DBMS RECORD

NAME : SHILPA G

USN: 1BM20CS414

PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below. The data types are specified.

PERSON (driver\_id: String, name: String, address: String)

CAR (reg\_num: String, model: String, year: int)

ACCIDENT (report\_num: int, accident\_date: date, location: String)

OWNS (driver\_id: String, reg\_num: String)

PARTICIPATED (driver\_id: String,reg\_num: String, report\_num: int, damage\_amount: int)

i) Create the above tables by properly specifying the primary keys and the foreign keys.

ii)Enter at least five tuples for each relation.

iii)Demonstrate how you

a.Update the damage amount to 25000 for the car with a specific reg-num(example 'K A053408') for which the accident report number was 12.

b.Add a new accident to the database.

iv)Find the total number of people who owned cars that involved in accidents in 2008.

v)Find the number of accidents in which cars belonging to a specific model (example )were involved.

**QUERY 1: Create the above tables by properly specifying the primary keys and the foreign keys.**

create database insurance

create table insurance.person (driver\_id varchar(10), name varchar(20), address varchar(30), primary key(driver\_id))

desc insurance.person;

insert into person VALUES('01','manoj','belagavi');

INSERT INTO person VALUES('02','sudha','bellary');

INSERT INTO person VALUES('03','naveena','mangalore');

INSERT INTO person VALUES('04','bhoomi','srinivas nagar');

create table insurance.car(reg\_num varchar(10),model varchar(10),year int,primary key(reg\_num));

desc insurance.car;

INSERT INTO car VALUES('ka01mp140','zen',

2014);

INSERT INTO car VALUES('ka02al2818','baleno',

2016);

INSERT INTO car VALUES('ka41mb643','vitarabrezza',

2019);

INSERT INTO car VALUES('ka42ml983','audi',

2018

select \*from insurance.car;

create table insurance.accident(report\_num int,accident\_date date,location varchar(20),primary key(report\_num));

desc insurance.accident;

insert into accident values('1','2019-10-19','bangalore');

insert into accident values('2','2018-9-11','mangalore');

insert into accident values('3','2020-10-6','chitradurga');

insert into accident values('4','2019-9-12','bellary');

create table insurance.owns(driver\_id varchar(10),reg\_num

varchar(10),

primary key(driver\_id,reg\_num),

foreign key(driver\_id) references insurance.person(driver\_id),

foreign key(reg\_num) references insurance.car(reg\_num));

INSERT into owns VALUES('01','ka01mp140');

INSERT into owns VALUES('02','ka42ml983');

INSERT into owns VALUES('03','ka01mp140');

INSERT into owns VALUES('04','ka41mb643');

INSERT into owns VALUES('04','ka02al2818');

[select](http://127.0.0.1/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html" \t "/home/shilpa/Documents\\x/mysql_doc) \*from insurance.owns

create table insurance.participated(driver\_id

varchar(10), reg\_num varchar(10),

report\_num int, damage\_amount int,

primary key(driver\_id,reg\_num,report\_num),

foreign key(driver\_id) references person(driver\_id),

foreign key(reg\_num) references car(reg\_num),

foreign key(report\_num) references accident(report\_num));

desc insurance.participated;

**QUERY 2: Enter at least five tuples for each relation**

INSERT INTO participated VALUES(‘01’,’ka02al2818’,1,1000);

INSERT INTO participated VALUES('03','ka02al2818',3,9000);

INSERT into participated VALUES('04','ka41mb643',4,12000);

**QUERY 3:**

**a) Update the damage amount to 25000 for the car with a specific reg\_num (example 'K A053408' ) for which the accident report number was 12.**

update participated set damage\_amount=25000 where reg\_num='KA053408' and report\_num=12;

**b) Add a new accident to the database.**

insert into accident values(5,'2019-10-21','Bangalore');

**QUERY 4: Find the total number of people who owned cars that were involved in accidents in 2008.**

select count(distinct driver\_id) CNT from participated a, accident b where a.report\_num=b.report\_num and b.accident\_date like '2019%';

QUERY 5: Find the number of accidents in which cars belonging to a specific model (example 'Lancer') were involved.

select count(report\_num) CNT from car c,participated p where c.reg\_num=p.reg\_num and model='baleno';

## ADDITIONAL QUERIES:

1. **LIST THE ENTIRE PARTICIPATED RELATION IN THE DESCENDING ORDER OF DAMAGE AMOUNT.**

SQL> SELECT \* FROM PARTICIPATED ORDER BY DAMAGE\_AMOUNTT DESC;

1. **FIND THE AVERAGE DAMAGE AMOUNT**

SQL> SELECT AVG(DAMAGE\_AMOUNTT) FROM PARTICIPATED;

1. **DELETE THE TUPLE WHOSE DAMAGE AMOUNT IS BELOW THE AVERAGE DAMAGE AMOUNT**

SQL> DELETE FROM PARTICIPATED WHERE

DAMAGE\_AMOUNTT<(SELECT AVG (DAMAGE\_AMOUNT) FROM PARTICIPATED);

1. **LIST THE NAME OF DRIVERS WHOSE DAMAGE IS GREATER THAN THE AVERAGE DAMAGE AMOUNT.**

SQL> SELECT NAME FROM PERSON A, PARTICIPATED B WHERE A.DRIVER\_ID = B.DRIVER\_ID AND DAMAGE\_AMOUNT>(SELECT AVG(DAMAGE\_AMOUNT) FROM PARTICIPATED);

1. **FIND MAXIMUM DAMAGE AMOUNT.**

SQL>SELECT MAX(DAMAGE\_AMOUNT) FROM PARTICIPATED;

PROGRAM-2

create database banking;

create table banking.branch(branch\_name varchar(20),branch\_city varchar(20),assests int,primary key(branch\_name));

insert into banking.branch values('SBI\_basaveshwarnagar','bangalore',10000);

insert into banking.branch values('SBI\_marutinagar','bangalore',2000);

insert into banking.branch values('SBI\_shivajiroad','delhi',30000);

insert into banking.branch values('SBI\_jayantinagar','bellary',25000);

insert into banking.branch values('SBI\_chandralayout','bangalore',15000);

create table banking.bank\_account(accno int,branch\_name varchar(30),balance numeric,primary key(accno),FOREIGN key(branch\_name) REFERENCES banking.branch(branch\_name));

insert into banking.bank\_account values(1,'SBI\_shivajiroad',12000.00);

insert into banking.bank\_account values(2,'SBI\_shivajiroad',12000.00);

insert into banking.bank\_account values(3,'SBI\_jayantinagar',8000.50);

insert into banking.bank\_account values(4,'SBI\_chandralayout',2500.00);

insert into banking.bank\_account values(5,'SBI\_marutinagar',2500.00);

select \*from banking.bank\_account

create table banking.bank\_customer(customer\_name varchar(30),customer\_street varchar(30),city varchar(20),primary key(customer\_name));

insert into banking.bank\_customer values('Dhruthi','Bommanahalli','Bangalore');

insert into banking.bank\_customer values('Ananya','BG road','Bangalore');

insert into banking.bank\_customer values('Vimala','Ambedkar\_street','Bangalore');

insert into banking.bank\_customer values('Dheeraj','pankaj\_road','Delhi');

insert into banking.bank\_customer values('Spandana','RR\_nagar','Delhi');

CREATE TABLE depositer(cust\_name varchar(20),accno int,primary key(cust\_name,accno),foreign key(cust\_name)references bank\_customer(customer\_name),foreign key(accno) references bank\_account(accno));

insert into banking.depositer values('Dhruthi',1);

insert into banking.depositer values('Ananya',2);

insert into banking.depositer values('Vimala',3);

insert into banking.depositer values('Dheeraj',4);

insert into banking.depositer values('Spandana',5);

CREATE TABLE loan(loan\_no int,branch\_name varchar(20),amount int,primary key(loan\_no),foreign key(branch\_name)references branch(branch\_name));

insert into banking.loan values(01,SBI\_basaveshwarnagar',10000);

insert into banking.loan values(02,'SBI\_marutinagar',1000);

insert into banking.loan values(03,'SBI\_shivajiroad',9000);

insert into banking.loan values(04,'SBI\_jayantinagar',20000);

insert into banking.loan values(05,'SBI\_basaveshwarnagar',15000);

insert into banking.loan values(06,'SBI\_chandralayout',25000);

**QUERY 3: Find Find all the customers who have at least two deposits at the same branch**

use banking;

select c.customer\_name

from bank\_customer c

where exists(

select d.cust\_name

from depositer d, bank\_account ba

where

d.accno=ba.accno and

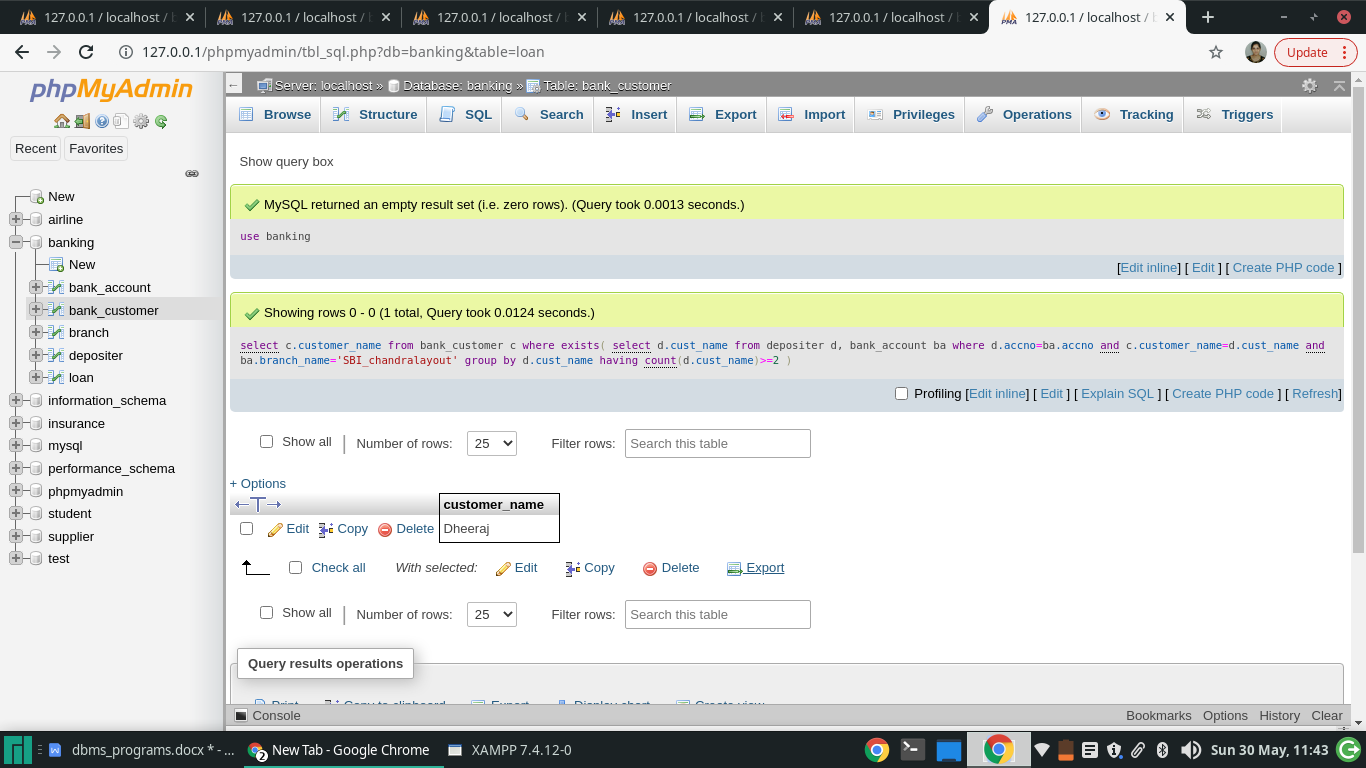
c.customer\_name=d.cust\_name and

ba.branch\_name='SBI\_chandralayout'

group by d.cust\_name

having count(d.cust\_name)>=2

);



**QUERY 4: Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).**

select d.cust\_name from depositer d,branch b,bank\_account ba

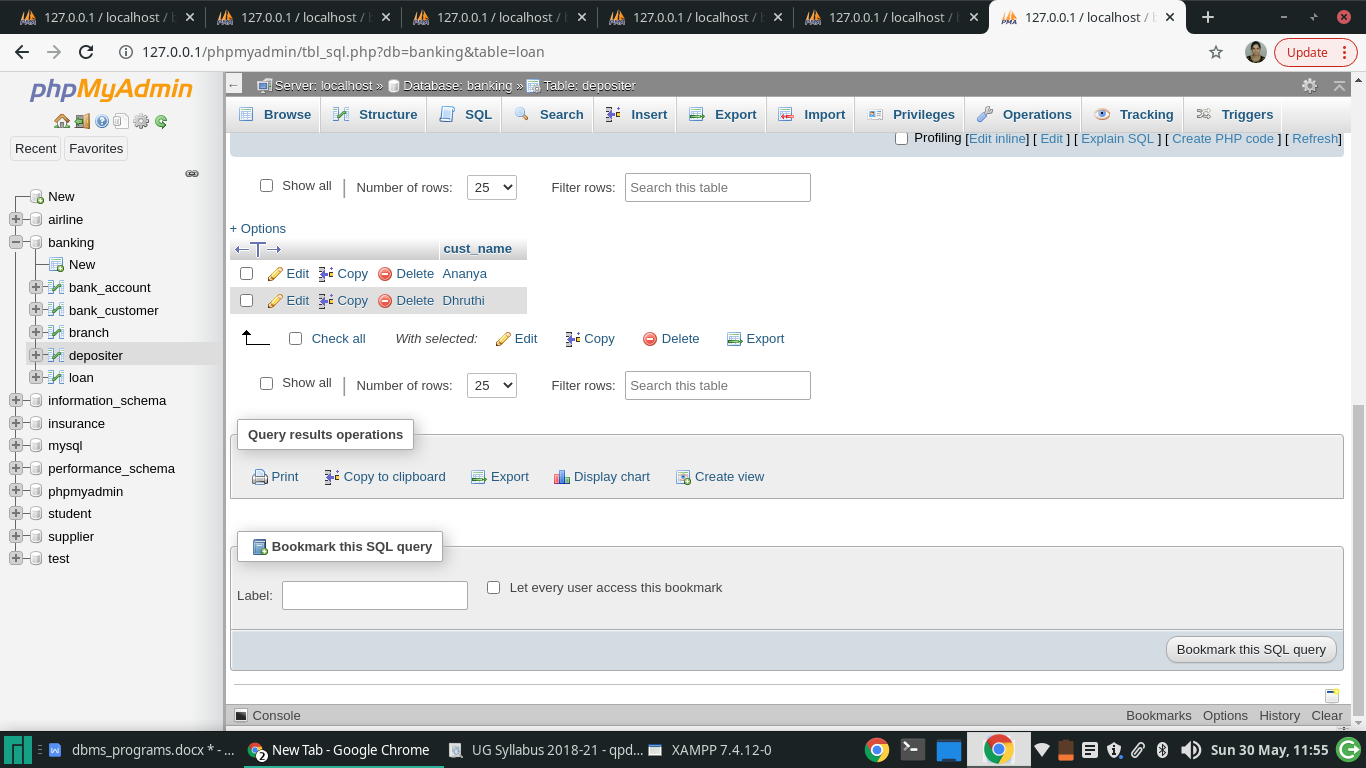
where b.branch\_name=ba.branch\_name

AND ba.accno=d.accno

and branch\_city='Delhi'

group by d.cust\_name

having COUNT(distinct b.branch\_name)=(select COUNT(branch\_name) from branch where branch\_city='Delhi');



**QUERY 5: Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bomay).**

use banking;

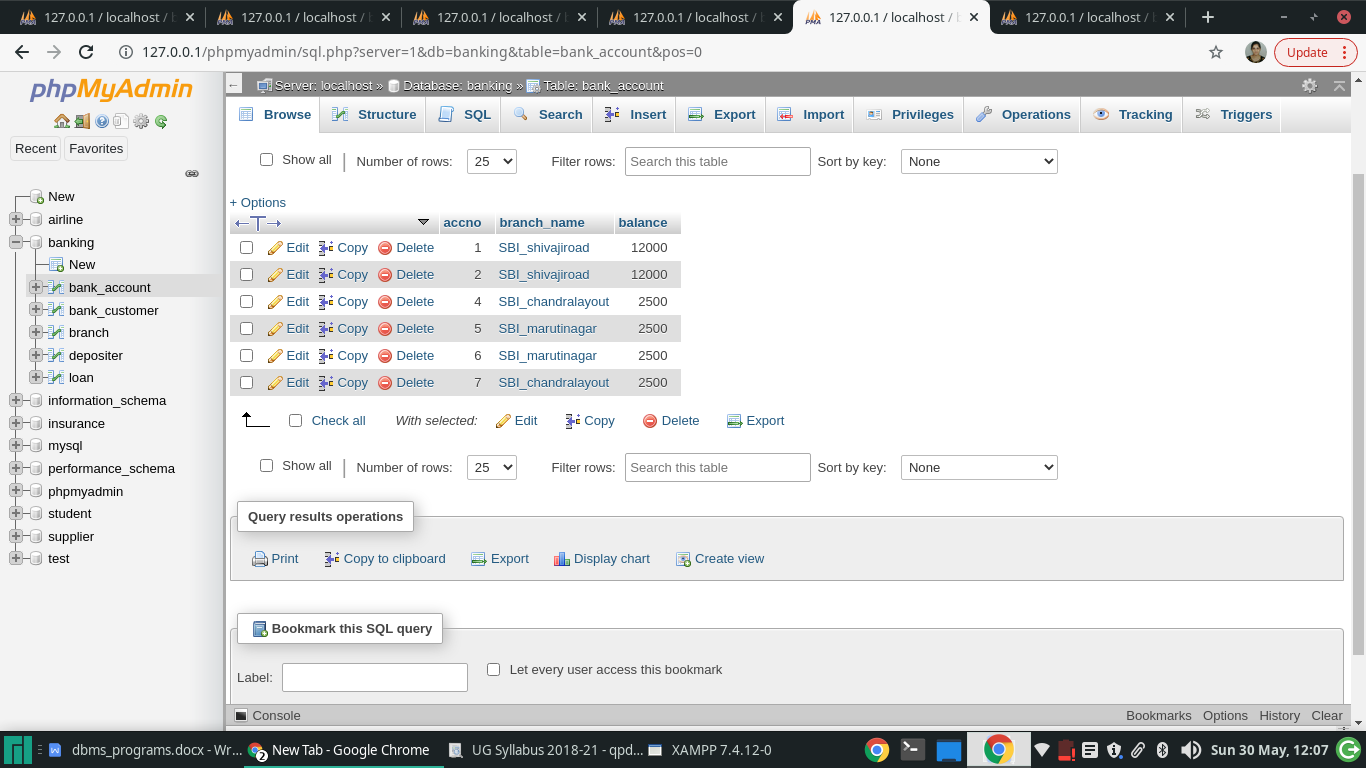
delete from bank\_account

where branch\_name IN(

select branch\_name

from branch

where branch\_city='bellary');



## ADDITIONAL QUERIES:

1. **LIST THE ENTIRE LOAN RELATION IN THE DESCENDING ORDER OF AMOUNT.**

SQL> SELECT \* FROM LOAN ORDER BY AMOUNT DESC;

1. **FIND ALL CUSTOMERS HAVING A LAON, AN ACCOUNT OR BOTH AT THE BANK**

SQL> (SELECT CUSTOMER\_NAME FROM DEPOSITOR ) UNION (SELECT CUSTOMER\_NAME FROM BORROWER);

1. **CREATE A VIEW WHICH GIVES EACH BRANCH THE SUM OF THE AMOUNT OF ALL THE LOANS AT THE BRANCH.**

SQL> CREATE VIEW BRANCH\_TOTAL\_LOAN (BRANCH\_NAME, TOTAL\_LOAN) AS SELECT BRANCH\_NAME, SUM(AMOUNT) FROM LOAN GROUP BY BRANCH\_NAME;

1. **THE ANNUAL INTEREST PAYMENTS ARE MADE AND ALL BRANCHES ARE TO BE INCREASED BY 5%.**

SQL> UPDATE ACCOUNT SET BALANCE=BALANCE \*1.05;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3rd program\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**PROGRAM 3: SUPPLIER DATABASE**

**Consider the following schema:**

**SUPPLIERS(sid: integer, sname: string, address: string)**

**PARTS(pid: integer, pname: string, color: string)**

**CATALOG(sid: integer, pid: integer, cost: real)**

**The Catalog relation lists the prices charged for parts by Suppliers.**

**Write the following queries in SQL:**

1. Find the pnames of parts for which there is some supplier.
2. Find the snames of suppliers who supply every part.
3. Find the snames of suppliers who supply every red part.
4. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
5. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
6. For each part, find the sname of the supplier who charges the most for that part.

use supplier;

create table SUPPLIERS(sid integer(5) primary key, sname varchar(20), city varchar(20));

use supplier;

create table PARTS(pid integer(5) primary key, pname varchar(20), color varchar(10));

use supplier;

create table CATALOG(sid integer(5), pid integer(5), foreign key(sid) references SUPPLIERS(sid), foreign key(pid) references PARTS(pid), cost float(6), primary key(sid, pid));

use supplier;

insert into SUPPLIERS values(10001, 'Acme Widget','Bangalore');

insert into SUPPLIERS values(10002, 'Johns','Kolkata');

insert into SUPPLIERS values(10003, 'Vimal','Mumbai');

insert into SUPPLIERS values(10004, 'Reliance','Delhi');

insert into SUPPLIERS values(10005, 'Mahindra','Mumbai');

use supplier;

insert into PARTS values(20001, 'Book','Red');

insert into PARTS values(20002, 'Pen','Red');

insert into PARTS values(20003, 'Pencil','Green');

insert into PARTS values(20004, 'Mobile','Green');

insert into PARTS values(20005, 'Charger','Black');

use supplier;

insert into CATALOG values(10001, '20001','10');

insert into CATALOG values(10001, '20002','10');

insert into CATALOG values(10001, '20003','30');

insert into CATALOG values(10001, '20004','10');

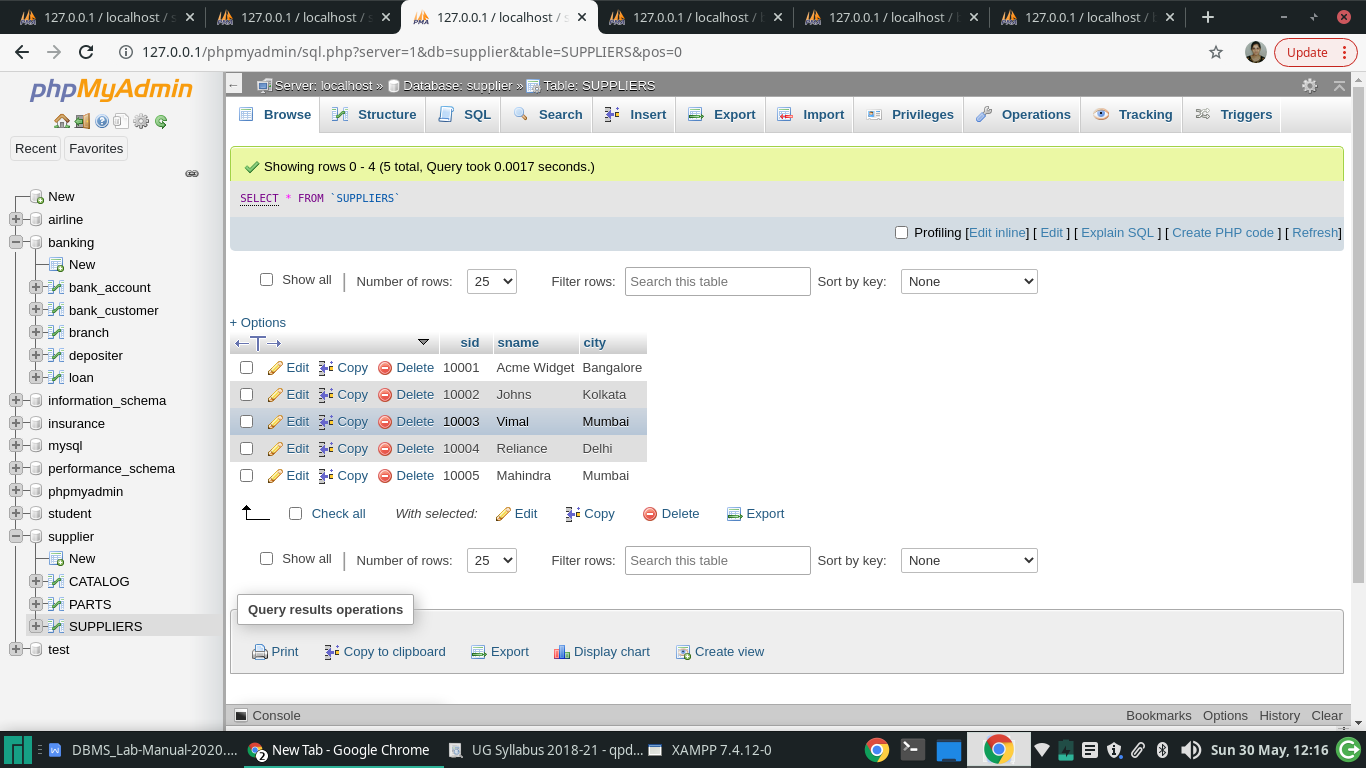
insert into CATALOG values(10001, '20005','10');

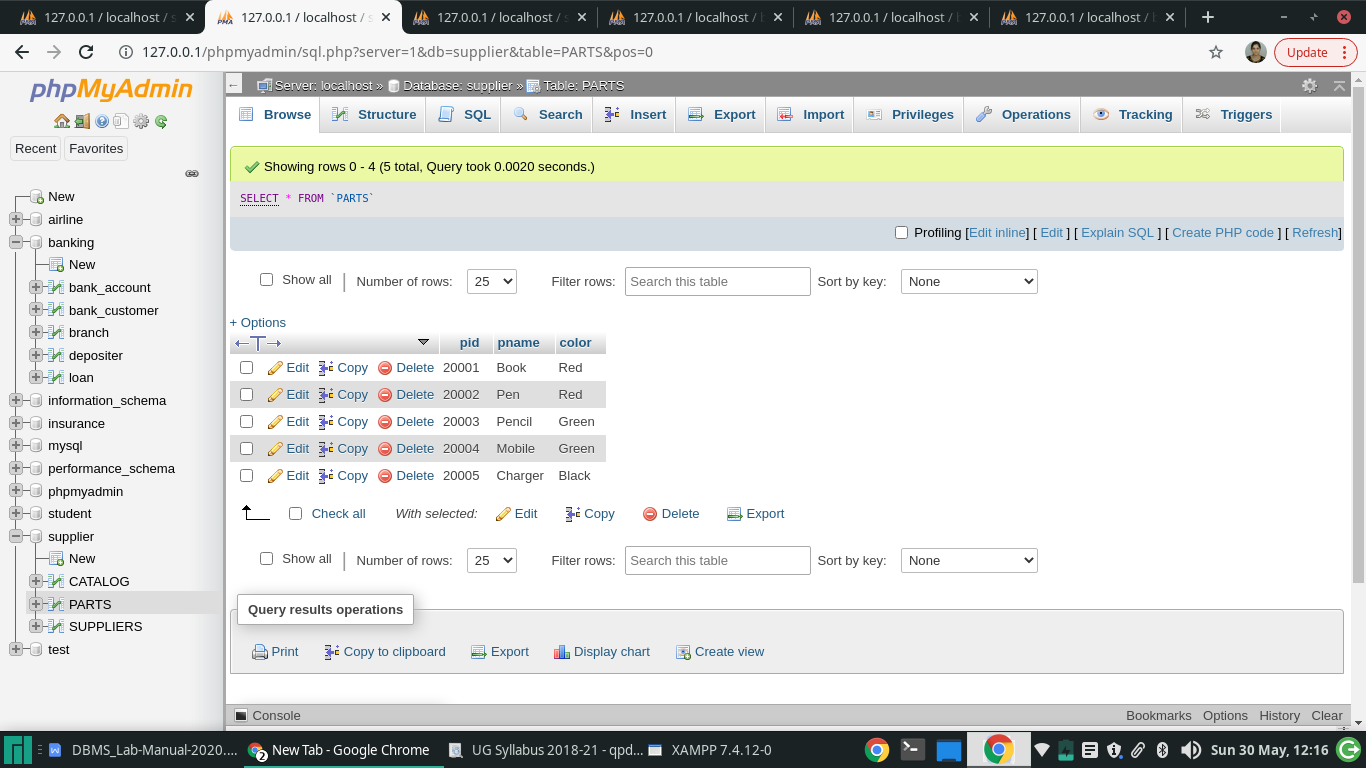
insert into CATALOG values(10002, '20001','10');

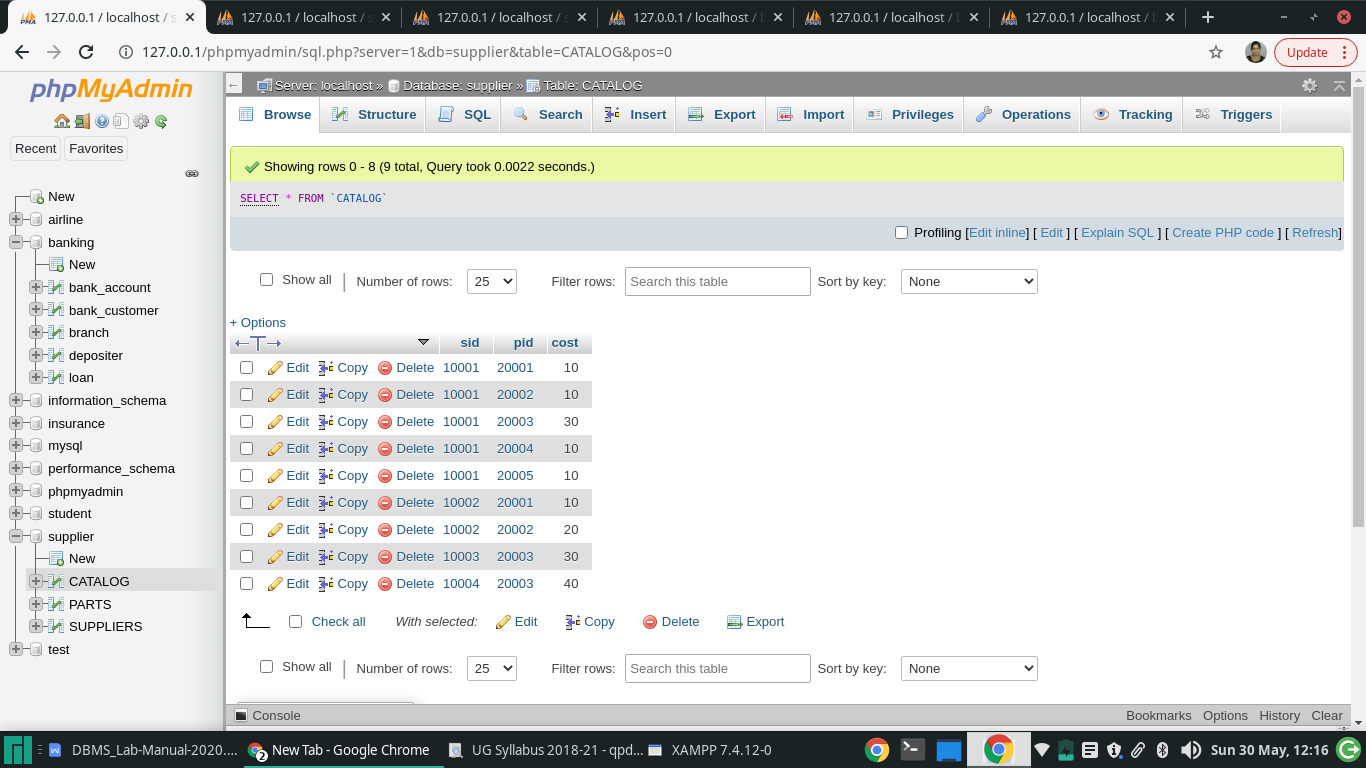
insert into CATALOG values(10002, '20002','20');

insert into CATALOG values(10003, '20003','30');

insert into CATALOG values(10004, '20003','40');





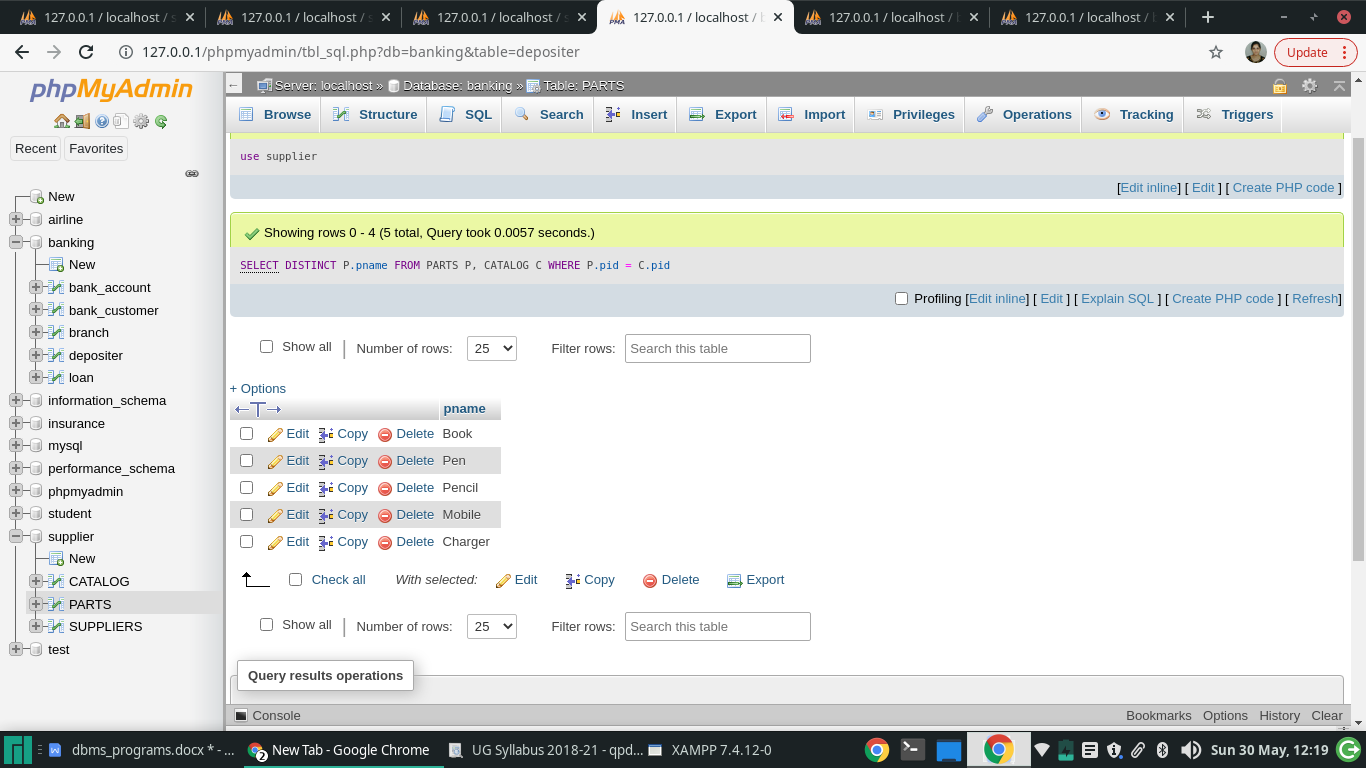


1. **Find the pnames of parts for which there is some supplier.**

use supplier;

select pname from PARTS p,CATALOG c WHERE c.pid=p.pid group by (c.pid);

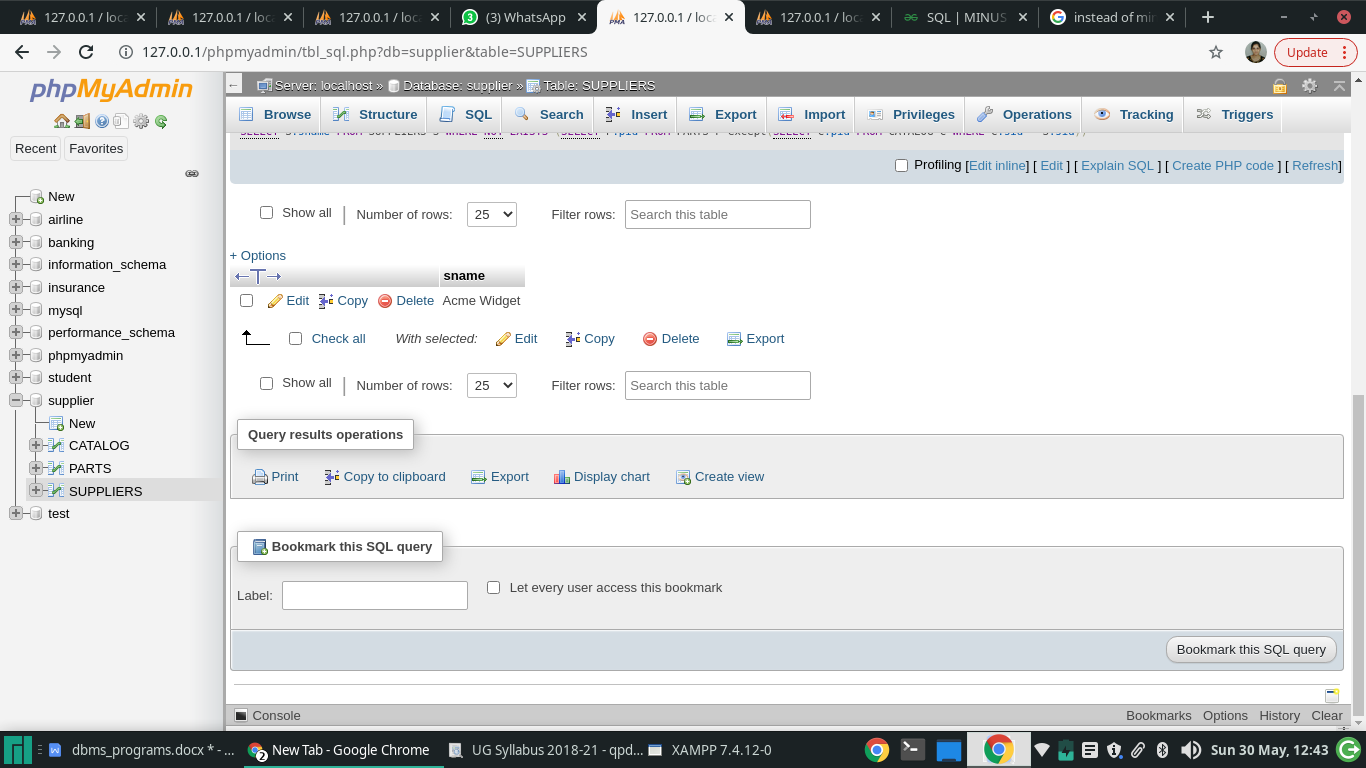
SELECT DISTINCT P.pname FROM PARTS P, CATALOG C WHERE P.pid = C.pid;



1. **Find the snames of suppliers who supply every part.**

use supplier;

SELECT S.sname FROM SUPPLIERS S WHERE NOT EXISTS (SELECT P.pid FROM PARTS P except(SELECT C.pid FROM CATALOG C WHERE C.sid = S.sid));



1. **Find the snames of suppliers who supply every red part.**

use supplier;

SELECT S.sname

FROM SUPPLIERS S

WHERE NOT EXISTS (( SELECT P.pid

FROM PARTS P

WHERE P.color = 'Red' )

EXCEPT

( SELECT C.pid

FROM CATALOG C, PARTS P

WHERE C.sid = S.sid AND

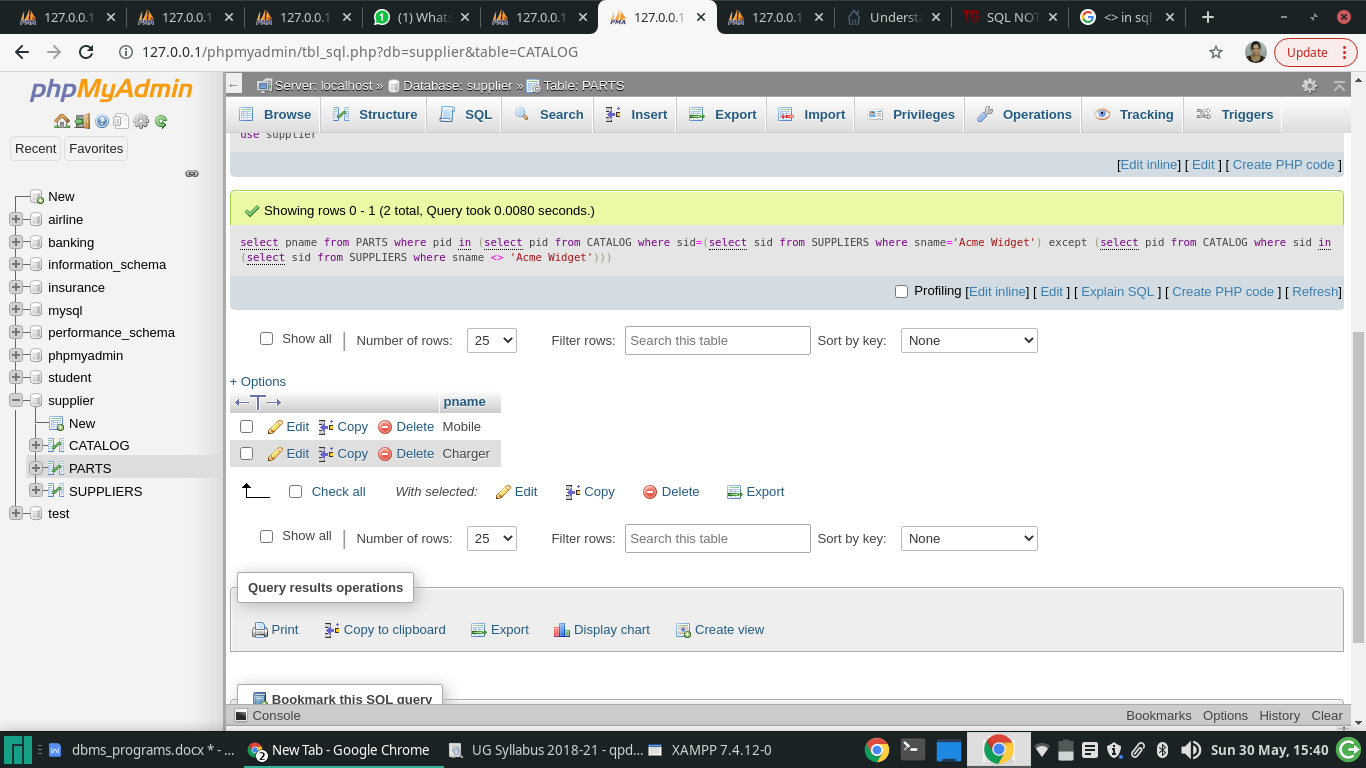
C.pid = P.pid AND P.color = 'Red' ));



1. **Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.**

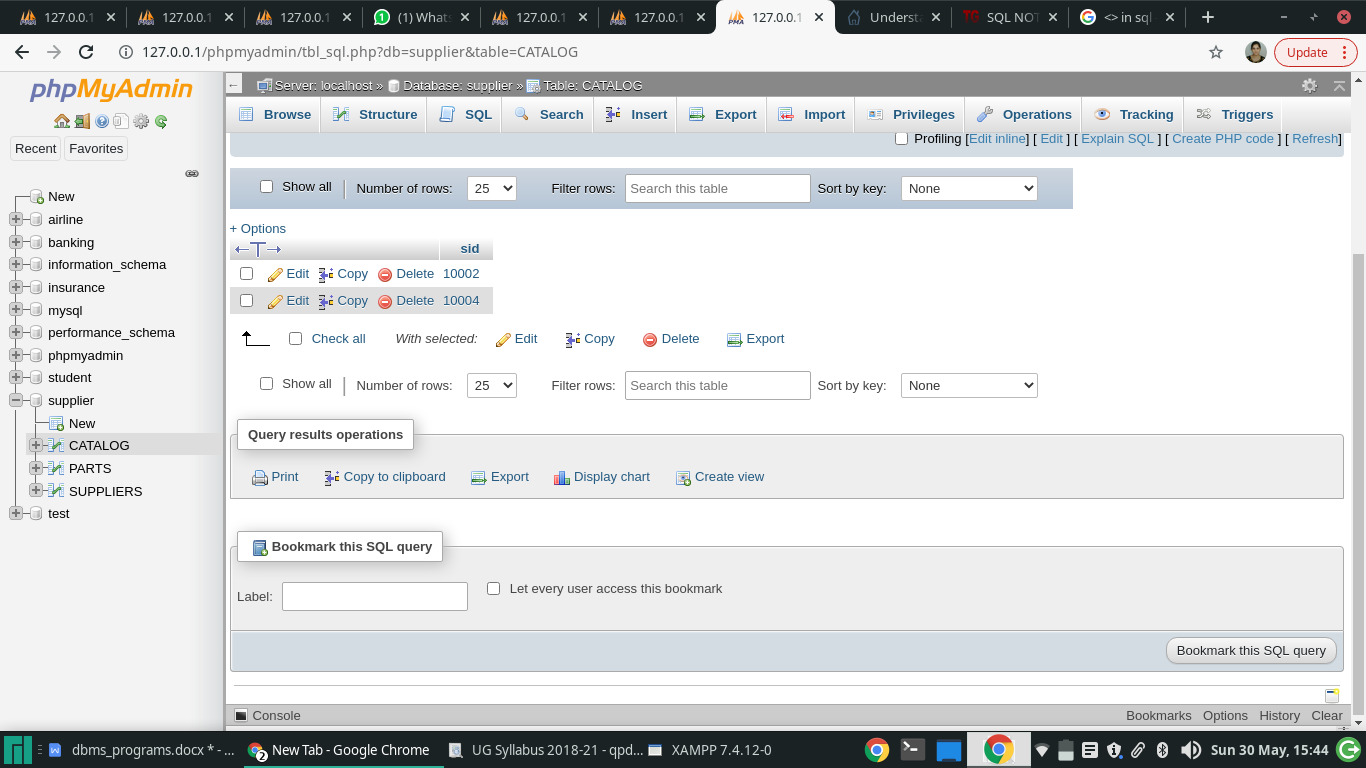
use supplier;

select pname from PARTS where pid in (select pid from CATALOG where sid=(select sid from SUPPLIERS where sname='Acme Widget') except (select pid from CATALOG where sid in (select sid from SUPPLIERS where sname <> 'Acme Widget')));



1. **Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).**

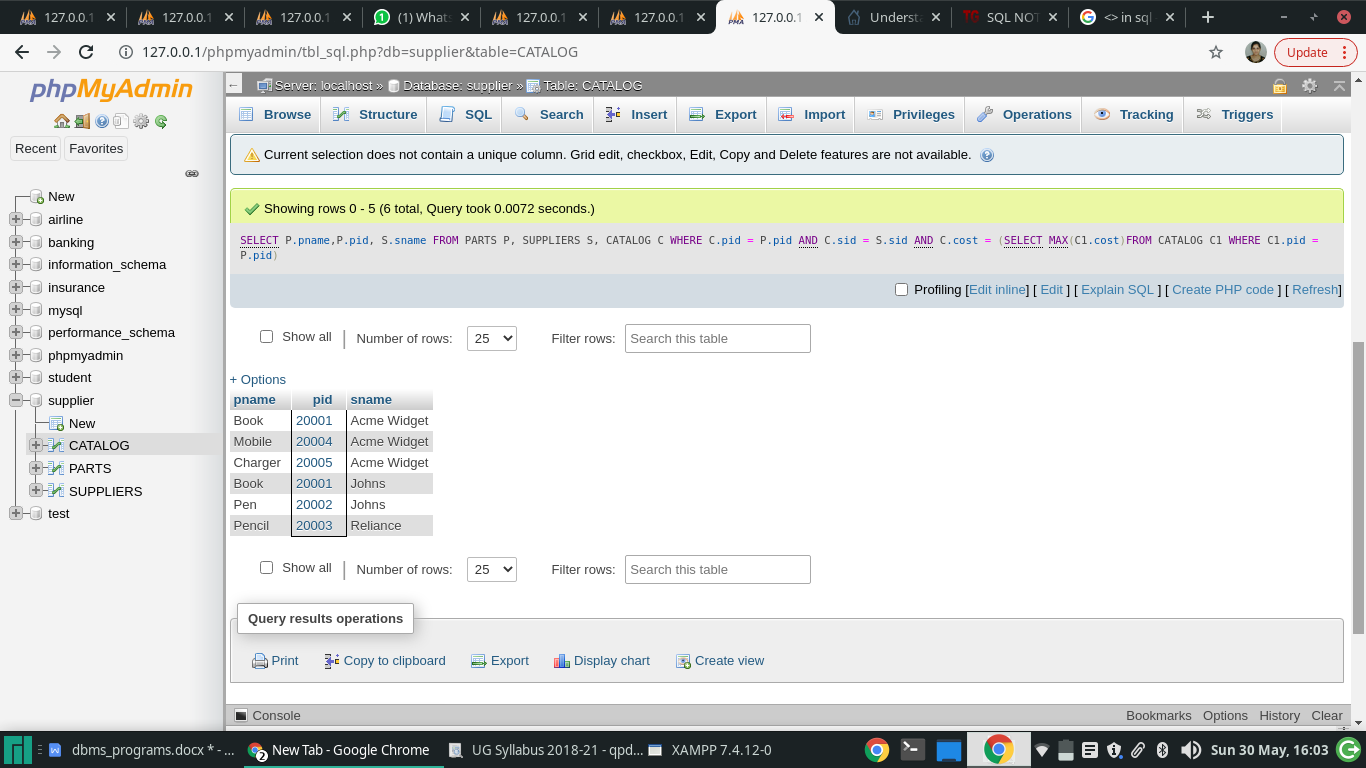
SELECT DISTINCT C.sid FROM CATALOG C WHERE C.cost > ( SELECT AVG (C1.cost) FROM CATALOG C1 WHERE C1.pid = C.pid );



1. **For each part, find the sname of the supplier who charges the most for that part.**

use supplier;

SELECT P.pname,P.pid, S.sname FROM PARTS P, SUPPLIERS S, CATALOG C WHERE C.pid = P.pid AND C.sid = S.sid AND C.cost = (SELECT MAX(C1.cost)FROM CATALOG C1 WHERE C1.pid = P.pid);



1. Find the sids of suppliers who supply only red parts.

use supplier;

select s.sid,s.sname from SUPPLIERS s where s.sid in(select c.sid from CATALOG c,PARTS p WHERE c.pid=p.pid and p.pname='Red');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*program 4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**PROGRAM 4: STUDENT FACULTY DATABASE**

**Consider the following database for student enrollment for course :**

**STUDENT(snum: integer, sname: string, major: string, lvl: string, age: integer)**

**CLASS(cname: string, meets at: time, room: string, fid: integer)**

**ENROLLED(snum: integer, cname: string)**

**FACULTY(fid: integer, fname: string, deptid: integer)**

**The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character code with 4 different values (example: Junior: JR etc)**

**Write the following queries in SQL. No duplicates should be printed in any of the answers.**

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by
2. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
3. Find the names of all students who are enrolled in two classes that meet at the same time.
4. Find the names of faculty members who teach in every room in which some class is taught.
5. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.
6. Find the names of students who are not enrolled in any class.
7. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

create database student;

CREATE TABLE student(snum INT,sname VARCHAR(10),major VARCHAR(2),lvl VARCHAR(2),age INT, primary key(snum));

CREATE TABLE faculty(fid INT,fname VARCHAR(20),deptid INT,PRIMARY KEY(fid));

CREATE TABLE class(cname VARCHAR(20),metts\_at TIMESTAMP,room VARCHAR(10),fid INT, PRIMARY KEY(cname), FOREIGN KEY(fid) REFERENCES faculty(fid));

use student;

CREATE TABLE enrolled(snum INT,cname VARCHAR(20),PRIMARY KEY(snum,cname),FOREIGN KEY(snum) REFERENCES student(snum),FOREIGN KEY(cname) REFERENCES class(cname));

use student;

INSERT INTO student VALUES(1, 'jhon', 'CS', 'Sr', 19);

INSERT INTO student VALUES(2, 'Smith', 'CS', 'Jr', 20);

INSERT INTO student VALUES(3 , 'Jacob', 'CV', 'Sr', 20);INSERT INTO student VALUES(4, 'Tom ', 'CS', 'Jr', 20)

;

INSERT INTO student VALUES(5, 'Rahul', 'CS', 'Jr', 20);

INSERT INTO student VALUES(6, 'Rita', 'CS', 'Sr', 21);

use student;

INSERT INTO faculty VALUES(11, 'Harish', 1000);

INSERT INTO faculty VALUES(12, 'MV', 1000);

INSERT INTO faculty VALUES(13 , 'Mira', 1001);

INSERT INTO faculty VALUES(14, 'Shiva', 1002);

INSERT INTO faculty VALUES(15, 'Nupur', 1000);

use student;

insert into class values('class1', '12/11/15 10:15:16', 'R1', 14);

insert into class values('class10', '12/11/15 10:15:16', 'R128', 14);

insert into class values('class2', '12/11/15 10:15:20', 'R2', 12);

insert into class values('class3', '12/11/15 10:15:25', 'R3', 12);

insert into class values('class4', '12/11/15 20:15:20', 'R4', 14);

insert into class values('class5', '12/11/15 20:15:20', 'R3', 15);

insert into class values('class6', '12/11/15 13:20:20', 'R2', 14);

insert into class values('class7', '12/11/15 10:10:10', 'R3', 14);

use student;

insert into enrolled values(1, 'class1');

insert into enrolled values(2, 'class1');

insert into enrolled values(3, 'class3');

insert into enrolled values(4, 'class3');

insert into enrolled values(5, 'class4');

insert into enrolled values(1, 'class5');

insert into enrolled values(2, 'class5');

insert into enrolled values(3, 'class5');

insert into enrolled values(4, 'class5');

insert into enrolled values(5, 'class5');

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*no\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8

**1)Find the names of all Juniors (level(lvl) = Jr) who are enrolled in a class taught by Harish.**

use student;

SELECT DISTINCT S.Sname

FROM student S, class C, enrolled E, faculty F

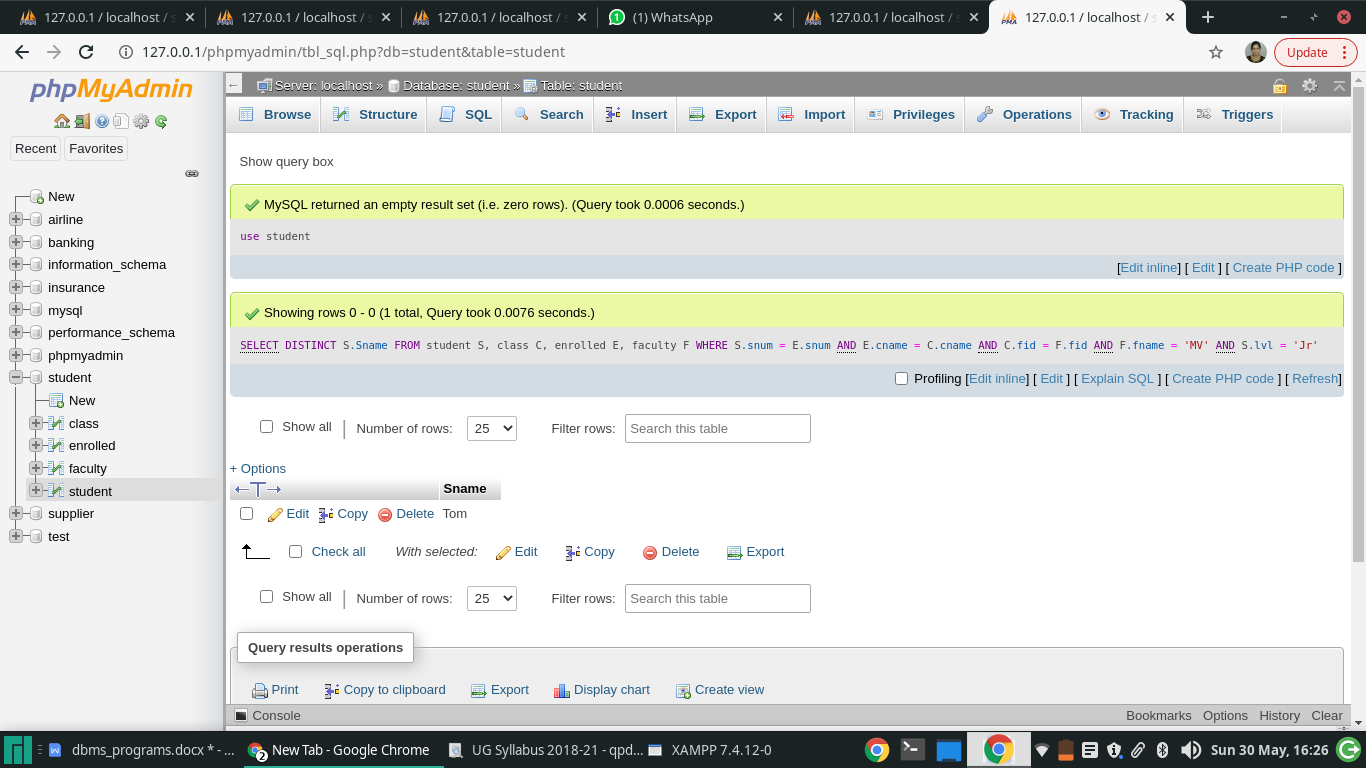
WHERE S.snum = E.snum

AND E.cname = C.cname

AND C.fid = F.fid

AND F.fname = 'MV'

AND S.lvl = 'Jr';



2)**Find the names of all classes that either meet in room R128 or have five or more Students enrolled.**

use student;

SELECT C.cname

FROM class C

WHERE C.room='R128'

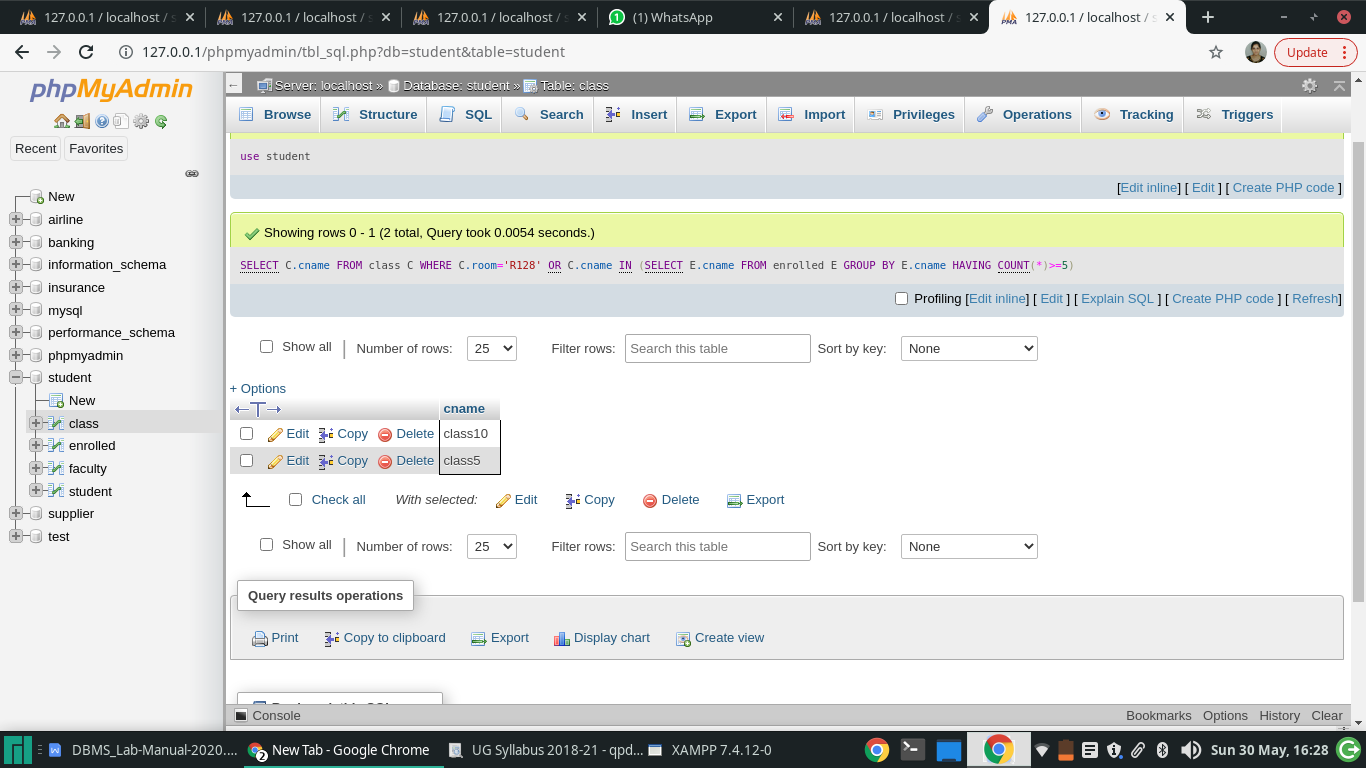
OR C.cname IN

(SELECT E.cname

FROM enrolled E

GROUP BY E.cname

HAVING COUNT(\*)>=5;



3)Find the names of all students who are enrolled in two classes that meet at the same time.

use student;

SELECT DISTINCT S.sname

FROM student S

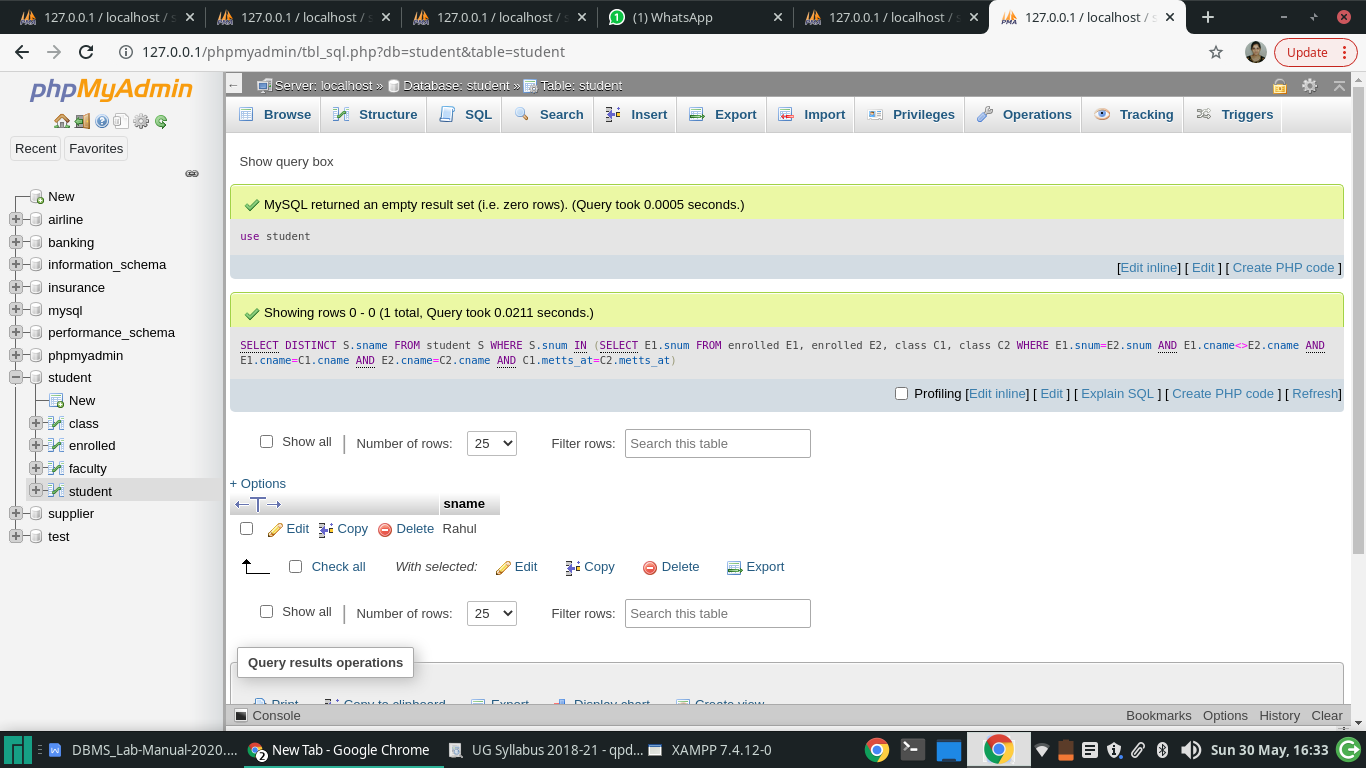
WHERE S.snum IN (SELECT E1.snum

FROM enrolled E1, enrolled E2, class C1, class C2

WHERE E1.snum=E2.snum AND E1.cname<>E2.cname

AND E1.cname=C1.cname

AND E2.cname=C2.cname AND C1.metts\_at=C2.metts\_at);



4)**Find the names of faculty members who teach in every room in which some class is taught.**

use student;

SELECT DISTINCT F.fname

FROM faculty F

WHERE NOT EXISTS

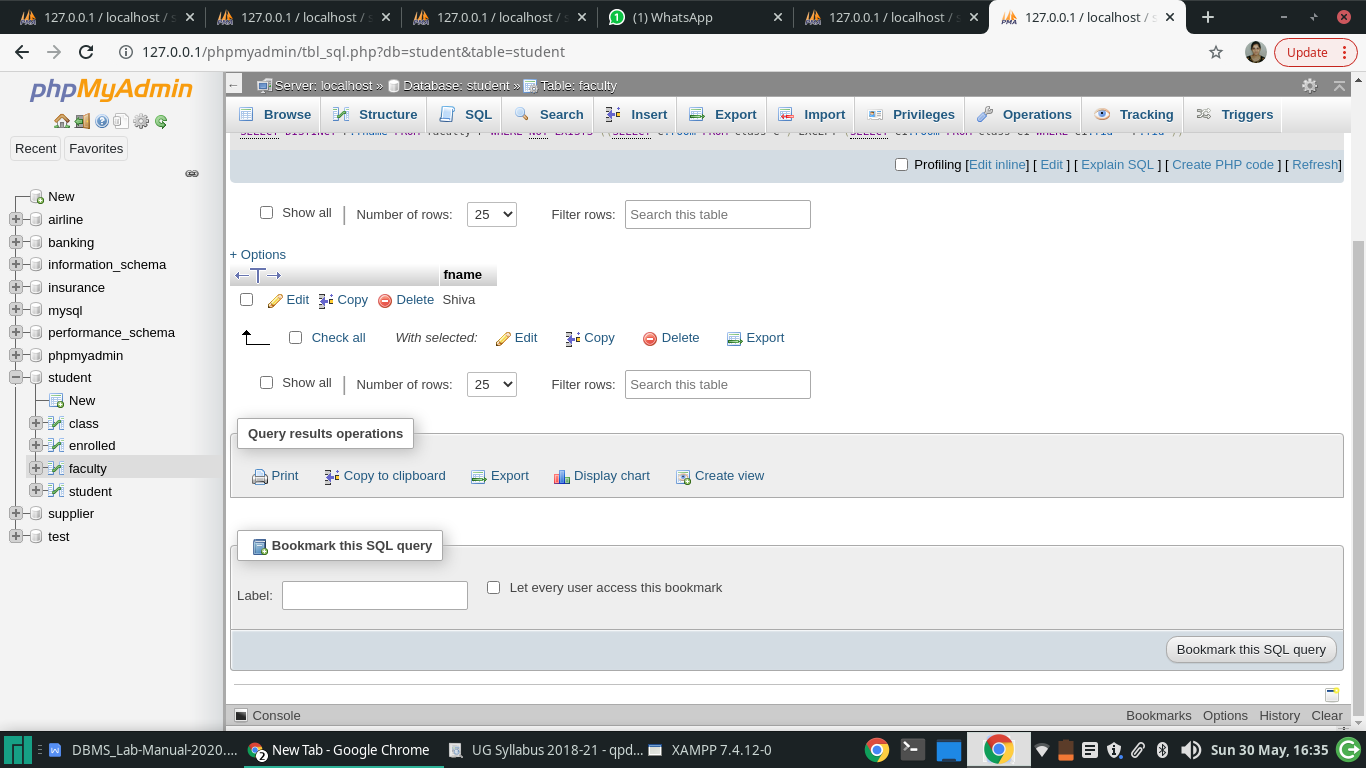
((SELECT C.room FROM class C )

EXCEPT

(SELECT C1.room

FROM class C1

WHERE C1.fid = F.fid ));



1. **Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.**

**use student;**

**SELECT DISTINCT F.fname**

**FROM faculty F**

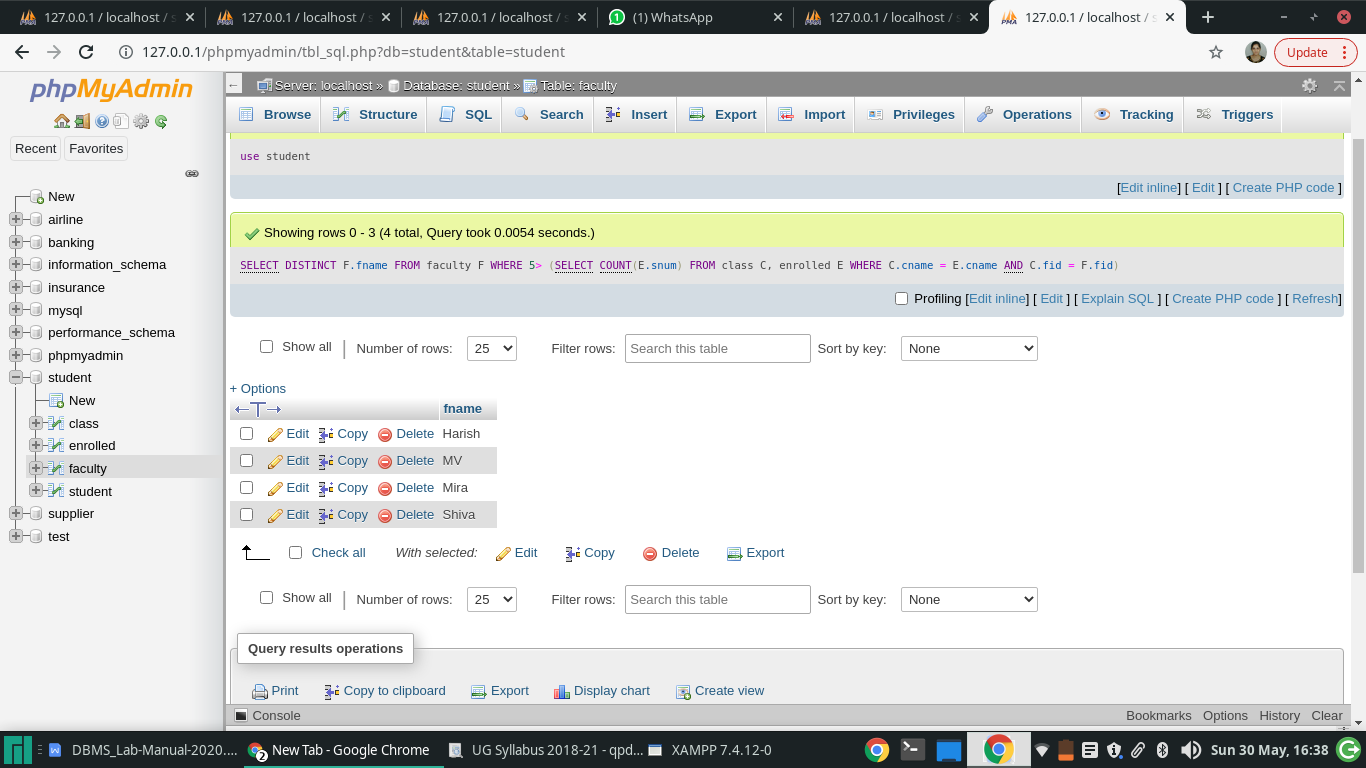
**WHERE 5>**

**(SELECT COUNT(E.snum)**

**FROM class C, enrolled E**

**WHERE C.cname = E.cname**

**AND C.fid = F.fid);**



7)**Find the names of students who are not enrolled in any class.**

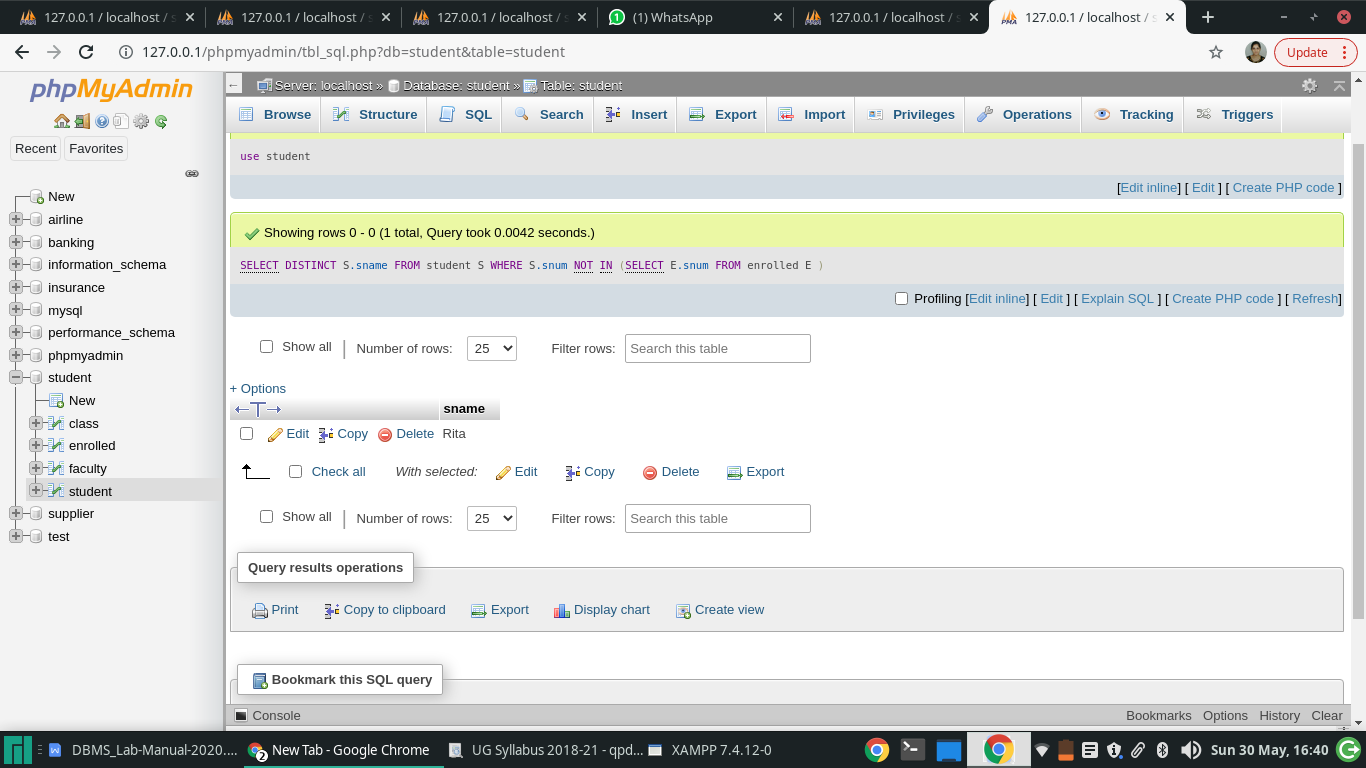
use student;

SELECT DISTINCT S.sname

FROM student S

WHERE S.snum NOT IN (SELECT E.snum

FROM enrolled E );



8)**For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).**

use student;

SELECT S.age, S.lvl

FROM student S

GROUP BY S.age, S.lvl

HAVING S.lvl IN

(SELECT S1.lvl FROM student S1

WHERE S1.age = S.age

GROUP BY S1.lvl, S1.age

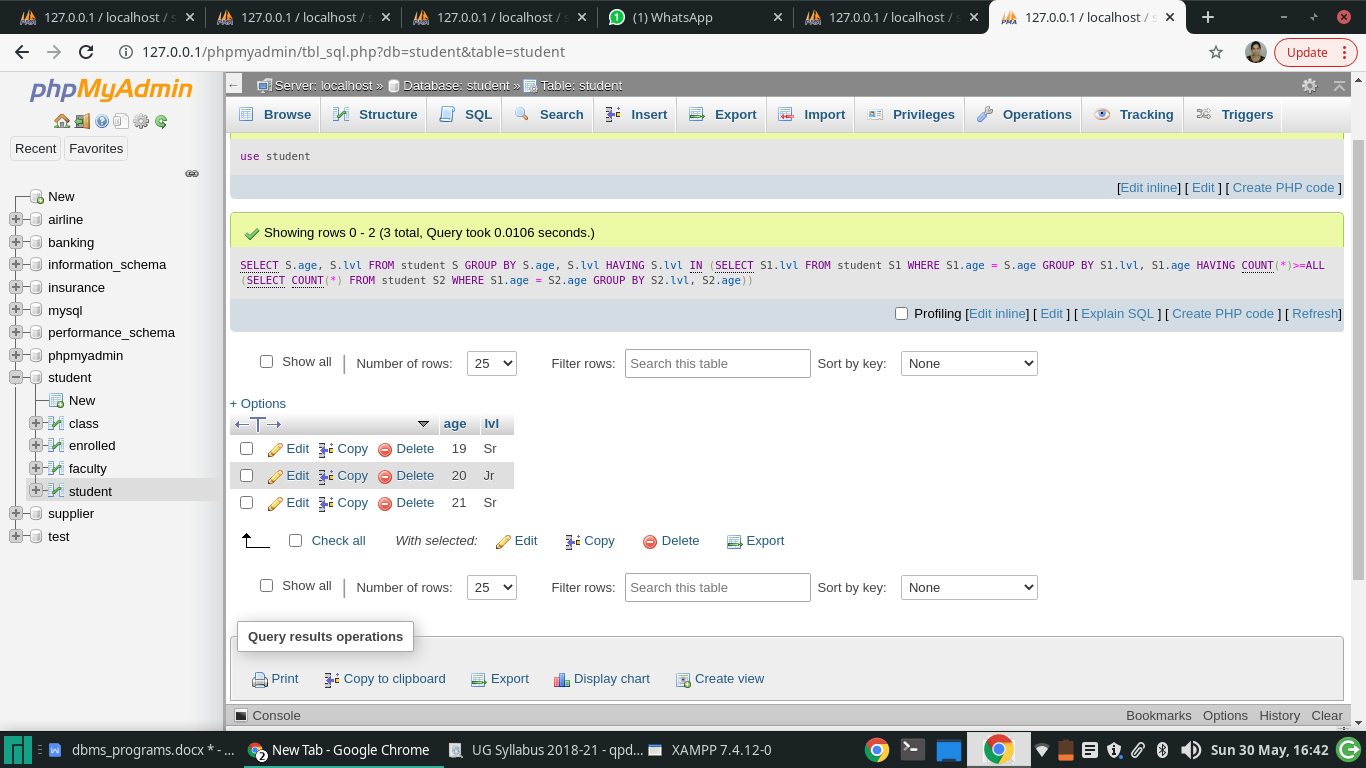
HAVING COUNT(\*)>=ALL

(SELECT COUNT (\*)

FROM student S2

WHERE S1.age = S2.age

GROUP BY S2.lvl, S2.age));



---------------------------program-5---------------------------

**PROGRAM 5: AIRLINE FLIGHT DATABASE**

**Consider the following database that keeps track of airline flight information:**

**FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)**

**AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)**

**CERTIFIED(eid: integer, aid: integer)**

**EMPLOYEES(eid: integer, ename: string, salary: integer)**

**Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.**

**Write each of the following queries in SQL.**

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.
2. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
4. For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
5. Find the names of pilots certified for some Boeing aircraft.
6. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
7. A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.

use airline;

CREATE TABLE FLIGHTS

(FLNO INTEGER PRIMARY KEY,

FFROM VARCHAR(15) NOT NULL,

TTO VARCHAR(15) NOT NULL,

DISTANCE INTEGER,

DEPARTS TIMESTAMP,

ARRIVES TIMESTAMP,

PRICE FLOAT(10,2));

use airline;

CREATE TABLE AIRCRAFT

(AID INTEGER PRIMARY KEY,

ANAME VARCHAR(10),

CRUISINGRANGE INTEGER);

use airline;

CREATE TABLE EMPLOYEES

(EID INTEGER PRIMARY KEY,

ENAME VARCHAR(15),

SALARY FLOAT(10,2));

use airline;

CREATE TABLE CERTIFIED

(EID INTEGER NOT NULL,

AID INTEGER NOT NULL,

PRIMARY KEY (EID, AID),

FOREIGN KEY (EID) REFERENCES EMPLOYEES (EID),

FOREIGN KEY (AID) REFERENCES AIRCRAFT (AID));

USE airline;

insert into AIRCRAFT values(101,'747',3000);

insert into AIRCRAFT values(102,'Boeing',900);

insert into AIRCRAFT values(103,'647',800);

insert into AIRCRAFT values(104,'Dreamliner',10000);

insert into AIRCRAFT values(105,'Boeing',3500);

insert into AIRCRAFT values(106,'707',1500);

insert into AIRCRAFT values(107,'Dream', 120000);

USE airline;

insert into EMPLOYEES values(701,'A',50000);

insert into EMPLOYEES values(702,'B',100000);

insert into EMPLOYEES values(703,'C',150000);

insert into EMPLOYEES values(704,'D',90000);

insert into EMPLOYEES values(705,'E',40000);

insert into EMPLOYEES values(706,'F',60000);

insert into EMPLOYEES values(707,'G',90000);

USE airline;

insert into CERTIFIED values(701,101);

insert into CERTIFIED values(701,102);

insert into CERTIFIED values(701,106);

insert into CERTIFIED values(701,105);

insert into CERTIFIED values(702,104);

insert into CERTIFIED values(703,104);

insert into CERTIFIED values(704,104);

insert into CERTIFIED values(702,107);

insert into CERTIFIED values(703,107);

insert into CERTIFIED values(704,107);

insert into CERTIFIED values(702,101);

insert into CERTIFIED values(703,105);

insert into CERTIFIED values(704,105);

insert into CERTIFIED values(705,103);

USE airline;

insert into FLIGHTS values(101,'Bangalore','Delhi',2500,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 17:15:31',5000);

insert into FLIGHTS values(102,'Bangalore','Lucknow',3000,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 11:15:31',6000);

insert into FLIGHTS values(103,'Lucknow','Delhi',500,TIMESTAMP '2005-05-13 12:15:31',TIMESTAMP ' 2005-05-13 17:15:31',3000);

insert into FLIGHTS values(107,'Bangalore','Frankfurt',8000,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 22:15:31',60000);

insert into FLIGHTS values(104,'Bangalore','Frankfurt',8500,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 23:15:31',75000);

insert into FLIGHTS values(105,'Kolkata','Delhi',3400,TIMESTAMP '2005-05-13 07:15:31',TIMESTAMP '2005-05-13 09:15:31',7000);

1)**Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.**

use airline;

SELECT DISTINCT A.aname

FROM AIRCRAFT A

WHERE A.Aid IN (SELECT C.aid

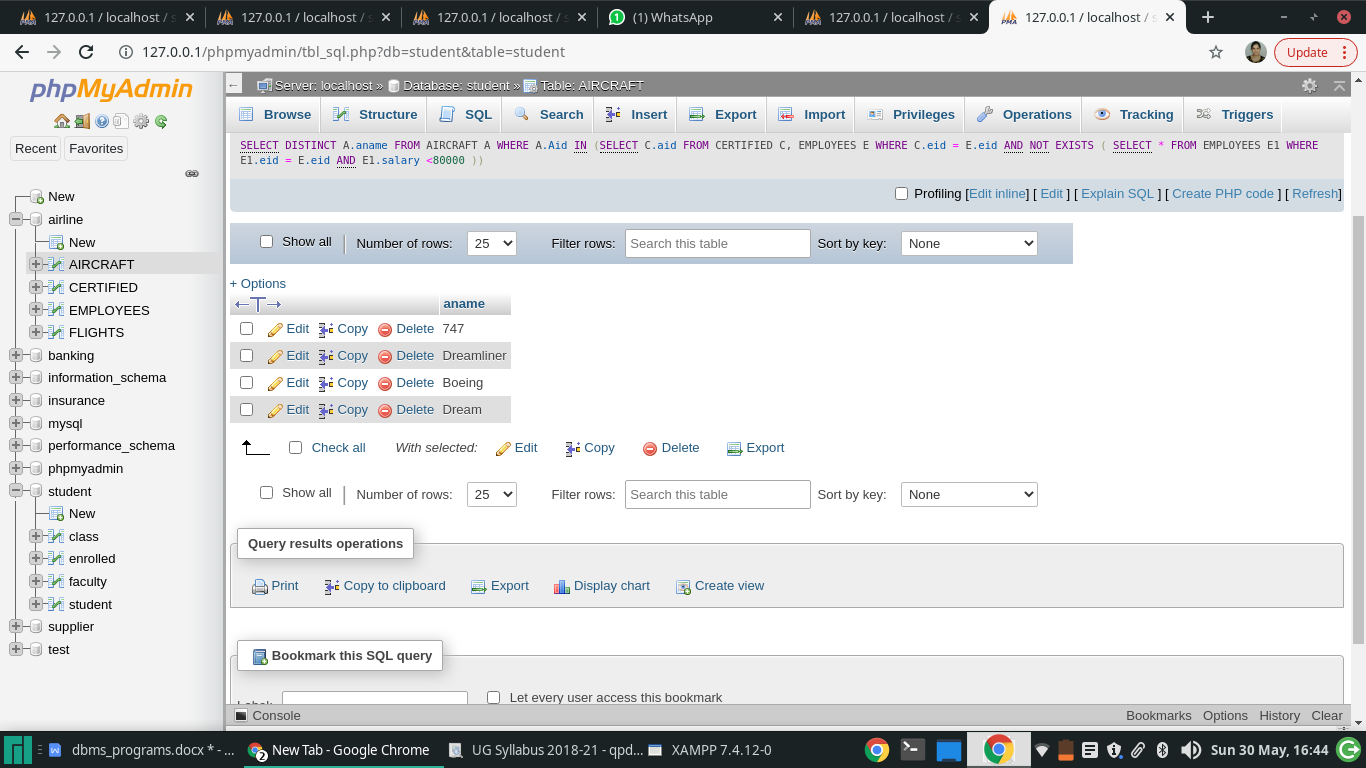
FROM CERTIFIED C, EMPLOYEES E

WHERE C.eid = E.eid AND

NOT EXISTS ( SELECT \*

FROM EMPLOYEES E1

WHERE E1.eid = E.eid AND E1.salary <80000 ));



2)**For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruising range of the aircraft for which she or he is certified.**

use airline;

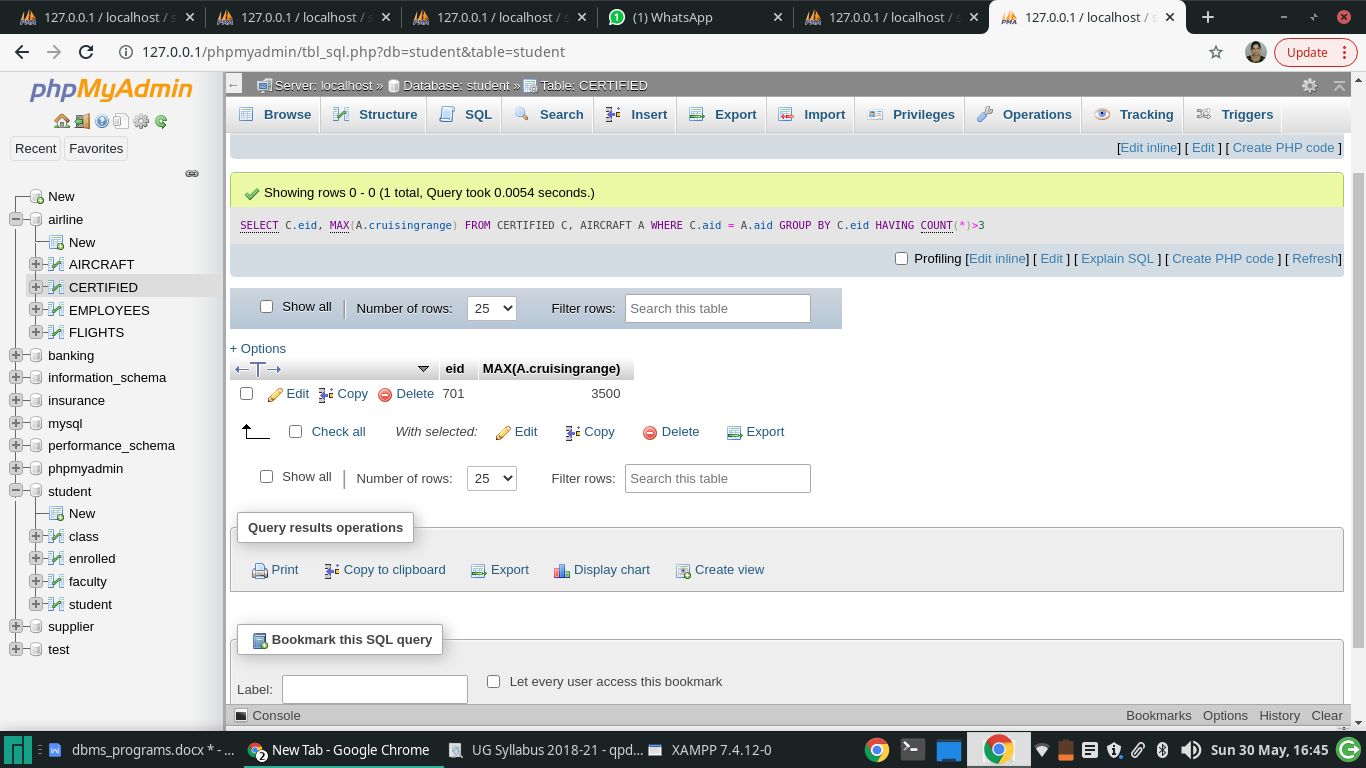
SELECT C.eid, MAX(A.cruisingrange)

FROM CERTIFIED C, AIRCRAFT A

WHERE C.aid = A.aid

GROUP BY C.eid

HAVING COUNT(\*)>3;



3)**Find the names of pilots whose salary is less than the price of the cheapest route from Bangalore to Frankfurt.**

use airline;

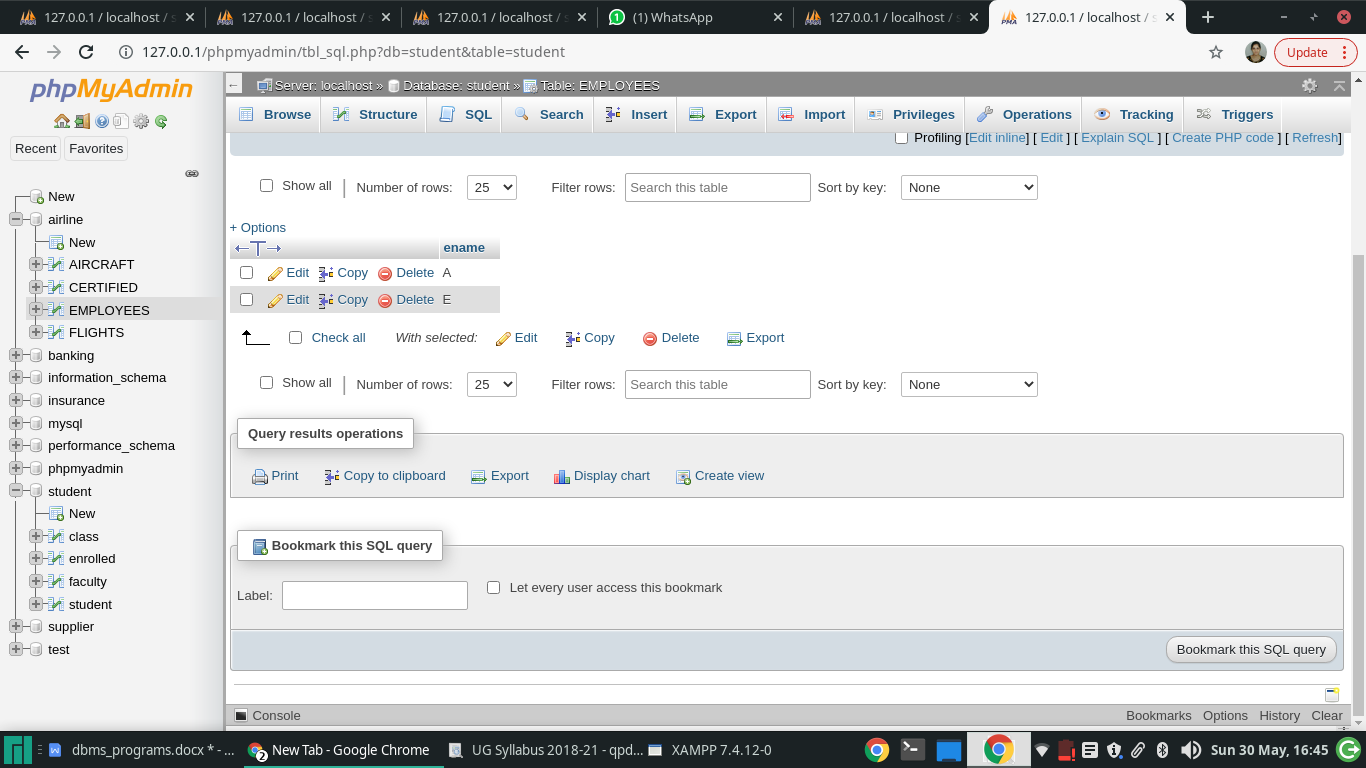
SELECT DISTINCT E.ename

FROM EMPLOYEES E

WHERE E.salary <( SELECT MIN(F.price)

FROM FLIGHTS F

WHERE F.ffrom = 'Bangalore' AND F.tto = 'Frankfurt' );



**4)For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.**

use airline;

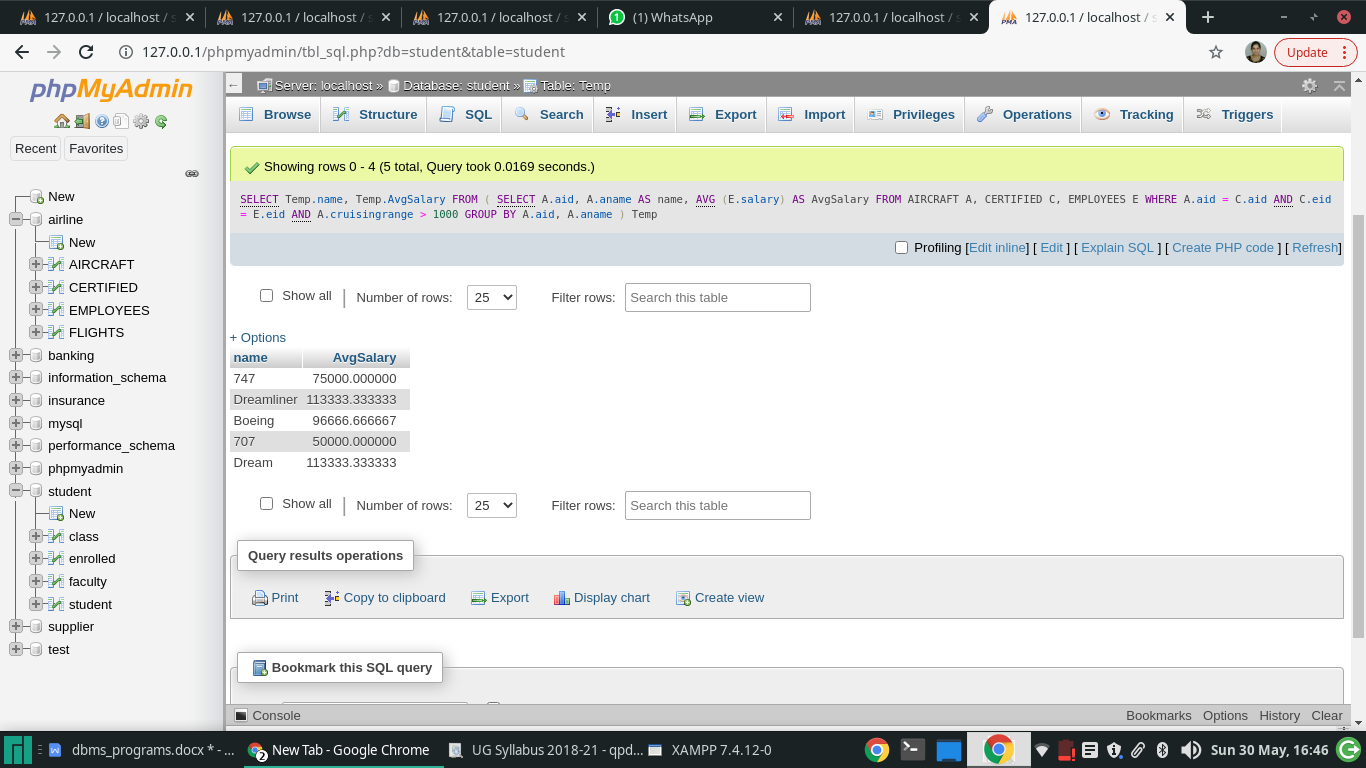
SELECT Temp.name, Temp.AvgSalary

FROM ( SELECT A.aid, A.aname AS name, AVG (E.salary) AS AvgSalary

FROM AIRCRAFT A, CERTIFIED C, EMPLOYEES E

WHERE A.aid = C.aid AND C.eid = E.eid AND A.cruisingrange > 1000

GROUP BY A.aid, A.aname ) Temp;



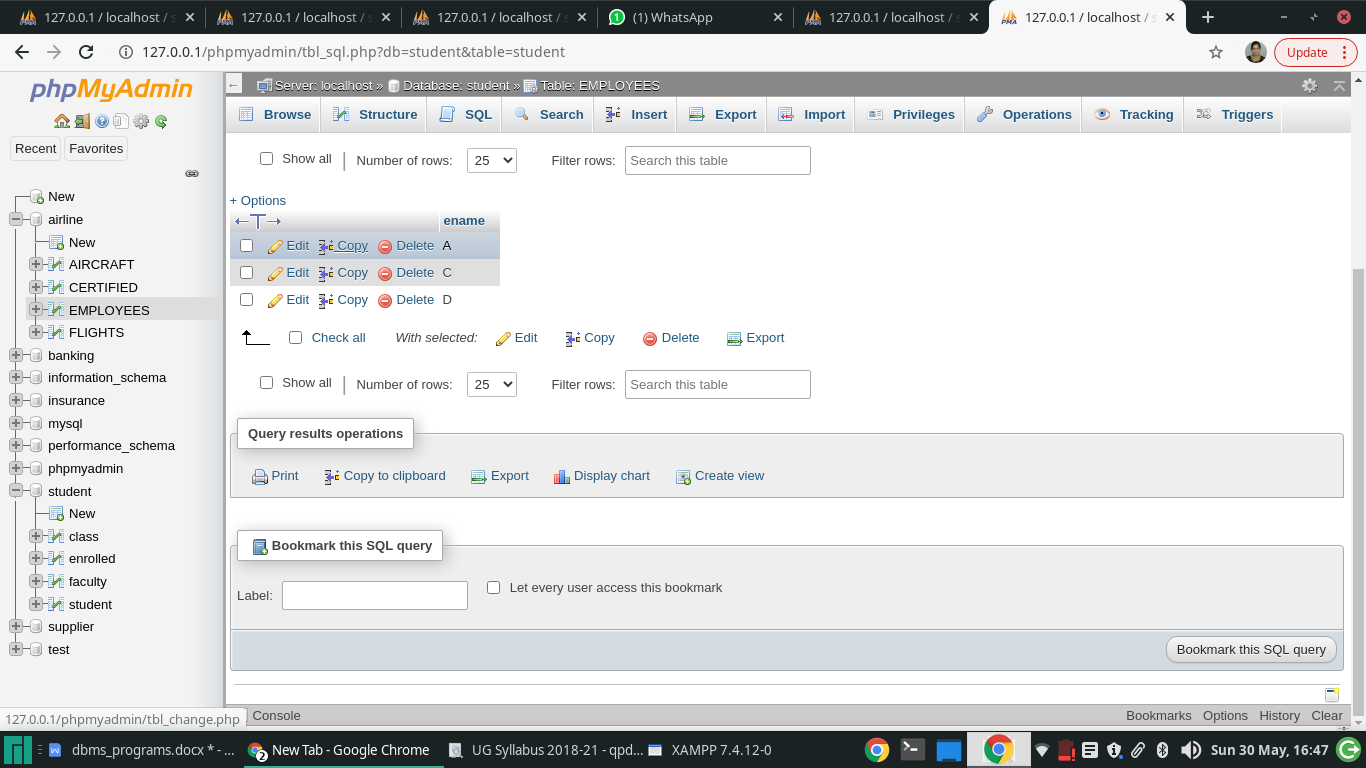
5)**Find the names of pilots certified for some Boeing aircraft.**

use airline;

SELECT DISTINCT E.ename

FROM EMPLOYEES E, CERTIFIED C, AIRCRAFT A

WHERE E.eid = C.eid AND C.aid = A.aid AND A.aname LIKE 'Boeing%';



6)**Find the aids of all aircraft that can be used on routes from Bangalore to Frankfurt.**

use airline;

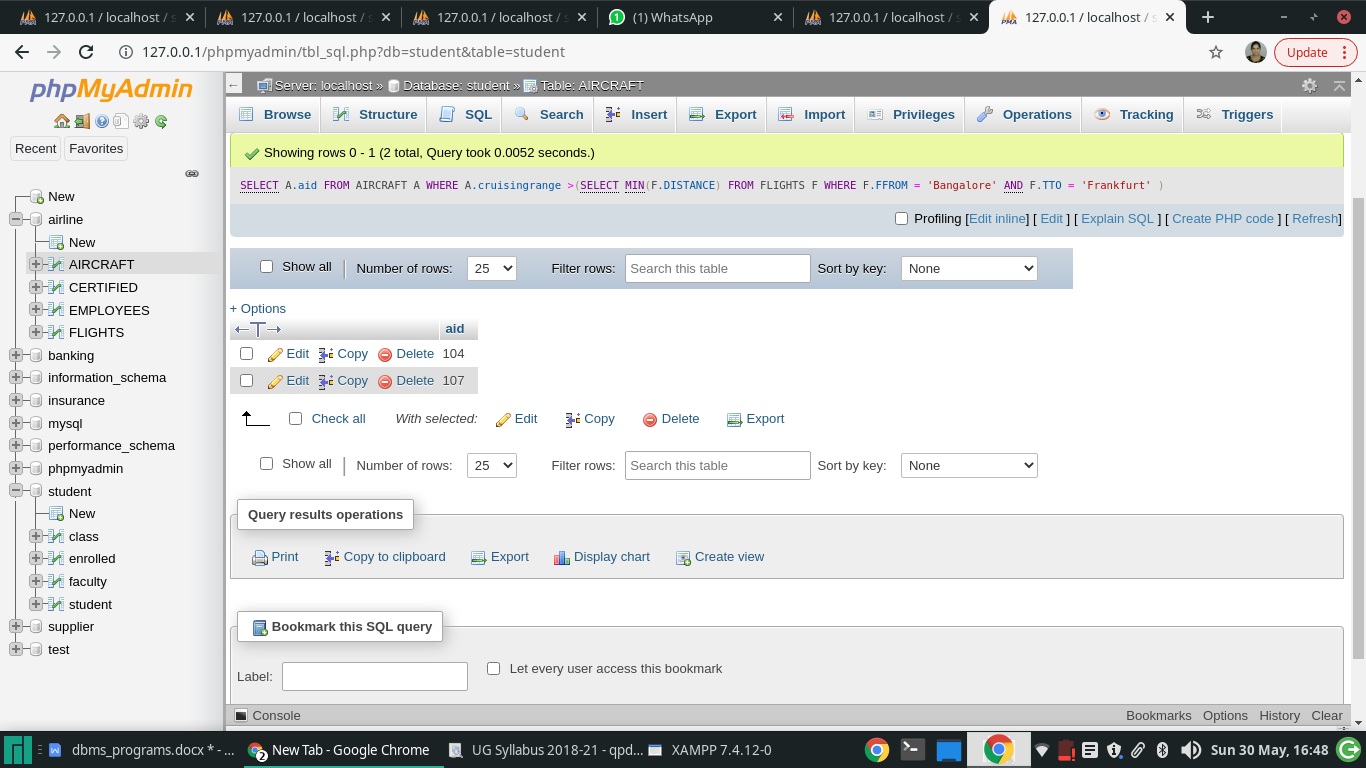
SELECT A.aid

FROM AIRCRAFT A

WHERE A.cruisingrange >(SELECT MIN(F.DISTANCE)

FROM FLIGHTS F

WHERE F.FFROM = 'Bangalore' AND F.TTO = 'Frankfurt' );



7)**A customer wants to travel from Bangalore to Delhi with no more than two changes of flight. List the choice of departure times from Bangalore if the customer wants to arrive in Delhi by 6 p.m.**

use airline;

SELECT F.departs

FROM FLIGHTS F

WHERE F.flno IN ( ( SELECT F0.flno

FROM FLIGHTS F0

WHERE F0.ffrom = 'Bangalore' AND F0.tto = 'Delhi'

AND extract(hour from F0.arrives) < 18 )

UNION

( SELECT F0.flno

FROM FLIGHTS F0, FLIGHTS F1

WHERE F0.ffrom = 'Bangalore' AND F0.tto <> 'Delhi'

AND F0.tto = F1.ffrom AND F1.tto = 'Delhi'

AND F1.departs > F0.arrives

AND extract(hour from F1.arrives) < 18)

UNION

( SELECT F0.flno

FROM FLIGHTS F0, FLIGHTS F1, FLIGHTS F2

WHERE F0.ffrom = 'Bangalore'

AND F0.tto = F1.ffrom

AND F1.tto = F2.ffrom

AND F2.tto = 'Delhi'

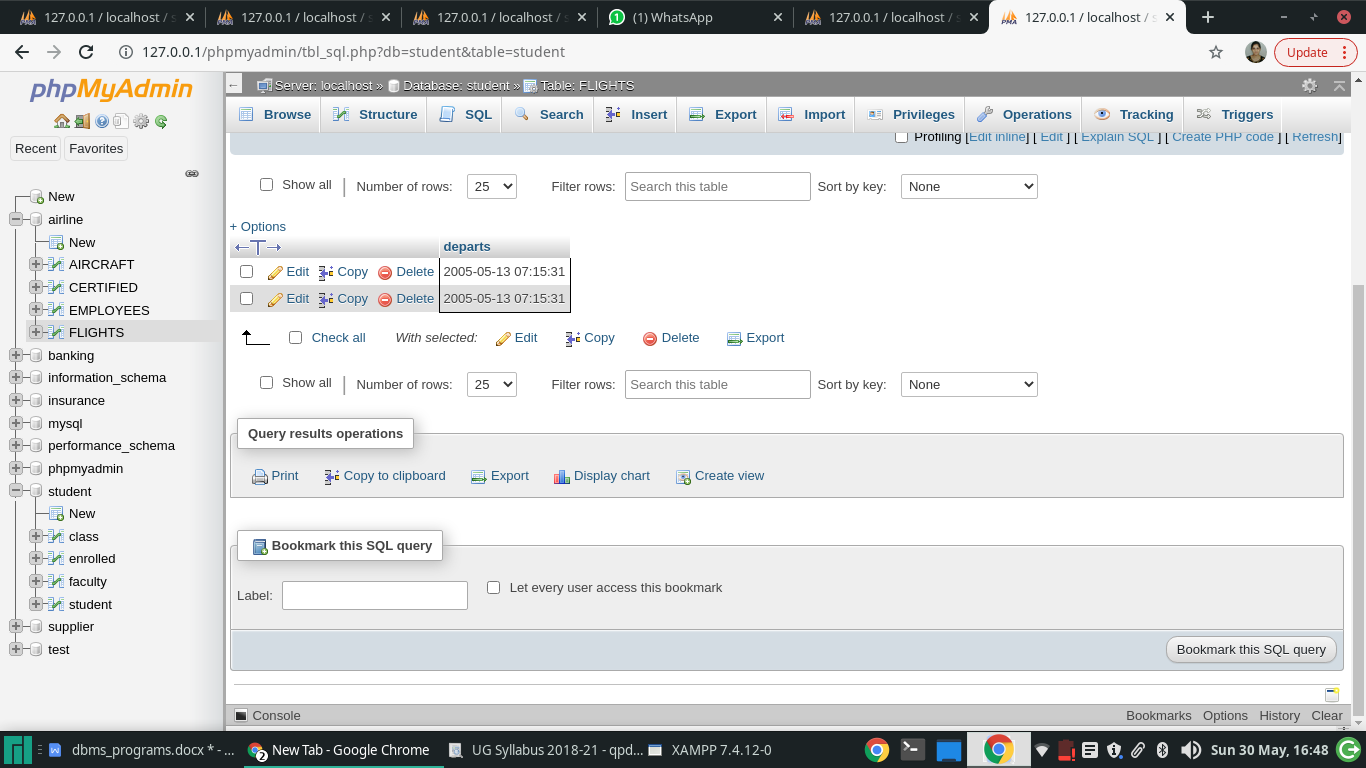
AND F0.tto <> 'Delhi'

AND F1.tto <> 'Delhi'

AND F1.departs > F0.arrives

AND F2.departs > F1.arrives

AND extract(hour from F2.arrives) < 18));



8)**Print the name and salary of every non-pilot whose salary is more than the average salary for pilots.**

use airline;

SELECT E.ename, E.salary

FROM EMPLOYEES E

WHERE E.eid NOT IN ( SELECT DISTINCT C.eid

FROM CERTIFIED C )

AND E.salary >( SELECT AVG(E1.salary)

FROM EMPLOYEES E1

WHERE E1.eid IN

( SELECT DISTINCT C1.eid

FROM CERTIFIED C1 ) );

