

# Oracle Database 19cにおける オプティマイザ統計収集の ベスト・プラクティス

Oracleホワイト・ペーパー / 2019年12月9日

## はじめに

Oracleオプティマイザは、SQL文に対して考えられるすべての計画を調査し、コストのもっとも低い計画を選択します。ここで、コストとは、特定の計画における推定リソース使用率のことです。オプティマイザが実行計画のコストを正確に判断するためには、SQL文でアクセスするすべてのオブジェクト（表と索引）に関する情報と、SQL文を実行するシステムに関する情報をオプティマイザが利用できる必要があります。

必要となるこの情報は一般に、オプティマイザ統計と呼ばれます。オプティマイザ統計を理解して管理することが、最適なSQL実行の鍵となります。統計を収集するタイミングや、統計を適切なタイミングで収集する方法を把握することは、許容可能なパフォーマンスを維持する上で不可欠です。このホワイト・ペーパーは、オプティマイザ統計に関する2部構成シリーズの第2部です。このシリーズの第1部『Oracle Database 19cのオプティマイザ統計について』は、統計の概要に焦点を当てており、本書でも、追加の情報源として何處か参照します。本書では、Oracle Databaseにおけるもっとも一般的なシナリオで統計を収集するタイミングおよび方法を詳しく説明します。トピックは以下のとおりです。

- 統計の収集方法
- 統計を収集するタイミング
- 統計の質の向上
- 統計のより迅速な収集
- 統計を収集すべきでない場合
- その他のタイプの統計の収集

## 免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意した Oracle Software License and Service Agreement の諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料にならないで下さい。本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

製品アーキテクチャの性質により、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

## 目次

はじめに .....	2
統計の収集方法 .....	4
統計を収集するタイミング .....	9
オプティマイザ統計の品質確保 .....	16
統計のより迅速な収集 .....	18
統計を収集すべきでない場合 .....	20
その他のタイプの統計の収集 .....	22
結論 .....	24
参考資料 .....	25

## 統計の収集方法

### 戦略

Oracleで統計を収集する推奨の方法は、自動統計収集を使用することです。十分に確立された手動の統計収集プロセッサーがすでにある場合は、代わりにそのプロセッサーを使用することを望むかもしれません。どのような方法を使用することにしても、デフォルトのグローバル・プリファレンスが自身のニーズを満たしているかどうかを確認することから始めてください。ほとんどの場合は満たしているでしょうが、変更が必要な場合は、SET\_GLOBAL\_PREFSを使用して変更できます。変更後、DBMS\_STATSの“set preference”プロセッサーを使用して、必要に応じてグローバルのデフォルト設定をオーバーライドします。たとえば、増分統計や特定のヒストグラム一式が必要な表で、SET\_TABLE\_PREFSを使用します。そうすることで、統計の収集方法を宣言したことになり、個々の“統計収集”操作のパラメータをカスタマイズする必要がなくなります。また、表/スキーマ/データベース統計の収集にデフォルトのパラメータを自由に使用して、選択した統計ポリシーを確実に順守できます。さらには、自動および手動の統計収集を自由に切り替えて使用できます。

この項では、この戦略の実装方法について説明します。

### 自動統計収集

Oracle Databaseは、統計がない、または統計が“古い”（期限切れの）データベース・オブジェクトの統計を収集します。この処理は、事前定義されたメンテナンス時間枠で実行される自動タスクによって行われます。Oracleでは、統計が必要なデータベース・オブジェクトが内部で優先順位付けされるため、最新の統計をもっとも必要とするオブジェクトが最初に処理されます。

自動統計収集ジョブでは、DBMS\_STATS.GATHER\_DATABASE\_STATS\_JOB\_PROCプロセッサーが使用されます。このプロセッサーは、他のDBMS\_STATS.GATHER\_\*\_STATSプロセッサーと同じデフォルト・パラメータ値を使用します。ほとんどの場合は、デフォルト値で十分です。ただし、統計収集/パラメータの1つでデフォルト値を変更する必要がある場合もあります。そのような場合は、DBMS\_STATS.SET\_\*\_PREFプロセッサーを使用します。パラメータ値は、可能な限り最小範囲で、理想としてはオブジェクトごとに変更する必要があります。たとえば、特定の表において、行がデフォルトの10 %ではなく、5 %しか変更されていないと、統計が古いと見なされるため、その表の古いしきい値を変更したいとします。その場合、DBMS\_STATS.SET\_TABLE\_PREFSプロセッサーを使用して、その表のSTALE\_PERCENT表プリファレンスを変更します。最小範囲でデフォルト値を変更することにより、手動で変更する必要があるデフォルト以外のパラメータ値の数が制限されます。例として、SALES表でSTALE\_PERCENTを5 %に変更する方法を以下に示します。

```
exec dbms_stats.set_table_prefs(user,'SALES','STALE_PERCENT','5')
```

どのプリファレンスが設定されているかを確認するには、DBMS\_STATS.GET\_PREFS関数を使用します。以下のように、パラメータ名、スキーマ名、表名という3つの引数を渡します。

```
select dbms_stats.get_prefs('STALE_PERCENT',user,'SALES') stale_percent
from   dual;
STALE_PERCENT
-----
5
```

## DBMS\_STATSプリファレンスの設定

上記で示したように、DBMS\_STATSプリファレンスを特定の対象オブジェクトとスキーマに設定し、必要に応じて自動統計収集の動作を変更できます。個々のDBMS\_STATS.GATHER\_\*\_STATSコマンドに対して、デフォルトではない特定のパラメータ値を指定することができますが、“対象の”DBMS\_STATS.SET\_\*\_PREFSプロシージャを使用して、必要に応じてデフォルト値をオーバーライドすることが推奨されます。

パラメータのオーバーライドは、次のプロシージャのいずれかを使用して、表、スキーマ、データベース、またはグローバル・レベルで指定できます (AUTOSTATS\_TARGETおよびCONCURRENTは、グローバル・レベルでのみ変更できます)。

```
SET_TABLE_PREFS
SET_SCHEMA_PREFS
SET_DATABASE_PREFS
SET_GLOBAL_PREFS
```

従来は、もっともよくオーバーライドされるプリファレンスは、ESTIMATE\_PERCENT（サンプルの行の割合を制御）とMETHOD\_OPT（ヒストグラムの作成を制御）でしたが、現在は、見積りの割合はデフォルト値のままにしておくのが適切です。理由については、この項の後半で説明します。

SET\_TABLE\_PREFSプロシージャでは、指定された表に限って、DBMS\_STATS.GATHER\_\*\_STATSプロシージャで使用されるパラメータのデフォルト値を変更できます。

SET\_SCHEMA\_PREFSプロシージャでは、指定されたスキーマのすべての既存表で、DBMS\_STATS.GATHER\_\*\_STATSプロシージャによって使用されるパラメータのデフォルト値を変更できます。このプロシージャは、指定されたスキーマの各表で、SET\_TABLE\_PREFSプロシージャを実際に呼び出します。SET\_TABLE\_PREFSを使用するため、このプロシージャの呼出しが、プロシージャの実行後に作成される新たなオブジェクトに影響を与えることはありません。新たなオブジェクトでは、すべてのパラメータにGLOBALプリファレンス値が使用されます。

SET\_DATABASE\_PREFSプロシージャでは、データベース内のユーザーによって定義されたすべてのスキーマで、DBMS\_STATS.GATHER\_\*\_STATSプロシージャによって使用されるパラメータのデフォルト値を変更できます。このプロシージャは、ユーザーによって定義されたスキーマごとの各表で、SET\_TABLE\_PREFSプロシージャを実際に呼び出します。SET\_TABLE\_PREFSを使用するため、このプロシージャが、プロシージャの実行後に作成される新たなオブジェクトに影響を与えることはありません。新たなオブジェクトでは、すべてのパラメータにGLOBALプリファレンス値が使用されます。ADD\_SYSパラメータをTRUEに設定することで、Oracleが所有するスキーマ (sys、systemなど) を含めることも可能です。

SET\_GLOBAL\_PREFSプロシージャでは、既存の表プリファレンスがないデータベースの任意のオブジェクトで、DBMS\_STATS.GATHER\_\*\_STATSプロシージャによって使用されるパラメータのデフォルト値を変更できます。すべてのパラメータは、表プリファレンスが設定されている場合、およびパラメータがGATHER\_\*\_STATSコマンドで明示的に設定されている場合を除き、グローバル設定がデフォルトとなります。このプロシージャによって加えられた変更は、プロシージャの実行後に作成される新たなオブジェクトに影響を与えます。新たなオブジェクトでは、すべてのパラメータにGLOBAL\_PREFS値が使用されます。

DBMS\_STATS.GATHER\_\*\_STATSプロシージャと自動統計収集タスクは、以下のパラメータ値の階層に従います。コマンドで明示的に設定されたパラメータ値により、それ以外の値がすべて無効になります。パラメータがコマンドで設定されていない場合、表レベルのプリファレンスを確認します。表レベルのプリファレンスが設定されていない場合、GLOBALプリファレンスを使用します。



図1：パラメータ値のDBMS\_STATS.GATHER\_\*\_STATS階層

Oracle Database 12c Release 2には、PREFERENCE\_OVERRIDES\_PARAMETERと呼ばれる新たなDBMS\_STATSプリファレンスが導入されています。このDBMS\_STATSプリファレンスの影響を図2に示します。このプリファレンスがTRUEに設定されている場合、プリファレンス設定によりDBMS\_STATSパラメータ値をオーバーライドできます。たとえば、グローバル・プリファレンスのESTIMATE\_PERCENTがDBMS\_STATS.AUTO\_SAMPLE\_SIZEに設定されている場合、既存の手動統計収集プロシージャが異なるパラメータ設定（例：10 %などのサンプル・サイズの固定割合）を使用する場合でも、このベスト・プラクティス設定が使用されることを表します。

PREFERENCE\_OVERRIDES\_PARAMETER



図2：DBMS\_STATSプリファレンスであるPREFERENCE\_OVERRIDES\_PARAMETERの使用

## ESTIMATE\_PERCENT

ESTIMATE\_PERCENT/パラメータは、統計を計算するために使用される行の割合を決定します。もっとも正確な統計は、表のすべての行が処理される（つまりサンプルが100 %の）ときに収集され、通常は算出された統計と呼ばれます。Oracle Database 11gでは、確定的な統計を実現するハッシュ・ベースの新しいサンプリング・アルゴリズムが導入されました。この新たなアプローチでは、最大で10 %のサンプルのコストで、100 %のサンプルに近い正確性を実現します。新たなアルゴリズムは、任意のDBMS\_STATS.GATHER\_\*\_STATSプロシージャにおいて、ESTIMATE\_PERCENTがAUTO\_SAMPLE\_SIZE（デフォルト）に設定されている場合に使用されます。Oracle Database 11g以前は、DBAはESTIMATE\_PERCENTパラメータを小さい値に設定して、統計を迅速に収集できるようにしていました。しかしながら、詳細なテストをせずに、どのサンプル・サイズを使用すれば正確な統計を取得できるかを把握することは困難です。Oracle Database 11g以降では、ESTIMATE\_PERCENTにはデフォルトのAUTO\_SAMPLE\_SIZEを使用することが強く推奨されます。新しいヒストグラム・タイプ（HYBRIDとTop-Frequency）は、自動サンプル・サイズが使用されている場合のみ作成できるため、これは特に重要です。

多くのシステムには、手動で見積りの割合を設定する古い統計収集スクリプトが含まれているため、Oracle Database 19cにアップグレードする際は、PREFERENCE\_OVERRIDES\_PARAMETERプリファレンス（上記参照）を使用して、自動サンプル・サイズの使用を適用することを検討してください。

## METHOD\_OPT

METHOD\_OPT/パラメータは、統計の収集中にヒストグラム<sup>1</sup>の作成を制御します。ヒストグラムは、表列のデータ配布に関する詳細な情報を提供するために作成された特殊な列統計です。

METHOD\_OPTのデフォルトの推奨値は、FOR ALL COLUMNS SIZE AUTOです。この値は、ヒストグラムの恩恵を受ける可能性が高い列に対して、ヒストグラムが作成されることを表します<sup>2</sup>。列は、WHERE col1='X'またはWHERE col1 BETWEEN 'A' and 'B'といった等式や範囲述語で使用されている場合や、とりわけ列値の配布に偏りがある場合に、ヒストグラムの候補となります。オプティマイザは、どの列が問合せ条件で使用されるかを把握しています。この情報は、ディクショナリ表SYS.COL\_USAGE\$に記録および保存されているためです。

一部のDBAは、いつ、どのようなヒストグラムが作成されるかを厳密に制御することを望んでいます。これを行う推奨の方法は、SET\_TABLE\_PREFSを使用して、表ごとにどのヒストグラムを作成するかを指定することです。SALESで、col1およびcol2のみ、ヒストグラムを作成するように指定する方法を以下に示します。

```
begin
  dbms_stats.set_table_prefs(user,
    'SALES',
    'method_opt',
    'for all columns size 1 for columns size 254 col1 col2');
end;
/
```

ヒストグラム（col1およびcol2）を作成する列を指定できることに加えて、オプティマイザは、次のように、追加のヒストグラムが有用かどうかを決定できます。

```
begin
  dbms_stats.set_table_prefs(user,
    'SALES',
    'method_opt',
    'for all columns size auto for columns size 254 col1 col2');
end;
/
```

<sup>1</sup> ヒストグラムの作成に関する詳細については、このホワイト・ペーパー・シリーズの第1部『Oracle Database 19cのオプティマイザ統計について』を参照してください。

<sup>2</sup> ホワイト・ペーパー / Oracle Database 19cにおけるオプティマイザ統計収集のベスト・プラクティス

ヒストグラムの作成は、METHOD\_OPTが'FOR ALL COLUMNS SIZE 1'に設定されている場合、無効になります。たとえば、METHOD\_OPTのDBMS\_STATSグローバル・プリファレンスを次のように変更できるため、ヒストグラムはデフォルトでは作成されません。

```
begin
  dbms_stats.set_global_prefs(
    'method_opt',
    'for all columns size 1');
end;
/
```

不要なヒストグラムは、すべての列統計情報を削除しなくとも、DBMS\_STATS.DELETE\_COLUMN\_STATSを使用して*col\_stat\_type*を'HISTOGRAM'に設定することで削除できます。

## 手動統計収集

十分に確立された統計収集プロシージャがすでにある場合、または何らかの理由で主要なアプリケーション・スキーマに対する自動統計収集を無効にしたい場合、ディクショナリ表に委ねることを検討してください。これを行うには、DBMS\_STATS.SET\_GLOBAL\_PREFSプロシージャを使用して、AUTOSTATS\_TARGETパラメータの値をAUTOではなくORACLEに変更します。

```
exec dbms_stats.set_global_prefs('autostats_target','oracle')
```

統計を手動で収集するには、PL/SQL DBMS\_STATSパッケージを使用する必要があります。廃止されたANALYZEコマンドは使用しないでください。DBMS\_STATSパッケージには、ユーザー・スキーマ・オブジェクト、ディクショナリ、および固定オブジェクトで統計を収集できるDBMS\_STATS.GATHER\_\*\_STATSプロシージャが複数用意されています。これらのプロシージャのパラメータはすべて、スキーマ名とオブジェクト名を除き、デフォルトのままにしておくのが理想的です。ほとんどの場合、オラクルの選択した適応性のあるデフォルトのパラメータ設定で十分です。

```
exec dbms_stats.gather_table_stats('sh','sales')
```

上記で述べたように、ある統計収集パラメータのデフォルト値を変更することが必要になった場合は、DBMS\_STATS.SET\_\*\_PREFプロシージャを使用して、可能な限り最小範囲で、理想としてはオブジェクトごとに、変更を加えます。

## 統計の保留

DBMS\_STATS.GATHER\_\*\_STATSプロシージャのパラメータのデフォルト値を変更する場合、本番環境に変更を加える前に、それらの変更を検証することが強く推奨されます。完全な規模のテスト環境がない場合は、保留中の統計を活用する必要があります。保留中の統計では、統計は通常のディクショナリ表に格納されずに、保留中の表に格納されるため、統計が公開され、システム全体で使用される前に、制御された方法で有効化してテストできます。保留中の統計収集を有効にするには、いずれかのDBMS\_STATS.SET\_\*\_PREFSプロシージャを使用して、保留中の統計を作成するオブジェクトについてパラメータPUBLISHの値をTRUE（デフォルト）からFALSEに変更する必要があります。以下の例では、保留中の統計がSHスキーマのSALES表で有効化され、次にSALES表で統計が収集されます。

```
exec dbms_stats.set_table_prefs('sh','sales','publish','false')
```

オブジェクトで統計を通常どおりに収集します。

```
exec dbms_stats.gather_table_stats('sh','sales')
```

これらのオブジェクトで収集された統計は、USER\_\*\_PENDING\_STATSと呼ばれるディクショナリ・ビューを使用して表示できます。

保留中の統計の使用を有効化するには、ALTER SESSIONコマンドを実行して初期化パラメータOPTIMIZER\_USE\_PENDING\_STATSをTRUEに設定します。保留中の統計を有効にした後は、このセッションで実行されるすべてのSQLワークロードで新たな非公開統計が使用されます。ワークロードでアクセスされる、保留中の統計がない表では、オプティマイザは標準データ・ディクショナリ表の現在の統計を使用します。保留中の統計の検証が終了したら、プロシージャDBMS\_STATS.PUBLISH\_PENDING\_STATSを使用して保留中の統計を公開できます。

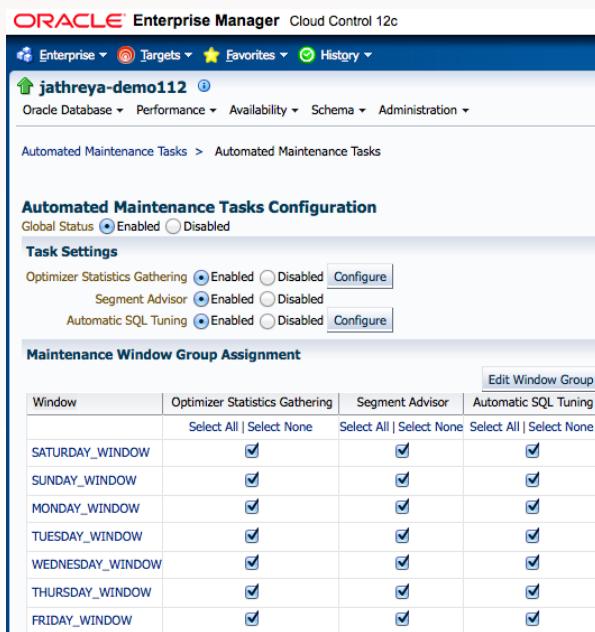
```
exec dbms_stats.publish_pending_stats('sh','sales')
```

## 統計を収集するタイミング

最適な実行計画を選択するには、オプティマイザで代表的な統計を利用する必要があります。代表的な統計は、必ずしも最新の詳細な統計ではなく、実行計画の各操作で求められる適切な行数をオプティマイザが判断できるようにする統計一式です。

### 自動統計収集タスク

オラクルでは、事前定義されたメンテナンス時間枠（平日の午後10時～午前2時、週末の午前6時～午前2時）に、統計がない、または統計が古いすべてのデータベース・オブジェクトで、統計が自動的に収集されます。ジョブが実行されるメンテナンス時間枠は、Enterprise Managerを介して、またはDBMS\_SCHEDULERおよびDBMS\_AUTO\_TASK\_ADMINパッケージを使用して変更できます。



Automated Maintenance Tasks Configuration

Global Status  Enabled  Disabled [Configure](#)

**Task Settings**

Optimizer Statistics Gathering	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled <a href="#">Configure</a>
Segment Advisor	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled <a href="#">Configure</a>
Automatic SQL Tuning	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled <a href="#">Configure</a>

**Maintenance Window Group Assignment**

Window	Optimizer Statistics Gathering	Segment Advisor	Automatic SQL Tuning
SATURDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None
SUNDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None
MONDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None
TUESDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None
WEDNESDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None
THURSDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None
FRIDAY_WINDOW	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None	<input checked="" type="checkbox"/> Select All   Select None

図3：自動統計収集ジョブが実行されるメンテナンス時間枠の変更

十分に確立された統計収集プロシージャがすでにある場合、または何らかの理由で自動統計収集を無効にしたい場合、次のようにタスクをまとめて無効にすることができます。

```
begin
  dbms_auto_task_admin.disable(
    client_name=>'auto optimizer stats collection',
    operation=>null,
    window_name=>null);
end;
/
```

## 手動統計収集

オプティマイザ統計を手動で保守することを計画している場合は、統計を収集するタイミングを決定する必要があります。

自動ジョブの場合と同様に古くなった統計を基に、または新しいデータがいつ環境にロードされたかに基づき、統計を収集するタイミングを決定できます。基本的なデータが大幅に変更されていない場合は、継続的に統計を再収集することは推奨されません。システム・リソースを不必要に浪費することになるためです。

あらかじめ定義されたETLまたはELTジョブの間にのみデータが環境にロードされる場合、統計収集処理はこのプロセスの一環としてスケジューリングできます。オンライン統計収集および増分統計を、統計保守戦略の一環として活用することを試みる必要があります。

### ダイレクト・パス・ロードのためのオンライン統計収集

ダイレクト・パス・ロードのためのオンライン統計収集は、Create Table As Select (CTAS) やInsert As Select (IAS) 操作などのダイレクト・パス・データ・ロード処理の一環として、統計収集を“巧みに利用”します。データ・ロード処理の一環として統計を収集すると、データ全体のスキャンを追加で実行しなくても、データがロードされた直後に統計を利用できるようになります。

```
SQL> Select to_char(sysdate,'DD-MON-YYYY HH24:MI:SS') current_wall_clock_time From dual;
CURRENT_WALL_CLOCK_TIME
05-JUN-2013 11:50:41
SQL>
SQL> -- Do CTAS command
SQL> Create Table SALES2 As Select * From SALES;
Table created.

SQL>
SQL>
SQL> -- Confirm online statistics gathering took place and we have stats on sales2
SQL> Select table_name, num_rows, to_char(last_analyzed, 'DD-MON-YYYY HH24:MI:SS') last_analyzed
  2 From user_tables Where table_name='SALES2';

TABLE_NAME      NUM_ROWS LAST_ANALYZED
SALES2          918843 05-JUN-2013 11:50:43                                ヒストグラムは作成されない

SQL>
SQL> Select column_name, num_distinct, num_nulls, histogram, notes
  2 From user_tab_col_statistics Where table_name='SALES2';

COLUMN_NAME      NUM_DISTINCT  NUM_NULLS HISTOGRAM      NOTES
PROD_ID           72            0  NONE          STATS_ON_LOAD
CUST_ID          7059           0  NONE          STATS_ON_LOAD
TIME_ID          1450           0  NONE          STATS_ON_LOAD
CHANNEL_ID        4             0  NONE          STATS_ON_LOAD
PROMO_ID          4             0  NONE          STATS_ON_LOAD
QUANTITY_SOLD     1             0  NONE          STATS_ON_LOAD
AMOUNT_SOLD       3586           0  NONE          STATS_ON_LOAD
```

図4：新たに作成されたSALES2表で表と列の両方の統計を提供するオンライン統計収集

オンライン統計収集では、ヒストグラムや索引統計は収集されません。これらのタイプの統計には、データ・スキャンが追加で必要であり、データ・ロードのパフォーマンスが大幅に低下する可能性があるためです。基本の列統計情報を再収集せずに、必要なヒストグラムと索引統計を収集するには、新しいオプション・パラメータをGATHER AUTOに設定したDBMS\_STATS.GATHER\_TABLE\_STATSプロシージャを使用します。

パフォーマンスの理由から、GATHER AUTOでは、表のすべての行ではなく、サンプルの行を使用してヒストグラムが構築されることに留意してください。

```
SQL> BEGIN
 2   DBMS_STATS.GATHER_TABLE_STATS('sh','sales2', options=>'GATHER AUTO');
 3 END;
4 /
PL/SQL procedure successfully completed.

SQL> Select column_name, num_distinct, num_nulls, histogram, notes
 2 From user_tab_col_statistics Where table_name='SALES2';

COLUMN_NAME      NUM_DISTINCT  NUM_NULLS HISTOGRAM      NOTES
-----  -----
AMOUNT_SOLD          583          0  NONE          STATS_ON_LOAD
QUANTITY_SOLD         44          0  NONE          STATS_ON_LOAD
PROMO_ID            116          0  NONE          STATS_ON_LOAD
CHANNEL_ID            5          0  NONE          STATS_ON_LOAD
TIME_ID              620          0  HYBRID        HISTOGRAM_ONLY
CUST_ID              630          0  HYBRID        HISTOGRAM_ONLY
PROD_ID              766          0  HYBRID        HISTOGRAM_ONLY
```

図5：オプションをGATHER AUTOに設定することで、基本統計にかかわらず、SALES2表でヒストグラムを作成

列"HISTOGRAM\_ONLY"は、基本の列統計情報を再収集せずに、ヒストグラムが収集されたことを表しています。オンライン統計収集が実行されたことを確認する方法には、実行計画を調べて新しい行ソースOPTIMIZER STATISTICS GATHERINGが計画に表示されていることを確認する方法と、USER\_TAB\_COL\_STATISTICS表の新しいNOTES列でステータスSTATS\_ON\_LOADを確認する方法の2つがあります。

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	CREATE TABLE STATEMENT				672 (100)
1	LOAD AS SELECT				
2	OPTIMIZER STATISTICS GATHERING		254KI	7462KI	387 (17)
3	PARTITION RANGE ALL		254KI	7462KI	387 (17)
4	TABLE ACCESS FULL	SALES	254KI	7462KI	387 (17)

図6：オンライン統計収集処理の実行計画

オンライン統計収集は、ダイレクト・パス・ロード処理がパフォーマンスに及ぼす影響を最小限に抑えるように設計されているため、データが空のオブジェクトにロードされる場合のみ実行されます。既存の表の新しいパーティションにロードする場合にオンライン統計収集が実行されるようにするには、拡張構文を使用してパーティションを明示的に指定します。この場合、パーティション・レベルの統計は作成されますが、グローバル・レベル（表レベル）の統計は更新されません。パーティション表に対して増分統計が有効になっている場合、データ・ロード処理の一部としてシノプシスが作成されます。

オンライン統計収集は、NO\_GATHER\_OPTIMIZER\_STATISTICSヒントを使用して、個々のSQL文に対して無効にすることができます。

## 高頻度オプティマイザ統計収集

このOracle Database 19cの新機能は、特定のOracle Database プラットフォームで利用できます。詳しくは、『Oracle Database ライセンス・ガイド』を参照してください。

### 正確なオプティマイザ統計を維持する上での課題

古い統計により、オプティマイザが誤った判断を下す可能性があり、場合によっては、次善の実行計画が生成される可能性もあります。この問題に対処するために、Oracle 11gで自動オプティマイザ統計収集が導入されました。自動化されたメンテナンス・タスク・インフラストラクチャによって、統計収集がメンテナンス時間枠に行われるようスケジューリングされます。デフォルトでは、1つの時間枠が週の各日にスケジューリングされており、統計収集が事前に定義されたすべてのメンテナンス時間枠で実行されます。

短時間にデータが大幅に変更されるような変化が激しい環境では、最適なSQL実行計画を維持するためには、1日1回の統計収集では不十分な場合があります。

### 高頻度自動統計収集

高頻度自動統計収集は、既存の自動タスクとリアルタイム統計（下記参照）の補完機能として、Oracle Database 19cで新たに導入されました。

自動タスクは、データベースを頻繁に検査して、古い統計があるかどうかを確認します。古い統計が見つかった場合、その統計は更新されます。このプロセスは、メンテナンス時間枠から分離されています。

この機能は以下のように有効化されます。

```
SQL> exec dbms_stats.set_global_prefs('AUTO_TASK_STATUS','ON')          /* 無効化する場合はOFF */
```

DBAは必要と判断した場合、特定のシステムの要件に基づき、検査の頻度と統計収集の最大実行時間（実行ごと）をカスタマイズできます。事前定義されたメンテナンス時間枠で実行されている既存の自動統計収集は影響を受けず、統計がメンテナンス時間枠で収集されている間は高頻度自動収集タスクは起動されないことに留意してください。デフォルトは以下のように調整できます（単位は秒です）。

```
SQL> exec dbms_stats.set_global_prefs('AUTO_TASK_INTERVAL','900')          /* This is the default */  
SQL> exec dbms_stats.set_global_prefs('AUTO_TASK_MAX_RUN_TIME','3600') /* This is the default */
```

現在の設定は以下のように確認できます。

```
SQL> select dbms_stats.get_prefs('AUTO_TASK_STATUS') from dual;  
SQL> select dbms_stats.get_prefs('AUTO_TASK_INTERVAL') from dual;  
SQL> select dbms_stats.get_prefs('AUTO_TASK_MAX_RUN_TIME') from dual;
```

メンテナンス時間枠に起動される自動統計収集ジョブとは異なり、高頻度タスクは古いオブジェクトの統計のみを収集します。他の補助的なアクション（統計アドバイザの呼出しなど）は、メンテナンス時間枠に行われます。

実行統計は以下のDBAビューで利用できます。

```
select start_time, end_time, failed, timed_out  
from    DBA_AUTO_STAT_EXECUTIONS  
where   origin = 'HIGH_FREQ_AUTO_TASK'  
order by start_time;
```

## リアルタイム統計

このOracle Database 19cの新機能は、特定のOracle Database プラットフォームで利用できます。詳しくは、『Oracle Database ライセンス・ガイド』を参照してください。

正確なオプティマイザ統計を維持するまでの課題

上記で述べたように、古い統計によって次善のSQL実行計画が生成される可能性がありますが、極めて頻繁に変更されるシステムで正確な統計を維持することは容易ではありません。高頻度統計収集はこの問題を解決するのに有効ですが、より理想的な解決策は、データベースのデータが変更されたときに統計を保守することです。

### リアルタイム統計

リアルタイム統計では、統計収集技法を従来型のDML操作 (INSERT、UPDATE、MERGE) に拡張します。これらのDML操作がデータベースのデータに実行されると、もっとも基本的なオプティマイザ統計がリアルタイムで保守されます。これは、個々の行操作と一括操作の両方に適用されます。

リアルタイム統計を利用すると、自動統計収集ジョブや高頻度統計収集によって収集される統計や、DBMS\_STATS APIを使用して手作業で収集される統計が補強されます。データベースのデータの正確な状態が常に保守されるため、より最適なSQL実行計画が生成されるようになります。

リアルタイム統計は自動的に管理され、データベース管理者の介入は不要です。開発者は、NO\_GATHER\_OPTIMIZER\_STATISTICSヒントを使用して、個々のSQL文に対してオンライン統計収集を無効にすることを選択できます。

リアルタイム統計は以下のように確認できます。

```
select table_name, num_rows, blocks
  from user_tab_statistics
 where notes = 'STATS_ON_CONVENTIONAL_DML';

select table_name, column_name, low_value, high_value
  from user_tab_col_statistics
 where notes = 'STATS_ON_CONVENTIONAL_DML';
```

リアルタイム統計はリアルタイムに計算され、値は定期的にディスクに保持されることに留意してください。値がデータ・ディクショナリ・ビューに表示されるまでに数分かかる場合があります。

リアルタイム統計を使用していることが、次のSQL実行計画で示されています。

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR(format=>'TYPICAL'));
```

```
PLAN_TABLE_OUTPUT
```

```
SQL_ID 9uq8aaahp0b3u, child number 0
```

```
select count(*) from load where a < 10
```

```
Plan hash value:3743925786
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELET STATEMENT				7 (100)	
1	SORT AGGREGATE		1	6		
2	TABLE ACCESS FULL	LOAD	10000	60000	7 (0)	00:00:01

```
Predicate Information (identified by operation id):
```

```
2 - filter("A"<10)
```

```
Note
```

```
- dynamic statistics used: statistics for conventional DML
```

## 増分統計とパーティション交換データ・ロード

パーティション表での統計の収集は、表レベルでの収集（グローバル統計）と（サブ）パーティション・レベルでの収集で構成されています。パーティション表のINCREMENTAL<sup>3</sup>プリファレンスがTRUEに設定され、DBMS\_STATS.GATHER\_\*\_STATSのパラメータGRANULARITYがGLOBALに設定され、ESTIMATE\_PERCENTがAUTO\_SAMPLE\_SIZEに設定されている場合、表全体ではなく、追加または変更のあったパーティションのみをスキャンすることで、グローバル・レベルのすべての統計が正確に抽出されます。増分グローバル統計は、表内の各パーティションのシノプシスを保存することで機能します。シノプシスは、そのパーティションおよびパーティション内の列の統計メタデータであり、パーティション・レベルの統計と各パーティションのシノプシスを集計することで、グローバル・レベルの統計が正確に生成されるため、表全体をスキャンする必要がありません。表に新しいパーティションが追加された場合は、新しいパーティションの統計を収集するだけです。新しいパーティションのシノプシスと既存のパーティションのシノプシスを使用して、表レベルの統計が自動的かつ正確に算出されます。

増分統計が有効な場合、パーティション統計はサブパーティション統計から集計されないことに注意してください。

パーティション交換ロードを使用しながら、増分統計を活用したいと考えている場合、非パーティション表におけるDBMS\_STATS表プリファレンスのINCREMENTAL\_LEVELを、パーティション交換ロードで使用するように設定する必要があります。

INCREMENTAL\_LEVELをTABLE（デフォルトはPARTITION）に設定することで、表で統計が収集された際にその表のシノプシスが自動的に作成されます。この表レベルのシノプシスは、交換後にパーティション・レベルのシノプシスになります。

ただし、トリクル・フィードや少數の行のみを挿入するオンライン・トランザクションが多く、そのような処理が1日中発生するような環境では、統計が古くなるタイミングを判断して自動統計収集タスクをトリガーする必要があります。USER\_TAB\_STATISTICSのSTALE\_STATS列を使用して、統計が古いかどうかを判断することを計画している場合、この情報は1日に一度のみ更新されることに留意する必要があります。表でどのようなDMLが発生しているかについて、よりタイムリーな情報が必要な場合は、USER\_TAB\_MODIFICATIONSを確認する必要があります。ここには、各表で発生したINSERTS、UPDATES、およびDELETESの数と、表が切り捨てられているかどうか（TRUNCATED列）が表示されるため、統計が古いかどうかをユーザー自身で判断できます。繰り返しになりますが、この情報はメモリから定期的に自動更新されることに注意してください。最新の情報が必要な場合は、DBMS\_STATS.FLUSH\_DATABASE\_MONITORING\_INFO関数を使用して、情報を手動でフラッシュする必要があります。

<sup>3</sup> 詳細については、このホワイト・ペーパー・シリーズの第1部『Oracle Database 19cのオプティマイザ統計について』を参照してください。

## "範囲外"の状態の回避

自動統計収集タスクを使用するか、手動で統計を収集するかにかかわらず、エンドユーザーが新しく挿入されたデータの問合せを統計の収集前に開始すると、表で変更された行が10 %未満であっても、統計が古いために、最適でない実行計画を取得する可能性があります。もっとも一般的な例の1つは、where句の条件に指定されている値が、列統計情報の最大値と最小値で表される値の範囲外となる場合です。これは、'範囲外'エラーと呼ばれています。この場合、オプティマイザは条件の値と最大値の差分に基づいて選択性を割り当てます（値が最大値よりも大きいと仮定した場合）。つまり、値が最大値または最小値から離れているほど、選択性は少なくなります。

このような例は、レンジ・パーティション表では非常に一般的です。新しいパーティションは既存のレンジ・パーティション表に追加され、行はそのパーティションにのみ挿入されます。エンドユーザーは、この新しいパーティションで統計が収集される前に、この新しいデータの問合せを開始します。パーティション表では、DBMS\_STATS.COPY\_TABLE\_STATS<sup>4</sup>プロシージャ（Oracle Database 10.2.0.4以降で利用可能）を使用して、"範囲外"の状態を回避できます。このプロシージャは、代表的なソース（サブ）パーティションの統計を、新しく作成された空の宛先（サブ）パーティションにコピーします。また、列、ローカル（パーティション）索引などの依存オブジェクトの統計もコピーし、パーティション値の上限をパーティション列の最大値に設定し、前のパーティションのパーティション値の上限をパーティション列の最小値に設定します。コピーされた統計は、パーティションの正確な統計を取得できるようになるまでの一時的な解決策と見なされる必要があります。統計のコピーを実際の統計収集の代わりに使用しないでください。

デフォルトでは、DBMS\_STATS.COPY\_TABLE\_STATSにより、パーティション統計のみが調整され、グローバル・レベルまたは表レベルの統計は調整されません。コピーの一環としてパーティション列に対するグローバル・レベルの統計を更新したい場合は、DBMS\_STATS.COPY\_TABLE\_STATSのフラグ・パラメータを8に設定する必要があります。

非パーティション表では、DBMS\_STATS.SET\_COLUMN\_STATSプロシージャを使用して列の最大値を手動で設定できます。この方法は一般的には推奨されず、実際の統計収集の代わりとなるものではありません。

'リアルタイム統計'では、範囲外のシナリオに遭遇するリスクが低減されます（13ページを参照）。

4 『Oracle Database 19cのオプティマイザ統計について』を参照してください。

## オプティマイザ統計の品質確保

質の高い統計は、最適なSQL実行計画の生成に不可欠ですが、統計の質が低いもののそのことに気づかない場合もあります。たとえば、古い“継承された”システムで使用されているスクリプトを、データベース管理者がもはや理解できないために、変更することを当然ながら躊躇しているような場合です。オラクルでは統計収集機能が継続的に改良されるため、そのようなシステムではベスト・プラクティスの推奨事項が遂行されない可能性があります。

Oracle Database 19cには、このような理由から、データベースにおける統計の品質向上を支援する、オプティマイザ統計アドバイザと呼ばれるアドバイザが搭載されています。この診断ソフトウェアは、データ・ディクショナリの情報を分析し、統計の品質を評価し、統計の収集方法を発見します。質の低い統計や欠けている統計を報告し、これらの問題を解決するための推奨事項を生成します。

この処理の背後にある原理は、ベスト・プラクティスのルールを適用して潜在的な問題を明らかにすることです。これらの問題は、一連の結果として報告され、その後、具体的な推奨事項を導く可能性があります。推奨事項は、アクションを使用して（即座に、またはデータベース管理者が自動生成スクリプトを実行することで）自動的に実装できます。



図7：オプティマイザ統計アドバイザ

アドバイザのタスクは、メンテナンス時間枠に自動的に実行されますが、オンデマンドで実行することもできます。アドバイザによって生成されたHTMLまたはテキストのレポートは、いつでも参照でき、アクションもいつでも実装できます。

以下の図8では、結果を導く具体的なルール、推奨事項、および問題を解決するためのアクションを例示しています。



図8：ルール、結果、推奨事項、アクションの例

アドバイザのタスクでは、データが収集され、データ・ディクショナリに保存されます。オプティマイザ統計と統計収集情報（すでにデータ・ディクショナリに保存済み）の分析が実行されるため、これはパフォーマンスのオーバーヘッドが低い処理です。アプリケーション・スキーマ・オブジェクトに保存されたデータのセカンダリ分析は行われません。



図9：データ・ディクショナリの読み取り、フィルタを使用したタスクの実行、および結果の保存

タスクが完了すると、レポートをHTMLまたはテキスト形式で生成でき、アクション（SQL）スクリプトも作成できます。

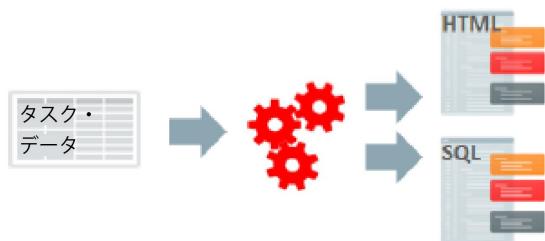


図10：アドバイザ・タスクのレポート作成、およびアクションSQLスクリプトの生成

自動化されたタスクによって生成されるレポートは、以下のように簡単に参照できます。

```
select dbms_stats.report_advisor_task('auto_stats_advisor_task') as report from dual;
```

また、ADVISOR権限を持つユーザーがタスクを手動で実行し、以下の3つのステップから成るプロセスを使用して結果のレポートを作成することもできます。

```
DECLARE
  tname  VARCHAR2(32767) := 'demo';  -- task name
BEGIN
  tname := dbms_stats.create_advisor_task(tname);
END;
/
DECLARE
  tname  VARCHAR2(32767) := 'demo';  -- task name
  ename VARCHAR2(32767) := NULL;      -- execute name
BEGIN
  ename := dbms_stats.execute_advisor_task(tname);
END;
/
SELECT dbms_stats.report_advisor_task('demo') AS report
FROM dual;
```

アドバイザによって生成されたアクションは、即座に実装できます。

```
DECLARE
  tname      VARCHAR2 (32767) := 'demo';-- タスク名
  impl_result CLOB;                  -- 実装をレポート
BEGIN
  impl_result := dbms_stats.implement_advisor_task(tname);
END;
/
```

さらに、Oracle Database 19c Real Application Testingには、SQL Performance Advisor Quick Checkをはじめとする有用なパフォーマンス保証機能が搭載されています。

## 統計のより迅速な収集

データ量が増加し、メンテナンス時間枠が短縮される中で、統計をタイムリーに収集することがこれまで以上に重要になっています。オラクルでは、統計収集処理を並列処理する、統計を収集する代わりに生成するなど、統計収集を加速させるさまざまな方法を提供しています。

### 並列処理の使用

統計収集では、並列処理を複数の方法で活用できます。

- DEGREEパラメータの使用
- 同時統計収集
- DEGREEパラメータと同時統計収集の組み合わせ

DEGREEパラメータの使用

DBMS\_STATSのDEGREEパラメータは、統計を収集するために使用されるパラレル実行プロセスの数を制御します。

デフォルトでは、データ・ディクショナリ内の表の属性（並列度）で指定された数と同じ数のパラレル・サーバー・プロセスが使用されます。Oracle Databaseのすべての表では、この属性がデフォルトで1に設定されています。大規模な表での統計収集では、統計収集を加速させるために、このパラメータを明示的に設定すると良いでしょう。

また、DEGREEをAUTO\_DEGREEに設定することもできます。そうすることで、オブジェクトのサイズを基に、統計の収集に使用されるパラレル・サーバー・プロセスの適切な数が自動的に決定されます。プロセスの数は、小規模なオブジェクトの場合の1（シリアル実行）から、大規模なオブジェクトの場合のDEFAULT\_DEGREE (PARALLEL\_THREADS\_PER\_CPU \* CPU\_COUNT) までとなります。

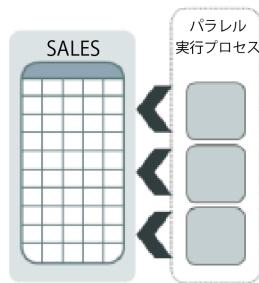


図11：DEGREEパラメータによる並列処理の使用

パーティション表でDEGREEを設定すると、各パーティションで統計を収集するために複数のパラレル・サーバー・プロセスが使用されますが、統計は異なるパーティションで同時に収集されるわけではないことに注意する必要があります。統計は、各パーティションで順番に収集されます。

### 同時統計収集

同時統計収集により、スキーマ（またはデータベース）内の複数の表、および表内の複数の（サブ）パーティションで統計を同時に収集できます。複数の表と（サブ）パーティションで統計を同時に収集すると、マルチプロセッサ環境を十分に活用できるため、全体的な統計収集時間を短縮できます。

同時統計収集は、グローバル・プリファレンスのCONCURRENTによって制御されます。このプリファレンスは、MANUAL、AUTOMATIC、ALL、OFFに設定できます。デフォルトでは、OFFに設定されています。CONCURRENTが有効な場合、Oracle Job SchedulerコンポーネントとAdvanced Queuingコンポーネントを使用して、複数の統計収集ジョブが同時に作成され、管理されます。

CONCURRENTがMANUALまたはALLに設定されている場合にパーティション表でDBMS\_STATS.GATHER\_TABLE\_STATSを呼び出すと、表内の（サブ）パーティションごとに別々の統計収集ジョブが作成されます。同時に実行できるジョブの数と、キューに入れることができるジョブの数は、使用可能なジョブ・キュー・プロセス（RAC環境のノードごとのJOB\_QUEUE\_PROCESSES初期化パラメータ）の数と、使用可能なシステム・リソースに基づきます。現在実行中のジョブが終了すると、すべての（サブ）パーティションで統計が収集されるまで、さらに多くのジョブがデキューされ、実行されます。

DBMS\_STATS.GATHER\_DATABASE\_STATS、DBMS\_STATS.GATHER\_SCHEMA\_STATS、またはDBMS\_STATS.GATHER\_DICTIONARY\_STATSを使用して統計を収集すると、非パーティション表と、パーティション表の（サブ）パーティションごとに、別の統計収集ジョブが作成されます。各パーティション表には、その（サブ）パーティション・ジョブを管理するコーディネータ・ジョブもあります。データベースでは、可能な限り多くの同時ジョブが実行され、実行中のジョブが終了するまで、残りのジョブはキューに入れられます。ただし、デッドロック発生の可能性を避けるため、複数のパーティション表を同時に処理することはありません。そのため、1つのパーティション表に対して複数のジョブが実行されている場合、スキーマ（またはデータベース、またはディクショナリ）内の別のパーティション表は、現在のジョブが終了するまで、キューに入れられます。非パーティション表には、このような制約はありません。

表12では、SHスキーマでDBMS\_STATS.GATHER\_SCHEMA\_STATSコマンドが発行されている場合に、異なるレベルでジョブを作成する方法を示します。以下の非パーティション表ごとに、統計収集ジョブ（図12のレベル1）が作成されます。

- CHANNELS
- COUNTRIES
- TIMES

各パーティション表に対して、コーディネータ・ジョブが作成されます。また、SALESおよびCOSTSでは、SALES表とCOSTS表のパーティションごとに、統計収集ジョブが作成されます（図12のレベル2）。

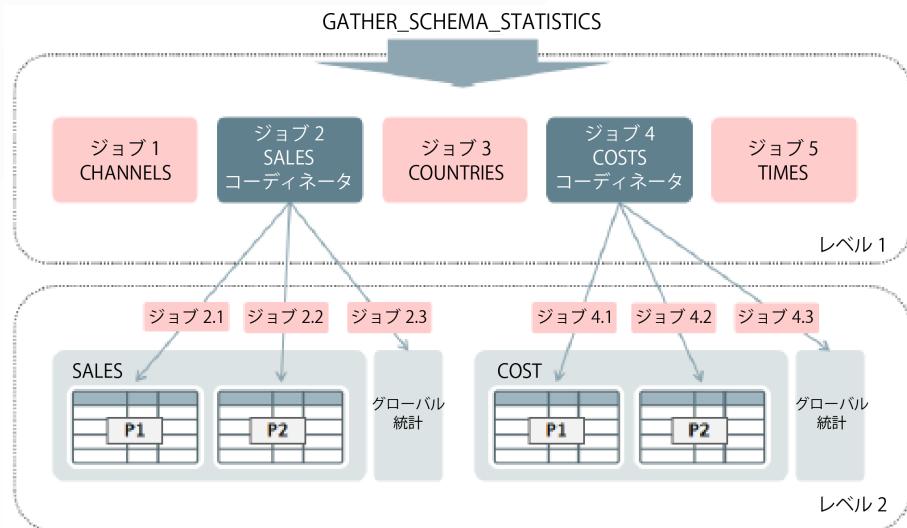


図12：SHスキーマで同時統計収集が発生した際に作成された統計収集ジョブの一覧

それぞれの統計収集ジョブでは、DEGREEパラメータが指定されていれば、パラレル実行も活用できます。

表、パーティション、またはサブパーティションが非常に小さい場合や空の場合、ジョブ・メンテナンスのオーバーヘッドを低減するため、データベースによってオブジェクトを他の小さいオブジェクトとともに1つのジョブに自動的にまとめます。

### 同時統計収集の構成

統計収集の同時実行性は、デフォルトでオフに設定されていますが、次のようにオンにすることができます。

```
exec dbms_stats.set_global_prefs('concurrent', 'all')
```

ただし、統計の収集に必要な通常の権限の他に、さらなる権限も必要となります。ユーザーには、以下のジョブ・スケジューラ権限とAQ権限が必要です。

- CREATE JOB
- MANAGE SCHEDULER
- MANAGE ANY QUEUE

ジョブ・スケジューラはその内部表とビューをSYSAUX表領域に保存するため、SYSAUX表領域はオンラインでなければなりません。最後に、統計収集プロセスが使用できる（または統計収集プロセスに割り当てられた）すべてのシステム・リソースを十分に活用できるように、JOB\_QUEUE\_PROCESSES/パラメータを設定する必要があります。パラレル実行を使用する予定がない場合は、JOB\_QUEUE\_PROCESSESを、2 x CPUコアの合計数に設定する必要があります（これは、RAC環境におけるノードあたりのパラメータです）。このパラメータは、セッション・レベル (ALTER SESSION) ではなく、システム全体で (ALTER SYSTEM、またはinit.oraファイル内で) 設定してください。

同時統計収集の一環としてパラレル実行を使用する場合、次のように、パラレル適応のマルチ・ユーザーを無効にする必要があります。

```
ALTER SYSTEM SET parallel_adaptive_multi_user=false;
```

リソース・マネージャは、たとえば以下のように有効にする必要があります。

```
ALTER SYSTEM SET resource_manager_plan = 'DEFAULT_PLAN';
```

パラレル文キューイングを有効にすることも推奨されます。これには、リソース・マネージャが有効化され、コンシューマ・グループ"OTHER\_GROUPS"にキューイングがある一時リソース・プランの作成が有効化されている必要があります。デフォルトでは、リソース・マネージャはメンテナンス時間枠でのみ有効化されます。以下のスクリプトは、一時リソース・プラン (pqq\_test) を作成し、そのプランでリソース・マネージャを有効にする方法の一例です。

```
-- dba権限のあるユーザーとして接続 begin
  dbms_resource_manager.create_pending_area();
  dbms_resource_manager.create_plan('pqq_test', 'pqq_test');
  dbms_resource_manager.create_plan_directive(
    'pqq_test', 'OTHER_GROUPS',
    'OTHER_GROUPS directive for pqq',
    parallel_target_percentage => 90);
  dbms_resource_manager.submit_pending_area(); end;
/
ALTER SYSTEM SET RESOURCE_MANAGER_PLAN = 'pqq_test' SID='*';
```

自動統計収集タスクで同時実行性を活用したい場合は、CONCURRENTをAUTOMATICまたはALLのいずれかに設定します。同時統計収集によってシステム・リソースが大量に消費されないようにするために、新しいORA\$AUTOTASKコンシューマ・グループが、メンテナンス時間枠に使用されるリソース・マネージャ・プランに追加されています。

## 統計を収集すべきでない場合

最適な計画を選択するには、オプティマイザが正確な統計を利用できる必要がありますが、統計の収集が困難である、コストがかかりすぎる、またはタイムリーに行うことができないために、代わりの戦略が必要となる場合もあります。

### 頻繁に変更される表

頻繁に変更される表とは、データ量が時間とともに大幅に変更する表です。朝は空である注文キューの表を例に挙げます。時間がたち、注文が行われるようになると、表は埋まり始めます。それぞれの注文は処理されると、表から削除されるため、一日の終わりには表は再び空になります。自動統計収集ジョブを使用して、そのような表の統計を保守する場合、ジョブが実行される夜間は表が空のため、統計では表は常に空となります。しかしながら、日中には表には数十万の行があるかもしれません。

そのような使用例では、代わりに高頻度統計収集（上記参照）の利用を選択するといいでしょう（プラットフォームとデータベースのバージョンが高頻度統計収集をサポートしている場合）。あるいは、表にデータが入力されている日中にその表の統計の代理セットを収集し、それをロックするとよい場合も多くあります。統計をロックすると、高頻度統計収集と自動統計収集ジョブがその統計を上書きできなくなります。この方法は、一日中行数が激しく増減するような頻繁に変更される表にとりわけ適しています。

代わりに、動的サンプリングを使用してそのような表で統計を収集することもできます。オプティマイザは、SQL文のコンパイル中に動的サンプリングを使用して表の基本的な統計を収集した後に、そのSQL文を最適化します。動的サンプリングによって収集した統計は、DBMS\_STATSパッケージを使用して収集した統計ほど高品質ではなく、完全ではありませんが、多くの場合はこの統計で十分です。

## グローバル時表

グローバル時表は、多くの場合、アプリケーション・コンテキストに中間結果を保存するために使用されます。グローバル時表 (GTT) は、システム全体の定義を適切な権限を持つすべてのユーザーと共有しますが、データ・コンテンツは常にセッション・プライベートとなります。表にデータが挿入されてない限り、物理ストレージは割り当てられません。グローバル時表は、トランザクション固有 (コミットした行を削除) か、セッション固有 (コミットした行を保持) のいずれかです。トランザクション固有の表で統計を収集すると、表の切り捨てにつながります。一方、(行が保持される) グローバル時表で統計を収集することは可能ですが、前リリースでは、それが常に得策というわけではありませんでした。GTTを使用するすべてのセッションが単一セットの統計を共有しなければならず、多くのシステムが動的統計に依存していました。

しかしながら、Oracle Database 19cでは、GTTを使用するセッションごとに別々の統計セットを使用できます。GTTで共有される統計は、新しいDBMS\_STATSのプリファレンスGLOBAL\_TEMP\_TABLE\_STATSを使用して制御されます。デフォルトでは、このプリファレンスはSESSIONに設定され、GTTにアクセスするセッションごとに独自の統計セットを使用できます。オプティマイザは、まず、セッション統計を使用するよう試みますが、セッション統計が存在しない場合、共有の統計を使用します。

```
SQL> -- Create Global Temporary Table
SQL> Create Global Temporary Table TG (col1 number);
Table created.

SQL> -- get table preference for TG
SQL> select dbms_stats.get_prefs('GLOBAL_TEMP_TABLE_STATS','SH','TG') from dual;
DBMS_STATS.GET_PREFS('GLOBAL_TEMP_TABLE_STATS','SH','TG')
-----
SESSION

SQL> --Change table preference for TG to SHARED
SQL> BEGIN
 2   dbms_stats.set_table_prefs('SH','TG','GLOBAL_TEMP_TABLE_STATS','SHARED');
 3 END;
 4 /
PL/SQL procedure successfully completed.

SQL> -- get table preference for TG
SQL> select dbms_stats.get_prefs('GLOBAL_TEMP_TABLE_STATS','SH','TG') from dual;
DBMS_STATS.GET_PREFS('GLOBAL_TEMP_TABLE_STATS','SH','TG')
-----
SHARED
```

図13：GTTで統計を共有しないというデフォルト動作を、統計の共有を適用するように変更

Oracle Database 11gからアップグレードし、データベース・アプリケーションがGTTのSESSION統計を活用するように変更されていない場合、DBMS\_STATSのプリファレンスGLOBAL\_TEMP\_TABLE\_STATSをSHAREDに設定することで、GTTの動作をアップグレード前の環境と同じままにしておくことができます（少なくともアプリケーションがアップデートされるまでは）。

ダイレクト・パス処理を使用してGTT（コミットした行が保持される）を生成する場合、オンライン統計収集のためにセッション・レベル統計が自動的に作成されるため、追加の統計収集コマンドを実行せずに済み、他のセッションによって使用される統計にも影響しません。

```
SQL> Create global temporary Table SALES2(
  2  PROD_ID      NUMBER(6),
  3  CUST_ID      NUMBER,
  4  TIME_ID      DATE,
  5  CHANNEL_ID   CHAR(1),
  6  PROMO_ID     NUMBER(6),
  7  QUANTITY_SOLD NUMBER(3),
  8  AMOUNT_SOLD  NUMBER(10,2));
Table created.

SQL>
SQL> insert /*+ APPEND */ into sales2 select * from sales;
254720 rows created.

SQL> commit;
Commit complete.

SQL>
SQL> Select column_name, num_distinct, num_nulls
  2  From user_tab_col_statistics Where table_name='SALES2';
COLUMN_NAME      NUM_DISTINCT  NUM_NULLS
PROD_ID          766           0
CUST_ID          630           0
TIME_ID          620           0
CHANNEL_ID       5              0
PROMO_ID         116           0
QUANTITY_SOLD    44             0
AMOUNT_SOLD      583           0
```

図14：ダイレクト・パス処理を使用してGTTを生成した結果、セッション・レベル統計を自動的に収集

## 中間作業表

中間作業表は、通常はELTプロセスまたは複雑なトランザクションの一部として使用されます。中間作業表は、書き込みや読み取りが1度のみ行われ、その後、切り捨てられるか削除されます。そのように統計が1度しか使用されない場合、統計収集のコストがその利点を上回ってしまいます。そのような場合は、代わりに動的サンプリングを使用する必要があります。永続的な中間作業表で統計をロックし、自動統計収集タスクが中間作業表の統計を収集しないようにすることが推奨されます。

## その他のタイプの統計の収集

現在、コストベース・オプティマイザがサポートされる唯一のオプティマイザであるため、すべてのディクショナリ表（SYS、SYSTEMなどが所有し、システム表領域やSYSAUX表領域に格納されている表）や、動的v\$パフォーマンス・ビューによって使用されるx\$表を含め、データベースのすべての表では統計が必要です。

### ディクショナリ統計

ディクショナリ表の統計は、夜間のメンテナンス時間枠に実行される自動統計収集タスクによって保守されます。おもなアプリケーション・スキーマに対して自動統計収集ジョブをオフにする場合でも、自動統計収集タスクによってディクショナリ統計が保守されるようにすることが推奨されます。これを行うには、プロシージャDBMS\_STATS.SET\_GLOBAL\_PREFSを使用してAUTOSTATS\_TARGETの値をAUTOからORACLEに変更します。

```
exec dbms_stats.set_global_prefs('autostats_target','oracle')
```

## 固定オブジェクト統計

固定オブジェクト統計は、それまで収集されていない場合は、自動統計収集タスクによって収集されます。また、データベースは固定オブジェクト統計を収集しません。他のデータベース表とは異なり、オプティマイザ統計が見つからない場合、X\$表が関係しているSQL文に動的サンプリングは自動的に使用されません。そのような場合は、オプティマイザは事前定義されたデフォルト値を統計に使用します。これらのデフォルト値は代表的な値ではないため、実行計画が最適化されずに、システムで重大なパフォーマンスの問題が発生する可能性があります。固定オブジェクト統計を手動で収集することを強く推奨しているのはそのためです。

固定オブジェクト統計は、DBMS\_STATS.GATHER\_FIXED\_OBJECTS\_STATSプロシージャを使用して収集できます。x\$表は本質的に一時的なものであるため、システムに代表的なワークロードが存在する場合は固定オブジェクト統計を収集することが重要です。ただし、大規模なシステムでは、統計を収集するためにさらなるリソースが必要となるため、これが常に可能であるとは限りません。ピーク・ロード時にこの操作を行うことができない場合は、システムのウォームアップが完了し、次の3つのタイプの固定オブジェクト表にデータが挿入された後に、この操作を行う必要があります。

- 構造化データ - データファイルを網羅するビュー、制御ファイルのコンテンツなど。
- セッション・ベースのデータ - v\$session、v\$accessなど。
- ワークロード・データ - v\$sql、v\$sql\_planなど。

主要なデータベースやアプリケーションの大幅なアップグレードを行う場合、新しいモジュールを実装する場合や、データベース構成を変更する場合は、固定オブジェクト統計を再収集することを推奨します。たとえば、SGAサイズを増加させると、v\$buffer\_poolやv\$shared\_pool\_adviceで使用されるx\$表など、バッファ・キャッシュと共有プールの情報が含まれるすべてのx\$表が大幅に変化する可能性があります。

## システム統計

システム統計により、オプティマイザは、文を実行する実際のシステム・ハードウェアの情報（CPU速度やIOパフォーマンスなど）を使用して実行計画を見積もることができます。

システム統計はデフォルト値を使用して自動的に初期化されます。この仕組みはほとんどのシステムに適しています。この仕組みを利用せずにシステム統計を収集することは推奨されません（DBMS\_STATS.GATHER\_SYSTEM\_STATS APIを使用する必要はありません）。

## 結論

Oracleオプティマイザが実行計画のコストを正確に判断するためには、SQL文でアクセスするすべてのオブジェクト（表と索引）に関する正確な統計と、SQL文を実行するシステムに関する情報をオプティマイザが利用できる必要があります。この2部構成のホワイト・ペーパー・シリーズでは、どのようなオプティマイザ統計が必要であるか、どのようにオプティマイザ統計を使用するか、そして利用できる各種統計収集方法について詳しく説明しています。

自動統計収集タスクと、このホワイト・ペーパーで取り上げた他の技法を組み合わせて使用することで、DBAは自身の環境の正確な統計セットを保守することができ、オプティマイザは最適な計画を選択するために必要な情報を常に利用できます。統計収集戦略の導入後は、戦略の変更は制御された方法で行う必要があります。また、アプリケーションのパフォーマンスに悪影響が及ばないようにするために、統計の保留といった主要な機能を活用することが必要です。

## 参考資料

1. Oracleホワイト・ペーパー：*Oracle Database 19cのオプティマイザ統計について*
2. Oracleホワイト・ペーパー：*Oracle Database 19cのオプティマイザ統計の収集と最適化*

## ORACLE CORPORATION

### Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

### 海外からのお問い合わせ窓口

電話 + 1.650.506.7000+ 1.800.ORACLE1

FAX + 1.650.506.7200  
oracle.com

### CONNECT WITH US

+1.800.ORACLE1までご連絡いただくな、[oracle.com](http://oracle.com)をご覧ください。北米以外の地域では、[oracle.com/contact](http://oracle.com/contact)で最寄りの営業所をご確認いただけます。

 [blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)

 [facebook.com/oracle](http://facebook.com/oracle)

 [twitter.com/oracle](http://twitter.com/oracle)

### Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による默示的保証を含め、商品性ないし特定目的適合性に関する默示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。このデバイスは、連邦通信委員会の規則によって義務付けられている認可を受けていません。認可を受けるまでは、このデバイスの販売またはリースを提案することも、このデバイスを販売またはリースすることもありません。（この連邦通信委員会の免責事項は必要でない場合があります。詳細については、2ページの免責事項の項を参照してください。）

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。1219

ホワイト・ペーパー Oracle Database 19cにおけるオプティマイザ統計収集のベスト・プラクティス Oracle Database 19cのSQL計画管理Oracle Database 19cのSQL計画管理

2019年12月 2019年12月 著者：[適宜入力]  
共著者：[適宜入力]