**PROGRAM -1**

**AIM:** To create a database containing table employee with employee details.Write PLSQL to update the experience level of employee as beginner,intermediate and advanced.

**CODE:**

create database company;

use company;

create table emp(emp\_id int primary key,emp\_name varchar(20),salary varchar(20));

create table dept(dept\_id int primary key,emp\_id int,designation varchar(20),experience int(10) ,foreign key(emp\_id) references emp(emp\_id));

insert into emp(emp\_id,emp\_name,salary)values(101,'Shibu',25000);

insert into emp(emp\_id,emp\_name,salary)values(102,'Raju',35000);

insert into emp(emp\_id,emp\_name,salary)values(103,'Shanku',50000);

select \* from emp;

insert into dept(dept\_id,emp\_id,designation,experience)values(201,101,'Peon',2);

insert into dept(dept\_id,emp\_id,designation,experience)values(202,102,'Clerk',6);

insert into dept(dept\_id,emp\_id,designation,experience)values(203,103,'Manager',12);

select \* from dept;

create table level(emp\_id int,dept\_id int,experience\_level varchar(20),foreign key(emp\_id) references emp(emp\_id),foreign key(dept\_id) references dept(dept\_id));

call exp(2,101,201);

call exp(6,102,201);

call exp(12,103,203);

select \* from level;

select emp.emp\_name,emp.salary,new\_salary(level.experience\_level,emp.salary) from emp,level where emp.emp\_id=level.emp\_id;

////STORED PROCEDURE

CREATE DEFINER=`root`@`localhost` PROCEDURE `exp`(experience int,emp\_id int,dept\_id int)

BEGIN

DECLARE

levels varchar(45);

if (experience > 0 && experience<5)

then set levels = 'beginner';

insert into employe(emp\_id,experience,salary,levels) values(emp\_id,experience,salary,levels);

end if;

if( exp>=6 && exp <10)

then set levels = 'intermediate';

insert into employe(emp\_id,experience,salary,levels) values(emp\_id,experience,salary,levels);

end if;

if (exp >= 10)

then set levels = 'Experienced';

insert into employe(emp\_id,experience,salary,levels) values(emp\_id,experience,salary,levels);

end if;

END

////FUNCTION////

CREATE DEFINER=`root`@`localhost` FUNCTION `new\_salary`(experience\_level varchar(20),sal varchar(10)) RETURNS int(11)

BEGIN

if(experience\_level = 'Experienced')

then

return(sal+1000);

else

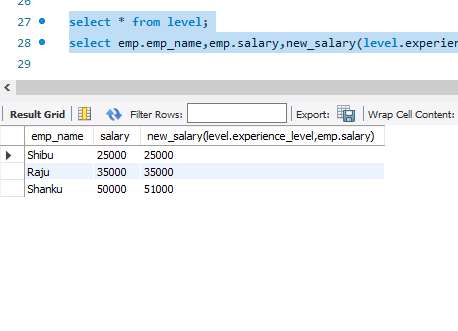
return(sal);

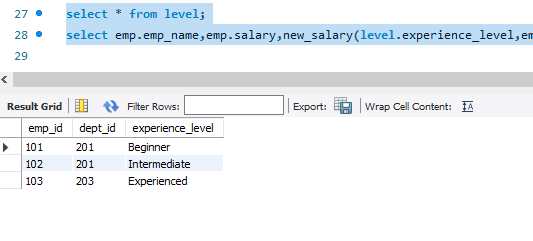
end if;

RETURN 1;

END

**OUTPUT:**

****

****

**PROGRAM -2**

**AIM:** Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation

**CODE:**

create database studentdb;

use studentdb;

create table T2(rollno int,name varchar(10),primary key(rollno));

call stud('101','Stefi');

call stud('102','Sreelaya');

select \* from T2;

///STORED PROCEDURE///

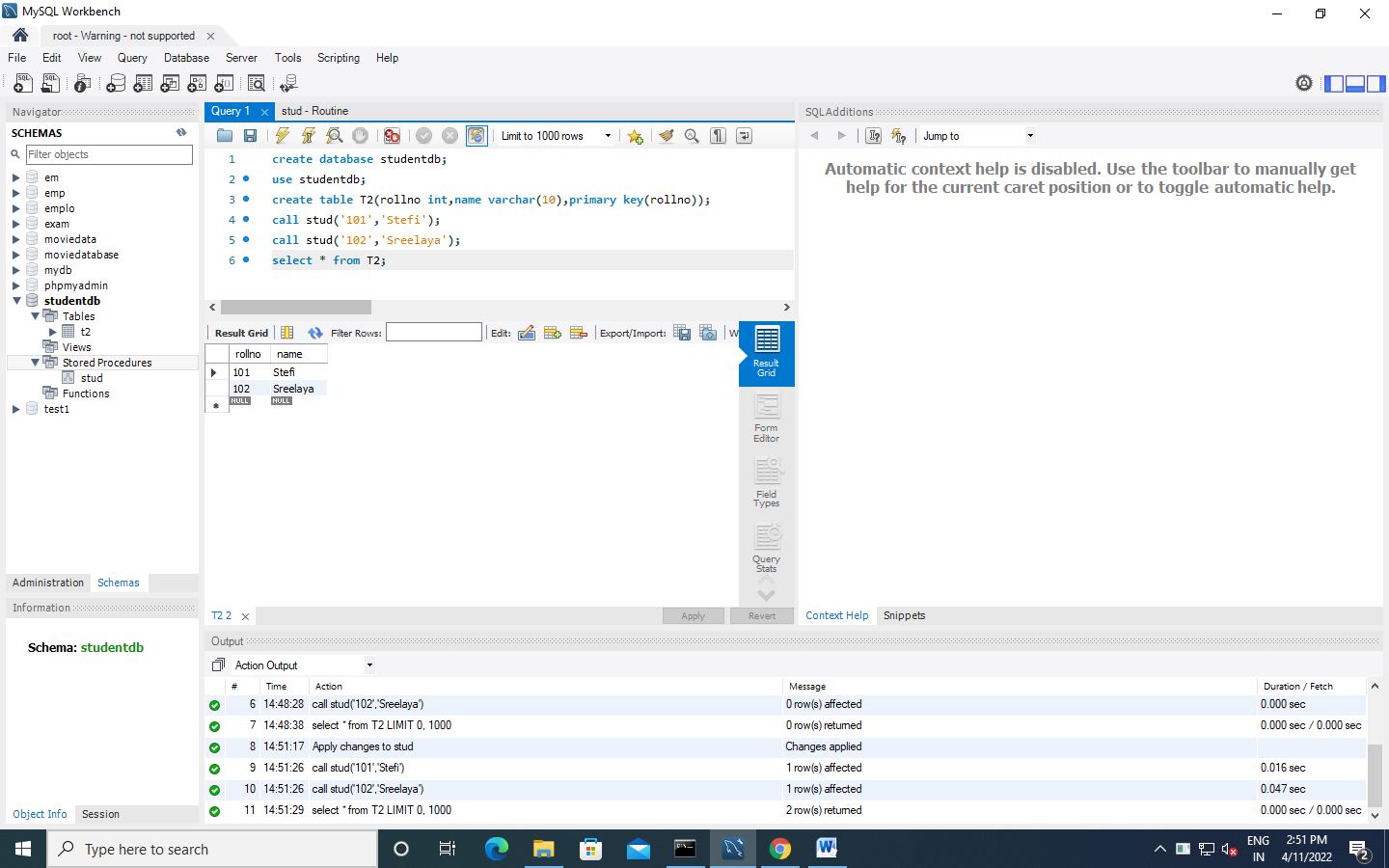
CREATE DEFINER=`root`@`localhost` PROCEDURE `stud`(rollno int,name varchar(10))

BEGIN

insert into T2 values(rollno,name);

END

**OUTPUT:**



**PROGRAM-3**

**AIM:** To write a PL/SQL block to calculate the incentive of an employee whose ID is 110

**CODE:**

create database employeedb2;

use employeedb2;

create table E1(empid int,empname varchar(10),salary int,primary key(empid));

insert into E1(empid,empname,salary)values('110','Stefi',2000);

insert into E1(empid,empname,salary)values('111','Sreelaya',50000);

select \* from E1;

select empid,empname,insentive(empid) from E1;

///FUNCTION///

CREATE DEFINER=`root`@`localhost` FUNCTION `insentive`(empid int) RETURNS varchar(20) CHARSET latin1

BEGIN

DECLARE i VARCHAR(20);

IF (empid=110)

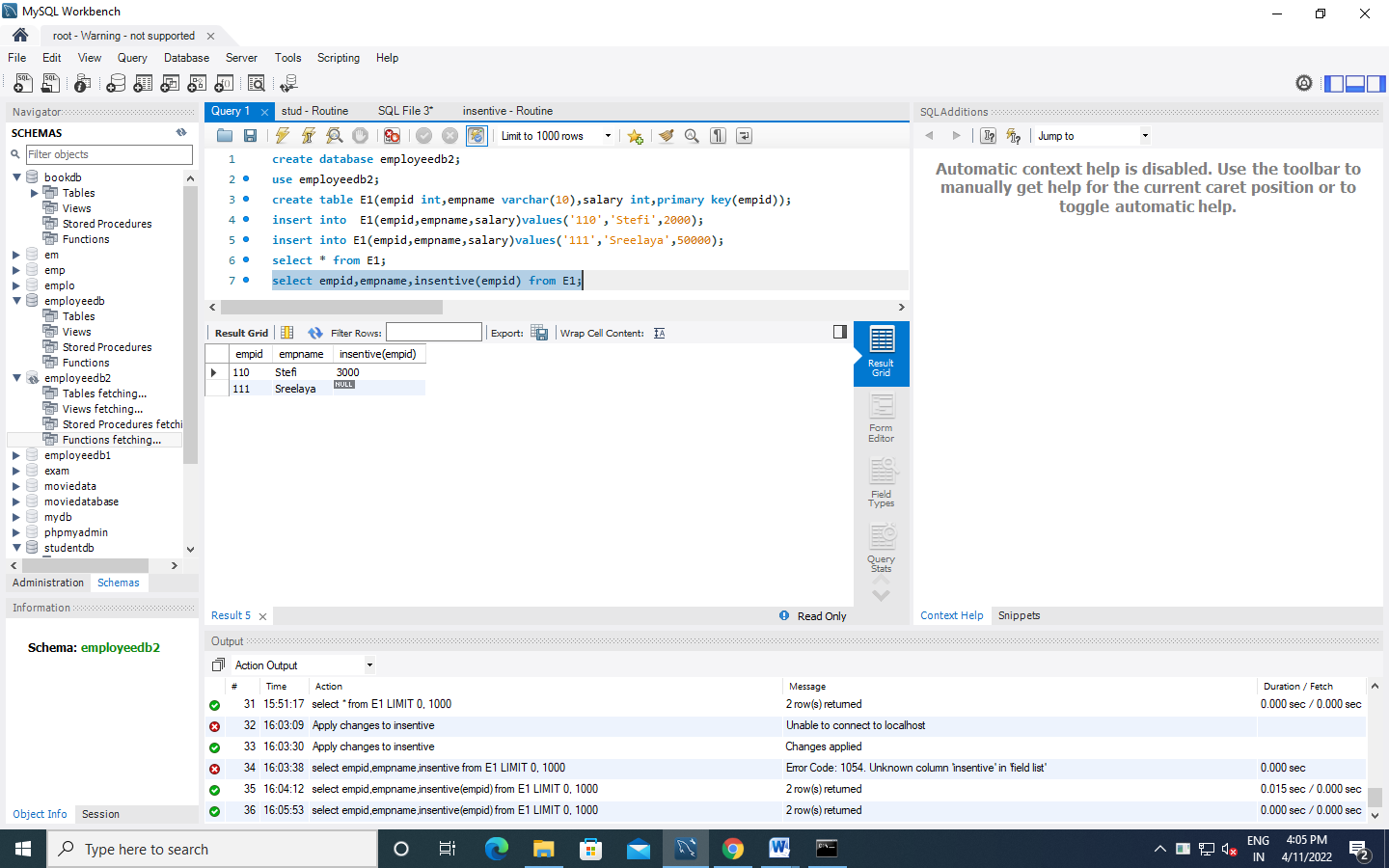
THEN SET i=3000;

END IF;

RETURN i ;

END

**OUTPUT:**



**PROGRAM-4**

**AIM:** To create the Book database and do the following: (Consider the attributes based on the question given)

book(book\_name, author\_name, price,quantity)

1. Write a query to update the quantity by double in the table book.
2. List all the book\_name whose price is greater than those of book named "Database for Dummies"
3. Retrieve the list of author\_name whose first letter is ’a’ along with the book\_name and price (Explore more about *Like* keyword)
4. Write a PL/SQL Procedure to find the total number of books of same author

**CODE:**

create database books;

use books;

create table book\_info(book\_name varchar (20),author varchar(20),price int,quantity int);

insert into book\_info values('randamoozham','MT',300,5);

insert into book\_info values('ikigai','hector',500,7);

insert into book\_info values('databse of dummies','xyz',250,7);

insert into book\_info values('wings of flare','APJ',500,7);

insert into book\_info values('oopol','MT',270,3);

select \* from book\_info;

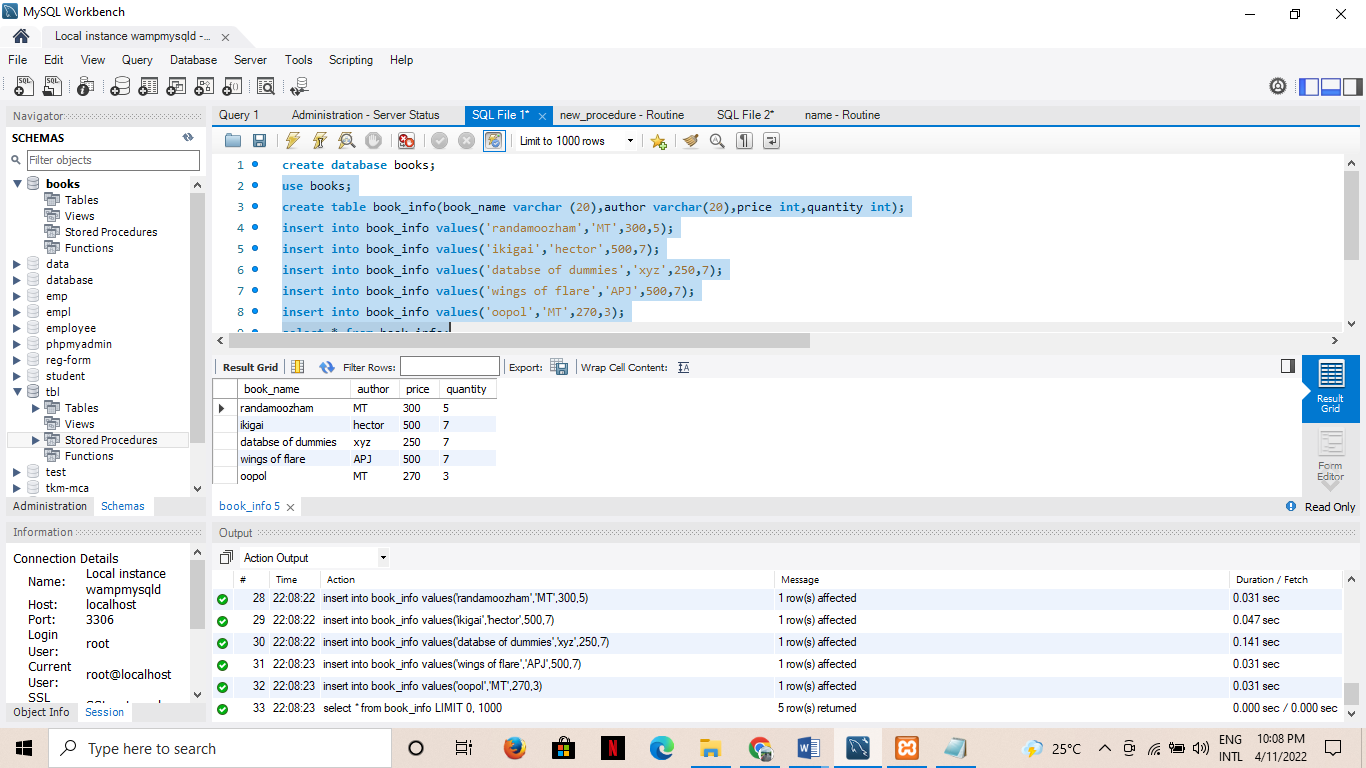
**a)** set sql\_safe\_updates=0;

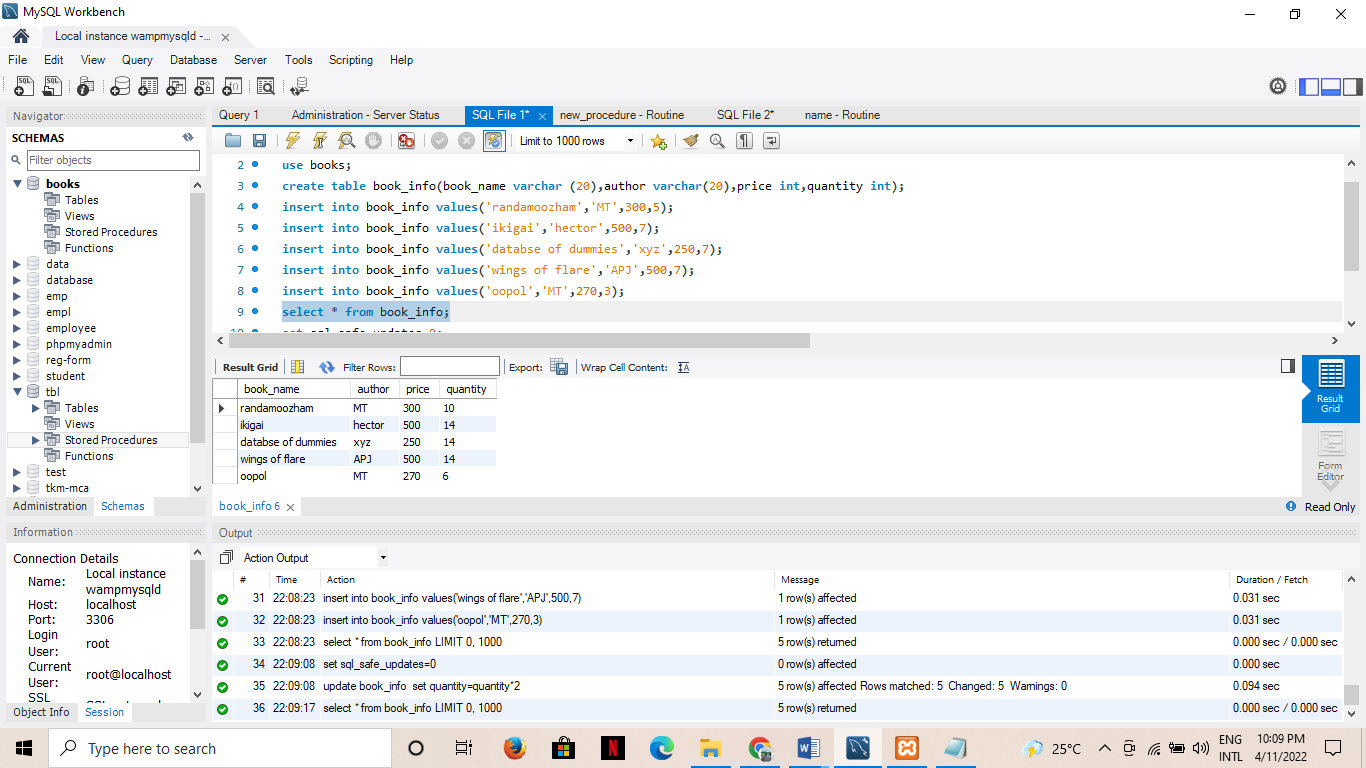
update book\_info set quantity=quantity\*2;

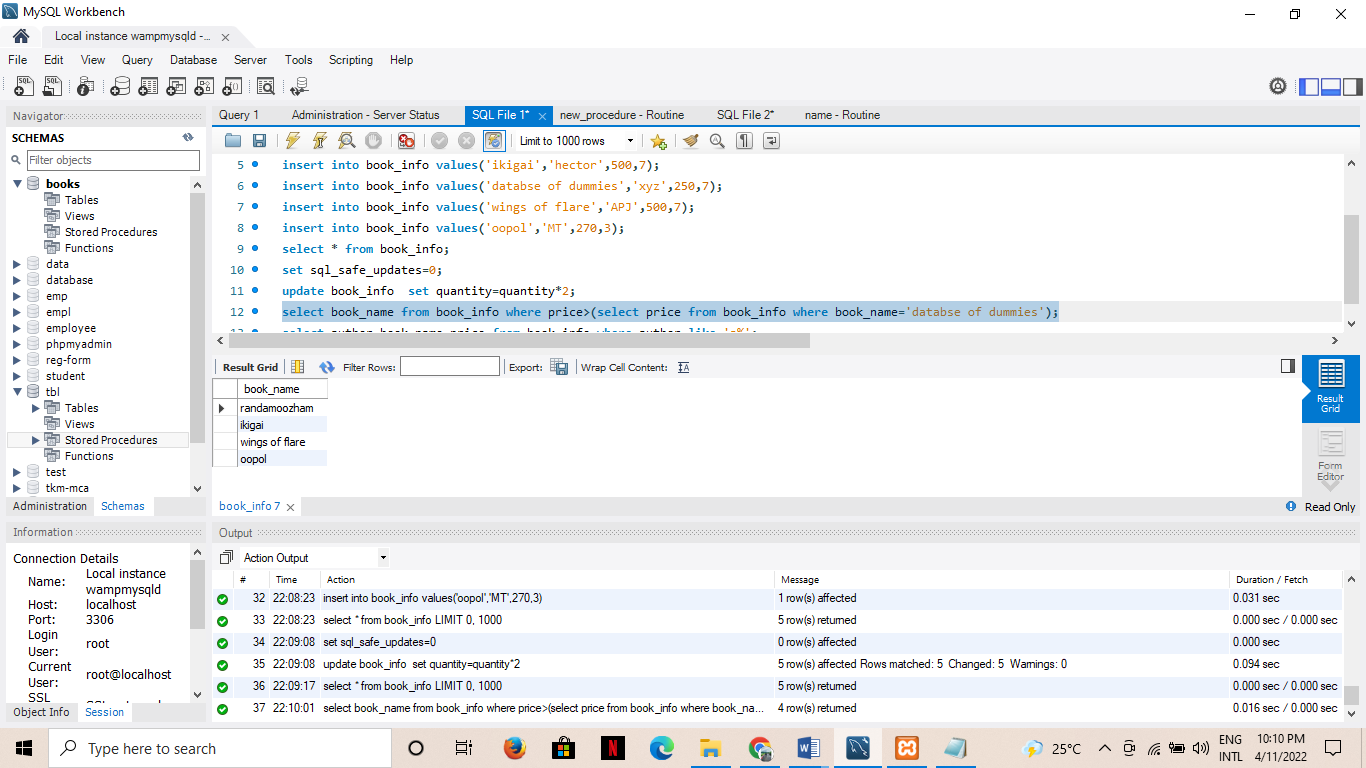
**b)** select book\_name from book\_info where price>(select price from book\_info where book\_name='databse of dummies');

**c)** select author,book\_name,price from book\_info where author like 'a%';

**OUTPUT:**

****

****

****

**PROGRAM-5**

**AIM:** Create the Company database with the following tables and do the following:

Administration (employee\_salary, development \_cost, fund\_ amount, turn\_over,bonus)

Emp\_details (emp\_no, emp\_name, DOB, address, doj, mobile\_no, dept\_no, salary).

1. Calculate the total and average salary amount of the employees of each department.
2. Display total salary spent for employees.
3. Develop a PL/SQL function to display total fund\_amount spent by the administration department

**CODE:**

create database company;

use company;

CREATE TABLE Admins(

emp\_sal double,

dvlp\_cost double,

fund\_amount double,

turn\_over double,

bonus double);

CREATE TABLE Emp\_details(

emp\_no int,

emp\_name varchar(20),

DOB date,

address varchar(20),

doj date,

mobile\_no int8,

dept\_no int,

salary double);

INSERT INTO Admins VALUES

(12000,25000,560000,65000,5000),

(70000,55000,860000,15000,1000),

(18000,45000,160000,75000,7000),

(10000,27000,520000,60000,5000),

(18000,27000,360000,35000,3000);

INSERT INTO Emp\_details VALUES

(1,"hamna","1999-10-10","Street - 2 xyz","2020-10-10",9865986598,10,12000),

(2,"ansi","1997-10-10","Street - 2 abc","2020-10-10",9865986598,10,12200),

(3,"sree","1996-10-10","Street ","2020-10-10",9865986598,11,12500),

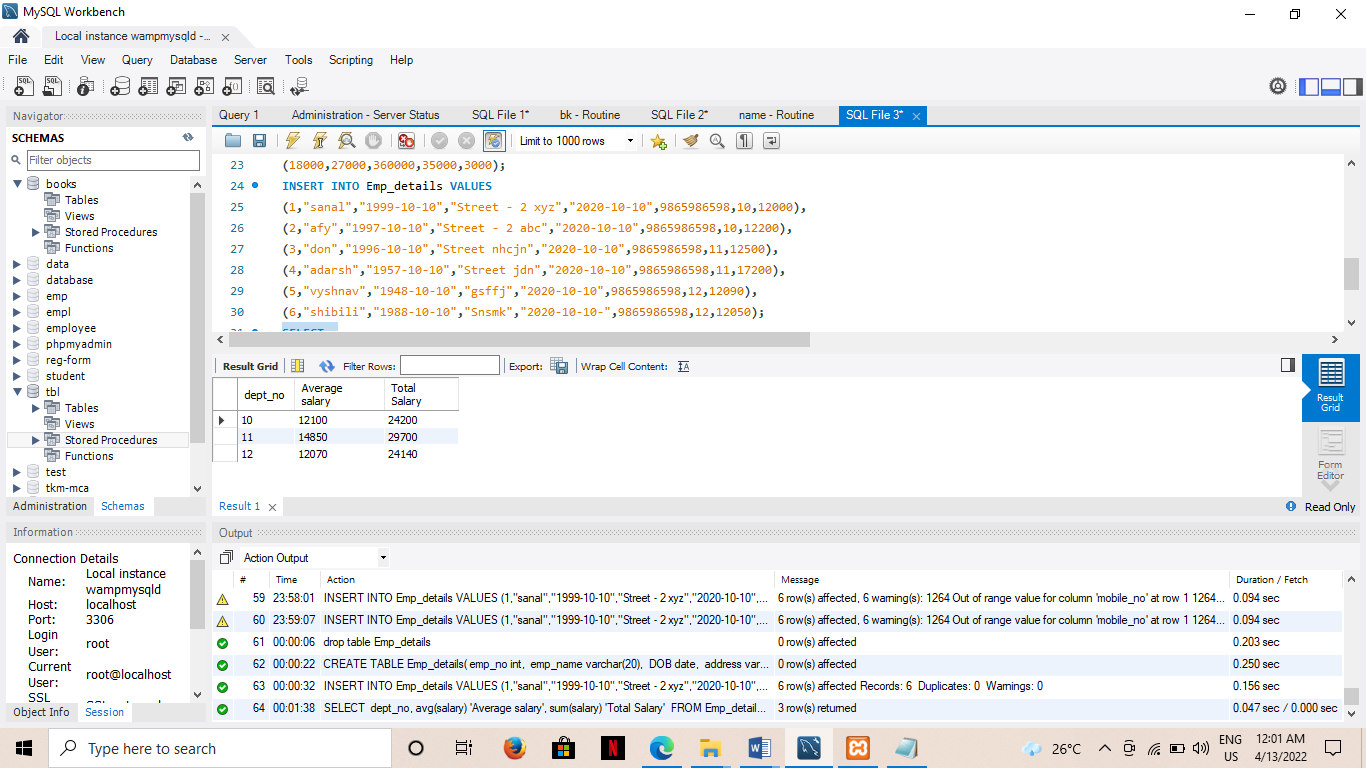
(4,"stef","1957-10-10","Street in","2020-10-10",9865986598,11,17200),

(5,"anu","1948-10-10","gared","2020-10-10",9865986598,12,12090),

(6,"shiva","1988-10-10","Sas","2020-10-10-",9865986598,12,12050);

**a)** SELECT dept\_no,avg(salary) 'Average salary',sum(salary) 'Total Salary' FROM Emp\_details GROUP BY dept\_no;

**OUTPUT**

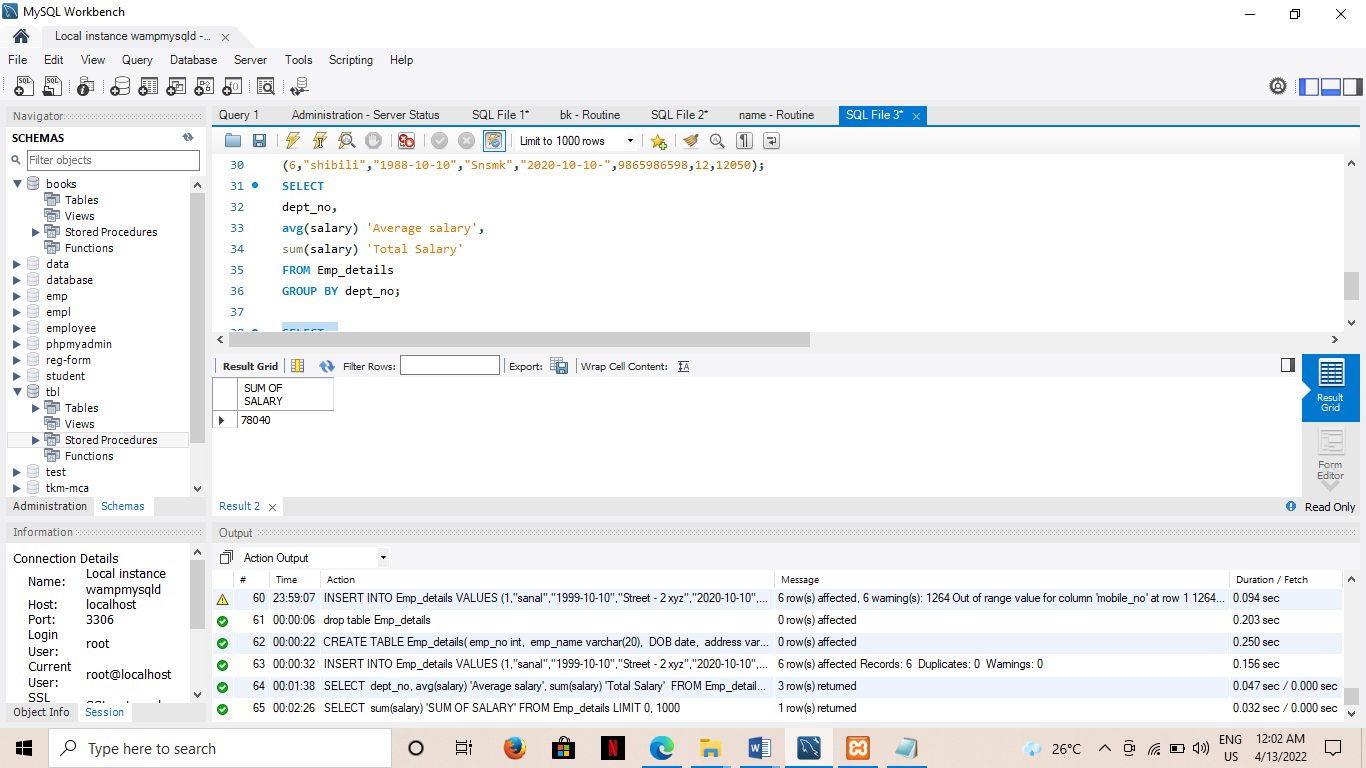


**b)** SELECT

sum(salary) 'SUM OF SALARY'

FROM Emp\_details;

**OUTPUT**

****

**c)**

//FUCTION//

CREATE DEFINER=`root`@`localhost` FUNCTION `fund\_total`() RETURNS double

BEGIN

DECLARE f DOUBLE;

DECLARE i DOUBLE;

SELECT SUM(fund\_amount)

FROM Admins;

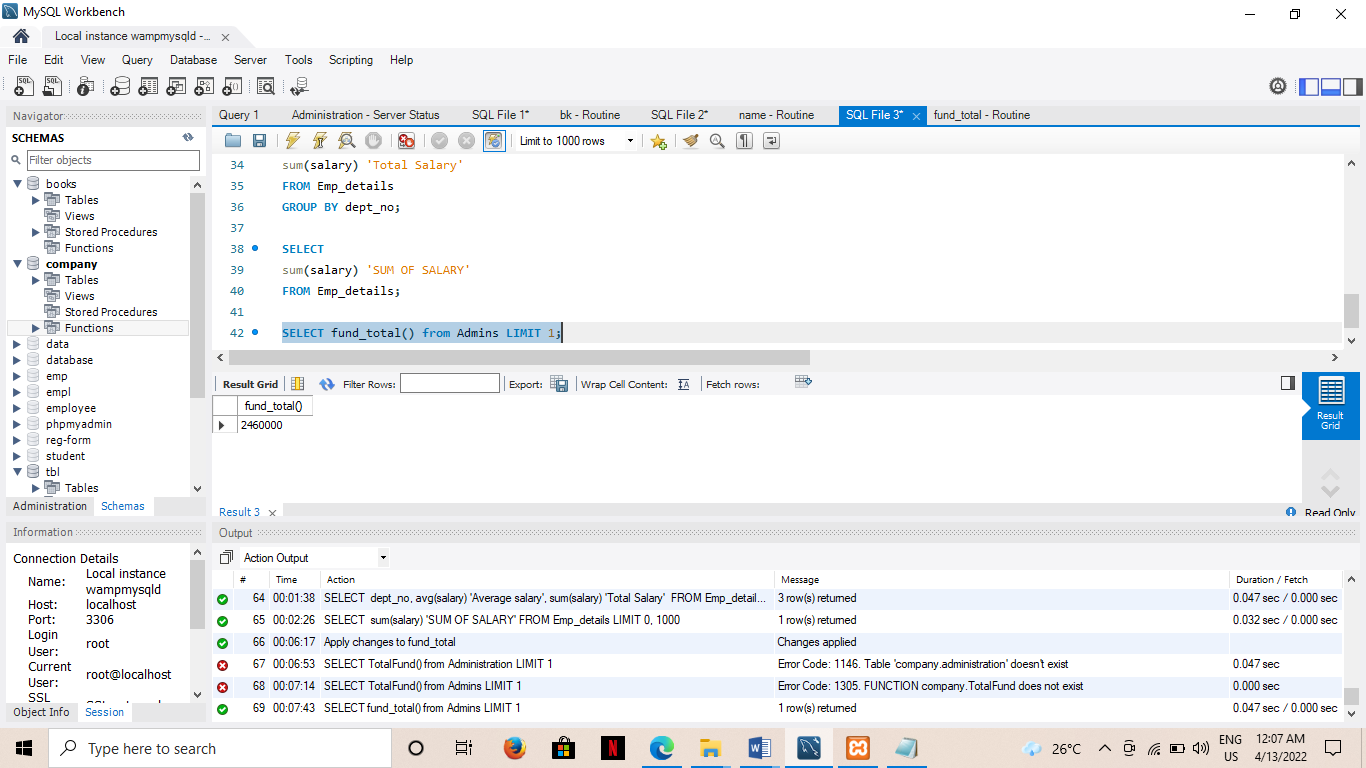
RETURN f;

END

//FUNCTION CALL//

SELECT fund\_total() from Admins LIMIT 1;

**OUTPUT**

****

**PROGRAM-6**

**AIM:** To write a program to implement trigger

**CODE:**

create database employees;

use employees;

create table employee(emp\_id int,emp\_name varchar(10),department\_name varchar(15));

insert into employee values(101,"Coen","mca");

insert into employee values(102,"aloe","mca");

insert into employee values(103,"Raimi","btech");

insert into employee values(104,"Ras","mca");

create table dpt\_mca(dept\_id int,dept\_name varchar(20), dept\_emp varchar(15));

create table dpt\_cs(dept\_id int,dept\_name varchar(20), dept\_emp varchar(15));

select \* from employee;

insert into employee values(105,"Anu","mca");

select \* from dpt\_mca;

insert into employee values(106,"R0se","CS");

select \* from dpt\_cs;

///TRIGGER///

CREATE DEFINER=`root`@`localhost` TRIGGER `employees`.`employee\_BEFORE\_INSERT` BEFORE INSERT ON `employee` FOR EACH ROW

BEGIN

if new.department\_name="mca" then

INSERT INTO dpt\_mca set dept\_id=new.emp\_id,dept\_name=new.emp\_name,dept\_emp="Asst.proff fill";

end if;

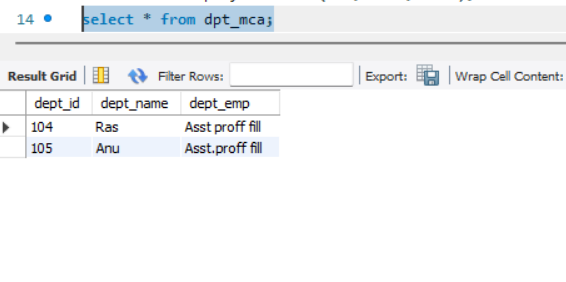
if new.department\_name="cs" then

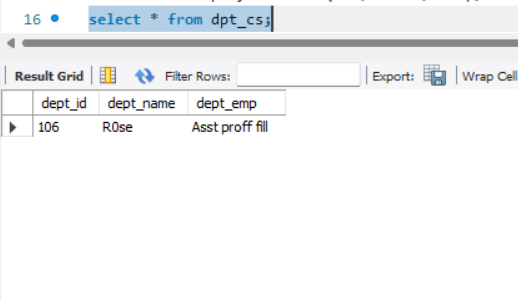
INSERT INTO dpt\_cs set dept\_id=new.emp\_id,dept\_name=new.emp\_name,dept\_emp="Asst.proff fill";

end if;

END

**OUTPUT**:

****

****

**PROGRAM-7**

**AIM:** . To create a student record database in which student marks assessment is recorded.In such schema,create a trigger so that the total and average of specified marks is automatically inserted whenever a record is inserted.

**CODE:**

create database students;

use students;

create table student(rollno varchar(10) primary key,studname varchar(10),sub1 varchar(10),sub2 varchar(10),sub3 varchar(10));

create table result(rollno varchar(10),studname varchar(10),total\_marks varchar(10),percentage varchar(10));

insert into student values("1","abhi","35","55","85"),

("2","adarsh","15","60","10"),

("3","anu","96","99","94");

insert into student values(8,"jain",67,90,76);

insert into student values(10,"bini",60,96,50);

select \* from student;

select \* from result;

///TRIGGER///

CREATE DEFINER=`root`@`localhost` TRIGGER `students`.`student\_AFTER\_INSERT` AFTER INSERT ON `student` FOR EACH ROW

BEGIN

declare total varchar(10);

declare perc varchar(10);

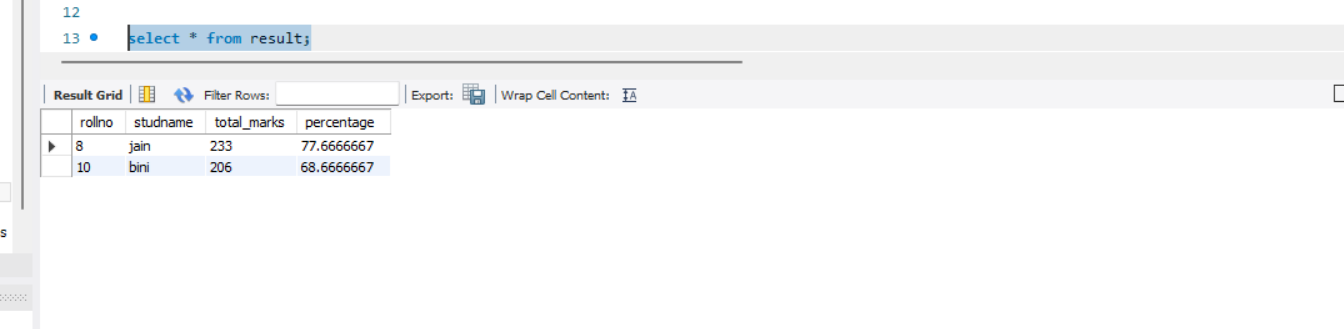
set total=new.sub1+new.sub2+new.sub3;

set perc=((total/300)\*100);

insert into result values(new.rollno,new.studname,total,perc);

END

**OUTPUT**:



**PROGRAM-8**

**AIM:** To write a program to implement cursors

**CODE:**

create database college1;

use college1;

create table library (shelf\_no int(10),category varchar(10),book\_name varchar(20));

insert into library values(11,'science','algebra');

insert into library values(12,'science','Data Mining');

insert into library values(21,'comic','New Avengers');

insert into library values(22,'comic','Spiderman');

insert into library values(31,'drama','romeo and juliet');

insert into library values(32,'drama','hamlet');

create table book\_by\_order(book\_shelf int (10),book\_category varchar(20),bookname varchar(20));

select \* from library;

call book\_details();

select \* from book\_by\_order;

///CURSOR///

CREATE DEFINER=`root`@`localhost` PROCEDURE `book\_details`()

BEGIN

declare book\_shelf int;

declare bookname varchar(20);

declare book\_category varchar(10);

declare C\_finished integer default 0;

declare C1 cursor for select shelf\_no,category,book\_name from library;

declare continue handler for not found set C\_finished = 1;

open C1;

book\_details:loop

if C\_finished=1 then

leave book\_details;

end if;

if C\_finished = 0 then

Fetch from C1 into book\_shelf,book\_category,bookname;

if book\_category = 'comic' then

insert into book\_by\_order values(book\_shelf,bookname,book\_category);

end if;

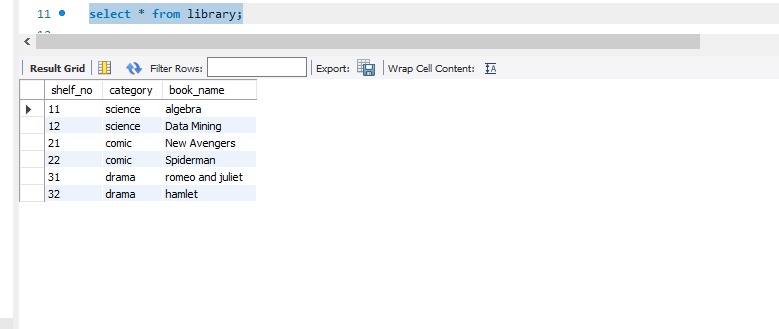
end if;

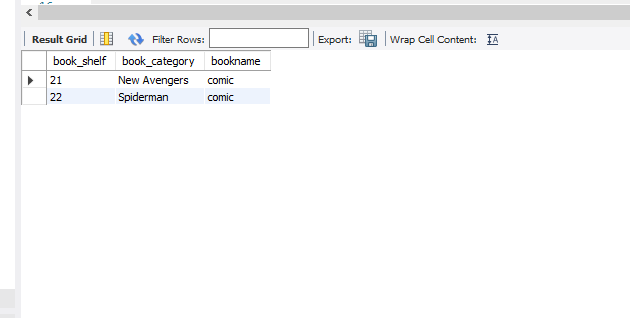
end loop;

close C1;

END

**OUTPUT** :



****