

プロシージャとファンクションの作成と呼出し —ロジック内からの使用—

プロシージャとファンクションを持った「無名 PL/SQL ブロック」の作成方法

DECLARE

```
-- カーソルや変数の定義宣言は、ファンクションやプロシージャより先に定義
wk_name emp.ename%type ;
CURSOR c_emp IS SELECT emp_no FROM emp ;
```

```
-- プロシージャの作成
```

```
PROCEDURE プロシージャ名 (
  i_param IN NUMBER ,
  o_param OUT NUMBER ,           -- 受渡し引数定義
  . . . . . ) IS ≡
  ローカル変数1 データ型 ≡ ;    -- ローカル変数定義
  ローカル変数2 データ型 ≡ ;    . . .
```

BEGIN

DECLARE

ブロック内限定のスコープ変数 定義

BEGIN

プログラム部

END ;

```
END プロシージャ名 ;
```

```
-- ファンクションの作成
```

```
FUNCTION ファンクション名 (
  i_param IN NUMBER ,
  o_param OUT NUMBER ,           -- 受渡し引数定義
  . . . . . )
```

```
RETURN 戻り値のデータ型 IS ≡
```

```
ローカル変数1 データ型 ≡ ;    -- ローカル変数定義
```

```
ローカル変数2 データ型 ≡ ;    . . . . .
```

BEGIN

DECLARE

ブロック内限定のスコープ変数 定義

BEGIN

プログラム部

END ;

```
END ファンクション名 ;
```

```

-- 無名 PL/SQL ブロック プログラム本体部分
BEGIN

-- ファンクションの呼出し
DECLARE
  a NUMBER;
  c NUMBER;
BEGIN
  a := func_b(2, c); -- ファンクションの呼出しと戻り値のセット
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(c));

END;

-- プロシージャの呼出し
DECLARE
  a VARCHAR2(20);
  c NUMBER;
BEGIN
  a := 0; -- 変数への値の代入
  proc_c(a, 2, c); -- プロシージャの呼出し
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(c));

END;
END; --無名 PL/SQL ブロック プログラム本体部の END
/
/* 実行のための「/」 */

```

エラーについて

Select 文単体では使用出来るが、**プロシージャ内からは使用できないビューが存在する**

このビューを使用すると、以下のようなエラーが発生した

警告: プロシージャが作成されましたが、コンパイル・エラーがあります。

LINE/COL	ERROR
----------	-------

6/1	PL/SQL: SQL Statement ignored
-----	-------------------------------

6/42	PL/SQL: ORA-00942: 表またはビューが存在しません。
------	------------------------------------

対応としては、SYS ユーザーを SYSDBA 接続してプロシージャを作成した