**Ashwini**

**1BM20CS402**

**CSE-4A**

**PROGRAM 1: INSURANCE DATABASE**

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

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

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

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

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

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

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

ii. Enter at least five tuples for each relation.

iii. Demonstrate how you

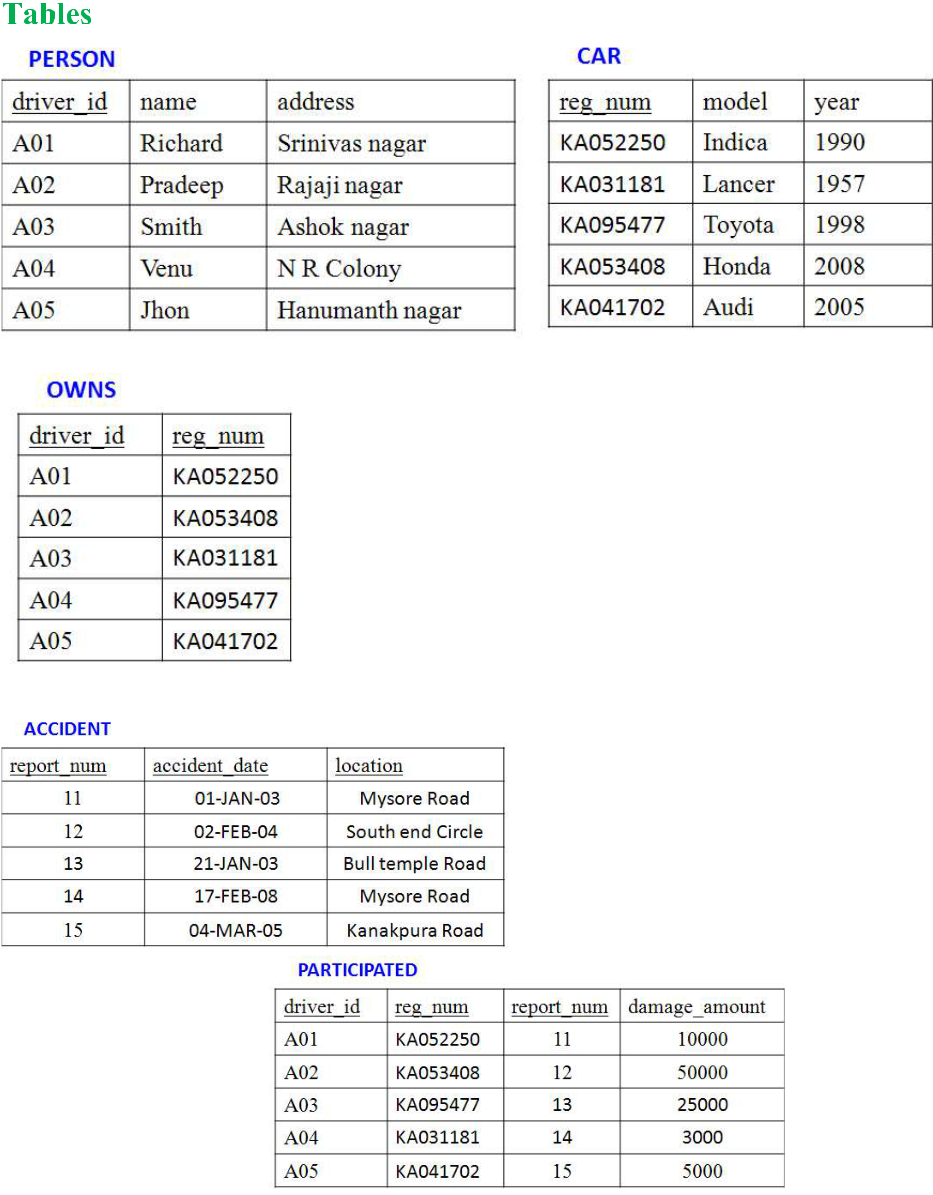
a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to

25000.

b. Add a new accident to the database.

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

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



CREATE DATABASE INSURANCE\_DATABASE;

USE INSURANCE\_DATABASE;

create table person

(

driver\_ id varchar(15) unique NOT NULL,

name varchar(20) NOT NULL,

address varchar(30),

primary key(driver\_id)

);

create table car

(

reg \_num varchar(20) unique NOT NULL,

model varchar(25),

year int,

primary key(reg\_num)

);

create table accident

(

report\_num int unique NOT NULL,

accident\_date date,

location varchar(30),

primary key(report\_num)

);

create table owns

(

driver\_id varchar(20),

reg\_num varchar(20),

FOREIGN KEY(driver\_id) REFERENCES person(driver\_id),

FOREIGN KEY(reg\_num) REFERENCES car(reg\_num)

);

create table participated

(

driver\_id varchar(15) unique NOT NULL,

reg\_num varchar(20) unique NOT NULL,

report\_num int unique NOT NULL,

damage\_amount int,

FOREIGN KEY(driver\_id) REFERENCES person(driver\_id),

FOREIGN KEY(reg\_num) REFERENCES car(reg\_num),

FOREIGN KEY(report\_num) REFERENCES accident(report\_num)

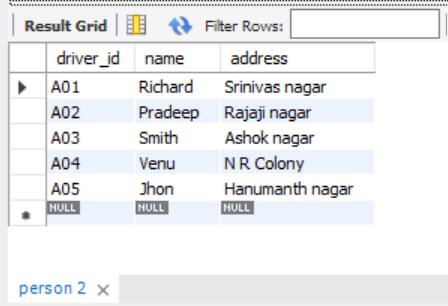
);

insert into person

values ("A01","Richard","Srinivas nagar"),("A02","Pradeep","Rajaji nagar"),

("A03","Smith","Ashok nagar"),("A04","Venu","N R Colony"),("A05","Jhon","Hanumanth nagar");

select \* from person;

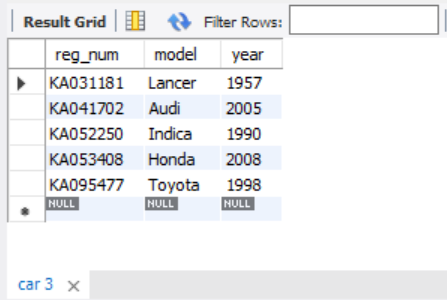


insert into car

values ("KA052250","Indica",1990),("KA031181","Lancer",1957),("KA095477","Toyota",1998),

("KA053408","Honda",2008),("KA041702","Audi",2005);

select \* from car;

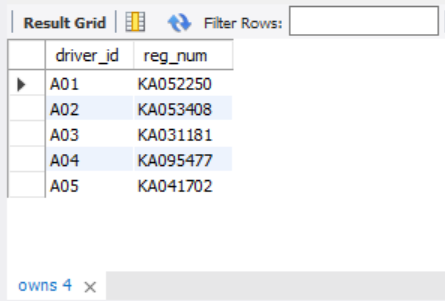


insert into owns

values ("A01","KA052250"),("A02","KA053408"),("A03","KA031181"),

("A04","KA095477"),("A05","KA041702");

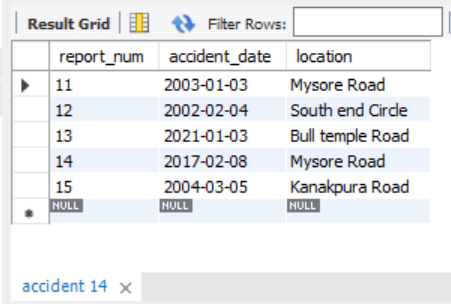
select \* from owns;



insert into accident

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

select \* from accident;

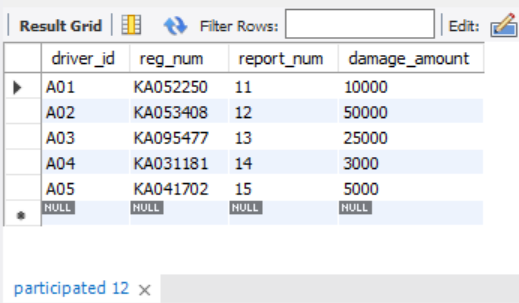


insert into participated

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

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

select \* from participated;



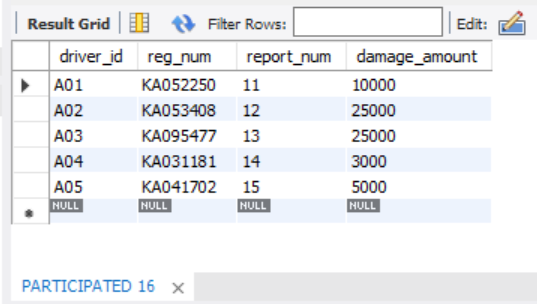
a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000

update PARTICIPATED

SET damage\_amount=25000

WHERE reg\_num="KA053408";

select\* from PARTICIPATED;

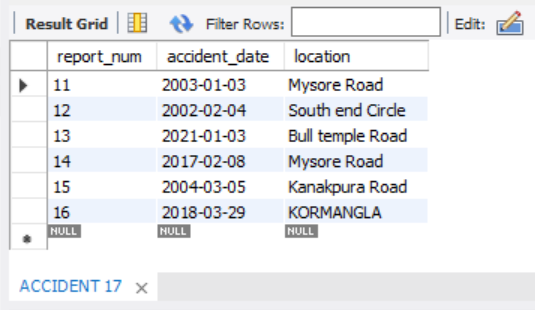


b. Add a new accident to the database.

insert into ACCIDENT

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

select\* from ACCIDENT;

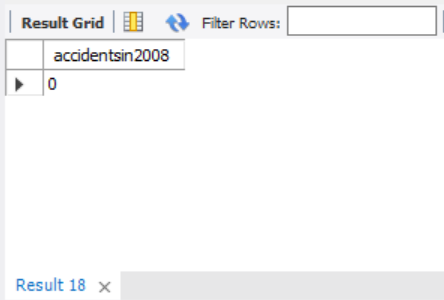


iv. Find the total number of people who owned cars that involved in accidents in 2008.

SELECT COUNT(accident\_date) AS accidentsin2008

FROM ACCIDENT

WHERE YEAR(accident\_date)=2008;

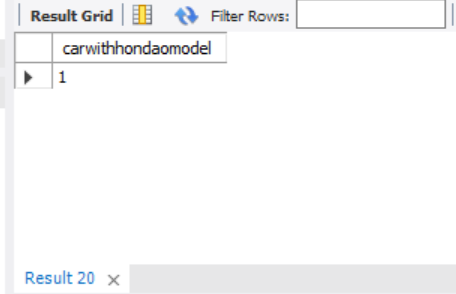


v. Find the number of accidents in which cars belonging to a specific model were involved

SELECT COUNT(model) AS carwithhondaomodel

FROM car

WHERE model="HONDA";



**PROGRAM 2: BANKING ENTERPRISE DATABASE**

Consider the following database for a banking enterprise.

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

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

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

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

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

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

foreign keys.

ii. Enter at least five tuples for each relation.

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

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

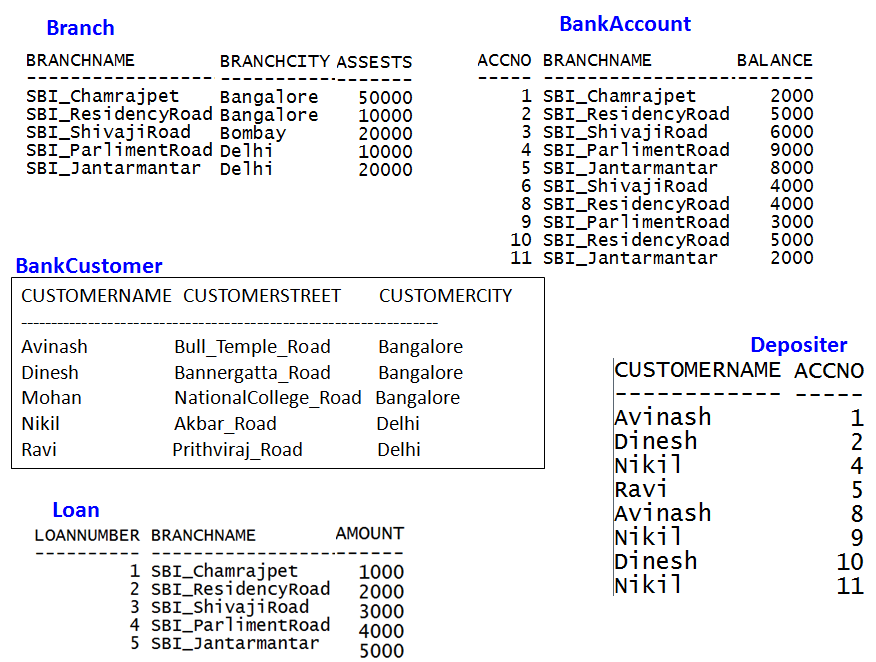
specific city (Ex. Delhi).

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

a specific city (Ex. Bombay).

**INTRODUCTION:** This database is developed for supporting banking facilities. Details of the branch along with the accounts and loans handled by them are recorded. Also details of the depositors of the corresponding branches are maintained.

**Sample Table data**



create database Bankingenterprise;

use Bankingenterprise;

CREATE TABLE BRANCH

(

branch\_name varchar(50),

branch\_city varchar(50),

assets real,

primary key(branch\_name)

);

CREATE TABLE BankAccount

(

accno int,

branch\_name varchar(50),

balance real,

primary key(accno),

FOREIGN KEY(branch\_name) REFERENCES BRANCH(branch\_name) ON DELETE SET NULL ON UPDATE CASCADE

);

CREATE TABLE BankCustomer

(

customer\_name varchar(50) primary key,

customer\_street varchar(50),

customer\_city varchar (50)

);

CREATE TABLE Depositor

(

customer\_name varchar(50),

accno int,

foreign key(accno) references BankAccount(accno),

foreign key(customer\_name) references BankCustomer(customer\_name)

);

CREATE TABLE Loan

(

loan\_number int,

branch\_name varchar(50),

amount real,

primary key(loan\_number),

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

);

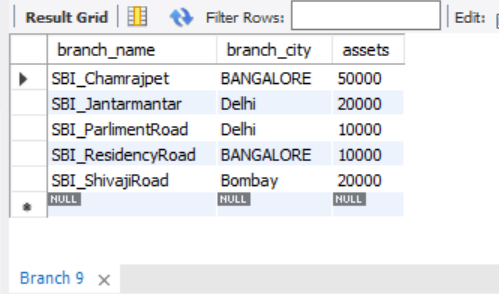
INSERT INTO Branch

VALUES ("SBI\_Chamrajpet", "BANGALORE",50000),("SBI\_ResidencyRoad", "BANGALORE",10000),

("SBI\_ShivajiRoad", "Bombay",20000),("SBI\_ParlimentRoad", "Delhi",10000),

("SBI\_Jantarmantar", "Delhi",20000);

select \* from Branch;



INSERT INTO BankAccount

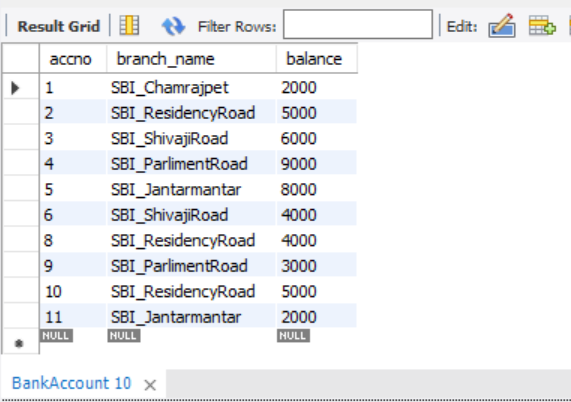
VALUES (1,"SBI\_Chamrajpet",2000),(2,"SBI\_ResidencyRoad",5000),

(3,"SBI\_ShivajiRoad",6000),(4,"SBI\_ParlimentRoad",9000),(5,"SBI\_Jantarmantar",8000),

(6,"SBI\_ShivajiRoad",4000),(8,"SBI\_ResidencyRoad",4000),(9,"SBI\_ParlimentRoad",3000),

(10,"SBI\_ResidencyRoad",5000),(11,"SBI\_Jantarmantar",2000);

select \* from BankAccount;



INSERT INTO BankCustomer

VALUES ("Avinash","Bull\_Temple\_Road","Bangalore");

INSERT INTO BankCustomer

VALUES("Dinesh","Bannergatta\_Road","Bangalore");

INSERT INTO BankCustomer

VALUES ("Mohan","NationtalCollege\_\_Road","Bangalore");

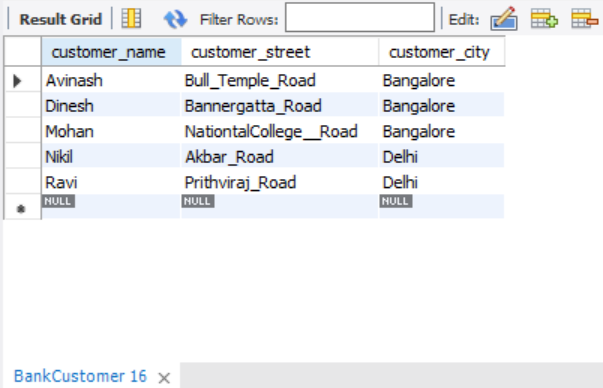
INSERT INTO BankCustomer

VALUES("Nikil","Akbar\_Road","Delhi");

INSERT INTO BankCustomer

VALUES ("Ravi","Prithviraj\_Road","Delhi");

select \* from BankCustomer;

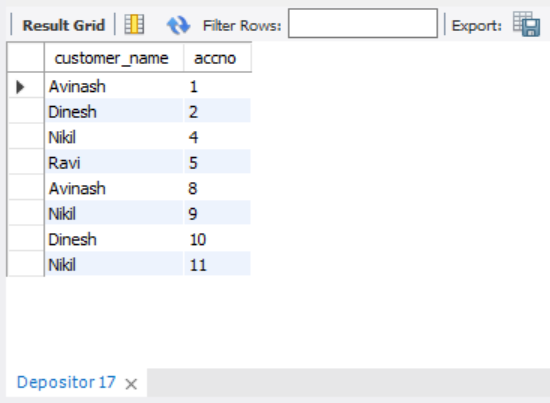


INSERT INTO Depositor

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

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

select \* from Depositor;

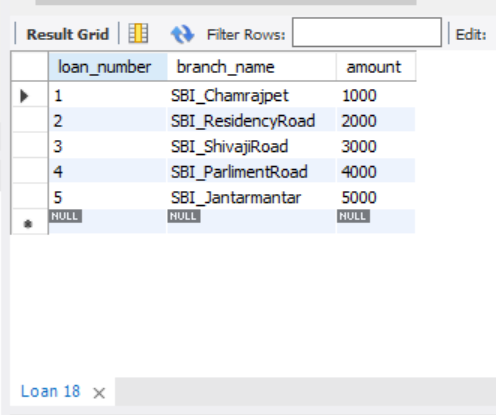


INSERT INTO Loan

VALUES (1,"SBI\_Chamrajpet",1000),(2,"SBI\_ResidencyRoad",2000),(3,"SBI\_ShivajiRoad",3000),

(4,"SBI\_ParlimentRoad",4000),(5,"SBI\_Jantarmantar",5000);

select \* from Loan;



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

SELECT \*

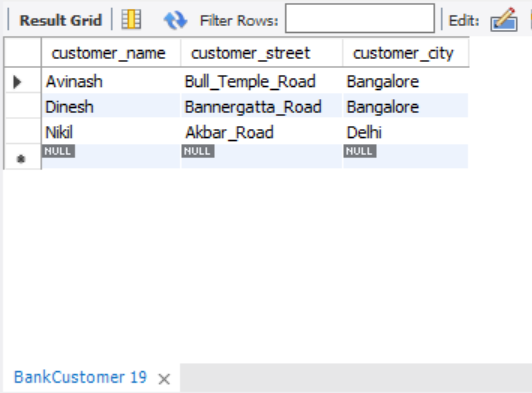
FROM BankCustomer

WHERE customer\_name IN ( SELECT customer\_name

FROM depositor

group by customer\_name

having COUNT(customer\_name)>=2);



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

specific city (Ex. Delhi).

SELECT d.customer\_name

FROM BankAccount a, Depositor d, Branch b

WHERE d.accno=a.accno AND b.branch\_name=a.branch\_name AND b.branch\_city="Bangalore"

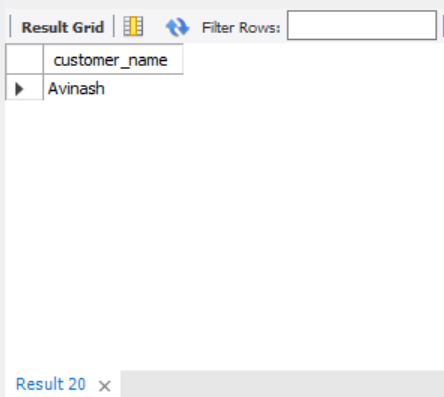
GROUP BY d.customer\_name

HAVING count(distinct b.branch\_name)=

(SELECT COUNT(branch\_name)

FROM branch

WHERE branch\_city="Bangalore");



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

a specific city (Ex. Bombay).

DELETE FROM depositor

WHERE accno IN

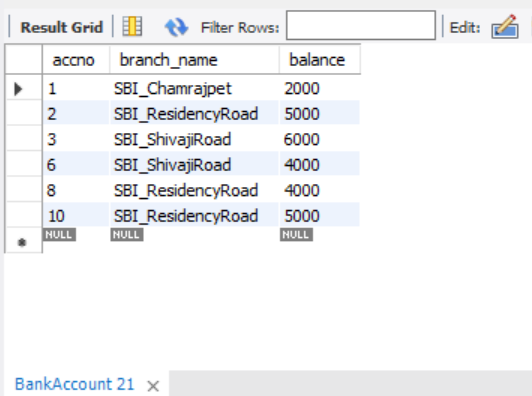
(SELECT accno

FROM Branch b, BankAccount a

WHERE branch\_city = 'delhi' and b.branch\_name = a.branch\_name);

DELETE FROM BankAccount WHERE branch\_name IN(SELECT branch\_name FROM BRANCH WHERE branch\_city='delhi');

SELECT \* FROM BankAccount;



**PROGRAM 3: SUPPLIER DATABASE**

**Consider the following schema:**

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

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

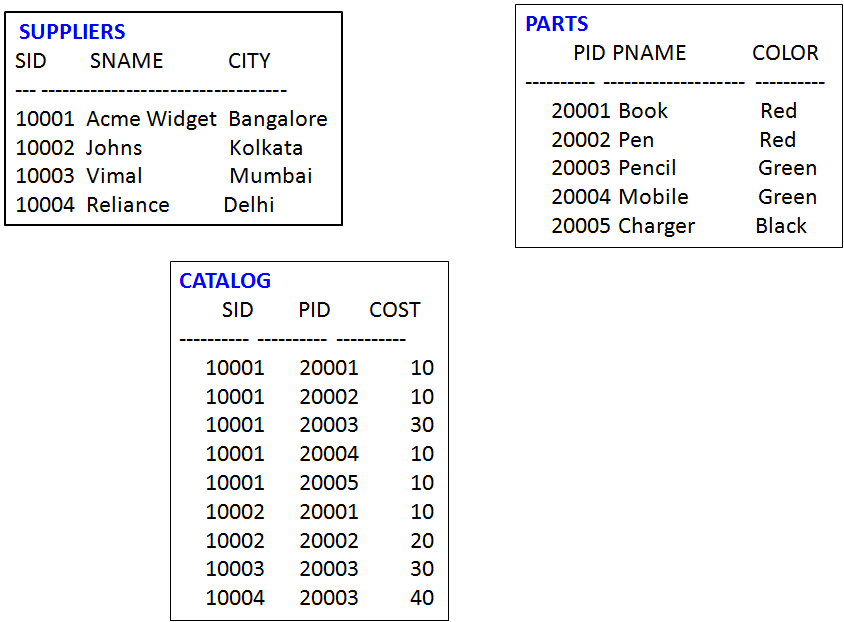
**CATALOG(sid: integer, pid: integer, cost: real)**

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

**Write the following queries in SQL:**

1. Find the pnames of parts for which there is some supplier.
2. Find the snames of suppliers who supply every part.
3. Find the snames of suppliers who supply every red part.
4. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
5. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
6. For each part, find the sname of the supplier who charges the most for that part.

**Table Data**

****

CREATE DATABASE Supplier\_Database1;

USE Supplier\_Database1;

CREATE TABLE SUPPLIER

(

sid int,

sname varchar(20),

address varchar(40),

primary key(sid)

);

CREATE TABLE PARTS

(

pid int,

pname varchar(40),

color varchar(20),

primary key(pid)

);

CREATE TABLE CATALOG

(

sid int,

pid int,

cost real,

FOREIGN KEY(sid) REFERENCES SUPPLIER(sid),

FOREIGN KEY(pid) REFERENCES PARTS(pid)

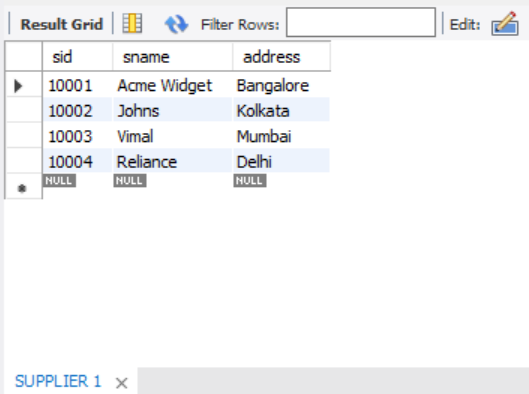
);

INSERT INTO SUPPLIER

VALUES (10001,"Acme Widget","Bangalore"),(10002,"Johns","Kolkata"),(10003,"Vimal","Mumbai"),

(10004,"Reliance","Delhi");

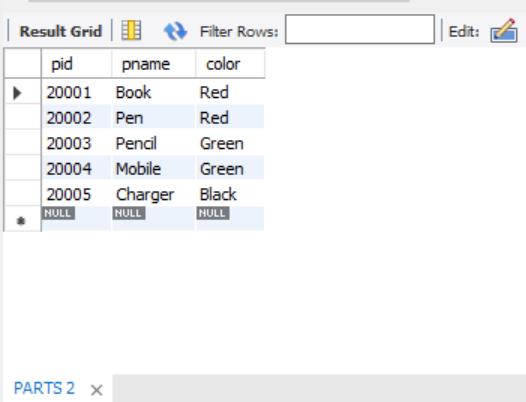
SELECT \* FROM SUPPLIER;



INSERT INTO PARTS

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

SELECT \* FROM PARTS;

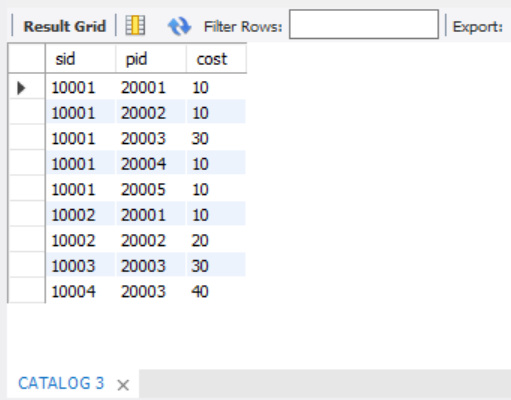


INSERT INTO CATALOG

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

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

SELECT \* FROM CATALOG;

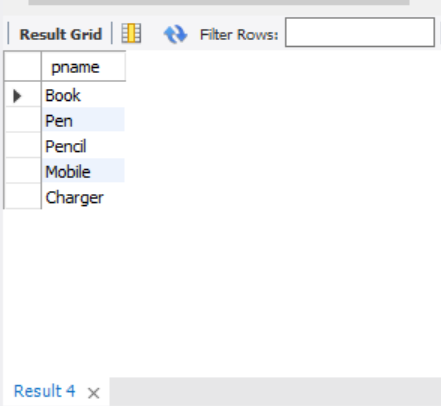


a.Find the pnames of parts for which there is some supplier.

SELECT DISTINCT p.pname

FROM parts p, catalog c

WHERE p.pid = c.pid;



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

SELECT SUPPLIER.sname, CATALOG.sid

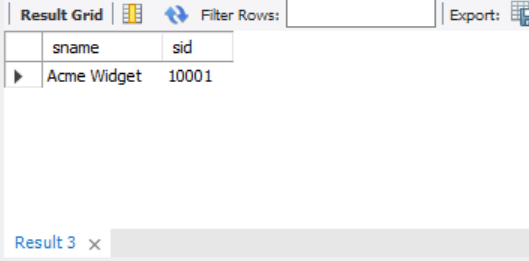
FROM SUPPLIER, CATALOG

WHERE SUPPLIER.sid = CATALOG.sid

group by SUPPLIER.sname

having count(CATALOG.sid)=(select count(pid)

from PARTS);

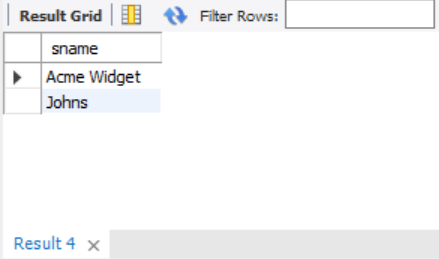


c.Find the snames of suppliers who supply every red part.

SELECT distinct s.sname

FROM SUPPLIER s, CATALOG c

WHERE s.sid = c.sid and c.pid in (select pid from PARTS WHERE color="Red");

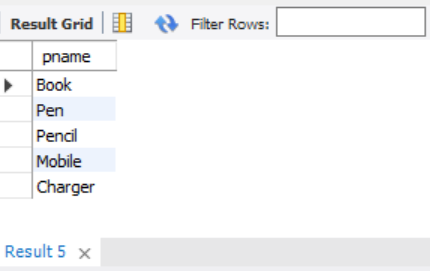


d. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.

SELECT pname

FROM PARTS P , CATALOG C

WHERE p.pid = c.pid AND sid in (SELECT sid from SUPPLIER WHERE sname = "Acme Widget");



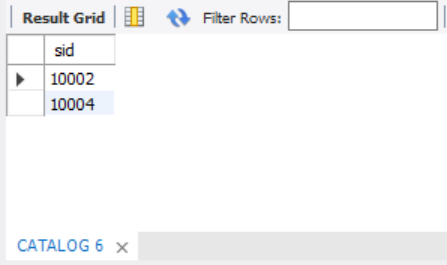
e. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).

SELECT c.sid

FROM CATALOG c WHERE c.cost > (SELECT AVG(cost)

FROM CATALOG

WHERE pid = c.pid) ;



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

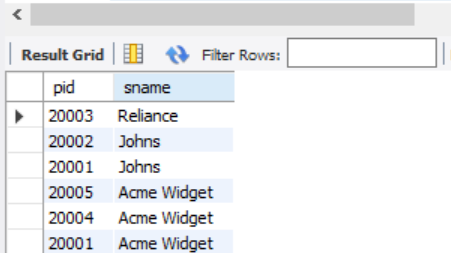
SELECT c.pid, s.sname

FROM PARTS p, SUPPLIER s, CATALOG c

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

FROM catalog

WHERE pid = c.pid);



**PROGRAM 4: STUDENT  FACULTY DATABASE**

**Consider the following database for**

**student enrollment for course :**

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

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

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

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

**The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character**

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

**Write the following queries in SQL.**

**No duplicates should be printed in any of the answers.**

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

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

Students enrolled.

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

time.

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

taught.

v. Find the names of faculty members for whom the combined enrollment of the courses

that they teach is less than five.

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

vii. For each age value that appears in Students, find the level value that appears most

often. For example, if there are more FR level students aged 18 than SR, JR, or

SO students aged 18, you should print the pair (18, FR).

**SQL> select \* from student;**

      SNUM SNAME      MA LV      AGE

---------- ---------- -- -- ----------

         1 jhon       CS Sr         19

         2 Smith      CS Jr          20

         3 Jacob      CV Sr         20

         4 Tom        CS Jr         20

         5 Rahul      CS Jr         20

         6 Rita       CS Sr         21

**SQL> select \* from faculty;**

       FID FNAME                    DEPTID

---------- -------------------- ----------

        11 Harish                     1000

        12 MV                         1000

        13 Mira                       1001

        14 Shiva                      1002

        15 Nupur                      1000

SQL> select \* from class;

CNAME     METTS\_A                        ROOM              FID

Class1         12/11/15 10:15:16.00000      R1                   14

Class10      12/11/15 10:15:16.00000      R128               14

Class2        12/11/15 10:15:20.000000    R2                 12

Class3        12/11/15 10:15:25.000000   R3                 11

Class4        12/11/15 20:15:20.000000   R4                 14

Class5        12/11/15 20:15:20.000000    R3                 15

Class6       12/11/15 13:20:20.000000    R2                 14

Class7       12/11/15 10:10:10.000000    R3                 14

**SQL> select \* from enrolled;**

      SNUM CNAME

---------- --------------------

         1 class1

         2 class1

         3 class3

         4 class3

         5 class4

CREATE DATABASE STUDENT\_FACULTY1;

USE STUDENT\_FACULTY1;

CREATE TABLE STUDENT

(

snum int,

sname varchar(40),

major varchar(30),

lvl varchar(20),

age int,

primary key(snum)

);

CREATE TABLE FACULTY

(

fid int,

fname varchar(40),

deptid int,

primary key(fid)

);

CREATE TABLE CLASS

(

cname varchar(40),

meetsat datetime;

room varchar(20),

fid int,

primary key(cname),

FOREIGN KEY(fid) REFERENCES FACULTY(fid)

);

CREATE TABLE ENROLLED

(

snum int,

cname varchar(40),

FOREIGN KEY(snum) REFERENCES STUDENT(snum),

FOREIGN KEY(cname) REFERENCES CLASS(cname)

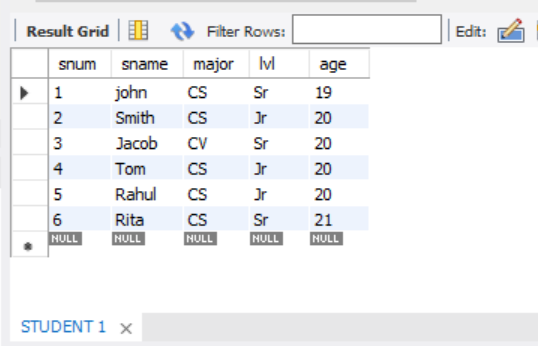
);

INSERT INTO STUDENT

VALUES (1,"john","CS","Sr",19),(2,"Smith","CS","Jr",20),(3,"Jacob","CV","Sr",20),

(4,"Tom","CS","Jr",20),(5,"Rahul","CS","Jr",20),(6,"Rita","CS","Sr",21);

SELECT \* FROM STUDENT;

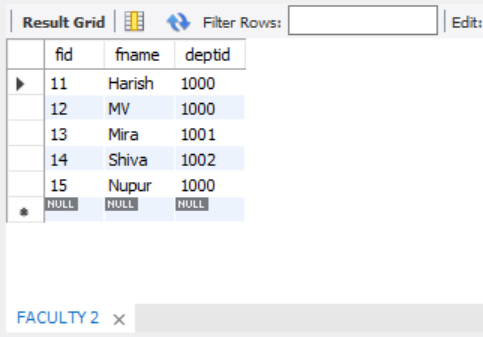


INSERT INTO FACULTY

VALUES (11,"Harish",1000),(12,"MV",1000),(13,"Mira",1001),(14,"Shiva",1002),

(15,"Nupur",1000);

SELECT \* FROM FACULTY;



INSERT INTO CLASS

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

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

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

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

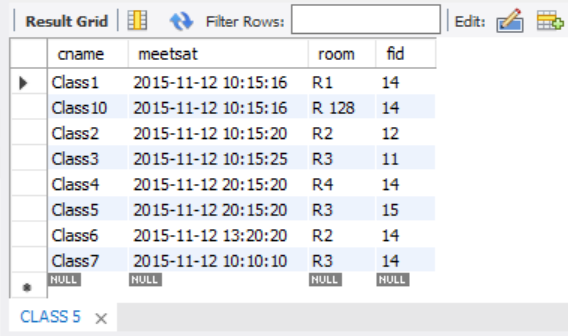
("Class4",'2015-11-12:20:15:20.00000',"R4",14),

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

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

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

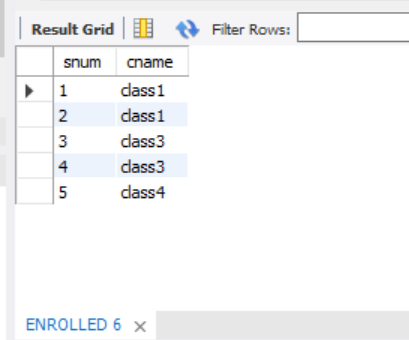
select \* from CLASS;



INSERT INTO ENROLLED

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

select \* from ENROLLED;



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

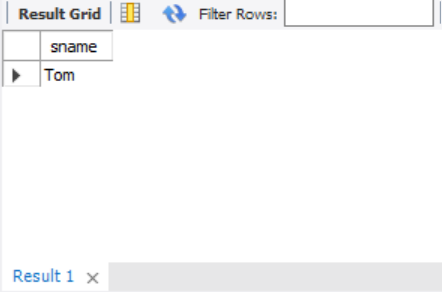
SELECT s.sname

FROM STUDENT s, CLASS c, ENROLLED e

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

from FACULTY

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



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

Students enrolled.

SELECT c.cname

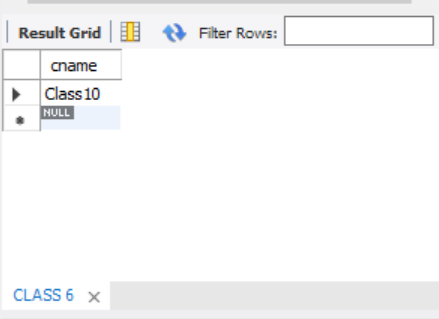
FROM CLASS c

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

from ENROLLED e

group by e.cname

having count(\*)>=5);



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

time.

SELECT distinct s.sname

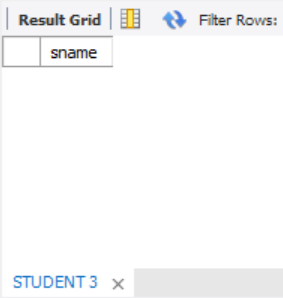
FROM STUDENT s

WHERE S.snum in (select e1.snum

from ENROLLED e1, ENROLLED e2, CLASS c1, CLASS c2

where e1.snum = e2.snum and e1.cname <> e2.cname and e1.cname = c1.cname

and e2.cname = c2.cname and c1.meetsat = c2.meetsat);



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

taught.

SELECT f.fname, c.fid

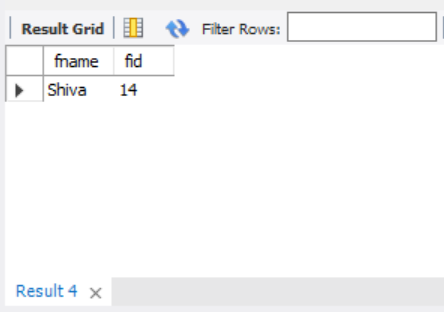
FROM FACULTY f, CLASS c

WHERE f.fid = c.fid

group by c.fid

having count(c.fid)=(select COUNT(distinct room)

from CLASS);



v. Find the names of faculty members for whom the combined enrollment of the courses

that they teach is less than five.

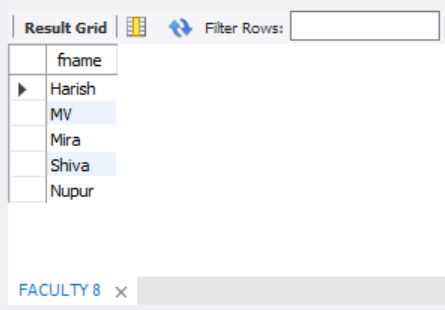
SELECT distinct fname

FROM FACULTY f

WHERE 5> (select COUNT(e.snum)

from ENROLLED e,CLASS c

where c.cname=e.cname and c.fid=f.fid);



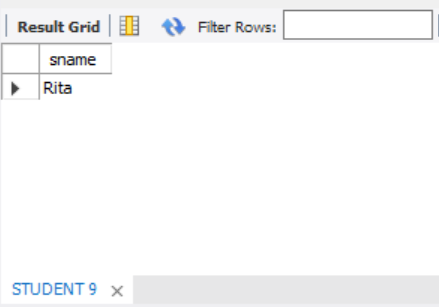
vi. Find the names of students who are not enrolled in any class.

SELECT s.sname

FROM STUDENT s

WHERE snum not in (select snum

from ENROLLED);



vii. For each age value that appears in Students, find the level value that appears most

often. For example, if there are more FR level students aged 18 than SR, JR, or

SO students aged 18, you should print the pair (18, FR).

SELECT s.age ,s.lvl

FROM STUDENT s

group by s.age

having s.lvl in (select s1.lvl

from STUDENT s1

where s1.age=s.age

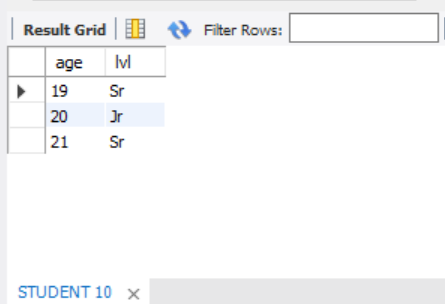
group by s1.age

having count(\*)>= all (select s2.lvl

from STUDENT s2

where s2.age=s1.age

group by s2.age));



**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.**

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.
2. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
4. For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
5. Find the names of pilots certified for some Boeing aircraft.
6. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
7. A customer wants to travel from Bangalore to Kolkata New with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in Kolkata by 6 p.m.

**SQL> select \* from Flights;**

 FLNO FFROM           TO               DISTANCE    DEPARTS    ARRIVES     PRICE

 101 Bangalore  Delhi  2500 13-MAY-05 07.15.31.000000 AM13-MAY-05 07.15.31.000000 A   5000

102 Bangalore  Lucknow               3000  05/05/13 07:15:31.000000 05/05/13 11:15:31.000000  6000

103 Lucknow         Delhi                   500 05/05/13 12:15:31.000000  05/05/13 17:15:31.000000  3000

107 Bangalore       Frankfurt             8000  05/05/13 07:15:31.000000  05/05/13 22:15:31.000000  60000

104 Bangalore       Frankfurt             8500  05/05/13 07:15:31.000000    05/05/13 23:15:31.00000   75000

105 Kolkata         Delhi                  3400   05/05/13 07:15:31.000000   05/05/13 09:15:31.000000  7000

**SQL> select \* from Aircraft;**

       AID ANAME      CRUISINGRANGE

---------- ---------- -------------

       101 747                 3000

       102 Boeing               900

       103 647                  800

       104 Dreamliner         10000

       105 Boeing              3500

       106 707                 1500

       107 Dream             120000

7 rows selected.

**SQL> select \* from Certified;**

       EID        AID

---------- ----------

       701        101

       701        102

       701        106

       701        105

       702        104

       703        104

       704        104

       702        107

       703        107

       704        107

       702        101

       EID        AID

---------- ----------

       703        105

       704        105

       705        103

14 rows selected.

**SQL> select \* from Employees;**

       EID ENAME               SALARY

---------- --------------- ----------

       701 A                    50000

       702 B                   100000

       703 C                   150000

       704 D                    90000

       705 E                    40000

       706 F                    60000

       707 G                    90000

7 rows selected.

CREATE DATABASE AIRLINE\_FLIGHT\_DATABASE;

USE AIRLINE\_FLIGHT\_DATABASE;

CREATE TABLE FLIGHTS

(

flno int,

ffrom varchar(40),

tto varchar(40),

distance int,

departs datetime,

arrives datetime,

price int,

primary key(flno)

);

CREATE TABLE AIRCRAFT

(

aid int,

aname varchar(40),

cruisingrange int,

primary key(aid)

);

CREATE TABLE EMPLOYEES

(

eid int,

ename varchar(40),

salary int,

primary key(eid)

);

CREATE TABLE CERTIFIED

(

eid int,

aid int,

FOREIGN KEY(aid) REFERENCES AIRCRAFT(aid),

FOREIGN KEY(eid) REFERENCES EMPLOYEES(eid)

);

INSERT INTO FLIGHTS

VALUES (101,"Bangalore","Delhi",2500,'2005-05-13:07:15:31.000000','2005-05-13:07:15:31.000000',5000),

(102,"Bangalore","Lucknow",3000,'2013-05-05:07:15:31.000000','2013-05-05:11:15:31.000000',6000),

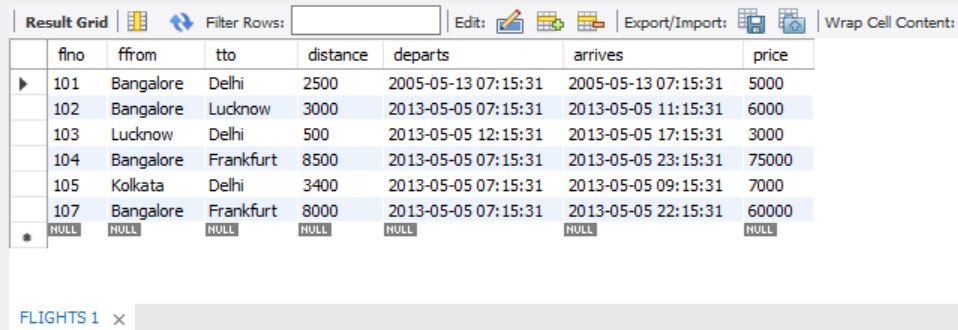
(103,"Lucknow","Delhi",500,'2013-05-05:12:15:31.000000','2013-05-05:17:15:31.000000',3000),

(107,"Bangalore","Frankfurt",8000,'2013-05-05:07:15:31.000000','2013-05-05:22:15:31.000000',60000),

(104,"Bangalore","Frankfurt",8500,'2013-05-05:07:15:31.000000','2013-05-05:23:15:31.000000',75000),

(105,"Kolkata","Delhi",3400,'2013-05-05:07:15:31.000000','2013-05-05:09:15:31.000000',7000);

SELECT \* FROM FLIGHTS;

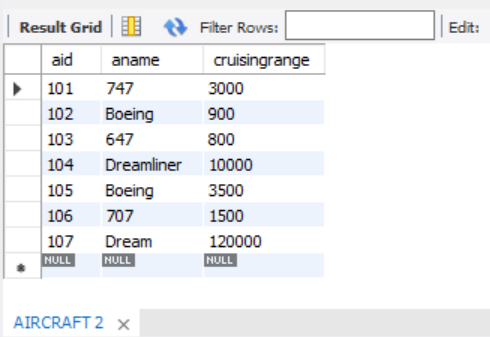


INSERT INTO AIRCRAFT

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

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

SELECT \* FROM AIRCRAFT;

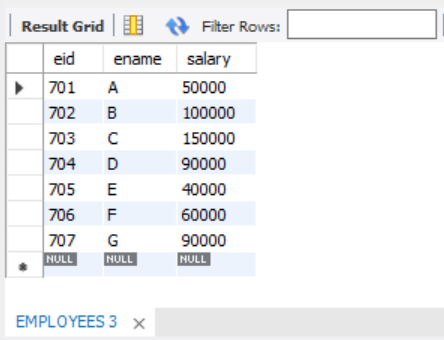


INSERT INTO EMPLOYEES

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

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

SELECT \* FROM EMPLOYEES;

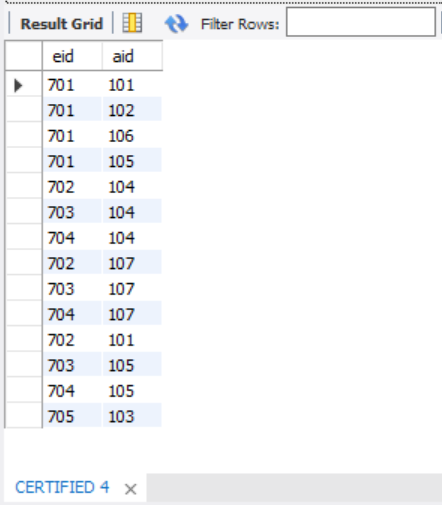


INSERT INTO CERTIFIED

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

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

SELECT \* FROM CERTIFIED;

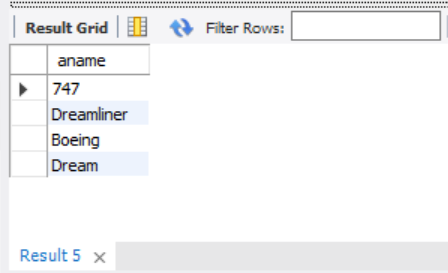


a.Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.

SELECT distinct a.aname

FROM AIRCRAFT a,EMPLOYEES e,CERTIFIED c

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



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

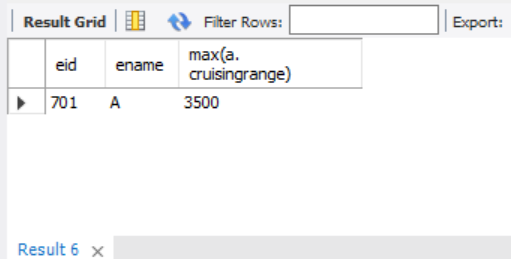
SELECT e.eid,e.ename,max(a. cruisingrange)

FROM EMPLOYEES e,CERTIFIED c,AIRCRAFT a

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

group by e.ename

having count(c.aid)>3;



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

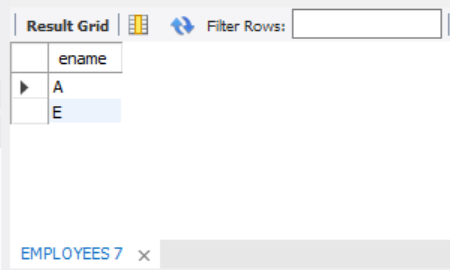
SELECT e.ename

FROM EMPLOYEES e

WHERE salary < (select min(price)

from FLIGHTS

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



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

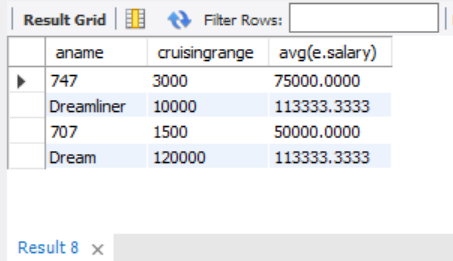
SELECT a.aname,a.cruisingrange,avg(e.salary)

FROM AIRCRAFT a,EMPLOYEES e,CERTIFIED c

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

group by a.aname

having a.cruisingrange > 1000;

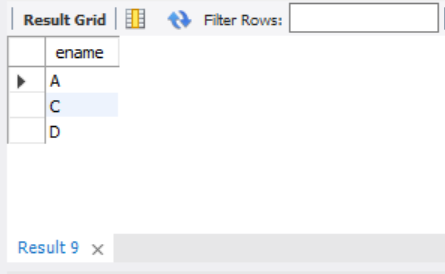


e.Find the names of pilots certified for some Boeing aircraft.

SELECT distinct e.ename

FROM EMPLOYEES e,CERTIFIED c,AIRCRAFT a

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



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

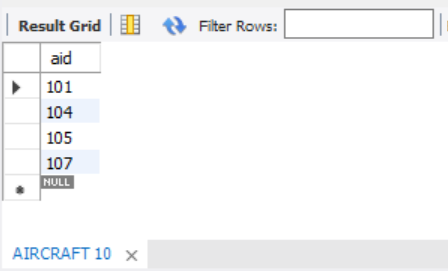
SELECT a.aid

FROM AIRCRAFT a

WHERE a. cruisingrange >= (select distance

from FLIGHTS

where ffrom="Bangalore" and tto="Delhi");



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

SELECT f.ffrom,f.tto,f.arrives

FROM FLIGHTS f

WHERE (f.ffrom="Bangalore" and f.tto=(select ffrom

from FLIGHTS

where tto="Kolkata")) or f.tto="Kolkata";

