Ann Liberman 4/14/2021
CISC 3140

Lab #9

First I learned about the overview of relational databases, and how they store data in a table, much like a spreadsheet. Thus, like a spreadsheet, it makes sense that each row would represent a new item, and each column would represent properties about that item. The overview explains how databases come with a query language, this language allows the user to *interact* with the database. Such interactions might include getting, adding, or changing data within that table. For the purpose of this assignment I will be focusing on SQL, which is a language designed entirely for accessing databases. As this is my first real introduction to SQL I will spend time watching as many Khan Academy videos on SQL as possible and practicing their in-screen demos.

**Creating an Empty Table**

```
CREATE TABLE groceries (id INTEGER PRIMARY KEY, name TEXT, quantity Integer);
```

// **groceries** is the name of the table
// **name** = new column, name of each grocery item
// **TEXT** = date type of **name**
// **quantity** = new column, how many of each item
// **INTEGER** = data type of **quantity**
// **id** = name of row identifier
// **INTEGER PRIMARY KEY** is a data type that also signals that each row must have a unique value for it

**Filling Table with Data**

```
INSERT INTO groceries VALUES (1, “Bananas”, 4);      // this puts 4 Bananas on the list, with the unique ID of “1”
INSERT INTO groceries VALUES (2, “Peanuts”, 1);      // this puts 1 Peanuts on the list, with the unique ID of “2”
INSERT INTO groceries VALUES (3, “Chocolate”, 2);    // this puts 2 Chocolates on the list, with the unique ID of “3”
```

**Challenge: Creating My Own Database of Favorite Books:**

```
CREATE TABLE favoriteBooks (id INTEGER PRIMARY KEY, name TEXT, rating INTEGER);
INSERT INTO favoriteBooks VALUES (1, "Harry Potter", 10);
INSERT INTO favoriteBooks VALUES (2, "Kite Runner", 10);
INSERT INTO favoriteBooks VALUES (3, "Idk", 7);
SELECT * FROM favoriteBooks;                          // SELECT is used to perform any query
SELECT name FROM favoriteBooks;                       // returns everything from the name column

SELECT * FROM groceries WHERE aisle > 5 ORDER BY aisle;    //returns only aisles greater than 5, in order
```

**Aggregate Functions**

An aggregate function is useful for things like getting the maximum, minimum, sum, and average of given values in a database. To do this you use:

```
SELECT SUM (quantity) FROM groceries;                 // returns the sum of of the quantity column

SELECT aisle, SUM (quantity) FROM groceries GROUP BY aisle;    // behind the scenes, first it groups the aisle
```

```
                                        // rows and then it performs the sum
                                        // function
```

**Recap**:

As someone who has never used SQL before, or any other type of data query language, I found the introduction to SQL to be fairly easy to navigate, and not too many difficult concepts to wrap my brain around. One of the patterns I noticed when writing a query is that the variables are always written in all lowercase, and the functions are always capitalized. This makes it easier to understand what a specific statement is meant to be doing, and what sort of data one might be working with. I would estimate that in total I spent about 10 hours learning various SQL query types / formats, and also practicing writing them on my own. Compared to some other coding languages I have learned / been introduced to this semester I found this one to be one of the easier ones to learn and understand. And, in a way, it seems like it's more about memorizing commands rather than tricky algorithms (at least so far). Also, as someone who is interested in pursuing a field within the data science realm it's important for me to know this skill, so I will be continuing the Khan Academy course after this assignment.