Michelle Skaf

Dealing With Data

June 23, 2021

SQL Assignment

SHOW databases;

USE imdb;

**#1. Use IMDB database to find how many actors are stored in the actors table**

SELECT count(\*) AS num\_factors

FROM actors;

**#2. In IMDB database, find information for all the actors with id 376249.**

SELECT \*

FROM actors

WHERE ID = 376249;

**#3. In the IMDB database, which actors played in the movie Ocean's Twelve**

SELECT movies.id

FROM movies

WHERE name = "Ocean's Twelve";

SELECT actor\_id

FROM roles

WHERE movie\_id ="238073"

Order by actor\_id;

SELECT first\_name, last\_name

FROM actors A

inner join roles R

inner join movies M

ON A.id = R.actor\_id and M.id = R.movie\_id

WHERE M.name = "Ocean's Twelve"

Order by first\_name;

**#4. In the IMDB database, find the movies with the word "Vietnam" in their title**

SELECT\*

FROM movies

WHERE name = "%Vietnam%";

**#5. In the IMDB database, find number of movies where each actors has played**

**#Show the actor ID and number of movies**

SELECT A.id as actor\_id, A.first\_name AS first\_name, A.last\_name AS last\_name,

COUNT(M.id) as Number\_of\_Movies

FROM actors A

Inner Join movies M

Inner Join roles R on A.id = R.actor\_id and M.id = R.movie\_id

GROUP by A.id

HAVING COUNT(M.id)

Order by A.id;

#How many movies brad pitt played in

SELECT count(\*)

FROM actors A

Inner Join movies M

Inner Join roles R

on M.id = R.movie\_id AND A.id = R.actor\_id

WHERE A.first\_name = 'Brad' AND last\_name ='Pitt';

#Movies in comedy and drama and year release date

SELECT MG1.movie\_id, MG1.genre, MG2.genre, M.year

FROM movies\_genres MG1

Inner join movies\_genres MG2 on MG1.movie\_id = MG2.movie\_id

Inner join movies M on M.id = MG1.movie\_id AND M.id = MG2.movie\_id

Where MG1.genre = 'Drama' AND MG2.genre = 'Comedy';

USE facebook;

SHOW tables;

**#6.In facebook database, find male students who are "Interested in" women, and are "LookingFor" or "Whatever I can get**"

SELECT\*

FROM Profiles;

SELECT\*

FROM Relationship;

SELECT\*

FROM LookingFor;

SELECT\*

FROM Orientation;

SELECT P.\*

FROM Profiles P

Inner join LookingFor LR

Inner join Orientation O

ON P.ProfileID = LR.ProfileID AND O.ProfileID = P.ProfileID

WHERE LR.LookingFor = 'Whatever I can get' and P.Sex = 'Male' AND O.InterestedIn = 'Women';

**#7. In facebook database, for each book have the number of women that like the book listed and limit the list to book on average 100 women like.**

SELECT\*

FROM FavoriteBooks;

use facebook;

SELECT FB.Book AS Favorite\_Books, COUNT(FB.ProfileID) as number\_of\_likes

FROM FavoriteBooks FB

inner join Profiles P

ON FB.ProfileID = P.ProfileID

WHERE P.Sex = 'Female'

GROUP BY Favorite\_Books

HAVING COUNT(FB.ProfileID) >=100

order by COUNT(FB.ProfileID) DESC;

**#8. In Facebook database, find the most commonly declared double majors/concentrations**

SELECT DISTINCT A.Concentration as major1, B.Concentration as major2, count(\*) as

num\_students from Concentration A join Concentration B

where A.Concentration < B.Concentration and

A.ProfileID=B.ProfileID

group by A.Concentration, B.Concentration

order by num\_students desc;

use imdb;

**#9. In IMDb database, analyze the first names of the act and correlate it to gender**

SELECT D.\*FROM

(select C.first\_name,count(1) as num\_actors from actors C

where not(C.first\_name like "%.%" or

C.first\_name like "%(%" or

C.first\_name like "%)%")

group by C.first\_name

having num\_actors >=50)D

WHERE D.first\_name NOT IN (select A.first\_name from actors A WHERE A.GENDER = 'F')

ORDER BY D.num\_actors DESC;

**#10. Report the names and frequencies for males and females whos names appear as both male and female**

use imdb;

SELECT C.first\_name, C.male\_count, D.female\_count,C.male\_count/D.female\_count as ratio

FROM(

SELECT A.first\_name, A.gender, count(1) as male\_count from actors A where A.gender = 'M'group by A.first\_name, A.gender) C

JOIN(select B.first\_name, B.gender, count(1) as female\_count from actors B where B.gender = 'F'group by B.first\_name, B.gender) D

ON C.first\_name = D.first\_name #and C.last\_name = D.last\_name

WHERE abs(C.male\_count/D.female\_count) = 1

ORDER BY ratio DESC;