<u>Procedures:</u>

A MySQL stored procedure is a set of pre-compiled SQL statements stored within the database, designed for execution upon request. These procedures encapsulate common business logic or database operations, allowing for reusability, improved performance, and enhanced security.

Creating a Stored Procedure:

The basic syntax for creating a stored procedure in MySQL involves using the **CREATE PROCEDURE** statement, defining a name, optionally specifying parameters, and enclosing the SQL statements within **BEGIN** and **END** blocks. A crucial step is to temporarily change the delimiter before and after the procedure definition, as MySQL uses a semicolon as a default statement terminator.

```
DELIMITER //

CREATE PROCEDURE procedure_name([IN|OUT|INOUT] parameter_name parameter_datatype, ...)

BEGIN

   -- SQL statements to be executed

   -- e.g., SELECT, INSERT, UPDATE, DELETE

END //

DELIMITER ;
```

Key Components:

DELIMITER // and DELIMITER ;: These statements change the default delimiter (semicolon) to allow the procedure's body to contain multiple SQL statements, each terminated by a semicolon, without prematurely ending the CREATE PROCEDURE statement.

CREATE PROCEDURE procedure_name(...): This initiates the creation of the procedure and assigns it a unique procedure_name.

Parameters: Procedures can accept parameters, which can be defined as **IN** (input), **OUT** (output), or **INOUT** (both input and output). Each parameter requires a name and a data type.

BEGIN ... END: This block encloses the SQL statements that constitute the procedure's logic.

Calling a Stored Procedure:

CALL procedure_name(parameter_value1, parameter_value2, ...);

Example:

```
CREATE PROCEDURE GetCustomerInfo(IN customerAge INT)
```

```
BEGIN

   SELECT * FROM CUSTOMERS WHERE AGE > customerAge;

END


CALL GetCustomerInfo(25);
```