

## 【プランスタビリティ（実行計画固定化）】

実行計画の固定化のための手段 ⇒SQL 文の処理高速化のため

### ストアアウトラインの活用目的

SQL 文が変更できないパッケージでの実行計画だけの変更

ストアアウトライン（オプティマイザヒントの集合セット）の作成方法

作成されたストアアウトラインを使った実行計画の変更方法

ストアアウトライン（オプティマイザヒントの集合セット）の使用

作成されているストアアウトラインの表示

## 【非推奨】

Oracle11g より現在は、ストアアウトラインの使用は非推奨となった

11 g からは、新たな機能として SQL Plan Management（SPM SQL 計画管理）が追加されて、この機能を使用するように推奨されている

ストアアウトラインとは、

まず、SQL 文を実行するためには、オプティマイザが SQL 文を解析して実行計画を作成します

現在使っている実行計画の計画内容を、SQL ヒントの状態記録したものが、ストアアウトラインになります

プランスタビリティの使用が有効にされたセッションでは、SQL 文を実行すると、「保存されている SQL ヒントが含まれたストアアウトライン」から、オプティマイザが実行計画を作成して、SQL 文の実行処理が行われる

このアーキテクチャーを利用して、「SQL ヒントが含まれたストアアウトライン」に明示的に作成した実行計画を保存することによって、意図した実行計画で SQL 文の実行を行うことが出来る

ストアダウトライン（オプティマイザヒントの集合セット）の保存先  
プランスタビリティは、OL\$、OL\$HINTS、OL\$NODES に保存されます

#### ストアダウトラインの活用目的

- SQL 文の実行において、（過去に使用していた）**明示的に指定した実行計画を使用させる** ことにある
- ストアドアウトラインを使用すると、SQL 文を修正することなく実行計画を固定出来ます
- また、ストアダウトラインに保存されているアウトラインヒントを**編集**して、実行計画を意図的に変更することが出来ます
- パッケージ・アプリケーションなどの **SQL 文が変更出来ない環境下で**、SQL 文の実行計画だけを**意図したテーブルへのアクセス方法**（アクセス・パス[Access Path]、結合順序[Join Order]、結合方法[Join Method]）**に変更する**ことが出来ます

## ストアアウトライン（オプティマイザヒントの集合セット）の作成方法

- ① ストアアウトライン作成の開始を宣言

```
alter session set create_stored_outlines = true;
```

- ② ストアアウトラインを作成する対象の SQL 文を実行

```
select * from emp where empno = 1;
```

- ③ ストアアウトライン作成の終了を宣言

```
alter session set create_stored_outlines = false;
```

※ この `create_stored_outlines` が `true` の間だけ、実行した SQL 文のストアアウトラインが作成される

## 作成されたストアアウトラインを使った実行計画の変更方法

作成されたストアアウトラインは、編集が可能です

## 作成されたストアアウトラインの使用方法（オプティマイザヒントの集合セット）

- ① ストアアウトラインの使用を宣言

```
alter session set use_stored_outlines = true;
```

- ② EXPLAIN PLAN で実行計画を確認

```
explain plan for select * from emp where empno = 1;  
@?/rdbms/admin/utlxpls.sql
```

- ③ ストアアウトラインの使用の終了を宣言

```
alter session set use_stored_outlines = false;
```

※ この `used_stored_outlines` が `true` の間だけ、実行しようとする SQL 文に対してストアアウトラインのヒントが適用され実行計画が作成される

`alter system set use_stored_outlines =` を利用すると、Oracle の全セッションでストアアウトラインが適用される

## 作成されたストアアウトラインを使った実行計画の変更方法

※ パッケージ・アプリケーションなどで、SQL 文の変更が出来ない。また、Oracle 接続時に `alter session set use_stored_outlines` も使えない場合での利用方法

1) 「ヒント句付き SQL 文」と「ヒント句無し SQL 文」で、ストアアウトラインを作成する

2) 「ヒント句無し SQL 文」のストアアウトラインを、「**ヒント句付き SQL 文**」のストアアウトラインで置換える

※ これで、「ヒント無し SQL 文」の実行計画が、意図した処理順序で実行されることになる

3) `alter system set use_stored_outlines = true ;`を使って、ストアアウトラインの適用範囲を Oracle 全体のセッションに適用する

※ SQL 文の変更が出来ない状態でも、意図した実行計画を適用させることができる

## 作成されているストアアウトラインの表示方法

```
select category, owner, name, used, timestamp, enabled, sql_text
       from dba_outlines ;
```

CATEGORY	NAME	USED	TIME_STAMP
DEFAULT	SYS_OUT_LINE_04112315391378506	UNUSED	04/11/23 15:39:13
ENABLED	SQL_TEXT		
ENABLED	select count(coll) from test1 where col = 30 (作成したストアアウトラインに対応している SQL 文)		

## ストアアウトラインが使用されたのかの確認方法

実行計画を表示させ、適用された場合には以下のような表示となる

```
SQL> @?/rdbms/admin/utlxpls.sql
```

```
PLAN_TABLE_OUTPUT
```

```
-----  
Plan hash value: 2949544139
```

```
-----  
| Id | Operation | Name | Rows | Bytes | Cost (%) |  
-----  
| 0 | SELECT STATEMENT | | 1 | 28 | 1 (0) |  
| 1 | TABLE ACCESS BY INDEX ROWID | EMP | 1 | 28 | 1 (0) |  
|* 2 | INDEX UNIQUE SCAN | PK_EMP | 1 | | 0 (0) |  
-----
```

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

```
-----  
2 - access("EMPNO">=1)
```

```
Note
```

```
-----  
- outline "SYS_OUTLINE_18052516533118201" used for this statement
```

※ ストアアウトラインが適用されたことを意味する