/\*  
EVALUATE { 1, 2, 3 }  
EVALUATE { (1), (2), (3) }  
EVALUATE { (1, 2, 3) }  
EVALUATE { 1, DATE(2017, 1, 1), TRUE, "A" }  
  
EVALUATE  
    {  
    (1,2),  
    (3,4),  
    (5,6)  
    }

\*/  
/\*  
EVALUATE  
    {  
        (1.5, DATE(2017, 1, 1), CURRENCY(199.99), "A"),  
        (2.5, DATE(2017, 1, 2), CURRENCY(249.99), "B"),  
        (3.5, DATE(2017, 1, 3), CURRENCY(299.99), "C")  
    }

\*/   
/\*  
DEFINE  
    VAR tbl =   
    {  
        (1.5, DATE(2017, 1, 1), CURRENCY(199.99), "A"),  
        (2.5, DATE(2017, 1, 2), CURRENCY(249.99), "B"),  
        (3.5, DATE(2017, 1, 3), CURRENCY(299.99), "C")  
    }  
EVALUATE  
    tbl  --Directly to table  
EVALUATE  
    UNION(tbl,tbl) --Table expression\*/  
/\*  
DEFINE  
    VAR x = 22  
    VAR y = 7  
EVALUATE  
{  
    ("22+7",22 + 7),  
    ("22-7",22 - 7),  
    ("22\*7",22 \* 7),  
    ("22/7",22 / 7),  
    ("22^7",22 ^ 7)  
  
}  
EVALUATE  
{  
    ("22/7",22/7),  
    ("FLOOR(22/7,0.1)",FLOOR(22/7,0.1)),  
    ("FLOOR(22/7,0.5)",FLOOR(22/7,0.5)),  
    ("FLOOR(22/7,1)",FLOOR(22/7,1)),  
    ("MOD(22,7)",MOD(22,7))  
}  
//Your challenge: Use this method to test CEILING, ROUND, ROUNDUP, ROUNDDOWN functions  
\*/  
/\*  
EVALUATE  
  DataTable("User Name", STRING,  "Password", STRING,{    
        {"User1","Pass-1"},    
        {"User2","Pass-2"},    
        {"User3",},    
        {"User4",""},    
        {"User5",BLANK()},    
        {"User6","Pass-6"}    
      }    
    )  
ORDER BY [Password]

\*/  
  
/\*  
EVALUATE TblEmployee  
EVALUATE TblEmployee ORDER BY TblEmployee[Name] DESC  
EVALUATE TblEmployee ORDER BY TblEmployee[Department] DESC,TblEmployee[Age] ASC  
\*/  
/\*  
EVALUATE  
    --VALUES(TblEmployee[Department])  
    VALUES(TblEmployee)  
\*/  
/\*  
EVALUATE  
    --FILTERS(TblEmployee[Name])        
    FILTERS(TblEmployee[Gender])

\*/  
/\*  
EVALUATE TblEmployee  
EVALUATE  
    ADDCOLUMNS(TblEmployee,"Supervisor Name",LOOKUPVALUE(TblEmployee[Name],TblEmployee[EID],TblEmployee[Supervisor]))      
  
-- Add [Department Name] to TblEmployee  
EVALUATE  
    ADDCOLUMNS(TblEmployee,"Department Name",RELATED(TblDepartment[Department Name]))      
  
-- Add [Head] for department head name to TblDepartment  
EVALUATE  
    ADDCOLUMNS(TblDepartment,"Head",LOOKUPVALUE(TblEmployee[Name],TblEmployee[EID],[HID]))        
\*/  
/\*  
EVALUATE TblDepartment   
EVALUATE  
    ADDCOLUMNS(TblDepartment,  
    "1",SUM(TblEmployee[Basic Salary]),  
    "2",SUMX(TblEmployee,[Basic Salary]),  
    --"3",SUM(RELATEDTABLE(TblEmployee),[Basic Salary]),  
    "3",SUMX(RELATEDTABLE(TblEmployee),[Basic Salary]),  
    "4",CALCULATE(SUMX(TblEmployee,[Basic Salary])),  
    "5",CALCULATE(SUMX(TblEmployee,[Basic Salary]),TblEmployee[Department]="IT"),  
    "6",CALCULATE(SUMX(TblEmployee,[Basic Salary]),TblEmployee[Department]="IT",ALL()),  
    "7",CALCULATE(SUMX(TblEmployee,[Basic Salary]),TblEmployee[Race]="Malay"),  
    "8",CALCULATE(SUMX(TblEmployee,[Basic Salary]),TblEmployee[Race]="Malay",TblEmployee[Department]="IT"),  
    "9",CALCULATE(SUMX(TblEmployee,[Basic Salary]),TblEmployee[Race]="Malay",TblEmployee[Department]="IT",ALL())  
    )

\*/  
/\*  
EVALUATE  
  SUMMARIZECOLUMNS(   
    TblDepartment[Department Name],   
    "Total Basic Salary", SUM(TblEmployee[Basic Salary] )  
  )  
  
EVALUATE  
  ADDMISSINGITEMS (  
    TblDepartment[Department Name],  
      SUMMARIZECOLUMNS(   
        TblDepartment[Department Name],   
        "Total Basic Salary", SUM(TblEmployee[Basic Salary] )  
      ),  
    TblDepartment[Department Name]  
  )

\*/  
/\*  
EVALUATE DISTINCT(TblEmployee[Gender])  
  
EVALUATE UNION(TblODD,TblPRIME)  
  
EVALUATE DISTINCT(UNION(TblODD,TblPRIME))

\*/  
/\*  
EVALUATE {("Result:",COUNTROWS(FILTERS(TblEmployee[Race])))}  
\*/  
/\*  
DEFINE  
VAR   AverageBasicSalary = AVERAGEX ( TblEmployee, [TotalBasicSalary] )  
VAR ClassifiedEmployees =  
    SUMMARIZECOLUMNS (  
        TblEmployee[Name],  
        "Employee Category",  
            IF ( [TotalBasicSalary] >= AverageBasicSalary, "Rich", "Poor" )  
    )  
VAR Result =  
    GROUPBY (  
        ClassifiedEmployees,  
        [Employee Category],  
        "# Customers", COUNTX ( CURRENTGROUP (), 1 )  
    )      
EVALUATE TblEmployee  
EVALUATE {("Average Basic Salary:",ROUND(AverageBasicSalary,2))}  
EVALUATE ClassifiedEmployees  
EVALUATE Result

\*/  
  
/\*  
//SELECT Department,AVG([Basic Salary]) AS "Average Basic Salary" FROM Employee  
//GROUP BY Department  
  
//SELECT Department,AVG([Basic Salary]) AS "Average Basic Salary" FROM Employee  
//GROUP BY Department  
//HAVING AVG([Basic Salary])>5000  
  
DEFINE  
    VAR Tbl =   
      SUMMARIZE (  
        TblEmployee,  
        [Department],  
        --TblDepartment[Department Name],  
        "Average Basic Salary", ROUND(AVERAGE ( TblEmployee[Basic Salary] ),2)  
      )  
EVALUATE Tbl  
EVALUATE FILTER (Tbl,[Average Basic Salary] > 5000)  
\*/  
/\*  
DEFINE  
    VAR Tbl = SUMMARIZE (

TblEmployee, [Name], [Age], TblDepartment[Department Name], TblRace[Race Name] )  
EVALUATE Tbl  
EVALUATE Tbl ORDER BY TblDepartment[Department Name]   
\*/  
/\*  -------------- Your Challenges ---------------------------  
1) Study https://www.sqlbi.com/articles/introducing-summarizecolumns/ to understand the differenct between   
    SUMMARIZE and SUMMARIZECOLUMNS functions  
  
Notes: The following does not need to show departments that are not involve any employee  
2) Prepare a tabluar result to show average age of all employees per department  
3) Prepare a tabluar result to show average age of all FEMALE employees per department  
  
4) Repeat the challenge of 2) and 3) to include departments that are not involved as well  
  
DEFINE  
    MEASURE TblEmployee[Average Age] = CALCULATE(AVERAGEX(TblEmployee,[Age]))  
    MEASURE TblEmployee[Average Female Age] = CALCULATE(AVERAGEX(TblEmployee,[Age]),TblEmployee[Gender]="F")  
EVALUATE  
    SUMMARIZECOLUMNS (  
        TblDepartment[Department Name],  
        "Average Age",TblEmployee[Average Age])  
  
EVALUATE  
    SUMMARIZECOLUMNS (  
        TblDepartment[Department Name],  
        "Average Female Age",TblEmployee[Average Female Age])  
\*/  
/\*  
DEFINE  
    --VAR managementDepartments = DATATABLE("Department Name",STRING,{{"Human Resource"},{"Finance"}})  
    VAR managementDepartments = {"Human Resource","Finance"}  
EVALUATE   
    {  
        ("Result:",    CALCULATE(SUM(TblEmployee[Basic Salary]),   
            TREATAS(managementDepartments,TblDepartment[Department Name]))  
        )  
  
    }  
  
  
--TREATAS can be used as an alternative syntax to apply a filter in CALCULATE/CALCULATETABLE  
DEFINE   
    MEASURE TblEmployee[Technology] =   
        CALCULATE(SUM(TblEmployee[Basic Salary]),TblDepartment[DID] IN {"IT","QA"})  
    MEASURE TblEmployee[Technology (TREATAS)] =   
        CALCULATE(SUM(TblEmployee[Basic Salary]),TREATAS({"IT","QA"},TblDepartment[DID]))  
  
EVALUATE  
    SUMMARIZECOLUMNS (  
        TblEmployee[Gender],  
        "Technology", [Technology],  
        "Technology (TREATAS)", [Technology (TREATAS)]  
    )  
--https://www.mssqltips.com/sqlservertip/5482/how-to-use-the-treatas-function-in-dax/  
--https://docs.microsoft.com/en-us/power-bi/transform-model/desktop-composite-models  
--https://www.sqlbi.com/articles/strong-and-weak-relationships-in-power-bi/  
\*/