Adnansami Madar
1BM20CS400
CSE-4A

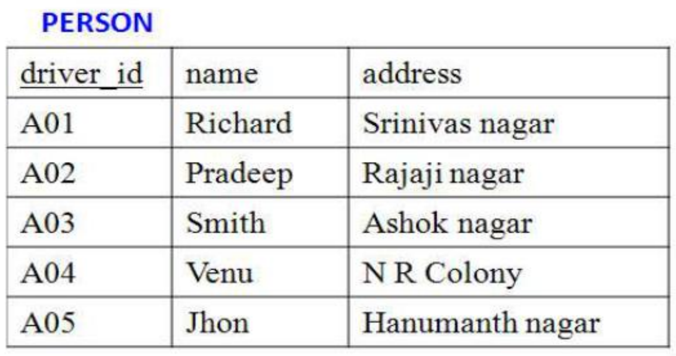
PROGRAM 1: INSURANCE DATABASE

Consider the Insurance database given below. The primary keys are underlined and the data types are specified.

PERSON (driver-id #: String, name: String, address: String)
CAR (Regno: String, model: String, year: int)
ACCIDENT (report-number: int, date: date, location: String)
OWNS (driver-id #: String, Regno: String)
PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

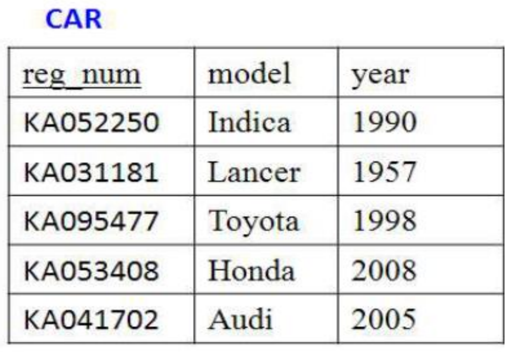
i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Demonstrate how you
a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.
b. Add a new accident to the 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

Tables



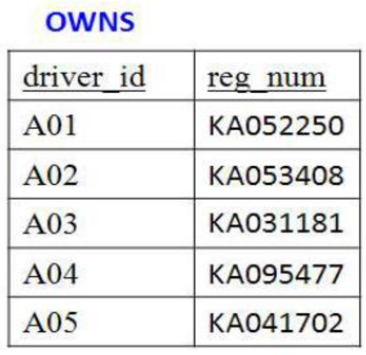
PERSON

driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	Jhon	Hanumanth nagar



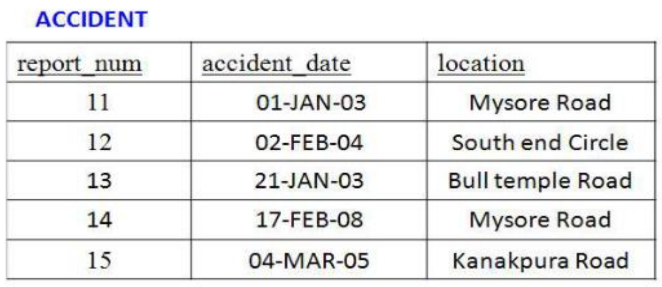
CAR

reg_num	model	year
KA052250	Indica	1990
KA031181	Lancer	1957
KA095477	Toyota	1998
KA053408	Honda	2008
KA041702	Audi	2005



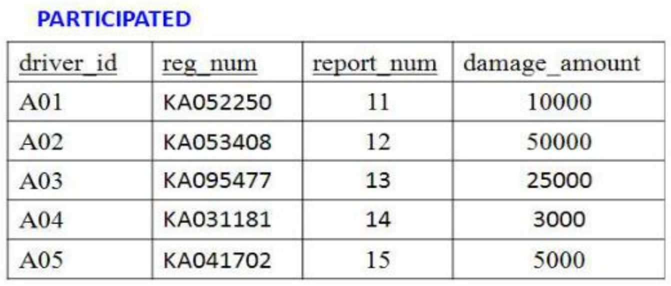
OWNS

driver_id	reg_num
A01	KA052250
A02	KA053408
A03	KA031181
A04	KA095477
A05	KA041702



ACCIDENT

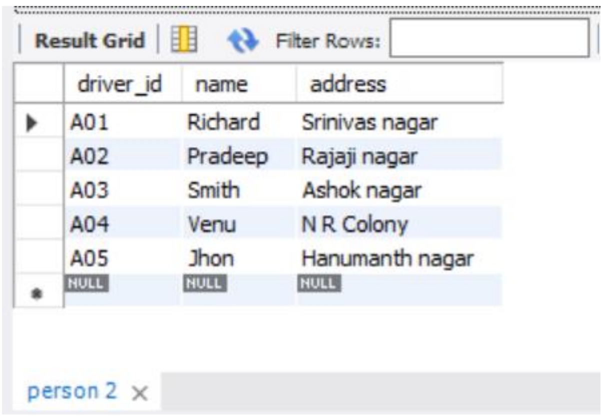
report_num	accident_date	location
11	01-JAN-03	Mysore Road
12	02-FEB-04	South end Circle
13	21-JAN-03	Bull temple Road
14	17-FEB-08	Mysore Road
15	04-MAR-05	Kanakpura Road



PARTICIPATED

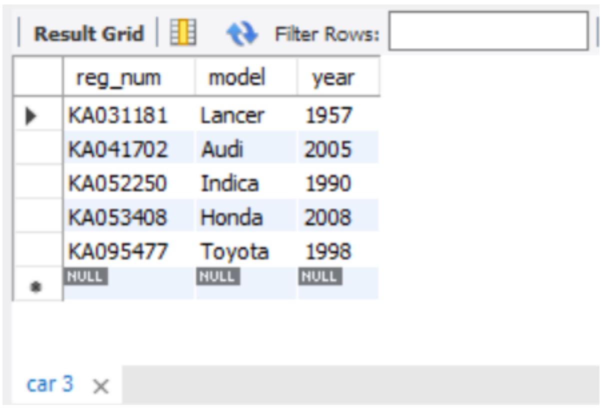
driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000

```
CREATE DATABASE INSURANCE_DATABASE;

USE INSURANCE_DATABASE;


create table person

(

   driver_ id varchar(15) unique NOT NULL,
   name varchar(20) NOT NULL,

   address varchar(30),

   primary key(driver_id)

);


create table car

(

    reg _num varchar(20) unique NOT NULL,

    model varchar(25),

    year int,

    primary key(reg_num)

);


create table accident

(

      report_num int unique NOT NULL,

     accident_date date,

     location varchar(30),

     primary key(report_num)

);


create table owns

(

       driver_id varchar(20),

       reg_num varchar(20),

       FOREIGN KEY(driver_id) REFERENCES person(driver_id),

       FOREIGN KEY(reg_num) REFERENCES car(reg_num)

);


create table participated

(

       driver_id varchar(15) unique NOT NULL,

      reg_num varchar(20) unique NOT NULL,
    report_num int unique NOT NULL,
    damage_amount int,
    FOREIGN KEY(driver_id) REFERENCES person(driver_id),
    FOREIGN KEY(reg_num) REFERENCES car(reg_num),
    FOREIGN KEY(report_num) REFERENCES accident(report_num)
);

insert into person
values ("A01","Richard","Srinivas nagar"),("A02","Pradeep","Rajaji nagar"),
("A03","Smith","Ashok nagar"),("A04","Venu","N R Colony"),("A05","Jhon","Hanumanth nagar");
select * from person;
```



driver_id	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	Jhon	Hanumanth nagar
NULL	NULL	NULL

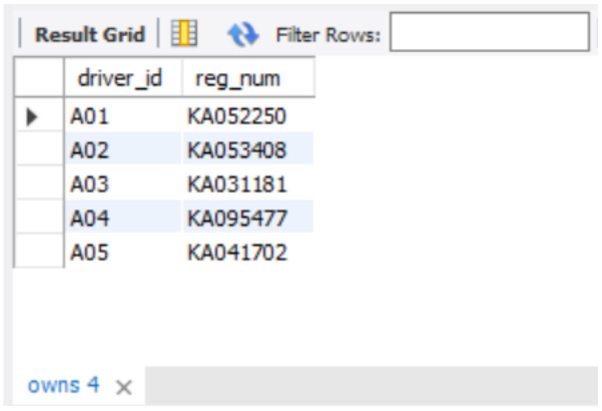
```
insert into car
values ("KA052250","Indica",1990),("KA031181","Lancer",1957),("KA095477","Toyota",1998),
("KA053408","Honda",2008),("KA041702","Audi",2005);
select * from car;
```



reg_num	model	year
KA031181	Lancer	1957
KA041702	Audi	2005
KA052250	Indica	1990
KA053408	Honda	2008
KA095477	Toyota	1998
NULL	NULL	NULL

```
insert into owns
values ("A01","KA052250"),("A02","KA053408"),("A03","KA031181"),
("A04","KA095477"),("A05","KA041702");
```

select * from owns;



Result Grid | Filter Rows:

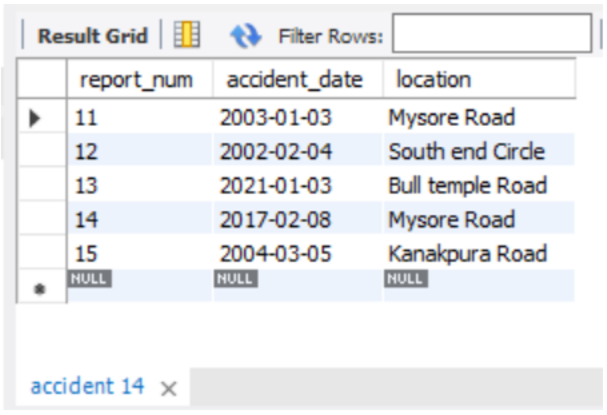
driver_id	reg_num
A01	KA052250
A02	KA053408
A03	KA031181
A04	KA095477
A05	KA041702

owns 4 ×

insert into accident

values (11,'2003-01-03',"Mysore Road") , (12,'2002-02-04',"South end Circle"),(13,'2021-01-03',"Bull temple Road"), (14,'2017-02-08',"Mysore Road"),(15,'2004-03-05',"Kanakpura Road");

select * from accident;



Result Grid | Filter Rows:

report_num	accident_date	location
11	2003-01-03	Mysore Road
12	2002-02-04	South end Circle
13	2021-01-03	Bull temple Road
14	2017-02-08	Mysore Road
15	2004-03-05	Kanakpura Road
NULL	NULL	NULL

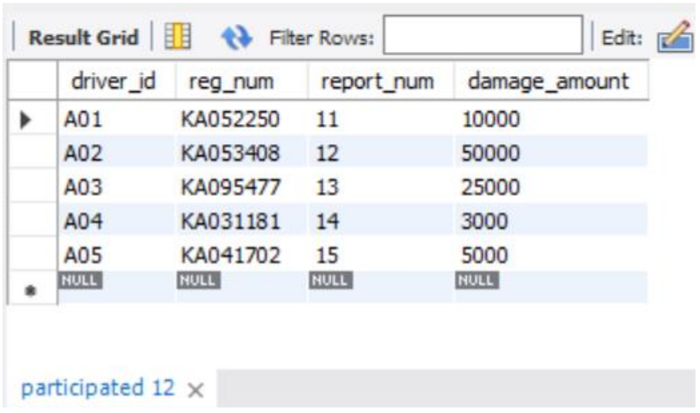
accident 14 ×

insert into participated

values ("A01","KA052250",11,10000),("A02","KA053408",12,50000),("A03","KA095477",13,25000),

("A04","KA031181",14,3000),("A05","KA041702",15,5000);

select * from participated;



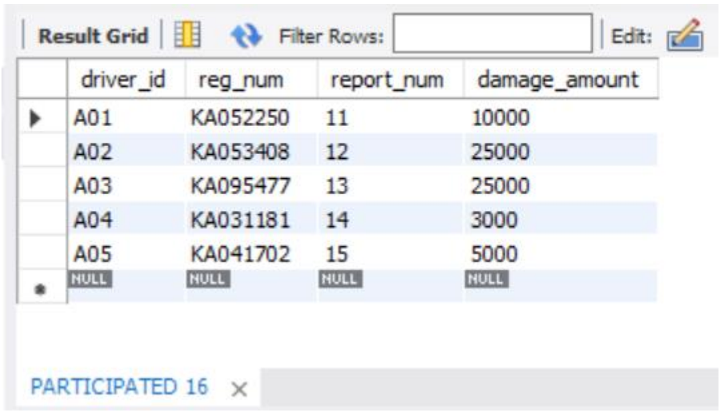
Result Grid | Filter Rows: | Edit:

driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

participated 12 ×

a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000

```
update PARTICIPATED

SET damage_amount=25000

WHERE reg_num="KA053408";

select* from PARTICIPATED;
```

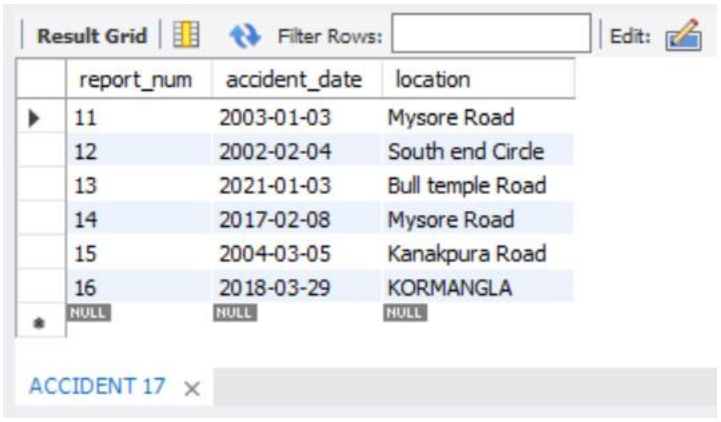


driver_id	reg_num	report_num	damage_amount
A01	KA052250	11	10000
A02	KA053408	12	25000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000
NULL	NULL	NULL	NULL

PARTICIPATED 16

b. Add a new accident to the database.

```
insert into ACCIDENT

values (16,"2018-03-29","KORMANGLA");

select* from ACCIDENT;
```

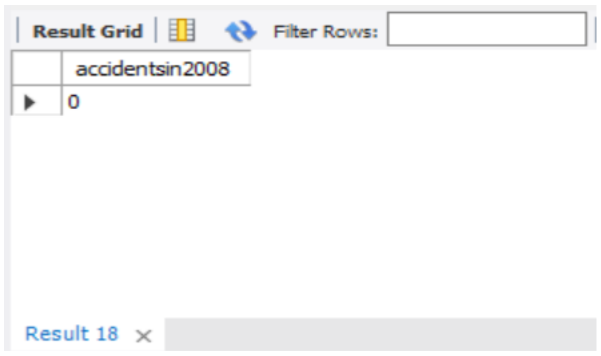


report_num	accident_date	location
11	2003-01-03	Mysore Road
12	2002-02-04	South end Circle
13	2021-01-03	Bull temple Road
14	2017-02-08	Mysore Road
15	2004-03-05	Kanakpura Road
16	2018-03-29	KORMANGLA
NULL	NULL	NULL

ACCIDENT 17

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

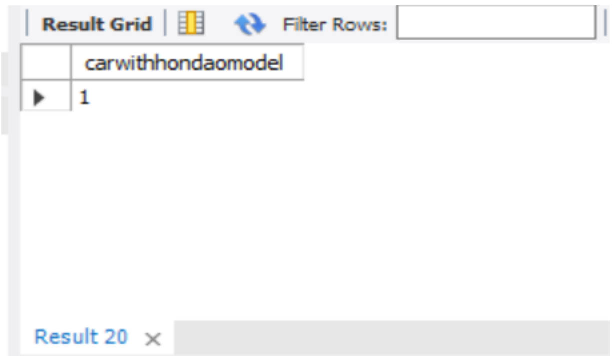
```
SELECT COUNT(accident_date) AS accidentsin2008

FROM ACCIDENT

WHERE YEAR(accident_date)=2008;
```



```
accidentsin2008
0
```

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

```
SELECT COUNT(model) AS carwithhondaomodel
FROM car
WHERE model="HONDA";
```



```
carwithhondaomodel
1
```

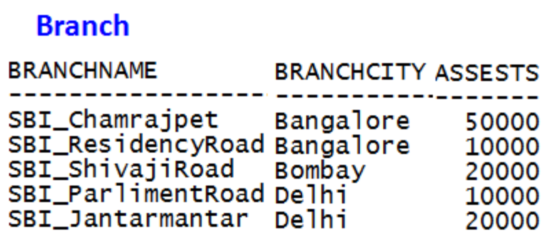
PROGRAM 2: BANKING ENTERPRISE DATABASE

Consider the following database for a banking enterprise.

Branch (branch-name: String, branch-city: String, assets: real)
BankAccount(accno: int, branch-name: String, balance: real)
BankCustomer (customer-name: String, customer-street: String, customer-city: String)
Depositer(customer-name: String, accno: int)
Loan (loan-number: int, branch-name: String, amount: real)

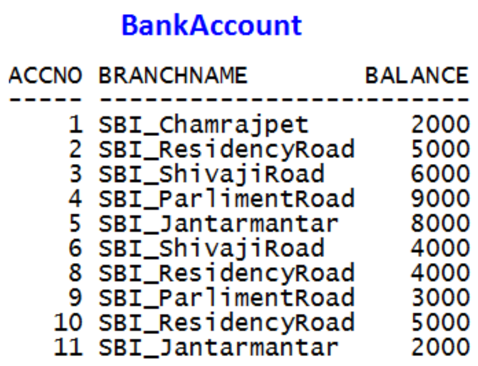
i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).
iv. Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).
v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).
INTRODUCTION: This database is developed for supporting banking facilities. Details of the branch along with the accounts and loans handled by them are recorded. Also details of the depositors of the corresponding branches are maintained.

Sample Table data



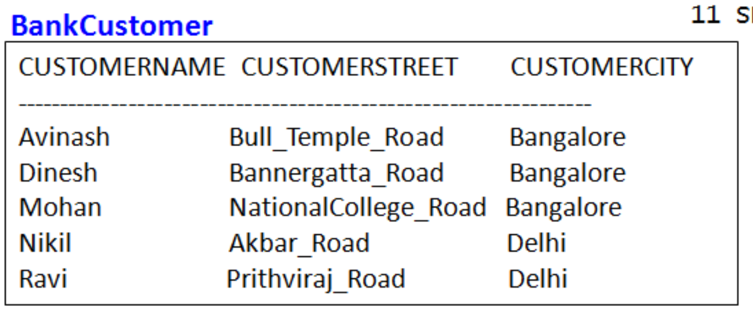
Branch

```
BRANCHNAME        BRANCHCITY ASSESTS
----------------- ---------- -------
SBI_Chamrajpet    Bangalore    50000
SBI_ResidencyRoad Bangalore    10000
SBI_ShivajiRoad   Bombay       20000
SBI_ParlimentRoad Delhi        10000
SBI_Jantarmantar  Delhi        20000
```



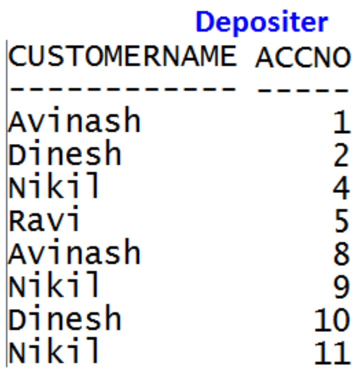
BankAccount

```
ACCNO BRANCHNAME                BALANCE
----- --------------------------------
    1 SBI_Chamrajpet               2000
    2 SBI_ResidencyRoad            5000
    3 SBI_ShivajiRoad              6000
    4 SBI_ParlimentRoad            9000
    5 SBI_Jantarmantar             8000
    6 SBI_ShivajiRoad              4000
    8 SBI_ResidencyRoad            4000
    9 SBI_ParlimentRoad            3000
   10 SBI_ResidencyRoad            5000
   11 SBI_Jantarmantar             2000
```



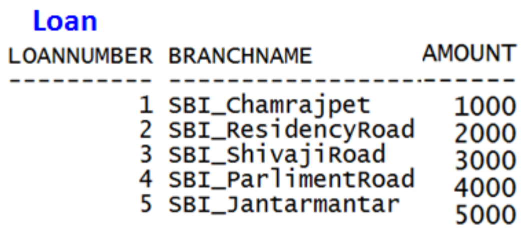
BankCustomer

CUSTOMERNAME	CUSTOMERSTREET	CUSTOMERCITY
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	NationalCollege_Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi



Depositer

```
CUSTOMERNAME ACCNO
------------ -----
Avinash          1
Dinesh           2
Nikil            4
Ravi             5
Avinash          8
Nikil            9
Dinesh          10
Nikil           11
```

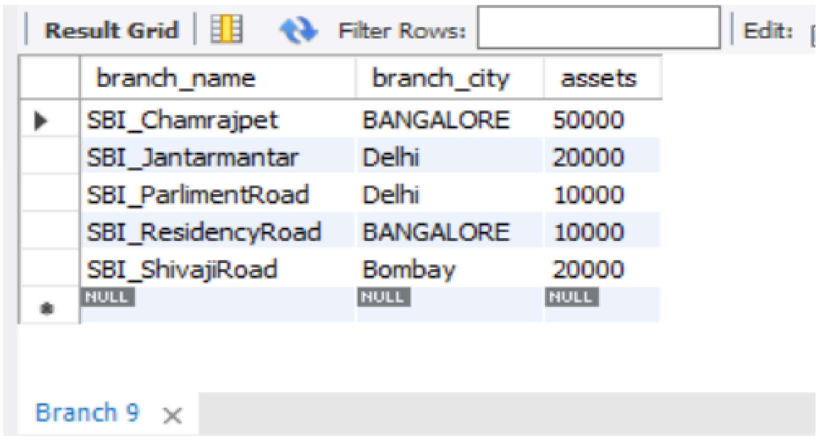


Loan

```
LOANNUMBER BRANCHNAME           AMOUNT
---------- ----------------------------
         1 SBI_Chamrajpet        1000
         2 SBI_ResidencyRoad     2000
         3 SBI_ShivajiRoad       3000
         4 SBI_ParlimentRoad     4000
         5 SBI_Jantarmantar      5000
```

create database Bankingenterprise;

```
use Bankingenterprise;

CREATE TABLE BRANCH
(
        branch_name varchar(50),
        branch_city varchar(50),
        assets real,
       primary key(branch_name)
);

CREATE TABLE BankAccount
(
             accno int,
            branch_name varchar(50),
            balance real,
           primary key(accno),
   FOREIGN KEY(branch_name) REFERENCES BRANCH(branch_name) ON DELETE SET NULL ON UPDATE
CASCADE
);

CREATE TABLE BankCustomer
(
         customer_name varchar(50) primary key,
        customer_street varchar(50),
       customer_city varchar (50)
);

CREATE TABLE Depositor
(
        customer_name varchar(50),
        accno int,
       foreign key(accno) references BankAccount(accno),
        foreign key(customer_name) references BankCustomer(customer_name)
);

CREATE TABLE Loan
(
         loan_number int,
         branch_name varchar(50),
         amount real,
        primary key(loan_number),
       foreign key(branch_name) references Branch(branch_name)
);


INSERT INTO Branch
VALUES ("SBI_Chamrajpet", "BANGALORE",50000),("SBI_ResidencyRoad", "BANGALORE",10000),
                  ("SBI_ShivajiRoad", "Bombay",20000),("SBI_ParlimentRoad", "Delhi",10000),
                  ("SBI_Jantarmantar", "Delhi",20000);
select * from Branch;
```

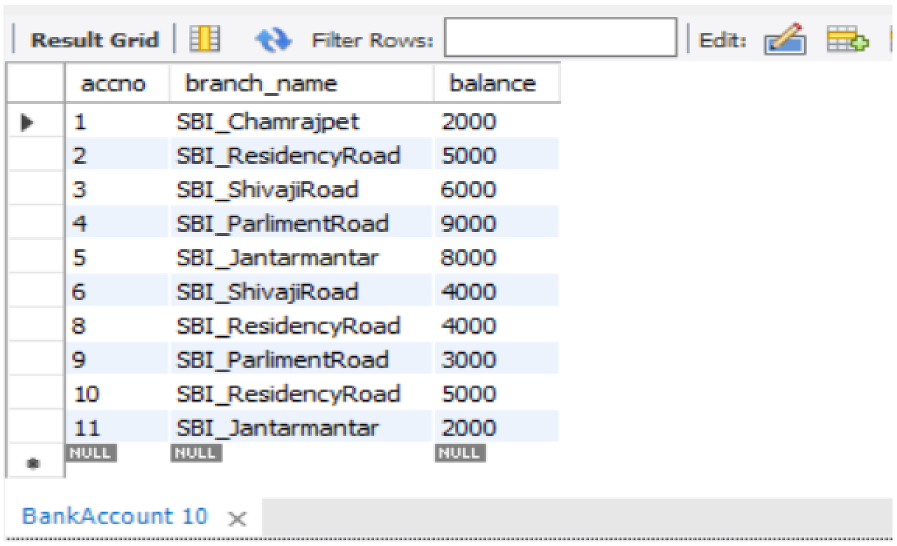


Result Grid | Filter Rows: | Edit:

branch_name	branch_city	assets
SBI_Chamrajpet	BANGALORE	50000
SBI_Jantarmantar	Delhi	20000
SBI_ParlimentRoad	Delhi	10000
SBI_ResidencyRoad	BANGALORE	10000
SBI_ShivajiRoad	Bombay	20000
NULL	NULL	NULL

Branch 9

```
INSERT INTO BankAccount
VALUES (1,"SBI_Chamrajpet",2000),(2,"SBI_ResidencyRoad",5000),
            (3,"SBI_ShivajiRoad",6000),(4,"SBI_ParlimentRoad",9000),(5,"SBI_Jantarmantar",8000),
    (6,"SBI_ShivajiRoad",4000),(8,"SBI_ResidencyRoad",4000),(9,"SBI_ParlimentRoad",3000),
    (10,"SBI_ResidencyRoad",5000),(11,"SBI_Jantarmantar",2000);
select * from BankAccount;
```


Result Grid | Filter Rows: | Edit:

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

BankAccount 10

INSERT INTO BankCustomer

VALUES ("Avinash","Bull_Temple_Road","Bangalore");

INSERT INTO BankCustomer

VALUES("Dinesh","Bannergatta_Road","Bangalore");

INSERT INTO BankCustomer

VALUES ("Mohan","NationtalCollege__Road","Bangalore");

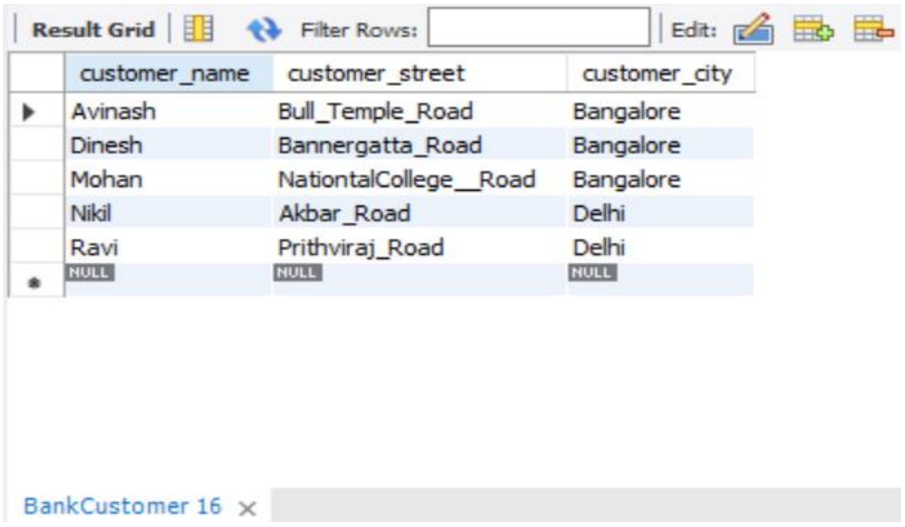
INSERT INTO BankCustomer

VALUES("Nikil","Akbar_Road","Delhi");

INSERT INTO BankCustomer

VALUES ("Ravi","Prithviraj_Road","Delhi");

select * from BankCustomer;


Result Grid | Filter Rows: | Edit:

customer_name	customer_street	customer_city
Avinash	Bull_Temple_Road	Bangalore
Dinesh	Bannergatta_Road	Bangalore
Mohan	NationtalCollege__Road	Bangalore
Nikil	Akbar_Road	Delhi
Ravi	Prithviraj_Road	Delhi
NULL	NULL	NULL

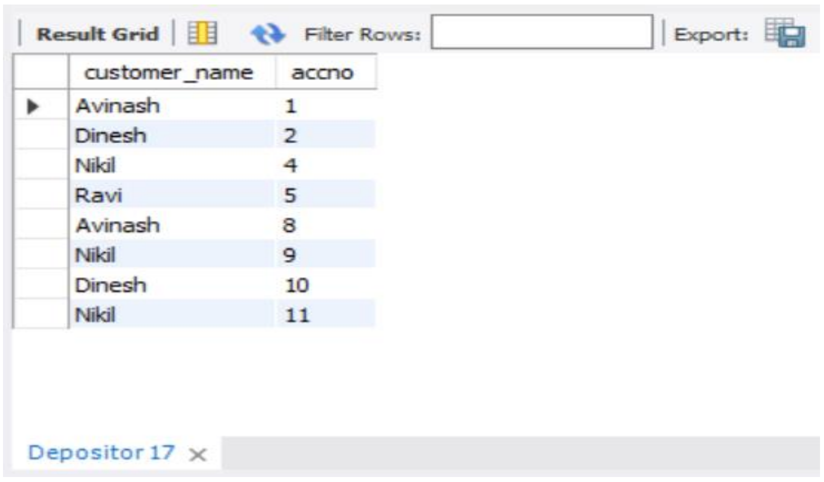
BankCustomer 16

INSERT INTO Depositor

VALUES("Avinash",1),("Dinesh",2),("Nikil",4),

("Ravi",5),("Avinash",8),("Nikil",9),("Dinesh",10),("Nikil",11);

select * from Depositor;



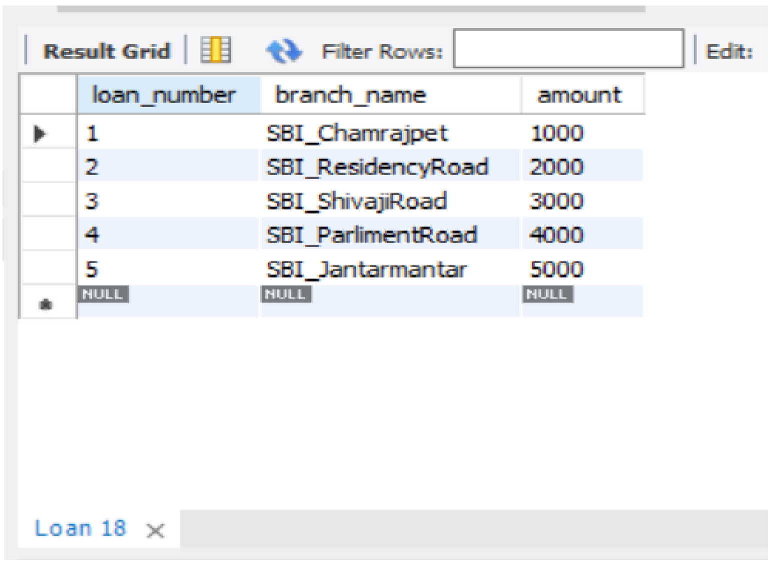
customer_name	accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11

INSERT INTO Loan

VALUES (1,"SBI_Chamrajpet",1000),(2,"SBI_ResidencyRoad",2000),(3,"SBI_ShivajiRoad",3000),

(4,"SBI_ParlimentRoad",4000),(5,"SBI_Jantarmantar",5000);

select * from Loan;



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

iii. Find all the customers who have at least two accounts at the *Main* branch (ex. SBI_ResidencyRoad).

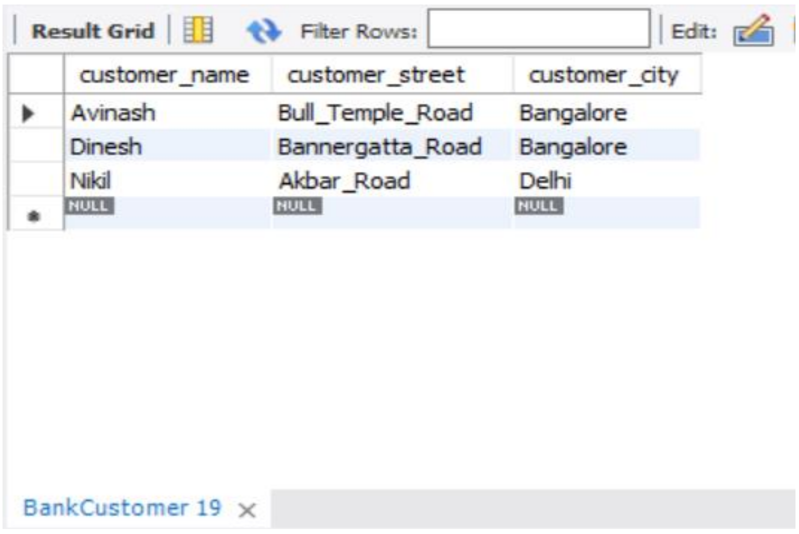
SELECT *

FROM BankCustomer

WHERE customer_name IN (SELECT customer_name

FROM depositor

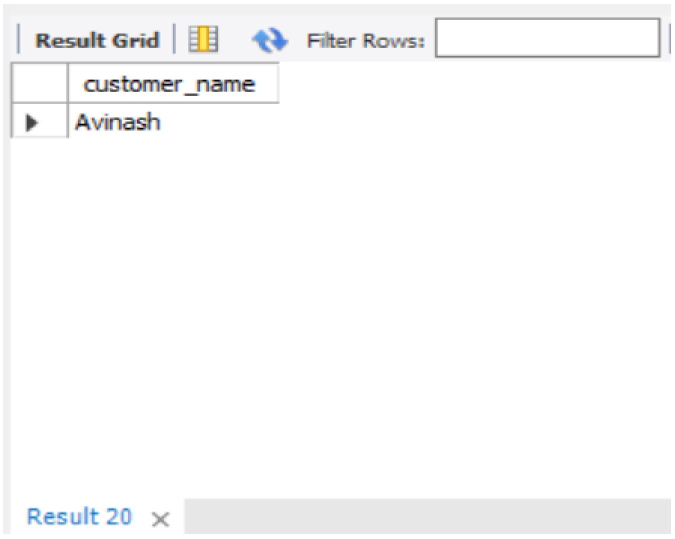
```
                    group by customer_name

                    having COUNT(customer_name)>=2);
```



	customer_name	customer_street	customer_city
▶	Avinash	Bull_Temple_Road	Bangalore
	Dinesh	Bannergatta_Road	Bangalore
	Nikil	Akbar_Road	Delhi
*	NULL	NULL	NULL

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

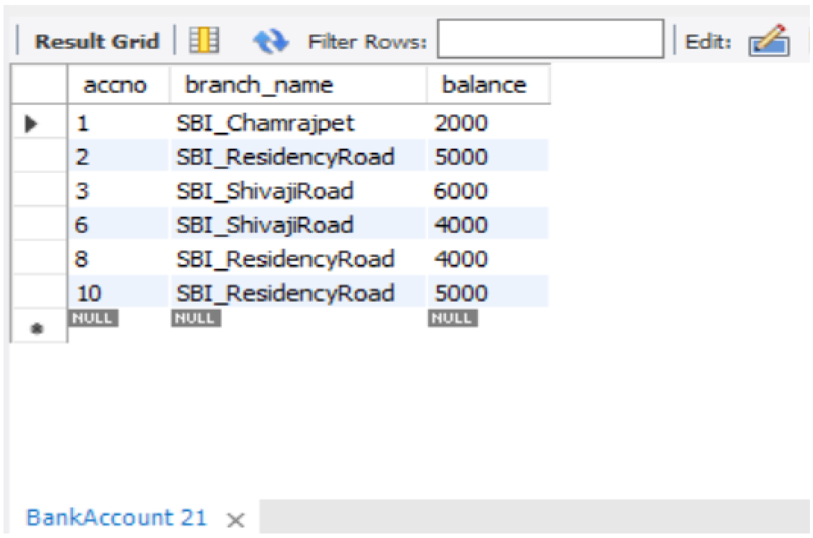
```
SELECT d.customer_name

FROM BankAccount a, Depositor d, Branch b

WHERE d.accno=a.accno AND b.branch_name=a.branch_name AND b.branch_city="Bangalore"

GROUP BY d.customer_name

HAVING count(distinct b.branch_name)=

                                    (SELECT COUNT(branch_name)

      FROM branch

      WHERE branch_city="Bangalore");
```



	customer_name
▶	Avinash

v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. Bombay).

```
DELETE FROM depositor

WHERE accno IN

(SELECT accno

FROM Branch b, BankAccount a

WHERE branch_city = 'delhi' and b.branch_name = a.branch_name);

DELETE FROM BankAccount WHERE branch_name IN(SELECT branch_name FROM BRANCH WHERE branch_city='delhi');


SELECT * FROM BankAccount;
```



accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
3	SBI_ShivajiRoad	6000
6	SBI_ShivajiRoad	4000
8	SBI_ResidencyRoad	4000
10	SBI_ResidencyRoad	5000
NULL	NULL	NULL

PROGRAM 3: SUPPLIER DATABASE

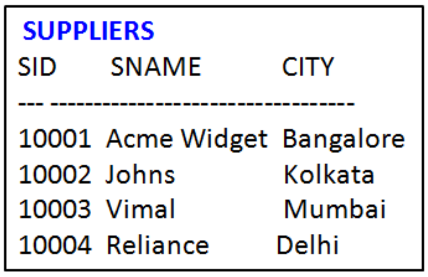
Consider the following schema:

SUPPLIERS(sid: integer, sname: string, address: string)
PARTS(pid: integer, pname: string, color: string)
CATALOG(sid: integer, pid: integer, cost: real)
The Catalog relation lists the prices charged for parts by Suppliers.

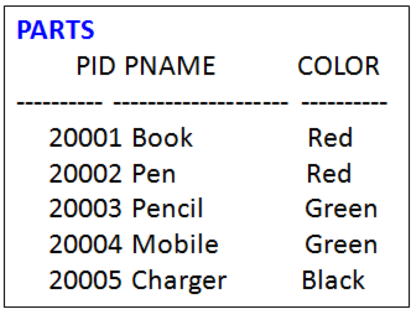
Write the following queries in SQL:

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.

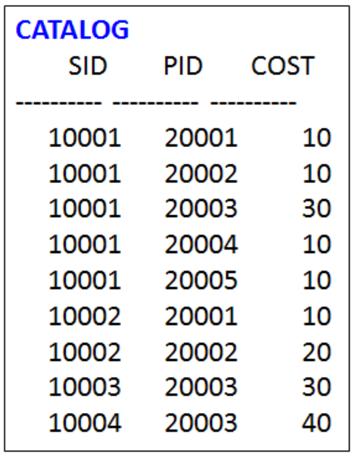
Table Data



```
SUPPLIERS
SID     SNAME          CITY
--- ----------------------------------
10001  Acme Widget  Bangalore
10002  Johns           Kolkata
10003  Vimal           Mumbai
10004  Reliance       Delhi
```



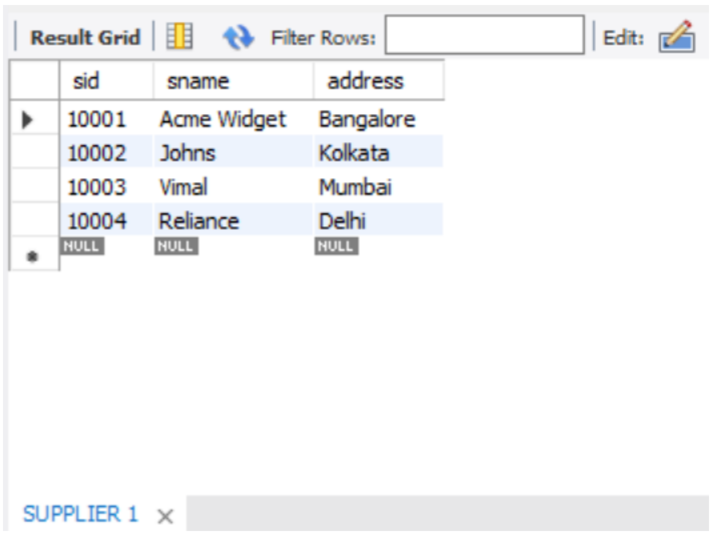
```
PARTS
      PID PNAME          COLOR
---------- ------------------- ----------
   20001 Book              Red
   20002 Pen               Red
   20003 Pencil            Green
   20004 Mobile            Green
   20005 Charger           Black
```



```
CATALOG
     SID        PID       COST
---------- ---------- ----------
   10001     20001        10
   10001     20002        10
   10001     20003        30
   10001     20004        10
   10001     20005        10
   10002     20001        10
   10002     20002        20
   10003     20003        30
   10004     20003        40
```

```
CREATE DATABASE Supplier_Database1;

USE Supplier_Database1;
```

```
CREATE TABLE SUPPLIER
(
	sid int,
	sname varchar(20),
	address varchar(40),
	primary key(sid)
);

CREATE TABLE PARTS
(
	pid int,
	pname varchar(40),
	color varchar(20),
	primary key(pid)
);

CREATE TABLE CATALOG
(
	sid int,
	pid int,
	cost real,
	FOREIGN KEY(sid) REFERENCES SUPPLIER(sid),
	FOREIGN KEY(pid) REFERENCES PARTS(pid)
);

INSERT INTO SUPPLIER
VALUES (10001,"Acme Widget","Bangalore"),(10002,"Johns","Kolkata"),(10003,"Vimal","Mumbai"),
			(10004,"Reliance","Delhi");
 SELECT * FROM SUPPLIER;
```

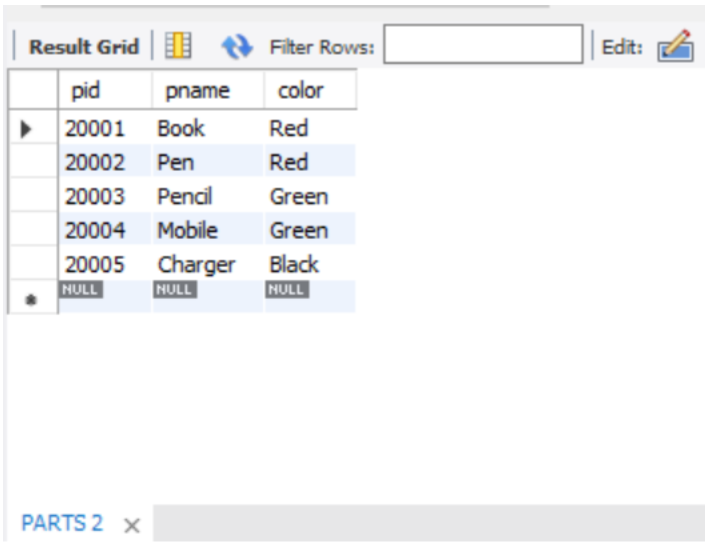


sid	sname	address
10001	Acme Widget	Bangalore
10002	Johns	Kolkata
10003	Vimal	Mumbai
10004	Reliance	Delhi
NULL	NULL	NULL

INSERT INTO PARTS

VALUES (20001,"Book","Red"), (20002,"Pen","Red"), (20003,"Pencil","Green"), (20004,"Mobile","Green"), (20005,"Charger","Black");

SELECT * FROM PARTS;



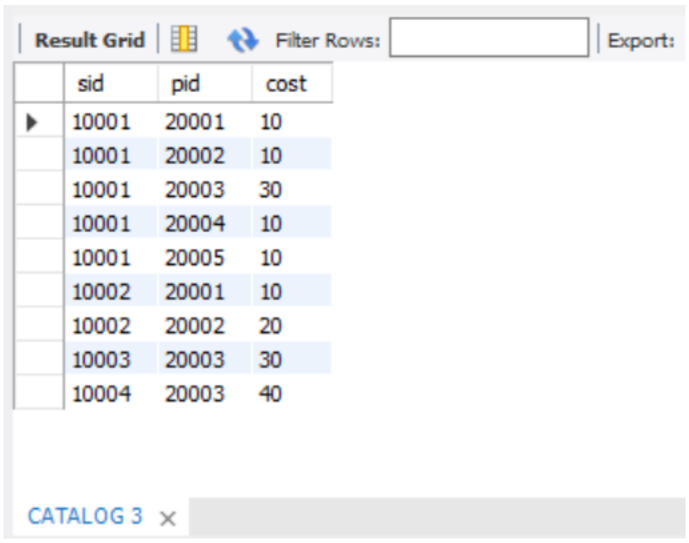
pid	pname	color
20001	Book	Red
20002	Pen	Red
20003	Pencil	Green
20004	Mobile	Green
20005	Charger	Black
NULL	NULL	NULL

INSERT INTO CATALOG

VALUES (10001,20001,10),(10001,20002,10),(10001,20003,30),(10001,20004,10),(10001,20005,10),

(10002,20001,10),(10002,20002,20),(10003,20003,30),(10004,20003,40);

SELECT * FROM CATALOG;

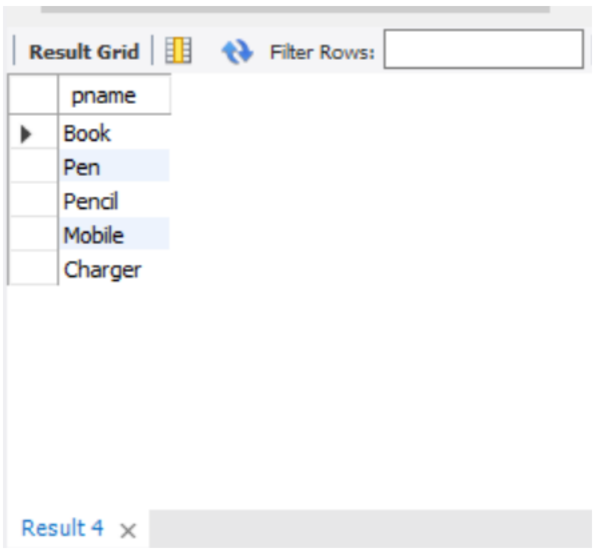

Result Grid | Filter Rows: | Export:

sid	pid	cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40

CATALOG 3

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

```
SELECT DISTINCT p.pname

FROM parts p, catalog c

WHERE p.pid = c.pid;
```

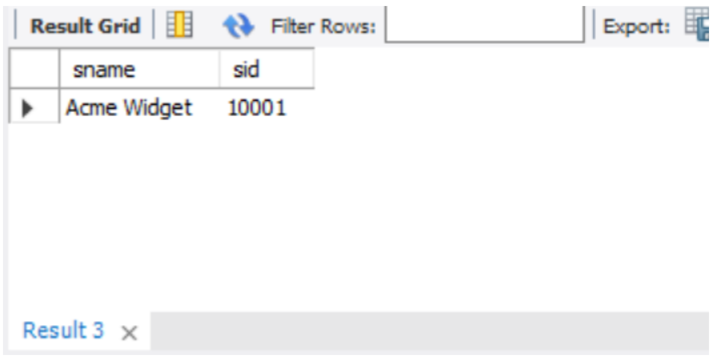

Result Grid | Filter Rows:

pname
Book
Pen
Pencil
Mobile
Charger

Result 4

b.Find the snames of suppliers who supply every part.

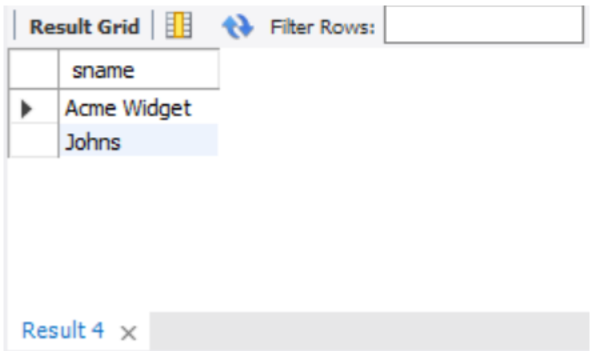
```
SELECT SUPPLIER.sname, CATALOG.sid
FROM SUPPLIER, CATALOG
WHERE SUPPLIER.sid = CATALOG.sid
group by SUPPLIER.sname
having count(CATALOG.sid)=(select count(pid)
                        from PARTS);
```



	sname	sid
▶	Acme Widget	10001

Result 3

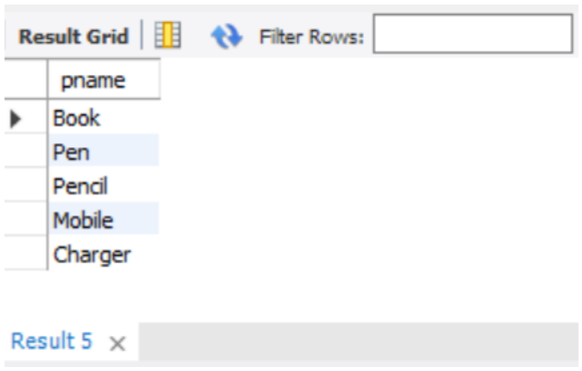
c.Find the snames of suppliers who supply every red part.
SELECT distinct s.sname
FROM SUPPLIER s, CATALOG c
WHERE s.sid = c.sid and c.pid in (select pid from PARTS WHERE color="Red");



	sname
▶	Acme Widget
	Johns

Result 4

d. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
SELECT pname
FROM PARTS P , CATALOG C
WHERE p.pid = c.pid AND sid in (SELECT sid from SUPPLIER WHERE sname = "Acme Widget");

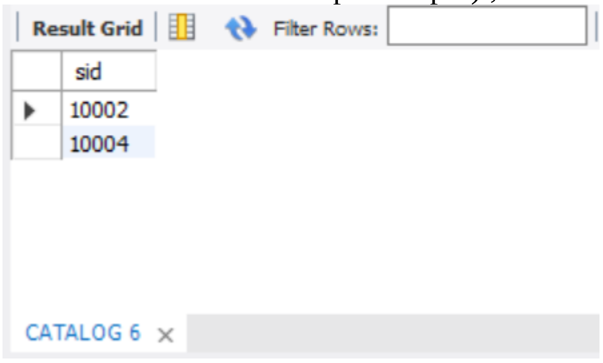


	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

Result 5

e. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
SELECT c.sid

```
FROM CATALOG c WHERE c.cost > (SELECT AVG(cost)
                               FROM CATALOG
                               WHERE pid = c.pid) ;
```



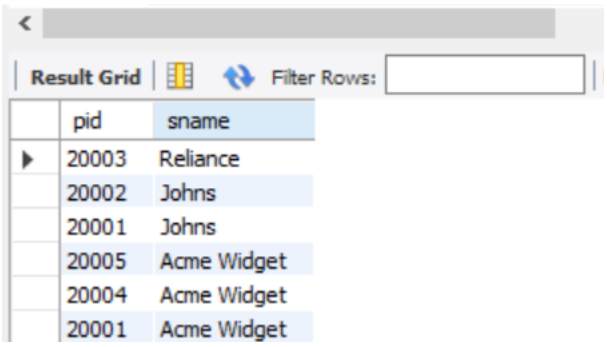
Result Grid | Filter Rows:

sid
10002
10004

CATALOG 6 ×

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

```
SELECT c.pid, s.sname

FROM PARTS p, SUPPLIER s, CATALOG c

WHERE s.sid = c.sid AND c.cost = (SELECT MAX(cost)

                                  FROM catalog

                                 WHERE pid = c.pid);
```



Result Grid | Filter Rows:

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

PROGRAM 4: STUDENT FACULTY DATABASE

Consider the following database for
student enrollment for course :

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

CLASS(cname: string, meetsat: time, room: string, fid: integer)

ENROLLED(snum: integer, cname:string)

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

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

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

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 enrollment 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).

```
SQL> select * from student;

      SNUM    SNAME          MA    LV    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
```

SQL> select * from faculty;

```
       FID FNAME                DEPTID
---------- -------------------- ----------
        11 Harish                 1000
        12 MV                     1000
        13 Mira                   1001
        14 Shiva                  1002
        15 Nupur                  1000
```

SQL> select * from class;

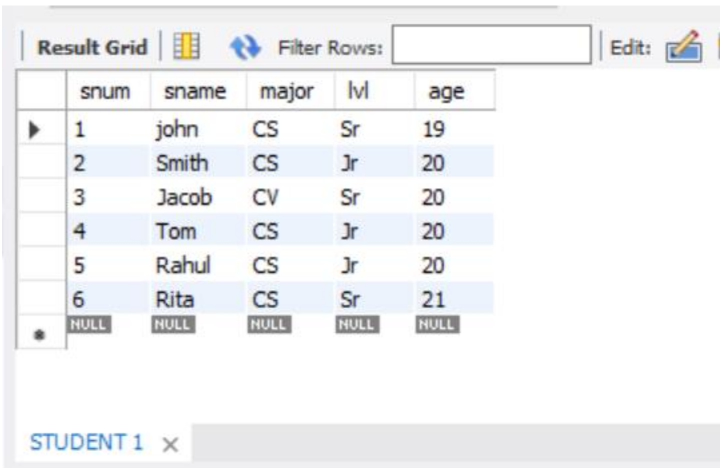
```
CNAME     METTS_A                      ROOM          FID
Class1     12/11/15 10:15:16.00000      R1             14
Class10    12/11/15 10:15:16.00000      R128           14
Class2     12/11/15 10:15:20.000000    R2             12
Class3     12/11/15 10:15:25.000000   R3             11
Class4     12/11/15 20:15:20.000000   R4             14
Class5     12/11/15 20:15:20.000000    R3             15
Class6    12/11/15 13:20:20.000000    R2             14
Class7    12/11/15 10:10:10.000000    R3             14
```

SQL> select * from enrolled;

```
      SNUM CNAME
---------- --------------------
         1 class1
         2 class1
         3 class3
         4 class3
         5 class4
```

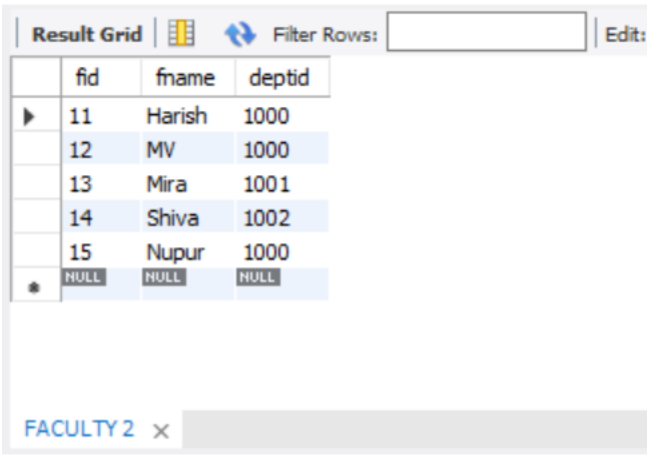
```
CREATE DATABASE STUDENT_FACULTY1;

USE STUDENT_FACULTY1;
```

```
CREATE TABLE STUDENT
(
            snum int,
            sname varchar(40),
            major varchar(30),
            lvl varchar(20),
            age int,
            primary key(snum)
);

CREATE TABLE FACULTY
(
       fid int,
       fname varchar(40),
       deptid int,
       primary key(fid)
);

CREATE TABLE CLASS
(
            cname varchar(40),
            meetsat datetime;
            room varchar(20),
            fid int,
            primary key(cname),
           FOREIGN KEY(fid) REFERENCES FACULTY(fid)
);

CREATE TABLE ENROLLED
(
            snum int,
        cname varchar(40),
        FOREIGN KEY(snum) REFERENCES STUDENT(snum),
        FOREIGN KEY(cname) REFERENCES CLASS(cname)
);
INSERT INTO STUDENT
VALUES (1,"john","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);
SELECT * FROM STUDENT;
```



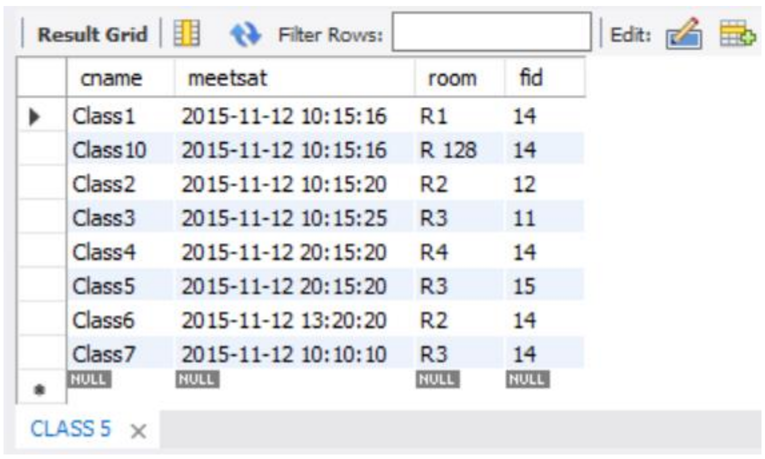
snum	sname	major	lvl	age
1	john	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Rita	CS	Sr	21
NULL	NULL	NULL	NULL	NULL

```
INSERT INTO FACULTY
VALUES (11,"Harish",1000),(12,"MV",1000),(13,"Mira",1001),(14,"Shiva",1002),
        (15,"Nupur",1000);
SELECT * FROM FACULTY;
```



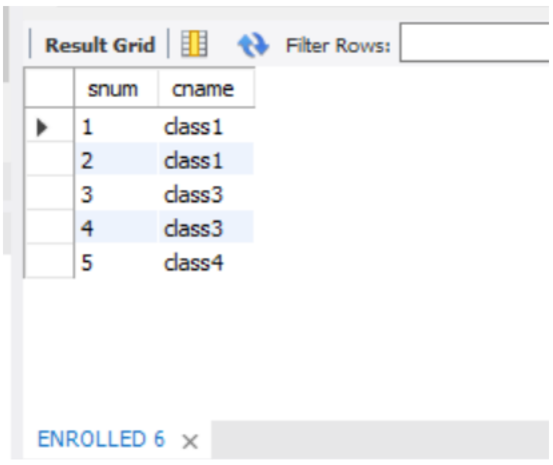
fid	fname	deptid
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000
NULL	NULL	NULL

```
INSERT INTO CLASS

VALUES ("Class1",'2015-11-12:10:15:16.00000',"R1",14);

              ("Class10",'2015-11-12:10:15:16.00000',"R 128",14),

             ("Class2",'2015-11-12:10:15:20.00000',"R2",12),

             ("Class3",'2015-11-12:10:15:25.00000',"R3",11),

            ("Class4",'2015-11-12:20:15:20.00000',"R4",14),

            ("Class5",'2015-11-12:20:15:20.00000',"R3",15),

            ("Class6",'2015-11-12:13:20:20.00000',"R2",14),

            ("Class7",'2015-11-12:10:10:10.00000',"R3",14);

select * from CLASS;
```



cname	meetsat	room	fid
Class1	2015-11-12 10:15:16	R1	14
Class10	2015-11-12 10:15:16	R 128	14
Class2	2015-11-12 10:15:20	R2	12
Class3	2015-11-12 10:15:25	R3	11
Class4	2015-11-12 20:15:20	R4	14
Class5	2015-11-12 20:15:20	R3	15
Class6	2015-11-12 13:20:20	R2	14
Class7	2015-11-12 10:10:10	R3	14
NULL	NULL	NULL	NULL

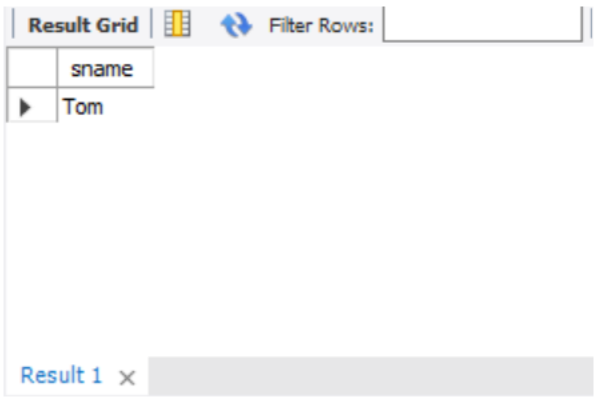
```
INSERT INTO ENROLLED

VALUES (1,"class1"),(2,"class1"),(3,"class3"),(4,"class3"),(5,"class4");

select * from ENROLLED;
```



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

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

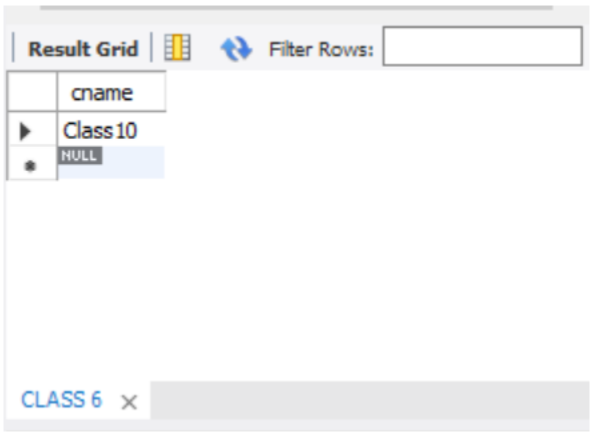
```
SELECT s.sname

FROM STUDENT s, CLASS c, ENROLLED e

WHERE s.snum=e.snum and  c.cname=e.cname and c.fid = (select fid

                                                  from FACULTY

                                                  where fname="Harish") and S.lvl="Jr";
```



	sname
▶	Tom

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

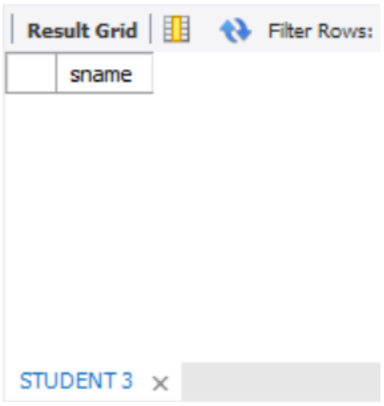
```
SELECT c.cname

FROM CLASS c

WHERE c.room = 'R 128' or c.cname in (select e.cname

                                      from ENROLLED e

                                      group by e.cname
having count(*)>=5);
```



```
cname
Class10
NULL
```

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

```
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.meetsat = c2.meetsat);
```

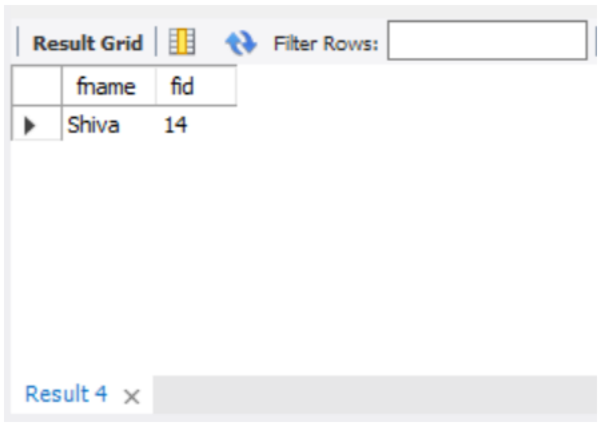


```
sname
```

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

```
SELECT f.fname, c.fid

FROM FACULTY f, CLASS c

WHERE f.fid = c.fid
```

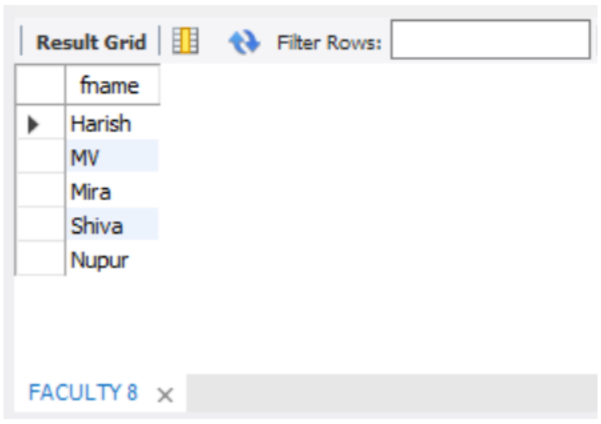
```
group by c.fid
having count(c.fid)=(select COUNT(distinct room)
                     from CLASS);
```



```
fname   fid
Shiva   14
```

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

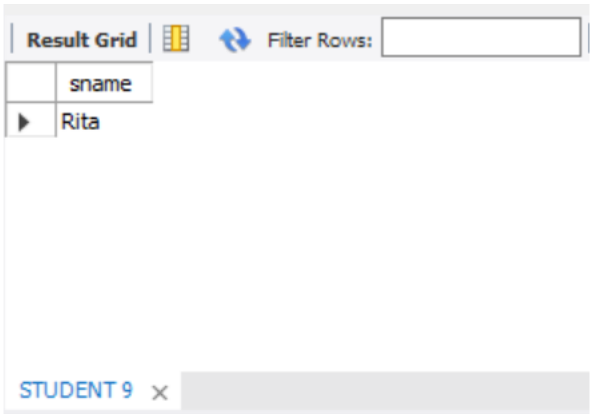
```
SELECT distinct fname
FROM FACULTY f
WHERE 5> (select COUNT(e.snum)
            from ENROLLED e,CLASS c
            where c.cname=e.cname and c.fid=f.fid);
```



```
fname
Harish
MV
Mira
Shiva
Nupur
```

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

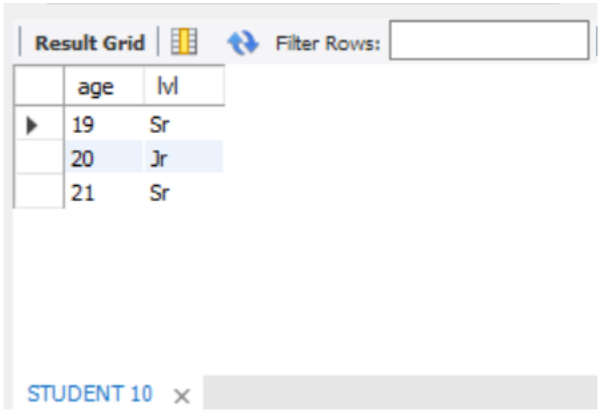
```
SELECT s.sname
FROM STUDENT s
WHERE snum not in (select snum
                   from ENROLLED);
```



sname
Rita

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).

```
SELECT  s.age ,s.lvl
 FROM STUDENT s
 group by s.age
 having s.lvl in (select s1.lvl
                     from STUDENT s1
                    where s1.age=s.age
                    group by s1.age
                  having count(*)>= all (select s2.lvl
                                          from STUDENT s2
                                         where s2.age=s1.age
                                         group by s2.age));
```



age	lvl
19	Sr
20	Jr
21	Sr

PROGRAM 5: AIRLINE FLIGHT DATABASE

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

FLIGHTS(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)
AIRCRAFT(aid: integer, aname: string, cruisingrange: integer)
CERTIFIED(eid: integer, aid: integer)
EMPLOYEES(eid: integer, ename: string, salary: integer)
Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write each of the following queries in SQL.

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 cruisingrange 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 cruisingrange 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 Bangalore to Kolkata New with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in Kolkata by 6 p.m.

```
SQL> select * from Flights;
 FLNO FFROM          TO             DISTANCE    DEPARTS    ARRIVES     PRICE
 101 Bangalore  Delhi  2500 13-MAY-05 07.15.31.000000 AM13-MAY-05 07.15.31.000000 A   5000
102 Bangalore  Lucknow              3000  05/05/13 07:15:31.000000 05/05/13 11:15:31.000000  6000
103 Lucknow        Delhi                500 05/05/13 12:15:31.000000  05/05/13 17:15:31.000000  3000
107 Bangalore      Frankfurt           8000  05/05/13 07:15:31.000000  05/05/13 22:15:31.000000  60000
104 Bangalore      Frankfurt           8500  05/05/13 07:15:31.000000    05/05/13 23:15:31.00000   75000
105 Kolkata        Delhi               3400   05/05/13 07:15:31.000000   05/05/13 09:15:31.000000  7000
```

```
SQL> 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

7 rows selected.
```

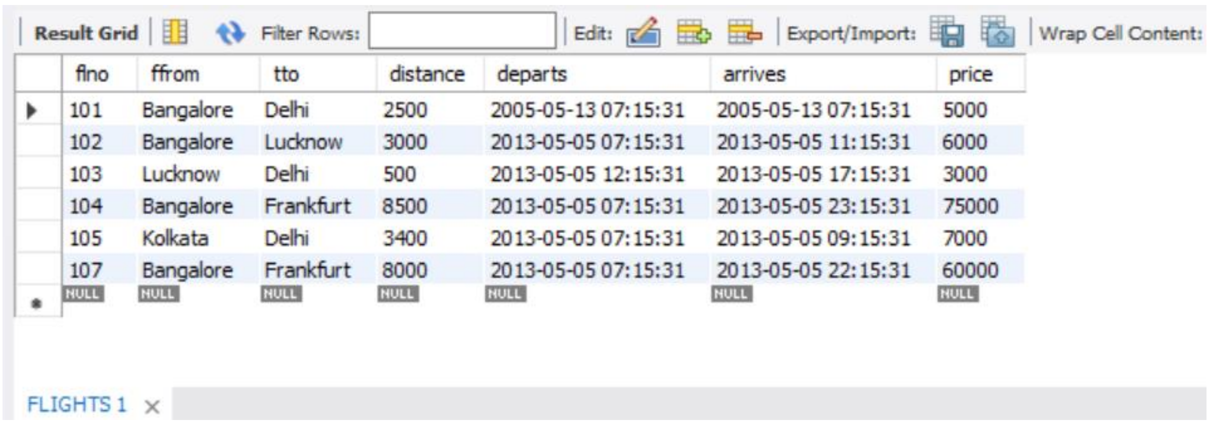
```
SQL> select * from Certified;

       EID        AID
---------- ----------
       701        101
       701        102
       701        106
       701        105
       702        104
       703        104
       704        104
       702        107
       703        107
       704        107
       702        101

       EID        AID
---------- ----------
       703        105
       704        105
       705        103

14 rows selected.
```

SQL> 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

7 rows selected.
```

```
CREATE DATABASE AIRLINE_FLIGHT_DATABASE;

USE AIRLINE_FLIGHT_DATABASE;


CREATE TABLE FLIGHTS

(

            flno int,

            ffrom varchar(40),

           tto varchar(40),

     distance int,

     departs datetime,

     arrives datetime,

     price int,

     primary key(flno)

);


CREATE TABLE AIRCRAFT

(

            aid int,

            aname varchar(40),

            cruisingrange int,

            primary key(aid)

);
```

```
CREATE TABLE EMPLOYEES
(
	eid int,
	ename varchar(40),
	salary int,
	primary key(eid)
);

CREATE TABLE CERTIFIED
(
	eid int,
	 aid int,
	FOREIGN KEY(aid) REFERENCES AIRCRAFT(aid),
	FOREIGN KEY(eid) REFERENCES EMPLOYEES(eid)
);

INSERT INTO FLIGHTS
VALUES (101,"Bangalore","Delhi",2500,'2005-05-13:07:15:31.000000','2005-05-
13:07:15:31.000000',5000),
(102,"Bangalore","Lucknow",3000,'2013-05-05:07:15:31.000000','2013-05-
05:11:15:31.000000',6000),
(103,"Lucknow","Delhi",500,'2013-05-05:12:15:31.000000','2013-05-05:17:15:31.000000',3000),
(107,"Bangalore","Frankfurt",8000,'2013-05-05:07:15:31.000000','2013-05-
05:22:15:31.000000',60000),
(104,"Bangalore","Frankfurt",8500,'2013-05-05:07:15:31.000000','2013-05-
05:23:15:31.000000',75000),
(105,"Kolkata","Delhi",3400,'2013-05-05:07:15:31.000000','2013-05-05:09:15:31.000000',7000);
SELECT * FROM FLIGHTS;
```



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

flno	ffrom	tto	distance	departs	arrives	price
101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 07:15:31	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
107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

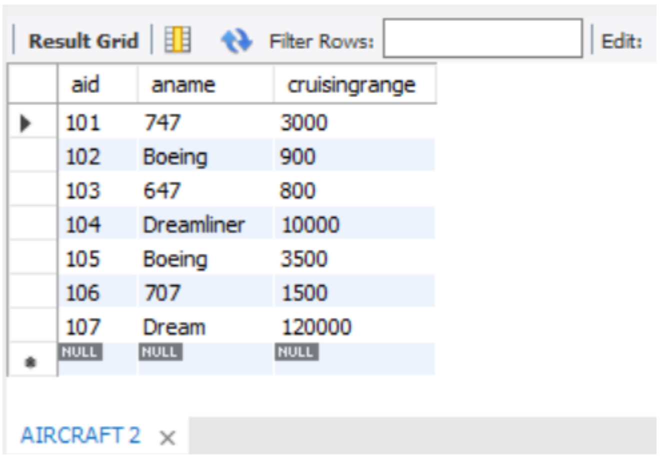
FLIGHTS 1 ×

INSERT INTO AIRCRAFT

VALUES (101,747,3000),(102,"Boeing",900),(103,647,800),(104,"Dreamliner",10000),

(105,"Boeing",3500),(106,707,1500),(107,"Dream",120000);

SELECT * FROM AIRCRAFT;



Result Grid | Filter Rows: | Edit:

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

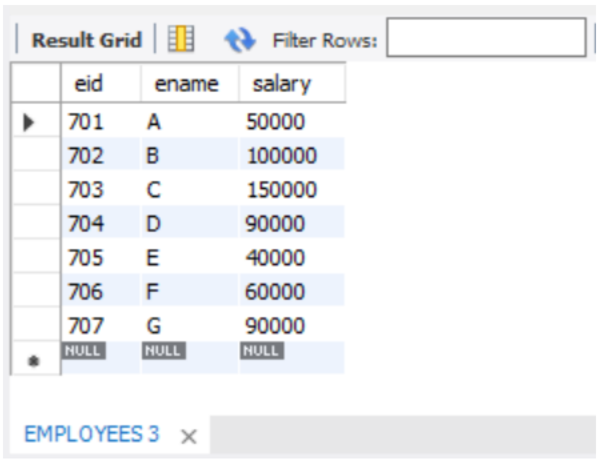
AIRCRAFT 2 ×

INSERT INTO EMPLOYEES

VALUES (701,"A",50000),(702,"B",100000),(703,"C",150000),(704,"D",90000),

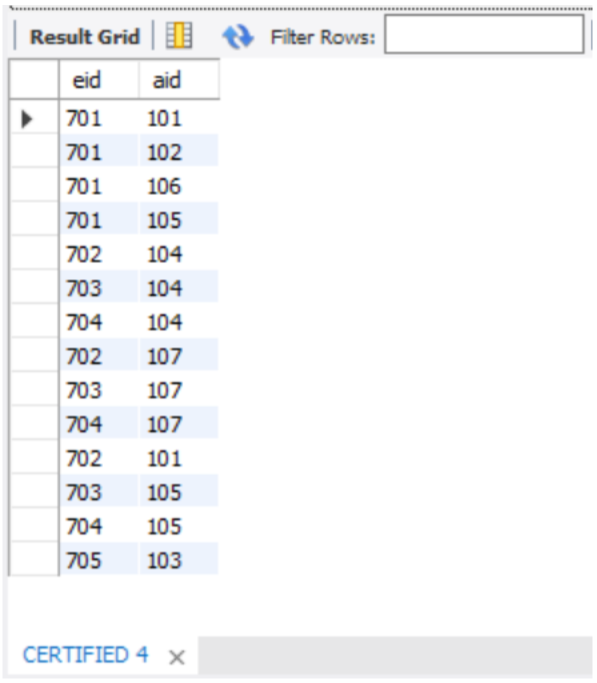
(705,"E",40000),(706,"F",60000),(707,"G",90000);

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

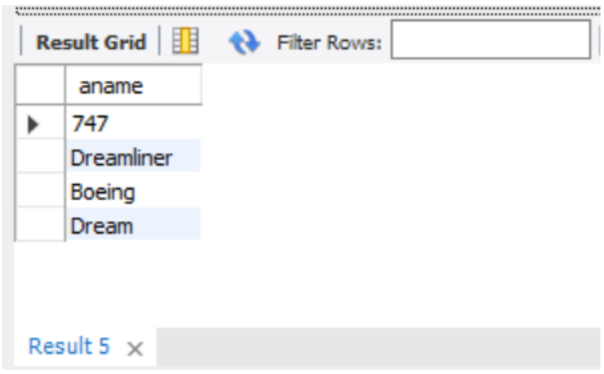
```
INSERT INTO CERTIFIED

VALUES (701,101),(701,102),(701,106),(701,105),(702,104),(703,104),(704,104),(702,107),

           (703,107),(704,107),(702,101),(703,105),(704,105),(705,103);

SELECT * FROM CERTIFIED;
```



eid	aid
701	101
701	102
701	106
701	105
702	104
703	104
704	104
702	107
703	107
704	107
702	101
703	105
704	105
705	103

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

```
SELECT distinct a.aname
FROM  AIRCRAFT a,EMPLOYEES e,CERTIFIED c

WHERE a.aid=c.aid and e.eid=c.eid and e.salary>80000;
```

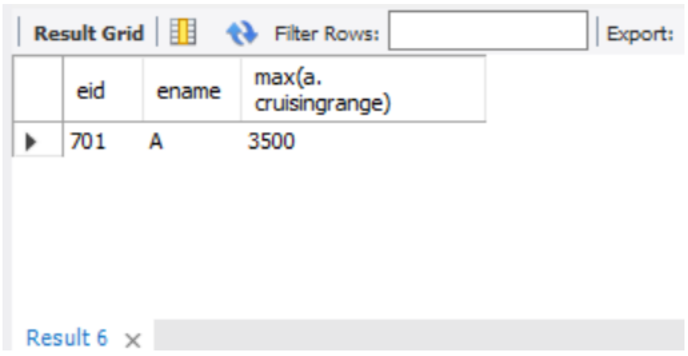


aname
747
Dreamliner
Boeing
Dream

Result 5

b.For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.

```
SELECT e.eid,e.ename,max(a. cruisingrange)

FROM EMPLOYEES e,CERTIFIED c,AIRCRAFT a

WHERE e.eid=c.eid and a.aid=c.aid

group by e.ename

having count(c.aid)>3;
```

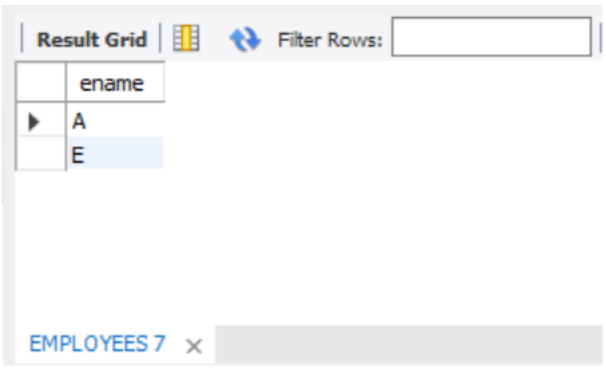


eid	ename	max(a. cruisingrange)
701	A	3500

Result 6

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

```
SELECT e.ename

FROM EMPLOYEES e

WHERE salary < (select min(price)

                from FLIGHTS

                where ffrom="Bangalore" and tto="Frankfurt");
```



Result Grid | Filter Rows:

	ename
▶	A
	E

EMPLOYEES 7 ×

d.For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

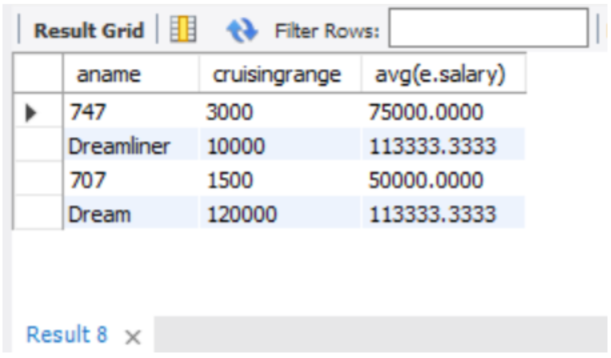
SELECT a.aname,a.cruisingrange,avg(e.salary)

FROM AIRCRAFT a,EMPLOYEES e,CERTIFIED c

WHERE c.eid=e.eid and c.aid=a.aid

group by a.aname

having a.cruisingrange > 1000;



Result Grid | Filter Rows:

	aname	cruisingrange	avg(e.salary)
▶	747	3000	75000.0000
	Dreamliner	10000	113333.3333
	707	1500	50000.0000
	Dream	120000	113333.3333

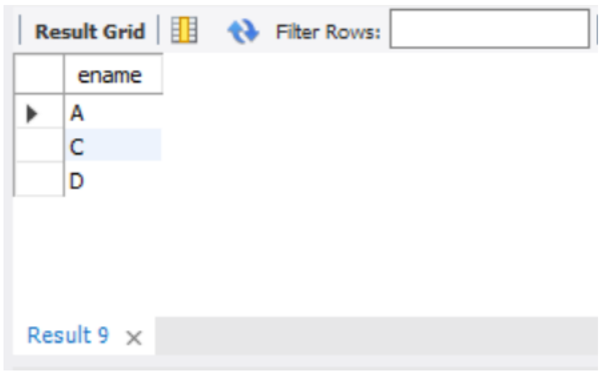
Result 8 ×

e.Find the names of pilots certified for some Boeing aircraft.

SELECT distinct e.ename

FROM EMPLOYEES e,CERTIFIED c,AIRCRAFT a

WHERE e.eid=c.eid and a.aid=c.aid and aname like "Boeing";



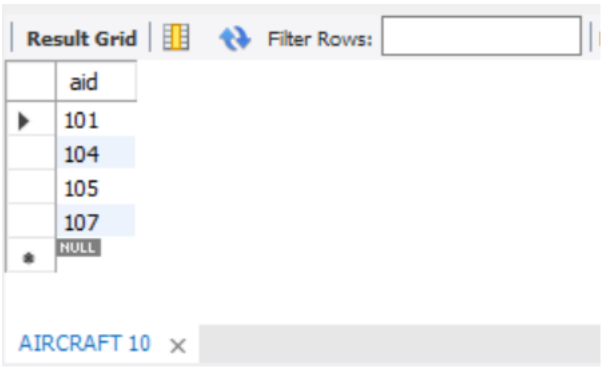
Result Grid | Filter Rows:

	ename
▶	A
	C
	D

Result 9 ×

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

```
SELECT a.aid
FROM AIRCRAFT a
WHERE a. cruisingrange >= (select distance
                                   from FLIGHTS
                           where ffrom="Bangalore" and tto="Delhi");
```

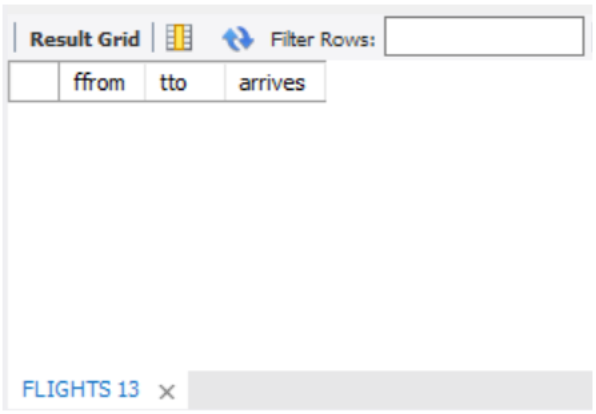


```
aid
101
104
105
107
NULL
```

AIRCRAFT 10

g. A customer wants to travel from Bangalore to Kolkata New with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in Kolkata by 6 p.m.

```
SELECT  f.ffrom,f.tto,f.arrives
FROM    FLIGHTS f
WHERE (f.ffrom="Bangalore" and f.tto=(select ffrom
                                          from FLIGHTS
                                          where tto="Kolkata")) or f.tto="Kolkata";
```



```
ffrom   tto   arrives
```

FLIGHTS 13