

SQL*PLUS の Select 結果を基に、SQL スクリプトを作成する手順

作成方法

手順 1.

サイレント・モードでデータベースに接続する

```
sqlplus -s /nolog
CONNECT ユーザー名/パスワード@接続識別子
```

手順 2.

Sql*Plus の出力環境を整備する

```
-- 改ページを行わない
SET NEWP NONE

-- 1 ページの最大行数は、無限
SET PAGES 0

-- 出力に対する列名は、表示させない
SET HEAD OFF

-- 実行コマンドに対しての結果フィードバックは行わない
-- ON の場合には、Select 実行に対して、「1 行が選択されました。」と出力さ
   れてしまう
SET FEEDBACK OFF

-- 実行結果は、スプールへ出力する
SET TRIMSPOOL ON

-- 作成する SQL スクリプト・ファイルのディレクトリ・パスと名前を指定す
   る
SPOOL <ディレクトリ・パス>¥<出力結果の保存ファイル名>.sql
```

手順 3.

調査対象となる SELECT 文を作成し、実行する

この時に、出力列の欄にスクリプトで実行させる文字列となるように編集する

※ 「'」内の文字列」と「検索結果」を||を使用して結合し SQL 文を生成

```
SELECT スクリプトで実行させる文字列 FROM . . . . . ;
```

手順 4.

スプール出力を閉じる

```
SPOOL OFF
```

手順 5.

スクリプトを実行させたいユーザーにて、データベースに接続し直す

```
disconnect
```

```
sqlplus ==s /nolog
```

```
CONNECT ユーザー名/パスワード@接続識別子
```

手順 6.

作成された SQL スクリプト・ファイルの内容を確認する

```
host type <ディレクトリ・パス>¥<出力結果の保存ファイル名>.sql
```

手順 7.

作成した SQL スクリプトを実行する

```
@<ディレクトリ・パス>¥<出力結果の保存ファイル名>.sql
```

例 1)

特定ユーザーのセッションを切断する SQL の生成し、実行する
そのために、GV\$SESSION 動的ビューを select して、
その結果から

```
ALTER SYSTEM KILL SESSION '<sid 値>, <serial#値>,'  
@<インスタンス ID>' ;
```

文字列をスクリプト・ファイルに作成する

-- 手順 1.

-- サイレント・モードでデータベースに接続する

```
sqlplus -s /nolog
```

```
CONNECT ユーザー名/パスワード@接続識別子
```

-- 手順 2.

-- Sql*Plus の出力環境を整備する

```
SET NEWP NONE
```

```
SET PAGES 0
```

```
SET HEAD OFF
```

```
SET FEEDBACK OFF
```

```
SET TRIMSPOOL ON
```

```
SPOOL D:¥temp¥test_script.sql
```

-- 手順 3.

-- 調査対象となる SELECT 文を作成し、実行する

```
select 'ALTER SYSTEM KILL SESSION "' || sid || ','
```

```
|| serial# || ',@' || INST_ID || "' ;'
```

```
FROM GV$SESSION
```

```
WHERE USERNAME = 'KOZUE';
```

-- 手順 4.

-- スクリプトを実行させたいユーザーにて、データベースに接続し直す

```
SPOOL OFF
```

- 手順 5.
 - スクリプトを実行させたいユーザーにて、データベースに接続し直す
- ```
disconnect

exit

sqlplus ==@ /nolog
CONNECT ユーザー名/パスワード@接続識別子
```
- 手順 6.
    - 作成された SQL スクリプト・ファイルの内容を確認する
- ```
host type D:¥temp¥test_script.sql
```
- 手順 7.
 - 作成した SQL スクリプトを実行する
- ```
@D:¥temp¥test_script.sql
```

作成されたスクリプト

```
ALTER SYSTEM KILL SESSION '1, 33067, @1' ;
ALTER SYSTEM KILL SESSION '2, 9737, @1' ;
ALTER SYSTEM KILL SESSION '3, 5431, @1' ;
```

## 例 2)

作成したいスクリプト

統計情報が取得されていない表の統計情報取得用 SQL の生成し、実行する  
(この例では TEST ユーザーを対象としています)



DBA\_TABLES デictionary を select して、  
その結果から

```
EXEC DBMS_STATS.GATHER_TABLE_STATS('<ユーザー名>',
 '<テーブル名>');
```

文字列をスクリプト・ファイルに作成する

作成手順

```
rem 手順 1.
rem サイレント・モードでデータベースに接続する
sqlplus -s /nolog
CONNECT ユーザー名/パスワード@接続識別子

-- 手順 2.
-- Sql*Plus の出力環境を整備する

SET NEWP NONE
SET PAGES 0
SET HEAD OFF
SET FEEDBACK OFF
SET TRIMSPOOL ON

SPOOL D:¥temp¥test_script.sql

-- 手順 3.
-- 調査対象となる SELECT 文を作成し、実行する

SELECT 'EXEC DBMS_STATS.GATHER_TABLE_STATS("SYSTEM",
 " || TABLE_NAME || ");'
FROM DBA_TABLES
WHERE OWNER = 'SYSTEM'
 AND LAST_ANALYZED IS NULL ;

-- 手順 4.
-- スクリプトを実行させたいユーザーにて、データベースに接続し直す

SPOOL OFF
```

- 手順 5.
  - スクリプトを実行させたいユーザーにて、データベースに接続し直す

```
disconnect
```

```
exit
```

```
sqlplus ==> /nolog
```

```
CONNECT ユーザー名/パスワード@接続識別子
```

- 手順 6.
  - 作成された SQL スクリプト・ファイルの内容を確認する

```
host type D:¥temp¥test_script.sql
```

- 手順 7.
  - 作成した SQL スクリプトを実行する

```
SET PAGES 200
```

```
SET FEEDBACK OFF
```

```
SET TRIMSPOOL ON
```

```
SPOOL <出力ログファイル名>.log ← この後の SQL 処理を実行した時
 に出力されるメッセージの出力
 先ログの指定
```

```
@D:¥temp¥test_script.sql
```

```
SPOOL OFF
```

作成されたスクリプト

```
EXEC DBMS_STATS.GATHER_TABLE_STATS('SYSTEM', 'OL$');
EXEC DBMS_STATS.GATHER_TABLE_STATS('SYSTEM', 'OL$HINTS');
.....
EXEC DBMS_STATS.GATHER_TABLE_STATS('SYSTEM', 'TEST1');
```