

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | **B.Tech** | Semester | : | **Winter 19 - 20** |
| Course | : | **Database Management Systems** | Code | : | **CSE2004** |
| Faculty | : | **Dr.Bhuvaneswari A&Dr.Ayesha Sheik** | Slot | : | **L27 + L28** |

Name**: Wilson Vidyut Doloy**

Registration No.**: 19BCE1603**

Ex. No. 6

17.01.2020

**SQL Correlated queries & views**

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 for the following

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.

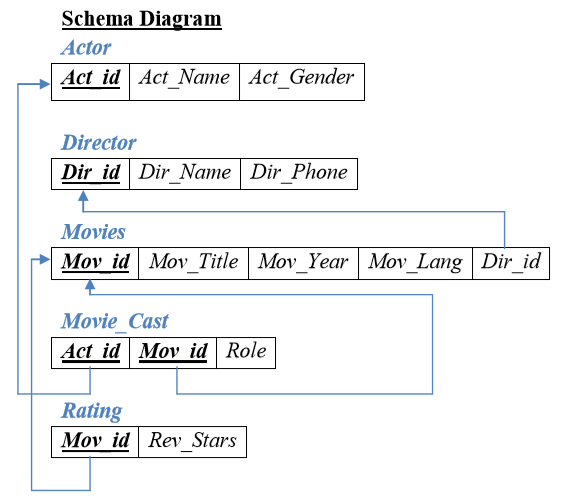
5. Update rating of all movies directed by ‘Steven Spielberg’ to rating value 5.

6. Create view“MovieDetails”to show the movie id, name, Movie language and its rating star released between Year 2015 to 2019.

7. Update the view “MovieDetails” to change movie language from “English” to “Hindi”

8. Delete rows in the view “MovieDetails” where the rating is 3.

9. Drop the view “MovieDetails”



CREATE TABLE ACTOR ( ACT\_ID NUMBER (3), ACT\_NAME VARCHAR (20), ACT\_GENDER CHAR (1), PRIMARY KEY (ACT\_ID));

CREATE TABLE DIRECTOR ( DIR\_ID NUMBER (3), DIR\_NAME VARCHAR (20), DIR\_PHONE NUMBER (10), PRIMARY KEY (DIR\_ID));

CREATE TABLE MOVIES ( MOV\_ID NUMBER (4), MOV\_TITLE VARCHAR (25), MOV\_YEAR NUMBER (4), MOV\_LANG VARCHAR (12), DIR\_ID NUMBER (3), PRIMARY KEY (MOV\_ID), FOREIGN KEY (DIR\_ID) REFERENCES DIRECTOR (DIR\_ID));

CREATE TABLE MOVIE\_CAST ( ACT\_ID NUMBER (3), MOV\_ID NUMBER (4), 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 NUMBER (4), REV\_STARS VARCHAR (25), PRIMARY KEY (MOV\_ID), FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

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');

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);

INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, 'TELAGU', 60);

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

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

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

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');

INSERT INTO RATING VALUES (1001, 4);

INSERT INTO RATING VALUES (1002, 2);

INSERT INTO RATING VALUES (1003, 5);

INSERT INTO RATING VALUES (1004, 4);

1. select mov\_title from movies where dir\_id=(select dir\_id from director where dir\_name='HITCHCOCK');

MOV\_TITLE

-------------------------

AKASH

1. 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(\*)>1;

MOV\_TITLE

-------------------------

BAHUBALI-1

1. select \* from actor a,movies m,movie\_cast mc where a.act\_id=mc.act\_id and m

c.mov\_id=m.mov\_id and (m.mov\_year < 2000 or m.mov\_year>2015);

ACT\_ID ACT\_NAME A MOV\_ID MOV\_TITLE

---------- -------------------- - ---------- -------------------------

MOV\_YEAR MOV\_LANG DIR\_ID ACT\_ID MOV\_ID ROLE

---------- ------------ ---------- ---------- ---------- ----------

301 ANUSHKA F 1001 BAHUBALI-2

2017 ENGLISH 60 301 1001 HEROINE

1. 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)

------------------------- -------------------------

AKASH 5

BAHUBALI-1 2

BAHUBALI-2 4

WAR HORSE 4

1. 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'));

1 row updated.

select \* from rating;

MOV\_ID REV\_STARS

---------- -------------------------

1001 4

1002 2

1003 5

1004 5

1. create view moviedetails as (select movies.mov\_id,movies.mov\_title,movies.mov\_lang,rating.rev\_stars from movies,rating where movies.mov\_id=rating.mov\_id);

View created.

select \* from moviedetails;

MOV\_ID MOV\_TITLE MOV\_LANG REV\_STARS

---------- ------------------------- ------------ -------------------------

1001 BAHUBALI-2 TELAGU 4

1002 BAHUBALI-1 TELAGU 2

1003 AKASH KANNADA 5

1004 WAR HORSE ENGLISH 5

1. update moviedetails set mov\_lang='ENGLISH';

4 rows updated.

select \* from moviedetails;

MOV\_ID MOV\_TITLE MOV\_LANG REV\_STARS

---------- ------------------------- ------------ -------------------------

1001 BAHUBALI-2 ENGLISH 4

1002 BAHUBALI-1 ENGLISH 2

1003 AKASH ENGLISH 5

1004 WAR HORSE ENGLISH 5

1. delete from moviedetails where rev\_stars=3;

0 rows deleted.

1. drop view moviedetails;

View dropped.