Lab 03 – SQL

Single Table Queries Statements

# **Objectives**

The purpose of this lab is to start learning SQL by writing basic SELECT and DML statements involving a single table.

**Submission**

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

* Save your SQL statements as a script named: DBS211\_L03\_Group#.sql

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

Example Submission

|  |
| --- |
| -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Name: Your Name  -- ID: #########  -- Date: The current date  -- Purpose: Lab 03 DBS211  -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Q1 SOLUTION --  SELECT \* FROM TABLE;  -- Q2 Solution –  SELECT \* FROM TABLE; |

## **Setup**

Create a new worksheet in SQL developer and add an appropriate comment header that includes all the group member’s name, student id, the date and the purpose of the file (i.e. DBS211 – Lab 03).

## **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)

See the following sample:

SELECT columns

FROM tables

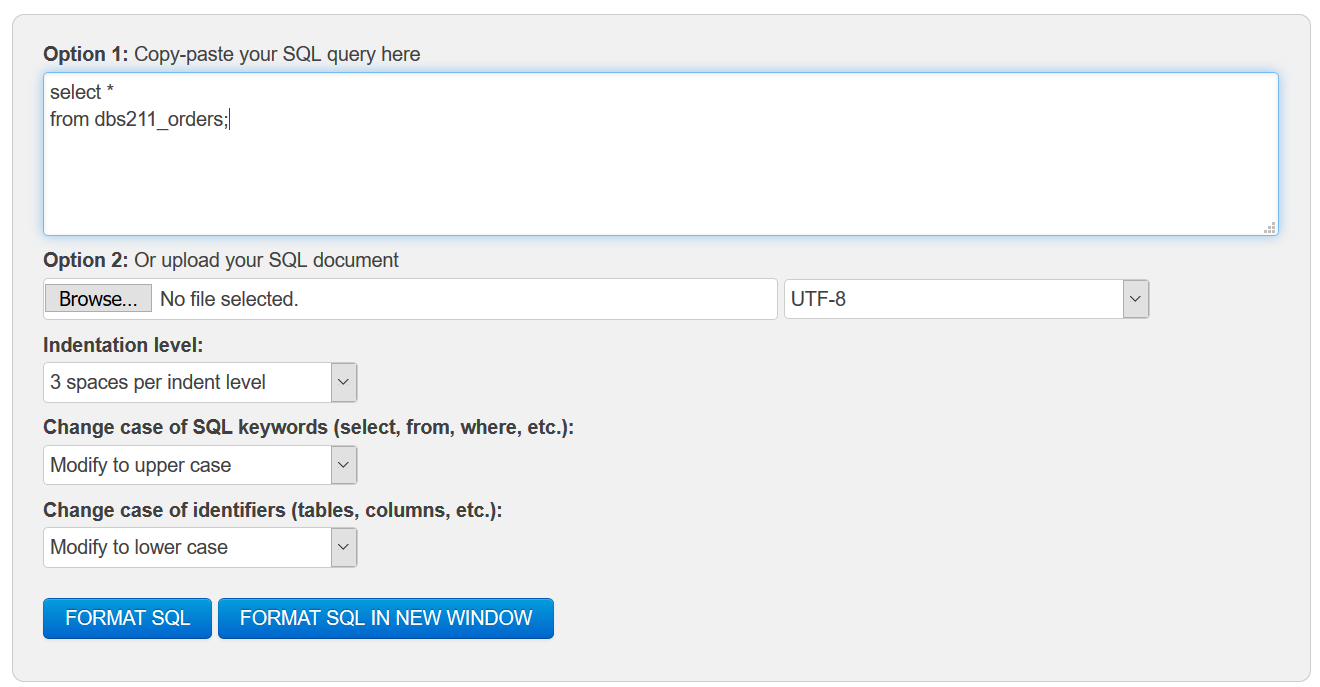
WHERE conditions

ORDER BY column1, column2;

To save time, you can write all SQL statement in your SQL developer. To make sure that your SQL statements style follows the standard SQL style guide, copy and paste your SQL statement onto the following website and click on “FORMAT SQL” or “FORMAT SQL IN NEW WINDOW”.

<https://www.freeformatter.com/sql-formatter.html#ad-output>

You can also upload your SQL file. See the setting in the following image. Have SQL keywords (SELECT, INSERT, UPDATE, etc) uppercase and user defined objects and identifiers (tables, columns, etc.) lowercase.



**Marking Scheme**

|  |  |  |  |
| --- | --- | --- | --- |
| **Question** | **Weight** | **Question** | **Weight** |
| 1 | 6% | 6 | 6% |
| 2 | 6% | 7 | 6% |
| 3 | 6% | 8 | 6% |
| 4 | 6% | 9 | 6% |
| 5 | 6% | 10 | 6% |

**Grade Policy**

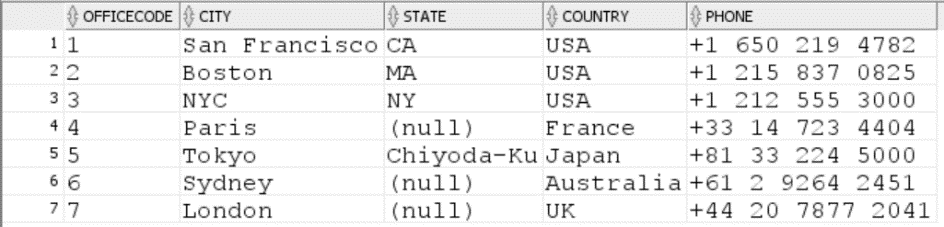
* Questions with errors do not get any marks. (They get zero.)
  + Execute your *.sql* file using the “Run Script” button to make sure there is no errors in your file.
* If your result in a question does not match the sample output results, you do not get any marks.
* You do not receive marks for the missing solutions.

**Tasks:**

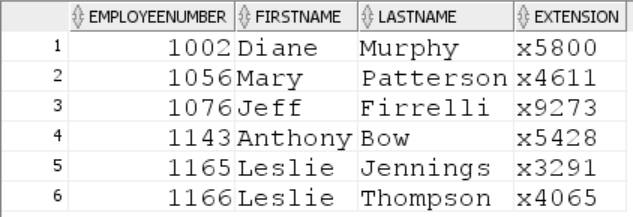
**SQL SELECT**

Remember to comment the question number for each solution.

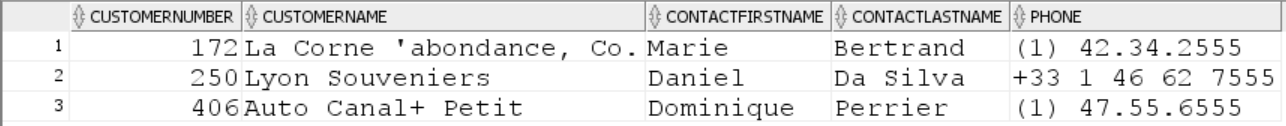
1. Display the data for all offices. *Display office code, city, state, country, and phone for all offices.*



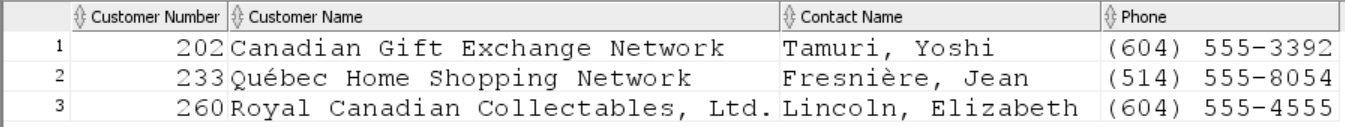
1. Display *employee number, first name, last name, and extension* for all employees whose office code is 1. *Sort the result based on the employee number.*



1. 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) *Sort the result based on the customer number.*



1. Repeat the previous Query with a couple small changes:
   1. The contact’s first and last name should be in a single column in the format “lastname, firstname”.
   2. Show customers who are in Canada
   3. *Sort the result based on the customer name.*

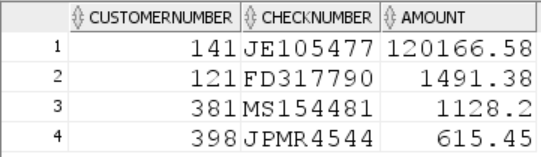


1. Display customer number for customers who have payments. Do not included any repeated   
   values. *Sort the result based on the customer number.* (**Hints**: How do you know a customer has made a payment? You will need to access only one table for this query)

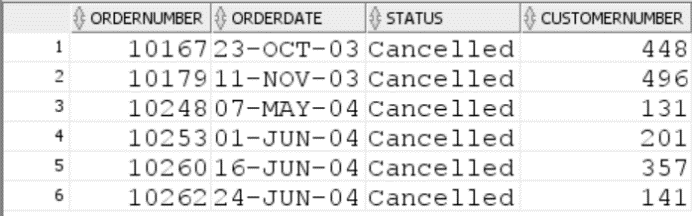
*The first 10 rows of the output result. The query returns 98 rows.*



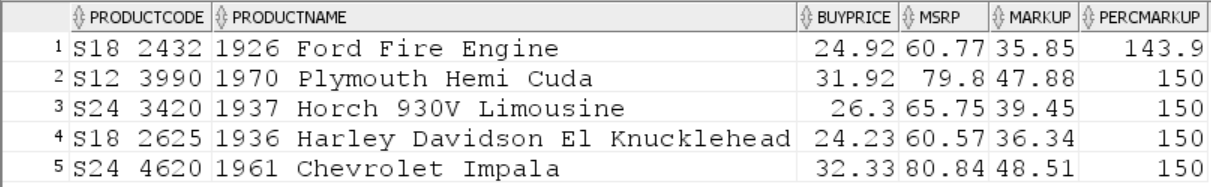
1. List customer numbers, check number, and amount for customers whose payment amount is not in the range of *$1,500* to *$120,000*. Sort the output by top payments amount first.



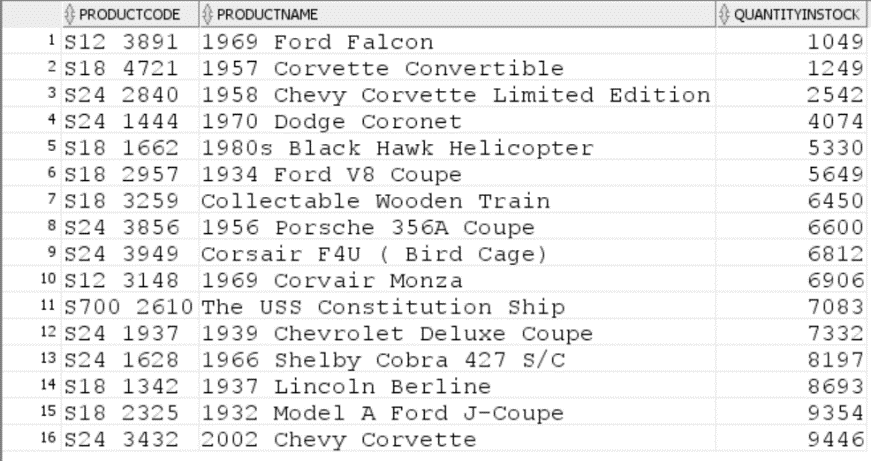
1. Display *order number, order date, status, and customer number* for all orders that are cancelled. *Sort the result according to order date.*



1. 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
   1. The difference between MSRP and BuyPrice (i.e. MSRP-BuyPrice) called *markup*
   2. The percentage markup (100 \* calculated by difference / BuyPrice) called *percmarkup*rounded to 1 decimal place.
   3. *Sort the result according to percmarkup.*
   4. *Show products with percmarkup greater than 140.*



1. Display *product code, product name, and quantity in stock the information* of all products with string ‘***co’*** in their product name. (c and o can be lower or upper case). *Sort the result according to quantity in stock.*



1. Display *customer number, contact first name, contact last name* for all customers whose contact first name starts with letter ***s*** (both lowercase and uppercase) and includes letter ***e*** (both lowercase and uppercase). *Sort the result according to customer number.*

