

RDBMS-02-Function & Procedure

Functions

- Functions is a standalone PL/SQL subprogram. Functions have a unique name by which it can be referred. These are stored as PL/SQL database objects
- **Characteristics of Functions:**
 - Functions are a standalone block that is mainly used for calculation purpose.
 - Function use RETURN keyword to return the value, and the datatype of this is defined at the time of creation
 - A Function should either return a value or raise the exception, i.e. return is mandatory in functions
 - Function with no DML statements can be directly called in SELECT query whereas the function with DML operation can only be called from other PL/SQL blocks.
 - It can have nested blocks, or it can be defined and nested inside the other blocks or packages.
 - It contains declaration part (optional), execution part, exception handling part (optional).
 - The values can be passed into the function or fetched from the procedure through the parameters.
 - These parameters should be included in the calling statement.
 - A PLSQL function can also return the value through OUT parameters other than using RETURN.

```
CREATE [OR REPLACE] FUNCTION function_name [parameters]
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN return_datatype
{IS | AS}
BEGIN
    < function_body >
END [function_name];
```

- The function must contain a return statement. RETURN clause specifies that data type you are going to return from the function.
- Function body contains the executable part.
- The AS keyword is used instead of the IS keyword for creating a standalone function.

```
CREATE OR REPLACE FUNCTION adder(n1 IN number, n2 IN number)
RETURN number
IS
n3 number(8);
BEGIN
    n3 :=n1+n2;
    RETURN n3;
END;
```

```
-- To call the function --
DECLARE
    n3 number(2);
BEGIN
    n3 := adder(11,22);
    dbms_output.put_line('Addition is: ' || n3);
END;
```

```
DECLARE
    a number;
    b number;
    c number;
FUNCTION findMax(x IN number, y IN number)
RETURN number
IS
    z number;
BEGIN
    IF x > y THEN
        z:= x;
    ELSE
        z:= y;
    END IF;
    RETURN z;
END;
BEGIN
    a:= 23;
    b:= 45;
    c := findMax(a, b);
    dbms_output.put_line(' Maximum of (23,45): ' || c);
END;
```

- To remove already created function use `DROP FUNCTION function_name`

Procedure

- A procedure is a PL/SQL block which performs one or more specific tasks.
- The procedure contains a header and a body
 - **Header:** The header contains the name of the procedure and the parameters or variables passed to the procedure.
 - **Body:** The body contains a declaration section, execution section and exception section similar to a general PL/SQL block.
- When you want to create a procedure or function, you have to define parameters.
- There are three ways to pass parameters in procedure:
 - **IN parameters:** The IN parameter can be referenced by the procedure or function. The value of the parameter cannot be overwritten by the procedure or the function.
 - **OUT parameters:** The OUT parameter cannot be referenced by the procedure or function, but the value of the parameter can be overwritten by the procedure or function.
 - **INOUT parameters:** The INOUT parameter can be referenced by the procedure or function and the value of the parameter can be overwritten by the procedure or

function.

```
CREATE [OR REPLACE] PROCEDURE procedure_name
    [ (parameter [,parameter]) ]
IS [declaration_section]
BEGIN
    executable_section
EXCEPTION
    exception_section
END procedure_name;
```

```
-- Table Creation --
CREATE TABLE user(id number(10) primary key,name varchar2(100));
-- Creating a Procedure --
CREATE OR REPLACE PROCEDURE "INSERTUSER"
    (id IN NUMBER, name IN VARCHAR2)
IS
BEGIN
    INSERT INTO user VALUES(id,name);
END;
-- Program to Call Procedure --
BEGIN
    insertuser(101,'xyz');
    dbms_output.put_line('record inserted successfully');
END;
```

- To drop already created procedure use `DROP PROCEDURE name;`

Stored procedure

- A stored procedure is a series of declarative SQL statements which can be stored in the database catalogue.
- A procedure can be thought of as a function or a method.
- They can be invoked through triggers, other procedures, or applications on Java, PHP etc.
- **Advantages:**
 - They result in performance improvement of the application.
 - If a procedure is being called frequently in an application in a single connection, then the compiled version of the procedure is delivered.
 - They reduce the traffic between the database and the application, since the lengthy statements are already fed into the database and need not be sent again and again via the application.
 - They add to code reusability, similar to how functions and methods work in other languages such as C/C++ and Java.
- **Disadvantages:**
 - Stored procedures can cause a lot of memory usage. The database administrator should decide an upper bound as to how many stored procedures are feasible for a particular application.
 - MySQL does not provide the functionality of debugging the stored procedures

Difference between Procedure and Function

Procedure	Function
Used mainly to execute certain business logic with DML and DRL statements	Used mainly to perform some computational process and returning the result of that process.
Procedure can return zero or more values as output.	Function can return only single value as output.
Procedure cannot call with select statement, but can call from a block or from a procedure.	Function can call with select statement, if function does not contain any DML statements and DDL statements. function with DML and DDL statements can call with select statement with some special cases (using Pragma autonomous transaction)
OUT keyword is used to return a value from procedure	RETURN keyword is used to return a value from a function
It is not mandatory to return the value.	It is mandatory to return the value
RETURN will simply exit the control from subprogram.	RETURN will exit the control from subprogram and also returns the value.
Return datatype will not be specified at the time of creation.	Return datatype is mandatory at the time of creation

Questions

1. Explain PLSQL function with example. (MID-05M)
2. Explain basic difference between Procedure and Function. (MID-05M)
3. Explain types of parameters in stored procedure and how to pass parameters in procedure. (MID-05M)
4. Explain syntax of creating stored procedure using PLSQL (MID-06M)



@SOUBCA

Join Us on Telegram @[SOU BCA](https://t.me/SOU_BCA)