Guided Lab 304.4.1
ORDER BY Clause

Introduction

When you use the **`SELECT`** statement to query data from a table, the order of rows in the result set is unspecified. To sort the rows in the result set, add the **`ORDER BY`** clause to the **`SELECT`** statement.

Learning Objectives

This lab will demonstrate how to sort the rows in a result set using the **`ORDER BY`** clause. By the end of this lab, learners will be able to use the **`ORDER BY`** clause in SQL.

Prerequisites

For this lab, you must have the "**`classicmodels`**" database. If you do not have the **`classicmodels`** database setup, **click here to download the database script file.**

Instructions

Example 1: Sort a result set by an expression.

See the following **orderdetails** table from the sample database:

orderdetails
* orderNumber * productCode quantityOrdered priceEach orderLineNumber

The following query selects the order line items from the **orderdetails** table. It calculates the subtotal for each line item, and sorts the result based on the subtotal.

```
Unset
SELECT    orderNumber, orderlinenumber, quantityOrdered * priceEach
FROM    orderdetails
ORDER BY   quantityOrdered * priceEach DESC;
```

Output

orderNumber	orderLineNumber	quantityOrdered * priceEach
10403	9	11503.14
10405	5	11170.52
10407	2	10723.60
10404	3	10460.16
10312	3	10286.40

To make the query more readable, you can assign an **alias** to a column in the **SELECT** statement and use that column alias in the **ORDER BY** clause:

```
Unset
SELECT
    orderNumber,
    orderLineNumber,
    quantityOrdered * priceEach AS subtotal
FROM    orderdetails
ORDER BY subtotal DESC;
```

In this example, we use **subtotal** as the column alias for the expression **quantityOrdered * priceEach,** and sort the result set by the **subtotal** alias.

Output

orderNumber	orderLineNumber	subtotal
10403	9	11503.14
10405	5	11170.52
10407	2	10723.60
10404	3	10460.16
10312	3	10286.40

Example 2: MySQL ORDER BY and NULL Values

In SQL, **NULL** comes before non-**NULL** values. Therefore, when you use the **ORDER BY** clause with the **ASC** option, **NULLs** appear first in the result set.

For example, the following query uses the **ORDER BY** clause to sort employees by values in the **reportsTo** column:

```
Unset
SELECT    firstName, lastName, reportsTo
FROM    employees
ORDER BY reportsTo;
```

Output

firstName	lastName	reportsTo
Diane	Murphy	NULL
Mary	Patterson	1002
Jeff	Firrelli	1002
William	Patterson	1056
Gerard	Bondur	1056

However, if you use the **ORDER BY** clause with the **DESC** option, **NULLs** will appear last in the result set. For example:

```
Unset
SELECT    firstName, lastName, reportsTo
FROM    employees
ORDER BY reportsTo DESC;
```

The result of this is included on the following page.

Output

firstName	lastName	reportsTo
...	data included before	...
Mami	Nishi	1056
Mary	Patterson	1002
Jeff	Firrelli	1002
Diane	Murphy	NULL

Summary

- Use the **`ORDER BY`** clause to sort the result set by one or more columns.
- Use the **`ASC`** option to sort the result set in ascending order, and the **`DESC`** option to sort the result set in descending order.
- The **`ORDER BY`** clause is evaluated after the **`FROM`** and **`SELECT`** clauses.
- In MySQL, **`NULL`** is lower than non-**`NULL`** values.

Submission

Please include the following deliverables in your Canvas submission:

- All queries, which should be written and submitted in a single SQL script file.
    - Example: **`<your_name_labname>.sql`**.
    - **Do not add the questions in your SQL script file.**

Submit your SQL script file using the **Start Assignment** button on the assignment page in Canvas.