**New Measures :**

1. Load [Student Details\_DAX.xlsx](https://docs.google.com/spreadsheets/d/1DQNP9KLWZ42Oq2wo9sikSAbtS1nmJkyG/edit?usp=drive_link&ouid=116581504019999500009&rtpof=true&sd=true) excel file.
2. Select marks table and select new column
3. Add Obtained Marks = marks[Mid term Marks] + marks[Final Term Marks]
4. Add new column, Total Marks = 100
5. Add new column, Marks % = marks[Obtained Marks]/ marks[Total Marks]
6. Change data type of Marks % to percent.
7. Drag student name and marks%.
8. Now add Obtained marks and total marks.
9. **sum() : adds values of only column name**
   1. Add new measure, Correct MarksDrag correct marks %
   2. Drag subjects
10. **sumx() :** 
    1. Add measure total marks obtained with sumx(table, expression)
11. **if() :** 
    1. **Add new column, Result**
    2. **Add new Column Change Grade (marks >=65 : “A”, marks <65 and >=50 : “B”, else “c”)**
    3. **Add column, Groups : (subject : math/physics >> group1 else group0)**
    4. **Change Groups : (subjects either math/physics/ computer : group1 else group0**
    5. **Change Groups : Mid term Marks>15 and Final Term Marks>50 or subject computer : group1 else group0**
12. **in() : (subjects either math/physics/ computer : group1 else group0)**
13. **Switch() : for each value returns a new value (abbreviation)**
    1. **Add new column subject\_abbrev** 
       1. **subject\_abbrev**  = SWITCH(marks[Subjects],

"Physics","Py",

"Math","Mh",

"Chemistry","Che",

"Computer","Comp","Other")

1. **iserror() :** 
   1. Add Column, TestValue = if(marks[Subjects]="Computer",0,2)
   2. Iserror = if(ISERROR(marks[Obtained Marks]/marks[TestValue]),"Error","No Error")
2. **iferror() :**
3. Add column, Iferror = IFERROR(marks[Obtained Marks]/marks[TestValue],BLANK())
4. Simpleerror =IF(marks[TestValue]>0,marks[Obtained Marks]/marks[TestValue],BLANK())
5. **divide() :** 
   1. Add column, Iserror = divide(marks[Obtained Marks],marks[TestValue],0)
6. **Use variables :** 
   1. Add measure, Rating w Variable =

VAR obtainedmarks = sumx(marks,marks[Mid term Marks]+marks[Final Term Marks])

VAR totalmark = sum(marks[Total Marks])

VAR percentage = obtainedmarks/ totalmark

RETURN

if(percentage>0.65,"A","B")

1. **Related() : Returns a related value from another table**

**Data Link :** [Toys Details\_DAX.xlsx](https://docs.google.com/spreadsheets/d/1O3dV2HA2BEvrxefbBz7bH9SaLXe-VkKx/edit?usp=drive_link&ouid=116581504019999500009&rtpof=true&sd=true)

* 1. **Select sales table and select new column**, Country = RELATED(Country[CountryName])
  2. **Add new column, city** = RELATED(City[CityName])
  3. **Add new column, Product** = RELATED('Product'[ProductName])
  4. **Make the change in Country formula,**
     1. **Country** = IF(

RELATED(Country[CountryName]) = BLANK(),

"Country not found",

RELATED(Country[CountryName]))

1. **RELATEDTABLE Function : Evaluates a table expression in a context modified by the given filters.**
   1. Select City table, add new column Number of sales = COUNTROWS(RELATEDTABLE(Sales))
   2. Select Product table, add new column Total Sales = COUNTROWS(RELATEDTABLE(Sales))
2. **FILTER Function : Returns a table that represents a subset of another table or expression.**
3. Select new table and add Mumbai Sales = FILTER(Sales,Sales[city]="Mumbai").
4. Select new measure and add Total Sales = SUMX(Sales,Sales[Price]).
5. In report view add table, and add this measure.
6. Now add city to this table and use slicer to change filter context.
7. Select new measure and add Filter Sales = SUMX(FILTER(Sales,Sales[Price]>5),Sales[Price])
8. Add this new measure in the table of report view.

**13. ALL Function : Returns all the rows in a table, or all the values in a column, ignoring**

**any filters that might have been applied. This function is useful for clearing filters**

1. **Select new measure and add All Sales** = SUMX(ALL(Sales),Sales[Price])
2. **Now add this measure** in the table of report view.
3. **Select new measure and add Sales %** = [Total Sales]/[All Sales].
4. Change format to percentage.
5. Add this new measure in the report view table.

**14. ALLSELECTED Function : Removes context filters from columns and rows in the current query, while retaining all other context filters or explicit filters.**

1. Select **new measure and add AllSelected Sales** = SUMX(ALLSELECTED(Sales),Sales[Price])
2. Now, change Sales % to Sales % = [Total Sales]/[AllSelected Sales]

**15. CALCULATE Function : Evaluates an expression in a modified filter context.**

1. Add **new measure** as Count Sales = count(Sales[SalesId])
2. Now create new report page.
3. Add a table and add product and count sales in the table.
4. Add a slices and add country in it.
5. Add new measure, Count Owl Sales = CALCULATE(COUNT(Sales[SalesId]), 'Product'[ProductName]="Olly Owl")
6. Add this measure to the table in report view.
7. Now add city to table and remove product.

**15.2. CALCULATE 2 :** [Student Details\_DAX.xlsx](https://docs.google.com/spreadsheets/d/1tTDquN54lWSoLoyZ3EkTs_ceF8F5Tklw/edit?usp=drive_link&ouid=116581504019999500009&rtpof=true&sd=true)

1. In report page, add table and add subject and total marks obtained.
2. Add new measure, Math Mark = CALCULATE([Total marks obtained],marks[Subjects]="Math")
3. Add this measure in report
4. Remove subject and student name.
5. Now again in the table just add subject and total marks obtained.
6. Add new measure, Grand Total = CALCULATE([Total marks obtained],all(marks[Subjects]))
7. Add this measure to the table
8. Add a slicer with student name.
9. Add new measure, Subject % = DIVIDE([Total marks obtained],[Grand Total])
10. Add this measure to the table.

**16. DATEDIFF()**

1. Add new measure datediffdemo = DATEDIFF(DATE(2021,1,1),TODAY(),HOUR)

**17. VALUES() :** [Toys Details\_DAX.xlsx](https://docs.google.com/spreadsheets/d/1O3dV2HA2BEvrxefbBz7bH9SaLXe-VkKx/edit?usp=drive_link&ouid=116581504019999500009&rtpof=true&sd=true)

1. Add new table as ProductIDs = values(Sales[ProductId])

Create a new Metrics Table using following steps:

1. From the Home tab, select Enter Data.
2. In Create table dialog box, give the table name as Metrics and select Load
3. Create the new measure CountSales = COUNTROWS(Sales).
4. Delete the Column1
5. CountAllProduct = CALCULATE(COUNTROWS(Sales),ALL(Sales))
6. CountAllProductLondon = CALCULATE(COUNTROWS(Sales),Sales[CityName]="London")
7. CountOwls = CALCULATE(COUNTROWS(Sales),Sales[ProductName]="Olly Owl")
8. Cities = VALUES(City[CityName])//error
9. Cities = if(COUNTROWS(VALUES(City[CityName]))=1,VALUES(City[CityName]), "More than one city")
10. Cities = if(HASONEVALUE(City[CityName]), VALUES(City[CityName]), "More than one city")
11. Cities = if(HASONEVALUE(City[CityName]), VALUES(City[CityName]), CONCATENATEX(VALUES(City[CityName]), City[CityName],",",City[CityName],DESC))

**18 . Time-Intelligent Functions**

1. Create a new table with the following function: DateTable = CALENDAR("01-01-2023","31-12-2024")
2. Goto Model view, Drag the Date column from Datetable to the SalesDate column in the Sales table.
3. Year-to-date = CALCULATE(SUM(Sales[Price]), DATESYTD(DateTable[Date]))
4. Previous year 1 = CALCULATE(SUM(Sales[Price]), SAMEPERIODLASTYEAR(DateTable[Date]))

* **Key Points :**

**sumx(), maxx(), averagex(), minx() :** all x suffix functions

iterator functions >> repetitive evaluates expression row by row