

## SQL 文の実行計画を別の SQL 文（ヒント句付き）の実行計画に置き換えて SPM の「SQL 管理ベース」へ登録（保存）する方法

⇒『ヒント句を使用して変更した SQL 文の実行計画』を、元の SQL 文の実行計画として置き換えて使用する方法

※ [Enterprise 版限定の機能]

### 手動登録

（作成済みの実行計画を SQL 計画ベースラインへ個別に登録する操作方法）

- 3) SQL 文の実行計画を、別の SQL 文（ヒント句付き）の実行計画に置き換えて SPM の「SQL 管理ベース」へ登録（保存）する方法

#### 【概要手順】

1. 性能の悪い現行の SQL 文の実行計画（PLAN-A）を、SQL 計画ベースラインとして登録する
2. ヒント句を追加した性能の良い SQL 文の実行計画（PLAN-B）を実行する
3. 元の実行計画（PLAN-A）を性能の良い SQL 文の実行計画（PLAN-B）に置換える  
（元の実行計画（PLAN-A）は、SQL 計画ベースラインから削除する

## 【詳細手順】

1. 性能の悪い現行の SQL 文の実行計画 (PLAN-A) を、SQL 計画ベースラインとして登録する
  - 1-1. アプリケーションを操作して、性能の悪い現行の SQL 文を実行する
  - 1-2. SQL\_ID と PLAN\_HASH\_VALUE を確認する
    - ※ SQL テキスト文が不明ならば、SQL トレースを取得する方法もある
  - 1-3. カーソルの実行計画を確認する
  - 1-4. 現行の対象 SQL 文の実行計画を SQL 計画ベースラインとして登録する
  - 1-5. 登録した SQL 計画ベースラインの SQL\_HANDLE と PLAN\_NAME を調査する
  
2. ヒント句を追加した性能の良い SQL 文の実行計画 (PLAN-B) を実行する
  - 2-1. ヒント句を追加して、SQL 文を実行する
    - ※ 但し、アプリケーションを実行するユーザーで実行を行うこと
  - 2-2. ヒント句を追加した SQL 文の SQL\_ID と PLAN\_HASH\_VALUE を確認する
  - 2-3. カーソルの実行計画を確認する
  
3. 元の実行計画 (PLAN-A) を性能の良い SQL 文の実行計画 (PLAN-B) に置換える (元の実行計画 (PLAN-A) は、SQL 計画ベースラインから削除する)
  - 3-1. 元の SQL 文の SQL\_HANDLE を指定して、性能の良い実行計画 (PLAN-B) を SQL 計画ベースラインに追加登録する
  - 3-2. 登録済みの SQL 計画ベースラインを確認する
  - 3-3. 元の実行計画 (PLAN-A) を、SQL 計画ベースラインから削除する
  - 3-4. 元の実行計画 (PLAN-A) だけが、SQL 計画ベースラインから削除されたことを確認する

## 【詳細手順 実操作】

1. 性能の悪い現行の SQL 文の実行計画 (PLAN-A) を、SQL 計画ベースラインとして登録する

- 1-1. アプリケーションを操作して、性能の悪い現行の SQL 文を実行する

```
Select * From Table1 Where <検索条件> ;
```

- 1-2. SQL\_ID と PLAN\_HASH\_VALUE を確認する

実行した SQL ステートメントを Where 条件に設定して、v\$sql ビューを検索し、SQL\_ID と PLAN\_HASH\_VALUE を求める  
ドキュメントの A) →v\$sql ビュー部分を参照

```
select SQL_TEXT, SQL_ID, PLAN_HASH_VALUE,  
       EXACT_MATCHING_SIGNATURE,  
       SQL_PLAN_BASELINE, CHILD_NUMBER  
from v$sql  
where SQL_TEXT like 'select . . . from . . . %' ;
```

※ SQL テキスト文が不明ならば、SQL トレースを取得する方法もある

- 1-3. カーソルの実行計画を確認する

1-2. で求めた SQL\_ID を Where 条件に設定して、DBMS\_XPLAN.DISPLAY\_CURSOR ファンクションを実行し、実行計画を求める

ドキュメントの E) →DBMS\_XPLAN.DISPLAY\_CURSOR ファンクション部分を参照

```
select * from table( dbms_xplan.display_cursor( '<SQL_ID 値>  
' ) );
```

- 1-4. 現行の対象 SQL 文の実行計画を SQL 計画ベースラインとして登録する

1-2. で求めた SQL\_ID と PLAN\_HASH\_VALUE を使って、dbms\_spm.load\_plans\_form\_cursor\_cache を実行し、実行計画を SQL 計画ベースラインとして登録する

ドキュメントの N) -1→dbms\_spm.load\_plans\_form\_cursor\_cache 部分を参照

「SQL 管理ベース」へ新規登録させる時のキー (sql_handle) 値は、ロジックにて算出される
---

```

set serveroutput on
declare
    ret number ;
begin
    ret := dbms_spm.load_plans_from_cursor_cache(
        sql_id => '<SQL_ID 値>',
        plan_hash_value => <PLAN_HASH_VALUE 値> );
    dbms_output.put_line(' LOAD PLANS : ' || ret ) ;
end;
/

```

1-5. 登録した SQL 計画ベースラインの SQL\_HANDLE と PLAN\_NAME を調査する

1-1. で実行した SQL ステートメントを Where 条件に設定して、dba\_sql\_plan\_baseline デictionary を検索して、実行計画が SQL 計画ベースラインに SQL\_HANDLE と PLAN\_NAME を求める  
 (表示されれば、SQL 計画ベースラインへの登録 OK となる)  
 ドキュメントの J) → dba\_sql\_plan\_baseline dictionary 部分を参照

```

select SQL_HANDLE , PLAN_NAME ,
       to_char( CREATED, 'YYYY/MM/DD HH24:MI:SS') CREATED ,
       ACCEPTED , ENABLED , SQL_TEXT, SIGNATURE ,
       REPRODUCED , to_char( LAST_EXECUTED ,
       'YYYY/MM/DDHH 24:MI:SS') LAST_EXECUTED
from dba_sql_plan_baselines
where SQL_TEXT like 'select . . . from . . . %'
order by CREATED ;

```

2. ヒント句を追加した性能の良い SQL 文の実行計画 (PLAN-B) を実行する

2-1. ヒント句を追加して、性能の良い SQL 文を実行する

**※ 但し、アプリケーションを実行するユーザーで実行を行うこと**

```

select /*+ FULL( e ) FULL( d ) LEADING( d ) USE_HASH( e ) */
       e.empno, e.ename, d.dname from emp e , dept d ;

```

2-2. ヒント句を追加した SQL 文の SQL\_ID と PLAN\_HASH\_VALUE を確認する

2-1. で実行した SQL ステートメントを Where 条件に設定して、v\$sql ビューを検索し、SQL\_ID と PLAN\_HASH\_VALUE を求める  
ドキュメントのA) →v\$sql ビュー部分を参照

```
select SQL_TEXT, SQL_ID, PLAN_HASH_VALUE,  
       EXACT_MATCHING_SIGNATURE,  
       SQL_PLAN_BASELINE, CHILD_NUMBER  
from v$sql  
where SQL_TEXT like 'select /*+ FULL( e ) FULL( d )  
                    LEADING( d ) . . . %' ;
```

2-3. カーソルの実行計画を確認する

2-2. で求めた SQL\_ID を Where 条件に設定して、DBMS\_XPLAN.DISPLAY\_CURSOR ファンクションを実行し、実行計画を求める

ドキュメントのE) →DBMS\_XPLAN.DISPLAY\_CURSOR ファンクション部分を参照

```
select * from table( dbms_xplan.display_cursor( '<SQL_ID 値>  
                    ' ) );
```

3. 元の実行計画 (PLAN-A) を性能の良い SQL 文の実行計画 (PLAN-B) に置換える (元の実行計画 (PLAN-A) は、SQL 計画ベースラインから削除する)

3-1. 元の SQL 文の SQL\_HANDLE を指定して、性能の良い実行計画 (PLAN-B) を SQL 計画ベースラインに追加登録する

2-2. で求めた SQL\_ID と PLAN\_HASH\_VALUE と 1-5. で求めた SQL\_HANDLE を使って、dbms\_spm.load\_plans\_form\_cursor\_cache を実行し、実行計画を SQL 計画ベースラインとして追加登録する

ドキュメントのN) -2 →dbms\_spm.load\_plans\_form\_cursor\_cache 部分を参照

```

set serveroutput on
declare
    ret number;
begin
    ret := dbms_spm.load_plans_from_cursor_cache(
        sql_id => '<SQL_ID 値>',
        plan_hash_value => <PLAN_HASH_VALUE 値> ,
        sql_handle => '<SQL_HANDLE 値>');
    dbms_output.put_line('LOAD PLANS : ' || ret) ;
end;
/

```

登録させるライブラリ・キャッシュ  
の実行計画を示すためのキー値

「SQL管理ベース」へ追加登録  
させるためのキー値

### 3-2. 登録済みの SQL 計画ベースラインを確認する

1-1. で実行した SQL ステートメントを Where 条件に設定して、  
dba\_sql\_plan\_baseline デictionary を検索して、実行計画が SQL 計  
画ベースラインに表示されるか確認する

ドキュメントの J) → dba\_sql\_plan\_baseline dictionary 部分を  
参照

```

select SQL_HANDLE, PLAN_NAME,
       to_char(CREATED, 'YYYY/MM/DD HH24:MI:SS') CREATED,
       ACCEPTED, ENABLED, SQL_TEXT, SIGNATURE,
       REPRODUCED, to_char(LAST_EXECUTED,
       'YYYY/MM/DDHH 24:MI:SS') LAST_EXECUTED
from dba_sql_plan_baselines
where SQL_TEXT like 'select . . . from . . . %'
order by CREATED ;

```

※ PLAN-A と PLAN-B の 2 件の登録が確認できること

### 3-3. 元の実行計画 (PLAN-A) を、SQL 計画ベースラインから削除する

1-5. で求めた SQL\_HANDLE と PLAN\_NAME を Where 条件に  
して、dbms\_spm.drop\_plans\_from\_cursor\_cache を実行し、SQL 計  
画ベースラインから PLAN-A の実行計画を削除する

ドキュメントの O) → dbms\_spm.drop\_sql\_plan\_baseline 部分を参照

```

set serveroutput on
declare
    ret number ;
begin
    ret := dbms_spm.drop_sql_plan_baseline(
        sql_handle => '<SQL_HANDLE 値>',
        plan_name => '<PLAN_NAME 値>'); ←
    dbms_output.put_line(' LOAD PLANS : ' || ret) ;
end;
/

```

効率が悪い方の実行計画の plan\_name 値  
 を指定する

3-4. 元の実行計画 (PLAN-A) だけが、SQL 計画ベースラインから削除されたことを確認する

1-1. で実行した SQL ステートメントを Where 条件に設定して、`dba_sql_plan_baseline` デictionaryを検索して、PLAN-B の実行計画だけが SQL 計画ベースラインに表示されるか確認する

ドキュメントの J) →`dba_sql_plan_baseline` dictionary部分を参照

```

select  SQL_HANDLE ,  PLAN_NAME ,
        to_char( CREATED, 'YYYY/MM/DD HH24:MI:SS')  CREATED ,
        ACCEPTED ,  ENABLED ,  SQL_TEXT,  SIGNATURE ,
        REPRODUCED ,  to_char( LAST_EXECUTED ,
        'YYYY/MM/DDHH 24:MI:SS')  LAST_EXECUTED
from  dba_sql_plan_baselines
where  SQL_TEXT like 'select . . . from . . . %'
order by  CREATED ;

```