Database Design & Development HTTP-5126

Final Retrospective Report

Submitted to: Prof. Nithya Thayananthan

Submitted by: Samson Maveli Mathews, N01751546

Real world scenario

The database is a TV show tracking and ranking system designed to store, manage, and process information related to users' viewing progress and preferences. It keeps detailed records of:

- User profiles and viewing habits.
- Series metadata, including title, genre, total episodes, and production details.
- Ratings from both external sources (e.g., IMDb) and users' upcoming episode ratings.
- Automatically calculated scores for ranking TV shows based on a defined formula.

Database Name and Purpose

Name: TV tracker
Purpose: To manage the tv shows watched by the user

Problems this database solves

- Prevents loss of track for how many episodes a user has watched.
- Eliminates inconsistent data (e.g., next episode ratings for shows already completed).
- Combines multiple factors into a personalized ranking score for each series.
- Replaces manual tracking spreadsheets or notes with automated, queryable records.
- Makes data-driven recommendations possible for users and companies.

Relationships

table_name	***Relationship Type***	***table_name***	***Justification***
user	*one-to-many*	user_series	One user can have many entries in the user_series table
series	*many-to-many*	user_series	Many series can be in multiple user_series tables
company	*one-to-many*	series	One network company can have multiple series

Database Tools

- **View: user_series_ranking_view**
  A view called 'user_series_ranking' that calculates the scores of each tv series based on the episodes they have watched, imdb rating and the rating for the next episode they are going to watch. All the episodes the user has watched will be displayed in the descending order of the score

- **Trigger: trg_update_series_rating**
  A trigger to check if there are any next episodes to watch and if there aren't any next episodes, make the next episode rating as 0

- **Function: fn_get_series_score**
  A function to calculate the series score for a given user and series.
  My current formula for the would be ((watched_episodes/total_episodes)*100) + (imdb_rating * 10) + (next_episode_rating *10))/3

Table Drafts

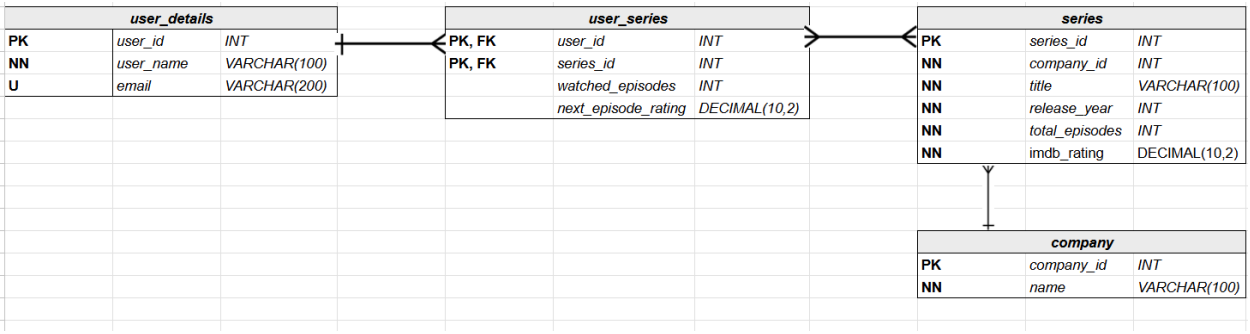
user		
PK	user_id	INT
	user_name	VARCHAR(100)
	email	VARCHAR(200)

series		
PK	series_id	INT
FK	company_id	INT
	title	VARCHAR(100)
	release_year	INT
	total_episodes	INT
	imdb_rating	DECIMAL(10,2)

company		
PK	company_id	INT
	name	VARCHAR(100)

user_series		
PK, FK	user_id	INT
PK, FK	series_id	INT
	watched_episodes	INT
	next_episode_rating	DECIMAL(10,2)

ERD (Entity Relationship Diagram)



Retrospective Section

Feedback

Initially, I only had the **user_series** table and the **series** table. Then after getting feedback from the professor, I added in a **user_details** table, where you store the details of the users. Then, I created

a **company** table, where the details of the network production company that produced the series are stored.

Challenges

Ensuring that user progress, episode details, and series information remain synchronized was a challenge. Handling large data volumes as the number of users and tracked shows grows, optimizing performance for fast queries becomes critical. Different users rate shows differently, which may affect the fairness of the ranking system.

Future Implementation

- Social Sharing – Let users share their rankings and watchlists with friends or fan communities.
- Automated Data Updates – Connect APIs to automatically update show details, episode counts, and ratings.

Conclusion

The TV show tracking and ranking database efficiently manages viewing progress and generates personalized rankings. By automating validation, preventing inconsistencies, and combining multiple rating factors, it enhances user experience and provides valuable insights. Future features like recommendations, mobile access, and social tools could make it indispensable for viewers and streaming platforms alike.