**PROJECT REPORT**

**Group A**

**Online Streaming Organizational Database**

**Professor**

**NAVYA REDDY VELVERTHI**

## Team Members:

## SOWMYA KONDETI

**Online Streaming Organizational Database**

## PROJECT DESCRIPTION:

It is technically difficult to maintain all records on paper. Sometimes it causes issues like errors, time consuming, missing crucial information etc. Recording and maintenance of the data on paper is highly impossible.

The project aims to design and develop an organizational database for an online streaming service. The database will be designed to store and manage various types of data related to the streaming service, such as user information, content information, payment information, and subscription information. The database will be designed to ensure data accuracy, consistency, and security.

The user information will include user profiles, viewing history, and payment history. The content information will include the title, genre, language, and ratings of the content. The payment information will include payment methods, payment status, and payment amounts. The subscription information will include the subscription plan, subscription start date, and subscription end date.

The database will be designed to support various types of queries and reports to enable the organization to gain insights into user behavior, content popularity, payment behavior, and subscription trends. The database will also support data analytics and machine learning algorithms to improve user recommendations, content distribution, and subscription plans.

The project will involve the following phases:

Requirements gathering: Gather the requirements for the database from stakeholders, such as the organization's management, users, and technical staff.

Data modeling: Develop the data model for the database, including the entity-relationship diagram, data dictionary, and normalization.

Database design: Design the database schema, including the tables, columns, data types, primary keys, foreign keys, indexes, and constraints.

Database implementation: Implement the database schema using a relational database management system (RDBMS) such as MySQL.

Database population: Populate the database with sample data to test the functionality and performance of the database.

Query and report development: Develop various types of queries and reports to extract insights from the database.

Data analytics: Apply data analytics to improve user recommendations, content distribution, and subscription plans.

Database maintenance: Maintain the database by monitoring performance, optimizing queries, and updating the database schema.

The project will improve the efficiency, accuracy, and security of the organization's data management and enable the organization to make informed business decisions.

## OBJECTIVES:

## The objective of this database system is to provide an online streaming service that enables users to access and view several types of media content, such as movies, TV shows, and documentaries. The database aims to facilitate user interactions with the platform by managing user profiles, content catalogs, playlists, comments, ratings, and viewing histories.

* Details of Users, their payment details and Content information.
* Managing information about the movies, TV shows, and other content available on the platform.
* Updating description, cast of the content accordingly.
* Updating the payment details according to user’s preferred subscription plan.
* Sending an alert when users subscription plan is going to end.Managing the history and playlist of the user and assisting them to keep track of their viewing history and stored content.
* Storing and managing user comments related to the content available on the streaming platform.
* Helps to track and store user ratings for the content available on the platform.
* The record should be updated whenever the transaction is done.
* Up to date information about the content available and user details.

## SCOPE:

## The scope of this database includes storing and managing data related to users, content, and their interactions. Users can create their profiles, manage their subscription plans and payment methods, search, and view content, and interact with other users by rating, commenting, and creating playlists. The content catalog includes details about the available media content, such as titles, descriptions, genres, actors, directors, and ratings.

## The database also supports personalization features, such as recommendations based on user preferences and viewing history. Users can also access their viewing histories and ratings, which can be used to improve the recommendation algorithms.

## Furthermore, the database primary role is to provide a secure and reliable service by managing user authentication and access control, as well as implementing data privacy and security measures. Payment information is also securely stored and managed.

## PROJECT REQUIREMENTS:

**Operating System:** Windows

**Database:** Online Streaming Organizational Database

**Applications:** MYSQL SERVER

## DATABASE REQUIREMENTS:

The following information contains the data tables for the database collection:

1. User Information Table
2. Payment Information Table
3. Content Information Table
4. Users Playlist details Table
5. Users History details Table
6. Rating Table
7. Comments Table

## USER REQUIREMENTS:

* User authentication and account management - Users should be able to create an account, sign in, and manage their personal information, subscription plan, and payment method.
* Content discovery and search - Users should be able to browse and search for content by various categories, such as title, genre, actor/actress, director, and release year.
* Content streaming and playback - Users should be able to stream content in high quality and with smooth playback. They should also be able to adjust the playback settings, such as subtitles and audio language.
* Offline viewing - Users should be able to download content for offline viewing and be able to access the downloaded content within a specified time frame.
* Personalized recommendations - The system should provide personalized recommendations based on the user's viewing history, ratings, and playlists.
* User-generated playlists - Users should be able to create, manage, and share their own playlists with others.
* Rating and review system - Users should be able to rate and review the content they have watched.
* Multiple device access - Users should be able to access the service on multiple devices, such as mobile phones, tablets, laptops, and smart TVs.
* Customer support - The system should provide reliable and responsive customer support via email or phone.
* Data privacy and security - The system should protect user data and privacy by implementing robust security measures such as encryption and user access control.
* Parental controls - The system should provide parental controls to limit access to age-appropriate content and prevent children from accessing adult content.
* Social features - The system should allow users to connect with friends and share content through social features such as watch parties, chat rooms, and social media integrations.
* History tracking - Users should be able to view their viewing history and resume playback from where they left off.
* Multiple payment options - The system should support multiple payment options, such as credit cards, PayPal, and gift cards, to allow users to choose the payment method that works best for them.

## BUSINESS RULES:

1. Each user will have a unique user id.
2. Each content will have a unique content id.
3. Only users with valid accounts and subscriptions can access the content.
4. Each user should provide a unique email id.
5. Each subscription can only belong to one user, but each user can have multiple subscriptions.
6. Each user can have one or many playlists.
7. Each user profile can have one or many history records based on content.
8. Each user can comment and rate one or many times.
9. Each content can have one or more comments.
10. Each content can have one or more ratings.
11. Each content can have one or more records.
12. A Playlist can have one or more content and vice versa.

## Entity Relationship Diagram:

Diagram

Description automatically generated

**Data Dictionary:**

A Data Dictionary is a catalogue – a repository – of the elements in a system. As the name suggests, these elements center on the data and the way they are structured to meet user requirements and organization needs. In a Data Dictionary you will find a list of all the elements that are data flow data stores and processes. The Data Dictionary stores details and descriptions of these elements.

* To manage the detail in the large system.
* To document the features of the system.
* To facilitate analysis of the details to evaluate characteristics and determine where system should be made.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Contents** | **Type** | **Format** | **Range** | **Required** | **PK or FK** | **FK Referenced Table** |
| **User\_Profile** | User\_id | Unique id which is created by the database using sequence | Number | 9999 | - | Y | PK |  |
| Username | Username given by user or email\_id | Varchar (50) | XXXXX | - | Y |  |  |
| Password | Password given by the user | Varchar (30) | XXXXX | - | Y |  |  |
| First\_name | First name of the user | Varchar (20) | XXXXX | - | Y |  |  |
| Last\_name | Last name of the user | Varchar (20) | XXXXX | - | Y |  |  |
| DOB | Date of Birth of user which used to calculate age | Date | DD:MM:YYYY HH:MI:SS | - | Y |  |  |
| Email\_id | Email Address of the user | Varchar (20) | XXXXX | - | Y |  |  |
| **Payment** | Payment\_id | Payment id generated by the database | Number | 9999 | - | Y | PK |  |
| User\_id | Unique id which is created by the database using sequence (User\_profile table) | Number | 9999 | - | Y | FK | User\_Profile |
| Payment\_method | Method of payment done | Varchar (20) | XXXXX | - | Y |  |  |
| Payment\_date | Date of the payment including time | Date | DD:MM:YYYY HH:MI:SS | - | Y |  |  |
| Payment\_status | Status of the payment made | Varchar (20) | XXXXX | - | Y |  |  |
| Payment\_amount | Cost of the plan including taxes | Number | 9999 | - | Y |  |  |
| Subscription\_plan | Type of the plan | Varchar (20) | XXXXX | - | Y |  |  |
| Billing\_address | Address of the user on the card | Varchar (50) | XXXXX | - | Y |  |  |
| **Content** | Content\_id | Unique id which is created by the database using sequence | Number | 9999 | - | Y | PK |  |
| Title | Name of the film/show/etc.. | Varchar (100) | XXXXX | - | Y |  |  |
| Description | Description about the movie/show/etc.. | Varchar (1000) | XXXXX | - | Y |  |  |
| Cast | Name of the actors | Varchar (60) | XXXXX | - | Y |  |  |
| Genre | Name of the Genre | Varchar (100) | XXXXX | - | Y |  |  |
| Language | Language of the film | Varchar (20) | XXXXX | - | Y |  |  |
| Run\_time | Length of the content in mins | Number | 9999 | - | Y |  |  |
| Release\_date | Date on which the show/movie released | Date | DD:MM:YYYY HH:MI:SS | - | Y |  |  |
| **Comments** | Comment\_id | Unique id which is created by the database using sequence | Number | 9999 | - | Y | PK |  |
| Content\_id | Unique id which is created by the database using sequence (Content table) | Number | 9999 | - | Y | FK | Content |
| User\_id | Unique id which is created by the database using sequence (User\_profile) | Number | 9999 | - | Y | FK | User\_Profile |
| Comment | Text entered by the user about the movie/show/etc.. | Varchar (80) | XXXXX | - | Y |  |  |
| **Play\_list** | User\_id | Unique id which is created by the database using sequence (User\_profile table) | Number | 9999 | - | Y | FK | User\_Profile |
| Playlist\_id | Unique id which is created by the database using sequence | Number | 9999 | - | Y | PK |  |
| Playlist\_name | Name of the playlist or profile created by the user | Varchar (20) | XXXXX | - | Y |  |  |
| Created\_name | Name of the user created | Varchar (20) | XXXXX | - | Y |  |  |
| Content\_id | Unique id which is created by the database using sequence (content table) | Number | 9999 | - | Y | FK | Content |
| Playlist\_content | Content in the playlist added by the user | Varchar (100) | XXXXX | - | Y |  |  |
| **History** | History\_id | Unique id which is created by the database using sequence | Number | 9999 | - | Y | PK |  |
| User\_id | Unique id which is created by the database using sequence (User\_profile table) | Number | 9999 | - | Y | FK | User\_Profile |
| Content\_id | Unique id which is created by the database using sequence (contents table) | Number | 9999 | - | Y | FK | Content |
| Date | On the date which the content was seen | Date | DD:MM:YYYY HH:MI:SS | - | Y |  |  |
| Duration | Amount of time the content is watched in mins | Number | 9999 | - | Y |  |  |
| **Rating** | Rating\_id | Unique id which is created by the database using sequence | Number | 9999 | - | Y | PK |  |
| User\_id | Unique id which is created by the database using sequence (user\_profile table) | Number | 9999 | - | Y | FK | User\_Profile |
| Ratings | Rating given by the users from 1 - 5. | Number | 9999 | - | Y |  |  |
| Reviews | Review given the users | Varchar (100) | XXXXX | - | Y |  |  |
| Content\_id | Unique id which is created by the database using sequence (content table) | Number | 9999 | - | Y | FK | Content |

# Creating User profile Table

## First, we have created the database as Online Streaming Organizational Database.

* Then we created table **User\_profile** by using following create statement.

**Query:** CREATE TABLE User\_Profile (

User\_id INT AUTO\_INCREMENT PRIMARY KEY,

Username VARCHAR(50) NOT NULL,

Password VARCHAR(30) NOT NULL,

First\_name VARCHAR(20) NOT NULL,

Last\_name VARCHAR(20) NOT NULL,

DOB DATETIME NOT NULL,

Email\_id VARCHAR(50) NOT NULL

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO user\_profile (Username, Password, First\_name, Last\_name, DOB, Email\_id)

VALUES

('john\_doe', 'pass123', 'John', 'Doe', '1990-01-01 00:00:00', 'john.doe@email.com'),

('jane\_doe', 'pass456', 'Jane', 'Doe', '1992-02-02 00:00:00', 'jane.doe@email.com'),

('bob\_smith', 'pass789', 'Bob', 'Smith', '1995-03-03 00:00:00', 'bob.smith@email.com'),

('alice\_smith', 'pass987', 'Alice', 'Smith', '1998-04-04 00:00:00', 'alice.smith@email.com'),

('tom\_jones', 'pass654', 'Tom', 'Jones', '1985-05-05 00:00:00', 'tom.jones@email.com'),

('sara\_jones', 'pass321', 'Sara', 'Jones', '1988-06-06 00:00:00', 'sara.jones@email.com'),

('peter\_wilson', 'pass246', 'Peter', 'Wilson', '1991-07-07 00:00:00', 'peter.wilson@email.com'),

('amy\_wilson', 'pass135', 'Amy', 'Wilson', '1994-08-08 00:00:00', 'amy.wilson@email.com'),

('jason\_brown', 'pass789', 'Jason', 'Brown', '1986-09-09 00:00:00', 'jason.brown@email.com'),

('emily\_brown', 'pass456', 'Emily', 'Brown', '1989-10-10 00:00:00', 'emily.brown@email.com'),

('david\_kim', 'pass321', 'David', 'Kim', '1992-11-11 00:00:00', 'david.kim@email.com'),

('anna\_kim', 'pass654', 'Anna', 'Kim', '1995-12-12 00:00:00', 'anna.kim@email.com');

INSERT INTO User\_Profile (Username, Password, First\_name, Last\_name, DOB, Email\_id) VALUES

('john123', 'pass123', 'John', 'Doe', '1990-05-15 12:00:00', 'john123@gmail.com'),

('lisa96', 'lisa96pass', 'Lisa', 'Smith', '1996-01-21 08:30:00', 'lisa96@yahoo.com'),

('mike28', 'mike28pass', 'Mike', 'Johnson', '1993-11-02 15:45:00', 'mike.johnson28@gmail.com'),

('amy89', 'amy89pass', 'Amy', 'Nguyen', '1989-07-17 18:20:00', 'amy.nguyen89@yahoo.com'),

('david71', 'david71pass', 'David', 'Lee', '1971-12-03 10:00:00', 'david.lee71@gmail.com'),

('jane87', 'jane87pass', 'Jane', 'Chen', '1987-08-29 17:10:00', 'janechen87@hotmail.com'),

('matt99', 'matt99pass', 'Matt', 'Davis', '1999-03-08 14:30:00', 'matt\_d99@yahoo.com'),

('julia95', 'julia95pass', 'Julia', 'Kim', '1995-06-14 12:45:00', 'juliakim95@gmail.com'),

('ryan93', 'ryan93pass', 'Ryan', 'Garcia', '1993-02-19 09:15:00', 'ryan93@hotmail.com'),

('sarah92', 'sarah92pass', 'Sarah', 'Johnson', '1992-09-23 11:20:00', 'sarah.johnson92@gmail.com'),

('kevin85', 'kevin85pass', 'Kevin', 'Nguyen', '1985-04-30 15:00:00', 'kevin.nguyen85@yahoo.com'),

('emily94', 'emily94pass', 'Emily', 'Lee', '1994-11-08 16:50:00', 'emilylee94@hotmail.com'),

('jason86', 'jason86pass', 'Jason', 'Wu', '1986-10-12 13:20:00', 'jasonwu86@gmail.com'),

('grace91', 'grace91pass', 'Grace', 'Chen', '1991-07-06 20:15:00', 'grace.chen91@yahoo.com'),

('steven88', 'steven88pass', 'Steven', 'Kim', '1988-12-20 07:40:00', 'steven.kim88@hotmail.com'),

('olivia90', 'olivia90pass', 'Olivia', 'Davis', '1990-09-01 16:00:00', 'oliviad90@yahoo.com');

commit;

INSERT INTO User\_Profile (Username, Password, First\_name, Last\_name, DOB, Email\_id) VALUES

('johndoe23', 'pass123', 'John', 'Doe', '1990-07-15 00:00:00', 'johndoe23@email.com'),

('katie96', 'password', 'Katie', 'Smith', '1996-03-24 00:00:00', 'katie96@hotmail.com'),

('jamesbrown', 'brown123', 'James', 'Brown', '1988-11-02 00:00:00', 'jamesbrown@gmail.com'),

('emilybaker', 'baker321', 'Emily', 'Baker', '1992-05-09 00:00:00', 'emilybaker@yahoo.com'),

('jenniferkim', 'jkim123', 'Jennifer', 'Kim', '1985-09-20 00:00:00', 'jenniferkim@gmail.com'),

('michaeljohnson', 'mikej123', 'Michael', 'Johnson', '1995-02-18 00:00:00', 'michaeljohnson@yahoo.com'),

('sarahlee', 'sarahlee12', 'Sarah', 'Lee', '1991-08-28 00:00:00', 'sarahlee@gmail.com'),

('adamgupta', 'adamg123', 'Adam', 'Gupta', '1987-04-06 00:00:00', 'adamgupta@hotmail.com'),

('amywilson', 'wilsona12', 'Amy', 'Wilson', '1994-12-12 00:00:00', 'amywilson@gmail.com'),

('stevenharris', 'harris78', 'Steven', 'Harris', '1983-11-22 00:00:00', 'stevenharris@yahoo.com');

commit;

**Output:**

**A screenshot of a computer

Description automatically generated**

# Creating Content Table

* Then we created content table by using the following create statement.

**Query:** CREATE TABLE Content (

Content\_id INT AUTO\_INCREMENT PRIMARY KEY,

Title VARCHAR(40) NOT NULL,

Description VARCHAR(60) NOT NULL,

Cast VARCHAR(60) NOT NULL,

Genre VARCHAR(20) NOT NULL,

Language VARCHAR(20) NOT NULL,

Run\_time INT NOT NULL,

Release\_date DATE NOT NULL

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO Content (Title, Description, Cast, Genre, Language, Run\_time, Release\_date)

VALUES

('The Godfather', 'The aging patriarch of an organized.', 'Marlon Brando, Al Pacino, James Caan', 'Crime, Drama', 'English', 175, '1972-03-24');

commit;

INSERT INTO Content (Title, Description, Cast, Genre, Language, Run\_time, Release\_date) VALUES

('Inception', 'A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a C.E.O', 'Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen Page', 'Action, Adventure, Sci-Fi', 'English', 148, '2010-07-16'),

('The Social Network', 'Harvard student Mark Zuckerberg creates the social networking site that would become known as Facebook, but is later sued by two brothers who claimed he stole their idea, and the co-founder who was later squeezed out of the business', 'Jesse Eisenberg, Andrew Garfield, Justin Timberlake', 'Biography, Drama', 'English', 120, '2010-10-01'),

('Goodfellas', 'The story of Henry Hill and his life in the mob, covering his relationship with his wife Karen Hill and his mob partners Jimmy Conway and Tommy DeVito in the Italian-American crime syndicate.', 'Robert De Niro, Ray Liotta, Joe Pesci', 'Biography, Crime, Drama', 'English', 146, '1990-09-19'),

('Pulp Fiction', 'The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.', 'John Travolta, Uma Thurman, Samuel L. Jackson', 'Crime, Drama', 'English', 154, '1994-10-14'),

('The Godfather', 'The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.', 'Marlon Brando, Al Pacino, James Caan', 'Crime, Drama', 'English', 175, '1972-03-24'),

('The Dark Knight', 'When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.', 'Christian Bale, Heath Ledger, Aaron Eckhart', 'Action, Crime, Drama', 'English', 152, '2008-07-18'),

('The Shawshank Redemption', 'Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.', 'Tim Robbins, Morgan Freeman, Bob Gunton', 'Drama', 'English', 142, '1994-10-14'),

('Forrest Gump', 'The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate, and other history unfold through the perspective of an Alabama man with an IQ of 75.', 'Tom Hanks, Robin Wright, Gary Sinise', 'Drama, Romance', 'English', 142, '1994-07-06'),

('The Matrix', 'A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.', 'Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss', 'Action, Sci-Fi', 'English', 136, '1999-03-31'),

('Jurassic Park', 'A pragmatic paleontologist visiting an almost complete theme park is tasked with protecting a couple of kids after a power failure causes the park\'s cloned dinosaurs to run loose.', 'Sam Neill, Laura Dern, Jeff Goldblum', 'Action, Adventure, Sci-Fi', 'English', 127, '1993-06-11');

commit;

INSERT INTO Content (Title, Description, Cast, Genre, Language, Run\_time, Release\_date)

VALUES

('The Shawshank Redemption', 'Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.', 'Tim Robbins, Morgan Freeman', 'Drama', 'English', 142, '1994-09-23'),

('The Godfather', 'The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.', 'Marlon Brando, Al Pacino', 'Crime', 'English', 175, '1972-03-24'),

('The Dark Knight', 'When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.', 'Christian Bale, Heath Ledger', 'Action', 'English', 152, '2008-07-18'),

('The Lord of the Rings: The Fellowship of the Ring', 'A meek Hobbit from the Shire and eight companions set out on a journey to destroy the powerful One Ring and save Middle-earth from the Dark Lord Sauron.', 'Elijah Wood, Ian McKellen', 'Fantasy', 'English', 178, '2001-12-19'),

('Forrest Gump', 'The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate and other historical events unfold through the perspective of an Alabama man with an IQ of 75, whose only desire is to be reunited with his childhood sweetheart.', 'Tom Hanks, Robin Wright', 'Drama', 'English', 142, '1994-07-06'),

('Inception', 'A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a C.E.O.', 'Leonardo DiCaprio, Joseph Gordon-Levitt', 'Action', 'English', 148, '2010-07-16'),

('The Matrix', 'A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.', 'Keanu Reeves, Laurence Fishburne', 'Action', 'English', 136, '1999-03-31'),

('Goodfellas', 'The story of Henry Hill and his life in the mob, covering his relationship with his wife Karen Hill and his mob partners Jimmy Conway and Tommy DeVito in the Italian-American crime syndicate.', 'Robert De Niro, Ray Liotta', 'Crime', 'English', 146, '1990-09-19'),

('Pulp Fiction', 'The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.', 'John Travolta, Uma Thurman', 'Crime', 'English', 154, '1994-05-21'),

('The Silence of the Lambs', 'A young F.B.I. cadet must receive the help of an incarcerated and manipulative cannibal killer to help catch another serial killer, a madman who skins his victims.', 'Jodie Foster, Anthony Hopkins', 'Thriller', 'English', 118, '1991-02-14');

commit;

INSERT INTO Content (Title, Description, Cast, Genre, Language, Run\_time, Release\_date) VALUES

('The Queen''s Gambit', 'A young orphaned girl discovers a talent for chess', 'Anya Taylor-Joy, Thomas Brodie-Sangster', 'Drama', 'English', 394, '2020-10-23'),

('Bridgerton', 'A period drama about the lives of the wealthy Bridgerton family', 'Phoebe Dynevor, Regé-Jean Page', 'Romance', 'English', 378, '2020-12-25'),

('Stranger Things', 'A group of friends investigate supernatural occurrences in their town', 'Millie Bobby Brown, Finn Wolfhard', 'Science fiction', 'English', 417, '2016-07-15'),

('The Handmaid''s Tale', 'A woman tries to survive in a totalitarian society', 'Elisabeth Moss, Joseph Fiennes', 'Drama', 'English', 420, '2017-04-26'),

('Breaking Bad', 'A high school chemistry teacher starts making and selling crystal meth to provide for his family', 'Bryan Cranston, Aaron Paul', 'Crime', 'English', 327, '2008-01-20'),

('The Sopranos', 'A mob boss tries to balance his criminal life with his family life', 'James Gandolfini, Lorraine Bracco', 'Crime', 'English', 502, '1999-01-10'),

('Game of Thrones', 'The power struggle between several noble families for control of the Iron Throne', 'Emilia Clarke, Kit Harington', 'Fantasy', 'English', 439, '2011-04-17'),

('The Crown', 'The life of Queen Elizabeth II from her early reign to the present day', 'Claire Foy, Olivia Colman', 'Drama', 'English', 515, '2016-11-04'),

('The Big Bang Theory', 'A group of nerdy friends navigate their lives and relationships', 'Jim Parsons, Kaley Cuoco', 'Sitcom', 'English', 279, '2007-09-24'),

('How I Met Your Mother', 'A man recounts the story of how he met his children''s mother to his friends', 'Josh Radnor, Jason Segel', 'Sitcom', 'English', 208, '2005-09-19');

commit;

INSERT INTO Content (Title, Description, Cast, Genre, Language, Run\_time, Release\_date) VALUES

('Baahubali: The Beginning', 'A warrior prince tries to regain his kingdom from his cruel uncle with the help of a skilled warrior and a mysterious woman.', 'Prabhas, Rana Daggubati, Anushka Shetty, Tamannaah', 'Action, Drama, Fantasy', 'Telugu', 159, '2015-07-10'),

('3 Idiots', 'Two friends embark on a quest for a lost buddy. On this journey, they encounter a long-forgotten bet, a wedding they must crash, and a funeral that goes, ridiculously out of control.', 'Aamir Khan, R. Madhavan, Sharman Joshi', 'Comedy, Drama', 'Hindi', 170, '2009-12-23'),

('Dangal', 'Former wrestler Mahavir Singh Phogat and his two wrestler daughters struggle towards glory at the Commonwealth Games in the face of societal oppression.', 'Aamir Khan, Sakshi Tanwar,Sanya Malhotra', 'Action, Biography, Drama', 'Hindi', 161, '2016-12-23'),

('Kabhi Khushi Kabhie Gham', 'Rahul, the adoptive son of business magnate Yash Raichand, feels eternal gratitude to his father for rescuing him from a life of poverty. However, when Rahul is disinherited, he sets out to seek revenge.', 'Shah Rukh Khan, Kajol, Amitabh Bachchan, Jaya Bachchan', 'Drama, Musical, Romance', 'Hindi', 210, '2001-12-14'),

('Vikram Vedha', 'Vikram, a no-nonsense police officer, accompanied by Simon, his partner, is on the hunt to capture Vedha, a smuggler and a murderer. Vedha tries to change Vikrams life, which leads to a conflict.', 'Madhavan, Vijay Sethupathi, Shraddha Srinath, Kathir', 'Action, Crime, Drama', 'Tamil', 147, '2017-07-21');

commit;

INSERT INTO Content (Title, Description, Cast, Genre, Language, Run\_time, Release\_date) VALUES

("KGF: Chapter 1", "A period action film", "Yash, Srinidhi Shetty, Ramachandra Raju", "Action, Drama", "Kannada", 155, "2018-12-21"),

("Baahubali 2: The Conclusion", "A historical action film", "Prabhas, Anushka Shetty, Rana Daggubati", "Action, Drama", "Telugu", 168, "2017-04-28"),

("Sarkar", "A political action thriller film", "Vijay, Keerthy Suresh, Varalaxmi Sarathkumar", "Action, Drama", "Tamil", 163, "2018-11-06"),

("Master", "An action thriller film", "Vijay, Vijay Sethupathi, Malavika Mohanan", "Action, Drama", "Tamil", 179, "2021-01-13"),

("Rangasthalam", "A period action drama film", "Ram Charan, Samantha Akkineni, Jagapathi Babu", "Action, Drama", "Telugu", 179, "2018-03-30"),

("Ala Vaikunthapurramuloo", "A comedy-drama film", "Allu Arjun, Pooja Hegde, Tabu", "Comedy, Drama", "Telugu", 165, "2020-01-12"),

("Geetha Govindam", "A romantic comedy film", "Vijay Deverakonda, Rashmika Mandanna, Nithya Menen", "Romance, Comedy", "Telugu", 142, "2018-08-15"),

("Dear Comrade", "A romantic action drama film", "Vijay Deverakonda, Rashmika Mandanna, Shruti Ramachandran", "Romance, Drama", "Telugu", 169, "2019-07-26"),

("Arjun Reddy", "A romantic drama film", "Vijay Deverakonda, Shalini Pandey, Jia Sharma", "Romance, Drama", "Telugu", 187, "2017-08-25"),

("Jathi Ratnalu", "A comedy film", "Naveen Polishetty, Priyadarshi Pulikonda, Rahul Ramakrishna", "Comedy, Drama", "Telugu", 148, "2021-03-11"),

("Maharshi", "A action drama film", "Mahesh Babu, Pooja Hegde, Allari Naresh", "Action, Drama", "Telugu", 176, "2019-05-09"),

("Krack", "An action thriller film", "Ravi Teja, Shruti Haasan, Varalaxmi Sarathkumar", "Action, Thriller", "Telugu", 154, "2021-01-09"),

("Pushpa: The Rise", "An action thriller film", "Allu Arjun, Rashmika Mandanna, Fahadh Faasil", "Action, Thriller", "Telugu", 165, "2021-12-17");

commit;

**Output:**

# A screenshot of a computer Description automatically generated with medium confidence

# Creating Payment Table

# We have created a payment table by using the following create statement.

**Query:** CREATE TABLE Payment (

Payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

User\_id INT NOT NULL,

Payment\_method VARCHAR(20) NOT NULL,

Payment\_date DATETIME NOT NULL,

Payment\_status VARCHAR(20) NOT NULL,

Payment\_amount DECIMAL(10,2) NOT NULL,

Subscription\_plan VARCHAR(20) NOT NULL,

Billing\_address VARCHAR(50) NOT NULL,

FOREIGN KEY (User\_id) REFERENCES User\_Profile(User\_id)

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO Payment (User\_id, Payment\_method, Payment\_date, Payment\_status, Payment\_amount, Subscription\_plan, Billing\_address)

VALUES

(1, 'Credit Card', '2023-05-01 10:30:00', 'Completed', 99.99, 'Premium', '123 Main St, Anytown, USA'),

(2, 'PayPal', '2023-05-01 12:15:00', 'Completed', 49.99, 'Basic', '456 High St, Anytown, USA'),

(3, 'Credit Card', '2023-05-02 14:00:00', 'Pending', 149.99, 'Premium', '789 Elm St, Anytown, USA'),

(4, 'PayPal', '2023-05-03 16:45:00', 'Completed', 79.99, 'Basic', '321 Oak St, Anytown, USA'),

(5, 'Credit Card', '2023-05-04 18:30:00', 'Completed', 129.99, 'Premium', '654 Pine St, Anytown, USA'),

(6, 'PayPal', '2023-05-05 20:15:00', 'Pending', 69.99, 'Basic', '987 Maple St, Anytown, USA'),

(7, 'Credit Card', '2023-05-06 22:00:00', 'Completed', 179.99, 'Premium', '111 Cherry St, Anytown, USA'),

(8, 'PayPal', '2023-05-07 23:45:00', 'Completed', 99.99, 'Basic', '222 Peach St, Anytown, USA'),

(9, 'Credit Card', '2023-05-08 01:30:00', 'Pending', 199.99, 'Premium', '333 Plum St, Anytown, USA'),

(10, 'PayPal', '2023-05-09 03:15:00', 'Completed', 149.99, 'Basic', '444 Grape St, Anytown, USA'),

(11, 'Credit Card', '2023-05-10 05:00:00', 'Completed', 249.99, 'Premium', '555 Apple St, Anytown, USA'),

(12, 'PayPal', '2023-05-11 06:45:00', 'Pending', 119.99, 'Basic', '666 Pear St, Anytown, USA'),

(13, 'Credit Card', '2023-05-12 08:30:00', 'Completed', 299.99, 'Premium', '777 Banana St, Anytown, USA'),

(14, 'PayPal', '2023-05-13 10:15:00', 'Completed', 199.99, 'Basic', '888 Orange St, Anytown, USA'),

(15, 'Credit Card', '2023-05-14 12:00:00', 'Pending', 399.99, 'Premium', '999 Lemon St, Anytown, USA');

INSERT INTO Payment (User\_id, Payment\_method, Payment\_date, Payment\_status, Payment\_amount, Subscription\_plan, Billing\_address) VALUES

(16, 'Credit Card', '2023-05-10 10:30:00', 'Paid', 29.99, 'Basic', '123 Main St, Anytown USA'),

(17, 'PayPal', '2023-05-11 11:45:00', 'Paid', 49.99, 'Premium', '456 Broad Ave, Anytown USA'),

(18, 'Venmo', '2023-05-12 09:15:00', 'Paid', 19.99, 'Basic', '789 Oak St, Anytown USA'),

(19, 'Credit Card', '2023-05-13 13:00:00', 'Pending', 29.99, 'Basic', '246 Elm St, Anytown USA'),

(20, 'PayPal', '2023-05-14 14:30:00', 'Paid', 99.99, 'Premium', '369 Pine St, Anytown USA'),

(21, 'Venmo', '2023-05-15 16:45:00', 'Paid', 39.99, 'Basic', '258 Maple Ave, Anytown USA'),

(22, 'Credit Card', '2023-05-16 08:30:00', 'Paid', 19.99, 'Basic', '147 Cherry St, Anytown USA'),

(23, 'PayPal', '2023-05-17 10:00:00', 'Paid', 49.99, 'Premium', '369 Oak St, Anytown USA'),

(24, 'Venmo', '2023-05-18 12:15:00', 'Pending', 29.99, 'Basic', '258 Main St, Anytown USA'),

(25, 'Credit Card', '2023-05-19 15:30:00', 'Paid', 79.99, 'Premium', '123 Broad Ave, Anytown USA'),

(26, 'PayPal', '2023-05-20 17:45:00', 'Paid', 19.99, 'Basic', '789 Pine St, Anytown USA'),

(27, 'Venmo', '2023-05-21 08:30:00', 'Paid', 29.99, 'Basic', '456 Oak St, Anytown USA'),

(28, 'Credit Card', '2023-05-22 10:00:00', 'Paid', 39.99, 'Basic', '369 Maple Ave, Anytown USA'),

(29, 'PayPal', '2023-05-23 12:15:00', 'Pending', 29.99, 'Basic', '147 Main St, Anytown USA'),

(30, 'Venmo', '2023-05-24 15:30:00', 'Paid', 99.99, 'Premium', '258 Broad Ave, Anytown USA');

INSERT INTO Payment (User\_id, Payment\_method, Payment\_date, Payment\_status, Payment\_amount, Subscription\_plan, Billing\_address)

VALUES

(31, 'Credit Card', '2023-05-10 14:30:00', 'Completed', 15.00, 'Monthly', '123 Main St'),

(32, 'PayPal', '2023-05-10 15:00:00', 'Completed', 20.00, 'Monthly', '456 Oak Ave'),

(33, 'Debit Card', '2023-05-10 16:30:00', 'Pending', 10.00, 'Monthly', '789 Elm St'),

(34, 'Credit Card', '2023-05-10 17:00:00', 'Completed', 25.00, 'Yearly', '456 Maple Ave'),

(35, 'PayPal', '2023-05-10 18:30:00', 'Pending', 30.00, 'Yearly', '789 Oak St'),

(36, 'Debit Card', '2023-05-10 19:00:00', 'Pending', 40.00, 'Yearly', '123 Elm St'),

(37, 'Credit Card', '2023-05-10 20:30:00', 'Pending', 50.00, 'Yearly', '456 Pine Ave'),

(38, 'PayPal', '2023-05-10 21:00:00', 'Completed', 60.00, 'Yearly', '789 Maple St');

commit;

**Output:**

A screenshot of a computer

Description automatically generated

# Creating Comment Table

* Then we created table comment by using the following create statement.

**Query:** CREATE TABLE Comment (

Comment\_id INT AUTO\_INCREMENT PRIMARY KEY,

Content\_id INT NOT NULL,

User\_id INT NOT NULL,

Comment VARCHAR(80) NOT NULL,

FOREIGN KEY (Content\_id) REFERENCES Content(Content\_id),

FOREIGN KEY (User\_id) REFERENCES User\_Profile(User\_id)

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO Comment (Content\_id, User\_id, Comment)

VALUES (1, 2, 'Great article, very informative!'),

(1, 3, 'I found this article really helpful, thanks!'),

(2, 4, 'Interesting perspective, I never thought of it that way.'),

(3, 1, 'This is amazing, thank you for sharing!'),

(4, 5, 'I completely disagree with the author on this topic.'),

(4, 6, 'I think the author makes some good points, but overall I disagree.'),

(5, 2, 'I love this book, its one of my favorites!'),

(5, 3, 'I read this book for a class and found it very insightful.'),

(5, 4, 'I didnt really enjoy this book, but I can see why others might.'),

(6, 1, 'I think this movie is overrated.'),

(6, 5, 'This movie is a classic, one of my all-time favorites.'),

(7, 3, 'This song is so catchy, I cant stop listening to it!'),

(7, 4, 'I dont really like this genre of music, but I appreciate the talent.'),

(8, 2, 'Ive never heard of this band before, but I really like their sound.'),

(9, 6, 'I think the lyrics of this song are really powerful.'),

(10, 1, 'Ive always loved this painting, its so beautiful.'),

(10, 4, 'I dont really understand this style of art, but I respect it.'),

(11, 5, 'This sculpture is amazing, I cant believe its made out of marble.'),

(12, 2, 'I saw this play last week and it was incredible.'),

(13, 1, 'I think this poem is really touching and relatable.'),

(14, 3, 'This video game is so much fun, I cant put it down!'),

(15, 6, 'Im not a fan of video games, but I can see why people enjoy this one. ');

INSERT INTO Comment (Content\_id, User\_id, Comment) VALUES

(1, 4, 'Great article, thanks for sharing!'),

(1, 2, 'I found this really informative.'),

(2, 5, 'I disagree with some of the points made here.'),

(2, 6, 'Interesting perspective.'),

(3, 3, 'Ive been looking for something like this.'),

(3, 1, 'This was exactly what I needed.'),

(4, 2, 'This article needs more research.'),

(4, 6, 'I appreciate the author for taking the time to write this.'),

(5, 5, 'I had a similar experience.'),

(5, 1, 'This was a great read.'),

(6, 4, 'I found this to be really helpful.'),

(6, 3, 'Thanks for sharing your knowledge.'),

(7, 2, 'I couldnt agree more.'),

(7, 1, 'This was a bit confusing.'),

(8, 5, 'I would love to see more on this topic.'),

(8, 6, 'Great job on this article.'),

(9, 3, 'Ive been struggling with this issue.'),

(9, 4, 'This was a game changer for me.'),

(10, 1, 'I think this could have been more detailed.'),

(10, 2, 'This was well-written and informative.');

INSERT INTO Comment (Content\_id, User\_id, Comment) VALUES

(11, 6, 'I disagree with some of the points made here.'),

(11, 5, 'This article needs more research.'),

(12, 3, 'I found this really informative.'),

(12, 4, 'Thanks for sharing your knowledge.'),

(13, 1, 'Interesting perspective.'),

(13, 2, 'Ive been looking for something like this.'),

(14, 6, 'I appreciate the author for taking the time to write this.'),

(14, 5, 'I had a similar experience.'),

(15, 4, 'This was a great read.'),

(15, 3, 'I found this to be really helpful.'),

(16, 2, 'I couldnt agree more.'),

(16, 1, 'This was a bit confusing.'),

(17, 5, 'Great job on this article.'),

(17, 6, 'I would love to see more on this topic.'),

(18, 4, 'Ive been struggling with this issue.'),

(18, 3, 'This was a game changer for me.'),

(19, 1, 'I think this could have been more detailed.'),

(19, 2, 'This was well-written and informative.'),

(20, 6, 'Interesting read, thanks for sharing.');

INSERT INTO Comment (Content\_id, User\_id, Comment) VALUES

(1, 1, 'I completely agree with this.'),

(1, 2, 'This article was well-written and informative.'),

(1, 3, 'I had a different experience, but still found this interesting.'),

(12, 1, 'Great read, thanks for sharing.'),

(12, 4, 'This was really helpful for me.'),

(13, 5, 'I had never thought of it this way before.'),

(13, 6, 'This was a bit confusing, could you clarify?'),

(14, 3, 'Thanks for sharing your experience.'),

(14, 2, 'This was really insightful.'),

(15, 6, 'I disagree with some of the points made here.'),

(15, 5, 'This article needs more research.'),

(16, 4, 'I found this really informative.'),

(16, 3, 'Thanks for sharing your knowledge.'),

(17, 1, 'Interesting perspective.'),

(17, 2, 'Ive been looking for something like this.'),

(18, 6, 'I appreciate the author for taking the time to write this.'),

(18, 5, 'I had a similar experience.'),

(19, 4, 'This was a great read.'),

(19, 3, 'I found this to be really helpful.'),

(20, 2, 'I couldnt agree more.'),

(20, 1, 'This was a bit confusing.'),

(21, 5, 'Great job on this article.'),

(21, 6, 'I would love to see more on this topic.'),

(22, 4, 'Ive been struggling with this issue.'),

(22, 3, 'This was a game changer for me.'),

(23, 1, 'I think this could have been more detailed.'),

(23, 2, 'This was well-written and informative.'),

(24, 6, 'Interesting read, thanks for sharing.'),

(24, 5, 'This gave me a lot to think about.'),

(25, 4, 'I had a different perspective, but this was still insightful.'),

(25, 3, 'I found this to be really helpful.'),

(26, 1, 'This was a bit confusing, could you clarify?'),

(26, 2, 'I completely agree with this.'),

(27, 6, 'This article needs more research.'),

(27, 5, 'Thanks for sharing your experience.'),

(28, 4, 'This was really helpful for me.'),

(28, 3, 'I had a different experience, but still found this interesting.'),

(29, 1, 'I had never thought of it this way before.'),

(29, 2, 'This was a bit confusing, could you clarify?'),

(30, 6, 'This was really insightful.'),

(30, 5, 'Thanks for sharing your knowledge.'),

(31, 4, 'Interesting perspective.'),

(31, 3, 'Ive been looking for something like this.'),

(32, 1, 'I appreciate the author for taking the time to write this.'),

(32, 2, 'I would love to see more on this topic.'),

(33, 6, 'Ive been struggling with this issue.'),

(33, 5, 'This was a game changer for me.'),

(34, 4, 'This was a bit confusing, could you clarify?'),

(34, 3, 'I completely agree with this.');

commit;

**Output:**

**A screenshot of a computer

Description automatically generated with medium confidence**

# Creating History Table

* Then we created history table by using the following create statement.

**Query:** CREATE TABLE History (

History\_id INT(4) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

User\_id INT(4) NOT NULL,

Content\_id INT(4) NOT NULL,

Date DATETIME NOT NULL,

Duration INT(4) UNSIGNED NOT NULL,

CONSTRAINT fk\_user\_history FOREIGN KEY (User\_id) REFERENCES User\_Profile(User\_id),

CONSTRAINT fk\_content\_history FOREIGN KEY (Content\_id) REFERENCES Content(Content\_id)

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO History (User\_id, Content\_id, Date, Duration) VALUES

(1, 1, '2022-05-09 10:32:00', 10),

(2, 1, '2022-05-09 10:45:00', 15),

(3, 1, '2022-05-09 11:02:00', 20),

(4, 1, '2022-05-09 11:25:00', 30),

(5, 1, '2022-05-09 12:02:00', 45),

(6, 1, '2022-05-09 12:45:00', 10),

(1, 12, '2022-05-09 13:15:00', 20),

(2, 12, '2022-05-09 13:35:00', 25),

(3, 12, '2022-05-09 14:02:00', 30),

(4, 12, '2022-05-09 14:32:00', 40),

(5, 12, '2022-05-09 15:12:00', 15),

(6, 12, '2022-05-09 15:27:00', 10),

(1, 13, '2022-05-09 16:05:00', 20),

(2, 13, '2022-05-09 16:25:00', 25),

(3, 13, '2022-05-09 16:52:00', 30),

(4, 13, '2022-05-09 17:22:00', 40),

(5, 13, '2022-05-09 18:02:00', 15),

(6, 13, '2022-05-09 18:17:00', 10),

(1, 14, '2022-05-09 18:45:00', 20),

(2, 14, '2022-05-09 19:05:00', 25);

commit;

INSERT INTO History (User\_id, Content\_id, Date, Duration) VALUES

(1, 29, '2022-04-01 08:21:12', 150),

(2, 35, '2022-05-02 14:11:23', 75),

(3, 24, '2022-04-19 11:45:34', 120),

(4, 27, '2022-04-15 16:30:45', 90),

(5, 30, '2022-04-02 19:12:56', 60),

(6, 33, '2022-04-10 22:40:07', 180),

(1, 38, '2022-05-06 07:05:18', 45),

(2, 40, '2022-05-08 13:20:29', 90),

(3, 36, '2022-04-23 09:15:40', 120),

(4, 39, '2022-05-05 17:30:51', 60);

commit;

INSERT INTO History (User\_id, Content\_id, Date, Duration) VALUES

(5, 14, '2023-05-06 10:23:40', 150),

(1, 19, '2023-05-06 11:45:12', 75),

(6, 30, '2023-05-06 12:33:27', 60),

(4, 36, '2023-05-06 13:15:58', 90),

(2, 27, '2023-05-06 14:10:24', 120),

(3, 33, '2023-05-06 15:22:15', 45),

(5, 26, '2023-05-06 16:05:59', 30),

(1, 38, '2023-05-06 17:10:40', 120),

(6, 22, '2023-05-06 18:30:10', 90),

(4, 40, '2023-05-06 19:45:52', 60),

(3, 35, '2023-05-07 10:18:09', 75),

(2, 31, '2023-05-07 11:35:24', 45),

(1, 16, '2023-05-07 12:40:02', 30),

(6, 20, '2023-05-07 13:25:58', 60),

(5, 25, '2023-05-07 14:15:12', 90),

(4, 23, '2023-05-07 15:32:45', 120),

(3, 29, '2023-05-07 16:25:50', 75),

(2, 37, '2023-05-07 17:55:40', 30),

(1, 34, '2023-05-07 18:40:11', 45),

(6, 39, '2023-05-07 19:25:16', 60);

commit;

**Output:**

A screenshot of a data

Description automatically generated with medium confidence

# Creating Play List Table

* Then we created Play\_list table by using the following create statement.

**Query:** CREATE TABLE Play\_List (

User\_id INT NOT NULL,

Playlist\_id INT AUTO\_INCREMENT PRIMARY KEY,

Playlist\_name VARCHAR(20) NOT NULL,

Created\_name VARCHAR(20) NOT NULL,

Content\_id INT NOT NULL,

Playlist\_content VARCHAR(100) NOT NULL,

FOREIGN KEY (User\_id) REFERENCES User\_Profile(User\_id),

FOREIGN KEY (Content\_id) REFERENCES Content(Content\_id)

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO Play\_List (User\_id, Playlist\_name, Created\_name, Content\_id, Playlist\_content)

VALUES

(1, 'My Favorites', 'John', 12, 'Inception'),

(1, 'My Favorites', 'John', 15, 'Pulp Fiction'),

(2, 'Watch Later', 'Jane', 14, 'Goodfellas'),

(2, 'Watch Later', 'Jane', 18, 'The Shawshank Redemption'),

(3, 'Action Movies', 'Bob', 20, 'The Matrix'),

(3, 'Action Movies', 'Bob', 21, 'Jurassic Park');

commit;

INSERT INTO Play\_List (User\_id, Playlist\_name, Created\_name, Content\_id, Playlist\_content) VALUES

(2, 'Favorite Crime Films', 'Me', 1, 'The Godfather'),

(2, 'Favorite Crime Films', 'Me', 14, 'Goodfellas'),

(2, 'Favorite Crime Films', 'Me', 15, 'Pulp Fiction'),

(3, 'Top Movies of All Time', 'Amy', 12, 'Inception'),

(3, 'Top Movies of All Time', 'Amy', 13, 'The Social Network'),

(3, 'Top Movies of All Time', 'Amy', 18, 'The Shawshank Redemption'),

(3, 'Top Movies of All Time', 'Amy', 19, 'Forrest Gump'),

(3, 'Top Movies of All Time', 'Amy', 25, 'The Lord of the Rings: The Fellowship of the Ring'),

(4, 'Sci-Fi Flicks', 'Sam', 12, 'Inception'),

(4, 'Sci-Fi Flicks', 'Sam', 20, 'The Matrix'),

(4, 'Sci-Fi Flicks', 'Sam', 21, 'Jurassic Park'),

(5, 'All About Crime', 'Dan', 1, 'The Godfather'),

(5, 'All About Crime', 'Dan', 14, 'Goodfellas'),

(5, 'All About Crime', 'Dan', 17, 'The Dark Knight'),

(5, 'All About Crime', 'Dan', 23, 'The Godfather');

commit;

INSERT INTO Play\_List (User\_id, Playlist\_name, Created\_name, Content\_id, Playlist\_content) VALUES

(5, 'Sci-Fi Movies', 'Maya', 12, 'Inception, 2010'),

(5, 'Drama Classics', 'Maya', 14, 'Goodfellas, 1990'),

(3, 'Movie Night', 'John', 15, 'Pulp Fiction, 1994'),

(4, 'My Favorite Movies', 'Sarah', 16, 'The Godfather, 1972'),

(4, 'Action-Packed', 'Sarah', 17, 'The Dark Knight, 2008'),

(2, 'Timeless Classics', 'David', 18, 'The Shawshank Redemption, 1994'),

(1, 'Blockbusters', 'Emily', 19, 'Forrest Gump, 1994'),

(2, 'Mind-Bending Movies', 'David', 20, 'The Matrix, 1999'),

(3, 'Adventure Films', 'John', 21, 'Jurassic Park, 1993'),

(1, 'All-Time Favourites', 'Emily', 22, 'The Shawshank Redemption, 1994');

commit;

**Output:**

**A screenshot of a computer

Description automatically generated with medium confidence**

# Creating Rating Table

* Then we created Rating table by using the following create statement.

**Query:** CREATE TABLE rating (

Rating\_id INT(4) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

User\_id INT(4) NOT NULL,

Ratings INT(1) NOT NULL,

Reviews VARCHAR(100),

Content\_id INT(4) NOT NULL,

CONSTRAINT fk\_user\_rating FOREIGN KEY (User\_id) REFERENCES User\_Profile(User\_id),

CONSTRAINT fk\_content\_rating FOREIGN KEY (Content\_id) REFERENCES Content(Content\_id)

);

**Inserting Records**

* Then we inserted the following records by using insert statement.

INSERT INTO rating (User\_id, Ratings, Reviews, Content\_id) VALUES

(1, 4, 'Great movie', 1),

(2, 3, 'Okay documentary', 13),

(3, 2, 'Disappointing ending', 15),

(4, 5, 'Amazing show', 29),

(5, 4, 'Really enjoyed this', 21),

(6, 1, 'Not my cup of tea', 18),

(1, 3, 'Decent watch', 27),

(2, 4, 'Highly recommend', 12),

(3, 2, 'Expected more', 23),

(4, 5, 'One of the best I have seen', 30),

(5, 4, 'Good movie overall', 16),

(6, 3, 'Solid performance by the actors', 35),

(1, 1, 'Waste of time', 38),

(2, 4, 'Great cinematography', 14),

(3, 3, 'Interesting plot', 31),

(4, 2, 'Not what I expected', 19),

(5, 5, 'Absolutely loved it', 26),

(6, 4, 'Definitely worth watching', 40),

(1, 2, 'Didn''t live up to the hype', 22),

(2, 3, 'Entertaining show', 34);

commit;

INSERT INTO rating (User\_id, Ratings, Reviews, Content\_id) VALUES

(1, 5, 'Great content!', 17),

(2, 4, 'Good read', 20),

(3, 3, 'Needs improvement', 24),

(4, 2, 'Not what I expected', 28),

(5, 5, 'Excellent work', 31),

(6, 4, 'Interesting insights', 35),

(1, 3, 'Meh', 39),

(2, 4, 'Liked it', 13),

(3, 2, 'Disappointing', 14),

(4, 5, 'Really enjoyed this', 21),

(5, 1, 'Waste of time', 23),

(6, 4, 'Well written', 26),

(1, 2, 'Could be better', 29),

(2, 5, 'Impressed', 32),

(3, 4, 'Good stuff', 36),

(4, 3, 'Decent article', 37),

(5, 2, 'Needs more depth', 15),

(6, 5, 'Brilliant!', 16),

(1, 4, 'Informative', 19),

(2, 3, 'Average', 22);

commit;

INSERT INTO rating (User\_id, Ratings, Reviews, Content\_id) VALUES

(1, 3, 'This was okay', 25),

(2, 4, 'I really enjoyed this', 26),

(3, 2, 'Not my favorite', 27),

(4, 5, 'This was amazing!', 28),

(5, 4, 'Really informative', 29),

(6, 3, 'Could be better', 30),

(1, 5, 'Loved it!', 31),

(2, 3, 'Interesting topic', 32),

(3, 1, 'Disappointing', 33),

(4, 4, 'Great read', 34),

(5, 2, 'Could have been more detailed', 35),

(6, 4, 'Thoroughly enjoyed this', 36),

(1, 4, 'Worth a read', 37),

(2, 5, 'Highly recommend', 38),

(3, 3, 'Decent enough', 39);

commit;

INSERT INTO rating (User\_id, Ratings, Reviews, Content\_id) VALUES

(5, 4, 'Great content!', 27),

(2, 2, 'Could be better.', 38),

(1, 5, 'Loved it!', 21);

commit;

**Output:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Data retrieval from Queries**

**1.Number of ratings per category:** The below query is used to retrieve the number of ratings based on ratings in the table i.e., 1,2,3,4,5.

We can see that the number of ratings for 4 is more than others while for 1 is less than others.

Select ratings as Rating, count(ratings) as NO\_OF\_RATINGS

from rating

group by ratings

order by Rating desc;

A screenshot of a computer

Description automatically generated with low confidence

**2.Content watched by single user:** The below query is used to retrieve the number of content watched by the user.  
 The query gives the user\_id, user full name by concatenating the first and last name. We joined the history table and user\_profile table by user\_id as the key.

We can see that user John Doe who has user\_id as ‘1’ watched more content than others.

Select h.user\_id as USER\_ID,CONCAT(u.First\_name,' ',u.last\_name) as FULL\_NAME,

count(history\_id) as No\_of\_content\_watched

from history h , user\_profile u

where h.user\_id=u.user\_id group by user\_id;

A screenshot of a computer

Description automatically generated with medium confidence

**3. Pending payments with different payment methods:**

The below query is used to retrieve the number of payments made by users using different types of payment methods and payment\_status is ‘Pending’.

We can see that the number of payments from credit card types has more number which means that this type of payment is more like to make payments later than others.

select payment\_method,count(payment\_method) No\_of\_Payments

from payment

where upper(payment\_status) ='PENDING'

group by payment\_method;

A screenshot of a computer

Description automatically generated with medium confidence

**4. Completed payments with different payment methods:**

The below query is used to retrieve the number of payments made by users using several types of payment methods and payment\_status is ‘Completed.

We can see that the number of payments from credit card type has more number which means that this type of payment is more like used to make payments by users and it has a greater number of completed status of payment.

select payment\_method,count(payment\_method) No\_of\_Payments

from payment

where upper(payment\_status) ='COMPLETED'

group by payment\_method;

A screenshot of a computer

Description automatically generated with medium confidence

**5. Total number of payments with different payment methods:**

The below query is used to retrieve the number of payments made by users using several types of payment methods.

We can see that the number of payments from credit card type has more number which means that this type of payment is more like used to make payments by users.

select payment\_method,count(payment\_method) No\_of\_payments

from payment

group by payment\_method;

A screenshot of a computer

Description automatically generated with medium confidence

**6.Number of Subscribers per subscription plan.**

The below query is used to retrieve the number of subscriptions taken by users for each subscription\_plan.

We can see that **premium** subscription\_plan has the greater number of subscriptions taken by users than others while monthly plan has the less number of subscriptions.

select Subscription\_plan ,count(Subscription\_plan) NO\_OF\_SUBSCRIBERS

from payment

group by Subscription\_plan;

A screenshot of a computer

Description automatically generated with medium confidence

**7.Content per language.**

The below query is used to retrieve the number of contents/films per language.

We can see that the number of content/films for English is more than the other languages.

select count(\*) No\_of\_films,language

from content

group by language;

A screenshot of a computer

Description automatically generated with medium confidence

**8. Action Content with Language**

The below query is used to retrieve the Title, Genre and language of the content which has genre has ‘Action’.

select Title,Genre,Language

from content

where genre like '%Action%';

A screenshot of a computer

Description automatically generated with medium confidence

**Conclusion:**

In conclusion, the above data provides insights on various aspects of the business such as user behavior, payment methods, subscriptions, and content. The number of ratings for 4 is more than others while for 1 is less than others. User John Doe watched more content than others. The credit card payment method is more likely to make payments later and is more commonly used for completed payments. Premium subscription\_plan has the greater number of subscriptions taken by users than others while monthly plan has the smaller number of subscriptions. English language has the highest number of contents/films. Finally, the query for action content provides the Title, Genre, and Language of the content. These insights can be used to make informed decisions and improve business strategy.