

# Oracle Direct Seminar



**ORACLE®**

## 実践！開発フェーズでのDBチューニングTips

日本オラクル株式会社  
**Oracle** Direct



# Agenda

- データベース・チューニングの概要
- 開発フェーズでのEMの活用例
  - 例1 複雑な結合のチューニング
  - 例2 構文に問題のあるSQL文のチューニング
  - 例3 待機の発見と解決
  - 例4 効率が悪い処理のパフォーマンス改善

## Oracle Directの無償技術サービス

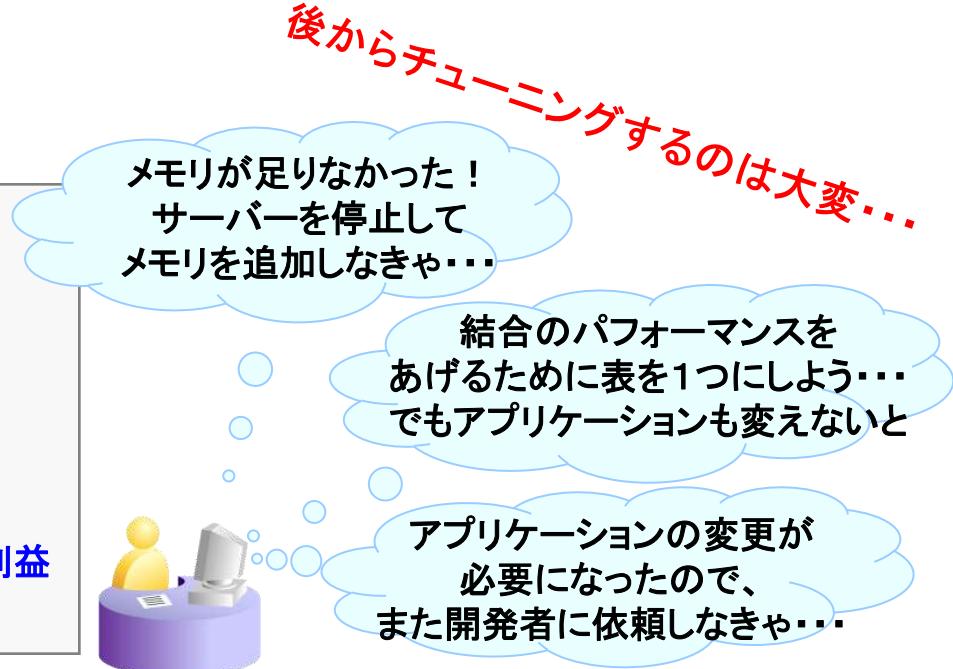
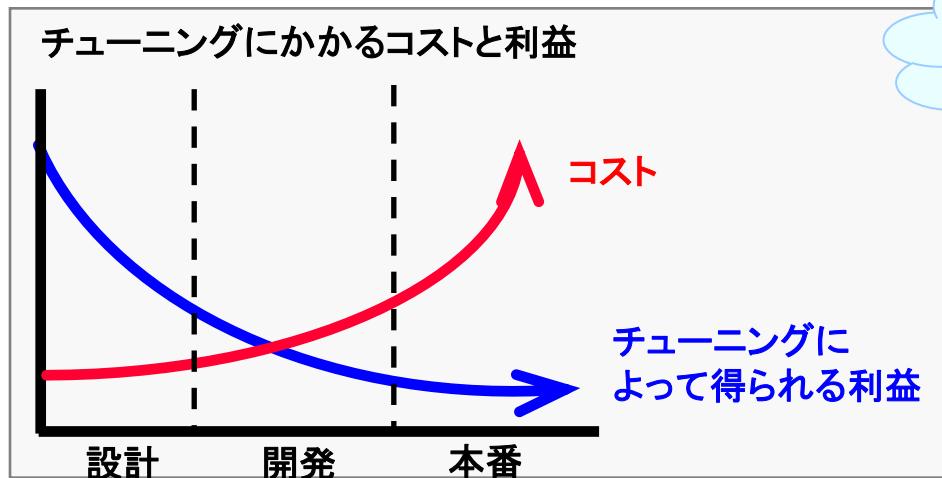
- SQL Serverからの移行アセスメント
- MySQLからの移行相談
- PostgreSQLからの移行相談
- Accessからの移行アセスメント
- Application Server 移行相談
- Oracle Database バージョンアップ支援
- Oracle Developer/2000 Webアップグレード相談
- パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
- システム連携アセスメント

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

# チューニングは、設計、開発段階から

- 設計、開発段階からパフォーマンスを意識することが重要
  - 設計、開発、本番稼動後の格段階で、各担当者がパフォーマンスを考慮する必要がある
  - 後から実施するチューニング作業は、労力とコストに対して効果が得にくい傾向にある



# 開発フェーズにおける問題①

- プログラマーごとにSQLの品質が異なる
  - 開発標準(コーディング・ルール)の不統一
  - 開発リーダーがすべてのSQLをチェックすることは難しい
- 全員が同等のチューニング・スキルを持つことは難しい

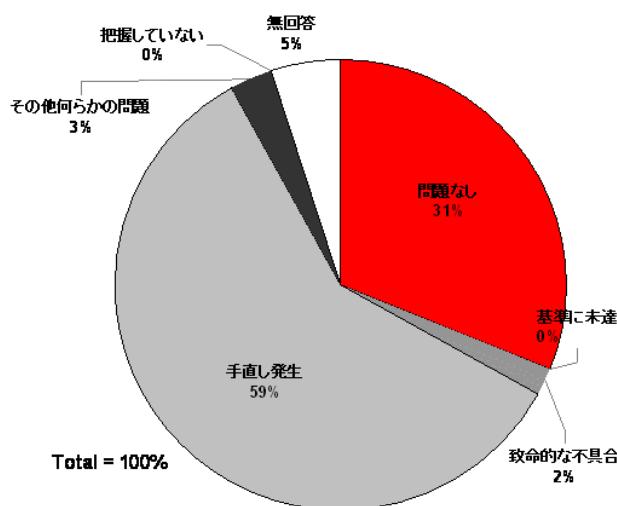


統一されたインターフェースを使って、  
標準的なチューニングを行う必要がある

ORACLE

# 開発フェーズにおける問題②

- 開発中、および開発後にテストを行う時間が十分に取れない
  - ビジネスのスピードに合わせ、ITシステムも迅速に対応するため、短納期で開発を行う必要がある
  - 膨大なテスト項目のすべてを実施することは難しい



自動化できる部分は自動化し  
効率化する必要がある

# 従来のOracleデータベースのチューニング



**定期的なパフォーマンス診断**  
OSの情報、V\$表の検索、Statspack、実行計画

Top 5 Timed Events			
Event	Waits	Time (s)	% Total
db file sequential read	51,925	8,172	91.49
log file sync	29,367	286	4.32
db file scattered read	1,030	103	0.19

**SQLチューニング**  
パラメータ設定、SQL書き換え

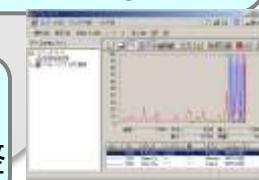
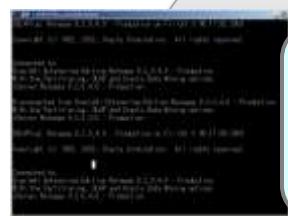
**最適な  
バックアップ&リカバリ**  
OSコマンド、バッチの作成

**領域管理**  
データファイル追加

**I/O分散を意識した  
ストレージ管理**  
Volume manager

**パフォーマンスを保つための  
定期メンテナンス**  
バッチスクリプトの作成

**リソース管理**  
サーバ負荷確認、メモリサイズ調整

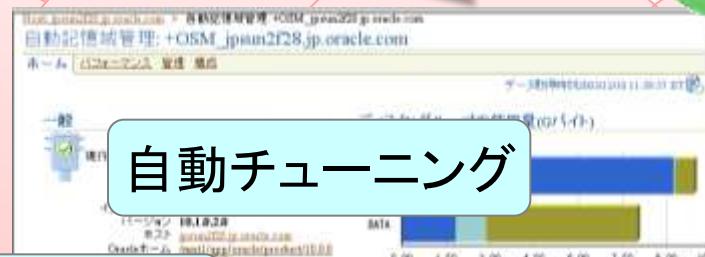
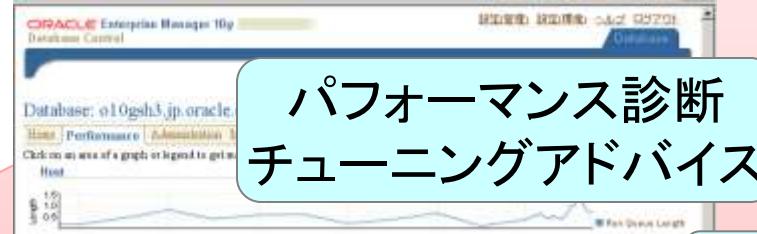


システムのパフォーマンスに関わる項目は多く  
これらを全て手作業で行うのは大変…

# Enterprise Managerを活用したチューニング

- GUIのわかりやすい画面から、全ての管理作業を実行
- これまで手動で行っていた作業を、データベースが自動的に実行

GUIの分かりやすいツールを使って  
簡単にチューニング！

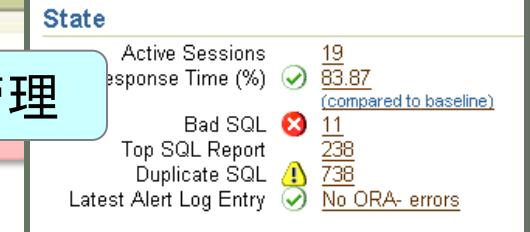


領域管理

自動ストレージ管理



リソース管理



ORACLE

# データベース・チューニングに有効な機能



## 自動ワークロードリポジトリ (AWR)

データベース稼動状況を保持しておくためのリポジトリ



## Enterprise Manager Database Console

## Automatic Database Diagnostic Monitor (ADDM)

AWRに収集されたデータを定期的に分析しデータベースのパフォーマンスをモニタ、診断する

## 各種アドバイザ機能

最適な設定や問題点を解消する方法を提示

- ・ SQLチューニング・アドバイザ
- ・ SQLアクセス・アドバイザ
- ・ メモリ・アドバイザ
- ・ セグメント・アドバイザ
- ・ UNDOアドバイザ
- ・ リカバリ・アドバイザ
- ・ SQL修復アドバイザ

ORACLE

# 開発フェーズでのEnterprise Managerの活用例

- 特定のSQL文のチューニング
  - 例1 複雑な結合のチューニング
  - 例2 構文に問題のあるSQL文のチューニング
- 非効率なアプリケーション・ロジックの発見と解決
  - 例3 待機の発見と解決
  - 例4 効率が悪い処理のパフォーマンス改善

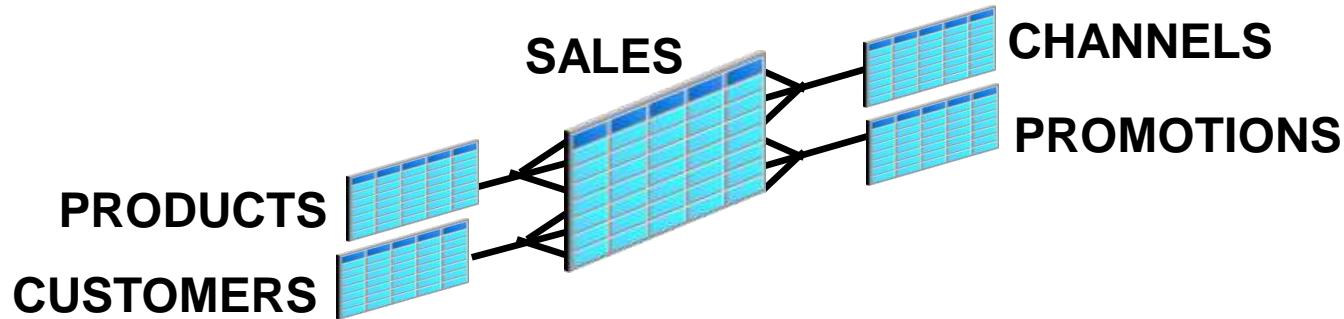


ORACLE®

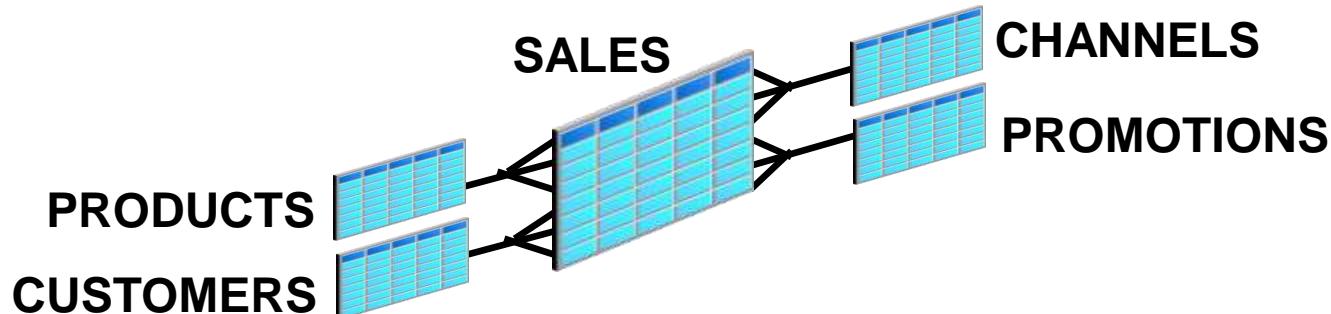
# 例1 複雑な結合のチューニング



特にSQL文が複雑になりがちなのでスタートキー…  
とりあえずSQL文を書いてみたけど、  
本当にこれでいいの？改善する方法はあるの？



# 例1：複雑な結合のチューニング



```
SELECT p.prod_name, sum(s.quantity_sold)
FROM sales s, channels ch, customers c, products p, promotions pr
WHERE s.channel_id = ch.channel_id
AND s.cust_id = c.cust_id
AND s.prod_id = p.prod_id
AND s.promo_id = pr.promo_id
AND EXISTS (SELECT 1 FROM channels ch2 WHERE ch.channel_id = ch2.channel_id
AND ch2.channel_id = 3)
AND EXISTS (SELECT 1 FROM customers c2 WHERE c.cust_id = c2.cust_id
AND c2.cust_city = 'Kyoto')
AND EXISTS (SELECT 1 FROM products p2 WHERE p.prod_id = p2.prod_id
AND p2.prod_name = 'Deluxe Mouse')
AND EXISTS (SELECT 1 FROM promotions pr2 WHERE pr.promo_id = pr2.promo_id
AND pr2.promo_id = 999)
GROUP BY p.prod_name;
```

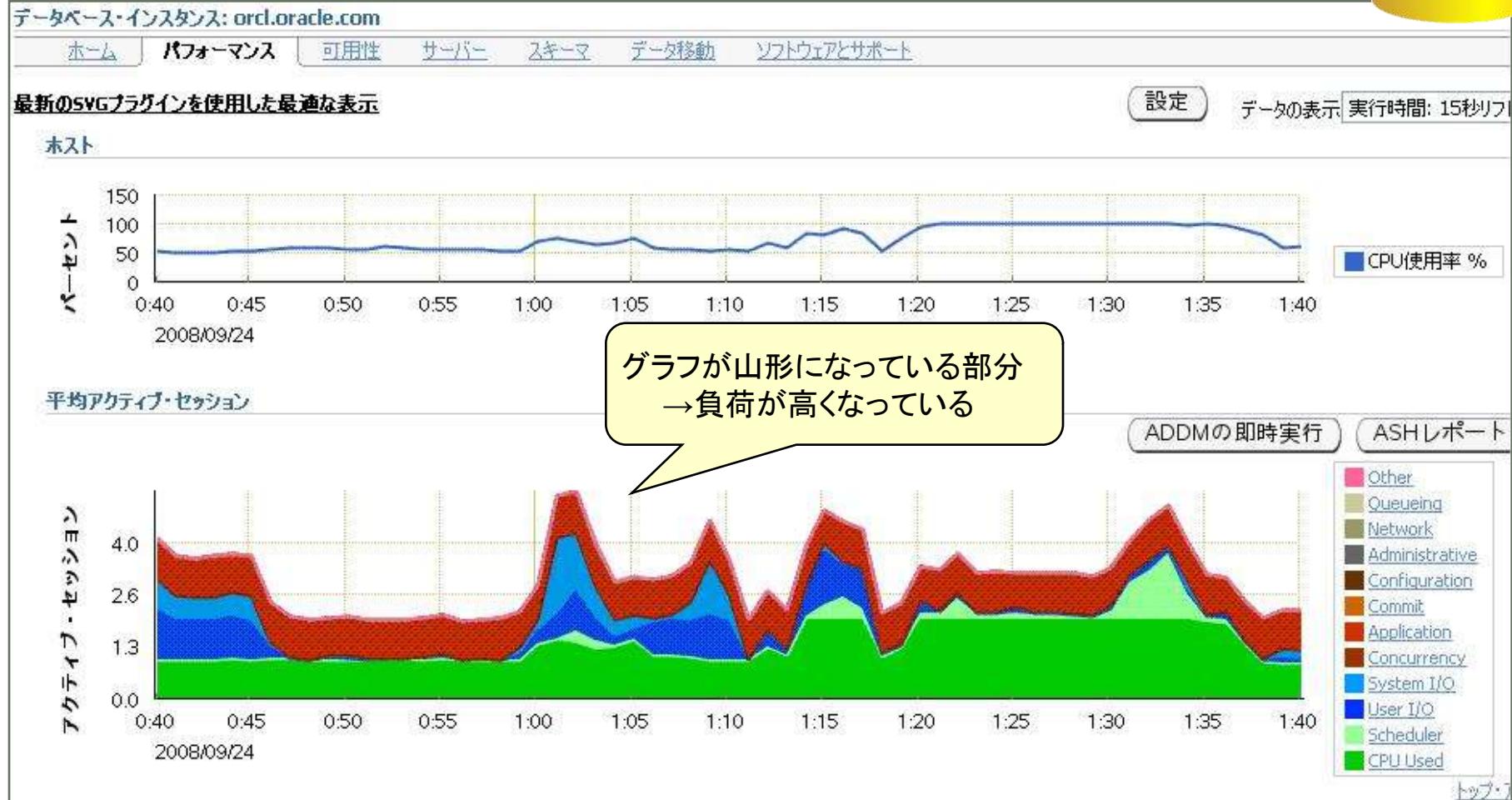
ORACLE

# 例1:複雑な結合のチューニング

負荷が高くなっている



- 該当のSQL文を実行し、Enterprise Managerから監視



# 例1:複雑な結合のチューニング

負荷の高い  
作業を確認！

- 負荷の高いSQL文を調べるために、トップ・アクティビティ画面に遷移



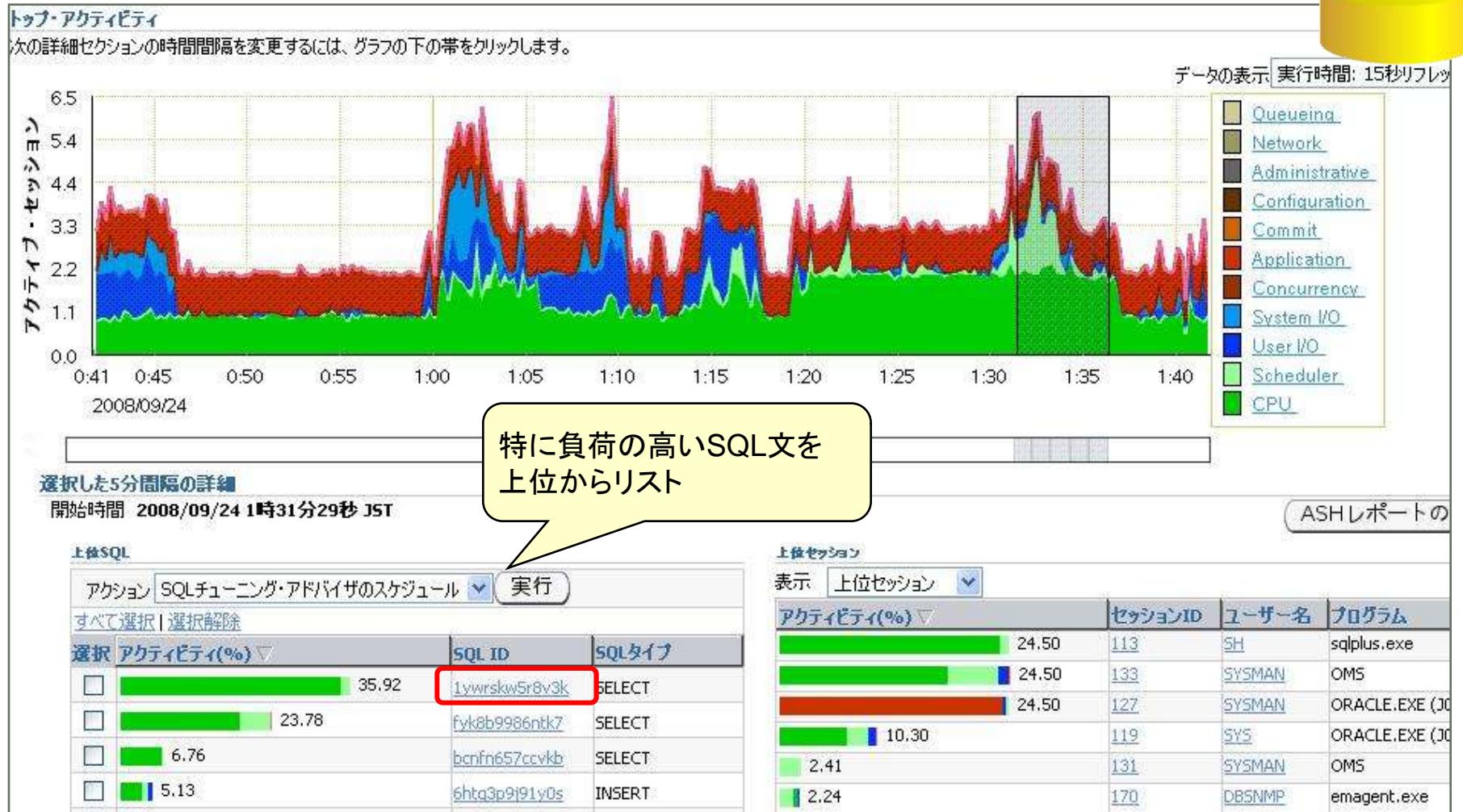
「トップ・アクティビティ」の画面から  
特に負荷の高い作業を監視

ORACLE

# 例1:複雑な結合のチューニング

負荷の高い  
処理を特定！

- 「トップ・アクティビティ・ページ」から該当のSQL文を特定



ORACLE

# 例1:複雑な結合のチューニング

SQL文の  
詳細は?

- 該当のSQL文を確認
- SQLチューニング・アドバイザを実行

**SQLチューニング・アドバイザ:**  
SQL文を分析し、改善アドバイスを提示する  
データベースのアドバイス機能

SQLの詳細: 1ywrskw5r8y3k

SQL IDに切替え  実行 実行時間: 手動リフレッシュ リフレッシュ SQLワークシート SQLチューニング・アドバイザのスケジュール

データの表示

▼テキスト

```
SELECT p.prod_name, sum(s.quantity_sold)
  FROM sales s, channels ch, customers c, products p, promotions pr
 WHERE s.channel_id = ch.channel_id AND s.cust_id = c.cust_id AND s.prod_id = p.prod_id AND s.promo_id =
 pr.promo_id AND EXISTS
 (SELECT 1
   FROM channels ch2
  WHERE ch.channel_id = ch2.channel_id AND ch2.channel_id = 3) AND EXISTS
 (SELECT 1
   FROM customers c2
  WHERE c.cust_id = c2.cust_id AND c2.cust_city = 'Kyoto') AND EXISTS
 (SELECT 1
   FROM products p2
  WHERE p.prod_id = p2.prod_id AND p2.prod_name = 'Deluxe Mouse') AND EXISTS
 (SELECT 1
   FROM promotions pr2
  WHERE pr.promo_id = pr2.promo_id AND pr2.promo_id = 999) group by p.prod_name
```

ORACLE

# 例1:複雑な結合のチューニング

- SQLチューニング・アドバイザの実行

SQLチューニング・アドバイザのスケジュール

次のパラメータを指定して、SQLチューニング・アドバイザの実行ジョブをスケジュールします。

\*名前: SQL\_TUNING\_1222188369570

説明: [入力欄]

▶SQL文

有効範囲

合計時間の制限(分): 30

分析の有効範囲:  包括  
この分析にはSQLプロファイル推奨が含まれますが、長時間かかることがあります。  
文ごとの時間制限(分): 5

スケジュール

タイムゾーン: Asia/Tokyo

即時

後で

日付: 2008/09/24

時間: 1:45:00 AM

発行

他の処理に影響が出ないように、  
時間制限をしたり、スケジューリングして  
後で実行することも可能

処理中: SQLチューニング・アドバイザのタスク SQL\_TUNING\_1222188369570

SQLチューニング・アドバイザのタスクを実行中です。前のページに戻るには戻るボタンをクリックします。SQLチューニング・アドバイザのタスクは実行して処理されます。「アドバイザ・セントラル」ページでそのステータスを確認し、推奨を表示できます。現在の実行を中断するには、「割込み」ボタンをクリックします。

ステータス: EXECUTING  
SQL ID: 1ywrskw5r8v3k  
経過時間(秒): 0

起動済: 2008/09/24 1:48:51  
時間制限(秒): 1800

SQL文: [入力欄]

新規SQLチューニング・タスクの作成  
0 / 1 SQL文を実行します。  
0% 増強された潜在パフォーマンスがアーカイブされました。

ヒント: ベースウインドウを開いてもプロセスは取り消されません。

ORACLE®

# 例1:複雑な結合のチューニング

どれを  
実装する?



- SQLチューニング・アドバイザの結果
- 推奨項目の確認

## SQLテキスト

```
SELECT p.prod_name,sum(s.quantity_sold) FROM sales s,channels  
AND s.prod_id = p.prod_id AND s.p...
```

## 推奨の選択

元の実行計画(注釈付き)

実装

- SQLチューニング・アドバイザの推奨
- 統計の再取得
  - SQLプロファイルの実装
  - 検索対象表に索引を作成

選択タイプ	結果	推奨	論理	新規 ベネフィ ット(%)	実行 計画	実行計 画の比 較
<input type="radio"/>	統計	表"SH"."SALES"のオプティマイザ統計は失効しています。	この表およびその索引に対するオプティマイザ統計の収集を検討してください。			
<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 type="radio"/>	索引	索引を1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の実行か、推奨される索引の作成を検討してください。 SH.CUSTOMERS("CUST_CITY") SH.PRODUCTS("PROD_NAME") SH.SALES("CUST_ID")	66.74		

これらを実装することによる  
実行計画の変化を確認できる

ORACLE

# 例1:複雑な結合のチューニング

効果が  
ありそうだ…

- SQLプロファイル実装による実行計画の比較

## 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 2211206465

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
▼ SELECT STATEMENT	0			25		6,934	8,482,315	101,788	79,692,496,896
▼ HASH GROUP BY	1			24		6,934	8,482,315	101,788	79,692,496,896
▼ NESTED LOOPS SEMI	2			23		18,945	8,482,314	101,788	79,670,280,192
▼ NESTED LOOPS SEMI	3			20		9,620,522	8,366,366	100,397	78,582,980,608
▼ NESTED LOOPS	4			17		224,101,531	4,192,423	50,310	43,570,847,744

## SQLプロファイルのある新しい実行計画

計画ハッシュ値 507535506

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
▼ SELECT STATEMENT	0			27		6,934	1,886	23	194,536,416
▼ HASH GROUP BY	1			26		6,934	1,886	23	194,536,416
▼ HASH JOIN RIGHT SEMI	2			25		9,668	1,885	23	172,355,936
TABLE ACCESS FULL	3	SH.CUSTOMERS	TABLE	1		1,318	405	5	30,562,460
▼ NESTED LOOPS	4			24		4,878,037	1,479	18	124,827,920
▼ NESTED LOOPS	5			22		4,591,094	1,474	18	13,172,517
▼ NESTED LOOPS	6			14		0.072	4	1	63,989
▼ NESTED LOOPS	7			11		0.043	3	1	55,647

ORACLE

# 例1:複雑な結合のチューニング

これも効果がありそう…



- 索引実装による実行計画の比較

## 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 2211206465

索引を作成することにより  
このSQL文の処理にかかる時間が大幅に短縮されることが分かる

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	バイト	コスト	時間	CPUコスト
▼ SELECT STATEMENT	0			25	6.934	8,482,315	101,788	79,692,496,896
▼ HASH GROUP BY	1			24	6.934	8,482,315	101,788	79,692,496,896
▼ NESTED LOOPS SEMI	2			23	18.945	8,482,314	101,788	79,670,280,192
▼ NESTED LOOPS SEMI	3			20	9,620,522	8,366,366	100,397	78,582,980,608
▼ NESTED LOOPS	4			17	224,101,531	4,192,423	50,310	43,570,847,744
▼ NESTED LOOPS	5			15	203,728,656	4,192,065	50,305	35,643,359,232

## 索引のある新しい実行計画

計画ハッシュ値 3673996401

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	バイト	コスト	時間	CPUコスト
▼ SELECT STATEMENT	0			27	0.098		627.8	27,414,236
▼ HASH GROUP BY	1			26	0.098		627.8	27,414,236
▼ NESTED LOOPS	2			25				
▼ NESTED LOOPS	3			23	0.098		626.8	5,263,323
▼ NESTED LOOPS	4			21	0.092		131.2	1,130,079
▼ NESTED LOOPS	5			18	0.062		130.2	1,121,737
▼ MERGE JOIN CARTESIAN	6			16	0.058		130.2	1,119,837

ORACLE

# 例1:複雑な結合のチューニング

実装ボタンで  
簡単に実装！



- SQLプロファイルの実装

推奨の選択  
元の実行計画(注釈付き)

実装

選択タイプ	結果	推奨	論理	新規実行計画	実行計画の比較
<input type="radio"/> 統計	表"SH"."SALES"のオプティマイザ統計は失効しています。	この表およびその索引に対するオプティマイザ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オプティマイザ統計が必要です。		
<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 type="radio"/> 索引	索引を1つ以上作成すると、この文の実行計画を改善できます。	物理入出力マ設計を改善する 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。 SH.CUSTOMERS("CUST_CITY") SH.PRODUCTS("PROD_NAME") SH.SALES("CUST_ID")	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 LOAD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・オーバーヘッドおよび追加領域消費が考慮された包括的な索引推奨事項を取得できます。	66.74	●●● ●●●

「実装」ボタンを押すことによって  
推奨事項を実装し、パフォーマンスを  
改善できる



ORACLE

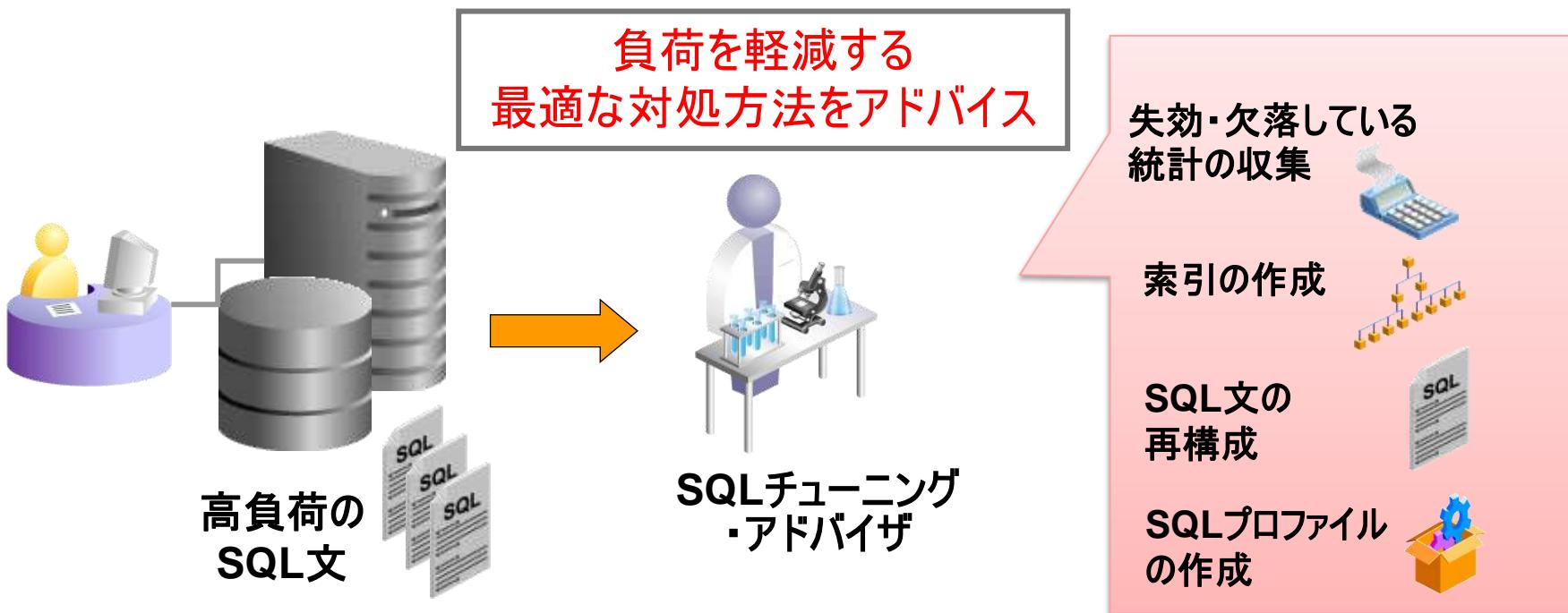
# 【補足】SQLチューニング・アドバイザ

EE

Diag

Tun

- 高負荷で問題となる特定のSQL文を診断し、負荷を軽減する最適な対処方法を提示する機能



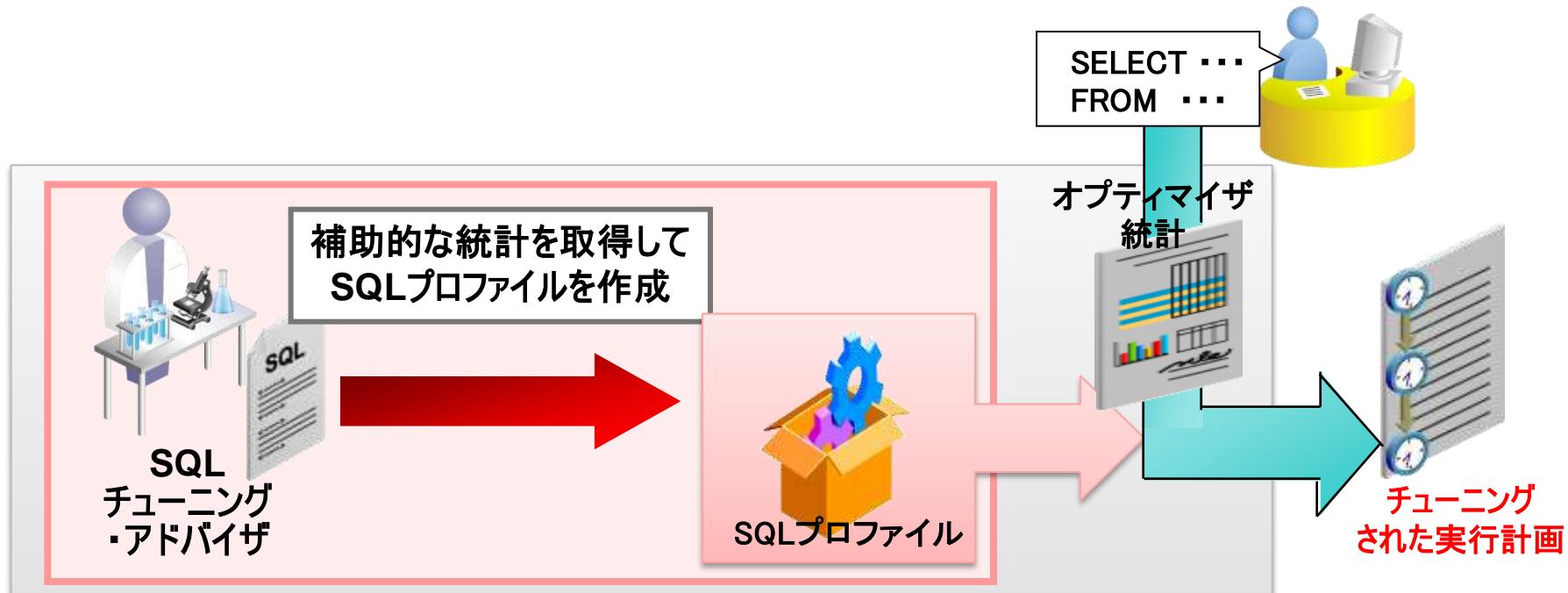
# 【補足】SQLプロファイル

EE

Diag

Tun

- SQLごとに取得する固有の補助的な統計情報
- 標準モードのオプティマイザでは収集しない追加情報を収集し、より最適な実行計画を生成
- コードの変更が不要なので、パッケージアプリケーションにも有効



ORACLE

## 例2:構文に問題のあるSQL文のチューニング

```
SELECT prod_name FROM products  
WHERE price*1.05 = 210;
```

索引を作っているのに、  
索引をつかっていないようなのだけど…?  
なぜ?



# 例2: 構文に問題のあるSQL文のチューニング

- 問題のSQL文を確認するために、Enterprise Managerから該当セッションを検索

実行しているユーザーが特定できる場合  
ユーザーから検索することも可能

スルーフット I/O パラレル実行 サービス

その他の監視リンク

ASHからの上位セッションおよび上位SQLのデータは、「トップ・アクティビティ」ページで検索できます。

- トップ・アクティビティ
- 上位コンシューマ
- 重複したSQL
- ブロックしているセッション
- ハング分析
- インスタンス・ロック
- ハイカウント・アクティビティ
- セッションの検索**
- SQLの検索
- スナップショット
- AWRベースライン
- SQLチューニング・セット
- SQLパフォーマンス・アナライザ

ホーム パフォーマンス 可用性 サーバー スキーマ データ移動 ソフトウェアとサポート

セッションの検索

検索

検索基準を指定

フィルタ DBユーザー  test

DBユーザーの場合、検索は大/小文字を区別しません。完全一致検索または大/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みます。他のフィルタの

結果

SID	DBユーザー	プログラム	サービス	モジュール	アクション	クライアント	マシン
113	TEST	sqlplus.exe	SYS\$USERS	SQL*Plus			JP-ORACLE#TSG

このSQL文を  
調査しよう

は常に使用できま



# 例2: 構文に問題のあるSQL文のチューニング

- 該当セッションの詳細を表示

セッションの詳細: 113 (TEST)  
ターゲットからの収集 2008/09/24 14時37分52秒 JST

「前のSQL」=そのセッションで一番最後に実行されたSQL文

セッションの中断 SQLトレース

一般 アクティビティ 統計 オープン・カーソル ブロックしているソース 待機イベントの履歴 パラレルSQL

**サーバー**

現行のステータス <b>INACTIVE</b>
シリアル番号 <b>2745</b>
DBユーザー名 <b>TEST</b>
OSプロセスID <b>5932</b>
ログイン時間 <b>2008/09/24 14:36:20</b>
ログイン維続時間 <b>1:34 (mm:ss)</b>
接続タイプ <b>DEDICATED</b>
タイプ <b>USER</b>
リソース・コンシューマ・グループ <b>使用不可</b>

**クライアント**

OSユーザー名 <b>TSG#tsg</b>
OSプロセスID <b>2456:3948</b>
ホスト <b>JP-ORACLE#TSG</b>
端末 <b>TSG</b>
現行のクライアントID <b>使用不可</b>
現行のクライアント情報 <b>使用不可</b>

**アプリケーション**

現行のSQL <b>なし</b>
現行のSQLコマンド <b>UNKNOWN</b>
前のSQL <b>fs94wnvx32d4x</b>
最終コールの維続時間 <b>1:17 (mm:ss)</b>
SQLトレース <b>DISABLED</b>
現行のSQLトレース・レベル <b>1</b>
待機情報を使用したトレース <b>DISABLED</b>
バインド情報を使用したトレース <b>DISABLED</b>
オープン・カーソル <b>2</b>
プログラム <b>sqlplus.exe</b>
サービス <b>SYS\$USERS</b>
現行のモジュール <b>SQL*Plus</b>
現行のアクション <b>使用不可</b>

**待機**

現在の待機イベント <b>SQL*Net message from client</b>
現行の待機クラス <b>Idle</b>
待機の維続時間 <b>1:18 (mm:ss)</b>

**その他**

解析スキーマ <b>TEST</b>
フェイルオーバー・タイプ <b>NONE</b>
フェイルオーバー・メソッド <b>NONE</b>

このSQL文を  
調査しよう



# 例2: 構文に問題のあるSQL文のチューニング

- SQL文の詳細情報(実行計画等)表示

SQLの詳細: fs94wnvx32d4x

SQL ID(切り替え) 実行 実行時間: 手動リフレッシュ リフレッシュ SQLワークシート SQLチューニング・アドバイザのスケジュール データの表示

テキスト

```
SELECT prod_name
  FROM products
 WHERE price*1.05 = 210
```

実行計画を確認すると、全表走査(索引を使わず、すべてのデータにアクセスして検索)されている

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 1954719464

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 2008/09/24 14:38:39 (UTC+09:00) 解析スキーマ TEST オプティマイザ・モード ALL\_ROWS

追加情報

すべて聞く | すべて閉じる

操作	オブジェクト	順序	行バイト	コスト	CPU (%)	問合せブロック名/オブジェクト名
▼ SELECT STATEMENT		2		10,735	100	
TABLE ACCESS FULL	PRODUCTS	1	60,000	2.06M	10,735	1 0:2:9 SEL\$1 / PRODUCTS@SEL\$1

リライトの説明の表示

操作

オブジェクト

順序

行バイト

コスト

CPU (%)

問合せブロック名/オブジェクト名

▼ SELECT STATEMENT

TABLE ACCESS FULL

PRODUCTS

1

60,000

2.06M

10,735

1 0:2:9 SEL\$1 / PRODUCTS@SEL\$1

索引があるのに使われていない?



ORACLE

# 例2: 構文に問題のあるSQL文のチューニング

- SQLチューニング・アドバイザの実行

SQLチューニング・アドバイザのスケジュール

次のパラメータを指定して、SQLチューニング・アドバイザの実行ジョブをスケジュールします。

\*名前 SQL\_TUNING\_1222234761984

説明

▶SQL文

有効範囲

合計時間の制限(分) 30

分析の有効範囲  包括  
この分析にはSQLプロファイル権限が含まれますが、長時間かかることがあります。  
文ごとの時間制限(分) 5

スケジュール

タイムゾーン Asia/Tokyo

即時

後で

日付 2008/09/24

時間 2 39 00 AM



他の処理に影響が出ないように、  
時間制限をしたり、スケジューリングして  
後で実行することも可能

処理中: SQLチューニング・アドバイザのタスク SQL\_TUNING\_1222234761984

SQLチューニング・アドバイザのタスクを実行中です。前のページに戻るには「戻る」ボタンをクリックします。SQLチューニング・アドバイザのタスクは実行して処理されます。「アドバイザ・セントラル」ページでそのステータスを確認し、推奨を表示できます。現在の実行を中断するには、「削除」ボタンをクリックします。

ステータス: EXECUTING	起動済 2008/09/24 14:39:56
SQL ID: 1s94wwvv32d4x	時間制限(分) 1800
経過時間(秒) 0	

SQL文

✓ 新規SQLチューニングタスクの作成  
0 / 1 900文書を実行します。  
0% 累積された潜在バネフィットがアーカイブされました。

※ヒント: ページのウインドウを閉じてもプロセスは削除されません

ORACLE

# 例2: 構文に問題のあるSQL文のチューニング

## • SQLチューニング・アドバイザの結果 推奨の表示

SQL IDの推奨:fs94whvwx32d4x

推奨される実装が1つ見つかっています。

SQLテキスト

```
SELECT prod_name FROM products WHERE price*1.05 = 210
```

索引が使われるよう、SQL文の構文を変更することが推奨

TEST.PRODUCTS("PRICE"\*\*1.5)  
という式を含んだ索引を推奨  
式を含んだ索引=ファンクション索引

物理スキーマ設計を改善する  
Access Advisorの実行か、推奨さ  
れる索引の作成を検討してく  
ださい。  
TEST.PRODUCTS("PRICE"\*\*1.05)

タイプ	結果	推奨	論理
<input checked="" type="radio"/> 索引	索引を1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の実行か、推奨さ れる索引の作成を検討してく ださい。 TEST.PRODUCTS("PRICE"**1.05)	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 LOAD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・オーバーヘッドおよび追加領域消費が考慮 された包括的な索引推奨事項を取得できます。
<input type="radio"/> SQL の再 構築	述語"PRODUCTS", "PRICE"**1.05=210(実行計画の行ID 1)は、 索引付けされた列"PRICE"に式が含まれています。この式のために オプティマイザは表"TEST", "PRODUCTS"の索引を選択できません。	索引を活用するため、述語を同 等の形式にリライトしてください。また は、式にファンクション索引を作成し てください。	述語が非等価性条件の場合や、索引付けされ た列のデータ型変換が存在する場合、オプティマ イザは索引を選択できません。

述語"PRODUCTS", "PRICE"\*\*1.05=210(実行計画の行ID 1)は、  
索引付けされた列"PRICE"に式が含まれています。この式のために  
オプティマイザは表"TEST", "PRODUCTS"の索引を選択できません。

索引を活用するため、述語を同  
等の形式にリライトしてください。また  
は、式にファンクション索引を作成し  
てください。

構文が悪くて  
索引が使わ  
れて  
いなか  
つた！



ORACLE

# 例2: 構文に問題のあるSQL文のチューニング

- ファンクション索引実装による実行計画の比較

## 実行計画の比較

### 元の実行計画(注釈付き)

計画ハッシュ値 1954719464

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
▼ SELECT STATEMENT	0					2,109,375	10,735	129	2,019,822,976
TABLE ACCESS FULL	1	TEST.PRODUCTS	TABLE		1	2,109,375	10,735	129	2,019,822,976

ファンクション索引を作成することにより  
このSQL文の処理にかかる時間が大幅に  
短縮されることが分かる

### 索引のある新しい実行計画

計画ハッシュ値 672487663

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
▼ SELECT STATEMENT	0					3	0.035	4 1	28,876
▼ TABLE ACCESS BY INDEX ROWID	1	TEST.PRODUCTS	TABLE		2	0.035	4 1	28,876	4
INDEX RANGE SCAN	2	IDX\$\$_00440001	INDEX						21,564

推奨された  
索引の効果は?



ORACLE

# 例2: 構文に問題のあるSQL文のチューニング

- ファンクション索引の実装

推奨の選択

元の実行計画(注釈付き)

実装

索引の作成を選択し、実装ボタンで実装

選択	タイプ	結果	推奨	論理
<input checked="" type="radio"/>	索引	索引を1つ以上作成すると、この文の実行計画を改善できます。	物理入キーマ設計を改善する Access Advisorの実行か、推奨される索引の作成を検討してください。 TEST.PRODUCTS("PRICE"**1.05)	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ワーカードを使用した"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・オーバーヘッドおよび追加領域消費が考慮された包括的な索引推奨事項を取得できます。
<input type="radio"/>	SQLの再構築	述語"PRODUCTS", "PRICE"**1.05=210(実行計画の行ID 1)は、索引付けされた列"PRICE"に式が含まれています。この式のためにオプティマイザは表"TEST", "PRODUCTS"の索引を選択できません。	索引を活用するために、述語を同等の形式にリライトしてください。または、式にファンクション索引を作成してください。	述語が非等価性条件の場合や、索引付けされた列に式または暗黙的データ型変換が存在する場合、オプティマイザでは索引を使用できません。

SQL表示

```
declare
cmd varchar2(400);
sname varchar2(400);
begin
cmd := 'create index TEST.IDX$$_00440001 on TEST.PRODUCTS("PRICE"**1.05)';
EXECUTE IMMEDIATE cmd;
END;
```

CREATE INDEX 索引名  
ON TEST.PRODUCTS("PRICE"\*\*1.05)

索引作成時に式を指定



ORACLE

# 【補足】WHERE句に式を含む条件の注意点

- 式を含む条件指定の注意:

条件となる列に対して、式・関数・演算子を使用して集計した場合、通常の索引が使用できない

```
SELECT prod_name FROM products  
WHERE price*1.05=210 ;
```



## 解決方法

- SQL文の構文変更  $price=210/1.05$
- ファンクション索引

# 【補足】ファンクション索引

- 関数(ファンクション)や式(計算等)を含んだ索引
  - 式の処理結果に対して索引を作成したもの
  - WHERE句にファンクションや式を含む文を効率化
  - Oracle8iから使用可能
  - コストベース・オプティマイザ(CBO)のみで利用可能

例: データベースの中の文字データが大文字か小文字か分からぬ場合

```
SELECT * FROM employees
WHERE UPPER(first_name) = 'RICHARD' ;
```



```
CREATE INDEX uppercase_idx
ON employees (UPPER(first_name));
```

## 例3:待機の発見と解決

一つずつのSQL文はチューニングしたのに  
性能が出ない！なぜ？



### 順序表(number\_table)から番号を順次採番

- ①採番用の表から最新の番号を取得  
(順番が抜かされないように、FOR UPDATEでロック)

```
SELECT * FROM number_table FOR UPDATE;
```

- ②「現在の値+1」を最新の順序番号として登録

```
UPDATE number_table
SET current_number=current_number + 1,
last_updated=localtimestamp;
```

# 例3:待機の発見と解決

待機の原因は  
なに?



- Enterprise Managerで状況を監視



ORACLE

# 例3:待機の発見と解決

ADDMに  
分析させてみよう



- ADDMを実行して、状況を分析

**ADDM:**  
データベース全体のパフォーマンス状況を分析  
し、状況診断・改善アドバイスをする機能

ADDMの即時実行

平均アクティブ・セッション

アクティブ・セッション

13:16 13:20 13:25 13:30 13:35 13:40 13:45 13:50 13:55 14:00 14:05 14:10 14:15  
2008/09/25

確認

新規のAWRスナップショットを作成し、このスナップショットおよび以前のスナップショットでADDMを実行しますか。

いいえ はい

ADDMを自動的に実行するスナップショットを取得中です

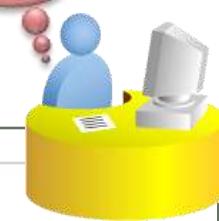
ADDMを実行するスナップショットを取得中

The screenshot shows a performance trend chart for 'Average Active Sessions' over time on 2008/09/25. The chart shows a sharp increase in activity starting around 14:00. A callout box defines ADDM as a function to analyze database performance and provide diagnostic and improvement advice. A confirmation dialog box is displayed, asking if a new AWR snapshot should be created for ADDM execution. The bottom panel indicates that an automatic snapshot for ADDM execution is being obtained.

ORACLE

# 例3:待機の発見と解決

ロックによる待機が発生していた！



- ADDMの分析結果

自動データベース診断モニター(ADDM)

データベース・アクティビティ

グラフ下の選択済アイコンは、ADDM分析期間を示しています。異なる分析期間を選択するには、別のアイコンをクリックしてください。

アクティブ・セッション  
2008/09/24 16:00 ~ 25:00

待機 (Orange)  
ユーザーI/O (Blue)  
CPU (Green)

④ ヒント このページで使用するアイコンおよび記号の説明は、次を参照してください: アイコンキー

ADDMパフォーマンス分析

タスク名 ADDM:1193534749\_1\_49

タスク所有者 SYS 平均アクティブ・セッション 3.8

影響(%) ▾

98.9	結果
97.2	行ロック待機

結果

行ロック待機

DB時間別の上位SQL

行ロックが発生していることが判明

ORACLE

# 例3:待機の発見と解決

具体的に  
どの処理でロックが  
発生している?



- ADDMの分析結果詳細

パフォーマンス結果の詳細: 行ロック待機

結果	行ロック待機を待機中のSQL文が見つかりました。	<a href="#">結果履歴</a>
影響(アクティブ・セッション)	3.78	
影響(%)	98.9	
期間開始時間	2008/09/25 14時21分26秒 JST	
持続期間(分)	6.1	
フィルタ処理済	いいえ <a href="#">[フィルタ]</a>	

競合が発生している表と  
待機しているSQL文が判明

推奨

すべての詳細を表示 | すべての詳細を非表示

詳細 カテゴリ ベネフィット(%) ▾

非表示 Application Analysis 98.9

アクション 重大な行の競合がTABLE "TEST.NUMBER\_TABLE"(オブジェクトID 70964)で検出されました。所定のロックSQLを使用して、アプリケーション・ロジック内の行の競合の原因をトレースしてください。

データベース・オブジェクト [TEST.NUMBER\\_TABLE](#)

論理 SQL\_ID "f9uwy02txajz6" のSQL文が、行ロックにおいてロックされました。

SQLテキスト [SELECT \\* FROM number\\_table FOR UPDATE](#)

SQL ID f9uwy02txajz6

論理 ID "110" のセッション、ユーザー ID "oo"、プログラム "sqlplus.exe" およびモジュール "SQL\*Plus" は、この推奨事項の利点の17%を占める、ロック・セッションでした。

ORACLE

# 例3:待機の発見と解決

ロックを取得している  
セッションを特定！

- 「ロックしているセッション」画面から現在のロック状況を確認



他の監視リンク

ASHからの上位セッションおよび上位SQLのデータは、「トップ・アクティビティ」ページで検索できます。

- トップ・アクティビティ
- 上位コンシューマ
- 重複したSQL
- ロックしているセッション**
- アクション
- インスタンス・ロック
- インスタンス・アクティビティ
- セッションの検索
- SQLの検索
- スナップショット
- AWRベースライン
- SQLチューニング・ナット
- SQLパフォーマンス・アナライザ

## ロックしているセッション

セッションの表示 セッションの中止

すべて開く | すべて閉じる

選択 ユーザー名	ロックされているセッション	セッションID	シリアル番号	SQL ID	待機クラス
<input type="radio"/> ▼ ブロックしているセッション					
<input checked="" type="radio"/> ▼ TEST		3	107	4962	Idle
<input type="radio"/> TEST		0	122	862	f9uwy02txajz6 Application
<input type="radio"/> TEST		0	109	2628	gwbmy541x90cr Application
<input type="radio"/> TEST		0	119	2592	f9uwy02txajz6 Application

一番上の「セッションID: 107」の処理が  
下の処理を待たせている

# 例3:待機の発見と解決

セッション同士の  
関係は？

- 「ハング分析」画面から現在のロック状況をグラフィカルに表示



## その他の監視リンク

ASHからの上位セッションおよび上位SQLのデータは、「トップ・アクティビティ」ページで検索できます。

- トップ・アクティビティ
- 上位エンクューマ
- 重複したSQL
- ロックしているセッション
- ハング分析
- インスタンス・ロック
- インスタンス・アクティビティ
- セッションの検索
- AWRベースライン
- SQLチューニング・セット
- SQLパフォーマンス・アナライザ

## ハング分析

インスタンスの待機グラフを下に表示します

表示されているセッションは即時にロックされた(緑)か、長期間待機中(黄)か、ハング(赤)しています。

合計セッション  
待機中セッション数 4

ブロックしているセッション数 3  
ルート・ブロッカ 1



ORACLE

# 例3:待機の発見と解決

問題のセッションが  
判明！

- 待機の原因となっているセッションの詳細を確認

プロックしているセッション

セッションの表示  セッションの中止

すべて開く | すべて閉じる

選択	ユーザー名	プロックしているセッション	セッションID	シリアル番号	SQL ID	待機クラス
<input type="radio"/>	▼ ブロックしているセッション					
<input checked="" type="radio"/>	▼ TEST		3	107	4962	Idle
<input type="radio"/>	TEST		0	122	862	f9uwy02txaiz6 Application
<input type="radio"/>	TEST					
<input type="radio"/>	TEST					

セッションの詳細: 107 (TEST)  
ターゲットからの収集 2008/09/25 14時17分14秒 JST

一般 アクティビティ 統計 オープン・カーソル ブロックしているリソース 待機イベントの履歴 パラレルSQL

サーバー クライアント

現行のステータス <b>INACTIVE</b>	OSユーザー名 <b>TSG¥tsg</b>
シリアル番号 <b>5031</b>	OSプロセスID <b>4108:5712</b>
DBユーザー名 <b>TEST</b>	ホスト <b>JP-ORACLE¥TSG</b>
OSプロセスID <b>5512</b>	端末 <b>TSG</b>
ログイン時間 <b>2008/09/25 14:02:00</b>	現行のクライアントID <b>使用不可</b>
ログイン継続時間 <b>15:13 (mm:ss)</b>	現行のクライアント情報 <b>使用不可</b>
接続タイプ <b>DEDICATED</b>	
タイプ <b>USER</b>	
リソース・コンシューマ・グループ <b>使用不可</b>	

ORACLE®

# 例3:待機の発見と解決

利用者が特定できれば  
処理を終わってもらっても

- 待機の原因となっているセッションを切断



ブロックしているセッション

選択	ユーザー名	ブロックされているセッション	セッションID	シリアル番号	SQL ID	待機クラス
<input type="radio"/>	▼ ブロックしているセッション					
<input checked="" type="radio"/>	▼ TEST	3	107	4962		Idle
<input type="radio"/>	TEST	0	122	862	f9uwyy02txajz6	Application
<input type="radio"/>	TEST	0	109	2628	gwbmy541x90cr	Application
<input type="radio"/>	TEST	0				

確認

このセッションを中断してもいいですか。

SID 107  
DBユーザー TEST  
プログラム sqlplus.exe  
オプション  ただちに中断  
 トランザクション後

SQL表示 いいえ はい

ORACLE

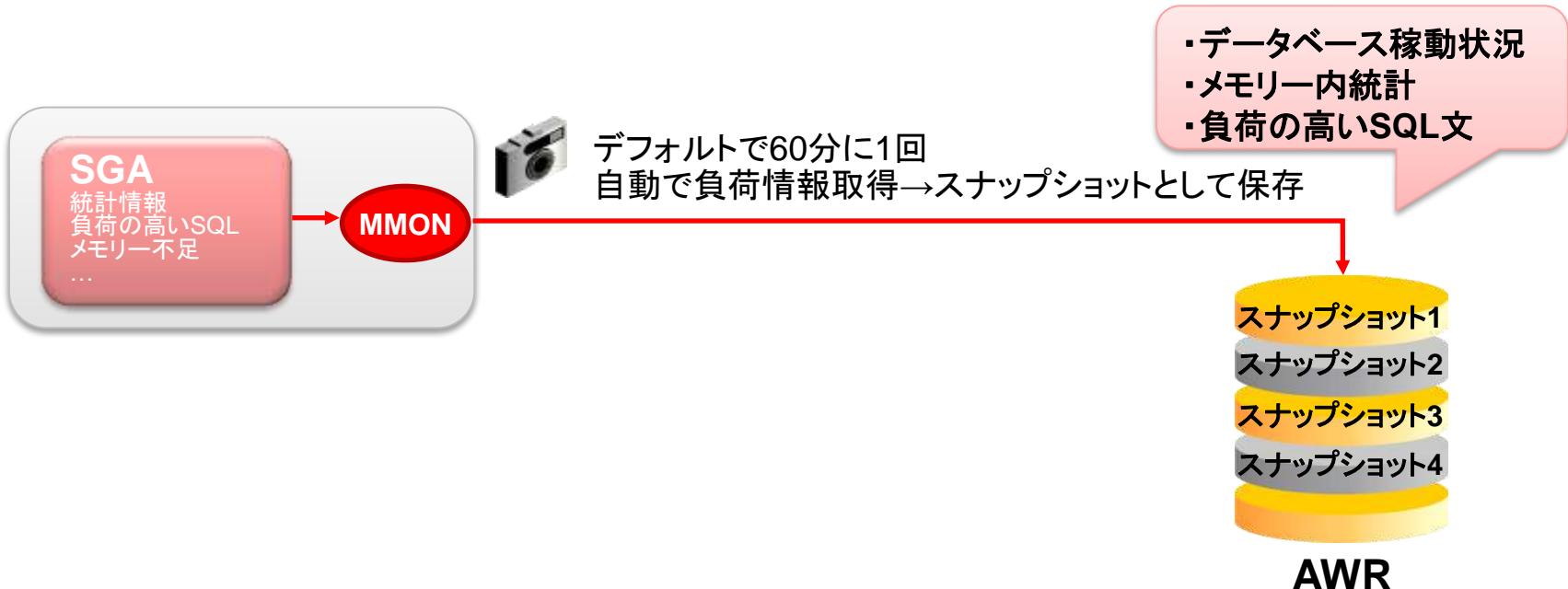
# 【補足】自動ワークロード・リポジトリ「AWR」

EE

Diag

- パフォーマンスの情報格納リポジトリ

- Oracle Databaseは、デフォルトで定期的にSGA内の負荷情報を収集し「スナップショット」としてAWR(Automatic Workload Repository)に格納
- AWRに格納されたスナップショットは、Oracle Databaseの自動管理機能の基礎情報として使用される

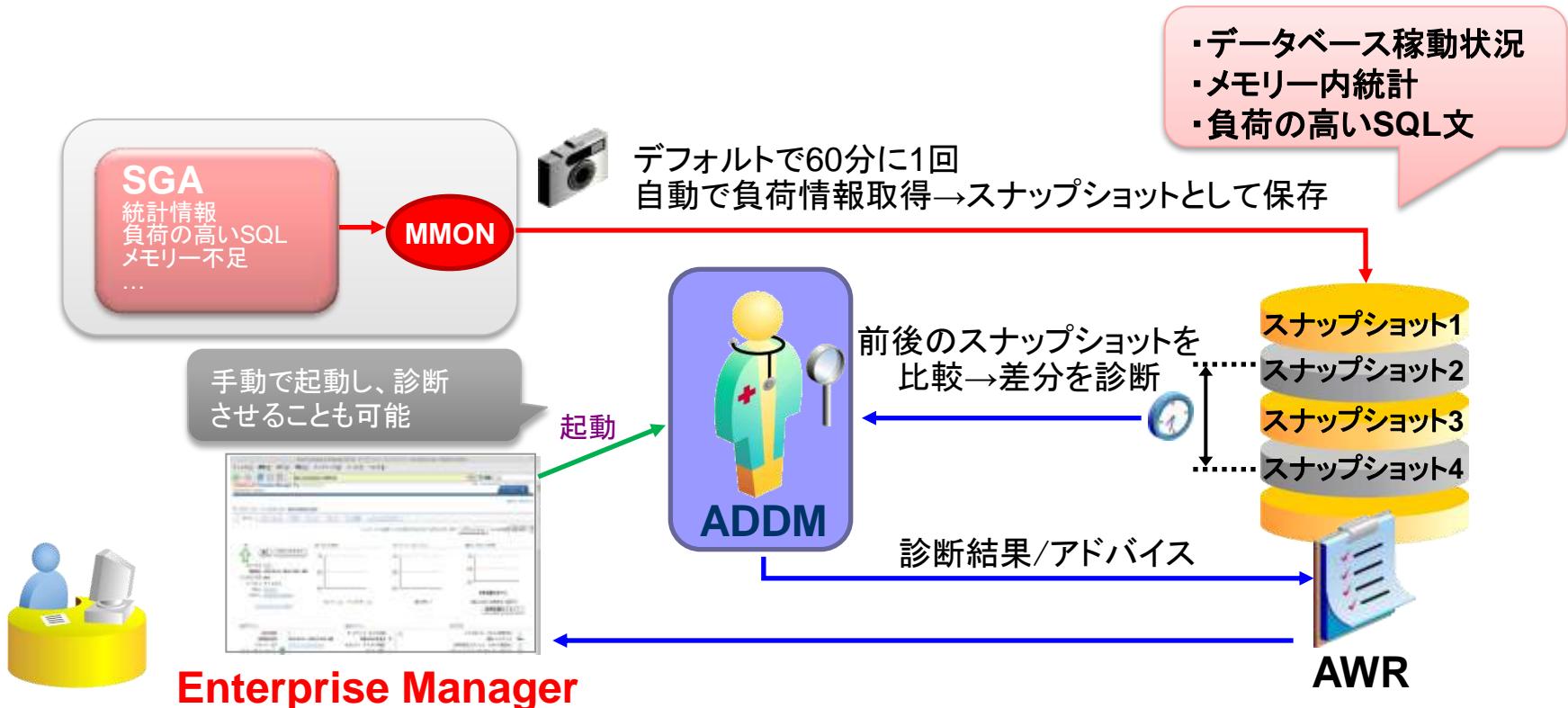


# 【補足】自動診斷機能「ADDM」

EE

## Diag

- **ADDM(Automatic Database Diagnostics Monitor)**
    - AWRスナップショットを分析し、自動的にデータベースを診断
    - 診断の結果、問題が発見されると、対処方法を管理者に提示



# 【補足】ADDM診断レポート例

EE

Diag

ADDMパフォーマンス分析  
タスク名 ADDM:1192317831\_1\_6

タスク所有者 SYS 平均アクティブ・セッション 2.9

期間開始時間 2008/09/04 15時00分39秒 JST 持続期間 (分) 60.6

影響(%)	結果	発生数(過去24時間)
100	CPU使用率	5/5
69.6	DB時間別の上位SQL	5/5
19.2	ハード解析	2/5
16.8	行ロック待機	1/5
10.1	I/Oスループット	3/5
9.8	I/Oの割合	5/5
3.5	サイズ不足のインスタンス・メモリー	

推奨

SQLチューニング・アドバイザのスケジュール

すべて選択 | 選択解除 | すべての詳細を表示 | すべての詳細を非表示

選択 詳細 カテゴリ

非表示 Host Configuration

アクション また Oracle Database Resource Manager を使用して、様々なコンシューマ・グループからのワークロードに優先度を付けることも検討してください。

アクション ホストに対するCPUの追加を検討するか、他のホスト上のデータベースを構成している

他のアドバイザや機能 (リソース・マネージャ) の推奨

推奨

すべての詳細を表示 | すべての詳細を非表示

詳細 カテゴリ

非表示 DB Configuration

アクション "memory\_target" パラメータを500Mに設定し、インスタンスに割り当てたメモリーを増やしてください。

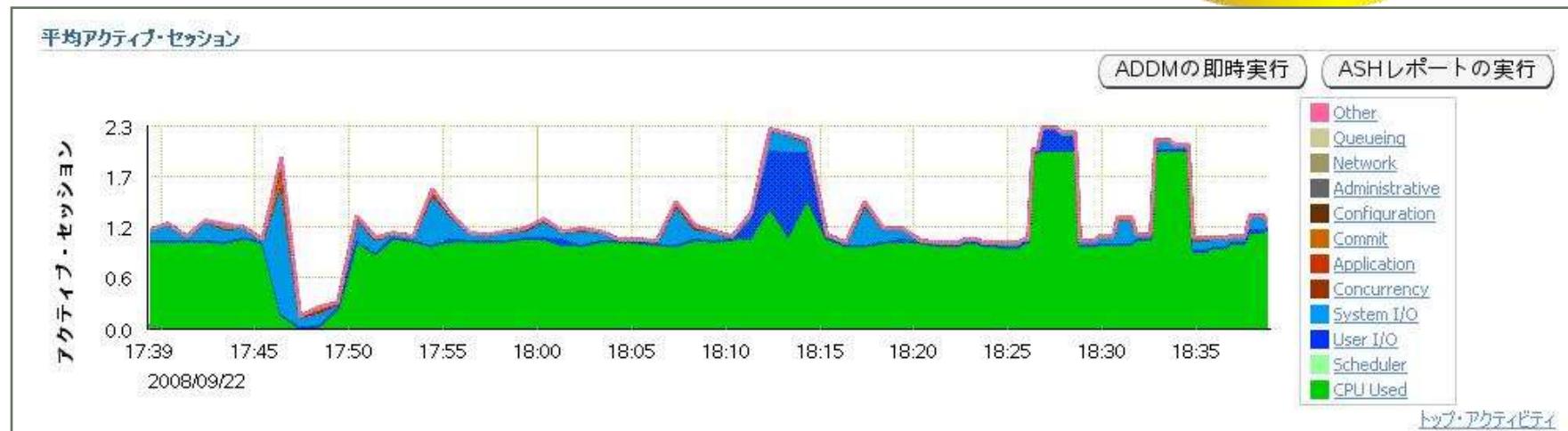
メモリの割り当てについて  
具体的なアドバイスを提示



ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

個々のSQL文はチューニング済み！  
でもまだパフォーマンスが悪い…



ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- トップ・アクティビティから負荷の高いSQL文を確認

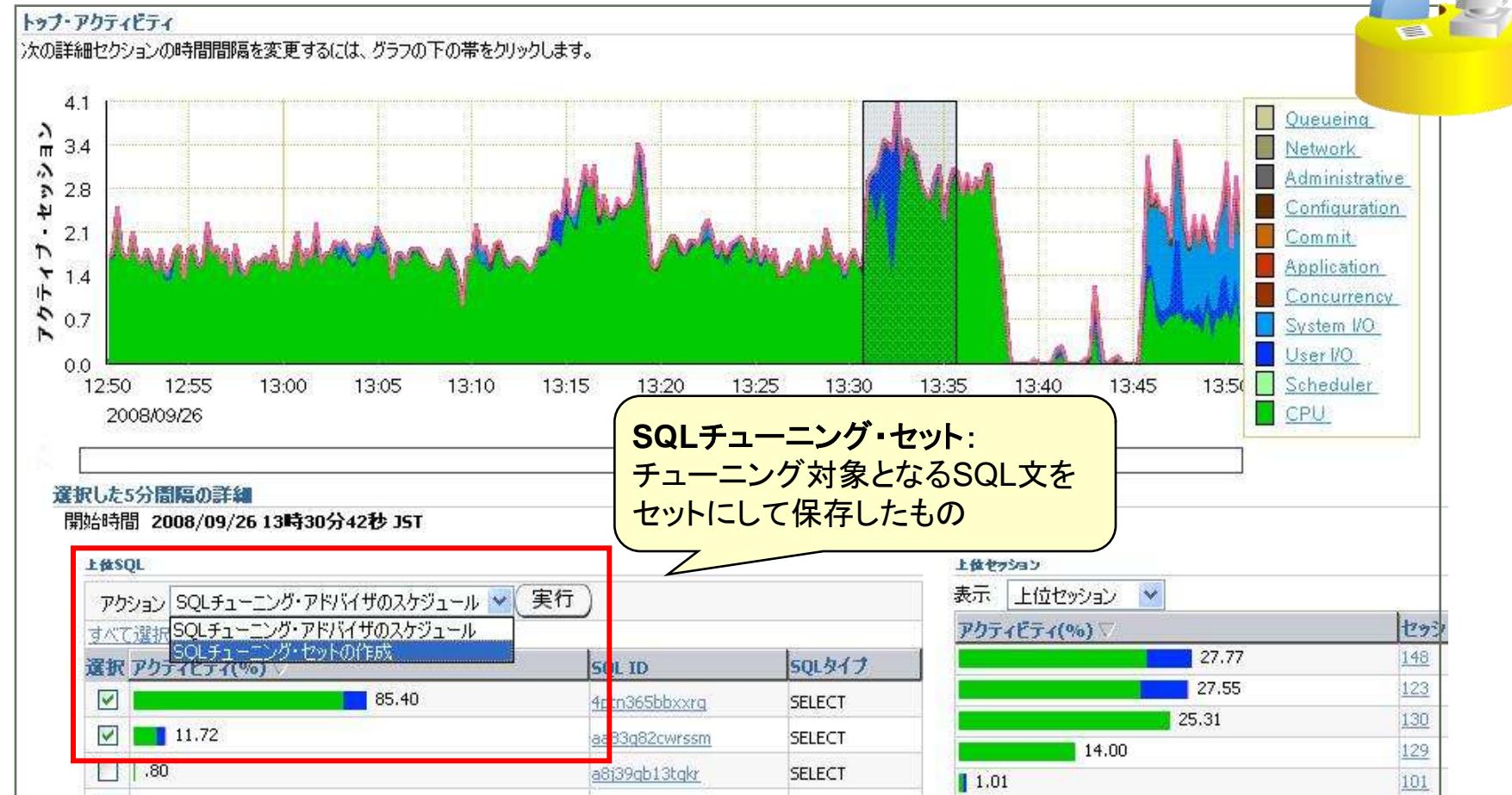
負荷が高い  
時間を調査



# 例4: 効率が悪い処理のパフォーマンス改善

高負荷SQLを  
まとめて分析

- 負荷の高いSQL文をまとめて「SQLチューニング・セット」を作成



ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- チューニング・セットをSQLアクセス・アドバイザで分析

アクセス・アドバイザで  
ワークロード分析



SQLチューニング・セットの作成

\*名前: TOP\_SQL\_1222404090359  
説明: 上位SQLによる自動生成

SQLテキスト

```
SELECT s.time_id sales_tid, c.time_id costs_tid FROM sales s, products p, costs c WHERE s.prod_id = p.prod_id AND c.prod_id = p.prod_id AND p.prod_name IN (SELECT prod_name FROM products)
SELECT c.cust_last_name, SUM(amount_sold) AS sum_amount_sold FROM customers c, sales s WHERE s.cust_id = c.cust_id GROUP BY c.cust_last_name
```

データベース・インスタンス: orcl.oracle.com > sysとしてログイン

確認: SQLチューニング・セットは正常に作成されました。

SQLチューニング・セット

SQLチューニング・セットは、チューニング目的に使用できるSQL文の集まりです。

検索: 名前または名前の一部のフィルタ 実行

SQLアクセス・アドバイザ:

一連の処理(ワークロード)を分析し  
パフォーマンスを改善できる構造(索引、マテリアライズド・ビュー、パーティション)を推奨する機能

作成 インポート

詳細	削除	エクスポート	SQLチューニング・アドバイザのスケジュール	SQLアクセス・アドバイザのスケジュール
選択	名前	スキーマ	説明	SQLの件数 作成 最終変更
<input checked="" type="radio"/>	TOP_SQL_1222404090359	SYS	上位SQLによる自動生成	2 08/09/26 13:41 08/09/26 13:41
<input type="radio"/>	TOP_SQL_1222403732468	SYS	上位SQLによる自動生成	1 08/09/26 13:35 08/09/26 13:35
<input type="radio"/>	TOP_SQL_1222403693218	SYS	上位SQLによる自動生成	2 08/09/26 13:34 08/09/26 13:34

ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- SQLアクセス・アドバイザでチューニングする対象ワークロードを指定

分析対象を指定



SQLアクセス・アドバイザ: ワークロード・ソース

データベース orcl.oracle.com  
ログイン時の権限 SYS

分析に使用するワークロードのソースを選択します。基礎となる表にアクセスするすべてのSQL文を代表するワークロードが、最も適しています。

現在と最近のSQLアクティビティ  
SQLがキャッシュから選択されます。

既存のSQLチューニング・セットを使用  
SQLチューニング・セット SYS.TOP\_SQL\_1222404090359

次のスキーマと表から仮想ワークロードを作成  
表にディメンションまたは主キー/外部キー制約が含まれていれば、アドバイザにより仮想のワークロードが作成されます。

スキーマと表

カソマ区切りリスト

追加

④ヒント そのスキーマに属するすべての表を指定する場合は、スキーマ名を入力します。

SQLアクセス・アドバイザで分析する対象を選択  
最近実行されたSQL文から自動選択したり、SQLチューニング・セットを指定することができる

ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- SQLアクセス・アドバイザで推奨する項目を指定

分析項目3つ  
から選択

SQLアクセス・アドバイザ: 推奨オプション

データベース orcl.oracle.com  
ログイン時の 権限 SYS

**推奨するアクセス構造**

- 索引
- マテリアライズド・ビュー (複雑な問合せの結果をあらかじめ集計して提供する、実体のあるビュー)
- パーティショニング (大規模な表を分割して管理するしくみ)

**有効範囲**

アドバイザは、制限モードまたは包括モードのどちらかで実行できます。制限モードでは、完全な分析が行われます。

制限  
分析は最もコストの大きい方に統合されます

包括  
完全な分析が行われます

SQLアクセス・アドバイザは以下の3つを推奨

- ・索引
- ・マテリアライズド・ビュー (複雑な問合せの結果をあらかじめ集計して提供する、実体のあるビュー)
- ・パーティショニング (大規模な表を分割して管理するしくみ)

SQLアクセス・アドバイザ: 確認

データベース orcl.oracle.com  
ログイン時の 権限 SYS

選択したSQLアクセス・アドバイザのオプションおよび値を確認してください。

タスク名	SQLACCESS52875530
タスクの説明	SQLアクセス・アドバイザ
スケジュール開始時間	即時実行

ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- SQLアクセス・アドバイザの結果確認



アドバイザ・タスク

検索  
結果セットに表示されるデータをフィルタ処理するには、アドバイザ・タイプを選択し、オプションでタスク名を入力してください。

アドバイザ・タイプ タスク名 アドバイザ実行 ステータス  
SQL Access Advisor 最後の実行 すべて 実行

デフォルトでは、検索を行うと、入力した文字列で始まるすべて大文字の一覧結果が表示されます。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。二重引用符で囲んだ文字列は、検索結果に表示されません。

結果  
結果の表示 削除 アクション サマリー ワークローク SQL文 詳細  
選択 アドバイザ・タイプ  
SQL Access Advisor

SQLアクセス・アドバイザの推奨により、パフォーマンスが改善することが分かる

改善の可能性

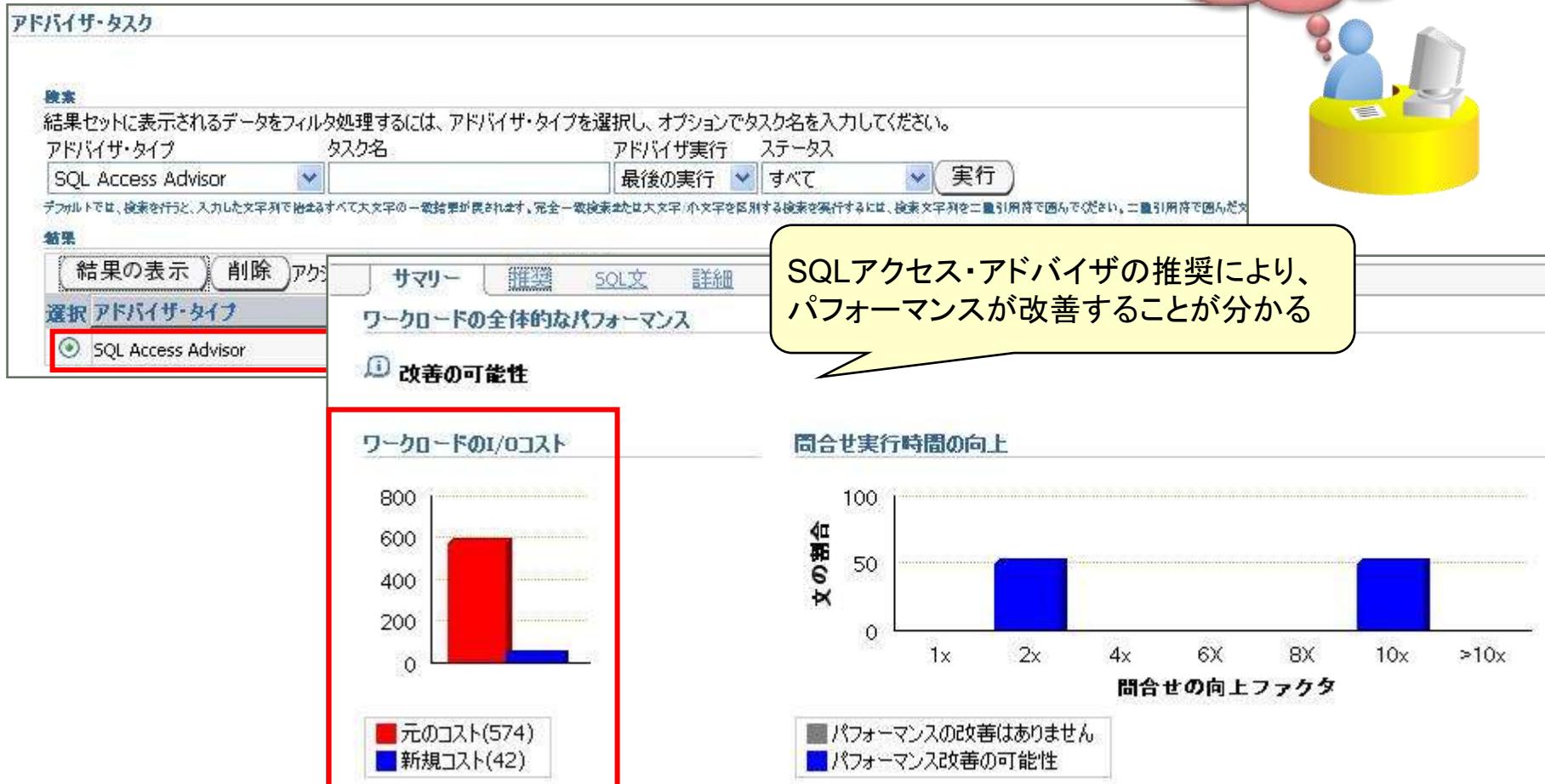
ワークロードのI/Oコスト

コスト	割合
元のコスト(574)	600
新規コスト(42)	100

間合せ実行時間の向上

間合せの向上ファクタ	割合
1x	50
2x	50
4x	0
8x	0
10x	0
>10x	0

■パフォーマンスの改善はありません  
■パフォーマンス改善の可能性



# 例4:効率が悪い処理のパフォーマンス改善

- SQLアクセス・アドバイザの結果 推奨項目の確認



このグラフと表には、最もコストが改善された順に推奨事項がリストされます。上位の推奨事項を実装すると、合計パフォーマンスが最も向上します。

コスト改善順の推奨事項

ID	コスト改善
1	525
2	7

実装用の推奨の選択

RETAINアクションを含む

推奨事項の詳細  スケジュール実装  SQL表示

マテリアライズド・ビューの作成が  
推奨されている

実装ステータス	ID	アクション	アクション・タイプ	コスト改善	コスト改善(%)
<input checked="" type="checkbox"/>	1	7	■ ■ ■ ■	525	98.68
<input checked="" type="checkbox"/>	2	4	■ ■ ■	7	1.32

ヒント凡例  索引  マテリアライズド・ビュー  マテリアライズド・ビュー・ログ  パーティション  その他

ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- SQLアクセス・アドバイザの結果 改善するSQL文

これらのSQL文が改善するようだ



このグラフと表には、最もコストが改善された順にSQL文がリストされます。上位のSQL文は、関連する推奨事項を実装することで最も向上します。

コスト改善順のSQL文

コスト

文ID

5005

5006

マテリアライズド・ビューを作成することにより、これらのSQL文のパフォーマンスが改善される

推奨ID	元のコスト	新規コスト	コスト改善
1	546	21	525
2	28	21	7

選択 文ID 文

5005 SELECT s.time\_id sales\_tid, c.time\_id costs\_tid FROM sales s, products p, costs c WHERE s.prod\_id = p.prod\_id AND c.prod\_id = p.prod\_id AND p.prod\_name IN (SELECT prod\_name FROM products)

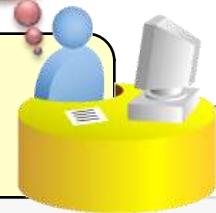
5006 SELECT c.cust\_last\_name, SUM(amount\_sold) AS sum\_amount\_sold FROM customers c, sales s WHERE s.cust\_id = c.cust\_id GROUP BY c.cust\_last\_name

ORACLE

# 例4:効率が悪い処理のパフォーマンス改善

- 実装するSQL文を確認し、実装

実装ボタンで  
さっそく作成！



改善するSQL文の選択

影響を受けるSQL文

スケジュール実装 SQL表示

すべて選択 | 選択解除

選択 文ID 文

5005 SELECT s.time\_id sales\_tid, c.time\_id costs\_tid FROM sales s, products p, costs c WHERE s.prod\_id = p.prod\_id AND s.time\_id = c.time\_id AND s.prod\_name = p.prod\_name FROM products

5006 SELECT c.cust\_last\_name, SUM(amount\_sold) AS sum\_amount\_sold FROM customers c, sales s WHERE c.cust\_id = s.cust\_id GROUP BY c.cust\_last\_name

**CREATE MATERIALIZED VIEW "SYS"."MV\$\$\_007C0000"**  
**REFRESH FAST WITH ROWID**  
**ENABLE QUERY REWRITE**  
**AS SELECT SH.PRODUCTS.ROWID C1, SH.COSTS.ROWID C2, SH.SALES.ROWID C3, "SH"."COSTS"."TIME\_ID" M1, "SH"."SALES"."TIME\_ID" M2 FROM SH.PRODUCTS, SH.COSTS, SH.SALES WHERE SH.SALES.PROD\_ID = SH.PRODUCTS.PROD\_ID **AND** SH.SALES.PROD\_ID = SH.COSTS.PROD\_ID;**

**確認**  
SQLアクセス・アドバイザの実装ジョブSYS.SQLACCESSIMPL4807292は正常に作成されました。

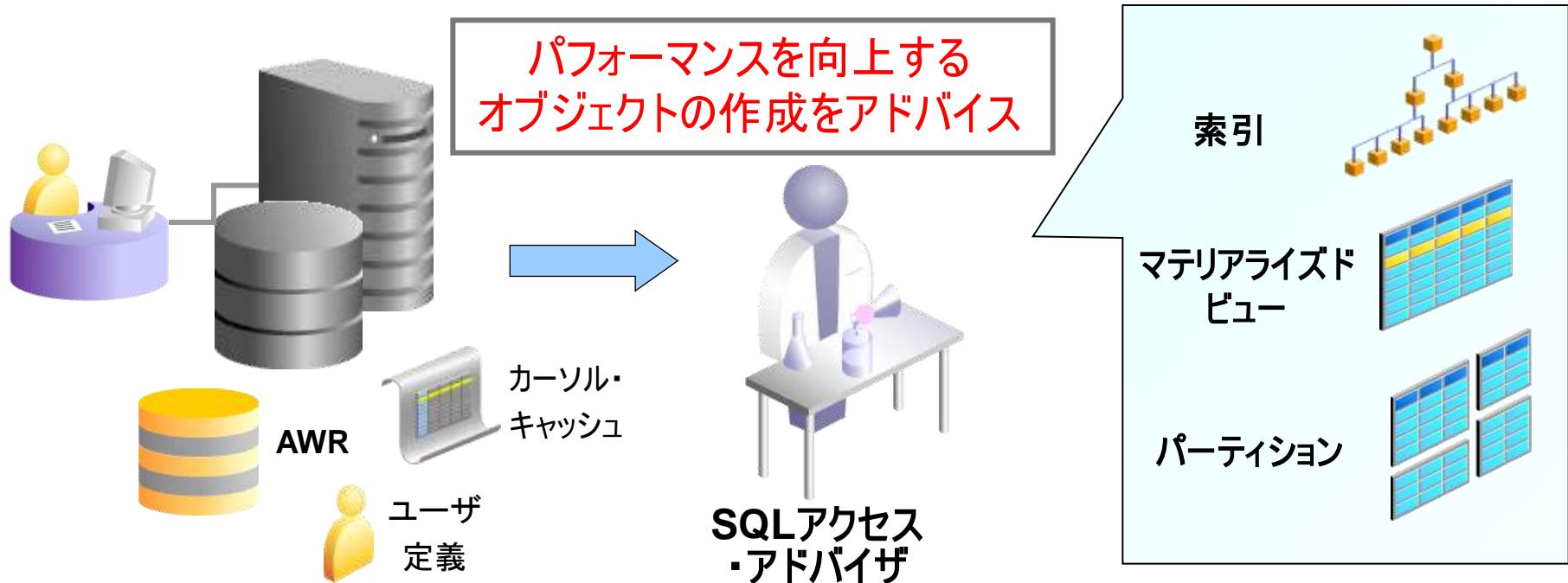
# 【補足】SQLアクセス・アドバイザ

EE

Diag

Tun

- ・ ワークロード(一連のSQL文)に対して最適なアクセスかどうかを診断
- ・ 適切な索引、マテリアライズド・ビュー、パーティションの作成をアドバイス



# 【補足】アドバイザのコマンドラインからの実行

- **PL/SQLパッケージ**により起動することが可能
- 下記の権限が必要
  - **SELECT ANY DICTIONARY**
  - **OEM\_ADVISOR**ロール
  - **EXECUTE ON SYS.DBMS\_SQLTUNE**

## ■バインド変数定義

```
VARIABLE sql_id VARCHAR2(13);
VARIABLE task_name VARCHAR2(64);
```

## ■直近のSQLのSQL\_IDを取得

```
begin
  SELECT prev_sql_id INTO :sql_id FROM v$session
  WHERE sid=sys_context('USERENV','SID');
end;
```

## ■SQL\_IDを指定してチューニング・タスクの作成

```
exec :task_name := DBMS_SQLTUNE.CREATE_TUNING_TASK(sql_id=>:sql_id)
```

## ■チューニング・タスクの実行

```
exec DBMS_SQLTUNE.EXECUTE_TUNING_TASK(:task_name)
```

## ■レポートの出力

```
SELECT DBMS_SQLTUNE.REPORT_TUNING_TASK(:task_name) FROM dual;
```

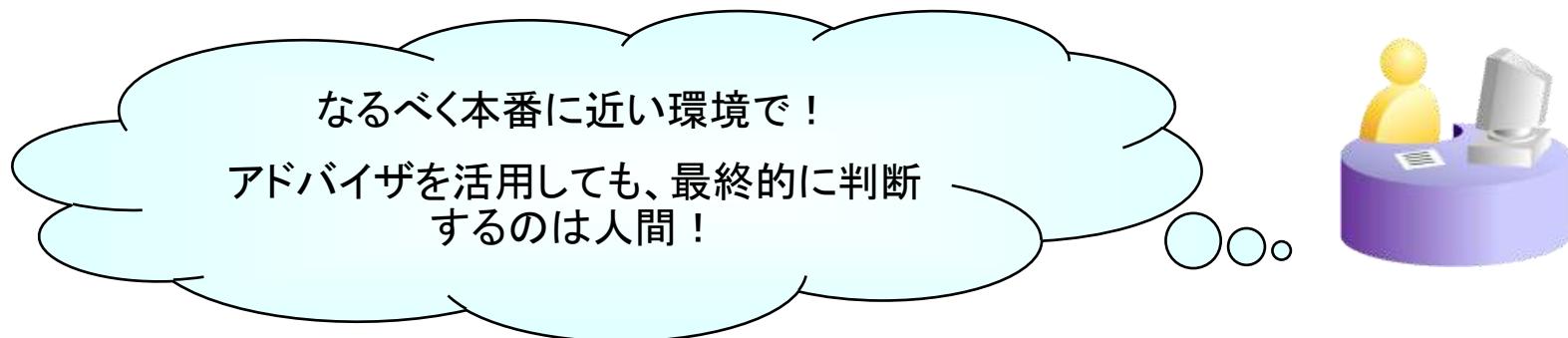
## ■チューニング・タスクの削除

```
exec DBMS_SQLTUNE.DROP_TUNING_TASK(:task_name)
```

ORACLE

# 【補足】テスト時の注意点

- 本番環境と開発環境のデータ量が大きく異なる場合、テスト環境のチューニング結果が正しく反映されない可能性がある
- 負荷が低すぎると、検出されない可能性がある
- アドバイザによる推奨を受け入れるかどうかを判断する必要
- 「影響(%)」の大きな項目から順に対応
- コーディング・ルールが不要になるわけではない



ORACLE®

# Oracle Enterprise Manager ご採用企業例



SOA インフラのデプロイが **92% 高速** に



データベースの管理にかかる時間と工数が **80% 削減**



オンラインショップでの収益損失を **25% 回避**



IT の生産性を **25% 改善**



資産の使用効率を **70% まで向上**



構成管理の工数を **90% 削減**



Oracle Enterprise Manager で **約2億円のコスト削減**



Oracle Enterprise Manager で年間 **1700万円のコスト削減**



手作業を自動化することにより、**50% の時間を削減**



データベースのテストにかかる時間を **90% 削減**



パッチ適用にかかる工数を **75% 削減**



アプリケーションのテストにかかる時間を **1週間 分削減**



アプリケーションのテストにかかる時間を **数週間から数時間に削減**



クリティカル・パッチの適用時間を **80% 削減**



Oracle Enterprise Manager により **24/7 の運用が可能** に

# 事例紹介

## 複数のデータベースを統合管理

### お客様 概要

- ・業種:通信業
- ・企業名:非公開(日本)
- ・従業員数:約4,000名
- ・対象業務:複数システムが対象

### 背景、課題

- ・50以上のデータベース環境を効率的に管理したい
- ・複数のデータベースがあるが、監視方法や監視項目がバラバラ
- ・問題発生時の対処の方法が属人的で、障害発生から検知、原因解決までの時間がかかる
- ・キャパシティ・プランニングができていなかった
- ・性能問題やリソース不足で発生する問題に対して都度の対応で後手後手に回ることが多かった

### 解決・利用方法

- ・Oracle Enterprise Managerによる統合管理
- ・自動監視と通知、ユーザー定義メトリック
- ・レポートによる見える化

### 導入効果

- ・全てのデータベースに対して、**同レベルの監視**が実施可能になり、新規にデータベースを追加する際にも、簡単に監視の対象に
- ・性能問題やリソース不足に対して、**プロアクティブな対応**が可能に
- ・「Oracle Enterprise Managerを運用の中核と位置づけて活用しています。**もうこれが無いと運用がまわらない**です。ついこの前も、とあるRACのロードバランスがうまく行われていないことをEMのレポートがあったから判明できました。どうしてもデータベースの数が多い(数十個)ので、**視覚化できない**と気づけないんですよね。」

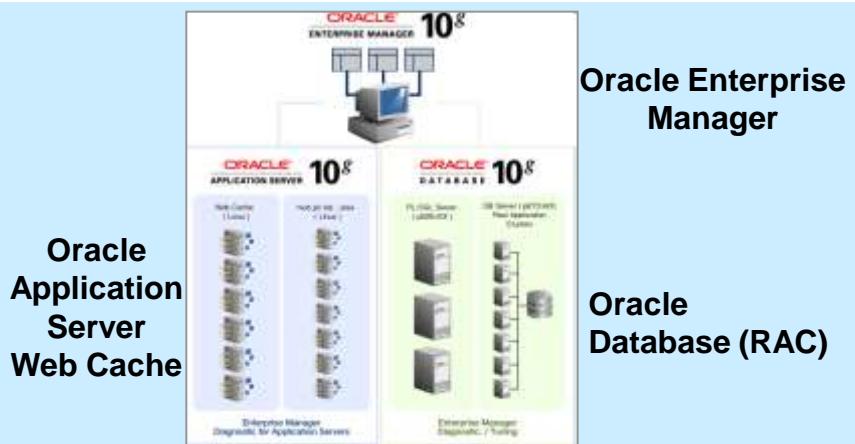
ORACLE

### お客様 概要

- ・業種: 旅行業
- ・企業名: 楽天トラベル株式会社様
- ・従業員数: 約250名
- ・対象業務: インターネット総合旅行サービス

### 背景、課題

- ・Oracle Real Application Clusters、Oracle Application Server の採用
- ・新しいアーキテクチャやコーディングスタイルを社内に浸透させ、きちんと管理、運用したい
- ・管理者の手間をかけず、社員がデータベース情報を見たい



### 解決・利用方法

- ・Oracle Enterprise Manager で統合管理と自動監視
- ・データベース情報の開示

### 導入効果

- ・管理コストの削減
- ・「役員レベルでも直接自分で状況が見られるので、さまざまなことの報告も簡単にできるようになりました。」
- ・「それまではスーパーDBAがあたかも魔法を使って管理していたのが、みんな魔法を使えるようになつた。Oracle Enterprise Manager 10gという魔法の杖で、チーム全体のレベルを底上げしてやっていけるという感じです。」

### お客様 概要

- ・業種: 情報・通信システムの統合サポート
- ・企業名: 日立電子サービス株式会社様
- ・従業員数: 約5,400名
- ・対象業務: 業務システム開発

### 背景、課題

- ・各部門のニーズに逐一対応すると開発量が倍増してしまうため、開発プロセスの品質を高め、手戻りやミスをなくす必要があった
- ・単に人を増員しても開発量の増大に対応できるわけではなく、人的コストを抑えるには生産性の向上も大きな課題
- ・極端なケースでは品質を確保するために詳細設計や基本設計まで戻ってやり直さなければならぬ場合もあるため、早い段階からテストを繰り返す必要があった

### 解決・利用方法

- ・Oracle Application Testing Suiteにより、機能テスト/負荷テスト/テスト工程管理の改善と効率化

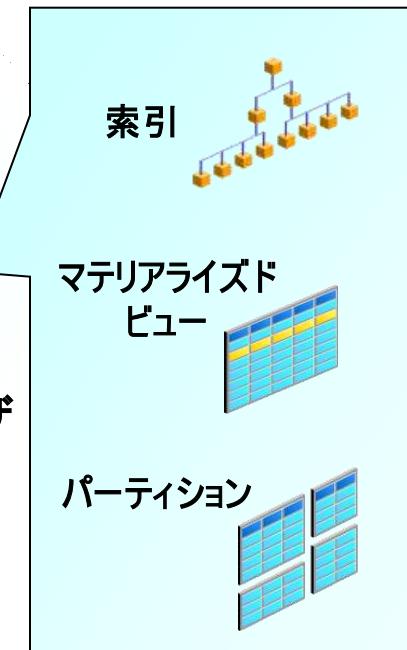
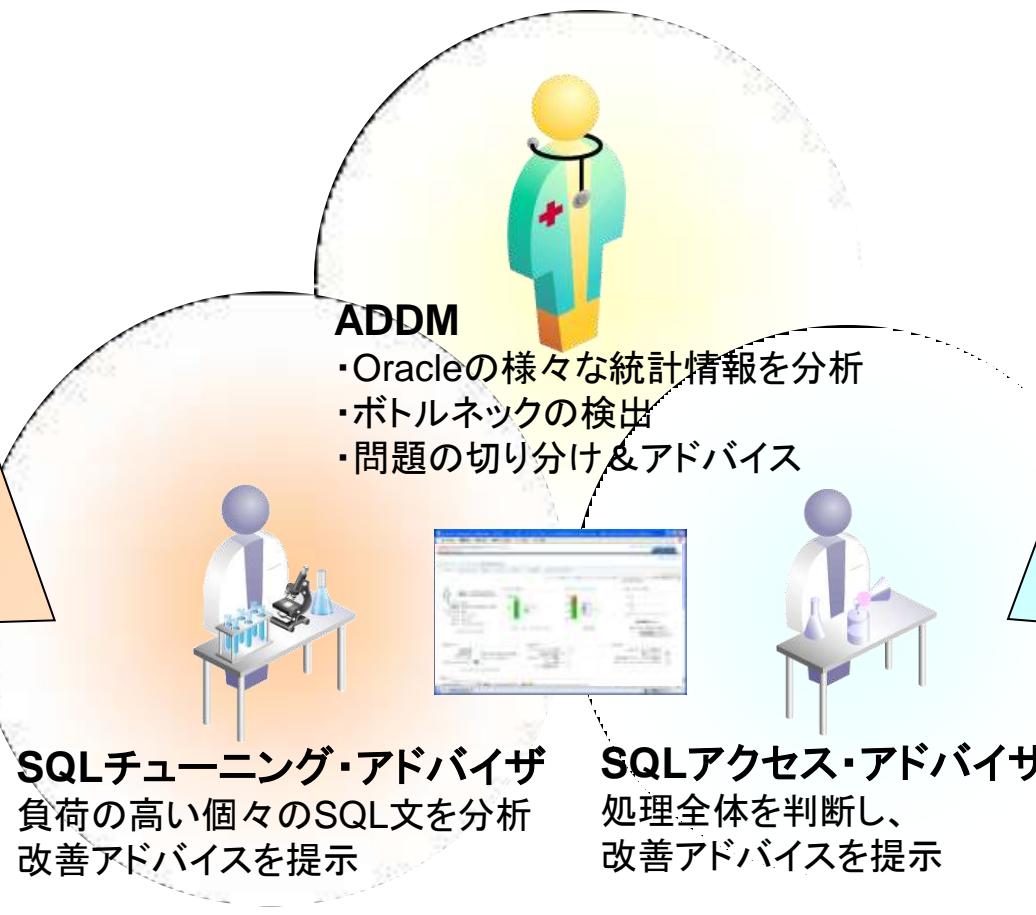
### Oracle の採用理由

- ・ユーザー視点のテストをGUIで容易に実現
- ・機能テストと負荷テストで同じスクリプトが利用できるので準備作業が削減できる
- ・充実したサポート体制

### 導入効果

- ・テストプロセスの自動化、簡略化によりシステム開発の品質と生産性が大幅に向上
- ・テスト項目を2倍に増やしてもテスト時間は30%短縮でき、テストの網羅性を高められたことが最大の効果
- ・テストのデータ分析とノウハウ蓄積による開発者の技術力向上および育成体制の確立

# まとめ



Enterprise Managerとアドバイザの活用により  
問題を開発・テスト段階で検出できる！

ORACLE

# OTN × ダイセミ でスキルアップ!!



- ・技術的な内容について疑問点を解消したい！
  - ・一般的なその解決方法などを知りたい！
  - ・セミナ資料など技術コンテンツがほしい！

**Oracle Technology Network(OTN)を御活用下さい。**

<http://otn.oracle.co.jp/forum/index.jspa?categoryId=2>

技術的な疑問点は、OTN掲示版の  
「データベース一般」へ

※OTN掲示版は、基本的にOracleユーザー有志からの回答となるため100%回答があるとは限りません。  
ただ、過去の履歴を見ると、質問の大多数に関してなんらかの回答が書き込まれております。

<http://www.oracle.com/technology/global/jp/ondemand/otn-seminar/index.html>

過去のセミナ資料、動画コンテンツはOTNの  
「OTNセミナー オンデマンド コンテンツ」へ

※ダイセミ事務局にダイセミ資料を請求頂いても、お受けできない可能性がございますので予めご了承ください。  
ダイセミ資料はOTNコンテンツ オン デマンドか、セミナ実施時間内にダウンロード頂くようお願い致します。

**ORACLE**

# OTNセミナー オンデマンド コンテンツ

期間限定にて、ダイセミの人気セミナーを動画配信中!!  
ダイセミのライブ感はそのままに、お好きな時間で受講頂けます。

最新のコンテンツ



 エンジニアのための ITIL実践術 再生時間: 60分	 ここからはじめよう Oracle PL/SQL入門 再生時間: 60分	 実践!!高可用システム構築 -RAC基本 再生時間: 60分	 お悩み解決! Oracle のサイジング 再生時間: 60分
--	--	---	---

Database



 今さら聞けない!?'バック アップ・リカバリ入 門 再生時間: 60分	 意外と簡単! Oracle Database 11g -セ 再生時間: 60分	 実践!!バックアップ ・リカバリ 再生時間: 60分	 意外と簡単! Oracle Database 11g -デ 再生時間: 60分
---	--	--	--

» もっと見る

OTN オンデマンド

検索

※掲載のコンテンツ内容は予告なく変更になる可能性があります。  
期間限定での配信コンテンツも含まれております。お早めにダウンロード頂くことをお勧めいたします。

ORACLE

# オラクル クルクルキャンペーン

あのOracle Database Enterprise Editionが超おトク!!



- ライセンス使用期間 を**5年間に**設定
- 初期のライセンスコストがなんと**67%OFF** !
- テクニカル・サポート価格も**53%OFF** !

Oracle Databaseの  
ライセンス価格を**大幅に抑えて**  
ご導入いただけます

- 多くのお客様でサーバー使用期間とされる  
5年間にライセンス期間を限定
- ・期間途中で永久ライセンスへ差額移行
  - ・5年後に新規ライセンスを購入し継続利用
  - ・5年後に新システムへデータを移行



Enterprise Editionはここが違う!!

- ・圧倒的なパフォーマンス!
- ・データベース管理がカンタン!
- ・データベースを止めなくていい!
- ・もちろん障害対策も万全!



- Oracle Databaseの機能を**存分に使える**!
- 2ノードRAC構成も可能!
- サーバー構成によって計4種類のパックから**選べる**!

詳しくはコチラ

<http://www.oracle.co.jp/campaign/kurukuru/index.html>

Oracle Direct 0120-155-096

お問い合わせフォーム

[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

ORACLE

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

検索



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、Oracle Direct Seminar申込時と同じ  
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、ご登録さ  
れている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



以上の事項は、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。

