Ex. No : 9

Implementation of set operators nested queries, and join queries

CO2: Construct queries using SQL for database creation, interaction, modification, and updation. (Cognitive Knowledge Level: Apply)

AIM

Consider the schema for MovieDatabase:

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 moremovies.
3. List all actors who acted in a movie before 2000 and also in a movieafter 2015 (use JOINoperation).
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’ to5.

Table Creation

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)); Act_Gender Mov_Year Mov_Lang Dir_id FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));

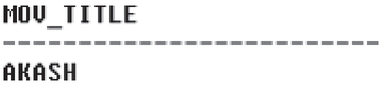
```
CREATE TABLE MOVIE_CAST ( ACT_ID NUMBER (3), MOV_ID NUMBER(4), ROLE
VARCHAR(10), PRIMARY KEY (ACT_ID, MOV_ID), FOREIGN KEY (ACT_ID)
REFERENCES ACTOR (ACT_ID), FOREIGN KEY (MOV_ID) REFERENCES MOVIES
(MOV_ID));

CREATE TABLE RATING ( MOV_ID NUMBER (4), REV_STARS VARCHAR (25),
PRIMARY KEY (MOV_ID), FOREIGN KEY (MOV_ID) REFERENCES MOVIES
(MOV_ID));
```

Insertion of Values to 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‘);

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);
```

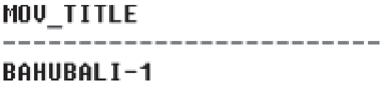
1. List the titles of all movies directed by‘Hitchcock’.

```
SELECT MOV_TITLE FROM MOVIES WHERE DIR_ID IN (SELECT DIR_ID FROM
DIRECTOR WHERE DIR_NAME = _HITCHCOCK‘);
```



```
MOV_TITLE
-------------------------
AKASH
```

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

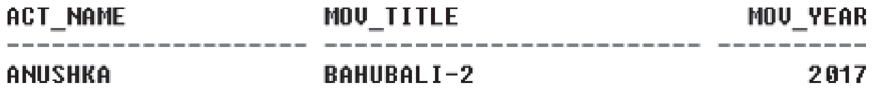


```
MOV_TITLE
-------------------------
BAHUBALI-1
```

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

```
SELECT ACT_NAME, MOV_TITLE, MOV_YEAR

FROM ACTOR A JOIN

MOVIE_CASTC

 ON A.ACT_ID=C.ACT_ID

JOIN MOVIESM

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;



```
ACT_NAME             MOV_TITLE                  MOV_YEAR
-------------------- ------------------------- ----------
ANUSHKA              BAHUBALI-2                      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.

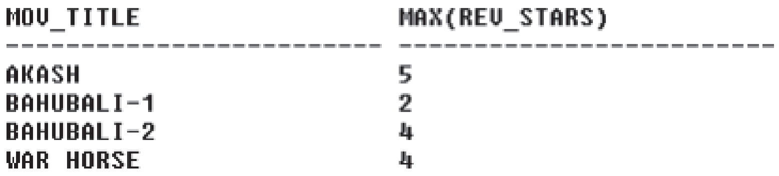
SELECT MOV_TITLE, MAX (REV_STARS)

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
```

5. Update rating of all movies directed by ‘Steven Spielberg’ to5 KL

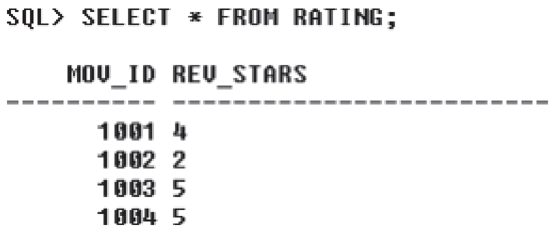
UPDATE RATING

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‘));



```
SQL> SELECT * FROM RATING;

    MOV_ID REV_STARS
---------- -------------------------
      1001 4
      1002 2
      1003 5
      1004 5
```