

**NAME : LIKITHA B**

**USN : 1BM19CS079**

**DATE : 09/06/21**

**DBMS LAB PROGRAM - 6**

**MOVIE DATABASE**

**QUESTION :**

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 CODE :**

```
CREATE DATABASE MOVIE;
USE MOVIE;
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 REAL,
```

```
PRIMARY KEY (DIR_ID));
```

```
CREATE TABLE MOVIES (  
MOV_ID INT,  
MOV_TITLE VARCHAR (25),  
MOV_YEAR INT,  
MOV_LANG VARCHAR (12),  
DIR_ID INT,  
PRIMARY KEY (MOV_ID),  
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));
```

```
CREATE TABLE MOVIE_CAST (  
ACT_ID INT,  
MOV_ID INT,  
ROLE VARCHAR(10),  
PRIMARY KEY (ACT_ID, MOV_ID),  
FOREIGN KEY (ACT_ID) REFERENCES ACTOR (ACT_ID),  
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));
```

```
CREATE TABLE RATING (  
MOV_ID INT,  
REV_STARS VARCHAR (25),  
PRIMARY KEY (MOV_ID),  
FOREIGN KEY (MOV_ID) REFERENCES MOVIES  
(MOV_ID)); SHOW TABLES;
```

```
INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');  
INSERT INTO ACTOR VALUES (302,'PRABHAS','M');  
INSERT INTO ACTOR VALUES (303,'PUNITH','M');  
INSERT INTO ACTOR VALUES (304,'JERMY','M');  
SELECT * FROM ACTOR;
```

```
INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 8751611001);  
INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 7766138911);  
INSERT INTO DIRECTOR VALUES (62,'FARAN', 9986776531);  
INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG',  
8989776530); SELECT * FROM DIRECTOR;
```

```
INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, 'TELUGU', 60);  
INSERT INTO MOVIES VALUES (1002,'BAHUBALI-1', 2015,'TELUGU', 60);  
INSERT INTO MOVIES VALUES (1003,'AKASH', 2008,'KANNADA', 61);  
INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);  
SELECT * FROM MOVIES;
```

```
INSERT INTO MOVIE_CAST VALUES (301, 1002, 'HEROINE');  
INSERT INTO MOVIE_CAST VALUES (301, 1001, 'HEROINE');  
INSERT INTO MOVIE_CAST VALUES (303, 1003, 'HERO');  
INSERT INTO MOVIE_CAST VALUES (303, 1002, 'GUEST');
```

```
INSERT INTO MOVIE_CAST VALUES (304, 1004,  
'HERO'); SELECT * FROM MOVIE_CAST;
```

```
INSERT INTO RATING VALUES (1001,'4');  
INSERT INTO RATING VALUES (1002,'2');  
INSERT INTO RATING VALUES (1003, '5');  
INSERT INTO RATING VALUES (1004, '4');  
SELECT * FROM RATING;
```

```
/*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');
```

```
/*2. 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)  
GROUP BY MOV_TITLE HAVING COUNT(MOV_TITLE)>1;
```

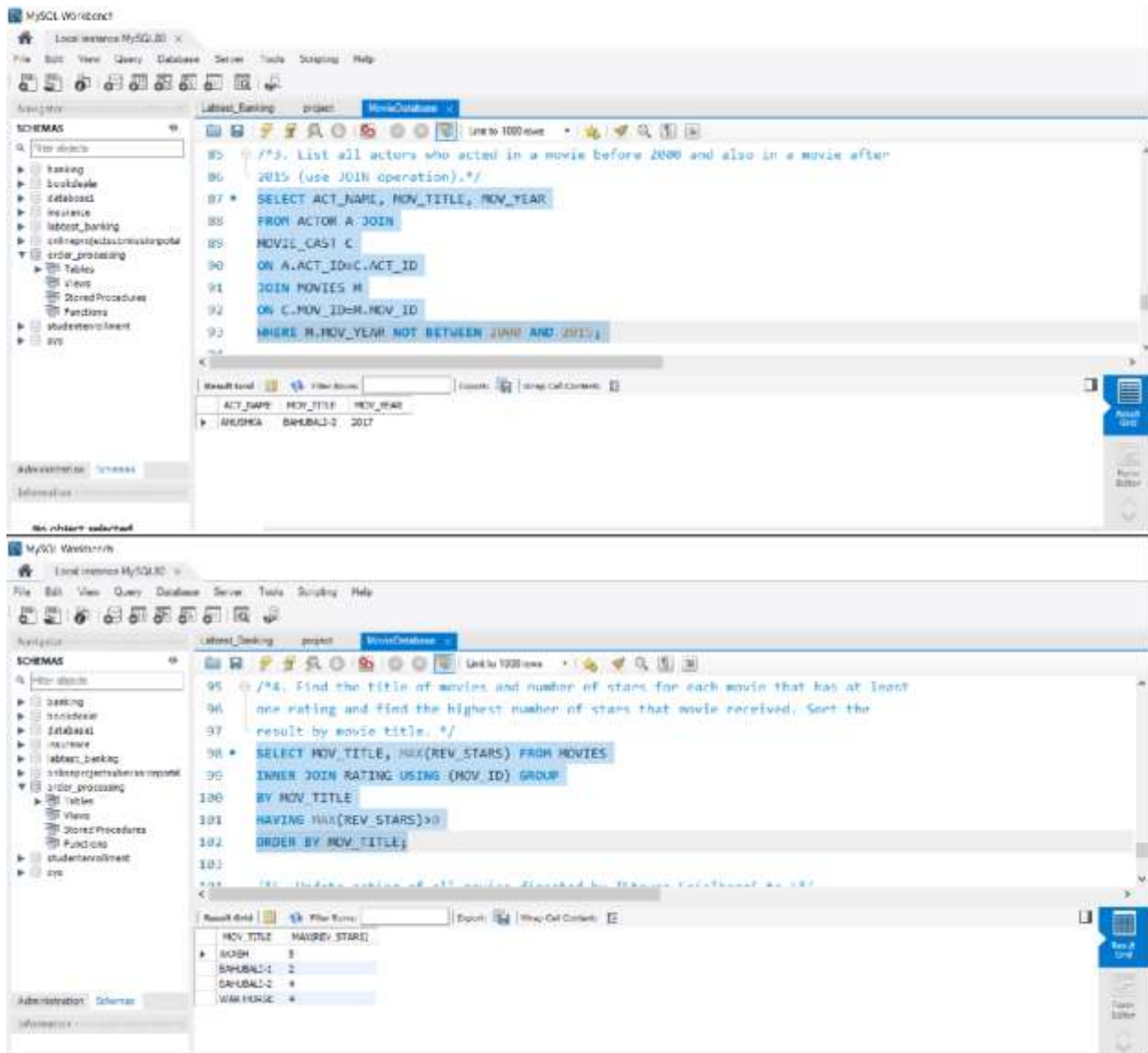
```
/*3. LIST ALL ACTORS WHO ACTED IN A MOVIE BEFORE 2000 AND ALSO IN A MOVIE  
AFTER 2015 (USE JOIN OPERATION).*/  
SELECT ACT_NAME, MOV_TITLE, MOV_YEAR  
FROM ACTOR A JOIN  
MOVIE_CAST C  
ON A.ACT_ID=C.ACT_ID  
JOIN MOVIES M  
ON C.MOV_ID=M.MOV_ID  
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

```
/*4. FIND THE TITLE OF MOVIES AND NUMBER OF STARS FOR EACH MOVIE THAT HAS AT  
LEAST ONE RATING AND FIND THE HIGHEST NUMBER OF STARS THAT MOVIE RECEIVED. SORT  
THE RESULT BY MOVIE TITLE. */  
SELECT MOV_TITLE, MAX(REV_STARS) FROM  
MOVIES INNER JOIN RATING USING (MOV_ID)  
GROUP BY MOV_TITLE  
HAVING MAX(REV_STARS)>0  
ORDER BY MOV_TITLE;
```

```
/*5. UPDATE RATING OF ALL MOVIES DIRECTED BY 'STEVEN SPIELBERG' TO 5*/  
UPDATE RATING  
SET REV_STARS='5'  
WHERE MOV_ID = (SELECT MOV_ID FROM MOVIES  
WHERE DIR_ID = (SELECT DIR_ID FROM DIRECTOR
```

```
WHERE DIR_NAME ='STEVEN
SPIELBERG')));
SELECT * FROM RATING;
```

## SCREENSHOTS OF THE PROGRAM OUTPUT :



The first screenshot shows a MySQL Workbench window with a query editor. The query is as follows:

```

--3. List all actors who acted in a movie before 2000 and also in a movie after
2015 (use JOIN operation).*/
3 * SELECT ACT_NAME, MOV_TITLE, MOV_YEAR
4 FROM ACTOR A JOIN
5 MOVIE_CAST C
6 ON A.ACT_ID=C.ACT_ID
7 JOIN MOVIES M
8 ON C.MOV_ID=M.MOV_ID
9 WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

The result set shows the following data:

ACT_NAME	MOV_TITLE	MOV_YEAR
ARONKA	SAHUBAL-2	2017

The second screenshot shows a MySQL Workbench window with a query editor. The query is as follows:

```

--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. */
4 * SELECT MOV_TITLE, MAX(REV_STARS) FROM MOVIES
5 INNER JOIN RATING USING (MOV_ID) GROUP
6 BY MOV_TITLE
7 HAVING MAX(REV_STARS)>0
8 ORDER BY MOV_TITLE;
```

The result set shows the following data:

MOV_TITLE	MAX(REV_STARS)
BOOM	5
SAHUBAL-1	2
SAHUBAL-2	4
WAK HAKSE	4

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest\_Ranking project MovieDatabase

SCHEMAS

Filter objects

- banking
- bookstore
- database1
- insurance
- latest\_booking
- onlineprojectsubmissionsportal
- order\_processing
- Tables
- Views
- Stored Procedures
- Functions
- studentenrollment
- xyz

102 ORDER BY MOV\_TITLE;

103

104 /\*% Update rating of all movies directed by 'Steven Spielberg' to 5\*/

105 \* UPDATE RATING

106 SET REV\_STARS=5

107 WHERE MOV\_ID = (SELECT MOV\_ID FROM MOVIES

108 WHERE DIR\_ID = (SELECT DIR\_ID FROM DIRECTOR

109 WHERE DIR\_NAME = 'STEVEN SPIELBERG'));

110 \* select \* from rating;

111

Result Grid

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	1
1005	1

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest\_Ranking project MovieDatabase

SCHEMAS

Filter objects

- banking
- bookstore
- database1
- insurance
- latest\_booking
- onlineprojectsubmissionsportal
- order\_processing
- Tables
- Views
- Stored Procedures
- Functions
- studentenrollment
- xyz

28 PRIMARY KEY (ACT\_ID, MOV\_ID),

29 FOREIGN KEY (ACT\_ID) REFERENCES ACTOR (ACT\_ID),

30 FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

31

32 CREATE TABLE RATING (

33 MOV\_ID INT,

34 REV\_STARS VARCHAR (25),

35 PRIMARY KEY (MOV\_ID),

36 FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

37

Result Grid

Tables in movie
actor
director
movie_said
movies
rating

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest\_Ranking project MovieDatabase

SCHEMAS

Filter objects

- banking
- bookstore
- database1
- insurance
- latest\_booking
- onlineprojectsubmissionsportal
- order\_processing
- Tables
- Views
- Stored Procedures
- Functions
- studentenrollment
- xyz

37 \* show tables;

38

39 \* INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');

40 INSERT INTO ACTOR VALUES (302,'PRABHAS','M');

41 INSERT INTO ACTOR VALUES (303,'PUNITH','M');

42 \* INSERT INTO ACTOR VALUES (304,'JEREMY','M');

43 \* SELECT \* FROM ACTOR;

44

45 INSERT INTO DIRECTOR VALUES (68,'RAJAMOULI',8751613081);

46

Result Grid

ACT_ID	ACT_NAME	ACT_GENDER
301	ANUSHKA	F
302	PRABHAS	M
303	PUNITH	M
304	JEREMY	M

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest\_Sinking project MovieDatabase

SCHEMAS

46 INSERT INTO DIRECTOR VALUES (61, 'HITCHCOCK', 7766138911);

47 \* INSERT INTO DIRECTOR VALUES (62, 'FARAN', 9986776531);

48 \* INSERT INTO DIRECTOR VALUES (63, 'STEVEN SPIELBERG', 8989776530);

49 \* SELECT \* FROM DIRECTOR;

50

51 INSERT INTO MOVIES VALUES (1001, 'DARKBALL-2', 2017, 'TELUGU', 60);

52 \* INSERT INTO MOVIES VALUES (1002, 'DARKBALL-1', 2015, 'TELUGU', 60);

53 \* INSERT INTO MOVIES VALUES (1003, 'AKASHI', 2000, 'KANNADA', 61);

54 \* INSERT INTO MOVIES VALUES (1004, 'WAR HEROSE', 2011, 'ENGLISH', 63);

55 \* SELECT \* FROM MOVIES;

Result Grid

DID_ID	DOR_NAME	DOR_PHONE
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest\_Sinking project MovieDatabase

SCHEMAS

52 \* INSERT INTO MOVIES VALUES (1002, 'DARKBALL-1', 2015, 'TELUGU', 60);

53 \* INSERT INTO MOVIES VALUES (1003, 'AKASHI', 2000, 'KANNADA', 61);

54 \* INSERT INTO MOVIES VALUES (1004, 'WAR HEROSE', 2011, 'ENGLISH', 63);

55 \* SELECT \* FROM MOVIES;

56

57 \* INSERT INTO MOVIE\_CAST VALUES (301, 1002, 'HEROINE');

58 \* INSERT INTO MOVIE\_CAST VALUES (301, 1001, 'HEROINE');

59 \* INSERT INTO MOVIE\_CAST VALUES (303, 1003, 'HERO');

60 \* INSERT INTO MOVIE\_CAST VALUES (303, 1002, 'QUEST');

61 \* INSERT INTO MOVIE\_CAST VALUES (304, 1004, 'HERO');

Result Grid

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANGUAGE	MOV_GENRE
1001	DARKBALL-2	2017	TELUGU	60
1002	DARKBALL-1	2015	TELUGU	60
1003	AKASHI	2000	KANNADA	61
1004	WAR HEROSE	2011	ENGLISH	63

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest\_Sinking project MovieDatabase

SCHEMAS

61 \* INSERT INTO MOVIE\_CAST VALUES (304, 1004, 'HERO');

62 \* SELECT \* FROM MOVIE\_CAST;

63

64 \* INSERT INTO RATING VALUES (1001, '4');

65 \* INSERT INTO RATING VALUES (1002, '2');

66 \* INSERT INTO RATING VALUES (1003, '5');

67 \* INSERT INTO RATING VALUES (1004, '4');

68 \* SELECT \* FROM RATING;

69

70 \* SELECT \* FROM MOVIE\_CAST;

Result Grid

ACT_ID	MOV_ID	ROLE
301	1001	HEROINE
301	1002	HEROINE
303	1002	QUEST
303	1003	HERO
304	1004	HERO

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest Running project MovieDatabase

SCHEMAS

- MovieDatabase
  - Tables
    - MOVIES
    - DIRECTOR
    - MOVIE\_CAST
    - RATING

61 \* INSERT INTO MOVIE\_CAST VALUES (304, 1884, 'HERO')  
 62 \*  
 63  
 64 \* INSERT INTO RATING VALUES (1001, '4')  
 65 \* INSERT INTO RATING VALUES (1002, '2')  
 66 \* INSERT INTO RATING VALUES (1003, '5')  
 67 \* INSERT INTO RATING VALUES (1004, '4')  
 68 \* SELECT \* FROM RATING  
 69

Result Grid

MOV_ID	RAT Stars
1001	4
1002	2
1003	5
1004	4

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest Running project MovieDatabase

SCHEMAS

- MovieDatabase
  - Tables
    - MOVIES
    - DIRECTOR
    - MOVIE\_CAST
    - RATING

70 /\*1. list the titles of all movies directed by 'Hitchcock'.\*/  
 71 \* SELECT MOV\_TITLE  
 72 FROM MOVIES  
 73 WHERE DIR\_ID IN (SELECT DIR\_ID  
 74 FROM DIRECTOR  
 75 WHERE DIR\_NAME = 'HITCHCOCK')  
 76  
 77 /\*2. Find the movie names where one or more actors acted in two or more movies.\*/  
 78 \* SELECT MOV\_TITLE  
 79 FROM MOVIES M, MOVIE\_CAST MC  
 80 WHERE M.MOV\_ID=MC.MOV\_ID AND ACT\_ID IN (SELECT ACT\_ID  
 81 FROM MOVIE\_CAST GROUP BY ACT\_ID HAVING  
 82 COUNT(ACT\_ID)>1)  
 83 GROUP BY MOV\_TITLE HAVING COUNT(MOV\_TITLE)>1  
 84

Result Grid

MOV_TITLE
TOUCH

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Latest Running project MovieDatabase

SCHEMAS

- MovieDatabase
  - Tables
    - MOVIES
    - DIRECTOR
    - MOVIE\_CAST
    - RATING

76  
 77 /\*2. Find the movie names where one or more actors acted in two or more movies.\*/  
 78 \* SELECT MOV\_TITLE  
 79 FROM MOVIES M, MOVIE\_CAST MC  
 80 WHERE M.MOV\_ID=MC.MOV\_ID AND ACT\_ID IN (SELECT ACT\_ID  
 81 FROM MOVIE\_CAST GROUP BY ACT\_ID HAVING  
 82 COUNT(ACT\_ID)>1)  
 83 GROUP BY MOV\_TITLE HAVING COUNT(MOV\_TITLE)>1  
 84  
 85 /\*3. list all actors who acted in a movie before 2010 and also in a movie after  
 86 2010\*/  
 87 \* SELECT ACT\_NAME  
 88 FROM MOVIE\_CAST  
 89 WHERE MOV\_ID IN (SELECT MOV\_ID  
 90 FROM MOVIES  
 91 WHERE YEAR < 2010)  
 92 AND MOV\_ID IN (SELECT MOV\_ID  
 93 FROM MOVIES  
 94 WHERE YEAR > 2010)  
 95 GROUP BY ACT\_NAME  
 96

Result Grid

MOV_TITLE
SALESLAU-1

**\*\* END OF LAB PROGRAM WEEK - 6\*\***