MS - Excel

1). What is power Query editor?

⊙ power Query editor is a tool which is used for (ETL) → process

E - Extract

T - Transform

L - Load

2). How to (Load the data) directly into (data model) And connections without Loading it into the spread sheets?

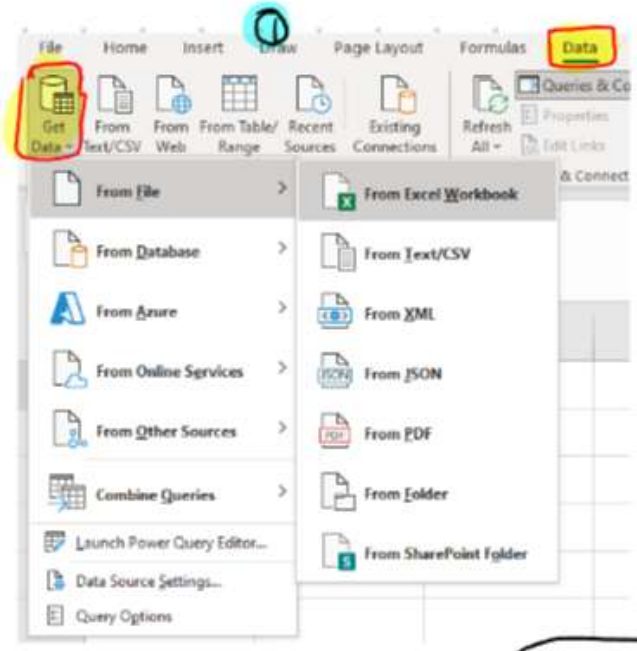
→ **Data model**: Data model is the place our data is stored in A compressed form, so that we can use (Data model) to store Large Amount of data.

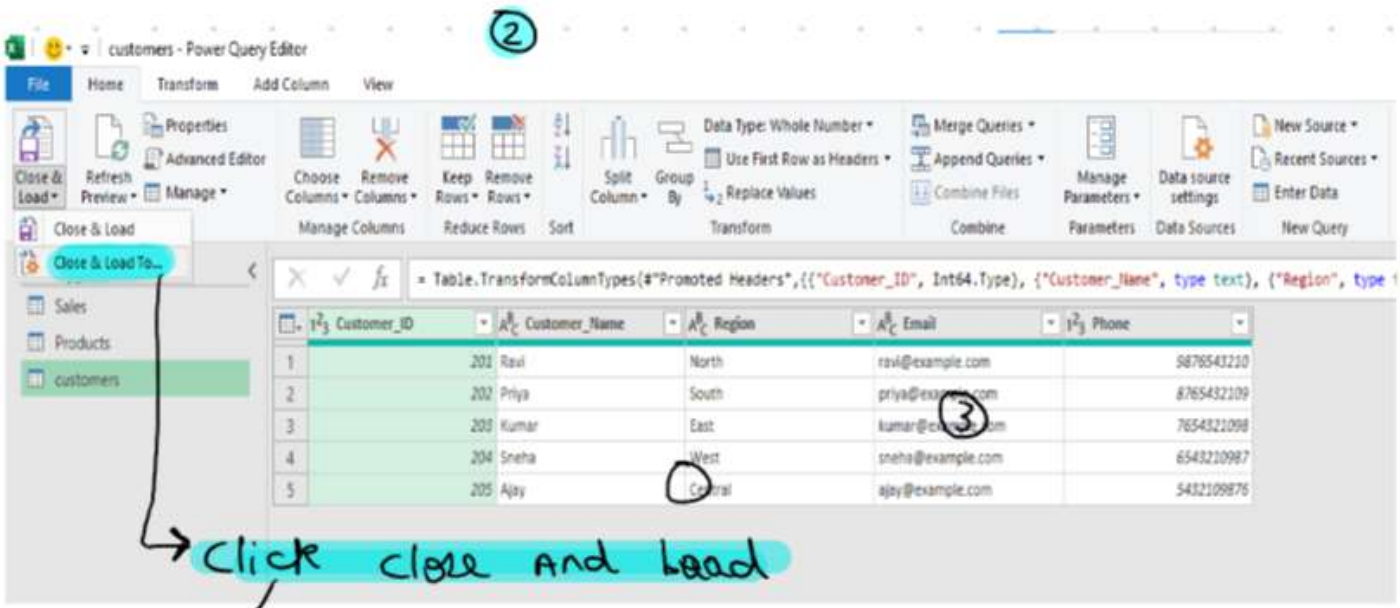
→ **views in Datamodel**:

(i) **Data view**: used to create calculated columns using DAX

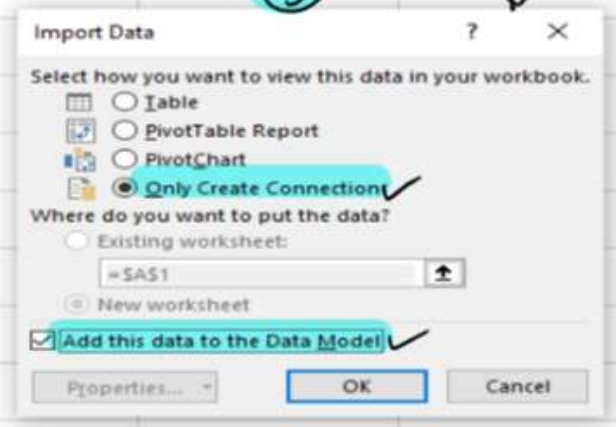
(ii) **Diagram view**: used to establish relationships between tables similar to powerbi

Note: once we load our data to data model & connections the data will be only Available in the (power Query editor) & the Data model.





Click close And Load



→ Now your Data will be added to power Query editor And data model

3) DAX functions in Excel?

⊙ DAX - stands for Data Analytics expression, generally used to create calculated columns in (Power Query editor) or Data model

⊙ The formula language used in DAX is (M - language) it contains various built-in functions Like if(), sum() etc..

→ Creating calculated column in Data model using if() function?

→ click to go to the Data model

① if() → function:

[new... | = IF('01 Churn-Dataset'[Churn] = "Yes",1,0)

(IF(LogicalTest, ResultIfTrue, [ResultIfFalse])

	TotalCharges	numAdminTickets	numTechTickets	Churn	new_churn
1	903.6	0	0	Yes	1
2	893.55	0	0	Yes	1
3	259.8	0	0	Yes	1
4	676.35	0	0	Yes	1
5	5084.65	0	0	Yes	1

Syntax:

if (logical_test, Result when true, Result when false)

⊙ && → and operator to check multiple conditions:

[1-ye... ✕ ✓ fx = IF([Contract] = "One Year" && [new_churn] = 1, 1,0)

IF(LogicalTest, ResultIfTrue, [ResultIfFalse])

	TotalCharges	numAdminTickets	numTechTickets	Churn	new_churn	1-year-churn	female-or-parterner
1	903.6	0	0	Yes	1	1	0
2	893.55	0	0	Yes	1	1	1
3	259.8	0	0	Yes	1	1	1
4	676.35	0	0	Yes	1	1	0

Syntax: if (condition 1 && condition 2, result_true, result_false)

↳ Here im creating a new column - (1-year churn)

↳ The column contains value - 1 for All the customers of (one year contract) and (churn = 1) And 0 for else condition

vice versa

→ || or operator to check multiple conditions:

[fema... ▾ | fx | = IF([gender] = "Female" || [Partner] = "Yes", 1,0)

	TotalCharges	numAdminTickets	numTechTickets	Churn	new_churn	1-year-churn	female-or-parterner
1	903.6	0	0	Yes	1	1	0
2	893.55	0	0	Yes	1	1	1
3	259.8	0	0	Yes	1	1	1
4	676.35	0	0	Yes	1	1	0

→ creating calculated column in Data model using Switch() function?

```
tenure_category := SWITCH(TRUE(),
                    [tenure] <= 12, "low tenure customer",
                    [tenure] <=33, "medium tenure customer",
                    [tenure] <=72, "high tenure customer", "no data")
```

→ formula

SWITCH(Expression, Value1, Result1, ..., [Value4], [Result4], ..., [Else])
SWITCH(Expression, Value1, Result1, ..., [Value3], [Result3], [Else])

ckets	numTechTickets	Churn	new_churn	1-year-churn	female-or-parterner	tenure_category
0	0	Yes	1	1	0	medium tenure customer
0	0	Yes	1	1	1	medium tenure customer
0	0	Yes	1	1	1	low tenure customer
0	0	Yes	1	1	0	medium tenure customer

→ here in creating new column called (tenure_category) using Switch() → function

The `SWITCH` function in DAX (Data Analysis Expressions) is a versatile conditional function that allows you to evaluate an expression or condition against multiple cases and return specific results. Let's break it down with the syntax and the given formula:

Explanation of Formula

1. `TRUE()` **as the expression**: Makes SWITCH work like an `IF-ELSE` block, evaluating conditions sequentially.
2. **Conditions**:
   - `[tenure] <= 12` : If the `tenure` is 12 months or less, it returns "low tenure customer".
   - `[tenure] <= 33` : If the `tenure` is greater than 12 but less than or equal to 33, it returns "medium tenure customer".
   - `[tenure] <= 72` : If the `tenure` is greater than 33 but less than or equal to 72, it returns "high tenure customer".
3. **Fallback**:
   - `"no data"` : If none of the above conditions are true (e.g., `tenure > 72` or `tenure` is blank), it returns this default value.

→ Calculate() function in Excel ?

⇒ Note :- Calculate() function is generally used to create Measures

The `CALCULATE` function in Excel is a **DAX (Data Analysis Expressions)** function used in Power BI, not in standard Excel formulas. It modifies the context of calculations by applying filters to data. If you are using Power BI for your analysis, `CALCULATE` becomes highly useful to create metrics and conditional aggregations.

Let me explain with an example using your **churn dataset**.

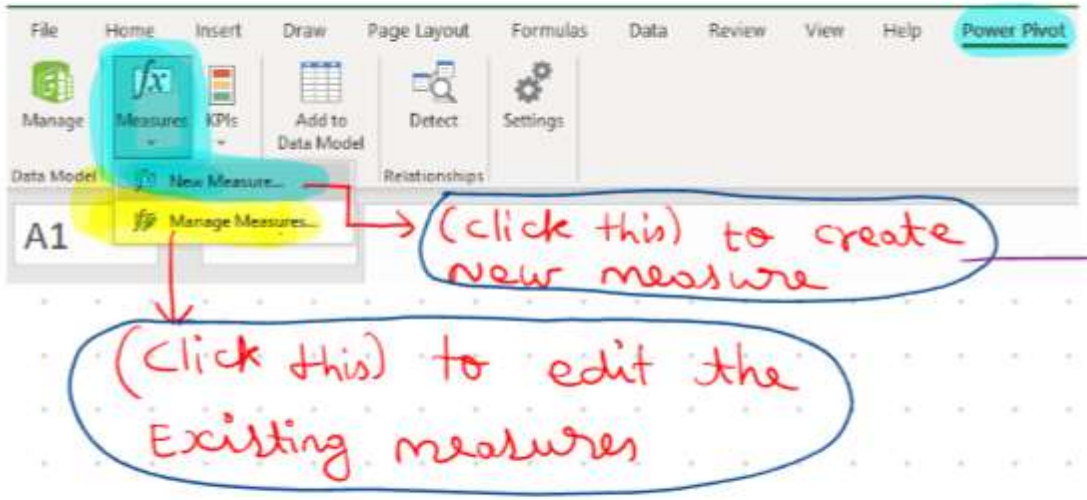
Syntax

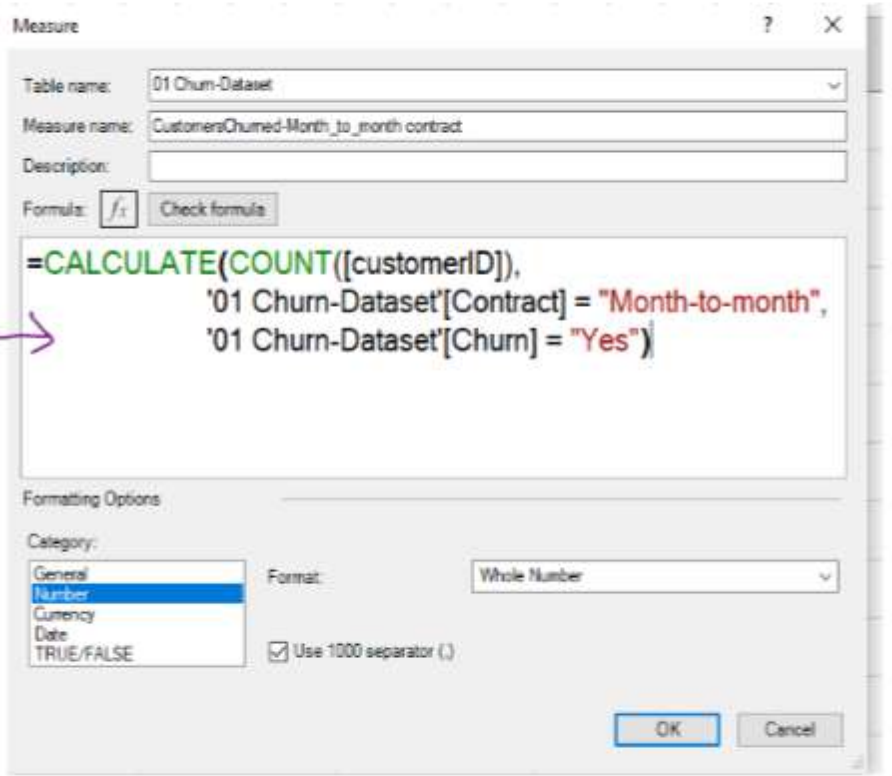
```
DAX

CALCULATE(<expression>, <filter1>, <filter2>, ...)
```

- `<expression>` : The calculation to be performed (e.g., SUM, AVERAGE).
- `<filter>` : A condition or filter to modify the data context.

→ Let's see how to create a measure using calculate() function in Data model ?





This DAX formula calculates the **count of customers** (`customerID`) who meet the following criteria:

- Their **contract type** is "**Month-to-month**".
- They have **churned** (i.e., `churn = "Yes"`).

expression
filter1
filter2

Key Points

- `CALCULATE` is used to modify the **filter context** of a calculation, allowing you to perform conditional aggregations or apply filters.
- Functions like `DIVIDE` , `MIN` , `MAX` , `SUM` , `AVERAGE` , etc., can be used inside `CALCULATE` to create more powerful and flexible calculations.
- `CALCULATE` can handle **multiple filters** simultaneously, allowing for more specific and detailed analyses.

→ let's see how to create calculated column using Related() → function?

Returns

Returned	Order ID
Yes	ES-2015-RA1994545-42218
Yes	ES-2012-SC208458-41070
Yes	ES-2015-CC1210045-42182
Yes	ES-2015-MM1792045-42199
Yes	ES-2015-BB1154548-42336

Orders

Row ID	Order ID	Order Date
10536	ES-2015-LC168...	17-03-2015 ...
18645	ES-2015-DC128...	17-04-2015 ...
10469	ES-2014-JE157...	15-02-2014 ...
17393	ES-2012-AB101...	28-09-2012 ...
16528	ES-2013-SC200...	27-04-2013 ...
17565	ES-2013-PM18...	15-12-2013 ...
16719	ES-2014-MV18...	06-08-2014 ...
16527	ES-2013-SC200...	27-04-2013 ...
19349	ES-2014-NC183...	12-06-2014 ...
16729	ES-2015-CB120...	25-09-2015 ...
11137	ES-2013-JH154...	06-08-2013 ...
11662	IT-2015-TB210...	26-11-2015 ...
19975	ES-2014-LC171...	15-11-2014 ...
19163	ES-2015-GA14...	13-10-2015 ...

⊙ create a column in orders table using the existing column from the returns table.

⊙ we can use Related() function, since both the tables are in relationship.

Returned Products := IF(RELATED(Returns[Returned])="Yes",1,0)

Sales	Quantity	Discount	Profit	Shipping Cost	Order Priority	Loss	Returned Products
165.96	2	0	26.52	17.2	Medium	0	0
206.76	2	0	28.92	13.99	Medium	0	0
167.34	2	0	55.2	10.24	Medium	0	0
89.58	2	0	30.42	9.6	Medium	0	0
109.8	2	0	19.74	9.46	Medium	0	0
113.64	2	0	17.04	9.42	Medium	0	0
91.38	2	0	14.58	9.16	Medium	0	0
93.42	2	0	14.94	8.82	Medium	0	0
214.14	2	0	87.78	8.05	Medium	0	0
95.22	2	0	18.06	7.97	Medium	0	0
95.64	2	0	12.42	7.77	Medium	0	0
97.86	2	0	8.76	7.58	Medium	0	0
98.4	2	0	42.3	7.11	Medium	0	0
107.82	2	0	24.78	6.63	Medium	0	0

1. `RELATED(Returns[Returned])`:
   - The `RELATED` function retrieves values from a related table.
   - Here, it accesses the column `[Returned]` in the `Returns` table.
   - For this to work, there must be an established relationship between the current table and the `Returns` table (e.g., via a key column like `Product ID`).
2. `IF(... , 1, 0)`:
   - This `IF` statement checks a condition:
     - If the value in `Returns[Returned]` is `"Yes"`, it assigns the value `1`.
     - Otherwise, it assigns the value `0`.
3. **Output:**
   - The calculated column `Returned Products` will show `1` if the product was returned (`Returned` = `"Yes"`) and `0` otherwise.

Returned Products := IF(RELATED(Returns[Returned])="Yes",1,0) → formula

⇒ let's discuss about ALL(), ALLSELECTED(), ALLExpected() DAX functions in Excel?

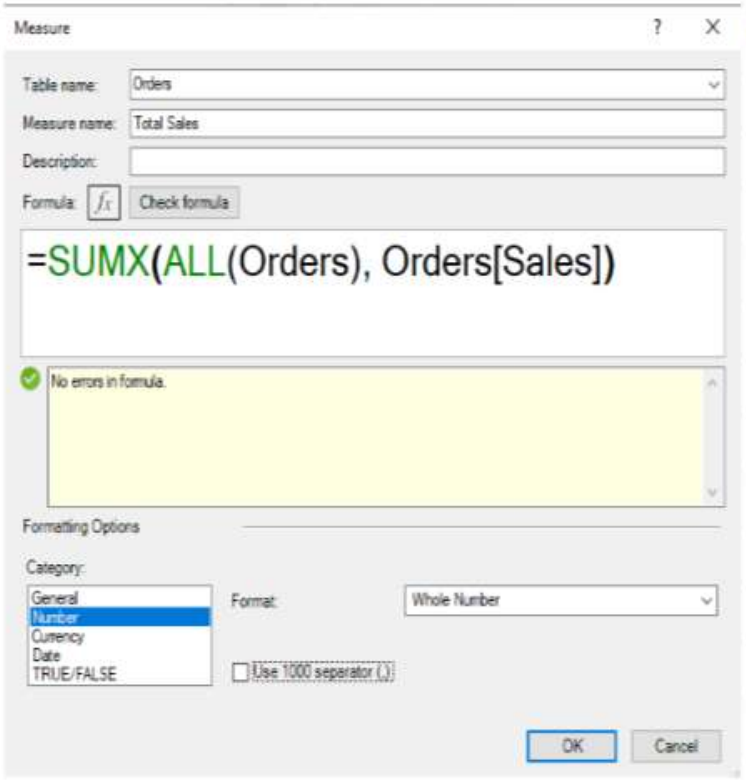
Summary of Differences:

Function	Purpose	Filters Retained
ALL	Removes all filters on the specified table/column.	None
ALLSELECTED	Removes filters except those applied in the current visual or selection.	Filters from current visual or slicer remain.
ALLEXCEPT	Removes all filters except those on specified columns.	Specified columns retain filters.

These formulas are essential in building advanced measures and managing context in DAX calculations.

⇒ ALL() function?

Function	Purpose	Filters Retained
ALL	Removes all filters on the specified table/column.	None



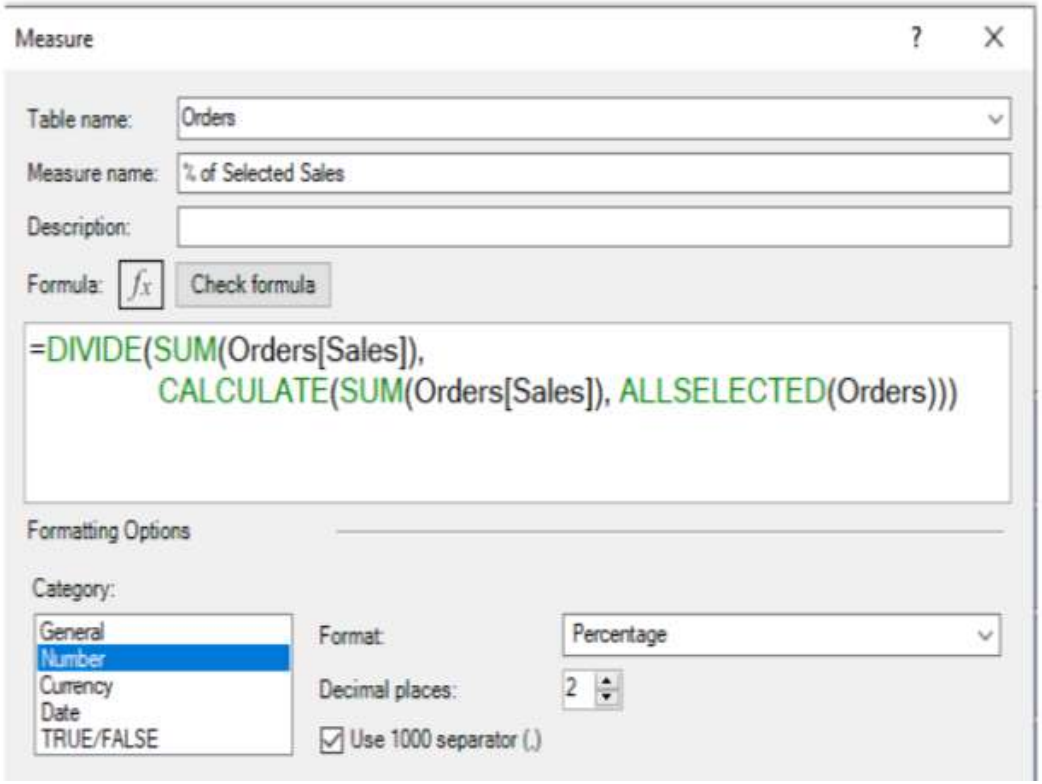
Explanation:

- `ALL(Orders)` : Removes all filters on the `Orders` table.
- `SUMX` : Iterates over each row of the unfiltered `Orders` table and sums the values in the `Sales` column.

Purpose: Calculates the **total sales** across the entire dataset, ignoring any filters applied in the report or visual.

→ ALLSELECTED() function :

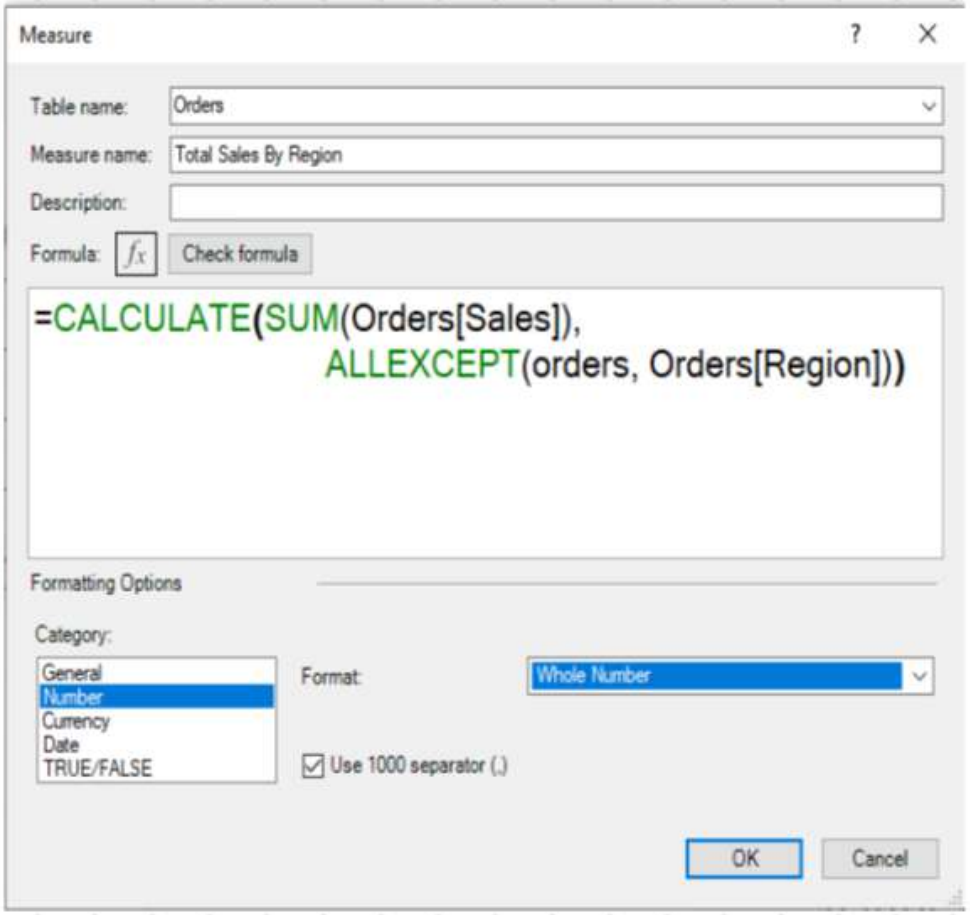
Function	Purpose	Filters Retained
ALLSELECTED	Removes filters except those applied in the current visual or selection.	Filters from current visual or slicer remain.



Explanation:

- `SUM(Orders[Sales])` : Calculates the sum of sales for the current filter context.
- `ALLSELECTED(Orders)` : Keeps the filters applied by the user through slicers or visuals.
- `DIVIDE(...)` : Divides the current sales by the total sales for the selected data.

→ ALLEXPECTED() function:



Explanation:

- `SUM(Orders[Sales])` : Sums the sales for the table.
- `ALLEXCEPT(Orders, Orders[Region])` : Keeps the filter on the `Region` column and removes all other filters.

ALLEXCEPT Formula Example

This formula calculates **total sales by region**, ignoring all other filters except `Region`:

⇒ Real time use of ALL, ALLSELected, ALLExpected ?

Row Labels	Total Sales	Total Sales By Region	% of Selected Sales
Africa	12642502	17,77,251	8.08%
Asia Pacific	12642502	17,77,251	1.09%
LATAM	12642502	17,77,251	87.07%
USCA	12642502	17,77,251	3.77%
Grand Total	**12642502**	**17,77,251**	**100.00%**

```
=SUMX(ALL(Orders), Orders[Sales])
```

```
=DIVIDE(SUM(Orders[Sales]),
        CALCULATE(SUM(Orders[Sales]), ALLSELECTED(Orders)))
```

```
=CALCULATE(SUM(Orders[Sales]),
           ALLEXCEPT(orders, Orders[Region]))
```