**LIBRARY DATABASE**

Write SQL queries to

* 1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
  2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017
  3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
  4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
  5. Create a view of all books and its number of copies that are currently available in the Library.

CREATE TABLE PUBLISHER (

name varchar(20) primary key,

address varchar(30),

phone integer

);

CREATE TABLE BOOK (

book\_id integer primary key,

title varchar(20),

pub\_name varchar(20),

pub\_year integer,

foreign key(pub\_name) references PUBLISHER(name) on delete cascade

);

CREATE TABLE AUTHORS (

book\_id integer,

auth\_name varchar(20),

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

primary key(book\_id)

);

CREATE TABLE LIBRARY\_BRANCH (

branch\_id integer primary key,

branch\_name varchar(20),

address varchar(20)

);

CREATE TABLE BOOK\_COPIES (

book\_id integer,

branch\_id integer,

no\_of\_copies integer,

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

foreign key(branch\_id) references LIBRARY\_BRANCH(branch\_id) on delete cascade ,

primary key(book\_id , branch\_id)

);

CREATE TABLE BOOK\_LENDING (

book\_id integer,

branch\_id integer,

card\_no integer,

date\_out date,

due\_date date,

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

foreign key(branch\_id) references LIBRARY\_BRANCH(branch\_id) on delete cascade ,

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

);

INSERT INTO PUBLISHER VALUES ('abc pub' , 'mumbai' , 9812564738);

INSERT INTO PUBLISHER VALUES ('def pub' , 'karnataka' , 9812564738);

INSERT INTO PUBLISHER VALUES ('ghi pub' , 'chennai' , 9812564738);

INSERT INTO PUBLISHER VALUES ('jkl pub' , 'hyderabad' , 9812564738);

INSERT INTO PUBLISHER VALUES ('mno pub' , 'goa' , 9812564738);

SELECT \* FROM PUBLISHER;

INSERT INTO BOOK VALUES (1 , 'python' , 'abc\_pub' , 1995);

INSERT INTO BOOK VALUES (2 , 'cpp' , 'jkl\_pub' , 1997);

INSERT INTO BOOK VALUES (3 , 'c' , 'mno\_pub' , 1998);

INSERT INTO BOOK VALUES (4 , 'accounts' , 'def\_pub' , 1996);

INSERT INTO BOOK VALUES (5 , 'economics' , 'jkl\_pub' , 1995);

SELECT \* FROM BOOK;

INSERT INTO AUTHORS VALUES (1 , 'ap prakash');

INSERT INTO AUTHORS VALUES (2 , 'bb taylor');

INSERT INTO AUTHORS VALUES (3 , 'kavitha reddy');

INSERT INTO AUTHORS VALUES (4 , 'sudhanshu sharma');

INSERT INTO AUTHORS VALUES (5 , 'akash sharma');

SELECT \* FROM AUTHORS;

INSERT INTO LIBRARY\_BRANCH VALUES (11 , 'ap' , '123 street');

INSERT INTO LIBRARY\_BRANCH VALUES (22 , 'lucknow' , '456 street');

INSERT INTO LIBRARY\_BRANCH VALUES (33 , 'goa' , '789 street');

INSERT INTO LIBRARY\_BRANCH VALUES (44 , 'karnataka' , '101 street');

INSERT INTO LIBRARY\_BRANCH VALUES (55 , 'mumbai' , '121 street');

SELECT \* FROM LIBRARY\_BRANCH;

INSERT INTO BOOK\_COPIES VALUES (1 , 22 , 10);

INSERT INTO BOOK\_COPIES VALUES (2 , 11 , 20);

INSERT INTO BOOK\_COPIES VALUES (3 , 55 , 5);

INSERT INTO BOOK\_COPIES VALUES (4 , 33 , 15);

INSERT INTO BOOK\_COPIES VALUES (5 , 44 , 25);

SELECT \* FROM BOOK\_COPIES;

INSERT INTO BOOK\_LENDING VALUES (1 , 22 , 111 , '10-jan-2020' , '25-jan-2020');

INSERT INTO BOOK\_LENDING VALUES (2 , 44 , 112 , '23-jun-2020' , '08-jul-2020');

INSERT INTO BOOK\_LENDING VALUES (4 , 55 , 113 , '20-jul-2020' , '04-aug-2020');

INSERT INTO BOOK\_LENDING VALUES (5 , 11 , 114 , '05-aug-2020' , '20-aug-2020');

INSERT INTO BOOK\_LENDING VALUES (3 , 33 , 115 , '15-dec-2020' , '30-dec-2020');

SELECT \* FROM BOOK\_LENDING;

SELECT lb.branch\_name, b.book\_id , b.title , b.pub\_name , a.auth\_name , bc.no\_of\_copies FROM LIBRARY\_BRANCH lb , BOOK b , AUTHORS a , BOOK\_COPIES bc WHERE b.book\_id = a.book\_id AND a.book\_id = bc.book\_id AND bc.branch\_id = lb.branch\_id GROUP BY lb.branch\_name , b.book\_id , b.title , b.pub\_name , a.auth\_name , bc.no\_of\_copies;

INSERT INTO BOOK\_LENDING VALUES (2 , 33 , 115 , '02-jan-2020' , '17-jan-2020');

INSERT INTO BOOK\_LENDING VALUES (1 , 33 , 115 , '17-mar-2020' , '29-jan-2020');

INSERT INTO BOOK\_LENDING VALUES (4 , 33 , 115 , '15-apr-2020' , '30-apr-2020');

SELECT card\_no FROM BOOK\_LENDING WHERE date\_out BETWEEN '01-jan-2020' AND '30-jun-2020' GROUP BY card\_no HAVING COUNT(\*) > 3;

#extra

SELECT book\_id , title , pub\_name , pub\_year FROM BOOK GROUP BY pub\_year , book\_id , title , pub\_name;

#extra ends

SELECT \* FROM BOOK GROUP BY pub\_year , title;

CREATE VIEW available\_books as SELECT b.book\_id , b.title , bc.no\_of\_copies FROM BOOK b , BOOK\_COPIES bc , LIBRARY\_BRANCH lb WHERE b.book\_id = bc.book\_id AND lb.branch\_id = bc.branch\_id;

SELECT \* FROM available\_books;

DELETE FROM BOOK WHERE book\_id = 5;

SELECT \* FROM BOOK;

**OUTPUT**:

abc pub|mumbai|9812564738

def pub|karnataka|9812564738

ghi pub|chennai|9812564738

jkl pub|hyderabad|9812564738

mno pub|goa|9812564738

1|python|abc\_pub|1995

2|cpp|jkl\_pub|1997

3|c|mno\_pub|1998

4|accounts|def\_pub|1996

5|economics|jkl\_pub|1995

1|ap prakash

2|bb taylor

3|kavitha reddy

4|sudhanshu sharma

5|akash sharma

11|ap|123 street

22|lucknow|456 street

33|goa|789 street

44|karnataka|101 street

55|mumbai|121 street

1|22|10

2|11|20

3|55|5

4|33|15

5|44|25

1|22|111|10-jan-2020|25-jan-2020

2|44|112|23-jun-2020|08-jul-2020

4|55|113|20-jul-2020|04-aug-2020

5|11|114|05-aug-2020|20-aug-2020

3|33|115|15-dec-2020|30-dec-2020

ap|2|cpp|jkl\_pub|bb taylor|20

goa|4|accounts|def\_pub|sudhanshu sharma|15

karnataka|5|economics|jkl\_pub|akash sharma|25

lucknow|1|python|abc\_pub|ap prakash|10

mumbai|3|c|mno\_pub|kavitha reddy|5

115

#extra

1|python|abc\_pub|1995

2|cpp|jkl\_pub|1997

3|c|mno\_pub|1998

4|accounts|def\_pub|1996

5|economics|jkl\_pub|1995

#extra ends

5|economics|jkl\_pub|1995

1|python|abc\_pub|1995

4|accounts|def\_pub|1996

2|cpp|jkl\_pub|1997

3|c|mno\_pub|1998

1|python|10

2|cpp|20

3|c|5

4|accounts|15

5|economics|25

1|python|abc\_pub|1995

2|cpp|jkl\_pub|1997

3|c|mno\_pub|1998

4|accounts|def\_pub|1996

**ORDER DATABASE**

Write SQL queries to

1. Count the customers with grades above Bangalore’s average.

2. Find the name and numbers of all salesmen who had more than one customer.

3. List all salesmen and indicate those who have and don’t have customers in their

cities (Use UNION operation.)

4. Create a view that finds the salesman who has the customer with the highest

order of a day.

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his

orders must also be deleted.

CREATE TABLE SALESMAN (

sid integer primary key,

name varchar(30),

city varchar(10),

commission decimal(8 , 2)

);

CREATE TABLE CUSTOMER (

cid integer primary key,

cname varchar(30),

city varchar(10),

grade integer,

sid integer,

foreign key(sid) references SALESMAN(sid) on delete set NULL

);

CREATE TABLE ORDERS (

ono integer,

pur\_amt decimal(8 , 2),

o\_date date,

cid integer,

sid integer,

foreign key(cid) references CUSTOMER(cid),

foreign key(sid) references SALESMAN(sid) on delete cascade,

primary key(cid , sid , ono)

);

INSERT INTO SALESMAN VALUES (1 , 'abc' , 'mumbai' , 100);

INSERT INTO SALESMAN VALUES (2 , 'def' , 'pune' , 150);

INSERT INTO SALESMAN VALUES (3 , 'ghi' , 'hyderabad' , 200);

INSERT INTO SALESMAN VALUES (4 , 'jkl' , 'blore' , 50);

INSERT INTO SALESMAN VALUES (5 , 'mno' , 'goa' , 70);

SELECT \* FROM SALESMAN;

INSERT INTO CUSTOMER VALUES (11 , 'xyz' , 'pune' , 10 , 2);

INSERT INTO CUSTOMER VALUES (12 , 'rst' , 'blore' , 15 , 4);

INSERT INTO CUSTOMER VALUES (13 , 'pqr' , 'kpur' , 9 , 1);

INSERT INTO CUSTOMER VALUES (14 , 'uvw' , 'hyderabad' , 5 , 3);

INSERT INTO CUSTOMER VALUES (15 , 'vwx' , 'seoul' , 14 , 5);

SELECT \* FROM CUSTOMER;

INSERT INTO ORDERS VALUES (111 , 10567 , '10-jan-2020' , 11 , 2);

INSERT INTO ORDERS VALUES (112 , 78564 , '12-jun-2020' , 33 , 5);

INSERT INTO ORDERS VALUES (113 , 12989 , '10-apr-2020' , 44 , 4);

INSERT INTO ORDERS VALUES (114 , 24366 , '10-sep-2020' , 22 , 3);

INSERT INTO ORDERS VALUES (115 , 10990 , '10-dec-2020' , 55 , 1);

SELECT \* FROM ORDERS;

INSERT INTO CUSTOMER VALUES (16 , 'vwxx' , 'seoul' , 7 , 5);

INSERT INTO CUSTOMER VALUES (17 , 'vwxw' , 'seoul' , 6 , 1);

INSERT INTO CUSTOMER VALUES (18 , 'vwxv' , 'seoul' , 2 , 1);

SELECT \* FROM CUSTOMER;

SELECT COUNT(cid) FROM CUSTOMER WHERE grade > (SELECT avg(grade) FROM CUSTOMER WHERE city = "seoul");

SELECT s.name , COUNT(c.cid) FROM SALESMAN S , CUSTOMER C WHERE c.sid = s.sid GROUP BY name HAVING COUNT(cid) > 1;

SELECT name FROM CUSTOMER c , SALESMAN s WHERE s.sid = c.cid AND s.city = c.city UNION SELECT name from SALESMAN WHERE sid NOT IN (SELECT name FROM CUSTOMER c1 , SALESMAN s1 WHERE s1.sid = c1.cid AND s1.city = c1.city);

INSERT INTO ORDERS VALUES (116 , 10990 , '01-jan-2020' , 55 , 1);

INSERT INTO ORDERS VALUES (117 , 11990 , '01-jan-2020' , 44 , 2);

INSERT INTO ORDERS VALUES (118 , 20990 , '01-jan-2020' , 33 , 3);

INSERT INTO ORDERS VALUES (119 , 40990 , '01-jan-2020' , 22 , 4);

INSERT INTO ORDERS VALUES (120 , 13990 , '01-jan-2020' , 11 , 5);

CREATE VIEW high\_order as SELECT sid , pur\_amt FROM ORDERS WHERE pur\_amt = (SELECT max(pur\_amt) FROM ORDERS O WHERE o\_date = '01-jan-2020');

SELECT \* FROM high\_order;

DELETE FROM SALESMAN WHERE sid = 1;

SELECT \* FROM SALESMAN;

SELECT \* FROM CUSTOMER;

SELECT \* FROM ORDERS;

**OUTPUT:**

1|abc|mumbai|100

2|def|pune|150

3|ghi|hyderabad|200

4|jkl|blore|50

5|mno|goa|70

11|xyz|pune|10|2

12|rst|blore|15|4

13|pqr|kpur|9|1

14|uvw|hyderabad|5|3

15|vwx|seoul|14|5

111|10567|10-jan-2020|11|2

112|78564|12-jun-2020|33|5

113|12989|10-apr-2020|44|4

114|24366|10-sep-2020|22|3

115|10990|10-dec-2020|55|1

11|xyz|pune|10|2

12|rst|blore|15|4

13|pqr|kpur|9|1

14|uvw|hyderabad|5|3

15|vwx|seoul|14|5

16|vwxx|seoul|7|5

17|vwxw|seoul|6|1

18|vwxv|seoul|2|1

4

abc|3

mno|2

abc

def

ghi

jkl

mno

4|40990

2|def|pune|150

3|ghi|hyderabad|200

4|jkl|blore|50

5|mno|goa|70

11|xyz|pune|10|2

12|rst|blore|15|4

13|pqr|kpur|9|1

14|uvw|hyderabad|5|3

15|vwx|seoul|14|5

16|vwxx|seoul|7|5

17|vwxw|seoul|6|1

18|vwxv|seoul|2|1

111|10567|10-jan-2020|11|2

112|78564|12-jun-2020|33|5

113|12989|10-apr-2020|44|4

114|24366|10-sep-2020|22|3

115|10990|10-dec-2020|55|1

116|10990|01-jan-2020|55|1

117|11990|01-jan-2020|44|2

118|20990|01-jan-2020|33|3

119|40990|01-jan-2020|22|4

120|13990|01-jan-2020|11|5

**AIRLINE DATABASE**

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.

CREATE TABLE FLIGHTS (

num integer PRIMARY KEY,

fplace varchar(10),

tplace varchar(10),

dist integer,

departs varchar(10),

arrives varchar(10),

price real

);

CREATE TABLE AIRCRAFT (

aid integer PRIMARY KEY,

aname varchar(10),

crange integer

);

CREATE TABLE EMPLOYEES (

eid integer PRIMARY KEY,

ename varchar(10),

salary integer

);

CREATE TABLE CERTIFIED (

aid integer,

eid integer,

FOREIGN KEY(aid) REFERENCES AIRCRAFT(aid),

FOREIGN KEY(eid) REFERENCES EMPLOYEES(eid),

PRIMARY KEY(aid , eid)

);

INSERT INTO FLIGHTS VALUES (1 , 'blore' , 'mlore' , 12000 , '18:00' , '23:00' , 7000);

INSERT INTO FLIGHTS VALUES (2 , 'mlore' , 'chennai' , 72000 , '15:10' , '20:00' , 27000);

INSERT INTO FLIGHTS VALUES (3 , 'busan' , 'seoul' , 45787 , '00:00' , '03:00' , 7000);

INSERT INTO FLIGHTS VALUES (4 , 'seoul' , 'daegu' , 13990 , '06:00' , '08:50' , 7000);

INSERT INTO FLIGHTS VALUES (5 , 'daegu' , 'busan' , 54000 , '06:00' , '10:00' , 7000);

SELECT \* FROM FLIGHTS;

INSERT INTO AIRCRAFT VALUES (11 , 'abc al' , 3000);

INSERT INTO AIRCRAFT VALUES (22 , 'def al' , 6000);

INSERT INTO AIRCRAFT VALUES (33 , 'ghi al' , 2000);

INSERT INTO AIRCRAFT VALUES (44 , 'jkl al' , 8000);

INSERT INTO AIRCRAFT VALUES (55 , 'mno al' , 6000);

SELECT \* FROM AIRCRAFT;

INSERT INTO EMPLOYEES VALUES (111 , 'xyz' , 30000);

INSERT INTO EMPLOYEES VALUES (112 , 'uvw' , 90000);

INSERT INTO EMPLOYEES VALUES (113 , 'rst' , 60800);

INSERT INTO EMPLOYEES VALUES (114 , 'opq' , 28900);

INSERT INTO EMPLOYEES VALUES (115 , 'lmn' , 12000);

SELECT \* FROM EMPLOYEES;

INSERT INTO CERTIFIED VALUES (11 , 111);

INSERT INTO CERTIFIED VALUES (11 , 112);

INSERT INTO CERTIFIED VALUES (11 , 113);

INSERT INTO CERTIFIED VALUES (11 , 114);

INSERT INTO CERTIFIED VALUES (11 , 115);

INSERT INTO CERTIFIED VALUES (22 , 112);

INSERT INTO CERTIFIED VALUES (33 , 115);

INSERT INTO CERTIFIED VALUES (22 , 113);

INSERT INTO CERTIFIED VALUES (44 , 114);

SELECT \* FROM CERTIFIED;

SELECT DISTINCT a.aname FROM AIRCRAFT a , EMPLOYEES e , CERTIFIED c WHERE c.aid = a.aid AND e.eid = c.eid AND a.aname NOT IN (SELECT a1.aname FROM AIRCRAFT a1 , EMPLOYEES e1 , CERTIFIED c1 WHERE c1.aid = a1.aid AND e1.eid = c1.eid AND e1.salary < 60000);

INSERT INTO CERTIFIED VALUES (44 , 115);

INSERT INTO CERTIFIED VALUES (55 , 115);

INSERT INTO CERTIFIED VALUES (22 , 115);

SELECT c.eid , max(a.crange) FROM AIRCRAFT a , CERTIFIED c WHERE a.aid = c.aid GROUP BY c.eid HAVING COUNT(\*) > 3;

INSERT INTO FLIGHTS VALUES (6 , 'blore' , 'frankfurt' , 12900 , '18:00' , '23:00' , 27000);

INSERT INTO FLIGHTS VALUES (7 , 'blore' , 'frankfurt' , 19000 , '18:00' , '03:00' , 17000);

INSERT INTO FLIGHTS VALUES (8 , 'blore' , 'frankfurt' , 10000 , '18:00' , '01:00' , 23000);

SELECT \* FROM FLIGHTS;

INSERT INTO EMPLOYEES VALUES (116 , 'lmnn' , 12000);

INSERT INTO EMPLOYEES VALUES (117 , 'lmnm' , 10000);

INSERT INTO EMPLOYEES VALUES (118 , 'lmnl' , 8000);

SELECT \* FROM EMPLOYEES;

SELECT DISTINCT ename FROM EMPLOYEES WHERE salary < (SELECT min(price) FROM FLIGHTS WHERE fplace = "blore" AND tplace = "frankfurt");

SELECT a.aname , avg(e.salary) FROM EMPLOYEES e , AIRCRAFT a , CERTIFIED c WHERE a.crange > 5000 AND c.aid = a.aid AND c.eid = e.eid GROUP BY a.aid;

SELECT DISTINCT e.ename FROM AIRCRAFT a , CERTIFIED c , EMPLOYEES e WHERE a.aname = 'def al' AND a.aid = c.aid AND c.eid = e.eid;

INSERT INTO AIRCRAFT VALUES (66 , 'mnoo al' , 15000);

INSERT INTO AIRCRAFT VALUES (77 , 'mnon al' , 14000);

INSERT INTO AIRCRAFT VALUES (88 , 'mnom al' , 12000);

SELECT a.aid FROM AIRCRAFT a WHERE a.crange > (SELECT min(f.dist) FROM FLIGHTS f WHERE fplace = 'blore' AND tplace = 'frankfurt');

**OUTPUT:**

1|blore|mlore|12000|18:00|23:00|7000.0

2|mlore|chennai|72000|15:10|20:00|27000.0

3|busan|seoul|45787|00:00|03:00|7000.0

4|seoul|daegu|13990|06:00|08:50|7000.0

5|daegu|busan|54000|06:00|10:00|7000.0

11|abc al|3000

22|def al|6000

33|ghi al|2000

44|jkl al|8000

55|mno al|6000

111|xyz|30000

112|uvw|90000

113|rst|60800

114|opq|28900

115|lmn|12000

11|111

11|112

11|113

11|114

11|115

22|112

33|115

22|113

44|114

def al

115|8000

1|blore|mlore|12000|18:00|23:00|7000.0

2|mlore|chennai|72000|15:10|20:00|27000.0

3|busan|seoul|45787|00:00|03:00|7000.0

4|seoul|daegu|13990|06:00|08:50|7000.0

5|daegu|busan|54000|06:00|10:00|7000.0

6|blore|frankfurt|12900|18:00|23:00|27000.0

7|blore|frankfurt|19000|18:00|03:00|17000.0

8|blore|frankfurt|10000|18:00|01:00|23000.0

111|xyz|30000

112|uvw|90000

113|rst|60800

114|opq|28900

115|lmn|12000

116|lmnn|12000

117|lmnm|10000

118|lmnl|8000

lmn

lmnn

lmnm

lmnl

def al|54266.6666666667

jkl al|20450.0

mno al|12000.0

uvw

rst

lmn

66

77

88

**MOVIE DATABASE**

Write SQL queries to

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

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

3. List all actors who acted in a movie before 2000 and also in a movie

after 2015 (use JOIN operation).

4. Find the title of movies and number of stars for each movie that has at least one

rating and find the highest number of stars that movie received. Sort the result by

movie title.

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

CREATE TABLE ACTOR (

act\_id integer PRIMARY KEY,

act\_name varchar(20),

act\_gender varchar(2)

);

CREATE TABLE DIRECTOR (

dir\_id integer PRIMARY KEY,

dir\_name varchar(20),

dir\_phone decimal(10,0)

);

CREATE TABLE MOVIES (

mov\_id integer PRIMARY KEY,

mov\_title varchar(20),

mov\_year decimal(4,0),

mov\_lang varchar(10),

dir\_id integer,

FOREIGN KEY(dir\_id) REFERENCES DIRECTOR(dir\_id)

);

CREATE TABLE MOVIE\_CAST (

mov\_id integer,

act\_id integer,

role varchar(10),

FOREIGN KEY(act\_id) REFERENCES ACTOR(act\_id),

FOREIGN KEY(mov\_id) REFERENCES MOVIES(mov\_id)

);

CREATE TABLE RATING (

mov\_id integer,

stars integer,

FOREIGN KEY(mov\_id) REFERENCES MOVIES(mov\_id)

);

INSERT INTO ACTOR VALUES (1 , 'srk' , 'm');

INSERT INTO ACTOR VALUES (2 , 'ben' , 'm');

INSERT INTO ACTOR VALUES (3 , 'anushka' , 'f');

INSERT INTO ACTOR VALUES (4 , 'kajol' , 'f');

INSERT INTO ACTOR VALUES (5 , 'deeps' , 'f');

SELECT \* FROM ACTOR;

INSERT INTO DIRECTOR VALUES (11 , 'abc' , 1290875644);

INSERT INTO DIRECTOR VALUES (22 , 'def' , 9867678900);

INSERT INTO DIRECTOR VALUES (33 , 'ghi' , 9845673322);

INSERT INTO DIRECTOR VALUES (44 , 'jkl' , 9923456788);

INSERT INTO DIRECTOR VALUES (55 , 'mno' , 9987654321);

SELECT \* FROM DIRECTOR;

INSERT INTO MOVIES VALUES (111 , 'hny' , 2016 , 'hindi' , 11);

INSERT INTO MOVIES VALUES (112 , 'khnk' , 2007 , 'hindi' , 22);

INSERT INTO MOVIES VALUES (113 , 'kank' , 2001 , 'hindi' , 33);

INSERT INTO MOVIES VALUES (114 , 'kkhh' , 1996 , 'hindi' , 44);

INSERT INTO MOVIES VALUES (115 , 'imitation game' , 2014 , 'hindi' , 44);

SELECT \* FROM MOVIES;

INSERT INTO MOVIE\_CAST VALUES (111 , 1 ,'hero');

INSERT INTO MOVIE\_CAST VALUES (111 , 5 ,'heroine');

INSERT INTO MOVIE\_CAST VALUES (115 , 2 ,'hero');

INSERT INTO MOVIE\_CAST VALUES (113 , 3 ,'heroine');

INSERT INTO MOVIE\_CAST VALUES (114 , 1 ,'cameo');

SELECT \* FROM MOVIE\_CAST;

INSERT INTO RATING VALUES (111 , 5);

INSERT INTO RATING VALUES (112 , 3);

INSERT INTO RATING VALUES (113 , 4);

INSERT INTO RATING VALUES (114 , 2);

INSERT INTO RATING VALUES (115 , 5);

SELECT \* FROM RATING;

INSERT INTO MOVIES VALUES (116 , 'ce' , 2014 , 'hindi' , 11);

INSERT INTO MOVIES VALUES (117 , 'dz' , 2017 , 'hindi' , 11);

SELECT m.mov\_title FROM MOVIES m , DIRECTOR d WHERE m.dir\_id = d.dir\_id AND d.dir\_name = 'abc';

SELECT DISTINCT m.mov\_title FROM MOVIES m , MOVIE\_CAST mc WHERE m.mov\_id = mc.mov\_id AND mc.act\_id IN (SELECT act\_id FROM MOVIE\_CAST GROUP BY act\_id HAVING COUNT (mov\_id) >= 2);

SELECT a.act\_name FROM ACTOR a INNER JOIN MOVIE\_CAST mc ON a.act\_id = mc.act\_id INNER JOIN MOVIES m ON mc.mov\_id = m.mov\_id WHERE m.mov\_year < 2000 AND act\_name IN (SELECT act\_name FROM ACTOR a INNER JOIN MOVIE\_CAST mc ON a.act\_id = mc.act\_id INNER JOIN MOVIES m ON mc.mov\_id = m.mov\_id WHERE m.mov\_year > 2015);

SELECT m.mov\_title , r.stars FROM MOVIES m , RATING r WHERE m.mov\_id = r.mov\_id AND r.stars >= 1 ORDER BY m.mov\_title;

INSERT INTO RATING VALUES (116 , 3);

INSERT INTO RATING VALUES (117 , 4);

UPDATE RATING SET stars = 5 WHERE mov\_id IN (SELECT mov\_id FROM MOVIES m , DIRECTOR d WHERE d.dir\_id = m.dir\_id AND d.dir\_name = 'abc');

SELECT \* FROM RATING;

**OUTPUT**:

1|srk|m

2|ben|m

3|anushka|f

4|kajol|f

5|deeps|f

11|abc|1290875644

22|def|9867678900

33|ghi|9845673322

44|jkl|9923456788

55|mno|9987654321

111|hny|2016|hindi|11

112|khnk|2007|hindi|22

113|kank|2001|hindi|33

114|kkhh|1996|hindi|44

115|imitation game|2014|hindi|44

111|1|hero

111|5|heroine

115|2|hero

113|3|heroine

114|1|cameo

111|5

112|3

113|4

114|2

115|5

hny

ce

dz

hny

kkhh

srk

hny|5

imitation game|5

kank|4

khnk|3

kkhh|2

111|5

112|3

113|4

114|2

115|5

116|5

117|5

**BANK DATABASE**

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

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

ii. Enter at least five tuples for each relation

iii. Find all the customers who have at least two accounts at the *Main* branch.

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

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

vi. Find the names of all depositors of a specific branch.

vii. Find the details of all loan holder of a specific branch.

CREATE TABLE BRANCH (

b\_name varchar(20) PRIMARY KEY,

b\_city varchar(20) ,

assets real(10,2)

);

CREATE TABLE ACCOUNT (

acc\_no integer(8) PRIMARY KEY,

b\_name varchar(20) ,

balance decimal(8,0),

FOREIGN KEY(b\_name) REFERENCES BRANCH(b\_name) ON DELETE CASCADE

);

CREATE TABLE CUSTOMER (

c\_name varchar(20) PRIMARY KEY,

c\_street varchar(20),

c\_city varchar(20)

);

CREATE TABLE DEPOSITOR (

c\_name varchar(20),

acc\_no integer(8),

PRIMARY KEY(c\_name , acc\_no),

FOREIGN KEY(c\_name) REFERENCES CUSTOMER(c\_name),

FOREIGN KEY(acc\_no) REFERENCES ACCOUNT(acc\_no) ON DELETE CASCADE

);

CREATE TABLE LOAN (

l\_num integer(4) PRIMARY KEY,

b\_name varchar(20),

amt decimal(8,0),

FOREIGN KEY(b\_name) REFERENCES BRANCH(b\_name)

);

CREATE TABLE BORROWER (

c\_name varchar(20),

l\_num integer(4),

PRIMARY KEY(c\_name , l\_num),

FOREIGN KEY(c\_name) REFERENCES CUSTOMER(c\_name),

FOREIGN KEY(l\_num) REFERENCES LOAN(l\_num)

);

INSERT INTO BRANCH VALUES ('main' , 'seoul' , 100000 );

INSERT INTO BRANCH VALUES ('sub' , 'busan' , 250000 );

INSERT INTO BRANCH VALUES ('sub1' , 'daegu' , 300000 );

INSERT INTO BRANCH VALUES ('sub2' , 'mumbai' , 550000 );

INSERT INTO BRANCH VALUES ('sub3' , 'pune' , 990000 );

SELECT \* FROM BRANCH;

INSERT INTO ACCOUNT VALUES (1 , 'main' , 1100000 );

INSERT INTO ACCOUNT VALUES (2 , 'sub' , 1200000 );

INSERT INTO ACCOUNT VALUES (3 , 'sub1' , 1300000 );

INSERT INTO ACCOUNT VALUES (4 , 'sub2' , 4100000 );

INSERT INTO ACCOUNT VALUES (5 , 'sub3' , 1500000 );

SELECT \* FROM ACCOUNT;

INSERT INTO ACCOUNT VALUES (6 , 'main' , 1000000 );

INSERT INTO ACCOUNT VALUES (7 , 'main' , 1800000 );

SELECT \* FROM ACCOUNT;

INSERT INTO CUSTOMER VALUES ('aabc' , '123 street' , 'busan' );

INSERT INTO CUSTOMER VALUES ('ddef' , '456 street' , 'seoul' );

INSERT INTO CUSTOMER VALUES ('gghi' , '789 street' , 'daegu' );

INSERT INTO CUSTOMER VALUES ('jjkl' , '234 street' , 'busan' );

INSERT INTO CUSTOMER VALUES ('mmno' , '567 street' , 'daegu' );

SELECT \* FROM CUSTOMER;

INSERT INTO DEPOSITOR VALUES ('abc' , 1 );

INSERT INTO DEPOSITOR VALUES ('def' , 1 );

INSERT INTO DEPOSITOR VALUES ('ghi' , 2 );

INSERT INTO DEPOSITOR VALUES ('jkl' , 5 );

INSERT INTO DEPOSITOR VALUES ('mno' , 3 );

SELECT \* FROM DEPOSITOR;

INSERT INTO LOAN VALUES (1111 , 'main' , 300000 );

INSERT INTO LOAN VALUES (2222 , 'sub1' , 500000 );

INSERT INTO LOAN VALUES (3333 , 'main' , 30000 );

INSERT INTO LOAN VALUES (4444 , 'sub' , 670000 );

INSERT INTO LOAN VALUES (5555 , 'sub2' , 300000 );

SELECT \* FROM LOAN;

INSERT INTO BORROWER VALUES ('aabc' , 1111);

INSERT INTO BORROWER VALUES ('ddef' , 2222);

INSERT INTO BORROWER VALUES ('gghi' , 3333);

INSERT INTO BORROWER VALUES ('jjkl' , 4444);

INSERT INTO BORROWER VALUES ('mmno' , 5555);

SELECT \* FROM BORROWER;

INSERT INTO DEPOSITOR VALUES ('abc' , 6 );

INSERT INTO DEPOSITOR VALUES ('abc' , 7 );

INSERT INTO DEPOSITOR VALUES ('def' , 7 );

INSERT INTO DEPOSITOR VALUES ('ghi' , 7 );

SELECT d.c\_name FROM DEPOSITOR d , ACCOUNT a WHERE a.acc\_no = d.acc\_no AND a.b\_name = 'main' GROUP BY d.c\_name HAVING COUNT(\*) >= 2;

INSERT INTO BRANCH VALUES ('sub4' , 'busan' , 990000 );

INSERT INTO ACCOUNT VALUES (8 , 'sub4' , 1800000 );

INSERT INTO DEPOSITOR VALUES ('ghi' , 8 );

SELECT d.c\_name FROM DEPOSITOR d , ACCOUNT a , BRANCH b WHERE a.acc\_no = d.acc\_no AND b.b\_name = a.b\_name AND b.b\_city = 'busan' GROUP BY d.c\_name HAVING COUNT(DISTINCT b.b\_name) = (SELECT COUNT(b\_name) FROM BRANCH WHERE b\_city = 'busan');

SELECT \* FROM ACCOUNT;

DELETE FROM ACCOUNT WHERE b\_name IN (SELECT b\_name FROM BRANCH WHERE b\_city = 'busan');

SELECT \* FROM ACCOUNT;

SELECT DISTINCT d.c\_name FROM DEPOSITOR d, ACCOUNT a WHERE d.acc\_no = a.acc\_no AND a.b\_name = 'main';

SELECT DISTINCT b.c\_name FROM BORROWER b , LOAN l WHERE b.l\_num = l.l\_num AND b\_name = 'main';

**OUTPUT:**

main|seoul|100000.0

sub|busan|250000.0

sub1|daegu|300000.0

sub2|mumbai|550000.0

sub3|pune|990000.0

1|main|1100000

2|sub|1200000

3|sub1|1300000

4|sub2|4100000

5|sub3|1500000

1|main|1100000

2|sub|1200000

3|sub1|1300000

4|sub2|4100000

5|sub3|1500000

6|main|1000000

7|main|1800000

aabc|123 street|busan

ddef|456 street|seoul

gghi|789 street|daegu

jjkl|234 street|busan

mmno|567 street|daegu

abc|1

def|1

ghi|2

jkl|5

mno|3

1111|main|300000

2222|sub1|500000

3333|main|30000

4444|sub|670000

5555|sub2|300000

aabc|1111

ddef|2222

gghi|3333

jjkl|4444

mmno|5555

abc

def

ghi

1|main|1100000

2|sub|1200000

3|sub1|1300000

4|sub2|4100000

5|sub3|1500000

6|main|1000000

7|main|1800000

8|sub4|1800000

1|main|1100000

3|sub1|1300000

4|sub2|4100000

5|sub3|1500000

6|main|1000000

7|main|1800000

abc

def

ghi

aabc

gghi