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

# DBMS Practical Implementation, Lab 8.

## Task 1

1.Write a stored procedure to accept salary of the employee and display grade of employee accordingly.

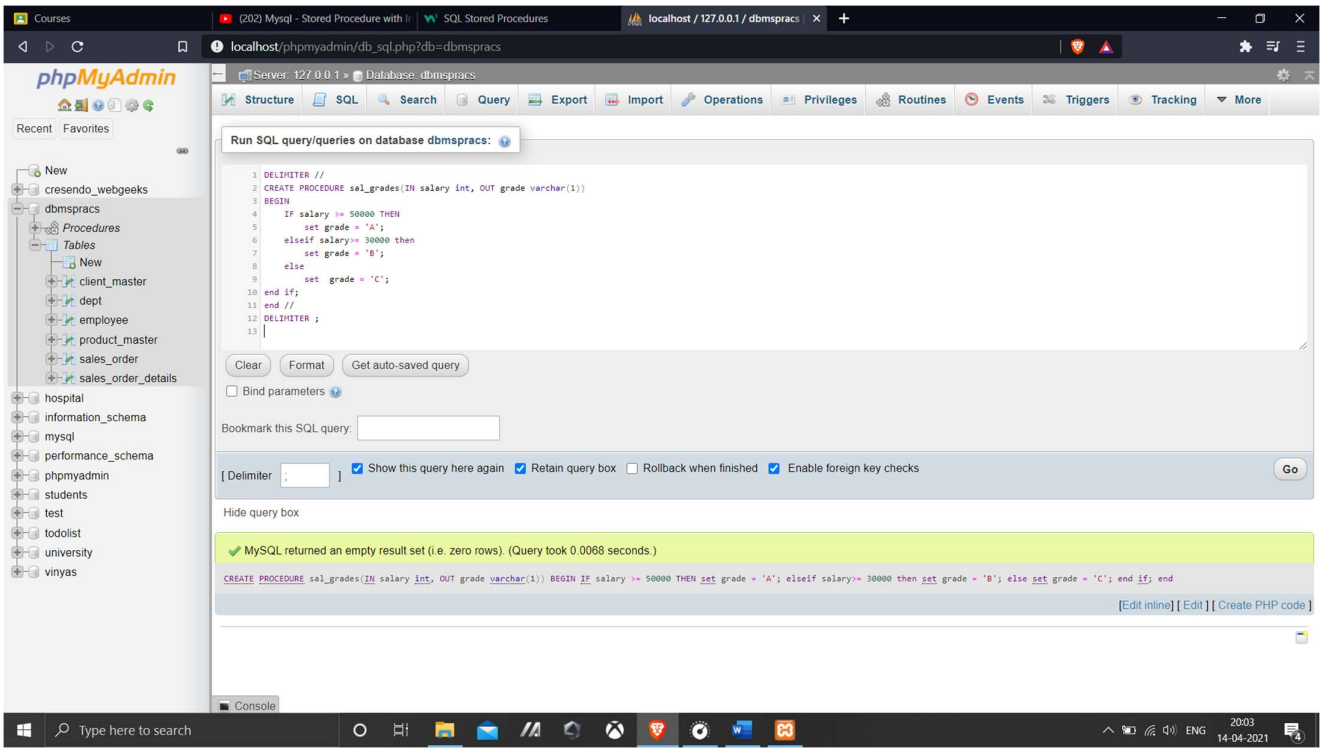
If salary > 50000 then grade is ‘A’

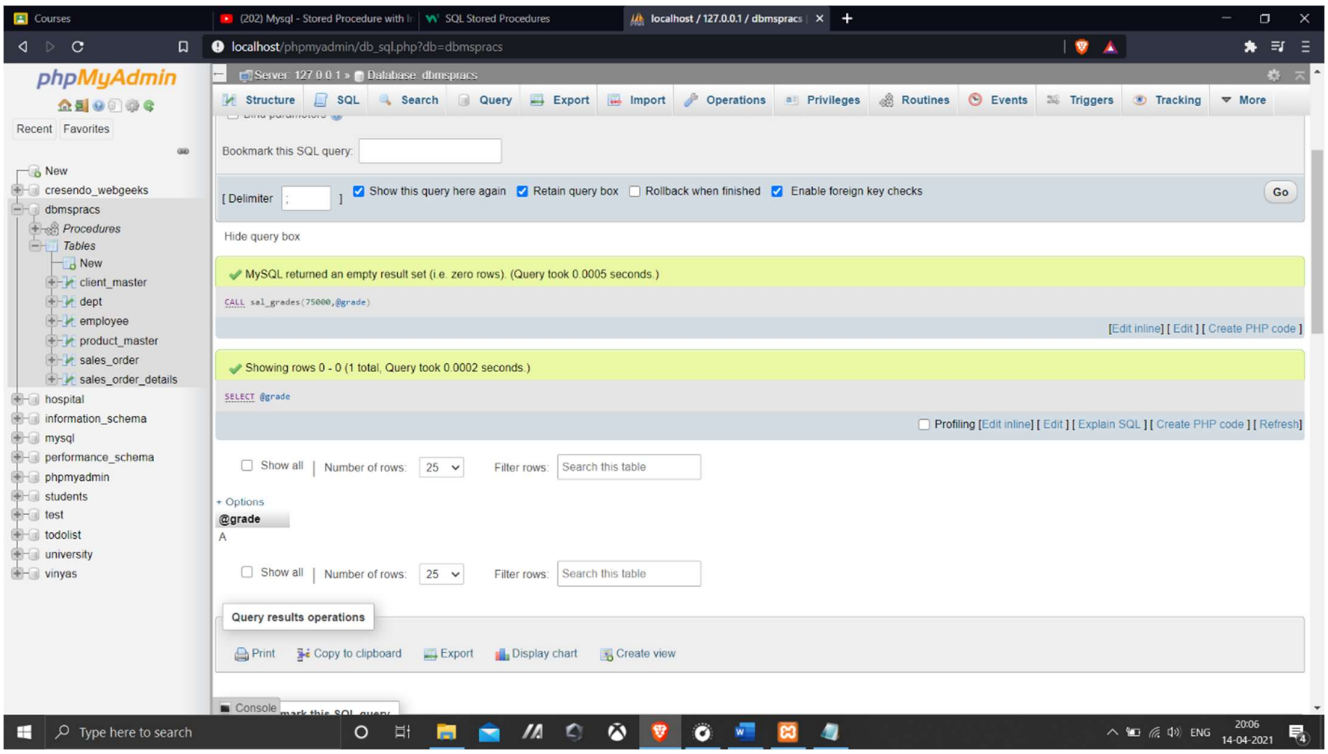
If salary between 30000 to 50000 the grade is ‘B’ and

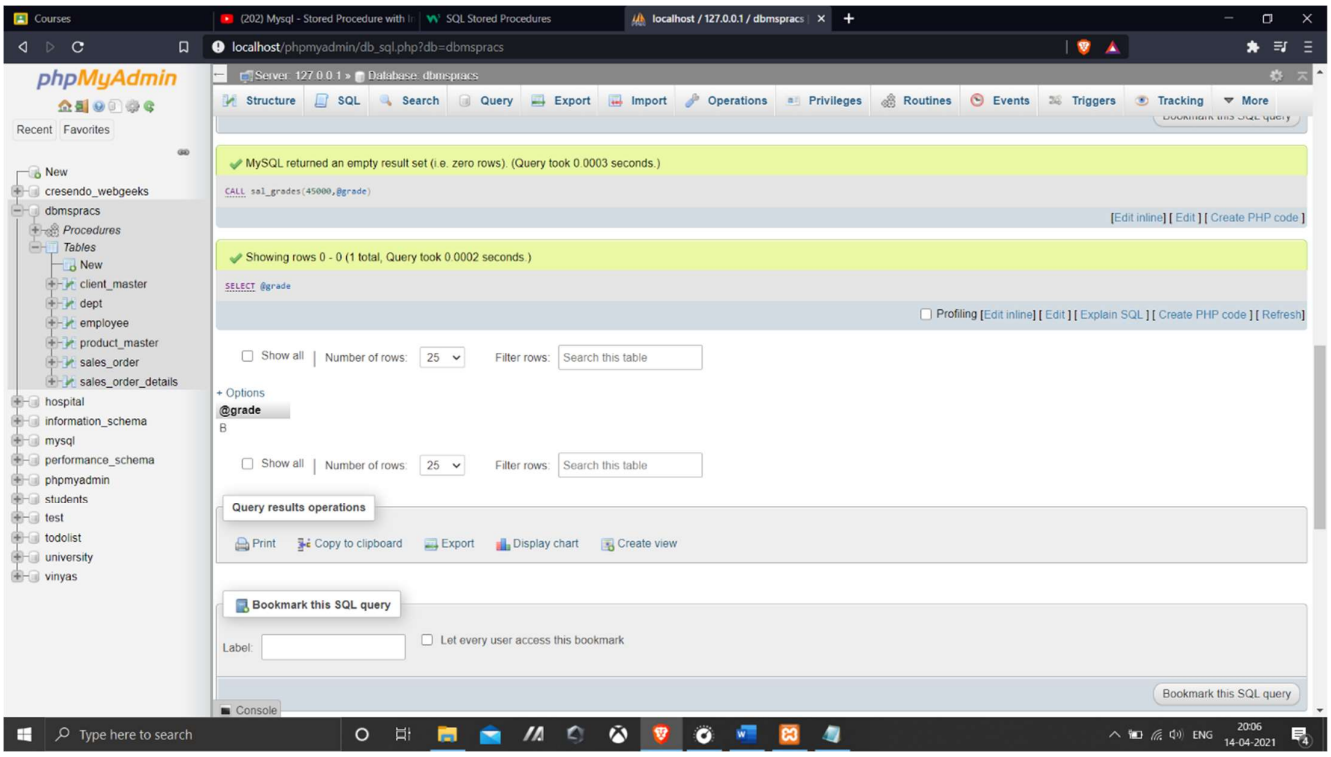
If salary < 30000 then grade is ‘C’:

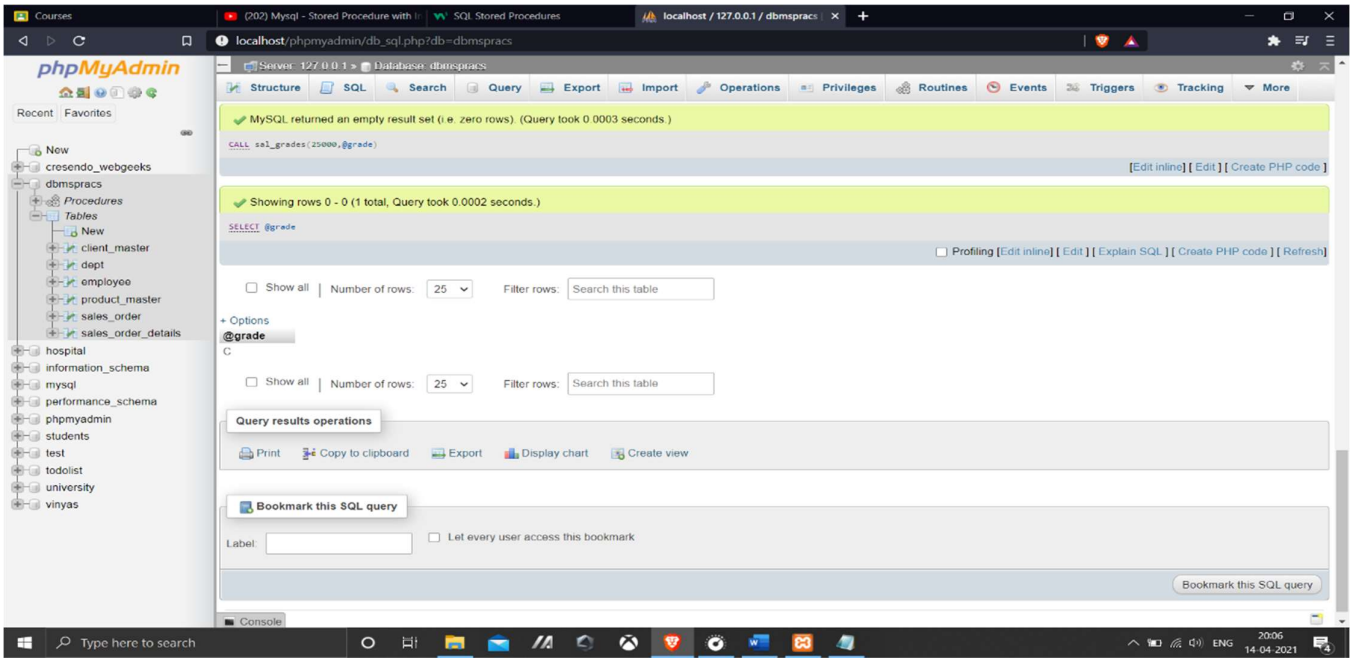
```
DELIMITER //
CREATE PROCEDURE sal_grades(IN salary int, OUT grade varchar(1))
BEGIN
        IF salary >= 50000 THEN
                set grade = 'A';
        elseif salary>= 30000 then
       set grade = 'B';
   else
       set  grade = 'C';
end if;
end //
DELIMITER ;

CALL sal_grades(75000,@grade);
SELECT @grade;
CALL sal_grades(45000,@grade);
SELECT @grade;
CALL sal_grades(25000,@grade);
SELECT @grade;
```


localhost/phpmyadmin/db_sql.php?db=dbmspracs
Server: 127.0.0.1 » Database: dbmspracs
Run SQL query/queries on database dbmspracs:
DELIMITER //
CREATE PROCEDURE sal_grades(IN salary int, OUT grade varchar(1))
BEGIN
IF salary >= 50000 THEN
set grade = 'A';
elseif salary>= 30000 then
set grade = 'B';
else
set grade = 'C';
end if;
end //
DELIMITER ;
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0068 seconds.)
CREATE PROCEDURE sal_grades(IN salary int, OUT grade varchar(1)) BEGIN IF salary >= 50000 THEN set grade = 'A'; elseif salary>= 30000 then set grade = 'B'; else set grade = 'C'; end if; end


localhost/phpmyadmin/db_sql.php?db=dbmspracs
Server: 127.0.0.1 » Database: dbmspracs
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)
CALL sal_grades(75000,@grade)
Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)
SELECT @grade
@grade
A
Query results operations

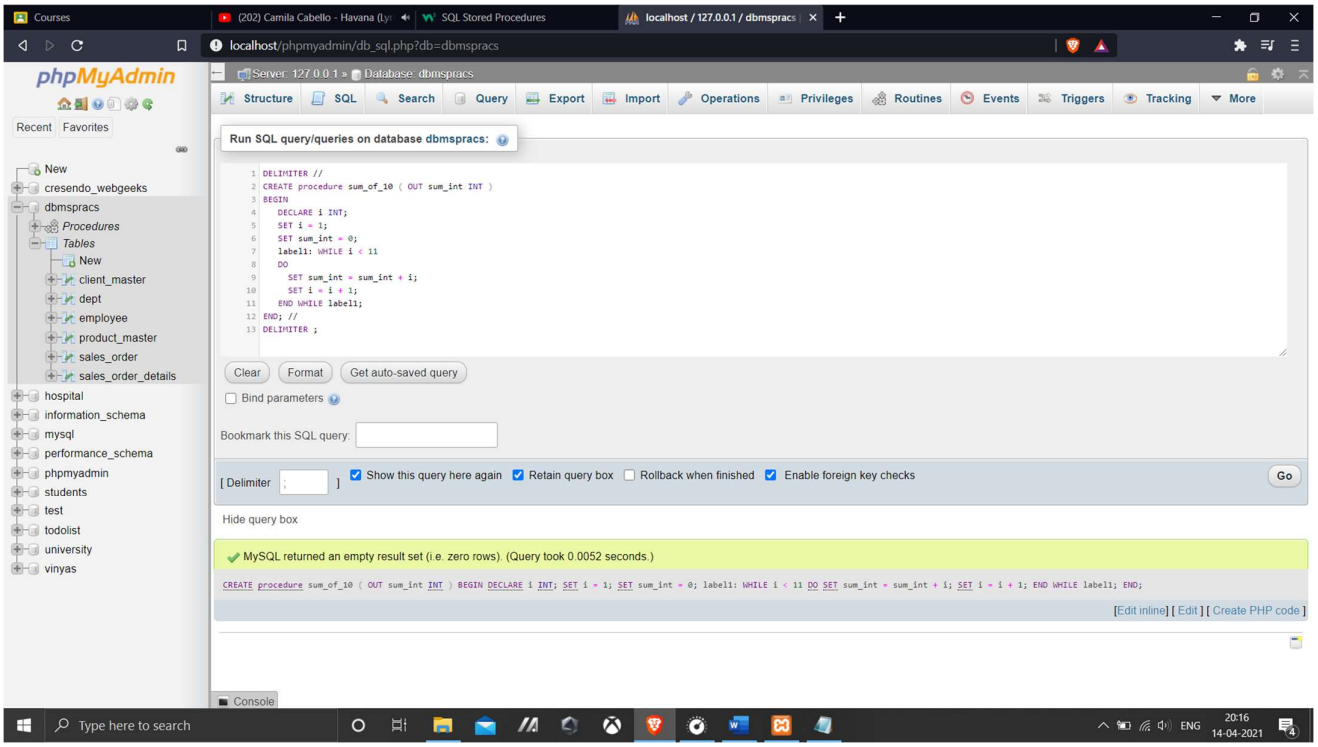

localhost/phpmyadmin/db_sql.php?db=dbmspracs
Server: 127.0.0.1 » Database: dbmspracs
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)
CALL sal_grades(45000,@grade)
[Edit inline] [ Edit ] [ Create PHP code ]
Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)
SELECT @grade
Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
+ Options
@grade
B
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Console

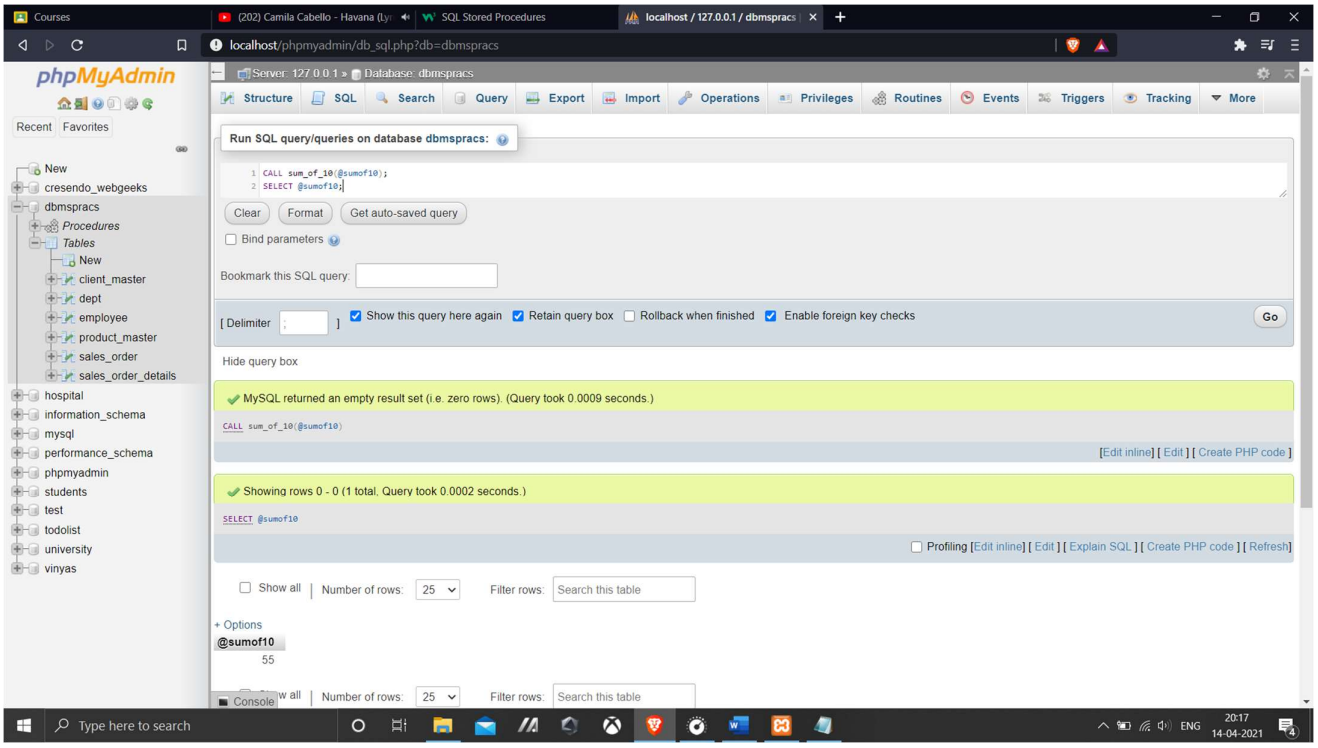

localhost/phpmyadmin/db_sql.php?db=dbmspracs
Server: 127.0.0.1 » Database: dbmspracs
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)
CALL sal_grades(25000,@grade)
[Edit inline] [ Edit ] [ Create PHP code ]
Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)
SELECT @grade
Profiling [Edit inline] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
+ Options
@grade
C
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Label:
Let every user access this bookmark
Console

2. To Create Table Employee: 2. Write a block to display sum of 1 to 10 numbers

```
DELIMITER //
CREATE procedure sum_of_10 ( OUT sum_int INT )
BEGIN
  DECLARE i INT;
  SET i = 1;
  SET sum_int = 0;
  label1: WHILE i < 11
  DO
        SET sum_int = sum_int + i;
    SET i = i + 1;
  END WHILE label1;
END; //
DELIMITER ;

CALL sum_of_10(@sumof10);
SELECT @sumof10;
```

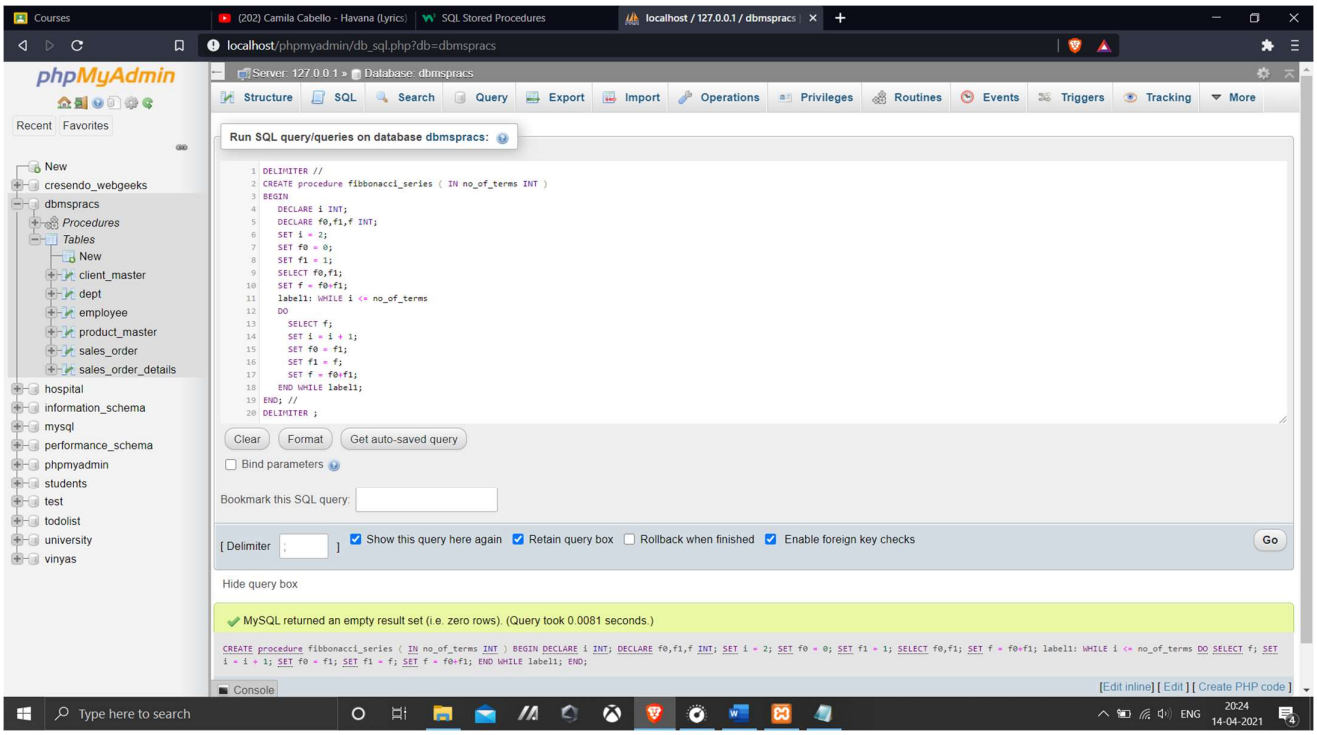

Run SQL query/queries on database dbmspracs:
DELIMITER //
CREATE procedure sum_of_10 ( OUT sum_int INT )
BEGIN
DECLARE i INT;
SET i = 1;
SET sum_int = 0;
label1: WHILE i < 11
DO
SET sum_int = sum_int + i;
SET i = i + 1;
END WHILE label1;
END; //
DELIMITER ;
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0052 seconds.)
CREATE procedure sum_of_10 ( OUT sum_int INT ) BEGIN DECLARE i INT; SET i = 1; SET sum_int = 0; label1: WHILE i < 11 DO SET sum_int = sum_int + i; SET i = i + 1; END WHILE label1; END;

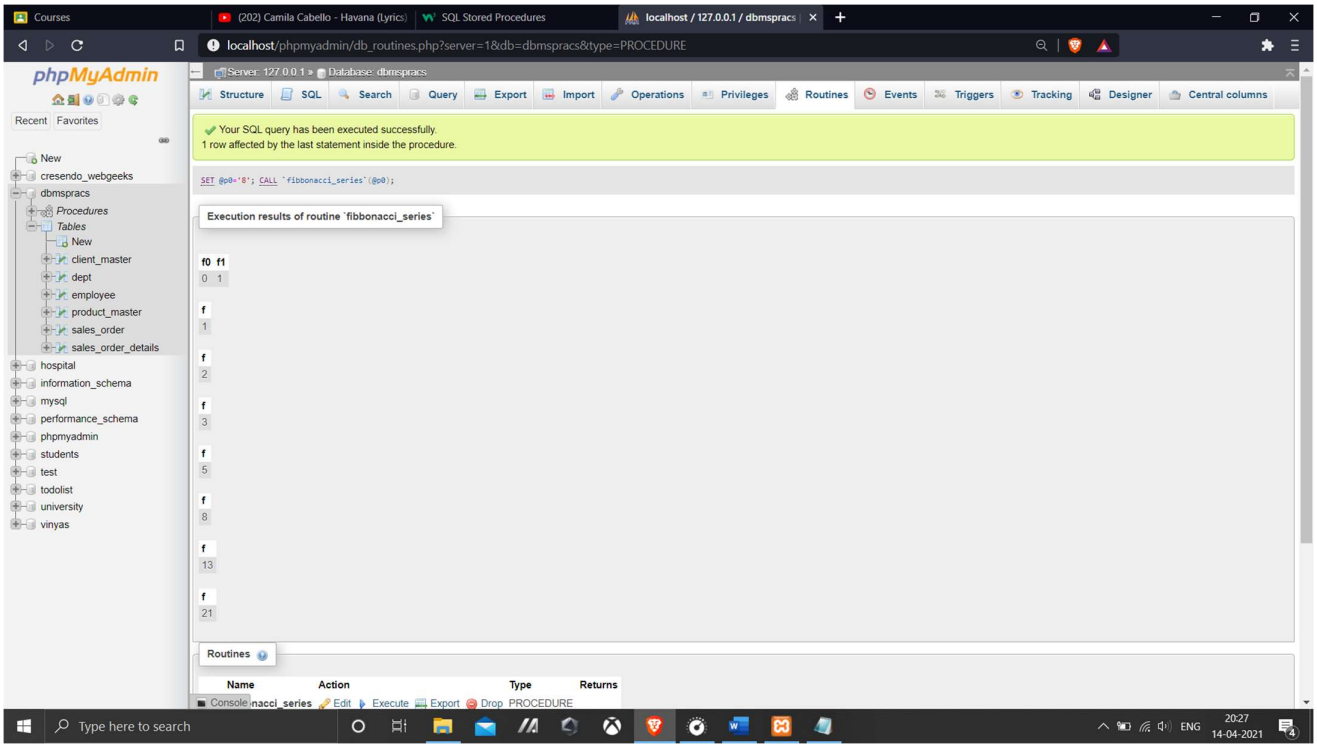

Run SQL query/queries on database dbmspracs:
CALL sum_of_10(@sumof10);
SELECT @sumof10;
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)
CALL sum_of_10(@sumof10)
Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)
SELECT @sumof10
@sumof10
55

3. Write a block to display Fibonacci series upto 8th term (start with 0,1)

```
DELIMITER //
CREATE procedure fibbonacci_series ( IN no_of_terms INT )
BEGIN
  DECLARE i INT;
  DECLARE f0,f1,f INT;
  SET i = 2;
  SET f0 = 0;
  SET f1 = 1;
  SELECT f0,f1;
  SET f = f0+f1;
  label1: WHILE i <= no_of_terms
  DO
        SELECT f;
    SET i = i + 1;
    SET f0 = f1;
    SET f1 = f;
    SET f = f0+f1;
  END WHILE label1;
END; //
DELIMITER ;

SET @p0='8';
CALL `fibbonacci_series`(@p0);
```

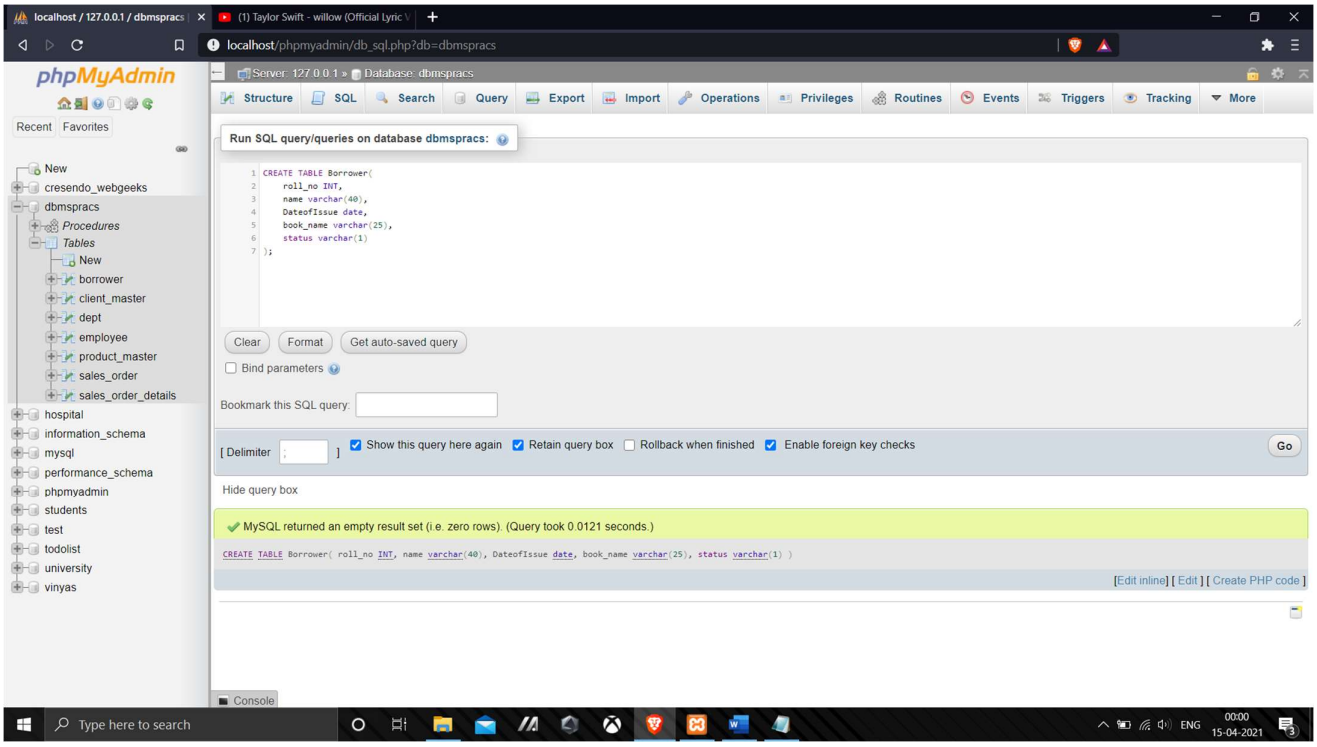

localhost/phpmyadmin/db_sql.php?db=dbmspracs
Server: 127.0.0.1 » Database: dbmspracs
Run SQL query/queries on database dbmspracs:
DELIMITER //
CREATE procedure fibbonacci_series ( IN no_of_terms INT )
BEGIN
DECLARE i INT;
DECLARE f0,f1,f INT;
SET i = 2;
SET f0 = 0;
SET f1 = 1;
SELECT f0,f1;
SET f = f0+f1;
label1: WHILE i <= no_of_terms
DO
SELECT f;
SET i = i + 1;
SET f0 = f1;
SET f1 = f;
SET f = f0+f1;
END WHILE label1;
END; //
DELIMITER ;
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0081 seconds.)


localhost/phpmyadmin/db_routines.php?server=1&db=dbmspracs&type=PROCEDURE
Server: 127.0.0.1 » Database: dbmspracs
Your SQL query has been executed successfully.
1 row affected by the last statement inside the procedure.
SET @p0='8'; CALL `fibbonacci_series`(@p0);
Execution results of routine `fibbonacci_series`
f0 f1
0 1
f
1
f
2
f
3
f
5
f
8
f
13
f
21
Routines

Task 2

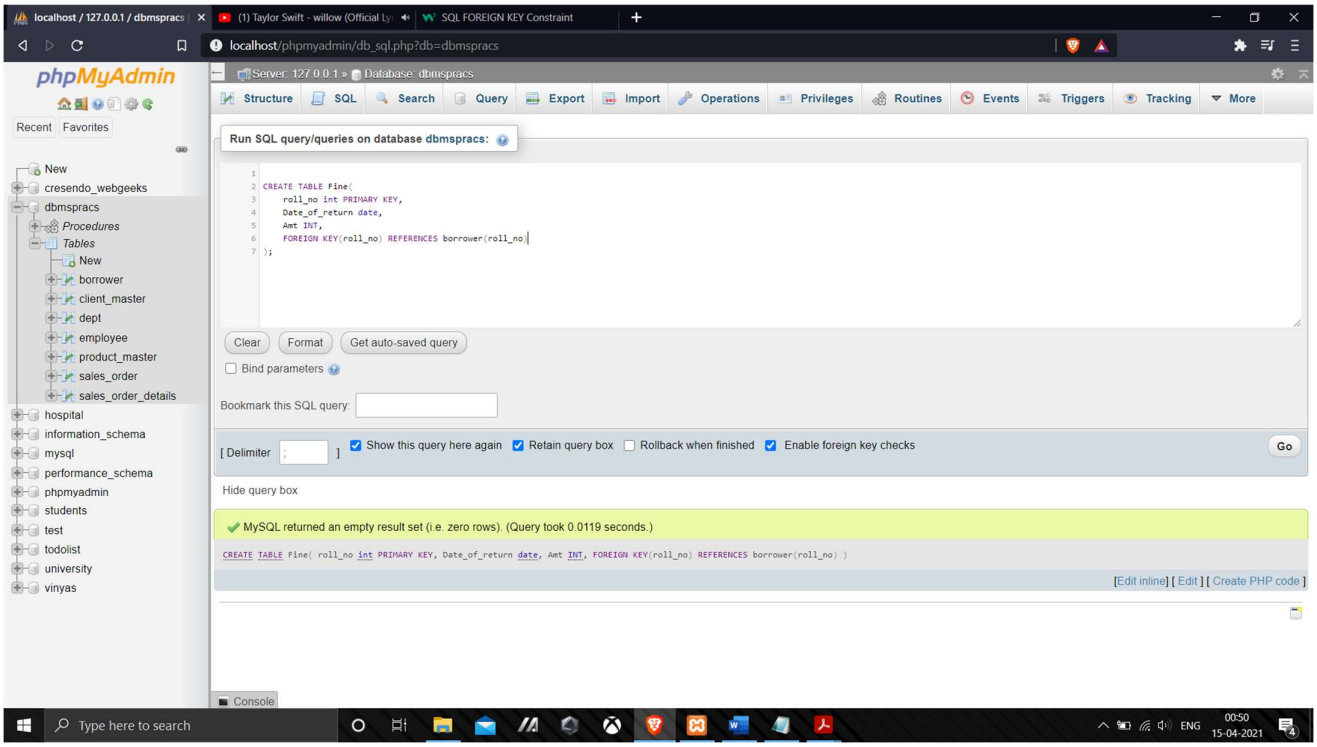
## 1. Creating Table Borrower:

```
CREATE TABLE Borrower(
    roll_no INT PRIMARY KEY,
    name varchar(40),
    DateofIssue date,
    book_name varchar(25),
    status varchar(1)
);
```



## 2. Creating Table fine:

```
CREATE TABLE Fine(
    roll_no int PRIMARY KEY,
    Date_of_return date,
    Amt INT,
    FOREIGN KEY(roll_no) REFERENCES borrower(roll_no)
);
```



## 3. Inserting the necessary values:

INSERT INTO `borrower` (`roll_no`, `name`, `DateofIssue`, `book_name`, `status`) VALUES ('1', 'Amrita', '2021-04-01', 'Java', 'I');

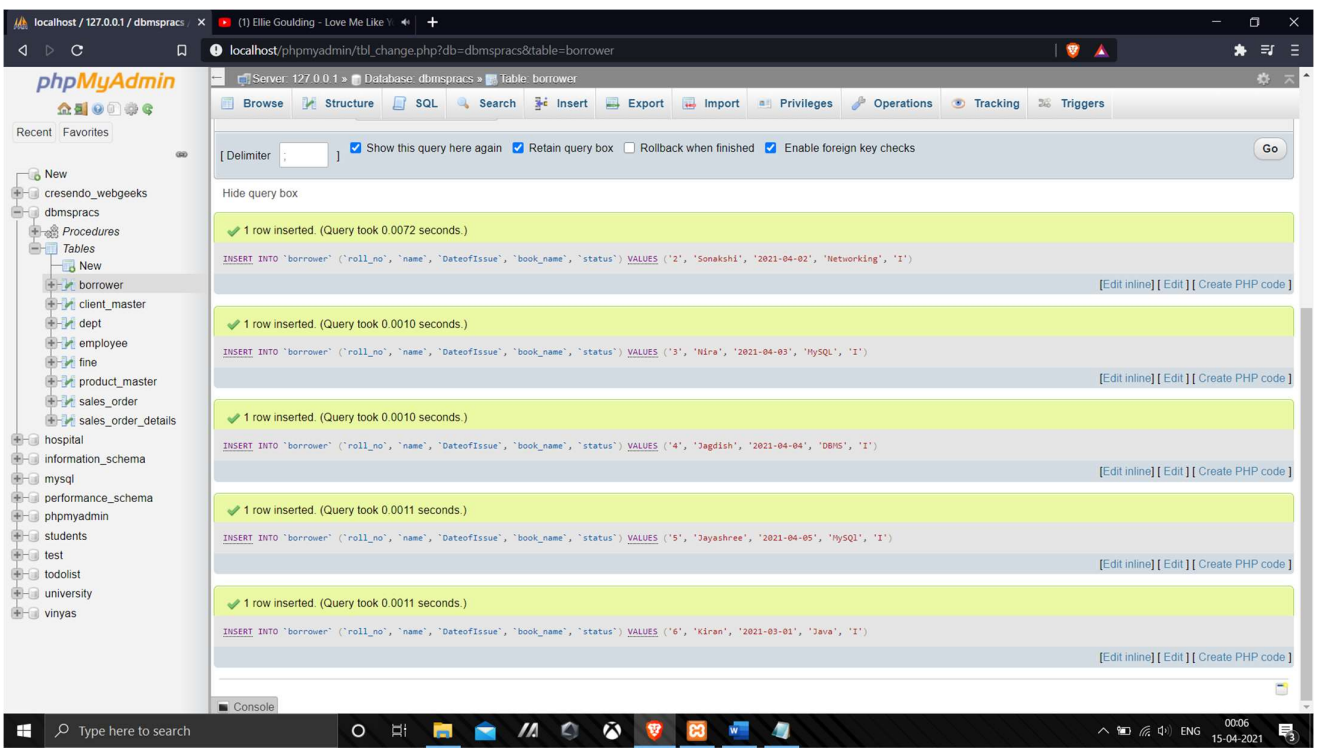
INSERT INTO `borrower` (`roll_no`, `name`, `DateofIssue`, `book_name`, `status`) VALUES ('2', 'Sonakshi', '2021-04-02', 'Networking', 'I');

INSERT INTO `borrower` (`roll_no`, `name`, `DateofIssue`, `book_name`, `status`) VALUES ('3', 'Nira', '2021-04-03', 'MySQL', 'I');

INSERT INTO `borrower` (`roll_no`, `name`, `DateofIssue`, `book_name`, `status`) VALUES ('4', 'Jagdish', '2021-04-04', 'DBMS', 'I');

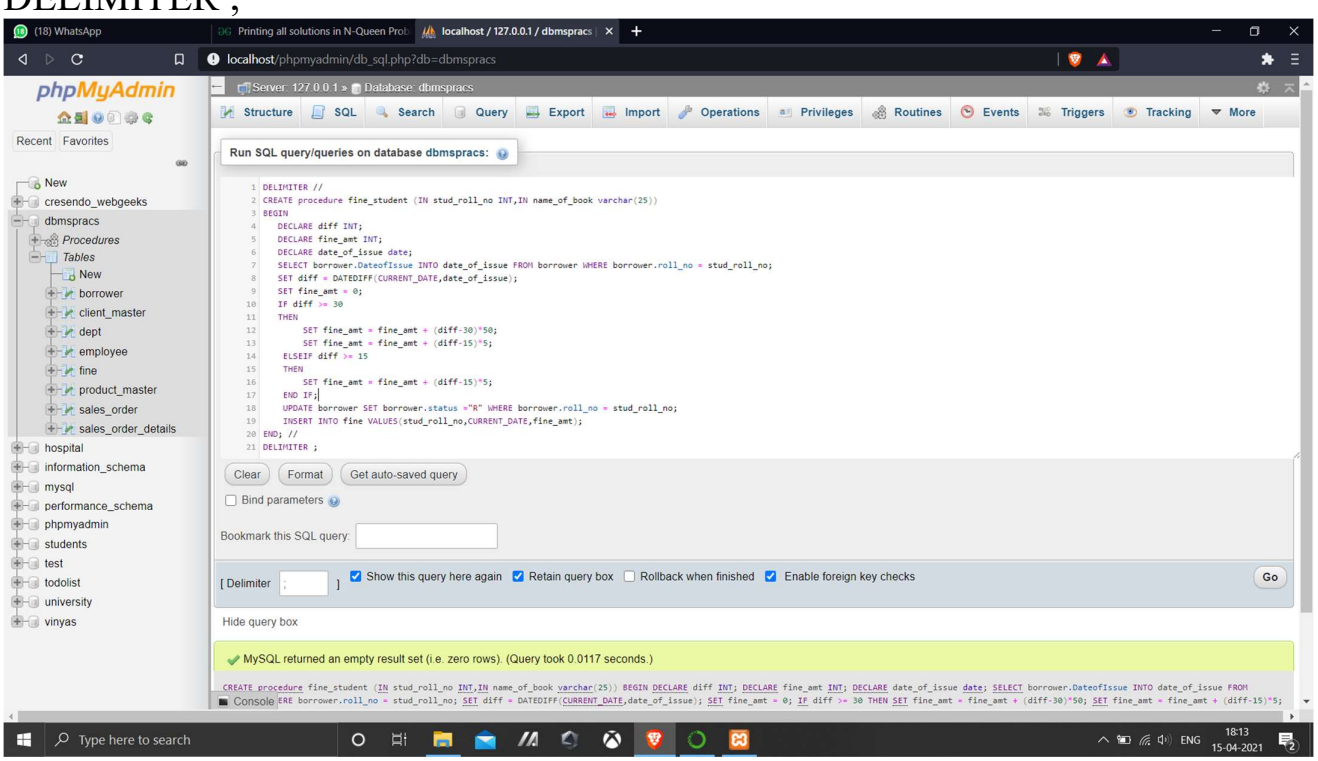
INSERT INTO `borrower` (`roll_no`, `name`, `DateofIssue`, `book_name`, `status`) VALUES ('5', 'Jayashree', '2021-04-05', 'MySQl', 'I');

INSERT INTO `borrower` (`roll_no`, `name`, `DateofIssue`, `book_name`, `status`) VALUES ('6', 'Kiran', '2021-03-01', 'Java', 'I');

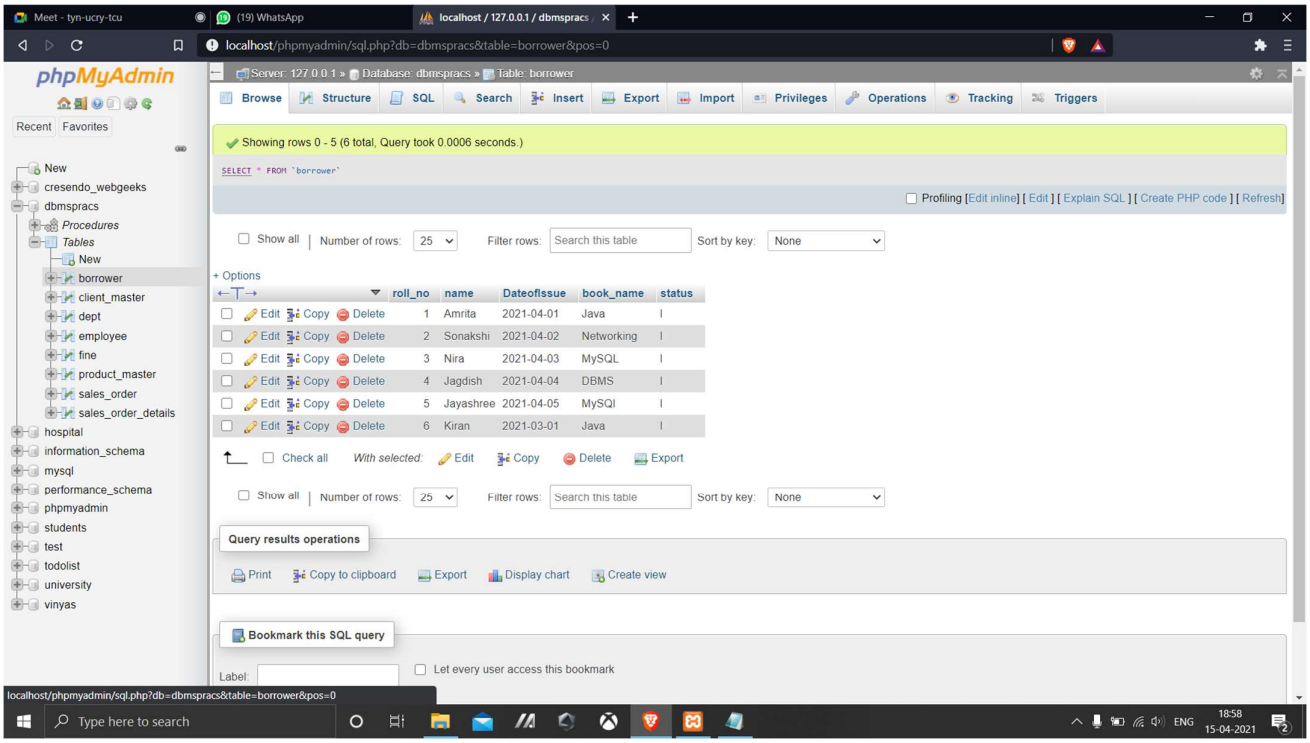


## 4. Creating the Procedure:

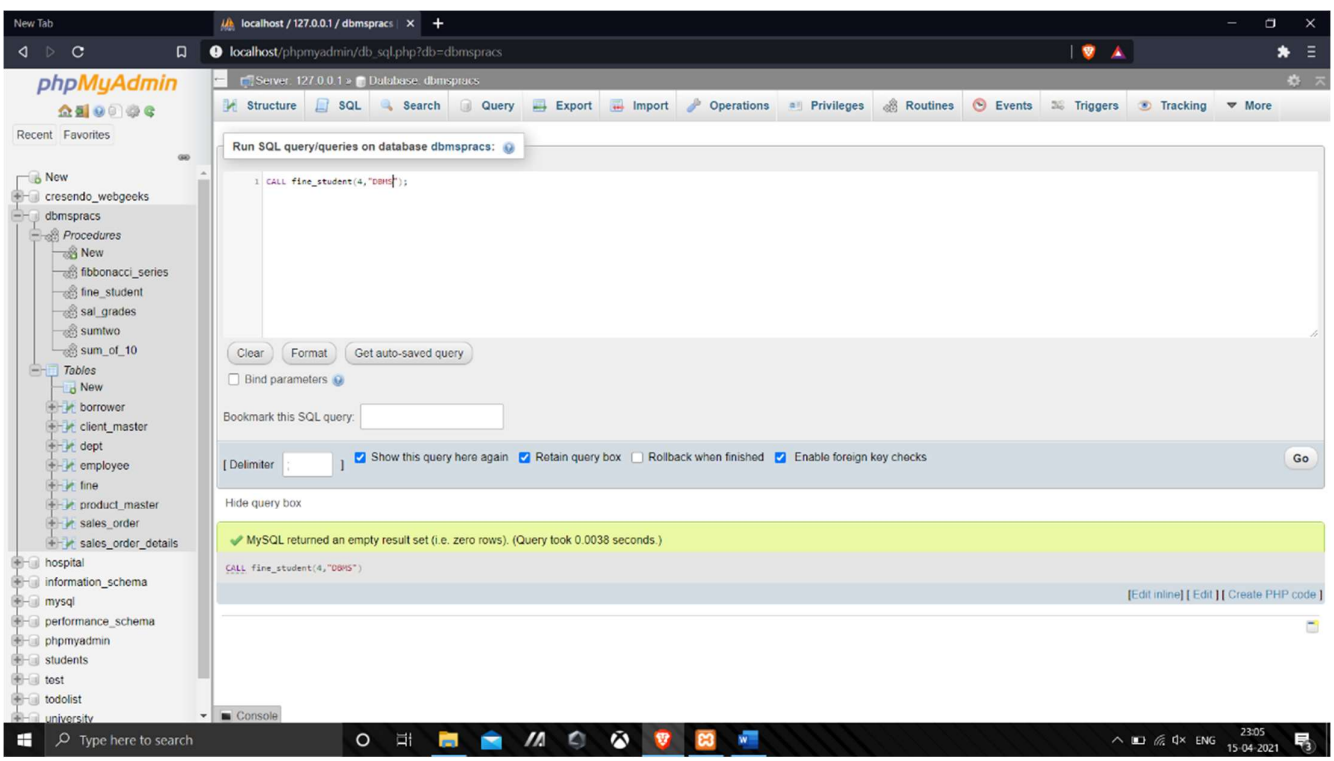
```
DELIMITER //
CREATE procedure fine_student (IN stud_roll_no INT,IN name_of_book
varchar(25))
BEGIN
  DECLARE diff INT;
  DECLARE fine_amt INT;
  DECLARE date_of_issue date;
  SELECT borrower.DateofIssue INTO date_of_issue FROM borrower WHERE
borrower.roll_no = stud_roll_no;
  SET diff = DATEDIFF(CURRENT_DATE,date_of_issue);
  SET fine_amt = 0;
  IF diff >= 30
  THEN
            SET fine_amt = fine_amt + (diff-30)*50;
     SET fine_amt = fine_amt + (diff-15)*5;
   ELSEIF diff >= 15
   THEN
      SET fine_amt = fine_amt + (diff-15)*5;
   END IF;
   UPDATE borrower SET borrower.status ="R" WHERE borrower.roll_no =
stud_roll_no;
   INSERT INTO fine VALUES(stud_roll_no,CURRENT_DATE,fine_amt);
END; //
DELIMITER ;
```



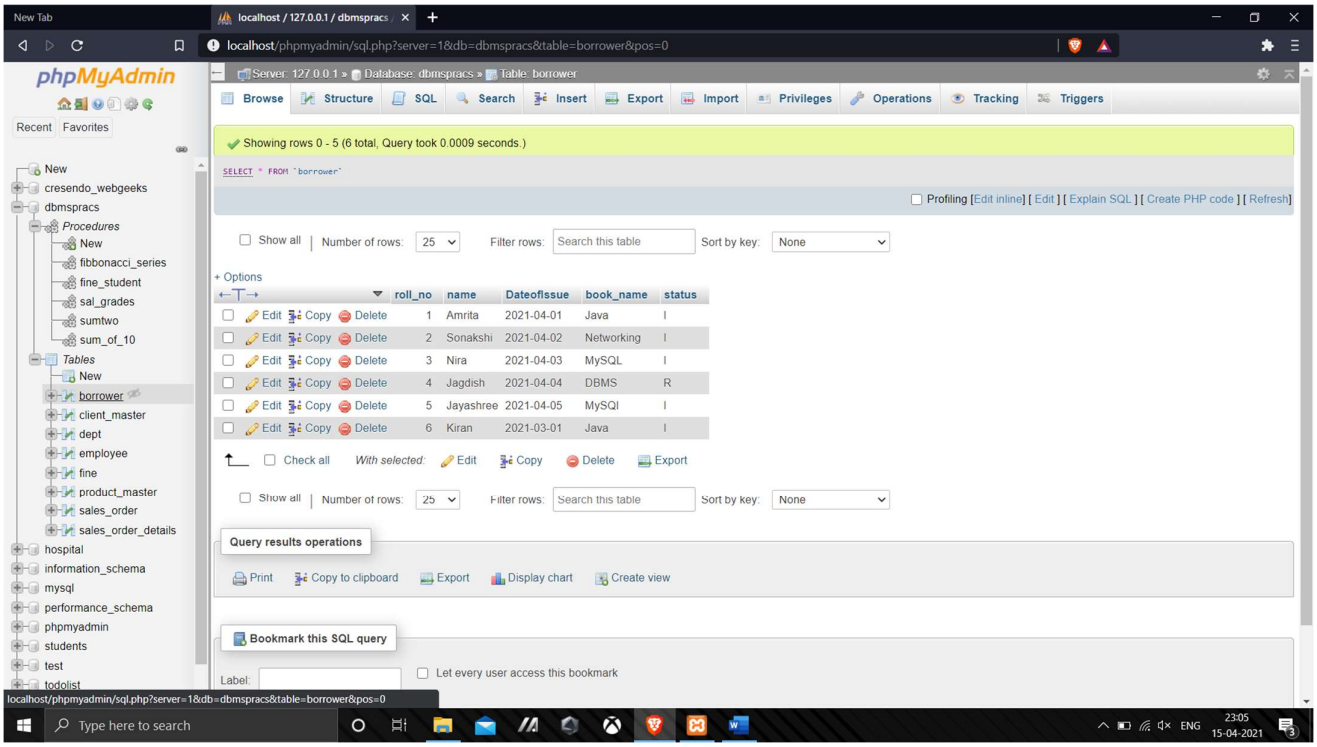
Before Calling fine_student procedure:

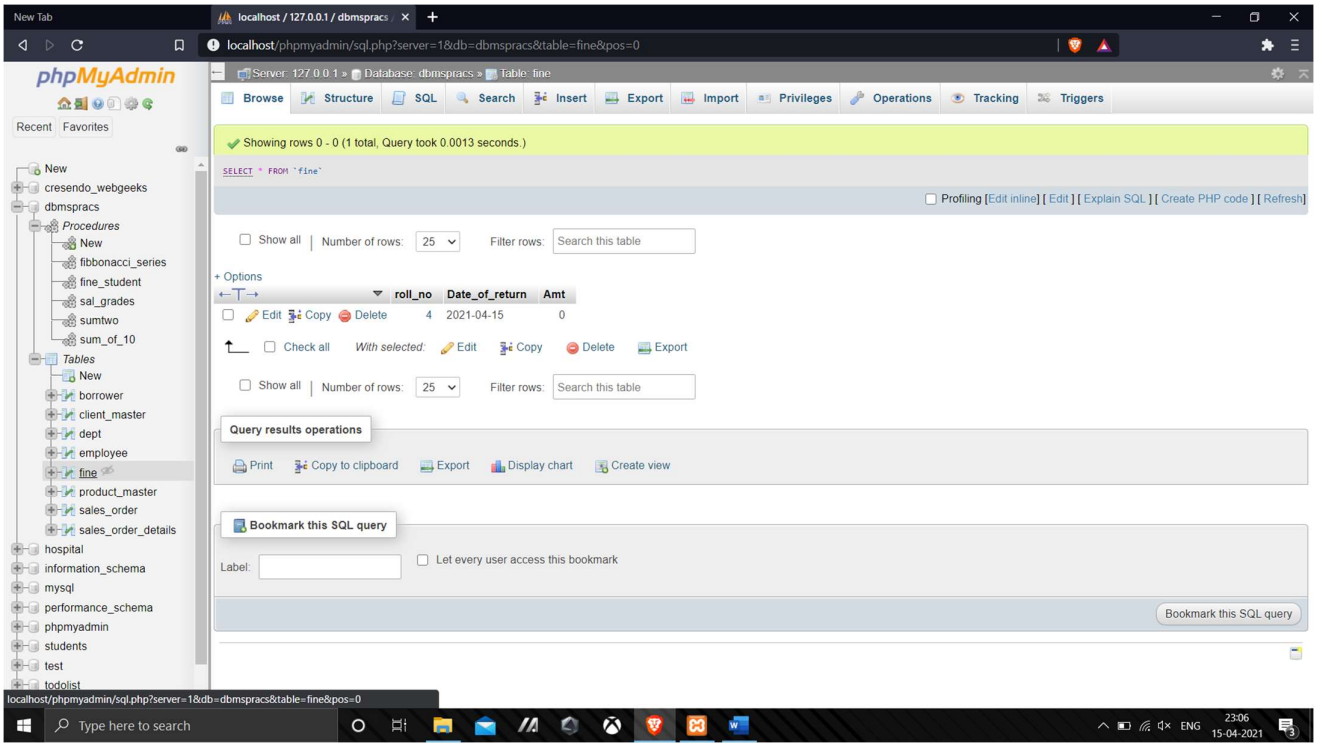



Calling fine_student procedure:



After Calling fine_student procedure:



Brendon Lucas
8953
BE Comps B

Exp - 8 - Postlab

(1) Advantages of PL SQL vs SQL

Ans

(a) PL/SQL is a database programming language using SQL, while SQL is only database query language.

(b) PL/SQL allows _____, declaration of data variable while SQL does not.

(c) Control structures are available like, Forloop, while loop whereas SQL does not support Control structures.

(d) PL/SQL block performs Group of Operations as single block whereas SQL performs Single Query operations

(e) It does not interacts directly with the database server and PLSQL is application oriented Language in Contrast SQL is Data oriented language and It directly interacts with database server.

(f) PL/SQL is accustomed write program blocks, functions, procedures, triggers and packages. in Contrast SQL is used to write queries, DDL and DML Statements.

Q2 Explain data types of PL/SQL

Ans

(1) NUMERIC DATATYPES:-

(a) INT

ANSI specific integer type with maximum precision of 38 decimal digits

(b) FLOAT.

ANSI and IBM specific floating point type with maximum precision of 126 binary digits.

(2) CHARACTER DATATYPES:-

Varchar2 :- Variable-length character string with maximum size of 32,767 bytes

(3) DATETIME and INTERVAL DATATYPES:-

DAY

01 to 31 (valid by month & Year).

MONTH

01 to 12

YEAR

-4712 to 9999

LARGE OBJECT (LOB) DATATYPES

BLOB

Used to store large binary object upto 128 TB.