



SQL Practice Assignment

February 4, 2025

1. List all movies released in the year 2000.

```
SELECT name FROM movies WHERE year = 2000;
```

2. Find the names of all directors who have directed a movie in the 'Action' genre.

```
SELECT DISTINCT d.first_name, d.last_name  
FROM directors d  
JOIN movies_directors md ON d.id = md.director_id  
JOIN movies_genres mg ON md.movie_id = mg.movie_id  
WHERE mg.genre = 'Action';
```

3. Get the total number of movies directed by each director.

```
SELECT d.first_name, d.last_name, COUNT(md.movie_id) AS total_movies  
FROM directors d  
JOIN movies_directors md ON d.id = md.director_id  
GROUP BY d.id;
```

4. Find all actors who have played a role in a movie with a rank greater than 8.

```
SELECT DISTINCT a.first_name, a.last_name  
FROM actors a  
JOIN roles r ON a.id = r.actor_id  
JOIN movies m ON r.movie_id = m.id  
WHERE m.rank > 8;
```

5. List all movies along with their genres.

```
SELECT m.name, mg.genre  
FROM movies m  
JOIN movies_genres mg ON m.id = mg.movie_id;
```

6. Find the top 5 directors with the highest average movie rank.

```

SELECT d.first_name, d.last_name, AVG(m.rank) AS avg_rank
FROM directors d
JOIN movies_directors md ON d.id = md.director_id
JOIN movies m ON md.movie_id = m.id
GROUP BY d.id
ORDER BY avg_rank DESC
LIMIT 5;

```

7. Get the names of all actors who have played more than 10 roles.

```

SELECT a.first_name, a.last_name
FROM actors a
JOIN roles r ON a.id = r.actor_id
GROUP BY a.id
HAVING COUNT(r.role) > 10;

```

8. Find all movies that have more than one genre.

```

SELECT m.name
FROM movies m
JOIN movies_genres mg ON m.id = mg.movie_id
GROUP BY m.id
HAVING COUNT(mg.genre) > 1;

```

9. List all directors and the number of genres they have directed movies in.

```

SELECT d.first_name, d.last_name, COUNT(DISTINCT mg.genre) AS num_genres
FROM directors d
JOIN movies_directors md ON d.id = md.director_id
JOIN movies_genres mg ON md.movie_id = mg.movie_id
GROUP BY d.id;

```

10. Find the movie with the highest rank.

```

SELECT name FROM movies ORDER BY rank DESC LIMIT 1;

```

11. List all actors who have acted in a movie directed by a specific director (e.g., director_id = 1).

```

SELECT DISTINCT a.first_name, a.last_name
FROM actors a
JOIN roles r ON a.id = r.actor_id
JOIN movies_directors md ON r.movie_id = md.movie_id
WHERE md.director_id = 1;

```

12. Get the average rank of movies for each genre.

```

SELECT mg.genre, AVG(m.rank) AS avg_rank
FROM movies_genres mg
JOIN movies m ON mg.movie_id = m.id
GROUP BY mg.genre;

```

13. Find all movies that have the same name but were released in different years.

```

SELECT name, COUNT(DISTINCT year) AS num_years
FROM movies
GROUP BY name
HAVING num_years > 1;

```

14. List all directors who have directed movies in more than one genre.

```
SELECT d.first_name, d.last_name
FROM directors d
JOIN movies_directors md ON d.id = md.director_id
JOIN movies_genres mg ON md.movie_id = mg.movie_id
GROUP BY d.id
HAVING COUNT(DISTINCT mg.genre) > 1;
```

15. Find the actor who has acted in the most movies.

```
SELECT a.first_name, a.last_name, COUNT(r.movie_id) AS num_movies
FROM actors a
JOIN roles r ON a.id = r.actor_id
GROUP BY a.id
ORDER BY num_movies DESC
LIMIT 1;
```

16. Get the total number of movies in each genre.

```
SELECT genre, COUNT(movie_id) AS num_movies
FROM movies_genres
GROUP BY genre;
```

17. List all movies that do not have a genre assigned.

```
SELECT m.name
FROM movies m
LEFT JOIN movies_genres mg ON m.id = mg.movie_id
WHERE mg.genre IS NULL;
```

18. Find the director who has directed the most movies.

```
SELECT d.first_name, d.last_name, COUNT(md.movie_id) AS num_movies
FROM directors d
JOIN movies_directors md ON d.id = md.director_id
GROUP BY d.id
ORDER BY num_movies DESC
LIMIT 1;
```

19. Get the names of all movies and their directors released after the year 2000.

```
SELECT m.name, d.first_name, d.last_name
FROM movies m
JOIN movies_directors md ON m.id = md.movie_id
JOIN directors d ON md.director_id = d.id
WHERE m.year > 2000;
```

20. Find the genre with the highest average movie rank.

```
SELECT mg.genre, AVG(m.rank) AS avg_rank
FROM movies_genres mg
JOIN movies m ON mg.movie_id = m.id
GROUP BY mg.genre
ORDER BY avg_rank DESC
LIMIT 1;
```

21. Using a CTE, find the average rank of movies for each director and list directors with an average rank greater than 8.

```
WITH DirectorAvgRank AS (  
    SELECT d.id, d.first_name, d.last_name, AVG(m.rank) AS avg_rank  
    FROM directors d  
    JOIN movies_directors md ON d.id = md.director_id  
    JOIN movies m ON md.movie_id = m.id  
    GROUP BY d.id  
)  
SELECT first_name, last_name, avg_rank  
FROM DirectorAvgRank  
WHERE avg_rank > 8;
```

22. Using a CTE, find the number of movies each actor has acted in and list actors who have acted in more than 5 movies.

```
WITH ActorMovieCount AS (  
    SELECT a.id, a.first_name, a.last_name, COUNT(r.movie_id) AS num_movies  
    FROM actors a  
    JOIN roles r ON a.id = r.actor_id  
    GROUP BY a.id  
)  
SELECT first_name, last_name, num_movies  
FROM ActorMovieCount  
WHERE num_movies > 5;
```

23. Using window functions, rank movies within each genre based on their rank.

```
SELECT m.name, mg.genre, m.rank,  
       RANK() OVER (PARTITION BY mg.genre ORDER BY m.rank DESC) AS genre_rank  
FROM movies m  
JOIN movies_genres mg ON m.id = mg.movie_id;
```

24. Using window functions, find the top 3 highest-ranked movies for each genre.

```
WITH RankedMovies AS (  
    SELECT m.name, mg.genre, m.rank,  
           RANK() OVER (PARTITION BY mg.genre ORDER BY m.rank DESC) AS genre_rank  
    FROM movies m  
    JOIN movies_genres mg ON m.id = mg.movie_id  
)  
SELECT name, genre, rank  
FROM RankedMovies  
WHERE genre_rank <= 3;
```

25. Using a CTE and window functions, calculate the cumulative number of movies directed by each director over the years.

```
WITH DirectorMovies AS (  
    SELECT d.id, d.first_name, d.last_name, m.year, COUNT(m.id) AS num_movies  
    FROM directors d  
    JOIN movies_directors md ON d.id = md.director_id  
    JOIN movies m ON md.movie_id = m.id  
    GROUP BY d.id, m.year  
)
```

```
SELECT first_name, last_name, year,  
       SUM(num_movies) OVER (PARTITION BY id ORDER BY year) AS cumulative_movies  
FROM DirectorMovies;
```