Introduction:

This pdf outlines the research we did for this assignment, we also have the functional requirements and explanations on how we implemented some of the ideas in the functional requirements as well as the tasks that needed a written answer.

Research

General Overview of the Entertainment Industry

The entertainment industry, particularly the sectors involving movies and TV series, has undergone significant transformation over the last decade. The traditional dominance of cinema and broadcast television has been challenged by the advent of streaming platforms such as Netflix, Amazon Prime Video, Disney+, and Hulu. These platforms have revolutionized content production, distribution, and consumption, fostering a shift towards on-demand viewing.

Streaming platforms offer a vast array of content accessible worldwide at the click of a button, breaking geographical barriers and expanding audience reach. The global film and video market, which includes movies and television shows, was valued at approximately $234 billion in 2020 and is projected to grow steadily. This growth is driven by increasing digital consumption, international expansion of streaming services, and innovations in content delivery.

The rise of streaming has also influenced content creation, with platforms investing heavily in original productions to attract and retain subscribers. This has led to a diversification of available content, catering to a wider range of tastes and preferences. Additionally, technological advancements such as high-definition streaming, virtual reality, and augmented reality are set to further enhance the viewing experience.

Different Types or Genres of Movies and TV Series

1. Action and Adventure:

Characterized by excitement, physical stunts, and fast-paced narratives. Action films often feature heroes overcoming obstacles and engaging in combat or chases. Adventure films transport audiences to exotic locales with epic quests and explorations. Notable examples include the "Indiana Jones" and "Star Wars" franchises, which blend both action and adventure elements.

2. Comedy:

Utilizes humor to engage audiences, ranging from slapstick and parody to dark comedy and romantic comedies. Comedic films often involve exaggerations, misunderstandings, and satirical takes on various subjects. Classic examples include "Spaceballs" and "The Naked Gun, “and the “scary movie” franchise known for their humorous takes on other film genres.

3. Drama:

Focuses on in-depth exploration of real-life issues, personal relationships, and intense character development. Dramas often delve into emotional and contentious scenarios, providing a profound understanding of characters' motivations and growth. Films like "The Shawshank Redemption" and TV series like "Breaking Bad" exemplify this genre.

4. Horror:

Designed to elicit fear and suspense, horror films may involve supernatural elements, survival scenarios, or psychological thrills. Classics like "The Exorcist" and "A Nightmare on Elm Street" explore dark themes and provoke dread through suspenseful narratives.

5. Science Fiction and Fantasy:

These genres transport audiences to speculative and imaginative worlds. Science fiction often focuses on futuristic technology, space exploration, and scientific advancements. Fantasy involves magical elements, mythical creatures, and ancient lore. "Guardians of the Galaxy" (sci-fi) and "The Lord of the Rings" (fantasy) are quintessential examples.

6. Romance:

Centers on love stories and romantic relationships. These films explore the complexities of love, often culminating in emotional and heartwarming resolutions. Examples include "The Notebook" and "Pride and Prejudice."

There are many film genre that exist and captivate views in fact Yale University’s film studies department reported over 40 film genres, styles, categories and series in their research catalogue.

7. Thriller and Mystery:

Thrillers keep audiences on the edge of their seats with suspense, tension, and unexpected twists. Mysteries involve solving crimes or uncovering secrets. Notable examples include "Inception" (thriller) and "Sherlock" (mystery).

8. Documentaries:

Non-fiction films that explore real-life subjects, events, and people. Documentaries aim to inform and educate viewers on various topics, from history to social issues. Examples include "The Last Dance" and "Planet Earth."

Information on How Content is Rated or Categorized

Content in the film and television industry is rated and categorized to guide viewers on age-appropriateness and content suitability. Different countries have their own rating systems:

1. MPAA (Motion Picture Association of America):

Provides ratings such as G (General Audiences), PG (Parental Guidance), PG-13 (Parents Strongly Cautioned), R (Restricted), and NC-17 (Adults Only) in the United States.

2. BBFC (British Board of Film Classification):

Uses classifications like U (Universal), PG (Parental Guidance), 12A (12 and over, but children under 12 can watch with an adult), 15, and 18 in the UK.

3. TV Parental Guidelines:

Used for TV shows in the U.S., with labels from TV-Y (All Children) to TV-MA (Mature Audience Only).

4. Other International Systems:

Countries like Canada, Australia, and India have their own rating systems reflecting cultural and social standards.

Additional Features and Information

1. Recommendations:

Modern streaming platforms use algorithms to analyse viewing habits and suggest personalized content. This enhances user experience and keeps audiences engaged by offering relevant movies and TV series.

2. User Reviews:

Platforms like Rotten Tomatoes, IMDb, and Metacritic allow viewers to rate and review content. These reviews provide insights into the popularity and quality of movies and series, influencing viewing choices.

3. Behind-the-Scenes Content:

Many platforms offer additional content such as interviews, making-of documentaries, and behind-the-scenes footage, giving viewers a deeper appreciation of the production process.

4. Interactive Content:

Some platforms experiment with interactive storytelling, allowing viewers to make choices that affect the narrative, such as Netflix's "Black Mirror: Bandersnatch."

Functional Requirements for Hoop Streaming Service

1. Content Discovery and Filtering

- **1.1 Search and Discovery**
    - **Requirement**: The application shall provide robust search functionality.
    - **Implementation**: Utilize Elasticsearch to enable efficient full-text search capabilities, enhancing the ability to quickly locate content.
- **1.2 Filtering**
    - **Requirement**: Users shall filter content based on genre, ratings, and release dates.
    - **Implementation**: Implement dynamic SQL queries within the application backend to filter data based on user-selected criteria.

2. Detailed Content Information

- **2.1 Content Details**
    - **Requirement**: Display detailed information for each title.
    - **Implementation**: Use a RESTful API architecture to retrieve detailed content data from the database and present it using responsive UI components on the front end.

3. User Profiles and Social Features

- **3.1 User Profiles**
    - **Requirement**: Users can create and manage profiles.
    - **Implementation**: Implement user authentication and profile management using JWT (JSON Web Tokens) for secure session management.
- **3.2 Rating and Reviews**
    - **Requirement**: Ability to rate and review titles.
    - **Implementation**: Integrate a star-rating system and text review input that updates the backend database in real-time.
- **3.3 Social Sharing**
    - **Requirement**: Share content with friends.
    - **Implementation**: Use social media APIs to facilitate content sharing directly through the platform.

4. Data Utilization and Integration

- **4.1 Data Sources**
    - **Requirement**: Leverage multiple data sources.
    - **Implementation**: Integrate external APIs provided in the spec.
- **4.2 Data Analysis and Curation**
    - **Requirement**: Analyse and curate data effectively.
    - **Implementation**: Use data analytics platforms like Apache Spark for processing and analysing large datasets, ensuring curated and relevant content suggestions.

5. Database Design and Implementation

- **5.1 Database Schema Design**
    - **Requirement**: Efficient relational database schema.
    - **Implementation**: Design and normalize the database schema using tools like MySQL Workbench to ensure efficient data retrieval.
- **5.2 Database Implementation**
    - **Requirement**: Implement and populate the database.
    - **Implementation**: Use SQL scripts to create tables and populate them with initial mock data for testing purposes.

6. Web-based Application Development

- **6.1 Graphical User Interface (GUI)**
    - **Requirement**: Develop an intuitive GUI.
- **6.2 Database Connectivity**
    - **Requirement**: GUI must execute connections to the RDBMS.
- **6.3 Functionality Execution**
    - **Requirement**: Support all user and system requirements.
    - **Implementation**: Ensure comprehensive error handling and unit testing to maintain robust functionality under various scenarios.

7. Application Performance

- **7.1 Performance Requirements**
    - **Requirement**: Effective performance under load.
- **7.2 Reliability**
    - **Requirement**: Reliable data handling.

Sample data

Explanation of Data Sources

The sample data we extracted from the TV maze API is broad and comprehensive, we selected the necessary information to populate certain parts of our relational schema this includes:

- Show Information: Title, Genre, Language, Premiered Date, Status, Summary, and Image URLs.
- Episodes: Episode Number, Season Number, Title, Air Date, Summary, and Image URLs.
- Cast and Crew: Actor Names, Character Names, Roles (e.g., Director, Writer.

We then subdivided the information we extracted into several tables which include:

- Shows
- Seasons
- Episodes
- People
- Cast_credits(for the cast members)
- Crew_credits(for the crew members)
- Update (To see when the latest update to a show was from the Api)
- API requests(this was mainly to see if the API was populating from the selected endpoints)
- Cache table

The reason we created a cache table is because:

Faster Data Retrieval: Cache tables store frequently accessed or computationally expensive data, allowing for quicker data retrieval compared to fetching data from the main tables repeatedly. This reduces the load on the database and speeds up query performance.

Data Entry methods (Script)

Script Method:

To populate the database, we used a script. This means we used a method that automates the data entry process by fetching data from the TVmaze API and inserting it into the database. Here's a general outline of how we did it:

API Requests: We used PHP to send requests to the TVmaze API endpoints.

Data Extraction: Extract relevant data fields from the JSON responses.

Database Connection: We then Established a connection to Ntokozo's' MySQL database using PhpMyAdmin.

Data Insertion: We then wrote SQL queries to insert the extracted data into the appropriate tables.

(Special thanks to Ntokozo for allowing us to use his PhpMyAdmin)

Quality and Relevance of Data

Quality of data

- Accuracy: TVmaze aims to be accurate and complete. The platform continually improves its data to ensure accuracy. In fact, it's considered more reliable than other TV community websites
- Completeness: TVmaze is a comprehensive TV show and web series database that provides a wealth of information for TV enthusiasts. It is not as fully complete as in the documentation of TVmaze and based on reviews of the REST API old shows that are not popular anymore are often removed however, we have noticed how complete and comprehensive TVmaze is when we were using it to populate the data base, it has an extensive array of data for just about everything we needed it to have, for our purposes we chose not do use everything on the REST API.
- Reliability: It is important to note that TVmaze is a community-driven TV guide and REST API, thus any errors reported a swiftly delt with, this also means that the reliability can vary based on factors such as the completeness of its database and the region it covers. In the documentation and based on its community it has coverage in wide region and aims to be as complete as possible with the exception of removing old, unpopular shows as we have already mentioned.
- Timeliness: Data is current and updated. TVmaze's API provides real-time data, which means your database remains current with the latest information.

Relevance of Data:

- Contextual Relevance: For what is what going to be used for we found TVmaze to be more than enough, it had all the relevant and necessary endpoints for our needs and for populating the database.
- Applicability: It was not extremely difficult to populate using Tvmaze as it was well documented and provided the Json, we needed to see what the API would send to the database, the API provided all the necessary information that we needed.

TASK 7

1. Analysis of the Query Execution Plan

Initial Query and Execution Plan

1. **Initial Query:**

   SELECT id, name, genre FROM shows WHERE id > 100;

2. **Execution Plan**:

EXPLAIN SELECT id, name, genre FROM shows WHERE id > 100;

3. **Analyse the Execution Plan:**
    - **id:** 1
    - **select_type:** SIMPLE
    - **table:** shows
    - **type:** range
    - **possible_keys:** PRIMARY
    - **key:** PRIMARY
    - **key_len:** 4
    - **ref:** NULL
    - **rows:** 195
    - **Extra:** Using where

Initial Performance Metrics

- **Execution Time:** the time taken to execute the query was 0.0014s.
- **Rows Examined:** 195
- Showing rows 0 - 24 (391 total, Query took 0.0014 seconds.), we did this 7 times and kept getting 0.0014 seconds.

2. Optimization Strategy: Creating a Covering Index

Explanation of Optimization

- **Objective:** Improve query performance by creating a covering index that includes all columns needed by the query (**id**, **name**, **genre**). A covering index allows the query to be satisfied entirely from the index, avoiding the need to access the table data.
- **Proposed Optimization:**

CREATE INDEX idx_shows_covering ON shows (id, name, genre);

3. Implementing the Optimization

Steps to Implement

1. **Create the Covering Index we use:**

```
CREATE INDEX idx_shows_covering ON shows (id, name, genre);
```

2. **Verifying the New Execution Plan we use:**

```
EXPLAIN SELECT id, name, genre FROM shows WHERE id > 100;
```

4. Post-Optimization Analysis

Post-Optimization Query Performance

- **Execution Time:** Execution time measures now is 0.0013s a marginal improvement.
- **Rows Examined:** The number of rows examined remains 195, which means it accessed all the same rows but marginally faster.
- Showing rows 0 - 24 (294 total, Query took 0.0013 seconds.), we also tested this 7 times to be fair and it consistently came back as 0.0013 seconds.

5. Reporting on Performance Gains

Comparison of Metrics

- **Before Optimization:**
    - Execution Time: 0.0014 seconds
    - Rows Examined: 195
- **After Optimization:**
    - Execution Time: 0.0013 seconds
    - Rows Examined: 195

Explanation of Results

Why Performance Gains Were Observed:

- **Efficient Data Retrieval:** The covering index includes all columns required by the query, allowing the database to retrieve the data directly from the index without accessing the table, reducing the I/O operations.
- **Reduced Latency:** With the data being served directly from the index, the overall execution time decreases.
- **Optimized Query Plan:** The query execution plan shows a more efficient path, leveraging the covering index to minimize data access time.

6. Conclusion

Summary:

- Implementing a covering index on the **shows** table for the query **SELECT id, name, genres FROM shows WHERE id > 100;** marginal improved performance.

- The execution time decreased from 0.0014 seconds to 0.0013 seconds, demonstrating the efficiency of the covering index.
- The covering index allowed the database to serve the query directly from the index, reducing the need to access the main table data and improving query response time.

TASK 2

Original EER Diagram Summary

The original EER diagram included entities such as:

- **Director**, **Production_Studio**, **Games**, **Media**, **Movies**, **Series**, **Reviews**, **Users**, **History**, **Watchlist**

New EER Diagram Summary

The new EER diagram includes the following key entities and relationships:

- **Entities: CACHE**, **API_REQUESTS**, **API_REQUESTS_CACHE**, **API_REQUESTS_SHOWS**, **SEASON**, **SHOWS**, **PEOPLE**, **CREW_CREDITS**, **CAST_CREDITS**, **EPISODES**, **USERS**, **USER_SHOWS**, **LAST_UPDATED**, **UPDATES**
- **Relationships:**
    - Shows have multiple seasons (**HAS** relationship).
    - Shows have multiple episodes (**HAS** relationship).
    - Shows are associated with **CAST_CREDITS** and **CREW_CREDITS**.
    - **API_REQUESTS** are linked to **CACHE** through **API_REQUESTS_CACHE**.
    - **USER_SHOWS** indicates the relationship between users and shows.
    - **LAST_UPDATED** tracks updates to shows.
    - **UPDATES** are linked to **LAST_UPDATED**.

Improvements and Changes

1. **Inclusion of Cache Management:**
    - **Entities Added: CACHE**, **API_REQUESTS**, **API_REQUESTS_CACHE**
    - **Purpose:** These entities help manage and store API responses, reducing redundant API calls and improving performance.
    - **Benefit:** Enhances the efficiency of data retrieval by caching frequent API responses.
2. **Detailed People Management:**
    - **Entity Added: PEOPLE**
    - **Attributes:** Includes details such as **birthday**, **deathday**, **gender**, **country**, **biography**, and **image**.
    - **Relationships:** Links to **CAST_CREDITS** and **CREW_CREDITS**.

- **Benefit:** Provides a detailed and centralized management of people involved in shows, improving data organization and retrieval.

3. **Enhanced Show Management:**
    - **Entities Added: SEASON**, **SHOWS**, **EPISODES**
    - **Attributes:** Includes details like **number**, **premiere_date**, **summary**, **end_date** for seasons; **status**, **premiered**, **officialSite**, **summary**, **language**, **rating** for shows; **name**, **airdate**, **runtime** for episodes.
    - **Relationships:** Better structure with relationships indicating the hierarchy and connections between shows, seasons, and episodes.
    - **Benefit:** Improved clarity and granularity in show data management, facilitating more efficient queries and data integrity.
4. **User Interaction Tracking:**
    - **Entities Added: USER_SHOWS**, **LAST_UPDATED**, **UPDATES**
    - **Purpose:** Tracks which users are following which shows, and logs updates and changes to show data.
    - **Benefit:** Enhances user engagement tracking and ensures users get updates about their favourite shows.
5. **API Request Logging:**
    - **Entities Added: API_REQUESTS**, **API_REQUESTS_SHOWS**
    - **Attributes:** Logs details such as **endpoint**, **request_time**, **status**, **response**.
    - **Benefit:** Helps monitor and analyse API usage and performance, improving the reliability and efficiency of API interactions.

Conclusion

Overall Improvements:

- **Performance Optimization:** The addition of caching entities (**CACHE**, **API_REQUESTS_CACHE**) will reduce redundant API calls, enhancing performance.
- **Data Organization:** More detailed and structured management of shows, seasons, episodes, and people, leading to improved data integrity and query efficiency.
- **User Engagement:** Better tracking of user interactions with shows (**USER_SHOWS**), providing a more personalized user experience.
- **API Monitoring:** Logging API requests and responses helps in monitoring API performance and troubleshooting issues effectively.

References

References for the research

Guttmann, A. (2023). *Entertainment and Media Industry Market Size | Statista*. [online] Statista. Available at: https://www.statista.com/statistics/237749/value-of-the-global-entertainment-and-media-market/.

BFC (n.d.). British Board of Film Classification | British Board of Film Classification. [online] www.bbfc.co.uk. Available at: https://www.bbfc.co.uk/.

IMDb. (2024). *IMDb - Movies, TV and Celebrities*. Retrieved from IMDb: https://www.imdb.com/

Motion Picture Association. (2024, May 9). *MPAA Film Rating System*. Retrieved from mpaa.org: https://www.mpaa.org/

Kerbel, M. (n.d.). *Yale University Library Research Guides: Film Studies Research Guide: Genres, Styles, Categories, Series*. [online] guides.library.yale.edu. Available at: https://guides.library.yale.edu/c.php?g=295800&p=1975072.

References for cache table

Databasejournal.com. (2019). *Database Journal Daily Database Management & Administration News and Tutorials*. [online] Available at: https://www.databasejournal.com/.

References of how good TV maze is

www.tvmaze.com. (n.d.). *TVmaze.com - Your personal TV guide*. [online] Available at: https://www.tvmaze.com/ [Accessed 29 May 2024].