LAB-9 PROCEDURE AND FUNCTION

A stored procedure is a group of one or more pre-compiled SQL statements into a logical unit. It is stored as an object inside the database server. It is a subroutine or a subprogram in the common computing language that has been created and stored in the database. Each procedure in SQL Server always contains a name, parameter lists, and Transact-SQL statements. The SQL Database Server stores the stored procedures as named objects. We can invoke the procedures by using triggers, other procedures, and applications such as Java, Python, PHP, etc. It can support almost all relational database systems.

In MySQL, Function can also be created. A function always returns a value using the return statement. The function can be used in SQL queries.

Parameter:

Function_name: name of the function

Parameter: number of parameter. It can be one or more than one.

return_datatype: return value datatype of the function

declaration_section: all variables are declared.

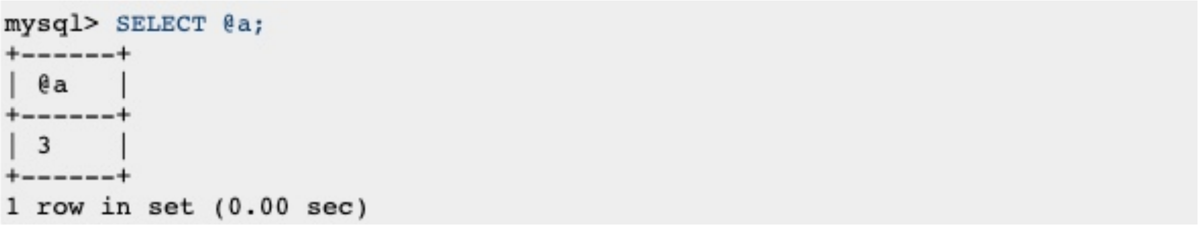
executable_section: code for the function is written here.

CREATE PROCEDURE AND CREATE FUNCTION SYNTAX

```
CREATE
    [DEFINER = { user | CURRENT_USER }]
    PROCEDURE sp_name ([proc_parameter[,...]])
    [characteristic ...] routine_body

CREATE
    [DEFINER = { user | CURRENT_USER }]
    FUNCTION sp_name ([func_parameter[,...]])
    RETURNS type
    [characteristic ...] routine_body

proc_parameter:
    [ IN | OUT | INOUT ] param_name type

func_parameter:
    param_name type

type:
    Any valid MySQL data type

characteristic:
    LANGUAGE SQL
  | [NOT] DETERMINISTIC
  | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
  | SQL SECURITY { DEFINER | INVOKER }
  | COMMENT 'string'
routine_body:
    Valid SQL procedure statement
```

The following is an example of a simple stored procedure that uses an OUT parameter. The example uses the mysql client delimiter command to change the statement delimiter from ; to // while the procedure is being defined. This allows the ; delimiter used in the procedure body to be passed through to the server rather than being interpreted by mysql itself.

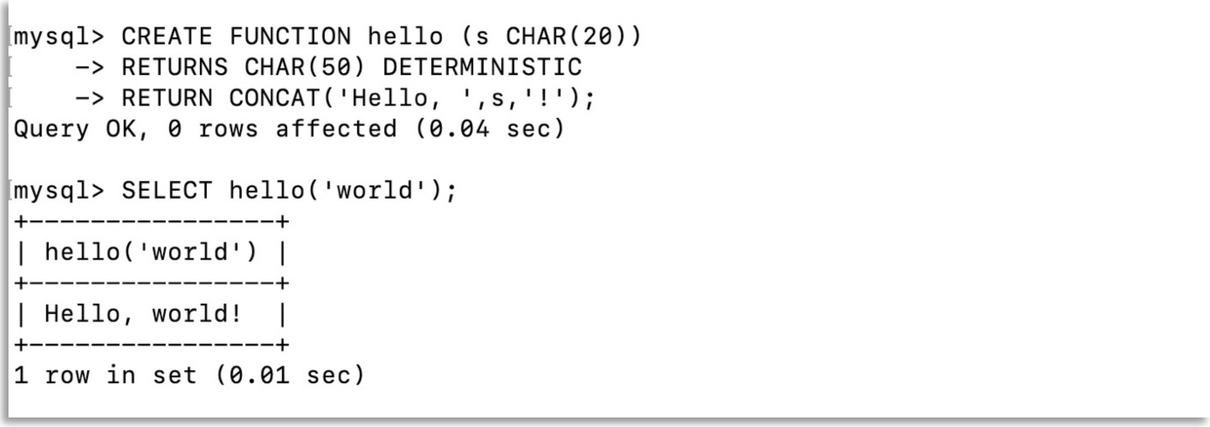
```
mysql> delimiter //

mysql> CREATE PROCEDURE simpleproc (OUT param1 INT)
   -> BEGIN
   ->   SELECT COUNT(*) INTO param1 FROM t;
   -> END;
   -> //
mysql> delimiter ;
mysql> CALL simpleproc(@a);
```



```
mysql> SELECT @a;
+------+
| @a   |
+------+
| 3    |
+------+
1 row in set (0.00 sec)
```

The following is an example of a function that takes a parameter, performs an operation using an SQL function, and returns the result. In this case, it is unnecessary to use delimiter because the function definition contains no internal ; statement delimiters:

```
mysql>CREATE FUNCTION hello (s CHAR(20))
        -> RETURNS CHAR(50) DETERMINISTIC
        -> RETURN CONCAT('Hello, ',s,'!');
```



```
mysql> CREATE FUNCTION hello (s CHAR(20))
    -> RETURNS CHAR(50) DETERMINISTIC
    -> RETURN CONCAT('Hello, ',s,'!');
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT hello('world');
+----------------+
| hello('world') |
+----------------+
| Hello, world!  |
+----------------+
1 row in set (0.01 sec)
```