

NAME : KARTHIK S
USN : 1BM19CS070
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 screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of databases including 'banking', 'bookdealer', 'database1', 'insurance', 'labtest_banking', 'onlineprojectsubmissionportal', 'order_processing', 'Tables', 'Views', 'Stored Procedures', 'Functions', 'studentenrollment', and 'sys'. The main editor window contains a SQL query:

```
85 /*3. List all actors who acted in a movie before 2000 and also in a movie after  
86 2015 (use JOIN operation).*/  
87 • SELECT ACT_NAME, MOV_TITLE, MOV_YEAR  
88 FROM ACTOR A JOIN  
89 MOVIE_CAST C  
90 ON A.ACT_ID=C.ACT_ID  
91 JOIN MOVIES M  
92 ON C.MOV_ID=M.MOV_ID  
93 WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;
```

Below the query, the 'Result Grid' is displayed with the following data:

ACT_NAME	MOV_TITLE	MOV_YEAR
ANUSHKA	BAHUBALI-2	2017

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view of databases including 'banking', 'bookdealer', 'database1', 'insurance', 'labtest_banking', 'onlineprojectsubmissionportal', 'order_processing', 'Tables', 'Views', 'Stored Procedures', 'Functions', 'studentenrollment', and 'sys'. The main editor window contains a SQL query:

```
95 /*4. Find the title of movies and number of stars for each movie that has at least  
96 one rating and find the highest number of stars that movie received. Sort the  
97 result by movie title. */  
98 • SELECT MOV_TITLE, MAX(REV_STARS) FROM MOVIES  
99 INNER JOIN RATING USING (MOV_ID) GROUP  
100 BY MOV_TITLE  
101 HAVING MAX(REV_STARS)>0  
102 ORDER BY MOV_TITLE;  
103  
104 /*5. Update rating of all movies directed by 'Steven Spielberg' to 5*/
```

Below the query, the 'Result Grid' is displayed with the following data:

MOV_TITLE	MAX(REV_STARS)
AKASH	5
BAHUBALI-1	2
BAHUBALI-2	4
WAR HORSE	4

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase x

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
- Tables
- Views
- Stored Procedures
- Functions
- studentenrollment
- sys

102 ORDER BY MOV_TITLE;

103

104 /*5. 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	5

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase x

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
- Tables
- Views
- Stored Procedures
- Functions
- studentenrollment
- sys

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_cast
movies
rating

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase x

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
- Tables
- Views
- Stored Procedures
- Functions
- studentenrollment
- sys

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,'JERMY','M');

43 • SELECT * FROM ACTOR;

44

45 INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 8751611001);

46 INSERT INTO DIRECTOR VALUES (61,'VITCHACK', 7766120011);

Result Grid

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

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase x

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- studentenrollment
- sys

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,'BAHUBALI-2', 2017, 'TELUGU', 60);

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

53 INSERT INTO MOVIES VALUES (1003,'AKASH', 2008, 'KANNADA', 61);

54 INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);

55 SELECT * FROM MOVIES;

Result Grid

DIR_ID	DIR_NAME	DIR_PHONE
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase x

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- studentenrollment
- sys

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

53 INSERT INTO MOVIES VALUES (1003,'AKASH', 2008, 'KANNADA', 61);

54 INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 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, 'GUEST');

61 INSERT INTO MOVIE_CAST VALUES (304, 1004, 'HERO');

Result Grid

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1001	BAHUBALI-2	2017	TELUGU	60
1002	BAHUBALI-1	2015	TELUGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase x

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- studentenrollment
- sys

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

Result Grid

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

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- studentenrollment
- sys

Administration Schemas Information

61 • INSERT INTO MOVIE_CAST VALUES (304, 1004, 'HERO');

62 • SE Execute the selected portion of the script or everything, if there is no selection

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

Result Grid

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	4

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- studentenrollment
- sys

Administration Schemas Information

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 MV

Result Grid

MOV_TITLE
AKASH

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Labtest_Banking project MovieDatabase

SCHEMAS

Filter objects

- banking
- bookdealer
- database1
- insurance
- labtest_banking
- onlineprojectsubmissionportal
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- studentenrollment
- sys

Administration Schemas Information

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 MV

80 WHERE M.MOV_ID=MV.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 2000 and also in a movie after

Result Grid

MOV_TITLE
BAHUBALI-1

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