**CS3431-A24 Wong**

**Assignment 1: Basic SQL**

Due Date: R 8-29 at 11:59pm.

Late Policy: 10% off until F 8/30 at 5pm. 0 points afterwards. Maximum grade is 100 points.

Submission: submit your cruise1.sql file to Canvas using the Assignment 1 link.

The homework is to be done individually. You may speak to your classmates about the assignment but cannot exchange information on the actual SQL code that needs to be written.

**Use Oracle SQL Developer do this assignment or use LiveSQL.oracle.com if you have installation problems.** You will be creating a database to keep track of Cruise that vacationers have reserved. The data is in the spreadsheet attached to the assignment, CS3431-A24 Assignment 1**v2**.xlsx. There are 5 tables, one on each spreadsheet tab: Reservation, Customer, Cruise, TravelAgent and Company. The 6th tab is an example of how to automatically generate the insert statements so you do not have to manually type every line. Remember double quotes are used in Excel for strings. Single quotes are used in SQL for strings.

**Receive 1 point for going to the course Slack site and responding to my Pre-Assignment message on the #general channel.**

Use a text editor to create cruise1.sql that will include all your SQL commands:

1. The first commands will delete the Reservation, Customer, Cruise, TravelAgent and Company tables so you can run your cruise1.sql file over and over without needing to reset your session. Note that you will need to drop the tables in a specific order because of integrity constraints. Do NOT use the following command for this assignment: drop table <TableName> cascade constraints;  
   Also delete the sequences that you will create for the primary keys (read next section).
2. (30 points) Write the SQL commands to create the five tables following the instructions below. All constraints should be named! Note that due to referential integrity constraints, you will need to be careful about the order you create the tables and insert records into them.
   1. For each table, the field name and datatypes are given in the spreadsheet. Use the EXACT given table and field names.
   2. For the field names that are supposed to auto increment, create the field as a number data type and use sequences. Start with 0 and increment by 5. In other words, it should result in 0,5,10,15,20… Do NOT manually enter these values.
   3. The first column of each table is the primary key. Make each one a named constraint.
   4. In the Company table, the stock symbol is a candidate key but not a primary key. Make this a named constraint.
   5. In the Customer table, the phone field must be non-null. The combination of first name, last name, and phone number is a candidate key, but not a primary key. Use named constraints. Hint: for the phone field, use a value check.
   6. The Reservation table contains 3 foreign keys each referencing another table. The Cruise table has one foreign key referencing the Company table.
   7. In the TravelAgent table, the title is constrained to be Assistant, Agent, and Manager.
   8. For the Customer, Cruise, and TravelAgent tables, the referential integrity should be set so if a record in one of those tables that is referenced by the Reservation table is deleted, then the record in Reservation will also be deleted.
   9. If a Company is deleted, then the referential integrity should be set so the Company field in Cruises referencing it will be set to null.
3. Note: you may want to keep the following SQL commands in a separate file until you are finished. That way you can rerun the SQL queries without have to create the tables repeatedly. Create a schema of all the tables so you can easily see the tables, keys, and attributes all in one place. Write the following SQL commands to
   1. (10 points) Write a single SQL command that increases the price by 15% of cruises that leave from Miami departure ports. Round the results to the nearest cent. After the price updates, list the cruise name, the departure port, the ship name and the price. Use the to\_char function to display the price with a leading dollar sign and 2 decimal points for the results. For example: $150.40. The heading for this new calculated price should be NEWPRICE.
   2. (20 points) List all the cruises where customers over the age of 50 are departing from a Miami port. Include just the company name and the cruise name. Do not display duplicate records in the results. Sort the results in alphabetical order by company name. Do not use natural joins for this solution.
   3. (20 points) List 7-day Cruise that have been booked by **assistant** travel agents. Include the cruise name, travel date, travel agent first name and last name as a single field called ‘bookingAgent’, and ship name. Use natural joins. Sort by cruise name and then by bookingAgent.
   4. (20 points) List ALL Customer under 65 years of age and the ships they are traveling on, if any. Sort by customer last name, customer first name, and ship name. Do not display duplicate records in the results. **Hint: There is at least one record where a person has no ship.**

If you encounter problems with step 3, create a part of the SQL command, check it, and then add to it. For example, in part 3.d., write the SQL statement for just Customer over 65. After verifying that it works, then add the code for the ships. Then include the code to do the sorting.