**Streaming Platform**

Implementing a comprehensive data mart solution to enhance viewer engagement and optimize content strategy for films and series on our platform.

(By analyzing the performance of content across various dimensions such as genres, countries, user demographics, and viewing patterns, the data mart will provide actionable insights to improve decision-making processes)

Team:

Hazim Ghulam Farooq – 25148

Sameed Khatri - 25249

# Spaghetti Diagram



# Database Schema Report

## Overview

The database schema is designed to model a simplified version of a streaming service similar to Netflix. It includes tables for users, films, series, episodes, genres, actors, countries, languages, payment information, and user activity. The relationships between these entities are defined to maintain data integrity and to support the functionality of the streaming service.

## Table Explanations and Assumptions

### User:

Columns: id\_user, Name, password, email, age, PhoneNumber, creditCardID, countryId.

Assumptions: Each user has a unique profile with personal information and is linked to a credit card and country.

### User\_Profile:

Columns: id\_user\_profile, UserId, LanguagesId.

Assumptions: Each user can have multiple profiles, potentially with different language preferences.

### Watching\_History:

Columns: id\_watchingHistory, userProfileID, episodeID, filmID, ratings, starttime, endtime.

Assumptions: The episodeID is null for films, and filmID is null for series episodes to distinguish between the two types of content.

### Film:

Columns: id\_Film, title, duration, production\_year, age\_categoryid, filmDescription, addDate, filmPhoto, countryID.

Assumptions: Represents individual films with associated metadata.

### Series:

Columns: id\_serie, title, production\_year, age\_categoryid, season, serieDescription, addDate, seriePhoto, countryID.

Assumptions: Represents a series with metadata and is linked to multiple episodes.

### Episode:

Columns: id\_episode, episode\_title, seasonNumber, episodeDescription, seriesid, episodeDuration, episodePhoto.

Assumptions: Each episode belongs to a series and is uniquely identified within that series by season and title.

### Genre:

Columns: id\_genre, genre\_name.

Assumptions: Represents different genres of films and series.

### Genre\_Film:

Columns: Id\_genre, Id\_Film.

Assumptions: Many-to-many relationship between films and genres.

### Serie\_Genre:

Columns: Id\_Serie, Id\_genre.

Assumptions: Many-to-many relationship between series and genres.

### Actor:

Columns: id\_actor, name, photo.

Assumptions: Stores information about actors.

### Actor\_Film:

Columns: Id\_actor, Id\_FIlm.

Assumptions: Many-to-many relationship between actors and films.

### Actor\_Serie:

Columns: Id\_actor, Id\_serie.

Assumptions: Many-to-many relationship between actors and series.

### Country:

Columns: id\_country, country\_name, Continent, Region.

Assumptions: Represents countries, useful for user location and film production information.

### Film\_Country:

Columns: Id\_film, Id\_country.

Assumptions: Many-to-many relationship between films and countries of production.

### Serie\_Country:

Columns: Id\_serie, Id\_country.

Assumptions: Many-to-many relationship between series and countries of production.

### Languages:

Columns: id\_languages, language.

Assumptions: Represents languages available in the system.

### Plan:

Columns: id\_plan, planDescription, price, Duration.

Assumptions: Different subscription plans available for users.

### Payment:

Columns: id\_payment, id\_user, payment\_date, expiry\_date, id\_plan.

Assumptions: Represents payment records linked to users.

### Credit\_Card:

Columns: id\_credit\_card, cardNumber, CVC, expirationDate.

Assumptions: Represents credit card information for user payments.

### Film\_Wishlist:

Columns: id\_film\_wishlist, Id\_film, id\_user\_profile.

Assumptions: Represents films added to a user's wishlist.

### Serie\_Wishlist:

Columns: id\_serie\_wishlist, Id\_serie, id\_user\_profile.

Assumptions: Represents series added to a user's wishlist.

## Assumptions Made

Distinct Content Types: In the Watching\_History table, episodeID will be null for movies, and filmID will be null for series episodes to distinguish between films and series.

User Profiles: Each user can have multiple profiles with possibly different language settings.

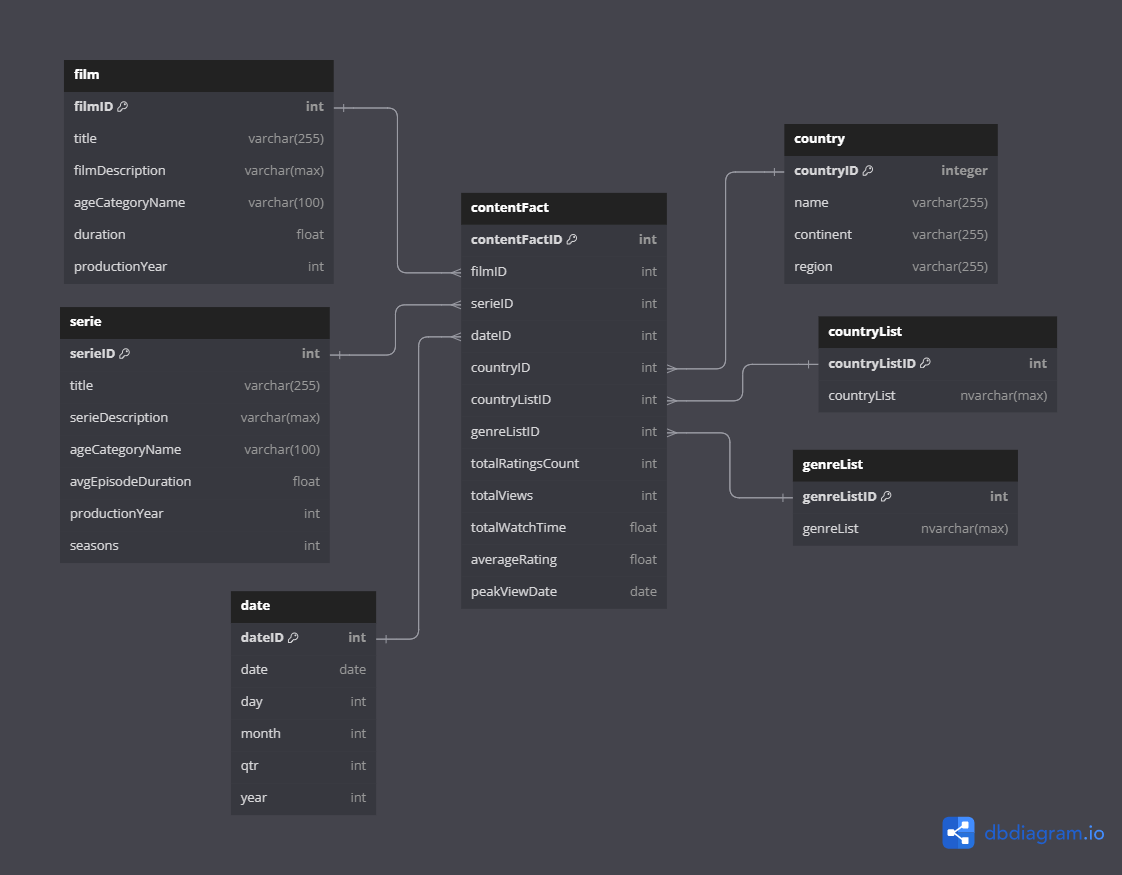
Many-to-Many Relationships: Multiple tables (like Genre\_Film, Actor\_Film) are used to manage many-to-many relationships (Bridge Tables).

Credit Card Management: Payment information and credit card details are stored separately but linked via user ID.

Content Localization: Countries and languages are used to support localization and regional content management.

# Data spaghetti to Data-Mart

## Data Mart



## Table Explanations

### **Film**

**Columns**: filmID, title, filmDescription, ageCategoryName, duration, productionYear.

**Purpose**: Stores metadata for films, allowing for quick access and analysis of film-related data.

### **Serie**

**Columns: serieID, title, serieDescription, ageCategoryName, avgEpisodeDuration, productionYear, seasons.**

**Purpose: Stores metadata for series, facilitating analysis on series-specific metrics.**

### **Country**

**Columns: countryID, name, continent, region.**

**Purpose: Stores information about countries, useful for regional analysis of content consumption.**

### **Date**

**Columns**: dateID, date, day, month, qtr, year.

**Purpose**: Date dimension table to allow for time-based aggregations and analysis.

### **ContentFact**

**Columns: contentFactID, filmID, serieID, dateID, countryID, countryListID, genreListID, totalRatingsCount, totalViews, totalWatchTime, averageRating, peakViewDate.**

**Purpose: Fact table that stores aggregated data for films and series, linking to dimensions such as date, country, and genre. This table is central to performance and usage analytics.**

### **CountryList**

**Columns**: countryListID, countryList.

**Purpose**: Supports the ContentFact table by providing lists of countries associated with content (countries where respective content is available), enabling multi-country content analysis.

### **GenreList**

**Columns: genreListID, genreList.**

**Purpose: Supports the ContentFact table by providing lists of genres associated with content (genres linked to respective content), facilitating genre-based analysis.**

# Integration with Main Database Schema

## Watching History Importance

The **Watching\_History** table from the main schema is crucial for feeding data into the data mart. It records user interactions with content, including start and end times, ratings, and whether the content is a film or a series episode. This data is essential for populating the **ContentFact** table with aggregated metrics.

Data Flow and Usage

**Data Ingestion**: Data from the main database spaghetti, particularly from tables like **Watching\_History**, **Film**, **Serie**, **Country**, **Genre\_Film**, and **Genre\_Serie**, are extracted and transformed into summary statistics.

**ETL Process**: The ETL (Extract, Transform, Load) process consolidates and processes this data:

Extracts user activity, content metadata, and contextual information.

Transforms the data into meaningful aggregations such as total views, total watch time, and average ratings. Stores the preprocessed data in the staging area (as csv files).

Loads the transformed data from the csv files of staging area into the **ContentFact** table.

**Business Intelligence**: The data mart supports BI tools (power BI) for creating dashboards and reports, enabling data-driven decision-making for content acquisition, user engagement strategies, and marketing campaigns.

# Conclusion

The integration of the main database spaghetti schema with the data mart allows for robust data analysis and reporting capabilities. The **Watching\_History** table serves as a foundational element, providing the raw user interaction data necessary for building comprehensive content consumption metrics in the data mart. This setup ensures that the streaming service can leverage data insights to enhance user experience and optimize content offerings.