SOICT

Hanoi University of Science and Technology
School of Information and Communication Technology

Project Report

Movies Streaming System Database

Group 8

Course: Database Lab – IT3290E
Supervisor: Dr. Vu Tuyet Trinh

PROJECT DISTRIBUTION

GROUP 8

	Name	Task
1	Pham Quang Anh 20220071	Index, Function, DB Demo
2	Ngo Minh Trung 20226004	ERD, Schema, Trigger, Report
3	Nguyen Long Vu 20226006	Create tables, Data Generation, View

TABLE OF CONTENT

01 PROJECT DESCRIPTION

The movie streaming service database system is designed to manage user subscriptions, payments, and content.

Each account is uniquely identified by an id, some attributes of user account including password hash, account status (active/inactive), their name (first name and last name) and their nationality. An account can subscribe to a pre-determined subscription pack. The available subscription packs differs by level of access (1, 2, 3 – Default level is 1) and duration (6 months and 12 months). When they pay for the pack, a record of that subscription is added. Each level can access a certain number of contents. When a user is created, the user will automatically subscribe to a level 1 pack with unlimited duration.

Each content (includes movies and series) contains at least 1 episode. We consider movies as a series that have only one episode. Each content can belong to many different genres and involves different actors or actresses. Each content also has a list of available subtitles (language) which correspond to some countries.

Each time an user watches an episode, the timestamp and the last checkpoint of the episodes will be recorded. We also consider if the user has finished their movies or not. Users can also mark some contents as their favorite and rate each content by giving a point from 1 to 5.

02. REQUIREMENTS

Authentication

User Registration: Implement functionality for new users to register accounts.

Account Attributes: Store user account details including: unique account ID, password hash, account status (active, on hold, suspended, deleted), first name, last name, nationality

Subscription management

Subscription Packs: Define pre-determined subscription packs with attributes: level of access (1, 2, 3; default is 1), duration (6 months, 12 months)

Automatic Subscription: Automatically subscribe new users to a level 1 pack with unlimited duration.

Payment Records: Maintain records of subscription purchases and payment history.

Unsubscribe: Allow users to unsubscribe from their subscription packs.

Content management

Content Types: Store information about content, differentiating between movies and series.

Episodes: Each content must have at least one episode.

Genres: Allow content to belong to multiple genres.

Actors/Actresses: Store information about actors and actresses associated with each content.

Language: Maintain a list of available languages for each content, corresponding to different countries.

User interaction

Watch History: Record the timestamp and last checkpoint for each episode watched by a user.

Completion Status: Track whether users have finished watching movies or episodes.

Favorites: Allow users to mark content as favorites.

Content Rating: Enable users to rate content on a scale from 1 to 5. A user can only rate a content if they have finished one of episode in that content.

Browsing and Searching

Content Search: Implement search functionality for users to find content by:

- Genre
- Keywords (in title)
- Actor

Sorting: Display search results sorted by rating.

Personalized experience

Viewing History: Maintain a history of content watched by each user.

Recommendation System: Implement a recommendation system based on:

- Top viewed in the user’s country
- Based on the most viewed genre in user’s view history

03 DATABASE DESIGN

Entities

Country: Information about countries where the service operates.

Users: Details about the accounts registered on the platform.

Content: Details of all available content, including movies and series.

View_history (Weak Entity): Tracks the viewing activity of users for specific episodes.

Episode (Weak Entity): Represents individual episodes of series or standalone episodes for movies.

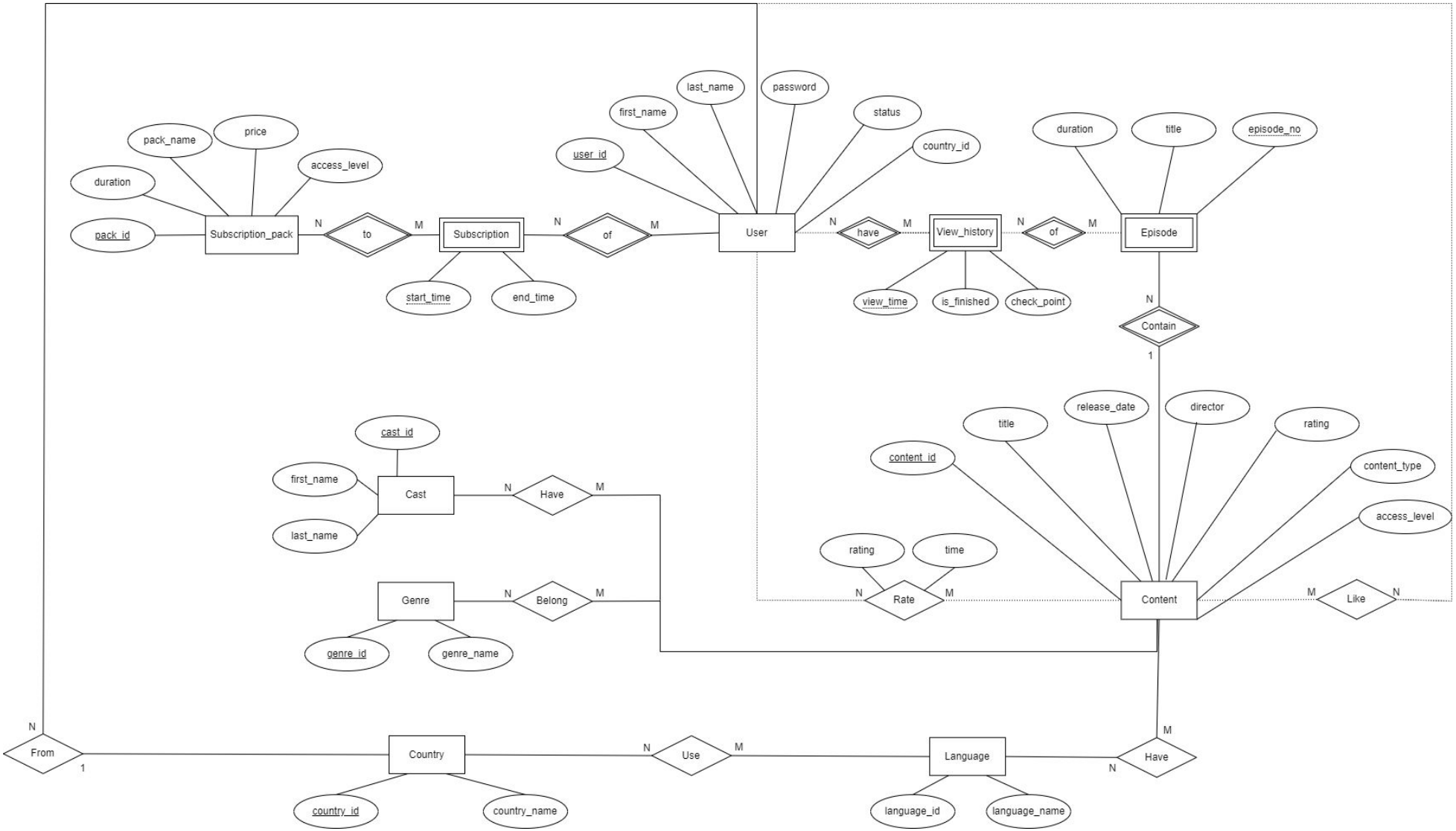
Language: Details of languages available on the platform.

Casts: Information about actors and actresses involved in the content.

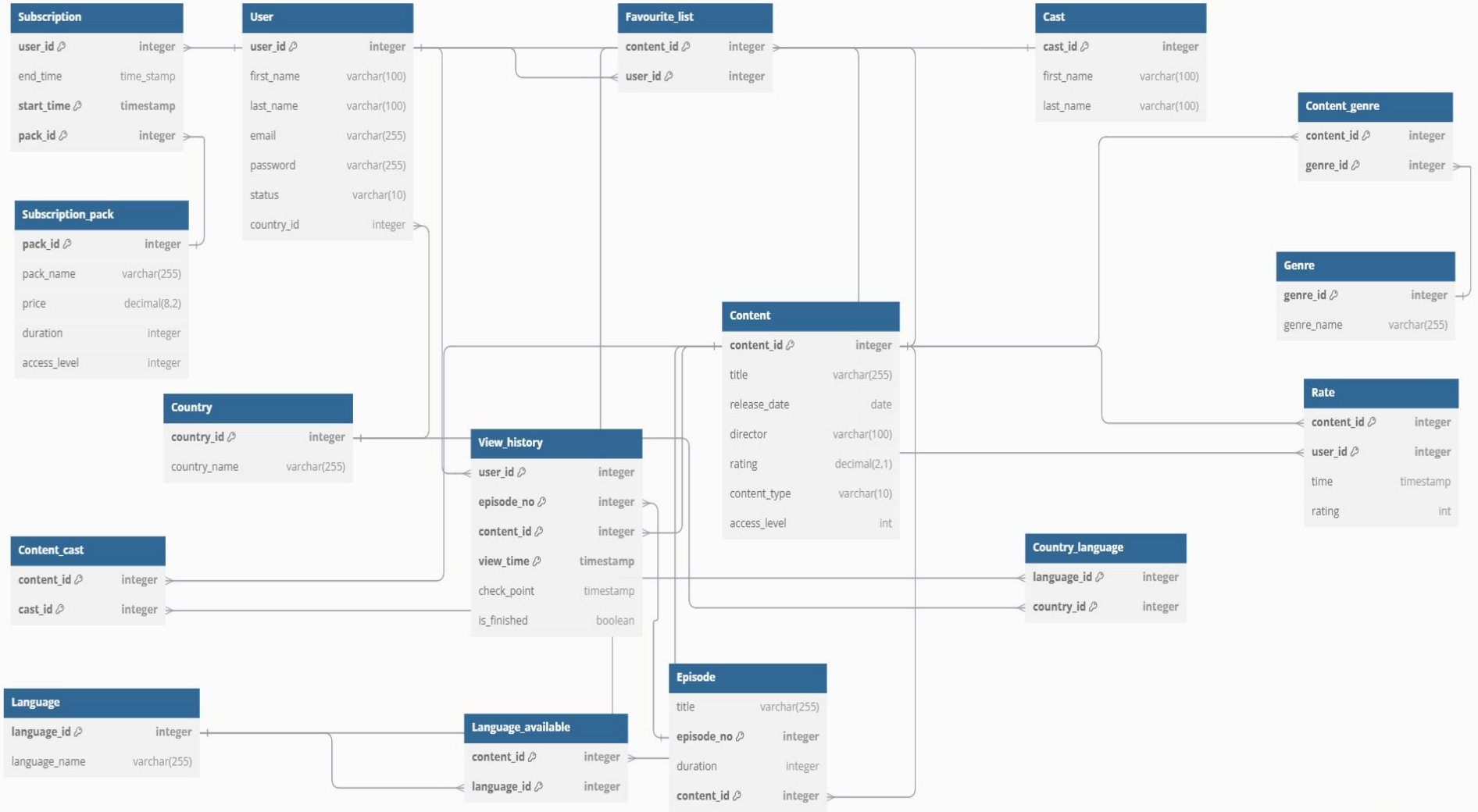
Genre: Categories of content available on the platform, such as drama, comedy, or thriller.

Subscription_pack: Details of subscription plans offered by the service: Free, Standard (6/12 months), Premium (6/12 months).

ERD


pack_name
price
duration
access_level
pack_id
Subscription_pack
N
to
M
Subscription
start_time
end_time
N
of
M
last_name
password
first_name
status
user_id
country_id
User
N
have
M
View_history
N
of
M
duration
title
episode_no
Episode
view_time
is_finished
check_point
N
Contain
1
cast_id
first_name
Cast
last_name
N
Have
M
Genre
N
Belong
M
genre_id
genre_name
title
release_date
director
rating
content_id
content_type
access_level
rating
time
N
Rate
M
Content
M
Like
N
N
From
1
Country
country_id
country_name
N
Use
M
Language
language_id
language_name
M
Have
N

Relational schema



04. TABLES

Country

Represents the countries where the platform is available.

country_id: Unique identifier for each country (Primary Key).

country_name: Name of the country (e.g., 'USA', 'India').

Users

Stores user information and their registration details.

user_id: Unique identifier for each user (Primary Key).

first_name: User's first name.

last_name: User's last name.

email: Unique email address for the user

password: User's account password

status: Current status of the user (active, inactive).

country_id: Country associated with the user (Foreign Key referencing Country.country_id).

Content

Represents movies and series available on the platform.

content_id: Unique identifier for each content (Primary Key).

title: Title of the movie or series.

release_date: Release date of the content.

director: Name of the director.

rating: Viewer rating for the content (1.0 to 5.0).

content_type: Type of content ('movie' or 'series').

access_level: Minimum access level required to view the content (1 to 3).

Episode

Details episodes for series content.

content_id: Identifier for the series (Primary Key, Foreign Key referencing Content.content_id).

episode_no: Episode number within the series (Primary Key in combination with content_id).

title: Title of the episode.

duration: Duration of the episode.

Genre

Defines the different genres for content classification.

genre_id: Unique identifier for each genre (Primary Key).

genre_name: Name of the genre (e.g., 'Action', 'Comedy').

Casts

Stores information about actors or actresses in content.

cast_id: Unique identifier for each cast member (Primary Key).

first_name: First name of the cast member.

last_name: Last name of the cast member.

Rate

Records user ratings for content.

content_id: Identifier of the content (Foreign Key referencing Content.content_id).

user_id: Identifier of the user (Foreign Key referencing Users.user_id).

time: Timestamp when the rating was given.

rating: User's rating for the content (1 to 5).

Subscription_pack

Details the subscription plans offered by the platform.

pack_id: Unique identifier for each subscription plan (Primary Key).

pack_name: Name of the subscription plan.

price: Cost of the subscription plan.

duration: Duration of the subscription ('6', '12', or 'infinity').

access_level: Access level provided by the subscription plan (1 to 3).

Subscription

Tracks users' subscriptions.

user_id: Identifier of the user (Foreign Key referencing Users.user_id).

pack_id: Identifier of the subscription plan (Foreign Key referencing Subscription_pack.pack_id).

start_time: Start date of the subscription (Primary Key in combination with user_id and pack_id).

end_time: End date of the subscription, defaults to 'infinity'.

Content_genre

Maps content items to their respective genres.

content_id: Identifier of a content (Foreign Key referencing Content.content_id).

genre_id: Identifier of the genre (Foreign Key referencing Genre.genre_id).

Content_cast

Links content items to their cast members.

content_id: Identifier of the content (Foreign Key referencing Content.content_id).

cast_id: Identifier of a cast member (Foreign Key referencing Casts.cast_id).

Favourite list

Tracks users' favorite content.

content_id: Identifier of a content (Foreign Key referencing Content.content_id).

user_id: Identifier of the user (Foreign Key referencing Users.user_id).

Language

Represents the available languages.

language_id: Unique identifier for each language (Primary Key).

language_name: Name of the language (e.g., 'English', 'Spanish').

Language_available

Tracks languages in which content is available.

content_id: Identifier of the content (Foreign Key referencing Content.content_id).

language_id: Identifier of the language (Foreign Key referencing Language.language_id).

Country_language

Maps countries to their officially supported languages.

country_id: Identifier of the country (Foreign Key referencing Country.country_id).

language_id: Identifier of the language (Foreign Key referencing Language.language_id).

View_history

Tracks users' viewing activities on the platform.

user_id: Identifier of the user (Foreign Key referencing Users.user_id).

content_id: Identifier of the content (Foreign Key referencing Episode.content_id).

episode_no: Episode number (Foreign Key referencing Episode.episode_no).

view_time: Timestamp of when the viewing occurred (Primary Key).

check_point: The timestamp in the video where the user paused or stopped.

PK(user_id, content_id, episode_id, view_time)

05. TRIGGER

Trigger 1:
update_content_rating

This trigger will update the average rating attributes of each content in table Content whenever a record of rating of that content is inserted or deleted in **Rate** table.

Trigger 2:
check_subscription_overlapping

This trigger will check whether a user can subscribe to a new pack. The condition is that the access level of the new pack must be greater than the current pack. If the condition is false, the trigger will raise exception.

Trigger 3:
new_user_pack

Whenever there is a new user, this trigger will automatically add a subscription record for that user with the free pack with unlimited duration.

05. TRIGGER

Trigger 4:
check_user_can_rate_content

This trigger check if a user can rate a content. If a user haven't finished any episode of that content, they cannot rate it. If we try to add a new record into **Rate** table, the database will raise an exception.

Trigger 5:
check_user_access

This trigger checks whether users can access a content based on their current subscribed pack. If a user cannot access into a content, inserting new record into **View_history** corresponding to that user and that content will raise an exception.

Trigger 6:
before_rate_insert

This will delete old record of users' rating to a content if they are trying to rate that content again. This makes sure that there is only a record corresponding to a pair of user and content.

06. VIEWS

1. User subscription history

This view provides a detailed history of users' subscription activities, including their full name, the subscription pack they chose, the price of the pack, and the subscription's start and end times. It helps track users' subscription patterns and analyze subscription trends.

2. User favourite list

This view displays the list of content marked as favorites by users. It includes the user's full name, the title of the content, its type (e.g., movie or series), and its rating. This is useful for understanding user preferences and building recommendation systems.

3. User watch history

This view shows users' watch histories, including the user's full name, the content title, episode number (if applicable), viewing progress (checkpoint), whether they finished watching, and the time of viewing. It is essential for tracking user engagement and building personalized viewing experiences.

06. VIEWS

4. Content management

This view provides an overview of content metadata, including the title, release date, director, rating, content type, access level, and associated genres. It groups content by genre and is useful for managing and categorizing the content library.

5. Episode management

This view lists detailed information about episodes in a series, including the content title, episode number, episode title, and duration. It focuses exclusively on series content and is useful for managing episode-level data.

6. User management

This view displays user account information, including the user's full name, email, account status, and country of residence. It helps administrators manage user accounts and monitor user activity geographically.

06. VIEWS

7. Top rated content

This view lists the highest-rated content, including the title, rating, and associated genres. It is sorted by rating in descending order and is useful for highlighting popular or critically acclaimed content.

07. FUNCTION

1. recommend_content_by_location (user_id_input INTEGER)

Purpose: This function recommends content to a user based on the content viewed by others in the same location (country) within the last 30 days. The recommendations are based on popularity (most viewed) and rating.

How it works:

- The function first finds the country of the input user (user_id_input).
- It then aggregates content viewed by users from the same country in the last 30 days, joining view_history, content, content_genre, and genre tables.
- The results are ordered by the most viewed content and highest rating, limiting the results to the top 10 content.
- The STRING_AGG function is used to gather distinct genre names for each content.

07. FUNCTION

2. recommend_content_by_genre (user_id_input INTEGER, top_genres_count INTEGER DEFAULT 3)

Purpose: This function recommends content based on the user's most viewed genres. It returns content matching the user's favorite genres that the user has not yet watched, with the highest-rated and newest content prioritized.

How it works:

- The function first identifies the top N genres (default is 3) that the user has watched most frequently, using the view_history and genre tables.
- Then, it selects content that belongs to these genres, ensuring the content has not been watched by the user.
- The results are sorted by rating and release date, limiting the results to the top 10 recommendations.
- STRING_AGG again aggregates all genres associated with the content.

07. FUNCTION

3. subscribe_to_pack(NEW_USER_ID INT, NEW_PACK_ID INT)

Purpose: This function subscribes a user to a specific subscription pack, ensuring that they are not already subscribed to the same pack and updating their subscription if necessary.

4. unsubscribe(NEW_USER_ID INT)

Purpose: This function allows a user to unsubscribe from their current pack and revert to the default subscription pack (pack_id = 1).

5. search_content_by_keyword (search_keyword TEXT)

Purpose: This performs a keyword-based search across multiple related tables (content, genre, casts, view_history) and returns a list of content items matching the search keyword.

The search is case-insensitive and matches the search_keyword in the content title, cast members' names (first or last), or genre names. The results are ordered by the number of completed views (view_count) in descending order, and then by rating in descending order.

08. INDEX

Purpose: These indices are trying to optimize the process of recommendation by users' most viewed genre and their country

1. btree index on user_id, view_time desc, content_id in View_history table

2. brin index on content_id in View_history table

3. btree index on user_id, country_id in Users table

4. btree index on content_id, genre_id in Content_genre table

5. btree index on content_id in Content table

6. btree index on rating desc in Content table

7. btree index on genre_id in Genre table

08. INDEX

Recommendation by location

Without indices: 309 ms
With indices: 173 ms

Average processed time over 10 times

Recommendation by most viewed genre

Without indices: 146 ms
With indices: 140 ms

Average processed time over 10 times