**User-Defined Functions in MySQL with Examples**

Here, I am going to discuss How to Create and Call User-Defined Functions in MySQL with examples. you will understand what is a user-defined function in MySQL and how to create and use MySQL User-Defined functions with examples.

**What is a function in MySQL?**

A function in MySQL is a subprogram that is used to perform an action such as complex calculations and returns the result of the action as a value. There are two types of functions available in MySQL. They as follows:

**1. System Defined Function**

**2. User-Defined Function**

The functions which are already defined or predefined by MySQL and ready to be used by the developer are called as system-defined function whereas if the function is defined by the developer, then such functions are called as a user-defined function.

Some functions take parameters; do some processing and returning some results back. For example, SELECT SQRT(16); will return the square root of 16 i.e. 4.

Some functions do not take any parameters but return some results back. For example, SELECT NOW(); will return the current date-time like 2021-07-09 07:11:07.

So, we can conclude that a function may have a parameter(s) that is optional but a function should always return some value that is mandatory.

Note: We already discussed System Defined Functions in our previous articles. Here, in this article, we will keep the focus on User Defined Functions in MySQL.

**MySQL User Defined Functions**

The function which is defined by the user is called a user-defined function. MySQL user-defined functions may or may not have parameters that are optional, but it always returns a single value that is mandatory. The returned value which is return by the MySQL Function can be of any valid MySQL data type.

**How to Create User-Defined Functions in MySQL?**

The following is the basic syntax to create MySQL User-defined function.

How to Create User-Defined Functions in MySQL?



First, we need to specify the name of the user-defined function that you want to create after the CREATE FUNCTION statement.

Second, list all the input parameters of the user-defined function inside the parentheses followed by the function name. By default, all the parameters are IN parameters. The point that you need to remember is, you cannot specify IN, OUT or INOUT modifiers to the parameters in MySQL.

Third, specify the data type of the return value in the RETURNS statement, which can be any valid MySQL data type.

Fourth, specify if the function is deterministic or not using the DETERMINISTIC keyword. It is optional. If we don’t specify DETERMINISTIC or NOT DETERMINISTIC, by default. MySQL uses the NOT DETERMINISTIC option. A deterministic function in MySQL always returns the same result for the same input parameters whereas a non-deterministic function returns different results for the same input parameters. We will discuss this concept in detail in our upcoming article.

Fifth, write the code in the body of the user-defined function within the BEGIN & END block. Inside the body section, you need to specify at least one RETURN statement. The RETURN statement returns a value to the calling programs. Whenever the RETURN statement is reached, the execution of the stored function is terminated immediately.

The syntax for calling a User Defined Function in MySQL is as follows:

SELECT <Function\_Name>(Value);

Let us discuss some MySQL User-defined Function examples for better understanding.

**Example: Create a Function in MySQL which should return the cube of a given value.**

Let us first create a database called FunctionDB by executing the below SQL Statement as well as use the database for further queries.

**CREATE DATABASE FunctionDB ;**

**USE FunctionDB ;**

Now execute the following SQL statements which will create a user-defined function that takes an input parameter of type INT and return its cube which is also INT type. So, we specify the input parameter type as INT as well as the return value data type as also INT.

**DELIMITER $$**

**CREATE FUNCTION Func\_Cube**

**(**

**Num INT**

**)**

**RETURNS INT**

**DETERMINISTIC**

**BEGIN**

**DECLARE TotalCube INT;**

**SET TotalCube = Num \* Num \* Num;**

**RETURN TotalCube;**

**END$$**

**DELIMITER ;**

A function is a database object in MySQL. So, once the function is created, you can view it in MySQL Workbench under the Functions section as shown in the below image.

**User-Defined Functions in MySQL with Examples**

Or you can also view all the user-defined functions in the current FunctionDB database by executing the SHOW FUNCTION STATUS statement as follows:

**SHOW FUNCTION STATUS WHERE db = ‘FunctionDB’;**

Once you execute the above query, you will get the following output. Here, you will see the list of user-defined functions created in the FunctionDB database.

**SHOW FUNCTION STATUS Statement in MySQL**

**How to Call a User Defined Function in MySQL?**

You can call a stored function within a select statement or from a stored procedure. In this article, I will show you how to call the stored function from a select statement, and in our upcoming article when we discuss the stored procedure, we will see how to call a stored function from a stored procedure. To call our Func\_Cube stored function from select statement, execute like below.

**SELECT Func\_Cube(5);**

Output: 125

**Example: Create a User-defined Function in MySQL to calculate the age.**

Let us create a user-defined stored function that will calculate and returns the age of an employee. To compute the age, we require the date of birth. So, this function will take the date of birth as an input parameter and then calculate the age and return the age. So, the return type is int.

**DELIMITER $$**

**CREATE FUNCTION Func\_Calculate\_Age**

**(**

**Age date**

**)**

**RETURNS INT DETERMINISTIC**

**BEGIN**

**DECLARE TodayDate DATE;**

**SELECT CURRENT\_DATE() INTO TodayDate;**

**RETURN YEAR(TodayDate) - YEAR(Age);**

**END$$**

**DELIMITER ;**

**Calling the above MySQL function:**

**SELECT Func\_Calculate\_Age(‘1988-02-29’);**

**SELECT Func\_Calculate\_Age(‘1988-02-29’) AS AGE;**

**Real-Time Example of User-Defined Function in MySQL with Examples:**

We are going to use the following Employee table to understand how to use the user-defined functions in MySQL with real-time scenarios.

**How to Call a User Defined Function in MySQL?**

Please execute the following SQL script to create and populate the Employee with the required sample data.

**-- Create Employee Table**

**CREATE TABLE Employee**

**(**

**EmployeeId INT PRIMARY KEY,**

**Name VARCHAR(50),**

**Salary INT,**

**DOB Date**

**);**

-- Populate Employee table

**INSERT INTO Employee(EmployeeId, Name, Salary, DOB) VALUES(1001, 'Pranaya', 10000, '1988-02-29');**

**INSERT INTO Employee(EmployeeId, Name, Salary, DOB) VALUES(1002, 'Anurag', 20000, '1992-06-22');**

**INSERT INTO Employee(EmployeeId, Name, Salary, DOB) VALUES(1003, 'Sambit', 30000, '1978-04-12');**

Now, we need to fetch all the Employee’s data from the Employee table along with the Employee Age. As in the Employees table, we store only the DOB, so we will call the Func\_Calculate\_Age function in the select clause bypassing the employee dob as an input parameter as shown in the below query.

**SELECT EmployeeId, Name, Salary, DOB, Func\_Calculate\_Age(DOB) AS Age FROM Employee;**

When you execute the above query, you will get the following output and please observe the Age column in the output.



**Real-Time Example of User-Defined Function in MySQL with Examples**

Let’s see how to use the user-defined stored function Func\_Calculate\_Age in the where clause of an SQL Query. The following SQL query will find out the details of those employees whose age is greater than 30.

**SELECT EmployeeId, Name, Salary, DOB, Func\_Calculate\_Age(DOB) AS Age**

**FROM Employee**

**WHERE Func\_Calculate\_Age(DOB) > 30;**

When you execute the above query, you will get the following output.

