DALHOUSIE UNIVERSITY

Data Management, Warehousing and Analytics

Jaskaran Singh

MACS

B00948857

js356337@dal.ca

Lab Assignment 1

Gitlab Repository link:

https://git.cs.dal.ca/singh16/csci5408_s23_b00948857_jaskaran_singh.git

Queries

1. Check how many directors are present in iMDB:

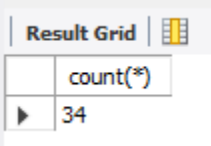
Here, required data can be fetched from a single table.

We can use 'SELECT' command (DDL) to retrieve data from directors table.

To get number of directors, we can use count() function which basically returns the number of rows that matches a specified criterion.

Query: SELECT count(*) FROM directors;

Output:



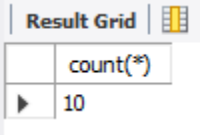
2. Check how many movies are released post-year 2000:

Here, we can use 'where' clause to filter out movies released post-year 2000.

Query: SELECT count(*) FROM movies WHERE movies.year > 2000;

(Note – movies released in the year 2000 not included in the above query)

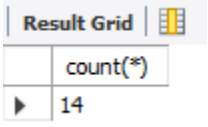
Output:



Query: SELECT count(*) FROM movies WHERE movies.year >= 2000;

(Note – movies released in the year 2000 included in the above query)

Output:



3. Find the list of genres of movies directed by Andrew Adamson:

Here, required data can't be fetched from a single table.

Basically, we have to use 'SELECT' command (DDL) to retrieve data from a group of tables.

Data related to genres is present in **directors_genres** table and directors' data is present in **directors** table.

We can use **sub-queries** or **SQL joins** to retrieve records from multiple tables that have

common fields.

Additionally, **concat**() function is used to add strings and **lower**() function is used to remove case mismatch scenario.

Query:

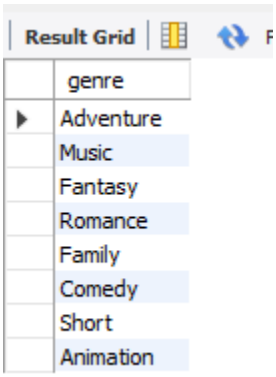
Using Inner Join:

```
SELECT G.genre FROM directors_genres G INNER JOIN directors D ON G.director_id = D.id
AND LOWER(concat(first_name,' ', last_name)) = LOWER('Andrew Adamson');
```

Using sub-query:

```
SELECT genre FROM directors_genres WHERE director_id IN
(SELECT id FROM directors WHERE LOWER(concat(first_name,' ', last_name)) =
LOWER('Andrew Adamson'));
```

Output:



4. List of directors whose movies are ranked between 7 to 8 ranking.

Here, required data can't be fetched from a single table.

Basically, we have to use 'SELECT' command (DDL) to retrieve data from a group of tables.

Required data is present in **directors**, **movies_directors** and **movies** tables.

We can use **sub-queries** or **SQL joins** to retrieve records from multiple tables that have common fields.

Query:

Using Inner Join:

```
SELECT D.*,M.name, M.rank FROM directors D INNER JOIN movies_directors MD ON D.id =
MD.director_id INNER JOIN movies M ON MD.movie_id = M.id WHERE M.rank>=7 AND
M.rank<=8;
```

Note:- If movies having ranking 7 and 8 not required, remove '=' operator from the above query in 'where'clause.

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

id	first_name	last_name	name	rank
44291	John (I)	Landis	Animal House	7.5
35573	Ron	Howard	Apollo 13	7.5
65940	Rob	Reiner	Few Good Men, A	7.5
76524	Oliver (I)	Stone	JFK	7.8
14927	Ron	Clements	Little Mermaid, The	7.3
56332	John	Musker	Little Mermaid, The	7.3
15906	Sofia	Coppola	Lost in Translation	8
15092	Ethan	Coen	O Brother, Where Art Thou?	7.8
15093	Joel	Coen	O Brother, Where Art Thou?	7.8
74758	Steven	Soderbergh	Ocean's Eleven	7.5
38746	Mike (I)	Judge	Office Space	7.6
2931	Darren	Aronofsky	Pi	7.5
35838	John (I)	Hughes	Planes, Trains & Automobiles	7.2
66849	Guy	Ritchie	Snatch.	7.9
41975	David	Koepp	Stir of Echoes	7

Using sub-query:

```
SELECT * FROM directors WHERE id IN
(SELECT director_id FROM movies_directors WHERE movie_id IN
(SELECT id FROM movies WHERE movies.rank>=7 and movies.rank<=8));
```

Note:- If movies having ranking 7 and 8 not required, remove '=' operator from the above query in 'where'clause.

Result Grid | Filter Rows: | Edit

id	first_name	last_name
44291	John (I)	Landis
35573	Ron	Howard
65940	Rob	Reiner
76524	Oliver (I)	Stone
14927	Ron	Clements
56332	John	Musker
15906	Sofia	Coppola
15092	Ethan	Coen
15093	Joel	Coen
74758	Steven	Soderbergh
38746	Mike (I)	Judge
2931	Darren	Aronofsky
35838	John (I)	Hughes
66849	Guy	Ritchie
41975	David	Koepp

5. Find the role of Julliet Akinyi in Lost in Translation movie:

Here, required data can't be fetched from a single table.

Basically, we have to use 'SELECT' command (DDL) to retrieve data from a group of tables.

Required data is present in **movies**, **roles** and **roles** tables.

We can use **sub-queries** or **SQL joins** to retrieve records from multiple tables that have common fields.

In addition, **upper**() function is used to remove case mismatch scenario.

Query:

Using Inner Join:

```
SELECT R.* FROM movies M INNER JOIN roles R ON M.id = R.movie_id and
UPPER(M.name) = UPPER('Lost in Translation') INNER JOIN  actors A ON R.actor_id = A.id
AND UPPER(concat(A.first_name,' ',A.last_name)) = UPPER('Julliet Akinyi');
```

Result Grid | Filter Rows:

actor_id	movie_id	role
4856	194874	Hans

Using sub-query:

```
SELECT * FROM directors WHERE id IN
(SELECT director_id FROM movies_directors WHERE movie_id IN
(SELECT id FROM movies WHERE movies.rank>=7 and movies.rank<=8));
```

Result Grid | Filter Rows:

actor_id	movie_id	role
4856	194874	Hans

6. List of the movies that contain the letter ‘o' in any position:

Here, **LIKE** command can be used in **WHERE** clause to retrieve required records.

Query: SELECT * FROM movies WHERE name LIKE '%o%';

Output:

Result Grid | Filter Rows: | Edit:

id	name	year	rank
17173	Animal House	1978	7.5
18979	Apollo 13	1995	7.5
109093	Fargo	1996	8.2
111813	Few Good Men, A	1992	7.5
116907	Footloose	1984	5.8
130128	Godfather, The	1972	9
147603	Hollow Man	2000	5.3
176711	Kill Bill: Vol. 1	2003	8.4
176712	Kill Bill: Vol. 2	2004	8.2
194874	Lost in Translation	2003	8
210511	Memento	2000	8.7
237431	O Brother, Wher…	2000	7.8
238072	Ocean's Eleven	2001	7.5
238695	Office Space	1999	7.6
256630	Pirates of the Ca…	2003	NULL
257264	Planes, Trains & …	1987	7.2
267038	Pulp Fiction	1994	8.7
276217	Reservoir Dogs	1992	8.3
297838	Shawshank Rede…	1994	9
314965	Stir of Echoes	1999	7

References:

1. W3Schools, SQL Tutorial

https://www.w3schools.com/sql/default.asp