

# 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](https://git.cs.dal.ca/sukumaran/csci5408_f23_b00948977_balaji_sukumaran)

## Table of contents

---

<b>Problem Statement 1:</b> Check how many unique actors are present in IMDB dataset.....	<b>1</b>
<b>Problem Statement 2:</b> Check how many movies are released between the year 1990s till 2000 .....	<b>2</b>
<b>Problem Statement 3:</b> Find the list of genres of movies directed by Christopher Nolan .....	<b>3</b>
<b>Problem Statement 4:</b> 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 .....	<b>4</b>
<b>Problem Statement 5:</b> 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 .....	<b>6</b>
<b>Problem Statement 6:</b> Find the list of the movies that start the letter 'f' .....	<b>7</b>



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

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

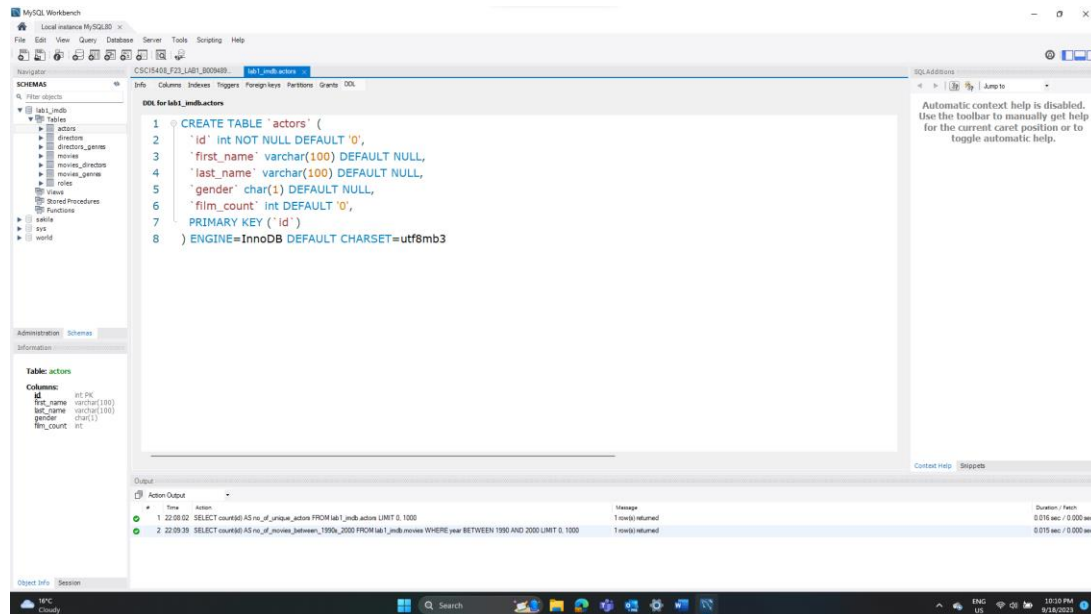


Figure 1: DDL of Actors table

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

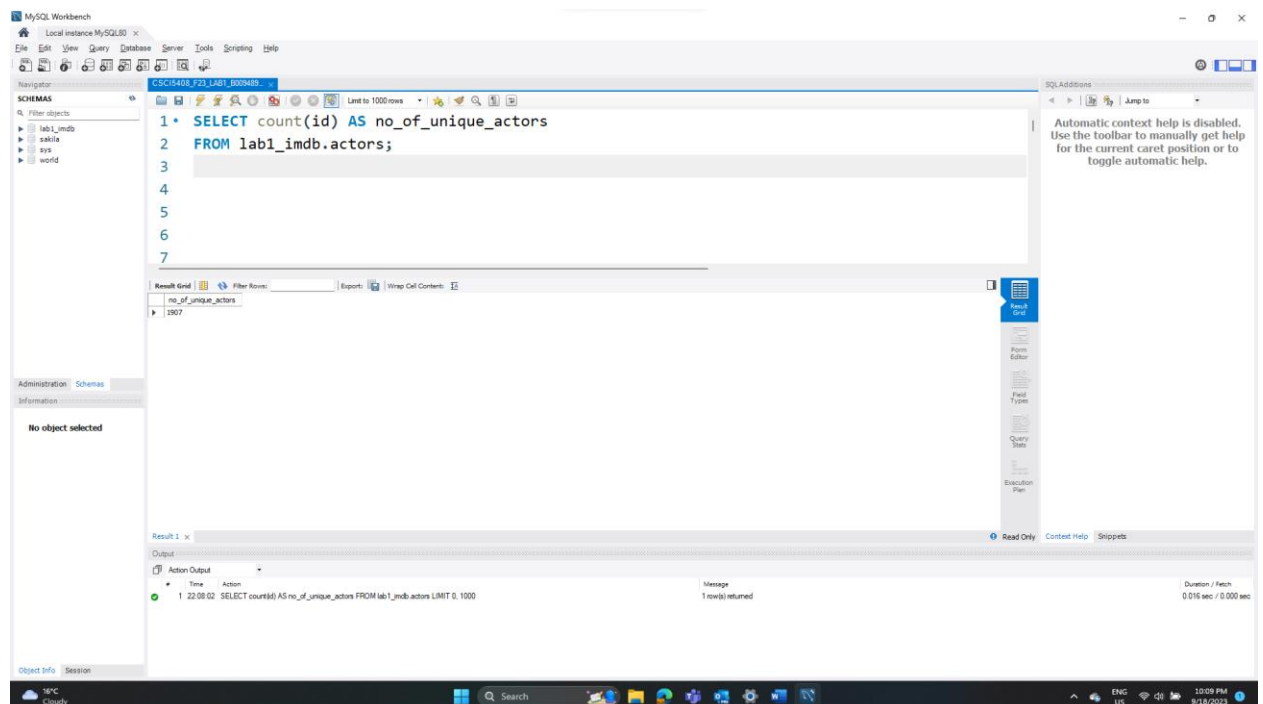


Figure 2: Select query for finding the number of unique actors

## 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

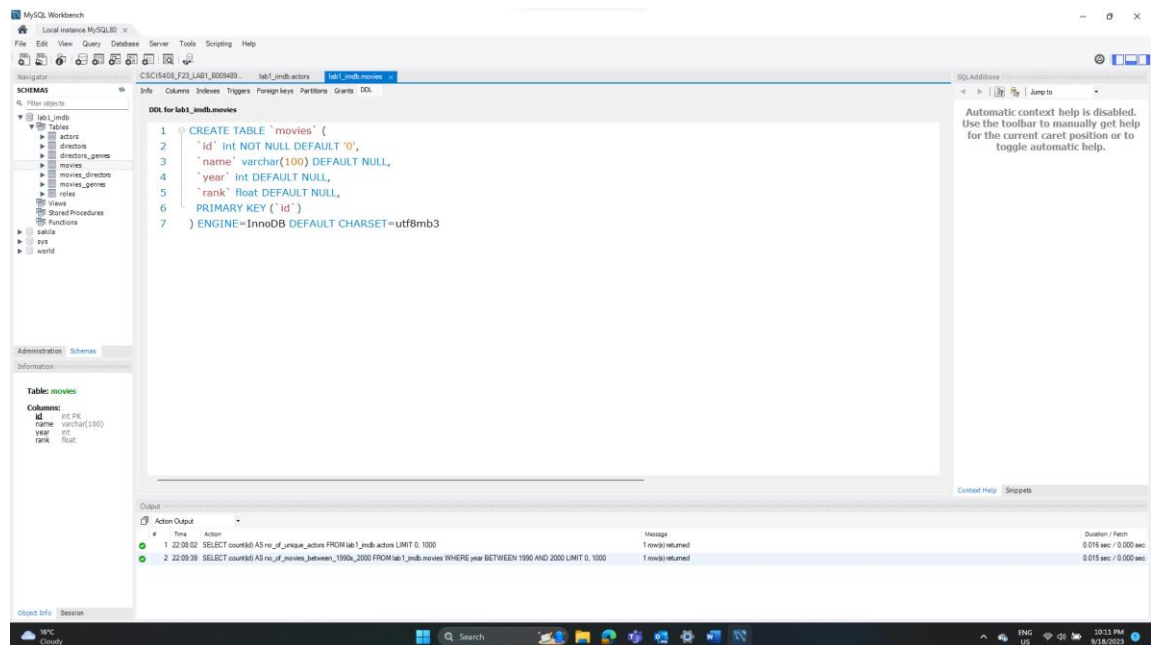


Figure 3: DDL of Movies table

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

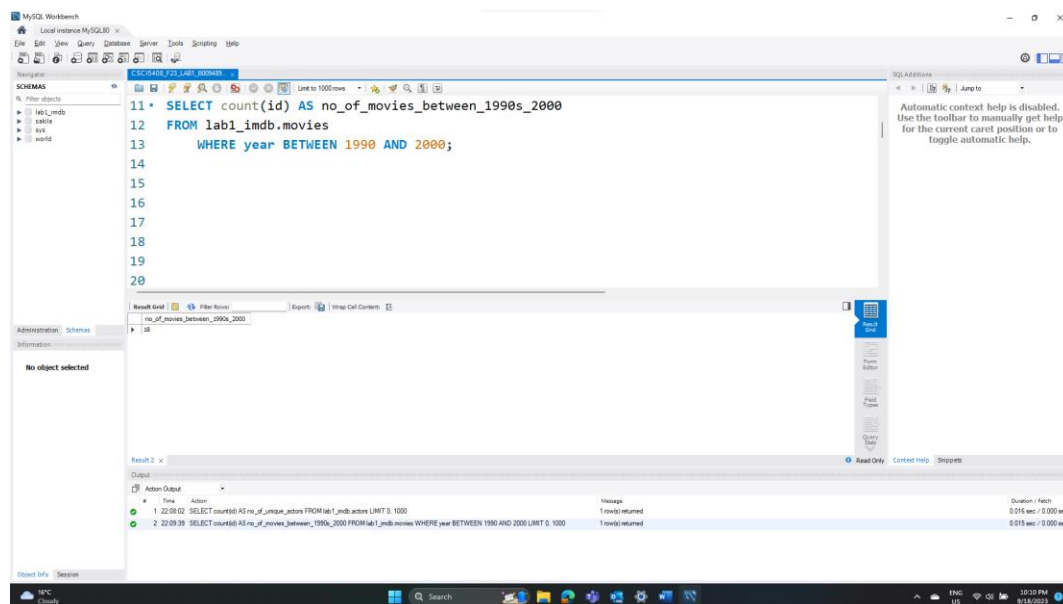


Figure 4: select query to return the number of movies released between 1990s and 2000s

### 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**.

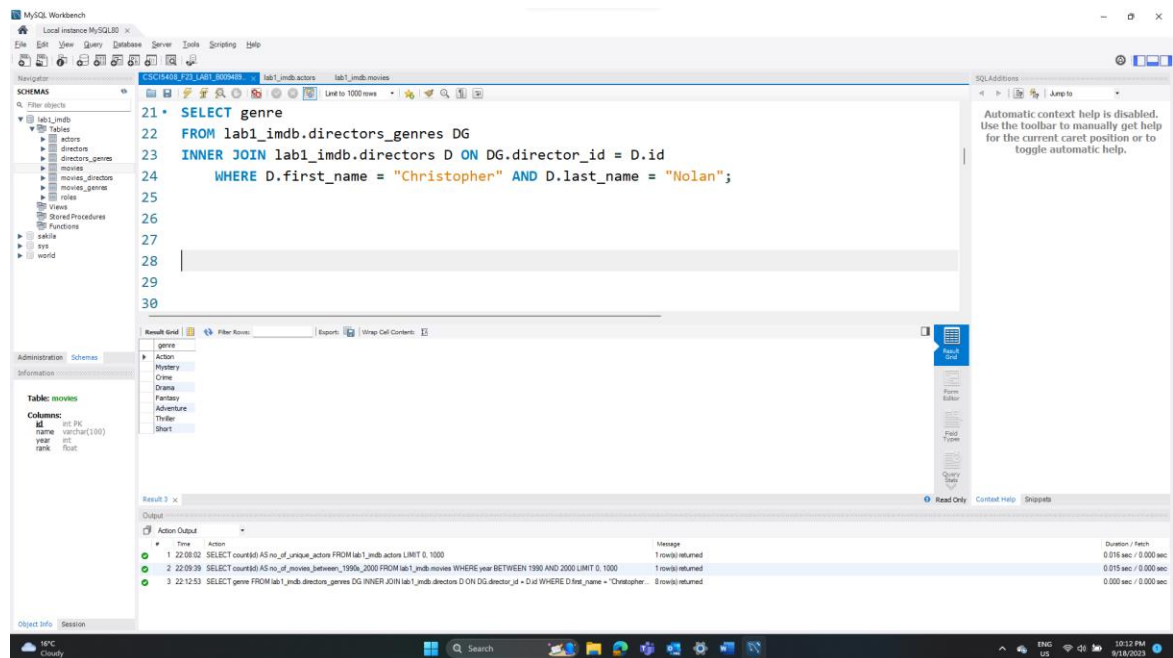


Figure 5: Inner joined directors\_genre and directors table

Alternate solution, using subquery.

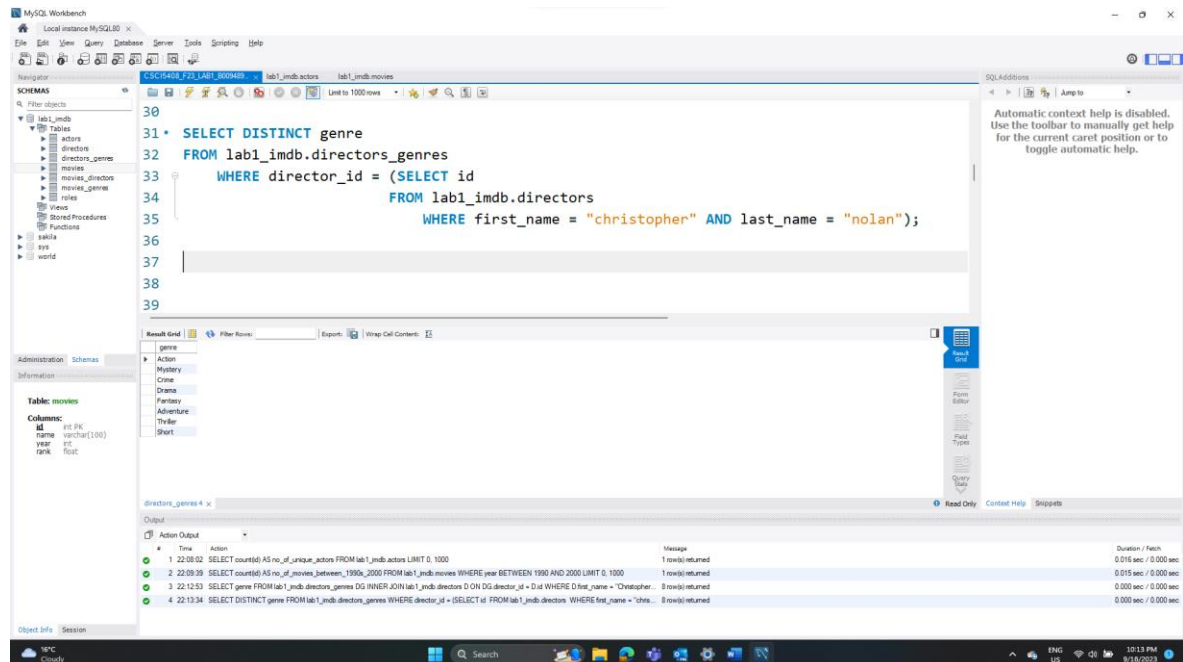


Figure 6: Alternate solution using subquery

**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.

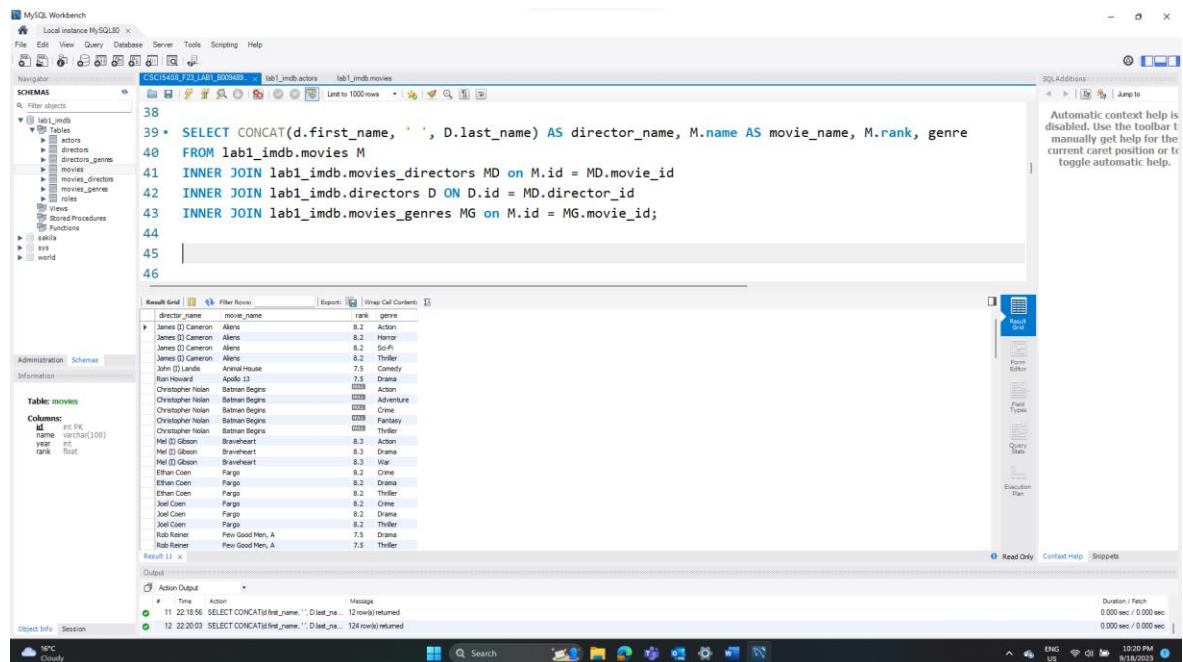


Figure 7: inner joined movies\_directors, movies, directors, movies\_genres table

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

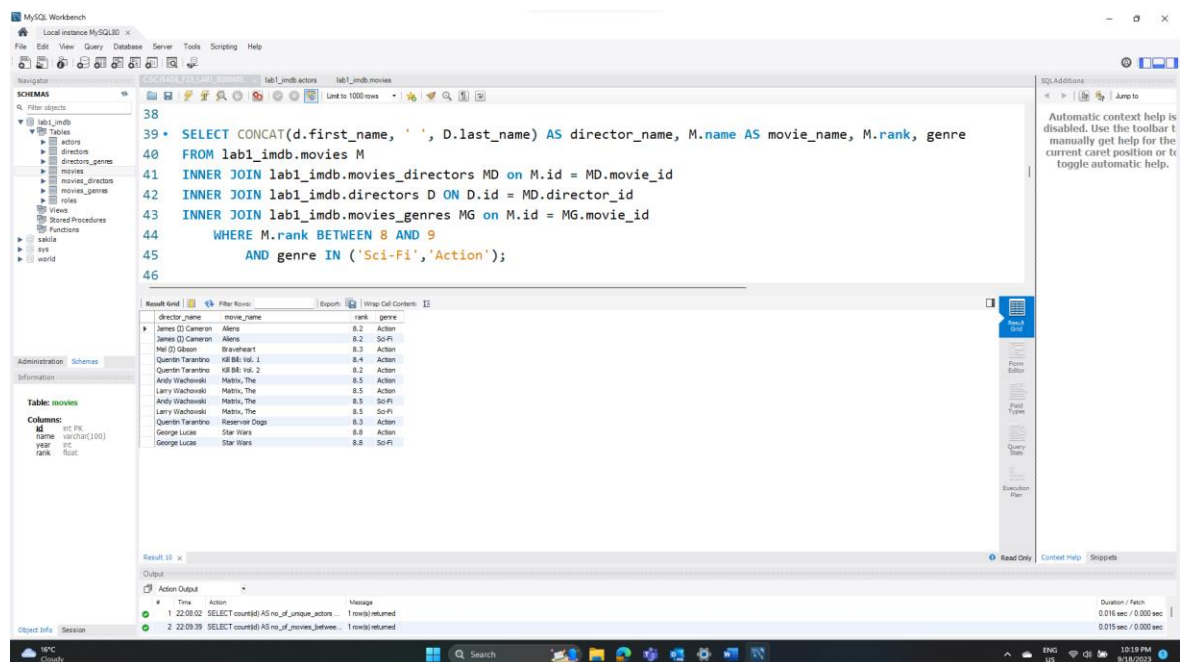


Figure 8: Applied where clause to the joined query

**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.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

38
39 • SELECT CONCAT(d.first_name, ' ', D.last_name) AS director_name, M.name AS movie_name
40 FROM lab1_imdb.movies M
41 INNER JOIN lab1_imdb.movies_directors MD on M.id = MD.movie_id
42 INNER JOIN lab1_imdb.directors D ON D.id = MD.director_id
43 INNER JOIN lab1_imdb.movies_genres MG on M.id = MG.movie_id
44 WHERE M.rank BETWEEN 8 AND 9
45 AND genre IN ('Sci-Fi','Action')
46 GROUP BY director_name, movie_name
47 HAVING count(MG.genre) = 2;
48
49

```

The Results Grid shows the following data:

director_name	movie_name
James (J) Cameron	Aliens
Andy Wachowski	Matrix, The
Larry Wachowski	Matrix, The
George Lucas	Star Wars

The Output pane shows the execution log:

```

12 22:20:03 SELECT CONCAT(d.first_name, ' ', D.last_name) AS director_name, M.name AS movie_name
13 22:21:08 SELECT CONCAT(d.first_name, ' ', D.last_name) AS director_name, M.name AS movie_name

```

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.

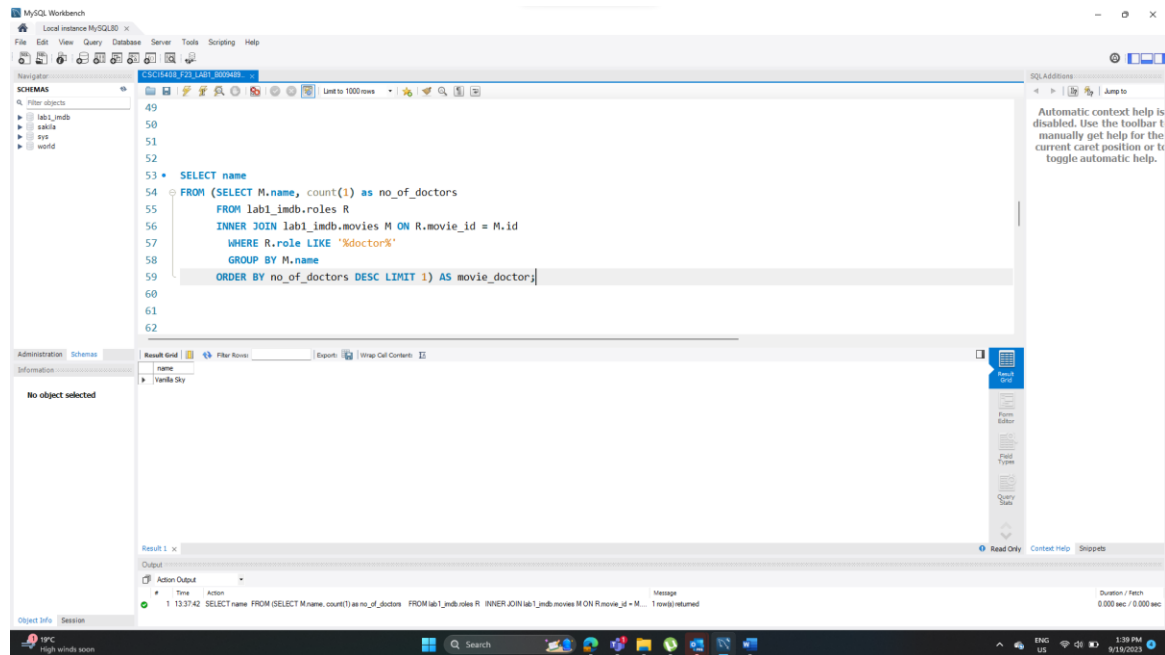


Figure 10: Query to find the highest number of roles of doctor in a movie

### 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'

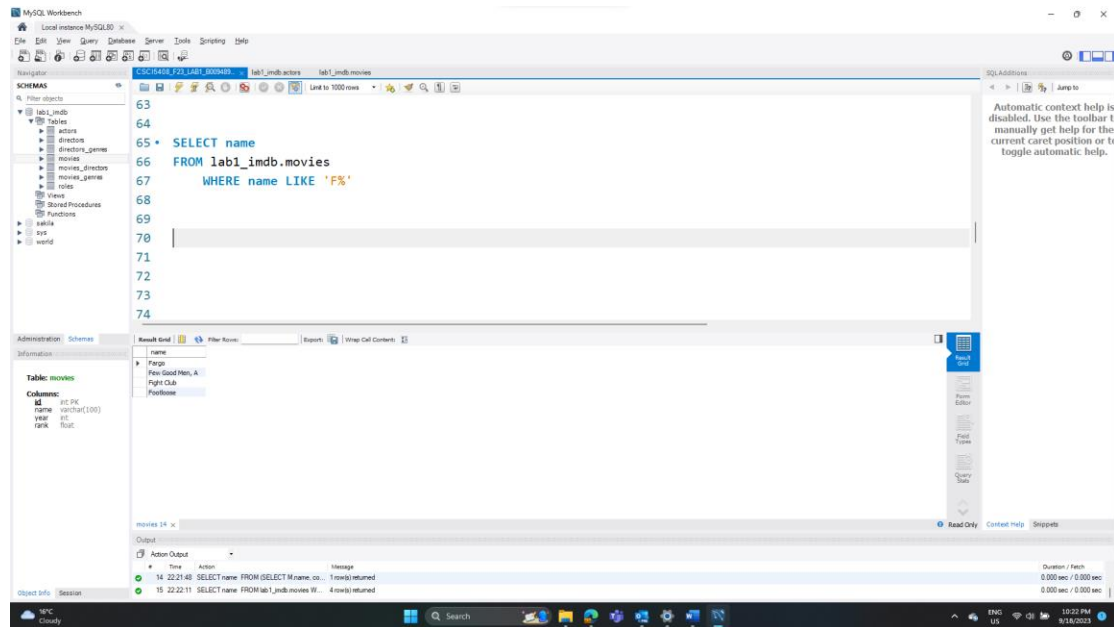


Figure 11: query to find the movies starting with letter F