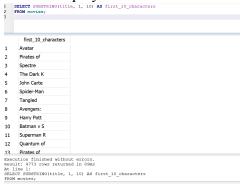
Week 14. Homework 9. Advance SQL-III

Please copy your codes and attach screenshots of your outputs under each question when you submit it in Canvas. Your submission file can be in Word or PDF format. The due date for this Homework assignment is April 26 at 1 PM.

1. Display the first 10 characters of each movie's title.



SELECT SUBSTRING(title, 1, 10) AS first_10_characters FROM movies;

2. Retrieve the titles of all movies directed by James Cameron.



SELECT movies.title
FROM movies
JOIN directors ON movies.director_id = directors.id
WHERE directors.name = 'James Cameron';

3. Find the titles of movies with a vote count greater than the average vote count of all movies.

```
SELECT title
FROM movies
FROM movies

WHERE vote_count > (SELECT AVG(vote_count) FROM movies);

title

title

Avatar

Pirates of the Caribbean: At ...

Spectre

The Dark Knight Rises

John Carter

Spider-Man 3

Tangled

Avengers: Age of Ultron

Harry Potter and the Half-Bloo...

Batman v Superman: Dawn of ...

Superman Returns

Quantum of Solace

Pirates of the Caribbean: Dead...

Execution finished without errors.

Result: 1263 rows returned in 26ms

At line 1:

SELECT title
FROM movies
```

SELECT title FROM movies WHERE vote_count > (SELECT AVG(vote_count) FROM movies);

4. List the names of directors who have never directed a movie with a budget of 0.

```
SELECT DISTINCT directors.name
    FROM directors
   ₩HERE NOT EXISTS (
         FROM movies
         WHERE movies.director_id = directors.id
AND movies.budget = 0
10
11
     James Cameron
1
2
     Gore Verbinski
3
     Sam Mendes
4
     Christopher Nolan
    Andrew Stanton
6
     Byron Howard
     Joss Whedon
8
    David Yates
     Zack Snyder
Result: 1441 rows returned in 7318ms
At line 1:
SELECT DISTINCT directors.name
```

```
SELECT DISTINCT directors.name
FROM directors
WHERE NOT EXISTS (
    SELECT 1
    FROM movies
    WHERE movies.director_id = directors.id
    AND movies.budget = 0
);
```

5. List movies with 'Life' in the title and a popularity score above the average popularity of all movies.



SELECT title
FROM movies
WHERE title LIKE '%Life%'
AND popularity > (SELECT AVG(popularity) FROM movies);

6. Find directors who have directed movies with a budget of 0.



SELECT DISTINCT directors.name FROM directors JOIN movies ON directors.id = movies.director_id WHERE movies.budget = 0;

7. Display the maximum number of votes received by a movie for each year since 2010.

SELECT SUBSTR(release_date, 1, 4) AS year, MAX(vote_count) AS max_votes FROM movies WHERE SUBSTR(release_date, 1, 4) >= '2010' GROUP BY SUBSTR(release_date, 1, 4);

8. For movies released after the year 2000, display the year and the minimum revenue generated in that year for movies with more than 10 votes. Exclude years where the minimum revenue was 0.

```
I SELECT SUBSTR(release_date, 1, 4) AS year, MIN(revenue) AS min_revenue
PROM movies

WHERE CAST(SUBSTR(release_date, 1, 4) AS INTEGER) > 2000

AND vote_count > 10

GROUP BY SUBSTR(release_date, 1, 4)

HAVING MIN(revenue) > 0;

8

9

10

11

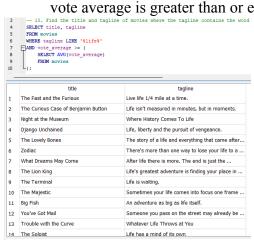
Execution finished without errors.
Result: 0 rows returned in 112ms
At line 1:
SELECT SUBSTR(release_date, 1, 4) AS year, MIN(revenue) AS min_revenue
FROM movies
```

SELECT SUBSTR(release_date, 1, 4) AS year, MIN(revenue) AS min_revenue FROM movies
WHERE CAST(SUBSTR(release_date, 1, 4) AS INTEGER) > 2000
AND vote_count > 10
GROUP BY SUBSTR(release_date, 1, 4)
HAVING MIN(revenue) > 0;

9. List the average budget of the movies for each year since 2000, showing only those years where the average budget was more than 2 million.

SELECT SUBSTR(release_date, 1, 4) AS year, AVG(budget) AS average_budget FROM movies
WHERE CAST(SUBSTR(release_date, 1, 4) AS INTEGER) >= 2000
GROUP BY SUBSTR(release_date, 1, 4)
HAVING AVG(budget) > 2000000;

10. Find the title and tagline of movies where the tagline contains the word 'life', and the vote average is greater than or equal to the average vote average of all movies.



11. Show the difference between the highest and lowest budget for movies released in '1997'.

```
12 -- 11. Show the difference between the highest and lowest
13 SELECT MAX(budget) - MIN(budget) AS budget_difference
14 FROM movies
15 WHERE strftime('%Y', release_date) = '1997';
15 budget_difference
1 200000000
```

12. Find the movies with a vote_average score within the top 10% of all movies, and order them by the number of votes they received.

13. Show the title of movies and their corresponding director's name, where the movie has more than 100 votes and its director has directed more than 5 movies in total.



14. List all movies where the second word in the title is 'Love', considering titles with more than one word.



15. Display the titles of movies that contain the director's last name within the movie title. Assume the last name starts after the last space in the director's name.

```
45 SELECT title
46 FROM movies
47 JOIN directors ON movies.director_id = directors.id
48 WHERE title LIKE '\(\frac{1}{2}\) | | SUBSTR(directors.name, INSTR(directors.name, '') + 1) || '\(\frac{1}{2}\)';

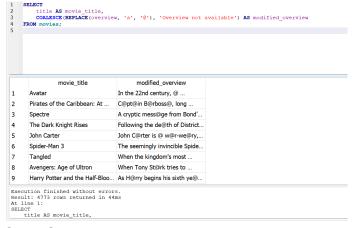
title

1 Coach Carter
2 An Alan Smithee Film: Burn, Hollywood, Burn
3 Martin Lawrence Live: Runteldat
```

16. List movies with 'NULL' in their overview or tagline, replacing 'NULL' with 'No data provided'.



17. Show the movie titles along with a modified overview where 'a' is replaced by '@', and if the overview is 'NULL', display 'Overview not available'.



SELECT

title AS movie_title, COALESCE(REPLACE(overview, 'a', '@'), 'Overview not available') AS modified_overview FROM movies;

18. Find directors with a last name starting with 'Mc' or 'Mac' and list their movies, replacing any occurrence of 'Mc' or 'Mac' in movie titles with 'Mc(Mac)'.



SELECT directors.name AS director_name, REPLACE(title, 'Mc', 'Mc(Mac)') AS movie_title FROM directors

JOIN movies ON directors.id = movies.director_id WHERE directors.name LIKE 'Mc%' OR directors.name LIKE 'Mac%';