

【配列の定義と使い方】 カーソルを使用した複数レコード処理

配列の種類

- ・ 結合配列
- ・ ネストした表
- ・ VARRAY
- ・ レコード型配列
- ・ レコード型配列の 2 次元配列

※ 格納要素上限個数 : 2,147,483,647 個

配列の定義方法

- ステップ 1. 配列用のデータ型の名前宣言
- ステップ 2. 実際の配列の変数定義
- ステップ 3. 配列の初期化

【注意】

ネストした表や VARRAY 型の配列に値をセットする場合は、事前に初期化していなければならない (P. 75)

ステップ 1. 配列用のデータ型の名前宣言

- ・ 結合配列のデータ型の宣言

```
TYPE 配列用のデータ型の名前 IS TABLE OF データ型  
INDEX BY 添字のデータ型 ;
```

- ~~・ ネストした表のデータ型の宣言 (INDEX なし)~~

```
TYPE 配列用のデータ型の名前 IS TABLE OF データ型 ;  
(添字は、整数で自動付番)
```

- ・ ネストした表のデータ型の宣言

```
TYPE 配列用のデータ型の名前 IS TABLE OF データ型  
INDEX BY BINARY_INTEGER ;
```

- ~~・ VARRAY のデータ型の宣言~~

```
TYPE 配列用のデータ型の名前 IS VARRAY (要素の上限) OF データ型 ;  
(添字は、整数で自動付番)
```

- ・レコード型配列の宣言 (P.86)

```
TYPE 配列用のデータ型の名前 IS RECORD (  
  列1の名前 列1のデータ型 ,  
  列2の名前 列2のデータ型 ,  
  列3の名前 列3のデータ型 );  
      (添字は、なし)
```

ステップ2. 実際の配列の変数定義

- ・実際の配列の変数定義 (結合配列、ネストした表、VARRYとも共通)

```
配列名 配列用のデータ型の名前 ;  
配列名 テーブルの名前%ROWTYPE ; (P.87) レコード型配列の宣言は不要
```

【注意】

EXECUTE IMMEDIATE SQL文 BULK COLLECT INTO を使用して、複数レコードを一括処理で読み込む場合には、ステップ1. のレコード型配列の宣言が必要

3 ページ後に説明記入

```
/* カーソルを使ったテーブル型のデータ型定義 */
```

```
TYPE ユーザー配列定義型 IS TABLE OF テーブル名%ROWTYPE  
INDEX BY PLS_INTEGER ;
```

```
/* 配列の変数定義 */
```

```
配列用変数名 ユーザー配列定義型 ;
```

```
BEGIN
```

```
-- SQL文の実行
```

```
EXECUTE IMMEDIATE SQL文 BULK COLLECT INTO  
      配列用変数名 ;
```

```
-- データの使用方法
```

```
配列用変数名 (添字). テーブル列名
```

ステップ 3. 配列の初期化

【注意】

ネストした表や VARRAY 型の配列に値をセットする場合は、事前に初期化して
いなければならない (P. 75)

- 実際の配列の変数定義 (結合配列、ネストした表、VARRY とも共通)

変数の名前 (定義用) 配列用のデータ型の名前 := 配列用のデータ型の名
前(") ; -- 数値型の場合は、0 で初期化のこと

(使用例)

ステップ 1. 配列用のデータ型の名前宣言

```
TYPE sample_type IS number INDEX BY VARCHARA2( 40 );
TYPE sample_type IS VARCHARA2( 20 ) INDEX BY PLS_INTEGER ;
TYPE sample_type IS number INDEX BY VARCHARA2( 40 );
```

P. 86

```
TYPE sample_type IS RECORD (
    empno emp.empno%TYPE
    ename emp.ename%TYPE
    job emp.job%TYPE ) ;
```

~~/* コーソルを使ったレコード型の配列データ型定義 */~~

```
TYPE typ_emp_tab IS TABLE OF emp_table%ROWTYPE
    INDEX BY PLS_INTEGER ;
```

ステップ 2. 実際の配列の変数定義

```
dim_sample sample_type ;
```

-- この場合には、レコード型配列宣言の記述は無し (1レコード分の領域のみ)

```
dim_sample sample_table%ROWTYPE ;
```

-- 複数レコードに対応したレコード型配列宣言について

【注意】

EXECUTE IMMEDIATE SQL 文 **BULK COLLECT INTO** を使用して、テ
ーブルを読み込む場合には、ステップ 1. のレコード型配列の宣言が必要

~~/* コーソルを使ったテーブル型の配列データ型定義 */~~

```
TYPE typ_emp_tab IS TABLE OF emp_table%ROWTYPE
    INDEX BY PLS_INTEGER ;
```

~~/* 配列の変数定義 */~~

```
dim_emp_record typ_emp_tab ;
```

```
BEGIN
/* 動的 SQL 文の組み立て          */
wk_sql := 'SELECT * FROM emp_table ' ;
wk_sql := wk_sql || str_where;      -- WHERE 条件の追加
-- SQL 文の実行
EXECUTE IMMEDIATE wk_sql BULK COLLECT INTO dim_emp_record;
```

配列への値のセット

```
配列名 (添字) := 値 ;
配列名 (添字). 列の名前 := 値 ;
```

```
sample(10) := '値'
```

【レコード型配列の2次元配列】

配列の定義

```
TYPE 配列名 IS TABLE OF テーブル名%ROWTYPE
      INDEX BY PLS_INTEGER ;
```

もしくは、

```
CURSOR カーソル名 IS SELECT 文 ;
TYPE 配列名 IS TABLE OF カーソル名%ROWTYPE
      INDEX BY PLS_INTEGER ;
```

配列への値のセット

```
配列名 (添字). 列の名前:= 値 ;
```

(カーソルを使ったレコード読取り例) (P.94)

DECLARE

```
/* カーソルの定義 */
```

```
CURSOR cur_emp IS SELECT emp_table.ename retsu1, sal_table.sal retsu2
      FROM emp_table, sal_table ;
```

```
/* カーソルを使ったテーブル型のデータ型定義 */
```

```
TYPE typ_emp_tab IS TABLE OF cur_emp%ROWTYPE
      INDEX BY PLS_INTEGER ;
```

```
/* 配列の変数定義 */
```

```
dim_emp typ_emp_tab ;
```

BEGIN

```
/* カーソルのオープン */
```

```
OPEN cur_emp ;
```

```
/* フェッチ操作 */
```

```
FETCH cur_emp BULK COLLECT INTO dim_emp ;
```

```
-- BLUK COLLECT INTO は、SELECT INTO の複数レコード対応型
```

```
/* カーソルのクローズ */
```

```
CLOSE cur_emp ;
```

```
/* 配列に入ったデータに対する処理 */
```

```
IF dim_emp.COUNT = 0 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('レコード 0件 ');
```

```
ELSE
```

```
    FOR i IN dim_emp.FIRST.. dim_emp.LAST LOOP
```

```
        DBMS_OUTPUT.PUT_LINE( dim_emp( i ). retsu1 ||  
            ':' || dim_emp(i).retsu2 ) ;  
    END LOOP ;  
END IF ;  
END ;
```

サンプル・コード（実行実証済）

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```
/* カーソルの定義 */
```

```
CURSOR cur_emp IS SELECT empno retsu1,  ename  retsu2  
                    FROM scott.emp ;
```

```
/* カーソルを使ったテーブル型のデータ型定義 */
```

```
TYPE typ_emp_tab IS TABLE OF cur_emp%ROWTYPE  
                    INDEX BY PLS_INTEGER ;
```

```
/* 配列の変数定義 */
```

```
dim_emp typ_emp_tab ;
```

```
BEGIN
```

```
/* カーソルのオープン */
```

```
OPEN cur_emp ;
```

```
/* フェッチ操作 */
```

```
FETCH cur_emp BULK COLLECT INTO dim_emp ;
```

```
-- BLUK COLLECT INTO は、SELECT INTO の複数レコード対応型
```

```
/* カーソルのクローズ */
```

```
CLOSE cur_emp ;
```

```
/* 配列に入ったデータに対する処理 */
```

```
IF dim_emp.COUNT = 0 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('レコード 0件 ');
```

```
ELSE
```

```
    FOR i IN dim_emp.FIRST.. dim_emp.LAST LOOP
```

```
        DBMS_OUTPUT.PUT_LINE( dim_emp( i ). retsu1 ||  
                               ':' || dim_emp(i). retsu2 ) ;
```

```
    END LOOP ;
```

```
END IF ;
```

```
END ;
```

```
/
```