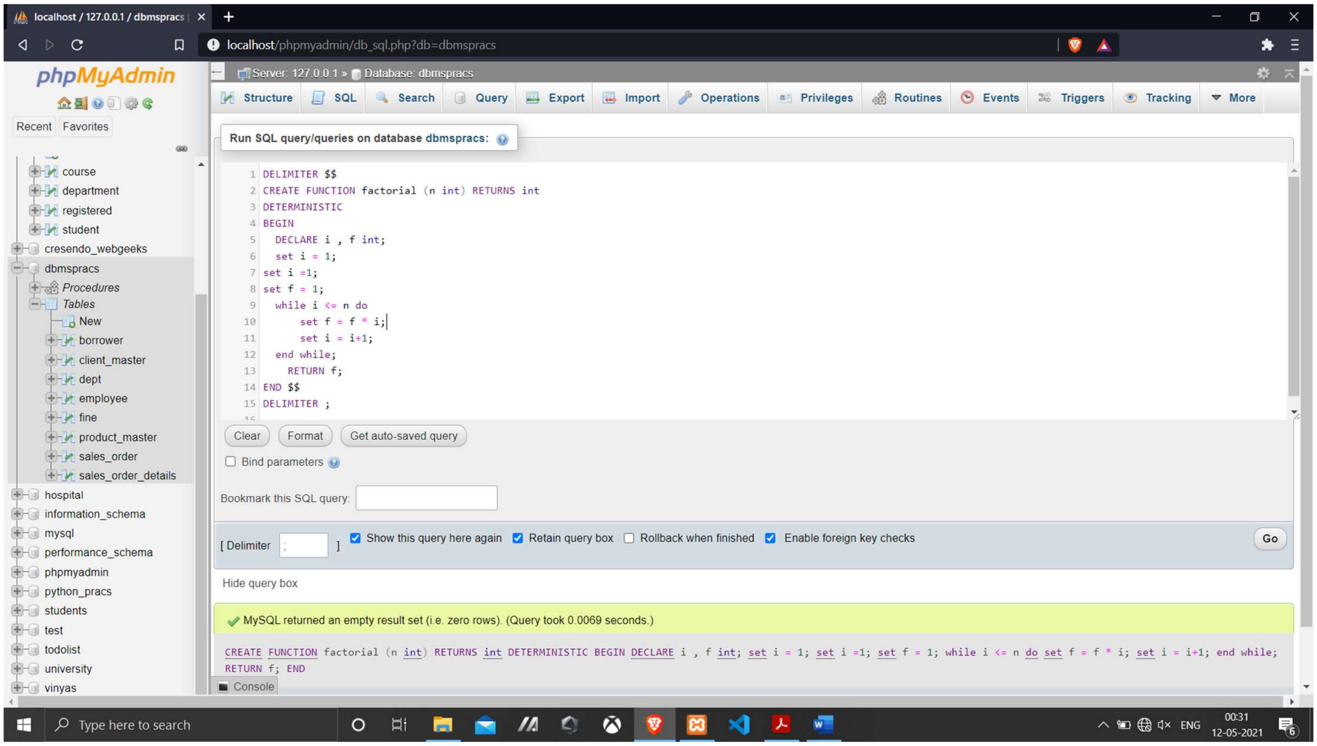
Name: Brendan Lucas, Div: SE COMP B, Roll No: 8953

DBMS Practical Implementation, Lab 9.

1. Write a function to find factorial of a number.

```
DELIMITER $$
CREATE FUNCTION factorial (n int) RETURNS int
DETERMINISTIC
BEGIN
  DECLARE i , f int;
set i =1;
set f = 1;
  while i <= n do
      set f = f * i;
      set i = i+1;
  end while;
         RETURN f;
END$$
DELIMITER;
```



Run SQL query/queries on database dbmspracs:

```
DELIMITER $$
CREATE FUNCTION factorial (n int) RETURNS int
DETERMINISTIC
BEGIN
  DECLARE i , f int;
  set i = 1;
set i =1;
set f = 1;
  while i <= n do
      set f = f * i;
      set i = i+1;
  end while;
    RETURN f;
END $$
DELIMITER ;
```

□ Bind parameters

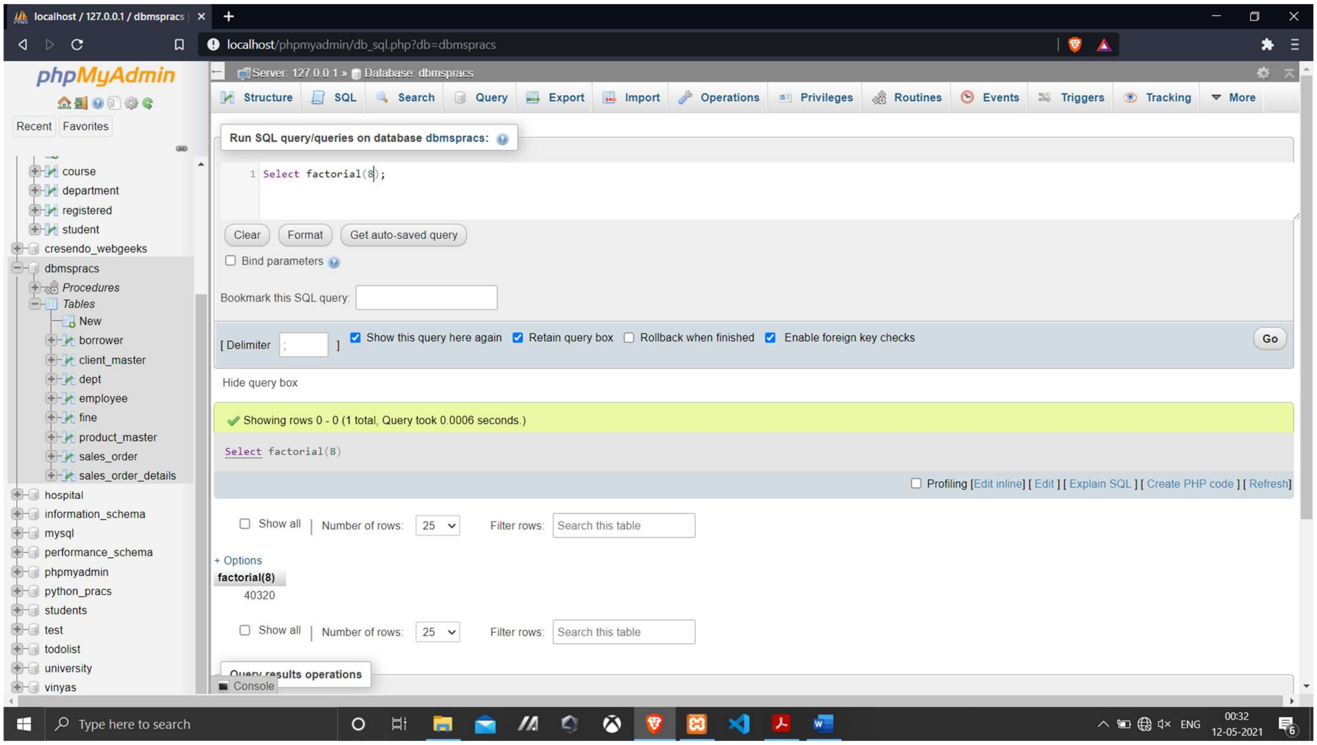
Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0069 seconds.)

```
CREATE FUNCTION factorial (n int) RETURNS int DETERMINISTIC BEGIN DECLARE i , f int; set i = 1; set i =1; set f = 1; while i <= n do set f = f * i; set i = i+1; end while; RETURN f; END
```

Select factorial(8);



Run SQL query/queries on database dbmspracs:

```
Select factorial(8);
```

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

✓ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
Select factorial(8)
```

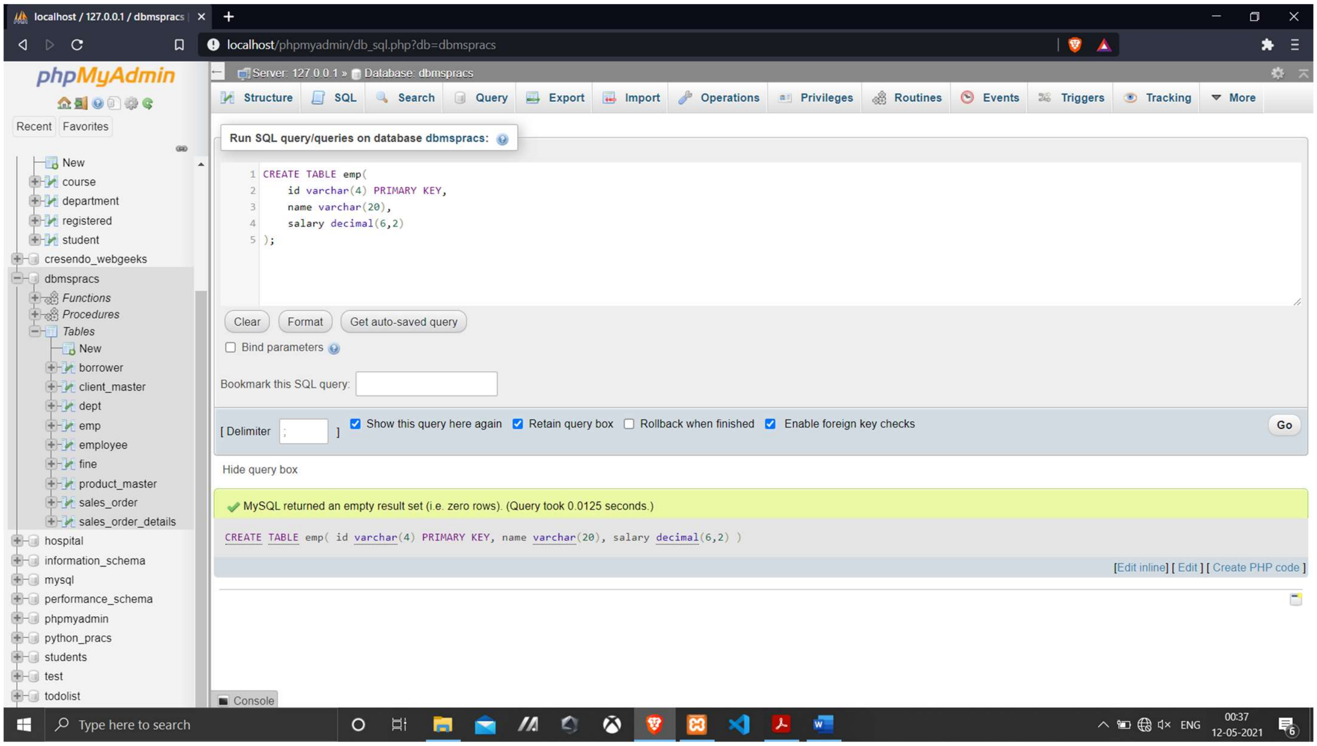
□ Show all | Number of rows: 25 Filter rows: Search this table

factorial(8)
40320

□ Show all | Number of rows: 25 Filter rows: Search this table

2. Create table emp (id,name,salary) and insert 3 records in it.

```
create table emp(id varchar(4) PRIMARY KEY,
                    name varchar(20),
                    salary decimal(6,2)
);
```



```
CREATE TABLE emp(
    id varchar(4) PRIMARY KEY,
    name varchar(20),
    salary decimal(6,2)
);
```

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0125 seconds.)

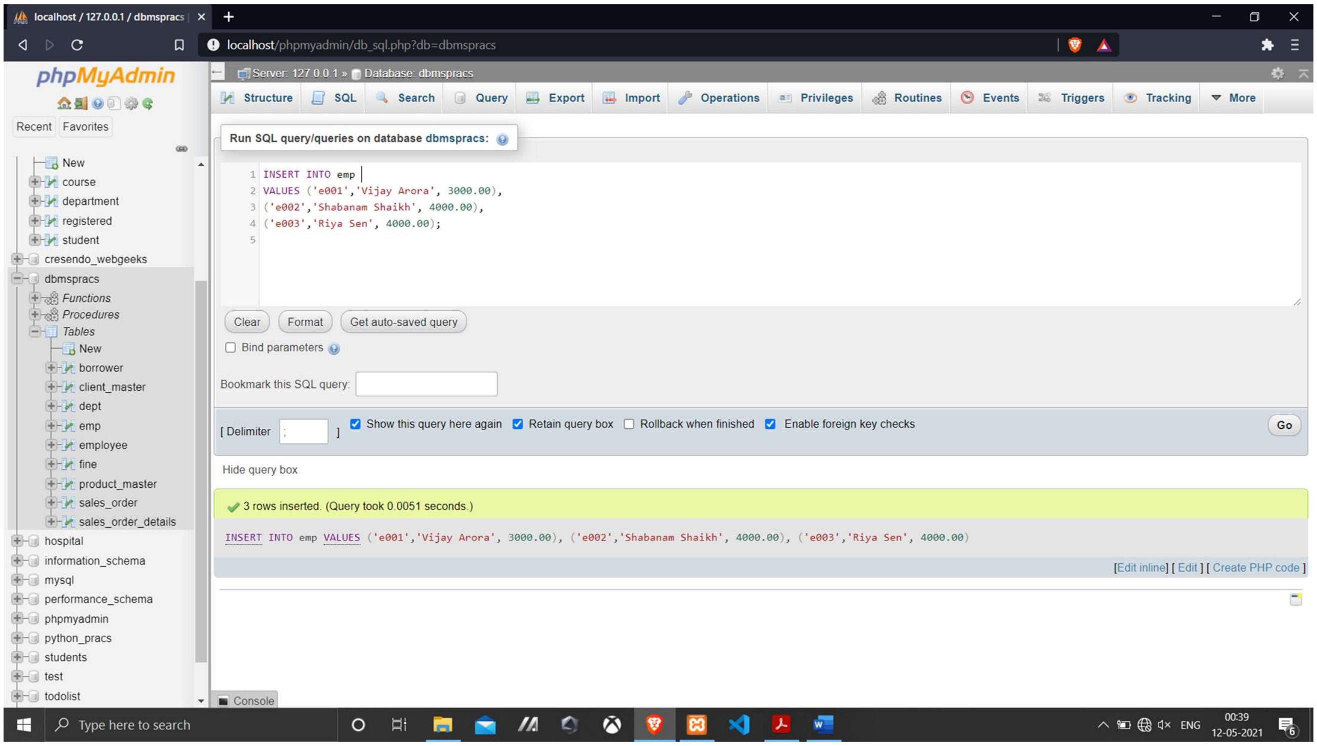
```
CREATE TABLE emp( id varchar(4) PRIMARY KEY, name varchar(20), salary decimal(6,2) )
```

insert into emp

values('e001','Vijay Arora', 3000.00),

('e002','Shabanam Shaikh', 4000.00),

('e003','Riya Sen', 4000.00);



Run SQL query/queries on database dbmspracs:

```
INSERT INTO emp
VALUES ('e001','Vijay Arora', 3000.00),
('e002','Shabanam Shaikh', 4000.00),
('e003','Riya Sen', 4000.00);
```

□ Bind parameters

Bookmark this SQL query:

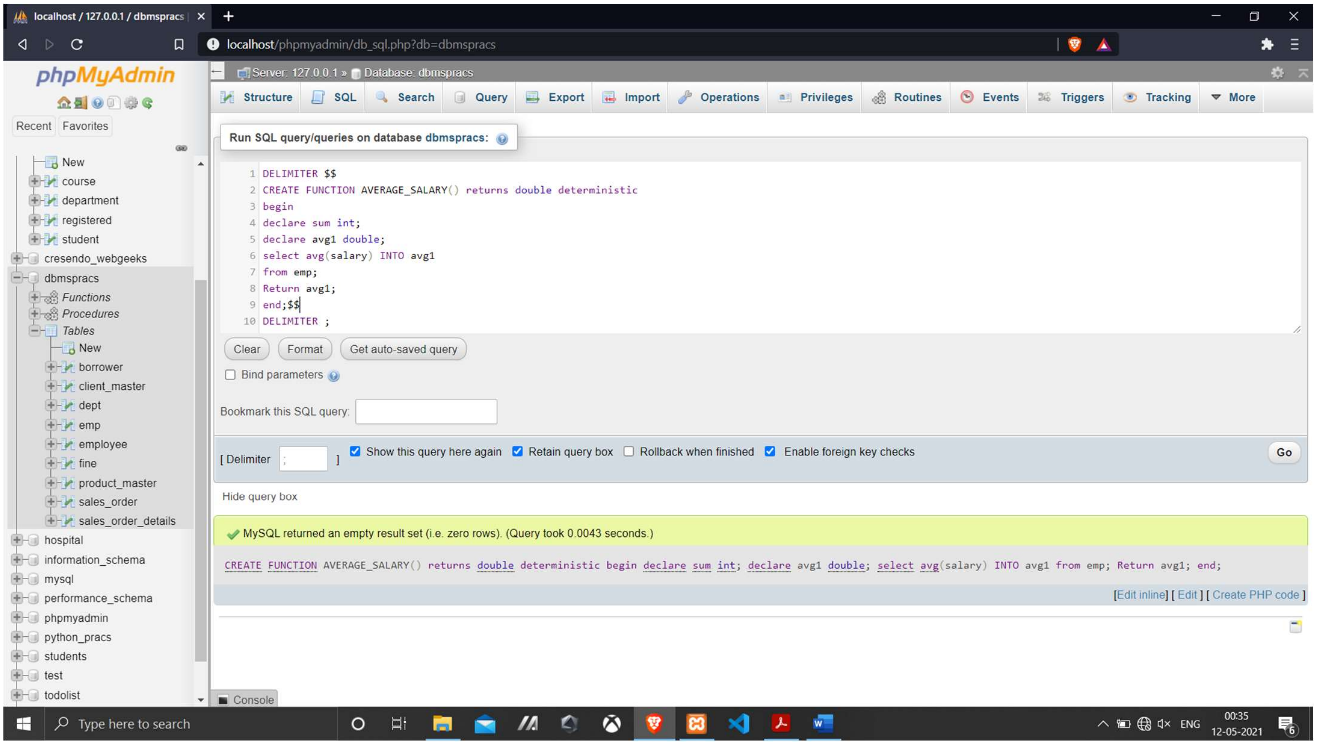
[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks Go

✓ 3 rows inserted. (Query took 0.0051 seconds.)

```
INSERT INTO emp VALUES ('e001','Vijay Arora', 3000.00), ('e002','Shabanam Shaikh', 4000.00), ('e003','Riya Sen', 4000.00)
```

3. Write a function to find average salary from emp table

```
DELIMITER $$
CREATE FUNCTION AVERAGE_SALARY() returns double deterministic
begin
declare sum int;
declare avg1 double;
select avg(salary) INTO avg1
from emp;
Return avg1;
end;$$
DELIMITER ;
```



Run SQL query/queries on database dbmspracs:

```
DELIMITER $$
CREATE FUNCTION AVERAGE_SALARY() returns double deterministic
begin
declare sum int;
declare avg1 double;
select avg(salary) INTO avg1
from emp;
Return avg1;
end;$$
DELIMITER ;
```

□ Bind parameters

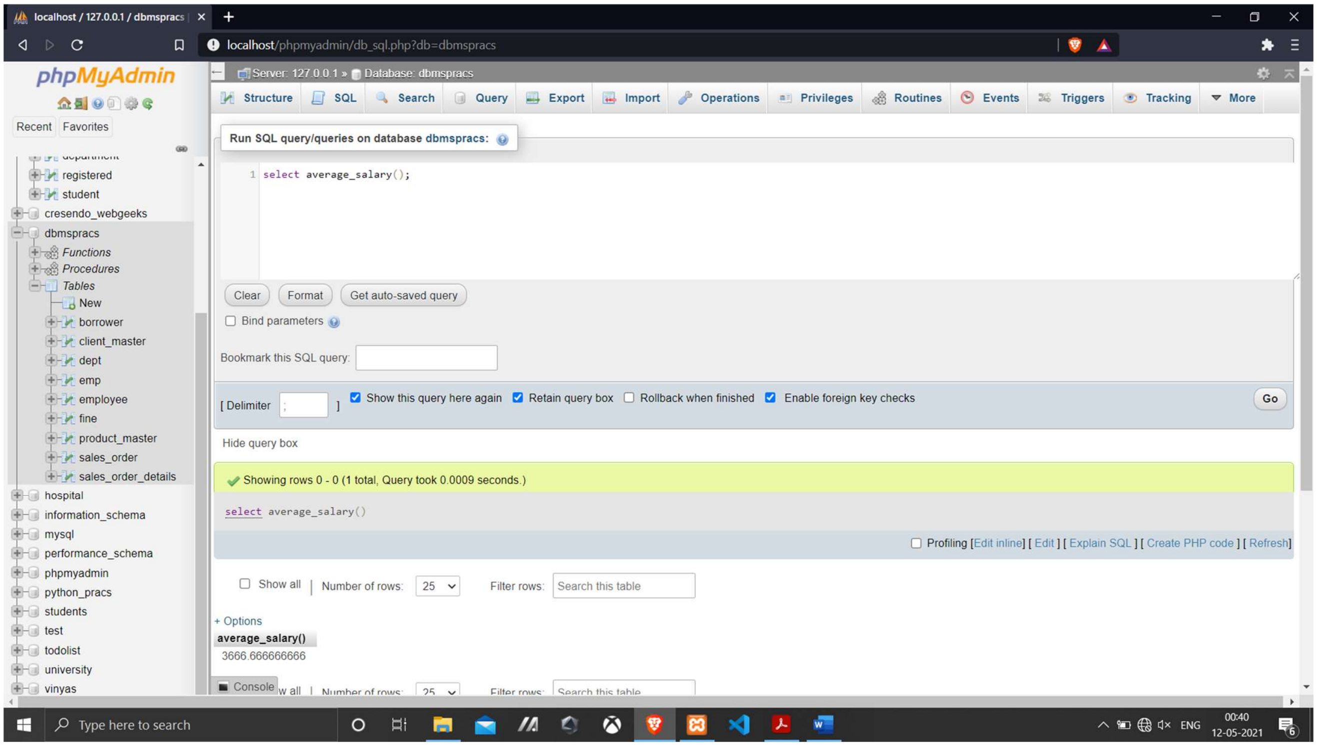
Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0043 seconds.)

```
CREATE FUNCTION AVERAGE_SALARY() returns double deterministic begin declare sum int; declare avg1 double; select avg(salary) INTO avg1 from emp; Return avg1; end;
```

select average_salary();



```
select average_salary();
```

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

✓ Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
select average_salary()
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

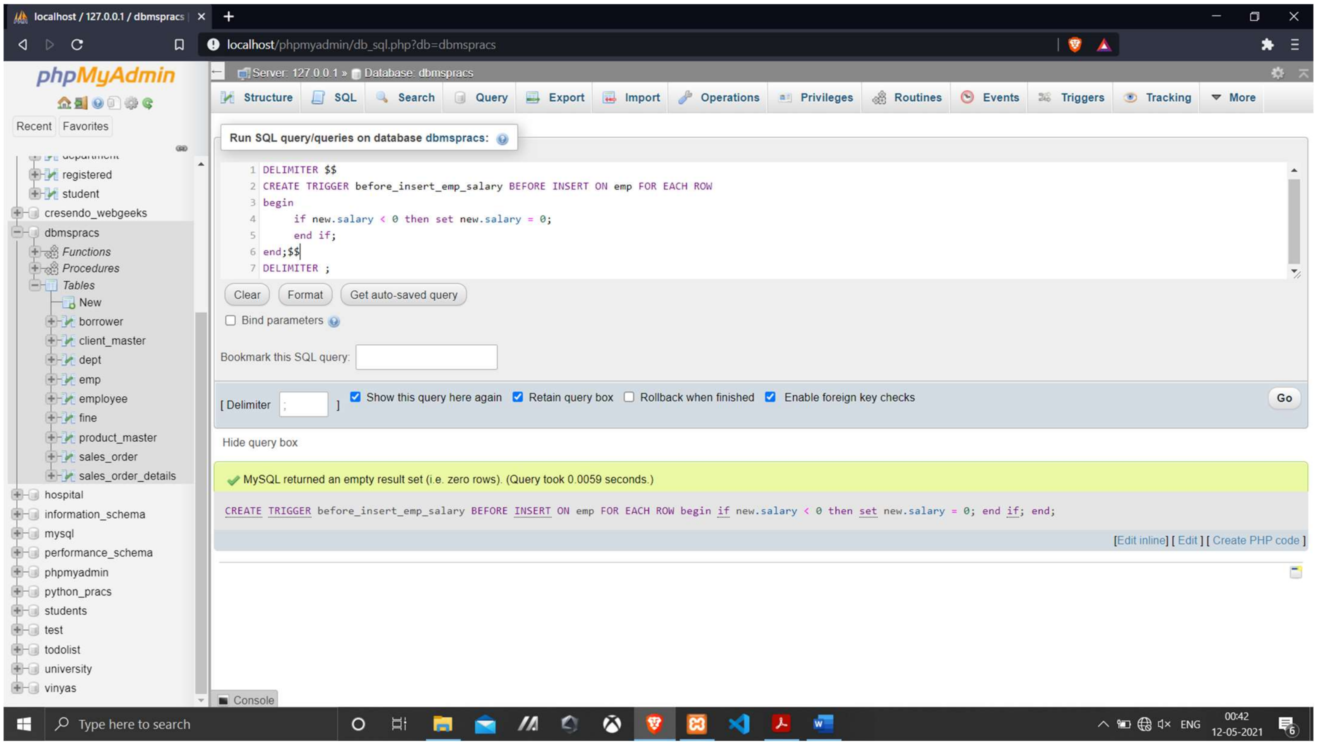
□ Show all | Number of rows: 25 Filter rows: Search this table

average_salary()
3666.666666666

4. Write a row-level trigger that would fire before insert and checks id salary < 0 then it sets the salary = 0.

This trigger is activated before each insert statement into emp table and checks the if salary < 0 then sets the salary =0.

```
DELIMITER $$
CREATE TRIGGER before_insert_emp_salary BEFORE INSERT ON emp
FOR EACH ROW
begin
    if new.salary < 0 then set new.salary = 0;
    end if;
end;$$
DELIMITER ;
```



Run SQL query/queries on database dbmspracs:

```
DELIMITER $$
CREATE TRIGGER before_insert_emp_salary BEFORE INSERT ON emp FOR EACH ROW
begin
     if new.salary < 0 then set new.salary = 0;
     end if;
end;$$
DELIMITER ;
```

☐ Bind parameters

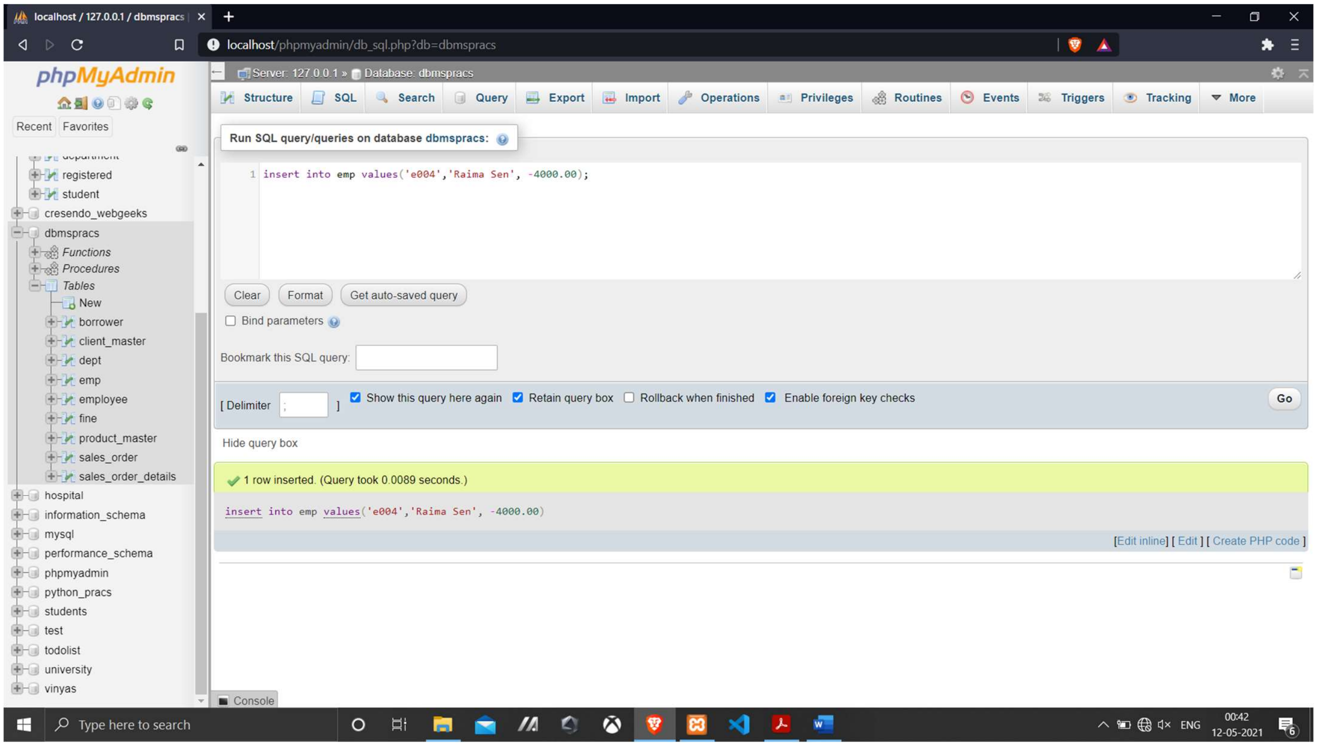
Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box ☐ Rollback when finished ☑ Enable foreign key checks

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0059 seconds.)

```
CREATE TRIGGER before_insert_emp_salary BEFORE INSERT ON emp FOR EACH ROW begin if new.salary < 0 then set new.salary = 0; end if; end;
```

insert into emp values('e004','Raima Sen', -4000.00);



Run SQL query/queries on database dbmspracs:

```
insert into emp values('e004','Raima Sen', -4000.00);
```

☐ Bind parameters

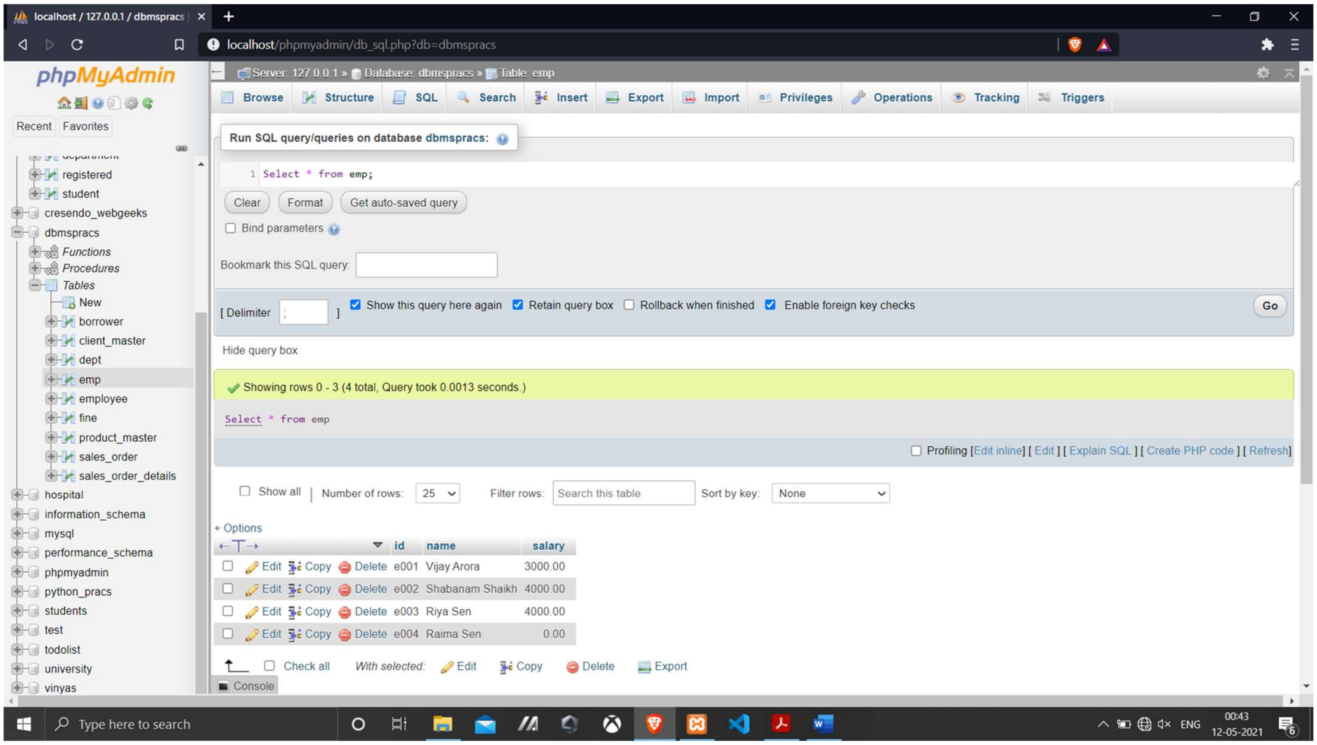
Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box ☐ Rollback when finished ☑ Enable foreign key checks

✓ 1 row inserted. (Query took 0.0089 seconds.)

```
insert into emp values('e004','Raima Sen', -4000.00)
```

Select * from emp;


localhost / 127.0.0.1 / dbmspracs
localhost/phpmyadmin/db_sql.php?db=dbmspracs
phpMyAdmin
Recent Favorites
Server: 127.0.0.1 » Database: dbmspracs » Table: emp
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
registered
student
cresendo_webgeeks
dbmspracs
Functions
Procedures
Tables
New
borrower
client_master
dept
emp
employee
fine
product_master
sales_order
sales_order_details
hospital
information_schema
mysql
performance_schema
phpmyadmin
python_pracs
students
test
todolist
university
vinyas
Clear Format Get auto-saved query
Go
Hide query box
☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
+ Options
☐ Check all With selected: Edit Copy Delete Export
Console
Type here to search
ENG 00:43 12-05-2021

Run SQL query/queries on database dbmspracs:

```
Select * from emp;
```

☐ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box ☐ Rollback when finished ☑ Enable foreign key checks

✓ Showing rows 0 - 3 (4 total, Query took 0.0013 seconds.)

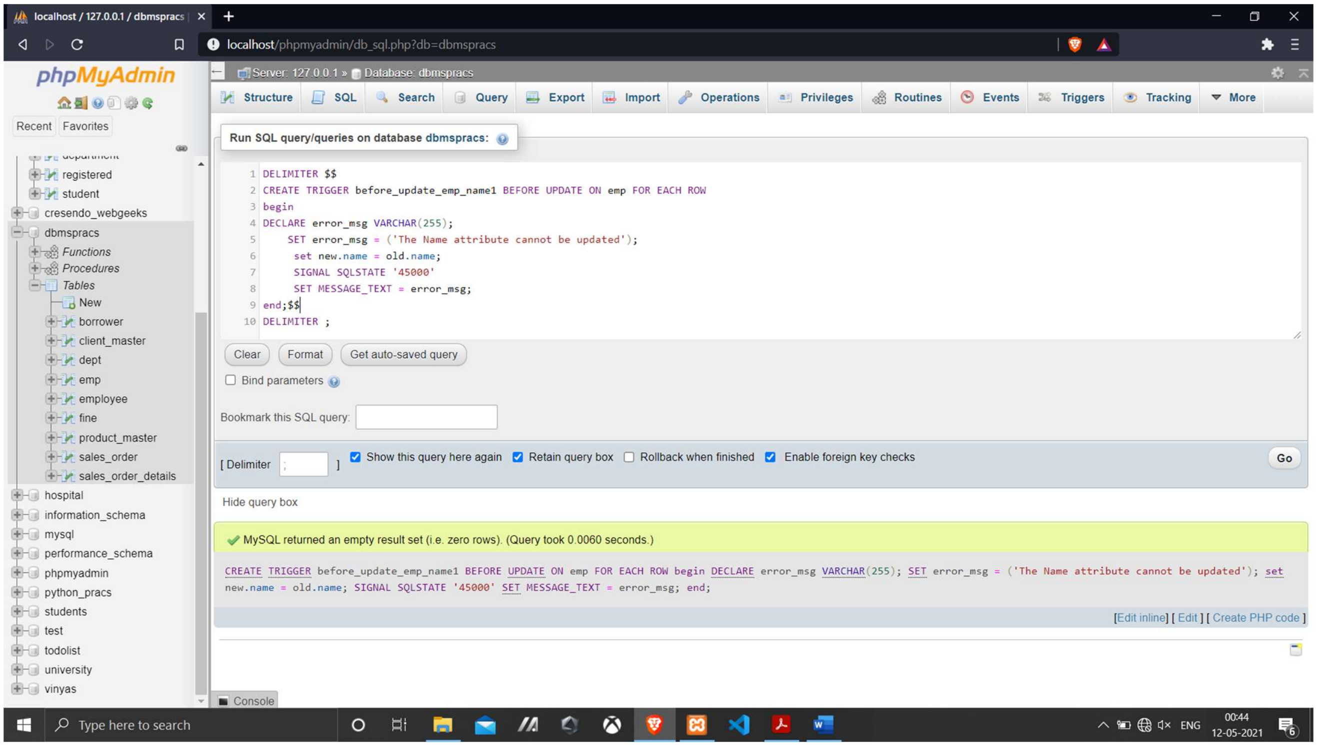
```
Select * from emp
```

☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	name	salary
☐ Edit Copy Delete	e001	Vijay Arora	3000.00
☐ Edit Copy Delete	e002	Shabanam Shaikh	4000.00
☐ Edit Copy Delete	e003	Riya Sen	4000.00
☐ Edit Copy Delete	e004	Raima Sen	0.00

5. Write a row –level trigger that would fire when user tries to update name. Triggers should not allow user to update name field and displays appropriate message

```
DELIMITER $$

CREATE TRIGGER before_update_emp_name1 BEFORE UPDATE ON emp
FOR EACH ROW

begin

DECLARE error_msg VARCHAR(255);

   SET error_msg = ('The Name attribute cannot be updated');

    set new.name = old.name;

    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = error_msg;

end;$$

DELIMITER ;
```



Run SQL query/queries on database dbmspracs:

```
DELIMITER $$
CREATE TRIGGER before_update_emp_name1 BEFORE UPDATE ON emp FOR EACH ROW
begin
DECLARE error_msg VARCHAR(255);
    SET error_msg = ('The Name attribute cannot be updated');
     set new.name = old.name;
     SIGNAL SQLSTATE '45000'
     SET MESSAGE_TEXT = error_msg;
end;$$
DELIMITER ;
```

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

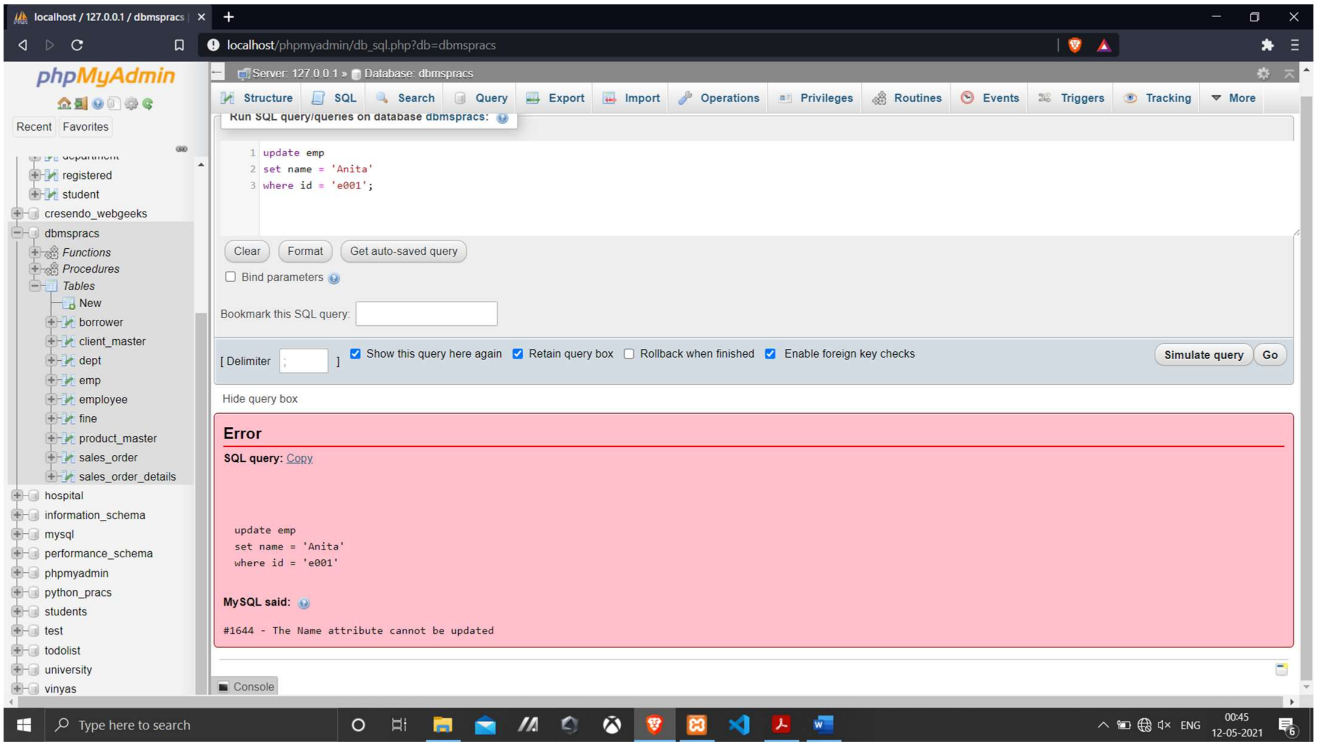
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0060 seconds.)

```
CREATE TRIGGER before_update_emp_name1 BEFORE UPDATE ON emp FOR EACH ROW begin DECLARE error_msg VARCHAR(255); SET error_msg = ('The Name attribute cannot be updated'); set new.name = old.name; SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = error_msg; end;
```

update emp

set name = 'Anita'

where id = 'e001';



Run SQL query/queries on database dbmspracs:

```
update emp
set name = 'Anita'
where id = 'e001';
```

□ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box □ Rollback when finished ☑ Enable foreign key checks

Error

SQL query: Copy

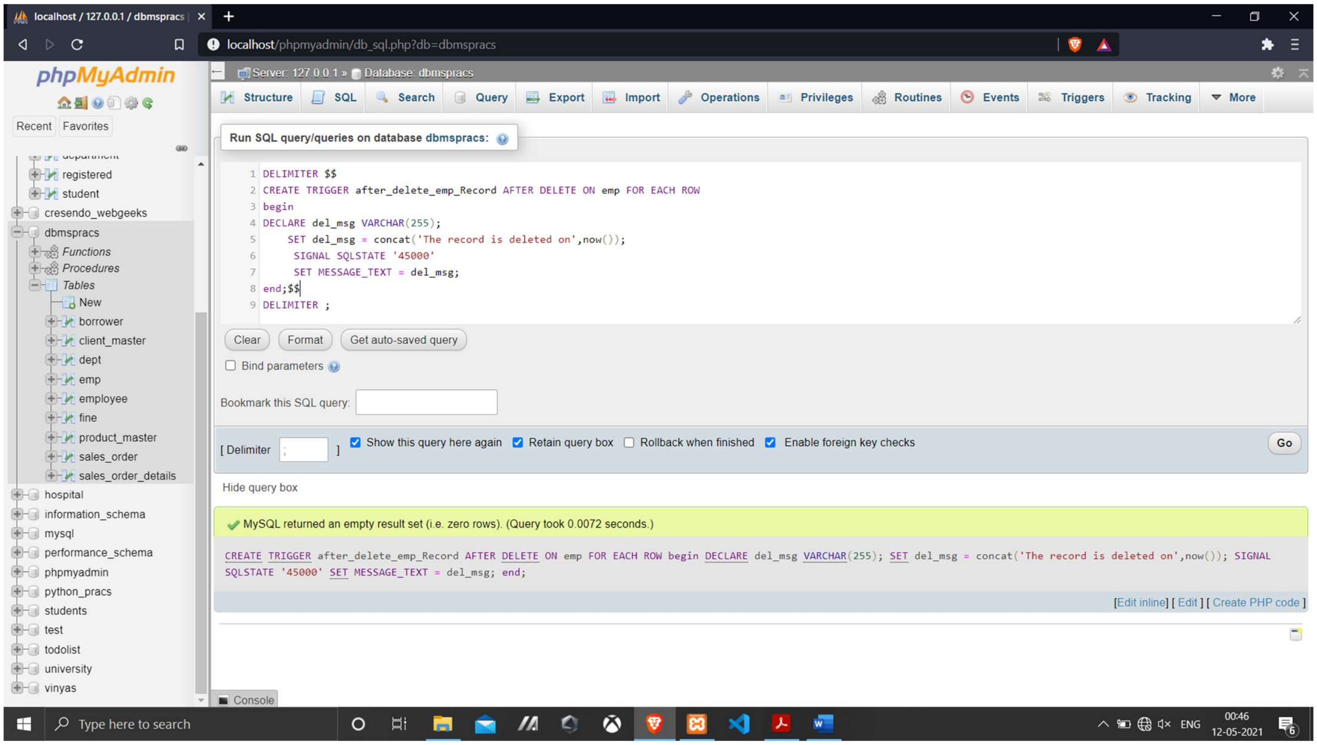
```
update emp
set name = 'Anita'
where id = 'e001'
```

MySQL said:

```
#1644 - The Name attribute cannot be updated
```

6. Write a row level trigger that would fire AFTER delete operation is performed on emp table displaying date on which data is deleted

```
DELIMITER $$

CREATE TRIGGER after_delete_emp_Record AFTER DELETE ON emp FOR EACH ROW

begin

DECLARE del_msg VARCHAR(255);

    SET del_msg = concat('The record is deleted on',now());

     SIGNAL SQLSTATE '45000'

     SET MESSAGE_TEXT = del_msg;

end;$$

DELIMITER ;
```



Run SQL query/queries on database dbmspracs:

```
DELIMITER $$
CREATE TRIGGER after_delete_emp_Record AFTER DELETE ON emp FOR EACH ROW
begin
DECLARE del_msg VARCHAR(255);
    SET del_msg = concat('The record is deleted on',now());
     SIGNAL SQLSTATE '45000'
     SET MESSAGE_TEXT = del_msg;
end;$$
DELIMITER ;
```

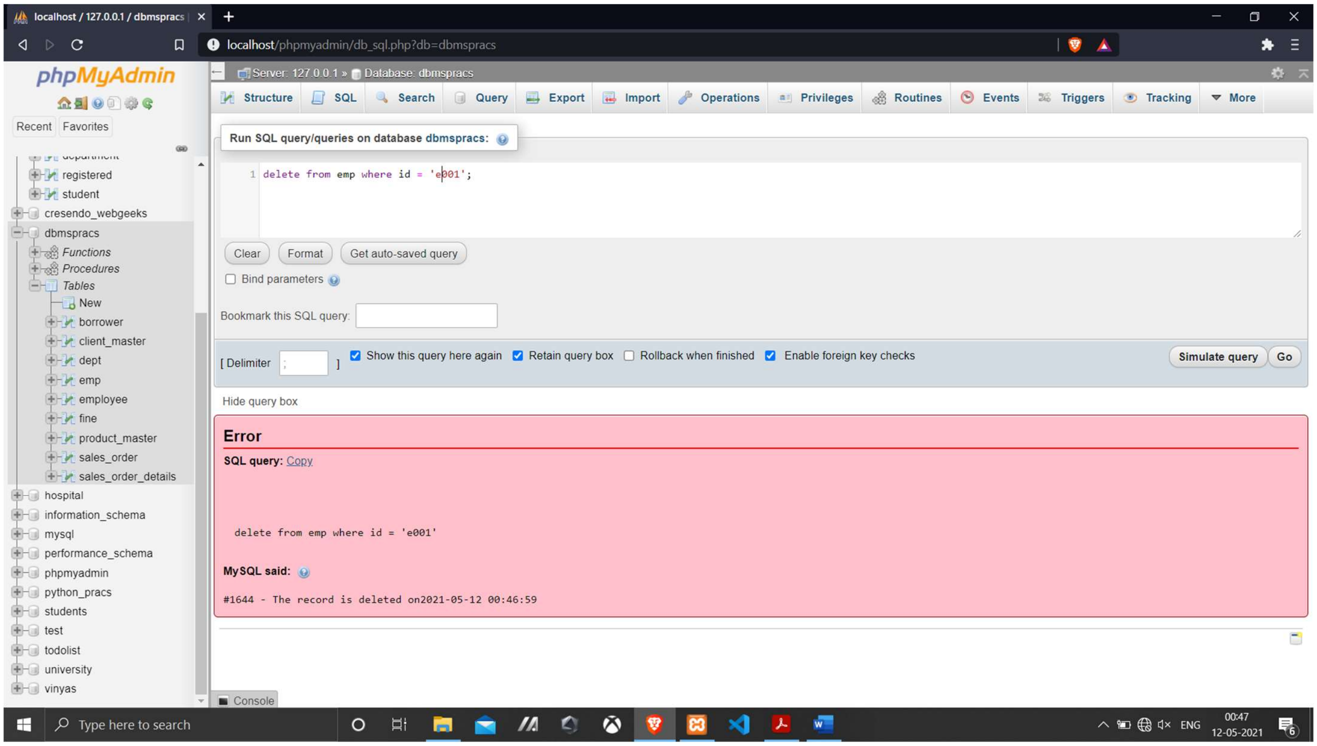
☐ Bind parameters

[Delimiter ;] ☑ Show this query here again ☑ Retain query box ☐ Rollback when finished ☑ Enable foreign key checks

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0072 seconds.)

```
CREATE TRIGGER after_delete_emp_Record AFTER DELETE ON emp FOR EACH ROW begin DECLARE del_msg VARCHAR(255); SET del_msg = concat('The record is deleted on',now()); SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = del_msg; end;
```

delete from emp where id = 'e001';



```
delete from emp where id = 'e001';
```

Error

SQL query: Copy

```
delete from emp where id = 'e001'
```

MySQL said:

```
#1644 - The record is deleted on2021-05-12 00:46:59
```

Postlab:

Difference between procedures and Function on SQL.

Ans.

Functions

- A function has a return type and returns a value.
- You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions.
- A function does not allow output parameters
- You cannot manage transactions inside a function.
- You cannot call stored procedures from a function
- You can call a function using a select statement.

Stored Procedures

- A procedure does not have a return type. But it returns values using the OUT parameters.
- You can use DML queries such as insert, update, select etc… with procedures.
- A procedure allows both input and output parameters.
- You can manage transactions inside a function.
- You can call a function from a stored procedure.
- You cannot call a procedure using select statements.