DBMS LAB 1 :-

```
CREATE TABLE PERSON(  driver_id VARCHAR(10),  name VARCHAR(20),  address VARCHAR(50),
 PRIMARY KEY(driver_id)
);
CREATE TABLE CAR(  regno
VARCHAR(10),  model
VARCHAR(20),  year
VARCHAR(4),
 PRIMARY KEY(regno)
);
CREATE TABLE ACCIDENT(
 report_no INT,  date DATE,
location VARCHAR(50),
 PRIMARY KEY(report_no)
);
CREATE        TABLE        OWNS(
driver_id VARCHAR(10),  regno
VARCHAR(20),
 PRIMARY KEY(driver_id,regno),
 FOREIGN KEY(driver_id) REFERENCES PERSON(driver_id)
 ON delete CASCADE ON update CASCADE,
 FOREIGN KEY(regno) REFERENCES CAR(regno)
 ON delete CASCADE ON update CASCADE);
CREATE TABLE PARTICIPATED(  driver_id
VARCHAR(10),
 regno VARCHAR(20),  report_no INT,
damage_amount INT,
 FOREIGN KEY(driver_id,regno) REFERENCES OWNS(driver_id,regno)
 ON delete SET NULL ON update CASCADE,
```

```
 FOREIGN KEY(report_no) REFERENCES ACCIDENT(report_no)
 ON delete SET NULL ON update CASCADE
);
INSERT INTO PERSON VALUES('A01','Richard','Srinivas nagar');
INSERT INTO PERSON VALUES('A02','Pradeep','Rajaji nagar');
INSERT INTO PERSON VALUES('A03','Smith','Ashok nagar');
INSERT INTO PERSON VALUES('A04','Venu','N R Colony');
INSERT INTO PERSON VALUES('A05','Jhon','Hanumanth nagar');
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');
INSERT INTO ACCIDENT VALUES(11,'2003-01-01','Mysore Road');
INSERT INTO ACCIDENT VALUES(12,'2004-02-02','South end Circle');
INSERT INTO ACCIDENT VALUES(13,'2003-01-21','Bull temple Road');
INSERT INTO ACCIDENT VALUES(14,'2008-02-17','Mysore Road');
INSERT INTO ACCIDENT VALUES(15,'2005-03-04','Kanakpura Road');
INSERT INTO OWNS VALUES('A01','KA052250');
INSERT INTO OWNS VALUES('A02','KA053408');
INSERT INTO OWNS VALUES('A03','KA031181');INSERT INTO OWNS VALUES('A04','KA095477');
INSERT INTO OWNS VALUES('A05','KA041702');
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 PERSON;
```

```
$sqlite3 database.sdb < main.sql
A01|Richard|Srinivas nagar
A02|Pradeep|Rajaji nagar
A03|Smith|Ashok nagar
A04|Venu|N R Colony
A05|Jhon|Hanumanth nagar
```

SELECT * FROM CAR;

```
$sqlite3 database.sdb < main.sql
KA052250|Indica|1990
KA031181|Lancer|1957
KA095477|Toyota|1998
KA053408|Honda|2008
KA041702|Audi|2005
```

SELECT * FROM ACCIDENT;

```
$sqlite3 database.sdb < main.sql
11|2003-01-01|Mysore Road
12|2004-02-02|South end Circle
13|2003-01-21|Bull temple Road
14|2008-02-17|Mysore Road
15|2005-03-04|Kanakpura Road
```

SELECT * FROM OWNS;

```
$sqlite3 database.sdb < main.sql
A01|KA052250
A02|KA053408
A03|KA031181
A04|KA095477
A05|KA041702
```

SELECT * FROM PARTICIPATED;

```
$sqlite3 database.sdb < main.sql
A01|KA052250|11|10000
A02|KA053408|12|50000
A03|KA095477|13|25000
A04|KA031181|14|3000
A05|KA041702|15|5000
```

UPDATE PARTICIPATED SET damage_amount=25000 WHERE REPORT_NO =12 AND REGNO='KA053408';

```
A01|KA052250|11|10000
A02|KA053408|12|25000
A03|KA095477|13|25000
A04|KA031181|14|3000
A05|KA041702|15|5000

[Program exited with exit code 0]
```

INSERT INTO ACCIDENT VALUES(16,'2018-04-08','MYSORE');

```
11|2003-01-01|Mysore Road
12|2004-02-02|South end Circle
13|2003-01-21|Bull temple Road
14|2008-02-17|Mysore Road
15|2005-03-04|Kanakpura Road
16|2018-04-08|MYSORE

[Program exited with exit code 0]
```

SELECT COUNT(*) FROM ACCIDENT WHERE DATE LIKE '2008-__-__';

```
1

[Program exited with exit code 0]
```

SELECT COUNT(*) FROM CAR WHERE model LIKE 'Indica';

```
1

[Program exited with exit code 0]
```

DBMS LAB 2 :-

```
CREATE TABLE branch ( branch_name VARCHAR(20), branch_city VARCHAR(20), assets REAL,
PRIMARY KEY(branch_name)
);
CREATE TABLE accounts
( acc_no INT,
branch_name VARCHAR(50), balance
REAL,
PRIMARY KEY(acc_no),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE customer (
customer_name VARCHAR(20),
customer_street VARCHAR(50),
customer_city VARCHAR(20),
PRIMARY KEY(customer_name)
);
CREATE TABLE depositor (
customer_name VARCHAR(20),
acc_no INT,
PRIMARY KEY(customer_name, acc_no),
FOREIGN KEY(customer_name) REFERENCES customer(customer_name)ON UPDATE CASCADE ON
DELETE CASCADE,
FOREIGN KEY(acc_no) REFERENCES accounts(acc_no)
ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE loan (
loan_number INT,
```

```
branch_name VARCHAR(50),
amount REAL,
PRIMARY KEY(loan_number),
FOREIGN KEY(branch_name) REFERENCES branch(branch_name)
ON UPDATE CASCADE ON DELETE CASCADE
);
INSERT INTO branch(branch_name,branch_city,assets) VALUES
('SBI_Chamrajpet','Bangalore',50000),('SBI_ResidencyRoad','Bangalore',10000),('SBI_ShivajiR
oad','Bombay',20000),('SBI_ParlimentRoad','Delhi',10000),('SBI_Jantarmantar','Delhi',20000)
;
INSERT INTO accounts(acc_no,branch_name,balance) VALUES
(1,'SBI_Chamrajpet',2000),(2,'SBI_ResidencyRoad',5000),(3,'SBI_ShivajiRoad',6000),(4,'SBI_P
arlimentRoad',9000),(5,'SBI_Jantarmantar',8000),(6,'SBI_ShivajiRoad',4000),(8,'SBI_Residenc
yRoad',4000),(9,'SBI_ParlimentRoad',3000),(10,'SBI_ResidencyRoad',5000),(11,'SBI_Jantarm
antar',2000);
INSERT INTO customer(customer_name,customer_street,customer_city) VALUES
('Avinash','Bull_Temple_Road','Bangalore'),('Dinesh','Bannergatta_Road','Bangalore'),('Moh
an','NationalCollege_Road','Bangalore'),('Nikil','Akbar_Road','Delhi'),('Ravi','Prithviraj_Road', 'Delhi');
INSERT INTO depositor(customer_name,acc_no) VALUES
('Avinash',1),('Dinesh',2),('Nikil',4),('Ravi',5),('Avinash',8),('Nikil',9),('Dinesh',10),('Nikil',11);INSERT
INTO loan(loan_number,branch_name,amount) VALUES
(1,'SBI_Chamrajpet',1000),(2,'SBI_ResidencyRoad',2000),(3,'SBI_ShivajiRoad',3000),(4,'SBI_P
arlimentRoad',4000),(5,'SBI_Jantarmantar',5000);
```

SELECT * FROM branch;

```
$sqlite3 database.sdb < main.sql
SBI_Chamrajpet|Bangalore|50000.0
SBI_ResidencyRoad|Bangalore|10000.0
SBI_ShivajiRoad|Bombay|20000.0
SBI_ParlimentRoad|Delhi|10000.0
SBI_Jantarmantar|Delhi|20000.0
```

SELECT * FROM accounts;

```
$sqlite3 database.sdb < main.sql
1|SBI_Chamrajpet|2000.0
2|SBI_ResidencyRoad|5000.0
3|SBI_ShivajiRoad|6000.0
4|SBI_ParlimentRoad|9000.0
5|SBI_Jantarmantar|8000.0
6|SBI_ShivajiRoad|4000.0
8|SBI_ResidencyRoad|4000.0
9|SBI_ParlimentRoad|3000.0
10|SBI_ResidencyRoad|5000.0
11|SBI_Jantarmantar|2000.0
```

SELECT * FROM customer;

```
$sqlite3 database.sdb < main.sql
Avinash|Bull_Temple_Road|Bangalore
Dinesh|Bannergatta_Road|Bangalore
Mohan|NationalCollege_Road|Bangalore
Nikil|Akbar_Road|Delhi
Ravi|Prithviraj_Road|Delhi
```

SELECT * FROM depositor;

```
$sqlite3 database.sdb < main.sql
Avinash|1
Dinesh|2
Nikil|4
Ravi|5
Avinash|8
Nikil|9
Dinesh|10
Nikil|11
```

SELECT * FROM loan;

```
$sqlite3 database.sdb < main.sql
1|SBI_Chamrajpet|1000.0
2|SBI_ResidencyRoad|2000.0
3|SBI_ShivajiRoad|3000.0
4|SBI_ParlimentRoad|4000.0
5|SBI_Jantarmantar|5000.0
```

SELECT d.customer_name FROM accounts a, depositor d WHERE d.acc_no=a.acc_no AND a.branch_name="SBI_ResidencyRoad" GROUP BY d.customer_name having count(d.customer_name>=2);

```
Avinash
Dinesh

[Program exited with exit code 0]
```

SELECT d.customer_name FROM accounts a, depositor d,branch b WHERE d.acc_no=a.acc_no AND b.branch_name=a.branch_name AND b.branch_city="Delhi" GROUP BY d.customer_name having count(distinct b.branch_name)=(SELECT COUNT(branch_name) FROM branch WHERE branch_city="Delhi");

```
Nikil

[Program exited with exit code 0]
```

DELETE FROM ACCOUNTS WHERE branch_name IN(SELECT branch_name FROM BRANCH WHERE branch_city='Bombay');

```
1|SBI_Chamrajpet|2000.0
2|SBI_ResidencyRoad|5000.0
4|SBI_ParlimentRoad|9000.0
5|SBI_Jantarmantar|8000.0
8|SBI_ResidencyRoad|4000.0
9|SBI_ParlimentRoad|3000.0
10|SBI_ResidencyRoad|5000.0
11|SBI_Jantarmantar|2000.0

[Program exited with exit code 0]
```

DBMS LAB 3 :-

```
CREATE TABLE suppliers(
 sid INT,
 sname VARCHAR(20),
address VARCHAR(50),
 PRIMARY KEY (sid)
);
CREATE TABLE parts(
 pid INT,
 pname VARCHAR(20),
color VARCHAR(10),
 PRIMARY KEY (pid)
);
CREATE TABLE catalog(
 sid INT,
pid INT,
cost REAL,
 PRIMARY KEY(sid,pid),
 FOREIGN KEY(sid) REFERENCES suppliers(sid)
 ON delete CASCADE ON update CASCADE,
 FOREIGN KEY(pid) REFERENCES parts(pid)
 ON delete CASCADE ON update CASCADE
);
insert into suppliers values (10001,'Acme Widget','Bangalore'), (10002,'Johns','Kolkata'),
(10003,'Vimal','Mumbai'),(10004,'Reliance','Delhi'); insert
into parts values
(20001,'Book','Red'),(20002,'Pen','Red'),(20003,'Pencil','Green'),(20004,'Mobile','Green'),(20
005,'Charger','Black');insert into catalog
values(10001,20001,10),(10001,20002,10),(10001,20003,30),(10001,20004,10),(10001,2000
5,10),(10002,20001,10),(10002,20002,20),(10003,20003,30),(10004,20003,40);
```

SELECT * FROM suppliers;

```
10001|Acme Widget|Bangalore
10002|Johns|Kolkata
10003|Vimal|Mumbai
10004|Reliance|Delhi

[Program exited with exit code 0]
```

SELECT * FROM parts;

```
20001|Book|Red
20002|Pen|Red
20003|Pencil|Green
20004|Mobile|Green
20005|Charger|Black

[Program exited with exit code 0]
```

SELECT * FROM catalog;

```
10001|20001|10.0
10001|20002|10.0
10001|20003|30.0
10001|20004|10.0
10001|20005|10.0
10002|20001|10.0
10002|20002|20.0
10003|20003|30.0
10004|20003|40.0

[Program exited with exit code 0]
```

SELECT DISTINCT p.pname FROM parts p, catalog c WHERE p.pid = c.pid;

```
Book
Pen
Pencil
Mobile
Charger

[Program exited with exit code 0]
```

SELECT s.sname FROM suppliers s WHERE NOT EXISTS(SELECT p.pid FROM parts p EXCEPT SELECT c.pid FROM catalog c WHERE c.sid = s.sid);

```
Acme Widget

[Program exited with exit code 0]
```

SELECT s.sname FROM suppliers 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');

```
Acme Widget
Johns

[Program exited with exit code 0]
```

SELECT p.pname FROM parts p, catalog c, suppliers s WHERE p.pid = c.pid AND c.sid = s.sid AND s.sname = 'Acme Widget' AND NOT EXISTS (SELECT * FROM catalog c1, suppliers s1 WHERE p.pid = c1.pid AND c1.sid = s1.sid AND s1.sname <> 'Acme Widget');

```
Mobile
Charger

[Program exited with exit code 0]
```

SELECT DISTINCT c.sid FROM catalog c WHERE c.cost > (SELECT AVG(C1.cost) FROM catalog c1 WHERE c1.pid = c.pid) ;

```
10002
10004

[Program exited with exit code 0]
```

SELECT p.pid, s.sname FROM parts p, suppliers 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);

```
20001|Acme Widget
20004|Acme Widget
20005|Acme Widget
20001|Johns
20002|Johns
20003|Reliance

[Program exited with exit code 0]
```

DBMS LAB 4 :-

```
CREATE TABLE student(  snum INT,  sname VARCHAR(20),  major VARCHAR(10),  lvl VARCHAR(2),  age
INT,
 PRIMARY KEY(snum)
);
CREATE TABLE
class(  cname
VARCHAR(20),
meets_at
TIME(0),  room
VARCHAR(10),
 fid INT,
 PRIMARY KEY(cname),
 FOREIGN KEY(fid) REFERENCES faculty(fid)
 ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE
enrolled(
snum INT,
cname
VARCHAR(20),
 PRIMARY KEY(snum,cname),
 FOREIGN KEY(snum) REFERENCES student(snum)
 ON UPDATE CASCADE ON DELETE CASCADE,
 FOREIGN KEY(cname) REFERENCES class(cname)
 ON UPDATE CASCADE ON DELETE CASCADE
);CREATE TABLE faculty(
 fid INT,
 fname VARCHAR(20),
```

```
 deptid INT,
 PRIMARY KEY(fid)
);
INSERT INTO student (snum,sname,major,lvl,age) VALUES
(1, 'hari', 'MATH', 'FR', 18),
(2, 'mahantesh', 'MATH', 'SR', 18),
(3, 'shreshtha', 'TFCS', 'SR', 19),
(4, 'subhas', 'TFCS', 'FR', 19),
(5, 'saquib', 'DBMS', 'JR', 20),
(6, 'krishan', 'DBMS', 'JR', 21),
(7, 'jaques', 'ADA', 'SR', 21),
(8, 'ashutosh' , 'MATH', 'FR', 18),
(9, 'divyanshu' , 'MATH', 'JR', 21),
(10, 'derek' , 'MATH', 'SR', 19);
INSERT INTO class (cname,meets_at,room,fid) VALUES
('A', '01:02:00',
'R124', 1), ('B',
'02:02:00', 'R125',
2), ('C', '03:02:00',
'R126', 3),
('D', '03:02:00',
'R125', 4), ('G',
'06:02:00', 'R126', 4),
('H', '01:02:00',
'R127', 4),
('F', '05:02:00', 'R124', 4),
('E', '04:02:00', 'R128', 4);INSERT INTO enrolled (snum,cname) VALUES
(1, 'A'),
(
2
,
```

'

B

'

)

,

(

3

,

'

C

'

)

,

(3, 'D'),

(4, 'D'),

(5, 'E'),

(6, 'A'),

(7, 'B');

INSERT INTO faculty (fid,fname,deptid) VALUES

(1, 'RAM', 1),

(2, 'SHYAM', 2),

(3, 'TOM', 3),

(4, 'DOM', 4);

SELECT * FROM student;

```
1|hari|MATH|FR|18
2|mahantesh|MATH|SR|18
3|shreshtha|TFCS|SR|19
4|subhas|TFCS|FR|19
5|saquib|DBMS|JR|20
6|krishan|DBMS|JR|21
7|jaques|ADA|SR|21
8|ashutosh|MATH|FR|18
9|divyanshu|MATH|JR|21
10|derek|MATH|SR|19

[Program exited with exit code 0]
```

SELECT * FROM class;

```
A|01:02:00|R124|1
B|02:02:00|R125|2
C|03:02:00|R126|3
D|03:02:00|R125|4
G|06:02:00|R126|4
H|01:02:00|R127|4
F|05:02:00|R124|4
E|04:02:00|R128|4

[Program exited with exit code 0]
```

SELECT * FROM enrolled;

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

[Program exited with exit code 0]
```

SELECT * FROM faculty;

```
1|RAM|1
2|SHYAM|2
3|TOM|3
4|DOM|4

[Program exited with exit code 0]
```

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by RAM. 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 = 'RAM' AND s.lvl = 'JR';

```
krishan

[Program exited with exit code 0]
```

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 (e.snum)>4);

```
E

[Program exited with exit code 0]
```

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.meets_at = c2.meets_at);

```
shreshtha

[Program exited with exit code 0]
```

iv. Find the names of faculty members who teach in every room in which some class is taught. SELECT DISTINCT f.fname FROM faculty f WHERE NOT EXISTS(SELECT c.room FROM class c EXCEPT SELECT c1.room FROM class c1 WHERE c1.fid = f.fid);

```
DOM

[Program exited with exit code 0]
```

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);

```
RAM
SHYAM
TOM
DOM

[Program exited with exit code 0]
```

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);

```
ashutosh
divyanshu
derek

[Program exited with exit code 0]
```

vii. For each age value that appears in Students, find the level value that appears most
often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR). SELECT s.age, s.lvl FROM student s GROUP BY s.age, s.lvl HAVING s.lvl IN (SELECT s1.lvl FROM student s1 WHERE s1.age=s.age GROUP BY s1.age, s1.lvl HAVING COUNT(*) >= ALL (SELECT COUNT(*)
FROM student s2 WHERE s1.age=s2.age GROUP BY s2.lvl, s2.age)) ORDER BY s.age;

	age	lvl
1	18	FR
2	19	SR
3	20	JR
4	21	JR

DBMS LAB 5 :-

```
CREATE TABLE flight(
'flno' INT,
'from' VARCHAR(20),
'to' VARCHAR(20),
'distance' INT,
'departs' VARCHAR(20),
'arrives' VARCHAR(20),
'price' REAL,
PRIMARY KEY('flno'));
CREATE TABLE aircraft(
'aid' INT,
'aname' VARCHAR(20),
'cruisingrange' INT,
PRIMARY KEY('aid'));
CREATE TABLE employees(
'eid' INT,
'ename' VARCHAR(20),
'salary' INT,
PRIMARY KEY('eid'));
CREATE TABLE certified(
'eid' INT, 'aid'
INT,
PRIMARY KEY('eid','aid'),
FOREIGN KEY ('eid') REFERENCES employees('eid')
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY ('aid') REFERENCES aircraft('aid')ON DELETE CASCADE ON UPDATE CASCADE);
INSERT INTO flight VALUES
(1,'Mumbai','Bangaluru',360,'10:45:00','12:00:00',10000),
(2,'Bangaluru','Delhi',1700,'12:15:00','04:30:00',5000),
```

```
(3,'Mumbai','Sydney',3500,'02:15:00','05:25:00',30000),
(4,'Delhi','Mumbai',1300,'10:15:00','12:05:00',4500),
(5,'Delhi','Frankfurt',18000,'07:15:00','05:30:00',90000),
(6,'Bangaluru','Frankfurt',19500,'10:00:00','07:45:00',95000),
(7,'Bengaluru','Frankfurt',17000,'12:00:00','06:30:00',99000),
(8,'Madison','Chicago',236,'12:00:00','13:00:00',5000),
(9,'Chicago','New York',1281,'15:00:00','17:30:00',15000),
(10,'Madison','New York',1514,'13:00:00','16:30:00',25000);
INSERT INTO aircraft VALUES
(111,'Airbus',1000),
(222,'Boeing',5000),
(333,'Jet01',5000),
(444,'Airbus380',8000),
(559,'Boeing747',500),
(880,'AirbusA310',800),
(909,'Concorde',1000);
INSERT INTO employees VALUES
(1,'Santino',30000),
(2,'Ajith',85000),
(3,'Krishan',50000),
(4,'Joseph',45000),
(5,'Abhimanyu',90000),
(6,'Ryan',75000),(7,'Ram',100000);
INSERT INTO certified VALUES
(1,111),
(2,111),
(1,222),
(5,222),
(7,222),
(1,333),
(2,333),
```

(1,444),

(2,444),

(4,444),

(6,559),

(3,559),

(5,880),

(6,880),

(3,909),

(1,909),

(1,880);

SELECT * FROM flight;

```
1|Mumbai|Bangaluru|360|10:45:00|12:00:00|10000.0
2|Bengaluru|Delhi|1700|12:15:00|04:30:00|5000.0
3|Mumbai|Sydney|3500|02:15:00|05:25:00|30000.0
4|Delhi|Mumbai|1300|10:15:00|12:05:00|4500.0
5|Delhi|Frankfurt|18000|07:15:00|05:30:00|90000.0
6|Bangaluru|Frankfurt|19500|10:00:00|07:45:00|95000.0
7|Bengaluru|Frankfurt|17000|12:00:00|06:30:00|99000.0
8|Madison|Chicago|236|12:00:00|13:00:00|5000.0
9|Chicago|New York|1281|15:00:00|17:30:00|15000.0
10|Madison|New York|1514|13:00:00|16:30:00|25000.0

[Program exited with exit code 0]
```

SELECT * FROM aircraft;

```
111|Airbus|1000
222|Boeing|5000
333|Jet01|5000
444|Airbus380|8000
559|Boeing747|500
880|AirbusA310|800
909|Concorde|1000

[Program exited with exit code 0]
```

SELECT * FROM employees;

```
1|Santino|30000
2|Ajith|85000
3|Krishan|50000
4|Joseph|45000
5|Abhimanyu|90000
6|Ryan|75000
7|Ram|100000

[Program exited with exit code 0]
```

SELECT * FROM certified;

```
1|111
1|222
1|333
1|444
1|880
1|909
2|111
2|333
2|444
3|559
3|909
4|444
5|222
5|880
6|559
6|880
7|222

[Program exited with exit code 0]
```

• 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 certified c, employees e, aircraft a WHERE c.eid=e.eid AND c.aid=a.aid AND e.salary>80000;

```
Airbus
Jet01
Airbus380
Boeing
AirbusA310

[Program exited with exit code 0]
```

- 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 c.eid, MAX(a.cruisingrange) FROM certified c, aircraft a WHERE c.aid = a.aid GROUP BY c.eid HAVING COUNT(a.aname)>3;

```
1|8000

[Program exited with exit code 0]
```

- Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt. SELECT DISTINCT e.ename FROM employees e WHERE e.salary

```
Santino
Ajith
Krishan
Joseph
Abhimanyu
Ryan

[Program exited with exit code 0]
```

- For all aircraft with cruisingrange over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft. SELECT a.aname,AVG(e.salary) FROM aircraft a,certified c,employees e WHERE a.aid=c.aid AND c.eid=e.eid AND a.cruisingrange>1000 GROUP BY a.aname;

```
Airbus380|53333.3333333333
Boeing|73333.3333333333
Jet01|57500.0

[Program exited with exit code 0]
```

- Find the names of pilots certified for some Boeing aircraft. SELECT DISTINCT e.ename FROM employees e,aircraft a,certified c WHERE e.eid=c.eid AND c.aid=a.aid AND a.aname like 'Boeing%';

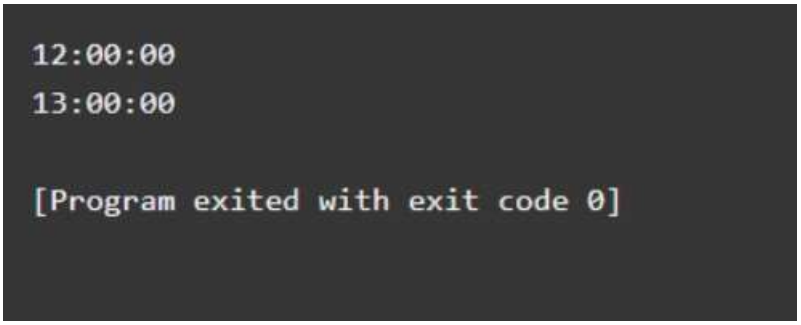
```
Santino
Abhimanyu
Ram
Krishan
Ryan

[Program exited with exit code 0]
```

- Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi. SELECT a.aid FROM aircraft a WHERE a.cruisingrange>(SELECT MIN(f.distance) FROM flight f WHERE f.'from'='Bangaluru' AND f.'to'='Delhi');

```
222
333
444

[Program exited with exit code 0]
```

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

```
SELECT f.departs FROM flight f WHERE f.'from'= 'Madison' AND f.'to'= 'New York' AND time(f.arrives)<'18:00:00'
UNION SELECT f1.departs FROM flight f1 INNER JOIN flight f2 ON f1.'to' = f2.'from' WHERE f1.'from'= 'Madison' AND f2.'to' = 'New York' AND time(f1.arrives) < time(f2.departs) AND time(f2.arrives) < '18:00:00'
UNION
SELECT f3.departs FROM flight f3 INNER JOIN flight f4 ON f3.'to' = f4.'from' INNER JOIN flight f5 ON f4.'to' = f5.'from' WHERE f3.'from'= 'Madison' AND f5.'to' = 'New York' AND time(f3.arrives) < time(f4.departs) AND time(f4.arrives) < time(f5.departs) AND time(f5.arrives) < '18:00:00';
```



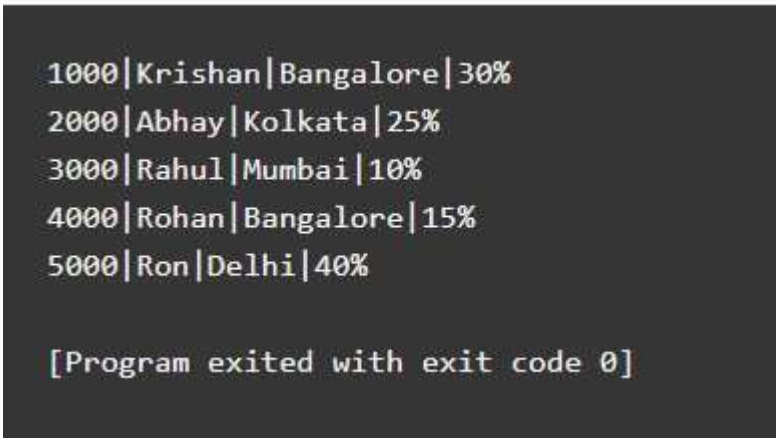
Lab 6 :-

```
CREATE TABLE salesman (salesman_id INT, name VARCHAR(20), city VARCHAR(20), commission

VARCHAR(20),

PRIMARY KEY (salesman_id));


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 UPDATE CASCADE ON DELETE CASCADE);


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 UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)  ON UPDATE CASCADE ON DELETE
CASCADE);



INSERT INTO SALESMAN VALUES(1000,'Krishan','Bangalore','30%'),

(2000,'Abhay','Kolkata','25%'),

(3000,'Rahul','Mumbai','10%'),

(4000,'Rohan','Bangalore','15%'),

(5000,'Ron','Delhi','40%');
```

```
INSERT INTO CUSTOMER VALUES(10,'Sian','Bangalore',5,4000),
(11,'Juan','Bangalore',4,4000),
(12,'Rial','Hyderabad',5,2000),
(13,'Genic','Ahemdabad',1,1000),
(14,'Vron','Delhi',3,3000);
```

```
INSERT INTO ORDERS VALUES(101,10000,'2021-06-12',10,4000),
(102,15000,'2018-05-07',10,2000),
(103,8000,'2020-11-01',14,3000),
(104,800,'2015-10-25',13,1000),
(105,100,'2020-11-01',11,2000);
```

```
SELECT * FROM salesman;
```




```
SELECT * FROM customer;
```

```
10|Sian|Bangalore|5|4000
11|Juan|Bangalore|4|4000
12|Rial|Hyderabad|5|2000
13|Genic|Ahemdabad|1|1000
14|Vron|Delhi|3|3000

[Program exited with exit code 0]
```

SELECT * FROM orders;

```
101|10000|2021-06-12|10|4000
102|15000|2018-05-07|10|2000
103|8000|2020-11-01|14|3000
104|800|2015-10-25|13|1000
105|100|2020-11-01|11|2000

[Program exited with exit code 0]
```

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');

```
5|2

[Program exited with exit code 0]
```

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

SELECT s.salesman_id, s.name FROM salesman s, customer c WHERE c.salesman_id=s.salesman_id GROUP BY s.salesman_id HAVING COUNT(s.salesman_id)>1;

```
4000|Rohan

[Program exited with exit code 0]
```

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

SELECT s.salesman_id, s.name, c.cust_name, s.commission FROM salesman s, customer c WHERE s.city = c.city

UNION

SELECT salesman_id, name, 'NO CUSTOMER', commission FROM salesman WHERE city NOT IN (SELECT city FROM customer);

```
1000|Krishan|Juan|30%
1000|Krishan|Sian|30%
2000|Abhay|NO CUSTOMER|25%
3000|Rahul|NO CUSTOMER|10%
4000|Rohan|Juan|15%
4000|Rohan|Sian|15%
5000|Ron|Vron|40%

[Program exited with exit code 0]
```

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

CREATE VIEW high_order AS

SELECT o.ord_date, s.salesman_id, s.name FROM salesman s, orders o WHERE s.salesman_id = o.salesman_id

AND o.purchase_amt = (SELECT MAX (purchase_amt) FROM orders o1 WHERE o1.ord_date = o.ord_date);

SELECT * FROM high_order;

```
2021-06-12|4000|Rohan
2018-05-07|2000|Abhay
2020-11-01|3000|Rahul
2015-10-25|1000|Krishan

[Program exited with exit code 0]
```

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 orders;

ord_no	purchase_amt	ord_date	customer_id	salesman_id
101	10000	2021-06-12	10	4000
102	15000	2018-05-07	10	2000
103	8000	2020-11-01	14	3000
105	100	2020-11-01	11	2000
NULL	NULL	NULL	NULL	NULL

Lab 7 :-

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

```
CREATE TABLE author( author_id INT,
author_name     VARCHAR(20),
author_city       VARCHAR(20),
author_country VARCHAR(20),
PRIMARY KEY(author_id));

CREATE TABLE publisher( publisher_id
INT,
publisher_name     VARCHAR(20),
publisher_city       VARCHAR(20),
publisher_country VARCHAR(20),
PRIMARY KEY(publisher_id));

CREATE TABLE category(
category_id INT, description
VARCHAR(20),
PRIMARY KEY(category_id));

CREATE TABLE catalog(
book_id INT, book_title
VARCHAR(30), author_id
INT, publisher_id INT,
category_id INT, year
INT, price INT,
PRIMARY KEY(book_id),
FOREIGN KEY(author_id) REFERENCES author(author_id)
ON UPDATE CASCADE ON DELETE CASCADE,
```

```
FOREIGN KEY(publisher_id) REFERENCES publisher(publisher_id)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY(category_id) REFERENCES category(category_id)
ON UPDATE CASCADE ON DELETE CASCADE);

CREATE TABLE orderdetails(
order_id INT, book_id INT,
quantity INT,
PRIMARY KEY(order_id),
FOREIGN KEY(book_id) REFERENCES catalog(book_id)
ON UPDATE CASCADE ON DELETE CASCADE);
```

ii. Enter at least five tuples for each relation.

```
INSERT INTO author VALUES
(101,'JK Rowling','London','England'),
(102,'William Shakespeare','Stratford-upon-Avon','England'),
(103,'Chetan Bhagat','Mumbai','India'),
(104,'Robert Brown','California','USA'),
(105,'Juan Gen','Sydbey','Australia');

INSERT INTO publisher VALUES
(10,'Bloomsbury','London','England'),
(11,'Scholastic','Washington','USA'),
(12,'Pearson','Manchester','England'),
(13,'Geetanjali','Delhi','India'),
(14,'Saraswati','Mumbai','India');

INSERT INTO category VALUES
(51,'fantasy'),
(52,'horror'),
(53,'thriller'),
```

(54,'action'),

(55,'fiction');

INSERT INTO catalog VALUES

(1,'HP and the Half Blood Prince',101,10,51,2005,1000),

(2,'HP and the Order Of the Phoenix',101,10,51,2005,950),

(3,'First Folio',102,12,55,1623,1000),

(4,'3 Mistakes of my life',103,14,55,2007,800),

(5,'Get Trapped',104,11,53,2004,750),

(6,'Fight at Will',105,13,54,2000,500),

(7,'Intelligence of Demons',105,11,52,2005,1000) ;

INSERT INTO orderdetails VALUES

(71,1,60),

(72,2,55),

(73,3,40),

(74,4,10),

(75,5,50),

(76,6,70),

(77,7,20), (78,7,60);

SELECT * FROM

author;

```
101|JK Rowling|London|England
102|William Shakespeare|Stratford-upon-Avon|England
103|Chetan Bhagat|Mumbai|India
104|Robert Brown|California|USA
105|Juan Gen|Sydbey|Australia

[Program exited with exit code 0]
```

SELECT * FROM publisher;

```
10|Bloomsbury|London|England
11|Scholastic|Washington|USA
12|Pearson|Manchester|England
13|Geetanjali|Delhi|India
14|Saraswati|Mumbai|India

[Program exited with exit code 0]
```

SELECT * FROM category;

```
51|fantasy
52|horror
53|thriller
54|action
55|fiction

[Program exited with exit code 0]
```

SELECT * FROM catalog;

```
1|HP and the Half Blood Prince|101|10|51|2005|1000
2|HP and the Order Of the Phoenix|101|10|51|2005|950
3|First Folio|102|12|55|1623|1000
4|3 Mistakes of my life|103|14|55|2007|800
5|Get Trapped|104|11|53|2004|750
6|Fight at Will|105|13|54|2000|500
7|Intelligence of Demons|105|11|52|2005|1000

[Program exited with exit code 0]
```

SELECT * FROM orderdetails;

```
71|1|60
72|2|55
73|3|40
74|4|10
75|5|50
76|6|70
77|7|20
78|7|60

[Program exited with exit code 0]
```

iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.

SELECT a.author_id,author_name,author_city,author_country FROM author a,catalog c WHERE a.author_id=c.author_id AND c.year>2000 GROUP BY c.author_id HAVING COUNT(c.author_id)>=2;

```
101|JK Rowling|London|England

[Program exited with exit code 0]
```

iv. Find the author of the book which has maximum sales.

SELECT author_name FROM author a,catalog c WHERE a.author_id=c.author_id AND c.book_id IN (SELECT o.book_id FROM orderdetails o WHERE quantity=(SELECT MAX(quantity) FROM orderdetails));

```
Juan Gen

[Program exited with exit code 0]
```

v. Demonstrate how you increase the price of books published by a specific publisher by 10%.

UPDATE catalog SET price=1.1*price WHERE publisher_id IN

(SELECT publisher_id FROM publisher WHERE publisher_name='Scholastic'); SELECT

* FROM catalog;

```
1|HP and the Half Blood Prince|101|10|51|2005|1000
2|HP and the Order Of the Phoenix|101|10|51|2005|950
3|First Folio|102|12|55|1623|1000
4|3 Mistakes of my life|103|14|55|2007|800
5|Get Trapped|104|11|53|2004|825.0
6|Fight at Will|105|13|54|2000|500
7|Intelligence of Demons|105|11|52|2005|1100

[Program exited with exit code 0]
```

Lab 8 :-

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

```
CREATE TABLE student( regno VARCHAR(20), name VARCHAR(20), major VARCHAR(20), bdate DATE,
PRIMARY KEY(regno));

CREATE TABLE course( course
INT,
cname VARCHAR(20), dept
VARCHAR(20),
PRIMARY KEY(course));

CREATE TABLE enroll(
regno   VARCHAR(20),
cname VARCHAR(20),
sem INT, marks
INT,
PRIMARY KEY (regno,cname),
FOREIGN KEY (regno) REFERENCES student (regno)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (cname) REFERENCES course (cname)
ON UPDATE CASCADE ON DELETE CASCADE);

CREATE TABLE text(
book_isbn INT, book_title
VARCHAR(20), publisher
VARCHAR(20), author
VARCHAR(20),
PRIMARY KEY(book_isbn));

CREATE TABLE book_adoption(
```

```
course INT, sem
INT, book_isbn
INT,
PRIMARY KEY(course,book_isbn),
FOREIGN KEY (course) REFERENCES course (course)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (book_isbn) REFERENCES text(book_isbn)
ON UPDATE CASCADE ON DELETE CASCADE);
```

ii. Enter at least five tuples for each relation.

```
INSERT INTO student VALUES
('1BM19CS001','a','maths','1999-09-11'),
('1bm19cs002','b','physics','1998-07-21'),
('1bm19cs003','c','maths','2000-11-30'),
('1BM19CS004','d','maths','2001-12-01'),
('1BM19CS005','e','chemistry','1998-03-06');

INSERT INTO course VALUES
(111,'OS','CSE'),
(112,'JAVA','CSE'),
(113,'LOD','ISE'),
(114,'DBMS','CSE'),
(115,'IOT','ECE');

INSERT INTO enroll VALUES
('1BM19CS001','OS',4,100),
('1BM19CS002','DBMS',3,80),
('1BM19CS003','LOD',5,100),
('1BM19CS004','OS',4,40),
('1BM19CS005','JAVA',3,90);
```

```
INSERT INTO text VALUES
(10,'DATABASE SYSTEMS','PEARSON','SONAM'),
(11,'OPERATING SYSTEM','PEARSON','JUAN'),
(12,'OIJ','HEAL','RON'),
(13,'CIRCUIT DESIGNS','MCGARW','JACOB'),
(14,'SCHEDULING','PEARSON','PATIL');

INSERT INTO book_adoption VALUES
(111,4,11),
(111,4,14),
(112,3,12),
(113,5,13),
(114,5,10),
(115,3,13);
```

```
SELECT * FROM student;
```

```
1BM19CS001|a|maths|1999-09-11
1bm19cs002|b|physics|1998-07-21
1bm19cs003|c|maths|2000-11-30
1BM19CS004|d|maths|2001-12-01
1BM19CS005|e|chemistry|1998-03-06

[Program exited with exit code 0]
```

```
SELECT * FROM course;
```

```
111|OS|CSE
112|JAVA|CSE
113|LOD|ISE
114|DBMS|CSE
115|IOT|ECE

[Program exited with exit code 0]
```

SELECT * FROM enroll;

```
1BM19CS001|OS|4|100
1BM19CS002|DBMS|3|80
1BM19CS003|LOD|5|100
1BM19CS004|OS|4|40
1BM19CS005|JAVA|3|90

[Program exited with exit code 0]
```

SELECT * FROM text;

```
10|DATABASE SYSTEMS|PEARSON|SONAM
11|OPERATING SYSTEM|PEARSON|JUAN
12|OIJ|HEAL|RON
13|CIRCUIT DESIGNS|MCGRAW|JACOB
14|SCHEDULING|PEARSON|PATIL

[Program exited with exit code 0]
```

SELECT * FROM book_adoption;

```
111|4|11
111|4|14
112|3|12
113|5|13
114|5|10
115|3|13

[Program exited with exit code 0]
```

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

Insert into text values (15,'COMPLETE REFERENCE TO OS','TATA MAC','HOBERT');

SELECT * FROM text;

```
10|DATABASE SYSTEMS|PEARSON|SONAM
11|OPERATING SYSTEM|PEARSON|JUAN
12|OIJ|HEAL|RON
13|CIRCUIT DESIGNS|MCGRAW|JACOB
14|SCHEDULING|PEARSON|PATIL
15|COMPLETE REFERENCE TO OS|TATA MAC|HOBERT

[Program exited with exit code 0]
```

INSERT INTO book_adoption VALUES (111,4,15);

SELECT * FROM book_adoption;

```
111|4|11
111|4|14
112|3|12
113|5|13
114|5|10
115|3|13
111|4|15

[Program exited with exit code 0]
```

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.course,t.book_isbn,t.book_title FROM course c,book_adoption b,text t
WHERE c.course=b.course AND b.book_isbn=t.book_isbn AND c.dept='CSE'
AND (SELECT COUNT(book_isbn) FROM book_adoption b1
WHERE c.course=b1.course)>2 ORDER BY t.book_title;
```

```
111|15|COMPLETE REFERENCE TO OS
111|11|OPERATING SYSTEM
111|14|SCHEDULING

[Program exited with exit code 0]
```

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,text t
WHERE c.course=b.course AND t.book_isbn=b.book_isbn AND t.publisher='MCGRAW')
AND c.dept NOT IN (SELECT c1.dept FROM course c1,book_adoption b1,text t1 WHERE
c1.course=b1.course
AND t1.book_isbn=b1.book_isbn AND t1.publisher != 'MCGRAW');
```

```
ISE
ECE

[Program exited with exit code 0]
```

Lab 9 :-

```
CREATE TABLE ACTOR(
ACT_ID INT,
ACT_NAME VARCHAR (20),
ACT_GENDER CHAR(1),
PRIMARY KEY (ACT_ID));

CREATE TABLE DIRECTOR( DIR_ID INT,
DIR_NAME VARCHAR(20),
DIR_PHONE INT,
PRIMARY KEY (DIR_ID));

CREATE TABLE MOVIES(
MOV_ID INT,
MOV_TITLE VARCHAR (50),
MOV_YEAR INT,
MOV_LANG VARCHAR (20),
DIR_ID INT,
PRIMARY KEY (MOV_ID),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));

CREATE TABLE MOVIE_CAST(
ACT_ID INT,
MOV_ID INT,
ROLE VARCHAR (20),
PRIMARY KEY (ACT_ID, MOV_ID),
FOREIGN KEY (ACT_ID) REFERENCES ACTOR (ACT_ID),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID)); CREATE TABLE RATING(
MOV_ID INT,
REV_STARS VARCHAR (20),
PRIMARY KEY (MOV_ID),
```

```
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));

INSERT INTO ACTOR VALUES (101,'HENA','F'),
(102,'RON','M'),
(103,'SAMA','F'),
(104,'JERMY','M'),
(105,'RIA','F');

INSERT INTO DIRECTOR VALUES (50,'IAN', 8751611001),
(51,'HITCHCOCK', 7766138911),
(52,'STEFF', 9986776531),
(53,'STEVEN SPIELBERG', 8989776530),
(54,'STANLEY KUBRICK', 8745963210);

INSERT INTO MOVIES VALUES (1001,'MINORITY REPORT', 2002,'ENGLISH',53),
(1002,'LIFE', 2016,'ENGLISH',50),
(1003,'HOW TO BE GOOD', 2008,'FRENCH',51), (1004,'WAR OF
WORLDS', 2005,'ENGLISH', 53),
(1005,'2002:SPACE ODDESSY',1968,'ENGLISH', 54);

INSERT INTO MOVIE_CAST VALUES (101, 1002,'HEROINE'),
(103, 1001,'HEROINE'),
(102, 1003,'HERO'),
(103, 1004,'GUEST'),
(105, 1005,'GUEST'),
(105, 1002,'GUEST'),
(104, 1004,'HERO');

INSERT INTO RATING VALUES (1001,4),
(1002,2),
(1003,3),
```

(1004,5),

(1005,4);

SELECT * FROM ACTOR;

```
101|HENA|F
102|RON|M
103|SAMA|F
104|JERMY|M
105|RIA|F

[Program exited with exit code 0]
```

SELECT * FROM DIRECTOR;

```
50|IAN|8751611001
51|HITCHCOCK|7766138911
52|STEFF|9986776531
53|STEVEN SPIELBERG|8989776530
54|STANLEY KUBRICK|8745963210

[Program exited with exit code 0]
```

SELECT * FROM MOVIES;

```
1001|MINORITY REPORT|2002|ENGLISH|53
1002|LIFE|2016|ENGLISH|50
1003|HOW TO BE GOOD|2008|FRENCH|51
1004|WAR OF WORLDS|2005|ENGLISH|53
1005|2002:SPACE ODDESSY|1968|ENGLISH|54

[Program exited with exit code 0]
```

SELECT * FROM MOVIE_CAST;

```
101|1002|HEROINE
103|1001|HEROINE
102|1003|HERO
103|1004|GUEST
105|1005|GUEST
105|1002|GUEST
104|1004|HERO

[Program exited with exit code 0]
```

SELECT * FROM RATING;

```
1001|4
1002|2
1003|3
1004|5
1005|4

[Program exited with exit code 0]
```

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

SELECT MOV_TITLE FROM MOVIES WHERE DIR_ID IN

(SELECT DIR_ID FROM DIRECTOR WHERE DIR_NAME = 'HITCHCOCK');

```
HOW TO BE GOOD

[Program exited with exit code 0]
```

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

SELECT MOV_TITLE FROM MOVIES M, MOVIE_CAST MV

WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN

(SELECT ACT_ID FROM MOVIE_CAST

GROUP BY ACT_ID HAVING COUNT (ACT_ID)>1);

```
MINORITY REPORT
WAR OF WORLDS
LIFE
2002:SPACE ODDESSY

[Program exited with exit code 0]
```

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

```
SELECT DISTINCT ACT_NAME FROM ACTOR A
JOIN MOVIE_CAST C
ON A.ACT_ID=C.ACT_ID
JOIN MOVIES M
ON C.MOV_ID=M.MOV_ID
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

```
HENA
RIA

[Program exited with exit code 0]
```

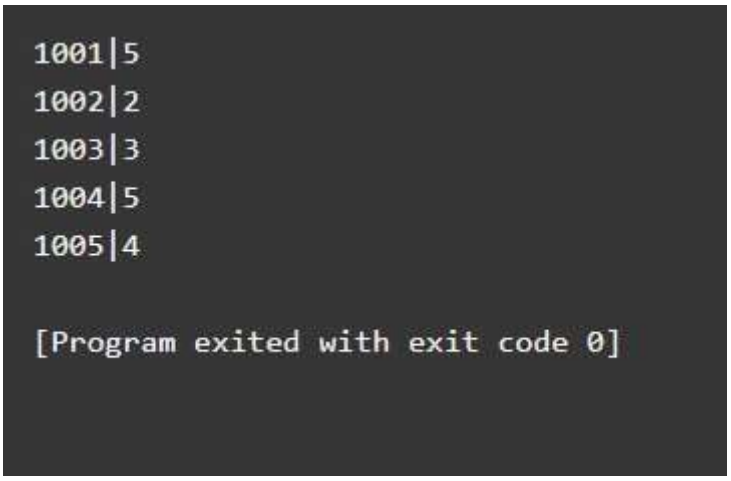
iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

```
SELECT MOV_TITLE, MAX (REV_STARS) FROM MOVIES
INNER JOIN RATING USING (MOV_ID)
GROUP BY MOV_TITLE HAVING MAX (REV_STARS)>0 ORDER BY MOV_TITLE;
```

```
2002:SPACE ODDESSY|4
HOW TO BE GOOD|3
LIFE|2
MINORITY REPORT|4
WAR OF WORLDS|5

[Program exited with exit code 0]
```

v. 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;
```



Lab 10 :-

```
CREATE TABLE STUDENT(
USN VARCHAR (10) PRIMARY KEY,
SNAME VARCHAR (25),
ADDRESS VARCHAR (25),
PHONE INT,
GENDER VARCHAR(1));

CREATE TABLE SEMSEC(
SSID VARCHAR(5) PRIMARY KEY,
SEM INT,
SEC VARCHAR(1));

CREATE TABLE CLASS(
USN VARCHAR (10),
SSID VARCHAR (10),
PRIMARY KEY (USN, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID)
ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE SUBJECT(
SUBCODE INT,
TITLE INT,
SEM INT,
CREDITS INT,
PRIMARY KEY (SUBCODE));

CREATE TABLE IAMARKS(
USN VARCHAR (10),
```

```
SUBCODE VARCHAR (8),
SSID VARCHAR (5),
TEST1 INT,
TEST2 INT,
TEST3 INT,
FINALIA INT,
PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID)
ON DELETE CASCADE ON UPDATE CASCADE);


INSERT INTO STUDENT VALUES ('1RN14CS025','ASMI','BENGALURU', 7894737377,'F'),
('1RN15CS011','AJAY','TUMKUR', 9845091341,'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','CHIKAMAGALUR',8800880011,'M');

INSERT INTO SEMSEC VALUES ('CSE5A', 5,'A');
INSERT INTO SEMSEC VALUES ('CSE5B', 5,'B');
INSERT INTO SEMSEC VALUES ('CSE5C', 5,'C');
INSERT INTO SEMSEC VALUES ('CSE4A', 4,'A');
INSERT INTO SEMSEC VALUES ('CSE4B', 4,'B');
INSERT INTO SEMSEC VALUES ('CSE4C', 4,'C');
INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
```

```
INSERT INTO SEMSEC VALUES ('CSE3B', 3,'B');
INSERT INTO SEMSEC VALUES ('CSE3C', 3,'C');
INSERT INTO SEMSEC VALUES ('CSE2A', 2,'A');
INSERT INTO SEMSEC VALUES ('CSE2B', 2,'B');
INSERT INTO SEMSEC VALUES ('CSE2C', 2,'C');

INSERT INTO CLASS VALUES ('1RN14CS025','CSE5B');
INSERT INTO CLASS VALUES ('1RN15CS011','CSE4A');
INSERT INTO CLASS VALUES ('1RN15CS029','CSE4A'); INSERT
INTO CLASS VALUES ('1RN15CS045','CSE4B'); INSERT INTO
CLASS VALUES ('1RN15CS091','CSE4C');
INSERT INTO CLASS VALUES ('1RN16CS045','CSE3A'); INSERT
INTO CLASS VALUES ('1RN16CS088','CSE3B'); INSERT INTO
CLASS VALUES ('1RN16CS122','CSE3C');

INSERT INTO SUBJECT VALUES ('15CS51', 'ME', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS52','CN', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS53','DBMS', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS54','ATC', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS41','M4', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS42','SE', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS43','DAA', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS31','M3', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS32','ADE', 3, 4);

INSERT INTO IAMARKS VALUES ('1RN15CS091','15CS41','CSE4C', 15, 16, 18, 34);
INSERT INTO IAMARKS VALUES ('1RN15CS091','15CS42','CSE4C', 12, 19, 14, 33); INSERT
INTO IAMARKS VALUES ('1RN15CS091','15CS43','CSE4C', 19, 15, 20, 39);
```

INSERT INTO IAMARKS VALUES ('1RN16CS122','15CS31','CSE3C', 20, 16, 19, 39); INSERT INTO IAMARKS VALUES ('1RN16CS122','15CS32','CSE3C', 15, 15, 12, 30);

SELECT * FROM STUDENT;

```
1RN14CS025|ASMI|BENGALURU|7894737377|F
1RN15CS011|AJAY|TUMKUR|9845091341|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|CHIKAMAGALUR|8800880011|M

[Program exited with exit code 0]
```

SELECT * FROM SEMSEC;

```
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

[Program exited with exit code 0]
```

SELECT * FROM CLASS;

```
1RN14CS025|CSE5B
1RN15CS011|CSE4A
1RN15CS029|CSE4A
1RN15CS045|CSE4B
1RN15CS091|CSE4C
1RN16CS045|CSE3A
1RN16CS088|CSE3B
1RN16CS122|CSE3C

[Program exited with exit code 0]
```

SELECT * FROM SUBJECT;

```
15CS51|ME|5|4
15CS52|CN|5|4
15CS53|DBMS|5|4
15CS54|ATC|5|4
15CS41|M4|4|4
15CS42|SE|4|4
15CS43|DAA|4|4
15CS31|M3|3|4
15CS32|ADE|3|4

[Program exited with exit code 0]
```

SELECT * FROM IAMARKS;

```
1RN15CS091|15CS41|CSE4C|15|16|18|34
1RN15CS091|15CS42|CSE4C|12|19|14|33
1RN15CS091|15CS43|CSE4C|19|15|20|39
1RN16CS122|15CS31|CSE3C|20|16|19|39
1RN16CS122|15CS32|CSE3C|15|15|12|30

[Program exited with exit code 0]
```

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';

```
1RN15CS091|SANTOSH|MANGALURU|8812332201|M|4|C

[Program exited with exit code 0]
```

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;

```
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
5|B|F|1

[Program exited with exit code 0]
```

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

```
CREATE VIEW STU_TEST1_MARKS_VIEW AS

SELECT TEST1, SUBCODE FROM IAMARKS

WHERE USN = '1RN15CS091';

SELECT * FROM STU_TEST1_MARKS_VIEW;
```

```
15|15CS41
12|15CS42
19|15CS43

[Program exited with exit code 0]
```