**CS3431-A20 Wong**

**Assignment 2: SQL 1-4**

Due Date: F 9/11 at 11:59pm.

Late Policy: 10% off until Sa 9/12 at 5pm. 0 points afterwards. Maximum grade is 100 points.

Submission: zip your tour3.sql and tour4.sql files (you can name the file tour.zip) and upload them to Canvas using the Assignment 2 heading link.

The homework is to be done individually. You may speak to your classmates or post to Slack 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 another database to keep track of tours that vacationers have reserved. The data is located in the spreadsheet CS3431-A20 Assignment2.xlsx. There are now 5 tables, one on each spreadsheet tab: Reservation, Customer, Tour, Guide, and a new table called Location.

Use a text editor to create **tour3.sql** that will include the following SQL commands:

1. (5 points) The first five commands will delete the Reservation, Customer, Tour, Guide, and Location tables and sequences so you can run your SQL files over and over
2. (15 points) Write the SQL commands to create the 5 tables following the instructions below:
   1. For each table, the field name and datatypes are given in the spreadsheet.
   2. The first column of each table is the primary key.
   3. Primary keys that are a number datatype should use a sequence for automatic incrementing. Make sure that the generated keys match those given in the spreadsheet.
   4. Reservation now has a new price field which starts off as null for every record. Make sure that the insert statement correctly enters this information! This table also contains 3 foreign keys referencing the other three tables.
   5. In the Customer table, the age field must be at 5 years or older, and the phone number must be non-null and unique.
   6. The new Location table contains 1 foreign key referencing the Tour table. Examine the data and fill in the values for the table.
   7. The vehicleType field, in both the Tour and Guide tables, can only have the values ‘duckboat’, ‘bus’ and ‘car’

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 PK to indicate a primary key, UQ to indicate Unique (a candidate key), and FK to indicate that a field is a foreign key. For example,

Book(ISBN PK, FirstName, LastName, Title UQ, PubID FK)

Use a text editor to create **tour4.sql** that will include the SQL commands below. You may want to build the queries part-by-part and seeing the intermediate results before putting together the entire query.

1. (20 points) Create a query that, for each location type (GROUP BY), lists the number of unique customers (COUNT) who will visit that location type. For example, exactly 7 different (DISTINCT) customers will be going to parks. Display the results under LocationType and NumberOfCustomers headings.
2. (20 points) Create a query that will display the reservations where the assigned tour guide is not permitted to drive the required vehicle type. Display the reservation ID, the tour name, the vehicle type for the tour, the first and last name of the guide, and the vehicle type that the guide is permitted to drive. Sort the results by tour name.
3. (20 points) Create a query with a subquery that shows the city with the most reservation tours. The output should be displayed under the headers: City, NumberOfTours  
   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
4. (20 points) Update the Reservation records to reflect the following prices:

a. and b. Boston tour prices are $100 for ages over but not including 21, and $25 otherwise.

You should have 2 separate update statements, each one with a nested select clause.

c. create a query that displays the reservationID, customer’s firstName, customer’s lastName, tourName, city, age and price; sorted by city and then age but only display those records where the tour name includes the word Tour (exact capitalization) anywhere in its name. For example, records that contain a tour name field with the following values would be displayed:

Tour of the Americas

Wildlife Tour

New York Touring Expedition

The following records that contain a tour name field with the following values would NOT be displayed:

Asiatic tour

TOUR Of A Lifetime