**Netflix User Analytics Database Project**

## **Group Members**

Mehreen Ali Gillani

Paula Brown

Taha Malik

## **Collaboration Tools**

## **Communication**

* **Zoom**: daily team meetings, brainstorming sessions, and collaborative work
* **Email**: Formal communications, meeting summaries, and document sharing

### **Code & Data Management**

* **GitHub**: Version control for all SQL scripts, R analysis code, and project documentation

### **Project Documentation**

* **GitHub Wiki:** Project requirements, meeting notes, and technical specifications
* **Markdown Files:** Code documentation and analysis explanations

## **Data Sources**

### **Primary Dataset**

**Source**: Kaggle - Netflix User Behavior Dataset

**URL**: <https://www.kaggle.com/datasets/sayeeduddin/netflix-2025user-behavior-dataset-210k-records>

**Records**: 210,000+ user behavior records

**Content**: User demographics, movies, watch history, and recommendation\_logs

**Data Loading Methods**

1. MySQL Workbench Import Wizard for initial data loading
2. R Studio with RMySQL package for data transformation and analysis
3. SQL LOAD DATA INFILE for bulk database imports

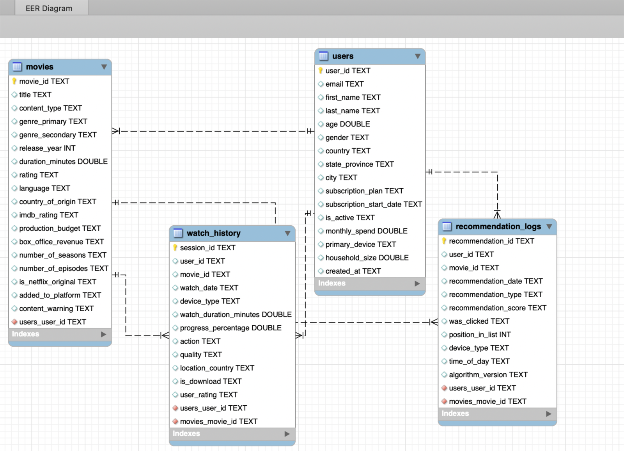
## **Database Design**

### **Logical Model - Normalized Schema**

### **Entities:**

1. **users** - Customer demographic and subscription information
2. **movies** - Content catalog and metadata
3. **watch\_history** - User viewing behavior and engagement
4. **recommendation\_logs** - Recommendation engine interactions

## **Entity-Relationship Diagram**



### **Relationship Definitions**

1. **users** → **watch\_history** (1:M)
   1. One user can have multiple watch history records
   2. **Primary Key:** users.user\_id → **Foreign Key:** watch\_history.user\_id
2. **movies** → **watch\_history** (1:M)
   * One movie can be watched by multiple users
   * **Primary Key:** movies.movie\_id → **Foreign Key:** watch\_history.movie\_id
3. **users** → **recommendation\_logs** (1:M)
   * One user can receive multiple recommendations
   * **Primary Key:** users.user\_id → **Foreign Key:** recommendation\_logs.user\_id
4. **movies** → **recommendation\_logs** (1:M)
   * One movie can be recommended to multiple users
   * **Primary Key:** movies.movie\_id → **Foreign Key:** recommendation\_logs.recommended\_movie\_id

## **Next Steps**

1. Data cleaning and Exploratory data analysis
2. Develop analytical queries in SQL
3. Create R scripts for statistical analysis
4. Implement movie recommendation system
5. Document findings and insights

## **Data Cleaning & EDA Plan**

* Remove or impute missing values
* Standardize data types for database import
* Deduplicate records in watch\_history
* Perform EDA: summary statistics, distribution plots, correlations

## **Key Analytical Questions**

* What user demographics most influence viewing habits?
* Which genres are most frequently recommended and watched?
* How effective is the recommendation system for different user groups?

## **Backup & Recovery Plan**

* All code and documentation version-controlled via GitHub
* Weekly MySQL database dumps for backup
* Shared cloud drive for redundant storage