

CS3200 Project Final Report- Movie Entertainment

Group Name: GuptaKRongoneSLeeJ

Group Members: Kinjal Gupta, Stephanie Rongone, Jethro Ronald Lee

README

Steps to run GuptaKRongoneSLeeJ_application.py

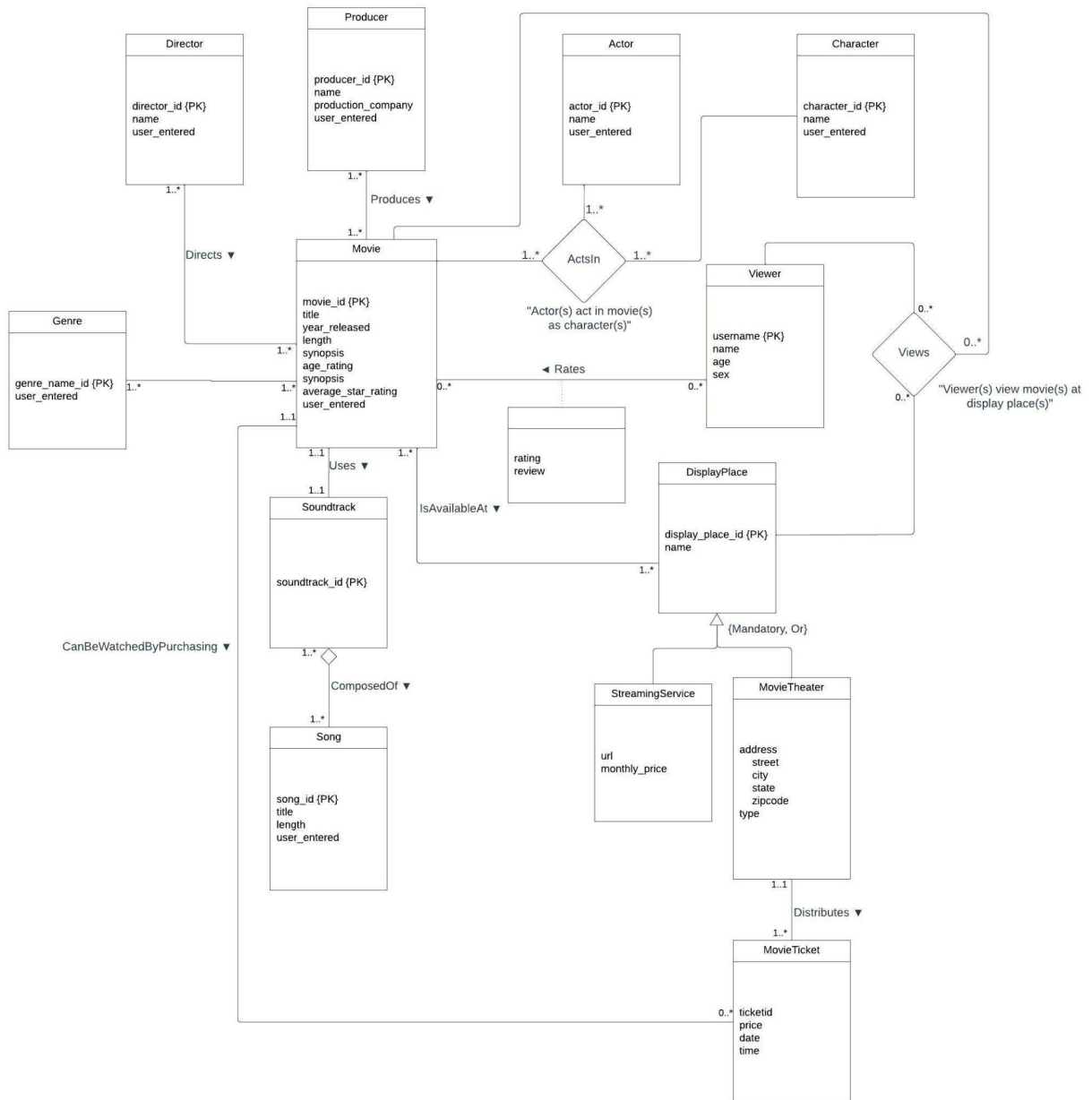
- 1) Download the zip file GuptaKRongoneSLeeJ_project.zip. Unzip the folder.
- 2) Open MySQLWorkbench.
- 3) Log into my MySQLWorkbench. Establish a MySQL connection.
 - a) Link to download MySQLWorkbench:
<https://dev.mysql.com/downloads/workbench/>
- 4) From the downloaded folder, open the following file:
GuptaKRongoneSLeeJ_schemadump.sql in MySQLWorkbench. Execute the script. This should create the “entertainment” schema with all of its procedures and functions.
- 5) Check that one of the latest versions of Python (3.8.9 or newer) is installed on your computer.
 - a) Run the command “Python” in the terminal to determine if it is installed.
If it is not installed, install it.
- 6) Open PyCharm. This is the IDE that will be used to run the application.
 - a) Link to download PyCharm:
<https://www.jetbrains.com/pycharm/download/#section=mac>
- 7) Create a new project or open an existing project in PyCharm.
- 8) In the top toolbar, click “PyCharm”, then “Preferences”. This should open a “Preferences” window.
- 9) On the left side, go to the tab “Project: [project name]”. Under it, select “Python Interpreter.”
- 10) If not installed already, ensure that the following libraries are installed. Click the “+” sign on the window to install the necessary libraries. If these libraries are already installed, skip to step 14:
 - a) PyMySQL; version 1.0.2; Author: yutaka.matsubara
 - b) cryptography; Version 37.0.2; Author: The Python Cryptographic Authority and individual contributors
 - c) matplotlib; Version 3.5.2; Author: Michael Droettboom
- 11) Exit out of the package installation tab.
- 12) Click “OK,” which is located on the blue button in the “Preferences” tab.
- 13) Access the terminal by clicking on “Terminal” at the bottom. Type “Python”, “pip install pymysql”, “pip install cryptography”, and “pip install matplotlib” to double check that the machine being used has all the necessary installments complete.

- 14) On the top toolbar, click “File” then “Open”. Select GuptaKRongoneSLeeJ_application.py”
- 15) Run GuptaKRongoneSLeeJ_application.py in PyCharm.
- 16) If the above steps are done correctly, when the program is run, it should prompt an entry for MySQL username and MySQL password so that a connection to MySQL (“entertainment” schema) can be established. Once the connection is established, the application will run, prompting the user for inputs as needed.

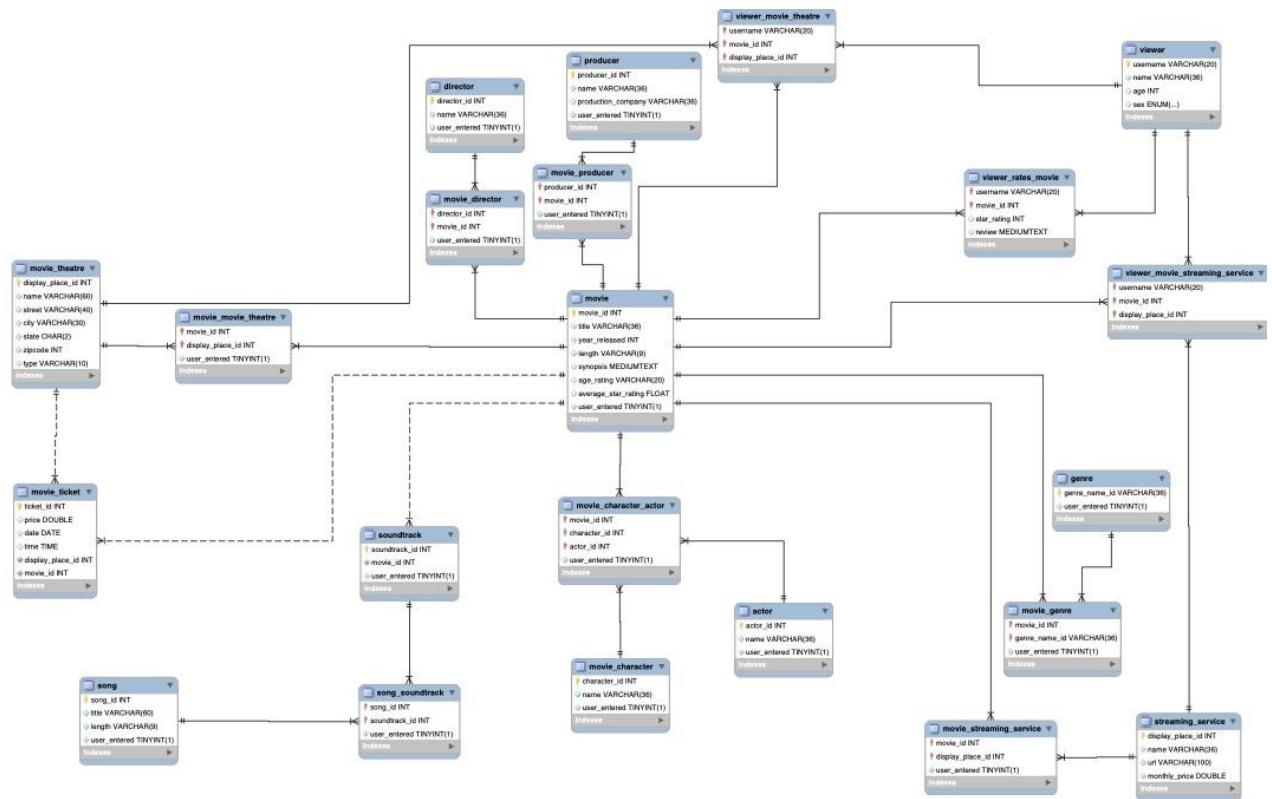
Technical Specifications

This project was created using SQL and Python. Specifically, we utilized MySQLWorkbench to develop the movie entertainment database (using “CREATE TABLE” commands). It was populated with tuples by importing CSV files, which we created, using the “Table Data Import Wizard.” Additionally, we used SQL to develop functions and procedures. These functions and procedures allow for operations to be performed on the database. To allow for front-end functionality, we use the Python-based IDE software called PyCharm. Within our Python-based application, we utilized the PyMySQL library to gain a connection to the MySQL server. This connection then allows us to access our SQL-based database and its functions, procedures, and triggers inside of our Python-based application. Ultimately, this connection allows navigation of the database (including the creation, reading, deletion, and updating of data) in a user-friendly manner. The development of this project required no additional hardware. No machine restrictions were present in the development of this project.

Conceptual Design



Logical Design



Final User Flow of the system

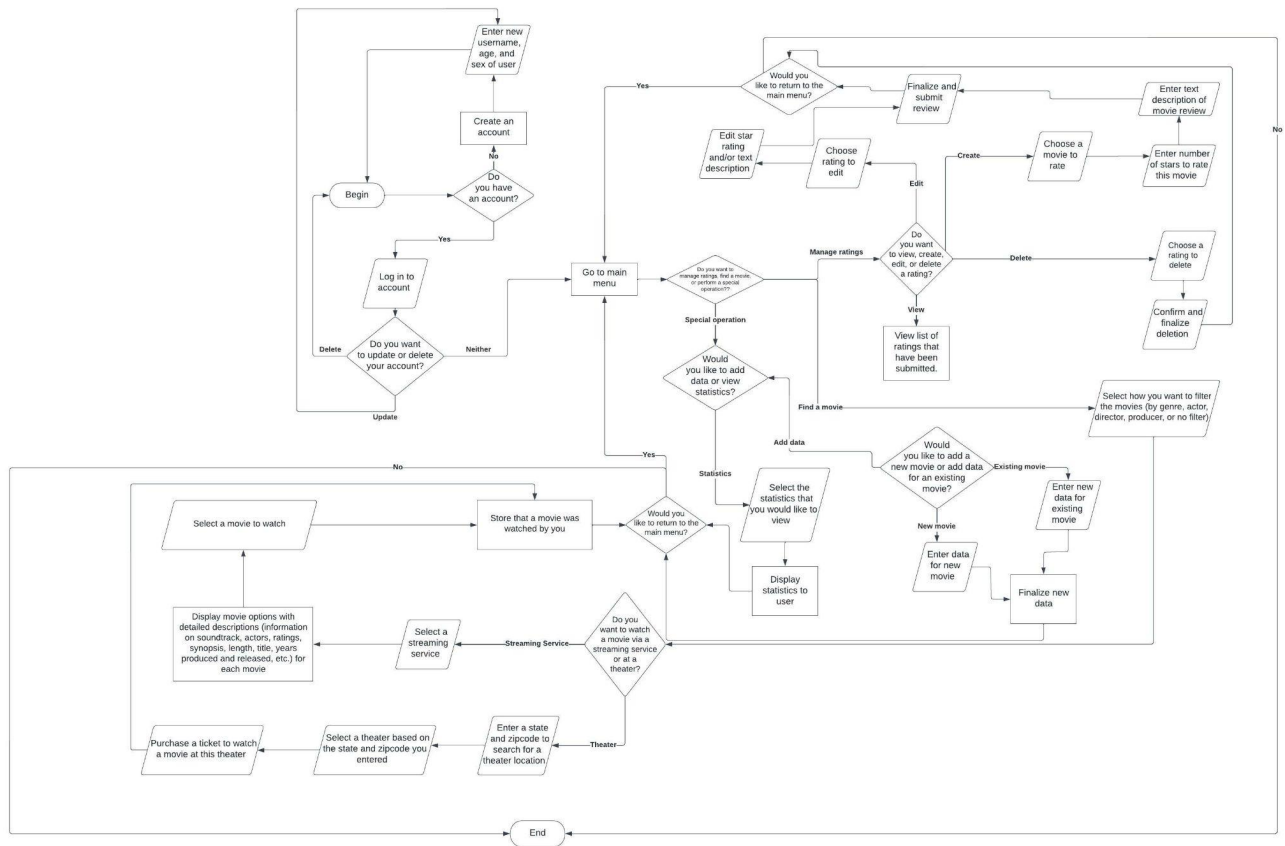
- Enter MySQL username
- Enter MySQL password
- **Log in:** Do you have an account?
 - Yes
 - Enter your account username
 - Enter your account password
 - Do you want to update or delete your account? Or neither?
 - Update Account
 - Enter an account username
 - Enter your name
 - Enter your age

- Enter your sex
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Delete Account
 - Neither
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
- No
 - Create Account
 - Enter an account username
 - Enter your name
 - Enter your age
 - Enter your sex
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
- **Main Menu:** Do you want to rate, find, or watch a movie? Or do you want to perform a special operation? (“Rate”/“Find”/“Watch”/“Other”?)
 - Rate:
 - Do you want to view, create, edit, or delete a review? (“View”/“Create”/“Edit”/“Delete”)
 - View
 - Select a movie to see its ratings
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Create
 - Select a movie you have not rated yet
 - Enter the number of stars you want to give this movie
 - Enter a review or ‘SKIP’ to enter no review
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Delete
 - Select a movie you have rated to delete your rating for that movie
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**

- Log out: Exit out of the application
- Find:
 - Select a genre to use as a filter or ‘SKIP’ if you have no genre preference
 - Select an actor to use as a filter or ‘SKIP’ if you have no genre preference
 - Select a director to use as a filter or ‘SKIP’ if you have no genre preference
 - Select a producer to use as a filter or ‘SKIP’ if you have no genre preference
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
- Watch:
 - Do you want to watch a movie via a streaming service or at a theater?
 - Streaming Service
 - Select a streaming service you want to use
 - Choose a movie from the list of movies available at this streaming service
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Theater
 - Select a theater you want to watch at
 - Select a ticket available at this theater
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
- Other:
 - Do you want to enter data or review interesting statistics?
 - Enter
 - Do you want to enter data for an existing movie, enter data for a new movie, or add a new genre?
 - New Movie
 - Enter the title of the movie you wish to add
 - Enter the release year of the movie you wish to add
 - Enter the synopsis of this movie
 - Choose a theater that is showing this movie, or type “SKIP” if you don’t wish to

- Choose a streaming service that is showing this movie, or type 'SKIP' if you don't wish to
 - Add at least one character for this movie
 - Add at least one producer for this movie
 - Add at least one director for this movie
 - Add at least one genre for this movie
 - Add at least one song for this movie
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
- Existing movie
- Enter the id of an existing movie in the database
 - Choose a new theater that is showing this movie, or type "SKIP" if you don't wish to
 - Choose a new streaming service that is showing this movie, or type 'SKIP' if you don't wish to
 - Do you wish to add new characters?
 - Yes: Add at least one character for this movie
 - No: Skip
 - Do you wish to add new producers?
 - Yes: Add at least one producer for this movie
 - No: Skip
 - Do you wish to add new directors?
 - Yes: Add at least one director for this movie
 - No: Skip
 - Do you wish to add new genres?
 - Yes: Add at least one genre for this movie
 - No: Skip
 - Do you wish to add new songs?
 - Yes: Add at least one song for this movie
 - No: Skip
 - Continue to the main **menu** or log out?

- Continue: Return to **main menu**
 - Log out: Exit out of the application
- New Genre:
 - Enter a new genre to add
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
- Stats
 - Do you want to find the top five movies by genre, find the average age of viewers for a movie, find the five most popular actors today, find the top five movies today, or find the average sentiment rating of all the comments for a certain movie while also generating a pie chart illustrating the overall sentimentality of those comments?
 - Top five movies by genre
 - Choose a genre you want to find the top five movies for
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Average age of viewers for a movie
 - Select a movie to find the average age of the viewers for it
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Top five most popular actors today
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Top five movies today
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application
 - Sentiment analysis
 - Select a movie for the application to perform sentiment analysis on
 - Continue to the main **menu** or log out?
 - Continue: Return to **main menu**
 - Log out: Exit out of the application



Lessons Learned

Technical Expertise Gained

This project helped us develop various technical skills that will assist us in our future professional endeavors. We previously had little to no experience with MySQL, so this project helped us to become more familiar with the language. While relational database systems are not perfect, they are still a strong candidate for information storage, making SQL a reputable language for so many companies. Throughout the course we learned various SQL techniques including creating, reading, updating, and deleting data from a relation; using procedures, functions, triggers, and prepared statements; taking advantage of transactions; creating foreign key and data domain constraints; creating a database dump; creating a logical schema; and more. We were able to apply many of these techniques within our project and gain insight into their purpose in a real-world and front-end-based application. It was also interesting to learn how Python, a language we are familiar with, could be used on the front-end of an application while supported by SQL code on the back-end to develop a robust user-friendly application. Particularly, we learned to take advantage of the procedures we create on the back-end and call them on

the front-end using cursors. SQL appears to have limited functionality as the basis for the front-end of an application, but its role in managing a database in the back-end emphasizes its value. We also learned how to take greater advantage of the data import wizard as a way to insert data into the relations we created on MySQL Workbench. However, we did run into some issues that we did not anticipate. *We extracted all of the data for this project manually and created spreadsheets with Google Sheets, which we then downloaded as CSV files.* However, we kept getting error messages stating that there were problems with the 'ASCII' ordinal not being in range(128). Hence, we had to alter our data to avoid receiving that message, ensuring that every character in our data could be correctly interpreted by the data wizard. To deal with error handling and invalid data input, we learned to implement try except statements in Python. We also implemented error handling in SQL. For example, we handled errors in SQL in which a user tries to insert data that already exists in the database. We also implemented error handling for a case in which a user, who is required to input a value in our database, enters a value that does not exist. We also gained familiarity with sentimental analysis, particularly for the rating comments in our database. In summary, we generated a list of positive and negative words that our application uses to determine the sentiment score of a rating. Based on those scores, the comments are classified as positive, neutral, or negative. The distribution of these classifications for a certain movie is shown on a pie chart.

Insights, Time Management Insights, Data Domain Insights, etc.

Especially with how the time we had to work on the project was already limited due to us taking CS3200 in a shortened summer semester, we struggled to prioritize building our application based on our current ideas over thinking of the broader capabilities that we could incorporate during the building process. Working on this project has been analogous to a pile of snow being shoveled away as more snowflakes keep accumulating on it. We had to become comfortable with sacrificing big aspirations we have for the project for the sake of the basic functionality of certain elements. We also learned that having a concrete plan on how to execute our project before building the project was very beneficial. Devoting two days to creating pseudocode for our project helped us focus on building our application for an extended period of time without worrying about missing scenarios and cases. We also think that we would have benefited from collecting our data earlier so we could test out bits of code as we worked on our project. We had to instead code as much as possible without testing it until we finished collecting our data. Hence, most of the debugging for this project was done near the project's due date, which is not good coding practice as coding is iterative, but we had to do what we could with the limited time and resources we had available. Understanding what we could feasibly do for this project based on the resources we currently have was a crucial skill that got us through the project. For instance, a week before we became

comfortable with using PyMySQL to connect with MySQL via PyCharm, we decided to devote time to creating the tables in the MySQL workbench and creating pseudocode for our project. Finally, a big barrier we encountered for this project was the volume of data we didn't realize we needed to collect and the fact that some data was harder to obtain than others. Some data, such as the production completion year of a movie, was too hard to find and was initially going to be incorporated into the database but was later removed due to our time constraints. We also realized that it would take too much time to collect ALL actors, characters, songs, genres, theater showings, and streaming service showings that pertain to a movie with the number of movies we had to collect data for. Hence, we restricted the amount of data we collected for the sake of completing our project on time and avoiding human errors.

Realized or Contemplated Alternative Design/Approaches to the Project

If we had more familiarity with NoSQL databases earlier in the semester, we likely would have benefitted from them a lot. The rules of normalization generally aligned with our project interests, but some rules were more restrictive than helpful. For example, it would have been easier to store the artists for a song as a set since a song could have multiple artists. Otherwise, we would have to create a whole new relation for artists, but for the sake of time, we had to delete the artist field instead. If we stored the artists as a set, our "song" relation would not even be in first normal form. We personally think that a graph database would better have served our interests. Graph databases can provide certain insights that relational databases cannot, such as how much one actor typically works with another, what types of genres a director is usually associated with, how frequently a producer decides to create fictitious characters, etc. Each different type of entity (e.g. actor, title, producer, director, etc.) could be represented as nodes with different colors. The thickness of the edges between the nodes would indicate how strong of a relationship is possessed between two nodes. An edge can be labeled with attributes to highlight the data that needs to be tracked for a particular relationship. We also realize that with relational database management systems, if we decide to enhance our application in the future, we may need to download more libraries, which would make managing the database very expensive. Finally, movies are going to keep being produced for many years, probably at increasing rates. NoSQL databases are designed to manage big data increasing in volume, variety, and velocity, but RDBMS are not. Other benefits of NoSQL databases such as graph databases that would have been helpful include the ability to easily provide data across many hosts, the lack of need for high management, its inexpensive commodity servers to track data and transaction volumes, and that using a NoSQL database is cheaper per gigabyte or transaction/second.

Future Work

Planned Uses of the Database

As life gets more entailing, many people find comfort in savoring precious times in their lives when they can relax by watching a movie. Unfortunately, especially for indecisive or picky people, the process of choosing what movie to watch can be so stressful that it gives them even more unneeded stress than they already have. Our project aims to have people avoid that stress by enabling a user to find a movie they want to watch with the help of some filters, ratings, and special queries. A movie may pertain to a user due to a variety of factors that we considered for this project: its director, producer, actor(s), character(s), genre(s), soundtrack, and the way it is displayed (at a theater or streaming service). Hence, the main planned use of this database is to help users find movies that capture their interests with the help of the database's stored information. However, the database also serves other purposes. It can help provide a list of theaters that allows a user to see a movie they want to watch as quickly as possible. It also lists the movies available at a streaming service that a user may use. The user can ultimately select a movie to watch. They can even rate movies and see others' ratings for movies if they are curious. By containing so much information about a movie, the project also allows users to appreciate what makes a movie so compelling, especially in terms of factors behind the scenes. Usually, most people judge a movie based on upfront factors such as genres, actors, and characters. Knowledge of elusive factors such as the producers and directors of a movie, which may motivate a watcher to view more of certain producers' and directors' works, helps people better appreciate every aspect of what makes a movie whole. Our project even allows users to compare their movie tastes with others, which could perhaps encourage users to expand their movie interests based on the current trends. We believe this functionality can even bring the movie-watching community closer together as they find others who have similar movie tastes. Particularly, users can find the top five movies by genre, the average age of viewers for a movie, the top five rated movies in the database, the top five actors, the average sentiment rating of all the comments for a certain movie, and generate a plot illustrating the overall sentimentality of the comments for a movie on a graph.

Potential Areas for Added Functionality

If we had more time with this project, we definitely would have given users a "back" function so they could return to the previous command of our application. However, with the limited amount of time we had available, we had to resort to prompting users whether they want to continue to the main menu or log out of our database application after performing an operation. The navigation complications

associated with a “back” function also unmotivated us to add it to our project. We also would have enabled users to filter multiple factors of the same category at a time (e.g. multiple actors, directors, songs, etc). Unfortunately, this feature was too difficult to add with the limited time we had. We initially thought of allowing users to store multiple filters as a list in Python and passing that list as an array parameter for our movie filter procedure in MySQL. We also may have allowed users to suggest new genres, actors, directors, producers, songs, and other movie factors after they receive an error message stating that the characteristic they are trying to add as a filter does not exist. Another complicated feature we were thinking about adding was creating a website or graphical user interface for our project. We think that our application is easy to navigate through, but a command line interface does not provide the aesthetic appeal we hoped for when building this project. A website would contain visual elements that would provide users with a pleasing experience when navigating through our application. In addition, as we mentioned before, our application could be converted to a NoSQL database. Especially considering how the volume of film data will keep evolving, a relational database management system will not be feasible to use for our application in the long run. We also would definitely need a better way to extract all of our data. There are so many movies, actors, producers, songs, streaming services, theaters, etc. that our database does not even include. We don’t even include every single characteristic of a movie in our database. Some missing key factors include production costs, the location it was filmed, and costume designers. We would probably need to form a technical team in charge of importing that large volume of data to our database. Furthermore, a lot of our commands require that users enter an id indicating the entity they want to reference (for example, director id, producer id, song id, etc.) However, a website or GUI could provide images that would provide a better representation for those entities over the ids they are associated with. We were thinking of having users reference certain entities with their names, but due to the risk of actors, producers, songs, characters, and directors having the same name, we thought it would be unwise to do so. We also think that enabling users with more customization features such as suggesting new theaters and streaming services would be a good consideration for the future. Implementing more statistical operations for the viewers is a task to look into as well.

Additional Notes

- The sentimental analysis section of our Python-based code was extracted from Jethro’s previous work in DS2000 and adjusted to fit the needs of our application.
- Some data about movie tickets was made up randomly in our CSV files pertaining to movie tickets to give variety to the data.
- The CSV files were created by us and all the data was inserted into them by us. We navigated through several sites to create the data and thus populate the CSV files.

Sources

Images in the Presentation Given in the Video:

CRUD Operations:

- [Create Image](#)
- [Read Image](#)
- [Rating Image](#)

Project Architecture:

- [PyCharm Image](#)
- [pymysql Image](#)
- [Matplotlib Image](#)
- [MySQLWorkbench Image](#)

General:

- [Icons and Presentation Template](#)

CSV Files' Data:

Finding Streaming Services, Movie Information, Starring Actors, and Directors:

- [Netflix](#)
 - [Netflix Pricing](#)
- [Hulu](#)
 - [Hulu Pricing](#)
 - Movies:
 - [Running With the Devil](#)
 - [The Boss Baby: Family Business](#)
 - [The Devil Wears Prada](#)
 - [I Trapped the Devil](#)
 - [Last Survivors](#)
 - [Think Like A Man](#)
 - [100% Wolf](#)
 - [Sea Fever](#)
 - [Savage State](#)
 - [The Tiger Rising](#)
- [Amazon Prime Video](#)
 - [Amazon Prime Video Pricing](#)
 - Movies:
 - [Running With the Devil](#)
 - [The Boss Baby: Family Business](#)
 - [The Devil Wears Prada](#)
 - [I Trapped the Devil](#)
 - [Last Survivors](#)
 - [Think Like A Man](#)

- [100% Wolf](#)
- [Sea Fever](#)
- [Savage State](#)
- [The Tiger Rising](#)
- [Wikipedia](#)
 - Movies:
 - [The Takedown](#)
 - [Senior Year](#)
 - [Miss Americana](#)
 - [The Privilege](#)
 - [The Adam Project](#)
 - [The Kissing Booth](#)
 - [A Whisker Away](#)
 - [The Weekend Away](#)
 - [The Power of the Dog](#)
 - [Black Crab](#)
 - [Everything Everywhere All At Once](#)
 - [The Bad Guys](#)
 - [Top Gun: Maverick](#)
 - [Crimes of the Future](#)

Finding Cast (Actor, Character):

- [TMDb](#)
 - Movies:
 - [Running With the Devil](#)
 - [The Boss Baby: Family Business](#)
 - [The Devil Wears Prada](#)
 - [I Trapped the Devil](#)
 - [Last Survivors](#)
 - [Think Like A Man](#)
 - [100% Wolf](#)
 - [Sea Fever](#)
 - [Savage State](#)
 - [The Tiger Rising](#)
- [Wikipedia](#)
 - Movies:
 - [The Takedown](#)
 - [Senior Year](#)
 - [Miss Americana](#)
 - [The Privilege](#)
 - [The Adam Project](#)

- [The Kissing Booth](#)
- [A Whisker Away](#)
- [The Weekend Away](#)
- [The Power of the Dog](#)
- [Black Crab](#)
- [Everything Everywhere All At Once](#)
- [The Bad Guys](#)
- [Top Gun: Maverick](#)
- [Crimes of the Future](#)

Finding Producers:

- [IMDB](#)
 - Movies:
 - [Running With the Devil](#)
 - [The Boss Baby: Family Business](#)
 - [The Devil Wears Prada](#)
 - [I Trapped the Devil](#)
 - [Last Survivors](#)
 - [Think Like A Man](#)
 - [Sea Fever](#)
 - [Savage State](#)
 - [The Tiger Rising](#)
- [100% Wolf](#)
- [Wikipedia](#)
 - Movies:
 - [The Takedown](#)
 - [Senior Year](#)
 - [Miss Americana](#)
 - [The Privilege](#)
 - [The Adam Project](#)
 - [The Kissing Booth](#)
 - [A Whisker Away](#)
 - [The Weekend Away](#)
 - [The Power of the Dog](#)
 - [Black Crab](#)
 - [Everything Everywhere All At Once](#)
 - [The Bad Guys](#)
 - [Top Gun: Maverick](#)
 - [Crimes of the Future](#)

Finding Songs in Each Movie:

- [Tunefind](#)

- [Running With the Devil](#)
- [The Boss Baby: Family Business](#)
- [The Devil Wears Prada](#)
- [Last Survivors](#)
- [Think Like A Man](#)
- [Sea Fever](#)
- [I Trapped the Devil](#) (I Trapped the Devil)
- [The Tiger Rising](#) (The Tiger Rising)
- [100% Wolf](#) (100% Wolf)
- [Savage State](#) (Savage State)
- [The Takedown](#) (The Takedown)
- [Senior Year](#) (Senior Year)
- [The Privilege](#) (The Privilege)
- [A Whisker Away](#) (A Whisker Away)
- [Spotify](#) (For every other movie)

Movie Synopsis:

- [Wikipedia](#)
 - Movies:
 - [The Takedown](#)
 - [Senior Year](#)
 - [Miss Americana](#)
 - [The Privilege](#)
 - [The Adam Project](#)
 - [The Kissing Booth](#)
 - [A Whisker Away](#)
 - [The Weekend Away](#)
 - [The Power of the Dog](#)
 - [Black Crab](#)
 - [Everything Everywhere All At Once](#)
 - [The Bad Guys](#)
 - [Top Gun: Maverick](#)
 - [Crimes of the Future](#)

Movie Theater, Tickets:

- [Showcase Cinemas Quacker Lane](#)
- [Rustic Tri Drive-In](#)
- [AMC Braintree](#)
- [Regal Colonie Center](#)
- [Hollywood Drive-In Theater](#)