Government College of Engineering, Jalgaon		
(An Autonomous Institute of Government of Maharashtra)		
Name :	**Semester :** V	**PRN :**
Class : T. Y. B.Tech Computer	**Academic Year :** 2024-25	**Subject :** CO307U DBMS Lab
Course Teacher : Mr. Vinit Kakde		
Date of Performance :	**Date of Completion :**	

Practical no. 9

Aim: Creation of stored procedures, Execution of procedure and modification of procedures.

Theory:

PROCEDURES

Procedure (often called a stored procedure) is a collection of pre-compiled SQL statements stored inside the database. It is a subroutine or a subprogram in the regular computing language. A procedure always contains a name, parameter lists, and SQL statements. We can invoke the procedures by using triggers, other procedures and applications such as Java, Python, PHP, etc. It was first introduced in MySQL version 5. Presently, it can be supported by almost all relational database systems.

Creating a procedure:

The following syntax is used for creating a stored procedure in MySQL. It can return one or more value through parameters or sometimes may not return at all. By default, a procedure is associated with our current database. But we can also create it into another database from the current database by specifying the name as database_name.procedure_name.

```
DELIMITER &&
CREATE PROCEDURE procedure_name [[IN | OUT | INOUT] parameter_name datatype [,
parameter datatype])]
```

```
BEGIN
Declaration section
Executable section
END &&
DELIMITER;
```

Parameter Explanations

The procedure syntax has the following parameters:

Parameter Name	Descriptions
procedure_name	It represents the name of the stored procedure.
parameter	It represents the number of parameters. It can be one or more than one
Declaration_section	It represents the declarations of all variables
Executable_section	It represents the code for the function execution.

MySQL procedure parameter has one of three modes:

IN parameter

It is the default mode. It takes a parameter as input, such as an attribute. When we define it, the calling program has to pass an argument to the stored procedure. This parameter's value is always protected.

OUT parameters

It is used to pass a parameter as output. Its value can be changed inside the stored procedure, and the changed (new) value is passed back to the calling program. It is noted that a procedure cannot access the OUT parameter's initial value when it starts.

INOUT parameters

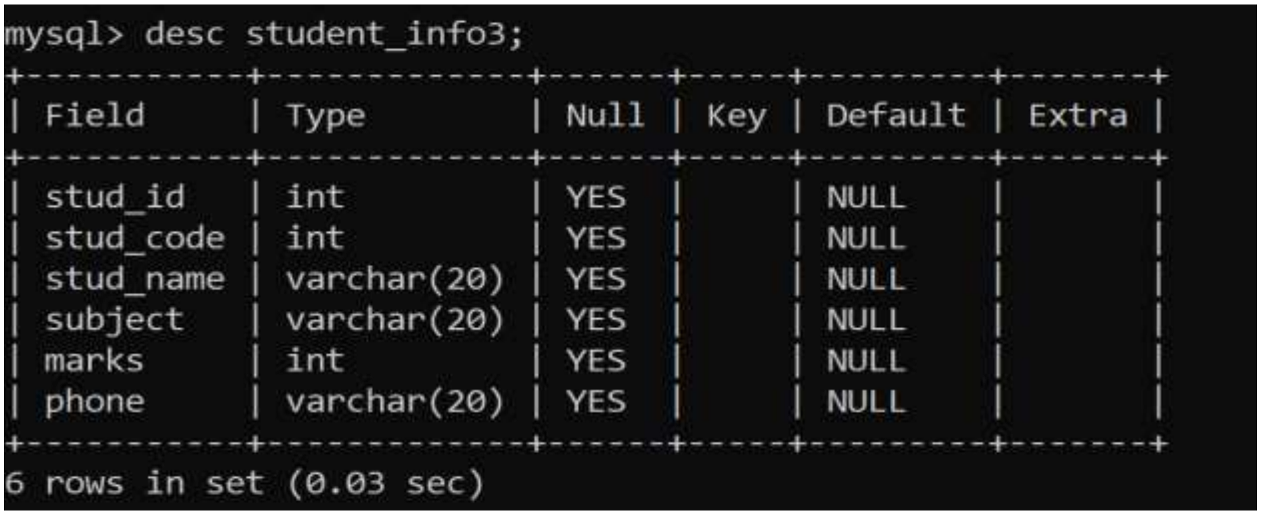
It is a combination of IN and OUT parameters. It means the calling program can pass the argument, and the procedure can modify the INOUT parameter, and then passes the new value back to the calling program.

How to call a stored procedure?
CALL procedure_name (parameter(s))

Queries and output:

1. **Syntax:**
Create table table_name(column1 datatype, column2 datatype,….. column-N datatype);

Query:
Create table student_info3(stud_id integer, stud_code integer, stud_name varchar(20), subject varchar(20), marks integer, phone varchar(20));

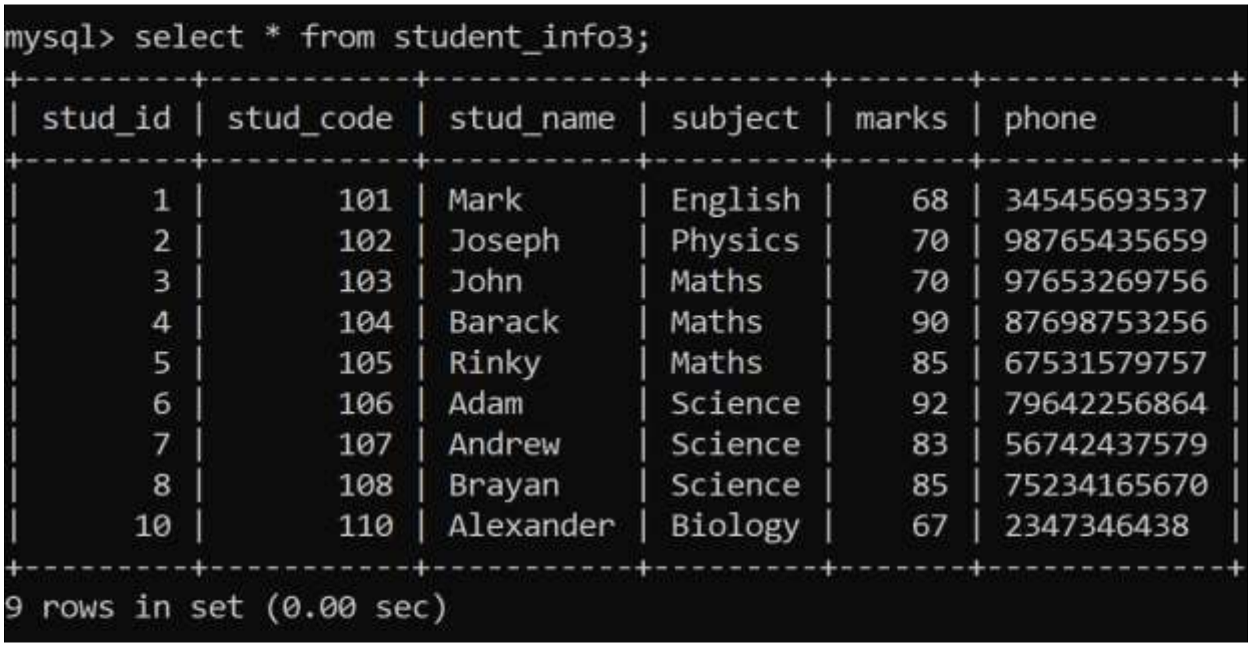


```
mysql> desc student_info3;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| stud_id   | int         | YES  |     | NULL    |       |
| stud_code | int         | YES  |     | NULL    |       |
| stud_name | varchar(20) | YES  |     | NULL    |       |
| subject   | varchar(20) | YES  |     | NULL    |       |
| marks     | int         | YES  |     | NULL    |       |
| phone     | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

2. **Syntax:**
insert into orders values(&o_id,&orderno,&p_id);

Query:
Insert into student_info3 values(1,101,'Mark','English',68,'34545693537');
Insert into student_info3 values(2,102,'Joseph','Physics',70,'98765435659');
Insert into student_info3 values(3,103,'John','Maths',70,'97653269756');
Insert into student_info3 values(4,104,'Barack','Maths',90,'87698753256');

Insert into student_info3 values(5,105,'Rinky','Maths',85,'67531579757');
Insert into student_info3 values(6,106,'Adam','Science',92,'79642256864');
Insert into student_info3 values(7,107,’Andrew’,Science’,83,’ '56742437579’);
Insert into student_info3 values(8,108,'Brayan','Science',85,'75234165670');
Insert into student_info3 values(10,110,’Alexander’,’Biology’,67,’2347346438’);



```
mysql> select * from student_info3;
+---------+-----------+-----------+---------+-------+-------------+
| stud_id | stud_code | stud_name | subject | marks | phone       |
+---------+-----------+-----------+---------+-------+-------------+
|       1 |       101 | Mark      | English |    68 | 34545693537 |
|       2 |       102 | Joseph    | Physics |    70 | 98765435659 |
|       3 |       103 | John      | Maths   |    70 | 97653269756 |
|       4 |       104 | Barack    | Maths   |    90 | 87698753256 |
|       5 |       105 | Rinky     | Maths   |    85 | 67531579757 |
|       6 |       106 | Adam      | Science |    92 | 79642256864 |
|       7 |       107 | Andrew    | Science |    83 | 56742437579 |
|       8 |       108 | Brayan    | Science |    85 | 75234165670 |
|      10 |       110 | Alexander | Biology |    67 | 2347346438  |
+---------+-----------+-----------+---------+-------+-------------+
9 rows in set (0.00 sec)
```

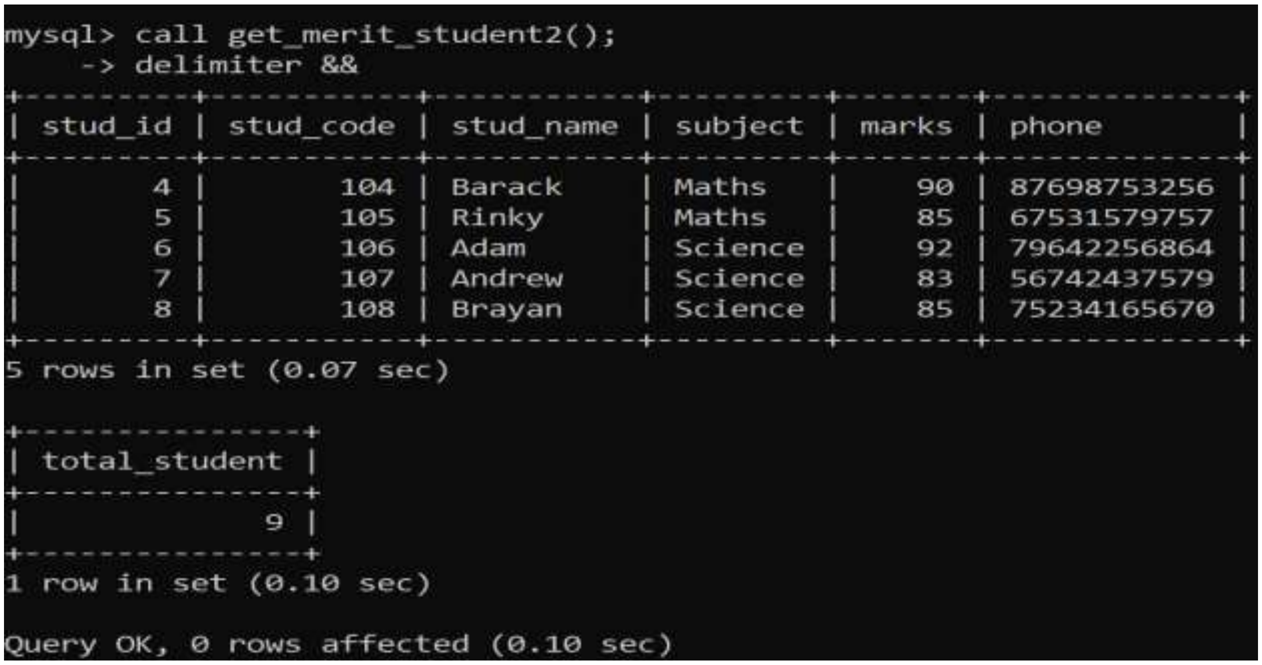
3. Procedure without Parameter

Syntax:

```
CREATE PROCEDURE ProcedureName()
BEGIN
   -- Procedure code
END;
```

Query:

```
mysql>delimiter &&
mysql> create procedure get_merit_student2()
        -> begin
        -> select * from student_info3 where marks > 70;
        -> select count(stud_code) as total_student from student_info3;
       -> end &&
```



```
mysql> call get_merit_student2();
    -> delimiter &&
+---------+-----------+-----------+---------+-------+-------------+
| stud_id | stud_code | stud_name | subject | marks | phone       |
+---------+-----------+-----------+---------+-------+-------------+
|       4 |       104 | Barack    | Maths   |    90 | 87698753256 |
|       5 |       105 | Rinky     | Maths   |    85 | 67531579757 |
|       6 |       106 | Adam      | Science |    92 | 79642256864 |
|       7 |       107 | Andrew    | Science |    83 | 56742437579 |
|       8 |       108 | Brayan    | Science |    85 | 75234165670 |
+---------+-----------+-----------+---------+-------+-------------+
5 rows in set (0.07 sec)

+---------------+
| total_student |
+---------------+
|             9 |
+---------------+
1 row in set (0.10 sec)

Query OK, 0 rows affected (0.10 sec)
```

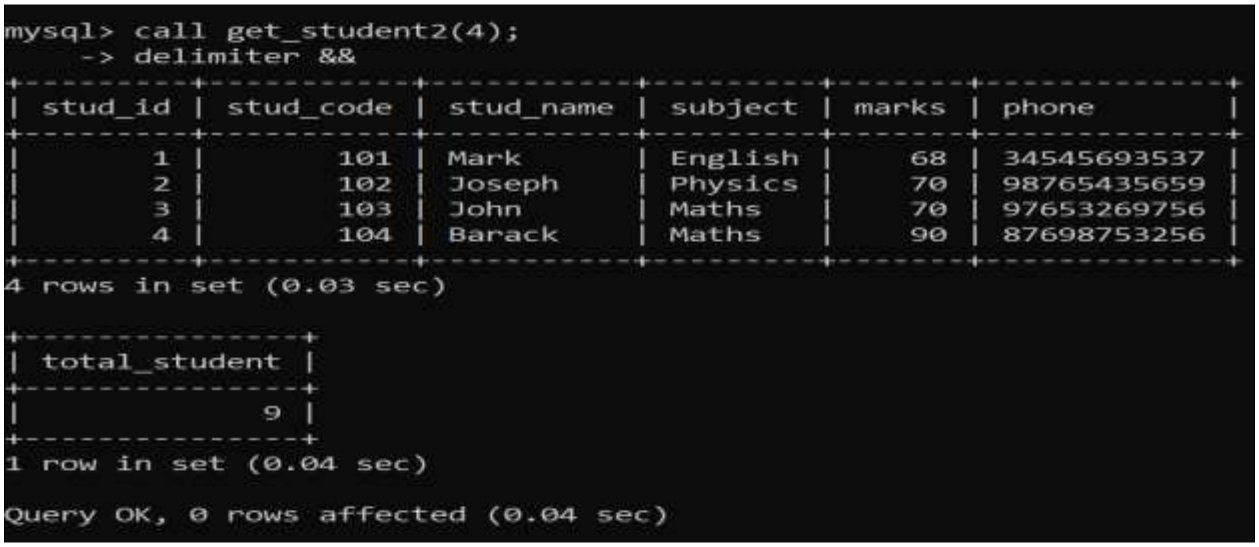
4. Procedure with IN Parameter

Syntax:

```
CREATE PROCEDURE ProcedureName(IN paramName DataType)
BEGIN
   -- Procedure code using paramName
END;
```

Query:

```
mysql> delimiter &&
mysql> create procedure get_student2(in var1 int)
       -> begin
       -> select * from student_info3 limit var1;
       -> select count(stud_code) as total_student from student_info3;
       -> end &&
```


```
mysql> call get_student2(4);
    -> delimiter &&
+---------+-----------+-----------+---------+-------+-------------+
| stud_id | stud_code | stud_name | subject | marks | phone       |
+---------+-----------+-----------+---------+-------+-------------+
|       1 |       101 | Mark      | English |    68 | 34545693537 |
|       2 |       102 | Joseph    | Physics |    70 | 98765435659 |
|       3 |       103 | John      | Maths   |    70 | 97653269756 |
|       4 |       104 | Barack    | Maths   |    90 | 87698753256 |
+---------+-----------+-----------+---------+-------+-------------+
4 rows in set (0.03 sec)

+---------------+
| total_student |
+---------------+
|             9 |
+---------------+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.04 sec)
```

5. Procedure with OUT Parameter

Syntax:

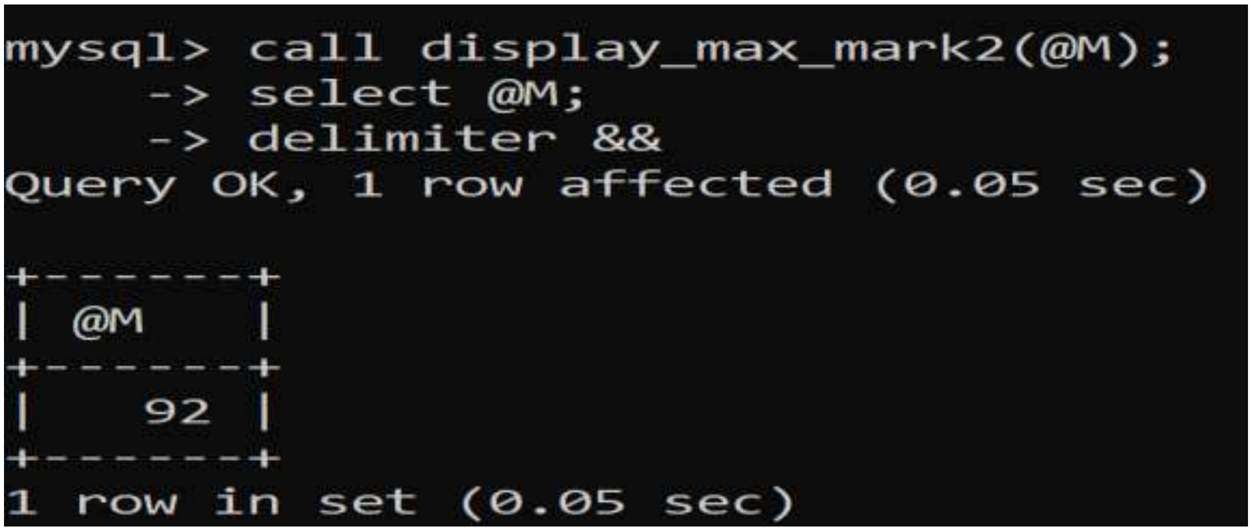
```
DELIMITER //

CREATE PROCEDURE GetTotalEmployees(OUT total_count INT)
BEGIN
   SELECT COUNT(*) INTO total_count
   FROM employees;
END //

DELIMITER ;
```

Query:

```
mysql> delimiter &&
mysql> create procedure display_max_mark2(out highestmark int)
       -> begin
       -> select max(marks) into highestmark from student_info3;
       -> end &&
```


```
mysql> call display_max_mark2(@M);
    -> select @M;
    -> delimiter &&
Query OK, 1 row affected (0.05 sec)

+------+
| @M   |
+------+
|   92 |
+------+
1 row in set (0.05 sec)
```

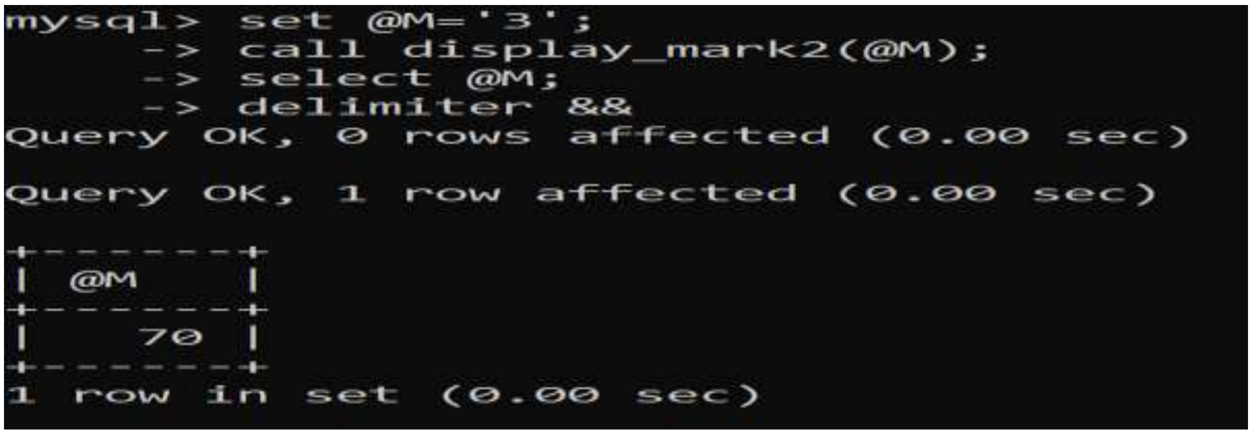
6. Procedure with INOUT Parameter

   Syntax:

   CREATE PROCEDURE ProcedureName(INOUT parameter_name DataType)
   BEGIN
      -- Procedure logic here
      -- You can read from and write to the INOUT parameter
   END;

   Query:

   mysql> delimiter &&
   mysql> create procedure display_mark2(inout var1 int)
         -> begin
         -> select marks into var1 from student_info3 where stud_id=var1;
         -> end &&



```
mysql> set @M='3';
    -> call display_mark2(@M);
    -> select @M;
    -> delimiter &&
Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+------+
| @M   |
+------+
|   70 |
+------+
1 row in set (0.00 sec)
```

Conclusion:

In conclusion, a procedure in SQL is a powerful tool for encapsulating and reusing business logic within a database. It allows you to define a set of operations that can be executed with a single call, promoting code modularity and reducing redundancy.

Questions and answers

1. What do you know about stored procedures?

Interviewers could ask this question at the beginning of your interview to gauge your basic understanding of stored procedures. You can provide a simple, cohesive answer that demonstrates your knowledge of and experience with the concept instead of a bland, textbook definition. If you can, use an example to demonstrate your use of the concept.

2. What are the different uses of stored procedures?

An interviewer may ask this question to determine if you know where to apply stored procedures. You can mention company-specific use cases and applications in your answer. You can also describe the features and benefits of stored procedures.

3. What types of stored procedures are in an SQL server?

Interviewers commonly ask this question to test your comprehension of the differences between the various types of stored procedures. You can demonstrate your understanding by listing the principal features of each type. Providing a brief explanation of each procedure can demonstrate your understanding of the concept to a potential employer.

4. What are the advantages of stored procedures?

Since stored procedures are easy to use, they have a variety of advantages. Briefly elaborate on each key benefit. This can ensure the interviewer that you are aware of the process's benefits and able to leverage them in your work.

5. What are the disadvantages of stored procedures?

Every process can have areas of opportunity or disadvantages. Interviewers may ask this question to test your awareness of the system's shortcomings. You can list and elaborate on the disadvantages of stored procedures from your experience.

Name & Sign of Course Teacher

Mr. Vinit Kakde