Table Functions

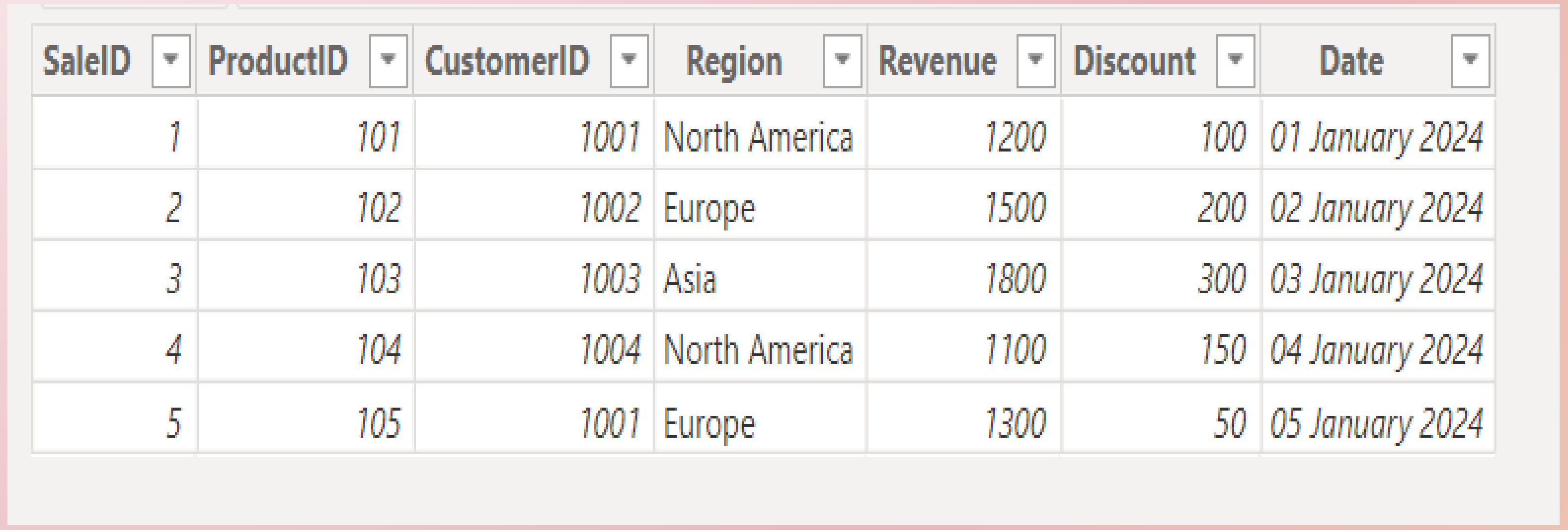
DAX Part-4

What are Table Functions?

Table functions in DAX are powerful tools that allow you to create virtual datasets within Power BI. Table functions return a table as a result instead of a single value. They are powerful for performing complex calculations and transforming data within Power BI.

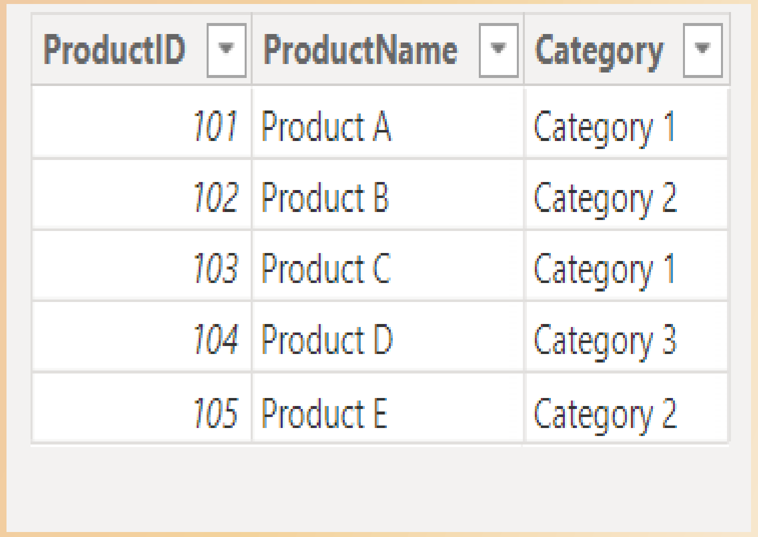
Table used :

Sales



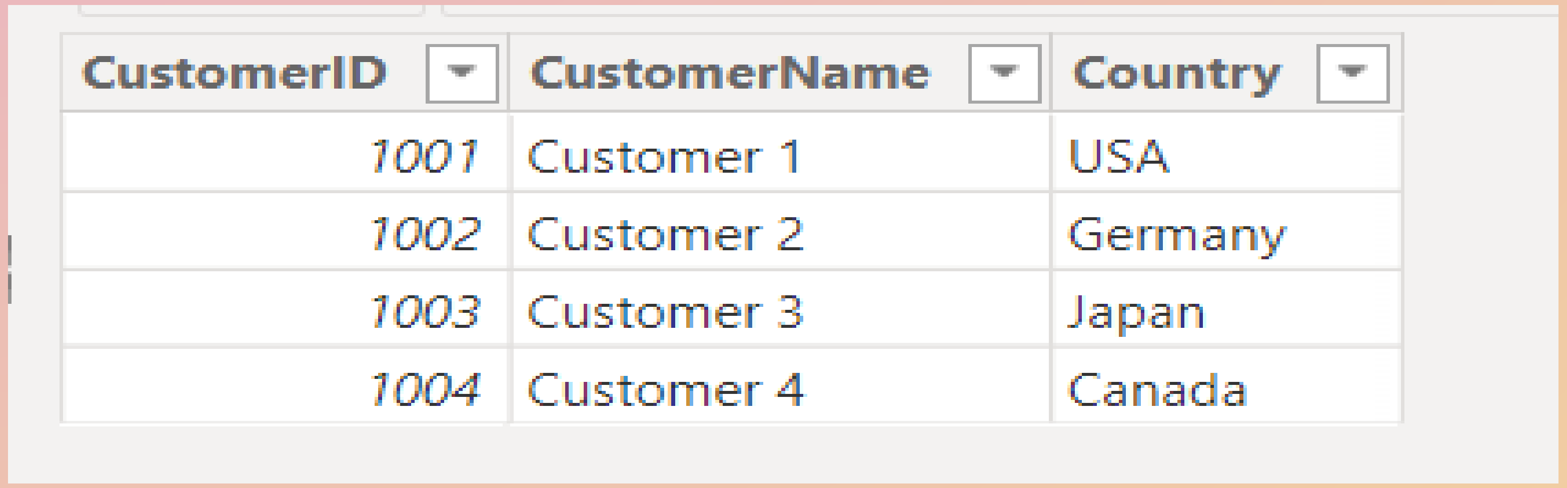
SaleID	ProductID	CustomerID	Region	Revenue	Discount	Date
1	101	1001	North America	1200	100	01 January 2024
2	102	1002	Europe	1500	200	02 January 2024
3	103	1003	Asia	1800	300	03 January 2024
4	104	1004	North America	1100	150	04 January 2024
5	105	1001	Europe	1300	50	05 January 2024

Products



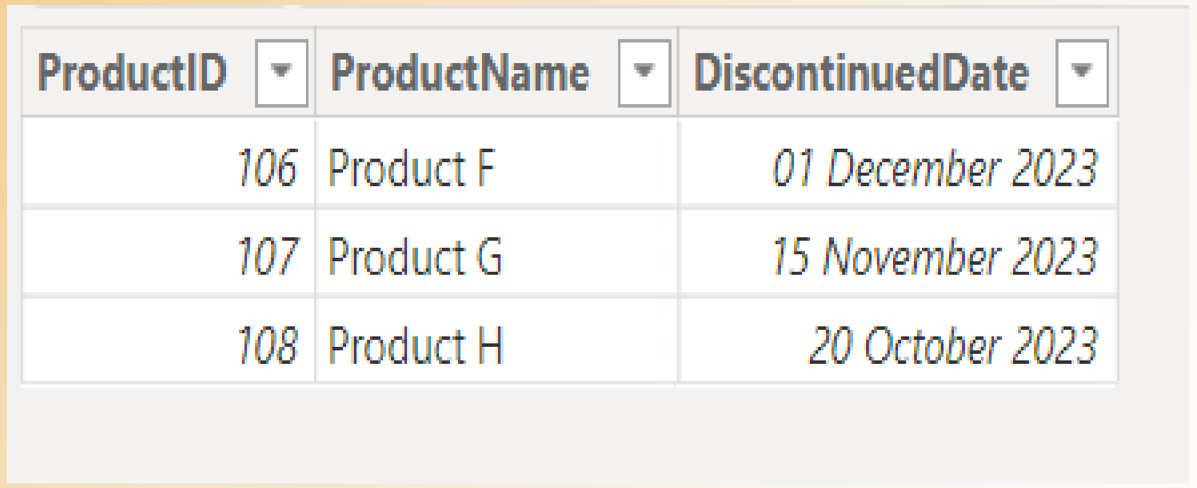
ProductID	ProductName	Category
101	Product A	Category 1
102	Product B	Category 2
103	Product C	Category 1
104	Product D	Category 3
105	Product E	Category 2

Customers



CustomerID	CustomerName	Country
1001	Customer 1	USA
1002	Customer 2	Germany
1003	Customer 3	Japan
1004	Customer 4	Canada

Discontinuedproducts



ProductID	ProductName	DiscontinuedDate
106	Product F	01 December 2023
107	Product G	15 November 2023
108	Product H	20 October 2023

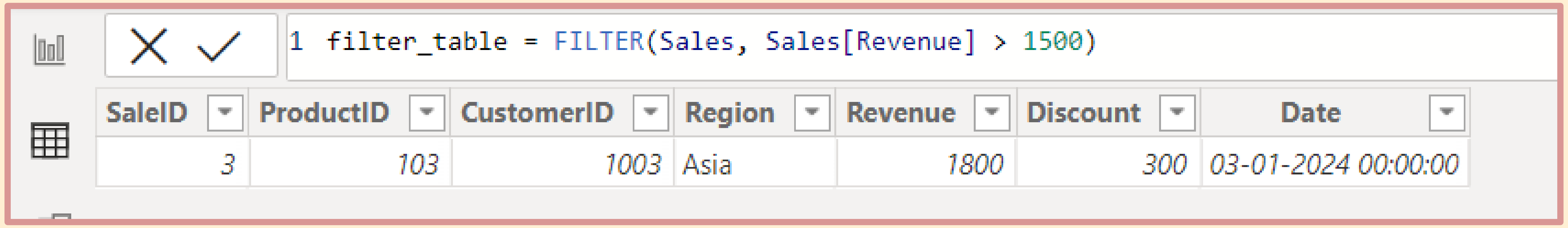
Filter

- **Description :** Returns a table that represents a subset of another table or expression.

- **Example :**

  FILTER(Sales, Sales[Revenue] > 1000)

  This returns a table with rows from the Sales table where the Revenue is greater than 1000.



```
filter_table = FILTER(Sales, Sales[Revenue] > 1500)
```

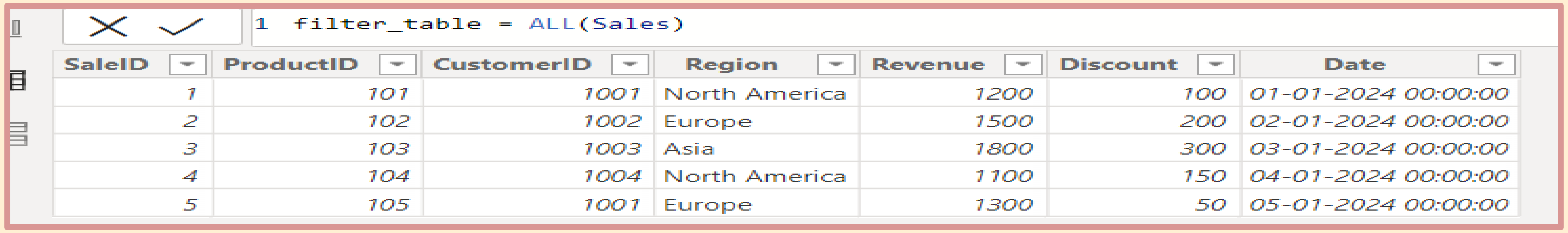
SaleID	ProductID	CustomerID	Region	Revenue	Discount	Date
3	103	1003	Asia	1800	300	03-01-2024 00:00:00

ALL

- **Description :** Removes filters from columns or tables.
- **Example :**

  ALL(Sales)

  This returns the entire Sales table, ignoring any applied filters



```
filter_table = ALL(Sales)
```

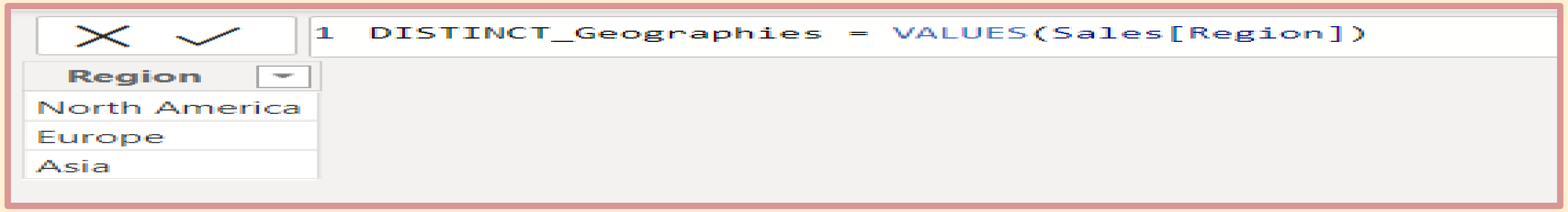
SaleID	ProductID	CustomerID	Region	Revenue	Discount	Date
1	101	1001	North America	1200	100	01-01-2024 00:00:00
2	102	1002	Europe	1500	200	02-01-2024 00:00:00
3	103	1003	Asia	1800	300	03-01-2024 00:00:00
4	104	1004	North America	1100	150	04-01-2024 00:00:00
5	105	1001	Europe	1300	50	05-01-2024 00:00:00

VALUES

- **Description :** The VALUES function in DAX returns a single-column table that contains the distinct values from a column, considering any filters that might be applied to the current context. It is useful when you want to retrieve all distinct values from a column for further calculations or analysis.
- **Example :**

  DISTINCT_Geographies = VALUES(Sales[Country])

  This would create a table containing all distinct countries from the Sales table, considering any filters currently applied.



```
DISTINCT_Geographies = VALUES(Sales[Region])
```

Region
North America
Europe
Asia

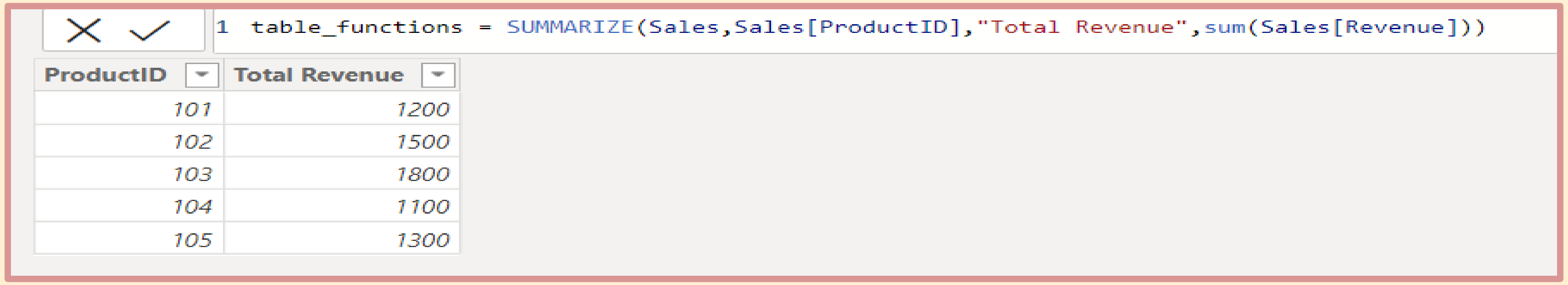
SUMMARIZE

- **Description :** Returns a summary table for the requested totals over a set of groups.

- **Example :**

  SUMMARIZE(Sales, Sales[ProductID], "Total Revenue", SUM(Sales[Revenue]))

  This creates a summary table that groups Sales by ProductID and calculates Total Revenue for each product.



```
table_functions = SUMMARIZE(Sales,Sales[ProductID],"Total Revenue",sum(Sales[Revenue]))
```

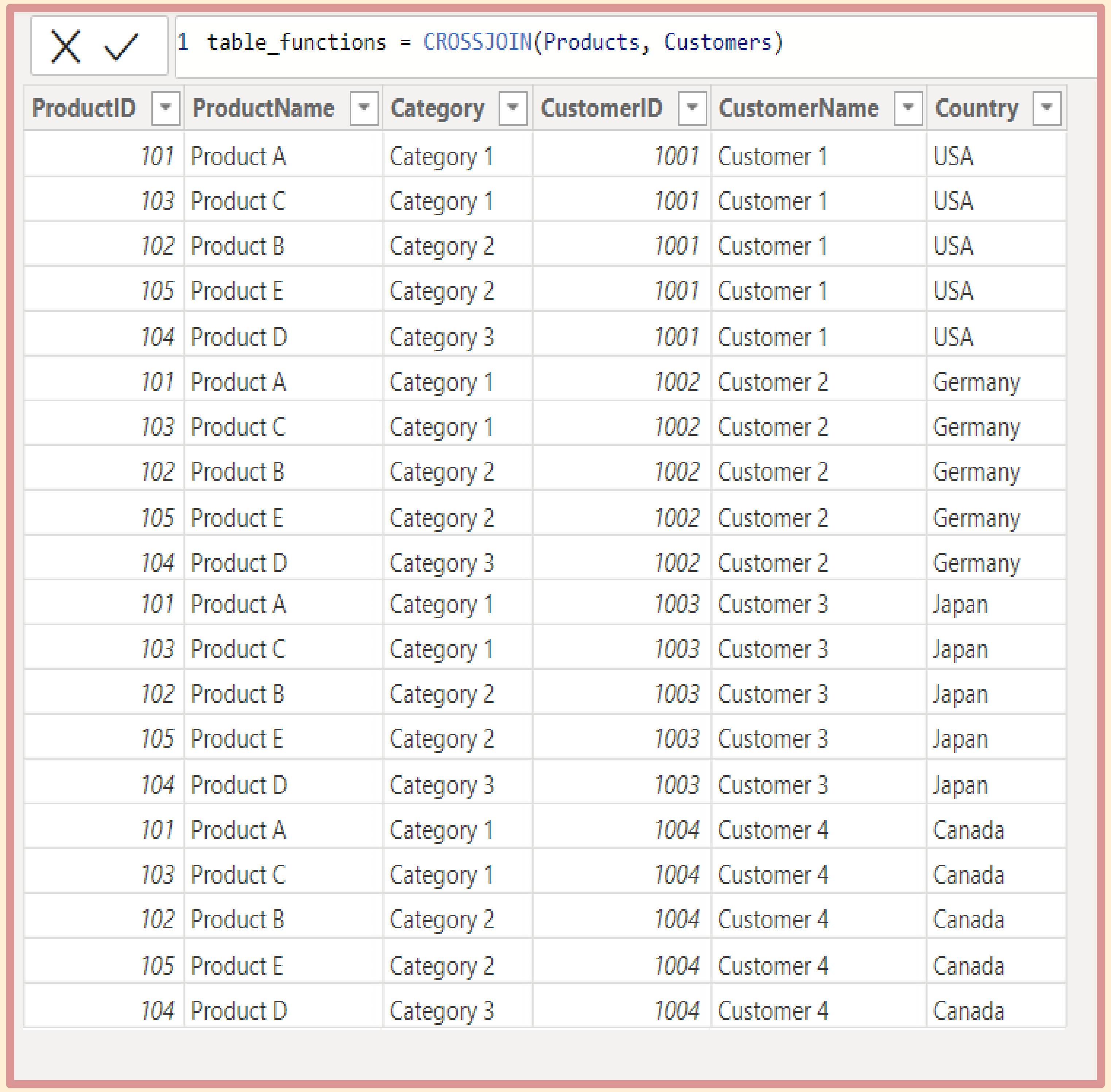
ProductID	Total Revenue
101	1200
102	1500
103	1800
104	1100
105	1300

CROSSJOIN

- **Description :** Returns a table that is a Cartesian product of all the tables specified.
- **Example :**

  CROSSJOIN(Products, Customers)

  This returns a table that is the Cartesian product of the Products and Customers tables.



```
1  table_functions = CROSSJOIN(Products, Customers)
```

ProductID	ProductName	Category	CustomerID	CustomerName	Country
101	Product A	Category 1	1001	Customer 1	USA
103	Product C	Category 1	1001	Customer 1	USA
102	Product B	Category 2	1001	Customer 1	USA
105	Product E	Category 2	1001	Customer 1	USA
104	Product D	Category 3	1001	Customer 1	USA
101	Product A	Category 1	1002	Customer 2	Germany
103	Product C	Category 1	1002	Customer 2	Germany
102	Product B	Category 2	1002	Customer 2	Germany
105	Product E	Category 2	1002	Customer 2	Germany
104	Product D	Category 3	1002	Customer 2	Germany
101	Product A	Category 1	1003	Customer 3	Japan
103	Product C	Category 1	1003	Customer 3	Japan
102	Product B	Category 2	1003	Customer 3	Japan
105	Product E	Category 2	1003	Customer 3	Japan
104	Product D	Category 3	1003	Customer 3	Japan
101	Product A	Category 1	1004	Customer 4	Canada
103	Product C	Category 1	1004	Customer 4	Canada
102	Product B	Category 2	1004	Customer 4	Canada
105	Product E	Category 2	1004	Customer 4	Canada
104	Product D	Category 3	1004	Customer 4	Canada

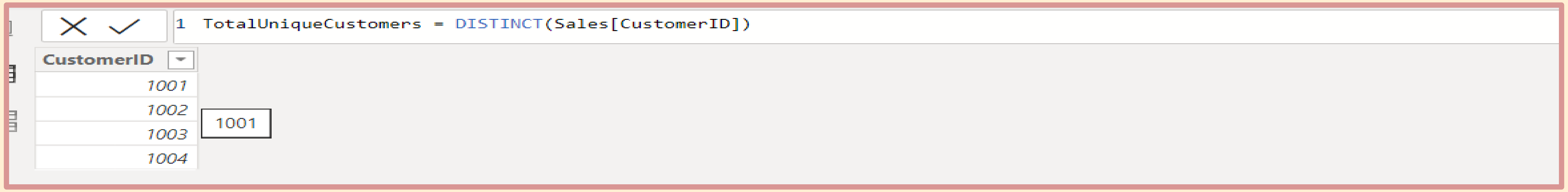
DISTINCT

- **Description :** The DISTINCT function in DAX, on the other hand, is used within aggregation functions to ensure that duplicate values are not counted multiple times.It is typically used in conjunction with aggregation functions like COUNT or SUM to ensure that calculations are based on unique values.

- **Example :**

  TotalUniqueCustomers = DISTINCT(Sales[CustomerID])

  This calculates the count of unique CustomerID values in the Sales table, ensuring that each customer is counted only once, even if they have multiple transactions.

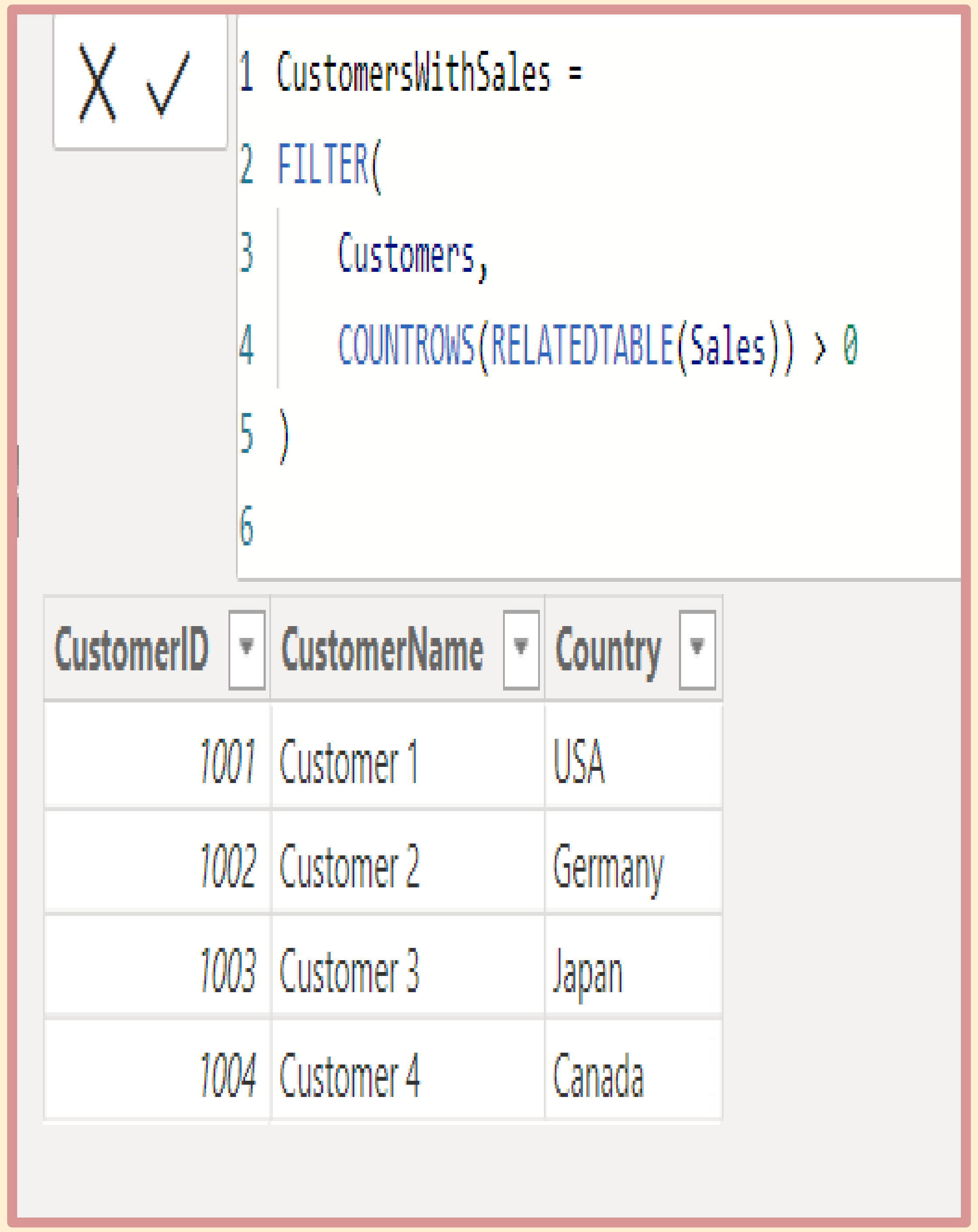


```
TotalUniqueCustomers = DISTINCT(Sales[CustomerID])
```

CustomerID
1001
1002
1003
1004

```
1001
```

RELATEDTABLE



```
CustomersWithSales =
FILTER(
    Customers,
    COUNTROWS(RELATEDTABLE(Sales)) > 0
)
```

CustomerID	CustomerName	Country
1001	Customer 1	USA
1002	Customer 2	Germany
1003	Customer 3	Japan
1004	Customer 4	Canada

- **Description :** The RELATEDTABLE function in DAX is used to retrieve a table related to the current table based on a defined relationship.
- **Example :**

```
CustomersWithSales =
FILTER(
    Customers,
    COUNTROWS(RELATEDTABLE(Sales)) > 0)
```

RELATEDTABLE(Sales) retrieves a table of related sales records for each customer in the Customers table.

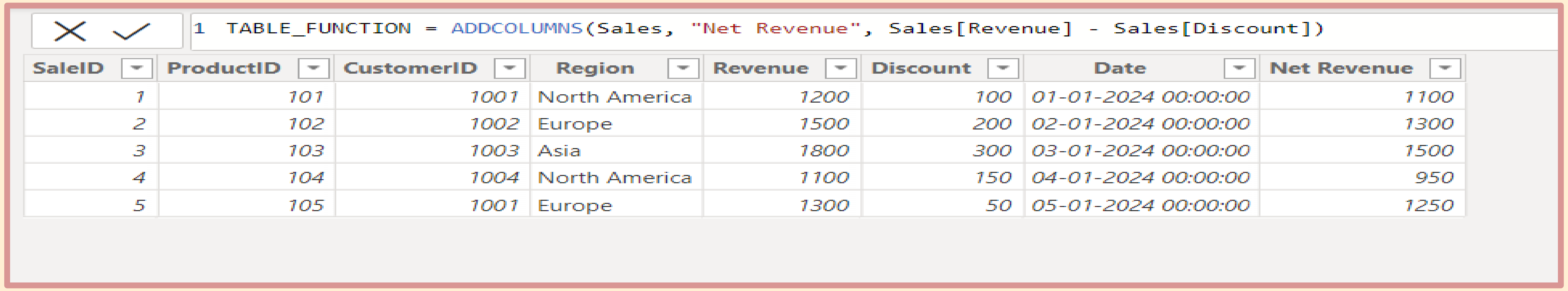
ADDCOLUMNS

- **Description :** Adds calculated columns to the given table or table expression.

- **Example :**

  ADDCOLUMNS(Sales, "Net Revenue", Sales[Revenue] - Sales[Discount])

  This adds a new column "Net Revenue" to the Sales table.



```
1  TABLE_FUNCTION = ADDCOLUMNS(Sales, "Net Revenue", Sales[Revenue] - Sales[Discount])
```

SaleID	ProductID	CustomerID	Region	Revenue	Discount	Date	Net Revenue
1	101	1001	North America	1200	100	01-01-2024 00:00:00	1100
2	102	1002	Europe	1500	200	02-01-2024 00:00:00	1300
3	103	1003	Asia	1800	300	03-01-2024 00:00:00	1500
4	104	1004	North America	1100	150	04-01-2024 00:00:00	950
5	105	1001	Europe	1300	50	05-01-2024 00:00:00	1250

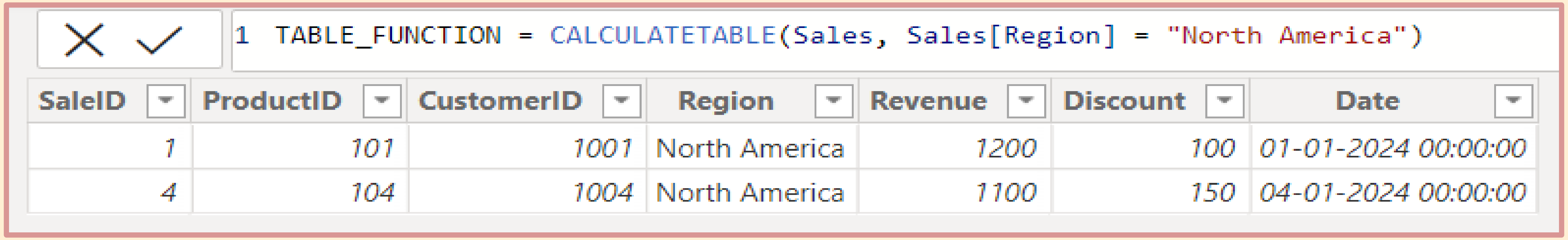
CALCULATETABLE

- **Description :** Evaluates a table expression in a modified filter context.

- **Example :**

  CALCULATETABLE(Sales, Sales[Region] = "North America")

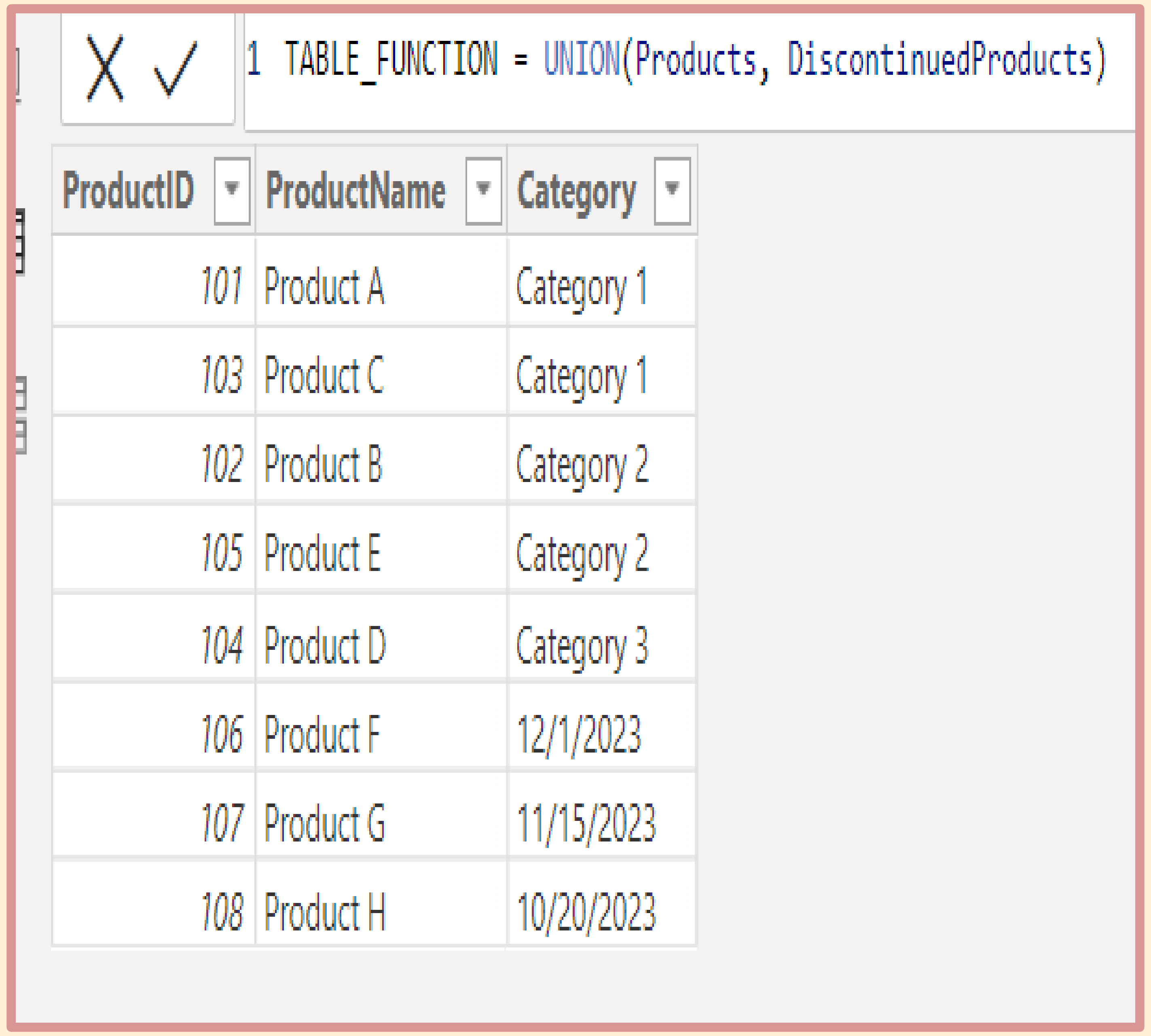
  This returns a table with rows from the Sales table where the Region is "North America".



```
1  TABLE_FUNCTION = CALCULATETABLE(Sales, Sales[Region] = "North America")
```

SaleID	ProductID	CustomerID	Region	Revenue	Discount	Date
1	101	1001	North America	1200	100	01-01-2024 00:00:00
4	104	1004	North America	1100	150	04-01-2024 00:00:00

UNION



```
TABLE_FUNCTION = UNION(Products, DiscontinuedProducts)
```

ProductID	ProductName	Category
101	Product A	Category 1
103	Product C	Category 1
102	Product B	Category 2
105	Product E	Category 2
104	Product D	Category 3
106	Product F	12/1/2023
107	Product G	11/15/2023
108	Product H	10/20/2023

- **Description :** Returns a table that is the union of all the tables specified.

- **Example :**

  UNION(Products, DiscontinuedProducts)

  This returns a table that is the union of the Products and DiscontinuedProducts tables.

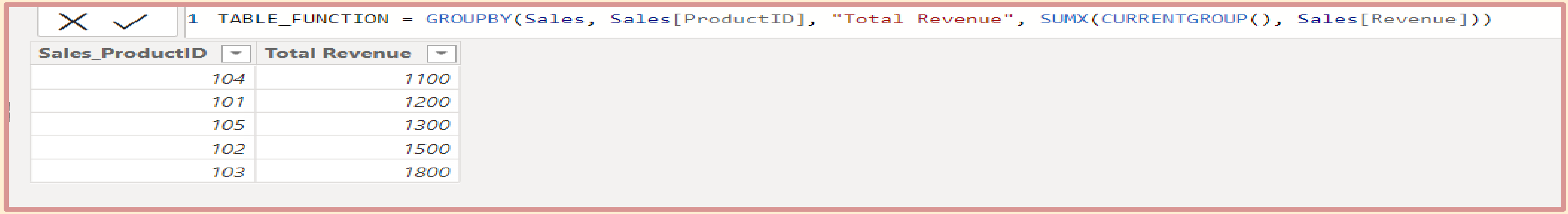
GROUPBY

- **Description :** Returns a table with the selected columns grouped by specified columns.

- **Example :**

  GROUPBY(Sales, Sales[ProductID], "Total Revenue", SUMX(CURRENTGROUP(), Sales[Revenue]))

  This creates a summary table that groups Sales by ProductID and calculates Total Revenue for each product.



```
TABLE_FUNCTION = GROUPBY(Sales, Sales[ProductID], "Total Revenue", SUMX(CURRENTGROUP(), Sales[Revenue]))
```

Sales_ProductID	Total Revenue
104	1100
101	1200
105	1300
102	1500
103	1800

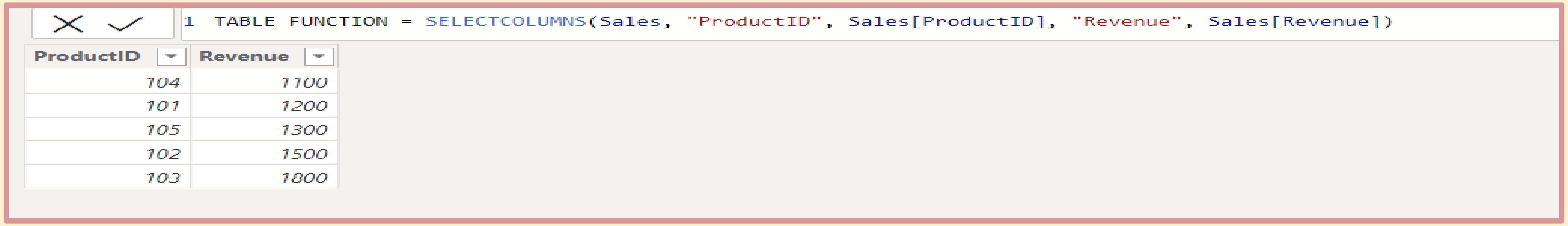
SELECTCOLUMNS

- **Description :** Returns a table with selected columns from the given table and with new names specified by the DAX expressions.

- **Example :**

  SELECTCOLUMNS(Sales, "ProductID", Sales[ProductID], "Revenue", Sales[Revenue])

  This returns a table with only the ProductID and Revenue columns from the Sales table.



```
TABLE_FUNCTION = SELECTCOLUMNS(Sales, "ProductID", Sales[ProductID], "Revenue", Sales[Revenue])
```

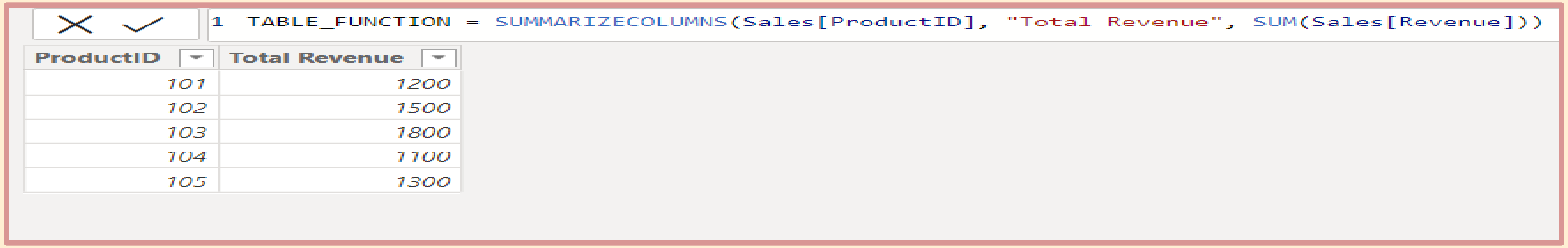
ProductID	Revenue
104	1100
101	1200
105	1300
102	1500
103	1800

SUMMARIZECOLUMNS

- **Description :** Returns a summary table over a set of groups.
- **Example :**

  SUMMARIZECOLUMNS(Sales[ProductID], "Total Revenue", SUM(Sales[Revenue]))

This creates a summary table that groups Sales by ProductID and calculates Total Revenue for each product.



```
1  TABLE_FUNCTION = SUMMARIZECOLUMNS(Sales[ProductID], "Total Revenue", SUM(Sales[Revenue]))
```

ProductID	Total Revenue
101	1200
102	1500
103	1800
104	1100
105	1300