

# **DBMS LAB RECORD**

NAME – SUDESHNA BHUSHAN

USN – 1BM19CS189

SEMESTER – 4

SECTION – D

## 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)

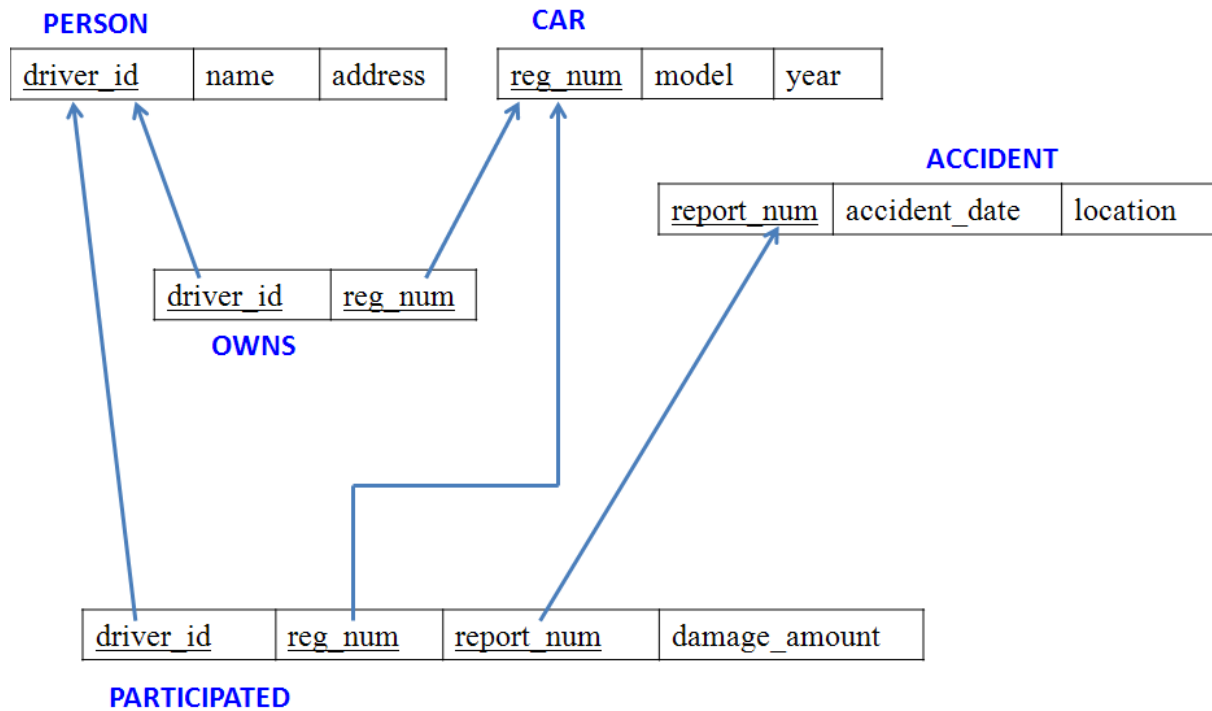
ACCIDENT (report\_num: int, accident\_date: date, location: String)

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.
- ii) Enter at least five tuples for each relation.
- iii) Demonstrate how you
  - a. Update the damage amount to 25000 for the car with a specific reg-num (example 'KA053408') for which the accident report number was 12.
  - 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 (example ) were involved.

### Schema diagram



## Tables

**PERSON**

<u>driver_id</u>	name	address
A01	Richard	Srinivas nagar
A02	Pradeep	Rajaji nagar
A03	Smith	Ashok nagar
A04	Venu	N R Colony
A05	Jhon	Hanumanth nagar

**CAR**

<u>reg_num</u>	model	year
KA052250	Indica	1990
KA031181	Lancer	1957
KA095477	Toyota	1998
KA053408	Honda	2008
KA041702	Audi	2005

**OWNS**

<u>driver_id</u>	<u>reg_num</u>
A01	KA052250
A02	KA053408
A03	KA031181
A04	KA095477
A05	KA041702

#### ACCIDENT

<u>report_num</u>	<u>accident_date</u>	<u>location</u>
11	01-JAN-03	Mysore Road
12	02-FEB-04	South end Circle
13	21-JAN-03	Bull temple Road
14	17-FEB-08	Mysore Road
15	04-MAR-05	Kanakpura Road

#### PARTICIPATED

<u>driver_id</u>	<u>reg_num</u>	<u>report_num</u>	<u>damage_amount</u>
A01	KA052250	11	10000
A02	KA053408	12	50000
A03	KA095477	13	25000
A04	KA031181	14	3000
A05	KA041702	15	5000

```
create database insurance;  
use insurance;
```

```
create table person(  
    driver_id varchar(10),  
    name varchar(20),  
    address varchar(30),  
    primary key(driver_id)  
);
```

```
desc person;
```

```
create table car(  
    reg_num varchar(10),  
    model varchar(10),  
    year int,  
    primary key(reg_num)  
);
```

```
desc car;
```

```
create table accident(  
    report_num int,  
    accident_date date,  
    location varchar(20),  
    primary key(report_num)  
);
```

```
desc accident;
```

```
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)
    );

desc owns;

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)
);

desc participated;

insert into person values('A01','Raghu','Electronic City');
insert into person values('A02','Rishab','Orange County');
insert into person values('A03','Rufus','NR Colony');
insert into person values('A04','Jamal','Lawrence Park');
insert into person values('A05','Kevin','Rosedale');

commit;

select * from person;

insert into car values('KA031111','Accord',2005);
insert into car values('KA041122','MX-5',2019);
insert into car values('KA051133','Indica',2010);
insert into car values('KA061144','Prius',2015);
insert into car values('KA071155','Camry',2020);
insert into car values('KA01010','Accord', 2002);
commit;

select * from car;

insert into accident values(111,'2020-01-01','NR Road');
insert into accident values(122,'2020-02-02','Dalhousie Road');
insert into accident values(133,'2020-03-03','Henry Road');
insert into accident values(144,'2020-04-04','Beehive Road');
insert into accident values(155,'2020-05-05','Orange Street');
insert into accident values(200, '2008-12-01', 'Pinto Road');
commit;

```

```
select * from accident;
```

```
insert into owns values ('A01','KA031111');  
insert into owns values ('A02','KA041122');  
insert into owns values ('A03','KA051133');  
insert into owns values ('A04','KA061144');  
insert into owns values ('A05','KA071155');  
insert into owns values('A02', 'KA01010');
```

```
commit;
```

```
select * from owns;
```

```
insert into participated values ('A01','KA031111',111, 10000);  
insert into participated values ('A02','KA041122',122, 20000);  
insert into participated values ('A03','KA051133',133, 30000);  
insert into participated values ('A04','KA061144',144, 40000);  
insert into participated values ('A05','KA071155',155, 50000);
```

```
insert into participated values('A02', 'KA01010', 200, 500);  
commit;
```

```
select * from participated;
```

```
-- Query 3a  
update participated  
set damage_amount = 2500  
where reg_num='KA031111';
```

```
select * from participated;  
-- Query 3b  
insert into accident values(101,'2020-12-01','Xavier Road');  
insert into participated values('A01','KA031111',101, 1001);  
commit;  
select * from accident;  
select * from participated;
```

```
-- Query 4  
select count(*) from accident where year(accident_date)=2008;
```

```
-- Query 5  
select count(*) from participated where reg_num in ( select reg_num  
from car where model="Accord");
```









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

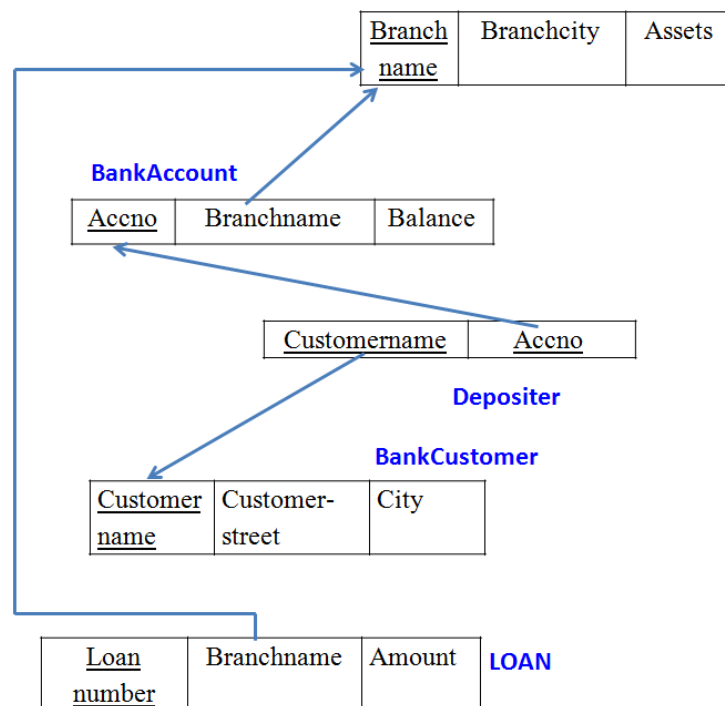
**Depositer**(customer-name: String, accno: int)

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

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.
- Find all the customers who have at least two accounts at the *Main* branch (ex. SBI\_ResidencyRoad).
- Find all the customers who have an account at *all* the branches located in a specific city (Ex. Delhi).
- 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.

### Schema Diagram



### Branch

BRANCHNAME	BRANCHCITY	ASSETS
SBI_Chamrajpet	Bangalore	50000
SBI_ResidencyRoad	Bangalore	10000
SBI_ShivajiRoad	Bombay	20000
SBI_ParlimentRoad	Delhi	10000
SBI_Jantarmanatar	Delhi	20000

### BankAccount

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

### BankCustomer

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

### Depositer

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

### Loan

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

create database bank;

use bank;

```
create table branch (  
    branch_name varchar(25),  
    branch_city varchar(15),  
    assets int,  
    primary key (branch_name)  
);
```

```
create table bank_account (  
    accno int,  
    branch_name varchar(25),  
    balance int,  
    primary key (accno),  
    foreign key (branch_name) references branch(branch_name)  
);
```

```
create table bank_customer (  
    customer_name varchar(10),  
    customer_street varchar(25),  
    customer_city varchar(15),  
    primary key (customer_name)
```

);

```
create table depositer (  
    customer_name varchar(10),  
    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(25),  
    amount int,  
    primary key (loan_number),  
    foreign key (branch_name) references branch(branch_name)  
);
```

```
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_Jantarmanatar', 'Delhi', 20000);  
commit;
```

```
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_Jantarmanatar', 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_Jantarmanatar', 2000);  
commit;
```

```
insert into bank_customer values ('Avinash', 'Bull_Temple_Road', 'Bangalore');  
insert into bank_customer values ('Dinesh', 'Bannerghatta_Road', 'Bangalore');  
insert into bank_customer values ('Mohan', 'National_College_Road', 'Bangalore');  
insert into bank_customer values ('Nikhil', 'Akbar_Road', 'Delhi');  
insert into bank_customer values ('Ravi', 'Prithviraj_Road', 'Delhi');  
commit;
```

```
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);
commit;
```

```
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_Jantarmanatar', 5000);
commit;
```

```
select * from branch;
select * from bank_account;
select * from bank_customer;
select * from depositer;
select * from loan;
```

```
select distinct c.customer_name from bank_customer c, bank_account b where exists(select
d.customer_name, count(d.customer_name) from depositer d, bank_account ba where
ba.accno = d.accno and
c.customer_name = d.customer_name and ba.branch_name = 'SBI_ResidencyRoad' group by
d.customer_name having count(d.customer_name) >= 2);
```

```
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');
```

```
delete from bank_account where branch_name in (select branch_name from branch where
branch_city = 'Bombay');
```

**data**

[illegible]

100% 1:1				
Result Grid Filter Rows: Search Edit: Export/Import:				
customer_name	accno			
Avinash	1			
Dinesh	2			
Nikhil	4			
Ravi	5			
Avinash	8			
Nikhil	9			
Dinesh	10			
Nikhil	11			
HULL	HULL			
depositor 1		Apply Revert		
Action Output				
	Time	Action	Response	Duration / Fetch Time
✓	20 20:41:00	SELECT * FROM bank.loan LIMIT 0, 1000	5 row(s) returned	0.0018 sec / 0.00001...
✓	21 20:43:28	SELECT * FROM bank.bank_customer LIMIT 0, 1000	5 row(s) returned	0.0035 sec / 0.00001...

100% 1:1				
Result Grid Filter Rows: Search Edit: Export/Import:				
loan_number	branch_name	amount		
1	SBI_Chamrajpet	1000		
2	SBI_ResidencyRoad	2000		
3	SBI_ShivajiRoad	3000		
4	SBI_ParliamentRoad	4000		
5	SBI_Jantarmantra	5000		
HULL	HULL	HULL		
loan 1		Apply Revert		
Action Output				
	Time	Action	Response	Duration / Fetch Time
✓	20 20:41:00	SELECT * FROM bank.loan LIMIT 0, 1000	5 row(s) returned	0.0018 sec / 0.00001...
✓	21 20:43:28	SELECT * FROM bank.bank_customer LIMIT 0, 1000	5 row(s) returned	0.0035 sec / 0.00001...

### 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)

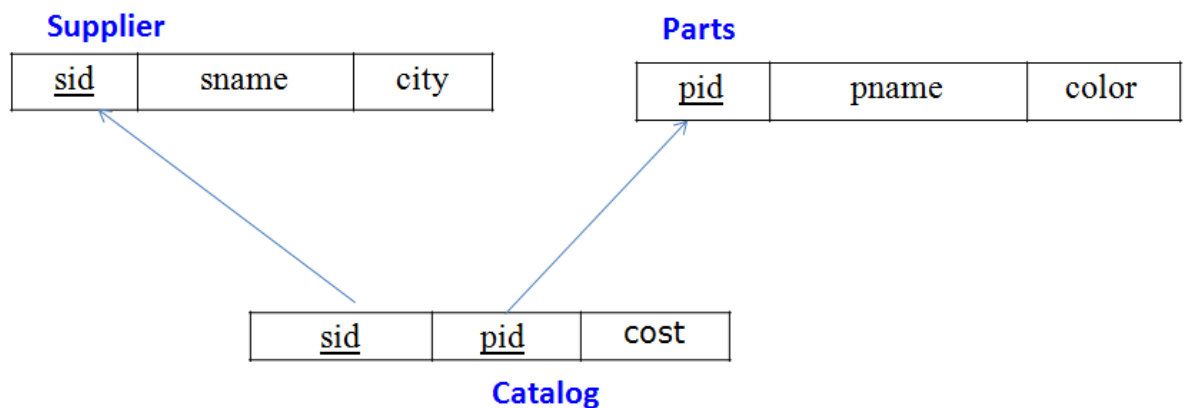
The Catalog relation lists the prices charged for parts by Suppliers.

Write the following queries in SQL:

- Find the pnames of parts for which there is some supplier.
- Find the snames of suppliers who supply every part.
- Find the snames of suppliers who supply every red part.

- iv) Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
- 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).
- vi) For each part, find the sname of the supplier who charges the most for that part.

### Schema Diagram



### Table Data

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

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

CATALOG		
SID	PID	COST
10001	20001	10
10001	20002	10
10001	20003	30
10001	20004	10
10001	20005	10
10002	20001	10
10002	20002	20
10003	20003	30
10004	20003	40

```

create database supplier;
use supplier;
create table suppliers(
    sid int primary key,

```



```

    sname varchar(30),
    address varchar(30)
);
create table parts(
    pid int primary key,
    pname varchar(30),
    color varchar(30)
);
create table catalog (
    sid int ,
    pid int ,
    cost real,
    constraint c_sid foreign key(sid) references suppliers(sid) ,
    constraint c_pid foreign key(pid) references parts(pid)
);
insert into suppliers values(1,'Acme Widget','kolkata') ;
insert into suppliers values(2,'Tata','bengaluru') ;
insert into suppliers values(3,'Reebok','delhi') ;
insert into suppliers values(4,'Nike','delhi') ;
insert into suppliers values(5,'Reliance','delhi') ;

```

```

insert into parts values(1,'paint','red') ;
insert into parts values(2,'steel','black') ;
insert into parts values(3,'spray','red') ;
insert into parts values(4,'sheet','green');
insert into parts values(5,'tiles','blue');
delete from parts where pid=5;

```

```

insert into catalog values(1,1,100);
insert into catalog values(1,2,200);
insert into catalog values(1,3,200);
insert into catalog values(1,4,100);
insert into catalog values(2,1,300);
insert into catalog values(2,2,100);
insert into catalog values(3,2,90);
insert into catalog values(3,3,110);
insert into catalog values(3,4,110);
insert into catalog values(4,1,100);
insert into catalog values(4,3,120);
insert into catalog values(4,4,130);

```

```

select * from catalog;
select * from parts;

```

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

```

insert into parts values(5,'tiles','blue');
select p.pname from parts p where p.pid in (select pid from catalog c group by c.pid
having count(c.sid)>0);
insert into catalog values(1,5,140);

```

```
select p.pname from parts p where p.pid in (select pid from catalog c group by c.pid
having count(c.sid)>0);
delete from catalog where pid=5;
delete from parts where pid=5;
```

-- ii. Find the snames of suppliers who supply every part.

```
select s.sname from suppliers s where s.sid in (select c.sid from catalog c group by c.sid
having count(distinct (c.pid))=(select count(p.pid) from parts p));
```

-- iii. Find the snames of suppliers who supply every red part.

```
select s.sname from suppliers s where s.sid in (select ca.sid from catalog ca,parts p
where ca.pid=p.pid and p.color='red' group by ca.sid having count(ca.pid)=(select
count(*) from parts p where p.color='red'));
```

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

```
select ca.pid from catalog ca where ca.sid=(select s.sid from suppliers s where s.sname
='Acme Widget') having (select count(c.pid) from catalog c where c.pid=ca.pid)=1;
```

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

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

```
select s.sname from suppliers s where s.sid in (select c.sid from catalog c where
c.cost=(select max(cost) from catalog ca where ca.pid=c.pid));
```

-- vii. select supplier who sell only red parts

```
select s.sname from suppliers s where s.sid in(select c.sid from catalog c where c.sid not
in (select distinct(ca.sid) from catalog ca,parts p where ca.pid=p.pid and
p.color!='red'));
```

```
insert into catalog values(5,1,140);
```

```
select s.sname from suppliers s where s.sid in(select c.sid from catalog c where c.sid not
in (select distinct(ca.sid) from catalog ca,parts p where ca.pid=p.pid and
p.color!='red'));
```

```
delete from catalog where sid=5;
```



sid	pid
1	2
1	3
2	1
4	4

sname
Acme Widget
Tata
Nike

sname
Reliance

## 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, meets at: 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).

```
create database studentdb;
```

```
use studentdb;
```

```
create table student(
```

```
    snum int,
```

```
    sname varchar(10),
```

```
    major varchar(2),
```

```
    lvl varchar(2),
```

```
    age int,
```

```
    primary key(snum)
```

```
);
```

desc student;

create table class(

    cname varchar(8),

    meetsat time,

    room varchar(5),

    fid int,

    primary key (cname)

);

desc class;

create table enrolled(

    snum int,

    cname varchar(10),

    primary key (snum, cname),

    foreign key (snum) references student(snum),

    foreign key (cname) references class(cname)

);

desc enrolled;

create table faculty(

    fid int,

    fname varchar(10),

    deptid int,

    primary key (fid)

);

```
desc faculty;
```

```
insert into student values(1,'John', 'CS','Jr',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 student values(7,'McGregor', 'CV','Sr',22);
```

```
insert into student values(8,'Smilga', 'CS','Jr',19);
```

```
insert into student values(9,'Price', 'CV','Sr',22);
```

```
commit;
```

```
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);
```

```
commit;
```

```
insert into class values('Class1', '10:15:15' , 'R1', 14);
```

```
insert into class values('Class10', '10:15:16' , 'R128', 11);
```

```
insert into class values('Class11', '10:15:16' , 'R1', 11);
```

```
insert into class values('Class3', '10:15:16' , 'R3', 11);
```

```
insert into class values('Class13', '10:15:16' , 'R2', 11);
```

```
insert into class values('Class12', '10:15:16' , 'R4', 11);
```

```
insert into class values('Class2', '10:15:20' , 'R2', 12);
```

```
insert into class values('Class3', '10:15:45' , 'R3', 11);
```

```
insert into class values('Class4', '10:15:20' , 'R4', 12);
```

```
insert into class values('Class5', '20:15:20' , 'R3', 15);
```

```
insert into class values('Class6', '13:20:20' , 'R2', 12);
```

```
insert into class values('Class7', '10:10:10' , 'R3', 12);
```

```
commit;
```

```
select * from class;
```

```
insert into enrolled values(1, 'Class1');
```

```
insert into enrolled values(2, 'Class1');
```

```
insert into enrolled values(6, 'Class1');
```

```
insert into enrolled values(7, 'Class1');
```

```
insert into enrolled values(8, 'Class1');
```

```
insert into enrolled values(3, 'Class3');
```

```
insert into enrolled values(4, 'Class2');
```

```
insert into enrolled values(5, 'Class4');
```

```
commit;
```

```
select * from student;
```



```
select * from faculty;
```

```
select * from class;
```

```
select * from enrolled;
```

```
-- Query 1
```

```
select sname from student where lvl='Jr' and snum in
```

```
    (select snum from enrolled where cname in
```

```
        (select cname from class where fid in
```

```
            (select fid from faculty where fname='Shiva')
```

```
    ));
```

```
-- Query 2
```

```
select cname from class where cname in(
```

```
select cname from class where room = 'R128') or cname in
```

```
(select cname from enrolled group by cname having count(cname)>=5);
```

```
-- Query 3
```

```
select sname from student where snum in(
```

```
select snum from enrolled where cname in(
```

```
select cname from class where meetsat in (select meetsat from class group by meetsat having  
count(meetsat)>1)));
```

-- Query 4

SELECT f.fname,f.fid

FROM faculty f

WHERE f.fid in ( SELECT fid FROM class

GROUP BY fid HAVING COUNT(\*)=(SELECT COUNT(DISTINCT  
room) FROM class) );

-- Query 5

select distinct fid from class where cname in (select cname from enrolled group by cname  
having count(cname)<5) or cname not in (select distinct cname from enrolled);

-- Query 6

select sname from student where snum not in (select distinct snum from enrolled);

-- Query 7

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.age, S1.lvl

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

```
WHERE S1.age=S2.age

GROUP BY S2.lvl, S2.age))

ORDER BY S.age;
```

```
GROUP BY S2.lvl, S2.age))

ORDER BY S.age;
```

ORDER BY S.age;

[illegible]



[illegible]

## PROGRAM 5: AIRLINE FLIGHT DATABASE

**Consider the following database that keeps track of airline flight information:**

**FLIGHTS(fno: 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.**

- i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.
- ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
- iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
- iv. For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

- v. Find the names of pilots certified for some Boeing aircraft.
- vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
- 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.

```
create database flightdb;
```

```
use flightdb;
```

```
create table flights(
```

```
    flno int,
```

```
    fromplace varchar(15),
```

```
    toplace varchar(15),
```

```
    distance int,
```

```
    departs datetime,
```

```
    arrives datetime,
```

```
    price int,
```

```
    primary key (flno)
```

```
);
```

```
desc flights;
```

```
create table aircraft(
```

```
    aid int,
```

```
    aname varchar(15),
```

```
    cruisingrange int,
```

```
        primary key (aid)

);

desc aircraft;

create table employees (

        eid int,

        ename varchar(15),

        salary int,

        primary key (eid)

);

desc employees;

create table certified (

        eid int,

        aid int,

        foreign key (eid) references employees(eid),

        foreign key (aid) references aircraft(aid)

);

desc certified;

insert into flights values(101, 'Bangalore', 'Delhi', 2500, '2005-05-13 07:15:31', '2005-05-13
18:15:31', 5000);

insert into flights values(102, 'Bangalore', 'Lucknow', 3000, '2013-05-05 07:15:31', '2013-05-
05 11:15:31', 6000);

insert into flights values(103, 'Lucknow', 'Delhi', 500, '2013-05-05 12:15:31', '2013-05-05
17:15:31', 3000);
```

```
insert into flights values(107, 'Bangalore', 'Frankfurt', 8000, '2013-05-05 07:15:31', '2013-05-05 22:15:31', 60000);
```

```
insert into flights values(104, 'Bangalore', 'Frankfurt', 8500, '2013-05-05 07:15:31', '2013-05-05 23:15:31', 75000);
```

```
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31', 7000);
```

```
insert into flights values(106, 'Bangalore', 'Kolkata', 1000, '2013-05-05 01:15:30', '2013-05-05 09:20:30', 10000);
```

```
insert into flights values(108, 'Lucknow', 'Kolkata', 1000, '2013-05-05 11:30:30', '2013-05-05 15:20:30', 10000);
```

```
commit;
```

```
select * from flights;
```

```
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', 120000);
```

```
insert into aircraft values(108, '707', 760);
```

```
insert into aircraft values(109, '747', 1000);
```



```
commit;
```

```
select * from aircraft;
```

```
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);
```

```
commit;
```

```
select * from employees;
```

```
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, 108);
```

```
insert into certified values(701, 109);
```

```
commit;
```

```
select * from certified;
```

```
-- Query 1
```

```
select distinct a.aname from aircraft a where a.aid in (
```

```
    select c.aid from certified c, employees e where
```

```
    c.aid = e.aid and not exists(
```

```
        select * from employees e1 where e1.aid=e.aid and e1.salary<80000
```

```
    )
```

```
);
```

```
-- Query 2
```

```
select max(a.cruisingrange), c.aid from certified c, aircraft a where c.aid = a.aid group by  
c.aid having count(c.aid)>3;
```

-- Query 3

```
select ename from employees where salary <(  
select min(price) from flights where fromplace='Bangalore' and toplace='Frankfurt');
```

-- Query 4

```
select avg(e.salary), c.aid from certified c, employees e where c.aid in(  
select aid from aircraft where cruisingrange>1000) and e.eid = c.eid group by c.aid;
```

-- Query 5

```
select ename from employees where eid in(  
select eid from certified where aid in(  
select aid from aircraft where aname = 'Boeing'));
```

-- Query 6

```
select aname from aircraft where cruisingrange > any (select distance from flights where  
fromplace='Bangalore' and toplace='Delhi');
```

-- Query 7

```
SELECT F.flno, F.departs  
  
FROM flights F
```

```

WHERE F.flno IN ( ( SELECT F0.flno

FROM flights F0

WHERE F0.fromplace = 'Bangalore' AND F0.toplace = 'Kolkata'

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

UNION

( SELECT F0.flno

FROM flights F0, flights F1

WHERE F0.fromplace = 'Bangalore' AND F0.toplace <> 'Kolkata'

AND F0.toplace = F1.fromplace AND F1.toplace = 'Kolkata'

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.fromplace = 'Bangalore'

AND F0.toplace = F1.fromplace

AND F1.toplace = F2.fromplace

AND F2.toplace = 'Kolkata'

AND F0.toplace <> 'Kolkata'

AND F1.toplace <> 'Kolkata'

AND F1.departs > F0.arrives

AND F2.departs > F1.arrives

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

```

Field	Type	Null	Key	Default	Extra	
► aid	int	NO	PRI	NULL		
aname	varchar(15)	YES		NULL		
cruisingrange	int	YES		NULL		

Result 2

aid	aname	cruisingrange	
► 101	747	3000	
102	Boeing	900	
103	647	800	
104	Dreamliner	10000	
105	Boeing	3500	
106	707	1500	
107	Dream	120000	
108	707	760	
109	747	1000	
NULL	NULL	NULL	

aircraft 6

Field	Type	Null	Key	Default	Extra	
► eid	int	YES	MUL	NULL		
aid	int	YES	MUL	NULL		

Result 4

eid	aid	
► 701	101	
701	102	
701	106	
701	105	
702	104	
703	104	
704	104	
702	107	
703	107	
704	107	

certified 9

Field	Type	Null	Key	Default	Extra	
► eid	int	NO	PRI	NULL		
ename	varchar(15)	YES		NULL		
salary	int	YES		NULL		

Result 3



