CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB ASSIGNMENT - 1

Banner ID: B00948977

Git Assignment Link :
https://git.cs.dal.ca/sukumaran/csci5408_f23_b00948977_balaji_sukumaran

Table of contents

Problem Statement 1: **Check how many unique actors are present in IMDB dataset**

Step 1: Since '**id**' is the primary key for table **actors**

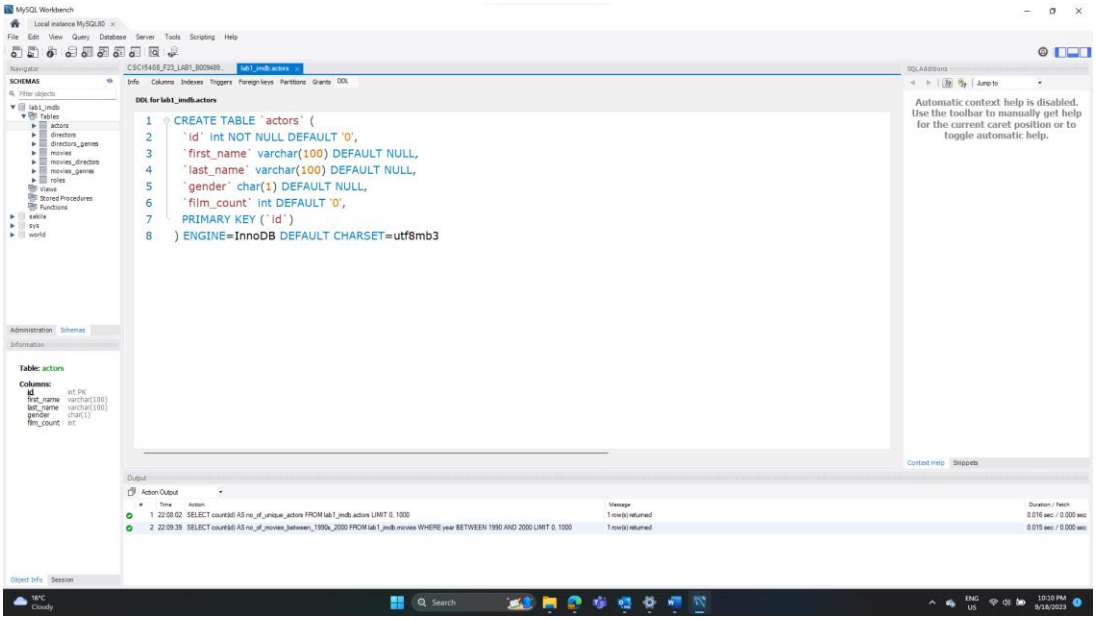

Figure 1: DDL of Actors table

Step 2: Finding the number of primary key gives me the number of unique actors even when the actors have same name.

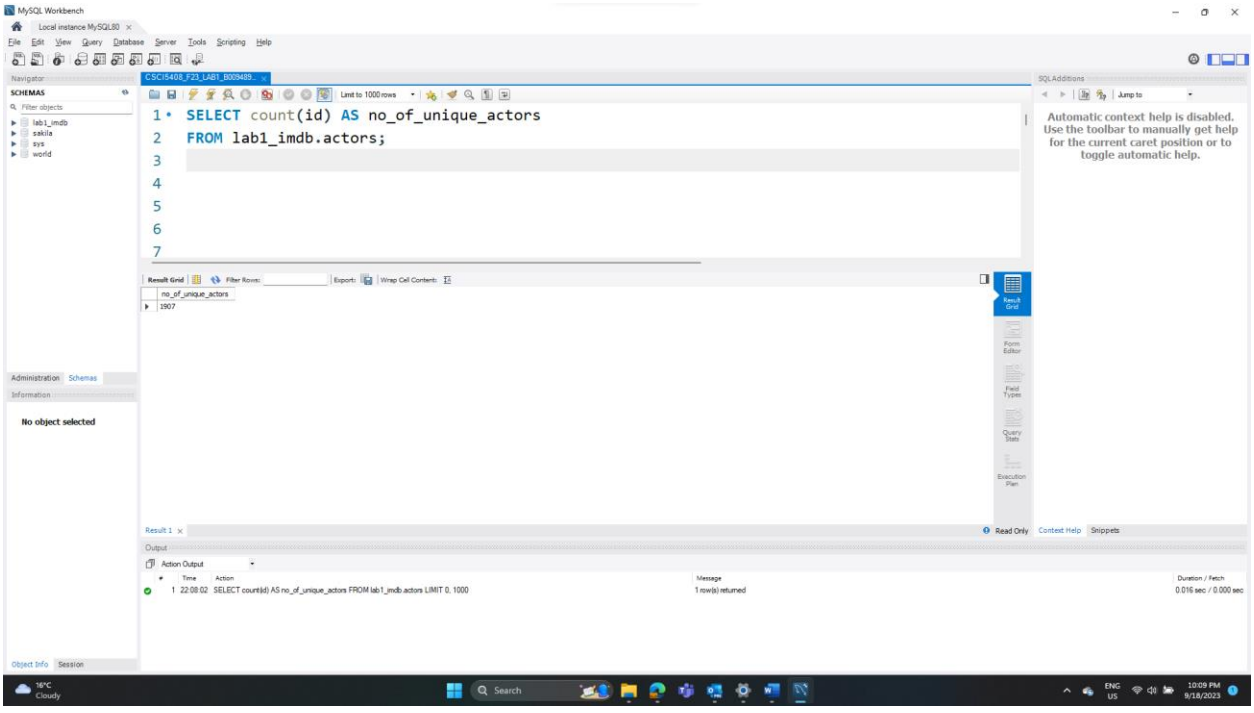

Figure 2: Select query for finding the number of unique actors

Problem Statement 2: Check how many movies are released between the year 1990s till 2000

Step 1: Since ‘id’ is the primary key of the movies table. Which uniquely identifies each row in the table I can count(id) instead of count(*) to improve performance

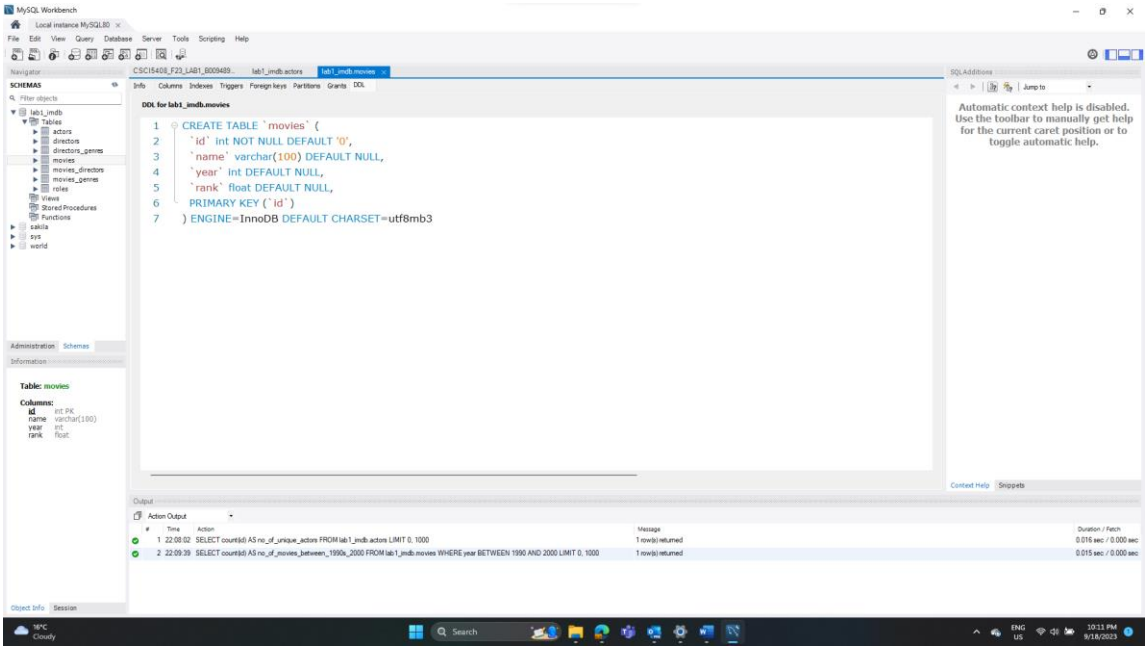

Figure 3: DDL of Movies table

Step 2: Selected the count of movies which released in the years between 1990s till 2000 using where clause.

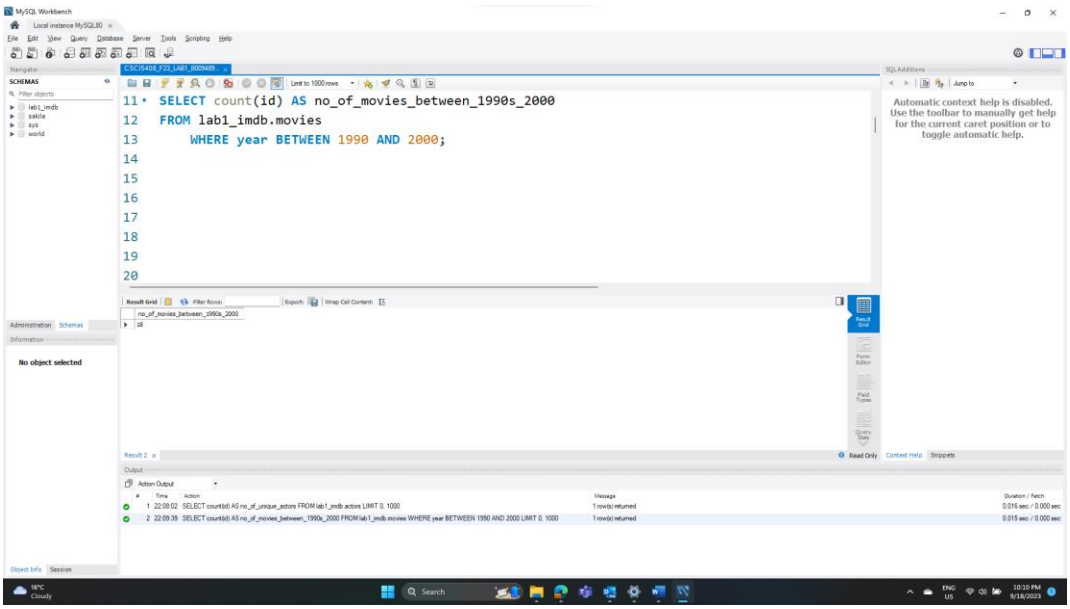

Figure 4: select query to return the number of movies released between 1990s and 2000s

Problem Statement 3: Find the list of genres of movies directed by Christopher Nolan

Step 1: Inner joined the **directors_genres** table and **directors** table using director_id and **filtered** the results based on director **first name and last name**.

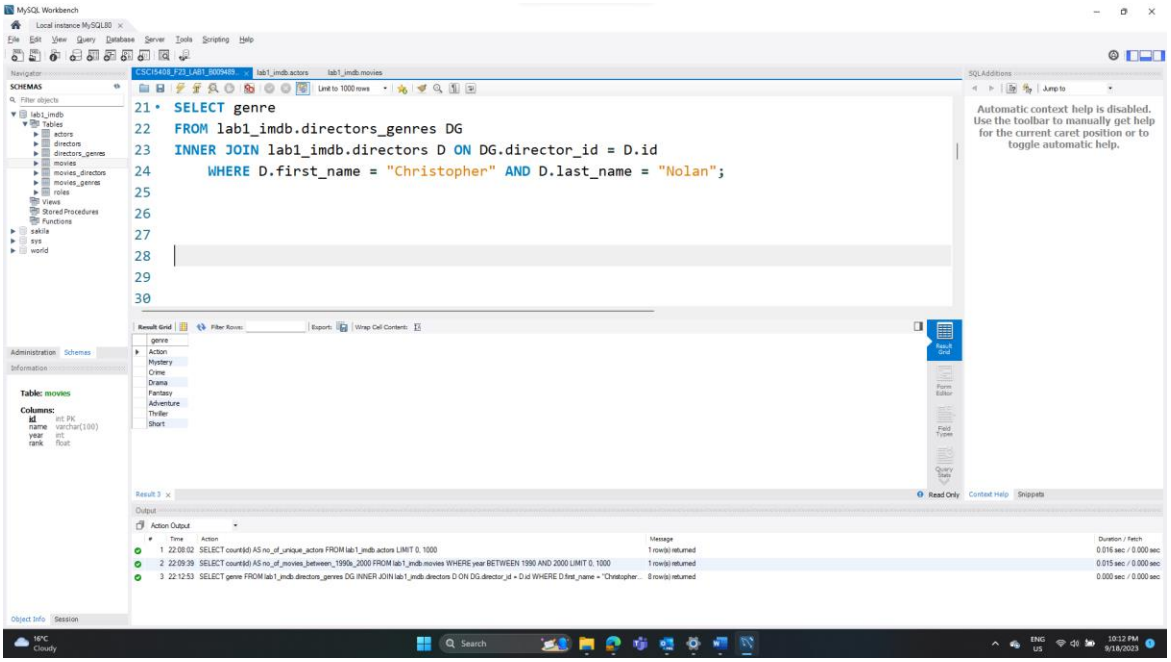

Figure 5: Inner joined directors_genre and directors table

Alternate solution, using subquery.

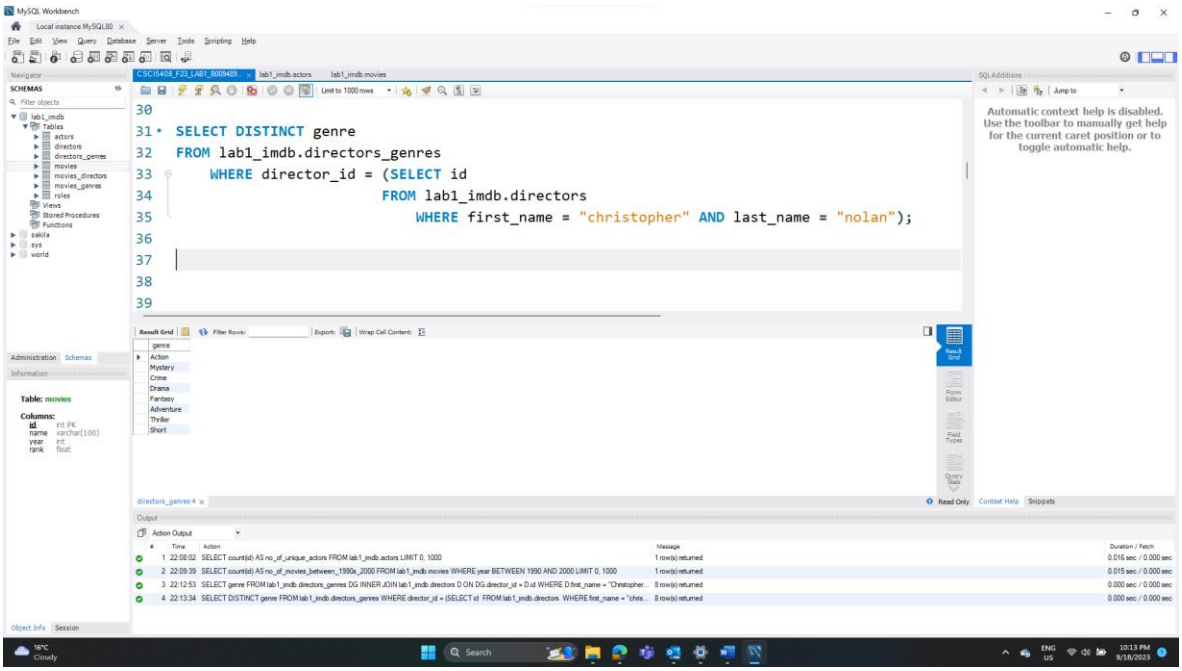

Figure 6: Alternate solution using subquery

Problem Statement 4: Find the list of all directors, and the movie name which are ranked between 8 to 9 and have a genre of Sci-Fi and Action.

Step 1: By inner joining the **movies**, **movies_directors**, **directors, movies_genres** table I was able to get the director name, movie name, rank, genre.

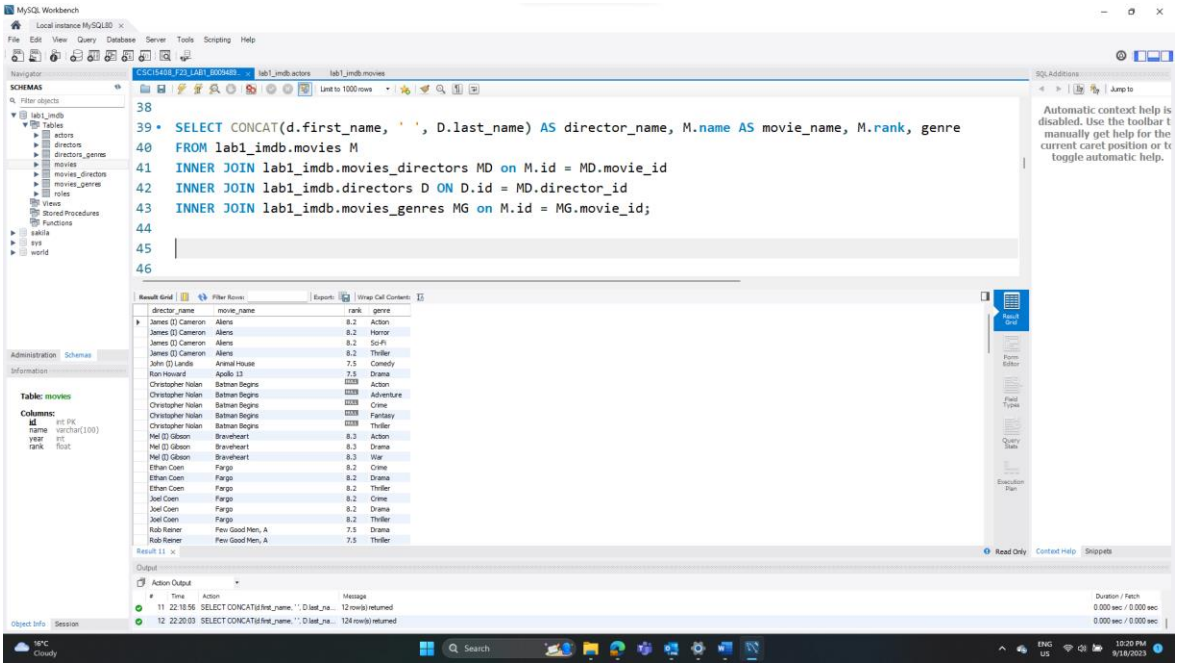

Figure 7: inner joined movies_directors, movies, directosm movies_genres table

Step 2: Filtered the results based on the rank (between 8 and 9) and genre (action, sci-fi)

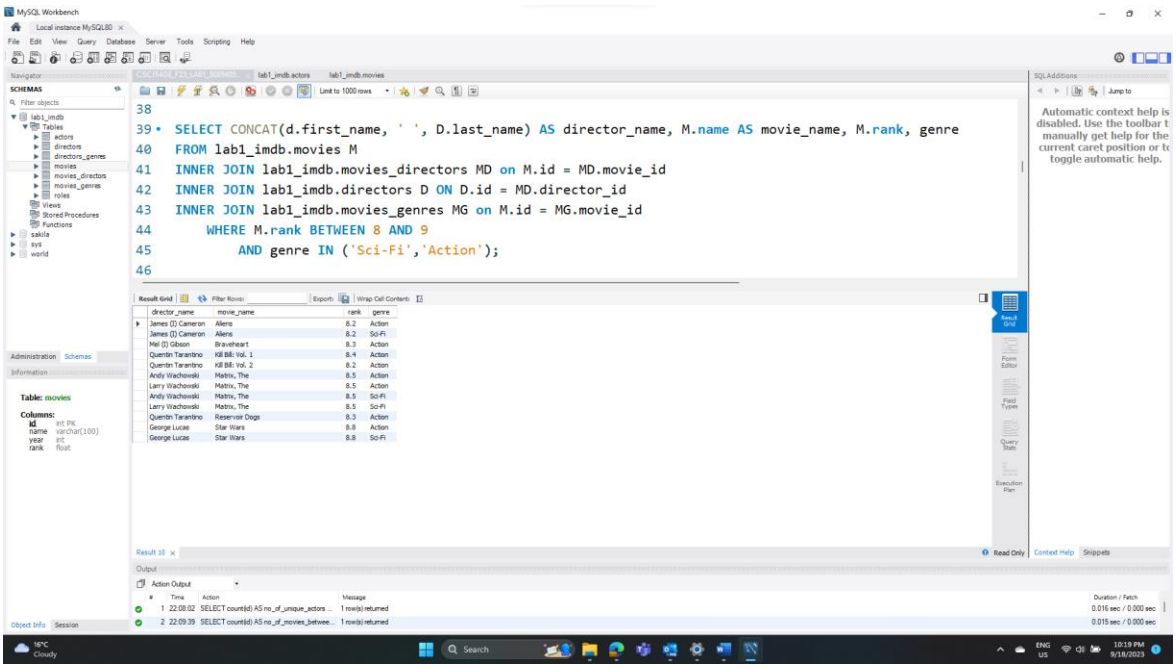

Figure 8: Applied where clause to the joined query

Step 3: Grouping the above query by **director_name, movie_name** gives me the count of genre in this combination (which will always be either 1 or 2 because of the genre where clause) filtering only count_genre having 2 gives me the result of movie, director_name whose genre is both Action **and** Sci-Fi.

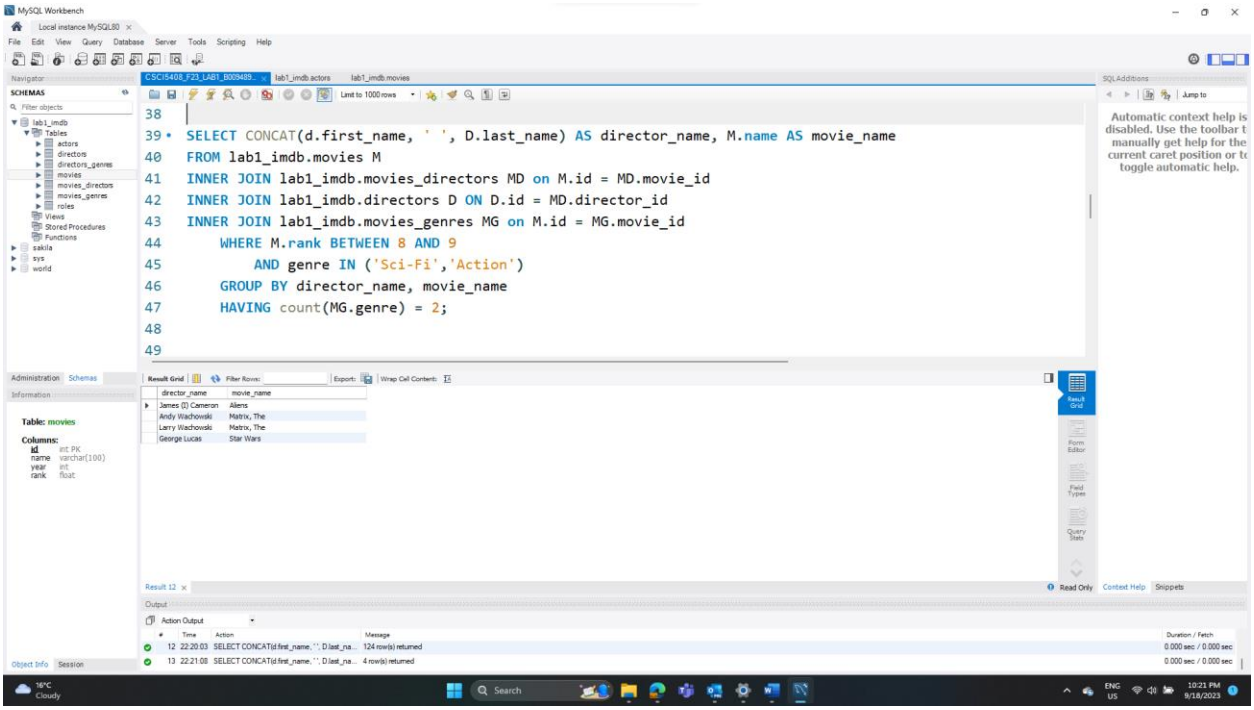

Figure 9: Grouped the query based on director_name, movie_name

Problem Statement 5: Find the name of the movie in which the actor's role is any doctor, and the movie has the highest number of roles of doctor.

`

Step 1: Joined the roles and movies table by id and filtered the results where role starts with a 'Dr' tag

Step 2: Grouping the results by name gives me the number of doctors in each movie.

Step 3: Ordered the results in desc order of number of doctors in each movie and selected the top 1 for the movie with most number of doctors.

Step 4: Kept all these inside a subquery to select just name, not the name and count of doctors.

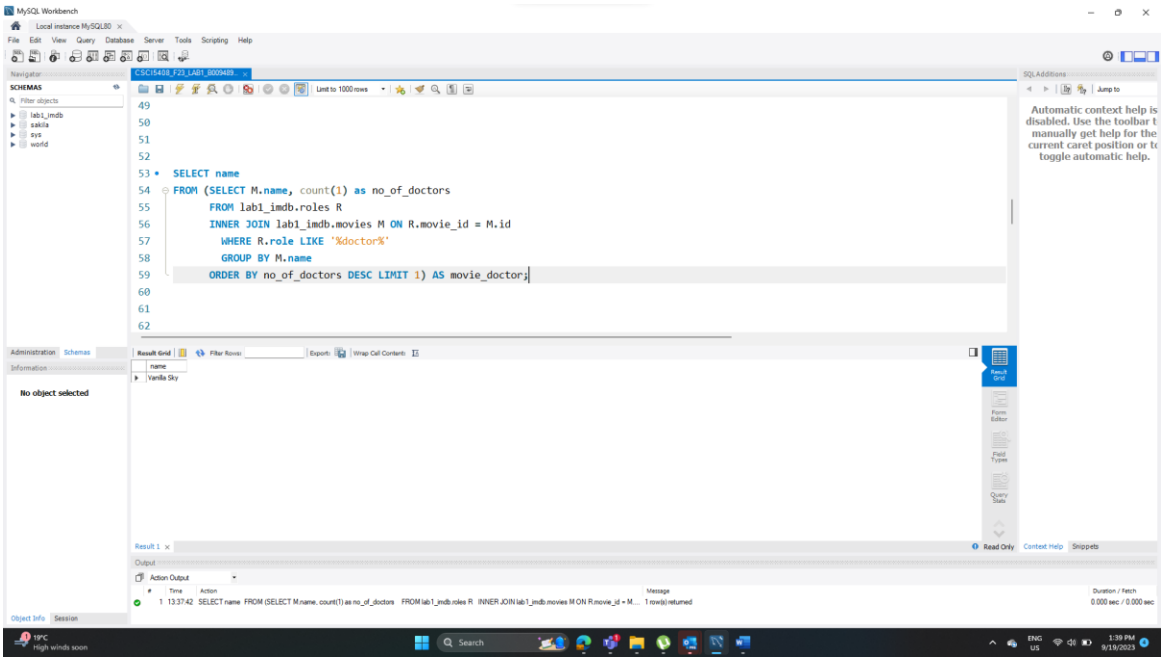

Figure 10: Query to find the highest number of roles of doctor in a movie

Problem Statement 6: **Find the list of the movies that start the letter ‘f’**

Select query to get the movie name and used a where clause to filter the ones whose name starts with ‘f’

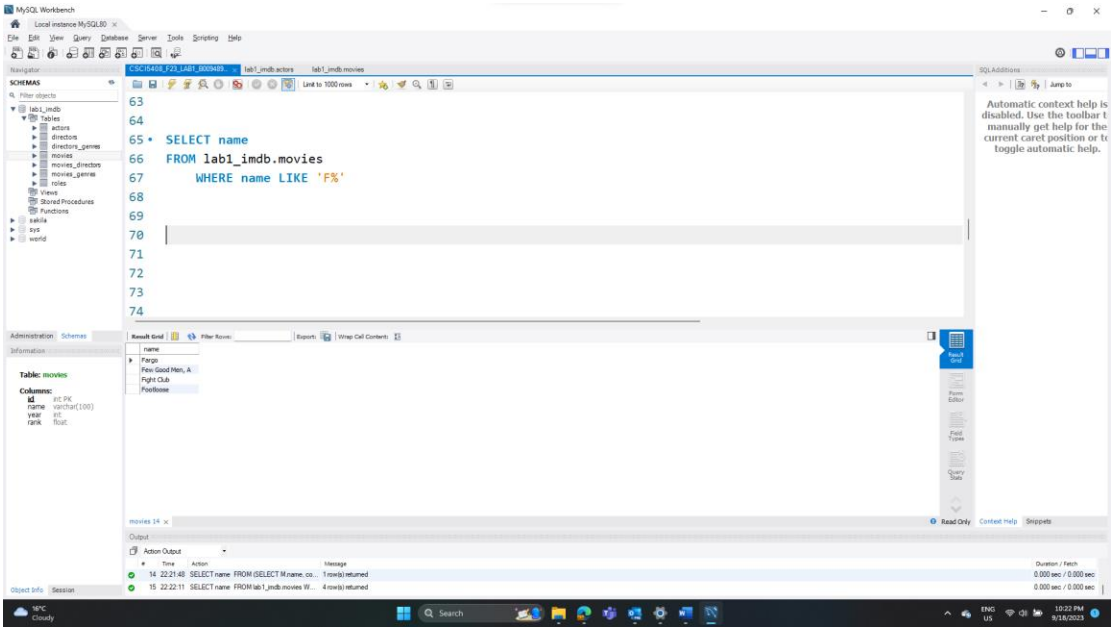

Figure 11: query to find the movies starting with letter F