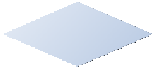
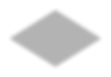
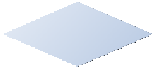
**Entity-Relationship Diagram**



**Schema Diagram**

***Actor***



***Director***

***Movie\_Cast***

***Rating***

***Mov\_id*** *Rev\_Stars*

|  |  |  |
| --- | --- | --- |
| ***Act\_id*** | *Act\_Name* | *Act\_Gender* |

|  |  |  |
| --- | --- | --- |
| ***Dir\_id*** | *Dir\_Name* | *Dir\_Phone* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Movies*** | | | | |  |
| ***Mov\_id*** | *Mov\_Title* | *Mov\_Year* | *Mov\_Lang* | *Dir\_id* | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Act\_id*** | ***Mov\_id*** | *Role* |

-- CREATION OF TABLES

-- ACTOR

CREATE TABLE ACTOR (

ACT\_ID INTEGER,

ACT\_NAME VARCHAR(20),

ACT\_GENDER CHAR (1),

PRIMARY KEY (ACT\_ID));

-- DIRECTOR

CREATE TABLE DIRECTOR (

DIR\_ID INTEGER,

DIR\_NAME VARCHAR (20),

DIR\_PHONE INTEGER,

PRIMARY KEY (DIR\_ID));

-- MOVIES

CREATE TABLE MOVIES (

MOV\_ID INTEGER,

MOV\_TITLE VARCHAR (25),

MOV\_YEAR INTEGER,

MOV\_LANG VARCHAR (12),

DIR\_ID INTEGER,

PRIMARY KEY (MOV\_ID),

FOREIGN KEY (DIR\_ID) REFERENCES DIRECTOR (DIR\_ID));

-- MOVIE\_CAST

CREATE TABLE MOVIE\_CAST (

ACT\_ID INTEGER,

MOV\_ID INTEGER,

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));

-- RATING

CREATE TABLE RATING (

MOV\_ID INTEGER,

REV\_STARS VARCHAR (25),

PRIMARY KEY (MOV\_ID),

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

-- VALUES FOR TABLES

-- ACTOR

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');

-- DIRECTOR

INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 875161100);

INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 776613891);

INSERT INTO DIRECTOR VALUES (62,'FARAN', 998677653);

INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', 898977653);

-- MOVIES

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);

-- MOVIE\_CAST

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');

-- RATING

INSERT INTO RATING VALUES (1001, 4);

INSERT INTO RATING VALUES (1002, 2);

INSERT INTO RATING VALUES (1003, 5);

INSERT INTO RATING VALUES (1004, 4);

-- DISPLAYING TABLES

-- ACTOR

SELECT \* FROM ACTOR;

-- DIRECTOR

SELECT \* FROM DIRECTOR;;

-- MOVIES

SELECT \* FROM MOVIES;

-- MOVIE\_CAST

SELECT \* FROM MOVIE\_CAST;

-- RATING

SELECT \* FROM RATING;

-- 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');

-- 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(\*) > 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;

-- OR

SELECT A.ACT\_NAME, A.ACT\_NAME, C.MOV\_TITLE, C.MOV\_YEAR

FROM ACTOR A, MOVIE\_CAST B, MOVIES C

WHERE A.ACT\_ID=B.ACT\_ID

AND B.MOV\_ID=C.MOV\_ID

AND C.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 5KL

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'));