



クエリ・パフォーマンスの調査

Version 2024.1
2024-06-03

クエリ・パフォーマンスの調査

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼働および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

1 SQL 文と SQL 統計の分析	1
1.1 SQL 文を作成する操作	1
1.1.1 その他の SQL 文の操作	2
1.2 SQL 文のリスト	2
1.2.1 リストの列	3
1.2.2 プランの状態	4
1.2.3 SQL 文テキスト	4
1.2.4 古い SQL 文	5
1.3 データ管理 (DML) SQL 文	5
1.3.1 SELECT コマンド	6
1.4 SQL 文の詳細	6
1.4.1 [文の詳細] セクション	7
1.4.2 [コンパイル設定] セクション	8
1.4.3 [ルーチン] セクションと [リレーション] セクション	9
1.5 SQL 文の問い合わせ	10
1.6 SQL 文のエクスポートとインポート	11
1.6.1 SQL 文のエクスポート	11
1.6.2 SQL 文のインポート	12
1.6.3 バックグラウンド・タスクの表示と削除	12
1.7 SQL 実行時統計情報	12
1.7.1 実行時統計とプラン表示	13
2 SQL クエリ・プランの解釈	15
2.1 プランの表示	15
2.1.1 EXPLAIN の使用法	16
2.1.2 管理ポータルでのプラン表示の使用法	16
2.2 プランの構造	17
2.3 プランの読み取り	18
2.3.1 マップへのアクセス	18
2.3.2 条件および式	18
2.3.3 ループ	19
2.3.4 一時ファイル	19
2.3.5 モジュール	19
2.3.6 処理のために送信されるクエリ	19
2.3.7 サブクエリ、JOIN、および UNION	19
2.4 別のプランを表示	20
2.4.1 統計	20
3 SQL パフォーマンス分析ツールキット	21
3.1 SQL パフォーマンス統計の有効化	21
3.1.1 アクション・オプション	22
3.1.2 収集オプション	22
3.1.3 終了オプション	22
3.1.4 管理ポータルでのパフォーマンス統計の有効化	23
3.2 統計設定の取得	23
3.3 クエリ・パフォーマンス統計のエクスポート	23
3.3.1 統計値	24
3.4 クエリ・パフォーマンス統計の削除	24

3.5 パフォーマンス統計の例	25
4 SQL パフォーマンスのヘルプの入手	27
4.1 レポート生成	27

1

SQL 文と SQL 統計の分析

このページでは、SQL 文と SQL 実行時統計情報を表示して読み取り、クエリのパフォーマンスを分析する方法について説明します。これらのツールはどちらも、システム上でクエリがどの程度のパフォーマンスで動作しているかを概要レベルで把握するうえで有用です。

SQL 文は、SQL クエリとその他の操作のレコードをテーブルごとに提供します。このような操作として、挿入、更新、削除などがあります。これらの文は、クエリ・プランにリンクしていて、このリンクは[そのクエリ・プランを凍結する](#)オプションを提供します。SQL DML 操作のたびに SQL 文が作成されてリストに格納されます。このリストは、管理ポータルで確認できます。テーブル定義を変更する場合は、このリストを使用して、それぞれの SQL 操作のクエリ・プランがその DML 変更の影響を受けるかどうか、また SQL 操作を変更する必要があるかどうかを判断できます。その後で、以下のことが可能になります。

- ・ それぞれの SQL 操作に、どのクエリ・プランを使用するかを判断が可能になります。テーブル定義の変更を反映して改訂したクエリ・プランを使用できるほか、現在のクエリ・プランを凍結して、テーブル定義を変更する前に生成されたクエリ・プランを維持することもできます。
- ・ テーブル定義への変更に基づいて、そのテーブルに対して SQL 操作を実行するルーチンにコードの変更を加えるかどうかの判断が可能になります。

注釈 SQL 文は、テーブル定義への変更の影響を受ける可能性がある SQL ルーチンのリストです。テーブル定義やテーブル・データへの変更の履歴としては使用しないでください。

SQL 実行時統計情報により、すべての SQL クエリの実行に関する基本的なメトリックを追跡して、クエリのパフォーマンス概要の履歴を入手できます。この統計情報の収集を有効にしておくと、収集プロセスがタイムアウトするまで、クエリの実行時パフォーマンスに関するメトリックが自動的に収集されます。この統計情報を分析すると、スキーマがどのようなパフォーマンスで動作しているかに関する洞察が得られ、[SQL パフォーマンス分析ツールキット](#)を使用して、どのクエリをさらに詳しく調査する必要があるかを判断できます。

1.1 SQL 文を作成する操作

以下の SQL 操作により、対応する SQL 文が作成されます。

- ・ **データ管理 (DML) 操作**: テーブルに対するクエリと、挿入操作、更新操作、削除操作が含まれます。各データ管理 (DML) 操作 ([ダイナミック SQL](#) と [埋め込み SQL](#) の両方) では、操作の実行時に SQL 文が作成されます。[ダイナミック SQL](#) の SELECT コマンドは、管理ポータルの [クエリ・キャッシュ](#) リストにエントリを作成します。
- ・ **埋め込み SQL** カーソルベースの SELECT コマンドでは、[OPEN](#) コマンドで DECLARED クエリを呼び出した場合にのみ、SQL 文が生成されます。管理ポータルの [\[クエリキャッシュ\]](#) リストに別個のエントリは作成されません。

クエリが複数のテーブルを参照している場合は、参照されるすべてのテーブルを[テーブル/ビュー/プロシージャ名]列にリストする 1 つの SQL 文がネームスペースの [SQLステートメント] リストに作成されます。参照される個々のテーブルそれぞれの [テーブルの SQL 文] リストに、そのクエリに対応するエントリが含まれます。

SQL 文は、クエリが初めて準備される際に作成されます。複数のクライアントが同じクエリを発行した場合は、最初の準備のみが記録されます。例えば、JDBC がクエリを発行し、続いて ODBC が同一のクエリを発行した場合、SQL 文のインデックスには最初の JDBC クライアントに関する情報のみが設定され、ODBC クライアントに関する情報は設定されません。

ほとんどの SQL 文には、クエリ・プランが関連付けられています。クエリ・プランは、作成時には凍結されていません。その後で、このクエリ・プランを凍結プランとして指定できます。クエリ・プランを持つ SQL 文には、SELECT 操作を伴う DML コマンドが含まれます。クエリプランを持たない SQL 文については、後述する“プランの状態”のセクションに示します。

注釈 SQL 文は、最新バージョンの SQL 操作のみをリストします。SQL 文を凍結していなければ、InterSystems IRIS® データ・プラットフォームによって次のバージョンに置き換えられます。そのため、ルーチン内で SQL コードを書き換えて呼び出すと、古い SQL コードは SQL 文から削除されます。

1.1.1 その他の SQL 文の操作

以下の SQL コマンドは、より複雑な SQL 文の操作を実行します。

- CREATE TRIGGER:トリガが定義されたテーブルでは、トリガの定義時とトリガの起動時のどちらの場合も SQL 文は作成されません。ただし、そのトリガが別のテーブルに対して DML 操作を実行する場合は、トリガを定義することで、そのトリガのコードで変更されるテーブルに対する SQL 文が作成されます。[場所] には、トリガが定義されているテーブルを指定します。SQL 文は、トリガの定義時に定義されます。トリガを削除すると、その SQL 文は削除されます。トリガを起動しても、SQL 文は作成されません。
- CREATE VIEW は、コンパイルされるものがないため、SQL 文が作成されません。また、ソース・テーブルに含まれる SQL 文のプラン・タイムスタンプが変更されることもありません。ただし、ビューの DML コマンドをコンパイルすると、そのビューに対する SQL 文が作成されます。

1.2 SQL 文のリスト

以下のように、管理ポータルの SQL インタフェースから SQL 文をリストできます。

- [SQL 文] タブ: ネームスペース内のすべての SQL 文がリストされます。リストはスキーマごとの照合シーケンスで示されてから、それぞれのスキーマ内のテーブル名/ビュー名ごとに示されます。このリストには、現在のユーザが権限を持つテーブル/ビューのみが含まれます。SQL 文が複数のテーブルを参照する場合は、参照されるすべてのテーブルが [テーブル/ビュー/プロシージャ名] 列にアルファベット順にリストされます。

列のヘッダをクリックすると、SQL 文のリストを [テーブル/ビュー/プロシージャ名]、[プランの状態]、[場所]、[SQL 文テキスト] などのリストの列でソートできます。これらのソート可能な列を使用すると、例えば、すべての凍結プラン ([プランの状態])、すべてのクエリ・キャッシュ ([場所])、最も遅いクエリ ([平均時間]) などをすばやく検索できます。

このタブにある [フィルタ] オプションを使用すると、リストされている SQL 文を特定のサブセットに絞り込むことができます。指定したフィルタ文字列により、SQL 文のリストに含まれるすべてのデータにフィルタが適用されます。最も役立つフィルタは、スキーマ名またはスキーマ名.テーブル名、ルーチンの場所、または SQL 文テキスト内の部分文字列に対するフィルタです。フィルタ文字列では、大文字と小文字は区別されません。ただし、文テキストの句読点の空白に従う必要があります (name,age ではなく name , age)。クエリが複数のテーブルを参照している場合、[テーブル/ビュー/プロシージャ名] 列で参照されるテーブルを選択すると、[フィルタ] には SQL 文が含まれます。[フィルタ] オプションは、ユーザによってカスタマイズされます。

[最大行数] オプションの既定値は 1,000 です。最大値は 10000 です。最小値は 10 です。10,000 を超える SQL 文をリストするには、[INFORMATION_SCHEMA.STATEMENTS](#) を使用します。[ページサイズ] オプションと [最大行数] オプションは、[ユーザによってカスタマイズ](#)されます。

- ・ [カタログの詳細] タブ: テーブルを選択して、そのテーブルのカタログ詳細を表示します。このタブには、[テーブルの SQL 文] ボタンがあり、そのテーブルに関連付けられている SQL 文を表示できます。SQL 文が複数のテーブルを参照している場合、その SQL 文は参照されるテーブルごとに [テーブルの SQL 文] リストに示されますが、[テーブル名] 列には現在選択されているテーブルのみが示されます。

列のヘッダをクリックすると、リストに含まれる任意の列でテーブルの SQL 文のリストをソートできます。

[INFORMATION_SCHEMA.STATEMENTS](#) を使用すると、さまざまな基準で選択した SQL 文をリストできます。詳細は、後述の “[SQL 文の問い合わせ](#)” を参照してください。

1.2.1 リストの列

[SQL 文] タブには、ネームスペースのすべての SQL 文がリストされます。[カタログの詳細] タブの [テーブルの SQL 文] ボタンをクリックすると、選択したテーブルの SQL 文がリストされます。どちらのリストにも、以下の列見出しが含まれています。

- ・ #: リスト行の連続番号。これらの番号は、特定の SQL 文に関連付けられているわけではありません。
- ・ テーブル/ビュー/プロシージャ名: [SQL テーブル \(またはビューまたはプロシージャ\) の修飾名](#)。クエリで複数のテーブルまたはビューを参照する場合、以下のようになります。
 - ネームスペースの [SQL ステートメント] タブの [テーブル/ビュー/プロシージャ名] に、すべてのテーブルとビューが照合順 (大文字と小文字を区別せず、アルファベット順) にリストされます。テーブルとビューを見分けるには、[SQL 文] を選択して [SQL 文の詳細] の [\[ステートメントは以下のリレーションを使用します\]](#) を表示します。
 - [テーブルの SQL 文] には、そのテーブルを参照するすべての SQL 文がリストされます。[テーブル/ビュー/プロシージャ名] には、指定したテーブルのみがリストされます。したがって、複数のテーブルの [テーブルの SQL 文] には、同じ SQL 文が表示されることがあります。
- ・ プランの状態
- ・ 新しいプラン
- ・ 実行回数
- ・ 実行回数/日
- ・ 合計時間
- ・ 平均時間
- ・ StdDev 時間
- ・ 行数
- ・ 行数/日
- ・ 場所: コンパイルされたクエリの場所:
 - Dynamic SQL: クエリ・キャッシュ名。例えば %sqlcq.USER.cls2.1 のようになります。
 - 埋め込み SQL: ルーチン名。例えば MySQL のようになります。
 - ストアド・プロシージャ: ストアド・プロシージャのクラス名。例えば Sample.procNamesJoinSP.1 のようになります。
- ・ SQL 文テキスト: 正規化された形式の SQL 文テキスト (128 文字に切り捨てられます)。後述の “[SQL 文テキスト](#)” に記載されているように、これはコマンド・テキストとは異なる場合があります。

このタブに表示される [フィルタ] オプションを使用して、[場所] 列の値でフィルタ処理することができます。

1.2.2 プランの状態

[プランの状態] には、以下のいずれかがリストされます。

- ・ [Unfrozen](#):凍結されていませんが、凍結できます。
- ・ [Unfrozen/Parallel](#):凍結されておらず、凍結することもできません。
- ・ [Frozen/Explicit](#):ユーザによって凍結されていますが、未凍結にすることができます。
- ・ [Frozen/Upgrade](#):InterSystems IRIS バージョンのアップグレードによって凍結されていますが、凍結解除できます。
- ・ 空欄:関連付けられたクエリ・プランがありません。
 - INSERT...VALUES() コマンドは、関連付けられたクエリ・プランがない SQL 文を作成するため、未凍結にすることも凍結することもできません ([プランの状態] 列は空欄になります)。この SQL コマンドはクエリ・プランを生成しませんが、SQL 文のリストに加えておくくと便利です。こうすることで、このテーブルに対するすべての SQL 操作をすばやく見つけることができます。例えば、テーブルに列を追加する場合は、そのテーブルに対する SQL INSERT が存在する場所をすべて調べて、この新しい列が含まれるように該当するコマンドを更新する必要があります。
 - カーソル・ベースの UPDATE コマンドや DELETE コマンドには、関連付けられるクエリ・プランがありません。そのため、未凍結にすることも凍結することもできません ([プランの状態] 列は空欄になります)。宣言されたカーソルに対して OPEN コマンドを実行すると、クエリ・プランが関連付けられた SQL 文が生成されます。カーソル (FETCH cursor、UPDATE...WHERE CURRENT OF cursor、DELETE...WHERE CURRENT OF cursor、および CLOSE cursor) を使用する埋め込み SQL 文は、別個の SQL 文を生成しません。カーソル・ベースの UPDATE または DELETE はクエリ・プランを生成しませんが、SQL 文のリストに加えておくくと便利です。こうすることで、このテーブルに対するすべての SQL 操作をすばやく見つけることができます。

1.2.3 SQL 文テキスト

一般に、SQL 文テキストは SQL コマンドとは異なります。これは、SQL 文の生成によって、大文字/小文字と空白が正規化されるためです。それ以外の相違点は、以下のとおりです。

- ・ 管理ポータル・インタフェースまたは SQL シェル・インタフェースからクエリを発行すると、結果の SQL 文は、SELECT 文の前に DECLARE QRS CURSOR FOR が付加される点で、クエリとは異なります (“QRS” は、生成されたカーソル名的一种になります)。これにより、その文テキストはダイナミック SQL クエリ・キャッシュの文テキストと照合できるようになります。
- ・ SQL コマンドで未修飾のテーブル名またはビュー名を指定した場合、結果の SQL 文では、[スキーマ検索パス](#) (DML の場合で、指定されているとき) または [既定のスキーマ名](#) を使用してスキーマが提供されます。
- ・ SQL 文テキストは、1,024 文字以降が切り捨てられます。完全な SQL 文テキストを確認するには、[\[SQL 文の詳細\]](#) を表示します。
- ・ 単一の SQL コマンドが、複数の SQL 文を生成する場合があります。例えば、クエリがビューを参照する場合、[\[SQL 文\]](#) には、2 つの文テキストが表示されます。1 つはビュー名の下側にリストされ、もう 1 つは基になるテーブル名の下側にリストされます。どちらかの文を凍結すると、両方の文の [\[プランの状態\]](#) が [凍結] になります。
- ・ SQL 文がデータベース・ドライバを介して準備された場合、SQL 文の生成によって、SQL コメント・オプション (#OPTIONS) が文テキストに追加されます ([文 Index のハッシュ](#) を生成するために、それらのオプションが必要な場合)。以下に例を示します。

```
DECLARE C CURSOR FOR SELECT * INTO :%col(1) , :%col(2) , :%col(3) , :%col(4) , :%col(5)
FROM SAMPLE . COMPANY /*#OPTIONS {"xDBCisoLevel":0} */
```


1.2.4 古い SQL 文

SQL 文に関連付けられているルーチンまたはクラスを削除した場合、その SQL 文リストは自動的に削除されません。このようなタイプの SQL 文リストを「Stale (古い)」と呼びます。この履歴情報や SQL 文に関連付けられた使用率統計にアクセスできると役立つことが多いため、これらの古いエントリは管理ポータル の [SQL 文] リストに保持されています。

[古いものをクリーン] ボタンを使用して、これらの古いエントリを削除できます。**[Clean stale]** を使用すると、関連付けられたルーチンまたはクラス (テーブル) が存在しなくなったか、SQL 文クエリが含まれなくなった、凍結されていない SQL 文がすべて削除されます。**[Clean stale]** では、凍結された SQL 文は削除されません。`$SYSTEM.SQL.Statement.Clean()` メソッドを使用して、同じ Clean stale 操作を実行できます。

SQL 文に関連付けられたテーブル (永続クラス) を削除した場合、**[テーブル/ビュー/プロシージャ名]** 列が変更されます (例: `SAMPLE.MYTESTTABLE - Deleted??`)。削除されたテーブルの名前はすべて大文字に変換され、`"Deleted??"` のフラグが設定されます。または、SQL 文が複数のテーブルを参照している場合は、`SAMPLE.MYTESTTABLE - Deleted?? Sample.Person` のようになります。

- ・ ダイナミック SQL クエリの場合、テーブルを削除すると、そのテーブルに関連付けられているクエリ・キャッシュがすべて自動的に削除されるため、**[場所]** 列は空白になります。**[Clean stale]** によって SQL 文が削除されます。
- ・ 埋め込み SQL クエリの場合、**[場所]** 列には、クエリの実行に使用されるルーチンの名前が含まれます。ルーチンを変更したために元のクエリが実行されなくなった場合、**[場所]** 列は空白になります。**[Clean stale]** によって SQL 文が削除されます。クエリで使用されているテーブルを削除した場合、テーブルには `"Deleted??"` のフラグが設定されます。**[Clean stale]** で SQL 文は削除されません。

注釈 SQL 文のインデックスの中には、古くなっているものやルーチン参照が古くなっているものがあるので、それらをクリーンアップするためのシステム・タスクが、すべてのネームスペースで 1 時間に 1 回、自動的に実行されます。この処理はシステム・パフォーマンスを維持するために実行されます。この内部的なクリーンアップは、管理ポータル の SQL 文リストに反映されません。

1.3 データ管理 (DML) SQL 文

SQL 文を作成するデータ管理言語 (DML) のコマンドは、INSERT、UPDATE、INSERT OR UPDATE、DELETE、TRUNCATE TABLE、SELECT、および宣言されたカーソルベースの SELECT に対する OPEN カーソルです。ダイナミック SQL や埋め込み SQL を使用して DML コマンドを呼び出すことができます。DML コマンドは 1 つのテーブルまたは 1 つのビューに対して呼び出すことができます。InterSystems IRIS は対応する SQL 文を 1 つ作成します。

注釈 ダイナミック SQL が準備される時、または埋め込み SQL カーソルを開くときに SQL 文が作成されますが、DML コマンドが実行される時には作成されません。SQL 文のタイムスタンプには、この SQL コードの呼び出しが発生した時刻が記録されますが、そのクエリが実行された時刻 (または実行されたかどうか) は記録されません。そのため、SQL 文が実際には一度も実行されたことのないテーブル・データへの変更を表していることがあります。

ダイナミック SQL DML コマンドを準備すると、対応する SQL 文が作成されます。この SQL 文に関連付けられている **[場所]** は、クエリ・キャッシュになります。ダイナミック SQL は、**管理ポータル SQL インタフェース** または **SQL Shell インタフェース** から SQL を実行したときや、**.txt ファイルからインポート** したときに準備されます。未凍結のクエリ・キャッシュを削除すると、対応する SQL 文に **[Clean stale]** の削除対象としてフラグが設定されます。凍結されたクエリ・キャッシュを削除すると、対応する SQL 文の **[場所]** の値が削除されます。SQL 文を凍結解除すると、その SQL 文に **[Clean stale]** の削除対象としてフラグが設定されます。

非カーソル埋め込み SQL データ管理言語 (DML) コマンドを実行すると、対応する SQL 文が作成されます。埋め込み SQL DML コマンドごとに、対応する SQL 文が作成されます。1 つのルーチンに複数の埋め込み SQL コマンドが含ま

れている場合は、それぞれの埋め込み SQL コマンドごとに個別の SQL 文が作成されます(一部の埋め込み SQL コマンドは、複数の SQL 文を作成します)。SQL 文リストの **[場所]** 列には、埋め込み SQL を含んでいるルーチンが明示されます。この方法で、SQL 文は埋め込み SQL DML コマンドごとのレコードを維持します。

カーソル・ベースの埋め込み SQL データ管理言語 (DML) ルーチンを開くと、SQL 文がクエリ・プランと共に作成されます。関連する埋め込み SQL 文 (FETCH cursor、CLOSE cursor) により、別個の SQL 文が生成されることはありません。FETCH cursor に続く、関連する UPDATE table WHERE CURRENT OF cursor または DELETE FROM table WHERE CURRENT OF cursor は個別の SQL 文を生成しますが、個別のクエリ・プランは生成しません。

リテラル値を挿入する INSERT コマンドは、**[プランの状態]** 列が空白の SQL 文を作成します。このコマンドはクエリ・プランを作成しないため、SQL 文を凍結することはできません。

1.3.1 SELECT コマンド

クエリを呼び出すと、対応する SQL 文が作成されます。これは、単純な SELECT 操作、または CURSOR ベースの SELECT/FETCH 操作になります。クエリは、テーブルまたはビューに対して発行できます。このような種類のクエリの重要な詳細を以下に挙げます。

- ・ **JOIN** を含んでいるクエリは、各テーブルにまったく同じ SQL 文を作成します。**[場所]** は、各テーブルのリストに含まれる同一のストアド・クエリになります。**[ステートメントは以下のリレーションを使用します]** には、“SQL 文の詳細”の **“[ルーチン] セクションと [リレーション] セクション”** で説明するように、すべてのテーブルがリストされます。
- ・ **selectItem サブクエリ** を含んでいるクエリは、各テーブルにまったく同じ SQL 文を作成します。**[場所]** は、各テーブルのリストに含まれる同一のストアド・クエリになります。**[ステートメントは以下のリレーションを使用します]** には、“SQL 文の詳細”の **“[ルーチン] セクションと [リレーション] セクション”** で説明するように、すべてのテーブルがリストされます。
- ・ 外部 (リンクされた) テーブルを参照するクエリは凍結できません。
- ・ FROM 節に **%PARALLEL** キーワードを含んでいるクエリは、複数の SQL 文を作成することがあります。生成されたこれらの SQL 文を表示するには、以下を呼び出します。

SQL

```
SELECT * FROM INFORMATION_SCHEMA.STATEMENT_CHILDREN
```

これにより、元のクエリの文ハッシュを含む Statement 列と、生成されたバージョンのクエリの文ハッシュを含む ParentHash 列が表示されます。

%PARALLEL クエリの SQL 文のプランの状態は [Unfrozen/Parallel] であり、凍結することはできません。

- ・ FROM 節を含んでいないためにテーブルを参照しないクエリも、SQL 文を作成します。例えば、**SELECT \$LENGTH('this string')** は、**[テーブル]** 列の値が **%TSQL_sys.snf** の SQL 文を作成します。

1.4 SQL 文の詳細

[SQL 文の詳細] は、以下の 2 つの方法で表示できます。

- ・ **[SQL 文]** タブで、左側の列の **[テーブル/ビュー/プロシージャ名]** リンクをクリックして、SQL 文を選択します。これによって、別のタブに **[SQL 文の詳細]** が表示されます。このインタフェースを使用すると、複数のタブを開いて比較できます。このインタフェースでは、**[SQL 実行時統計]** ページを表示するための **[クエリ・テスト]** ボタンも提供されます。
- ・ テーブルの **[カタログの詳細]** タブ (または **[SQL 文]** タブ) で、右側の列の **[文テキスト]** リンクをクリックして、SQL 文を選択します。これによって、ポップアップ・ウィンドウに **[SQL 文の詳細]** が表示されます。

どちらかの [SQL 文の詳細] 表示を使用して、クエリ・プランを確認したり、特定のクエリ・プランの凍結および凍結解除を実行したりできます。

[SQL 文の詳細] には、クエリ・プランを凍結または凍結解除するためのボタンが用意されています。また、[ステートメント詳細] に表示された使用率統計をクリアするための [SQL 統計をクリア] ボタン、1 つ以上の SQL 文をファイルにエクスポートするための [エクスポート] ボタン、ページを更新するためのボタンや閉じるためのボタンも用意されています。

[SQL 文の詳細] 表示には以下のセクションがあります。これらの各セクションのタイトルの横にある矢印アイコンを選択することによって、各セクションを展開したり、折りたたんだりすることができます。

- ・ 使用率統計が表示される [ステートメント詳細]
- ・ [コンパイル設定]
- ・ [文は以下のルーチンで定義されています]
- ・ [文は以下のリレーションを使用します]
- ・ [文テキストとクエリ・プラン] (他の場所で説明しています)

1.4.1 [文の詳細] セクション

[文の詳細] セクション:

- ・ **[プランの状態]:** [凍結/明示]、[凍結/アップグレード]、[凍結解除]、または [未凍結/パラレル]。[Frozen/Explicit] は、この文のプランがユーザの明示的な操作によって凍結されていて、この SQL 文を生成したコードへの変更にかかわらず、この凍結プランがクエリ・プランとして使用されることを表しています。[Frozen/Upgrade] は、この文のプランが InterSystems IRIS バージョンのアップグレードによって自動的に凍結されていることを表しています。[Unfrozen] は、現在、プランが凍結されていないことと、凍結される可能性があることを表しています。[Unfrozen/Parallel] は、プランが凍結されていないことと、%PARALLEL 処理を使用するために凍結できないことを表しています。NULL (空) のプランの状態は、関連付けられているクエリ・プランが存在しないことを表します。
- ・ **[合計時間]:** このクエリの実行にかかった時間 (秒)。
- ・ **[バージョン]:** プランを作成した InterSystems IRIS のバージョン。[プランの状態] が [Frozen/Upgrade] の場合は、InterSystems IRIS の以前のバージョンです。クエリ・プランの凍結を解除すると、[プランの状態] は [Unfrozen] に変わり、[バージョン] は現在の InterSystems IRIS バージョンに変わります。
- ・ **[実行回数]:** このクエリが実行された回数を示す整数値。このクエリのクエリ・プランが別のプランになるような変更があると (テーブルへのインデックスの追加など)、この回数はリセットされます。
- ・ **[平均時間]:** このクエリの実行にかかった平均時間 (秒)。クエリがキャッシュ・クエリの場合、初めてクエリを実行したときにかかる時間は、次回以降にそのクエリ・キャッシュから最適化されたクエリを実行する際にかかる時間よりも大幅に長い可能性が高いです。
- ・ **[行数]:** このクエリで返された行または変更された行の合計数。
- ・ **[最初に見た日付]:** クエリが初めて実行されたときの日付。これは、クエリが準備された日時である [最終コンパイル日時] とは異なります。
- ・ **[実行回数/日]:** このクエリが 1 日に実行された平均回数。
- ・ **[StdDev 時間]:** 平均実行時間を基準とした、このクエリの合計実行時間の標準偏差。この標準偏差は、このクエリの実行時間が示す変動性の尺度です。
- ・ **[行数/日]:** このクエリによって 1 日に返された行数または変更された行数の平均値。
- ・ **[タイムスタンプ]:** 最初のタイムスタンプは、プランが作成されたときのものです。このタイムスタンプは、凍結/凍結解除の後に更新され、プランが未凍結になった時刻が記録されます (プランが再コンパイルされた時刻ではありません)。凍結解除のタイムスタンプを表示するには、[ページの更新] ボタンのクリックが必要になることがあります。文を

含んでいるルーチン/クラスの日付/時刻値を[プラン・タイムスタンプ]と比較することで、ルーチン/クラスが再コンパイルされた場合に、同じクエリ・プランを使用していないことを確認できます。

- ・ **[凍結プランが異なる]**: プランを凍結すると、この追加フィールドが表示され、凍結プランが未凍結プランと異なるかどうかを示されます。プランを凍結すると、[\[文テキストとクエリ・プラン\]](#) に凍結プランと未凍結プランが並んで表示され、比較しやすくなります。
- ・ **[ステートメント・ハッシュ]**: 文定義の内部ハッシュ表現です。これは、SQL 文 Index のキーとして使用されます (内部使用専用)。場合によっては、同じ SQL 文のように見えても、文のハッシュ・エントリが異なることがあります。SQL 文の異なるコード生成を必要とする設定/オプションの相違によって、異なる文ハッシュが生成されます。これは、異なる内部最適化をサポートする異なるクライアント・バージョンや異なるプラットフォームで発生することがあります。

1.4.1.1 使用率統計

[ステートメント詳細] セクションでは、以下の各フィールドに SQL 文の使用率統計が表示されます。

- ・ 合計時間
- ・ 実行回数
- ・ 平均時間
- ・ 最初に見た日付
- ・ 実行回数/日
- ・ StdDev 時間
- ・ 行数
- ・ 行数/日

この情報は、最も遅いクエリと実行回数が最も多いクエリを確認するために使用できます。この情報を使用して、最適化によってパフォーマンスを大幅に向上できるクエリを特定できます。

クエリの使用率統計は定期的に更新され、完了したクエリ実行が反映されます。[\[SQL 統計をクリア\]](#) ボタンを使用して、これらのフィールドの値をクリアできます。

InterSystems IRIS では、%PARALLEL サブクエリの使用率統計が独立して記録されません。%PARALLEL サブクエリの統計は、外側のクエリの統計と共に合計されます。並列で実行されるように実装によって生成されたクエリには、それぞれを別々に追跡する使用率統計はありません。

複数の SQL 文を対象としたこれらのクエリ使用率統計は、[\[SQL 文\] タブの表示](#)で確認できます。[\[SQL 文\]](#) タブのリストは任意の列で並べ替えることができます。これによって、例えば、平均時間が最大のクエリを簡単に特定できます。

["SQL 文の問い合わせ"](#) の説明にあるように、`INFORMATION.SCHEMA.STATEMENTS` クラス・プロパティをクエリすることによって、これらのクエリの使用率統計にアクセスできます。また、[管理ポータル](#)の [\[SQL アクティビティ\]](#) ページには、進行中のクエリについて、これらの使用率統計が表示されます。

1.4.2 [コンパイル設定] セクション

[コンパイル設定] セクション:

- ・ **[選択モード]**: 文がコンパイルされたときの SelectMode。DML コマンドの場合は、[#sqlcompile select](#) を使用して設定できます。既定値は Logical です。[#sqlcompile select=Runtime](#) を指定した場合は、`$SYSTEM.SQL.Util.SetOption()` メソッドの SelectMode オプション呼び出しにより、クエリ結果セットの表示を変更できますが、**[選択モード]** の値は変更されず、Runtime のままです。
- ・ **[デフォルト・スキーマ]**: 文がコンパイルされたときに設定された既定のスキーマ名。これは一般的に、コマンドが発行されたときに適用されていた既定のスキーマです。ただし、SQL は、既定のスキーマではなく、[スキーマ検索パス](#)

(指定されている場合)を使用して未修飾名のスキーマを解決した可能性があります。文が、1 つ以上の `#Import` マクロ指示文を使用した埋め込み SQL 内の DML コマンドである場合は、`#import` 指示文によって指定されたスキーマがここにリストされます。

- ・ **[スキーマ・パス]:** 文がコンパイルされたときに定義されたスキーマ・パス。これは **スキーマ検索パス** です (指定されている場合)。スキーマ検索パスが指定されていない場合、この設定は空白です。ただし、`#Import` マクロ指示文で検索パスが指定された DML 埋め込み SQL コマンドの場合は、`#import` 検索パスが **[デフォルト・スキーマ]** 設定に表示され、この **[スキーマ・パス]** 設定は空白になります。
- ・ **[プランのエラー]:** このフィールドは、凍結プランの使用時にエラーが発生した場合にのみ表示されます。例えば、クエリ・プランでインデックスが使用されていて、そのクエリ・プランが凍結されている場合、そのインデックスをテーブルから削除すると、**[プランのエラー]** に次のようなエラーが発生します: `Map 'NameIDX' not defined in table 'Sample.Person', but it was specified in the frozen plan for the query.` インデックスを削除または追加すると、テーブルがリコンパイルされて、**[最終コンパイル日時]** の値が変更されます。エラーの原因になった状態を (例えば、欠落しているインデックスを再作成して) 修正した後で、**[エラーをクリア]** ボタンを使用して **[プランのエラー]** フィールドをクリアできます。エラー状態の修正後に **[エラーをクリア]** ボタンを使用すると、**[プランのエラー]** フィールドと **[エラーをクリア]** ボタンはどちらも表示されなくなります。詳細は、“[エラー状態の凍結プラン](#)” を参照してください。

1.4.3 [ルーチン] セクションと [リレーション] セクション

[文は以下のルーチンで定義されています] セクション:

- ・ **[ルーチン]:** クエリ・キャッシュ (ダイナミック SQL DML の場合)、またはルーチン名 (埋め込み SQL DML の場合) に関連付けられたクラス名。
- ・ **[タイプ]:** [クラスメソッド] または [MAC ルーチン] (埋め込み SQL DML の場合)。
- ・ **[最終コンパイル日時]:** ルーチンの最終コンパイル日時または準備日時。SQL 文が凍結されていない場合は、MAC ルーチンを再コンパイルすることで、このタイムスタンプと **[プランのタイムスタンプ]** の両方が更新されます。SQL 文が凍結されている場合は、MAC ルーチンを再コンパイルすることで、このタイムスタンプのみが更新されます。**[プラン・タイムスタンプ]** はプランの凍結を解除するまで変更されません。その後、**[プラン・タイムスタンプ]** にはプランの凍結を解除した時刻が表示されます。

[文は以下のリレーションを使用します] セクションには、クエリ・プランの作成に使用された 1 つ以上の定義済みテーブルがリストされます。別のテーブルから値を抽出するクエリを使用する INSERT や、別のテーブルを参照する FROM 節を使用する UPDATE または DELETE の場合、ここには両方のテーブルがリストされます。それぞれのテーブルについて、以下の値がリストされます。

- ・ **[テーブル名またはビュー名]:** テーブルまたはビューの修飾名。
- ・ **[タイプ]:** [テーブル] または [ビュー]。
- ・ **[最終コンパイル日時]:** テーブル (永続クラス) が最後にコンパイルされた日時。
- ・ **[クラス名]:** テーブルに関連付けられているクラス名。

このセクションには、クラスを再コンパイルするための **[クラスのコンパイル]** オプションがあります。未凍結のプランを再コンパイルすると、3 つの時刻フィールドのすべてが更新されます。凍結プランを再コンパイルすると、2 つの **[最終コンパイル日時]** フィールドが更新されますが、**[プラン・タイムスタンプ]** は更新されません。プランの凍結を解除して、**[ページの更新]** ボタンをクリックすると、**[プラン・タイムスタンプ]** はプランの凍結を解除した時刻に更新されます。

1.5 SQL 文の問い合わせ

INFORMATION_SCHEMA パッケージのテーブルを使用すると、SQL 文のリストを問い合わせることができます。InterSystems IRIS では、以下のクラスがサポートされます。

- ・ **INFORMATION_SCHEMA.STATEMENTS**: 現在のネームスペース内で現在のユーザがアクセス可能な SQL 文の Index エントリを格納しています。
- ・ **INFORMATION_SCHEMA.STATEMENT_LOCATIONS**: SQL 文が呼び出された各ルーチンの場所を格納しています。永続クラス名またはクエリ・キャッシュ名です。
- ・ **INFORMATION_SCHEMA.STATEMENT_RELATIONS**: SQL 文で使用される各テーブルまたはビューのエントリを格納しています。
- ・ **INFORMATION_SCHEMA.CURRENT_STATEMENTS**: システムのいずれかのネームスペースで現在実行されている SQL 文 Index エントリを格納しています。**%Admin_Operate** リソースへのアクセス権を持つユーザであれば、管理ポータル [\[SQL アクティビティ\]](#) ページで、このテーブルのコンテンツをいつでも調査できます。

これらのクラスを使用するクエリの例を以下に挙げます。

1. SQL

```
SELECT Hash,Frozen,Timestamp,Statement
FROM INFORMATION_SCHEMA.STATEMENTS
```

この例では、ネームスペース内のすべての SQL 文が返され、ハッシュ値 (正規化された SQL 文を一意に識別する計算された ID)、凍結ステータスのフラグ (値 0 から 3)、文が作成されてプランが保存されたときのローカル・タイムスタンプ、および文テキスト自体がリストされます。

2. SQL

```
SELECT Frozen,FrozenDifferent,Timestamp,Statement
FROM INFORMATION_SCHEMA.STATEMENTS
WHERE Frozen=1 OR Frozen=2
```

この例では、すべての凍結プランの SQL 文が返され、凍結しない場合に想定されるプランと凍結したプランが異なるかどうかを示されます。未凍結プランの文では Frozen=0 になる場合と Frozen=3 となる場合があることに注意してください。単一行の INSERT など、凍結できない文の場合、Frozen 列には NULL が表示されます。

3. SQL

```
SELECT Statement,Frozen,
       STATEMENT_LOCATIONS->Location AS Routine,STATEMENT_LOCATIONS->Type AS RoutineType
FROM INFORMATION_SCHEMA.STATEMENTS
WHERE STATEMENT_RELATIONS->Relation='SAMPLE.PERSON'
```

この例では、特定の SQL テーブルについて、すべての SQL 文と、その文が含まれているルーチンを返します (テーブル名 (SAMPLE.PERSON) には、SQL 文テキストで使用されているものと同じ文字種 (すべて大文字) を使用する必要があります)。

4. SQL

```
SELECT Statement,Frozen,Frozen_Different,
       STATEMENT_LOCATIONS->Location AS Routine,STATEMENT_LOCATIONS->Type AS RoutineType
FROM INFORMATION_SCHEMA.STATEMENTS
WHERE Frozen=1 OR Frozen=2
```

この例では、現在のネームスペース内で凍結されたプランを持っているすべての SQL 文を返します。

5. SQL

```
SELECT Statement,Frozen,
       STATEMENT_LOCATIONS->Location AS Routine,STATEMENT_LOCATIONS->Type AS RoutineType
FROM INFORMATION_SCHEMA.STATEMENTS
WHERE Statement [ ' COUNT ( * ) ' ]
```

この例では、現在のネームスペース内で COUNT(*) 集約関数を含んでいるすべての SQL 文を返します(文テキスト (COUNT (*)) には、SQL 文テキストで使用されているものと同じ空白を指定する必要があります)。

1.6 SQL 文のエクスポートとインポート

SQL 文を XML 形式のテキスト・ファイルとしてエクスポートまたはインポートすることができます。これにより、凍結プランを別の場所に移動できます。SQL 文のエクスポートとインポートには、関連付けられているクエリ・プランが含まれます。

1 つの SQL 文をエクスポートすることも、ネームスペースのすべての SQL 文をエクスポートすることもできます。

1 つ以上の SQL 文を含む、以前にエクスポートした XML ファイルをインポートできます。

注釈 XML としての SQL 文のこのインポートをテキスト・ファイルからの [SQL DDL コードのインポートおよび実行](#)と混同しないでください。

1.6.1 SQL 文のエクスポート

1 つの SQL 文をエクスポートする場合は、以下の手順に従います。

- ・ [SQL 文の詳細] ページの **[エクスポート]** ボタンを使用します。管理ポータル [の \[システムエクスプローラ\]](#) SQL インタフェースから、**[SQL 文]** タブを選択し、文をクリックして **[SQL 文の詳細]** ページを開きます。**[エクスポート]** ボタンを選択します。ダイアログ・ボックスが開き、ファイルをサーバ (データ・ファイル) にエクスポートするか、ブラウザにエクスポートするかを選択できます。
 - サーバ (既定値): エクスポート xml ファイルの完全パス名を入力します。初めてエクスポートするとき、このファイルには既定の `statementexport.xml` という名前が付けられます。異なるパスやファイル名を指定することもできます。SQL 文のファイルを正常にエクスポートしたら、最後に使用したファイル名が既定値になります。
[エクスポートをバックグラウンドで実行] チェック・ボックスには、既定ではチェックが付いていません。
 - ブラウザ: ファイル `statementexport.xml` をユーザの既定のブラウザの新しいページにエクスポートします。ブラウザ・エクスポート・ファイルに別の名前を指定したり、異なるソフトウェア表示オプションを指定することもできます。
- ・ `$SYSTEM.SQL.Statement.ExportFrozenPlans()` メソッドを使用します。

ネームスペースのすべての SQL 文をエクスポートする場合は、以下の手順に従います。

- ・ 管理ポータル [の \[すべての文のエクスポート\]](#) [アクション](#)を使用します。管理ポータル [の \[システムエクスプローラ\]](#) SQL インタフェースから、**[アクション]** ドロップダウン・リストを選択します。そのリストから **[すべての文のエクスポート]** を選択します。ダイアログ・ボックスが開き、ネームスペースのすべての SQL 文をサーバ (データ・ファイル) にエクスポートするか、ブラウザにエクスポートするかを選択できます。
 - サーバ (既定値): エクスポート xml ファイルの完全パス名を入力します。初めてエクスポートするとき、このファイルには既定の `statementexport.xml` という名前が付けられます。異なるパスやファイル名を指定することもできます。SQL 文のファイルを正常にエクスポートしたら、最後に使用したファイル名が既定値になります。

[**エクスポートをバックグラウンドで実行**] チェック・ボックスには、既定でチェックが付いています。すべての SQL 文をエクスポートする際には、この設定をお勧めします。[**エクスポートをバックグラウンドで実行する**] にチェックが付いている場合、バックグラウンド・リスト・ページを表示するためのリンクが提供され、このページでバックグラウンド・ジョブのステータスを確認できます。

- ブラウザ:ファイル `statementexport.xml` をユーザの既定のブラウザの新しいページにエクスポートします。ブラウザ・エクスポート・ファイルに別の名前を指定したり、異なるソフトウェア表示オプションを指定することもできます。

- `$SYSTEM.SQL.Statement.ExportAllFrozenPlans()` メソッドを使用します。

1.6.2 SQL 文のインポート

以前にエクスポートしたファイルから 1 つの SQL 文または複数の SQL 文をインポートします。

- 管理ポータル の [文のインポート] [アクション](#) を使用します。管理ポータル の [システムエクスプローラ] SQL インタフェースから、[アクション] ドロップダウン・リストを選択します。そのリストから [文のインポート] を選択します。ダイアログ・ボックスが開き、インポート XML ファイルの完全パス名を指定できます。

[**インポートをバックグラウンドで実行**] チェック・ボックスには、既定でチェックが付いています。SQL 文のファイルをインポートする際には、この設定をお勧めします。[**インポートをバックグラウンドで実行する**] にチェックが付いている場合、バックグラウンド・リスト・ページを表示するためのリンクが提供され、このページでバックグラウンド・ジョブのステータスを確認できます。

- `$SYSTEM.SQL.Statement.ImportFrozenPlans()` メソッドを使用します。

1.6.3 バックグラウンド・タスクの表示と削除

管理ポータル の [システム処理] オプションから、[バックグラウンドタスク] を選択して、エクスポートおよびインポート・バックグラウンド・タスクのログを表示します。[ログを削除] ボタンを使用すると、このログをクリアすることができます。

1.7 SQL 実行時統計情報

SQL 実行時統計情報を使用して、SELECT のクエリ実行時や実行されたコマンドなどの一般的なパフォーマンス統計や、システムで実行されている DDL および DML 文を監視できます。クエリ操作を準備すると、SQL 実行時統計情報が収集されます。これらは、`INFORMATION_SCHEMA.STATEMENTS`、`STATEMENT_DAILY_STATS`、`STATEMENT_HOURLY_STATS` テーブルに表示できます。また、[パラメータ収集](#) が有効化されている場合、実行時パラメータに関して収集された統計情報を `INFORMATION_SCHEMA.STATEMENT_PARAMETER_STATS` テーブルに表示できます。

SQL 実行時統計情報の収集は常にオンになっており、オフにすることはできません。統計情報の収集をできる限り効率化するために、統計情報は設定された間隔でのみ書き込まれます。そのため、収集された実行時統計情報を表示するのに最大 30 分かかることがあります。現在実行中の文を表示するには、`INFORMATION_SCHEMA.CURRENT_STATEMENTS` テーブルに対してクエリを実行します。

実行時統計情報には以下の情報があります。

- [平均時間]: クエリの実行に要した時間の平均値 (秒)
- [実行回数]: クエリが実行された回数を示す値
- [平均行]: 返された行の平均数
- [平均コマンド]: 実行されたコマンドの平均数

SQL 実行時統計情報を**明示的に削除** (クリア) できます。クエリ・キャッシュを削除すると、関連するすべての SQL 実行時統計情報が削除されます。テーブルまたはビューを削除すると、関連するすべての SQL 実行時統計情報が削除されます。

注釈 プロセス固有の SQL クエリ統計をグローバル統計に集計するためのシステム・タスクが、すべてのネームスペースで 1 時間に 1 回、自動的に実行されます。したがって、グローバル統計には、1 時間以内に収集された統計が反映されていないことがあります。

1.7.1 実行時統計とプラン表示

[SQL 実行時統計] ツールは、実行時統計を含むクエリの**プラン表示**を表示するために使用できます。

[別のプランを表示] ツールは、クエリの実行時統計を表示して、プラン表示を統計と比較するために使用できます。[プラン表示のオプション] の [別のプランを表示] ツールには、クエリの推定統計が表示されます。実行時統計の収集がアクティブにされていると、[プラン表示を統計と比較] オプションにより、実際の実行時統計が表示されます。実行時統計がアクティブでない場合、このオプションにより推定統計が表示されます。

2

SQL クエリ・プランの解釈

この章では、システムで生成された InterSystems SQL クエリ・プランを読み解く方法について説明します。これらのプランの確認で利用できる各種ツールを紹介し、その表示内容を解釈する方法も取り上げます。

SQL クエリをコンパイルすると、そのクエリで指定したデータにアクセスする命令とそのデータを返す命令が生成されます。それらの命令およびその命令の実行シーケンスは、そのクエリに含まれるテーブルの構造や内容に関する SQL コンパイラの所有データの影響を受けます。コンパイラは、テーブル・サイズや使用可能なインデックスなどの情報を使用して、その一連の命令をできる限り効率化しようとします。

クエリ・アクセス・プラン ([プラン表示](#)) は、その命令の結果セットを人が読める形式に変換したものです。クエリ的设计者はこのクエリ・アクセス・プランを使用して、データへのアクセスがどのように行われるかを確認することができます。SQL コンパイラは、クエリによる指定に従って、データを最も効率的に使用できるようにしようとします。格納されているデータの特定の側面については、コンパイラよりもユーザの方が詳しいことがあります。その場合は、クエリ・プランを利用して、クエリのコンパイラにより多くの情報やガイダンスを提供できるように元のクエリを変更できます。

2.1 プランの表示

EXPLAIN ツールまたは [プラン表示] ツールを使用して、SELECT、DECLARE、UPDATE、DELETE、TRUNCATE TABLE の各操作と、一部の INSERT 操作の実行プランを表示できます。これらは、実行の一環として SELECT を使用するため、総称してクエリ操作と呼びます。InterSystems IRIS では、実行プランは、クエリ操作が準備されたときに生成されます。実行プランを生成するために、実際にクエリを実行する必要はありません。

これらのツールには既定で、InterSystems IRIS が最適であると見なしたクエリ・プランの内容が表示されます。ほとんどのクエリには、有効なクエリ・プランが複数存在します。InterSystems IRIS が最適であると見なしたクエリ・プランに加えて、[代替クエリ実行プラン](#)を生成および表示することもできます。

InterSystems IRIS には、以下のクエリ・プラン・ツールが用意されています。

- ・ [SQL EXPLAIN コマンド](#)。このコマンドを使用すると、XML 形式のクエリ・プランと、オプションで代替クエリ・プランおよび SQL 統計を生成・表示できます。生成されたすべてのクエリ・プランと統計は、Plan という名前の単一の結果セット・フィールドに含まれます。EXPLAIN コマンドは SELECT クエリでのみ使用できることに注意してください。
- ・ 管理ポータル→[\[システムエクスプローラ\]](#)→SQL インタフェースの [\[プラン表示\]](#) ボタン。
- ・ 管理ポータル→[\[システムエクスプローラ\]](#)→[\[ツール\]](#)→[\[SQL パフォーマンス・ツール\]](#)。
- ・ `$$SYSTEM.SQL.Explain()` メソッド。このメソッドを使用すると、XML 形式のクエリ・プランと、オプションで代替クエリ・プランを生成・表示できます。
- ・ [SQL シェル](#)でシェル・コマンドの SHOW PLAN と SHOW PLANALT を使用して、最近実行したクエリの実行プランを表示できます。

生成された %PARALLEL クエリおよびシャード・クエリの場合、これらのツールでは、該当するクエリ・プランがすべて表示されます。

2.1.1 EXPLAIN の使用法

以下の例のように EXPLAIN コマンドを実行することでクエリ実行プランを生成できます。

SQL

```
EXPLAIN SELECT TOP 10 Name,DOB FROM Sample.Person
```

EXPLAIN コマンドを使用するには、%Development:USE リソースを持っている必要があります。

ALT キーワードを使用すると、EXPLAIN コマンドによって代替のクエリ・プランが生成され、返されるクエリ・プランに反映されます。

STAT キーワードを使用すると、EXPLAIN コマンドによってクエリのモジュールごとにパフォーマンス統計が生成されます。モジュールごとに以下の統計が返されます。

- ・ <ModuleName> : モジュール名。
- ・ <TimeSpent> : モジュールの総実行時間 (秒単位)。
- ・ <GlobalRefs> : グローバル参照の数。
- ・ <LinesOfCode> : コード実行行数。
- ・ <DiskWait> : ディスク待機時間 (秒単位)。
- ・ <RowCount> : 結果セット内の行数。
- ・ <ModuleCount> : このモジュールが実行された回数。
- ・ <Counter> : このプログラムが実行された回数。

クエリ・プランは XML 形式の文字列で返されます。その最上位のタグは <plans> です。ALT キーワードを指定しない場合、<plans> タグの下位には <plan> タグが 1 つだけ存在します。そのタグの下位には、クエリを指定した <sql> タグ、この特定のプランの相対コストを記述した <cost> タグ、および SQL オプティマイザでクエリがどのように処理されたかの説明があります。ALT キーワードを指定した場合は、<plans> タグの下位に複数の <plan> タグが作成されます。STAT キーワードを指定した場合は、<cost> タグの下位に複数の <stats> タグが存在します。クエリの処理に関連するモジュールごとに別々の <stats> タグが生成されます。

2.1.2 管理ポータルでのプラン表示の使用法

[プラン表示] を使用すると、以下のどの方法でもクエリの実行プランを表示できます。

- ・ 管理ポータルの [SQL] インタフェースで [システムエクスプローラ]、[SQL] の順に選択します。ページ上部に表示されている現在のネームスペースの名前をクリックして、ネームスペースを選択します。(ユーザごとに管理ポータルの既定ネームスペースを設定できます)。クエリを作成し、[プラン表示] ボタンを押します(一覧表示されているクエリのプラン・オプションをクリックして、[履歴を表示] から [プラン表示] を呼び出すこともできます)。このドキュメントの“管理ポータルの SQL インタフェースの使用法”の章の“SQL 文の実行”を参照してください。
- ・ 管理ポータルの [ツール] インタフェースで、[システムエクスプローラ]、[ツール]、[SQL パフォーマンス・ツール]、[SQL 実行時統計情報] の順に選択します
 - － [クエリのテスト] タブから、ページ上部に表示されている現在のネームスペースの名前をクリックして、ネームスペースを選択します。テキスト・ボックスにクエリを入力します。次に、[SQL 統計によるプラン表示] ボタンを押します。これにより、クエリを実行することなく [プラン表示] を生成します。

- [統計情報表示] タブで、リストされているいずれかのクエリの [プラン表示] ボタンを押します。リストされるクエリには、[クエリ実行] で記述されたクエリと、[クエリ・テスト] で記述されたクエリの両方が含まれます。

既定では、[プラン表示] は論理モードで値を返します。ただし、管理ポータルまたは SQL Shell から [プラン表示] を呼び出すと、[プラン表示] は実行時モードを使用します。

2.2 プランの構造

[プラン表示] の実行プランは、**文テキスト**と**クエリ・プラン**の 2 つのコンポーネントで構成されています。

[ステートメント・テキスト] には元のクエリが複製されますが、変更点として、管理ポータルの [SQL] インタフェースにある [プラン表示] ボタンを使用すると、コメントと改行が削除された SQL 文が表示されます。空白は標準化されます。[プラン表示] ボタンは、各リテラルを ? で置換するリテラル置換の実行も表示します。ただし、二重の括弧でリテラル値を囲むことで、リテラル置換を抑制している場合を除きます。Explain() メソッドを使用してプラン表示を表示したときや、[SQL 実行時統計情報] ツールまたは [別のプランを表示] ツールを使用して表示したときには、このような変更は行われません。

[クエリプラン] には、クエリの実行に使用されるプランが示されます。クエリ・プランの内容は以下のとおりです。

- ・ [Frozen Plan凍結プラン] は、クエリ・プランが凍結されている場合に、[クエリプラン] の先頭行に表示されます。それ以外の場合、先頭行は空白になります。
- ・ [相対コスト] は、同じクエリの異なる実行プランの効率を比較するための抽象的な数値として、多数の要因から計算される整数値です。この計算では、さまざまな要因の中でも特にクエリの複雑さ、インデックスの有無、およびテーブルのサイズが考慮されます。相対コストは、2 つの異なるクエリの比較には有用ではありません。WHERE 節を指定していない COUNT(*) または MAX(%ID) などの特定の集約クエリによって “Relative cost not available” が返されます。
- ・ [クエリプラン] は、メイン・モジュールと 1 つ以上のサブコンポーネント (必要な場合) で構成されます。1 つ以上のモジュール・サブコンポーネントが表示される場合は、B から始まるアルファベット順の名前が付けられ (Module:B、Module:C など)、実行順にリストされます (アルファベット順とは限りません)。

名前が付いたサブクエリ・モジュールは、クエリ内のサブクエリごとに表示されます。サブクエリ・モジュールには、アルファベット順に名前が付けられます。サブクエリの命名時には、1 つ以上の文字をスキップして各サブクエリに名前が付けられます。アルファベットの最後に達すると、追加のサブクエリには、Z=26 と解析し、同じスキップ・シーケンスを使用して番号が付けられます。例えば、Subquery:F から始まる 2 つおきのサブクエリ命名シーケンスは、F、I、L、O、R、U、X、27、30、33 となります。また、Subquery:G から始まる 1 つおきのサブクエリ命名シーケンスは、G、I、K、M、O、Q、S、U、W、Y、27、29 となります。サブクエリがモジュールを呼び出す場合、そのモジュールは、スキップなしでサブクエリの後にアルファベット順に配置されます。例えば、Subquery:H を指定すると Module:I が呼び出されます。

- ・ メイン・モジュールの最初の箇条項目が “Read master map” である場合、非効率的なクエリ・プランであることを示しています。クエリ・プランは、Read master map... (使用可能なインデックスがない)、Read index map... (使用可能なインデックスを使用)、または Generate a stream of idkey values using the multi-index combination... (マルチ・インデックス、複数インデックスを使用) のいずれかのマップ・タイプの文で実行を開始します。マスタ・マップは、データのインデックスを読み取るのではなく、データ自体を読み取るため、ほとんどの場合、Read master map... は非効率的なクエリ・プランであることを示しています。テーブルが比較的小さなものでない限り、クエリ・プランを再生成したときに最初のマップに Read index map... と表示されるように、**インデックスを定義**する必要があります。クエリ・プランの解釈の詳細は、“SQL 実行プランの解釈” を参照してください。

一部の操作では、クエリ・プランが生成されない可能性のあることを示す [プラン表示] が作成されます。

- ・ 非クエリの INSERT:INSERT... VALUES() コマンドはクエリを実行しません。そのため、クエリ・プランは生成されません。

- ・ 常に FALSE のクエリ: InterSystems IRIS は、クエリ条件が常に false になるためにデータを返せないクエリを準備している状況を判断できます。この状況は、[プラン表示] の [クエリプラン] コンポーネントで通知されます。例えば、条件 WHERE %ID IS NULL または条件 WHERE Name %STARTSWITH('A') AND Name IS NULL を含むクエリはデータを返せないため、InterSystems IRIS は実行プランを生成しません。実行プランを生成する代わりに、[クエリプラン] は “Output no rows” というメッセージを示します。クエリに、これらの条件のいずれかを指定したサブクエリが含まれている場合、そのサブクエリ・モジュールについての [クエリプラン] には “Subquery result NULL, found no rows” というメッセージが表示されます。この条件チェックは、NULL が関与するいくつかの状況に制限されていて、自己矛盾するクエリ条件のすべてを補足することを意図していません。
- ・ 不正なクエリ: [プラン表示] には、ほとんどの不正なクエリに対して SQLCODE エラー・メッセージを表示します。ただし、[プラン表示] は空として表示することもあります。例えば、WHERE Name = \$\$\$\$ や WHERE Name %STARTSWITH('A') (単一引用符と二重引用符に注目してください) などが挙げられます。このような場合、[プラン表示] には [文テキスト] が表示されません。また、[クエリプラン] には [No plan created for this statement] というメッセージが表示されます。一般に、これはリテラルを区切っている引用符が釣り合っていない場合に発生します。また、[ユーザ定義 \(“外部”\) 関数](#)に正しい構文を指定しないで、複数のドル記号を先頭に指定した場合にも発生します。

2.3 プランの読み取り

プラン表示の結果は、クエリで指定したデータにアクセスして提示するためにどのような処理が実行されるかを示す一連の文です。以下では、プラン表示の文の解釈方法について説明します。

2.3.1 マップへのアクセス

SQL テーブルは、一連の [マップ](#) として格納されます。各テーブルには、テーブル内のすべてのデータが含まれた [マスタ・マップ](#) と、場合によっては、インデックス・マップやビットマップなどのその他のマップがあります。各マップは [多次元グローバル](#) として示すことができます。これには、1 つ以上の [フィールド](#) の一部のフィールド用のデータと、ノード値に格納された残りのフィールドがあります。添え字はアクセスされるデータを制御します。

- ・ [マスタ・マップ](#) では、通常、RowID フィールドまたは IDKEY フィールドがマップの添え字として使用されます。
- ・ インデックス・マップでは、通常、他のフィールドが先頭の添え字として使用され、RowID フィールドや IDKEY フィールドは追加の下位レベルの添え字として使用されます。
- ・ ビットマップでは、ビットマップ・レイヤを追加の RowID の添え字レベルと見なすことができます。ただし、ビットマップは、RowID が正の整数である場合しか使用できません。

クエリのプランはいくつかのテーブルにアクセスする可能性があります。テーブルにアクセスする際、プランは 1 つのマップ (インデックスまたはマスタ・マップ) にアクセスすることも、2 つのマップ (インデックス・マップとそれに続くマスタ・マップ)、あるいは、[マップ](#) の場合はいくつかのマップにアクセスすることがあります。

マップを使用したデータへのアクセスでは、プランは使用される添え字を示します。また、実際の添え字値の内容 (添え字に対してテーブルに存在する単一の特定値、一連の特定値、値の範囲、またはすべての値のいずれか) も示します。どれが選択されるかは、クエリで指定されている条件によって異なります。明らかに、アクセスする添え字値が単一もしくは少数である方が、その添え字レベルのすべての値にアクセスするよりも速く処理されます。

2.3.2 条件および式

クエリの実行時には、クエリで指定されているさまざまな条件がテストされます。前述のような添え字による制限などの特定の条件を除き、プラン表示では、その出力として条件のテストが明示的に示されることはありません。できるだけ早期に条件をテストすることをお勧めします。多様な条件をテストするために最適な場所は、プランの詳細から推測できます。

同様に、プラン表示では式および部分式の計算の詳細は示されません。簡潔性は別にして、その主な理由はほとんどのデータベース環境で、テーブルおよびインデックスのアクセスが処理にとってより重要な側面となるためです。ディスク・アクセス速度が依然 CPU 処理よりも桁違いに遅いので、テーブル・データの取得にかかる負荷がクエリ全体の負荷の大半を占めています。

2.3.3 ループ

テーブルのデータにアクセスする際は、複数の行を繰り返し検証する必要性が多くなります。そのようなアクセスは、によって示されます。受け渡しごとに実行される命令は、ループの と呼ばれます。それらはインデントされることによって視覚的に示されます。複数のテーブルが関係するデータベース・アクセスでは、ループ内にループが必要となることが一般的です。この場合、各ループ・レベルは前のレベルからさらにインデントされることによって示されます。

2.3.4 一時ファイル

2.3.4.1 定義

クエリ・プランは、中間一時ファイル()を作成して使用する必要性を示す場合もあります。これは、ローカル配列内の“スクラッチ”領域となります。これは並べ替えなどのさまざまな目的で一時的な結果を保存するのに使用されます。マップと同様に、一時ファイルは 1 つ以上の添え字 (場合によっては、ノード・データも) を使用します。

2.3.4.2 使用

一時ファイルには、単一テーブルの処理からのデータが含まれる場合があります。この場合、一時ファイルの作成は、そのテーブル内のデータの と見なされることもあります。そのような一時ファイルの読み取りの後には、ソース・テーブルのマスタ・マップへのアクセスが行われる場合とそうでない場合があります。これ以外の場合、一時ファイルには複数のテーブルの処理の結果が含まれる可能性があります。さらに別の状況では、一時ファイルは分類された集約値の格納や DISTINCT のチェックなどに使用されます。

2.3.5 モジュール

一時ファイルの作成もその他の処理と同じく、 と呼ばれる独立した作業ユニットで行われる場合があります。各モジュールには名前が付けられます。別々のモジュールがリストされている場合、プランは、各モジュールの呼び出し元を示しています。モジュールの実行が終了すると、処理がモジュールの呼び出しの直後の文から再開されます。

2.3.6 処理のために送信されるクエリ

ODBC または JDBC ゲートウェイ接続経由でリンクされた外部テーブルの場合、プランはリモートの SQL ゲートウェイ接続に送信中のクエリのテキストを示し、リモート・テーブルから要求データを取得します。

クエリの並列処理とシャードリングの場合、プランは、並列でまたはシャードで処理するために送信される各種クエリを示します。これらの各クエリに使用されるプランも表示されます。

2.3.7 サブクエリ、JOIN、および UNION

特定のクエリ内のサブクエリ (およびビュー) にも、個別に処理されるものがあります。そのプランは、独立した ・セクションで示されます。サブクエリ・セクションの呼び出し元の正確な場所は、プランには示されません。それは、これらが条件または式の処理の一部として呼び出されることが多いためです。

OUTER JOIN を指定するクエリでは、外部結合の意味の要件を満たすために一致する行が見つからない場合に、プランでは NULL の行が生成される可能性が示される場合があります。

UNION の場合は、そのプランに、別個のモジュールにおけるさまざまな UNION のサブクエリからの結果行の組み合わせが示される場合があります (このモジュールでは、それらの結果行のさらなる処理が行われることがあります)。

2.4 別のプランを表示

管理ポータルまたは Explain() メソッドを使用して、クエリの代替実行プランを表示できます。

管理ポータルからクエリの代替実行プランを表示するには、以下のいずれかを使用します。

- ・ [システムエクスプローラ]、[ツール]、[SQL パフォーマンス・ツール]、[代替表示プラン] の順に選択します。
- ・ [システムエクスプローラ]、[SQL] の順に選択し、[ツール] ドロップダウン・メニューから [代替表示プラン] を選択します。

[代替表示プラン] ツールを使用して、以下の手順を実行します。

1. SQL クエリ・テキストを入力するか、[履歴を表示] ボタンを使用して SQL クエリ・テキストを取得します。右側にある [X] の円をクリックすると、クエリ・テキスト・フィールドをクリアできます。
2. [プラン表示のオプション] ボタンを押して、複数の代替表示プランを表示します。[...をバックグラウンドで実行] チェック・ボックスは、既定ではチェックが外されています。ほとんどのクエリでは、この設定をお勧めします。大規模なクエリや複雑なクエリについては、[...をバックグラウンドで実行] チェック・ボックスにチェックを付けることをお勧めします。長いクエリがバックグラウンドで実行されている間、[プロセス表示] ボタンが表示されます。[プロセス表示] をクリックすると、[プロセス詳細] ページが新しいタブで開きます。[プロセス詳細] ページから、プロセスを参照できます。また、プロセスを一時停止、再開、または終了することもできます。
3. プランの候補は、マップ・タイプおよび開始マップと共に、コストに基づいて昇順でリストされます。それぞれのプランについて [プラン表示] (統計なし) または [統計付きプランを表示] のリンクを選択すると、詳細を表示できます。
4. プランの候補のリストから、チェック・ボックスを使用して、比較するプランを選択した後、[プラン表示を統計と比較] ボタンを押してそれらを実行し、それぞれの SQL 統計を表示します。

Explain() メソッドと共に all 修飾子を使用すると、1 つのクエリの実行プランがすべて表示されます。InterSystems IRIS が最適 (最小コスト) であると見なしたプランが最初に表示され、続いて代替プランが表示されます。代替プランは、コストの昇順でリストされます。

以下の例は、最適な実行プランを表示してから、代替クエリをリストします。

ObjectScript

```
DO $SYSTEM.SQL.SetSQLStatsFlagJob(3)
SET mysql=1
SET mysql(1)="SELECT TOP 4 Name,DOB FROM Sample.Person ORDER BY Age"
DO $SYSTEM.SQL.Explain(.mysql,{"all":1},,.plan)
ZWRITE plan
```

%SYS.PTools.StatsSQL クラス内の possiblePlans メソッドも参照してください。

2.4.1 統計

[プラン表示オプション] リストでは、各代替表示プランに [コスト] 値を割り当てます。この値により、実行プラン間での相対的な比較が可能になります。

[代替表示プラン] の詳細には、[クエリプラン] ごとに [クエリの合計] についての統計 (Stats) のセットが示されます。(該当する場合は) クエリ・プラン・モジュールごとにも、これが示されます。各モジュールの統計には、時間 (全体的なパフォーマンス (秒単位))、グローバル参照 (グローバル参照の数)、コマンド (実行された行の数)、および読み取り待ち時間 (ディスク待機時間 (ミリ秒単位)) が含まれます。[クエリの合計] の統計には、返された行の数も含まれます。

3

SQL パフォーマンス分析ツールキット

InterSystems IRIS は、特定の SQL 文の積極的なプロファイリングに使用できる分析ツールを用意しています。このツールは、そのような SQL 文の実行に関する詳細な情報を収集するので、クエリ・プランの中で特定の問題を特定するうえで効果的です。収集した情報を使用して、開発者は非効率な SQL 文のパフォーマンスを改善するための対策を講じることができます。ただし、積極的なプロファイリングによって、サーバに対する負荷が著しく増加することがあります。したがって、SQL パフォーマンス分析ツールキットでは、[SQL 実行時統計情報](#)の調査を経て詳しい検討を必要とするクエリを特定した後で、協調性のあるコード分析を進めることを目的としています。実行コードを継続的に監視するためのものではありません。

注釈 `%SYSTEM.SQL.PTools` クラス・メソッドは、この機能呼び出すための推奨 API です。`%SYSTEM.SQL.PTools` メソッド・インタフェースは、`%SYS.PTools` ベース・クラスのメソッドに実装された機能を再グループ化および再編成します。

3.1 SQL パフォーマンス統計の有効化

サポート・スペシャリストは、SQL パフォーマンス分析ツールキットを使用して、特定の SQL 文または SQL 文のグループをプロファイリングできます。特定の SQL 文の実行時にこれらのツールを使用してパフォーマンス統計を収集し、その情報を使用して問題のある文を切り分けて分析したり、実際の作業負荷で分析したりすることができます。

`%SYSTEM.SQL.PTools` クラスのメソッドを使用すると、高度なパフォーマンス統計の収集を開始できます。このような統計をさまざまな規模で収集するために以下のメソッドが用意されています。

- ・ システム全体 : `setSQLStatsFlag()`
- ・ 特定のネームスペース : `setSQLStatsFlagByNS()`
- ・ 現在のプロセスまたはジョブ : `setSQLStatsFlagJob()`
- ・ 指定したプロセスまたはジョブ : `setSQLStatsFlagByPID()`。最初のパラメータが未指定の場合、あるいは `$JOB` または空の文字列 ("") として指定されている場合、`setSQLStatsFlagJob()` が呼び出されます。したがって、`SET SQLStatsFlag=$SYSTEM.SQL.SetSQLStatsFlagByPID($JOB,3)` は `SET SQLStatsFlag=$SYSTEM.SQL.SetSQLStatsFlagJob(3)` と同等です。

これらのメソッドは、整数のアクション・オプションを取ります。返される情報は、コロンで区切られた文字列です。その最初の要素は、優先するアクション・オプションです。現在の設定を確認するには、`getSQLStatsFlag()` メソッドまたは `getSQLStatsFlagByPID()` メソッドを使用します。

これらのメソッドは、以下の例に示すように、ObjectScript から呼び出すことも、SQL から呼び出すこともできます。

- ・ ObjectScript から : `SET rtn=##class(%SYSTEM.SQL.PTools).setSQLStatsFlag(2,,8)`

- SQL から : `SELECT %SYSTEM_SQL.PTools_setSQLStatsFlag(2,,8)`

また、`SELECT` 文、`INSERT` 文、`UPDATE` 文、または `DELETE` 文で `%PROFILE` キーワード (`setSQLStatsFlagJob(2)` と同じ) または `%PROFILE_ALL` キーワード (`setSQLStatsFlagJob(3)` と同じ) を使用すると、その文のみのパフォーマンス分析統計を収集できます。

3.1.1 アクション・オプション

`setSQLStatsFlag()` と `setSQLStatsFlagByNS()` については、以下のいずれかのアクション・オプションを指定します。0 に設定すると、統計コードの生成が無効になります。1 に設定すると、すべてのクエリに対して統計コードの生成が有効になりますが、統計は収集されません (既定値)。2 に設定すると、クエリの外側のループのみの統計が記録されます (クエリのオープンおよびクローズ時に統計が収集されます)。3 に設定すると、クエリのすべてのモジュール・レベルに対して統計が記録されます。モジュールは入れ子にすることができます。その場合、MAIN モジュールの統計が包含的な数となり、全クエリに対する全体的な結果となります。

`setSQLStatsFlagJob()` と `setSQLStatsFlagByPID()` については、アクション・オプションが多少異なります。-1 を指定すると、このジョブの統計が無効になります。0 を指定すると、システム設定値が使用されます。1、2、および 3 のオプションは `setSQLStatsFlag()` と同様であり、システム設定がオーバーライドされます。既定値は 0 です。

SQL パフォーマンス統計を収集するには、統計コードの生成を有効にして (オプション 1、既定値) クエリをコンパイル (準備) する必要があります。

- 0 から 1 に変更する場合: アクション・オプションの変更後、SQL を含んでいるルーチンとクラスは統計コードの生成を実行するためにコンパイルする必要があります。ダイナミック SQL またはデータベース・ドライバを使用する場合は、コードの生成を強制するために、クエリ・キャッシュを削除する必要があります。
- 1 から 2 に変更する場合: アクション・オプションを変更するだけで、統計の収集が開始されます。これにより、最短の中断で稼働中のプロダクション環境に関する SQL パフォーマンスの分析を実行できるようになります。
- 1 から 3 (または 2 から 3) に変更する場合: アクション・オプションの変更後、SQL を含んでいるルーチンとクラスは、すべてのモジュール・レベルの統計を記録するためにコンパイルする必要があります。ダイナミック SQL またはデータベース・ドライバを使用する場合は、コードの生成を強制するために、クエリ・キャッシュを削除する必要があります。一般に、オプション 3 は、非プロダクション環境で特定したパフォーマンスの低いクエリにのみ使用します。
- 1, 2, または 3 から 0 に変更する場合: 統計コードの生成をオフにするために、クエリ・キャッシュを削除する必要はありません。

3.1.2 収集オプション

アクション・オプションが 2 または 3 である場合、これらのいずれかのメソッドを呼び出す際に、収集オプション値を指定して、収集するパフォーマンス統計を指定することができます。既定では、すべての統計が収集されます。

収集オプションを指定するには、収集するそれぞれの統計タイプに関連付けられた整数値を合計します。既定値は 15 (1 + 2 + 4 + 8) です。

これらのメソッドは、コロンで区切られた 2 つ目の要素としてこの収集オプションの以前の値を返します。現在の設定を確認するには、`getSQLStatsFlag()` メソッドまたは `getSQLStatsFlagByPID()` メソッドを使用します。既定では、すべての統計が収集され、2 つ目の要素の値として 15 が返されます。

詳細は、“%SYSTEM.SQL.PTools” を参照してください。

3.1.3 終了オプション

統計の収集は、終了するまで続行されます。既定では、別の `setSQLStatsFlag[nnn]()` メソッドを発行して終了するまで、収集が無期限に続行されます。また、アクション・オプションが 1、2、または 3 である場合は、`setSQLStatsFlag[nnn]()` 終了オプション (経過期間 (分単位) またはタイムスタンプ) を指定することもできます。続いて、その期間が経過したときに

リセットするアクション・オプションを指定します。例えば、"M:120:1" という文字列では、M (経過分数) が 120 分に設定され、その期間の終了時にアクション・オプションが 1 にリセットされます。他のオプションはすべて、そのアクション・オプションに応じた既定値にリセットされます。

これらのメソッドは、コロンで区切られた 5 つ目の要素としてこの終了オプションの以前の値をエンコードされた値で返します。"統計設定の取得" を参照してください。

3.1.4 管理ポータルでのパフォーマンス統計の有効化

以下のいずれかのパスを使用して、[SQL 実行時統計情報] ページの [設定] タブで管理ポータルのパフォーマンス統計を収集するためのアクション・オプションを設定できます。

- ・ [システムエクスプローラ]→[ツール]→[SQL パフォーマンス・ツール]→[SQL 実行時統計情報]
- ・ [システムエクスプローラ]→[SQL]→[ツール]→[SQL 実行時統計情報]

3.2 統計設定の取得

setSQLStatsFlag[nnn]() メソッドは、以前の統計設定をコロンで区切られた値として返します。現在の設定を確認するには、getSQLStatsFlag() メソッドまたは getSQLStatsFlagByPID() メソッドを使用します。

コロンで区切られた 1 つ目の値は、アクション・オプションの設定です。コロンで区切られた 2 つ目の値は、収集オプションです。コロンで区切られた 3 つ目の値と 4 つ目の値は、ネームスペース固有の統計の収集に使用されます。コロンで区切られた 5 つ目の値は、終了オプションをエンコードしたものです。コロンで区切られた 6 つ目の値では **FlagType** を指定します。0 = システム・フラグ、1 = プロセス/ジョブ・フラグです。

以下の例に示すように、ptInfo 配列を使用して、終了オプションの設定をより詳細に表示できます。

ObjectScript

```
KILL
DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL("USER")
DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagByNS("USER",3,,7,"M:5:1")
DisplaySettings
SET SQLStatsFlag = ##class(%SYSTEM.SQL.PTools).getSQLStatsFlag(0,.ptInfo)
WRITE "ptInfo array of performance statistics return value:",!
ZWRITE ptInfo,SQLStatsFlag
```

3.3 クエリ・パフォーマンス統計のエクスポート

%SYSTEM.SQL.PTools の exportSQLStats() メソッドを使用して、クエリ・パフォーマンス統計をファイルにエクスポートできます。このメソッドは、%SYSTEM.SQL.PTools クラスから統計データをファイルにエクスポートする場合に使用します。

exportSQLStats() は、以下の例に示すように呼び出すことができます。

- ・ ObjectScript から : SET status=##class(%SYSTEM.SQL.PTools).exportSQLStats("\$IO") (既定値は format T です)。
- ・ SQL から : CALL %SYSTEM_SQL.PTools_exportSQLStats('\$IO') (既定値は format H です)。

filename 引数を指定しなかった場合、このメソッドは現在のディレクトリにエクスポートします。既定では、このファイルの名前は、PT_StatsSQL_exportSQLStats_ の後に現在のローカル日時が YYYYMMDD_HHMMSS として付加されたものになります。**\$IO** を指定すると、データをターミナルまたは管理ポータルのディスプレイに出力できます。filename

引数を指定した場合、このメソッドは、現在のネームスペースの Mgr サブディレクトリ、または指定したパスの場所にファイルを作成します。このエクスポートは、現在のネームスペース内のデータに限られます。

出力ファイルの format を P (テキスト)、D (コンマ区切りのデータ)、X (XML マークアップ)、H (HTML マークアップ)、または Z (ユーザ定義の区切り文字) として指定できます。

既定では、このメソッドはクエリ・パフォーマンス統計をエクスポートします。以下の例に示すように、代わりに SQL クエリ・テキストまたは SQL [クエリ・プラン](#) データをエクスポートするように指定することもできます。

- ・ クエリ・テキスト : `CALL %SYSTEM_SQL.PTools_exportSQLStats('$IO',,0,1,0)`
- ・ クエリ・プラン : `CALL %SYSTEM_SQL.PTools_exportSQLStats('$IO',,0,1,1)`

`exportSQLStats()` は、コメントを削除し、リテラル置換を実行することにより、クエリ・テキストを変更します。

`ExportSQLQuery()` によって、同じクエリ・テキストおよびクエリ・プラン・データを返すことができます。

3.3.1 統計値

以下の統計が返されます。

- ・ RowCount - 指定されたクエリの MAIN モジュールで返された行の合計数。
- ・ RunCount - クエリが最後にコンパイルまたは準備された後で実行された合計回数。
- ・ ModuleCount - クエリの実行中に、指定されたモジュールに入った合計回数。
- ・ TimeToFirstRow - 指定されたクエリの MAIN モジュールに最初の結果セット行を返すまでに要した合計時間。
- ・ TimeSpent - 指定されたクエリの指定されたモジュールで費やされた合計時間。
- ・ GlobalRefs - 指定されたクエリの指定されたモジュールで行われたグローバル参照の合計数。
- ・ LinesOfCode - 指定されたクエリの指定されたモジュールで実行された ObjectScript コードの合計行数。
- ・ DiskWait (ディスク待ち時間とも呼ばれる) - 指定されたクエリの指定されたモジュールでディスク読み取りの待機に要した合計時間 (ミリ秒単位)。

SQL 統計では、実行されているクエリのすべてのコンポーネントについての集約カウンタ値を報告します。このようなコンポーネントには、並列クエリのすべてのパーティションや、シャード・クエリのすべてのシャードが含まれます。

3.4 クエリ・パフォーマンス統計の削除

`clearSQLStatsALL()` メソッドを使用すると、パフォーマンス統計を削除できます。既定では、現在のネームスペースのすべてのルーチンに対して収集された統計が削除されます。異なるネームスペースの指定、特定のルーチンへの削除操作の限定、またはその両方を指定できます。

また、管理ポータルの **[SQL 実行時統計情報]** ページで **[統計情報を削除]** ボタンを使用して、現在のネームスペースのすべてのクエリについて蓄積されたすべての統計情報を削除できます。成功した場合、メッセージには削除された統計の数が示されます。統計がなかった場合は、`"` というメッセージが表示されます。正常に削除できなかった場合は、エラー・メッセージが表示されます。

3.5 パフォーマンス統計の例

以下の例では、現在のプロセスによって準備されたクエリのメイン・モジュールに関するパフォーマンス統計を収集し（アクション・オプション 2）、その後、`exportSQLStats()` を使用してパフォーマンス統計をターミナルに表示します。

ObjectScript

```
DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL()
DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagJob(2)
SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
SET tStatement = ##class(%SQL.Statement).%New()
SET qStatus = tStatement.%Prepare(myquery)
IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
SET pStatus = ##class(%SYSTEM.SQL.PTools).exportSQLStats("$IO")
IF pStatus'=1 {WRITE "Performance stats display failed:"
DO $System.Status.DisplayError(qStatus) QUIT}
```

以下の例では、現在のプロセスによって準備されたクエリのすべてのモジュールに関するパフォーマンス統計を収集し（アクション・オプション 3）、その後、埋め込み SQL から `exportSQLStats()` を呼び出してパフォーマンス統計をターミナルに表示します。

ObjectScript

```
DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL()
DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagJob(3)
SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
SET tStatement = ##class(%SQL.Statement).%New()
SET qStatus = tStatement.%Prepare(myquery)
IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
&sql(CALL %SYSTEM_SQL.PTools_exportsSQLStats('$IO'))
```

以下の例では、現在のプロセスによって準備されたクエリのメイン・モジュールに関するパフォーマンス統計を収集し（アクション・オプション 2）、その後、`StatsSQLView` クエリを使用してこれらの統計を表示します。

ObjectScript

```
DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL()
DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagJob(2)
SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
SET tStatement = ##class(%SQL.Statement).%New()
SET qStatus = tStatement.%Prepare(myquery)
IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
GetStats
SET qStatus = tStatement.%Prepare("SELECT * FROM %SYS_PTools.StatsSQLView")
IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
SET rsstats = tStatement.%Execute()
DO rsstats.%Display()
WRITE !!, "End of SQL Statistics"
```

以下の例では、USER ネームスペースのすべてのクエリのすべてのモジュールに関するパフォーマンス統計を収集します（アクション・オプション 3）。1 分後に統計の収集時間が経過すると、アクション・オプション 2 にリセットされ、収集の範囲は、すべてのネームスペースで既定値の 15（すべての統計）に設定されます。

ObjectScript

```
DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL("USER")
DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagByNS("USER",3,,7,"M:1:2")
SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
SET tStatement = ##class(%SQL.Statement).%New()
SET qStatus = tStatement.%Prepare(myquery)
IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
GetStats
SET qStatus = tStatement.%Prepare("SELECT * FROM %SYS_PTools.StatsSQLView")
IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QUIT}
SET rsstats = tStatement.%Execute()
DO rsstats.%Display()
WRITE !!,"End of SQL Statistics",!
TerminateResetStats
WRITE "returns:  ",##class(%SYSTEM.SQL.PTools).getSQLStatsFlag(),!
HANG 100
WRITE "reset to: ",##class(%SYSTEM.SQL.PTools).getSQLStatsFlag()
```


4

SQL パフォーマンスのヘルプの入手

インターシステムズのサポート窓口 (WRC) は、クエリのパフォーマンスを解析するためのカスタマ・サポートを提供しています。WRC は、クエリの効率向上を目指す最適化の微調整でお客様を支援できます。クエリの実行のサポートで WRC に問い合わせるには、管理ポータルで以下のいずれかを使用して [レポートを生成] ツールを実行します。

- ・ [システムエクスプローラ]、[ツール]、[SQL パフォーマンス・ツール]、[レポートを生成] の順に選択します。
- ・ [システムエクスプローラ]、[SQL]、[ツール]、[レポートを生成] の順に選択します。

4.1 レポート生成

[レポートを生成] ツールを使用するには、以下の手順を実行します。

1. まず、WRC から WRC 追跡番号を取得する必要があります。それぞれの管理ポータル・ページの上部にある [お問い合わせ] ボタンを使用して、管理ポータルからインターシステムズのサポート窓口にお問い合わせることができます。[WRC 番号] 領域にこの追跡番号を入力します。この追跡番号を使用して、1 つのクエリまたは複数のクエリのパフォーマンスを報告することができます。
2. [SQL文] 領域にクエリ・テキストを入力します。[X] アイコンが右上隅に表示されます。このアイコンを使用して [SQL文] 領域をクリアできます。クエリの入力完了したら、[クエリを保存] ボタンを選択します。クエリ・プランが生成され、指定したクエリに関する実行時統計が収集されます。システム全体の実行時統計設定に関係なく、[レポート生成] ツールでは常に、[コレクション] オプション 3 で収集が行われます。つまり、クエリのすべてのモジュール・レベルの統計が記録されます。このレベルの統計の収集には時間がかかることがあるので、[クエリ保存処理をバックグラウンドで実行] チェック・ボックスにチェックを付けることを強くお勧めします。このチェック・ボックスには既定でチェックが付いています。

バックグラウンド・ジョブが開始されると、“お待ちください...” というメッセージが表示され、ページ上のすべてのフィールドが無効になり、新しい [プロセス表示] ボタンが表示されます。[プロセス表示] ボタンをクリックすると、[プロセス詳細] ページが新しいタブで開きます。[プロセス詳細] ページから、プロセスを参照できます。また、プロセスを一時停止、再開、または終了することもできます。プロセスのステータスは、[クエリを保存] ページに反映されます。プロセスが完了すると、[現在保存しているクエリ] テーブルが更新され、[プロセス表示] ボタンが表示されなくなり、ページ上のすべてのフィールドが有効になります。

3. 目的のそれぞれのクエリで手順 2 を実行します。それぞれのクエリが [現在保存しているクエリ] テーブルに追加されます。このテーブルには、同じ WRC 追跡番号のクエリが含まれていることもあれば、異なる追跡番号のクエリが含まれていることもあります。すべてのクエリが完了したら、手順 4 に進みます。

リストされているそれぞれのクエリについて、[詳細] リンクを選択できます。このリンクを選択すると、別のページが開き、完全な SQL 文、プロパティ (WRC 追跡番号と InterSystems IRIS のソフトウェア・バージョンを含む)、およびクエリ・プランと各モジュールのパフォーマンス統計が表示されます。

- ・ 個々のクエリを削除するには、[現在保存しているクエリ] テーブルでそれらのクエリに対応するチェック・ボックスにチェックを付け、[クリア] ボタンをクリックします。
 - ・ 1 つの WRC 追跡番号に関連付けられているすべてのクエリを削除するには、[現在保存しているクエリ] テーブルから行を選択します。ページの上部にある [WRC 番号] 領域に WRC 番号が表示されます。[クリア] ボタンをクリックすると、その WRC 番号のすべてのクエリが削除されます。
4. クエリのチェック・ボックスを使用して、WRC に報告するクエリを選択します。1 つの WRC 追跡番号に関連付けられているすべてのクエリを選択するには、チェック・ボックスを使用する代わりに、[現在保存しているクエリ] テーブルから行を選択します。どちらの場合も、その後、[レポート生成] ボタンを選択します。[レポート生成] ツールによって、クエリ文、クエリ・プランと実行時統計、クラス定義、および選択したそれぞれのクエリに関連付けられている sql int ファイルを含む xml ファイルが作成されます。
- 1 つの WRC 追跡番号に関連付けられているクエリを選択した場合、生成されるファイルの既定の名前は WRC12345.xml のようになります。複数の WRC 追跡番号に関連付けられているクエリを選択した場合、生成されるファイルの既定の名前は WRCMultiple.xml になります。
- ダイアログ・ボックスが表示され、レポートを保存する場所を指定するように求められます。レポートを保存したら、[メールを送信] リンクをクリックして、レポートをインターシステムズのサポート窓口に送信することができます。メール・クライアントの添付/挿入機能を使用して、ファイルを添付します。