Unit 14
Business Intelligence

DAX LECTURE 3

Calculated Columns VS. Measures

Calculated Columns

Static: Calculated once during data refresh and stored in the model, not changing until the next refresh.

Row-level: Each value is computed at the row level, often using data within the same row or related data.

Storage: Consumes memory in the data model because results are stored persistently.

Use: Suitable for creating new data elements that can be used as part of filters, in relationships, or as fixed values in your reports.

Measures

Dynamic: Calculated in real-time based on the current context, such as filters and slicers applied in reports.

Aggregate: Typically used to perform calculations that summarize data, such as sums, averages, or counts.

No Storage: Results are not stored but computed on-the-fly, so they don't consume memory in the data model.

Use: Ideal for metrics and KPIs that need to update dynamically as users interact with the report.

COUNTX Function

This function counts the number of non-blank results when evaluating an expression for each row in a table. If you want to count how many transactions have a unit price greater than $500, you could use:

```
High Price Transactions = COUNTX('Transaction', IF('Transaction'[UnitPrice]> 500, 1,
BLANK()))
```

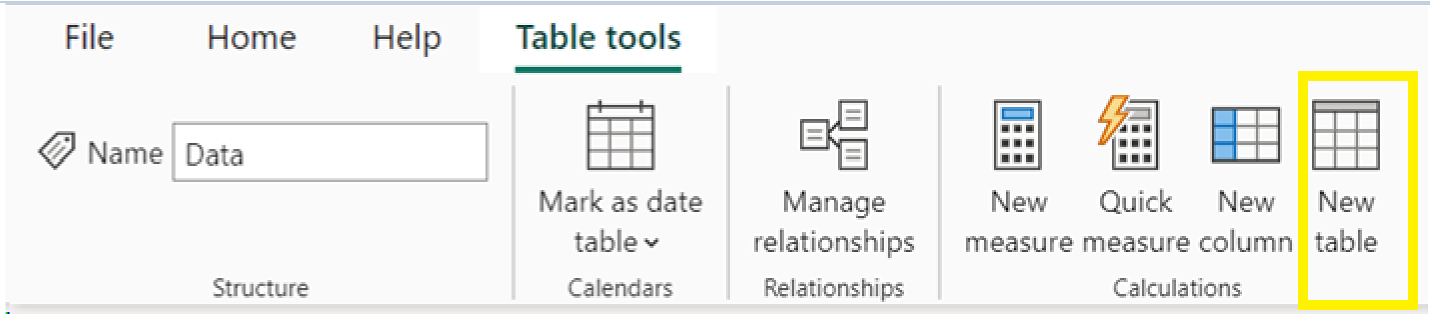
FILTER Function

- FILTER returns a table that represents a subset of another table, based solely on a defined expression that returns a Boolean value (TRUE or FALSE).

Basic Structure

FILTER(Table, Expression)

- FILTER goes row by row through the table, checks if the row meets the specified condition, and if it does, includes that row in the result.



FILTER Function

- **Transactions Over $500**: Create a new table of transactions where the total amount (Quantity * UnitPrice) exceeds $500.

```
Transactions Over 500 = FILTER('Transaction', 'Transaction'[Quantity] *
'Transaction'[UnitPrice] > 500)
```

- **Laptop Sales**: Filter the transactions to only include sales of Laptops.

```
Laptop Transactions = FILTER('Transaction', 'Transaction'[ProductName] = "Laptop")
```

Exercise

- **Purchases by Customers Under 25**: Create a table of transactions made by customers younger than 25 years old.
- **Sales with Above Average Tax Rate**: Assuming you want to filter the transactions to only those with a TaxRate above the average for the dataset.

SELECTCOLUMNS Function

- SELECTCOLUMNS allows you to transform a table into a new table by specifying which columns to include. You can either keep the original column names or provide new names.

Basic Structure

SELECTCOLUMNS(Table, "NewColumnName1", Expression1, "NewColumnName2", Expression2, ...)

- SELECTCOLUMNS essentially creates a new, smaller table from an existing table by specifying which columns to include. It's a way to streamline your data or create a table that's easier to work with based on your specific needs.

SELECTCOLUMNS Function

- Create a new table with just the customer first names and the products they purchased.

```
 CustomerProducts = SELECTCOLUMNS('Transaction', "Customer", 'Transaction'[First Name],
"Product", 'Transaction'[ProductName])
```

- SELECTCOLUMNS doesn't inherently filter, it's often used after filtering functions. Here’s a combined example with FILTER, selecting and renaming columns for transactions over $500:

```
 HighValueSalesInfo = SELECTCOLUMNS(FILTER('Transaction', 'Transaction'[Quantity] *
'Transaction'[UnitPrice] > 500), "TransactionID", 'Transaction'[TransactionID], "Amount",
'Transaction'[Quantity] * 'Transaction'[UnitPrice])
```

Exercise

1. Load EmployeeData.xlsx into Power BI.
2. Write a DAX expression to create a new table with only Name and Salary columns.
3. Create another table that includes only the employees from the HR department using the Filter function.