**Changelog Documentation**

|  |  |
| --- | --- |
| **Project Details** | |
| Client | Retail Company |
| **Project** | Retail Company Sales Performance Analysis |
| **Start Data** | 2023-10-26 |
| **End Date** | 2023-11-25 |
| **Email** | [Alaamhassan2001@gmail.com](mailto:Alaamhassan2001@gmail.com) |
| **GitHub** | <https://github.com/alaamhassan> |
| **LinkedIn** | [www.linkedin.com/in/alaamhassan](http://www.linkedin.com/in/alaamhassan) |
| **Portfolio** |  |

Table of Contents

[Metadata 3](#_Toc157472083)

[Changelog 3](#_Toc157472084)

[Cleaning Process 6](#_Toc157472085)

[1. Business Logic (does the data make sense?) 6](#_Toc157472086)

[2. Check for Duplicates 12](#_Toc157472087)

[3. Check for Mistyping 14](#_Toc157472088)

[4. Check for Blank Fields 16](#_Toc157472089)

[5. Check for mismatched data types 20](#_Toc157472090)

[Adjust and Format 30](#_Toc157472091)

# Metadata

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **Consumer ID** | Text | Unique identifier for each customer. |
| **Order ID** | Text | Unique identifier for each order, where the first part is the Consumer ID and the second is a unique alphabetic char. |
| **Month** | Text | The month when the order was placed. |
| **Year** | Text | The year when the order was placed. |
| **Total order value** | Number | The total value of the order before any discounts are applied. |
| **Discount** | Number | The amount of discount applied to the order. |
| **Line Value (net discount)** | Number | The value of the order after the discount has been applied. |
| **Line Category** | Text | The category of the product in the order. |
| **Line SKU** | Text | Unique identifier for each product. |
| **Line SKU Production Cost** | Number | The production cost of the product. |

# Changelog

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Change ID** | **Reason** | **Title of Change** | **Description** | **# Values** | **Process**  **Link** | **Version**  **Link** |
| **C1** | Data constraint violation. | Fixing mistyping of ‘Line SKU Production Cost’ Value. | Change Line SKU Production Cost from 432,000 € to 43.2 €. | 1 | [C1-P](#C1_P) | [C1-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/44bc2d7178ee4e43471d926f714bceb54e3c5ad2) |
| **C2** | Data Mistyping. | Fixing mistyping of ‘Line Category’ value. | Change ‘Line Category’ from Mini bag to Mini bags. | 3 | [C2-P](#C2_P) | [C2-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/4775555da2323c1b3277ec8e2369a74b49d53c44) |
| **C3** | Blank Data. | Filling Values for ‘Order ID’. | Fill Blank Fields to contain the ‘Consumer ID’ plus the two characters I and J. | 2 | [C3-P](#C3_P) | [C3-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/68a71176357a0a3d57d6c4cf6864acbff6a37464) |
| **C4** | Blank Data. | Filling Values for ‘Line SKU Production Cost’. | Fill Blank Field of ‘MIN3’ Line SKU to contain 247.26 €. | 1 | [C4-P](#C4_P) | [C4-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/a3abcdb7bb7848a8e645284c8b17339ea9188216) |
| **C5** | Blank Data. | Filling Values for ‘Line SKU Production Cost’. | Fill Blank Field of ‘MIN4’ Line SKU to contain 293.22 €. | 1 | [C5-P](#C5_P) | [C5-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/9540f8819414f51ac9ae31b4fd1752965ddad3cf) |
| **C6** | Blank Data. | Filling Values for ‘Line SKU Production Cost’. | Fill Blank Field of ‘MIN2’ Line SKU to contain 222.25 €. | 1 | [C6-P](#C6_P) | [C6-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/cbeeaaa8acd894b5422f963e4e2188aa694995ed) |
| **C7** | Blank Data. | Filling Values for ‘Line SKU Production Cost’. | Fill Blank Field of ‘MIN8’ Line SKU to contain 210.4 €. | 1 | [C7-P](#C7_P) | [C7-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/a84bf7e9dec151761f59d83629aeb89e54967cde) |
| **C8** | Mismatched Data Type. | Converting the data type of “Consumer ID”. | Convert the data type of “Consumer ID” from General to Text. | 15 | [C8-P](#C8_P) | [C8-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/1c59b9d98f4e35358907e5f0b6e1cf665bd11c22) |
| **C9** | Mismatched Data Type. | Converting the data type of “Order ID”. | Convert the data type of “Consumer ID” from General to Text. | 3678 | [C9-P](#C9_P) | [C9-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/8b7b2675d8c82e427f06fa6f2cf99a33b1f52c09) |
| **C10** | Mismatched Data Type. | Converting the data type of “Month”. | Convert the data type of “Month” from General to Text. | 3678 | [C10-P](#C10_P) | [C10-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/c11afa9b7b147d8f79e1c86a5e73a18bc3643af7) |
| **C11** | Mismatched Data Type. | Converting the data type of “Year”. | Convert the data type of “Year” from General to Text. | 3678 | [C11-P](#C11_P) | [C11-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/521e05a625ad8352853043a089a06f0f2e92422b) |
| **C12** | Mismatched Data Type. | Converting the data type of “Total order value”. | Convert the data type of “Total order value” from Currency to Number. | 3678 | [C12-P](#C12_P) | [C12-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/89c65afe961d47afff738cf70fd6aad53552b440) |
| **C13** | Mismatched Data Type. | Converting the data type of “Discount”. | Convert the data type of “Discount” from Currency to Number. | 3678 | [C13-P](#C13_P) | [C13-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/9e6dd5bf615cc680bd54c9a7afbc9ba09568f879) |
| **C14** | Mismatched Data Type. | Converting the data type of “Line Value (net discount)”. | Convert the data type of “Line Value (net discount)” from Currency to Number. | 3678 | [C14-P](#C14_P) | [C14-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/c46d3d6bf9533800a07918cd546e9b954274a698) |
| **C15** | Mismatched Data Type. | Converting the data type of “Line Category”. | Convert the data type of “Line Category” from General to Text. | 3678 | [C15-P](#C15_P) | [C15-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/3b4fedbaa53496829cb741a183f2b4c6306e8468) |
| **C16** | Mismatched Data Type. | Converting the data type of “Line SKU”. | Convert the data type of “Line SKU” from General to Text. | 3678 | [C16-P](#C16_P) | [C16-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/50a4e5613b8f674998d1c02a5abf534b3709f765) |
| **C17** | Mismatched Data Type. | Converting the data type of “Line SKU Production Cost”. | Convert the data type of “Line SKU Production Cost” from Currency to Number. | 3678 | [C17-P](#C17_P) | [C17-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/3b7c2198bd46ff08ac3ee68495b509837487d1a8) |
| **C18** | Adjusting columns for analysis. | Combine the “Month” and “Year” columns. | Combine “Month” and “Year” into “Date”, then remove the original columns. | 3678 | [C18-P](#C18_P) | [C18-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/a3a65d3c3c797806e70809816097943072868915) |
| **C19** | Adjusting columns for analysis. | Adding a “Net Profit” column. | Adding a Adding a “Net Profit” column using the formula “Line Value (net discount)” minus “Line SKU Production Cost”. | 3678 | [C19-P](#C19_P) | [C19-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/4f98743fb38ac8af79b40be059864ebdae643efc) |
| **C20** | Adjusting columns for analysis. | Adding a “Is New Consumer” column. | Adding a “Is New Consumer” column which equal ‘1’ in case of a new customer, ‘0’ otherwise. | 3678 | [C20-P](#C20_P) | [C20-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/8b40b8df39a245d0a3f564ae42ce0d5a6fbe6ba8) |
| **C21** | Adjusting columns for analysis. | Adding a “Is Last Order” column. | Adding a “Is Last Order” column which equal ‘1’ in case of a last order made by a customer, ‘0’ otherwise. | 3678 | [C21-P](#C21_P) | [C21-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/607a995982835d7a0d9ef3de6361b4b5052d6e7f) |
| **C22** | Adjusting columns for analysis. | Adding a “Is Discount” column. | Adding a “Is Discount” column which equal ‘1’ in case of the order is discounted, ‘0’ otherwise. | 3678 | [C22-P](#C22_P) | [C22-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/ad54f0c1276f355c37084865051d5c9ecc691887) |
| **C23** | Adjusting columns for analysis. | Adding a “Discount Category” column. | Adding a “Discount Category” column which represents the percentage of the order discount out of the total order value. | 3678 | [C23-P](#C23_P) | [C23-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/c04eb259e830e8fa6c6afb396f93bce71055c57b) |
| **C24** | Adjusting columns for analysis. | Adding a “Number of Orders” column. | Adding a “Number of Orders” column which represents the total number of orders purchased by a customer. | 3678 | [C24-P](#C24_P) | [C24-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/cb8b8e1f8d811b8e491f8b7a5c488bbdd7bcd41a) |
| **C25** | Adjusting columns for analysis. | Adding a “Order Number” column. | Adding a “Order Number” column which represents the current order number purchased by a customer. | 3678 | [C25-P](#C25_P) | [C25-V](https://github.com/alaamhassan/RetailCompany_PerformanceAnalysis/tree/179115c229921171445ea1c94e85cb43346dcf74) |

# Cleaning Process

## Business Logic (does the data make sense?)

Constrains based on data:

* Order ID = Consumer ID + Alphabetic Char
* Total order value >= Discount
* Line Value (net discount) = Total order value – Discount
* Line SKU Production Cost < Line Value (net discount)

Each constrain was checked using **conditional formatting**:

1. Order ID = Consumer ID + Alphabetic Char

To check for this constraint:

1. Add a new column called ‘Order ID - First Part’.
2. Extract the first part of the Order ID using both the **LEFT** and **LEN** function.
3. A screenshot of a computer

   Description automatically generatedUse **conditional formatting** to check the constraint.

**Output**:

There are 15 values highlighted as a violation, although they don’t seem as such.

A screenshot of a computer

Description automatically generated

**Observation**:

A further investigation has found that the highlighted values are because of different data types rather than violating the data constraint.

* A screenshot of a computer

  Description automatically generated“Consumer ID” that didn’t trigger a highlight in the “Order ID – First Part” was stored as text.
* A screenshot of a computer

  Description automatically generated“Consumer ID” that did trigger a highlight in the “Order ID – First Part” wasn’t stored as text.

**Correctness**:

**Steps**:

1. A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generatedConvert the data type of the “Consumer ID” from General to Text using **Text to Columns** feature.

**NOTE:**

* The Text to Columnsfeature was used rather than converting the type simply from the home tap, because it would require re-entering the data again.
* A screenshot of a computer

  Description automatically generatedThis change occurred as the 8th change.

2) Total order value >= Discount

A screenshot of a computer

Description automatically generated

**Output**:

No value violates the constraint.

3) Line Value (net discount) = Total order value – Discount

A screenshot of a computer

Description automatically generated

**Output**:

No value violates the constraint.

4) Line SKU Production Cost < Line Value (net discount)

A screenshot of a computer

Description automatically generated

**Output**:  
one value violates the constraint



**Correctness**:

**Steps**:

1. Filter for the ‘FRA5’ Line SKU

A screenshot of a computer

Description automatically generated

**Observations**:

170 records have the category ‘FRA5’. All these records have a Line SKU Production Cost of 43.2 €.

1. Change the Line SKU Production Cost of the first record from 432,000 € to 43.2 €.

A screenshot of a computer

Description automatically generated

## Check for Duplicates

Unique Constraints:

* A Record can’t be duplicated.
* Order ID can’t appear twice.

Check for each constrain using **Remove Duplicates** and **Conditional Formatting**:

1. A Record can’t be duplicated.

A screenshot of a computer

Description automatically generateduse **Remove Duplicates** to find and remove duplicate records.

**Output**:

A screenshot of a computer error

Description automatically generated No record violates the constraint.

1. Order ID can’t appear twice.

Check for duplicates using **conditional formatting**:

A screenshot of a computer

Description automatically generated

**Output**:

Two values violate the constraint.



as the value of the Order ID is N/A, no values will be removed.

## Check for Mistyping

In the ‘Line Category’ column there are two categories with the same name:

* Mini bags
* Mini bag

A screenshot of a computer

Description automatically generatedbut one is plural and the other is singular.

**Correctness:**

As the two ‘Line Category’ have records with the same ‘Line SKU’. Then the two are likely the same.

**steps:**

1. Filter for the ‘Mini bag’ Line Category.
2. Change ‘Mini bag’ to ‘Mini bags’ using **Find and Replace**.

A screenshot of a computer

Description automatically generated

**Output**:

A screenshot of a computer error

Description automatically generatedThree values were changed.

1. Validate using the Filter Menu.

A screenshot of a computer

Description automatically generated

## 4. Check for Blank Fields

There are blanks fields in:

* Order ID
* Line SKU Production Cost

Let’s investigate each column:

1. Order ID

Use Filtration to find Blank fields.

**Correctness**:

**Steps:**

1. A screenshot of a computer

   Description automatically generatedAs the blanks field were for the ‘Consumer ID’ => 13134019, Filter only this one.

1. A screenshot of a computer

   Description automatically generatedChange the two blank fields to contain the Consumer ID plus the two characters I and J.
2. Line SKU Production Cost

Use Filtration to find Blank fields.

**Correctness**:

**Steps:**

1. Filter for ‘Line SKU’ equal MIN3.

A screenshot of a computer

Description automatically generated

**Observations**:

27 records have the Line SKU ‘MIN3’. All these records have a Line SKU Production Cost of 247.26 €.

A screenshot of a computer

Description automatically generatedFill the ‘Line SKU Production Cost’ of the ‘MIN3’ to 247.26 €.

1. A screenshot of a computer

   Description automatically generatedFilter for ‘Line SKU’ equal MIN4.

**Observations**:

33 records have the Line SKU ‘MIN4’. All these records have a Line SKU Production Cost of 293.22 €.

Fill the ‘Line SKU Production Cost’ of the ‘MIN4’ to 293.22 €.

A screenshot of a computer

Description automatically generated

1. A screenshot of a computer

   Description automatically generatedFilter for ‘Line SKU’ equal MIN2.

**Observations**:

29 records have the Line SKU ‘MIN2’. All these records have a Line SKU Production Cost of 222.25 €.

A screenshot of a computer

Description automatically generatedFill the ‘Line SKU Production Cost’ of the ‘MIN2’ to 222.25 €.

1. A screenshot of a computer

   Description automatically generatedFilter for ‘Line SKU’ equal MIN8.

**Observations**:

17 records have the Line SKU ‘MIN8’. All these records have a Line SKU Production Cost of 210.4 €.

A screenshot of a computer

Description automatically generatedFill the ‘Line SKU Production Cost’ of the ‘MIN8’ to 210.4 €.

## Check for mismatched data types

From the Metadata:

* Consumer ID => Text
* Order ID => Text
* Month => Text
* Year => Text
* Total order value => Number
* Discount =>Number
* Line Value (net discount) => Number
* Line Category => Text
* Line SKU => Text
* Line SKU Production Cost => Number

As only the Consumer ID was in the right data type, the rest column’s data types will be converted.

**Note**:

1. Order ID => Text

This column will be converted from General to Text using the **Text to Columns** feature.

**NOTE:** The Text to Columnsfeature was used rather than converting the type simply from the home tap, because it would require re-entering the data again.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Month => Text

This column will be converted from General to Text using the **Text to Columns** feature.

**NOTE:** The Text to Columnsfeature was used rather than converting the type simply from the home tap, because it would require re-entering the data again.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

3. Year => Text

This column will be converted from General to Text using the **Text to Columns** feature.

A screenshot of a computer

Description automatically generated**NOTE:** The Text to Columnsfeature was used rather than converting the type simply from the home tap, because it would require re-entering the data again.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Total order value => Number

This column will be converted from Currency to Number.

**Steps**:

1. A screenshot of a computer

   Description automatically generatedconvert from currency to Number from the home tap.
2. As the numbers are still aligned at left, to make it right aligned as the standard, we will make the alignment general instead of right.

A screenshot of a computer

Description automatically generated

1. Discount => Number

This column will be converted from Currency to Number.

**Steps**:

1. A screenshot of a computer

   Description automatically generatedconvert from currency to Number from the home tap.
2. As the numbers are still aligned at left, to make it right aligned as the standard, we will make the alignment general instead of right.

A screenshot of a computer

Description automatically generated

1. Line Value (net discount) => Number

This column will be converted from Currency to Number.

**Steps**:

1. A screenshot of a computer

   Description automatically generatedconvert from currency to Number from the home tap.
2. As the numbers are still aligned at left, to make it right aligned as the standard, we will make the alignment general instead of right.

A screenshot of a computer

Description automatically generated

1. Line Category => Text

This column will be converted from General to Text using the **Text to Columns** feature.

A screenshot of a computer

Description automatically generated**NOTE:** The Text to Columnsfeature was used rather than converting the type simply from the home tap, because it would require re-entering the data again.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Line SKU => Text

This column will be converted from General to Text using the **Text to Columns** feature.

A screenshot of a computer

Description automatically generated**NOTE:** The Text to Columnsfeature was used rather than converting the type simply from the home tap, because it would require re-entering the data again.

A screenshot of a computer

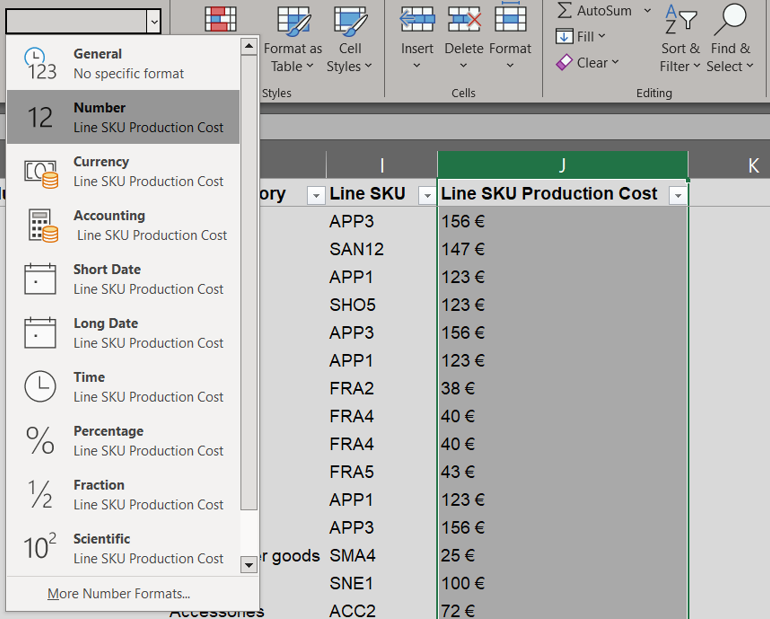
Description automatically generatedA screenshot of a computer

Description automatically generated

1. Line SKU Production Cost => Number

This column will be converted from Currency to Number.

**Steps**:

1. convert from currency to Number from the home tap.
2. As the numbers are still aligned at left, to make it right aligned as the standard, we will make the alignment general instead of right.

A screenshot of a computer

Description automatically generated

# Adjust and Format

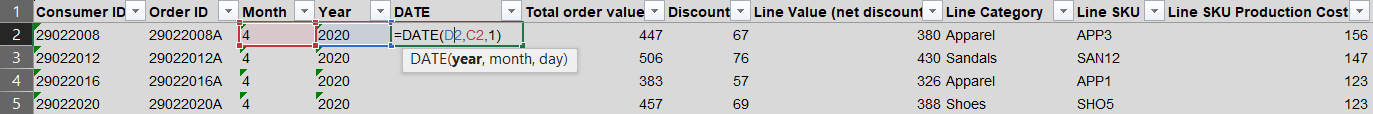
In this process, we will add any needed columns for the analysis process.

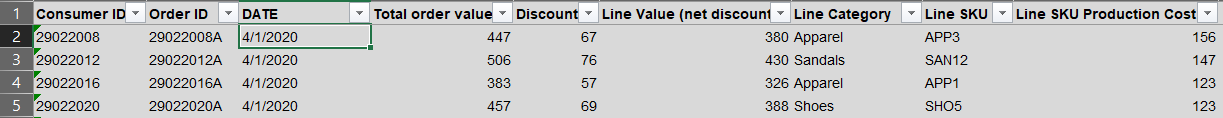
From the requirements, we need eight additional columns.

1. Date

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **Date** | Date “m/d/yyyy” | Merging of the “Month” and “Year” columns. |
| **Net Profit** | Number | Equal the “Line Value (net discount)” Minus “Line SKU Production Cost”. |
| **Is New Consumer** | Number | A Boolean variable equals ‘1’ if this is the first order made by a customer, and equals ‘0’ for any next order by the same customer. |
| **Is Last Order** | Number | A Boolean variable equals ‘1’ if this is the last order made by a customer, and equals ‘0’ otherwise. |
| **Is Discount** | Number | A Boolean variable equals ‘1’ if the order is discounted, ‘0’ otherwise. |
| **Discount Category** | General | A number representing the percentage of the order discount out of the total order value. |
| **Number of Orders** | Number | A Number representing the total number of orders purchased by a customer. |
| **Order Number** | Number | A Number representing the current order number purchased by a customer. |

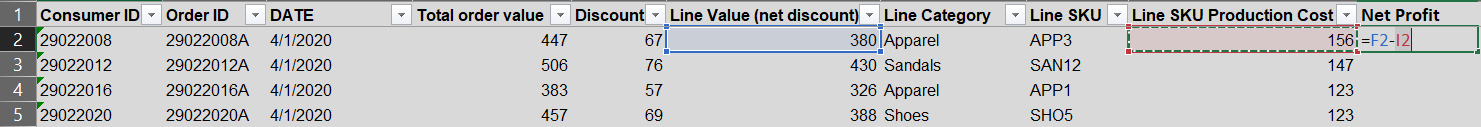
**Steps**:

1. Create a new column called “Date” and use the **DATE** function.
2. A screenshot of a computer

   Description automatically generatedCopy and paste as value.
3. Remove the “Month” and “Year” column, as they are no longer needed.

2. Net Profit

**Steps**:

1. Create a new column called “Net Profit” by applying the formula “Line Value (net discount)” minus “Line SKU Production Cost”.
2. A screenshot of a computer

   Description automatically generatedCopy and paste as value.

3. Is New Consumer

**Steps**:

A screenshot of a computer

Description automatically generated 1. Create a new column called “Is New Consumer” and use **XLOOKUP**, **MIN**, and **IF** functions.

**Explanation**:

* **Intuition**: The function is designed to check if the current Order ID (B2) is the one associated with the earliest date for the current Consumer ID (A2). If it is, the function returns 1; otherwise, it returns 0. In case of a tie (i.e., multiple orders from the same Consumer ID have the same earliest date), the function picks the Order ID that appears first in the OrderID\_Array.
* **XLOOKUP**(lookup\_value, lookup\_array, return\_array, [if\_not\_found], [match\_mode], [search\_mode]):
  + lookup\_value: This is the earliest date for the current Consumer ID (A2). It’s found by taking the minimum date from the DATE\_Array where the ConsumerID\_Array equals A2. If the ConsumerID\_Array doesn’t equal A2, the function uses 45261 (which represents a date far in the future, assuming Excel’s standard date system where 1 represents January 1, 1900).
  + lookup\_array: This is the same array used to find the lookup\_value.
  + return\_array: This is the OrderID\_Array from which the XLOOKUP function will return an Order ID.
  + [if\_not\_found]: If the lookup\_value is not found in the lookup\_array, the function returns 0.
  + [match\_mode]: This is set to 0 to find an exact match.
  + [search\_mode]: This is set to 1 to search from first to last.
* **IF**(logical\_test, [value\_if\_true], [value\_if\_false]):
  + logical\_test: This checks if the current Order ID (B2) is equal to the Order ID returned from the XLOOKUP function.
  + [value\_if\_true]: If the logical\_test is true (i.e., the current Order ID is the one associated with the earliest date for the current Consumer ID), the function returns 1.
  + [value\_if\_false]: If the logical\_test is false (i.e., the current Order ID is not the one associated with the earliest date for the current Consumer ID), the function returns 0.

2. Copy and paste as value.

A screenshot of a computer

Description automatically generated

4. Is Last Order

**Steps:**

A screenshot of a computer

Description automatically generated 1. Create a new column called “Is Last Order” by using **XLOOKUP**, **MAX**, and **IF** functions.

**Explanation**:

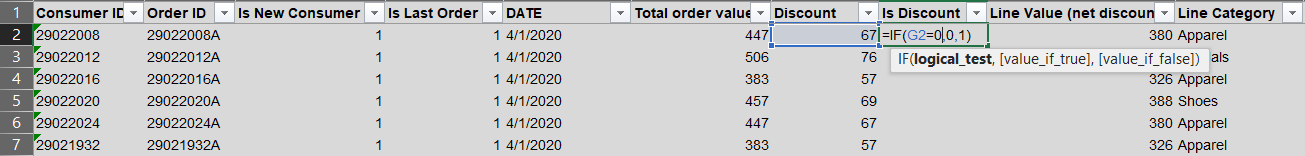
* **Intuition**: The function is designed to check if the current Order ID (B2) is the one associated with the latest date for the current Consumer ID (A2). If it is, the function returns 1; otherwise, it returns 0. In case of a tie (i.e., multiple orders from the same Consumer ID have the same latest date), the function picks the Order ID that appears last in the OrderID\_Array.
* **XLOOKUP**(lookup\_value, lookup\_array, return\_array, [if\_not\_found], [match\_mode], [search\_mode]):
  + lookup\_value: This is the latest date for the current Consumer ID (A2). It’s found by taking the maximum date from the DATE\_Array where the ConsumerID\_Array equals A2. If the ConsumerID\_Array doesn’t equal A2, the function uses 0.
  + lookup\_array: This is the same array used to find the lookup\_value.
  + return\_array: This is the OrderID\_Array from which the XLOOKUP function will return an Order ID.
  + [if\_not\_found]: If the lookup\_value is not found in the lookup\_array, the function returns 0.
  + [match\_mode]: This is set to 0 to find an exact match.
  + [search\_mode]: This is set to -1 to search from last to first.
* **IF**(logical\_test, [value\_if\_true], [value\_if\_false]):
  + logical\_test: This checks if the current Order ID (B2) is equal to the Order ID returned from the XLOOKUP function.
  + [value\_if\_true]: If the logical\_test is true (i.e., the current Order ID is the one associated with the latest date for the current Consumer ID), the function returns 1.
  + [value\_if\_false]: If the logical\_test is false (i.e., the current Order ID is not the one associated with the latest date for the current Consumer ID), the function returns 0.

A screenshot of a computer

Description automatically generated 2. Copy and paste as value.

5. Is Discount

**Steps**:

 1. Create a new column called “Is Discount” by using the **IF** function.

A screenshot of a computer

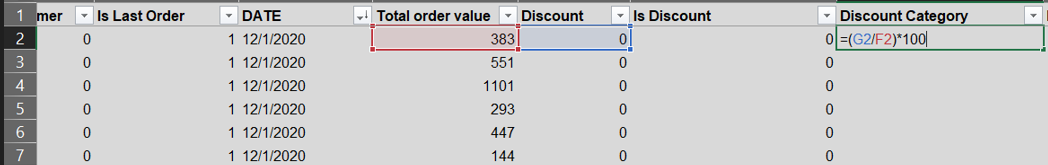
Description automatically generated 2. Copy and paste as value.

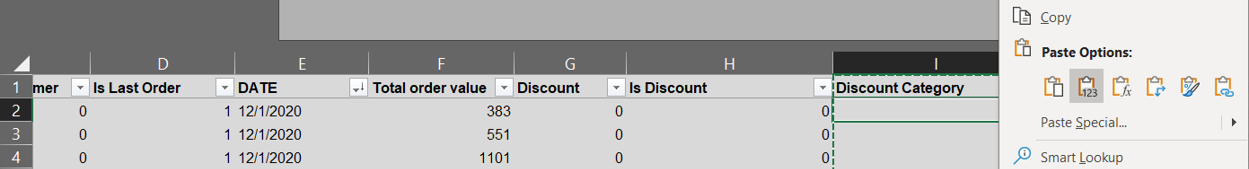
6. Discount Category

**Steps**:

1. Create a new column called “Discount Category” by applying the formula (Discount/Total

order value)\*100.



 2. Copy and paste as value.

7. Number of Orders

**Steps**:

A screenshot of a computer

Description automatically generated 1. Create a new column called “Number of Orders” by using **COUNTIF** functions.

A screenshot of a computer

Description automatically generated 2. Copy and paste as value.

7. Order Number

**Steps**:

A screenshot of a computer

Description automatically generated 1. Create a new column called “Number of Orders” by using **COUNTIF** and **IF** functions.

2. Copy and paste as value.

A screenshot of a computer

Description automatically generated