Beaver Gym

Matthew de Marte, Christine Regan, Elizabeth Lane

From a process perspective this project went really well for us because we were able to work efficiently through the tasks and didn’t hit any major road-blocks. One thing we could’ve improved on is how the actual tables were structured. For example, at first we didn’t realize that, when putting time in a column in our table, SQL cannot recognize two separate time stamps in one record. Instead, we could have added another table specifically relating to classes and their times, with each record representing an individual time the class is offered at.

Another problem we ran into was with entering the cell phone numbers of the Beaver Gym members into the Member table. We originally drafted the data in Excel, but quickly realized how tedious it was to copy and paste each value into SQL for the ‘INSERT INTO table VALUES (column1, column2, …)’ command. Therefore, we used VBA to concatenate each cell in each row into one cell. However, when doing this, the data type of the cell phone numbers couldn’t handle such a large number (e.g. 9302840202), so it converted the number into scientific notation. When we copied and pasted these values from Excel VBA to SQL, we got error messages because the data types weren’t matching up. After much trial and error as well as careful observation, we figured out that the scientific notation (e.g. 9.203E+09) was the issue because the SQL data type ‘bigint’ didn’t accept that format.

We ran into a few other data type errors when inserting the data into our tables. For instance, with the zip code column, we had to change the data type from a small integer to an integer so that it could handle 5-digit numbers. Also, we wanted to leave some table values blank (that were originally constrained to not null), so we ended up needing to change the data constraints of those columns to ‘null.’

Our project might not have been appropriately scoped. A gym generally has hundreds, and potentially thousands of members, and would employ many more workers and have many more class registrations. Due to the potential time it would take to produce data tables with thousands of rows, we thought it better to scale our data with less workers, members, and class offerings.

We had to teach ourselves some code for SQL. Writing scripts for every query posed some challenges and we sometimes used w3schools.com and stackoverflow.com for help. These websites helped us write the correct scripts to produce the necessary queries. Going forward what we learned will enable us to have a better understanding of SQL and how to use it.

When deciding how to divide up the work for this assignment, we decided to meet and do all the work together. We planned on doing this because each part of the project was intertwined. Therefore, when we were working on separate sections we would be able to talk about issues we were facing and ask for help if needed. In reality, we were able to execute our plan to complete the parts of the project while working together; our group did a really good job of working together to accomplish the tasks at hand. We completed the work by meeting together and dividing up tasks based on group member’s strengths (the members who were better at SQL tackled those sections). Our group worked really well together and we were able to do so conflict-free. There is nothing our group would change if we had to work together again; each member was diligent and worked hard at accomplishing what their team members asked of them.

“I pledge my honor that I neither received nor provided unauthorized assistance during the completion of this work”