

## **Project outline - Movie Streaming Service Database System**

### **1. Context**

The explosive growth of streaming platforms like Netflix, Disney+, and HBO Max has led to a significant demand for efficient data management systems. These platforms handle enormous amounts of data related to users, content, interactions, and payments. For example, Netflix processes data from millions of users daily, including their viewing history, subscription status, and personalized recommendations.

To meet these demands, the system must ensure fast data retrieval, high concurrency handling, and stability to serve millions of simultaneous users. Efficient data management is crucial for both operational performance and user experience. A system capable of analyzing user behavior, tracking watch times, and managing large-scale payments is essential for streaming platforms.

Moreover, users expect personalized content recommendations tailored to their tastes, meaning the system needs to aggregate and analyze viewership data to offer content suggestions in real-time. For instance, Netflix offers movie suggestions based on a user's past watching habits, providing a tailored experience that enhances engagement.

### **2. Descriptions**

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/on hold/suspended/deleted), their name (first name and last name) and their nationality.

An account can subscribe to a pre-determined subscription pack. The available subscription packs differ 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, however they will still be differentiated by content type: movies or series. Each content can belong to many different genres and involves different actors or actresses. Each content also has a list of available subtitles which correspond to some countries.

Each time a 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.

### 3. Requirements:

- **User authentication:** new users can register account(s) to use the service
- **Purchasing subscription:** users can choose between several pre-defined subscription packs, each with its own limitation, and purchase it for a certain amount of time. Users can choose to unsubscribe any time they want and see their payment history.
- **Browsing:** users can search for content by genre, keywords (in title/description), by actor, director. The system will show what contents match the keywords, sorted by rating.
- **Personalized experience:** users can watch content, add content to their own favourite list, rate the content, see their history, start from their last checkpoint of each episode.

**\*\* Recommendation system:** a procedure runs behind the system to recommend contents to users. Each action (viewing content, whether they finished watching it, view history, browsing history) will be accounted for the procedure.

### Different types of recommendations:

- **By accumulated views in recent time period.** Users are recommended to watch contents viewed most by accounts in the same geographical location. Criteria that also used including average age of user and release date. How much each criterion contributes to the order of recommendation is taken by weighted sum.
- **By user historical data.**  
Genre: Through watching history, selecting genre with most content watched by user, prioritized by recent date, then recommend contents from that genre.  
Director/actor: select director/actor with most content watched by user, prioritized by recent date, then recommend content also has that director/actor.  
Contents with available subtitles in user's language of choice will also be prioritized.
- **By similarity of 2 users:**  
For a target user, select some other users and compare their historical rating of the same content. Using Pearson correlation

### Regular user:

- Authentication process (registration, login procedure, managing account details)
- Content browsing, query for content based on genres, ratings, or recommendations by the system.

- User interactions: adding to favorite playlist, leaving ratings, purchasing subscriptions

#### **Content Producer:**

- Can create, delete content entries, updating metadata attributes (description) belonging to their ID
- Add, update, remove episodes within series entries belong to their ID

#### **Administrators:**

- Having full access to all database tables
- Create, read, update and delete all records across all tables
- Update system settings (modify relation & constraint)

#### \* Sub-division: **Content Manager**

- Limiting access to content-related (Content, Movies, Series, Episodes) tables
- Create, read, update, delete content entries

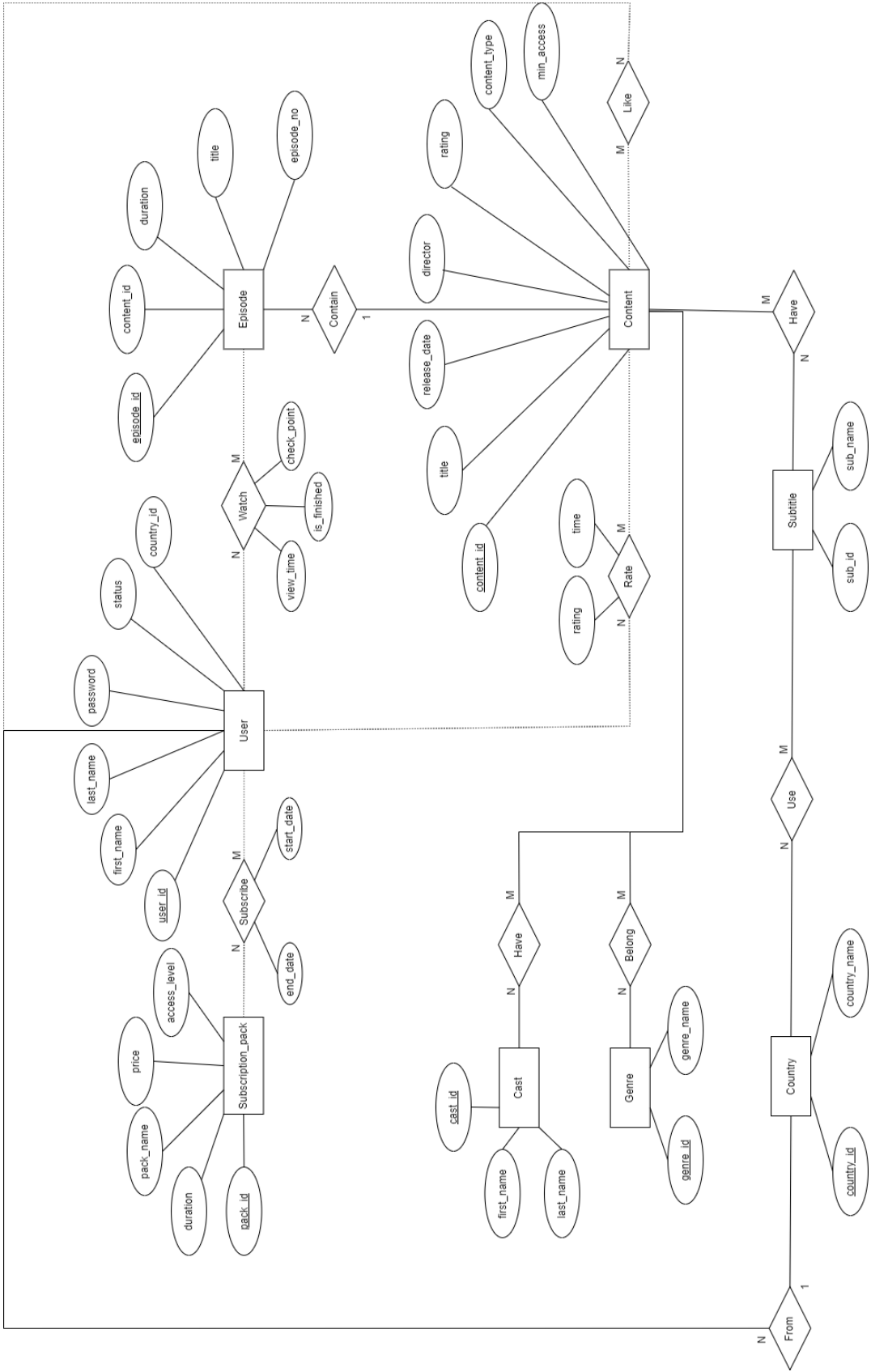
#### **User Manager**

- Limiting access to user profile related (User, Subscription, ...) tables
- Create, read, update, delete user account, update account status

#### **Data Analysts**

- Read access to all tables, including user activity logs and viewing statistics

### **4. ERD**



## Entities:

1. User  
Represents individuals who are using the streaming platform.
  2. Country  
Represents the countries where users are located.
  3. Subscription\_pack  
Represents various subscription plans available on the platform. These packs may differ in pricing, duration, and the level of access they provide.
  4. Content  
Represents movies or series. Each piece of content contains basic information, such as title and director, and it can belong to various genres. Content may also be rated by users and linked to multiple episodes or 1 for movies.
  5. Episode  
Represents individual episodes
  6. Cast  
Represents the actors and actresses featured in the platform's content. Multiple cast members can be associated with the same content, and a single actor can appear in multiple shows or movies.
  7. Genre  
Represents different categories or genres of content, such as drama, comedy, or action. A single content item can belong to multiple genres, and genres help users discover content of their interest.
  8. Subtitle  
Represents subtitles available for the content.
- 

## Relationships:

1. Subscribe (between User and Subscription\_pack)  
A user can subscribe to multiple subscription packs. Each subscription pack has its own duration, which is independent of other packs. Therefore, if a user subscribes to multiple packs, the total duration of their subscriptions is not cumulative.  
Many-to-Many (N-M)

2. Accessed (between User\_level and Content)  
Indicates which content items are accessible at a given user level. Higher levels may unlock more exclusive content.  
Many-to-Many (N-M)
3. From (between User and Country)  
Describes the association between a user and their country. A user must belong to a specific country, while multiple users can be from the same country.  
Many-to-One (N-1)
4. Watch (between User and Episode)  
Tracks which episodes a user has watched and how far they progressed (e.g., view time and checkpoints). It also records whether the episode was finished.  
Many-to-Many (N-M)
5. Contain (between Content and Episode)  
Describes the relationship between a content item (such as a TV show) and its episodes. A single content item can contain multiple episodes.  
One-to-Many (1-N)
6. Have (between Content and Cast)  
Represents the connection between content and the actors/actresses involved in it. A piece of content can feature many cast members, and a single actor can appear in multiple shows or movies.  
Many-to-Many (N-M)
7. Belong (between Content and Genre)  
Describes which genres a content item belongs to. Content can fall under multiple genres, and each genre can include many pieces of content.  
Many-to-Many (N-M)
8. Rate (between User and Content)  
Represents user feedback in the form of ratings. A user can rate multiple pieces of content, and each content item can have ratings from multiple users.  
Many-to-Many (N-M)
9. Like (between User and Content)  
Tracks which content is the favourite of a user. A user can like many content items, and each content item can receive likes from multiple users.  
Many-to-Many (M-N)

10. Use (between Country and Subtitle)

Represents the availability or use of specific subtitles in a given country. Different countries may prefer or require different subtitles.

Many-to-Many (N-M)

11. Have (between Content and Subtitle)

Describes the availability of subtitles for content. A piece of content can have multiple subtitles in different languages.

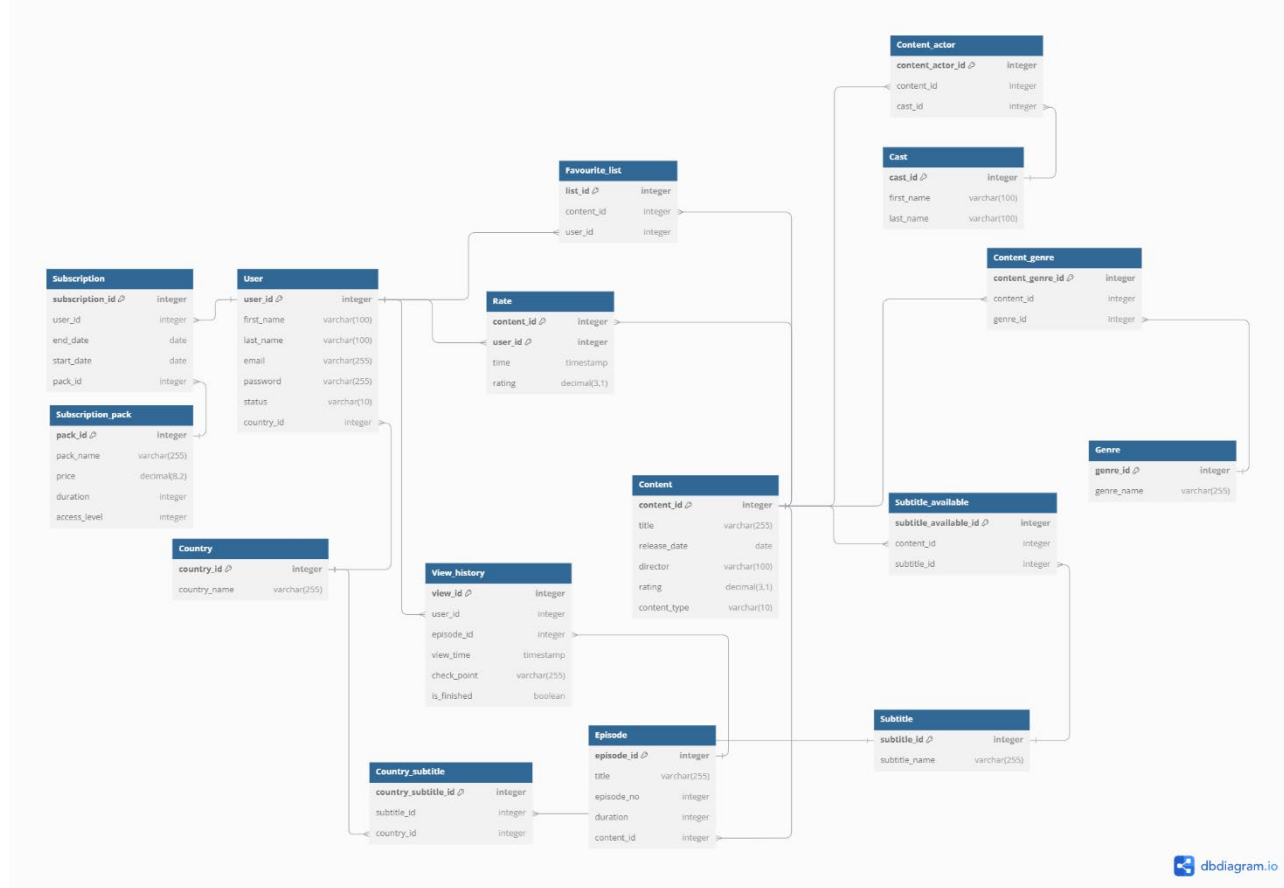
Many-to-Many (N-M)

12. Link (between User\_level and Subscription\_pack)

Defines which subscription packs are linked to specific user levels. Certain packs may correspond to higher access levels, giving users more content or features.

Many-to-Many (M-N)

## 5. Relational Schema



## Tables

### User Table

Stores information about users registered on the platform.

- 
- **user\_id:** Unique identifier for each user (Primary Key).
- **first\_name:** User's first name (up to 100 characters).
- **last\_name:** User's last name (up to 100 characters).
- **email:** User's unique email address (up to 255 characters).
- **password:** Encrypted password for authentication (up to 255 characters).



- - **status:** Indicates the user's current status (e.g., 'active', 'inactive').
  - **region\_id:** Foreign key linking the user to a **Region**.
- 

## Content Table

Holds information about the content available on the platform (movies, series, etc.).

- **content\_id:** Unique identifier for each piece of content (Primary Key).
  - **title:** Title of the content (up to 255 characters).
  - **release\_date:** The release date of the content.
  - **director:** Name of the content's director (up to 100 characters).
  - **rating:** Average rating of the content (up to 3 digits with 1 decimal).
  - **content\_type:** Type of content ('movie', 'series').
  - 
  - **min\_access:** The minimum level of level that can access to this content
- 

## Genre

### Table

res different genres available for content classification.

- **genre\_id:** Unique identifier for each genre (Primary Key).
  - **genre\_name:** Name of the genre (e.g., 'Action', 'Comedy').
- 

## Cast Table

Contains information about the cast members associated with content.

- **cast\_id:** Unique identifier for each cast member (Primary Key).
- **first\_name:** Cast member's first name (up to 100 characters).

- **last\_name:** Cast member's last name (up to 100 characters).
- 

### Subscription\_pack Table

Defines the available subscription plans.

- **pack\_id:** Unique identifier for each subscription pack (Primary Key).
  - **pack\_name:** Name of the subscription pack.
  - **price:** Price of the subscription plan (up to 8 digits with 2 decimals).
  - **access\_level:** Number of accessible contents differs by this level (1, 2, 3). The higher the level is, the more contents users can access.
  - **duration:** Duration of the subscription (in days).
- 

### User\_level Table

Categorizes users based on subscription level.

- **level\_id:** Unique identifier for each user level (Primary Key).
  - **level\_name:** Name of the user level (Free, Standard, Pro).
- 

### Episode Table

Stores details of individual episodes of a series.

- **episode\_id:** Unique identifier for each episode (Primary Key).
  - **title:** Title of the episode (up to 255 characters).
  - **episode\_no:** Episode number in the series.
  - **duration:** Length of the episode in minutes.
  - **content\_id:** Foreign key linking the episode to its parent content (series) in the **Content** table.
- 

### View\_history Table

Tracks content viewed by users, including progress and completion status.

- **view\_id:** Unique identifier for each view history entry (Primary Key).
  - **user\_id:** Foreign key referencing the **User** who viewed the content.
  - **content\_id:** Foreign key referencing the **Content** being viewed.
  - **view\_time:** Timestamp of when the content was viewed.
  - **check\_point:** Marks the point where the user left off (e.g., "01:15:30").
  - **is\_finished:** Boolean value indicating if the user finished the content.
- 

### Subscription Table

Tracks the subscription level and duration for each user.

- **subscription\_id:** Unique identifier for each subscription record (Primary Key).
  - **user\_id:** Foreign key referencing the User associated with the subscription.
  - **start\_date:** The start date of the subscription.
  - **end\_date:** The end date of the subscription.
- 

### Rate Table

Stores ratings given by users to content.

- **rate\_id:** Unique identifier for each rating entry (Primary Key).
  - **content\_id:** Foreign key referencing the rated Content.
  - **user\_id:** Foreign key referencing the **User** who gave the rating.
  - **time:** Timestamp when the rating was given.
  - **rating:** User's rating for the content (up to 3 digits with 1 decimal).
- 

### Content\_genre Table

Associates content with its respective genres.

- **content\_genre\_id:** Unique identifier for each content-genre relationship (Primary Key).

- **content\_id:** Foreign key referencing the **Content**.
  - **genre\_id:** Foreign key referencing the **Genre**.
- 

### **Content\_actor Table**

Maps cast members to the content they participated in.

- **content\_actor\_id:** Unique identifier for each content-cast relationship (Primary Key).
  - **content\_id:** Foreign key referencing the **Content**.
  - **cast\_id:** Foreign key referencing the **Cast**.
- 

### **Region Table**

Stores regions to customize user experience by location.

- **region\_id:** Unique identifier for each region (Primary Key).
  - **region\_name:** Name of the region (e.g., 'North America', 'Europe').
- 

### **Subtitle Table**

Contains different subtitle languages or options available for content.

- **subtitle\_id:** Unique identifier for each subtitle option (Primary Key).
  - **subtitle\_name:** Name or language of the subtitle (e.g., 'English', 'French').
- 

### **Country\_subtitle Table**

Defines the availability of subtitle options for specific regions.

- **country\_subtitle\_id:** Unique identifier for each region-subtitle relationship (Primary Key).
  - **subtitle\_id:** Foreign key referencing the **Subtitle**.
  - **country\_id:** Foreign key referencing the **Region**.
-

## Subtitle\_available Table

Tracks which subtitle options are available for specific content.

- **subtitle\_available\_id:** Unique identifier for each content-subtitle relationship (Primary Key).
  - **content\_id:** Foreign key referencing the Content.
  - **subtitle\_id:** Foreign key referencing the Subtitle.
- 

## 7. Constraint

### Primary Keys

- Unique IDs: user\_id, content\_id, genre\_id, cast\_id, pack\_id, level\_id, episode\_id, view\_id, subscription\_id, rate\_id, content\_genre\_id, content\_actor\_id, country\_id, subtitle\_id, country\_subtitle\_id, subtitle\_available\_id, access\_id.

### Foreign Keys

- User: region\_id → Region.region\_id
- Episode: content\_id → Content.content\_id
- View\_history: user\_id → User.user\_id, content\_id → Content.content\_id
- Subscription\_pack: level\_id → User\_level.level\_id
- Subscription: user\_id → User.user\_id
- Rate: content\_id → Content.content\_id, user\_id → User.user\_id
- Content\_genre: content\_id → Content.content\_id, genre\_id → Genre.genre\_id
- Content\_actor: content\_id → Content.content\_id, cast\_id → Cast.cast\_id
- Country\_subtitle: subtitle\_id → Subtitle.subtitle\_id, country\_id → Country.country\_id
- Subtitle\_available: content\_id → Content.content\_id, subtitle\_id → Subtitle.subtitle\_id

### Check

- **Content.content\_type:** (Series, Movies)

- **Subscription\_pack.access\_level, Content.minimum\_access:** (1, 2, 3)
- **Rate.rating:** between 1 and 5

#### **Cascade**

- **View\_history, Rate, Subscription:** Cascade on delete for related users or content.