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;
```

Result Grid | Filter Rows: | Edit: | Export/Import

driver_id	name	address
A01	Raghu	Electronic City
A02	Rishab	Orange County
A03	Rufus	NR Colony
A04	Jamal	Lawrence Park
A05	Kevin	Rosedale
NULL	NULL	NULL

Result Grid | Form Editor | Field Types

Result 7 | Result 8 | person 9 | car 10 | accident 11 | owns 12 | pa | Apply

```
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;
```

Result Grid | Filter Rows: | Edit: | Export/Import

reg_num	model	year
KA031111	Accord	2005
KA041122	MX-5	2019
KA051133	Indica	2010
KA061144	Prius	2015
KA071155	Camry	2020
NULL	NULL	NULL

Result Grid | Form Editor | Field Types

Result 7 | Result 8 | person 9 | car 10 | accident 11 | owns 12 | pa | Apply

```
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;
```

Result Grid | Filter Rows: | Edit: | Export/Import

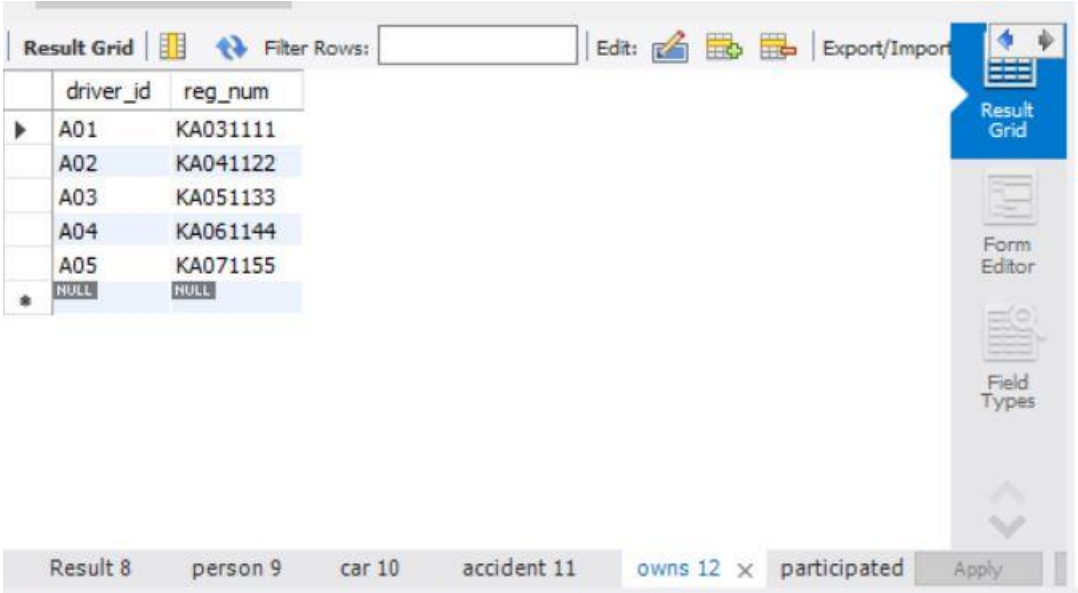
report_num	accident_date	location
11	2008-01-01	NR Road
12	2008-02-02	Dalhousie Road
13	2020-03-03	Henry Road
14	2020-04-04	Beehive Road
15	2020-05-05	Orange Street
NULL	NULL	NULL

Result Grid | Form Editor | Field Types

Result 7 | Result 8 | person 9 | car 10 | accident 11 | owns 12 | pa | Apply

```
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;
```



driver_jd	reg_num
A01	KA031111
A02	KA041122
A03	KA051133
A04	KA061144
A05	KA071155
NULL	NULL

```
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;

Result Grid | Filter Rows: | Edit: | Export/Import

driver_id	reg_num	report_num	damage_amount
A01	KA031111	11	10000
A02	KA041122	12	20000
A03	KA051133	13	30000
A04	KA061144	14	40000
A05	KA071155	15	50000
NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types

Result 8 | person 9 | car 10 | accident 11 | owns 12 | participated | Apply

Additional Queries:

update participated

set damage_amount = 2500

where reg_num='KA031111';

select * from participated;

```
damage_amount      |
+ -------------- + ------------ + --------------- +
------------------ +
| A01            | KA031111     | 11             | 10000
        |
| A02            | KA041122     | 12             | 25000
        |
| A03            | KA051133     | 13             | 30000
        |
| A04            | KA061144     | 14             | 40000
        |
| A05            | KA071155     | 15             | 50000
        |
| NULL           | NULL         | NULL           | NULL
```

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

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

commit;

select * from accident;

select * from participated;

report_num	accident_date	location
11	2008-01-01	NR Road
12	2008-02-02	Dalhousie Road
13	2020-03-03	Henry Road
14	2020-04-04	Beehive Road
15	2020-05-05	Orange Street
16	2008-03-08	Domlur
NULL	NULL	NULL

7 rows

participated 15 Query Output Query Output Apply

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);

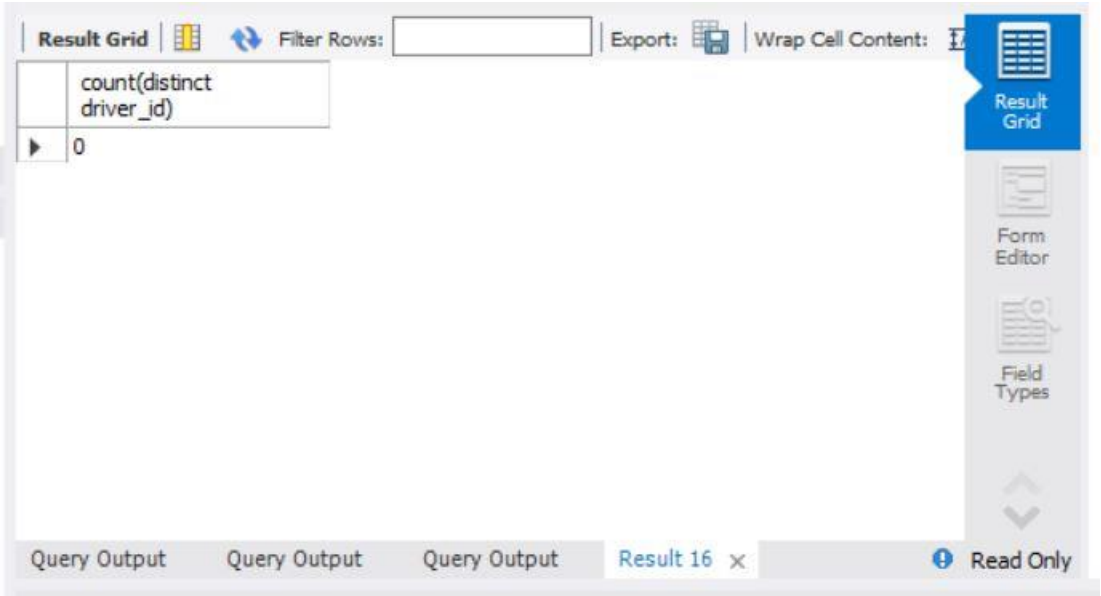
commit;

select * from car;

select * from owns;

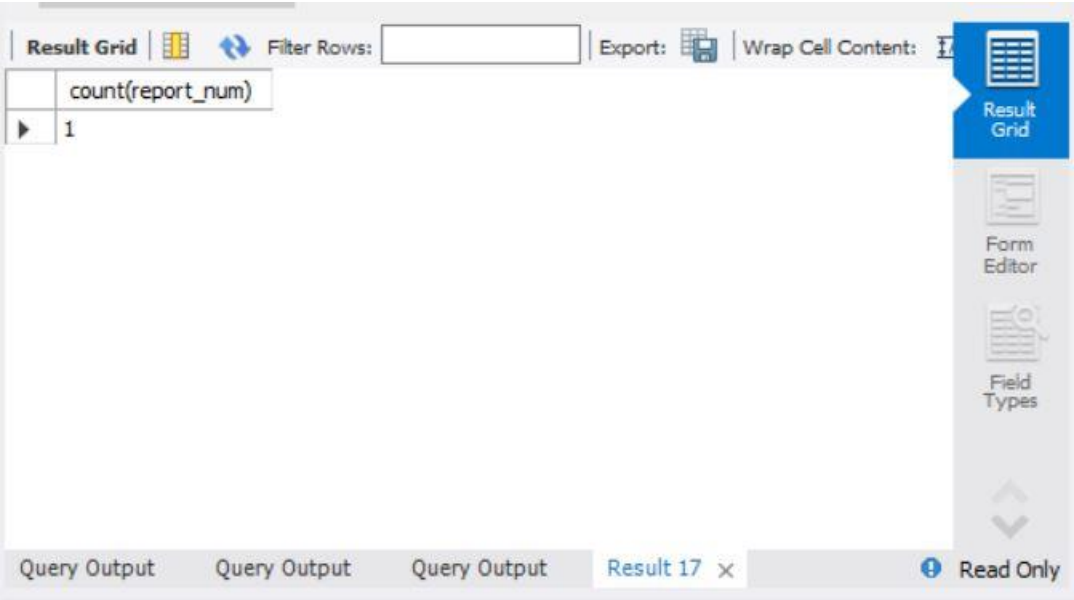
select * from accident;

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)
  );
```

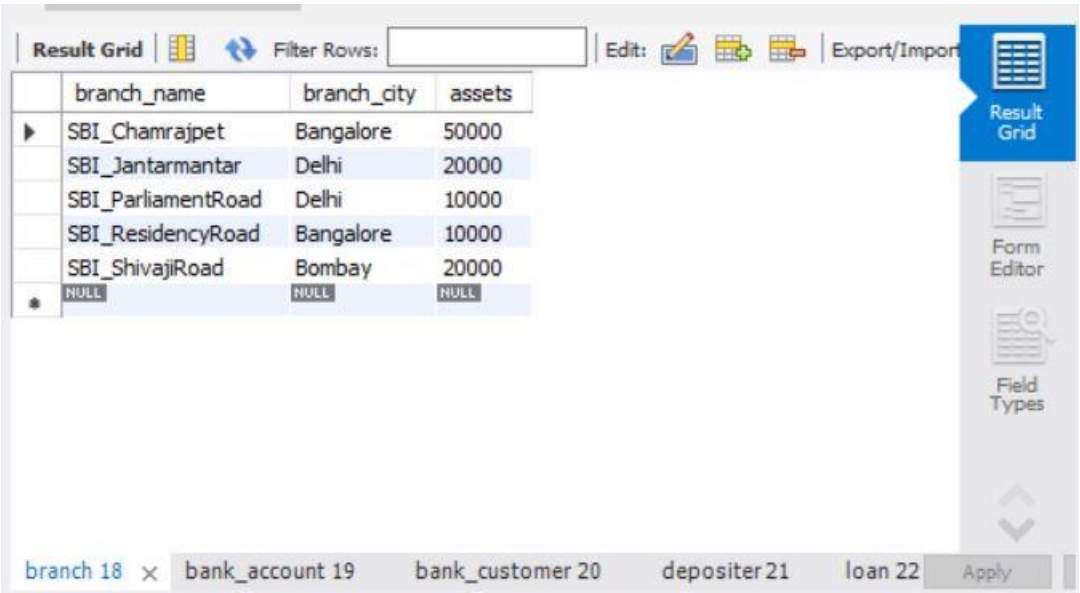
```
create table bank_account (
        accno int,
    branch_name varchar(25),
    balance int,
    primary key (accno),
    foreign key (branch_name) references branch(branch_name)
);


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;
```



branch_name	branch_city	assets
SBI_Chamrajpet	Bangalore	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParliamentRoad	Delhi	10000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

```
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);

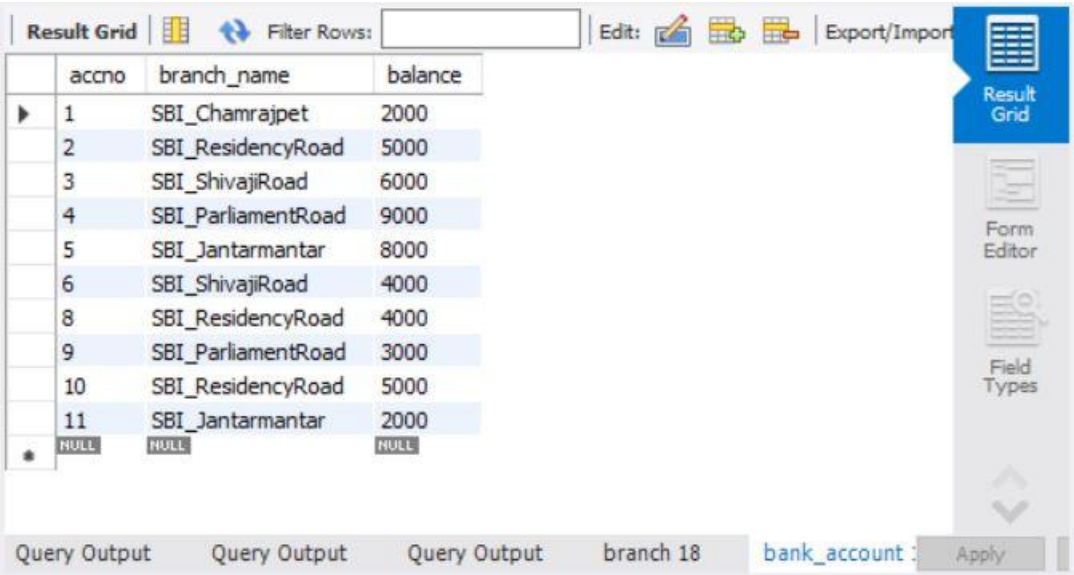
insert into bank_account values(6, 'SBI_ShivajiRoad', 4000);

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);



accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

commit;

insert into bank_customer values ('Abhay', 'Bull_Temple_Road', 'Bangalore');

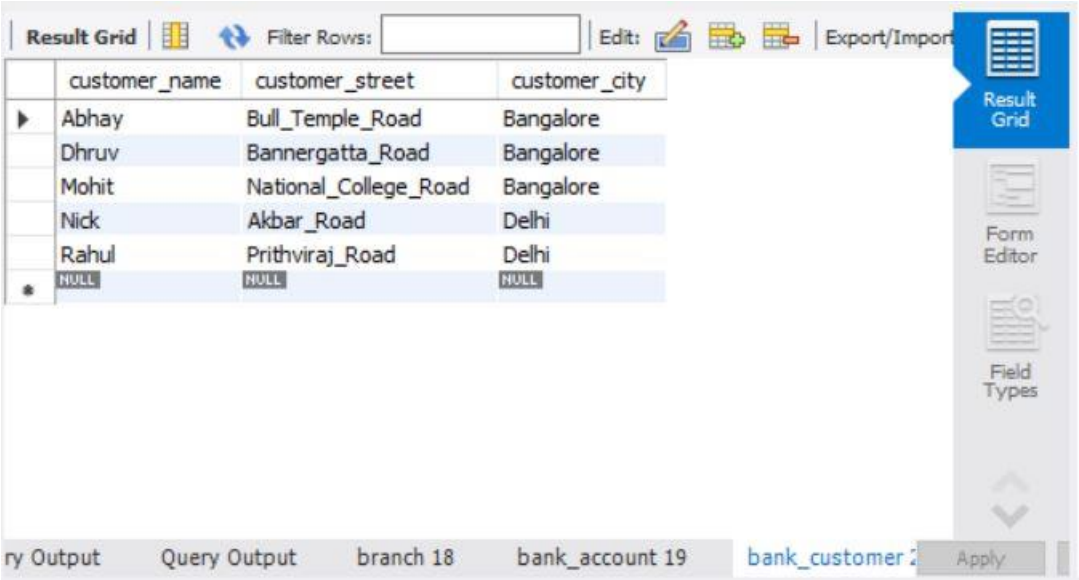
insert into bank_customer values ('Dhruv', 'Bannergatta_Road', 'Bangalore');

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;



customer_name	customer_street	customer_city
Abhay	Bull_Temple_Road	Bangalore
Dhruv	Bannergatta_Road	Bangalore
Mohit	National_College_Road	Bangalore
Nick	Akbar_Road	Delhi
Rahul	Prithviraj_Road	Delhi
NULL	NULL	NULL

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;

Result Grid | Filter Rows: | Edit: | Export/Import

customer_name	accno
Abhay	1
Dhruv	2
Nick	4
Rahul	5
Abhay	8
Nick	9
Dhruv	10
Nick	11
NULL	NULL

k_account 19 | bank_customer 20 | depositer 21 | loan 22 | bank_account 2 | Apply

```
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;
```

Result Grid | Filter Rows: | Edit: | Export/Import

loan_number	branch_name	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParliamentRoad	4000
5	SBI_Jantarmantar	5000
NULL	NULL	NULL

k_account 19 | bank_customer 20 | depositer 21 | loan 22 | bank_account 2 | Apply

select * from branch;

select * from bank_account;

select * from bank_customer;

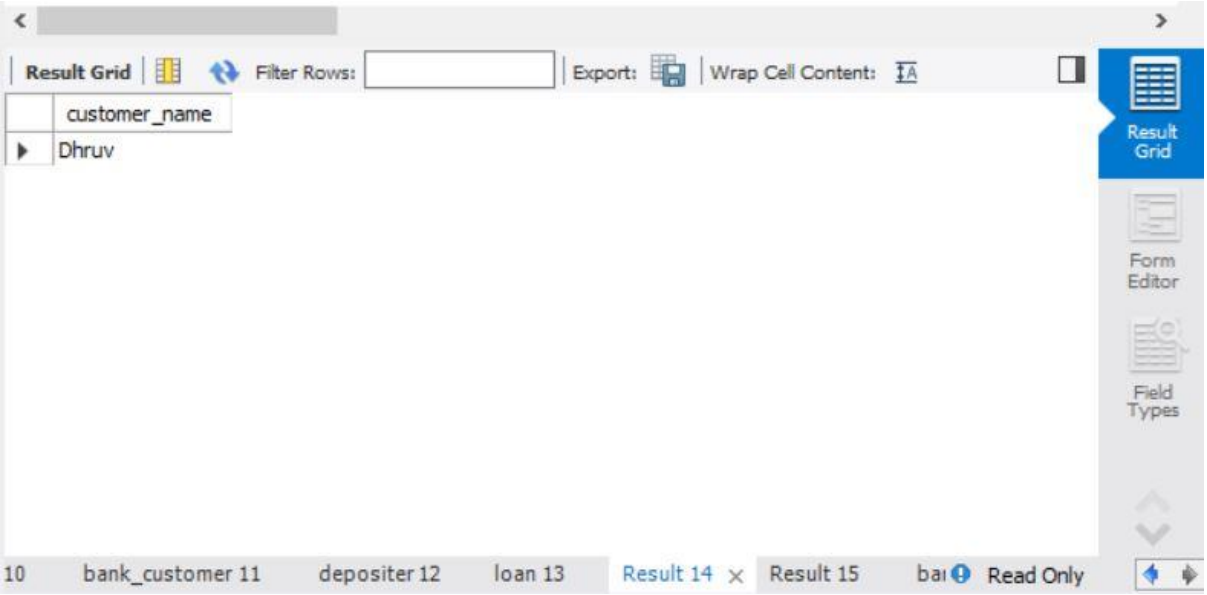
select * from depositer;

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);

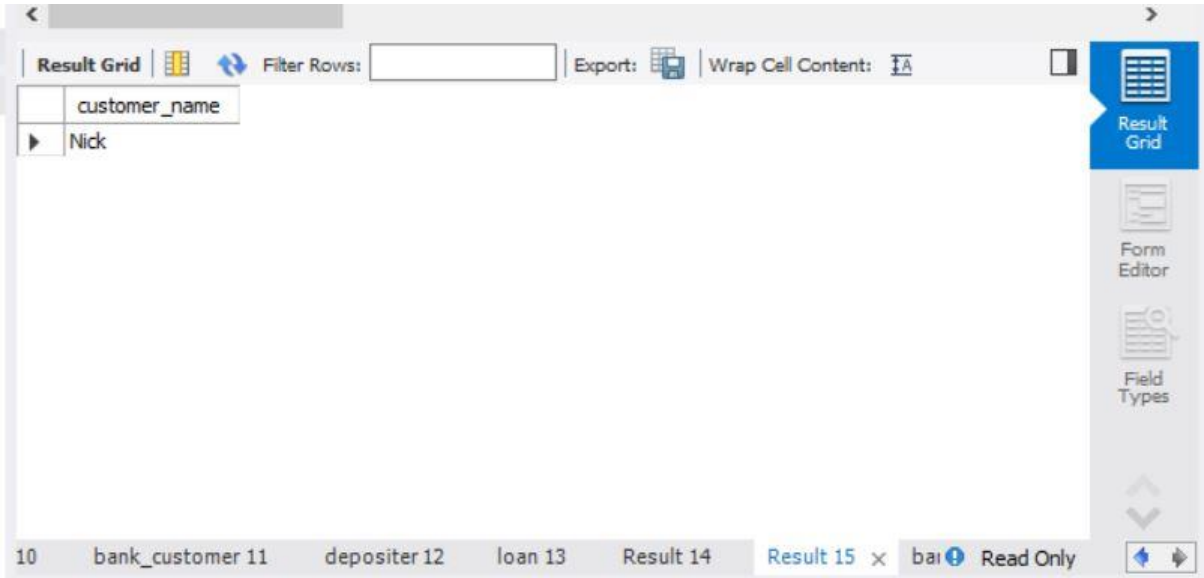


select d.customer_name from depositer d,branch b,bank_account a

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;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000
NULL	NULL	NULL

Result Grid | Form Editor | Field Types

10 | bank_customer 11 | depositer 12 | loan 13 | Result 14 | Result 15 | bai | Apply | Revert

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');

sid	sname	address
10001	Acme Widget	Bangalore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi
NULL	NULL	NULL

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;

pid	pname	color
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black
NULL	NULL	NULL

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;
```

pname
Book
Pen
Pencil
Mobile
Charger

```
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));
```

Result Grid | Filter Rows:

sname
Acme Widget

```
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)));

Result Grid | Filter Rows:

sname
Acme Widget
Johns

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');

Result Grid | Filter Rows:

pname
Mobile
Charger

SELECT DISTINCT C.sid FROM Catalog C

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

```
FROM Catalog C1
WHERE C1.pid = C.pid );
```

	sid
▶	10002
	10004

```
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);
```

Result Grid | Filter Rows:

	pid	sname
▶	20001	Acme Widget
	20004	Acme Widget
	20005	Acme Widget
	20001	Johns
	20002	Johns
	20003	Reliance

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);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

snum	sname	major	lvl	age
1	jhon	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Rita	CS	Sr	21
NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types

STUDENT 29 × | FACULTY 30 | CLASS 31 | ENROLLED 32 | Result 33 | class 34 | Studen | Apply

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);

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

fid	fname	deptid
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000
NULL	NULL	NULL

Result Grid | Form Editor | Field Types

STUDENT 29 | FACULTY 30 × | CLASS 31 | ENROLLED 32 | Result 33 | class 34 | Studen | Apply

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);

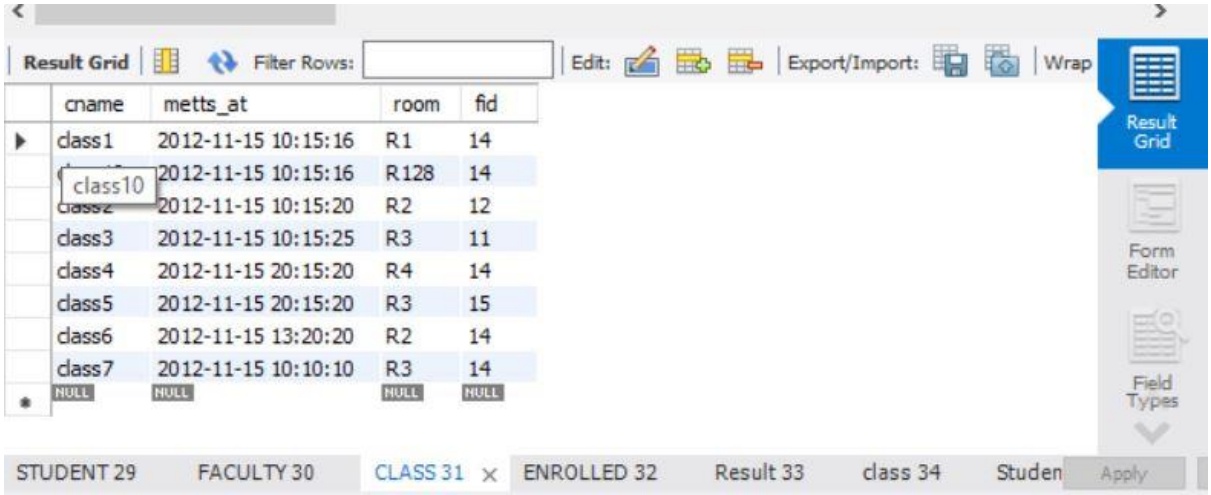
insert into class values('class3', '12/11/15 10:15:25', 'R3', 11);

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);



cname	metts_at	room	fid
class1	2012-11-15 10:15:16	R1	14
class10	2012-11-15 10:15:16	R128	14
class2	2012-11-15 10:15:20	R2	12
class3	2012-11-15 10:15:25	R3	11
class4	2012-11-15 20:15:20	R4	14
class5	2012-11-15 20:15:20	R3	15
class6	2012-11-15 13:20:20	R2	14
class7	2012-11-15 10:10:10	R3	14
NULL	NULL	NULL	NULL

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');

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

snum	cname
1	class1
2	class1
3	class3
4	class3
5	class4
1	class5
2	class5
3	class5
4	class5
5	class5

STUDENT 29 | FACULTY 30 | CLASS 31 | ENROLLED 32 | Result 33 | class 34 | Studen | Apply

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';

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Sname
Tom

ENROLLED 43 | Result 44 | class 45 | Student 46 | faculty 47 | Faculty 48 | Student 49 | Read Only

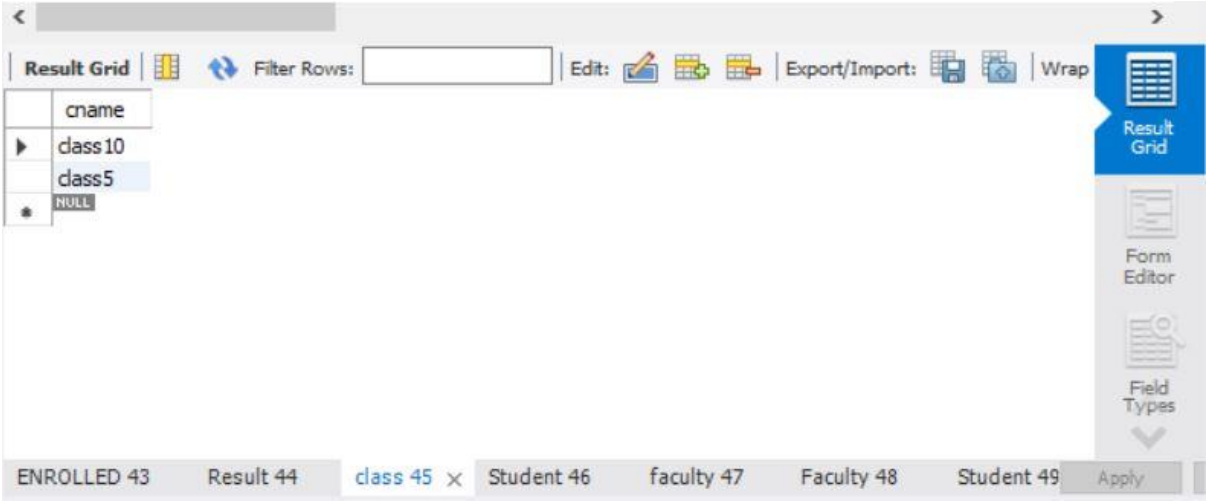
SELECT DISTINCT cname

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

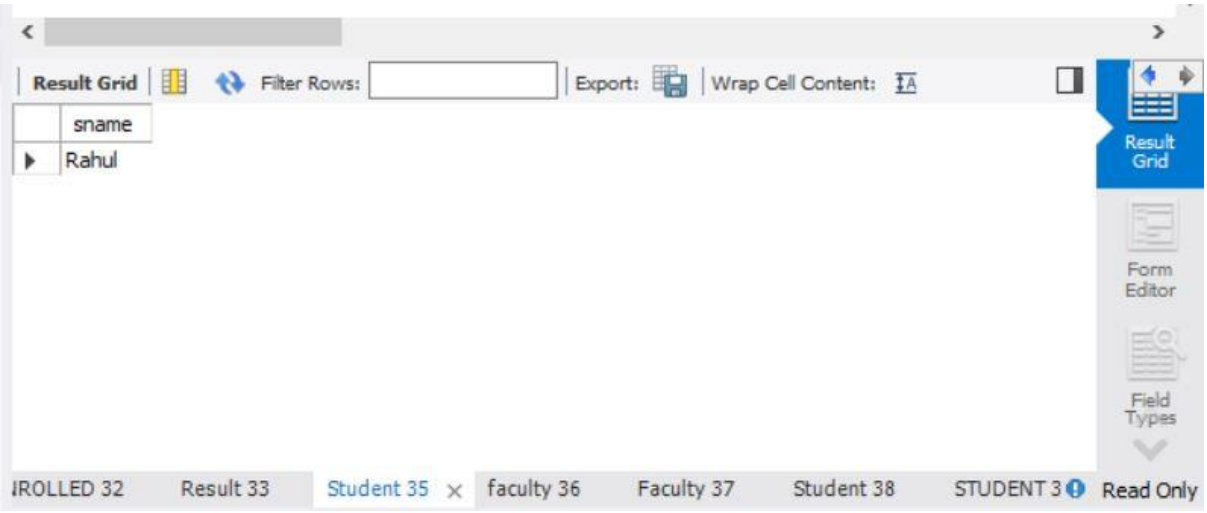
WHERE S.snum IN (SELECT E1.snum

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);

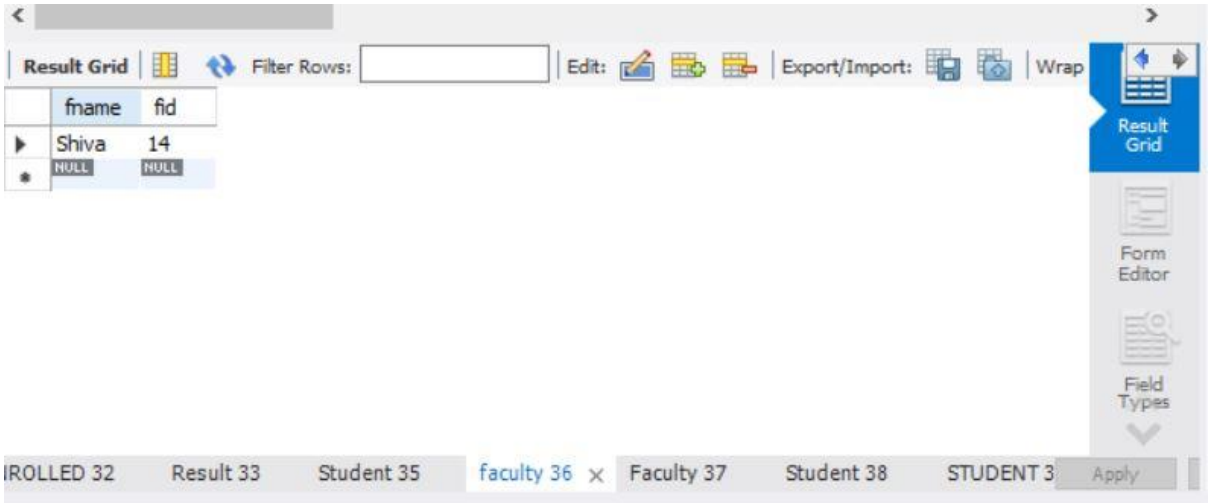


SELECT f.fname,f.fid

FROM faculty f

WHERE f.fid in (SELECT fid FROM class

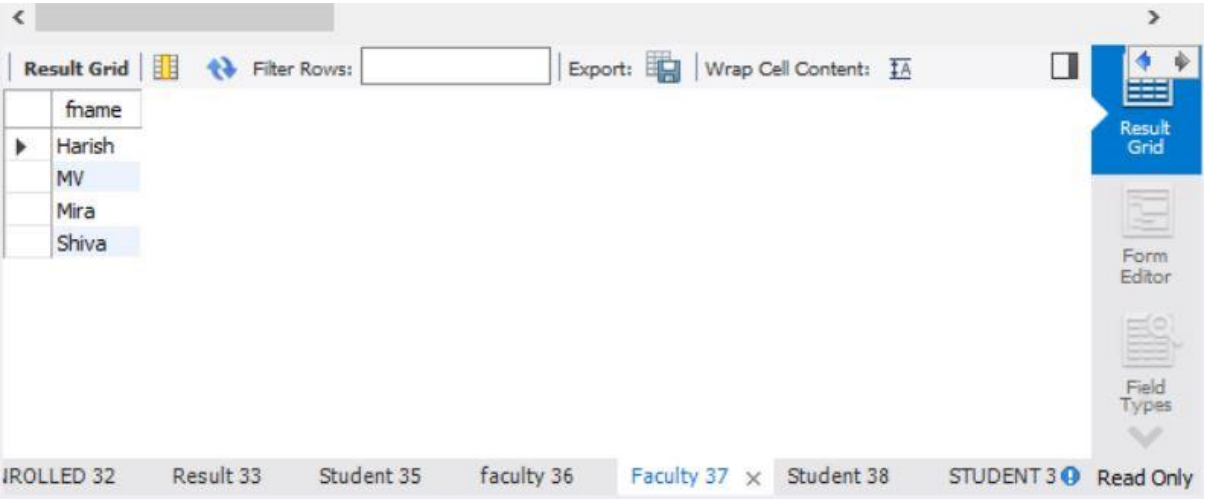
GROUP BY fid HAVING COUNT(*)=(SELECT COUNT(DISTINCT room) FROM class));



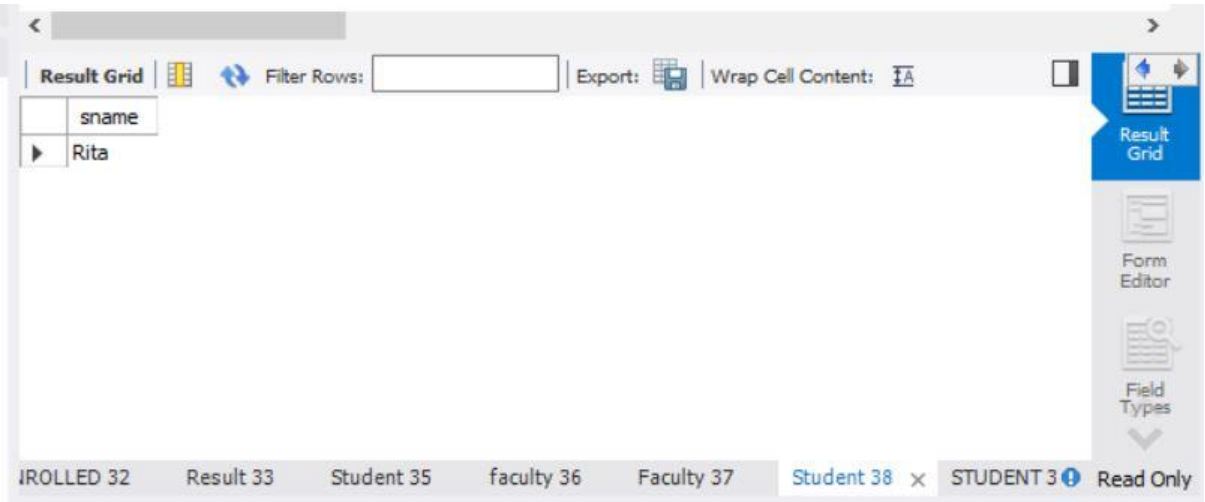
SELECT DISTINCT F.fname

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);

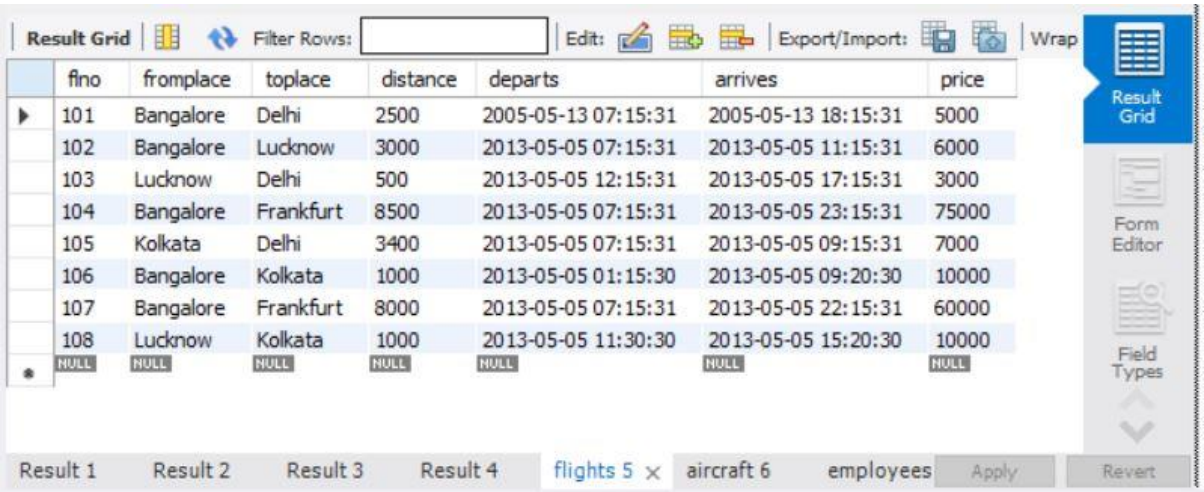
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31', 7000);

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;

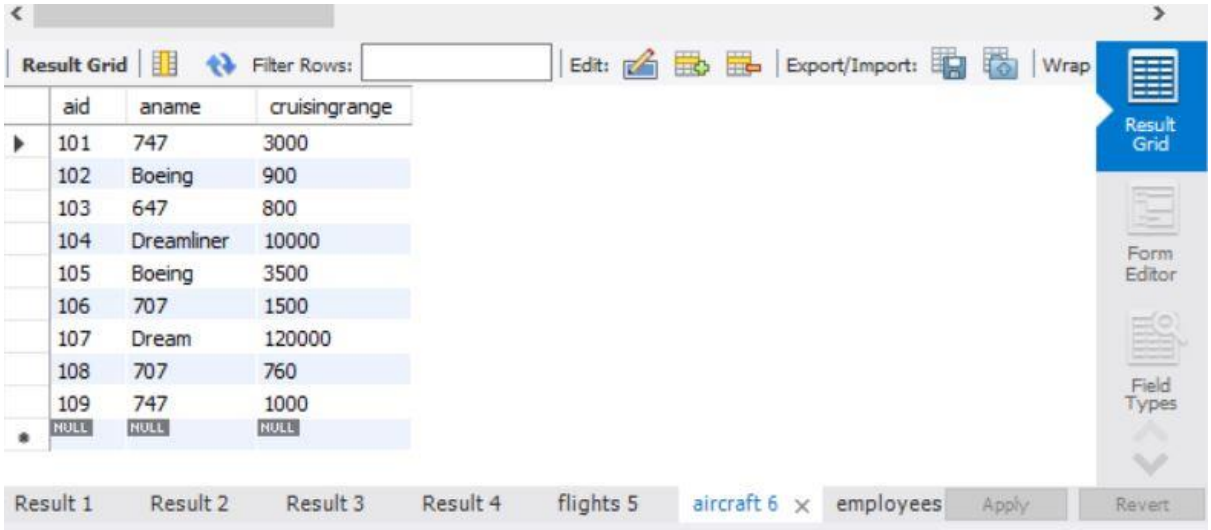
select * from flights;



fno	fromplace	toplace	distance	departs	arrives	price
101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 18:15:31	5000
102	Bangalore	Lucknow	3000	2013-05-05 07:15:31	2013-05-05 11:15:31	6000
103	Lucknow	Delhi	500	2013-05-05 12:15:31	2013-05-05 17:15:31	3000
104	Bangalore	Frankfurt	8500	2013-05-05 07:15:31	2013-05-05 23:15:31	75000
105	Kolkata	Delhi	3400	2013-05-05 07:15:31	2013-05-05 09:15:31	7000
106	Bangalore	Kolkata	1000	2013-05-05 01:15:30	2013-05-05 09:20:30	10000
107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
108	Lucknow	Kolkata	1000	2013-05-05 11:30:30	2013-05-05 15:20:30	10000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
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;
```

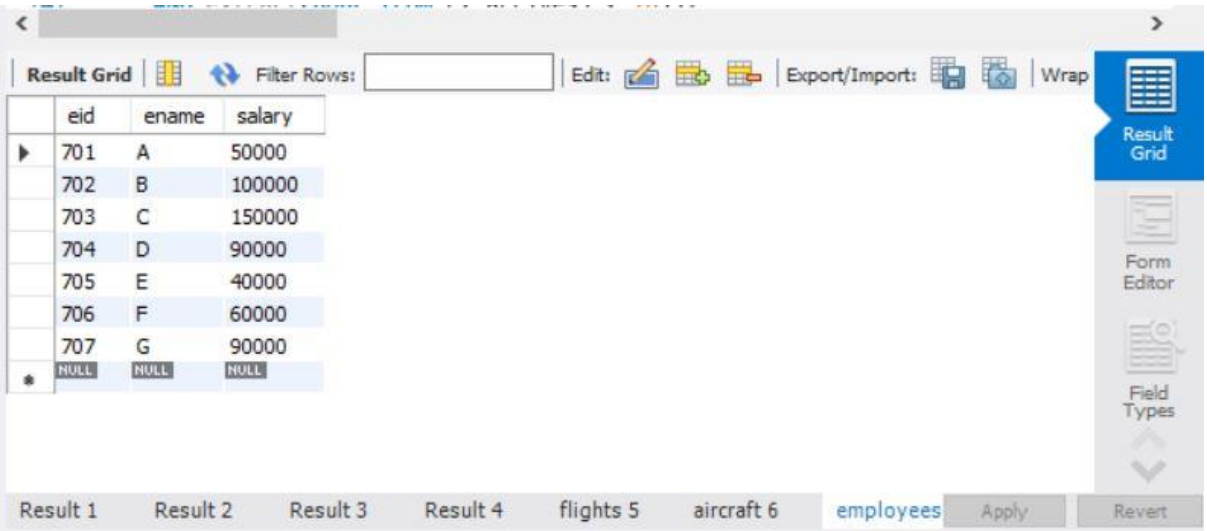


aid	aname	cruisingrange
101	747	3000
102	Boeing	900
103	647	800
104	Dreamliner	10000
105	Boeing	3500
106	707	1500
107	Dream	120000
108	707	760
109	747	1000
NULL	NULL	NULL

```
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;
```



eid	ename	salary
701	A	50000
702	B	100000
703	C	150000
704	D	90000
705	E	40000
706	F	60000
707	G	90000
NULL	NULL	NULL

```
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

eid	aid
701	101
701	102
701	106
701	105
702	104
703	104
704	104
702	107
703	107
704	107
702	101
702	108
701	109

Result Grid | Form Editor | Field Types | Query Stats

esult 1 | Result 2 | Result 3 | Result 4 | flights 5 | aircraft 6 | employees 7 | certified | Read Only

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
   )
);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

aname
747
Dreamliner
Dream
707

Result Grid | Form Editor | Field Types | Query Stats

Result 3 | Result 4 | flights 5 | aircraft 6 | employees 7 | certified 8 | aircraft 9 | Re | Read Only

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;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

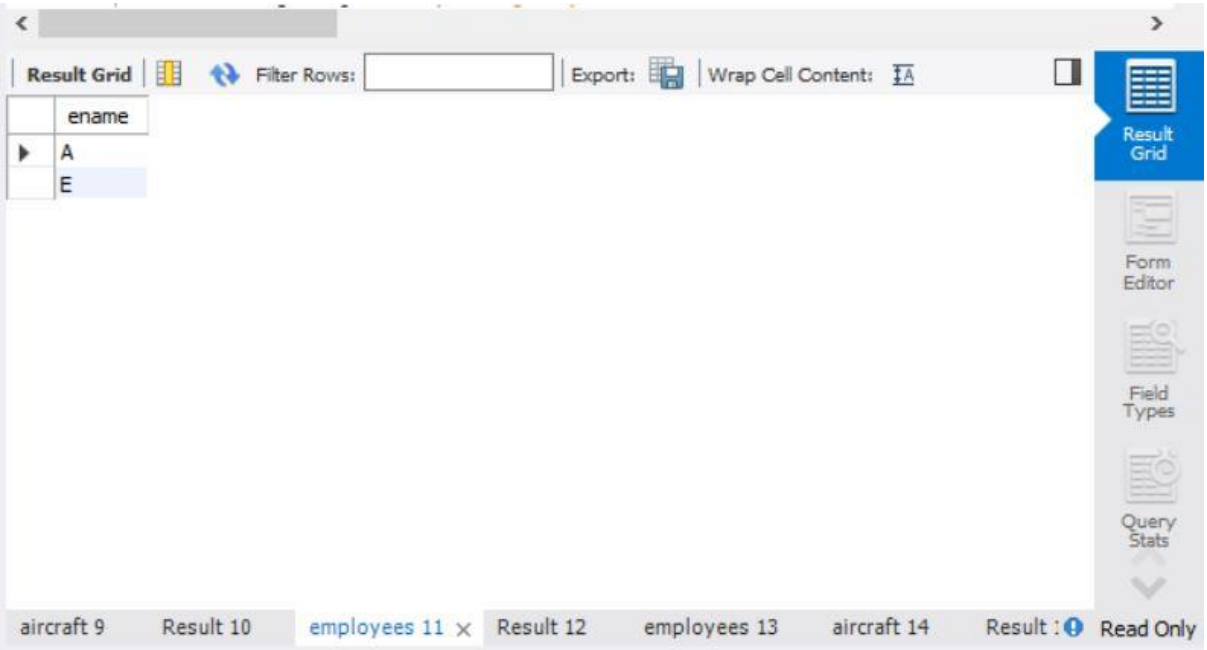
max(a.cruisingrange)	eid
3500	701
120000	702

Result Grid | Form Editor | Field Types | Query Stats

aircraft 9 | Result 10 | employees 11 | Result 12 | employees 13 | aircraft 14 | Result : | Read Only

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;

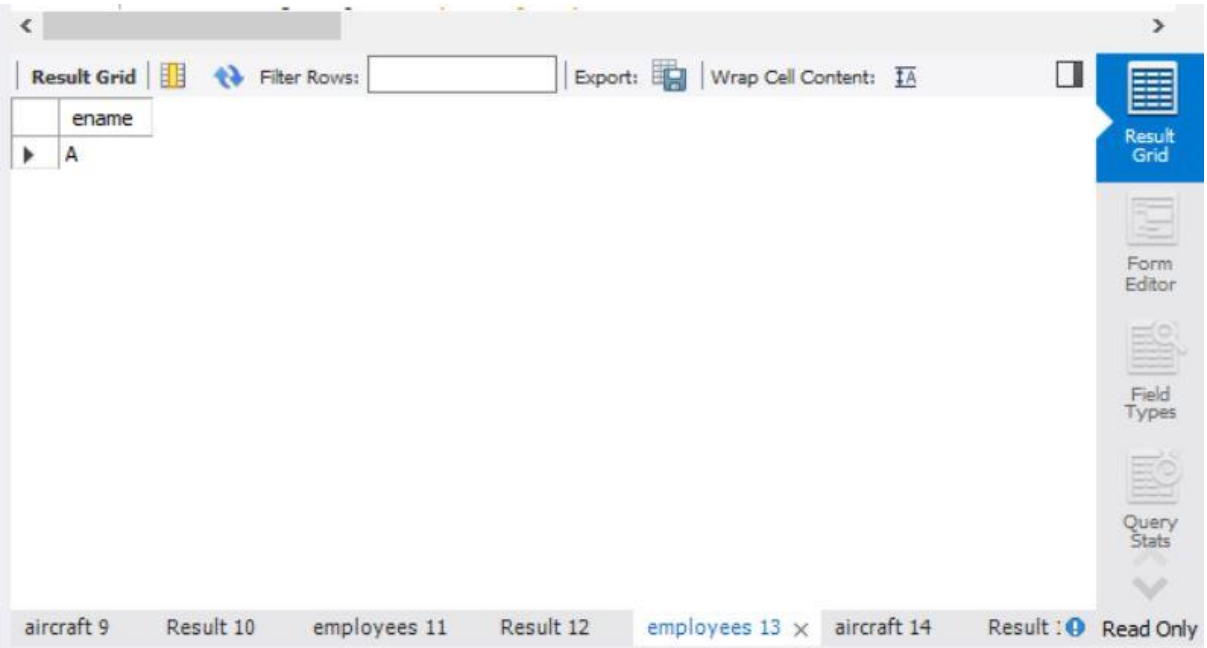
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg(e.salary)	aid
75000.0000	101
113333.3333	104
50000.0000	105
50000.0000	106
113333.3333	107

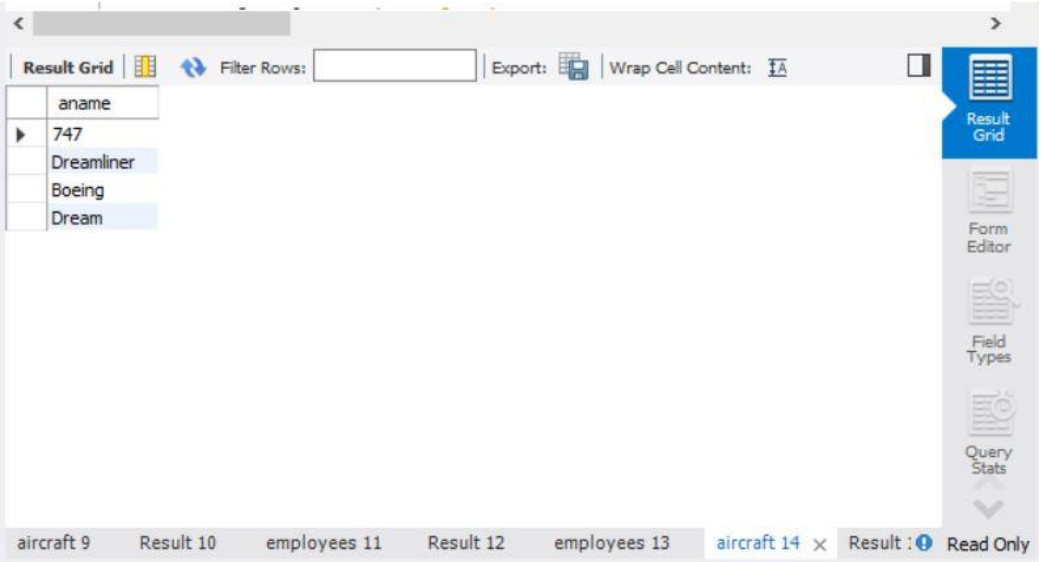
Result Grid | Form Editor | Field Types | Query Stats

aircraft 9 | Result 10 | employees 11 | Result 12 | employees 13 | aircraft 14 | Result 1 | Read Only

```
select ename from employees where eid in(
select eid from certified where aid in(
select aid from aircraft where aname = 'Boeing'));
```

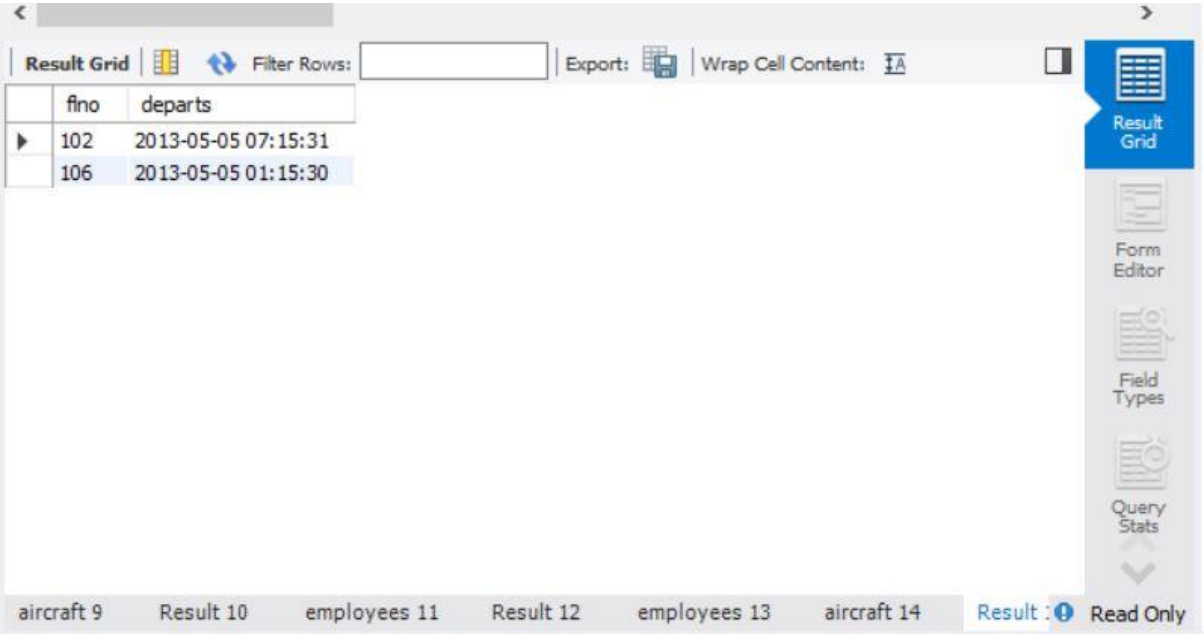


select aname from aircraft where cruisingrange > any (select distance from flights where fromplace='Bangalore' and toplace='Delhi');




```
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;
```

	Field	Type	Null	Key	Default	Extra
▶	salesm...	int	NO	PRI	NULL	
	name	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	
	commis...	varchar(20)	YES		NULL	

```
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;
```

	Field	Type	Null	Key	Default	Extra
▶	custom...	int	NO	PRI	NULL	
	cust_n...	varchar(20)	YES		NULL	
	city	varchar(20)	YES		NULL	
	grade	int	YES		NULL	
	salesm...	int	YES	MUL	NULL	

```
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;
```

Field	Type	Null	Key	Default	Extra
ord_no	int	NO	PRI	NULL	
purcha...	int	YES		NULL	
ord_date	date	YES		NULL	
custom...	int	YES	MUL	NULL	
salesm...	int	YES	MUL	NULL	

```
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;
```

salesman_id	name	city	commission
1000	john	bangalore	25 %
2000	ravi	bangalore	20 %
3000	kumar	mysore	15 %
4000	smith	delhi	30 %
5000	harsha	hydrabad	15 %
NULL	NULL	NULL	NULL

```
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;
```

customer_id	cust_name	city	grade	salesman_id
10	preethi	bangalore	100	1000
11	vivek	mangalore	300	1000
12	bhaskar	chennai	400	2000
13	chethan	bangalore	200	2000
14	mamatha	bangalore	400	3000
NULL	NULL	NULL	NULL	NULL

```
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;
```

ord_no	purchase_amt	ord_date	customer_id	salesman_id
50	5000	2004-06-17	10	1000
51	450	2020-01-17	10	2000
52	1000	2024-02-17	13	2000
53	3500	2013-04-17	14	3000
54	550	2009-03-17	12	2000
NULL	NULL	NULL	NULL	NULL

Additional Queries

```
select grade, count(distinct customer_id)
from customer
group by grade
having grade > (select avg(grade)
from customer
where city='bangalore'
);
```

grade	count(distinct customer_id)
300	1
400	2

```
select salesman_id, name
from salesman a
where 1 < (select count(*)
from customer1
where salesman_id=a.salesman_id
);
```

salesman_id	name
1000	john
2000	ravi
NULL	NULL

```
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;
```

salesman_id	name	cust_name	commission
4000	smith	no match	30 %
2000	ravi	preethi	20 %
2000	ravi	chethan	20 %
2000	ravi	mamatha	20 %
3000	kumar	no match	15 %
1000	john	preethi	25 %
1000	john	chethan	25 %
1000	john	mamatha	25 %
5000	harsha	no match	15 %

```
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;
```

ord_date	salesman_id	name
2004-06-17	1000	john
2020-01-17	2000	ravi
2024-02-17	2000	ravi
2013-04-17	3000	kumar
2009-03-17	2000	ravi

```
        delete from salesman
        where salesman_id=1000;
        select * from salesman;
        select * from orders;
```

salesman_id	name	city	commission
2000	ravi	bangalore	20 %
3000	kumar	mysore	15 %
4000	smith	delhi	30 %
5000	harsha	hydrabad	15 %
NULL	NULL	NULL	NULL

ord_no	purchase_amt	ord_date	customer_id	salesman_id
51	450	2020-01-17	10	2000
52	1000	2024-02-17	13	2000
53	3500	2013-04-17	14	3000
54	550	2009-03-17	12	2000
NULL	NULL	NULL	NULL	NULL

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;
```

	Field	Type	Null	Key	Default	Extra
▶	name	varchar(20)	NO	PRI	NULL	
	phone	int	YES		NULL	
	address	varchar(20)	YES		NULL	

```
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;
```

	Field	Type	Null	Key	Default	Extra
▶	book_id	int	NO	PRI	NULL	
	title	varchar(20)	YES		NULL	
	pub_year	varchar(20)	YES		NULL	
	publisher_name	varchar(20)	YES	MUL	NULL	

```
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;
```

	Field	Type	Null	Key	Default	Extra
▶	author_name	varchar(20)	NO	PRI	NULL	
	book_id	int	NO	PRI	NULL	

```
    create table library_branch (
        branch_id integer primary key,
        branch_name varchar (50),
        address varchar (50)
    );
    desc library_branch;
```

	Field	Type	Null	Key	Default	Extra
▶	branch_id	int	NO	PRI	NULL	
	branch_name	varchar(50)	YES		NULL	
	address	varchar(50)	YES		NULL	

```
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;
```

no_of_copies	book_id	branch_id
10	1	10
5	1	11
2	2	12
5	2	13
7	3	14
3	4	11
1	5	10
NULL	NULL	NULL

```
    create table card (
        card_no integer primary key
    );
    desc card;
```

	Field	Type	Null	Key	Default	Extra
▶	card_no	int	NO	PRI	NULL	

```
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;
```

	Field	Type	Null	Key	Default	Extra
▶	date_out	date	YES		NULL	
	due_date	date	YES		NULL	
	book_id	int	NO	PRI	NULL	
	branch_id	int	NO	PRI	NULL	
	card_no	int	NO	PRI	NULL	

```
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;
```

name	phone	address
grupo planeta	77561	bangalore
hachette livre	897086	chenai
mcgraw-hill	99890	bangalore
pearson	98890	newdelhi
random house	74556	hyderabad
NULL	NULL	NULL

```
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;
```

book_id	title	pub_year	publisher_name
1	dbms	01-2017	mcgraw-hill
2	adbms	06-2016	mcgraw-hill
3	cn	09-2016	pearson
4	cg	09-2015	grupo planeta
5	os	05-2016	pearson
NULL	NULL	NULL	NULL

```
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;
```

author_name	book_id
navathe	1
navathe	2
tanenbaum	3
edward angel	4
galvin	5
NULL	NULL

```
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;
```

branch_id	branch_name	address
10	rr nagar	bangalore
11	rnsit	bangalore
12	rajaji nagar	bangalore
13	nitte	mangalore
14	manipal	udupi
NULL	NULL	NULL

```
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;
```

	Field	Type	Null	Key	Default	Extra
▶	no_of_copies	int	YES		NULL	
	book_id	int	NO	PRI	NULL	
	branch_id	int	NO	PRI	NULL	

```
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;
```

	card_no
▶	100
	101
	102
	103
	104
*	NULL

```
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;

date_out	due_date	book_id	branch_id	card_no
2001-01-17	2001-06-17	1	10	101
2012-08-17	2012-08-17	1	11	104
2021-02-17	2021-04-17	2	13	101
2011-01-17	2011-03-17	3	14	101
2015-03-17	2015-07-17	4	11	101
NULL	NULL	NULL	NULL	NULL

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;

book_id	title	pub_year	publisher_name	no_of_copies	author_name	branch_name
1	dbms	01-2017	mcgraw-hill	10	navathe	rr nagar
1	dbms	01-2017	mcgraw-hill	5	navathe	rnsit
2	adbms	06-2016	mcgraw-hill	2	navathe	rajaji nagar
2	adbms	06-2016	mcgraw-hill	5	navathe	nitte
3	cn	09-2016	pearson	7	tanenbaum	manipal
4	cg	09-2015	grupo planeta	3	edward angel	rnsit
5	os	05-2016	pearson	1	galvin	rr nagar

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 ;

	card_no
▶	101

delete from book where book_id = 3;

select * from book;

select * from book_authors;

select * from book_copies;

select * from book_lending;

	book_id	title	pub_year	publisher_name
▶	1	dbms	01-2017	mcgraw-hill
	2	adbms	06-2016	mcgraw-hill
	4	cg	09-2015	grupo planeta
	5	os	05-2016	pearson
*	NULL	NULL	NULL	NULL

	author_name	book_id
▶	navathe	1
	navathe	2
	edward angel	4
	galvin	5
*	NULL	NULL

	no_of_copies	book_id	branch_id
▶	10	1	10
	5	1	11
	2	2	12
	5	2	13
	3	4	11
	1	5	10
*	NULL	NULL	NULL

	date_out	due_date	book_id	branch_id	card_no
▶	2001-01-17	2001-06-17	1	10	101
	2012-08-17	2012-08-17	1	11	104
	2021-02-17	2021-04-17	2	13	101
	2015-03-17	2015-07-17	4	11	101
*	NULL	NULL	NULL	NULL	NULL

```
create view partition select pub_year from book;
select * from partition;
```

	pub_year
▶	01-2017
	06-2016
	09-2015
	05-2016

```
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;
```

	book_id	title	no_of_copies
▶	1	dbms	10
	1	dbms	5
	2	adbms	2
	2	adbms	5
	4	cg	3
	5	os	1

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;
```

regno	name	major	bdate
1BM11CS001	A	Sr	1993-12-30
1BM11CS002	B	Sr	1993-09-24
1BM11CS003	C	Sr	1993-11-27
1BM11CS004	D	Sr	1993-04-13
1BM11CS005	E	Jr	1994-08-24
NULL	NULL	NULL	NULL

```
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;
```

courseno	cname	dept
111	OS	CSE
112	EC	ECE
113	SS	ISE
114	DBMS	CSE
115	SIGNALS	ECE
NULL	NULL	NULL

```
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;
```

book_isbn	book_title	publisher	author
10	DATABASE SYSTEMS	PEARSON	SCHIELD
900	OPERATING SYSTEMS	PEARSON	LELAND
901	CIRCUITS	HALL INDIA	BOB
902	SYSTEM SOFTWARE	PETERSON	JACOB
903	SCHEDULING	PEARSON	PATIL
904	DATABASE SYSTEMS	PEARSON	JACOB
905	DATABASE MANAGER	PEARSON	BOB
906	SIGNALS	HALL INDIA	SUMIT
NULL	NULL	NULL	NULL

```
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;
```

regno	courseno	sem	marks
1BM11CS001	115	3	100
1BM11CS002	114	5	100
1BM11CS003	113	5	100
1BM11CS004	111	5	100
1BM11CS005	112	3	100
NULL	NULL	NULL	NULL

```
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;
```

courseno	sem	book_isbn
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
114	5	905
115	3	906
NULL	NULL	NULL

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;
```

book_isbn	book_title	publisher	author
10	DATABASE SYSTEMS	PEARSON	SCHIELD
900	OPERATING SYSTEMS	PEARSON	LELAND
901	CIRCUITS	HALL INDIA	BOB
902	SYSTEM SOFTWARE	PETERSON	JACOB
903	SCHEDULING	PEARSON	PATIL
904	DATABASE SYSTEMS	PEARSON	JACOB
905	DATABASE MANAGER	PEARSON	BOB
906	SIGNALS	HALL INDIA	SUMIT
908	UNIX CONCEPTS	TATA MCGRAW HILL	SUMITABHA DAS
NULL	NULL	NULL	NULL

courseno	sem	book_isbn
111	5	900
111	5	903
111	5	904
112	3	901
113	3	10
113	5	902
113	4	908
114	5	905
115	3	906
NULL	NULL	NULL

```
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;
```

courseno	book_isbn	book_title
111	904	DATABASE SYSTEMS
111	900	OPERATING SYSTEMS
111	903	SCHEDULING

```
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');
```

	dept
▶	CSE

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;
```

act_id	act_name	act_gender
301	ANUSHKA	F
302	PRABHAS	M
303	PUNITH	M
304	JERMY	M
NULL	NULL	NULL

```
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;
```

dir_id	dir_name	dir_phone
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530
NULL	NULL	NULL

```
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;
```

mov_id	mov_title	mov_year	mov_lang	dir_id
1001	BAHUBALI-2	2017	TELAGU	60
1002	BAHUBALI-1	2015	TELAGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63
NULL	NULL	NULL	NULL	NULL

```
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;

act_id	mov_id	role
301	1001	HEROINE
301	1002	HEROINE
303	1002	GUEST
303	1003	HERO
304	1004	HERO
NULL	NULL	NULL

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;

mov_id	rev_stars
1001	4
1002	2
1003	5
1004	4
NULL	NULL

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;

	mov_title
▶	AKASH

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;

	mov_title
▶	BAHUBALI-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;

act_name	mov_title	mov_year
ANUSHKA	BAHUBALI-2	2017

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;

mov_title	max(rev_stars)
AKASH	5
BAHUBALI-1	2
BAHUBALI-2	4
WAR HORSE	4

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;

mov_id	rev_stars
1001	4
1002	2
1003	5
1004	5
NULL	NULL

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;

	Field	Type	Null	Key	Default	Extra
▶	usn	varchar(30)	NO	PRI	NULL	
	sname	varchar(30)	YES		NULL	
	address	varchar(30)	YES		NULL	
	phone	double	YES		NULL	
	gender	varchar(30)	YES		NULL	

create table semsec (

ssid varchar (5),

sem int,

sec char (1),

desc semsec;

	Field	Type	Null	Key	Default	Extra
▶	ssid	varchar(30)	NO	PRI	NULL	
	sem	int	YES		NULL	
	sec	varchar(30)	YES		NULL	

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;

	Field	Type	Null	Key	Default	Extra
▶	usn	varchar(30)	NO	PRI	NULL	
	ssid	varchar(30)	NO	PRI	NULL	

create table subject (

subcode varchar (8),

title varchar (20),

sem int,

credits int,

primary key (subcode));

desc subject;

	Field	Type	Null	Key	Default	Extra
▶	code	varchar(30)	NO	PRI	NULL	
	title	varchar(30)	YES		NULL	
	sem	int	YES		NULL	
	credits	int	YES		NULL	

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;

	Field	Type	Null	Key	Default	Extra
▶	usn	varchar(30)	NO	PRI	NULL	
	code	varchar(30)	NO	PRI	NULL	
	ssid	varchar(30)	NO	PRI	NULL	
	test1	double	YES		NULL	
	test2	double	YES		NULL	
	test3	double	YES		NULL	
	final	double	YES		NULL	

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;

usn	sname	address	phone	gender
1RN13CS020	akshay	belagavi	8877881122	m
1RN13CS062	sandhya	bengaluru	7722829912	f
1RN13CS066	supriya	mangaluru	8877881122	f
1RN13CS091	teesha	bengaluru	7712312312	f
1RN14CS010	abhay	bengaluru	9900211201	m
1RN14CS025	asmi	bengaluru	7894737377	f
1RN14CS032	bhaskar	bengaluru	9923211099	m
1RN15CS011	ajay	tumkur	98545091341	m
1RN15CS029	chitra	davangere	7696772121	f
1RN15CS045	jeeva	bellary	9944850121	m
1RN15CS091	santosh	mangaluru	8812332201	m
1RN16CS045	ismail	kalburgi	9900232201	m
1RN16CS088	sameera	shimoga	9905542212	f
1RN16CS122	vinayaka	chikamag...	8800880011	m
NULL	NULL	NULL	NULL	NULL

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;
```

ssid	sem	sec
CSE1A	1	A
CSE1B	1	B
CSE1C	1	C
CSE2A	2	A
CSE2B	2	B
CSE2C	2	C
CSE3A	3	A
CSE3B	3	B
CSE3C	3	C
CSE4A	4	A
CSE4B	4	B
CSE4C	4	C
CSE5A	5	A
CSE5B	5	B
CSE5C	5	C
CSE6A	6	A
CSE6B	6	B
CSE6C	6	C
CSE7A	7	A
CSE7B	7	B
CSE7C	7	C
CSE8A	8	A
CSE8B	8	B
CSE8C	8	C
NULL	NULL	NULL

```
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;

usn	ssid
1RN16CS045	CSE3A
1RN16CS088	CSE3B
1RN16CS122	CSE3C
1RN15CS011	CSE4A
1RN15CS029	CSE4A
1RN15CS045	CSE4B
1RN15CS091	CSE4C
1RN14CS010	CSE7A
1RN14CS025	CSE7A
1RN14CS032	CSE7A
1RN13CS020	CSE8A
1RN13CS062	CSE8A
1RN13CS066	CSE8B
1RN13CS091	CSE8C
NULL	NULL

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;

	code	title	sem	credits
▶	10CS31	M3	3	4
	10CS32	ADE	3	4
	10CS33	DSA	3	4
	10CS34	CO	3	4
	10CS35	USP	3	3
	10CS36	DMS	3	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
	10CS51	ME	5	4
	10CS52	CN	5	4
	10CS53	DBMS	5	4
	10CS54	ATC	5	4

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

usn	code	ssid	test1	test2	test3	final
1RN13CS091	10CS81	CSE8C	15	16	18	NULL
1RN13CS091	10CS82	CSE8C	12	19	14	NULL
1RN13CS091	10CS83	CSE8C	19	15	20	NULL
1RN13CS091	10CS84	CSE8C	20	16	19	NULL
1RN13CS091	10CS85	CSE8C	15	15	12	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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';
```

usn	sname	address	phone	gender	sem	sec
1rn15cs091	santosh	mangaluru	8812332201	m	4	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;
```

sem	sec	gender	count
3	a	m	1
3	b	f	1
3	c	m	1
4	a	f	1
4	a	m	1
4	b	m	1
4	c	m	1
7	a	f	1
7	a	m	2
8	a	f	1
8	a	m	1
8	b	f	1
8	c	f	1

```
create view stu_test1_marks_view
as
select test1, subcode
from iamarks
where usn = '1rn13cs091';

select * from stu_test1_marks_view;
```

test1	subcode
15	10cs81
12	10cs82
19	10cs83
20	10cs84
15	10cs85

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;
```

usn	sname	address	phone	gender	cat
1rn13cs091	teesha	bengaluru	7712312312	f	weak
1rn13cs091	teesha	bengaluru	7712312312	f	weak
1rn13cs091	teesha	bengaluru	7712312312	f	weak
1rn13cs091	teesha	bengaluru	7712312312	f	weak
1rn13cs091	teesha	bengaluru	7712312312	f	weak