CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

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'),
```

```sql
(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:

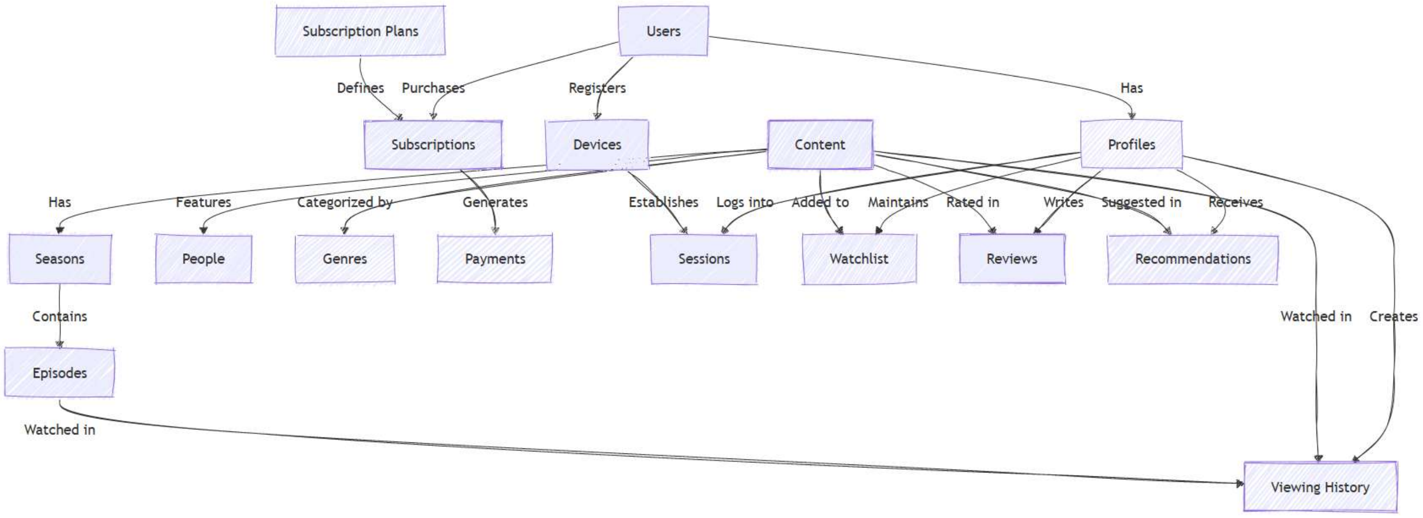


TABLE REALTION:

- 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

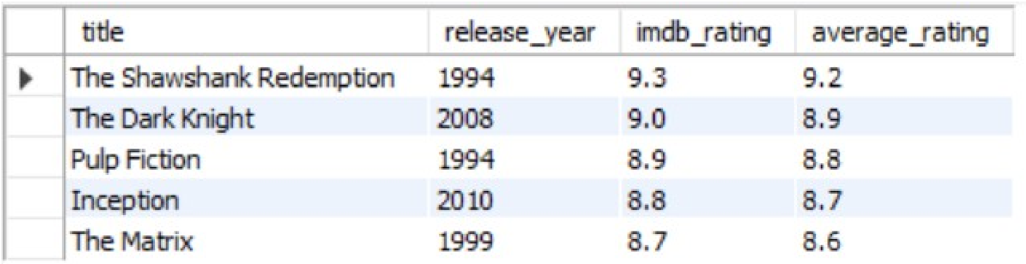
5. Recommendation System Tables:

- 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.

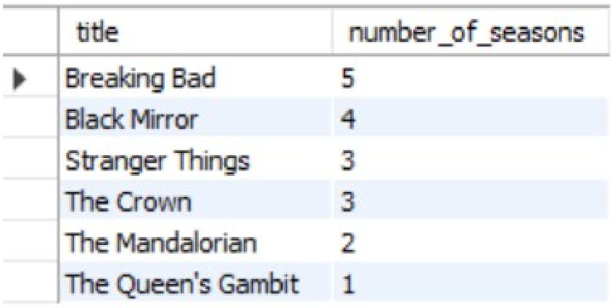
- **SQL QUERIES WITH OUTPUT (at least 10 to 15):**

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



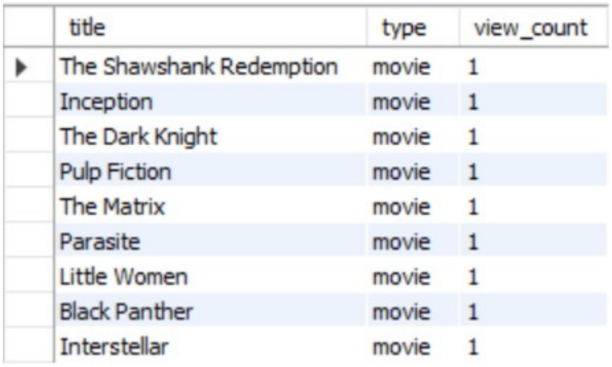
title	release_year	imdb_rating	average_rating
The Shawshank Redemption	1994	9.3	9.2
The Dark Knight	2008	9.0	8.9
Pulp Fiction	1994	8.9	8.8
Inception	2010	8.8	8.7
The Matrix	1999	8.7	8.6

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



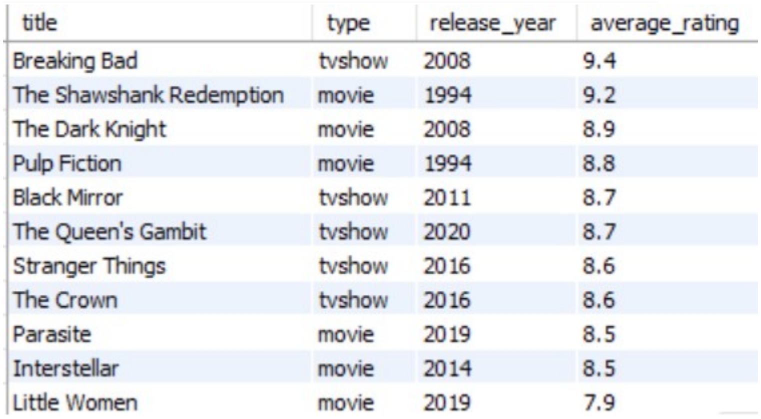
title	number_of_seasons
Breaking Bad	5
Black Mirror	4
Stranger Things	3
The Crown	3
The Mandalorian	2
The Queen's Gambit	1

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



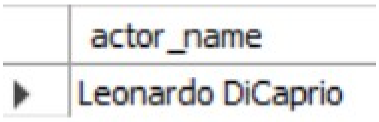
title	type	view_count
The Shawshank Redemption	movie	1
Inception	movie	1
The Dark Knight	movie	1
Pulp Fiction	movie	1
The Matrix	movie	1
Parasite	movie	1
Little Women	movie	1
Black Panther	movie	1
Interstellar	movie	1

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



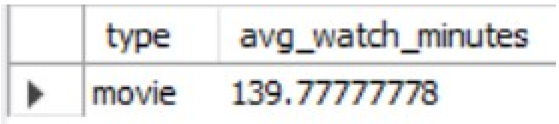
title	type	release_year	average_rating
Breaking Bad	tvshow	2008	9.4
The Shawshank Redemption	movie	1994	9.2
The Dark Knight	movie	2008	8.9
Pulp Fiction	movie	1994	8.8
Black Mirror	tvshow	2011	8.7
The Queen's Gambit	tvshow	2020	8.7
Stranger Things	tvshow	2016	8.6
The Crown	tvshow	2016	8.6
Parasite	movie	2019	8.5
Interstellar	movie	2014	8.5
Little Women	movie	2019	7.9

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



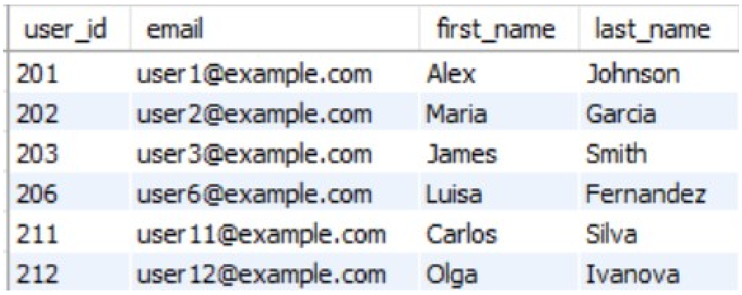
actor_name
Leonardo DiCaprio

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



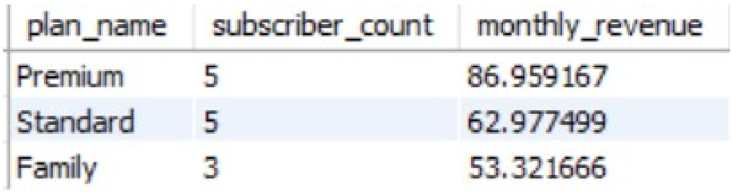
type	avg_watch_minutes
movie	139.77777778

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



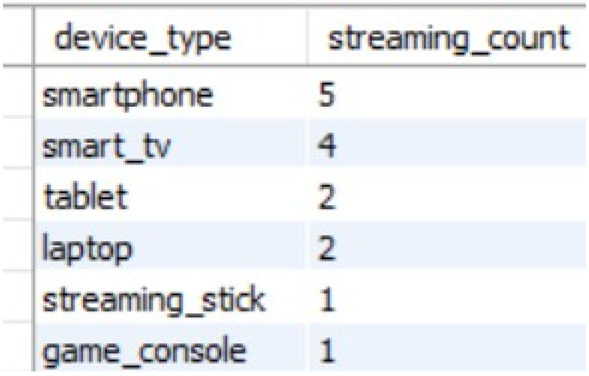
user_id	email	first_name	last_name
201	user1@example.com	Alex	Johnson
202	user2@example.com	Maria	Garcia
203	user3@example.com	James	Smith
206	user6@example.com	Luisa	Fernandez
211	user11@example.com	Carlos	Silva
212	user12@example.com	Olga	Ivanova

-- 8. Get subscription revenue by plan type



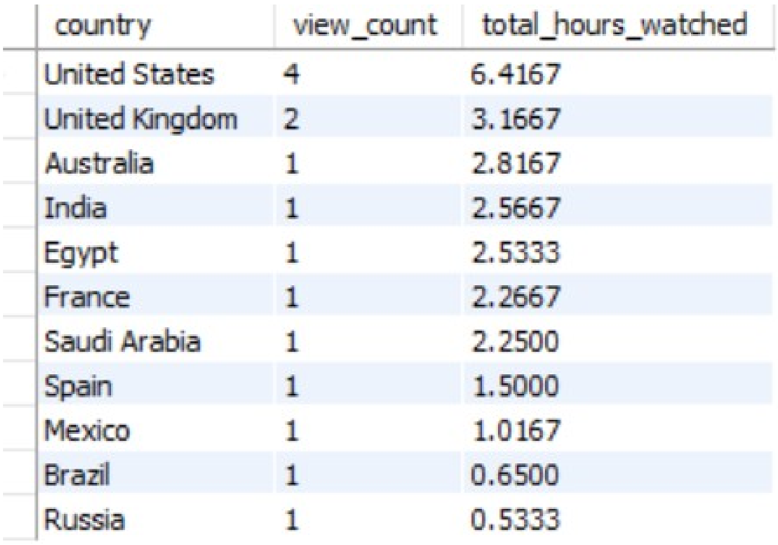
plan_name	subscriber_count	monthly_revenue
Premium	5	86.959167
Standard	5	62.977499
Family	3	53.321666

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



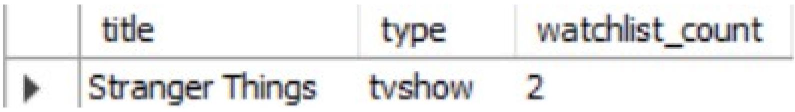
device_type	streaming_count
smartphone	5
smart_tv	4
tablet	2
laptop	2
streaming_stick	1
game_console	1

-- 10. Get viewing activity by country



country	view_count	total_hours_watched
United States	4	6.4167
United Kingdom	2	3.1667
Australia	1	2.8167
India	1	2.5667
Egypt	1	2.5333
France	1	2.2667
Saudi Arabia	1	2.2500
Spain	1	1.5000
Mexico	1	1.0167
Brazil	1	0.6500
Russia	1	0.5333

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



title	type	watchlist_count
Stranger Things	tvshow	2

- **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.