**B.M.S. College of Engineering**

**Autonomous Institute,affiliated to VTU**

**Department of Computer Science and Engineering**



**DATABASE MANAGAEMENT SYSTEM LAB RECORD (19CS4PCDBM)**

**NAME: SUPRIYA M LINGDAL**

**USN: 1BM20CS415**

**SEM: FOURTH SEM**

**CSE-D SECTION**

**NAME OF THE EXPERIMENTS:**

1 Insurance Database

2 Banking Enterprise Database

3 Supplier Database

4 Student Faculty Database

5 Airline Flight Database

**PROGRAM 1**

**INSURANCE DATABASE**

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

PERSON (driver-id #: String, name: String, address: String)

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

ACCIDENT (report-number: int, date: date, location: String)

OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int

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

create database Supriya;

use Supriya;

create table person(

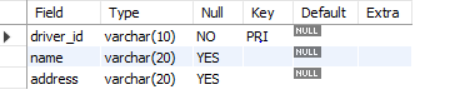
driver\_id varchar(10),

name varchar(20),

address varchar(20),

primary key(driver\_id));

desc person;



create table car

(

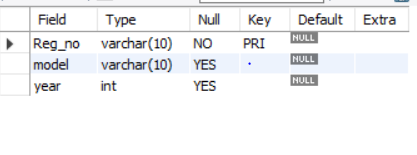
Reg\_no varchar(10),

model varchar(10),

year int(10),

primary key(Reg\_no));

desc car;



create table accident(

Report\_no int(10),

accident\_date date,

location varchar(20),

primary key(Report\_no));

desc accident;

create table owns

(

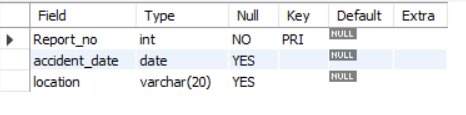
driver\_id varchar(10),

Reg\_no varchar(10),

primary key(driver\_id,Reg\_no),

foreign key(driver\_id)references person(driver\_id),

foreign key(Reg\_no)references car(Reg\_no));



create table participated

(

driver\_id varchar(10),

Reg\_no varchar(10),

report\_num int,

damage\_amount int,

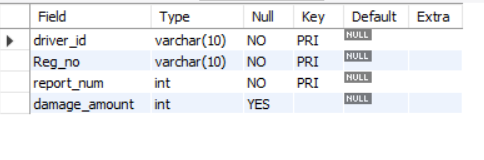
primary key(driver\_id,Reg\_no,report\_num),

foreign key(driver\_id)references person(driver\_id),

foreign key(Reg\_no)references car(Reg\_no),

foreign key(report\_num)references accident(Report\_no));

desc participated;



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

use Supriya;

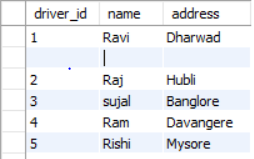
insert into person values(1,'Ravi','Dharwad');

insert into person values(2,'Raj','Hubli');

insert into person values(3,'sujal','Banglore');

insert into person values(4,'Ram','Davangere');

insert into person values(5,'Rishi','Mysore');



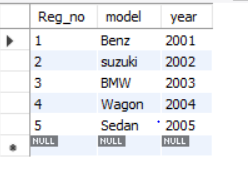
insert into car values(1,'Benz','2001');

insert into car values(2,'suzuki','2002');

insert into car values(3,'BMW','2003');

insert into car values(4,'Wagon','2004');

insert into car values(5,'Sedan','2005');



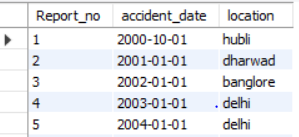
insert into accident values(1,'Ravi','Dharwad');

insert into accident values(2,'Raj','Hubli');

insert into accident values(3,'sujal','Banglore');

insert into accident values(4,'Ram','Davangere');

insert into accident values(5,'Rishi','Mysore');



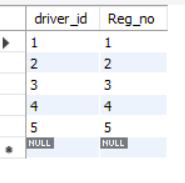
insert into owns values(1,1);

insert into owns values(2,2);

insert into owns values(3,3);

insert into owns values(4,4);

insert into owns values(5,5);

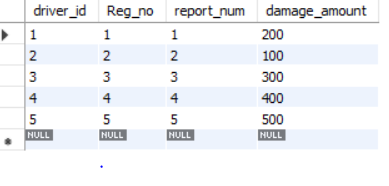


insert into participated values(1,1,1,200);

insert into participated values(2,2,2,100);

insert into participated values(3,3,3,300);

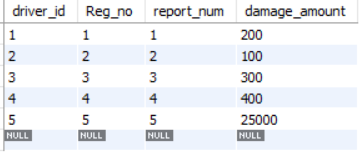
insert into participated values(4,4,4,400);

insert into participated values(5,5,5,500);

**iii) a.Update the damage amount for the car with a specific Reg\_no in the accident with report number 5 to 25000.**

use Supriya;

update participated set damage\_amount=25000 where Reg\_no=5 AND report\_num=5;



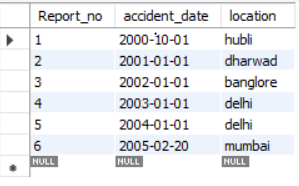
**b. Add a new accident to the database.**

use Supriya;

select \* from accident;

insert into accident values(6,'2005-02-20','mumbai');

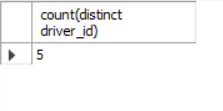
select \* from accident;



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

use Supriya;

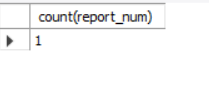
select count(distinct driver\_id)from participated p,accident a where p.report\_num and a.accident\_date like '2000-10-01';



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

use Supriya;

select count(report\_num)from car c,participated p where c.Reg\_no=p.Reg\_no and model='Benz';



**PROGRAM 2**

**BANKING ENTERPRISE DATABASE**

**Consider the following database for a banking enterprise.**

**BRANCH (branch-name: String, branch-city: String, assets: real)**

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

**DEPOSITOR (customer-name: String, customer-street: String, customer-city: String)**

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

**BORROWER (customer-name: String, loan-number: int)**

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

use Supriya;

create table branch

(

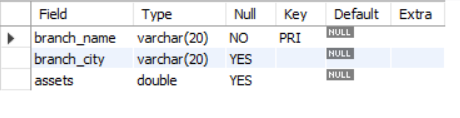
branch\_name varchar(20),

branch\_city varchar(20),

assets real,

primary key(branch\_name));

desc branch;



use Supriya;

create table accounts

(

branch\_name varchar(20),

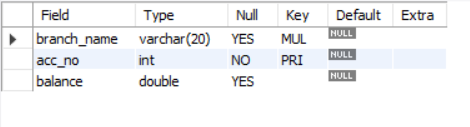
acc\_no integer,

balance real,

primary key(acc\_no),

foreign key(branch\_name)references branch(branch\_name));

desc accounts;



create table depositor

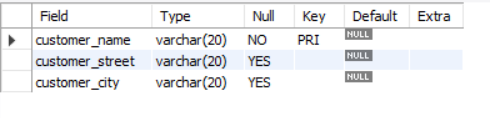
(

customer\_name varchar(20),

customer\_street varchar(20),

customer\_city varchar(20),

primary key(customer\_name));



create table loan

(

loan\_number int,

branch\_name varchar(20),

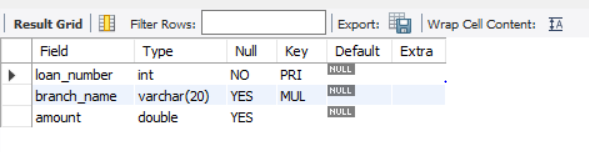
amount real,

primary key(loan\_number),

foreign key(branch\_name)references branch(branch\_name)

);

desc loan;



create table borrower

(

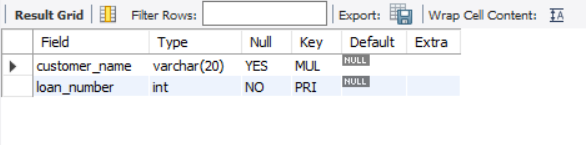
customer\_name varchar(20),

loan\_number int,

primary key(loan\_number),

foreign key(customer\_name)references depositor(customer\_name),

foreign key(loan\_number)references loan(loan\_number));



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

use Supriya;

insert into branch values('SBI','Bangalore','25.6');

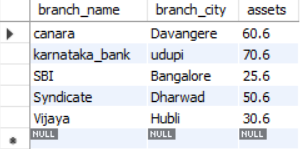
insert into branch values('Vijaya','Hubli','30.6');

insert into branch values('Syndicate','Dharwad','50.6');

insert into branch values('canara','Davangere','60.6');

insert into branch values('karnataka\_bank','udupi','70.6');

select \* from branch;



insert into accounts values('SBI','12345','25000.6');

insert into accounts values('Vijaya','23456','30000.6');

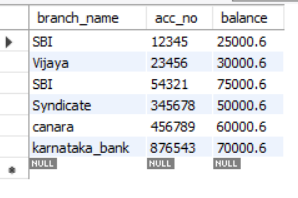
insert into accounts values('Syndicate','345678','50000.6');

insert into accounts values('canara','456789','60000.6');

insert into accounts values('karnataka\_bank','876543','70000.6');

insert into accounts values('SBI','54321','75000.6');

select \* from accounts;



insert into depositor values('Ravi','vidyagiri','Dharwad');

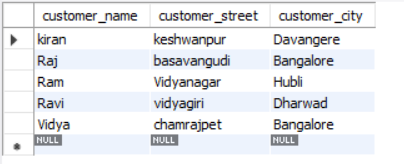
insert into depositor values('Vidya','chamrajpet','Bangalore');

insert into depositor values('Ram','Vidyanagar','Hubli');

insert into depositor values('Raj','basavangudi','Bangalore');

insert into depositor values('kiran','keshwanpur','Davangere');

select \* from depositor;



insert into loan values('1','SBI','25000.6');

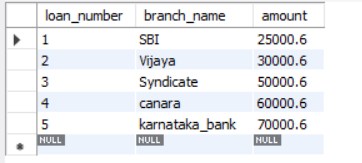
insert into loan values('2','Vijaya','30000.6');

insert into loan values('3','Syndicate','50000.6');

insert into loan values('4','canara','60000.6');

insert into loan values('5','karnataka\_bank','70000.6');

select \* from loan;



insert into borrower values('kiran',1);

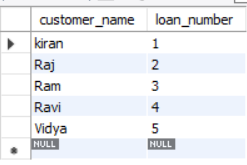
insert into borrower values('Raj',2);

insert into borrower values('Ram',3);

insert into borrower values('Ravi',4);

insert into borrower values('Vidya',5);

select \* from borrower;



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

select c.customer\_name

from bank\_customer c

where exists(

select d.cust\_name

from depositer d, bank\_account ba

where

d.accno=ba.accno and

c.customer\_name=d.cust\_name and

ba.branch\_name='SBI\_chandralayout'

group by d.cust\_name

having count(d.cust\_name)>=2

);

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

select d.cust\_name from depositer d,branch b,bank\_account ba

where b.branch\_name=ba.branch\_name

AND ba.accno=d.accno

and branch\_city='Delhi'

group by d.cust\_name

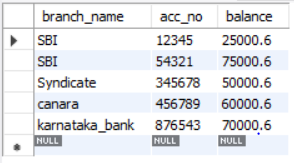
having COUNT(distinct b.branch\_name)=(select COUNT(branch\_name) from branch where branch\_city='Delhi');

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

use Supriya;

delete from accounts where branch\_name IN(select branch\_name from branch where branch\_city='Hubli');

select \* from accounts;



**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)**

use Supriya;

create table suppliers

(

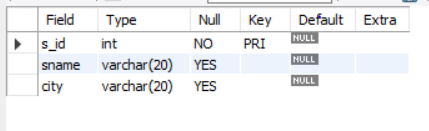
s\_id int(5),

sname varchar(20),

city varchar(20),

primary key (s\_id));

desc suppliers;



use Supriya;

create table parts

(

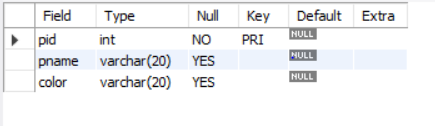
pid int(10),

pname varchar(20),

color varchar(20),

primary key(pid));

desc parts;



use Supriya;

create table catalog

(

s\_id int(10),

pid int(10),

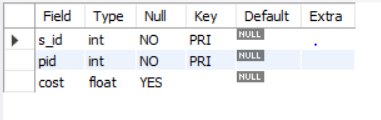
cost float(10),

primary key(pid,s\_id),

foreign key(s\_id)references suppliers(s\_id),

foreign key(pid)references parts(pid));

desc catalog;



use Supriya;

insert into suppliers values(1,'Reliance','Shrinagar');

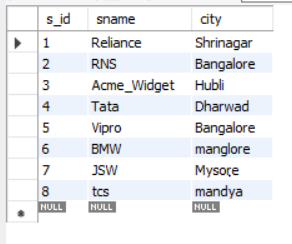
insert into suppliers values(2,'RNS','Bangalore');

insert into suppliers values(3,'Acme\_Widget','Hubli');

insert into suppliers values(4,'Tata','Dharwad');

insert into suppliers values(5,'Vipro','Bangalore');

select \* from suppliers;



use Supriya;

insert into parts values(1,'break','black');

insert into parts values(2,'engine','brown');

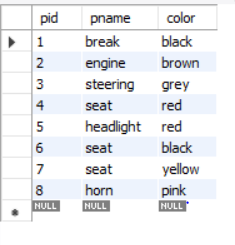
insert into parts values(3,'steering','grey');

insert into parts values(4,'seat','red');

insert into parts values(5,'headlight','red');

insert into parts values(6,'seat','black');

select \* from parts;



use Supriya;

desc catalog;

insert into catalog values(1,1,'1000.0');

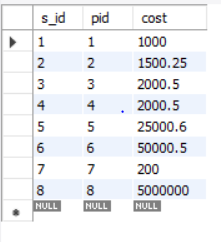
insert into catalog values(2,2,'1500.25');

insert into catalog values(3,3,'2000.50');

insert into catalog values(4,4,'2000.50');

insert into catalog values(5,5,'3000.25');

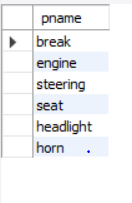
select \* from catalog;



**i)Find the pnames of parts for which there is some supplier.**

use Supriya;

SELECT DISTINCT P.pname FROM parts P, catalog C WHERE P.pid = C.pid;



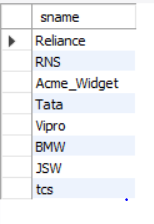
**ii)Find the snames of suppliers who supply every part.**

select s.sname

from suppliers s

where not exists(

select p.pid from parts p where not exists(select c.s\_id from catalog c where c.s\_id=s.s\_id));



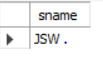
**iii)Find the snames of suppliers who supply every yellow part.**

select s.sname

from suppliers s

where not exists(select p.pid from parts p where p.color='yellow' and not exists(select c.s\_id from catalog c

where c.s\_id=s.s\_id and c.pid=p.pid));



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

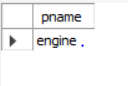
select p.pname

from parts p,catalog c,suppliers s

where p.pid=c.pid and

c.s\_id=s.s\_id and

s.sname='RNS' and not exists(select \* from catalog ca,suppliers su where p.pid=ca.pid and ca.s\_id=su.s\_id and su.sname<>'RNS');

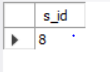


**v.Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).**

select distinct c.s\_id

from catalog c

where c.cost>(select avg(ca.cost) from catalog ca,catalog c where ca.pid=c.pid);



**vi. 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)

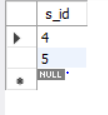
FROM Catalog C1

WHERE C1.pid = P.pid);

**vii. Find the sids of suppliers who supply only red parts**

select s.s\_id from suppliers s where exists(

select c.s\_id from catalog c,parts p where p.color='red' and p.pid=c.pid and c.s\_id=s.s\_id);



**PROGRAM 4.**

**STUDENT FACULTY DATABASE**

**Consider the following database for student enrolment for course:**

**STUDENT (snum: integer, sname: string, major: string, level: string, age: integer)**

**CLASS (name: string, meets at: time, room: string, fid: integer)**

**ENROLLED (snum: integer, cname: string)**

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

use Supriya;

create table student(

snum INT,

sname VARCHAR(10),

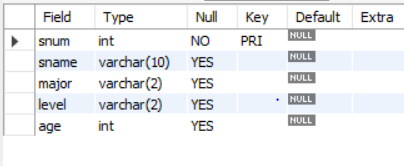
major VARCHAR(2),

level VARCHAR(2),

age INT,

primary key(snum));

desc student;



CREATE TABLE faculty(

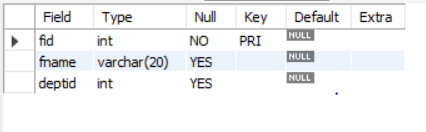
fid INT,

fname VARCHAR(20),

deptid INT,

PRIMARY KEY(fid));

desc faculty;



CREATE TABLE class(

cname VARCHAR(20),

meets\_at TIMESTAMP,

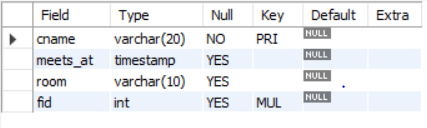
room VARCHAR(10),

fid INT,

PRIMARY KEY(cname),

FOREIGN KEY(fid) REFERENCES faculty(fid));

Desc class;



CREATE TABLE enrolled(

snum INT,

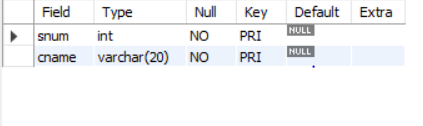
cname VARCHAR(20),

PRIMARY KEY(snum,cname),

FOREIGN KEY(snum) REFERENCES student(snum),

FOREIGN KEY(cname) REFERENCES class(cname));

desc enrolled;



insert into student values(1,'Ram','CS','JR',10);

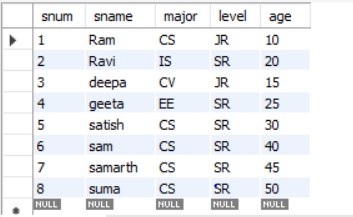
insert into student values(2,'Ravi','IS','SR',20);

insert into student values(3,'deepa','CV','JR',15);

insert into student values(4,'geeta','EE','SR',25);

insert into student values(5,'satish','CS','SR',30);

select \* from student;



insert into faculty values(1,'Seeta',10);

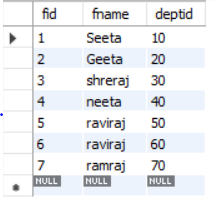
insert into faculty values(2,'Geeta',20);

insert into faculty values(3,'shreraj',30);

insert into faculty values(4,'neeta',40);

insert into faculty values(5,'raviraj',50);

select \* from faculty;



insert into class values('class1','12/11/15 10:15:16',1,1);

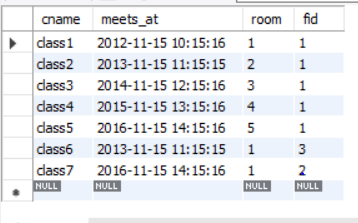
insert into class values('class2','13/11/15 11:15:15',2,2);

insert into class values('class3','14/11/15 12:15:16',3,3);

insert into class values('class4','15/11/15 13:15:16',4,4);

insert into class values('class5','16/11/15 14:15:16',5,5);

select \* from class;



insert into enrolled values(1,'class1');

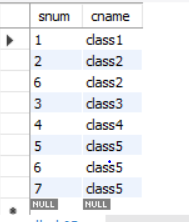
insert into enrolled values(2,'class2');

insert into enrolled values(3,'class3');

insert into enrolled values(4,'class4');

insert into enrolled values(5,'class5');

select \* from enrolled;



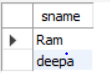
**i)Find the names of all Juniors (level= Jr) who are enrolled in a class taught by Seeta**

use Supriya;

select S.sname from student S where exists

(select S1.snum from student S1 where S.level='JR')AND EXISTS

(select F.fid from faculty F,enrolled E,class C where F.fname='Seeta' AND E.snum=S.snum AND E.cname = C.cname AND C.fid = F.fid );



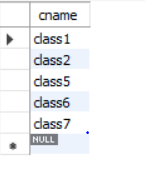
**ii. Find the names of all classes that either meet in room ‘ 1’ or have 2 or more Students enrolled.**

select c.cname

from class c

where c.room='1'

or c.cname in(select e.cname from enrolled e group by e.cname having count(\*)>=2);



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

select distinct s.sname

from student s

where s.snum in(select e1.snum

               from enrolled e1,enrolled e2,class c1, class c2

   where e1.snum=e2.snum and e1.cname<>e2.cname

   and e1.cname=c1.cname

               and e2.cname=c2.cname

               and c1.meets\_at=c2.meets\_at);

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

select f.fname

from faculty f

where f.fid in (select fid from class

group by fid having count(\*)=(select count(distinct room) from class));



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

SELECT DISTINCT F.fname

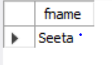
FROM faculty F

WHERE 5<(SELECT COUNT(E.snum)

FROM class C,enrolled E

WHERE C.cname = E.cname

AND C.fid = F.fid);



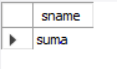
**vi)Find the names of students who are not enrolled in any class**

SELECT DISTINCT S.sname

FROM student S

WHERE S.snum NOT IN (SELECT E.snum

FROM enrolled E );



**vii. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR)**

SELECT S.age, S.level

FROM student S

GROUP BY S.age, S.level

HAVING S.level IN(SELECT S1.level FROM student S1

WHERE S1.age = S.age

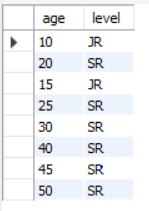
GROUP BY S1.level,S1.age

HAVING COUNT(\*)>= ALL(SELECT COUNT(\*)

FROM student S2

WHERE s1.age = S2.age

GROUP BY S2.level, 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)**

**EMPLOYEE (eid: integer, ename: string, salary: integer)**

use Supriya;

CREATE TABLE flights

(

flno INTEGER PRIMARY KEY,

ffrom VARCHAR(15) NOT NULL,

tto VARCHAR(15) NOT NULL,

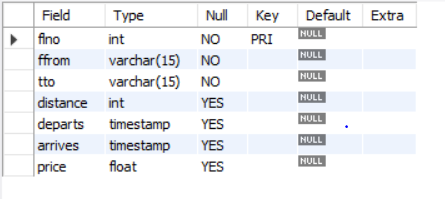
distance INTEGER,

departs TIMESTAMP,

arrives TIMESTAMP,

price float(10));

desc flights;



CREATE TABLE aircraft

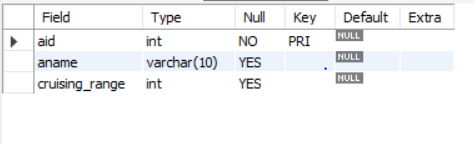
(

aid INTEGER PRIMARY KEY,

aname VARCHAR(10),

cruising\_range INTEGER);

desc aircraft;



CREATE TABLE employee

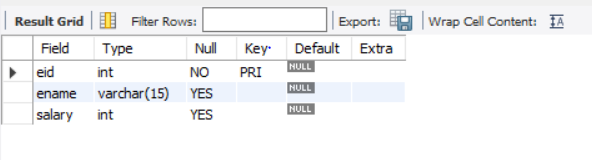
(

eid INTEGER PRIMARY KEY,

ename VARCHAR(15),

salary INTEGER(10));

desc employee;



CREATE TABLE certified

(

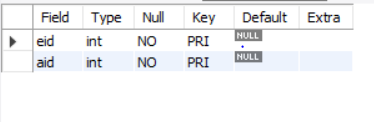
eid INTEGER NOT NULL,

aid INTEGER NOT NULL,

PRIMARY KEY (eid,aid),

FOREIGN KEY (eid) REFERENCES employee (eid), FOREIGN KEY (aid) REFERENCES aircraft (aid));

desc certified;



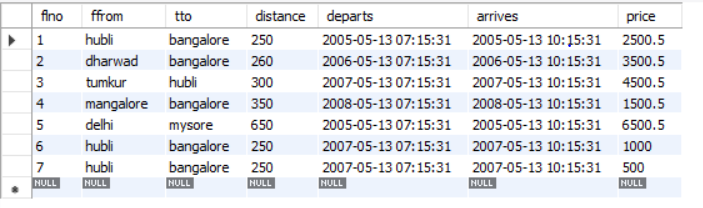
insert into flights values(1,'hubli','bangalore',250,'2005-05-13 07:15:31','2005-05-13 10:15:31','2500.5');

insert into flights values(2,'dharwad','bangalore',260,'2006-05-13 07:15:31','2006-05-13 10:15:31','3500.5');

insert into flights values(3,'tumkur','hubli',300,'2007-05-13 07:15:31','2007-05-13 10:15:31','4500.5');

insert into flights values(4,'mangalore','bangalore',350,'2008-05-13 07:15:31','2008-05-13 10:15:31','1500.5');

insert into flights values(5,'delhi','mysore',650,'2005-05-13 07:15:31','2005-05-13 10:15:31','6500.5');



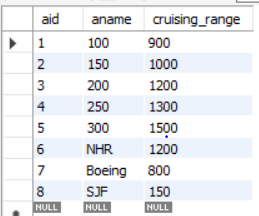
insert into aircraft values(1,'100','900');

insert into aircraft values(2,'150','1000');

insert into aircraft values(3,'200','1200');

insert into aircraft values(4,'250','1300');

insert into aircraft values(5,'300','1500');



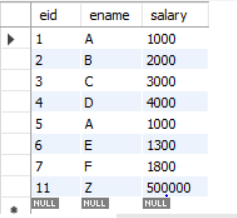
insert into employee values(1,'A',1000);

insert into employee values(2,'B',2000);

insert into employee values(3,'C',3000);

insert into employee values(4,'D',4000);

insert into employee values(5,'A',1000);



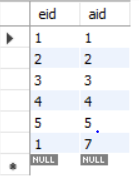
insert into certified values(1,1);

insert into certified values(2,2);

insert into certified values(3,3);

insert into certified values(4,4);

insert into certified values(5,5);



**i)Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.2000**

SELECT DISTINCT A.aname

FROM aircraft A

WHERE A.aid IN (SELECT C.aid

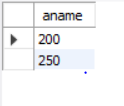
FROM certified C,employee E

WHERE C.eid = E.eid AND

EXISTS( SELECT \*

FROM employee E1

WHERE E1.eid = E.eid AND E1.salary>2000));



**ii)For each pilot who is certified for more than one aircrafts, find the eid and the maximum cruising range of the aircraft for which she or he is certified**

SELECT C.eid, MAX(A.cruising\_range)

FROM certified C,aircraft A

WHERE C.aid = A.aid

GROUP BY C.eid

HAVING COUNT(\*)>1;



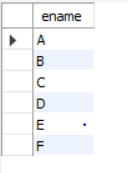
**iii)Find the names of pilots whose salary is less than the price of the cheapest route from delhi to mysore.**

SELECT DISTINCT E.ename

FROM employee E

WHERE E.salary<( SELECT MIN(F.price)

FROM flights F WHERE F.ffrom='delhi' AND F.tto='mysore');



**iv)For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.**

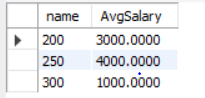
SELECT Temp.name,Temp.AvgSalary

FROM (SELECT A.aid, A.aname AS name, AVG(E.salary) AS AvgSalary

FROM aircraft A,certified C,employee E

WHERE A.aid = C.aid AND C.eid = E.eid AND A.cruising\_range>1000

GROUP BY A.aid, A.aname )Temp;

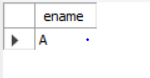


**v)Find the names of pilots certified for ‘300’ named aircraft.**

SELECT DISTINCT E.ename

FROM employee E, certified C, aircraft A

WHERE E.eid = C.eid AND C.aid = A.aid AND A.aname LIKE '300';



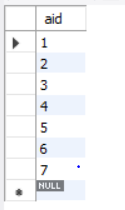
**vi)Find the aids of all aircraft that can be used on routes from tumkut to hubli.**

SELECT A.aid

FROM aircraft A

WHERE A.cruising\_range>(SELECT MIN(F.distance)

FROM flights F WHERE F.ffrom ='tumkur' AND F.tto='hubli');



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

SELECT F.departs

FROM flights F

WHERE F.flno IN (( SELECT F0.flno

FROM flights F0

WHERE F0.ffrom ='hubli' AND F0.tto = 'bangalore'

AND extract(hour from F0.arrives)<18 )

UNION

( SELECT F0.flno

FROM flights F0,flights F1

WHERE F0.ffrom = 'hubli' AND F0.tto <>'bangalore'

AND F0.tto = F1.ffrom AND F1.tto ='bangalore'

AND F1.departs > F0.arrives

AND extract(hour from F1.arrives) < 18)

UNION

( SELECT F0.flno

FROM flights F0,flights F1,flights F2

WHERE F0.ffrom ='hubli'

AND F0.tto = F1.ffrom

AND F1.tto = F2.ffrom

AND F2.tto = 'bangalore'

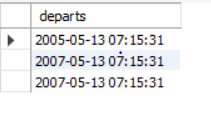
AND F0.tto <> 'bangalore'

AND F1.tto <> 'bangalore'

AND F1.departs > F0.arrives

AND F2.departs > F1.arrives

AND extract(hour from F2.arrives) < 18));



**viii. Print the name and salary of every non-pilot whose salary is more than the average salary for pilots**

SELECT E.ename, E.salary

FROM employee E

WHERE E.eid NOT IN( SELECT DISTINCT C.eid

FROM certified C)

AND E.salary>(SELECT AVG(E1.salary)

FROM employee E1

WHERE E1.eid IN

( SELECT DISTINCT C1.eid

FROM certified C1 ) );

