1. Load Student details 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.

SUM Function

1. Add new measure, Correct Marks % = SUM(marks[Obtained Marks])/SUM(marks[Total Marks])
2. Drag correct marks %
3. Drag subjects

IF function

1. Add new column, Grade = IF(marks[Obtained Marks]>=65, "Pass","Fail")
2. Change Grade = if(marks[Obtained Marks]>=65,"A",

if(and(marks[Obtained Marks]<65,marks[Obtained Marks]>=50),"B","C"))

1. Add column, Groups = if( or(marks[Subjects]="Math",marks[Subjects]="Physics"),"Group-1","Group-0")
2. Change Groups = if( or( or(marks[Subjects]="Math",marks[Subjects]="Physics"),marks[Subjects]="Computer"),"Group-1","Group-0")
3. Change Groups = if( marks[Subjects]="Math" || marks[Subjects]="Physics" || marks[Subjects]="Computer","Group-1","Group-0")
4. Change Groups = if(or(

and(marks[Mid term Marks]>15,marks[Final Term Marks]>50),marks[Subjects]="Computer"),"Group-1","Group-0")

SWITCH Function

1. Add column Abbr = SWITCH(marks[Subjects],

"Physics","Py",

"Math","Mh",

"Chemistry","Che",

"Computer","Comp","Oth")

IN Function

1. Change Groups = if(marks[Subjects] in {"Math","Physics","Chemistry"},"Group-1","Group-0")

ISERROR Function

1. Add Column, TestValue = if(marks[Subjects]="Computer",0,2)

Iserror = if(ISERROR(marks[Obtained Marks]/marks[TestValue]),"Error","No Error")

IFERROR Function

1. Add column, Iferror = IFERROR(marks[Obtained Marks]/marks[TestValue],BLANK())
2. Simpleerror =IF(marks[TestValue]>0,marks[Obtained Marks]/marks[TestValue],BLANK())

DIVIDE Function

1. Add column, Iserror = divide(marks[Obtained Marks],marks[TestValue],0)

SUM vs SUMX

1. Add measure, summarks = sum(marks[Obtained Marks])
2. In report view add subject and summarks
3. Add measure, SumXMarks = SUMX(marks,marks[Mid term Marks]+marks[Final Term Marks])
4. Rename SumXMarks to Total marks obtained.
5. Add measure, Marks Total = sum(marks[Total Marks])
6. Add measure, Student Marks % = [Total marks obtained]/marks[Marks Total]
7. Add measure, Rating = if([Student Marks %]>0.65,"A","B")

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")

RELATED Function

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

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

"Country not found",

RELATED(Country[CountryName]))

1. Make the change in Country formula, Country = IF(

ISBLANK(RELATED(Country[CountryName])),

"Country not found",

RELATED(Country[CountryName]))

RELATEDTABLE Function

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))

FILTER Function

1. Select new table and add Mumbai Sales = FILTER(Sales,Sales[city]="Mumbai").
2. Select new measure and add Total Sales = SUMX(Sales,Sales[Price]).
3. In report view add table, and add this measure.
4. Now add city to this table and use slicer to change filter context.
5. Select new measure and add Filter Sales = SUMX(FILTER(Sales,Sales[Price]>5),Sales[Price])
6. Add this new measure in the table of report view.

ALL Function

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.

ALLSELECTED Function

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

CALCULATE Function

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.

CALCULATE 2

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.

DATEDIFF

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

VALUES

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))

Time-Intelligent Functions

1. Create a new table with the following function: DateTable = CALENDAR("01-01-2017","31-12-2018")
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]))