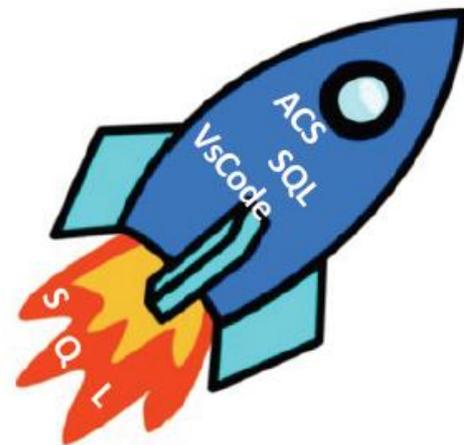


IBM i 2024

IBM i コンテンツ (2024年11月版)

IBM i SQL 中級編 - SQL パフォーマンスの最適化 -

日本アイ・ビー・エム株式会社
テクノロジー事業本部
IBM Power テクニカルセールス



IBM i のSQL中級編

IBM iでは、ストレージのボリューム管理 (IBM iの世界では、HDD、すなわちモーターで駆動する円盤装置は無いにも関わらず今も「ディスク」と呼ぶことが多いですが!) が不要です。また、統計情報が自動で収集され、アクセスプランが自動で再生成されるなどの理由で、いわゆるデータベース管理者としての運用管理も不要です。

そのように素晴らしいIBM iですが、データベースをより快適に使用するには、パフォーマンス・チューニングがお役に立ちます。

今月のコンテンツでは、SQL中級編として、「SQLパフォーマンスの最適化」をお届けします。

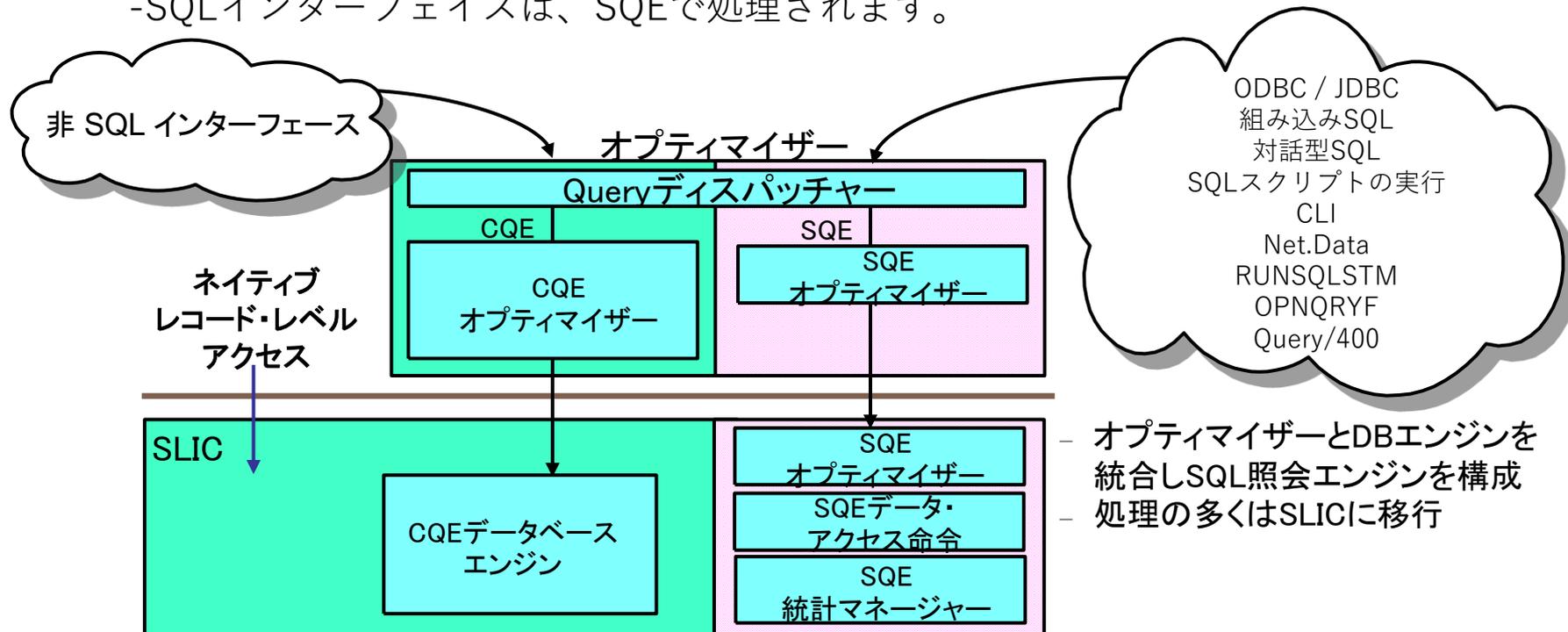
目次

1. SQLを処理するエンジン
2. 索引 を活用する
3. IBM iでのSQLパフォーマンスの基本Tips
4. 補足情報

1. SQL を処理するエンジン

(1) IBM Db2 for i には、照会を処理するための 2 つの照会エンジン

- Classic Query Engine (CQE) と SQL Query Engine (SQE) が用意されています。
- SQL インターフェイスは、SQE で処理されます。



解説：

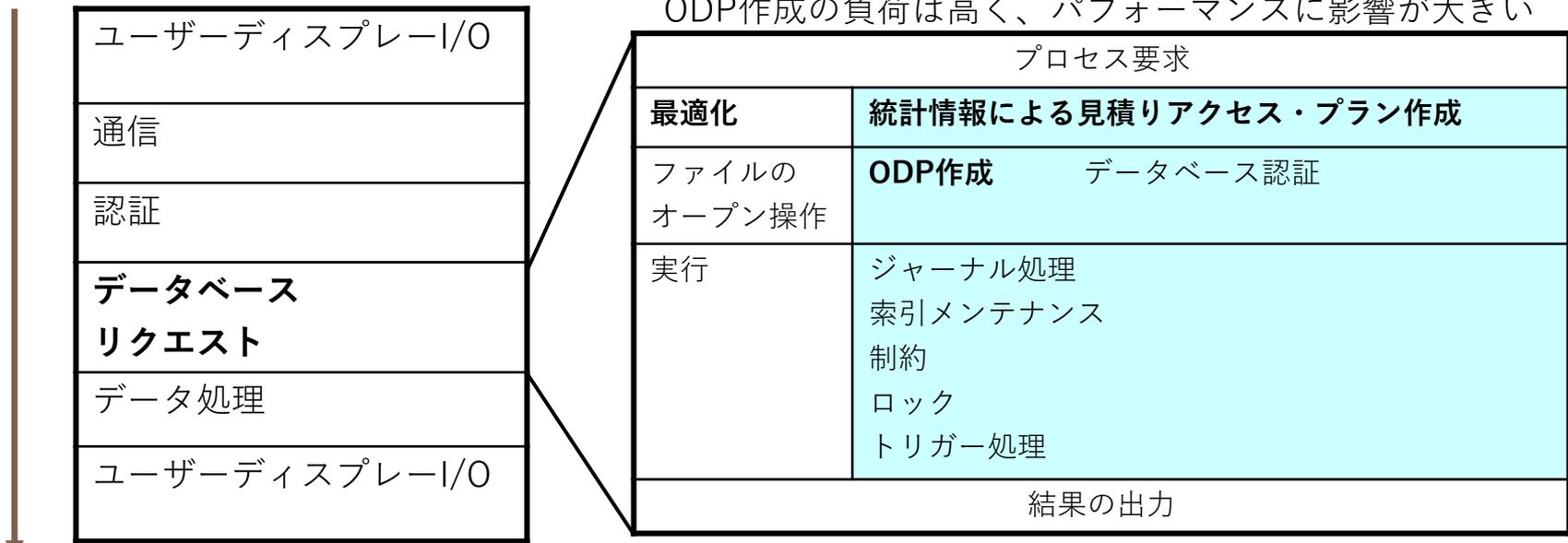
- SQLパフォーマンス最適化の目標は、システムのリソースを最大限に活用して、照会の応答時間を最小限にすることです。リソースを最大限に活用することには、ネットワーク・トラフィック、ディスク入出力、および CPU 時間を最小限にすることが含まれます。
- このSQL処理エンジンについて、SQL照会がどのパスをとるかを決定するプロセスを理解することによって、処理パフォーマンスについてより良い理解を得ることができます。ここでは、簡単にその経路を図示しました。
- ODBC、JDBC、CLI、Query Manager、Net.Data®、RUNSQLSTM、および組み込み SQL または対話式 SQL などの SQL ベースのインターフェースは、SQE (SQL Query Engine) を介して実行されます。
- 以前のリリースでは、Query/400® や OPNQRYPF などの非 SQL 照会の場合、照会のルーティングは QAQQINI SQE_NATIVE_ACCESS オプションによって制御できました。現在、SQE_NATIVE_ACCESS はサポートされなくなっており、非 SQL 照会では、上記の条件のいずれかが含まれていない限り、常に SQE が実行されます。
- プログラム内でのREAD/WRITEなどのコマンドでの「ネイティブ・レコードレベルアクセス」は CQE (Classic Query Engine) になります。

1. SQL を処理するエンジン

(2) SQL処理の実行コンポーネント

~SQL命令の実行時には、下記のような処理が実行されています。

最適化：オプティマイザーは、統計情報を使用してアクセス・プランを作成
 アクセス・プラン：SQL要求の実行に関する情報を含む制御構造
 ODP作成：要求されるすべてのIO操作で使用可能なコード情報を含む
 ODP作成の負荷は高く、パフォーマンスに影響が大きい



解説1：

- SQLパフォーマンス・チューニングのキーは、この「プロセス要求部分」で生じるボトルネックを最小化することです。
- SQLの命令を処理するために、次のようなステップを実行します。

①最適化（統計情報による見積りアクセス・プラン作成）

- 行と列に関する情報(統計)情報からコストを見積もり最適なアクセス・プラン*を作成
 - 統計情報は索引によって提供される
 - コストはリソース使用率ではなく、時間に依存

*アクセスプランとは

-SQL照会を完了するために必要な情報(実行計画)を記述。再使用によりパフォーマンスの向上が期待される

-実行環境の情報（前回、いつアクセス・プランが最適化されたか、メモリー、CPU）や、テーブルや索引

の情報（アクセスされるテーブルや索引）、アクセス方法（テーブル全体のスキャンか、索引で目指す行だけをポイントするか、などのアクセス方法)などを含みます。

アクセスプランの実行は、下記のステップになります。

1. 既存のアクセス・プランの妥当性を検査
2. 妥当でないと判断された場合（*）、再最適化とアクセス・プランの更新を実行
保管されたアクセス・パスを更新せずに、一時アクセス・プランを作成する場合があります
3. アクセス・プランの実行：ODP(オープン・データ・パス) の作成

解説2：(続き)

・アクセスプランの保管場所

- 静的SQLでは、組み込みSQLが含まれるプログラムやモジュールに関連した空間に保管 (コンパイル時に作成される)
- 拡張動的SQLでは、SQLパッケージ (永続オブジェクト) に保管
- その他の動的SQLでは、システムワイド・プラン・キャッシュに保管 (IPLで消去される)

②ファイルのオープン操作

- ・ODP (オープン・データ・パス) は、カーソルがオープンされる時、または他のSQLステートメントが実行される時に作成される内部オブジェクト
- ・ODPの作成負荷は大きい
 - SQLステートメントの最初の実行時には時間がかかる
 - 再使用可能なODPの有無がパフォーマンスに影響する
- ・ODPの再使用
 - ODPの作成を最小にするため、Db2 for iは、ODPをオープンしたままにしておき、ジョブで同じステートメントが実行される場合に再使用しようとする
 - ODPの再使用は、新規にODPを作成する場合の1/10~1/20のCPU負荷
 - 一時索引を、永久索引として作成すると、ODPが再使用される可能性が高くなる
 - 動的SQLでは、SQLステートメントが複数回実行される場合には、1回のPREPARE、複数のEXECUTEにより、フル・オープンを回避できる

2. 索引を活用する

(1) 索引を使うメリット

- ✓ パフォーマンス向上への貢献
 - テーブルの検索速度の向上
 - 統計情報としての利用（オプティマイザーの判断材料）

- ✓ システム資源使用率（CPU）の低減
 - オプティマイザーが、索引の作成を必要と判断した場合、一時索引が作成され、CPU資源を消費するため

- ✓ 索引を作成する際の考慮点
 - 保守コストの発生
 - データの挿入、削除、更新時にオーバーヘッドが発生
 - 不要な索引は削除する

解説：

- Db2 for i はテーブルにアクセスするための2つの基本的な手段を提供する。 テーブル行の 20% 未満が選択されるとき、索引による検索は通常、テーブル走査より効果的です。
- 1章で紹介した、最適化プログラムは最適化に基づいたコストを使用するので、データベースの行および列に関する情報が多いほど、照会のためにより効率的なアクセス・プランを作成できます。索引からの情報を使用して、最適化プログラムは要求 (ローカル選択、結合、グループ化、および順序付け) を処理する方法についてより良い選択をすることができます。
- IBM iでは、索引の実体は、キー付き論理ファイルです。
 - 作成方法に違いがあります。
 - 論理ファイルは、DDSから「CRTLFコマンド」で作成。索引は、「CREATE INDEXコマンド」で作成
 - 後述する、ACS付属の「索引アドバイザー」で、半自動作成できます。

詳しくは下記参照

<https://www.ibm.com/docs/ja/i/7.5?topic=optimization-creating-index-strategy>

2. 索引を活用する

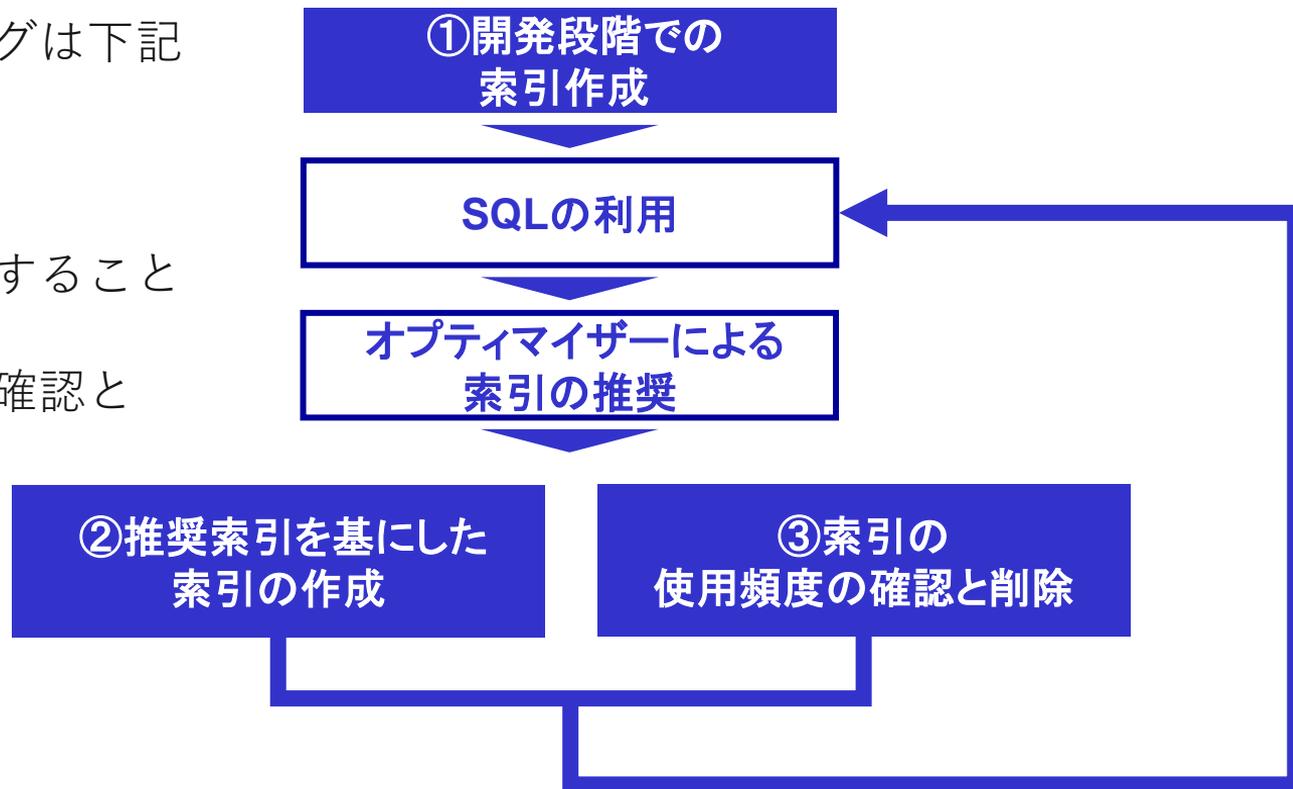
(2) 索引の作成 (索引の作成・確認・削除の流れ)

索引作成のタイミングは下記

- ①開発段階
- ②推奨索引の作成

定期的に下記を実施することを推奨します。

- ③索引の使用頻度の確認と削除



解説：

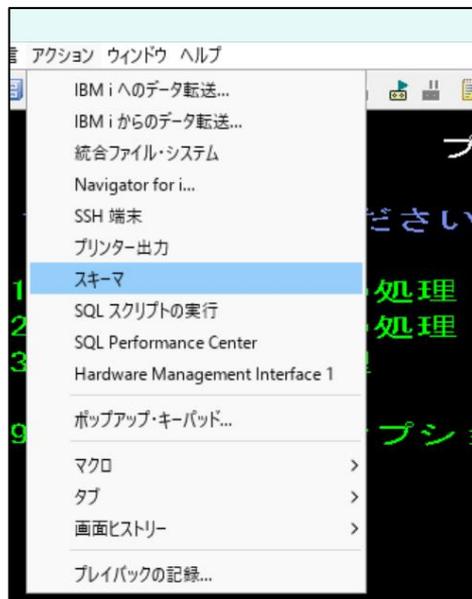
- ・ 推奨索引と一時索引について
 - Db2 for iが常にデータベース統計情報を自動収集している
 - SQL解析時に、存在していれば大きな改善が見られると考えられる索引を、「索引アドバイザー」が推奨する
 - 推奨された索引の内、オプティマイザーが関連するSQLの実行頻度を考慮し索引を作成した方が、パフォーマンス上有効だと判断したものを一時索引として作成する
 - 作成された一時索引は、プランキャッシュに蓄積され、他のSQLも含めて再利用される (IPLされると消去されます)
- ・ 索引アドバイザー情報へのアクセスはさまざまな方法で行えます。例えば、次のような方法があります。
 - IBM i Access Client Solutions (ACS) の索引アドバイザー・インターフェース
 - SQL パフォーマンス・モニターの Show ステートメント
 - Visual Explain インターフェース
 - データベース・モニター・ビュー 3020 の照会 - 推奨索引

2. 索引を活用する

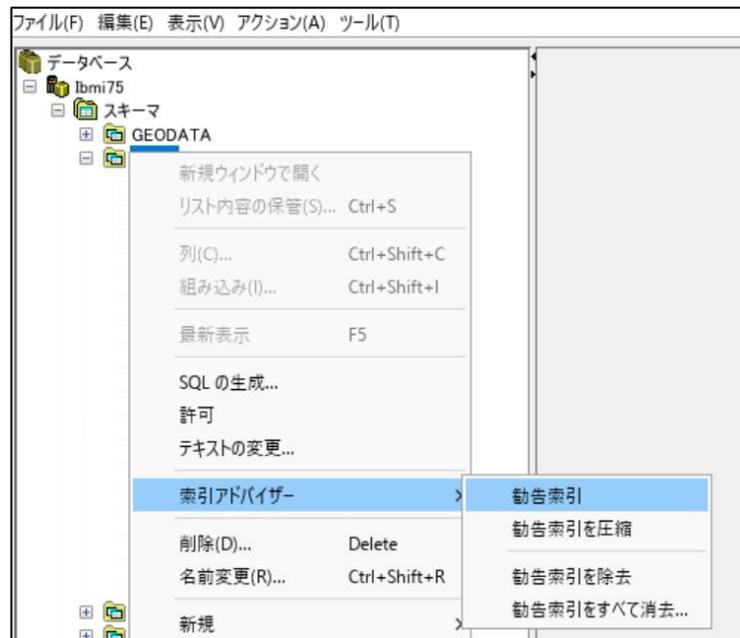
(3) 推奨索引を基にした索引の作成 -1

IBM i Access Client Solutions (ACS) の索引アドバイザー・インターフェースを使った索引の作成方法

① ACSの5250画面で、「アクション」→「スキーマ」を選択



② 「データベース」→「スキーマ」→該当のスキーマ「索引アドバイザー」→「勧告索引」を選択



2. 索引を活用する

(3) 推奨索引を基にした索引の作成 -2

- ③下記の画面のような勧告リストが表示される。
 実際は、右側には（下記では見えてないが）
 「勧告の回数」や、日付の表示がある。

ファイル(F) 編集(E) 表示(V) アクション(A)

QEOLの勧告索引

索引の勧告対象の表	システム名	スキーマ	システム・スキーマ	区画	勧告キー	順序独立先
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKTIKU	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKPOST	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKPOST	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKPOST, TKTIKU	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKPOST, TKTIKU	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKTIKU, TKGURI	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKTIKU, TKGURI	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKPOST, TKTIKU, TKADR1	
TOKMSP	TOKMSP	QEOL	QEOL	すべての区	TKPOST, TKTIKU, TKADR1	

- ④前のメニューにあった、「勧告索引の圧縮」を選択すると重複した索引が排除され、最小限の索引が表示される

ファイル(F) 編集(E) 表示(V) アクション(A)

QEOLの勧告索引(圧縮)

索引の勧告対象の表	システム名	スキーマ	システム・スキーマ	区画	勧告キー	勧告索引タイプ
TOKMSP	TOKMSP	QEOL	QEOL	すべての区画の場合	TKTIKU, TKGURI	2進基数
TOKMSP	TOKMSP	QEOL	QEOL	すべての区画の場合	TKPOST, TKTIKU, TKADR1	2進基数
TOKMSP	TOKMSP	QEOL	QEOL	すべての区画の場合	TKPOST, TKTIKU, TKADR1	エンコードV4
TOKMSP	TOKMSP	QEOL	QEOL	すべての区画の場合	TKTIKU, TKGURI	エンコードV4

2. 索引を活用する

(3) 推奨索引を基にした索引の作成 -3

- ⑤一番勧告回数が多かった、索引を作成します。右クリックで、勧告索引の表から「索引の作成」を選択します。

ファイル(F) 編集(E) 表示(V) アクション(A)

QEOL の勧告索引 (圧縮)

索引の勧告対象の表	システム名	スキーマ	システム・スキーマ	区画	勧告キー
TOKMSP	TOKMSP	QEOL	QEOL	すべての区画の場合	TKTIKU, TKGURI
TOKMSP	SP	QEOL	QEOL	すべての区画の場合	TKPOST, TKTIKU, TKADR1
TOKMSP	SP	QEOL	QEOL	すべての区画の場合	TKPOST, TKTIKU, TKADR1
TOKMSP	SP	QEOL	QEOL	すべての区画の場合	TKTIKU, TKGURI

右クリックメニュー:

- 索引の作成
- SQL の表示
- 処理 >
- 表 >

- ⑥下記で「名前」を入力します。(これだけです)
ここでは例として「TOKMSPXXL1」を入力しました。キーなど他の項目は自動入力されています。最下部の「OK」をクリックします。

索引 キー WHERE 文節 オプション

名前: TOKMSPXXL1

スキーマ: QEOL

システム名: システム生成

テーブル・スキーマ: QEOL

表名: TOKMSP

索引タイプ: 固有でない

ソリッド・ステート・ドライブを優先記憶媒体にする

メモリーに保持

テキスト: 索引アドバイザーから生成された索引

索引 キー WHERE 文節 オプション

順序	名前	式	システム名
昇順	TKTIKU		システム生成
昇順	TKGURI		システム生成

以上が、索引の作成手順になります。

解説：

- IBM® i Access Client Solutions (ACS) を使用して、オプティマイザーから索引アドバイザー情報を表示できます。（QSYS2/SYSIXADV システム・テーブルで検出された情報を表示します。）
- 「勧告索引の圧縮」が利用できない場合は、通常の「勧告索引」を選択する
- 勧告回数の多いものや、MTI(一時索引) の作成回数が確認でき、手動で索引を作成できる。
- 索引アドバイザーの各列項目についての詳細は下記を参照
<https://www.ibm.com/docs/ja/i/7.5?topic=advisor-column-descriptions>
- 索引を作成すると、下記のように「索引」を選択すると、作成済みの索引がリスト表示されます。



2. 索引を活用する

(4) 索引の使用頻度の確認と削除-1

- ✓ 索引の使用回数をリアルタイムで確認可能
 - 使用されていない索引は、更新系作業の無駄な負荷になるため削除することを推奨

- ✓ ただし、オプティマイザーへの情報提供は重要
 - 「照会使用回数」が「0」であっても「照会統計使用回数」が「0」でない場合は削除しない。

- ✓ 確認方法
 - 先ほどの「索引」をクリック
 - 右図のように、
-照会使用回数と
照会統計使用回数
が重要です。使用されている場合は削除しないこと

デキプロ	キー・カラム	照会の最終使用	照会統計の最終使用	照会使用回数	照会統計使用回数	最終使用
1/10 1..	QQC11, QQRID		2023/11/10 17:38:47	0	4	
5/10 1..	QQC11, QQRID		2024/05/10 17:29:03	0	3	
1/10 1..	QQC12, QQRID, ...		2023/11/10 17:38:47	0	11	
5/10 1..	QQC12, QQRID, ...		2024/05/10 17:29:04	0	11	
1/10 1..	QQC21, QQJFLD, ...	2023/11/10 17:38:45	2023/11/10 17:38:47	1	13	2023/11/10
5/10 1..	QQC21, QQJFLD, ...	2024/05/10 17:29:03	2024/05/10 17:29:04	1	13	2024/05/10
1/10 1..	QQJFLD, QQI5		2023/11/10 17:38:47	0	32	
5/10 1..	QQJFLD, QQI5		2024/05/10 17:29:04	0	32	
1/10 1..	QQJFLD, QQRID, ...	2023/11/10 17:38:47	2023/11/10 17:38:47	4	20	2023/11/10
5/10 1..	QQJFLD, QQRID, ...	2024/05/10 17:29:04	2024/05/10 17:29:04	4	20	2024/05/10
1/10 1..	QQRID, QQC21	2023/11/10 17:38:45	2023/11/10 17:38:46	1	7	2023/11/10
5/10 1..	QQRID, QQC21	2024/05/10 17:29:03	2024/05/10 17:29:03	2	8	2024/05/10
1/10 1..	QQRID, QQI5	2023/11/10 17:38:46	2023/11/10 17:38:47	13	23	2023/11/10
5/10 1..	QQRID, QQI5	2024/05/10 17:29:03	2024/05/10 17:29:04	12	22	2024/05/10
1/10 1..	QQUCNT		2023/11/10 17:38:47	0	5	
5/10 1..	QQUCNT		2024/05/10 17:29:04	0	5	

解説：

- ・照会使用回数は、「アクセス・プランの実装に使用された回数」です
- ・照会統計使用回数は、「最適化のための情報提供元として使用された回数」です。

3. IBM i でのSQLパフォーマンス基本Tips

Tips1: 繰り返しの多いSQLは、Prepared Statementを使用

- ✓ PreparedStatement(パラメータ・マーカーを使用) の利用により
 - 呼び出し回数の多いSQLで利用されているテーブルのOpen/Close処理にかかるシステム資源消費や、処理時間の短縮に効果がある
 - Web環境では、データソース側でのPrepareStatement キャッシュの効果も期待できる

【 Statement処理の動き 】

SQL Optimize

Describe

Table Open

Fetch

Table Close

【 PreparedStatementでの動き】

SQL Optimize

Prepare

Table Open

Fetch

SQL Optimize

Fetch

3. IBM i でのSQLパフォーマンス基本Tips

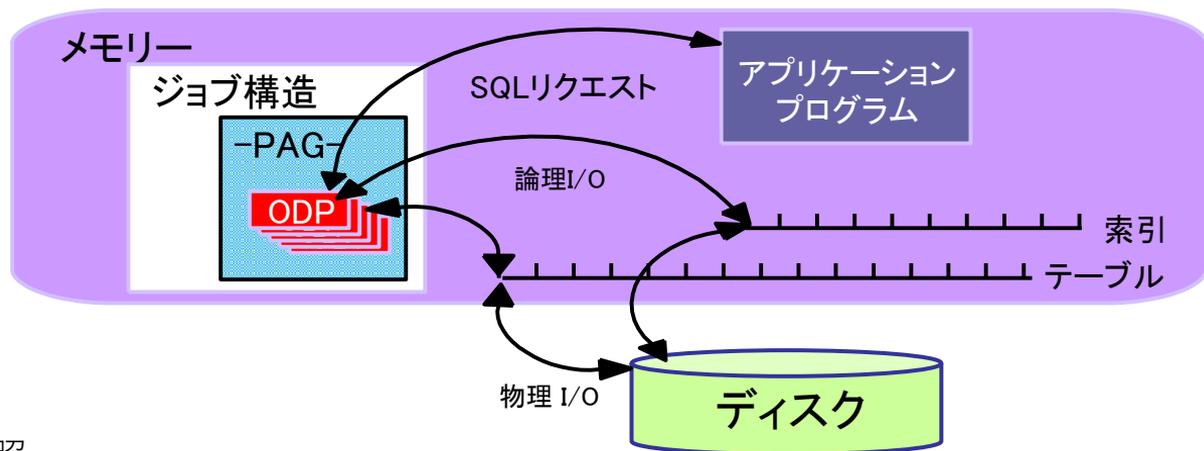
Tips2: テーブルオープン回数は最小限にする

- ✓ SQLテーブルのOpen/Closeはシステム資源を消費するタスク

- ✓ テーブルオープンを伴うSQLステートメント
 - OPEN ステートメント
 - SELECT INTO ステートメント
 - VALUES 文節を伴う INSERT ステートメント
 - WHERE 条件を伴う UPDATE ステートメント
 - WHERE CURRENT OF カーソル文節および演算子または関数を参照する SET 文節を伴う UPDATE ステートメント
 - 式を含む SET ステートメント
 - 式を含む VALUES INTO ステートメント
 - WHERE 条件を伴う DELETE ステートメント

解説：

- SQL データ操作言語ステートメントでは、データへのオープン・データ・パス (ODP) を作成するためにデータベースのオープン操作が必要になります。オープン・データ・パスは、テーブルに対する入出力操作すべてを行うためのパスです。ある意味では、これは SQL アプリケーションとテーブルを結び付けるものです。プログラムでのオープン操作の回数は、パフォーマンスに大きく影響します。
- オープンの回数を最小限にするために、Db2 for i はオープン データ パス (ODP) をオープンのままにしておき、ステートメントが再度実行されたときに ODP を再利用します



- より詳細な解説は下記のURLを参照

<https://www.ibm.com/docs/ja/i/7.5?topic=tips-reduce-open-operations>

3. IBM i でのSQLパフォーマンス基本Tips

Tips3: SELECT * の使用は避ける

- ✓ Native I/Oではレコードの読み書きの際は、全てのカラムデータがメモリーに読み込まれる
- ✓ SQLでは選択したカラムのみのデータが読み書き可能
- ✓ 索引に指定されているカラムのみの指定であれば、実テーブルへのアクセスは発生しない

解説：

- ・ 必要以外のカラムデータはアクセスの無駄になります。
- ・ パフォーマンスの観点からは、アプリケーションで必要とする列を明示的にコーディングする方がはるかに効率的です。この手法は、索引専用アクセスにおいて、あるいはすべての列がソート操作の対象とされる場合 (SELECT DISTINCT および SELECT UNION のときなど) において、特に重要です。
- ・ より詳細な解説は下記のURLを参照
<https://www.ibm.com/docs/ja/i/7.5?topic=techniques-optimize-columns-selected>

3. IBM i でのSQLパフォーマンス基本Tips

Tips4 : データタイプの変換は避ける

異なるタイプの比較では、索引があっても利用されず
テーブルスキャンになる可能性がある。

- ✓ カラムと比較する変数、定数のデータ型や属性が異なる場合
(例) Order_TotalカラムがDecimal 9.2の場合
where Order_Total = 200.0000にした場合
- ✓ 以下のことに注意
 - 同じデータタイプで比較
 - できる限り同じ桁数で比較
 - できる限り同じ精度で比較
- ✓ 比較式などでは、桁数や小数点の位置を合わせよう

3. IBM i でのSQLパフォーマンス基本Tips

Tips5 : 算術演算式の使用禁止

- ✓ 行選択における比較式の中での算術式は避ける
(Where句内での算術演算式は、データアクセスに悪影響を及ぼす)
- ✓ 索引が存在しても、SQLオプティマイザーは索引を使用しないように判断する

(例) 組み込みSQLなどでは、

- ✗ Where Order_Delivery = Current_Date + 7 Days
- NextWeek = %Date() + %Days(7)
Where Order_Delivery = :NextWeekで指定する

3. IBM i でのSQLパフォーマンス基本Tips

Tips6 : LIKE句使用の注意

- ✓ LIKE句の指定の際の考慮点
 - AA%などの指定では、索引が利用される
 - %AAなどは索引が利用されず、テーブルスキャンが発生
- ✓ スカラー関数SUBSTR()、POSTSTR()でLIKE句と同様の条件を記述できる

(例)

- ✗ WHERE Column = LIKE 'A%BB%'
- WHERE SUBSTR (Column, 1, 1) = 'A' and
POSTSTR(Column, 'BB') < > 0

- ✓ LIKE句で遅い場合は、SUBSTRなどを利用してみる

解説：

- ここで紹介したTipsは、すべての環境で有効ではありません。
- SQLパフォーマンスに関するその他のヒントは、下記のURLが参考になります
 - 一般的な Db2 for i パフォーマンスの考慮事項
<https://www.ibm.com/docs/ja/i/7.5?topic=optimization-performance-considerations>
 - データベース・パフォーマンスに関するアプリケーション設計のヒント
<https://www.ibm.com/docs/ja/i/7.5?topic=optimization-application-design-tips>
 - データベース・パフォーマンス向上のためのプログラミング方法
<https://www.ibm.com/docs/ja/i/7.5?topic=optimization-programming-techniques>
- SQLパフォーマンスは、お客様の実行環境に依存するもので、前提として、SQLパフォーマンスモニターと分析、チューニングが必要です。

(パフォーマンスモニターとVisual Explainは、次号のSQL上級編でご紹介する予定です。)

4. 補足情報

1. Db2 for i SQL 解説書 (IBM i 7.5版)
https://www.ibm.com/docs/ja/ssw_ibm_i_75/pdf/rbafzpdf.pdf
2. SQL プログラミング (IBM i 7.5版)
https://www.ibm.com/docs/ja/ssw_ibm_i_75/pdf/rbafypdf.pdf
3. データベース・パフォーマンスおよび Query 最適化
<https://www.ibm.com/docs/ja/i/7.5?topic=database-performance-query-optimization>

IBM i 関連情報

IBM i ポータル・サイト

<https://ibm.biz/ibmijapan>

i Magazine (IBM i 専門誌。春夏秋冬の年4回発刊)

<https://www.imagazine.co.jp/IBMi/>

IBM i World 2024 オンデマンド・セミナー

<https://video.ibm.com/recorded/133917616>

IBM i World 2023 オンデマンド・セミナー

<https://ibm.biz/ibmiworld2023>

IBM i World 2022 オンデマンド・セミナー

<https://video.ibm.com/recorded/132423205>

月イチIBM Power情報セミナー「IBM Power Salon」

<https://ibm.biz/power-salon>

IBM i 関連セミナー・イベント

<https://ibm.biz/powerevents-j>

IBM i Club (日本のIBM i ユーザー様のコミュニティー)

<https://ibm.biz/ibmiclubjapan>

IBM i 研修サービス (i-ラーニング社提供)

<https://www.i-learning.jp/service/it/iseries.html>

IBM TechXchange Powerユーザーコミュニティー (日本)

<https://ibm.biz/ibm-power-user-community>

IBM i 情報サイト iWorld

<https://ibm.biz/iworldweb>

IBM i サポートロードマップ

<https://public.dhe.ibm.com/systems/support/planning/transfer/IBM.i.Support.Roadmap.pdf>

IBM i 7.5 技術資料

<https://www.ibm.com/docs/ja/i/7.5>

IBM Power ソフトウェアのダウンロードサイト (ESS)

<https://ibm.biz/powerdownload>

Fix Central (HW・SWのFix情報提供)

<https://www.ibm.com/support/fixcentral/>

IBM My Notifications (IBM IDの登録 [無償] が必要)

「IBM i」「9009-41G」などPTF情報の必要な製品を選択して登録できます。

<https://www.ibm.com/support/mynotifications>

IBM i 各バージョンのライフサイクル

<https://www.ibm.com/support/pages/release-life-cycle>

IBM i 以外のSWのライフサイクル (個別検索)

<https://www.ibm.com/support/pages/lifecycle/>

IBM Power Systems Virtual Server 情報

<https://ibm.biz/pvsjapan>

IBM i Advantage 2024 開催決定!

12月3日(火)・4日(水)

IBM虎ノ門イノベーションスタジオ (メイン会場)

IBM大阪中之島フェスティバルタワー・ウエスト (中継)

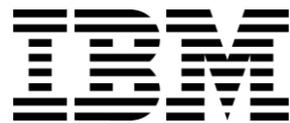
12月24日(火)

Webセミナー + Q&Aセッション

下記サイトよりお申込みをお願いします↓

<https://ibm.biz/ibmiadvantage2024>





ワークショップ、セッション、および資料は、IBMによって準備され、IBM独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる読者に対しても法律的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、読者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、Db2、Rational、Power、POWER8、POWER9、AIXは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。

他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。

現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporation または子会社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。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社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。