ATS - ATS Tracking System

Team Details:

Gunārs Ābeltiņš
Ričards Bubišs
Renārs Strūve

Business Overview

Imagine that you are a movie enthusiast and you watch a lot of movies. Over time, it can be difficult to remember which movies you have seen and which ones you haven't. This can be frustrating when trying to decide on a new movie to watch, and it can lead to accidentally re-watching movies you've already seen.

To solve this problem, you decide to create a movie-tracking database application. The application allows you to keep track of all the movies you've seen, including the title, director, cast, year of release, genre, and rating. You can also add a short summary or review of the movie, as well as a personal rating or ranking.

With this movie tracking database, you can easily search and filter your movie collection by various criteria, such as genre or rating. You can also see which movies you have already watched, and which ones are still on your watch list. This will make it much easier to decide on a new movie to watch, and you can use the database to keep track of movies you want to watch in the future.

Overall, the movie tracking database application solves the problem of forgetting which movies you have already seen and provides a convenient way to organize and manage your movie collection.

With a movie tracking database, you can not only keep track of which movies you've watched but also generate various statistics from the stored information. Some statistics that can be generated include:

1. Genre breakdown: By analyzing the genre field of each movie, you can generate a breakdown of how many movies you've seen in each genre. This can help you identify which genres you enjoy the most, and which ones you may want to explore more.
2. Director breakdown: By analyzing the director field of each movie, you can generate a breakdown of how many movies you've seen by each director. This can

help you identify your favorite directors, and which ones you may want to watch more movies from.
3. Actor breakdown: By analyzing the cast field of each movie, you can generate a breakdown of how many movies you've seen that feature each actor. This can help you identify your favorite actors, and which movies you may want to watch based on the actors.
4. Rating distribution: By analyzing the rating field of each movie, you can generate a distribution of your personal ratings. This can help you see how many movies you've rated highly, and how many you've rated poorly.
5. Release year distribution: By analyzing the year field of each movie, you can generate a distribution of how many movies you've seen from each year. This can help you identify which decades or time periods you enjoy the most, and which ones you may want to explore more.

By generating these statistics and analyzing the information stored in the movie tracking database, you can gain insights into your movie-watching habits and preferences. You can use this information to make more informed decisions about which movies to watch in the future, and to better understand your own tastes and preferences.

Business Requirements:

1. User Authentication: The movie tracking database application should have a login and registration system that requires users to create an account before they can access the features of the application.
2. Movie Data Input: Users should be able to input movie data including the title, director, cast, year of release, genre, and rating.
3. Movie Data Management: Users should be able to add, edit, and delete movie data as necessary.
4. Search and Filter: Users should be able to search and filter their movie collection by various criteria such as title, director, cast, genre, rating, and year of release.
5. Personalized Rating and Ranking: Users should be able to add their own personal ratings and ranking for each movie.
6. Review and Summary: Users should be able to add a short summary or review of the movie.
7. Watchlist: Users should be able to keep track of movies they want to watch in the future.
8. Statistics Generation: The application should generate various statistics including genre breakdown, director breakdown, actor breakdown, rating distribution, and release year distribution based on the stored information.

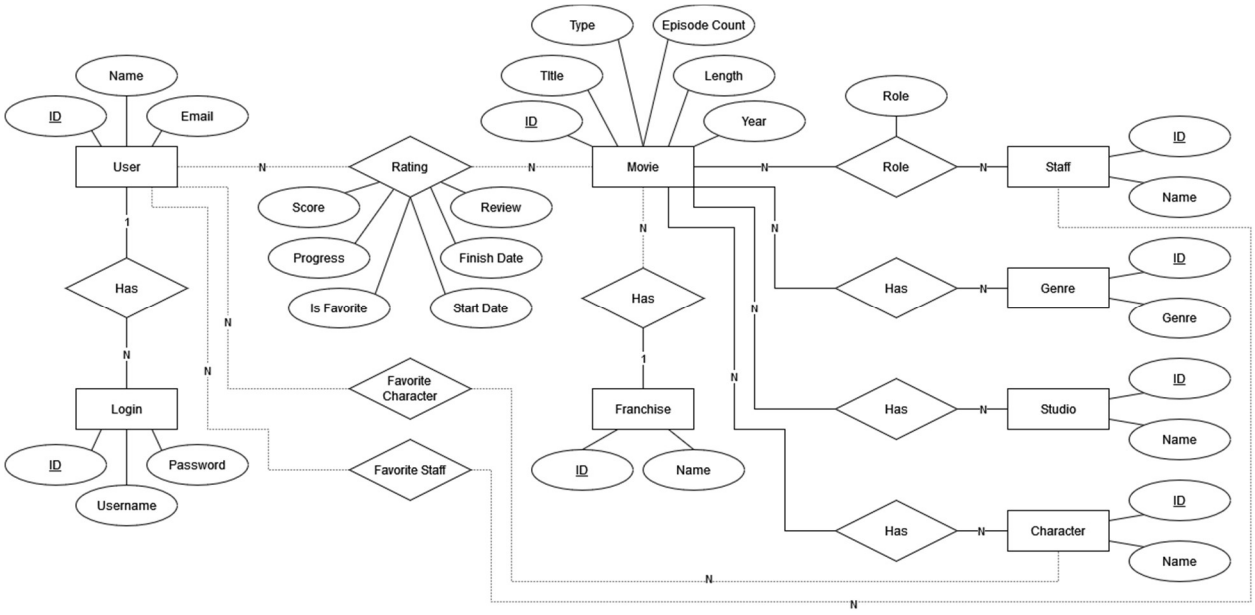
9. Insights: The application should provide insights into the user's movie-watching habits and preferences.
10. User Privacy: The application should be designed with the privacy of user data in mind, ensuring that sensitive user information is not exposed to unauthorized parties.

Business Rules:

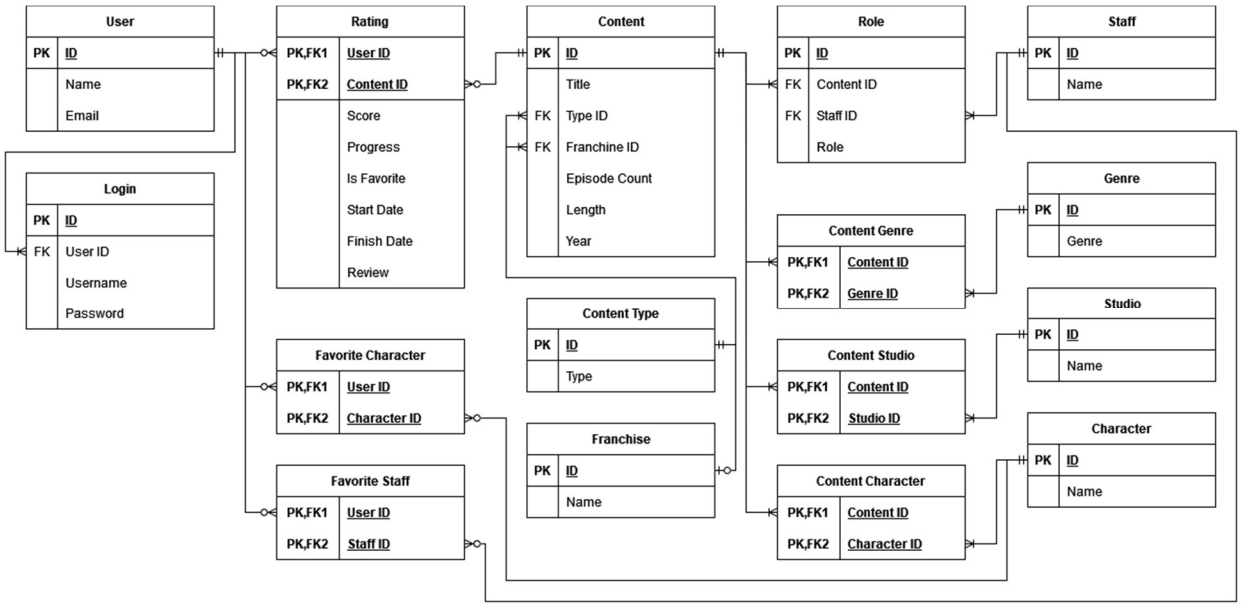
1. Data Validation: The application should validate the movie data input by the user, such as ensuring that the year of release is a valid year and that the rating is within a valid range.
2. Access Control: The application should have different levels of access control for different users, such as allowing some users to only view the movie collection while allowing others to add, edit, or delete movies.
3. Backup and Recovery: The application should have a backup and recovery system in place to prevent data loss in case of any system failures or crashes.
4. User Feedback: The application should provide users with a way to provide feedback or report issues, which can be used to improve the application and fix any bugs or errors.
5. System Security: The application should be designed with security in mind, such as using encryption to protect user data, and implementing measures to prevent unauthorized access or data breaches.
6. Data Ownership: The application should clearly define data ownership, ensuring that users retain ownership of their own data and that the application does not use or share user data without their consent.
7. User Support: The application should provide users with adequate support, such as a help desk or support team, that can assist with any issues or questions that users may have.
8. Compliance: The application should comply with all relevant laws and regulations, such as data privacy laws and regulations governing the use of user data.

Diagrams

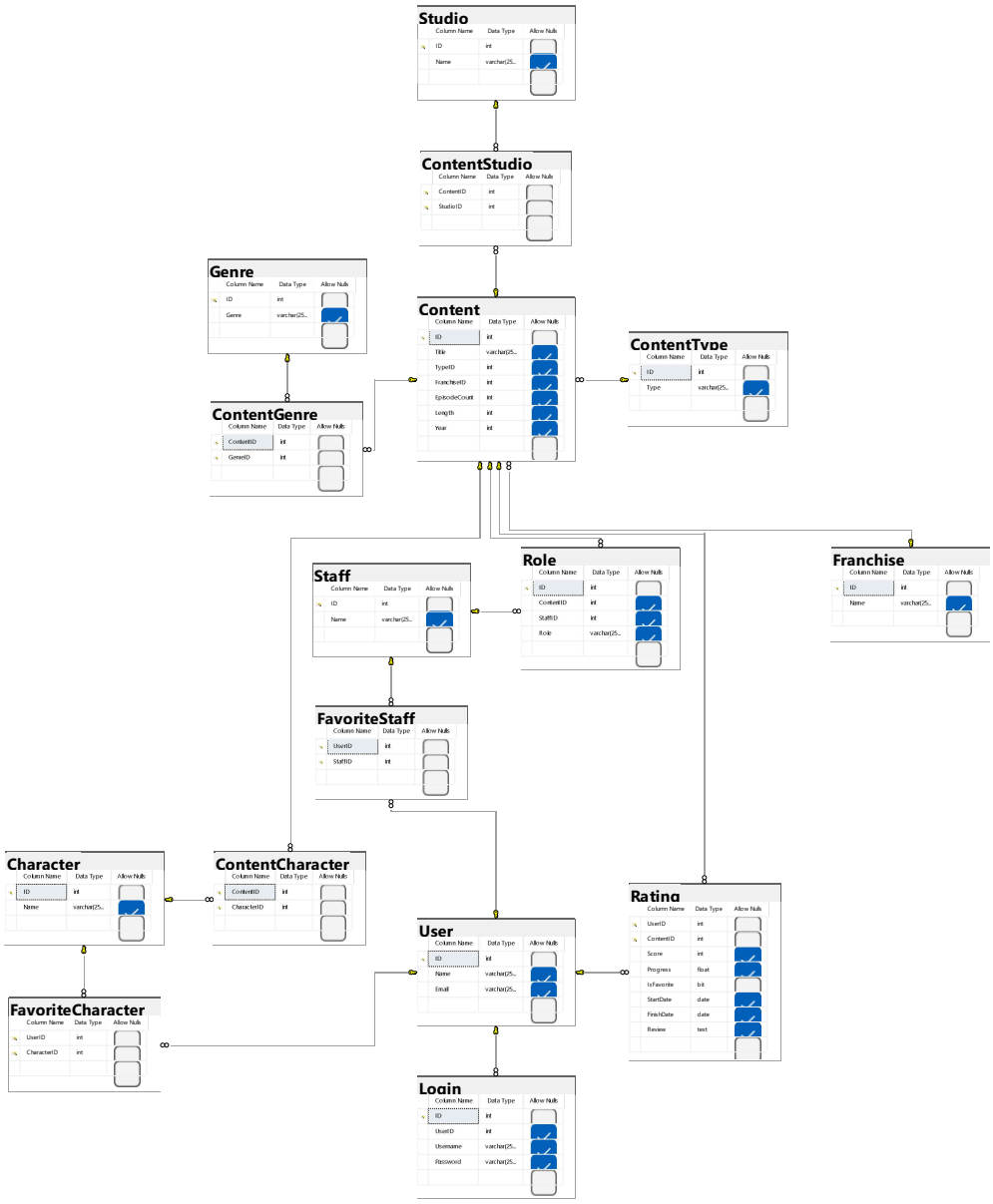
Conceptual model (Chen notation)



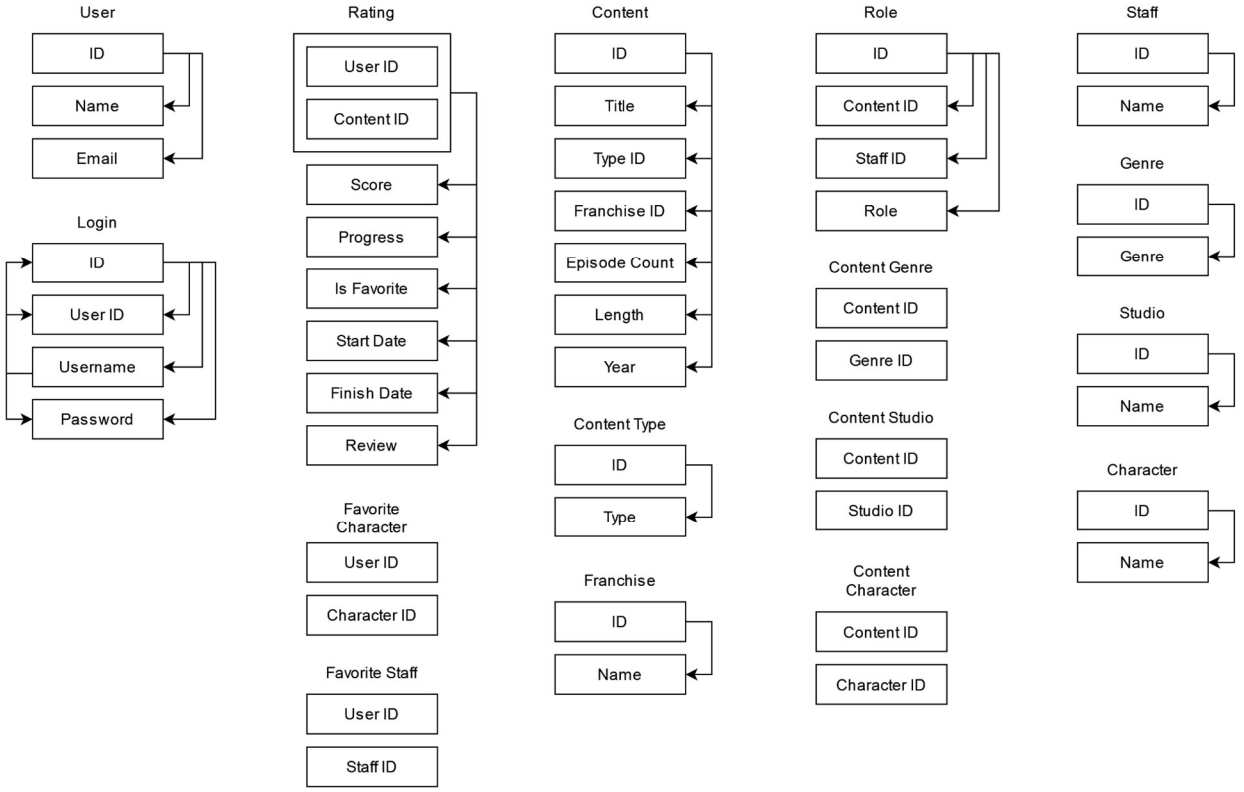
Logical model (Crow's foot notation)



Physical model (SQL Server Management Studio)



Functional dependency diagram



Normalization

- All tables are in the *1NF* form because all fields are atomic.
- All tables have primary keys. Therefore, all tables are in the *2NF*.
- All functional dependencies in all tables start from full primary or potential keys. Therefore, all tables are in *3NF*