MySQL | Creating stored function

The **CREATE FUNCTION** statement is used for creating a stored function and user-defined functions. A stored function is a set of SQL statements that perform some operation and return a single value.

Just like Mysql in-built function, it can be called from within a Mysql statement.

By default, the stored function is associated with the default database.

The **CREATE FUNCTION** statement require **CREATE ROUTINE** database privilege.

Syntax:
The syntax for **CREATE FUNCTION** statement in Mysql is:

```
CREATE FUNCTION function_name(func_parameter1, func_parameter2, ..)
          RETURN datatype [characteristics]
          func_body
```

Parameters used:

1. **function_name:**
   It is the name by which stored function is called. The name should not be same as native(built_in) function. In order to associate routine explicitly with a specific database function name should be given as *database_name.func_name*.
2. **func_parameter:**
   It is the argument whose value is used by the function inside its body. You can't specify to these parameters **IN, OUT, INOUT**. The parameter declaration inside parenthesis is provided as *func_parameter type*. Here, type represents a valid Mysql datatype.
3. **datatype:**
   It is datatype of value returned by function.
4. **characteristics:**
   The CREATE FUNCTION statement is accepted only if at least one of the characteristics { DETERMINISTIC, NO SQL, or READS SQL DATA } is specified in its declaration.

func_body is the set of Mysql statements that perform operation. It's structure is as follows:

```
BEGIN

        Mysql Statements

        RETURN expression;
END
```

The function body must contain one RETURN statement.

Example:

Consider following Employee Table-

emp_id	fname	lname	start_date
1	Michael	Smith	2001-06-22
2	Susan	Barker	2002-09-12
3	Robert	Tvler	2000-02-09
4	Susan	Hawthorne	2002-04-24

We have to find the number of years the employee has been in the company-

```
DELIMITER //

CREATE FUNCTION no_of_years(date1 date) RETURNS int DETERMINISTIC
BEGIN
 DECLARE date2 DATE;
  Select current_date()into date2;
  RETURN year(date2)-year(date1);
END

//

DELIMITER ;
```

Calling of above function:

```
Select emp_id, fname, lname, no_of_years(start_date) as 'years' from employee;
```

Output:

emp_id	fname	lname	years
1	Michael	Smith	18
2	Susan	Barker	17
3	Robert	Tvler	19
4	Susan	Hawthorne	17

One more example:

```
DELIMITER $$
CREATE FUNCTION get_designation_name(d_id INT) RETURNS VARCHAR(
20 )
BEGIN
DECLARE de_name VARCHAR( 20 ) DEFAULT "";
SELECT name INTO de_name FROM designation WHERE id = d_id;
RETURN de_name;
END $$
DELIMITER $$
```

Calling function in SQL Query:

SELECT id, **get_designation1(`d_id`)** as DESIGNATION, name FROM 'staff;

MySQL STORED PROCEDURES Vs FUNCTIONS

MySQL provides 2 ways to create methods or code to be re-used in the form of FUNCTIONS and PROCEDURES.

However, there are certain differences between both of them:

PROCEDURE	FUNCTION
Supports different type of parameters like IN, OUT and INOUT.	Supports only input parameters.
They can call functions.	Functions cannot call procedures.
Exceptions can be handled in procedures.	No exception handling possible in FUNCTIONS.
Might or might not return a value.	A FUNCTION is expected to return a result always.
These cannot be called from within SELECT statements.	Functions can be called from within SELECT statement.
They are mainly used to process repeatable tasks.	FUNCTIONS are used to compute values and return results to the caller.
These are pre-compiled - i.e. they are compiled once and the compiled code is reused for subsequent calls being made to the procedure.	FUNCTIONS are compiled every time when they are called.