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),

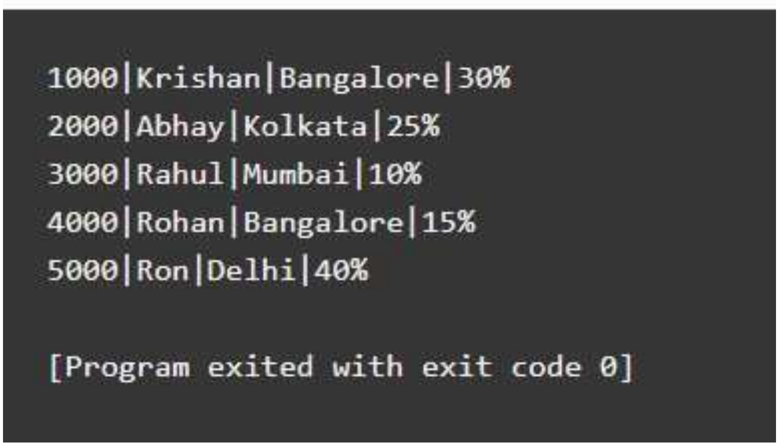
(102,15000,'2018-05-07',10,2000),

(103,8000,'2020-11-01',14,3000),

(104,800,'2015-10-25',13,1000),

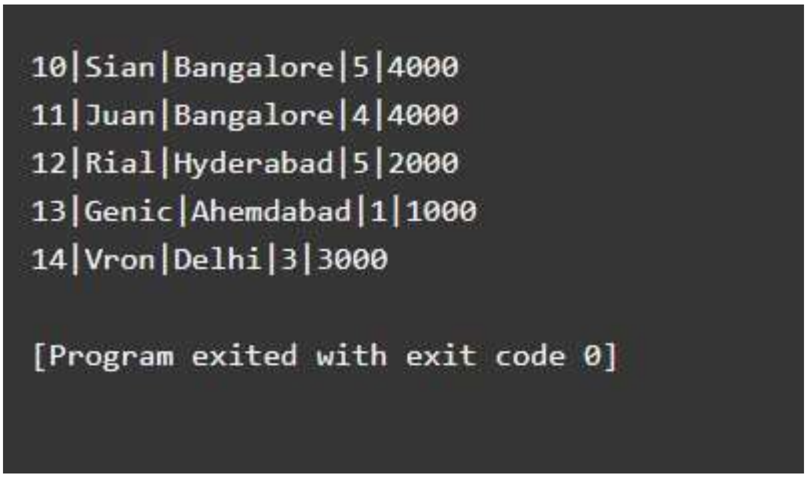
(105,100,'2020-11-01',11,2000);

SELECT * FROM salesman;



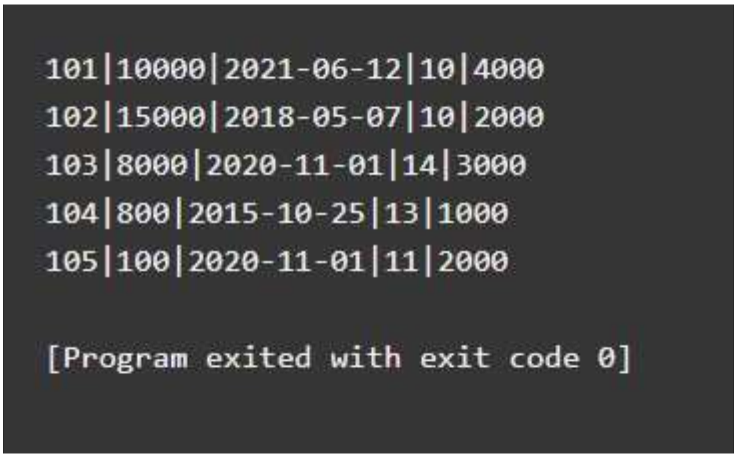
```
1000|Krishan|Bangalore|30%
2000|Abhay|Kolkata|25%
3000|Rahul|Mumbai|10%
4000|Rohan|Bangalore|15%
5000|Ron|Delhi|40%

[Program exited with exit code 0]
```

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;

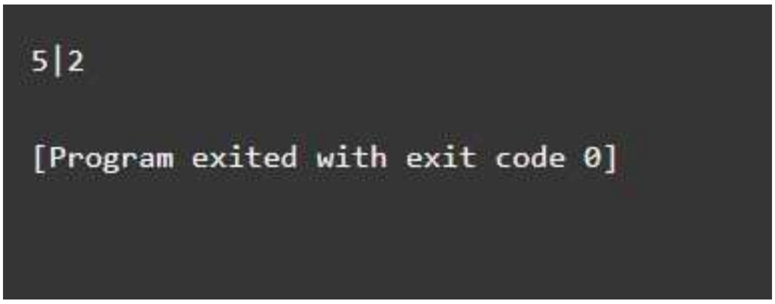


```
101|10000|2021-06-12|10|4000
102|15000|2018-05-07|10|2000
103|8000|2020-11-01|14|3000
104|800|2015-10-25|13|1000
105|100|2020-11-01|11|2000

[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

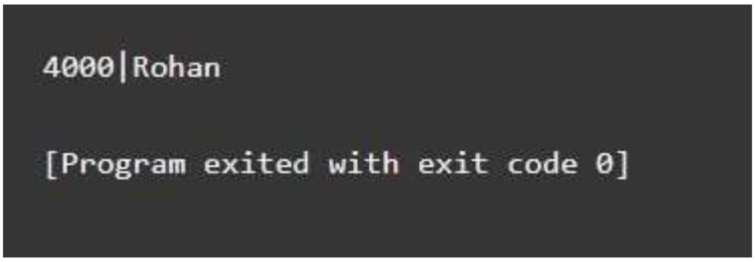
HAVING grade > (SELECT AVG(grade) FROM customer WHERE city = 'Bangalore');



```
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;



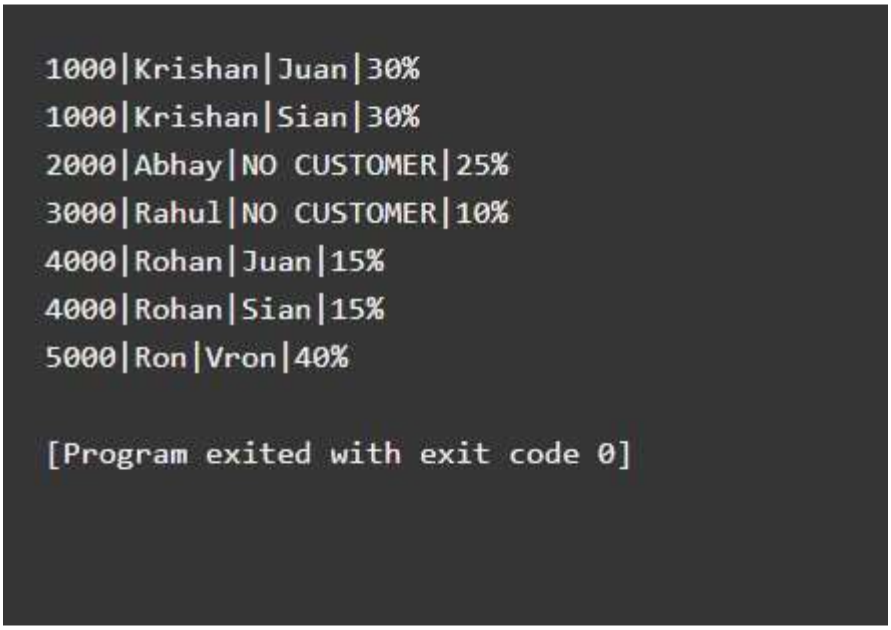
```
4000|Rohan

[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]
```

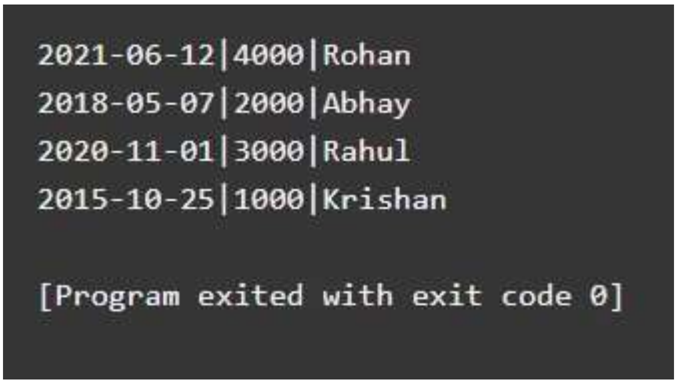
4. **Create a view that finds the salesman who has the customer with the highest order of a day.**

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;

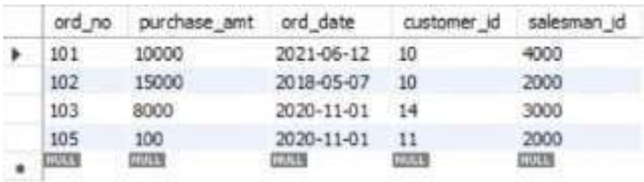


```
2021-06-12|4000|Rohan
2018-05-07|2000|Abhay
2020-11-01|3000|Rahul
2015-10-25|1000|Krishan

[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.

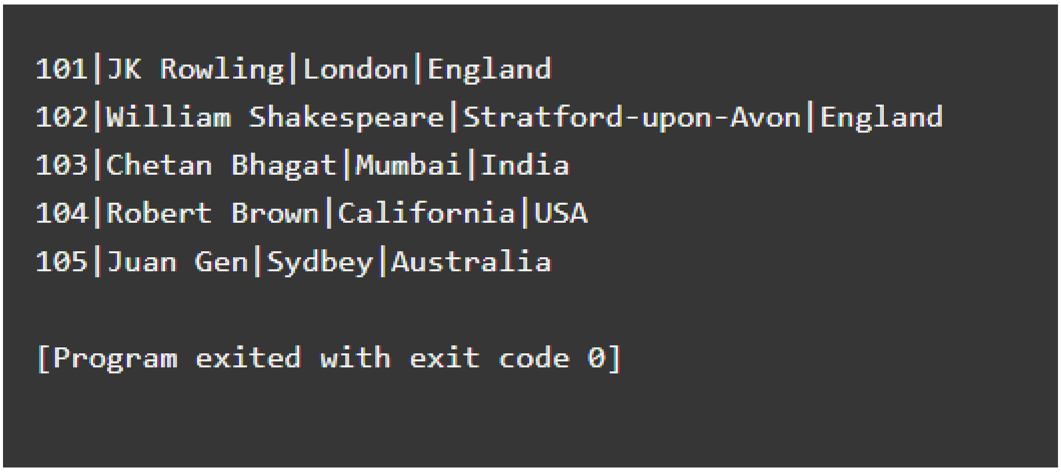
```
CREATE TABLE author(
author_id INT,
author_name VARCHAR(20),
author_city VARCHAR(20),
author_country VARCHAR(20),
PRIMARY KEY(author_id));

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');
```

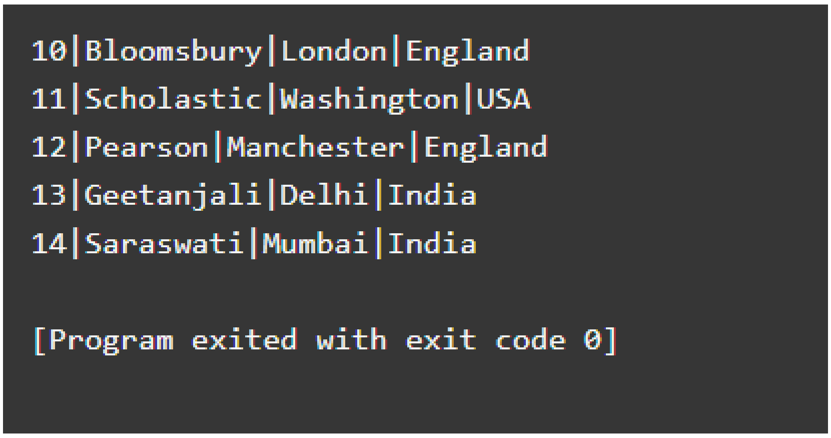
```
INSERT INTO category VALUES

(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;



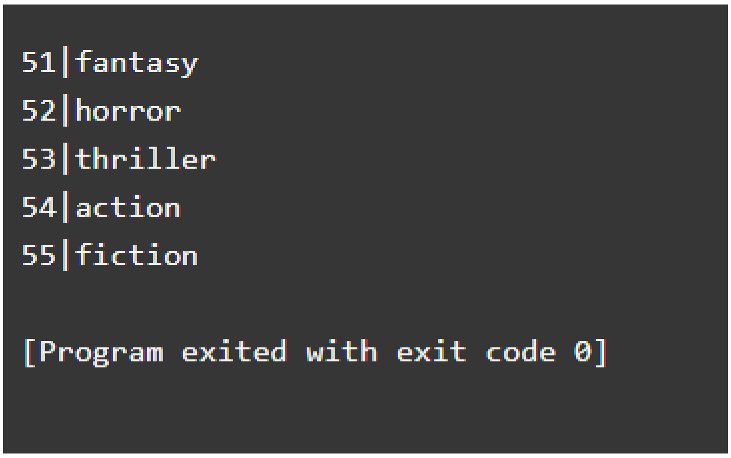
```
101|JK Rowling|London|England
102|William Shakespeare|Stratford-upon-Avon|England
103|Chetan Bhagat|Mumbai|India
104|Robert Brown|California|USA
105|Juan Gen|Sydbey|Australia

[Program exited with exit code 0]
```

SELECT * FROM publisher;



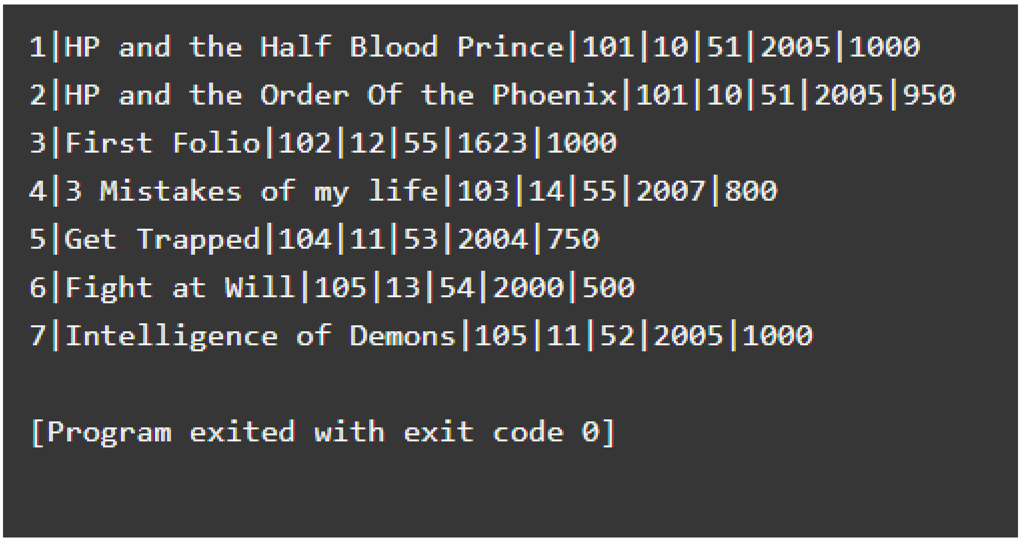
```
10|Bloomsbury|London|England
11|Scholastic|Washington|USA
12|Pearson|Manchester|England
13|Geetanjali|Delhi|India
14|Saraswati|Mumbai|India

[Program exited with exit code 0]
```

SELECT * FROM category;



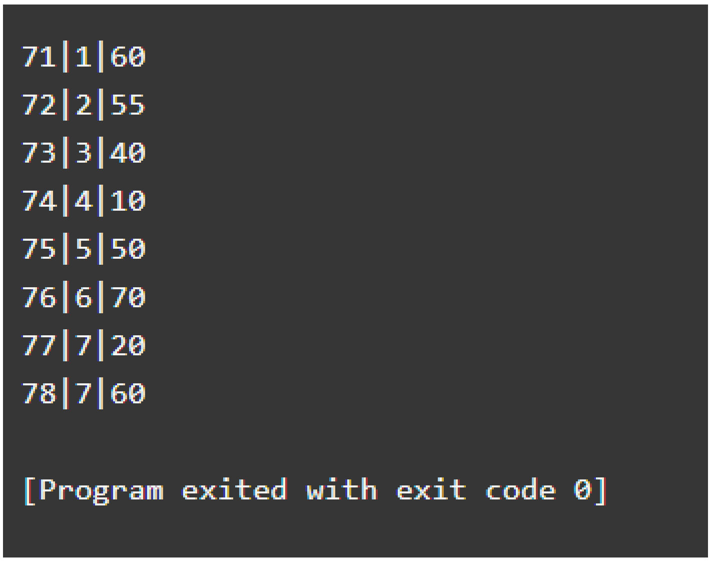
```
51|fantasy
52|horror
53|thriller
54|action
55|fiction

[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;



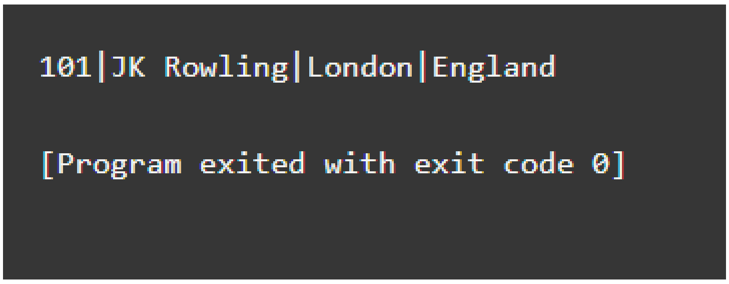
```
71|1|60
72|2|55
73|3|40
74|4|10
75|5|50
76|6|70
77|7|20
78|7|60

[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;

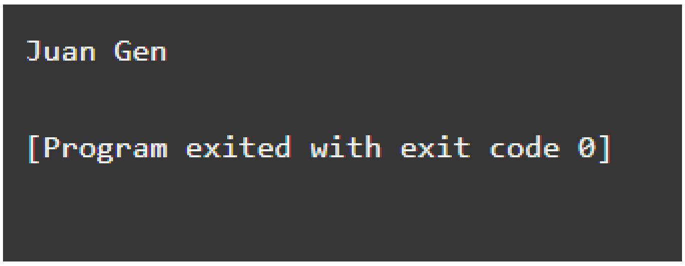


```
101|JK Rowling|London|England

[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));



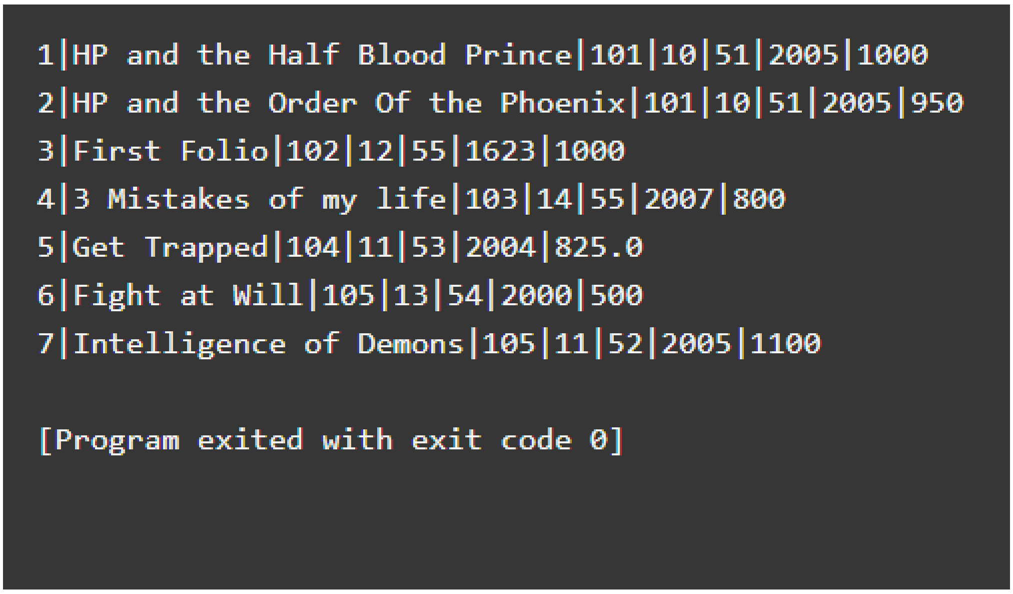
```
Juan Gen

[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);
```

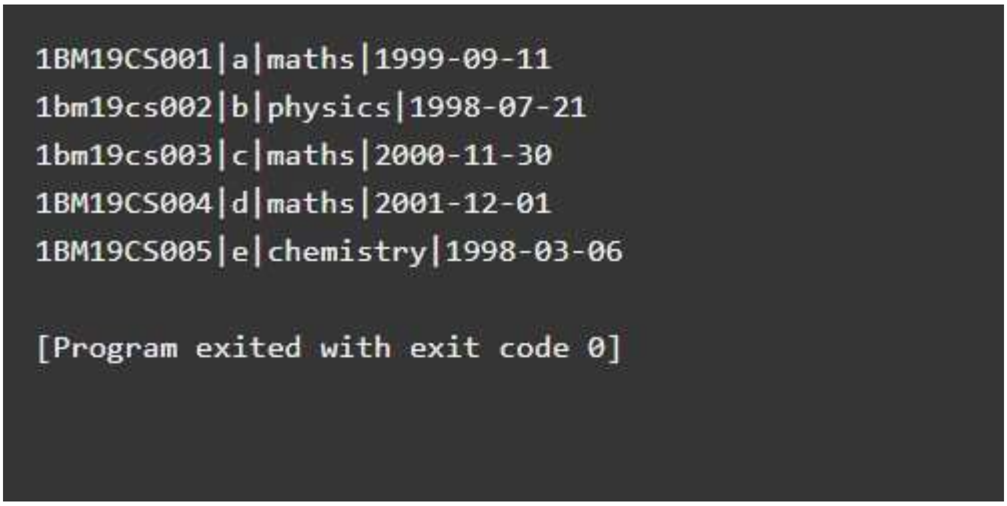
ii. Enter at least five tuples for each relation.

```
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');
```

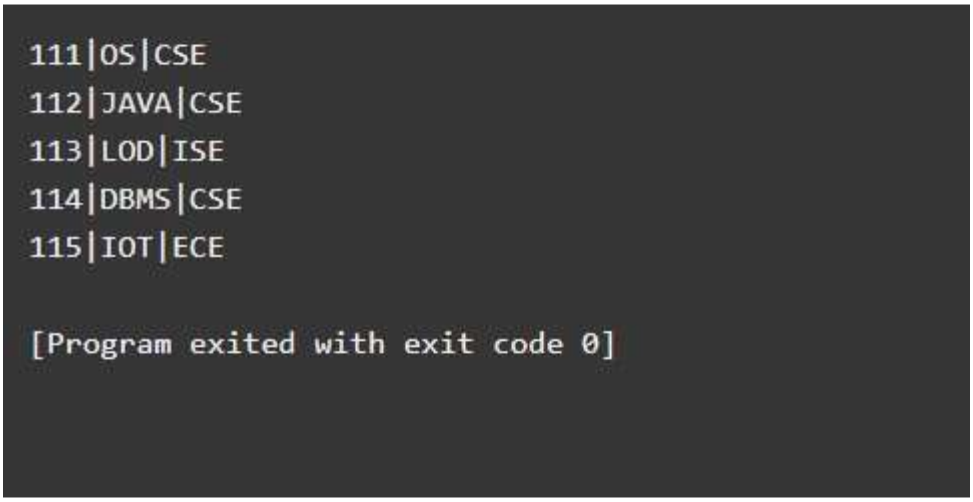
```
INSERT INTO book_adoption VALUES

(111,4,11),

(111,4,14),

(112,3,12),

(113,5,13),

(114,5,10),

(115,3,13);
```

```
SELECT * FROM student;
```



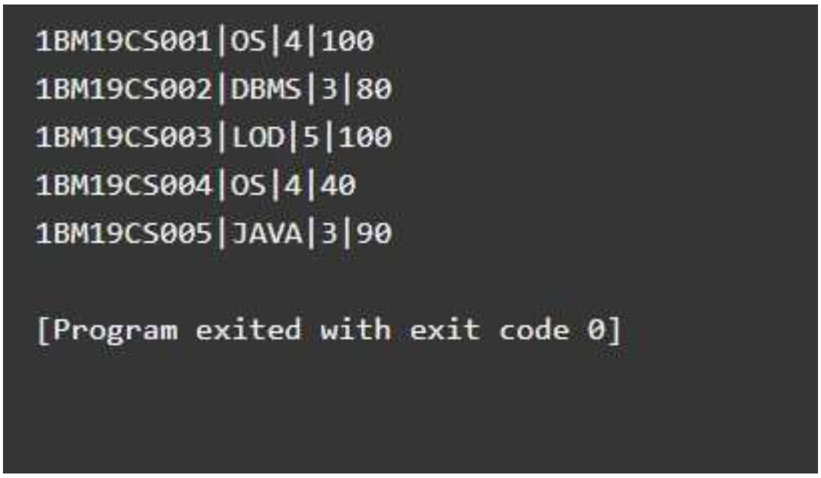
```
1BM19CS001|a|maths|1999-09-11
1bm19cs002|b|physics|1998-07-21
1bm19cs003|c|maths|2000-11-30
1BM19CS004|d|maths|2001-12-01
1BM19CS005|e|chemistry|1998-03-06

[Program exited with exit code 0]
```

```
SELECT * FROM course;
```



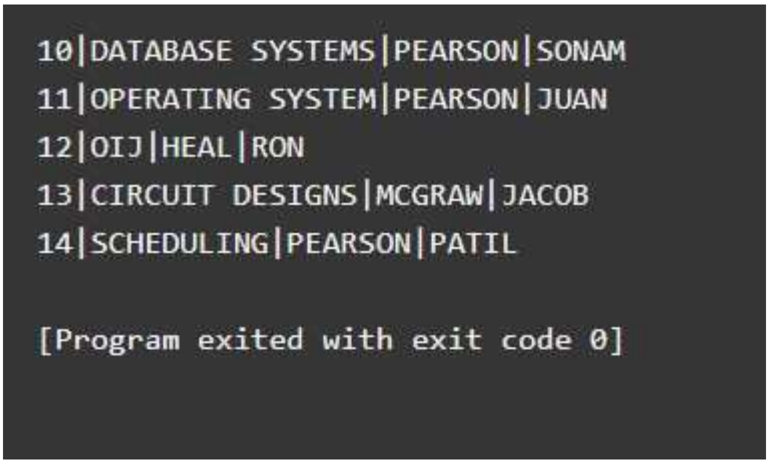
```
111|OS|CSE
112|JAVA|CSE
113|LOD|ISE
114|DBMS|CSE
115|IOT|ECE

[Program exited with exit code 0]
```

SELECT * FROM enroll;



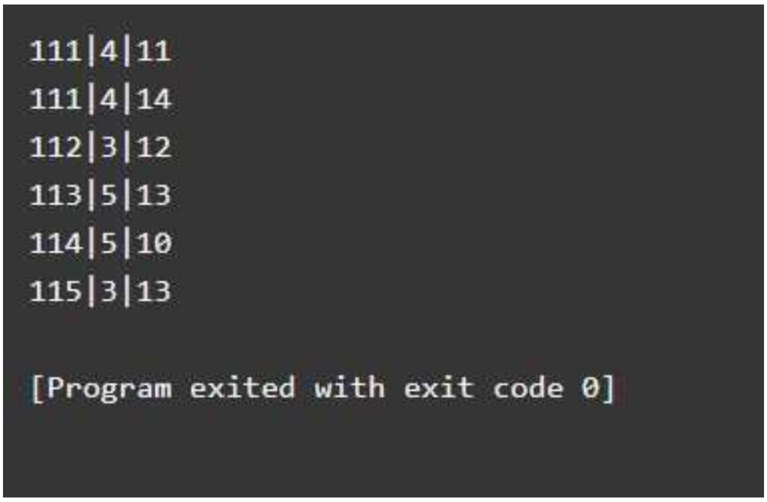
```
1BM19CS001|OS|4|100
1BM19CS002|DBMS|3|80
1BM19CS003|LOD|5|100
1BM19CS004|OS|4|40
1BM19CS005|JAVA|3|90

[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;

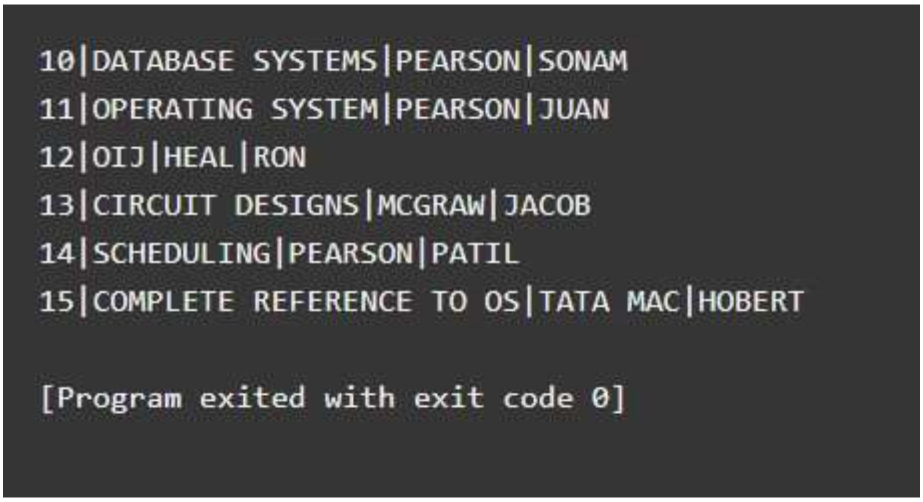


```
111|4|11
111|4|14
112|3|12
113|5|13
114|5|10
115|3|13

[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');

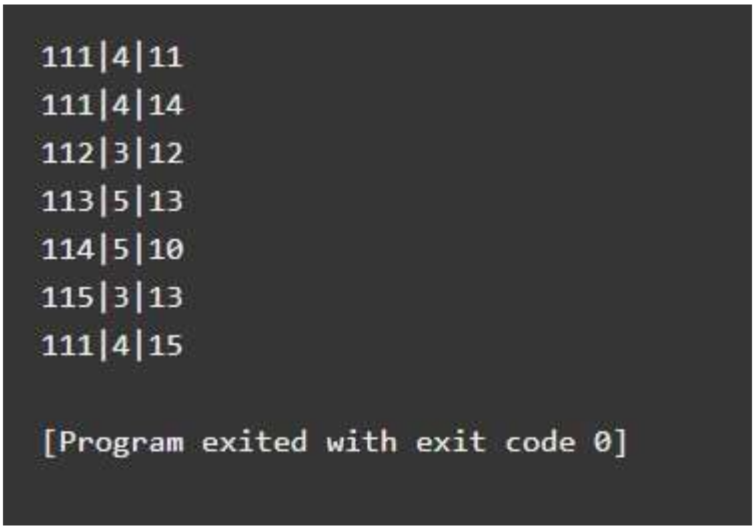
SELECT * FROM text;



```
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]
```

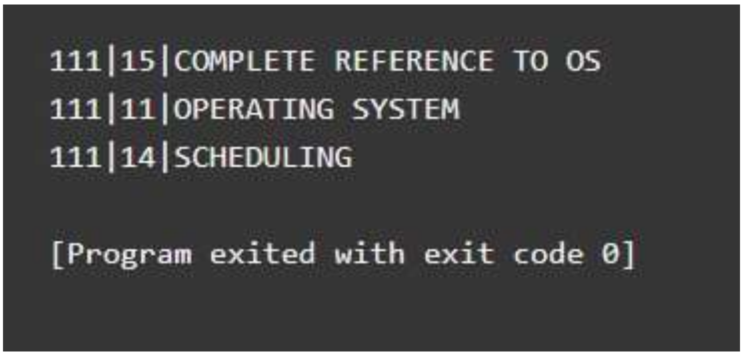
iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

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.

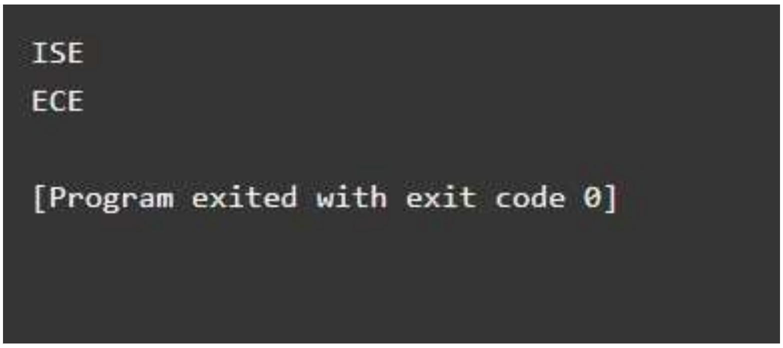
SELECT DISTINCT c.dept FROM course c WHERE c.dept IN

(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]
```

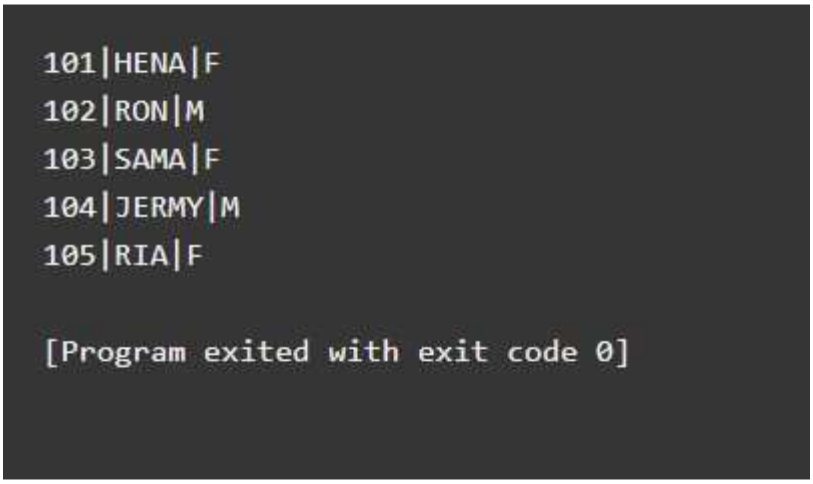
Lab 9 :-

```
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),

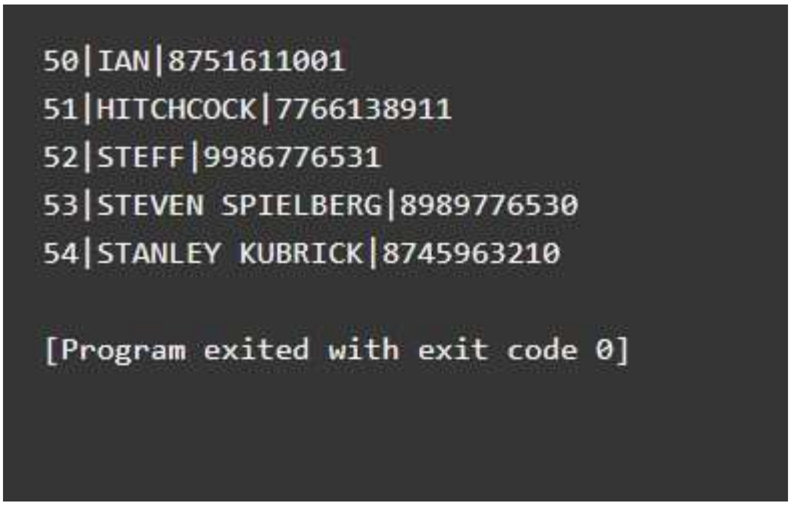
(1005,4);

SELECT * FROM ACTOR;



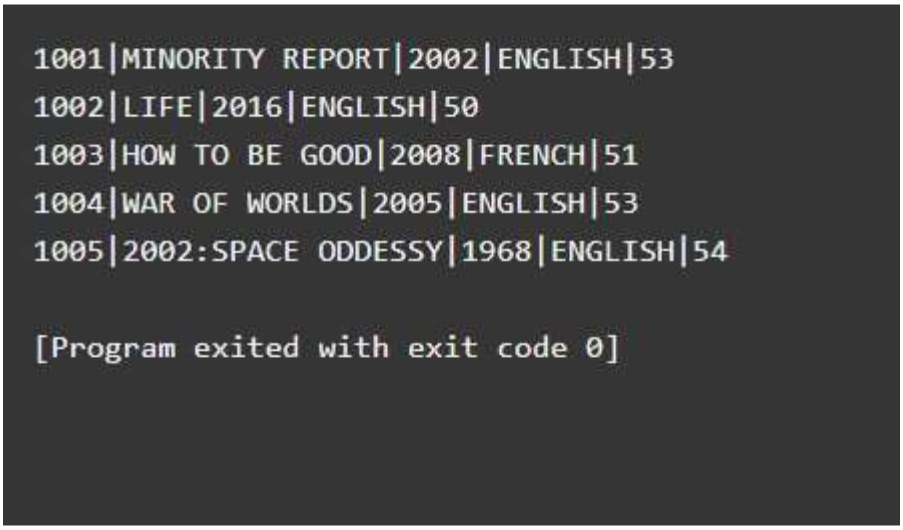
```
101|HENA|F
102|RON|M
103|SAMA|F
104|JERMY|M
105|RIA|F

[Program exited with exit code 0]
```

SELECT * FROM DIRECTOR;



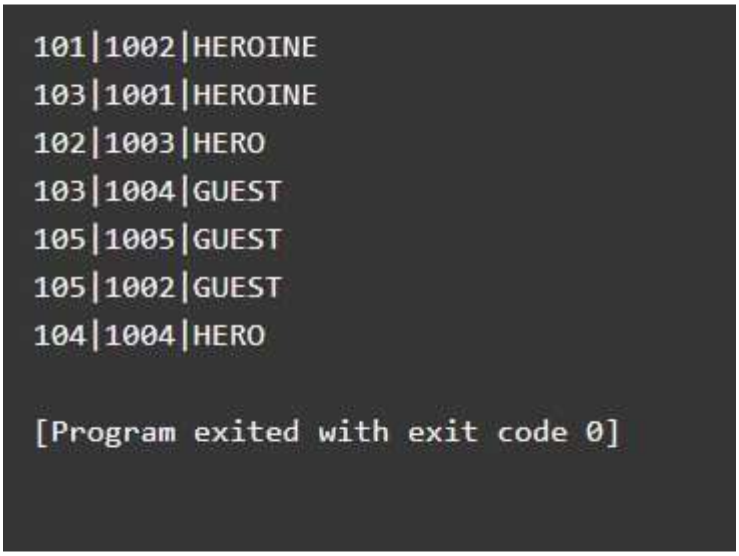
```
50|IAN|8751611001
51|HITCHCOCK|7766138911
52|STEFF|9986776531
53|STEVEN SPIELBERG|8989776530
54|STANLEY KUBRICK|8745963210

[Program exited with exit code 0]
```

SELECT * FROM MOVIES;



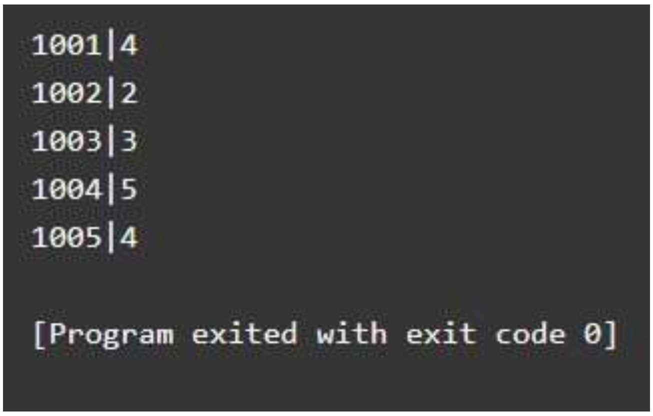
```
1001|MINORITY REPORT|2002|ENGLISH|53
1002|LIFE|2016|ENGLISH|50
1003|HOW TO BE GOOD|2008|FRENCH|51
1004|WAR OF WORLDS|2005|ENGLISH|53
1005|2002:SPACE ODDESSY|1968|ENGLISH|54

[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;

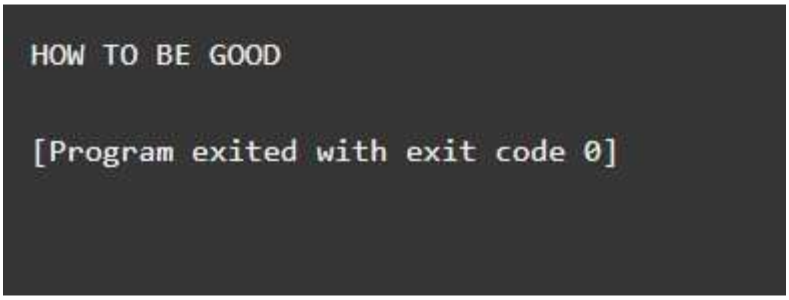


```
1001|4
1002|2
1003|3
1004|5
1005|4

[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]
```

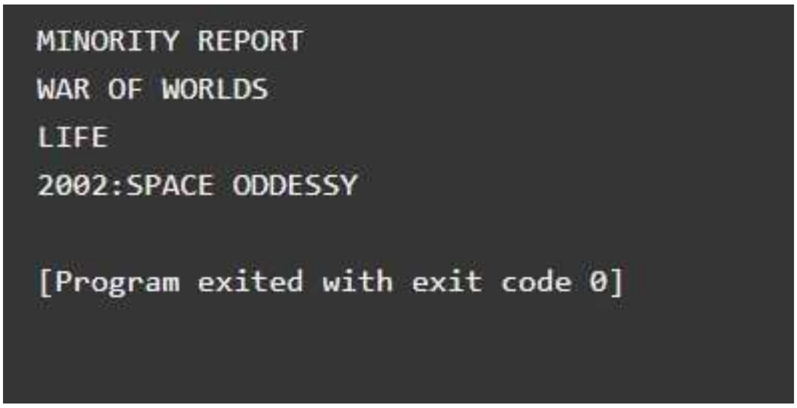
ii. Find the movie names where one or more actors acted in two or more movies.

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

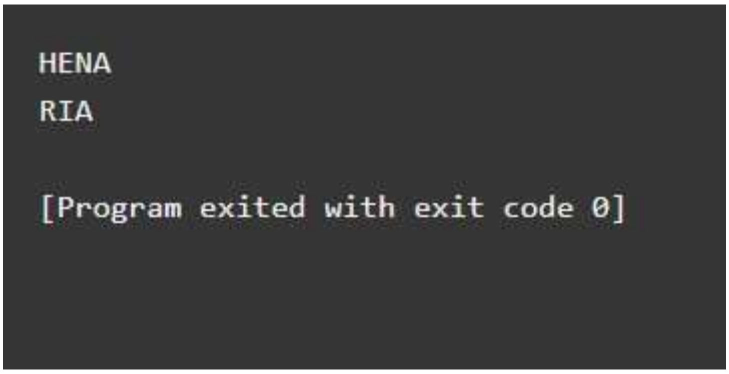
JOIN MOVIE_CAST C

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;



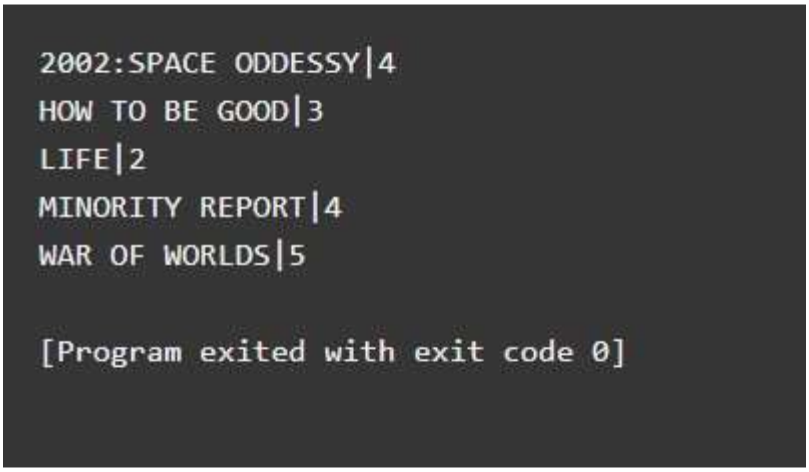
```
HENA
RIA

[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]
```

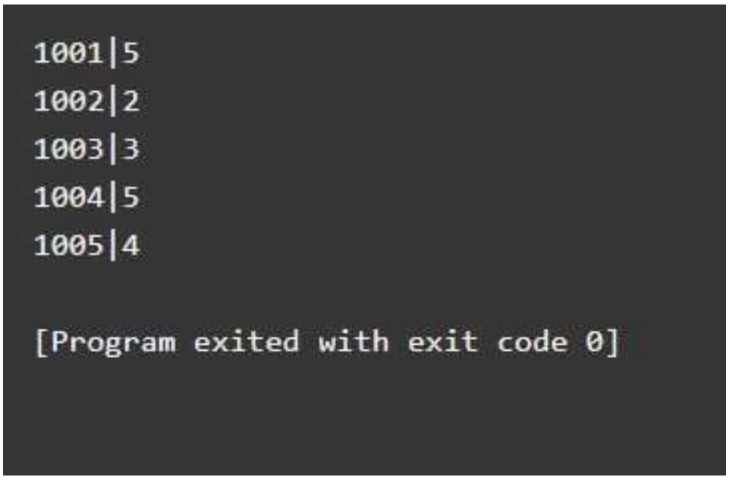
v. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

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;



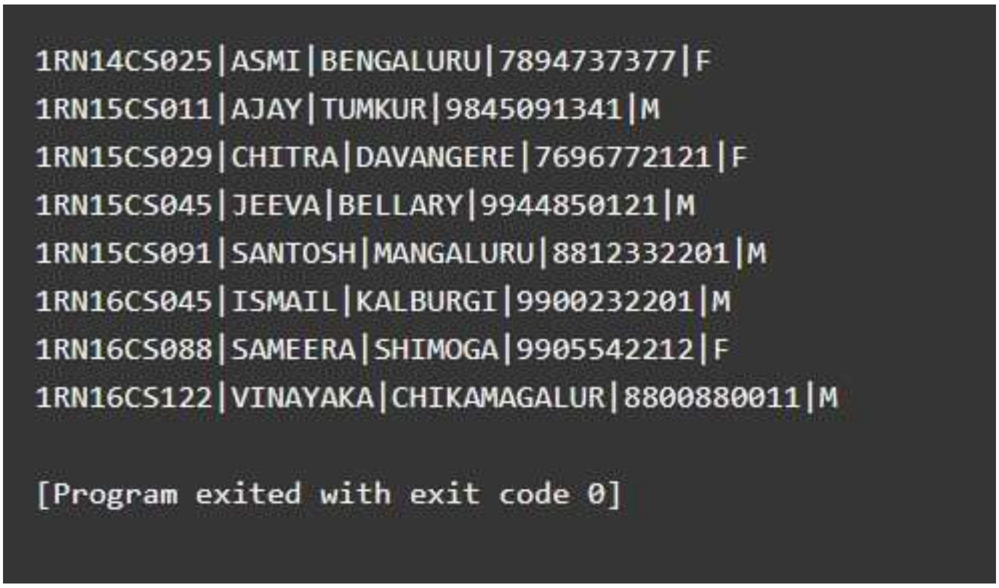
```
1001|5
1002|2
1003|3
1004|5
1005|4

[Program exited with exit code 0]
```

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);
```

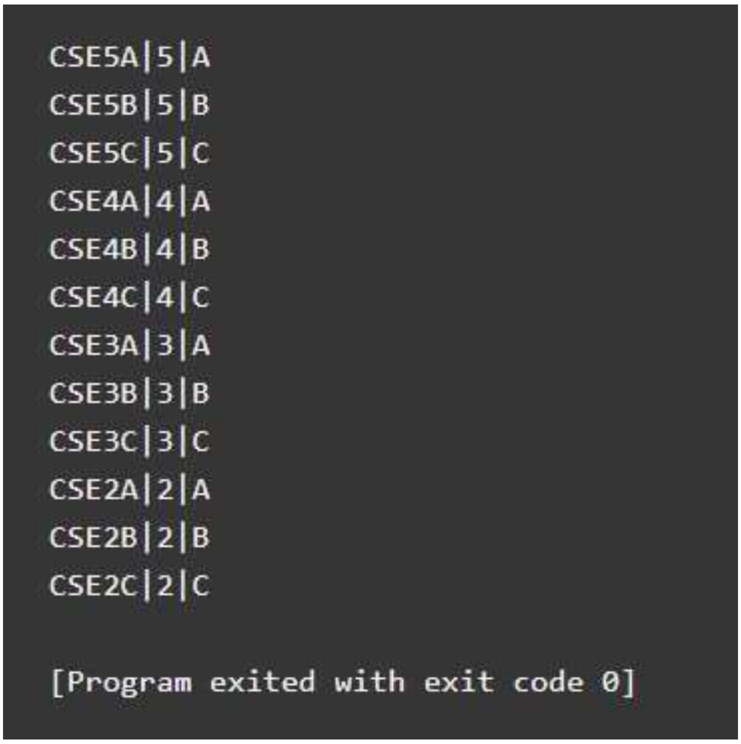
INSERT INTO IAMARKS VALUES ('1RN16CS122','15CS31','CSE3C', 20, 16, 19, 39); INSERT INTO IAMARKS VALUES ('1RN16CS122','15CS32','CSE3C', 15, 15, 12, 30);

SELECT * FROM STUDENT;



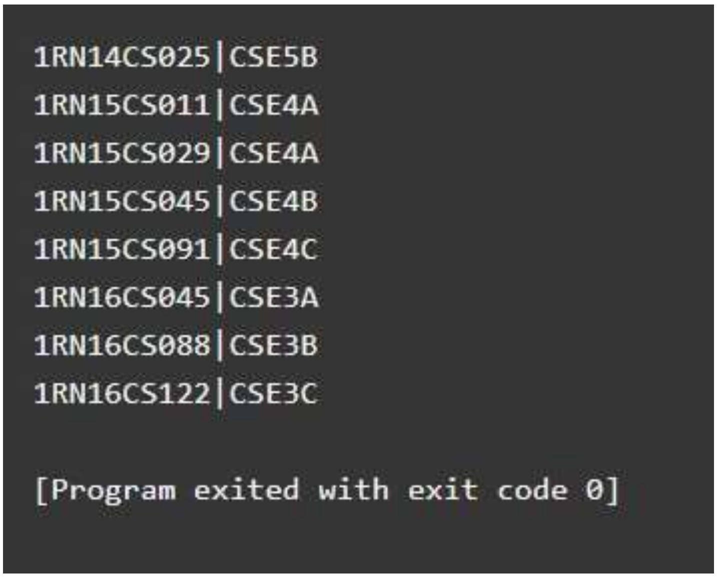
```
1RN14CS025|ASMI|BENGALURU|7894737377|F
1RN15CS011|AJAY|TUMKUR|9845091341|M
1RN15CS029|CHITRA|DAVANGERE|7696772121|F
1RN15CS045|JEEVA|BELLARY|9944850121|M
1RN15CS091|SANTOSH|MANGALURU|8812332201|M
1RN16CS045|ISMAIL|KALBURGI|9900232201|M
1RN16CS088|SAMEERA|SHIMOGA|9905542212|F
1RN16CS122|VINAYAKA|CHIKAMAGALUR|8800880011|M

[Program exited with exit code 0]
```

SELECT * FROM SEMSEC;



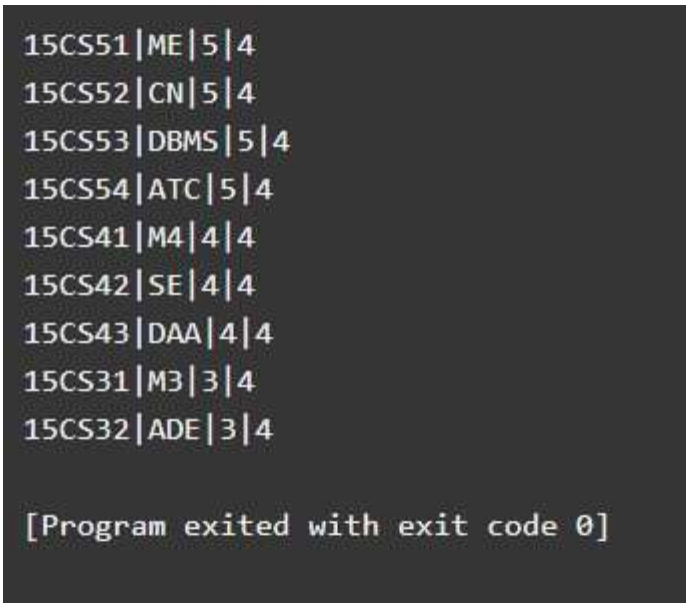
```
CSE5A|5|A
CSE5B|5|B
CSE5C|5|C
CSE4A|4|A
CSE4B|4|B
CSE4C|4|C
CSE3A|3|A
CSE3B|3|B
CSE3C|3|C
CSE2A|2|A
CSE2B|2|B
CSE2C|2|C

[Program exited with exit code 0]
```

SELECT * FROM CLASS;



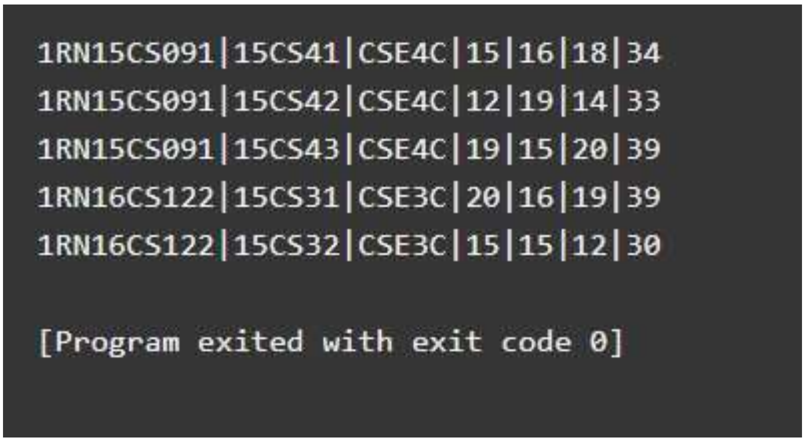
```
1RN14CS025|CSE5B
1RN15CS011|CSE4A
1RN15CS029|CSE4A
1RN15CS045|CSE4B
1RN15CS091|CSE4C
1RN16CS045|CSE3A
1RN16CS088|CSE3B
1RN16CS122|CSE3C

[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;

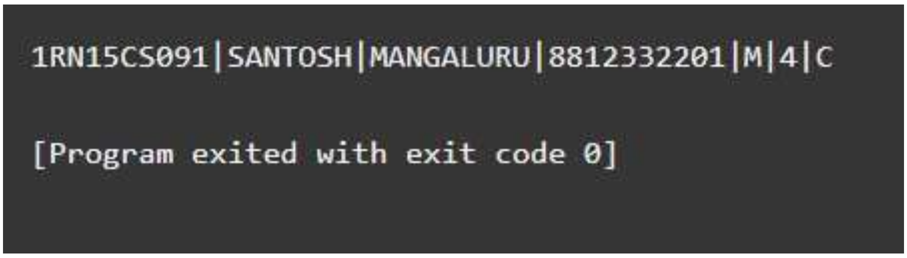


```
1RN15CS091|15CS41|CSE4C|15|16|18|34
1RN15CS091|15CS42|CSE4C|12|19|14|33
1RN15CS091|15CS43|CSE4C|19|15|20|39
1RN16CS122|15CS31|CSE3C|20|16|19|39
1RN16CS122|15CS32|CSE3C|15|15|12|30

[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';



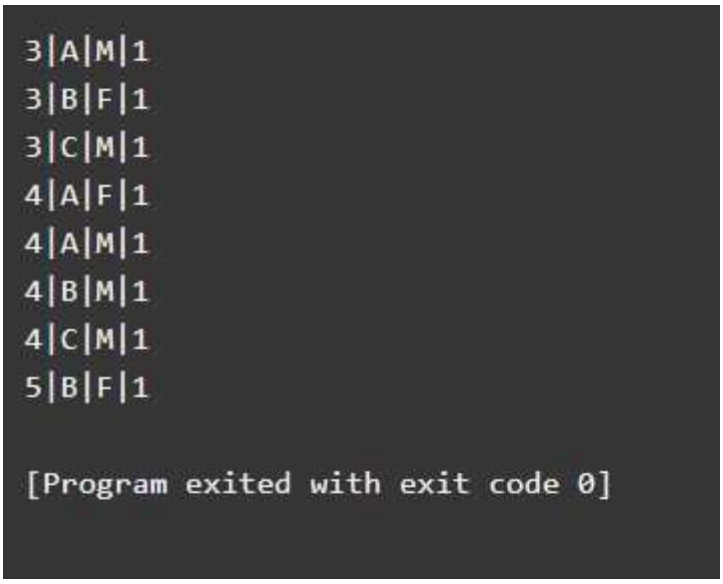
```
1RN15CS091|SANTOSH|MANGALURU|8812332201|M|4|C

[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]
```

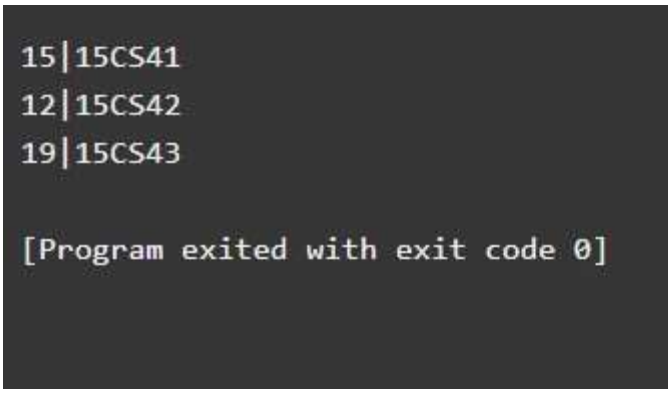
iii. Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.

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]
```