K. Abhishek
TCOA76

DBMSL

Camlin Page
Date / /

Assignment No. : 6

Aim

PL/SQL stored procedure and stored function. Write a stored procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks >=900 then student will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 and 825 category is Higher second class. Write a PL/SQL block for using procedure created with above requirement. Stud_Marks (name, total_marks) Result (Roll, Name, class).

Objective

Learning the concept of procedure, function & package in PL/SQL.

Theory

Procedure

A procedure is a subprogram that performs a specific action or task. A procedure has two parts.

i) Procedure specification: The procedure specification specifies the procedure name and the parameters it accept. It is not necessary to create a procedure that accepts parameters.

ii) Procedure body: The procedure body contains the declarative section without DECLARE keyword, the executable section and an exception section.

Syntax:

CREATE PROCEDURE procedure_name [(argument 1 type), ...]
IS/AS
Procedure_body

Deleting procedure

To remove a procedure from database

Drop procedure <procedure_name>;

Function

A function is a subprogram, which is used to compute values. It is similar to a procedure, functions also take arguments and can be in different modes. Functions also can be stored in the database. It is a PL/SQL block consisting of declarative, execute and exception section.

A function can return more than one value using OUT parameter.

Syntax:

Create FUNCTION function_name
[(argument1 [IN/OUT/INOUT] type), ...]
Return return_type IS/AS
Function_body.

Deleting a Function

To remove the subprogram from the database.

Drop function <function_name>;

Package

A package is a PL/SQL construct that allows related objects to be stored together. A package has 2 separate parts: the specification & body. Each of them stored separately in the data dictionary.

Conclusion

Thus, we learned concept of procedure, function & package in PL/SQL.

```
 > create table stud_marks(roll_no int primary key, name char(10), total_marks
    -> int);
Query OK, 0 rows affected (0.07 sec)

 > insert into stud_marks values(101,'abc', 933);
Query OK, 1 row affected (0.01 sec)

 > insert into stud_marks values(102,'xyz', 356);
Query OK, 1 row affected (0.02 sec)

 > insert into stud_marks values(103,'mnk', 450);
Query OK, 1 row affected (0.05 sec)

 > insert into stud_marks values(104,'pqr', 675);
Query OK, 1 row affected (0.02 sec)

 > insert into stud_marks values(105,'snk', 1300);
Query OK, 1 row affected (0.01 sec)

 > insert into stud_marks values(106,'jnk', 250);
Query OK, 1 row affected (0.01 sec)

 > select * from stud_marks;
+---------+------+-------------+
| roll_no | name | total_marks |
+---------+------+-------------+
|     101 | abc  |         933 |
|     102 | xyz  |         356 |
|     103 | mnk  |         450 |
|     104 | pqr  |         675 |
|     105 | snk  |        1300 |
|     106 | jnk  |         250 |
+---------+------+-------------+
6 rows in set (0.00 sec)

 > create table result (roll_no int, name char(30), class char(20))//
Query OK, 0 rows affected (0.05 sec)

 > delimiter //

 > create procedure proc_grade(in marks int, out class char(10))
    -> begin
    -> if marks<=1500 and marks>=990 then set class='DIST';
    -> end if;
    -> if marks<=989 and marks>=900 then set class='FC';
    -> end if;
    -> if marks<=899 and marks>=825 then set class='HSC';
    -> end if;
    -> if marks<=824 and marks>=750 then set class='SC';
    -> end if;
    -> if marks<=749 and marks>=650 then set class='PC';
    -> end if;
    -> if marks < 650 then set class='fail';
    -> end if;
```

```
    -> end ;
    -> //
Query OK, 0 rows affected, 3 warnings (0.02 sec)

 > create function find_result(roll_in int) returns int deterministic
    -> begin
    -> declare fmarks int;
    -> declare grade char(10);
    -> declare stud_name char(10);
    -> select stud_marks.total_marks,stud_marks.name into fmarks, stud_name from
    -> stud_marks where stud_marks.roll_no=roll_in;
    -> call proc_grade(fmarks,@grade);
    -> insert into result values(roll_in,stud_name, @grade);
    -> return roll_in; end;//
Query OK, 0 rows affected (0.00 sec)

 > select find_result(104)//
+------------------+
| find_result(104) |
+------------------+
|              104 |
+------------------+
1 row in set (0.02 sec)

 > select * from result //
+---------+------+-------+
| roll_no | name | class |
+---------+------+-------+
|     104 | pqr  | PC    |
+---------+------+-------+

 > select find_result (101)//
+--------------------+
| find_result (101) |
+--------------------+
|               101 |
+--------------------+
1 row in set (0.02 sec)

 > select * from result //
+---------+------+-------+
| roll_no | name | class |
+---------+------+-------+
|     104 | pqr  | PC    |
|     101 | abc  | FC    |
+---------+------+-------+
2 rows in set (0.00 sec)

 > select find_result (106)//
+--------------------+
| find_result (106) |
+--------------------+
|               106 |
+--------------------+
```

```
1 row in set (0.02 sec)

 > select * from result //
+---------+------+-------+
| roll_no | name | class |
+---------+------+-------+
|     104 | pqr  | PC    |
|     101 | abc  | FC    |
|     106 | jnk  | fail  |
+---------+------+-------+

 > select find_result (104)//
+--------------------+
| find_result (104) |
+--------------------+
|                104 |
+--------------------+
1 row in set (0.02 sec)

 > select * from result //
+---------+------+-------+
| roll_no | name | class |
+---------+------+-------+
|     104 | pqr  | PC    |
|     101 | abc  | FC    |
|     106 | jnk  | fail  |
|     104 | pqr  | PC    |
+---------+------+-------+
4 rows in set (0.00 sec)
```