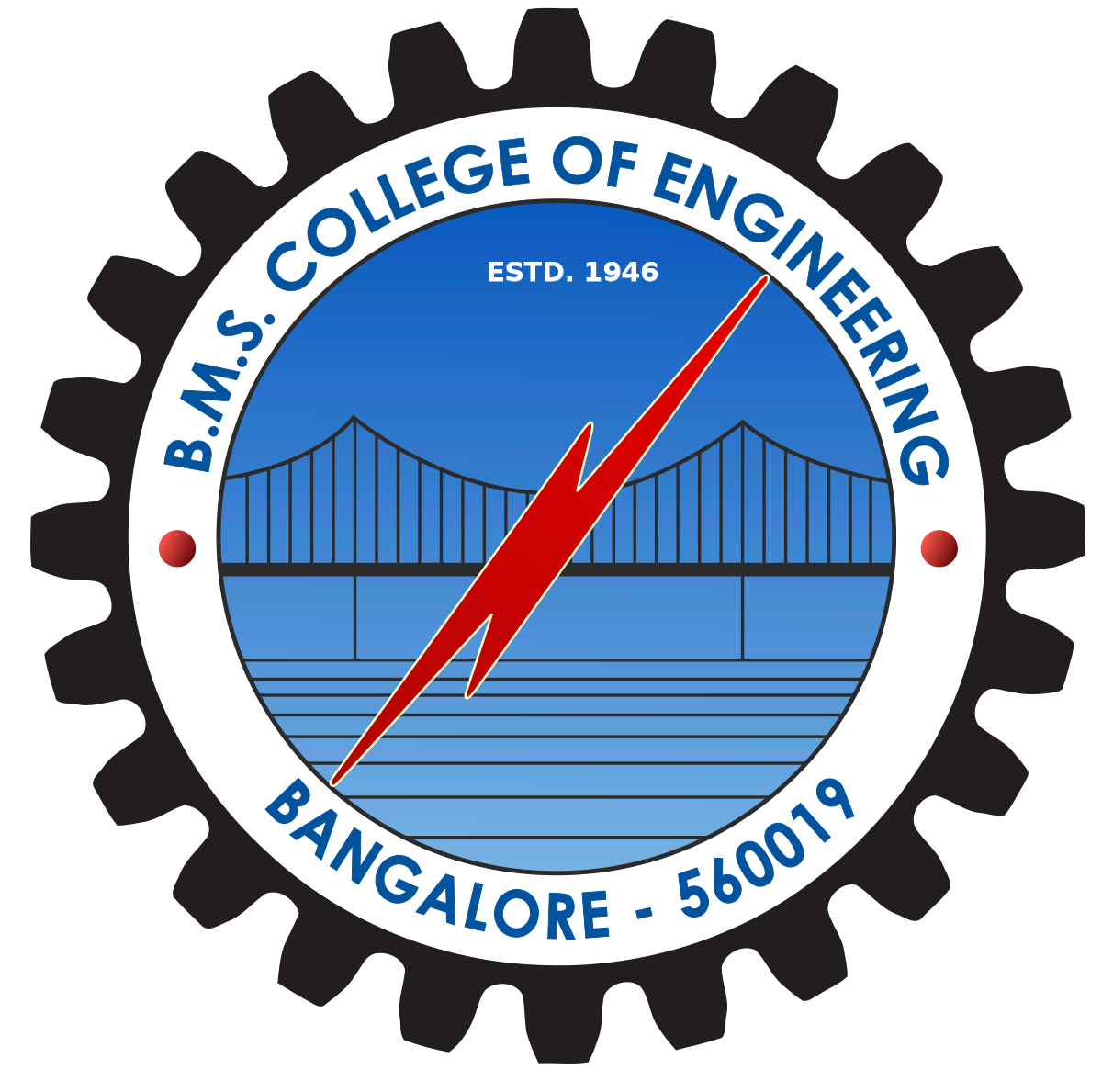
**B.M.S. COLLEGE OF ENGINEERING**

**(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)

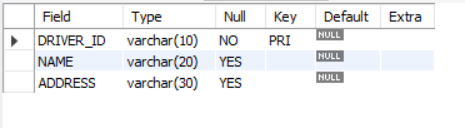
ACCIDENT (report\_num: int, accident\_date: date, location: String)

OWNS (driver\_id: String, reg\_num: String)

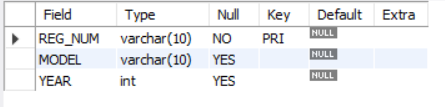
PARTICIPATED (driver\_id: String, reg\_num: String, report\_num: int, damage\_amount: int)

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

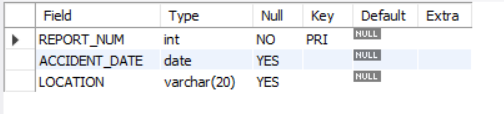
CREATE TABLE PERSON(DRIVER\_ID VARCHAR(10), NAME VARCHAR(20), ADDRESS VARCHAR(30), PRIMARY KEY (DRIVER\_ID));



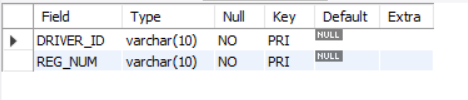
CREATE TABLE CAR(REG\_NUM VARCHAR(10), MODEL VARCHAR(10), YEAR INT, PRIMARY KEY(REG\_NUM));



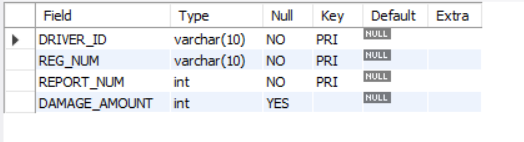
CREATE TABLE ACCIDENT(REPORT\_NUM INT, ACCIDENT\_DATE DATE, LOCATION VARCHAR(20), PRIMARY KEY(REPORT\_NUM));



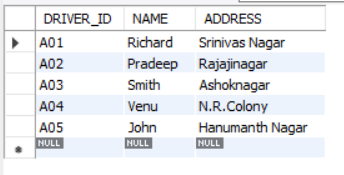
CREATE TABLE OWNS(DRIVER\_ID VARCHAR(10), REG\_NUM VARCHAR(10), PRIMARY KEY(DRIVER\_ID, REG\_NUM), FOREIGN KEY(DRIVER\_ID) REFERENCES PERSON(DRIVER\_ID), FOREIGN KEY(REG\_NUM) REFERENCES CAR (REG\_NUM));



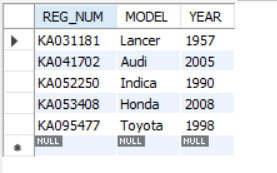
CREATE TABLE PARTICIPATED(DRIVER\_ID VARCHAR(10), REG\_NUM VARCHAR(10), REPORT\_NUM INT, DAMAGE\_AMOUNT INT, PRIMARY KEY(DRIVER\_ID, REG\_NUM, REPORT\_NUM), FOREIGN KEY(DRIVER\_ID) REFERENCES PERSON(DRIVER\_ID), FOREIGN KEY(REG\_NUM) REFERENCES CAR(REG\_NUM), FOREIGN KEY(REPORT\_NUM) REFERENCES ACCIDENT (REPORT\_NUM));

**ii)Enter at least five tuples for each relation.**

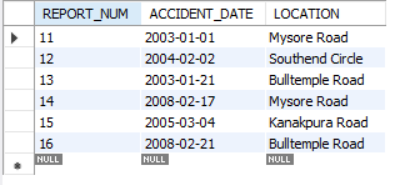
INSERT INTO PERSON VALUES('A01', 'Richard', 'Srinivas Nagar');  
INSERT INTO PERSON VALUES('A02', 'Pradeep', 'Rajajinagar');  
INSERT INTO PERSON VALUES('A03', 'Smith', 'Ashoknagar');  
INSERT INTO PERSON VALUES('A04', 'Venu', 'N.R.Colony');  
INSERT INTO PERSON VALUES('A05', 'John', 'Hanumanth Nagar');



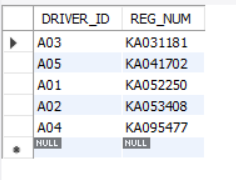
INSERT INTO CAR VALUES('KA052250', 'Indica', 1990);  
INSERT INTO CAR VALUES('KA031181', 'Lancer', 1957);  
INSERT INTO CAR VALUES('KA095477', 'Toyota', 1998);  
INSERT INTO CAR VALUES('KA053408', 'Honda', 2008);  
INSERT INTO CAR VALUES('KA041702', 'Audi', 2005);



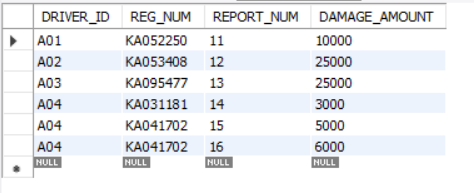
INSERT INTO ACCIDENT VALUES(11, '2003-01-01', 'Mysore Road');  
INSERT INTO ACCIDENT VALUES(12, '2004-02-02', 'Southend Circle');  
INSERT INTO ACCIDENT VALUES(13, '2003-01-21', 'Bulltemple Road');  
INSERT INTO ACCIDENT VALUES(14, '2008-02-17', 'Mysore Road');  
INSERT INTO ACCIDENT VALUES(15, '2005-03-04',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



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');

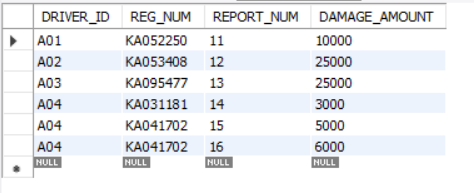


INSERT INTO PARTICIPATED VALUES('A01', 'KA052250', 11, 10000);  
INSERT INTO PARTICIPATED VALUES('A02', 'KA053408', 12, 50000);  
INSERT INTO PARTICIPATED VALUES('A03', 'KA095477', 13, 25000);  
INSERT INTO PARTICIPATED VALUES('A04', 'KA031181', 14, 3000);  
INSERT INTO PARTICIPATED VALUES('A05', 'KA041702',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



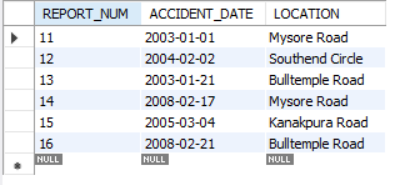
**iii) Demonstrate how you:  
  
a. Update the damage  
amount to 25000 for the car with a specific reg-num(example 'K A053408') for which the accident report number was 12.**

UPDATE PARTICIPATED SET DAMAGE\_AMOUNT = 25000 WHERE REPORT\_NUM = 12;



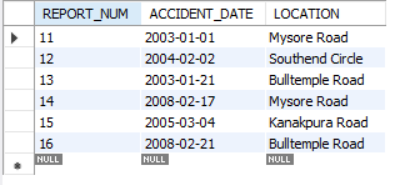
**b. Add a new accident to the database.**

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



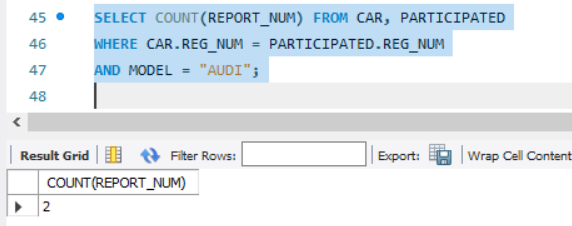
**iv) Find the total number of people who owned cars that involved in accidents in 2008.**

SELECT COUNT(DISTINCT DRIVER\_ID) FROM ACCIDENT, PARTICIPATED   
WHERE ACCIDENT.REPORT\_NUM = PARTICIPATED.REPORT\_NUM  
AND ACCIDENT\_DATE LIKE '2008%';



**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";



**PROGRAM 2: BANKING ENTERPRISE DATABASE**

Consider the following database for a banking enterprise.

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

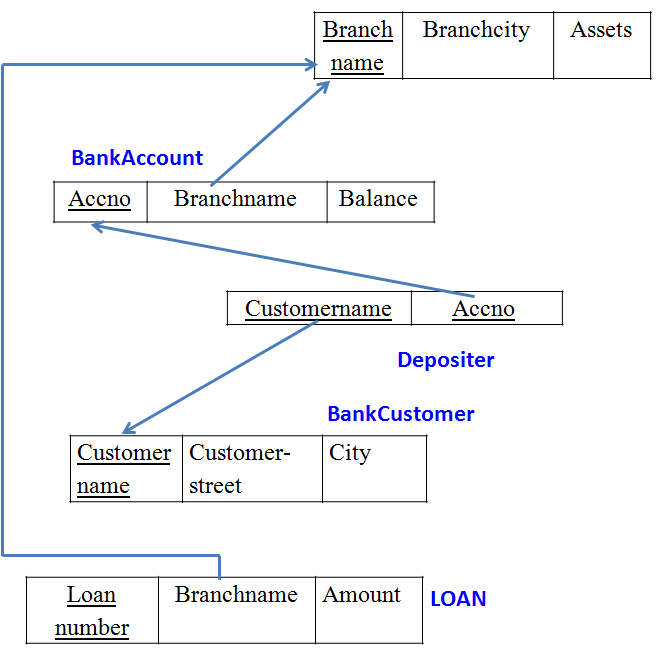
BankAccount(accno: int, branch-name: String, balance: real)

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

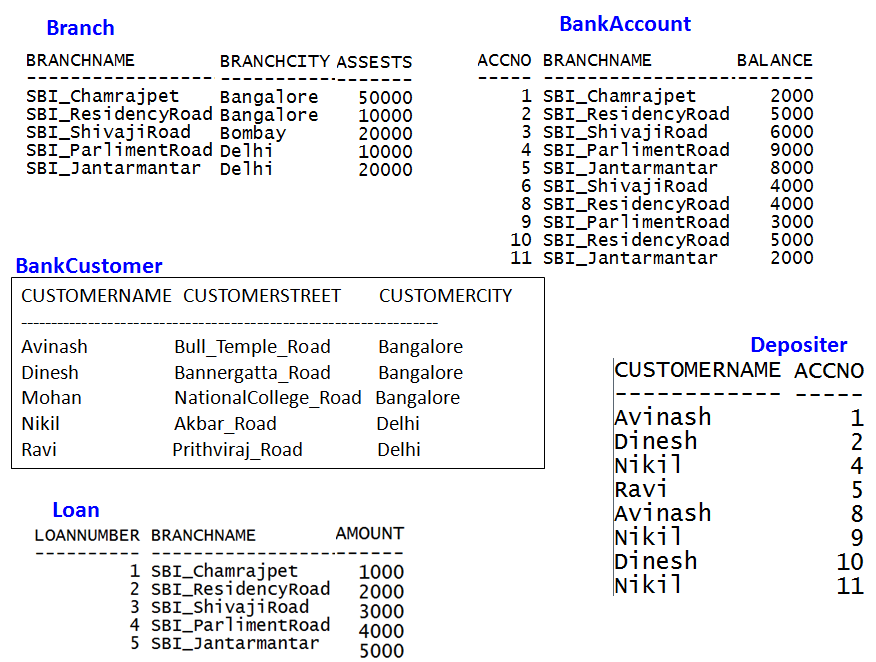
Depositer(customer-name: String, accno: int)

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

**Schema Diagram**

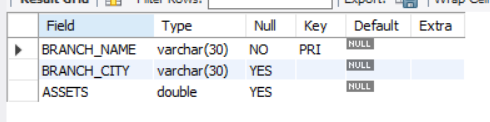


**Sample Table data**

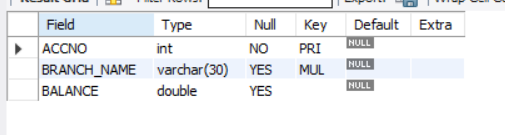


**i. Create the above tables by properly specifying the primary keys and the foreign keys.**

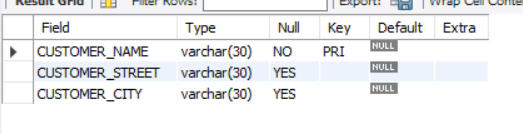
CREATE TABLE BRANCH (BRANCH\_NAME VARCHAR(30), BRANCH\_CITY VARCHAR(30), ASSETS REAL, PRIMARY KEY (BRANCH\_NAME));



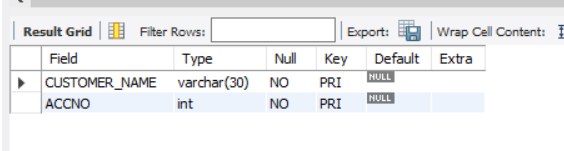
CREATE TABLE BANK\_ACCOUNT (ACCNO INT, BRANCH\_NAME VARCHAR(30), BALANCE REAL, PRIMARY KEY (ACCNO), FOREIGN KEY (BRANCH\_NAME) REFERENCES BRANCH(BRANCH\_NAME));



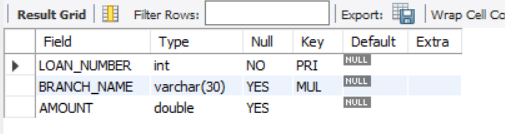
CREATE TABLE BANK\_CUSTOMER (CUSTOMER\_NAME VARCHAR(30), CUSTOMER\_STREET VARCHAR(30), CUSTOMER\_CITY VARCHAR(30), PRIMARY KEY(CUSTOMER\_NAME));



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));

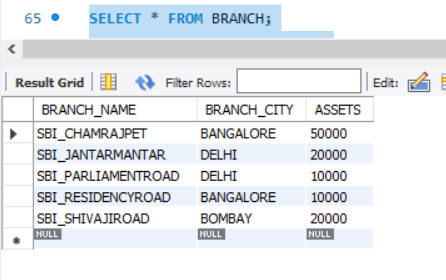


CREATE TABLE LOAN (LOAN\_NUMBER INT, BRANCH\_NAME VARCHAR(30), AMOUNT REAL, PRIMARY KEY (LOAN\_NUMBER), FOREIGN KEY (BRANCH\_NAME) REFERENCES BRANCH(BRANCH\_NAME));

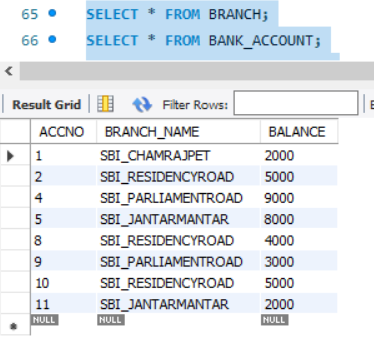


**ii. Enter at least five tuples for each relation.**

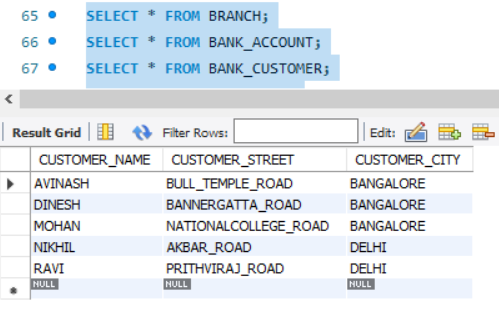
INSERT INTO BRANCH VALUES ('SBI\_CHAMRAJPET', 'BANGALORE', 50000);  
INSERT INTO BRANCH VALUES ('SBI\_RESIDENCYROAD', 'BANGALORE', 10000);  
INSERT INTO BRANCH VALUES ('SBI\_SHIVAJIROAD', 'BOMBAY', 20000);  
INSERT INTO BRANCH VALUES ('SBI\_PARLIAMENTROAD', 'DELHI', 10000);  
INSERT INTO BRANCH VALUES ('SBI\_JANTARMANTAR', 'DELHI', 20000);

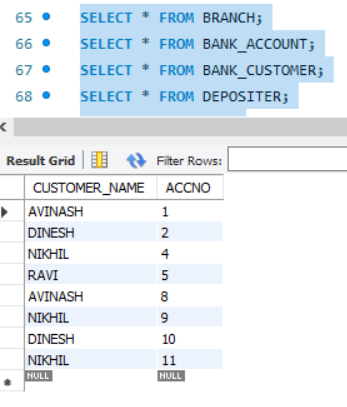


INSERT INTO BANK\_ACCOUNT VALUES ( 1,'SBI\_CHAMRAJPET', 2000);  
INSERT INTO BANK\_ACCOUNT VALUES ( 2,'SBI\_RESIDENCYROAD', 5000);  
INSERT INTO BANK\_ACCOUNT VALUES ( 3,'SBI\_SHIVAJIROAD', 6000);  
INSERT INTO BANK\_ACCOUNT VALUES ( 4,'SBI\_PARLIAMENTROAD', 9000);  
INSERT INTO BANK\_ACCOUNT VALUES ( 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);

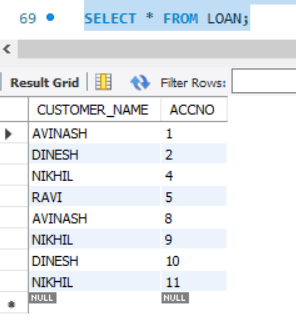


INSERT INTO BANK\_CUSTOMER VALUES ('AVINASH', 'BULL\_TEMPLE\_ROAD', 'BANGALORE');  
INSERT INTO BANK\_CUSTOMER VALUES ('DINESH', 'BANNERGATTA\_ROAD', 'BANGALORE');  
INSERT INTO BANK\_CUSTOMER VALUES ('MOHAN', 'NATIONALCOLLEGE\_ROAD', 'BANGALORE');  
INSERT INTO BANK\_CUSTOMER VALUES ('NIKHIL', 'AKBAR\_ROAD', 'DELHI');  
INSERT INTO BANK\_CUSTOMER VALUES ('RAVI', 'PRITHVIRAJ\_ROAD', 'DELHI');

  
  
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);

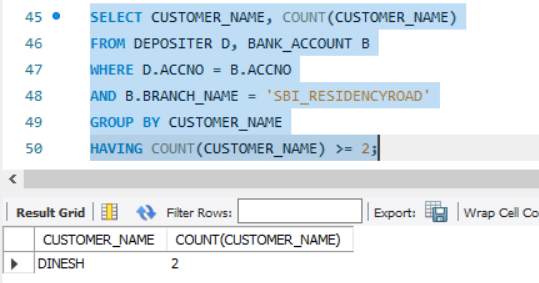


INSERT INTO LOAN VALUES (1, 'SBI\_CHAMRAJPET', 1000);  
INSERT INTO LOAN VALUES (2, 'SBI\_RESIDENCYROAD', 2000);  
INSERT INTO LOAN VALUES (3, 'SBI\_SHIVAJIROAD', 3000);  
INSERT INTO LOAN VALUES (4, 'SBI\_PARLIAMENTROAD', 4000);  
INSERT INTO LOAN VALUES (5, 'SBI\_JANTARMANTAR', 5000);



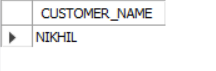
**iii. Find all the customers who have at least two accounts at the Main branch (ex. SBI\_ResidencyRoad).**

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;



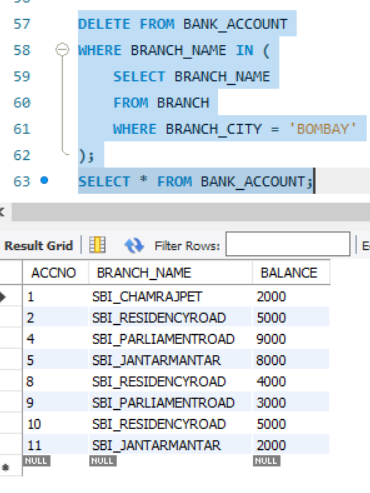
**iv. Find all the customers who have an account at all the branches located in a specific city (Ex. Delhi).**

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');



**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;



**PROGRAM 3: SUPPLIER DATABASE**

Consider the following schema:

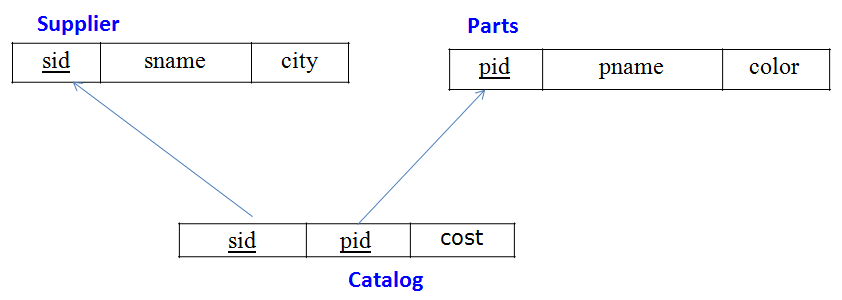
SUPPLIERS(sid: integer, sname: string, address: string)

PARTS(pid: integer, pname: string, color: string)

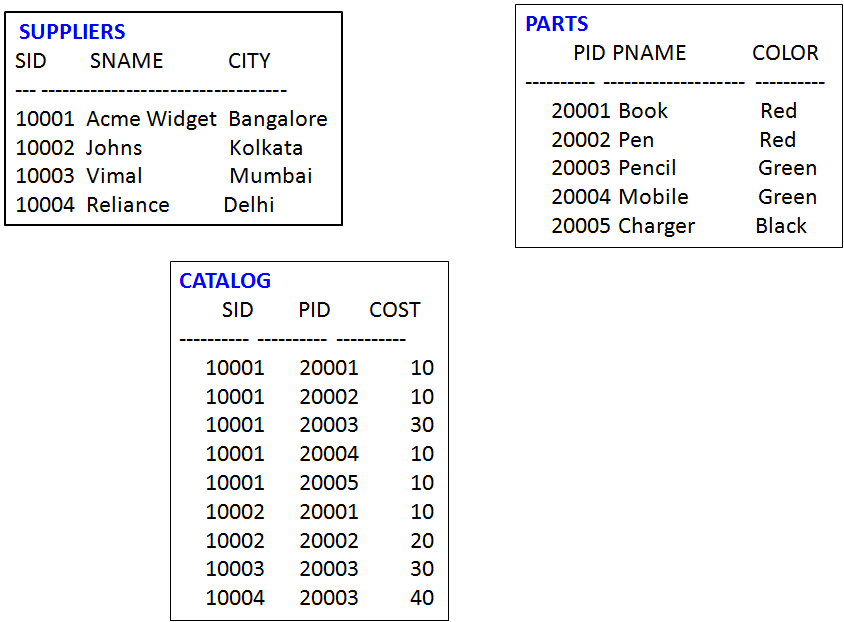
CATALOG(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers.

**Schema Diagram**

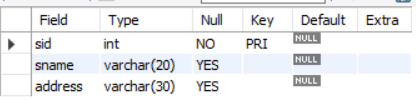


**Table Data**

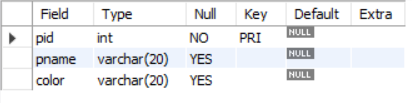


**Creation of Tables:**

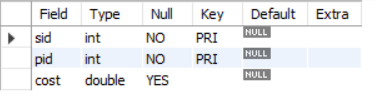
CREATE TABLE suppliers (  
 sid INT,   
 sname VARCHAR(20),  
 address VARCHAR(30),  
 PRIMARY KEY (sid)  
);



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

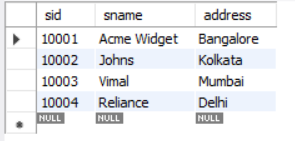


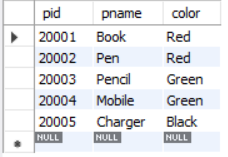
CREATE TABLE catalog (  
 sid INT,  
 pid INT,  
 cost REAL,  
 PRIMARY KEY(sid, pid),  
 FOREIGN KEY (sid) REFERENCES suppliers(sid),  
 FOREIGN KEY (pid) REFERENCES parts(pid)  
);

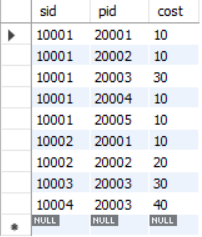


**Inserting Values into the tables:**

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');

  
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');

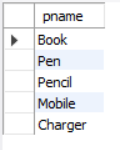
  
   
INSERT INTO catalog VALUES (10001, 20001, 10);  
INSERT INTO catalog VALUES (10001, 20002, 10);  
INSERT INTO catalog VALUES (10001, 20003, 30);  
INSERT INTO catalog VALUES (10001, 20004, 10);  
INSERT INTO catalog VALUES (10001, 20005, 10);  
INSERT INTO catalog VALUES (10002, 20001, 10);  
INSERT INTO catalog VALUES (10002, 20002, 20);  
INSERT INTO catalog VALUES (10003, 20003, 30);  
INSERT INTO catalog VALUES (10004, 20003, 40);



**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;



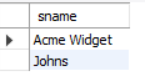
1. **Find the snames of suppliers who supply every part.**

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  
 )  
);



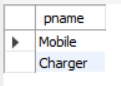
1. **Find the snames of suppliers who supply every red part.**

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  
 )  
);



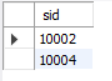
1. **Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.**

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'  
);



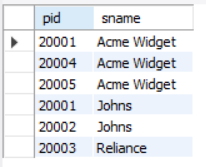
1. **Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).**

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



1. **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  
);



**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

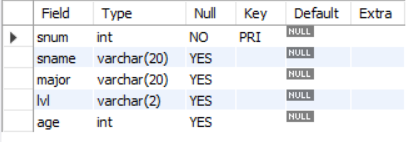
code with 4 different values (example: Junior: JR etc)

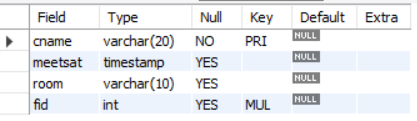
Write the following queries in SQL.

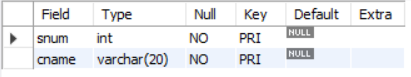
No duplicates should be printed in any of the answers.

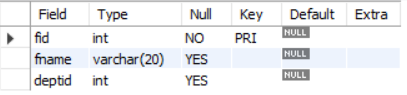
**Creation of Tables:**

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

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

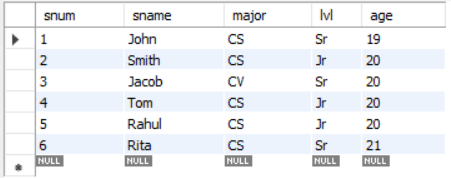
  
  
CREATE TABLE enrolled (  
 snum int,   
 cname varchar(20),  
 primary key(snum, cname),  
 foreign key (snum) references student(snum),  
 foreign key (cname) references class(cname)  
);

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

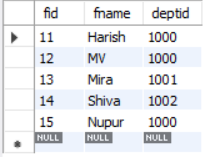


**Inserting Values into the tables:**

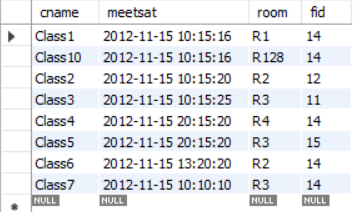
INSERT INTO student VALUES (1, 'John', 'CS', 'Sr', 19);  
INSERT INTO student VALUES (2, 'Smith', 'CS', 'Jr', 20);  
INSERT INTO student VALUES (3, 'Jacob', 'CV', 'Sr', 20);  
INSERT INTO student VALUES (4, 'Tom', 'CS', 'Jr', 20);  
INSERT INTO student VALUES (5, 'Rahul', 'CS', 'Jr', 20);  
INSERT INTO student VALUES (6, 'Rita', 'CS', 'Sr', 21);



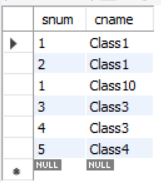
INSERT INTO faculty VALUES(11, 'Harish', 1000);  
INSERT INTO faculty VALUES(12, 'MV', 1000);  
INSERT INTO faculty VALUES(13, 'Mira', 1001);  
INSERT INTO faculty VALUES(14, 'Shiva', 1002);  
INSERT INTO faculty VALUES(15, 'Nupur', 1000);



INSERT INTO class VALUES ('Class1', '12/11/15 10:15:16.00000', '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);

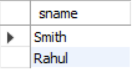


INSERT INTO enrolled VALUES (1, 'Class1');  
INSERT INTO enrolled VALUES (2, 'Class1');  
INSERT INTO enrolled VALUES (3, 'Class3');  
INSERT INTO enrolled VALUES (4, 'Class3');  
INSERT INTO enrolled VALUES (5, 'Class4');



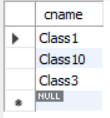
1. **Find the names of all Juniors (level = JR) who are enrolled in a class taught by “name”**

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';



1. **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  
);



1. **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  
);



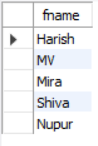
1. **Find the names of faculty members who teach in every room in which some class is taught.**

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  
)  
);



1. **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  
);



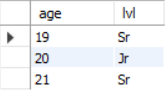
1. **Find the names of students who are not enrolled in any class.**

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



1. **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  
 )  
);



**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)

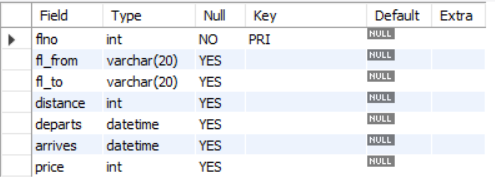
EMPLOYEES(eid: integer, ename: string, salary: integer)

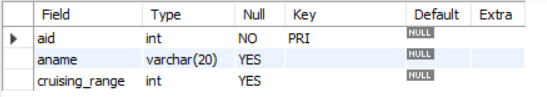
Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

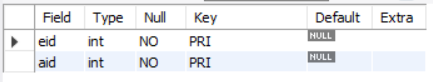
Write each of the following queries in SQL.

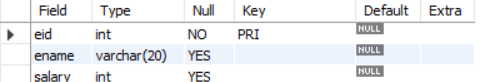
**Creation of Tables:**

CREATE TABLE flights(  
 flno INT,   
 fl\_from VARCHAR(20),   
 fl\_to VARCHAR(20),  
 distance INT,  
 departs DATETIME,  
 arrives DATETIME,  
 price INT,  
 PRIMARY KEY(flno)  
);

  
  
CREATE TABLE aircraft (  
 aid INT,  
 aname VARCHAR(20),  
 cruising\_range INT,  
 PRIMARY KEY(aid)  
);

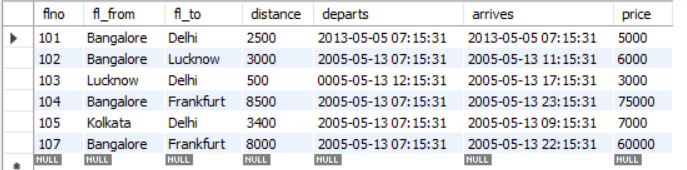
  
  
CREATE TABLE certified (  
 eid INT,  
 aid INT,  
 PRIMARY KEY(eid, aid),  
 FOREIGN KEY (eid) REFERENCES employees(eid),  
 FOREIGN KEY(aid) REFERENCES aircraft(aid)  
);

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

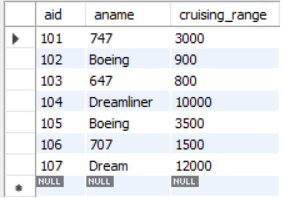


**Inserting Values into the tables:**

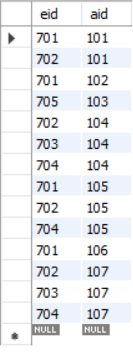
INSERT INTO flights VALUES (101, 'Bangalore', 'Delhi', 2500, '13-05-05 07.15.31.000000', '13-05-05 07.15.31.000000', 5000);  
INSERT INTO flights VALUES (102, 'Bangalore', 'Lucknow', 3000, '05/05/13 07:15:31', '05/05/13 11:15:31', 6000);  
INSERT INTO flights VALUES (103, 'Lucknow', 'Delhi', 500, '5/05/13 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);



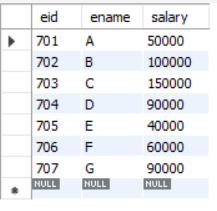
INSERT INTO aircraft VALUES (101, '747', 3000);  
INSERT INTO aircraft VALUES (102, 'Boeing', 900);  
INSERT INTO aircraft VALUES (103, '647', 800);  
INSERT INTO aircraft VALUES (104, 'Dreamliner', 10000);  
INSERT INTO aircraft VALUES (105, 'Boeing', 3500);  
INSERT INTO aircraft VALUES (106, '707', 1500);  
INSERT INTO aircraft VALUES (107, 'Dream', 12000);



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);

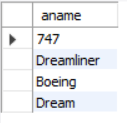


INSERT INTO employees VALUES (701, 'A', 50000);  
INSERT INTO employees VALUES (702, 'B', 100000);  
INSERT INTO employees VALUES (703, 'C', 150000);  
INSERT INTO employees VALUES (704, 'D', 90000);  
INSERT INTO employees VALUES (705, 'E', 40000);  
INSERT INTO employees VALUES (706, 'F', 60000);  
INSERT INTO employees VALUES (707, 'G', 90000);



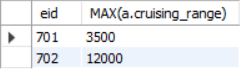
1. **Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.**

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;



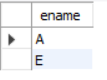
1. **For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.**

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;



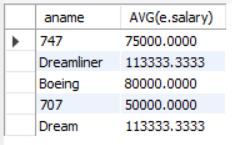
1. **Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.**

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'  
);



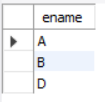
1. **For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.**

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;



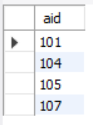
1. **Find the names of pilots certified for some Boeing aircraft.**

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';



1. **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';



1. **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.**