Assignment 07

Aim: Write and execute PL/SQL stored procedure and function to perform a suitable task on the database.

Objectives:

- To learn about MySQL stored procedures, their advantages and disadvantages.
- To learn about variables in stored procedure, its scope, how to declare and use variables.
- To learn how to write MySQL stored procedures with parameters.
- To learn how to create stored functions using CREATE FUNCTION statement.

Theory:

A subprogram is a program unit/module that performs a particular task. These subprograms are combined to form larger programs. This is basically called the 'Modular design'. A subprogram can be invoked by another subprogram or program which is called the calling program.

A subprogram can be created:

- At schema level
- Inside a package
- Inside a MYSQL block

Parts of a MYSQL Subprogram:

Each MYSQL subprogram has a name, and may have a parameter list. Like anonymous PL/SQL blocks and, the named blocks a subprograms will also have following three parts:

1. Declarative Part
2. Executable part
3. Exception-handling

Definition of stored procedures

A stored procedure is a segment of declarative SQL statements stored inside the database catalog. A stored procedure can be invoked by triggers, other stored procedures or applications such as Java, C#, PHP, etc.

A stored procedure that calls itself is known as a recursive stored procedure. Most database management system supports recursive stored procedures.

However MySQL does not support it very well. You should check your version of MySQL database before implementing recursive stored procedures in MySQL.

MySQL stored procedures advantages

- Typically stored procedures help increase the performance of the applications. Once created, stored procedures are compiled and stored in the database. However MySQL implements the stored procedures slightly different. MySQL stored procedures are compiled on demand. After compiling a stored procedure, MySQL puts it to a cache. And MySQL maintains its own stored procedure cache for every single connection. If an application uses a stored procedure multiple times in a single connection, the compiled version is used, otherwise the stored procedure works like a query.
- A stored procedure helps reduce the traffic between application and database server because instead of sending multiple lengthy SQL statements, the application has to send only name and parameters of the stored procedure.
- Stored procedures are reusable and transparent to any applications. Stored procedures expose the database interface to all applications so that developers don't have to develop functions that are already supported in stored procedures.
- Stored procedures are secure. Database administrator can grant appropriate permissions to applications that access stored procedures in the database without giving any permission on the underlying database tables.

Besides those advantages, stored procedures have their own disadvantages, which

you should be aware of before using the store procedures.

MySQL stored procedures disadvantages

- If you use a lot of stored procedures, the memory usage of every connection that is using those stored procedures will increase substantially. In addition, if you overuse a large number of logical operations inside store procedures, the CPU usage will also increase because database server is not well-designed for logical operations.
- A constructs of stored procedures make it more difficult to develop stored procedures that have complicated business logic.
- It is difficult to debug stored procedures. Only few database management systems allow you to debug stored procedures. Unfortunately, MySQL does not provide facilities for debugging stored procedures.
- It is not easy to develop and maintain stored procedures. Developing and maintaining stored procedures are often required specialized skill set that not all application developers possess. This may lead to problems in both application development and maintenance phases.

Creating a Procedure:

A procedure is created with the CREATE OR REPLACE PROCEDURE statement. The simplified syntax for the CREATE OR REPLACE PROCEDURE statement is as follows:

CREATE [OR REPLACE] PROCEDURE procedure_name [(parameter_name [IN | OUT | IN OUT] type [, ...])]

BEGIN

< procedure_body

> END ;

Where,

procedure-name specifies the name of the procedure.

[OR REPLACE] option allows modifying an existing procedure.

The optional parameter list contains name, mode and types of the parameters. IN represents, that value will be passed from outside and OUT represents that this parameter will be used to return a value outside of the procedure.

Procedure-body contains the executable part.

The IS keyword is used for creating a standalone procedure.

How to execute procedure?

Executing a Standalone Procedure

Calling the name of the procedure from a PL/SQL block

Call greeting( )

Deleting a Standalone Procedure

A standalone procedure is deleted with the DROP PROCEDURE statement. Syntax for deleting a procedure is:

DROP PROCEDURE procedure-name;

Parameter modes in PL/SQL subprograms:

1. IN:

An IN parameter lets you pass a value to the subprogram. It is a read-only parameter. It is the default mode of parameter passing. Parameters are passed by reference.

2. OUT:

An OUT parameter returns a value to the calling program. The actual parameter must be

variable and it is passed by value.

3. IN-OUT:

An IN OUT parameter passes an initial value to a subprogram and returns an updated value to the caller. Actual parameter is passed by value.

Calling Procedure:

```
mysql> delimiter ;
mysql> call addp();
```

```
mysql> delimiter //
mysql> create procedure difference (in a int,in b int, out c int)
-> begin
-> if a>b then
-> set c=1;
-> else if a=b then
-> set c=2;
-> else
-> set c=3;
-> end if;
```

```
mysql> call difference(5,9,@x);
-> select @x;
-> //
```

PROCEDURES ON TABLES

To run the procedures on table, lets create a sample table and insert some values in that.

```
mysql> create table student ( sid int(5) not null,
-> student_name varchar(9),
-> DOB date,
-> primary key(sid));
Query OK, 0 rows affected (0.06 sec)
```

Insert few rows in the table created above.

Conclusion:

In this assignment, we have learned how to write a simple stored procedure by using the CREATE PROCEDURE statement and call it by using the CALL statement. Also we have learnt, creation of function using CREATE FUNCTION and calling it by its name.