの	手順
<b>Sybase Central</b>	<ul style="list-style-type: none"><li>• [ファイル]-[サーバの停止] を選択します。</li><li>• 該当するサーバを右クリックし、[サーバの停止] を選択します。</li></ul>
<b>Interactive SQL</b>	<p><b>STOP ENGINE</b> 文を使用します。</p> <p>Interactive SQL で、次のコマンドを実行します。</p> <pre>STOP ENGINE [database-server-name] [UNCONDITIONALLY]</pre> <p>STOP ENGINE 文は、指定したデータベース・サーバを停止します。UNCONDITIONALLY キーワードを指定すると、別の接続がある場合でもデータベース・サーバを停止します。デフォルトでは、データベース・サーバは、そのサーバへの別の接続がある場合には停止されません。</p>

# データの選択

この項の目的は、基本的なクエリの構築と設計について理解を深めることです。ほとんどのトピックでは、基本的な単一テーブルの SELECT 文に重点を置き、クエリの作成に使用できる文の例を用意しています。

この項では、すべての例で Sybase IQ デモ・データベース (iqdemo.db) をデータ・ソースとして使用しています。この項内のクエリを実行するには、デモ・データベースを起動し、Interactive SQL を開き、iqdemo に接続します。

## テーブル情報の表示

クエリがデータベースから情報を取得します。すべてのクエリには、SELECT 文があり、クエリが戻すデータを特定する追加の句を含めることができます。クエリ結果は、結果セットという結果テーブルに格納されます。

### テーブルのリスト

ストアド・プロシージャを使用すると、デモ・データベース内のテーブルをリストできます。テーブルをリストするには、[SQL 文] ウィンドウで

```
sp_iqtable
```

を入力します。

### SELECT 文の使用

**SELECT** 文を使用して、データベース・テーブルからデータを取得します。

Employees テーブルのすべてのローとカラムを取得するには、次のように入力します。

```
SELECT * FROM Employees
```

SQL 文にあるアスタリスクは、テーブル内のすべてのカラムを取得することを示すワイルドカードです。[Interactive SQL 結果] ウィンドウに、次の結果が表示されます。

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
102	501	Whitney	Fran	100	...
105	501	Cobb	Matthew	100	...
129	902	Chin	Philip	200	...
148	1,293	Jordan	Julie	300	...

## データの選択

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
160	501	Breault	Robert	100	...
184	1,576	Espinoza	Melissa	400	...
191	703	Bertrand	Jeannette	500	...
195	902	Dill	Marc	200	...
207	1576	Francis	Jane	400	...

Employees テーブルでは、複数のローがカラムに編成されています。各カラムには、Surname や EmployeeID などの名前が付いています。会社の従業員 1 人ずつに 1 つのローがあり、それぞれのローは各カラムに値を持ちます。たとえば、EmployeeID が 102 の従業員は Fran Whitney であり、そのマネージャは ManagerID 501 です。

Interactive SQL の [メッセージ] ウィンドウにも一部の情報が表示されます。この情報については後で説明します。

---

**注意：**クエリ結果を表示しているこのマニュアルの表は、クエリによって返されるデータの一部分だけを含んでいる場合があります。省略記号が表示されたカラムとローは、その他のクエリ結果を示しています。

---

### 大文字と小文字の区別

デモ・データベースは、大文字と小文字を区別しません。つまり、比較や文字列の操作で、大文字と小文字の差異は考慮されません。例えば、テーブル名 Employees は、実際のテーブル名がすべて小文字の場合にも、先頭は大文字の E で表示されます。Sybase IQ データベースは、文字列の比較で大文字と小文字を区別するもの (デフォルト) と区別しないものを作成できますが、その識別子では常に大文字と小文字は区別されません。

---

**注意：**この項の例は、**CREATE DATABASE** 修飾子の **CASE IGNORE** を使用して、大文字と小文字を区別しないように作成されています。デフォルトは、**CASE RESPECT** であり、こちらの方がパフォーマンスが向上します。

---

**SELECT** の代わりに **select** または **Select** と入力することもできます。Sybase IQ では、キーワードを大文字、小文字、またはその両方の組み合わせで入力できます。このマニュアルでは、通常、SQL キーワードに大文字を使用しています。

## クエリ結果の順序付け

Sybase IQ は、結果セットのローを特定の順序で返しませんが、**ORDER BY** 句を **SELECT** 文に追加すると、その結果がアルファベット順または番号順に返されます。

従業員をアルファベット順にリストする

**ORDER BY** 句を **SELECT** 文に追加すると、その結果セットがアルファベット順または番号順に返されます。句の順序は重要です。**ORDER BY** 句は、**FROM** 句と **SELECT** 句の後に指定します。例：

```
SELECT * FROM Employees ORDER BY Surname
```

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
1,751	1,576	Ahmed	Alex	400	...
1,013	703	Barker	Joseph	500	...
591	1,576	Barletta	Irene	400	...
191	703	Bertrand	Jeannette	500	...
1,336	1,293	Bigelow	Janet	300	...
1,062	1,576	Blaikie	Barbara	400	...
750	703	Braun	Jane	500	...
160	501	Breault	Robert	100	...
1,191	1,576	Bucceri	Matthew	400	...

**注意：** **FROM** 句を省略した場合、またはクエリ内のすべてのテーブルがシステム DB 領域にある場合、クエリは Sybase IQ ではなく SQL Anywhere によって処理されます。このため、特に構文およびセマンティックの制限やオプションの設定方法の違いによって、動作が変わる可能性があります。処理に適用されるルールについては、SQL Anywhere のマニュアルを参照してください。

**FROM** 句を必要としないクエリがある場合は、**FROM iq\_dummy** 句を追加することによって、強制的に Sybase IQ で処理できます。この **iq\_dummy** は、ユーザが自分のデータベースに作成する 1 ロー 1 カラムのテーブルです。

## カラムとローの選択

---

多くの場合、表示する必要がある情報は、テーブル内のカラムだけです。たとえば、従業員への誕生日カードを作成するには、Surname、DepartmentID、BirthDate の各カラムを表示すれば十分です。

### 各従業員の姓、部署、誕生日のリスト

この例では、各従業員の誕生日、姓、部署 ID を選択します。次のように入力します。

```
SELECT Surname, DepartmentID, BirthDate
FROM Employees
```

Surname	DepartmentID	BirthDate
Whitney	100	1958-06-05
Cobb	100	1960-12-04
Chin	200	1966-10-30
Jordan	300	1951-12-13
Breault	100	1947-05-13

### カラムの並べ替え

この3つのカラムは、**SELECT** コマンドに入力した順序で表示されています。カラムを並べ替えるには、コマンドで指定するカラム名の順序を変更します。たとえば、BirthDate カラムを左側に配置するには、次のコマンドを使用します。

```
SELECT BirthDate, Surname, DepartmentID
FROM Employees
```

### ローの順序付け

次のように、特定のカラムだけを表示すると同時に、ローの順序を指定できます。

```
SELECT BirthDate, Surname, DepartmentID
FROM Employees
ORDER BY Surname
```

次のコマンドのアスタリスクは、テーブル内のすべてのカラムを表す省略記号です。

```
SELECT * FROM Employees
```

訳文不要

**注意：** 以前のリリースと比較して、非常に多くの NULL 値を持つカラムが含まれるクエリが高速で実行されます。ただし、非常に多くの NULL 値がテーブルに挿入される場合は、テーブルへのデータの挿入や更新の処理に時間がかかる可能性があります (以前のリリースと比較した場合)。

## 検索条件の使用

**WHERE** 句を **SELECT** 文に追加すると、特定の条件を満たすレコードだけが取得されます。

### 特定の条件を満たすレコードの選択

**Employees** テーブルにある一部の従業員の情報だけを表示したいことがあります。John という名前の従業員だけを表示する場合には、次のように入力します。

```
SELECT * FROM Employees WHERE GivenName = 'John'
```

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
318	1,576	Crow	John	400	...
862	501	Sheffield	John	100	...
1,483	1,293	Leticq	John	300	...

### アポストロフィおよび大文字と小文字の区別

名前 'John' はアポストロフィ (一重引用符) で囲む必要があります。アポストロフィは、John が文字列であることを示します。引用符 (二重引用符) には別の意味があります。引用符を使用すると、無効な文字列を有効なカラム名やその他の識別子として使用できるようになります。

サンプル・データベースでは大文字と小文字が区別されないため、'JOHN'、'john'、'John' のいずれで検索しても同じ結果が返ります。

次のように、これまで学習した句を組み合わせて実行できます。

```
SELECT GivenName, Surname, BirthDate FROM Employees
WHERE GivenName = 'John'
ORDER BY BirthDate
```

### 句の順序付け

句を指定する順序は重要です。最初に **FROM** 句、次に **WHERE** 句、最後に **ORDER BY** 句を指定します。これ以外の順序で句を入力すると、構文エラーが返されず。

### 行の分割

文を複数の行に分ける必要はありません。[SQL 文] ウィンドウに自由なフォーマットで文を入力できます。入力した文が画面の行数を超えると、[SQL 文] ウィンドウのテキストがスクロールします。

## クエリでの日付の比較

---

**WHERE** 句で比較を使用して、検索条件を満たす一連のローを選択します。

### 1964年3月3日より前に生まれた従業員をリストする

次の例は、日付の不等号検索条件の使い方を示します。次のコマンドを入力します。

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate < 'March 3, 1964'
```

Surname	BirthDate
Whitney	1958-06-05
Cobb	1960-12-04
Jordan	1951-12-13
Breault	1947-05-13
Espinoza	1939-12-14
Dill	1963-07-19
Francis	1954-09-12
Shishov	1949-04-22
...	...

Sybase IQ は、**BirthDate** カラムに日付が格納されていることを認識し、自動的に *March 3, 1964*' を日付に変換します。

## WHERE 句での複合検索条件

より大きい(>)、以上(>=)、以下(<=)、等しくない(<>)などの比較演算子を **AND** や **OR** とともに使うと、より複雑な検索条件を作成できます。

### リストの修飾

1964年3月3日より前に生まれた従業員のうち、Whitney という名前の従業員を除くすべての従業員をリストするには、次のコマンドを入力します。

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate < '1964-3-3'
AND Surname <> 'Whitney'
```

Surname	BirthDate
Cobb	1960-12-04
Jordan	1951-12-13
Breault	1947-05-13
Espinoza	1939-12-14
Dill	1963-07-19
Francis	1954-09-12
Shishov	1949-04-22
...	...

## 検索条件でのパターン・マッチング

**WHERE** 句内の **LIKE** 演算子を使って、カラム内のパターンを検索します。

### 特定の従業員のリスト

姓が "br" で始まるすべての従業員をリストするには、次のコマンドを入力します。

```
SELECT Surname, GivenName
FROM Employees
WHERE Surname LIKE 'br%'
```

Surname	GivenName
Breault	Robert

## データの選択

Surname	GivenName
Braun	Jane

検索条件内の % は、**BR** という文字の後に別の文字が何文字続いてもかまわないことを示します。

### 姓検索の修飾

姓が **BR** で始まり、その直後または数文字後に **T** という文字を含み、**T** で終わるかさらに別の文字が続くすべての従業員をリストするには、次のコマンドを入力します。

```
SELECT Surname, GivenName
FROM Employees
WHERE Surname LIKE 'BR*T%'
```

Surname	GivenName
Breault	Robert

最初の % 記号は文字列 "eaul" と一致し、2 番目の % 記号は空の文字列 (文字なし) と一致します。

**LIKE** とともに使用できるもう 1 つの特殊文字は、**\_** (アンダースコア) 文字です。この特殊文字は、厳密に 1 つの文字と対応します。

**BR\_U%** というパターンは、**BR** で始まり、4 番目の文字が **U** であるすべての名前と一致します。**Braun** では、**\_** が文字 **A** と一致し、**%** が **N** と一致します。

## 発音によるローのマッチング

**SOUNDEX** 関数を使用すると、スペルだけでなく発音によってもローをマッチングできます。

### 発音による姓の検索

"Brown" のように聞こえる名前を持つ従業員に対して電話メッセージが残されているとします。この場合、社内で Brown のように発音される名前を持つ従業員を見つける必要があります。Brown のように発音される姓を持つ従業員をリストするには、次のコマンドを入力します。

```
SELECT Surname, GivenName
FROM Employees
WHERE SOUNDEX( Surname ) = SOUNDEX( 'Brown' )
```

Surname	GivenName
Braun	Jane

この検索条件に一致する従業員は Jane Braun だけです。

## 検索条件のショートカットの使用

**WHERE** 句内の **BETWEEN** 演算子と **IN** 演算子を使って、複数の値を取得します。

### *BETWEEN* の使用

2つの値の間で日付の範囲を選択するには、**WHERE** 句内の **BETWEEN** 演算子を使用されます。たとえば、次の2つのクエリの例は同じです。

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate BETWEEN '1964-1-1'
AND '1965-3-31'
```

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate >= '1964-1-1'
AND BirthDate <= '1965-3-31'
```

### *IN* の使用

**IN** 演算子を使って、**WHERE** 句内にある複数の値を指定します。次の2つのクエリの例は同じです。

```
SELECT Surname, EmployeeID
FROM Employees
WHERE Surname IN ('Yeung', 'Bucceri', 'Charlton')
```

```
SELECT Surname, EmployeeID
FROM Employees
WHERE Surname = 'Yeung'
OR Surname = 'Bucceri'
OR Surname = 'Charlton'
```

## 集約データの取得

集合関数は、単一の値ではなく、一連の値に対して計算を実行します。

### *集合関数の概要*

データベース内の従業員の人数を調べるとします。次の文は、employee テーブルのローの数を取得します。

```
SELECT count( * )
FROM Employees
```

## データの選択

このクエリによって、1つのカラム (count(\*) というタイトル) と1つのロー (従業員数が格納されている) だけで構成されるテーブルが返されます。

Count()
75

次のコマンドは、やや複雑な集約クエリです。

```
SELECT count( * ),
min( BirthDate ),
max( BirthDate )
FROM Employees
```

このクエリの結果セットは、3つのカラムと1つのローで構成されます。3つのカラムには、従業員数、年齢が最も高い従業員の誕生日、年齢が最も低い従業員の誕生日が格納されています。

Count()	Min( BirthDate )	Max( BirthDate )
75	'1936-01-02'	'1973-01-18'

**COUNT**、**MIN**、**MAX** を集合関数と呼びます。集合関数は、**SELECT** 文の **GROUP BY** 句を使用して、データベースに含まれるテーブル全体を要約します。集合関数には、全部で **AVG**、**COUNT**、**MAX**、**MIN**、**STDDEV**、**SUM**、**VARIANCE** の7つがあります。すべての関数が、パラメータとしてカラム名または式を使用します。前述のように、**COUNT** はアスタリスクもパラメータとして使用し、各グループのローの数を返します。

## 集合関数によるグループ化されたデータの取得

集合関数をローのグループに対して使って、テーブル全体についての情報を取得します。

### ローのグループに対する集合関数の使用

各営業担当者が受け持つ注文数をリストするには、次のコマンドを入力します。

```
SELECT SalesRepresentative, count( * )
FROM SalesOrders
GROUP BY SalesRepresentative
```

SalesRepresentative	Count()
129	57
195	50
299	114

SalesRepresentative	Count()
467	56
667	54
690	52
856	55
902	47
949	53
1,142	57
1,596	53

このクエリの結果は、各営業担当者の ID 番号別に、営業担当者の ID が格納されたローと、SalesOrders テーブル内でその ID 番号を持つローの数で構成されます。

**GROUP BY** 句を使用すると、結果のテーブルには、**GROUP BY** で指定したカラムで見つかった値別のローが表示されます。

## グループの制限

**HAVING** 関数を **GROUP BY** 関数と併用して、特定の条件を満たす集合値を取得します。

### *GROUP BY 句の制限*

注文数が 55 を超えるすべての営業担当者をリストするには、次のコマンドを入力します。

```
SELECT SalesRepresentative, count( * )
FROM SalesOrders
GROUP BY SalesRepresentative
HAVING count( * ) > 55
```

SalesRepresentative	count(*)
129	57
299	114
467	56
1,142	57

**注意：** **GROUP BY** は常に **HAVING** の前に指定します。同様に、**WHERE** は **GROUP BY** の前に指定します。

### *WHERE と GROUP BY の使用*

注文数が 55 を超えており、ID が 1000 より大きいすべての営業担当者をリストするには、次のコマンドを入力します。

```
SELECT SalesRepresentative, count( * )
FROM SalesOrders
WHERE SalesRepresentative > 1000
GROUP BY SalesRepresentative
HAVING count( * ) > 55
```

Sybase IQ クエリ・オプティマイザは、述部を **HAVING** 句から **WHERE** 句に移動します。そうすることで、パフォーマンスが向上します。たとえば、上記の例では以下ようになります。

```
GROUP BY SalesRepresentative
HAVING count( * ) > 55
AND SalesRepresentative > 1000
```

**WHERE** 句の代わりに述部を次のように指定した場合、クエリ・オプティマイザは述部を **WHERE** 句に移動します。

Sybase IQ は、この最適化を (OR や IN を伴わない) 単純な条件を使って実行します。このため、**WHERE** 句と **HAVING** 句の両方を含むクエリを構築するときは、できるだけ多くの条件を **WHERE** 句で指定するようにします。

## 小計計算の活用

---

日付や場所などの次元によって異なるデータがある場合に、各次元でデータがどのように異なるかを調べることが必要になる場合があります。**ROLLUP** と **CUBE** は、**GROUP BY** 演算子の拡張機能です。この演算子を使用すると、レベルごとに別々のクエリを使用する代わりに、1つのクエリを使用し、複数レベルのグループ化を使用してデータを計算できます。

### **ROLLUP**

**ROLLUP** は、**GROUP BY** 演算子の拡張機能です。この演算子を使用すると、指定した次元のグループごとに異なる複数レベルの小計、および総計、**ROLLUP** を計算できます。小計は、最も詳細なレベルから総計まで「ロールアップ」します。

#### *製品在庫の集計*

製品テーブルの在庫を、名前、サイズ、色ごとに集計するとします。

```
SELECT Name, Size, Color, Sum(Quantity) FROM Products GROUP BY
ROLLUP (Name, Size, Color);
```

Name	Size	Color	sum(Products.Quantity)
Baseball Cap	One size fits all	Black	112
Baseball Cap	One size fits all	White	12
Baseball Cap	One size fits all	(NULL)	124
Baseball Cap	(NULL)	(NULL)	124
Shorts	Medium	Black	80
Shorts	Medium	(NULL)	80
Shorts	(NULL)	(NULL)	80
Sweatshirt	Large	Blue	32
Sweatshirt	Large	Green	39
Sweatshirt	Large	(NULL)	71
Sweatshirt	(NULL)	(NULL)	71
Tee Shirt	Medium	Orange	54
Tee Shirt	Medium	(NULL)	54
Tee Shirt	One size fits all	Black	75
Tee Shirt	One size fits all	(NULL)	75
Tee Shirt	Small	White	28
Tee Shirt	Small	(NULL)	28
Tee Shirt	(NULL)	(NULL)	157
Visor	One size fits all	Black	28
Visor	One size fits all	White	36
Visor	One size fits all	(NULL)	64
Visor	(NULL)	(NULL)	64
(NULL)	(NULL)	(NULL)	496

このクエリを処理するとき、Sybase IQ は最初に、指定された3つすべてのグループ化式 (name、size、color) によってデータをグループ化し、次に最後の式 (color) を除くすべてのグループ化式によってデータをグループ化します。3番目のローにある NULL 値は、color カラムの **ROLLUP** 値を示します。

## データの選択

**ROLLUP** 演算子には、引数としてグループ化式の順番リストを指定する必要があります。他のグループを含むグループをリストするときは、先に大きい方のグループをリストします。**ROLLUP** 演算子は、集合関数の **AVG**、**COUNT**、**MAX**、**MIN**、**STDDEV**、**SUM**、**VARIANCE** と併用できます。ただし、**ROLLUP** は、**COUNT DISTINCT** と **SUM DISTINCT** をサポートしていません。

## Cube

**CUBE** は、**GROUP BY** 演算子の拡張機能であり、データを複数の次元でグループ化することで分析します。

**CUBE** にはグループ化式 (次元) を順に並べたリストを引数として指定する必要があります。これにより、**SELECT** 文は組み合わせ可能なすべての次元のグループに対して、小計を計算できるようになります。

### 平均給与の計算

次のクエリでは、**Employees** テーブルのデータを使用して、州、性別、部署ごとの平均給与を計算します。

```
SELECT case grouping(state) WHEN 1 THEN 'ALL' ELSE state
       END AS c_state, case grouping(sex) WHEN 1 THEN 'ALL'
       ELSE sex end AS c_gender, case grouping(DepartmentId)
       WHEN 1 THEN 'ALL' ELSE cast(DepartmentId as char(4)) end
       AS c_dept, COUNT(*), CAST(ROUND(AVG(salary),2) AS
       NUMERIC(18,2))AS AVERAGE
FROM employees WHERE state IN ('MA' , 'CA')
GROUP BY CUBE(state, sex, DepartmentId)
ORDER BY 1,2,3;
```

c_state	c_gender	c_dept	COUNT()	AVERAGE
ALL	ALL	200	3	52,200.00
ALL	ALL	ALL	3	52,200.00
ALL	F	200	2	58,650.00
ALL	F	ALL	2	58,650.00
ALL	M	200	1	39,300.00
ALL	M	ALL	1	39,300.00
CA	ALL	200	3	52,200.00
CA	ALL	ALL	3	52,200.00
CA	F	200	2	58,650.00
CA	F	ALL	2	58,650.00

c_state	c_gender	c_dept	COUNT()	AVERAGE
CA	M	200	1	39,300.00
CA	M	ALL	1	39,300.00

**CUBE** でグループを計算するときに、**CUBE** は計算されたグループのカラムに NULL 値を挿入します。各ローが表すグループの種類と、その NULL がデータベースに格納されている NULL なのか、**CUBE** が挿入した NULL なのかを区別することは困難です。この問題を解決するのが **GROUPING** 関数です。指定されたカラムが上位レベルのグループにマージされている場合、この関数は 1 を返します。

**CUBE** が生成した小計ローを示す NULL 値が、クエリ内の指定によって小計ローで ALL に置き換わっています。

## 分析データの取得

分析関数は、結果セット内でデータを操作します。ランク付け統計関数は、グループ内の項目をランク付けします。逆分散統計関数は、K-理論パーセンタイル値を返します。

### ランク付け統計関数

ランク付け統計関数は、グループ内の項目をランク付けし、分布を計算して結果セットを複数のグループに分類します。ランク付け統計関数には、**RANK**、**DENSE\_RANK**、**PERCENT\_RANK**、**NTILE** があります。逆分散統計関数には、**PERCENTILE\_CONT** と **PERCENTILE\_DISC** があります。

#### 従業員の給与のランク付け

たとえば、従業員の給与をランク付けするとします。次の例では、**NTILE** 関数で、従業員の給与に基づいて、従業員を 4 つのグループに分類します。ntile のランクが 1 になっているのは、給与の範囲で上位 25% までの従業員です。

```
SELECT Name
       Salary,
       NTILE(4) OVER(ORDER BY salary DESC)as Ranking
FROM emp1;
```

Name	Salary	Ranking
Sandy	55,000	1
Peter	48,000	1
Lisa	38,000	1

Name	Salary	Ranking
Scott	29,000	1
Tim	29,000	2
Tom	28,000	2
Mike	28,000	2
Adam	25,000	3
Antonia	22,000	3
Jim	22,000	3
Anna	18,000	4
Jeff	18,000	4
Amy	18,000	4

**NTILE** はクエリ結果を指定された数のバケットに分散またはランク付けし、バケット内の各ローにバケット番号を割り当てる統計関数です。結果セットは 10 個 (十分位数)、4 個 (四分位数)、その他の数のグループに分割できます。

ランク付け統計関数では、**OVER (ORDER BY)** 句を指定する必要があります。

**ORDER BY** 句は、ランク付けを実行するパラメータと、各グループでローをソートする順序を指定します。この **ORDER BY** 句は、**OVER** 句の中でのみ使用されるもので、**SELECT** の **ORDER BY** とは異なりますので、注意してください。

**OVER** 句は、関数がクエリの結果セットに対して処理を行うことを示します。結果セットは、**FROM**、**WHERE**、**GROUP BY**、**HAVING** の各句がすべて評価された後に返されるローです。**OVER** 句は、ランク付け統計関数の計算の対象となるローのデータ・セットを定義します。

## 逆分散統計関数

逆分散統計関数は、K-理論パーセンタイル値を返します。これは、データ・セットのスレッシュホールド許容値を決定する際に使用します。

同様に、逆分散統計関数では、**WITHIN GROUP (ORDER BY)** 句を指定する必要があります。**ORDER BY** 句は、パーセンタイル関数を実行する式と、各グループでローをソートする順序を指定します。この **ORDER BY** 句は、**WITHIN GROUP** 句の中でのみ使用されるもので、**SELECT** の **ORDER BY** とは異なります。**WITHIN GROUP** 句は、クエリ結果を順序付けられたデータ・セットに分類します。関数はこのデータ・セットに基づいて結果を計算します。

## 重複したローの削除

---

**DISTINCT** キーワードを使用して、重複したローを削除します。

### *DISTINCT* ローの選択

たとえば、次のコマンドを実行すると、多くの重複したローが返ります。

```
SELECT city, state FROM Employees
```

市と州のユニークな組み合わせだけをリストするには、次のコマンドを使用します。

```
SELECT DISTINCT city, state FROM Employees
```

---

**注意：** **ROLLUP** 演算子と **CUBE** 演算子は、**DISTINCT** キーワードをサポートしていません。

---



# テーブルのジョイン

ジョインによって、複数のテーブルからクエリを構築できます。

## 外積ジョイン

---

2つの外積ジョインの返す結果セットによって、両方のテーブルにあるローの組み合わせで可能なものすべてが生成されます。結果セット内のロー数は、最初のテーブルのロー数と2番目のテーブルのロー数との積です。

### 例

サンプル・データベース内の `FinancialData` と `FinancialCodes` には、架空の会社の財務データが入っています。両方のテーブルにある `code` 列は、2つのテーブルをジョインする場合に使用できます。

次の **SELECT** 文によって、`FinancialCodes` テーブルと `FinancialData` テーブル内のすべてのデータがリストされます。

```
SELECT *  
FROM FinancialCodes, FinancialData
```

結果セットは、`FinancialData` テーブル内のすべてのローとともに、`FinancialCodes` テーブル内のすべてのローと一致します。このジョインを完全外積または直積と呼びます。各ローは、`FinancialCodes` テーブルのすべてのカラム、`FinancialData` テーブルのすべてのカラムの順で構成されます。

外積ジョインは、ジョインを理解するための単純な出発点にすぎず、それ自体はあまり役に立ちません。より選択性の高いジョインによって、外積テーブルに制限が適用されます。

## ジョインの制限

---

クエリに条件を追加すると、結果セットは特定の条件を満たすレコードだけに制限されます。ジョイン条件では比較演算子が使われることによって、外積の結果から一部のローが除外されます。

### 例 1

このクエリ内の **WHERE** 句は、`SalesOrders` テーブルの `SalesRepresentative` が `Employees` テーブルの `EmployeeID` と一致するこ

## テーブルのジョイン

と、という条件を示しています。これにより、各ローに注文と、その注文を担当する営業担当者の情報がある場合に、結果セットが作成されます。

```
SELECT *
FROM SalesOrders, Employees
WHERE SalesOrders.SalesRepresentative = Employees.EmployeeID
```

カラムを識別するために、テーブル名をプレフィクスとして指定します。この例では必ずしも必要ありませんが、テーブル名のプレフィクスを使用すると文が明確になります。2つのテーブルに同じ名前のカラムがあるときは、このプレフィクスを指定する必要があります。このようなコンテキストで使用するテーブル名を「修飾子」と呼びます。

### 例2

次のクエリでは、一部のカラムだけをフェッチし、結果を順序付けするように変更を加えています。

```
SELECT Employees.Surname, SalesOrders.id,
SalesOrders.OrderDate
FROM SalesOrders, Employees
WHERE SalesOrders.SalesRepresentative = Employees.EmployeeID
ORDER BY Employees.Surname
```

**SELECT** コマンドに多くのテーブルがある場合は、修飾子名をいくつも入力しなければならないことがあります。このようなときは、関連名を使用して入力の手間を省くことができます。

### 関連名

関連名は、テーブルの特定のインスタンスのエイリアスです。このエイリアスは、1つの文中でのみ有効です。関連名を作成するには、テーブル名のすぐ後ろに、テーブル名の省略形をキーワード **AS** で区切って指定します。それ以降は、修飾子としてテーブル名の代わりにこの省略形を使用してください。

```
SELECT E.Surname, S.id, S.OrderDate
FROM SalesOrders AS S, Employees AS E
WHERE S.SalesRepresentative = E.EmployeeID
ORDER BY E.Surname
```

この例では、SalesOrders テーブルと Employees テーブルに対応する S と E という2つの関連名を作成しています。

---

**注意：** テーブル名や関連名が必要になるのは、異なるテーブルに同じ名前のカラムがあり、不明確になることを避ける場合だけです。関連名を作成した場合は、テーブル名の代わりに必ず関連名を使用します。関連名を作成していない場合は、テーブル名を使用します。

---

## テーブル関係

---

あるテーブルが別のテーブルとどのくらい関係があるかが分かると、異なるタイプのジョインを構築できます。

### プライマリ・キー 識別子キー

iqdemo データベースのすべてのテーブルには、プライマリ・キーが設定されています。プライマリ・キーは、テーブル内のローをユニークに識別する1つまたは複数のカラムです。たとえば、従業員番号は従業員をユニークに識別するので、EmployeeID は Employees テーブルのプライマリ・キーになります。

SalesOrderItems テーブルは、2つのカラムでプライマリ・キーを構成しているテーブルの例です。注文ID だけでは、SalesOrderItems テーブルのローがユニークに識別されません。注文には複数の項目が含まれる場合があるからです。また、LineID 番号も SalesOrderItems テーブルのローをユニークに識別しません。SalesOrderItems テーブルのローをユニークに識別するには、注文ID 名と LineID の両方が必要です。両方のカラムが一緒になってテーブルのプライマリ・キーになります。

### テーブル関係の外部キー

iqdemo データベースのいくつかのテーブルは、データベース内の他のテーブルを参照しています。たとえば、SalesOrders テーブルには、注文を担当する従業員を示す SalesRepresentative カラムがあります。SalesOrders テーブルには、従業員をユニークに識別するために必要な最小限の情報だけが格納されています。SalesOrders テーブル内の SalesRepresentative カラムは、Employees テーブルに対する外部キーです。

外部キーは、他のテーブルの候補キーの値を含む1つまたは複数のカラムです(候補キーの詳細については、『システム管理ガイド：第1巻』の「データ整合性」の「データ整合性の概要」の「データ整合性ツール」を参照してください)。

## ジョイン演算子

---

ジョインを使用することで、テーブル内にあるカラム間の関係に基づいて複数のテーブルを検索できます。

多くの一般的なジョインは、キーで関連付けられた2つのテーブル間で行われます。最も一般的なジョインでは、あるテーブルの外部キーの値が別のテーブルのプライマリ・キーの値と等しいものに制限されます。すでに見てきた例では、

## テーブルのジョイン

SalesOrders テーブルの外部キーの値が、Employees テーブルの候補キーの値と等しいものに制限されています。

```
SELECT Surname,  
       EmployeeID,  
       OrderDate  
FROM SalesOrders, Employees  
WHERE SalesOrders.SalesRepresentative = Employees.EmployeeID
```

KEY JOIN を使用すると、クエリをより簡単に表現できます。これについて、次の項で説明します。

## キー・ジョイン

キー・ジョインが生成するジョイン条件は、データベース内の外部キー関係に基づいています。

キー・ジョインは、外部キーで関連付けられたテーブルを簡単にジョインする方法です。次の例では、2つの従業員ID番号カラムを結び付ける WHERE 句を使ったクエリと同じ結果が返されています。

```
SELECT Surname,  
       EmployeeID,  
       OrderDate  
FROM SalesOrders  
KEY JOIN Employees
```

```
SELECT Surname,  
       EmployeeID,  
       OrderDate  
FROM SalesOrders, Employees  
WHERE SalesOrders.SalesRepresentative = Employees.EmployeeID
```

ジョイン演算子 (KEY JOIN) は、単に WHERE 句の入力の手間を省くためのもので、2つのクエリはまったく同じものです。

キー・ジョインによるクエリで期待どおりの結果を得るには、アプリケーションで外部キーを強制的に適用する必要があります。

### 複数のテーブルのジョイン

ジョイン演算子を使用して、複数のテーブルをジョインできます。次のクエリでは、4つのテーブルを使用して、注文の合計額を顧客別にリストしています。customer、SalesOrders、SalesOrderItems、Products の4つのテーブルを、テーブルの各ペア間の1つの外部キー関係で接続しています。

```
SELECT CompanyName,  
       CAST( SUM(SalesOrderItems.Quantity *  
               Products.UnitPrice) AS INTEGER) AS Value  
FROM Customers  
KEY JOIN SalesOrders  
KEY JOIN SalesOrderItems
```

KEY JOIN Products  
GROUP BY CompanyName

CompanyName	値
The Power Group	5,808
The Birds Loft	4,404
Sampson & Sons	6,660
Hats Etc.	2,736
Howard Co.	5,388
...	...

このクエリで使用している CAST 関数は、式のデータ型を変換します。この例では、整数として返される合計が値に変換されます。

## ナチュラル・ジョイン

NATURAL JOIN 演算子は、共通のカラム名に基づいて2つのテーブルをジョインします。つまり、Sybase IQ が、各テーブルに共通するカラムを結び付ける WHERE 句を生成します。

### 例

たとえば、次のようなクエリがあるとします。

```
SELECT Surname,
       DepartmentName
FROM Employees
NATURAL JOIN Departments
```

この例では、データベース・サーバが2つのテーブルを参照し、共通するカラム名は DepartmentID だけであると判断します。次の ON フレーズが内部的に生成され、ジョインの実行に使用されます。

```
FROM Employees JOIN Departments
...
ON Employees.DepartmentID = Departments.DepartmentID
```

### NATURAL JOIN を使用したときのエラー

このジョイン演算子では、意図しないカラムを結び付けてしまう問題が起きる可能性があります。たとえば、次のクエリは意図しなかった結果をもたらします。

```
SELECT *
FROM SalesOrders
NATURAL JOIN Customers
```

このクエリの結果には、ローが1つもありません。

データベース・サーバは、内部的に次の ON フレーズを生成します。

```
FROM SalesOrders JOIN Customers
ON SalesOrders.ID = Customers.ID
```

SalesOrders テーブル内の id カラムは注文の ID 番号です。Customer テーブル内の id カラムは顧客の ID 番号です。これらの番号は 1 つも一致しません。たとえ一致する番号があったとしても、当然それは意味を持ちません。

ジョイン演算子をむやみに使用しないように注意してください。ジョイン演算子は、単に強制力のない外部キーや共通のカラム名で WHERE 句を入力する手間を省くためのものであることを忘れないでください。WHERE 句を注意して使用しないと、意図しない結果をもたらすクエリを作成してしまう可能性があります。

## アドホック・ジョインとジョイン・インデックスの使用

ジョイン・インデックスは、複数テーブル内のカラム間の関係を定義する内部構造です。アドホック・ジョインは、ジョイン・インデックスを使用しないジョインです。

クエリで参照されるジョイン・カラムにジョイン・インデックスを定義している場合、通常、Sybase IQ はそれらを使用して、テーブルをジョインするクエリを実行します (ジョイン・インデックスの定義については、『システム管理ガイド第 1 巻』の「Sybase IQ インデックス」を参照してください)。

ジョイン・インデックスを使用しないジョインをアドホック・ジョインと呼びます。クエリでいくつものテーブルを参照しており、その中にジョイン・インデックスが定義されていないテーブルがある場合、Sybase IQ は定義されているテーブルではジョイン・インデックスを、それ以外のテーブルではアドホック・ジョインを使用します。

可能なすべてのジョインに対してジョイン・インデックスを作成することはできないため、ときにはアドホック・ジョインが必要になることがあります。Sybase IQ の最適化によって、クエリはジョイン・インデックスなしでも同等かそれ以上のパフォーマンスで実行されます。

ジョイン・インデックスの作成には、次の制約があります。

- すべてのジョイン・インデックスは、完全な外部ジョインを使用して作成されます。ジョイン・インデックスを使用するクエリは、内部、左外部、または右外部のジョインになります。  
完全な外部ジョインでは、指定された左右両方のテーブルのすべてのローが結果に含まれ、対応するカラムに一致する値がないカラムについては NULL が返されます。
- ジョイン述部の ON 句で使用できる比較演算子は EQUALS だけです。

- ON 句の代わりに NATURAL キーワードを使用できますが、1 対のテーブルしか指定できません。
- 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。
- ジョイン・インデックスは、同じような数のローがあるテーブルを含んでいる場合、アドホック・ジョインに比べて優れたパフォーマンスを示す傾向があります。また、大きなテーブルと小さなテーブルの間に大幅な違いがある場合、アドホック・ジョインに比べてパフォーマンスが低下します。

## ジョインとデータ型

---

Sybase IQ によるジョイン・カラムのデータ型の処理について理解すると、クエリをより効率的に記述するときに役立ちます。

最適なパフォーマンスを得るには、ジョイン・カラムを類似のデータ型にする必要があります。Sybase IQ では、暗黙の変換が存在する任意のデータ型でアドホック・ジョインを行うことができます。ただし、ジョイン・カラムのデータ型が同じでない場合は、データ型とテーブルのサイズによって、パフォーマンスがさまざまな範囲で低下する可能性があります。たとえば、INT を BIGINT のカラムにジョインすることはできますが、このジョインによって特定の種類の最適化ができなくなります。Sybase IQ インデックス・アドバイザでは、パフォーマンスに影響する可能性があるこのような一致しないジョイン・データ型を識別できます。

同じようなデータ型のジョイン・キーは、幅の広いデータ型のキーより優れたパフォーマンスを示す傾向があり、整数データ型のジョイン・キーは数値データ型または文字データ型のキーより高速になる傾向があります。

これらのデータ型は優れたパフォーマンスを示しますが、データ型が一致するキーを選択する方が、高速でも一致しないデータ型のキーを選択するより効率のよいジョイントが得られます。データ型が同じでない場合、Sybase IQ では、カラムを同等にするために、パフォーマンスの低下を招く可能性のあるデータ型のいずれかを内部的に変換する必要があります。

暗黙的なデータ型変換の表については、『システム管理ガイド：第 1 巻』の「データベースへのデータの入出力」を参照してください。訳文不要

## ストア間またはデータベース間ジョインのサポート

---

指定された Sybase IQ データベース内では、あらゆる種類のジョインがサポートされます。つまり、IQ カタログ・ストアの任意のシステムまたはユーザ・テーブルを、IQ メイン・ストアの任意のテーブルに任意の順序でジョインできます。

### *Adaptive Server Enterprise* テーブルと *Sybase IQ* テーブルのジョイン

Sybase IQ テーブルと Adaptive Server Enterprise データベースのテーブルのジョインは、次の条件下でサポートされます。

- Sybase IQ データベースは、ローカル・データベースとリモート・データベースのどちらでもかまいません。
- ASE で Sybase IQ テーブルをプロキシ・テーブルとして使用する場合は、テーブル名を 30 文字以内にしてください。
- ローカルの Adaptive Server Enterprise テーブルをリモートの Sybase IQ 12 テーブルにジョインするには、ASE のバージョンが 11.9.2 以降である必要があります。また、次の適切なサーバ・クラスを使用してください。
  - Adaptive Server Enterprise 12.5 以降のフロントエンドからリモートの Sybase IQ 12.5 以降に接続するには、ASE 12.5 で追加された ASIQ サーバ・クラスを使用します。
  - Adaptive Server Enterprise 11.9.2 から 12.0 までのフロントエンドからリモートの Sybase IQ 12.x (または SQL Anywhere 6.x 以降) に接続するには、サーバ・クラス ASAnywhere を使用します。
- ローカルの Sybase IQ テーブルを任意のリモート・テーブルとジョインする場合は、ローカルのテーブルを **FROM** 句の最初に指定する必要があります。つまり、ローカルのテーブルは、ジョインの最も外側のテーブルになります。

Sybase IQ と Adaptive Server Enterprise の間のジョインは、コンポーネント統合サービス (CIS) に依存します。

Adaptive Server Enterprise データベースから Sybase IQ へのクエリの詳細については、Adaptive Server Enterprise 主要マニュアル・セットの『コンポーネント統合サービス・ユーザズ・ガイド』を参照してください。

Sybase IQ から他のデータベースへのクエリの詳細については、『リモート・データベースと異種データベースのクエリ』を参照してください。

### *SQL Anywhere* テーブルと *Sybase IQ* テーブルのジョイン

データベースが BLANK PADDING OFF を指定して構築された場合、CHAR データ型は SQL Anywhere と Sybase IQ の間で互換性がありません。文字データをジョイン・キーとして使用して、SQL Anywhere テーブルと Sybase IQ テーブルの間で

データベースのジョインを実行する場合は、BLANK PADDING ON を指定して CHAR データ型を使用します。

---

**注意：** Sybase IQ CREATE DATABASE では、新しいデータベースに BLANK PADDING OFF を指定できなくなりました。この変更は、既存のデータベースには影響しません。BlankPadding データベース・プロパティを使って、既存のデータベースの状態をテストできます。

```
select db_property ( 'BlankPadding' )
```

Sybase では、ジョイン結果が正しくなるように、BLANK PADDING OFF によって影響を受ける既存のカラムをすべて変更することを推奨しています。ジョイン・カラムを VARCHAR ではなく、CHAR データ型として再作成します。CHAR カラムでは、常にブランクが埋め込まれます。

---

## リモート・データベースと異種データベースのクエリ

---

コンポーネント統合サービス (CIS) を使用すると、リモート・データベースと非リレーショナル・データ・ソースにクエリを実行できます。CIS は Sybase IQ の一部としてインストールされます。

CIS を使用すると、リモート・サーバ上のテーブルに、ローカルのテーブルのようにアクセスできます。CIS は、複数のリモート異種サーバのテーブル間でジョインを実行し、1つのテーブルの内容を、サポートされているリモート・サーバへ転送します。

## リモート・データベースのジョイン

---

リモートのデータベースやデータ・ソースにクエリを実行するには、そのテーブルをローカル・プロキシ・テーブルにマッピングする必要があります。CIS は、データがローカルに格納されているかのように、プロキシ・テーブルをクライアント・アプリケーションに示します。テーブルにクエリを実行すると、CIS は実際のサーバ記憶位置を判別します。

1. プロキシ・テーブルを作成します。  
『システム管理ガイド：第2巻』を参照してください。
2. リモート・テーブルをプロキシ・テーブルにマッピングします。

プロキシ・データベース名を各リモート・テーブルの修飾名として使用し、**SELECT** 文でプロキシ・テーブルを参照します。次に例を示します。

```
SELECT a.c_custkey, b.o_orderkey FROM proxy_iqdemo..cust2 a,
iqdemo..orders b WHERE a.c_custkey = b.o_custkey
```

『システム管理ガイド：第2巻』の「リモート・データへのアクセス」と「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。

## ジョインとサブクエリ

サブクエリは、メイン・クエリの **SELECT** 句、**FROM** 句、**WHERE** 句、または **HAVING** 句の中に置かれます。これにより、別のテーブルから取得した指定に従って、1つのテーブルからローを選択できます。

ジョインは、複数のテーブルのデータから構築される結果テーブルを返します。サブクエリを使用して、同じ結果テーブルを取得することもできます。サブクエリは、単に別の **SELECT** 文の中にある **SELECT** 文です。より複雑で多くの情報を与えるクエリを構築するときに、このツールが役立ちます。

### ジョインの使用

1994年以降の注文 (order\_id、OrderDate、CompanyName) を時系列にリストとして作成するには、次のように入力します。

```
SELECT      SalesOrders.ID,
SalesOrders.OrderDate,
Customers.CompanyName
FROM SalesOrders
KEY JOIN Customers
WHERE OrderDate > '1994/01/01'
ORDER BY OrderDate
```

ID	OrderDate	CompanyName
2131	2000-01-02	BoSox Club
2126	2000-01-03	Leisure Time
2065	2000-01-03	Bloomfields
2127	2000-01-06	Creative Customs Inc.
2135	2000-01-06	East Coast Traders
2129	2000-01-07	Hospital Gifts
2132	2000-01-08	The Pep Squad
2136	2000-01-09	Divas Design
2133	2000-01-10	The Road Side Inn
2083	2000-01-13	Pollys Custom Design

### 外部ジョインの使用

前項のチュートリアルジョインは、より正確には **内部ジョイン** と呼びます。外部ジョインを明示的に指定します。この場合は、**GROUP BY** 句も必要です。

```
SELECT  CompanyName,
        MAX( SalesOrders.ID ),State
FROM    Customers
KEY LEFT OUTER JOIN SalesOrders
WHERE   State = 'WA'
GROUP BY CompanyName, State
```

CompanyName	MAX(SalesOrders.ID)	状態
Its a Hit!	(NULL)	WA
Custom Designs	2547	WA

### サブクエリの使用

在庫が少ない製品の注文項目をリストするには、次のコマンドを入力します。

```
SELECT *
FROM SalesOrderItems
WHERE ProductID IN
( SELECT ID
  FROM Products
  WHERE Quantity < 20 )
ORDER BY ShipDate DESC
```

ID	LineID	ProductID	Quantity	ShipDate
2082	1	401	48	2001-07-09
2053	1	401	60	2001-06-30
2125	2	401	36	2001-06-28
2027	1	401	12	2001-06-17
2062	1	401	36	2001-06-17
2023	1	401	24	2001-06-09
2031	2	401	48	2001-06-02
2073	1	401	24	2001-06-02
2647	1	401	36	2001-05-26
2639	1	401	36	2001-05-19

カッコで囲まれたフレーズが、この文のサブクエリです。

```
( SELECT ID
  FROM Products
  WHERE Quantity < 20 )
```

## テーブルのジョイン

サブクエリを使用すると、検索を1回のクエリだけで実行できるようになります。このため、在庫が少ない製品のリストをクエリで検索し、さらにその製品の注文を別のクエリで検索する必要がなくなります。

このサブクエリは、製品テーブルの `id` カラム内にある、**WHERE** 句の検索条件を満たすすべての値をリストします。

サブクエリについては、次の点に注意してください。

- **NOT EXISTS** 述部を使用するクエリなど、ジョインの構築に問題がある場合もサブクエリが役立つことがあります。
- サブクエリが返せるのは1つのカラムだけです。
- サブクエリは、比較の引数、**IN**、または **EXISTS** 句としてのみ使用できます。
- 外部ジョインの **ON** 句の中に、サブクエリを使用することはできません。

## データベースの管理

Sybase Central でオブジェクトに対して実行できるアクションを確認するには、オブジェクトを右クリックします。表示されるサブメニューを使用して、プロパティの表示や作成用のウィザードの起動など、さまざまな操作ができます。データベース内のオブジェクトを選択すると、作成ウィザード・ツールバー・ボタンも表示されます。

### テーブルの管理

適切なウィザードを使用して、データベースにテーブルを追加し、テーブルのプロパティを定義します。

表 7: テーブル作成ウィザード

対象	使用するウィザード
永続テーブル	テーブル作成ウィザード
接続時のみ存在するデータ。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。	グローバル・テンポラリ・テーブル作成ウィザード
リモート・サーバ上のテーブル。	プロキシ・テーブル作成ウィザード

### Office テーブルの作成

テーブルをデータベースに追加することで、データベースのユーティリティを拡張できます。次の例は、office テーブルをデモ・データベースに追加する方法です。

1. [DB 領域] を右クリックし、[新規] をポイントし、[DB 領域] を選択します。
2. 新しいテーブルに office という名前を付け、[次へ] をクリックします。
3. 新しいテーブルを格納するための DB 領域として **IQ\_SYSTEM\_MAIN** をクリックし、[次へ] をクリックします。
4. [コメント] フィールドに、Company offices.
5. [終了] をクリックします。

テーブル・ウィザードによって、office テーブルが作成され、右側のウィンドウ枠に [カラム] タブが表示され、カラムの定義を要求するメッセージが表示されます。次の手順に従って、新しい office テーブルにカラムを追加します。

## office テーブルのカラムの作成

[カラム] タブの各ローは 1 つのカラムを表し、そのカラムのすべての属性が格納されています。

- 次の表にある値を使用して最初のカラムを作成してください。[プライマリ・キー] チェックボックスはオンのままにします。ID、Size、Scale、Nulls、Unique、Comments の各属性は空白のままにします。

Pkey	[Name]	データ型
.	office_id	integer

- 新しいカラムを作成するには、以下を実行します。
  - 現在のローの最後までタブで移動します。
  - ツールバーにある [新しいカラム] ボタンをクリックします。
  - [ファイル] をクリックし、[新規] をポイントして [カラム] を選択します。
- 次の属性を使用して、office テーブルのカラムを作成します。

[Name]	データ型	Size	NULL
region	char	7	.
street	char	40	.
city	char	20	.
state	char	2	.
zip	char	9	.

- カラムの定義を保存するには、以下を実行します。
  - ツールバーにある [テーブルの保存] ボタンをクリックします。
  - [ファイル] をクリックし、[保存] を選択します。

Sybase Central の右側のウィンドウ枠内の [テーブル] タブに、office テーブルが表示されます。office テーブルは、iqdemo データベースに追加されました。データを追加するには、[データ] タブをクリックし、右クリックをすると表示されるメニューから [ローの追加] を選択するか、Interactive SQL のユーティリティを使用します。次の項では、既存のテーブルを編集する方法について説明します。

## カラム・プロパティの編集

テーブル内のすべてのカラムには、カラムの特性を定義する一連のプロパティがあります。カラムが作成され、そのカラム内にデータがある場合には、カラムのプロパティの変更が必要になる可能性があります。

次の例では、office テーブルのカラム名を変更する方法を示します。

1. カラムのプロパティを表示するには、以下を実行します。
  - [タスク] ビューで [その他のタスク] のリストからこのデータベース内の別のテーブルで行う作業を選択します。office テーブルをダブルクリックします。
  - [フォルダ] ビューで、office テーブルをクリックします。
2. 右側のウィンドウ枠で [カラム] タブを選択します。
3. zip カラムの名前を postal\_code に変更します。変更したローにカラムが変更されたことを示すアイコンが表示されていることに注意してください。
4. カラムの新しい定義を保存するには、以下を実行します。
  - ツールバーにある [テーブルの保存] ボタンをクリックします。
  - [ファイル] をクリックし、[保存] を選択します。

## Office テーブルのデモ・データベースからの削除

テーブルを削除すると、不要なテーブルをデータベースから削除できます。

次の例は、office テーブルをデモ・データベースから削除する方法です。

office テーブルを右クリックして、[削除] を選択します。

同じ方法でプロキシ・テーブルを削除できます。新しいローカル・リモート・オプションを指定して作成したプロキシ・テーブルを削除する場合にリモート・サーバにアクセスできない場合は、リモート・サーバにアクセスできないことを示すエラー・メッセージが返されます。

---

**警告！** 新しいローカルおよびリモート・テーブル・オプションを指定して作成したプロキシ・テーブルを削除すると、リモート・テーブルも削除されます。リモート・データベースに格納されているすべてのデータが失われます。

---

## プロシージャの管理

---

ストアド・プロシージャは、Sybase IQ サーバ上で特定のタスクを実行する SQL 文の集まりです。Sybase Central では、ストアド・プロシージャは、[プロシージャとファンクション] フォルダに格納されます。

Sybase Central のストアド・プロシージャを開くと、プロシージャのテストと修正に使用する、SQL エディタも開きます。SQL エディタには、次の機能があります。

- 構文の自動強調表示
- DDL 文の自動形成
- 文字列の検索と置換、および特定の行番号へのジャンプ機能
- データベースに対して DDL コードを実行する機能。

## ストアド・プロシージャの表示

デモ・データベースには、いくつかのストアド・プロシージャが含まれています。以降の手順では、ストアド・プロシージャを表示して編集する方法について説明します。

### プロシージャの表示と編集

[プロシージャとファンクション] フォルダには、デモ・データベースに対して実行できるいくつかのストアド・プロシージャがあります。次の例では、ストアド・プロシージャを開く方法、SQL 文を確認する方法、SQL エディタに変更を保存する方法を示します。

1. プロシージャを開くには、[プロシージャとファンクション] フォルダを開き、[カスタマの表示] を選択します。

Sybase Central の右側のウィンドウ枠の SQL エディタにプロシージャが表示されます。SQL エディタの任意の場所を右クリックすると、標準の編集オプションが表示されます。また、ツールバーの編集オプションを使用することもできます。

2. 次のいずれかを行います。
  - ツールバーにある [保存] ボタンをクリックします。
  - [ファイル] をクリックし、[保存] を選択します。

### プロシージャ・パーミッションの表示と変更

ストアド・プロシージャを実行するには、適切なパーミッションが与えられているユーザまたはグループのメンバである必要があります。次の例では、ユーザ・

パーミッションとグループ・パーミッションを確認および付与する方法を示します。

1. [プロシージャとファンクション] フォルダでプロシージャを右クリックし、[プロパティ] を選択します。
2. [パーミッション] タブをクリックし、このプロシージャに対するパーミッションを持っているユーザを表示します。
3. [付与] をクリックし、このプロシージャを実行するための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、[OK] をクリックします。
4. [適用] をクリックします。

### ストアド・プロシージャの実行

Sybase Central で、ストアド・プロシージャを起動および実行できます。

ストアド・プロシージャを実行するには、[フォルダ] ビューに [プロシージャとファンクション] フォルダを開き、実行するプロシージャを右クリックして表示されるメニューから [Interactive SQL から実行] を選択します。

## ユーザとグループの管理

---

ユーザとグループには、ユーザがデータベースに接続するときに適用されるログイン・ポリシーによる制御も行われます。

すべての新しいデータベースには、ルート・ログイン・ポリシーが含まれています。ルート・ログイン・ポリシーの値を変更することはできませんが、ポリシーは削除できません。ログイン・ポリシーは、ユーザ・ログインのルールのみを管理し、権限およびパーミッションとは切り離されています。ログイン・ポリシーは、グループ・メンバシップを通して継承されません。

次の設定は、ログイン・ポリシーによって管理されます。

- パスワードの有効期間
- パスワードの猶予期間
- 次回ログイン時にパスワードを失効
- ロック
- 最大接続数
- 失敗ログインの最大試行回数
- ログインからの最大経過日数
- 非 DBA の最大接続数

---

**注意：** ログイン・ポリシーの詳細については、Sybase IQ のオンライン・ヘルプの「ユーザとグループの管理」を参照してください。

---

## 新しいログイン・ポリシーの追加

ログイン・ポリシー作成ウィザードを使用して、新しいログイン・ポリシーを作成します。

1. [ログイン・ポリシー] を右クリックし、[新規] をポイントし [ログイン・ポリシー] を選択します。
2. 新しいログイン・ポリシーに Sales access と名前を付け、[次へ] をクリックします。
3. [ポリシー・オプション] ダイアログで、適切なポリシー・オプションを設定し、[次へ] をクリックすると [コメント] ダイアログが表示されます。
4. [コメント] ダイアログに適切なコメントを入力し、[完了] をクリックします。

## データベースへのグループの追加

グループ・ウィザードを使用して、Sales access のログイン・ポリシーのある Sales グループを、デモ・データベースに追加します。

1. [ユーザとグループ] を右クリックし、[新規] をポイントし、[グループ] を選択します。
2. 新しいグループに Sales という名前を付けて、[次へ] をクリックします。
3. [パスワードの有効化] ボックスをオンにします。
4. [パスワード] ボックスに適切なパスワードを入力してから、[パスワードの確認] ボックスにもう一度同じパスワードを入力し、[次へ] をクリックします。

[次回ログイン時に新規パスワードの指定が必要] ボックスをオンにすると、ログインを最初に行ったグループ・メンバに次回からのログインに使用する新しいパスワードを選択するように要求するメッセージが表示されます。

5. グループ・ログイン・ポリシーとして [Sales access] を選択し、[次へ] をクリックします。
6. このグループに割り当てる権限を選択し、[次へ] をクリックします。
7. [コメント] ボックスに適切なコメントを入力し、[完了] をクリックします。

## デモ・データベースへのユーザの追加

ユーザ作成ウィザードを使用して、Sales access パーミッションを持つ新しい販売担当者を追加します。

1. [ユーザとグループ] を右クリックし、[新規] をポイントし、[ユーザ] を選択します。

2. 新しいグループに "Sales person" という名前を付けて、[次へ] をクリックします。
3. [パスワードの有効化] ボックスをオンにします。
4. [パスワード] ボックスに適切なパスワードを入力してから、[パスワードの確認] ボックスにもう一度同じパスワードを入力し、[次へ] をクリックします。

[次回ログイン時に新規パスワードの指定が必要] ボックスをオンにすると、そのユーザの最初のログイン時に新しいパスワードを選択するように要求するメッセージが表示されます。

5. ログイン・ポリシーとして [Sales access] を選択し、[次へ] をクリックします。
6. このユーザに割り当てる権限を選択し、[次へ] をクリックします。

詳細については、『SQL Anywhere サーバ - データベース管理ガイド』>「データベースの設定」>「ユーザ ID、権限、パーミッションの管理」>「データベースのパーミッションと権限の概要」を参照してください。

7. [コメント] ボックスに適切なコメントを入力し、[完了] をクリックします。

## デモ・データベースの元の状態へのリストア

デモ・データベースをリストアするには、Sales グループとユーザ Sales person を削除します。

1. [ユーザとグループ] をクリックします。
2. 右側のウィンドウ枠で、削除するユーザまたはグループをクリックし、[削除] を選択します。

## ユーザ接続の管理

接続しているユーザは、サーバによって割り当てられている接続 ID で識別されません。

1. 右側のウィンドウ枠で、[接続しているユーザ] タブをクリックします。
2. 次のいずれかを行います。

作業内容	手順
接続しているユーザの表示	右側のウィンドウ枠で、[接続しているユーザ] タブをクリックします。  [接続しているユーザ] タブには、すべての接続中のユーザがリストされています。各ユーザは、サーバによって割り当てられている接続 ID で識別されます。
ユーザの切断	切断するユーザの [接続 ID] を右クリックし、[切断] を選択します。

## データベースの管理

作業内容	手順
現在の接続の終了	次のいずれかを実行します。 <ul style="list-style-type: none"><li>• ツールバーにある [切断] ボタンをクリックします。</li><li>• [接続] をクリックし、[切断] を選択します。</li></ul>

## インデックス機能とデータのロード

Sybase IQ のカラムベースのアーキテクチャによって、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選択してください。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。

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、B-tree インデックスという方法を使用してデータ・レコードを示しています。B-tree が役に立つのは、一意のデータ値を数多く使用してフィルタリングし、非常に少数のレコードのセットまで絞り込む場合に限られます。たとえば、トランザクション処理システムの受注番号や顧客名のカラムなどで使われます。

Sybase IQ のインデックスは実際に、データをさまざまなクエリ的高速化に使用できるように表現し、格納します。これは、データ・ウェアハウス環境、つまり一意の値が比較的少ない膨大なレコード群をクエリで検索し、集約関数の結果を求めるような環境に合わせて設計された方式です。

### カラム・インデックスの作成

---

テーブルを作成し、そのテーブルにカラムを指定すると、Sybase IQ によって、クエリ処理を最適化するために、特定のデフォルトの記憶領域構造体が自動的に作成されます。実行を予定しているクエリの種類が分かっている場合は、任意のカラムに複数のインデックスを追加できます。

必要なすべてのインデックスを作成してからデータベースにデータを挿入することで、操作に要する時間を短縮できます。オプションのインデックスは、不要になった場合は後から削除できます。

---

**警告！** そのインデックスが重要なものでないことを確認してから削除してください。同様なクエリであっても、クエリごとに異なるインデックスを使用します。間違ったインデックスを削除すると、パフォーマンスが悪影響を受けます。キャプチャ・クエリ、実行時間、クエリ・プランによって、必要なインデックスが決まります。

---

## コラム・インデックスのタイプ

各タイプのコラム・インデックスは、特定の種類のデータ処理を高速化するように設計されています。

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常に使用されます。クエリ・オプティマイザがコラムで使用するための理想的なインデックス・タイプが作成されなかった場合でも、Sybase IQ はそのコラムが含まれるクエリを解析できますが、正しいインデックス・タイプを使用した場合に比べて応答時間が遅くなる可能性があります。

1つのコラムまたは一連のコラムを **PRIMARY KEY** または **UNIQUE** として指定すると、そのコラムに対する High\_Group インデックスが Sybase IQ によって自動的に作成されます。テーブルのすべての **UNIQUE** 制約から 1つの **PRIMARY KEY** を選択します。**PRIMARY KEY** と **UNIQUE** のどちらの制約でも、NULL 値は使用できません。ただし、ユニーク・インデックスでは、NULL 値を使用できます。**UNIQUE** 制約は、クエリ・オプティマイザに対して、コラム制約に関するヒントも提供します。『システム管理ガイド：第1巻』では、各タイプのインデックスを使用するタイミングと、各タイプのインデックスの領域と時間に関するトレードオフについて説明しています。

これらのインデックス・タイプは、Sybase IQ データ固有のもので、SQL Anywhere テーブルには適用できません。詳細については、『システム管理ガイド：第1巻』を参照してください。

表 8 : Sybase IQ コラム・インデックスのタイプ

インデックス・タイプ	目的
Compare (CMP)	同じデータ型、精度、位取りの2つのコラムのバイナリ比較(<、>、=、<=、>=、または NE)を格納します。
DATE	<b>DATE</b> データ型のコラムのインデックス。日付量を扱うクエリを処理するときに使用されます。
Datetime (DTTM)	<b>DATETIME</b> または <b>TIMESTAMP</b> データ型のコ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 (HG)	カーディナリティの高いデータ上での演算が等号とグループを処理するために使用する改善された B-tree インデックス(1,000 個以上の重複のない値または 25,000 ロー未満のテーブルの場合に推奨)。
High_Non_Group (HNG)	範囲や集合関数を含む高カーディナリティの DSS 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・タイプ	目的
Low_Fast (LF)	カーディナリティの低いデータでクエリを処理するための値ベースのビットマップ(1,000個までの重複のない値、および25,000ローを超えるテーブルの場合に推奨。重複のない値を最大で10,000個までサポート可能)。
時刻	<b>TIME</b> 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	<b>CHAR</b> 、 <b>VARCHAR</b> 、または <b>LONG VARCHAR</b> 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。

**注意：** Sybase IQ では、**WHERE** 句と **GROUP BY** 句内にあるすべてのカラムに、LF インデックスまたは HG インデックスを追加することが前提になっています。

## カラムへの Low Fast インデックスの追加

この例では、適切なインデックス・タイプと格納領域の場所はすでに決定しています。単純に手順に従って、テーブル内のカラムにインデックスを追加してください。

この例では、適切なインデックス・タイプと格納領域の場所はすでに決定されています。単純に手順に従って、テーブル内のカラムにインデックスを追加してください。

1. *iqdemo* データベースに接続していない場合は、接続します。
2. [インデックス] を右クリックし、[新規] をポイントし [IQ テーブルのインデックス...] を選択します。
3. リストから *Customers* テーブルを選択し、新しいインデックスに *IQ\_IDX\_LF\_ID* という名前を付け、[次へ] をクリックします。
4. インデックス・タイプとして [Low Fast] オプションを選択し、[次へ] をクリックします。

このチュートリアルでは、通知を受けるまでに追加できるレコード数はデフォルトの数で十分です。通知カウントの詳細については、『システム管理ガイド：第1巻』を参照してください。

5. DB 領域のダイアログで、インデックスを格納する適切な DB 領域を選択します。このチュートリアルでは、*iq\_main* を選択し、[次へ] をクリックします。
6. [カラム] 画面で、*ID* カラムを選択し、[追加] をクリックし、[次へ] をクリックします。

7. [コメント] ボックスに `Low Fast index for id column` というコメントを入力し、[完了] をクリックしてインデックスを作成します。[Indexes] コンテナは新しいインデックスで更新されます。
8. 手順3～7を使用して、State カラムに Low Fast インデックスを追加します。

---

**注意：**

- テーブル内の残りのカラムには、オプションのインデックスは必要ありません。デフォルトの記憶領域構造体で十分です。
  - Sybase Central の代わりに **CREATE INDEX** コマンドを使用して、カラム・インデックスを作成することもできます。構文については、『リファレンス』の「文とオプション」の「CREATE INDEX 文」を参照してください。
- 

## ジョイン・インデックスの作成

---

ジョイン・インデックスは、関連テーブルのジョインを最適化する Sybase IQ の内部構造体です。

ジョイン・インデックスは、個々のカラムではなく、カラムのセットに対して作成されます。ジョイン・インデックスは、複数のテーブルの完全な外部ジョインを表します。クエリ・エンジンは、この完全な外部ジョインを、左外部、右外部、内部の各ジョインを含むクエリの開始点として使用します。ユーザがクエリを解析するために一般的にジョインする任意のカラムのセットに対して、管理者はジョイン・インデックスを作成できます。ガイドラインについては、『システム管理ガイド：第1巻』、および『パフォーマンス&チューニング・ガイド』の「テーブルのジョイン」を参照してください。

ジョイン・インデックスの作成には3つの重要なルールがあります。

- ジョイン・インデックスは、関連テーブルのカラム・インデックスに依存するため、カラムにインデックスを追加してから、ジョイン・インデックスを作成します。
- ジョイン・インデックスの作成者は、インデックスで使用されるテーブルの所有者でもある必要があります。そうでない場合、ジョイン・インデックスは失敗します。
- ジョイン・インデックス・カラムは、データ型と位取りが同じである必要があります。

---

**注意：** マルチプレックス・データベースの場合は、いくつかの追加の作業が必要です。Sybase IQ マルチプレックスにインデックスを作成する必要がある場合は、この章ではなく、「Sybase IQ Multiplex の使用」を参照してください。

---

## SalesOrders と SalesOrderItems のジョイン・インデックスの作成

この例では、ジョイン・インデックスの作成方法を示します。最初のジョイン・インデックス `so_soi_idx` は、`sales_order` テーブルと `sales_order_items` テーブルをジョインします。

1. [ジョイン・インデックス] を右クリックし、[新規] をポイントして [ジョイン・インデックス] を選択します。
2. [名前とテーブル] ダイアログで、`so_soi_idx` という名前を入力します。
3. 2つのテーブルの同じ名前 (`id`) の等価なカラムをジョインするため、[ジョイン・タイプ] ドロップダウン・リストで、[NATURAL] を選択します。
4. 左テーブルとして [SalesOrders] を選択します。
5. 右テーブルとして [SalesOrderItems] を選択します。
6. [次へ] をクリックします。Sybase IQ によってジョイン・カラムが検索されます。
7. [コメント] ボックスに、ジョイン・インデックスの説明として `Join index for SalesOrders and SalesOrderItems` と入力します。
8. [終了] をクリックします。[ジョイン・インデックス] フォルダに `so_soi_idx` ジョイン・インデックスが表示されます。

---

**注意：** NATURAL キーワードを使用してジョインする場合は、ジョインするカラムだけが同じ名前を持つように注意します。`(id=id)` など。`date` や `description` などの一般的なカラム名の場合、注意する必要があります。

---

## Department と Employee のジョイン・インデックスの作成

この例では、2つのテーブルのカラム内の等しい値を使用して、`Departments` テーブルと `Employees` テーブルの間のジョインを作成します。次の手順に従って、`dept_emp_idx` という名前のジョインされているインデックスを作成します。

1. [ジョイン・インデックス] を右クリックし、[新規] をポイントして [ジョイン・インデックス] を選択します。
2. [名前] ボックスに、`dept_emp_idx` という新しいジョイン・インデックスの名前を入力します。
3. [ジョイン・タイプ] ドロップダウン・ボックスで、[ON] を選択します。2つのテーブルにある、同じ名前 (`dept_id`) を持つ等価なカラムをジョインするには、**ON** 句が必要です。
4. 左テーブルとして [Departments] を選択します。

5. 右テーブルとして [Employees] を選択します。
6. [次へ] をクリックします。
7. [インデックス・カラムを選択] ダイアログで、[左テーブル・カラム] として [DepartmentID] を選択し、[右テーブル・カラム] として [DepartmentID] を選択します。
8. [追加] をクリックします。2つのカラムを追加すると、[ジョイン・インデックス・カラム] ボックスに `GROUP0.Departments.DepartmentID=GROUP0.employees.DepartmentID` と表示されます。(ジョイン全体を読むには、スクロールする必要がある場合もあります)。
9. [次へ] をクリックします。
10. [コメントの追加] 画面で、ジョイン・インデックスの説明として `Join index for Departments and Employees` と入力します。[終了] をクリックします。
11. [ジョイン・インデックス] フォルダに `dept_emp_jdx` ジョイン・インデックスが表示されます。

ジョイン・インデックスの計画と作成の詳細については、『システム管理ガイド：第1巻』を参照してください。

### ジョイン・インデックス・データの更新

テーブルをロードした後でジョイン・インデックスを同期します。

インデックスによって参照されるテーブルが更新されている間、他のユーザがジョイン・インデックスに書き込むことはできません。テーブルをロードしてから、そのテーブルを使用してジョイン・インデックスを作成する場合は、テーブルをロードした後でジョイン・インデックスを同期させる必要があります。

Sybase Central では、現在同期操作はサポートされていません。**SYNCHRONIZE JOIN INDEX** コマンドを使用してください。

### データのロード

Interactive SQL または Sybase Central を使用して、テーブルをロードします。

#### *Interactive SQL*

Interactive SQL を使ってテーブルをロードするには、次の手順に従います。

- **INSERT FROM SELECT** データベース、または接続しているデータベースから情報がロードされます。

- **INSERT VALUES** では、データを "手動で" 指定済みのテーブル・カラムに挿入できます。
- **LOAD** では、ファイルから ASCII データとバイナリ・データが挿入され、バックアップやアップグレードからバイナリ・データが挿入されます。

**INSERT** コマンドと **LOAD** コマンドの構文および例については、『リファレンス』の「文とオプション」を参照してください。

### *Sybase Central*

Sybase Central を使用してテーブルにローを追加するには、次の手順に従います。

- [フォルダ] ビューで [テーブル] フォルダを開き、更新するテーブルをクリックしてから右側のウィンドウ枠の [データ] タブをクリックします。ローを右クリックし、[追加] を選択します。
- [タスク] ビューで、[テーブル] フォルダをダブルクリックし、更新するテーブルを選択してから [データ] タブをクリックします。ローを右クリックし、[追加] を選択します。



## DB 領域の管理

Sybase IQ では、各デバイスを DB 領域として表現することで、ユーザ・データをアプリケーションのレベルで複数のディスクに分散します。DB 領域は、オペレーティング・システム・ファイル (dbfile) でも、ロー・ディスク・パーティションでもかまいません。DB 領域には、ユーザ・データと、起動、リカバリ、バックアップ、トランザクション管理に使用される内部データベース構造体の両方が格納できます。

「ストア」は、永続的データまたはテンポラリ・データを特別な目的のために格納する 1 つまたは複数の DB 領域です。Sybase IQ には 4 つのストアがあります。

- カタログ・ストアには、SYSTEM DB 領域と、その他に最大 12 個のカタログ DB 領域があります。
- Sybase IQ メイン・ストアには、IQ\_SYSTEM\_MAIN DB 領域およびその他のユーザ DB 領域があります。
- Sybase IQ テンポラリ・ストアには、IQ\_SYSTEM\_TEMP DB 領域があります。
- Sybase IQ 共有テンポラリ・ストアには、IQ\_SHARED\_TEMP DB 領域があります。

DB 領域の詳細については、『システム管理ガイド：第 1 巻』の「データベース・オブジェクトの使用」を参照してください。

## DB 領域の作成

---

DB 領域は 1 つ以上のファイルから構成されています。DB ファイルの総数は 16383 です。DB 領域を作成するマシン上で Sybase IQ Agent が実行されている必要があります。

1. [DB 領域] を右クリックし、[新規] をポイントし、[DB 領域] を選択します。
2. 名前、記憶領域の場所、ストライピング・オプションなど、作成する DB 領域について説明します。

ストライピング・オプションの詳細については、『リファレンス』の「文とオプション」の「ALTER DBSPACE 文」を参照してください。

3. [次へ] をクリックします。
4. [追加] をクリックします。
5. [ファイルの詳細] ダイアログで、新しいファイルのプロパティを次のように指定します。

オプション	説明
ファイルの論理名の指定	ファイルの論理名を指定します。 ファイルの論理名は DB 領域の実際のファイル名であり、必要に応じてパスを指定します。相対ディレクトリは最初の DB 領域に相対となります。ファイル名には、Sybase IQ メイン・ストアの場合は .iq、Sybase IQ テンポラリ・ストアの場合は .iqtmp というサフィックスを必ず指定するようにしてください。
ディスク上の物理ファイルへのパスを指定	ファイル・ロケーションへのパスを指定します。 明示的なパス参照がないファイル名は、そのストアの最初の DB 領域と同じディレクトリに作成されます。
これはロー・デバイス	DB 領域が dbfile の場合、This is a raw device のオプションをオフにします。オフにしたら、ファイル・サイズを、キロバイト (KB)、メガバイト (MB)、ギガバイト (GB)、またはテラバイト (TB) 単位で指定します。
予約サイズ...メガバイト	予約領域のサイズ (メガバイト単位) です。予約領域によって、将来 DB 領域のサイズを増加できます。サイズには、0 以上の任意の値を指定できます。DB 領域を作成した後に、予約サイズを変更することはできません。 予約サイズを指定すると、データベースは、内部 (フリー・リスト) 構造体に対して、より多くの領域を使用します。

6. [OK] をクリックします。
7. 次のいずれかをクリックします。
  - 別のファイルを DB 領域に追加します。
  - [完了] をクリックしてウィザードを閉じます。

### 次のステップ

データベースの作成直後に、2 番目のメイン DB 領域 (ユーザ・メイン DB 領域) を作成し、デフォルトの DB 領域として指定することをおすすめします。この作成と設定は、管理者が行います。ユーザが、IQ\_SYSTEM\_MAIN にテーブルやインデックスを配置しないようにするために、PUBLIC の IQ\_SYSTEM\_MAIN での CREATE 権限を取り消し、この新しいユーザ・メイン DB 領域に付与します。データベースの作成時に、必要なすべての DB 領域を作成するのが最良の方法ですが、Sybase IQ では、データベース管理者が、いつでも新しい DB 領域を追加して、データベースで使用可能な領域を増やすことができます。

DB 領域を作成すれば、ファイルを追加または削除できます。どの DB 領域にも複数のファイルを追加できますが、複数の DB 領域の追加先はメイン・ストアだけになります。

## DB 領域の変更

---

DB 領域を変更するには、DB 領域を右クリックし、[プロパティ] を選択します。

DB 領域のプロパティを変更するには、DBA 権限が必要です。

モードを変更するには、[モード] サブメニューでモードを選択します。

- 読み込み書き込み – 新規 DB 領域のデフォルトのモード。DB 領域への書き込みが許可されています。
- 読み込み専用 – ステータス変更時にアクティブなトランザクションの場合、エラーおよびロールバックが発生する可能性があります。



# データベースの更新

次の演習では、ローの挿入方法、変更のキャンセルと確認方法、データの削除方法を説明します。

## テーブルへのローの追加

**INSERT** 文を使用して、新しい部署に関するデータを追加します。

東部営業部 (Eastern Sales) を新しく作成し、そのマネージャは、現在の営業部 (Sales) のマネージャが務めるものとしします。Interactive SQL で次の INSERT 文を使用して、データベースにこの情報を追加できます。

```
INSERT
INTO Departments ( DepartmentID, DepartmentName, DepartmentHeadID )
VALUES ( 220, 'Eastern Sales', 902 )
```

誤りがあったり、指定し忘れたカラムがあった場合、Sybase IQ は次のエラーを返します。

```
Error at line 1
The number of insert VALUES does not match the column list.
```

**注意：** 値を挿入する場合は、大文字と小文字の区別が問題になることがあります。

### NULL

NULL 値は、不明または適用できないものを示す特殊な値です。ただし、NULL 値はすべてのデータ型で使用できます。NULL 値には、次のように個別で性質も異なる使用方法があります。

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

NULL 値が許可されるカラムと、許可されないカラムがあります。テーブル内のカラムが NULL 値を使用できるかどうかを調べるには、Interactive SQL で **sp\_iqcolumn** ストアド・プロシージャを使用します。たとえば、次に示すように入力します。

```
sp_iqcolumn Employees
```

*nulls* カラムの "Y" 値は、null 値を使用できることを示しています。

### *INSERT の省略形*

省略形を使用して、テーブル内のすべてのカラムに、テーブルに対して **SELECT \*** を実行するときカラムが表示される順序 (カラムが作成された順序) で値を入力できます。次のコマンドは、前の **INSERT** コマンドと同じです。

```
INSERT  
INTO Departments  
VALUES ( 220, 'Eastern Sales', 902 )
```

### *変更のキャンセル*

**ROLLBACK** 文は、最後に変更を確定した後にデータベースに対して行われたすべての変更を取り消します。

## 永続的な変更

---

**COMMIT** 文を使用して、すべての変更を永続的なものにします。

**DBISQL** のデフォルト動作では、終了時に **COMMIT** を実行します。これは、**dbisql** オプションの **COMMIT\_ON\_EXIT** で制御できます。

**dbisql** には、**AUTO\_COMMIT** という別のオプションがあります。このオプションがオンの場合、**DBISQL** はコマンドを実行するたびに **COMMIT** 操作を実行します。このオプションのデフォルトは **OFF** です。削除を実行して意図しない結果になった場合などに、変更に対して **ROLLBACK** を実行する機会が得られるように、通常は、このオプションを **OFF** にしておくことをおすすめします。

**dbisql** 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。

---

**注意：** **COMMIT** は注意して使用してください。このチュートリアル of 例を実行する場合は、データベースを永久に変更してもよいことを確認するまで、変更をコミットしないように注意してください。

**Sybase IQ** は 1 回のトランザクションで大量の変更を実行することに優れています。トランザクションあたりのオーバーヘッドは、従来の **OLTP** データベースより高くなります (文のレベルでも同様です)。つまり、**COMMIT** 文は、**OLTP** システムより完了に時間がかかる場合がありますが、データ・ウェアハウス環境では、コミットの回数は **OLTP** システムより少ない傾向があります。

頻繁に **COMMIT** 操作を実行すると、オーバーヘッドが増え、システムのパフォーマンスに影響します。

---

## データの一貫性の確保

---

Sybase IQ は、データ・ウェアハウス環境でデータの一貫性を確保する特別な機能を備えています。

データ・ウェアハウス環境では、多くのユーザがデータベースから読み込みを行います。データベースの更新を行う必要があるのは DBA のみです。しかし、他のユーザがクエリの結果を要求し受信し続けている場合は、頻繁に更新を行う必要があります。Sybase IQ は、「テーブル・レベルのスナップショットのバージョン管理」を使用して、データベースの変更を追跡します。これにより、ユーザが書き込みトランザクションを開始するときのテーブルの状態が記録されます。

### データの一貫性

Sybase IQ では、複数のユーザがテーブルを読み込みますが、書き込めるのは 1 人だけです。データの一貫性をテストするには、2 人の別々のユーザとして接続し、両方の接続から同時にテーブルへの書き込みを行います。1 つの文はロールバックされ、エラー・メッセージが返されますが、もう 1 つの文はコミットされます。

### チェックポイント

トランザクションがコミットするたびに、Sybase IQ は、更新されたデータ・ページをディスクに書き込みます。この方法は、1 つのアプリケーションで数百万ローモのデータを書き込む、データ・ウェアハウスにとって最適です。「チェックポイント」は、データベースがディスクに情報を書き込む場合の、トランザクション内のポイントです。ほとんどの OLTP データベースは、チェックポイントでデータをディスクに書き込みます。Sybase IQ は、チェックポイントを待たずに物理データを書き込みます。Sybase IQ は、内部追跡用に、ディスクに特定の情報を書き込むためにチェックポイントを使用します。この情報は、データベースをリカバリする必要がある場合に使用されます。チェックポイントは明示的に設定できますが、ほとんどの Sybase IQ チェックポイントは自動的に発生します。

## セーブポイント

---

Sybase IQ がディスクに書き込むデータの量を制限するには、セーブポイントを設定します。セーブポイントは、以降のすべての変更を ROLLBACK TO SAVEPOINT 文で取り消すことができる、トランザクション内のポイントを定義します。これによって、トランザクション全体が終了する前にデータをコミットできます。

## データ一貫性のテスト

この例では、データの一貫性をテストする方法を示します。

1. Interactive SQL で接続名を使用して次のようにデータベースに接続します。

```
CONNECT DATABASE iqdemo  
AS sales
```

2. 読み込み／書き込みのトランザクションを次のように開始します。

```
INSERT INTO FinancialCodes  
(Code, Type, Description)  
VALUES ('e6', 'expense', 'Services')
```

3. 別の接続名で同じデータベースに接続し、次のようにクエリを実行します。

```
CONNECT DATABASE iqdemo  
AS marketing;  
SELECT CustomerID, OrderDate  
FROM SalesOrder
```

Sybase IQ は、要求された情報を返します。これは、読み込み専用のトランザクションです。

4. 別の接続名を使用して、同じデータベースに次のように接続します。

```
CONNECT DATABASE iqdemo  
AS accounting
```

5. 読み込み／書き込みのトランザクションを次のように開始します。

```
INSERT INTO FinancialCodes  
(Code, Type, Description)  
VALUES ('r3', 'revenue', 'Sales & Marketing')
```

このコマンドは失敗し、ロールバックされます。

6. ユーザ sales として、トランザクションをコミットします。
7. ユーザ accounting として、トランザクションを再実行します。このトランザクションは、コミットまたはロールバックできます。

2つの類似したトランザクションを実行した時点に戻ることができるように、セーブポイントをいくつか設定します。ここでは、1番目のトランザクションが終了するまで、2番目のトランザクションを保留します。

## セーブポイントの設定

セーブポイントを使用して関連する文をグループ分けすると、トランザクション内の重要な状態を識別し、そこに戻ることができます。

1. データベースに接続します。

```
CONNECT DATABASE iqdemo
```

- 読み込み／書き込みのトランザクションを次のように開始します。

```
INSERT INTO FinancialCodes
(Code, Type, Description)
VALUES ('e8', 'expense', 'Services')
```

- セーブポイントを次のように設定します。

```
SAVEPOINT TUES9_45
```

オプションで、セーブポイントに名前を付けることができます (以下を参照)。『リファレンス』の「文とオプション」に記載されているオブジェクト名の規則に従ってください。

- 読み込み／書き込みのトランザクションを次のように開始します。

```
INSERT INTO FinancialCodes
(Code, Type, Description)
VALUES ('r3', 'fees', 'Administration')
COMMIT
```

この insert コマンドが失敗すると、トランザクションはセーブポイント `TUES9_45` にロールバックされます。手順 2 で挿入されたデータは失われません。

## 複数のアカウントによる複数のテーブルの更新

1つのデータベース内の複数のテーブルを操作している場合、書き込みを行うユーザが異なるテーブルに書き込んでいる限り、Sybase IQ では、1つのデータベースに複数のユーザが読み込みと書き込みを行うことができます。

- 接続名 `marketing` を使用して、デモ・データベースに接続します。

```
CONNECT DATABASE iqdemo
AS marketing
```

- 読み込み／書き込みのトランザクションを次のように開始します。

```
INSERT INTO FinancialCodes
(Code, Type, Description)
VALUES ('e9', 'expense', 'R & D')
```

- 別の接続名を使用して、同じデータベースに次のように接続します。

```
CONNECT DATABASE iqdemo
AS sales
```

- 別のテーブルを使用して、読み込み／書き込みのトランザクションを開始します。

```
INSERT INTO SalesOrders
(ID, CustomerID, OrderDate, SalesRepresentative)
VALUES ('2088', '140', '05-29-98', '195')
COMMIT
```

両方のテーブルとも `iqdemo` データベース内にありますが、挿入操作は別々のテーブルに影響するため、このトランザクションは直ちにコミットされます。実

行可能な変換の詳細については、『システム管理ガイド：第1巻』の「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。

## ローの削除

---

テーブルからローを削除する必要が生じる場合があります。

Rodrigo Guevara (従業員 ID 249) が会社を辞めたとします。次の文では、Rodrigo Guevara が employee テーブルから削除されます。

```
DELETE FROM Employees WHERE EmployeeID = 249
```

### *DELETE コマンドの使用*

1つのコマンドで複数のローを削除できます。たとえば、次の文では、termination\_date カラムが NULL でないすべての従業員が employee テーブルから削除されます。

```
DELETE FROM Employees WHERE TerminationDate IS NOT NULL
```

どの従業員も termination\_date カラムが NULL であるため、この文を実行しても、データベースから従業員は削除されません。

DELETE を使用して、必要に応じて検索条件を複雑にすることができます。たとえば、employee テーブルを再編成する場合、次の文を実行すると、市外局番が 617 で、902 というマネージャの従業員 ID を持つすべての従業員が、employee テーブルから削除されます。この WHERE 句は、関数 (LEFT) を含む複合検索条件です。

```
DELETE FROM Employees WHERE LEFT(Phone, 3 ) = '617' AND ManagerID = 902
```

データベースに対して行った変更を保持しない場合は、次に示すように、変更を取り消してください。

```
ROLLBACK
```

## ローの更新

---

特定のカラムの値に基づいて、特定のローを変更する場合があります。SET 句は、更新するカラムとその新しい値を指定し、WHERE 句は更新するローを指定します。

たとえば、市外局番が "508" から "978" に変更された場合、古い市外局番が登録されているすべてのローを、次のコマンドを入力することで新しい市外局番に変更できます。

```
UPDATE Customers SET LEFT(Phone,3) = '978' WHERE LEFT (Phone, 3) = '508'
```

## ジョイン・インデックスの同期

ジョイン・インデックスに関係するテーブルの更新中は、他のユーザはそのジョイン・インデックスに書き込むことができません。Sybase IQ では、ジョイン・インデックスを更新するタイミングを制御できます。これは、インデックスが参照するさまざまなテーブルの更新の順序によっては、更新に時間がかかる場合があります。

**SYNCHRONIZE JOIN INDEX** 文は、ユーザが接続している現在のデータベース内の、1つまたはすべてのジョイン・インデックスを更新します。インデックスを引数として指定しない場合、現在のデータベース内のすべてのジョイン・インデックスが同期します。

ジョイン・インデックスを更新するには、Sybase IQ は、ジョイン内のテーブルを参照する他のすべてのジョイン・インデックスと、そのジョイン・インデックス内のテーブルを参照するすべてのジョイン・インデックスを更新する必要があります。

**INSERT** 文または **DELETE** 文のグループを終了する前にこのコマンドを発行すると、コマンドが終了すると同時に、ユーザはデータを使用できます。マシンの負荷が軽くて、システムの処理速度が低下しないと分かっている場合は、バッチ・ジョブを実行してコマンドを発行することもできます。各 **SYNCHRONIZE JOIN INDEX** 文の後に、**COMMIT** 文を発行する必要があります。

Employees/SalesOrders ジョイン・インデックス (EMP\_SO\_JDX) を同期するには、次のコマンドを入力します。

```
SYNCHRONIZE JOIN INDEX EMP_SO_JDX COMMIT
```

**注意：**最適なパフォーマンスを実現するには、挿入や削除のたびに、**SYNCHRONIZE JOIN INDEX** コマンドを発行しないようにします。

## DB 領域、テーブル、インデックスの削除

Sybase Central または Interactive SQL を使って、データベース、DB 領域、テーブル、インデックス(ジョイン・インデックスを含む)を削除できます。

次のいずれかを行います。

- Sybase Central で、オブジェクトを右クリックし、[削除] を選択します。
- Interactive SQL で、**DROP** コマンドを使用します。完全な構文については、『リファレンス』の「文とオプション」を参照してください。



## ビューの使用

ビューとは、SQL クエリとして表現されるビュー定義の結果セットによって定義された計算テーブルです。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。

レギュラー・ビューは、ユーザがビューを参照するたびに再計算され、結果セットがディスクに格納されないビューのことです。このビューが、最も一般的に使用されるタイプのビューです。

マテリアライズド・ビューは、結果セットが事前計算され、ベース・テーブルの内容と類似する、ディスク上で実体化されたビューを示します。マテリアライズド・ビューは、実際のベース・テーブルへの頻繁なクエリによって非常に負荷がかかる場合に、データ・ウェアハウジング・シナリオで役に立つビューです。この項の例は、レギュラー・ビューを使用しています。

---

**注意：** マテリアライズド・ビューは、Sybase IQ カタログ・ストアの SQL Anywhere テーブルでのみサポートされます。

---

## ビューの定義

頻繁に使うクエリのビューを定義します。

部署別の従業員のリストを頻繁に作成する必要があるとします。iqdemo に対して実行できる次のクエリでは、部署別の全従業員が州 (State) 別にソートされてリストされます。

```
SELECT Departments.DepartmentID,
       Departments.DepartmentName, Employees.EmployeeID,
       Employees.GivenName, Employees.Surname,
       Employees.Phone
FROM Departments, Employees
ORDER BY Employees.State
```

このクエリを実行すると、次のような結果セットが作成されます。

DepartmentID	DepartmentName	EmployeeID	GivenName	Surname	Phone
100	R & D	148	Julie	Jordan	6175557835
200	Sales	148	Julie	Jordan	6175557835
300	Finance	148	Julie	Jordan	6175557835
400	Marketing	148	Julie	Jordan	6175557835

DepartmentID	DepartmentName	EmployeeID	GivenName	Surname	Phone
---	---	---	---	---	---

### ビューの作成

Interactive SQL では、CREATE VIEW 文を使用すると、ビューを作成および格納できます。

この例では、emp\_dept という名前のビューを iqdemo 内に作成します。

```
CREATE VIEW emp_dept AS
  SELECT Departments.DepartmentID,
     Departments.DepartmentName, Employees.EmployeeID,
     Employees.GivenName, Employees.Surname,
     Employees.Phone
FROM Departments, Employees
ORDER BY Employees.State
```

Sybase Central では、ウィザードを使用してビューを作成できます。ウィザードを起動するには、iqdemo に接続した後で、[ビュー] を右クリックし、[新規] をポイントし [ビュー] を選択します。ウィザードが起動したら、画面に表示される手順に従ってください。

### ビュー・データの表示

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ないことに注意してください。ビューを参照するたびに、SQL は関連する **SELECT** 文を実行して該当するデータを検索します。つまり、販売データが変更された場合、ビュー内の情報は自動的に最新の情報に更新されることを意味します。

ビューは、次のようにテーブルと同様にクエリを実行できます。

```
SELECT * FROM emp_dept
```

Sybase Central で結果を表示するには、[ビュー] を右クリックし、[Interactive SQL によるデータ表示] を選択します。

### ビューの変更

Interactive SQL では、ALTER VIEW 文を使用するとビュー定義を変更されたバージョンに変更できます。

emp\_dept ビューには、全従業員が部署ごとにリストされています。次のコマンドを使用すると、このビューを営業部門の従業員のみが表示されるように変更できます。

```
ALTER VIEW emp_dept AS
  SELECT Departments.DepartmentID,
     Departments.DepartmentName, Employees.EmployeeID,
     Employees.GivenName, Employees.Surname,
```

```

Employees.Street, Employees.City,
Employees.State, Employees.Phone
FROM Departments, Employees
WHERE Departments.DepartmentID = 200
ORDER BY Employees.State

```

ALTER VIEW は、ビュー内の既存の SELECT 文を ALTER VIEW コマンド内の SELECT で置換します。ビューに対する既存のパーミッションはそのまま維持されます。

Sybase Central でビューを変更するには、ビューを右クリックし、[新しいウィンドウで編集] を選択します。

## ビューの削除

ビューの削除には、Interactive SQL または Sybase Central を使用できます。

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。

```
DROP VIEW <view name>
```

Sybase Central でビューを削除するには、削除するビューを右クリックし、ショートカット・メニューから [削除] を選択します。

## ビューの制限と高度な使用

ビューは、仮想テーブルと考えることができます。ビュー定義では、**ORDER BY** を含むコマンド以外のすべての **SELECT** コマンドを使用できます。

ビューでは、**GROUP BY** 句、サブクエリ、ジョインを使用できます。**ORDER BY** を使用できないのは、リレーショナル・データベース内のテーブルのローを特定の順序で保存できないことと同じです。ビューを使用する場合は、**ORDER BY** を指定できます。

さらに複雑なクエリでも、ビューを使用できます。

```

CREATE VIEW Q1_Revenue AS
SELECT Products.Name AS Item, Products.Description AS Style,
SUM(SalesOrderItems.Quantity) AS Sales, Products.UnitPrice,
SUM(SalesOrderItems.Quantity * Products.UnitPrice) AS Revenue FROM
EmployeesJOIN SalesOrders ON SalesOrders.SalesRepresentative =
Employees.EmployeeID
JOIN SalesOrderItems ON SalesOrderItems.ID =
SalesOrders.ID
JOIN Products ON Products.ID =
SalesOrderItems.ProductID WHERE SalesOrders.OrderDate
>='2001-01-01' AND SalesOrders.OrderDate <='2001-04-30' GROUP BY
Products.Description, Products.Name, Products.UnitPrice

```

このクエリを実行すると、次のような結果セットが作成されます。

項目	Style	Sales	Unit Price	Revenue
Tee Shirt	Tank Top	744	9.00	6696.00
Sweatshirt	Hooded Sweatshirt	756	24.00	18144.00
Visor	Cloth Visor	756	7.00	5292.00
Baseball Cap	Wool cap	685	10.00	6850.00
Tee Shirt	Crew Neck	672	14.00	9408.00
Shorts	Cotton Shorts	1524	15.00	22860.00
Baseball Cap	Cotton Cap	900	9.00	8100.00
Tee Shirt	V-neck	780	14.00	10920.00
Sweatshirt	Zippered Sweatshirt	564	24.00	13536.00
Visor	Plastic Visor	684	7.00	4788.00

---

**注意：** ジョイン・インデックスではビューを使用できません。

---

## セキュリティを目的とするビューの使用

---

ビューを使用して、データベース内の情報へのアクセスを制限できます。

### シナリオ

営業部長の Moira Kelly のユーザ ID を作成し、そのユーザ ID を、営業部の従業員に関する情報のみ調査できるように制限しようと思います。

### 新しいユーザ ID の作成

最初に、Moira Kelly の新しいユーザ ID を作成します。作成には、**GRANT** 文を使用します。デモ・データベースに **dba** として接続した状態で、dbisql から次のコマンドを入力します。

```
GRANT CONNECT TO M_Kelly
IDENTIFIED BY SalesHead
```

### パーミッションの付与

次に、ユーザ M\_Kelly に、営業部の従業員を参照する権限を付与する必要があります。

```
CREATE VIEW SalesEmployee AS
SELECT EmployeeID, Surname, GivenName
FROM Employees
WHERE DepartmentID=200
```

次のコマンドを入力して、M\_Kelly に新しいビューを参照するパーミッションを付与します。

```
GRANT SELECT ON SalesEmployee TO M_Kelly
```

### ビューの参照

M\_Kelly としてデータベースに接続し、ビューの参照を試みます。

```
CONNECT USER M_Kelly IDENTIFIED BY SalesHead;
SELECT * FROM "dba".SalesEmployee
```

emp_id	emp_lname	emp_fname
129	Chin	Philip
195	Dill	Marc
299	Overbey	Rollin
467	Klobucher	James
641	Powell	Thomas
...	...	...

ただし、M\_Kelly には employee テーブルと department テーブルを直接参照するパーミッションはありません。次のコマンドを実行すると、パーミッション・エラーになります。

```
SELECT * FROM Employees;
SELECT * FROM Departments
```

### セキュア・ビュー用の組み込み関数の使用

組み込みの Sybase IQ 関数を利用して、セキュア・ビューを作成することもできます。次の例では、secure\_view ビューは、secure\_table 内のレコードへのアクセスを特定のユーザに制限することを目的としています。

ビュー定義では、各ユーザのサーバ・ユーザ名を格納する、**suser\_name()** という組み込みの Sybase IQ 関数を使用します。次のビューを使用すると、ユーザが次のすべてを持っているかどうかについて、レコードを確認できます。

- 特定のサーバ・ユーザ名 (suser\_name)
- ビューで参照されるレコード以上のグレード (security\_auth\_table 内のレコードで定義)

```
CREATE VIEW secure_view AS
SELECT a.secure_name, a.security_code
FROM secure_table a, security_auth_table b
WHERE SUSER_NAME() = b.opr_name
AND a.department_id = b.department_id
AND a.grade <= b.grade
```

各ユーザの一意的 ID を格納する組み込み関数 `suser_id()` を使用して、アクセスを制限するビューを作成することもできます。

## データベース内の情報へのアクセスの制限

---

ビューを定義する場合は、ビューの **SELECT** リスト内のクエリで使用されるカラムのみを指定してください。

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

以下の場合には、クエリ内のビューのパフォーマンスが低下する可能性があります (ビューを使用しない同様のクエリと比較した場合)。

- 外部ビューを参照するクエリによって使用されない、内部ビューと外部ビューの両方の `select` リストに指定されている項目を含むネスト・ビュー。
- ビューの内部と外部の両方にジョインがあり、ビューに明示的または暗黙的な **GROUP BY** 句が含まれない場合。
- ユーザがクエリの一部をビューに指定することでクエリを分割し、複数の述部がビューの内部と外部で同じカラムを参照する場合。
- ビューの外部の述部に、ビューの `select` リスト項目への参照と、外部参照または非相関サブクエリが含まれている場合。

ビューの仕様に外部ジョインが含まれている場合は、ビューを参照するクエリ内の述部をインデックスに「プッシュダウン」することはできません (外部ジョインのセマンティックを変更してしまうため)。このため、このようなクエリは、ビューを使用しない類似のクエリより実行速度が遅い場合があります。

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

# 用語解説

## Sybase IQ の用語と概念

- **Adaptive Server Enterprise** – Adaptive Server Enterprise は、高性能なリレーショナル・データベース管理システムです。Sybase IQ を使用すると、ASE データベース内のデータを検索できます。
- **カタログ・ストア** – カタログ・ストアは、各 Sybase IQ データベースのメタデータを格納する部分です。カタログ・ストアには、SYSTEM DB 領域と最大 12 の追加のカタログ DB 領域があります。デフォルト名は dbname.db です。
- **コンポーネント統合サービス (CIS)** – コンポーネント統合サービス (CIS) を使用すると、Sybase IQ ユーザは、メインフレーム、UNIX、または Windows サーバ上にある、リレーショナル・データベースまたは非リレーショナル・データベースへ直接アクセスできます。
- **接続プロファイル** – 接続プロファイルによって、実行中の Sybase IQ サーバに接続情報が格納されます。プロファイルは、サーバへのユーザ接続を簡単化するために主に使用されます。Sybase IQ では、接続プロファイルを拡張して、サーバの起動とデータベースの作成を促進しています。
- **Dbfile** – dbfile は、Sybase IQ データベースのデータを格納するために使用される、オペレーティング・システム・ファイルです。各 dbfile には、対応する論理ファイル名と物理ファイル・パスがあります。各 DB 領域名、dbfile 名、および物理ファイル・パスは、ユニークである必要があります。dbfile 名は、DB 領域名と同じものを使用できます。

*SYSDATABASE* ビューには、使用しているデータベースにあるすべての dbfile が表示されます。表示される dbfile には、カタログ DB 領域ファイル、IQ メッセージ・ファイル、IQ のメイン DB 領域とテンポラリ DB 領域にある dbfile、トランザクション・ログ・ファイル、SA テンポラリ・ファイルなどがあります。

- **DB 領域** – DB 領域は、dbfile の論理集合です。データベースに空き領域がなくなった場合、DB 領域を追加することでデータベースを拡張できます。Sybase IQ データのディスクからの移動やディスクのオフラインは、ダウン時間なしで実行できます。
- **フリー・リスト** – フリー・リストは、DB 領域で使用されているブロックを追跡するために Sybase IQ が使用する構造体です。
- **IQ データベース** – IQ データベースは、Sybase IQ サーバを使用して作成するデータベースです。IQ データベースには、Sybase IQ のクエリ速度を利用するために、特別なインデックスが作成されています。

作成する各 IQ データベースには、IQ メイン・ストア (データ用)、カタログ・ストア (メタデータ用)、および IQ テンポラリ・ストア (テンポラリ・データ用)

の3つのストアがあります。また、IQ メッセージ・ログ・ファイルも生成されます。

- **IQ ストア** – IQ メイン・ストアは、各 Sybase IQ データベースの一部であり、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 メイン・ストアは、バックアップ・メタデータやコミットされたトランザクションのロールバック・データなどの永続的なデータベース構造体を格納しています。

ユーザ・テーブルおよびインデックスは、IQ\_SYSTEM\_MAIN に保存せず、ユーザ・メイン DB 領域と呼ばれる追加の DB 領域を作成し保存することをおすすめします。IQ メイン・ストアは、IQ ストアと呼ばれることもあります。

- **IQ メッセージ・ログ** – 最初のユーザが IQ データベースに接続するときに作成される IQ メッセージ・ログ・ファイル。このファイルのデフォルト名は dbname.iqmsg です。

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。Sybase IQ は、このファイルにエラー・メッセージやステータス・メッセージを記録し、通知メッセージを挿入します。

- **IQ テンポラリ・ストア** – IQ テンポラリ・ストアには、IQ\_SYSTEM\_TEMP DB 領域があります。IQ テンポラリ・ストアは、IQ データベースの一部で、テンポラリ・テーブルやテンポラリ・スクラッチ領域データ構造体を格納しています。

データベース・サーバは、テンポラリ・データ構造体を使用して、データのソートや処理を行います。これらのテーブル内のデータが維持されるのは、ユーザがデータベースに接続中の間のみです。

- **IQ トランザクション・ログ** – IQ トランザクション・ログには、データベースに加えられた変更が記録されます。トランザクション・ログには、バージョン情報、空き容量、およびシステム障害からリカバリするために使用できるその他の情報があります。デフォルトでは、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。この dbfile のデフォルト名は dbname.log です。

- **ジョイン・インデックス** – ジョイン・インデックスは、Sybase IQ で使用する特別な種類のインデックスです。データベース内のインデックスは、概念的には書籍のインデックスに似ています。書籍の場合、インデックスは、インデックスが付けられた用語を、その用語が出現する1つのページまたは複数のページに関連付けます。データベースの場合、インデックスは、インデックスが付けられた各カラム値を、インデックス値を含むデータのローが格納される物理位置に関連付けます。

ジョイン・インデックスを使用すると、2つ以上のテーブルをジョインするクエリの応答時間を向上できます。場合によっては、ジョイン・インデックスではなく、アドホック・クエリの方が適している場合もあります。

- **メタデータ** – メタデータは、テーブル内の各カラムのサイズやデータ型など、データベース内のデータを記述するデータです。各 Sybase IQ データベースのメタデータは、カタログ・ストアに格納されます。
- **マルチプレックス** – クラスタード・サーバ構成を使用してアプリケーション・スケーラビリティを可能にする Sybase IQ の強力な機能です。Sybase IQ のマルチプレックスでは、同時実行データ・ロードおよび共有データ・ソースに接続された独立データ処理ノードを使用したクエリが実行できます。各マルチプレックス・サーバには、独自のカタログ・ストアおよび IQ テンポラリ・ストアがあります。また、すべてのサーバは、共通 IQ ストアを共有します。詳細については、『Sybase IQ マルチプレックスの使用』を参照してください。
- **オブジェクト** – オブジェクトには、ユーザが作成するテーブル、インデックス、またはジョイン・インデックスなどがあります。オブジェクトには、ユーザの接続が切断され、サーバが再起動してもデータベース内に存在し続ける永続オブジェクトと、現在のセッション中のみデータベース内に存在するテンポラリ・オブジェクトの2種類のオブジェクトがあります。永久テーブルは、ベース・テーブルとも呼ばれます。
- **パーティション・キー** – パーティション・キーは、テーブルのパーティション分割方法を決定するテーブル作成者によって定義されているテーブル・カラムです。
- **プロキシ・テーブル** – プロキシ・テーブルは、リモート・サーバ上のテーブルにマッピングするテーブル・オブジェクトです。プロキシ・テーブル上のカラム属性とインデックス情報が、リモート・ロケーションのオブジェクトから導出されます。

プロキシ・テーブルを使用すると、複数の SQL Anywhere サーバ、ASE データベース、および Sybase 以外のデータベースでデータを検索できます。逆に、自分の Sybase IQ データベース、Anywhere データベース、ASE データベースへのクエリを実行するプロキシ・テーブルを作成することもできます。

- **範囲パーティション** – 範囲パーティションは、単一のテーブル・カラムの値に基づくテーブル行の論理サブセットです。
- **SQL Anywhere** – SQL Anywhere は、Sybase のトランザクション処理リレーショナル・データベース管理システムであり、スタンドアロンで使用することも、マルチユーザのクライアント/サーバや3層環境内のネットワーク・サーバとして使用することもできます。

SQL Anywhere は、通常のデータベース管理システムより少ないメモリ量とディスク・リソースでも使用できるように設計されています。Sybase IQ は、

SQL Anywhere Server の拡張版です。このため、Sybase IQ では、SQL Anywhere Server と同じ機能が多数サポートされています。

- **SQL Anywhere データベース** – すべての Sybase IQ データベースは、カタログ・ストア用に SQL Anywhere データベースを使用します。
- **ストア** – ストアは、永続的データまたはテンポラリ・データを特別な目的のために格納する 1 つまたは複数の DB 領域です。Sybase IQ には、カタログストア、IQ メイン・ストア、および IQ テンポラリ・ストアという 3 つのストアがあります。
- **同期コマンド** – 同期コマンドによって、ジョイン・インデックス・データが更新されます。SYNCHRONIZE JOIN INDEX 文を使って実行します。ジョイン・インデックスに関連する可能性があるテーブルを複数のユーザが更新する場合、システム管理者は Sybase IQ のジョイン・インデックスを定期的に同期させる必要があります。
- **同期** – 同期を使用すると、古くなったマルチプレックス・セカンダリ・ノード・サーバを現在の情報で更新できます。
- **テーブル・パーティション** – テーブル・パーティションは、ユーザが作成したテーブルのサブセットである行の集合です。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。
- **テーブル領域** – テーブル領域は、総格納領域の論理サブセットとして管理されている可能性のある、データベース内の格納領域の単位です。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。Sybase IQ のテーブル領域は、DB 領域と呼ばれます。

# 索引

## A

Adaptive Server Enterprise 99  
 ALTER VIEW 文 94  
 AND キーワード 41

## B

BETWEEN 条件 43  
 BLANK PADDING  
   CREATE DATABASE 60  
   OFF のサポート 60  
   ジョインへの影響 60

## C

CIS 61  
   Adaptive Server Enterprise 60  
   SQL Anywhere 60  
 COMMIT 文 86, 91  
 COMMIT\_ON\_EXIT 文 86  
 CONNECT 文 96  
 COUNT 関数 43, 44  
 CREATE DATABASE 文  
   BLANK PADDING 60  
 CREATE VIEW 文 93, 94  
 CUBE 句 48

## D

DB 領域 99  
   IQ\_SHARED\_TEMP 81  
   IQ\_SYSTEM\_MAIN 81  
   IQ\_SYSTEM\_TEMP 81  
   Sybase Central での表示 30  
   SYSTEM 81  
   管理 81  
   作成 81  
   削除 91  
   変更 83  
 dba テーブル 10  
 dbfile 99  
 dbisql

次を参照： Interactive SQL

DELETE 文 91  
 DENSE\_RANK 関数 49  
 DROP VIEW 文 95  
 DROP 文 91

## E

Enterprise Edition 1  
 Evaluation Edition 1

## F

FROM 句 37, 53  
   とジョイン 53  
 From 句のないクエリの処理 37

## G

GRANT 文 96  
 GROUP BY CUBE 句 48  
 GROUP BY ROLLUP 句 46  
 GROUP BY 句 45, 46  
 GROUPO テーブル 10

## H

High\_Group インデックス 74  
 High\_Non\_Group インデックス 74

## I

IN 条件 43  
 INSERT FROM SELECT 文 78  
 INSERT VALUES 文 78  
 INSERT 文 78, 91  
   例 85  
 Interactive SQL 7  
   データのロード 78, 85  
   起動、Windows 33  
   接続パラメータ 32

## 索引

### IQ Agent

実行 26

IQ ストア 100

IQ データベース 99

IQ テンポラリ・ストア 100

IQ トランザクション・ログ 100

IQ メッセージ・ログ 100

iq\_dummy テーブル 10, 37

IQ\_SHARED\_TEMP 81

IQ\_SYSTEM\_MAIN 81

IQ\_SYSTEM\_TEMP 81

iqdemo.db ファイル

説明 10

## J

java サンプル・アプリケーション 8

## L

LOAD 文 78

Low Fast インデックス 75

Low\_Fast インデックス 74

## M

make ファイル 8

maximum 関数 43

## N

NTILE 関数 49

NULL 値

カラム内での許可 85

## O

ODBC サンプル・アプリケーション 8

ORDER BY 句 39, 50, 53

OVER 句 49

## P

PERCENT\_RANK 関数 49

PERCENTILE\_CONT 関数 49

PERCENTILE\_DISC 関数 49

## R

RANK 関数 49

readme ファイル 8

ROLLBACK TO SAVEPOINT 文 87

ROLLBACK 文 90

ROLLUP 句 46

## S

SELECT DISTINCT 文 51

SELECT 文 35, 94

説明 62

Service Manager 8

SET 句 90

Single Application Server Edition 1

Small Business Edition 1

sp\_iqtable プロシージャ 35

SQL 16

SQL Anywhere 101, 102

SQL 文 21

ALTER VIEW 文 94

COMMIT 文 86, 91

COMMIT\_ON\_EXIT 文 86

CONNECT 文 96

CREATE DATABASE 文 60

CREATE VIEW 文 93, 94

DELETE 文 91

DROP VIEW 文 95

DROP 文 91

FROM 句 53

GRANT 文 96

INSERT FROM SELECT 文 78

INSERT VALUES 文 78

INSERT 文 78, 85, 91

LOAD 文 78

ORDER BY 句 53

ROLLBACK TO SAVEPOINT 文 87

ROLLBACK 文 85, 90

SELECT DISTINCT 文 51

SELECT 文 35, 62, 94

SYNCHRONIZE JOIN INDEX 文 91

WHERE 句 53

start\_iq 9  
 start\_iq.exe 9  
 Structured Query Language  
   次を参照： SQL  
 SUM 関数 43  
 Sybase Central 7  
   UNIX 24  
   Windows 23  
   サーバの接続 23  
   プログラム・グループ内 8  
   メイン・ウィンドウ 25  
   ロード 78  
   説明 25  
   停止 26  
 Sybase IQ 65  
   DB 領域の管理 81  
   アーキテクチャ 6  
   オプション機能 1  
   コンポーネント 7  
   サービス・マネージャ 8  
   ジョイン 53  
   ストア 81  
   データのインデックス 73  
   データのロード 73  
   データベース・サーバ 9  
   データベースの更新 85  
   デモ・データベース 9, 10  
   ビュー 93  
   マニュアル 4  
   ライセンス 1  
   概念 13  
   関連製品 3  
   製品エディション 1  
   説明 1  
   定義 1  
   利点 6  
 Sybase IQ について 1  
 Sybase 製品、関連  
   PowerDesigner 3  
   SQL Anywhere 3  
   Sybase ETL 3  
   Sybase RAP – The Trading Edition 3  
   WorkSpace 3  
 SYNCHRONIZE JOIN INDEX 文 91  
 SYSTEM 81

## V

Very Large Database Management 1

## W

WD インデックス 74

WHERE 句 53, 90

  BETWEEN 条件 43

  ORDER BY 句 39

  とパターン一致 41

  日付の比較 40

  例 39

WITHIN GROUP 句 50

## あ

アーキテクチャ

  Sybase IQ 6

アドホック・クエリ 58

アポストロフィ

  使用 39

アルファベット順 37

## い

インデックス 58

  High\_Group インデックス 74

  High\_Non\_Group インデックス 74

  Low Fast インデックス 75

  Low\_Fast インデックス 74

  WD インデックス 74

  ジョイン・インデックス 76

  ジョイン・インデックスの同期 91

  タイプ 74

  削除 91

  定義 19, 30

  比較 74

## お

オブジェクト 101

オプション

  予期しない動作 37

オプション機能

  Very Large Database Management 1

## 索引

- マルチプレックス・グリッド 1
- 高度なセキュリティ 1
- 非構造化データ分析 1

## か

- カタログ
  - システム 21
- カタログ・ストア 37, 99
- カラム
  - テーブルからの選択 38
  - プロパティの編集 67
  - 作成 66
  - 順序付け 38
  - 説明 35
  - 非常に多くの NULL 値 38
- カラム・インデックス
  - High\_Group インデックス 74
  - High\_Non\_Group インデックス 74
  - Low\_Fast インデックスの追加 75
  - Low\_Fast インデックス 74
  - WD インデックス 74
  - ジョイン・インデックス 76
  - タイプ 74
  - 作成 73, 74
  - 比較 74

## き

- キー
  - プライマリ 18
  - 外部 18
- キー・ジョイン 56
  - 使用 55
- キーワード
  - AND 41
  - DISTINCT 51

## く

- クエリ
  - SQL Anywhere による処理 37
  - アドホック・クエリ 58
  - インデックス 58
  - ジョイン 20
  - 作成 35

- 射影 20
- 制限 20
- クライアント接続
  - UNIX 24
  - Windows 23
- グループ化されたデータ 43

## こ

- コマンド
  - 長い 39
- コマンド・ライン接続 32
- コンテナ
  - 選択 28
  - 展開 28
- コンポーネント統合サービス (CIS) 99

## さ

- サーバ
  - 接続 23
  - 停止 34
  - 定義 13
- サブクエリ
  - 外部ジョイン 62
  - 使用 62
- サンプル・アプリケーション 8

## し

- システム DB 領域 37
- システム・ストアド・プロシージャ 35
- システム・テーブル 21
- ジョイン 53
  - Adaptive Server Enterprise 60
  - BLANK PADDING 60
  - CIS 60, 61
  - FROM 句 53
  - ORDER BY 句 53
  - SQL Anywhere 60
  - WHERE 句 53
  - アドホック・ジョイン 58
  - サブクエリ 62
  - ジョイン・インデックス 58
  - データ型 59
  - ナチュラル 57

- プライマリ・キー 55
- リモート・データベース 61
- 異種データベース 61
- 演算子 55
- 外積 53
- 外部キー 55
- 外部ジョイン 62
- 制限 53
- 定義 20
- 同期 91
- 複数のテーブル 56
- ジョイン・インデックス 100
  - Department と Employee 77
  - SalesOrders と SalesOrderItems 77
  - 更新 78
  - 作成 76
- ジョイン演算子
  - キー・ジョイン 55, 56
  - ナチュラル・ジョイン 57

## す

- ストア 102
  - カタログ 81
  - テンポラリ 15, 81
  - メイン 81
  - 共有テンポラリ 81
- ストアド・プロシージャ 35
  - トリガ 68
  - パーミッション 68
  - 管理 68
  - 実行 69
  - 定義 19

## せ

- セーブポイント 87
  - データの一貫性 88
  - 設定 88

## た

- タプル 17
- ダミーの Sybase IQ テーブル 37

## ち

- チュートリアル
  - Sybase Central 25

## て

- データ
  - テーブル 15
  - ロード 78
- データ、選択 35
- データのインデックス 73
  - インデックスの比較 74
  - ジョイン・インデックス 76
  - タイプ 74
  - 作成 73, 74
- データのロード 73
  - Interactive SQL 78
  - Sybase Central 78
  - プログラミング・インタフェース 78
  - 可変長フォーマット 78
- データの一貫性
  - セーブポイント 87, 88
  - セーブポイントの設定 88
  - チェックポイント 87
- データの選択 35
- データベース
  - DB 領域 91
  - アクセス制限 98
  - インデックス 91
  - ジョイン・インデックスの同期 91
  - セーブポイント 87
  - データの一貫性 87
  - テーブル 91
  - デモ 10
  - ファイル 14
  - リモート 61
  - リモートのジョイン 61
  - リレーショナル 16
  - ローの更新 90
  - ローの削除 90
  - ローの追加 85
  - 異種 61
  - 管理 25, 65
  - 更新 85
  - 接続先 32
  - 複数テーブルの更新 89
  - 変更 86

## 索引

- データベース・サーバ
    - 起動 9
  - データベース・テーブル
    - Java 19
    - インデックス 19
    - カラム 17
    - ストアド・プロシージャ 19
    - その他のデータベース・オブジェクト 19
    - タプル 17
    - ビュー 19
    - プライマリ・キー 18
    - ユーザとグループ 19
    - ロー 17
    - 外部キー 18, 19
    - 特性 17
  - データベース、デモ
    - iqdemo 10
  - データベースからのユーザの切断 71
  - データベースの管理 65
    - テーブル 65
    - プロシージャ 68
    - ユーザとグループ 69
  - データベースへのユーザの追加 70
  - データ型
    - ジョイン 59
    - 説明 17
  - データ辞書 21
  - テーブル
    - iq\_dummy 37
    - Sybase Central での表示 29
    - カラムの作成 66
    - カラムの編集 67
    - プロキシ・テーブル 67
    - リスト 35
    - 管理 65
    - 作成 65
    - 削除 67, 91
    - 関連名 53
    - 特性 17
    - 複数テーブルの更新 89
  - テーブル・データ 15
  - テーブル・パーティション 102
  - テーブル、iqdemo
    - dba テーブル 10
    - GROUPO テーブル 10
    - iq\_dummy テーブル 10
  - テーブルのクエリ 37
  - テーブルの管理
    - office テーブルの作成 65
    - カラムの作成 66
    - カラムの編集 67
    - テーブルの削除 67
  - テーブル領域 102
  - デモ・データベース 9
    - dba テーブル 10
    - GROUPO テーブル 10
    - iq\_dummy テーブル 10
    - iqdemo 10
    - office テーブルの削除 67
    - ユーザの追加 70
    - リストア 71
    - 起動 8
    - 接続先 32
  - テンポラリ・ストア 15
- ## と
- トランザクション・ログ 15
- ## な
- ナチュラル・ジョイン 57
- ## ね
- ネットワーク・サーバ
    - 起動 9
  - ネットワーク・サーバの実行プログラム 9
- ## は
- パーティション・キー 101
  - パーミッション
    - ストアド・プロシージャ 68
  - パスワード
    - とパーミッション 96
  - パターン一致 41
  - パフォーマンス 15
    - 向上 45

## ひ

## ビュー

- CREATE VIEW 文 93, 94
- アクセス制限 98
- セキュリティ 96
- ビュー 96
- マテリアライズド・ビュー 93
- レギュラー・ビュー 93
- 作成 94
- 削除 95
- 使用 93
- 制限 95
- 定義 19, 93
- 表示 94
- 変更 94

## ふ

- プライマリ・キー 55
  - 定義 18
- プラグイン
  - 接続 26
  - 登録 26
- フリー・リスト 99
- プロキシ・データベース 101
- プロキシ・テーブル 67
- プログラミング・インタフェース 78
- プログラム・グループ 8
- プロシージャ
  - Sybase Central での表示 30
  - パーミッション 68
  - 管理 68
  - 実行 69
  - 表示 68

## へ

- ベース・テーブル 19

## ま

- マテリアライズド・ビュー 93
- マニュアル 4

- マルチプレックス 101
- マルチプレックス・グリッド 1

## め

- メタデータ 101
- メッセージ・ログ
  - IQ 15

## ゆ

- ユーザ
  - Sybase Central での表示 30
  - データベースへの追加 70
- ユーザ・マニュアル 4
- ユーザとグループ
  - グループの追加 70
  - デモ・データベースのリストア 71
- ユーザの接続 71
- ユーザの追加 70
- ログイン・ポリシーの追加 70
- 管理 69
- ユーザ接続
  - 管理 71

## ら

- ライセンス 1

## り

- リモート・データベース 61
  - ジョイン 61
- リレーショナル・データベースの概念 16

## ろ

- ロー
  - 更新 90
  - 説明 35
  - 選択 39
- ロー・デバイス・アクセス 8
- ログ
  - トランザクション 15
  - メッセージ 15

## 索引

- ログイン・ポリシー
  - グループの追加 70
  - デモ・データベースのリストア 71
- ユーザの接続 71
- ユーザの追加 70
- 追加 70