

CSC 370 – Database Systems

Dr. Sean Chester

Movie Rating System

Project Sprint: 1

June 2nd, 2024

Team27J

Alyssa Taylor V00987477

Karan Gosal V00979752

Note: Due to shortage of time in the video, we were not able to showcase running the commands that we have in a separate sql file “create_tables.sql”. However, it covers all from creating tables and all the commands learned in class covering all the content for this sprint.

SQLite0.1.3 beta

Table

Assign_Genres_to_Mo...

Casts

Genres

Movie_Episodes

Rates

Stars_In

Users

MariaDB

PostgreSQL

MS SQL

SQLite

1 -- These are the Sql commands to create all the required tables for the project.Stars_In

2 -- As per the coverage for sprint, the foreign key concept will be covered in the next sprint.

3

4 -- ***** Users Table *****

5

6 CREATE TABLE IF NOT EXISTS `Users` (

7 `user_id` INT PRIMARY KEY,

8 `username` VARCHAR(30),

9 `password` VARCHAR(30),

10 `email` VARCHAR(40),

11 `full_name` VARCHAR(30),

12 `is_admin` BOOLEAN DEFAULT FALSE,

13 `created_at` DATETIME

14);

15

16 INSERT INTO `Users` (`user_id`, `username`, `password`, `email`, `full_name`, `is_admin`, `created_at`)

17 VALUES (1, 'johndoe', 'securepassword', 'johndoe@example.com', 'John Doe', FALSE, '2024-06-01 12:34:56');

18

19 INSERT INTO `Users` (`user_id`, `username`, `password`, `email`, `full_name`, `is_admin`, `created_at`)

20 VALUES (2, 'sidhum', 'testpassword', 'sidhumoose@example.com', 'Sidhu Moose', TRUE, '2024-06-01 12:50:56');

21

22 -- SELECT * FROM `Users`;

23

24 -- ***** Casts Table *****

25

26 CREATE TABLE IF NOT EXISTS `Casts` (

!

SQLite0.1.3 beta

Table

Assign_Genres_to_Mo...

Casts

Genres

Movie_Episodes

Rates

Stars_In

Users

MariaDB

PostgreSQL

MS SQL

SQLite

160 -- Finding all the movies which are not series i.e. no episodes but just a movie

161 -- SELECT * FROM `Movie_Episodes`;

162 -- SELECT * FROM `Movie_Episodes`

163 -- WHERE `episode_number` = 0 AND `has_episodes` = 0;

164

165 -- **** LIMIT and ORDER BY ****

166

167 -- Finding all the movies or series only no episodes and ordering by title

168 */

169 -- SELECT * FROM `Movie_Episodes`;

170 SELECT * FROM `Movie_Episodes`

171 WHERE `episode_number` = 0

172 ORDER BY `title`;

173

174 -- SELECT DISTINCT * FROM `Movie_Episodes`

175 -- WHERE `episode_number` = 0

176 -- ORDER BY `title` LIMIT 1;

177

178 -- **** JOIN ****

179

180 -- SELECT Users.user_id, Users.username

181 -- FROM Users

182 -- JOIN Rates ON Users.user_id = Rates.user_id

183 -- WHERE Users.`user_id` = 1;

184

185

i	movie_episode_id	episode_number	title	release_year	synopsis	has_episodes
2		0	Big Bang Theory: Season 1	2012	The show about the scientists.	1
1		0	Die Hard: 4	2012	Movie about a police officer.	0

In this database schema, we have following tables that represent different entities and their relationships in a system for managing movies or shows, users, casts, genres, and their associations:

- Users:**

The Users table stores information about users who interact with the system.

Attributes include user_id (primary key), username, password, email, full_name, is_admin (indicating if the user is an administrator), and created_at (timestamp of user creation).

- Casts:**

The Casts table represents actors or cast members involved in movies or episodes.

Attributes include cast_id (primary key), first_name, and last_name.

- Genres:**

The Genres table lists different genres that movies or episodes can belong to.

Attributes include genre_id (primary key) and genre_name.

- Movie_Episodes:**

The Movie_Episodes table stores information about individual episodes of movies or series.

Attributes include movie_episode_id (primary key), episode_number (sequential number of the episode within the series), title, release_date, synopsis, and has_episodes (indicating if the movie has multiple episodes).

- Rates (Relationship):**

The Rates table establishes a relationship between users and movie episodes, where users rate and review movies or episodes.

Attributes include user_id, movie_episode_id, episode_number, rating (numeric scale for rating), review (optional review text), and show_username (boolean indicating if the username should be shown with the rating).

- Stars_In (Relationship):**

The Stars_In table establishes a relationship between casts and movie episodes, where casts star in episodes.

Attributes include cast_id, movie_episode_id, and episode_number.

- Assign_Genres_to_Movie_Episodes (Relationship):**

The Assign_Genres_to_Movie_Episodes table establishes a relationship between genres and movie episodes, assigning genres to episodes.

Attributes include genre_id, movie_episode_id, and episode_number.

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.

Some early-stage requirements and tasks that can be achieved using this schema:

User Management

- User Registration: Users can register by providing a username, password, email, and full name.
- User Login: Users can log in using their username and password.
- Admin Management: Certain users can be marked as admins who might have special privileges within the application.

Movie and Series Management

- Adding Movies and Series: Admins can add new movies and series to the Movie_Episodes table.
- Managing Episodes: For series, admins can add and manage multiple episodes, each with its own title, release year, and synopsis.

Casting Management

- Adding Cast Members: Admins can add new cast members to the Casts table.
- Assigning Cast to Movies/Series: Admins can assign cast members to specific movies or episodes through the Stars_In table.

Genre Management

- Adding Genres: Admins can add new genres to the Genres table.
- Assigning Genres to Movies/Series: Admins can assign genres to specific movies or episodes through the Assign_Genres_to_Movie_Episodes table.

User Reviews and Ratings

- Rating Movies/Episodes: Users can rate movies and episodes.
- Writing Reviews: Users can write reviews for movies and episodes, and choose whether to display their username with their review.

Searching and Filtering

- Filter by Genre: Users can filter movies and series by genre.
- Filter by Cast: Users can filter movies and series by cast members.
- Filter by Rating: Users can filter movies and series based on user ratings.

Viewing Content

- List Movies and Series: Users can view a list of all movies and series.
- View Episodes: Users can view episodes for a particular series.
- View Reviews and Ratings: Users can view reviews and ratings for movies and episodes.

Reporting and Analytics (Admin)

- Top Rated Movies/Series: Admins can view reports of the top-rated movies and series.
- User Engagement: Admins can view reports on user engagement, such as the number of ratings and reviews.

Entity-Relationship Diagram (ERD):

