CALENDARAUTO()

TotalSalesAll = CALCULATE(SUM(Sales[Price ($)]), ALL(Sales))

TotalSalesByCompany = CALCULATE(SUM(car\_data[Price ($)]), ALLEXCEPT(car\_data, car\_data[Company])) measure

SelectedRegionSales = CALCULATE(SUM(car\_data[Price ($)]), ALLSELECTED(car\_data[Dealer\_Region])) measure

HighValueSalesTabless = CALCULATETABLE(car\_data, car\_data[Price ($)] > 20000) return table

RankByPrice =

RANKX(

    ALL(car\_data),                     // Removes filters on the car\_data table

    CALCULATE(SUM(car\_data[Price ($)])), // Aggregates Price ($) for ranking

    ,                                   // No value for ties (default behavior)

    DESC,                               // Rank in descending order

    DENSE                               // Use dense ranking for ties

)

### ****MEASURES****

SelectedCompany = SELECTEDVALUE(car\_data[Company], "No Company Selected")

****Create columns****

SalesSummary = SUMMARIZE(car\_data, car\_data[Company], car\_data[Model], "Total Sales", SUM(car\_data[Price ($)]))

****Create table****

Top3ExpensiveSales = TOPN(3, car\_data, car\_data[Price ($)], DESC)

Create table

### ****### \*\*1. ALL (Ignore filters)\*\*****

### ****\*\*Business Question\*\*: What is the total sales value for all car sales, regardless of any filters?****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****TotalSalesAll = CALCULATE(SUM(Sales[Price ($)]), ALL(Sales))****

### ****```****

### ****- This formula will calculate the total sales for all cars in the dataset, ignoring any applied filters (such as car company, model, or dealer).****

### ****---****

### ****### \*\*2. ALLEXCEPT (Keep specific filters)\*\*****

### ****\*\*Business Question\*\*: What is the total sales value for each company, while ignoring any other filters?****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****TotalSalesByCompany = CALCULATE(SUM(Sales[Price ($)]), ALLEXCEPT(Sales, Sales[Company]))****

### ****```****

### ****- This formula calculates the total sales for each car company while ignoring other filters except for the `Company` column.****

### ****---****

### ****### \*\*3. ALLSELECTED (Respect user selections but ignore external filters)\*\*****

### ****\*\*Business Question\*\*: What is the sales total based on user-selected regions, ignoring external filters?****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****SelectedRegionSales = CALCULATE(SUM(Sales[Price ($)]), ALLSELECTED(Sales[Dealer\_Region]))****

### ****```****

### ****- This formula calculates the total sales for selected regions as per the user's filter but ignores filters applied outside the selection context.****

### ****---****

### ****### \*\*4. FILTER\*\*****

### ****\*\*Business Question\*\*: How many cars were sold where the price was greater than $20,000?****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****HighValueCars = COUNTROWS(FILTER(Sales, Sales[Price ($)] > 20000))****

### ****```****

### ****- This formula counts the number of cars sold with a price greater than $20,000.****

### ****---****

### ****### \*\*5. CALCULATETABLE\*\*****

### ****\*\*Business Question\*\*: Create a filtered table of sales where the price is greater than $20,000.****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****HighValueSalesTable = CALCULATETABLE(Sales, Sales[Price ($)] > 20000)****

### ****```****

### ****- This formula creates a table that only includes car sales where the price is greater than $20,000.****

### ****---****

### ****### \*\*6. RANKX\*\*****

### ****\*\*Business Question\*\*: Rank all car sales by price, from the most expensive to the least.****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****RankByPrice = RANKX(ALL(Sales), Sales[Price ($)], , DESC)****

### ****```****

### ****- This formula ranks all car sales by price in descending order.****

### ****---****

### ****### \*\*7. SELECTEDVALUE\*\*****

### ****\*\*Business Question\*\*: Return the selected company from a slicer.****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****SelectedCompany = SELECTEDVALUE(Sales[Company], "No Company Selected")****

### ****```****

### ****- This formula returns the company selected from a slicer or "No Company Selected" if none is chosen.****

### ****---****

### ****### \*\*8. SUMMARIZE\*\*****

### ****\*\*Business Question\*\*: Summarize total sales by company and model.****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****SalesSummary = SUMMARIZE(Sales, Sales[Company], Sales[Model], "Total Sales", SUM(Sales[Price ($)]))****

### ****```****

### ****- This formula summarizes the total sales by both the car company and model.****

### ****---****

### ****### \*\*9. TOPN\*\*****

### ****\*\*Business Question\*\*: What are the top 3 highest-priced car sales?****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****Top3ExpensiveSales = TOPN(3, Sales, Sales[Price ($)], DESC)****

### ****```****

### ****- This formula returns the top 3 most expensive car sales.****

### ****---****

### ****### \*\*Additional Examples:\*\*****

### ****#### \*\*Deriving Year, Quarter, and Month:\*\*****

### ****\*\*Business Question\*\*: How can we extract year, quarter, and month from the sale date?****

### ****#### \*\*Formula\*\*:****

### ****```DAX****

### ****Year = YEAR(Sales[Date])****

### ****Quarter = QUARTER(Sales[Date])****

### ****Month = MONTH(Sales[Date])****

### ****```****

### ****- These formulas derive the year, quarter, and month from the sales date for use in time-based analysis.****

### ****---****

### ****These formulas help analyze various aspects of car sales, such as total sales, ranking, filtering data by price, or summarizing sales by company. Data Analysis Expressions (DAX) with Syntax and Examples****

DAX is a powerful language used in Power BI to create custom calculations, columns, and measures. Below are some key concepts along with syntax and examples:

### ****Difference Between M Query and DAX Query****

**M Query** is used for data transformation before loading data into Power BI, while **DAX** is used for calculations once data is loaded into the model.

**M Query Example** (used in Power Query for data transformation):

let  
  Source = Excel.Workbook(File.Contents("file.xlsx"), null, true),  
  Sheet1 = Source{[Name="Sheet1"]}[Data],  
  FilteredData = Table.SelectRows(Sheet1, each [Sales] > 1000)  
in  
  FilteredData

**DAX Query Example** (used in Power BI for calculations):

TotalSales = SUM(Sales[Amount])

### ****DAX Data Types and DAX Operators****

#### ****DAX Data Types****:

* **Whole Number**: 1, 2, 100
* **Decimal Number**: 10.5, 3.14
* **Currency**: Fixed decimal number like 99.99
* **Date/Time**: 2024-09-25, 12/31/2024
* **Text**: "Product A", "Category B"
* **Boolean**: TRUE, FALSE

#### ****DAX Operators****:

**Arithmetic Operators**:

* + +: Addition ([Sales] + [Cost])
  + -: Subtraction ([Revenue] - [Expense])
  + \*: Multiplication ([Price] \* [Quantity])
  + /: Division ([TotalSales] / [Orders])

**Comparison Operators**:

* + =: Equals
  + <>: Not equals
  + >: Greater than
  + <: Less than

Example:

IsLargeSale = IF(Sales[Amount] > 1000, TRUE, FALSE)

### ****DAX Measures and Calculations: Row Context and Filter Context****

* **Row Context**: Refers to the row-by-row calculation.
* **Filter Context**: Refers to the filters applied to the data.

Example of a **DAX Measure**:

TotalRevenue = SUM(Sales[Revenue])

In the above example, **SUM()** operates in the **filter context** to aggregate data based on the current filter selection in Power BI.

### ****Aggregation Functions****

**SUM()**: Adds all numbers in a column.

  TotalSales = SUM(Sales[Amount])

**SUMX()**: Iterates over a table, performing row-by-row calculations.

  TotalProfit = SUMX(Sales, Sales[Quantity] \* Sales[UnitPrice])

**COUNT()**: Counts non-blank values in a column.

  OrderCount = COUNT(Orders[OrderID])

**MIN()**: Finds the minimum value in a column.

  MinSales = MIN(Sales[Amount])

**MAX()**: Finds the maximum value in a column.

  MaxSales = MAX(Sales[Amount])

**AVERAGE()**: Calculates the mean of a column.

  AvgSales = AVERAGE(Sales[Amount])

### ****Text Functions****

**LEFT()**: Extracts a substring from the left.

  FirstThreeChars = LEFT(Sales[ProductName], 3)

**CONCATENATE()**: Combines two strings.

  FullName = CONCATENATE(Employee[FirstName], Employee[LastName])

**LEN()**: Returns the length of a string.

  ProductNameLength = LEN(Sales[ProductName])

**SEARCH()**: Finds the position of a substring.

  PositionOfX = SEARCH("X", Sales[ProductCode])

### ****Date and Time Functions****

**CALENDAR()**: Creates a date table from a range.

  DateTable = CALENDAR(DATE(2020, 1, 1), DATE(2024, 12, 31))

**CALENDAR AUTO()**: Automatically generates a date table based on the data model.

  AutoDateTable = CALENDARAUTO()

**YEAR()**, **MONTH()**, **DAY()**: Extracts parts of a date.

  YearValue = YEAR(Sales[Date])  
  MonthValue = MONTH(Sales[Date])

### ****Filter Functions****

**ALL()**: Removes all filters on a column or table.

  TotalSalesAllRegions = CALCULATE(SUM(Sales[Amount]), ALL(Sales[Region]))

**ALLEXCEPT()**: Removes all filters except the one specified.

  TotalSalesExceptRegion = CALCULATE(SUM(Sales[Amount]), ALLEXCEPT(Sales, Sales[Region]))

**ALLSELECTED()**: Removes all filters except those from slicers or visuals.

  SalesAllSelected = CALCULATE(SUM(Sales[Amount]), ALLSELECTED(Sales))

**FILTER()**: Returns a table with rows that match a condition.

  HighSales = FILTER(Sales, Sales[Amount] > 1000)

**RANKX()**: Ranks items based on a specific criteria.

  SalesRank = RANKX(ALL(Sales), Sales[Amount])

**TOPN()**: Returns the top N rows from a table based on a ranking.

  Top5Sales = TOPN(5, Sales, Sales[Amount])

### ****Most Versatile DAX Function: CALCULATE****

**CALCULATE()** modifies the **filter context** for calculations. It is one of the most powerful functions in DAX.

Example:

TotalSalesNorth = CALCULATE(SUM(Sales[Amount]), Sales[Region] = "North")

The **CALCULATE()** function can add filters or modify the current filters applied to a calculation.

### ****IF and SWITCH in DAX****

**IF()**: Evaluates a condition and returns different results based on the result.

  HighOrLowSales = IF(Sales[Amount] > 1000, "High", "Low")

**SWITCH()**: Evaluates an expression against a list of possible values.

  SalesCategory = SWITCH(TRUE(),   
                         Sales[Amount] > 5000, "High",   
                         Sales[Amount] > 1000, "Medium",   
                         "Low")

### ****Assignment: DAX in Marketing Analysis****

#### Objective:

Analyze marketing spend and sales data using DAX functions.

#### Steps:

**Load Data**: Import data into Power BI with columns such as **Region**, **Ad Spend**, and **Sales**.

**Create Calculated Columns**

Derive **Year**, **Month**, and **Day** from the date column using **YEAR()**, **MONTH()**, **DAY()**.

Classify sales as "High" or "Low" based on thresholds using **IF()**.

Example:

SalesCategory = IF(Sales[Amount] > 1000, "High", "Low")

**Create Measures**:

* + **Total Sales**:

  TotalSales = SUM(Sales[Amount])

* + **Average Sales**:

  AvgSales = AVERAGE(Sales[Amount])

**Ranking**:

* + Rank regions by sales using **RANKX()**.

RankRegions = RANKX(ALL(Sales[Region]), SUM(Sales[Amount]))

**Filter Using CALCULATE()**:

* + Calculate total sales for the **North** region using **CALCULATE()**.

TotalSalesNorth = CALCULATE(SUM(Sales[Amount]), Sales[Region] = "North")