

## Laboratory 8

### Title of the Laboratory Exercise: Stored Procedure in MySQL

#### 1. Introduction and Purpose of Experiment

A stored Procedure is a procedure stored in a database which can be called by the database engine and connected programming languages. A stored procedure is invoked using the CALL statement. A procedure has a name, a parameter list, and SQL statement(s). The parameters make the stored procedure more flexible and useful. In MySQL, a parameter has one of three modes: IN, OUT, or INOUT. By doing this lab, students will be able to implement MySQL stored procedure.

#### 2. Aim and Objectives

Aim

- To implement MySQL stored procedure

Objectives

At the end of this lab, the student will be able to

- Design SQL procedures for the given problem statement
- Implement the procedures in MySQL

#### 3. Experimental Procedure

- Analyse the problem statement
- Create the table and its attributes with essential properties
- Design the procedures in MySQL
- Implement the procedures in MySQL
- Test the implemented procedures
- Document the Results
- Analyse and discuss the outcomes of your experiment

#### 4. Questions

- Design and implement a procedure which accepts one INOUT parameter (count) and one IN parameter (inc). Inside the stored procedure, increase the counter (count) by the value of the inc parameter.
- Create a table OFFICES in OFFICEDB with attributes such as OfficeCode, Name, City, Country, and Phone.
  - Write a 'GetPhoneNo' procedure in MySQL to take the name of a person as an input and display the phone number of that person.

- ii. Write a 'GetOffice' procedure in MySQL to take the name of a country as an input and display the number of offices located in a particular Country.

## 5. Presentation of Results

a.

```
DROP procedure IF EXISTS IncreaseCount;

DELIMITER //

CREATE procedure IncreaseCount ( INOUT ending_value INT, IN inc_value INT )

BEGIN

    SET ending_value = ending_value + inc_value ;

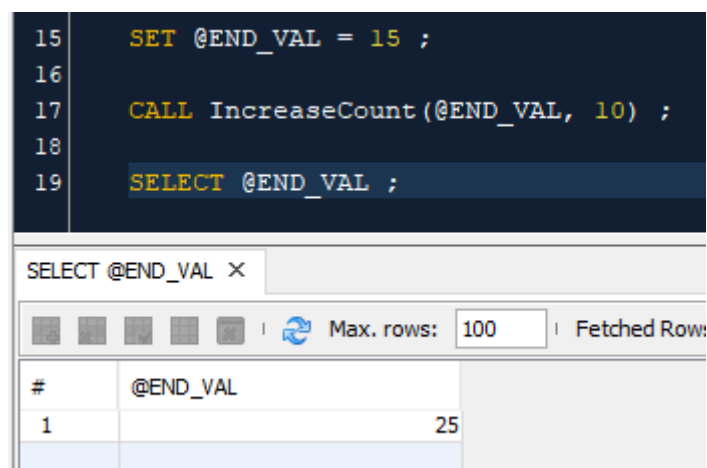
END; //

DELIMITER ;

SET @END_VAL = 15 ;

CALL IncreaseCount(@END_VAL, 10) ;

SELECT @END_VAL ;
```



The screenshot shows a MySQL command window with the following SQL statements executed:

```
15 SET @END_VAL = 15 ;
16
17 CALL IncreaseCount(@END_VAL, 10) ;
18
19 SELECT @END_VAL ;
```

Below the command window, the result set for the final SELECT statement is displayed:

#	@END_VAL
1	25

Figure 1 Output

b (i) :

```
DROP procedure IF EXISTS GetPhoneNo;

DELIMITER //

CREATE procedure GetPhoneNo ( IN name_value VARCHAR(20) )
```

```
BEGIN

    SELECT Phone, Name from OFFICES WHERE Name = name_value ;

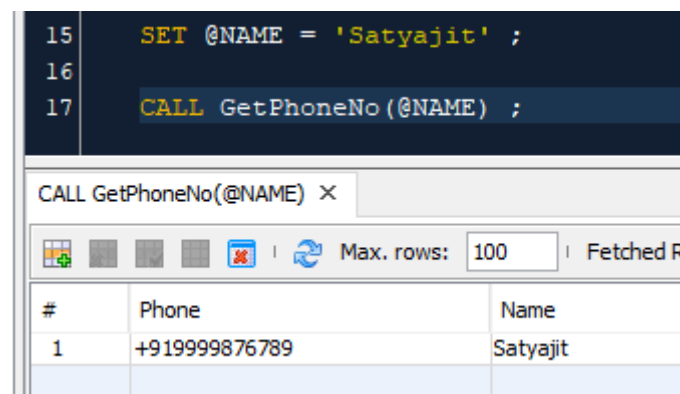
END; //
```

```
DELIMITER ;

SET @NAME = 'Satyajit' ;

CALL GetPhoneNo(@NAME) ;
```

b (ii) :



The screenshot shows a SQL query execution window with a dark background. The query being executed is:

```
15 SET @NAME = 'Satyajit' ;
16
17 CALL GetPhoneNo (@NAME) ;
```

Below the query, the window title is "CALL GetPhoneNo(@NAME) X". The output is displayed in a table with the following columns: #, Phone, and Name. The table contains one row of data.

#	Phone	Name
1	+919999876789	Satyajit

Figure 2 Output

```
DROP procedure IF EXISTS GetOffice ;

DELIMITER //
```

```
CREATE procedure GetOffice ( IN country_value VARCHAR(20) )

BEGIN

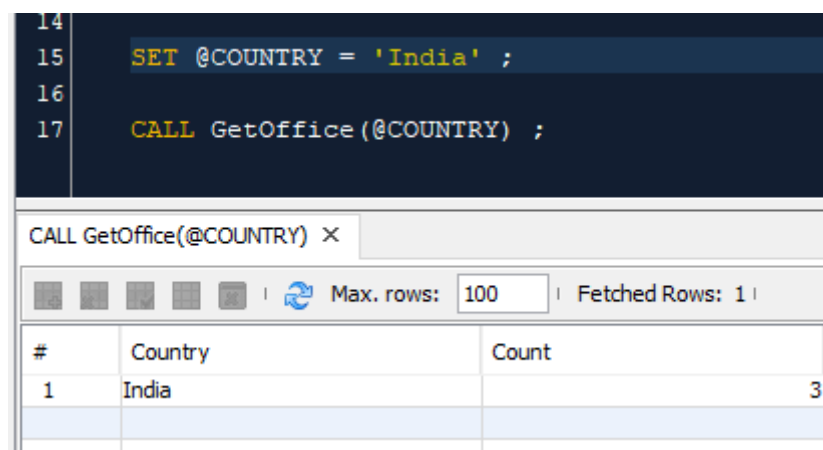
    SELECT Country, COUNT(*) as Count FROM OFFICES WHERE Country = country_value
;

END; //
```

```
DELIMITER ;

SET @COUNTRY = 'India' ;

CALL GetOffice(@COUNTRY) ;
```



```
14
15 SET @COUNTRY = 'India' ;
16
17 CALL GetOffice(@COUNTRY) ;
```

CALL GetOffice(@COUNTRY) X

Max. rows: 100 | Fetched Rows: 1 |

#	Country	Count
1	India	3

*Figure 3 Output*

## 6. Conclusions

A procedure is a subroutine (like a subprogram) in a regular scripting language, stored in a database. In the case of MySQL, procedures are written in MySQL and stored in the MySQL database/server. A MySQL procedure has a name, a parameter list, and SQL statement(s).

### IN parameters

IN is the default mode. When you define an IN parameter in a stored procedure, the calling program has to pass an argument to the stored procedure. In addition, the value of an IN parameter is protected. It means that even the value of the IN parameter is changed inside the stored procedure, its original value is retained after the stored procedure ends. In other words, the stored procedure only works on the copy of the IN parameter.

### OUT parameters

The value of an OUT parameter can be changed inside the stored procedure and its new value is passed back to the calling program. Notice that the stored procedure cannot access the initial value of the OUT parameter when it starts.

### INOUT parameters

An INOUT parameter is a combination of IN and OUT parameters. It means that the calling program may pass the argument, and the stored procedure can modify the INOUT parameter, and pass the new value back to the calling program.

## 7. Comments

### 1. Limitations of Experiments

- If you use many stored procedures, the memory usage of every connection will increase substantially.
- Besides, overusing a large number of logical operations in the stored procedures will increase the CPU usage because the MySQL is not well-designed for logical operations.

### 2. Limitations of Results

- It's difficult to debug stored procedures. Unfortunately, MySQL does not provide any facilities to debug stored procedures like other enterprise database products such as Oracle and SQL Server.

### 3. Learning happened

- We learnt how to use MySQL Procedures with different forms of parameters

### 3. Recommendations

- Developing and maintaining stored procedures often requires a specialized skill set that not all application developers possess. This may lead to problems in both application development and maintenance.