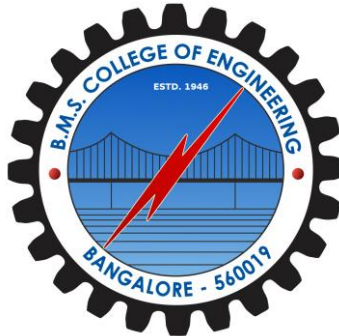


B.M.S. COLLEGE OF ENGINEERING

(AUTONOMOUS COLLEGE UNDER VTU)
BENGALURU-19



LAB TEST 2 RECORD(1-10)

NAME : TASMIYA FATHIMA
USN : 1BM19CS172
COURSE NAME : DATABASE MANAGEMENT SYSTEMS
COURSE TITLE : 19CS4PCDBM
SEM : 4
SECTION :D

PROGRAM 1- INSURANCE DATABASE

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

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

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

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

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

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

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

```
create database Supplier;
```

```
use Supplier;
```

```
create table person (driver_id varchar(10),
```

```
name varchar(20),
```

```
address varchar(30),
```

```
primary key(driver_id));
```

```
desc person;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|-------------|------|-----|---------|-------|
| ► | driver_id | varchar(10) | NO | PRI | NULL | |
| | name | varchar(20) | YES | | NULL | |
| | address | varchar(30) | YES | | NULL | |

```
create table car(reg_num varchar(10),model
```

```
varchar(10),year int,primary key(reg_num));
```

```
desc car;
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|-------------|------|-----|---------|-------|
| ► | reg_num | varchar(10) | NO | PRI | NULL | |
| | model | varchar(10) | YES | | NULL | |
| | year | int | YES | | NULL | |

```
create table accident(report_num int,accident_date
```

```
date,location varchar(20),primary key(report_num));
```

desc accident;

| | Field | Type | Null | Key | Default | Extra |
|---|---------------|-------------|------|-----|---------|-------|
| ► | report_num | int | NO | PRI | NULL | |
| | accident_date | date | YES | | NULL | |
| | location | varchar(20) | YES | | NULL | |

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;

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|-------------|------|-----|---------|-------|
| ► | driver_id | varchar(10) | NO | PRI | NULL | |
| | reg_num | varchar(10) | NO | PRI | NULL | |

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;

| | Field | Type | Null | Key | Default | Extra |
|---|---------------|-------------|------|-----|---------|-------|
| ► | driver_id | varchar(10) | NO | PRI | NULL | |
| | reg_num | varchar(10) | NO | PRI | NULL | |
| | report_num | int | NO | PRI | NULL | |
| | damage_amount | int | YES | | NULL | |

ii. Enter at least five tuples for each relation

insert into person values('A01','Richard','Srinivas Nagar');

insert into person values('A02','Pradeep','Rajajinagar');

```

insert into person values('A03','Smith','Ashoknagar');
insert into person values('A04','Venu','N.R.Colony');
insert into person values('A05','John','Hanumanth Nagar');
select * from person;

```

| | driver_id | name | address |
|---|-----------|---------|-----------------|
| ▶ | A01 | Richard | Srinivas Nagar |
| | A02 | Pradeep | Rajajinagar |
| | A03 | Smith | Ashoknagar |
| | A04 | Venu | N.R.Colony |
| | A05 | John | Hanumanth Nagar |
| * | NULL | NULL | NULL |

```

insert into car values('KA052250','Indica', 1990);
insert into car values('KA031181','Lancer', 1957);
insert into car values('KA095477','Toyota',1998);
insert into car values('KA053408','Honda',2008);
insert into car values('KA041702','Audi',2005);
select * from car;

```

| | reg_num | model | year |
|---|----------|--------|------|
| ▶ | KA031181 | Lancer | 1957 |
| | KA041702 | Audi | 2005 |
| | KA052250 | Indica | 1990 |
| | KA053408 | Honda | 2008 |
| | KA095477 | Toyota | 1998 |
| * | NULL | NULL | NULL |

```

insert into accident values(11,'2003-01-01','Mysore Road');
insert into accident values(12,'2004-02-02','Southend Circle');
insert into accident values(13,'2003-01-21','Bulltemple Road');
insert into accident values(14,'2008-02-17','Mysore Road');
insert into accident values(15,'2005-03-04','Kanakpura Road');
select * from accident;

```

| | report_num | accident_date | location |
|---|------------|---------------|-----------------|
| ▶ | 11 | 2003-01-01 | Mysore Road |
| | 12 | 2004-02-02 | Southend Circle |
| | 13 | 2003-01-21 | Bulltemple Road |
| | 14 | 2008-02-17 | Mysore Road |
| | 15 | 2005-03-04 | Kanakpura Road |
| ✱ | NULL | NULL | NULL |

insert into owns values('A01','KA052250');

insert into owns values('A02','KA053408');

insert into owns values('A03','KA095477');

insert into owns values('A04','KA031181');

insert into owns values('A05','KA041702');

select * from owns;

| | driver_id | reg_num |
|---|-----------|----------|
| ▶ | A04 | KA031181 |
| | A05 | KA041702 |
| | A01 | KA052250 |
| | A02 | KA053408 |
| | A03 | KA095477 |
| ✱ | NULL | NULL |

insert into participated values('A01','KA052250',11,10000);

insert into participated values('A02','KA053408',12,50000);

insert into participated values('A03','KA095477',13,25000);

insert into participated values('A04','KA031181',14,3000);

insert into participated values('A05','KA041702',15,5000);

select * from participated;

| | driver_id | reg_num | report_num | damage_amount |
|---|-----------|----------|------------|---------------|
| ▶ | A01 | KA052250 | 11 | 10000 |
| | A02 | KA053408 | 12 | 50000 |
| | A03 | KA095477 | 13 | 25000 |
| | A04 | KA031181 | 14 | 3000 |
| | A05 | KA041702 | 15 | 5000 |
| ✱ | NULL | NULL | NULL | NULL |

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.

update participated set damage_amount=25000 where 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 |

b. Add a new accident to the database.

insert into person values('A06','Josphe','Shanti Nagar');

insert into car values('KA012370','Honda', 2008);

insert into accident values(16,'2008-01-01','MG Road');

insert into owns values('A06','KA012370');

insert into participated values('A06','KA012370',16,15000);

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

select count(*) from accident where year(accident_date)=2008;

| | count(*) |
|---|----------|
| ▶ | 2 |

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

select count(*) as no_of_acc from participated where reg_num in(select reg_num from car where model='Lancer');

| | no_of_acc |
|---|-----------|
| ▶ | 2 |

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.

```
create database Banking;
use Banking;
```

```
create table Branch(branch_name varchar(30),branch_city varchar(30),assets real, primary
key(branch_name));
desc Branch;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ► | branch_name | varchar(30) | NO | PRI | NULL | |
| | branch_city | varchar(30) | YES | | NULL | |
| | assets | double | YES | | NULL | |

```
create      table      BankCustomer(customer_name      varchar(30),customer_street
varchar(30),customer_city varchar(30), primary key(customer_name));
desc BankCustomer;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-----------------|-------------|------|-----|---------|-------|
| ► | customer_name | varchar(30) | NO | PRI | NULL | |
| | customer_street | varchar(30) | YES | | NULL | |
| | customer_city | varchar(30) | YES | | NULL | |

```
create table BankAccount(
accno int,
branch_name varchar(20),
balance real,
primary key(accno),
foreign key(branch_name) references Branch(branch_name)
);
desc BankAccount;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ► | accno | int | NO | PRI | NULL | |
| | branch_name | varchar(20) | YES | MUL | NULL | |
| | balance | double | YES | | NULL | |

```
create table Depositer(
```

```

customer_name varchar(20),
accno int,
primary key(customer_name,accno),
foreign key(customer_name) references BankCustomer(customer_name),
foreign key(accno) references BankAccount(accno)
);
desc Depositer;

```

| | Field | Type | Null | Key | Default | Extra |
|---|---------------|-------------|------|-----|---------|-------|
| ► | customer_name | varchar(20) | NO | PRI | NULL | |
| | accno | int | NO | PRI | NULL | |

```

create table Loan(
loan_number int,
branch_name varchar(20),
Amount real,
primary key(loan_number),
foreign key(branch_name) references Branch(branch_name)
);
desc Loan;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ► | loan_number | int | NO | PRI | NULL | |
| | branch_name | varchar(20) | YES | MUL | NULL | |
| | Amount | double | YES | | NULL | |

ii. Enter at least five tuples for each relation.

```

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_ParlimentRoad','Delhi',10000);
insert into Branch values('SBI_Jantarmentar','Delhi',20000);
select *from Branch;

```

| | branch_name | branch_city | assests |
|---|-------------------|-------------|---------|
| ► | SBI_Chamrajpet | Bangalore | 50000 |
| | SBI_Jantarmentar | Delhi | 20000 |
| | SBI_ParlimentRoad | Delhi | 10000 |
| | SBI_ResidencyRoad | Bangalore | 10000 |
| | SBI_ShivajiRoad | Bombay | 20000 |
| * | NULL | NULL | NULL |


```

insert into Loan values(2,'SBI_ResidencyRoad',2000);
insert into Loan values(1,'SBI_Chamrajpet',1000);
insert into Loan values(3,'SBI_ShivajiRoad',3000);
insert into Loan values(4,'SBI_ParliamentRoad',4000);
insert into Loan values(5,'SBI_Jantarmanatar',3000);
select *from Loan;

```

| | loan_number | branch_name | Amount |
|---|-------------|--------------------|--------|
| ▶ | 1 | SBI_Chamrajpet | 10000 |
| | 2 | SBI_ResidencyRoad | 20000 |
| | 3 | SBI_ShivajiRoad | 30000 |
| | 4 | SBI_ParliamentRoad | 40000 |
| | 5 | SBI_Jantarmanatar | 30000 |
| • | NULL | NULL | NULL |

```

insert into BankAccount values(1,'SBI_Chamrajpet',2000);
insert into BankAccount values(2,'SBI_ResidencyRoad',5000);
insert into BankAccount values(3,'SBI_ShivajiRoad',6000);
insert into BankAccount values(4,'SBI_ParliamentRoad',9000);
insert into BankAccount values(5,'SBI_Jantarmanatar',8000);
insert into BankAccount values(6, 'SBI_ShivajiRoad', 4000);
insert into BankAccount values(8, 'SBI_ResidencyRoad', 4000);
insert into BankAccount values(9, 'SBI_ParliamentRoad', 3000);
insert into BankAccount values(10, 'SBI_ResidencyRoad', 5000);
insert into BankAccount values(11, 'SBI_Jantarmanatar', 2000);
select *from BankAccount;

```

| | accno | branch_name | balance |
|---|-------|--------------------|---------|
| ▶ | 1 | SBI_Chamrajpet | 2000 |
| | 2 | SBI_ResidencyRoad | 5000 |
| | 3 | SBI_ShivajiRoad | 6000 |
| | 4 | SBI_ParliamentRoad | 9000 |
| | 5 | SBI_Jantarmanatar | 8000 |
| | 6 | SBI_ShivajiRoad | 4000 |
| | 8 | SBI_ResidencyRoad | 4000 |
| | 9 | SBI_ParliamentRoad | 3000 |
| | 10 | SBI_ResidencyRoad | 5000 |
| | 11 | SBI_Jantarmanatar | 2000 |
| • | NULL | NULL | NULL |

```

insert into BankCustomer values ('Avinash', 'Bull_Temple_Road', 'Bangalore');
insert into BankCustomer values ('Dinesh', 'Bannerghatta_Road', 'Bangalore');
insert into BankCustomer values ('Mohan', 'National_College_Road', 'Bangalore');
insert into BankCustomer values ('Nikhil', 'Akbar_Road', 'Delhi');
insert into BankCustomer values ('Ravi', 'Prithviraj_Road', 'Delhi');
select *from BankCustomer;

```

| | customer_name | customer_street | customer_city |
|---|---------------|-----------------------|---------------|
| ▶ | Avinash | Bull_Temple_Road | Bangalore |
| | Dinesh | Bannergatta_Road | Bangalore |
| | Mohan | National_College_Road | Bangalore |
| | Nikhil | Akbar_Road | Delhi |
| | Ravi | Prithviraj_Road | Delhi |
| ★ | NULL | NULL | NULL |

```

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);
select *from Depositer;

```

| | customer_name | accno |
|---|---------------|-------|
| ▶ | Avinash | 1 |
| | Dinesh | 2 |
| | Nikhil | 4 |
| | Ravi | 5 |
| | Avinash | 8 |
| | Nikhil | 9 |
| | Dinesh | 10 |
| | Nikhil | 11 |
| ★ | NULL | NULL |

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

```

select c.customer_name
from BankCustomer c
where exists(
select d.customer_name
from Depositer d, BankAccount ba
where
d.accno=ba.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
);

```

| | customer_name |
|---|---------------|
| ▶ | Dinesh |
| • | NULL |

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

```
select distinct d.customer_name from Depositer d where exists( select * from BankAccount ba
where ba.accno=d.accno and exists (select * from Branch b where b.branch_name =
ba.branch_name and b.branch_city='Delhi'));
```

| | customer_name |
|---|---------------|
| ▶ | Ravi |
| | Nikhil |

v. Demonstrate how you delete all account tuples at every branch located in a specific city (Ex. 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_ParlimentRoad | 9000 |
| | 5 | SBI_Jantarmanatar | 8000 |
| | 8 | SBI_ResidencyRoad | 4000 |
| | 9 | SBI_ParlimentRoad | 3000 |
| | 10 | SBI_ResidencyRoad | 5000 |
| | 11 | SBI_Jantarmanatar | 2000 |
| • | NULL | NULL | NULL |

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.

```
create database Supplier;
```

```
use Supplier;
```

```
create table catalog(sid int,pid int,cost int);
```

```
desc catalog;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|------|------|-----|---------|-------|
| ► | sid | int | YES | | NULL | |
| | pid | int | YES | | NULL | |
| | cost | int | YES | | NULL | |

```
create table supplier(sid int,sname varchar(20),city varchar(15),primary key (sid));
```

```
desc supplier;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | sid | int | NO | PRI | NULL | |
| | sname | varchar(20) | YES | | NULL | |
| | city | varchar(15) | YES | | NULL | |

```
create table parts(pid int,pname varchar(15),color varchar(10),primary key (pid));
```

```
desc parts;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | pid | int | NO | PRI | NULL | |
| | pname | varchar(15) | YES | | NULL | |
| | color | varchar(10) | YES | | NULL | |

```
insert into supplier values(10001, 'Acme Widget','Bengaluru');
```

```
insert into supplier values(10002,'Johns','Kolkata');
```

```
insert into supplier values(10003, 'Vimal','Mumbai');
```

```
insert into supplier values(10004, 'Reliance','Delhi');
```

```
insert into supplier values(10005, 'Mahindra','Mumbai');
```

```
select * from supplier;
```

| | sid | sname | city |
|---|-------|-------------|-----------|
| ▶ | 10001 | Acme Widget | Bengaluru |
| | 10002 | Johns | Kolkata |
| | 10003 | Vimal | Mumbai |
| | 10004 | Reliance | Delhi |
| • | NULL | NULL | NULL |

```
insert into parts values(20001, 'Book','Red');
```

```
insert into parts values(20002, 'Pen','Red');
```

```
insert into parts values(20003, 'Pencil','Green');
```

```
insert into parts values(20004, 'Mobile','Green');
```

```
insert into parts values(20005, 'Charger','Black');
```

```
select * from parts;
```

| | pid | pname | color |
|---|-------|---------|-------|
| ▶ | 20001 | Book | Red |
| | 20002 | Pen | Red |
| | 20003 | Pencil | Green |
| | 20004 | Mobile | Green |
| | 20005 | Charger | Black |
| • | NULL | NULL | NULL |

```
insert into catalog values(10001, '20001','10');
```

```
insert into catalog values(10001, '20002','10');
```

```
insert into catalog values(10001, '20003','30');
```

```
insert into catalog values(10001, '20004','10');
```

```
insert into catalog values(10001, '20005','10');
```

```
insert into catalog values(10002, '20001','10');
```

```
insert into catalog values(10002, '20002','20');
```

```
insert into catalog values(10003, '20003','30');
```

```
insert into catalog values(10004, '20003','40');
```

select * from 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 |
| * | NULL | NULL | NULL |

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;

| | pname |
|---|---------|
| ▶ | Book |
| | Pen |
| | Pencil |
| | Mobile |
| | Charger |

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

select s.sname from supplier s

where not exists (select p.pid from parts p where not exists

(select c.sid from catalog c where c.sid = s.sid and c.pid = p.pid));

| | sname |
|---|-------------|
| ▶ | Acme Widget |

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' and

(not exists (select c.sid from catalog c where c.sid = s.sid and c.pid = p.pid)));

| | |
|---|-------------|
| | sname |
| ▶ | Acme Widget |
| | Johns |

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 = 'Acme Widget'

and not exists

(select * from catalog c1, supplier s1 where

p.pid = c1.pid and c1.sid = s1.sid and s1.sname <> 'Acme Widget');

| | |
|---|---------|
| | pname |
| ▶ | Mobile |
| | Charger |

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);

| | |
|---|-------|
| | sid |
| ▶ | 10002 |
| | 10004 |

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 |
|---|-------|-------------|
| ▶ | 20001 | Acme Widget |
| | 20004 | Acme Widget |
| | 20005 | Acme Widget |
| | 20001 | Johns |
| | 20002 | Johns |
| | 20003 | 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, meetsat: time, room: string, fid: integer)

ENROLLED(snum: integer, cname:string)

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

```
create database Students;
```

```
use Students;
```

```
create table student(snum int, sname varchar(10), major varchar(2), lvl varchar(2), age
int,primary key (snum));
```

```
desc student;
```


| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | snum | int | NO | PRI | NULL | |
| | sname | varchar(10) | YES | | NULL | |
| | major | varchar(2) | YES | | NULL | |
| | lvl | varchar(2) | YES | | NULL | |
| | age | int | YES | | NULL | |

create table faculty(fid int, fname varchar(20), deptid int,primary key(fid));

desc faculty;

| | Field | Type | Null | Key | Default | Extra |
|---|--------|-------------|------|-----|---------|-------|
| ► | fid | int | NO | PRI | NULL | |
| | fname | varchar(20) | YES | | NULL | |
| | deptid | int | YES | | NULL | |

create table class(cname varchar(20), meetsat timestamp, room varchar(10), fid int,primary key (cname),foreign key(fid) references faculty(fid));

desc class;

| | Field | Type | Null | Key | Default | Extra |
|---|---------|-------------|------|-----|---------|-------|
| ► | cname | varchar(20) | NO | PRI | NULL | |
| | meetsat | timestamp | YES | | NULL | |
| | room | varchar(10) | YES | | NULL | |
| | fid | int | YES | MUL | NULL | |

create table enrolled(snum int, cname varchar(20),primary key(snum,cname),

foreign key(snum) references student(snum),

foreign key(cname) references class(cname));

desc enrolled;

| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | snum | int | NO | PRI | NULL | |
| | cname | varchar(20) | NO | PRI | NULL | |

insert into student values(1, 'jhon', 'CS', 'Sr', 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);
select * from student;

```

| | snum | sname | major | lvl | 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 |
| ★ | NULL | NULL | NULL | NULL | NULL |

```

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);
select * from faculty;

```

| | fid | fname | deptid |
|---|------|--------|--------|
| ▶ | 11 | Harish | 1000 |
| | 12 | MV | 1000 |
| | 13 | Mira | 1001 |
| | 14 | Shiva | 1002 |
| | 15 | Nupur | 1000 |
| ★ | NULL | NULL | NULL |

```

insert into class values('class1', '12/11/15 10:15:16', 'R1', 14);
insert into class values('class10', '12/11/15 10:15:16', 'R128', 14);

```

```

insert into class values('class2', '12/11/15 10:15:20', 'R2', 12);
insert into class values('class3', '12/11/15 10:15:25', 'R3', 12);
insert into class values('class4', '12/11/15 20:15:20', 'R4', 14);
insert into class values('class5', '12/11/15 20:15:20', 'R3', 15);
insert into class values('class6', '12/11/15 13:20:20', 'R2', 14);
insert into class values('class7', '12/11/15 10:10:10', 'R3', 14);
select * from class;

```

| | cname | meetsat | room | fid |
|---|---------|---------------------|------|------|
| ▶ | class1 | 2012-11-15 10:15:16 | R1 | 14 |
| | class10 | 2012-11-15 10:15:16 | R128 | 14 |
| | class2 | 2012-11-15 10:15:20 | R2 | 12 |
| | class3 | 2012-11-15 10:15:25 | R3 | 11 |
| | class4 | 2012-11-15 20:15:20 | R4 | 14 |
| | class5 | 2012-11-15 20:15:20 | R3 | 15 |
| | class6 | 2012-11-15 13:20:20 | R2 | 14 |
| | class7 | 2012-11-15 10:10:10 | R3 | 14 |
| ✱ | NULL | NULL | NULL | NULL |

```

insert into enrolled values(1, 'class1');
insert into enrolled values(2, 'class1');
insert into enrolled values(3, 'class3');
insert into enrolled values(4, 'class3');
insert into enrolled values(5, 'class4');
insert into enrolled values(1, 'class5');
insert into enrolled values(2, 'class5');
insert into enrolled values(3, 'class5');
insert into enrolled values(4, 'class5');
insert into enrolled values(5, 'class5');
select * from enrolled;

```

| | snum | cname |
|---|------|--------|
| ▶ | 1 | class1 |
| | 2 | class1 |
| | 3 | class3 |
| | 4 | class3 |
| | 5 | class4 |
| | 1 | class5 |
| | 2 | class5 |
| | 3 | class5 |
| | 4 | class5 |
| | 5 | class5 |
| * | NULL | NULL |

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

SELECT DISTINCT S.sname

FROM student S, class C, enrolled E, faculty F

WHERE S.snum = E.snum AND E.cname = C.cname AND C.fid = F.fid AND

F.fname = 'Harish' AND S.lvl = 'Jr';

| | sname |
|---|-------|
| ▶ | Tom |

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 = 'R128'

OR C.cname IN (SELECT E.cname FROM enrolled E

GROUP BY E.cname HAVING COUNT(*) >= 5);

| | cname |
|---|---------|
| ▶ | class10 |
| | class5 |
| * | NULL |

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);

```

| | sname |
|---|-------|
| ▶ | Rahul |

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

```

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)
);

```

| | fname | fid |
|---|-------|------|
| ▶ | Shiva | 14 |
| * | HULL | HULL |

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

```

SELECT DISTINCT F.fname
FROM faculty F
WHERE 5 > (SELECT COUNT(E.snum)
FROM class C, enrolled E
WHERE C.cname = E.cname
AND C.fid = F.fid);

```

| | fname |
|---|--------|
| ▶ | Harish |
| | MV |
| | Mira |
| | Shiva |

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

```
SELECT DISTINCT S.sname
FROM student S
WHERE S.snum NOT IN (SELECT E.snum
FROM enrolled E );
```

| | sname |
|---|-------|
| ▶ | Rita |

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

```
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.lvl, S1.age
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
FROM Student S2
WHERE s1.age = S2.age
GROUP BY S2.lvl, S2.age));
```

| | age | lvl |
|---|-----|-----|
| ▶ | 19 | Sr |
| | 20 | Jr |
| | 21 | Sr |

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.

```
create database flight;
```

```
use flight;
```

```
create table flights(
```

```
    flno int,
```

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

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

```
    distance int,
```

```
    departs datetime,
```

```
    arrives datetime,
```

```
    price int,
```

```
    primary key (flno));
```

```
desc flights;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|-------------|------|-----|---------|-------|
| ► | fno | int | NO | PRI | NULL | |
| | fromplace | varchar(15) | YES | | NULL | |
| | toplace | varchar(15) | YES | | NULL | |
| | distance | int | YES | | NULL | |
| | departs | datetime | YES | | NULL | |
| | arrives | datetime | YES | | NULL | |
| | price | int | YES | | NULL | |

create table aircraft(

aid int,

aname varchar(15),

cruisingrange int,

primary key (aid));

desc aircraft;

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

create table employees (

eid int,

ename varchar(15),

salary int,

primary key (eid));

desc employees;

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

create table certified (


```

eid int,

aid int,

foreign key (eid) references employees(eid),

foreign key (aid) references aircraft(aid));

```

```

desc certified;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|------|------|-----|---------|-------|
| ► | eid | int | YES | MUL | HULL | |
| | aid | int | YES | MUL | HULL | |

```

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);

```

```

select * from flights;

```

| | fno | fromplace | toplace | distance | departs | arrives | price |
|---|------|-----------|-----------|----------|---------------------|---------------------|-------|
| ▶ | 101 | Bangalore | Delhi | 2500 | 2005-05-13 07:15:31 | 2005-05-13 18:15:31 | 5000 |
| | 102 | Bangalore | Lucknow | 3000 | 2013-05-05 07:15:31 | 2013-05-05 11:15:31 | 6000 |
| | 103 | Lucknow | Delhi | 500 | 2013-05-05 12:15:31 | 2013-05-05 17:15:31 | 3000 |
| | 104 | Bangalore | Frankfurt | 8500 | 2013-05-05 07:15:31 | 2013-05-05 23:15:31 | 75000 |
| | 105 | Kolkata | Delhi | 3400 | 2013-05-05 07:15:31 | 2013-05-05 09:15:31 | 7000 |
| | 106 | Bangalore | Kolkata | 1000 | 2013-05-05 01:15:30 | 2013-05-05 09:20:30 | 10000 |
| | 107 | Bangalore | Frankfurt | 8000 | 2013-05-05 07:15:31 | 2013-05-05 22:15:31 | 60000 |
| | 108 | Lucknow | Kolkata | 1000 | 2013-05-05 11:30:30 | 2013-05-05 15:20:30 | 10000 |
| • | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

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);

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 |
| | 108 | 707 | 760 |
| | 109 | 747 | 1000 |
| • | NULL | NULL | NULL |

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);
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 |
| * | NULL | NULL | NULL |

```

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);
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 |
| | 702 | 108 |
| | 701 | 109 |

i. 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 where a.aid in (

select c.aid from certified c, employees e where

c.eid = e.eid and not exists(

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

));

| | aname |
|---|------------|
| ▶ | 747 |
| | Dreamliner |
| | Dream |
| | 707 |

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.

select max(a.cruisingrange), c.eid from certified c, aircraft a

where c.aid = a.aid group by c.eid having count(c.aid)>3;

| | max(a.cruisingrange) | eid |
|---|----------------------|-----|
| ▶ | 3500 | 701 |
| | 120000 | 702 |

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

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

| | ename |
|---|-------|
| ▶ | A |
| | E |

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.

```
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;
```

| | avg(e.salary) | aid |
|---|---------------|-----|
| ▶ | 75000.0000 | 101 |
| | 113333.3333 | 104 |
| | 50000.0000 | 105 |
| | 50000.0000 | 106 |
| | 113333.3333 | 107 |

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

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

| | ename |
|---|-------|
| ▶ | A |

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

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

| | |
|---|------------|
| | aname |
| ▶ | 747 |
| | Dreamliner |
| | Boeing |
| | Dream |

vii. 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.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));

```

| | fno | departs |
|---|-----|---------------------|
| ▶ | 102 | 2013-05-05 07:15:31 |
| | 106 | 2013-05-05 01:15:30 |

Program 6 - Order Database

Consider the following schema for Order Database:

SALESMAN (*Salesman_id, Name, City, Commission*)

CUSTOMER (*Customer_id, Cust_Name, City, Grade, Salesman_id*)

ORDERS (*Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id*)

```

create database Order;
use Order;

```

```

create table salesman (
    salesman_id int,
    name varchar (20),
    city varchar (20),
    commission varchar (20),
    primary key (salesman_id));
desc salesman;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ▶ | salesman_id | int | NO | PRI | NULL | |
| | name | varchar(20) | YES | | NULL | |
| | city | varchar(20) | YES | | NULL | |
| | commission | varchar(20) | YES | | NULL | |

```

create table customer (
    customer_id int,

```

```

    cust_name varchar (20),
    city varchar (20),
    grade int ,
    salesman_id int,
    primary key (customer_id),
    foreign key (salesman_id) references salesman(salesman_id) on delete set
null);
desc customer;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ► | customer_id | int | NO | PRI | NULL | |
| | cust_name | varchar(20) | YES | | NULL | |
| | city | varchar(20) | YES | | NULL | |
| | grade | int | YES | | NULL | |
| | salesman_id | int | YES | MUL | NULL | |

```

create table orders (
    ord_no int ,
    purchase_amt int,
    ord_date date,
    customer_id int,
    salesman_id int,
    primary key (ord_no),
    foreign key (customer_id) references customer (customer_id) on delete
cascade,
    foreign key (salesman_id) references salesman (salesman_id) on delete
cascade);
desc orders;

```

| | Field | Type | Null | Key | Default | Extra |
|---|--------------|------|------|-----|---------|-------|
| ► | ord_no | int | NO | PRI | NULL | |
| | purchase_amt | int | YES | | NULL | |
| | ord_date | date | YES | | NULL | |
| | customer_id | int | YES | MUL | NULL | |
| | salesman_id | int | YES | MUL | NULL | |

```

insert into salesman values (1000, 'john','bangalore','25 %');
insert into salesman values (2000, 'ravi','bangalore','20 %');
insert into salesman values (3000, 'kumar','mysore','15 %');
insert into salesman values (4000, 'smith','delhi','30 %');
insert into salesman values (5000, 'harsha','hydrabad','15 %');
select * from salesman;

```


| | salesman_id | name | city | commission |
|---|-------------|--------|-----------|------------|
| ▶ | 1000 | john | bangalore | 25 % |
| | 2000 | ravi | bangalore | 20 % |
| | 3000 | kumar | mysore | 15 % |
| | 4000 | smith | delhi | 30 % |
| | 5000 | harsha | hydrabad | 15 % |
| • | NULL | NULL | NULL | NULL |

```

insert into customer values (10, 'preethi','bangalore', 100, 1000);
insert into customer values (11,'vivek','mangalore', 300, 1000);
insert into customer values (12, 'bhaskar','chennai', 400, 2000);
insert into customer values (13, 'chethan','bangalore', 200, 2000);
insert into customer values (14, 'mamatha','bangalore', 400, 3000);
select * from customer;

```

| | customer_id | cust_name | city | grade | salesman_id |
|---|-------------|-----------|-----------|-------|-------------|
| ▶ | 10 | preethi | bangalore | 100 | 1000 |
| | 11 | vivek | mangalore | 300 | 1000 |
| | 12 | bhaskar | chennai | 400 | 2000 |
| | 13 | chethan | bangalore | 200 | 2000 |
| | 14 | mamatha | bangalore | 400 | 3000 |
| • | NULL | NULL | NULL | NULL | NULL |

```

insert into orders values (50, 5000, '04-06-17', 10, 1000);
insert into orders values (51, 450, '20-01-17', 10, 2000);
insert into orders values (52, 1000, '24-02-17', 13, 2000);
insert into orders values (53, 3500, '13-04-17', 14, 3000);
insert into orders values (54, 550, '09-03-17', 12, 2000);
select * from orders;

```

| | ord_no | purchase_amt | ord_date | customer_id | salesman_id |
|---|--------|--------------|------------|-------------|-------------|
| ▶ | 50 | 5000 | 2004-06-17 | 10 | 1000 |
| | 51 | 450 | 2020-01-17 | 10 | 2000 |
| | 52 | 1000 | 2024-02-17 | 13 | 2000 |
| | 53 | 3500 | 2013-04-17 | 14 | 3000 |
| | 54 | 550 | 2009-03-17 | 12 | 2000 |
| • | NULL | NULL | NULL | NULL | NULL |

1. Count the customers with grades above Bangalore's average.

```
SELECT grade, count(DISTINCT customer_id)
```

```

FROM customer
GROUP BY grade
HAVING grade > (SELECT AVG(grade)
FROM customer
WHERE city='bangalore');

```

| | grade | count(DISTINCT customer_id) |
|---|-------|-----------------------------|
| ▶ | 300 | 1 |
| | 400 | 2 |

2. Find the name and numbers of all salesmen who had more than one customer.

```

SELECT salesman_id, NAME
FROM salesman a
WHERE 1 < (SELECT count(*)
FROM customer
WHERE salesman_id=a.salesman_id);

```

| | salesman_id | NAME |
|---|-------------|------|
| ▶ | 1000 | john |
| | 2000 | ravi |
| * | NULL | NULL |

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

```

SELECT salesman.salesman_id, NAME, cust_name, commission
FROM salesman, customer
WHERE salesman.city = customer.city
UNION
SELECT salesman_id, name, 'no customer', commission
FROM salesman
WHERE NOT city = ANY
(SELECT city
FROM customer)
ORDER BY 2 DESC;

```

| | salesman_id | NAME | cust_name | commission |
|---|-------------|--------|-------------|------------|
| ▶ | 4000 | smith | no customer | 30 % |
| | 2000 | ravi | preethi | 20 % |
| | 2000 | ravi | chethan | 20 % |
| | 2000 | ravi | mamatha | 20 % |
| | 3000 | kumar | no customer | 15 % |
| | 1000 | john | preethi | 25 % |
| | 1000 | john | chethan | 25 % |
| | 1000 | john | mamatha | 25 % |
| | 5000 | harsha | no customer | 15 % |

4. Create a view that finds the salesman who has the customer with the highest order of a day.

```

CREATE VIEW highsalesman AS
SELECT b.ord_date, a.salesman_id, a.name
FROM salesman a, orders b
WHERE a.salesman_id = b.salesman_id
AND b.purchase_amt=(SELECT max(purchase_amt)
FROM orders c
WHERE c.ord_date = b.ord_date);
SELECT * FROM highsalesman;

```

| | ord_date | salesman_id | name |
|---|------------|-------------|-------|
| ▶ | 2004-06-17 | 1000 | john |
| | 2020-01-17 | 2000 | ravi |
| | 2024-02-17 | 2000 | ravi |
| | 2013-04-17 | 3000 | kumar |
| | 2009-03-17 | 2000 | ravi |

5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

```

DELETE FROM salesman
WHERE salesman_id=1000;
SELECT * FROM salesman;
SELECT * FROM orders;

```

| | salesman_id | name | city | commission |
|---|-------------|--------|-----------|------------|
| ▶ | 2000 | ravi | bangalore | 20 % |
| | 3000 | kumar | mysore | 15 % |
| | 4000 | smith | delhi | 30 % |
| | 5000 | harsha | hydrabad | 15 % |
| * | NULL | NULL | NULL | NULL |

| | ord_no | purchase_amt | ord_date | customer_id | salesman_id |
|---|--------|--------------|------------|-------------|-------------|
| ▶ | 51 | 450 | 2020-01-17 | 10 | 2000 |
| | 52 | 1000 | 2024-02-17 | 13 | 2000 |
| | 53 | 3500 | 2013-04-17 | 14 | 3000 |
| | 54 | 550 | 2009-03-17 | 12 | 2000 |
| * | NULL | NULL | NULL | NULL | NULL |

Program 7 -Book Database

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

PUBLISHER (Name, Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address)

```
create database Books;
```

```
use Books;
```

```
create table publisher (
```

```
    name varchar (20) primary key,
```

```
    phone integer,
```

```
    address varchar (20)
```

```
);
```

```
desc publisher;
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|-------------|------|-----|---------|-------|
| ► | name | varchar(20) | NO | PRI | NULL | |
| | phone | int | YES | | NULL | |
| | address | varchar(20) | YES | | NULL | |

```
create table book (
```

```
    book_id integer primary key,
```

```
    title varchar (20),
```

```
    pub_year varchar (20),
```

```
    publisher_name varchar (20),
```

```
    foreign key (publisher_name) references publisher (name) on delete cascade
```

```
);
```

```
desc book;
```

| | Field | Type | Null | Key | Default | Extra |
|---|----------------|-------------|------|-----|---------|-------|
| ► | book_id | int | NO | PRI | NULL | |
| | title | varchar(20) | YES | | NULL | |
| | pub_year | varchar(20) | YES | | NULL | |
| | publisher_name | varchar(20) | YES | MUL | NULL | |

```

create table book_authors (
    author_name varchar (20),
    book_id integer,
    foreign key (book_id) references book (book_id) on delete cascade,
    primary key (book_id, author_name)
);

desc book_authors;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ► | author_name | varchar(20) | NO | PRI | NULL | |
| | book_id | int | NO | PRI | NULL | |

```

create table library_branch (
    branch_id integer primary key,
    branch_name varchar (50),
    address varchar (50)
);

desc library_branch;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------------|-------------|------|-----|---------|-------|
| ► | branch_id | int | NO | PRI | NULL | |
| | branch_name | varchar(50) | YES | | NULL | |
| | address | varchar(50) | YES | | NULL | |

```

create table book_copies (
    no_of_copies integer,
    book_id integer,
    branch_id integer,
    foreign key (book_id) references book (book_id) on delete cascade,
    foreign key (branch_id) references library_branch (branch_id) on delete cascade,
    primary key (book_id, branch_id)
);

desc book_copies;

```

| | no_of_copies | book_id | branch_id |
|---|--------------|---------|-----------|
| ▶ | 10 | 1 | 10 |
| | 5 | 1 | 11 |
| | 2 | 2 | 12 |
| | 5 | 2 | 13 |
| | 7 | 3 | 14 |
| | 3 | 4 | 11 |
| | 1 | 5 | 10 |
| * | NULL | NULL | NULL |

```
create table card (
    card_no integer primary key
);
desc card;
```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|------|------|-----|---------|-------|
| ▶ | card_no | int | NO | PRI | NULL | |

```
create table book_lending (
    date_out date,
    due_date date,
    book_id integer,
    branch_id integer,
    card_no integer,
    foreign key (book_id) references book (book_id) on delete cascade,
    foreign key (branch_id) references library_branch (branch_id) on delete cascade,
    foreign key (card_no) references card (card_no) on delete cascade,
    primary key (book_id, branch_id, card_no)
);
desc book_lending;
```

| | Field | Type | Null | Key | Default | Extra |
|---|-----------|------|------|-----|---------|-------|
| ▶ | date_out | date | YES | | NULL | |
| | due_date | date | YES | | NULL | |
| | book_id | int | NO | PRI | NULL | |
| | branch_id | int | NO | PRI | NULL | |
| | card_no | int | NO | PRI | NULL | |

```

insert into publisher values ('mcgraw-hill', 99890, 'bangalore');
insert into publisher values ('pearson', 98890, 'newdelhi');
insert into publisher values ('random house', 74556, 'hyderabad');
insert into publisher values ('hachette livre', 897086, 'chenai');
insert into publisher values ('grupo planeta', 77561, 'bangalore');
select * from publisher;

```

| | name | phone | address |
|---|----------------|--------|-----------|
| ▶ | grupo planeta | 77561 | bangalore |
| | hachette livre | 897086 | chenai |
| | mcgraw-hill | 99890 | bangalore |
| | pearson | 98890 | newdelhi |
| | random house | 74556 | hyderabad |
| • | NULL | NULL | NULL |

```

insert into book values (1,'dbms','01-2017', 'mcgraw-hill');
insert into book values (2,'adbms','06-2016', 'mcgraw-hill');
insert into book values (3,'cn','09-2016', 'pearson');
insert into book values (4,'cg','09-2015', 'grupo planeta');
insert into book values (5,'os','05-2016', 'pearson');
select * from book;

```

| | book_id | title | pub_year | publisher_name |
|---|---------|-------|----------|----------------|
| ▶ | 1 | dbms | 01-2017 | mcgraw-hill |
| | 2 | adbms | 06-2016 | mcgraw-hill |
| | 3 | cn | 09-2016 | pearson |
| | 4 | cg | 09-2015 | grupo planeta |
| | 5 | os | 05-2016 | pearson |
| • | NULL | NULL | NULL | NULL |

```

insert into book_authors values ('navathe', 1);
insert into book_authors values ('navathe', 2);
insert into book_authors values ('tanenbaum', 3);
insert into book_authors values ('edward angel', 4);
insert into book_authors values ('galvin', 5);

```

```
select * from book_authors;
```

| | author_name | book_id |
|---|--------------|---------|
| ▶ | navathe | 1 |
| | navathe | 2 |
| | tanenbaum | 3 |
| | edward angel | 4 |
| | galvin | 5 |
| • | NULL | NULL |

```
insert into library_branch values (10,'rr nagar','bangalore');
```

```
insert into library_branch values (11,'rnsit','bangalore');
```

```
insert into library_branch values (12,'rajaji nagar', 'bangalore');
```

```
insert into library_branch values (13,'nitte','mangalore');
```

```
insert into library_branch values (14,'manipal','udupi');
```

```
select * from library_branch;
```

| | branch_id | branch_name | address |
|---|-----------|--------------|-----------|
| ▶ | 10 | rr nagar | bangalore |
| | 11 | rnsit | bangalore |
| | 12 | rajaji nagar | bangalore |
| | 13 | nitte | mangalore |
| | 14 | manipal | udupi |
| • | NULL | NULL | NULL |

```
insert into book_copies values (10, 1, 10);
```

```
insert into book_copies values (5, 1, 11);
```

```
insert into book_copies values (2, 2, 12);
```

```
insert into book_copies values (5, 2, 13);
```

```
insert into book_copies values (7, 3, 14);
```

```
insert into book_copies values (1, 5, 10);
```

```
insert into book_copies values (3, 4, 11);
```

```
select * from book_copies;
```

| | Field | Type | Null | Key | Default | Extra |
|---|--------------|------|------|-----|---------|-------|
| ▶ | no_of_copies | int | YES | | NULL | |
| | book_id | int | NO | PRI | NULL | |
| | branch_id | int | NO | PRI | NULL | |


```
insert into card values (100);
```

```
insert into card values (101);
```

```
insert into card values (102);
```

```
insert into card values (103);
```

```
insert into card values (104);
```

```
select * from card;
```

| | card_no |
|---|---------|
| ▶ | 100 |
| | 101 |
| | 102 |
| | 103 |
| | 104 |
| • | NULL |

```
insert into book_lending values ('01-01-17','01-06-17', 1, 10, 101);
```

```
insert into book_lending values ('11-01-17','11-03-17', 3, 14, 101);
```

```
insert into book_lending values ('21-02-17','21-04-17', 2, 13, 101);
```

```
insert into book_lending values ('15-03-17','15-07-17', 4, 11, 101);
```

```
insert into book_lending values ('12-08-17','12-08-17', 1, 11, 104);
```

```
select * from book_lending;
```

| | date_out | due_date | book_id | branch_id | card_no |
|---|------------|------------|---------|-----------|---------|
| ▶ | 2001-01-17 | 2001-06-17 | 1 | 10 | 101 |
| | 2012-08-17 | 2012-08-17 | 1 | 11 | 104 |
| | 2021-02-17 | 2021-04-17 | 2 | 13 | 101 |
| | 2011-01-17 | 2011-03-17 | 3 | 14 | 101 |
| | 2015-03-17 | 2015-07-17 | 4 | 11 | 101 |
| • | NULL | NULL | NULL | NULL | NULL |

Write SQL queries to

1.Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.

```
select b.book_id, b.title, b.pub_year, b.publisher_name, bc.no_of_copies, ba.author_name, lb.branch_name from book b, book_authors ba, book_copies bc, book_branches lb;
```

library_branch lb, book_copies bc where b.book_id = ba.book_id and b.book_id = bc.book_id and lb.branch_id = bc.branch_id;

| | book_id | title | pub_year | publisher_name | no_of_copies | author_name | branch_name |
|---|---------|-------|----------|----------------|--------------|--------------|--------------|
| ▶ | 1 | dbms | 01-2017 | mcgraw-hill | 10 | navathe | rr nagar |
| | 1 | dbms | 01-2017 | mcgraw-hill | 5 | navathe | rnsit |
| | 2 | adbms | 06-2016 | mcgraw-hill | 2 | navathe | rajaji nagar |
| | 2 | adbms | 06-2016 | mcgraw-hill | 5 | navathe | nitte |
| | 3 | cn | 09-2016 | pearson | 7 | tanenbaum | manipal |
| | 4 | cg | 09-2015 | grupo planeta | 3 | edward angel | rnsit |
| | 5 | os | 05-2016 | pearson | 1 | galvin | rr nagar |

2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017

select card_no from book_lending where year(date_out) > 17 and month(date_out) < 7 group by card_no having count(card_no) > 2 ;

| | card_no |
|---|---------|
| ▶ | 101 |

3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.

delete from book where book_id = 3;

select * from book;

select * from book_authors;

select * from book_copies;

select * from book_lending;

| | book_id | title | pub_year | publisher_name |
|---|---------|-------|----------|----------------|
| ▶ | 1 | dbms | 01-2017 | mcgraw-hill |
| | 2 | adbms | 06-2016 | mcgraw-hill |
| | 4 | cg | 09-2015 | grupo planeta |
| | 5 | os | 05-2016 | pearson |
| ★ | NULL | NULL | NULL | NULL |

| | author_name | book_id |
|---|--------------|---------|
| ▶ | navathe | 1 |
| | navathe | 2 |
| | edward angel | 4 |
| | galvin | 5 |
| * | NULL | NULL |

| | no_of_copies | book_id | branch_id |
|---|--------------|---------|-----------|
| ▶ | 10 | 1 | 10 |
| | 5 | 1 | 11 |
| | 2 | 2 | 12 |
| | 5 | 2 | 13 |
| | 3 | 4 | 11 |
| | 1 | 5 | 10 |
| * | NULL | NULL | NULL |

| | date_out | due_date | book_id | branch_id | card_no |
|---|------------|------------|---------|-----------|---------|
| ▶ | 2001-01-17 | 2001-06-17 | 1 | 10 | 101 |
| | 2012-08-17 | 2012-08-17 | 1 | 11 | 104 |
| | 2021-02-17 | 2021-04-17 | 2 | 13 | 101 |
| | 2015-03-17 | 2015-07-17 | 4 | 11 | 101 |
| * | NULL | NULL | NULL | NULL | NULL |

4.Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

create view partition select pub_year from book;

select * from partition;

| | pub_year |
|---|----------|
| ▶ | 01-2017 |
| | 06-2016 |
| | 09-2015 |
| | 05-2016 |

5.Create a view of all books and its number of copies that are currently available in the Library.

```
create view book_copies as select b.book_id, b.title, bc.no_of_copies from book b,
book_copies bc where b.book_id = bc.book_id;

select * from book_copies;
```

| | book_id | title | no_of_copies |
|---|---------|-------|--------------|
| ▶ | 1 | dbms | 10 |
| | 1 | dbms | 5 |
| | 2 | adbms | 2 |
| | 2 | adbms | 5 |
| | 4 | cg | 3 |
| | 5 | os | 1 |

Program 8-STUDENT ENROLLMENT DATABASE

Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)

COURSE (course #:int, cname:string, dept:string)

ENROLL (regno:string, course#:int, sem:int, marks:int)

BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)

TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

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

```
create database student_enroll;

use student_enroll;
```

```
create table student(  
    regno varchar(15),  
    name varchar(20),  
    major varchar(20),  
    bdate date,  
    primary key(regno));  
desc student;
```

```
create table course(  
    courseno int,  
    cname varchar(20),  
    dept varchar(20),  
    primary key(courseno));  
desc course;
```

```
create table enroll(  
    regno varchar(15),  
    courseno int,  
    sem int,  
    marks int,  
    primary key(regno,courseno),  
    foreign key(regno) references student(regno),  
    foreign key(courseno) references course(courseno));  
desc enroll;
```

```
create table textbook(  
    book_isbn int,  
    book_title varchar(20),  
    publisher varchar(20),  
    author varchar(20),  
    primary key(book_isbn));
```

```
desc textbook;
```

```
create table book_adoption(  
  courseno int,  
  sem int,  
  book_isbn int,  
  primary key(courseno,book_isbn),  
  foreign key(courseno) references course(courseno),  
  foreign key(book_isbn) references textbook(book_isbn));  
desc book_adoption;
```

ii. Enter at least five tuples for each relation.

```
insert into student values('1BM11CS001','A','Sr','19931230');  
insert into student values('1BM11CS002','B','Sr','19930924');  
insert into student values('1BM11CS003','C','Sr','19931127');  
insert into student values('1BM11CS004','D','Sr','19930413');  
insert into student values('1BM11CS005','E','Jr','19940824');  
commit;  
select * from student;
```

| regno | name | major | bdate |
|------------|------|-------|------------|
| 1BM11CS001 | A | Sr | 1993-12-30 |
| 1BM11CS002 | B | Sr | 1993-09-24 |
| 1BM11CS003 | C | Sr | 1993-11-27 |
| 1BM11CS004 | D | Sr | 1993-04-13 |
| 1BM11CS005 | E | Jr | 1994-08-24 |
| * | NULL | NULL | NULL |

```
insert into course values(111,'OS','CSE');  
insert into course values(112,'EC','ECE');  
insert into course values(113,'SS','ISE');  
insert into course values(114,'DBMS','CSE');  
insert into course values(115,'SIGNALS','ECE');
```

commit;

select * from course;

| | courseno | cname | dept |
|---|----------|---------|------|
| ▶ | 111 | OS | CSE |
| | 112 | EC | ECE |
| | 113 | SS | ISE |
| | 114 | DBMS | CSE |
| | 115 | SIGNALS | ECE |
| * | NULL | NULL | NULL |

insert into textbook values(10,'DATABASE SYSTEMS','PEARSON','SCHIELD');

insert into textbook values(900,'OPERATING SYSTEMS','PEARSON','LELAND');

insert into textbook values(901,'CIRCUITS','HALL INDIA','BOB');

insert into textbook values(902,'SYSTEM SOFTWARE','PETERSON','JACOB');

insert into textbook values(903,'SCHEDULING','PEARSON','PATIL');

insert into textbook values(904,'DATABASE SYSTEMS','PEARSON','JACOB');

insert into textbook values(905,'DATABASE MANAGER','PEARSON','BOB');

insert into textbook values(906,'SIGNALS','HALL INDIA','SUMIT');

commit;

select * from textbook;

| | book_isbn | book_title | publisher | author |
|---|-----------|-------------------|------------|---------|
| ▶ | 10 | DATABASE SYSTEMS | PEARSON | SCHIELD |
| | 900 | OPERATING SYSTEMS | PEARSON | LELAND |
| | 901 | CIRCUITS | HALL INDIA | BOB |
| | 902 | SYSTEM SOFTWARE | PETERSON | JACOB |
| | 903 | SCHEDULING | PEARSON | PATIL |
| | 904 | DATABASE SYSTEMS | PEARSON | JACOB |
| | 905 | DATABASE MANAGER | PEARSON | BOB |
| | 906 | SIGNALS | HALL INDIA | SUMIT |
| * | NULL | NULL | NULL | NULL |

insert into enroll values('1BM11CS001',115,3,100);

insert into enroll values('1BM11CS002',114,5,100);

insert into enroll values('1BM11CS003',113,5,100);

insert into enroll values('1BM11CS004',111,5,100);

insert into enroll values('1BM11CS005',112,3,100);

commit;

select * from enroll;

| | regno | courseno | sem | marks |
|---|------------|----------|------|-------|
| ▶ | 1BM11CS001 | 115 | 3 | 100 |
| | 1BM11CS002 | 114 | 5 | 100 |
| | 1BM11CS003 | 113 | 5 | 100 |
| | 1BM11CS004 | 111 | 5 | 100 |
| | 1BM11CS005 | 112 | 3 | 100 |
| * | NULL | NULL | NULL | NULL |

insert into book_adoption values(111,5,900);

insert into book_adoption values(111,5,903);

insert into book_adoption values(111,5,904);

insert into book_adoption values(112,3,901);

insert into book_adoption values(113,3,10);

insert into book_adoption values(114,5,905);

insert into book_adoption values(113,5,902);

insert into book_adoption values(115,3,906);

commit;

select * from book_adoption;

| | courseno | sem | book_jsbn |
|---|----------|------|-----------|
| ▶ | 111 | 5 | 900 |
| | 111 | 5 | 903 |
| | 111 | 5 | 904 |
| | 112 | 3 | 901 |
| | 113 | 3 | 10 |
| | 113 | 5 | 902 |
| | 114 | 5 | 905 |
| | 115 | 3 | 906 |
| * | NULL | NULL | NULL |

iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.

insert into textbook values(908,'UNIX CONCEPTS','TATA MCGRAW HILL','SUMITABHA DAS');

insert into book_adoption values(113,4,908);

select * from textbook;


```
select * from book_adoption;
```

| | book_isbn | book_title | publisher | author |
|---|-----------|-------------------|------------------|---------------|
| ▶ | 10 | DATABASE SYSTEMS | PEARSON | SCHIELD |
| | 900 | OPERATING SYSTEMS | PEARSON | LELAND |
| | 901 | CIRCUITS | HALL INDIA | BOB |
| | 902 | SYSTEM SOFTWARE | PETERSON | JACOB |
| | 903 | SCHEDULING | PEARSON | PATIL |
| | 904 | DATABASE SYSTEMS | PEARSON | JACOB |
| | 905 | DATABASE MANAGER | PEARSON | BOB |
| | 906 | SIGNALS | HALL INDIA | SUMIT |
| | 908 | UNIX CONCEPTS | TATA MCGRAW HILL | SUMITABHA DAS |
| • | NULL | NULL | NULL | NULL |

| | courseno | sem | book_isbn |
|---|----------|------|-----------|
| ▶ | 111 | 5 | 900 |
| | 111 | 5 | 903 |
| | 111 | 5 | 904 |
| | 112 | 3 | 901 |
| | 113 | 3 | 10 |
| | 113 | 5 | 902 |
| | 113 | 4 | 908 |
| | 114 | 5 | 905 |
| | 115 | 3 | 906 |
| • | NULL | NULL | NULL |

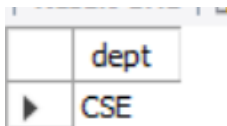
iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

```
select c.courseno,t.book_isbn,t.book_title
from course c,book_adoption ba,textbook t
where c.courseno=ba.courseno
and ba.book_isbn=t.book_isbn
and c.dept='CSE'
and 2<(select COUNT(book_isbn)
from book_adoption b
where c.courseno=b.courseno)
order by t.book_title;
```

| | courseno | book_isbn | book_title |
|---|----------|-----------|-------------------|
| ▶ | 111 | 904 | DATABASE SYSTEMS |
| | 111 | 900 | OPERATING SYSTEMS |
| | 111 | 903 | SCHEDULING |

v. List any department that has all its adopted books published by a specific publisher.

```
select distinct c.dept
from course c
where c.dept in(select c.dept
                from course c,book_adoption b,textbook t
                where c.courseno=b.courseno
                and t.book_isbn=b.book_isbn
                and t.publisher='PEARSON')
and c.dept not in(select c.dept
                  from course c,book_adoption b,textbook t
                  where c.courseno=b.courseno
                  and t.book_isbn=b.book_isbn
                  and t.publisher != 'PEARSON');
```



| | |
|---|------|
| | dept |
| ▶ | CSE |

Program 9: Movie database

Consider the schema for Movie Database:

ACTOR (Act_id, Act_Name, Act_Gender)

DIRECTOR (Dir_id, Dir_Name, Dir_Phone)

MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST (Act_id, Mov_id, Role)

RATING (Mov_id, Rev_Stars)

```
create database movie;
```

```
use movie;
```

```
create table actor(
```

```
act_id int,
```

```
act_name varchar(20),
```

```
act_gender char(1),
```

```
primary key(act_id));
```

```
desc actor;
```

```
create table director(
```

```
dir_id int,
```

```
dir_name varchar(20),
```

```
dir_phone int(10),
```

```
primary key(dir_id));
```

```
desc director;
```

```
alter table director
```

```
modify column dir_phone bigint;
```

```
desc director;
```

```
create table movies(
```

```
mov_id int,
```

```
mov_title varchar(25),
```

```
mov_year int,
```

```
mov_lang varchar(12),
```

```
dir_id int,
```

```
primary key(mov_id),
```

```
foreign key(dir_id) references director(dir_id));
```

```
desc movies;
```

```
create table movie_cast(  
  act_id int,  
  mov_id int,  
  role varchar(10),  
  primary key(act_id,mov_id),  
  foreign key(act_id) references actor(act_id),  
  foreign key(mov_id) references movies(mov_id));  
desc movie_cast;
```

```
create table rating(  
  mov_id int,  
  rev_stars varchar(25),  
  primary key(mov_id),  
  foreign key(mov_id) references movies(mov_id));  
desc rating;
```

```
insert into actor values(301,'ANUSHKA','F');  
insert into actor values (302,'PRABHAS','M');  
insert into actor values(303,'PUNITH','M');  
insert into actor values(304,'JERMY','M');  
commit;  
select * from actor;
```

| | act_id | act_name | act_gender |
|---|--------|----------|------------|
| ▶ | 301 | ANUSHKA | F |
| | 302 | PRABHAS | M |
| | 303 | PUNITH | M |
| | 304 | JERMY | M |
| * | NULL | NULL | NULL |

```
insert into director values(60,'RAJAMOULI', 8751611001);  
insert into director values(61,'HITCHCOCK', 7766138911);
```

| dir_id | dir_name | dir_phone |
|--------|------------------|------------|
| 60 | RAJAMOULI | 8751611001 |
| 61 | HITCHCOCK | 7766138911 |
| 62 | FARAN | 9986776531 |
| 63 | STEVEN SPIELBERG | 8989776530 |
| NULL | NULL | NULL |

insert into director values(62,'FARAN', 9986776531);

insert into director values(63,'STEVEN SPIELBERG', 8989776530);

commit;

select * from director;

insert into movies values(1001,'BAHUBALI-2', 2017, 'TELUGU', 60);

insert into movies values(1002,'BAHUBALI-1', 2015, 'TELUGU', 60);

insert into movies values(1003,'AKASH', 2008, 'KANNADA', 61);

insert into movies values(1004,'WAR HORSE', 2011, 'ENGLISH', 63);

commit;

select * from movies;

| mov_id | mov_title | mov_year | mov_lang | dir_id |
|--------|------------|----------|----------|--------|
| 1001 | BAHUBALI-2 | 2017 | TELUGU | 60 |
| 1002 | BAHUBALI-1 | 2015 | TELUGU | 60 |
| 1003 | AKASH | 2008 | KANNADA | 61 |
| 1004 | WAR HORSE | 2011 | ENGLISH | 63 |
| NULL | NULL | NULL | NULL | NULL |

insert into movie_cast values(301, 1002, 'HEROINE');

insert into movie_cast values(301, 1001, 'HEROINE');

insert into movie_cast values(303, 1003, 'HERO');

insert into movie_cast values(303, 1002, 'GUEST');

insert into movie_cast values(304, 1004, 'HERO');

commit;

select * from movie_cast;

| | act_id | mov_id | role |
|---|--------|--------|---------|
| ▶ | 301 | 1001 | HEROINE |
| | 301 | 1002 | HEROINE |
| | 303 | 1002 | GUEST |
| | 303 | 1003 | HERO |
| | 304 | 1004 | HERO |
| * | NULL | NULL | NULL |

insert into rating values(1001, 4);

insert into rating values(1002, 2);

insert into rating values(1003, 5);

insert into rating values(1004, 4);

commit;

select * from rating;

| | mov_id | rev_stars |
|---|--------|-----------|
| ▶ | 1001 | 4 |
| | 1002 | 2 |
| | 1003 | 5 |
| | 1004 | 4 |
| * | NULL | NULL |

1. List the titles of all movies directed by 'Hitchcock'.

select mov_title

from movies

where dir_id=(select dir_id from director where dir_name='Hitchcock')

group by mov_title;

| | mov_title |
|---|-----------|
| ▶ | AKASH |

2. Find the movie names where one or more actors acted in two or more movies.

```

select m.mov_title
from movies m, movie_cast mc
where m.mov_id=mc.mov_id
and mc.act_id in( select act_id from movie_cast group by act_id having count(act_id)>1)
group by mov_title
having count(*)>1;

```

| | mov_title |
|---|------------|
| ▶ | BAHUBALI-1 |

3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

```

select act_name,mov_title,mov_year
from actor a
join movie_cast mc
on a.act_id=mc.act_id
join movies m
on m.mov_id=mc.mov_id
where m.mov_year not between 2000 and 2015;

```

| | act_name | mov_title | mov_year |
|---|----------|------------|----------|
| ▶ | ANUSHKA | BAHUBALI-2 | 2017 |

4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

```

select mov_title,max(rev_stars)
from movies
inner join rating using(mov_id)
group by mov_id
having max(rev_stars)>0

```

order by mov_title;

| | mov_title | max(rev_stars) |
|---|------------|----------------|
| ▶ | AKASH | 5 |
| | BAHUBALI-1 | 2 |
| | BAHUBALI-2 | 4 |
| | WAR HORSE | 4 |

5. Update rating of all movies directed by 'Steven Spielberg' to 5.

update rating

set rev_stars=5

where mov_id in(select mov_id from movies

where dir_id in(select dir_id from director

where dir_name='Steven Spielberg'));

select * from rating;

| | mov_id | rev_stars |
|---|--------|-----------|
| ▶ | 1001 | 4 |
| | 1002 | 2 |
| | 1003 | 5 |
| | 1004 | 5 |
| * | NULL | NULL |

Program 10 -College Database

Consider the schema for College Database:

STUDENT (USN, SName, Address, Phone, Gender)

SEMSEC (SSID, Sem, Sec)

CLASS (USN, SSID)

SUBJECT (Subcode, Title, Sem, Credits)

IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

create database colg_db;


```

use colg_db;

create table student(

    usn varchar(30),

    sname varchar(30),

    address varchar(30),

    phone real,

    gender varchar(30),

    primary key(usn)

);

desc student;

```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|-------------|------|-----|---------|-------|
| ► | usn | varchar(30) | NO | PRI | NULL | |
| | sname | varchar(30) | YES | | NULL | |
| | address | varchar(30) | YES | | NULL | |
| | phone | double | YES | | NULL | |
| | gender | varchar(30) | YES | | NULL | |

```

create table semsec(

    ssid varchar(30),

    sem int,

    sec varchar(30),

    primary key(ssid)

);

desc semsec;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | ssid | varchar(30) | NO | PRI | NULL | |
| | sem | int | YES | | NULL | |
| | sec | varchar(30) | YES | | NULL | |

```

create table class(

    usn varchar(30),

    ssid varchar(30),

```

```

        primary key(usn,ssid),

        foreign key(usn) REFERENCES student(usn),

        foreign key(ssid) REFERENCES semsec(ssid)

    );

desc class;

```

| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | usn | varchar(30) | NO | PRI | NULL | |
| | ssid | varchar(30) | NO | PRI | NULL | |

```

create table subject(

    code varchar(30),

    title varchar(30),

    sem int,

    credits int,

    primary key(code)

);

desc subject;

```

| | Field | Type | Null | Key | Default | Extra |
|---|---------|-------------|------|-----|---------|-------|
| ► | code | varchar(30) | NO | PRI | NULL | |
| | title | varchar(30) | YES | | NULL | |
| | sem | int | YES | | NULL | |
| | credits | int | YES | | NULL | |

```

create table marks(

    usn varchar(30),code varchar(30),

    ssid varchar(30),

    test1 real, test2 real, test3 real, final real,

    primary key(usn,code,ssid),

    foreign key(usn) REFERENCES student(usn),

    foreign key(code) REFERENCES subject(code),

```

foreign key(ssid) REFERENCES semsec(ssid)

);

desc marks;

| | Field | Type | Null | Key | Default | Extra |
|---|-------|-------------|------|-----|---------|-------|
| ► | usn | varchar(30) | NO | PRI | NULL | |
| | code | varchar(30) | NO | PRI | NULL | |
| | ssid | varchar(30) | NO | PRI | NULL | |
| | test1 | double | YES | | NULL | |
| | test2 | double | YES | | NULL | |
| | test3 | double | YES | | NULL | |
| | final | double | YES | | NULL | |

insert into student values('1RN13CS020','akshay','belagavi',8877881122,'m'),

('1RN13CS062','sandhya','bengaluru',7722829912,'f'),

('1RN13CS091','teesha','bengaluru',7712312312,'f'),

('1RN13CS066','supriya','mangaluru',8877881122,'f'),

('1RN14CS010','abhay','bengaluru',9900211201,'m'),

('1RN14CS032','bhaskar','bengaluru',9923211099,'m'),

('1RN14CS025','asmi','bengaluru',7894737377,'f'),

('1RN15CS011','ajay','tumkur',98545091341,'m'),

('1RN15CS029','chitra','davangere',7696772121,'f'),

('1RN15CS045','jeeva','bellary',9944850121,'m'),

('1RN15CS091','santosh','mangaluru',8812332201,'m'),

('1RN16CS045','ismail','kalburgi',9900232201,'m'),

('1RN16CS088','sameera','shimoga',9905542212,'f'),

('1RN16CS122','vinayaka','chikamagaluru',8800880011,'m');

select * from student;

| | usn | sname | address | phone | gender |
|---|------------|----------|-------------|-------------|--------|
| ► | 1RN13CS020 | akshay | belagavi | 8877881122 | m |
| | 1RN13CS062 | sandhya | bengaluru | 7722829912 | f |
| | 1RN13CS066 | supriya | mangaluru | 8877881122 | f |
| | 1RN13CS091 | teesha | bengaluru | 7712312312 | f |
| | 1RN14CS010 | abhay | bengaluru | 9900211201 | m |
| | 1RN14CS025 | asmi | bengaluru | 7894737377 | f |
| | 1RN14CS032 | bhaskar | bengaluru | 9923211099 | m |
| | 1RN15CS011 | ajay | tumkur | 98545091341 | m |
| | 1RN15CS029 | chitra | davangere | 7696772121 | f |
| | 1RN15CS045 | jeeva | bellary | 9944850121 | m |
| | 1RN15CS091 | santosh | mangaluru | 8812332201 | m |
| | 1RN16CS045 | ismail | kalburgi | 9900232201 | m |
| | 1RN16CS088 | sameera | shimoga | 9905542212 | f |
| | 1RN16CS122 | vinayaka | chikamag... | 8800880011 | m |
| ★ | NULL | NULL | NULL | NULL | NULL |

```

insert into semsec values('CSE8A',8,'A'),
('CSE8B',8,'B'),('CSE8C',8,'C'),
('CSE7A',7,'A'),('CSE7B',7,'B'),('CSE7C',7,'C'),
('CSE6A',6,'A'),('CSE6B',6,'B'),('CSE6C',6,'C'),
('CSE5A',5,'A'),('CSE5B',5,'B'),('CSE5C',5,'C'),
('CSE4A',4,'A'),('CSE4B',4,'B'),('CSE4C',4,'C'),
('CSE3A',3,'A'),('CSE3B',3,'B'),('CSE3C',3,'C'),
('CSE2A',2,'A'),('CSE2B',2,'B'),('CSE2C',2,'C'),
('CSE1A',1,'A'),('CSE1B',1,'B'),('CSE1C',1,'C');

select * from semsec;

```

| | ssid | sem | sec |
|---|-------|-----|-----|
| ▶ | CSE1A | 1 | A |
| | CSE1B | 1 | B |
| | CSE1C | 1 | C |
| | CSE2A | 2 | A |
| | CSE2B | 2 | B |
| | CSE2C | 2 | C |
| | CSE3A | 3 | A |
| | CSE3B | 3 | B |
| | CSE3C | 3 | C |
| | CSE4A | 4 | A |
| | CSE4B | 4 | B |
| | CSE4C | 4 | C |
| | CSE5A | 5 | A |
| | CSE5B | 5 | B |
| | CSE5C | 5 | C |

| | | | |
|---|-------|------|------|
| | CSE6A | 6 | A |
| | CSE6B | 6 | B |
| | CSE6C | 6 | C |
| | CSE7A | 7 | A |
| | CSE7B | 7 | B |
| | CSE7C | 7 | C |
| | CSE8A | 8 | A |
| | CSE8B | 8 | B |
| | CSE8C | 8 | C |
| * | NULL | NULL | NULL |

```

insert into class values('1RN13CS020','CSE8A'),
('1RN13CS062','CSE8A'),('1RN13CS066','CSE8B'),('1RN13CS091','CSE8C'),
('1RN14CS010','CSE7A'),('1RN14CS025','CSE7A'),('1RN14CS032','CSE7A'),
('1RN15CS011','CSE4A'),('1RN15CS029','CSE4A'),('1RN15CS045','CSE4B'),
('1RN15CS091','CSE4C'),('1RN16CS045','CSE3A'),('1RN16CS088','CSE3B'),
('1RN16CS122','CSE3C');

select * from class;

```

| | usn | ssid |
|---|------------|-------|
| ▶ | 1RN16CS045 | CSE3A |
| | 1RN16CS088 | CSE3B |
| | 1RN16CS122 | CSE3C |
| | 1RN15CS011 | CSE4A |
| | 1RN15CS029 | CSE4A |
| | 1RN15CS045 | CSE4B |
| | 1RN15CS091 | CSE4C |
| | 1RN14CS010 | CSE7A |
| | 1RN14CS025 | CSE7A |
| | 1RN14CS032 | CSE7A |
| | 1RN13CS020 | CSE8A |
| | 1RN13CS062 | CSE8A |
| | 1RN13CS066 | CSE8B |
| | 1RN13CS091 | CSE8C |
| ★ | HULL | HULL |

```

insert into subject values('10CS81','ACA',8,4),
('10CS82','SSM',8,4),('10CS83','NM',8,4),
('10CS84','CC',8,4),('10CS85','PW',8,4),
('10CS71','OOAD',7,4),('10CS72','ECS',7,4),
('10CS73','PTW',7,4),('10CS74','DWDM',7,4),
('10CS75','JAVA',7,4),('10CS76','SAN',7,4),
('10CS51','ME',5,4),('10CS52','CN',5,4),
('10CS53','DBMS',5,4),('10CS54','ATC',5,4),
('10CS55','JAVA',5,3),('10CS56','AI',5,3),
('10CS41','M4',4,4),('10CS42','SE',4,4),
('10CS43','DAA',4,4),('10CS44','MPMC',4,4),
('10CS45','OOC',4,3),('10CS46','DC',4,3),
('10CS31','M3',3,4),('10CS32','ADE',3,4),
('10CS33','DSA',3,4),('10CS34','CO',3,4),
('10CS35','USP',3,3),('10CS36','DMS',3,3);

select * from subject;

```

| | code | title | sem | credits |
|---|--------|-------|-----|---------|
| ▶ | 10CS31 | M3 | 3 | 4 |
| | 10CS32 | ADE | 3 | 4 |
| | 10CS33 | DSA | 3 | 4 |
| | 10CS34 | CO | 3 | 4 |
| | 10CS35 | USP | 3 | 3 |
| | 10CS36 | DMS | 3 | 3 |
| | 10CS41 | M4 | 4 | 4 |
| | 10CS42 | SE | 4 | 4 |
| | 10CS43 | DAA | 4 | 4 |
| | 10CS44 | MPMC | 4 | 4 |
| | 10CS45 | OOC | 4 | 3 |
| | 10CS46 | DC | 4 | 3 |
| | 10CS51 | ME | 5 | 4 |
| | 10CS52 | CN | 5 | 4 |
| | 10CS53 | DBMS | 5 | 4 |
| | 10CS54 | ATC | 5 | 4 |

```
insert into marks(usn,code,ssid,test1,test2,test3) values('1RN13CS091','10CS81','CSE8C',15,16,18),
('1RN13CS091','10CS82','CSE8C',12,19,14),('1RN13CS091','10CS83','CSE8C',19,15,20),
('1RN13CS091','10CS84','CSE8C',20,16,19),('1RN13CS091','10CS85','CSE8C',15,15,12);

select * from marks;
```

| | usn | code | ssid | test1 | test2 | test3 | final |
|---|------------|--------|-------|-------|-------|-------|-------|
| ▶ | 1RN13CS091 | 10CS81 | CSE8C | 15 | 16 | 18 | NULL |
| | 1RN13CS091 | 10CS82 | CSE8C | 12 | 19 | 14 | NULL |
| | 1RN13CS091 | 10CS83 | CSE8C | 19 | 15 | 20 | NULL |
| | 1RN13CS091 | 10CS84 | CSE8C | 20 | 16 | 19 | NULL |
| | 1RN13CS091 | 10CS85 | CSE8C | 15 | 15 | 12 | NULL |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

i. List all the student details studying in fourth semester 'C' section.

```
select S.*, SS.sem, SS.sec
```

```
from student S, semsec SS, class C
```

```
where S.usn = C.usn AND SS.ssid = C.ssid AND SS.sem = 4 AND SS.sec = 'C';
```

| | usn | sname | address | phone | gender | sem | sec |
|---|------------|---------|-----------|------------|--------|-----|-----|
| ▶ | 1RN15CS091 | santosh | mangaluru | 8812332201 | m | 4 | C |

ii. Compute the total number of male and female students in each semester and in each section.

```
select SS.sem, SS.sec, S.gender, count(S.gender) as COUNT
```

```
from student S, semsec SS, class C
```

```
where S.usn = C.usn AND SS.ssid = C.ssid
```

```
group by SS.sem, SS.sec, S.gender ORDER by sem;
```

| | sem | sec | gender | COUNT |
|---|-----|-----|--------|-------|
| ► | 3 | A | m | 1 |
| | 3 | B | f | 1 |
| | 3 | C | m | 1 |
| | 4 | A | f | 1 |
| | 4 | A | m | 1 |
| | 4 | B | m | 1 |
| | 4 | C | m | 1 |
| | 7 | A | f | 1 |
| | 7 | A | m | 2 |
| | 8 | A | f | 1 |
| | 8 | A | m | 1 |
| | 8 | B | f | 1 |
| | 8 | C | f | 1 |

iii. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.

```
create view STU_test1_marks_view as
```

```
select test1, code
```

```
from marks
```

```
where usn = '1RN13CS091';
```

```
select * from STU_test1_marks_view;
```

| | test1 | code |
|---|-------|--------|
| ► | 15 | 10CS81 |
| | 12 | 10CS82 |
| | 19 | 10CS83 |
| | 20 | 10CS84 |
| | 15 | 10CS85 |

iv. Categorize students based on the following criterion:

If FinalIA = 17 to 20 then CAT = 'Outstanding'

If FinalIA = 12 to 16 then CAT = 'Average'

If FinalIA < 12 then CAT = 'Weak'

Give these details only for 8th semester A, B, and C section students.

select S.usn, S.sname, S.address, S.phone, S.gender,

(CASE

when IA.final between 17 and 20 then 'outstanding'

when IA.final between 12 and 16 then 'average'

else 'weak' end) AS CAT

from student S, semsec SS, marks IA, subject sub

where S.usn = IA.usn AND SS.ssid = IA.ssid AND sub.code = IA.code AND sub.sem = 8;

| | usn | sname | address | phone | gender | CAT |
|---|------------|--------|-----------|------------|--------|------|
| ▶ | 1RN13CS091 | teesha | bengaluru | 7712312312 | f | weak |
| | 1RN13CS091 | teesha | bengaluru | 7712312312 | f | weak |
| | 1RN13CS091 | teesha | bengaluru | 7712312312 | f | weak |
| | 1RN13CS091 | teesha | bengaluru | 7712312312 | f | weak |
| | 1RN13CS091 | teesha | bengaluru | 7712312312 | f | weak |