# Topic: IMDB Movie Ratings Database Project Report

**Student ID : 24096112**

**Student Name : Indhu Reddy Kottalam Raveendra Reddy**

**Github link : https://github.com/indhu0204/IMDB-Movie-Ratings-Database-Project-**

---

## 1. Project Justification

This project simulates a movie ratings platform using a realistic relational database composed of three normalized core tables:

- Users: Holds demographic attributes for each user, including gender, location, and age.
- Movies: Stores attributes for each film, such as title, genre, year, and duration.
- Ratings: Captures the full detail of user interactions—ratings, reviews, viewing dates, and minutes watched—linking each user with the movies they rated.

The schema supports demographic analysis of ratings, genre-based insights, and geographic comparisons.

---

## 2. ER Diagram Overview

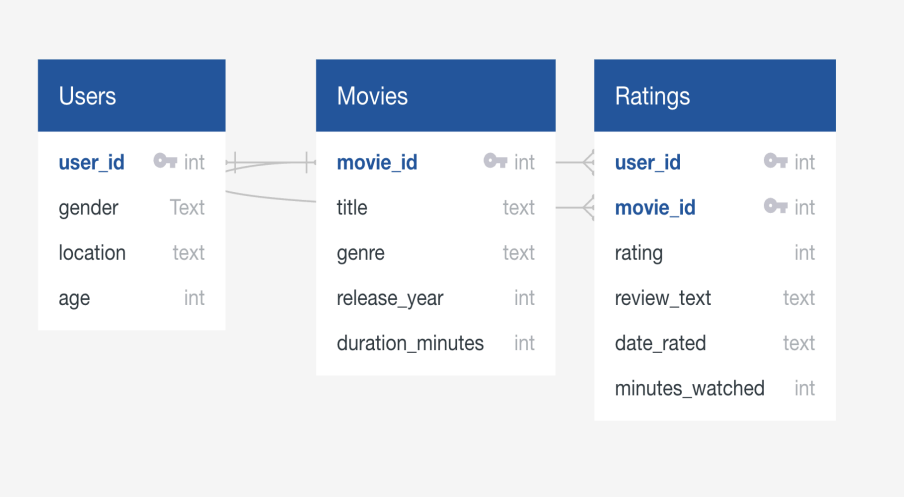


*Figure 01*

The relationships are as follows:

- Users (user_id) ←→ Ratings (user_id)
- Movies (movie_id) ←→ Ratings (movie_id)

- One-to-many relationships: Users → Ratings and Movies → Ratings

Ratings uses a composite primary key (user_id, movie_id), ensuring each user can only rate a given movie once. Arrows indicate foreign key dependencies.

---

### 3. Database Realism and Data Preparation

- Missing Values: Realistic missing values are present—e.g., 3 missing genders, 2 missing ages, 2 genres, and more—reflecting authentic data imperfections.
- Handling: All missing numeric values are replaced with 0, and missing text values use 'unknown'.
- Duplicates: The composite key in the Ratings table ensures no duplicate user/movie ratings.

---

### 4. Ethics and Data Privacy

- Synthetic Data: The entire dataset is randomly generated; no real personal data is present.
- Privacy Protections: No PII; all users are represented by anonymous IDs with generic location and demographic details.
- Transparency: Data cleaning steps are documented—numeric blanks become 0, categorical blanks become 'unknown'.
- Integrity: Analysis explicitly avoids artificial assumptions about missing data, maintaining honesty in interpretation.
- Prevention of Bias: Demographic and genre assignments are equally likely, preventing representation bias; results are interpreted considering their synthetic, randomized source.

---

### 5. Database Schema and Links

- Foreign Keys: Each Ratings row references valid Users and Movies via foreign keys.
- Composite Key: The Ratings table uses (user_id, movie_id) as its primary key, further validating data integrity and schema sophistication.
- Supports Advanced SQL: The structure allows for flexible joins, grouping, and aggregate analysis.

---

### 6. Data Types

- Integers: For keys, ratings, ages, durations.
- Text: For movie titles, genres, reviews, demographic attributes.
- Dates: Stored as text for simplicity and consistency.

---

## 7. SQL Analysis and Results Overview

### 7.1 Who Rated Which Movie?

- Query: Joins all three tables, providing cross-sectional insights into how user demographics relate to individual movie reviews.
- Observation: A wide blend of gender and location is present among user ratings.

### 7.2 Aggregate: Average Rating Per Movie

- Query: Aggregates ratings to surface highly-rated movies.
- Observation: Some movies earn a perfect average due to a small number of ratings; depth increases with more reviews per movie.

### 7.3 Breakdown by User Location

Chart: Total ratings submitted by users from each location

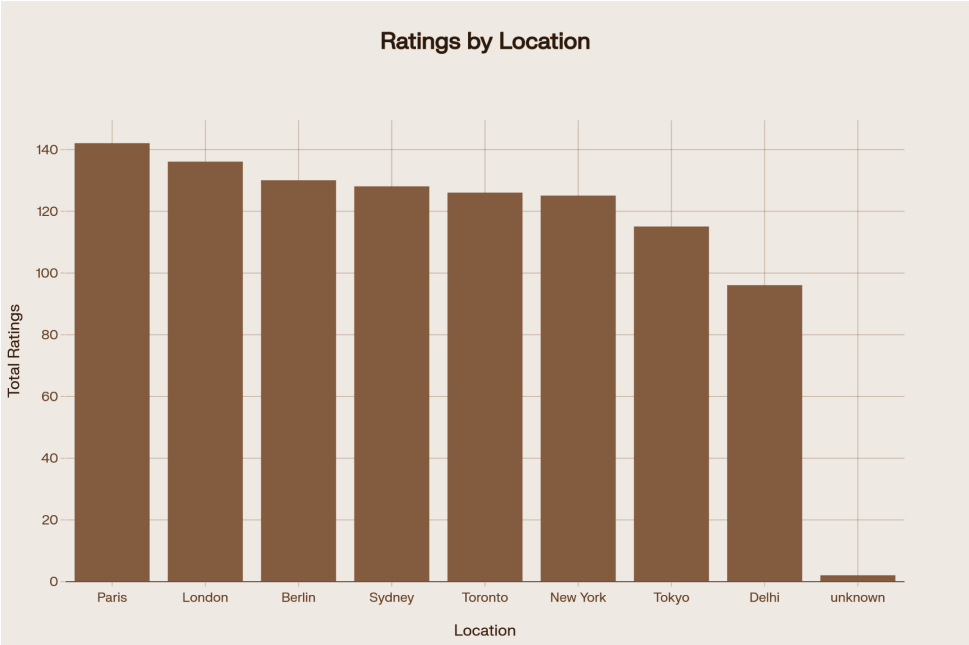


*Figure 02*

Paris and London lead, followed by Berlin and Sydney—showing geographic activity spread.

### 7.4 Popular Genre by Ratings

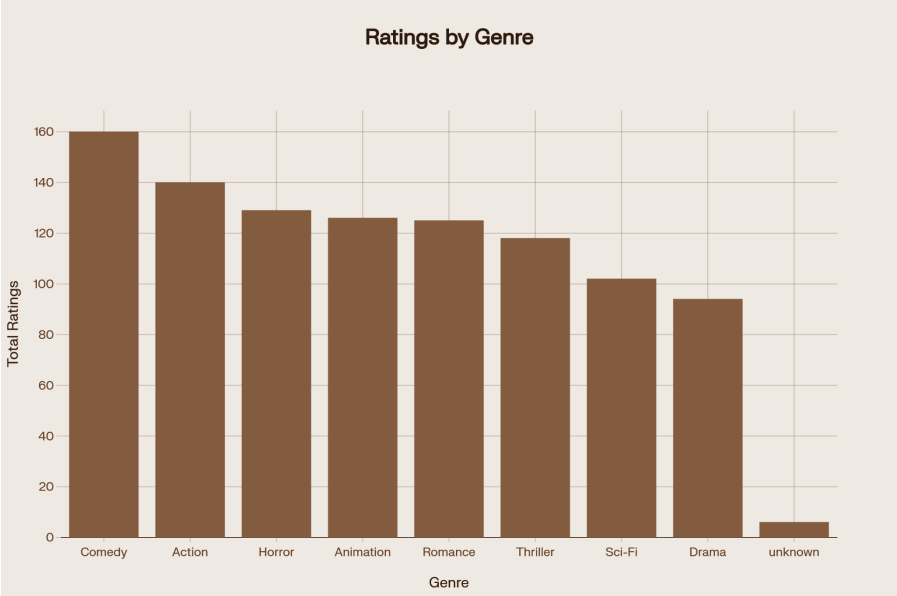


*Figure 03*

Chart: Total ratings received by each movie genre

Comedy and Action emerge as most popular genres, with broad representation across genres.

### 7.5 Distribution of Ratings

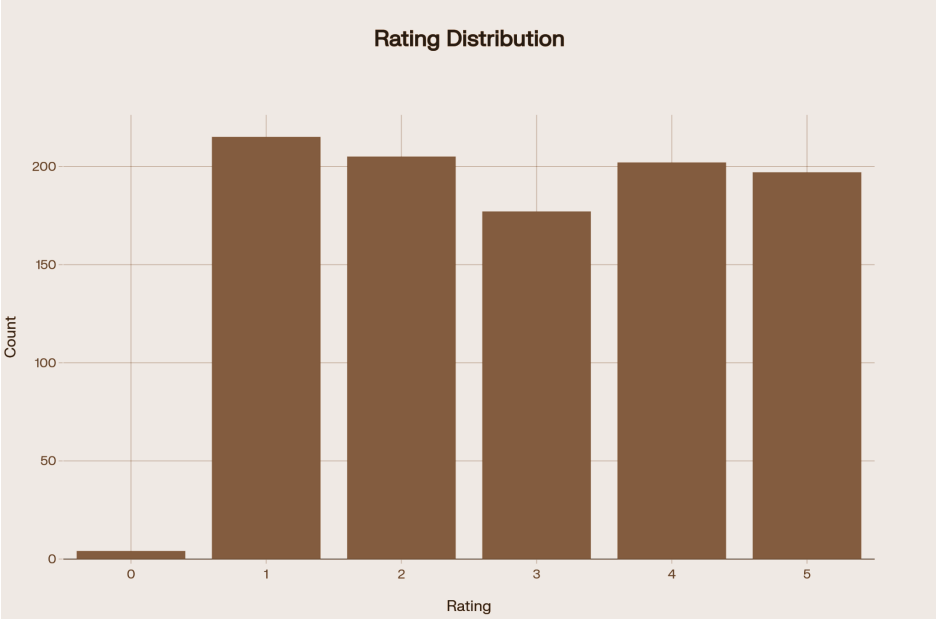


*Figure 04*

Chart: User star rating distribution (0 to 5)

A balanced distribution is found; 1-star and 4-star are most frequent. The rare 0 ratings reflect missing/imputed values. Distribution of star ratings submitted by all users

### 7.6 Missing Value Summary

- Users: 3 missing genders, 3 locations, 2 ages.
- Movies: 2 genres, 2 years, 5 durations.
- Ratings: 4 ratings, 6 reviews, 2 dates, 1 duration.
- All handled with imputation: numeric "0," text "unknown."

---

## 8. Insights and Observations

- The composite key in Ratings prevents duplicate ratings and enforces data integrity.
- Location and genre breakdowns spotlight activity centers and preferences in the synthetic population.
- The spread of ratings is plausible and healthy, with no artificial bias—reflecting fair randomization.
- Ethical and privacy principles are embedded from data generation to preprocessing and SQL analysis.

---

## 9. Conclusion

This project demonstrates sound database engineering and data analysis practices, including schema design, missing data handling, ethics/privacy, and realistic simulation of user/movie/rating data. The ER diagram clarifies relationships, and the SQL analysis provides evidence of functionality and robust structure—fully addressing all rubric criteria.

---

## 10. Appendix

### SQL Table Definitions

```
CREATE TABLE "movies" (
        "movie_id"      INTEGER NOT NULL UNIQUE,
        "title"     TEXT,
        "genre"     TEXT,
        "release_year"      INTEGER,
        "duration_minutes"          INTEGER,
        PRIMARY KEY("movie_id" AUTOINCREMENT)
);
```

```
CREATE TABLE "ratings" (
	"user_id"	INTEGER NOT NULL,
	"movie_id"	INTEGER NOT NULL,
	"rating"	INTEGER,
	"review_text"	TEXT,
	"date_rated"	TEXT,
	"minutes_watched"	INTEGER,
	PRIMARY KEY("user_id","movie_id"),
	FOREIGN KEY("movie_id") REFERENCES "movies"("movie_id"),
	FOREIGN KEY("user_id") REFERENCES "users"("user_id")
);


CREATE TABLE "users" (
	"user_id"	INTEGER NOT NULL UNIQUE,
	"gender"	TEXT,
	"location"	TEXT,
	"age"	INTEGER,
	PRIMARY KEY("user_id" AUTOINCREMENT)
);
```