

# Oracle DBA & Developer Days 2011

日本オラクル、今年最大の技術トレーニングイベント

2011年11月9日(水)～11月11日(金) シェラトン都ホテル東京



## ORACLE®

### 実践！コストベース・オプティマイザとの賢い付き合い方

日本オラクル株式会社 テクノロジー製品事業統括本部 ソリューション本部  
エンジニア 辻研一郎

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して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。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

# 本セッションの内容

CBOを前提とした日々の運用

- ・デフォルトでの自動統計情報収集の内容と注意点
- ・新しい実行計画の固定方法

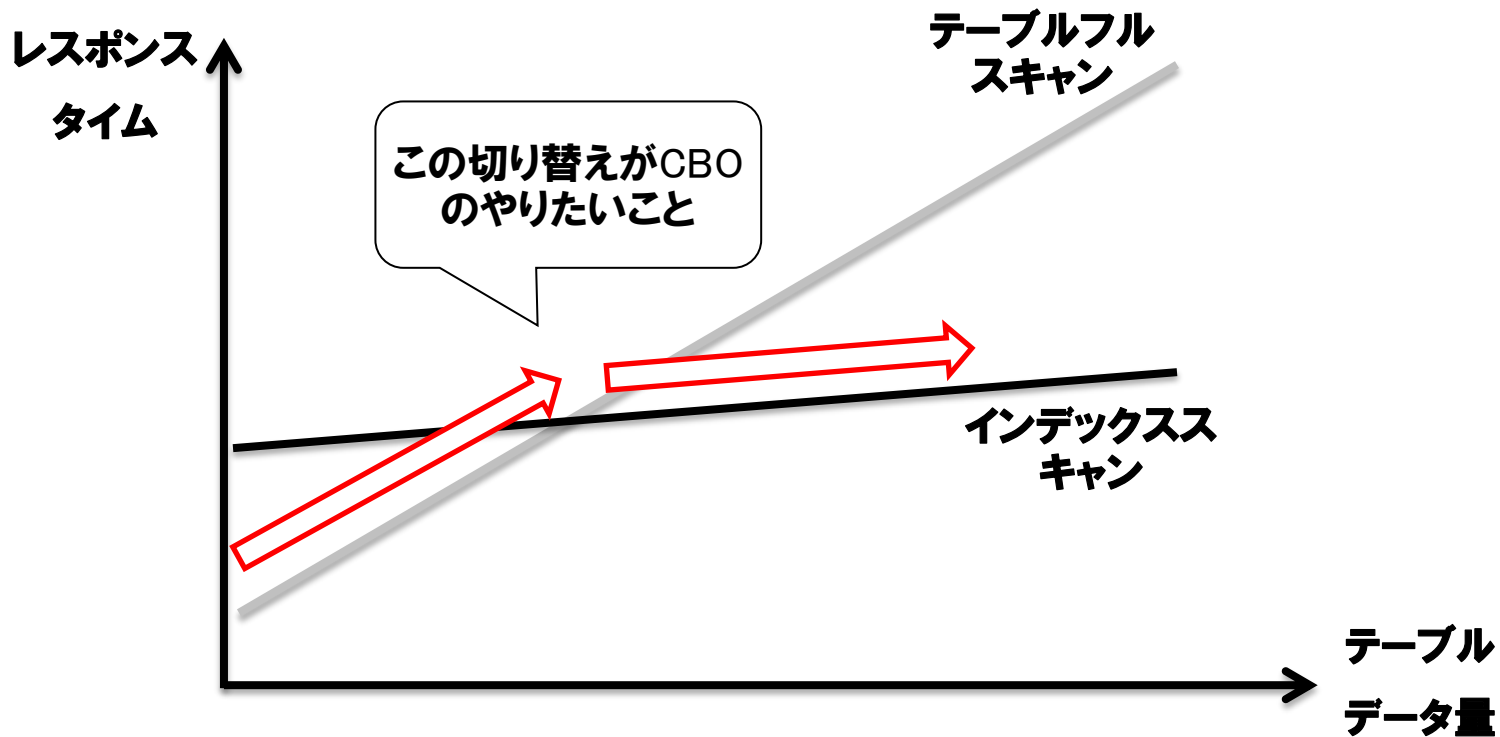
トラブル！！

トラブル発生時の対応

- ・再現情報の確保
- ・早期問題解決

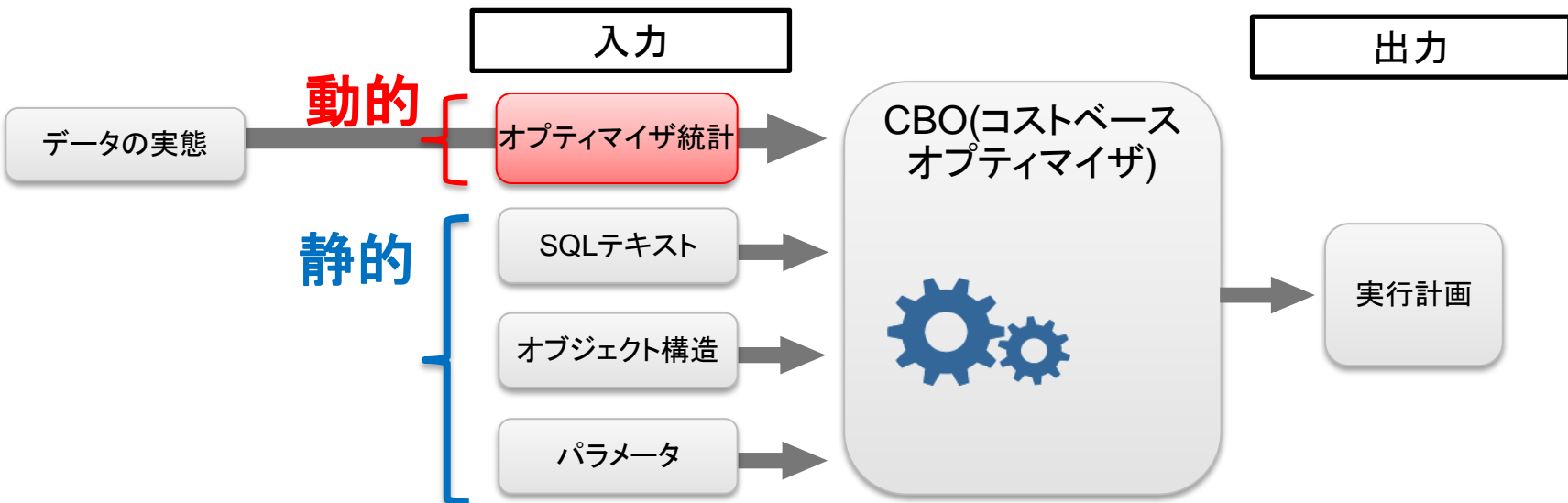
# CBOの役割

- ・ CBOの主な役割はデータ量や環境の変化に応じて最適な実行計画を生成すること



# オプティマイザ統計が重要な理由

- ・ CBOが実行計画生成に使用する入力データは、図の4つです。その内、オプティマイザ統計以外は固定的であることが多いため、**オプティマイザ統計の変動が実行計画の変化に及ぼす影響が大きい**と言えます。
- ・ オプティマイザ統計
  - ・ 実データ情報を認識し、データ変動に応じて適切な実行計画を生成するための情報



# デフォルトでのオプティマイザ統計情報収集

- 自動オプティマイザ統計情報収集
  - 実データの変動具合をOracleが適切に認識するため、定期的にオプティマイザ統計を新しい情報に自動更新する機能
  - 未収集であったり、変更行が一定以上のオブジェクトだけを収集対象にする。また、インデックス、パーティションおよびヒストグラムの取得もOracleが自動で行うためとり逃しを防止

	自動オプティマイザ統計情報収集(DBMS_STATS)	Analyze
定期的な自動実行	される(※後スライド参照)	されない
自動バックアップ	される(デフォルト31日間)	されない
一定以上更新があったオブジェクトのみ取得	可能(デフォルト10%以上の更新)	不可能

※自動オプティマイザ統計情報収集、主な機能のAnalyzeとの比較

# 自動オプティマイザ統計情報収集の注意点(1/2)

## デフォルトの自動収集時間帯とバッチ処理とのバッティング

10g

- ・ 月曜日～金曜日 22時から翌朝6時までの8時間
- ・ 土曜日、日曜日 土曜日の0時から2日間(つまり日曜日の23時59分59秒まで)

11g

- ・ 月曜日～金曜日 22時から翌2時までの4時間
- ・ 土曜日・日曜日 土曜日のAM6時から20時間(翌2時まで)

バッチ処理とかがぶらないように収集時間のスケジュール設定

ビュー 編集 削除 スケジュール 実行

選択	名前	リ
<input type="radio"/>	WEEKNIGHT_WINDOW	
<input type="radio"/>	WEEKEND_WINDOW	
<input checked="" type="radio"/>	WEDNESDAY_WINDOW	D
<input type="radio"/>	THU... WINDOW	D

タイムゾーン GMT +9:00

繰り返し

繰り返し 毎週

間隔(週) 1

曜日  月曜日  火曜日  水曜日  木曜日  金曜日

時間 10 00 00 AM PM

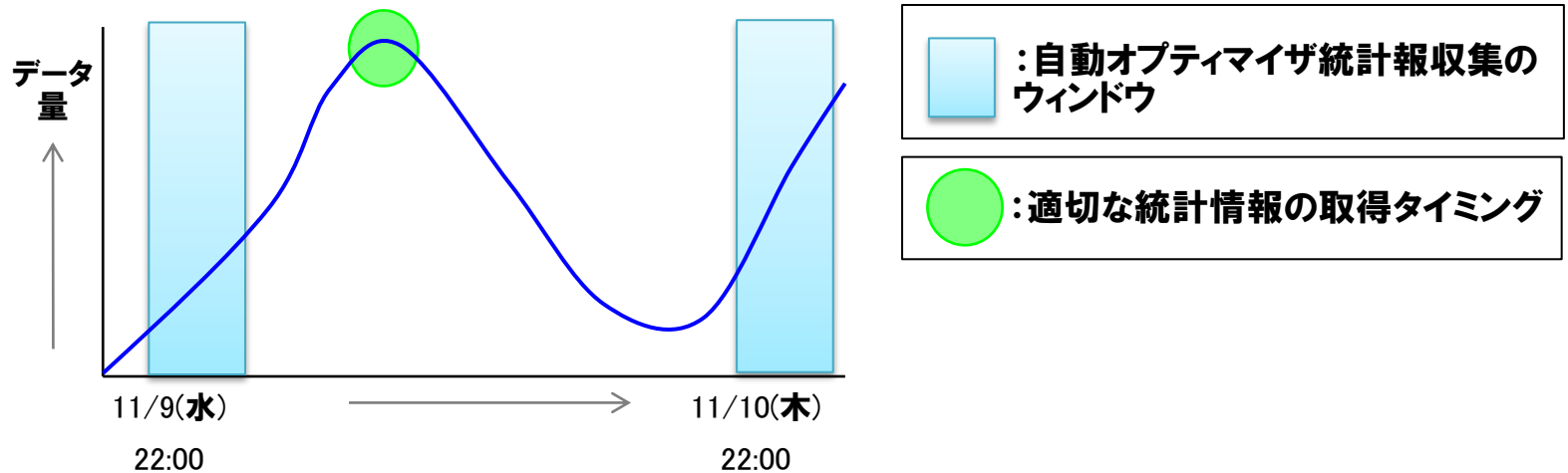
開始可能時間 即時

期間 4 時間 0 分

ウィンドウの編集により、スケジュールは変更可能!

# 自動オプティマイザ統計情報収集の注意点(2/2)

- 適切な統計情報収集のタイミング ≠ 定期的な収集
  - 下記のように一日の間でもデータ量変動が多い場合、たまたまデータ量が小さい状態でオプティマイザ統計が取得され、不適切な実行計画が生成される可能性がある



▶ データ量が最大となるタイミングで手動で取得してロック  
(ロックは自動オプティマイザ統計情報収集の対象からはずすため)



# 本セッションの内容

CBOを前提とした日々の運用

- ・デフォルトでの自動統計情報収集の内容と注意点
- ・新しい実行計画の固定方法

トラブル！！

トラブル発生時の対応

- ・再現情報の確保
- ・早期問題解決

# 今までの実行計画の固定方法(1/2)

## ・ ヒントで実行計画を固定

メリット

- ・ ヒントで固定することで、指定部分のオプティマイザ統計の変化と実行計画の変化を切り離すことが可能
- ・ 実行計画を意図したとおりに編集しやすい

注意点

- ・ SQL文の中に追記するので、アプリの改修が必要
- ・ 完全な実行計画の固定ではない(ヒント部分のみの固定)
- ・ 固定されるため、より最適な実行計画が生成される場合も関知できない

```
SELECT /*+ INDEX(e1 emp_emp_id_pk) */
e1.first_name, e1.last_name, j.job_id,
sum(e2.salary) total_sal
FROM employees e1, employees e2, job_history j
WHERE e1.employee_id = e2.manager_id
AND e1.employee_id = j.employee_id
AND e1.hire_date = j.start_date
GROUP BY e1.first_name, e1.last_name, j.job_id
ORDER BY total_sal;
```

# 今までの実行計画の固定方法(2/2)

## ▪ ストアドアウトラインで実行計画を固定

- ヒントで固定することで、指定部分のオプティマイザ統計の変化と実行計画の変化を切り離すことが可能
- 実行計画の完全な固定が可能
- SQL文はそのままなのでアプリの改修は不要

メリット

- ヒント付きのSQLの実行計画で固定したい場合、SQLもヒント付きである必要がある。(SQLそのまま実行計画の編集も可能ではあるが、機能制限がある)
- アプリケーションがSQL文をリテラルで発行するときに対応不可
- 固定されるため、より最適な実行計画が生成される場合も関知できない

注意点

- ①ALTER SESSION SET CREATE\_STORED\_OUTLINES = <category\_name>;  
(セッション単位、インスタンス単位での作成を指定可能)
- ②ストアド・アウトラインを作成したい(実行計画を移行したい)SQL を実行
- ③ALTER SESSION SET CREATE\_STORED\_OUTLINES = FALSE;

※本機能は11g以降での使用は非推奨です

(後のスライドで説明するSQL Plan Managementをご使用ください)

ORACLE

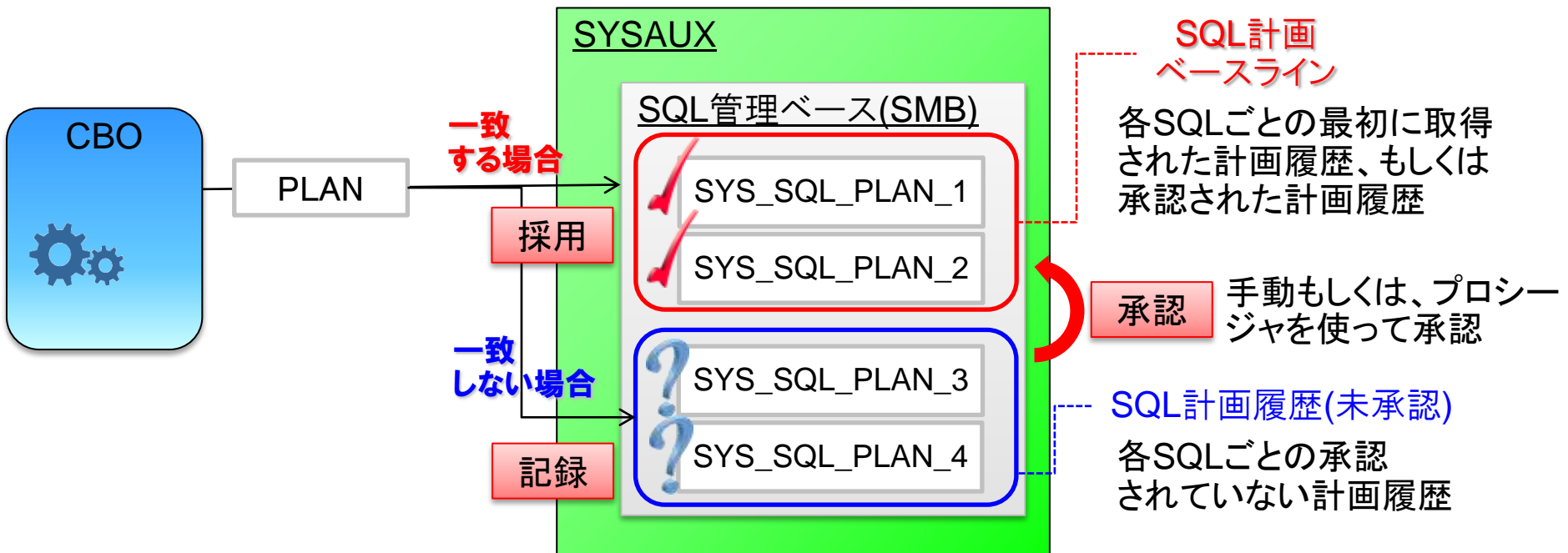
# 新しい(11gR1-)実行計画の固定方法

## SQL Plan Management(SPM)

	ヒント	ストアド・アウトライン	SQL Plan Management
オプティマイザ統計の変化と実行計画の変化を切り離し	○	○	○
実行計画を編集しやすい	○	△	○ (ヒントで編集)
完全なSQLの固定	△ 不要(指定した部分のみ)	○ (可能)	○ (可能)
実行計画変化の検知	× (不可能)	× (不可能)	○ (可能、変化した実行計画は履歴管理)
アプリの改修	× (必要)	△ (ヒントで実行計画を固定した場合必要)	○ (不要、ヒントで指定した実行計画をヒント無しのSQLに適用可能)
リテラル対応	○	×(バインド変数化)	×(バインド変数化)

# SQL Plan Management 機能概要

- SQL計画ベースラインに実行計画とSQLを登録
- CBOが生成した実行計画とSQL計画ベースラインの実行計画を比較、一致すれば採用、不一致ならSQL計画履歴に未承認の状態記録し、実行計画としてはSQL管理ベースのものが選択される
- 記録されたものは承認後、SQL計画ベースラインに追加される



# 新しい(11gR1-)実行計画の固定方法(再掲)

## SQL Plan Management(SPM)

	ヒント	ストアド・アウトライン	SQL Plan Management
オプティマイザ統計の変化と実行計画の変化を切り離し	○	○	○
実行計画を編集しやすい	○	△	○ (ヒントで編集)
完全なSQLの固定	△ 不要(指定した部分のみ)	○ (可能)	○ (可能)
実行計画変化の検知	× (不可能)	× (不可能)	○ (可能、変化した実行計画は履歴管理)
アプリの改修	× (必要)	△ (ヒントで実行計画を固定した場合必要)	○ (不要、ヒントで指定した実行計画をヒント無しのSQLに適用可能)
リテラル対応	○	×(バインド変数化)	×(バインド変数化)

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(1/9)

- 以下のSQLの実行計画をSPMを使って変更します。最初はINDEXスキャンです

```
select * from tab2 where c2 > 999;
```

## 実行計画

```
-----  
Plan hash value: 2200541503  
-----
```

Id	Operation	Name	Rows
0	SELECT STATEMENT		10
1	TABLE ACCESS BY INDEX ROWID	TAB2	10
* 2	<b><u>INDEX RANGE SCAN</u></b>	IND2	10

INDEX RANGE SCANの実行計画です。今回は、この実行計画をSQLを変更せずにFULLSCANに変えます

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(2/9)

- SPMを使うため、前ページのSQL情報をSQL管理ベースラインに登録
  - ※カーソルキャッシュに残っている情報から登録する方法の他に、optimizer\_capture\_sql\_plan\_baselines=trueにして2回実行する方法やSTSから指定する方法があります

検索

SQLテキスト

デフォルトでは、検索は大文字/小文字が区別されます。完全一致検索または大文字/小文字を区別する検索を実行するには、検索文字列を二重引用符で囲んでください。ワイルドカードとして '%' 記号を使用することもできます。

選択名前	SQLテキスト	有効	確定済	再現済	固定済	自動ページ	元	作成	最終変更
項目が見つかりません									

ヒント 表には最大2000行が表示されます。目的の結果を得るには検索基準を使用します。

SQL計画管理

SQL計画ベースラインのロード

既存のSQLチューニング・セットから、またはカーソル・キャッシュから直接、計画をバ

SQLチューニング・セット(STS)からの計画のロード

SQLチューニング・セット  .

カーソル・キャッシュからの計画のロード

SQL ID

SQLIDを指定します  
(ライトのマークをクリックするとSQLテキストの一部分からSQLIDを選択できるページが現れます)

本操作はプロシージャ  
dbms\_spm.load\_plans\_from\_cursor\_cacheを使用しても実施できます。



# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(3/9)

- ロードされると下記のように表示されます

**検索**

SQLテキスト

デフォルトでは、検索は大文字/小文字が区別されます。完全一致検索または大文字/小文字を区別する検索を実行するには、検索文字列を二重引用符で囲んでください。ワイルドカードとして'%'記号を使用することもできます。

固定済 - はい

すべて選択 | 選択解除

選択	名前	SQLテキスト	有効	確定済	再現済	固定済	自動パージ	元	作成	最終変更
<input type="checkbox"/>	SQL_PLAN_2a1h62h9gkrg371befc37	select * from tab2 where c2 > 999	YES	YES	YES	NO	YES	MANUAL-LOAD	2011/10/07 13:10:47	2011/10/07 13:10:47

ヒント 表には最大2000行が表示されます。目的の結果を得るには検索基準を使用します。

[SQLプロファイル](#) [SQLバッチ](#) [SQL計画ベースライン](#)

クリックしてベースラインの詳細を確認(次スライド)

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(4/9)

- 登録されたベースラインの状態やSQL文、実行計画の内容などを確認できます
- このヒント指定前のSQLのSQL handleは後で使用するのでメモします

SQL計画ベースライン詳細

SQL handleをメモ

---

SQL handle: SQL\_2506061412f95de3  
SQL text: select \* from tab2 where c2 > 999

---

Plan name: SQL\_PLAN\_2alh62h9gkrg371befc37      Plan id: 1908341815  
Enabled: YES      Fixed: NO      Accepted: YES      Origin: MANUAL-LOAD

---

Plan hash value: 2200541503

---

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10	2090	13 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID	TAB2	10	2090	13 (0)	00:00:01
* 2	<u>INDEX RANGE SCAN</u>	IND2	10		2 (0)	00:00:01

---

Predicate Information (identified by operation id):

---

2 - access("C2">999)

先ほどのSQL textとINDEX RANGE SCANの実行計画が登録されていることを確認

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(5/9)

- ヒントを使用してSQLの実行計画を変更します。今はFULLスキャン (TABLE ACCESS FULL)にします

```
select /*+ FULL (TAB2) */ * from tab2 where c2 > 999;
```

## 実行計画

```
-----  
Plan hash value: 2156729920  
-----
```

```
-----  
| Id  | Operation          | Name  | Rows  | Bytes |  
-----  
| 0   | SELECT STATEMENT   |      |     18 | 4104 |  
|* 1  | TABLE ACCESS FULL | TAB2  |     18 | 4104 |  
-----
```

- SQLIDとPLAN\_HASH\_VALUEを取得してメモします。以下は取得例

```
select sql_id, plan_hash_value from v$sql where sql_text like  
'select /*+ FULL (TAB2) */ * from tab2%';
```

```
SQL_ID
```

SQLIDをメモ

```
PLAN_HASH_VALUE
```

PLAN\_HASH\_VALUE  
をメモ

```
1j82gtnc0n4ps
```

```
2156729920
```

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(6/9)

- dbms\_spm.load\_plans\_from\_cursor\_cacheに、SQL管理ベースラインに今までメモした内容(ヒント指定後のSQL\_ID、ヒント指定後のPLAN\_HASH\_VALUE、**ヒント指定前のSQL\_HANDLE**)を入力してSQL管理ベースラインにFULL SCANの実行計画をロード

```
var res number
exec :res := dbms_spm.load_plans_from_cursor_cache( -
sql_id => '&hinted_SQL_ID', -
plan_hash_value => &hinted_plan_hash_value, -
sql_handle => '&sql_handle_for_original');

hinted_sql_idに値を入力してください: 1j82gtnc0n4ps
<--- ★ ヒント指定後の SQL_ID
hinted_plan_hash_valueに値を入力してください: 2156729920
<--- ★ ヒント指定後の PLAN_HASH_VALUE
sql_handle_for_originalに値を入力してください: SQL_2506061412f95de3
<--- ★ ヒント指定前の SQL_HANDLE
```

**ヒント指定前のSQL\_HANDLEを指定するところがポイント！！**

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(7/9)

- 下記のように、ヒント句無しのSQLテキストに対して新しく2つ目のベースラインが表示されます

## SQL計画管理

SQLプロファイル SQLバッチ SQL計画ベースライン

リフレッシュ

SQL計画ベースラインは、指定されたSQL文について許容可能なパフォーマンスを持つと見なされる実行計画です。

**設定**

SQL計画ベースラインの取得 **FALSE**  
SQL計画ベースラインの使用 **TRUE**  
計画の保存(週)

**SQL計画ベースラインに対するジョブ**

ジョブのロード	保留中	完了
		SPM_LOAD_20111007_131653

**検索**

SQLテキスト

デフォルトでは、検索は大文字/小文字が区別されます。完全一致検索または大文字/小文字を区別する検索を実行するには、検索文字列を二重引用符で囲ってください。ワイルドカードとして'%'記号を使用することもできます。

すべて選択 | 選択解除

選択	名前	SQLテキスト	有効	確定済	再現済	固定済	自動バージ元	作成	最終変更
<input type="checkbox"/>	SQL_PLAN_2a1h62h9gkr371befc37	select * from tab2 where c2 > 999	YES	YES	YES	NO	YES	MANUAL-LOAD 2011/10/07 14:01:56	2011/10/07 14:01:56
<input type="checkbox"/>	SQL_PLAN_2a1h62h9gkr3b157e6ce	select * from tab2 where c2 > 999	YES	YES	YES	NO	YES	MANUAL-LOAD 2011/10/07 14:08:32	2011/10/07 14:08:32

ヒント 表には最大2000行が表示されます。目的の結果を得るには検索基準を使用。

SQLプロファイル SQLバッチ SQL計画ベースライン

新しくできたFULL SCANのベースライン  
クリックして内容確認(次スライド)

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(8/9)

- 新しく作成されたベースラインの詳細を確認すると、ヒント句無しのSQLテキストに対してTABLE ACCESS FULLの実行計画が確認できます

## SQL計画ベースライン詳細

SQL handle: SQL\_2506061412f95de3  
SQL text: select \* from tab2 where c2 > 999

ヒント句無しのSQLテキスト

Plan name: SQL\_PLAN\_2a1h62h9gkrg3b157e6ce      Plan id: 2975327950  
Enabled: YES      Fixed: NO      Accepted: YES      Origin: MANUAL-LOAD

Plan hash value: 2156729920

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10	2090	102 (0)	00:00:02
* 1	<u>TABLE ACCESS FULL</u>	TAB2	10	2090	102 (0)	00:00:02

TABLE ACCESS FULLの実行計画

Predicate Information (ident

1 - filter("C2">999)

# ヒントで指定した実行計画をヒント句無しのSQLに適用 INDEXスキャンをFULLスキャンに変更する例(9/9)

- FULL SCANのベースラインのみを固定済み YESに指定することで実行計画をFULL SCANに固定できます
- このようにSQLテキストを変更なし、つまりアプリケーションの改修をせずにヒントで指定した実行計画に使用することができます
- また、一度登録すれば、使用する実行計画を簡単に切り替えることができるので、性能評価の際にも有効活用できます

## 検索

SQLテキスト

デフォルトでは、検索は大文字/小文字が区別されます。完全一致検索または大文字/小文字を区別する検索を実行するには、検索文字列を二重引用符で囲んでください。ワイルドカードとして'%'記号を使用することもできます。

すべて選択 | 選択解除

選択	名前	SQLテキスト	有効	確定済	再現済	固定済	自動バージ	元	作成	最終変更
<input type="checkbox"/>	SQL_PLAN_2a1h62h9gkrg371befc37	select * from tab2 where c2 > 999	YES	YES	YES	NO	YES	MANUAL-LOAD	2011/10/07 14:01:56	2011/10/07 14:01:56
<input type="checkbox"/>	SQL_PLAN_2a1h62h9gkrg3b157e6ce	select * from tab2 where c2 > 999	YES	YES	YES	YES	YES	MANUAL-LOAD	2011/10/07 14:08:32	2011/10/07 14:13:22

ヒント 表には最大2000行が表示されます。目的の結果を得るには検索基準を使用します。

[SQL プロファイル](#)

[SQL バッチ](#)

[SQL 計画ベースライン](#)

対象のみ「固定済=YES」とすることで使用する実行計画をひとつに固定できます。

※例で示しました、「固定=yes」は固定はするが履歴管理はしない設定になります

# 本セッションの内容

CBOを前提とした日々の運用

- ・デフォルトでの自動統計情報収集の内容と注意点
- ・新しい実行計画の固定方法

トラブル！！

トラブル発生時の対応

- ・再現情報の確保
- ・早期問題解決



# 再現情報確保の重要性

- 早期トラブル解決には再現情報の確保が重要
  - トラブル解決＝「一時的な処置」＋「原因説明＆運用への組み込み」
  - 再現情報の確保をしないまま「一時的な処置」をしてしまうと、特に「原因説明＆運用への組み込み」が長引き、時には不可能になる



トラブル解決のため慌てて統計情報を取り直して対処したため、過去の統計情報が残っていない

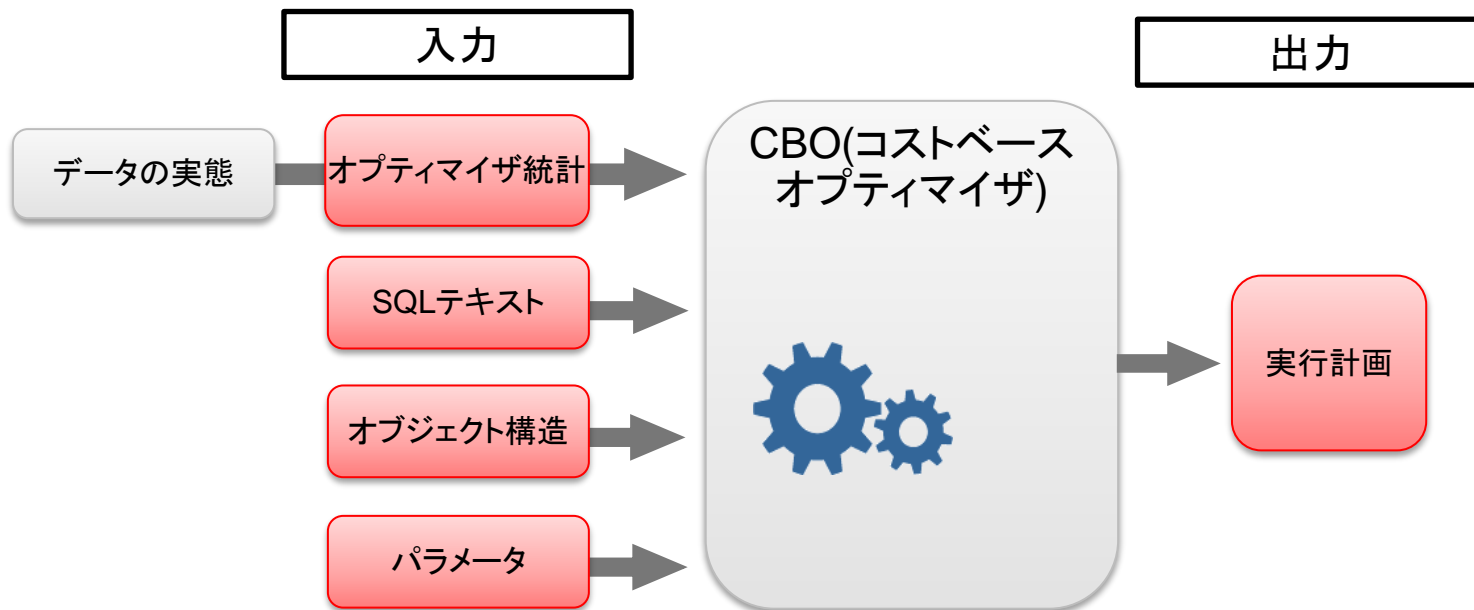
再現情報を得るには本番環境でもう一度状況を再現させて情報を再取得する必要があるが、そんなことはできない

処置の適切さが判断されないまま、一時的な処置が恒久的な処置として採用されてしまう

# 再現情報として必要な情報は何か

## オプティマイザ動作に起因している場合

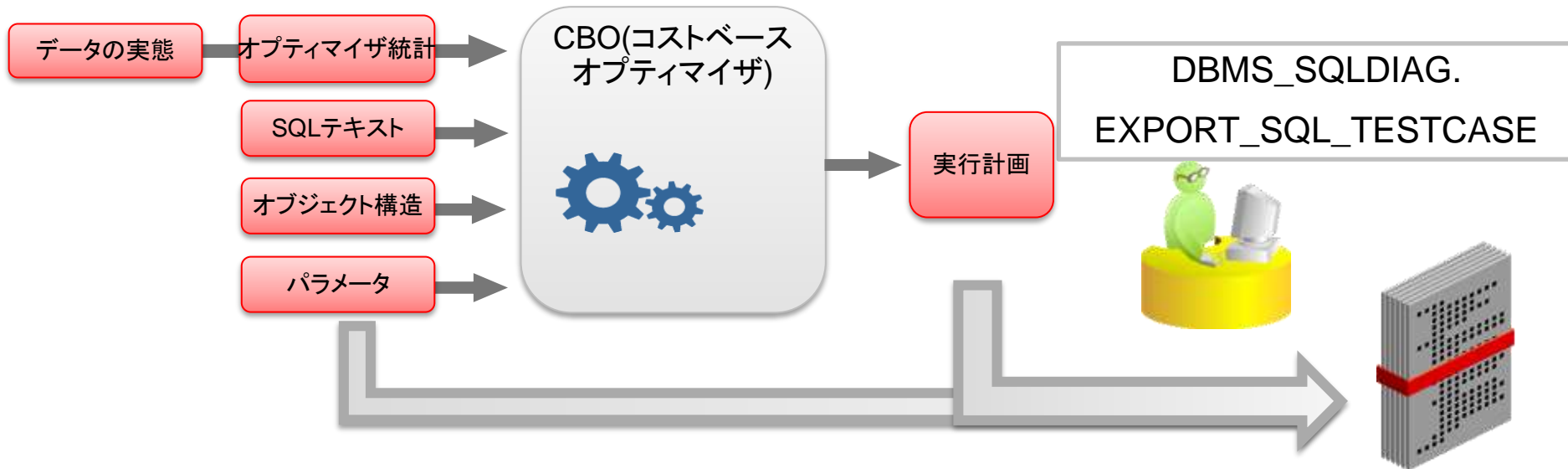
- ・ オプティマイザ動作に起因したトラブルの場合、状況を再現するためにはCBOの入力データの確保が重要
- ・ トラブル時に素早く実行する必要があるため事前に取得法を確立してしておく
- ・ どうやってこれらの情報を取得する手法を確立するかが問題



# 再現情報の確保

## Test Case Builder(10.2.0.4 -)

- ・ プロシージャで再現情報セットを一括取得
- ・ SQL テスト・ケース・ビルダーにより取得される情報
  - ・ SQL 文、表／索引の定義(実際のデータはオプションで選択可能) / SQL ファンクション / プロシージャ / パッケージ、オプティマイザ統計、初化パラメータ設定が含まれる



# Test Case Builder

## 共有プールに残っているカーソルの情報を指定する方法(1/2)

- 実行前準備: ターゲットSQLの情報取得と再現情報の保存先の設定

```
--該当SQLのSQL IDの取得
```

```
SQL> select sql_id,plan_hash_value,sql_text from v$sql
       where sql_text like '%SELECT EMPNO,ENAME,JOB,DNAME%';
```

```
SQL_ID          PLAN_HASH_VALUE
-----
```

```
SQL_TEXT
-----
```

```
410qucgpvnr0p 736908262
```

```
SELECT EMPNO,ENAME,JOB,DNAME FROM EMP E,DEPT D WHERE
E.DEPTNO = D.DEPTNO AND DNAME='SALES' ORDER BY EMPNO
```

```
-- SQL テスト・ケースの出力先のディレクトリ定義します
```

```
CREATE OR REPLACE DIRECTORY SQL_TEST_DIR AS '/tmp';
```

# Test Case Builder

## 共有プールに残っているカーソルの情報を指定する方法(2/2)

- 実行: DBMS\_SQLDIAG.EXPORT\_SQL\_TESTCASE プロシージャ

```
DECLARE
  V_TESTCASE CLOB;
BEGIN
  DBMS_SQLDIAG.EXPORT_SQL_TESTCASE(
    DIRECTORY => 'SQL_TEST_DIR', --出力先のディレクトリ・オブジェクトを指定
    sql_id => '410qucgpvnr0p',    -- 確認した SQL_ID を指定
    plan_hash_value => 1736908262, -- 確認した plan_hash_value を指定
    exportData => FALSE,        -- 表データをエクスポートする場合は TRUE
    testcase_name => 'tc3',     -- テストケース名を指定
    testcase => V_TESTCASE);    -- 宣言した CLOB 変数を指定
end;
/
```

TRUEの場合、出力データに実データが含まれるので取り扱いに注意

### --出力例

```
$ cd /tmp
```

```
$ ls tc3*
```

```
tc3README.txt tc3dpexp.log tc3dpimp.sql tc3ol.xml tc3ssimp.sql tc3xpl.txt tc3xplo.sql
```

```
tc3dpexp.dmp tc3dpexp.sql tc3main.xml tc3sql.xml tc3ts.xml tc3xplf.sql tc3xpls.sql
```

# 本セッションの内容

CBOを前提とした日々の運用

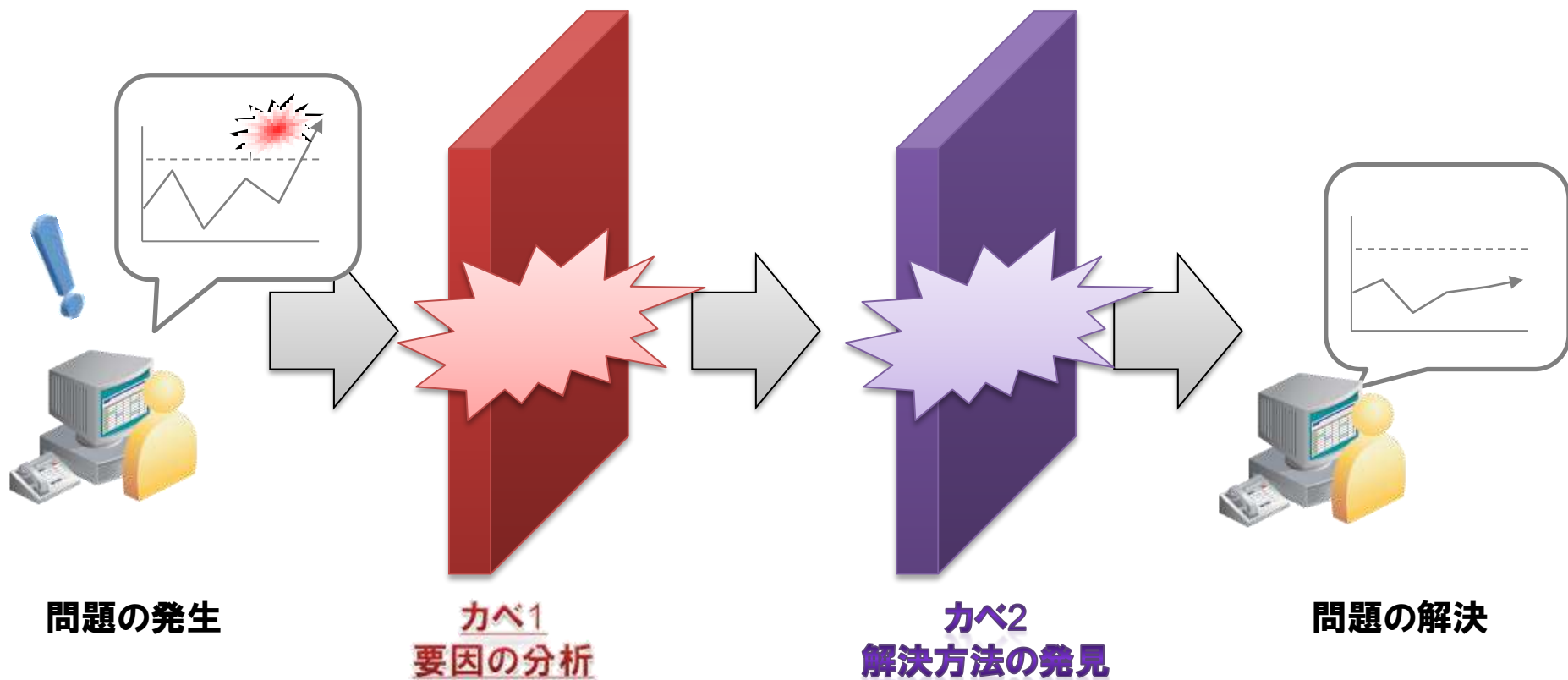
- ・Oracleが自動でやっていることおよび注意点
- ・新しい実行計画の固定方法

トラブル！！

トラブル発生時の対応

- ・再現情報の確保
- ・早期問題解決

# 性能問題解決までの2つのカベ



問題解決までには、おもに2つの「カベ」がある  
カベ2だけでなく、カベ1「要因の分析」に時間がかかる

# 従来の診断・チューニング

v\$sqlなど複数の表を検索して問題のあるセッションを特定

```
SQL> select username, event, sid from v$sqlsession . . .
```

このセッションから長時間実行されているSQLをv\$sqlなどから特定

```
SQL> select sql_id, elapsed_time/1000000, executions, disk_reads from v$sql . . .
```

Statspackレポートを取得し、実行時間の長かったSQLを特定

```
SQL> @?/rdbms/admin/spreport.sql  
begin snapに値を入力してください :
```

特定したSQLの実行計画を表示し、問題のある場所を探して対処策を模索

```
SQL> select * from table(dbms_xplan.display_cursor . . .
```

- ✓ 情報取得に多くの工数が必要
- ✓ チューニング方法を見つけるまでに時間がかかりやすい
- ✓ チューニング効果を事前に見極めるのが困難

Statspackレポート例

```
Plan hash value: 4112611536
```

Id	Operation	Name	Rows	Bytes	TempSp	Cost (CPU)	Time
0	SELECT STATEMENT					74184 (100)	
1	HASH GROUP BY		1422	145K		74184 (2)	00:14:51
* 2	HASH JOIN		8082K	807M	8776K	73853 (1)	00:14:44
3	TABLE ACCESS FULL	ITEM	1028	6576K		2019 (1)	00:00:25
* 4	HASH JOIN		8127K	370M	1576K	47723 (2)	00:09:33
5	TABLE ACCESS FULL	DATE_DIM	73049	713K		377 (1)	00:00:05
* 6	TABLE ACCESS FULL	STORE_SALES	8127K	303M		27571 (2)	00:05:31

Predicate information (identified by operation id):

```
2 - access("T1025" = "I_ITEM_SK" = "T1101" "SS_ITEM_SK")
4 - access("T983" = "D_DATE_SK" = "T1101" "SS_SOLD_DATE_SK")
6 - filter("T1101" , "SS_SOLD_DATE_SK" IS NOT NULL)
```

実行計画表示例



# SQL文のチューニング

## SQLチューニング・アドバイザによる自動チューニング

### • SQLチューニング・アドバイザ

- Oracle Database10gから実装されたアドバイス機能
- 高負荷で問題となるSQL文や実行計画を診断し、アドバイスを提示
  - 統計の再取得
  - SQL文の問題点を探し、SQL文の修正方法
  - 必要な索引の作成をアドバイス
  - SQLプロファイルの作成



# SQL文のチューニング

## SQLチューニング・アドバイザの実行例(1/3)

Enterprise Managerの「パフォーマンス・ページ」からデータベースの負荷状況を確認

トップ・アクティビティ

次の詳細セクションの時間間隔を変更するには、グラフの下の帯をクリックします。

データの表示: 実行時間: 15秒リフレッシュ

アクティビティ・セッション

2008/10/06

「トップ・アクティビティ」ページから、特に負荷の高いSQL文やセッションを特定

選択した5分間隔の詳細

開始時間 2008/10/06 18時24分56秒 JST

ASHレポートの実行

**上位SQL**

選択	アクティビティ(%)	SQL ID	SQLタイプ
<input type="checkbox"/>	99.50	by9m5m597zh19	SELECT
<input type="checkbox"/>	.34	c5brdpybqass6	UNKNOWN
<input type="checkbox"/>	.17	a8hw04p5qx1t5	SELECT

**上位セッション**

アクティビティ(%)	セッションID	ユーザー名	プログラム
48.84	110	AST	sqlplus.exe
48.84	101	AST	sqlplus.exe
1.32	160	SYS	ORACLE.EXE (CKPT)
.33	139	DBSNMP	emagent.exe
.17	157	SYS	ORACLE.EXE (MMON)

# SQL文のチューニング

## SQLチューニング・アドバイザの実行例(2/3)

SQLの詳細: by9m5m597zh19

SQL IDに代替   データの表示

実行時間: 手動リフレッシュ

テキスト

```
select /*+ USE_NL(s c) FULL(s) FULL(c) AST */ c.cust_id, sum(s.quantity_sold)
from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2 group by c.cust_id
```

「上位SQL」から、負荷の高いSQL文を特定  
→このSQL文の実行計画を確認

チューニング対象のSQL文を選び、  
「SQLチューニング・アドバイザのスケジュール」から実行

詳細

次の詳細を参照するには計画ハッシュ値を選択してください。 計画ハッシュ値: 4005616876

統計 アクティビティ プラン 計画管理 チューニング履歴

データソース カーソル・キャッシュ 取得時間 2008/10/06 18:32:22 (UTC+09:00) 解析スキーマ AST オプティマイザ・モード ALL\_ROWS

追加情報

すべて開く | すべて閉じる

操作	オブジェクト	順序	バイト	CPU	時間	パーティションの開始	同合せブロック名/オブジェクトの別名	フィルタ	予測
SELECT STATEMENT		6	9,398	100					
HASH GROUP BY		5	13 9,398	1 0:1:53			SEL\$1		"C"."CUST_ID"[NUMBER,22], SUM(...
NESTED LOOPS		4	13 9,397	1 0:1:53					"C"."CUST_ID"[NUMBER,22], "S"....
TABLE ACCESS FULL	SH_CUSTOMERS	1	5 405	0 0:0:5			SEL\$1 / C@SEL\$1	"C"."CUST_ID"<2	"C"."CUST_ID"[NUMBER,22]
PARTITION RANGE ALL		3	1 8 8,992	1 0:1:48 1 28					"S"."QUANTITY_SOLD"[NUMBER,22]
TABLE ACCESS FULL	SH_SALES	2	1 8 8,992	1 0:1:48 1 28			SEL\$1 / S@SEL\$1	("S"."CUST_ID"<2 AND "S"."CUST...	"S"."QUANTITY_SOLD"[NUMBER,22]

▶ライトの説明の表示

統計 アクティビティ プラン 計画管理 チューニング履歴

# SQL文のチューニング

## SQLチューニング・アドバイザの実行例(3/3)

実装はボタンをクリックするだけ

推奨	タイプ	結果	推奨	論理	ベネフィット(%)	新規実行計画	実行計画の比較
<input type="radio"/>	統計	表"SH","SALES"の最適化統計は失効しています。	この表およびその索引に対する最適化統計の収集を検討してください。	適切な実行計画を選択するには、表およびその索引の最新の最適化統計が必要です。			
<input checked="" type="radio"/>	SQLプロファイル	この文により適している可能性のある実行計画が見つかりました。	推奨されるSQLプロファイルの承認を検討してください。		99.36	<input checked="" type="radio"/>	<input checked="" type="radio"/>
<input type="radio"/>	索引	索引を一つ以上作成すると、この文の実行計画を改善できます。	物理スキーマ設計を改善するAccess Advisorの実行が、推奨される索引の作成を検討してください。	推奨される索引を作成すると、この文の実行計画が大きく改善されます。ただし、単一の文ではなく代理SQLワークロードを使用した"Access Advisor"の実行が適切な場合もあります。この処理により、索引メンテナンス・オーバーヘッド	66.74	<input checked="" type="radio"/>	<input checked="" type="radio"/>

実装前にコストと時間が大幅に改善されることが分かる

実行計画の比較										
元の実行計画(注釈付き)										
● SQLチューニング・アドバイザによる元のプランからの調整を示します 計画ハッシュ値 4005616876										
すべて開く   すべて閉じる										
操作	ラインID	オブジェクト	オブジェクト・タイプ	順序	行	バイト	コスト	時間	CPUコスト	I/Oコスト
SELECT STATEMENT	0			6		0.013	9,398	113	1,895,178,624	9,312
HASH GROUP BY	1			5		0.013	9,398	113	1,895,178,624	9,312
NESTED LOOPS	2			4		0.013	9,397	113	1,873,027,712	9,312
TABLE ACCESS FULL	3	SH.CUSTOMERS	TABLE	1		0.005	405	5	21,682,460	404
PARTITION RANGE ALL	4			3		0.008	8,992	108	1,851,345,152	8,908
TABLE ACCESS FULL	5	SH.SALES	TABLE	2		0.008	8,992	108	1,851,345,152	8,908

SQLプロファイルのある新しい実行計画										
計画ハッシュ値 1458810583										
すべて開く   すべて閉じる										
操作	ラインID	オブジェクト	オブジェクト・タイプ	順序	行	バイト	コスト	時間	CPUコスト	I/Oコスト
SELECT STATEMENT	0			6		0.013	5	1	22,182,248	4
HASH GROUP BY	1			5		0.013	5	1	22,182,248	4
NESTED LOOPS	2			4		0.013	4	1	31,336	4
TABLE ACCESS BY GLOBAL INDEX ROWID	3	SH.SALES	TABLE	2		0.008	4	1	29,386	4
INDEX RANGE SCAN	4	SH.SALES_CUST_ID_IDX	INDEX	1		3	1	1	21,764	3
INDEX UNIQUE SCAN	5	SH.CUSTOMERS_PK	INDEX (UNIQUE)	3		0.005	0	1	1,950	0

# リアルタイムSQL監視

- 「リアルタイムSQL監視」とは
  - 実行中のSQLを自動で監視し、詳細な統計を取得
  - EMのグラフィカルなレポート画面から分析ができる
  - Oracle Database 11g以降で使用可能
- 特長
  - GUIから簡単にボトルネックを突き止められる
  - 再現待ちや特別な設定をせずすぐに分析を始められる
  - レポートをエクスポートして外部で参照可能
  - オーバーヘッドがほとんどない



# リアルタイムSQL監視 分析するSQLの特定

時間のかかっているSQLが自動的に監視され  
リストされる (経過時間等でソート可能)

データベース・インスタンス: ora111.jp.oracle.com >  
監視されたSQL実行

現在までのアクティブ期間 1時間

リフレッシュ 10秒

ス...	期間	SQL ID	セッ...	パラレル	データベース時間	IO	開始	終了	SQLテキ...
✓	4.0m	42nv7jt4d	126		3.9m	108K	12:25:11	12:29:10	select distinct

このSQL実行  
全体の統計

データベース・インスタンス: ora111.jp.oracle.com > 監視されたSQL実行 >

監視されたSQL実行の詳細

SCOTTとしてログイン

Save Mail レポートの表示

**概要**

SQL ID 42nv7jt4dcz81

実行が開始しました 2010年4月5日 月 12:25:11

最終リフレッシュ時間 2010年4月5日 月 12:29:10

実行ID 16777217

セッション 126

フェッチ コール 6

**時間**

期間 4.0m

データベース時間 3.9m

PL/SQLとJava 0.0s

**IOと待機の統計**

IO数 108K

バッファ読取り 108K

待機アクティビティ% 100

**詳細**

計画ハッシュ値 4112611536

計画統計 アクティビティ

操作	名前	予測し...	コ...	時系列(240s)	実行	実行行	メモ...	一時...	CPUアクティビ...	待機アクティビ...
SELECT STATEMENT			74K		1	66	2593P			
HASH GROUP BY		1422	74K		1	66	2593P	4.76		
HASH JOIN		8062K	74K		1	8127P	9745P	2.6		
TABLE ACCESS FULL	ITEM	102K	201		1	102K		6.49		
HASH JOIN		8127K	48K		1	8127P	3586P	1.3		
TABLE ACCESS FULL	DATE_DIM	73K	377		1	73K			60	
TABLE ACCESS FULL	STORE_SALES	8127K	28K		1	8127P		85	40	

実行計画のステップ  
ごとの統計など

# リアルタイムSQL監視

## SQL全文とバインド変数の参照

ORACLE Enterprise Manager 11g Database Control

データベース-インスタンス: orcl112.jp.oracle.com > 監視されたSQL実行 >

監視されたSQL実行の詳細

概要

SQL ID 62tmqf38yku24 ⓘ

実行が開始しました 2010年3月31日 水 23:39:23

最終リフレッシュ時間 2010年3月31日 水 23:39:23

実行ID 16777217

ユーザー SCOTT

フェッチ コール 1

SQLテキスト

```
SELECT /*+MONITOR*/  
COUNT(*) FROM bonus WHERE sal > :sal
```

SQL\_ID横の「i」マークをクリックすると、SQL全文に加え、実行時にバインド変数に入っていた値も参照可能 (Oracle Database 11g R2以降)

バインド

名前	位置	タイプ	値
:SAL	1	NUMBER	200

計画統計 アクティビティ

計画ハッシュ値 4224507826

操作	名前
SELECT STATEMENT	
SORT AGGREGATE	
TABLE ACCESS FULL	BONUS

# リアルタイムSQL監視

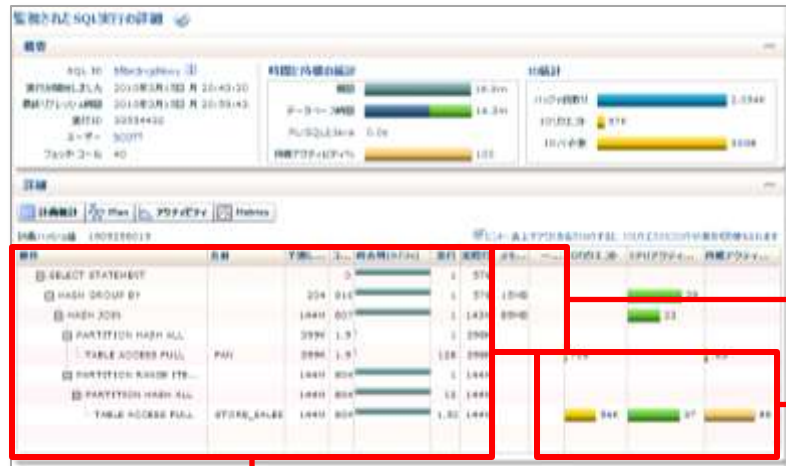
## SQLの実行状況の詳細確認

各ステップごとのメモリ (PGA)、一時表領域の使用状況

メモリ...	一時...
15MB	
89MB	

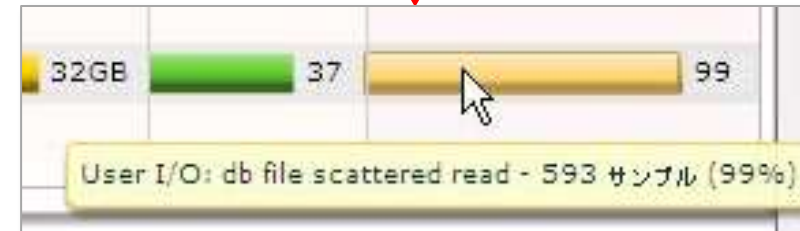
各ステップごとの実行タイミングや実行時間など  
(ここではITEM表、DATE\_DIM表、STORE\_SALES表の順に読み取りながら結合している)

実行計画で予想された行数と実際に返された行数の比較も容易



各ステップごとの待機イベントの発生個所やその内訳も簡単にわかる

操作	名前	予測し...	コ...	時系列(240s)	実行	実行行
SELECT STATEMENT			74K		1	66
HASH GROUP BY		1422	74K		1	66
HASH JOIN		8062K	74K		1	8127K
TABLE ACCESS FULL	ITEM	102K	201		1	102K
HASH JOIN		8127K	48K		1	8127K
TABLE ACCESS FULL	DATE_DIM	73K	377		1	73K
TABLE ACCESS FULL	STORE_SALE	8127K	28K		1	8127K





# リアルタイムSQL監視

## SQL単体のリソース使用量の時系列グラフ



# リアルタイムSQL監視

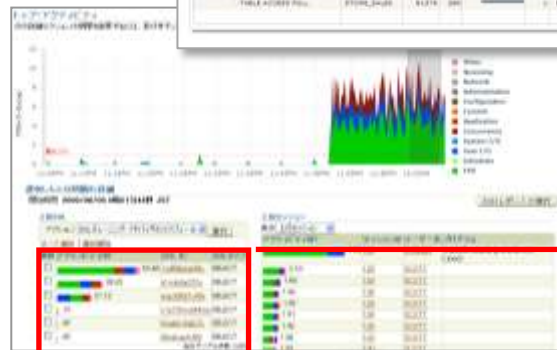
## 実行中のデータ参照(「今ここ！」マーク)

現在実行中であることを示すマーク

「今ここ！」

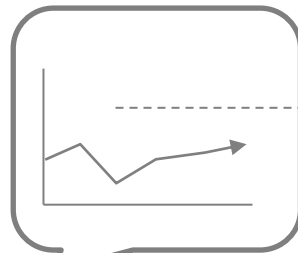
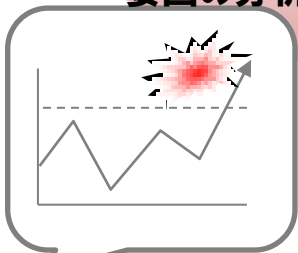
進行状況がわかるため、「あとどれくらいで(バッチなどの)処理が終了するか」、見当をつけられる

操作	名前	予測し...	コスト	時系列(50s)	実行	実際行	メモリー	一時	CPUアクティビ...	待機アクティビ...	進行状況%
SELECT STATEMENT			74K		1		2630K				
HASH GROUP BY		1422	74K		1		2630K	8.89			
HASH JOIN		8062K	74K		1	1287K	9794K				
TABLE ACCESS FULL	ITEM	102K	2019		1	102K		27		33	
HASH JOIN		8127K	48K		1	1287K	3586K	2.22			
TABLE ACCESS FULL	DATE_DIM	73K	377		1	73K				67	
TABLE ACCESS FULL	STORE_SALES	8127K	28K		1	1287K		62			16



要因の分析

解決方法の発見



Oracle Enterprise Managerを利用することで、2つのカベを早急にクリア

# まとめ

CBOを前提とした日々の運用

トラブル！！

トラブル発生時の対応

- ・Oracleが自動でやっていることおよび注意点
  - ⇒自動オプティマイザ統計情報収集
- ・新しい実行計画の固定方法
  - ⇒SQL Plan Management

ご紹介した機能とエディションおよびオプションの関係

- SQL Plan Management → Enterprise Edition
- Test Case Builder → Standard Edition
- SQL Tuning Advisor
  - Enterprise Edition Diagnostic Pack Tuning Pack
- リアルタイムSQL監視
  - Enterprise Edition Diagnostic Pack Tuning Pack

- ・再現情報の確保
  - ⇒Test Case Builder
- ・早期問題解決
  - ⇒SQL Tuning Advisor &リアルタイムSQL監視

# Appendix



# 自動統計情報収集のDBMS\_STATSデフォルト動作と設定値

	デフォルト値	デフォルト動作	他に設定可能な値	説明
収集対象とみなす表の更新% (stale_percent)	10%	DMLにより表の10%以上更新された場合	任意の%(11gから)	表の何%の行の変更が発生していたら収集対象とみなすか指定
サンプルサイズ (estimate_percent)	DBMS_STATS.AUTO_SAMPLE_SIZE	適切な統計を行うために最適なサンプル・サイズをOracleで決定	0.000001から100 代表的な設定を抜粋(詳細はマニュアル参照) 列値の種類の数 (NDV) <= [数字] : 頻度分布ヒストグラム 列値の種類の数 (NDV) > [数字] : 高さ調整ヒストグラム	行の%を指定
カラム統計収集 (method_opt)	FOR ALL COLUMNS SIZE AUTO	列のデータ配分とワークロードに基づいて、ヒストグラムを収集する列が判断されます	FOR ALL COLUMNS SIZE [数字(1から254)]	すべてのカラムに対して[数字]で指定したバケット数のヒストグラムを作成
			FOR COLUMNS [列名] SIZE [数字(1から254)]	[列名]のカラムに対して[数字]で指定したバケット数のヒストグラムを作成
パラレル度 (degree)	NULL (NULLだがパラレル実行される可能性があることに注意)	TABLE文のDEGREE句で指定された表のデフォルト値が使用されます	パラレル度	直接数字で指定
			DBMS_STATS.DEFAULT_DEGREE	初期化パラメタに基づくデフォルト値
			DBMS_STATS.AUTO_DEGREE	オブジェクトサイズに応じてシリアルもしくは、初期化パラメタに基づくデフォルト値

	デフォルト値	デフォルト動作	他に設定可能な値	説明
パーティションの統計収集 (granularity)	AUTO	パーティション化のタイプに基づいてOracleが細分化を決定します	ALL	すべての統計情報(サブパーティション、パーティションおよびグローバル)を収集
			GLOBAL	グローバル(表・索引レベル)の統計情報を収集
			GLOBAL AND PARTITION	グローバルおよびパーティション・レベルの統計情報を収集
			PARTITION /SUB PARTITION	パーティション・レベル/サブパーティション・レベルの統計情報を収集
インデックスの統計収集 (cascade)	DBMS_STATS.AUTO_CASCADE	索引の統計情報を収集するかどうかをOracleが決定する	TRUE	表の各索引にGATHER_INDEX_STATSを実行するのと同じ
			FALSE	索引の統計情報収集は取得しない
依存カーソルの無効化 (※) (no_invalidate)	DBMS_STATS.INVALIDATE	依存カーソルの無効化の時期をOracleが決定できるようにする	TRUE	依存カーソルを無効化しない
			FALSE	すぐに依存カーソルを無効化

(※)no\_invalidateは新しい実行計画を使用するため、今あるメモリ上の実行計画とのカーソルの依存関係を無効化するための設定です

# おすすめ資料(1/2)

- **オラクルコンサルが語る統計情報管理の真髄シリーズ Part1-4**
  - OTN セミナー オンデマンド内のオラクルコンサルが語るシリーズ内コンテンツ
  - Part 1:  
<http://www.oracle.com/technetwork/jp/ondemand/db-new/0630-1100-consultokei-part1-431396-ja.pdf>
    - **なぜ統計情報の管理が必要なのか**
      - 統計情報の詳細と使用する事のメリット
      - 統計情報と実データ間の差異が及ぼす影響
  - Part 2-3  
<http://www.oracle.com/technetwork/jp/ondemand/db-new/20110726-consul-part2-457249-ja.pdf>  
<http://www.oracle.com/technetwork/jp/ondemand/db-new/consultant-part3-134191-ja.pdf>
    - **統計情報管理方針を決定するためのアプローチの理解、実践方法解説**
      - 統計情報管理方針を決定する上で考慮するポイント
      - 統計情報管理のアプローチを実践し、実際に管理方針を決定するためのポイント
  - Part 4  
<http://www.oracle.com/technetwork/jp/ondemand/db-new/consultant-part4-250325-ja.pdf>
    - SPMの機能解説
    - 統計情報管理とSPMによる性能管理方法解説



- ORACLE OPEN WORLD 2011 Optimizer関連セッション
  - [Session 13961]Oracle Optimizer: Best Practices for Managing Optimizer Statistics  
[https://oracleus.wingateweb.com/published/oracleus2011/sessions/13961/13961\\_Cho134332.pdf](https://oracleus.wingateweb.com/published/oracleus2011/sessions/13961/13961_Cho134332.pdf)
    - 統計情報の取得方法(DBMS\_STATS)について新機能も交えた最適な使用方法を紹介
  - [Session 14069]Oracle Optimizer: Tips for Preventing Suboptimal Execution Plans  
[https://oracleus.wingateweb.com/published/oracleus2011/sessions/14069/14069\\_Cho227326.pdf](https://oracleus.wingateweb.com/published/oracleus2011/sessions/14069/14069_Cho227326.pdf)
    - 実行計画に関するTips集
      - 本セッションで説明したSPMを使って、ヒント句で編集した実行計画をヒント句無しのSQLに適用する方法も紹介。こちらではすべてプロシージャで実施しています

# OTNセミナーオンデマンド

コンテンツに対する  
ご意見・ご感想を是非お寄せください。

OTNオンデマンド 感想



[http://blogs.oracle.com/oracle4engineer/entry/otn\\_ondemand\\_questionnaire](http://blogs.oracle.com/oracle4engineer/entry/otn_ondemand_questionnaire)

上記に簡単なアンケート入力フォームをご用意しております。

セミナー講師/資料作成者にフィードバックし、  
コンテンツのより一層の改善に役立てさせていただきます。

是非ご協力をよろしくお願いいたします。

# OTNセミナーオンデマンド

日本オラクルのエンジニアが作成したセミナー資料・動画ダウンロードサイト

## 掲載コンテンツカテゴリ(一部抜粋)

Database 基礎

Database 現場テクニック

Database スペシャリストが語る

Java

WebLogic Server/アプリケーション・グリッド

EPM/BI 技術情報

サーバー

ストレージ



超入門! Oracle データベースって何  
再生時間: 60分

100以上のコンテンツをログイン不要でダウンロードし放題

データベースからハードウェアまで充実のラインナップ

毎月、旬なトピックの新作コンテンツが続々登場

## 例えばこんな使い方

- 製品概要を効率的につかむ
- 基礎を体系的に学ぶ/学ばせる
- 時間や場所を選ばず(オンデマンド)に受講
- スマートフォンで通勤中にも受講可能



毎月チェック!



コンテンツ一覧 はこちら

<http://www.oracle.com/technetwork/jp/ondemand/index.html>

新作&おすすめコンテンツ情報 はこちら

<http://oracletech.jp/seminar/recommended/000073.html>

OTNオンデマンド



# オラクルエンジニア通信

オラクル製品に関わるエンジニアの方のための技術情報サイト

## オラクルエンジニア通信 - 技術資料、マニュアル、セミナー

Oracleエンジニアのための技術情報サイト by Oracle Japan

**新着情報を知りたい**

**技術資料を探したい**

**セミナーを受けたい**

**About**

Oracleエンジニアの方がスキルアップしていただくために、厳選した情報をお届けしています

技術資料	<p><b>インストールガイド・設定チュートリアルetc. 欲しい資料への最短ルート</b></p>	アクセスランキング	<p><b>他のエンジニアは何を見ているのか？人気資料のランキングは毎月更新</b></p>
特集テーマ Pick UP	<p><b>性能管理やチューニングなど月間テーマを掘り下げて詳細にご説明</b></p>	技術コラム	<p><b>SQLスクリプト、索引メンテナンスetc. 当たり前運用/機能が見違える!?</b></p>

<http://blogs.oracle.com/oracle4engineer/>

オラクルエンジニア通信



製品/技術  
情報



Oracle Databaseっていくら？オプション機能も見積れる簡単ツールが大活躍

セミナー



基礎から最新技術までお勧めセミナーで自分にあった学習方法が見つかる

スキルアップ



ORACLE MASTER ! 試験頻出分野の模擬問題と解説を好評連載中

Viva!  
Developer



全国で活躍しているエンジニアにスポットライト。きらりと輝くスキルと視点を盗もう

<http://oracletech.jp/>

oracletech



あなたにいちばん近いオラクル



# Oracle Direct

まずはお問合せください

Oracle Direct



システムの検討・構築から運用まで、ITプロジェクト全般の相談窓口としてご支援いたします。  
システム構成やライセンス/購入方法などお気軽にお問い合わせ下さい。

## Web問い合わせフォーム

専用お問い合わせフォームにてご相談内容を承ります。  
[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

※フォームの入力にはログインが必要となります。  
※こちらから詳細確認のお電話を差し上げる場合がありますので  
ご登録の連絡先が最新のものになっているかご確認下さい。

## フリーダイヤル

0120-155-096

※月曜～金曜  
9:00～12:00、13:00～18:00  
(祝日および年末年始除く)

ORACLE

# **Hardware and Software Engineered to Work Together**

**ORACLE®**