Capital University of Science and Technology

Department of Software Engineering

SE2313 – Introduction to Database Systems

Project Report

Group Members:

Muhammad Saad Imran (BSE233117),

Abdul Waseh (BSE233073),

Zohaib Fayyaz (BSE233069).

Section: 02.

Date: January **19th** ,2025.

Submitted To: Mam Hina Rashid.

GITHUB
https://github.com/AbdulWaseh46/MovieRecommendation_DBMS

Table of Contents:

➢ Introduction: 3

1. Purpose and Overview: 3
2. Key Features: 3

➢ Database Development Life Cycle: 4

1. Database Planning: 4
2. System Definition: 5
3. Requirement Collection and Analysis: 6
4. Database Design: 8
5. Application Design: 11
6. DBMS Selection: 12
7. Prototyping: 12
8. Implementation: 13
9. Data Conversion and Loading: 14
10. Testing: 15
11. Operational Maintenance: 16

➢ Queries: 16

1. SQL Queries: 16
2. Relational Algebra Queries: 21

MOVIE RECOMMENDATION SYSTEM

➢ Introduction:

In today's world, there are so many streaming services and movies to choose from that picking the right movie can feel overwhelming. To make this easier, a Movie Recommendation System has been created. This system looks at user data, like what movies they've watched, their preferences, and the ratings they've given, to suggest movies they might enjoy.

This report explains how the system was built, step by step. It covers how the database was planned and designed, how the system was put together, and how it was tested to make sure it works well.

1. Purpose and Overview:

The Movie Recommendation System is designed to make movie-watching more enjoyable. It gives personalized movie suggestions, helps users discover different genres, and lets them share recommendations with friends. The system adjusts to each user's preferences and offers an easy-to-use, engaging interface.

2. Key Features:

- **Tracking what users watch and like:**

  The system keeps track of what users watch and their preferences to give better suggestions.

- **Collecting and using movie ratings:**

  User ratings help improve the recommendations.

- **Personalized movie suggestions:**

  The system uses smart methods to recommend movies based on what users like and similar user preferences.

- **Browsing movies by genres and categories:**

Users can explore movies by different genres and categories easily.

- **Real-time updates:**

- Recommendations change and improve based on what users do.

- **Spelling correction for searches:**

Fixes typos to make searching easier and more accurate.

- **Sharing movie recommendations:**

Users can share their favourite movie suggestions with friends for fun and interaction.

➢ Database Development Life Cycle:

1. Database Planning:

Goal: The goal of database planning is to organize the information that the movie recommendation system will use and store. The system needs a well-structured database to handle and manage all the data related to users, movies, ratings, and recommendations in an efficient way.

Key Parts:

- **User Information:**

This includes users' basic information, such as their email, preferences, and usernames, which will be saved in the Users table.

- **Movies:**

The Movies table will store movie details like title, release year, description, cast, and runtime.

- **Ratings:**

The Ratings table will store how users rate movies, along with the time they rated the movie and any reviews they provide.

- **Genres:**

The Genres table will store different movie genres and their popularity.

- **Viewing History:**

The Viewing History table will keep track of what movies users have watched, including when they watched them, for how long, and on which device.

- **Recommendations:**

The Recommendations table will store the movie suggestions generated for each user.

Features for Planning:

- **Tracking User Preferences:** The system will save what movies users like and their ratings, so it can give better recommendations.
- **Personalized Movie Suggestions:** It will recommend movies based on the movies that users have watched and rated highly.
- **Genre Browsing:** Users can explore movies by genre to find more options.
- **Auto Spelling Correction:** The system will fix common spelling errors when users enter data, making it more accurate.
- **Sharing Recommendations:** Users can share their favourite movies with friends.

Real-Time Updates: The system will update recommendations and data based on what the user does, keeping everything current.

2. System Definition:

Purpose of the System: The movie recommendation system's goal is to help users quickly find movies they will enjoy. It suggests movies based on each user's preferences, viewing history, and ratings. It also allows users to share movie suggestions with others and browse movies by genre.

System Features:

- **User Profiles:** The system will store each user's preferences, movie ratings, and what movies they've watched.
- **Movie Database:** A large collection of movie information, including details like title, genre, and user ratings.
- **Recommendation Engine:** The system will suggest movies based on what users have rated and watched, using methods that take into account user preferences.
- **Genre Browsing:** Users can filter and explore movies by genres like action, comedy, etc.
- **Real-Time Updates:** The recommendations will change based on what a user watches or rates recently.
- **Spelling Correction:** Automatically fix typing errors when users search for movies or enter ratings.
- **Social Features:** Users can share movie recommendations with friends or others, creating a community.

3. Requirement Collection and Analysis:

Methods for Gathering Information:

1. **Interviews:**
    - Interviews were conducted with potential users, both movie lovers and casual viewers.
    - **Findings:** Users wanted personalized recommendations based on their preferences, easy browsing, and a way to rate movies.
2. **Questionnaires:**
    - A survey was shared with users to understand their movie-watching habits and preferences.
    - **Findings:** Most respondents enjoy watching action and animated movies. They also wanted a system that could suggest movies based on ratings from other users.
3. **User Observation:**
    - Observing how people interact with current movie platforms helped to spot frustrations and difficulties in finding movies.
    - **Findings:** Users spend a lot of time scrolling through long lists without finding what they like and often rely on external reviews.

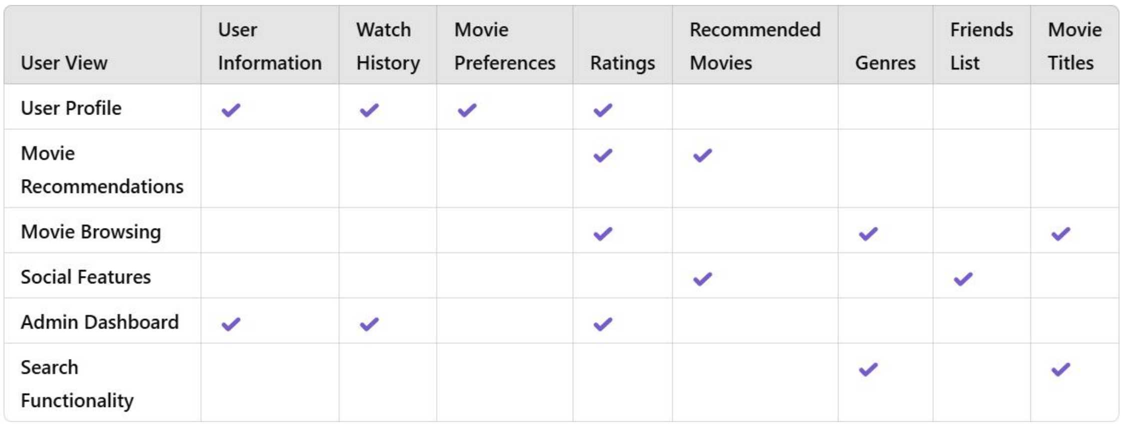
Key Requirements:

- **User Needs:** Users want personalized movie suggestions, an easy way to search for movies, and a way to share recommendations.
- **System Features:** The system should have user profiles, a recommendation engine, real-time updates, and auto-correction for misspelled inputs.
- **Data Management:** A solid database design is needed to store and manage all the information about users, movies, ratings, and movie-watching history.

- **User view:**

User View	**Main Types of Data Used**
User Profile	User Information (Name, email) Watch History Movie Preferences Rating
Movie Recommendations	Recommended Movies Rating Genres Watch History
Movie Browsing	Movie Titles Genres Ratings
Social Features	User Profile Recommended Movies Friends list
Admin Dashboard	Users Table Movies table Ratings Table Watch history Tables Recommendations Table
Search Functionality	Movie Titles Genres Ratings

Cross-Reference Table:



User View	User Information	Watch History	Movie Preferences	Ratings	Recommended Movies	Genres	Friends List	Movie Titles
User Profile	✓	✓	✓	✓				
Movie Recommendations				✓	✓			
Movie Browsing				✓		✓		✓
Social Features					✓		✓	
Admin Dashboard	✓	✓		✓				
Search Functionality						✓		✓

4. Database Design:

Entities and Attributes:

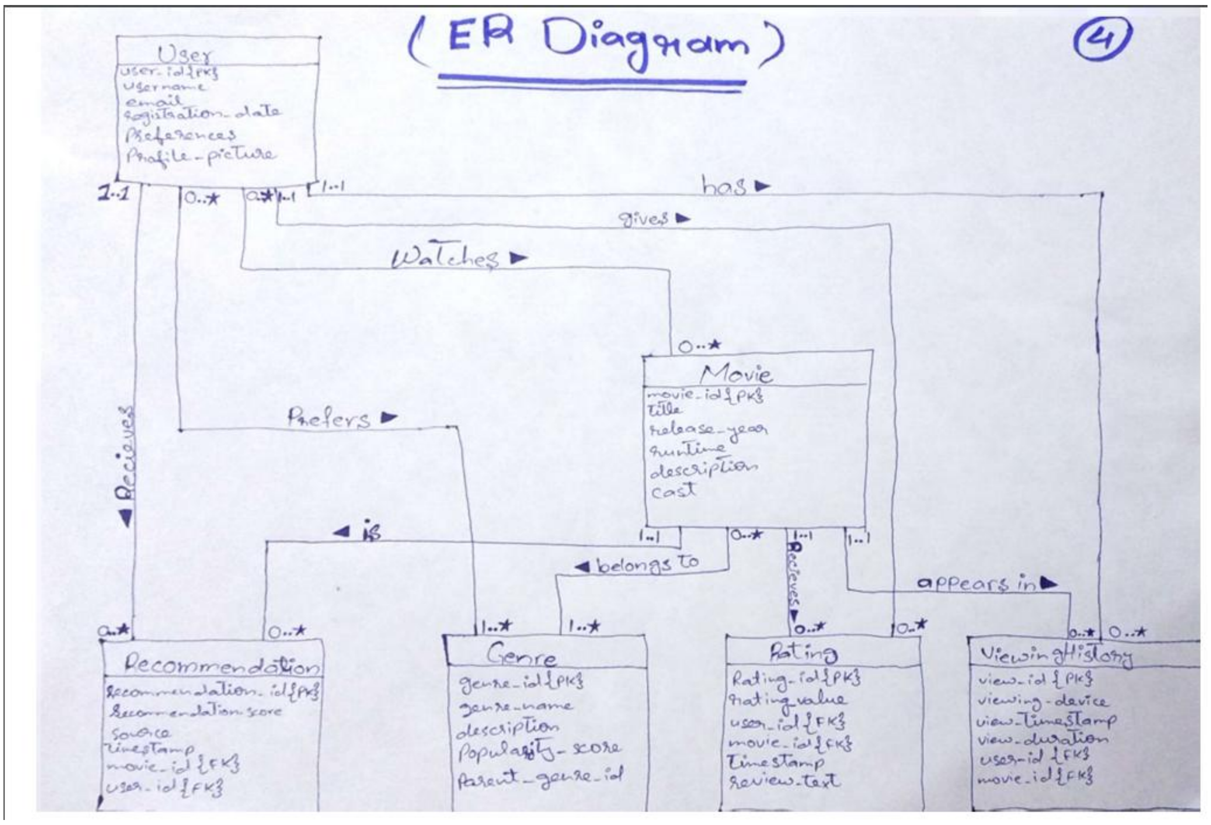
1. **User:** This stores user-related data like user-id, email, preferences, username, and profile picture.
2. **Movie:** This stores movie details, including movie-id, title, release year, description, and cast.
3. **Rating:** This keeps track of ratings, linking users and movies. It includes rating-id, user-id, movie-id, rating value, and timestamp.
4. **Genres:** This tracks genre and includes genre-id, genre name, description, and popularity score.
5. **Viewing History:** This table stores information on what movies users have watched, when, on what device, and for how long.
6. **Recommendations:** This table stores the movies suggested to each user, along with the recommendation score and timestamp.

Relationships:

- **User and Ratings:** One user can give ratings for many movies, and each rating is linked to a single user.
- **Movie and Ratings:** Many users can rate one movie, and each rating is tied to a single movie.
- **User and Viewing History:** A user can have many records in the viewing history, and each record is connected to a single user.
- **Movie and Viewing History:** A movie can be viewed by many users, and each viewing is logged separately.
- **User and Recommendations:** A user can get many movie recommendations.
- **Movie and Recommendations:** A movie can be recommended to multiple users.

- **User and Movie:** Many users can watch the same movie, and each user can watch many movies.
- **Movie and Genre:** Movies can belong to several genres, and genres can include many movies.
- **User and Genre:** A user can have preferences for multiple genres, and genres can be liked by many users.

ER Diagram:



Database Schema:

The schema is designed based on the Movie Recommendation System requirements and includes the following tables:

- **Users:(clients)**
  - **UserID** (Primary Key)
  - Name
  - Email
  - Password
  - JoinDate

- **Movies:**
  - **MovieID** (Primary Key)
  - Title

    - Genre (stores genre IDs as comma-separated values)
    - ReleaseDate
    - Director
    - Rating

- **Ratings:(review list)**

    - **RatingID** (Primary Key)
    - **UserID** (Foreign Key referencing Users.UserID)
    - **MovieID** (Foreign Key referencing Movies.MovieID)
    - Rating
    - Review

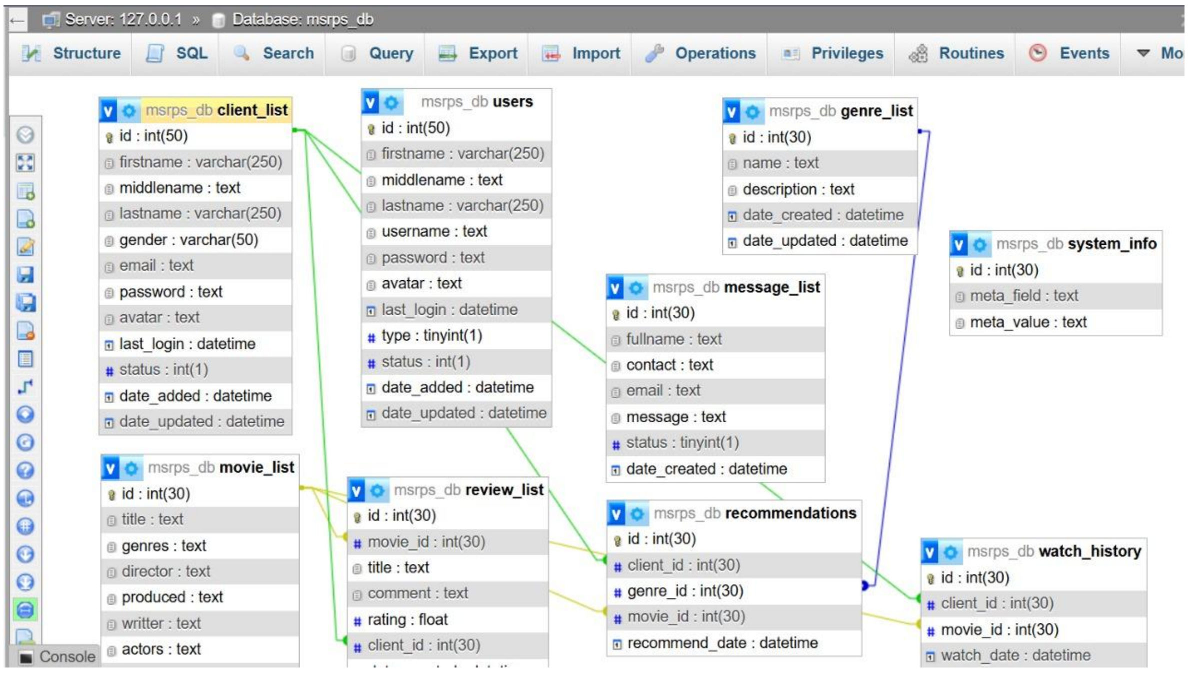
- **Recommendations:**

    - **RecommendationID** (Primary Key)
    - **UserID** (Foreign Key referencing Users.UserID)
    - **MovieID** (Foreign Key referencing Movies.MovieID)
    - RecommendationDate

- **WatchHistory:**

    - **WatchID** (Primary Key)
    - **UserID** (Foreign Key referencing Users.UserID)
    - **MovieID** (Foreign Key referencing Movies.MovieID)
    - WatchDate

- **Relational Schema:**



msrps_db **client_list**
- id : int(50)
- firstname : varchar(250)
- middlename : text
- lastname : varchar(250)
- gender : varchar(50)
- email : text
- password : text
- avatar : text
- last_login : datetime
- status : int(1)
- date_added : datetime
- date_updated : datetime

msrps_db **users**
- id : int(50)
- firstname : varchar(250)
- middlename : text
- lastname : varchar(250)
- username : text
- password : text
- avatar : text
- last_login : datetime
- type : tinyint(1)
- status : int(1)
- date_added : datetime
- date_updated : datetime

msrps_db **genre_list**
- id : int(30)
- name : text
- description : text
- date_created : datetime
- date_updated : datetime

msrps_db **system_info**
- id : int(30)
- meta_field : text
- meta_value : text

msrps_db **message_list**
- id : int(30)
- fullname : text
- contact : text
- email : text
- message : text
- status : tinyint(1)
- date_created : datetime

msrps_db **movie_list**
- id : int(30)
- title : text
- genres : text
- director : text
- produced : text
- writer : text
- actors : text

msrps_db **review_list**
- id : int(30)
- movie_id : int(30)
- title : text
- comment : text
- rating : float
- client_id : int(30)

msrps_db **recommendations**
- id : int(30)
- client_id : int(30)
- genre_id : int(30)
- movie_id : int(30)
- recommend_date : datetime

msrps_db **watch_history**
- id : int(30)
- client_id : int(30)
- movie_id : int(30)
- watch_date : datetime

- **Normalization:**

The database for the Movie Recommendation System is designed and organized in Third Normal Form (3NF) to ensure accuracy and remove duplicate data. Each table has a unique identifier (primary key) and uses foreign keys to connect related data, such as Users, Movies, Ratings, Genres, WatchHistory, and Recommendations. The design follows strict rules to avoid unnecessary dependencies, so each piece of information is directly linked to the table's primary key. This structure saves storage, keeps data accurate, and makes it easier to retrieve information, supporting the system's recommendation features and user-friendly design.

5. Application Design:

Features of the Application:s

- **User Interface (UI):** The application will have an easy-to-use interface where users can manage their profiles, browse movies by genre, rate films, and view recommendations.
- **Recommendation System:** The system will use algorithms to suggest movies based on the user's history, ratings, and preferences.
- **Database Interaction:** The application will pull data from the database to show personalized recommendations, movie details, and ratings.

Design Considerations:

- The interface will be simple and intuitive, so users can easily navigate through the system.
- Recommendations will be updated automatically as the user watches and rates movies.
- Users will be able to share their movie suggestions on social platforms or within the system.

6. DBMS Selection:

Criteria for Choosing the Right DBMS:

- **Scalability:**

The database system should be able to handle an increasing amount of data as the user base grows.

- **Data Integrity:**

The system must ensure that the data stored (such as ratings and user information) is consistent and accurate.

- **Performance:**

The DBMS should provide fast access to data, especially when retrieving personalized recommendations.

- **Compatibility:**

It should work well with the application to ensure smooth integration.

7. Prototyping:

Phase of Prototyping: The prototyping phase involves building an early version of the system to demonstrate its functionality and gather feedback for improvements.

Prototype Features:

- **User Profile Management:** Users can create and manage their profiles, updating preferences and viewing history.
- **Movie Recommendations:** A system that suggests movies based on the user’s previous activity and ratings.
- **Rating System:** Users can rate movies they’ve watched and see other users' ratings.
- **Genre Browsing:** Users can explore movies by their preferred genres.

- **Social Sharing:** Users can share their favourite movie recommendations with friends.

Prototype Goals:

- **Test the User Interface:** Ensure the interface is easy to use and navigable.
- **Validate Recommendations:** Make sure the recommendations are relevant to users based on their viewing history and ratings.
- **Gather Feedback:** Collect user feedback on the prototype to refine the system before fully developing it.

8. Implementation:

- **Frontend Development:**

The user interface was built using **HTML**, **CSS**, and **JavaScript**, with **Bootstrap** for responsive and visually appealing design.

- **Dynamic Web Application:**

The frontend was further enhanced with **React.js**, providing a seamless and interactive user experience.

- **Backend Development:**

The server-side logic was implemented using **PHP**, ensuring smooth communication between the database and the frontend.

- **Database Integration:**

The system utilized the database structure defined in the provided SQL file for storing and managing user data, movie details, and recommendations.

- **Cross-Platform Compatibility:**

The combination of these technologies ensured the system worked efficiently across different devices and platforms.

9. Data Conversion and Loading:

Data Population:

Sample data was inserted into the tables to ensure functionality and demonstrate query capabilities. Below are examples:

Users Table:

UserID	Name	Email	Password	JoinDate
1	Abdul waseh	abdul@gmail.com	********	2023-01-15
2	Saad Imran	saad@gmail.com	********	2023-02-20
3	Zohaib fayyaz	zohaib@gmail.com	********	2023-03-10

Movies Table:

MovieID	Title	Genre	ReleaseDate	Director	Rating
1	Inception	11,13	2010-07-16	Christopher Nolan	8.8
2	The Dark Knight	2	2008-07-18	Christopher Nolan	9.0
3	Interstellar	11	2014-11-07	Christopher Nolan	8.6

Ratings Table:

RatingID	UserID	MovieID	Rating	Review
1	1	1	5	Mind-blowing!
2	2	2	4.5	Outstanding action
3	3	3	4	Great visuals

RatingID	UserID	MovieID	Rating	Review

Recommendations Table:

RecommendationID	UserID	MovieID	RecommendationDate
1	1	2	2024-01-05
2	2	1	2024-01-06
3	3	3	2024-01-07

WatchHistory Table:

WatchID	UserID	MovieID	WatchDate
1	1	1	2024-01-01
2	2	2	2024-01-02
3	3	3	2024-01-03

10. Testing:

- **Unit Testing:** Checked if each part of the system, like the database and recommendation engine, worked correctly on its own.
- **Integration Testing:** Made sure all parts of the system, like the database, backend, and frontend, worked well together.
- **Performance Testing:** Tested how well the system handled many users at the same time to ensure it stayed fast and stable.
- **User Testing:** Asked a group of people to try the system and give feedback to make it easier and better to use.

11. Operational Maintenance:

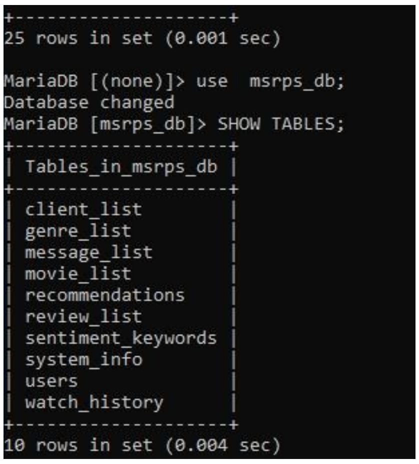
- New movies, ratings, and corrections are added to the database regularly to keep it up-to-date.
- The system is checked often to find and fix any issues or bugs quickly.
- The recommendation methods are improved over time to make suggestions more accurate and match user preferences.
- User feedback is collected regularly to improve features and make the system better for everyone.

➢ Queries:

1. SQL Queries:

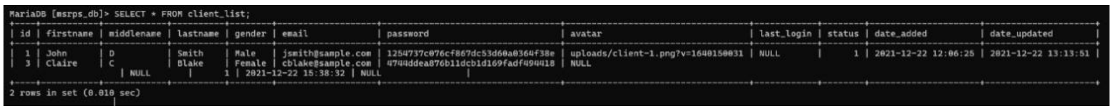
DATABASE msrps_db.

Tables:



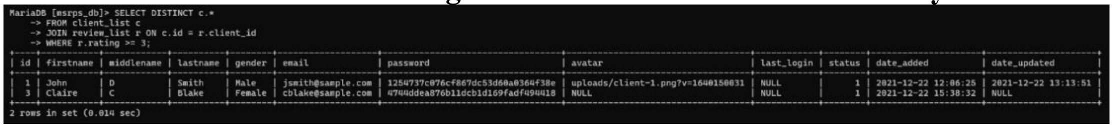
```
+--------------------+
25 rows in set (0.001 sec)

MariaDB [(none)]> use  msrps_db;
Database changed
MariaDB [msrps_db]> SHOW TABLES;
+--------------------+
| Tables_in_msrps_db |
+--------------------+
| client_list        |
| genre_list         |
| message_list       |
| movie_list         |
| recommendations    |
| review_list        |
| sentiment_keywords |
| system_info        |
| users              |
| watch_history      |
+--------------------+
10 rows in set (0.004 sec)
```

- **SELECT * FROM client_list;**



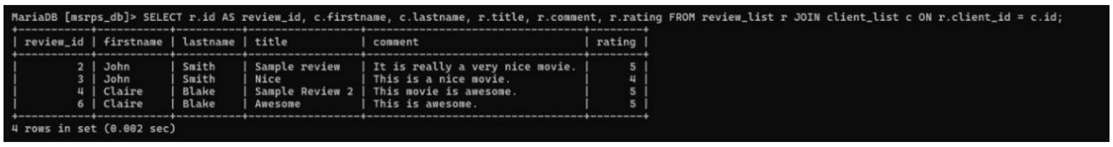
```
MariaDB [msrps_db]> SELECT * FROM client_list;
+----+-----------+------------+----------+--------+---------------------+----------------------------------+---------------------------------+------------+--------+---------------------+---------------------+
| id | firstname | middlename | lastname | gender | email               | password                         | avatar                          | last_login | status | date_added          | date_updated        |
+----+-----------+------------+----------+--------+---------------------+----------------------------------+---------------------------------+------------+--------+---------------------+---------------------+
|  1 | John      | D          | Smith    | Male   | jsmith@sample.com   | 1254737c076cf867dc53d60a0364f38e | uploads/client-1.png?v=1640150031 | NULL     |      1 | 2021-12-22 12:06:25 | 2021-12-22 13:13:51 |
|  3 | Claire    | C          | Blake    | Female | cblake@sample.com   | 4744ddea876b11dcb1d169fadf494418 | NULL
                | NULL       |        1 | 2021-12-22 15:38:32 | NULL                |
+----+-----------+------------+----------+--------+---------------------+----------------------------------+---------------------------------+------------+--------+---------------------+---------------------+
2 rows in set (0.010 sec)
```

- **Find all clients who have given reviews but have not rated any movie below 3:**



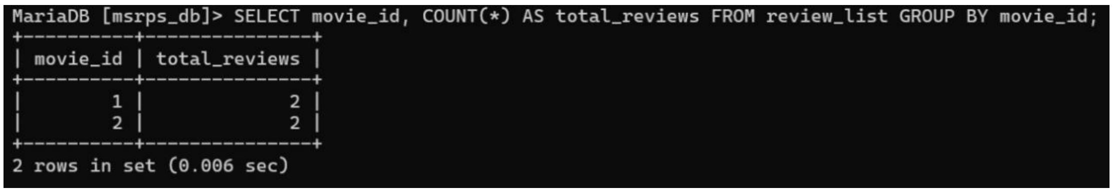
```
MariaDB [msrps_db]> SELECT DISTINCT c.*
    -> FROM client_list c
    -> JOIN review_list r ON c.id = r.client_id
    -> WHERE r.rating >= 3;
+----+-----------+------------+----------+--------+-------------------+----------------------------------+-----------------------------------+------------+--------+---------------------+---------------------+
| id | firstname | middlename | lastname | gender | email             | password                         | avatar                            | last_login | status | date_added          | date_updated        |
+----+-----------+------------+----------+--------+-------------------+----------------------------------+-----------------------------------+------------+--------+---------------------+---------------------+
|  1 | John      | D          | Smith    | Male   | jsmith@sample.com | 1254737c076cf867dc53d60a0364f38e | uploads/client-1.png?v=1640150031 | NULL       |      1 | 2021-12-22 12:06:25 | 2021-12-22 13:13:51 |
|  3 | Claire    | C          | Blake    | Female | cblake@sample.com | 4744ddea876b11dcb1d169fadf494418 | NULL                              | NULL       |      1 | 2021-12-22 15:38:32 | NULL                |
+----+-----------+------------+----------+--------+-------------------+----------------------------------+-----------------------------------+------------+--------+---------------------+---------------------+
2 rows in set (0.014 sec)
```

- **SELECT r.id AS review_id, c.firstname, c.lastname, r.title, r.comment, r.rating FROM review_list r JOIN client_list c ON r.client_id = c.id;**



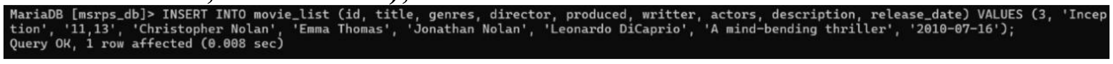
```
MariaDB [msrps_db]> SELECT r.id AS review_id, c.firstname, c.lastname, r.title, r.comment, r.rating FROM review_list r JOIN client_list c ON r.client_id = c.id;
+-----------+-----------+----------+------------------+-----------------------------------+--------+
| review_id | firstname | lastname | title            | comment                           | rating |
+-----------+-----------+----------+------------------+-----------------------------------+--------+
|         2 | John      | Smith    | Sample review    | It is really a very nice movie.   |      5 |
|         3 | John      | Smith    | Nice             | This is a nice movie.             |      4 |
|         4 | Claire    | Blake    | Sample Review 2  | This movie is awesome.            |      5 |
|         6 | Claire    | Blake    | Awesome          | This is awesome.                  |      5 |
+-----------+-----------+----------+------------------+-----------------------------------+--------+
4 rows in set (0.002 sec)
```

- **SELECT movie_id, COUNT(*) AS total_reviews FROM review_list GROUP BY movie_id;**



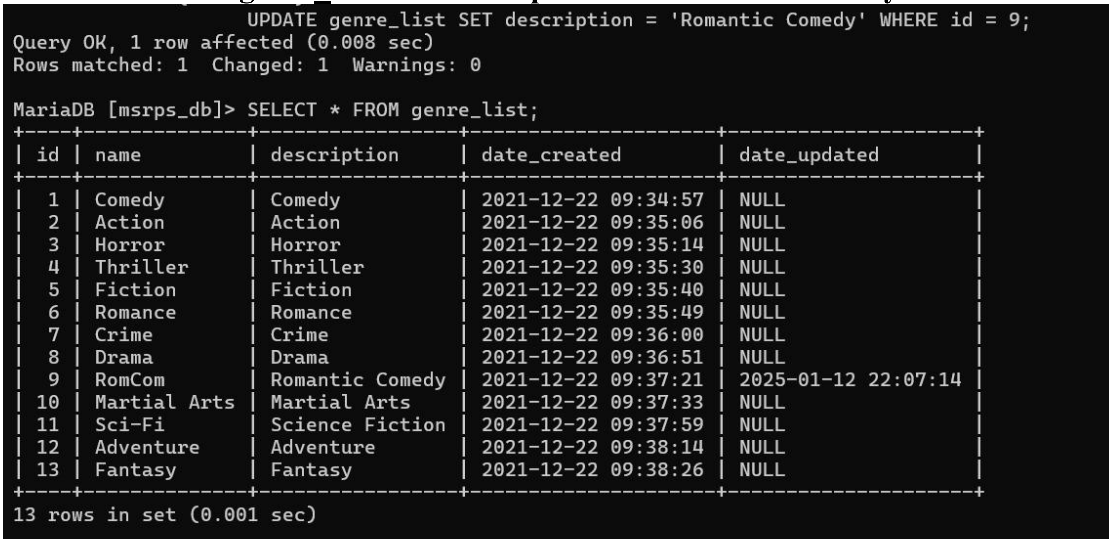
```
MariaDB [msrps_db]> SELECT movie_id, COUNT(*) AS total_reviews FROM review_list GROUP BY movie_id;
+----------+---------------+
| movie_id | total_reviews |
+----------+---------------+
|        1 |             2 |
|        2 |             2 |
+----------+---------------+
2 rows in set (0.006 sec)
```

- **INSERT INTO movie_list (id, title, genres, director, produced, writter, actors, description, release_date) VALUES (3, 'Inception', '11,13', 'Christopher Nolan', 'Emma Thomas', 'Jonathan Nolan', 'Leonardo DiCaprio', 'A mind-bending thriller', '2010-07-16');**



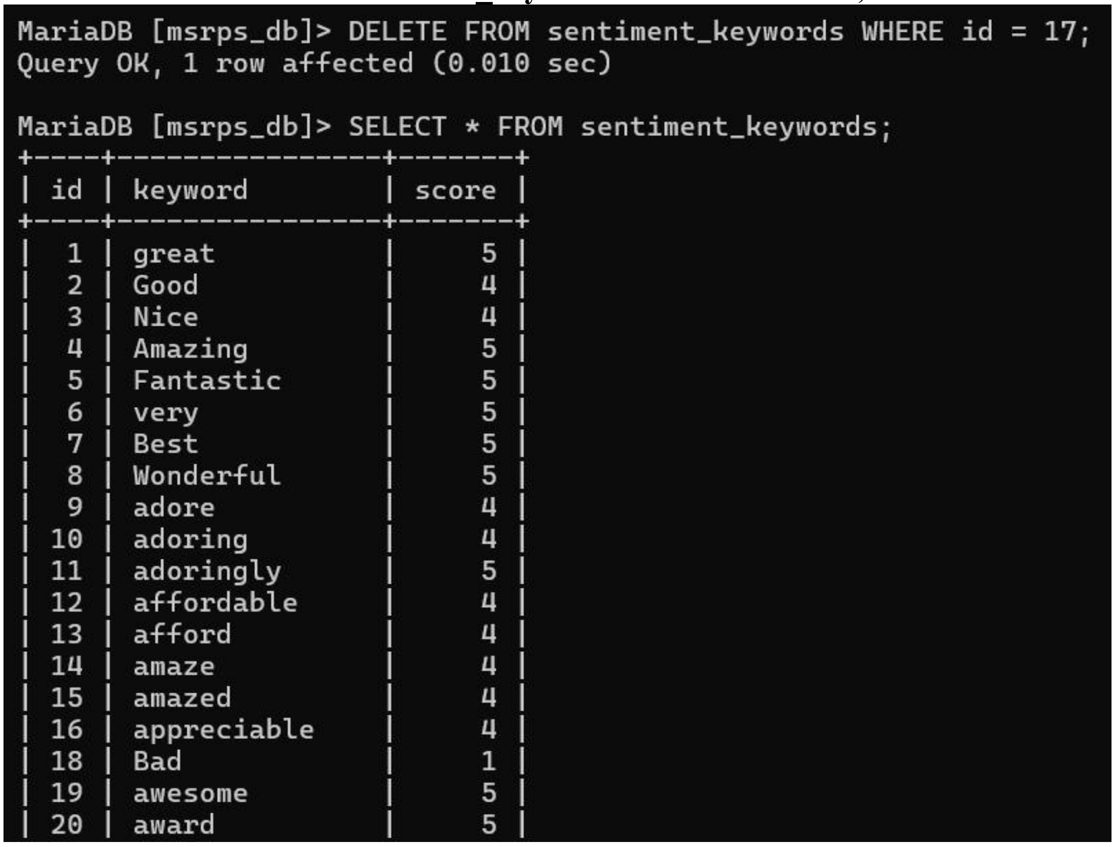
```
MariaDB [msrps_db]> INSERT INTO movie_list (id, title, genres, director, produced, writter, actors, description, release_date) VALUES (3, 'Incep
tion', '11,13', 'Christopher Nolan', 'Emma Thomas', 'Jonathan Nolan', 'Leonardo DiCaprio', 'A mind-bending thriller', '2010-07-16');
Query OK, 1 row affected (0.008 sec)
```

- **UPDATE genre_list SET description = 'Romantic Comedy' WHERE id = 9;**



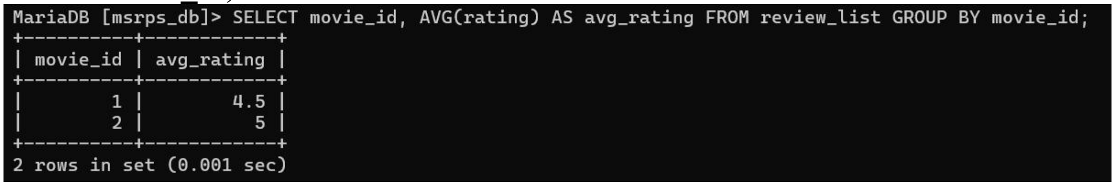
```
                    UPDATE genre_list SET description = 'Romantic Comedy' WHERE id = 9;
Query OK, 1 row affected (0.008 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [msrps_db]> SELECT * FROM genre_list;
+----+--------------+-----------------+---------------------+---------------------+
| id | name         | description     | date_created        | date_updated        |
+----+--------------+-----------------+---------------------+---------------------+
|  1 | Comedy       | Comedy          | 2021-12-22 09:34:57 | NULL                |
|  2 | Action       | Action          | 2021-12-22 09:35:06 | NULL                |
|  3 | Horror       | Horror          | 2021-12-22 09:35:14 | NULL                |
|  4 | Thriller     | Thriller        | 2021-12-22 09:35:30 | NULL                |
|  5 | Fiction      | Fiction         | 2021-12-22 09:35:40 | NULL                |
|  6 | Romance      | Romance         | 2021-12-22 09:35:49 | NULL                |
|  7 | Crime        | Crime           | 2021-12-22 09:36:00 | NULL                |
|  8 | Drama        | Drama           | 2021-12-22 09:36:51 | NULL                |
|  9 | RomCom       | Romantic Comedy | 2021-12-22 09:37:21 | 2025-01-12 22:07:14 |
| 10 | Martial Arts | Martial Arts    | 2021-12-22 09:37:33 | NULL                |
| 11 | Sci-Fi       | Science Fiction | 2021-12-22 09:37:59 | NULL                |
| 12 | Adventure    | Adventure       | 2021-12-22 09:38:14 | NULL                |
| 13 | Fantasy      | Fantasy         | 2021-12-22 09:38:26 | NULL                |
+----+--------------+-----------------+---------------------+---------------------+
13 rows in set (0.001 sec)
```

- **DELETE FROM sentiment_keywords WHERE id = 17;**



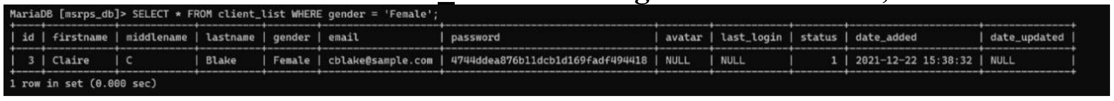
```
MariaDB [msrps_db]> DELETE FROM sentiment_keywords WHERE id = 17;
Query OK, 1 row affected (0.010 sec)

MariaDB [msrps_db]> SELECT * FROM sentiment_keywords;
+----+-----------------+--------+
| id | keyword         | score  |
+----+-----------------+--------+
|  1 | great           |      5 |
|  2 | Good            |      4 |
|  3 | Nice            |      4 |
|  4 | Amazing         |      5 |
|  5 | Fantastic       |      5 |
|  6 | very            |      5 |
|  7 | Best            |      5 |
|  8 | Wonderful       |      5 |
|  9 | adore           |      4 |
| 10 | adoring         |      4 |
| 11 | adoringly       |      5 |
| 12 | affordable      |      4 |
| 13 | afford          |      4 |
| 14 | amaze           |      4 |
| 15 | amazed          |      4 |
| 16 | appreciable     |      4 |
| 18 | Bad             |      1 |
| 19 | awesome         |      5 |
| 20 | award           |      5 |
```

- **SELECT movie_id, AVG(rating) AS avg_rating FROM review_list GROUP BY movie_id;**



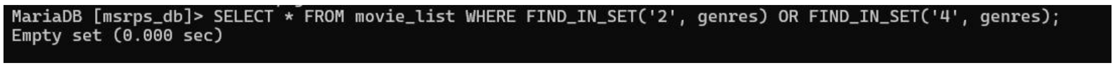
```
MariaDB [msrps_db]> SELECT movie_id, AVG(rating) AS avg_rating FROM review_list GROUP BY movie_id;
+----------+------------+
| movie_id | avg_rating |
+----------+------------+
|        1 |        4.5 |
|        2 |          5 |
+----------+------------+
2 rows in set (0.001 sec)
```

- **SELECT * FROM client_list WHERE gender = 'Female';**



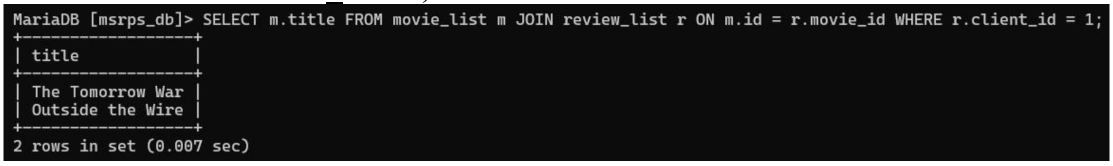
```
MariaDB [msrps_db]> SELECT * FROM client_list WHERE gender = 'Female';
+----+-----------+------------+----------+--------+--------------------+----------------------------------+--------+------------+--------+---------------------+--------------+
| id | firstname | middlename | lastname | gender | email              | password                         | avatar | last_login | status | date_added          | date_updated |
+----+-----------+------------+----------+--------+--------------------+----------------------------------+--------+------------+--------+---------------------+--------------+
|  3 | Claire    | C          | Blake    | Female | cblake@sample.com  | 4744ddea876b11dcb1d169fadf494418 | NULL   | NULL       |      1 | 2021-12-22 15:38:32 | NULL         |
+----+-----------+------------+----------+--------+--------------------+----------------------------------+--------+------------+--------+---------------------+--------------+
1 row in set (0.000 sec)
```

- **SELECT * FROM movie_list WHERE FIND_IN_SET('2', genres) OR FIND_IN_SET('4', genres);**



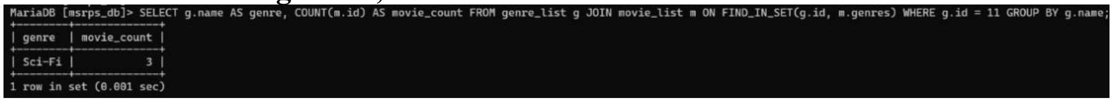
```
MariaDB [msrps_db]> SELECT * FROM movie_list WHERE FIND_IN_SET('2', genres) OR FIND_IN_SET('4', genres);
Empty set (0.000 sec)
```

- **SELECT m.title FROM movie_list m JOIN review_list r ON m.id = r.movie_id WHERE r.client_id = 1;**



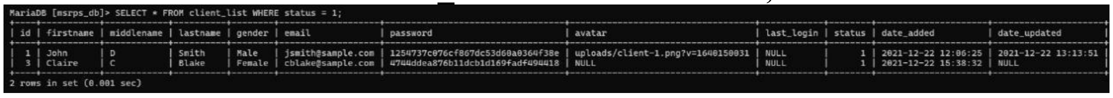
```
MariaDB [msrps_db]> SELECT m.title FROM movie_list m JOIN review_list r ON m.id = r.movie_id WHERE r.client_id = 1;
+------------------+
| title            |
+------------------+
| The Tomorrow War |
| Outside the Wire |
+------------------+
2 rows in set (0.007 sec)
```

- **SELECT g.name AS genre, COUNT(m.id) AS movie_count FROM genre_list g JOIN movie_list m ON FIND_IN_SET(g.id, m.genres) WHERE g.id = 11 GROUP BY g.name;**



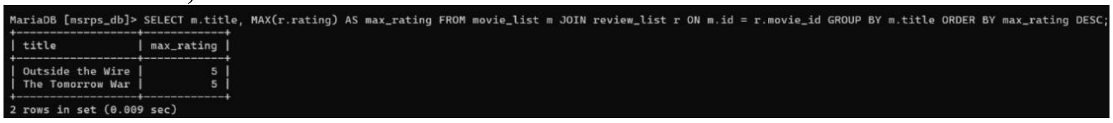
```
MariaDB [msrps_db]> SELECT g.name AS genre, COUNT(m.id) AS movie_count FROM genre_list g JOIN movie_list m ON FIND_IN_SET(g.id, m.genres) WHERE g.id = 11 GROUP BY g.name;
+--------+-------------+
| genre  | movie_count |
+--------+-------------+
| Sci-Fi |           3 |
+--------+-------------+
1 row in set (0.001 sec)
```

- **SELECT * FROM client_list WHERE status = 1;**



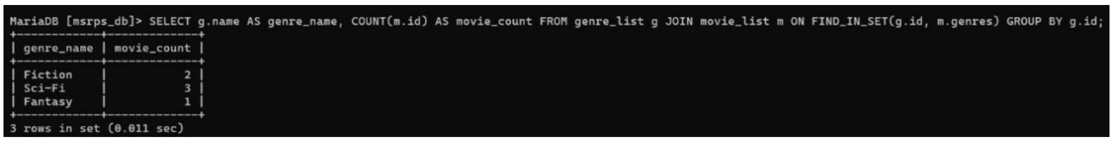
```
MariaDB [msrps_db]> SELECT * FROM client_list WHERE status = 1;
+----+-----------+------------+----------+--------+--------------------+----------------------------------+--------------------------------+------------+--------+---------------------+---------------------+
| id | firstname | middlename | lastname | gender | email              | password                         | avatar                         | last_login | status | date_added          | date_updated        |
+----+-----------+------------+----------+--------+--------------------+----------------------------------+--------------------------------+------------+--------+---------------------+---------------------+
|  1 | John      | D          | Smith    | Male   | jsmith@sample.com  | 1254737c076cf867dc53d60a0364f38e | uploads/client-1.png?v=1640150031 | NULL    |      1 | 2021-12-22 12:06:25 | 2021-12-22 13:13:51 |
|  3 | Claire    | C          | Blake    | Female | cblake@sample.com  | 4744ddea876b11dcb1d169fadf494418 | NULL                           | NULL       |      1 | 2021-12-22 15:38:32 | NULL                |
+----+-----------+------------+----------+--------+--------------------+----------------------------------+--------------------------------+------------+--------+---------------------+---------------------+
2 rows in set (0.001 sec)
```

- **SELECT m.title, MAX(r.rating) AS max_rating FROM movie_list m JOIN review_list r ON m.id = r.movie_id GROUP BY m.title ORDER BY max_rating DESC;**



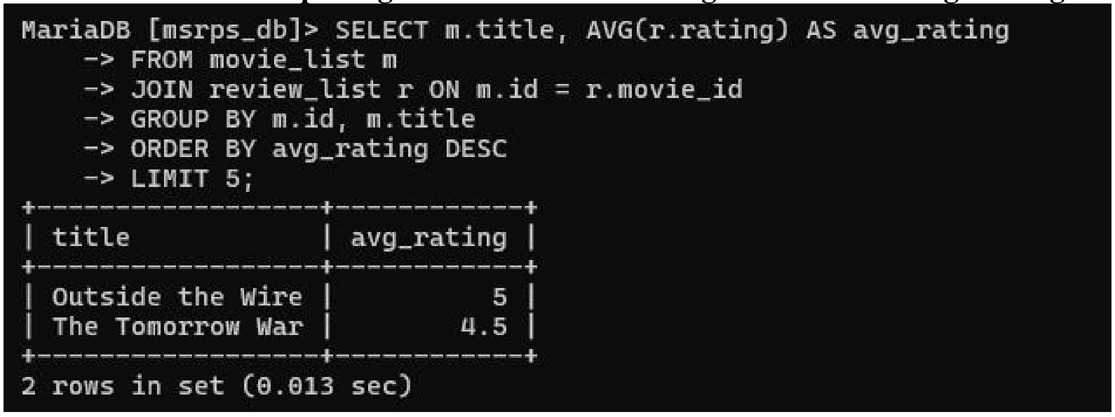
```
MariaDB [msrps_db]> SELECT m.title, MAX(r.rating) AS max_rating FROM movie_list m JOIN review_list r ON m.id = r.movie_id GROUP BY m.title ORDER BY max_rating DESC;
+------------------+------------+
| title            | max_rating |
+------------------+------------+
| Outside the Wire |          5 |
| The Tomorrow War |          5 |
+------------------+------------+
2 rows in set (0.009 sec)
```

- **SELECT g.name AS genre_name, COUNT(m.id) AS movie_count FROM genre_list g JOIN movie_list m ON FIND_IN_SET (g.id, m.genres) GROUP BY g.id;**



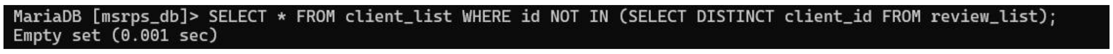
```
MariaDB [msrps_db]> SELECT g.name AS genre_name, COUNT(m.id) AS movie_count FROM genre_list g JOIN movie_list m ON FIND_IN_SET(g.id, m.genres) GROUP BY g.id;
+------------+-------------+
| genre_name | movie_count |
+------------+-------------+
| Fiction    |           2 |
| Sci-Fi     |           3 |
| Fantasy    |           1 |
+------------+-------------+
3 rows in set (0.011 sec)
```

- **Retrieve the top 5 highest-rated movies along with their average rating:**



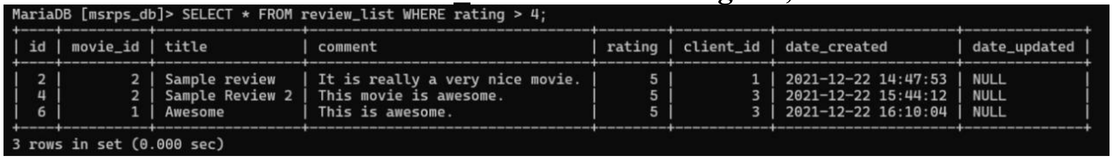
```
MariaDB [msrps_db]> SELECT m.title, AVG(r.rating) AS avg_rating
    -> FROM movie_list m
    -> JOIN review_list r ON m.id = r.movie_id
    -> GROUP BY m.id, m.title
    -> ORDER BY avg_rating DESC
    -> LIMIT 5;
+------------------+------------+
| title            | avg_rating |
+------------------+------------+
| Outside the Wire |          5 |
| The Tomorrow War |        4.5 |
+------------------+------------+
2 rows in set (0.013 sec)
```

- **SELECT * FROM client_list WHERE id NOT IN (SELECT DISTINCT client_id FROM review_list);**



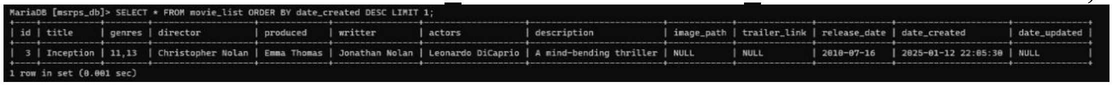
```
MariaDB [msrps_db]> SELECT * FROM client_list WHERE id NOT IN (SELECT DISTINCT client_id FROM review_list);
Empty set (0.001 sec)
```

- **SELECT * FROM review_list WHERE rating > 4;**



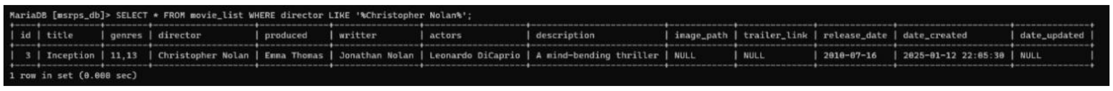
```
MariaDB [msrps_db]> SELECT * FROM review_list WHERE rating > 4;
+----+----------+------------------+------------------------------+--------+-----------+---------------------+--------------+
| id | movie_id | title            | comment                      | rating | client_id | date_created        | date_updated |
+----+----------+------------------+------------------------------+--------+-----------+---------------------+--------------+
|  2 |        2 | Sample review    | It is really a very nice movie. |   5 |         1 | 2021-12-22 14:47:53 | NULL         |
|  4 |        2 | Sample Review 2  | This movie is awesome.       |      5 |         3 | 2021-12-22 15:44:12 | NULL         |
|  6 |        1 | Awesome          | This is awesome.             |      5 |         3 | 2021-12-22 16:10:04 | NULL         |
+----+----------+------------------+------------------------------+--------+-----------+---------------------+--------------+
3 rows in set (0.000 sec)
```

- **SELECT * FROM movie_list ORDER BY date_created DESC LIMIT 1;**



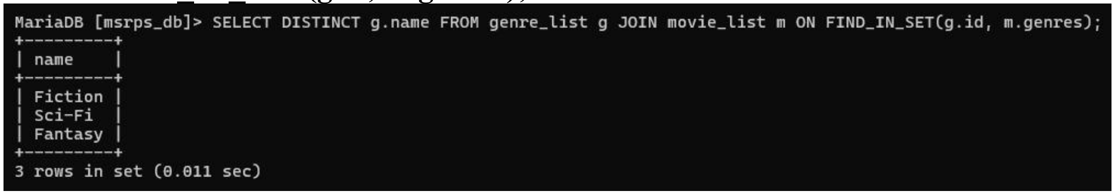
```
MariaDB [msrps_db]> SELECT * FROM movie_list ORDER BY date_created DESC LIMIT 1;
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
| id | title     | genres | director           | produced     | writter        | actors            | description             | image_path | trailer_link | release_date | date_created        | date_updated |
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
|  3 | Inception | 11,13  | Christopher Nolan  | Emma Thomas  | Jonathan Nolan | Leonardo DiCaprio | A mind-bending thriller | NULL       | NULL         | 2010-07-16   | 2025-01-12 22:05:30 | NULL         |
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
1 row in set (0.001 sec)
```

- **SELECT * FROM movie_list WHERE director LIKE '%Christopher Nolan%';**



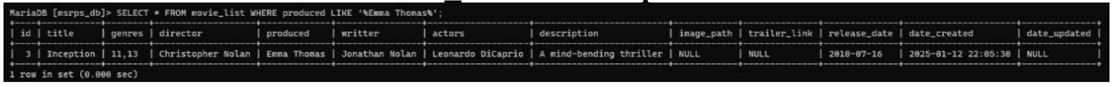
```
MariaDB [msrps_db]> SELECT * FROM movie_list WHERE director LIKE '%Christopher Nolan%';
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
| id | title     | genres | director           | produced     | writter        | actors            | description             | image_path | trailer_link | release_date | date_created        | date_updated |
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
|  3 | Inception | 11,13  | Christopher Nolan  | Emma Thomas  | Jonathan Nolan | Leonardo DiCaprio | A mind-bending thriller | NULL       | NULL         | 2010-07-16   | 2025-01-12 22:05:30 | NULL         |
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
1 row in set (0.000 sec)
```

- **SELECT DISTINCT g.name FROM genre_list g JOIN movie_list m ON FIND_IN_SET(g.id, m.genres);**



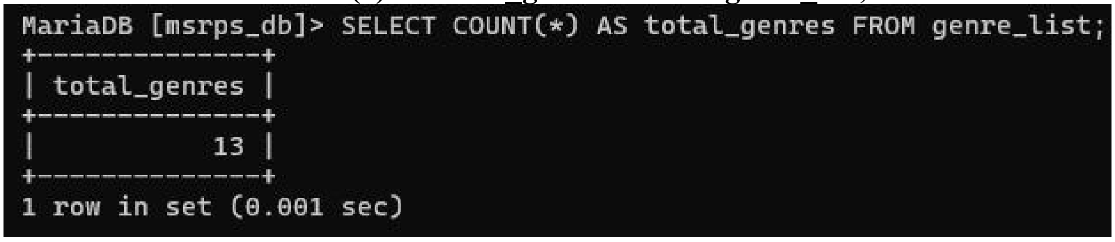
```
MariaDB [msrps_db]> SELECT DISTINCT g.name FROM genre_list g JOIN movie_list m ON FIND_IN_SET(g.id, m.genres);
+---------+
| name    |
+---------+
| Fiction |
| Sci-Fi  |
| Fantasy |
+---------+
3 rows in set (0.011 sec)
```

- **SELECT * FROM movie_list WHERE produced LIKE '%Emma Thomas%';**



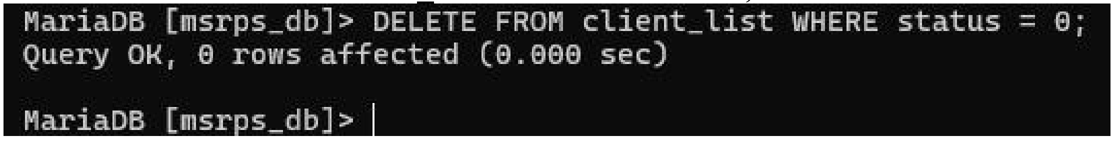
```
MariaDB [msrps_db]> SELECT * FROM movie_list WHERE produced LIKE '%Emma Thomas%';
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
| id | title     | genres | director           | produced     | writter        | actors            | description             | image_path | trailer_link | release_date | date_created        | date_updated |
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
|  3 | Inception | 11,13  | Christopher Nolan  | Emma Thomas  | Jonathan Nolan | Leonardo DiCaprio | A mind-bending thriller | NULL       | NULL         | 2010-07-16   | 2025-01-12 22:05:30 | NULL         |
+----+-----------+--------+--------------------+--------------+----------------+-------------------+-------------------------+------------+--------------+--------------+---------------------+--------------+
1 row in set (0.000 sec)
```

- **SELECT COUNT(*) AS total_genres FROM genre_list;**



```
MariaDB [msrps_db]> SELECT COUNT(*) AS total_genres FROM genre_list;
+--------------+
| total_genres |
+--------------+
|           13 |
+--------------+
1 row in set (0.001 sec)
```

- **DELETE FROM client_list WHERE status = 0;**

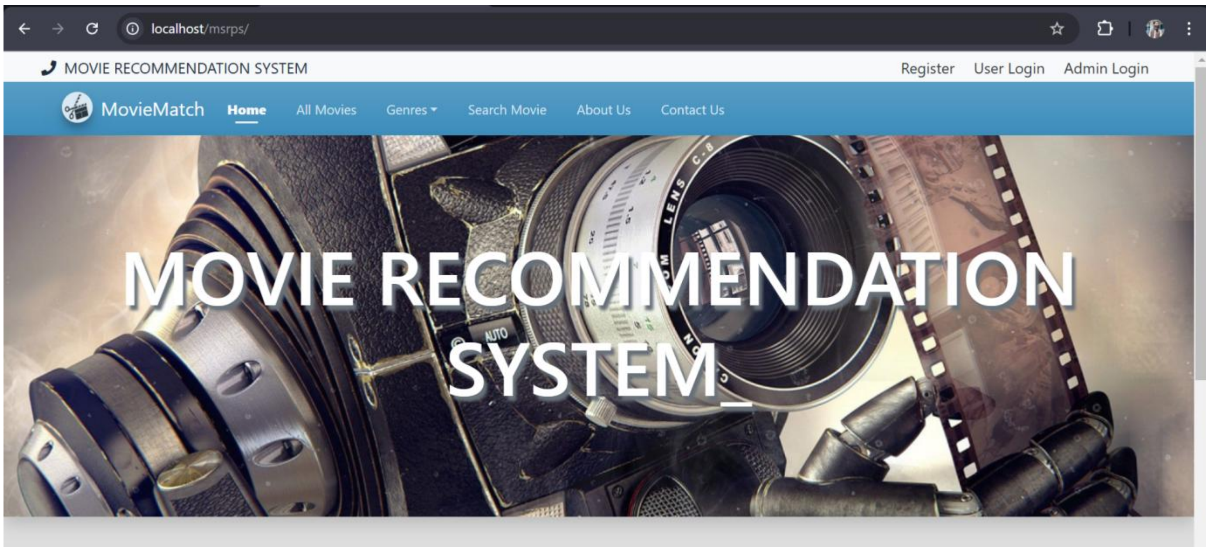


```
MariaDB [msrps_db]> DELETE FROM client_list WHERE status = 0;
Query OK, 0 rows affected (0.000 sec)

MariaDB [msrps_db]> |
```

2. Relational Algebra Queries:

- σ (client_list)
- σ gender = 'Female' (client_list)
- π firstname (client_list)
- client_list ⋈ client_list.id = review_list.client_id (review_list)
- σ rating > 4 (client_list ⋈ client_list.id = review_list.client_id)
- σ release_date > '2020-12-31' (movie_list)
- π title (movie_list)
- σ movie_id = 1 (review_list)
- client_list ∪ client_archive
- genre_list - { 'Action', 'Thriller' }
- genre_list ∩ other_genre_list
- movie_list × review_list
- σ status = 1 (client_list)
- genre_list - π genres (movie_list)
- movie_list ⋈ movie_list.id = review_list.movie_id (review_list)
- γ MAX(rating) (review_list)
- π director (movie_list)
- σ genres = '11' (movie_list)
- γ COUNT(review_id) GROUP BY movie_id (review_list)
- σ status = 0 (client_list)

Movie Recommendation System (DATABASE)



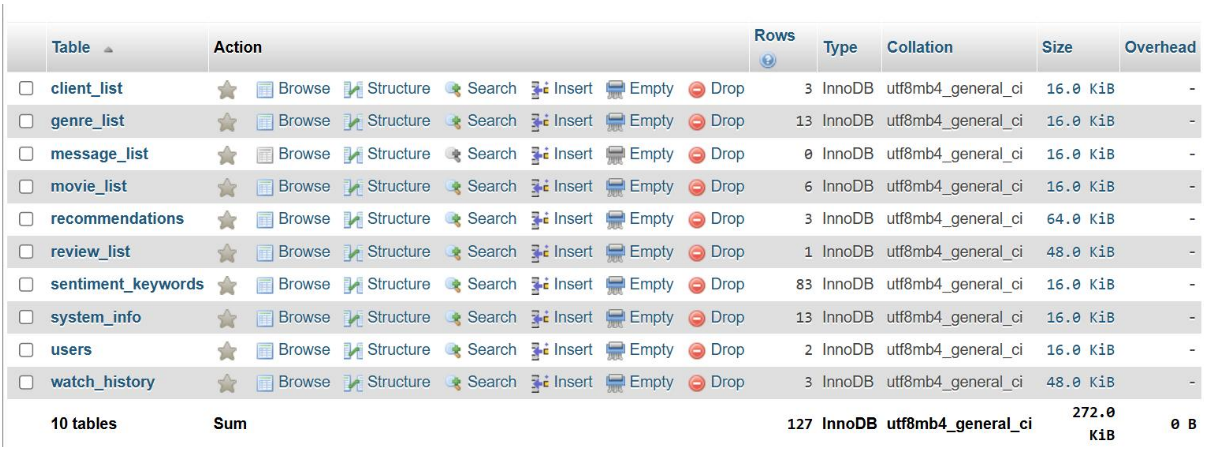
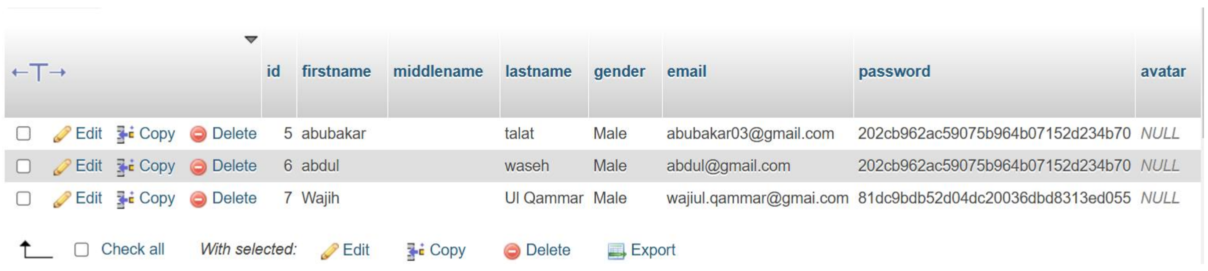


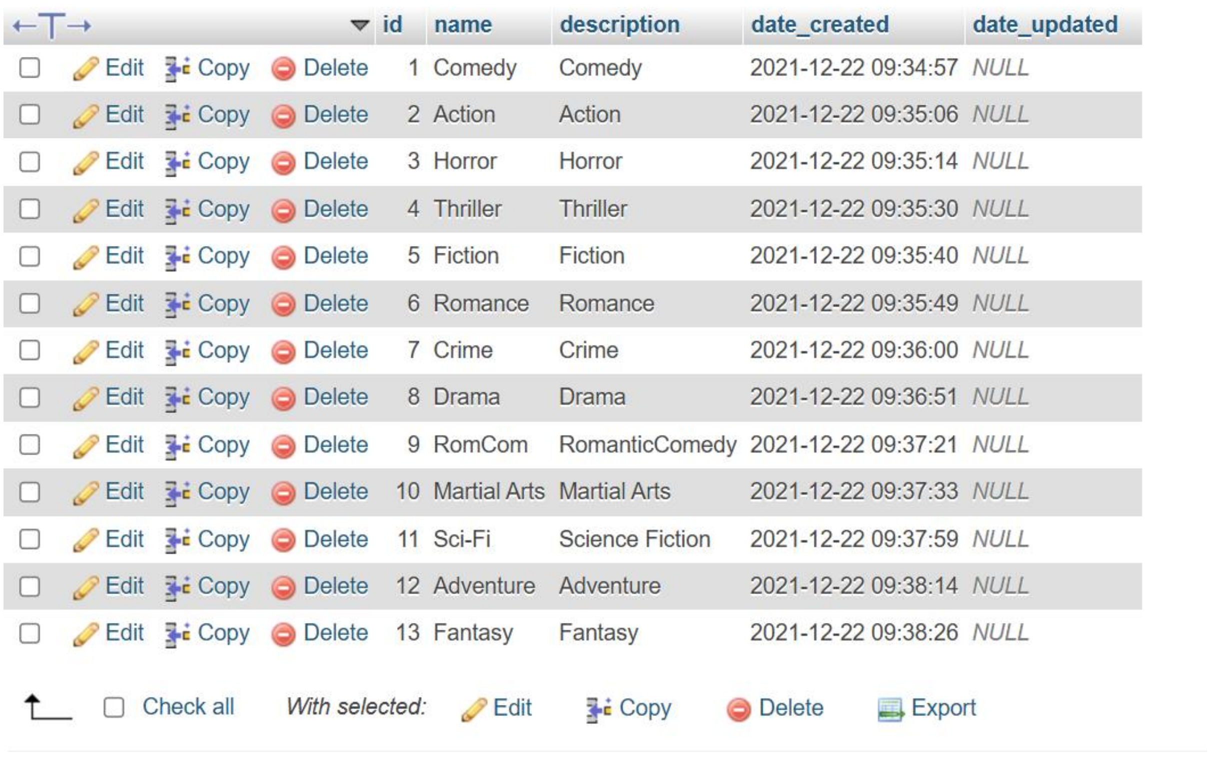
Table	Action	Rows	Type	Collation	Size	Overhead
client_list	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
genre_list	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_general_ci	16.0 KiB	-
message_list	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
movie_list	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
recommendations	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	64.0 KiB	-
review_list	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	48.0 KiB	-
sentiment_keywords	Browse Structure Search Insert Empty Drop	83	InnoDB	utf8mb4_general_ci	16.0 KiB	-
system_info	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
watch_history	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	48.0 KiB	-
10 tables	Sum	127	InnoDB	utf8mb4_general_ci	272.0 KiB	0 B

CLIENT



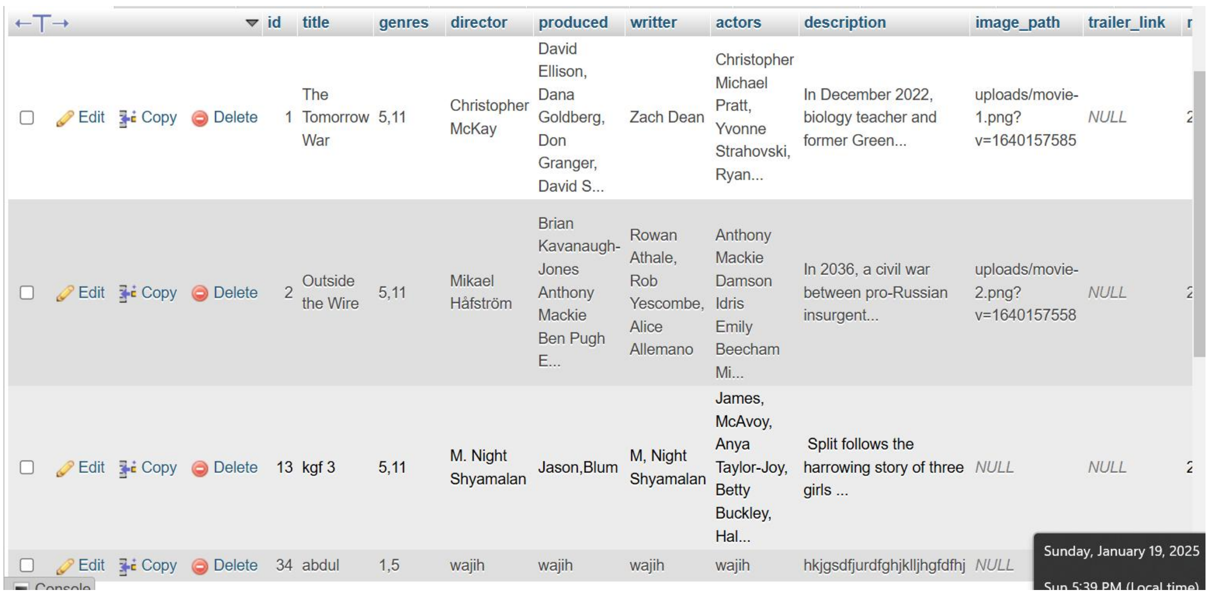
←T→				id	firstname	middlename	lastname	gender	email	password	avatar
☐	Edit	Copy	Delete	5	abubakar		talat	Male	abubakar03@gmail.com	202cb962ac59075b964b07152d234b70	*NULL*
☐	Edit	Copy	Delete	6	abdul		waseh	Male	abdul@gmail.com	202cb962ac59075b964b07152d234b70	*NULL*
☐	Edit	Copy	Delete	7	Wajih		Ul Qammar	Male	wajiul.qammar@gmai.com	81dc9bdb52d04dc20036dbd8313ed055	*NULL*

GENRE



				id	name	description	date_created	date_updated
☐	Edit	Copy	Delete	1	Comedy	Comedy	2021-12-22 09:34:57	NULL
☐	Edit	Copy	Delete	2	Action	Action	2021-12-22 09:35:06	NULL
☐	Edit	Copy	Delete	3	Horror	Horror	2021-12-22 09:35:14	NULL
☐	Edit	Copy	Delete	4	Thriller	Thriller	2021-12-22 09:35:30	NULL
☐	Edit	Copy	Delete	5	Fiction	Fiction	2021-12-22 09:35:40	NULL
☐	Edit	Copy	Delete	6	Romance	Romance	2021-12-22 09:35:49	NULL
☐	Edit	Copy	Delete	7	Crime	Crime	2021-12-22 09:36:00	NULL
☐	Edit	Copy	Delete	8	Drama	Drama	2021-12-22 09:36:51	NULL
☐	Edit	Copy	Delete	9	RomCom	RomanticComedy	2021-12-22 09:37:21	NULL
☐	Edit	Copy	Delete	10	Martial Arts	Martial Arts	2021-12-22 09:37:33	NULL
☐	Edit	Copy	Delete	11	Sci-Fi	Science Fiction	2021-12-22 09:37:59	NULL
☐	Edit	Copy	Delete	12	Adventure	Adventure	2021-12-22 09:38:14	NULL
☐	Edit	Copy	Delete	13	Fantasy	Fantasy	2021-12-22 09:38:26	NULL

MOVIE LIST



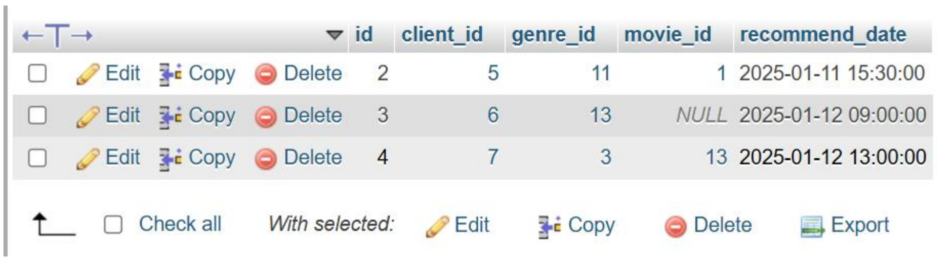
	id	title	genres	director	produced	writter	actors	description	image_path	trailer_link	r
Edit Copy Delete	1	The Tomorrow War	5,11	Christopher McKay	David Ellison, Dana Goldberg, Don Granger, David S...	Zach Dean	Christopher Michael Pratt, Yvonne Strahovski, Ryan...	In December 2022, biology teacher and former Green...	uploads/movie-1.png?v=1640157585	NULL	2
Edit Copy Delete	2	Outside the Wire	5,11	Mikael Håfström	Brian Kavanaugh-Jones Anthony Mackie Ben Pugh E...	Rowan Athale, Rob Yescombe, Alice Allemano	Anthony Mackie Damson Idris Emily Beecham Mi...	In 2036, a civil war between pro-Russian insurgent...	uploads/movie-2.png?v=1640157558	NULL	2
Edit Copy Delete	13	kgf 3	5,11	M. Night Shyamalan	Jason,Blum	M, Night Shyamalan	James, McAvoy, Anya Taylor-Joy, Betty Buckley, Hal...	Split follows the harrowing story of three girls ...	NULL	NULL	2
Edit Copy Delete	34	abdul	1,5	wajih	wajih	wajih	wajih	hkjgsdfjurdfghjklljhgfdfhj	NULL		

Console

Sunday, January 19, 2025

Sun 5:39 PM (Local time)

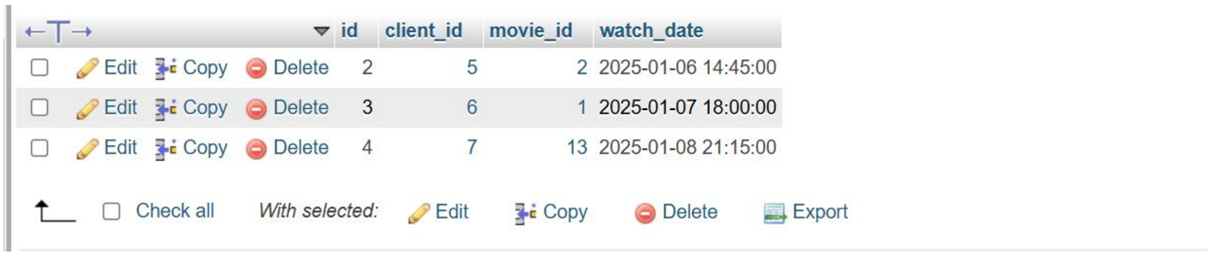
Movie Recommendation



	id	client_id	genre_id	movie_id	recommend_date
Edit Copy Delete	2	5	11	1	2025-01-11 15:30:00
Edit Copy Delete	3	6	13	NULL	2025-01-12 09:00:00
Edit Copy Delete	4	7	3	13	2025-01-12 13:00:00

Check all *With selected:* Edit Copy Delete Export

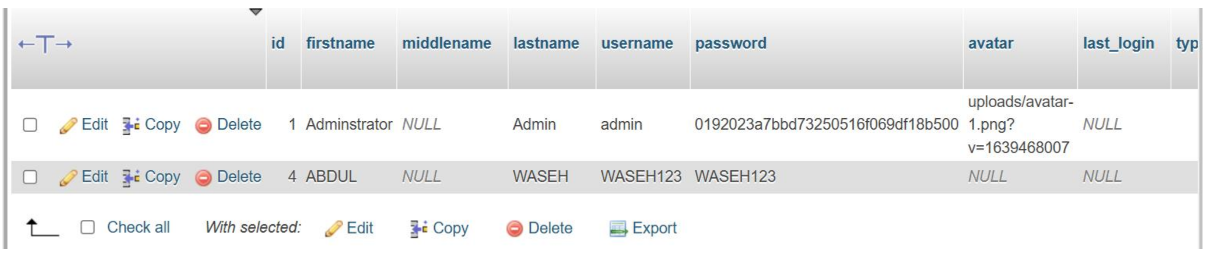
Watch history



	id	client_id	movie_id	watch_date
Edit Copy Delete	2	5	2	2025-01-06 14:45:00
Edit Copy Delete	3	6	1	2025-01-07 18:00:00
Edit Copy Delete	4	7	13	2025-01-08 21:15:00

Check all *With selected:* Edit Copy Delete Export

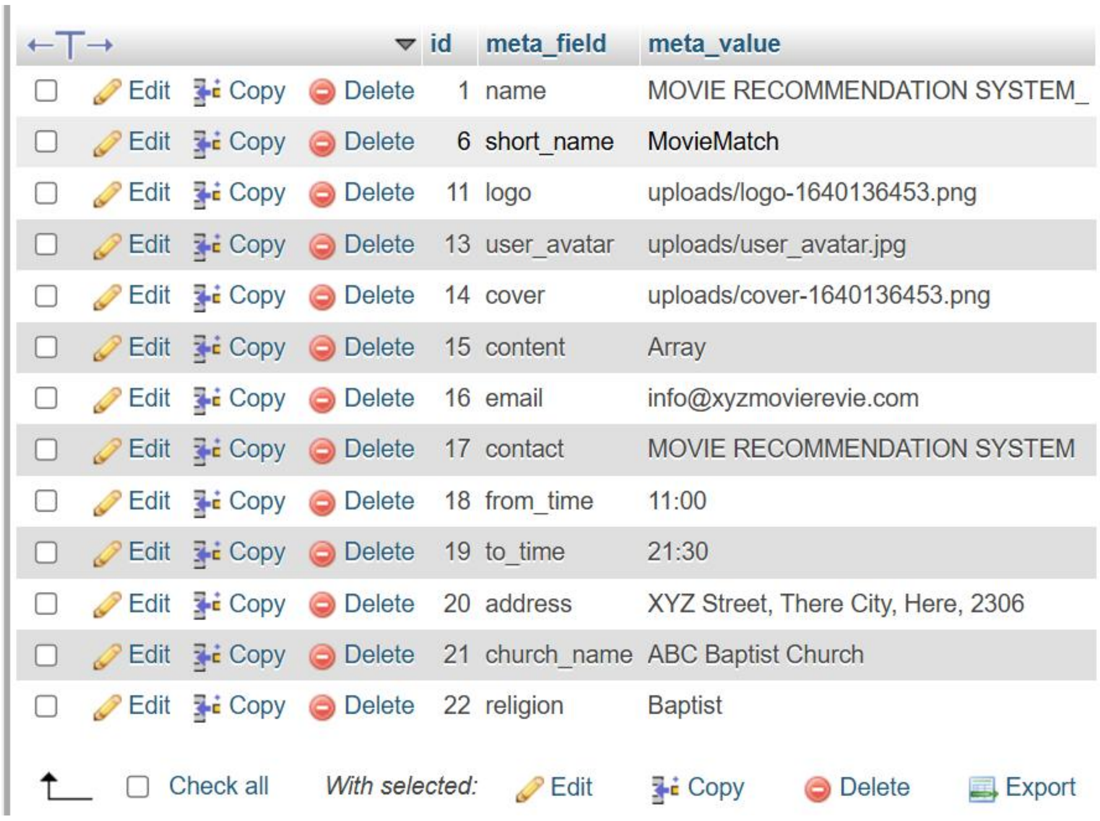
USERS



←T→	id	firstname	middlename	lastname	username	password	avatar	last_login	typ
☐ Edit Copy Delete	1	Adminstrator	*NULL*	Admin	admin	0192023a7bbd73250516f069df18b500	uploads/avatar-1.png?v=1639468007	*NULL*	
☐ Edit Copy Delete	4	ABDUL	*NULL*	WASEH	WASEH123	WASEH123	*NULL*	*NULL*	

☐ Check all *With selected:* Edit Copy Delete Export

SYSTEM INFO



←T→	id	meta_field	meta_value
☐ Edit Copy Delete	1	name	MOVIE RECOMMENDATION SYSTEM_
☐ Edit Copy Delete	6	short_name	MovieMatch
☐ Edit Copy Delete	11	logo	uploads/logo-1640136453.png
☐ Edit Copy Delete	13	user_avatar	uploads/user_avatar.jpg
☐ Edit Copy Delete	14	cover	uploads/cover-1640136453.png
☐ Edit Copy Delete	15	content	Array
☐ Edit Copy Delete	16	email	info@xyzmovierevie.com
☐ Edit Copy Delete	17	contact	MOVIE RECOMMENDATION SYSTEM
☐ Edit Copy Delete	18	from_time	11:00
☐ Edit Copy Delete	19	to_time	21:30
☐ Edit Copy Delete	20	address	XYZ Street, There City, Here, 2306
☐ Edit Copy Delete	21	church_name	ABC Baptist Church
☐ Edit Copy Delete	22	religion	Baptist

☐ Check all *With selected:* Edit Copy Delete Export