MySQL - PREPARE Statement

A prepared statement in MySQL represents a precompiled statement. A statement is compiled and stored in a prepared statement and you can later execute this multiple times. Instead of values we pass place holders to this statement.

If you want to execute several identical queries (that differ by values only). You can use prepared statements. You can execute these statements in client libraries as well as in SQL scripts.

A SQL prepared statement is based on three statements namely –

- PREPARE
- EXECUTE
- DEALLOCATE PREPARE

The PREPARE statement/command among the above three prepares the statement. Once you prepare it you need to set values for the place holders using the SET statement.

Syntax

Following is the syntax of the PREPARE statement –

```
PREPARE stmt_name FROM preparable_stmt
```

Where stmt_name is the name of the prepared statement which you can refer later.

Example

Suppose we have created a table named **Employee** in the database using the CREATE statement and inserted three records in it as shown below –

```
mysql> CREATE TABLE Employee(
       Name VARCHAR(255),
       Salary INT,
       Location VARCHAR(255));
```

Following statements inserts records in the above created table –

```
mysql> INSERT INTO Employee VALUES ('Amit', 6554, 'Hyderabad');
mysql> INSERT INTO Employee VALUES ('Sumith', 5981, 'Vishakhapatnam');
mysql> INSERT INTO Employee VALUES ('Sudha', 7887, 'Vijayawada');
```

If you observe the above statements except the values all the insert statements are identical. You can prepare a statement with place holders instead of values as –

```
mysql> PREPARE prepared_stmt FROM 'INSERT INTO EMPLOYE VALUES (?, ?, ?)';
Query OK, 0 rows affected (0.02 sec)
Statement prepared
```

Once you prepare the statement you need to set values for the place holders as follows –

```
mysql> SET @name = 'Raghu';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @sal = 9878;
Query OK, 0 rows affected (0.00 sec)

mysql> SET @loc = 'Delhi';
Query OK, 0 rows affected (0.00 sec)
```

You can execute the above prepared statement using the EXECUTE statement –

```
mysql> EXECUTE prepared_stmt USING @name, @sal, @loc;
Query OK, 1 row affected (0.19 sec)
```

Verification

After executing the prepared statement if you verify the contents of the employee table you can observe the newly inserted row –

```
mysql> SELECT * FROM EMPLOYEE;
+--------+--------+----------------+
| Name   | Salary | Location       |
+--------+--------+----------------+
| Amit   | 6554   | Hyderabad      |
| Sumith | 5981   | Vishakhapatnam |
| Sudha  | 7887   | Vijayawada     |
| Raghu  | 9878   | Delhi          |
+--------+--------+----------------+
4 rows in set (0.09 sec)
```

Example

Following is a prepared statement with the SELECT query –

```
--Preparing the statement
mysql> PREPARE prepared_stmt FROM 'SELECT ? FROM EMPLOYE where Name = ?';
Query OK, 0 rows affected (0.00 sec)
Statement prepared

--Setting the values
mysql> SET @col = 'Salary';
Query OK, 0 rows affected (0.00 sec)

mysql> SET @name = 'Raghu';
Query OK, 0 rows affected (0.00 sec)
--Executing the statement
mysql> EXECUTE prepared_stmt USING @col, @name;
+--------+
| ?      |
+--------+
| Salary |
+--------+
1 row in set (0.11 sec)
```

Example

Assume we have created another table and populated it using the following queries –

```
mysql> Create table Student(Name Varchar(35), age INT, Score INT);
Query OK, 0 rows affected (1.28 sec)
mysql> INSERT INTO student values ('Jeevan', 22, 8);
mysql> INSERT INTO student values ('Raghav', 26, -3);
mysql> INSERT INTO student values ('Khaleel', 21, -9);
mysql> INSERT INTO student values ('Deva', 30, 9);
```

You can choose the table to execute a query dynamically using this statement as shown below –

```
--Setting the table name dynamically
mysql> SET @table = 'Student';
Query OK, 0 rows affected (0.00 sec)
mysql> SET @statement = CONCAT('SELECT * FROM ', @table);
Query OK, 0 rows affected (0.08 sec)

--Preparing the statement
mysql> PREPARE prepared_stmt FROM @statement;
Query OK, 0 rows affected (0.16 sec)
Statement prepared
```

```
--Executing the statement
mysql> EXECUTE prepared_stmt;
+---------+------+-------+
| Name    | age  | Score |
+---------+------+-------+
| Jeevan  | 22   | 8     |
| Raghav  | 26   | -3    |
| Khaleel | 21   | -9    |
| Deva    | 30   | 9     |
+---------+------+-------+
4 rows in set (0.10 sec)
```