Assignment 01 – SQL

# **Objectives**

The purpose of this assignment is to assess the students’ knowledge about storing and querying data from a database considering the relationships between tables in that database. Students will query data from a database (single/multiple tables) and use the relationships to reconnect some tables when extracting data and obtaining information. Student will:

* Produce query results containing data from multiple tables using joins and demonstrate their knowledge of inner, outer and full joins.
* Troubleshoot queries to handle potentially ambiguous fields across multiple tables through the use of aliases.

**Submission**

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

DBS211\_Assignment01\_Group#.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: Assignment 01 DBS211  -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Q1 SOLUTION --  SELECT \* FROM TABLE;  -- Q2 Solution –  SELECT \* FROM TABLE; |

Locate, select, and submit the file to the assignment 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 – Assignment 01).

## **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

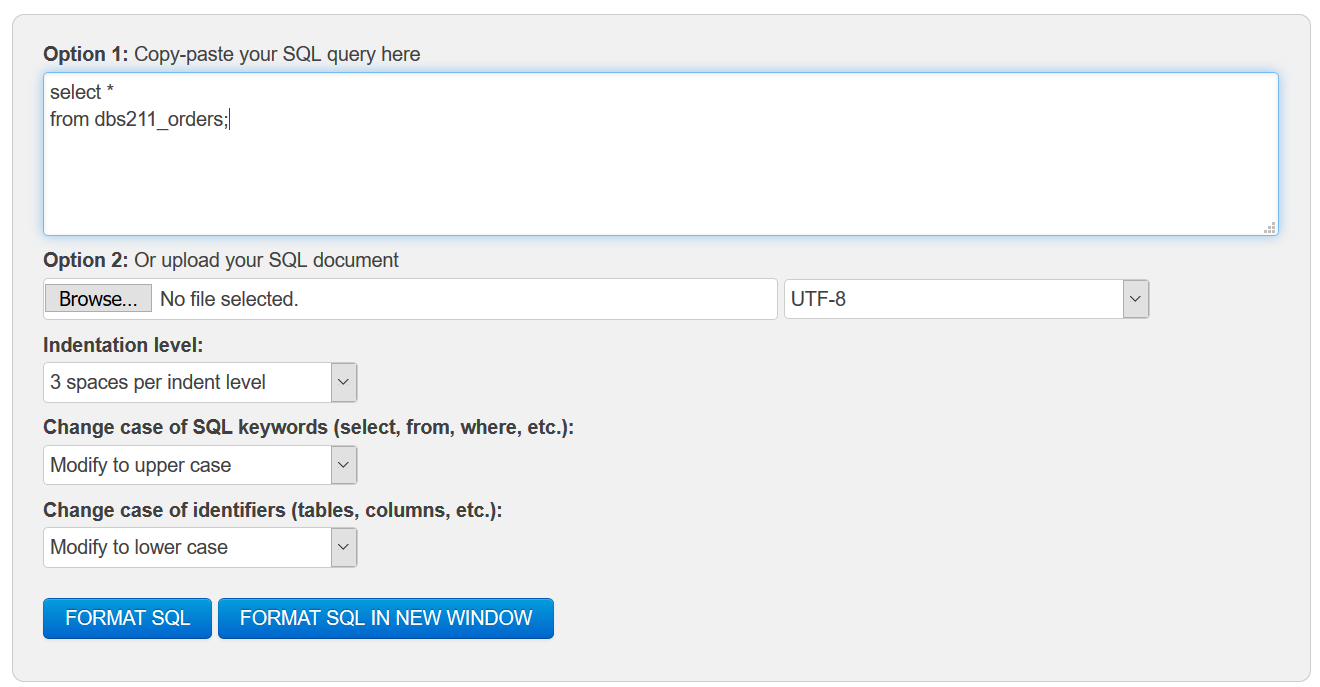
WHERE conditions

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** | 1.5 | **6.a** | 1.5 |
| **2** | 1.5 | **6.b** | 0.5 |
| **3** | 1.5 |  |  |
| **4** | 2 |  |  |
| **5** | 1.5 |  |  |

Total: 10

**Grade Policy**

* Questions with errors do not get any marks. (They get zero.)
  + Execute your *.sql* file by selecting all the statements and using the “Run statement” button to make sure there is no errors in your file and check the output result returned by each query.
* Make sure your result in a question matches the sample output result.
* You do not receive marks for the missing or incomplete solutions.
* Late penalty of 10% marks will apply for each day.

**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.

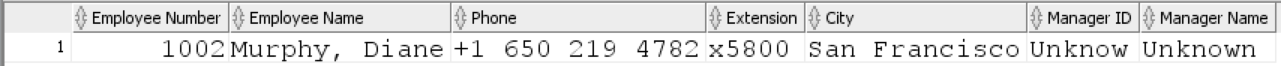
IF you do not have a solution for any question, write the following statement for that question.

See the example:

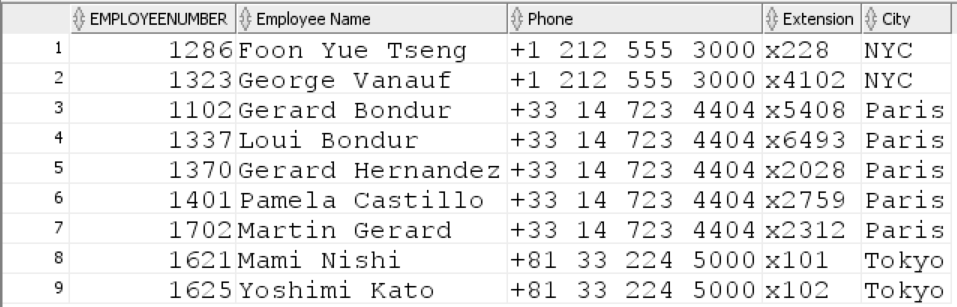
**SELECT 'No Solution'**

**FROM dual;**

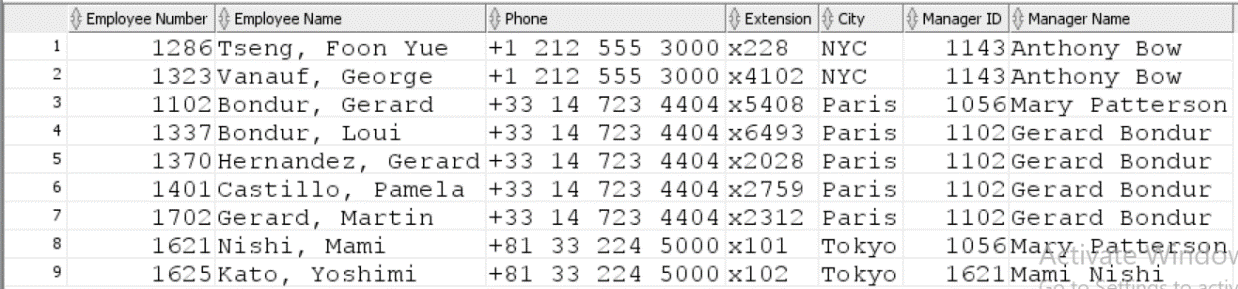
1. Display employee number, employee full name, phone, extension, city, manager ID, and manager name for employees who do not report to a manager. (See the following output) *Sort the result based on the city and the employee number*. **(1.5 Marks)**



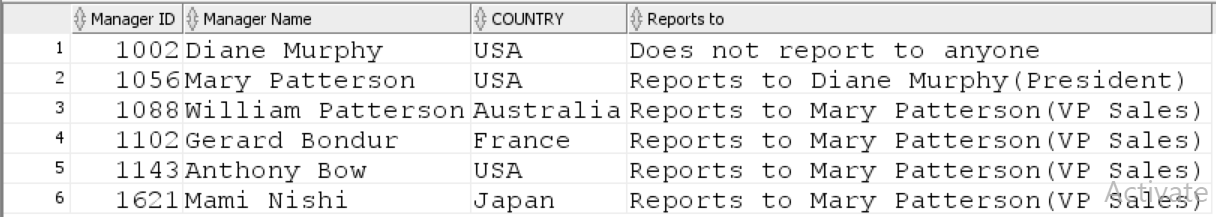
1. Display employee number, employee full name, phone, extension, and city for employees who work in NYC, Tokyo, and Paris. *Sort the result based on the city and the employee number*. **(1.5 Marks)**



1. Modify the query in Question 2 to display the manager ID and the manager name for the employees returned by the previous query. *Sort the result based on the city and the employee number.* **(1.5 Marks)**



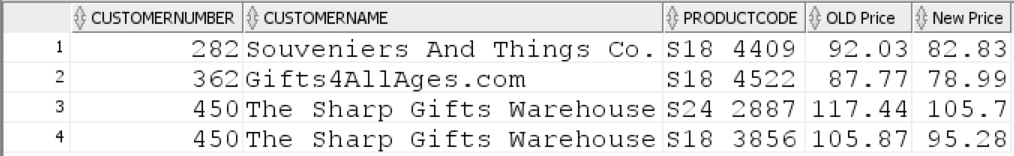
1. For all managers, display manager ID, manager name, country, and the person who the manager reports to. See the following output. *Sort the result according to the manager ID*. **(2 Marks)**



1. The vendor 'Exoto Designs' offers a discount on some particular products. The customers who purchased products with quantity higher than 55 from this vendor are eligible to receive the discount. They will get 10% discount to purchase the same products they have purchased with high quantity (> 55).

Use the msrp column in the product table as the old price and calculate the new price using this column. In your report, display customernumber, customername, productcode, old price, and the new price **(1.5 Marks)**

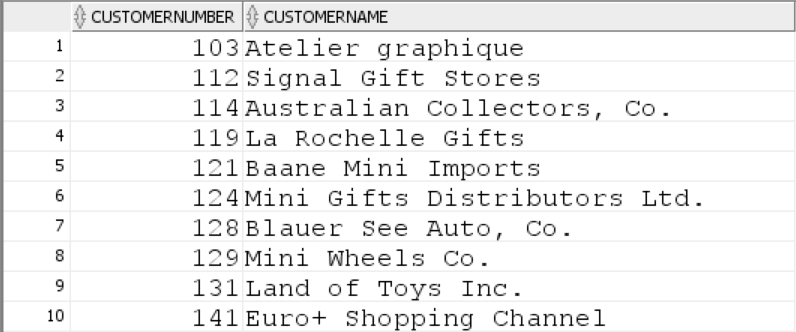
**Hint:** First, find products from this vendor that were sold with quantity greater than 55. Then, you need to find the customers who purchased them. For these products, calculate the new price.



1. Answer the following questions:
   1. Find customers who have more than one order: **(1.5 Marks)**

Show customer number and customer name in your result. Sort the result according to customer number. (Do not show duplicate values)

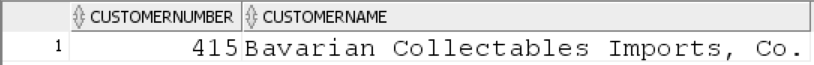
The query returns 97 rows. Please see the first 10 rows of the result. (You query must return all 97 rows.)



Do not use aggregate functions (COUNT()) to answer this question.

**Hint*:*** To find customers with different order numbers, you need to join the table dbs211\_orders with itself. (If for the same customer, there are different orders.) To get the given result, you may need to join more tables. (**0.5 Marks**)

* 1. Modify the query in Q6.a to return customers who have only one order. Sort the result by customer number and customer name. The query returns one customer.



Good Luck