Bryan Burnett

SNHU CS-499 Computer Science Capstone

Prof. Brooke Goggin, M.S., M.S., M.S., Ed.D (ABD) Computer Science Adjunct Faculty

June 5, 2023

Final Enhancement

In attempting to recreate this project, learned how to make a database from the MySQL Workbench. This involved going through some tutorials, the Microsoft and MySQL documentation, and a fair bit of trial and error with the date-time format. This was something new to learn, and I'm very pleased to say the database works, as do the queries I intended to showcase. I have included the code, [C:\Users\javab\OneDrive\Documents\oh\_shno\_the\_sql-mile2.txt](file:///C:\Users\javab\OneDrive\Documents\oh_shno_the_sql-mile2.txt) and screenshots of the process.

First, I created the database, in this case it was a “Schema”, named it ‘*flight\_reservations*’. Then I re-created the tables “*customer*”, and “*flight*”. I then added the information into the tables:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

The workbench shows the code in proper MySQL format:

A screenshot of a computer

Description automatically generated

The workbench then displays a confirmation of what was changed.

A screenshot of a computer

Description automatically generated

The workbench has taken manually writing code and made it a GUI interface, much like creating a program in Visual Studio.

The queries, and subsequent code are still manual and provide some insight as to how MySQL works.

A screenshot of a computer

Description automatically generated with medium confidence

I needed to add some data to the customer table, to do this I used the INSERT INTO command:

A screenshot of a computer

Description automatically generated with medium confidence

Which resulted in an updated list in the customer table:

A screenshot of a computer

Description automatically generated

Next, I did the same for the flight table:

A screenshot of a computer

Description automatically generated

This also resulting in an updated list:

A screenshot of a computer

Description automatically generated

Next, I add myself to the customer table, and realize I made a mistake in the ‘*customer\_age’* column, and that I entered myself twice.

A screenshot of a computer

Description automatically generated

I can fix this by using UPDATE in the customer table, to SET the appropriate age, WHERE ‘*customer\_id*’ is the assigned id number, in this case ‘11’. Here’s the result of this:

A screenshot of a computer

Description automatically generated

Now, to remove that pesky duplicate file, I use DELETE FROM the customer table, WHERE the ‘*customer\_id’* was a duplicate, in this case ’12’.

A screenshot of a computer

Description automatically generated

This resolved the duplicate:

A screenshot of a computer

Description automatically generated

Now to join the two tables. This was joined on the ‘*flight\_id’,* and ‘*customer\_id’*, so where there are no more flights associated with the id numbers, the default value is NULL, as seen highlighted in blue below.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

To exclusively show the people on the flight over the age of 60 and have them go to “Miami”. Here’s the code and the output:

A screenshot of a computer

Description automatically generated

If I wanted to see everyone in the customer table, EXCEPT those aged 72, I would eliminate Robert De Niro by using the EXCEPT command in the query. This looks like this:

A screenshot of a computer

Description automatically generated

The output gets rid of Mr. De Niro and returns every other name on the list.