DBMS Lab report 1~ Mahantesh Gattina 1BM19CS219

Program 1 : INSURANCE DATABASE~

```
CREATE SCHEMA insurance;

CREATE TABLE person(
      driver_id CHAR(10) NOT NULL,
      name CHAR(20) NOT NULL,
      address VARCHAR(30) NOT NULL,
      PRIMARY KEY(driver_id)
      );

CREATE TABLE car(
      reg_num CHAR(10) NOT NULL,
      model CHAR(10) NOT NULL,
      year INT NOT NULL,
      PRIMARY KEY(reg_num)
);

CREATE table accident(
      report_num INT NOT NULL,
      accident_date DATE,
      location VARCHAR(30) NOT NULL,
      PRIMARY KEY(report_num)
);

CREATE TABLE owns(
      driver_id CHAR(10) NOT NULL,
      reg_num CHAR(10) NOT NULL
);

ALTER TABLE owns
ADD FOREIGN KEY(driver_id) REFERENCES person(driver_id),
ADD FOREIGN KEY(reg_num) REFERENCES car(reg_num);

CREATE TABLE participated(
      driver_id CHAR(10) NOT NULL,
      reg_num   CHAR(10) NOT NULL,
      report_num INT NOT NULL,
      damage_amount INT NOT NULL,
      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, name, address)
VALUES
('A01','DAVID','Srinivas Nagar'),
('A02','BRIAN','Ashok Nagar'),
('A03','DOUGH','Majestic'),
('A04','EMMA','Kadugodi'),
('A05','LLOYD','Malleshwaram');
```

```
INSERT INTO car(reg_num,model,year)
VALUES
('KA051234','Indica',2016),
('KA051235','Baleno',2017),
('KA051236','Tavera',2018),
('KA051237','Scorpio',2019),
('KA051238','Swift',2020);


INSERT INTO owns
(driver_id,reg_num)
VALUES
('A01','KA051234'),
('A02','KA051235'),
('A03','KA051236'),
('A04','KA051237'),
('A05','KA051238');
```

```
INSERT INTO participated (driver_id,reg_num,report_num,damage_amount) VALUES ('A01','KA051234',1,5000), ('A02','KA051235',2,10000), ('A03','KA051236',3,15000), ('A04','KA051237',4,20000), ('A05','KA051238',5,25000)
```

```
INSERT INTO person (driver_id, name, address) VALUES ('A06','RICHARD','ShivNagara')
INSERT INTO car (reg_num, model, year) VALUES ('KA051239','Tavera',2021)
INSERT INTO owns (driver_id, reg_num) VALUES ('A06', 'KA051239');
```

```
INSERT INTO accident (report_num, accident_date, location) VALUES (12, '2005-03-12', 'MG road')
INSERT  INTO participated
(driver_id, reg_num, report_num, damage_amount)
VALUES
('A06','KA051239',12,30000);
```

Dumping data for table accident

1	2003-01-01	Mysore Road
2	2005-01-02	Mangalore Road
3	2007-01-06	Ashok Nagar
4	2008-05-05	Nagarabhavi
5	2021-01-01	MG road
12	2005-03-12	MG road

Dumping data for table car

KA051234	Indica	2016

KA051235 Baleno 2017
KA051236 Tavera 2018
KA051237 Scorpio 2019
KA051238 Swift 2020
KA051239 Tavera 2021

Dumping data for table owns

A01 KA051234
A02 KA051235
A03 KA051236
A04 KA051237
A05 KA051238
A06 KA051239
A06 KA051239

Dumping data for table participated

A01 KA051234 1 5000
A02 KA051235 2 10000
A03 KA051236 3 15000
A04 KA051237 4 20000
A05 KA051238 5 25000
A06 KA051239 12 30000

Dumping data for table person

A01 DAVID Srinivas Nagar
A02 BRIAN Ashok Nagar
A03 DOUGH Majestic
A04 EMMA Kadugodi
A05 LLOYD Malleshwaram
A06 RICHARD ShivNagara

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 report_num = 12
```

```
AND reg_num = 'KA051239'
```

driver_id	reg_num	report_num	damage_amount
A01	KA051234	1	5000
A02	KA051235	2	10000
A03	KA051236	3	15000
A04	KA051237	4	20000
A05	KA051238	5	25000
A06	KA051239	12	25000

b. Add a new accident to the database.

```
INSERT INTO accident
(report_num, accident_date, location)
VALUES
(13,'2021-05-19','Ballari');
```

eport_num	accident_date
1	2003-01-01
2	2005-01-02
3	2007-01-06
4	2008-05-05
5	2021-01-01
12	2005-03-12
13	2021-05-19

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

COUNT(DISTINCT driver_id)
1

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

count(reg_num)
1

Mahantesh Gattina ~ 1BM19CS219

PROGRAM-2: BANKING ENTERPRISE DATABASE

```
CREATE DATABASE bank
CREATE TABLE branch( branch_name VARCHAR(20) NOT NULL, branch_city VARCHAR(20) NOT NULL, assets REAL NOT NULL, PRIMARY KEY(branch_name) )
desc branch
```

Field	Type	Null	Key	Default	Extra
branch_name	varchar(20)	NO	PRI	NULL	
branch_city	varchar(20)	NO		NULL	
assets	double	NO		NULL	

```
CREATE TABLE banck_account( 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) )
ALTER TABLE banck_account RENAME TO bank_account
desc bank_account
```

Field	Type	Null	Key	Default	Extra
accno	int(11)	NO	PRI	NULL	
branch_name	varchar(20)	NO	MUL	NULL	
balance	double	NO		NULL	

```
CREATE TABLE bank_customer( customer_name VARCHAR(20) NOT NULL, customer_street VARCHAR(30) NOT NULL, city VARCHAR(20) NOT NULL, PRIMARY KEY(customer_name) )
DESC bank_customer
```

Field	Type	Null	Key	Default	Extra
customer_name	varchar(20)	NO	PRI	NULL	

customer_street	varchar(30)	NO	NULL
city	varchar(20)	NO	NULL

```
CREATE TABLE depositor( customer_name VARCHAR(20) NOT NULL, accno INT NOT NUL
L, FOREIGN KEY(customer_name) REFERENCES bank_customer(customer_name), FOREIG
N KEY (accno) REFERENCES bank_account(accno) )
DESC depositor
```

Field	Type	Null	Key	Default	Extra
customer_name	varchar(20)	NO	MUL	NULL	
accno	int(11)	NO	MUL	NULL	

```
CREATE TABLE loan( loan_number INT NOT NULL, branch_name VARCHAR(20) NOT NUL
L, amount REAL NOT NULL, FOREIGN KEY(branch_name) REFERENCES branch(branch_na
me) )
DESC loan
```

Field	Type	Null	Key	Default	Extra
loan_number	int(11)	NO		NULL	
branch_name	varchar(20)	NO	MUL	NULL	
amount	double	NO		NULL	

```
INSERT INTO branch (branch_name,branch_city,assets) VALUES ('SBI_Chamrajpet','Bangal
ore',50000), ('SBI_ResidencyRoad','Bangalore',10000), ('SBI_ShivajiRoad','Bangalore',200
00), ('SBI_ParliamentRoad','Delhi',10000), ('SBI_Jantarmantar','Delhi',20000)
SELECT * FROM branch
```

←T→			branch_name	branch_city	assets
Edit	Copy	Delete	SBI_Chamrajpet	Bangalore	50000
Edit	Copy	Delete	SBI_Jantarmantar	Delhi	20000
Edit	Copy	Delete	SBI_ParliamentRoad	Delhi	10000
Edit	Copy	Delete	SBI_ResidencyRoad	Bangalore	10000
Edit	Copy	Delete	SBI_ShivajiRoad	Bangalore	20000

```
INSERT INTO loan (branch_name,loan_number,amount) VALUES ('SBI_Chamrajpet',1,1000), ('SBI_ResidencyRoad',2,2000), ('SBI_ShivajiRoad',3,3000), ('SBI_ParliamentRoad',4,4000), ('SBI_Jantarmantar',5,5000)
SELECT * FROM loan
```

loan_number	branch_name	amount
1	SBI_Chamrajpet	1000
2	SBI_ResidencyRoad	2000
3	SBI_ShivajiRoad	3000
4	SBI_ParliamentRoad	4000
5	SBI_Jantarmantar	5000

```
INSERT INTO bank_account (accno, branch_name, balance) VALUES (5,'SBI_Jantarmantar', 8000);
INSERT INTO bank_account (accno, branch_name, balance) VALUES (11,'SBI_Jantarmantar',8000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (10,'SBI_ResidencyRoad',5000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (8,'SBI_ResidencyRoad',4000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (9,'SBI_ParliamentRoad',3000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (6,'SBI_ShivajiRoad',4000)
```

```
INSERT INTO bank_account (accno, branch_name, balance) VALUES (4,' SBI_ParliamentRoad',5000);
INSERT INTO bank_account (accno, branch_name, balance) VALUES (1,'SBI_Chamrajpet',2000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (2,'SBI_ResidencyRoad',3000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (3,'SBI_ShivajiRoad',4000)
INSERT INTO bank_account (accno, branch_name, balance) VALUES (4,'ParlimentRoad',5000);
```

```
SELECT * FROM `bank_account`
```

←T→ accno branch_name balance

Edit	Copy	Delete	1	SBI_Chamrajpet	2000
Edit	Copy	Delete	2	SBI_ResidencyRoad	3000
Edit	Copy	Delete	3	SBI_ShivajiRoad	4000
Edit	Copy	Delete	4	SBI_ParliamentRoad	5000
Edit	Copy	Delete	5	SBI_Jantarmantar	8000
Edit	Copy	Delete	6	SBI_ShivajiRoad	4000
Edit	Copy	Delete	8	SBI_ResidencyRoad	4000
Edit	Copy	Delete	9	SBI_ParliamentRoad	3000
Edit	Copy	Delete	10	SBI_ResidencyRoad	5000
Edit	Copy	Delete	11	SBI_Jantarmantar	8000

```
insert into bank_customer (customer_name,customer_street,city) VALUES ('Avinash','a','Bangalore')
insert into bank_customer (customer_name,customer_street,city) VALUES ('Dinesh','b','Bangalore')
insert into bank_customer (customer_name,customer_street,city) VALUES ('Mohan','c','Bangalore')
insert into bank_customer (customer_name,customer_street,city) VALUES ('Nikil','d','Delhi')
insert into bank_customer (customer_name,customer_street,city) VALUES ('Ravi','e','Delhi')
select * from bank_customer
```

←T→

			customer_name	customer_street	city
Edit	Copy	Delete	Avinash	a	Bangalore
Edit	Copy	Delete	Dinesh	b	Bangalore
Edit	Copy	Delete	Mohan	c	Bangalore
Edit	Copy	Delete	Nikil	d	Delhi
Edit	Copy	Delete	Ravi	e	Delhi

```
insert into depositor (customer_name,accno) VALUES ('Avinash',1)
```

```
insert into depositor (customer_name,accno) VALUES ('Avinash',8)
insert into depositor (customer_name,accno) VALUES ('Dinesh',2)
insert into depositor (customer_name,accno) VALUES ('Dinesh',10)
insert into depositor (customer_name,accno) VALUES ('Nikil',4)
insert into depositor (customer_name,accno) VALUES ('Nikil',9)
insert into depositor (customer_name,accno) VALUES ('Nikil',11)
insert into depositor (customer_name,accno) VALUES ('Ravi',5)
```

```
select * from depositor
```

customer_name	accno
Avinash	1
Dinesh	2
Nikil	4
Ravi	5
Avinash	8
Nikil	9
Dinesh	10
Nikil	11

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

```
use bank;
select customer_name from depositor
where acc_no in (select acc_no from bank_account where branch_name =
'SBI_RESIDENCYROAD')
group by customer_name HAVING count(customer_name) >= 2;
```

DINESH

```
select customer_name from depositor
where acc_no in (select acc_no from bank_account where branch_name =
'SBI_PARLIAMENTROAD')
group by customer_name HAVING count(customer_name) >= 2
```

NIKIL

Or

```
select DISTINCT c.customer_name from depositor c where 2 <= (select
count(b.branch_name) from depositor d join bank_account b on d.acc_no =
b.acc_no where d.customer_name = c.customer_name and b.branch_name =
'SBI_ParliamentRoad')
```

NIKIL

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

```
use bank

SELECT distinct S.customer_name FROM depositor as S
WHERE not exists (
       (SELECT branch_name
        FROM branch
        WHERE branch_city = 'DELHI')
        EXCEPT
        (SELECT R.branch_name
         FROM depositor as T, bank_account as R
         WHERE T.acc_no = R.acc_no and
                 S.customer_name = T.customer_name ))
```

NIKIL

```
alter table bank_account
add foreign key(branch_name) references branch(branch_name) on delete
CASCADE;
```

MAHANTESH GATTINA ~ 1BM19CS219

Program3: SUPPLIER DATABASE MANAGAMENT

```
use supplier;
create table supplier(
      sid int not null,
      sname varchar(20) not null,
      address varchar(30) not null,
      primary key(sid)
      );


      create table parts(
            pid int not null,
            pname varchar(20),
            color varchar(10),
            primary key(pid)
           );

      create table catalog(
            sid int not null,
            pid int not null,
            cost real not null,
            foreign key(sid) references supplier(sid) on delete cascade,
            foreign key(pid) REFERENCES parts(pid) on delete CASCADE
            );


      insert into supplier
      (sid,sname,address)
      VALUES
      (1,'AWS','DELHI'),
      (2,'BWS','SURAT');
```

←T→			sid	sname	address
Edit	Copy	Delete	1	AWS	DELHI
Edit	Copy	Delete	2	BWS	SURAT

```
      insert into parts
      (pid, pname, color)
      VALUES
      (1,'A','RED'),
      (2,'B', 'RED'),
      (3,'C','BLUE'),
      (4,'D','GREEN');

      SELECT * from parts
```

+ Options

←T→			pid	pname	color
Edit	Copy	Delete	1	A	RED

Edit	Copy	Delete	2	B	RED
Edit	Copy	Delete	3	C	BLUE
Edit	Copy	Delete	4	D	GREEN

```
insert into catalog
(sid,pid,cost)
VALUES
      (1,1,100),
      (1,2,200),
      (2,2,300),
      (2,3,400),
      (2,4,250);

select * from catalog;
```

sid	pid	cost
1	1	100
1	2	200
2	2	300
2	3	400
2	4	250

1. 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;
```

←T→			pname
Edit	Copy	Delete	A
Edit	Copy	Delete	B
Edit	Copy	Delete	C
Edit	Copy	Delete	D

Or
select p.pname from parts p where exists(select * from catalog where pid = p.pid)

pname
A
B
C
D

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

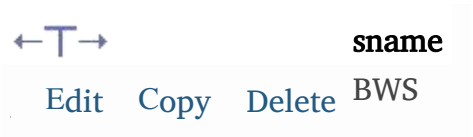
insert into catalog
(sid, pid, cost)
VALUES
(2,1,150);

Select * from catalog;

sid	pid	cost
1	1	100
1	2	200
2	2	300
2	3	400
2	4	250
2	1	150

SELECT s.sname
FROM supplier s
WHERE NOT EXISTS ((SELECT p.pid

```
FROM parts p )
EXCEPT
( SELECT c.pid
FROM catalog c
WHERE c.sid = s.sid ));
```



←T→	sname
Edit Copy Delete	BWS

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

```
SELECT S.sname
FROM supplier S
WHERE NOT EXISTS (( SELECT P.pid
FROM parts P
WHERE P.color = 'RED')
EXCEPT
( SELECT C.pid
FROM catalog C, parts P
WHERE C.sid = S.sid AND
C.  pid = P.pid AND P.color = 'RED' ))
```

←T→	sname
Edit Copy Delete	AWS
Edit Copy Delete	BWS

Or

```
select s1.sname from supplier s1 where not exists((select pid from parts where
color = 'red') except (select pid from catalog where sid = s1.sid and pid in ((select
pid from parts where color = 'red'))))
```

```
sname
AWS
BWS
```

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

```
SELECT P.pname
FROM parts P, catalog C, supplier S
WHERE P.pid = C.pid AND C.sid = S.sid
AND S.sname = 'AWS'
AND NOT EXISTS ( SELECT *
```

```
FROM catalog C1, supplier S1
WHERE P.pid = C1.pid AND C1.sid = S1.sid AND
S1.sname <> 'AWS' )
```

pname

```
select pname from parts where pid in (select pid from catalog where sid = (select sid from supplier where sname = 'aws')) except (select pid from catalog where sid = (select sid from supplier where sname <> 'aws'))
```

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).

```
SELECT DISTINCT C.sid
FROM catalog C
WHERE C.cost > ( SELECT AVG (C1.cost)
FROM catalog C1
WHERE C1.pid = C.pid group by C1,pid)
```

sid
2

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

```
SELECT P.pid, S.sname
FROM parts P, supplier S, catalog C
WHERE C.pid = P.pid
AND C.sid = S.sid
AND C.cost = (SELECT max(c1.cost) from catalog c1 where c1.pid = P.pid)
```

pid	sname
2	BWS
3	BWS
4	BWS
1	BWS

Or

```
select p.pname,s.sname from catalog c,parts p, supplier s where c.cost =
```

```
(select max(cost) from catalog where pid = c.pid) and c.pid = p.pid and c.sid = s.sid
order by p.pname
```

pname	sname
A	BWS
B	BWS
C	BWS
D	BWS

7.Find the sids of suppliers who supply only red parts.

```
select c.SID
from catalog c JOIN
      parts p
      ON c.PID = p.PID
group by c.SID
having min(p.color) = max(p.color) and
        min(p.color) = 'RED'
```

SID
1

Or

```
select DISTINCT c.sid from catalog c where
not exists ((
(select pname from parts where pid in (select pid from catalog where sid = c.sid))
except
     select pname from parts where color = 'red'))
```

1

Name: Mahantesh Gattina
USN: 1BM19CS219

Program4: student faculty database

```
use `student faculty database`;
create table student(
snum int not null,
sname varchar(20) not null,
major varchar(20) not null,
level char(2) not null,
primary key(snum)
     );

     create table faculty(
          fid int not null,
          fname varchar(20) not null,
          depid int not null,
          primary key(fid)
```

```
);

create table class(
    name varchar(20) not null,
    meets_at time not null,
    room varchar(20) not null,
    fid int not null,
    primary key(name),
    foreign key(fid) references faculty(fid) on delete cascade on update
CASCADE
    );

insert into student
(snum,sname,major,level,age)
VALUES
(1, 'A', 'MATH','FR',18),
(2,'B','MATH','FR',18),
(3,'C','TFCS','SO',19),
(4,'D','TFCS','SO',19),
(5,'E','DBMS','JR',20),
(6,'F','DBMS','JR',21),
(7,'G','ADA','SR',21);

Select * from student;
```

+ Options

←T→			snum	sname	major	level	age
Edit	Copy	Delete	1	A	MATH	FR	18
Edit	Copy	Delete	2	B	MATH	FR	18
Edit	Copy	Delete	3	C	TFCS	SO	19
Edit	Copy	Delete	4	D	TFCS	SO	19
Edit	Copy	Delete	5	E	DBMS	JR	20
Edit	Copy	Delete	6	F	DBMS	JR	21
Edit	Copy	Delete	7	G	ADA	SR	21

```
insert into faculty
(fid,fname,depid)
VALUES
(1,'RAM',1),
(2,'SHYAM',2),
(3,'TOM',3),
(4,'DOM',4);


use `student faculty database`


SELECT * FROM faculty
```

fid	fname	depid
1	RAM	1
2	SHYAM	2
3	TOM	3
4	DOM	4

```
INSERT INTO class
(name, meets_at,room,fid)
VALUES
('A','1:2:0','R124',1);

INSERT INTO class
(name, meets_at,room,fid)
VALUES
('B','2:2:0','R125',2),
('C','3:2:0','R126',3),
('D','3:2:0','R127',4),
('E','4:2:0','R128',4);


use `student faculty database`


SELECT * FROM class
```

name	meets_at	room	fid
A	01:02:00	R124	1
B	02:02:00	R125	2
C	03:02:00	R126	3
D	03:02:00	R127	4
E	04:02:00	R128	4

```
insert into enrolled
(snum, cname)
VALUES
(1,'A'),
(2,'B'),
(3,'C'),
(4,'D'),
(5,'E'),
(6,'A'),
(7,'B');

SELECT * from enrolled
```

snum	cname
1	A
2	B
3	C
4	D
5	E
6	A
7	B

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

use `student faculty database`

1st approach:

SELECT sname from student where level = 'JR' AND snum in (select snum from enrolled where cname in (select name from class where fid = (select fid from faculty where fname = 'RAM')))

sname
F

2nd approach:

select sname from student s, enrolled e, class c, faculty f where s.snum = e.snum and e.cname = c.name and c.fid = f.fid and s.level = 'JR' and f.fname = 'RAM';

sname
F

3rd approach:

select s.sname from student s where exists (select e.cname from enrolled e where e.snum = s.snum and e.cname in (select c.name from class c, faculty f where c.fid = f.fid and f.fname = 'RAM')) and s.level = 'JR'

4TH APPROACH:

select DISTINCT s.sname from student s join enrolled e on s.snum = e.snum join class c on c.name = e.cname join faculty f on c.fid = f.fid where f.fname = 'RAM' AND s.level = 'JR';

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

use `student faculty database`;
insert into enrolled
VALUES
(2,'A'),
(3,'A'),
(4,'A'),
(5,'A'),
(6,'A')

Select * from enrolled;

snum cname

1	A
2	B
3	C
4	D
5	E
6	A
7	B
2	A
3	A
4	A
5	A
6	A

```
select distinct c.name from class c where c.room = 'R128' or exists (select * from enrolled where cname = c.name group by cname having count(*) >= 5)
```

```
name
A
E
I
```

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

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

```
use `student faculty database`
```

```
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.name and e2.cname = c2.name and c1.meets_at = c2.meets_at)
```

sname
C

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

insert into class
VALUES
('F', '04:02:00','R124',3),
('G', '05:02:00','R125',3),
('H', '06:02:00','R127',3),
('I', '07:02:00','R128',3)

```
select f.fname from faculty f where not exists((select room from class c) except (select c1.room from class c1 where c1.fid = f.fid))
```

fname
TOM

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

```
use `student faculty database`
```

```
select f.fname from faculty f where (SELECT count(e.snum) from class c, enrolled e where c.name = e.cname and c.fid = f.fid) < 5
```

fname
SHYAM
TOM
DOM

```
select f.fname from class c join enrolled e on c.name = e.cname join faculty f on f.fid = c.fid group by c.fid having count(*) < 5
```

fname
SHYAM
TOM
DOM

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

```
insert into student
VALUES
(8,'H','ADA','SR',21);

use `student faculty database`

select sname from student where snum not in (select snum from enrolled)
```

H

Or

```
select s.sname from student s where not exists(select * from enrolled where snum
= s.snum)
```

H

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).

```
use `student faculty database`

select s.age, s.level from student s group by s.age,s.level having s.level in
(select s1.level from student s1 where s1.age = s.age group by s1.level,s1.age
having count(*) >= all (select count(*) from student s2 where s1.age = s2.age
group by s2.level,s2.age))
```

```
age level
18   FR
19   SO
20   JR
21   SR
```

Or

```
select distinct s.age, s.level from   student s where s.level = (select level from
student where age = s.age group by level order by count(level) desc limit 0,1)
```

```
age level
18   FR
19   SO
20   JR
21   SR
```

Mahantesh Gattina
1BM19CS219

Program5: Airline database

```
use airline;
create table flights(
      flno integer not null,
      ffrom varchar(20) not null,
      fto 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 employee(
      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,
      foreign key(eid) REFERENCES employee(eid) on delete cascade on update
cascade,
      foreign key(aid) references aircraft(aid) on delete cascade on update cascade);

       INSERT INTO flights
       VALUES
                    (1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000),
                    (2,'Bangalore','Delhi',5000,'12:15:00','04:30:00',25000),
                    (3,'Bangalore','Mumbai',3500,'02:15:00','05:25:00',30000),
                    (4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000),
                    (5,'Delhi','Frankfurt',18000,'07:15:00','05:30:00',90000),
                    (6,'Bangalore','Frankfurt',19500,'10:00:00','07:45:00',95000),
                    (7,'Bangalore','Frankfurt',17000,'12:00:00','06:30:00',99000);

      INSERT INTO aircraft (aid,aname,cruisingrange) values
                (123,'Airbus',1000),
                (302,'Boeing',5000),
                (306,'Jet01',5000),
                (378,'Airbus380',8000),
                (456,'Aircraft',500),
                (789,'Aircraft02',800),
                (951,'Aircraft03',1000);

      INSERT INTO employee (eid,ename,salary) VALUES
                (1,'Ajay',30000),
```

```
        (2,'Ajith',85000),
        (3,'Arnab',50000),
        (4,'Harry',45000),
        (5,'Ron',90000),
        (6,'Josh',75000),
        (7,'Ram',100000);

INSERT INTO certified (eid,aid) VALUES
        (1,123),
        (2,123),
        (1,302),
        (5,302),
        (7,302),
        (1,306),
        (2,306),
        (1,378),
        (2,378),
        (4,378),
        (6,456),
        (3,456),
        (5,789),
        (6,789),
        (3,951),
        (1,951),
        (1,789);

use airline


select * from flights


flno ffrom     fto  distance      departs arrives   price
1    Bangalore    Mangalore   360 10:45:00     12:00:00     10000
2    Bangalore    Delhi     5000     12:15:00     04:30:00     25000
3    Bangalore    Mumbai 3500     02:15:00     05:25:00     30000
4    Delhi     Mumbai 4500     10:15:00     12:05:00     35000
5    Delhi     Frankfurt     18000   07:15:00     05:30:00     90000
6    Bangalore    Frankfurt     19500   10:00:00     07:45:00     95000
7    Bangalore    Frankfurt     17000   12:00:00     06:30:00     99000




use airline


select * from aircraft


aid  aname   cruisingrange
123 Airbus   1000
302 Boeing   5000
306 Jet01    5000
378 Airbus380    8000
456 Aircraft  500
789 Aircraft02    800
951 Aircraft03    1000
```

use airline

select * from employee

eid	ename	salary
1	Ajay	30000
2	Ajith	85000
3	Arnab	50000
4	Harry	45000
5	Ron	90000
6	Josh	75000
7	Ram	100000

use airline

select * from certified

eid	aid
1	123
2	123
1	302
5	302
7	302
1	306
2	306
1	378
2	378
4	378
6	456
3	456
5	789
6	789
3	951
1	951
1	789

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

Approach1:

use airline

select distinct aname from aircraft where aid in (select aid from certified where eid in (select eid from employee where salary > 80000))

aname
Airbus
Boeing
Jet01

Airbus380
Aircraft02
Approach2:

select DISTINCT aname from aircraft where aid in (select c.aid from certified c, employee e where e.eid = c.eid and e.salary > 80000)

aname
Airbus
Boeing
Jet01
Airbus380
Aircraft02

Approach 3:

select a.aname from aircraft a where exists (select * from certified c, employee e where c.aid = a.aid and c.eid = e.eid and e.salary > 80000)

aname
Airbus
Boeing
Jet01
Airbus380
Aircraft02

Approach 4:

use airline
select distinct a.aname from aircraft a, certified c, employee e where a.aid = c.aid and c.eid = e.eid and exists (select * from employee e1 where e1.eid = e.eid and e1.salary > 80000)

aname
Airbus
Boeing
Jet01
Airbus380
Aircraft02

Approach 5:

use airline

select distinct a.aname from aircraft a, certified c, employee e where a.aid = c.aid and c.eid = e.eid and not exists (select * from employee e1 where e1.eid = e.eid and e1.salary < 80000)

aname
Airbus
Boeing
Jet01
Airbus380
Aircraft02

Approach 6:

select distinct a.aname from employee e join certified c on e.eid = c.eid join aircraft a on c.aid = a.aid where e.salary > 80000

aname
Airbus
Boeing
Jet01
Airbus380
Aircraft02

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.

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

1 8000

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

Approach1:

select e.ename from employee e where exists (select * from certified c where c.eid = e.eid) and e.salary < (select min(price) from flights where ffrom = 'Bangalore' and fto = 'Frankfurt')

ename
Ajay
Ajith
Arnab
Harry
Ron
Josh

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

Approach1:

select a.aname, avg(e.salary) from aircraft a, certified c, employee e where c.aid = a.aid and c.eid = e.eid and a.cruisingrange > 1000 group by a.aname

aname avg(e.salary)
Airbus380 53333.3333

Boeing 73333.3333
Jet01 57500.0000

Approach2:

select a.aid, a.aname, avg(e.salary) from aircraft a, certified c, employee e where c.aid = a.aid and c.eid = e.eid and a.cruisingrange > 1000 group by a.aid

aid aname avg(e.salary)
302 Boeing 73333.3333
306 Jet01 57500.0000
378 Airbus380 53333.3333

Approach3:

select a.aname,avg(e.salary) from aircraft a,employee e, certified c where a.aname in (select aname from aircraft where cruisingrange > 1000)
and e.eid = c.eid and c.aid = a.aid group by a.aname

aname avg(e.salary)
Airbus380 53333.3333
Boeing 73333.3333
Jet01 57500.0000

Approach4:

select a.aname, avg(e.salary) from employee e join certified c on e.eid = c.eid join aircraft a on a.aid = c.aid where a.cruisingrange > 1000 group by a.aid

aname avg(e.salary)
Boeing 73333.3333
Jet01 57500.0000
Airbus380 53333.3333

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

Approach1:

select e.ename from employee e, certified c, aircraft a where a.aname like '%Boeing%' and a.aid = c.aid and c.eid = e.eid

ename
Ajay
Ron
Ram

Approach2:

select e.ename from employee e where e.eid in(select c.eid from certified c where c.aid in (select aid from aircraft where aname = 'Boeing'))

ename
Ajay
Ron
Ram

Approach3:

select e.ename from employee e where exists(select * from certified c where c.eid = e.eid and exists(select * from aircraft a where aname = 'Boeing' and a.aid = c.aid))

ename
Ajay
Ron
Ram

Approach4:

select e.ename from employee e join certified c on e.eid = c.eid join aircraft a on c.aid = a.aid where a.aname like '%boeing%'

ename
Ajay
Ron
Ram

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

localhost/airline/aircraft/
http://localhost/phpmyadmin/index.php?route=/database/sql&db=airline

Showing rows 0 - 0 (1 total, Query took 0.0019 seconds.)

select aid from aircraft where cruisingrange > (select distance from flights where ffrom = 'Bangalore' and fto = 'Delhi')

378

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

```
insert into employee
VALUES
(10,'VIRAJ',100000),
```

(11,'APPU',150000);

select e1.ename, e1.salary from employee e1 where e1.salary > (select avg(e.salary) from employee e where e.eid in (select eid from certified)) and not exists(select * from certified c where c.eid = e1.eid)

ename	salary
VIRAJ	100000
APPU	150000

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

select f.departs from flights f where f.flno in ((select f0.flno from flights f0 where f0.ffrom = 'Bangalore' and f0.fto = 'Ballari' and f0.arrives < '18:00:00') UNION (SELECT f0.flno from flights f0, flights f1 where f0.ffrom = 'Bangalore' and f0.fto <> 'Ballari' and f1.ffrom = f0.fto and f0.arrives < f1.departs and f1.fto = 'Ballari' and f1.arrives < '18:00:00') union (select f0.flno from flights f0, flights f1, flights f2 where f0.ffrom = 'Bangalore' and f0.fto <> 'Ballari' and f1.ffrom = f0.fto and f0.arrives < f1.departs and f1.fto <> 'Ballari' AND f2.ffrom = f1.fto and f2.fto = 'Ballari' and f1.arrives < f2.departs and f2.arrives < '18:00:00'));

departs
10:45:00
15:45:00