B.M.S. COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE UNDER VTU)
BENGALURU-19

LAB TEST 1 REPORT

NAME: Ishan Bhandari

USN: 1BM19CS198

COURSE NAME: DATABASE MANAGEMENT SYSTEMS

COURSE TITLE: 19CS4PCDBM

SEMESTER: 4

SECTION: D

LAB PROGRAMS 1-5:

PROGRAM 1: INSURANCE DATABASE

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

PERSON (driver_id: String, name: String, address: String)

CAR (reg_num: String, model: String, year: int)

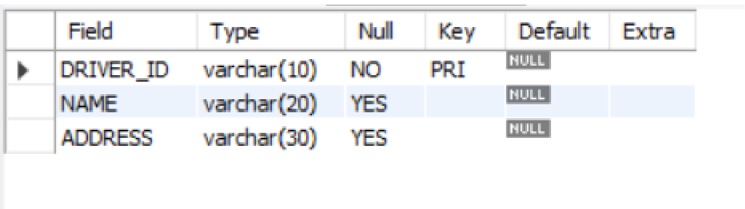
ACCIDENT (report_num: int, accident_date: date, location: String)

OWNS (driver_id: String, reg_num: String)

PARTICIPATED (driver_id: String, reg_num: String, report_num: int, damage_amount: int)

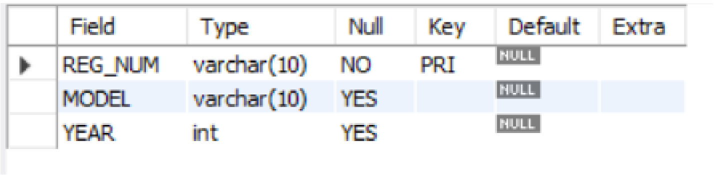
i) Create the above tables by properly specifying the primary keys and the foreign keys.

CREATE TABLE PERSON(DRIVER_ID VARCHAR(10), NAME VARCHAR(20), ADDRESS VARCHAR(30), PRIMARY KEY (DRIVER_ID));



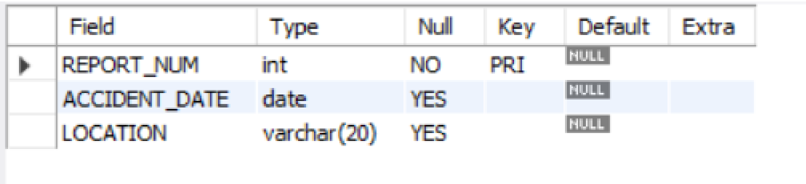
	Field	Type	Null	Key	Default	Extra
▸	DRIVER_ID	varchar(10)	NO	PRI	NULL	
	NAME	varchar(20)	YES		NULL	
	ADDRESS	varchar(30)	YES		NULL	

CREATE
TABLE CAR(REG_NUM VARCHAR(10), MODEL VARCHAR(10), YEAR INT, PRIMARY KEY(REG_NUM));



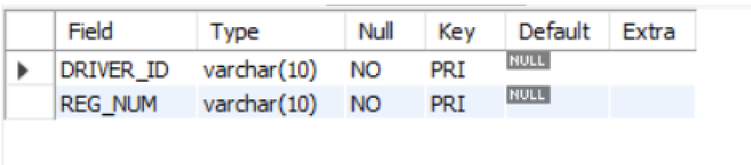
	Field	Type	Null	Key	Default	Extra
▸	REG_NUM	varchar(10)	NO	PRI	NULL	
	MODEL	varchar(10)	YES		NULL	
	YEAR	int	YES		NULL	

CREATE TABLE ACCIDENT(REPORT_NUM INT, ACCIDENT_DATE DATE, LOCATION VARCHAR(20), PRIMARY KEY(REPORT_NUM));



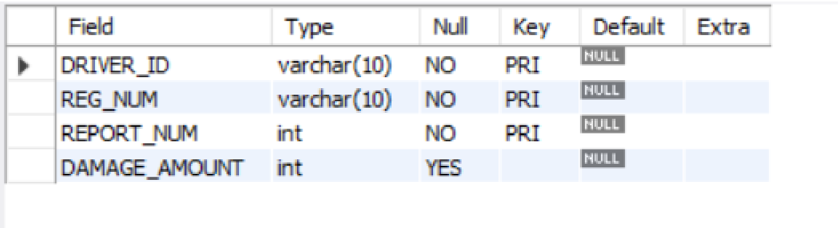
	Field	Type	Null	Key	Default	Extra
▶	REPORT_NUM	int	NO	PRI	NULL	
	ACCIDENT_DATE	date	YES		NULL	
	LOCATION	varchar(20)	YES		NULL	

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));



	Field	Type	Null	Key	Default	Extra
▶	DRIVER_ID	varchar(10)	NO	PRI	NULL	
	REG_NUM	varchar(10)	NO	PRI	NULL	

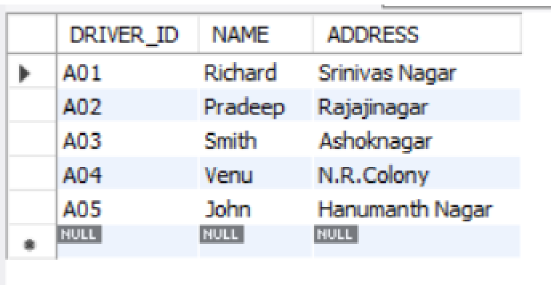
CREATE TABLE PARTICIPATED(DRIVER_ID VARCHAR(10), REG_NUM VARCHAR(10), REPORT_NUM INT, DAMAGE_AMOUNT INT, PRIMARY KEY(DRIVER_ID, REG_NUM, REPORT_NUM), FOREIGN KEY(DRIVER_ID) REFERENCES PERSON(DRIVER_ID), FOREIGN KEY(REG_NUM) REFERENCES CAR(REG_NUM), FOREIGN KEY(REPORT_NUM) REFERENCES ACCIDENT (REPORT_NUM));



	Field	Type	Null	Key	Default	Extra
▶	DRIVER_ID	varchar(10)	NO	PRI	NULL	
	REG_NUM	varchar(10)	NO	PRI	NULL	
	REPORT_NUM	int	NO	PRI	NULL	
	DAMAGE_AMOUNT	int	YES		NULL	

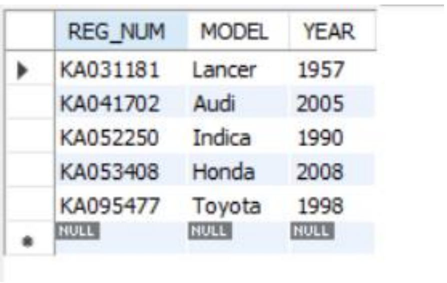
ii)Enter at least five tuples for each relation.

INSERT INTO PERSON VALUES('A01', 'Richard', 'Srinivas Nagar');
INSERT INTO PERSON VALUES('A02', 'Pradeep', 'Rajajinagar');
INSERT INTO PERSON VALUES('A03', 'Smith', 'Ashoknagar');
INSERT INTO PERSON VALUES('A04', 'Venu', 'N.R.Colony');
INSERT INTO PERSON VALUES('A05', 'John', 'Hanumanth Nagar');



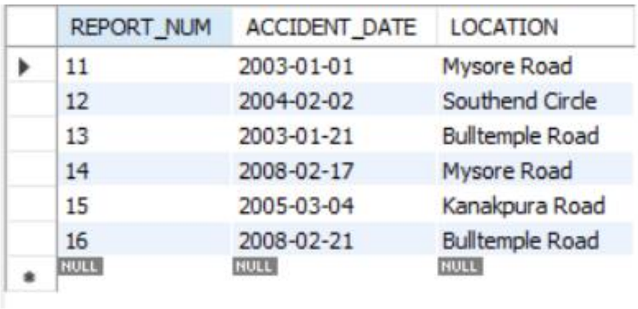
	DRIVER_ID	NAME	ADDRESS
▶	A01	Richard	Srinivas Nagar
	A02	Pradeep	Rajajinagar
	A03	Smith	Ashoknagar
	A04	Venu	N.R.Colony
	A05	John	Hanumanth Nagar
*	NULL	NULL	NULL

```
INSERT INTO CAR VALUES('KA052250', 'Indica', 1990);
INSERT INTO CAR VALUES('KA031181', 'Lancer', 1957);
INSERT INTO CAR VALUES('KA095477', 'Toyota', 1998);
INSERT INTO CAR VALUES('KA053408', 'Honda', 2008);
INSERT INTO CAR VALUES('KA041702', 'Audi', 2005);
```



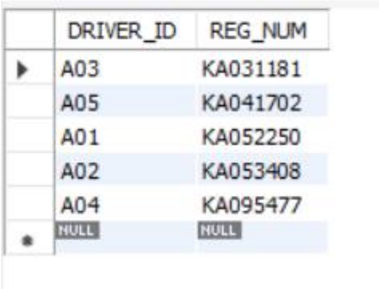
	REG_NUM	MODEL	YEAR
▶	KA031181	Lancer	1957
	KA041702	Audi	2005
	KA052250	Indica	1990
	KA053408	Honda	2008
	KA095477	Toyota	1998
*	NULL	NULL	NULL

```
INSERT INTO ACCIDENT VALUES(11, '2003-01-01', 'Mysore Road');
INSERT INTO ACCIDENT VALUES(12, '2004-02-02', 'Southend Circle');
INSERT INTO ACCIDENT VALUES(13, '2003-01-21', 'Bulltemple Road');
INSERT INTO ACCIDENT VALUES(14, '2008-02-17', 'Mysore Road');
INSERT INTO ACCIDENT VALUES(15, '2005-03-04', 'Kanakpura Road');
```



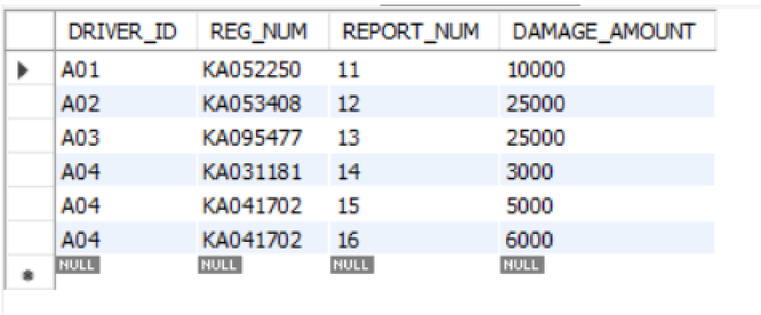
	REPORT_NUM	ACCIDENT_DATE	LOCATION
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	Southend Circle
	13	2003-01-21	Bulltemple Road
	14	2008-02-17	Mysore Road
	15	2005-03-04	Kanakpura Road
	16	2008-02-21	Bulltemple Road
*	NULL	NULL	NULL

```
INSERT INTO OWNS VALUES('A01', 'KA052250');
INSERT INTO OWNS VALUES('A02', 'KA053408');
INSERT INTO OWNS VALUES('A03', 'KA031181');
INSERT INTO OWNS VALUES('A04', 'KA095477');
INSERT INTO OWNS VALUES('A05', 'KA041702');
```



	DRIVER_ID	REG_NUM
▶	A03	KA031181
	A05	KA041702
	A01	KA052250
	A02	KA053408
	A04	KA095477
*	NULL	NULL

INSERT INTO PARTICIPATED VALUES('A01', 'KA052250', 11, 10000);
INSERT INTO PARTICIPATED VALUES('A02', 'KA053408', 12, 50000);
INSERT INTO PARTICIPATED VALUES('A03', 'KA095477', 13, 25000);
INSERT INTO PARTICIPATED VALUES('A04', 'KA031181', 14, 3000);
INSERT INTO PARTICIPATED VALUES('A05', 'KA041702', 15, 5000);

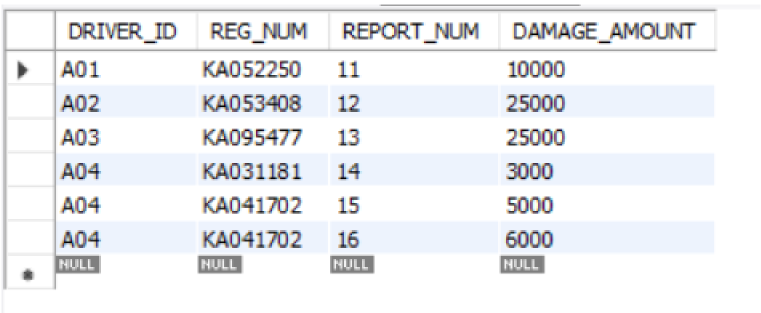


	DRIVER_ID	REG_NUM	REPORT_NUM	DAMAGE_AMOUNT
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A04	KA041702	15	5000
	A04	KA041702	16	6000
*	NULL	NULL	NULL	NULL

iii) Demonstrate how you:

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

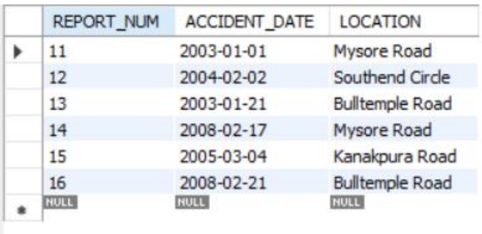
UPDATE PARTICIPATED SET DAMAGE_AMOUNT = 25000 WHERE REPORT_NUM = 12;



	DRIVER_ID	REG_NUM	REPORT_NUM	DAMAGE_AMOUNT
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A04	KA041702	15	5000
	A04	KA041702	16	6000
*	NULL	NULL	NULL	NULL

b. Add a new accident to the database.

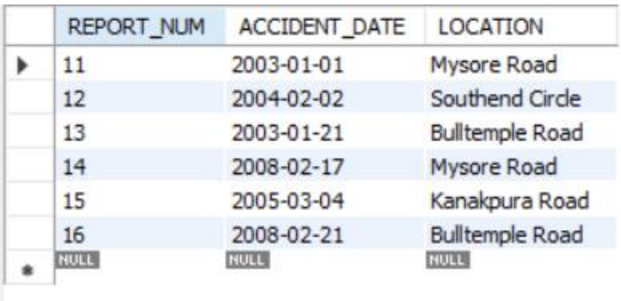
INSERT INTO ACCIDENT VALUES(16, '2008-02-21', 'Bulltemple Road');



REPORT_NUM	ACCIDENT_DATE	LOCATION
11	2003-01-01	Mysore Road
12	2004-02-02	Southend Circle
13	2003-01-21	Bulltemple Road
14	2008-02-17	Mysore Road
15	2005-03-04	Kanakpura Road
16	2008-02-21	Bulltemple Road
NULL	NULL	NULL

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

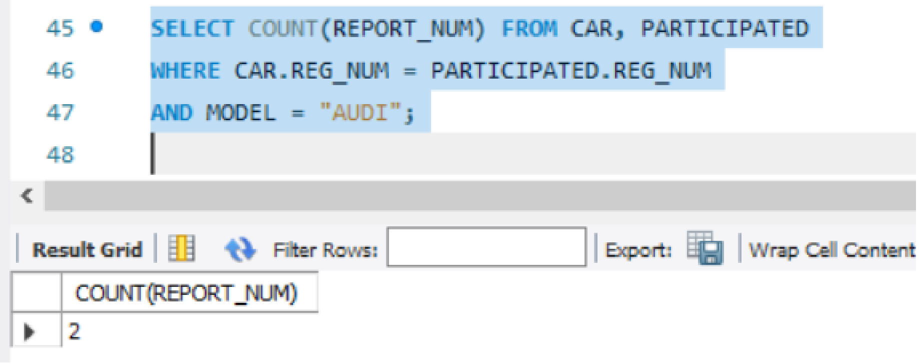
SELECT COUNT(DISTINCT DRIVER_ID) FROM ACCIDENT, PARTICIPATED
WHERE ACCIDENT.REPORT_NUM = PARTICIPATED.REPORT_NUM
AND ACCIDENT_DATE LIKE '2008%';



REPORT_NUM	ACCIDENT_DATE	LOCATION
11	2003-01-01	Mysore Road
12	2004-02-02	Southend Circle
13	2003-01-21	Bulltemple Road
14	2008-02-17	Mysore Road
15	2005-03-04	Kanakpura Road
16	2008-02-21	Bulltemple Road
NULL	NULL	NULL

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

SELECT COUNT(REPORT_NUM) FROM CAR, PARTICIPATED
WHERE CAR.REG_NUM = PARTICIPATED.REG_NUM
AND MODEL = "AUDI";



```
SELECT COUNT(REPORT_NUM) FROM CAR, PARTICIPATED
WHERE CAR.REG_NUM = PARTICIPATED.REG_NUM
AND MODEL = "AUDI";
```

COUNT(REPORT_NUM)
2

PROGRAM 2: BANKING ENTERPRISE DATABASE

Consider the following database for a banking enterprise.

Branch (branch-name: String, branch-city: String, assets: real)

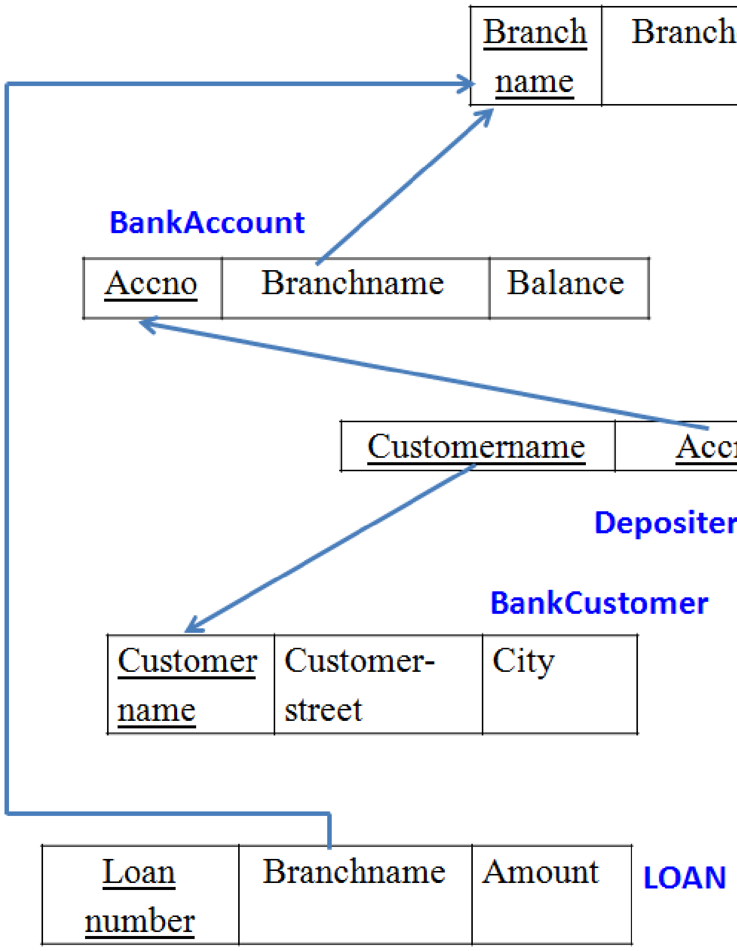
BankAccount(accno: int, branch-name: String, balance: real)

BankCustomer (customer-name: String, customer-street: String, customer-city: String)

Depositer(customer-name: String, accno: int)

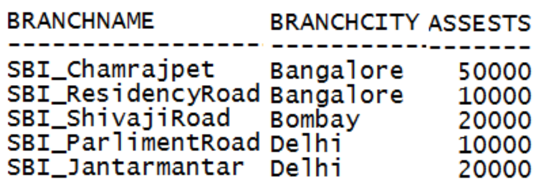
Loan (loan-number: int, branch-name: String, amount: real)

Schema Diagram



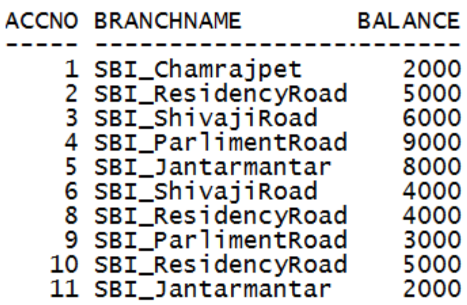
Sample Table data

Branch



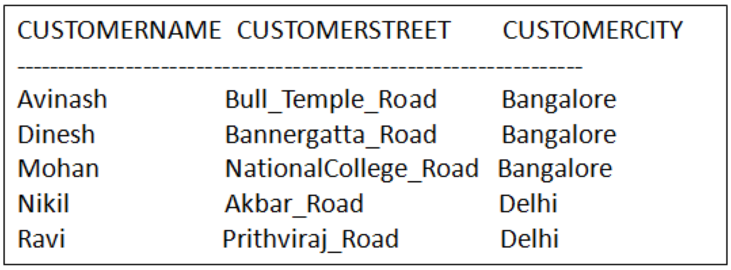
```
BRANCHNAME        BRANCHCITY ASSESTS
----------------- ---------- -------
SBI_Chamrajpet    Bangalore    50000
SBI_ResidencyRoad Bangalore    10000
SBI_ShivajiRoad   Bombay       20000
SBI_ParlimentRoad Delhi        10000
SBI_Jantarmantar  Delhi        20000
```

BankAccount



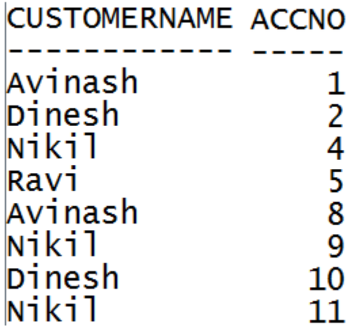
```
ACCNO BRANCHNAME        BALANCE
----- ----------------- -------
    1 SBI_Chamrajpet       2000
    2 SBI_ResidencyRoad    5000
    3 SBI_ShivajiRoad      6000
    4 SBI_ParlimentRoad    9000
    5 SBI_Jantarmantar     8000
    6 SBI_ShivajiRoad      4000
    8 SBI_ResidencyRoad    4000
    9 SBI_ParlimentRoad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

BankCustomer



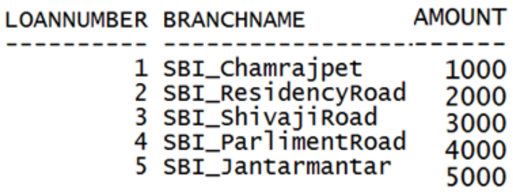
```
CUSTOMERNAME  CUSTOMERSTREET        CUSTOMERCITY
----------------------------------------------------------------
Avinash       Bull_Temple_Road      Bangalore
Dinesh        Bannergatta_Road      Bangalore
Mohan         NationalCollege_Road  Bangalore
Nikil         Akbar_Road            Delhi
Ravi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

Depositer



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

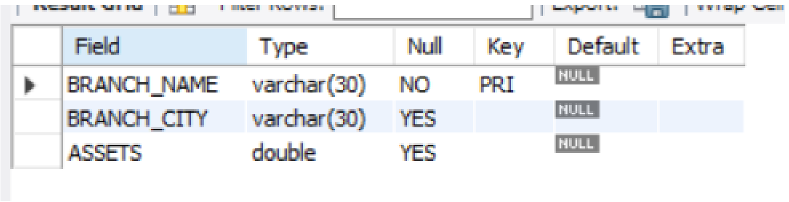
Loan



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

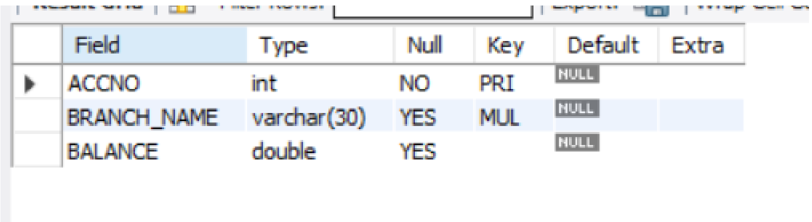
i. Create the above tables by properly specifying the primary keys and the foreign keys.

CREATE TABLE BRANCH (BRANCH_NAME VARCHAR(30), BRANCH_CITY VARCHAR(30), ASSETS REAL, PRIMARY KEY (BRANCH_NAME));



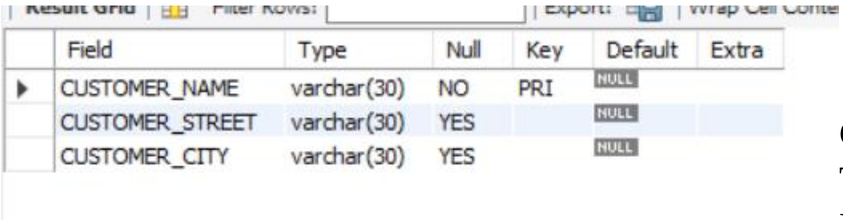
Field	Type	Null	Key	Default	Extra
BRANCH_NAME	varchar(30)	NO	PRI	NULL	
BRANCH_CITY	varchar(30)	YES		NULL	
ASSETS	double	YES		NULL	

CREATE TABLE BANK_ACCOUNT (ACCNO INT, BRANCH_NAME VARCHAR(30), BALANCE REAL, PRIMARY KEY (ACCNO), FOREIGN KEY (BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));



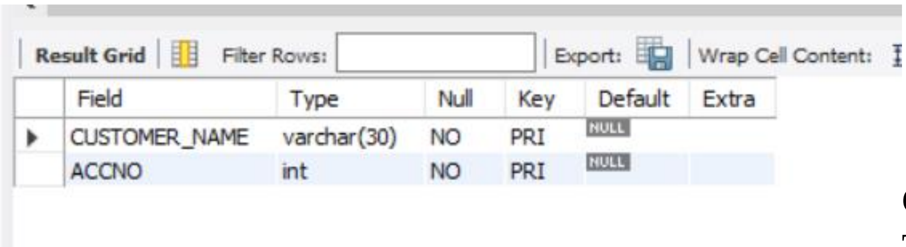
Field	Type	Null	Key	Default	Extra
ACCNO	int	NO	PRI	NULL	
BRANCH_NAME	varchar(30)	YES	MUL	NULL	
BALANCE	double	YES		NULL	

CREATE TABLE BANK_CUSTOMER (CUSTOMER_NAME VARCHAR(30), CUSTOMER_STREET VARCHAR(30), CUSTOMER_CITY VARCHAR(30), PRIMARY KEY(CUSTOMER_NAME));



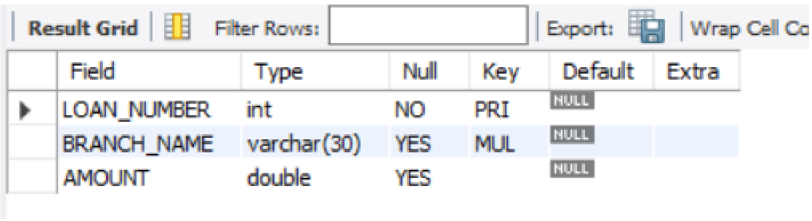
Field	Type	Null	Key	Default	Extra
CUSTOMER_NAME	varchar(30)	NO	PRI	NULL	
CUSTOMER_STREET	varchar(30)	YES		NULL	
CUSTOMER_CITY	varchar(30)	YES		NULL	

CREATE TABLE DEPOSITER (CUSTOMER_NAME VARCHAR(30), ACCNO INT, PRIMARY KEY(CUSTOMER_NAME, ACCNO), FOREIGN KEY (CUSTOMER_NAME) REFERENCES BANK_CUSTOMER (CUSTOMER_NAME), FOREIGN KEY (ACCNO) REFERENCES BANK_ACCOUNT(ACCNO));



Field	Type	Null	Key	Default	Extra
CUSTOMER_NAME	varchar(30)	NO	PRI	NULL	
ACCNO	int	NO	PRI	NULL	

CREATE TABLE LOAN (LOAN_NUMBER INT, BRANCH_NAME VARCHAR(30), AMOUNT REAL, PRIMARY KEY (LOAN_NUMBER), FOREIGN KEY (BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));

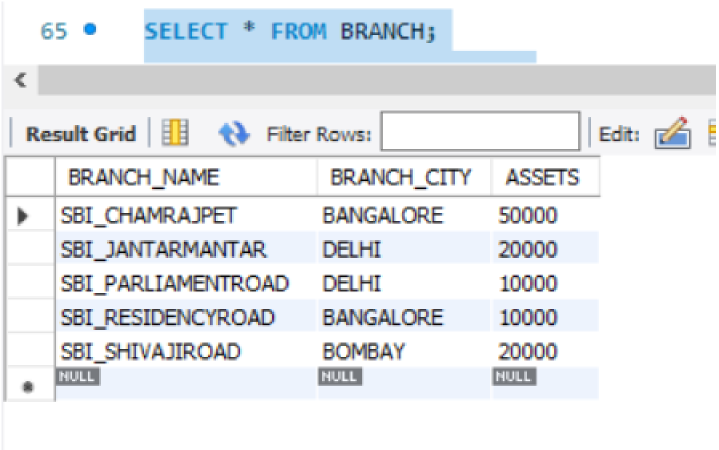


Field	Type	Null	Key	Default	Extra
LOAN_NUMBER	int	NO	PRI	NULL	
BRANCH_NAME	varchar(30)	YES	MUL	NULL	
AMOUNT	double	YES		NULL	

ii. Enter at least five tuples for each relation.

INSERT INTO BRANCH VALUES ('SBI_CHAMRAJPET', 'BANGALORE', 50000);

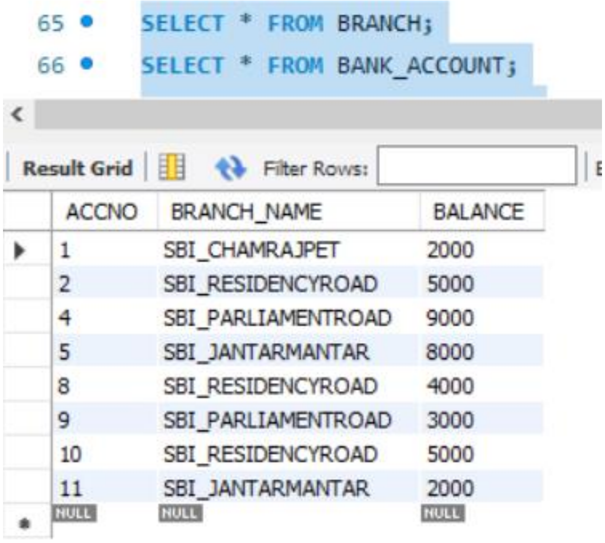
```
INSERT INTO BRANCH VALUES ('SBI_RESIDENCYROAD', 'BANGALORE', 10000);
INSERT INTO BRANCH VALUES ('SBI_SHIVAJIROAD', 'BOMBAY', 20000);
INSERT INTO BRANCH VALUES ('SBI_PARLIAMENTROAD', 'DELHI', 10000);
INSERT INTO BRANCH VALUES ('SBI_JANTARMANTAR', 'DELHI', 20000);
```



```
SELECT * FROM BRANCH;
```

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

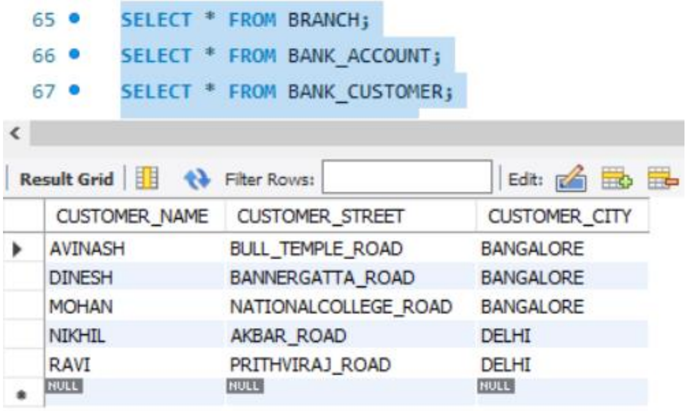
```
INSERT INTO BANK_ACCOUNT VALUES ( 1,'SBI_CHAMRAJPET', 2000);
INSERT INTO BANK_ACCOUNT VALUES ( 2,'SBI_RESIDENCYROAD', 5000);
INSERT INTO BANK_ACCOUNT VALUES ( 3,'SBI_SHIVAJIROAD', 6000);
INSERT INTO BANK_ACCOUNT VALUES ( 4,'SBI_PARLIAMENTROAD', 9000);
INSERT INTO BANK_ACCOUNT VALUES ( 5,'SBI_JANTARMANTAR', 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);
```



```
SELECT * FROM BRANCH;
SELECT * FROM BANK_ACCOUNT;
```

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

INSERT INTO BANK_CUSTOMER VALUES ('AVINASH', 'BULL_TEMPLE_ROAD', 'BANGALORE');
INSERT INTO BANK_CUSTOMER VALUES ('DINESH', 'BANNERGATTA_ROAD', 'BANGALORE');
INSERT INTO BANK_CUSTOMER VALUES ('MOHAN', 'NATIONALCOLLEGE_ROAD', 'BANGALORE');
INSERT INTO BANK_CUSTOMER VALUES ('NIKHIL', 'AKBAR_ROAD', 'DELHI');
INSERT INTO BANK_CUSTOMER VALUES ('RAVI', 'PRITHVIRAJ_ROAD', 'DELHI');

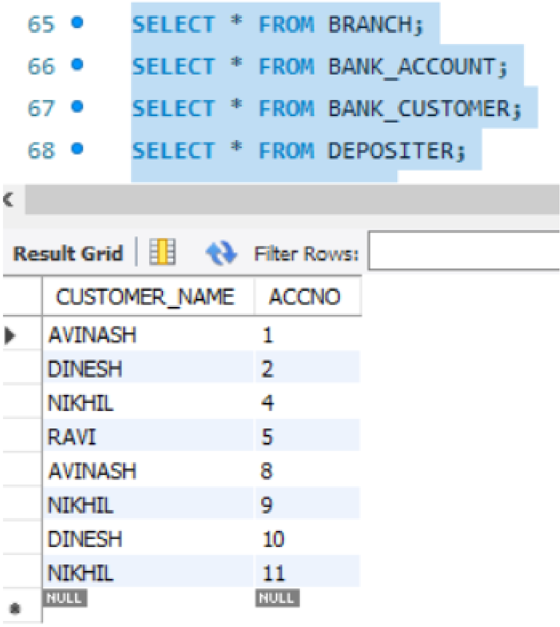


```
SELECT * FROM BRANCH;
SELECT * FROM BANK_ACCOUNT;
SELECT * FROM BANK_CUSTOMER;
```

CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
AVINASH	BULL_TEMPLE_ROAD	BANGALORE
DINESH	BANNERGATTA_ROAD	BANGALORE
MOHAN	NATIONALCOLLEGE_ROAD	BANGALORE
NIKHIL	AKBAR_ROAD	DELHI
RAVI	PRITHVIRAJ_ROAD	DELHI
NULL	NULL	NULL

INSERT INTO DEPOSITER VALUES('AVINASH', 1);
INSERT INTO DEPOSITER VALUES('DINESH', 2);
INSERT INTO DEPOSITER VALUES('NIKHIL', 4);
INSERT INTO DEPOSITER VALUES('RAVI', 5);
INSERT INTO DEPOSITER VALUES('AVINASH', 8);

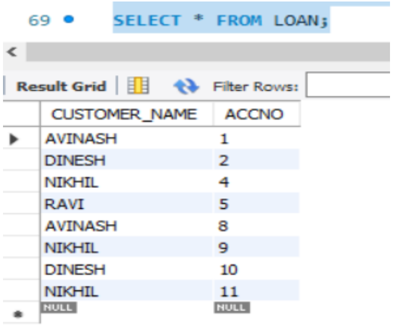
INSERT INTO DEPOSITER VALUES('NIKHIL', 9);
INSERT INTO DEPOSITER VALUES('DINESH', 10);
INSERT INTO DEPOSITER VALUES('NIKHIL', 11);



```
SELECT * FROM BRANCH;
SELECT * FROM BANK_ACCOUNT;
SELECT * FROM BANK_CUSTOMER;
SELECT * FROM DEPOSITER;
```

CUSTOMER_NAME	ACCNO
AVINASH	1
DINESH	2
NIKHIL	4
RAVI	5
AVINASH	8
NIKHIL	9
DINESH	10
NIKHIL	11
NULL	NULL

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);

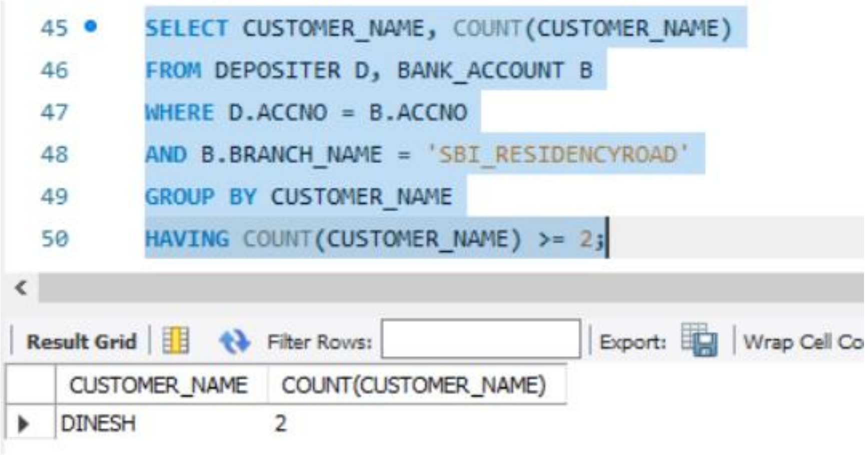


```
SELECT * FROM LOAN;
```

CUSTOMER_NAME	ACCNO
AVINASH	1
DINESH	2
NIKHIL	4
RAVI	5
AVINASH	8
NIKHIL	9
DINESH	10
NIKHIL	11
NULL	NULL

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

SELECT CUSTOMER_NAME, COUNT(CUSTOMER_NAME)
FROM DEPOSITER D, BANK_ACCOUNT B
WHERE D.ACCNO = B.ACCNO
AND B.BRANCH_NAME = 'SBI_RESIDENCYROAD'
GROUP BY CUSTOMER_NAME
HAVING COUNT(CUSTOMER_NAME) >= 2;

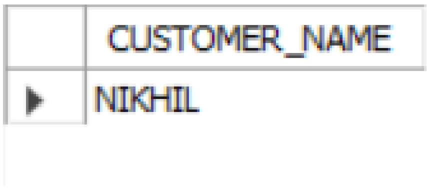


```
SELECT CUSTOMER_NAME, COUNT(CUSTOMER_NAME)
FROM DEPOSITER D, BANK_ACCOUNT B
WHERE D.ACCNO = B.ACCNO
AND B.BRANCH_NAME = 'SBI_RESIDENCYROAD'
GROUP BY CUSTOMER_NAME
HAVING COUNT(CUSTOMER_NAME) >= 2;
```

CUSTOMER_NAME	COUNT(CUSTOMER_NAME)
DINESH	2

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

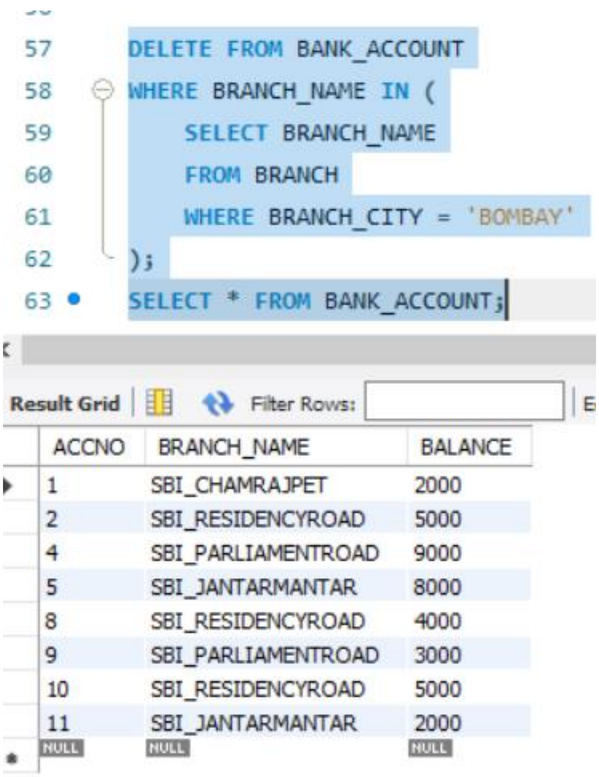
```
SELECT D.CUSTOMER_NAME
FROM DEPOSITER D,BRANCH B,BANK_ACCOUNT A
WHERE B.BRANCH_NAME=A.BRANCH_NAME
AND A.ACCNO=D.ACCNO
AND BRANCH_CITY='DELHI'
GROUP BY D.CUSTOMER_NAME
 HAVING COUNT(DISTINCT B.BRANCH_NAME)=(
                SELECT COUNT(BRANCH_NAME)
                FROM BRANCH
                WHERE BRANCH_CITY='DELHI');
```



CUSTOMER_NAME
NIKHIL

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

```
DELETE FROM BANK_ACCOUNT
WHERE BRANCH_NAME IN (
          SELECT BRANCH_NAME
    FROM BRANCH
    WHERE BRANCH_CITY = 'BOMBAY'
);
SELECT * FROM BANK_ACCOUNT;
```



```
DELETE FROM BANK_ACCOUNT
WHERE BRANCH_NAME IN (
    SELECT BRANCH_NAME
    FROM BRANCH
    WHERE BRANCH_CITY = 'BOMBAY'
);
SELECT * FROM BANK_ACCOUNT;
```

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

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.

Schema Diagram

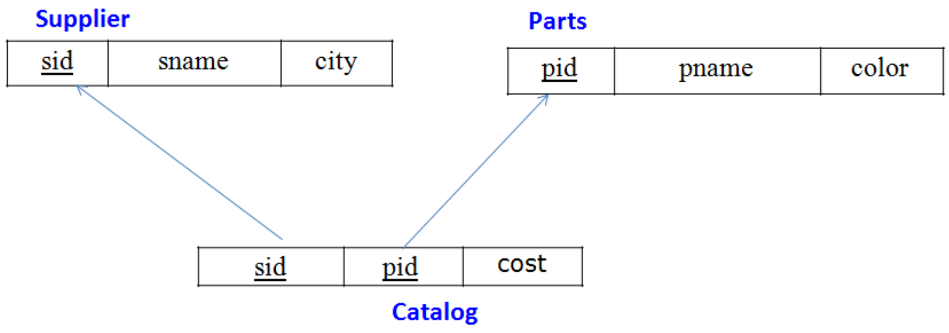
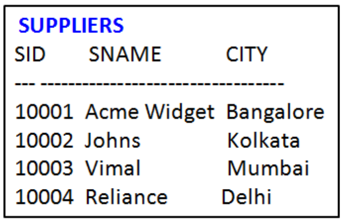
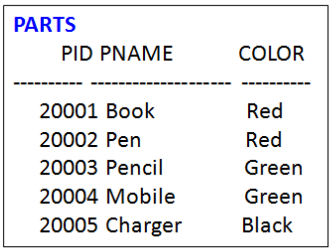


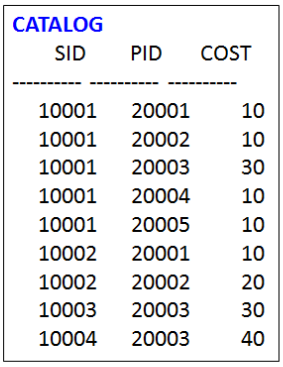
Table Data



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



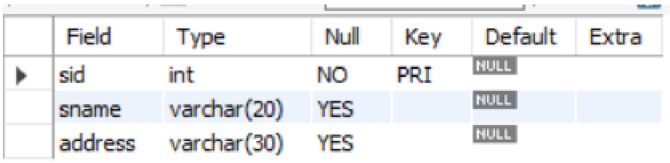
```
PARTS
      PID PNAME              COLOR
---------- ------------------- ----------
   20001 Book                 Red
   20002 Pen                  Red
   20003 Pencil               Green
   20004 Mobile               Green
   20005 Charger              Black
```



```
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
```

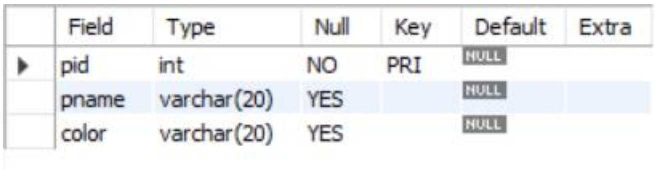
Creation of Tables:

```
CREATE TABLE suppliers (
        sid INT,
   sname VARCHAR(20),
   address VARCHAR(30),
    PRIMARY KEY (sid)
);
```



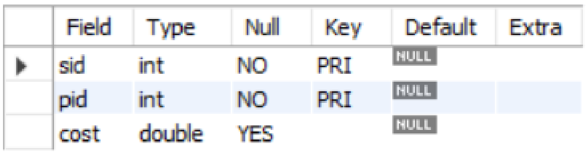
	Field	Type	Null	Key	Default	Extra
▶	sid	int	NO	PRI	NULL	
	sname	varchar(20)	YES		NULL	
	address	varchar(30)	YES		NULL	

```
CREATE TABLE parts (
        pid INT,
    pname VARCHAR(20),
    color VARCHAR(20),
    PRIMARY KEY (pid)
);
```



	Field	Type	Null	Key	Default	Extra
▶	pid	int	NO	PRI	NULL	
	pname	varchar(20)	YES		NULL	
	color	varchar(20)	YES		NULL	

```
CREATE TABLE catalog (
    sid INT,
    pid INT,
    cost REAL,
    PRIMARY KEY(sid, pid),
    FOREIGN KEY (sid) REFERENCES suppliers(sid),
    FOREIGN KEY (pid) REFERENCES parts(pid)
);
```



	Field	Type	Null	Key	Default	Extra
▶	sid	int	NO	PRI	NULL	
	pid	int	NO	PRI	NULL	
	cost	double	YES		NULL	

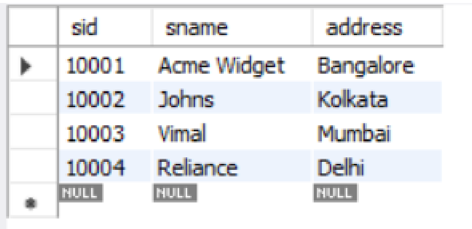
Inserting Values into the tables:

INSERT INTO suppliers VALUES (10001, 'Acme Widget', 'Bangalore');

INSERT INTO suppliers VALUES (10002, 'Johns', 'Kolkata');

INSERT INTO suppliers VALUES (10003, 'Vimal', 'Mumbai');

INSERT INTO suppliers VALUES (10004, 'Reliance', 'Delhi');



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

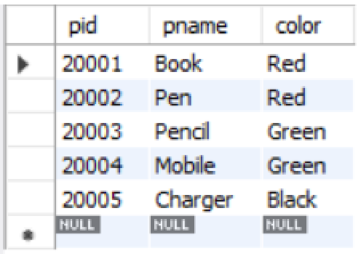
INSERT INTO parts VALUES (20001, 'Book', 'Red');

INSERT INTO parts VALUES (20002, 'Pen', 'Red');

INSERT INTO parts VALUES (20003, 'Pencil', 'Green');

INSERT INTO parts VALUES (20004, 'Mobile', 'Green');

INSERT INTO parts VALUES (20005, 'Charger', 'Black');



	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);

INSERT INTO catalog VALUES (10001, 20002, 10);

INSERT INTO catalog VALUES (10001, 20003, 30);

INSERT INTO catalog VALUES (10001, 20004, 10);

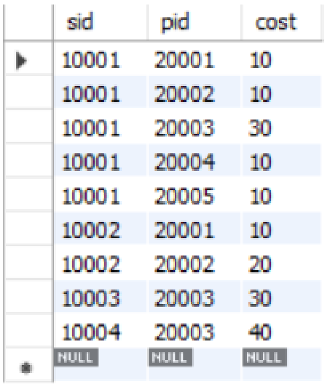
INSERT INTO catalog VALUES (10001, 20005, 10);

INSERT INTO catalog VALUES (10002, 20001, 10);

INSERT INTO catalog VALUES (10002, 20002, 20);

INSERT INTO catalog VALUES (10003, 20003, 30);

INSERT INTO catalog VALUES (10004, 20003, 40);

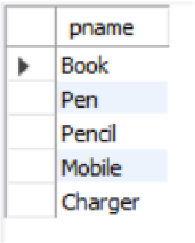


sid	pid	cost
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40
NULL	NULL	NULL

Write the following queries in SQL:

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

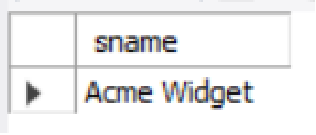
```
SELECT DISTINCT(pname)

FROM parts p, catalog c

WHERE p.pid = c.pid

AND c.sid IS NOT NULL;
```



pname
Book
Pen
Pencil
Mobile
Charger

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

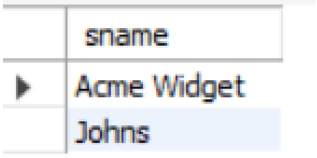
```
SELECT s.sname
FROM suppliers s
WHERE NOT EXISTS (
 SELECT p.pid
 FROM parts p
 WHERE NOT EXISTS (
 SELECT c.sid
 FROM catalog c
 WHERE c.sid = s.sid
 AND c.pid = p.pid
 )
);
```



sname
Acme Widget

3. Find the snames of suppliers who supply every red part.

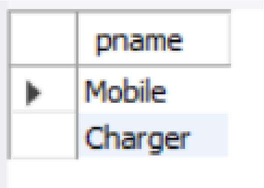
```
SELECT s.sname
FROM suppliers s
WHERE NOT EXISTS (
 SELECT p.pid
 FROM parts p
 WHERE p.color = 'Red'
 AND NOT EXISTS (
 SELECT c.sid
 FROM catalog c
 WHERE c.sid = s.sid
 AND c.pid = p.pid
 )
);
```



sname
Acme Widget
Johns

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

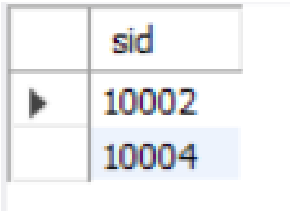
```
SELECT p.pname
FROM parts p, suppliers s, catalog c
WHERE c.sid = s.sid
AND p.pid = c.pid
AND s.sname = 'Acme Widget'
AND NOT EXISTS (
SELECT c1.pid
FROM catalog c1, suppliers s1
WHERE c1.pid = p.pid
AND c1.sid = s1.sid
AND s1.sname <> 'Acme Widget'
);
```



pname
Mobile
Charger

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

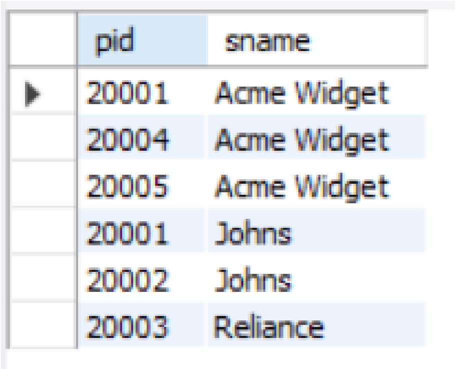
```
SELECT DISTINCT sid
FROM catalog c
WHERE c.cost > (
 SELECT AVG(c1.cost)
 FROM catalog c1
    WHERE c1.pid = c.pid
);
```



sid
10002
10004

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

```
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
);
```



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

PROGRAM 4: STUDENT FACULTY DATABASE

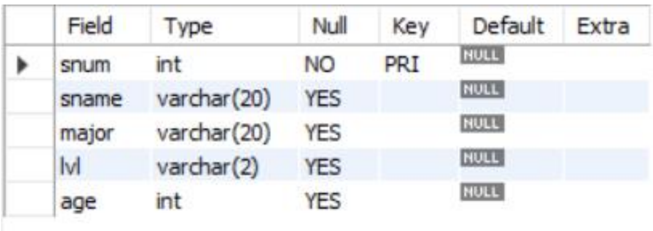
Consider the following database for student enrolment for course:
STUDENT(snum: integer, sname: string, major: string, lvl: string, age: integer)
CLASS(cname: string, meetsat: time, room: string, fid: integer)
ENROLLED(snum: integer, cname: string)
FACULTY(fid: integer, fname: string, deptid: integer)

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.

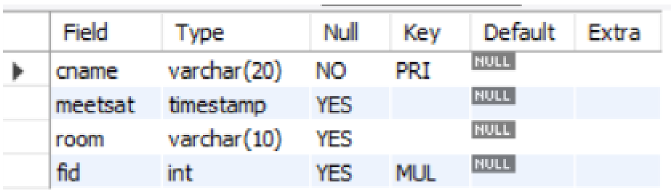
Creation of Tables:

```
CREATE TABLE student (
        snum int,
   sname varchar(20),
   major varchar(20),
   lvl varchar(2),
   age int,
   primary key(snum)
);
```



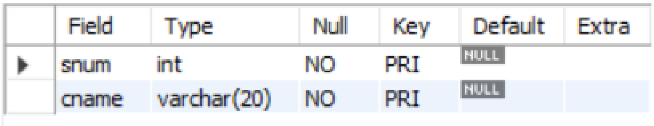
Field	Type	Null	Key	Default	Extra
snum	int	NO	PRI	NULL	
sname	varchar(20)	YES		NULL	
major	varchar(20)	YES		NULL	
lvl	varchar(2)	YES		NULL	
age	int	YES		NULL	

```
CREATE
TABLE class (
        cname varchar(20),
   meetsat timestamp,
   room varchar(10),
   fid int,
   primary key(cname),
   foreign key(fid) references faculty(fid)
);
```



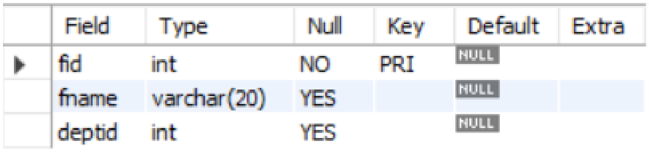
Field	Type	Null	Key	Default	Extra
cname	varchar(20)	NO	PRI	NULL	
meetsat	timestamp	YES		NULL	
room	varchar(10)	YES		NULL	
fid	int	YES	MUL	NULL	

```
CREATE TABLE
enrolled (
        snum int,
   cname varchar(20),
   primary key(snum, cname),
   foreign key (snum) references student(snum),
    foreign key (cname) references class(cname)
);
```



Field	Type	Null	Key	Default	Extra
snum	int	NO	PRI	NULL	
cname	varchar(20)	NO	PRI	NULL	

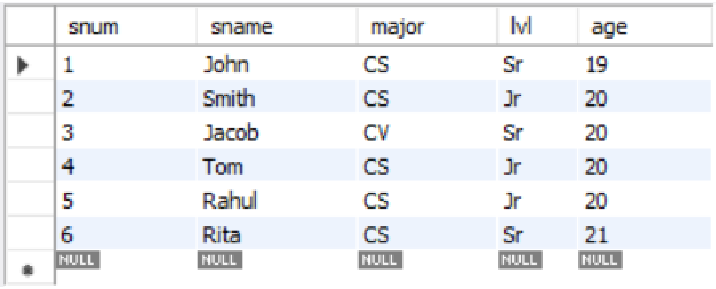
```
CREATE
TABLE faculty (
        fid int,
    fname varchar(20),
    deptid int,
    primary key(fid)
);
```



Field	Type	Null	Key	Default	Extra
fid	int	NO	PRI	NULL	
fname	varchar(20)	YES		NULL	
deptid	int	YES		NULL	

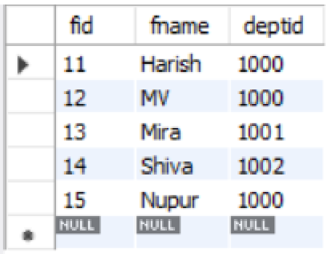
Inserting Values into the tables:

```
INSERT INTO student VALUES (1, 'John', '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);
```



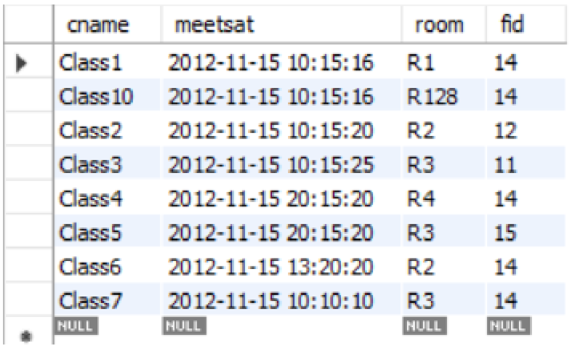
snum	sname	major	lvl	age
1	John	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Rita	CS	Sr	21
NULL	NULL	NULL	NULL	NULL

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



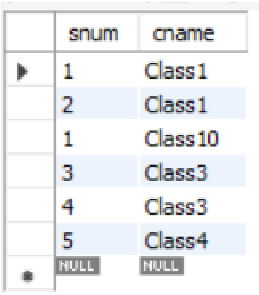
fid	fname	deptid
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000
NULL	NULL	NULL

```
INSERT INTO class VALUES ('Class1', '12/11/15 10:15:16.00000', 'R1', 14);
INSERT INTO class VALUES ('Class10', '12/11/15 10:15:16.00000', 'R128', 14);
INSERT INTO class VALUES ('Class2', '12/11/15 10:15:20.000000', 'R2', 12);
INSERT INTO class VALUES ('Class3', '12/11/15 10:15:25.000000', 'R3', 11);
INSERT INTO class VALUES ('Class4', '12/11/15 20:15:20.000000', 'R4', 14);
INSERT INTO class VALUES ('Class5', '12/11/15 20:15:20.000000', 'R3', 15);
INSERT INTO class VALUES ('Class6', '12/11/15 13:20:20.000000', 'R2', 14);
INSERT INTO class VALUES ('Class7', '12/11/15 10:10:10.000000', 'R3', 14);
```



cname	meetsat	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

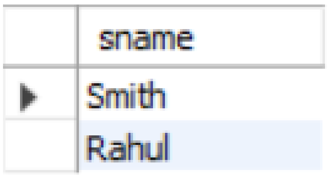
```
INSERT INTO enrolled VALUES (1, 'Class1');
INSERT INTO enrolled VALUES (2, 'Class1');
INSERT INTO enrolled VALUES (3, 'Class3');
INSERT INTO enrolled VALUES (4, 'Class3');
INSERT INTO enrolled VALUES (5, 'Class4');
```



snum	cname
1	Class1
2	Class1
1	Class10
3	Class3
4	Class3
5	Class4
NULL	NULL

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

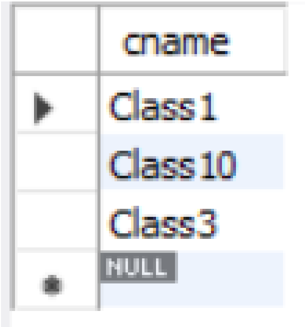
```
SELECT s.sname
FROM student s, enrolled e, class c, faculty f
WHERE s.lvl = 'Jr'
AND s.snum = e.snum
AND c.cname = e.cname
AND c.fid = f.fid
AND f.fname = 'Shiva';
```



```
sname
Smith
Rahul
```

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

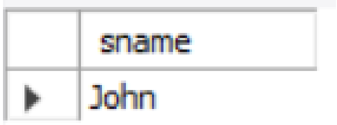
```
SELECT c.cname
FROM class c
WHERE c.room = 'R128'
OR c.cname
IN (
SELECT e.cname
FROM enrolled e
GROUP BY e.cname
HAVING COUNT(e.cname) >= 2
);
```



	cname
▶	Class1
	Class10
	Class3
*	NULL

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

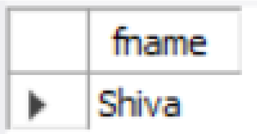
```
SELECT s.sname
FROM student s
WHERE s.snum IN (
SELECT e1.snum
FROM enrolled e1, enrolled e2, class c1, class c2
WHERE e1.snum = e2.snum
AND e1.cname <> e2.cname
AND e1.cname = c1.cname
AND c1.meetsat = c2.meetsat
);
```



	sname
▶	John

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

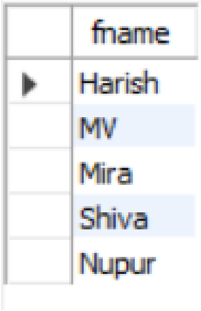
```
SELECT DISTINCT f.fname
FROM faculty f, class c
WHERE f.fid
IN (
SELECT fid
FROM class c
GROUP BY fid
HAVING COUNT(*) = (
SELECT COUNT(DISTINCT room)
FROM class
)
);
```



fname
Shiva

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

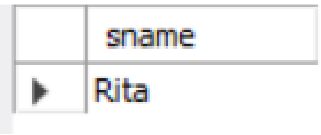
```
SELECT f.fname
FROM faculty f
WHERE 5 > (
SELECT COUNT(e.snum)
FROM class c, enrolled e
WHERE c.cname = e.cname
AND c.fid = f.fid
);
```



fname
Harish
MV
Mira
Shiva
Nupur

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

```
SELECT sname
FROM student
WHERE snum NOT IN (
SELECT e.snum
FROM enrolled e
);
```

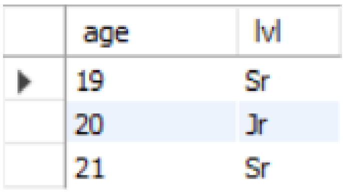


sname
Rita

7. **For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR**

level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

```
SELECT s.age, s.lvl
FROM student s
GROUP BY s.age, s.lvl
HAVING s.lvl IN (
        SELECT s1.lvl FROM student s1
        WHERE s1.age = s.age
        GROUP BY s1.lvl, s1.age
        HAVING COUNT(*) >= ALL (
                SELECT COUNT(*)
                FROM Student s2
                WHERE s1.age = s2.age
                GROUP BY s2.lvl, s2.age
   )
);
```



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)

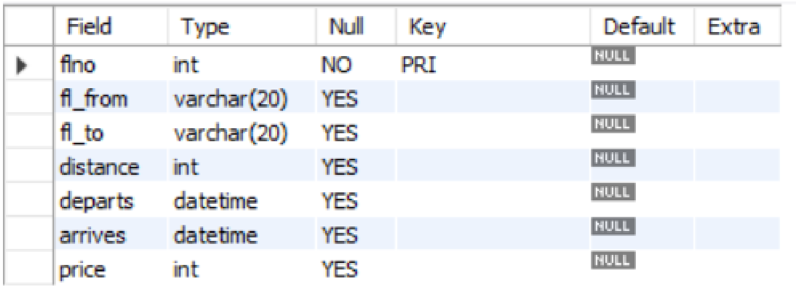
CERTIFIED(eid: integer, aid: integer)

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.

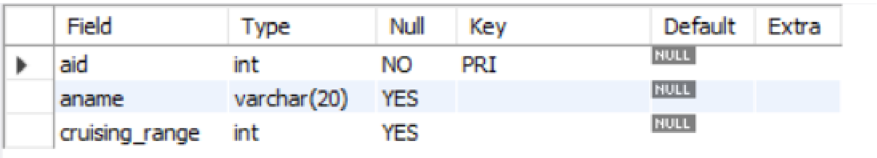
Creation of Tables:

```
CREATE TABLE flights(
        flno INT,
   fl_from VARCHAR(20),
   fl_to VARCHAR(20),
   distance INT,
   departs DATETIME,
   arrives DATETIME,
   price INT,
   PRIMARY KEY(flno)
);
```



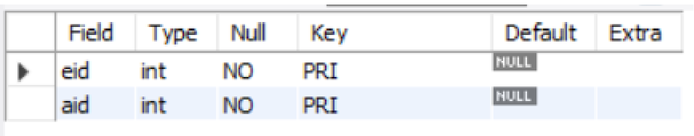
	Field	Type	Null	Key	Default	Extra
▶	flno	int	NO	PRI	NULL	
	fl_from	varchar(20)	YES		NULL	
	fl_to	varchar(20)	YES		NULL	
	distance	int	YES		NULL	
	departs	datetime	YES		NULL	
	arrives	datetime	YES		NULL	
	price	int	YES		NULL	

```
CREATE TABLE aircraft (
        aid INT,
    aname VARCHAR(20),
    cruising_range INT,
    PRIMARY KEY(aid)
);
```



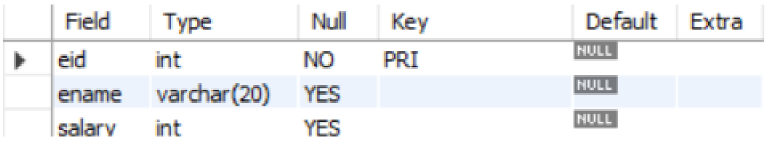
	Field	Type	Null	Key	Default	Extra
▶	aid	int	NO	PRI	NULL	
	aname	varchar(20)	YES		NULL	
	cruising_range	int	YES		NULL	

```
CREATE TABLE certified (
        eid INT,
    aid INT,
    PRIMARY KEY(eid, aid),
    FOREIGN KEY (eid) REFERENCES employees(eid),
    FOREIGN KEY(aid) REFERENCES aircraft(aid)
);
```



	Field	Type	Null	Key	Default	Extra
▶	eid	int	NO	PRI	NULL	
	aid	int	NO	PRI	NULL	

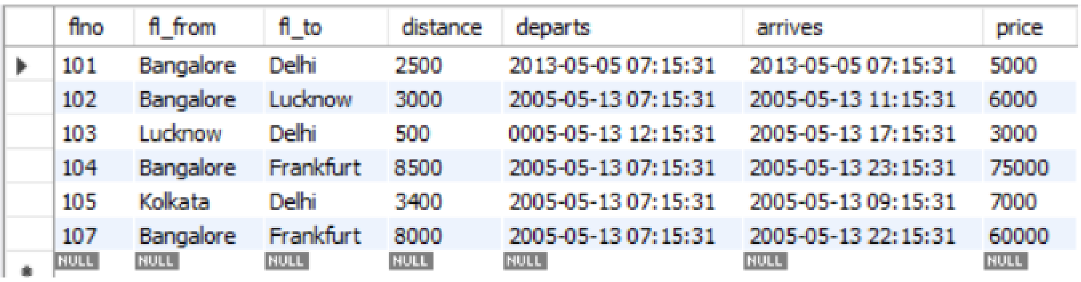
```
CREATE TABLE employees (
        eid INT,
    ename VARCHAR(20),
    salary INT,
    PRIMARY KEY(eid)
);
```



	Field	Type	Null	Key	Default	Extra
▶	eid	int	NO	PRI	NULL	
	ename	varchar(20)	YES		NULL	
	salary	int	YES		NULL	

Inserting Values into the tables:

```
INSERT INTO flights VALUES (101, 'Bangalore', 'Delhi', 2500, '13-05-05 07.15.31.000000', '13-05-05 07.15.31.000000', 5000);
INSERT INTO flights VALUES (102, 'Bangalore', 'Lucknow', 3000, '05/05/13 07:15:31', '05/05/13 11:15:31', 6000);
INSERT INTO flights VALUES (103, 'Lucknow', 'Delhi', 500, '5/05/13 12:15:31', '05/05/13 17:15:31', 3000);
INSERT INTO flights VALUES (107, 'Bangalore', 'Frankfurt', 8000, '05/05/13 07:15:31', '05/05/13 22:15:31', 60000);
INSERT INTO flights VALUES (104, 'Bangalore', 'Frankfurt', 8500, '05/05/13 07:15:31', '05/05/13 23:15:31', 75000);
INSERT INTO flights VALUES (105, 'Kolkata', 'Delhi', 3400, '05/05/13 07:15:31', ' 05/05/13 09:15:31', 7000);
```



fino	fl_from	fl_to	distance	departs	arrives	price
101	Bangalore	Delhi	2500	2013-05-05 07:15:31	2013-05-05 07:15:31	5000
102	Bangalore	Lucknow	3000	2005-05-13 07:15:31	2005-05-13 11:15:31	6000
103	Lucknow	Delhi	500	0005-05-13 12:15:31	2005-05-13 17:15:31	3000
104	Bangalore	Frankfurt	8500	2005-05-13 07:15:31	2005-05-13 23:15:31	75000
105	Kolkata	Delhi	3400	2005-05-13 07:15:31	2005-05-13 09:15:31	7000
107	Bangalore	Frankfurt	8000	2005-05-13 07:15:31	2005-05-13 22:15:31	60000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

INSERT INTO aircraft VALUES (101, '747', 3000);

INSERT INTO aircraft VALUES (102, 'Boeing', 900);

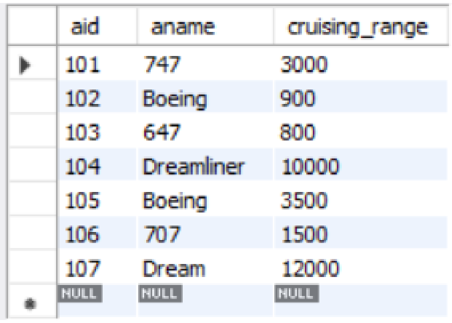
INSERT INTO aircraft VALUES (103, '647', 800);

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', 12000);



aid	aname	cruising_range
101	747	3000
102	Boeing	900
103	647	800
104	Dreamliner	10000
105	Boeing	3500
106	707	1500
107	Dream	12000
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);

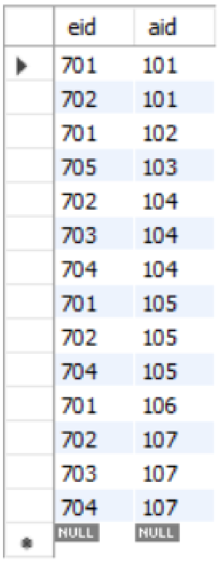
INSERT INTO certified VALUES (703, 104);

INSERT INTO certified VALUES (704, 104);

INSERT INTO certified VALUES (702, 107);

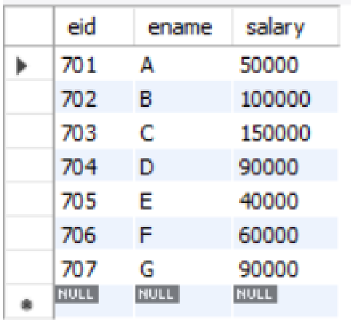
INSERT INTO certified VALUES (703, 107);

```
INSERT INTO certified VALUES (704, 107);

INSERT INTO certified VALUES (702, 101);

INSERT INTO certified VALUES (702, 105);

INSERT INTO certified VALUES (704, 105);

INSERT INTO certified VALUES (705, 103);
```



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

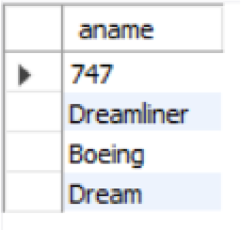
```
INSERT INTO employees VALUES (701, 'A', 50000);

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);
```



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

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

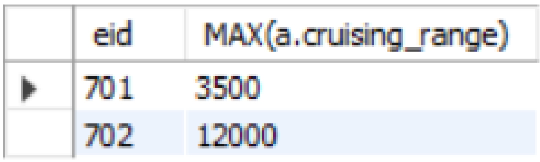
```
SELECT DISTINCT a.aname
FROM aircraft a, certified c, employees e
WHERE a.aid = c.aid
AND c.eid = e.eid
AND e.salary>80000;
```



aname
747
Dreamliner
Boeing
Dream

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

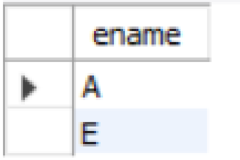
```
SELECT c.eid, MAX(a.cruising_range)
FROM aircraft a, certified c, employees e
WHERE e.eid = c.eid
AND a.aid = c.aid
GROUP BY c.eid
HAVING COUNT(*) > 3;
```



eid	MAX(a.cruising_range)
701	3500
702	12000

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

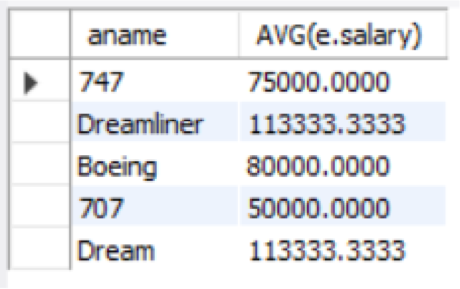
```
SELECT e.ename
FROM employees e
WHERE e.salary < (
 SELECT MIN(f.price)
 FROM flights f
 WHERE f.fl_from = 'Bangalore'
 AND f.fl_to = 'Frankfurt'
);
```



	ename
▶	A
	E

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

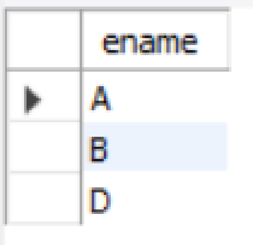
```
SELECT a.aname, AVG(e.salary)
FROM aircraft a, certified c, employees e
WHERE a.cruising_range > 1000
AND a.aid = c.aid
AND e.eid = c.eid
GROUP BY a.aname;
```



	aname	AVG(e.salary)
▶	747	75000.0000
	Dreamliner	113333.3333
	Boeing	80000.0000
	707	50000.0000
	Dream	113333.3333

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

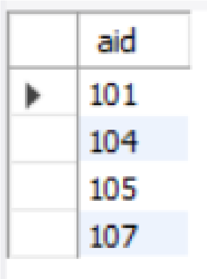
```
SELECT DISTINCT e.ename
FROM employees e, aircraft a, certified c
WHERE e.eid = c.eid
AND a.aid = c.aid
AND a.aname = 'Boeing';
```



	ename
▶	A
	B
	D

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

```
SELECT a.aid
FROM aircraft a, flights f
WHERE a.cruising_range >= f.distance
AND f.fl_from = 'Bangalore'
AND f.fl_to = 'Delhi';
```



aid
101
104
105
107

7. **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.**