CSC 370: Database Systems

Sprint 1

Summer 2024

By Alyssa Taylor (V00987477) and Karanbir Gosal (V00979752)

Recap: Last week's goals for Sprint 1

Last week we set out a few goals:

- Revise the requirements set in the project kick off
- Create an Entity-Relationship diagram (ERD)
- Come up with some basic tables and commands that would be used by the system

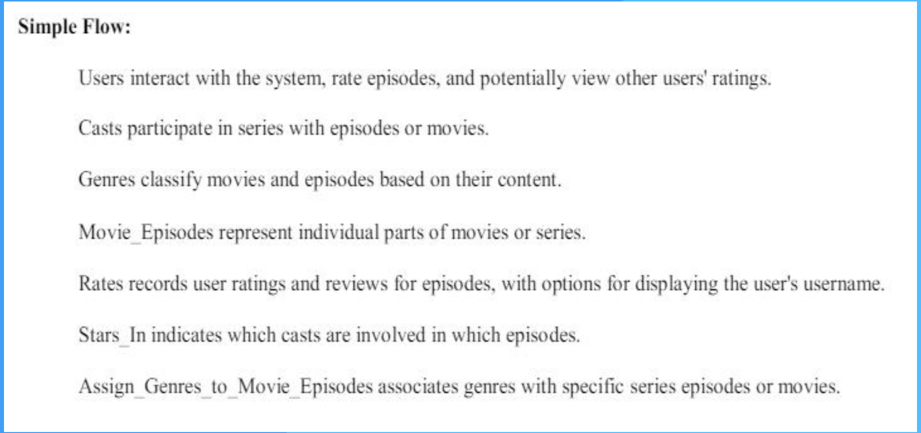
Over the next few minutes we will delve into how we met these goals.

Revised Requirements

From last sprint the requirements did not change significantly. All basic ideas have remained the same and very little was actually edited.

The requirements were looked into with the creation of the ERD as discussed in later slides as they played a large role in deciding what entities, relationships and attributes were needed. They also helped to decide what keys would be needed to better access and search through the database once more data and features are developed and added.

Additionally, the requirements helped to develop a simple workflow for the system as seen to the right.



Simple Flow:

Users interact with the system, rate episodes, and potentially view other users' ratings.

Casts participate in series with episodes or movies.

Genres classify movies and episodes based on their content.

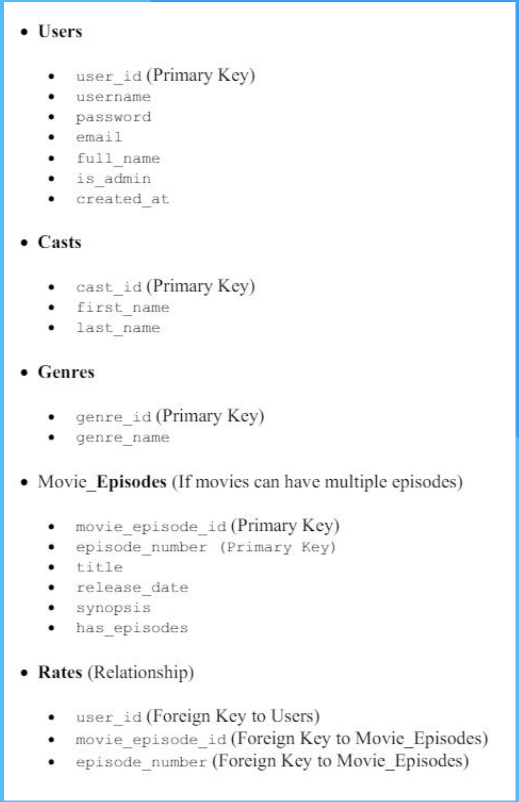
Movie_Episodes represent individual parts of movies or series.

Rates records user ratings and reviews for episodes, with options for displaying the user's username.

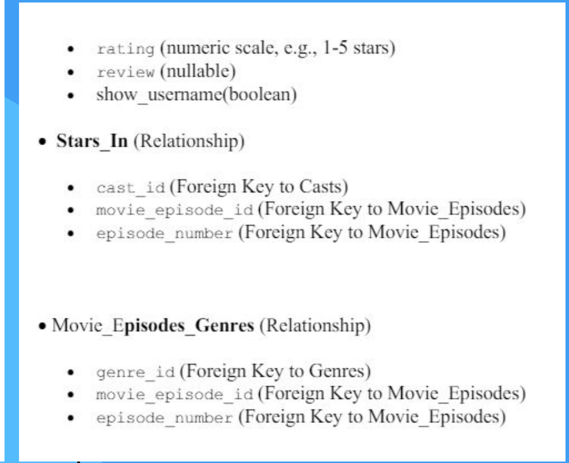
Stars_In indicates which casts are involved in which episodes.

Assign_Genres_to_Movie_Episodes associates genres with specific series episodes or movies.

Entity-Relationship Diagram



- **Users**
  - user_id (Primary Key)
  - username
  - password
  - email
  - full_name
  - is_admin
  - created_at
- **Casts**
  - cast_id (Primary Key)
  - first_name
  - last_name
- **Genres**
  - genre_id (Primary Key)
  - genre_name
- Movie_**Episodes** (If movies can have multiple episodes)
  - movie_episode_id (Primary Key)
  - episode_number (Primary Key)
  - title
  - release_date
  - synopsis
  - has_episodes
- **Rates** (Relationship)
  - user_id (Foreign Key to Users)
  - movie_episode_id (Foreign Key to Movie_Episodes)
  - episode_number (Foreign Key to Movie_Episodes)



  - rating (numeric scale, e.g., 1-5 stars)
  - review (nullable)
  - show_username(boolean)
- **Stars_In** (Relationship)
  - cast_id (Foreign Key to Casts)
  - movie_episode_id (Foreign Key to Movie_Episodes)
  - episode_number (Foreign Key to Movie_Episodes)
- Movie_**Episodes**_**Genres** (Relationship)
  - genre_id (Foreign Key to Genres)
  - movie_episode_id (Foreign Key to Movie_Episodes)
  - episode_number (Foreign Key to Movie_Episodes)

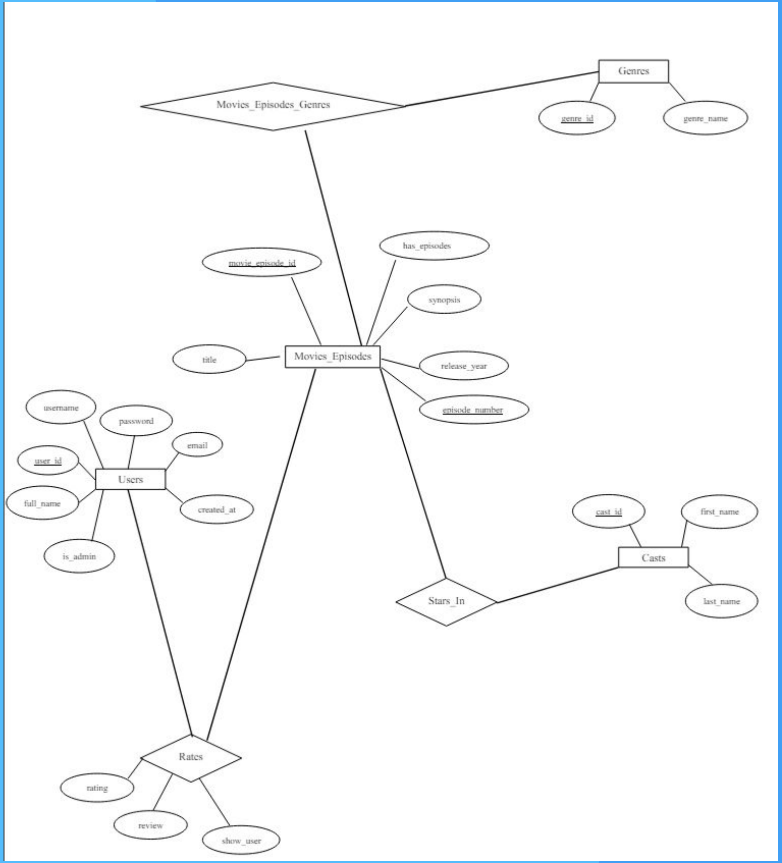
From the revised requirements, the early stages of the ERD diagram began.

Entity sets such as Users, Casts, Genres and Movie-Episodes were decided as a necessity along with their connected relationships of Rates, Stars-in and Movie-Episode-Genres.

Needed attributes/keys were decided upon and assigned to each entity and relationship as well.

Possible foreign keys were also added to guide further development.

Entity-Relationship Diagram



From the Entity sets, attributes list, and relationships, the ERD was created.

This ERD shows the connection between all aspects of the system and can be related to how users would be able to navigate from one data location to another while in use.

For example, a user looking at a specific movie can see the ratings and (if allowed by the reviewer) who made them. Allowing the user to navigate to a reviewers' profile and see what other films they've rated.

Table creation & Basic commands

As seen in the GitHub repository, some SQL code was added for the creation of tables.

Within the “create_tables.sql” file, the tables “users”,”casts”,”genres”,”movie episodes”,”rates”,”stars in” and “Assign_Genres_to_Movie_Episodes” can be now be seen.

Along with the creation of basic database tables, some commands and data were also added into the github project. These include the insertion of some mock data into the tables specified above to demonstrate their intended use.

Plan for next Sprint

We have a few goals for the next 2 week sprint, such as:

- Functional Dependencies (FD’s) & Keys as well as Data Anomalies.
- Apply BCNF Decomposition to normalize the tables.
- Add the referential integrity in the tables like foreign keys.
- Grouping & Aggregation.

THANK YOU