ＡＷＲ（自動ワークリポジトリ）スナップショットの SQL 情報を基に、実行計画を SPM の「SQL 管理ベース」へ登録（保存）する方法

⇒『過去のＳＱＬ実行計画』を使って、SQL 文の処理を行わせる方法

SQL チューニング・セット（STS）をコマンドで作成する方法

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）

【ライセンス】

Oracle Diagnostic Pack と Oracle Tuning Pack ライセンスが必要となる

【概要手順】

1．SQL 文の性能が良かった過去の実行計画が AWR に存在することを確認する

2．過去の実行計画を AWR から STS（SQL チューニングセット）を作成する

3．STS（SQL チューニングセット）から SQL 計画ベースライン（承認済実行計画）を作成する

【詳細手順】

対象とする SQL 文の選定は、現場からの遅延報告に基づいて、遅くなった処理（トランザクション）を構成する SQL 文の処理時間を調査して決定する

1．SQL 文の性能が良かった過去の実行計画が AWR に存在することを確認する

1-1. AWR 自動ワークリポジトリから該当の SQL_ID を取得する

1-2. AWR 自動ワークリポジトリから該当の SQL の過去の PLAN_HASH_VALUE とスナップショットの ID を取得する

※ 1-2.で PLAN_HASH_VALUE が**複数存在していたら**、そのすべての実行計画の **AWR スナップショットの SNAP_ID** を抜き出し、「SQL 管理ベース」を登録（保存）する

そして、1つずつ対象の「SQL 管理ベース」の ACCEPT フラグを変更して、実行計画を実行する
実行した後に、その実行統計情報を比較して効率が良い方を判断する

1-3. AWR 自動ワークリポジトリから該当の SQL の実行計画を確認する

2．過去の実行計画を AWR から STS（SQL チューニングセット）を作成する

SQL チューニング・セット（STS）をコマンドで作成する方法

2-1. STS（SQL チューニングセット）を作成する

2-2. SQL とその実行計画を STS（SQL チューニングセット）へ格納する

2-3. STS（SQL チューニングセット）へ格納された SQL の件数を確認する

2-4. STS（SQL チューニングセット）へ格納された SQL と PLAN_HASH_VALUE を確認する

2-5. STS（SQL チューニングセット）へ格納された SQL 実行計画を確認する

3．STS（SQL チューニングセット）から SQL 計画ベースライン（承認済実行計画）を作成する

3-1. STS（SQL チューニングセット）から SQL 計画ベースライン（承認済実行計画）を登録する

3-2. SQL 計画ベースライン（承認済実行計画）として登録されたことを確認する

3-3. SQL 計画ベースライン（承認済実行計画）として登録された SQL 実行計画を表示する

3-4. STS（SQL チューニングセット）を削除する

【詳細手順 実操作】

対象とする SQL 文の選定は、現場からの遅延報告に基づいて、遅くなった処理（トランザクション）を構成する SQL 文の処理時間を調査して決定する

1．SQL 文の性能が良かった過去の実行計画が AWR に存在することを確認する

1-1. AWR 自動ワークリポジトリから該当の SQL_ID を取得する

調査したい SQL ステートメントを Where 条件に設定して、dba_hist_sqltext ディクショナリを検索し、SQL_ID を求める

ドキュメントのB）→dba_hist_sqltext ディクショナリ部分を参照

```
select  SQL_ID , SQL_TEXT  from  dba_hist_sqltext
    where  SQL_TEXT  like  'select ・・ from  ・・・%' ;
```

1-2. AWR 自動ワークリポジトリから該当の SQL の過去の PLAN_HASH_VALUE とスナップショットの ID を取得する

1-1. で求めた SQL_ID を Where 条件に設定して、dba_hist_sqlstat ディクショナリ、dba_hist_snapshot ディクショナリを検索し、AWR スナップショットの SNAP_ID を求める

ドキュメントのC）→dba_hist_sqlstat ディクショナリ、dba_hist_snapshot ディクショナリ部分を参照

```
select  sql.SNAP_ID ,
        to_char( snap.BEGIN_INTERVAL_TIME, 'yyyy/mm/dd hh24:mi:ss' ),
        sql.SQL_ID ,  sql.PLAN_HASH_VALUE
  from  dba_hist_sqlstat  sql ,  dba_hist_snapshot  snap
  where  sql.dbid  =  snap.dbid
    and  sql.instance_number  =  snap.instance_number
    and  sql.snap_id  =  snap.snap_id
    and  SQL_ID  =  '＜SQL_ID 値＞'
  order by  sql.snap_id  ;
```

※ 1-2.で PLAN_HASH_VALUE が**複数存在していたら**、そのすべての実行計画の **AWR スナップショットの SNAP_ID** を抜き出し、「SQL 管理ベース」を登録（保存）する

そして、１つずつ対象の「SQL 管理ベース」の ACCEPT フラグを変更して、実行計画を実行する

実行した後に、その実行統計情報を比較して効率が良い方を判断する

1-3. AWR 自動ワークリポジトリから該当の SQL の実行計画を確認する

1-1. で求めた AWR スナップショットの SNAP_ID を Where 条件に設定して、DBMS_XPLAN.DISPLAY_AWR ファンクションを実行し、実行計画を表示する

ドキュメントの F）→DBMS_XPLAN.DISPLAY_AWR ファンクション部分を参照

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

2．過去の実行計画を AWR から STS（SQL チューニングセット）を作成する

SQL チューニング・セット（STS）をコマンドで作成する方法

2-1. STS（SQL チューニングセット）を作成する

```
exec  DBMS_SQLTUNE.CREATE_SQLSET( 'SQL チューニングセット名' ) ;
```

この名前で、新規作成される

2-2. SQL 文とその実行計画を STS（SQL チューニングセット）へ格納する

1-2. で求めた AWR スナップショットの SNAP_ID と 1-3.で求めた PLAN_HASH_VALUE とスナップショットの ID をパラメータに指定して、STS（SQL チューニングセット）へ SQL 文と実行計画を格納する

```
declare
    cur  DBMS_SQLTUNE.SQLSET_CURSOR  ;
begin
    open  cur  for
    select  VALUE( p )  from  table(
      DBMS_SQLTUNE.SELECT_WORKLOAD_REPOSITORY(
        begin_snap => ＜SNAP_ID  - 1 の値＞,
        end_snap => ＜SNAP_ID の値＞,
        basic_filter => 'sql_id = "＜SQL_ID 値＞"  and  plan_hash_value
                          =  ＜PLAN_HASH_VALUE 値＞'  ) )  p ;
    DBMS_SQLTUNE.LOAD_SQLSET(
                          'SQL チューニングセット名' , cur) ;
  end;
  /
```

2-1.で、新規作成された名前

2-3. STS（SQL チューニングセット）へ格納された SQL の件数を確認する

```
select  name , owner , created , statement_count
  from  dba_sqlset
  where  name  =  'SQL チューニングセット名' ;
```

NAME	OWNER	CREATED	STATEMENT_COUNT
STS_SPM01	SYS	2014-08-02 16:12:00	1

2-4. STS（SQL チューニングセット）へ格納された SQL と PLAN_HASH_VALUE を確認する

1-1.で作成した SQL チューニングセット名を Where 条件に指定して、dba_sqlset_statement ディクショナリを検索し、登録されている SQL_ID、PLAN_HASH_VALUE、PARSING_SCHEMA_NAME、SQL_TEST を表示する

ドキュメントのD）→dba_sqlset_statement ディクショナリ部分を参照

```
select  sql_id , plan_hash_value , parsing_schema_name ,
        substr( sql_text , 1 , 100 )  sql_text
  from  dba_sqlset_statements
  where  sqlset_name  =  '＜SQL チューニングセット名＞'
  order by  sql_id  ;
```

2-5. STS（SQL チューニングセット）へ格納された SQL 実行計画を確認する

2-1.で作成した SQL チューニングセット名と 1-1.で求めた SQL_ID をパラメータに指定して、DBMS_XPLAN.DISPLAY_SQLSET ファンクションを実行し、STS（SQL チューニングセット）へ格納された SQL 実行計画を表示する

ドキュメントのG）→DBMS_XPLAN.DISPLAY_SQLSET ファンクション部分を参照

```
select  *  from  table( dbms_xplan.display_sqlset( '＜SQL チューニングセット名＞', '＜SQL_ID 値＞' )) ;
```

３．STS（SQL チューニングセット）から SQL 計画ベースライン（承認済実行計画）を作成する

３-1. STS（SQL チューニングセット）から SQL 計画ベースライン（承認済実行計画）を登録する

２-1.で作成し、２-2 で実行計画を格納した STS（SQL チューニングセット）を使って、dbms_spm.load_plans_form_sqlset ファンクションを実行して、実行計画を SQL 計画ベースライン（承認済実行計画）として登録する

```
declare
   ret   number ;
begin
   ret   :=   dbms_spm.load_plans_from_sqlset(
                    sqlset_name   =>   '＜SQL チューニングセット名＞' ,
                    sqlset_owner   =>   '＜STS の所有スキーマ＞' ,
                    enabled   =>   'YES' ）  ;
   dbms_output.put_line( ' LOAD PLANS : ' || ret ）  ;
end;
/
```

２-1.で、作成した名前

実行結果

LOAD PLANS： 1　　※ 登録した SQL 計画ベースラインの数が戻り値として表示されます

３-2. SQL 計画ベースライン（承認済実行計画）として登録されたことを確認する

１-1. で調査対象とした SQL ステートメントを Where 条件に設定して、dba_sql_plan_baseline ディクショナリを検索して、SQL_HANDLE と実行計画が 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 ディクショナリ部分を参照

```
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   ;
```

```
SQL_HANDLE                     PLAN_NAME
------------------------------ ------------------------------------------------
SQL_e18f9b7e02f01395           SQL_PLAN_f33wvgs1g04tu8447c07a

                    CREATED
                    ---------------------------
                    2016/11/02 16:20:45
```

3-3. SQL 計画ベースライン（承認済実行計画）として登録された SQL 実行計画を表示する

3-2.で調査した SQL_HANDLE を使って、DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE ファンクションを実行して、登録された SQL 実行計画の内容を確認する

ドキュメントの K-1）→DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE ファンクション部分参照

```
select  *  from  table( dbms_xplan.display_sql_plan_baseline(
                                        '＜SQL_HANDLE 値＞'));
```

3-4. STS（SQL チューニングセット）を削除する

2-1.で作成した STS（SQL チューニングセット）を削除する

```
exec  DBMS_SQLTUNE.DROP_SQLSET(' SQL チューニングセット名');
```

2-1.で、作成した名前