

## Overview

In this project, I delved into a movie database with Oracle SQL Developer. I tackled 52 SQL queries at basic, intermediate, and advanced levels to enhance my SQL skills and delve into the film industry's actors, directors, genres, and ratings. These queries focused on data retrieval, joins, aggregation, and pattern analysis, providing hands-on experience in movie data analysis within practical scenarios.

## Objectives

This project aims to analyze a movie dataset in an Oracle SQL database to derive insights on trends, actor collaborations, genre popularity, director performance, and viewer ratings through SQL queries. The analysis assists entertainment businesses, data journalists, and researchers in understanding film industry patterns.

## Project Scope

Analyze actor-director collaborations, genre popularity and ratings, top-rated movies by genre and year, prolific contributors' rankings, and reviewer demographics and rating behavior.

## Tools and Technologies

- Database Management System: Oracle SQL Developer
- Query Language: SQL

## Dataset Description

The movie dataset is divided into normalized tables to maintain data integrity and minimize redundancy. Here is a short overview of each table.

### movie\_actor:

- act\_id – this is a unique ID for each actor
- act\_fname – this is the first name of each actor
- act\_lname – this is the last name of each actor
- act\_gender – this is the gender of each actor

### movie\_genres:

- gen\_id – this is a unique ID for each genres
- gen\_title – this is the description of the genres

### movie\_genres:

- mov\_id – this is the ID of the movie, which is referencing the mov\_id column of the table movie
- gen\_id – this is the ID of each genres, which is referencing the gen\_id column of the table genres

## **movie\_director:**

- `dir_id` – this is a unique ID for each director
- `dir_fname` – this is the first name of the director
- `dir_lname` – this is the last name of the director

## **movie\_movie:**

- `mov_id` – this is the unique ID for each movie
- `mov_title` – this column represents the name of the movie
- `mov_year` – this is the year of making the movie
- `mov_time` – duration of the movie i.e. how long it was running
- `mov_lang` – the language in which movie was casted
- `mov_dt_rel` – this is the release date of the movie
- `mov_rel_country` – this is the name of the country(s) where the movie was released

## **movie\_direction:**

- `dir_id` – this is the ID for each director, which is referencing the `dir_id` column of the table director
- `mov_id` – this is the ID of the movie, which is referencing the `mov_id` column of the table movie

## **movie\_reviewer:**

- `rev_id` – this is the unique ID for each reviewer
- `rev_name` – this is the name of the reviewer

## **movie\_rating:**

- `mov_id` – this is the ID of the movie, which is referencing the `mov_id` column of the table movie
- `rev_id` – this is the ID of the reviewer, which is referencing the `rev_id` column of the table reviewer
- `rev_stars` – this indicates how many stars a reviewer rated for a review of a movie
- `num_o_rating` – this indicates how many ratings a movie achieved till date

## **movie\_cast:**

- `act_id` – this is ID of actor, which is referencing the `act_id` column of actor table
- `mov_id` – this is the ID of the movie, which is referencing the `mov_id` column of the table movie
- `role` – this is the name of a character in the movie, an actor acted for that character

### Queries (Solution: check the *movie\_data\_analysis.sql* file)

Query-1: Write a SQL query to find the name and year of the movies.

Return movie title, movie release year with name of the month and day also.

Query-2: Write a SQL query to find when the movie 'American Beauty' is made.

Return the year of making the movie.

Query-3: Write a SQL query to find the movie that was made in 1999. Return the movie title with ID.

Query-4: Write a SQL query to find those movies, which were released before 1998.

Return movie title with released date.

Query-5: Write a SQL query to find the name of all reviewers and movies together in a single list.

Query-6: Write a SQL query to find all reviewers who have rated 7 or more stars to their rating.

Return reviewer name with their rating.

Query-7: Write a SQL query to find the movies with their rating is NULL.

Return movie title with their rating is NULL.

Query-8: Write a SQL query to find the movies without any rating. Return movie title.

Query-9: Write a SQL query to find the movies with ID 905 or 907 or 917. Return the movie title with their ID.

Query-10: Write a SQL query to find the movie titles that contain the word 'Boogie Nights'.

Sort the result-set in ascending order by the movie making a year.

Return movie ID, movie title, movie making year and movie release date.

Query-11: Write a SQL query to find those actors with the first name 'Woddy' and the last name 'Allen'.

Return details.

Query-12: Write a SQL query to find the actors who played a role in the movie 'Annie Hall'.

Return all the fields of the actor table.

Query-13: Write a SQL query to find the director of a film that cast a role in 'Miss Giddens'.

Return director ID and full name.

Query-14: Write a SQL query to find the actor of a film in 'Eyes Wide Shut'.

Return actor ID and full name.

Query-15: Write a SQL query to find those movies that have been release in countries other than the United Kingdom. Return movie title, movie making year, movie time, date of release and releasing country.

Query-16: Write a SQL query to find for movies whose reviewer is NULL.

Return movie title, movie making year, date of release, the full name of director and actor.

Query-17: Write a SQL query to find those movies directed by Woody Allen. Return movie title.

- Query-18: Write a SQL query to determine those years in which there was at least one movie that received a rating of at least three stars. Sort the result set in ascending order by movie year.  
Return to the movie year.
- Query-19: Write a SQL query to search for movies that do not have any ratings. Return movie title.
- Query-20: Write a SQL query to find those reviewers who have not given a rating of certain films.  
Return reviewer name.
- Query-21: Write a SQL query to find movies that have been reviewed by a reviewer and received a rating.  
Sort the result set ascending order by reviewer name, movie title, review stars.  
Return reviewer name, movie title, review stars.
- Query-22: Write a SQL query to find movies that have been reviewed by a reviewer and received a rating.  
Group the result set on the reviewer's name, movie title. Return reviewer's name, movie title.
- Query-23: Write a SQL query to find those movies, which have received the highest number of stars.  
Group the result set on the movie title and sort the result set in ascending order by the movie title.  
Return movie title and maximum number of review stars.
- Query-24: Write a SQL query to find all reviewers who rated the movie 'American Beauty'.  
Return reviewer name.
- Query-25: Write a SQL query to find the movies that have not been reviewed by any reviewer body other than 'Paul Monks'. Return movie title.
- Query-26: Write a SQL query to find the movies with the lowest ratings.  
Return reviewer name, movie title and number of stars for those movies.
- Query-27: Write a SQL query to find the movies directed by 'James Cameron'. Return movie title.
- Query-28: Write a SQL query to find the movies in which one or more actors appeared in more than one film.
- Query-29: Write a SQL query to find all reviewers whose ratings contain a NULL value.  
Return reviewer name.
- Query-30: Write a SQL query to find out who was cast in the movie 'Annie Hall'.  
Return the full name of the actor and role.
- Query-31: Write a SQL query to find the director who directed a movie that featured a role in 'Elizabeth Darko'. Return the name of the director and movie.
- Query-32: Write a SQL query to find the director of a movie is Eyes Wide Shut.  
Return the full name of director and role.
- Query-33: Write a SQL query to find out which actors have not appeared in any movies between 1990 and 2000 (Begin and end values are included). Return the full name of the actor, movie title and date of release.

Query-34: Write a SQL query to find the directors who have directed films in a variety of genres.

Group the result set on director first name, last name and generic title.

Sort the result set in ascending order by director first name and last name.

Return director first name, last name, generic title and number of genres of movies.

Query-35: Write a SQL query to find the movies with year and genres.

Return movie title, movie make year and generic title.

Query-36: Write a SQL query to find all the movies with year, genres and name of the director.

Query-37: Write a SQL query to find the movies released before 1978.

Sort the result set in descending order by movie make year.

Return movie title, movie year, date of release, duration and the name of the director.

Query-38: Write a SQL query to calculate the average movie length and count the number of movies in each genre. Return genre title, average time and number of movies for each genre.

Query-39: Write a SQL query to find the movies with the shortest duration.

Return movie title, movie year, director name, actor name and role.

Query-40: Write a SQL query to find the years in which a movie received a rating of 3 or 4.

Sort the result in increasing order on movie year.

Query-41: Write a SQL query to get the reviewer name, movie title, and stars in an order that reviewer name will come first, then by movie title, and lastly by number of stars.

Query-42: Write a SQL query to find those movies that have at least one rating and received the most stars.

Sort the result-set on the movie title. Return movie title and maximum review stars.

Query-43: Write a SQL query to find out which movies have received a number of ratings.

Return movie title, director name and number of ratings.

Query-44: Write a SQL query to find movies in which one or more actors have acted in more than one film.

Return movie title, actor name, and the role.

Query-45: Write a SQL query to find the actor whose first name is 'Claire' and last name is 'Danes'.

Return movie title, the full name of director and actor and their role.

Query-46: Write a SQL query to find for actors whose films have been directed by them.

Return actor name, movie title and role.

Query-47: Write a SQL query to find the cast list of the movie 'Chinatown'. Return the name of the actor.

Query-48: Write a SQL query to find those movies where the actor's name is 'Harrison Ford'.

Return movie title.

Query-49: Write a SQL query to find the highest-rated movies.

Return movie title, movie year, review stars and releasing country.

Query-50: Write a SQL query to find the highest-rated 'Thriller' movies.

Return the movie title, movie year and rating.

Query-51: Write a SQL query to find the years when most of the 'Crime' movies produced.

Count the number of generic titles and compute their average rating.

Group the result set on movie release year, generic title.

Return movie year, generic title, number of generic titles and average rating.

Query-52: Write a SQL query to generate a report, which contain the fields movie title,

name of the female actor, year of the movie, role, movie genres, the director, date of release, and the number of ratings for that movie.

## Challenges and Solutions

To address complex relationships, multiple joins were implemented with normalized keys for data consistency. Aggregate functions such as AVG() with FILTER clauses handled missing ratings, and DISTINCT and GROUP BY clauses prevented genre duplication in multi-genre movies.

## Future Works

Future works include data visualization with Python (Matplotlib/Seaborn) or Power BI, advanced SQL queries using window functions and CTEs, and predictive analytics to forecast sales trends with machine learning.

## Conclusion

This project shows how to analyze movie datasets using Oracle SQL. By utilizing normalized tables, strong SQL queries, and effective schema navigation, it uncovers insights for content strategy, production planning, and audience engagement in the movie industry.

**\*\*\*End of Document\*\*\***