



パフォーマンス & チューニング・ガイド

## **Sybase IQ**

15.1

ドキュメント ID : DC00283-01-1510-01

改訂 : 2009 年 7 月

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

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

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

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

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

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

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

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

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

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

# 目次

はじめに.....	vii
<b>第 1 章</b>	<b>データベース・テーブルからのデータの選択 ..... 1</b>
前提条件 .....	2
テーブル情報の表示 .....	2
クエリ結果の順序付け .....	4
カラムとローの選択 .....	6
検索条件の使用 .....	7
クエリでの日付の比較 .....	8
WHERE 句での複合検索条件 .....	8
検索条件でのパターン・マッチング .....	9
発音によるローのマッチング .....	10
検索条件のショートカットの使用 .....	10
集約データの取得 .....	11
集合関数によるグループ化されたデータの取得 .....	12
グループの制限 .....	13
小計計算の活用 .....	14
分析データの取得 .....	17
重複したローの削除 .....	19
<b>第 2 章</b>	<b>テーブルのジョイン ..... 21</b>
外積を使用したテーブルのジョイン .....	22
ジョインの制限 .....	22
テーブル間の関係 .....	24
プライマリ・キー識別子キー .....	24
テーブル関係の外部キー .....	24
ジョイン演算子 .....	25
キー・ジョインを使用したテーブルのジョイン .....	25
ナチュラル・ジョインを使用したテーブルのジョイン .....	26
アドホック・ジョインとジョイン・インデックスの使用 .....	27
ジョインとデータ型 .....	29
ストアまたはデータベース間ジョインのサポート .....	29

	リモート・データベースと異種データベースのクエリ .....	31
	サブクエリによるジョインの置き換え .....	32
<b>第 3 章</b>	<b>クエリと削除の最適化 .....</b>	<b>35</b>
	クエリ構築のヒント .....	36
	UNION ALL での GROUP BY がクエリ・パフォーマンスに 与える影響 .....	36
	ORDER BY クエリ・パフォーマンスの強化 .....	39
	クエリ内の並列処理の機能強化 .....	40
	サブクエリのパフォーマンスの改善 .....	40
	キャッシュ方法の使用 .....	40
	SQL Anywhere による処理を引き起こす条件 .....	41
	クエリ・プラン .....	41
	クエリ評価オプション .....	42
	クエリ・ツリー .....	44
	クエリ・プランの使用 .....	44
	クエリ処理の制御 .....	46
	クエリの時間制限の設定 .....	46
	クエリの優先度の設定 .....	46
	クエリ最適化オプションの設定 .....	47
	ユーザ指定の条件ヒントの設定 .....	48
	負荷のモニタリング .....	49
	削除オペレーションの最適化 .....	50
	HG 削除オペレーション .....	50
	WD 削除オペレーション .....	51
<b>第 4 章</b>	<b>システム・リソースの管理 .....</b>	<b>53</b>
	パフォーマンス用語の概要 .....	54
	パフォーマンス向上のための設計 .....	54
	メモリ使用の概要 .....	54
	ページングによる使用可能メモリの増加 .....	55
	スワッピングをモニタするためのユーティリティ .....	56
	サーバ・メモリ .....	56
	バッファ・キャッシュの管理 .....	57
	バッファ・キャッシュ・サイズの決定 .....	58
	バッファ・キャッシュ・サイズの設定 .....	63
	ページ・サイズの指定 .....	65
	メモリの節約 .....	66
	ユーザが多数存在する場合の最適化 .....	68
	プラットフォーム固有のメモリ・オプション .....	70
	メモリを増やすその他の方法 .....	73

プロセス・スレッド・モデル .....	74
スレッド不足エラー .....	75
スレッド使用を管理するための Sybase IQ オプション .....	75
I/O の分散 .....	76
ロー I/O (UNIX オペレーティング・システム) .....	76
ディスク・ストライピングの使用 .....	77
内部ストライピング .....	79
複数のファイルの使用 .....	80
戦略的なファイルの格納 .....	80
挿入、削除、同期のための作業領域 .....	84
予約領域のオプションの設定 .....	85
リソース使用を調整するオプション .....	85
同時クエリの制限 .....	86
使用可能な CPU 数の設定 .....	86
クエリによるテンポラリ DB 領域の使用の制限 .....	86
返されるローによるクエリの制限 .....	87
カーソルのスクロールの禁止 .....	87
カーソル数の制限 .....	88
文の数の制限 .....	88
キャッシュ・ページのプリフェッチ .....	88
一般的な使用のための最適化 .....	89
プリフェッチされるローの数の制御 .....	89
リソースを効率的に利用するための他の方法 .....	89
マルチプレックス・データベースのディスク領域の管理 .....	90
クエリ・サーバ間のロード・バランス .....	90
データベース・アクセスの制限 .....	91
ディスクのキャッシュ .....	91
インデックスのヒント .....	91
正しいインデックス・タイプの選択 .....	92
ジョイン・インデックスの使用 .....	93
削除のための十分なディスク領域の確保 .....	93
データベース・サイズと構造の管理 .....	93
データベース・サイズの管理 .....	93
インデックスの断片化の制御 .....	94
カタログ・ファイル増大の最小化 .....	95
パフォーマンス向上のための非正規化 .....	95
非正規化のリスク .....	95
非正規化の短所 .....	96
非正規化のパフォーマンスの利点 .....	96
非正規化の決定 .....	97
ロードを高速化するための UNION ALL ビューの使用 .....	97
UNION ALL ビューを参照するクエリの最適化 .....	98
UNION ALL ビューのパフォーマンスの管理 .....	99
大きい単一 (ファクト) テーブルのロードの改善 .....	100

	ネットワーク・パフォーマンス .....	100
	大量のデータ転送の向上 .....	100
	ヘビー・ネットワーク・ユーザの分離 .....	101
	少量のデータを小さなパケットに入れる .....	102
	大量のデータを大きなパケットに入れる .....	102
	サーバ・レベルのプロセス .....	102
<b>第 5 章</b>	<b>パフォーマンスのモニタリングとチューニング .....</b>	<b>103</b>
	Sybase IQ 環境の表示 .....	104
	パフォーマンス統計のモニタリング .....	104
	ストアド・プロシージャを使用して情報を取得する .....	113
	データベース・プロシージャのプロファイリング .....	113
	バッファ・キャッシュのモニタリング .....	123
	バッファ・キャッシュ・モニタの起動 .....	123
	モニタ実行中の結果の確認 .....	130
	バッファ・キャッシュ・モニタの停止 .....	131
	モニタリング結果の検査と保存 .....	131
	モニタリング結果の例 .....	132
	バッファ・キャッシュの構造 .....	136
	バッファ・マネージャのスラッシングの回避 .....	137
	Windows システムでのページングのモニタリング .....	139
	UNIX システムでのページングのモニタリング .....	139
	バッファ・キャッシュ・モニタリング・チェックリスト .....	141
	CPU 使用率をモニタリングするシステム・ユーティリティ .....	146
<b>第 6 章</b>	<b>32 ビット Windows システムでのサーバのチューニング .....</b>	<b>147</b>
	パフォーマンスについての一般的なガイドライン .....	148
	スループットの最大化 .....	148
	メモリの割り付け超過の防止 .....	149
	物理メモリのモニタリング .....	149
	ファイル・システム .....	149
	パフォーマンスのモニタリング .....	150
	仮想アドレス空間とワーキング・セットのモニタリング .....	150
	ページ・フォールトのモニタリング .....	151
	NTFS キャッシュの使用 .....	152
	挿入とクエリのチューニング .....	152
	適切にチューニングされた挿入オペレーションの特性 .....	153
	クエリのチューニング .....	154
	バックアップ操作のチューニング .....	154
<b>索引</b> .....		<b>157</b>

# はじめに

## このマニュアルの内容

このマニュアルでは、パフォーマンスとチューニングの推奨事項について説明します。

## 対象読者

このマニュアルは、パフォーマンス上の問題を理解する必要があるシステム管理者とデータベース管理者を対象としています。リレーショナル・データベース・システムの基礎知識と、Sybase IQのユーザ・レベルの基礎的な経験があることを前提にしています。このマニュアルは、他のマニュアルとともに使用してください。

## このマニュアルの使用方法

次のリストは、行う作業や必要性に応じてどの章を参照すべきかを示します。

- 「[第 1 章 データベース・テーブルからのデータの選択](#)」では、基本的なクエリの構築について説明しています。
- 「[第 2 章 テーブルのジョイン](#)」では、複数のデータベース・テーブルから情報を取得する方法について説明しています。
- 「[第 3 章 クエリと削除の最適化](#)」では、クエリ・パフォーマンスを向上させる方法について説明しています。
- 「[第 4 章 システム・リソースの管理](#)」では、メモリ、ディスク I/O、および CPU の管理方法について説明しています。
- 「[第 5 章 パフォーマンスのモニタリングとチューニング](#)」では、システムが使用可能なリソースを最大限に利用しているかどうかを確認するために使用するツールについて説明しています。
- 「[第 6 章 32 ビット Windows システムでのサーバのチューニング](#)」では、Windows のパフォーマンスとチューニングについて説明しています。

## 関連マニュアル

Sybase IQ 15.1 のマニュアル・セットには、以下が含まれています。

- 『リリース・ノート』では、製品およびマニュアルに加えられた最新の変更内容について説明しています。

- 
- 『インストールおよび設定ガイド』では、プラットフォーム固有のインストール手順、新バージョンへのマイグレート、特定のプラットフォームでの Sybase IQ の設定について説明しています。
  - 『Sybase IQ による高度なセキュリティ』では、Sybase IQ データ・レポジトリ内でのユーザによるカラムの暗号化の使用について説明しています。このオプションの製品をインストールするには、別のライセンスが必要です。
  - 『エラー・メッセージ』は、Sybase エラー・コード、SQLCode、および SQLState によって参照される Sybase IQ エラー・メッセージ、および SQL プリプロセッサのエラーと警告を示します。
  - 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 巻 C 統計ライブラリ』には、IMSL C 統計ライブラリの時系列 C 関数の簡潔な説明が記載されています。
  - 『Sybase IQ の概要』には、Sybase IQ や Sybase Central™ データベース管理ツールの操作に慣れていないユーザのために、操作の練習問題が含まれています。
  - 『Sybase IQ によるラージ・オブジェクト管理』では、Sybase IQ データ・レポジトリ内での BLOB (バイナリ・ラージ・オブジェクト) および CLOB (キャラクタ・ラージ・オブジェクト) の格納と取得について説明しています。このオプションの製品をインストールするには、別のライセンスが必要です。
  - 『Sybase IQ 15.0 の新機能』には、バージョン 15.0 の新機能と動作変更が記載されています。
  - 『新機能の概要 Sybase IQ 15.1』には、最新バージョンの新機能と動作変更がまとめられています。
  - 『クイック・スタート』には、Sybase IQ のソフトウェア・インストールを確定するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う手順が記載されています。デモ・データベースをマルチプレックスに変換するための情報も記載されています。
  - 『リファレンス・マニュアル』— Sybase IQ の 2 冊のリファレンス・ガイドで構成されています。
    - 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』では、Sybase IQ がサポートしている SQL、ストアド・プロシージャ、データ型、およびシステム・テーブルについて説明しています。



- 『リファレンス：文とオプション』では、Sybase IQ がサポートしている SQL 文およびオプションについて説明しています。
- 『システム管理ガイド』— 2 巻構成です。
  - 『システム管理ガイド：第 1 巻』では、起動、接続、データベース作成、自動入力とインデックス作成、バージョン管理、照合、システムのバックアップとリカバリ、トラブルシューティング、およびデータベースの修復について説明しています。
  - 『システム管理ガイド：第 2 巻』では、プロシージャとバッチの作成および実行、OLAP でのプログラミング、リモート・データへのアクセス、Open Server としての IQ の設定、スケジューリングとイベント処理、XML でのプログラミング、およびデバッグについて説明しています。
- 『ユーザ定義関数ガイド』には、ユーザ定義関数、そのパラメータ、および考えられる使用事例が記載されています。
- 『Sybase IQ マルチプレックスの使用』では、複数のノードにまたがって発生する大きなクエリの負荷を管理するために設計されているマルチプレックス機能の使用方法について説明しています。
- 『ユーティリティ・ガイド』では、Sybase IQ ユーティリティ・プログラムのリファレンス項目 (使用可能な構文、パラメータ、オプションなど) について説明しています。

---

### Sybase IQ および SQL Anywhere

Sybase IQ は、SQL Anywhere サーバの拡張機能であり、SQL Anywhere® パッケージのコンポーネントです。このため、Sybase IQ は、SQL Anywhere サーバの多くの機能をサポートしています。IQ のマニュアル・セットでは、該当する場合、SQL Anywhere のマニュアルを参照しています。

---

SQL Anywhere には、次のマニュアルがあります。

- 『SQL Anywhere Server — データベース管理ガイド』では、SQL Anywhere データベースの実行、管理、および設定方法について説明しています。データベース接続、データベース・サーバ、データベース・ファイル、バックアップ手順、セキュリティ、高可用性、Replication Server® での複製、管理ユーティリティおよびオプションについても説明します。

- 
- 『SQL Anywhere Server — プログラミング・ガイド』では、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET プログラミング言語を使用してデータベース・アプリケーションを構築し配備する方法について説明しています。このマニュアルでは、ADO.NET や ODBC などの各種プログラミング・インタフェースについても説明します。
  - 『SQL Anywhere Server — SQL リファレンス・マニュアル』は、システム・プロシージャおよびカタログ (システム・テーブルおよびビュー) に関する参照情報を示します。また、SQL 言語の SQL Anywhere の実装 (検索条件、構文、データ型、関数) についても説明します。
  - 『SQL Anywhere Server — SQL 使用法』では、データベースの設計/作成方法、データのインポート/エクスポート/変更方法、データの検索方法、およびストアド・プロシージャとトリガの作成方法について説明します。

また、[Product Manuals \(http://www.sybase.com/support/manuals/\)](http://www.sybase.com/support/manuals/) の SQL Anywhere 11.01 コレクション、および [DocCommentXchange \(http://dcx.sybase.com/dcx\\_home.php\)](http://dcx.sybase.com/dcx_home.php) の SQL Anywhere マニュアルを参照することもできます。

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

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

## その他の情報

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

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

- SyBooks CD には製品マニュアルが収録されています。この CD はソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザでは、使いやすい HTML 形式のマニュアルにアクセスできます。

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

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

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

Sybase Product Manuals Web サイトは、[Product Manuals \(http://www.sybase.com/support/manuals/\)](http://www.sybase.com/support/manuals/) にあります。

## Web 上の Sybase 製品の動作確認情報

Sybase Web サイトの技術的な資料は頻繁に更新されます。

### ❖ 製品動作確認の最新情報にアクセスする

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

### ❖ コンポーネント動作確認の最新情報にアクセスする

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

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

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

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

❖ EBF とソフトウェア・メンテナンスの最新情報にアクセスする

- 1 Web ブラウザで [Sybase Support Page](http://www.sybase.com/support) (<http://www.sybase.com/support>) を指定します。
- 2 [EBFs/Maintenance] を選択します。ユーザ名とパスワードの入力を求められたら、MySybase のユーザ名とパスワードを入力します。
- 3 製品を選択します。
- 4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

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

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

SQL 構文の表記規則

このマニュアルでは、構文の説明に次の表記規則を使用します。

- **キーワード** SQL キーワードは大文字で示します。ただし、SQL キーワードは大文字と小文字の区別がないので、入力するときはどちらで入力してもかまいません。たとえば、SELECT は Select でも select でも同じです。
- **プレースホルダ** 適切な識別子または式で置き換えられる項目は、斜体で表記します。

- **継続** 省略記号 (...) で始まる行は、前の行から文が続いていることを表します。
- **繰り返し項目** 繰り返し項目のリストは、リストの要素の後ろに省略記号 (...) を付けて表します。1 つまたは複数の要素を指定できます。複数の要素を指定する場合は、各要素間はカンマで区切る必要があります。
- **オプション部分** 文のオプション指定部分は、角カッコで囲みます。例を示します。

RELEASE SAVEPOINT [ **セーブポイント名** ]

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

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

[ ASC | DESC ]

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

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

QUOTES { ON | OFF }

中カッコは、ON か OFF のいずれかを含めなければいけないことを示します。中カッコは入力しないでください。

## 書体の表記規則

表 1 に、このマニュアルで使用している書体の表記規則を示します。

**表 1：書体の表記規則**

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

---

## デモ・データベース

Sybase IQ にはデモ・データベース (*iqdemo.db*) を作成するためのスクリプトが用意されています。このマニュアルで紹介している多くのクエリおよびコード例は、このデモ・データベースをデータ・ソースに使用しています。

デモ・データベースは、小規模会社の内部情報 (従業員、部署、財務データ) に加えて、製品と販売情報 (注文、顧客、担当者) で構成されています。

デモ・データベースの詳細については、使用しているプラットフォームの『Sybase IQ インストール・ガイド』を参照するか、システム管理者に相談してください。

## アクセシビリティ機能

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

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

---

### アクセシビリティ・ツールの設定

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

---

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

## 不明な点があるときは

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

# データベース・テーブルからのデータの選択

この章について

この章では、基本的なクエリの構築と、製品設計を活用するための手法について説明します。ここではテーブルの内容の表示、クエリ結果の順序付け、カラムとローの選択、検索条件を使ったクエリの絞り込みを行うチュートリアル作業を実行します。

高度なクエリ・パフォーマンスの推奨事項については、「[第 3 章 クエリと削除の最適化](#)」を参照してください。

内容

トピック名	ページ
<a href="#">前提条件</a>	<a href="#">2</a>
<a href="#">テーブル情報の表示</a>	<a href="#">2</a>
<a href="#">クエリ結果の順序付け</a>	<a href="#">4</a>
<a href="#">カラムとローの選択</a>	<a href="#">6</a>
<a href="#">検索条件の使用</a>	<a href="#">7</a>
<a href="#">集約データの取得</a>	<a href="#">11</a>
<a href="#">分析データの取得</a>	<a href="#">17</a>
<a href="#">重複したローの削除</a>	<a href="#">19</a>

## 前提条件

Interactive SQL の代わりにグラフィカルなフロントエンド・ツールを使用してデータベースへのクエリを実行すると、ツールが生成する SQL 構文を表示できる場合があります。たとえば、InfoMaker では [テーブル] ペインタ・バーの [SQL 構文] ボタンを選択して SQL 文を表示できます。

このチュートリアルでは、データベースから情報を取得するときに使用する SELECT 文について説明します。SELECT 文のことを一般にクエリと呼びます。これは、SELECT 文がデータベース内の情報についてデータベース・サーバに問い合わせるためです。

---

**注意** SELECT 文は用途の広いコマンドです。大きなデータベースから非常に具体的な情報を取得するアプリケーションでは、SELECT 文がきわめて複雑になる場合があります。このチュートリアルでは、単純な SELECT 文だけを使用します。以降のチュートリアルで、より高度なクエリについて説明します。SELECT 文の完全な構文については、『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」の「[SELECT 文](#)」を参照してください。

---

チュートリアルのレッスンを読んで実行している間は、コンピュータで Sybase IQ ソフトウェアを実行しておくことが理想的です。

このチュートリアルでは、すでに Interactive SQL を起動し、サンプル・データベースに接続していることを前提にしています。まだこれらを行っていない場合は、『ユーティリティ・ガイド』の「[第 2 章 Interactive SQL \(dbisql\) の使用](#)」を参照してください。

## テーブル情報の表示

ここでは、Employees テーブルのデータを表示します。

このチュートリアルで使用するサンプル・データベースは、『Sybase IQ の概要』で取り上げた架空の会社と同じものです。データベースには、従業員、部署、注文などについての情報が格納されています。すべての情報はテーブルに編成されています。



## テーブルのリスト

『Sybase IQ の概要』では、Sybase Central で [テーブル] フォルダを開いてテーブルのリストを表示する方法について説明しました。システム・ストアド・プロシージャの `sp_iqtable` を使用して、Interactive SQL からユーザ・テーブルをリストすることもできます。システム・ストアド・プロシージャは、Sybase IQ にストアド・プロシージャとして実装されているシステム関数です。

[SQL 文] ウィンドウで `sp_iqtable` と入力し、同じ名前のシステム・ストアド・プロシージャを実行します。

システム・ストアド・プロシージャの詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」を参照してください。

## SELECT 文の使用

このレッスンでは、データベース内のテーブルの 1 つを表示します。使用するコマンドは、`Employees` という名前のテーブル全体を表示します。

次のコマンドを実行します。

```
SELECT * FROM Employees
```

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 文は `Employees` テーブルのすべてのローとカラムを取得し、Interactive SQL の [結果] ウィンドウに次の該当するものが表示されます。

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
102	501	Whitney	Fran	100	...
105	501	Cobb	Matthew	100	...
129	902	Chin	Philip	200	...
148	1293	Jordan	Julie	300	...
160	501	Breault	Robert	100	...
184	1576	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

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** このマニュアルの例は、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。

---

データベースの作成方法については、『システム管理ガイド：第 1 巻』の「[第 5 章 データベース・オブジェクトの使用](#)」を参照してください。

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。

Interactive SQL 環境の操作方法と Interactive SQL の使用法は、オペレーティング・システムによって異なります。

データをスクロールして Interactive SQL 環境を操作する方法については、『ユーティリティ・ガイド』の「[第 2 章 Interactive SQL \(dbisql\) の使用](#)」を参照してください。

## クエリ結果の順序付け

ここでは、SELECT 文に ORDER BY 句を追加して、結果をアルファベット順または数値順に表示します。

特に指定しないかぎり、Sybase IQ ではテーブルのローが順不同で表示されます。テーブルのローを意味のある順序で表示した方が便利ながよくあります。たとえば、従業員をアルファベット順で表示したいような場合です。

従業員をアルファベット順にリストする

次の例は、SELECT 文に ORDER BY 句を追加して、結果をアルファベット順に取得する方法を示します。

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

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
1751	1576	Ahmed	Alex	400	...
1013	703	Barker	Joseph	500	...
591	1576	Barletta	Irene	400	...
191	703	Bertrand	Jeannette	500	...
1336	1293	Bigelow	Janet	300	...
1062	1576	Blaikie	Barbara	400	...
750	703	Braun	Jane	500	...
160	501	Breault	Robert	100	...
1191	1576	Bucceri	Matthew	400	...

注意

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。

**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 で処理することができます。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

ここでは、各従業員の誕生日、姓、部署 ID を選択します。次のコマンドを入力します。

```
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
```

---

**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

ここでは、WHERE 句の複合検索条件、パターン・マッチング、検索条件ショートカットを使用して、日付を比較する手順について説明します。

**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。SELECT 文に WHERE 句を追加すると、テーブルから一部のローだけを選択できます。

たとえば、John という名前の従業員だけを表示するとします。

❖ John という名前の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 *
FROM Employees
WHERE GivenName = 'John'
```

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
318	1576	Crow	John	400	...
862	501	Sheffield	John	100	...
1483	1293	Letiecq	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ると実行できます。

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

注意

- 句を指定する順序は重要です。FROM 句を最初に指定し、その後に WHERE 句、ORDER BY 句の順に指定します。これ以外の順序で句を入力すると、構文エラーが返されます。

- 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

検索対象の正確な値がわからない場合や、一連の値を表示したい場合があります。WHERE 句で比較を使用すると、検索条件を満たす一連のローを選択できます。

1964 年 3 月 3 日より前に生まれた従業員のリスト

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

これまでに、比較演算子の等号 (=) と未満 (<) を見てきました。Sybase IQ では、より大きい (>)、以上 (>=)、以下 (<=)、等しくない (<>) などのその他の比較演算子もサポートされています。

これらの条件を AND や OR を使って組み合わせると、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

もう 1 つの便利な検索方法が、パターンによる検索です。SQL では、LIKE という語を使用してパターンを検索します。LIKE の使い方について、例を挙げて説明します。

## 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
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 で使用できるもう 1 つの特殊文字に \_ (アンダースコア) 文字があります。これは 1 文字と一致します。

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、スペルだけでなく読みによってもローをマッチングできます。たとえば、電話メッセージが残されていて、その宛先が "Ms. Brown" のように発音されていたとします。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。

発音による姓の検索

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

省略形 BETWEEN の使用

SQL には、検索条件を入力するための省略形が 2 つあります。1 つは BETWEEN であり、値を範囲で検索するときに使用します。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'
```

#### 省略形 IN の使用

もう 1 つの省略形 **IN** は、複数の値のいずれかを検索するときに使用します。次にコマンド例を示します。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

ここでは、集約情報を返すクエリを構築する方法について説明します。集約情報の例を次に示します。

- カラム内のすべての値の合計
- カラム内のエントリの数
- カラム内のエントリの平均値

#### 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
```

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

テーブル全体についての情報を取得することに加えて、集合関数をローのグループに対して使うこともできます。

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
467	56
667	54
690	52
856	55
902	47
949	53

SalesRepresentative	Count()
1142	57
1596	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

WHERE 句を使用して、クエリでローを制限する方法についてはすでに説明しました。GROUP BY 句の制限には、HAVING キーワードを使用します。

### 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
1142	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・オブティマイザは、それによってパフォーマンスが向上する場合、述部を **HAVING** 句から **WHERE** 句に移動します。たとえば、上記の例で **WHERE** 句の代わりに述部を次のように指定した場合、クエリ・オブティマイザは述部を **WHERE** 句に移動します。

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

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必要がある場合があります。**ROLLUP** 演算子と **CUBE** 演算子を使用すると、グループ化カラムへの参照のリストから複数レベルの小計と総計を作成できます。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。たとえば、販売データを分析している場合は、同じクエリを使用して全体の平均と年別の平均販売数を計算できます。

ROLLUP の使用

年別、モデル別、色別の合計自動車販売数を選択するには、次のコマンドを使用します。

```
SELECT year, model, color, sum(sales)
FROM sales_tab
GROUP BY ROLLUP (year, model, color);
```

year	model	color	sales
1990	Chevrolet	red	5
1990	Chevrolet	white	87
1990	Chevrolet	blue	62
1990	Chevrolet	NULL	154
1990	Ford	blue	64
1990	Ford	red	62
1990	Ford	white	63
1990	Ford	NULL	189
1990	NULL	NULL	343
1991	Chevrolet	blue	54

year	model	color	sales
1991	Chevrolet	red	95
1991	Chevrolet	white	49
1991	Chevrolet	NULL	198
1991	Ford	blue	52
1991	Ford	red	55
1991	Ford	white	9
1991	Ford	NULL	116
1991	NULL	NULL	314
NULL	NULL	NULL	657

このクエリを処理するときに、Sybase IQ は最初に、指定された 3 つすべてのグループ化式 (year、model、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。5 番目のローの NULL は、color カラムの ROLLUP 値、つまり、そのモデルのすべての色の合計販売数を示します。343 は、1990 年のすべてのモデルと色の合計販売数を表し、314 は 1991 年の合計販売数を表します。最後のローは、すべての年のすべてのモデルとすべての色の合計販売数を表します。

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 (たとえば、state をリストしてから city をリストします)。

ROLLUP 演算子は、集合関数の AVG、COUNT、MAX、MIN、STDDEV、SUM、VARIANCE とともに使用できます。ただし、ROLLUP は COUNT DISTINCT と SUM DISTINCT をサポートしていません。

## CUBE の使用

次のクエリでは、人々の州 (地理的位置)、性別、教育水準、所得を含む Employees テーブルのデータを使用します。GROUP BY 句の CUBE 拡張を使用すると、census テーブル内の国勢調査データを 1 回参照するだけで、州、性別、教育水準の国勢調査全体の平均所得を計算し、state、gender、education の各カラムの可能なすべての組み合わせの平均所得を計算できます。CUBE 演算子を使用すると、たとえば、すべての州における全女性の平均収入を計算したり、調査対象者全員の平均収入を、各自の教育別および州別に計算したりすることができます。

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。

次のクエリは、**GROUPING** 関数を **GROUP BY CUBE** と組み合わせた使用例です。

```
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;
```

このクエリの結果は次のとおりです。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。

c_state	c_gender	c_dept	COUNT()	AVERAGE
ALL	ALL	200	3	52200.00
ALL	ALL	ALL	3	52200.00
ALL	F	200	2	58650.00
ALL	F	ALL	2	58650.00
ALL	M	200	1	39300.00
ALL	M	ALL	1	39300.00
CA	ALL	200	3	52200.00
CA	ALL	ALL	3	52200.00
CA	F	200	2	58650.00
CA	F	ALL	2	58650.00
CA	M	200	1	39300.00
CA	M	ALL	1	39300.00

ROLLUP と CUBE は、データ・ウェアハウス管理者が次のような操作を行うときに特に役立ちます。

- 地理や時間などの階層的な次元での小計 (たとえば、年/月/日や国/州/市)
- 要約テーブルへのデータの格納

ROLLUP と CUBE 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 演算子と CUBE 演算子の詳細については、『リファレンス：文とオプション』の「SQL 文」の「SELECT 文」を参照してください。

## 分析データの取得

ここでは、分析情報を返すクエリを構築する方法について説明します。統計関数には、ランク付けと逆分散統計の 2 種類があります。ランク付け統計関数は、グループ内の項目をランク付けしたり、分散統計を計算したり、結果セットを複数のグループに分割したりします。逆分散統計関数は、K-理論パーセンタイル値を返します。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。

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。

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 empl;
```

Name	Salary	Ranking
Sandy	55000	1
Peter	48000	1
Lisa	38000	1
Scott	29000	1
Tim	29000	2
Tom	28000	2

Name	Salary	Ranking
Mike	28000	2
Adam	25000	3
Antonia	22000	3
Jim	22000	3
Anna	18000	4
Jeff	18000	4
Amy	18000	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。

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

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。

統計関数の詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 4 章 SQL 関数](#)」の「[分析関数](#)」を参照してください。個別の統計関数については、同じ章の各関数の項を参照してください。



## 重複したローの削除

SELECT 文の結果テーブルに、重複したローが含まれることがあります。DISTINCT キーワードを使用すると、重複したローを削除できます。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。

---

この章では、単一テーブルの SELECT 文の概要について説明しました。単一テーブルの SELECT 文の詳細については、以下を参照してください。

- 『システム管理ガイド：第 1 巻』の「[第 5 章 データベース・オブジェクトの使用](#)」
- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 2 章 SQL 言語の要素](#)」
- 『リファレンス：文とオプション』の「[第 1 章 SQL 文](#)」の「[SELECT 文](#)」

次の章では、SELECT 文の高度な使い方について説明します。



# テーブルのジョイン

## この章について

この章では、複数のテーブルにある情報を参照する方法と、さまざまな種類のジョインについて説明します。ここでは、テーブルをジョインするチュートリアル作業を実行します。

## 内容

トピック名	ページ
<a href="#">外積を使用したテーブルのジョイン</a>	22
<a href="#">ジョインの制限</a>	22
<a href="#">テーブル間の関係</a>	24
<a href="#">ジョイン演算子</a>	25
<a href="#">アドホック・ジョインとジョイン・インデックスの使用</a>	27
<a href="#">ジョインとデータ型</a>	29
<a href="#">ストアまたはデータベース間ジョインのサポート</a>	29
<a href="#">リモート・データベースと異種データベースのクエリ</a>	31
<a href="#">サブクエリによるジョインの置き換え</a>	32

## 外積を使用したテーブルのジョイン

サンプル・データベースに、会社の財務データをリストする **FinancialData** というテーブルがあります。各データ・レコードには、そのレコードの部署と、それが支出レコードか収入レコードかを識別する **code** カラムがあります。

2 つのテーブルから同時に情報を取り出すには、**SELECT** クエリの **FROM** 句で、両方のテーブルをカンマで区切ってリストします。

例 次の **SELECT** コマンドは、**FinancialCodes** テーブルと **FinancialData** テーブルのすべてのデータをリストします。

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

**Interactive SQL** [データ] ウィンドウに表示されるこのクエリの結果は、**FinancialCodes** テーブルのすべてのローと **FinancialData** テーブルのすべてのローに一致します。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、**FinancialData** テーブルのすべてのカラム、**FinancialCodes** 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。これ以降の項で、より選択性の高いジョインを構築する方法について説明します。このジョインは、外積テーブルへの制限の適用と考えることができます。

## ジョインの制限

外積ジョインを有効に利用するには、何らかの条件を満たすローだけを結果に含めます。ジョイン条件と呼ばれるこの条件では、比較演算子(=、=>、<など)を使用して、あるテーブルの 1 つのカラムを別のテーブルの 1 つのカラムと比較します。これにより、外積の結果から一部のローを除外します。

たとえば、前の項のジョインを有効に利用するには、**SalesOrders** テーブルの **SalesRepresentative** と **Employees** テーブルの従業員番号が一致するローだけを結果に含めるように指定します。これにより、各ローには注文と、その注文を担当する営業担当者の情報が格納されます。

例 1 これを実行するには、前のクエリに **WHERE** 句を追加し、従業員とその担当登録のリストを表示します。

```
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、**相関名**を使用して入力の手間を省くことができます。

## 相関名

相関名は、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。相関名を作成するには、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する必要があります。

```
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つの相関名を作成しています。

**注意** テーブル名や相関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。相関名を作成した場合は、テーブル名の代わりに必ず相関名を使用します。相関名を作成していない場合は、テーブル名を使用します。

## テーブル間の関係

他の種類のジョインを構築するには、あるテーブルの情報が別のテーブルの情報とどのように関係するかを先に理解する必要があります。

テーブルのプライマリ・キーは、そのテーブル内の各ローを識別します。各テーブルは、外部キーを使って互いに関連付けられます。

ここでは、プライマリ・キーと外部キーを使用して、複数のテーブルからクエリを構築する方法について説明します。

### 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「[第 5 章 データベース・オブジェクトの使用](#)」を参照してください)。

## ジョイン演算子

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、**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・ジョインを使用したテーブルのジョイン

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。次の例は、次のように 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を返します。

```
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。

複数のテーブルの  
ジョイン

『Sybase IQ の概要』の `iqdemo` データベースの図では、外部キーがテーブル間の線で表されています。図中で 2 つのテーブルが線で結合されていれば、`KEY JOIN` 演算子を使用できます。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。

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・ジョインを使用したテーブルのジョイン

`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。言い換えると、`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つ也没有せん。

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** テーブルの **id** カラムは、注文の ID 番号です。customer テーブルの **id** カラムは、顧客の ID 番号です。これらの番号は一致しません。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。

ジョイン演算子の使用には注意が必要です。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

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「[第6章 Sybase IQ インデックスの使用](#)」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 句で利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必要があります。
- 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

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・アドバイザは、データ型が異なるジョイン・カラムにパフォーマンス上の問題があると見なします。

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、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「[第7章 データベースへのデータの入出力](#)」を参照してください。

## ストアまたはデータベース間ジョインのサポート

この項では、ストア間またはデータベース間ジョインに対する現在のサポートを明確にします。

Sybase IQ データベース内でのテーブルのジョイン

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、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 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・ユーザーズ・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 で新しいデータベースに BLANK PADDING OFF を指定することは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・プロパティで既存のデータベースの状態をテストすることができます。

```
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

ここでは、Sybase IQ をコンポーネント統合サービス (CIS) と組み合わせて使用する方法について説明します。CIS を使用すると、Sybase IQ を通して Adaptive Server Enterprise データベースとリモート・データベースまたは非リレーショナル・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 は実際のサーバの記憶領域の場所を判別します。

### ❖ リモート・データベースをジョインするには、次の手順に従います

- 1 『システム管理ガイド：第2巻』の手順に従って、プロキシ・テーブルを作成します。
- 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。
- 3 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
```

詳細については、「[第4章 リモート・データへのアクセス](#)」および『システム管理ガイド：第2巻』の「[第5章 リモート・データ・アクセス用のサーバ・クラス](#)」を参照してください。

## サブクエリによるジョインの置き換え

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。

たとえば、注文とその発注先の会社を時系列にリストする必要がある、顧客 ID の代わりに会社名を使用すると仮定します。この結果を得るには、次のようなジョインを使用します。

### ジョインの使用

1994 年の年初以降の各注文の order\_id、OrderDate、CompanyName をリストするには、次のコマンドを入力します。

```
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」と呼ばれます。

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)	State
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

ID	LineID	ProductID	Quantity	ShipDate
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 )
```

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** カラムで **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。



# クエリと削除の最適化

## この章について

この章では、クエリと削除のパフォーマンスに関する推奨事項について説明します。

## 内容

トピック名	ページ
<a href="#">クエリ構築のヒント</a>	36
<a href="#">クエリ・プラン</a>	41
<a href="#">クエリ処理の制御</a>	46
<a href="#">削除オペレーションの最適化</a>	50

## クエリ構築のヒント

クエリ構造を改善するためのヒントを次に示します。

- サブクエリを含むコマンド文をジョインとして構成することによって、実行速度を高めることができます。
- **GROUP BY** 句で複数のカラムをグループ化する場合、可能であれば、カラムに対応するユニークな値をもとに降順にカラムをリストします。これによって最適なクエリのパフォーマンスが実現されます。
- ジョイン・インデックスを使用すると、多くの場合、ジョイン・クエリはアドホック・ジョインより高速に実行されますが、より多くのディスク領域が必要となり、ロード時間が大幅に増加します。ただし、ジョイン・クエリがマルチテーブル・ジョイン・インデックスの最大のテーブルを参照しない場合や、小さいテーブルと大きいテーブルの間のロー・カウントの差が大きい場合は、通常、アドホック・ジョインの方がジョイン・インデックスよりパフォーマンスが高くなります。
- 追加のカラムを使用して、頻繁に行う計算の結果を格納すると、パフォーマンスを向上させることができます。

---

**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)。

---

## UNION ALL での GROUP BY がクエリ・パフォーマンスに与える影響

ロード・パフォーマンスを向上させるために、非常に大きなテーブルを複数の小さなテーブルにセグメント化し、ビューで UNION ALL を使用してアクセスすることがあります。このようなビューを GROUP BY とともに使用する特定の非常に個別的なクエリでは、Sybase IQ オプティマイザがいくつかの GROUP BY 操作を UNION ALL の各分岐にコピーして、操作を並列に実行し、結果を結合することでパフォーマンスを向上させることができます。分割 GROUP BY と呼ばれるこの方法では、最上位レベルの GROUP BY で処理されるデータの量が減少し、その結果、クエリ処理時間が減少します。

パフォーマンスが向上するのは、UNION ALL で GROUP BY を使用する特定のクエリだけです。たとえば、次の簡単なクエリは分割 GROUP BY によってパフォーマンスが向上します。

```
CREATE VIEW vtable (v1 int, v2 char(4)) AS
SELECT a1, a2 FROM tableA
UNION ALL
SELECT b1, b2 FROM tableB;

SELECT COUNT(*), SUM(v1) FROM vtable GROUP BY v2;
```

このクエリを分析するときに、オブティマイザは先に tableA で COUNT(\*) GROUP BY を実行し、tableB で COUNT(\*) GROUP BY を実行した後、結果を最上位レベルの GROUP BY に渡します。最上位レベルの GROUP BY は、2つの COUNT(\*) の結果の SUM を実行し、最終的なクエリ結果を生成します。最上位レベルの GROUP BY の役割が変化していることに注意してください。最上位レベルの GROUP BY が使用している集合関数は COUNT ではなく SUM です。

#### 分割 GROUP BY の制限

分割 GROUP BY によってパフォーマンスが向上する状況とクエリには、いくつかの制限があります。

- クエリで UNION ではなく UNION ALL を使用している場合に、分割 GROUP BY によってクエリのパフォーマンスが向上する可能性があります。次のクエリでは UNION で GROUP BY を使用しているため、分割 GROUP BY によるメリットはありません。

```
CREATE VIEW viewA (va1 int, va2 int, va3 int,
va4 int) AS
SELECT b1, b2, b3, b4 FROM tableB
UNION
SELECT c1, c2, c3, c4 FROM tableC;

SELECT SUM(va1) FROM viewA GROUP BY va3;
```

- クエリ内の集合関数で DISTINCT が指定されていない場合に、分割 GROUP BY によってクエリのパフォーマンスが向上する可能性があります。次のクエリでは SUM DISTINCT を使用しているため、分割 GROUP BY によるメリットはありません。

```
CREATE VIEW viewA (va1 int, va2 int, va3 int,
va4 int) AS
SELECT b1, b2, b3, b4 FROM tableB
UNION ALL
SELECT c1, c2, c3, c4 FROM tableC;

SELECT SUM(DISTINCT va1) FROM viewA GROUP BY va3;
```

- 分割 GROUP BY によってクエリのパフォーマンスを向上させるには、追加の GROUP BY 演算子の処理に使われる集合情報とデータを格納するために、テンポラリ共有バッファ・キャッシュに十分なメモリが必要です。

```
CREATE VIEW viewA (va1 int, va2 int, va3 int,
va4 int) AS
SELECT b1, b2, b3, b4 FROM tableB
UNION ALL
SELECT c1, c2, c3, c4 FROM tableC
UNION ALL
SELECT d1, d2, d3, d4 FROM tableD
UNION ALL
SELECT e1, e2, e3, e4 FROM tableE
UNION ALL
SELECT f1, f2, f3, f4 FROM tableF
UNION ALL
SELECT g1, g2, g3, g4 FROM tableG;

SELECT SUM(va1) FROM viewA GROUP BY va3;
```

この例では、Sybase IQ オプティマイザが GROUP BY を分割し、6 個の GROUP BY 演算子をクエリ・プランに挿入しています。これにより、集合情報とデータを格納するために、クエリにより多くのテンポラリ・キャッシュが必要となります。システムが十分なキャッシュを割り付けられない場合、オプティマイザは GROUP BY を分割しません。

メモリに空きがある場合は、TEMP\_CACHE\_MEMORY\_MB データベース・オプションを使用してテンポラリ・キャッシュのサイズを増やすことができます。バッファ・キャッシュのサイズ設定については、「[バッファ・キャッシュ・サイズの決定](#)」(58 ページ)を参照してください。

- 分割 GROUP BY によってクエリのパフォーマンスを向上させるには、AGGREGATION\_PREFERENCE データベース・オプションをデフォルト値の 0 に設定します。これにより、Sybase IQ オプティマイザは GROUP BY に適用する最善のアルゴリズムを判断できるようになります。Sybase IQ オプティマイザが GROUP BY の処理にソート・アルゴリズムを選択するように AGGREGATION\_PREFERENCE の値が設定されている場合は、分割 GROUP BY によるメリットはありません。AGGREGATION\_PREFERENCE オプションを使用すると、オプティマイザが GROUP BY の処理に選択するアルゴリズムを上書きできます。分割 GROUP BY では、この値を 1 または 2 に設定しないでください。

## 分割 GROUP BY の例

次の例では、tableA という大きなテーブルを、tabA1、tabA2、tabA3、tabA4 という 4 つの小さなテーブルにセグメント化しています。この 4 つの小さなテーブルと UNION ALL を使用して、unionTab ビューを作成します。

```
CREATE VIEW unionTab (v1 int, v2 int, v3 int, v4 int) AS
SELECT a, b, c, d FROM tabA1
UNION ALL
SELECT a, b, c, d FROM tabA2
UNION ALL
SELECT a, b, c, d FROM tabA3
UNION ALL
SELECT a, b, c, d FROM tabA4;
```

Sybase IQ オプティマイザは GROUP BY 操作を次のクエリに分割し、クエリのパフォーマンスを向上させます。

```
SELECT v1, v2, SUM(v3), COUNT(*) FROM unionTab
GROUP BY v1, v2;

SELECT v3, SUM(v1*v2) FROM unionTab
GROUP BY v3;
```

## ORDER BY クエリ・パフォーマンスの強化

マルチカラム HG インデックスを使用して、単一テーブル・クエリ内の複数のカラムへの参照がある ORDER BY クエリのパフォーマンスを強化します。この変更をユーザが意識することはありませんが、クエリ・パフォーマンスが向上します。

ORDER BY 句に複数のカラムが含まれるクエリは、マルチカラム HG インデックスを使用した方が処理速度が向上する可能性があります。たとえばテーブル T にマルチカラム・インデックス HG(x,y,z) がある場合、このインデックスは射影の順序付けに使用されます。

```
SELECT abs (x) FROM T
ORDER BY x, y
```

上記の例で、HG インデックスは x と y をソート順に縦に表示します。

ROWID() 関数が SELECT リスト式内にある場合、マルチカラム HG インデックスも使用されます。例を示します。

```
SELECT rowid()+x, z FROM T
ORDER BY x,y,z
```

ROWID() が ORDER BY リストの末尾にあり、ROWID() を除くそのリストのカラムがインデックス内に存在し、順序付けキーが先行する HG カラムの順序に一致する場合、マルチカラム・インデックスがクエリに使用されます。例を示します。

```
SELECT z,y FROM T
ORDER BY x,y,z,ROWID()
```

## クエリ内の並列処理の機能強化

以前のバージョンの Sybase IQ では、1 つのクエリの実行中に多数の CPU を使用する機能が限られていました。Sybase IQ では、Sybase IQ クエリ・エンジンで自動並行処理される演算子の種類が大幅に増えました。この機能はデフォルトで有効になっているため、クエリ構文の変更や特定のチューニングは必要ありません。

## サブクエリのパフォーマンスの改善

サブクエリのフラット化は、オプティマイザがサブクエリの入ったクエリをジョインを使用するクエリに書き換える最適化方法です。Sybase IQ によって多くのサブクエリがフラット化されますが、すべてではありません。新しいデータベース・オプションの SUBQUERY\_FLATTENING\_PREFERENCE と SUBQUERY\_FLATTENING\_PERCENT でオプティマイザがこの最適化を使用する状況を制御します。

FLATTEN\_SUBQUERIES オプションは Sybase IQ 15.0 で廃止されることになりました。

## キャッシュ方法の使用

関連サブクエリにはサブクエリ外の 1 つまたは複数のテーブルへの参照が含まれており、参照されるカラムの値が変更されるたびに再実行されます。Sybase IQ では、SUBQUERY\_CACHING\_PREFERENCE オプションを使用して関連サブクエリを実行する際のキャッシュ方法を選択できます。

## SQL Anywhere による処理を引き起こす条件

Sybase IQ アーキテクチャには、コンポーネント統合サービス (CIS) と呼ばれる、SQL Anywhere のルールに従ってクエリを処理する製品の部分が含まれています。この CIS (以前は OMNI) 機能を使用すると、Sybase IQ のセマンティックで直接サポートされないクエリを処理できますが、パフォーマンスが大幅に低下します。

CIS は次のクエリを傍受します。

- ユーザ定義関数を参照するクエリ
- データベース間のジョインまたはプロキシ・テーブルを含むクエリ
- 特定のシステム関数を含むクエリ
- カタログ・ストア・テーブルまたは SYSTEM dbspace で作成されたテーブルを参照するクエリ

Sybase IQ と SQL Anywhere の違いの詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[付録 A 他の Sybase データベースとの互換性](#)」を参照してください。

## クエリ・プラン

最も効果的な構文を使用していなくても、正しいインデックスを作成していれば、通常は Sybase IQ クエリ・オプティマイザによって、最も効率的な方法でクエリを実行できます。もちろん、クエリを正しく設計することは重要です。クエリを計画する場合に、クエリの実行速度と得られる結果の正確さが主要な問題点となります。

クエリを実行する前に、Sybase IQ クエリ・オプティマイザはクエリ・プランを作成します。Sybase IQ では、これ以降の項で説明するオプションを使用して、クエリ・プランを調査および変更し、クエリを評価できます。これらのオプションを指定する方法の詳細については、『リファレンス：文とオプション』を参照してください。

**注意** 整数値を指定できるデータベース・オプションでは、小数の *option-value* の設定が常に整数値にトランケートされます。たとえば、3.8 という値は 3 にトランケートされます。

## クエリ評価オプション

次のオプションは、クエリ・プランの評価に役立ちます。詳細については、『リファレンス：文とオプション』を参照してください。

- **INDEX\_ADVISOR** — このオプションを ON に設定すると、インデックス・アドバイザは、Sybase IQ クエリ・プランの一部として、または、クエリ・プランが無効の場合には、Sybase IQ メッセージ・ログ・ファイル内の独立したメッセージとして、推奨されるインデックスを表示します。これらのメッセージは、“Index Advisor:” という文字列で始まります。この文字列を検索することで、Sybase IQ メッセージ・ファイルからこれらのメッセージをフィルタできます。このオプションはメッセージを OWNER.TABLE.COLUMN 形式で出力します。このオプションのデフォルト設定は OFF です。

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」の「[sp\\_iqindexadvice プロシージャ](#)」も参照してください。

- **INDEX\_ADVISOR\_MAX\_ROWS** — このオプションはインデックス・アドバイザによって格納されるメッセージの数を制限します。指定した制限値に到達すると、INDEX\_ADVISOR は新しいアドバイスの保存を停止します。ただし、既存のアドバイスのカウントとタイムスタンプの更新は続行します。
- **NOEXEC** — このオプションを ON に設定すると、Sybase IQ によってクエリ・プランが生成されますが、クエリ全体が実行されません。EARLY\_PREDICATE\_EXECUTION オプションを ON に設定すると、クエリの一部は実行されます。

EARLY\_PREDICATE\_EXECUTION を OFF に設定すると、クエリを通常どおり実行する場合に比べてクエリ・プランが大きく異なる可能性があるため、OFF に設定しないでください。

- **QUERY\_DETAIL** — このオプションと、QUERY\_PLAN または QUERY\_PLAN\_AS\_HTML のいずれかを ON に設定すると、クエリ・プランの作成時にクエリに関する追加情報が表示されます。QUERY\_PLAN および QUERY\_PLAN\_AS\_HTML が OFF の場合は、このオプションは無視されます。
- **QUERY\_PLAN** — このオプションが ON に設定されている場合 (デフォルト)、Sybase IQ はクエリについてのメッセージを生成します。ジョイン・インデックスの使用法、ジョイン順序、クエリのジョイン・アルゴリズムについてのメッセージなどが生成されます。



- **QUERY\_PLAN\_TEXT\_ACCESS** — このオプションが ON の場合、Interactive SQL クライアントから IQ クエリ・プランを表示、保存、出力できます。**QUERY\_PLAN\_ACCESS\_FROM\_CLIENT** オプションが OFF の場合、クエリ・プランはキャッシュされないため、クエリ・プランが関係する他のデータベース・オプションが Interactive SQL クライアントのクエリ・プランの表示に影響を与えることはありません。このオプションはデフォルトで OFF になっています。

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[GRAPHICAL\\_PLAN 関数 \[文字列\]](#)」と「[HTML\\_PLAN 関数 \[文字列\]](#)」も参照してください。

- **QUERY\_PLAN\_AFTER\_RUN** — ON に設定されている場合、クエリの実行が完了するとクエリ・プランが出力されます。これにより、クエリの各ノードから渡された実際のローの数など、追加情報をプランに含めることができます。このオプションを使用するには、**QUERY\_PLAN** を ON にします。このオプションはデフォルトで OFF になっています。
- **QUERY\_PLAN\_AS\_HTML** — Web ブラウザで表示できるように、HTML 形式のグラフィカルなクエリ・プランを生成します。HTML 形式では、ノード間にハイパーリンクが設定されるため、*.iqmsg* ファイルのテキスト形式よりはるかに使いやすくなります。クエリ・プランのファイル名にクエリ名を含めるには、**QUERY\_NAME** オプションを使用します。このオプションはデフォルトで OFF になっています。
- **QUERY\_PLAN\_AS\_HTML\_DIRECTORY** — **QUERY\_PLAN\_AS\_HTML** が ON で、**QUERY\_PLAN\_AS\_HTML\_DIRECTORY** でディレクトリが指定されている場合、Sybase IQ は指定されたディレクトリに HTML のクエリ・プランを書き込みます。
- **QUERY\_PLAN\_TEXT\_CACHING** — プランをキャッシュできるようにリソースを制御できます。このオプションが OFF の場合 (デフォルト)、クエリ・プランは該当するユーザ接続でキャッシュされません。

**QUERY\_PLAN\_TEXT\_ACCESS** オプションを OFF にすると、**QUERY\_PLAN\_TEXT\_CACHING** の設定にかかわらず、そのユーザからの接続ではクエリ・プランはキャッシュされなくなります。

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[GRAPHICAL\\_PLAN 関数 \[文字列\]](#)」と「[HTML\\_PLAN 関数 \[文字列\]](#)」も参照してください。

- **QUERY\_TIMING** — サブクエリのタイミング統計の収集などクエリ・エンジンの反復的な機能を制御します。非常に短い関連サブクエリの場合、各サブクエリを実行するタイミングを合わせる処理のために、全体のパフォーマンスが大幅に低下するため、このオプションは、通常、OFF (デフォルト) にします。

---

**注意** クエリ・プランを生成すると、`.iqmsg` ファイルに大量のテキストが追加される場合があります。`QUERY_PLAN` が ON の場合、`QUERY_DETAIL` も ON である場合は特に、メッセージ・ログ・ラッピングまたはメッセージ・ログのアーカイブを有効にしてメッセージ・ログ・ファイルが満杯にならないようにしてください。詳細については、『システム管理ガイド：第1巻』の「[第1章 Sybase IQ システム管理の概要](#)」の「[メッセージ・ログ・ラッピング](#)」を参照してください。

---

## クエリ・ツリー

オプティマイザは、クエリ内のデータの流れを表すクエリ「ツリー」を作成します。クエリ・プランでは、クエリ・ツリーが `.iqmsg` ファイル内にテキスト形式で表示されます。オプションで、グラフィカル形式のクエリ・ツリーも作成できます。

クエリ・ツリーはノードで構成されます。それぞれのノードは処理の段階を表します。ツリーが一番下のノードはリーフ・ノードです。各リーフ・ノードは、クエリ内のテーブルまたはプリジョイン・インデックス・セットを表します。

プランの最上部にあるのは、演算子ツリーのルートです。情報はテーブルから上方向に、ジョイン、ソート、フィルタ、格納、集合、サブクエリを表す演算子を通じて流れます。

## クエリ・プランの使用

クエリ・プランの使用を開始する際は、`QUERY_PLAN_AS_HTML` オプションを ON に設定することをおすすめします。このオプションを設定すると、`.iqmsg` ファイルと同じディレクトリにグラフィカル版のクエリ・プランが作成されます。このファイルは、ほとんどの Web ブラウザで表示できます。

HTML クエリ・プランでは、ツリーの各ノードが詳細へのハイパーリンクになっています。各ボックスが上位のツリーへハイパーリンクされています。任意のノードをクリックし、プラン内をすばやく移動できます。

承認されたユーザは、クエリ・プランを Java ベースの Interactive SQL プラン・ウィンドウに表示できます。また、サーバ上の *.iqmsg* ファイルまたはクエリ・プラン・ファイルにアクセスしなくても、クエリ・プランを Interactive SQL から保存、出力できます。

SQL 関数 `GRAPHICAL_PLAN` と `HTML_PLAN` は、IQ クエリ・プランを文字列結果セットとしてそれぞれ XML フォーマットと HTML フォーマットで返します。データベース・オプション

`QUERY_PLAN_TEXT_ACCESS` と `QUERY_PLAN_TEXT_CACHING` は、新しい関数の動作を制御します。

次の方法でクエリ・プランを Interactive SQL プラン・ウィンドウから確認します。

- SQL 関数 `GRAPHICAL_PLAN` と `HTML_PLAN` を使用して、クエリ・プランを文字列結果として返させる。

クエリ・プランにアクセスするには、クエリ `SELECT`、`UPDATE`、`DELETE`、`INSERT SELECT`、`SELECT INTO` に対して SQL 関数 `GRAPHICAL_PLAN` と `HTML_PLAN` を使用します。

Interactive SQL のクエリ・プランを保存するには、`GRAPHICAL_PLAN` または `HTML_PLAN` を使用してクエリ・プランを取得し、`OUTPUT` 文を使用して出力をファイルに保存します。

保存したプランを表示するには、Interactive SQL クライアント・メニューから [ファイル] - [オープン] を選択して、プランを保存したディレクトリに移動します。[ファイル] - [印刷] を選択して、プラン・ウィンドウに表示されているプランを印刷することもできます。

詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[GRAPHICAL\\_PLAN 関数 \[文字列\]](#)」および「[HTML\\_PLAN 関数 \[文字列\]](#)」(186 ページ)を参照してください。これらのクエリ・プラン関数をサポートするオプションについては、『リファレンス：文とオプション』の「[QUERY\\_PLAN\\_TEXT\\_ACCESS オプション](#)」および「[QUERY\\_PLAN\\_TEXT\\_CACHING オプション](#)」を参照してください。

## クエリ処理の制御

すべてのユーザが、特定のクエリの処理にかかる時間に制限を設定できます。DBA 権限を持つユーザは、特定のユーザのクエリに他のクエリより高い優先度を与えることや、処理のアルゴリズムを変更し、クエリ処理の速度を操作することができます。この項で説明するオプションの詳細については、『リファレンス：文とオプション』を参照してください。

### クエリの時間制限の設定

MAX\_QUERY\_TIME オプションを設定すると、ユーザは長い時間がかかるクエリを禁止できます。指定した時間よりクエリの実行時間が長くかかった場合、Sybase IQ は適切なエラーを表示してクエリを停止します。

---

**注意** Sybase IQ では、小数の *option-value* の設定がすべて整数値にトランケートされます。たとえば、3.8 という値は 3 にトランケートされます。

---

### クエリの優先度の設定

処理をキューで待機しているクエリは、そのクエリを送信したユーザの優先度、そしてクエリが送信された順序の順に実行されます。優先度の高いクエリがすべて実行されるまで、優先度の低いキューのクエリは実行されません。

次のオプションは、クエリにユーザ別の処理の優先度を割り当てます。

- IQGOVERN\_PRIORITY — 処理キューで待機しているクエリに数字の優先度 (1、2、または 3 で、1 が最も高い) を割り当てます。
- IQGOVERN\_MAX\_PRIORITY — DBA はユーザまたはグループの IQGOVERN\_PRIORITY に上限値を設定できます。
- IQ\_GOVERN\_PRIORITY\_TIME — 優先度の高い (優先度 1 の) クエリが、指定した時間より長く -iqgovern キューで待機している場合に、優先度の高いユーザを開始できます。

クエリの優先度を調べるには、sp\_iqcontext ストアド・プロシージャによって返される IQGovernPriority 属性を確認します。

## クエリ最適化オプションの設定

次のオプションは、クエリの処理速度に影響を与えます。

- **AGGREGATION\_PREFERENCE** — 集合関数 (GROUP BY、DISTINCT、SET) を処理するためのアルゴリズムの選択を制御します。このオプションは、主に内部用として設計されているため、経験のあるデータベース管理者のみが使用してください。
- **DEFAULT\_HAVING\_SELECTIVITY\_PPM** — クエリ内のすべての HAVING 述部の選択性を設定します。これが、HAVING 句によってフィルタされるロー数についてのオブティマイザの見積もりに優先して使用されます。
- **DEFAULT\_LIKE\_MATCH\_SELECTIVITY\_PPM** — 一般的な LIKE 述部のデフォルトの選択性を設定します。たとえば、LIKE '*string%string*' (% はワイルドカード文字) のように設定します。他に選択性に関する情報が提供されておらず、一致文字列が、一連の定数文字と 1 つのワイルドカードで始まっていない場合、オブティマイザはこのオプションを参照します。
- **DEFAULT\_LIKE\_RANGE\_SELECTIVITY\_PPM** — LIKE '*string%*' という形で先行定数 LIKE 述部のデフォルトの選択性を設定します。ここで、一致文字列は一連の定数文字とその後のワイルドカード文字 (%) 1 文字で構成されます。選択性に関する情報が提供されていない場合、オブティマイザはこのオプションを参照します。
- **EARLY\_PREDICATE\_EXECUTION** — ジョインの最適化の前に単純なローカル述部が実行されるかどうかを制御します。通常は、このオプションを変更しないでください。
- **IN\_SUBQUERY\_PREFERENCE** — IN サブクエリを処理するためのアルゴリズムの選択を制御します。このオプションは、主に内部用として設計されているため、経験のあるデータベース管理者のみが使用してください。
- **INDEX\_PREFERENCE** — クエリ処理に使用するインデックスを設定します。Sybase IQ オブティマイザは、通常最適なインデックスを使用して、ローカルな WHERE 句の述部など、1 つの IQ インデックスの範囲内で処理できる操作を実行します。このオプションは、テスト目的にオブティマイザの選択を無効にするために使用します。通常の使用の際はこのオプションの値を変更しないでください。

- **JOIN\_PREFERENCE** — ジョインを処理するときのアルゴリズムの選択を制御します。このオプションは、主に内部用として設計されているため、経験のあるデータベース管理者のみが使用してください。
- **JOIN\_SIMPLIFICATION\_THRESHOLD** — ジョイン・オブティマイザの単純化が適用される前にジョインされるテーブルの最小数を制御します。通常、この値を変更する必要はありません。
- **MAX\_HASH\_ROWS** — クエリ・オブティマイザがハッシュ・アルゴリズムを使用するときに考慮する最大ロー数の推測値を設定します。デフォルトは、1,250,000 のローです。たとえば、2 つのテーブル間にジョインがあり、両方のテーブルからジョインに入力されるロー数がこのオプションで設定された値を超えると、オブティマイザはハッシュ・ジョインを選択肢から外します。  
**TEMP\_CACHE\_MEMORY\_MB** がユーザあたり 50 MB を超えるシステムの場合は、このオプションにさらに大きな値を設定します。
- **MAX\_JOIN\_ENUMERATION** — オブティマイザの単純化が適用された後で、ジョイン順序のために最適化するテーブルの最大数を設定します。通常は、このオプションを設定する必要はありません。

## ユーザ指定の条件ヒントの設定

Sybase IQ クエリ・オブティマイザは、使用可能なインデックスからの情報を使用して、クエリを実行するための適切な方式を選択します。クエリ内の各条件について、オブティマイザはインデックスを使用して条件を実行できるかどうかを決定します。条件を実行できる場合、オブティマイザはインデックスを選択し、そのテーブル上の他の条件に対する順序を決定します。これらの決定で最も重要な要因になるのは、条件の選択性、つまり条件を満たすテーブル・ローの端数です。

オブティマイザは通常、ユーザの介入なしで決定を行います。また一般的に、オプションの決定も行います。ただし、状況によっては、オブティマイザが条件の実行前にその選択性を正確に決定できない場合があります。これらの状況は通常、条件が適切なインデックスを使用できないカラムを対象としている場合、または算術演算または関数式が含まれるために条件が複雑すぎてオブティマイザが正確に予測できない場合に発生します。

構文、パラメータ、使用例については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第2章 SQL 言語の要素](#)」の「[ユーザ指定の条件ヒント](#)」を参照してください。

## 負荷のモニタリング

最適化のメタデータを提供するため、またユニーク性とプライマリ/外部キーの関係を確保するために、インデックスが作成されることがよくあります。ただし、いったんインデックスが作成されると、インデックスがもたらす利点を数量化するという難題が DBA に発生します。

複数の接続によりアクセスされるか、または長期間アクセスされる必要があるデータを一時的に記憶するために、IQ メイン・ストアにテーブルが作成されることがよくあります。このテーブルは、貴重なディスク領域を継続的に使用しているうちに忘れられてしまう可能性があります。さらに、データ・ウェアハウス内のテーブルの数が多くなりすぎたり、負荷が複雑すぎて手作業で使用状況を分析できなくなったりします。

そのため、使用されていないインデックスとテーブルは、ディスク領域の浪費、バックアップ・タイムの延長、DML パフォーマンスの低下の原因となります。

Sybase IQ には、指定された負荷の統計情報の収集と分析を行うための各種ツールが用意されています。DBA はクエリで参照されているので維持する必要があるデータベース・オブジェクトをすぐに判断できます。使用されていないテーブル、カラム、インデックスを削除して、浪費される領域の低減、DML パフォーマンスの向上、バックアップ・タイムの短縮を達成できます。

負荷モニタリングはストアド・プロシージャを使用して実現されます。ストアド・プロシージャは、収集処理を制御し、テーブルとカラムの使用状況とインデックス情報を詳細に報告します。このプロシージャは INDEX\_ADVISOR 機能を補完します。この機能はクエリのパフォーマンスを向上させる可能性があるカラム・インデックスを追加するようすすめるメッセージを生成します。推奨されたインデックスがいったん追加されると、その使用状況を追跡して、保持するだけの価値があるかどうかを判断できます。

負荷のモニタリング・プロシージャの詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[sp\\_iqcolumnuse プロシージャ](#)」、「[sp\\_iqindexadvice プロシージャ](#)」、「[sp\\_iqindexuse プロシージャ](#)」、「[sp\\_iqtableuse プロシージャ](#)」、「[sp\\_iqunusedcolumn プロシージャ](#)」、「[sp\\_iqunusedindex プロシージャ](#)」、「[sp\\_iqunusedtable プロシージャ](#)」、および「[sp\\_iqworkmon プロシージャ](#)」を参照してください。

『リファレンス：文とオプション』の「[INDEX\\_ADVISOR オプション](#)」も参照してください。

## 削除オペレーションの最適化

Sybase IQ では、HG および WD インデックスが付いたカラムでの削除オペレーションを処理するために 3 つのアルゴリズムから最適なものを選択します。

### HG 削除オペレーション

Sybase IQ は、HG (High\_Group) インデックスが付いたカラムで削除オペレーションを処理するために、次の 3 つのアルゴリズムから 1 つを選択します。

- **スモール・デリート** 非常に少数のグループからローを削除するときに最適なパフォーマンスが得られます。通常は、削除するローが 1 つだけか、HG インデックスを持つカラムに等号述部がある場合に選択されます。スモール・デリート・アルゴリズムは、HG にランダムにアクセスできます。最悪の場合、I/O はアクセスされるグループの数に比例します。
- **ミッド・デリート** いくつかのグループからローを削除するときに最適なパフォーマンスが得られます。ただし、それらのグループが十分に分散されているか、十分に少なく、あまり多くの HG ページがアクセスされないことが条件です。ミッド・デリート・アルゴリズムは、HG への順序付けられたアクセスを提供します。最悪の場合、I/O はインデックス・ページ数によって制限されます。ミッド・デリートは、削除するレコードのソートという追加的なコストを伴います。
- **ラージ・デリート** 多数のグループからローを削除するときに最適なパフォーマンスが得られます。ラージ・デリートでは、すべてのローが削除されるまで HG が順番にスキャンされます。最悪の場合、I/O はインデックス・ページ数によって制限されます。ラージ・デリートは並列処理ですが、並列処理はインデックスの内部構造および削除対象のグループの分散度によって制限されます。HG カラムの範囲述部を使用して、ラージ・デリートのスキャン範囲を狭めることができます。

#### HG 削除コスト

Sybase IQ 12.6 より前の HG 削除コスト・モデルでは、最悪の場合の I/O パフォーマンスだけが考慮されていたため、多くの場合はラージ・デリートが優先的に使用されていました。現在のコスト・モデルでは、I/O コスト、CPU コスト、使用可能なリソース、インデックス・メタデータ、並列度、クエリから使用できる述部など、多数の要素が考慮されます。



HG インデックスを持つカラムの述部を指定すると、コストが大幅に改善されます。HG コスト計算でラージ・デリート以外のアルゴリズムを選択するためには、削除によって影響を受ける重複しない個別の値の数を判定する必要があります。個別カウント数は、初めはインデックス・グループの数および削除されるローの数より少ないものと見なされます。述部は個別カウント数の改善された見積もりや、正確な見積もりでさえも提供できます。

現在のコスト計算では、ラージ・デリートにおける範囲述部の効果を考慮していません。そのため、ラージ・デリートのほうが速いケースでミッド・デリートが選択されることもあります。そうしたケースでは、必要に応じて強制的にラージ・デリート・アルゴリズムを適用できます。これについては、次の項で説明します。

#### HG 削除パフォーマンス・オプションの使用

HG\_DELETE\_METHOD オプションを使用すると、HG 削除パフォーマンスを制御できます。

HG\_DELETE\_METHOD オプションで指定されたパラメータの値は、指定した削除アルゴリズムを強制的に使用できます。

- 1 = スモール・デリート
- 2 = ラージ・デリート
- 3 = ミッド・デリート

HG\_DELETE\_METHOD データベース・オプションの詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」の「[HG\\_DELETE\\_METHOD オプション](#)」を参照してください。

## WD 削除オペレーション

Sybase IQ は、WD (Word) インデックスが付いたカラムで削除オペレーションを処理するために、次の3つのアルゴリズムから1つを選択します。

- **スモール・デリート** WD のスモール・デリートでは、削除されたローに個別の単語が少数しか含まれておらず、多くの WD ページにアクセスする必要がない場合に、最適なパフォーマンスが得られます。WD スモール・デリート・アルゴリズムは、WD への順序付けられたアクセスを実行します。最悪の場合、I/O はインデックス・ページ数によって制限されます。スモール・デリートは、削除するレコードに単語とレコード ID のソートのコストを取り込みます。

- **ミッド・デリート** WD のミッド・デリートは、WD スモール・デリートの一種で、スモール・デリートと同じ条件下では便利です。つまり、削除されたローに個別の単語が少数しかない場合です。WD のミッド・デリートでは、削除するレコード内の単語のみをソートします。このソートは並列処理ですが、並列処理は使用可能な単語数と CPU スレッド数によって制限されます。Word インデックスの場合、通常は、ミッド・デリートを使用した方がスモール・デリートより高速です。
- **ラージ・デリート** WD のラージ・デリートでは、削除されたローに個別の単語が多数含まれているために、インデックスの多数の「グループ」にアクセスする必要がある場合、最適なパフォーマンスが得られます。ラージ・デリートは、すべてのローが削除されるまで WD を順番にスキャンします。最悪の場合、I/O はインデックス・ページ数によって制限されます。ラージ・デリートは並列処理ですが、並列処理はインデックスの内部構造および削除対象のグループの分散度によって制限されます。

### WD 削除コスト

WD 削除コスト・モデルでは、I/O コスト、CPU コスト、使用可能なリソース、インデックス・メタデータ、並列度など、多数の要素が考慮されます。

WD\_DELETE\_METHOD データベース・オプションを使用すると、WD 削除パフォーマンスを制御できます。

### WD 削除パフォーマンス・オプションの使用

WD\_DELETE\_METHOD オプションで指定されたパラメータの値は、指定した削除アルゴリズムを強制的に使用できます。

- 0 = コスト・モデルで選択されたミッド・デリートまたはラージ・デリート
- 1 = スモール・デリート
- 2 = ラージ・デリート
- 3 = ミッド・デリート

WD\_DELETE\_METHOD データベース・オプションの詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」の「[WD\\_DELETE\\_METHOD オプション](#)」を参照してください。

## この章について

この章では、Sybase IQ によるメモリ、ディスク I/O、CPU の使用方法と、これらの要素間の関係について説明します。また、DBA がリソース使用量を調整してパフォーマンスをチューニングする方法についても説明します。

この章の説明は、一般的な内容となっています。ご使用のハードウェアとソフトウェアの設定に合わせて調整してください。プラットフォームごとの推奨事項については、それぞれの『インストールおよび設定ガイド』を参照してください。

## 内容

トピック名	ページ
<a href="#">パフォーマンス用語の概要</a>	54
<a href="#">パフォーマンス向上のための設計</a>	54
<a href="#">メモリ使用の概要</a>	54
<a href="#">プロセス・スレッド・モデル</a>	74
<a href="#">I/O の分散</a>	76
<a href="#">リソース使用を調整するオプション</a>	85
<a href="#">リソースを効率的に利用するための他の方法</a>	89
<a href="#">インデックスのヒント</a>	91
<a href="#">データベース・サイズと構造の管理</a>	93
<a href="#">ロードを高速化するための UNION ALL ビューの使用</a>	97
<a href="#">ネットワーク・パフォーマンス</a>	100

## パフォーマンス用語の概要

パフォーマンスとは、コンピュータ・ビジネス・アプリケーションまたは同じ環境内で動作する複数のアプリケーションの効率を表す尺度です。通常、この効率は応答時間とスループットで測定します。

応答時間とは、1つのタスクが完了するまでにかかる時間のことです。応答時間は、次の項目の影響により変化します。

- 競合の軽減と待機時間 (特にディスク I/O 待機時間) の短縮
- より高速なコンポーネントの使用
- リソースに必要な時間の短縮 (同時実行性の向上)

スループットは、一定の時間にどれだけの作業量が完了したかを表します。スループットは、通常、1秒あたりのトランザクション数で表されますが、1分、1時間、1日などの単位で測定する場合もあります。

## パフォーマンス向上のための設計

適正なデータベース設計、緻密なクエリ分析、適切なインデックス付けを行うことにより、アプリケーションはより高いパフォーマンスを発揮することができます。適正な設計を行い、適切なインデックス付け方式を選択することによって、パフォーマンスを最も向上させることができます。

その他、ハードウェアやネットワークを分析することによって、インストール環境のボトルネックを特定できます。

詳細については、「[第3章 クエリと削除の最適化](#)」を参照してください。

## メモリ使用の概要

Sybase IQ では、次のような目的でメモリを使用します。

- クエリの解析用にディスクから読み込むデータのバッファ
- フラット・ファイルからロードするときにディスクから読み込むデータののためのバッファ

- 接続、トランザクション、バッファ、データベース・オブジェクトを管理するためのオーバーヘッド

以降の各項では、オペレーティング・システムが Sybase IQ のメモリ使用をサポートする方法、さまざまな目的のために Sybase IQ でメモリを割り付ける方法、パフォーマンスを改善するためにユーザがメモリ割り付けを調整する方法、また Sybase IQ に十分なメモリが使用できるようにオペレーティング・システムを設定する手順について説明します。

## ページングによる使用可能メモリの増加

システムのメモリが不足している場合、パフォーマンスが大幅に低下することがあります。このような場合、使用可能なメモリを増やす必要があります。RDBMS ソフトウェアのように、Sybase IQ にも多くのメモリが必要です。Sybase IQ に割り付け可能なメモリが多ければ多いほど、パフォーマンスも向上します。

ただし、システム内のメモリ量には常に一定の制限があるため、データの一部のみがメモリに格納され、残りのデータはディスク上に格納されるという状況が発生します。オペレーティング・システムが、ディスク上のデータを検索して取り出し、メモリ要求に対応する必要がある場合、これをページングまたはスワッピングと呼びます。メモリを適切に管理することの主な目的は、ページングやスワッピングを回避したり、最小限に抑えたりすることです。

最も頻繁に使用されるオペレーティング・システム・ファイルは、スワップ・ファイルです。メモリが消耗している場合、オペレーティング・システムがメモリのページをディスクにスワップして、新しいデータの領域を確保します。スワップされたページを再び呼び出すと、他のページがスワップされて、要求されたメモリ・ページが元に戻ります。ユーザのディスク使用率が高い場合、スワッピングには時間がかかります。通常は、スワッピングが起らないようなメモリ編成にして、オペレーティング・システム・ファイルの使用を最小限に抑えてください。スワッピングを最小限に抑えるためのメモリの設定方法については、「[プラットフォーム固有のメモリ・オプション](#)」(70 ページ)を参照してください。

Sybase IQ では、物理メモリを最大限利用するために、データベースに対するすべての読み込みと書き込みにバッファ・キャッシュを使用します。

---

**注意** ディスク上のスワップ領域には、少なくとも物理メモリ全体を収容できるだけのサイズを確保します。

---

## スワッピングをモニタするためのユーティリティ

UNIX `vmstat` コマンド、UNIX `sar` コマンド、または Windows タスク・マネージャを使用すると、実行中のプロセス数、ページアウト回数、スワップ回数についての統計を表示できます。この統計によって得た情報を使用して、システムでページングが過度に発生していないかどうかを調べてください。必要に応じて、設定を調整します。たとえば、特殊な高速ディスクにスワップ・ファイルを配置します。

`vmstat` の出力例については、「[UNIX システムでのページングのモニタリング](#)」を参照してください。

## サーバ・メモリ

Sybase IQ によって、バッファ、トランザクション、データベース、およびサーバのヒープ・メモリが割り付けられます。共有メモリも使用できますが、非常に少量です。

オペレーティング・システム・レベルでは、Sybase IQ サーバ・メモリはヒープ・メモリで構成されます。ほとんどの場合、Sybase IQ で使用されるメモリがヒープ・メモリか共有メモリかを気にする必要はありません。メモリ割り付けは、すべて自動的に処理されます。ただし、Sybase IQ を実行する前に、オペレーティング・システム・カーネルが共有メモリを使用するように正しく構成されていることを確認してください。詳細については、プラットフォームに対応した『インストールおよび設定ガイド』を参照してください。

### マルチプレックスのメモリ管理

マルチプレックスの各サーバは、独自のホスト上にある場合と、ホストを他のサーバと共有している場合があります。複数のサーバが同じシステム上にある場合、作業負荷の処理にかかる CPU 時間は、単一の組み合わせられたサーバの場合とほとんど変わりません。しかし、独立した複数のサーバでは、単一の組み合わせられたサーバより多くの物理メモリが必要になります。これは、各サーバが使用するメモリを他のサーバが共有できないからです。

### ロード、挿入、更新、同期、削除のためのメモリ

Sybase IQ は、ほとんどのロード操作にバッファ・キャッシュ・メモリを使用します。ただし、大量のヒープ・メモリを使用するロードの種類もあります。マシンの物理メモリが過度に割り付けられるのを防ぐには、ロードが発生する操作に対して `LOAD_MEMORY_MB` データベース・オプションを設定します。テーブルにワイド・データ型の列がある場合は、特に重要です。LOAD 操作に加えて、このオプションは、INSERT、UPDATE、SYNCHRONIZE、および DELETE 操作にも影響を及ぼします。LOAD\_MEMORY\_MB オプションでは、それ以降のロードで利用できるヒープ・メモリ量の上限を MB 単位で指定します。ロードとバッファ・キャッシュの使用については、「[ロードのメモリ要件](#)」(60 ページ)を参照してください。LOAD\_MEMORY\_MB オプションの詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」を参照してください。

### プロセスの中止による共有メモリへの影響

**警告！** UNIX システムで中止されたプロセスは、自動的にクリーンアップされずに、セマフォまたは共有メモリにそのまま残されます。UNIX 上の Sybase IQ サーバを正常に停止するには、『システム管理ガイド：第1巻』の「[第2章 Sybase IQ の実行](#)」の「[データベース・サーバの停止](#)」に説明されている `stop_iq` ユーティリティを使用します。ipcs と ipcrm を使用して、異常終了後のクリーンアップを行う方法については、『システム管理ガイド：第1巻』の「[第14章 トラブルシューティングのヒント](#)」を参照してください。

## バッファ・キャッシュの管理

Sybase IQ では、バッファ・キャッシュに最も多くのメモリが必要です。Sybase IQ には、IQ ストア用とテンポラリ・ストア用の2つのバッファ・キャッシュがあります。ページング、データベースへの挿入、バックアップやリストアなどのすべてのデータベース I/O 操作にこの2つのバッファ・キャッシュが使用されます。メモリ内にあるデータは、この2つのいずれかに格納されます。すべてのユーザ接続は、これらのバッファ・キャッシュを共有します。Sybase IQ が、各接続にどのデータが関連付けられているかを追跡します。

バッファ・キャッシュの管理の詳細については、以降の項を参照してください。

- メモリ要件の計算方法については、「バッファ・キャッシュ・サイズの決定」を参照してください。
- 設定サイズがわかっているバッファ・キャッシュ・サイズの設定方法については、「バッファ・キャッシュ・サイズの設定」を参照してください。

## バッファ・キャッシュ・サイズの決定

IQ ストアとテンポラリー・ストアに指定するバッファ・キャッシュ・サイズは、さまざまな要因によって決まります。デフォルト値 (メインに 16MB、テンポラリー・キャッシュに 12MB) では、ほとんどのデータベースでサイズが不足します。アプリケーションに必要な実際の値は、次の要因によって決まります。

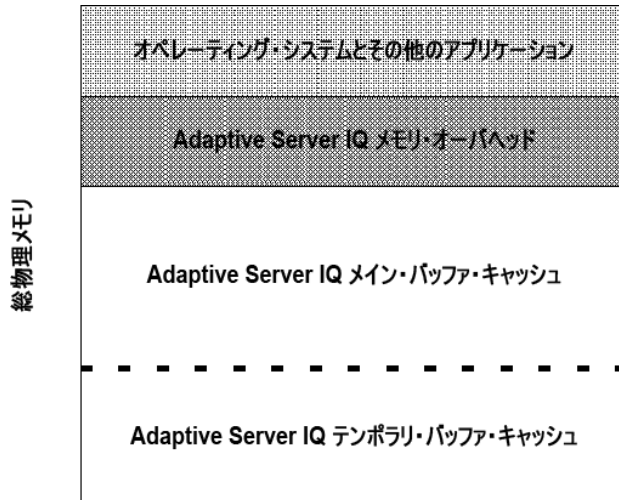
- システムに搭載されている物理メモリの合計量
- Sybase IQ、オペレーティング・システム、その他のアプリケーションがそれぞれのタスクを実行するのに必要な物理メモリの量
- ロード、クエリ、またはその両方を実行するかどうか
- スキーマ設定とクエリ負荷

サイトに最適な設定を決定するときは、これ以降の各項で示されているガイドラインを参考にしてください。

次の図は、バッファ・キャッシュとその他のメモリ消費との関係を示します。



図 4-1：物理メモリに対するバッファ・キャッシュ



各部分に関する詳細な説明、各部分に必要なメモリ量の判断に役立つガイドラインについては、以降の項で説明します。

## オペレーティング・システムとその他のアプリケーション

このメモリの量は、プラットフォームの種類やシステムの使用方法によって異なります。たとえば、UNIX ファイル・システムでは、UNIX ロー・パーティションよりも多くのファイル・バッファリングが実行されるため、オペレーティング・システムに必要なメモリは多くなります。ほとんどのオペレーティング・システムは、ファイル・システム・バッファリングに使用できるメモリの多くを使用します。使用するオペレーティング・システムのバッファリング・ポリシーを理解して、メモリの過度の割り付けを回避してください。

また、Sybase IQ とともに動作する他のアプリケーション(クエリ・ツールなど)にもそれぞれメモリ要件があります。メモリ要件については、アプリケーションおよびオペレーティング・システムのマニュアルを参照してください。

### Sybase IQ メモリ・オーバヘッド

オペレーティング・システムや他のアプリケーションで使用する物理メモリ量を決定すると、Sybase IQ がタスクの実行に必要とする残りのメモリ量を計算できます。このオーバヘッドに影響を与える要因については、以降の各項で説明します。

#### ロー・パーティションとファイル・システム

UNIX システムの場合、ロー・パーティションではなくファイル・システムを使用するデータベースには、オペレーティング・システムによるファイル・バッファリング処理のために残りのメモリの 30% がさらに必要になります。Windows では、`OS_FILE_CACHE_BUFFERING = 'OFF'` に設定し (新しいデータベースのデフォルト)、ファイル・システム・キャッシュを無効にしてください。詳細については、プラットフォームに対応した『インストールおよび設定ガイド』を参照してください。

#### マルチユーザのデータベース・アクセス

マルチユーザがデータベースをクエリする場合、Sybase IQ には「アクティブ」ユーザ 1 人あたり約 10MB のメモリが必要です。アクティブ・ユーザとは、同時にデータベースにアクセスしたり、データベースに対して問い合わせを行ったりするユーザのことです。たとえば、Sybase IQ に接続しているユーザが 30 人でも、アクティブにデータベースを同時に使用しているユーザは 10 人ほどしかいないことがあります。

#### ロードのメモリ要件

Sybase IQ では、バッファ・キャッシュの他に、ロード操作、同期、削除を実行するためのメモリも必要です。このメモリは、フラット・ファイルの I/O バッファリングに使用されます。Sybase IQ ではメモリを使用して、ディスクから読み込んだデータをバッファリングします。この読み込みのサイズは、BLOCK FACTOR に入力レコードのサイズを乗算した値になります。BLOCK FACTOR は、LOAD TABLE コマンドのオプションです。デフォルト値の 10,000 を使用した場合、データの入力ローを 200 バイトとすると、Sybase IQ が I/O バッファリングに使用するサイズは合計 2MB になります。ロードのメモリ要件は、ローの数ではなく、カラムの数と幅によって決まります。

このメモリが必要になるのは、フラット・ファイルからロードする場合、INSERT..LOCATION または INSERT..SELECT を使用する場合があります。削除と更新に必要なメモリは比較的少量です。

## スレッド・スタックのメモリ

スレッドの処理には、少量のメモリが必要です。使用する Sybase IQ 処理スレッドが多くなるにつれ、必要なメモリも多くなります。`-iqmt` サーバ・スイッチは、Sybase IQ のスレッド数を制御します。`-iqtss` サーバ・スイッチは、各スレッドに割り付けられたスタック・メモリの容量を制御します。IQ スタックに割り付けられたメモリの総量は、`-iqmt` 値と `iqtss` 値の積です。

ユーザの数に比例して、カタログ・ストア処理スレッドに必要なメモリも増加しますが、必要とされるメモリは比較的少量です。`-gn` スイッチによって、カタログ・ストア処理スレッドを制御します。

スレッドの合計数 (`-iqmt` と `-gn` の合計) が、現在のプラットフォームで利用できるスレッド数を超えないようにします。詳細については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」を参照してください。

## その他のメモリ使用

すべてのコマンドとトランザクションが、ある程度のメモリを使用します。これまで説明してきた要因の他に、メモリを大量に使用する操作には次のものがあります。

**バックアップ** バックアップに使用される仮想メモリの量は、データベース作成時に指定された `IQ PAGE SIZE` によって決まります。この値はおおよそ  $2 * \text{CPUの数} * 20 * (\text{IQ PAGE SIZE}/16)$  です。プラットフォームによっては、`BACKUP` コマンドの `BLOCK FACTOR` を調整するとバックアップのパフォーマンスが向上する場合がありますが、`BLOCK FACTOR` を増やすとメモリの使用量も増加します。詳細については、『システム管理ガイド：第1巻』の「[第12章 データのバックアップ、リカバリ、アーカイブ](#)」の「バックアップ中の使用メモリの増加」を参照してください。

**データベースの検証と修復** データベース全体を検証すると、`sp_iqcheckdb` プロシージャは処理を開始する前に、すべての Sybase IQ テーブル、テーブルのそれぞれのフィールドとインデックスを開きます。Sybase IQ テーブルの数、およびテーブル内のカラムとインデックスの累積数によって、`sp_iqcheckdb` に必要な仮想メモリの量は大幅に異なります。必要なメモリ量を制限するには、`sp_iqcheckdb` オプションを使用して1つのインデックスまたはテーブルを検証または修復します。

**リーク・ブロックの削除** リーク削除操作でも、すべての Sybase IQ テーブル、ファイル、インデックスを開く必要があるため、データベース全体を検証するときに `sp_iqcheckdb` が使用するのと同じ容量の仮想メモリを使用します。Sybase IQ テンポラリ・バッファ・キャッシュを使用して、使用ブロックを追跡します。

### Sybase IQ のメイン・バッファ・キャッシュとテンポラリ・バッファ・キャッシュ

Sybase IQ に必要なオーバーヘッド・メモリを決定したら、メイン・バッファ・キャッシュとテンポラリ・バッファ・キャッシュの間でどのように残りのメモリを分散するかを決定してください。[図 4-1](#) では、2つの領域が点線で分割されていますが、これはさまざまな要因によって分割線の位置が移動する可能性があることを示します。

他のほとんどのデータベースと異なり、Sybase IQ の一般的なガイドラインでは、メイン・バッファ・キャッシュに 40 %、テンポラリ・バッファ・キャッシュに 60% のメモリを割り付けます。ただし、この比率は固定されているわけではありません。HG インデックスが関連する大きなソートマージ・ジョインや挿入を使用するクエリなどの操作では、通常とは異なり、メイン・バッファ・キャッシュよりもテンポラリ・バッファ・キャッシュに多くのメモリが必要です。また、他のアプリケーションでは要件が異なることがあります。

---

**注意** 上記のガイドラインは、システムでアクティブなデータベースが一度に 1 つであることを前提としています。つまり、すべての Sybase IQ ユーザが 1 つのデータベースだけにアクセスしている場合です。複数のアクティブなデータベースがある場合は、使用するデータベース間で残りのメモリをさらに分ける必要があります。

---

最初は、ここに記載した一般ガイドラインに従い、「[バッファ・キャッシュのモニタリング](#)」(123 ページ) で説明されているモニタ・ツールと、現在のプラットフォームに対応する『インストールおよび設定ガイド』に記載された個別のツールを使用して、Sybase IQ のパフォーマンスをモニタリングすることを強くおすすめします。

### バッファ・キャッシュと物理メモリ

Sybase IQ のメイン・バッファ・キャッシュとテンポラリ・バッファ・キャッシュに使用するメモリと、Sybase IQ メモリ・オーバーヘッド、オペレーティング・システムとその他のアプリケーションに使用するメモリの合計が、システムの物理メモリを超えないようにしてください。

ほとんどの場合、デフォルトのテンポラリ・バッファ・キャッシュ 8 MB では小さすぎます。最良のパフォーマンスを得るためには、IQ メイン・バッファおよびテンポラリ・バッファのキャッシュにできるだけ多くのメモリを割り付けます。たとえば、使用するマシンに Sybase IQ 用に 4GB の共有メモリがある場合、共有バッファ・キャッシュをメインとテンポラリに分けることができます。

---

**注意** 一部の UNIX プラットフォームでは、他のサーバ・スイッチを設定してバッファ・キャッシュに使用可能なメモリを増やす必要があります。詳細については、「[プラットフォーム固有のメモリ・オプション](#)」(70 ページ)を参照してください。

---

## その他の注意事項

Sybase IQ のバッファ・キャッシュ・サイズは、データベースの使用状況によって異なります。パフォーマンスを最大にするには、データベースの挿入、問い合わせ、およびその両方を使用するそれぞれの場合に応じて設定を変更してください。ただし、データベースへの挿入と問い合わせを両方使用する環境では、すべてのユーザによるデータベースの使用を中止し、バッファ・キャッシュ・オプションをリセットすることは容易ではありません。このような場合は、ロードまたはクエリのどちらかのパフォーマンスを優先させてください。

バッファ・キャッシュとメモリ・オーバヘッドのガイドラインも、プラットフォームによって異なります。その他の事項については、『インストールおよび設定ガイド』を参照してください。

## バッファ・キャッシュ・サイズの設定

Sybase IQ では、メイン・バッファ・キャッシュのサイズは 32MB に、テンポラリ・バッファ・キャッシュのサイズは 24MB にデフォルト設定されています。ほとんどのアプリケーションでは、これよりはるかに大きな値を必要とします。キャッシュのサイズは、物理メモリの合計によって制限されます。使用中のシステムの適切な設定を決定するには、前の各項を参照してください。

必要な設定がわかったら、[表 4-1](#) に示すオプションを使用してバッファ・キャッシュ・サイズを設定します。[表 4-2](#) に示すオプションを使用して、バッファ・キャッシュに使用するメモリを増やすこともできます。

表 4-1：バッファ・キャッシュ・サイズを変更する設定

方法	使用時期	設定の有効期間	参照先
-iqmc および -iqtc サーバ・スイッチ	次の方法が推奨されます。データベースとサーバの動作時以外にキャッシュ・サイズを設定する。4GB を超えるキャッシュ・サイズを使用できる。  64 ビット・プラットフォームを使用している場合、またはキャッシュ・サイズのデータベース・オプションがシステムの許容量を超えて設定されている場合に特に有用。	サーバが起動してから停止するまで。	<a href="#">「バッファ・キャッシュ・サイズのサーバ・スイッチの設定」(64 ページ)</a>

表 4-2：バッファ・キャッシュに使用できるメモリ容量に影響する設定

方法	使用時期	設定の有効期間	参照先
LOAD_MEMORY_MB データベース・オプション	ロードに使用可能なメモリを制御して、バッファ・キャッシュ・サイズに間接的に影響を与える場合。プラットフォームによっては、ロードに無制限のメモリを許可すると、バッファ・キャッシュに使用可能なメモリが少なくなる。	すぐに有効 (オプションをリセットするまで)	<a href="#">「ロード、挿入、更新、同期、削除のためのメモリ」(57 ページ)</a>

## バッファ・キャッシュ・サイズのサーバ・スイッチの設定

バッファ・キャッシュ・サイズを変更するにはサーバを再起動する必要があります。-iqmc および -iqtc サーバ起動オプションは、サーバが実行されている間だけ有効なため、サーバを再起動するたびに指定する必要があります。

## ページ・サイズの指定

データベース作成時に、Sybase IQ ページのサイズを設定します。このパラメータは、バッファ・キャッシュのサイズとともに、そのデータベースのメモリ使用とディスク I/O スループットに影響します。

---

**注意** ページ・サイズは変更できません。ページ・サイズによって、一部のデータベース・オブジェクトのサイズの上限と LOB 機能を使用できるかどうかが決まります。

---

## ページ・サイズの設定

Sybase IQ では、ページ単位でデータがメモリ内外にスワップされます。データベースを作成するときに、カタログ・ストアと IQ ストアに別々のページ・サイズを指定します。テンポラリ・ストアのページ・サイズは、IQ ストアと同じです。

パフォーマンスを最大にするための Sybase IQ ページ・サイズの推奨値については、『システム管理ガイド：第1巻』の「[第5章 データベース・オブジェクトの使用](#)」の「[IQ ページ・サイズの選択](#)」を参照してください。

カタログ・ストアで発生する I/O はわずかなため、カタログ・ストアのページ・サイズがパフォーマンスに与える影響はほとんどありません。デフォルト値 4096 バイトで十分です。

IQ ページ・サイズによって、データベースのデフォルト I/O 転送ブロック・サイズと最大データ圧縮の2つのパフォーマンスの要因が決まります。以降の項で、これらの要因について説明します。

## ブロック・サイズ

すべての I/O は、ブロック単位で発生します。これらのブロックのサイズは、Sybase IQ データベースを作成したときに設定したものです。このサイズを変更するには、データベースを再作成します。デフォルトでは、IQ ページ・サイズによって I/O 転送ブロック・サイズが決まります。たとえば、デフォルト IQ ページ・サイズが 128KB の場合、デフォルトのブロック・サイズは 8,192 バイトになります。通常、Sybase IQ はこのページ・サイズに対するデフォルトのブロック・サイズの割合の使用だけでなく、他の要因も考慮します。

ほとんどのシステムでは、デフォルトのブロック・サイズを使用することによって、I/O 転送率とディスク領域の使用率のバランスを最適化できます。ただし、パフォーマンスよりもディスク領域の節約が優先されます。デフォルトのブロック・サイズでシステムのパフォーマンスが不十分な場合は、1 ページに 2 ブロック以上 16 ブロック以下という制約に従って 4096 から 32768 までの 2 の累乗の値を指定してください。次のような場合には、ブロック・サイズを明示的に設定できます。

- ディスク・アレイを使用したロー・ディスク・インストールの場合、ブロックが大きいほどパフォーマンスは向上しますが、使用されるディスク領域が多くなります。
- ファイル・システム・インストールでは、オペレーティング・システムにネイティブ・ブロック・サイズがある場合は、そのブロック・サイズ以上の IQ ブロック・サイズを設定すると、ディスク領域のパフォーマンスが最適化されます。IQ ブロック・サイズがファイル・システムのブロック・サイズと一致する場合は、I/O 率が向上する可能性もあります。

表 4-3 は、各 IQ ページ・サイズのデフォルトのブロック・サイズを示します。

**表 4-3 : デフォルトのブロック・サイズ**

<b>IQ ページ・サイズ (KB)</b>	<b>デフォルトのブロック・サイズ (バイト)</b>
64	4096
128 (新しいデータベースのデフォルト)	8192
256	16384
512	32768

## データ圧縮

Sybase IQ では、常にデータを圧縮してディスクに保存します。データ圧縮によって、ディスク領域の必要量が減少し、パフォーマンスも向上します。データ圧縮の量は、IQ ページ・サイズに基づいて自動的に決定されます。

## メモリの節約

マシンのメモリが十分でない場合は、次のように調整してメモリを節約します。



## バッファ・キャッシュ設定の縮小

バッファ・キャッシュ・サイズを小さくすることによって、メモリを節約できます。ただし、バッファ・キャッシュを小さくしすぎると、バッファの不足によって、データのロードまたはデータの問い合わせが非効率的になったり、完了できなくなったりすることがあります。

## ロードに使用するメモリの削減

LOAD\_MEMORY\_MB オプションを設定すると、ロードや他の同様の操作に使用するヒープ・メモリ量を制限できます。詳細については、「[ロード、挿入、更新、同期、削除のためのメモリ](#)」(57 ページ)を参照してください。

## ロードのためのブロッキング係数の調整

BLOCK FACTOR を使用して、フラット・ファイルからロードする場合の I/O スループットをチューニングします。LOAD コマンドの BLOCK FACTOR オプションにより、入力ファイルの作成時に使用したブロッキング係数(ブロックあたりのレコード数)を指定します。デフォルトの BLOCK FACTOR は 10,000 です。

このロード・オプションの構文は、次のとおりです。

**BLOCK FACTOR = integer**

BLOCK FACTOR を決定するときは、次のガイドラインに従ってください。

**レコード・サイズ \* BLOCK FACTOR = 必要なメモリ**

このオプションを使用するには、バッファのメモリ以外に追加メモリが必要となります。使用可能なメモリが大量にある場合、または同時にアクティブになっているユーザが 1 人のみ場合は、BLOCK FACTOR の値を大きくすることによって、ロード・パフォーマンスを向上できます。

## ユーザが多数存在する場合の最適化

Sybase IQ で処理できるユーザ接続の最大数は、32 ビット・プラットフォームでは 200、64 ビット・プラットフォームでは 1,000 です。64 ビット・システムで最大数のユーザをサポートするには、オペレーティング・システムのパラメータと `start_iq` サーバ・パラメータの両方を調整する必要があります。推奨事項については、これ以降の各項と『インストールおよび設定ガイド』を参照してください。

## ユーザが多数存在する場合の Sybase IQ コマンド・ライン・オプションの変更

次の `start_iq` のスイッチは、ユーザ数が多い場合に操作に影響を与えます。

`-gm #_connections_to_support`

`-iqgovern #_ACTIVE_queries_to_support`

`-gn #_catalog_store_front_end_threads`

`-c catalog_store_cache_size`

`-ch size`

`-cl size`

`-gm`

これは、サーバで処理できる接続の合計数です。ここで設定する接続の合計数のすべてがアクティブにデータベースを使用するのではなく、一部の接続はアイドル状態となっていることが想定されます。

`-iqgovern`

Sybase IQ には、1,000 人のユーザが接続できます。ただし、一度にクエリを許可されるユーザが少ないほど、より良いスループットを得ることができ、各ユーザが効率的に操作できるリソースを十分に確保できます。`-iqgovern` 値を指定すると、一度に実行されるクエリの最大数が制限されます。`-iqgovern` の制限を超えるユーザがクエリを発行した場合は、アクティブなクエリのいずれかが完了するまで、新しいクエリはキューイングされます。

`-iqgovern` の最適な値は、クエリの性質、CPU の数、Sybase IQ バッファ・キャッシュのサイズによって異なります。デフォルト値は  $2 \times \text{CPU の数} + 10$  です。接続ユーザ数が多い場合は、このオプションを  $2 \times \text{CPU の数} + 4$  に設定するとスループットが向上する場合があります。

**-gn** `-gn` の適正值は、`-gm` の値によって決まります。`start_iq` ユーティリティが `-gn` を計算し、値を適切に設定します。`-gn` の設定値が小さすぎると、サーバが正しく機能しなくなることがあります。`-gn` は、480 以下に設定することをおすすめします。

**-c** カタログ・ストア・バッファ・キャッシュは、カタログ・ストアの汎用メモリ・プールでもあります。MB 単位で指定するには、`-c nM` 形式を使用します。たとえば、`-c 64M` のように指定します。Sybase の推奨値は次のとおりです。

**表 4-4 : カタログ・バッファ・キャッシュの設定**

ユーザ数	プラットフォーム	-c で設定する最小値
1000 まで	64 ビットのみ	64MB
200 まで	64 ビット	48MB (64 ビットの場合の <code>start_iq</code> のデフォルト値)。ユーザ数がこれより多い場合は 64MB に設定すると有効
200 まで	32 ビット	32MB (32 ビットの場合の <code>start_iq</code> のデフォルト値)

大量の解析を必要とする特定のクエリに対応する場合など、標準のカタログ・キャッシュ・サイズでは小さすぎる場合があります。このような場合は、`-cl` と `-ch` を設定すると有効なことがあります。たとえば、32 ビット・プラットフォームでは次のように設定してみます。

```
-cl 128M
-ch 256M
```

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

**警告！** カタログ・ストアのキャッシュ・サイズを明示的に制御するには、サーバ起動用の設定ファイル (`.cfg`) または UNIX コマンド・ラインで、次のいずれか一方を実行します。両方を実行しないでください。

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

上記のパラメータをこれ以外の組み合わせで指定すると、予期しない結果が生じることがあります。

### -iqmt

スレッドの不足を防止するためにマシンに多数の CPU コアがある場合、-iqmt の値を大きくすることが必要な場合があります。指定された接続数に対して -iqmt の設定値が小さすぎる場合は、要求された接続数を処理するためにスレッド数が増加します。つまり、-gm が -iqmt を上書きします。ただし、-iqmt オプションによって Sybase IQ スレッド数が増加した場合は、「ユーザが多数存在する場合のオペレーティング・システム・パラメータの設定」で説明する制限の設定時に、このスレッド数を使用します。

### ユーザが多数存在する場合の Sybase IQ テンポラリ領域の増加

場合によっては、より多くのユーザに対応できるようにテンポラリ DB 領域を増やす必要がある場合もあります。

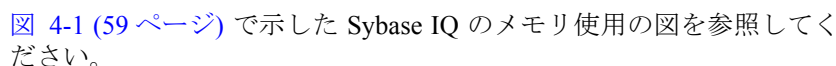
### 新規接続と既存接続との優先順位

Sybase IQ が、接続されているユーザの対応でビジー状態の場合は、新規の接続要求への応答が遅くなることがあります。たとえば、サーバが挿入のためにビジー状態のときに、テスト・スクリプトなど、ループ内の数百個の接続が起動されるような極端な例の場合、新規の接続要求がタイムアウトになることがあります。このような場合、サーバは単にビジー状態になっているだけにも関わらず、停止しているように見えます。このような状況が発生した場合は、再び接続を行い、接続タイムアウト・パラメータの値を大きくすることを検討してください。

## プラットフォーム固有のメモリ・オプション

プラットフォームの種類を問わず、Sybase IQ では次の 4 つの目的のためにメモリを使用します。

- メイン・バッファ・キャッシュ
- テンポラリ・バッファ・キャッシュ
- Sybase IQ メモリ・オーバーヘッド(スレッド・スタックを含む)
- ロード・バッファ

 **図 4-1 (59 ページ)** で示した Sybase IQ のメモリ使用の図を参照してください。

すべての 64 ビット・プラットフォームでは、使用可能な合計メモリ量は実質的に無制限です。システムの仮想メモリが唯一の制限となります。

HP-UX システムにおけるパフォーマンス・チューニングのヒントについては、そのプラットフォームの『インストールおよび設定ガイド』を参照してください。

32 ビット・プラットフォームでは制限があります。詳細については、次の表を参照してください。

**表 4-5 : 32 ビット・プラットフォームで使用可能な合計メモリ量**

プラットフォーム	使用可能メモリ量
RedHat Linux 2.1	約 1.7GB を Sybase IQ に使用可能
RedHat Linux 3.0	約 2.7GB を Sybase IQ に使用可能
Windows 2000/2003/XP <sup>a</sup>	2.75GB を Sybase IQ に使用可能

<sup>a</sup>このメモリ量を確保するには、Windows 2000 Advanced Server または Datacenter Server、Windows Server 2003 Standard、Enterprise または Datacenter Edition、もしくは Windows XP Professional を使用し、/3GB スイッチを設定する必要があります。このスイッチを設定しないと、2GB に制限されます。これはプロセスに使用できる合計メモリ量です。/3GB スイッチを設定した場合でも、Windows サーバではバッファ・キャッシュの合計サイズが 2GB を超えることはできません。詳細については、Windows 用の『インストールおよび設定ガイド』を参照してください。

Sybase IQ サーバ内での仮想メモリの使用パターンが原因で、Windows プラットフォーム上で仮想メモリの断片化によって処理が過度に増大する可能性があります。このような状況に陥る可能性を小さくするため、Sybase IQ では Windows XP と Windows Server 2003 について Microsoft の LFH (低断片化ヒープ) の使用をサポートしています。

Windows プラットフォームにおけるパフォーマンス・チューニングのその他のヒントについては、「第 6 章 32 ビット Windows システムでのサーバのチューニング」を参照してください。

UNIX システムの場合にかぎり、Sybase IQ にはメモリの管理に役立つ 2 つのコマンドライン・オプションが用意されています。

**連結メモリ・プール**

HP と Sun のプラットフォームでは、指定した量のメモリを「連結」メモリとして指定できます。連結メモリは、物理メモリにロックされた共有メモリです。カーネルはこのメモリを物理メモリからページ・アウトできません。

他のアプリケーションが同じマシン上で同時に実行されている場合は、連結メモリによって Sybase IQ のパフォーマンスが向上することがあります。ただし、連結メモリを Sybase IQ 専用割り付けると、そのメモリはマシン上の他のアプリケーションから利用できなくなります。

これらの UNIX プラットフォームにのみ「連結」メモリのプールを作成するには、`-iqwmem` コマンドライン・スイッチを指定して、連結メモリの MB 数を指定します。Sun 以外のプラットフォームで `-iqwmem` を設定するには、`root` ユーザである必要があります。64 ビット・プラットフォームでは、マシンの物理メモリのみが `-iqwmem` の上限となります。

たとえば、14GB のメモリを搭載するマシンで、10GB の連結メモリを確保するとします。そのためには、次のように指定します。

```
-iqwmem 10000
```

---

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

---

---

**注意** このバージョンでは、次の点に注意してください。

- Sun Solaris では、`-iqwmem` を指定すると、常に連結メモリが有効になります。
  - HP では、サーバを `root` ユーザで起動した場合に、`-iqwmem` を指定すると連結メモリが有効になります。`root` ユーザ以外のユーザでサーバを起動した場合は、非連結メモリが有効になります。この動作は、将来のバージョンで変更される可能性があります。
- 

### 他のアプリケーション とデータベースの影響

サーバに使用されるメモリは、すべてのアプリケーションとデータベースに使用されるメモリ・プール内のメモリです。複数のサーバまたは複数のデータベースを同時に同じマシン上で実行したり、他のアプリケーションを実行したりしている場合は、サーバが要求するメモリ量を減らす必要があります。

また、UNIX コマンド `ipcs -mb` を発行して、実際のセグメント数を表示することもできます。

### HP のメモリ問題の トラブルシューティング

HP-UX でメモリ問題が発生している場合は、`maxdsiz_64bit` カーネル・パラメータの値を調べます。このパラメータは、64 ビット HP プロセッサ上で Sybase IQ が使用できる仮想メモリの量を制限します。『インストールおよび設定ガイド』で推奨値を参照してください。

## ファイル・システム・バッファリングの制御

Solaris UFS および Windows ファイル・システムの場合にかぎり、ファイル・システム・バッファリングのオンとオフを切り替えることができます。ファイル・システム・バッファリングをオフにすると、ファイル・システム・バッファからのデータのコピーはメイン IQ バッファ・キャッシュにキャッシュされます。通常、データ・コピーの保存によってページングが減り、パフォーマンスが向上します。ただし、例外が 1 つあります。データベースの IQ ページ・サイズがファイル・システムのブロック・サイズよりも小さい場合 (通常は、テスト状況の場合のみ)、ファイル・システム・バッファリングをオフにすると、特にマルチユーザ操作中にパフォーマンスが低下することがあります。

新しく作成される Sybase IQ データベースでは、ファイル・システム・バッファリングがデフォルトでオフになります。

既存のデータベースのファイル・システム・バッファリングを無効にするには、次の文を発行します。

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING = OFF
```

このオプションは、PUBLIC グループにのみ設定できます。変更を有効にするには、データベースを停止し、再起動させてください。

**注意** Solaris には、ファイル・システム・バッファ・キャッシュのサイズを制限するカーネル・パラメータはありません。時間の経過とともにファイル・システム・バッファ・キャッシュが大きくなり、IQ バッファ・キャッシュ・ページを置き換えるため、オペレーティング・システムに過度のページング・アクティビティが発生し、Sybase IQ パフォーマンスが低下します。

Windows では、ファイル・システムを犠牲にしてアプリケーションを優先させるページング・アルゴリズムを使用します。これは、Sybase IQ パフォーマンスの向上に役立ちます。詳細については、「[第 6 章 32 ビット Windows システムでのサーバのチューニング](#)」を参照してください。

## メモリを増やすその他の方法

特定の環境においては、他のオプションを調整して Sybase IQ に使用可能なメモリを増やすことができます。

## Java 実行可能のデータベースのオプション

SET OPTION コマンドの `JAVA_HEAP_SIZE` オプションは、Java アプリケーションに対して接続ベースごとに割り付けるメモリの最大サイズ (バイト単位) を設定します。通常、接続ごとに割り付けられるメモリはユーザの作業領域として使用され、その内訳は `Java` 変数と `Java` アプリケーション・スタック領域です。Java アプリケーションの実行中、接続ごとの割り付けはデータベース・サーバの固定キャッシュを使用するため、制御できなくなった Java アプリケーションがメモリを使いすぎないようにすることが重要です。

## プロセス・スレッド・モデル

Sybase IQ では、最大限のパフォーマンスを得るために、オペレーティング・システムのカーネル・スレッドを使用します。スレッドは、ユーザ・レベルとカーネル・レベルにあります。ライトウェイト・プロセスは、カーネルでサポートされるコントロールの基本となるスレッドです。オペレーティング・システムによって、どのライトウェイト・プロセス (LWP) をどのプロセッサでいつ実行するかが決定されます。オペレーティング・システムはユーザ・スレッドのことは関知しませんが、ユーザ・スレッドが待機中か実行可能かは認識しています。

オペレーティング・システムのカーネルによって、LWP が CPU リソース上にスケジューリングされます。この場合、LWP のスケジューリング・クラスと優先度を使用します。各 LWP はカーネルによって個別にディスパッチされ、独立したシステム・コールを実行し、独立したページ・フォールトを発生させ、マルチプロセッサ・システム上で並列実行します。

高度にスレッド化された単一のプロセスが、すべての Sybase IQ ユーザの処理を実行します。Sybase IQ は、接続によって実行される処理の種類、使用可能な合計スレッド数、さまざまなオプションの設定に基づいて、各ユーザ接続にさまざまな数のカーネル・スレッドを割り当てます。



## スレッド不足エラー

発行したクエリの開始に必要なサーバ・スレッドが不足している場合は、次のようなエラーが発生します。

このクエリで利用可能なサーバ・スレッドが不十分です。

この状況は、すぐに解消される場合もあります。他のクエリが完了してからクエリを発行すると、使用可能なスレッドが増えるため、クエリが成功する場合があります。この状況が続く場合は、次の項で説明するように、サーバを再起動し、Sybase IQ スレッド数にさらに大きな値を指定する必要があります。接続数に対して `-iqmt` に設定されている値が小さすぎる可能性もあります。

## スレッド使用を管理するための Sybase IQ オプション

Sybase IQ には、スレッド使用の管理に役立つ次のオプションがあります。

- Sybase IQ に使用可能な最大スレッド数を設定するには、サーバ起動オプション `-iqmt` を設定します。デフォルト値は接続数と CPU 数によって計算され、通常、デフォルト値をそのまま使用できます。
- サーバ内の内部実行スレッドのスタック・サイズを設定するには、サーバ起動オプション `-iqtss` を設定します。通常はデフォルト値で十分ですが、複雑なクエリを実行したときに、スタックの深さがこの制限を超えていることを示すエラーが返された場合は、値を増やします。`-iqmt` および `-iqtss` の詳細については、『ユーティリティ・ガイド』の「[第1章 データベース・サーバの実行](#)」を参照してください。
- ユーザ1人あたりに使用するスレッド数の最大値を設定するには、コマンド `SET OPTION MAX_IQ_THREADS_PER_CONNECTION` を発行します。一部の操作では、スレッドの「チーム」を割り付けて使用しようとしています。次に「チーム」に使用できるスレッド数を設定するには、コマンド `SET OPTION MAX_IQ_THREADS_PER_TEAM` を発行します。特定の操作に使用するリソースの量を制御する場合にも、これらのオプションを使用します。たとえば、`INSERT`、`LOAD`、`BACKUP`、または `RESTORE` コマンドを発行する前にこのオプションを設定できます。

## I/O の分散

この項では、システムの I/O の分散がなぜ重要かについて説明します。パフォーマンスを向上させるための、ディスク・ストライピングの使用方法および別のディスク上でのファイルの検索方法について説明します。また、メッセージ・ログ・ファイルのサイズを制御する方法についても説明します。

### ロー I/O (UNIX オペレーティング・システム)

ほとんどの UNIX ファイル・システムでは、ディスクは固定サイズのパーティションに分割されます。パーティションは、オペレーティング・システムが個別にアクセスする物理的なディスク領域です。ディスク・パーティションは、通常、ファイル・システム・モード (UFS ファイル・システム) またはロー・モードの 2 つのモードでアクセスされます。ロー・モード (文字モードとも呼ばれる) ではバッファを使用しない I/O を行い、通常、読み取りまたは書き込みシステム・コールごとにデバイスに対するデータ転送を行います。UNIX ファイル・システムである UFS モードは、バッファを使用する I/O であり、バッファにデータを蓄積してからバッファ全体を一度に転送します。

作成したデータベースまたは DB 領域は、ロー・デバイスまたはファイル・システム・ファイルに格納できます。Sybase IQ は指定されたパス名から、それがロー・パーティションかファイル・システム・ファイルかを自動的に判断します。ロー・パーティションは任意のサイズに設定できます。

詳細については、『システム管理ガイド：第 1 巻』の「[第 5 章 データベース・オブジェクトの使用](#)」の「[データベース・オブジェクトの使用](#)」を参照してください。

## ディスク・ストライピングの使用

ディスク・ストライピングは、複数のディスク・ドライブに1つのファイルのデータを分散する場合に使用する一般的な方法です。ディスク・ストライピングを使用すると、連続するディスク・ブロックをストライプ・ディスク・ドライブに格納できます。ストライピングによって、1つの論理ディスクに1つ以上の物理ディスク (またはディスク・パーティション) が結合されます。ストライプ・ディスクでは、I/Oの転送が、コンポーネントの複数の物理デバイスに分散され、並列実行されます。このため、1台のディスクを使用するよりも、パフォーマンスが大幅に向上します。

ディスク・ストライピングでは、異なるディスクにブロックを格納します。最初のブロックは、最初のドライブに格納されます。2番目のブロックは、2番目のドライブに格納されます。すべてのドライブを使用すると、最初に戻り、追加のブロックを各ドライブに格納していきます。ディスク・ストライピングの最大の特長は、複数のディスク・ドライブに対してランダムにデータを分散できる点です。ストライプ・ディスクに格納されたファイルにランダムな操作を行うため、ストライプ・セットのすべてのドライブが均等にビジーになり、1秒あたりのディスク操作の数が最大となります。これは、データベース環境において非常に有効な方法です。

オペレーティング・システムやハードウェアが提供するディスク・ストライピング、または Sybase IQ に内蔵されたディスク・ストライピングを使用できます。

## UNIX におけるディスク・ストライピングの設定

ディスク・ストライピングに対応している UNIX システムには、物理ディスクをストライプ・デバイスに設定するユーティリティがあります。詳細については、UNIX または記憶管理システムのマニュアルを参照してください。

## Windows におけるディスク・ストライピングの設定

Windows システムでは、適切な SCSI-2 ディスク・コントローラの機能によって、ハードウェア・ディスク・ストライピングを使用します。ハードウェア・ストライピングをサポートしていないマシンでも、データベースに複数のディスクを使用できる場合は、Windows ストライピングを使用して、ディスク I/O を複数のディスクに分散できます。[ディスクの管理] を使用して Windows ストライピングを設定してください。

## 推奨されるディスク・ストライピング

次に、ディスク・ストライピングの基本的な規則について説明します。

- パフォーマンスを最大にするには、ストライプ・ファイル・システムのディスクを複数のディスク・コントローラに分散させる必要があります。ただし、1 台のディスク・コントローラに多くのディスクを割り当てないように注意してください。通常の SCSI マシンでは、1 台のコントローラに 2 ～ 3 台のディスクを割り当てることができます。詳細については、ご使用のハードウェアのマニュアルを参照してください。
- ディスクは、テープ・ドライブまたは CD-ROM などの低速デバイスとは異なるコントローラに配置してください。同じコントローラにディスクを配置すると、ディスク・コントローラの速度が低下します。
- ストライプのサーバ CPU 1 台あたりに 4 台のディスクを割り付けます。
- 個々のディスクは、同等のデバイスである必要があります。つまり、サイズとフォーマットが同じで、できるだけブランドのデバイスを使用する必要があります。仕様が異なると、通常は最小のディスクのサイズが使用されるため、他のディスク領域が無駄になる場合があります。また、最も低速のディスクの速度が使用されることがあります。
- 通常、ファイル・ストライピングに使用するディスクをスワップ・パーティションとして使用するなど、他の目的には使用しないでください。
- ルート・ファイル・システムを含むディスクは、決してストライプ・デバイスの一部として使用しないでください。
- パフォーマンスを最大限にするにはロー・パーティションを使用します。

通常は、可能なかぎりディスク・ストライピングを使用してください。

---

**注意** データをロードする際に最良の結果を得るには、ストライプ・ディスクにあるフラット・ファイルにデータをダンプしてから、LOAD TABLE コマンドを使用して、Sybase IQ にデータを読み込みます。

---

## 内部ストライピング

Sybase IQ では、一連の DB 領域 (ファイルまたはデバイスのロー・パーティション) にブロック単位で情報を格納します。ディスク・ストライピングを使用している場合、Sybase IQ は、使用可能な領域があるすべての DB 領域にデータを分散させます。この方法では複数のディスク・スピンドルを使用して、高速な並列ディスク書き込みを可能にします。

## ディスク・ストライピング・オプション

この項では、Sybase IQ で提供されているオプションを使用して、サードパーティ製のソフトウェアを使用しないでディスク・ストライピングを行う方法について説明します。サードパーティ製のソフトウェアとハードウェアによるディスク・ストライピングを使用している場合は、次の説明に従う必要はありません。CREATE DBSPACE コマンドに STRIPING ON オプションを指定することにより、ディスク・ストライピングを有効にできます。

### ディスク・ストライピングの ON/OFF

DB 領域の作成時にデフォルトのストライピングを変更するために使用する構文は、次のとおりです。

```
SET OPTION "PUBLIC".DEFAULT_DISK_STRIPING = { ON | OFF }
```

すべてのプラットフォームで DEFAULT\_DISK\_STRIPING オプションのデフォルト値は ON です。ディスク・ストライピングが ON で、DB 領域に使用可能な領域がある場合は、すべての DB 領域に入力データが分散されます。ディスク・ストライピングが OFF の場合は、論理ファイルの先頭から DB 領域 (ディスク・セグメント) に格納され、一度に 1 つのディスク・セグメントが格納されます。

DEFAULT\_DISK\_STRIPING の値を変更する場合、ストライピングの優先を指定しないすべての後続の CREATE DBSPACE 操作に影響を与えます。

ディスク・ストライピングがオンの場合、ALTER DBSPACE DROP コマンドを使用して DB 領域からファイルを削除できます。ただし、DB 領域を削除する前に sp\_iqemptyfile ストアド・プロシージャを使用して、DB 領域内のすべてのデータを再配置します。ディスク・ストライピングでは、データが複数のファイルに分散されるため、sp\_iqemptyfile プロセスには、多数のテーブルとインデックスの再配置が必要になることがあります。sp\_iqdbspaceinfo および sp\_iqdbspace ストアド・プロシージャを使用して、DB 領域上に存在するテーブルとインデックスを確認します。

## 複数のファイルの使用

DB 領域で複数のファイルを使用することにより、オペレーティング・システムの複数のファイルまたはパーティションに Sybase IQ とテンポラリ・データを分散できます。複数のファイルを使用することで、スループットが向上し、DB 領域の平均遅延時間が短縮します。

ALTER DBSPACE コマンドを使用して、DB 領域にファイルを追加できます。

### ファイルの追加時

可能なかぎり、DB 領域の作成時にデータを均等に分配するようすべてのファイルを割り付けます。

ファイルを後で追加する場合、Sybase IQ は古い DB 領域と新しい DB 領域の両方に新しいデータをストライプします。更新のタイプによって、ストライピングのバランスがとれる場合や、バランスが崩れたままになる場合があります。ストライピングのバランスが再びとれるかどうかは、バージョン管理で「入れ替わる」ページ数によって決まります。

## 戦略的なファイルの格納

ランダム・アクセス・ファイル専用のディスク・ドライブ数、およびこれらのファイルに対して実行される 1 秒あたりの操作数を増やすことによって、ランダム・アクセス・ファイルに関連するパフォーマンスを向上させることができます。ランダム・ファイルには、IQ ストア、テンポラリ・ストア、カタログ・ストア、プログラム (Sybase IQ 実行ファイル、ユーザおよびストアド・プロシージャ、アプリケーション)、オペレーティング・システム・ファイルのランダム・ファイルがあります。

一方、順次アクセス・ファイルに関連するパフォーマンスは、専用ディスク・ドライブに格納し、他のプロセスとの競合をなくすことによって向上させることができます。順次ファイルには、トランザクション・ログやメッセージ・ログ・ファイルがあります。

ディスク・ボトルネックを防止するために、次の注意に従ってください。

- ランダム・ディスク I/O を順次ディスク I/O から分離する。
- Adaptive Server Enterprise などの他のデータベースのプロキシ・テーブルの I/O から Sybase IQ データベース I/O を分離する。

- IQ ストア、カタログ・ストア、テンポラリ・ストア、Adaptive Server Enterprise などのプロキシ・データベースから、トランザクション・ログとメッセージ・ログを分離する。
- データベース・ファイル、テンポラリ DB 領域、トランザクション・ログ・ファイルをデータベース・サーバと同じ物理マシン上に配置する。

## トランザクション・ログ・ファイル

トランザクション・ログ・ファイルには、Sybase IQ がシステム障害から復旧するための情報が記録されています。監査には、トランザクション・ログも必要です。このファイルのデフォルトのファイル拡張子は `.log` です。

トランザクション・ログ・ファイルを移動したり、名前を変更したりするには、トランザクション・ログ・ユーティリティ (`dblog`) を使用します。構文と詳細については、『ユーティリティ・ガイド』の「[第3章 データベース管理ユーティリティ](#)」を参照してください。

---

**警告！** Sybase IQ のトランザクション・ログ・ファイルは、多くのリレーショナル・データベースのトランザクション・ログ・ファイルとは異なります。なんらかの理由で (ログ・ファイルではなく) データベース・ファイルが失われた場合は、データベースが失われます。ただし、バックアップを正しく実行している場合は、データベースを再ロードできます。

---

### トランザクション・ログのトランケーション

Sybase IQ は、システム障害からリカバリするために必要な情報をトランザクション・ログに記録します。コミットされるトランザクションごとにログに記録される情報は少量ですが、トランザクション・ログのサイズは増え続けます。データを変更するトランザクション数が多いシステムでは、時間の経過とともにログが非常に大きくなる場合があります。

ログをトランケートするには、関係する Sybase IQ サーバをオフラインにする必要があります。ログをトランケートする頻度は、実際のところ Sybase IQ システムのサポートを担当している DBA 次第であり、ログ・ファイルの増大の度合いとサイトの運用手順によって異なります。最低でも月 1 回、ログ・ファイルが 100MB を超えている場合はそれより多い頻度でログのトランケーション・プロシージャをスケジュールしてください。

表 4-6 に、Sybase IQ でトランザクション・ログをトランケートする方法を示します。

表 4-6：トランザクション・ログのトランケーション

データベースの種類	使用する方法	参照先
非マルチプレックス	-m スイッチ。これにより、すべてのデータベースで各チェックポイント後にトランザクション・ログがトランケートされる。	<a href="#">非マルチプレックス・データベースのトランザクション・ログをトランケートする</a>
マルチプレックス	DELETE_OLD_LOGS データベース・オプション	<a href="#">マルチプレックス・データベースのトランザクション・ログをトランケートする</a>
稼働中	dbbackup コマンド・ライン・ユーティリティ	『ユーティリティ・ガイド』の「バックアップ・ユーティリティ (dbbackup)」

適切な方法を使用してください。Sybase IQ データベースの複写は、本質的にトランザクション・ログの情報に依存します。このため、マルチプレックス・データベースでは DELETE\_OLD\_LOGS オプションだけを使用します (「[マルチプレックス・データベースのトランザクション・ログのトランケーション](#)」を参照してください)。また、トランザクション・ログには、Sybase の製品の保守契約を結んでいるサポート・センタが問題を診断し、再現するための有用な情報が記録されています。どちらの方法を使用する場合も、後でサポート・センタが診断するときにログが必要になった場合に備えて、既存のログのアーカイブ処理 (ログのコピーの保管) を指定してください。

非マルチプレックス・データベースのトランザクション・ログのトランケーション

-m サーバ起動スイッチを使用して、非マルチプレックス・データベースのトランザクション・ログをトランケートします。-m サーバ起動スイッチを永続的に設定したままにすることはおすすめしません。このスイッチは、トランザクション・ログのトランケーションのために Sybase IQ を起動するときだけ使用してください。これをどのように行うかは DBA 次第ですが、次に示す手順を参考にしてください。

❖ 非マルチプレックス・データベースのトランザクション・ログをトランケートする

- 1 サーバ・スイッチ .cfg ファイルのコピーを作成し、ログのトランケーション設定用のファイルであることを示す名前を付けます。このファイルを編集し、-m スイッチを追加します。



- 2 `.db` ファイルおよび `.log` ファイルのコピー作成を含めて、完全なバックアップ手順を実行します。
- 3 Sybase IQ を停止します。`iq.msg` ファイルに 'CloseDatabase' が書き込まれたことを確認します。
- 4 `-m` オプションが含まれる設定ファイルを使用して Sybase IQ を再起動します。この時点では、ユーザ・アクセスやトランザクションを許可しないでください。
- 5 Sybase IQ を停止し、`-m` オプションが設定されていない設定ファイルを使用して再起動します。

マルチプレックス・  
データベースのトラン  
ザクション・ログのト  
ランケーション

#### ❖ マルチプレックス・データベースのトランザクション・ログをトランケートする

- 1 書き込みサーバのデータベースをバックアップしていない場合は、バックアップします。
- 2 書き込みサーバで、次のように `DELETE_OLD_LOGS` オプションを設定します。

```
SET OPTION Public.Delete_Old_Logs='On'
```

- 3 書き込みサーバの `dbremote` を停止し、`-x` コマンド・ライン・スイッチを指定して再起動します (そのために、書き込みサーバのデータベース・ディレクトリにある `start_dbremote.bat` スクリプトの特別バージョンを作成します)。これにより、書き込みサーバでログがトランケートされます。例を示します。

```
cd %Server01%\mpxdb%cmd /c
start dbremote -q -v -x -o
"d:%Server01%\mpxdb%dbremote.log" -c
"uid=DBA;pwd=SQL;eng=Server01;dbf=
d:%Server01%\mpxdb%mpxdb;
links=tcip{port=1704;host=FIONA-PC}"
```

- 4 書き込みサーバで、次のように `DELETE_OLD_LOGS` オプションをオフにします。

```
SET OPTION Public.Delete_Old_Logs='Off'
```

---

**注意** クエリ・サーバのトランザクション・ログは、書き込みサーバのログが最後にトランケートされた時期にかかわらず、同期中に常にトランケートされます。

---

## メッセージ・ログ

データベースごとにメッセージ・ログ・ファイルが作成されます。このファイルのデフォルトのファイル名は、*dbname.iqmsg* です。ただし、データベースを作成するときに別の名前を付けることができます。メッセージ・ログ・ファイルは、データベースの作成後に初めてそのデータベースを起動したときに作成されます。

デフォルトでは、Sybase IQ はエラー、状態、挿入通知メッセージを含むすべてのメッセージをメッセージ・ログ・ファイルに記録します。LOAD および INSERT 文のパラメータを使用して、通知メッセージを OFF に設定できます。

サイトによっては、挿入の数、LOAD オプションと NOTIFY\_MODULUS データベース・オプションの設定、その他の条件が原因で、メッセージ・ログ・ファイルが急速に増大する傾向があります。Sybase IQ では、メッセージ・ログを循環させる、またはファイルの最大サイズを設定してアクティブな IQ メッセージ・ログが満杯になったときにログ・ファイルをアーカイブすることで、ファイルのサイズを制限できます。

ログ・ファイルの最大サイズの設定、メッセージ・ログ・ファイルのアーカイブ、およびメッセージ・ログ循環の有効化の詳細については、『システム管理ガイド：第 1 巻』の「[第 1 章 Sybase IQ システム管理の概要](#)」の「[メッセージ・ロギング](#)」を参照してください。

## 挿入、削除、同期のための作業領域

データの挿入や削除、ジョイン・インデックスの同期を行う場合、Sybase IQ では、IQ ストアに作業領域が必要となります。作業領域を必要とするトランザクションがコミットされると、この領域は他の目的に再利用されます。

通常、IQ ストアに適切な割合の空き領域が維持されるかぎり、十分な空き領域を確保できます。ただし、データを削除する場合、データのサイズやデータベース・ページ間のデータの分散によって、大きな作業領域が必要となることがあります。多数のページにデータが分散しているデータベースの大部分を削除する場合は、データベースのサイズを一時的に2倍にできます。

## 予約領域のオプションの設定

MAIN\_RESERVED\_DBSPACE\_MB と TEMP\_RESERVED\_DBSPACE\_MB の2つのデータベース・オプションは、Sybase IQ が特定の操作のために予約する領域の量を制御します。詳細については、『システム管理ガイド：第1巻』の「[第5章 データベース・オブジェクトの使用](#)」の「[IQ メイン・ストアと IQ テンポラリ・ストアの領域管理](#)」を参照してください。

## リソース使用を調整するオプション

Sybase IQ データベースの同時ユーザ数、実行するクエリ、使用可能な処理スレッドおよびメモリによって、パフォーマンス、メモリ使用、ディスク I/O に大きな影響を与える場合があります。Sybase IQ にあるリソースの使用を調整するオプションによって、さまざまなユーザ数やクエリに対応できます。次のオプションがあります。

- 現在のデータベースにのみ影響を与える SET OPTION コマンド・オプション
- データベース・サーバ全体に影響を与えるコマンドライン・オプション
- 現在の接続にのみ影響を与える接続パラメータ

パラメータや、オプションが有効になる時期、オプションを単一の接続と PUBLIC グループの両方に設定できるかどうかを含めて、これらのオプションの詳細については、『リファレンス：文とオプション』を参照してください。

テーブルの最適化については、『システム管理ガイド：第1巻』の「[第5章 データベース・オブジェクトの使用](#)」の「[記憶領域とクエリ・パフォーマンスの最適化](#)」を参照してください。

## 同時クエリの制限

-iqgovern スイッチは、サーバの同時クエリ数を指定します。これは、ライセンスによって規制される接続数とは異なります。-iqgovern スイッチを指定することによって、IQ はディスクへのバッファ・データのページングを最適化し、メモリの過剰使用を防止できます。

-iqgovern のデフォルト値は  $(2 \times \text{CPU 数}) + 10$  です。場合によっては、いろいろな値を試して最適な値を見つける必要があります。アクティブな接続が多数あるサイトの場合は、-iqgovern を多少低めに設定してみてください。

## 使用可能な CPU 数の設定

Sybase IQ 起動コマンドの -iqnumbercpus スイッチを使用すると、使用できる CPU の数を指定できます。このスイッチは、次のマシンでのみ使用することをおすすめします。

- Intel® の CPU を搭載し、ハイパースレッディングが有効になっているマシン
- オペレーティング・システムのユーティリティを使用して、Sybase IQ をマシン内の CPU の一部に制限しているマシン

詳細については、『システム管理ガイド：第 1 巻』の「[第 2 章 Sybase IQ の実行](#)」の「[CPU 数の設定](#)」を参照してください。

## クエリによるテンポラリ DB 領域の使用の制限

QUERY\_TEMP\_SPACE\_LIMIT オプションは、予測されるテンポラリ領域の使用率が、指定されたサイズを超える場合、クエリを拒否します。デフォルトでは、クエリによるテンポラリ・ストアの使用率に制限はありません。

クエリが発行されると、Sybase IQ は、クエリの解析に必要なテンポラリ領域を推定します。ソート、ハッシュ、ロー・ストアへの使用が予想されるテンポラリ結果領域の合計が、現在の

QUERY\_TEMP\_SPACE\_LIMIT 設定を超える場合、クエリは拒否され、次のようなメッセージが表示されます。

```
Query rejected because it exceeds total space resource limit
```

このオプションを 0 (デフォルト) に設定すると、制限がないため、テンポラリ領域の条件によってクエリが拒否されることはありません。

接続ごとのテンポラリ・ストアの実際の使用率を制限するため、DBA はクエリを含むすべての DML 文に

**MAX\_TEMP\_SPACE\_PER\_CONNECTION** オプションを設定します。

**MAX\_TEMP\_SPACE\_PER\_CONNECTION** は、文ごとに実際の実行時のテンポラリ・ストアの使用率をモニタして制限します。接続が

**MAX\_TEMP\_SPACE\_PER\_CONNECTION** オプションで設定されたクォータを超える場合、エラーが返され現在の文がロール・バックされます。

## 返されるローによるクエリの制限

**QUERY\_ROWS\_RETURNED\_LIMIT** オプションを設定すると、クエリ・オブティマイザは、大量のリソースを消費する可能性のあるクエリを拒否します。クエリからの結果セットがこのオプションの値を超えると推定される場合、クエリ・オブティマイザはクエリを拒否し、次のメッセージが表示されます。

```
Query rejected because it exceed resource:  
Query_Rows_Returned_Limit
```

このオプションを使用する場合は、大量のリソースを消費するクエリのみを拒否するように設定します。

## カーソルのスクロールの禁止

ホスト変数を宣言せずにカーソルのスクロールを使用すると、Sybase IQ は、クエリ結果をバッファする一時的なストア・ノードを作成します。これは、テンポラリ・ストア・バッファ・キャッシュとは異なります。百万単位のローなど、非常に多くローを検索する場合、このストア・ノードには多くのメモリが必要となります。

すべてのカーソルがスクロールしないように設定することによって、このテンポラリ・ストア・ノードを除去できます。そのためには、**FORCE\_NO\_SCROLL\_CURSORS** オプションを **ON** に設定します。数百万といった大量のローを取り出すような場合、このオプションを使用して、テンポラリ記憶領域の必要量を節約できます。このオプションは、新しいクエリに対してすぐに有効になります。

アプリケーションでカーソルのスクロールを使用しない場合、このオプションを **PUBLIC** に対して永久値として設定してください。メモリの節約になるため、クエリのパフォーマンスが大幅に向上します。

### カーソル数の制限

**MAX\_CURSOR\_COUNT** オプションは、接続が一度に使用できるカーソルの最大数を制限するリソース・ガバナーを指定します。デフォルトの値は 50 です。このオプションを 0 に設定すると、カーソル数は無制限になります。

### 文の数の制限

**MAX\_STATEMENT\_COUNT** オプションは、接続が一度に使用できる準備文の最大数を制限するリソース・ガバナーを指定します。

### キャッシュ・ページのプリフェッチ

**SET** コマンドの **PREFETCH\_BUFFER\_LIMIT** オプションは、Sybase IQ がプリフェッチ (データベース・ページの先読み) に使用できるキャッシュ・ページの数进行を定義します。このオプションのデフォルト値は 0 です。このオプションは、Sybase 製品の保守契約を結んでいるサポート・センタから指示があった場合にだけ設定してください。詳細については、『リファレンス：文とオプション』の「[PREFETCH\\_BUFFER\\_LIMIT オプション](#)」を参照してください。

**SET** コマンドの **BT\_PREFETCH\_MAX\_MISS** オプションは、特定のクエリでページのプリフェッチを継続するかどうかを決定します。HG インデックスを使用するクエリの実行速度が予想より遅い場合は、このオプションの値を徐々に増やしてみます。詳細については、『リファレンス：文とオプション』の「[BT\\_PREFETCH\\_MAX\\_MISS オプション](#)」を参照してください。

## 一般的な使用のための最適化

Sybase IQ は、開いたカーソルの数を追跡して、メモリを割り付けます。特定の状況においては、`USER_RESOURCE_RESERVATION` オプションによって、製品を使用していると思われる現在のカーソル数の最小値を調整し、テンポラリ・キャッシュから割り付けるメモリをさらに節約できます。

このオプションは、慎重な分析の結果、実際に必要であると判断された場合にのみ設定する必要があります。このオプションを設定する場合は、Sybase 製品の保守契約を結んでいるサポート・センタに連絡してください。

## プリフェッチされるローの数の制御

プリフェッチは、相対位置 1 または相対位置 0 のみをフェッチするカーソルのパフォーマンスを向上させるために使用します。2 つの接続パラメータを使用して、カーソル・プリフェッチのデフォルトを変更できます。`PrefetchRows (PROWS)` は、プリフェッチされるローの数を設定します。`PrefetchBuffer (PBUF)` は、プリフェッチされたローを格納するために、この接続に使用できるメモリを設定します。プリフェッチするローの数を増やすと、次の特定の条件ではパフォーマンスが向上する可能性があります。

- アプリケーションが数回の絶対フェッチで数多くのロー (数百ロー以上) をフェッチする場合
- アプリケーションがローを大量にフェッチし、かつ、クライアントとサーバが同じマシン上にあるか高速ネットワークで接続されている場合
- クライアント/サーバ通信がダイヤルアップ・リンクやワイド・エリア・ネットワークなどの低速ネットワークで行われている場合

## リソースを効率的に利用するための他の方法

この項では、パフォーマンスを向上させ、ディスク領域をさらに有効に活用するためのシステムの調整方法について説明します。

## マルチプレックス・データベースのディスク領域の管理

ユーザがいずれかのサーバで、古いバージョンのテーブルを必要とするトランザクションを実行している間は、Sybase IQ はそのテーブルを削除できません。このため、マルチプレックス・データベースでテーブルの更新とクエリが同時に発生すると、Sybase IQ が大量のディスク領域を消費することがあります。消費される領域の量は、データとインデックスの性質および更新の頻度によって決まります。

クエリする必要がなくなった古いバージョンを書き込みサーバが削除できるようにすれば、ディスク・ブロックを解放できます。古いテーブル・バージョンをリカバリできるように、すべてのサーバのユーザ全員が現在のトランザクションを定期的にコミットする必要があります。これで、サーバは稼働し続けることができ、すべての機能を利用できます。`sp_iqversionuse` ストアド・プロシージャを使用して、リモート・サーバで使用されているバージョンを表示できます。

## クエリ・サーバ間のロード・バランス

IQ ネットワーク・クライアントを使用して、マルチプレックス・クエリ・サーバ間のクエリ負荷のバランスをとれる場合があります。この方法では、プール内のマシンの作業負荷に応じて、各マシンにクライアント接続をディスパッチする中間システムが必要となります。

この方法を使用するには、クライアント・システムで、中間ロード・バランス・システムの IP アドレスとポート番号および汎用サーバ名を指定し、`VerifyServerName` 接続パラメータを NO に設定した特別な ODBC DSN を作成します。クライアントがこの DSN を使って接続すると、ロード・バランスは負荷が最も少ないと判断したマシンに対して接続を確立します。

クエリ・サーバのロード・バランスで使用する ODBC DSN の定義方法については、『システム管理ガイド：第 1 巻』の「[第 4 章 接続パラメータと通信パラメータ](#)」の「[VerifyServerName 通信パラメータ \(Verify\)](#)」を参照してください。



## データベース・アクセスの制限

クエリのパフォーマンスを向上させるには、可能なかぎり、データベースを読み取り専用に設定するか、重要な更新を使用頻度の少ない時間帯にスケジュールします。Sybase IQ では、テーブルへの挿入や削除を実行している間に、複数のクエリ・ユーザがそのテーブルを読み込むことができます。ただし、データベースを同時更新している間は、パフォーマンスが低下します。

## ディスクのキャッシュ

ディスク・キャッシュとは、ディスク・ブロックのコピーを一時的に格納するために、オペレーティング・システムによって使用されるメモリです。ファイル・システムに基づくディスクの読み書きは、通常、すべてディスク・キャッシュを通じて行われます。アプリケーションから見ると、ディスク・キャッシュによる読み書きは、すべて実際のディスク操作と同等です。

オペレーティング・システムは、固定された方法と動的方法を使用してディスク・キャッシュにメモリを割り付けます。固定された割り付けでは、あらかじめ規定されたメモリ量が使用されます。通常、10～15%のメモリが割り付けられます。オペレーティング・システムは通常、LRU(一番最後に使用された)アルゴリズムを使用してこの作業領域を管理します。動的割り付けでは、オペレーティング・システムが実行中にディスク・キャッシュの割り付けを決定します。これによって、できるだけ多くのメモリを有効に使用して、実際のメモリの需要とディスクのデータの必要性のバランスを保ちます。

## インデックスのヒント

以下の項では、インデックスの選択と管理に関するヒントについて説明します。詳細については、『システム管理ガイド：第1巻』の「[第6章 Sybase IQ インデックスの使用](#)」を参照してください。

## 正しいインデックス・タイプの選択

カラム・データに適したインデックス・タイプを選択することが重要です。Sybase IQ は、いくつかのインデックスを自動的に設定します。具体的には、射影を最適化するデフォルト・インデックスをすべてのカラムに設定し、UNIQUE、PRIMARY KEYS、FOREIGN KEYS に HG インデックスを設定します。これらのインデックスはいくつかの目的には役立ちますが、特定のクエリをできるだけ迅速に処理するには別のインデックスが必要となります。

Sybase IQ のクエリ・オブティマイザには、クエリの 1 つまたは複数のカラムに追加のインデックスがあるとオブティマイザに有効な場合にメッセージを生成するインデックス・アドバイザという機能があります。インデックス・アドバイザをアクティブにするには、INDEX\_ADVISOR オプションを ON に設定します。メッセージはクエリ・プランの一部として出力されます。クエリ・プランが有効になっていない場合は、メッセージ・ログ (.iqmsg) に単独のメッセージとして出力されます。出力の形式は OWNER.TABLE.COLUMN となります。詳細については、『リファレンス：文とオプション』の「[第 2 章 データベース・オプション](#)」の「[INDEX\\_ADVISOR オプション](#)」を参照してください。

カラムが列挙型 FP 記憶領域を使用していない場合、ジョイン・クエリの WHERE 句で参照されるグループ化カラムの LF または HG インデックスの作成を検討する必要があります。Sybase IQ オブティマイザは、最適なクエリ・プランを作成するために、列挙型 FP または HG/LF インデックスからのメタデータを必要とする場合があります。HAVING 句で非集合カラムが参照される場合、クエリを最適化するには、LF または HG インデックスの使用が有効です。例を示します。

```
SELECT c.name, SUM(l.price * (1 - l.discount))
FROM customer c, orders o, lineitem l
WHERE c.custkey = o.custkey
      AND o.orderkey = l.orderkey
      AND o.orderdate >= "1994-01-01"
      AND o.orderdate < "1995-01-01"
GROUP BY c.name
HAVING c.name NOT LIKE "I%"
      AND SUM(l.price * (1 - l.discount)) > 0.50
ORDER BY 2 desc
```

インデックスの追加は、記憶領域要件とロード時間の増大につながるため、クエリ・パフォーマンスが向上する場合にのみ実行してください。

## ジョイン・インデックスの使用

ユーザは同時に複数のテーブルのデータを参照することがよくあります。このデータは、クエリ作成時にジョインするか、またはジョイン・インデックスを作成することによって事前にジョインできます。常に同じ方法でジョインされるカラムにジョイン・インデックスを作成すると、クエリのパフォーマンスが向上することがあります。

ジョイン・インデックスのロードには、かなりの時間と領域が必要なので、定期的に必要となるジョインにのみジョイン・インデックスを作成します。Sybase IQ のジョイン・インデックスでは、1 対多および 1 対 1 のジョイン関係がサポートされます。

## 削除のための十分なディスク領域の確保

データ・ローを削除する場合、Sybase IQ は、削除するデータを含むデータベース・ページごとにバージョン・ページを作成します。削除トランザクションが実行されるまで、バージョンは保持されます。このため、データを削除する場合、ディスク領域の追加が必要な場合があります。詳細については、「[重複したバージョンと削除](#)」(394 ページ)を参照してください。

## データベース・サイズと構造の管理

この項では、データベース設計を改善し、データを管理するための考え方について説明します。

### データベース・サイズの管理

データベースのサイズは、作成するインデックスと格納するデータ量に大きく依存します。ユーザが発行するクエリに必要とされるインデックスをすべて作成することによって、クエリ処理を高速化できます。ただし、テーブルやインデックスが必要ない場合は、削除できます。削除することによって、ディスク領域を解放し、ロードとバックアップの速度を向上させ、バックアップに必要なアーカイブの記憶領域を小さくできます。

任意のテーブルに格納されたデータ量を制御するには、必要のないデータ・ローを最適な方法で削除してください。SQL Anywhere データベースのデータがデータベースに含まれる場合は、Anywhere データの削除を実行するだけで不要なデータを削除できます。コマンド構文は互換性があります。Sybase IQ は、Transact-SQL と互換性があるため、Adaptive Server Enterprise データベースのデータも同様に削除できます。

## インデックスの断片化の制御

内部インデックスの断片化は、インデックス・ページが最大ボリュームまで使用されていないときに発生します。

ローの断片化は、ローが削除されたときに発生することがあります。ページのロー全体を削除した場合、そのページは解放されますが、ページの一部のローが未使用の場合は、未使用領域がディスクに残ります。

テーブルに対する DML 操作 (INSERT、UPDATE、DELETE) によって、インデックスの断片化が発生します。断片化をレポートする 2 つのストアド・プロシージャがあります。

- `sp_iqrowdensity` は、デフォルト・インデックス・レベルでのローの断片化をレポートします。詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」の「[sp\\_iqrowdensity プロシージャ](#)」を参照してください。
- `sp_iqindexfragmentation` は、補助インデックス内の内部断片化をレポートします。詳細については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 7 章 システム・プロシージャ](#)」の「[sp\\_iqindexfragmentation プロシージャ](#)」を参照してください。

データベース管理者は、カラムのデフォルト・インデックスを補助する別のインデックスを作成することがあります。テーブルからローが削除されると、これらのインデックスが必要以上の領域を使用する場合があります。

どちらのプロシージャも対応策は示しません。データベース管理者は、レポートされた情報を調べて、インデックスの再作成、再編成、再構築などの対応策をとるかどうかを判断する必要があります。

## カタログ・ファイル増大の最小化

カタログ・ファイルが増大するのは正常なことで、その割合はアプリケーションとカタログの内容によって異なります。`.db` ファイルのサイズがパフォーマンスに影響を与えることはなく、`.db` ファイル内の空きページが必要に応じて再利用されます。カタログ・ファイルの増大を最小限に抑えるには、次の方法を使用します。

- `CREATE TABLE` 文で `IN SYSTEM` を使用しない。
- システム・ストアド・プロシージャを実行した後で `COMMIT` 文を発行する。
- 長時間実行されるトランザクションの最中に `COMMIT` 文を発行する。

## パフォーマンス向上のための非正規化

正規化フォームでデータベースを作成した場合、ベンチマークを実行して、意図的に正規化を解除してパフォーマンスを向上させることができます。非正規化については、以下のことが言えます。

- テーブルまたはカラムに対して可能
- 事前に正規化されていることが前提
- データの使用方法の理解が必要

非正規化する理由を次に示します。

- すべてのクエリには、「フル」セットのジョイン・データへのアクセスが必要である。
- 導出カラムの計算は複雑なため、計算のための記憶領域が必要である。

## 非正規化のリスク

非正規化を正しく行うには、アプリケーションに関する十分な知識が必要となるため、パフォーマンスに問題がある場合にのみ非正規化を実行してください。非正規化を行う場合、変更によってデータを最新の状態に保つためにどれだけの作業が必要かを考慮する必要があります。

これは、大量のデータの要約が頻繁に必要とされる意志決定支援アプリケーションと個別にデータ変更を行うトランザクション処理要求との違いを示す良い例です。非正規化を行う場合、特定の処理の効率を向上させるために、他の処理の効率が低下することがあります。

どのような非正規化の方式にも、データの整合性に問題がある可能性があるため、アプリケーションの設計時に慎重に文書化し、注意する必要があります。

## 非正規化の短所

正規化の解除には、次の短所があります。

- 非正規化を行うと、通常、検索は速くなりますが、更新は遅くなります。これは、DSS 環境ではさほど問題になりません。
- 非正規化は、必ずアプリケーションごとに行われるため、アプリケーションを変更した場合は、再評価が必要となります。
- 非正規化によって、テーブルのサイズが大きくなる場合があります。Sybase IQ では、カラム・データの格納を最適化できるため、このことは問題ではありません。詳細については、『リファレンス：文とオプション』の「[CREATE TABLE 文](#)」の「[IQ UNIQUE 制約](#)」と「[MAX\\_QUERY\\_TIME オプション](#)」を参照してください。
- 非正規化によって、コーディングが簡単になる場合と、逆に複雑になる場合があります。

## 非正規化のパフォーマンスの利点

次に、何によってパフォーマンスが向上するかを示します。

- ジョインの必要性の最小化
- 集約値の再計算 (選択時ではなくデータ変更時の計算)
- テーブル数の低減

## 非正規化の決定

非正規化を行うかどうかを決定する場合、使用中の環境におけるアプリケーションのデータ・アクセス要件と実際のパフォーマンス特性を分析する必要があります。非正規化を行う場合は、次の項目について検討します。

- 重要なクエリおよび予想される応答時間
- 使用するテーブルまたはカラム、および1アクセスあたりのロー数
- 通常のソート順
- 同時予測
- アクセス頻度が最も高いテーブルのサイズ
- 要約を計算するプロセスの有無
- パフォーマンス向上のためのジョイン・インデックス作成の有無

## ロードを高速化するための UNION ALL ビューの使用

テーブル内のすべてのローに二次的なインデックスを維持するにはコストがかかる場合、UNION ALL ビューを使用するとロード・パフォーマンスが向上します。Sybase IQ では、日付などでデータを複数のベース・テーブルに分けることができます。データは、これらの小さいテーブルにロードします。そして、UNION ALL ビューを使ってテーブルを1つの論理的な統一体に結合し、この統一体に対してクエリを実行します。

UNION ALL ビューを使用すると効率よく管理できます。たとえば、データを月ごとに分割している場合は、テーブルを削除し、UNION ALL ビューの定義を適切に更新することで、月全体のデータを削除できます。日付の範囲述部を追加することなく、年、四半期などに対応する多くのビュー定義を作成できます。

UNION ALL ビューを作成するには、ベース・テーブルを別々の物理テーブルに分割する論理的手段を選択します。最も一般的なのは、月ごとに分割する方法です。

たとえば、第一四半期のすべての月を含むビューを作成するには、次のコマンドを入力します。

```
CREATE VIEW
SELECT * JANUARY
UNION ALL
SELECT * FEBRUARY
UNION ALL
SELECT * MARCH
UNION ALL
```

月ごとに、1 つのベース・テーブル (この例では JANUARY、FEBRUARY、または MARCH) にデータをロードできます。次の月は、同じカラムと同じインデックス・タイプで構成された新しいテーブルにデータをロードします。

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

---

**注意** UNION ALL ビューに対して INSERT...SELECT を実行することはできません。このリリースの UNION ALL 演算子は、完全に並列ではありません。使用すると、クエリの並列処理が制限される場合があります。

---

## UNION ALL ビューを参照するクエリの最適化

最適化が効果を発揮するためには、UNION ALL ビューのすべてのパーティションにすべてのインデックスが定義されている必要があります。

DISTINCT を指定するクエリでは、UNION ALL ビューを使用すると、ベース・テーブルを使用するよりも実行速度が遅くなる傾向があります。

Sybase IQ には、次のような UNION ALL ビューの特許取得済みの最適化が用意されています。

- UNION ALL ビューでの分割 GROUP BY
- UNION ALL ビューへのプッシュダウン・ジョイン

UNION ALL ビューを参照するクエリのパフォーマンスを調整する必要がある場合は、JOIN\_PREFERENCE データベース・オプションを設定してください。このオプションは、UNION ALL ビュー間のジョインに影響を与えます。これらのオプションの詳細については、『リファレンス：文とオプション』の「[第2章 データベース・オプション](#)」を参照してください。



UNION を分割されたテーブルとして扱えるのは、以下の制約条件がすべて満たされている場合にかぎられます。

- 1 つまたは複数の UNION ALL が含まれる。
- UNION の各アームの FROM 句にテーブルが 1 つだけ含まれており、そのテーブルが物理ベース・テーブルである。
- UNION のどのアームにも、DISTINCT、RANK、集合関数、または GROUP BY 句がない。
- UNION の各アームに含まれる SELECT 句の中の各項目がカラムである。
- 最初の UNION アームの SELECT リスト内のカラムのデータ型のシーケンスが、UNION の後続の各アームにおけるシーケンスと同じである。

『リファレンス:文とオプション』の「[SELECT 文](#)」も参照してください。

## UNION ALL ビューのパフォーマンスの管理

UNION より下の DISTINCT を評価する最適化は、DESC 順序に適用されません。そのため、ORDER BY が DESC の場合、UNION ALL ビュー内への DISTINCT 演算子のプッシュをはじめとした一部の最適化は適用されません。たとえば、次のクエリはパフォーマンスに影響を与えます。

```
SELECT DISTINCT state FROM testVU ORDER BY state DESC;
```

このパフォーマンス上の問題を回避するには、クエリで ORDER BY の前に DISTINCT 演算子进行评估する必要があります。こうすることにより、ソート順が ASC になり、最適化を適用できるようになります。

```
SELECT c.state FROM (SELECT DISTINCT state  
                     FROM testVUA) c  
ORDER BY c.state DESC;
```

『リファレンス:文とオプション』の「[SELECT 文](#)」も参照してください。

## 大きい単一 (ファクト) テーブルのロードの改善

情報の急激な増加、およびデータへのリアルタイム・アクセスに対する需要の高まりに対応するため、Sybase IQ では、ロードされているテーブルにアクセスできるトランザクションで High\_Group (HG) インデックスと包含 (別名 WORD) (WD) インデックスをロードするパフォーマンスが大幅に強化されました。HG インデックスと WD インデックスのロード・パフォーマンスの向上は、以下のコマンドに影響があります。

- INSERT...SELECT
- INSERT...LOCATION
- LOAD
- UPDATE
- CREATE INDEX
- 更新可能なカーソル

LOAD TABLE 文では、HG インデックスと WD インデックスの並列ロードが実行されるようになったため、これまでのリリースよりも速く実行することが可能です。

## ネットワーク・パフォーマンス

以降の項では、ネットワーク・パフォーマンスの問題を解決するための方法を示します。

### 大量のデータ転送の向上

大量のデータを同時に転送すると、スループット全体が低下して、平均応答時間が増加します。次に、このような場合にパフォーマンスを向上させるための方法を示します。

- 大量のデータ転送は、できるかぎり勤務時間外に行う。
- 大量のデータ転送中は同時クエリの数を制限する。
- 大量のデータ転送するときに、クエリと挿入を同時に実行しない。

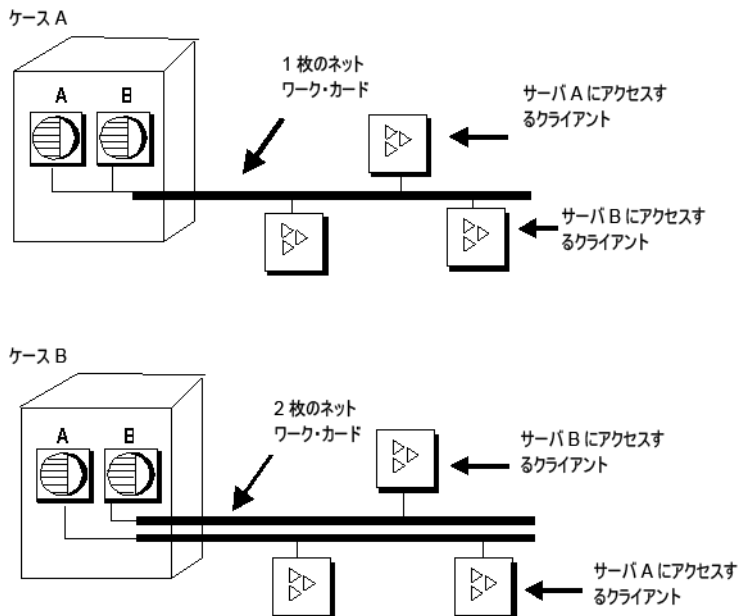
- ストアド・プロシージャを使用して、トラフィックを低減する。
- ロー・バッファリングを使用して、ネットワーク上の大きなバッチを移動する。
- 大量のデータ転送を日常的に行う場合は、そのような転送に適したネットワーク・ハードウェアの設置を検討する。例を示します。
  - トークン・リング — 大量のデータを転送する場合、イーサネットより応答が向上する。
  - 光ファイバー — 非常に高い帯域幅を提供するが、ネットワーク全体で使用するには高価すぎる。
  - 別のネットワーク — 最大ボリュームのワークステーションとサーバ間のネットワーク・トラフィックを処理するために使用する。

## ヘビー・ネットワーク・ユーザの分離

図 12-4 のケース A では、2つの異なるデータベース・サーバにアクセスするクライアントが 1 枚のネットワーク・カードを使用しています。このため、サーバ A とサーバ B にアクセスするクライアントは、ネットワーク上とネットワーク・カードで競合します。ケース B では、サーバ A にアクセスするクライアントとサーバ B にアクセスするクライアントが別々のネットワーク・カードを使用しています。

異なるマシンをデータベース・サーバにすると、さらにパフォーマンスが向上します。異なるデータベースのヘビー・ユーザを異なるマシンに分けることもできます。

図 4-2 : ヘビー・ネットワーク・ユーザの分離



## 少量のデータを小さなパケットに入れる

ネットワーク上で少量のデータを送信する場合は、デフォルトのネットワーク・パケット・サイズを小さいまま使用します (デフォルトは 512 バイトです)。-p サーバ起動オプションは、最大パケット・サイズを指定するために使用します。クライアント・アプリケーションを使用してパケット・サイズを設定できます。

## 大量のデータを大きなパケットに入れる

大量のデータを送受信するアプリケーションが多い場合は、デフォルトのネットワーク・パケット・サイズを大きくします。転送の数は少なくなりますが、データ転送量は多くなります。

## サーバ・レベルのプロセス

サーバ・レベルで、できるかぎり多くのデータをフィルタします。

# パフォーマンスのモニタリングとチューニング

## この章について

この章では、システムが使用可能なリソースを最大限に利用しているかどうかを確認するために使用するツールについて説明します。

## 内容

トピック名	ページ
<a href="#">Sybase IQ 環境の表示</a>	104
<a href="#">バッファ・キャッシュのモニタリング</a>	123
<a href="#">バッファ・キャッシュの構造</a>	136
<a href="#">バッファ・マネージャのスラッシングの回避</a>	137
<a href="#">バッファ・キャッシュ・モニタリング・チェックリスト</a>	141
<a href="#">CPU 使用率をモニタリングするシステム・ユーティリティ</a>	146

## Sybase IQ 環境の表示

Sybase IQ のパフォーマンスをチューニングする最初の手順は、環境を調べることです。これは以下の方法で行うことができます。

- システム・モニタリング・ツールを使用します (システムとサイトごとに、異なるツールが適切な場所に存在します)。
- Sybase Central の動的パフォーマンス・モニタを使用します。詳細については、「[パフォーマンス統計のモニタリング](#)」(104 ページ) を参照してください。マルチプレックス・パフォーマンスのモニタリングについては、『Sybase IQ マルチプレックスの使用』を参照してください。
- Sybase IQ についての情報を表示するいずれかのストアド・プロシージャを使用します。詳細については、「[ストアド・プロシージャを使用して情報を取得する](#)」(113 ページ) を参照してください。
- ストアド・プロシージャ、関数、イベントの実行時間を追跡するためにプロシージャ・プロファイリングを使用します。詳細については、「[データベース・プロシージャのプロファイリング](#)」(113 ページ) を参照してください。
- インデックス・タイプが適切かどうかを確認します。インデックス・タイプの選択の詳細については、『システム管理ガイド: 第 1 巻』の「[第 6 章 Sybase IQ インデックスの使用](#)」を参照してください。
- 画面に表示される情報のうち、挿入通知メッセージと削除通知メッセージを調べます。これらのメッセージの詳細については、『システム管理ガイド: 第 1 巻』の「[第 7 章 データベースへのデータの入出力](#)」を参照してください。
- Sybase IQ メッセージ・ファイルを調べます。デフォルトは `dbname.iqmsg` です。

## パフォーマンス統計のモニタリング

Sybase Central のパフォーマンス・モニタは、1 つまたは複数の参加ノードの統計のコレクションが表示されます。統計は、リアル・タイムな動的チャートで表示されます。

パフォーマンス・モニタは、次の 2 つのレベルでアクセスできます。

- マルチプレックス・レベル — マルチプレックス・サーバ全体の 1 つの統計のみをモニタできます。
- サーバ・レベル — 単一サーバまたはマルチプレックス・サーバ上で、同時に最大 10 個の統計までモニタできます。

この項では、単一サーバでのサーバ・レベルのアクセスについてのみ説明します。マルチプレックス・サーバをマルチプレックス・レベルおよびサーバ・レベルで使用するには、『Sybase IQ マルチプレックスの使用』を参照してください。

#### ❖ サーバ・レベルでのパフォーマンスのモニタリング

- Sybase Central のツリー・ビューでサーバ名をクリックし、[パフォーマンス・モニタ] タブに切り替えます。

## 統計表示のカスタマイズ

パフォーマンス・モニタ・グラフの種類や内容は変更できます。

#### ❖ モニタリングする統計の変更

サーバレベル・パフォーマンス・モニタからこのダイアログを起動すると、同時にモニタリングする統計を最大 10 項目まで選択できます (このダイアログの動作は、マルチプレックスレベルのモニタリングでは異なります。詳細については、『Sybase IQ マルチプレックスの使用』を参照してください)。

- 1 サーバをクリックして、[パフォーマンス・モニタ] タブに切り替えます。
- 2 [パフォーマンス・モニタ] タブのチャート領域を右クリックし、ショートカット・メニューから [統計の変更] を選択します。
- 3 [統計の変更] ダイアログで、モニタリングする統計を選択します。

#### ❖ ビットマップとしてのチャートの保存

チャートは JPEG イメージ・ファイルとして保存できます。

- 1 パフォーマンス・モニタのチャートを右クリックします。
- 2 ポップアップ・メニューから [名前を付けてチャートを保存] を選択します。
- 3 JPEG ファイルの名前を入力します。
- 4 [保存] をクリックします。

#### ❖ チャートの印刷

パフォーマンス・モニタのチャートを印刷できます。

- 1 パフォーマンス・モニタのチャートを右クリックします。
- 2 ポップアップ・メニューから [チャートの印刷] を選択します。
- 3 印刷オプションを確認し、印刷するプリンタを指定してページを印刷します。

#### ❖ チャート・ビューの切り替え

チャート・ビューは、時系列、2 次元棒グラフ、3 次元棒グラフから選択できます。

- 1 パフォーマンス・モニタのチャートを右クリックします。
- 2 表示されたメニューから [チャート・ビューの切り替え] を選択します。
- 3 [時系列グラフ]、[棒グラフ 2 D 縦]、または [棒グラフ 3 D 縦] を選択します。

#### ❖ チャートのカスタマイズ

チャートの設定およびチャートの更新レート (モニタ GUI のハートビート・レート) を変更します。

- 1 パフォーマンス・モニタのチャート領域を右クリックします。
- 2 [チャートのカスタマイズ] を選択します。

[チャートのカスタマイズ] ダイアログは、次のコンポーネントで構成されています。

- [期間] — チャートが時系列チャートである場合にのみ表示されます。データを表示する期間を指定します。最小値は 1 分です。最大値は 240 分 (4 時間) です。
  - [チャート更新レート] — チャートのデータを更新する頻度を秒単位で指定します。この値は、パフォーマンス・モニタ・ウィンドウ枠の下部にも表示されます。
  - [実際の値と正規化された値] — 実際の値 (デフォルト) は実際のデータを反映します。正規化された値は、チャート・データを固定範囲にスケールします。このオプションは表示のみを目的としたもので、異なる範囲の統計を 1 つのチャートでより適切に表示できます。
  - [凡例] — 凡例の表示または非表示を選択できます。
- 3 変更内容を保存するには、[OK] をクリックします。



## 統計のカテゴリ

統計は、次のカテゴリにグループ化されています。

- CPU 使用率統計
- メモリ使用状況統計
- キャッシュ統計
- スレッド統計
- 接続統計
- 要求統計
- トランザクション統計
- ストア I/O 統計
- DB 領域使用状況
- ネットワーク統計

## CPU 使用率統計

表 5-1 : CPU 使用率

名前	説明	デフォルトでの モニタリング
CPU 使用率	IQ プロセスの CPU 使用率をパーセンテージで示します。この値には、システムによる使用とユーザによる使用の両方が含まれます。	あり
CPU システム使用率	IQ プロセスの CPU システム使用率をパーセンテージで示します。	なし
CPU ユーザ使用率	IQ プロセスの CPU ユーザ使用率をパーセンテージで示します。	なし

## メモリ使用状況統計

表 5-2 : メモリ使用状況

名前	説明	デフォルトでの モニタリング
割り付けられたメモリ	IQ サーバによって割り付けられているメモリ量 (MB 単位)	あり
割り付けられる最大メモリ	IQ サーバによって割り付けられる最大のメモリ容量 (MB 単位)	なし

## キャッシュ統計

表 5-3 : キャッシュ統計

名前	説明	デフォルトでの モニタリング
カタログ・キャッシュ・ヒット数	1 秒あたりのカタログ・キャッシュ・ヒット数。	なし
テンポラリ・キャッシュ・ヒット数	1 秒あたりのテンポラリ・キャッシュ・ヒット数。	なし
メイン・キャッシュ・ヒット数	1 秒あたりのメイン・キャッシュ・ヒット数。	なし
カタログ・キャッシュ・リード数	1 秒あたりのカタログ・キャッシュ・リード数。	あり
テンポラリ・キャッシュ・リード数	1 秒あたりのテンポラリ・キャッシュ・リード数。	なし
メイン・キャッシュ・リード数	1 秒あたりのメイン・キャッシュ・リード数。	なし
現在のカタログ・キャッシュ・サイズ	現在のカタログ・キャッシュ・サイズ (MB 単位)。	なし
現在のテンポラリ・キャッシュ・サイズ	現在のテンポラリ・キャッシュ・サイズ (MB 単位)。	なし
現在のメイン・キャッシュ・サイズ	現在のメイン・キャッシュ・サイズ (MB 単位)。	なし
カタログ・キャッシュ使用率 (パーセンテージ)	パーセントで表現されたカタログ・キャッシュの使用率。	なし
テンポラリ・キャッシュ使用率 (パーセンテージ)	パーセントで表現されたテンポラリ・キャッシュの使用率。	なし

名前	説明	デフォルトでのモニタリング
メイン・キャッシュ使用率 (パーセンテージ)	パーセントで表現されたメイン・キャッシュの使用率。	なし
固定されたカタログ・キャッシュ	固定されたカタログ・キャッシュ・ページ数。	なし
固定されたテンポラリ・キャッシュ	固定されたテンポラリ・キャッシュ・ページ数。	なし
固定されたメイン・キャッシュ	固定されたメイン・キャッシュ・ページ数。	なし
カタログ・キャッシュの固定率 (パーセンテージ)	固定されたカタログ・キャッシュのパーセントで表現された比率。	なし
テンポラリ・キャッシュの固定率 (パーセンテージ)	固定されたテンポラリ・キャッシュのパーセントで表現された比率。	なし
メイン・キャッシュの固定率 (パーセンテージ)	固定されたメイン・キャッシュのパーセントで表現された比率。	なし
ダーティなカタログ・キャッシュ・ページの割合 (パーセンテージ)	パーセントで表現されたダーティなカタログ・キャッシュ・ページの比率。	なし
ダーティなテンポラリ・キャッシュ・ページの割合 (パーセンテージ)	パーセントで表現されたダーティなテンポラリ・キャッシュ・ページの比率。	なし
ダーティなメイン・キャッシュ・ページの割合 (パーセンテージ)	パーセントで表現されたダーティなメイン・キャッシュ・ページの比率。	なし

## スレッド統計

表 5-4 : スレッド統計

名前	説明	デフォルトでのモニタリング
使用中の IQ スレッド数	IQ サーバによって使用されているスレッド数。	なし
使用可能な IQ スレッド数	IQ サーバが使用できるスレッドの数。	なし
使用中の SA スレッド数	SQL Anywhere エンジンによって使用されているスレッド数。	なし

接続統計

表 5-5 : 接続統計

名前	説明	デフォルトでの モニタリング
接続数合計	ユーザ接続および INC 接続を含めた接続総数。	あり
ユーザ接続数	ユーザ接続の数。	なし
INC 受信接続数	INC 受信接続の数。	なし
INC 送信接続数	INC 送信接続の数。	なし
1 分あたりのユーザ 接続数	1 分あたりのユーザ接続 の数。	なし
1 分あたりのユーザ 切断数	1 分あたりのユーザ切断 の数。	なし

要求統計

表 5-6 : 要求統計

名前	説明	デフォルトでの モニタリング
要求	新しい要求処理または 既存の要求の処理の継 続のためにサーバに 入った 1 秒あたりの 回数。	なし
スケジュールされてい ない要求	現在キュー内で使用可 能なサーバ・スレッドを 待機している要求の数。	なし
IQ 待機オペレーション	リソース・ガバナを待機 している IQ オペレー ションの数。	なし
IQ アクティブ・オペ レーション	アクティブな IQ オペ レーションの数。	なし

## トランザクション統計

表 5-7 : トランザクション統計

名前	説明	デフォルトでの モニタリング
トランザクション数 合計	ユーザ・トランザクションおよび INC トランザクションを含めたアクティブなトランザクションの総数。	なし
ユーザ・トランザクション数	アクティブなユーザ・トランザクションの数。	なし
INC トランザクション数	アクティブな INC トランザクションの数。	なし
アクティブな LOAD TABLE 文	アクティブな LOAD TABLE 文の数。	なし

## ストア I/O 統計

表 5-8 : ストア I/O 統計

名前	説明	デフォルトでの モニタリング
カタログ・ストア・ ディスク・リード数	カタログ・ストアから読み取られた 1 秒あたりのキロバイト数。	なし
テンポラリ・ストア・ ディスク・リード数	テンポラリ・ストアから読み取られた 1 秒あたりのキロバイト数。	なし
メイン・ストア・ディ スク・リード数	メイン・ストアから読み取られた 1 秒あたりのキロバイト数。	なし
カタログ・ストア・ ディスク・ライト数	カタログ・ストアに書き込まれた 1 秒あたりのキロバイト数。	なし
テンポラリ・ストア・ ディスク・ライト数	テンポラリ・ストアに書き込まれた 1 秒あたりのキロバイト数。	なし
メイン・ストア・ディ スク・ライト数	メイン・ストアに書き込まれた 1 秒あたりのキロバイト数。	なし

DB 領域使用状況

表 5-9 : DB 領域使用状況

名前	説明	デフォルトでの モニタリング
使用中の DB 領域ファイル・サイズ	使用中の DB 領域のサイズ。DB 領域ごとにこの統計があります。	なし
使用可能な DB 領域サイズ	各 DB 領域ファイルで使用可能な空き領域をパーセンテージで示します。DB 領域のファイルごとにこの統計があります。	なし

ネットワーク統計

表 5-10 : ネットワーク統計

名前	説明	デフォルトでの モニタリング
受信バイト数	クライアント/サーバ間の通信で受信された 1 秒あたりのバイト数。	あり
展開した状態で受信されたバイト数	圧縮が無効であると仮定した場合にクライアント/サーバ間の通信で受信されたと想定される 1 秒あたりのバイト数。	なし
送信されたバイト数	クライアント/サーバ間の通信で送信された 1 秒あたりのバイト数。	あり
展開した状態で送信されたバイト数	圧縮が無効であると仮定した場合にクライアント/サーバ間の通信で送信されたと想定される 1 秒あたりのバイト数。	なし
未使用の通信バッファ数	使用可能なネットワーク通信バッファ数。	なし
通信バッファ数合計	ネットワーク通信バッファ数合計。	なし

## ストアド・プロシージャを使用して情報を取得する

Sybase IQ には、データベース情報を表示するストアド・プロシージャがいくつかあります。

- `sp_iqconnection` は、ユーザ接続とバージョンについての統計を表示します。
- `sp_iqcontext` は、実行中の文についての情報を表示します。
- `sp_iqcheckdb` は、現在のデータベースの妥当性を検査します。
- `sp_iqdbstatistics` は、最後に実行された `sp_iqcheckdb` の結果をレポートします。
- `sp_iqdbsize` は、現在のデータベースのサイズを取得します。
- `sp_iqspaceinfo` は、データベース内の各オブジェクトによる領域の使用状況を表示します。
- `sp_iqstatus` は、データベースのその他のステータス情報を表示します。
- `sp_iqtablesize` は、指定したテーブルのサイズを取得します。
- `sp_iqgroupsize` は、指定したグループのメンバをリストします。

Sybase IQ の全ストアド・プロシージャの構文の詳細と例については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』を参照してください。

## データベース・プロシージャのプロファイリング

プロシージャ・プロファイリングでは、ストアド・プロシージャ、関数、イベント、システム・トリガ、トリガの実行にかかる時間が表示されます。プロシージャの各行の実行時間も表示できます。データベース・プロファイリング情報を使用すると、チューニングによってデータベース内のどのプロシージャのパフォーマンス向上が可能かを判断できます。

プロファイリングが有効になっている場合、Sybase IQ は、使用されているストアド・プロシージャ、関数、イベント、システム・トリガ、トリガをモニタし、それらの実行時間と、それぞれの呼び出し回数を追跡します。

プロファイリング情報はサーバによってメモリに格納され、Sybase Central の [プロファイル] タブまたは Interactive SQL で表示できます。プロファイリングが有効になると、プロファイリングを無効にするかサーバが停止するまで、データベースはプロファイリング情報を収集します。

Interactive SQL でのプロファイリング情報の取得の詳細については、[「Interactive SQL でのプロシージャ・プロファイリング情報の表示」](#) (121 ページ) を参照してください。

## プロシージャ・プロファイリングの有効化

プロシージャ・プロファイリングは、すべての接続によるプロシージャとトリガの使用状況を追跡します。Sybase Central または Interactive SQL のいずれかで、プロファイリングを有効化できます。プロシージャ・プロファイリングの有効化および使用には、DBA 権限が必要です。

### ❖ プロファイリングを有効にする (Sybase Central の場合)

- 1 DBA 権限のあるユーザとしてデータベースに接続します。
- 2 左ウィンドウ枠でデータベースを選択します。
- 3 [ファイル]-[プロパティ] を選択します。  
[データベース] プロパティ・シートが表示されます。
- 4 [プロファイリング] タブで、[このデータベースでプロファイリングを可能にする] を選択します。
- 5 [OK] をクリックして、プロパティ・シートを閉じます。

#### 注意

Sybase Central でデータベースを右クリックしても、プロファイリングを有効化できます。ポップアップ・メニューから [プロファイリング]-[プロファイリングの開始] を選択します。

### ❖ プロファイリングを有効にする (SQL の場合)

- 1 DBA 権限のあるユーザとしてデータベースに接続します。
- 2 ON 設定を指定して `sa_server_option` ストアド・プロシージャを呼び出します。

たとえば、次のように入力します。

```
CALL sa_server_option ( 'procedure_profiling', 'ON')
```



もし必要なら、他の接続でのデータベースの使用を妨害せずに、特定のユーザが使用しているプロシージャを確認できます。その接続がすでに存在するか、複数のユーザが同じユーザ ID で接続する場合は、この機能が便利です。

#### ❖ プロシージャ・プロファイリングをユーザでフィルタする

- 1 DBA 権限のあるユーザとしてデータベースに接続します。
- 2 次のプロシージャを呼び出します。

```
CALL sa_server_option
('ProfileFilterUser','userid')
```

*userid* の値は、モニタするユーザの名前です。

### プロシージャ・プロファイリングのリセット

プロファイリングをリセットすると、データベースは古い情報をクリアし、プロシージャ、関数、イベント、トリガに関する新しい情報の収集をただちに開始します。

以下の項では、DBA 権限を持つユーザとしてデータベースに接続しており、プロシージャ・プロファイリングが有効になっているものとします。

#### ❖ プロファイリングをリセットする (Sybase Central の場合)

- 1 左ウィンドウ枠でデータベースを選択します。
- 2 [ファイル]-[プロパティ]を選択します。  
[データベース]プロパティ・シートが表示されます。
- 3 [プロファイリング]タブで、[すぐにリセット]をクリックします。
- 4 [OK]をクリックして、プロパティ・シートを閉じます。

#### 注意

Sybase Central で右クリックしても、プロファイリングをリセットできません。ポップアップ・メニューから [プロファイリング]-[プロファイリング情報のリセット]を選択します。

#### ❖ プロファイリングをリセットする (SQL の場合)

- RESET 設定を指定して `sa_server_option` ストアド・プロシージャを呼び出します。

たとえば、次のように入力します。

```
CALL sa_server_option ('procedure_profiling',
'RESET')
```

## プロシージャ・プロファイリングの無効化

プロファイリング情報の使用が終了したら、プロファイリングを無効にするか、プロファイリングをクリアすることができます。プロファイリングを無効にすると、データベースはプロファイリング情報の収集を停止しますが、その時点までに収集された情報は Sybase Central の [プロファイル] タブに引き続き表示されます。プロファイリングを無効にすると、データベースはプロファイリングをオフにし、Sybase Central の [プロファイル] タブからすべてのプロファイリング・データを削除します。

### ❖ プロファイリングを無効にする (Sybase Central の場合)

- 1 左ウィンドウ枠でデータベースを選択します。
- 2 [ファイル]-[プロパティ] を選択します。  
[データベース] プロパティ・シートが表示されます。
- 3 [プロファイリング] タブで、[このデータベースでプロファイリングを可能にする] オプションをクリアします。
- 4 [OK] をクリックして、プロパティ・シートを閉じます。

#### 注意

Sybase Central でデータベースを右クリックしても、プロファイリングを無効化できます。ポップアップ・メニューから [プロファイリング]-[プロファイリングの停止] を選択します。

### ❖ プロファイリングを無効にする (SQL の場合)

- OFF 設定を指定して `sa_server_option` ストアド・プロシージャを呼び出します。  
たとえば、次のように入力します。

```
CALL sa_server_option ('procedure_profiling',
'OFF')
```

### ❖ プロファイリングをクリアする (Sybase Central の場合)

- 1 左ウィンドウ枠でデータベースを選択します。
- 2 [ファイル]-[プロパティ] を選択します。  
[データベース] プロパティ・シートが表示されます。
- 3 [プロファイリング] タブで、[すぐにクリア] をクリックします。  
プロファイリングをクリアできるのは、プロファイリングが有効になっている場合だけです。
- 4 [OK] をクリックして、プロパティ・シートを閉じます。

## 注意

Sybase Central でデータベースを右クリックしても、プロファイリングをクリアできます。ポップアップ・メニューから [プロファイリング]-[プロファイリング情報のクリア] を選択します。

## ❖ プロファイリングをクリアする (SQL の場合)

- CLEAR 設定を指定して `sa_server_option` ストアド・プロシージャを呼び出します。

たとえば、次のように入力します。

```
CALL sa_server_option ('procedure_profiling',  
'CLEAR')
```

## Sybase Central でのプロファイリング情報の表示

プロシージャ・プロファイリングでは、データベース全体、特定のタイプのオブジェクト、特定のプロシージャのいずれの情報を調べたいかによって、それぞれ異なる情報を表示することができます。次のような情報を表示できます。

- データベース内のプロファイル対象のすべてのオブジェクトについての詳細
- すべてのストアド・プロシージャと関数についての詳細
- すべてのイベントについての詳細
- すべてのトリガについての詳細
- すべてのシステム・トリガについての詳細
- プロファイル対象の個別のオブジェクトについての詳細

プロファイリング情報を表示するためには、データベースに接続し、プロファイリングを有効にする必要があります。

データベース全体のプロファイリング情報を表示すると、次のカラムが表示されます。

- **名前** オブジェクトの名前をリストします。
- **所有者** オブジェクトの所有者をリストします。
- **表** トリガが属するテーブルをリストします (このカラムはデータベースの [プロファイル] タブにのみ表示されます)。
- **イベント** システム・トリガのトリガのタイプを表示します。  
Update、Delete のいずれかです。

- **種類** オブジェクトのタイプ (たとえばプロシージャ) をリストします。
- **# Exes.** 各オブジェクトが呼び出された回数をリストします。
- **#msecs.** 各オブジェクトの総実行時間をリストします。

これらのカラムには、データベース内で実行されたすべてのプロシージャに関するプロファイリング情報の要約が表示されます。プロシージャは他のプロシージャを呼び出せるため、ユーザが明示的に呼び出したものより多くの項目が存在する可能性があります。

❖ **ストア・プロシージャと関数の要約プロファイリング情報を表示する**

- 1 左ウィンドウ枠で [プロシージャと関数] フォルダを選択します。
- 2 右ウィンドウ枠で [プロファイル] タブをクリックします。

データベース内のすべてのストア・プロシージャと関数のプロファイリング情報が [プロファイル] タブに表示されます。

❖ **イベントの要約プロファイリング情報を表示する**

- 1 左ウィンドウ枠で [イベント] フォルダを開きます。  
データベース内のすべてのイベントのリストが右ウィンドウ枠の [イベント] タブに表示されます。
- 2 右ウィンドウ枠で [プロファイル] タブをクリックします。  
データベース内のすべてのイベントのプロファイリング情報が [プロファイル] タブに表示されます。

❖ **トリガの要約プロファイリング情報を表示する**

- 1 左ウィンドウ枠で [トリガ] フォルダを開きます。  
データベース内のすべてのトリガのリストが [トリガ] タブに表示されます。
- 2 右ウィンドウ枠で [プロファイル] タブをクリックします。  
データベース内のすべてのトリガのプロファイリング情報が [プロファイル] タブに表示されます。

❖ **システム・トリガの要約プロファイリング情報を表示する**

- 1 左ウィンドウ枠で [システム・トリガ] フォルダを開きます。  
データベース内のすべてのトリガのリストが [システム・トリガ] タブに表示されます。

- 2 右ウィンドウ枠で [プロファイル] タブをクリックします。

データベース内のすべてのシステム・トリガのプロファイリング情報が [プロファイル] タブに表示されます。

### 特定のプロシージャのプロファイリング情報の表示

Sybase IQ は、個別のストアド・プロシージャ、関数、イベント、トリガのプロシージャ・プロファイリング情報を提供します。Sybase Central には、個別のプロシージャについての情報が表示されます。この情報は、すべてのストアド・プロシージャ、関数、イベント、またはトリガについて表示される情報とは異なります。

特定のプロシージャのプロファイリング情報を表示すると、次のカラムが表示されます。

- **呼び出し** オブジェクトが呼び出された回数をリストします。
- **ミリ秒** 各オブジェクトの総実行時間をリストします。
- **行** プロシージャの各行に行番号を付加します。
- **送信元** SQL プロシージャを 1 行ずつ表示します。

プロシージャは行単位に分割されるので、どの行の実行時間が長いかわかることができ、変更によってプロシージャのパフォーマンスを高められる可能性があります。プロシージャ・プロファイリング情報にアクセスするためには、データベースに接続すること、プロファイリングを有効にすること、DBA 権限を持っていることが必要です。

#### ❖ ストアド・プロシージャまたは関数のプロファイリング情報を表示する

- 1 左ウィンドウ枠でデータベースを展開します。
- 2 左ウィンドウ枠で [プロシージャと関数] フォルダを選択します。  
データベース内のすべてのストアド・プロシージャと関数のリストが右ウィンドウ枠の [プロシージャと関数] タブに表示されます。
- 3 左ウィンドウ枠で、プロファイル情報を取得するストアド・プロシージャまたは関数をクリックします。
- 4 右ウィンドウ枠で [プロファイル] タブをクリックします。

指定したストアド・プロシージャまたは関数のプロファイリング情報が右ウィンドウ枠の [プロファイル] タブに表示されます。

❖ **イベントのプロファイリング情報を表示する**

- 1 左ウィンドウ枠でデータベースを展開します。
- 2 左ウィンドウ枠で [イベント] フォルダを選択します。  
データベース内のすべてのイベントのリストが右ウィンドウ枠の [イベント] タブに表示されます。
- 3 左ウィンドウ枠で、プロファイル情報を取得するイベントをクリックします。
- 4 右ウィンドウ枠で [プロファイル] タブをクリックします。  
指定したイベントのプロファイリング情報が右ウィンドウ枠の [プロファイル] タブに表示されます。

❖ **トリガのプロファイリング情報を表示する**

- 1 左ウィンドウ枠でデータベースを展開します。
- 2 左ウィンドウ枠で [トリガ] フォルダを開きます。  
すべてのトリガのリストが右ウィンドウ枠の [トリガ] タブに表示されます。
- 3 右ウィンドウ枠で、プロファイル情報を取得するトリガを選択します。
- 4 右ウィンドウ枠で [プロファイル] タブをクリックします。  
指定したトリガのプロファイリング情報が右ウィンドウ枠の [プロファイル] タブに表示されます。

❖ **システム・トリガのプロファイリング情報を表示する**

- 1 左ウィンドウ枠でデータベースを展開します。
- 2 左ウィンドウ枠で [システム・トリガ] フォルダを開きます。  
すべてのシステム・トリガのリストが右ウィンドウ枠の [システム・トリガ] タブに表示されます。
- 3 右ウィンドウ枠で、プロファイル情報を取得するシステム・トリガを選択します。
- 4 右ウィンドウ枠で [プロファイル] タブをクリックします。  
指定したシステム・トリガのプロファイリング情報が右ウィンドウ枠の [プロファイル] タブに表示されます。

## Interactive SQL でのプロシージャ・プロファイリング情報の表示

ストアド・プロシージャを使用して、プロシージャ・プロファイリング情報を表示できます。Sybase Central でも Interactive SQL でも、表示されるプロファイリング情報は同じです。

**sa\_procedure\_profile\_summary** ストアド・プロシージャは、データベース内のすべてのプロシージャに関する情報を表示します。このプロシージャを使用すると、同じ結果セット内のストアド・プロシージャ、関数、イベント、システム・トリガ、トリガについてのプロファイリング・データを表示できます。次のパラメータはプロシージャが返すローを制限します。

- **p\_object\_name** プロファイル情報を取得するオブジェクトの名前を指定します。
- **p\_owner\_name** プロファイル情報を取得するオブジェクトの所有者を指定します。
- **p\_table\_name** トリガのプロファイル情報を取得するテーブルを指定します。
- **p\_object\_type** プロファイル情報を取得するオブジェクトのタイプを指定します。次の 5 つの選択肢があります。これらの値のいずれかを選択すると、結果セットは指定したタイプのオブジェクトだけに制限されます。
  - **P** ストアド・プロシージャ
  - **F** 関数
  - **T** トリガ
  - **E** イベント
  - **S** システム・トリガ
- **p\_ordering** 結果セットのソート順を指定します。

プロシージャは別のプロシージャを呼び出せるため、ユーザが明示的に呼び出したものより多くの項目がリストされる可能性があります。

以下の項では、DBA 権限を持つユーザとしてデータベースに接続しており、プロシージャ・プロファイリングが有効になっているものとします。

### ❖ すべてのプロシージャの要約プロファイリング情報を表示する

- 1 **sa\_procedure\_profile\_summary** ストアド・プロシージャを実行します。  
たとえば、次のように入力します。

```
CALL sa_procedure_profile_summary
```

2 [SQL] - [実行] を選択します。

データベース内のすべてのプロシージャの情報が含まれる結果セットが [結果] ウィンドウ枠に表示されます。

`sa_procedure_profile_summary` ストアド・プロシージャの詳細については、『SQL Anywhere Server — SQL リファレンス・マニュアル』を参照してください。

## Interactive SQL での特定のプロシージャのプロファイリング情報の表示

`sa_procedure_profile` ストアド・プロシージャは、特定のプロシージャ内の個々の行に関する情報を表示します。結果セットには、プロシージャ内の各行について、行番号、実行時間、合計実行時間に対する割合が含まれています。次のパラメータによって、プロシージャが返すローを制限できます。

- **p\_object\_name** プロファイル情報を取得するオブジェクトの名前を指定します。
- **p\_owner\_name** プロファイル情報を取得するオブジェクトの所有者を指定します。
- **p\_table\_name** トリガのプロファイル情報を取得するテーブルを指定します。

クエリでパラメータを指定しない場合、プロシージャは、呼び出されたすべてのプロシージャのプロファイリング情報を返します。

### ❖ プロシージャ内の特定の行のプロファイリング情報を表示する

1 `sa_procedure_profile` ストアド・プロシージャを実行します。

たとえば、次のように入力します。

```
CALL sa_procedure_profile
```

2 [SQL] - [実行] を選択します。

プロシージャの個々の行のプロファイリング情報が含まれる結果セットが [結果] ウィンドウ枠に表示されます。

`sa_procedure_profile` ストアド・プロシージャの詳細については、『SQL Anywhere Server — SQL リファレンス・マニュアル』を参照してください。



## バッファ・キャッシュのモニタリング

Sybase IQ には、バッファ・キャッシュのパフォーマンスをモニタリングするツールが用意されています。このモニタは、バッファ・キャッシュ、メモリ、Sybase IQ 内で行われた I/O 関数の統計情報を収集し、それらをログ・ファイルに保管します。

バッファ・キャッシュのパフォーマンスは、Sybase IQ 全体のパフォーマンスにとって重要な要因です。モニタが提供する情報を使用して、メイン・バッファ・キャッシュとテンポラリー・バッファ・キャッシュに割り付けるメモリの量を微調整できます。あるキャッシュが他のキャッシュよりもかなり多くの I/O を実行している場合、メモリを再割り付けするには、キャッシュ割り付けの 10 % ほどを少しずつ繰り返して行います。再割り付けが終了したら、負荷を再実行し、パフォーマンス上の変化をモニタリングします。

## バッファ・キャッシュ・モニタの起動

Sybase IQ バッファ・キャッシュ・モニタを Interactive SQL から起動します。モニタを起動するたびに、各モニタは Sybase IQ 内で別々のカーネル・スレッドとして実行されます。

次の構文を使って、モニタを起動します。

```
IQ UTILITIES { MAIN | PRIVATE }  
  INTO dummy_table_name  
  START MONITOR 'monitor_options [ ... ]'
```

MAIN を指定すると、接続先データベースの IQ ストア内のすべてのテーブルについて、メイン・バッファ・キャッシュのモニタリングを開始します。

PRIVATE を指定すると、接続先データベースのテンポラリー・ストア内のすべてのテーブルについて、テンポラリー・バッファ・キャッシュのモニタリングを開始します。

バッファ・キャッシュごとにコマンドを別々に発行する必要があります。モニタが結果を収集している間は、これらの各セッションを開いておく必要があります。接続を閉じると、モニタは実行を停止します。接続は最大で 2 つのモニタの実行まで開くことができます。1 つはメイン・バッファ・キャッシュ用で、もう 1 つはテンポラリー・バッファ・キャッシュ用です。

*dummy\_table\_name* には、任意の Sybase IQ ベース・テーブルまたはテンポラリ・テーブルを指定します。他の **IQ UTILITIES** コマンドと構文上の互換性を持たせるために、テーブル名を指定する必要があります。最も望ましいのは、モニタリング専用のテーブルを作成することです。

モニタリング出力ファイルのディレクトリ位置を制御するには、**MONITOR\_OUTPUT\_DIRECTORY** オプションを設定します。このオプションを設定しない場合は、データベースと同じディレクトリに結果が出力されます。モニタを実行している間、すべてのモニタリング出力ファイルが使用されます。モニタの実行が停止した後も、ファイルはそのまま残ります。

マルチプレックス・クエリ・サーバを作成する前に、モニタリングで使用するテンポラリ・テーブルを宣言するか、新しいデータベースの作成時に永続的なダミー・テーブルを作成してください。これによって DDL の変更を回避し、実際の運用稼働時にデータがクエリ・サーバにとどまるようにします。

---

### ヒント

モニタを簡単に使用するためには、ストアド・プロシージャを作成してダミー・テーブルを宣言し、出力ロケーションを指定して、モニタを起動します。

---

'*monitor\_options*' には次の値を 1 つ以上指定できます。

- **-summary** は、メインとテンポラリの両方のバッファ・キャッシュのサマリ情報を表示します。モニタ・オプションを何も指定しないと、サマリ・レポートが表示されます。他のオプションで説明されているフィールドに加えて、次のフィールドが表示されます。
  - **[Users]**: バッファ・キャッシュに接続しているユーザ数
  - **[IO]**: バッファ・キャッシュによる物理読み込みと物理書き込みの合計
- **-cache** は、メイン・バッファ・キャッシュまたはテンポラリ・バッファ・キャッシュに対するアクティビティの詳細を表示します。重要なフィールドは **[Finds]**、**[HR%]**、**[BWait]** です。次のフィールドが表示されます。
  - **[Finds]**: バッファ・キャッシュへの検索要求。Finds の値がゼロに急降下してゼロのままなら、サーバにデッドロックが発生しています。サーバでアクティビティがあれば、Finds はゼロ以外の値を示すはずです。
  - **[Creates]**: データベース内の 1 ページの作成要求

- [Dests] : データベース内の 1 ページの破棄要求
- [Dirty] : バッファがダーティ (変更) された回数
- [HR%] : ヒット率。I/O 要求なしで、バッファ・キャッシュによって応じることのできたパーセンテージ。ヒット率が高いほど効率が良くなります。キャッシュ・サイズを十分な大きさに設定した場合、通常は 90% ~ 100% になります。大きいクエリの場合、最初はヒット率が下がることがありますが、プリフェッチが機能し始めると上昇します。
- [BWait] : ビジー・ページ (ページ・フレーム競合) のために待機させられた検索要求。通常は小さい数値ですが、特別な場合には大きくなる場合があります。たとえば、まったく同じクエリが同時に開始された場合は、両方のクエリが同じページを要求するため、最初の要求がディスクからページを取得するまで 2 番目の要求は待機します。
- [ReReads] : 同一トランザクション内でストアの同じ部分がキャッシュ内に再読み込みされた概算の回数。この数値は常に小さいはずですが、Sybase IQ 12.4.2 以降では数値が大きくても重要ではありません。
- [FMiss] : 不正な失敗。バッファ・キャッシュがメモリ内のページの検索に複数回のルックアップを必要とした回数。この数値は 0 か、ごく小さな値になるようにしてください。この値が大きい場合は、ロールバックが発生し、特定の操作が繰り返し要求されていると考えられます
- [Cloned] : Sybase IQ が同時読み込み用に既存のバッファを保持しながら、書き込み用に新しいバッファを作成する必要があった回数。ページのクローンが作成されるのは、他のユーザがそのページを参照している場合だけです。
- [Reads/Writes] : バッファ・キャッシュによって実行された物理読み込みと物理書き込み
- [PF/PFRead] : プリフェッチ要求、およびプリフェッチ用に実行した読み込み。
- [GDirty] : LRU バッファがダーティな状態で取り込まれたため、Sybase IQ がそのバッファを使用する前に書き出した回数。この数値が 0 より大きい状態が長時間続かないようにしてください。0 より大きい状態が続く場合は、スリーパ・スレッドの数を増やすか、ウォッシュ・マーカを移動する必要があります。

- `[Pin%]`: バッファ・キャッシュ内で使用中でありロックされているページ数のパーセンテージ。
- `[Dirty%]`: 変更されたバッファ・ブロックのパーセンテージ。この数値が 85 ~ 90% を超えないようにします。それ以上になると、`[GDirty]` が 0 より大きくなります。
- `-cache_by_type` が生成する結果は `-cache` と同じですが、結果を IQ ページ・タイプごとに集計します (`[Bwaits]` カラムは例外で、合計だけを表示します)。この形式は、Sybase 製品の保守契約を結んでいるサポート・センタに情報を送る場合にたいへん有効です。
- `-file_suffix suffix` は、`<dbname>.<connid>-<main_or_temp>-<suffix>` の名前で、モニタリング出力ファイルを作成します。`suffix` を指定しないと、デフォルトで `iqmon` に設定されます。
- `-io` は、指定した期間のメインまたはテンポラリー (プライベート) のバッファ・キャッシュの I/O 率と圧縮比を表示します。これらのカウンタは、サーバのすべてのアクティビティを表します。情報はデバイス別に集計されません。次のフィールドが表示されます。
  - `[Reads]`: バッファ・キャッシュによって実行された物理読み込み
  - `[Lrd(KB)]`: 読み込まれた論理キロバイト数 (ページ・サイズに要求数を乗算した数値)
  - `[Prd(KB)]`: 読み込まれた物理キロバイト数
  - `[Rratio]`: 読み込まれた物理データに対する論理データの圧縮比。読み込み時のディスクに対する圧縮効率の度合い
  - `[Writes]`: バッファ・キャッシュによって実行された物理書き込み
  - `[Lwrt(KB)]`: 書き込まれた論理キロバイト数
  - `[Pwrt(KB)]`: 書き込まれた物理キロバイト数
  - `[Wratio]`: 書き込まれた物理データに対する論理データの圧縮比
- `-bufalloc` は、ソート、ハッシュ、ビットマップなどのオブジェクト用にバッファ・キャッシュ内の領域を予約する、メインまたはテンポラリー・バッファ・アロケータ情報を表示します。
  - `[OU]`: User\_Resource\_Reservation オプション設定 (以前は Optimize\_For\_This\_Many\_Users)
  - `[AU]`: 現在アクティブなユーザ数

- `[MaxBuf]`: バッファ・アロケータに制御されているバッファ数
- `[Avail]`: 現在ピン・クォータ割り付けに使用可能なバッファ数
- `[AvPF]`: 現在プリフェッチ・クォータ割り付けに使用可能なバッファ数
- `[Slots]`: バッファ・キャッシュ・クォータを使用中の、現在登録されているオブジェクト数
- `[PinUser]`: ピン・クォータを使用中のオブジェクト数 (ハッシュ、ソート、B ツリー・オブジェクトなど)
- `[PFUsr]`: プリフェッチ・クォータを使用中のオブジェクト数
- `[Posted]`: あらかじめプランされたクォータ・ユーザであるオブジェクト数
- `[UnPost]`: 特定のクォータ・ユーザであるオブジェクト数
- `[Locks]`: バッファ・アロケータで処理されたミューテックス・ロック数
- `[Waits]`: ロックのためにスレッドが待機する必要があった回数
- `-contention` は、多くの重要なバッファ・キャッシュとメモリ・マネージャ・ロックを表示します。これらのロック・カウンタとミューテックス・カウンタは、バッファ・キャッシュおよびヒープ・メモリ内のアクティビティと、これらのロックがどれだけ迅速に解消されたかを示します。タイムアウトの数値に注目してください。システム時間が 20% を超えている場合は、問題の発生を示しています。

---

**注意** オペレーティング・システムが進歩したため、Sybase IQ ではスピン・ロックを使用しなくなりました。このため、`[woTO]`、`[Loops]`、`[TOs]` の統計はめったに使用されません。

---

- `[AU]`: 現在アクティブなユーザ数
- `[LRULks]`: LRU がロックされた (テンポラリ・キャッシュ用に繰り返された) 回数
- `[woTO]`: タイムアウトなしにロックが付与された (テンポラリ・キャッシュ用に繰り返された) 回数
- `[Loops]`: ロックが付与される前に Sybase IQ がリトライした (テンポラリ・キャッシュ用に繰り返された) 回数

- *[TOs]* : Sybase IQ がタイムアウトして、ロックのために待機する必要があった (テンポラリ・キャッシュ用に繰り返された) 回数
- *[BWait]* : キャッシュ内のバッファに対する “Busy Waits” の (テンポラリ・キャッシュ用に繰り返された) 回数
- *[IOLock]* : Sybase IQ が圧縮化 I/O プールをロックした (テンポラリ・キャッシュ用に繰り返された) 回数。無視してかまわない
- *[IOWait]* : 圧縮化 I/O プール上のロックのために Sybase IQ が待機する必要があった (テンポラリ・キャッシュ用に繰り返された) 回数。無視してかまわない
- *[HTLock]* : Sybase IQ がブロック・マップ・ハッシュ・テーブルをロックした (テンポラリ・キャッシュ用に繰り返された) 回数
- *[HTWait]* : ブロック・マップ・ハッシュ・テーブルのために Sybase IQ が待機する必要があった (テンポラリ・キャッシュ用に繰り返された) 回数。HTLock と HTWait は、使用中のブロック・マップ数を示す
- *[FLLock]* : Sybase IQ がフリー・リストをロックする必要があった (テンポラリ・キャッシュ用に繰り返された) 回数
- *[FLWait]* : フリー・リスト上のロックのために Sybase IQ が待機する必要があった (テンポラリ・キャッシュ用に繰り返された) 回数
- *[MemLks]* : Sybase IQ がメモリ・マネージャ (ヒープ) をロックした回数
- *[MemWts]* : メモリ・マネージャ・ロックのために Sybase IQ が待機する必要があった回数
- *-threads* は、処理スレッド・マネージャが使用したカウンタを表示します。値はサーバワイドです (つまり、メインとプライベートのどちらでこのオプションを選択するかは無関係です)。レポートの最後のページ以降の新しいイベントを表します。
  - *[cpus]* : Sybase IQ が使用している CPU の数。システムに搭載されている数より少ない場合がある
  - *[Limit]* : Sybase IQ が使用できるスレッドの最大数
  - *[NTeams]* : 現在使用中のスレッド・チームの数

- [MaxTms] : 今まで使用されたチームの最大数
- [NThrds] : 既存スレッドの現在の数
- [Resrvd] : システム (接続) での使用のために予約されているスレッドの数
- [Free] : 割り当てに使用可能なスレッドの数。モニタリングが必要。この数値が非常に小さい場合は、スレッドの不足を示している。
- [Locks] : スレッド・マネージャで処理されたロックの数
- [Waits] : スレッド・マネージャ上のロックのために Sybase IQ が待機する必要があった回数

---

**注意** オブジェクトまたはクエリが作業を必要としている場合、Sybase IQ はスレッド・チームと呼ばれる処理スレッドのグループを割り付けます。処理スレッドの調整を行う際に、利用可能なオプションとしては、データベース・オプションの MAX\_IQ\_THREADS\_PER\_CONNECTION と MAX\_IQ\_THREADS\_PER\_TEAM、および Sybase IQ が使用できるスレッド数を指定するサーバ・オプションの -iqmt があります。

---

- -interval は、レポート間隔を秒単位で指定します。デフォルトは 60 秒ごとです。最小値は 2 秒ごとです。通常、クエリの実行中やパフォーマンスに問題があるときに、モニタをデフォルトの間隔で実行すると、有益な結果を取得できます。間隔が短すぎると、意味のある結果を取得できないことがあります。ジョブ時間に見合った間隔を指定してください。通常は 1 分で十分です。

最初の表示では、サーバの起動からのカウンタが示されます。それ以降の記録では、前の表示との差が示されます。

- -append | -truncate は、前者は既存の出力ファイルに追加、後者は既存の出力ファイルをトランケートします。デフォルトでは、トランケートされます。

- `-debug` は、主に、Sybase 製品の保守契約を結んでいるサポート・センタに情報を提供するために使用されます。これを指定すると、同じ情報を扱う標準表示モードがあるかどうかにかかわらず、パフォーマンス・モニタで使用可能な情報がすべて表示されます。ページの上部には、ディスク・ブロック・タイプごとの統計の配列が表示されます。次に、他のバッファ・キャッシュの統計、メモリ・マネージャの統計、スレッド・マネージャの統計、フリー・リストの統計、CPU 使用率、そして最後にバッファ・アロケータの統計が表示されます。バッファ・アロケータの統計はさらにクライアント・タイプ (ハッシュ、ソートなど) ごとに集計され、最後に行われたバッファ割り付けのヒストグラムが表示されます。メモリ割り付けは、レポートの最後のページ以降に割り付けられた量を示すことに注意してください。

---

**注意** 表示する間隔は、ページ単位ではなく、出力行単位です。ただし、次の 2 つの場合は例外です。`-cache_by_type` と `-debug` では、表示ごとに新しいページが開始されます。

---

## モニタ実行中の結果の確認

UNIX システムでは、クエリの実行中にモニタリング出力を確認できます。

たとえば、次のコマンドを使用してモニタを起動するとします。

```
iq utilities main into monitor_tab
start monitor "-cache -interval 2 -file_suffix iqmon"
```

このコマンドを実行すると、結果が `dbname.conn#-[main|temp]-iqmon` という名前の ASCII ファイルに出力されます。したがって、データベース `iqdemo` では、結果が `iqdemo.2-main-iqmon` に出力されます。

結果を確認するには、システム・プロンプトで次のコマンドを入力します。

```
$ tail -f iqdemo.2-main-iqmon
```



## バッファ・キャッシュ・モニタの停止

モニタの停止コマンドは起動コマンドとほぼ同じですが、オプションを指定する必要はありません。次の構文を使って Sybase IQ バッファ・キャッシュ・モニタを停止します。

```
IQ UTILITIES { MAIN | PRIVATE }  
INTO dummy_table_name STOP MONITOR
```

---

**注意** 特定のオプション設定を有効にするためには、データベースを再起動します。モニタを実行している場合は、データベースを再起動できるようにモニタをシャットダウンする必要があります。

---

## モニタリング結果の検査と保存

モニタは、結果を普通のテキスト・ファイルに保存します。このファイルのデフォルトは、次のとおりです。

- `dbname.connection#-main-iqmon` (メイン・バッファ・キャッシュの結果の場合)
- `dbname.connection#-temp-iqmon` (テンポラリ・バッファ・キャッシュの結果の場合)

プレフィクス `dbname.connection#` は、データベース名と接続番号を示します。接続番号が複数あって、どれが自分のものかわからない場合は、カタログ・ストアド・プロシージャ `sa_conn_info` を実行してください。このプロシージャを実行すると、アクティブなデータベース接続のそれぞれについて、接続番号、ユーザ ID などの情報が表示されます。

IQ UTILITIES コマンドで `-file_suffix` パラメータを使用すると、サフィックス `iqmon` を任意のサフィックスに変更できます。

モニタの実行結果を表示するには、テキスト・エディタを使用するか、ファイルの表示や印刷に通常使用している方法がほかにあれば、それを使用してください。

同じデータベースから同じ接続番号を使ってモニタを再度起動する場合、デフォルトでは前回の結果が上書きされます。モニタの実行結果を保存する場合は、ファイルを別の場所にコピーした後で同じデータベースからモニタを再度起動するか、`-append` オプションを使用してください。

モニタリング結果の例

この項では、いろいろなモニタリング・オプションを使用したサンプル結果を示します。

-summary オプションを使用すると、次のような結果が生成されます。  
IQ UTILITIES コマンドでの指定にかかわらず、メインとテンポラリ両方のバッファ・キャッシュの統計が表示されることに注意してください。

```
Sybase Adaptive Server IQ Performance Monitor

-----

Version 3.2

Options string for Main cache: "-summary -interval 5"

Summary
2004-07-16 13:53:24

Active|          Main Cache          |          Temp Cache
Users| Finds  HR% Reads/Writes GDirty Pin% Dirty% InUse%| Finds  HR%
Reads/Writes GDirty Pin% Dirty% InUse%

    0    286 99.3    2/34      0  0.0   1.6   26.2    608 99.7
2/47      0  0.0    3.6   20.0

    1   2621 99.4   16/155      0  5.6   8.7   81.7   4121 99.6
16/163      0 11.4   23.2   67.3

    1   2646 99.8    6/48      0  1.6  13.5  100.0   3388 99.8
6/70      1  4.1   40.9   94.5

    1   2684 99.9    7/78      0  5.6  14.3  100.0   3497 99.9
8/103      1 10.9   42.3   99.1

    1   1993 99.9   17/22      0  4.0  31.0  100.0   3342 98.7
122/149      0  8.2   41.4   91.4

    1   2479 99.9   32/110      0  5.6  13.5  100.0   3370 99.8
55/112      0 11.4   45.5   95.9

    1   3273 100.0    0/0      0  5.6  23.8  100.0   3951 100.0
0/108      1 13.6   49.1  100.0

    1   2512 99.9    2/0      0  1.6  31.0  100.0   3916 98.9
88/173      0  5.5   48.6  100.0

    1   1264 99.9   66/131      0  4.0  45.2  100.0   4317 98.9
378/305      0  6.4   40.0   77.3
```

```

1      2122 99.8      30/125      0  5.6  12.7  99.2      3122 99.7
67/127      0 12.3  40.0  90.5

1      3370 100.0      2/0      0  5.6  23.0  100.0      4034 100.0
2/98      2 13.2  46.4  98.2

1      2981 99.9      2/0      0  5.6  31.7  100.0      3715 99.9
2/110      0 14.1  53.2  100.0

1      3351 99.6      13/3      0  5.6  39.7  100.0      4131 99.7
13/123      0 14.1  57.7  100.0

1      3286 99.6      13/13      0  5.6  40.5  100.0      4135 99.6
15/139      0 12.3  55.9  97.7

1      296 100.0      0/0      0  1.6  41.3  100.0      3646 96.9
366/320      0  7.3  53.2  100.0

1      1230 99.4      71/129      0  6.3  58.7  100.0      4221 98.9
390/297      0  9.5  59.1  91.8

1      1900 100.0      125/279      0  4.0  50.0  100.0      4102 100.0
344/279      0  7.7  38.6  72.3

```

Sybase Adaptive Server IQ Performance Monitor

-----

Shutting Down

```

0      422 98.8      16/99      0  0.0  0.8  99.2      853 98.9
34/101      0  0.0  1.8  59.1

```

-cache オプションを使用すると、次のような結果が生成されます。  
これはテンポラリ・バッファ・キャッシュのものです。

Options string for Temp cache: "-cache -interval 10"

Temp Shared Buffer Cache

2001-02-18 17:43:55

Finds Creats Dests Dirty HR% BWaitis ReReads FMiss Cloned Reads/ PF/  
GDirty Pin% Dirty%

```

Tm:   640      82   57   84 99.4      0      4      0      0      Writes PFRead
0 0.0  2.8      4/0  0/0
Tm:  1139     109   83  109 100.0      0      0      0      0      0/0  0/0
0 0.0  5.5
Tm:  6794     754  749  754 100.0      0      0      0      0      0/0  0/0
0 0.0  6.1
Tm: 10759    1646 1646 1646 100.0      0      0      0      0      0/0  0/0
0 0.0  6.1

```

-io オプションを使用すると、次のような結果が生成されます。これはメイン・バッファ・キャッシュのものです。

Options string for main cache: "-IO -interval 5"

Main Buffer Cache  
2001-02-18 13:58:48

	Input					Output			
	Reads	Lrd(KB)	Prd(KB)	Rratio	Writes	Lwrt(KB)	Pwrt(KB)	Wratio	
Mn:	10	40	34	1.18	14	56	23	2.43	
Mn:	0	0	0	0.00	21	84	34	2.43	
Mn:	0	0	0	0.00	7	28	11	2.43	
Mn:	0	0	0	0.00	22	88	35	2.48	
Mn:	0	0	0	0.00	63	252	100	2.51	
Mn:	0	0	0	0.00	54	216	93	2.32	
Mn:	0	0	0	0.00	64	256	101	2.52	
Mn:	0	0	0	0.00	62	248	94	2.62	
Mn:	0	0	0	0.00	73	292	110	2.65	
Mn:	0	0	0	0.00	105	420	121	3.47	

-bufalloc オプションを使用すると、次のような結果が生成されます。

Options string for Main cache: "-bufalloc -file\_suffix bufalloc-iqmon -append -interval 10"

Buffer Allocation

2001-02-18 10:58:39

OU/AU	MaxBuf	Avail	AvPF	Slots	PinUsr	PFUsr	Posted	UnPost	Quota	Locks	Waits
1/0	1592	1592	20	0	0	0	0	0	0	1	0
1/1	1592	1592	20	0	0	0	0	0	0	1	0
1/1	1592	1592	20	0	0	0	0	0	0	1	0

---

**注意** 実際の -contention 出力では、メイン・キャッシュ、テンポラリ・キャッシュ、メモリ・マネージャの結果が同じ行に表示されます。このフォーマットでは横幅が大変広くなってしまうため、ここでは各カラム・セットを別々に示します。

---

メイン・キャッシュの -contention 結果は、次のとおりです。

Options string for Main cache:

"-contention -file\_suffix contention-iqmon -append -interval 10"

Contention  
2001-02-18 10:57:03

AU	Main Cache											
	LRULks	woTO	Loops	TOs	BWaits	IOLock	IOWait	HTLock	HTWait	FLLock	FLWait	
0	66	0	0	0	0	1	0	5	0	4	0	
1	2958	0	0	0	0	160	0	1117	0	6	0	
1	1513	0	0	0	1	378	0	2	0	8	0	
1	370	0	0	0	0	94	0	2	0	10	0	
1	156	0	0	0	0	46	0	2	0	12	0	
1	885	0	0	0	0	248	0	2	0	14	0	
1	1223	0	0	0	0	332	1	2	0	16	0	
1	346	0	0	0	0	66	0	2	0	18	0	

テンポラリ・キャッシュの -contention 結果は、次のとおりです。

LRULks	woTO	Loops	Temp Cache							
			TOS	BWaits	IOLock	IOWait	HTLock	HTWait	FLLock	FLWait
70	0	0	0	0	1	0	4	0	5	0
466	0	0	0	0	2	0	15	0	12	0
963	0	0	0	0	2	0	8	0	20	1
1186	0	0	0	0	2	0	2	0	23	1
357	0	0	0	0	2	0	2	0	25	1
444	0	0	0	0	2	0	3	0	29	0
884	0	0	0	0	2	0	2	0	31	1
1573	0	0	0	0	2	0	5	0	37	1

メモリ・マネージャの結果は、次のとおりです。

Memory Mgr	
MemLks	MemWts
55483	13
5705	0
2048	0
186	4
2	0
137	0
22	0
203	3

-threads オプションを使用すると、次のような結果が生成されます。

Options string for Main cache: "-threads -file\_suffix threads-iqmon -append -interval 10"

Threads

2001-02-18 10:59:24

CPUs	Limit	NTeams	MaxTms	NThrds	Resrvd	Free	Locks	Waits
10	100	4	12	100	13	68	106	590
10	100	6	12	100	12	63	4	6
10	100	6	12	100	12	63	0	0
10	100	7	12	100	12	62	1	1
10	100	7	12	100	12	62	0	0
10	100	7	12	100	12	58	1	5
10	100	7	12	100	12	58	0	0

## バッファ・キャッシュの構造

Sybase IQ では、システム上の CPU の数に応じて、バッファ・キャッシュのキャッシュ・パーティションの数が自動的に計算されます。マルチ CPU 構成でロードまたはクエリのパフォーマンスが予想より悪い場合は、`CACHE_PARTITIONS` データベース・オプションの値を変更するとパフォーマンスが向上することがあります。詳細については、『リファレンス：文とオプション』の「[CACHE\\_PARTITIONS オプション](#)」を参照してください。

バッファは、キャッシュの LRU (Least Recently Used) 側の終端に近づく、ウォッシュ・マーカを越えます。Sybase IQ は最も古いページ (ウォッシュ・マーカを越えたページ) をディスクに書き出して、そのページが占有していたキャッシュ領域を再利用できるようにします。スリーパ・スレッドと呼ばれる Sybase IQ 処理スレッドのチームが、最も古いバッファを一掃し (書き出し) ます。

データのページをキャッシュに読み込む必要がある場合、Sybase IQ は LRU バッファを取り込みます。バッファがまだ「ダーティな」(変更された) 状態の場合は、先にバッファをディスクに書き込む必要があります。モニタ `-cache` レポートの `[Gdirty]` カラムは、LRU バッファをダーティな状態で取り込んだために、Sybase IQ がそのバッファを使用する前に書き出す必要があった回数を示します。

通常、Sybase IQ では [Gdirty] の値が 0 に維持されます。この値が 0 より大きい状態が長時間続く場合は、スイーパー・スレッドの数とウォッシュ・マーカを制御するデータベース・オプションを調整する必要があります。詳細については、『リファレンス：文とオプション』の「[SWEEPER\\_THREADS\\_PERCENT オプション](#)」または「[WASH\\_AREA\\_BUFFERS\\_PERCENT オプション](#)」を参照してください。

## バッファ・マネージャのスラッシングの回避

システムが要求されたページを読み込む前にダーティ・ページを書き込む必要がある場合、スラッシングが発生し、システムの速度が大幅に低下します。最適なパフォーマンスを得るためには、ページ・ライタが空き領域の需要に対応できるように常に十分な空きメモリを割り付けます。

バッファ・キャッシュ・スラッシングはシステム・スラッシングに似ていて、読み込みに使用できるクリーン・バッファが十分でない場合に発生します。このスラッシングは、キャッシュで「書き込んでから読み込む」場合と同様の遅延を生じ、バッファ・キャッシュが不十分でクエリで参照されたすべてのオブジェクトに対応できない場合に発生します。

バッファ・キャッシュ・スラッシングを排除するには、バッファ・キャッシュに十分なメモリを割り付ける必要があります。バッファ・キャッシュへの過度の割り付けには注意してください。データベース・バッファ・キャッシュにメモリを過度に割り付けると、システム・スラッシングを誘発します。極端な場合、メモリの過度の割り付けはバッファ・キャッシュ・スラッシング問題を解決しないまま、複数のレベルのスラッシングを引き起こす可能性があります。

マルチユーザ・コンテキストにおいて、または、クエリに使用可能なキャッシュに適した数より非常に多くの値で HASH オブジェクトを構築する必要のある環境で複雑なクエリに起因するスキューや不確実性が原因でオプティマイザが HASH アルゴリズムを選択する場合に、より不明確なバッファ・キャッシュ・スラッシングが発生する可能性もあります。

バッファ・サイズを設定するときは、次のトレードオフに注意してください。

- Sybase IQ バッファ・キャッシュが大きすぎると、Sybase IQ が全メモリを使用しようとするため、オペレーティング・システムでページングが強制的に行われる。
- Sybase IQ バッファ・キャッシュが小さすぎると、Sybase IQ はクエリ・データをキャッシュに収めきれないため、スラッシングしてしまう。

深刻なパフォーマンスの問題が発生した場合は、ページングをモニタリングして、スラッシングが問題かどうかを確認してください。スラッシングが問題だった場合は、「[バッファ・キャッシュの管理](#)」の説明に従って、バッファ・サイズをリセットしてください。

ページングをモニタリングして、スラッシングが問題と判断した場合は、ハッシュ・アルゴリズムを伴うクエリが含まれる文の実行時のスラッシングの量を制限することもできます。

`HASH_THRASHING_PERCENT` データベース・オプションを調整し、許容するハード・ディスク I/O の割合を制御します。この割合を超えると、文がロールバックされてエラーが返されます。

`HASH_THRASHING_PERCENT` のデフォルト値は 10% です。

`HASH_THRASHING_PERCENT` の値を増やすと、ロールバックが起きるまでに許容するディスクへのページングが増え、

`HASH_THRASHING_PERCENT` の値を減らすと、ロールバックが起きるまでに許容するページングが減ります。

以前のバージョンの Sybase IQ では実行されていた、ハッシュ・アルゴリズムを伴うクエリが、デフォルトの `HASH_THRASHING_PERCENT` の制限に達するとロールバックされるようになります。Sybase IQ は Hash insert thrashing detected または Hash find thrashing detected エラーをレポートします。実行に必要なリソースをクエリに割り当てるには、次の 1 つ以上の対応策を講じてください。

- `HASH_THRASHING_PERCENT` の値を増やし、ページングの制限を緩和する。
- テンポラリ・キャッシュのサイズを増やす (DBA のみ)。システム・スラッシングが生じる可能性を回避するため、テンポラリ・キャッシュのサイズを増やすと、メイン・キャッシュ割り付けで同じサイズが減ることに注意してください。
- Sybase IQ がこの文の 1 つ以上のハッシュ・サイズの見積もりを誤っている原因を突き止めて改善する。たとえば、LF または HG インデックスを必要とするすべてのカラムにそれがいるかどうかを確認します。また、複数カラムのインデックスが適切かどうかにも検討します。



- データベース・オプション `HASH_PINNABLE_CACHE_PERCENT` の値を減らします。

これらのデータベース・オプションの詳細については、『リファレンス: 文とオプション』の「[HASH\\_THRASHING\\_PERCENT オプション](#)」と「[HASH\\_PINNABLE\\_CACHE\\_PERCENT オプション](#)」の項を参照してください。

クエリで起きている可能性のある問題を特定するには、テンポラリ・データベース・オプション `QUERY_PLAN = 'ON'` と `QUERY_DETAIL = 'ON'` を指定してクエリを実行し、クエリ・プランを生成します。次に、クエリ・プランの見積もりを調査します。生成されたクエリ・プランはメッセージ・ログ・ファイルにあります。

## Windows システムでのページングのモニタリング

Windows のシステム モニタを使用して、ページングをモニタリングできます。アクセスするには、[論理ディスク] オブジェクト、`PAGEFILE.SYS` ファイルが格納されているディスクのインスタンス、[ディスク転送/秒] カウンタを選択します。Windows ページ・ファイルはデータベース DB 領域デバイスとは別のディスクに格納してください。[オブジェクト・メモリ] と [ページ/秒] カウンタもモニタリングできます。ただし、この値はソフト・フォールトとハード・フォールトの両方を含む全メモリ・フォールトの合計となります。

## UNIX システムでのページングのモニタリング

UNIX のシステム・コマンド `vmstat` を使用して、ページングなどのシステム・アクティビティをモニタリングできます。コマンドの省略形構文は次のとおりです。

**`vmstat interval count`**

*interval* には次の行を出力するまでの時間間隔を、*count* には出力行の表示回数を指定します。オプションとフィールドの説明を含めた `vmstat` の詳細については、使用しているオペレーティング・システムのマニュアルを参照してください。たとえば次のようになります。

```
> vmstat 2 3
procs      memory                page              disk      faults      cpu
r b w swap    free re mf pi po fr de sr s0 s1 sd in  sy  cs us sy id
```

```
0 0 0 3312376 31840 0 8 0 0 0 0 0 0 0 0 297 201 472 82 4 14
0 0 0 3312376 31484 2 3 0 0 0 0 0 0 0 0 260 169 597 80 3 17
0 0 0 3312368 31116 0 8 0 0 0 0 0 0 0 0 205 1202 396 67 4 29
```

上記の出力では、マシンの物理メモリが過度に割り付けられていないため、**Sybase IQ** のクエリ状態が安定していることが示されています。システムのページ・フォールトはほとんど発生していません。次の例では、問題があることが **vmstat** 出力に示されています (見やすくなるように、一部のフィールドを省略しています)。

procs				memory		page								faults		cpu			
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	in	sy	cs	us	sy	id		
0	0	0	217348	272784	0	148	11	3	9	0	2	251	1835	601	6	3	91		
0	0	0	3487124	205572	0	5	0	0	0	0	0	86	131	133	0	1	99		
0	0	0	3487124	205572	0	5	0	0	0	0	0	71	162	121	0	0	100		
0	0	0	3483912	204500	0	425	36	0	0	0	0	169	642	355	2	2	96		
0	0	0	3482740	203372	0	17	6	0	0	0	0	158	370	210	1	3	97		
0	0	0	3482676	203300	0	4	10	0	0	0	0	160	1344	225	1	2	97		
0	0	0	3343272	199964	1	2123	36	0	0	0	0	213	131	399	7	8	85		
0	0	0	3343264	185096	0	194	84	0	0	0	0	283	796	732	1	6	93		
0	0	0	3342988	183972	0	17	58	0	0	0	0	276	1051	746	2	4	94		
0	0	0	3342860	183632	0	119	314	0	0	0	0	203	1660	529	3	4	94		
0	0	0	3342748	182316	2	109	184	0	0	0	0	187	620	488	4	2	95		
0	0	0	3342312	181104	2	147	96	0	0	0	0	115	256	260	9	2	89		
0	0	0	3340748	179180	0	899	26	0	0	0	0	163	836	531	4	4	92		
0	0	0	3328704	167224	0	2993	6	0	0	0	0	82	2195	222	4	7	89		

上記の出力の最初の行には、マシンが起動されてからのシステム・アクティビティの概要が示されています。最初の 3 行は、約 200MB の空き物理メモリがあり、マシンがアイドル状態であることを示しています。4 行目は、**Sybase IQ** が最初に起動されたときと対応しています。8 行目から、空きメモリの量が急速に減少し始めています。これは、**Sybase IQ** バッファ・キャッシュが割り付けられ、データベース・ページがディスクから読み込まれているときと対応しています (CPU 使用率が増加していることに注意してください)。この時点ではクエリが開始されていないため、ユーザの CPU 時間がほとんどありません。

procs				memory		page								faults		cpu			
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	in	sy	cs	us	sy	id		
7	0	0	3247636	58920	0	1880	1664	0	0	0	0	1131	442	1668	80	18	3		
18	0	0	3246568	43732	0	709	1696	0	0	0	0	1084	223	1308	90	10	1		
12	0	0	3246604	37004	0	358	656	0	0	0	0	600	236	722	95	5	0		

```
15 0 0 3246628 32156 0 356 1606 0 0 0 0 1141 226 1317 91 9 0
19 0 0 3246612 26748 0 273 1248 0 0 0 0 950 394 1180 92 7 0
```

上記の出力は少し後の、クエリが進行中のときのものです。これは、ユーザ・モードの CPU レベル (*us* フィールド) を見ると明らかです。依然としてページイン・フォールト (ページインされた KB を示す、*pi* フィールド) が発生しており、空きメモリの量が減少しているため、バッファ・キャッシュはまだ満杯になっていません。

procs			memory		page					faults				cpu			
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	in	sy	cs	us	sy	id
21	0	0	3246608	22100	0	201	1600	0	0	0	1208	1257	1413	88	12	0	
18	0	0	3246608	17196	0	370	1520	0	464	0	139	988	209	1155	91	8	0
11	0	0	3251116	16664	0	483	2064	138	2408	0	760	1315	218	1488	88	12	0
30	0	0	3251112	15764	0	475	2480	310	4450	0	1432	1498	199	1717	87	13	0

上記の出力は、さらに後のものです。出力の 3 行目は、システムが管理できる空きメモリ量のスレッシュホールドに達したことを示しています。この時点でページアウト (ページアウトされた KB を示す、*po* フィールド) が発生し、それに応じてシステム・モードの CPU レベル (*sy* フィールド) が増加しています。この状況が発生するのは、物理メモリが過度に割り付けられたためです。つまり、Sybase IQ バッファ・キャッシュのサイズが、このマシンには大きすぎるのです。この問題を解決するには、メインとテンポラリのバッファ・キャッシュのどちらか、または両方のサイズを減らします。

## バッファ・キャッシュ・モニタリング・チェックリスト

次の表は、モニタリング結果で着目すべき最も一般的な項目の要約と、動作が正常な範囲を逸脱している場合に講じる必要がある対応策を示します。[統計] カラムには、標準のモニタ・レポートに表示される名前が示されています。デバッグ・レポートでは別の名前が表示される統計については、その名前も示されています。

どのモニタリング統計でも、新しいクエリが開始されたときなど、システムの状態が変化している間は一時的な異常が起きる可能性があることに注意してください。

表 5-11 : バッファ・キャッシュ・モニタリング・チェックリスト

統計	正常な動作	調整が必要な動作	推奨される対応策
HR% (Cache hit rate)	90% 以上。 GARRAY、BARRAY、Bitmap (bm)、hash object、sort object、variable-length btree (btrees)、fixed-length btree (btrees)、bit vector (bv)、dbext、dbid、vdo、store、checkpoint block (ckpt) などの個別の内部データ構造体の場合、クエリの実行中はヒット率が 90% を上回る。最初は 90% を下回る場合がある。プリフェッチが機能し始めると (PF または PrefetchReqs > 0)、ヒット率が徐々に上昇して 90% を超える。	プリフェッチが機能した後もヒット率が 90% を下回る。  <b>注意</b> オブジェクトによってはプリフェッチを行わないものがあるため、これらのヒット率は一般に低くなる。	-iqmc と -iqtc を調整し、メインとテンポラリのキャッシュ・サイズのバランスをとり直してみる。  PREFETCH_THREADS_PERCENT オプションを調整し、プリフェッチ・スレッド数を増やしてみる。
Gdirty (Grabbed Dirty)	適度なキャッシュ・サイズ (< 10GB) が設定されたシステムでは 0。	GDirty > 0  <b>注意</b> スイープ・スレッドがアクティブになるのは、ダーティ・ページの数がおウッシュ・エリアの一定の割合に達した場合だけである。 GDirty/GrabbedDirty が 0 より大きく、I/O 率 (Writes) が低い場合は、単にシステムに軽い負荷がかかっていると考えられるため対応策は不要。	SWEEPER_THREADS_PERCENT オプション (デフォルトは 10%) または WASH_AREA_BUFFERS_PERCENT オプション (デフォルトは 20%) を調整し、おウッシュ・エリアのサイズを増やす。

統計	正常な動作	調整が必要な動作	推奨される対応策
BWaits (Buffer Busy Waits)	0	> 0 の状態が持続し、複数のジョブが同じバッファで衝突していることを示している。	I/O 率 (Writes) が高い場合は、キャッシュのスラッシングが原因で Busy Waits が起きていると考えられる。 キャッシュ・レポートでヒット率を調べて、メインとテンポラリのキャッシュのバランスをとり直す必要があるかどうかを確認する。 ほぼ同一の多数のクエリを同時にバッチ・ジョブで開始している場合は、開始時刻をずらしてみる。
LRU Waits (デバッグ・レポートでは LRU Num TimeOuts percentage)	20% 以下	> 20%。これは重大な競合問題が起きていることを示す。	オペレーティング・システムのパッチ・レベルやその他の環境設定を確認する。これはオペレーティング・システムの問題の可能性が高い。
IOWait (IOWait)	10% 以下	> 10%	ディスク・エラーや I/O リトライを調べる。
FLWait (FLMutex Waits)	20% 以下	> 20%	DB 領域の設定を確認する。 データベース領域が不足しかかっていないか？ DISK_STRIPING が ON になっているか？ sp_iqcheckdb が 15% を超える断片化をレポートしていないか？

統計	正常な動作	調整が必要な動作	推奨される対応策
HTWait (BmapHTNum Waits)  MemWts (MemNtimes Waited)  (PFMgrCond VarWaits)	10% 以下	> 10%	Sybase 製品の保守契約 を結んでいるサポー ト・センタに問い合わ せる。
CPU time (デバッグ・ レポートでは CPU Sys Seconds、CPU Total Seconds)	CPU Sys Seconds < 20%	CPU Sys Seconds > 20%  CPU Total Seconds も低い使用 率を示しており、システムが ビジー状態になるだけのジョ ブがある場合は、キャッシュ がスラッシングしているか、 並列度が失われていると考え られる。	-iqgovern を調整し、実 行できる同時クエリの 合計数を減らす。  キャッシュ・レポート でヒット率と I/O 率を 調べて、キャッシュが スラッシングしてい ないかどうかを確認 する。また、 <b>cache_by_type</b> (または デバッグ) レポートで ハッシュ・オブジェク トのヒット率を調べ て、ハッシュ・オブジェ クトがスラッシングし ていないかどうかを確認 する：ヒット率が <90% で I/O 率 (Writes) が高くないか？  試行された並列処理を クエリ・プランで確認 する。十分なスレッド が使用可能だったか？  システムに大量の CPU が搭載されているか？ マルチプレックス構成 などの対策が必要な場 合もある。

統計	正常な動作	調整が必要な動作	推奨される対応策
InUse% (Buffers in use)	起動時以外は 100% かそれに近い値	100% 未満	バッファ・キャッシュが大きすぎる可能性がある。  -iqmc と -iqtc を調整し、メインとテンポラリのキャッシュ・サイズのバランスをとり直してみる。
Pin% (Pinned buffers)	< 90%	> 90 ~ 95%。システムがバッファ不足状態に危険なほど近づいていることを示す。この状態になるとトランザクションがロールバックされる。	メインとテンポラリのキャッシュ・サイズのバランスをとり直してみる。  バッファ・キャッシュ・サイズのバランスをとり直すことができない場合は、-iqgovern を減らして、同時に実行されるジョブの数を制限してみる。
Free threads (ThrNumFree)	空き > 予約済み	空きスレッドの数が予約済みの数まで減少している場合は、システムのスレッドが不足していると考えられる。	次のいずれかを試してみる。  -iqmt を設定してスレッドの数を増やす。  スレッド関連の次のオプションの値を減らす： MAX_IQ_THREADS_PER_CONNECTION、 MAX_IQ_THREADS_PER_TEAM。 USER_RESOURCE_RESERVATION を設定し、クエリ・エンジンのリソース割り付けを制限する。  -iqgovern を設定し、ジョブの数を制限する。
FlOutOfSpace (デバッグのみ)	0。このストアのフリー・リストが満杯でないことを示す。割り付けられていないページが使用可能	1。このストア (メインまたはテンポラリ) が全部割り付けられていることを示す。	ストアに DB 領域を追加する。

## CPU 使用率をモニタリングするシステム・ユーティリティ

Sybase Central のパフォーマンス・モニタのほか、オペレーティング・システム・ユーティリティを使用して、Sybase IQ を使用中に CPU 使用率をモニタリングできます。

OS	ユーティリティ	説明
UNIX	top (Sun、Linux、HP-UX) topas (IBM-AIX)	プロセッサ・アクティビティを継続してリアル・タイムで参照します。
	ps	プロセス・ステータスをレポートします。
	vmstat	システム・プロセス、メモリ、ページング、ブロック IQ、トラップ、および CPU アクティビティに関する情報を表示します。
	iostat -x	ディスクのサブシステム情報を表示します。
Windows	システム・モニタタスク・マネージャ	コンピュータのパフォーマンス、実行しているアプリケーション、プロセス、CPU 使用率、および他のシステム・サービスに関する詳細情報を提供します。



## 32 ビット Windows システムでの サーバのチューニング

### この章について

この章では、Windows システムで Sybase IQ を実行するときに特有のパフォーマンスとチューニングのガイドラインについて説明します。この章の説明は、「[第 4 章 システム・リソースの管理](#)」の説明と併せてお読みください。

### 内容

トピック名	ページ
<a href="#">パフォーマンスについての一般的なガイドライン</a>	148
<a href="#">パフォーマンスのモニタリング</a>	150
<a href="#">NTFS キャッシュの使用</a>	152
<a href="#">挿入とクエリのチューニング</a>	152
<a href="#">バックアップ操作のチューニング</a>	154

## パフォーマンスについての一般的なガイドライン

この項では、データのロードとクエリの両方に適用される一般的なガイドラインについて説明します。Windows で Sybase IQ を実行するときに推奨される最小限のメモリ量 (RAM) は 512MB です。最高のパフォーマンスを得るには、大部分のアプリケーションで 4GB をおすすめします。boot.ini で /3GB スイッチと /PAE スイッチを設定し、Sybase IQ が Windows 32 ビット・システム上でできるだけ多くのメモリを使用できるようにしてください。

## スループットの最大化

Windows で実行する場合は、ネットワーク・サービスのサーバのオプションで [ネットワーク アプリケーションのスループットを最大にする] が有効になっていることを確認してください。

### ❖ ネットワーク アプリケーションのスループットを最大にする

- 1 コントロールパネルの [ネットワーク接続] をダブルクリックし、[ローカル エリア接続] を右クリックし、[プロパティ] を選択します。
- 2 [Microsoft ネットワーク用ファイルとプリンタ共有] を選択し、[プロパティ] をクリックします。
- 3 [最適化] から、[ネットワーク アプリケーションのデータ スループットを最大にする] を選択します。

---

**注意** Windows の一部のバージョンでは、ネットワーク・アプリケーションのデータ・スループットを最大にするために Microsoft インターネット インフォメーションサービス (IIS) をインストールしてサーバ・プロパティを設定する必要があります。

---

## メモリの割り付け超過の防止

マシンの物理メモリ (RAM) を過度に割り付けると、システムのページ・フォールトが頻繁に発生します。ページ・フォールトが頻繁に発生すると、Sybase IQ のパフォーマンスが著しく低下します。Sybase IQ のバッファを慎重に割り付け、Sybase IQ プロセスの仮想アドレス空間と使用可能な物理メモリをモニタリングすることにより、メモリの割り付けの超過を防ぐことができます。この項では、Sybase IQ によるマシンの物理メモリの使用状況をモニタリングするためのガイドラインについて説明します。

## 物理メモリのモニタリング

アプリケーション (Sybase IQ) が使用できる物理メモリ量は、[物理メモリ (KB)] の下に表示されます。[利用可能] の値が常に 5000 未満である場合は、マシンの物理メモリが過度に割り付けられている可能性があります。これは、値が 5000(KB) 台になると、最低限 5MB の空きメモリを維持するために Windows がページ・フォールトを発生させるからです。

物理メモリをモニタリングするには、[タスク マネージャ] アプレットの [パフォーマンス] タブをクリックします。

## ファイル・システム

Windows ファイル・システムでは、ファイル、ディレクトリ、ボリューム・レベルでの圧縮がサポートされています。Sybase IQ データベースを格納するすべてのディスクとボリュームで圧縮オプションをオンにし、Windows ファイル・システムの圧縮を無効にします。これは、Sybase IQ が組み込みの圧縮を提供しているためです。ファイル・システムの圧縮を使用しても、それ以上データベースのサイズを縮小することはできず、読み込みや書き込みを実行するときに CPU のオーバーヘッドが増えるだけです。

## パフォーマンスのモニタリング

Sybase IQ のパフォーマンスをモニタリングするための主なツールは、Sybase IQ パフォーマンス・モニタです。このツールについては、「[第5章 パフォーマンスのモニタリングとチューニング](#)」に記載されています。さらに、Windows タスク マネージャと Windows パフォーマンス ツールを使用してもシステム・パフォーマンスをモニタリングできます。

### タスク マネージャ

Windows タスク マネージャは、コンピュータ上で実行されているアプリケーションとプロセスに関する情報を提供します。マシンのパフォーマンスとリソースの詳細も、タスク マネージャに表示されます。

- タスク マネージャを開くには、タスクバーの空のスペースを右クリックし、[タスク マネージャ]を選択します。
- Windows タスク マネージャの使用法の詳細については、Windows タスク マネージャで[ヘルプ]をクリックし、タスク マネージャのヘルプ・トピックを選択します。

### パフォーマンス

Windows パフォーマンス ツールには、システム モニタおよびパフォーマンス ログと警告があります。システム モニタには、メモリ、ディスク、プロセッサ、およびネットワーク活動のリアルタイムのデータが表示されます。パフォーマンス ログと警告では、パフォーマンス・データを記録したり、特定のパフォーマンス活動の警告を設定したりすることができます。

- Windows パフォーマンス ツールを開くには、コントロール パネルの[管理ツール]をクリックして[パフォーマンス]を選択します。
- パフォーマンス ツールの使用法の詳細については、[ヘルプ]をクリックしてパフォーマンスのヘルプ・トピックを選択します。

## 仮想アドレス空間とワーキング・セットのモニタリング

プロセスの仮想アドレス空間は、プロセスの合計サイズです。プロセスのワーキング・セットは、現在プロセスに割り付けられている物理メモリ量です。ほとんどの環境では、過度なシステムのページ・フォールト避けるために、Sybase IQ プロセスの仮想アドレス空間をマシンの物理メモリより小さくしてください。

Sybase IQ サーバ内での仮想メモリの使用パターンが原因で、Windows プラットフォーム上で仮想メモリの断片化によって処理が過度に増大する可能性があります。このような状況に陥る可能性を小さくするため、Sybase IQ では Windows XP と Windows Server 2003 について Microsoft の LFH (低断片化ヒープ) の使用をサポートしています。

#### ❖ 仮想アドレス空間とメモリ使用状況のモニタリング

- 1 タスクバーの空のスペースを右クリックし、[タスク マネージャ] を選択し、[プロセス] タブをクリックします。
- 2 [表示] をクリックし、[列の選択] を選択します。
- 3 [列の選択] ダイアログで、次のカラムを選択します。
  - [メモリ使用量]
  - [メモリ使用量デルタ]
  - [最大メモリ使用量]
  - [ページフォルト]
  - [ページフォルト デルタ]
  - [仮想メモリ サイズ]

## ページ・フォールトのモニタリング

上記と同じように、Windows パフォーマンス モニタで [Sybase IQ] プロセスを選択します。[ページ・フォールト/秒] カウンタを選択します。このカウンタには、「ソフト」ページ・フォールトと「ハード」ページ・フォールトの両方が含まれます。ハード・ページ・フォールトは、ディスク I/O を引き起こすページ・フォールトです。ソフト・ページ・フォールトは一般に、パフォーマンスの問題には関係しません。

ハード・ページ・フォールトの数値を調べるには、[論理ディスク] オブジェクト、および *pagefile.sys* ファイルが格納されている場所 (このファイルは Sybase IQ データベースとは別のボリュームに格納してください) のインスタンスを選択します。[ディスク転送/秒] カウンタを選択します。この値を [ページ・フォールト/秒] の値と比較すると、ページ・フォールトに対するハード・ページ・フォールトの割合がわかります。ページ・ファイルへの I/O アクティビティはほとんどないことが理想です。ただし、メモリが少ない構成では、ページングが発生することが多くなります。

ハード・ページ・フォールト率が毎秒 20 を超える状態が持続する場合は、マシンの物理メモリが過度に割り付けられています。

## NTFS キャッシュの使用

ネットワーク・サービスのサーバ・オプション「[ネットワーク アプリケーションのスループットを最大にする](#)」(148 ページ) を有効にし、NTFS およびその関連キャッシュを使用すると、挿入とクエリの両方で Sybase IQ データベースのパフォーマンスが向上します。これは、物理メモリが同じ量でも NTFS は Sybase IQ バッファ・キャッシュよりも非常に多くのデータを格納できるからです。この利点は、メイン・キャッシュ・サイズを小さくし、メモリをより有効に使用できる NTFS に割り当てることで最も効果的に活かされます。IQ 15.x では、ロードおよびクエリ処理におけるテンポラリ・キャッシュで使用されます。結果として、NTFS キャッシュの使用はメイン・キャッシュには有効な手段ですが、テンポラリ・キャッシュでは同等の効果が得られません。

Sybase IQ バッファ・キャッシュには、Sybase IQ データ (ページ) が非圧縮形式で格納されます。つまり、100MB の Sybase IQ バッファ・キャッシュには、100MB 分のデータを格納できます。これに対し、NTFS キャッシュでは Sybase IQ データを圧縮形式で管理します。このため、圧縮率が 2:1 の場合、100MB の NTFS キャッシュには 200MB の Sybase IQ データが格納されている可能性があります。したがって、NTFS キャッシュでは、キャッシュのヒット率をより高く維持して、I/O を減らすことが可能です。たとえば NTFS キャッシュから Sybase IQ バッファ・キャッシュに移動するデータを圧縮解除するために計算のオーバーヘッドが生じて、I/O の削減効果の方が重要です。

## 挿入とクエリのチューニング

この項では、Windows プラットフォームでの挿入とクエリをチューニングするための追加のガイドラインについて説明します。

## 適切にチューニングされた挿入オペレーションの特性

適切にチューニングされた Sybase IQ の挿入オペレーションには、ある特性が見られます。これらの特性は、Windows タスク マネージャと Windows パフォーマンス モニタで確認できます。

- 挿入オペレーションの大部分は、CPU の能力に依存します。システム内のすべての CPU は、能力の 100% 近くを使用して実行されます。CPU の 95% 以上がユーザ・モードで実行されます。このことは、Windows タスク マネージャの [パフォーマンス] タブをクリックし、[表示]-[カーネル時間を表示する] オプションを設定すると簡単に確認できます。
- 物理メモリを過度に割り付けないように注意してください。特に、Sybase IQ プロセスの仮想アドレス空間は、マシンの物理メモリ (RAM) より小さくする必要があります。
- ハード・ページ・フォールト (*pagefile.sys* が格納されているボリュームへの I/O) を少なくし、理想的には 0 (ゼロ) に近づけてください。
- IQ ストアへの I/O 操作が安定して実行され、ディスク・サブシステムの I/O 処理能力を超えないようにします。

Sybase IQ では、ファイルを順次アクセス用に読み込むことを指定する Windows CreateFile オプションを (ファイルの作成時および開くときの両方に) 使用します。LOAD TABLE コマンドで指定したファイルには、このオプションが使用されます。これにより、先読みが行われ、NTFS キャッシュ・メモリの使用量が減少するため、ロードのパフォーマンスが向上します。

ロードのパフォーマンスは、さらに向上する可能性があります。場合によっては大幅に向上します。それには、Sybase IQ メイン・バッファ・キャッシュのサイズを「[Sybase IQ のメイン・バッファ・キャッシュとテンポラリ・バッファ・キャッシュ](#)」(62 ページ) で推奨されている計算上の値よりもかなり小さく設定します。これでパフォーマンスが向上する理由は、「[NTFS キャッシュの使用](#)」(152 ページ) に記載されています。Sybase IQ メイン・バッファ・キャッシュを、計算上の推奨値より最大 50% 小さく設定できます。

## クエリのチューニング

クエリのパフォーマンスも向上する可能性があります。それには、上述のように、メイン・バッファ・キャッシュのサイズを小さく設定します。クエリ・プラン、構造、オプションの詳細については、「[第3章 クエリと削除の最適化](#)」を参照してください。

## バックアップ操作のチューニング

Windows では、固定長の I/O デバイスのみがサポートされています。このため、テープの読み込みや書き込みは、その前後の読み込みや書き込みと同じサイズで行う必要があります。読み込みや書き込みの操作がハードウェア・デバイスの容量を超えた場合、操作は失敗します。したがって、バックアップやリストアの操作では、ハードウェアが設定されているサイズですべての書き込み(または読み込み)を行わないと、バックアップ(またはリストア)が失敗します。

Sybase IQ では、どのプラットフォームでもできるだけ効率的に読み込みと書き込みの操作が行われるようにデフォルトが設定されています。ただし、Sybase IQ データベースの作成時にデフォルトのブロック・サイズを上書きしている場合は、そのデータベースをバックアップするときにブロック係数を調整する必要があります。

バックアップまたはリストアでは、次の式が適用されます。

$$\text{ブロック・サイズ} \times \text{ブロック係数} = \text{I/O サイズ}$$

Windows システムでブロック係数を調整するには、使用中のテープ・デバイスで処理できる物理ブロックの最大サイズの情報が必要です。通常、ドライブの製造元のマニュアルには、この情報は記載されていません。値(通常は 64KB)を調べるには、WIN32 API 呼び出しを使って小さなアプレットを作成する必要があります。次に、データベースのブロック・サイズと BACKUP コマンドの BLOCK FACTOR オプションを使用して、バックアップのパフォーマンスを最適化します。完全な構文と使用方法については、『リファレンス：文とオプション』を参照してください。

各 I/O 操作が最大ブロック・サイズに近づくほど、バックアップのパフォーマンスが向上します。BLOCK FACTOR の整数値にブロック・サイズを乗算したときに、ドライブのブロック・サイズにできるだけ近い値が得られるようにします。



データの整合性を保つために、Sybase IQ は書き込んだ各ブロックに余分なデータを追加することに注意してください。このため、データベースのブロック・サイズが 8192 で、テープ・デバイスで処理できる最大ブロック・サイズが 128KB の場合、本来なら  $8192 * 16 = 128KB$  となるはずですが、ブロック係数に 16 は使用できません。各 I/O 操作で Sybase IQ が追加する余分なデータを計算に入れ、BLOCK FACTOR に 15 を使用する必要があります。15 という値は、Windows でデフォルトのデータベース・ブロック・サイズとデフォルトの IQ ページ・サイズ (128KB) を使用したときのデフォルトのブロック係数です。



# 索引

## A

AGGREGATION\_ALGORITHM\_PREFERENCE  
    オプション 47  
AND キーワード 8  
AVG 関数 12

## B

BETWEEN 条件 10  
BLANK PADDING  
    OFF のサポート 31  
    ジョインへの影響 31  
BLOCK FACTOR  
    ロード・オプション 67

## C

CIS 機能補正  
    パフォーマンスの影響 41  
COUNT 関数 11, 12  
CPU  
    Sybase IQ に利用可能な数 86  
CPU 使用率  
    モニタリング 139, 146  
CPU 使用率統計 107  
CREATE DBSPACE 文 80

## D

DB 領域  
    最適なパフォーマンスのための検索 80  
    使用状況統計 112  
DEFAULT\_HAVING\_SELECTIVITY オプ  
    ション 47

DEFAULT\_LIKE\_MATCH\_SELECTIVITY オプ  
    ション 47  
DEFAULT\_LIKE\_RANGE\_SELECTIVITY オプ  
    ション 47  
DENSE\_RANK 関数 17

## E

EARLY\_PREDICATE\_EXECUTION オプ  
    ション 47  
EBF xii

## F

FLATTEN\_SUBQUERIES オプション 40  
FROM 句 5, 98  
    ジョイン 22  
From 句のないクエリの処理 5

## G

Getting Started CD x  
GROUP BY  
    パフォーマンスの推奨事項 36

## H

HASH\_THRASHING\_PERCENT オプション 138  
HG インデックス  
    マルチカラム 39

## I

- I/O
  - パフォーマンスの推奨事項 76
- I/O デバイス 154
- IN\_SUBQUERY\_PREFERENCE オプション 47
- INDEX\_ADVISOR オプション 42
- INDEX\_PREFERENCE オプション 47
- Interactive SQL での特定のプロシージャのプロファイリング情報の表示 122
- Interactive SQL でのプロシージャ・プロファイリング情報の表示 121
- IN 条件 10
- IQ PATH オプション
  - ロー・デバイスの選択 76
- iq\_dummy テーブル 5
- IQGOVERN\_MAX\_PRIORITY オプション 46
- IQGOVERN\_PRIORITY 46
- iqgovern スイッチ
  - パフォーマンスを向上させるためのクエリの制限 86
- IQMSG ログ
  - 最大サイズの設定 84
- iqnumbercpus
  - CPU 数の設定 86
- iqwmem スイッチ 71
- IQ ストア
  - バッファのキャッシュ・サイズ 63
- IQ ページ・サイズ
  - 決定 65

## J

- JOIN\_ALGORITHM\_PREFERENCE オプション 48

## L

- LIST 関数 12

## M

- MAIN\_CACHE\_MEMORY\_MB オプション 63
- MAX\_CURSOR\_COUNT オプション 88

- MAX\_HASH\_ROWS オプション 48
- MAX\_QUERY\_TIME オプション 46
- MAX\_STATEMENT\_COUNT オプション 88
- MAX\_TEMP\_SPACE\_PER\_CONNECTION オプション 86
- maximum 関数 12
- MAX 関数 12
- MIN 関数 12
- MySybase
  - EBF xii
  - 自分専用のビューの作成 xi

## N

- NOEXEC オプション 42
- NT CreateFile オプション 153
- NTFS キャッシュ
  - パフォーマンスの向上 152
  - メモリの横取り 148
- NTILE 関数 17
- NT タスク マネージャ 150
- NT パフォーマンス モニタ 150

## O

- ORDER BY 句 7, 39
- OR キーワード 11
- OS\_FILE\_CACHE\_BUFFERING オプション 73
- OVER 句 18

## P

- PERCENT\_RANK 関数 17
- PERCENTILE\_CONT 関数 17
- PERCENTILE\_DISC 関数 17
- PREFETCH\_BUFFER\_LIMIT オプション 88

## Q

- QUERY\_DETAIL オプション 42
- QUERY\_PLAN\_AFTER\_RUN オプション 43

QUERY\_PLAN\_AS\_HTML\_DIRECTORY オプション 43  
 QUERY\_PLAN\_AS\_HTML オプション 43  
 QUERY\_PLAN\_TEXT\_ACCESS オプション 43  
 QUERY\_PLAN\_TEXT\_CACHING オプション 43  
 QUERY\_PLAN オプション 42  
 QUERY\_TEMP\_SPACE\_LIMIT オプション 86  
 QUERY\_TIMING オプション 44

## R

RANK 関数 17  
 RAWDETECT  
   ディスク・ストライピング・オプション 79

## S

SELECT 文  
   説明 2, 32  
 sp\_iqcolumnuse プロシージャ  
   sp\_iqindexuse プロシージャ 49  
 sp\_iqtableuse プロシージャ 49  
 sp\_iqtable プロシージャ 3  
 sp\_iqunusedcolumn プロシージャ 49  
 sp\_iqunusedindex プロシージャ 49  
 sp\_iqunusedtable プロシージャ 49  
 sp\_iqworkmon プロシージャ 49  
 SUM 関数 12  
 SyBooks CD x

## T

TEMP\_CACHE\_MEMORY\_MB オプション 63

## U

UNION ALL  
   ビュー 98  
   ルール 98

## V

vmstat コマンド  
   UNIX 上でバッファ・キャッシュをモニタリング 139

## W

WHERE 句  
   BETWEEN 条件 10  
   ORDER BY 句 7  
   パターン・マッチング 9  
   日付の比較 8  
   例 7  
 Windows テープ・デバイス 154  
 WITHIN GROUP 句 18

## あ

アクセシビリティ  
   マニュアル xiv  
 アドレス空間  
   仮想 150  
 アポストロフィ  
   使用 7  
 アルファベット順 5

## い

イベント  
   個別プロファイリング情報の表示 120  
   プロファイリング・データの表示 113  
   要約プロファイリング・データの表示 118  
 インデックス  
   マルチカラム 39  
 インデックス・アドバイザ 92  
 インデックス・タイプ  
   パフォーマンス向上のための選択 92  
 引用符  
   使用 7

## お

- 応答時間 54
- 大文字と小文字の区別 3, 7
- オプション
  - FLATTEN\_SUBQUERIES 40
  - INDEX\_ADVISOR 42
  - NOEXEC 42
  - QUERY\_DETAIL 42
  - QUERY\_PLAN 42
  - QUERY\_PLAN\_AFTER\_RUN 43
  - QUERY\_PLAN\_AS\_HTML 43
  - QUERY\_PLAN\_AS\_HTML\_DIRECTORY 43
  - QUERY\_PLAN\_TEXT\_ACCESS 43
  - QUERY\_PLAN\_TEXT\_CACHING 43
  - QUERY\_TIMING 44
- 予期しない動作 5
- オプション値
  - トランケーション 41

## か

- カーソル
  - 数の制限 88
- 外部参照
  - 定義 33
- 仮想アドレス空間 150
- 仮想メモリ
  - 断片化 71
- カタログ・ストア 5
  - ファイルの増大 95
- カラム
  - 順序 6
  - 説明 3, 4
  - テーブルからの選択 6
  - 非常に多くの NULL 値 6, 36
- 関数
  - SOUNDEX 関数 10
  - 個別プロファイリング情報の表示 119
  - 集合 11
  - 統計 17
  - プロファイリング・データの表示 113
  - 要約プロファイリング・データの表示 118

## き

- キー・ジョイン
  - 使用 25
- キャッシュ
  - NTFS 152
  - バッファ 152
  - 「バッファ・キャッシュ」参照 123
- キャッシュ・サイズ
  - IQ メイン・バッファとテンポラリ・バッファ 63
- キャッシュ統計 108
- キャッシュ・ページ
  - プリフェッチ 88

## &lt;

- クエリ
  - SQL Anywhere による処理 5
  - 構築 36
  - 最適化 41, 47, 92
  - 推奨するインデックス 91
  - チューニング 154
  - 同時クエリの制限 86
  - メモリ使用の制限 86
- クエリ・サーバ
  - ロード・バランス 90
- クエリの最適化 39, 92
- クエリのパフォーマンス
  - CIS 機能補正の影響 41
  - SQL Anywhere のルールによる処理 41
  - カタログ・ストア・テーブル 41
  - データベース間のジョイン 41
- クエリ・プラン 41
  - グラフィカル 44
  - 実行せずに生成 42
  - 使用 44
- クエリ並行処理 40
- グループ化されたデータ 11

## け

検索条件  
サブクエリ 32

## こ

構文  
マニュアル表記規則 xii  
固定長 I/O  
Windows 154  
コマンド  
long 7  
コンポーネント  
動作確認 xi

## さ

サーバ  
パフォーマンスのモニタリング 105  
サブクエリ  
使用 32  
サブクエリのパフォーマンス 40  
サブクエリのフラット化 40  
サポート、テクニカル xiv  
サンプル・データベース xiv

## し

システム DB 領域 5  
システム・ストアド・プロシージャ 3  
システム・トリガ  
個別プロファイリング情報の表示 120  
プロファイリング・データの表示 113  
要約プロファイリング・データの表示 118  
システム・プロシージャ  
sp\_iqcolumnuse 49  
sp\_iqindexuse 49  
sp\_iqtableuse 49  
sp\_iqunusedcolumn 49  
sp\_iqunusedindex 49  
sp\_iqunusedtable 49  
sp\_iqworkmon 49

順次ディスク I/O 80  
ジョイン  
BLANK PADDING 31  
オブティマイザの単純化 48  
外積 22  
データ型 29  
ジョイン・インデックス  
パフォーマンスの影響 93  
条件  
GROUP BY 句 13  
探索 7, 8, 10  
書体  
表記規則 xiii  
マニュアル xii

## す

スイーパ・スレッド 136  
ストア I/O 統計 111  
ストアド・プロシージャ 3  
個別プロファイリング情報の表示 119  
パフォーマンスのモニタリング 113  
プロファイリング・データの表示 113  
要約プロファイリング・データの表示 118  
スラッシング  
HASH\_THRASHING\_PERCENT オプ  
ション 138  
回避策 137  
スループット 54  
最大化 148  
スレッド  
-iqnumbercpus スイッチによる使用の制御 86  
管理オプション 75  
バッファ・キャッシュ 136  
モニタリング 128  
スレッド統計 109  
スワッピング  
パフォーマンスへの影響 55  
必要なディスク領域 56  
メモリ 55  
スワップ・ファイル  
パフォーマンスへの影響 55

## 索引

### せ

- 制限 7
- 製品マニュアル x
- セグメント
  - データベース、複数のセグメントの使用 80
- 接続
  - 文の制限 88
- 接続統計 110

### そ

- 関連名
  - 説明 23
  - 定義 33
- 挿入オペレーション
  - チューニング 153

### た

- 第 508 条
  - 法令遵守 xiv
- タスク マネージャ 150
- ダミーの Sybase IQ テーブル 5
- 検索条件
  - 概要 7
  - ショートカット 10
  - 日付の比較 8
- 断片化 71

### ち

- チューニング 147
  - クエリ 154
  - 挿入オペレーション 153

### て

- ディスク・キャッシュ
  - 定義 91

- ディスク・ストライピング
  - Sybase IQ 77
  - 定義 77
  - 内部 79
  - ルール 78
  - ロードにおける使用 78
- ディスクのキャッシュ
  - パフォーマンスの影響 91
- ディスク容量
  - マルチプレックス・データベース 90
- ディスク領域
  - スワップ領域 56
- 低断片化ヒープ 71
- データ
  - 記憶域 152
- データ型
  - ジョインの要件 29
- データのロード
  - HG インデックスおよび WD インデックス 100
  - ストライプ・ディスクの使用 78
  - チューニング 153
  - パフォーマンス 67
  - 並列ロード 100
  - メモリ要件 60
  - ロード中のアクセス 100
- データベース
  - 管理 93
  - サンプル xiv
  - パフォーマンス向上のための非正規化 95
  - 非正規化の利点 96
- データベース・オプション
  - FLATTEN\_SUBQUERIES 40
- データベース間のジョイン
  - パフォーマンスの影響 41
- データベース・セグメント
  - 最適なパフォーマンスのための検索 80
- データベース・プロシージャ
  - プロファイリング・データの表示 113
- データベース・プロシージャのプロファイリング
  - 説明 113
- データ・ロードの改善 100
- テープ・デバイス
  - Windows 154



## テーブル

iq\_dummy 5

外部キー 24

結合 93

ジョイン 93

関連名 23

プライマリ・キー 24

リスト 3

テーブルのクエリ 5

テクニカル・サポート xiv

## デバイス

Windows の I/O 154

テンポラリー・ストア

バッファのキャッシュ・サイズ 63

## と

## 統計

動的 104

## 動作確認

マニュアル

更新 xi

動的パフォーマンス・モニタ 104

トランザクション・ステータス

モニタリング 104

トランザクション統計 111

トランザクション・ログ

説明 81

トランケート 81

マルチプレックスのトランケーション 83

## トリガ

個別プロファイリング情報の表示 120

要約プロファイリング・データの表示 118

## ね

## ネットワーク

設定 100

大量のデータ転送 100

パフォーマンス向上の推奨方法 100

ネットワーク統計 112

## は

## パーティション

定義 76

## ハイパースレッディング

サーバ・スイッチ 86

## パターン・マッチング 9

## バックアップ 83

ブロック・サイズのチューニング 154

## バッファ

オペレーティング・システム・バッファリング

の無効化 73

バッファ・キャッシュ 63, 152

サイズの決定 58

サイズの設定 63

モニタリング 123

レイアウト 136

バッファ・キャッシュのモニタリング 123

例 132

## バッファ・マネージャ

スラッシング 137

## パフォーマンス 147

CIS 機能補正の影響 41

I/O の分散 76

iqgovern によるクエリの制限 86

向上 14

向上のための設計 54

サブクエリ 40

正しいインデックス・タイプの選択 92

定義 54

ディスクのキャッシュ 91

データベースの非正規化の利点 96

動的モニタ 104

マルチユーザ 88

モニタリング 123

パフォーマンスとチューニングの問題 154

## パフォーマンスのチューニング

概要 104

## パフォーマンスのモニタリング

2 次元棒グラフ 106

3 次元棒グラフ 106

サーバ・レベル 105

時系列ビュー 106

## 索引

- チャートの保存 105
- 統計のカスタマイズ 105
- パフォーマンス・モニタ
  - チャートの印刷 106
  - チャートのカスタマイズ 106
  - チャート・ビューの切り替え 106
  - モニタ済みの統計 107
  - 例 132
- パラメータ
  - 関数 12
- 範囲 8

## ひ

- ヒープ
  - 低断片化 71
- 比較
  - 説明 7, 8
- 非正規化
  - 短所 96
  - パフォーマンスの利点 96
  - 理由 95
- 日付 8, 10
- ビュー
  - MySybase、自分専用の作成 xi
- 表記規則
  - 構文 xii
  - 書体 xiii
  - マニュアル xii, xiii
- 標準
  - 第 508 条への準拠 xiv

## ふ

- ファイル
  - 最適なパフォーマンスのための検索 80
- 負荷のモニタリング 49
- プッシュダウン・ジョイン 98, 99
- 物理メモリ
  - モニタリング 149
- 不等号、テスト 8
- プリフェッチされたキャッシュ・ページ 88

- プロシージャ・プロファイリング
  - Interactive SQL でのデータの表示 121
  - SQL でのクリア 116
  - SQL での無効化 116
  - SQL での有効化 114
  - SQL でのリセット 115
  - Sybase Central でデータを表示 113, 117
  - Sybase Central でのクリア 116
  - Sybase Central での無効化 116
  - Sybase Central での有効化 114
  - Sybase Central でのリセット 115
  - イベント 118, 120
  - 個別プロシージャの情報 119, 122
  - システム・トリガ 118, 120
  - ストアド・プロシージャと関数 118, 119
  - トリガ 118, 120
  - プロシージャの要約 121
- プロシージャ・プロファイリング・データの表示
  - Sybase Central 117
- プロシージャ・プロファイリングのクリア
  - SQL 116
  - Sybase Central 116
- プロシージャ・プロファイリングの無効化
  - SQL 116
  - Sybase Central 116
- プロシージャ・プロファイリングの有効化
  - SQL 114
  - Sybase Central 114
- プロシージャ・プロファイリングのリセット
  - SQL 115
  - Sybase Central 115
- プロセス
  - 増加 71
  - モニタリング 150
  - ワーキング・セット 150
- プロセス・スレッド・モデル 74
- ブロック・サイズ
  - IQ ページ・サイズとの関係 65
- プロファイリング情報
  - イベント 120
  - システム・トリガ 120
  - ストアド・プロシージャと関数 119
  - トリガ 120
- 分割されたテーブル 98, 99

## へ

並行処理

クエリ・ツリー 40

ページ

圧縮解除 152

ページ・フォールト 148

モニタリング 151

ページング

UNIXでのモニタリング 139

Windowsでのモニタリング 139

パフォーマンスへの影響 55

メモリ 55

## ほ

法令遵守

第 508 条 xiv

## ま

マニュアル

CD x

SQL Anywhere ix

Sybase IQ vii

アクセシビリティ機能 xiv

オンライン x

動作確認 xi

表記規則 xii, xiii

マルチカラム・インデックス 39

マルチスレッド

パフォーマンスの影響 74

マルチプレックス

パフォーマンスのモニタリング 104

マルチプレックス・データベース

ディスク容量 90

メモリ 57

## め

メイン・データベース

バッファのキャッシュ・サイズ 63

メッセージ・ログ

Sybase IQ 84

メモリ

オーバヘッド 60

クエリによる使用の制限 86

断片化 71

「バッファ・キャッシュ」参照 63

必要量の削減 66

ページング 55

マルチプレックス・データベース 57

連結 71

ロードの要件 60

メモリ使用状況統計 108

メンテナンス

ソフトウェア xii

メンテナンス、製品 xii

## も

モニタ

IQ UTILITIES の構文 123

起動と停止 123

出力ファイルの場所の指定 124

モニタリング

仮想アドレス空間 150

トランザクション・ステータス 104

物理メモリ 149

ページ・フォールト 151

ワーキング・セット 150

## ゆ

ユーザ指定の条件

クエリ 48

ユーザ定義関数

パフォーマンスの影響 41

## 索引

### よ

要求統計 110  
要約プロファイリング・データ  
イベント 118  
システム・トリガ 118  
ストアド・プロシージャと関数 118  
トリガ 118

### ら

ライトウェイト・プロセス 74

### れ

連結メモリ 71  
連邦リハビリテーション法  
第 508 条 xiv

### ろ

ロー  
説明 3  
選択 7  
ロー・デバイス  
パフォーマンスへの影響 76  
ロードの改善 100  
ロード・バランス  
クエリ・サーバ間 90  
ロー・パーティション  
メモリの使用 60

### わ

ワーキング・セット 150