Power Query & DAX Function Study Note (2023-01-04 ~ 2023/01/15)

In the table DimCustomer, can you create a new calculated column called FullName which combines Title, FirstName, MiddleName and LastName into one column.

Solution

```
FullName = DimCustomer[Title] & “ “ & DimCustomer[FirstName] & “ “ & DimCustomer[MiddleName] & “ “ & DimCustomer[LastName]
```

Create a new calculated column called NoMiddleName which shows "No middle name" if there is no MiddleName, and a blank if there is a MiddleName.

Solution

```
NoMiddleName = IF(ISBLANK(DimCustomer[MiddleName]), “No middle name”, BLANK())
```

Create a new calculated column called HasBothHouseAndCar.

It should have "Yes" if both HouseOwnerFlag and NumberCarsOwned are at least 1 each. Use the function AND(,)

If not, then you can either do:

a. "No", or

b. If you are up for a challenge, it should have the values "Car Only", "House Only", and "Neither", depending on the values of these fields.

Solution

```
HasBothHouseAndCar = IF(AND(DimCustomer[HouseOwnerFlag] >= 1, DimCustomer[NumberCarsOwned] >= 1), “Yes”, IF(DimCustomer[HouseOwnerFlag] >=1, “House Only”, IF(DimCustomer[NumberCarsOwned] >=1, “Car Only”, “Neither”)))
```

In the table FactInternetSales, can you create a new calculated column called QuarterNumber which gives the QuarterNr of the OrderDate.

Solution

```
QuarterNumber = FactInternetSales[OrderDate].[QuarterNo]
```

Using this calculated column, can you create a new calculated column called Season so that it shows the words Spring, Summer, Autumn and Winter for the numbers 1, 2, 3 and 4 (or you can use 3, 4, 1, 2, if you are in the southern hemisphere!). Use the SWITCH function.

Solution

```
Season = SWITCH(FactInternetSales[QuarterNumber], 1, “Spring”, 2, “Summer”, 3, “Autumn”, 4, “Winter”, “Error”)
```

In the DimSalesTerritory table, create a new calculated column called InUS. It should have the values "In US" or "Outside of US", depending on whether the SalesTerritoryCountry field says "United Sales" or not.

Solution

InUS = IF(DimSalesTerritory[SalesTerritoryCountry] = “United States”, “In US”, “Outside of US”)

Please create a Calculated Column called NumberPeople which shows the number of people in the family.

If MaritalStatus is "M", then assume that there are two people in the family, plus the TotalChildren field.

If MaritalStatus is "S", then assume that there are one person in the family, plus the TotalChildren field.

Solution

NumberPeople = SWITCH(DimCustomer[MaritalStatus], “M”, 2, “S”, 1) + DimCustomer[TotalChildren]

Each family has a YearlyIncome column. Please create a Calculated Column called IncomePerPerson which gives the answer YearlyIncome divided by NumberPeople.

Solution

IncomePerPerson = DimCustomer[YearlyIncome] / DimCustomer[NumberPeople]

Please create a Measure called MeasureIncomePerPerson which gives the average of the income per person, without using all the Calculated Columns we have just created.

Solution

MeasureIncomePerPerson = AverageX(DimCustomer, DimCustomer[YearlyIncome]/ DimCustomer[NumberPeople])

Please create a calculated column called RankEQ which calculates the RANK.EQ of the IncomePerPerson.

Solution

RankEQ = RANK.EQ(DimCustomer[IncomePerPerson], DimCustomer[IncomePerPerson], DESC)

The IncomePerPerson calculated column includes values such as 15714.2857142857. Please round this column to the nearest 1,000, firstly using ROUND, and then change it to use MROUND.

Solution

IncomePerPerson = ROUND(DimCustomer[YearlyIncome] / DimCustomer[NumberPeople], -3)

- Set it as -3 to rounding up to the nearest 1000.

IncomePerPerson = MROUND(DimCustomer[YearlyIncome] / DimCustomer[NumberPeople], 1000)

- Set it as 1000 to rounding up to the nearest 1000.

Please split the RankEQ calculated column into two calculated columns - the number of Hundreds it has, and the last two digits. For example, if RankEQ = 5440, then the "Hundreds" column should equal 54, and the "Remainder" column should equal 40.

Solution

Hundreds = INT(DimCustomer[RankEQ]/100)

Remainder = MOD(DimCustomer[RankEQ],100)

Let's suppose that there is a target of 10,000 income for each person. We have calculated the IncomePerPerson in a previous Practice Activity. Please use the SIGN and SWITCH to create a new Calculated Column called GoalStatus give answers such as "Above by 20,000", "In line with target" or "Below by 2,000".

Solution

GoalStatus = SWITCH(SIGN(DimCustomer[IncomePerPerson] – 10000), 1, “Above by ” & (DimCustomer[IncomePerPerson] - 10000), 0, “In line with target”, -1, “Below by ” & ABS(DimCustomer[IncomePerPerson] – 10000))

Please create a Calculated Column called "AddressStart". It should have the first part of the "AddressLine1" field, up to (but not including) the first space.

Solution

AddressStart = LEFT(DimCustomer[AddressLine1], FIND(“ “, DimCustomer[AddressLine1] & “ “)-1)

- FIND(“ “, DimCustomer[AddressLine1] & “ “): This function will find the position of the first space. Added a space at the end to counter values with no space.
- LEFT: This function will return the number of characters defined at the start of a text string.
- We added -1 at the <num_chars> to remove extra space at the end for those with no space.

Create a Calculated Column called "AddressEnd". It should contain the rest of the "AddressLine1" field, in upper case.

Solution

AddressEnd = MID(DimCustomer[AddressLine1], FIND(“ “, DimCustomer[AddressLine1] & “ “) + 1, LEN(DimCustomer[AddressLine1])

- MID: MID function will return a string of characters from the middle of a text string. We can set a starting position up to the end position.

Create a BirthDateFormatted Calculated Column, which says (for example) "This person was born in May 1975". It should format the BirthDate column using the format "mmmm yyyy".

Solution

BirthDateFormatted = “This person was born in May 1975” & FORMAT(DimCustomer[BirthDate], “mmmm yyyy”)

This table includes the GeographyKey. Can you create a new Calculated Column called "Country" and use the LOOKUPVALUE function to retrieve the EnglishCountryRegionName from the table DimGeography (both of these tables include the GeographyKey field).

Solution

Country = LOOKUPVALUE(DimGeography[EnglishCountryRegionName], DimGeography[Geography], DimCustomer[GeographyKey])

- DimGeography[EnglishCountryRegionName]: The output column
- DimGeography[Geography]: The search column
- DimCustomer[GeographyKey]: The search value

In the FactInternetSales table, use the RELATED function to create a new Calculated Column called EmailAddress, which should retrieve the EmailAddress field from the DimCustomer table.

Solution

EmailAddress = RELATED(DimCustomer[EmailAddress])

- RELATED: The RELATED function will return a related value from another table.

In the DimSalesTerritory table, use the RELATEDTABLE and COUNTROWS functions to create a new Calculated Column called NumberOfSales. It should count the number of relevant rows in the FactInternetSales table.

Solution

NumberOfSales = COUNTROWS(RELATEDTABLE(FactInternetSales))

In the FactInternetSales table, using the CALCULATE and FILTER function, so that it only shows the Sum of SalesAmount where the Country column is "United States". In other words, the result of the Matrix should be as before, except that all of the values except those under "United States" should be BLANK().

Solution

SalesAmountMeasure = CALCULATE(SUM(FactInternetSales[SalesAmount]), FILTER(DimSalesTerritory, DimSalesTerritory[SalesTerritoryCountry] = “United States”))

- CALCULATE: Evaluates an expression in a modified filter context.
- FILTER: Return a table that represents a subset of another table or expression.
- FILTER(DimSalesTerritory, DimSalesTerritory[SalesTerritoryCountry] = “United States”): This will filter out values with “United States” in Country column.

Change this measure again, so that it shows the grand total (use the ALL function. To remove all filters, you could use ALL() without specifying a table.

Can you use this measure to calculate the percentage of original SUM calculation compared to the Grand Total? So, Australia in 2011 was 2,514,328. You could now divide it by the Grand Total of 29,358,677 to get a percentage of the Grand Total.

Solution

SalesAmountMeasure = SUM(FactInternetSales[SalesAmount]) /
CALCULATE(SUM(FactInternetSales[SalesAmount]), ALL())

Change this measure again, so that it ignores the Country context, but uses the other contexts (such as Year) - use the ALLSELECTED function.

Can you use this measure to calculate the percentage of original SUM calculation compared to the Total for that year? So, Australia in 2011 was 2,514,328. You could now divide it by the 2011 Total of 6,852,489 to get a percentage of the Total for 2011.

Solution

SalesAmountMeasure = SUM(FactInternetSales[SalesAmount]) /
CALCULATE(SUM(FactInternetSales[SalesAmount]),
ALLSELECTED(DimSalesTerritory[SalesTerritoryCountry]))

- Rows: Year
- Columns: SalesTerritoryCountry
- Values: SalesAmountMeasure
- ALLSELECTED function will sum of all country's sales together which equals to the total sales of that year.

Add a Measure called Duration to the Matrix, which is the LASTDATE minus the FIRSTDATE of the Due Date, plus one. To convert it to a number, use the FORMAT function and the format "0". Check that it works at different date hierarchy levels.

Solution

Duration = FORMAT(LASTDATE(FactInternetSales[DueDate]) –
FIRSTDATE(FactInternetSales[DueDate]), "0") + 1

- Add "0" to convert dates in number of day.

Add a Measure to the Matrix which shows the STARTOFYEAR.

Solution

STARTOFYEAR = STARTOFYEAR(FactInternetSales[DueDate])

Add the sum of SalesAmount to the Matrix, and create a Measure with a Year to Date figure.

Solution

SalesAmountYTD = CALCULATE(SUM(FactInternetSales[SalesAmount]),
DATESYTD(FactInternetSales[DueDate]))

Calculate a rolling sum over the current and the previous days. Use the CALCULATE and the DATESINPERIOD function, using the DueDate column, FIRSTDATE([DueDate]), -2, DAY.

Solution

RollingSum = CALCULATE(SUM(FactInternetSales[SalesAmount]),
DATESINPERIOD(FactInternetSales[DueDate], FIRSTDATE(FactInternetSales[DueDate], -2, Day))

Use the SAMEPERIODLASTYEAR function to get the SalesAmount from the previous year.

Solution

PreviousYear = CALCULATE(SUM(FactInternetSales[SalesAmount]), SAMEPERIODLASTYEAR(FactInternetSales[DueDate]))

Alter this formula to use the PARALLELPERIOD function to get the total for the current month.

Solution

TotalCurrentMonth = CALCULATE(SUM(FactInternetSales[SalesAmount]), PARALLELPERIOD(FactInternetSales[DueDate], 0, MONTH))