LAB REPORT

Lab 4
Advanced Data Manipulation

CSE 4308
DATABASE MANAGEMENT SYSTEMS LAB

NAME: CHOWDHURY ASHFAQ
STUDENT ID: 200042123
PROGRAM: SWE
GROUP: 1A
DATE: 18/09/22

Tasks:

1. Find the name of the actors/actresses that are also directors (with and without 'intersect' clause).
2. Find the list of all the first names stored in the database.
3. Find the movie titles that did not receive any ratings (with and without 'minus' clause).
4. Find the average rating of all movies.
5. Find the minimum rating for each movie and display them in descending order of rating.
6. Find the last name of actors/actresses and the number of ratings received by the movies that they played a role in.
7. Find the last name and average runtime of movies of different actors/actresses. Do not include any actor/actress who worked with James Cameron' (with and without 'having' clause).
8. Find the first name and last name of the director of the movie having the highest average rating (with and without 'all' clause).
9. Find all the movie related information of movies acted and directed by the same person.
10. Find the title and average rating of the movies that have average rating more than 7 (with and without using 'having' clause).
11. Find the title of the movies having average rating higher than the average rating of all the movies.
12. Find the title and average rating of the movies without using the group by statement.
13. Find the actresses with the same first name.

14. Find the title and maximum rating of the movies that has at least 10 reviews and has a female actress. One of the reviewers of the movie should be 'Neal Wruck'. Do not include any movie that received less than 4 stars rating or any movies from directors that have did not direct more than one movie.

In this Lab Task we were given to perform various DDL (Data Definition Language) and DML (Data Modification Language) operations.

Analysis of the problem:

A movie.sql file was given which contained all the necessary tables and Primary Key and Foreign Key. From that database we had to extract different information using different queries and clauses like Having , minus, all etc.

Solution:

```
1) SELECT DISTINCT ACT_FIRSTNAME, ACT_LASTNAME FROM ACTOR, DIRECTOR
 WHERE ACTOR.ACT_FIRSTNAME= DIRECTOR.DIR_FIRSTNAME AND ACTOR.ACT_LASTNAME=DIRECTOR.DIR_LASTNAME;

 SELECT ACT_FIRSTNAME, ACT_LASTNAME FROM ACTOR
 INTERSECT
 SELECT DIR_FIRSTNAME,DIR_LASTNAME FROM DIRECTOR;

 2)SELECT DISTINCT ACT_FIRSTNAME AS FIRSTNAME FROM ACTOR
 UNION
 SELECT DISTINCT DIR_FIRSTNAME FROM DIRECTOR;

 3)SELECT MOV_TITLE
 FROM MOVIE,RATING
 WHERE MOVIE.MOV_ID=RATING.MOV_ID AND RATING.REV_STARS=NULL;

 SELECT MOV_TITLE
 FROM MOVIE
 MINUS
 SELECT MOV_TITLE
 FROM MOVIE, RATING
 WHERE MOVIE.MOV_ID = RATING.MOV_ID;


 4)  SELECT MOV_ID, AVG(REV_STARS)
 FROM RATING;
```

```
 5)SELECT MOV_TITLE,MIN(REV_STARS) AS RATE FROM RATING
 JOIN MOVIE ON MOVIE.MOV_ID=RATING.MOV_ID
 GROUP BY MOVIE.MOV_TITLE
 ORDER BY RATE DESC;



 6)SELECT ACT_LASTNAME, COUNT(REV_STARS) AS RATING_COUNT
 FROM ACTOR,CASTS,RATING
 WHERE ACTOR.ACT_ID= CASTS.ACT_ID AND CASTS.MOV_ID=RATING.MOV_ID
 GROUP BY ACTOR.ACT_LASTNAME;

 7)SELECT ACT_LASTNAME, AVG(MOV_TIME)
 FROM ACTOR,CASTS,MOVIE
 WHERE ACTOR.ACT_ID= CASTS.ACT_ID AND CASTS.MOV_ID=MOVIE.MOV_ID
 GROUP BY ACTOR.ACT_LASTNAME
 MINUS
 SELECT ACT_LASTNAME, AVG(MOVIE.MOV_TIME)
 FROM ACTOR, CASTS, MOVIE, DIRECTION, DIRECTOR
 WHERE ACTOR.ACT_ID = CASTS.ACT_ID AND CASTS.MOV_ID =
 MOVIE.MOV_ID AND MOVIE.MOV_ID = DIRECTION.MOV_ID AND
 DIRECTION.DIR_ID = DIRECTOR.DIR_ID AND
 DIRECTOR.DIR_FIRSTNAME = 'James' AND DIRECTOR.DIR_LASTNAME =
 'Cameron'
 GROUP BY ACTOR.ACT_LASTNAME;
```

```
8)SELECT MAX(DIR_FIRSTNAME) FIRSTNAME, MAX(DIR_LASTNAME)
LASTNAME, MAX(A.REVIEW_AVERAGE)
FROM DIRECTOR D1, DIRECTION D2, RATING R, (SELECT M.MOV_ID,
avg(REV_STARS) REVIEW_AVERAGE
FROM RATING R, MOVIE M
WHERE M.MOV_ID = R.MOV_ID
GROUP BY M.MOV_ID) A
WHERE (D1.DIR_ID = D2.DIR_ID) AND (D2.MOV_ID = R.MOV_ID);

SELECT MAX(ALL DIR_FIRSTNAME) FIRSTNAME, MAX(ALL
DIR_LASTNAME) LASTNAME, MAX(A.REVIEW_AVERAGE)
FROM DIRECTOR D1, DIRECTION D2, RATING R, (SELECT ALL
M.MOV_ID, avg(REV_STARS) REVIEW_AVERAGE
FROM RATING R, MOVIE M
WHERE M.MOV_ID = R.MOV_ID
GROUP BY M.MOV_ID) A
WHERE (D1.DIR_ID = D2.DIR_ID) AND (D2.MOV_ID = R.MOV_ID);


9)SELECT MOVIE.MOV_ID, MOVIE.MOV_TITLE, MOVIE.MOV_YEAR,
MOVIE.MOV_TIME, MOVIE.MOV_LANGUAGE, MOVIE.MOV_RELEASEDATE,
MOVIE.MOV_COUNTRY
FROM MOVIE, ACTOR, CASTS, DIRECTION, DIRECTOR
WHERE ACTOR.ACT_ID = CASTS.ACT_ID AND CASTS.MOV_ID =
MOVIE.MOV_ID AND MOVIE.MOV_ID = DIRECTION.MOV_ID AND
DIRECTION.DIR_ID = DIRECTOR.DIR_ID AND
DIRECTOR.DIR_FIRSTNAME = ACTOR.ACT_FIRSTNAME AND
DIRECTOR.DIR_LASTNAME = ACTOR.ACT_LASTNAME;
```

```
10)SELECT MOV_TITLE, AVG_RATING
FROM(SELECT MOV_TITLE, AVG(NVL(REV_STARS,0)) AS AVG_RATING
FROM MOVIE, RATING
WHERE MOVIE.MOV_ID = RATING.MOV_ID
GROUP BY MOVIE.MOV_TITLE)
WHERE AVG_RATING > 7;


SELECT MOV_TITLE, AVG(NVL(REV_STARS,0))
FROM MOVIE, RATING
WHERE MOVIE.MOV_ID = RATING.MOV_ID
GROUP BY MOVIE.MOV_TITLE
HAVING AVG(NVL(REV_STARS,0)) > 7;


11)SELECT MOV_TITLE, AVG(NVL(REV_STARS,0))
FROM MOVIE, RATING
WHERE MOVIE.MOV_ID = RATING.MOV_ID
GROUP BY MOVIE.MOV_TITLE
HAVING AVG(NVL(REV_STARS,0)) > (SELECT AVG(NVL(REV_STARS,0))
FROM RATING);

12)SELECT DISTINCT MOV_TITLE,(SELECT AVG(NVL(REV_STARS,0))
FROM RATING
WHERE MOVIE.MOV_ID = RATING.MOV_ID)
FROM MOVIE, RATING
WHERE MOVIE.MOV_ID = RATING.MOV_ID;


13)SELECT AC1.ACT_FIRSTNAME
FROM ACTOR AC1
INNER JOIN ACTOR AC2 ON AC1.ACT_FIRSTNAME =
AC2.ACT_FIRSTNAME
WHERE AC1.ACT_GENDER = 'F';
```

Interesting Findings:

- Learnt about intersect, union, minus and having.

Problems faced and solution:

- All of these things aren't taught previously and we don't even know what would be given in our next lab so it's very hard for us to solve them at lab without prior knowledge.