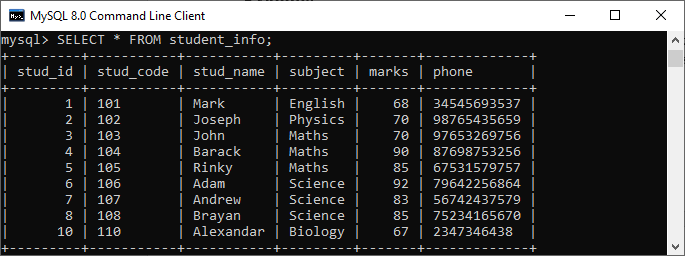
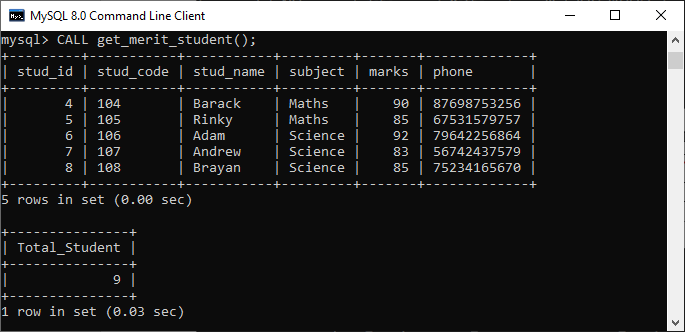
**Procedures**

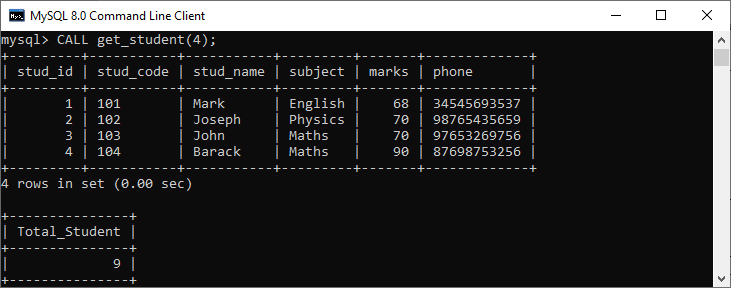
1. DELIMITER &&
2. **CREATE** **PROCEDURE** procedure\_name [[IN | **OUT** | INOUT] parameter\_name datatype [, parameter datatype]) ]
3. **BEGIN**
4. Declaration\_section
5. Executable\_section
6. **END** &&
7. DELIMITER ;
8. CALL procedure\_name ( parameter(s))
   * Use database name;
   * Table student\_info



1. DELIMITER &&
2. **CREATE** **PROCEDURE** get\_merit\_student ()
3. **BEGIN**
4. **SELECT** \* **FROM** student\_info **WHERE** marks > 70;
5. **SELECT** COUNT(stud\_code) **AS** Total\_Student **FROM** student\_info;
6. **END** &&
7. DELIMITER ;
8. mysql> CALL get\_merit\_student();



1. DELIMITER &&
2. **CREATE** **PROCEDURE** get\_student (IN var1 **INT**)
3. **BEGIN**
4. **SELECT** \* **FROM** student\_info LIMIT var1;
5. **SELECT** COUNT(stud\_code) **AS** Total\_Student **FROM** student\_info;
6. **END** &&
7. DELIMITER ;
8. mysql> CALL get\_student(4);



1. mysql> **DROP** **PROCEDURE** display\_marks;

Creating a function

1. **CREATE** **FUNCTION** function\_name [ (parameter datatype [, parameter datatype]) ]
2. **RETURNS** return\_datatype
3. **BEGIN**
4. Declaration\_section
5. Executable\_section
6. **END**;

**Function\_name:** name of the function

**Parameter:** number of parameter. It can be one or more than one.

**return\_datatype:** return value datatype of the function

**declaration\_section:** all variables are declared.

**executable\_section:** code for the function is written here.

**Database:** employee

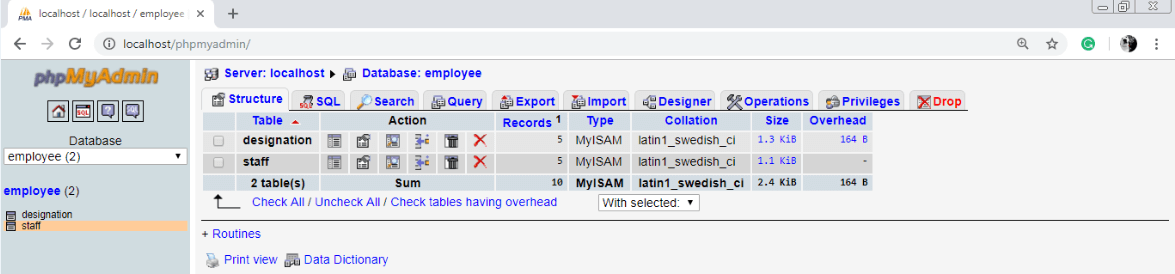


Table 1 : designation

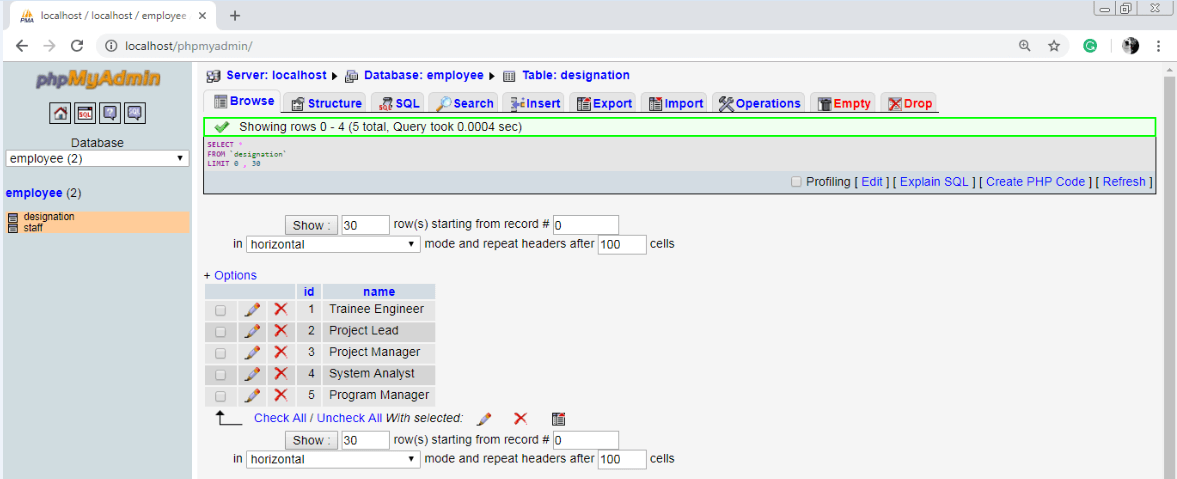
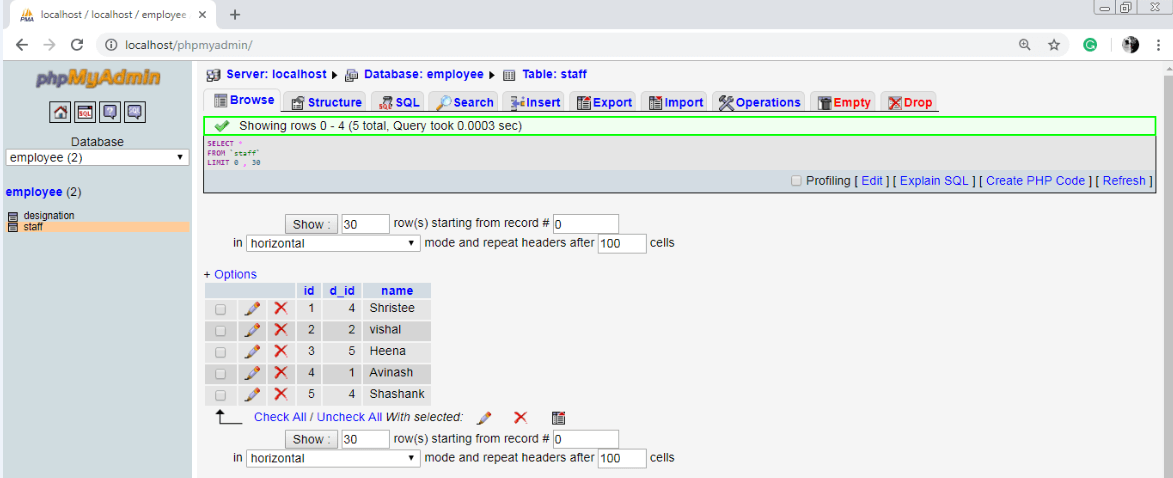
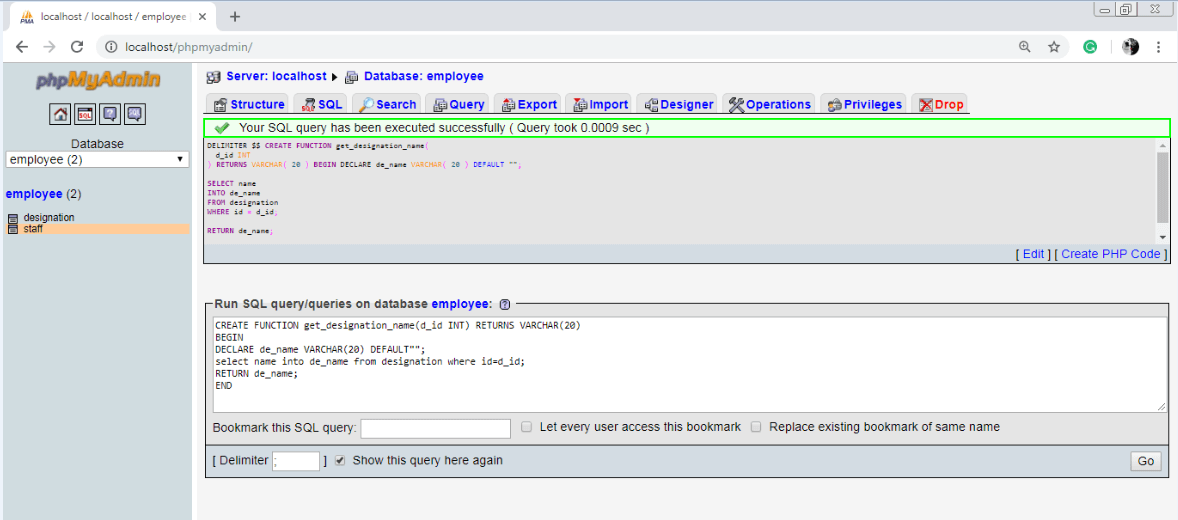


Table 2 : staff

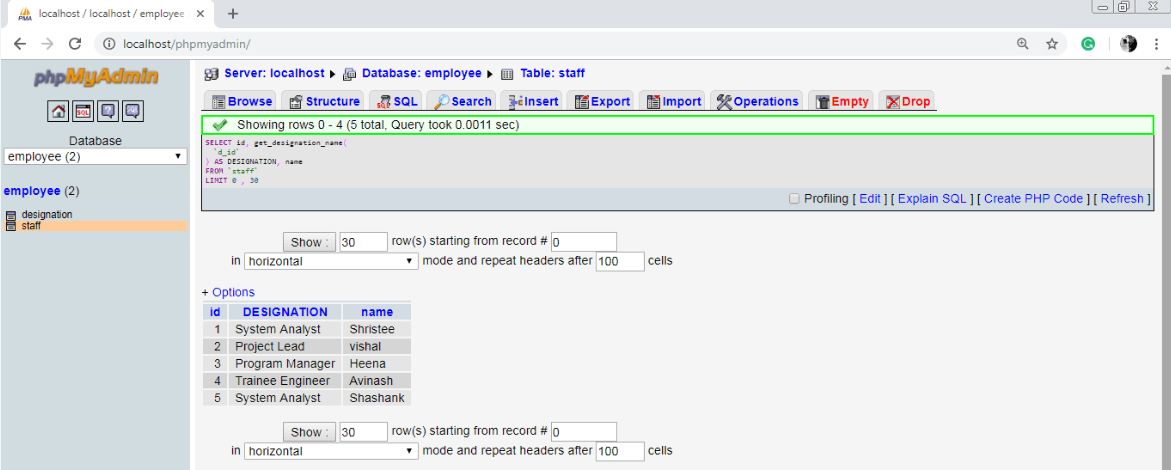


1. DELIMITER $$
2. **CREATE** **FUNCTION** get\_designation\_name(d\_id **INT**) **RETURNS** **VARCHAR**( 20 )
3. **BEGIN**
4. **DECLARE** de\_name **VARCHAR**( 20 ) **DEFAULT** "";
5. **SELECT** **name** **INTO** de\_name **FROM** designation **WHERE** id = d\_id;
6. **RETURN** de\_name;
7. **END** $$



**Query :**

SELECT id, get\_designation1(`d\_id`) as DESIGNATION, name FROM 'staff'



Drop a function

In MySQL Function can also be dropped. When A function id dropped, it is removed from the database.

Syntax:

1. **Drop** **function** [ IF EXISTS ] function\_name;

**Triggers**

Introduction to MySQL triggers

BEFORE INSERT – activated before data is inserted into the table.

AFTER INSERT – activated after data is inserted into the table.

BEFORE UPDATE – activated before data in the table is updated.

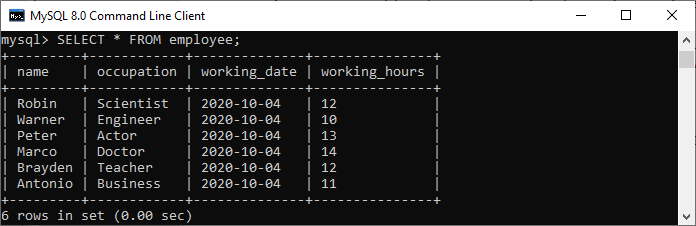
AFTER UPDATE – activated after data in the table is updated.

BEFORE DELETE – activated before data is removed from the table.

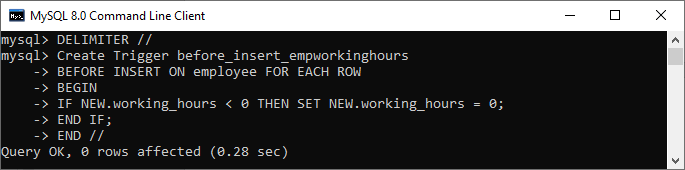
**Trigger Activation Time:** BEFORE | AFTER

**Trigger Event:** INSERT | UPDATE | DELETE

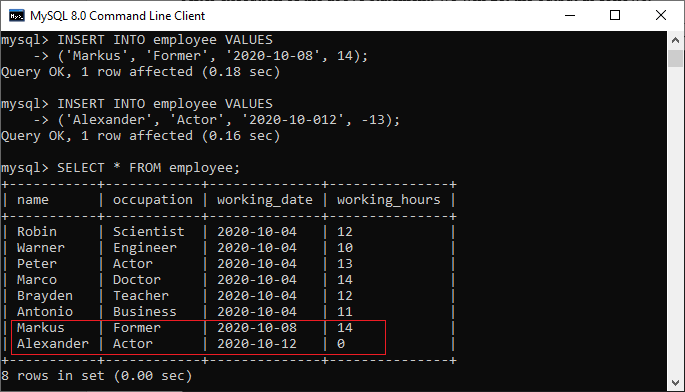
1. **CREATE** **TRIGGER** trigger\_name
2. (**AFTER** | BEFORE) (**INSERT** | **UPDATE** | **DELETE**)
3. **ON** table\_name **FOR** EACH ROW
4. **BEGIN**
5. --variable declarations
6. --trigger code
7. **END**;
8. **CREATE** **TABLE** employee(
9. **name** **varchar**(45) NOT NULL,
10. occupation **varchar**(35) NOT NULL,
11. working\_date **date**,
12. working\_hours **varchar**(10)
13. );
14. **INSERT** **INTO** employee **VALUES**
15. ('Robin', 'Scientist', '2020-10-04', 12),
16. ('Warner', 'Engineer', '2020-10-04', 10),
17. ('Peter', 'Actor', '2020-10-04', 13),
18. ('Marco', 'Doctor', '2020-10-04', 14),
19. ('Brayden', 'Teacher', '2020-10-04', 12),
20. ('Antonio', 'Business', '2020-10-04', 11);



1. mysql> DELIMITER //
2. mysql> **Create** **Trigger** before\_insert\_empworkinghours
3. BEFORE **INSERT** **ON** employee **FOR** EACH ROW
4. **BEGIN**
5. IF NEW.working\_hours < 0 **THEN** **SET** NEW.working\_hours = 0;
6. **END** IF;
7. **END** //



1. mysql> **INSERT** **INTO** employee **VALUES**
2. ('Markus', 'Former', '2020-10-08', 14);
4. mysql> **INSERT** **INTO** employee **VALUES**
5. ('Alexander', 'Actor', '2020-10-012', -13);



1. mysql>SHOW TRIGGERS;