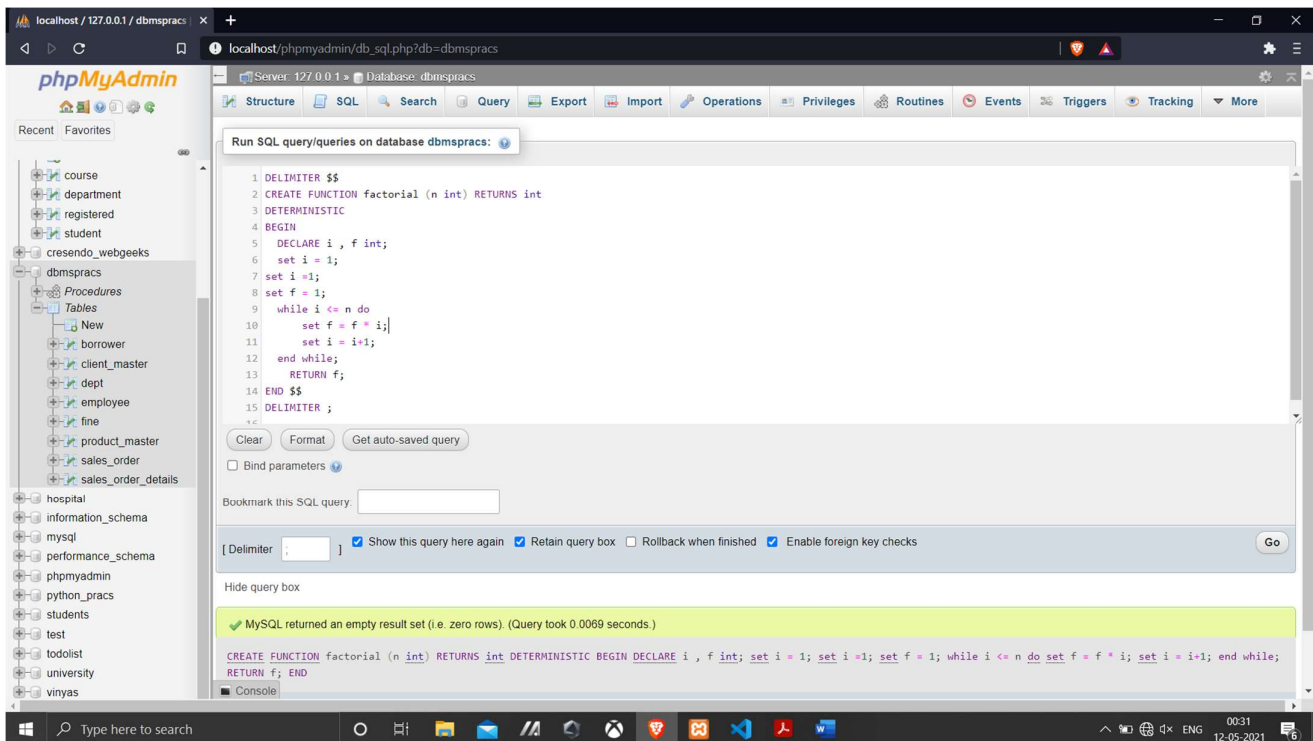


DBMS Practical Implementation, Lab 9.

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;
```



Select factorial(8);

The screenshot shows the phpMyAdmin web interface in a browser window. The address bar indicates the URL is localhost/phpmyadmin/db_sql.php?db=dbmspracs. The interface includes a sidebar with a database structure tree on the left, listing databases like 'course', 'department', 'registered', 'student', 'cresendo_webgeeks', and 'dbmspracs'. The 'dbmspracs' database is selected, showing tables like 'New', 'borrower', 'client_master', 'dept', 'employee', 'fine', 'product_master', 'sales_order', and 'sales_order_details'. The main panel is titled 'Run SQL query/queries on database dbmspracs:'. It contains a text area with the SQL query '1 Select factorial(8);'. Below the query area are buttons for 'Clear', 'Format', and 'Get auto-saved query'. There is a checkbox for 'Bind parameters' and a 'Bookmark this SQL query:' field. A row of checkboxes includes 'Show this query here again' (checked), 'Retain query box' (checked), 'Rollback when finished' (unchecked), and 'Enable foreign key checks' (checked). A 'Go' button is on the right. Below this, a green status bar says 'Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)'. The query result is displayed in a table with one row: 'factorial(8)' with the value '40320'. At the bottom, there are tabs for 'Query results operations' and 'Console'. The Windows taskbar at the very bottom shows the time as 00:32 on 12-05-2021.

localhost / 127.0.0.1 / dbmspracs

localhost/phpmyadmin/db_sql.php?db=dbmspracs

Server: 127.0.0.1 » Database: dbmspracs

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More

Run SQL query/queries on database dbmspracs:

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

Clear Format Get auto-saved query

☐ Bind parameters

Bookmark this SQL query:

[Delimiter :] ☒ Show this query here again ☒ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks Go

Hide query box

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

Select factorial(8)

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

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

+ Options

factorial(8)
40320

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

Query results operations

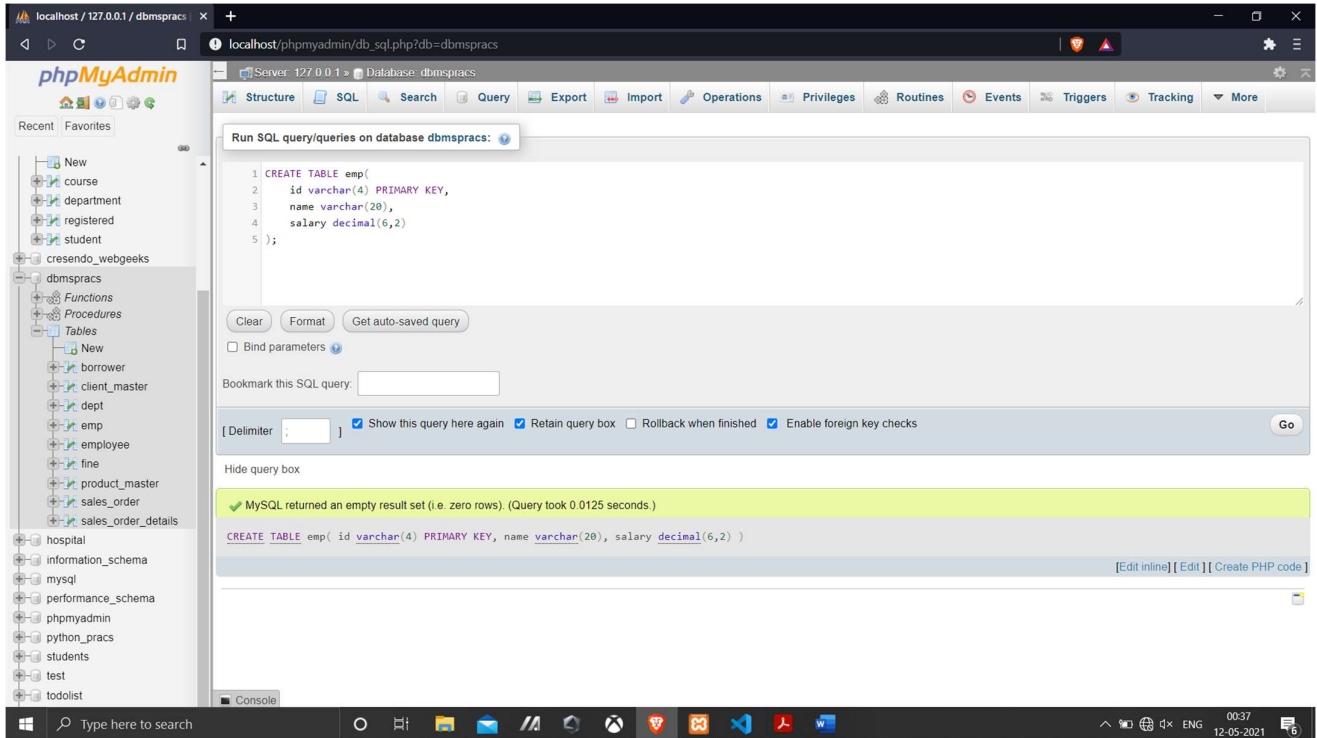
Console

Type here to search

00:32 12-05-2021

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)
);

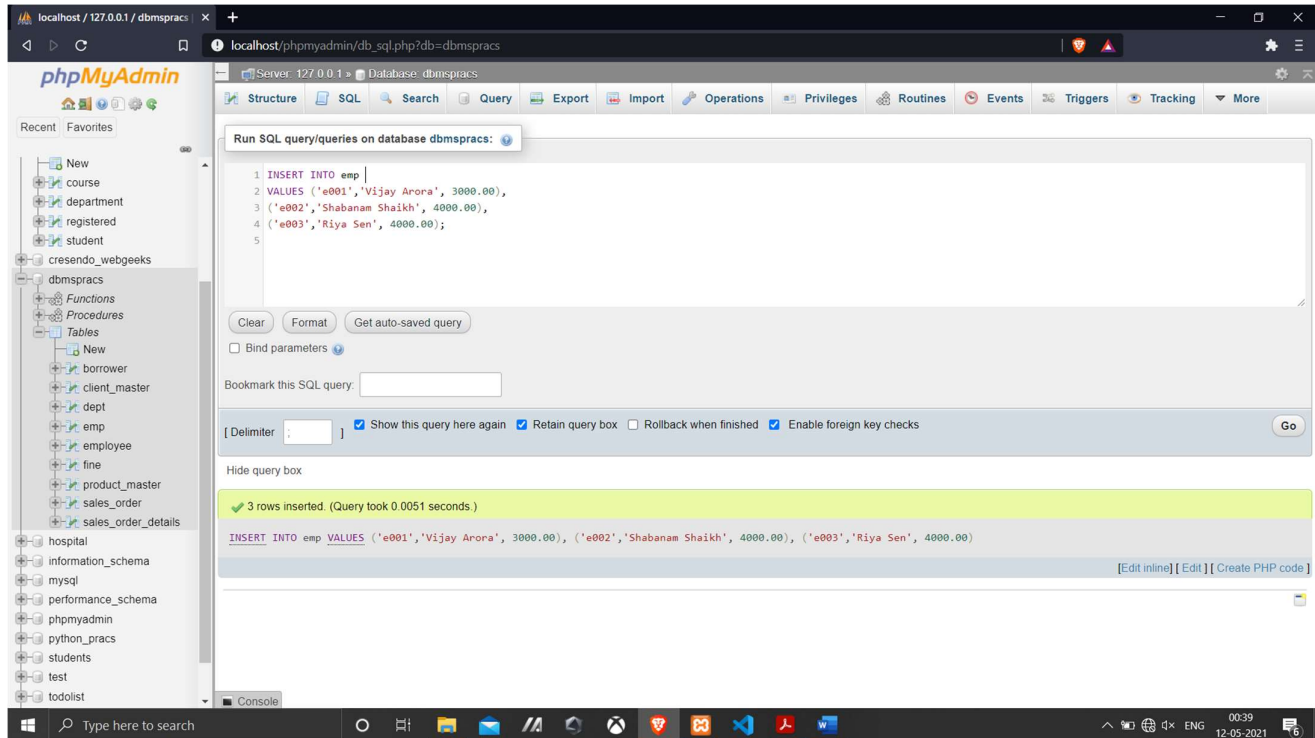


insert into emp

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;

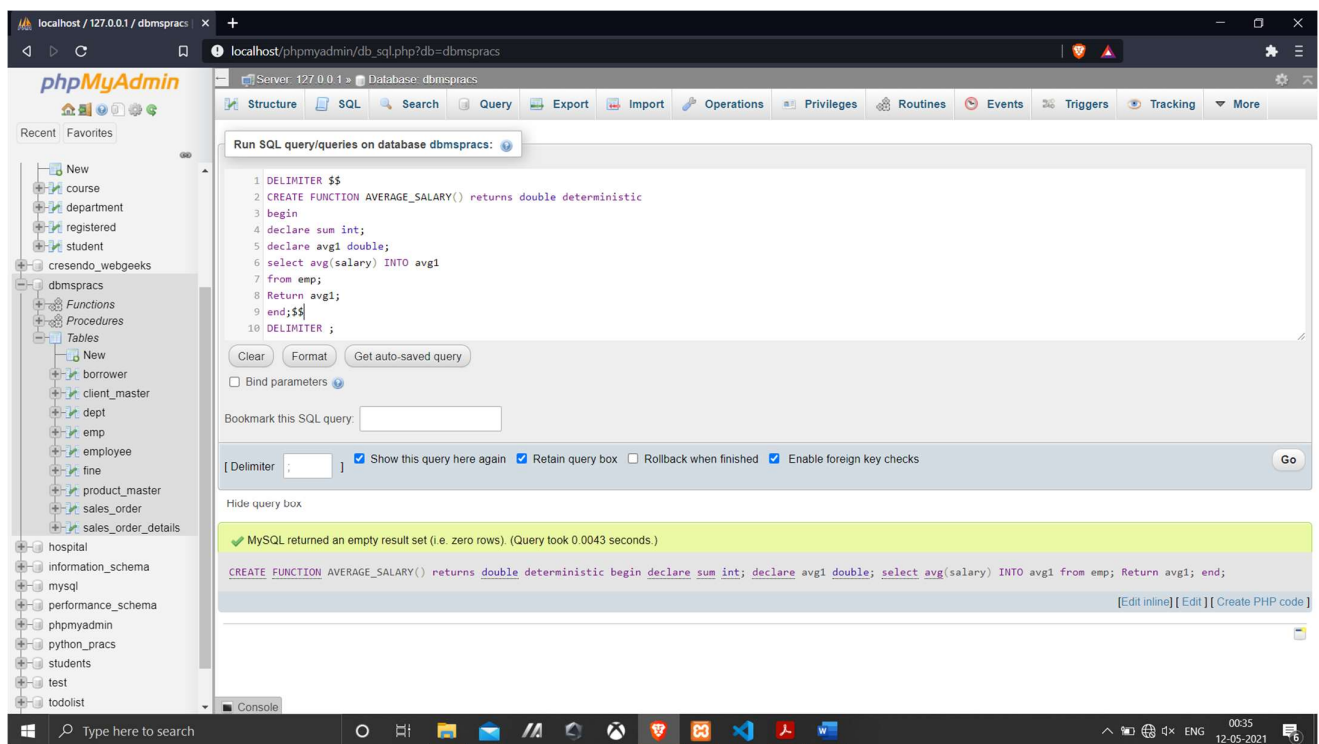
select avg(salary) INTO avg1

from emp;

Return avg1;

end;\$\$

DELIMITER ;



select average_salary();

The screenshot shows the phpMyAdmin web interface in a browser window. The address bar indicates the URL is `localhost/phpmyadmin/db_sql.php?db=dbmspracs`. The left sidebar displays a database structure tree for `dbmspracs`, including tables like `registered`, `student`, `client_master`, `dept`, `emp`, `employee`, `fine`, `product_master`, `sales_order`, and `sales_order_details`.

The main panel is titled "Run SQL query/queries on database dbmspracs:". The SQL query editor contains the query `1 select average_salary();`. Below the editor are buttons for "Clear", "Format", and "Get auto-saved query", along with a "Bind parameters" checkbox. A "Bookmark this SQL query:" field is also present. The "Go" button is located at the bottom right of the query execution area.

Below the query execution area, a green status bar indicates "Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)". The results table shows a single row with the value `3666.666666666` for the `average_salary()` function. The table has a header row with the column name `average_salary()`.

The bottom of the interface shows a Windows taskbar with the search bar and various application icons. The system clock in the bottom right corner displays the time as 00:40 on 12-05-2021.

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
```

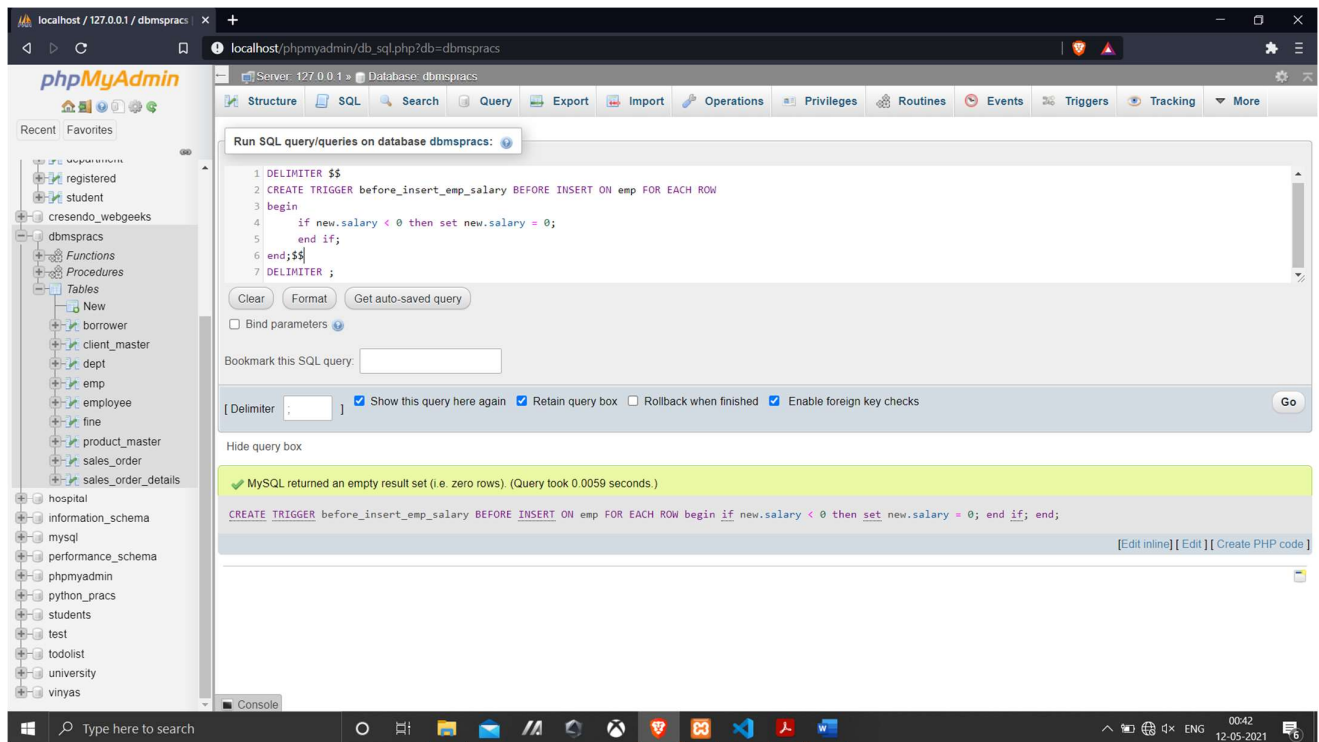
```
begin
```

```
    if new.salary < 0 then set new.salary = 0;
```

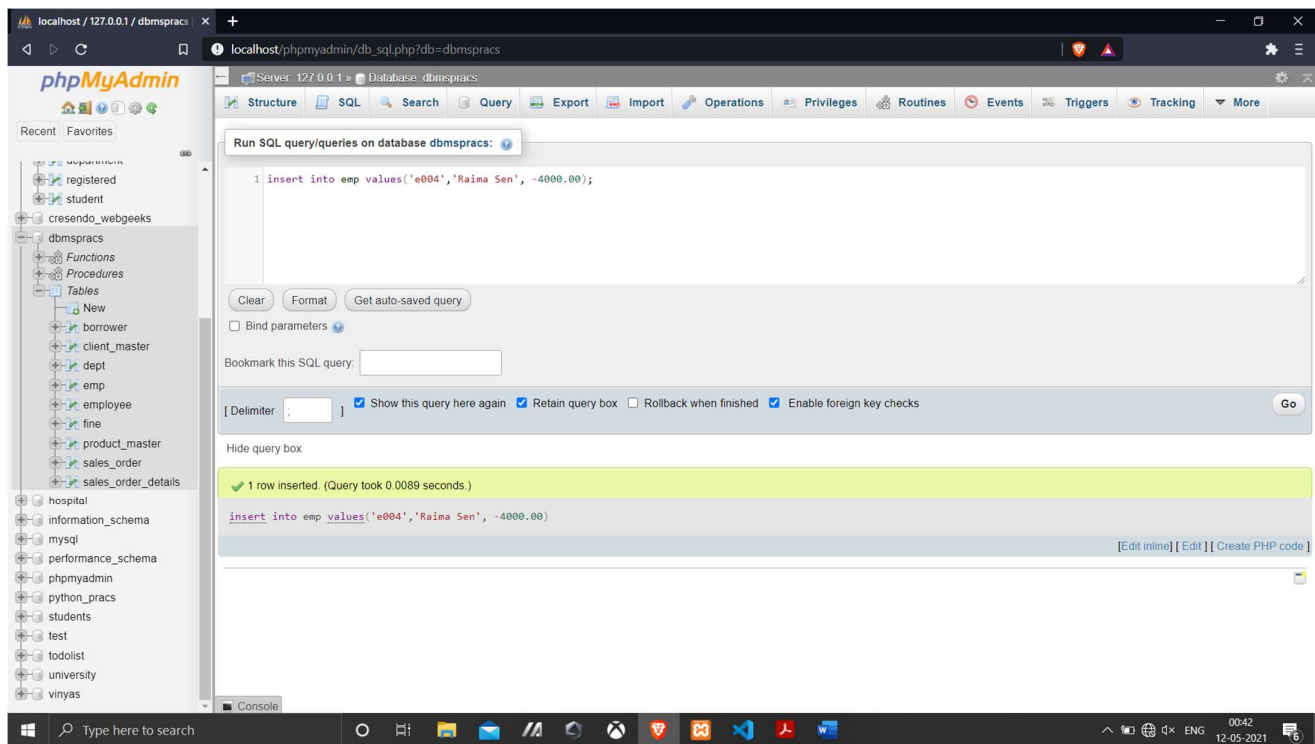
```
    end if;
```

```
end;$$
```

```
DELIMITER ;
```

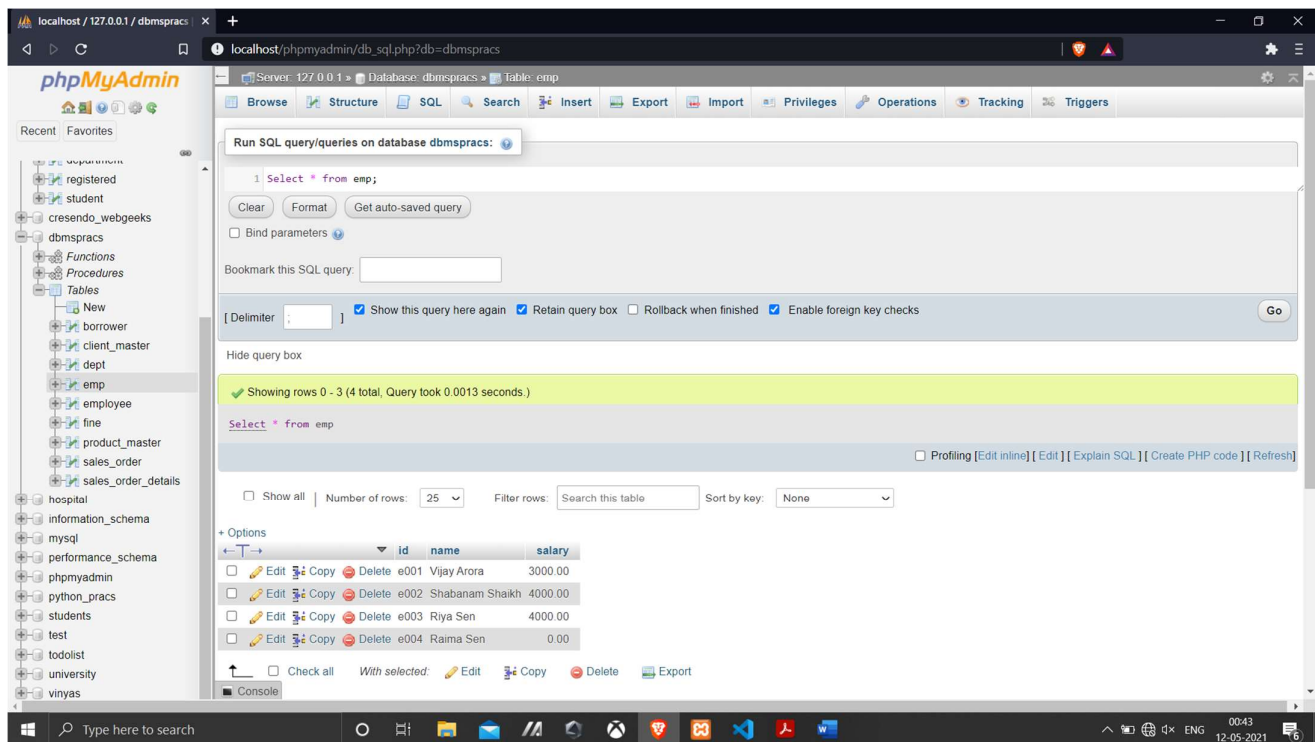


insert into emp values('e004','Raima Sen', -4000.00);



The screenshot shows the phpMyAdmin interface with the 'emp' table selected in the 'dbmspracs' database. The SQL query 'insert into emp values('e004','Raima Sen', -4000.00);' has been executed successfully. The console displays a green message: '1 row inserted. (Query took 0.0089 seconds)'. Below the message, the executed query is shown: 'insert into emp values('e004','Raima Sen', -4000.00)'. The left sidebar shows the database structure, and the top navigation bar includes options like Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, and Tracking.

Select * from emp;



The screenshot shows the phpMyAdmin interface with the 'emp' table selected. The SQL query 'Select * from emp;' has been executed. The console displays a green message: 'Showing rows 0 - 3 (4 total. Query took 0.0013 seconds)'. Below the message, the results of the query are shown in a table with columns 'id', 'name', and 'salary'. The table contains four rows of data. The left sidebar shows the database structure, and the top navigation bar includes options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.

id	name	salary
e001	Vijay Arora	3000.00
e002	Shabanam Shaikh	4000.00
e003	Riya Sen	4000.00
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');
```

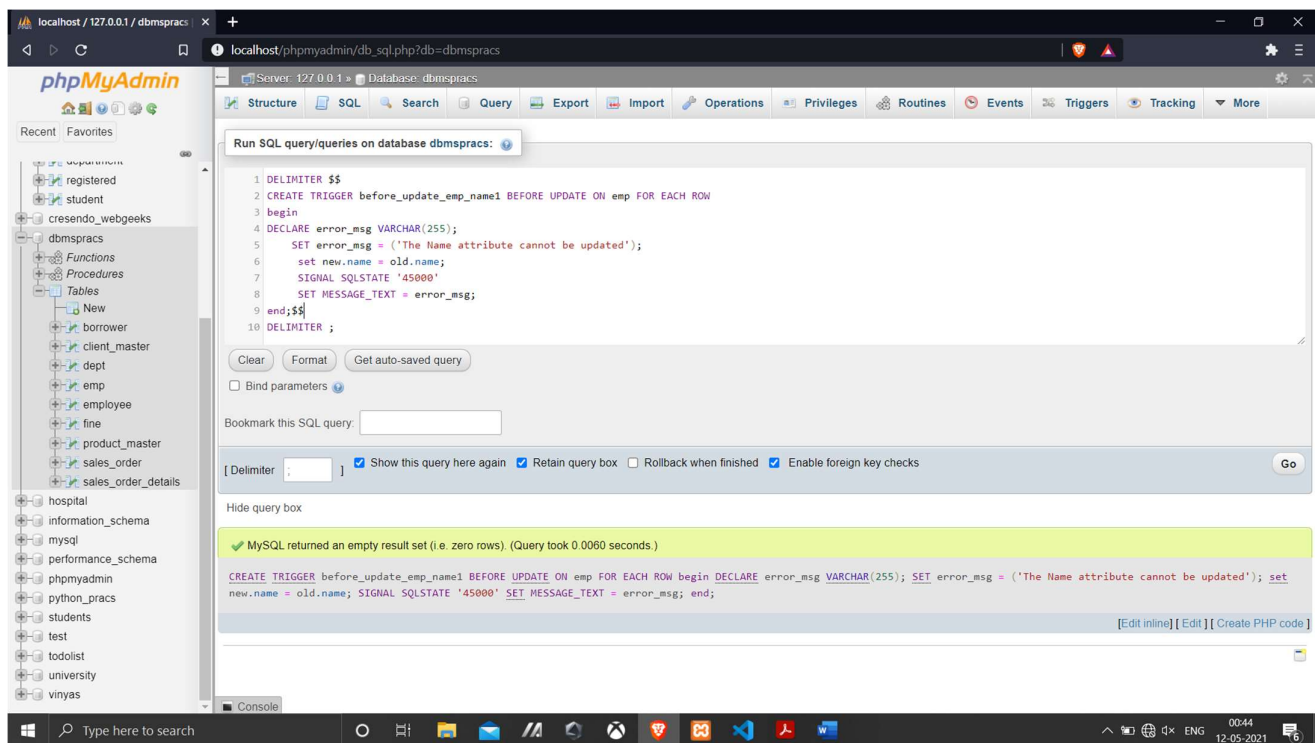
```
set new.name = old.name;
```

```
SIGNAL SQLSTATE '45000'
```

```
SET MESSAGE_TEXT = error_msg;
```

```
end;$$
```

```
DELIMITER ;
```



update emp

set name = 'Anita'

where id = 'e001';

The screenshot shows the phpMyAdmin web interface in a browser window. The address bar shows the URL `localhost/phpmyadmin/db_sql.php?db=dbmspracs`. The interface is for the database `dbmspracs`. In the SQL tab, the following query is entered:

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

Below the query editor, there are buttons for "Clear", "Format", and "Get auto-saved query". There is also a checkbox for "Bind parameters". A "Bookmark this SQL query" field is present. At the bottom of the query editor, there are checkboxes for "Show this query here again", "Retain query box", "Rollback when finished", and "Enable foreign key checks". A "Simulate query" button and a "Go" button are also visible.

An error message is displayed in a red box:

Error
SQL query: [Copy](#)

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

MySQL said:

#1644 - The Name attribute cannot be updated

The left sidebar shows a tree view of databases and tables. The right sidebar is empty. The bottom of the screen shows the Windows taskbar with the date and time 00:45 12-05-2021.

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);

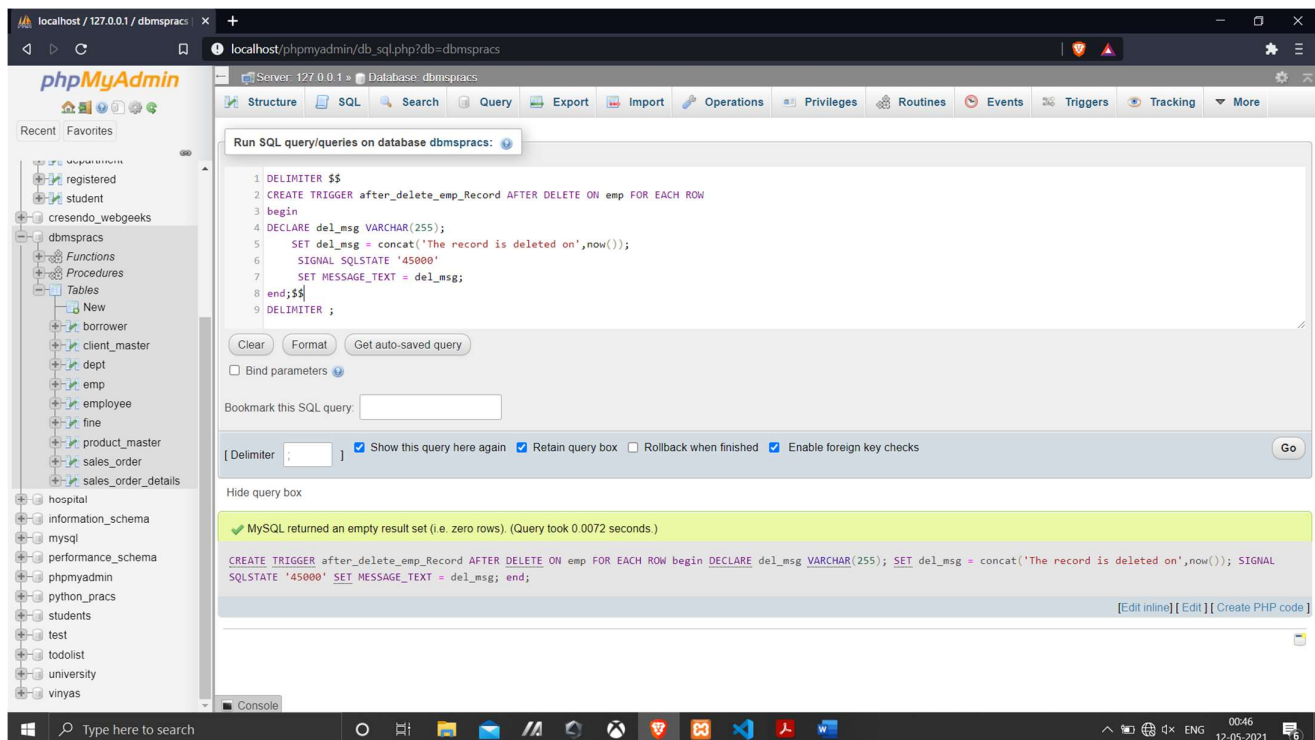
SET del_msg = concat('The record is deleted on',now());

SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = del_msg;

end;\$\$

DELIMITER ;



delete from emp where id = 'e001';

The screenshot shows the phpMyAdmin web interface in a browser. The left sidebar displays a database structure tree with the 'dbmspracs' database selected. The main panel is titled 'Run SQL query/queries on database dbmspracs:'. The SQL query editor contains the text: `1 delete from emp where id = 'e001';`. Below the editor are buttons for 'Clear', 'Format', and 'Get auto-saved query', along with a 'Bind parameters' checkbox. A 'Bookmark this SQL query' field is also present. The execution options section includes checkboxes for 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks'. The 'Simulate query' and 'Go' buttons are at the bottom right of the options. Below the query editor, a red 'Error' box is visible, containing the text: 'SQL query: COPY', 'delete from emp where id = 'e001'', 'MySQL said: ', and '#1644 - The record is deleted on 2021-05-12 00:46:59'. The browser's address bar shows 'localhost/phpmyadmin/db_sql.php?db=dbmspracs'. The Windows taskbar at the bottom shows the time as 00:47 on 12-05-2021.

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.