

CS 5200-02 – Database Management Systems Final Report

Group Name:
LeeK

Kristen Lee

README Section

Below there is a link to a README.md file for this project. This file can also be found in the project structure submission in the “res” folder.

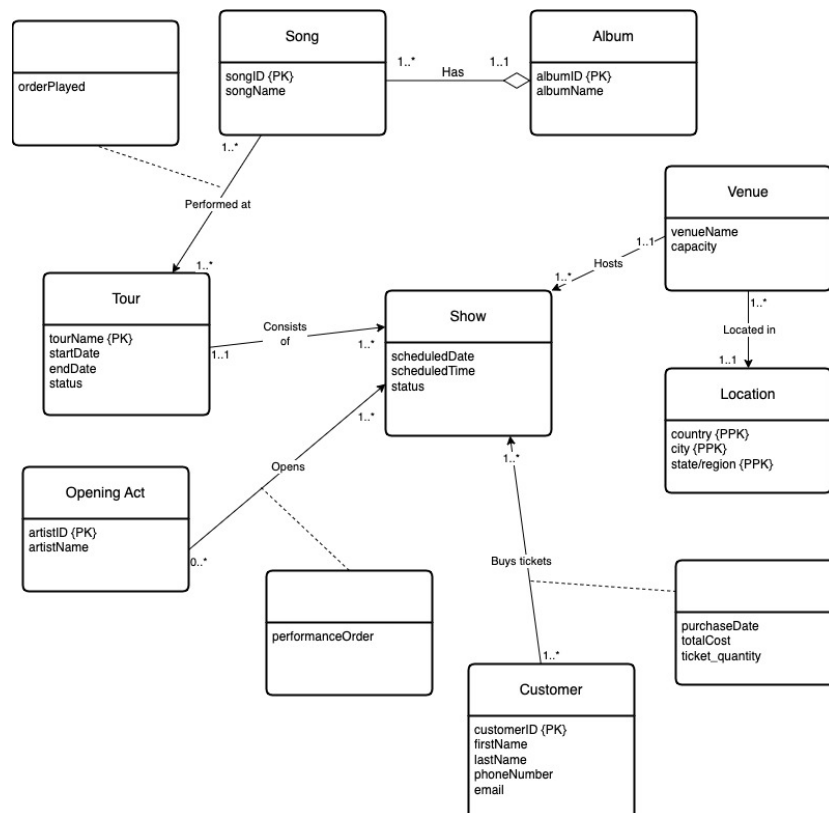
[README.md](#)

Technical Specifications

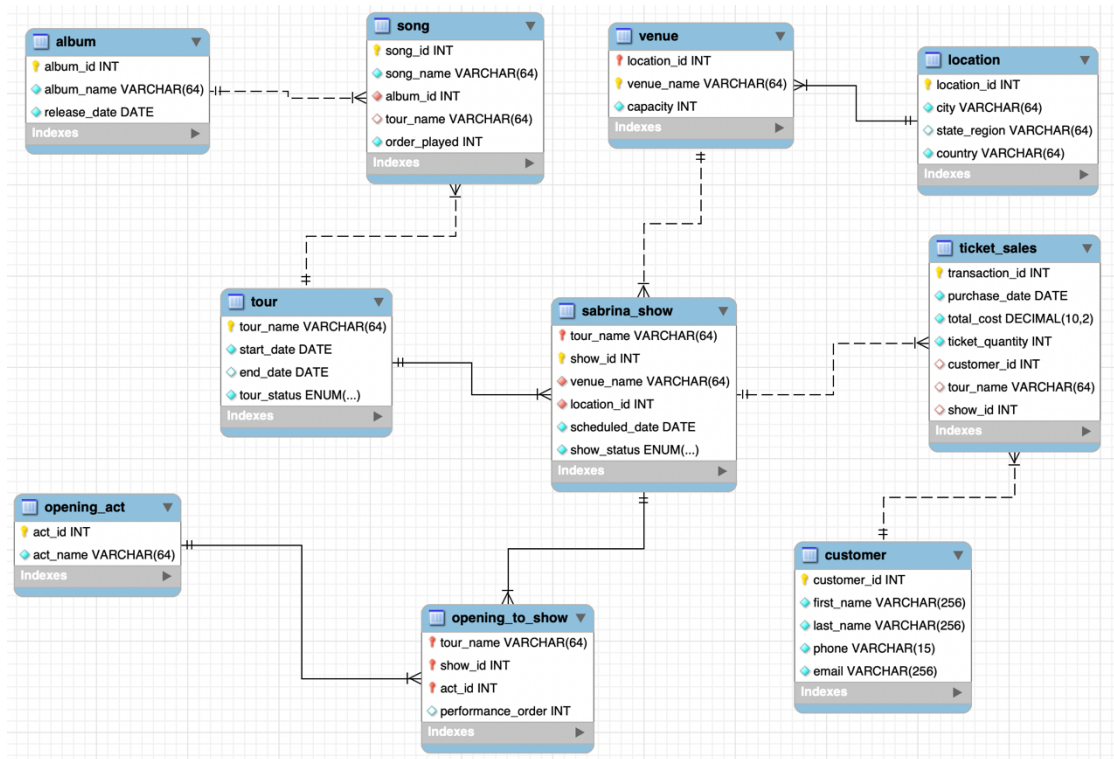
To build this project, I used java 11 as the host language in the IDE (integrated development environment) IntelliJ IDEA. For building the relational database I used the latest versions of MySQL server and MySQL Workbench. Download pages and instructions can be found in the README. Below I list the libraries and frameworks that I used in building this database application

- MySQL Connector/J: JDBC Driver to connect to MySQL
- Java Swing: GUI framework used for the application's user interface

Current UML Diagram



Logical Design



Final User Flow

Here are the commands/methods the user performs to interact with the application:

- Login page
 - o This is where credentials will be validated (for our purpose their credentials are those for MySQL Workbench)
 - o They can type username and password and then click the login button
 - o If successfully logged in, they will see a success message and the main window will come into view
 - o If unsuccessful, there will be an error message, and the user will be able to try again
- “Show Table”
 - o After clicking, user will see pop-up window with dropdown menu of tables
 - o Select table from the drop down and click ok once satisfied with selection
 - o Selected table will populate table environment
- “Update Tuple”
 - o After clicking, user will see pop-up window with dropdown menu with tables
 - o There are also text fields in the pop-up window environment that correspond with the selected table
 - These fields are the attributes for tables one of them being the unique identifier(s) for the tuple the user wishes to update
 - User must input an identifier

- User can leave some fields blank if they only wish to update some, and not all, fields in a tuple
 - User can input desired information
 - Once satisfied click ok
 - If successful, user will see success message and table with updated tuple will populate table environment
 - If unsuccessful, user will see appropriate error message and have the opportunity to try again
- “Add Tuple”
 - After clicking, user will see similar pop-up window as that seen with the update button
 - Drop down menu with tables
 - Text fields that represent the selected table’s attributes
 - Text fields correspond to table that is selected
 - User can input desired information
 - Once satisfied, user can click ok
 - If successful, user will see success message and an updated table will be shown
 - If unsuccessful, user will see appropriate error message and have the opportunity to try again
- “Delete Tuple”
 - After button is clicked, user will see similar pop-up window as seen in update and add
 - Text fields will correspond to primary and composite primary key attributes that correspond to the selected table
 - These are required for the deletion of a tuple
 - Once satisfied, user can click ok
 - If successful, user will see success message and table will be shown with the chosen tuple deleted
 - If unsuccessful, user will see appropriate error message and have the opportunity to try again
- “Future Shows”
 - Once clicked, user will see table in the environment showing Sabrina’s upcoming shows
 - They will see attributes such as tour name, scheduled date, venue and location information, and seat sales and availability
- If the user wishes to exit the program, they can simply use the typical exit button

Lessons Learned

This project was a valuable learning experience, offering opportunities to refine technical skills, enhance project management skills, and explore alternative design approaches. This has served as a platform to apply classroom concepts into a real-world context.

Throughout this project, I was able to gain a deeper knowledge of managing SQL procedures and events. In addition, I improved my ability to write efficient queries and manage database transactions. This semester, I was enrolled in another class that uses Java, and we just finished an assignment using a graphical user interface. This database application allowed me to enhance my

previous knowledge and skills of designing and implementing user interfaces using Java Swing. This project required that we develop a well-structured database schema to support the application. This enhanced my understanding of classroom concepts such as normalization, primary and foreign key constraints, and referential integrity. In completing this project and Homework 8, I learned how to successfully establish and manage a connection between a MySQL database and host languages such as Java and R.

We were given the opportunity to choose groups for this project. Groups of one, two, or three. I chose to work independently because throughout my undergraduate studies, all projects, assignments, and presentations were group oriented. I found that there is great importance in initiating and independently working on self-driven projects. This allowed me to explore and apply concepts beyond classroom learning. While I did enjoy working independently, there were times at which I found myself looking for the support and sense of teamwork that comes with working in a group. It highlighted the idea of how collaborative work can bring different perspectives to problem solving. From the project proposal submission to the final submission due date, there was plenty of time to work on this project, but I unfortunately continued to push back working on it when other assignments, projects, and exams popped up. I like to think that I have been able to develop my time management skills decently, but there is always room for improvement and this project emphasized that fact to me. We were only given a final submission due date, and this made me realize the necessity of creating personal milestones and concrete checkpoints to maintain consistent progress, especially in the absence of formal checkpoint deadlines.

Originally when I proposed my project, I wanted to use R programming language and RShiny to create visualizations of data in the user interface for Sabrina Carpenter's tour and show information. As I was creating the relational database for this project, I didn't think I would be able to learn how to implement a user interface with RShiny by the time the project needed to be completed. I decided to change the host language to Java because I already have previous experience using it and wanted to maintain my momentum with this skill to deliver a functional application within the available time. I also changed one of the relationships from my original project proposal. Originally, I had song/setlist interacting with show, but while gathering data and attempting to insert this data, I found that each show having its own setlist would take too long and was unmanageable for just myself. One problem I had was none of my .csv files would import into MySQL Workbench, leading me to insert data by hand using SQL insert statements. This was a deciding factor in changing the song to show relationship to song to tour.

Future Work

This database application demonstrates the integration of relational database management, Java programming, and graphical user interfaces to create a functional, user-friendly system. My hope for this application is that it can continue to be a useful tool to track tour information for Sabrina Carpenter. As she continues in her music career, I assume there will be future tours that need to be documented. When new tours are announced and more detailed information is released, such as show dates and cities, this database application can be used to log that information.

While this application is fully functional as it stands, there are some additional features that could increase usability. One potential area for added functionality would be allowing a user to do joins on tables they wish to use. Right now, the user interface only allows the user to view a single table with no joins. One way to implement this functionality would use check boxes or

text boxes for the user to define the tables they wish to join as well as for the attributes they wish to see in the result table. This would not require the user to know SQL syntax. Another way to do this could include the ability for the user to write an SQL statement, but this would require them to understand SQL syntax. Another potential area of functionality is the addition of reporting and analytics. There could be the addition of dashboards for revenue and show attendance analysis. This would allow users such as organizers and Sabrina's tour/show management team to make decisions that are more data driven. This could affect how they choose to market tours and shows as well as which cities to visit.

Demo Video

Here is the link for my demo video, this will be included in the submission comment as well.

[Final_project_video_leek.mp4](#)