

LAB EXERCISE - 3

MOVIE DATABASE

A series of horizontal lines in teal and light blue colors, with varying lengths and thicknesses, extending from the left edge of the slide towards the right.

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)

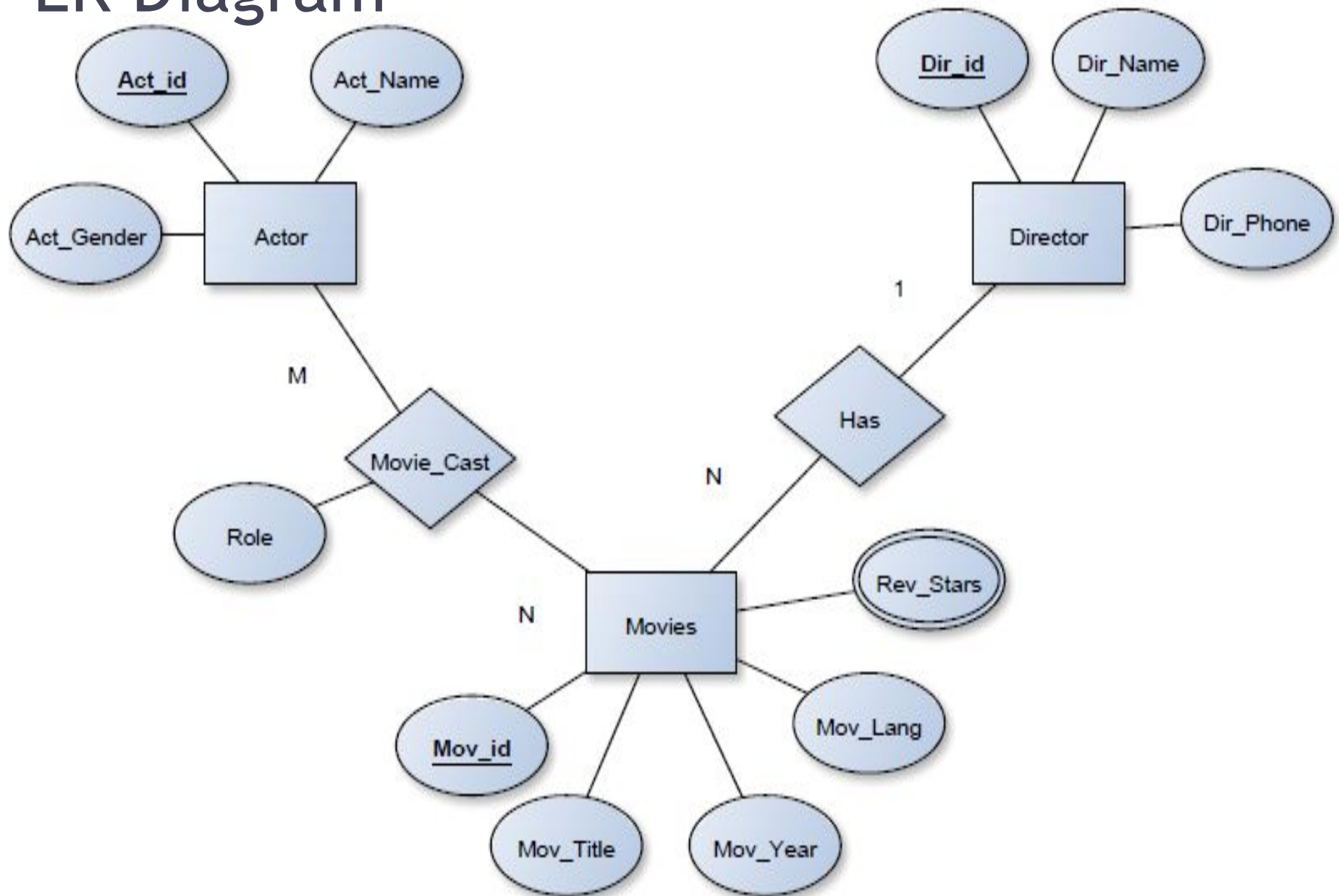
MOVIE_CAST(Act_id, Mov_id, Role)

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.

ER Diagram



Schema Diagram

ACTOR

<u>Act_id</u>	Act_Name	Act_Gender
---------------	----------	------------

DIRECTOR

<u>Dir_id</u>	Dir_Name	Dir_Phone
---------------	----------	-----------

MOVIES

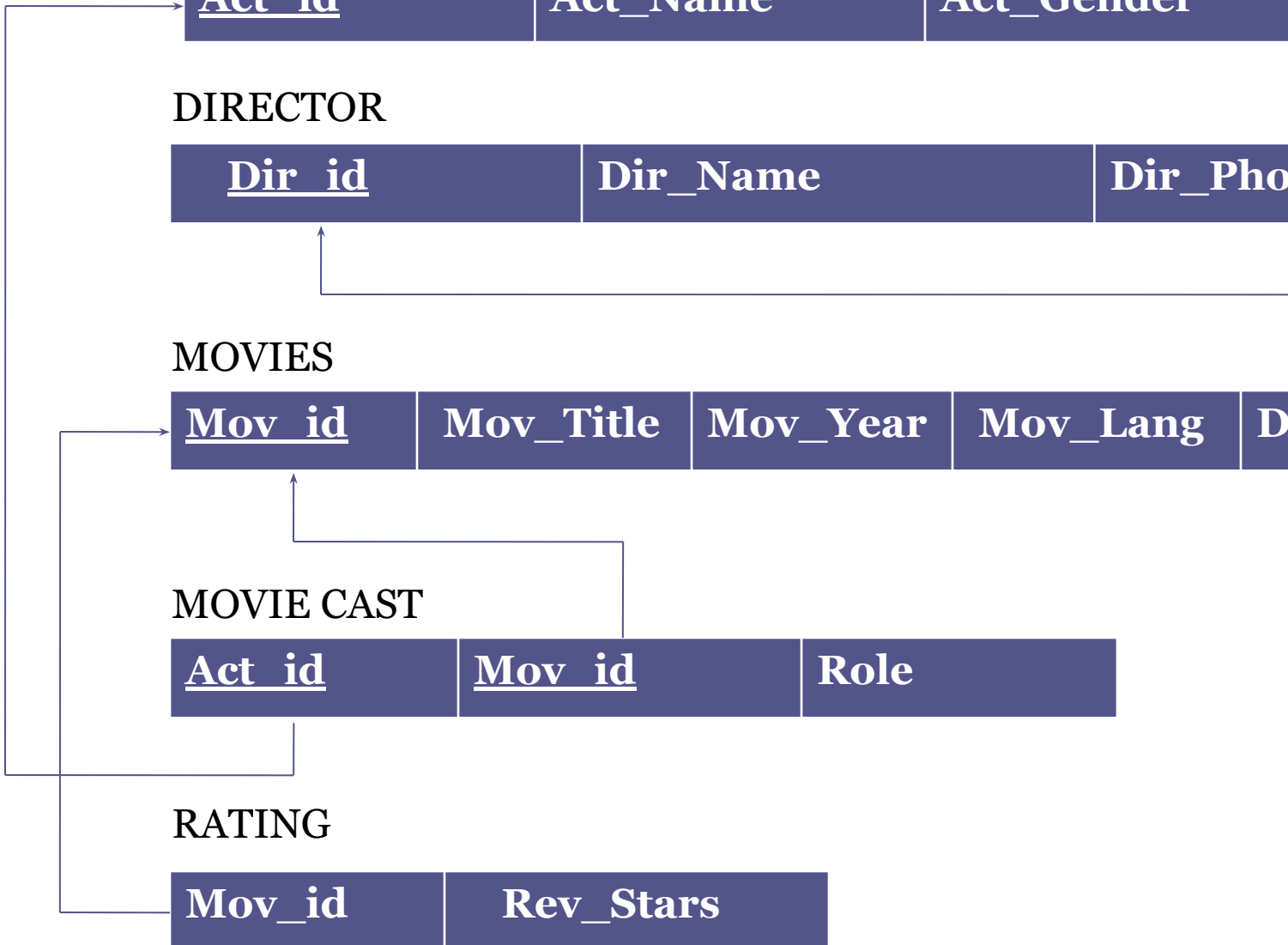
<u>Mov_id</u>	Mov_Title	Mov_Year	Mov_Lang	Dir_id
---------------	-----------	----------	----------	--------

MOVIE CAST

<u>Act_id</u>	<u>Mov_id</u>	Role
---------------	---------------	------

RATING

Mov_id	Rev_Stars
--------	-----------



CREATE TABLE ACTOR

(

ACT_ID VARCHAR(4),

ACT_NAME VARCHAR2(10),

ACT_GENDER VARCHAR2(10),

PRIMARY KEY(ACT_ID)

);

CREATE TABLE DIRECTOR

(

DIR_ID VARCHAR(5),

DIR_NAME VARCHAR(20),

DIR_PHONE NUMBER(10),

PRIMARY KEY(DIR_ID)

);

CREATE TABLE MOVIES

(

**MOV_ID VARCHAR(5),
MOV_TITLE VARCHAR(20),
MOV_YEAR INT,
MOV_LANG CHAR(10),
DIR_ID VARCHAR(10),
PRIMARY KEY(MOV_ID),
FOREIGN KEY(DIR_ID) REFERENCES
DIRECTOR(DIR_ID)
ON DELETE CASCADE**

);

CREATE TABLE MOVIE_CAST

(

ACT_ID VARCHAR(5),

MOV_ID VARCHAR(5),

ROLE VARCHAR(10),

PRIMARY KEY(ACT_ID, MOV_ID),

**FOREIGN KEY(ACT_ID) REFERENCES
ACTOR(ACT_ID)**

ON DELETE CASCADE,

**FOREIGN KEY(MOV_ID) REFERENCES
MOVIES(MOV_ID)**

ON DELETE CASCADE

);

CREATE TABLE RATING

(

MOV_ID VARCHAR(5),

REV_STARS INT,

FOREIGN KEY(MOV_ID) REFERENCES

MOVIES(MOV_ID)

ON DELETE CASCADE

);

- INSERT INTO ACTOR VALUES
('&ACT_ID', '&ACT_NAME', '&ACT_GENDER');
- INSERT INTO DIRECTOR VALUES ('&DIR_ID',
'&DIR_NAME', &DIR_PHONE);
- INSERT INTO MOVIES VALUES ('&MOV_ID',
'&MOV_TITLE ', &MOV_YEAR, '&MOV_LANG', '&DIR_ID');
- INSERT INTO MOVIES_CAST VALUES ('&ACT_ID',
('&MOV_ID', '&ROLE ') ;
- INSERT INTO RATING VALUES ('&MOV_ID',
&REV_STARS) ;

- INSERT INTO ACTOR VALUES ('1', 'TAYLOR KITSCH','M');
- INSERT INTO ACTOR VALUES ('2', 'SRK', 'M');
- INSERT INTO ACTOR VALUES ('3', 'JAMES STEWART','M');
- INSERT INTO ACTOR VALUES ('4', 'PETER LORRE', 'M');
- INSERT INTO ACTOR VALUES ('5', 'KATRINA KAIF', 'F');
- INSERT INTO ACTOR VALUES ('6', 'KAJOL', 'F');
- INSERT INTO ACTOR VALUES ('7', 'PUNITH', 'M');
- INSERT INTO ACTOR VALUES ('8', 'MERYL STREEP', 'F');

ACTOR

ACT_I	ACT_NAME	A
1	TAYLOR KITSCH	M
3	JAMES STEWART	M
4	PETER LORRE	M
5	KATRINA KAIF	F
6	KAJOL	F
7	PUNITH	M
8	MERYL STREEP	F

- INSERT INTO DIRECTOR VALUES ('101', 'PETER BERG',NULL);
- INSERT INTO DIRECTOR VALUES ('202', 'FARAN',9876543211);
- INSERT INTO DIRECTOR VALUES ('303', 'KARAN JOHAR',1123456789);
- INSERT INTO DIRECTOR VALUES ('404', 'YASH CHOPRA', 5678912345);
- INSERT INTO DIRECTOR VALUES ('505', 'HITCHCOCK',9876543212);
- INSERT INTO DIRECTOR VALUES ('606', 'MAHESH BABU',9876512345);
- INSERT INTO DIRECTOR VALUES ('707', 'STEVEN SPEILSBERG',1234567899);

DIRECTOR

```
SQL> SELECT * FROM DIRECTOR;
```

DIR_I	DIR_NAME	DIR_PHONE
-----	-----	-----
202	FARAN	9876543211
303	KARAN JOHAR	1123456789
404	YASH CHOPRA	5678912345
505	HITCHCOCK	9876543212
606	MAHESH BABU	9876512345
707	STEVEN SPEILSBERG	1234567899
101	PETER BERG	

- INSERT INTO MOVIES VALUES ('1', 'BATTELSHIP',2012,'ENGLISH','101');
- INSERT INTO MOVIES VALUES ('2', 'KKHH',1998,'HINDI','303');
- INSERT INTO MOVIES VALUES ('3', 'KKKG',2001,'HINDI','303');
- INSERT INTO MOVIES VALUES ('4', 'ROPE',1948,'ENGLISH','505');
- INSERT INTO MOVIES VALUES ('5', 'SECRET AGENT',1936,'ENGLISH','505');
- INSERT INTO MOVIES VALUES ('6', 'JTHJ',2012,'HINDI','404');
- INSERT INTO MOVIES VALUES ('7', 'DDLJ',1995,'HINDI','404');
- INSERT INTO MOVIES VALUES ('8', 'AAKASH',2005,'KANNADA','606');
- INSERT INTO MOVIES VALUES ('9', 'ARASU',2007,'KANNADA','606');
- INSERT INTO MOVIES VALUES ('10', 'THE POST',2017,'ENGLISH','707');

MOVIES

MOV_I	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1	BATTELSHIP	2012	ENGLISH	101
2	KKHH	1998	HINDI	303
3	KKKG	2001	HINDI	303
4	ROPE	1948	ENGLISH	505
5	SECRET AGENT	1936	ENGLISH	505
6	JTHJ	2012	HINDI	404
7	DDLJ	1995	HINDI	404
8	AAKASH	2005	KANNADA	606
9	ARASU	2007	KANNADA	606
10	THE POST	2017	ENGLISH	707

- INSERT INTO MOVIE_CAST VALUES ('1','1','HERO');
- INSERT INTO MOVIE_CAST VALUES ('2','2','HERO');
- INSERT INTO MOVIE_CAST VALUES ('2','3','HERO');
- INSERT INTO MOVIE_CAST VALUES ('3','4','HERO');
- INSERT INTO MOVIE_CAST VALUES ('4','5','HERO');
- INSERT INTO MOVIE_CAST VALUES ('5','6','HEROINE');
- INSERT INTO MOVIE_CAST VALUES ('6','7','HEROINE');
- INSERT INTO MOVIE_CAST VALUES ('7','8','HERO');
- INSERT INTO MOVIE_CAST VALUES ('7','9','HERO');
- INSERT INTO MOVIE_CAST VALUES ('8','10','HERO');

MOVIES_CAST

```
SQL> SELECT * FROM MOVIE_CAST;
```

ACT_I	MOV_I	ROLE
-----	-----	-----
1	1	HERO
3	4	HERO
4	5	HERO
5	6	HEROINE
6	7	HEROINE
7	8	HERO
7	9	HERO
8	10	HERO

- INSERT INTO RATING VALUES ('1',2);
- INSERT INTO RATING VALUES ('1',5);
- INSERT INTO RATING VALUES ('2',5);
- INSERT INTO RATING VALUES ('3',5);
- INSERT INTO RATING VALUES ('4',3);
- INSERT INTO RATING VALUES ('5',3);
- INSERT INTO RATING VALUES ('6',2);
- INSERT INTO RATING VALUES ('6',5);
- INSERT INTO RATING VALUES ('7',5);
- INSERT INTO RATING VALUES ('10',1);

RATING

```
SQL> SELECT * FROM RATING;
```

<u>MOV_I</u>	<u>REV_STARS</u>
1	5
1	2
2	5
3	5
4	3
5	3
6	2
7	5
10	1

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

```
SQL>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 D.DIR_ID = M.DIR_ID  
AND  
DIR_NAME = 'HITCHCOCK';
```

```
SQL>SELECT MOV_ID, MOV_TITLE  
FROM MOVIES NATURAL JOIN DIRECTOR  
WHERE DIR_NAME = 'HITCHCOCK';  
OR
```

```
SELECT MOV_ID, MOV_TITLE  
FROM MOVIES INNER JOIN DIRECTOR  
USING (DIR_ID)  
WHERE DIR_NAME = 'HITCHCOCK';
```

MOV_I	MOV_TITLE
4	ROPE
5	SECRET AGENT

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

- **SQL**>SELECT MOV_TITLE
FROM MOVIES
WHERE MOV_ID IN
(SELECT MOV_ID
FROM MOVIE_CAST
WHERE ACT_ID IN
(
SELECT ACT_ID
FROM MOVIE_CAST
GROUP BY ACT_ID
HAVING COUNT(*)>=2
)
);

MOV_TITLE

KKKG

AAKASH

DDLJ

ARASU

KKHH

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

- **SQL**> SELECT ACT_NAME
FROM ACTOR
WHERE ACT_ID IN
(
SELECT ACT_ID
FROM MOVIES M, MOVIE_CAST MC
WHERE M.MOV_ID = MC.MOV_ID
AND
MOV_YEAR < 2000
UNION
SELECT ACT_ID
FROM MOVIES M, MOVIE_CAST MC
WHERE M.MOV_ID = MC.MOV_ID
AND
MOV_YEAR > 2015
);

//Gives partial output

```
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 MC.MOV_ID = M.MOV_ID
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

OR

```
SELECT A.ACT_NAME, M.MOV_TITLE, M.MOV_YEAR
FROM ACTOR A, MOVIE_CAST MC, MOVIES M
WHERE A.ACT_ID = MC.ACT_ID
AND MC.MOV_ID = M.MOV_ID
AND M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

ACT_NAME	MOV_TITLE	MOV_YEAR
SRK	KKHH	1998
SRK	DDLJ	1995
JAMES STEWART	ROPE	1948
PETER LORRE	SECRET AGENT	1936
KAJOL	DDLJ	1995
MERYL STREEP	THE POST	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.

- **SQL>** SELECT MOV_TITLE, MAX(REV_STARS)
FROM MOVIES M, RATING R WHERE
M.MOV_ID=R.MOV_ID GROUP
BY(MOV_TITLE)
ORDER BY MOV_TITLE;

OR

SQL> 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;

MOV_TITLE	MAX<REV_STARS>
BATTLESHIP	5
DDLJ	5
JTHJ	5
KKHH	5
KKKG	5
ROPE	3
SECRET AGENT	3
THE POST	1

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

- **SQL>** 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
SPEILSBERG')));

1 row updated.

SQL> SELECT *FROM RATING;

MOV_I	REV_STARS
1	2
2	5
3	5
4	3
5	3
7	5
10	5

7 rows selected.