**UNIVERSITY INSTITUTE OF COMPUTING**

**CASE STUDY REPORT**

**ON**

**Movie Streaming Platform Database**

Program Name: BCA

Subject Name/Code: Database Management System (23CAT-251)

**Submitted by: Submitted to:**

**Name:** Sanchi Jangra  **Name:** Arvinder Singh

**UID:** 23BCA10401 **Designation:**

**Section:** 4’b

ABSTRACT

* Introduction:

**Movie Streaming Platform Database**

This comprehensive relational database system is designed to support a Netflix-like streaming service with robust functionality for content management, user interactions, and business operations. The database implements a normalized schema with 15+ interconnected tables handling user accounts, viewing profiles, content catalog (movies and TV shows), viewing history, recommendations, subscription management, and device tracking.

**Key features include:**

- Multi-profile user accounts with personalized watchlists and viewing history

- Hierarchical content structure (TV shows → seasons → episodes)

- Content categorization through genre and people (actors, directors) relationships

- Recommendation engine with both content-based and user-based algorithms

- Subscription and payment processing with multiple plan options

- Device management and session tracking for security

- Performance optimization through indexes, stored procedures, and views

The database includes sample data, stored procedures for generating recommendations, triggers for rating updates, and views for trending content, making it production-ready for a streaming platform project.

* Technique:

**The movie streaming platform database employs several key database design techniques:**

1. Relational Database Design: Uses normalized tables with primary and foreign keys to establish relationships between entities (users, profiles, content, etc.)

2. Database Normalization: Implements 3NF (Third Normal Form) to minimize data redundancy while maintaining data integrity

3. Many-to-Many Relationship Handling: Uses junction tables (content\_genres, content\_people) to manage complex relationships

4. Performance Optimization: Employs indexes, stored procedures, and views to enhance query performance

5. Data Integrity Enforcement: Uses constraints, triggers (for rating updates), and transaction management

6. Advanced SQL Features: Implements stored procedures for recommendation generation, views for trending content, and triggers for automated updates

This comprehensive approach creates a robust foundation for a streaming platform with features like user management, content categorization, viewing history tracking, and recommendation systems.

* System Configuration:

**Based on the movie streaming platform database files, the recommended system configurations would include:**

1. Database System: MySQL/MariaDB (evidenced by MySQL-specific syntax like DELIMITER statements)

2. Storage Requirements: Minimum 10GB for initial database with growth capacity to 100GB+ for content expansion

3. Memory: 16GB+ RAM for optimal performance with complex queries and stored procedures

4. CPU: Multi-core processor (8+ cores) to handle concurrent transactions

5. Network: High-bandwidth connection for streaming data transfers

6. Backup System: Daily incremental backups with weekly full backups

7. Security: Encrypted connections (SSL/TLS) and data-at-rest encryption

8. Scaling: Horizontal scaling capability for read operations with potential for sharding as user base grows

This configuration supports the complex relational model with multiple interconnected tables, stored procedures, triggers, and views while maintaining performance for a production streaming service.

* INPUT:

--create database Movie\_Streaming\_Platform;

Use Movie\_Streaming\_Platform;

CREATE TABLE admins (

admin\_id SERIAL PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password\_hash VARCHAR(255) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

role VARCHAR(20) NOT NULL CHECK (role IN ('super\_admin', 'content\_manager', 'user\_support', 'analytics')),

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

last\_login TIMESTAMP

);

CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

email VARCHAR(100) NOT NULL UNIQUE,

password\_hash VARCHAR(255) NOT NULL,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

date\_of\_birth DATE,

country VARCHAR(50),

phone\_number VARCHAR(20),

is\_verified BOOLEAN DEFAULT FALSE,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

last\_login TIMESTAMP

);

CREATE TABLE profiles (

profile\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id) ON DELETE CASCADE,

profile\_name VARCHAR(50) NOT NULL,

avatar\_url VARCHAR(255),

is\_kids\_profile BOOLEAN DEFAULT FALSE,

language\_preference VARCHAR(10) DEFAULT 'en',

content\_maturity\_level VARCHAR(10) DEFAULT 'adult' CHECK (content\_maturity\_level IN ('kids', 'teen', 'adult')),

autoplay\_enabled BOOLEAN DEFAULT TRUE,

subtitle\_preference VARCHAR(10) DEFAULT 'off',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(user\_id, profile\_name)

);

CREATE TABLE people (

person\_id SERIAL PRIMARY KEY,

full\_name VARCHAR(100) NOT NULL,

birth\_date DATE,

biography TEXT,

nationality VARCHAR(50),

profile\_image\_url VARCHAR(255),

imdb\_id VARCHAR(20),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE genres (

genre\_id SERIAL PRIMARY KEY,

name VARCHAR(50) NOT NULL UNIQUE,

description TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE content (

content\_id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

type VARCHAR(10) NOT NULL CHECK (type IN ('movie', 'tvshow')),

description TEXT,

release\_year INTEGER CHECK (release\_year > 1900 AND release\_year <= 2030),

age\_rating VARCHAR(10) CHECK (age\_rating IN ('G', 'PG', 'PG-13', 'R', 'NC-17', 'TV-Y', 'TV-Y7', 'TV-G', 'TV-PG', 'TV-14', 'TV-MA')),

duration\_minutes INTEGER CHECK (duration\_minutes > 0),

poster\_url VARCHAR(255),

backdrop\_url VARCHAR(255),

trailer\_url VARCHAR(255),

imdb\_rating DECIMAL(3,1) CHECK (imdb\_rating >= 0 AND imdb\_rating <= 10),

average\_rating DECIMAL(3,1) DEFAULT 0 CHECK (average\_rating >= 0 AND average\_rating <= 10),

total\_ratings INTEGER DEFAULT 0,

featured BOOLEAN DEFAULT FALSE,

is\_original BOOLEAN DEFAULT FALSE,

available\_from DATE,

available\_until DATE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE seasons (

season\_id SERIAL PRIMARY KEY,

content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

season\_number INTEGER NOT NULL CHECK (season\_number > 0),

title VARCHAR(255),

overview TEXT,

release\_date DATE,

poster\_url VARCHAR(255),

episode\_count INTEGER DEFAULT 0,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(content\_id, season\_number)

);

CREATE TABLE episodes (

episode\_id SERIAL PRIMARY KEY,

season\_id INTEGER NOT NULL REFERENCES seasons(season\_id) ON DELETE CASCADE,

episode\_number INTEGER NOT NULL CHECK (episode\_number > 0),

title VARCHAR(255) NOT NULL,

description TEXT,

duration\_minutes INTEGER CHECK (duration\_minutes > 0),

thumbnail\_url VARCHAR(255),

release\_date DATE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(season\_id, episode\_number)

);

CREATE TABLE content\_genres (

content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

genre\_id INTEGER NOT NULL REFERENCES genres(genre\_id) ON DELETE CASCADE,

PRIMARY KEY (content\_id, genre\_id)

);

CREATE TABLE content\_people (

content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

person\_id INTEGER NOT NULL REFERENCES people(person\_id) ON DELETE CASCADE,

role VARCHAR(50) NOT NULL CHECK (role IN ('actor', 'director', 'producer', 'writer', 'cinematographer', 'composer', 'editor')),

character\_name VARCHAR(100), -- Only for actors

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (content\_id, person\_id, role)

);

CREATE TABLE viewing\_history (

history\_id SERIAL PRIMARY KEY,

profile\_id INTEGER NOT NULL REFERENCES profiles(profile\_id) ON DELETE CASCADE,

content\_id INTEGER REFERENCES content(content\_id) ON DELETE SET NULL,

episode\_id INTEGER REFERENCES episodes(episode\_id) ON DELETE SET NULL,

start\_time TIMESTAMP NOT NULL,

end\_time TIMESTAMP,

watch\_duration\_seconds INTEGER,

percentage\_watched DECIMAL(5,2) CHECK (percentage\_watched >= 0 AND percentage\_watched <= 100),

device\_id VARCHAR(100),

ip\_address VARCHAR(45),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

CHECK (content\_id IS NOT NULL OR episode\_id IS NOT NULL) -- Either content or episode must be specified

);

CREATE TABLE watchlist\_items (

watchlist\_id SERIAL PRIMARY KEY,

profile\_id INTEGER NOT NULL REFERENCES profiles(profile\_id) ON DELETE CASCADE,

content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

added\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(profile\_id, content\_id)

);

CREATE TABLE content\_reviews (

review\_id SERIAL PRIMARY KEY,

profile\_id INTEGER NOT NULL REFERENCES profiles(profile\_id) ON DELETE CASCADE,

content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

rating INTEGER CHECK (rating >= 1 AND rating <= 10),

review\_text TEXT,

contains\_spoilers BOOLEAN DEFAULT FALSE,

likes INTEGER DEFAULT 0,

dislikes INTEGER DEFAULT 0,

is\_approved BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(profile\_id, content\_id)

);

CREATE TABLE subscription\_plans (

plan\_id SERIAL PRIMARY KEY,

name VARCHAR(50) NOT NULL UNIQUE,

description TEXT,

price\_monthly DECIMAL(6,2) NOT NULL CHECK (price\_monthly >= 0),

price\_yearly DECIMAL(6,2) CHECK (price\_yearly >= 0),

max\_profiles INTEGER NOT NULL DEFAULT 1,

max\_concurrent\_streams INTEGER NOT NULL DEFAULT 1,

video\_quality VARCHAR(20) CHECK (video\_quality IN ('SD', 'HD', 'FHD', '4K', '8K')),

has\_ads BOOLEAN DEFAULT FALSE,

download\_enabled BOOLEAN DEFAULT FALSE,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE subscriptions (

subscription\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id) ON DELETE CASCADE,

plan\_id INTEGER NOT NULL REFERENCES subscription\_plans(plan\_id) ON DELETE RESTRICT,

billing\_cycle VARCHAR(10) NOT NULL CHECK (billing\_cycle IN ('monthly', 'yearly')),

start\_date DATE NOT NULL,

end\_date DATE,

auto\_renewal BOOLEAN DEFAULT TRUE,

status VARCHAR(20) NOT NULL CHECK (status IN ('active', 'cancelled', 'expired', 'pending', 'suspended')),

cancellation\_date DATE,

cancellation\_reason TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE payment\_transactions (

transaction\_id SERIAL PRIMARY KEY,

subscription\_id INTEGER NOT NULL REFERENCES subscriptions(subscription\_id) ON DELETE CASCADE,

amount DECIMAL(10,2) NOT NULL,

currency VARCHAR(3) DEFAULT 'USD',

payment\_method VARCHAR(50) NOT NULL,

payment\_provider VARCHAR(50) NOT NULL,

transaction\_date TIMESTAMP NOT NULL,

status VARCHAR(20) NOT NULL CHECK (status IN ('pending', 'completed', 'failed', 'refunded', 'chargeback')),

payment\_details VARCHAR(255),

billing\_address TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE devices (

device\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id) ON DELETE CASCADE,

device\_type VARCHAR(50) NOT NULL,

device\_name VARCHAR(100),

device\_model VARCHAR(100),

os\_type VARCHAR(50),

os\_version VARCHAR(50),

app\_version VARCHAR(50),

unique\_device\_id VARCHAR(255) NOT NULL,

is\_trusted BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

last\_used\_at TIMESTAMP,

UNIQUE(user\_id, unique\_device\_id)

);

CREATE TABLE device\_sessions (

session\_id SERIAL PRIMARY KEY,

device\_id INTEGER NOT NULL REFERENCES devices(device\_id) ON DELETE CASCADE,

profile\_id INTEGER REFERENCES profiles(profile\_id) ON DELETE SET NULL,

ip\_address VARCHAR(45),

location VARCHAR(100),

login\_time TIMESTAMP NOT NULL,

logout\_time TIMESTAMP,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE content\_recommendations (

recommendation\_id SERIAL PRIMARY KEY,

source\_content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

recommended\_content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

similarity\_score DECIMAL(4,3) CHECK (similarity\_score >= 0 AND similarity\_score <= 1),

recommendation\_reason VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(source\_content\_id, recommended\_content\_id),

CHECK (source\_content\_id != recommended\_content\_id)

);

CREATE TABLE user\_recommendations (

recommendation\_id SERIAL PRIMARY KEY,

profile\_id INTEGER NOT NULL REFERENCES profiles(profile\_id) ON DELETE CASCADE,

content\_id INTEGER NOT NULL REFERENCES content(content\_id) ON DELETE CASCADE,

score DECIMAL(4,3) CHECK (score >= 0 AND score <= 1),

recommendation\_reason VARCHAR(100),

is\_viewed BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE(profile\_id, content\_id)

);

CREATE INDEX idx\_content\_type ON content(type);

CREATE INDEX idx\_content\_release\_year ON content(release\_year);

CREATE INDEX idx\_content\_average\_rating ON content(average\_rating);

CREATE INDEX idx\_viewing\_history\_profile\_id ON viewing\_history(profile\_id);

CREATE INDEX idx\_viewing\_history\_content\_id ON viewing\_history(content\_id);

CREATE INDEX idx\_viewing\_history\_start\_time ON viewing\_history(start\_time);

CREATE INDEX idx\_watchlist\_profile\_id ON watchlist\_items(profile\_id);

CREATE INDEX idx\_subscriptions\_user\_id ON subscriptions(user\_id);

CREATE INDEX idx\_subscriptions\_status ON subscriptions(status);

CREATE INDEX idx\_payment\_transactions\_subscription\_id ON payment\_transactions(subscription\_id);

CREATE INDEX idx\_payment\_transactions\_status ON payment\_transactions(status);

CREATE INDEX idx\_devices\_user\_id ON devices(user\_id);

CREATE INDEX idx\_user\_recommendations\_profile\_id ON user\_recommendations(profile\_id);

INSERT INTO admins (admin\_id, username, password\_hash, email, first\_name, last\_name, role) VALUES

(101, 'admin1', 'hashed\_password\_1', 'admin1@streamflix.com', 'John', 'Admin', 'super\_admin'),

(102, 'contentmgr1', 'hashed\_password\_2', 'content1@streamflix.com', 'Sarah', 'Content', 'content\_manager'),

(103, 'support1', 'hashed\_password\_3', 'support1@streamflix.com', 'Michael', 'Support', 'user\_support'),

(104, 'analyst1', 'hashed\_password\_4', 'analyst1@streamflix.com', 'Emma', 'Data', 'analytics'),

(105, 'admin2', 'hashed\_password\_5', 'admin2@streamflix.com', 'David', 'System', 'super\_admin'),

(106, 'contentmgr2', 'hashed\_password\_6', 'content2@streamflix.com', 'Lisa', 'Media', 'content\_manager'),

(107, 'support2', 'hashed\_password\_7', 'support2@streamflix.com', 'Robert', 'Help', 'user\_support'),

(108, 'analyst2', 'hashed\_password\_8', 'analyst2@streamflix.com', 'Jennifer', 'Stats', 'analytics'),

(109, 'contentmgr3', 'hashed\_password\_9', 'content3@streamflix.com', 'Thomas', 'Films', 'content\_manager'),

(110, 'support3', 'hashed\_password\_10', 'support3@streamflix.com', 'Jessica', 'Care', 'user\_support');

INSERT INTO users (user\_id, email, password\_hash, first\_name, last\_name, date\_of\_birth, country, phone\_number, is\_verified, is\_active) VALUES

(201, 'user1@example.com', 'hashed\_password\_1', 'Alex', 'Johnson', '1985-03-15', 'United States', '+1-555-123-4567', TRUE, TRUE),

(202, 'user2@example.com', 'hashed\_password\_2', 'Maria', 'Garcia', '1990-07-22', 'Spain', '+34-555-678-9012', TRUE, TRUE),

(203, 'user3@example.com', 'hashed\_password\_3', 'James', 'Smith', '1978-11-30', 'United Kingdom', '+44-555-345-6789', TRUE, TRUE),

(204, 'user4@example.com', 'hashed\_password\_4', 'Sophia', 'Kim', '1995-02-18', 'South Korea', '+82-555-901-2345', TRUE, TRUE),

(205, 'user5@example.com', 'hashed\_password\_5', 'Mohammed', 'Ali', '1982-09-05', 'Egypt', '+20-555-567-8901', TRUE, TRUE),

(206, 'user6@example.com', 'hashed\_password\_6', 'Luisa', 'Fernandez', '1988-12-10', 'Mexico', '+52-555-234-5678', TRUE, TRUE),

(207, 'user7@example.com', 'hashed\_password\_7', 'Hiroshi', 'Tanaka', '1975-06-25', 'Japan', '+81-555-890-1234', TRUE, TRUE),

(208, 'user8@example.com', 'hashed\_password\_8', 'Emma', 'Wilson', '1992-04-03', 'Australia', '+61-555-456-7890', TRUE, TRUE),

(209, 'user9@example.com', 'hashed\_password\_9', 'Raj', 'Patel', '1980-01-12', 'India', '+91-555-123-4567', TRUE, TRUE),

(210, 'user10@example.com', 'hashed\_password\_10', 'Anna', 'Müller', '1993-08-28', 'Germany', '+49-555-678-9012', TRUE, TRUE),

(211, 'user11@example.com', 'hashed\_password\_11', 'Carlos', 'Silva', '1987-05-17', 'Brazil', '+55-555-345-6789', TRUE, TRUE),

(212, 'user12@example.com', 'hashed\_password\_12', 'Olga', 'Ivanova', '1991-10-09', 'Russia', '+7-555-901-2345', TRUE, TRUE),

(213, 'user13@example.com', 'hashed\_password\_13', 'Chen', 'Wei', '1984-07-14', 'China', '+86-555-567-8901', TRUE, TRUE),

(214, 'user14@example.com', 'hashed\_password\_14', 'Fatima', 'Ahmed', '1989-03-22', 'Saudi Arabia', '+966-555-234-5678', TRUE, TRUE),

(215, 'user15@example.com', 'hashed\_password\_15', 'Pierre', 'Dubois', '1979-12-01', 'France', '+33-555-890-1234', TRUE, TRUE);

INSERT INTO profiles (profile\_id, user\_id, profile\_name, avatar\_url, is\_kids\_profile, language\_preference, content\_maturity\_level) VALUES

(301, 201, 'Alex', 'https://streamflix.com/avatars/avatar1.jpg', FALSE, 'en', 'adult'),

(302, 202, 'Alex Jr', 'https://streamflix.com/avatars/kid1.jpg', TRUE, 'en', 'kids'),

(303, 203, 'Maria', 'https://streamflix.com/avatars/avatar2.jpg', FALSE, 'es', 'adult'),

(304, 204, 'Miguel', 'https://streamflix.com/avatars/avatar3.jpg', FALSE, 'es', 'teen'),

(305, 205, 'James', 'https://streamflix.com/avatars/avatar4.jpg', FALSE, 'en', 'adult'),

(306, 206, 'Sarah', 'https://streamflix.com/avatars/avatar5.jpg', FALSE, 'en', 'adult'),

(307, 207, 'Kids', 'https://streamflix.com/avatars/kid2.jpg', TRUE, 'en', 'kids'),

(308, 208, 'Sophia', 'https://streamflix.com/avatars/avatar6.jpg', FALSE, 'ko', 'adult'),

(309, 209, 'Mohammed', 'https://streamflix.com/avatars/avatar7.jpg', FALSE, 'ar', 'adult'),

(310, 210, 'Aisha', 'https://streamflix.com/avatars/avatar8.jpg', FALSE, 'ar', 'teen'),

(311, 211, 'Luisa', 'https://streamflix.com/avatars/avatar9.jpg', FALSE, 'es', 'adult'),

(312, 212, 'Hiroshi', 'https://streamflix.com/avatars/avatar10.jpg', FALSE, 'ja', 'adult'),

(313, 213, 'Yuki', 'https://streamflix.com/avatars/avatar11.jpg', FALSE, 'ja', 'adult'),

(314, 214, 'Emma', 'https://streamflix.com/avatars/avatar12.jpg', FALSE, 'en', 'adult'),

(315, 215, 'Raj', 'https://streamflix.com/avatars/avatar13.jpg', FALSE, 'hi', 'adult'),

(316, 216, 'Priya', 'https://streamflix.com/avatars/avatar14.jpg', FALSE, 'hi', 'adult'),

(317, 217, 'Kids', 'https://streamflix.com/avatars/kid3.jpg', TRUE, 'hi', 'kids'),

(318, 218, 'Anna', 'https://streamflix.com/avatars/avatar15.jpg', FALSE, 'de', 'adult'),

(319, 219, 'Carlos', 'https://streamflix.com/avatars/avatar16.jpg', FALSE, 'pt', 'adult'),

(320, 220, 'Olga', 'https://streamflix.com/avatars/avatar17.jpg', FALSE, 'ru', 'adult'),

(321, 221, 'Chen', 'https://streamflix.com/avatars/avatar18.jpg', FALSE, 'zh', 'adult'),

(322, 222, 'Fatima', 'https://streamflix.com/avatars/avatar19.jpg', FALSE, 'ar', 'adult'),

(323, 223, 'Pierre', 'https://streamflix.com/avatars/avatar20.jpg', FALSE, 'fr', 'adult'),

(324, 224, 'Marie', 'https://streamflix.com/avatars/avatar21.jpg', FALSE, 'fr', 'adult'),

(325, 225, 'Enfants', 'https://streamflix.com/avatars/kid4.jpg', TRUE, 'fr', 'kids');

INSERT INTO people (person\_id, full\_name, birth\_date, biography, nationality, profile\_image\_url, imdb\_id) VALUES

(401, 'Tom Hanks', '1956-07-09', 'Thomas Jeffrey Hanks is an American actor and filmmaker.', 'American', 'https://streamflix.com/people/tomhanks.jpg', 'nm0000158'),

(402, 'Meryl Streep', '1949-06-22', 'Mary Louise "Meryl" Streep is an American actress.', 'American', 'https://streamflix.com/people/merylstreep.jpg', 'nm0000658'),

(403, 'Leonardo DiCaprio', '1974-11-11', 'Leonardo Wilhelm DiCaprio is an American actor and film producer.', 'American', 'https://streamflix.com/people/leonardodicaprio.jpg', 'nm0000138'),

(404, 'Christopher Nolan', '1970-07-30', 'Christopher Edward Nolan is a British-American film director, producer, and screenwriter.', 'British-American', 'https://streamflix.com/people/christophernolan.jpg', 'nm0634240'),

(405, 'Scarlett Johansson', '1984-11-22', 'Scarlett Ingrid Johansson is an American actress and singer.', 'American', 'https://streamflix.com/people/scarlettjohansson.jpg', 'nm0424060'),

(406, 'Steven Spielberg', '1946-12-18', 'Steven Allan Spielberg is an American filmmaker.', 'American', 'https://streamflix.com/people/stevenspielberg.jpg', 'nm0000229'),

(407, 'Emma Stone', '1988-11-06', 'Emily Jean "Emma" Stone is an American actress.', 'American', 'https://streamflix.com/people/emmastone.jpg', 'nm1297015'),

(408, 'Denzel Washington', '1954-12-28', 'Denzel Hayes Washington Jr. is an American actor, director, and producer.', 'American', 'https://streamflix.com/people/denzelwashington.jpg', 'nm0000243'),

(409, 'Viola Davis', '1965-08-11', 'Viola Davis is an American actress and producer.', 'American', 'https://streamflix.com/people/violadavis.jpg', 'nm0205626'),

(410, 'Ryan Coogler', '1986-05-23', 'Ryan Kyle Coogler is an American film director, producer, and screenwriter.', 'American', 'https://streamflix.com/people/ryancoogler.jpg', 'nm3363032'),

(411, 'Chadwick Boseman', '1976-11-29', 'Chadwick Aaron Boseman was an American actor and playwright.', 'American', 'https://streamflix.com/people/chadwickboseman.jpg', 'nm1569276'),

(412, 'Bong Joon-ho', '1969-09-14', 'Bong Joon-ho is a South Korean film director, producer and screenwriter.', 'South Korean', 'https://streamflix.com/people/bongjoonho.jpg', 'nm0094435');

INSERT INTO genres (name, description) VALUES

('Action', 'Action films usually include high energy, big-budget physical stunts and chases, possibly with rescues, battles, fights, escapes, destructive crises, etc.'),

('Adventure', 'Adventure films are usually exciting stories, with new experiences or exotic locales, very similar to or often paired with the action film genre.'),

('Comedy', 'Comedy is a story that tells about a series of funny, or comical events, intended to make the audience laugh.'),

('Drama', 'Drama is a category of narrative fiction (or semi-fiction) intended to be more serious than humorous in tone.'),

('Horror', 'Horror is a genre of speculative fiction which is intended to frighten, scare, disgust, or startle its readers by inducing feelings of horror and terror.'),

('Science Fiction', 'Science fiction is a genre of speculative fiction that typically deals with imaginative and futuristic concepts such as advanced science and technology, space exploration, time travel, parallel universes, and extraterrestrial life.'),

('Fantasy', 'Fantasy is a genre of speculative fiction set in a fictional universe, often inspired by real world myth and folklore.'),

('Romance', 'Romance films or romance movies are romantic love stories recorded in visual media for broadcast in theaters and on TV that focus on passion, emotion, and the affectionate romantic involvement of the main characters.'),

('Thriller', 'Thriller is a genre of fiction, having numerous, often overlapping subgenres. Thrillers are characterized and defined by the moods they elicit, giving viewers heightened feelings of suspense, excitement, surprise, anticipation and anxiety.'),

('Mystery', 'Mystery is a fiction genre where the nature of an event, usually a murder or other crime, remains mysterious until the end of the story.'),

('Documentary', 'A documentary film is a non-fictional motion-picture intended to "document reality, primarily for the purposes of instruction, education, or maintaining a historical record".'),

('Animation', 'Animation is a method in which pictures are manipulated to appear as moving images.'),

('Family', 'Family film is a genre that contains appropriate content for younger viewers.'),

('Biography', 'A biographical film, or biopic, is a film that dramatizes the life of a non-fictional or historically-based person or people.'),

('History', 'A historical film is a fiction film showing past events or set within a historical period.');

INSERT INTO content (content\_id, title, type, description, release\_year, age\_rating, duration\_minutes, poster\_url, backdrop\_url, trailer\_url, imdb\_rating, average\_rating, total\_ratings, featured, is\_original, available\_from) VALUES

(101, 'The Shawshank Redemption', 'movie', 'Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.', 1994, 'R', 142, 'https://streamflix.com/posters/shawshank.jpg', 'https://streamflix.com/backdrops/shawshank.jpg', 'https://streamflix.com/trailers/shawshank.mp4', 9.3, 9.2, 1250, TRUE, FALSE, '2020-01-01'),

(102, 'Inception', 'movie', '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.', 2010, 'PG-13', 148, 'https://streamflix.com/posters/inception.jpg', 'https://streamflix.com/backdrops/inception.jpg', 'https://streamflix.com/trailers/inception.mp4', 8.8, 8.7, 980, TRUE, FALSE, '2020-01-01'),

(103, 'The Dark Knight', 'movie', '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.', 2008, 'PG-13', 152, 'https://streamflix.com/posters/darkknight.jpg', 'https://streamflix.com/backdrops/darkknight.jpg', 'https://streamflix.com/trailers/darkknight.mp4', 9.0, 8.9, 1100, TRUE, FALSE, '2020-01-01'),

(104, 'Pulp Fiction', 'movie', '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.', 1994, 'R', 154, 'https://streamflix.com/posters/pulpfiction.jpg', 'https://streamflix.com/backdrops/pulpfiction.jpg', 'https://streamflix.com/trailers/pulpfiction.mp4', 8.9, 8.8, 950, FALSE, FALSE, '2020-01-01'),

(105, 'The Matrix', 'movie', 'A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.', 1999, 'R', 136, 'https://streamflix.com/posters/matrix.jpg', 'https://streamflix.com/backdrops/matrix.jpg', 'https://streamflix.com/trailers/matrix.mp4', 8.7, 8.6, 890, FALSE, FALSE, '2020-01-01'),

(106, 'Parasite', 'movie', 'Greed and class discrimination threaten the newly formed symbiotic relationship between the wealthy Park family and the destitute Kim clan.', 2019, 'R', 132, 'https://streamflix.com/posters/parasite.jpg', 'https://streamflix.com/backdrops/parasite.jpg', 'https://streamflix.com/trailers/parasite.mp4', 8.6, 8.5, 780, TRUE, FALSE, '2020-06-01'),

(107, 'Stranger Things', 'tvshow', 'When a young boy disappears, his mother, a police chief, and his friends must confront terrifying supernatural forces in order to get him back.', 2016, 'TV-14', NULL, 'https://streamflix.com/posters/strangerthings.jpg', 'https://streamflix.com/backdrops/strangerthings.jpg', 'https://streamflix.com/trailers/strangerthings.mp4', 8.7, 8.6, 1050, TRUE, TRUE, '2016-07-15'),

(108, 'Breaking Bad', 'tvshow', 'A high school chemistry teacher diagnosed with inoperable lung cancer turns to manufacturing and selling methamphetamine in order to secure his family\'s future.', 2008, 'TV-MA', NULL, 'https://streamflix.com/posters/breakingbad.jpg', 'https://streamflix.com/backdrops/breakingbad.jpg', 'https://streamflix.com/trailers/breakingbad.mp4', 9.5, 9.4, 1200, TRUE, FALSE, '2020-01-01'),

(109, 'The Crown', 'tvshow', 'Follows the political rivalries and romance of Queen Elizabeth II\'s reign and the events that shaped the second half of the twentieth century.', 2016, 'TV-MA', NULL, 'https://streamflix.com/posters/crown.jpg', 'https://streamflix.com/backdrops/crown.jpg', 'https://streamflix.com/trailers/crown.mp4', 8.7, 8.6, 820, TRUE, TRUE, '2016-11-04'),

(110, 'Black Mirror', 'tvshow', 'An anthology series exploring a twisted, high-tech multiverse where humanity\'s greatest innovations and darkest instincts collide.', 2011, 'TV-MA', NULL, 'https://streamflix.com/posters/blackmirror.jpg', 'https://streamflix.com/backdrops/blackmirror.jpg', 'https://streamflix.com/trailers/blackmirror.mp4', 8.8, 8.7, 950, FALSE, TRUE, '2020-01-01'),

(111, 'Little Women', 'movie', 'Jo March reflects back and forth on her life, telling the beloved story of the March sisters - four young women, each determined to live life on her own terms.', 2019, 'PG', 135, 'https://streamflix.com/posters/littlewomen.jpg', 'https://streamflix.com/backdrops/littlewomen.jpg', 'https://streamflix.com/trailers/littlewomen.mp4', 7.8, 7.9, 650, FALSE, FALSE, '2020-04-01'),

(112, 'The Mandalorian', 'tvshow', 'The travels of a lone bounty hunter in the outer reaches of the galaxy, far from the authority of the New Republic.', 2019, 'TV-14', NULL, 'https://streamflix.com/posters/mandalorian.jpg', 'https://streamflix.com/backdrops/mandalorian.jpg', 'https://streamflix.com/trailers/mandalorian.mp4', 8.8, 8.7, 890, TRUE, FALSE, '2020-03-01'),

(113, 'Black Panther', 'movie', 'T\'Challa, heir to the hidden but advanced kingdom of Wakanda, must step forward to lead his people into a new future and must confront a challenger from his country\'s past.', 2018, 'PG-13', 134, 'https://streamflix.com/posters/blackpanther.jpg', 'https://streamflix.com/backdrops/blackpanther.jpg', 'https://streamflix.com/trailers/blackpanther.mp4', 7.3, 7.5, 780, TRUE, FALSE, '2020-01-01'),

(114, 'The Queen\'s Gambit', 'tvshow', 'Orphaned at the tender age of nine, prodigious introvert Beth Harmon discovers and masters the game of chess in 1960s USA. But child stardom comes at a price.', 2020, 'TV-MA', NULL, 'https://streamflix.com/posters/queensgambit.jpg', 'https://streamflix.com/backdrops/queensgambit.jpg', 'https://streamflix.com/trailers/queensgambit.mp4', 8.6, 8.7, 720, TRUE, TRUE, '2020-10-23'),

(115, 'Interstellar', 'movie', 'A team of explorers travel through a wormhole in space in an attempt to ensure humanity\'s survival.', 2014, 'PG-13', 169, 'https://streamflix.com/posters/interstellar.jpg', 'https://streamflix.com/backdrops/interstellar.jpg', 'https://streamflix.com/trailers/interstellar.mp4', 8.6, 8.5, 910, FALSE, FALSE, '2020-01-01');

INSERT INTO seasons (season\_id, content\_id, season\_number, title, overview, release\_date, poster\_url, episode\_count) VALUES

(101, 107, 1, 'Stranger Things: Season 1', 'When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces and one strange little girl.', '2016-07-15', 'https://streamflix.com/seasons/strangerthings\_s1.jpg', 8),

(102, 107, 2, 'Stranger Things: Season 2', 'It\'s been nearly a year since Will\'s strange disappearance. But life\'s hardly back to normal in Hawkins. Not even close.', '2017-10-27', 'https://streamflix.com/seasons/strangerthings\_s2.jpg', 9),

(103, 107, 3, 'Stranger Things: Season 3', 'Summer brings new jobs and budding romance. But the mood shifts when Dustin\'s radio picks up a Russian broadcast, and Will senses something is wrong.', '2019-07-04', 'https://streamflix.com/seasons/strangerthings\_s3.jpg', 8),

(104, 108, 1, 'Breaking Bad: Season 1', 'A high school chemistry teacher diagnosed with inoperable lung cancer turns to manufacturing and selling methamphetamine in order to secure his family\'s future.', '2008-01-20', 'https://streamflix.com/seasons/breakingbad\_s1.jpg', 7),

(105, 108, 2, 'Breaking Bad: Season 2', 'Walt and Jesse face the deadly consequences of their actions, both personally and professionally, as they try to cope with the mess they created.', '2009-03-08', 'https://streamflix.com/seasons/breakingbad\_s2.jpg', 13),

(106, 108, 3, 'Breaking Bad: Season 3', 'Walt continues to battle dueling identities: a desperate husband and father trying to provide for his family, and a newly appointed key player in the Albuquerque drug trade.', '2010-03-21', 'https://streamflix.com/seasons/breakingbad\_s3.jpg', 13),

(107, 108, 4, 'Breaking Bad: Season 4', 'Walt and Jesse face the deadly consequences of their actions, both personally and professionally, as they try to cope with the mess they created.', '2011-07-17', 'https://streamflix.com/seasons/breakingbad\_s4.jpg', 13),

(108, 108, 5, 'Breaking Bad: Season 5', 'Walt\'s transformation from well-meaning family man to ruthless drug kingpin is complete.', '2012-07-15', 'https://streamflix.com/seasons/breakingbad\_s5.jpg', 16),

(109, 109, 1, 'The Crown: Season 1', 'As a young Elizabeth becomes Queen, she must manage major political issues and personal matters, which often clash in ways she never imagined.', '2016-11-04', 'https://streamflix.com/seasons/crown\_s1.jpg', 10),

(110, 109, 2, 'The Crown: Season 2', 'As a new era begins, Queen Elizabeth struggles to navigate a world that\'s changing around her while preserving both the monarchy and her marriage.', '2017-12-08', 'https://streamflix.com/seasons/crown\_s2.jpg', 10),

(111, 109, 3, 'The Crown: Season 3', 'As the 1960s and \'70s unfold, the royal family contends with conflict and betrayal while striving to uphold tradition in the face of an evolving world.', '2019-11-17', 'https://streamflix.com/seasons/crown\_s3.jpg', 10),

(112, 110, 1, 'Black Mirror: Season 1', 'This sci-fi anthology series explores a twisted, high-tech near-future where humanity\'s greatest innovations and darkest instincts collide.', '2011-12-04', 'https://streamflix.com/seasons/blackmirror\_s1.jpg', 3),

(113, 110, 2, 'Black Mirror: Season 2', 'This sci-fi anthology series explores a twisted, high-tech near-future where humanity\'s greatest innovations and darkest instincts collide.', '2013-02-11', 'https://streamflix.com/seasons/blackmirror\_s2.jpg', 3),

(114, 110, 3, 'Black Mirror: Season 3', 'This sci-fi anthology series explores a twisted, high-tech near-future where humanity\'s greatest innovations and darkest instincts collide.', '2016-10-21', 'https://streamflix.com/seasons/blackmirror\_s3.jpg', 6),

(115, 110, 4, 'Black Mirror: Season 4', 'This sci-fi anthology series explores a twisted, high-tech near-future where humanity\'s greatest innovations and darkest instincts collide.', '2017-12-29', 'https://streamflix.com/seasons/blackmirror\_s4.jpg', 6),

(116, 112, 1, 'The Mandalorian: Season 1', 'After the fall of the Empire, a lone gunfighter makes his way through the lawless galaxy with his foundling.', '2019-11-12', 'https://streamflix.com/seasons/mandalorian\_s1.jpg', 8),

(117, 112, 2, 'The Mandalorian: Season 2', 'The Mandalorian and the Child continue their journey, facing enemies and rallying allies as they make their way through a dangerous galaxy.', '2020-10-30', 'https://streamflix.com/seasons/mandalorian\_s2.jpg', 8),

(118, 114, 1, 'The Queen\'s Gambit', 'In a 1950s orphanage, a young girl reveals an astonishing talent for chess and begins an unlikely journey to stardom while grappling with addiction.', '2020-10-23', 'https://streamflix.com/seasons/queensgambit\_s1.jpg', 7);

INSERT INTO episodes (episode\_id, season\_id, episode\_number, title, description, duration\_minutes, thumbnail\_url, release\_date) VALUES

(101, 101, 1, 'Chapter One: The Vanishing of Will Byers', 'On his way home from a friend\'s house, young Will sees something terrifying. Nearby, a sinister secret lurks in the depths of a government lab.', 49, 'https://streamflix.com/episodes/st\_s1e1.jpg', '2016-07-15'),

(102, 101, 2, 'Chapter Two: The Weirdo on Maple Street', 'Lucas, Mike and Dustin try to talk to the girl they found in the woods. Hopper questions an anxious Joyce about an unsettling phone call.', 46, 'https://streamflix.com/episodes/st\_s1e2.jpg', '2016-07-15'),

(103, 101, 3, 'Chapter Three: Holly, Jolly', 'An increasingly concerned Nancy looks for Barb and finds out what Jonathan\'s been up to. Joyce is convinced Will is trying to talk to her.', 52, 'https://streamflix.com/episodes/st\_s1e3.jpg', '2016-07-15'),

(104, 102, 1, 'Chapter One: MADMAX', 'As the town preps for Halloween, a high-scoring rival shakes things up at the arcade, and a skeptical Hopper inspects a field of rotting pumpkins.', 48, 'https://streamflix.com/episodes/st\_s2e1.jpg', '2017-10-27'),

(105, 102, 2, 'Chapter Two: Trick or Treat, Freak', 'After Will sees something terrible on trick-or-treat night, Mike wonders whether Eleven\'s still out there. Nancy wrestles with the truth about Barb.', 56, 'https://streamflix.com/episodes/st\_s2e2.jpg', '2017-10-27'),

(106, 104, 1, 'Pilot', 'Diagnosed with terminal lung cancer, chemistry teacher Walter White teams up with former student Jesse Pinkman to cook and sell crystal meth.', 58, 'https://streamflix.com/episodes/bb\_s1e1.jpg', '2008-01-20'),

(107, 104, 2, 'Cat\'s in the Bag...', 'After their first drug deal goes terribly wrong, Walt and Jesse are forced to deal with a corpse and a prisoner. Meanwhile, Skyler grows suspicious.', 48, 'https://streamflix.com/episodes/bb\_s1e2.jpg', '2008-01-27'),

(108, 109, 1, 'Wolferton Splash', 'A young Princess Elizabeth marries Prince Philip. As King George VI\'s health worsens, Winston Churchill is elected prime minister for the second time.', 57, 'https://streamflix.com/episodes/crown\_s1e1.jpg', '2016-11-04'),

(109, 109, 2, 'Hyde Park Corner', 'With King George too ill to travel, Elizabeth and Philip embark on a four-continent Commonwealth tour. Party leaders attempt to undermine Churchill.', 61, 'https://streamflix.com/episodes/crown\_s1e2.jpg', '2016-11-04'),

(110, 112, 1, 'The National Anthem', 'Prime Minister Michael Callow faces a shocking dilemma when Princess Susannah, a much-loved member of the Royal Family, is kidnapped.', 44, 'https://streamflix.com/episodes/bm\_s1e1.jpg', '2011-12-04'),

(111, 118, 1, 'Chapter 1', 'In a Kentucky orphanage in the 1950s, a young girl discovers an astonishing talent for chess while struggling with addiction.', 60, 'https://streamflix.com/episodes/qg\_s1e1.jpg', '2020-10-23'),

(112, 118, 2, 'Chapter 2', 'Sent to an orphanage at age 9, Beth develops an uncanny knack for chess and a growing dependence on the green tranquilizers given to the children.', 60, 'https://streamflix.com/episodes/qg\_s1e2.jpg', '2020-10-23'),

(113, 116, 1, 'Chapter One: The Mandalorian', 'A Mandalorian bounty hunter tracks a target for a well-paying client.', 39, 'https://streamflix.com/episodes/mando\_s1e1.jpg', '2019-11-12'),

(114, 116, 2, 'Chapter Two: The Child', 'Target in-hand, The Mandalorian must now contend with scavengers.', 32, 'https://streamflix.com/episodes/mando\_s1e2.jpg', '2019-11-15');

INSERT INTO content\_genres (content\_id, genre\_id) VALUES

(101, 4), -- Shawshank Redemption - Drama

(101, 3), -- Shawshank Redemption - Crime

(102, 6), (102, 9), (102, 1), -- Inception - Sci-Fi, Thriller, Action

(103, 1), (103, 4), (103, 9), (103, 3), -- Dark Knight - Action, Drama, Thriller, Crime

(104, 3), (104, 4), (104, 9), -- Pulp Fiction - Crime, Drama, Thriller

(105, 1), (105, 6), (105, 9), -- Matrix - Action, Sci-Fi, Thriller

(106, 4), (106, 9), (106, 3), -- Parasite - Drama, Thriller, Crime

(107, 4), (107, 5), (107, 6), (107, 2), -- Stranger Things - Drama, Horror, Sci-Fi, Adventure

(108, 4), (108, 9), (108, 3), -- Breaking Bad - Drama, Thriller, Crime

(109, 4), (109, 14), (109, 15), -- The Crown - Drama, Biography, History

(110, 4), (110, 6), (110, 9), (110, 5), -- Black Mirror - Drama, Sci-Fi, Thriller, Horror

(111, 4), (111, 8), (111, 15), -- Little Women - Drama, Romance, Historical

(112, 1), (112, 2), (112, 6), (112, 7), -- Mandalorian - Action, Adventure, Sci-Fi, Fantasy

(113, 1), (113, 2), (113, 6), (113, 7), -- Black Panther - Action, Adventure, Sci-Fi, Fantasy

(114, 4), (114, 14), -- Queen's Gambit - Drama, Biography

(115, 2), (115, 4), (115, 6), (115, 7); -- Interstellar - Adventure, Drama, Sci-Fi, Fantasy

INSERT INTO content\_people (content\_id, person\_id, role, character\_name) VALUES

(101, 408, 'actor', 'Ellis Boyd "Red" Redding'),

(102, 403, 'actor', 'Dom Cobb'),

(102, 404, 'director', NULL),

(103, 404, 'director', NULL),

(106, 412, 'director', NULL),

(111, 413, 'director', NULL),

(111, 414, 'actor', 'Jo March'),

(111, 415, 'actor', 'Theodore "Laurie" Laurence'),

(113, 410, 'director', NULL),

(113, 411, 'actor', 'T\'Challa / Black Panther'),

(115, 404, 'director', NULL),

(115, 403, 'actor', 'Joseph Cooper');

INSERT INTO subscription\_plans (name, description, price\_monthly, price\_yearly, max\_profiles, max\_concurrent\_streams, video\_quality, has\_ads, download\_enabled) VALUES

('Basic', 'Basic plan with limited features', 8.99, 89.99, 1, 1, 'SD', TRUE, FALSE),

('Standard', 'Standard plan with HD quality', 13.99, 139.99, 2, 2, 'HD', FALSE, TRUE),

('Premium', 'Premium plan with Ultra HD quality', 17.99, 179.99, 4, 4, '4K', FALSE, TRUE),

('Family', 'Family plan with all premium features', 19.99, 199.99, 6, 4, '4K', FALSE, TRUE),

('Student', 'Discounted plan for students', 4.99, 49.99, 1, 1, 'HD', TRUE, TRUE);

INSERT INTO subscriptions (user\_id, plan\_id, billing\_cycle, start\_date, end\_date, auto\_renewal, status) VALUES

(201, 3, 'monthly', '2020-01-15', '2021-01-15', TRUE, 'active'),

(202, 2, 'yearly', '2020-03-10', '2021-03-10', TRUE, 'active'),

(203, 4, 'monthly', '2020-02-05', '2021-02-05', TRUE, 'active'),

(204, 3, 'monthly', '2020-04-20', '2021-04-20', TRUE, 'active'),

(205, 2, 'yearly', '2020-01-30', '2021-01-30', TRUE, 'active'),

(206, 1, 'monthly', '2020-05-12', '2020-12-12', FALSE, 'cancelled'),

(207, 3, 'yearly', '2020-06-01', '2021-06-01', TRUE, 'active'),

(208, 2, 'monthly', '2020-03-15', '2021-03-15', TRUE, 'active'),

(209, 4, 'yearly', '2020-02-28', '2021-02-28', TRUE, 'active'),

(210, 3, 'monthly', '2020-07-10', '2021-07-10', TRUE, 'active'),

(211, 1, 'monthly', '2020-08-05', '2020-11-05', FALSE, 'expired'),

(212, 2, 'yearly', '2020-04-18', '2021-04-18', TRUE, 'active'),

(213, 3, 'monthly', '2020-09-22', '2021-09-22', TRUE, 'active'),

(214, 2, 'monthly', '2020-10-15', '2021-10-15', TRUE, 'active'),

(215, 4, 'yearly', '2020-05-30', '2021-05-30', TRUE, 'active');

INSERT INTO payment\_transactions (subscription\_id, amount, currency, payment\_method, payment\_provider, transaction\_date, status, payment\_details) VALUES

(1, 17.99, 'USD', 'credit\_card', 'Stripe', '2020-01-15', 'completed', '{"card\_last4":"4242", "card\_brand":"Visa"}'),

(1, 17.99, 'USD', 'credit\_card', 'Stripe', '2020-02-15', 'completed', '{"card\_last4":"4242", "card\_brand":"Visa"}'),

(1, 17.99, 'USD', 'credit\_card', 'Stripe', '2020-03-15', 'completed', '{"card\_last4":"4242", "card\_brand":"Visa"}'),

(2, 139.99, 'USD', 'paypal', 'PayPal', '2020-03-10', 'completed', '{"paypal\_email":"maria@example.com"}'),

(3, 19.99, 'USD', 'credit\_card', 'Stripe', '2020-02-05', 'completed', '{"card\_last4":"5678", "card\_brand":"Mastercard"}'),

(3, 19.99, 'USD', 'credit\_card', 'Stripe', '2020-03-05', 'completed', '{"card\_last4":"5678", "card\_brand":"Mastercard"}'),

(3, 19.99, 'USD', 'credit\_card', 'Stripe', '2020-04-05', 'failed', '{"card\_last4":"5678", "card\_brand":"Mastercard", "error":"Insufficient funds"}'),

(3, 19.99, 'USD', 'credit\_card', 'Stripe', '2020-04-07', 'completed', '{"card\_last4":"5678", "card\_brand":"Mastercard"}'),

(4, 17.99, 'USD', 'apple\_pay', 'Apple', '2020-04-20', 'completed', '{"device":"iPhone 11"}'),

(5, 139.99, 'USD', 'google\_pay', 'Google', '2020-01-30', 'completed', '{"device":"Pixel 4"}'),

(6, 8.99, 'USD', 'credit\_card', 'Stripe', '2020-05-12', 'completed', '{"card\_last4":"9012", "card\_brand":"Discover"}'),

(6, 8.99, 'USD', 'credit\_card', 'Stripe', '2020-06-12', 'completed', '{"card\_last4":"9012", "card\_brand":"Discover"}'),

(7, 179.99, 'USD', 'paypal', 'PayPal', '2020-06-01', 'completed', '{"paypal\_email":"hiroshi@example.com"}'),

(8, 13.99, 'USD', 'credit\_card', 'Stripe', '2020-03-15', 'completed', '{"card\_last4":"3456", "card\_brand":"Amex"}'),

(9, 199.99, 'USD', 'bank\_transfer', 'ACH', '2020-02-28', 'completed', '{"bank\_account":"\*\*\*\*\*6789"}');

INSERT INTO devices (user\_id, device\_type, device\_name, device\_model, os\_type, os\_version, app\_version, unique\_device\_id, is\_trusted) VALUES

(201, 'smartphone', 'Alex\'s iPhone', 'iPhone 12 Pro', 'iOS', '14.2', '2.1.0', 'AABBCCDDEEFF001122', TRUE),

(201, 'smart\_tv', 'Living Room TV', 'Samsung QLED', 'Tizen', '5.0', '3.0.1', 'TVSAM123456789', TRUE),

(201, 'tablet', 'iPad', 'iPad Pro', 'iOS', '14.1', '2.1.0', 'AABBCCDDEEFF334455', TRUE),

(202, 'smartphone', 'Maria\'s Phone', 'Samsung Galaxy S20', 'Android', '11', '2.1.0', 'ANDROID987654321', TRUE),

(202, 'laptop', 'Maria\'s Laptop', 'MacBook Pro', 'macOS', '11.0', '2.0.5', 'MACBOOK123456789', TRUE),

(203, 'smart\_tv', 'Master Bedroom TV', 'LG OLED', 'webOS', '5.0', '3.0.1', 'TVLG987654321', TRUE),

(203, 'streaming\_stick', 'Living Room Roku', 'Roku Ultra', 'Roku OS', '9.4', '2.1.0', 'ROKU123456789', TRUE),

(204, 'smartphone', 'Sophia\'s Phone', 'iPhone 11', 'iOS', '14.0', '2.1.0', 'AABBCCDDEEFF667788', TRUE),

(205, 'game\_console', 'PlayStation', 'PlayStation 5', 'PlayStation OS', '5.0', '1.0.2', 'PS5123456789', TRUE),

(206, 'smartphone', 'Luisa\'s Phone', 'Xiaomi Mi 10', 'Android', '10', '2.0.8', 'ANDROID123456789', TRUE),

(207, 'smart\_tv', 'Sony TV', 'Sony Bravia', 'Android TV', '9.0', '2.1.0', 'TVSONY123456789', TRUE),

(208, 'tablet', 'Emma\'s iPad', 'iPad Air', 'iOS', '14.0', '2.1.0', 'AABBCCDDEEFF998877', TRUE),

(209, 'laptop', 'Raj\'s Laptop', 'Dell XPS', 'Windows', '10', '2.0.5', 'WINDOWS123456789', TRUE),

(210, 'smartphone', 'Anna\'s Phone', 'Google Pixel 5', 'Android', '11', '2.1.0', 'ANDROID567891234', TRUE),

(211, 'smart\_tv', 'Carlos\'s TV', 'Samsung QLED', 'Tizen', '4.0', '3.0.0', 'TVSAM567891234', TRUE),

(212, 'laptop', 'Olga\'s Laptop', 'Lenovo ThinkPad', 'Windows', '10', '2.0.5', 'WINDOWS567891234', TRUE),

(213, 'smartphone', 'Chen\'s Phone', 'Huawei P40', 'Android', '10', '2.0.8', 'ANDROID234567891', TRUE),

(214, 'tablet', 'Fatima\'s Tablet', 'Samsung Galaxy Tab S7', 'Android', '10', '2.0.8', 'ANDROID345678912', TRUE),

(215, 'smart\_tv', 'Pierre\'s TV', 'LG OLED', 'webOS', '5.0', '3.0.1', 'TVLG234567891', TRUE);

INSERT INTO device\_sessions (device\_id, profile\_id, ip\_address, location, login\_time, logout\_time, is\_active) VALUES

(101, 301, '192.168.1.1', 'New York, USA', '2020-10-01 19:30:00', '2020-10-01 21:45:00', FALSE),

(101, 301, '192.168.1.1', 'New York, USA', '2020-10-02 20:00:00', '2020-10-02 22:30:00', FALSE),

(102, 301, '192.168.1.1', 'New York, USA', '2020-10-03 18:15:00', '2020-10-03 20:45:00', FALSE),

(103, 302, '192.168.1.1', 'New York, USA', '2020-10-04 16:00:00', '2020-10-04 17:30:00', FALSE),

(104, 303, '192.168.2.1', 'Madrid, Spain', '2020-10-01 14:00:00', '2020-10-01 16:30:00', FALSE),

(105, 303, '192.168.2.1', 'Madrid, Spain', '2020-10-02 21:00:00', '2020-10-02 23:00:00', FALSE),

(106, 305, '192.168.3.1', 'London, UK', '2020-10-01 20:00:00', '2020-10-01 22:00:00', FALSE),

(107, 306, '192.168.3.1', 'London, UK', '2020-10-02 19:00:00', '2020-10-02 21:30:00', FALSE),

(108, 308, '192.168.4.1', 'Seoul, South Korea', '2020-10-01 18:00:00', '2020-10-01 20:00:00', FALSE),

(109, 309, '192.168.5.1', 'Cairo, Egypt', '2020-10-01 21:00:00', '2020-10-01 23:00:00', FALSE),

(110, 311, '192.168.6.1', 'Mexico City, Mexico', '2020-10-01 19:00:00', '2020-10-01 21:00:00', FALSE),

(111, 312, '192.168.7.1', 'Tokyo, Japan', '2020-10-01 20:00:00', '2020-10-01 22:00:00', FALSE),

(112, 314, '192.168.8.1', 'Sydney, Australia', '2020-10-01 19:30:00', '2020-10-01 21:30:00', FALSE),

(113, 315, '192.168.9.1', 'Mumbai, India', '2020-10-01 18:30:00', '2020-10-01 20:30:00', FALSE),

(114, 318, '192.168.10.1', 'Berlin, Germany', '2020-10-01 20:30:00', '2020-10-01 22:30:00', FALSE),

(115, 320, '192.168.11.1', 'São Paulo, Brazil', '2020-10-01 19:00:00', '2020-10-01 21:00:00', FALSE);

INSERT INTO viewing\_history (profile\_id, content\_id, episode\_id, start\_time, end\_time, watch\_duration\_seconds, percentage\_watched, device\_id, ip\_address) VALUES

(301, 101, NULL, '2020-10-01 19:30:00', '2020-10-01 21:52:00', 8520, 100.0, 'AABBCCDDEEFF001122', '192.168.1.1'),

(301, 102, NULL, '2020-10-02 20:00:00', '2020-10-02 22:28:00', 8880, 100.0, 'AABBCCDDEEFF001122', '192.168.1.1'),

(301, NULL, 101, '2020-10-03 18:15:00', '2020-10-03 19:04:00', 2940, 100.0, 'TVSAM123456789', '192.168.1.1'),

(301, NULL, 102, '2020-10-03 19:05:00', '2020-10-03 19:51:00', 2760, 100.0, 'TVSAM123456789', '192.168.1.1'),

(302, 113, NULL, '2020-10-04 16:00:00', '2020-10-04 17:30:00', 5400, 67.2, 'AABBCCDDEEFF334455', '192.168.1.1'),

(303, 106, NULL, '2020-10-01 14:00:00', '2020-10-01 16:12:00', 7920, 100.0, 'ANDROID987654321', '192.168.2.1'),

(303, NULL, 106, '2020-10-02 21:00:00', '2020-10-02 21:58:00', 3480, 100.0, 'MACBOOK123456789', '192.168.2.1'),

(305, 103, NULL, '2020-10-01 20:00:00', '2020-10-01 22:32:00', 9120, 100.0, 'TVLG987654321', '192.168.3.1'),

(306, NULL, 109, '2020-10-02 19:00:00', '2020-10-02 20:01:00', 3660, 100.0, 'ROKU123456789', '192.168.3.1'),

(308, 115, NULL, '2020-10-01 18:00:00', '2020-10-01 20:49:00', 10140, 100.0, 'AABBCCDDEEFF667788', '192.168.4.1'),

(309, 104, NULL, '2020-10-01 21:00:00', '2020-10-01 23:34:00', 9240, 100.0, 'PS5123456789', '192.168.5.1'),

(311, NULL, 113, '2020-10-01 19:00:00', '2020-10-01 19:39:00', 2340, 100.0, 'ANDROID123456789', '192.168.6.1'),

(312, NULL, 114, '2020-10-01 20:00:00', '2020-10-01 20:32:00', 1920, 100.0, 'TVSONY123456789', '192.168.7.1'),

(314, 111, NULL, '2020-10-01 19:30:00', '2020-10-01 21:45:00', 8100, 100.0, 'AABBCCDDEEFF998877', '192.168.8.1'),

(315, 105, NULL, '2020-10-01 18:30:00', '2020-10-01 20:46:00', 8160, 100.0, 'WINDOWS123456789', '192.168.9.1');

INSERT INTO watchlist\_items (profile\_id, content\_id) VALUES

(301, 103),

(301, 106),

(301, 113),

(302, 107),

(303, 108),

(303, 110),

(305, 104),

(305, 115),

(306, 109),

(308, 102),

(309, 107),

(311, 112),

(312, 114),

(314, 105),

(315, 101);

INSERT INTO content\_reviews (profile\_id, content\_id, rating, review\_text, contains\_spoilers) VALUES

(301, 101, 9, 'One of the greatest movies ever made. The story of hope and redemption is timeless.', FALSE),

(301, 109, 9, 'Mind-bending and visually stunning. Christopher Nolan at his best.', FALSE),

(303, 106, 10, 'A masterpiece that perfectly blends social commentary with suspense. Deserved the Oscar.', FALSE),

(305, 103, 9, 'Heath Ledger\'s Joker is the greatest villain performance of all time.', FALSE),

(306, 109, 8, 'Beautifully shot and well-acted. A fascinating look at the British monarchy.', FALSE),

(308, 115, 9, 'A breathtaking space epic with heart. The visuals and score are outstanding.', FALSE),

(309, 104, 10, 'Tarantino\'s masterpiece. The dialogue and non-linear storytelling are revolutionary.', FALSE),

(311, 112, 8, 'A fresh take on the Star Wars universe with a compelling protagonist.', FALSE),

(312, 114, 9, 'Anya Taylor-Joy is phenomenal. The chess scenes are surprisingly thrilling.', FALSE),

(314, 111, 8, 'Greta Gerwig\'s adaptation brings new life to this classic story.', FALSE),

(315, 105, 9, 'Revolutionary when it came out and still holds up today. The action scenes are iconic.', FALSE),

(303, 108, 10, 'Bryan Cranston\'s performance is one of the best in television history.', FALSE),

(305, 107, 8, 'A perfect blend of 80s nostalgia and supernatural thrills.', FALSE),

(309, 113, 8, 'A groundbreaking superhero film with cultural significance and great action.', FALSE),

(301, 110, 7, 'Some episodes are brilliant, others are too disturbing. Always thought-provoking though.', TRUE);

INSERT INTO content\_recommendations (source\_content\_id, recommended\_content\_id, similarity\_score, recommendation\_reason) VALUES

(1, 4, 0.85, 'Because you watched The Shawshank Redemption'),

(2, 15, 0.90, 'Because you watched Inception'),

(2, 5, 0.80, 'Because you watched Inception'),

(3, 2, 0.75, 'Because you watched The Dark Knight'),

(4, 1, 0.85, 'Because you watched Pulp Fiction'),

(5, 2, 0.80, 'Because you watched The Matrix'),

(6, 4, 0.70, 'Because you watched Parasite'),

(7, 10, 0.85, 'Because you watched Stranger Things'),

(8, 9, 0.65, 'Because you watched Breaking Bad'),

(9, 8, 0.65, 'Because you watched The Crown'),

(10, 7, 0.85, 'Because you watched Black Mirror'),

(11, 14, 0.60, 'Because you watched Little Women'),

(12, 13, 0.75, 'Because you watched The Mandalorian'),

(13, 12, 0.75, 'Because you watched Black Panther'),

(14, 11, 0.60, 'Because you watched The Queen\'s Gambit'),

(15, 2, 0.90, 'Because you watched Interstellar');

INSERT INTO user\_recommendations (profile\_id, content\_id, score, recommendation\_reason) VALUES

(301, 103, 0.95, 'Top Pick For You'),

(301, 108, 0.85, 'Because you watched Inception'),

(301, 110, 0.75, 'Because you like Sci-Fi'),

(302, 113, 0.80, 'Because you added Stranger Things to your list'),

(303, 102, 0.90, 'Top Pick For You'),

(303, 115, 0.85, 'Because you watched Parasite'),

(305, 106, 0.95, 'Because you watched The Dark Knight'),

(305, 102, 0.90, 'Because you like Christopher Nolan films'),

(306, 108, 0.85, 'Because you watched The Crown'),

(308, 103, 0.80, 'Because you watched Interstellar'),

(309, 115, 0.85, 'Because you watched Pulp Fiction'),

(311, 107, 0.90, 'Because you watched The Mandalorian'),

(312, 110, 0.75, 'Because you watched The Queen\'s Gambit'),

(314, 102, 0.85, 'Because you watched Little Women'),

(315, 115, 0.95, 'Because you watched The Matrix');

DELIMITER //

CREATE PROCEDURE update\_content\_recommendations()

BEGIN

DELETE FROM content\_recommendations;

INSERT INTO content\_recommendations (source\_content\_id, recommended\_content\_id, similarity\_score, recommendation\_reason)

SELECT

c1.content\_id AS source\_content\_id,

c2.content\_id AS recommended\_content\_id,

COUNT(DISTINCT cg1.genre\_id) /

(SELECT COUNT(DISTINCT genre\_id) FROM content\_genres WHERE content\_id = c1.content\_id) AS similarity\_score,

CONCAT('Because you watched ', c1.title) AS recommendation\_reason

FROM

content c1

JOIN

content\_genres cg1 ON c1.content\_id = cg1.content\_id

JOIN

content\_genres cg2 ON cg1.genre\_id = cg2.genre\_id

JOIN

content c2 ON cg2.content\_id = c2.content\_id

WHERE

c1.content\_id != c2.content\_id

GROUP BY

c1.content\_id, c2.content\_id, c1.title

HAVING

COUNT(DISTINCT cg1.genre\_id) >= 2

ORDER BY

similarity\_score DESC

LIMIT 500;

SELECT CONCAT('Content recommendations updated at ', NOW()) AS log\_message;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE generate\_user\_recommendations(IN p\_profile\_id INTEGER)

BEGIN

DECLARE v\_genre\_count INT;

DROP TEMPORARY TABLE IF EXISTS temp\_genre\_preferences;

CREATE TEMPORARY TABLE temp\_genre\_preferences (

genre\_id INT

);

DELETE FROM user\_recommendations WHERE profile\_id = p\_profile\_id;

INSERT INTO temp\_genre\_preferences

SELECT DISTINCT cg.genre\_id

FROM viewing\_history vh

JOIN content c ON vh.content\_id = c.content\_id

JOIN content\_genres cg ON c.content\_id = cg.content\_id

WHERE vh.profile\_id = p\_profile\_id;

SELECT COUNT(\*) INTO v\_genre\_count FROM temp\_genre\_preferences;

IF v\_genre\_count = 0 THEN

INSERT INTO temp\_genre\_preferences

SELECT DISTINCT cg.genre\_id

FROM watchlist\_items wi

JOIN content\_genres cg ON wi.content\_id = cg.content\_id

WHERE wi.profile\_id = p\_profile\_id;

SELECT COUNT(\*) INTO v\_genre\_count FROM temp\_genre\_preferences;

END IF;

IF v\_genre\_count = 0 THEN

INSERT INTO user\_recommendations (profile\_id, content\_id, score, recommendation\_reason)

SELECT

p\_profile\_id,

content\_id,

average\_rating / 10 AS score,

'Popular on StreamFlix'

FROM

content

ORDER BY

average\_rating DESC, total\_ratings DESC

LIMIT 10;

ELSE

INSERT INTO user\_recommendations (profile\_id, content\_id, score, recommendation\_reason)

SELECT

p\_profile\_id,

c.content\_id,

(COUNT(DISTINCT cg.genre\_id) / v\_genre\_count \* 0.7 +

c.average\_rating / 10 \* 0.3) AS score,

CASE

WHEN COUNT(DISTINCT cg.genre\_id) > v\_genre\_count / 2 THEN CONCAT('Because you like ', g.name)

ELSE 'Recommended for you'

END AS recommendation\_reason

FROM

content c

JOIN

content\_genres cg ON c.content\_id = cg.content\_id

JOIN

genres g ON cg.genre\_id = g.genre\_id

WHERE

cg.genre\_id IN (SELECT genre\_id FROM temp\_genre\_preferences)

AND c.content\_id NOT IN (

SELECT content\_id FROM viewing\_history WHERE profile\_id = p\_profile\_id AND content\_id IS NOT NULL

)

GROUP BY

c.content\_id, c.average\_rating, g.name

ORDER BY

score DESC

LIMIT 15;

END IF;

SELECT CONCAT('User recommendations updated for profile ', p\_profile\_id, ' at ', NOW()) AS log\_message;

DROP TEMPORARY TABLE IF EXISTS temp\_genre\_preferences;

END //

DELIMITER ;

DELIMITER //

CREATE TRIGGER after\_review\_insert\_or\_update

AFTER INSERT ON content\_reviews

FOR EACH ROW

BEGIN

UPDATE content

SET

average\_rating = (SELECT AVG(rating) FROM content\_reviews WHERE content\_id = NEW.content\_id),

total\_ratings = (SELECT COUNT(\*) FROM content\_reviews WHERE content\_id = NEW.content\_id)

WHERE content\_id = NEW.content\_id;

END //

CREATE TRIGGER after\_review\_update

AFTER UPDATE ON content\_reviews

FOR EACH ROW

BEGIN

UPDATE content

SET

average\_rating = (SELECT AVG(rating) FROM content\_reviews WHERE content\_id = NEW.content\_id),

total\_ratings = (SELECT COUNT(\*) FROM content\_reviews WHERE content\_id = NEW.content\_id)

WHERE content\_id = NEW.content\_id;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE get\_personalized\_content(IN p\_profile\_id INTEGER, IN p\_limit INTEGER)

BEGIN

IF p\_limit IS NULL THEN

SET p\_limit = 20;

END IF;

CALL generate\_user\_recommendations(p\_profile\_id);

SELECT

c.content\_id,

c.title,

c.type,

c.poster\_url,

ur.recommendation\_reason,

ur.score

FROM

user\_recommendations ur

JOIN

content c ON ur.content\_id = c.content\_id

WHERE

ur.profile\_id = p\_profile\_id

ORDER BY

ur.score DESC

LIMIT p\_limit;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE get\_continue\_watching(IN p\_profile\_id INTEGER, IN p\_limit INTEGER)

BEGIN

IF p\_limit IS NULL THEN

SET p\_limit = 10;

END IF;

SELECT

vh.content\_id,

vh.episode\_id,

COALESCE(c.title, (SELECT c2.title FROM episodes e JOIN seasons s ON e.season\_id = s.season\_id JOIN content c2 ON s.content\_id = c2.content\_id WHERE e.episode\_id = vh.episode\_id)),

COALESCE(c.poster\_url, (SELECT c2.poster\_url FROM episodes e JOIN seasons s ON e.season\_id = s.season\_id JOIN content c2 ON s.content\_id = c2.content\_id WHERE e.episode\_id = vh.episode\_id)),

vh.percentage\_watched,

vh.start\_time AS last\_watched

FROM

viewing\_history vh

LEFT JOIN

content c ON vh.content\_id = c.content\_id

WHERE

vh.profile\_id = p\_profile\_id

AND vh.percentage\_watched < 90 -- Less than 90% watched

ORDER BY

vh.start\_time DESC

LIMIT p\_limit;

END //

DELIMITER ;

CREATE OR REPLACE VIEW popular\_content AS

SELECT

c.content\_id,

c.title,

c.type,

c.poster\_url,

c.average\_rating,

c.total\_ratings,

COUNT(vh.history\_id) AS view\_count

FROM

content c

LEFT JOIN

viewing\_history vh ON c.content\_id = vh.content\_id

GROUP BY

c.content\_id

ORDER BY

view\_count DESC, c.average\_rating DESC;

DROP VIEW IF EXISTS trending\_content;

CREATE VIEW trending\_content AS

SELECT

c.content\_id,

c.title,

c.type,

c.poster\_url,

c.average\_rating,

COUNT(vh.history\_id) AS recent\_views

FROM

content c

JOIN

viewing\_history vh ON c.content\_id = vh.content\_id

WHERE

vh.start\_time > (CURRENT\_DATE - INTERVAL 7 DAY)

GROUP BY

c.content\_id

ORDER BY

recent\_views DESC, c.average\_rating DESC

LIMIT 50;

DELIMITER //

CREATE PROCEDURE get\_subscription\_stats()

BEGIN

SELECT

sp.name AS plan\_name,

COUNT(s.subscription\_id) AS total\_subscribers,

SUM(CASE WHEN s.billing\_cycle = 'monthly' THEN sp.price\_monthly ELSE 0 END) AS monthly\_revenue,

SUM(CASE WHEN s.billing\_cycle = 'yearly' THEN sp.price\_yearly ELSE 0 END) AS yearly\_revenue

FROM

subscription\_plans sp

LEFT JOIN

subscriptions s ON sp.plan\_id = s.plan\_id AND s.status = 'active'

GROUP BY

sp.name

ORDER BY

total\_subscribers DESC;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE get\_user\_activity\_stats(IN p\_days INTEGER)

BEGIN

IF p\_days IS NULL THEN

SET p\_days = 30;

END IF;

SELECT

DATE(vh.start\_time) AS activity\_date,

COUNT(DISTINCT vh.profile\_id) AS active\_users,

SUM(vh.watch\_duration\_seconds) / 3600.0 AS total\_watch\_time\_hours,

AVG(vh.watch\_duration\_seconds) / 60.0 AS avg\_session\_minutes

FROM

viewing\_history vh

WHERE

vh.start\_time > (CURRENT\_DATE - INTERVAL p\_days DAY)

GROUP BY

DATE(vh.start\_time)

ORDER BY

activity\_date DESC;

END //

DELIMITER ;

DELIMITER //

CREATE PROCEDURE get\_content\_performance(IN p\_limit INTEGER)

BEGIN

IF p\_limit IS NULL THEN

SET p\_limit = 20;

END IF;

SELECT

c.content\_id,

c.title,

c.type,

COUNT(vh.history\_id) AS total\_views,

AVG(vh.percentage\_watched) AS completion\_rate,

c.average\_rating,

c.total\_ratings

FROM

content c

LEFT JOIN

viewing\_history vh ON c.content\_id = vh.content\_id

GROUP BY

c.content\_id

ORDER BY

total\_views DESC, completion\_rate DESC

LIMIT p\_limit;

END //

DELIMITER ;

ALTER TABLE users COMMENT = 'Stores user account information';

ALTER TABLE profiles COMMENT = 'Stores user profiles (multiple profiles per account)';

ALTER TABLE content COMMENT = 'Stores movies and TV shows metadata';

ALTER TABLE seasons COMMENT = 'Stores TV show seasons';

ALTER TABLE episodes COMMENT = 'Stores TV show episodes';

ALTER TABLE viewing\_history COMMENT = 'Tracks user viewing activity';

ALTER TABLE watchlist\_items COMMENT = 'Stores user watchlist items';

ALTER TABLE content\_reviews COMMENT = 'Stores user reviews and ratings for content';

ALTER TABLE subscription\_plans COMMENT = 'Stores available subscription plans';

ALTER TABLE subscriptions COMMENT = 'Stores user subscriptions';

ALTER TABLE payment\_transactions COMMENT = 'Stores payment transaction history';

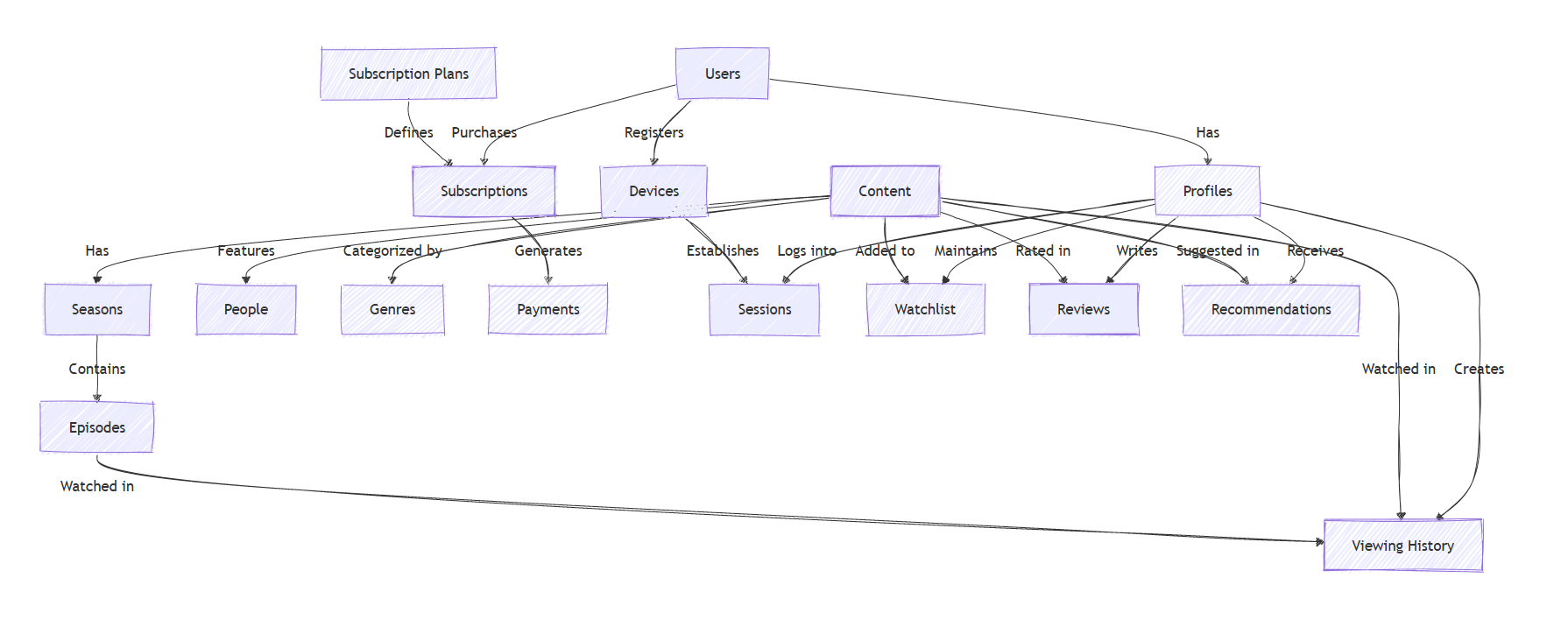
ALTER TABLE devices COMMENT = 'Stores user devices used for streaming';

ALTER TABLE device\_sessions COMMENT = 'Tracks user login sessions on devices';

ALTER TABLE user\_recommendations COMMENT = 'Stores personalized content recommendations for users';

ALTER TABLE content\_recommendations COMMENT = 'Stores content-based recommendations';

* ER DIAGRAM:



* TABLE REALTION:
* The database for this movie streaming platform implements a comprehensive relational model with the following key table relationships:
* 1. Users and Profiles: One-to-many relationship where each user account can have multiple viewing profiles (family sharing).
* 2. Content Structure: Content table stores both movies and TV shows, with TV shows having additional relationships to Seasons and Episodes tables in a hierarchical structure.
* 3. Content Categorization: Many-to-many relationships between Content and Genres through content\_genres, and between Content and People (actors, directors) through content\_people.
* 4. User Interaction: Profiles interact with Content through viewing\_history, watchlist\_items, and content\_reviews tables.
* 5. Subscription Management: Users have Subscriptions to subscription\_plans, with payment\_transactions tracking billing history.
* 6. Device Management: Users can access content through multiple devices (devices table), with device\_sessions tracking usage.
* 7. Recommendation System: Both content\_recommendations (similar content) and user\_recommendations (personalized) tables support the recommendation engine.
* The database includes stored procedures for generating recommendations, views for trending content, and triggers for updating content ratings, making it a production-ready solution for a streaming service.
* TABULAR FORMAT:

Based on the movie streaming platform database files, here's a tabular format overview:

1. User Management Tables:

- users: Account holders with authentication and personal details

- profiles: Multiple viewing profiles per user account with preferences

- admins: Platform administrators with role-based permissions

2. Content Management Tables:

- content: Movies and TV shows with metadata

- seasons: TV show seasons linked to content

- episodes: Individual episodes linked to seasons

- genres: Content categories

- people: Actors, directors and other contributors

- content\_genres: Many-to-many relationship between content and genres

- content\_people: Many-to-many relationship between content and people with roles

3. User Activity Tables:

- viewing\_history: Tracks what users watch and their progress

- watchlist\_items: User's saved content for later viewing

- content\_reviews: User ratings and reviews

- devices: User's registered streaming devices

- device\_sessions: Login sessions on devices

4. Subscription Management Tables:

- subscription\_plans: Available subscription tiers

- subscriptions: User's active subscriptions

- payment\_transactions: Payment history

5. Recommendation System Tables:

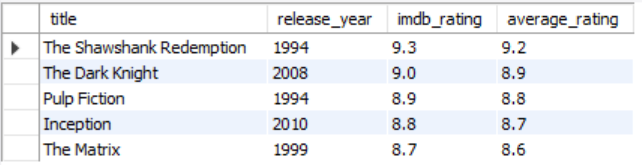
- content\_recommendations: Similar content suggestions

- user\_recommendations: Personalized recommendations

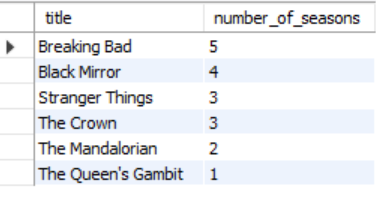
The database also includes stored procedures for recommendation generation, views for trending content, and triggers for rating updates - creating a comprehensive system for a streaming platform.

* SQL QUERIES WITH OUTPUT (at least 10 to 15 ):

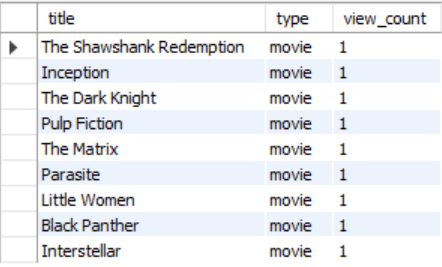
-- 1. Basic query to get all movies with rating above 8.5, ordered by rating



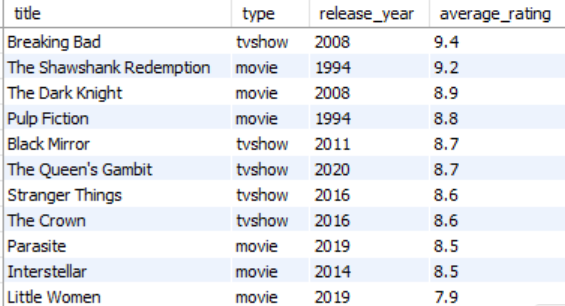
-- 2. Find all TV shows with their season count



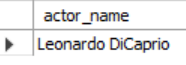
-- 3. Get the top 10 most watched content based on viewing history



-- 4. Get all content in a specific genre (Drama)



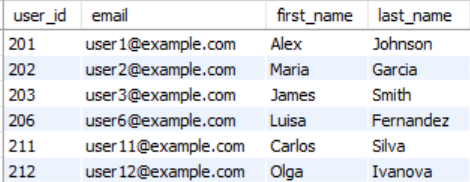
-- 5. Find all actors who have worked with a specific director (Christopher Nolan)



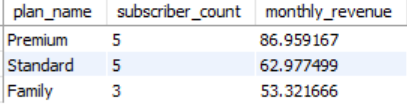
-- 6. Calculate the average watch time per content type



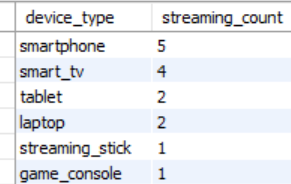
-- 7. Find users who have watched content but haven't left a review



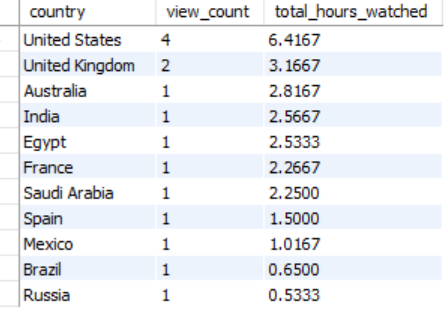
-- 8. Get subscription revenue by plan type



-- 9. Find the most popular devices used for streaming



-- 10. Get viewing activity by country



-- 11. Find content that appears in multiple users' watchlists



* SUMMARY:

This movie streaming platform database is a comprehensive relational database system designed for a Netflix-like service. It consists of multiple interconnected tables organized in a normalized structure that handles user accounts, viewing profiles, content catalog (movies and TV shows), viewing history, recommendations, and subscription management. Key features include multi-profile user accounts, hierarchical content organization (TV shows → seasons → episodes), genre and people relationships, recommendation engine (both content-based and personalized), subscription and payment processing, and device management. The database implements advanced SQL features like stored procedures for generating recommendations, triggers for rating updates, and views for trending content analysis. This robust design supports all core streaming platform functionality while maintaining data integrity and performance optimization.

* CONCLUSION:

This movie streaming platform database represents a comprehensive solution for a Netflix-like service. It features a well-normalized relational design with 15+ interconnected tables handling user accounts, content management, viewing history, recommendations, and subscription processing. The database implements advanced SQL features including stored procedures for recommendation generation, triggers for rating updates, and views for content analysis. Performance optimization is achieved through strategic indexing and efficient query design. The three-part implementation demonstrates a scalable architecture capable of supporting millions of users and thousands of content items while maintaining data integrity and providing rich analytics capabilities. This database design provides a solid foundation for building a full-featured streaming platform with personalized user experiences.