

MySQL User-defined Functions (LAB Lecture)



Department of Information Technology
Database Systems II (IT216)
Spring 2025-2026
Week 9 – March 31, 2026
Lecturer: Soma Soleimanzadeh



1

Contents

- User-Defined Functions
- Syntax of Creating a Function in MySQL
- Variables
- Delimiter

2

Syntax of Creating Function in MySQL

DELIMITER //

```
CREATE FUNCTION function_name (input_parameter1 datatype1,  
                                input_parameter2 datatype2,  
                                ... )
```

RETURNS return_datatype

BEGIN

<Declaration Part of Function>

<Execution Part of Function>

RETURN return_value;

END//

DELIMITER ;

3

Let's Create the Database and Tables

- Create a database named **StockItems** and activate it.
- Create the following table and enter the data inside the table.

Item

<u>ItemID</u>	ItemName	Price	StockQuantity
1	Calculator	20	31
2	Punching Machine	15	25
3	Scissors	5	100

4

Let's Create the Database and Tables

```
create database StockItems;
use StockItems;

create table Item
(ItemID int auto_increment,
ItemName varchar(100),
Price int,
StockQuantity int,
primary key (ItemID));

insert into Item(ItemName,Price,StockQuantity)
values ('Calculator',20,31),('Punching Machine',15,25),('Scissors',5,100);
```

5

Example #1

- Create a function named **averagePrice** to calculate and return the average price of all items in the **item** table.

Without Using Local Variable

```
DELIMITER %%
CREATE FUNCTION averagePrice ()
RETURNS DECIMAL(5,2) DETERMINISTIC
BEGIN
    RETURN (SELECT avg(price) from Item);
END %%
DELIMITER ;
```

Using Local Variable

```
DELIMITER %%
CREATE FUNCTION averagePrice ()
RETURNS DECIMAL(5,2) DETERMINISTIC
BEGIN
    DECLARE avgprice DECIMAL(5,2);
    SELECT avg(price) INTO avgprice FROM Item;
    RETURN avgprice;
END %%
DELIMITER ;
```

6

Now Invoke(Call) the Function!

```
DELIMITER %%  
CREATE FUNCTION averagePrice ( )  
RETURNS DECIMAL(5,2) DETERMINISTIC  
BEGIN  
    DECLARE avgprice DECIMAL(5,2);  
    SELECT avg(price) INTO avgprice FROM Item;  
    RETURN avgprice;  
END %%  
DELIMITER ;
```

Step 1 – Creating the Function

```
SELECT averagePrice( );
```

Step 2 – Calling (Invoking) the Function

You can call the function in two ways:

- Calling the function without storing the return value in a variable.
- Calling the function and storing its return value in a user_defined variable.

```
SET @result = 0;  
SELECT averagePrice( ) INTO @result;  
SELECT @result; - - - - -> Output: 13.33
```

Example #2

- Create a function named **TotalQuantity** to calculate and return the total quantity of all items in the **item** table.
 - Use a **local variable** in the function.

```
DELIMITER $$  
CREATE FUNCTION TotalQuantity ( )  
RETURNS INT DETERMINISTIC  
BEGIN  
    DECLARE total INT;  
    SELECT sum(StockQuantity) INTO total FROM Item;  
    RETURN total;  
END $$  
DELIMITER ;
```

Item

ItemID	ItemName	Price	StockQuantity
1	Calculator	20	31
2	Punching Machine	15	25
3	Scissors	5	100

Step 1 – Creating the Function

```
SET @result = 0;  
SELECT TotalQuantity( ) INTO @result;  
SELECT @result; - - - - -> Output: 156
```

Step 2 – Calling the Function

Lab Work!

- Create a function named `getPrice` to take `ID` an item and return its price.

```
DELIMITER $$  
CREATE FUNCTION getPrice ( ID int)  
RETURNS INT DETERMINISTIC  
BEGIN  
    DECLARE PriceVar INT;  
    SELECT price INTO PriceVar FROM Item WHERE ItemID = ID;  
    RETURN priceVar;  
END $$  
DELIMITER ;
```

Item

ItemID	ItemName	Price	StockQuantity
1	Calculator	20	31
2	Punching Machine	15	25
3	Scissors	5	100

```
SET @result = 0;  
SELECT getPrice(3) INTO @result;  
SELECT @result;
```

Step 1 – Creating the Function

Step 2 – Calling the Function

Output: 5