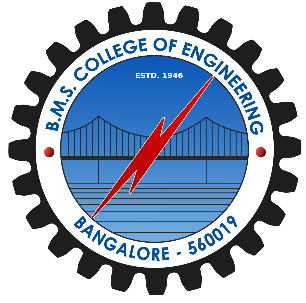
**BMS COLLEGE OF ENGINEERING  
(An autonomous college under vtu, Belagavi)  
bull temple road, Bangalore- 560019**

****

LAB RECORD- DBMS (10 PROGRAMS)

Anitej Prasad

1BM19CS194

4-D

**LAB 1 QUERIES:**

**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 for the car with a specific Regno in the accident with report number 12 to**

**25000.**

**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 were involved.**

create database insurance;

use insurance;

create table person(

driver\_id varchar(10),

name varchar(20),

address varchar(30),

primary key(driver\_id)

);

desc person;

create table car(

reg\_num varchar(10),

model varchar(10),

year int,

primary key(reg\_num)

);

desc car;

create table accident(

report\_num int,

accident\_date date,

location varchar(20),

primary key(report\_num)

);

create table owns(

driver\_id varchar(10),

reg\_num varchar(10),

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

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

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

);

desc owns;

create table 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 participated;

insert into person values('A01','Raghu','Electronic City');

insert into person values('A02','Rishab','Orange County');

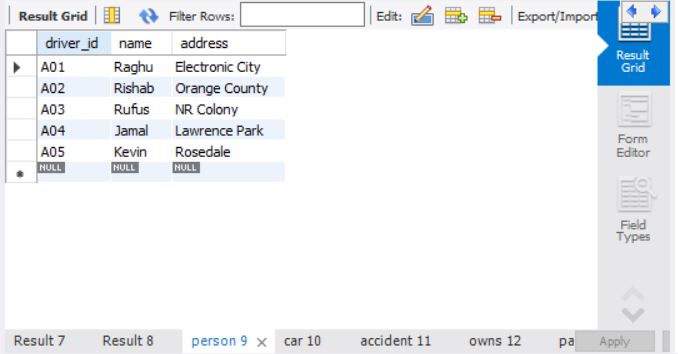
insert into person values('A03','Rufus','NR Colony');

insert into person values('A04','Jamal','Lawrence Park');

insert into person values('A05','Kevin','Rosedale');

commit;

select \* from person;



insert into car values('KA031111','Accord',2005);

insert into car values('KA041122','MX-5',2019);

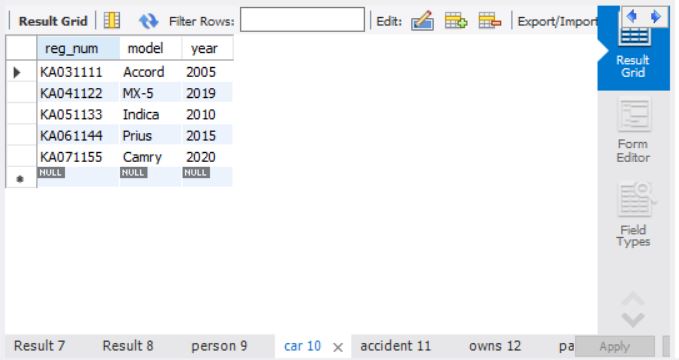
insert into car values('KA051133','Indica',2010);

insert into car values('KA061144','Prius',2015);

insert into car values('KA071155','Camry',2020);

commit;

select \* from car;



insert into accident values(111,'2020-01-01','NR Road');

insert into accident values(122,'2020-02-02','Dalhousie Road');

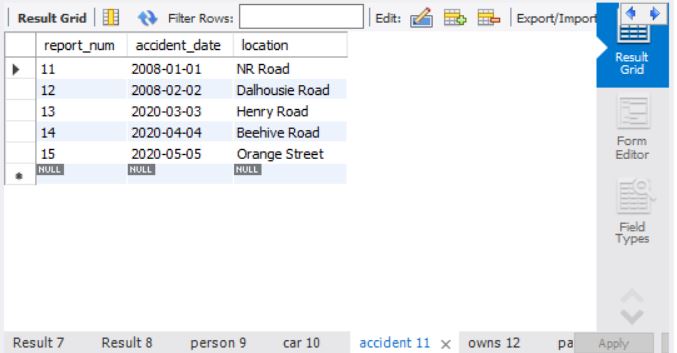
insert into accident values(133,'2020-03-03','Henry Road');

insert into accident values(144,'2020-04-04','Beehive Road');

insert into accident values(155,'2020-05-05','Orange Street');

commit;

select \* from accident;



insert into owns values ('A01','KA031111');

insert into owns values ('A02','KA041122');

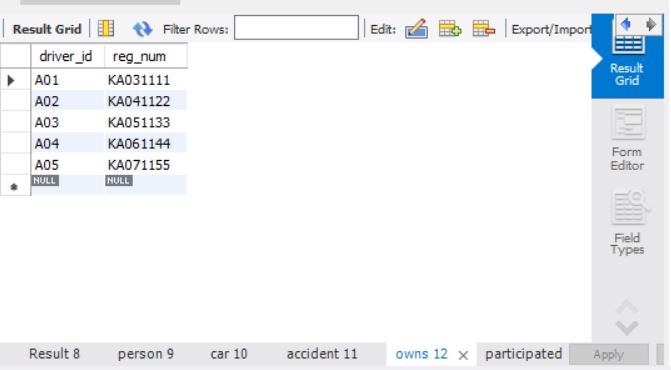
insert into owns values ('A03','KA051133');

insert into owns values ('A04','KA061144');

insert into owns values ('A05','KA071155');

commit;

select \* from owns;



insert into participated values ('A01','KA031111',111, 10000);

insert into participated values ('A02','KA041122',122, 20000);

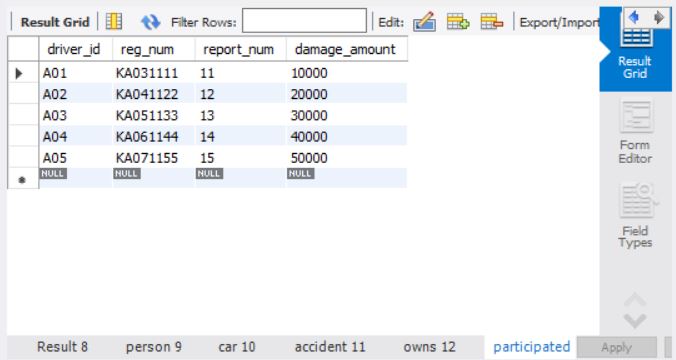
insert into participated values ('A03','KA051133',133, 30000);

insert into participated values ('A04','KA061144',144, 40000);

insert into participated values ('A05','KA071155',155, 50000);

commit;

select \* from participated;

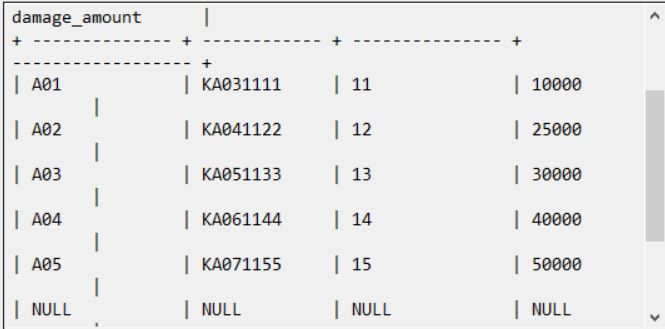


Additional Queries:

update participated

set damage\_amount = 2500

where reg\_num='KA031111';

select \* from participated;

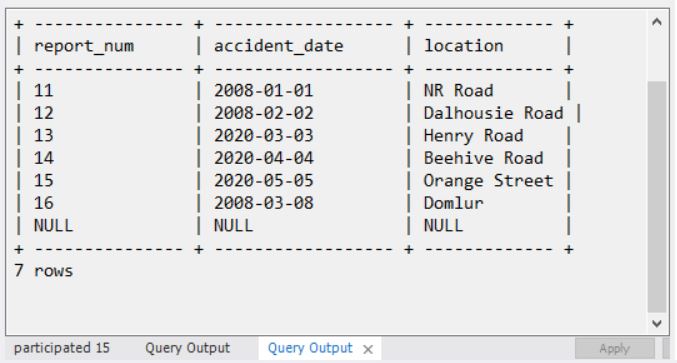
insert into accident values(101,'2008-03-08',Domlur);

insert into participated values('A01','KA031111',101, 1001);

commit;

select \* from accident;

select \* from participated;



insert into car values('KA01010', 'Indica', 2002);

insert into owns values('A02', 'KA01010');

insert into accident values(200, '2008-12-01', 'Pinto Road');

insert into participated values('A02', 'KA01010', 200, 500);

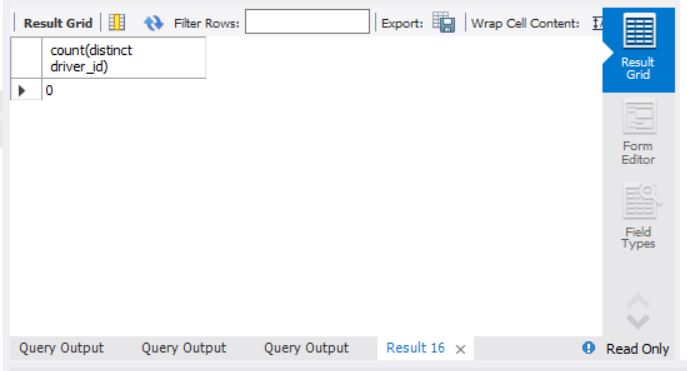
commit;

select \* from car;

select \* from owns;

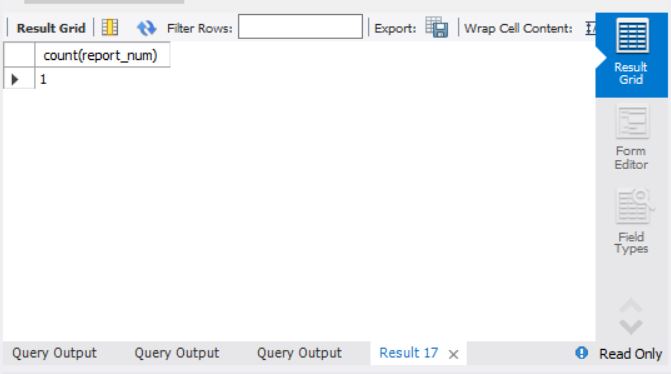
select \* from accident;

select \* from participated;



select count(\*) from accident where year(accident\_date)=2008;

select count(\*) from participated where reg\_num in ( select reg\_num from car where model="Indica");



**LAB 2- QUERIES**

**Consider the following database for a banking enterprise.**

**BRANCH (branch-name: String, branch-city: String, assets: real)**

**ACCOUNTS (accno: int, branch-name: String, balance: real)**

**DEPOSITOR (customer-name: String, customer-street: String, customer-city: String)**

**LOAN (loan-number: int, branch-name: String, amount: real)**

**BORROWER (customer-name: String, loan-number: 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. Find all the customers who have at least two accounts at the Main branch.**

**iv. Find all the customers who have an account at all the branches located in a specific city.**

**v. Demonstrate how you delete all account tuples at every branch located in a specific city.**

create database bank;

use bank;

create table branch (

branch\_name varchar(25),

branch\_city varchar(15),

assets int,

primary key (branch\_name)

);

create table bank\_account (

accno int,

branch\_name varchar(25),

balance int,

primary key (accno),

foreign key (branch\_name) references branch(branch\_name)

);

create table bank\_customer (

customer\_name varchar(10),

customer\_street varchar(25),

customer\_city varchar(15),

primary key (customer\_name)

);

create table depositer (

customer\_name varchar(10),

accno int,

primary key(customer\_name, accno),

foreign key (customer\_name) references bank\_customer(customer\_name),

foreign key (accno) references bank\_account(accno)

);

create table loan (

loan\_number int,

branch\_name varchar(25),

amount int,

primary key (loan\_number),

foreign key (branch\_name) references branch(branch\_name)

);

insert into branch values('SBI\_Chamrajpet', 'Bangalore', 50000);

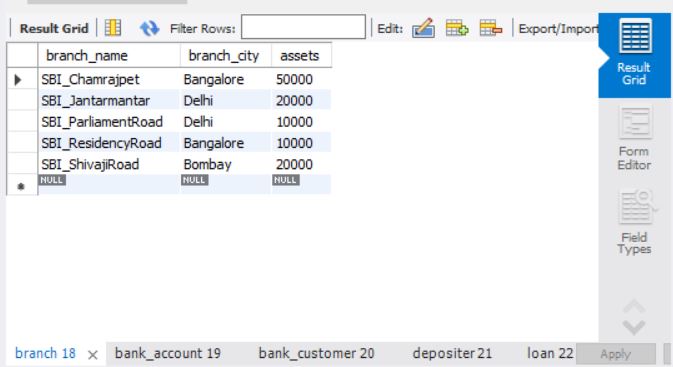
insert into branch values('SBI\_ResidencyRoad', 'Bangalore', 10000);

insert into branch values('SBI\_ShivajiRoad', 'Bombay', 20000);

insert into branch values('SBI\_ParliamentRoad', 'Delhi', 10000);

insert into branch values('SBI\_Jantarmantar', 'Delhi', 20000);

commit;



insert into bank\_account values(1, 'SBI\_Chamrajpet', 2000);

insert into bank\_account values(2, 'SBI\_ResidencyRoad', 5000);

insert into bank\_account values(3, 'SBI\_ShivajiRoad', 6000);

insert into bank\_account values(4, 'SBI\_ParliamentRoad', 9000);

insert into bank\_account values(5, 'SBI\_Jantarmantar', 8000);

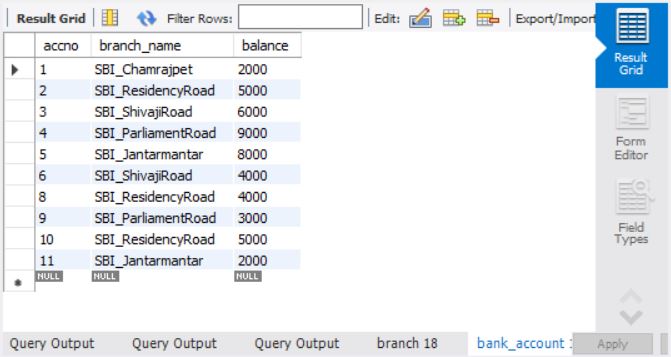
insert into bank\_account values(6, 'SBI\_ShivajiRoad', 4000);

insert into bank\_account values(8, 'SBI\_ResidencyRoad', 4000);

insert into bank\_account values(9, 'SBI\_ParliamentRoad', 3000);

insert into bank\_account values(10, 'SBI\_ResidencyRoad', 5000);

insert into bank\_account values(11, 'SBI\_Jantarmantar', 2000);



commit;

insert into bank\_customer values ('Abhay', 'Bull\_Temple\_Road', 'Bangalore');

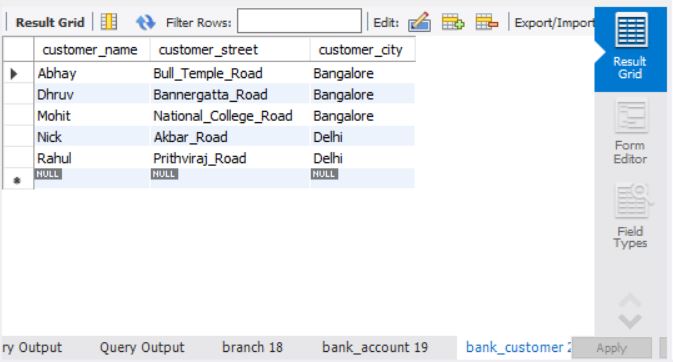
insert into bank\_customer values ('Dhruv', 'Bannergatta\_Road', 'Bangalore');

insert into bank\_customer values ('Mohit', 'National\_College\_Road', 'Bangalore');

insert into bank\_customer values ('Nick', 'Akbar\_Road', 'Delhi');

insert into bank\_customer values ('Rahul', 'Prithviraj\_Road', 'Delhi');

commit;



insert into depositer values('Abhay', 1);

insert into depositer values('Dhruv', 2);

insert into depositer values('Nick', 4);

insert into depositer values('Rahul', 5);

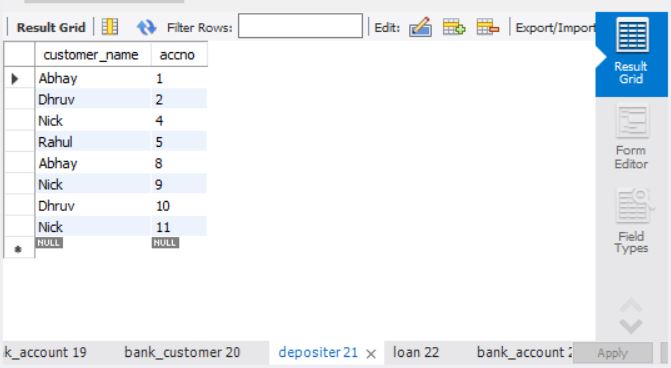
insert into depositer values('Abhay', 8);

insert into depositer values('Nick', 9);

insert into depositer values('Dhruv', 10);

insert into depositer values('Nick', 11);

commit;



insert into loan values(1, 'SBI\_Chamrajpet', 1000);

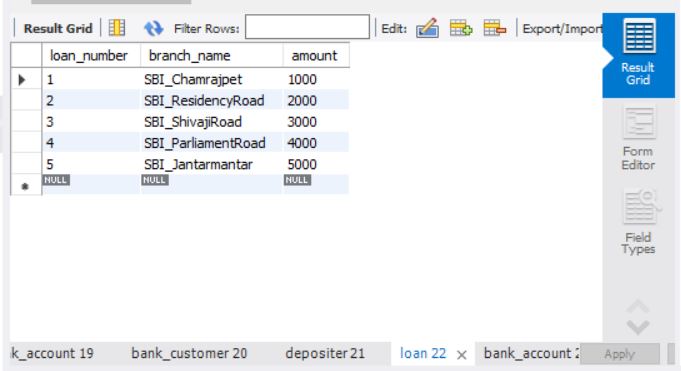
insert into loan values(2, 'SBI\_ResidencyRoad', 2000);

insert into loan values(3, 'SBI\_ShivajiRoad', 3000);

insert into loan values(4, 'SBI\_ParliamentRoad', 4000);

insert into loan values(5, 'SBI\_Jantarmantar', 5000);

commit;



select \* from branch;

select \* from bank\_account;

select \* from bank\_customer;

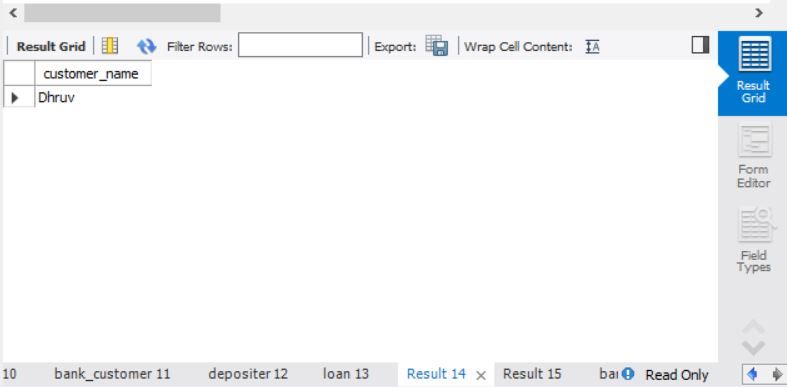
select \* from depositer;

select \* from loan;

Additional Queries:

select distinct c.customer\_name from bank\_customer c,bank\_account b where exists(select d.customer\_name,count(d.customer\_name) from depositer d,bank\_account ba where ba.accno = d.accno and

c.customer\_name = d.customer\_name and ba.branch\_name = 'SBI\_ResidencyRoad' group by d.customer\_name having count(d.customer\_name)>=2);



select d.customer\_name from depositer d,branch b,bank\_account a

where b.branch\_name=a.branch\_name

AND a.accno=d.accno

and branch\_city='Delhi'

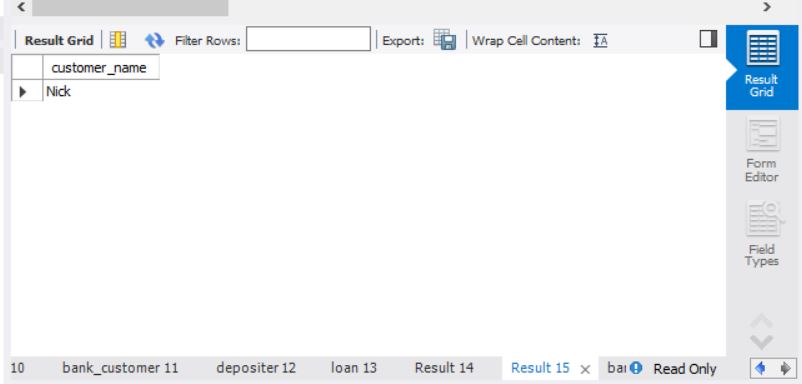
group by d.customer\_name

HAVING COUNT(distinct b.branch\_name)=(

SELECT COUNT(branch\_name)

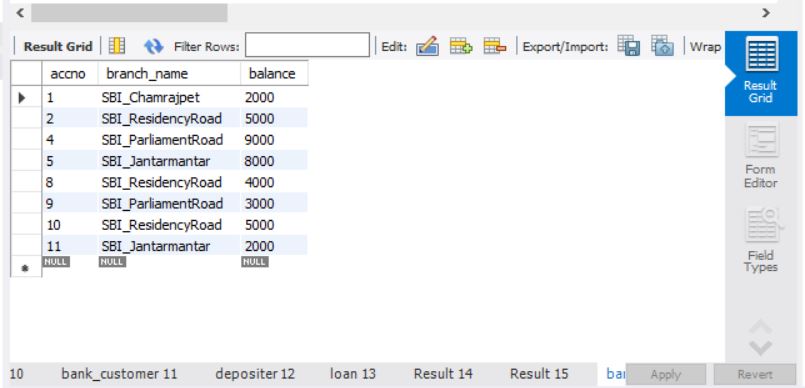
FROM branch

WHERE branch\_city='Delhi');



delete from bank\_account where branch\_name in (select branch\_name from branch where branch\_city = 'Bombay');

select \* from bank\_account;



**LAB 3 QUERIES:**

**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:**

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

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

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

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

**v. 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).**

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

**vii. Find the sids of suppliers who supply only red parts.**

create database supplier;

use supplier;

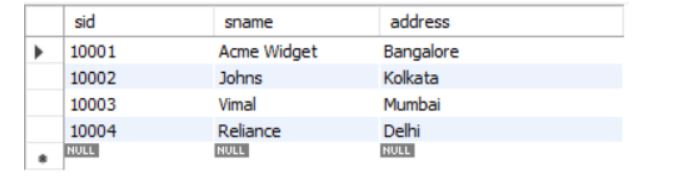
create table SUPPLIERS(sid integer,sname varchar(20),address varchar(40),primary key(sid));

INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10001', 'Acme Widget', 'Bangalore');

INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10002', 'Johns', 'Kolkata');

INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10003', 'Vimal', 'Mumbai');

INSERT INTO `supplier`.`suppliers` (`sid`, `sname`, `address`) VALUES ('10004', 'Reliance', 'Delhi');



commit;

select\* from SUPPLIERS;

create table PARTS(pid integer,pname varchar(20),color varchar(30),primary key(pid));

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20001', 'Book', 'Red');

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20002', 'Pen', 'Red');

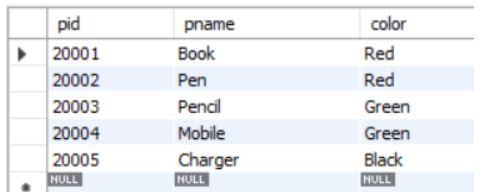
INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20003', 'Pencil', 'Green');

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20004', 'Mobile', 'Green');

INSERT INTO `supplier`.`parts` (`pid`, `pname`, `color`) VALUES ('20005', 'Charger', 'Black');

commit;

select\* from PART;



create table CATALOG(sid integer,pid integer,foreign key(sid) references SUPPLIERS(sid),foreign key(pid) references PARTS(pid),

cost integer,primary key(sid,pid));

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20001', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20002', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20003', '30');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20004', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10001', '20005', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10002', '20001', '10');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10002', '20002', '20');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10003', '20003', '30');

INSERT INTO `supplier`.`catalog` (`sid`, `pid`, `cost`) VALUES ('10004', '20003', '40');

commit;

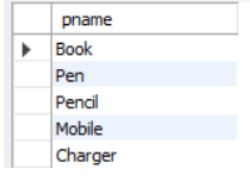
select\* from CATALOG;

Additional Queries:

SELECT DISTINCT P.pname

FROM Parts P, Catalog C

WHERE P.pid = C.pid;



select S.sname from SUPPLIERS S where not exists

(select P.pid from PARTS P where not exists

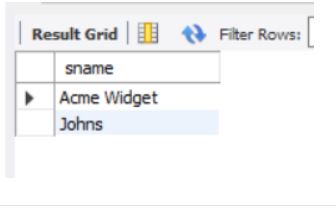
(select C.sid from CATALOG C where C.sid = S.sid and C.pid = P.pid));



select S.sname from SUPPLIERS S where not exists

(select P.pid from PARTS P where P.color = 'Red' and

(not exists (select C.sid from CATALOG C where C.sid = S.sid and C.pid = P.pid)));



select P.pname from PARTS P, CATALOG C, SUPPLIERS S

where P.pid = C.pid and C.sid = S.sid and S.sname = 'Acme Widget'

and not exists (select \* from CATALOG C1, SUPPLIERS S1

where P.pid = C1.pid and C1.sid = S1.sid and S1.sname <> 'Acme Widget');

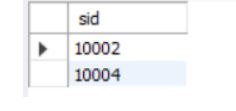


SELECT DISTINCT C.sid FROM Catalog C

WHERE C.cost > ( SELECT AVG (C1.cost)

FROM Catalog C1

WHERE C1.pid = C.pid );



SELECT P.pid, S.sname

FROM Parts P, Suppliers S, Catalog C

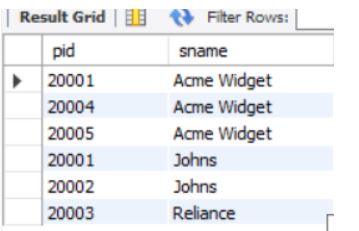
WHERE C.pid = P.pid

AND C.sid = S.sid

AND C.cost = (SELECT MAX(C1.cost)

FROM Catalog C1

WHERE C1.pid = P.pid);



**LAB 4 QUERIES:**

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

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

**CLASS (name: 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 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.**

**i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by**

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

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

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

**v. Find the names of faculty members for whom the combined enrolment of the courses that they teach is less**

**than five.**

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

**vii. 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\_faculty;

USE student\_faculty;

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));

CREATE TABLE enrolled(

snum INT,

cname VARCHAR(20),

PRIMARY KEY(snum,cname),

FOREIGN KEY(snum) REFERENCES student(snum),

FOREIGN KEY(cname) REFERENCES class(cname));

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

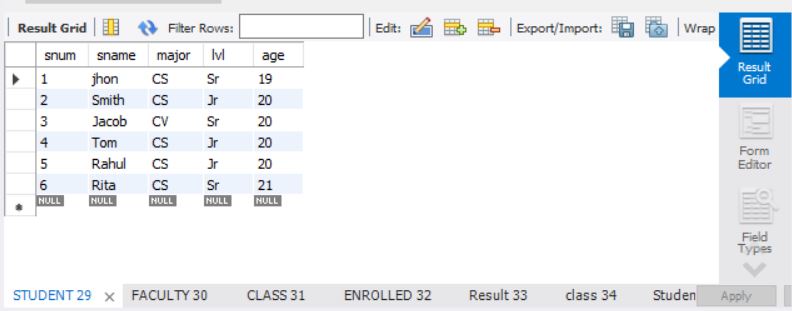
INSERT INTO STUDENT VALUES(2, 'Smith', 'CS', 'Jr', 20);

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);



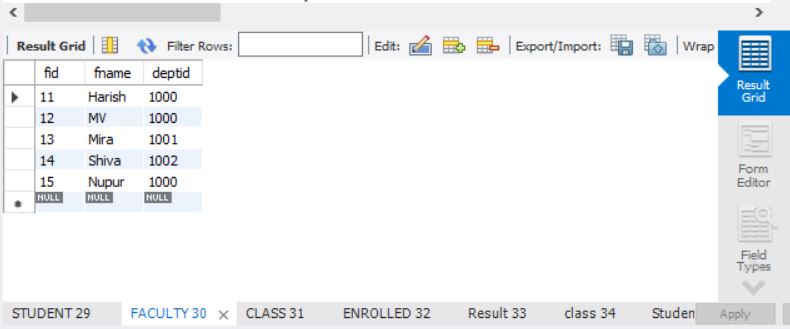
INSERT INTO FACULTY VALUES(11, 'Harish', 1000);

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);



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);

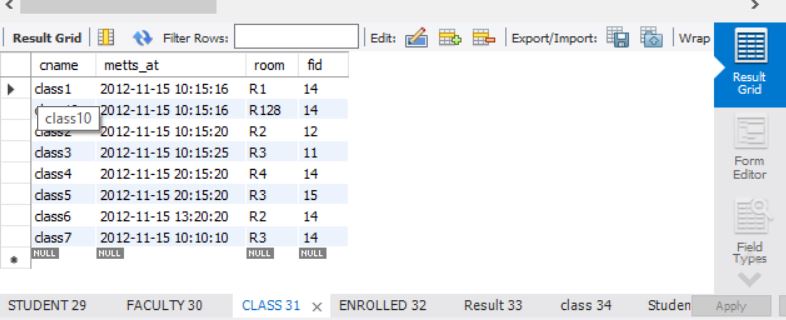
insert into class values('class3', '12/11/15 10:15:25', 'R3', 11);

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);



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');

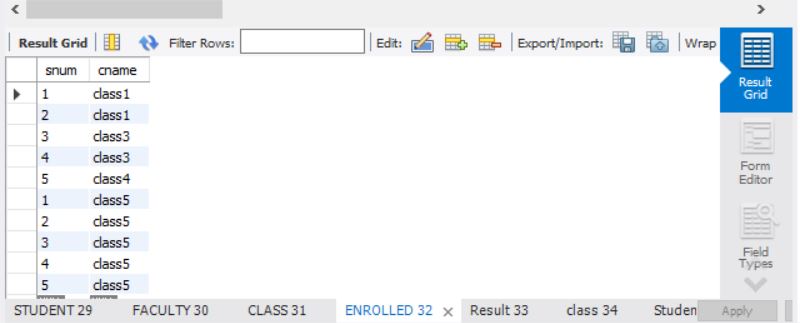
insert into enrolled values(1, 'class5');

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

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

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

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



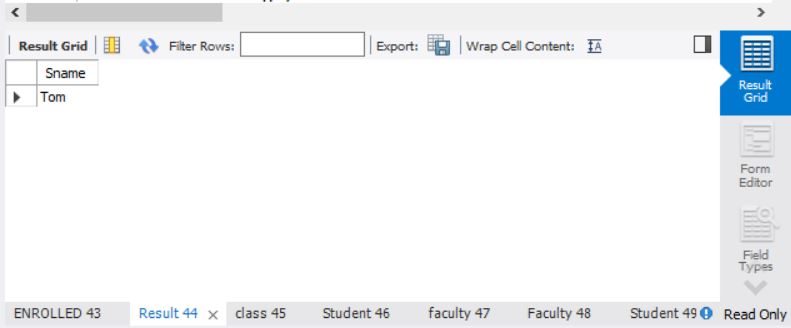
Additional Queries:

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 = 'Harish' AND S.lvl = 'Jr';



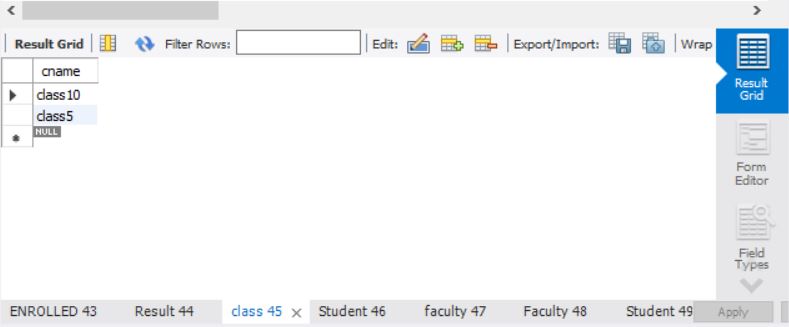
SELECT DISTINCT cname

FROM class

WHERE room='R128'

OR

cname IN (SELECT e.cname FROM enrolled e GROUP BY e.cname HAVING COUNT(\*)>=5);



SELECT DISTINCT S.sname

FROM Student S

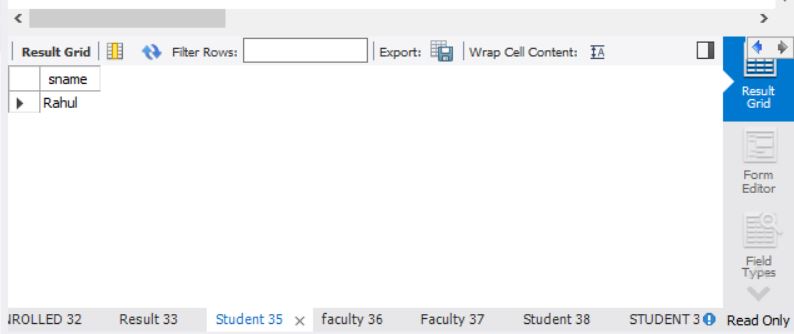
WHERE S.snum IN (SELECT E1.snum

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);

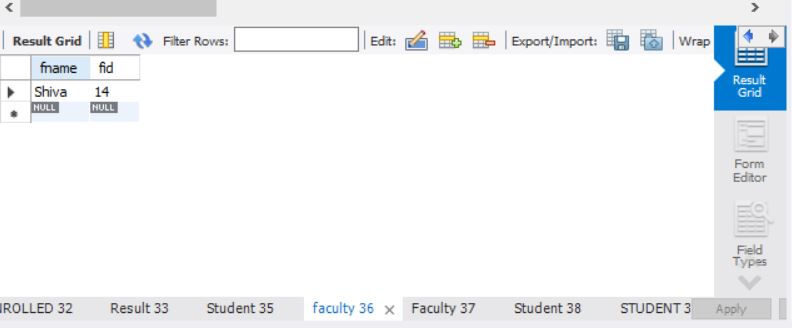


SELECT f.fname,f.fid

FROM faculty f

WHERE f.fid in ( SELECT fid FROM class

GROUP BY fid HAVING COUNT(\*)=(SELECT COUNT(DISTINCT room) FROM class) );



SELECT DISTINCT F.fname

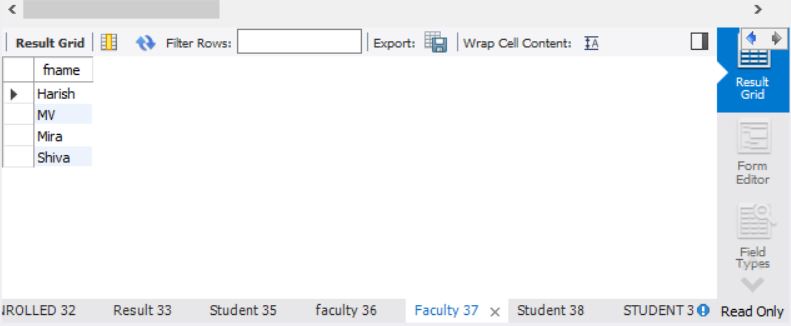
FROM Faculty F

WHERE 5 > (SELECT COUNT(E.snum)

FROM Class C, Enrolled E

WHERE C.cname = E.cname

AND C.fid = F.fid);



SELECT DISTINCT S.sname

FROM Student S

WHERE S.snum NOT IN (SELECT E.snum

FROM Enrolled E );



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.age, S1.lvl

HAVING COUNT(\*) >= ALL (SELECT COUNT(\*)

FROM STUDENT S2

WHERE S1.age=S2.age

GROUP BY S2.lvl, S2.age))

ORDER BY S.age;



**LAB 5 QUERIES:**

**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)**

**EMPLOYEE (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.**

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

**ii. 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.**

**iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to**

**Frankfurt.**

**iv. 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.**

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

**vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.**

**vii. 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.**

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

create database flightdb;

use flightdb;

create table flights(

flno int,

fromplace varchar(15),

toplace varchar(15),

distance int,

departs datetime,

arrives datetime,

price int,

primary key (flno)

);

desc flights;

create table aircraft(

aid int,

aname varchar(15),

cruisingrange int,

primary key (aid)

);

desc aircraft;

create table employees (

eid int,

ename varchar(15),

salary int,

primary key (eid)

);

desc employees;

create table certified (

eid int,

aid int,

foreign key (eid) references employees(eid),

foreign key (aid) references aircraft(aid)

);

desc certified;

insert into flights values(101, 'Bangalore', 'Delhi', 2500, '2005-05-13 07:15:31', '2005-05-13 18:15:31', 5000);

insert into flights values(102, 'Bangalore', 'Lucknow', 3000, '2013-05-05 07:15:31', '2013-05-05 11:15:31', 6000);

insert into flights values(103, 'Lucknow', 'Delhi', 500, '2013-05-05 12:15:31', '2013-05-05 17:15:31', 3000);

insert into flights values(107, 'Bangalore', 'Frankfurt', 8000, '2013-05-05 07:15:31', '2013-05-05 22:15:31', 60000);

insert into flights values(104, 'Bangalore', 'Frankfurt', 8500, '2013-05-05 07:15:31', '2013-05-05 23:15:31', 75000);

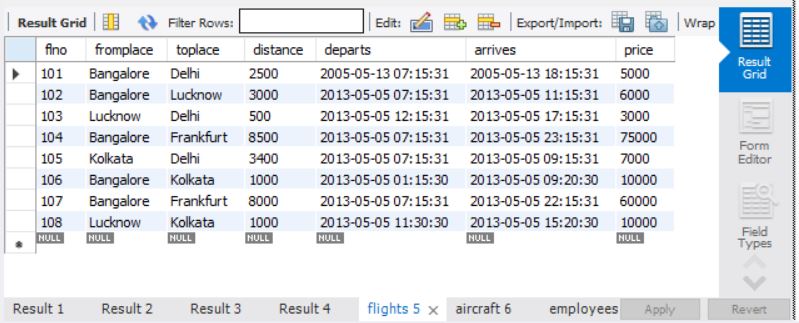
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31', 7000);

insert into flights values(106, 'Bangalore', 'Kolkata', 1000, '2013-05-05 01:15:30', '2013-05-05 09:20:30', 10000);

insert into flights values(108, 'Lucknow', 'Kolkata', 1000, '2013-05-05 11:30:30', '2013-05-05 15:20:30', 10000);

commit;

select \* from flights;



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);

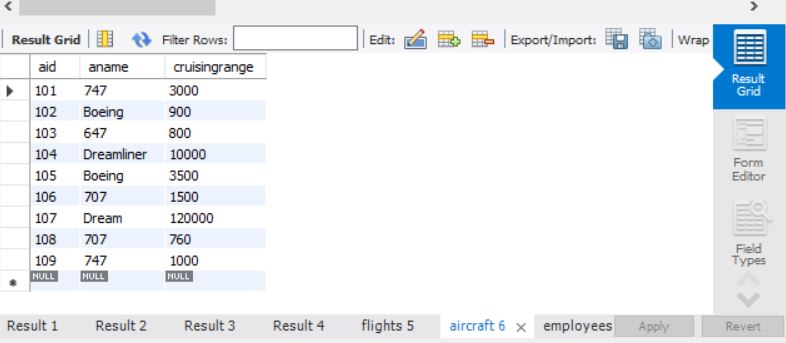
insert into aircraft values(107, 'Dream', 120000);

insert into aircraft values(108, '707', 760);

insert into aircraft values(109, '747', 1000);

commit;

select \* from aircraft;



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);

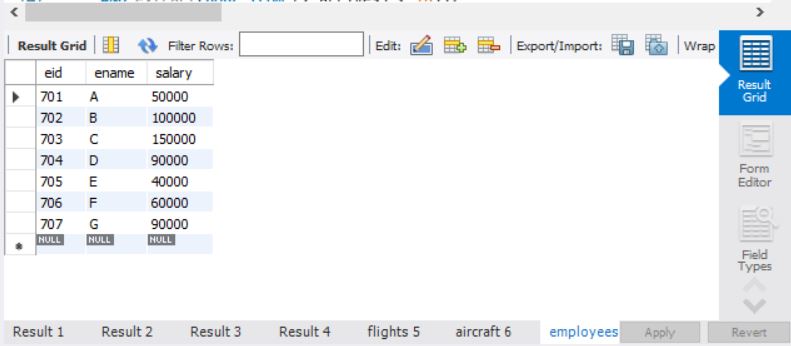
insert into employees values(705, 'E', 40000);

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

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

commit;

select \* from employees;



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);

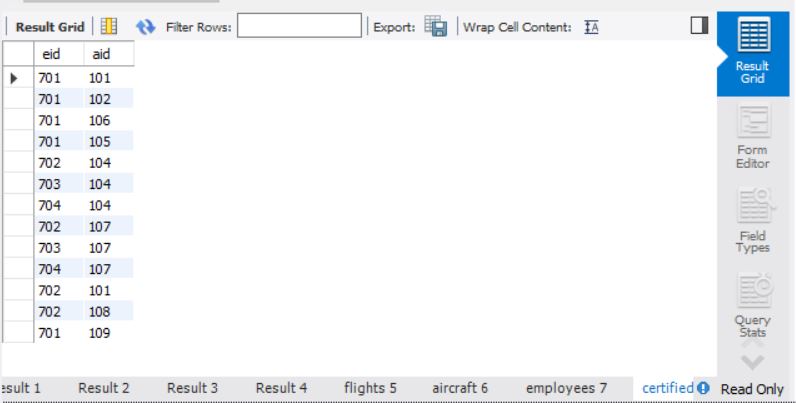
insert into certified values(702, 101);

insert into certified values(702, 108);

insert into certified values(701, 109);

commit;

select \* from certified;



Additional Queries:

select distinct a.aname from aircraft a where a.aid in (

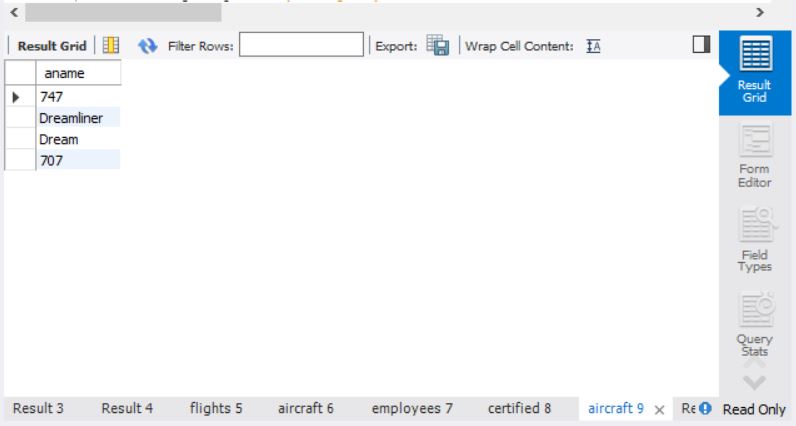
select c.aid from certified c, employees e where

c.eid = e.eid and not exists(

select \* from employees e1 where e1.eid=e.eid and e1.salary<80000

)

);

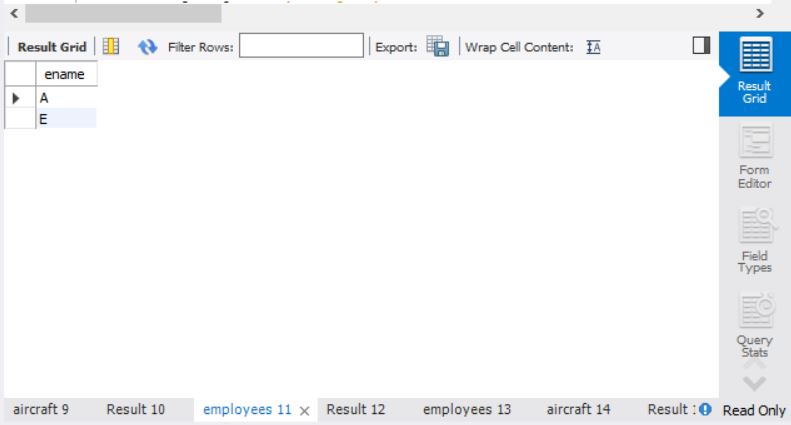
n

select max(a.cruisingrange), c.eid from certified c, aircraft a where c.aid = a.aid group by c.eid having count(c.eid)>3;



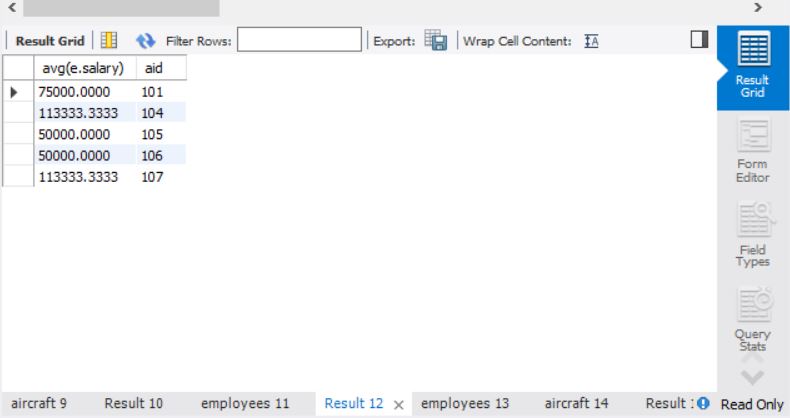
select ename from employees where salary <(

select min(price) from flights where fromplace='Bangalore' and toplace='Frankfurt');



select avg(e.salary), c.aid from certified c, employees e where c.aid in(

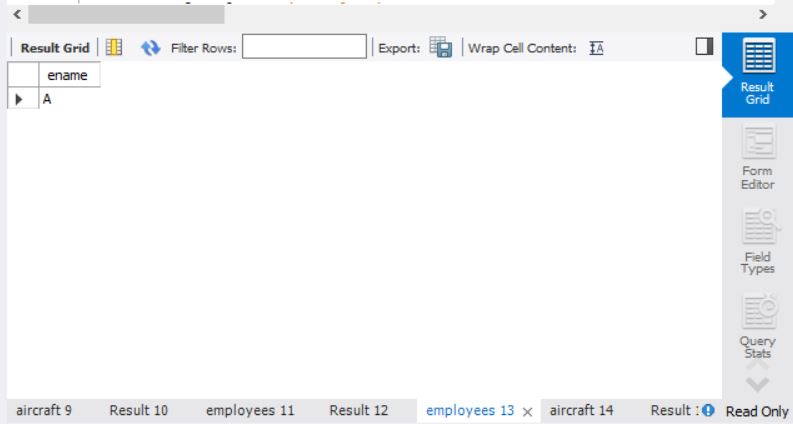
select aid from aircraft where cruisingrange>1000) and e.eid = c.eid group by c.aid;



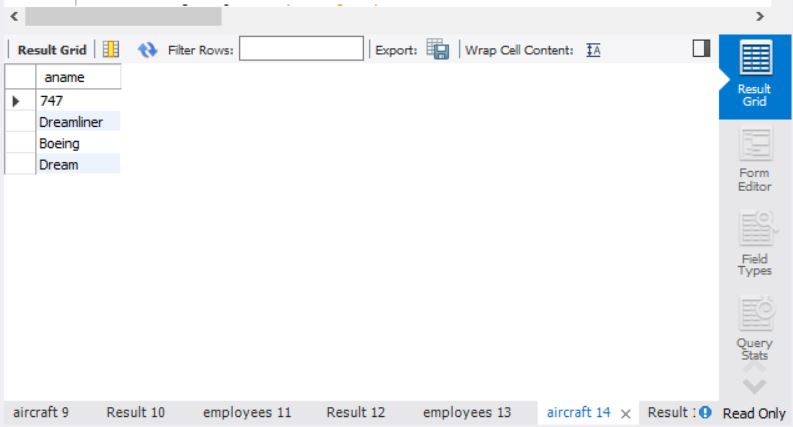
select ename from employees where eid in(

select eid from certified where aid in(

select aid from aircraft where aname = 'Boeing'));



select aname from aircraft where cruisingrange > any (select distance from flights where fromplace='Bangalore' and toplace='Delhi');



SELECT F.flno, F.departs

FROM flights F

WHERE F.flno IN ( ( SELECT F0.flno

FROM flights F0

WHERE F0.fromplace = 'Bangalore' AND F0.toplace = 'Kolkata'

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

UNION

( SELECT F0.flno

FROM flights F0, flights F1

WHERE F0.fromplace = 'Bangalore' AND F0.toplace <> 'Kolkata'

AND F0.toplace = F1.fromplace AND F1.toplace = 'Kolkata'

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.fromplace = 'Bangalore'

AND F0.toplace = F1.fromplace

AND F1.toplace = F2.fromplace

AND F2.toplace = 'Kolkata'

AND F0.toplace <> 'Kolkata'

AND F1.toplace <> 'Kolkata'

AND F1.departs > F0.arrives

AND F2.departs > F1.arrives

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



**LAB 6 QUERIES**

**Consider the following relations for an Order Processing database application in a company.**

**CUSTOMER (CUST #: int, cname: String, city: String)**

**ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)**

**ITEM (item #: int, unit-price: int)**

**ORDER-ITEM (order #: int, item #: int, qty: int)**

**WAREHOUSE (warehouse #: int, city: String)**

**SHIPMENT (order #: int, warehouse #: int, ship-date: date)**

create database orderdb;

use orderdb;

create table salesman (

salesman\_id int(4),

name varchar (20),

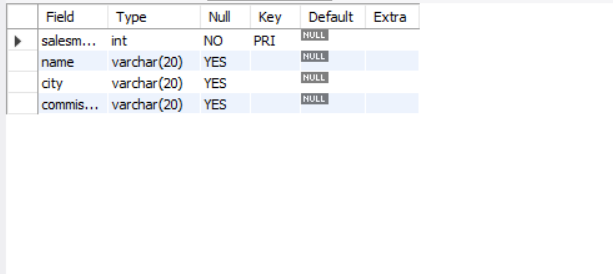
city varchar (20),

commission varchar (20),

primary key (salesman\_id)

);

desc salesman;



create table customer (

customer\_id int(4),

cust\_name varchar (20),

city varchar (20),

grade int (3),

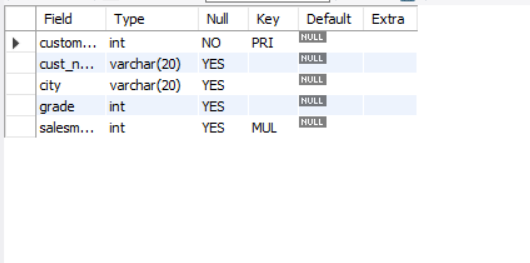
salesman\_id int(4),

primary key (customer\_id),

foreign key (salesman\_id) references salesman(salesman\_id) on delete set null

);

desc customer;



create table orders (

ord\_no int (5),

purchase\_amt int (10),

ord\_date date,

customer\_id int(4),

salesman\_id int(4),

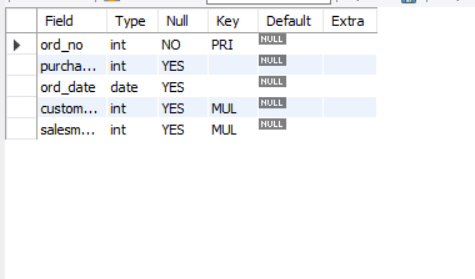
primary key (ord\_no),

foreign key (customer\_id) references customer1 (customer\_id) on delete cascade,

foreign key (salesman\_id) references salesman (salesman\_id) on delete cascade

);

desc orders;



insert into salesman values (1000, 'john','bangalore','25 %');

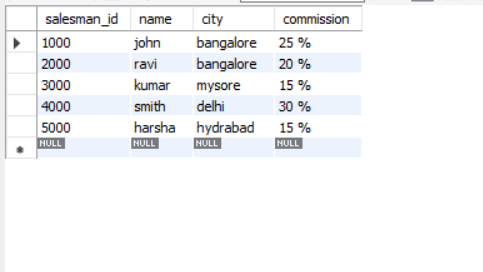
insert into salesman values (2000, 'ravi','bangalore','20 %');

insert into salesman values (3000, 'kumar','mysore','15 %');

insert into salesman values (4000, 'smith','delhi','30 %');

insert into salesman values (5000, 'harsha','hydrabad','15 %');

select \* from salesman;



insert into customer values (10, 'preethi','bangalore', 100, 1000);

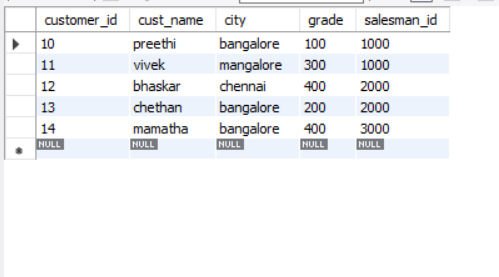
insert into customer values (11,'vivek','mangalore', 300, 1000);

insert into customer values (12, 'bhaskar','chennai', 400, 2000);

insert into customer values (13, 'chethan','bangalore', 200, 2000);

insert into customer values (14, 'mamatha','bangalore', 400, 3000);

select \* from customer;



insert into orders values (50, 5000, '04-06-17', 10, 1000);

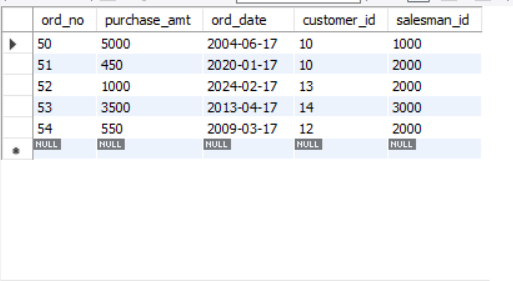
insert into orders values (51, 450, '20-01-17', 10, 2000);

insert into orders values (52, 1000, '24-02-17', 13, 2000);

insert into orders values (53, 3500, '13-04-17', 14, 3000);

insert into orders values (54, 550, '09-03-17', 12, 2000);

select \* from orders;



Additional Queries

select grade, count(distinct customer\_id)

from customer

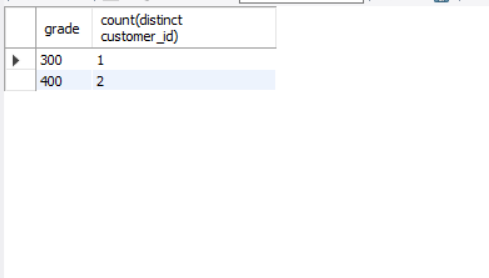
group by grade

having grade > (select avg(grade)

from customer

where city='bangalore'

);



select salesman\_id, name

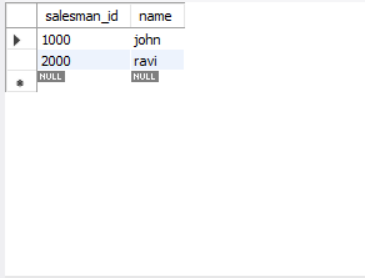
from salesman a

where 1 < (select count(\*)

from customer1

where salesman\_id=a.salesman\_id

);



Select salesman.salesman\_id, name, cust\_name, commission

from salesman, customer

where salesman.city = customer.city

union

select salesman\_id, name, 'no match', commission

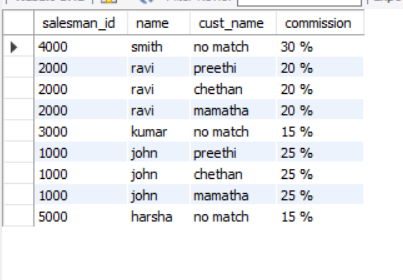
from salesman

where not city = any

(select city

from customer)

order by 2 desc;



create view highsalesman as

select b.ord\_date, a.salesman\_id, a.name

from salesman a, orders b

where a.salesman\_id = b.salesman\_id

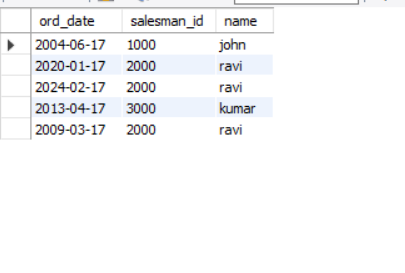
and b.purchase\_amt=(select max(purchase\_amt)

from orders c

where c.ord\_date = b.ord\_date

);

select \* from highsalesman;

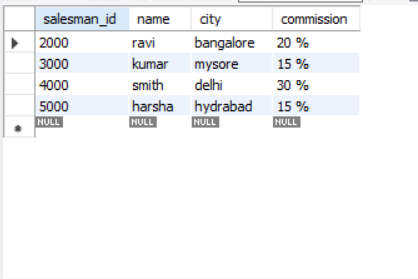


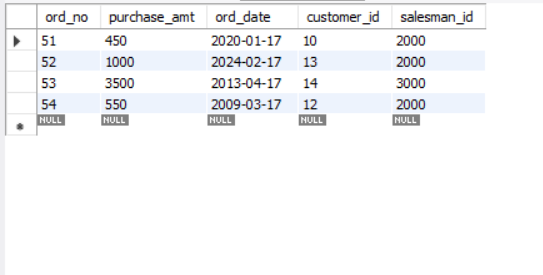
delete from salesman

where salesman\_id=1000;

select \* from salesman;

select \* from orders;





**LAB 7 QUERIES**

**The following tables are maintained by a book dealer:**

**AUTHOR(author-id: int, name: String, city: String, country: String)**

**PUBLISHER(publisher-id: int, name: String, city: String, country: String)**

**CATALOG (book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)**

**CATEGORY(category-id: int, description: String)**

**ORDER-DETAILS(order-no: int, book-id: int, quantity: 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. Give the details of the authors who have 2 or more books in the catalog and the price of the books in the**

**catalog and the year of publication is after 2000.**

**iv. Find the author of the book which has maximum sales.**

**v. Demonstrate how you increase the price of books published by a specific publisher by 10%.**

create database bookdb;

use bookdb;

create table publisher (

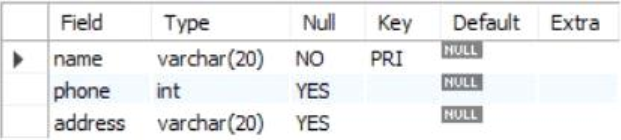
name varchar (20) primary key,

phone integer,

address varchar (20)

);

desc publisher;



create table book (

book\_id integer primary key,

title varchar (20),

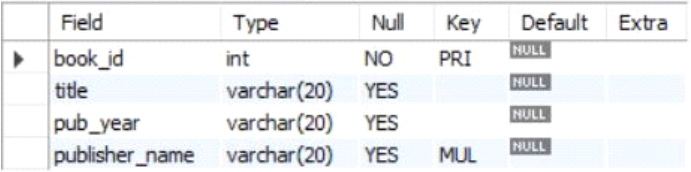
pub\_year varchar (20),

publisher\_name varchar (20),

foreign key (publisher\_name) references publisher (name) on delete cascade

);

desc book;



create table book\_authors (

author\_name varchar (20),

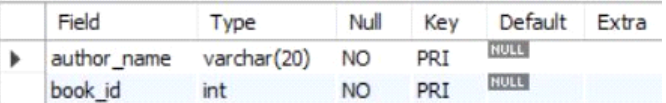
book\_id integer,

foreign key (book\_id) references book (book\_id) on delete cascade,

primary key (book\_id, author\_name)

);

desc book\_authors;



create table library\_branch (

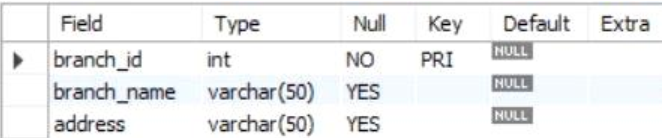
branch\_id integer primary key,

branch\_name varchar (50),

address varchar (50)

);

desc library\_branch;



create table book\_copies (

no\_of\_copies integer,

book\_id integer,

branch\_id integer,

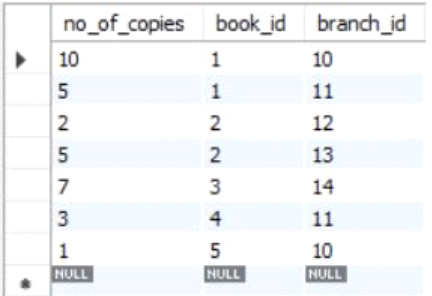
foreign key (book\_id) references book (book\_id) on delete cascade,

foreign key (branch\_id) references library\_branch (branch\_id) on delete cascade,

primary key (book\_id, branch\_id)

);

desc book\_copies;



create table card (

card\_no integer primary key

);

desc card;



create table book\_lending (

date\_out date,

due\_date date,

book\_id integer,

branch\_id integer,

card\_no integer,

foreign key (book\_id) references book (book\_id) on delete cascade,

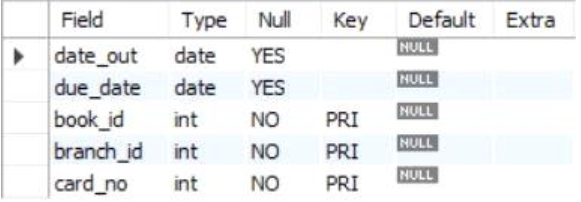
foreign key (branch\_id) references library\_branch (branch\_id) on delete cascade,

foreign key (card\_no) references card (card\_no) on delete cascade,

primary key (book\_id, branch\_id, card\_no)

);

desc book\_lending;



insert into publisher values ('mcgraw-hill', 99890, 'bangalore');

insert into publisher values ('pearson', 98890, 'newdelhi');

insert into publisher values ('random house', 74556, 'hydrabad');

insert into publisher values ('hachette livre', 897086, 'chenai');

insert into publisher values ('grupo planeta', 77561, 'bangalore');

select \* from publisher;



insert into book values (1,'dbms','01-2017', 'mcgraw-hill');

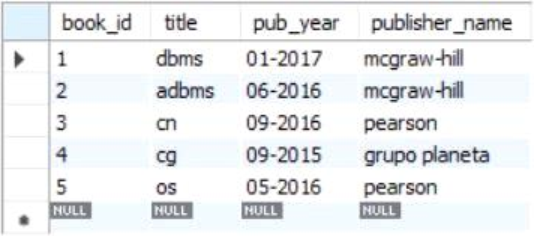
insert into book values (2,'adbms','06-2016', 'mcgraw-hill');

insert into book values (3,'cn','09-2016', 'pearson');

insert into book values (4,'cg','09-2015', 'grupo planeta');

insert into book values (5,'os','05-2016', 'pearson');

select \* from book;



insert into book\_authors values ('navathe', 1);

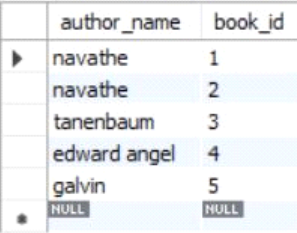
insert into book\_authors values ('navathe', 2);

insert into book\_authors values ('tanenbaum', 3);

insert into book\_authors values ('edward angel', 4);

insert into book\_authors values ('galvin', 5);

select \* from book\_authors;



insert into library\_branch values (10,'rr nagar','bangalore');

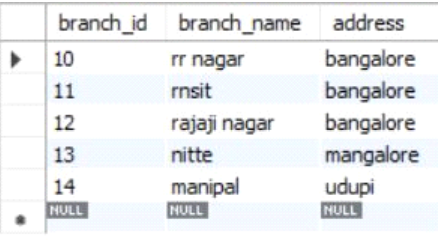
insert into library\_branch values (11,'rnsit','bangalore');

insert into library\_branch values (12,'rajaji nagar', 'bangalore');

insert into library\_branch values (13,'nitte','mangalore');

insert into library\_branch values (14,'manipal','udupi');

select \* from library\_branch;



insert into book\_copies values (10, 1, 10);

insert into book\_copies values (5, 1, 11);

insert into book\_copies values (2, 2, 12);

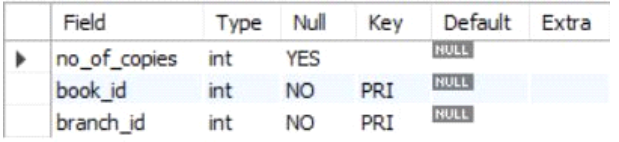
insert into book\_copies values (5, 2, 13);

insert into book\_copies values (7, 3, 14);

insert into book\_copies values (1, 5, 10);

insert into book\_copies values (3, 4, 11);

select \* from book\_copies;



insert into card values (100);

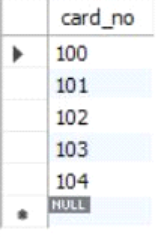
insert into card values (101);

insert into card values (102);

insert into card values (103);

insert into card values (104);

select \* from card;



insert into book\_lending values ('01-01-17','01-06-17', 1, 10, 101);

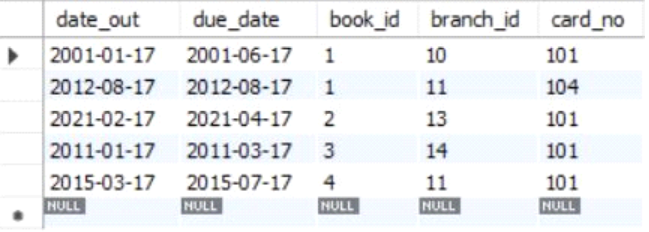
insert into book\_lending values ('11-01-17','11-03-17', 3, 14, 101);

insert into book\_lending values ('21-02-17','21-04-17', 2, 13, 101);

insert into book\_lending values ('15-03-17','15-07-17', 4, 11, 101);

insert into book\_lending values ('12-08-17','12-08-17', 1, 11, 104);

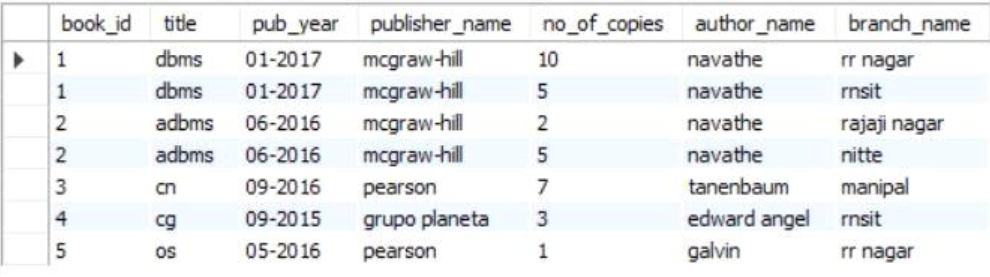
select \* from book\_lending;



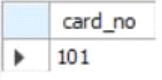
Additional Queries

select b.book\_id, b.title, b.pub\_year, b.publisher\_name, bc.no\_of\_copies, ba.author\_name, lb.branch\_name from book b, book\_authors ba,

library\_branch lb, book\_copies bc where b.book\_id = ba.book\_id and b. book\_id = bc.book\_id and lb.branch\_id = bc.branch\_id;



select card\_no from book\_lending where year(date\_out) >17 and month(date\_out)<7 group by card\_no having count(card\_no) >2 ;



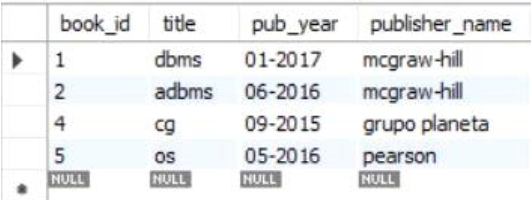
delete from book where book\_id = 3;

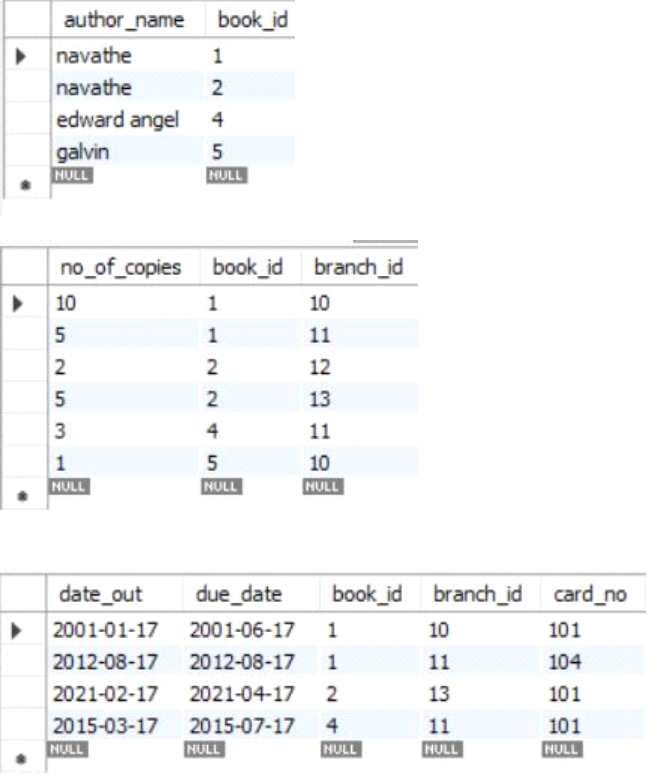
select \* from book;

select \* from book\_authors;

select \* from book\_copies;

select \* from book\_lending;





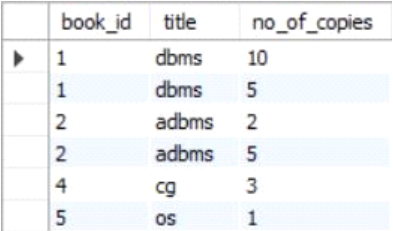
create view partition select pub\_year from book;  
select \* from partition;



create view q5\_view as select b.book\_id, b.title, bc.no\_of\_copies from book b,

book\_copies bc where b.book\_id = bc.book\_id;

select \* from q5\_view;



**LAB 8 QUERIES**

**Consider the following database of student enrollment in courses and books adopted for each course.**

**STUDENT (regno: String, name: String, major: String, bdate: date)**

**COURSE (course #: int, cname: String, dept: String)**

**ENROLL (regno: String, cname: String, sem: int, marks: int)**

**BOOK\_ADOPTION (course #: int, sem: int, book-ISBN: int)**

**TEXT(book-ISBN:int, book-title:String, publisher:String, author:String)**

**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 add a new text book to the database and make this book be adopted by some**

**department.**

**iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses**

**offered by the ‘CS’ department that use more than two books.**

**v. List any department that has all its adopted books published by a specific publisher.**

create database student\_enroll;

use student\_enroll;

create table student(

regno varchar(15),

name varchar(20),

major varchar(20),

bdate date,

primary key(regno));

desc student;

create table course(

courseno int,

cname varchar(20),

dept varchar(20),

primary key(courseno));

desc course;

create table enroll(

regno varchar(15),

courseno int,

sem int,

marks int,

primary key(regno,courseno),

foreign key(regno) references student(regno),

foreign key(courseno) references course(courseno));

desc enroll;

create table textbook(

book\_isbn int,

book\_title varchar(20),

publisher varchar(20),

author varchar(20),

primary key(book\_isbn));

desc textbook;

create table book\_adoption(

courseno int,

sem int,

book\_isbn int,

primary key(courseno,book\_isbn),

foreign key(courseno) references course(courseno),

foreign key(book\_isbn) references textbook(book\_isbn));

desc book\_adoption;

insert into student values('1BM11CS001','A','Sr','19931230');

insert into student values('1BM11CS002','B','Sr','19930924');

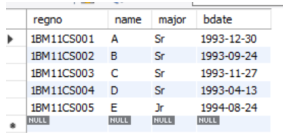
insert into student values('1BM11CS003','C','Sr','19931127');

insert into student values('1BM11CS004','D','Sr','19930413');

insert into student values('1BM11CS005','E','Jr','19940824');

commit;

select \* from student;



insert into course values(111,'OS','CSE');

insert into course values(112,'EC','ECE');

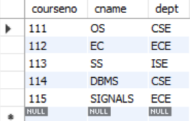
insert into course values(113,'SS','ISE');

insert into course values(114,'DBMS','CSE');

insert into course values(115,'SIGNALS','ECE');

commit;

select \* from course;



insert into textbook values(10,'DATABASE SYSTEMS','PEARSON','SCHIELD');

insert into textbook values(900,'OPERATING SYSTEMS','PEARSON','LELAND');

insert into textbook values(901,'CIRCUITS','HALL INDIA','BOB');

insert into textbook values(902,'SYSTEM SOFTWARE','PETERSON','JACOB');

insert into textbook values(903,'SCHEDULING','PEARSON','PATIL');

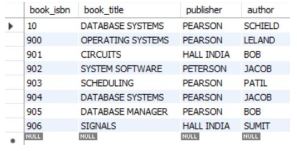
insert into textbook values(904,'DATABASE SYSTEMS','PEARSON','JACOB');

insert into textbook values(905,'DATABASE MANAGER','PEARSON','BOB');

insert into textbook values(906,'SIGNALS','HALL INDIA','SUMIT');

commit;

select \* from textbook;



insert into enroll values('1BM11CS001',115,3,100);

insert into enroll values('1BM11CS002',114,5,100);

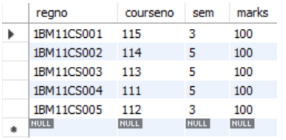
insert into enroll values('1BM11CS003',113,5,100);

insert into enroll values('1BM11CS004',111,5,100);

insert into enroll values('1BM11CS005',112,3,100);

commit;

select \* from enroll;



insert into book\_adoption values(111,5,900);

insert into book\_adoption values(111,5,903);

insert into book\_adoption values(111,5,904);

insert into book\_adoption values(112,3,901);

insert into book\_adoption values(113,3,10);

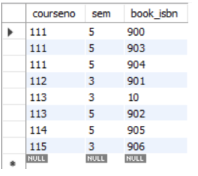
insert into book\_adoption values(114,5,905);

insert into book\_adoption values(113,5,902);

insert into book\_adoption values(115,3,906);

commit;

select \* from book\_adoption;



Additional Queries:

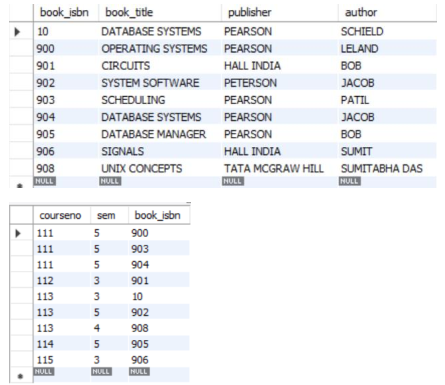
insert into textbook values(908,'UNIX CONCEPTS','TATA MCGRAW

HILL','SUMITABHA DAS');

insert into book\_adoption values(113,4,908);

select \* from textbook;

select \* from book\_adoption;



select c.courseno,t.book\_isbn,t.book\_title

from course c,book\_adoption ba,textbook t

where c.courseno=ba.courseno

and ba.book\_isbn=t.book\_isbn

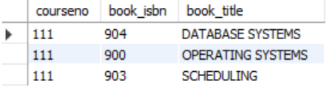
and c.dept='CSE'

and 2<(select COUNT(book\_isbn)

from book\_adoption b

where c.courseno=b.courseno)

order by t.book\_title;



select distinct c.dept

from course c

where c.dept in(select c.dept

from course c,book\_adoption b,textbook t

where c.courseno=b.courseno

and t.book\_isbn=b.book\_isbn

and t.publisher='PEARSON')

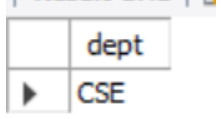
and c.dept not in(select c.dept

from course c,book\_adoption b,textbook t

where c.courseno=b.courseno

and t.book\_isbn=b.book\_isbn

and t.publisher != 'PEARSON');



**LAB 9 QUERIES**

**Consider the schema for Movie Database:**

**ACTOR(Act\_id, Act\_Name, Act\_Gender)**

**DIRECTOR(Dir\_id, Dir\_Name, Dir\_Phone)**

**MOVIES(Mov\_id, Mov\_Title, Mov\_Year, Mov\_Lang, Dir\_id)**

**MOVIE\_CAST(Act\_id, Mov\_id, Role)**

**RATING(Mov\_id, Rev\_Stars)**

**Write SQL queries to**

**i. List the titles of all movies directed by ‘Hitchcock’.**

**ii. Find the movie names where one or more actors acted in two or more movies.**

**iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).**

**iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest**

**number of stars that movie received. Sort the result by movie title.**

**v. Update rating of all movies directed by ‘Steven Spielberg’ to 5.**

create database movie;

use movie;

create table actor(

act\_id int,

act\_name varchar(20),

act\_gender char(1),

primary key(act\_id));

desc actor;

create table director(

dir\_id int,

dir\_name varchar(20),

dir\_phone int(10),

primary key(dir\_id));

desc director;

alter table director

modify column dir\_phone bigint;

desc director;

create table movies(

mov\_id int,

mov\_title varchar(25),

mov\_year int,

mov\_lang varchar(12),

dir\_id int,

primary key(mov\_id),

foreign key(dir\_id) references director(dir\_id));

desc movies;

create table movie\_cast(

act\_id int,

mov\_id int,

role varchar(10),

primary key(act\_id,mov\_id),

foreign key(act\_id) references actor(act\_id),

foreign key(mov\_id) references movies(mov\_id));

desc movie\_cast;

create table rating(

mov\_id int,

rev\_stars varchar(25),

primary key(mov\_id),

foreign key(mov\_id) references movies(mov\_id));

desc rating;

insert into actor values(301,'ANUSHKA','F');

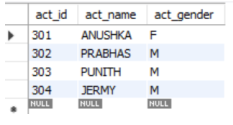
insert into actor values (302,'PRABHAS','M');

insert into actor values(303,'PUNITH','M');

insert into actor values(304,'JERMY','M');

commit;

select \* from actor;



insert into director values(60,'RAJAMOULI', 8751611001);

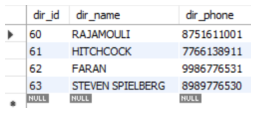
insert into director values(61,'HITCHCOCK', 7766138911);

insert into director values(62,'FARAN', 9986776531);

insert into director values(63,'STEVEN SPIELBERG', 8989776530);

commit;

select \* from director;



insert into movies values(1001,'BAHUBALI-2', 2017, 'TELAGU', 60);

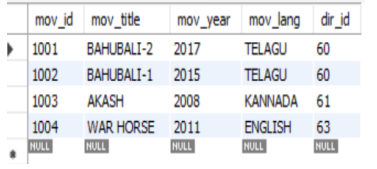
insert into movies values(1002,'BAHUBALI-1', 2015, 'TELAGU', 60);

insert into movies values(1003,'AKASH', 2008, 'KANNADA', 61);

insert into movies values(1004,'WAR HORSE', 2011, 'ENGLISH', 63);

commit;

select \* from movies;



insert into movie\_cast values(301, 1002, 'HEROINE');

insert into movie\_cast values(301, 1001, 'HEROINE');

insert into movie\_cast values(303, 1003, 'HERO');

insert into movie\_cast values(303, 1002, 'GUEST');

insert into movie\_cast values(304, 1004, 'HERO');

commit;

select \* from movie\_cast;



insert into rating values(1001, 4);

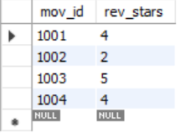
insert into rating values(1002, 2);

insert into rating values(1003, 5);

insert into rating values(1004, 4);

commit;

select \* from rating;



Additional Queries:

select mov\_title from movies m where dir\_id=(select dir\_id from director where dir\_name='Hitchcock');

select mov\_title from movies m,director d where m.dir\_id=d.dir\_id and d.dir\_name='Hitchcock';

group by mov\_title;



select m.mov\_title

from movies m, movie\_cast mc

where m.mov\_id=mc.mov\_id

and mc.act\_id in( select act\_id from movie\_cast group by act\_id having count(act\_id)>1)

group by mov\_title

having count(\*)>1;



select mov\_title,max(rev\_stars)

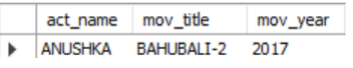
from movies

inner join rating using(mov\_id)

group by mov\_id

having max(rev\_stars)>0

order by mov\_title;



select mov\_title,max(rev\_stars) from movies m,rating r

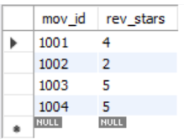
where m.mov\_id=r.mov\_id group by r.mov\_id having max(rev\_stars)>0 order by mov\_title;



update rating set rev\_stars=5

where mov\_id in(select mov\_id from movies where dir\_id in(select dir\_id from director where dir\_name='Steven Spielberg'));

select \*from rating;



**LAB 10 QUERIES**

**Consider the schema for College Database:**

**STUDENT(USN, SName, Address, Phone, Gender)**

**SEMSEC(SSID, Sem, Sec)**

**CLASS(USN, SSID)**

**SUBJECT(Subcode, Title, Sem, Credits)**

**IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)**

**Write SQL queries to**

**i. List all the student details studying in fourth semester ‘C’ section.**

**ii. Compute the total number of male and female students in each semester and in each section.**

**iii. Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.**

**iv. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.**

**v. Categorize students based on the following criterion:**

**If FinalIA = 17 to 20 then CAT = ‘Outstanding’**

**If FinalIA = 12 to 16 then CAT = ‘Average’**

**If FinalIA< 12 then CAT = ‘Weak’**

**Give these details only for 8th semester A, B, and C section students.**

create database collegedb;

use collegedb;

create table student (

usn varchar (10),

sname varchar (25),

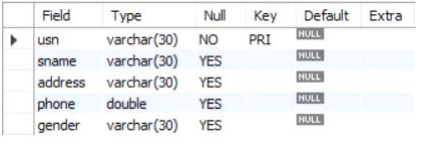
address varchar (25),

phone long,

gender char (1),

primary key (usn));

desc student;



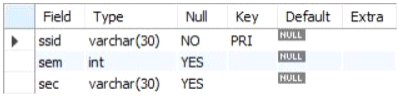
create table semsec (

ssid varchar (5),

sem int,

sec char (1),

desc semsec;



create table class (

usn varchar (10),

ssid varchar (5),

primary key (usn, ssid),

foreign key (usn) references student (usn),

foreign key (ssid) references semsec (ssid));

desc class;



create table subject (

subcode varchar (8),

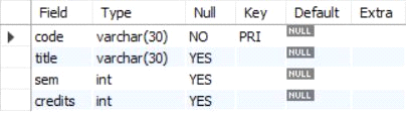
title varchar (20),

sem int,

credits int,

primary key (subcode));

desc subject;



create table iamarks (

usn varchar (10),

subcode varchar (8),

ssid varchar (5),

test1 int,

test2 int,

test3 int,

finalia int,

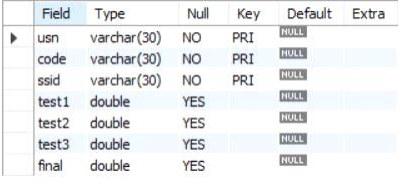
primary key (usn, subcode, ssid),

foreign key (usn) references student (usn),

foreign key (subcode) references subject (subcode),

foreign key (ssid) references semsec (ssid));

desc marks;



insert into student values ('1rn13cs020','akshay','belagavi', 8877881122,'m');

insert into student values ('1rn13cs062','sandhya','bengaluru', 7722829912,'f');

insert into student values ('1rn13cs091','teesha','bengaluru', 7712312312,'f');

insert into student values ('1rn13cs066','supriya','mangaluru', 8877881122,'f');

insert into student values ('1rn14cs010','abhay','bengaluru', 9900211201,'m');

insert into student values ('1rn14cs032','bhaskar','bengaluru', 9923211099,'m');

insert into student values ('1rn14cs025','asmi','bengaluru', 7894737377,'f');

insert into student values ('1rn15cs011','ajay','tumkur', 9845091341,'m');

insert into student values ('1rn15cs029','chitra','davangere', 7696772121,'f');

insert into student values ('1rn15cs045','jeeva','bellary', 9944850121,'m');

insert into student values ('1rn15cs091','santosh','mangaluru', 8812332201,'m');

insert into student values ('1rn16cs045','ismail','kalburgi', 9900232201,'m');

insert into student values ('1rn16cs088','sameera','shimoga', 9905542212,'f');

insert into student values ('1rn16cs122','vinayaka','chikamagalur', 8800880011,'m');

select \* from student;



insert into semsec values ('cse8a', 8,'a');

insert into semsec values ('cse8b', 8,'b');

insert into semsec values ('cse8c', 8,'c');

insert into semsec values ('cse7a', 7,'a');

insert into semsec values ('cse7b', 7,'b');

insert into semsec values ('cse7c', 7,'c');

insert into semsec values ('cse6a', 6,'a');

insert into semsec values ('cse6b', 6,'b');

insert into semsec values ('cse6c', 6,'c');

insert into semsec values ('cse5a', 5,'a');

insert into semsec values ('cse5b', 5,'b');

insert into semsec values ('cse5c', 5,'c');

insert into semsec values ('cse4a', 4,'a');

insert into semsec values ('cse4b', 4,'b');

insert into semsec values ('cse4c', 4,'c');

insert into semsec values ('cse3a', 3,'a');

insert into semsec values ('cse3b', 3,'b');

insert into semsec values ('cse3c', 3,'c');

insert into semsec values ('cse2a', 2,'a');

insert into semsec values ('cse2b', 2,'b');

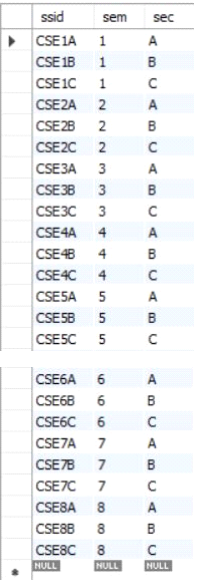
insert into semsec values ('cse2c', 2,'c');

insert into semsec values ('cse1a', 1,'a');

insert into semsec values ('cse1b', 1,'b');

insert into semsec values ('cse1c', 1,'c');

select \* from semsec;



insert into class values ('1rn13cs020','cse8a');

insert into class values ('1rn13cs062','cse8a');

insert into class values ('1rn13cs066','cse8b');

insert into class values ('1rn13cs091','cse8c');

insert into class values ('1rn14cs010','cse7a');

insert into class values ('1rn14cs025','cse7a');

insert into class values ('1rn14cs032','cse7a');

insert into class values ('1rn15cs011','cse4a');

insert into class values ('1rn15cs029','cse4a');

insert into class values ('1rn15cs045','cse4b');

insert into class values ('1rn15cs091','cse4c');

insert into class values ('1rn16cs045','cse3a');

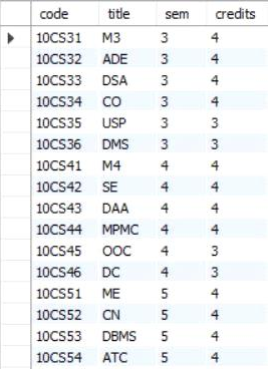
insert into class values ('1rn16cs088','cse3b');

insert into class values ('1rn16cs122','cse3c');

select \* from class;



insert into subject values('10cs81','aca',8,4),  
('10cs82','ssm',8,4),('10cs83','nm',8,4),  
('10cs84','cc',8,4),('10cs85','pw',8,4),  
('10cs71','ooad',7,4),('10cs72','ecs',7,4),  
('10cs73','ptw',7,4),('10cs74','dwdm',7,4),  
('10cs75','java',7,4),('10cs76','san',7,4),  
('10cs51','me',5,4),('10cs52','cn',5,4),  
('10cs53','dbms',5,4),('10cs54','atc',5,4),  
('10cs55','java',5,3),('10cs56','ai',5,3),  
('10cs41','m4',4,4),('10cs42','se',4,4),  
('10cs43','daa',4,4),('10cs44','mpmc',4,4),  
('10cs45','ooc',4,3),('10cs46','dc',4,3),  
('10cs31','m3',3,4),('10cs32','ade',3,4),  
('10cs33','dsa',3,4),('10cs34','co',3,4),  
('10cs35','usp',3,3),('10cs36','dms',3,3);  
select \* from subject;



insert into marks (usn, subcode, ssid, test1, test2, test3) values ('1rn13cs091','10cs81','cse8c', 15, 16, 18);

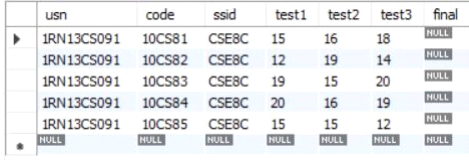
insert into marks (usn, subcode, ssid, test1, test2, test3) values ('1rn13cs091','10cs82','cse8c', 12, 19, 14);

insert into marks (usn, subcode, ssid, test1, test2, test3) values ('1rn13cs091','10cs83','cse8c', 19, 15, 20);

insert into marks (usn, subcode, ssid, test1, test2, test3) values ('1rn13cs091','10cs84','cse8c', 20, 16, 19);

insert into marks (usn, subcode, ssid, test1, test2, test3) values ('1rn13cs091','10cs85','cse8c', 15, 15, 12);

select \* from marks



Additional Queries:

select s.\*, ss.sem, ss.sec

from student s, semsec ss, class c

where s.usn = c.usn and

ss.ssid = c.ssid and

ss.sem = 4 and ss.sec='c';



select ss.sem, ss.sec, s.gender, count(s.gender) as count

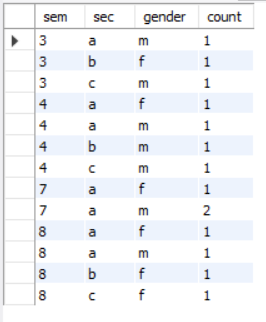
from student s, semsec ss, class c

where s.usn = c.usn and

ss.ssid = c.ssid

group by ss.sem, ss.sec, s.gender

order by sem;



create view stu\_test1\_marks\_view

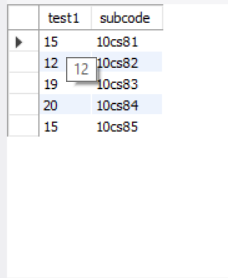
as

select test1, subcode

from iamarks

where usn = '1rn13cs091';

select \* from stu\_test1\_marks\_view;



if finalia = 17 to 20 then cat = ‘outstanding’

if finalia = 12 to 16 then cat = ‘average’

if finalia< 12 then cat = ‘weak’

give these details only for 8th semester a, b, and c section students. \*/

select s.usn,s.sname,s.address,s.phone,s.gender,

(case

when ia.finalia between 17 and 20 then 'outstanding'

when ia.finalia between 12 and 16 then 'average'

else 'weak'

end) as cat

from student s, semsec ss, iamarks ia, subject sub

where s.usn = ia.usn and

ss.ssid = ia.ssid and

sub.subcode = ia.subcode and

sub.sem = 8;

