

SQL*Plus と PL/SQL の関係

SQL*Plus は、クライアント側の処理として動作する

PL/SQL は、Oracle インスタンスの中のプロセス（サーバー・プロセスの 1 つの中で）として動作する。（無名 PL/SQL ブロックを含め）

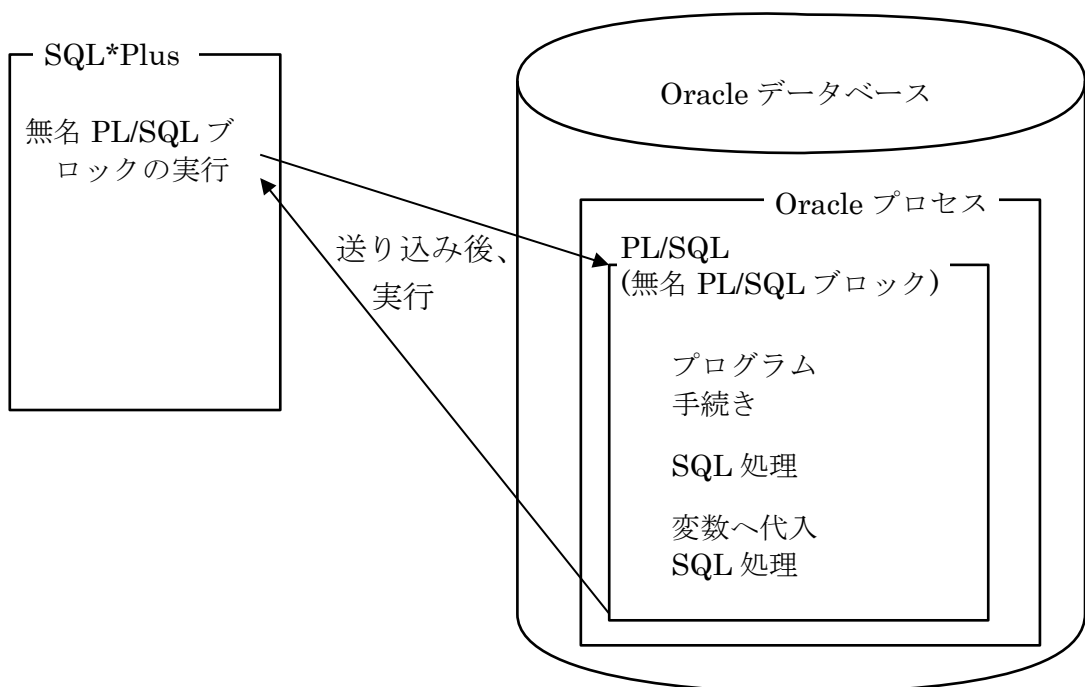
よって、SQL*Plus の中で PL/SQL の実行を行うということは、クライアントと Oracle システムのプロセスとの通信（ネットワーク通信、プロセス間通信）が行われており、オーバーヘッドが発生する

PL/SQL の種類

- ・ストアードプロシージャ
- ・ストアードアクション
- ・無名 PL/SQL ブロック（SQL*Plus での直接記述使用）

ストアードプロシージャ、ストアードアクションの使い方としては、SQL*Plus の中から Oracle 関数を呼出す（コールする）イメージで実行が行われている

また、無名 PL/SQL ブロックについては、SQL*Plus からデータベース側のサーバープロセスへ、**記述内容のすべてを送り込んだ後に**、Oracle プロセスの中で実行が行われている



変数 (PL/SQL)、変数 (SQL*Plus)、ホスト変数、バインド変数の違い

変数 (PL/SQL) とは、

変数(PL/SQL)とは、**無名 PL/SQL ブロック**やPL/SQLプロシージャやPL/SQLファンクションの中で定義されている変数を指す

この変数のメモリは、データベース・サーバーの Oracle インスタンス (サーバー・プロセス) の中に確保されている

実行が終わった後では、SQL*Plus からその値を使用することは出来ない

無名 PL/SQL ブロック

```
sql> DECLARE
sql>     a NUMBER ;
sql> BEGIN
sql>     .
sql>     .
sql> END ;
```

変数 (SQL*Plus) とは、

||

ホスト変数とは、

ホスト変数とは、SQL*Plus の処理記述において 無名 PL/SQL ブロックの外で定義されている変数を指す

この変数のメモリは、クライアント端末の中で SQL*Plus の変数としてメモリ確保される

なお、PL/SQL プロシージャや PL/SQL ファンクションの中からは、直接にバインド変数を使用することは出来ない

PL/SQL プロシージャや PL/SQL ファンクションの呼び出し時に、実引数として使用することだけが可能となるだけである

```
sql> VARIABLE a NUMBER ;

sql> DECLARE
sql> BEGIN
sql> END ;
```

バインド変数とは、

バインド変数とは、ホスト変数として定義された変数を、無名 PL/SQL ブロックの中へ取込んで使用する場合に、この変数のことをバインド変数と呼ぶ

バインド変数は、ホスト変数の変数名にコロン「:」を付けたものに自動的になる

なお、バインド変数は、PL/SQL プロシージャや PL/SQL ファンクションの中では使用出来ない

ホスト変数の値を使用したい場合には、呼び出し時にバインド変数をプロシージャ等の実引数パラメータとして受け渡す方法しかない

~~PL/SQL プロシージャや PL/SQL ファンクションの中でバインド変数を使用される場合、処理が実行される時に値が確定されるので、SQL 文の中にバインド変数が含まれている場合には、動的 SQL の実行のためのコーディング (EXECUTE IMMEDIATE 構文) が必要となる~~

ホスト変数の定義とバインド変数の使用例

```
/* ホスト変数の定義 */
sql> VARIABLE i_name VARCHAR2(20);

/* ホスト変数をバインド変数として使用し、値をセット */
sql> EXECUTE :i_name := 'SCOTT';
/* 「:」コロンの記述を忘れないように */
```

バインド変数を無名 PL/SQL ブロックで使用方法

```
/* ホスト変数の定義 */
sql> VARIABLE i_name VARCHAR2(20);

/* ホスト変数をバインド変数として使用し、値をセット */
sql> EXECUTE :i_name := 'SCOTT';

/* 無名 PL/SQL ブロック */
sql> DECLARE

/* PL/SQL の変数の定義 */
sql> VARIABLE user_name VARCHAR2(20);

sql> BEGIN

/* バインド変数の使用 */
sql> user_name := :i_name ;
/* 「:」コロンの記述を忘れないように */

sql> END;
```

バインド変数を PL/SQL プロシージャ等の呼び出し時に、実引数パラメータとして使用方法

```
/* ホスト変数の定義 */
sql> VARIABLE o_name VARCHAR2(20);

/* ホスト変数をバインド変数として使用し、値をセット */
sql> EXECUTE :o_name := 'SCOTT'
/* 「:」コロンの記述を忘れないように */
/* プロシージャの呼出し バインド変数での引数指定 */
sql> EXECUTE proc_A(1, :o_name) ;
```

バインド変数の使用と動的 SQL の実行テスト

無名 PL/SQL ブロックの中でのバインド変数の使用

結論)

無名 PL/SQL ブロックの中では、通常にバインド変数が見える

```
SET SERVEROUTPUT ON;
```

```
VARIABLE input_string VARCHAR2(300);
```

```
execute :input_string := 'START TEST';
```

```
DECLARE
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('SQL 文 = ' || :input_string);
```

```
:input_string := 'END TEST';
```

```
END;
```

```
/
```

```
print input_string
```

PL/SQL プロシージャの中でのバインド変数の使用

結論)

PL/SQL プロシージャの中では、バインド変数は使用出来ない

```
SET SERVEROUTPUT ON;
```

```
VARIABLE input_string VARCHAR2(300);
```

```
execute :input_string := 'AAAA';
```

```
CREATE or REPLACE PROCEDURE TEST_procedure1
```

```
is
```

```
    SQL_string          VARCHAR2(300);
```

```
BEGIN
```

```
    SQL_string := '-- START TEST --';
```

```
    DBMS_OUTPUT.PUT_LINE('SQL_string= ' || SQL_string);
```

```
-- DBMS_OUTPUT.PUT_LINE('input_string= ' || :input_string);
```

```
-- × PLS-00049: バインド変数 INPUT_STRING が正しくありません。
```

```
-- :input_string := 'DDDD';
```

```
-- × PLS-00049: バインド変数 INPUT_STRING が正しくありません。
```

```
END;
```

```
/
```

```
show errors
```

```
EXECUTE TEST_procedure1();
```

```
print input_string
```

無名 PL/SQL ブロックの中でのバインド変数を使用した動的 SQL 文の作成と実行

結論)

動的 SQL 文以外の処では、通常にバインド変数は使用出来る

ただし、動的 SQL 文中に使われるバインド変数は、SQL*Plus で定義したホスト変数とは異なるものである

PL/SQL ブロックの動的 SQL 文中の中だけで有効となる特殊な変数となる

動的 SQL 文中に使うバインド変数は、データ型の定義をする必要はない
また、この動的 SQL 文中に使うバインド変数に値をセットする方法は、EXECUTE IMMEDIATE の USING 句で指定する方法に限られる

```
SET SERVEROUTPUT ON;
```

```
VARIABLE input_string VARCHAR2(300);
```

```
VARIABLE output_string VARCHAR2(300);
```

```
execute :input_string := 'AAAA';
```

```
execute :output_string := 'BBBB';
```

```
DECLARE
```

```
    response_string VARCHAR2(300);
```

```
    SQL_string      VARCHAR2(300);
```

```
BEGIN
```

```
-- ○ SQL_string := 'Select "CCCC" From dual Where 5 = 5';
```

```
-- × SQL_string := 'Select "CCCC" From dual Where :input_string  
                = :output_string';
```

```
--          ORA-01008: バインドされていない変数があります。
```

```
--          ORA-06512: 行 14
```

```
--          原因 : バインド変数:output_string に値がセットできていない
```

```
SQL_string := 'Select "CCCC"      From dual Where :JyoukenA  
              = :JyoukenB';
```

```
DBMS_OUTPUT.PUT_LINE('SQL 文 = ' || SQL_string);
```

```
EXECUTE IMMEDIATE SQL_string INTO :output_string
```

```
        USING 'AAAA', 'AAAA';

END;

/

print output_string
```

PL/SQL プロシージャの中でのバインド変数を使用した動的 SQL 文の作成と実行

結論)

PL/SQL プロシージャの中では、バインド変数は使用出来ない

また、PL/SQL プロシージャの動的 SQL 文中に使われるバインド変数は、前記の無名 PL/SQL ブロックの中のバインド変数と同様で、SQL*Plus で定義したホスト変数とは異なるものである

PL/SQL ブロックの動的 SQL 文中の中だけで有効となる特殊な変数となる

動的に SQL 文の条件を変更させるための方法は、プロシージャの呼び出し時の引き数にパラメータをセットして受け渡す

PL/SQL プロシージャの中で、動的 SQL 文の実行時の EXECUTE IMMEDIATE の USING 句でプロシージャの仮引数の値を、動的 SQL 文中に使われているバインド変数に受け渡すようにして実行を行う

```
EXECUTE proc_A(1, :o_name) ;
```