Prior Instructions

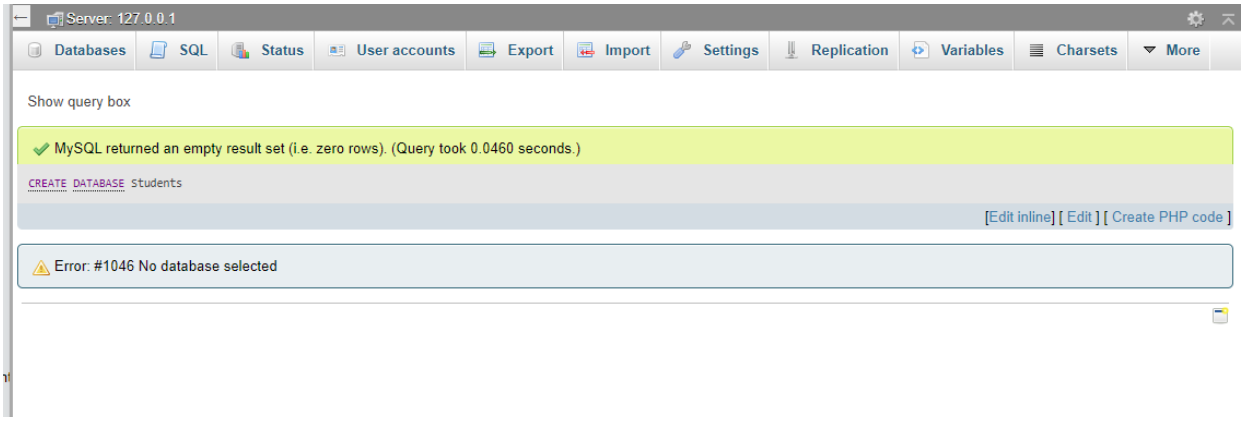
- **Please do read all the questions before performing any operations in the database**
- **Once you have fully gone through the questions then likewise decide the contents and table columns and follow the below instructions**

---

1. Create Student Database

   Solution:-

   create database Student;



2. Create the following table under the Student Database:

a. StudentBasicInformation

i. Columns

1. StudentName
2. StudentSurname
3. StudentRollNo
4. StudentAddress
5. Add more three basic columns of the name of your own

b. StudentAdmissionPaymentDetails

i. Columns

1. StudentRollNo
2. AmountPaid
3. AmountBalance
4. Add more four basic columns of the name of your own

c. StudentSubjectInformation

i. Columns

1. SubjectOpted
2. StudentRollNo

3. SubjectTotalMarks
4. SubjectObtainedMarks
5. StudentMarksPercentage
6. Add more one columns of the name of your own

d. SubjectScholarshipInformation

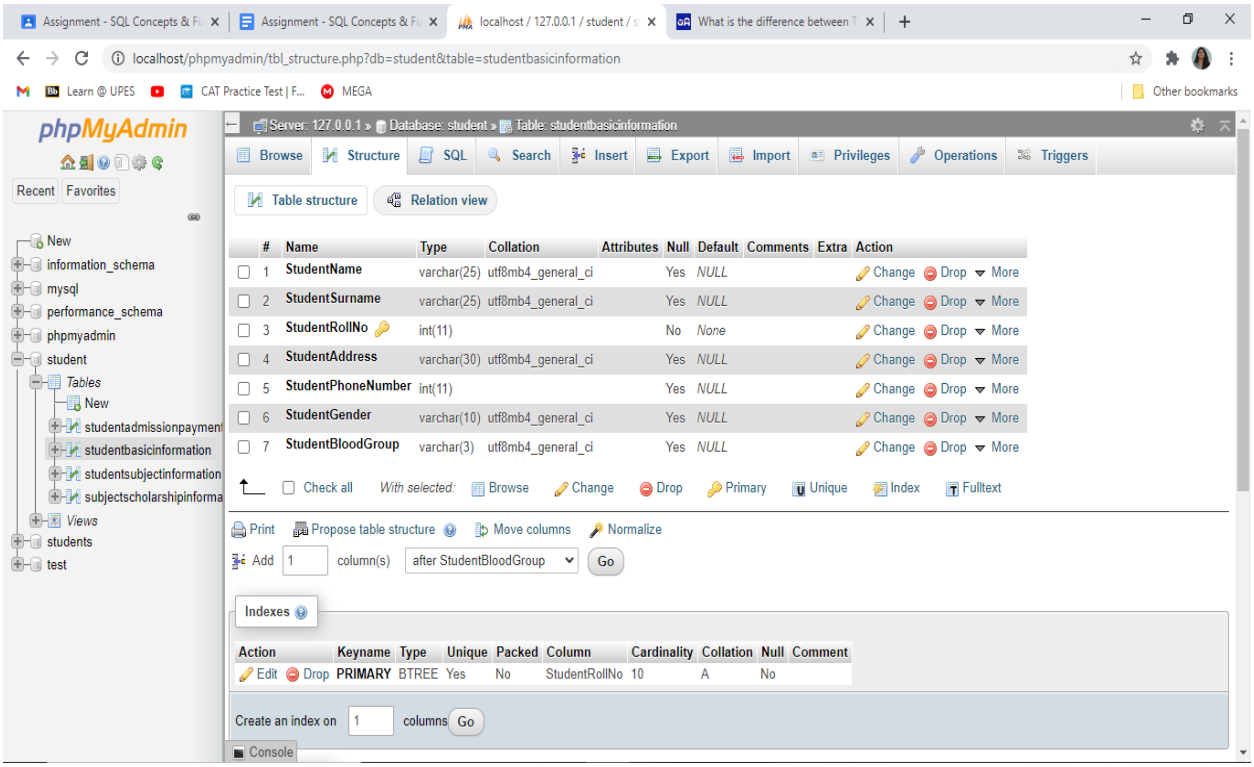
i. Columns

1. StudentRollNo
2. ScholarshipName
3. ScholarshipDescription
4. ScholarshipAmount
5. ScholarshipCategory
6. Add more two columns of the name of your own

Solution:-

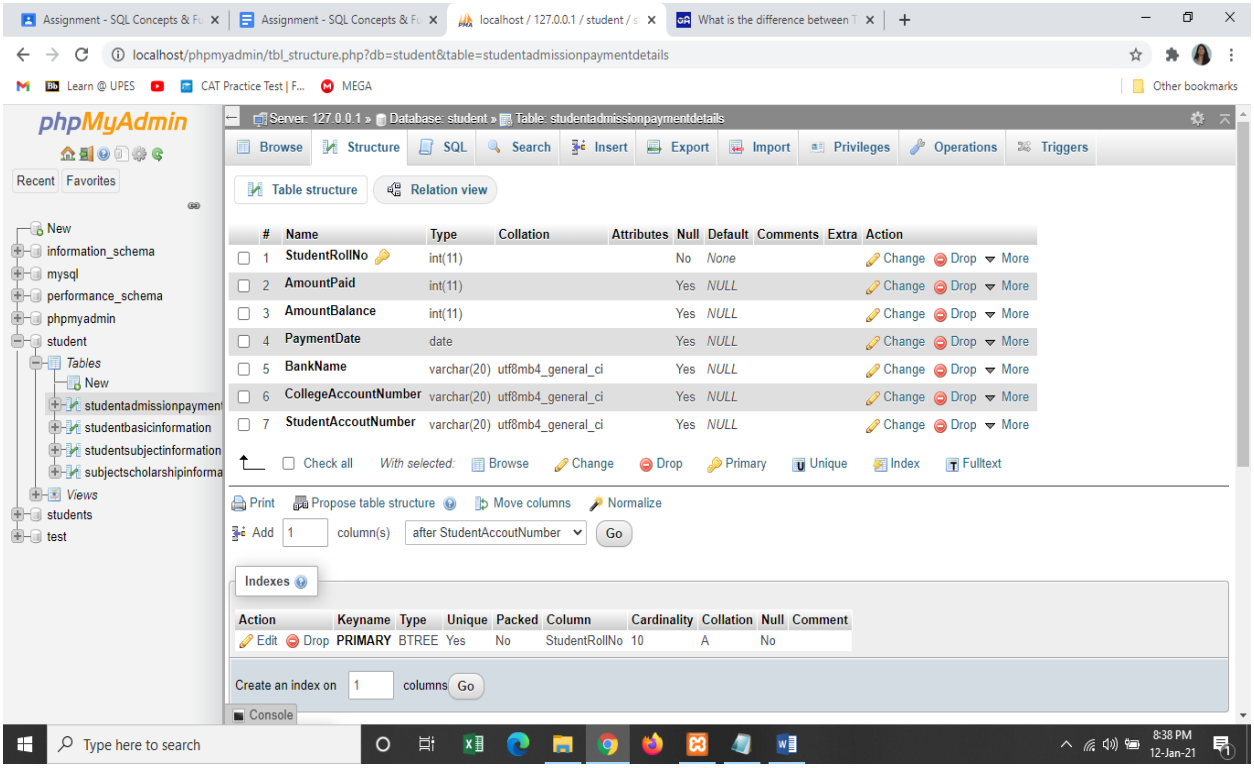
a)

create table StudentBasicInformation(StudentName varchar(25), StudentSurname varchar(25), StudentRollNo int primary key,StudentAdress varchar(30),StudentPhoneNumber bigint, StudentGender varchar(10), StudentBloodGroup varchar(3));



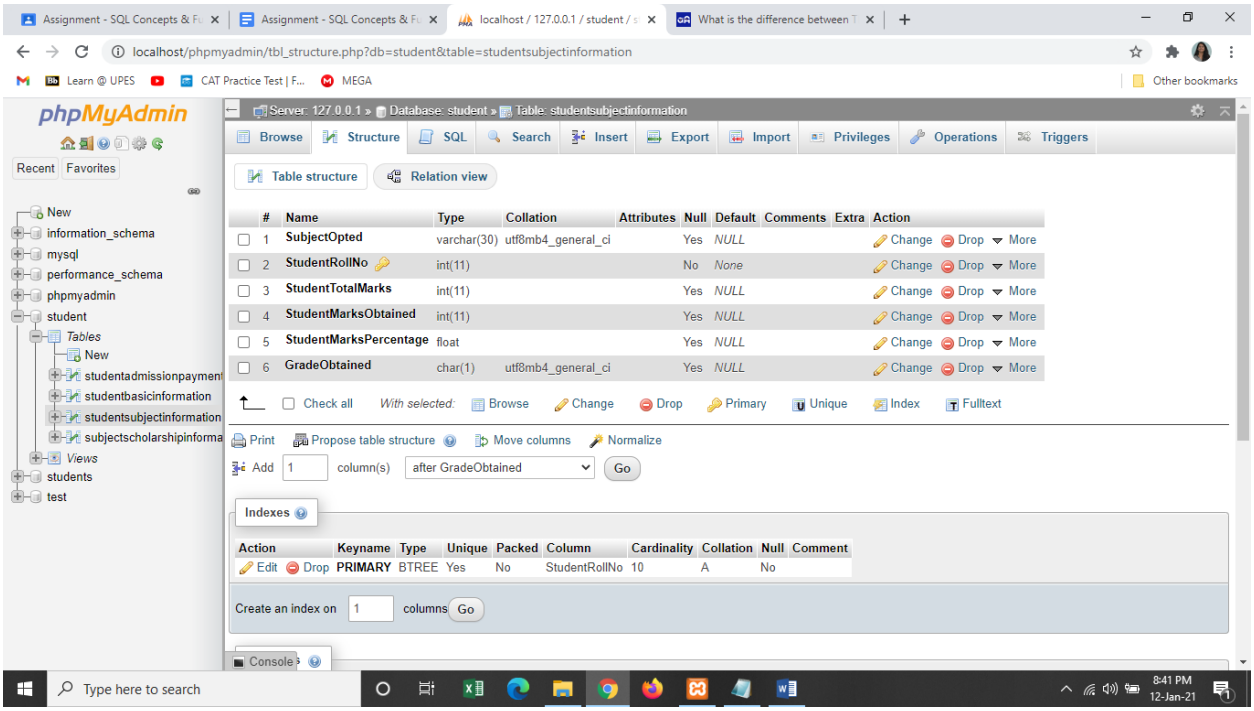
b)

create table StudentAdmissionPaymentDetails (StudentRollNo int primary key, AmountPaid int, AmountBalance int, PaymentDate date, BankName varchar(20), CollegeAccoutNumber varchar(20), StudentAccoutNumber varchar(20) ,foreign key(StudentRollNo) references StudentBasicInformation(StudentRollNo));



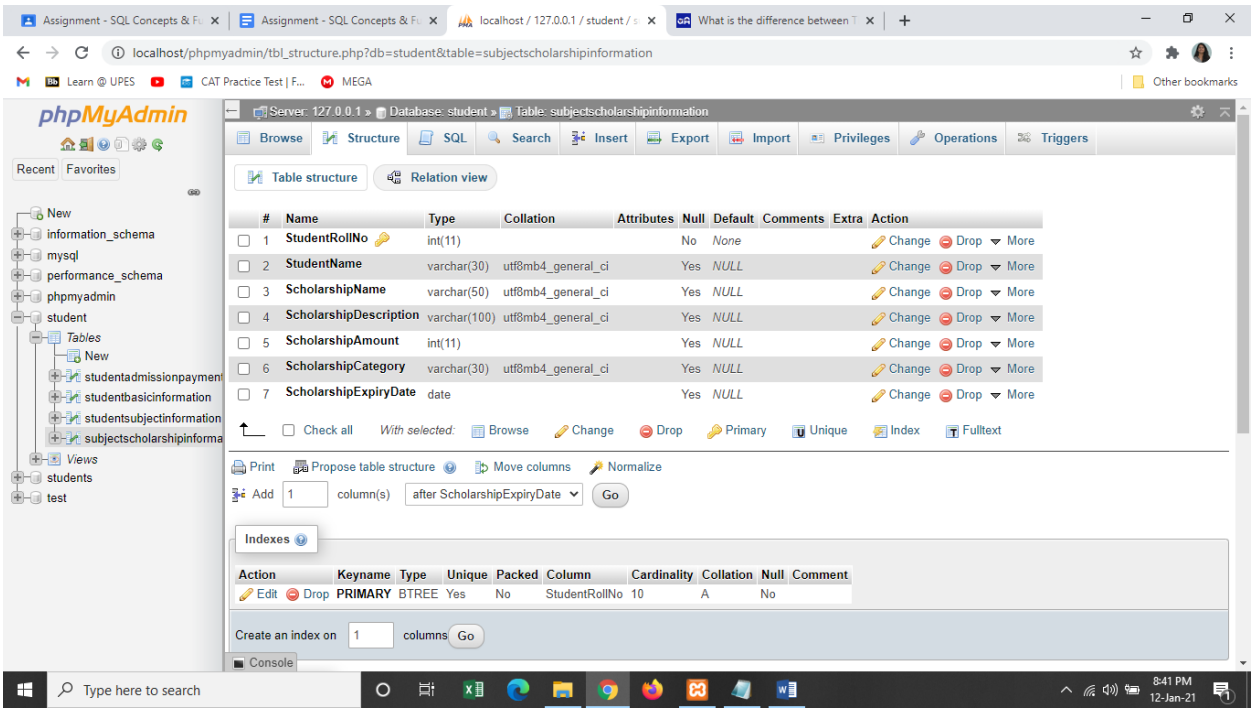
c)

create table StudentSubjectInformation (SubjectOpted varchar(10), StudentRollNo int , SubjectTotalMarks int,SubjectObtainedMarks int, StudentMarksPercentage int, StudentGrade varchar(2),foreign key(StudentRollNo) references StudentBasicInformation(StudentRollNo));



d)

create table StudentScholarshipInformation(StudentRollNo int, ScholarshipName varchar(30), ScholarshipName varchar(50), ScholarshipDescription varchar(100), ScholarshipAmount int, ScholarshipCategory varchar(10), ScholarshipExpiryDate date, StudentSchlarshipValidity int,foreign key(StudentRollNo) references StudentBasicInformation(StudentRollNo));



3. Insert more than 10 records in each and every table created
4. Snap of the all the tables once the insertion is completed

Solution:-

Table Name:- StudentBasicInformation

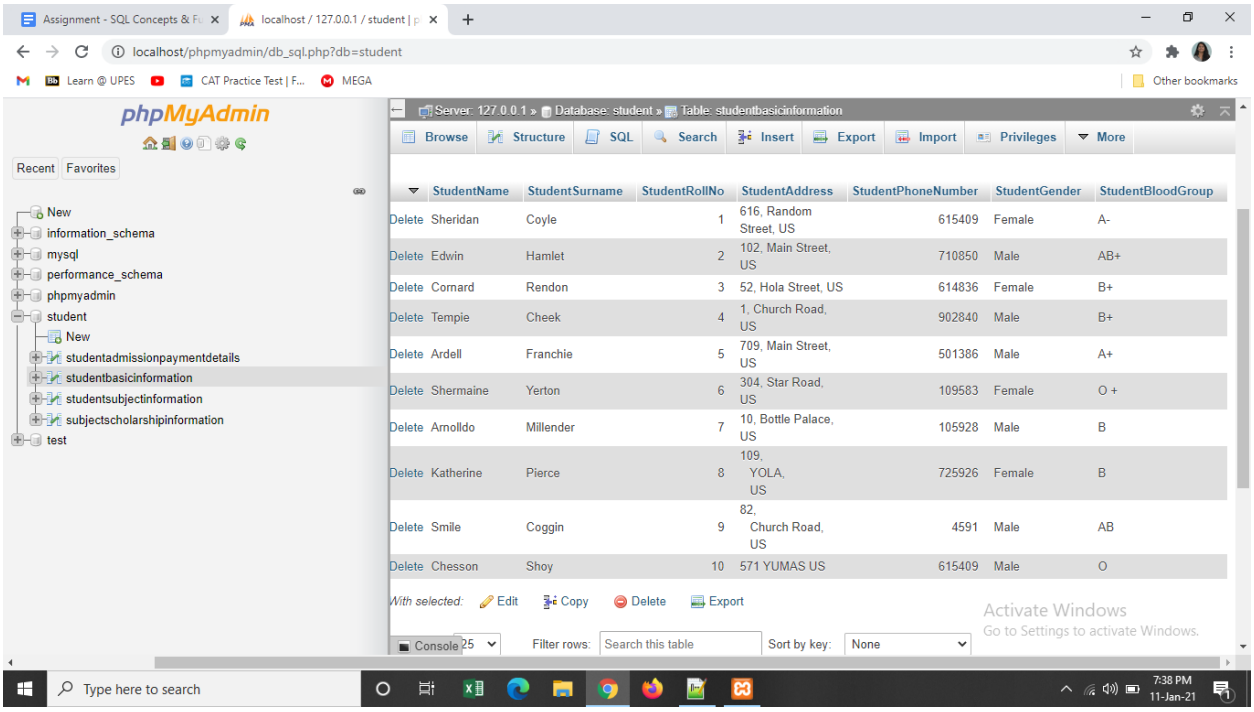


Table Name:- StudentAdmissionPaymentDetails

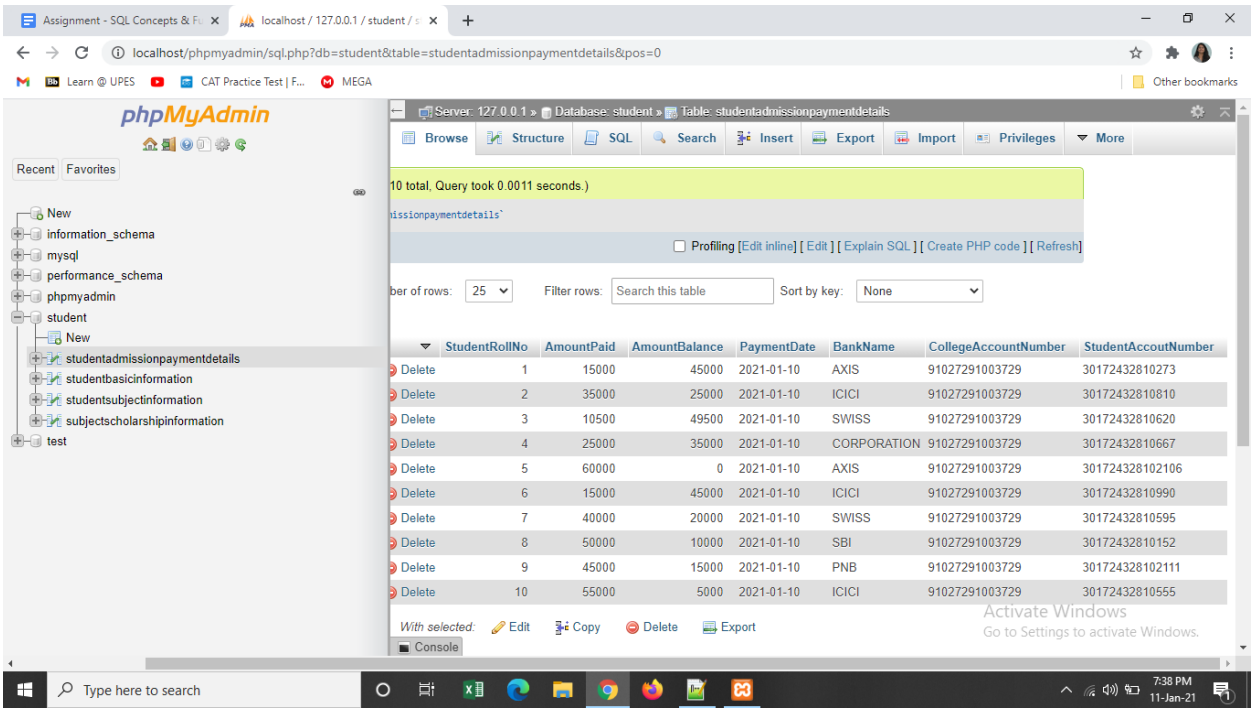


Table Name:- StudentSubjectInformation

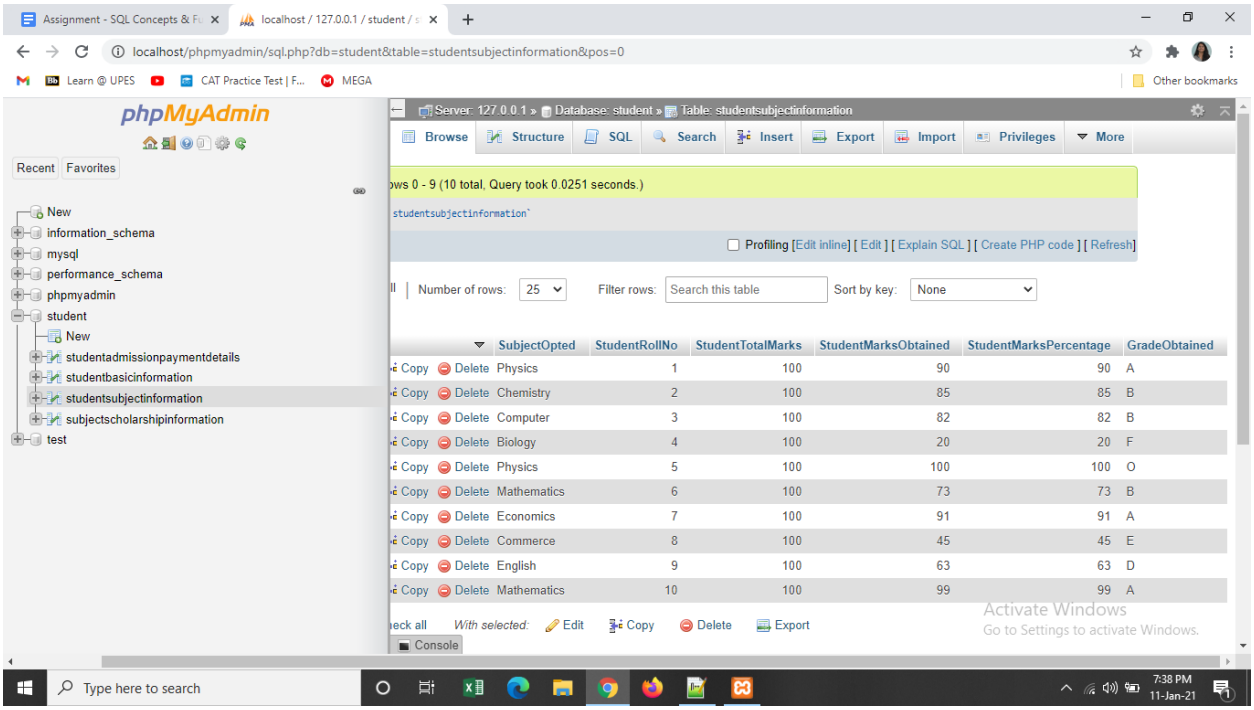
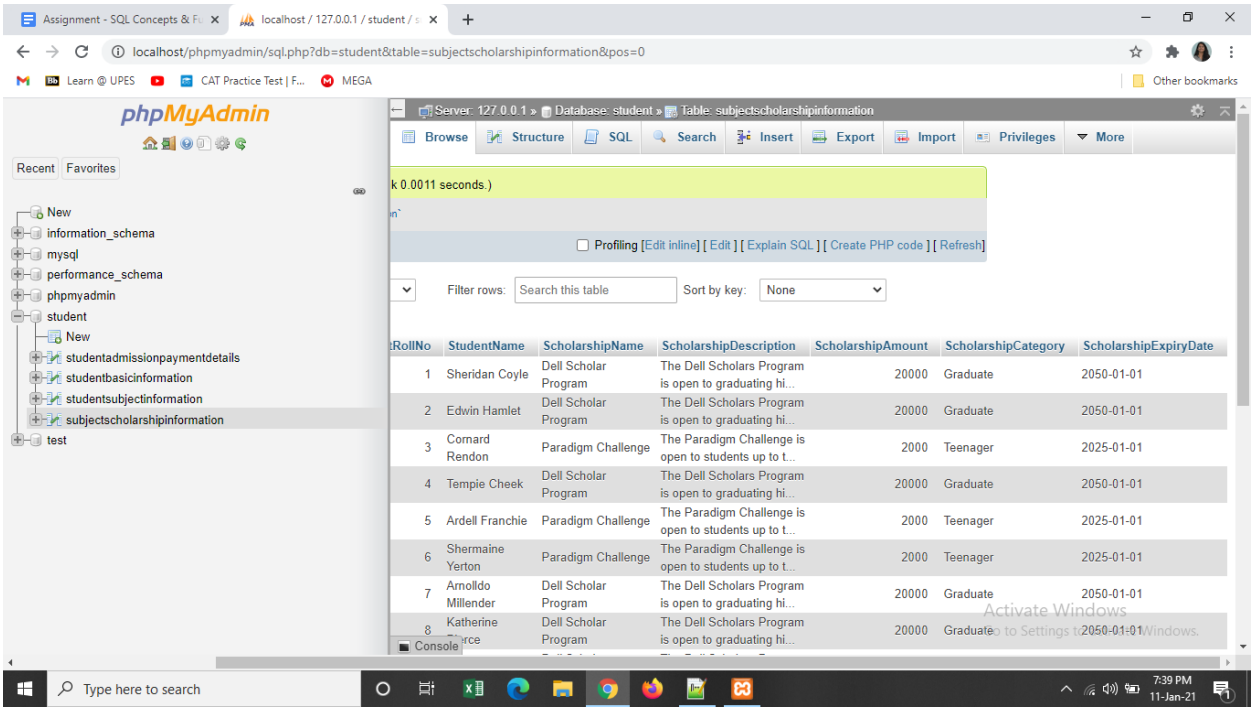


Table Name:- StudentScholarshipInformation



5. Update any 5 records of your choice in any table like update the StudentAddress with some other address content and likewise so on with any records of any table of your choice

   Solution:-

   a)
   update studentbasicinformation set StudentAddressr=”56 Gold Street US” where StudentRollNo=1;

   b)
   update studentscholarshipinformation set ScholarshipAmount=5000 where StudentName = “Katherine Pierce”;
   c)
   update StudentAdmissionPayment set BankName ="ICICI" where AmountBalance=0;

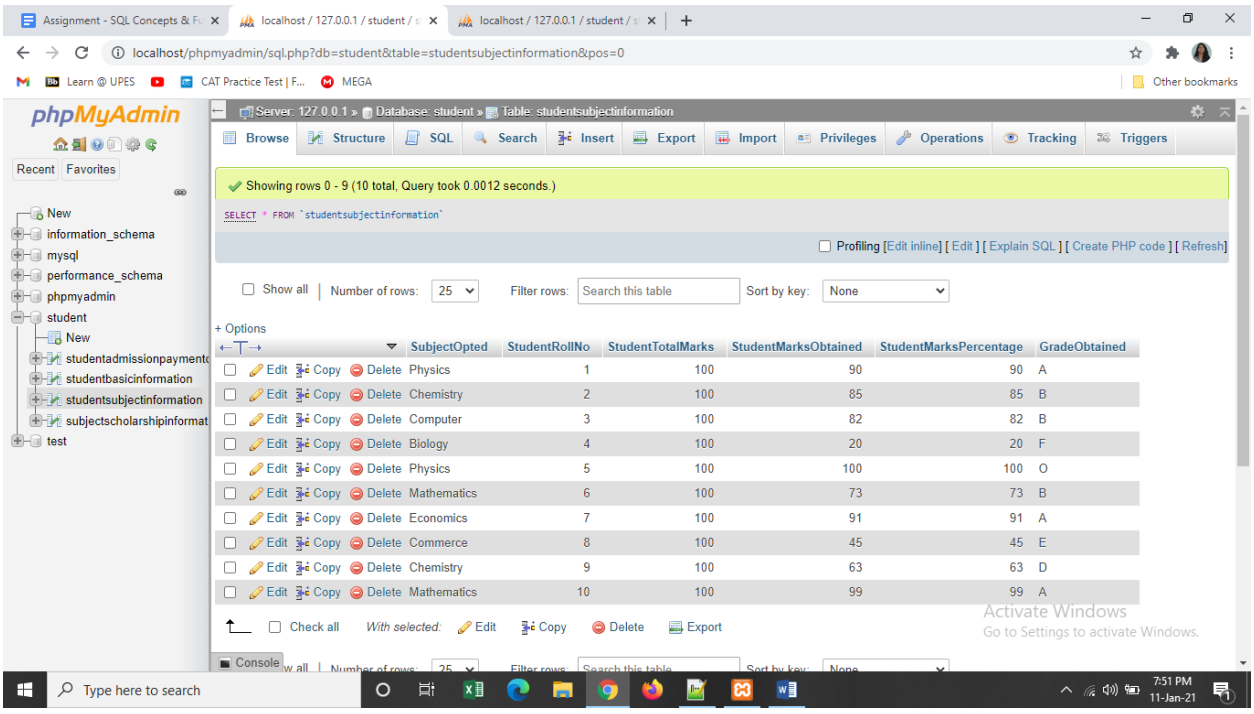
   d)
   update StudentSubjectInformation set SubjectOpted="Chemistry" where StudentRollNo=9;

   e)
   update StudentAdmissionPayment set BankName ="PNB" where StudentRollNo=6;

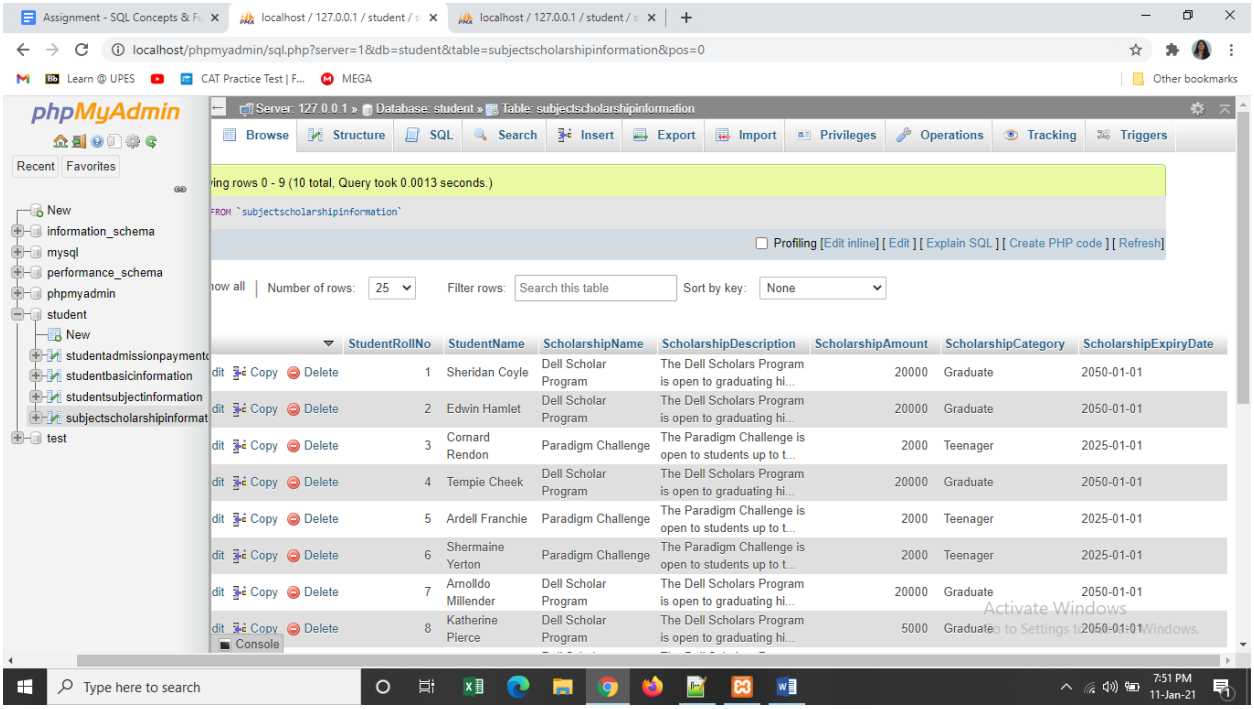
6. Snap of the all the tables post updation

   Solution:-

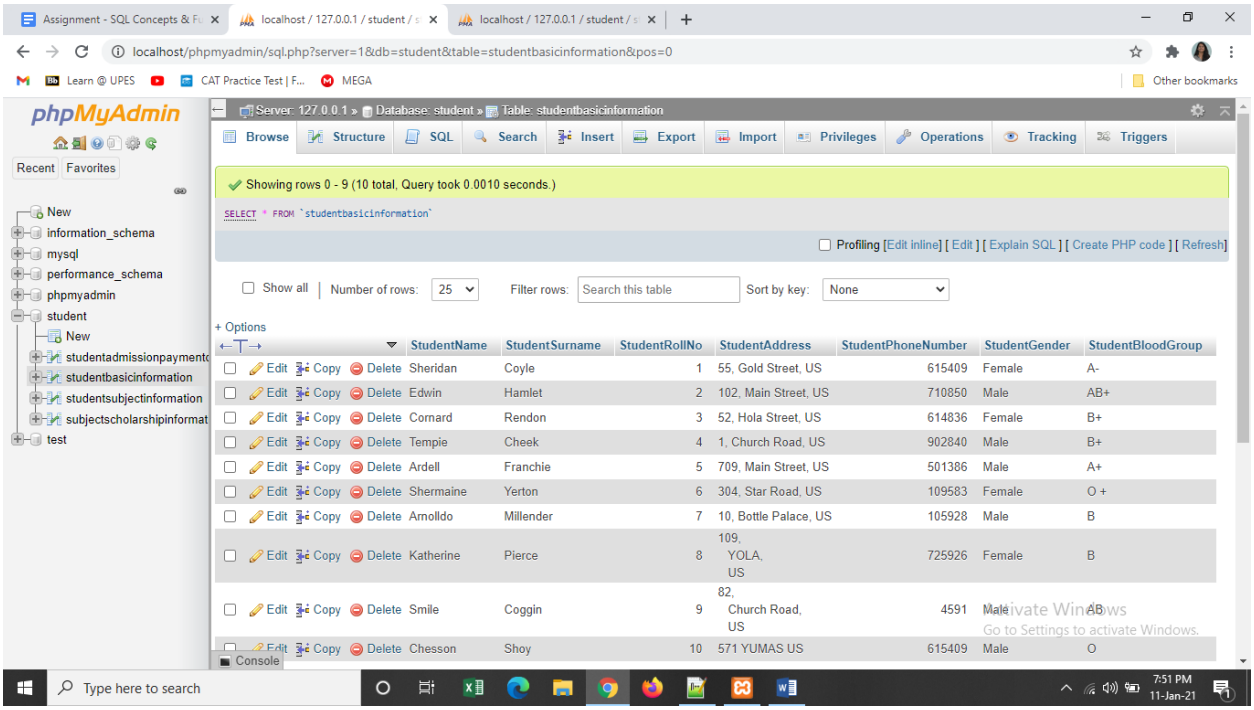
   a)



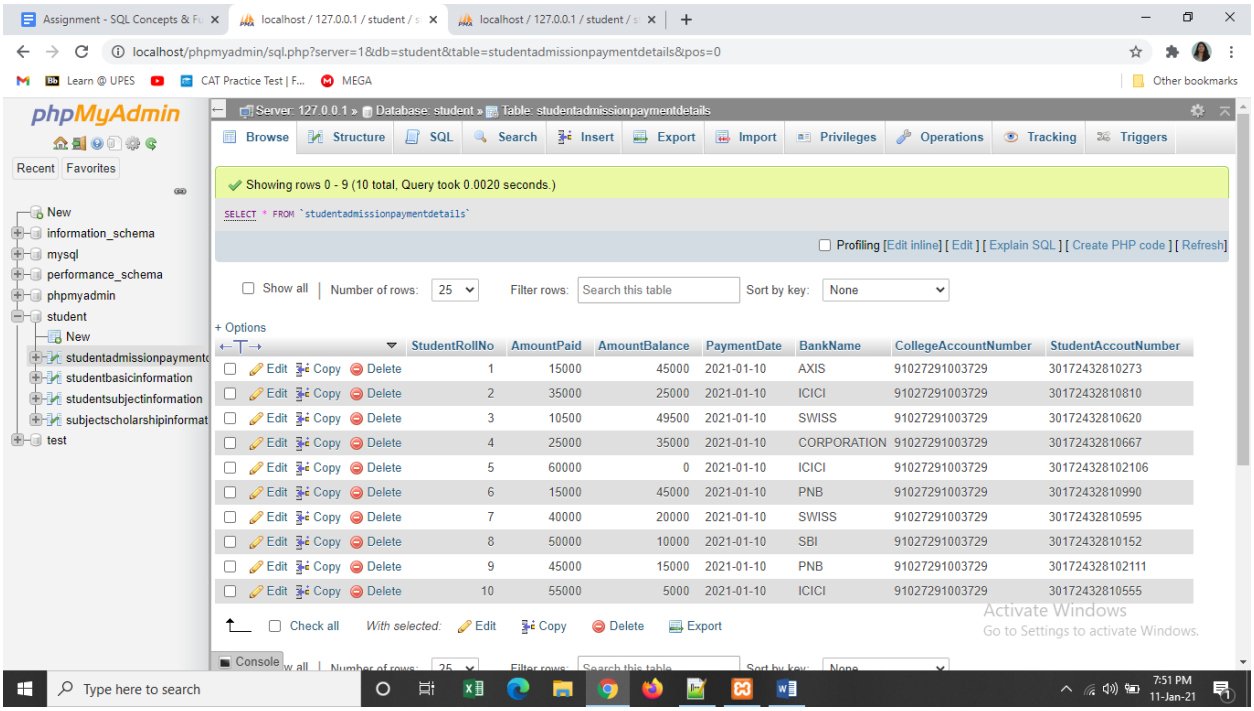
b)


Assignment - SQL Concepts & Fu
localhost / 127.0.0.1 / student /
localhost / 127.0.0.1 / student /
localhost/phpmyadmin/sql.php?server=1&db=student&table=subjectscholarshipinformation&pos=0
Learn @ UPES
CAT Practice Test | F...
MEGA
Other bookmarks
phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
student
New
studentadmissionpaymento
studentbasicinformation
studentsubjectinformation
subjectscholarshipinformat
test
Server: 127.0.0.1 » Database: student » Table: subjectscholarshipinformation
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
ing rows 0 - 9 (10 total, Query took 0.0013 seconds.)
FROM `subjectscholarshipinformation`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
how all | Number of rows: 25 Filter rows: Search this table Sort by key: None
StudentRollNo StudentName ScholarshipName ScholarshipDescription ScholarshipAmount ScholarshipCategory ScholarshipExpiryDate
dit Copy Delete 1 Sheridan Coyle Dell Scholar Program The Dell Scholars Program is open to graduating hi... 20000 Graduate 2050-01-01
dit Copy Delete 2 Edwin Hamlet Dell Scholar Program The Dell Scholars Program is open to graduating hi... 20000 Graduate 2050-01-01
dit Copy Delete 3 Cornard Rendon Paradigm Challenge The Paradigm Challenge is open to students up to t... 2000 Teenager 2025-01-01
dit Copy Delete 4 Tempie Cheek Dell Scholar Program The Dell Scholars Program is open to graduating hi... 20000 Graduate 2050-01-01
dit Copy Delete 5 Ardell Franchie Paradigm Challenge The Paradigm Challenge is open to students up to t... 2000 Teenager 2025-01-01
dit Copy Delete 6 Shermaine Yerton Paradigm Challenge The Paradigm Challenge is open to students up to t... 2000 Teenager 2025-01-01
dit Copy Delete 7 Arnolldo Millender Dell Scholar Program The Dell Scholars Program is open to graduating hi... 20000 Graduate 2050-01-01
dit Copy Delete 8 Katherine Pierce Dell Scholar Program The Dell Scholars Program is open to graduating hi... 5000 Graduate 2050-01-01
Activate Windows
Console
Type here to search
7:51 PM
11-Jan-21

c)


Assignment - SQL Concepts & Fu
localhost / 127.0.0.1 / student /
localhost / 127.0.0.1 / student /
localhost/phpmyadmin/sql.php?server=1&db=student&table=studentbasicinformation&pos=0
Learn @ UPES
CAT Practice Test | F...
MEGA
Other bookmarks
phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
student
New
studentadmissionpaymento
studentbasicinformation
studentsubjectinformation
subjectscholarshipinformat
test
Server: 127.0.0.1 » Database: student » Table: studentbasicinformation
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Showing rows 0 - 9 (10 total, Query took 0.0010 seconds.)
SELECT * FROM `studentbasicinformation`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
StudentName StudentSurname StudentRollNo StudentAddress StudentPhoneNumber StudentGender StudentBloodGroup
Edit Copy Delete Sheridan Coyle 1 55, Gold Street, US 615409 Female A-
Edit Copy Delete Edwin Hamlet 2 102, Main Street, US 710850 Male AB+
Edit Copy Delete Cornard Rendon 3 52, Hola Street, US 614836 Female B+
Edit Copy Delete Tempie Cheek 4 1, Church Road, US 902840 Male B+
Edit Copy Delete Ardell Franchie 5 709, Main Street, US 501386 Male A+
Edit Copy Delete Shermaine Yerton 6 304, Star Road, US 109583 Female O +
Edit Copy Delete Arnolldo Millender 7 10, Bottle Palace, US 105928 Male B
Edit Copy Delete Katherine Pierce 8 109, YOLA, US 725926 Female B
Edit Copy Delete Smile Coggin 9 82, Church Road, US 4591 Male AB
Edit Copy Delete Chesson Shoy 10 571 YUMAS US 615409 Male O
Activate Windows
Go to Settings to activate Windows.
Console
Type here to search
7:51 PM
11-Jan-21

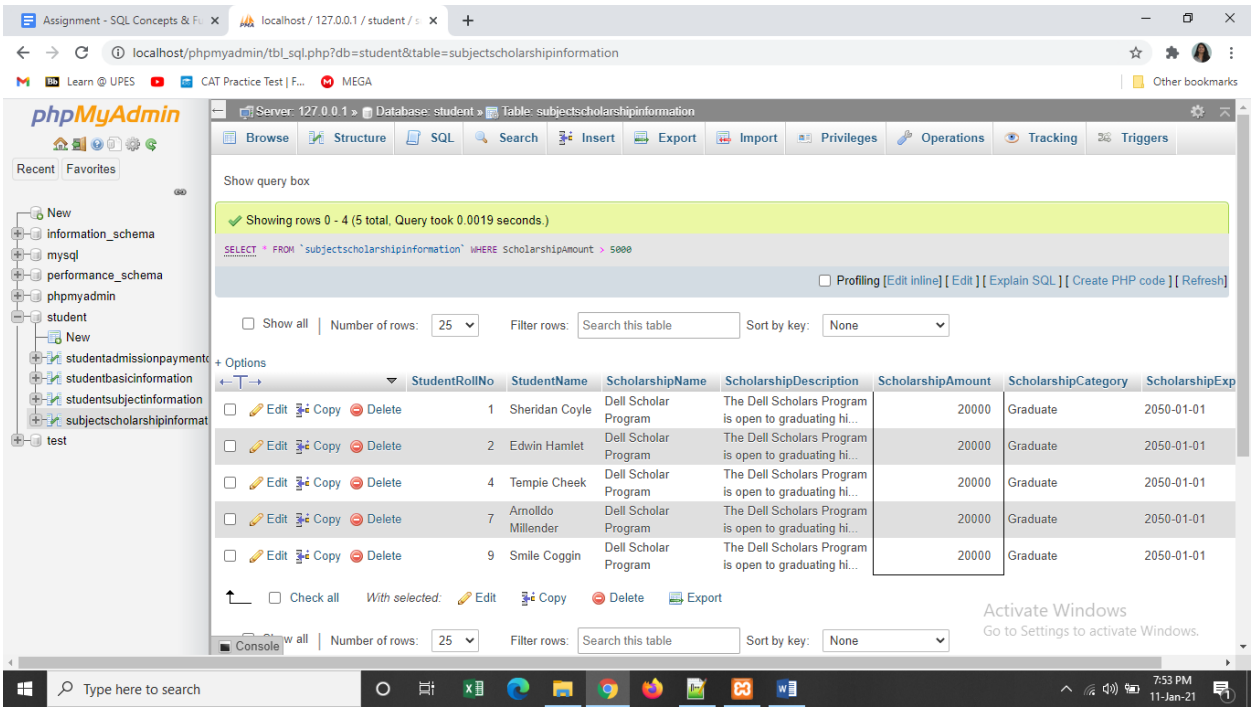
d)



7. Select the student details records who has received the scholarship more than 5000Rs/-

Solution:-

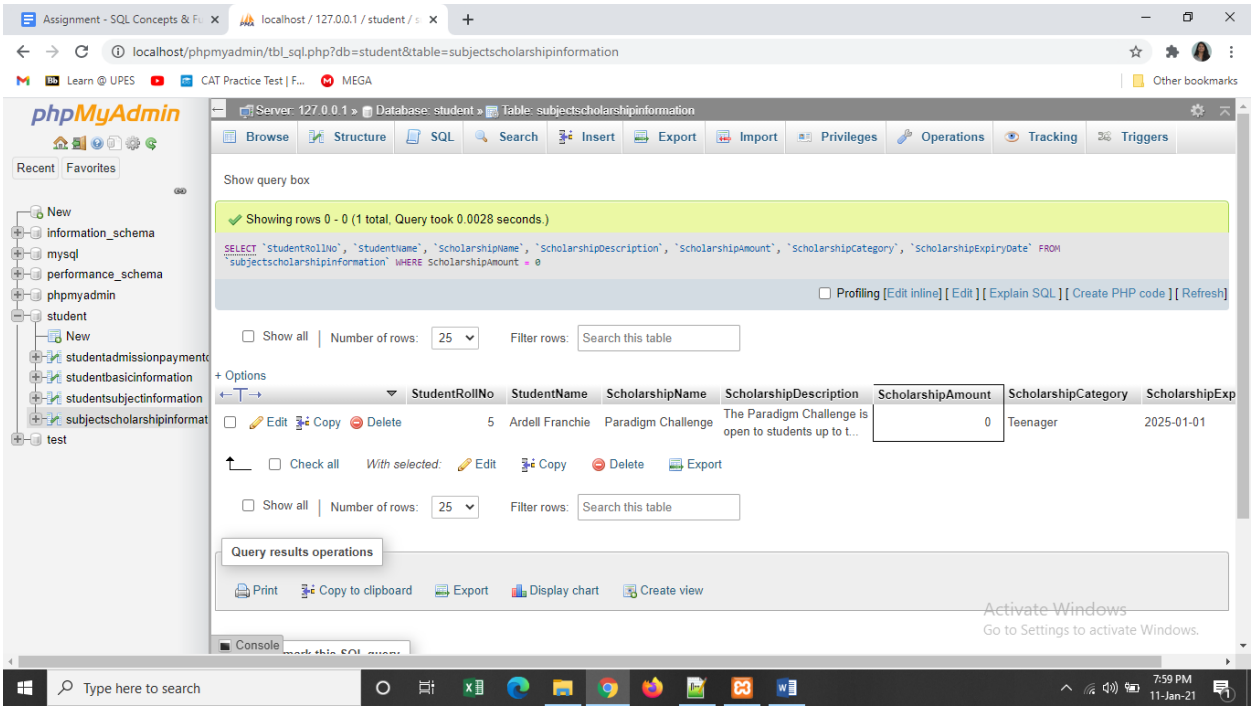
select * from subjectscholarshipinformation where scholarshipamount>5000;



8. Select the students who opted for scholarship but has not got the scholarship

   Solution:-

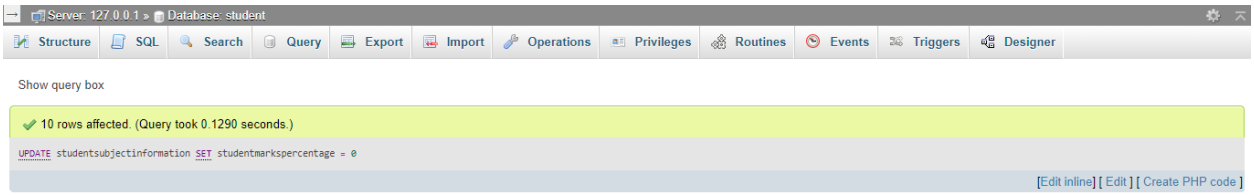
   select StudentRollNo, StudentName, ScholarshipName, ScholarshipDescription, ScholarshipAmount, ScholarshipCategory, ScholarshipExpiryDate from subjectscholarshipinformation where scholarshipamount = 0

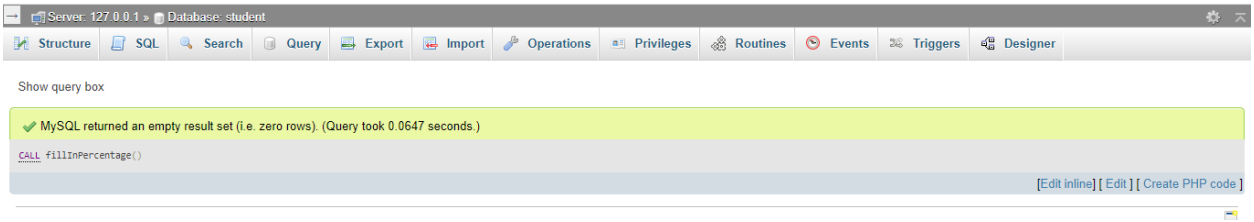


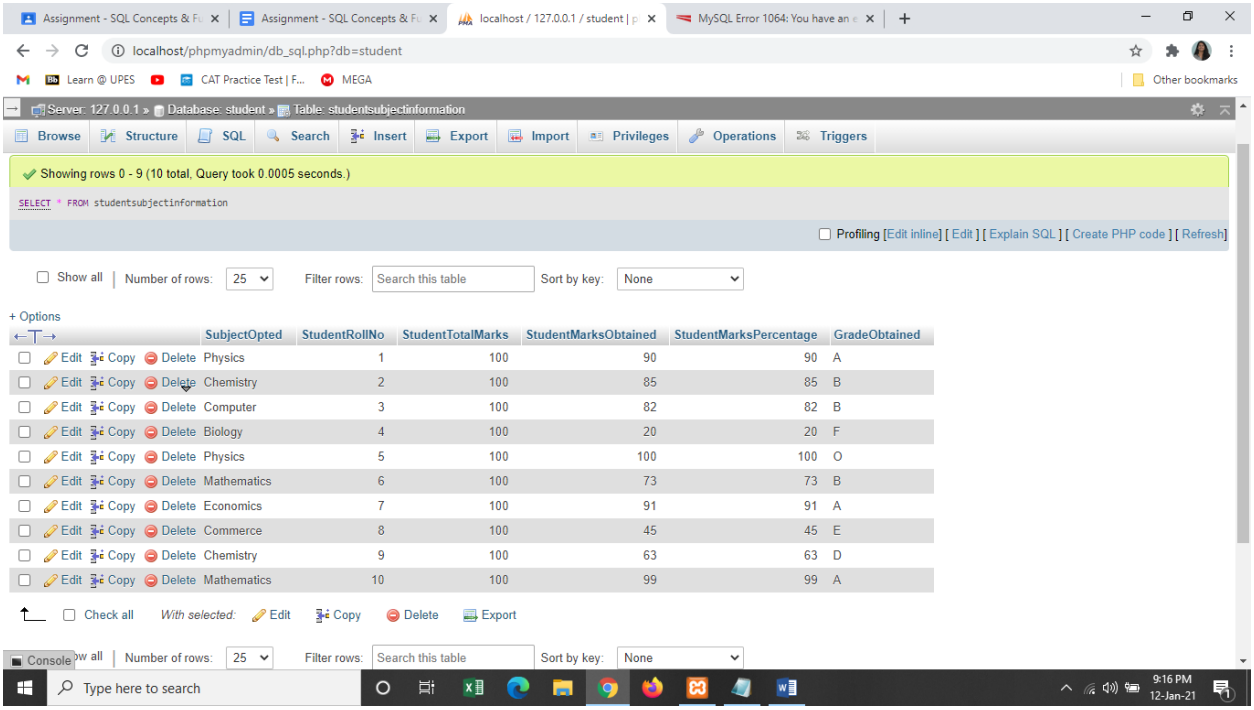
9. Fill in data for the percentage column i.e. StudentMarksPercentage in the table StudentSubjectInformation by creating and using the stored procedure created

   Solution:-

```
delimiter //
create procedure insertpercentages()
begin
update studentsubjectinformation set
studentmarkspercentage=100*(subjectobtainedmarks/subjecttotalmarks);
end //
delimiter ;

call insertpercentage();
```





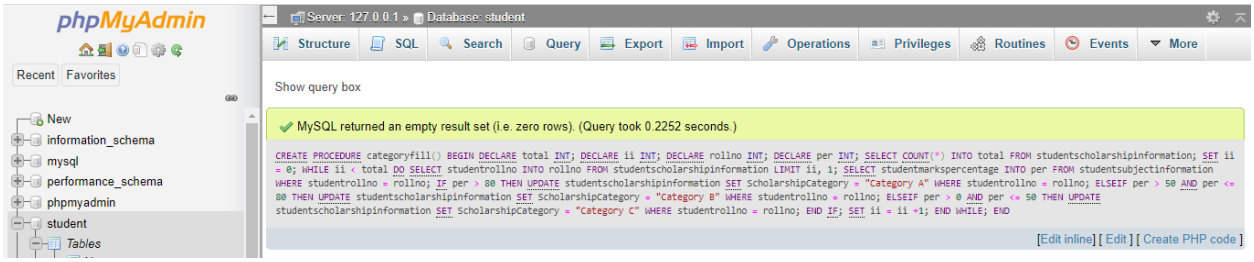


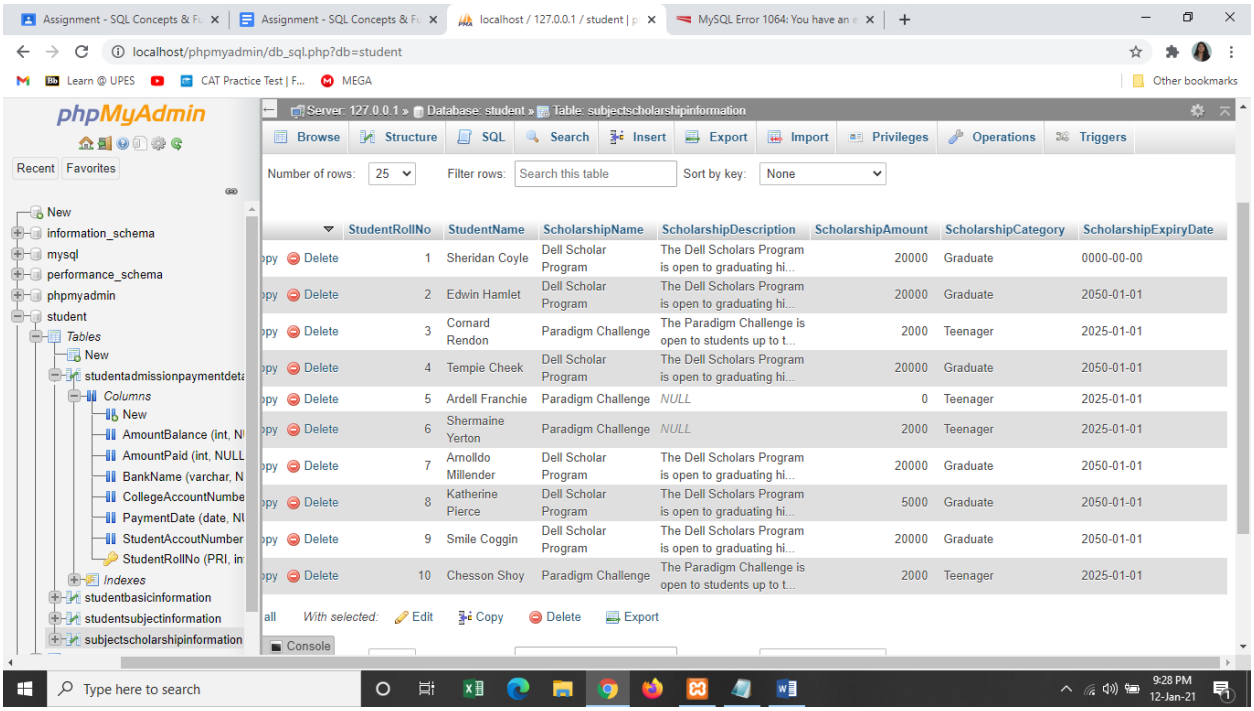
10. Decide the category of the scholarship depending upon the marks/percentage obtained by the student and likewise update the ScholarshipCategory column, create a stored procedure in order to handle this operation

Solution:-

```
delimiter //
create procedure categoryfill()
begin
declare total int;
declare ii int;
declare rollno int;
```

```
declare per int;
select count(*) into total from studentscholarshipinformation;
set ii=0;
while ii<total do
select studentrollno into rollno from studentscholarshipinformation limit ii,1;
select studentmarkspercentage into per from studentsubjectinformation where
studentrollno=rollno;
if per >80 then
update studentscholarshipinformation set ScholarshipCategory="Category A" where
studentrollno = rollno;
elseif per >50 and per <=80 then
update studentscholarshipinformation set ScholarshipCategory="Category B" where
studentrollno = rollno;
elseif per>0 and per<=50 then
update studentscholarshipinformation set ScholarshipCategory="Category C" where
studentrollno = rollno;
end if;
set ii = ii+1;
end while;
end //
delimiter ;

call categoryfill();
```

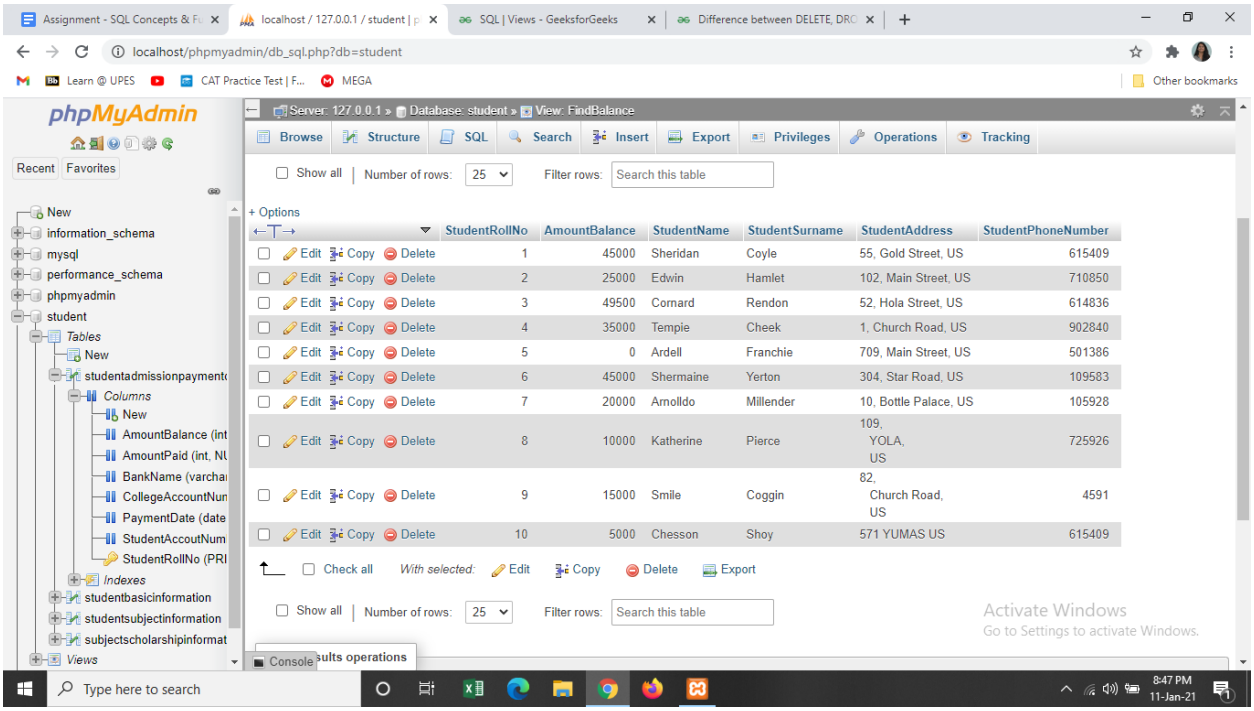




11. Create the View which shows balance amount to be paid by the student along with the student detailed information (use join)

Solution:-

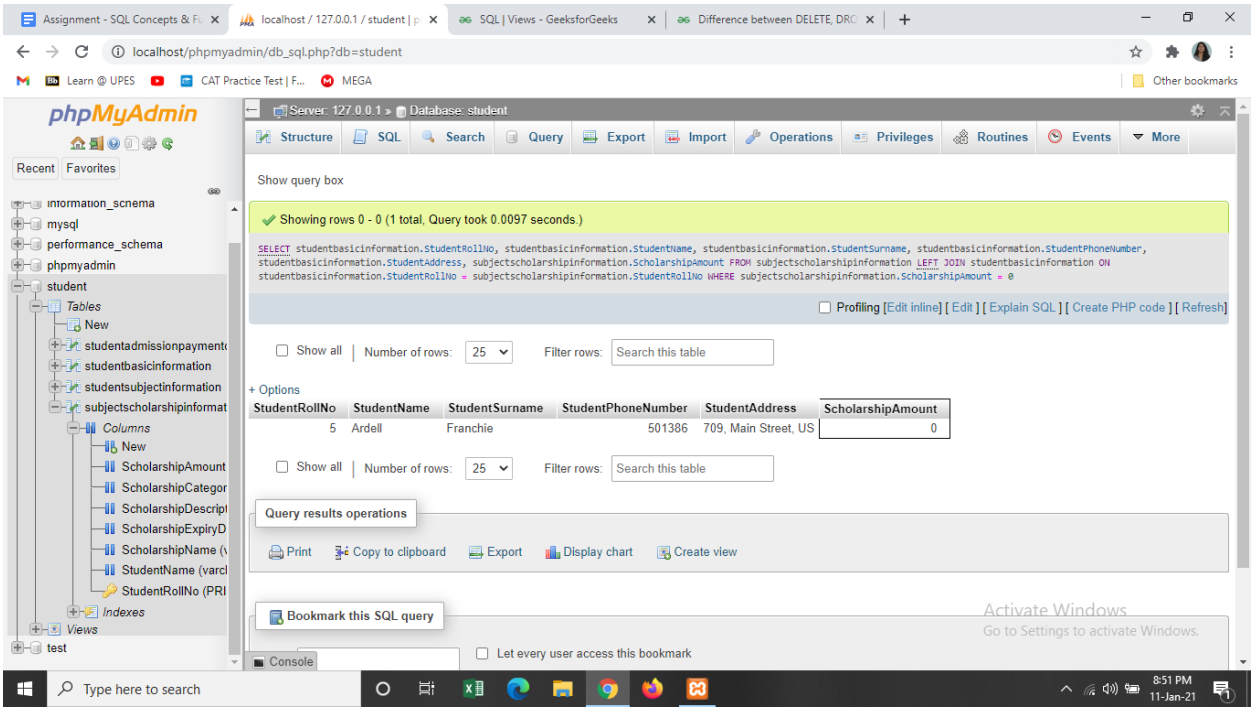
Create view FindBalance as studentbasicinformation.StudentRollNo, studentadmissionpaymentdetails.AmountBalance, studentbasicinformation.StudentName, studentbasicinformation.StudentSurname, studentbasicinformation.StudentAddress, studentbasicinformation.StudentPhoneNumber from studentadmissionpaymentdetails right join studentbasicinformation on studentbasicinformation.StudentRollNo = studentadmissionpaymentdetails.StudentRollNo;



12. Get the details of the students who haven't got any scholarship (use joins/subqueries)

Solution:-

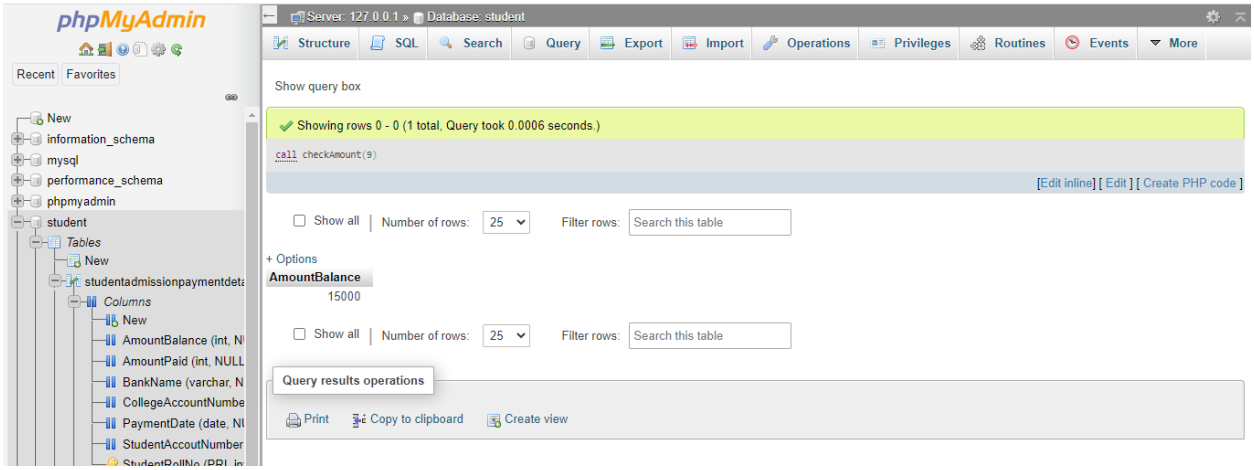
select studentbasicinformation.studentrollno, studentbasicinformation.studentname, studentbasicinformation.studentsurname, studentbasicinformation.studentphonenumber, studentbasicinformation.studentaddress, subjectscholarshipinformation.scholarshipamount from subjectscholarshipinformation left join studentbasicinformation on studentbasicinformation.studentrollno = subjectscholarshipinformation.studentrollno where subjectscholarshipinformationscholarshipamount = 0;



13. Create Stored Procedure which will be return the amount balance to be paid by the student as per the student roll number passed through the stored procedure as the input

Solution:-

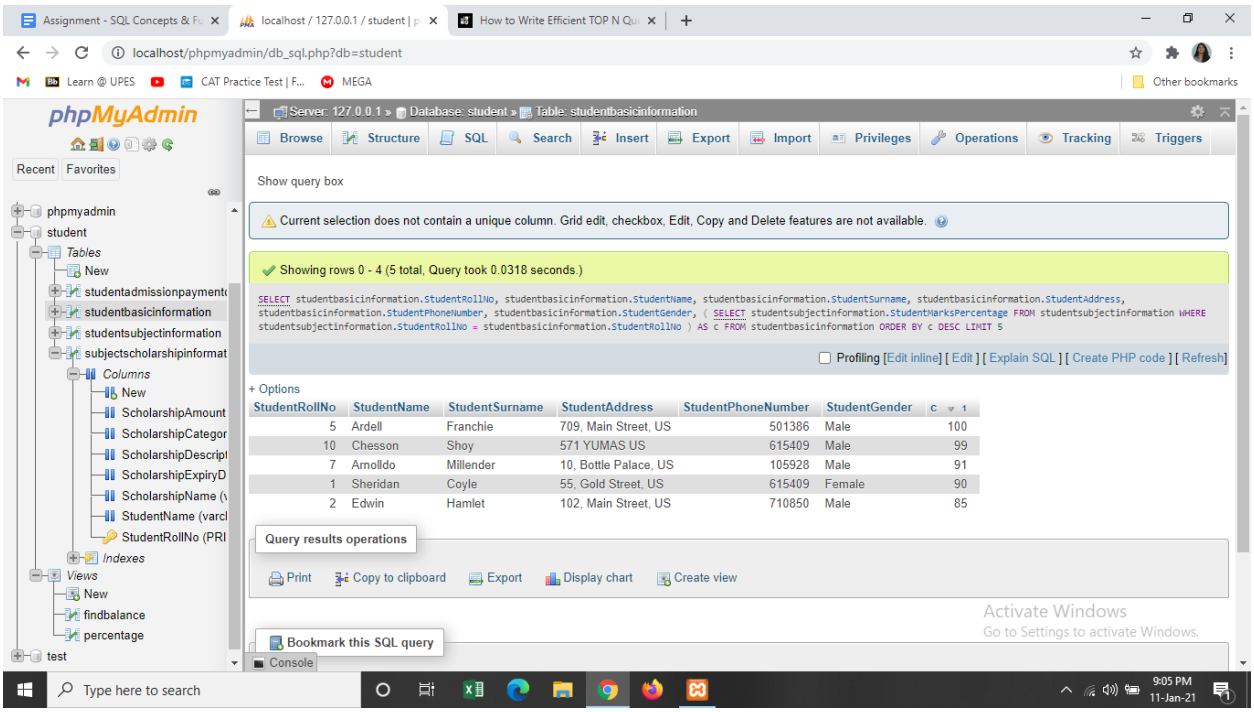
```
delimiter //
create procedure amountcheck ( in rollno int )
begin
select amountbalance from studentadmissionpayment where studentrollno=rollno;
end //
delimiter ;

call amountcheck(RollNo);
```



14. Retrieve the top five student details as per the StudentMarksPercentage values (use subqueries)

Solution:-

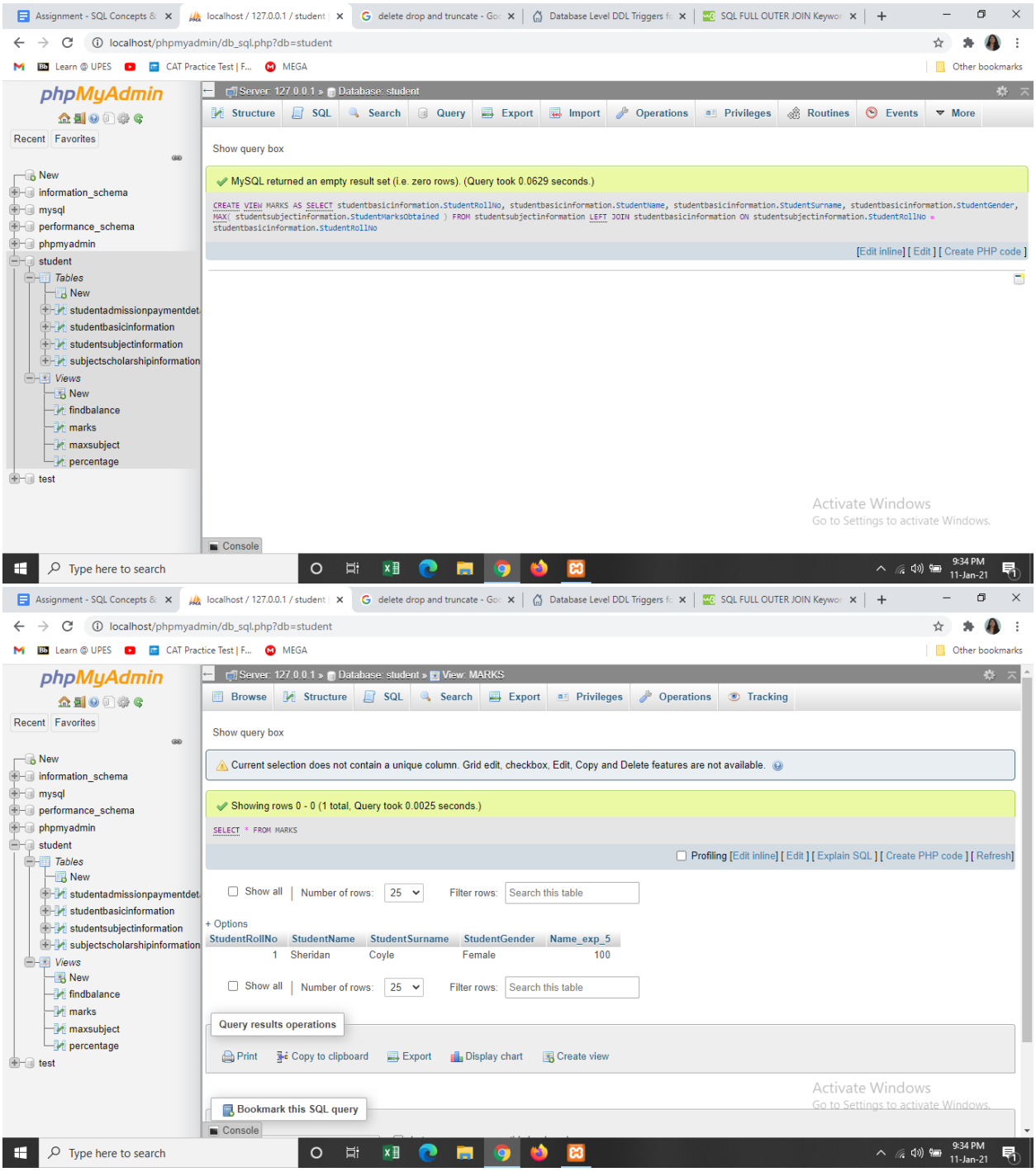
SELECT studentbasicinfromation.studentRollNo, studentbasicinformation.studentnane, studenitbasicinformation.Studertsurnane, studentbasicinformation.studentaddress, studentbasicinfromation.StudentPhoneNumber, studentbasicinformation.studentgender, (SELECT studentsubjectinformation.studentmarkspercentage FROM studentsubjectinformation WHERE studentsubjectinformation.studentrollno = studentbasicinformation.studentrollno) as C ROMR studentbasicinformationn ORDER BY C DESC LIMIT 5



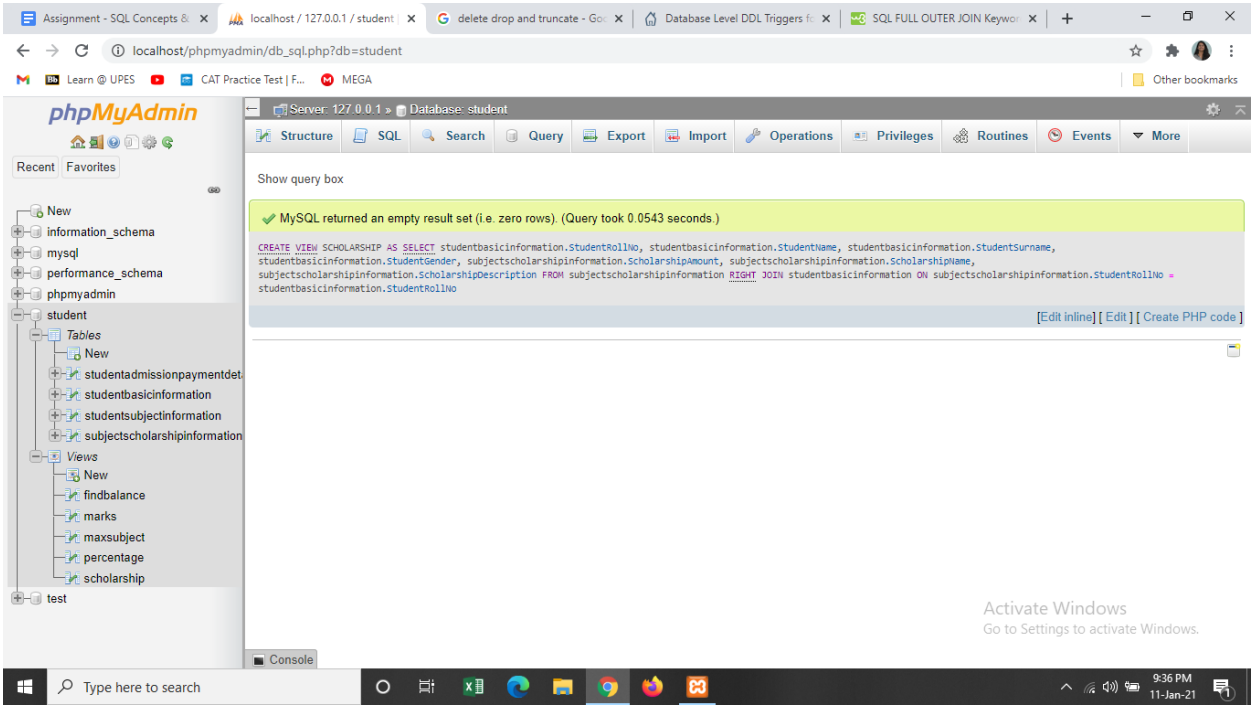
15. Try to use all the three types of join learned today in a relevant way, and explain the same why you thought of using that particular join for your selected scenarios (try to cover relevant and real time scenarios for all the three studied joins)

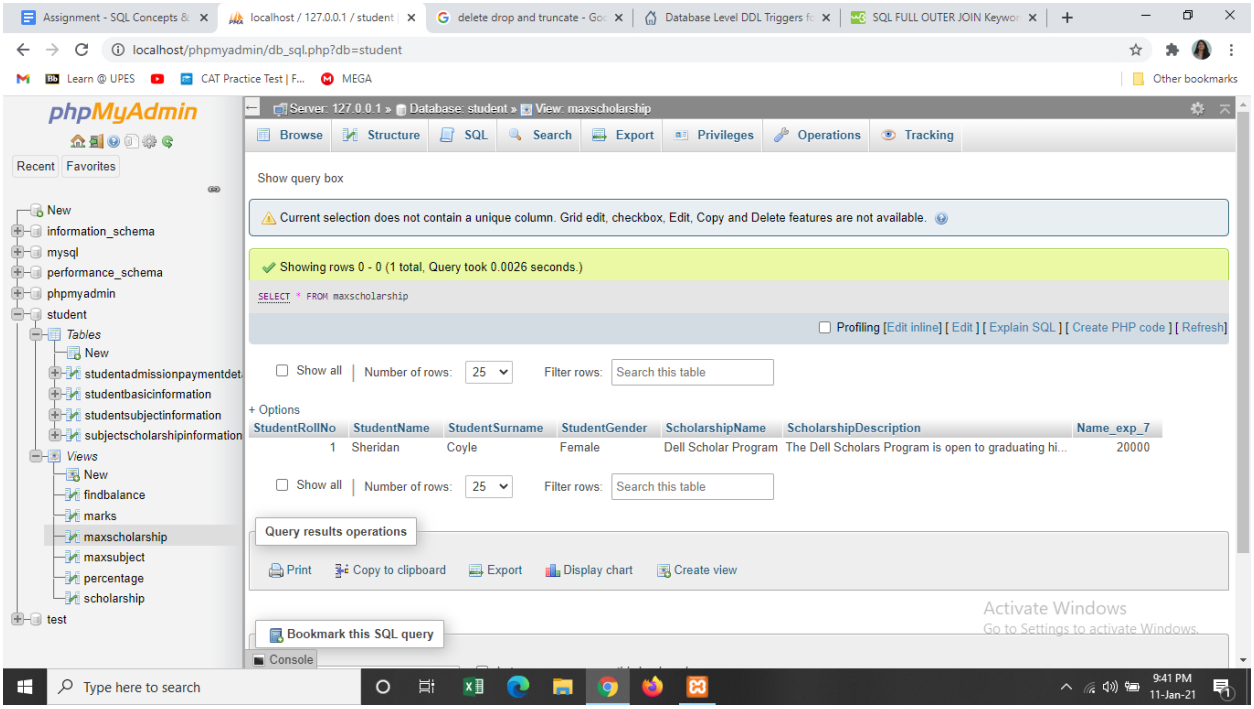
Solution:-

LEFT JOIN: Creating a view using left join so as to obtain the data of the student with maximum marks.

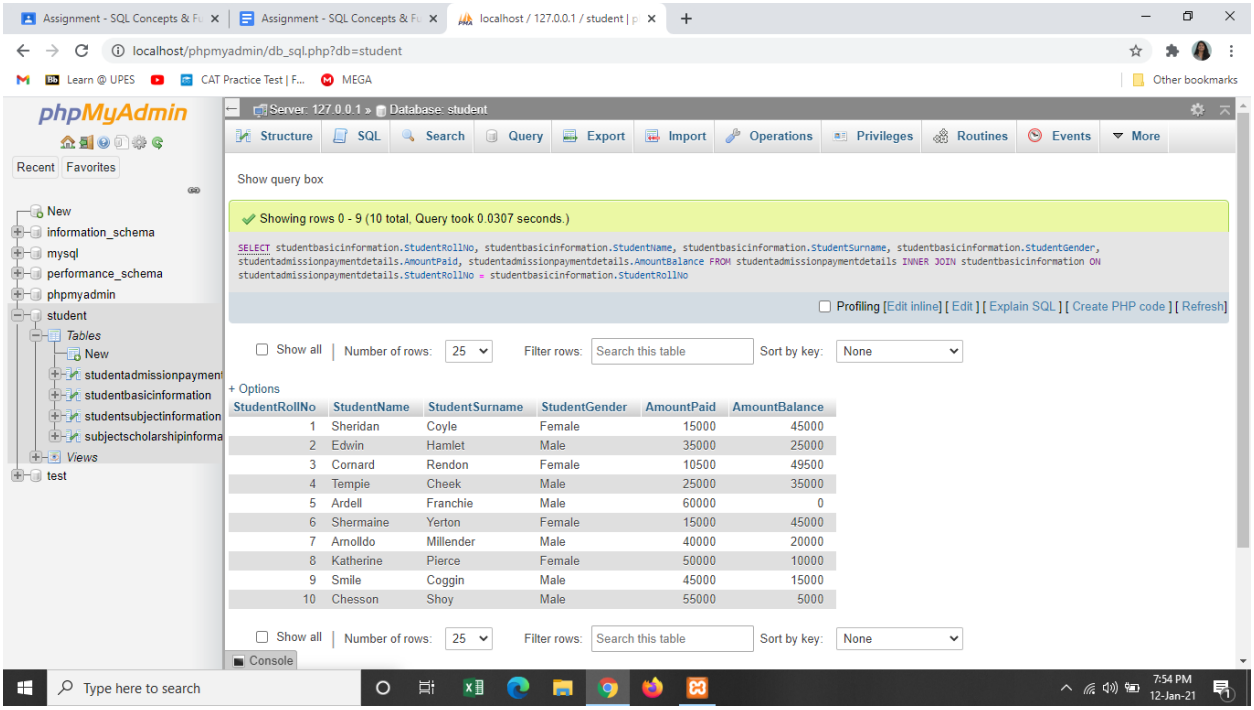


RIGHT JOIN: Creating a view using right join to obtain data of student with highest scholarship.





INNER JOIN: To retrieve data of students along with their paid fees and amount left to be paid.



16. Mention the differences between the delete, drop and truncate commands

Solution:-

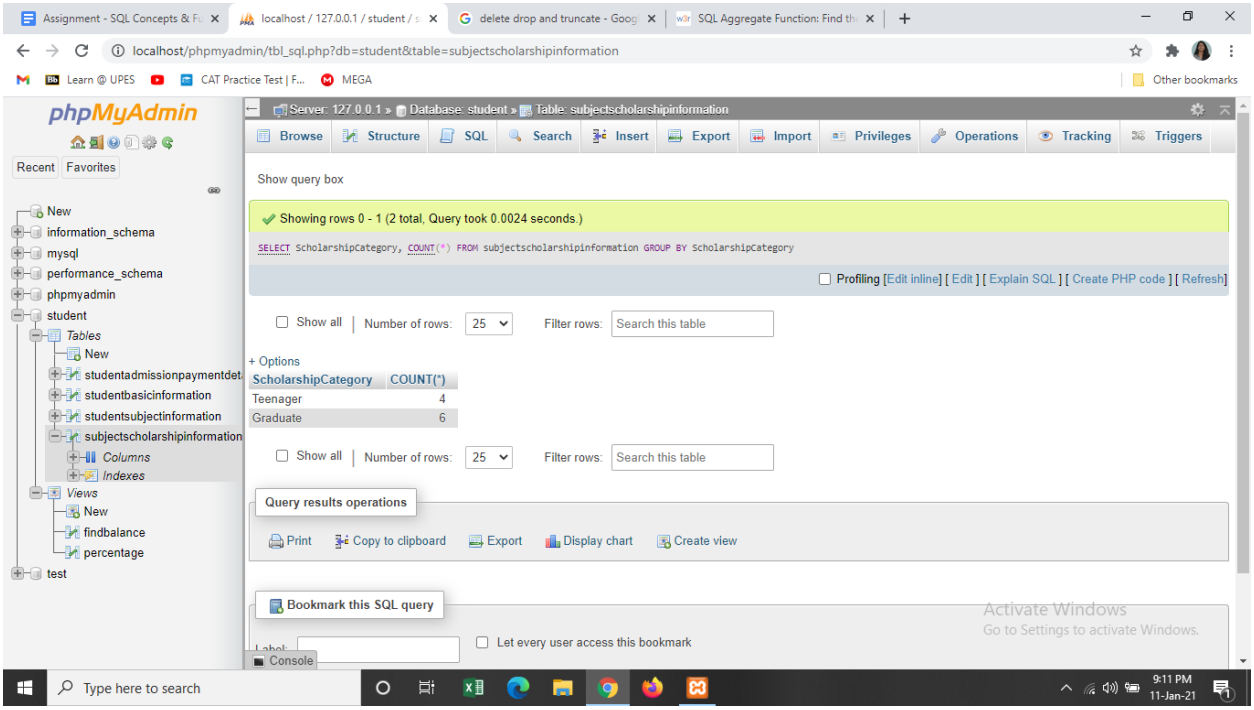
Parameter	DELETE	DROP	TRUNCATE
Type	DELETE is a Data Manipulation Language (DML) command	DROP is a Data Definition Language (DDL) command	TRUNCATE is a Data Definition Language (DDL) command
Use	It is used to delete the data in a table, but not the table structure.	It is used to delete the data and the table structure.	Like DELETE, It is used to delete the data and not the structure.
Single Row	We can delete a single row using WHERE clause	We cannot delete a single row only.	We cannot delete a single row as WHERE can't be used.
Rollback	We can use ROLLBACK to restore	We can't restore by using ROLLBACK	We can't restore by using ROLLBACK

Syntax	DELETE from STUDENT; DELETE from STUDENT where Age<18;	DROP table STUDENT;	TRUNCATE table STUDENT,

17. Get the count of the Scholarship category which is highly been availed by the students, i.e. get the count of the total number of students corresponding to the each scholarships category

    Solution:-

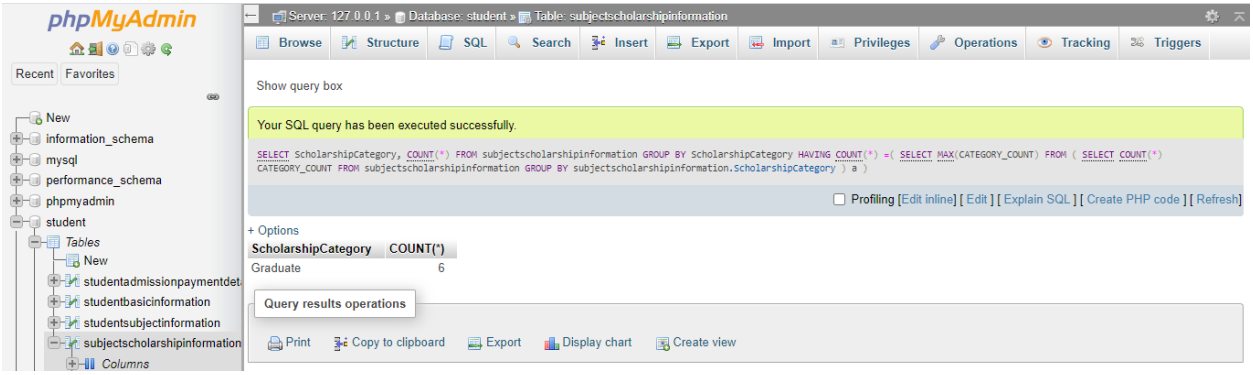
    select ScholarshipCategory,count(*) from studentscholarshipinformation group by ScholarshipCategory ;



18. Along with the assignment no. 17 try to retrieve the maximum used scholarship category

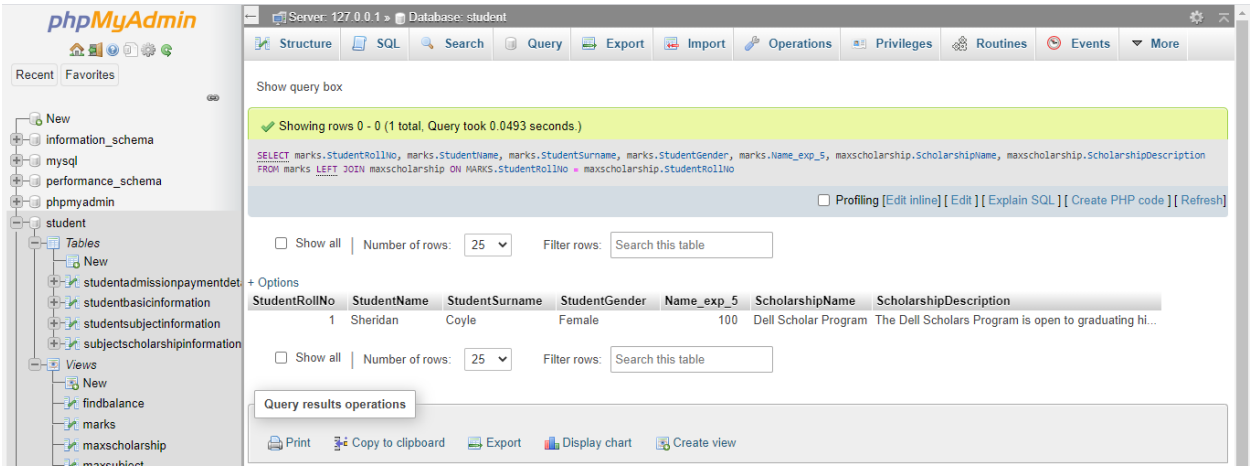
    Solution:-

    select ScholarshipCategory,count(*) from studentscholarshipinformation group by ScholarshipCategory having count(*) = (
    select max(CATEGORY_COUNT) from (
    select count(*) CATEGORY_COUNT from studentscholarshipinformation group by studentscholarshipinformation .scholarshipcategory
) a);



19. Retrieve the percentage of the students along with students detailed information who has scored the highest percentage along with availing the maximum scholarship amount
Solution:-

SELECT marks.studentrollNo, marks.Studentname, marks.Studentsurname, marks.studentgender, maxscholarship.scholarshipname, maxscholarship.scholarshipdescription FROM marks LEFT JOIN maxscholarship ON MARKS. Studentrollo = maxcholarship.tudentRollno;



20. Difference between the Triggers, Stored Procedures, Views and Functions

Solution:-

Triggers: A trigger runs automatically when an event occurs in the Database server.

1. Triggers run implicitly when an insert, update or delete is performed on a table.
2. We cannot pass any parameters in triggers.
3. They do not return any value.

Stored Procedure: A stored procedure is a collection of SQL statements that can be reused.

1. Stored procedure is called explicitly.
2. We can pass parameters in a stored procedure.
3. They may or may not return any value

Views: A view is a virtual table which may contain fields from one or more tables.

1. A view can be used in reference with the SELECT command.
2. It does not accept any parameters.
3. It is faster than stored procedures as it displays data from a table whereas stored procedure executes a series of sql commands.

Functions: User defined functions are a block of code that accept parameters, do some task and return a value.

1. We can call functions from a SELECT statement.
2. We can pass parameters in User defined functions.
3. Functions always return a value.