**DAX**

**Q1. WHAT IS DAX ?**

DATA ANALYSIS EXPRESSION LANGUAGE. APPLICABLE FOR EXCEL, MSBI-SSAS, ADF, POWER BI.

PURPOSE : USED TO CREATE NEW CALCULATED FIELDS FOR DYNAMIC REPORTING.

USED TO DESIGN ANALYTICAL REPORTS

USED TO IMPLEMENT "**ROW LEVEL SECURITY**" = **RLS**

**Q2. REPORT DESIGN OPERATIONS WE CAN PERFORM USING DAX IN POWER BI?**

1. USING DAX, WE CAN DEFINE NEW **COLUMNS** & **MEASURES**

**Columns: These are additional calculated fields, Visible in Data Button.**

**Refers to Pre-generated, static computations.**

**Scope: Local to a table**

Example:

TOTAL\_SALES = 'SALES DATA'[Sale 2016] + 'SALES DATA'[Sale 2017] + 'SALES DATA'[Sale 2018]

**Measure: These are additional calculated fields, Not Visible in Data Button.**

**Refers to Auto-generated, dynamic computations. Aggregations.**

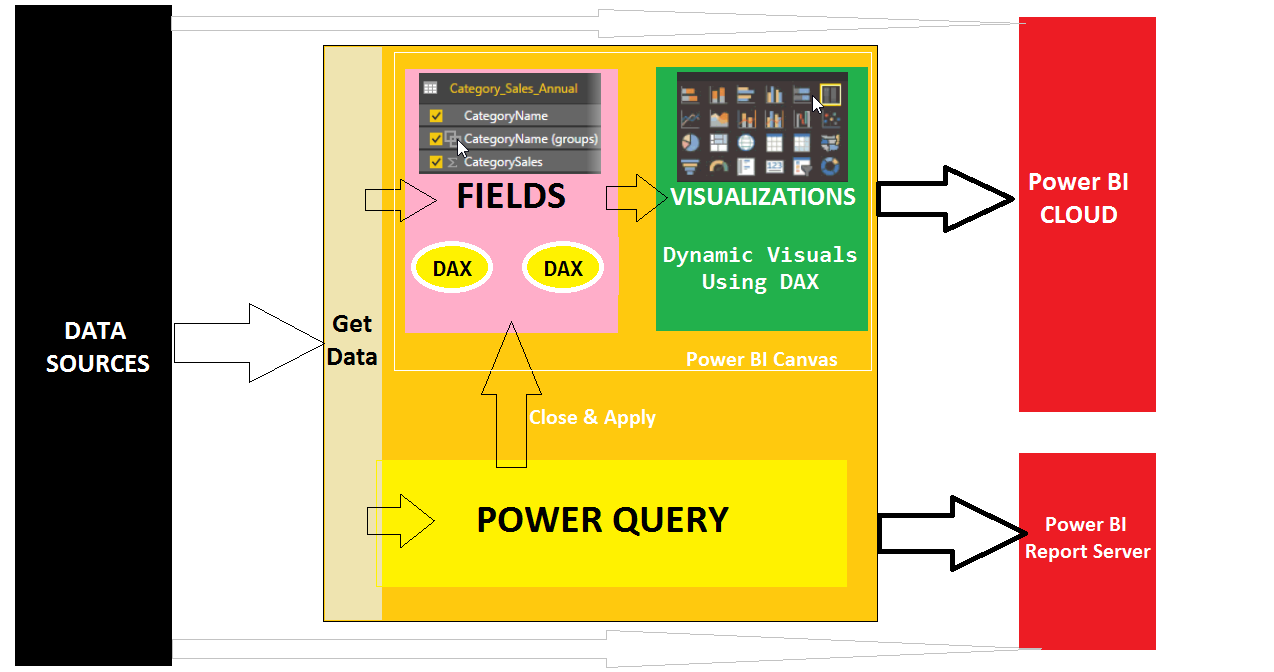
**Scope: Global to all tables in a given Dataset**

**Example: Sum(TOTAL\_SALES)**

2. USING DAX, WE CAN DESIGN (**MODEL**) THE SECURITY ROLES.

3. USING DAX, WE CAN IMPLEMENT **FOR ROW LEVEL SECURITY**(**RLS**) IN POWER BI CLOUD.

4. USING DAX WE CAN DESIGN **ANALYTICAL REPORTS**. POWER BI CLOUD SOURCE, EXCEL REPORT ENV.



**Q3. WHAT ARE THE DATA TYPES IN DAX?**

1. INTEGER 2. REAL 3. CURRENCY

4. BOOLEAN 5. DATETIME 6. STRING

**Q4. WHAT ARE THE DIFFERENT OPERATORS SUPPORTED IN DAX ?**

|  |  |  |
| --- | --- | --- |
|  | **OPERATOR TYPE** | **SYMBOL** & **USE** |
| 1 | Parenthesis Operators | () precedence order and grouping of arguments (parameters) |
| 2 | Arithmetic Operators | + (addition)  - (subtraction/sign)  \* (multiplication)  / (division)  ^ (exponentiation) |
| 3 | Comparison Operators | = (equal to)  > (greater than)  < (less than)  >= (greater than or equal to)  <= (less than or equal to)  <> (not equal to) |
| 4 | Text Concatenation Operators | & (concatenation) |
| 5 | Logic Operators | && (AND)  || (OR) |

**Q5. SYNTAX FOR DAX EXPRESSION :**

We need to specify the Entity Name or Table Name followed by Column Name in [ ]

If Entity Name has space or special characters, then embed it in **' '**

Format: ENTITY NAME[COLUMN NAME]

OR 'ENTITY NAME'[COLUMN NAME]

Example1: TOTAL SALES1 = 'SALES DATA' [SALE 2017] + 'SALES DATA' [SALE 2018] + 'SALES DATA' [SALE 2019]

Example2: TOTAL SALES2 = SUM ('SALES DATA' [SALE 2017])

**Q6. TYPES OF DAX FUNCTIONS**

**Date and Time Functions (DAX)** - These functions in DAX are similar to date and time functions in Microsoft Excel and Microsoft SQL Server T-SQL.

**Time Intelligence Functions (DAX)** - These functions help you create calculations that use built-in knowledge about calendars and dates. By using time and date ranges with aggregations, We build meaningful comparisons across comparable time periods for sales, inventory, and so on.

**Filter Functions (DAX)** - These functions help you return specific data types, look up values in related tables, and filter by related values. Lookup functions work by using tables and relationships between them. Filtering functions let you manipulate data context to create dynamic calculations.

**Information Functions (DAX)** - These functions look at a table or column provided as an argument to another function and tells you whether the value matches the expected type.

**Logical Functions (DAX)** - These functions return information about values in an expression. For example, the TRUE function tell whether an expression that you are evaluating returns a TRUE value.

**Math and Trig Functions (DAX)** - Mathematical functions in DAX are similar to Excel's mathematical and trigonometric functions.

**Parent and Child Functions (DAX)** - These Data Analysis Expressions (DAX) functions help users manage data that is presented as a parent/child hierarchy in their data models (RELATIONSHIP DIAGRAM).

**Statistical Functions (DAX)** - These functions perform aggregations. In addition to creating sums and averages, or finding minimum and maximum values, in DAX you can also filter a column before aggregating or create aggregations based on related tables.

**Text Functions (DAX)** - With these functions, you can return part of a string, search for text within a string, or concatenate string values. Additional functions are for controlling the formats for dates, times, and numbers.

**Other Functions (DAX)** - These functions perform unique actions that cannot be defined by any of the categories most other functions belong to. Ex: CALCUATE

|  |  |
| --- | --- |
| **Date and Time Functions**  DATE Function  DATEVALUE Function  DAY Function  EDATE Function  EOMONTH Function  HOUR Function  MINUTE Function  MONTH Function  NOW Function  SECOND Function  TIME Function  TIMEVALUE Function  TODAY Function  WEEKDAY Function  WEEKNUM Function  YEAR Function  YEARFRAC Function  **FILTER FUNCTION**  ALL Function  ALLEXCEPT Function  ALLNOBLANKROW Function  ALLSELECTED Function  CALCULATE Function  CALCULATETABLE Function  DISTINCT Function  EARLIER Function  EARLIEST Function  FILTER Function  FILTERS Function  HASONEFILTER Function  HASONEVALUE Function  ISCROSSFILTERED Function  ISFILTERED Function  KEEPFILTERS Function  RELATED Function  RELATEDTABLE Function  USERELATIONSHIP Function  VALUES Function | **Time Intelligence Functions**  CLOSINGBALANCEMONTH Function (DAX)  CLOSINGBALANCEQUARTER Function (DAX)  CLOSINGBALANCEYEAR Function (DAX)  DATEADD Function (DAX)  DATESBETWEEN Function (DAX)  DATESINPERIOD Function (DAX)  DATESMTD Function (DAX)  DATESQTD Function (DAX)  DATESYTD Function (DAX)  ENDOFMONTH Function (DAX)  ENDOFQUARTER Function (DAX)  ENDOFYEAR Function (DAX)  FIRSTDATE Function (DAX)  FIRSTNONBLANK Function (DAX)  LASTDATE Function (DAX)  LASTNONBLANK Function (DAX)  NEXTDAY Function (DAX)  NEXTMONTH Function (DAX)  NEXTQUARTER Function (DAX)  NEXTYEAR Function (DAX)  OPENINGBALANCEMONTH Function (DAX)  OPENINGBALANCEQUARTER Function (DAX)  OPENINGBALANCEYEAR Function (DAX)  PARALLELPERIOD Function (DAX)  PREVIOUSDAY Function (DAX)  PREVIOUSMONTH Function (DAX)  PREVIOUSQUARTER Function (DAX)  PREVIOUSYEAR Function (DAX)  SAMEPERIODLASTYEAR Function (DAX)  STARTOFMONTH Function (DAX)  STARTOFQUARTER Function (DAX)  STARTOFYEAR Function (DAX)  TOTALMTD Function (DAX)  TOTALQTD Function (DAX)  TOTALYTD Function (DAX) |
| **Information Functions**  CONTAINS Function  CUSTOMDATA Function  ISBLANK Function  ISERROR Function  ISLOGICAL Function | ISNONTEXT Function  ISNUMBER Function  ISTEXT Function  LOOKUPVALUE Function  PATH Function  PATHCONTAINS Function |
| **Logical Functions**  AND Function  FALSE Function  IF Function  IFERROR Function  NOT Function  OR Function  SWITCH Function  TRUE Function  Math and Trig Functions  ABS Function  CEILING Function  CURRENCY Function  EXP Function  FACT Function  FLOOR Function  INT Function  ISO.CEILING Function  LN Function  LOG Function  LOG10 Function  MOD Function  MROUND Function  PI Function  POWER Function  QUOTIENT Function  RAND Function  RANDBETWEEN Function  ROUND Function  ROUNDDOWN Function  ROUNDUP Function  SIGN Function  SQRT Function  SUM Function  SUMX Function  TRUNC Function | **Statistical Functions**  ADDCOLUMNS Function  AVERAGE Function  AVERAGEA Function  AVERAGEX Function  COUNT Function  COUNTA Function  COUNTAX Function  COUNTBLANK Function  COUNTROWS Function  COUNTX Function  CROSSJOIN Function  DISTINCTCOUNT Function  GENERATE Function  GENERATEALL Function  MAX Function  MAXA Function  MAXX Function  MIN Function  MINA Function  MINX Function  RANK.EQ Function  RANKX Function  ROW Function  STDEV.S Function  STDEV.P Function  STDEVX.S Function  STDEVX.P Function  SUMMARIZE Function  TOPN Function  VAR.S Function  VAR.P Function  VARX.S Function  VARX.P Function  Text Functions  BLANK Function  CONCATENATE Function  EXACT Function  FIND Function  FIXED Function |

**Additional Examples:**

**Q7. WHAT ARE THE CONDITIONS TO WORK WITH DAX EXPRESSIONS IN POWER BI ?**

EVERY DAX EXPRESSION SHOULD EVALUATE TO ONE OF THE BELOW CONTEXT (OR SCOPE):

1. **ROW CONTEXT : DAX OPERATION APPLIED ON ALL ROWS IN A TABLE.**

EX: COMPUTING SUM OF SALES

2. **FILTER CONTEXT : DAX OPERATION APPLIED ON FEW ROWS IN A TABLE.**

EX: COMPUTING SUM OF SALES FOR FEW SPECIFIC PRODUCTS

**Q8. HOW TO DEFINE DAX CALCULATIONS (COLUMNS** OR **MEASURES) FROM POWER BI DESKTOP ?**

1. RIGHT CLICK FIELDS > NEW COLUMN

2. RIGHT CLICK FIELDS > NEW MEASURE

3. RIGHT CLICK FIELDS > QUICK MEASURE

FOR ANALYTICAL REPORTS, ATELAST ONE MEASURE IS MANDATORY.

**PRACTICE LAB - LEVEL 1**

**HOW TO DEFINE NEW CALCULATED COLUMN IN DAX ?**

LAUNCH POWER BI DESKTOP RS > GET DATA > CSV > BROWSE FOR THE GIVEN FILE > LOAD.

FIELDS : RIGHT CLICK TABLE> NEW **COLUMN**> THEN WE SEE ONE DAX EXPRESSION WINDOW.

TYPE BELOW EXPRESSION:

TOTAL SALES = 'SALES DATA' [SALE 2016] + 'SALES DATA' [SALE 2017] + 'SALES DATA' [SALE 2018]

CLICK @ TICK MARK TO VALIDATE (VERIFY) AND THEN COMMIT THE DAX EXPRESSION.

GO TO "DATA" BUTTON AND VERIFY THAT WE SEE THE ABOVE COMPUTED COLUMN AND ITS VALUES.

FIELDS : RIGHT CLICK TABLE > NEW **COLUMN**> THEN WE SEE ONE DAX EXPRESSION WINDOW.

TYPE BELOW EXPRESSION:

IsBlank2016 = ISBLANK('SalesData'[Sale 2016])

CLICK @ TICK MARK TO VALIDATE (VERIFY) AND THEN COMMIT THE DAX EXPRESSION. GO TO "DATA" BUTTON AND VERIFY THAT WE SEE THE ABOVE COMPUTED COLUMN AND ITS VALUES

FIELDS : RIGHT CLICK TABLE> NEW **COLUMN**> THEN WE SEE ONE DAX EXPRESSION WINDOW.

TYPE BELOW EXPRESSION:

ReplacedSale2016 = IF(ISBLANK('SalesData'[Sale 2016]), 0, 'SalesData'[Sale 2016])

CLICK @ TICK MARK TO VALIDATE (VERIFY) AND THEN COMMIT THE DAX EXPRESSION.

GO TO "DATA" BUTTON AND VERIFY THAT WE SEE THE ABOVE COMPUTED COLUMN AND ITS VALUES

**HOW TO DEFINE NEW DAX MEASURE?**

DAX measures are applicable for Runtime Calculations, involve aggregations. Unlike DAX Columns, DAX measures are NOT calculated beforehand. Instead, they are calculated only during Visual Interactions.

FROM FIELDS WINDOW: RIGHT CLICK TABLE > NEW MEASURE > PASTE BELOW CODE:

SUM\_SALES2016 = SUM('SalesData'[ReplacedSale2016])

Go to Data Button: We do not see measure column nor its computations.

|  |  |
| --- | --- |
| **CALCULATED COLUMN** | **CALCULATED MEASURE** |
| Local to a given Entity or Table | Global to all Entities (Tables) in Power BI Report |
| Can be Verified in Data Button | Cannot be Verified in Data Button |
| Pre-Computed Calculations | Runtime Aggregations @ Visual Analytics |
| **Faster**, Operates at Row Level | Slow, Operates at Visual Level |
| Not Auto Indexed | Auto Indexed @ In-Memory Power BI |
| Regular Calculations are allowed (+, -, / \*&....) | Aggregations are mandatory (SUM, AVG, COUNT..) |
| Aggregations are enforced during Reporting | No additional aggregations during reporting |

**HOW TO USE ABOVE DEFINED DAX COLUMNS?**

FROM VISUALIZATIONS PANE > INCLUDE TABLE VISUAL.

SELECT COUNTRY, COMPANY, MONTH, SALE 2017, SALE 2018, SALE 2019 FIELDS.

FORMAT OPTIONS : GRID > SET THE FONT SIZE.

INCLUDE THE 1ST DAX COLUMN : OBSERVE THAT NO ADDITIONAL AGGREGATIONS INVOLVED.

IN NEW PAGE:

INCLUDE A PIE CHART

SELECT COUNTRY, 1ST DAX COLUMN. THEN AVERAGE AGGREGATION IS AUTO ENFORCED.

INCLUDE A TREEMAP

SELECT COUNTRY, 1ST DAX COLUMN. THEN AVERAGE AGGREGATION IS AUTO ENFORCED.

**HOW TO USE ABOVE DEFINED DAX MEASURE?**

IN NEW PAGE:

INCLUDE A PIE CHART

SELECT COUNTRY, 1ST DAX MEASURE. THEN NO AGGREGATION IS AUTO ENFORCED.

DAX COLUMNS: USED FOR ROW BY ROW COMPUTATIONS. DAX MEASURES USED FOR AGGREGATIONS.

**1. PBIX Report 1: DAX Cheat Sheet, Includes all DAX Formulas and Expression Syntax**

**MS Reference @ examples**

**2. PBIX Report 2: DAX Cheat Sheet, Includes all DAX Formulas and Examples with a Biz Scenario**