Guided Lab - 304.5.2 - Operators

# Objective:

# In this lab, we will demonstrate and utilize common SQL operators.

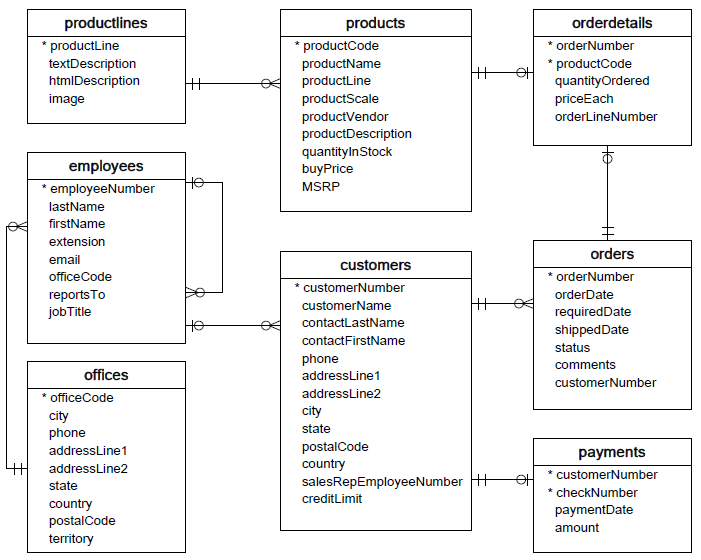
# **Learning Objective:**

After this lab, learners will have demonstrated the ability to use SQL Operators.

**Prerequisites:**

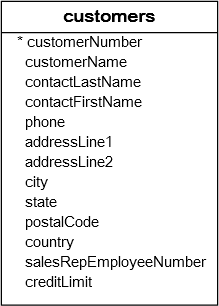
For this lab, you must have a “**classicmodels”** database. If you do not have a ‘**classicmodels ‘** database setup, [**click here to download the database script file**.](https://drive.google.com/file/d/1JoT6N-kNhJ048ahXvvSgWNE0737NAdbb/view?usp=sharing)

**The classicmodels database schema**



# Example: “OR” Operator Examples

Consider the customer's table.



For example, to get the customers who are located in the USA or France, you can use the **OR** operator in the WHERE clause as follows:

|  | SELECT    customername, country  FROM    customers  WHERE    country = 'USA' OR country = 'France'; |
| --- | --- |

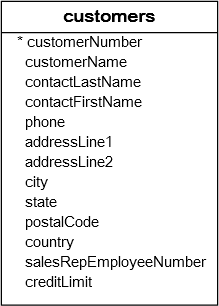
**Result:**

 As you can see in the result, the query returns customers who are located in either the USA or France.

# Example: “AND” Operator Examples

The AND operator is a logical operator that combines two or more **Boolean** expressions, and returns true only if both expressions evaluate as true. The AND operator returns false if one of the two expressions evaluates as false.

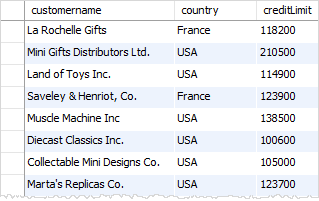
Let’s use the customer's table.



1. The following query will return the customer's records who are located in the USA or France and have credit limits greater than 100,000.

|  | SELECT   customername, country, creditLimit  FROM    customers  WHERE (country = 'USA' **OR** country = 'France') AND creditlimit > 100000; |
| --- | --- |

**Result**



1. The below query will return the customers who are located in the USA or the customers who are located in France with a credit limit greater than 10000.

|  | SELECT  customername, country, creditLimit FROM    customers  WHERE    country = 'USA' **OR** country = 'France' AND creditlimit > 100000; |
| --- | --- |

*Note: We did not use the parentheses.*

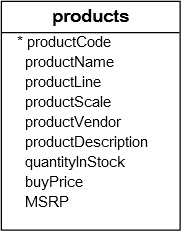
**Result**



# Example: BETWEEN and NOT BETWEEN

* The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.
* The BETWEEN operator is inclusive: Begin and End values are included.

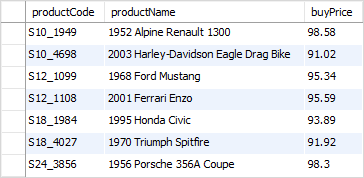
Consider the following products table in the classicmodels database:



1. Suppose you want to find products whose buy prices are within the ranges of 90 and 100. You can use the BETWEEN operator, as in the following query:

|  | SELECT     productCode, productName, buyPrice  FROM    products  WHERE     buyPrice **BETWEEN** 90 **AND** 100; |
| --- | --- |

**Result**



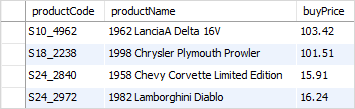
1. You can achieve the same effect by using the greater than or equal to (>=) and the less than or equal to ( <= ) operators, as in the following query:

|  | SELECT productCode, productName, buyPrice  FROM products  WHERE buyPrice **>=** 90 AND buyPrice **<=** 100; |
| --- | --- |

1. To find the product whose buy price is not between $20 and $100, you can combine the BETWEEN operator with the NOT operator as follows:

|  | SELECT productCode, productName, buyPrice FROM products  WHERE buyPrice **NOT BETWEEN** 20 **AND** 100; |
| --- | --- |

**Result**



You can rewrite the query above using the less than (>), greater than (>), and logical operator (AND) as the following query.

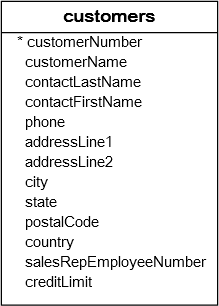
|  | SELECT productCode, productName, buyPrice FROM products  WHERE buyPrice **<** 20 **OR** buyPrice **>** 100; |
| --- | --- |

## 

# 

# 4) Example: “IS NULL” Operator

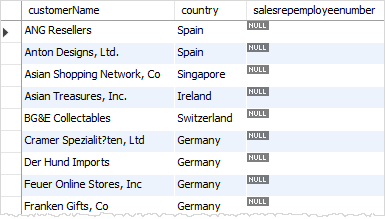
### Consider the customers table in the “classicmodels” database:



1. The following query uses the **IS NULL** operator to find customers who do not have a sales representative:

| **SELECT** customerName, country, salesrepemployeenumber  **FROM** customers  **WHERE** salesrepemployeenumber **IS** NULL  **ORDER** **BY** customerName; |
| --- |

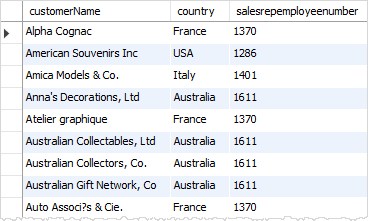
**Result:**



1. The below query uses the **IS NOT NULL** operator to get the customers who have a sales representative:

| **SELECT** customerName, country, salesrepemployeenumber  **FROM** customers  **WHERE** salesrepemployeenumber **IS** **NOT** NULL  **ORDER** **BY** customerName; |
| --- |

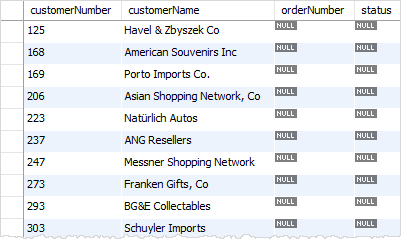
**Result:**



1. The following query uses the **LEFT JOIN predicate** and the IS NULL **operator**  to find customers who have no order:

| SELECT c.customerNumber, c.customerName, orderNumber, o.STATUS  FROM customers c  LEFT JOIN orders o  ON c.customerNumber = o.customerNumber  WHERE orderNumber IS NULL; |
| --- |

**Result:**



**Canvas submission Instructions:** Please include the following deliverables in your submission -

* + All queries should be written and submitted in a single SQL script file named, for example :**<your\_name\_labname>.sql**.
  + Submit your SQL script file using the **Start** **Assignment** button in the top-right corner of the assignment page in Canvas.

**CANVAS STAFF USE ONLY: Canvas Submission Guideline:**

| **Instructions for Canvas Assignment Creation** |
| --- |
| **Assignment Name: GLAB - 304.5.2 - Operators**  **Points:** **100**  **Assignment Group: Module 304 - Relational Databases and SQL - (Not Graded)**  **Display Grade As: Complete/Incomplete**  **Do not count this assignment towards the final grade: Checked**  **Submission Types: Files Uploads**  **Allowed Attempts: Unlimited**  **Everything else is the default.** |

## 

# 