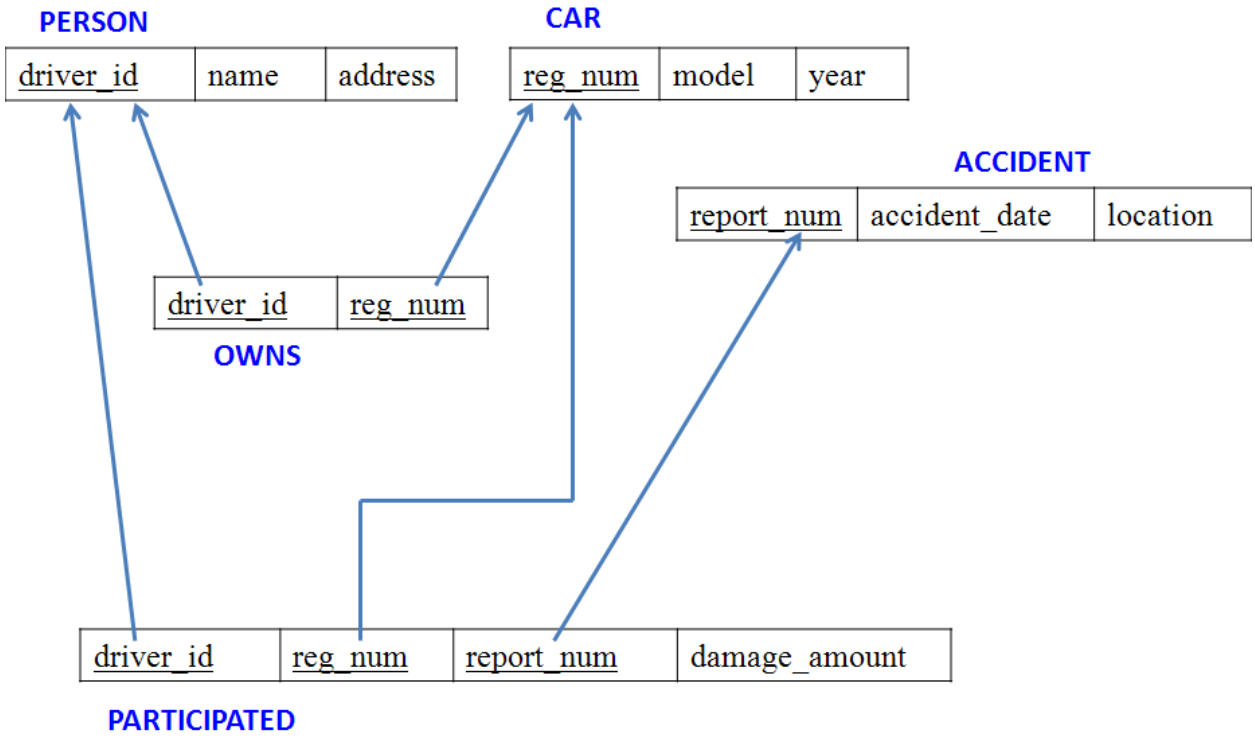
Medha Madhusudhan
1BM19CS230
CSE-4A

DBMS Record for Lab Test 1

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 'K A053408') 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 were involved in accidents in 2008.
v)Find the number of accidents in which cars belonging to a specific model (example)were involved.



```
create database Insurance;
use Insurance;

create table Person(
        driver_id varchar(20) not null,
        driver_name varchar(20) not null,
        address varchar(20) not null,
        primary key(driver_id)
);

create table Car(
        reg_num varchar(20) not null,
        model varchar(20) not null,
        year_purchase int not null,
        primary key(reg_num)
);

create table Accident(
        report_num int not null,
        accident_date date not null,
        location varchar(20) not null,
        primary key(report_num)
);

create table Owns(
        driver_id varchar(20) not null,
        reg_num varchar(20) not null,
        primary key(driver_id,reg_num),
        foreign key(driver_id) references Person(driver_id),
        foreign key(reg_num) references Car(reg_num)
);

create table Participated(
        driver_id varchar(20) not null,
        reg_num varchar(20) not null,
        report_num int not null,
        damage_amount int not null,
        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)
);

insert into Person(driver_id,driver_name,address)
        values ('A01','Richard','Srinivas Nagar'),
                ('A02','Pradeep','Rajajinagar'),
                ('A03','Smith','Ashok Nagar'),
```

```
                ('A04','Venu','NR Colony'),
                ('A05','Jhon','Hanumanth Nagar');

insert into Car(reg_num,model,year_purchase)
        values ('KA052250','Indica',1990),
                ('KA031181','Lancer',1957),
                ('KA095477','Toyota',1998),
                ('KA053408','Honda',2008),
                ('KA041702','Audi',2005);

insert into Owns(driver_id,reg_num)
        values ('A01','KA052250'),
                ('A02','KA053408'),
                ('A03','KA031181'),
                ('A04','KA095477'),
                ('A05','KA041702');

insert into Accident(report_num,accident_date,location)
        values (11,'2003-01-01','Mysore Road'),
                (12,'2004-02-02','South End Circle'),
                (13,'2003-01-21','Bull Temple Road'),
                (14,'2008-02-17','Mysore Road'),
                (15,'2005-03-04','Kanakpura Road');

insert into Participated(driver_id,reg_num,report_num,damage_amount)
        values ('A01','KA052250',11,10000),
                ('A02','KA053408',12,50000),
                ('A03','KA095477',13,25000),
                ('A04','KA031181',14,3000),
                ('A05','KA041702',15,5000);
```

-- demonstrate how you update damage amount to 25000 for reg number 'KA053408' and report num 12

```
update Participated
        set damage_amount = 25000 where reg_num = 'KA053408' and report_num = 12;
select * from Participated;
```

	driver_id	reg_num	report_num	damage_amount
▶	A01	KA052250	11	10000
	A02	KA053408	12	25000
	A03	KA095477	13	25000
	A04	KA031181	14	3000
	A05	KA041702	15	5000
*	NULL	NULL	NULL	NULL

-- add a new accident to the database

```
insert into Accident values (16,'2007-07-08','Jayanagar');
```

select * from Accident;

	report_num	accident_date	location
▶	11	2003-01-01	Mysore Road
	12	2004-02-02	South End Circle
	13	2003-01-21	Bull Temple Road
	14	2008-02-17	Mysore Road
	15	2005-03-04	Kanakpura Road
	16	2007-07-08	Jayanagar

-- find the total no. of people who owned cars involved in accident in 2008

select count(*) as accidents_2008 from Accident where accident_date between '2008-01-01' and '2008-12-31';

	accidents_2008
▶	1

-- find the number of accidents in which cars belonging to model 'Lancer' were involved

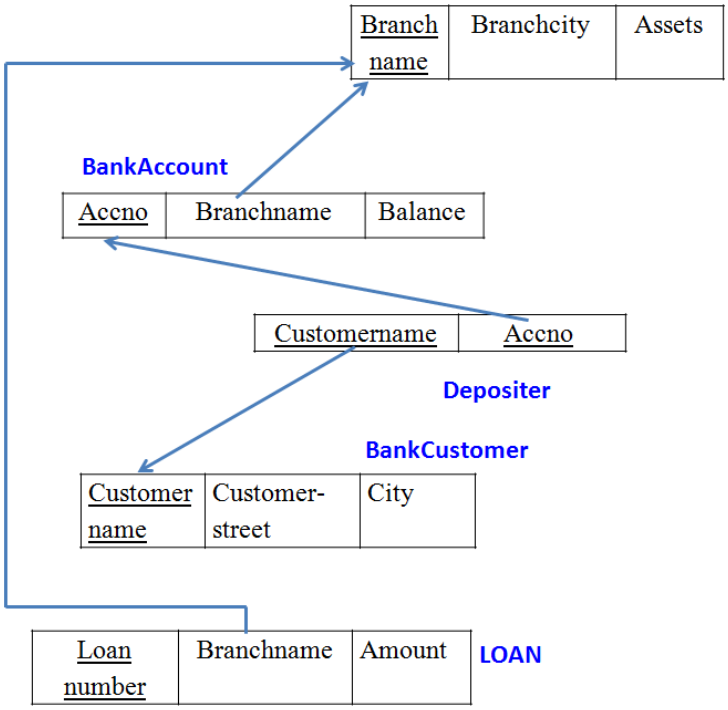
select count(p.report_num) from Participated p,Car c
where p.reg_num = c.reg_num and c.model = 'Lancer';

	count(p.report_num)
▶	1

Program 2 - Banking 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)
Depositor(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).




```
create database banking;
use banking;

create table Branch(
        branch_name varchar(20) not null,
        branch_city varchar(20) not null,
        assets real not null,
```

```
        primary key(branch_name)
);

create table BankAccount(
        accno int not null,
        branch_name varchar(20) not null,
        balance real not null,
        primary key(accno),
        foreign key(branch_name) references Branch(branch_name)
);

create table BankCustomer(
        customer_name varchar(20) not null,
        customer_street varchar(20) not null,
        customer_city varchar(20) not null,
        primary key(customer_name)
);

create table Depositor(
        customer_name varchar(20) not null,
        accno int not null,
        primary key(customer_name,accno),
        foreign key(customer_name) references BankCustomer(customer_name),
        foreign key(accno) references BankAccount(accno)
);

create table Loan(
        loanno int not null,
        branch_name varchar(20) not null,
        amount real not null,
        primary key(loanno),
        foreign key(branch_name) references Branch(branch_name)
);

insert into Branch(branch_name,branch_city,assets)
        values ('SBI_Chamrajpet','Bangalore',50000),
                ('SBI_ResidencyRoad','Bangalore',10000),
                ('SBI_ShivajiRoad','Bombay',20000),
                ('SBI_ParliamentRoad','Delhi',10000),
                ('SBI_Jantarmantar','Delhi',20000);
```

```
insert into BankAccount(accno,branch_name,balance)
        values (1,'SBI_Chamrajpet',2000),
                (2,'SBI_ResidencyRoad',5000),
                (3,'SBI_ShivajiRoad',6000),
                (4,'SBI_ParliamentRoad',9000),
                (5,'SBI_Jantarmantar',8000),
                (6,'SBI_ShivajiRoad',4000),
                (8,'SBI_ResidencyRoad',4000),
                (9,'SBI_ParliamentRoad',3000),
                (10,'SBI_ResidencyRoad',5000),
                (11,'SBI_Jantarmantar',2000);

insert into BankCustomer(customer_name,customer_street,customer_city)
        values ('Avinash','Bull_Temple_Road','Bangalore'),
                ('Dinesh','Bannerghatta_Road','Bangalore'),
                ('Mohan','NationalCollege_Road','Bangalore'),
                ('Nikil','Akbar_Road','Delhi'),
                ('Ravi','Prithviraj_Road','Delhi');

insert into Depositor(customer_name,accno)
        values ('Avinash',1),
                ('Dinesh',2),
                ('Nikil',4),
                ('Ravi',5),
                ('Avinash',8),
                ('Nikil',9),
                ('Dinesh',10),
                ('Nikil',11);

insert into Loan(loanno,branch_name,amount)
        values (1,'SBI_Chamrajpet',1000),
                (2,'SBI_ResidencyRoad',2000),
                (3,'SBI_ShivajiRoad',3000),
                (4,'SBI_ParliamentRoad',4000),
                (5,'SBI_Jantarmantar',5000);

-- Branch,BankAccount,BankCustomer,Depositor,Loan

-- find all the customers who have atleast two accounts at the 'SBI_ResidencyRoad'
select distinct d.customer_name from Depositor d,BankAccount ba,Branch b
```

```
where d.accno = ba.accno and ba.branch_name = b.branch_name and ba.branch_name =
'SBI_ResidencyRoad'
and 2 <= (select count(accno) from Depositor where customer_name = d.customer_name);
```

customer_name
Dinesh
Avinash

-- find all the customers who have an account at all branches in Delhi

```
select d.customer_name from Depositor d,BankAccount ba,Branch b
where d.accno = ba.accno and b.branch_name = ba.branch_name and b.branch_city =
'Delhi'
group by d.customer_name
having count(d.accno) >= (select count(branch_name) from Branch where branch_city =
'Delhi');
```

customer_name
Nikil

-- demonstrate how you delete all account tuples at every branch located at Bombay

```
delete from BankAccount
where branch_name in (select branch_name from Branch where branch_city =
'Bombay');
select * from BankAccount;
```

accno	branch_name	balance
1	SBI_Chamrajpet	2000
2	SBI_ResidencyRoad	5000
4	SBI_ParliamentRoad	9000
5	SBI_Jantarmantar	8000
8	SBI_ResidencyRoad	4000
9	SBI_ParliamentRoad	3000
10	SBI_ResidencyRoad	5000
11	SBI_Jantarmantar	2000

Program 3 - Suppliers 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:

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

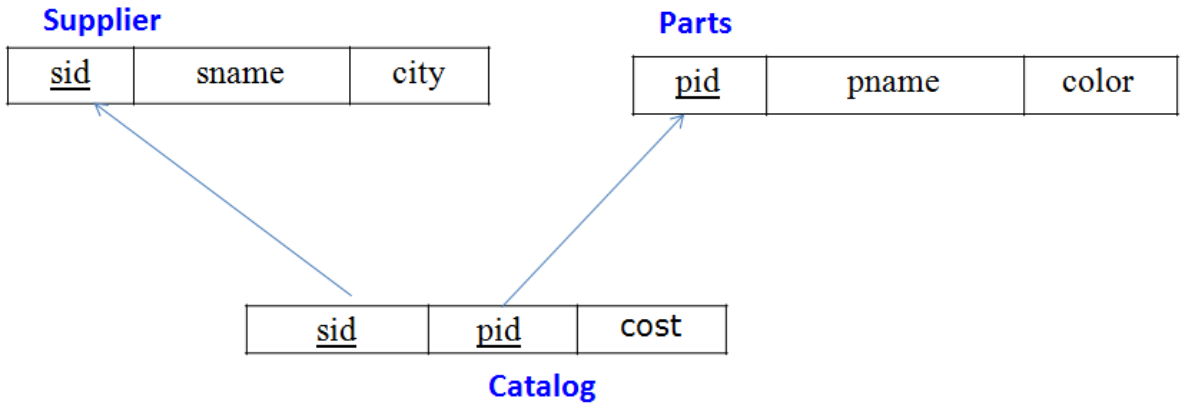
ii) Find the snames of suppliers who supply every part.

iii) 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.




```
create database supplierdb;
use supplierdb;

create table Suppliers(
        sid int not null,
        sname varchar(20) not null,
        address varchar(20) not null,
        primary key(sid)
);
```

```
create table Parts(
        pid int not null,
        pname varchar(20) not null,
        color varchar(20) not null,
        primary key(pid)
);

create table Catalog(
        sid int not null,
        pid int not null,
        cost real not null,
        primary key(sid,pid),
        foreign key(sid) references Suppliers(sid),
        foreign key(pid) references Parts(pid)
);

insert into Suppliers(sid,sname,address)
        values (10001,'Acme Widget','Bangalore'),
                (10002,'Johns','Kolkata'),
                (10003,'Vimal','Mumbai'),
                (10004,'Reliance','Delhi');

insert into Parts(pid,pname,color)
        values (20001,'Book','Red'),
                (20002,'Pen','Red'),
                (20003,'Pencil','Green'),
                (20004,'Mobile','Green'),
                (20005,'Charger','Black');

insert into Catalog(sid,pid,cost)
```

```
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);
```

-- 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 and exists (select 'X' from Catalog where pid = p.pid);
```

	pname
▶	Book
	Pen
	Pencil
	Mobile
	Charger

-- find the snames of suppliers who supply every part

```
select s.sname from Suppliers s,Catalog c
where s.sid = c.sid
group by s.sname
having count(c.pid) = (select count(*) from Parts);
```

	sname
▶	Acme Widget

-- find the snames of suppliers who supply every red part

```
select s.sname from Suppliers s,Parts p,Catalog c
where s.sid = c.sid and p.pid = c.pid
```

and p.color = 'Red'

group by s.sname

having count(c.pid) = (select count(*) from Parts where color = 'Red');

	sname
▶	Acme Widget
	Johns

-- find the pname of parts supplied by acme widget and no one else

select p.pname from Parts p,Catalog c

where p.pid = c.pid

and p.pid not in (select pid from Catalog where sid in (select sid from Suppliers where sname <> 'Acme Widget'));

	pname
▶	Mobile
	Charger

-- find the sids of suppliers who charge more for some part than average cost of that part

select distinct c.pid,c.sid from Catalog c

where c.cost > (select avg(c1.cost) from Catalog c1 where c1.pid = c.pid);

	pid	sid
▶	20002	10002
	20003	10004

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

select p.pname,s.sname from Parts p,Suppliers s,Catalog c

where p.pid = c.pid and s.sid = c.sid

and c.cost = (select max(cost) from Catalog where pid = p.pid);

pname	sname
Book	Acme Widget
Mobile	Acme Widget
Charger	Acme Widget
Book	Johns
Pen	Johns
Pencil	Reliance

Program 4 - Student 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.

```
create database studentfaculty;
use studentfaculty;


create table Student(
        snum int not null,
        sname varchar(10) not null,
```

```
        major varchar(2) not null,
        lvl varchar(2) not null,
        age int not null,
        primary key(snum)
);

create table Faculty(
        fid int not null,
        fname varchar(10) not null,
        deptid int not null,
        primary key(fid)
);

create table Class(
        cname varchar(10) not null,
        meetsat time not null,
        room varchar(10) not null,
        fid int not null,
        primary key(cname),
        foreign key(fid) references Faculty(fid)
);

create table Enrolled(
        snum int not null,
        cname varchar(10) not null,
        primary key(snum,cname),
        foreign key(snum) references Student(snum),
```

```
        foreign key(cname) references Class(cname)
);

insert into Student(snum,sname,major,lvl,age)
        values (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);

insert into Faculty(fid,fname,deptid)
        values (11,'Harish',1000),
                (12,'MV',1000),
                (13,'Mira',1001),
                (14,'Shiva',1002),
                (15,'Nupur',1000);

insert into Class(cname,meetsat,room,fid)
        values ('Class1','10:15:16','R1',14),
                ('Class10','10:15:16','R128',14),
                ('Class2','10:15:20','R2',12),
                ('Class3','10:15:25','R3',11),
                ('Class4','20:15:20','R4',14),
                ('Class5','20:15:20','R3',15),
                ('Class6','13:20:20','R2',14),
                ('Class7','10:10:10','R3',14);
```

```
insert into Enrolled(snum,cname)
        values (1,'Class1'),
                (2,'Class1'),
                (3,'Class3'),
                (4,'Class3'),
                (5,'Class4');


insert into Enrolled(snum,cname)
        values (1,'Class5'),
                (2,'Class5'),
                (3,'Class5'),
                (4,'Class5'),
                (5,'Class5');
```

-- find names of all juniors who are enrolled in a class taught by 'Harish'

```
select s.sname from Student s,Enrolled e
where s.snum = e.snum
and s.lvl = 'Jr'
and e.cname in (select cname from Class where fid = (select fid from Faculty where fname = 'Harish'));
```

	sname
▶	Tom

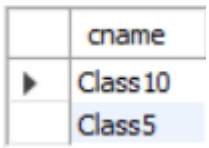
-- find the name of all classes that either meet in room128 or have >=5 students

```
select c.cname from Class c
where c.room = 'R128'
```

union

select distinct e.cname from Enrolled e

where 5 <= (select count(snum) from Enrolled where cname = e.cname);



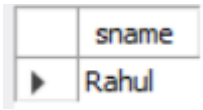
	cname
▶	Class10
	Class5

-- 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,Class c,Enrolled e

where s.snum = e.snum and c.cname = e.cname

and exists (select 'X' from Class c1,Enrolled e1 where c1.cname = e1.cname and c1.meetsat = c.meetsat and e1.snum = e.snum and c1.cname != e.cname);



	sname
▶	Rahul

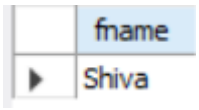
-- find the names of faculty who teach in every room in which class is taught

select f.fname from Faculty f,Class c

where f.fid = c.fid

group by f.fid

having count(f.fid) = (select count(distinct room) from Class);



	fname
▶	Shiva

-- find the names of faculty members for whom combined enrollment of courses that they teach is less than five

select distinct f.fname

from Class c,Faculty f

where c.fid = f.fid

and 5 > (select count(snum) from enrolled where cname in (select cname from Class where Class.fid = c.fid));

	fname
▶	Harish
	MV
	Shiva

-- find the names of students who are not enrolled in any class

select s.sname from Student s

where not exists (select 'X' from Enrolled where snum = s.snum);

	sname
▶	Rita

-- for each age value that appears in Students, find the level that appears the most

select s.age,s.lvl from Student s

where s.lvl = 'Jr'

group by s.age

having count(s.lvl) > (select count(lvl) from Student where lvl ='Sr' and age = s.age)

union

select s.age,s.lvl from Student s

where s.lvl = 'Sr'

group by s.age

having count(s.lvl) > (select count(lvl) from Student where lvl ='Jr' and age = s.age);

	age	lvl
▶	20	Jr
	19	Sr
	21	Sr

Program 5 - Airlines 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.

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 cruising range 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 airlinesdb;
use airlinesdb;

create table Flights(
        flno int not null,
        from_loc varchar(20) not null,
        to_loc varchar(20) not null,
        distance int not null,
        departs time not null,
```

```
        arrives time not null,
        price int not null,
        primary key(flno)
);

create table Aircraft(
        aid int not null,
        aname varchar(20) not null,
        cruisingrange int not null,
        primary key(aid)
);

create table Employees(
        eid int not null,
        ename varchar(20) not null,
        salary int not null,
        primary key(eid)
);

create table Certified(
        eid int not null,
        aid int not null,
        primary key(eid,aid),
        foreign key(eid) references Employees(eid),
        foreign key(aid) references Aircraft(aid)
);

insert into Flights(flno,from_loc,to_loc,distance,departs,arrives,price)
        values (101,"Bangalore","Delhi",2500,"07:15:31","12:15:31",5000),
                (102,"Bangalore","Lucknow",3000,"07:15:31","11:15:31",6000),
                (103,"Lucknow","Delhi",500,"12:15:31","17:15:31",3000),
                (107,"Bangalore","Frankfurt",8000,"07:15:31","22:15:31",60000),
                (104,"Bangalore","Frankfurt",8500,"07:15:31","23:15:31",75000),
                (105,"Kolkata","Delhi",3400,"07:15:31","09:15:31",7000);

insert into Flights(flno,from_loc,to_loc,distance,departs,arrives,price)
        values (106,"Delhi","Kolkata",3400,"12:15:35","14:20:00",7000);

insert into Aircraft(aid,aname,cruisingrange)
        values (101,"747",3000),
        (102,"Boeing",900),
        (103,"647",800),
        (104,"Dreamliner",10000),
        (105,"Boeing",3500),
        (106,"707",1500),
        (107,"Dream",12000);
```

```
insert into Employees(eid,ename,salary)
        values (701,'A',50000),
        (702,'B',100000),
        (703,'C',150000),
        (704,'D',90000),
        (705,'E',40000),
        (706,'F',60000),
        (707,'G',90000);

insert into Certified(eid,aid)
        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);
```

-- 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;
```

	aname
▶	747
	Dreamliner
	Boeing
	Dream

-- 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;
```

	eid	ename	max(a.cruisingrange)
▶	701	A	3500

-- 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
```

from_loc = "Bangalore" and to_loc = "Frankfurt");

	ename
▶	A
	E

-- 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 ;
```

	aname	cruisingrange	avg(e.salary)
▶	747	3000	75000.0000
	Dreamliner	10000	113333.3333
	707	1500	50000.0000
	Dream	12000	113333.3333

-- 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";
```

	ename
▶	A
	C
	D

-- 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
from_loc = "Bangalore" and to_loc = "Delhi");
```

	aid
▶	101
	104
	105
	107
*	NULL

-- a customer wants to travel from bangalore to kolkata with no more than two changes of flight list the choice of departure times customer wants to arrive by 6 p.m.

```
select f.from_loc,f.to_loc,f.arrives from Flights f
where (f.from_loc = "Bangalore" and f.to_loc = (select from_loc from Flights where
to_loc = "Kolkata")) or f.to_loc = "Kolkata";
```

	from_loc	to_loc	arrives
▶	Bangalore	Delhi	12:15:31
	Delhi	Kolkata	14:20:00