Zach Wilson

Saba Jamalian

CS 370-03

5.1.23

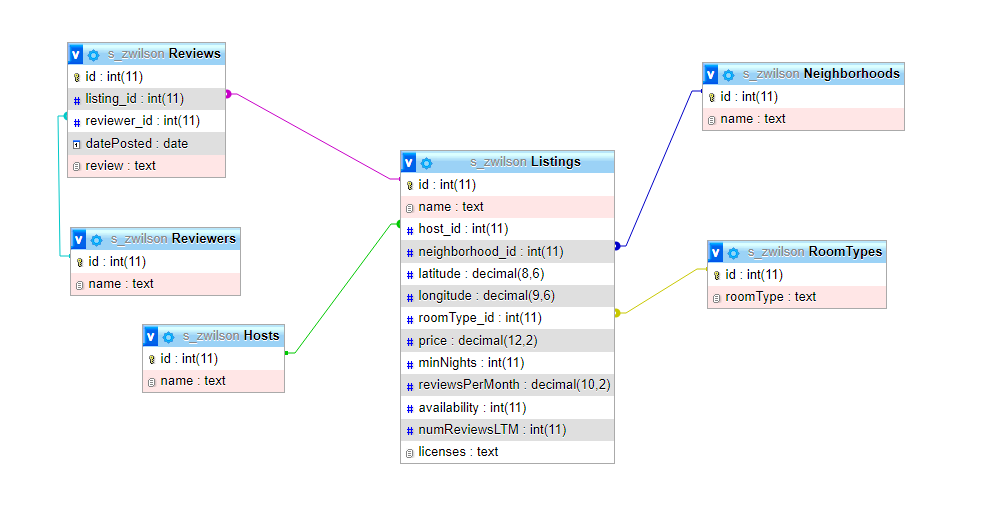
Final Project Report

This project seemed very daunting when I first looked at these CSV files. The fear was compounded by the fact that I also tried opening them in excel that way I could play with the formatting using conditional formatting to help me visualize how I needed to break up the data into tables. You can see the result of my visualizing the data as excel files in the extra resources directory in [my GitHub Repo](https://github.com/PlatFormPlayZ/CS-370-Final-Proj) for this project. One thing I noticed right off the bat was that none of the data was any sort of sanitized and so in order to align with the project description of importing the data from the csv files into a database I decided that sanitization was not my job today. Therefore, a few design choices that I do not love but did as a time saver is that basically any string is just the data type of TEXT that way, I did not have to worry about the length of what I was inserting into the database. This worked well for me to simplify the headache as I knew just parsing through the data alone to put into my SQLite database was going to be work enough.

I could tell early on that there were a few tables I would want to make out of the listings csv file which were the room type column and the host’s information. The neighborhood was something else I decided to break out into its own table which in hindsight I could have added the neighborhood group column to that table but since there was no data in that column anywhere, I did not bother giving it much thought. There were in fact several columns in the listings csv file that I knew I did not need to keep track of in the database as they could generally be solved with good queries and one column that I had no idea what to do with. Those columns that did not need to be stored in the database were number\_of\_reviews, calculated\_host\_listings\_count, and likely reviews\_per\_month but that is one of two which I decided to store anyways because it seemed rather annoying to figure out later. The other rather annoying column was number\_of\_reviews\_ltm which I could only guess to mean last twelve months (ltm) but that did not seem quite right with what I could see in my excel sheet so I just decided to store it since I did not know what it really stood for.

The reviews data file was quite nerve racking because when you open it in a text editor, it is completely malformed. Luckily excel made it at least legible but the file is so long it was a huge pain to do any sort of meaningful analysis using conditional formatting and the like. Due to it only having 6 columns with two of them being able to be broken out into a separate reviewer table. It was just then a matter of parsing the data properly and inserting it into the database.

This all left me with a total of 6 tables with reviews and listings being largest tables. You can see the layout of the tables in figure 1 which is my Entity Relationship Diagram generated by phpMyAdmin on my student playground server.



Figure

For the SQL queries to extract insights from the data, I created each query and saved them in their own .sql file. The only things of note for those other than I just made a query that follows the instructions give, was that the eleventh and twelfth queries gave the most issues. The eleventh query was vague and I did not know really what to do for it so I wrote a query that for each listing for each month printed out the number of reviews so to demonstrate the seasonality of the listings. Then for the twelfth query, I was very unsure how to show the correlation between two columns in the traditional math sense so I had it return the number of reviews over the number of minimum nights. Each of the twelve queries have their own .sql file and .csv file which is the saved output from when I ran the query.