

CSCI 5408

DATA MANAGEMENT AND
WAREHOUSING

LAB-1: Introduction to MySQL

Table of Contents

1. Check how many unique actors are present in IMDB dataset.....	3
2. Check how many movies are released between the year 1990s till 2000.....	4
3. Find the list of genres of movies directed by Christopher Nolan.....	5
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.....	6
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.....	7
6. Find the list of the movies that start with the letter 'f'.....	8

Query 1: Check how many unique actors are present in IMDB dataset.

SQL Query :

```
use imdb;
SELECT Count(*) FROM actors;
```

Explanation :

- Using the DISTINCT keyword we will be able to get unique actors in the IMDB dataset, and using the COUNT keyword helps us to get the total number of unique actors.

Output :

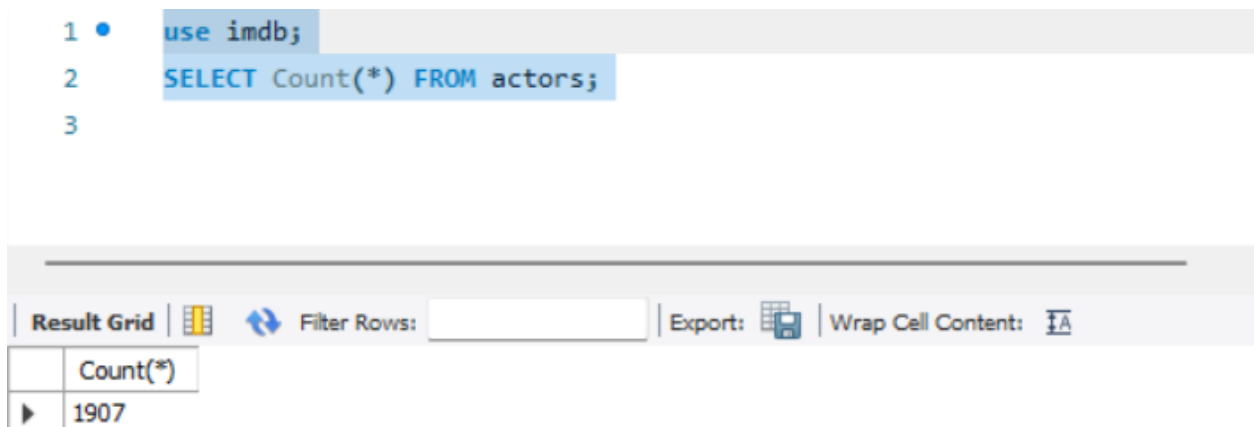


Figure 1: MySQL Workbench output of Query 1

Query 2: Check how many movies are released between the year 1990s till 2000.

SQL Query :

use imdb;

Select count(*) FROM movies WHERE year>=1990 AND year<=2000;

Explanation :

- Here we want to only those movie names that were released within two given years we can achieve this data by either using the BETWEEN keyword or just using the AND keyword.

Output :

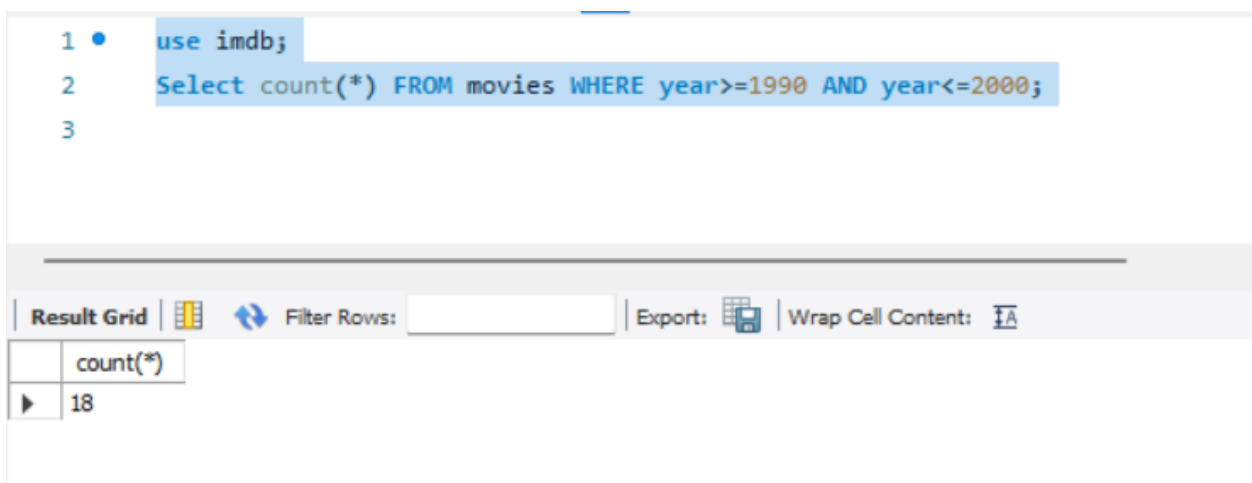


Figure 2: MySQL Workbench output of Query 2

Query 3: Find the list of genres of movies directed by Christopher Nolan.

SQL Query :

```
use imdb;
select distinct genre from directors_genres where director_id IN (select id from directors where
first_name="Christopher" AND last_name="Nolan");
```

Explanation :

- Here I use a subquery that fetches a director ID whose name is Christopher Nolan and using that ID we fetch the genre of that ID from directors_genres. Lastly, we use distinct keywords to get the unique genre.

Output :

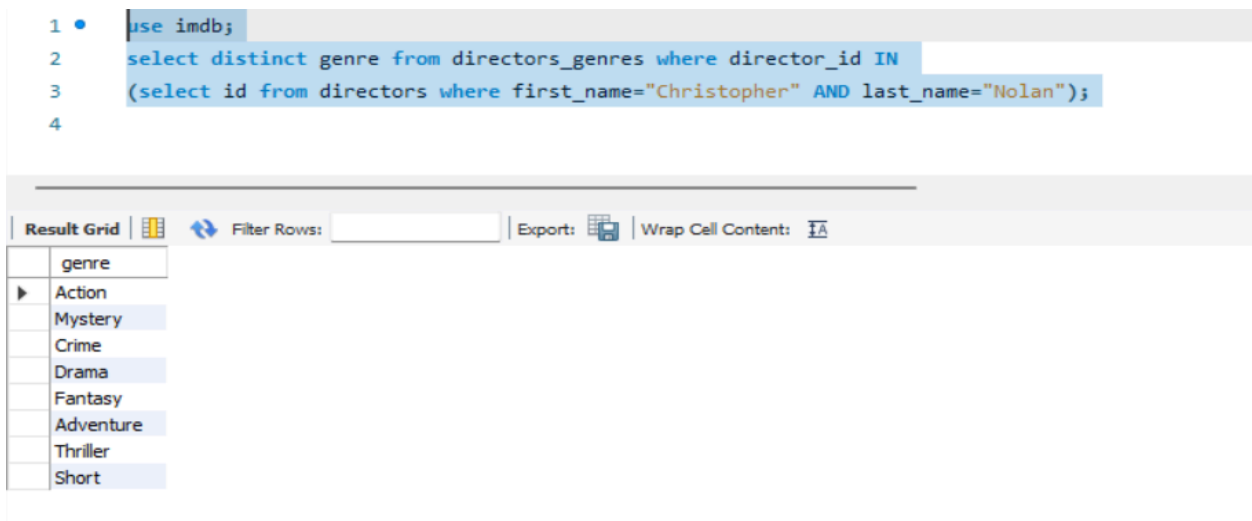


Figure 3: MySQL Workbench output of Query 3

Query 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.

SQL Query :

```
USE imdb;
```

```
SELECT DISTINCT md.director_id  
FROM movies m  
JOIN movies_genres mg ON m.id = mg.movie_id  
JOIN movies_directors md ON mg.movie_id = md.movie_id  
WHERE (m.rank BETWEEN 8 AND 9)  
AND (mg.genre = "Sci-Fi" OR mg.genre = "Action");
```

Explanation :

- Here instead of using subquery I use the JOIN keyword to join 3 tables. First I join movies and movie_genres using movie id and then result of I join with movies_directors with the same movie id. Now I filter data based on rank and genre.

Output :

The screenshot displays the MySQL Workbench interface. The top pane shows the SQL query for Query 4, which is highlighted in blue. The bottom pane shows the 'Result Grid' with the output of the query. The output is a single column named 'director_id' containing six values: 11652, 28395, 78273, 83616, 83617, and 48115. The interface also includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox.

```
1 • USE imdb;  
2  
3  
4 • SELECT DISTINCT md.director_id  
5 FROM movies m  
6 JOIN movies_genres mg ON m.id = mg.movie_id  
7 JOIN movies_directors md ON mg.movie_id = md.movie_id  
8 WHERE (m.rank BETWEEN 8 AND 9)  
9 AND (mg.genre = "Sci-Fi" OR mg.genre = "Action");  
10
```

director_id
11652
28395
78273
83616
83617
48115

Figure 4: MySQL Workbench output of Query 4

Query 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.

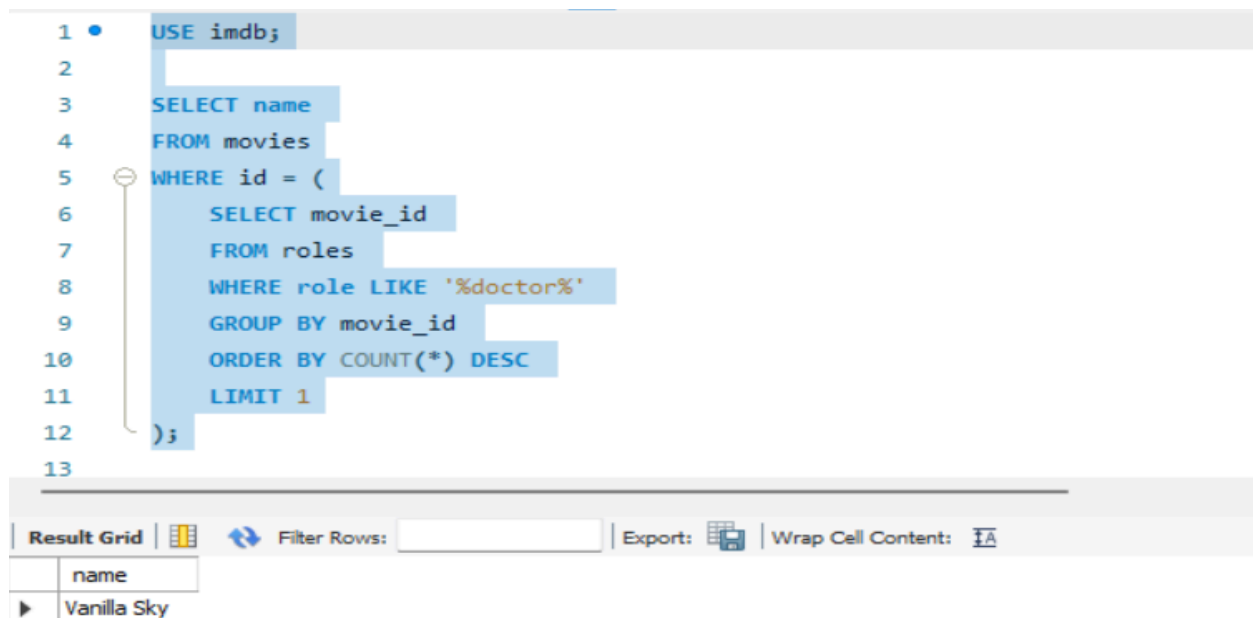
SQL Query :

```
USE imdb;
SELECT name
FROM movies
WHERE id = (
    SELECT movie_id
    FROM roles
    WHERE role LIKE '%doctor%'
    GROUP BY movie_id
    ORDER BY COUNT(*) DESC
    LIMIT 1
);
```

Explanation :

- Using the Like keyword we get the data who contains that given info. Moreover using Group By we group the data which have similar movie_id and finally, I sort the data into descending order and take out the top row.

Output :



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

```
1 • USE imdb;
2
3 SELECT name
4 FROM movies
5 WHERE id = (
6     SELECT movie_id
7     FROM roles
8     WHERE role LIKE '%doctor%'
9     GROUP BY movie_id
10    ORDER BY COUNT(*) DESC
11    LIMIT 1
12 );
13
```

Below the editor, the 'Result Grid' tab is active, displaying the output of the query:

name
Vanilla Sky

Figure 5: MySQL Workbench output of Query 5

Query 6: Find the list of the movies that start with the letter 'f'.

SQL Query :

```
USE imdb;
```

```
SELECT name FROM movies where name LIKE 'f%';
```

Explanation :

- It is the same as the previous query. Using the Like keyword we find the names of movies that start with f.

Output :

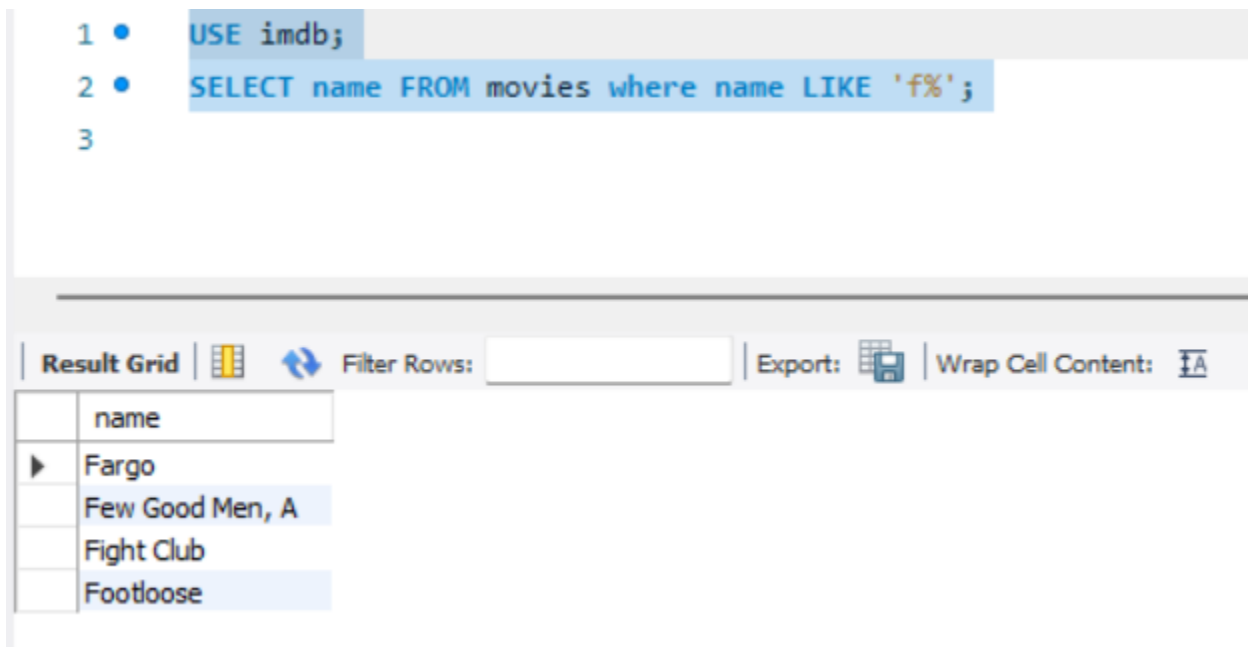


Figure 6: MySQL Workbench output of Query 6