

SQL*PLUSの使い方

SQL*PLUS をサイレント・モードで起動する

SQL コマンドを実行しても、画面表示されない

SELECT コマンドの実行結果は、スプールの中だけに出力される

```
$ sqlplus -s / as sysdba
```

Help 機能の表示

コマンド一覧は、「help index」

個別の操作コマンドに対するパラメータ表示は、上記の help index で表示された操作コマンドに対して「help 操作コマンド」

SQL/Plus 内で使用されている各種パラメータの設定値の表示に対しては、「show all」

Oracle の初期化パラメータの設定値表示は、「show parameters」

現在の SQL/Plus に対する設定内容確認 および、特定パラメータの設定内容確認

例 sql> show all

sql> show <パラメータ名>

実行した直前コマンドのメモ帳での編集

「edit」コマンドで、直前コマンドのメモ帳で開きます

内容を変更して、メモ帳を保存、終了します ※ 保存されるファイルは、afiedt.buf

SQL*Plus 画面で、「/」を実行すると編集した内容のコマンドが実行されます

SQL*Plus からスクリプトの実行

@スクリプトファイル名

例 SQL> @D:¥Script-dir¥SelectNo.sql

直前コマンドの再実行

/ (スラッシュ)

もしくは、「run」 ※ 実行結果に実行コマンドも表示されます

↑ キーで、過去に実行したコマンドを遡れます

SQL 実行結果のプール出力時の注意点

例)

set pages 0	←	1 ページの最大行数は、無限
set head off	←	出力に対する列名は、表示させない
set newp none	←	1 行目の改行削除
set feedback off	←	コマンドの確認メッセージ非表示
set trimspool on	←	行末の不要空白の削除
set colsep ,	←	列出力の区切り文字を『,』に変更

spool <ディレクトリ・パス>¥<結果の出力先ファイル名>.txt
← 実行結果が出力される先のファイル名を指定

1 ページ文のデータ表示後の画面ストップ制御

例 sql> set pause on

列の表示形式の設定

例 sql> col <列名> for <表示書式>
col object_name **format** a10

<表示書式>

a10 : 文字 10 個

s99999.99 : 符号 整数部 5 桁 小数点以下 2 桁

SQL 実行結果のプール出力

```
sql> spool <ファイル名>
sql> SQL コマンド;
sql> spool off
```

結果出力の折返しカラム数の変更

確認 「show linesize」

変更 「set linesize カラム数」

SQL*Plus から OS コマンドを直接実行

「host OS コマンド」で、OS コマンドが実行されます

「! OS コマンド」でも、OS コマンドが実行されます

例 sql> cat error.log

XXXXXX I/O error 発生者：～～～

XXXXXX USER error 発生者：～～～

sql> ! rm error.log

入力画面への時刻表示

コマンド入力画面のプロンプト前に時刻を表示させる

例 SQL> set time on

SQL 文の実行時間の測定方法

実行する SQL コマンドの直前に「set timing on」を実行する

例 SQL> set timing on

SQL> Select * From torihikisaki_table ;

「time on」と「set timing on」の違い

SQL 文の実行時間の測定方法

例 sql> set time on

← 時間表示

19:14:01 sql> set timing on

← コマンド時間測定 ON

19:14:03 sql> Select * From torihikisaki_table ;

経過 : 00 : 00 : 05

← 経過時間

19:14:15 sql> set timing off

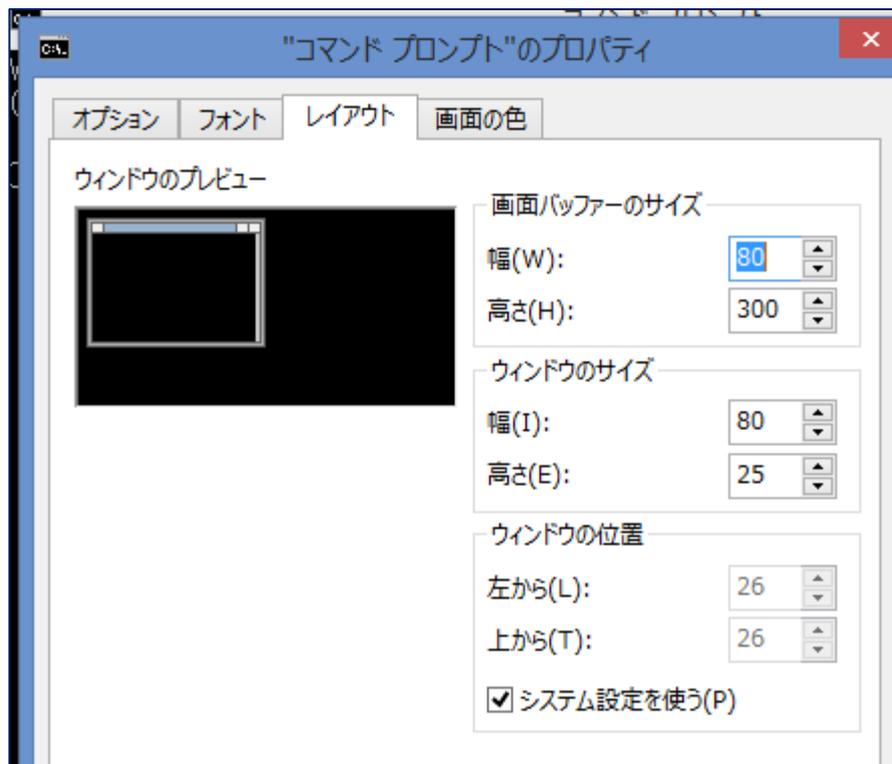
← コマンド時間測定 OFF

19:14:18 sql> set time off

← 時間表示解除

コマンドプロンプトの折返しカラム数変更

コマンドプロンプト画面の右上のアイコンをクリック → プロパティ
[レイアウト] タブのウィンドウサイズの幅を変更



—— デバッグ方法 ——

PL/SQL プログラム中の変数の値を SQL*PLUS 画面へ出力する方法

(DBMS_OUTPUT.PUT_LINE(TO_CHAR(変数)) の出力の有効化)

「SET SERVEROUT[PUT] ON」を実行する

```
例 SQL> SET SERVEROUT[PUT] ON
SQL> DECLARE
      a NUMBER ;
BEGIN
      SELECT COUNT(*) INTO a FROM emp_table ;
      DBMS_OUTPUT.PUT_LINE(TO_CHAR(a));
      /* a の値が SQL*PLUS 画面に出力される */
END ;
/
```

SQL 文 (INSERT 文など) で、『**文字列や日付の値**』を指定して使うときには、「'」 (シングルクォーテーション) で囲む

日付形式データの書き方

' (シングルクォーテーション) で囲みます

例 'YY-MM-DD HH:MI:SS'

Oracle バックアップ・リカバリ実践テクニック

```
SET UNTIL TIME =
```

```
"to_date( '2007-01-01:00:00:00' , 'YYYY-MM-DD:HH24:MI:SS' );"
```

日付形式のデータの表示フォーマットの指定

デフォルトの日付表示フォーマットを宣言する

```
alter session set NLS_DATE_FORMAT = 'yyyy/mm/dd hh24:mi:ss';
select sysdate from dual;
```

```
SYSDATE
```

```
-----
```

```
2018/06/28 09:22:28
```

文字列の使用

Insert 文や Where 条件のデータに文字列を含む場合には、「'」 (シングルクォーテーション) で囲みます

例 '愛川 こずえ'

```
select * from scott.emp where ename = 'JAMES';
```

```
EMPNO  ENAME  JOB
-----  -
7900  JAMES  CLERK
```

```
SQL> execute DBMS_OUTPUT.PUT_LINE('データ使用サイズ' );
```

```
データ使用サイズ
```

```
PL/SQL プロシージャが正常に完了しました。
```

Where 句での NULL 条件の指定方法

Where 列名 is NULL

Where 列名 is not NULL

Select 文結果での Null 値の出力順序指定

※ Oracle では、Null は最大値となる

開始行表示： order by 列名 NULLS FIRST

最終行表示： order desc by 列名 NULLS LAST

テーブル名や列名の指定

空白の文字を含む列名には、「"」（ダブルクォーテーション）で囲みます

例 "USER NAME"

テーブル名や列名の名前付けに小文字が含まれる場合に使用します

「'」シングルクォーテーションで区切った場合には、すべて大文字変換した名前で登録されてしまいます

例 Create Table "Kozue"."Table1" ("USER NAME" VARCHAR(2),

SQL 文の終端文字

SQL 文の終端は、; (セミコロン) を使用します

Select 文の結果への列名の表示サイクル

確認 「show pagesize」

変更 「set pagesize カラム数」

列名 1 列名 2 列名 3 列名 4

列の区切り文字の設定

変更 「set colsep 区切り文字」

列名 1	列名 2	列名 3	列名 4

大量の Select 結果を途中で止まらせる場合

「set pause on」

【 注意事項 】

—— SQL/Plus で行われたコマンドが使用するリソース ——

- SQL/Plus が使用するメモリ

Oracle サーバーの Oracle インスタンス内のサーバー・プロセス

- SQL/Plus で指示した SQL コマンド

Oracle サーバーの Oracle インスタンスの内部で実行される

- host コマンドの内容が実行される対象コンピュータ

cmd.exe プログラムが実行されているローカル・コンピュータ

- スプール出力が行われる対象のストレージ

cmd.exe プログラムが実行されているローカル・コンピュータのディスク

- SQL コマンドがディスクに対して行う操作

Oracle インスタンスが起動しているサーバー

例) コントロール・ファイルのバイナリ形式バックアップ

```
ALTER DATABASE BACKUP CONTROLFILE  
TO 'ファイル名' REUSE;
```

バインド変数を使用した SQL 文の実行方法

```
例 sql> var バインド変数 データ型          ← バインド変数定義
          サンプル varchar2(30)

sql> exec :バインド変数 := '値'          ← バインド変数への値セット
sql> SQL コマンド
sql> select * from user_table where user_name = :バインド変数 ;
```