Database Management Systems Lab Experiment No: - 10

Aim: - Implementation of functions & procedures in SQL

Procedures:-

A procedure is a named PL/SQL block which performs one or more specific task. This is similar to a procedure in other programming languages. A procedure has a header and a body. The header consists of thename of the procedure and the parameters or variables passed to the procedure. The body consists or declaration section, execution section and exception section similar to a general PL/SQL Block. A procedureis similar to an anonymous PL/SQL Block but it is named for repeated usage.

We can pass parameters to procedures in three ways :

Parameters	Description
IN type	These types of parameters are used to send values to stored procedures.
OUT type	These types of parameters are used to get values from stored procedures. This is similar to a return type in functions.
IN OUT type	These types of parameters are used to send values and get values from stored procedures.

A procedure may or may not return any value.

Examples :

1)

```
mysql> delimiter &&
mysql> create procedure pro(in cn int)
    -> begin
    -> select * from Client_master limit cn;
    -> end;
    -> &&
Query OK, 0 rows affected (0.02 sec)

mysql> call pro(3);
    -> &&
+-----------+-----------------+--------+-------------+---------+----------+
| Client_no | Name            | City   | State       | Pincode | Bal_due  |
+-----------+-----------------+--------+-------------+---------+----------+
| C00001    | Ivan Bayross    | Bombay | Maharashtra |  400054 | 15000.00 |
| C00002    | Vandana Saitwal | Madras | Tamil Nadu  |   78001 |     0.00 |
| C00003    | Pramada Jagutse | Bombay | Maharashtra |  400057 |  5000.00 |
+-----------+-----------------+--------+-------------+---------+----------+
3 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

Aim: - Implementation of functions & procedures in SQL

2)

```
mysql> delimiter &&
mysql> create procedure pro2(out client_no int)
    -> begin
    -> select count(*)into client_no from Client_master;
    -> end;
    -> &&
Query OK, 0 rows affected (0.00 sec)

mysql> call pro2(@s)
    -> ;
    -> &&
Query OK, 1 row affected (0.00 sec)

mysql> select @s;
    -> &&
+------+
| @s   |
+------+
|    6 |
+------+
1 row in set (0.00 sec)
```

Database Management Systems Lab Experiment No: - 10

Aim: - Implementation of functions & procedures in SQL

Functions:-

A function is a named PL/SQL Block which is similar to a procedure. The major difference between a procedure and a function is, a function must always return a value, but a procedure may or may not return a value.

Query:

```
mysql> $$ create function proprice(sell_price double)RETURNS varchar(20)
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE lvl varchar(20);
    -> IF sell_price<1000 THEN SET lvl='CHEAP'; ELSEIF sell_price>3000 THEN SET lvl='EXPENSIVE'; END IF; RETURN (lvl); END; $$
    -> ;
```

Output:

```
mysql> select product_no,proprice(sell_price) from Product_master;
    -> &&
+------------+----------------------+
| product_no | proprice(sell_price) |
+------------+----------------------+
| P00001     | CHEAP                |
| P03453     | EXPENSIVE            |
| P06734     | NULL                 |
| P07865     | CHEAP                |
| P07868     | EXPENSIVE            |
| P07885     | EXPENSIVE            |
| P07965     | EXPENSIVE            |
| P07975     | NULL                 |
| P08865     | NULL                 |
+------------+----------------------+
9 rows in set (0.00 sec)
```

Database Management Systems Lab Experiment No: - 10

Aim: - Implementation of functions & procedures in SQL

Query:

```
mysql> DELIMITER $$
mysql> CREATE FUNCTION calculate_discount(original_price DECIMAL(10, 2), discount_rate DECIMAL(4, 2)) RETURNS DECIMAL(10, 2)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE discounted_price DECIMAL(10, 2);
    ->
    ->     SET discounted_price = original_price - (original_price * (discount_rate / 100));
    ->
    ->     RETURN discounted_price;
    -> END$$
Query OK, 0 rows affected (0.00 sec)
```

Output:

```
mysql> SELECT calculate_discount(100, 10);
+-----------------------------+
| calculate_discount(100, 10) |
+-----------------------------+
|                       90.00 |
+-----------------------------+
1 row in set (0.00 sec)
```

Database Management Systems Lab Experiment No: - 10

Aim: - Implementation of functions & procedures in SQL

Procedures VS Functions:

- A function MUST return a value
- A procedure cannot return a value
- Procedures and functions can both return data in OUT and IN OUT parameters
- The return statement in a function returns control to the calling program and returns the results of the function
- The return statement of a procedure returns control to the calling program and cannot return a value
- Functions can be called from SQL, procedure cannot
- Functions are considered expressions, procedure are not

Functions	Procedures
A function has a return type and returns a value.	A procedure does not have a return type. But it returns values using the OUT parameters.
You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions.	You can use DML queries such as insert, update, select etc… with procedures.
A function does not allow output parameters	A procedure allows both input and output parameters.
You cannot manage transactions inside a function.	You can manage transactions inside a function.
You cannot call stored procedures from a function	You can call a function from a stored procedure.
You can call a function using a select statement.	You cannot call a procedure using select statements.

Conclusion:- LO mapped LO3, LO4.