

## テーブルのレコードに対するデータ型と変数定義 Select . . . into 変数 による変数への値代入 (PL/SQL)

テーブル (表) の列と同じデータ型に合わせて、自動的に変数の型を定義する  
(%TYPE)

```
変数名 スキーマ名. テーブル名. 列名%TYPE ;  
(例)  
user_name user1.member_table.user_name%TYPE ;
```

テーブル (表) のレコードと同じデータ型に合わせて、自動的に変数の型を定義する  
(%ROWTYPE)

```
変数名 スキーマ名. テーブル名%ROWTYPE ;  
(例)  
DECLARE  
    user_record emp_table%ROWTYPE ;  
BEGIN  
    SELECT * INTO user_record FROM emp_table WHERE empno = 4 ;  
    DBMS_OUTPUT.PUT_LINE( user_record. ename ) ;  
END;  
/
```

```
DECLARE  
    TYPE user_record_type IS TABLE OF emp_table%ROWTYPE  
        INDEX BY BINARY_INTEGER;  
    -- TYPE 文で指定した構造体データ型を使った配列データ型の定義  
    user_record user_record_type ;  
    -- TYPE 文で指定した配列データ型を使った変数の定義  
BEGIN  
    SELECT * BULK COLLECT INTO user_record FROM emp_table  
        order by empno ;  
    DBMS_OUTPUT.PUT_LINE( user_record(1). ename ) ;  
    DBMS_OUTPUT.PUT_LINE( user_record(2). ename ) ;  
    DBMS_OUTPUT.PUT_LINE( user_record(3). ename ) ;  
    DBMS_OUTPUT.PUT_LINE( user_record(4). ename ) ;  
END;  
/
```

SQL\*PLUS からの SELECT 文の直接の呼出し方法  
Select . . . into 変数 による変数への値代入 (SQL\*PLUS)

ホスト変数とバインド変数の使い方

SQL\*PLUS のコマンドから、SELECT 文を直接実行する

Select into 文は、条件該当が 1 レコードの時のみしか使用できない

※ パッケージの呼出しには、パッケージ名を修飾してファンクション名を指定する  
戻り値変数 = パッケージ名. ファンクション名 (引数) ;

```
          /* ホスト変数の定義 */
SQL> VARIABLE a NUMBER ;
SQL> VARIABLE b VARCHAR2(50) ;
SQL> VARIABLE c VARCHAR2(50) ;

          /* ファンクションの呼出しと戻り値のセット */
          /* 引数は、バインド変数にして引渡す */
SQL> BEGIN
/* 複数列の場合は、カンマを使って列名を並べる */
SELECT empno,  ename,  other INTO :a, :b, :c
      FROM emp WHERE empno = 4 ;
END ;      -- 無名 PL/SQL ブロック
/
          /* //は、SELECT 文の実行のため */

SQL> print a == b == c
          /* SQL*Plus でのホスト変数値の表示 */
```

※ この方法は、PL/SQL 無名プロシージャの BEGIN 節の中でも記述  
できる