Stored Procedures in MySQL

Instructor: Sriya Ivaturi

What is a Stored Procedure?

➔ **Definition:**

- A stored procedure is a precompiled set of SQL statements stored in the database under a unique name.

➔ **Key Idea:**

- Execute complex or repetitive tasks with a **single** `CALL` statement.

➔ **Purpose:**

- **Automate routine operations**
- **Reduce code duplication**
- **Improve consistency and security**
- **Enhance performance**

Benefits Of Stored Procedures

- **Reusability:** Write once, use often.
- **Security:** Users can execute without seeing internal SQL.
- **Efficiency:** Reduces network traffic.
- **Maintainability:** Changes in one place.
- **Performance:** Compiled and cached by server and reduces network traffic. Complex operations run closer to the data, minimizing data transfer and improving speed.

Syntax of Stored Procedure

```
DELIMITER $$

CREATE PROCEDURE procedure_name([parameters])
BEGIN
  -- SQL statements
END $$

DELIMITER ;
```

Q) Why is the delimiter changed?

A: To avoid conflict with semicolons inside the procedure.

Basic Procedure

```
DELIMITER $$
CREATE PROCEDURE SayHello()
BEGIN
  SELECT 'Hello from MySQL!';
END $$
DELIMITER ;

CALL SayHello();
```

Q: What happens if we call SayHello before creating it?

A: MySQL will return: PROCEDURE SayHello does not exist.

Procedure with IN Parameter

```
DELIMITER $$
CREATE PROCEDURE GreetUser(IN userName VARCHAR(50))
BEGIN
  SELECT CONCAT('Hello, ', userName, '!') AS Greeting;
END $$
DELIMITER ;

CALL GreetUser('Ravi');
```

Q: What does the IN keyword do?

A: Allows passing a value into the procedure.

Procedure to add two numbers

```
DELIMITER $$

CREATE PROCEDURE AddTwoNumbers(IN a INT, IN b INT)
BEGIN
   SELECT a + b AS Sum;
END $$

DELIMITER ;

CALL AddTwoNumbers(10, 20);
```

Procedure with OUT Parameter

```
DELIMITER $$
CREATE PROCEDURE SquareNumber(IN input INT, OUT result INT)
BEGIN
   SET result = input * input;
END $$
DELIMITER ;

CALL SquareNumber(5, @out);
SELECT @out;
```

Q: What does OUT do?

Session Variables in MySQL

- **Session:**
  - Every time a client (like MySQL Workbench, PHP, Java app, etc.) connects to the MySQL server, it establishes a **connection**. A **session** is the period of time that a **single connection** remains open.
- **Session Variables in MySQL:**
  - Variables starting with `@` are **user-defined session variables**.
  - They:
    - Exist only while the session/connection is open.
    - Are private to that connection — other connections can't see them.
    - Disappear automatically when you disconnect.

A: Sends result out of the procedure to a session variable.

Procedure with INOUT Parameter

```
DELIMITER $$
CREATE PROCEDURE DoubleValue(INOUT num INT)
BEGIN
   SET num = num * 2;
END $$
DELIMITER ;

SET @val = 10;
CALL DoubleValue(@val);
SELECT @val;
```

Q: What does INOUT do?

A: It receives and modifies a value.

```sql
DELIMITER $$
CREATE PROCEDURE ProcessMarks(
    IN name VARCHAR(50),
    IN mark1 INT,
    IN mark2 INT,
    OUT total INT,
    OUT average DECIMAL(5,2))
BEGIN
  SET total = mark1 + mark2;
  SET average = total / 2;
END $$
DELIMITER ;

CALL ProcessMarks('Amit', 80, 90, @tot, @avg);
SELECT @tot, @avg;
```

Q: Can we have multiple OUTs?

Procedure Calling Another Procedure

```sql
DELIMITER $$
CREATE PROCEDURE OuterProcedure()
BEGIN
   CALL SayHello();
   CALL AddTwoNumbers(5, 7);
END $$
DELIMITER ;

CALL OuterProcedure();
```

Q: Can we nest procedures?

Q: What is the difference between a query and a procedure?

A: A query is written and executed every time, a procedure is stored and reused.

Summary

- → Procedures encapsulate SQL logic
- → Use IN, OUT, INOUT
- → Improve security and performance
- → Nesting supported

Practice

1. Greet user and show current date.

2. Swap two numbers using INOUT.

3. Calculate area and perimeter of rectangle.