**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) Write SQL queries to**

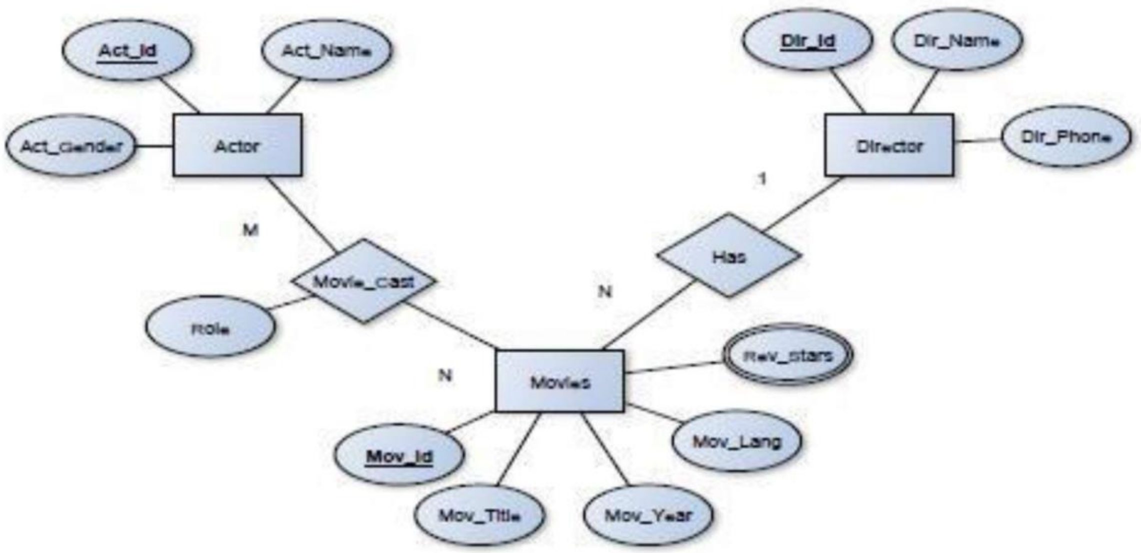
1. **List the titles of all movies directed by 'Hitchcock'.**
2. **Find the movie names where one or more actors acted in two or more movies.**
3. **List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).**
4. **Find the title of movies and number of stars 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.**
5. **Update rating of all movies directed by 'Steven Spielberg' to 5.**

**Program Objectives:**

**This course will enable students to**

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

**Solution:**
**Entity-Relationship Diagram**



**Schema Diagram**

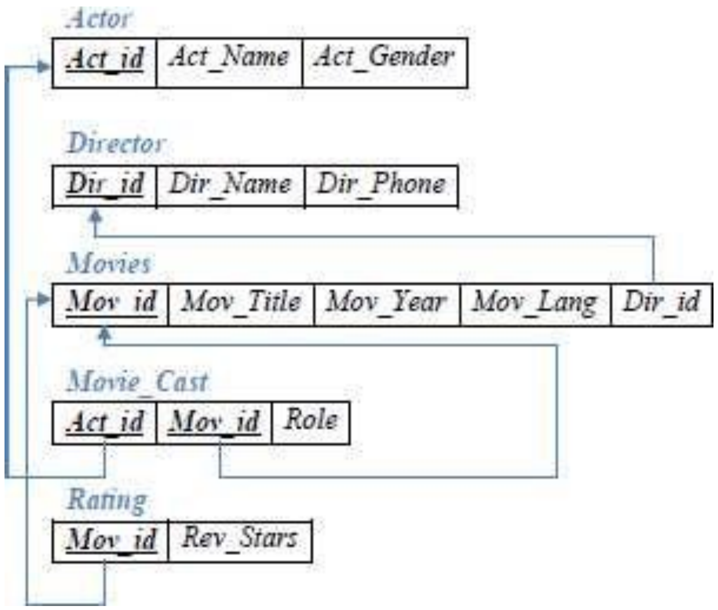


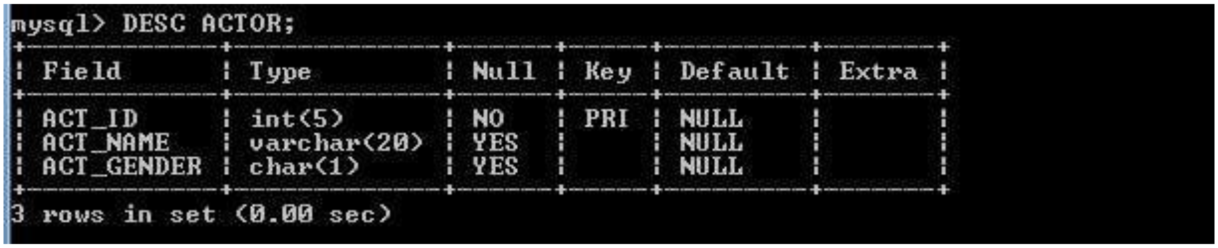
**Table Creation**

```
CREATE TABLE ACTOR (
ACT_ID NUMBER (5) PRIMARY
KEY, ACT_NAME VARCHAR
(20),
ACT_GENDER CHAR (1));

CREATE TABLE DIRECTOR (
DIR_ID NUMBER (5) PRIMARY
KEY, DIR_NAME VARCHAR (20),
DIR_PHONE NUMBER);

CREATE TABLE MOVIES
(MOV_ID NUMBER (4) PRIMARY
KEY, MOV_TITLE VARCHAR
(50),
MOV_YEAR NUMBER (4),
MOV_LANG VARCHAR (20),
DIR_ID NUMBER (5),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR(DIR_ID));

CREATE TABLE MOVIES_CAST (
ACT_ID NUMBER (5),
MOV_ID NUMBER (5),
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 NUMBER (5) PRIMARY
KEY, REV_STARS VARCHAR
(25),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));

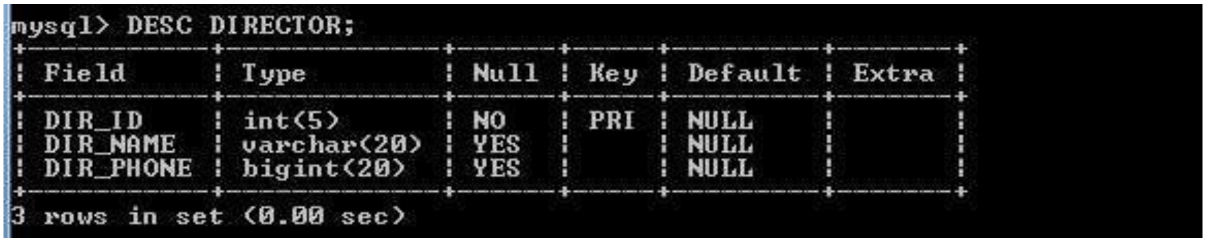
**Table Descriptions**

DESC ACTOR;



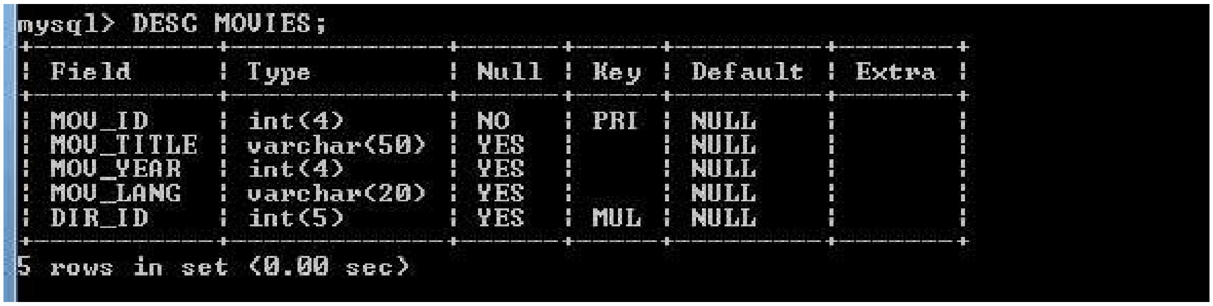
```
mysql> DESC ACTOR;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| ACT_ID     | int(5)      | NO   | PRI | NULL    |       |
| ACT_NAME   | varchar(20) | YES  |     | NULL    |       |
| ACT_GENDER | char(1)     | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

DESC DIRECTOR;



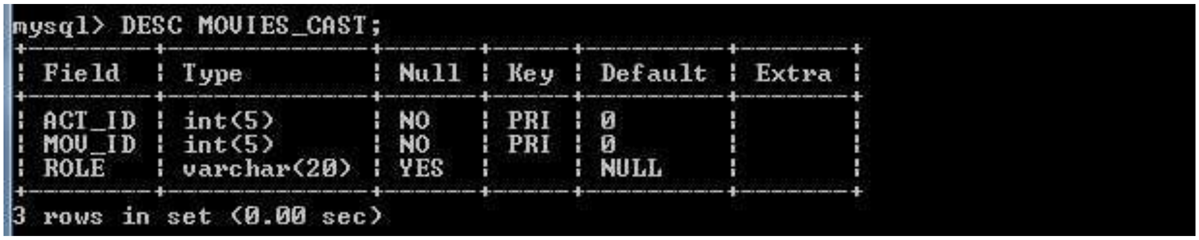
```
mysql> DESC DIRECTOR;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| DIR_ID    | int(5)      | NO   | PRI | NULL    |       |
| DIR_NAME  | varchar(20) | YES  |     | NULL    |       |
| DIR_PHONE | bigint(20)  | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

DESC MOVIES;



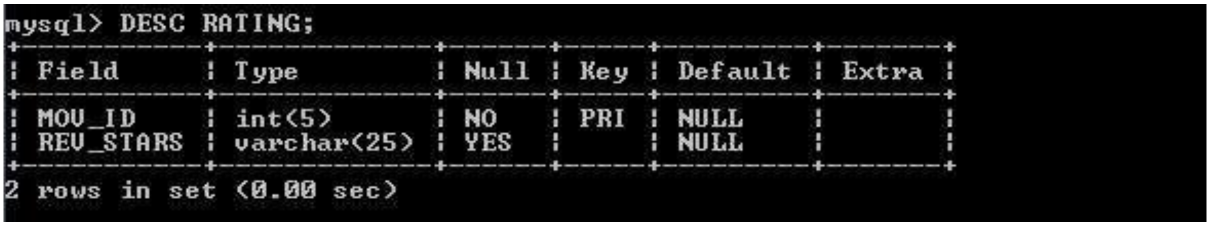
```
mysql> DESC MOVIES;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| MOV_ID    | int(4)      | NO   | PRI | NULL    |       |
| MOV_TITLE | varchar(50) | YES  |     | NULL    |       |
| MOV_YEAR  | int(4)      | YES  |     | NULL    |       |
| MOV_LANG  | varchar(20) | YES  |     | NULL    |       |
| DIR_ID    | int(5)      | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

DESC MOVIES_CAST;



```
mysql> DESC MOVIES_CAST;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| ACT_ID | int(5)      | NO   | PRI | 0       |       |
| MOV_ID | int(5)      | NO   | PRI | 0       |       |
| ROLE   | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

DESC RATING;



```
mysql> DESC RATING;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| MOV_ID    | int(5)      | NO   | PRI | NULL    |       |
| REV_STARS | varchar(25) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

**Insertion of Values to Tables**

```
INSERT INTO ACTOR VALUES (1,’MADHURI DIXIT’,’F’);
INSERT INTO ACTOR VALUES (2,’AAMIR KHAN’,’M’);
INSERT INTO ACTOR VALUES (3,’JUHI CHAWLA’,’F’);
INSERT INTO ACTOR VALUES (4,’SRIDEVI’,’F’);

INSERT INTO DIRECTOR VALUES (100,’SUBHASH KAPOOR’, 9563400156);
INSERT INTO DIRECTOR VALUES(102,'ALAN TAYLOR',9971960035);
INSERT INTO DIRECTOR VALUES (103,’SANTHOSH ANANDDRAM’, 9934611125);
INSERT INTO DIRECTOR VALUES (104,’IMTIAZ ALI’, 8539920975);
INSERT INTO DIRECTOR VALUES (105,'HITCHCOCK',7766138911);
INSERT INTO DIRECTOR VALUES (106,'STEVEN SPIELBERG',9966138934);


INSERT INTO MOVIES VALUES (501,'JAB HARRY MET SEJAL',2017,'HINDI',104);
INSERT INTO MOVIES VALUES (502,'RAJAKUMARA',2017,'KANNADA',103);
INSERT INTO MOVIES VALUES (503,'JOLLY LLB 2', 2013,'HINDI', 100);
INSERT INTO MOVIES VALUES (504,'TERMINATOR GENESYS',2015,'ENGLISH',102);
INSERT INTO MOVIES VALUES (505,'JAWS',1975,'ENGLISH',106);
INSERT INTO MOVIES VALUES (506,'BRIDGE OF SPIES',2015,'ENGLISH', 106);
INSERT INTO MOVIES VALUES (507,'VERTIGO',1943,'ENGLISH',105);
INSERT INTO MOVIES VALUES (508,’SHADOW OF A DOUBT',1943,'ENGLISH', 105);

INSERT INTO MOVIES_CAST VALUES (1, 501,'HEROINE');
INSERT INTO MOVIES_CAST VALUES (1, 502,'HEROINE');
INSERT INTO MOVIES_CAST VALUES (3, 503,'COMEDIAN');
INSERT INTO MOVIES_CAST VALUES (4, 504,'GUEST');
INSERT INTO MOVIES_CAST VALUES (4, 501,'HERO');

INSERT INTO RATING VALUES (501, 4);
INSERT INTO RATING VALUES (502, 2);
INSERT INTO RATING VALUES (503, 5);
INSERT INTO RATING VALUES (504, 4);
INSERT INTO RATING VALUES (505, 3);
INSERT INTO RATING VALUES (506, 2);
```

SELECT * FROM ACTOR;

| ACT_ID | ACT_NAME | ACT |
|---|---|---|
| 1 | MADHURI DIXIT | F |
| 2 | AAMIR KHAN | M |
| 3 | JUHI CHAWLA | F |
| 4 | SRIDEVI | F |

SELECT * FROM DIRECTOR;

| DIR_ID | DIR_NAME | DIR_PHONE |
|---|---|---|
| 100 | SUBHASH KAPOOR | 56340015 |
| 102 | ALAN TAYLOR | 719600310 |
| 103 | SANTHOSH ANANDDRAM | 99346111 |
| 104 | IMTIAZ ALI | 85399209 |
| 105 | HITCHCOCK | 7766138911 |
| 106 | STEVEN SPIELBERG | 9966138934 |

SELECT * FROM MOVIES;

| MOV_ID | MOV_TITLE | MOV_YEAR | MOV_LANG | DIR_ID |
|---|---|---|---|---|
| 501 | JAB HARRY MET SEJAL | 2017 | HINDI | 104 |
| 502 | RAJAKUMARA | 2017 | KANNADA | 103 |
| 503 | JOLLY LLB 2 | 2013 | HINDI | 100 |
| 504 | TERMINATOR GENESYS | 2015 | ENGLISH | 102 |
| 505 | JAWS | 1975 | ENGLISH | 106 |
| 506 | BRIDGE OF SPIES | 2015 | ENGLISH | 106 |
| 507 | VERTIGO | 1958 | ENGLISH | 105 |
| 508 | SHADOW OF A DOUBT | 1943 | ENGLISH | 105 |

SELECT * FROM MOVIE_CAST;

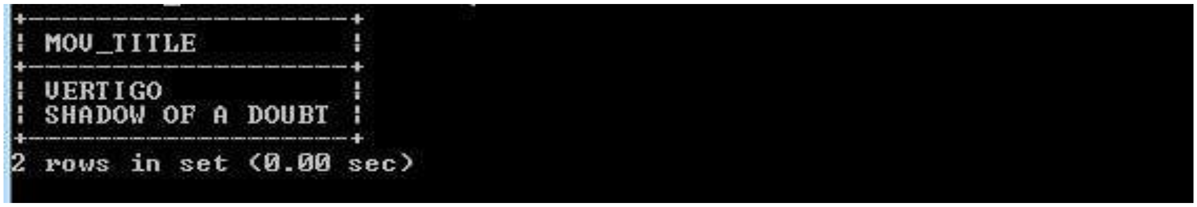
| MOV_ID | MOV_TITLE | MOV_YEAR | MOV_LANG | DIR_ID |
|---|---|---|---|---|
| 501 | JAB HARRY MET SEJAL | 2017 | HINDI | 104 |
| 502 | RAJAKUMARA | 2017 | KANNADA | 103 |
| 503 | JOLLY LLB 2 | 2013 | HINDI | 100 |
| 504 | TERMINATOR GENESYS | 2015 | ENGLISH | 102 |
| 505 | JAWS | 1975 | ENGLISH | 106 |
| 506 | BRIDGE OF SPIES | 2015 | ENGLISH | 106 |
| 507 | VERTIGO | 1958 | ENGLISH | 105 |
| 508 | SHADOW OF A DOUBT | 1943 | ENGLISH | 105 |

SELECT * FROM RATING;

| MOV_ID | REV_STARS |
|---|---|
| 501 | 4 |
| 502 | 2 |
| 503 | 5 |
| 504 | 4 |
| 505 | 3 |
| 506 | 2 |
| 507 | 2 |
| 508 | 4 |

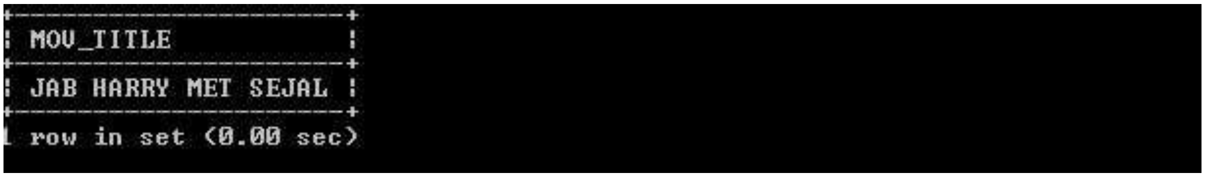
**Queries:**

1. **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');
   **OR**
   SELECT MOV_TITLE FROM MOVIES M, DIRECTOR D WHERE M.DIR_ID=D.DIR_ID AND DIR_NAME='HITCHCOCK';



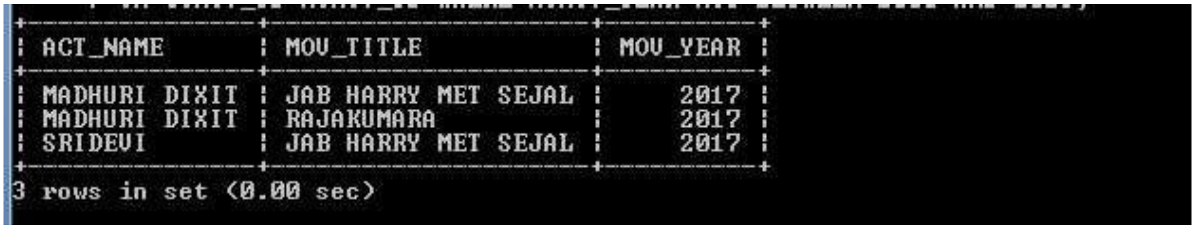
```
+--------------------+
| MOV_TITLE          |
+--------------------+
| VERTIGO            |
| SHADOW OF A DOUBT  |
+--------------------+
2 rows in set (0.00 sec)
```

2. **Find the movie names where one or more actors acted in two or more movies.**
   SELECT MOV_TITLE FROM MOVIES M,MOVIES_CAST MV
   WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN(SELECT ACT_ID FROM MOVIES_CAST GROUP BY ACT_ID HAVING COUNT(ACT_ID)>1) GROUP BY MOV_TITLE HAVING COUNT(*)>1;



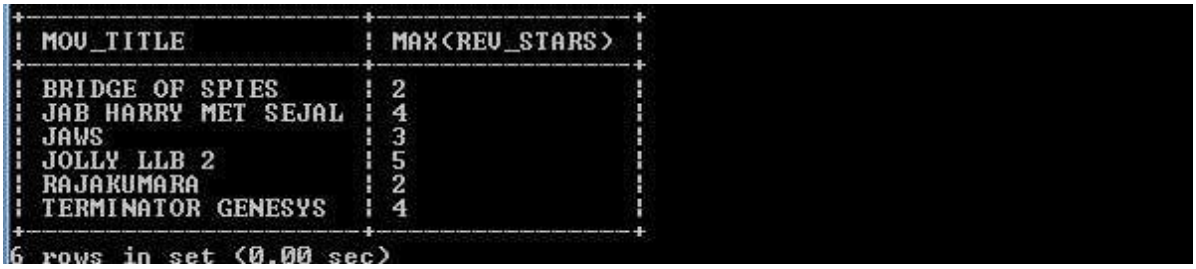
```
+--------------------+
| MOV_TITLE          |
+--------------------+
| JAB HARRY MET SEJAL |
+--------------------+
1 row in set (0.00 sec)
```

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 C ON A.ACT_ID=C.ACT_ID INNER JOIN MOVIES M
   ON C.MOV_ID=M.MOV_ID WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;



```
+---------------+---------------------+----------+
| ACT_NAME      | MOV_TITLE           | MOV_YEAR |
+---------------+---------------------+----------+
| MADHURI DIXIT | JAB HARRY MET SEJAL |     2017 |
| MADHURI DIXIT | RAJAKUMARA          |     2017 |
| SRIDEVI       | JAB HARRY MET SEJAL |     2017 |
+---------------+---------------------+----------+
3 rows in set (0.00 sec)
```

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 M ,RATING R WHERE M.MOV_ID=R.MOV_ID GROUP BY MOV_TITLE HAVING MAX(REV_STARS)>0 ORDER BY MOV_TITLE;



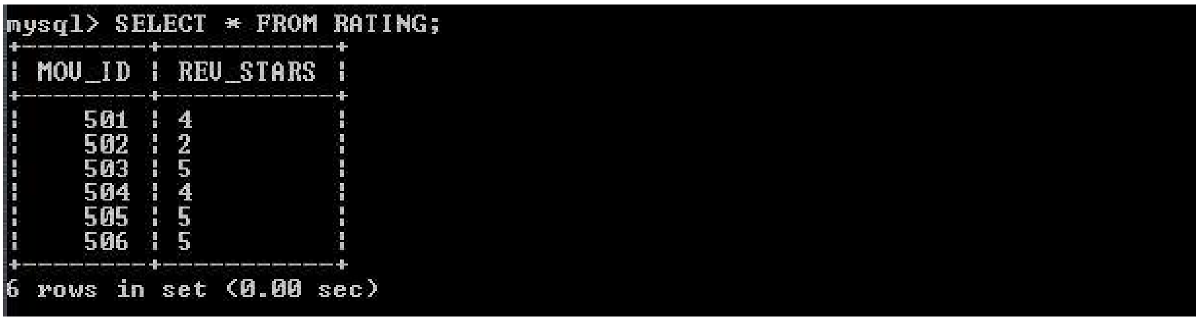
```
+-----------------------+------------------+
| MOV_TITLE             | MAX(REV_STARS)   |
+-----------------------+------------------+
| BRIDGE OF SPIES       | 2                |
| JAB HARRY MET SEJAL   | 4                |
| JAWS                  | 3                |
| JOLLY LLB 2           | 5                |
| RAJAKUMARA            | 2                |
| TERMINATOR GENESYS    | 4                |
+-----------------------+------------------+
6 rows in set (0.00 sec)
```

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'));

**OR**

UPDATE RATING R, MOVIES M, DIRECTOR D SET REV_STARS=5 WHERE R.MOV_ID=M.MOV_ID AND M.DIR_ID=D.DIR_ID AND DIR_NAME='STEVEN SPIELBERG';



```
mysql> SELECT * FROM RATING;
+--------+-----------+
| MOV_ID | REV_STARS |
+--------+-----------+
|    501 | 4         |
|    502 | 2         |
|    503 | 5         |
|    504 | 4         |
|    505 | 5         |
|    506 | 5         |
+--------+-----------+
6 rows in set (0.00 sec)
```

**Program Outcomes:**

**The students are able to**

- Create, Update and query on the database.
- Demonstrate the working of different concepts of DBMS
- Implement, analyze and evaluate the project developed for an application.