**TITLE : IMDb Dataset Analysis Using Advanced SQL**

**NAME : M.SUDHARSANAN**

**1. Introduction**

The IMDb dataset contains extensive information about movies, directors, actors, and ratings. The purpose of this project is to analyze the dataset using SQL to extract meaningful insights, such as identifying the highest-rated movies, most frequent directors, and popular genres. This project utilizes **MySQL 8.0.40** and **MySQL Workbench** to store, manage, and query the data efficiently.

**2. Database Setup**

1. **Importing the Dataset** – The IMDb dataset was loaded into MySQL using the IMDB Data Import.sql script.
2. **Creating Tables** – Tables such as Movie, Genre, Names, Ratings, Director\_Mapping, and Role\_Mapping were created and populated.
3. **Verification** – The successful import was verified using commands like:
4. SHOW DATABASES;
5. USE IMDB;
6. SHOW TABLES;

SELECT \* FROM Movie LIMIT 5;

1. **Data Integrity Check** – Checking for missing values, duplicates, and inconsistencies.

**3. Data Exploration & Analysis**

After setting up the database, exploratory queries were executed to understand the structure and distribution of the data:

* **Checking the total number of records in each table:**
* SELECT COUNT(\*) FROM Movie;

SELECT COUNT(\*) FROM Genre;

* **Identifying unique genres and their distribution:**
* SELECT Genre, COUNT(\*) AS Total\_Movies

FROM Genre GROUP BY Genre ORDER BY Total\_Movies DESC;

* **Finding the highest-rated movies:**
* SELECT M.Title, R.Avg\_Rating
* FROM Movie M
* JOIN Ratings R ON M.Movie\_ID = R.Movie\_ID
* ORDER BY R.Avg\_Rating DESC

LIMIT 10;

* **Identifying the most frequent directors:**
* SELECT N.Name, COUNT(D.Movie\_ID) AS Movies\_Directed
* FROM Director\_Mapping D
* JOIN Names N ON D.Person\_ID = N.Person\_ID
* GROUP BY N.Name ORDER BY Movies\_Directed DESC

LIMIT 5;

**4. Challenges & Solutions**

Several challenges were encountered during the project, including:

* **MySQL Connection Issues** – Resolved by checking server configurations and restarting MySQL.
* **Data Inconsistencies** – Some records had missing values, which were handled using IS NULL queries and filtering.
* **Performance Optimization** – Indexing key columns like Movie\_ID improved query performance.

**5. Conclusion & Future Scope**

This analysis provided insights into the most successful movies, genres, and directors within the IMDb dataset. Future improvements could include:

* Performing time-based trend analysis on movie ratings.
* Integrating visualization tools like Tableau or Power BI for enhanced data representation.
* Expanding the dataset with additional attributes for deeper insights.