Lab 03 – SQL – Single Table Queries

Objectives

The purpose of this lab is to start learning SQL by writing basic DML statements involving a single table. You will learn to create basic CRUD statements (queries as well as insert, update and delete).

Submission:

Your submission will be a single SQL file with the solutions provided. (with a .sql file extension)

Setup

Create a new worksheet in SQL developer and add an appropriate comment header that includes your name, Oracle USER ID, student id, the date and the purpose of the file (i.e., DBS211C50 _Lab03).

Immediately under the comment header, enter the following line and then execute it:

```
SET AUTOCOMMIT ON;
```

You will need to execute this statement each time you login to the server until the completion of this lab.

Save the script as: `DBS211Lab03.sql`

Style Guide

Your SQL should be written using the standard coding style:

- all keywords are to be upper case,
- all user-defined names are to be lower case, (example: table and field names)
- there should be a carriage return before each major part of the SQL statements (i.e. before SELECT, FROM, WHERE and ORDER BY)

Using comments to number the question answers, write the SQL code to complete the following tasks.

Tasks:

1. Display the data for all offices.

2. Display the employee number for all employees whose office code is 1.

3. Display customer number, customer name, contact first name and contact last name, and phone for all customers in Paris. (**hint**: be wary of case sensitivity)

4. Repeat the previous Query with a couple small changes:
    a. The contact's first and last name should be in a single column in the format "lastname, firstname".

    b. Show customers who are in Canada

5. Display customer number for customers who have payments. Do not included any repeated values. (**hints**: how do you know a customer has made a payment? You will need to access only one table for this query)

6. List customer numbers, check number, and amount for customers whose payment amount is not in the range of $30,000 to $65,000. Sort the output by top payments amount first.

7. Display the order information for all orders that are cancelled.

8. The company needs to know the percentage markup for each product sold. Produce a query that outputs the ProductCode, ProductName, BuyPrice, MSRP in addition to
    a. The difference between MSRP and BuyPrice (i.e. MSRP-BuyPrice) called *markup*
    b. The percentage markup (100 * calculated by difference / BuyPrice) called *percmarkup* rounded to 1 decimal place.

9. Display the information of all products with string '***co'*** in their product name. (c and o can be lower or upper case).

10. Display all customers whose contact first name starts with letter ***s*** (both lowercase and uppercase) and includes letter ***e*** (both lowercase and uppercase).

11. Create a statement that will insert yourself as an employee of the company.
    a. Use a unique employee number of your choice
    b. Use your school email address
    c. Your job title will be "Cashier"
    d. Office code will be 4
    e. You will report to employee 1088

12. Create a query that displays your, and only your, employee data

13. Create a statement to update your job title to "Head Cashier"

14. Create a statement to insert another fictional employee into the database. This employee will be a "Cashier" and will report to you. Make up fake data for the other fields.

15. Create a statement to Delete yourself from the database. Did it work? If not, why?

16. Create a statement to delete the fake employee from the database and then rerun the statement to delete yourself. Did it work?

17. Create a **single** statement that will insert both yourself and the fake employee at the same time. This time the fake employee will report to 1088 as well.

18. Create a **single** statement to delete both yourself and the fake employee.