**CS3431 A17 Wong**

**Assignment 2: SQL 1 – 4**

Due Date: R 9/7 at 11:59pm.

Late Policy: 10% off until F 9/8 at 5pm. 0 points afterwards. Maximum grade is 100 points.

Submission: submit your tours2.sql file to Canvas using the Assignment 2 link. Only the most recently submitted assignment will be graded and checked for lateness.

This homework is to be done individually. You may speak to your classmates about the assignment but you cannot exchange information on the actual SQL code that needs to be written. This means you cannot look at another student’s code. However, you can give a similar SQL example that involves code.

You will be creating an expanded version of the database from assignment 1. The data is located in the spreadsheet CS3431-A17 Assignment2.xlsx. There are now 5 tables of data, one on each spreadsheet tab: ReservedTours, Customers, Tours, Locations, and Guides. **Changes from the previous assignment are highlighted with a red background and consist of the following:**

1. ReservedTours now has a price field
2. There is a new Locations table
3. Guides now has a licenseType field

Use a text editor to create **tours2.sql** that will include all of your SQL commands:

1. The first five commands will drop the ReservedTours, Customers, Tours, Locations and Guides tables so you can run your tours2.sql file over and over. Do not use “Cascade Constraints” with your drop commands.
2. (20 points) Write the SQL commands to create the 5 tables following the instructions below:
   1. Create the tables with the given table names, field names, and datatypes given in the spreadsheet.
   2. The first column of each table is the primary key.
   3. Include all of the constraints from assignment 1 (given below for your convenience):
      1. In the Guides table, the driverLicense field is unique and must be non-null.
      2. The ReservedTours table contains 3 foreign keys referencing the other three tables.
      3. In the Tours table, the vehicleType is constrained to be boat, bus, or car.
      4. For the Customers, Tours, and Guides tables, the referential integrity should be set so if a record is referenced by the ReservedTours table, the referring field in ReservedTours will be set to null when the record is deleted.
3. The new Locations table contains 1 foreign key referencing the Tours table. If a tour is deleted then the records referencing the deleted records are automatically deleted as well. Examine the data and fill in the values for the blank TourID fields.
4. ReservedTours now has a new price field that is initially null for every record.
5. In the Customers table, the age field must be non-null and the phone number must be unique.
6. The licenseType field in the Guides table can only have the values ‘land’, ‘sea’, or ‘both’. Land licenses permit the use of buses and cars. Sea licenses permit the use of boats. ‘Both’ licenses permit the use of buses, cars, and boats.

Before you proceed to the next section, you will want to create a high level summary of the database schema (refer to section 2.2.2 in your textbook) so you can see all of the tables and its attributes in one place. You can use italics to indicate that a field is a foreign key. For example,

Books (ISBN, FirstName, LastName, Title, *PubID*)

An alternative method is to use PK and FK after a field name to indicate primary and foreign keys:

Books (ISBN PK, FirstName, LastName, Title, PubID FK)

1. Write the following SQL commands (20 points each)
   1. Update the ReservedTours prices based on the prices in Tours.   
      Note: You will need to research how to do a nested SELECT statement in a SET clause.
   2. List for each level of guide - junior guide, guide or senior guide - how many mismatches there are between the required tour’s vehicle type and the guide's license type. Sort by the title.
   3. For each customer, list the first name, last name, and total amount being spent for land-based tours. Format the price so it is displayed with two decimals and with the heading TotalLandPrice. Sort by last name, and then first name.
   4. Determine the customer who will make the most number of visits to tour locations. List the firstName, lastName, and the number of location visits (use the heading Visits)

Note: if you encounter difficulties, try the following: 1) create parts of the queries before putting them together, and 2) refer to the Order of Execution slide in the SQL 3 PowerPoint slides