

## プロシージャとファンクションの作成と呼出し —ストアードプログラムとして—

### ストアードプロシージャの定義方法

```
CREATE OR REPLACE 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 ;  
    RETURN ;  
END プロシージャ名 ;
```

### ファンクションの定義方法

```
CREATE OR REPLACE 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 NUMBER 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 ;  
    RETURN 戻り値 ;  
END ファンクション名 ;
```

サンプルコード（テスト済）

———ファンクション———

```
CREATE OR REPLACE FUNCTION Func_A(  
    i_param    IN    NUMBER ,  
    o_param    OUT    NUMBER )  
RETURN NUMBER IS  
    o_Return   NUMBER;  
    wk_data    VARCHAR2(20);  
BEGIN  
    o_Return   :=  5;  
    return  o_Return;  
END  Func_A  ;  
/  
SHOW ERROR FUNCTION  Func_A  ;
```

——— プロシージャ ———

```
CREATE OR REPLACE PROCEDURE Proc_A(  
    i_param    IN    NUMBER ,  
    o_param    OUT    VARCHAR2  
    ) IS  
    wk_data    VARCHAR2(20);  
BEGIN  
    o_param :=  '呼出し成功!';  
    return;  
END  Proc_A  ;  
/  
SHOW ERROR PROCEDURE  Proc_A  ;
```

## プロシージャやファンクションのコンパイル・エラー発生箇所の表示方法

ストアドプログラムの登録(CREATE OR REPLACE PROCEDURE 文)では、登録時にコンパイル・エラーが発生しても、登録処理は行われます。

どこでコンパイル・エラーが発生したか確認は、別途コマンドで行います。

```
SHOW ERROR PROCEDURE プロシージャ名 ;
```

```
SHOW ERROR FUNCTION ファンクション名 ;
```

### 【SQL\*PLUS】

#### プロシージャ登録時のコンパイル・エラー対処の方法

PL/SQL セッションに対する、警告エラー表示受付の設定

```
SQL> ALTER SESSION SET PLSQL_WARNINGS = 'ENABLE:ALL';
```

PL/SQL プロシージャの登録

```
SQL>
```

PL/SQL 無名ブロックを使って、プロシージャの登録

```
( CREATE OR REPLACE PROCEDURE . . . . . )
```

コンパイル時のエラー表示操作

```
SQL> SHOW ERRORS
```

#### 実行時のエラー確認コマンド

どのオブジェクトのどの部分の実行で、エラーが発生したか調査を行う

```
select NAME, LINE, POSITION, TEXT, MESSAGE_NUMBER  
from user_errors ;
```

データベースに登録されたオブジェクト  
(テーブル、ストアドプロシージャ)の存在確認

## PL/SQL プログラムからの プロシージャ、ストアドプロシージャ、パッケージ・プロシージャの呼出し方法

プロシージャの呼出し方法

ファンクションの呼出し方法

プロシージャ名(引数);

戻り値の変数 = ファンクション名(引数);

ストアドプロシージャの呼出し方法

ストアドファンクションの呼出し方法

スキーマ名. プロシージャ名(引数);

戻り値の変数 = スキーマ名. ファンクション名(引数);

↑

ファンクション作成ユーザー

パッケージ・プロシージャの呼出し方法

パッケージ・ファンクションの呼出し方法

パッケージ名. プロシージャ名(引数);

戻り値の変数 = パッケージ名. ファンクション名(引数);

—— パッケージ・プロシージャの呼出し 例 ——

```
BEGIN
```

```
  DECLARE
```

```
    a  VARCHAR2 ( 20 );
```

```
    c  NUMBER ;
```

```
  BEGIN
```

```
    a := 0 ;                                -- 変数への値の代入
```

```
    EXECUTE スキーマ名.proc_c ( a, 2, c ) ; -- プロシージャの呼出し
```

```
    DBMS_OUTPUT.PUT_LINE ( TO_CHAR ( c ) ) ;
```

```
  END ;
```

```
END ;
```

```
/
```

PL/SQL からのファンクションの呼出し方と変数定義

※ 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);

```
BEGIN
  DECLARE
    a  NUMBER ;
    c  NUMBER ;
  BEGIN
    a  := func_A ( 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 ( a ) ) ;

  END ;

END ;
/
```

SQL\*PLUS からの無名 PL/SQL ブロックのコーディングと実行

```
SQL> BEGIN
      DECLARE
        i_param  NUMBER ;          -- 変数定義
        o_param  VARCHAR2(50) ;    -- 変数定義
      BEGIN
        /*   プログラム部   */
        i_param := 5 ;
        proc_A ( i_param , o_param ) ;    -- プロシージャ呼出し
        DBMS_OUTPUT.PUT_LINE ( o_param ) ;
      END ;
    END ;
SQL> /
/*   実行させるために、「/」をキー入力する   */
```

## ストアドプロシージャの実行権限の付与

ストアドプロシージャの作成者とは異なるユーザーがストアドプロシージャの実行を行う場合は、実行ユーザーへの実行権限の付与が必要となる

GRANT EXECUTE ON ストアドプログラム名 TO ユーザー名 ;

※ ユーザー名に、PUBLIC を指定すると、すべての人に実行権限が付与される

## プロシージャのデバッグを行うための権限付与

作成中のプロシージャについて、デバッグ操作を行うためには、操作を行うユーザーへ以下の権限の付与については、

## プロシージャの登録におけるエラーについて

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