Lab 04 – SQL (Join/View)

# **Objectives**

The purpose of this lab is to introduce students to querying data from multiple tables. Relationships are used in relational databases to reduce redundant and repetitive data, but it is necessary to reconnect these tables when extracting data and obtaining information. Student will be able to:

* Produce query results containing data from multiple tables using joins and demonstrate their knowledge of inner, outer and full joins.
* To actively troubleshoot queries to handle potentially ambiguous fields across multiple tables through the use of aliases.
* Students learn to create and modify views.

**Submission**

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

DBS211\_L04\_LastName.sql

Your submission needs to include a comment header block and be commented top clearly indicate the answers to each question. Make sure every SQL statement terminates with a semicolon.

Example Submission

|  |
| --- |
| -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Name: Your Name  -- ID: #########  -- Date: The current date  -- Purpose: Lab 04 DBS211  -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Q1 SOLUTION --  SELECT \* FROM TABLE;  -- Q2 Solution –  SELECT \* FROM TABLE; |

Locate, select, and submit the file to the Lab 04 link.

## **Setup**

Create a new worksheet in SQL developer and add an appropriate comment header that includes your name, student id, the date and the purpose of the file (i.e. DBS211 – Lab 04).

## **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

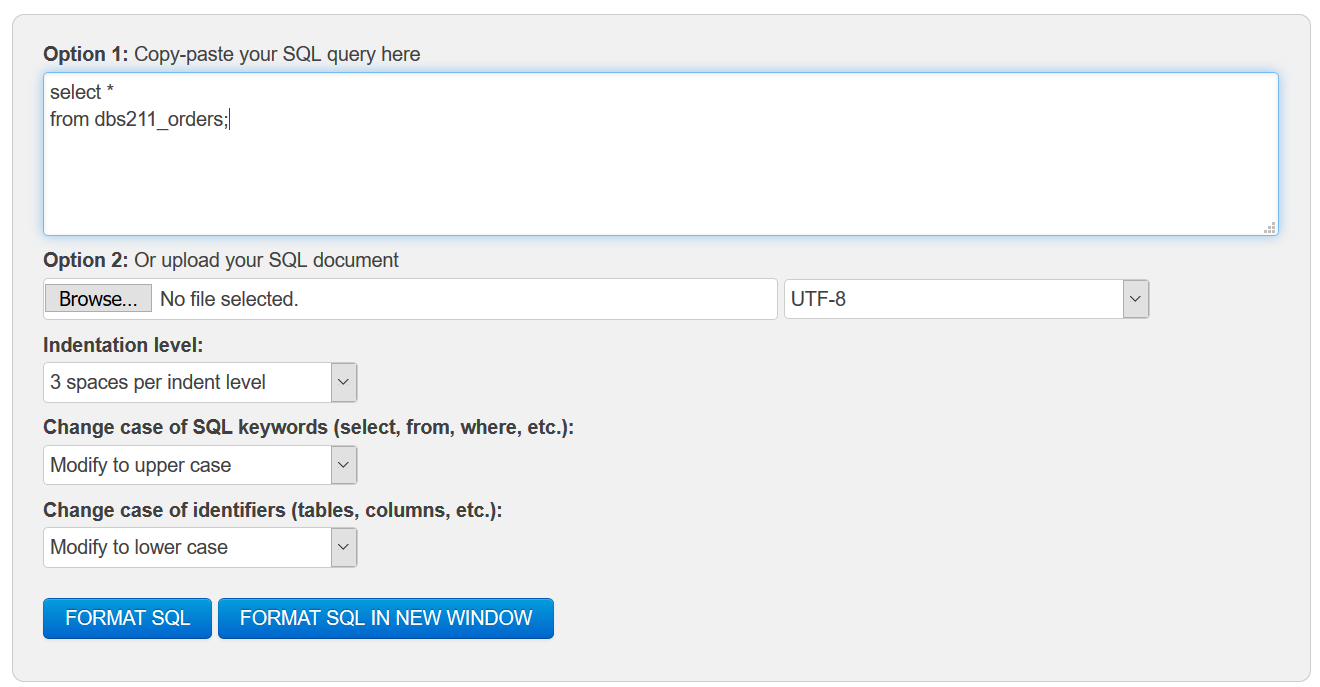
WHERE conditions

ORDER BY column1, column2;

To save time, you can write the SQL statements in your SQL developer. To make sure that your SQL statements style follows the standard SQL style guideline, copy and paste your SQL statements 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** | **Points** | **Question** | **Points** |
| **1** | 1 | **6** | 1 |
| **2** | 1 | **7** | 1 |
| **3** | 1 | **8** | 1 |
| **4** | 1 | **9** | 1 |
| **5** | 1 | **10** | 1 |

Total: 10

**Grade Policy**

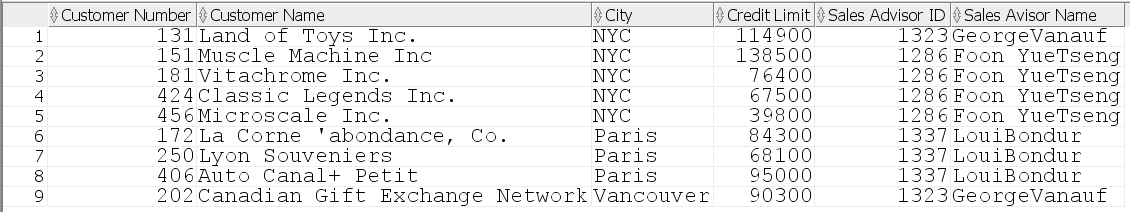
* Submissions with errors do not get any marks. (They get zero.)
  + Execute your *.sql* file using the “Run Script” button to make sure there is no errors in your file.
* If your result in a question does not match the sample output results, you do not get any marks.
* You do not receive marks for the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

**Tasks:**

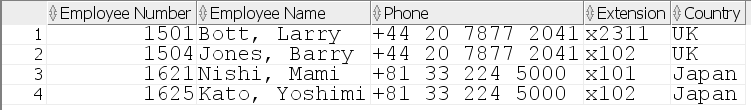
**Important:** **For each question, the title of columns and the output result must match the provided output result in that question to receive a full mark.**  
Remember to comment the question number for each solution.

1. For customers in Vancouver, Paris, and NYC, display customer number, customer name, city, credit limit, sales advisor ID (salesrepemployeenumber), and sales advisor name.

*Sort the output result according to first the column city and then customer number*.



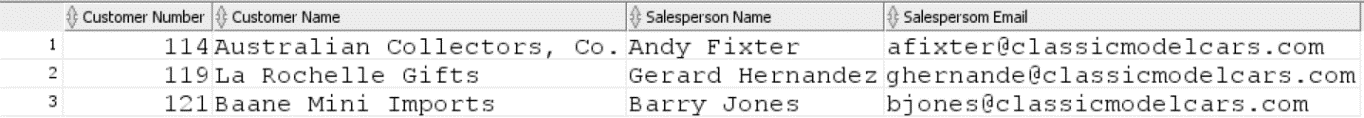
1. Display employee number, employee full name, phone, extension, and country for employees who work in Japan and UK. *Sort the result based on the employee number*.



1. For customers with customer numer 114, 119, and 121, show the customer number, customer name, and the following data for their salespersons

* sales person name
* and email.

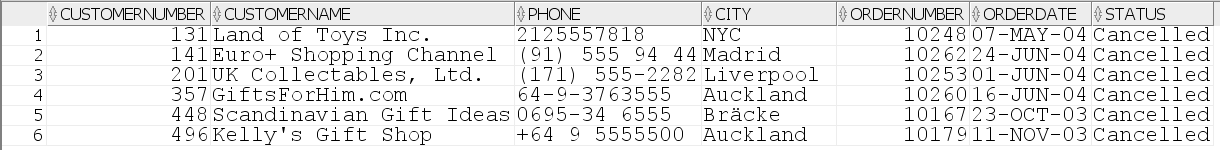
*Sort the result based on customer number.*



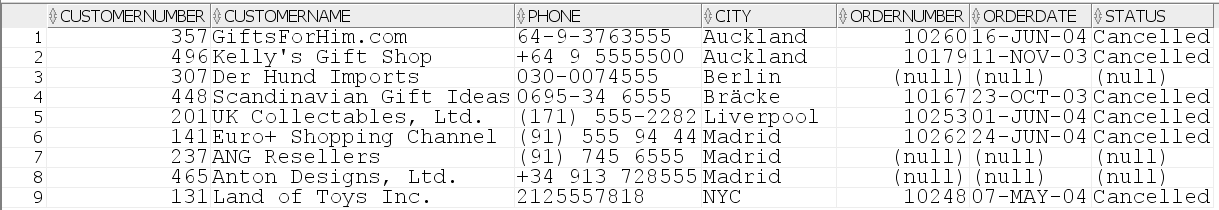
1. Display product code, product name, quantity, and price for products that have not been ordered. *Sort the result according to the product code*.

****

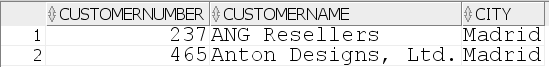
1. Create a view named customer\_report to display customer number, customer name, phone number, city, order number, order date, and order status for customers who have cancelled their orders.
2. Write a SQL statement to display the result of the customer\_report view. *Sort the output based on the customer number.*



1. Modify the customer\_report view to display
   * Customers with cancelled orders
   * Or, customers who do not have any orders and live in Madrid or Berlin.
2. Write a SQL statement to display the result of the customer\_report view. *Sort the result first based on the column city and then customer number.*



1. Using the customer\_report view, show customer number, customer name, and city for customers from Madrid who do not have any orders. *Sort the output result according to the customer number.*



1. Write a SQL statement to delete the customer\_report view.

Good luck