

([DAX])

Advanced DAX for Power BI

Chapters 1

Chapters	Description	Page
1) Helpful DAX Resources	<ul style="list-style-type: none">• Introduction - Helpful DAX Resources	6
2) The DAX Engines	<ul style="list-style-type: none">• Introduction - The DAX Engines• DAX Formula & Storage Engines• DAX Query Evaluation In Depth• Data & Storage Types• VertiPaq Columnar Data Structure• VertiPaq Compression & Encoding• Value Encoding• Hash Encoding• Run Length Encoding• VertiPaq Relationships• Summary: DAX Engines	38
3) Tips & Best Practices	<ul style="list-style-type: none">• Introduction - Tips & Best Practices• DAX Shortcuts• Formatting Best Practices• DAX Evaluation Order• Commenting Your Code• PRO TIP: Dedicated Measure Table• Error Handling• Creating & Using DAX Variables• Variable Evaluation Order• PRO TIP: Using Variables to Text & Debug DAX	53
4) Scalar Functions	<ul style="list-style-type: none">• Introduction - Scalar Functions• Common Scalar Functions• Aggregation Functions• PRO TIP: SUM & SUMX• Rounding Functions• ASSIGNMENT Rounding Functions• Information Functions• Conversion Functions• ASSIGNMENT Conversion Functions• Logical Functions: SWITCH• Logical Functions: COALESCE• ASSIGNMENT Logical Functions	109

Chapters 2

Chapters	Description	Page
5) Advanced CALCULATE	<ul style="list-style-type: none">• Introduction - Advanced CALCULATE• Expanded Tables• Context Transition• Evaluation Order• CALCULATE Modifiers• REMOVEFILTERS• KEEPFILTERS• ASSIGNMENT Advanced CALCULATE• PRO TIP: Common CALCULATE Patterns	149
6) Table & Filter Functions	<ul style="list-style-type: none">• Introduction - Table & Filter Functions• Common Table & Filter Functions• Review: Calculated Tables• DISTINCT• VALUES• ASSIGNMENT: VALUES & DISTINCT BLANK ROW• SELECTEDVALUE• ALLEXCEPT• ASSIGNMENT: ALLEXCEPT• ALLSELECTED• ASSIGNMENT: ALLSELECTED• SELECTCOLUMNS• ADDCOLUMNS• SUMMARIZE• ASSIGNMENT: SUMMARIZE• Generating New Data• ROW• DATATABLE• GENERATESERIES• The Table Constructor• ASSIGNMENT Generating Data	198

Chapters 3

Chapters	Description	Page
7) Calculated Table Joins	<ul style="list-style-type: none"> • Introduction - Calculated Table Joins • CROSSJOIN • UNION • EXCEPT • INTERSECT • ASSIGNMENT Calculated Table Joins 	305
8) Relationship Functions	<ul style="list-style-type: none"> • Introduction - Relationship Functions • Physical vs. Virtual Relationships • RELATED • RELATEDTABLE • ASSIGNMENT RELATED & RELATEDTABLE • USERRELATIONSHIPS • CROSSFILTER • ASSIGNMENT CROSSFILTER • TREATAS • ASSIGNMENT Sales Targets (TREATAS) 	326
9) Iterator Functions	<ul style="list-style-type: none"> • Introduction - Iterator Functions • Review: Basic Iterators • Iterator Cardinality • CONCATENATEX • PRO TIP: Adding Dynamic Labels • ASSIGNMENT: CONCATENATEX • AVERAGEX • PRO TIP: Moving Averages • ASSIGNMENT: Moving Averages • RANKX • ASSIGNMENT: RANKX 	379

Chapters 4

Chapters	Description	Page
10) Advanced Time Intelligence	<ul style="list-style-type: none">• Introduction - Advanced Time Intelligence• Automatic Date Tables• Date Table Requirements• CALENDAR• CALENDARAUTO• PRO TIP: Building a Reusable Date Table• Date Formatting• ASSIGNMENT: Date Formatting• Common Time Intelligence Functions• PARALLELPERIOD• PREVIOUSQUARTER• SAMEPERIODLASTYEAR• ASSIGNMENT: Time Periods• Week-based Calculations• Previous Fiscal Week• Fiscal Period to Date• Fiscal Previous Period• ASSIGNMENT: 4-5-4 Calendar	435
11) Performance Tuning	<ul style="list-style-type: none">• Introduction - Performance Tuning• Power BI Performance Analyzer• PRO TIP: Copy Query• DAX Studio• DAX Optimization Workflow	511

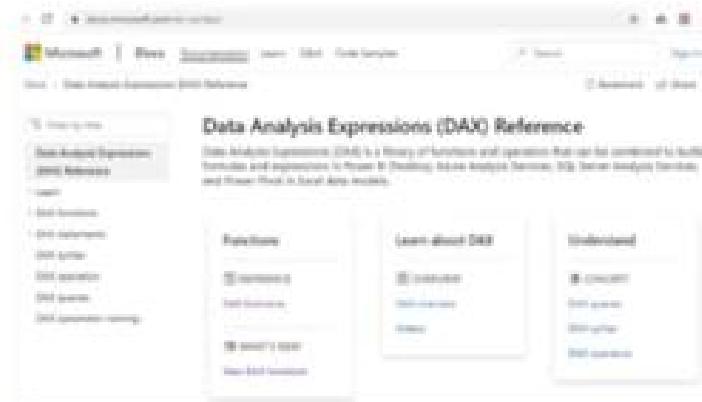
1) Helpful DAX Resources

Introduction - Helpful DAX Resources

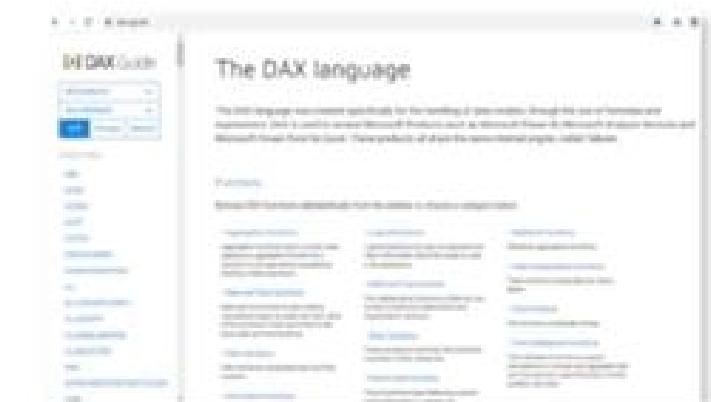


```
Sales based on selected store =  
VAR Store =  
    SELECTVALUE ('Store Lookup'[sales_outlet_id])  
RETURN  
    CALCULATE (  
        SUM('Sales by Store'[quantity]),  
        FILTER ('Sales by Store', 'Sales by Store'[sales_outlet_id] = Store)  
    )
```

DAX Formatter (daxformatter.com) by sqlbi.com is a great tool for cleaning and formatting your DAX code, with options to customize based on regional settings or personal preferences



[www.Docs.Microsoft.com/en-us/dax](https://docs.microsoft.com/en-us/dax/) is the official Microsoft DAX reference guide, and a great resource for exploring DAX functions



DAX Guide (dax.guide) by sqlbi.com is a great resource to learn and explore DAX functions, category groups, product compatibility, and more

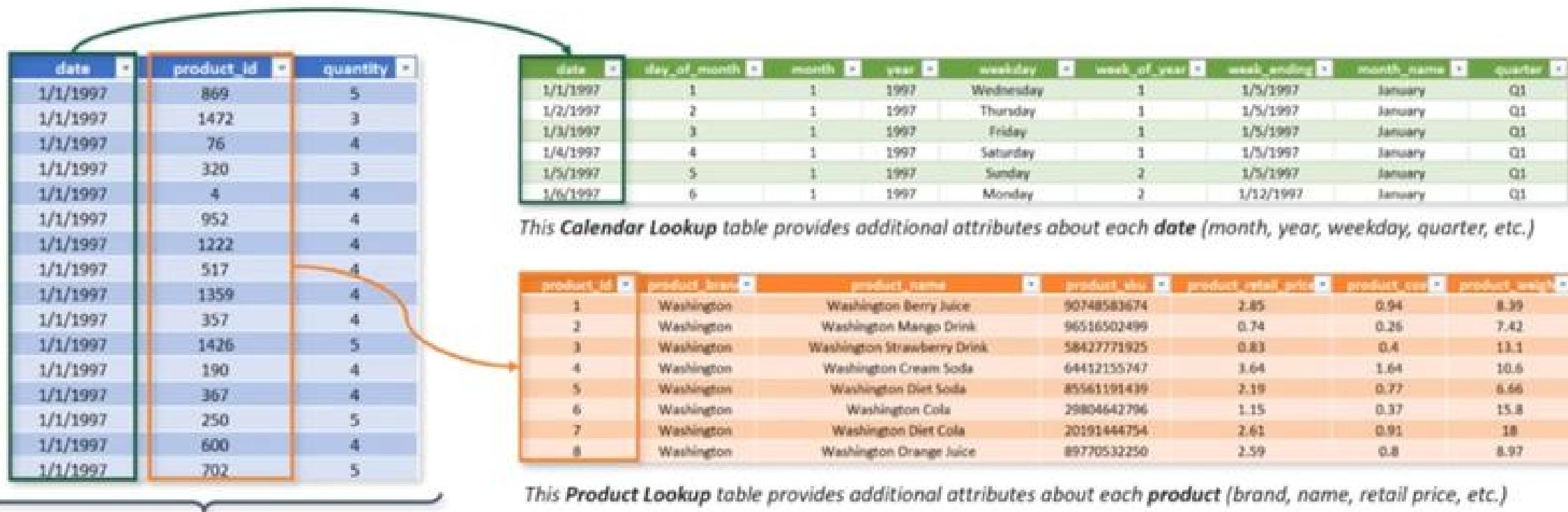


DAX Studio is an open source tool that allows you to write, execute and analyze DAX queries, as well as troubleshoot and optimize your code

DATA TABLES VS. LOOKUP TABLES

Models generally contain two types of tables: **data** (or “*fact*”) tables, and **lookup** (or “*dimension*”) tables

- **Data tables** contain measurable *values* or *metrics* about the business (*quantity, revenue, pageviews, etc.*)
- **Lookup tables** provide descriptive *attributes* about each dimension in your model (*customers, products, etc.*)



This **Data Table** contains “*quantity*” values, and connects to **lookup tables** via the “*date*” and “*product_id*” columns

PRIMARY VS. FOREIGN KEYS

date	product_id	quantity
1/1/1997	869	5
1/1/1997	1472	3
1/1/1997	76	4
1/1/1997	320	3
1/1/1997	4	4
1/1/1997	952	4
1/1/1997	1222	4
1/1/1997	517	4
1/1/1997	1359	4
1/1/1997	357	4
1/1/1997	1426	5
1/1/1997	190	4
1/1/1997	367	4
1/1/1997	250	5
1/1/1997	600	4
1/1/1997	702	5



These columns are **foreign keys**; they contain *multiple* instances of each value, and are used to match the **primary keys** in related lookup tables

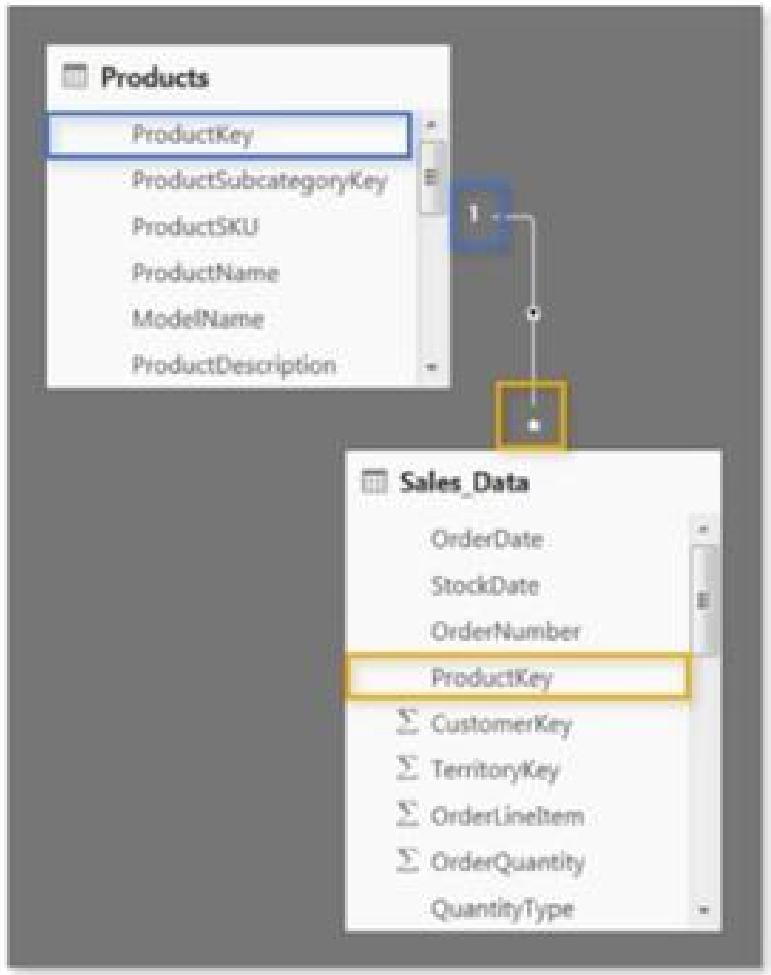
date	day_of_month	month	year	weekday	week_of_year	week_starting	month_name	quarter
1/1/1997	1	1	1997	Wednesday	1	1/5/1997	January	Q1
1/2/1997	2	1	1997	Thursday	1	1/5/1997	January	Q1
1/3/1997	3	1	1997	Friday	1	1/5/1997	January	Q1
1/4/1997	4	1	1997	Saturday	1	1/5/1997	January	Q1
1/5/1997	5	1	1997	Sunday	2	1/5/1997	January	Q1
1/6/1997	6	1	1997	Monday	2	1/12/1997	January	Q1

product_id	product_name	product_desc	product_size	product_retail_price	product_cost	product_wt
1	Washington	Washington Berry Juice	807485803674	2.85	0.94	8.39
2	Washington	Washington Mango Drink	960165024999	0.74	0.26	7.42
3	Washington	Washington Strawberry Drink	584277715925	0.83	0.4	13.1
4	Washington	Washington Cream Soda	64412155747	3.64	1.64	10.6
5	Washington	Washington Diet Soda	855611914399	2.19	0.77	6.66
6	Washington	Washington Cola	29804642796	1.15	0.37	15.8
7	Washington	Washington Diet Cola	20191444754	2.61	0.91	18
8	Washington	Washington Orange Juice	89770532250	2.59	0.8	8.97



These columns are **primary keys**; they *uniquely* identify each row of a table, and match the **foreign keys** in related data tables

RELATIONSHIP CARDINALITY



Cardinality refers to the *uniqueness of values* in a column

- For our purposes, all relationships in the data model should follow a “**one-to-many**” cardinality; **one** instance of each *primary key*, but potentially **many** instances of each *foreign key*

*In this case, there is only **ONE** instance of each **ProductKey** in the **Products** table (noted by the “1”), since each row contains **attributes of a single product** (*Name, SKU, Description, Retail Price, etc*)*

*There are **MANY** instances of each **ProductKey** in the **Sales_Data** table (noted by the asterisk *), since there are **multiple sales associated with each product***

FILTER FLOW



Here we have two data tables (**Sales_Data** and **Returns_Data**), connected to **Territory_Lookup**

Note the filter directions (shown as arrows) in each relationship; by default, **these will point from the “one” side of the relationship (lookups) to the “many” side (data)**

- When you filter a table, that filter context is passed along to all related “*downstream*” tables (following the direction of the arrow)
- Filters **cannot** flow “*upstream*” (against the direction of the arrow)



PRO TIP:

*Arrange your lookup tables **above** your data tables in your model as a visual reminder that filters flow “*downstream*”*

FILTER FLOW (CONT.)



In this case, the only valid way filter both Sales and Returns data by Territory is to use the **TerritoryKey** field from the **Territory_Lookup** table, which is upstream and related to both data tables

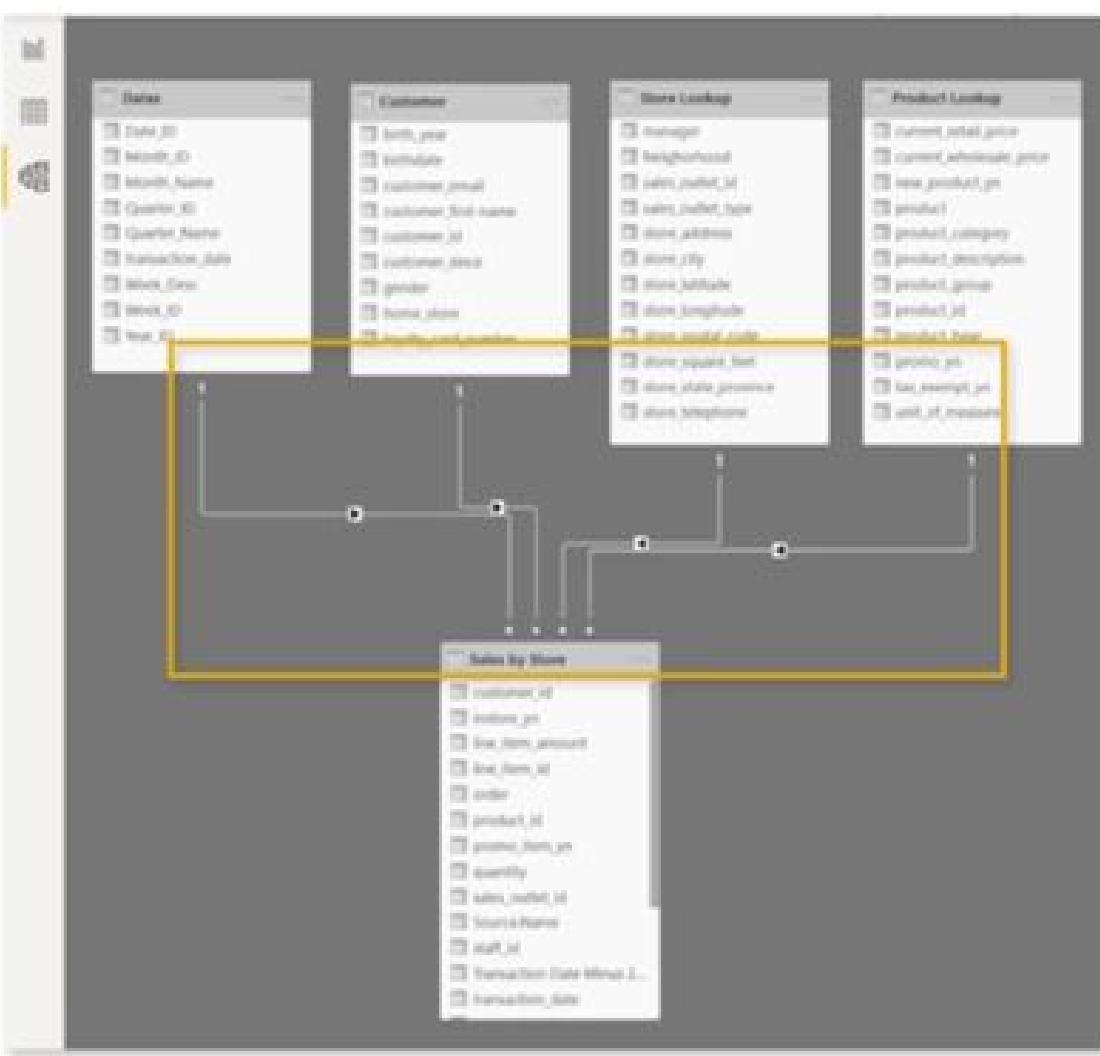
- Filtering using **TerritoryKey** from the **Sales** table yields incorrect **Returns** values, since the filter context **cannot flow upstream** to either one of the other tables
- Similarly, filtering using **TerritoryKey** from the **Returns** table yields incorrect **Sales** data; in addition, **only territories that registered returns are visible in the table** (even though they registered sales)

1) Filtering using TerritoryKey from the Territory_Lookup table

2) Filtering using TerritoryKey from the Sales_Data table

3) Filtering using TerritoryKey from the Returns_Data table

DATA MODEL BEST PRACTICES



A well-designed model is **critical** and ideally should:

- ✓ Use a star schema with **one-to-many** ($1:*$) relationships
- ✓ Contain relationships with **one-way filters** (vs. *bidirectional*)
- ✓ Contain tables that each serve a *specific purpose*, including **data (fact) tables** and **lookup (dim) tables**
- ✓ Only include the data you need for analysis (*no redundant or unnecessary records or fields*)
- ✓ Split out individual **date** and **time** components from **DateTime** fields

DAX OPERATORS

Arithmetic Operator	Meaning	Example
+	Addition	2 + 7
-	Subtraction	5 - 3
*	Multiplication	2 * 6
/	Division	4 / 2
^	Exponent	2 ^ 5

Comparison Operator	Meaning	Example
=	Equal to	[City] = "Boston"
>	Greater than	[Quantity] > 10
<	Less than	[Quantity] < 10
>=	Greater than or equal to	[Unit_Price] >= 2.5
<=	Less than or equal to	[Unit_Price] <= 2.5
<>	Not equal to	[Country] <> "Mexico"

Text/Logical Operator	Meaning	Example
&	Concatenates two values to produce one text string	[City] & " " & [State]
&&	Create an AND condition between two logical expressions	(([State] = "MA") && ([Quantity] > 10))
(double pipe)	Create an OR condition between two logical expressions	(([State] = "MA") ([State] = "CT"))
IN	Creates a logical OR condition based on a given list (using curly brackets)	'Store Lookup'[State] IN { "MA", "CT", "NY" }

*Head to www.microsoft.com for more information about DAX syntax, operators, troubleshooting, etc.

COMMON DAX FUNCTION CATEGORIES

MATH & STATS Functions

Basic aggregation functions as well as "iterators" evaluated at the row-level

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

Functions for returning information about values in a given conditional expression

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

Functions to manipulate text strings or control formats for dates, times or numbers

Common Examples:

- CONCATENATE
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- PROPER
- LEN
- SEARCH/FIND
- REPLACE
- REPT
- SUBSTITUTE
- TRIM
- UNICHAR

FILTER Functions

Lookup functions based on related tables and filtering functions for dynamic calculations

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- RELATED
- RELATEDTABLE
- DISTINCT
- VALUES
- EARLIER/EARLIEST
- HASONEVALUE
- HASONEFILTER
- ISFILTERED
- USERELATIONSHIP

DATE & TIME Functions

Basic date and time functions as well as advanced time intelligence operations

Common Examples:

- DATEDIFF
- YEARFRAC
- YEAR/MONTH/DAY
- HOUR/MINUTE/SECOND
- TODAY/NOW
- WEEKDAY/WEEKNUM

Time Intelligence Functions:

- DATESYTD
- DATESQTD
- DATESMTD
- DATEADD
- DATESINPERIOD

CALCULATE

CALCULATE()

Evaluates a given expression or formula under a set of defined filters

=CALCULATE(Expression, [Filter1], [Filter2],...)

Name of an existing measure, or a DAX formula for a valid measure

Examples:

- `[Total Orders]`
- `SUM(Returns[ReturnQuantity])`

*List of simple Boolean (True/False) filter expressions
(Note: these require simple, fixed values; you cannot create filters based on other measures)*

Examples:

- `Territory_Lookup[Country] = "USA"`
- `Calendar[Year] > 1998`

PRO TIP:

 CALCULATE works just like **SUMIF** or **COUNTIF** in Excel, except it can evaluate measures based on ANY sort of calculation (not just a sum, count, etc.); it may help to think of it like "**CALCULATEIF**"

CALCULATE (EXAMPLE)

```
X ✓ Bike Returns = CALCULATE([Total Returns], Products[CategoryName] = "Bikes") ▾
```

CategoryName	Total Returns	Bike Returns
Accessories	1,115	342
Bikes	342	342
Clothing	267	342
Components		342
Total	1,724	342

Here we've defined a new measure named "**Bike Returns**", which evaluates the "**Total Returns**" measure when the *CategoryName* in the **Products** table equals "**Bikes**"

Wait, why do we see the **same repeating values** when we view a matrix with different categories on rows?

Shouldn't these cells have different filter contexts for **Accessories**, **Clothing**, **Components**, etc?



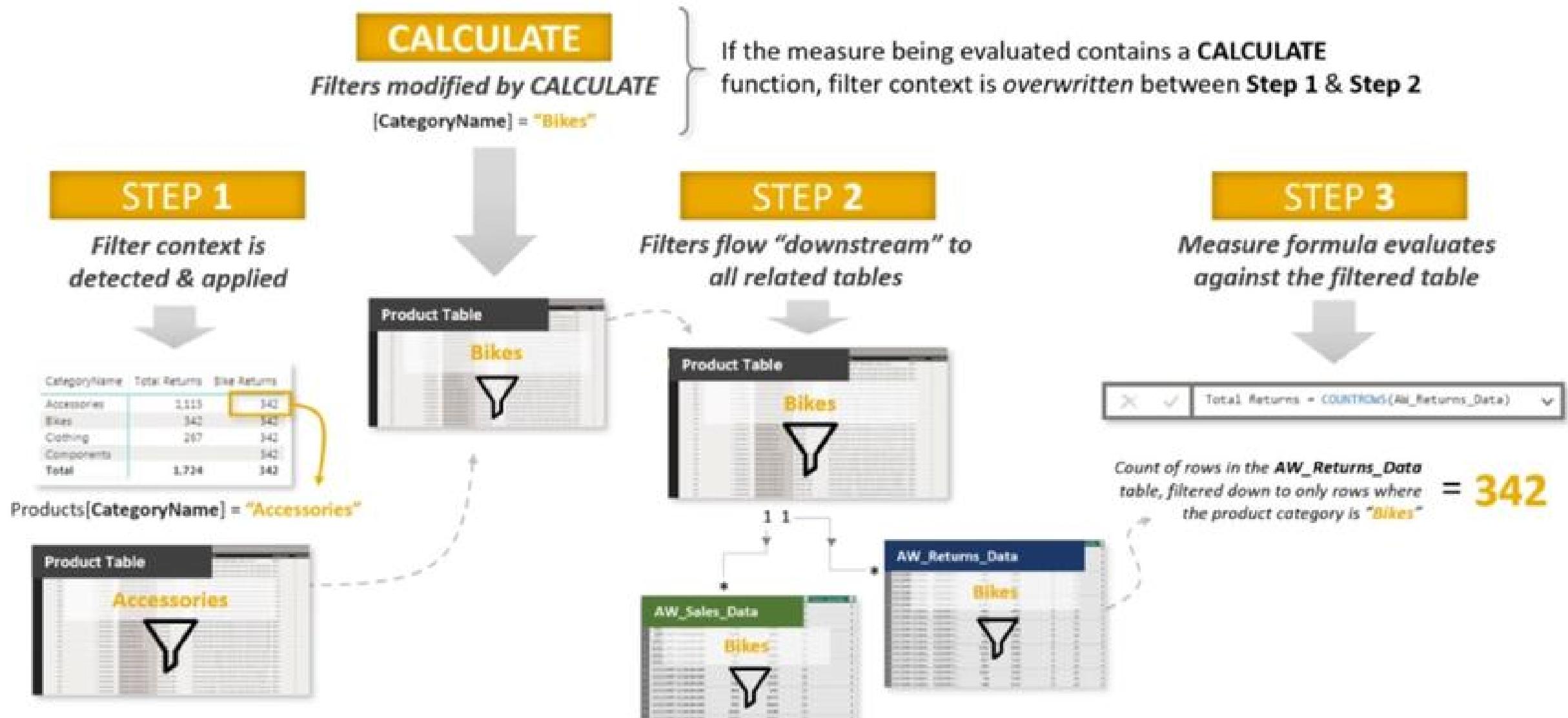
HEY THIS IS IMPORTANT!

CALCULATE **modifies** and **overrules** any competing filter context!

In this example, the "Clothing" row has filter context of CategoryName = "Clothing" (*defined by the row label*) **and** CategoryName= "Bikes" (*defined by the CALCULATE function*)

Both cannot be true at the same time, so the "Clothing" filter is overwritten and the "Bikes" filter (from CALCULATE) takes priority

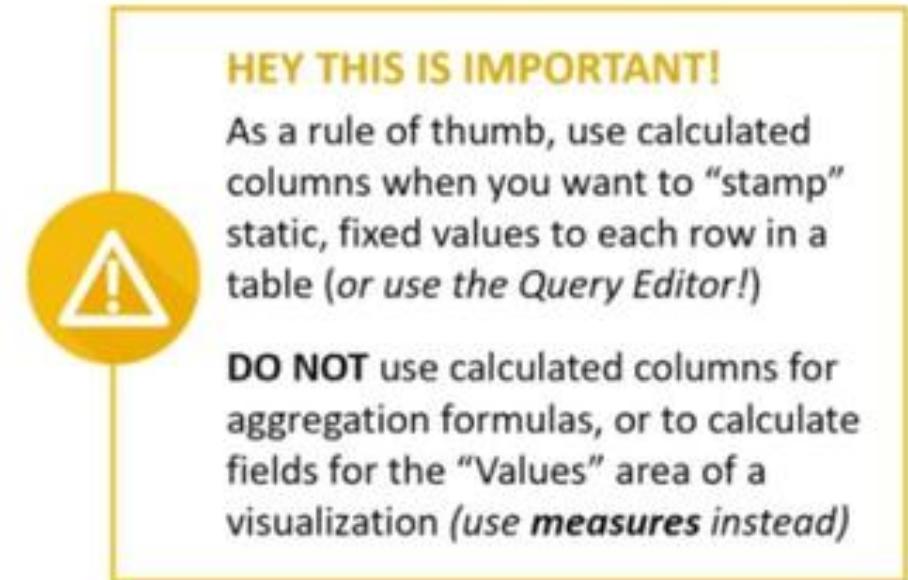
CALCULATE CREATES NEW FILTER CONTEXT



CALCULATED COLUMNS

Calculated columns allow you to add new, formula-based columns to tables

- No “A1-style” references; calculated columns refer to **entire tables or columns**
- Calculated columns generate values for each row, which are **visible within tables in the Data view**
- Calculated columns understand **row context**; they’re great for defining properties based on information in each row, but generally useless for aggregation (*SUM*, *COUNT*, etc.)



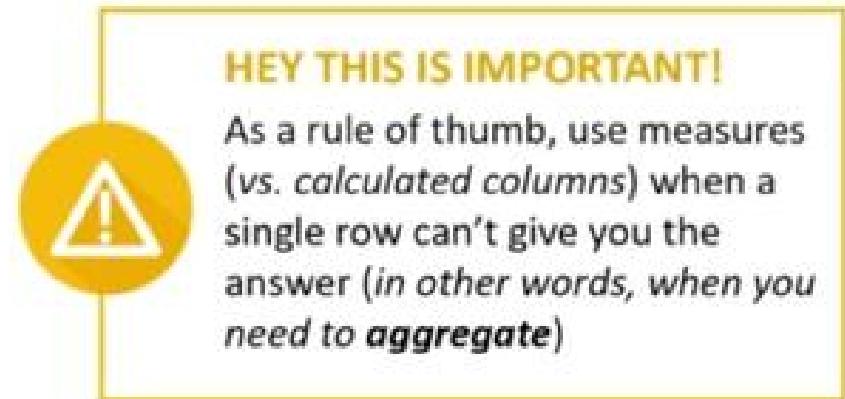
PRO TIP:

Calculated columns are typically used for filtering data, rather than creating numerical or aggregated values

MEASURES

Measures are DAX formulas used to generate new calculated values

- Like calculated columns, measures reference **entire tables or columns (no A1-style or “grid” references)**
- *Unlike calculated columns, **measure** values aren’t visible within tables; they can only be “seen” within a visualization like a chart or matrix (similar to a calculated field in an Excel pivot)*
- Measures are evaluated based on **filter context**, which means they recalculate when the fields or filters around them change (*like when new row or column labels are pulled into a matrix or when new filters are applied to a report*)



PRO TIP:

Use measures to create numerical, calculated values that can be analyzed in the “values” field of a report visual

EVALUATION CONTEXT

Filter & row context tells DAX exactly how to evaluate measures and calculated columns in your data model (this is key for understanding advanced DAX topics)

FILTER CONTEXT

- Filter context **filters** the tables in your data model
- DAX creates filter context when dimensions are added to **rows, columns, slicers & filters** in a report
- **CALCULATE** can be used to systematically create or modify existing filter context
- Filter context always travels (propagates) from the **ONE** side to the **MANY** side of a table relationship

ROW CONTEXT

- Row context **iterates** through the rows in a table
- DAX creates row context when you add **calculated columns** to your data model
- **Iterator functions (SUMX, RANKX, etc.)** use row context to evaluate row-level calculations
- Row context doesn't automatically propagate through table relationships (*need to use **RELATED** or **RELATEDTABLE** functions*)

DATA MODEL SETUP

Untitled - Power BI Desktop

File Home Insert Modeling View Help

Power Get Data Excel Power BI SQL Enter Recent Transform, Refresh New Text More Publish

Data datasets Server data sources - Visuals Visuals Calculations Share

Filters Visualizations Fields

Search

Filters on this page Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Keep all filters On

Add drill-through fields here

Save Cancel

File name: Power BI Dummy Project

Save as type: Power BI file (.pbix)

Organize New folder

Name Date modified Type Size

Advanced DAX for Power BI 15/10/2020 13:33 File folder

old 07/10/2020 15:29 File folder

Created Training Guides > Advanced DAX for Power BI - Training Guide

Page 1 +

Open up Power BI, provide a name & Save the Power BI file accordingly.

Power BI Dummy Project - Power BI Desktop

File Home Insert Modeling View Help

Transform data +

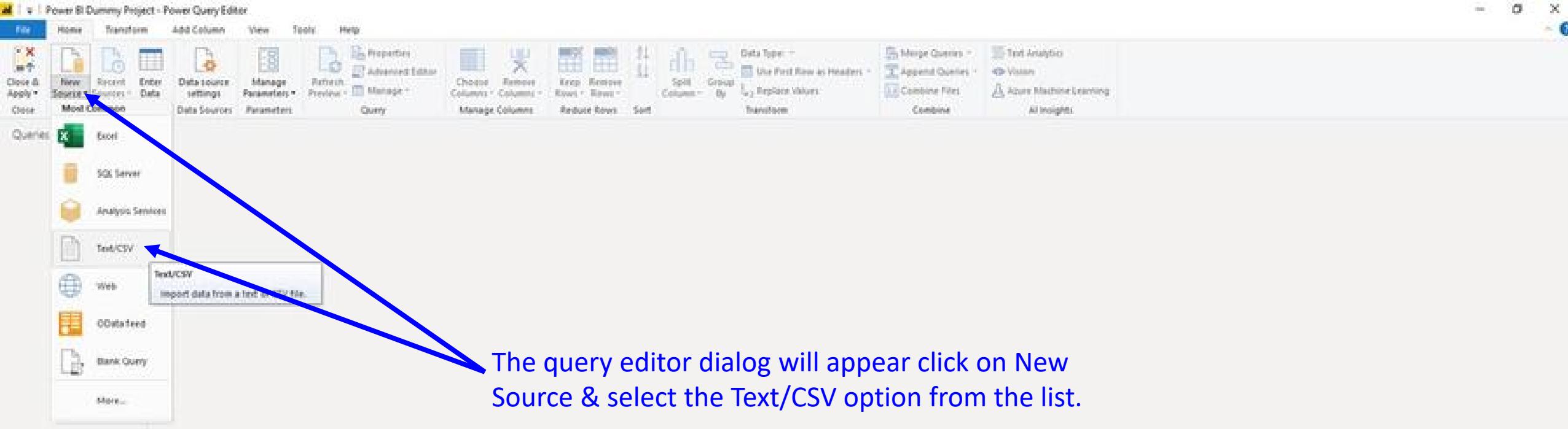
Use the Power Query editor to connect, prepare, and transform data.

Filters

Visualizations Fields

Click on Transform data to head over to the query editor.

The screenshot shows the Microsoft Power BI Desktop ribbon. The 'Home' tab is selected. A blue arrow points from the text 'Click on Transform data to head over to the query editor.' to the 'Transform data' button in the ribbon. A tooltip above the button says 'Use the Power Query editor to connect, prepare, and transform data.' To the right of the ribbon, there are sections for 'Filters', 'Visualizations', and 'Fields'. The 'Filters' section contains buttons for 'Add data fields here' and 'Add para fields here'. The 'Visualizations' section shows various chart icons, and the 'Fields' section shows field icons.



The query editor dialog will appear click on New Source & select the Text/CSV option from the list.

Also import these text files one by one for the data model **except Product Lookup (Updated)**.

Connect the Calendar table by selecting the Calendar file & clicking on open.

The screenshot shows the Power BI Query Editor interface with a 'Queries (0)' tab. Above the editor is a ribbon menu with tabs like File, Home, Transform, Add Column, View, Tools, and Help. Below the ribbon are various icons for managing data sources, queries, and transformations. The main area of the editor is currently empty, indicated by the '(0)' in the title bar. An 'Open' dialog box is overlaid on the editor, showing a file selection window. The window has a title bar 'Open' and a path bar '1. Getting Started - DONE / 1.1 Advanced DAX Data Files'. It lists several files: '4-3-4 Calendar' (a folder), 'Calendar' (selected and highlighted with a blue border), 'Customer Lookup', 'Employee Lookup', 'Food Inventory', 'Product Lookup (Updated)', 'Product Lookup', 'Sales by Store', and 'Store Lookup'. The 'Calendar' file is selected. The 'File name' field at the bottom contains 'Calendar'. The 'Open' button is highlighted with a blue arrow. The 'Cancel' button is also visible.



Queries [0]

0

File Origin: 1232: Western European (Windows) Delimiter: Comma Data Type Detection: Based on first 200 rows.

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Yr
01/01/2017	20170001	1	Week 1	1	January	1	Q1	2017	Week 1-1
02/01/2017	20170002	1	Week 1	1	January	1	Q1	2017	Week 1-2
03/01/2017	20170003	1	Week 1	1	January	1	Q1	2017	Week 1-3
04/01/2017	20170004	1	Week 1	1	January	1	Q1	2017	Week 1-4
05/01/2017	20170005	1	Week 1	1	January	1	Q1	2017	Week 1-5
06/01/2017	20170006	1	Week 1	1	January	1	Q1	2017	Week 1-6
07/01/2017	20170007	1	Week 1	1	January	1	Q1	2017	Week 1-7
08/01/2017	20170008	1	Week 1	1	January	1	Q1	2017	Week 1-8
09/01/2017	20170009	2	Week 2	1	January	1	Q1	2017	Week 2-1
10/01/2017	20170010	2	Week 2	1	January	1	Q1	2017	Week 2-2
11/01/2017	20170011	2	Week 2	1	January	1	Q1	2017	Week 2-3
12/01/2017	20170012	2	Week 2	1	January	1	Q1	2017	Week 2-4
13/01/2017	20170013	2	Week 2	1	January	1	Q1	2017	Week 2-5
14/01/2017	20170014	2	Week 2	1	January	1	Q1	2017	Week 2-6
15/01/2017	20170015	3	Week 3	1	January	1	Q1	2017	Week 3-1
16/01/2017	20170016	3	Week 3	1	January	1	Q1	2017	Week 3-2
17/01/2017	20170017	3	Week 3	1	January	1	Q1	2017	Week 3-3
18/01/2017	20170018	3	Week 3	1	January	1	Q1	2017	Week 3-4
19/01/2017	20170019	3	Week 3	1	January	1	Q1	2017	Week 3-5
20/01/2017	20170020	3	Week 3	1	January	1	Q1	2017	Week 3-6

The data preview dialog box will appear
everything looks good click on OK.

The data preview dialog box will appear everything looks good click on OK.

This screenshot shows the Microsoft Power BI Query Editor interface. The main area displays a table with columns: Transaction_Date, p2_date_id, p2_week_id, p2_week_desc, p2_month_id, p2_month_name, p2_quarter_id, p2_quarter_name, and p2_year_id. A blue arrow points from the text above to the 'OK' button in a data preview dialog box overlaid on the table. The dialog box contains fields for 'File Origin' (1352: Western European (Windows)), 'Delimiter' (Comma), and 'Data Type Detection' (Based on first 200 rows). The 'APPLIED STEPS' pane on the right shows a step labeled 'Changed Type'.

The data preview dialog box will appear everything looks good click on OK.

File Home Transform Add Column View Tools Help

New Recent Enter Data Data source settings Manage Refresh Preview Advanced Editor Properties Choose Columns Remove Columns Keep Rows Remove Rows Split Column Group By Replace Values Data Type: Whole Number Use First Row as Headers Merge Queries Append Queries Combine Files Combine Azure Machine Learning AI insights

Queries (1) P1_customer_M P2_home_store A1_customer_first_name A1_customer_email A1_customer_since A1_lowest_card_number Birthdate A1_gender A1_birth_year

Customer Lookup

File Origin Delimiter Data Type Detection

1253: Western European (Windows) Comma Based on first 200 rows

staff_id	first_name	last_name	position	start_date	location
1	Marc	Brewer	CFO	01/08/2001	HQ
2	Ivan	Leinen	CEO	01/08/2001	HQ
3	Jamie	Tozer	Head Roaster	24/03/2007	WH
4	Chellise	Claudia	Roaster	01/03/2008	WH
5	Adam	Songs	Head Barista	02/04/2008	WH
6	Karen	Coppo	Store Manager	24/03/2009	3
7	Reiley	Cameron	Coffee Wrangler	18/02/2009	3
8	Hamilton	Elli	Coffee Wrangler	09/03/2009	3
9	Caldwell	Veda	Coffee Wrangler	09/03/2012	3
10	Ima	Winsted	Coffee Wrangler	18/12/2016	3
11	Ruth	Leslie	Store Manager	27/06/2009	4
12	Britanni	Iorden	Coffee Wrangler	25/03/2006	4
13	Berk	Derek	Coffee Wrangler	11/11/2009	4
14	Damon	Sasha	Coffee Wrangler	06/06/2010	4
15	Remedios	Mari	Coffee Wrangler	26/05/2012	4
16	Darren	Xu	Store Manager	25/05/2011	5
17	Quail	Octavia	Coffee Wrangler	09/12/2009	5
18	Ezeiel	Rachael	Coffee Wrangler	18/11/2009	5
19	Peter	Paloma	Coffee Wrangler	12/06/2011	5
20	Ronan	Magee	Coffee Wrangler	15/02/2007	5

The data in the preview has been truncated due to size limits.

OK Cancel

Query Settings

PROPERTIES Name Customer Lookup All Properties

APPLIED STEPS Source Promoted Headers X Changed Type

9 COLUMNS, 996+ ROWS - Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 14:18

The data preview dialog box will appear everything looks good click on OK.

This screenshot shows the Microsoft Power Query Editor interface. On the left, the 'Queries' pane lists several queries, with 'Employee Lookup' currently selected. The main area displays a preview of the 'Employee' table, showing columns like Employee ID, First Name, Last Name, Position, Start Date, and Location. A context menu is open over the 'Position' column, with options like 'Data Type', 'Merge Queries', 'Append Queries', 'Combine', and 'All Insights'. On the right, the 'Query Settings' pane shows properties for the selected query, including 'Name' (Employee Lookup) and 'Applied Steps' (which includes 'Promoted Headers' and 'Changed Type'). A blue arrow points from the text at the bottom to the 'OK' button in the data preview dialog box, which is overlaid on the main preview area. The dialog box contains settings for 'File Origin' (CSV file), 'Delimiter' (Comma), and 'Data Type Detection' (Based on first 200 rows). The preview itself shows a subset of the employee data.

store_id	baked_date	transaction_date	product_id	quantity_start_of_day	quantity_sold
3	25/12/2016	01/01/2017	69	18	3
3	26/12/2016	01/01/2017	69	18	4
3	26/12/2016	01/01/2017	70	18	2
3	26/12/2016	01/01/2017	70	18	2
3	21/12/2016	01/01/2017	71	18	2
3	21/12/2016	01/01/2017	71	18	5
3	21/12/2016	01/01/2017	71	18	2
3	25/12/2016	01/01/2017	72	48	8
3	25/12/2016	01/01/2017	72	48	3
3	21/12/2016	01/01/2017	73	18	2
3	20/12/2016	01/01/2017	74	18	1
3	23/12/2016	01/01/2017	74	18	2
3	21/12/2016	01/01/2017	75	18	4
3	21/12/2016	01/01/2017	75	18	6
3	25/12/2016	01/01/2017	76	18	2
3	28/12/2016	01/01/2017	76	18	3
3	21/12/2016	01/01/2017	76	18	3
3	28/12/2016	01/01/2017	77	48	3
3	28/12/2016	01/01/2017	77	48	3
3	28/12/2016	01/01/2017	78	18	3

The data preview dialog box will appear everything looks good click on OK.

File Home Transform Add Column View Tools Help

New Recent Enter Data Data source settings Manage Parameters Refresh Preview Manage Choose Columns Remove Columns Keep Rows Remove Rows Group By Data Type: Whole Number Use First Row as Headers Merge Queries Append Queries Combine Files Text Analytics Vision Azure Machine Learning Close New Query Data Sources Query Manage Columns Reduce Rows Sort Transform Combine All Insights

Queries (4)

Calendar Customer Lookup Employee Lookup Food Inventory

Food Inventory

File Origin Delimiter Data Type Detection

1353: Western European (Windows) Comma Based on first 200 rows

product_id	product_group	product_category	product_type	product	product_description
1	Whole Bean/Tear	Coffee beans	Organic Beans	Brasilian - Organic	It's like Carnival in a cup. Clean and smooth.
2	Whole Bean/Tear	Coffee beans	House Blend Beans	Our Old Time Diver Blend	Our packed blend of beans that is reminiscent of
3	Whole Bean/Tear	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.
4	Whole Bean/Tear	Coffee beans	Espresso Beans	Primo Espresso Roast	Our premium single source of hand roasted bean
5	Whole Bean/Tear	Coffee beans	Gourmet Beans	Columbian Medium Roast	A smooth cup of coffee any time of day.
6	Whole Bean/Tear	Coffee beans	Gourmet Beans	Ethiopia	From the home of coffee.
7	Whole Bean/Tear	Coffee beans	Premium Beans	Jamaican Coffee River	Ya man, it will start your day off right.
8	Whole Bean/Tear	Coffee beans	Premium Beans	Over Caff	The most expensive coffee in the world, the over
9	Whole Bean/Tear	Coffee beans	Organic Beans	Organic Decaf Blend	Our blend of hand picked organic beans that hav
10	Whole Bean/Tear	Coffee beans	Green Beans	Guatemalan Sustainably Grown	Green beans you can trust yourself.
11	Whole Bean/Tear	Loose Tea	Herbal tea	Lemon Grass	You will think you are Thailand as you sip your co
12	Whole Bean/Tear	Loose Tea	Herbal tea	Peppermint	Cool and refreshing to help calm your nerves.
13	Whole Bean/Tear	Loose Tea	Black tea	English Breakfast	The traditional cup to start your day.
14	Whole Bean/Tear	Loose Tea	Black tea	Earl Grey	A full leaf of Orange Pekoe blended with organic
15	Whole Bean/Tear	Loose Tea	Green tea	Serenity Green Tea	Mountain grown and harvested at the optimal ti
16	Whole Bean/Tear	Loose Tea	Chai tea	Traditional Blend Chai	A traditional blend.
17	Whole Bean/Tear	Loose Tea	Chai tea	Morning Sunrise Chai	Fair trade and organic and has a warm kick.
18	Whole Bean/Tear	Loose Tea	Chai tea	Spicy Eye Opener Chai	A spicier blend to awaken your taste buds.
19	Whole Bean/Tear	Packaged Chocolate	Dringing Chocolate	Dark chocolate	This drinking chocolate is smooth and creamy.
20	Whole Bean/Tear	Packaged Chocolate	Organic Chocolate	Guatemala Grown, Organic	Certified organic, containing the highest quality raw
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					

OK Cancel

6 COLUMNS, 999+ ROWS / Column profiling based on top 1000 rows

PREVIEW DOWN/DATED AT 14/25

The screenshot shows the Power BI Query Editor interface. On the left, there's a sidebar with various data source options like Calendar, Customer Lookup, Employee Lookup, Food Inventory, and Product Lookup. The main area displays a query with multiple columns: product_id, product_group, product_category, product_type, product, product_description, unit_of_measure, and current_cost. A context menu is open over the 'product' column, showing options like 'Choose Column', 'Remove Column', 'Group By', 'Replace Values', 'Transform', and 'Combine'. Below the table, a 'Data Type Detection' dialog box is displayed, containing tabs for 'File Origin' (set to '1252 - Western European (Windows)'), 'Delimiter' (set to 'Comma'), and 'Data Type Detection' (set to 'Based on first 200 rows'). The dialog box also shows a preview of the transaction data with columns: transaction_id, transaction_date, transaction_time, store_id, staff_id, customer_id, store_yn, order, line_item_id, and product. A blue arrow points from the text at the bottom of the image to the 'OK' button in the dialog box. Another blue arrow points from the text to the 'OK' button.

The data preview dialog box will appear everything looks good click on OK.

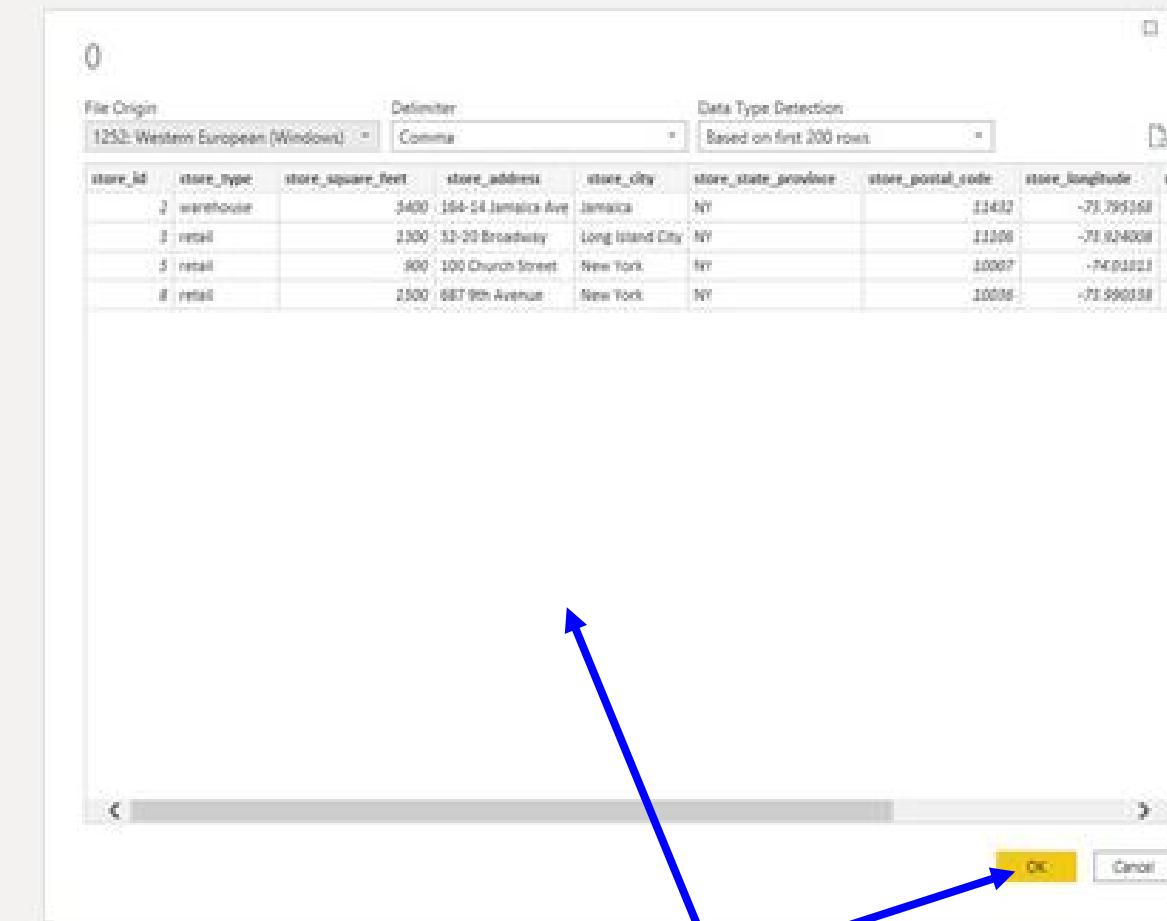


Queries [6]

- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store



These are the text files imported into Power BI from our imports.



The data preview dialog box will appear everything looks good click on OK.

Power BI Dummy Project - Power Query Editor

File Home Transform Add Column View Tools Help

Close & Apply New Source Recent Data source settings Manage Parameters Refresh Preview Advanced Editor Choose Columns Remove Columns Keep Rows Remove Rows Group By Split Columns Replace Values Data Type: Whole Number Use First Row as Headers Merge Queries Append Queries Combine Files Text Analytics Vision Azure Machine Learning Close New Query Queries: Store Lookups

Store Lookups

Click here on Close & Apply.

11 COLUMNS, 4 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 14:32

P1_store_id	P1_store_name	P1_store_square_feet	P1_store_address	P1_store_city	P1_store_state_province	P1_store_postal_code	L1_store_longitude	L1_store_latitude
1	warehouse	3400	184-14 Jamaica Ave	Jamaica	NY	11482	-73.793288	40.7052
2	retail	2100	32-22 Broadway	Long Island City	NY	11206	-73.924058	40.7611
3	retail	800	100 Church Street	New York	NY	10007	-74.05023	40.7113
4	retail	1500	687 8th Avenue	New York	NY	10016	-73.990338	40.7614

Query Settings PROPERTIES Name: Store Lookups All Properties APPLIED STEPS Source Promoted Headers P1_Changed Type

Power BI Dummy Project - Power BI Desktop

File Home Insert Modeling View Help

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform data - New visual Text box More visuals Publish

Power BI Dummy Project - Power BI Desktop

There are pending changes in your queries that haven't been applied.

Apply changes Discard changes

Visualizations Fields

Search

Filters on this page Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Click on Save here once changes/Updates are complete.

Apply query changes

- Calendar Creating connection in model..
- Customer lookup Creating connection in model..
- Employee lookup Creating connection in model..
- Food inventory Creating connection in model..
- Product lookup Creating connection in model..

The changes will be applied & everything will be updated.

Page 1 +

Power BI Dummy Project - Power BI Desktop

Save Home Insert Modeling View Help

File & Save Data Out data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Queries Calculations Share

Model

Click here to navigate over to the Model view to start wiring up the data model.

Filters Visualizations Fields

Search

Add data fields here

Filters on all pages Add data fields here

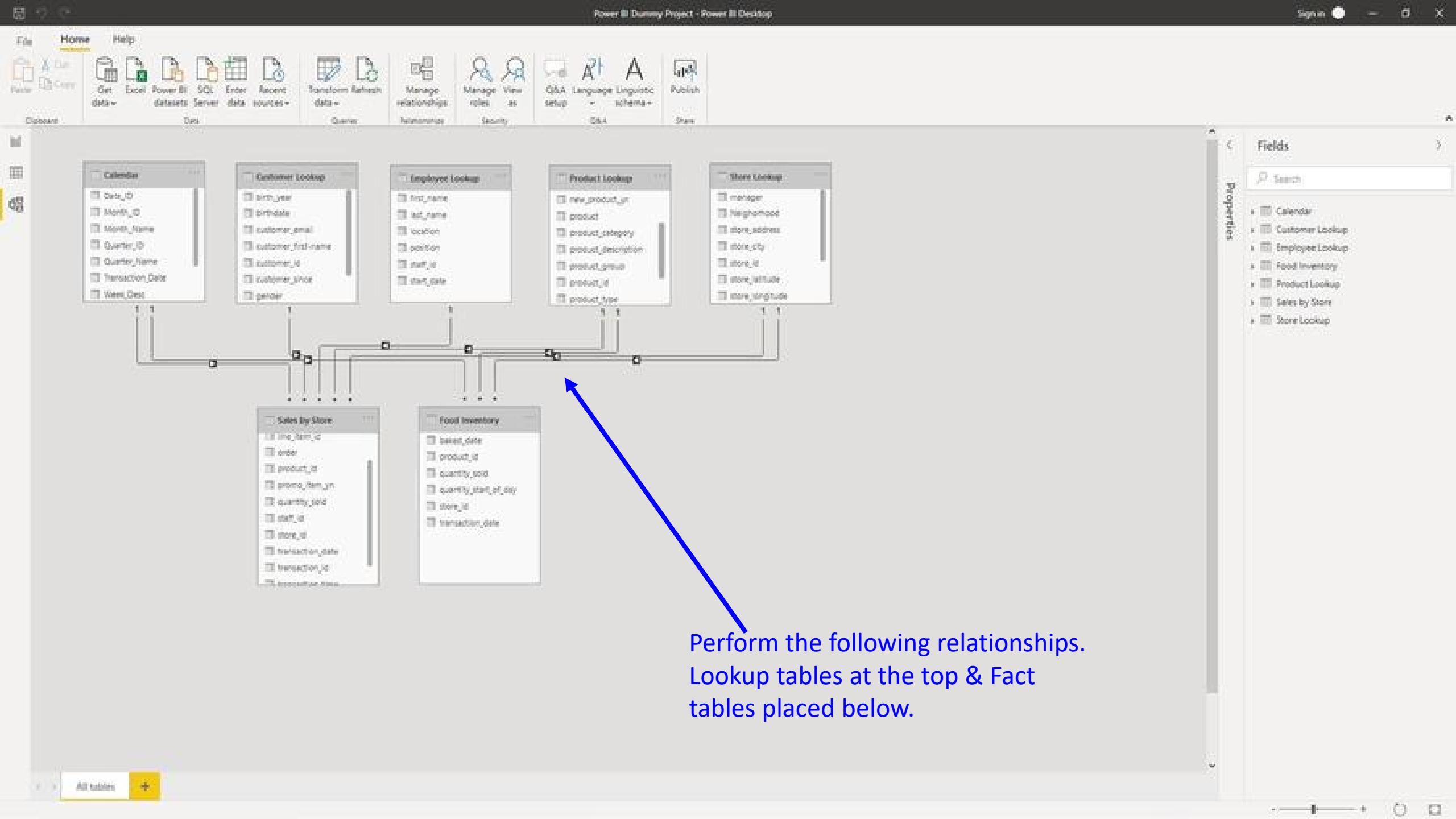
Values Drill through

Off On

Add drill-through fields here

Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup

Page 1



2) The DAX Engines

Introduction - The DAX Engines



DAX is powered by two internal engines (**formula engine & storage engine**) which work together to compress & encode raw data and evaluate DAX queries

- Although these engines operate entirely behind the scenes, knowing how they work will help you build advanced skills by understanding exactly how DAX *thinks*

TOPICS WE'LL REVIEW:

Internal DAX Engines

Query Evaluation

Data & Storage Types

Columnar Structures

Compression & Encoding

VertiPaq Relationships

DAX Formula & Storage Engines

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

There are two distinct engines that work together to process every DAX query:
the **Formula Engine** and the **Storage Engine**

FORMULA ENGINE

- Receives, interprets and executes all DAX requests
- Processes the DAX query then generates a list of logical steps called a **query plan**
- Works with the **datacache** sent back from the storage engine to evaluate the DAX query and return a result

STORAGE ENGINE

- Compresses and encodes raw data, and only communicates with the formula engine (*doesn't understand the DAX language*)
- Receives a query plan from Formula Engine, executes it, and returns a **datacache**
- **NOTE:** There are two types of storage engines, based on the type of connection you're using:
 1. **VertiPaq** is used for data stored in-memory (*connected to Power BI via import mode*)
 2. **DirectQuery** is used for data read directly from the source (*i.e. Azure, PostgreSQL, SAP*)

DAX Query Evaluation In Depth

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

DAX QUERY:

```
1 Sales to Females =  
2 CALCULATE(  
3     [Customer Sales],  
4     FILTER(  
5         'Customer Lookup',  
6         'Customer Lookup'[gender] = "F"  
7     ))
```

$f(x)$

DAX query is sent to
the Formula Engine



Storage
Engine



Storage engine executes the
plan and sends a **datacache**
back to the formula engine



Formula
Engine



Formula engine
interprets the query
and sends a **query plan**
to the storage engine



Formula
Engine



Formula engine runs the DAX
query against the datacache,
and evaluates the result

MEASURE OUTPUT:

Sales to Females:
\$912,184

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

DAX QUERY:

```
1 Sales to Females =  
2 CALCULATE(  
3     [Total Sales],  
4     FILTER(  
5         Customer,  
6         Customer[gender] = "F"  
7     )  
8 )
```

*DAX here! Aaron would like
me to evaluate this query*

MEASURE OUTPUT:

Sales to Females:

\$912,184



*Sure! Here's the data you'll
need to evaluate that query*



*Got it. Hey storage
engine, can you grab
some data for me?*



*Great! Hey Aaron, the result
for that measure is \$912,184!*

Data & Storage Types

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

DAX uses **6 data types** to store values:

DAX Data Type	Power BI Data Type	Storage Type	Example
Integer	Whole Number	64-bit Value	<i>Max: 9,223,372,036,854,775,807</i>
Decimal	Decimal Number	Double-precision floating-point value	<i>64-bit precision</i>
Currency	Fixed Decimal Number	Fixed Decimal Number (Stored as Integer)	<i>317.9899</i>
DateTime	DateTime, Date, Time	Floating-point number	<i>1/1/2020 12:00p = 43830.50</i>
Boolean	True/False	True/False	<i>True/False</i>
String	Unicode String	16-bit characters	<i>"Maven Analytics" = "MAVEN ANALYTICS"</i>



HEY THIS IS IMPORTANT!

Data Types represent how values are *stored* by the DAX storage engine
Formatting represents how values *appear* to end users (%, date, \$, etc.)

VertiPaq Columnar Data Structure

Internal DAX Engines

Query Evaluation

Data & Storage Types

Columnar Structures

Compression & Encoding

VertiPaq Relationships

VertiPaq uses a **columnar data structure**, which stores data as individual columns (*rather than rows or full tables*) to quickly and efficiently evaluate DAX queries

product_id	product_group	product_category	wholesale_price
9	Whole Bean/Teas	Coffee beans	\$18.00
17	Whole Bean/Teas	Loose Tea	\$7.60
21	Whole Bean/Teas	Packaged Chocolate	\$10.66
57	Beverages	Tea	\$0.78
61	Beverages	Drinking Chocolate	\$3.56
65	Add-ons	Flavours	\$0.04
79	Food	Bakery	\$2.44
83	Merchandise	Branded	\$4.48
88	Beverages	Coffee	\$0.42



VertiPaq Compression & Encoding

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

The goal of **compression** and **encoding** is to reduce the amount of memory needed to evaluate a DAX query.

Based on a sample of data, one (or more) of the following methods will be used:

1. Value Encoding

- Mathematical process used to reduce the number of bits needed to store **integer** values

2. Hash Encoding (*aka Dictionary Encoding*)

- Identifies the distinct **string** values and creates a new table with indexes

3. Run Length Encoding (RLE)

- Reduces the size of a dataset by identifying repeated values found in adjacent rows



HEY THIS IS IMPORTANT!

The actual storage algorithms are proprietary, so not all details are available ☺

Value Encoding

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

Validating
Relationships

Value Encoding uses a mathematical process to determine relationships between the values in a column, and convert them into smaller values for storage

- Value encoding only works for integer values (*including currency*), and cannot be applied to strings or floating-point values

City ID
10014
10106
10215
10007
10002
10021
10036

City ID
14
106
215
7
2
21
36

- *Max = 10215*
- *14 bits needed*
- *Max = 215*
- *9 bits needed (36% reduction!)*

Hash Encoding

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPax
Relationships

Hash Encoding builds a “dictionary” of distinct items in a column, assigns a unique integer value (*index*) to each item, and stores the data using the index values rather than the full text strings

- With hash encoding, storage requirements are defined by the number of unique items in the column (*cardinality*), **NOT** by the length of the string values themselves

Product	Index	Row	Product	Index
Coffee Beans	0	1	Coffee Beans	0
Tea	1	2	Tea	1
Bakery	2	3	Bakery	2
Bakery	2	4	Encoding dictionary	
Coffee Beans	0	5		
Coffee Beans	0	6		
Bakery	2	7		
Tea	1	8		

Run Length Encoding

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

Run Length Encoding (RLE) reduces the size of a column by replacing duplicate rows with a table containing each distinct value and the count of instances

- NOTE: RLE only works when the same value is repeated in consecutive rows, but the VertiPaq engine automatically sorts data on import and refresh to find the optimal compression*

This is what you see

City Zip
59716
59716
59716
02215
02215
05672
05672
05672
05672

3 instances

2 instances

4 instances

This is how the data is stored

City Zip	Count
59716	3
02215	2
05672	4

One row per distinct value

Internal DBX
Engines

Query Evaluation

Data & Storage
Types

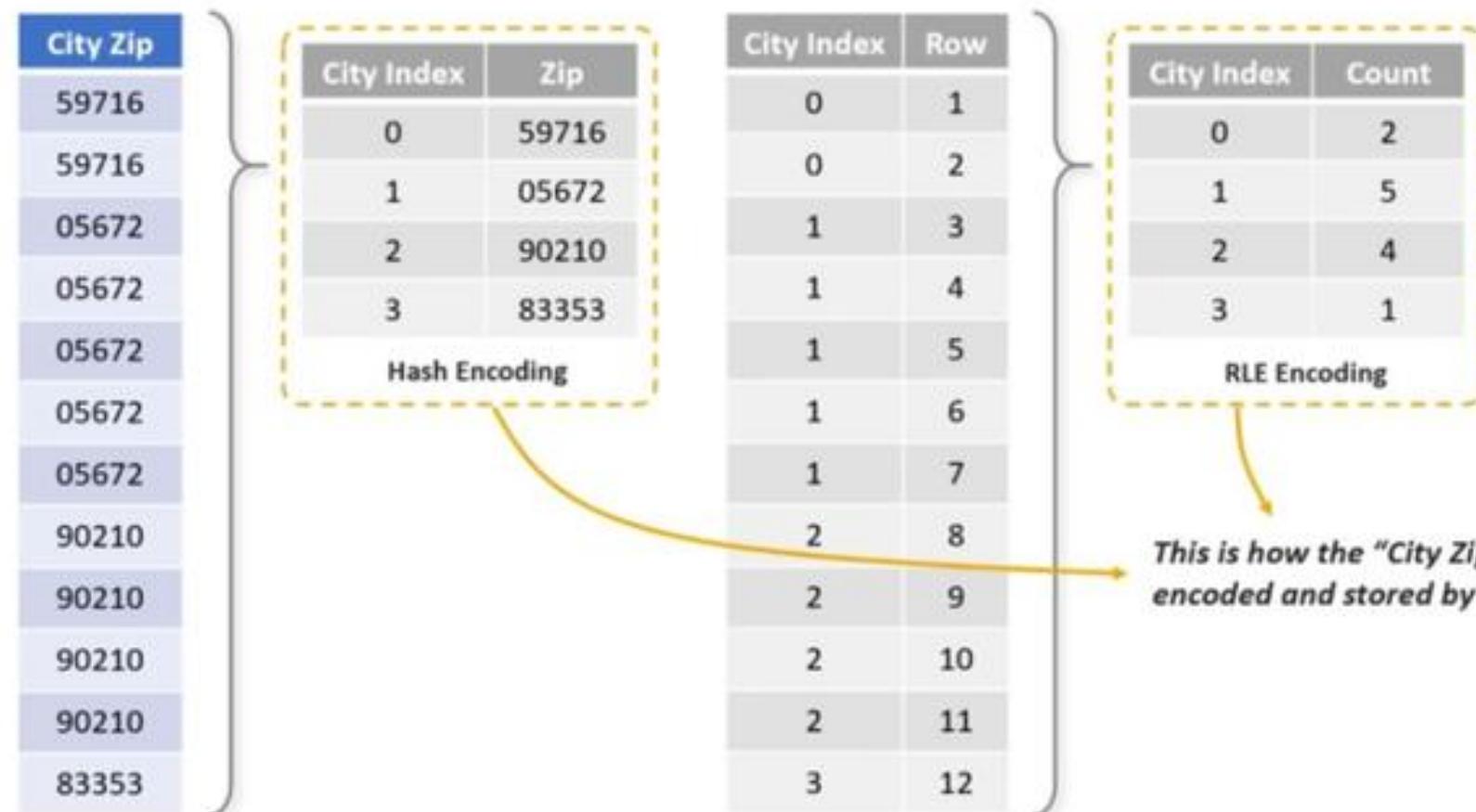
Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

Columns can have both **Run Length** and either **Hash (Dictionary)** or **Value** encoding

- Compression type is determined by cardinality, number of repeat values, row count, and data type



This is how the "City Zip" data is encoded and stored by VertiPaq

VertiPaq Relationships

Internal DAX
Engines

Query Evaluation

Data & Storage
Types

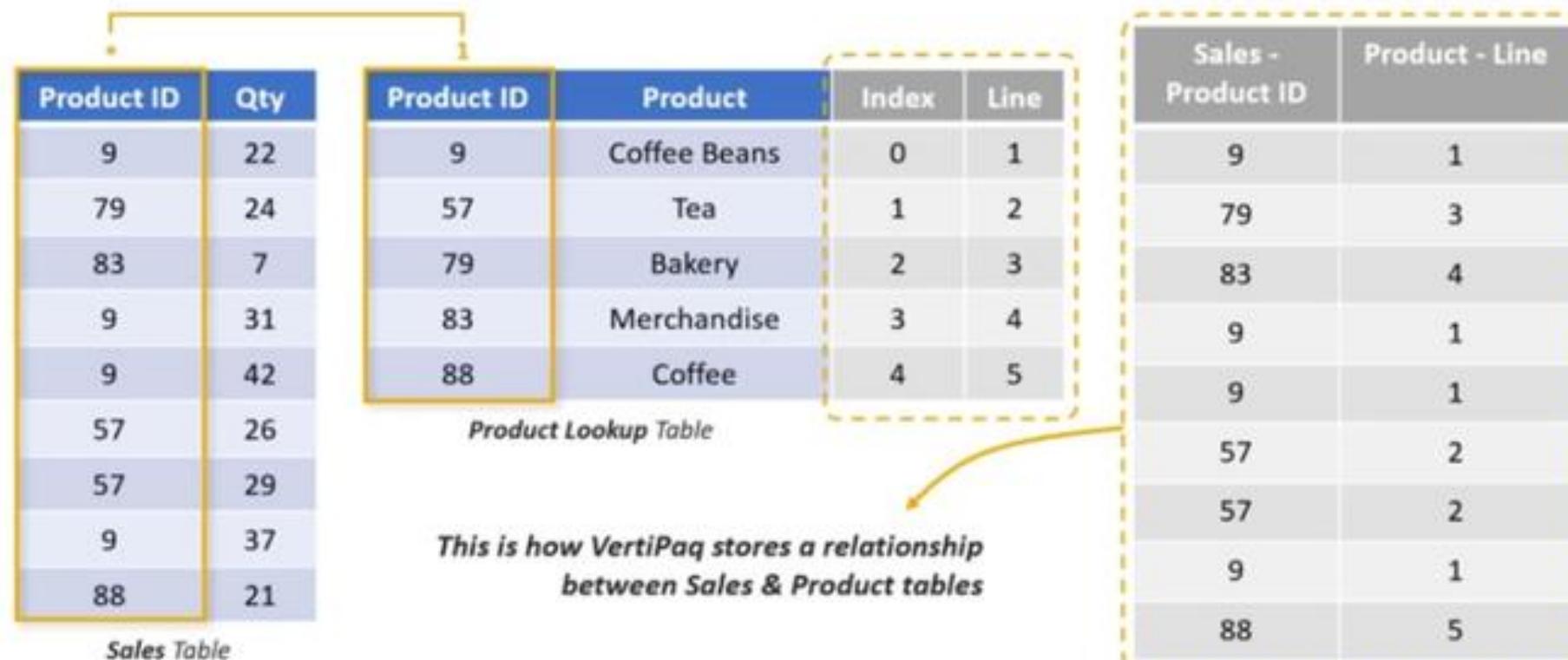
Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

VertiPaq has a special way of **mapping relationships** between columns in your data model, which allows it to evaluate complex, multi-column queries

- *NOTE: This is NOT the same as creating table relationships in your data model; this is essentially a blueprint that VertiPaq uses to map pairs of primary & foreign keys across related tables*



Internal DAX
Engines

Query Evaluation

Data & Storage
Types

Columnar
Structures

Compression &
Encoding

VertiPaq
Relationships

```
1 Coffee Bean Sales =  
2 CALCULATE(  
3     [Quantity Sold],  
4     FILTER(  
5         'Product Lookup',  
6         'Product Lookup'[product] = "Coffee Beans"  
7     )
```

STEP 1: Search for “Coffee Beans” in Product dictionary to find line number from product lookup

Product	Index	Line
Coffee Beans	0	1
Tea	1	2
Bakery	2	3
Merchandise	3	4
Coffee	4	5

STEP 2: Use **relationship** to find all rows where product line = 1

Sales - Product ID	Product - Line
9	1
79	3
83	4
9	1
9	1
57	2
57	2
9	1
88	5

STEP 3: Return a **datacache** containing a filtered sales table and sends to formula engine

Product ID	Qty
9	22
9	31
9	42
9	37

STEP 4: Formula engine evaluates the **[Quantity Sold]** measure against datacache

[Quantity Sold] = **132**

Summary: DAX Engines



DAX uses two engines: the **Formula Engine & Storage Engine**

- *The Formula Engine interprets & evaluates DAX, and the Storage Engine compresses & encodes data*



There are two types of storage engines: **VertiPaq & DirectQuery**

- *VertiPaq is used for in-memory storage and DirectQuery is used for direct connections to external sources; both create **datacaches** and send them to the formula engine for DAX query evaluation*



VertiPaq stores data using a **columnar database setup**

- *Data is stored in individual columns that can be accessed quickly, but queries that call multiple columns may require more complex logic to produce a datacache*



Raw data is **compressed & encoded** to optimize processing

- *Data can be compressed using **Value**, **Hash** (Dictionary), or **Run Length encoding (RLE)**, based on cardinality, repeat values, row count, and data type*

3) Tips & Best Practices

Introduction - Tips & Best Practices



In this section we'll cover some common techniques and best practices for working efficiently with DAX, including **shortcuts**, **comments**, **measure tables**, and **variables**

TOPICS WE'LL COVER:

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

COMMON USE CASES:

- *Saving time with keyboard shortcuts*
- *Adding comments to help others understand exactly what each line of your DAX code is doing*
- *Using error checking functions for testing & QA*
- *Adding variables to troubleshoot code, enhance readability and improve DAX performance*

DAX Shortcuts

Shortcuts

Formatting

Evaluation Order

Comments

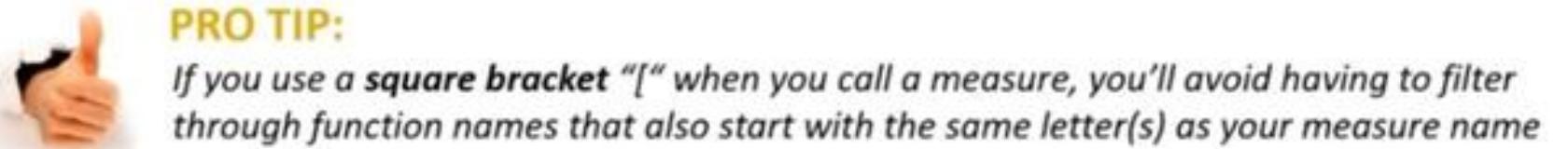
Measure Tables

Error Handling

Variables

Action	Keyboard Shortcut
Insert Line Below	Shift + Enter
Insert Line Above	Control + Shift + Enter
Select Current Line	Control + I
Cut Current Line	Control + X
Paste Cut Line	Control + V
Duplicate Current Row	Shift + Alt + ↓↑
Go to Line	Control + G
Enlarge Code Type	Control + Mouse Scroll

PRO TIP:



Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools

Name: Sales by Store

Mark as date table = Counters Manage relationships New measure Quick New measure column New table

Revenue Calculations

Create a new measure to calculate customer sales using the Sales by Store table i.e. quantity_sold multiply by unit_price

Filters Visualizations Fields

Add data fields here

Add data fields here

New measure

New New measure New quick measure Refresh data Edit query Incremental refresh Manage aggregations Rename Delete Hide Mark as date table View hidden Unhide all Collapse all Expand all

Page 1 +

Maven Router - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: General Data category: Uncategorized

Home table: Sales by Store \$ - % . Auto New measure Quick measure measure

Structure Formatting Properties Calculations

Customer Sales =
sum('Sales by Store') + 'Sales by Store'[unit_price]

Visualizations Fields

Filters on this page Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Off —

Keep all filters On —

Add drill-through fields here

Customer Sales

customer_id

instock_yn

line_item_id

order

product_id

promo_id

quantity_sold

staff_id

store_id

transaction_id

transaction_type

unit_price

Store Lookup

Page 1 +

Typing shift + Enter will bring you to line 2 & 3.

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: General Data category: Uncategorized

Home table: Sales by Store \$ - % , . Auto

New Quick measure measure Calculations

1 Customer Sales =
2 SUMX('Sales by Store',
3 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price])

Visualizations Fields

Add data fields here

Filters on all pages Add data fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Customer Sales

- customer_id
- instance_id
- line_item_id
- order
- product_id
- promo_id
- quantity_sold
- staff_id
- store_id
- transaction_id
- transaction_qty
- unit_price

Store Lookup

Page 1 +

By hitting Control & moving the scroller in your mouse you will be able to increase/decrease the size of the formulae bar. So that the code is a little bit easier to see & read.

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: General Data category: Uncategorized

Home table: Sales by Store \$ - % # Auto New Quick Measure measure Calculations

Structure Formatting Properties Calculations

Visualizations > Fields >

Customer Sales =
SUMX('Sales by Store',
'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price])

Add data fields here

Filters on all pages

Add data fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Customer Sales

- customer_id
- employee_id
- line_item_id
- order
- product_id
- promo_id
- quantity_sold
- staff_id
- store_id
- transaction_id
- transaction_qty
- unit_price

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

Page 1 +

By hitting Control & I will select the current row.

By hitting Control & X will cut the row.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: Whole number Data category: Uncategorized

Home table: Sales by Store \$ - % .0

New Quick measure measure Calculations Properties

Structure Formatting Calculations Properties

Visualizations Fields

Customer Sales

To use special characters in a measure name, enclose the entire name in brackets ([]) and add a [] to any closing brackets in the name.

1
2 SUMX('Sales by Store',
3 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price])

Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Off On

Keep all filters

Add drill-through fields here

Customer Sales

customer_id

customer_yn

line_item_id

order

product_id

promo_id

quantity_sold

staff_id

store_id

transaction_id

transaction_qty

transaction_type

unit_price

Store Lookup

By hitting Control + Shift + Enter to re add a line above.

Maven Router - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: Whole number Data category: Uncategorized

Home table: Sales by Store \$ - % .0

New measure Quick measure measure

Structure Formatting Properties Calculations

1 Customer Sales =
2 SUMX('Sales by Store',
3 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price])

Visualizations Fields

Add data fields here

Filters on all pages Add data fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Customer Sales

customer_id

isactive_yn

line_item_id

order

product_id

promo_id

quantity_sold

staff_id

store_id

transaction_id

transaction_type

unit_price

Store Lookup

By hitting Control + V to paste the first line of code back in.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: Whole number Data category: Uncategorized

Home table: Sales by Store \$ - % , . 0

New Quick measure measure

Structure Symmetry Properties Calculations

1 Customer Sales =
2 Customer Sales =
3 SUMX('Sales by Store',
4 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price])

Visualizations > Fields >

Search: Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Customer 1... customer_id instance_id line_item_id order product_id promo_id quantity_sold staff_id store_id transaction_... transaction_... transaction_... unit_price

By hitting ALT + Shift up or down (depending on where you will want to duplicate the line) you will be able to duplicate line of code.

Formatting Best Practices

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Formatting is **KEY** for readability. Can you tell what this code does in **10 seconds**?

Time's Up!

```
1 Sales for Selected Store = VAR Store = SELECTEDVALUE('Store Lookup'  
[store_id]) RETURN CALCULATE(SUM('Sales by Store'[Quantity_Sold]) *  
1.05,FILTER('Sales by Store','Sales by Store'[Store_ID] = Store ))
```

```
1 Sales for Selected Store =  
2 VAR Store =  
3 SELECTEDVALUE('Store Lookup'[store_id]) -- Define column for single value  
4  
5 RETURN  
6 CALCULATE(  
7     SUM(  
8         'Sales by Store'[Quantity_Sold]) * 1.05,  
9         FILTER(  
10            'Sales by Store',  
11            'Sales by Store'[Store_ID] = Store -- where selected store has one value  
12        )  
13    )
```



PRO TIP

 Use **shift + enter** to split out and indent each component of your DAX formulas to make them more human readable (TIP: try using daxformatter.com to quickly format your code)

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales Format: Whole number Data category: Uncategorized

Home table: Sales by Store \$ - % , . 0 New measure Quick measure measure

Structure Formatting Properties Calculations

Customer Sales =
SUMX(
'Sales by Store',
'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price])
)

Visualizations > Fields >

Add data fields here

Filters on all pages

Add drill-through fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add data fields here

Customer Sales

customer_id
instore_qty
line_item_id
order
product_id
promo_qty
quantity_sold
staff_id
store_id
transaction_id
transaction_qty
transaction_qty
unit_price

Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
Store Lookup

Page 1 +

Using best practices the above Customer Sales measure has been cleaned up for better readability of code.

Maven Roaders - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Customer Sales Format Whole number Data category Uncategorized

Home table Sales by Store \$ - % , . 0

New measure measure

Write a DAX expression that calculates a value from your data

1 Customer Sales =
2 SUMX(
3 'Sales by Store',
4 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
5)

Click here to create a new measure.

Visualizations Fields

Search

Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store

Customer_Lookup_id customer_id instance_id line_item_id order_id product_id promo_id quantity_sold staff_id store_id transaction_id transaction_line_id unit_price

Add data fields here

Filters on all pages

Add drill-through fields here

Drill through

Cross-report

Off On Keep all filters

On

Page 1

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table spots Measure tools

Name: Cost Format: \$ - % Data category: Uncategorized

Home table: Sales by Store Calculations

Structure Formatting Properties

1 Cost =
2 SUMX(
3 'Sales by Store',
4 'Sales by Store'[quantity_sold] *
5 RELATED(
6 'Product Lookup'[current_cost])
7)
8)

Multiplying values between 2 fields in different tables. RELATED essentially means that there is a relationship between Sales by Store & Product Lookup for the calculation to be performed.

Visualizations Fields

Search: Sales by Store

Values: Cost, Customer S..., customer_id, instance_id, line_item_id, order, product_id, promo_id, quantity_sold, staff_id, store_id, transaction_id, transaction_qty, unit_price

Add drill-through fields here

Drill through: Off

Cross-report: Off

Keep all filters: On

Add drill-through fields here

Page 1

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Measure Format: \$ - % Date category: Uncategorized

Home table: Calendar \$ - % Date category: Uncategorized

Structure Formatting Calculations

New Current measure Connections

1 Profit =
2 []

[Cost]
[Customer Sales]

Visualizations > Fields >

Search

Calendar

Measure

Month_ID

Month_Name

Quarter_ID

Quarter_Name

Transaction_ID

Week_Desc

Week_ID

Week_Year

Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Cost

Customer_L...

customer_id

instore_yn

line_item_id

order

product_id

promo_code

quantity_sold

staff_id

store_id

transaction...

transaction...

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off

On

Keep all filters

Add drill-through fields here

Page 1 +

Typing the square bracket will display the measures we created earlier i.e. Cost & Customer Sales.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Measure Format: \$ - % Date category: Uncategorized

Home table: Calendar \$ - % Date category: Uncategorized

New measure Quick measure measure

Structure Formatting Properties Calculations

1 Profit =
2 [Customer Sales] - [Cost]

Click Enter to add the Profit measure.

Visualizations Fields

Add data fields here

Add data fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Cost Customer_L... customer_id instore_y... line_item_id order product_id promo_ty... quantity_sold staff_id store_id transaction_... transaction_ty...

Filters on this page

Filters on all pages

Page 1 +

Page 1 of 1

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Calculations Publish

Matrix option.

1) Drag the measure created to the report.

2) Change the visualisation to a Matrix so that all measure calculations can be displayed via values.

Filters Visualizations Fields

Search

Cost is (All)

Customer Sales is (All)

Profit is (All)

Add data fields here

Rows

Columns

Values

Cost Customer Sales Profit

Filters on this page

Filters on all pages

Drill-through

ON OFF

Keep all filters

On Off

Add drill-through here

Customer Lookup Employee Lookup Food Inventory ProductLookup Sales by Store

Cost Customer Sales Profit

customer_id customer_yn line_item_id order product_id

promo_item_yn quantity_sold staff_id store_id transaction_date transaction_id

Page 1

DAX Evaluation Order

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Evaluation order is the process by which DAX evaluates the parameters in a function

- **Individual functions** typically evaluate from **left-to-right**, starting with the first parameter (*followed by the second, third, etc.*)
- **Nested functions** evaluate from the **inside-out**, starting with the innermost function and working outward from there

Non-nested:

1 **2** **3**
`=IF(LogicalTest, ResultIfTrue, [ResultIfFalse])`

Nested:

`=SUMX(
 FILTER(
 FILTER('Table',
 RELATED('Table'[Column]), 1
 RELATED('Table'[Column]), 2
 'Table'[Column])) 3`

NOTE: The **CALCULATE** function evaluates using its own unique set of rules (*more on this later!*)

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Non-Nested

```
1 Customer Sales =  
2 SUMX(  
3     'Sales by Store',  
4     'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price] -- 1. DAX returns the Sales by Store table  
5 ) -- 2. and then evaluates the expression
```

Nested

```
1 Store 3 Sales of Whole Bean/Teas (SUMX) =  
2 SUMX(  
3     FILTER(  
4         FILTER(  
5             'Sales by Store',  
6             RELATED(  
7                 'Store Lookup'[sales_outlet_id]) = 3 -- 1  
8             ),  
9             RELATED(  
10                'Product Lookup'[product_group]) = "Whole Bean/Teas" -- 2  
11             ),  
12             'Sales by Store'[Quantity_Sold] * 'Sales by Store'[Unit_Price] -- 3  
13 )
```

Commenting Your Code

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Comments can help other users interpret your code, and can be particularly helpful for complex queries with multiple lines, nested functions, etc.

Comment Type	Marker
Single Line Comment	-- or //
Multi Line Comment	/* ... */

```
1 Sales based on Selected Store =
2 VAR Store =
3 SELECTEDVALUE('Store Lookup'[sales_outlet_id]) -- This is an example of a single line comment
4
5 RETURN
6 CALCULATE(
7     SUM('Sales by Store'[quantity]),
8     FILTER('Sales by Store', 'Sales by Store'[sales_outlet_id] = Store) /*Or, you can explain how you're using
9     filter to return a table where the store id is based on a selected value with a multiline comment */
10 )
```

PRO TIP

 Avoid putting comments **at the end of your DAX query (below the last closing parenthesis)**, as they can be missed or omitted by users and formatting tools

Maven Robotics - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Cost Format: General Data category: Uncategorized

Home table: Sales by Store \$ - % # . Auto

Structure Formatting Properties Calculations

Visualizations Fields

Search

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Day
- Week_ID
- Week_Year
- Year_ID

Rows

Add data fields here

Columns

Add data fields here

Values

- Cost
- Customer_Sales
- Profit

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Cost

Customer_Sales

customer_id

instance_yr

line_item_id

order

product_id

Profit

promo_item_id

quantity_sold

staff_id

store_id

transaction_date

transaction_id

1 Cost =
2 SUMX(
3 'Sales by Store',
4 'Sales by Store'[quantity_sold] *
5 RELATED(
6 'Product Lookup'[current_cost]
7)
8)

--Returns the Sales by Store table to
--evaluate expression quantity sold times cost from a related table

Example of how comments can be added to the Cost measure.

Page 1 +

PRO TIP: Dedicated Measure Table

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

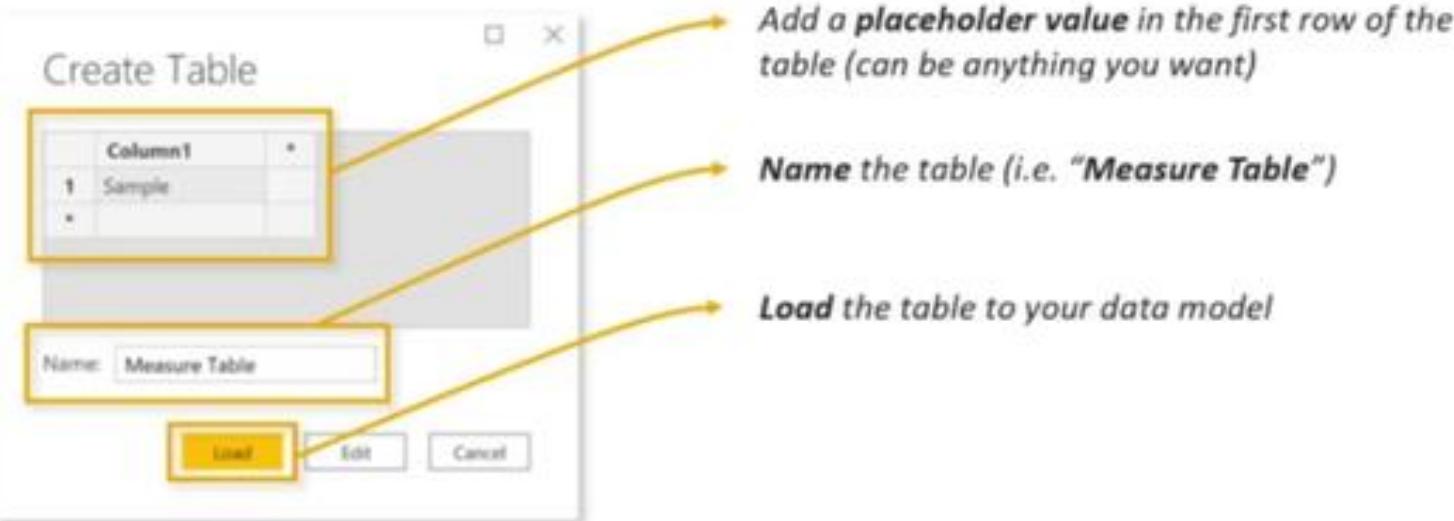
Error Handling

Variables

Creating a **separate table** to contain your DAX measures is a great way to stay organized (*you can even group your measures into folders within the table!*)



Enter Data to create a new table



Add a placeholder value in the first row of the table (can be anything you want)

Name the table (i.e. "Measure Table")

Load the table to your data model

Maven Readers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Get data Excel Power BI datasets SQL Server Enter current data source Transform Refresh data New visual Text box More visuals New Quick measure Publish

Insert Calculations Share

Create a new table by typing or pasting in new content.

Cost Customer Sales Profit
1,699,176.27 4232705 3,168,328.61

Click on Enter data.

Filters Visualizations Fields

Search

Add data fields here

Filters on all pages Add data fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

- Cost
- Customer Sales
- customer_id
- instore_yn
- line_item_id
- order
- product_id
- Profit
- promo_item_yn
- quantity_sold
- staff_id
- store_id
- transaction_data
- transaction_id

Page 1 +

Page 1 of 1

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Enter data Refresh Recent sources Transform data New visual Text box More visuals Publish

Dataset Data Queries Calculations Share

Create Table

Column1

1 Sample

Name: Measure Table

Load Edit Cancel

This dialog box will appear fill in the appropriate information & click on load.

Visualizations Fields

Search

here here

Values Add data fields here

Drill through

Cross-report

Off On Keep all filters

Add drill-through fields here

Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store

Cost Customer Sales customer_id instance_id line_item_id order product_id Profit promo_item_id quantity_id staff_id store_id transaction_date transaction_id

Maven Rooters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools

Name Measure Table

Mark as data table Manage relationships New measure Quick New measure column New table

Measure Calculators Relationships Calculations

Cost Customer Sales Profit

1,089,176.27 423,270.5 3,163,526.81

A Measure Table is created here where it has “Column1”

Filters Visualizations Fields

Search

Add data fields here

Filters on all pages Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Search

Calendar Customer Lookup Employee Lookup Food Inventory Measure Table Column1 Product Lookup Sales by Store Store Lookup

Page 1 +

Maven Roadmap - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Cost Format: General Data category: Uncategorized

New Quick measure measure Calculations

Home table: Sales by Store Calendar Formatting Properties Calculations

Customer Lookup Employee Lookup Food Inventory Measure Table Product Lookup Sales by Store Store Lookup

7)
8)

sales by Store', --Returns the Sales by Store table to
Product Lookup' [quantity_sold] * --evaluate expression quantity sold times cost from a related table
ATED(
'Product Lookup' [current_cost]

The Cost measure can be moved to the Measure Table by amending the Home Table to Measure Table.

Visualizations Fields

P search

Calendar Customer Lookup Employee Lookup Food Inventory Measure Table Product Lookup Sales by Store Cost Customer Sales customer_id instore_yn line_item_id order product_id Profit promo_item_yn quantity_sold staff_id store_id transaction_date transaction_id transaction_time unit_price Store Lookup

Page 1 +

Maven Roadmap - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Customer Sales

Home table: Sales by Store

Format: Whole number

Data category: Uncategorized

New Quick measure measure Calculations Properties

Customer Sales =

```
    Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
```

Visualizations Fields

Search:

- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Measure Table
- Product Lookup
- Sales by Store
- Store Lookup

Add data fields here

Filters on all pages

Add data fields here

Values

Drill through

Cross-report

Off

On

Keep all filters

Add drill-through fields here

Customer Sales

- customer_id
- store_id
- line_item_id
- order
- product_id
- Profit
- promo_item_id
- quantity_sold
- staff_id
- store_id
- transaction_date
- transaction_id
- transaction_time
- unit_price

Store Lookup

The Customer Sales measure can be moved to the Measure Table by amending the Home Table to Measure Table.

Page 1 +

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Profit

Home table Sales by Store

Format General

Data category Uncategorized

New measure measure

Quick measure measure

Calendar

Formatting Properties Calculations

Sales by Store [Cost] - [Sales]

Customer Sales: Profit

Measure Table

Product Lookup

Sales by Store

Store Lookup

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Visualizations

Fields

P Search

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

Column1

Cost

Customer Sales

Product Lookup

Sales by Store

customer_id

instance_idn

line_item_id

order

product_id

Profit

promo_item_idn

quantity_id

staff_id

store_id

transaction_date

transaction_id

transaction_time

unit_price

Store Lookup

The Profit measure can be moved to the Measure Table by amending the Home Table to Measure Table.

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Get data from Power BI datasets Server Enter data Refresh sources Transform data New visual Text box More visuals New measure, measure Quick Publish

Cost Customer Sales Profit

Cost	Customer Sales	Profit
1,009,179.27	423,270.0	1,163,528.61

Filters Visualizations Fields

Search

Add data fields here

Filters on all pages Add data fields here

Values

Column1

Check

New hierarchy

New measure

New column

New quick measure

Rename

Delete

Hide

Delete

View hidden

Unhide all

Collapse all

Expand all

New group

Add to filters

Add to drill through

Within the Measure Table Column1 can be deleted as its not doing anything.

Page 1

Page 1 of 1

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get Data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Depend Data Queries Calculations Share

Cost Customer Sales Profit
1,089,176.27 423,270.5 3,143,526.61

Delete column
Are you sure you want to delete 'Column1'?
Delete Cancel

Filters Visualizations Fields

Search Search

Add data fields here

Add data fields here

Values Add data fields here

Drill through
Cross-report
Off On
Keep all filters
On

Add drill-through fields here

Calendar Customer Lookup Employee Lookup Food Inventory Measure Table Column1 Cost Customer Sales Profit Product Lookup Sales by Store Store Lookup

Click on Delete.

The below measures are now moved here from previous steps performed.

- a) Cost
- b) Customer Sales
- c) Profit

Page 1 +

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help

Get data - Excel Power BI datasets SQL Server Enter data Recent sources - Transform Refresh data Manage relationships Relationships Security Manage roles View Q&A Language schema Q&A Publish Share

Properties Fields

Search

General Name: Cost Description: Enter a description Synonyms: cost Display folder: Tip & Best Practices Is hidden: No Formatted General Percentage format: No Thousands separator: No Decimal places: Auto Advanced

Product Lookup

- current_user
- current_user_price
- current_wholesale_price
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- product_weight
- product_weight_m
- unit_of_measure

Measure Table

- U_Cost
- Customer Sales
- Profit

3) Expand the Properties sections so that the options can be viewed.

1) Click on the Model view to get to this view.

2) Click on the Cost measure.

4) Type in the appropriate folder name.

The screenshot shows the Power BI Desktop interface. On the left, the 'Model' view displays two tables: 'Product Lookup' and 'Measure Table'. The 'Product Lookup' table contains fields like current_user, current_user_price, and product. The 'Measure Table' table contains fields like U_Cost, Customer Sales, and Profit. In the center, a 'Cost' measure is selected in the 'Fields' pane. A blue arrow points from the text '1) Click on the Model view to get to this view.' to the 'Model' view icon in the top-left corner. Another blue arrow points from the text '2) Click on the Cost measure.' to the 'Cost' measure in the 'Fields' pane. A yellow arrow points from the text '4) Type in the appropriate folder name.' to the 'Display folder' section in the 'Properties' pane. A blue arrow points from the text '3) Expand the Properties sections so that the options can be viewed.' to the 'Properties' pane itself.



The Data view shows three tables:

- Product Lookup:** Contains columns like current_id, current_total_price, current_wholesale_price, product, product_category, product_description, product_group, product_id, product_type, product_uom, total_weight_uom, and unit_of_measure.
- Store Lookup:** Contains columns like manager, neighborhood, store_address, store_city, store_id, store_latitude, store_longitude, store_postal_code, store_square_feet, and storeטלפון.
- Measure Table:** Contains columns like Cost, Customer Sales, and Profit.

The Properties pane is open for the 'Customer Sales' measure in the Measure Table. It includes sections for General (Name: Customer Sales, Description: Enter a description), Synonyms (customer_sales, sales), Display folder (Tip & Best Practices), Is hidden (No), and Formatting (Format: Whole number, Percentage format: No, Thousands separator: No, Decimal places: 0). A blue bracket on the right side of the pane groups the 'Customer Sales' and 'Profit' items under the 'Tip & Best Practices' folder.

Customer Sales & Profit measures can also be grouped (drag & drop) to the Tips & Best Practices folder within the Measure Table for better management rather than trying to find where all the measures are located.

Error Handling

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Error handling functions can be used to help identify missing data, and can be particularly useful for quality assurance and testing

IFERROR()

Returns a value if the first expression is an error and the value of the expression itself otherwise

=**IFERROR**(Value,ValueIfError)

ISBLANK()

Checks to see if a value is blank, returns True or False

=**ISBLANK**(Value)

```
1 Error Checking Example =  
2 IFERROR(  
3     1/0,  
4     BLANK()  
5 )
```

```
1 Customer Sales LY (ISBLANK) =  
2 IF(  
3     ISBLANK(  
4         [Customer Sales (Last Year)]  
5     ),  
6     "No Sales",  
7     [Customer Sales (Last Year)]  
8 )
```

PRO TIP:

VertiPaq can't optimize **IFERROR** and **ISBLANK** functions, so avoid using them in your operational code (a better use is for temporary QA & testing)

Maven Roosten - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools

Name: Measure Table

Mark as data table Manage relationships New measure Quick New measure column New table

Structure

Write a DAX expression that calculates a value from your data.

Cost Customer Sales Profit

1,089,176.27 423,270.5 3,163,328.61

2) Click on New measure.

1) Select the Measure Table.

Filters Visualizations Fields

Search

Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Measure Table

Calendar Customer Lookup Employee Lookup Food Inventory Tip & Best Practices Product Lookup Sales by Store Store Lookup

Page 1 +

Page 1 of 1

Maven Router - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table spots Measure tools

Name: Customer Sales (La... Data category: Uncategorized

Home table: Measure Table \$ - % .00 : New Quick measure measure Calculations

Structure Formatting Properties Calculations

1 Customer Sales (Last Year) =
2 CALCULATE(
3 [Customer Sales],
4 DATEADD(
5 'Calendar'[Transaction_Date],
6 -1,
7 Year
8)
9)

Visualizations Fields

Search: Customer Sales (Last Year)

Calender Customer Lookup Employee Lookup Food Inventory Measure Table Tip & Best Practices Cost Customer Sales Profit ProductLookup Sales by Store Store Lookup

Values Add data field here Drill through Cross-report Off On Keep all filters On Add drill-through fields here

Page 1 +

5) Change Format to currency & decimal place to 2.

3) Type in the following code to create the Customer Sales (Last Year) measure.

4) Click on the Customer Sales (Last Year) measure created.

Maven Router - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get Data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Home

6) Bring in the Transaction_Date from the Calendar table into the visual.

Filters

Visualizations

Fields

Search

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Keep all filters

On

Add drill-through fields here

Calendar

- Transaction_Date
- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Week_Day
- Week_ID
- Week_Year
- Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

- Customer Sales (Last Year)
- Tip & Best Practices
 - Cost
 - Customer Sales
 - Profit
- Product Lookup
- Sales by Store
- Store Lookup

Cost Customer Sales Profit

Cost	Customer Sales	Profit
1,089,176.27	425,270.5	3,140,526.61

Transaction_Date

Transaction_Date
01 January 2017
02 January 2017
03 January 2017
04 January 2017
05 January 2017
06 January 2017
07 January 2017
08 January 2017
09 January 2017
10 January 2017
11 January 2017
12 January 2017
13 January 2017

Page 1

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dashboard Data Queries Insert Calculations Share

Cost Customer Sales Profit
1,089,176.27 423,703 3,163,528.41

Transaction_Date
01 January 2017
02 January 2017
03 January 2017
04 January 2017
05 January 2017
06 January 2017
07 January 2017
08 January 2017
09 January 2017
10 January 2017
11 January 2017
12 January 2017
13 January 2017
14 January 2017

7) Change the visual to a matrix.

Filters Visualizations Fields

Search

Filters on this visual Transaction_Date is (All)

Add data fields here

Rows Transaction_Date

Filters on this page Add data fields here

Columns Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Off

Keep all filters On

Add drill-through fields here

Calendar

- Customer_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Day
- Week_ID
- Week_Year
- Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

- Customer Sales (Last Year)
- Tip & Best Practices
 - Cost
 - Customer Sales
 - Profit

Product Lookup

Sales by Store

Store Lookup

Page 1 +

Page 1 of 1

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Column tools

Name Transaction_Date

Date type Date

Structure

114/03/2001 (dd/mm/yyyy)

14 March 2001 (dd/mmmm yyyy)

14 March 2001 (d mmmm yyyy)

Wednesday, 14 March 2001 (ddddd, d mmmm yyyy)

Wednesday, 14 March 2001 (ddddd, dd mmmm yyyy)

14/03/01 (dd/mm/yy)

14/3/01 (d/m/yy)

14.3.01 (d.m.y)

2001-03-14 (yyyy-mm-dd)

March 2001 (mmmmm yyyy)

2001-03 (yyyy-mm)

14 March (d mmmm)

01 (yyl)

2001 (yyy)

Sort by column ▾

Data groups ▾

Manage relationships ▾

New column ▾

Σ Summarization ▾

Don't summarize ▾

Page 1 +

8) Click on the Transaction_Date & change the date format.

Filters

Search

Transaction_Date

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Visualizations

Fields

Calendar

Week Desc

Week ID

Week Year

Year ID

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

Customer Sales (Last Year)

Tip & Best Practices

Cost

Customer Sales

Profit

Product Lookup

Sales by Store

Store Lookup

Off

On

Keep all filters

Add drill-through fields here

Page 1 of 1

Maven Rooster - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data -> Excel Power BI datasets Server Enter data Refresh sources Transform data -> New visual Text box More visuals -> New measure measure Quick Publish

Dashboard Data Queries Insert Calculations Share

Cost Customer Sales Profit

1,089,176.27 4252704.88 3,163,328.81

Transaction Date Customer Sales Customer Sales (Last Year)

Transaction Date	Customer Sales	Customer Sales (Last Year)
20/12/2017	5306.48	
21/12/2017	5867.91	
22/12/2017	5603.81	
23/12/2017	5913.01	
24/12/2017	6160.35	
25/12/2017	5876.95	
26/12/2017	5733.48	
27/12/2017	5932.53	
28/12/2017	4773.40	
29/12/2017	4479.15	
30/12/2017	5184.03	
31/12/2017	5037.98	
01/01/2018	4360.80	\$2,508.20
02/01/2018	4071.05	\$2,403.35
03/01/2018	4203.75	\$2,368.00
04/01/2018	4013.15	\$2,229.10
05/01/2018	4333.80	\$2,418.85
06/01/2018	3845.90	\$2,271.85
07/01/2018	4379.18	\$2,767.00
Total	4252704.88	\$1,236,644.83

You will notice that there are lots of blank values here. A ISBLANK new measure can fix this issue.

Filters Visualizations Fields

Search

Customer Sales is (All)

Customer Sales (Last Year) is (All)

Transaction Date is (All)

Add data fields here

Row:

Transaction Date

Columns:

Add data fields here

Values:

Add data fields here

Customer Sales

Customer Sales (Last Year)

Drill through:

Customer Sales (Last Year)

Tip & Best Practices

Cost

Customer Sales

Product Lookup

Sales by Store

Store Lookup

Keep all filters

On

Add drill-through fields here

Page 1 +

9) Drag in the Customer Sales & Customer Sales (Last Year) measures to the Values section.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Customer Sales (LY ISBLANK)
Home table: Measure Table

Measure tools: New Quick measure measure

Structure, Formatting, Properties, Calculations

Visualizations > Fields >

Search: Calendar

- Customer Sales (Last Year)
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_ID

Rows: Transaction_Date

Add data fields here

Filters on this page

Add data fields here

Values: Customer Sales (Last Year)

Customer Sales (Last Year)

Customer Sales LY (ISBLANK)

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Tip & Best Practices

- Cost
- Customer Sales
- Profit

Product Lookup

Sales by Store

Store Lookup

Customer Sales LY (ISBLANK)

1 Customer Sales LY (ISBLANK) =
2 IF(
3 ISBLANK(
4 [Customer Sales (Last Year)]
5),
6 "No Sales",
7 [Customer Sales (Last Year)]
8)

26/12/2017 4771.40
28/12/2017 4478.15
30/12/2017 5184.03
31/12/2017 5037.99
01/01/2018 4160.80
02/01/2018 4071.05
03/01/2018 4203.73
04/01/2018 4011.15
05/01/2018 4033.60
06/01/2018 3943.90
07/01/2018 4079.18
Total 4252704.88 \$2,236,644.83

10) Create the Customer Sales LY (ISBLANK) formulae here. Return “No Sales” when Customer Sales are blank.

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter Options Out data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text Box More visual New Measure measure Publish

Queries Data Overview Item Descriptions Share

Cost Customer Sales Profit

	Customer Sales	Customer Sales (Last Year)	Customer Sales LY (ISBLANK)
20/12/2017	5506.48		No Sales
21/12/2017	5667.91		No Sales
22/12/2017	5632.81		No Sales
23/12/2017	5513.01		No Sales
24/12/2017	6160.35		No Sales
25/12/2017	5675.95		No Sales
26/12/2017	5731.48		No Sales
27/12/2017	5832.53		No Sales
28/12/2017	4773.40		No Sales
29/12/2017	4479.15		No Sales
30/12/2017	5184.03		No Sales
31/12/2017	5037.98		No Sales
01/01/2018	4560.80	\$2,560.20	2,560.20
02/01/2018	4671.05	\$2,403.35	2,403.35
03/01/2018	4203.75	\$2,565.00	2,565.00
04/01/2018	4013.15	\$2,220.10	2,220.10
05/01/2018	4333.69	\$2,416.85	2,416.85
06/01/2018	3843.90	\$2,271.85	2,271.85
07/01/2018	4379.18	\$2,787.00	2,787.00
Total	4252704.88	\$1,2370,644.83	2,236,644.83

Filters Visualizations Fields

Search

Customer Sales (All)

Customer Sales (Last Year) (All)

Customer Sales LY (ISBLANK) (All)

Transaction Date (All)

Add data fields here

Row

Transaction Date

Columns

Add data fields here

Values

Customer Sales

Customer Sales (Last Year)

Customer Sales LY (ISBLANK)

Measure Table

Customer Sales (Last Year)

Customer Sales LY (ISBLANK)

Tip & Best Practices

Cost

Customer Sales

Profit

Product Lookup

Sales by Store

Store Lookup

Calendar

Date ID

Month ID

Month Name

Quarter ID

Quarter Name

Transaction Date

Week Desc

Week ID

Week Year

Year ID

Customer Lookup

Employee Lookup

Food Inventory

Customer Sales (ISBLANK)

Drill through

Cross-report

Off On

Keep all items

On

Add drill-through fields here

Page 1

Page 1 of 1

11) Bring the Customer Sales LY (ISBLANK) measure into the matrix to see the output. You will notice that the blank entries are now populated with "No Sales" from the applied error handling from the previous steps.

Creating & Using DAX Variables

Abbreviations

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Variables (VAR) are DAX expressions which can be reused multiple times within a query, and are commonly used for **two key purposes**:

- **Readability:** Variables make complex code more human readable
- **Performance:** Variables are only evaluated *once* no matter how often they are used within a query



What they **CAN** do:

- **Simplify and streamline** DAX code
- **Improve efficiency** by eliminating redundant expressions
- Evaluate **in the order they're written** (*variables can only reference previously declared variables within the query*)
- Store either **table** or **scalar** values



What they **CANNOT** do:

- Start with a **number**
- Include **spaces** or **special characters** (*except underscores “_”*)
- **Share the name** of another table in the model
- Be accessed **outside the query** in which they are declared
- Contain only **certain keywords** reserved for DAX (*SUM, Date, CALCULATE, etc.*)

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

DAX queries which use variables must include two key components: the **declaration expression** and the **return expression**:

VAR <name> = <expression> **RETURN <result_expression>**

```
1 Orders by Females =
2 VAR TotalOrders_Female =
3 CALCULATE(
4     SUM(
5         'Sales by Store'[quantity_sold]
6     ),
7     FILTER(
8         'Customer Lookup',
9         'Customer Lookup'[gender] = "F"
10    )
11 )
12 RETURN
13 TotalOrders_Female
```

The **declaration expression (VAR)** is where you declare a new variable, assign a name, and write an expression to define the variable

The **return expression (RETURN)** is where you evaluate the rest of your DAX query, and reference any previously declared variables

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Orders by Females Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % .0

New Quick measure measure

Structure Formatting Properties Calculations

1 Orders by Females =
2 VAR TotalOrders_Female =
3 CALCULATE(
4 SUM(
5 'Sales by Store'[quantity_sold]
6),
7 FILTER(
8 'Customer Lookup',
9 'Customer Lookup'[gender] = "F"
10)
11)
12 RETURN
13 TotalOrders_Female

Visualizations Fields

Search

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Day
- Week_ID
- Week_Year
- Year_ID

Values

Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

- Customer Sales (Last Year)
- Customer Sales LY (\$81,680)
- Orders by Females

Tip & Best Practices

- Cost
- Customer Sales
- Profit

ProductLookup

Sales by Store

Store Lookup

Page 1 +

Create a “TotalOrders_Female” variable with a RETURN statement if skipped out a error message will appear.



Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dataset Data Query Insert Calculations Share

Transaction_Date Customer Sales Orders by Females

Transaction_Date	Customer Sales	Orders by Females
20/12/2017	5306.48	199
21/12/2017	5867.91	276
22/12/2017	5632.81	256
23/12/2017	5913.01	212
24/12/2017	6160.35	282
25/12/2017	5676.95	257
26/12/2017	5733.48	258
27/12/2017	5932.51	263
28/12/2017	4771.40	350
29/12/2017	4479.15	318
30/12/2017	5184.03	313
31/12/2017	5037.90	336
01/01/2018	4360.00	416
02/01/2018	4071.05	582
03/01/2018	4260.75	576
04/01/2018	4013.15	583
05/01/2018	4331.60	632
06/01/2018	3845.90	480
07/01/2018	4379.18	297
Total	4252704.88	283427

Filters

Visualizations

Fields

Search

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off

Keep all filters

Add drill-through fields here

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

Customer Sales (Last Year)

Customer Sales LY (381,840)

Orders by Females

Tip & Best Practices

Cost

Customer Sales

Profit

ProductLookup

Sales by Store

Store Lookup

Page 1

+

Bring in the Orders by Females measure into the matrix. So that's how you create variables & use them.

Variable Evaluation Order

Variables

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Variables are “locked in” as soon as the DAX engine reads them; this means you cannot modify how a variable is defined later in your query (*i.e. through a CALCULATE function*)

Total quantity sold to Female customers = **283,427**

```
1 Orders by Females - Wrong (VAR Evaluation Order) =
2 VAR TotalOrders =
3 SUM(
4     'Sales by Store'[quantity_sold]
5 )
6 RETURN
7 CALCULATE(
8     TotalOrders,
9     FILTER(
10        'Customer Lookup',
11        'Customer Lookup'[gender] = "F"
12    )
13 )
```

= **1,305,637**



```
1 Orders by Females - Right (VAR Evaluation Order) =
2 VAR TotalOrders_Female =
3 CALCULATE(
4     SUM(
5         'Sales by Store'[quantity_sold]
6     ),
7     FILTER(
8         'Customer Lookup',
9         'Customer Lookup'[gender] = "F"
10    )
11 )
12 RETURN
13 TotalOrders_Female
```

= **283,427**

Variable can't be modified
after it's been defined



Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Orders by Females... Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % .0

New Quick measure measure

Structure Formatting Properties Calculations

1 Orders by Females (Wrong) =
2 VAR TotalOrder =
3 SUM(
4 'Sales by Store'[quantity_sold]
5)
6 RETURN
7 CALCULATE(
8 TotalOrder,
9 FILTER(
10 'Customer Lookup',
11 'Customer Lookup'[gender] = "F"
12)
13)

Visualizations Fields

Search

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Day
- Week_ID
- Week_Year
- Year_ID

Values

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

- Customer Sales (Last Year)
- Customer Sales LY (381,484)
- Orders by Females
- Orders by Females (Wrong)

Tip & Best Practices

- Cost
- Customer Sales
- Profit

Product Lookup

Sales by Store

Store Lookup

The “TotalOrder” variable is accepted & no errors are reported.

	06/01/2018	3845.90	480
07/01/2018	4379.18	297	
Total	4252704.88	283427	

Page 1 +

Maven Rooters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Data Transform Refresh New Quick Publish

Out Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh New measure Insert Calculations Share

Format painter

Document Data Queries Insert Calculations Share

Filters

Visualizations

Fields

Search

Customer Sales is (All)

Orders by Females is (All)

Orders by Females (M... is (All)

Transaction Date is (All)

Add data fields here

Transaction Date

Rows

Columns

Values

Customer Sales

Orders by Females

Orders by Females (Wron...

Drill through

Cross-report

Keep all filters

On

Add drill-through fields here

Search

Calendar

Customer ID

Month ID

Month Name

Quarter ID

Quarter Name

Transaction Date

Week Desc

Week ID

Week Year

Year ID

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

Customer Sales (Last Year)

Customer Sales LY (SBLANK)

Orders by Females

Orders by Females (Wrong)

Tip & Best Practices

Cost

Customer Sales

Profit

Product Lookup

Sales by Store

Store Lookup

Transaction Date Customer Sales Orders by Females Orders by Females (Wrong)

Transaction Date	Customer Sales	Orders by Females	Orders by Females (Wrong)
20/12/2017	5306.48	193	1651
21/12/2017	5867.91	276	1736
22/12/2017	5832.81	256	1733
23/12/2017	5913.01	212	1664
24/12/2017	6160.35	282	1811
25/12/2017	5676.95	257	1782
26/12/2017	5733.48	250	1789
27/12/2017	5932.53	263	1951
28/12/2017	4773.40	250	1546
29/12/2017	4479.15	318	1432
30/12/2017	5184.03	311	1580
31/12/2017	5937.98	220	1531
01/01/2018	4360.80	16	1401
02/01/2018	4071.05	562	1316
03/01/2018	4369.73	376	1346
04/01/2018	4811.15	583	1300
05/01/2018	4333.60	632	1379
06/01/2018	3845.90	480	1237
07/01/2018	4379.18	297	1347
Total	4252704.88	283427	1905637

When you evaluate the “Orders by Female (Wrong)” you will notice that the output is incorrect.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Orders by Females ... Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % , . 0

Measure Formatting Properties Calculations

Visualizations Fields

Search

Calendar

Date_ID Month_ID Month_Name Quarter_ID Quarter_Name Transaction_Date Week_Day Week_ID Week_Year Year_ID

Rows

Transaction_Date

Columns

Add data fields here

Values

Customer Sales Orders by Females Orders by Females (Wc) Orders by Females (Wrong)

Customer Lookup Employee Lookup Food Inventory Measure Table

Customer Sales (Last Year) Customer Sales LY (SBLANK) Orders by Females Orders by Females (Wrong)

Tip & Best Practices Cost Customer Sales Profit

Product Lookup Sales by Store Store Lookup

1 Orders by Females (Wrong) =
2 VAR TotalOrder =
3 SUM(
4 'Sales by Store'[quantity_sold]
5)
6 RETURN
7 CALCULATE(
8 TotalOrder,
9 FILTER(
10 'Customer Lookup',
11 'Customer Lookup'[gender] = "F"
12)
13)

06/01/2018 3645.00 460 1237
07/01/2018 4379.18 297 1347
Total 4252704.88 283427 1306637

Add data fields here

Filters on all pages Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Page 1 +

Once the TotalOrder variable is defined over here the VertiPaq engine locks the variable here. If the TotalOrder variable is referenced later in the syntax the operations will be ignored.

PRO TIP: Using Variables to Text & Debug DAX

Shortcuts

Formatting

Evaluation Order

Comments

Measure Tables

Error Handling

Variables

Variables can be a helpful tool for **testing or debugging** your DAX code

```
1 % Quantity Sold to Females (VAR) =  
2 VAR TotalFemaleOrders =  
3 CALCULATE(  
4     SUM('Sales by Store'[quantity_sold]),  
5     FILTER(  
6         'Customer Lookup',  
7         'Customer Lookup'[gender] = "F"  
8     )  
9 )  
10 VAR QuantitySold =  
11 SUM(  
12     'Sales by Store'[quantity_sold]  
13 )  
14 VAR Ratio =  
15 DIVIDE(  
16     TotalFemaleOrders,  
17     QuantitySold,  
18     "_"  
19 )  
20  
21 RETURN  
22 Ratio
```

In this example we're able to:

- Use variables to define individual components of a larger, more complex measure
- Use the RETURN expression to quickly check the output of each variable
- Identify which specific component is the root cause in the case of an error

Maven Routes - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name % Quantity Sold to Females

Home table Measure Table

Format Whole number

Data category Uncategorized

New Quick measure measure

Syntax

Properties Calculations

```
1 % Quantity Sold to Females =  
2 VAR TotalOrders_Female =  
3 CALCULATE(  
4     SUM(  
5         'Sales by Store'[quantity_sold]  
6     ),  
7     FILTER(  
8         'Customer Lookup',  
9         'Customer Lookup'[gender] = "F"  
10    )  
11 )  
12 -- The second QuantitySold variable is added below  
13 VAR QuantitySold =  
14 sum(  
15     'Sales by Store'[quantity_sold]  
16 )  
17 -- The third Ratio variable is added below  
18 VAR Ratio =  
19 DIVIDE(  
20     TotalOrders_Female,  
21     QuantitySold,  
22     "..."  
23 )  
24 RETURN  
25 TotalOrders_Female
```

Visualizations >

Fields

Search

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desk
- Week_ID
- Week_Year
- Year_ID

Rows

Transaction_Date

Columns

Add data fields here

Values

Customer Sales

% Quantity Sold to Females

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

% Quantity Sold to Females

- Customer Sales (Last Year)
- Customer Sales (Y OYOLAN)
- Orders by Females (Wrong)

Tip & Best Practices

- Cost
- Customer Sales
- Profit

Product Lookup

Sales by Store

Store Lookup

Type in the above syntax you will notice that two additional variable have been added for QuantitySold & Ratio.

Maven Reactors - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New Measure measure Quick Publish

Queries Datasets Data Insert Calculations Share

Filters Visualizations Fields

Search

Filters on this page Add data fields here

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Customer Lookup Employee Lookup Food Inventory Measure Table

% Quantity Sold to Females Customer Sales (Last Year) Customer Sales LY (ISBLANK) Orders by Females (Wrong) Tip & Best Practices Cost Customer Sales Profit Product Lookup Sales by Store Store Lookup

Transaction Date Customer Sales % Quantity Sold to Females

Transaction Date	Customer Sales	% Quantity Sold to Females
01/01/2017	2508.20	157
02/01/2017	2403.35	367
03/01/2017	2565.00	366
04/01/2017	2220.10	316
05/01/2017	2418.85	333
06/01/2017	2271.85	287
07/01/2017	2787.00	200
08/01/2017	2636.33	187
09/01/2017	2674.61	125
10/01/2017	2685.65	182
11/01/2017	2533.73	216
12/01/2017	2327.70	186
13/01/2017	3031.60	216
14/01/2017	2682.51	115
15/01/2017	2167.71	110
16/01/2017	2829.16	93
17/01/2017	3285.80	97
18/01/2017	2735.96	85
19/01/2017	2913.68	73
Total	4252704.88	283427

Name of measure is updated correctly although the values in the table are incorrect.

Page 1

Maven Router - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name % Quantity Sold to Females % Format Whole number Data category Uncategorized New measure Quick measure measure

Home table Measure Table \$ - % .00 Properties Calculations

Structure Formatting

1 % Quantity Sold to Females =
2 VAR TotalOrders_Female =
3 CALCULATE(
4 SUM(
5 'Sales by Store'[quantity_sold]
6),
7 FILTER(
8 'Customer Lookup',
9 'Customer Lookup'[gender] = "F"
10)
11)
12 -- The second QuantitySold variable is added below
13 VAR QuantitySold =
14 sum(
15 'Sales by Store'[quantity_sold]
16)
17 -- The third Ratio variable is added below
18 VAR Ratio =
19 DIVIDE(
20 TotalOrders_Female,
21 QuantitySold,
22 "-"
23)
24 RETURN
25 QuantitySold

Visualizations Fields

Search

Values Add data fields here

Drill through

Cross-report

Off On Keep all filters

Add drill-through fields here

Customer Lookup Employee Lookup Food Inventory Measure Table

% Quantity Sold to Females Customer Sales (Last Year) Customer Sales LY (US\$LAVG) Orders by Females (Wrong) Tip & Best Practices Cost Customer Sales Profit Product Lookup Sales by Store Store Lookup

Change the variable “TotalOrders_Female” to “QuantitySold” as shown above.

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get Data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dashboard Data Queries Calculations Share

Filters Visualizations Fields

Search

Filters on this page Add data fields here

Filters on all pages Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off On Keep all filters

On

Add drill-through fields here

Customer Lookup Employee Lookup Food Inventory Measure Table

% Quantity Sold to Females Customer Sales (Last Year) Customer Sales (Y (SBLANG)) Orders by Females (Wrong) Tip & Best Practices Cost Customer Sales Profit Product Lookup Sales by Store Store Lookup

Transaction Date Customer Sales % Quantity Sold to Females

Transaction Date	Customer Sales	% Quantity Sold to Females
01/01/2017	2508.20	602
02/01/2017	2403.33	790
03/01/2017	2345.00	823
04/01/2017	2220.19	726
05/01/2017	2416.85	778
06/01/2017	2271.85	734
07/01/2017	2187.00	836
08/01/2017	2638.33	806
09/01/2017	2674.61	742
10/01/2017	2685.65	835
11/01/2017	2553.75	762
12/01/2017	2327.70	739
13/01/2017	3033.60	950
14/01/2017	2682.51	771
15/01/2017	2167.71	927
16/01/2017	2829.16	855
17/01/2017	2385.80	762
18/01/2017	2735.98	827
19/01/2017	2913.68	858
Total	4252704.88	1305637

QuantitySold values show correctly.

Maven Routes - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name % Quantity Sold to Female

Home table Measure Table

Source %

Formatting

Properties Calculations

1 % Quantity Sold to Female :=
2 VAR TotalOrders_Female =
3 CALCULATE(
4 SUM(
5 'Sales by Store'[quantity_sold]
6),
7 FILTER(
8 'Customer Lookup',
9 'Customer Lookup'[gender] = "F"
10)
11)
12 -- The second QuantitySold variable is added below
13 VAR QuantitySold =
14 sum(
15 'Sales by Store'[quantity_sold]
16)
17 -- The third Ratio variable is added below
18 VAR Ratio =
19 DIVIDE(
20 TotalOrders_Female,
21 QuantitySold,
22 "")
23)
24 RETURN
25 Ratio

Visualizations

Fields

Search

Calendar

Date_ID

Month_ID

Month_Name

Quarter_ID

Quarter_Name

Transaction_Date

Week_Day

Week_ID

Week_Year

Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

% Quantity Sold to Female

Customer Sales

Customer Sales (Last Year)

Customer Sales YTD (SALARY)

Orders by Females (Wrong)

Tip & Best Practices

Cost

Customer Sales

Profit

Product Lookup

Sales by Store

Store Lookup

Page 1

1) Change the variable “QuantitySold” to “Ratio” as shown above.

2) Change the format to % & apply 2 decimal places so that the values appear correctly in the matrix table.

Maven Reporters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals = Quick measure measure Publish

Dashboard Data Queries Insert Calculations Share

Transaction Date Customer Sales % Quantity Sold to Females

Transaction Date	Customer Sales	% Quantity Sold to Females
01/01/2017	2500.20	44.51%
02/01/2017	2400.85	46.46%
03/01/2017	2365.00	44.47%
04/01/2017	2220.10	41.93%
05/01/2017	2410.85	42.80%
06/01/2017	2371.85	38.99%
07/01/2017	2787.00	21.92%
08/01/2017	2636.33	23.20%
09/01/2017	2676.61	16.85%
10/01/2017	2685.65	21.29%
11/01/2017	2555.75	27.62%
12/01/2017	2327.70	24.51%
13/01/2017	3031.60	22.74%
14/01/2017	2682.51	14.92%
15/01/2017	3167.71	11.87%
16/01/2017	2829.16	10.88%
17/01/2017	3285.80	12.73%
18/01/2017	2735.96	10.26%
19/01/2017	2911.68	8.51%
Total	4252794.88	21.71%

The values for % Quantity Sold to Females displays correctly also. Variables are a great way to test & debug your code is functioning correctly i.e. returning values that you are expecting.

Filters

Visualizations

Fields

Search

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Day
- Week_ID
- Week_Year
- Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Measure Table

- % Quantity Sold to Females
- Customer Sales (Last Year)
- Customer Sales LY (USING)
- Orders by Females (Wrong)

Tip & Best Practices

- Cost
- Customer Sales
- Profit

Product Lookup

Sales by Store

Store Lookup

Page 1

Page 1 of 1

4) Scalar Functions

Introduction - Scalar Functions



Scalar functions return a **single value**, rather than a column or table; common examples include aggregation, conversion, rounding, and logical functions

TOPICS WE'LL COVER:

Aggregation
Functions

Rounding
Functions

Information
Functions

Conversion
Functions

Logical
Functions

COMMON USE CASES:

- *Aggregating a column of values into a single number (i.e. average customer age, maximum product price, sum of revenue, count of orders, etc.)*
- *Converting fields into desired formats (i.e. text to dates, integers to currency, etc.)*
- *Evaluating logical tests and returning values for TRUE and FALSE responses*

Common Scalar Functions

AGGREGATION Functions

Functions that can be used to **dynamically aggregate** values within a column

Common Examples:

- SUM
- AVERAGE
- MAX
- MIN
- COUNT
- COUNTA
- DISTINCTCOUNT
- PRODUCT
- ITERATOR ("X")
- FUNCTIONS

ROUNDING Functions

Functions that can be used to **round values** to different levels of precision

Common Examples:

- FLOOR
- TRUNC
- ROUNDDOWN
- MROUND
- ROUND
- CEILING
- ISO.CEILING
- ROUNDUP
- INT
- FIXED

INFORMATION Functions

Functions that can be used to analyze the **data type** or output of an expression

Common Examples:

- ISBLANK
- ISERROR
- ISLOGICAL
- ISNONTEXT
- ISNUMBER
- ISTEXT

CONVERSION Functions

Functions that are used to force a specific **data type conversion**

Common Examples:

- CURRENCY
- INT
- FORMAT
- DATE
- TIME
- DATEVALUE
- VALUE

LOGICAL Functions

Functions for returning information about values in a **conditional expression**

Common Examples:

- IF
- AND
- OR
- NOT
- TRUE/FALSE
- SWITCH
- COALESCE

Aggregation Functions

Aggregation Functions

Rounding Functions

Information Functions

Conversion Functions

Logical Functions

SUM()

Evaluates the sum of a column

=SUM(Column**Name**)

AVERAGE()

Returns the average (arithmetic mean) of all the numbers in a column

=AVERAGE(Column**Name**)

MAX()

Returns the largest value in a column or between two scalar expressions

=MAX(Column**Name**) or
=MAX(Scalar1, [Scalar2])

MIN()

Returns the smallest value in a column or between two scalar expressions

=MIN(Column**Name**) or
=MIN(Scalar1, [Scalar2])

COUNT()

Counts the number of cells in a column that contain numbers

=COUNT(Column**Name**)

DISTINCTCOUNT()

Counts the number of distinct or unique values in a column

=DISTINCTCOUNT(Column**Name**)

COUNTROWS()

Counts the number of rows in the specified table, or a table defined by an expression

=COUNTROWS(Table)



PRO TIP:

For large datasets (1M+ rows) using **COUNTROWS & VALUES** may put less strain on the DAX engines than **DISTINCTCOUNT**

```
1 Total Employees =  
2 COUNTROWS(  
3     VALUES(  
4         'Employee Lookup')
```

Maven Rooters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Table tools Measure tools

Name: Total Customers Format: Whole number Data category: Uncategorized

Home table: Sales

New measure Quick measure measure Calculations Properties

Structure Formatting

1 Total Customers =
2 DISTINCTCOUNT(
3 'Customer Lookup'[customer_id]
4)

02/01/2017 3403.35 46.49%
03/01/2017 2565.00 44.47%
04/01/2017 2220.10 43.53%
05/01/2017 2418.85 42.80%
06/01/2017 2273.85 38.99%
07/01/2017 2787.00 23.92%
08/01/2017 2638.53 23.20%
09/01/2017 2676.61 16.85%
10/01/2017 2685.65 21.29%
11/01/2017 2555.75 27.62%
12/01/2017 2327.70 24.51%
13/01/2017 3033.60 22.74%
14/01/2017 2682.51 14.92%
15/01/2017 3167.71 11.87%
16/01/2017 2829.16 10.88%
17/01/2017 3285.80 12.73%
18/01/2017 2735.96 10.28%
19/01/2017 2913.68 8.51%
Total 4252704.88 21.71%

Filters on all pages Add data fields here

Add data fields here

Values

Drill through

Cross-report

Off On

Keep all filters

Add drill-through fields here

Visualizations Fields

Measure Table Calendar

Total Customers Transaction_Week Date_ID Month_ID Month_Name Quarter_ID Quarter_Name Year_ID Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup

Page 1 +

Page 1 of 1

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Total Customers

Home table **Customer Lookup**

Format Whole number Data category Uncategorized

Calendar New measure measure Quick measure measure

Properties Calculations

Customer Lookup
Employee Lookup
Food Inventory
Measure Table
Product Lookup
Sales by Store
Store Lookup

Measure Table

Customer Lookup'[customer_id]

Customers =
= COUNT(
Customer Lookup'[customer_id])

01/01/2017 2403.35 46.46%
02/01/2017 2565.00 44.47%
03/01/2017 2220.10 43.53%
04/01/2017 2418.85 42.80%
05/01/2017 2273.85 38.99%
06/01/2017 2787.00 23.92%
07/01/2017 2638.53 23.20%
08/01/2017 2676.61 16.85%
09/01/2017 2685.65 21.29%
10/01/2017 2555.75 27.62%
11/01/2017 2327.70 24.51%
12/01/2017 3033.60 22.74%
13/01/2017 2682.51 14.92%
14/01/2017 3167.71 11.87%
15/01/2017 2829.16 10.88%
16/01/2017 3285.80 12.73%
17/01/2017 2735.96 10.28%
18/01/2017 2913.68 8.51%
Total 4252704.88 21.71%

Filters on all pages Add data fields here

Visualizations Fields

Search

Measure Table

Calendar

Data ID Month ID Month Name Quarter ID Quarter Name

Total Customers Transaction Week Desc Week ID Week Year

Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup

Move the Total Customers measure to the Measure Table.

Page 1

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Total Customers Format Whole number Data category Uncategorized New Quick measure measure

Home table Microsoft Data \$ - % ?

Issue Display the values in this column with commas as a thousands separator.

Properties Calculations

1 Total Customers =
2 DISTINCTCOUNT(
3 'Customer Lookup'[customer_id]
4)

02/01/2017 2409.35 46.68%
03/01/2017 2565.00 44.47%
04/01/2017 2220.10 43.51%
05/01/2017 2419.85 42.80%
06/01/2017 2273.85 38.99%
07/01/2017 2787.00 23.92%
08/01/2017 2638.53 23.20%
09/01/2017 2676.61 16.89%
10/01/2017 2665.65 21.29%
11/01/2017 2555.75 27.62%
12/01/2017 2327.70 24.51%
13/01/2017 3033.60 22.74%
14/01/2017 2682.51 14.92%
15/01/2017 3167.71 11.07%
16/01/2017 2829.16 10.88%
17/01/2017 3205.80 12.73%
18/01/2017 2735.95 10.28%
19/01/2017 2913.60 8.51%
Total 4253704.88 21.71%

Filters on all pages Add data fields here

Visualizations Fields

Search

Measure Table

- % Quantity
- Customer S...
- Customer S...
- Orders by F...
- Total Cust...

Tip & Best Practices

Calendar

- Date_ID
- Month_ID
- Month_Na...
- Quarter_ID
- Quarter_Na...
- Transaction...
- Week_Desk
- Week_ID
- Week_Near
- Year_ID

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

Page 1

Page 1 of 1

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Total Employees Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % ↴ ↵ ↷ ↸ New Quick measure measure

Structure Formatting Properties Calculations

Click here to display the Total Employees with commas.

1 Total Employees =
2 COUNTROWS(
3 'Employee Lookup'
4)

02/01/2017 2403.35 46.48%
03/01/2017 2565.00 44.47%
04/01/2017 2220.10 43.53%
05/01/2017 2418.85 42.80%
06/01/2017 2273.85 38.99%
07/01/2017 2787.00 23.92%
08/01/2017 2638.53 23.20%
09/01/2017 2676.61 16.83%
10/01/2017 2685.65 21.29%
11/01/2017 2555.75 27.62%
12/01/2017 2927.70 24.51%
13/01/2017 3033.60 22.74%
14/01/2017 2682.51 14.92%
15/01/2017 3167.71 11.87%
16/01/2017 2829.56 10.88%
17/01/2017 3285.80 12.73%
18/01/2017 2755.96 10.20%
19/01/2017 2913.68 8.51%
Total 4252704.88 21.71%

Filters on all pages Add data fields here

Values Add data fields here

Drill through

Cross-report

Off On

Keep all filters

Add drill-through fields here

Visualizations Fields

Measure Table

% Quantity ... Customer_S_... Customer_S_... Orders by F... Tip & Best Practices Total Cust... Total Emplo... Calendar Date_ID Month_ID Month_Na... Quarter_ID Quarter_Na... Transaction... Week_Dest Week_ID Week_Year Year_ID Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup

Total Employees COUNTROWS calculation.

1) Click here to get to the Model view.

2) Select the Total Customers measure.

3) Assign the display folder Scalar Measures.

The screenshot shows the Power BI Desktop interface with the 'Model' tab selected in the ribbon. On the left, the Data Model view displays various tables: Calendar, Customer Lookup, Employee Lookup, Product Lookup, Store Lookup, Measure Table, Sales by Store, and Food Inventory. Relationships between these tables are visible as lines connecting their keys. A blue arrow points from the text '1) Click here to get to the Model view.' to the 'Model' icon in the ribbon. Another blue arrow points from the text '2) Select the Total Customers measure.' to the 'Total Customers' measure in the Properties pane. A third blue arrow points from the text '3) Assign the display folder Scalar Measures.' to the 'Scalar Measures' section in the Properties pane, where 'Total Customers' is highlighted.

Maven Routers - Analysis - Power BI Desktop

File Home Help

Get data - Excel Power BI Datasets SQL Server Enter data Recent sources - Transform Refresh data - Manage relationships Relationships Security Q&A setup Language schema - Q&A Publish Share

All tables +

Properties Fields

Name: Total Customers

Description: Enter a description

Synonyms: total customers, customers

Display folder: Scalar Measures

Scalar Measures: Total Customers (highlighted)

Is hidden: No

Formatting:

- Format: Whole number
- Percentage format: No
- Thousands separator: Yes
- Decimal places: 0

Advanced

Maven Routers - Analysis - Power BI Desktop

File Home Help

Get data - Excel Power BI Datasets SQL Server Enter data Recent sources - Transform Refresh data - Manage relationships Relationships Security Q&A Language setup Q&A Publish Share

Properties Fields

Search

Name: Total Employees

Description: Enter a description

Synonyms: total employees, employees

Display folder: Scalar Measures

Is hidden: No

Formatting

Format: Whole number

Percentage format: No

Thousands separator: Yes

Decimal places: 0

Advanced

4) Drag the Total Employees measure into the Scalar Measures folder.

```
graph LR; TotalEmployees[Total Employees] --> ScalarMeasures[Scalar Measures]
```

PRO TIP: SUM & SUMX

There are several cases where DAX evaluates a basic query using a more complex method behind the scenes. The simplified query is often called “syntax sugar”

- For example, **SUM** is read internally as **SUMX**

How it's written:

```
1 Total Sales =  
2 SUM('Sales by Store'[quantity_sold])
```

How it's interpreted by DAX:

```
1 Total Sales =  
2 SUMX(  
3   'Sales by Store'  
4   'Sales by Store'[quantity_sold])
```



HEY THIS IS IMPORTANT!

This is how *all* aggregation functions are processed internally by DAX (*SUM, AVERAGE, MAX, etc.*)

Rounding Functions

Aggregation Functions

Rounding Functions

Information Functions

Conversion Functions

Logical Functions

INT()

Rounds a number down to the nearest integer

=INT(Number)

ROUND()

Rounds a number to a specific number of digits

=ROUND(Number, NumberOfDigits)

ROUNDUP()

Rounds a number up, away from zero

=ROUNDUP(Number, NumberOfDigits)

ROUNDDOWN()

Rounds a number down, toward zero

=ROUNDDOWN(Number NumberOfDigits)

MROUND()

Rounds a number to the desired multiple

=MROUND(Number, Multiple)

TRUNC()

Truncates a number to an integer by removing the decimal part of the number

=TRUNC(Number, [NumberOfDigits])

FIXED()

Rounds number down to specified number of decimals and returns result as text

=FIXED(Number, [Decimals], [No.Commas])

CEILING()

Rounds a number up, to the nearest integer or nearest unit of significance

=CEILING(Number, Significance)

FLOOR()

Rounds a number down, toward zero, to the nearest multiple of significance

=FLOOR(Number, Significance)

Aggregation Functions

Rounding Functions

Information Functions

Conversion Functions

Logical Functions

Decimal Value: 3.12438

INT (3.12438) = 3

Useful as a component in DAX statements, like calendar tables

ROUND (3.12438, 2) = 3.12

ROUNDUP (3.12438, 2) = 3.13

ROUNDDOWN (3.12438, 2) = 3.12

Useful to specify the precision of a number, like customer age

FIXED (3.12438, 2) = 3.12

Useful when you want to convert a number to text

Time value: 9:34:14 AM

MROUND (9:34:14, "0:15") = 9:30:00 AM

Rounds the minute component up or down based on the multiple

FLOOR (9:34:14, "0:15") = 9:30:00 AM

Rounds the minute component down to the nearest multiple

CEILING (9:34:14, "0:15") = 9:45:00 AM

Rounds the minute component up to the nearest multiple

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: transaction_time Format: 133035 (hh:mm:ss) Summarization: Don't summarize Data category: Unspecified Sort by column: Date groups: Data relationships: Manage relationships New column: Relationships Calculations

transaction_id transaction_date store_id staff_id customer_id instore_yn order line_order_id product_id quantity_sold unit_price promo_item_yn

1) Click on the Data view.

2) Select the Sales by Store table.

Fields

Measure Table

- % Quantity Sold
- Customer Sales (L)
- Customer Sales L
- Orders by Female

Scalar Measures

- Tip & Best Practices

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

- customer_id
- instore_yn
- line_item_id
- order
- product_id
- promo_item_yn
- quantity_sold
- staff_id
- store_id
- transaction_id
- transaction_time
- unit_price

Table: Sales by Store (907,841 rows) Column: transaction_time (24,074 distinct values)

Maven Roasters - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: transaction_time | Format: 13:30:55 (Universal) | Summarization: Don't summarize | Data category: Uncategorized | Sort by column | Data groups | Manage relationships | New column

3) Click here to add a new column (calculated).

Fields

Search

Measure Table

- % Quantity Sold
- Customer Sales
- Customer Sales L
- Orders by Female
- Scalar Measures
- Tip & Best Practices
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store

customer_id

instore_yn

line_item_id

order

product_id

promo_item_yn

quantity_sold

staff_id

store_id

transaction_date

transaction_id

transaction_time

unit_price

Store Lookup

transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	instore_yn	order	line_item_id	product_id	quantity_sold	unit_price	promo_item_yn
181	10 January 2017	08:45:47	8	42	8800	Y	1		1	1	1	N
182	21 January 2017	09:17:14	8	42	8800	Y	1		1	1	1	N
480	28 January 2017	09:28:07	8	42	8800	Y	1		1	1	1	N
21	11 January 2017	07:33:48	8	42	8800	Y	1		1	1	1	N
281	22 February 2017	08:45:47	8	42	8800	Y	1		1	1	1	N
21	22 February 2017	07:33:48	8	42	8800	Y	1		1	1	1	N
273	28 February 2017	08:58:39	8	42	8800	Y	1		1	1	1	N
21	12 March 2017	07:33:45	8	42	8800	Y	1		1	1	1	N
291	14 March 2017	09:12:37	8	42	8800	Y	1		1	1	1	N
724	17 March 2017	09:05:44	8	42	8800	Y	1		1	1	1	N
171	19 March 2017	08:58:39	8	42	8800	Y	1		1	1	1	N
180	21 March 2017	11:19:20	8	42	8800	Y	1		1	1	1	N
24	26 March 2017	08:34:09	8	42	8800	Y	1		1	1	1	N
480	28 March 2017	09:28:07	8	42	8800	Y	1		1	1	1	N
277	31 March 2017	08:59:26	8	42	8800	Y	1		1	1	1	N
281	09 April 2017	09:40:46	8	42	8800	Y	1		1	1	1	N
724	17 April 2017	09:25:44	8	42	8800	Y	1		1	1	1	N
291	21 April 2017	09:37:14	8	42	8800	Y	1		1	1	1	N
724	22 April 2017	08:47:55	8	42	8800	Y	1		1	1	1	N
24	26 April 2017	08:54:09	8	42	8800	Y	1		1	1	1	N
231	30 April 2017	09:12:37	8	42	8800	Y	1		1	1	1	N
181	10 May 2017	08:45:47	8	42	8800	Y	1		1	1	1	N
21	12 May 2017	07:33:48	8	42	8800	Y	1		1	1	1	N
2860	16 May 2017	08:58:20	8	42	8800	Y	1		1	1	1	N
1120	16 May 2017	08:17:56	8	42	8800	Y	1		1	1	1	N
724	17 May 2017	09:05:44	8	42	8800	Y	1		1	1	1	N
171	19 May 2017	08:58:39	8	42	8800	Y	1		1	1	1	N
180	21 May 2017	11:19:20	8	42	8800	Y	1		1	1	1	N
277	28 May 2017	08:59:26	8	42	8800	Y	1		1	1	1	N
480	28 May 2017	09:28:07	8	42	8800	Y	1		1	1	1	N
181	20 June 2017	08:45:47	8	42	8800	Y	1		1	1	1	N
21	22 June 2017	07:33:48	8	42	8800	Y	1		1	1	1	N
2860	26 June 2017	08:58:20	8	42	8800	Y	1		1	1	1	N
1120	26 June 2017	08:17:56	8	42	8800	Y	1		1	1	1	N
182	21 June 2017	09:37:14	8	42	8800	Y	1		1	1	1	N
180	23 June 2017	11:19:20	8	42	8800	Y	1		1	1	1	N
24	26 June 2017	08:34:09	8	42	8800	Y	1		1	1	1	N
277	28 June 2017	08:59:26	8	42	8800	Y	1		1	1	1	N

Table: Sales by Store (907,641 rows). Column: transaction_time (26,074 distinct values)

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name Time Group Format 13:30:55 (Min:sec) Summarization Don't summarize Sort by column Data groups Manage relationships New column

Data type Structure Formatting Properties

1 Time Group =
2 MROUND(
3 'Sales by Store'[transaction_time],
4 "1:00"
5)

4) Type in the following code so that the transaction time will be rounded up by 1 hour.

5) Change the data type to time.

Fields

Measure Table
% Quantity Sold
Customer Sales
Customer Sales
Orders by Female
Scalar Measures
Tip & Best Practices
Calendar
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
customer_id
instore_yn
line_item_id
order
product_id
promo_item_yn
quantity_sold
staff_id
store_id
Time Group
transaction_date
transaction_id
transaction_time
unit_price
store_lookup

6) The Time Group values are rounded up by 1 hour as expected.

transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	instore_yn	order	line_item_id	product_id	quantity_sold	unit_price	promo_item_yn	Time Group
181	10 January 2013	08:45:47	1	42	8600	Y	11	1	32	1	32	N	07:00:00
282	23 January 2013	09:37:14	1	42	8600	Y	11	1	32	1	32	N	10:00:00
460	28 January 2013	09:28:07	1	42	8600	Y	11	1	32	1	32	N	09:00:00
21	13 January 2013	07:33:43	1	42	8600	Y	11	1	32	1	32	N	06:00:00
581	10 February 2013	08:43:47	1	42	8600	Y	11	1	32	1	32	N	07:00:00
21	12 February 2013	07:33:43	1	42	8600	Y	11	1	32	1	32	N	06:00:00
373	19 February 2013	08:58:59	1	42	8600	Y	11	1	32	1	32	N	09:00:00
21	12 March 2013	07:33:43	1	42	8600	Y	11	1	32	1	32	N	06:00:00
211	14 March 2013	09:12:37	1	42	8600	Y	11	1	32	1	32	N	09:00:00
721	17 March 2013	09:03:44	1	42	8600	Y	11	1	32	1	32	N	09:00:00
373	18 March 2013	08:58:59	1	42	8600	Y	11	1	32	1	32	N	09:00:00
190	23 March 2013	02:18:10	1	42	8600	Y	11	1	32	1	32	N	11:00:00
21	26 March 2013	08:34:09	1	42	8600	Y	11	1	32	1	32	N	07:00:00
460	28 March 2013	09:28:07	1	42	8600	Y	11	1	32	1	32	N	08:00:00
237	31 March 2013	08:39:16	1	42	8600	Y	11	1	32	1	32	N	08:00:00
281	08 April 2013	09:40:48	1	42	8600	Y	11	1	32	1	32	N	10:00:00
721	17 April 2013	09:03:44	1	42	8600	Y	11	1	32	1	32	N	09:00:00
291	21 April 2013	09:37:14	1	42	8600	Y	11	1	32	1	32	N	10:00:00
721	23 April 2013	09:47:35	1	42	8600	Y	11	1	32	1	32	N	07:00:00
24	26 April 2013	08:34:09	1	42	8600	Y	11	1	32	1	32	N	07:00:00
211	30 April 2013	09:12:37	1	42	8600	Y	11	1	32	1	32	N	09:00:00
181	20 May 2013	08:43:47	1	42	8600	Y	11	1	32	1	32	N	07:00:00
21	22 May 2013	07:33:43	1	42	8600	Y	11	1	32	1	32	N	08:00:00
2880	24 May 2013	08:58:10	1	42	8600	Y	11	1	32	1	32	N	07:00:00
2320	26 May 2013	08:17:38	1	42	8600	Y	11	1	32	1	32	N	08:00:00
721	27 May 2013	09:03:44	1	42	8600	Y	11	1	32	1	32	N	09:00:00
373	29 May 2013	08:58:59	1	42	8600	Y	11	1	32	1	32	N	09:00:00
180	30 May 2013	12:16:10	1	42	8600	Y	11	1	32	1	32	N	11:00:00

Table: Sales by Store (507,841 rows) Column: Time Group (21 distinct values)

Maven Router - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: Time Group Format: 133035 (Min:Max) Summarization: Don't summarize Data category: Unmanaged Sort by column: Sort Data groups: Groups Manage relationships: Relationships New column: Calculations

1 Time Group =
2 FLOOR(
 'Sales by Store'[transaction_time],
 "1:00"
)

1) Change the MROUND text to FLOOR so that the hours can be rounded down by 1 hour.

2) The Time Group values are rounded down by 1 hour as expected.

transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	income_yn	order	line_item_id	product_id	quantity_sold	unit_price	promo_item_yn	Time Group
151	10 January 2017	08:43:47	1	42	8600	N	1	1	82	1	2	N	08:00:00
152	11 January 2017	09:37:14	1	42	8600	N	1	1	82	1	2	N	09:00:00
460	20 January 2017	09:28:07	1	42	8600	Y	1	1	82	1	2	N	09:00:00
21	11 January 2017	07:33:43	1	42	8600	Y	1	1	82	1	2	N	07:00:00
153	20 February 2017	08:40:47	1	42	8600	Y	1	1	82	1	2	N	08:00:00
21	22 February 2017	07:33:43	1	42	8600	Y	1	1	82	1	2	N	07:00:00
373	19 February 2017	08:58:39	1	42	8600	Y	1	1	82	1	2	N	08:00:00
21	12 March 2017	07:33:43	1	42	8600	Y	1	1	82	1	2	N	07:00:00
211	14 March 2017	08:12:37	1	42	8600	Y	1	1	82	1	2	N	08:00:00
724	17 March 2017	08:05:44	1	42	8600	Y	1	1	82	1	2	N	08:00:00
873	19 March 2017	08:58:39	1	42	8600	Y	1	1	82	1	2	N	08:00:00
190	21 March 2017	11:18:10	1	42	8600	Y	1	1	82	1	2	N	11:00:00
24	26 March 2017	08:34:09	1	42	8600	Y	1	1	82	1	2	N	08:00:00
460	28 March 2017	09:28:07	1	42	8600	Y	1	1	82	1	2	N	09:00:00
277	31 March 2017	08:59:16	1	42	8600	Y	1	1	82	1	2	N	08:00:00
181	09 April 2017	09:40:46	1	42	8600	Y	1	1	82	1	2	N	09:00:00
724	17 April 2017	09:05:44	1	42	8600	Y	1	1	82	1	2	N	09:00:00
181	21 April 2017	09:37:14	1	42	8600	Y	1	1	82	1	2	N	09:00:00
724	23 April 2017	08:47:53	1	42	8600	Y	1	1	82	1	2	N	08:00:00
24	26 April 2017	08:34:09	1	42	8600	Y	1	1	82	1	2	N	08:00:00
211	30 April 2017	09:12:37	1	42	8600	Y	1	1	82	1	2	N	09:00:00
181	20 May 2017	08:40:47	1	42	8600	Y	1	1	82	1	2	N	08:00:00
21	22 May 2017	07:33:43	1	42	8600	Y	1	1	82	1	2	N	07:00:00
2080	24 May 2017	08:58:10	1	42	8600	Y	1	1	82	1	2	N	08:00:00
1220	26 May 2017	08:17:36	1	42	8600	Y	1	1	82	1	2	N	08:00:00
724	27 May 2017	08:05:44	1	42	8600	Y	1	1	82	1	2	N	08:00:00
873	29 May 2017	08:58:39	1	42	8600	Y	1	1	82	1	2	N	08:00:00
180	30 May 2017	11:18:10	1	42	8600	Y	1	1	82	1	2	N	11:00:00

Table: Sales by Store (907,841 rows) Column: Time Group (29 distinct values)

Fields

Measure Table
% Quantity Sold
Customer Sales
Customer Sales L...
Orders by Female
Scalar Measures
Tip & Best Practices
Calendar
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
customer_id
instore_yn
line_item_id
order
product_id
promo_item_yn
quantity_sold
staff_id
store_id
Time Group
transaction_date
transaction_id
transaction_time
unit_price
Store Lookup

ASSIGNMENT Rounding Functions

KEY OBJECTIVES

- 1) Add a new column in your Customers table to calculate age & use a rounding function to make sure that ages are defined properly & formatted as whole numbers.

File Home Help Table tools Column tools

Name: Current Age
Data type: Whole number

Format: General
\$ - % , . Auto

Summarization: Don't summarize
Data category: Uncategorized

Sort by column
Data groups
Manage relationships
New column
Calculated

Fields

Search

Measure Table
% Quantity Sold t...
Customer Sales L...
Customer Sales L...
Orders by Female...
Scalar Measures
Tip & Best Practices

Calendar
Customer Lookup
birth_year
birthdate
Current_Age
customer_email
customer_first_na...
customer_id
Customer_JobC
gender
home_store
loyalty_card_num...
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
Store Lookup

1 Current Age =

2 FLOOR(

3 DATEDIFF(

4 'Customer Lookup'[birthdate],

5 TODAY(),

6 DAY

7) / 365.25,

8 1

9)

1) Objective 1 to add a new column in your Customers table to calculate age & use a rounding function to make sure that ages are defined properly & formatted as whole numbers.. Calculated column to calculate Current age using birthdate & Today with DATEDIFF & FLOOR with 1 decimal precision.

customer_id	home_store	customer_firstname	customer_email	customer_surname	loyalty_card_number	birthdate	gender	birth_year	Current_Age
301	J	Alice Rivers	louise@elquiem.us	04 January 2017	021-443-0682	13 May 1950	F	1950	70
302	J	Sachin Wall	yvette@erat.org	06 January 2017	433-102-9277	29 June 1950	F	1950	70
303	J	Kaya Hampton	Martina@imperiusada.com	08 January 2017	479-763-8984	14 August 1950	F	1950	70
304	J	Belle Hayes	Sierra@bubbles.net	10 January 2017	830-000-5889	07 September 1950	F	1950	70
305	J	Brooke Munoz	William@moneys.com	12 January 2017	873-882-0754	16 November 1950	F	1950	70
306	J	Tanisha Wolf	Clinton@verus.us	14 January 2017	368-284-2140	02 January 1951	F	1951	69
307	J	Quintessa Franklin	Audrey@mathics.net	16 January 2017	977-781-9813	17 February 1951	F	1951	69
308	J	Charissa Cobb	Regan@nivamus.net	18 January 2017	823-483-1818	04 April 1951	F	1951	68
309	J	Susan Metha	Cody@tlocaduct.edu	20 January 2017	726-654-4314	21 May 1951	F	1951	68
310	J	Cynthia Huney	Miriam@eru.edu	22 January 2017	369-573-4564	07 July 1951	F	1951	68
311	J	Melodie Dawson	Stacey@nordix.us	24 January 2017	654-557-8163	22 August 1951	F	1951	68
312	J	Charonette Weiss	Phelan@runum.org	26 January 2017	342-155-7542	08 October 1951	F	1951	68
313	J	Melyssa Jones	Dellian@Dvi.edu	28 January 2017	823-893-0916	21 November 1951	F	1951	68
314	J	Yuri Jacobson	Shea@unite.gov	30 January 2017	873-921-0474	09 January 1952	F	1952	68
315	J	Kellie Barton	Hyacinth@urua.edu	01 February 2017	238-217-0554	25 February 1952	F	1952	68
316	J	Camille Giacci	Hedley@adapting.gov	04 February 2017	448-528-3198	11 April 1952	F	1952	68
317	J	Keagan Roberts	Echo@integragedu	06 February 2017	324-063-4567	28 May 1952	F	1952	68
318	J	Chelsea Riva	Willow@sevn.net	08 February 2017	288-183-8654	10 July 1952	F	1952	68
319	J	Leanne Benson	Laura@leeland.net	10 February 2017	689-948-0110	29 August 1952	F	1952	68

Table: Customer Lookup (2,251 rows) Column: Current_Age (52 distinct values)

**Key objective 1 is now complete.
Add a new column in your
Customers table to calculate age &
use a rounding function to make
sure that ages are defined properly
& formatted as whole numbers.**

Information Functions

Aggregation
Functions

Rounding
Functions

Information
Functions

Conversion
Functions

Logical
Functions

ISBLANK()

Checks whether a value is blank, and returns TRUE or FALSE

=**ISBLANK**(Value)

ISERROR()

Checks whether a value is an error, and returns TRUE or FALSE

=**ISERROR**(Value)

ISLOGICAL()

Checks whether a value is a logical value (TRUE or FALSE), and returns TRUE or FALSE

=**ISLOGICAL**(Value)

ISNUMBER()

Checks whether a value is a number, and returns TRUE or FALSE

=**ISNUMBER**(Value)

ISNONTEXT()

Checks whether a value is not text (blank cells are not text), and returns TRUE or FALSE

=**ISNONTEXT**(Value)

ISTEXT()

Checks whether a value is text, and returns TRUE or FALSE

=**ISTEXT**(Value)

Maven Roadmap - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: ISNUMBER
Data type: True/False
Structure: Row values
Summarization: Don't summarize
Data category: Unspecified
Sort by column: Sort
Data groups: Groups
Manage relationships: Relationships
New column: Calculations

1 ISNUMBER =

2 ISNUMBER('Calendar'[Week_ID])

Sort ascending
Sort descending
Clear sort
Clear filter
Clear all filters
(Selected all)
True
OK Cancel

1) Calculated column to test that the Week_ID all rows information evaluates to a number & not text values. Can confirm all values are numbers.

Transaction Date	Data ID	Week_ID	Week Desc	Month ID	Month Name	Quarter ID	Quarter Name	Year ID	Week Year	ISNUMBER
01/07/2017	20170701	26	Week 26	7	July	1	Q3	2017	Week 26-2017	True
02/07/2017	20170702	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
03/07/2017	20170703	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
04/07/2017	20170704	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
05/07/2017	20170705	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
06/07/2017	20170706	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
07/07/2017	20170707	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
08/07/2017	20170708	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
09/07/2017	20170709	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
10/07/2017	20170710	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
11/07/2017	20170711	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
12/07/2017	20170712	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
13/07/2017	20170713	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
14/07/2017	20170714	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
15/07/2017	20170715	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
16/07/2017	20170716	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
17/07/2017	20170717	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
18/07/2017	20170718	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
19/07/2017	20170719	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
20/07/2017	20170720	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
21/07/2017	20170721	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
22/07/2017	20170722	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
23/07/2017	20170723	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
24/07/2017	20170724	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
25/07/2017	20170725	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
26/07/2017	20170726	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
27/07/2017	20170727	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
28/07/2017	20170728	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
29/07/2017	20170729	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
30/07/2017	20170730	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True
31/07/2017	20170731	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True
01/08/2017	20170801	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True
02/08/2017	20170802	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True
03/08/2017	20170803	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True
04/08/2017	20170804	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True

Table: Calendar (350 rows) Column: ISNUMBER (1 distinct values)

Fields

- Measure Table
- % Quantity Sold L...
- Customer Sales L...
- Customer Sales L...
- Orders by Female...
- Scalar Measures
- Tip & Best Practices
- Calendar
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_ID
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store
- Store Lookup

Maven Roasters - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: ISNUMBER
Data type: True/False

Σ Summarization: Don't summarize
Data category: Unsegmented

Sort by column
Data group
Manage relationships
New column
Calculated columns

Structure Formatted Properties Sort Group Refresh Calculations

1 ISNUMBER =

2 ISNUMBER('Calendar'[Week_Desc])

Fields

Search

Measure Table

% Quantity Sold

Customer Sales

Customer Sales L

Orders by Female

Scalar Measures

Tip & Best Practices

Calendar

Date_ID

ISNUMBER

Month_ID

Month_Name

Quarter_ID

Quarter_Name

Transaction_Date

Week_Desc

Week_ID

Year_ID

Week_Year

OK Cancel

Sort ascending
Sort descending
Clear sort
Clear filter
Clear all filters
(Selected All)
False

1) Amend to Week_Desc as these values are text based.

2) Can confirm all values are text i.e. False output from test.

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	ISNUMBER
01/07/2017	20170701	26	Week 26	7	July	1	Q3	2017	Week 26-2017	True
02/07/2017	20170702	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
03/07/2017	20170703	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
04/07/2017	20170704	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
05/07/2017	20170705	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
06/07/2017	20170706	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
07/07/2017	20170707	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
08/07/2017	20170708	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
09/07/2017	20170709	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
10/07/2017	20170710	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
11/07/2017	20170711	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
12/07/2017	20170712	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
13/07/2017	20170713	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
14/07/2017	20170714	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
15/07/2017	20170715	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
16/07/2017	20170716	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
17/07/2017	20170717	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
18/07/2017	20170718	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
19/07/2017	20170719	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
20/07/2017	20170720	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
21/07/2017	20170721	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
22/07/2017	20170722	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
23/07/2017	20170723	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
24/07/2017	20170724	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
25/07/2017	20170725	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
26/07/2017	20170726	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
27/07/2017	20170727	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
28/07/2017	20170728	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
29/07/2017	20170729	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
30/07/2017	20170730	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True
31/07/2017	20170731	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True
01/08/2017	20170801	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True
02/08/2017	20170802	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True
03/08/2017	20170803	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True
04/08/2017	20170804	31	Week 31	7	August	1	Q3	2017	Week 31-2017	True

Table: Calendar (850 rows) Column: ISNUMBER (1 distinct value)

Maven Roadmap - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: ISNUMBER
Data type: True/False
Summarization: Don't summarize
Sort by column: Sort
Data category: Unassigned
Manage relationships
New column
Structure
Power Query
Properties
Relationships
Calculated columns

1 ISNUMBER =
2 ISTEXT('Calendar'[Week_Desc])

Fields

Search

Measure Table
% Quantity Sold L...
Customer Sales L...
Customer Sales L...
Orders by Product...
Scalar Measures
Tip & Best Practices

Calendar
Date_ID
ISNUMBER
Month_ID
Month_Name
Quarter_ID
Quarter_Name
Transaction_Date
Week_Desc
Week_ID
Week_Year
Year_ID
Year_Name

Sort ascending
Sort descending
Clear sort
Clear filter
Clear all filters
(Selected all)
True

OK Cancel

1) Calculated column to test that the Week_Desc all rows information evaluates to a text & not number values. Can confirm all values are text.

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	ISNUMBER
01/07/2017	20170701	26	Week 26	7	July	1	Q3	2017	Week 26-2017	True
02/07/2017	20170702	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
03/07/2017	20170703	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
04/07/2017	20170704	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
05/07/2017	20170705	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
06/07/2017	20170706	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
07/07/2017	20170707	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
08/07/2017	20170708	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True
09/07/2017	20170709	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
10/07/2017	20170710	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
11/07/2017	20170711	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
12/07/2017	20170712	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
13/07/2017	20170713	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
14/07/2017	20170714	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
15/07/2017	20170715	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True
16/07/2017	20170716	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
17/07/2017	20170717	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
18/07/2017	20170718	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
19/07/2017	20170719	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
20/07/2017	20170720	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
21/07/2017	20170721	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
22/07/2017	20170722	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True
23/07/2017	20170723	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
24/07/2017	20170724	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
25/07/2017	20170725	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
26/07/2017	20170726	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
27/07/2017	20170727	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
28/07/2017	20170728	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
29/07/2017	20170729	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True
30/07/2017	20170730	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True
31/07/2017	20170731	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True
01/08/2017	20170801	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True
02/08/2017	20170802	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True
03/08/2017	20170803	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True
04/08/2017	20170804	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True

Table: Calendar (350 rows) Column: ISNUMBER (1 distinct values)

Conversion Functions

Aggregation Functions

Rounding Functions

Information Functions

Conversion Functions

Logical Functions

CURRENCY()

Evaluates the argument and returns the result as a currency data type

=CURRENCY(Value)

FORMAT()

Converts a value to text in the specified number format

=FORMAT(Value, Format)

DATE()

Returns the specified date in datetime format

=DATE(Year, Month, Day)

TIME()

Converts hours, minutes, and seconds given as numbers to a time in datetime format

=TIME(Hours, Minute, Second)

DATEVALUE()

Converts a date in the form of text to a date in datetime format

=DATEVALUE(DateText)

VALUE()

Converts a text string that represents a number to a number

=VALUE(Text)

ASSIGNMENT Conversion Functions

KEY OBJECTIVES

- 1) Use a conversion function to modify the [Cost] measure & ensure that values are always displayed as currency (without simply changing the format).
- 2) Create a new calculated column to convert Transaction Date to yyyy-mm-dd format.

Maven Robotics - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name Total - Revenue Format Currency Summation Sum Data category Uncategorized Sort by column Data groups Manage relationships New column

Data type Fixed decimal number \$ - % Auto Sort Group Relationships Calculations

Structure Formatting Properties

1 Total - Revenue = 2 CURRENCY(3 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price] 4)

Key objective 1 is now complete. Use a conversion function to modify the [Cost] measure & ensure that values are always displayed as currency (without simply changing the format).

1) Objective 1 to create a calculated column to calculate Total Revenue, quantity sold multiplied by unit price & output provided in £ currency.

2) Expected values in £ currency using a simple conversion function.

transaction_id	transaction_date	transaction_time	store_id	staff_id	customer_id	bucause_yn	order	line_item_id	product_id	quantity_sold	unit_price	promo_item_yn	Time Group	Total - Revenue
191	22 January 2017	08:45:47	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
192	22 January 2017	09:17:14	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
460	28 January 2017	09:28:07	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
21	31 January 2017	07:33:43	8	42	8800	Y	1		12	1	1	N	07:00:00	£1
181	10 February 2017	08:45:47	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
21	12 February 2017	07:33:43	8	42	8800	Y	1		12	1	1	N	07:00:00	£1
371	14 February 2017	08:58:59	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
21	13 March 2017	07:33:43	8	42	8800	Y	1		12	1	1	N	07:00:00	£1
111	14 March 2017	08:12:37	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
226	17 March 2017	09:00:44	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
373	19 March 2017	08:58:59	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
190	21 March 2017	11:19:10	8	42	8800	Y	1		12	1	1	N	11:00:00	£1
24	26 March 2017	08:34:09	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
460	28 March 2017	09:28:07	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
277	31 March 2017	08:39:16	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
181	09 April 2017	09:40:46	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
128	17 April 2017	09:25:44	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
191	21 April 2017	09:37:14	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
723	22 April 2017	08:47:33	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
24	26 April 2017	08:34:09	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
211	30 April 2017	09:17:37	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
181	09 May 2017	08:45:47	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
21	12 May 2017	07:33:43	8	42	8800	Y	1		12	1	1	N	07:00:00	£1
2060	16 May 2017	08:58:10	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
1520	16 May 2017	08:17:16	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
726	17 May 2017	09:05:44	8	42	8800	Y	1		12	1	1	N	09:00:00	£1
173	19 May 2017	08:58:39	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
190	23 May 2017	11:39:10	8	42	8800	Y	1		12	1	1	N	11:00:00	£1
277	26 May 2017	08:39:16	8	42	8800	Y	1		12	1	1	N	08:00:00	£1
460	28 May 2017	09:28:07	8	42	8800	Y	1		12	1	1	N	09:00:00	£1

Table: Sales by Store (907,641 rows) Column: Total - Revenue (73 distinct values)

Fields

Measure Table % Quantity Sold Y Customer Sales L... Customer Sales L... Orders by Female Scalar Measures Tip & Best Practices Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store customer_id instore_yn line_item_id order product_id promo_item_yn quantity_sold staff_id store_id Time Group Total - Revenue transaction_id transaction_time unit_price Store Lookup

Maven Roasters - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name Date Format (yyyy-mm-dd) Data type Text Summarization Don't summarize Data category Uncategorized Sort by column Data groups Manage relationships column Calculations

1 Date Format (yyyy-mm-dd)
2 FORMAT(
3 'Calendar'[Transaction_Date],
4 "yyyy-mm-dd"
5)

Fields

Search

Measure Table
Calendar
Date Format (yyyy-mm-dd)
Data_ID
ISNUMBER
Month_ID
Month_Name
Quarter_ID
Quarter_Name
Year_ID
Week_Year
TRANSACTION
Transaction_Date
Week_Day
Week_ID
Month_ID
Month_Name
Quarter_ID
Quarter_Name
Year_ID
Week_Year
TRANSACTION
Date Format (yyyy-mm-dd)

2) Expected converted date values shown in "yyyy-mm-dd" format.

Key objective 2 is now complete. Create a new calculated column to convert Transaction Date to yyyy-mm-dd format.

1) Objective 2 to create a new calculated column to convert Transaction Date to yyyy-mm-dd format. Calculated column to change the format of the transaction date to "yyyy-mm-dd".

Transaction_Date	Date_ID	Week_ID	Week_Day	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	TRANSACTION	Date Format (yyyy-mm-dd)
01/07/2017	20170701	26	Week 26	3	July	1	Q3	2017	Week 26-2017	True	2017-07-03
02/07/2017	20170702	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-02
03/07/2017	20170703	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-03
04/07/2017	20170704	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-04
05/07/2017	20170705	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-05
06/07/2017	20170706	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-06
07/07/2017	20170707	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-07
08/07/2017	20170708	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-08
09/07/2017	20170709	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-09
10/07/2017	20170710	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-10
11/07/2017	20170711	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-11
12/07/2017	20170712	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-12
13/07/2017	20170713	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-13
14/07/2017	20170714	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-14
15/07/2017	20170715	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-15
16/07/2017	20170716	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-16
17/07/2017	20170717	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-17
18/07/2017	20170718	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-18
19/07/2017	20170719	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-19
20/07/2017	20170720	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-20
21/07/2017	20170721	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-21
22/07/2017	20170722	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-22
23/07/2017	20170723	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-23
24/07/2017	20170724	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-24
25/07/2017	20170725	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-25
26/07/2017	20170726	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-26
27/07/2017	20170727	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-27
28/07/2017	20170728	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-28

Table: Calendar (250 rows) Column: Date Format (yyyy-mm-dd) (250 distinct values)

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name Cost (CURRENCY) Format Currency Data category Uncategorized

Home table Measure Table \$ - % , . Auto

Measure Formatting Properties

1 Cost (CURRENCY) =
2 CURRENCY([Cost])

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females (All)
- Cost (All)
- Cost (CURRENCY) (All)
- Customer Sales (All)
- Transaction Date (All)
- Add data fields here

Values

- Customer Sales
- % Quantity Sold to Females
- Cost
- Cost (CURRENCY)

Filters on this page

Drill through

Cross-report

Keep all filters

Off

Add drill-through fields here

1) Cost (CURRENCY) measure to convert the cost measure to currency rather than a number output.

2) Expected Cost column conversion to £ currency format.

Transaction Date	Customer Sales	% Quantity Sold to Females	Cost	Cost (CURRENCY)
01/01/2017	2308.20	44.51%	632.79	£632.795
02/01/2017	2403.35	46.48%	636.71	£636.711
03/01/2017	2163.00	44.47%	671.09	£671.0925
04/01/2017	2230.10	43.13%	578.19	£578.19
05/01/2017	2418.85	42.80%	630.60	£630.6
06/01/2017	2273.85	38.99%	599.98	£599.985
07/01/2017	2787.00	23.92%	714.67	£714.6725
08/01/2017	2801.53	23.20%	678.64	£678.6435
09/01/2017	2876.61	16.85%	680.01	£680.029
10/01/2017	2683.45	21.29%	689.85	£689.845
11/01/2017	2553.75	27.62%	648.23	£648.2775
12/01/2017	2327.70	34.31%	692.74	£692.74
13/01/2017	3031.60	22.74%	779.32	£779.32
14/01/2017	2682.51	14.92%	682.77	£682.772
15/01/2017	3187.71	11.47%	795.98	£795.982
16/01/2017	2829.18	10.88%	758.66	£758.667
17/01/2017	3285.80	12.73%	784.84	£784.84
18/01/2017	2733.99	10.28%	686.13	£686.127
19/01/2017	2913.68	8.51%	739.56	£739.5585
Total	4252704.88	21.71%	1089.17627	£1,089,176.2695

Maven Robotics - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Cost (CURRENCY) Format: Currency Data category: Uncategorized

Home table: Measure Table \$ - % # . Auto

Structure, Formatting, Properties, Calculations

1 Cost (CURRENCY) =
2 ROUND(
3 CURRENCY([Cost])
4 ,2
5)

1) Rounding function used for 2 decimal places.

2) Expected Cost column conversion to £ currency format & to 2 decimal places.

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)

Customer Sales (Last Year)

Customer Sales LY (ISBLA...)

Orders by Females (Wrong)

Scalar Measures

Tip & Best Practices

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

Rows: Transaction_Date

Columns: Add data fields here

Values: Customer Sales, % Quantity Sold to Females, Cost, Cost (CURRENCY)

Filters on this page: Add data fields here

Drill through: Cross-report, Keep all filters, On

Add drill-through fields here

Page 1

Page 1 of 1

Logical Functions: SWITCH

Aggregation Functions

Rounding Functions

Information Functions

Conversion Functions

Logical Functions

IF()

Checks if a given condition is met, and returns one value if the condition is TRUE, and another if the condition is FALSE

=**IF(LogicalTest, ResultIfTrue, [ResultIfFalse])**

AND()

Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE, otherwise returns FALSE

=**AND(Logical1, Logical2)**

OR()

Checks whether one of the arguments is TRUE to return TRUE, and returns FALSE if both arguments are FALSE

=**OR(Logical1, Logical2)**

Note: Use the && and || operators if you want to include more than two conditions!

Aggregation
Functions

Rounding
Functions

Information
Functions

Conversion
Functions

Logical
Functions

SWITCH()

Evaluates an expression against a list of values and returns one of multiple possible expressions

=SWITCH(Expression, Value1, Result1, ..., [Else])

Any DAX expression that returns a single scalar value, evaluated multiples times

Examples:

- Calendar[Month_ID]
- Product Lookup[product_group]

List of values produced by the expression, each paired with a result to return for rows/cases that match

Examples:

- =SWITCH(Calendar[Month_ID],
1, "January",
2, "February", etc.

Value returned if the expression doesn't match any value argument



PRO TIP

SWITCH (TRUE) is a common Pattern to replace nested IF statements

Maven Readers - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: Year_Half
Data type: Text
Summarization: Don't summarize
Data category: Uncategorized
Sort by column
Data groups
Manage relationships
New column

Structure Specifying Properties

Auto recovery contains some recovered files that haven't been opened.

View recovered files X

Fields

Search

Measure Table

Calendar

Date_ID

iGulfUSA

Month_ID

Month_Name

Quarter_ID

Quarter_Name

Transaction_Date

Week_Desc

Week_ID

Week_Year

YTD

Date Format (yyyy-mm-dd)

Year_Half

1) Calculated column to calculate Year Half using the SWITCH function against the Calendar table via the Month ID field

2) Expected results for first & second half output.

Sort ascending

Sort descending

Clear sort

Clear filters

Clear columns

Text filters

Search

(Select all)

1H

2H

OK Cancel

Transaction Date Date_ID Week_ID Week_Desc Month_ID Month_Name Quarter_ID Quarter_Name Year_ID Week_Year YTD Date Format (yyyy-mm-dd) Year_Half

01/07/2017 201707001 26 Week 26 7 July 1 Q3 2017 Week 26-2017 True 2017-07-01 1H

02/07/2017 201707002 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-02 1H

03/07/2017 201707003 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-03 1H

04/07/2017 201707004 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-04 1H

05/07/2017 201707005 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-05 1H

06/07/2017 201707006 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-06 1H

07/07/2017 201707007 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-07 1H

08/07/2017 201707008 27 Week 27 7 July 1 Q3 2017 Week 27-2017 True 2017-07-08 1H

09/07/2017 201707009 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-09 2H

10/07/2017 201707010 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-10 2H

11/07/2017 201707011 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-11 2H

12/07/2017 201707012 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-12 2H

13/07/2017 201707013 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-13 2H

14/07/2017 201707014 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-14 2H

15/07/2017 201707015 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-15 2H

16/07/2017 201707016 28 Week 28 7 July 1 Q3 2017 Week 28-2017 True 2017-07-16 2H

17/07/2017 201707017 29 Week 29 7 July 1 Q3 2017 Week 29-2017 True 2017-07-17 2H

18/07/2017 201707018 29 Week 29 7 July 1 Q3 2017 Week 29-2017 True 2017-07-18 2H

19/07/2017 201707019 29 Week 29 7 July 1 Q3 2017 Week 29-2017 True 2017-07-19 2H

Maven Roosters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill Table tools

Name: Calendar

Mark as date table Manage relationships New measure Quick New measure column New table

Measure Calculations Relationships Calculations

Auto recovery contains some recovered files that haven't been opened.

Transaction_Date Customer Sales % Quantity Sold to Females Cost

20/01/2017	2007.10	16.90%	540.85
20/01/2017	2040.75	19.84%	551.05
04/02/2017	2201.40	16.20%	577.62
04/02/2017	2220.10	43.37%	378.19
11/01/2017	2334.13	21.24%	580.71
06/01/2017	2271.89	38.99%	599.99
12/01/2017	2327.70	24.51%	603.74
28/02/2017	2311.10	21.87%	605.80
17/02/2017	2300.73	10.87%	606.17
03/02/2017	2304.70	44.52%	607.85
22/01/2017	2367.33	14.40%	613.09
03/01/2017	2418.85	42.80%	630.60
30/01/2017	2476.41	18.47%	631.73
02/01/2017	2409.33	46.40%	636.71
11/01/2017	2355.73	27.62%	649.28
01/02/2017	2466.20	42.71%	649.48
11/02/2017	2326.74	28.95%	652.43
09/02/2017	2810.83	19.84%	652.57
01/01/2017	2356.20	44.51%	652.79
Total	4252704.88	21.71%	1,089,176.27

4) Click here so that the H1 & H2 views collapse within the visual.

1) Add in the Cost measure to the visual & change the table to a matrix.

2) This is the matrix visual.

3) Add the Year ID & Year Half fields to the Rows.

SWITCH makes the code generated in this example less complex to write in comparison to a if statement which would took ages to generate. Plus the code is easily readable in this example.

View recovered files X

Visualizations >

Fields

Search

Filters on this visual

Cost is (All)

Year_Half is (All)

Year_ID is (All)

Add data fields here

Columns

Values

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Drill through

Cross-report

On —

Keep all filters

Add drill-through fields here

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

Page 1 +

Page 1 of 1

Logical Functions: COALESCE

Aggregation Functions

Rounding Functions

Information Functions

Conversion Functions

Logical Functions

COALESCE()

Returns the first argument that does not evaluate to BLANK. If all arguments evaluate to BLANK, BLANK is returned.

=COALESCE(Expression1, Expression2, [...])

Any value or expression that returns a scalar value

Examples:

- COALESCE(
 SUM('Sales by Store'[quantity_sold]),
 0)
- COALESCE(
 [Yesterday Customer Revenue],
 "-")

PRO TIP:

COALESCE replaces the IF + ISBLANK pattern, makes your code more readable, and can be optimized by the engines



Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table spots Measure tools

Name: Customer Sales LY (COALESCE)
Home table: Measure Table
Measure Type: % - %
Data Category: Unassigned

New Quick measure measure

Source: Properties Calculations

Auto recovery contains some recovered files that haven't been opened.

View recovered files

Visualizations > Fields

Customer Sales LY (COALESCE) =
VAR Customer_Sales_LY =
CALCULATE(
[Customer Sales],
DATEADD(
'Calendar'[Transaction_Date],
-1,
Year)
)
RETURN
COALESCE(
Customer_Sales_LY,
"-")

1) Code for the Customer Sales LY COALESCE measure. For blank Customer Sales entries a “-” will be applied by default.

Measure Table
% Quantity Sold to Females
Customer Sales (Last Year)
Customer Sales LY (COALESCE)
Customer Sales LY (ISBLANK)
Orders by Females (Wrong)
Scalar Measures
Total Customers
Total Employees
Tip & Best Practices
Cost
Customer Sales
Profit
Calendar
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
Store Lookup

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

Queries Calculations Share

Customer Sales LY (COALESCE)

Transaction_Date	Customer Sales	Customer Sales LY (COALESCE)	% Quantity Sold to Females
17/12/2017	\$5,144.80	-	11.79%
18/12/2017	\$5,863.78	-	11.43%
19/12/2017	\$6,308.61	-	10.75%
20/12/2017	\$5,506.46	-	11.81%
21/12/2017	\$5,867.91	-	15.00%
22/12/2017	\$5,632.81	-	14.44%
23/12/2017	\$5,513.01	-	12.59%
24/12/2017	\$6,160.83	-	14.76%
25/12/2017	\$5,876.95	-	14.42%
26/12/2017	\$5,733.48	-	14.42%
27/12/2017	\$5,952.53	-	13.48%
28/12/2017	\$4,773.40	-	22.64%
29/12/2017	\$4,479.15	-	22.21%
30/12/2017	\$5,184.03	-	19.69%
31/12/2017	\$5,037.98	-	21.95%
01/01/2018	\$4,300.80	2,508.30	44.11%
02/01/2018	\$4,071.03	2,403.31	44.22%
03/01/2018	\$4,203.79	2,565.00	42.79%
04/01/2018	\$4,013.13	2,220.10	44.57%
05/01/2018	\$4,333.60	2,418.85	45.83%
06/01/2018	\$3,845.90	2,273.85	38.80%
07/01/2018	\$4,379.18	2,787.00	22.09%
08/01/2018	\$5,209.79	2,608.53	19.82%
09/01/2018	\$5,086.96	2,676.61	16.47%
10/01/2018	\$4,364.03	2,685.65	23.71%
11/01/2018	\$4,231.86	2,353.73	26.23%
12/01/2018	\$4,310.90	2,327.70	24.24%
13/01/2018	\$4,961.34	3,013.60	22.64%
Total	\$4,252,704.83	2,236,644.83	21.71%

Year_ID Cost

Year_ID	Cost
2017	429,835.22
TH	179,450.01
ZH	250,385.21
2018	490,813.53
TH	232,129.49
ZH	258,683.64
2019	168,527.52
TH	168,527.52
Total	1,089,176.27

Filters

Search

Filters on this visual:

- % Quantity Sold to Females (All)
- Customer Sales (All)
- Customer Sales LY (COALESCE) (All)
- Transaction Date (All)

Add data fields here

Filters on this page:

- Customer Sales
- Customer Sales LY (COALESCE)
- % Quantity Sold to Females

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Visualizations

Measure Table, Calendar, Customer Lookup, Employee Lookup, Food Inventory, Product Lookup, Sales by Store, Store Lookup

Fields

Search

Measure Table, Calendar, Customer Lookup, Employee Lookup, Food Inventory, Product Lookup, Sales by Store, Store Lookup

Rows: Transaction Date

Columns: Add data fields here

Values: Customer Sales, Customer Sales LY (COALESCE), % Quantity Sold to Females

Filters on all pages: Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

3) Expected results “-” applied for blank Customer Sales.

2) Add in the Customer Sales LY (COALESCE) measure & amend other fields accordingly

In this example we took two measures & combined those into one measure & also wrote the code in a more efficient way. The code can now be optimised by the DAX engine using COALESCE & variables.

ASSIGNMENT Logical Functions

KEY OBJECTIVES

- 1) Create a calculated column using **variables** & **SWITCH(TRUE())** to create a single column containing both quarter & year (i.e. Q1-2020, Q2-2020, etc.).

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: Quarter & Year
Data type: Text

Summarization: Don't summarize
Data category: Uncategorized

Sort by column
Data groups
Manage relationships
New column
Calculations

Fields

Search

Measure Table
Calendar
Data Format (yyyy-mm-dd)
Date_ID
IDNUMBER
Month_ID
Month_Name
Quarter & Year
Quarter_ID
Quarter_Name
Transaction_Date
Week_Desc
Week_ID
Week_Year
Year_Half
Year_ID
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
Store Lookup

1 Quarter & Year =
 2 VAR Q1 = 'Calendar'[Month_ID] IN {1,2,3}
 3 VAR Q2 = 'Calendar'[Month_ID] IN {4,5,6}
 4 VAR Q3 = 'Calendar'[Month_ID] IN {7,8,9}
 5 VAR Q4 = 'Calendar'[Month_ID] IN {10,11,12}
 6
 7 RETURN
 8 SWITCH(
 9 TRUE(),
 10 Q1, "Q1" & "-" & 'Calendar'[Year_ID],
 11 Q2, "Q2" & "-" & 'Calendar'[Year_ID],
 12 Q3, "Q3" & "-" & 'Calendar'[Year_ID],
 13 Q4, "Q4" & "-" & 'Calendar'[Year_ID],
 14 "
 15)

1) Objective 1 to create a calculated column using variables & SWITCH(TRUE()) to create a single column containing both quarter & year (i.e. Q1-2020, Q2-2020, etc.).

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	IDNUMBER	Date Format (yyyy-mm-dd)	Year_Half	Quarter & Year
02/07/2017	20170701	26	Week 26	7	July	3	Q3	2017	Week 26-2017	Type	2017-07-03	2H	Q3-2017
02/07/2017	20170702	27	Week 27	7	July	3	Q3	2017	Week 27-2017	Type	2017-07-04	2H	Q3-2017
03/07/2017	20170703	27	Week 27	7	July	3	Q3	2017	Week 27-2017	Type	2017-07-05	2H	Q3-2017
04/07/2017	20170704	27	Week 27	7	July	3	Q3	2017	Week 27-2017	Type	2017-07-06	2H	Q3-2017
05/07/2017	20170705	27	Week 27	7	July	3	Q3	2017	Week 27-2017	Type	2017-07-07	2H	Q3-2017

Table: Calendar (250 rows) Column: Quarter & Year (10 distinct values)

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name: Calendar

Mark as date table = Manage relationships New measure Quick New measure column New table Calculations

Rowsource: X ✓

Fields

Search

Measure Table

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

Key objective 1 is now complete. Create a calculated column using variables & SWITCH(TRUE()) to create a single column containing both quarter & year (i.e. Q1-2020, Q2-2020, etc.).

Table: Calendar (850 rows)

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	WEEKNUMBER	Date Format (yyyy-mm-dd)	Year Half	Quarter & Year
01/07/2017	201707001	26	Week 26		July	1	Q3		2017 Week 26-2017	True	2017-07-01	2H	Q3-2017
02/07/2017	201707002	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-02	2H	Q3-2017
03/07/2017	201707003	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-03	2H	Q3-2017
04/07/2017	201707004	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-04	2H	Q3-2017
05/07/2017	201707005	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-05	2H	Q3-2017
06/07/2017	201707006	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-06	2H	Q3-2017
07/07/2017	201707007	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-07	2H	Q3-2017
08/07/2017	201707008	27	Week 27		July	1	Q3		2017 Week 27-2017	True	2017-07-08	2H	Q3-2017
09/07/2017	201707009	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-09	2H	Q3-2017
10/07/2017	201707010	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-10	2H	Q3-2017
11/07/2017	201707011	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-11	2H	Q3-2017
12/07/2017	201707012	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-12	2H	Q3-2017
13/07/2017	201707013	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-13	2H	Q3-2017
14/07/2017	201707014	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-14	2H	Q3-2017
15/07/2017	201707015	28	Week 28		July	1	Q3		2017 Week 28-2017	True	2017-07-15	2H	Q3-2017
16/07/2017	201707016	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-16	2H	Q3-2017
17/07/2017	201707017	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-17	2H	Q3-2017
18/07/2017	201707018	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-18	2H	Q3-2017
19/07/2017	201707019	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-19	2H	Q3-2017
20/07/2017	201707020	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-20	2H	Q3-2017
21/07/2017	201707021	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-21	2H	Q3-2017
22/07/2017	201707022	29	Week 29		July	1	Q3		2017 Week 29-2017	True	2017-07-22	2H	Q3-2017
23/07/2017	201707023	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-23	2H	Q3-2017
24/07/2017	201707024	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-24	2H	Q3-2017
25/07/2017	201707025	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-25	2H	Q3-2017
26/07/2017	201707026	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-26	2H	Q3-2017
27/07/2017	201707027	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-27	2H	Q3-2017
28/07/2017	201707028	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-28	2H	Q3-2017
29/07/2017	201707029	30	Week 30		July	1	Q3		2017 Week 30-2017	True	2017-07-29	2H	Q3-2017
30/07/2017	201707030	31	Week 31		July	1	Q3		2017 Week 31-2017	True	2017-07-30	2H	Q3-2017
31/07/2017	201707031	31	Week 31		July	1	Q3		2017 Week 31-2017	True	2017-07-31	2H	Q3-2017
01/08/2017	201708001	31	Week 31		August	1	Q3		2017 Week 31-2017	True	2017-08-01	2H	Q3-2017
02/08/2017	201708002	31	Week 31		August	1	Q3		2017 Week 31-2017	True	2017-08-02	2H	Q3-2017
03/08/2017	201708003	31	Week 31		August	1	Q3		2017 Week 31-2017	True	2017-08-03	2H	Q3-2017
04/08/2017	201708004	31	Week 31		August	1	Q3		2017 Week 31-2017	True	2017-08-04	2H	Q3-2017
05/08/2017	201708005	31	Week 31		August	1	Q3		2017 Week 31-2017	True	2017-08-05	2H	Q3-2017
06/08/2017	201708006	31	Week 31		August	1	Q3		2017 Week 31-2017	True	2017-08-06	2H	Q3-2017

2) Expected results shown here i.e. Quarter & Year values.

Maven Rooters - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: Quarter & Year
Data type: Text

Summarization: Don't summarize
Data category: Uncategorized

Sort by column
Data groups
Manage relationships
New column
Calculated columns

Structure
Formatting
Properties

1 Quarter & Year =
2 VAR Q1 = 'Calendar'[Month_ID] IN {1,2,3}
3 VAR Q2 = 'Calendar'[Month_ID] IN {4,5,6}
4 VAR Q3 = 'Calendar'[Month_ID] IN {7,8,9}
5 VAR Q4 = 'Calendar'[Month_ID] IN {10,11,12}
6
7 RETURN
8 SWITCH(
9 TRUE(),
10 Q1, "Q1" & "-" & 'Calendar'[Year_ID],
11 Q2, "Q2" & "-" & 'Calendar'[Year_ID],
12 Q3, "Q3" & "-" & 'Calendar'[Year_ID],
13 Q4, "Q4" & "-" & 'Calendar'[Year_ID],
14 "
15)

Fields

Search

Measure Table
Calendar
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
Store Lookup

Sort ascending
Sort descending
Clear sort
Clear filter
Clear all filters
Text filters
Search
(Select all)
Q1-2017
Q1-2018
Q1-2019
Q2-2017
Q2-2018
Q2-2019
Q3-2017
Q3-2018
Q3-2019
Q4-2017
Q4-2018
Q4-2019

OK Cancel

3) Column filter can be used for analysis.

Transaction_Date Date_ID Week_ID Week_Desc Month_ID Month_Name Quarter_ID Quarter_Name Year_ID Week_Year ISNUMBER Date Format (yyyy-mm-dd) Year_Matt Quarter & Year

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	ISNUMBER	Date Format (yyyy-mm-dd)	Year_Matt	Quarter & Year
01/07/2017	20170701	26	Week 26	7	July	1	Q3	2017	Week 26-2017	True	2017-07-01	2H	Q3-2017
02/07/2017	20170702	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-02	2H	Q3-2017
03/07/2017	20170703	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-03	2H	Q3-2017
04/07/2017	20170704	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-04	2H	Q3-2017
05/07/2017	20170705	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-05	2H	Q3-2017

Table: Calendar (250 rows) Column: Quarter & Year (10 distinct values)

5) Advanced CALCULATE

Introduction - Advanced CALCULATE



CALCULATE is a powerful DAX function which is commonly used to modify filter context; in this section, we'll explore advanced topics like **expanded tables, context transition, evaluation order and modifiers**

TOPICS WE'LL COVER:

Expanded Tables

Context Transition

Evaluation Order

Modifiers

COMMON USE CASES:

- Simplifying measures by adding new filter context that would normally use multiple nested functions
- Modifying filter context to override preexisting filters using modifiers like ALL, KEEPFILTERS or REMOVEFILTERS
- Creating calculated inputs for measures like % of total, % of category, etc.

Expanded Tables

Expanded Tables

Context Transition

Evaluation Order

Modifiers

An **expanded table** consists of the base table (which is visible to the user), along with columns from any related table connected via a 1-to-1 or many-to-1 relationship

Product Lookup Table			Store Lookup Table		
Product ID	Name	Category	Store ID	Address	Manager
1	Brazilian - Organic	Coffee beans	3	32-20 Broadway	6
2	Old Time Diner	Coffee beans	4	604 Union Street	11
3	Espresso Roast	Coffee beans	5	100 Church Street	16
4	Primo Roast	Coffee beans	6	122 E Broadway	21
5	Columbian Medium	Coffee beans	7	224 E 57 th Street	26

Trans ID	Product ID	Store ID	Quantity	Name	Category	Address	Manager
1	4	4	32	Primo Roast	Coffee beans	604 Union Street	11
2	2	3	17	Old Time Diner	Coffee beans	32-20 Broadway	6
3	3	7	113	Espresso Roast	Coffee beans	224 E 57 th Street	26
4	3	3	14	Espresso Roast	Coffee beans	32-20 Broadway	6
5	1	4	55	Brazilian - Organic	Coffee beans	604 Union Street	11

Transactions Table

Related fields from the **Product** and **Store** tables are accessible to DAX as part of the "expanded" **Transactions** table

As you begin to write more complex DAX & CALCULATE statements, understanding expanded tables is critical!

Expanded Tables

Context Transition

Evaluation Order

Modifiers



*Maven Roasters Data Model

Expanded tables in Models *always* go from the **MANY** side to the **ONE** side of the relationship

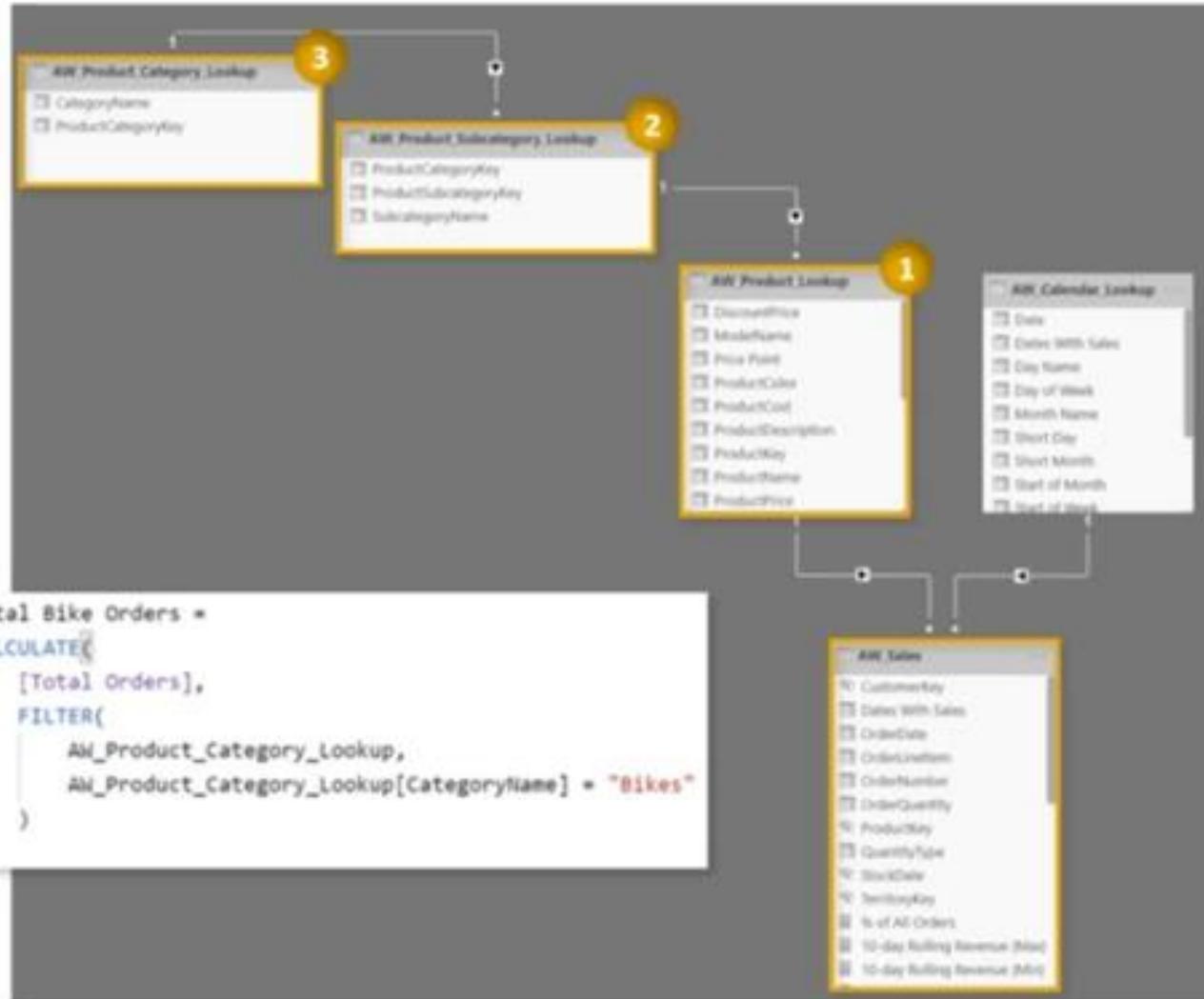
- Expanded tables contain all the columns of the original table **plus** all the columns on the one-side
- Understanding expanded tables is useful because it provides a clear understanding of how filter context propagates in DAX measures
- Once a filter is applied to a column, **all expanded tables containing that column are also filtered**

Expanded Tables

Context
Transition

Evaluation Order

Modifiers



*Adventure Works data model

In this case, the expanded version of the **AW_Sales** table contains all fields from the **product**, **subcategory**, and **category** lookup tables

Context Transition

Expanded Tables

Context Transition

Evaluation Order

Modifiers

Context Transition is the process of turning row context into filter context

- By default, calculated columns understand **row** context but not **filter** context
- To create filter context at the row-level, you can use **CALCULATE**

product_id	quantity_sold
70	9
70	4
70	2
70	2
70	5
70	6
70	8
70	7
70	3
70	11
70	10
70	12
70	15
70	13
70	14
70	16
70	17

Sum of Quantity
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153
153

If we add a column to calculate the sum of quantity_sold, we get repeating totals since there is **no filter context**, and we can't determine a unique value per row

To solve this, we can generate **filter context** at the **row-level**, using **CALCULATE** with **SUMX**

```
1 Sum of Quantity =  
2 CALCULATE(  
3   SUM(  
4     'Table'[quantity_sold])  
5   )
```

CALCULATE Sum of Quantity
9
4
1
2
5
6
8
7
3
11
10
12
15
13
14
16
17

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: Quantity Sold (SUM) Format: Whole number Data type: Whole number Summation: Sum Data category: Uncategorized Sort by column: Sort Group: Data groups Manage relationships: Relationships New column: Calculations

1 Quantity Sold (SUM) =
2 SUM(
3 'Food Inventory'[quantity_sold]
4)

1) Calculated column for Quantity Sold (SUM) is created here.

2) You will notice that “141433” value is repeating within the column which is incorrect.

store_id	baked_date	transaction_date	product_id	quantity_start_of_day	quantity_sold	Quantity Sold (SUM)
1	23 December 2016	06 January 2017	69	28	4	141433
1	23 December 2016	07 January 2017	69	28	4	141433
1	28 December 2016	08 January 2017	69	28	4	141433
1	31 December 2016	12 January 2017	69	28	4	141433
1	02 January 2017	13 January 2017	69	28	4	141433
1	07 January 2017	18 January 2017	69	28	4	141433
1	04 January 2017	16 January 2017	69	28	4	141433
1	18 January 2017	30 January 2017	69	28	4	141433
1	25 January 2017	07 February 2017	69	28	4	141433
1	01 February 2017	12 February 2017	69	28	4	141433
1	01 February 2017	18 February 2017	69	28	4	141433
1	11 February 2017	26 February 2017	69	28	4	141433
1	17 February 2017	21 February 2017	69	28	4	141433
1	20 March 2017	17 March 2017	69	28	4	141433
1	26 March 2017	29 March 2017	69	28	4	141433
1	03 April 2017	13 April 2017	69	28	4	141433
1	03 April 2017	14 April 2017	69	28	4	141433
1	05 April 2017	18 April 2017	69	28	4	141433
1	14 April 2017	23 April 2017	69	28	4	141433
1	19 April 2017	02 May 2017	69	28	4	141433
1	27 April 2017	07 May 2017	69	28	4	141433
1	30 April 2017	09 May 2017	69	28	4	141433
1	24 May 2017	23 May 2017	69	28	4	141433
1	24 May 2017	28 May 2017	69	28	4	141433
1	21 May 2017	02 June 2017	69	28	4	141433
1	24 May 2017	05 June 2017	69	28	4	141433
1	21 May 2017	10 June 2017	69	28	4	141433
1	07 June 2017	14 June 2017	69	28	4	141433
1	05 June 2017	19 June 2017	69	28	4	141433
1	07 June 2017	23 June 2017	69	28	4	141433

Table: Food Inventory (21,649 rows) Column: Quantity Sold (SUM) (1 distinct values)

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: Quantity Sold (CALCULATE) Format: Whole number Data type: Whole number Summarization: Sum Data category: Uncategorized Sort by column: Sort Data groups: Group Manage relationships: Relationship New column: Calculations

Structure Formatting Properties

1 Quantity Sold (CALCULATE) =
2 CALCULATE(
3 SUM(
4 'Food Inventory'[quantity_sold]
5)
6)

Fields

Search

Measure Table
Calendar
Customer Lookup
Employee Lookup
Food Inventory
baked_date
product_id
Quantity Sold (CALCULATE)
Quantity Sold (SUM)
quantity_sold
quantity_start_of_day
store_id
transaction_date
Product Lookup
Sales by Store
Store Lookup

store_id baked_date transaction_date product_id quantity_start_of_day quantity_sold Quantity Sold (SUM) Quantity Sold (CALCULATE)

1	23 December 2016	06 January 2017	89	18	4	141433	141433
2	25 December 2016	07 January 2017	89	18	4	141433	141433
3	28 December 2016	08 January 2017	89	18	4	141433	141433
4	31 December 2016	11 January 2017	89	18	4	141433	141433
5	02 January 2017	12 January 2017	89	18	4	141433	141433
6	07 January 2017	13 January 2017	89	18	4	141433	141433
7	04 January 2017	16 January 2017	89	18	4	141433	141433
8	18 January 2017	19 January 2017	89	18	4	141433	141433
9	29 January 2017	01 February 2017	89	18	4	141433	141433
10	01 February 2017	12 February 2017	89	18	4	141433	141433
11	07 February 2017	18 February 2017	89	18	4	141433	141433
12	11 February 2017	20 February 2017	89	18	4	141433	141433
13	17 February 2017	21 February 2017	89	18	4	141433	141433
14	20 March 2017	17 March 2017	89	18	4	141433	141433
15	25 March 2017	29 March 2017	89	18	4	141433	141433
16	03 April 2017	13 April 2017	89	18	4	141433	141433
17	01 April 2017	14 April 2017	89	18	4	141433	141433
18	06 April 2017	18 April 2017	89	18	4	141433	141433
19	14 April 2017	28 April 2017	89	18	4	141433	141433
20	19 April 2017	02 May 2017	89	18	4	141433	141433
21	27 April 2017	07 May 2017	89	18	4	141433	141433
22	30 April 2017	09 May 2017	89	18	4	141433	141433
23	14 May 2017	21 May 2017	89	18	4	141433	141433
24	24 May 2017	28 May 2017	89	18	4	141433	141433
25	21 May 2017	02 June 2017	89	18	4	141433	141433

Table: Food Inventory (27,649 rows) Column: Quantity Sold (CALCULATE) (23 distinct values)

1) Calculated column for Quantity Sold (CALCULATE) is created here.
2) The Quantity Sold (CALCULATE) column has been calculated correctly here.

Consider the following formula, defined as both a **measure** and a **calculated column**:

Expanded Tables

Context Transition

Evaluation Order

Modifiers

product_id

quantity_sold

Sum of Quantity

CALCULATE Sum of Quantity

69	3	959	959
69	1	959	959
69	2	959	959
69	5	959	959
69	6	959	959
69	8	959	959
69	7	959	959
72	20	959	959
72	18	959	959
72	17	959	959
72	21	959	959
72	19	959	959

product_id	Sum of Quantity Sold	CALCULATE Sum of Quantity Sold
69	154	154
70	153	153
71	326	326
72	326	326
Total	959	959

As a **calculated column**, we see the same total repeated on each row (no filter context)

As a **measure**, DAX automatically evaluates a CALCULATE & SUMX measure (shown below), to create the filter context needed to produce the correct values

```
1 Sum of Quantity Sold =
2 CALCULATE(
3   SUMX(
4     'Table',
5     'Table'[quantity_sold]))
6 )
```

Maven Roasters - Analysis - Power BI Desktop

Sign in  -   

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Out data Excel Power BI datasets Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Calculations Publish

Power BI Data Query Insert Calculations Share

Filters

Search

Filters on this visual

product_id is (All)

Add data fields here

Rows

product_id

Columns

Add data fields here

Values

Add data fields here

Drill through

Product Lookup

Cross-report

Sales by Store

Store Lookup

Keep all filters

On 

Add drill-through fields here

Visualizations

Fields

Search

Measure Table

Calendar

Customer Lookup

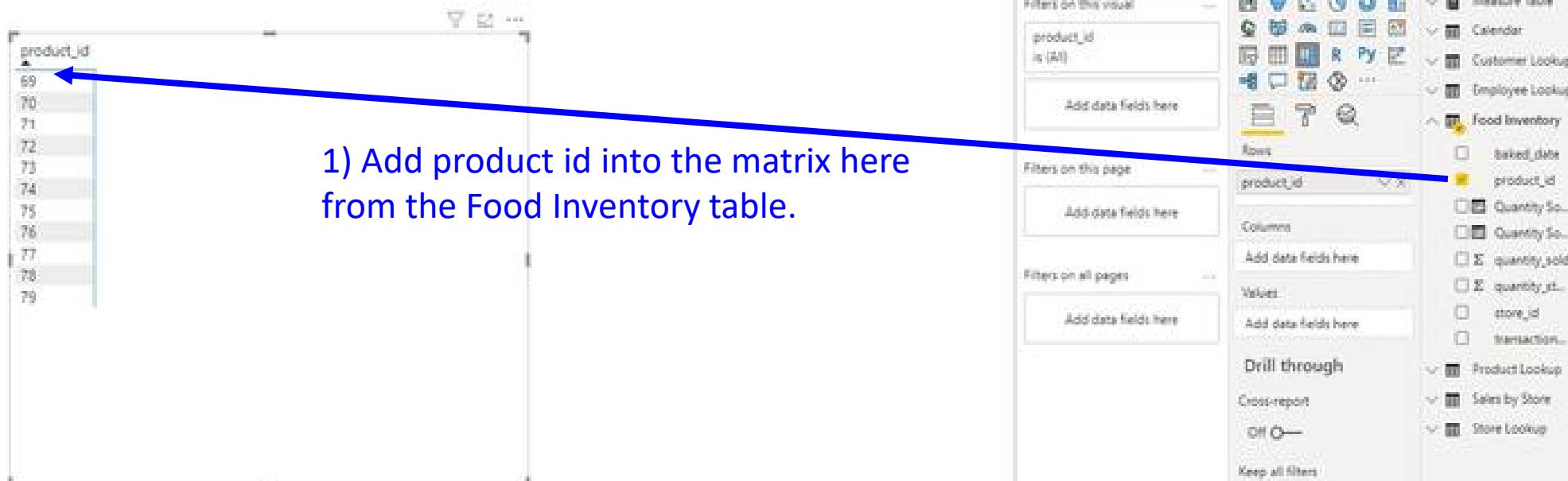
Employee Lookup

Food Inventory

- baked_date
- product_id
- Quantity Sold
- Quantity Sold
- quantity_sold
- quantity_st...
- store_id
- transaction...

Page 1 Advanced CALCULATE 

Page 2 of 2



Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: SUM Quantity Sold Format: Whole number Data category: Uncategorized

Home table: Food Inventory \$ - %

Structure, Formatting, Properties, Calculations

2) Click on the comma to format the number i.e. 1000 separator.

1 SUM Quantity Sold =
2 SUM(
3 'Food Inventory'[quantity_sold]
4)

Add data fields here

Add data fields here

Values

Drill through

Cross-report

On

Keep all filters

Add drill-through fields here

Product Lookup

Sales by Store

Store Lookup

Page 1 Advanced CALCULATE +

1) Create the SUM Quantity Sold first measure.

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: CALCULATE Sum Q... Format: Whole number Data category: Uncategorized New Quick Measure Measure

Home table: Food Inventory \$ - %

Structure Formatting Properties Calculations

2) Click on the comma to format the number i.e. 1000 separator.

1 CALCULATE Sum Quantity Sold =
2 CALCULATE(
3 SUM(
4 'Food Inventory'[quantity_sold]
5)
6)

1) Create the CALCULATE Sum Quantity Sold second measure.

Visualizations Fields

Measure Table Calendar

Date Format (yyyy-mm-dd) Date_ID DNUMBER Month_ID Month_Name Quarter & Year Quarter_ID Quarter_Name Transaction_Date Week_Desc Week_ID Week_Year Year_Half Year_ID

Customer Lookup Employee Lookup

Food Inventory

baked_date
CALCULATE Sum Quantity S...
product_id
Quantity Sold (CALCULATE)
Quantity Sold (SUM)
quantity_sold
quantity_start_of_day
store_id
SUM Quantity Sold
transaction_date

Product Lookup

Page 1 Advanced CALCULATE

Maven Reactors - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Out data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More New measure measure Quick Calculations Publish

Queries Data

Filters Visualizations Fields

Search

Filters on this page Add data fields here

Filters on all pages Add data fields here

Add drill-through fields here

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add both the below measures to the matrix.

1) CALCULATE Sum Quantity Sold

2) SUM Quantity Sold

You will notice that both the measures evaluate the exact same regardless of having the CALCULATE explicitly written or it being implied by the engine.

The screenshot shows a Power BI desktop interface. In the center, there is a matrix visualization with two columns: 'product_id' and 'CALCULATE Sum Quantity Sold / SUM Quantity Sold'. The matrix contains 16 rows of data, with the last row being the 'Total'. The 'Total' row shows values 141,433 for both columns. A blue bracket is placed around the second column of the matrix, and a blue arrow points from this bracket to the 'Fields' pane on the right. The 'Fields' pane lists various measures and columns, including 'product_id', 'Quantity Sold (CALCULATE)', 'Quantity Sold (SUM)', 'quantity_sold', 'quantity_start_of_day', 'store_id', 'SUM Quantity Sold', and 'transaction_date'. The 'Measures' section contains 'CALCULATE Sum Quantity Sold' and 'SUM Quantity Sold'. The 'Visualizations' section lists various charts and tables. The 'Fields' section lists columns like 'baked_date', 'customer_lookup', 'employee_lookup', 'Food Inventory', 'Product Lookup', 'Sales by Store', and 'Store Lookup'.

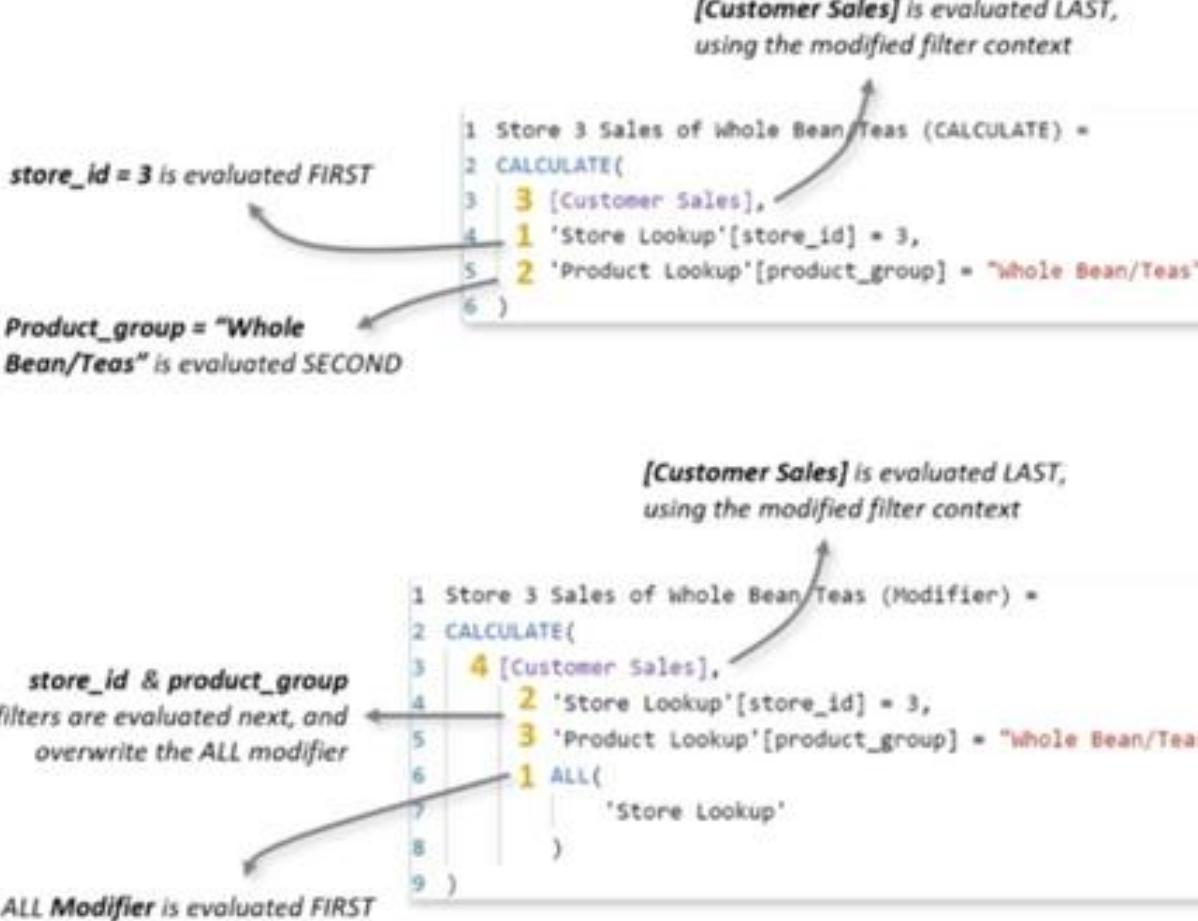
Evaluation Order

Expanded Tables

Context Transition

Evaluation Order

Modifiers



Transaction_Date	Store 3 Sales of Whole Bean/Teas (CALCULATE)	Store 3 Sales of Whole Bean/Teas (Modifier)
1/8/2017	\$89.80	\$89.80
1/9/2017	\$154.41	\$154.41
1/10/2017	\$61.65	\$61.65
1/11/2017	\$9.50	\$9.50
1/12/2017	\$84.20	\$84.20
1/14/2017	\$105.20	\$105.20
1/15/2017	\$257.01	\$257.01
Total	\$761.79	\$761.79

Both measures evaluate the same, since the ALL modifier evaluates first and is overwritten by the store_id and product_group filters



HEY THIS IS IMPORTANT!

DAX engines combine multiple filter arguments into a single filter context

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Store 3 Sales of whole Beans & Teas
Home table: Measure Table
Format: General
Data category: Uncategorized

New measure Quick measure measure

Structure Formatting Properties Calculations

1 Store 3 Sales of whole Beans & Teas (CALCULATE) =
2 CALCULATE(
3 [Customer Sales],
4 'Store Lookup'[store_id] = 3,
5 'Product Lookup'[product_group] = "Whole Bean/Teas"
6)

Values Add data fields here

Drill through Cross-report Off On Keep all filters Add drill-through fields here

Measure Table % Quantity ... Cost (CUSD) Customer L... Customer L... Customer L... Customer L... Orders by P... Scalar Measures Store 3 Sale... Tip & Best Practices Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup

Page 1 Advanced CALCULATE +

1) Create the first Store 3 Sales of whole Beans & Teas (CALCULATE) measure.

75		11,041	11,041
76		11,616	11,616
77		11,034	11,034
78		11,712	11,712
79		12,544	12,544
Total		141,433	141,433

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Queries Data Calculations Share

product_id CALCULATE Sum Quantity Sold SUM Quantity Sold

product_id	CALCULATE Sum Quantity Sold	SUM Quantity Sold
69	12,298	12,298
70	12,354	12,354
71	19,217	19,217
72	15,849	15,849
73	11,761	11,761
74	11,007	11,007
75	11,841	11,841
76	11,616	11,616
77	11,034	11,034
78	11,712	11,712
79	12,544	12,544
Total	141,413	141,413

Transaction Date Store 3 Sales of whole Beans & Teas (CALCULATE)

Transaction Date	Store 3 Sales of whole Beans & Teas (CALCULATE)
08/01/2017	\$19.00
09/01/2017	\$194.41
10/01/2017	\$41.63
11/01/2017	\$9.50
12/01/2017	\$24.20
13/01/2017	\$105.20
14/01/2017	\$237.00
15/01/2017	\$73.00
16/01/2017	\$103.43
17/01/2017	\$105.43
18/01/2017	\$146.18
19/01/2017	\$125.30
20/01/2017	\$258.00
21/01/2017	\$34.20
Total	\$90,829.49

Filters

Visualizations

Fields

Add data fields here

Add data fields here

Add data fields here

Values

Drill through

Cross-report

Keep all filters

Add drill-through fields here

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (SUMMARIZE)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Tip & Best Practices

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

Page 1 Advanced CALCULATE

Page 2 of 2

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Store 3 Sales of whole Beans & Teas (ALL Modifier) \$ - % Data category: Uncategorized

Home table: Measure Table \$ - % New measure Quick measure

Structure Formatting Properties Calculations

1 Store 3 Sales of whole Beans & Teas (ALL Modifier) =
2 CALCULATE(
3 [Customer Sales],
4 'Store Lookup'[store_id] = 3,
5 'Product Lookup'[product_group] = "Whole Bean/Teas",
6 ALL(
7 'Store Lookup'
8)
9)

Visualizations Fields

Search: Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales IF (COALESCE)
- Customer Sales IF (ISBLANK)
- Orders by Females (Wrong)
- Scalar Measures
 - Store 3 Sales of whole Beans & Teas (ALL Modifier)
 - Store 3 Sales of whole Beans & Teas (CALCULATE)
- Tip & Best Practices
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store
- Store Lookup

1) Create the second Store 3 Sales of whole Beans & Teas (All Modifier) measure.

Page 1 Advanced CALCULATE

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get Data Excel Power BI datasets SQL Server Enter data Refresh data New visual Text box More visuals Quick measure measure Publish

Openers Data Queries Calculations Share

product_id CALCULATE Sum Quantity Sold: SUM Quantity Sold

product_id	Transaction Date	Store 3 Sales of whole Beans & Teas (CALCULATE)	Store 3 Sales of whole Beans & Teas (ALL Modifier)
69	08/01/2017	\$89.80	\$89.80
70	09/01/2017	\$154.41	\$154.41
71	10/01/2017	\$61.65	\$61.65
72	11/01/2017	\$9.30	\$9.30
73	12/01/2017	\$84.20	\$84.20
74	13/01/2017	\$105.20	\$105.20
75	14/01/2017	\$257.03	\$257.03
76	15/01/2017	\$73.88	\$73.88
77	16/01/2017	\$105.43	\$105.43
78	17/01/2017	\$166.18	\$166.18
79	18/01/2017	\$125.38	\$125.38
Total	19/01/2017	\$258.88	\$258.88
	20/01/2017	\$84.20	\$84.20
	Total	\$90,829.40	\$90,829.40

Filters Visualizations Fields

Add data fields here

Add data fields here

Measure Table

- % Quantity Sold to Females
- Cost (CURRENTCITY)
- Customer Sales (Last Year)
- Customer Sales IF (COALESCE)
- Customer Sales IF (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales of whole Beans & Teas (ALL Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)

Tip & Best Practices

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales By Store

Store Lookup

Perform the below actions.

- 1) Add the Store 3 Sales of whole Beans & Teas (ALL Modifier) measure to the matrix.
- 2) Format the measure to currency & 2 decimal places.

Each of the rows have exactly the same values from the two measures created.

Page 1 Advanced CALCULATE

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

Insert Calculations Share

product_id CALCULATE Sum Quantity Sold SUM Quantity Sold

product_id	CALCULATE Sum Quantity Sold	SUM Quantity Sold
69	12,298	12,298
70	12,554	12,554
71	19,217	19,217
72	15,849	15,849
73	11,781	11,781
74	11,007	11,007
75	11,841	11,841
76	11,616	11,616
77	11,034	11,034
78	11,712	11,712
79	12,544	12,544
Total	141,433	141,433

store_id Store 3 Sales of whole Beans & Teas (CALCULATE) Store 3 Sales of whole Beans & Teas (ALL Modifier)

store_id	Store 3 Sales of whole Beans & Teas (CALCULATE)	Store 3 Sales of whole Beans & Teas (ALL Modifier)
2	\$90,829.40	\$90,829.40
3	\$90,829.40	\$90,829.40
5	\$90,829.40	\$90,829.40
8	\$90,829.40	\$90,829.40
Total	\$90,829.40	\$90,829.40

Filters

Visualizations

Fields

Search

Filters on this visual

- Store 3 Sales of whole Beans & Teas (CALCULATE) is (All)
- Store 3 Sales of whole Beans & Teas (ALL Modifier) is (All)
- store_id is (All)
- transaction_date is (All)

Add data fields here

Rows

store_id

Transaction_date

Columns

Values

Drill through

Cross-report

Keep all filters

On

Add drill-through fields here

Page 1 Advanced CALCULATE

Page 2 of 2

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Data Queries Insert Calculations Share

product_id CALCULATE Sum Quantity Sold SUM(Quantity Sold)

product_id	12,298	12,298
70	12,554	12,554
71	19,217	19,217
72	15,849	15,849
73	11,761	11,761
74	11,007	11,007
75	11,641	11,641
76	11,616	11,616
77	11,634	11,634
78	11,712	11,712
79	12,544	12,544
Total	141,433	141,433

store_id Store 3 Sales of whole Beans & Teas (CALCULATE) Store 3 Sales of whole Beans & Teas (ALL, Modified)

store_id	1	2	3
1	\$90,829.40	\$90,829.40	\$90,829.40
2	\$90,829.40	\$90,829.40	\$90,829.40
3	\$90,829.40	\$90,829.40	\$90,829.40
Total	\$90,829.40	\$90,829.40	\$90,829.40

Filters

Visualizations

Fields

Search

Rows on this visual

Store 3 Sales of whole Beans & Teas (All)

Store 3 Sales of whole Beans & Teas (All)

store_id is not 2

Filter type Basic filtering

Select all

Require single selection

Transaction Date is (All)

Add data fields here

Values

Store 3 Sales of whole Beans & Teas (All)

Store 3 Sales of whole Beans & Teas (All)

Drill through

Cross-report

Off

Keep all filters

On

Add anti-through fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Page 1 Advanced CALCULATE

Perform the below actions.

- 1) Change filter type to Basic Filtering from Advanced Filtering.
- 2) Uncheck store id 2.

Applying the above steps each of the rows still have exactly the same repeating values from the two measures created.

The screenshot shows the Microsoft Power BI interface. At the top is the ribbon with Home, Insert, Modeling, View, Help, Format, and Data / Drill options. Below the ribbon are several icons for file operations like Cut, Copy, Paste, and Format painter, along with links for Get data, Excel, Power BI datasets, SQL Server, Enter data, Recent sources, Transform Refresh data, New visual, Text box, More visuals, New measure, Quick measure, and Publish.

The main area displays two tables. The first table on the left has columns for product_id, CALCULATE Sum Quantity Sold, and SUM Quantity Sold. The second table on the right has columns for store_id, Store 3 Sales of whole Beans & Teas (CALCULATE), and Store 3 Sales of whole Beans & Teas (All Modified). A blue arrow points from the text below to the second table.

Filters pane on the right shows various filters applied, including a Row filter for store_id (not 2) and a Value filter for Transaction Date (is All). The **Visualizations** pane shows a list of available visual types, and the **Fields** pane shows the data model with nodes like Food Inventory, More Id, Sales by Store, More Id, and Store Lookup, with store_id selected.

When drilling in by day you receive the same exact values for the both measures this is what we would of expected to happen. The CALCULATE filters its calculate arguments which is really important to understand especially when it comes to modifiers.

CALCULATE Modifiers

Expanded Tables

Context Transition

Evaluation Order

Modifiers

Modifiers are used to alter the way CALCULATE creates filter context, and are added as *filter* arguments within a CALCULATE function

- Modifiers are typically used to change filter context, access inactive table relationships, or change the way filters propagate (*i.e. one-way to bidirectional*)

Modify Filters

Use Relationships

Change Filter Propagation

Common Examples:

- ALL
- ALLSELECTED
- ALLNOBLANKROW
- ALLEXCEPT
- KEEPFILTERS
- REMOVEFILTERS

Common Examples:

- USERELATIONSHIP

Common Examples:

- CROSSFILTER

=CALCULATE(Expression, [Filter1], [Filter2],...)

Expanded Tables

Context Transition

Evaluation Order

Modifiers

CALCULATE filter expressions accept both boolean & table functions (*individually or at the same time!*), but all filter arguments are automatically converted into a table

```
1 Store 3 Sales =  
2 CALCULATE(  
3     [Customer Sales],  
4     'Store Lookup'[store_id] = 3  
5 )
```

DAX interprets this as a table!

- Any time you use write a function that contains a logical statement (IN, >, <, =, etc.) you're **creating a table** (*internally processed with FILTER & ALL*)

```
1 All Store Sales =  
2 CALCULATE(  
3     [Customer Sales],  
4     ALL(  
5         'Store Lookup'  
6     )  
7 )
```

And this too, obviously



REMOVEFILTERS

REMOVEFILTERS()

Clears filters from the specified tables or columns

Expanded Tables

Context Transition

Evaluation Order

Modifiers

=**REMOVEFILTERS** (**TableNameorColumnName**, [**ColumnName**], [...])

Table or column that you want to clear the filters from

(Optional) Repeatable column names that allow you to specifically remove filters from individual columns within the base table

Examples:

- **CALCULATE**(
[Customer Sales],
REMOVEFILTERS(
'Store Lookup'))

Examples:

- **CALCULATE**(
[Customer Sales],
REMOVEFILTERS(
'Store Lookup'[store_id]))



PRO TIP:

REMOVEFILTERS is an alias for ALL, but can only be used as a CALCULATE modifier (not as a table function)

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals

New measure measure Quick Calculations Publish

Dashboard Data Queries Insert Calculations Share

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LF (COALESCE)
- Customer Sales LF (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales of whole Beans & Tea (ALL.Modifier)
- Store 3 Sales of whole Beans & Tea (CALCULATE)

Tip & Best Practices

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales By Store

Store Lookup

Rows

product

Columns

Add data fields here

Values

Profit

Drill through

Cross-report

On

Keep all items

On

Add drill-through fields here

product Profit

product	Profit
	46,949.63
Almond Croissant	25,990.11
Brazilian - Organic	18,561.60
Brazilian Lg	68,252.63
Brazilian Pg	60,417.00
Brazilian Sm	44,007.15
Cappuccino	74,123.44
Cappuccino Lg	82,332.25
Carmel syrup	9,274.80
Chili Mayan	8,299.01
Chocolate Chip Biscotti	23,809.43
Chocolate Croissant	42,562.25
Chocolate syrup	9,630.60
Civet Cat	53,244.00
Columbian Medium Roast	10,896.00
Columbian Medium Roast Lg	58,432.50
Total	3,163,528.61

Perform the below actions.

- 1) Add a matrix to the visual.
- 2) Drag in the product field into the matrix.
- 3) Drag in the profit measure into the matrix.

Page 1 Advanced CALCULATE Page 2 +

Maven Roastem - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name Total Profit

Home table Measure Table

Format: % Format: Currency

Data category: Uncategorized

New Quick measure measure

Structure

Formatting Properties Calculators

1 Total Profit =
2 CALCULATE(
3 [Profit],
4 REMOVEFILTERS(
5 'Sales by Store'
6)
7)

Caramel Syrup 9,274.80
Chili Mayan 8,299.01
Chocolate Chip Biscotti 21,809.41
Chocolate Croissant 42,562.25
Chocolate Syrup 9,630.60
Civet Cat 53,244.00
Columbian Medium Roast 10,896.00
Columbian Medium Roast 58,412.50
Total 3,163,528.61

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales BY (COALESCE)
- Customer Sales BY (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales of whole Beans & Teas (ALL Modified)
- Store 3 Sales of whole Beans & Teas (CALCULATE)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- DayNumber
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction Date
- Week Desc
- Week_ID
- Week_Year
- Year Half
- Year_ID

Customer Lookups

Employee Lookups

Food Inventory

2) Change the formatting accordingly over here.

1) Create the Total Profit measure.

Page 1 Advanced CALCULATE Page 2 +

Maven Rooters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data from Excel Power BI datasets SQL Server Enter data from recent sources Transform Refresh data New visual Text box More Quick measure measure

New measure Publish

Datasets Data Queries Items Calculations Share

The same values appear on each row as the filters for Sales by Store have been removed in the measure created earlier. Input is beneficial when computing % for profit. Used as an input for another measure.

product Profit Total Profit

product	Profit	Total Profit
Almond Croissant	25.990,61	\$3,163,528,61
Brazilian - Organic	18.551,60	\$3,163,528,61
Brazilian Lg	68.212,63	\$3,163,528,61
Brazilian Pg	60.117,00	\$3,163,528,61
Brazilian Sm	34.007,15	\$3,163,528,61
Cappuccino	74.123,44	\$3,163,528,61
Cappuccino Lg	82.332,25	\$3,163,528,61
Caramel syrup	9.274,00	\$3,163,528,61
Chili Mayan	9.299,01	\$3,163,528,61
Chocolate Chip Biscotti	23.400,43	\$3,163,528,61
Chocolate Croissant	42.562,23	\$3,163,528,61
Chocolate syrup	9.430,60	\$3,163,528,61
Civet Cat	53.244,00	\$3,163,528,61
Columbian Medium Roast	10.896,00	\$3,163,528,61
Columbian Medium Roast Lg	58.432,50	\$3,163,528,61
Total	3,163,528,61	\$3,163,528,61

3) Add the Total Profit measure to the matrix.

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LF (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Scalar Measures
- Store 3 Sales of whole Beans & Teas (ALL, Modifier)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Tip & Best Practices
- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- DATEPART
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Day
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Customer Lookup

Employee Lookup

Food Inventory

Page 1 Advanced CALCULATE Page 2 +

KEEPFILTERS

Expanded Tables

Context Transition

Evaluation Order

Modifiers

KEEPFILTERS()

Does not remove an existing column or table filter for an individual CALCULATE expression

=KEEPFILTERS(Expression)

CALCULATE or CALCULATETABLE
function expression or filter

Examples:

- `CALCULATE([Measure],
KEEPFILTERS('Table'[ID] = "Value"))`

- ✓ Allows you to control **which specific filters** are applied to a calculation
- ✓ KEEPFILTERS does not actually *modify* existing filters, but adds new filter context (*think inner join*)

1) Create the first Store 5 Profit (KEEPFILTERS) measure.

2) Change the formatting accordingly over here.

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Store 5 Profit (KEEPFILTERS)
Home table: Measure Table
Format: Currency
Data category: Uncategorized

Structure Properties Calculations

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales IF (COALESCE)
- Customer Sales IF (ISBLANK)
- Orders by Females (Wrong)
- Scalar Measures
- Store 3 Sales of whole Beans & Teas (ALL, Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- GNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Customer Lookups

Employee Lookups

1 Store 5 Profit (KEEPFILTERS) =
2 CALCULATE(
 [Profit],
 KEEPFILTERS(
 'Store Lookup'[store_id] = 5
)
)

Caramel Syrup	9,274.00	\$3,163,528.61	
Chili Mocha	8,299.01	\$3,163,528.61	
Chocolate Chip Biscotti	23,809.43	\$3,163,528.61	
Chocolate Croissant	42,562.25	\$3,163,528.61	
Chocolate Syrup	9,630.60	\$3,163,528.61	
Civet Cat	53,244.00	\$3,163,528.61	
Columbian Medium Roast	10,896.00	\$3,163,528.61	
Columbian Medium Roast Lg	58,432.50	\$3,163,528.61	
Total	3,163,528.61	\$3,163,528.61	

Page 1 Advanced CALCULATE Page 2

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Out Data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh Data New visual Text box More visuals Quick measure measure Publish

Disorders Data Queries Calculations Share

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales IF (COALESCE)
- Customer Sales IF (ISBLANK)
- Orders by Females (Wrong)

Rows

product

Columns

Add data fields here

Values

Profit

Total Profit

Store 5 Profit (KEEPFILTERS)

Scalar Measures

- Store 5 Sales of whole Beans & Teas (ALL Modified)
- Store 5 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit (KEEPMODEL)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- NUMBER

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Product

Page 1 Advanced CALCULATE Page 2

Page 3 of 3

product	Profit	Total Profit	Store 5 Profit (KEEPFILTERS)
Almond Croissant	\$9,949.63	\$3,163,528.61	\$19,006.73
Brazilian - Organic	25,990.11	\$3,163,528.61	\$8,933.33
Brazilian Lg	18,561.60	\$3,163,528.61	\$7,372.80
Brazilian Rg	68,252.63	\$3,163,528.61	\$21,683.00
Brazilian Sm	60,417.00	\$3,163,528.61	\$19,703.23
Cappuccino	44,007.15	\$3,163,528.61	\$14,397.99
Cappuccino Lg	74,173.48	\$3,163,528.61	\$23,847.19
Carmel syrup	82,332.23	\$3,163,528.61	\$29,340.94
Chili Mayan	9,274.80	\$3,163,528.61	\$4,029.60
Chocolate Chip Biscotti	9,299.01	\$3,163,528.61	\$3,604.43
Chocolate Croissant	23,809.43	\$3,163,528.61	\$8,399.98
Chocolate syrup	42,562.25	\$3,163,528.61	\$14,395.80
Civet Cat	9,630.60	\$3,163,528.61	\$4,027.20
Columbian Medium Roast	53,244.00	\$3,163,528.61	\$11,052.00
Total		\$3,163,528.61	\$1,048,465.23

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data from Excel Power BI datasets SQL Server Enter data Refresh data Transform data New visual Text box More visuals New measure Quick measure Publish

Dataset Data Query Insert Calculations Share

store_id

product Profit Total Profit Store 5 Profit (DOLLARS)

Almond Croissant	49,949.61	\$3,163,528.61	\$19,006.73
Brazilian - Organic	25,990.11	\$3,163,528.61	\$8,933.33
Brazilian Lg	18,561.69	\$3,163,528.61	\$7,372.80
Brazilian Rg	68,252.61	\$3,163,528.61	\$21,693.00
Brazilian Sm	60,417.00	\$3,163,528.61	\$19,703.25
Cappuccino	74,123.44	\$3,163,528.61	\$23,847.19
Cappuccino Lg	82,352.25	\$3,163,528.61	\$29,340.94
Carmel syrup	9,274.80	\$3,163,528.61	\$4,029.60
Chili Mayan	8,299.01	\$3,163,528.61	\$3,604.43
Chocolate Chip Biscotti	23,809.43	\$3,163,528.61	\$9,398.98
Chocolate Croissant	42,562.25	\$3,163,528.61	\$14,395.80
Chocolate syrup	9,630.60	\$3,163,528.61	\$4,027.20
Civet Cat	53,244.00	\$3,163,528.61	\$11,052.00
Columbian Medium Roast	10,895.00	\$3,163,528.61	\$2,808.00
Total	3,163,528.61	\$3,163,528.61	\$1,048,468.23

Visualizations Fields

Filters

Food Inventory

store_id

Sales by Store

store_id

Store Lookup

store_id

Drill through

Cross-report

Off

Keep all items

On

Add drill-through fields here

Perform the below actions.

- 1) Add a slicer to the visual.
- 2) Add the store id from the Store Lookup table to the slicer.

Page 1 Advanced CALCULATE Page 2

Page 3 of 3

Maven Router - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals

New measure measure Publish

Dashboard Data Queries Calculations Share

store_id

1 8

List

Dropdown

Between

Less than or equal to

Greater than or equal to

product Profit Total Profit Store # Profit (K) [Profit(\$)]

product	Profit	Total Profit	Store # Profit (K) [Profit(\$)]
Almond Croissant	49,949.63	\$3,163,528.61	\$19,006.73
Brazilian - Organic	23,990.11	\$3,163,528.61	\$8,993.33
Brazilian Lg	18,561.60	\$3,163,528.61	\$7,372.80
Brazilian M	68,252.63	\$3,163,528.61	\$21,693.00
Brazilian S	60,417.00	\$3,163,528.61	\$19,703.25
Brazilian Sm	44,007.15	\$3,163,528.61	\$14,387.90
Cappuccino	74,123.44	\$3,163,528.61	\$23,547.19
Cappuccino Lg	82,352.25	\$3,163,528.61	\$29,349.94
Caramel syrup	9,274.80	\$3,163,528.61	\$4,029.60
Chili Mayan	9,299.01	\$3,163,528.61	\$3,604.43
Chocolate Chip Biscotti	21,809.43	\$3,163,528.61	\$9,399.68
Chocolate Croissant	42,562.25	\$3,163,528.61	\$14,395.00
Chocolate syrup	9,630.60	\$3,163,528.61	\$4,017.20
Cheet Cat	53,244.00	\$3,163,528.61	\$11,052.00
Columbian Medium Roast	10,896.00	\$3,163,528.61	\$2,808.00
Total	3,163,528.61	\$3,163,528.61	\$1,048,465.23

Visualizations Fields

store_id

Food Inventory

Sales by Store

store_id

Store Lookups

store_id

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Page 1 Advanced CALCULATE Page 2 +

Page 3 of 3

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

New Calculations Share

store_id

product Profit Total Profit Store 5 Profit (X)EPILT(RS)

Almond Croissant	19,606.73	\$3,163,528.61	\$19,606.73
Brazilian - Organic	6,911.33	\$3,163,528.61	\$6,911.33
Brazilian Lg	7,372.00	\$3,163,528.61	\$7,372.00
Brazilian Rg	21,693.00	\$3,163,528.61	\$21,693.00
Brazilian Sm	19,703.25	\$3,163,528.61	\$19,703.25
Cappuccino	14,397.00	\$3,163,528.61	\$14,397.00
Cappuccino Lg	23,847.19	\$3,163,528.61	\$23,847.19
Caramel syrup	29,340.94	\$3,163,528.61	\$29,340.94
Chili Mayan	4,039.60	\$3,163,528.61	\$4,039.60
Chocolate Chip R	9,604.43	\$3,163,528.61	\$9,604.43
Chocolate Croissant	9,399.98	\$3,163,528.61	\$9,399.98
Chocolate syrup	14,395.80	\$3,163,528.61	\$14,395.80
Civet Cat	4,027.20	\$3,163,528.61	\$4,027.20
Columbian Medium Roast	11,692.00	\$3,163,528.61	\$11,692.00
Total	1,048,465.23	\$3,163,528.61	\$1,048,465.23

Slicer list is now present here.
Values will only displayed when the store id 5 is activated.

Visualizations Fields

Filters

Food Inventory

Store Lookup

Drill through

Cross-report

Off —

Keep all items

On —

Add drill-through fields here

Page 1 Advanced CALCULATE Page 2

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Calculated Share

store_id

product Profit Total Profit Store 5 Profit (KEEPITRS)

Almond Croissant	11,600.50	\$3,163,528.61
Brazilian - Organic	6,386.95	\$3,163,528.61
Brazilian Lg	4,910.40	\$3,163,528.61
Brazilian Pg	22,624.88	\$3,163,528.61
Brazilian Sm	20,706.75	\$3,163,528.61
Cappuccino	14,201.55	\$3,163,528.61
Cappuccino Lg	25,121.25	\$3,163,528.61
Caramel syrup	27,817.31	\$3,163,528.61
Chili Mayan	1,856.60	\$3,163,528.61
Chocolate Chip Brownie	3,679.28	\$3,163,528.61
Chocolate Croissant	7,937.60	\$3,163,528.61
Chocolate syrup	10,830.18	\$3,163,528.61
Coffee syrup	2,439.00	\$3,163,528.61
Cresc Cat	12,636.00	\$3,163,528.61
Columbian Medium Roast	3,780.00	\$3,163,528.61
Total	1,057,182.15	\$3,163,528.61

No values will be displayed when the store id 2, 3 & 8 is activated.

Visualizations Fields

Filters

Food Inventory

Sales by Store

Store Lookup

store_id

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Page 1 Advanced CALCULATE Page 2

File Home Insert Modeling View Help Table tools Measure tools

Name: Store 5 Profit Format: Currency Data category: Uncategorized

Home table: Measure Table \$ - % # . : New Quick measure measure

Structure Properties Calculations

1 Store 5 Profit =
2 CALCULATE(
3 [Profit],
4 'Store Lookup'[store_id] = 5
5)

Earl Grey Lg 20,027.23 \$1,163,528.61 \$20,027.23
Earl Grey Pg 16,689.38 \$1,163,528.61 \$16,689.38
English Breakfast 3,147.95 \$1,163,528.61 \$3,147.95
English Breakfast Lg 16,666.00 \$1,163,528.61 \$16,666.00
English Breakfast Pg 17,304.38 \$1,163,528.61 \$17,304.38
Espresso Roast 3,363.00 \$1,163,528.61 \$3,363.00
Grand Total 1,048,465.23 \$3,163,528.61 \$1,048,465.23

Visualizations Fields

Add data fields here

Scalar Measures

- Store 3 Sales of whole Beans & Tea (ALL.Modifier)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Date Format (yyyy-mm-dd)

Date_ID

isNUMBER

Month_ID

Month_Name

Quarter & Year

Quarter_ID

Quarter_Name

Transaction Date

Week Desc

Week_ID

Week_Year

Year Half

Year_ID

Customer Lookup

Customer ID

Page 1 Advanced CALCULATE Page 2 +

2) Change the formatting accordingly over here.

1) Create the second Store 5 Profit measure.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

Queries Data Calculations Share

store_id

- 2
- 3
- 4**
- 5

product Profit Total Profit Store 5 Profit (KEEPFILTERS) Store 5 Profit

product	Profit	Total Profit	Store 5 Profit (KEEPFILTERS)	Store 5 Profit
Almond Croissant	8,933.33	\$3,163,528.61	\$8,933.33	\$8,933.33
Brazilian - Organic	7,372.80	\$3,163,528.61	\$7,372.80	\$7,372.80
Brazilian Lg	21,693.00	\$3,163,528.61	\$21,693.00	\$21,693.00
Brazilian Pg	19,700.23	\$3,163,528.61	\$19,700.23	\$19,700.23
Brazilian Sm	14,397.90	\$3,163,528.61	\$14,397.90	\$14,397.90
Cappuccino	23,047.19	\$3,163,528.61	\$23,047.19	\$23,047.19
Cappuccino Lg	29,340.94	\$3,163,528.61	\$29,340.94	\$29,340.94
Carmel syrup	4,029.60	\$3,163,528.61	\$4,029.60	\$4,029.60
Chili Mayan	3,604.43	\$3,163,528.61	\$3,604.43	\$3,604.43
Chocolate Chip Biscotti	9,399.00	\$3,163,528.61	\$9,399.00	\$9,399.00
Chocolate Croissant	14,395.87	\$3,163,528.61	\$14,395.87	\$14,395.87
Chocolate syrup	4,027.20	\$3,163,528.61	\$4,027.20	\$4,027.20
Civet Cat	11,052.00	\$3,163,528.61	\$11,052.00	\$11,052.00
Columbian Medium Roast	2,000.00	\$3,163,528.61	\$2,000.00	\$2,000.00
Total	1,048,465.23	\$3,163,528.61	\$1,048,465.23	\$1,048,465.23

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales IF (COALESCE)
- Customer Sales IF (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales of whole Beans & Teas (ALL Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPMODEL)
- Store 5 Profit (KEEPMODEL,KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- ENumber
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Customer Lookup

Environment Lookups

3) Add the Store 5 Profit measure to the matrix. You will notice that the values for the below measures are evaluating to the same numbers as you would expect.

- Store 5 Profit (KEEPFILTERS)
- Store 5 Profit

Page 1 Advanced CALCULATE Page 2 +

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More

New measure Quick measure Publish

Power Data Query Calculations Share

store_id

product Profit Total Profit Store 5 Profit (KEEPFILTERS) Store 5 Profit

	Profit	Total Profit	Store 5 Profit (KEEPFILTERS)	Store 5 Profit
Almond Croissant	13,500.50	\$3,163,528.61	\$19,006.73	\$19,006.73
Brazilian - Organic	8,386.95	\$3,163,528.61	\$8,933.93	\$8,933.93
Brazilian Lg	4,910.40	\$3,163,528.61	\$7,372.80	\$7,372.80
Brazilian Rg	22,624.88	\$3,163,528.61	\$21,693.00	\$21,693.00
Brazilian Sm	20,706.75	\$3,163,528.61	\$19,703.25	\$19,703.25
Cappuccino	14,201.55	\$3,163,528.61	\$14,397.90	\$14,397.90
Cappuccino Lg	23,121.29	\$3,163,528.61	\$23,647.19	\$23,647.19
Carmel syrup	27,617.31	\$3,163,528.61	\$29,340.94	\$29,340.94
Chili Mayan	1,956.60	\$3,163,528.61	\$4,629.60	\$4,629.60
Chocolate Chip Biscotti	3,879.28	\$3,163,528.61	\$3,804.43	\$3,804.43
Chocolate Croissant	7,937.60	\$3,163,528.61	\$9,399.98	\$9,399.98
Chocolate syrup	13,830.18	\$3,163,528.61	\$14,395.80	\$14,395.80
Civet Cat	2,430.15	\$3,163,528.61	\$4,627.20	\$4,627.20
Columbian Medium Roast	3,700.00	\$3,163,528.61	\$11,052.00	\$11,052.00
Total	1,057,182.15	\$3,163,528.61	\$1,048,465.23	\$1,048,465.23

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (DAILYSCENE)
- Customer Sales LY (BALANCE)
- Orders by Females (Wrong)
- Scalar Measures
- Store 3 Sales of whole Beans & Teas (ALL Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Top & Best Practices
- Total Profit

Drill through

Cross-report

Off —

Keep all items

On —

Add drill-through fields here

- Date Format (yyyy-mm-dd)
- Date_ID
- NUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Customer Lookup

Environment Overview

Page 1 Advanced CALCULATE Page 2 +

Page 3 of 3

ASSIGNMENT Advanced CALCULATE

KEY OBJECTIVES

- 1) Create a matrix containing **product group** & **store id** on rows.
- 2) Use **KEEPFILTERS** to create measures that show sales for each of the 3 stores.
- 3) Use **REMOVEFILTERS** with **variables** to create a single measure for % of Store Sales.

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Calculations Publish

store_id

- 2
- 3
- 5
- 8

product_group

- Add-ons
- Beverages
- Food
- Merchandise
- Whole Bean/Teas

product Profit Total Profit Store 5 Profit (KEEPFILTERS) Store 5 Profit

product	Profit	Total Profit	Store 5 Profit (KEEPFILTERS)	Store 5 Profit
Almond Croissant	11,500.50	\$3,163,528.61	\$19,006.73	
Brazilian - Organic	6,386.95	\$3,163,528.61	\$8,913.33	
Brazilian Lg	4,910.40	\$3,163,528.61	\$7,372.80	
Brazilian Lg	22,624.68	\$3,163,528.61	\$21,691.00	
Brazilian Sm	20,756.73	\$3,163,528.61	\$19,703.23	
Cappuccino	14,201.55	\$3,163,528.61	\$14,387.60	
Cappuccino Lg	25,121.25	\$3,163,528.61	\$23,847.19	
Caramel syrup	27,817.31	\$3,163,528.61	\$29,340.64	
Chili Mayan	1,956.60	\$3,163,528.61	\$4,029.60	
Chocolate Chip Biscotti	2,879.28	\$3,163,528.61	\$3,604.43	
Chocolate Croissant	7,937.60	\$3,163,528.61	\$9,399.98	
Chocolate syrup	13,830.18	\$3,163,528.61	\$14,395.80	
Coffee Cat	2,430.00	\$3,163,528.61	\$4,027.20	
Columbian Medium Roast	12,636.00	\$3,163,528.61	\$11,052.00	
Total	1,780.00	\$3,163,528.61	\$2,808.00	
	1,057,182.15	\$3,163,528.61	\$1,048,465.23	

Visualizations

Rows

- product_group
- store_id

Columns

Values

Drill through

Cross-report

Off —

Keep all items

On —

Add drill-through fields here

Fields

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (DENSE)
- Customer Sales LY (BLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales of whole Beans & Teas (ALL Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date Format (yy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Customer Lookup

Previous Analysis

Page 1 Advanced CALCULATE Keep Remove Filters +

Key objective 1 is now complete. Create a matrix containing product group & store id on rows.

1) Objective 1 to create a matrix containing product group & store id on rows. Perform the below actions.

- 1) Add a matrix here
- 2) Add the product group field to the rows
- 3) Add the store id field to the rows

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Store 3 Sales (KEEPFILTERS) Format: Currency Data category: Uncategorized

Home table: Measure Table \$ - % , . : New Quick measure measure Properties Calculations

1 Store 3 Sales (KEEPFILTERS) =
2 CALCULATE(
3 [Customer Sales],
4 KEEPFILTERS(
5 'Store Lookup'[store_id] = 3
6)
7)

product: Product Name: Customer Sales (KEEPFILTERS)

Product	Price	Customer Sales	Customer Sales (KEEPFILTERS)
Almond Croissant	\$13.50	\$3,163,528.61	\$19,006.73
Brazilian - Organic	\$8.88	\$3,163,528.61	\$8,993.33
Brazilian Lg	\$4,910.40	\$3,163,528.61	\$7,372.80
Brazilian Sm	\$22,624.88	\$3,163,528.61	\$21,693.00
Brazilian Lg	\$20,706.73	\$3,163,528.61	\$19,700.23
Cappuccino	\$14,291.55	\$3,163,528.61	\$14,397.60
Cappuccino Lg	\$25,121.25	\$3,163,528.61	\$21,847.19
Carmel syrup	\$27,817.31	\$3,163,528.61	\$29,340.64
Chili Mayan	\$1,956.60	\$3,163,528.61	\$4,029.60
Chocolate Chip Biscotti	\$2,879.28	\$3,163,528.61	\$1,604.48
Chocolate Croissant	\$7,937.60	\$3,163,528.61	\$9,399.00
Chocolate syrup	\$13,830.18	\$3,163,528.61	\$14,395.80
Civet Cat	\$2,430.00	\$3,163,528.61	\$4,027.20
Columbian Medium Roast	\$12,636.00	\$3,163,528.61	\$11,052.00
Total	\$1,780.00	\$3,163,528.61	\$2,808.00
	1,057,182.15	\$3,163,528.61	\$1,048,465.23

2) Change the formatting accordingly over here.

Visualizations > Fields

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales IF (COALESCE)
- Customer Sales IF (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL_Modified)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date Format(yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Customer Lookup

Page 1 Advanced CALCULATE Keep Remove Filters

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Store 5 Sales (KEEPFILTERS) Format: Currency Data category: Uncategorized

Home table: Measure Table \$ - % # , . : New Quick Measure measure Properties Calculations

1 Store 5 Sales (KEEPFILTERS) =
2 CALCULATE(
3 [Customer Sales],
4 KEEPFILTERS(
5 'Store Lookup'[store_id] = 5
6)
7)

product

	Profit	Total Profit	Store 5 Profit (KEEPFILTERS)	Store 5 Profit
Almond Croissant	11,500.50	\$3,163,528.61	\$19,006.73	
Brazilian - Organic	8,386.95	\$3,163,528.61	\$8,913.33	
Brazilian Lg	4,910.40	\$3,163,528.61	\$7,372.80	
Brazilian Sm	22,624.88	\$3,163,528.61	\$21,693.00	
Cappuccino	20,706.73	\$3,163,528.61	\$19,703.23	
Cappuccino Lg	14,201.55	\$3,163,528.61	\$14,397.60	
Carmel syrup	25,121.25	\$3,163,528.61	\$23,847.19	
Chili Mayan	1,956.60	\$3,163,528.61	\$4,029.60	
Chocolate Chip Biscotti	2,879.28	\$3,163,528.61	\$3,604.43	
Chocolate Croissant	7,937.60	\$3,163,528.61	\$9,399.90	
Chocolate syrup	10,830.18	\$3,163,528.61	\$14,395.80	
Civet Cat	2,430.00	\$3,163,528.61	\$4,027.20	
Columbian Medium Roast	12,636.00	\$3,163,528.61	\$11,052.00	
Total	1,780.00	\$3,163,528.61	\$2,808.00	
	1,057,182.15	\$3,163,528.61	\$1,048,465.23	

Visualizations > Fields >

Search

Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Scalar Measures
- Store 5 Sales (KEEPFILTERS)
- Store 5 Sales of whole Beans & Teas (ALL Modifier)
- Store 5 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- To & Best Practices
- Total Profit

Add data fields here

Drill through

Cross-report

Off —

Keep all filters:

- On

Add drill-through fields here

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half
- Year_ID

Page 1 Advanced CALCULATE Keep Remove Filters

2) Change the formatting accordingly over here.

1) Create the Store 5 Sales measure.

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Store 8 Sales (KEEPFILTERS) Home table: Measure Table

Format: Currency Data category: Uncategorized

2) Change the formatting accordingly over here.

1 Store 8 Sales (KEEPFILTERS) =
2 CALCULATE(
3 [Customer Sales],
4 KEEPFILTERS(
5 'Store Lookup'[store_id] = 8
6)
7)

product

	Profit	Customer Profit	Customer Profit (KEEPFILTERS)	Customer Profit (KEEPFILTERS)
Almond Croissant	\$1,500.50	\$3,163,528.61	\$19,006.73	\$19,006.73
Brazilian - Organic	\$3,988.95	\$3,163,528.61	\$6,933.33	\$6,933.33
Brazilian Lg	\$4,910.40	\$3,163,528.61	\$7,372.80	\$7,372.80
Brazilian Rg	\$22,624.88	\$3,163,528.61	\$21,693.00	\$21,693.00
Brazilian Sm	\$9,706.75	\$3,163,528.61	\$19,703.25	\$19,703.25
Cappuccino	\$14,201.55	\$3,163,528.61	\$14,397.99	\$14,397.99
Cappuccino Lg	\$25,121.25	\$3,163,528.61	\$21,847.19	\$21,847.19
Carmel syrup	\$27,817.31	\$3,163,528.61	\$29,340.94	\$29,340.94
Chili Mayan	\$1,956.60	\$3,163,528.61	\$4,029.60	\$4,029.60
Chocolate Chip Biscotti	\$2,879.28	\$3,163,528.61	\$3,604.43	\$3,604.43
Chocolate Croissant	\$7,937.60	\$3,163,528.61	\$9,399.98	\$9,399.98
Chocolate syrup	\$13,830.18	\$3,163,528.61	\$14,393.80	\$14,393.80
Civet Cat	\$2,430.00	\$3,163,528.61	\$4,027.20	\$4,027.20
Columbian Medium Roast	\$12,636.00	\$3,163,528.61	\$11,052.00	\$11,052.00
Total	\$1,067,182.15	\$3,163,528.61	\$1,048,465.23	\$1,048,465.23

2) Change the formatting accordingly over here.

Visualizations > Fields

Search: Measure Table

- % Quantity Sold to Females
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (CURRENCY)
- Customer Sales LY (SBLANG)
- Orders by Females (Wrong)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL_Matched)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 3 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)
- Trip & Bean Practices
- Total Profit
- Calendar
- Date format (yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_Half

Page 1 Advanced CALCULATE Keep Remove Filters +

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform refresh data New visual Text box More visuals Quick measure measure Calculations Publish

store_id

store_id	1	2	3	4
product_group	Store 3 Sales (KEEPFILTERS)	Store 5 Sales (KEEPFILTERS)	Store 8 Sales (KEEPFILTERS)	
Add-ons	\$11,500.00	\$19,006.73	\$17,442.40	
Beverages	\$10,912.00	\$22,988.00	\$17,160.00	
Food	\$112,984.95	\$1,073,219.50	\$1,085,934.10	
Merchandise	\$148,900.25	\$153,905.94	\$148,535.90	
Whole Bean/Teas	\$13,614.00	\$39,372.00	\$10,798.00	
Total	\$1,420,721.10	\$1,409,796.51	\$1,422,147.27	

product Profit Total Profit Store 3 Profit Store 5 Profit

product	Profit	Total Profit	Store 3 Profit (KEEPFILTERS)	Store 5 Profit
Dark chocolate Lg	94,948.00	\$3,163,528.61	\$30,429.00	\$30,429.00
Sustainably Grown Organic Lg	96,821.69	\$3,163,528.61	\$29,376.38	\$29,376.38
Cappuccino Lg	62,152.23	\$3,163,528.61	\$29,340.94	\$29,340.94
Latte Lg	66,511.64	\$3,163,528.61	\$28,183.88	\$28,183.88
Latte	78,050.13	\$3,163,528.61	\$28,116.88	\$28,116.88
Jamaican Coffee River Lg	74,500.31	\$3,163,528.61	\$25,680.04	\$25,680.04
Morning Sunrise Chai Lg	60,811.00	\$3,163,528.61	\$25,188.00	\$25,188.00
Sustainably Grown Organic Rg	74,278.13	\$3,163,528.61	\$24,322.50	\$24,322.50
Cappuccino	74,123.44	\$3,163,528.61	\$21,847.19	\$21,847.19
Ethiopia Lg	66,633.23	\$3,163,528.61	\$22,170.75	\$22,170.75
Brazilian Lg	68,232.63	\$3,163,528.61	\$21,691.00	\$21,691.00
Peppermint Lg	57,982.50	\$3,163,528.61	\$20,940.75	\$20,940.75
Ethiopia Rg	61,395.75	\$3,163,528.61	\$20,804.50	\$20,804.50
Earl Grey Lg	58,088.25	\$3,163,528.61	\$20,027.25	\$20,027.25
Columbian Medium Roast Lg	58,432.50	\$3,163,528.61	\$19,921.75	\$19,921.75
Total	\$1,163,528.61	\$3,163,528.61	\$1,048,465.20	\$1,048,465.20

We are now able to compare the performance of each of the stores side by side.

Perform the below actions.

- 1) Add the Store 3 Sales (KEEPFILTERS) measure to the rows
- 2) Add the Store 5 Sales (KEEPFILTERS) measure to the rows
- 3) Add the Store 8 Sales (KEEPFILTERS) measure to the rows

Key objective 2 is now complete. Use KEEPFILTERS to create measures that show sales for each of the 3 stores.

Page 1 Advanced CALCULATE Keep Remove Filters +

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Pages

store_id

product_group

Add-ons

Beverages

Food

Merchandise

Total

Store 3 Sales (KEEPPFILTERS)

Store 3 Sales (KEEPPFILTERS)

Store 3 Sales (KEEPPFILTERS)

\$16,912.00

\$22,888.00

\$17,160.00

\$10,912.00

\$22,360.00

\$17,160.00

\$1,122,144.95

\$1,073,219.50

\$1,085,304.19

\$44,999.95

\$153,905.84

\$148,381.59

\$10,614.00

\$19,372.00

\$10,794.00

\$1,487,226.40

\$1,789.78

\$1,404,744.87

product

Profit

Total Profit

Store 3 Profit (KEEPPFILTERS)

Store 5 Profit

Dark chocolate lg

Sustainable Green Organic lg

Cappuccino lg

Latte lg

Latte

Jamaican Coffee River lg

Morning Sunrise Chai lg

Sustainable Green Organic lg

Cappuccino

Espresso lg

Brasserie lg

Peppermint lg

Frappuccino lg

Vanilla lg

Columbian Medium Roast lg

Total

\$4,940.00

\$3,163,528.61

\$10,429.00

\$10,429.00

\$6,821.63

\$3,163,528.61

\$29,376.36

\$29,376.36

\$2,352.23

\$3,163,528.61

\$29,340.94

\$29,340.94

\$6,911.94

\$3,163,528.61

\$29,163.88

\$29,163.88

\$1,299.13

\$3,163,528.61

\$28,116.88

\$28,116.88

\$4,900.91

\$3,163,528.61

\$25,680.94

\$25,680.94

\$8,811.00

\$3,163,528.61

\$25,188.00

\$25,188.00

\$4,276.13

\$3,163,528.61

\$24,322.50

\$24,322.50

\$4,125.44

\$3,163,528.61

\$23,547.19

\$23,547.19

\$6,839.28

\$3,163,528.61

\$22,170.75

\$22,170.75

\$6,232.63

\$3,163,528.61

\$21,693.00

\$21,693.00

\$7,982.50

\$3,163,528.61

\$20,940.75

\$20,940.75

\$1,395.75

\$3,163,528.61

\$20,884.59

\$20,884.59

\$6,086.23

\$3,163,528.61

\$20,027.25

\$20,027.25

\$8,432.30

\$3,163,528.61

\$19,923.75

\$19,923.75

\$1,462,528.61

\$3,163,528.61

\$1,048,463.21

\$1,048,463.21

Filters

Search

Filters on this visual

product_group is not (Blank)

Filter type

Basic Filtering

Search

Select all

(Blank)

Add-ons

Beverages

Food

Merchandise

Whole Beans/Tea

Require single selection

Store 3 Sales (KEEPPFILTERS) is (All)

Store 5 Sales (KEEPPFILTERS) is (All)

Store 8 Sales (KEEPPFILTERS) is (All)

Store 3 Sales (KEEPPFILTERS)

Store 5 Sales (KEEPPFILTERS)

Store 8 Sales (KEEPPFILTERS)

Drill through

Cross-report

Off

On

Keep all filters

Add drill-through fields here

Filters on this page

Add data fields here

Visualizations

Fields

Measure Table

% Quantity Sold to Females

Cost (CURRENCY)

Customer Sales (Last Year)

Customer Sales LF (ODADESC)

Customer Sales LF (OSBLANK)

Orders by Females (Wrong)

Scalar Measures

Store 3 Sales (KEEPPFILTERS)

Store 3 Sales of whole Beans & Tea (ALL_Modifier)

Store 3 Sales of whole Beans & Tea (CALCULATE)

Store 5 Profit

Store 5 Profit (KEEPPFILTERS)

Store 5 Sales (KEEPPFILTERS)

Store 5 Sales (KEEPPFILTERS)

Store 8 Sales (KEEPPFILTERS)

Tip & Best Practices

Total Profit

Calendar

Date Format (yyyy-mm-dd)

Date_ID

INumber

Month_ID

Month_Name

Quarter & Year

Quarter_ID

Quarter_Name

Transaction Date

Week_Desc

Week_ID

Week_Year

Year Half

Page 1 Advanced CALCULATE Keep Remove Filters +

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name % Store Sales (REMOVEDFILTERS) Format Percentage Data category Uncategorized New Quick Create Calculations Properties

Home table Measure Table \$ - % , . ; : Structure Formatting

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL MODIFIER)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction Date
- Week_Desc
- Week_ID
- Week_Year

1 % Store Sales (REMOVEDFILTERS) =
2 VAR AllStoreSales =
3 CALCULATE(
4 [Customer Sales],
5 REMOVEFILTERS(
6 'Store Lookup'[store_id]
7)
8)
9 VAR Ratio =
10 DIVIDE(
11 [Customer Sales],
12 AllStoreSales
13)
14 RETURN
15 Ratio

2) Change the formatting accordingly over here.

1) Objective 3 to use REMOVEFILTERS with variables to create a single measure for % of Store Sales.. Create the % Store Sales (REMOVEDFILTERS) measure.

Page 1 Advanced CALCULATE Keep Remove Filters +

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Out Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data - New visual Text box More visuals - Quick measure measure Publish

Calculated

Data

Queries

Insert Calculations Share

store_id

product_group	Store 3 Sales (KEEPPFILTERS)	Store 5 Sales (KEEPPFILTERS)	Store 8 Sales (KEEPPFILTERS)	% Store Sales (REMOVEDFILTERS)
Add-ons	\$15,912.00	\$22,988.00	\$17,669.00	100.00%
	\$10,912.00	\$12,988.00	\$17,160.00	21.37%
			\$17,160.00	43.02%
				33.61%
Beverages	\$1,122,964.95	\$1,071,279.50	\$1,063,934.50	100.00%
Food	\$146,966.75	\$153,966.50	\$148,935.50	100.00%
Merchandise	\$11,614.00	\$16,177.00	\$16,768.00	100.00%
Total	\$1,487,220.60	\$1,390,789.75	\$1,404,744.87	100.00%

product

Profit	Total Profit	Store 3 Profit (KEEPPFILTERS)	Store 5 Profit
Dark chocolate lg	\$4,848.00	\$3,163,528.61	\$30,429.00
Sustainable Green Organic lg	\$6,821.63	\$3,163,528.61	\$29,379.38
Cappuccino lg	\$2,352.23	\$3,163,528.61	\$29,345.34
Latte lg	\$6,311.94	\$3,163,528.61	\$28,763.88
Latte	\$1,298.13	\$3,163,528.61	\$24,116.88
Jamaican Coffee River lg	\$4,930.51	\$3,163,528.61	\$25,620.94
Morning Sunrise Chai lg	\$0,831.00	\$3,163,528.61	\$25,188.00
Sustainable Green Organic lg	\$4,276.13	\$3,163,528.61	\$24,922.50
Cappuccino	\$4,123.44	\$3,163,528.61	\$23,947.19
Latte lg	\$6,839.21	\$3,163,528.61	\$22,170.75
Brasserie lg	\$8,232.43	\$3,163,528.61	\$21,899.00
Peppermint lg	\$7,962.55	\$3,163,528.61	\$20,945.75
Latte lg	\$1,395.73	\$3,163,528.61	\$20,864.50
Vanilla lg	\$4,586.23	\$3,163,528.61	\$20,027.25
Columbian Medium Roast lg	\$6,432.50	\$3,163,528.61	\$19,923.75
Total	\$1,463,528.61	\$3,163,528.61	\$1,048,465.25

3) Add in the “% Store Sales (REMOVEDFILTERS) ” measure to the matrix. You will notice that the percentages appear within the table.

Key objective 3 is now complete. The calculations have been performed correctly. Each column represents 100% & each of the product group contains the store percentage split breakdown. The filter context has been removed from the store & not the product group. Percentage context is provided for each store based on the category.

Filters

Visualizations

Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (SEALANT)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL, Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 3 Profit
- Store 5 Profit (KEEPPFILTERS)
- Store 5 Sales (KEEPPFILTERS)
- Store 5 Sales (KEEPPFILTERS)
- Store 8 Sales (KEEPPFILTERS)
- Store 8 Sales (KEEPPFILTERS)

Drill through

Cross-report

Keep all filters

Add drill-through fields here

Filters on this page

Page 1 Advanced CALCULATE Keep Remove filters +

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get Data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure Quick calculate Share

4) Matrix visual output when selecting store id 3 from the slicer Merchandise has the highest percentage in store sales i.e. 40.12%.

The screenshot shows a Power BI desktop interface with a matrix visual on the left and various toolbars and panes on the right. A blue arrow points from the text above to the matrix visual. The matrix visual displays sales data for different products across four stores. The columns are labeled: product name, Store 3 Sales (\$169,769), Store 3 Sales (\$169,769), Store 3 Sales (\$169,769), and % Store Sales (%MOVEDFILTER). The rows include Add-ons, 3, Beverages, Food, Merchandise, Whole Beans/Tea, and Total. The 'Merchandise' row has the highest percentage at 40.12%. A blue arrow also points to the '% Store Sales' column header in the matrix.

product name	Store 3 Sales (\$169,769)	Store 3 Sales (\$169,769)	Store 3 Sales (\$169,769)	% Store Sales (%MOVEDFILTER)
Add-ons	\$16,912.00			21.37%
3	\$10,912.00			11.27%
Beverages	\$1,122,964.93			34.21%
Food	\$148,960.25			32.99%
Merchandise	\$33,614.00			40.12%
Whole Beans/Tea	\$96,829.40			27.16%
Total	\$1,487,779.60			33.48%

product	Profit	Total Profit	Store 3 Profit (\$169,769)	Store 3 Profit
Almond Croissant	13,500.00	\$3,163,328.61	\$19,006.73	
Brassien - Organic	8,386.93	\$3,163,328.61	\$8,903.33	
Brassien Lg	22,624.00	\$3,163,328.61	\$7,372.00	
Brassien lg	22,726.73	\$3,163,328.61	\$21,693.00	
Brassien Sm	14,201.35	\$3,163,328.61	\$14,380.93	
Cappuccino	29,121.21	\$3,163,328.61	\$23,847.19	
Cappuccino Lg	27,897.21	\$3,163,328.61	\$29,342.84	
Caramel syrup	1,956.43	\$3,163,328.61	\$4,029.60	
Chili Mayan	3,879.28	\$3,163,328.61	\$8,854.43	
Chocolate Chio Biscotti	2,837.60	\$3,163,328.61	\$9,399.88	
Chocolate Croissant	13,293.18	\$3,163,328.61	\$14,398.80	
Chocolate syrup	2,450.00	\$3,163,328.61	\$4,017.20	
Civet Cat	12,896.00	\$3,163,328.61	\$11,062.00	
Colombian Medium Roast	3,780.00	\$3,163,328.61	\$2,808.00	
Total	1,957,182.15	\$3,163,328.61	\$1,648,465.23	

Filters

Visualizations

Fields

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Scalar Measures

- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL, Modified)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPPFILTERS)
- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales (CALCULATE)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year

Page 1 Advanced CALCULATE Keep Remove Filters +

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Get data - Excel Power BI datasets - SQL Server Enter data Recent sources - Transform Refresh data - New visual Text box More visuals - Quick measure measure Publish

Store 3 Sales (%REMOVEDFILTERS) Store 3 Sales (%REMOVEDFILTERS) Store 3 Sales (%REMOVEDFILTERS) % Store Sales (%REMOVEDFILTERS)

product_group Add-ons 116,912.00 \$12,988.00 21.37% 64.91%
3 110,912.00 112,000.00 43.07%
Beverages \$1,122,964.95 \$1,073,219.39 66.91%
Food \$148,900.25 \$153,900.94 67.09%
Merchandise \$33,814.00 \$39,372.00 87.11%
Whole Restaurants 160,879.00 1101,354.14 17.47%
Total \$1,487,220.00 \$1,390,789.39 66.54%

product Profit Total Profit Store 3 Profit (%REMOVEDFILTERS) Store 3 Profit
Dark chocolate Lg 45,748.00 \$3,143,528.61 \$30,429.00 100,429.00
Sustainably Grown Organic Lg 64,185.56 \$3,143,528.61 \$29,376.36 129,376.36
Cappuccino Lg 17,158.23 \$3,143,528.61 \$23,345.34 129,345.34
Latte Pg 17,814.06 \$3,143,528.61 \$26,183.88 126,183.88
Latte 33,589.69 \$3,143,528.61 \$24,116.88 124,116.88
Jamaican Coffee River Lg 91,798.44 \$3,143,528.61 \$25,650.94 125,650.94
Morning Sunrise Chai Lg 34,327.03 \$3,143,528.61 \$21,188.00 121,188.00
Sustainably Grown Organic Pg 48,952.19 \$3,143,528.61 \$24,322.50 124,322.50
Cappuccino 48,952.44 \$3,143,528.61 \$21,947.19 123,947.19
Espresso Lg 40,370.86 \$3,143,528.61 \$23,170.75 123,170.75
Briskin Lg 44,317.88 \$3,143,528.61 \$21,689.00 121,689.00
Peppermint Lg 40,182.50 \$3,143,528.61 \$20,940.75 120,940.75
Erico's Pg 40,974.21 \$3,143,528.61 \$24,884.59 129,884.59
Earl Grey Lg 40,041.00 \$3,143,528.61 \$20,007.25 120,007.25
Columbian Medium Roast Lg 28,870.00 \$3,143,528.61 \$19,923.75 119,923.75
Total 2,195,647.38 \$3,143,528.61 \$1,048,465.25 \$1,048,465.25

Filters Visualizations Fields

Search

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Scalar Measures

Store 3 Sales (%REMOVEDFILTERS)

Store 3 Sales of whole Beans & Teas (ALL, Modified)

Store 3 Sales of whole Beans & Teas (CALCULATE)

Store 3 Profit

Store 3 Profit (%REMOVEDFILTERS)

Store 3 Sales (%REMOVEDFILTERS)

Store 3 Sales (%REMOVEDFILTERS)

Store 3 Sales (%REMOVEDFILTERS)

Tip & Best Practices

Total Profit

Calendar

Date Format (yyyy-mm-dd)

Date_ID

ISNUMBER

Month_ID

Month_Name

Quarter & Year

Quarter_ID

Quarter_Name

Transaction_Date

Week_Desc

Week_ID

Week_Year

Page 1 Advanced CALCULATE Keep Remove Filters +

5) Another visual example with store 3 & 5 displayed for comparative analysis.

PRO TIP: Common CALCULATE Patterns

Cumulative Total

```
1 Cumulative Total =  
2 CALCULATE(  
3     SUM(  
4         'Sales by Store'[quantity_sold]  
5     ),  
6     FILTER(  
7         ALL(  
8             'Calendar'[Transaction_Date]  
9         ),  
10        'Calendar'[Transaction_Date] <= MAX('Calendar'[Transaction_Date])  
11    )  
12 )
```

Percent of Total

```
1 Percent of Total Sales =  
2 VAR CurrentSales =  
3 SUMX(  
4     'Sales by Store',  
5     'Sales by Store'[unit_price] * 'Sales by Store'[quantity_sold]  
6 )  
7 VAR ALLStoreSales =  
8 CALCULATE(  
9     SUMX(  
10        'Sales by Store',  
11        'Sales by Store'[unit_price] * 'Sales by Store'[quantity_sold]  
12    ),  
13    ALL(  
14        'Sales by Store'  
15    ))  
16 )  
17 VAR Ratio =  
18 DIVIDE(  
19     CurrentSales,  
20     ALLStoreSales  
21 )  
22 RETURN  
23 Ratio
```

Overall Total

```
1 Overall Total =  
2 CALCULATE(  
3     SUMX(  
4         'Sales by Store',  
5         'Sales by Store'[unit_price] * 'Sales by Store'[quantity_sold]  
6     ),  
7     ALL(  
8         'Sales by Store'  
9     ))  
10 )
```

6) Table & Filter Functions

Introduction - Table & Filter Functions



Table and **Filter** functions return **columns** or **tables** rather than scalar values, and can be used to either generate new data or serve as table inputs within DAX measures

TOPICS WE'LL COVER:

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

COMMON USE CASES:

- *Returning columns or tables which are subsets of existing data sources*
- *Adding new columns of data to an existing table*
- *Filtering out totals from measure results*
- *Reducing the number of rows returned from a table*
- *Generating new data from scratch*

Common Table & Filter Functions

Filter Data

Functions used to return filtered tables or filter results of measures

Common Examples:

- ALL
- FILTER
- DISTINCT
- VALUES
- ALLEXCEPT
- ALLSELECTED

Add Data

Functions used to specify or add columns based on existing data in the model

Common Examples:

- SELECTCOLUMNS
- ADDCOLUMNS
- SUMMARIZE

Create Data

Functions used to generate new rows, columns & tables from scratch

Common Examples:

- ROW
- DATATABLE
- GENERATESERIES
- {} Table Constructor

Review: Calculated Tables

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

DAX functions with **table** arguments can typically accept either **physical tables (i.e. 'Sales by Store')** or **calculated, virtual tables (with functions like FILTER, VALUES, etc.)**

=SUMX(Table, Expression)

```
1 Total Sales =  
2 SUMX(  
3     'Sales by Store',  
4     'Sales by Store'[quantity_sold])
```

```
1 Total Sales =  
2 SUMX(  
3     FILTER(  
4         'Sales by Store',  
5         'Sales by Store'[quantity_sold] > 3  
6     ),  
7     'Sales by Store'[quantity_sold] )
```

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

FILTER()

Returns a filtered table, based on one or more filter expressions

=FILTER(Expression, [Filter1], [Filter2],...)

- FILTER is both a **table function** and an **iterator**
- Often used to **reduce** the number of rows to scan

ALL()

Returns all the rows in a table, or all the values in a column, ignoring any filters

=ALL(Table or ColumnName, [ColumnName1],...)

- ALL is both a **table filter** and a **CALCULATE modifier**
- Removes initial **filter context**
- Does not accept **table expressions** (*only physical table references*)

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

STEP 1 Measure is written

```
1 Sales Where Order Quantity More Than 3 =  
2 CALCULATE(  
3     [Customer Sales],  
4     FILTER(  
5         'Sales by Store',  
6         'Sales by Store'[Quantity_Sold] >3  
7     )  
8 )
```

\$29,643

Sales Where Order Quantity More Than 3

STEP 3

DAX evaluates [Customer Sales] against the filtered table (quantity sold > 3)

STEP 2 FILTER creates a virtual, calculated table

Order ID	Customer Name	Product ID	Product Name	Quantity Sold	Unit Price	Total Price
101	John Doe	1001	Product A	10	\$10.00	\$100.00
102	Jane Smith	1002	Product B	20	\$20.00	\$400.00
103	Bob Johnson	1003	Product C	15	\$30.00	\$450.00
104	Sarah Lee	1004	Product D	5	\$40.00	\$200.00
105	Mike Williams	1005	Product E	10	\$50.00	\$500.00
106	Alice Green	1006	Product F	15	\$60.00	\$900.00
107	David Blue	1007	Product G	20	\$70.00	\$1400.00
108	Emily Red	1008	Product H	10	\$80.00	\$800.00
109	Frank Yellow	1009	Product I	15	\$90.00	\$1350.00
110	Gina Purple	1010	Product J	20	\$100.00	\$2000.00
111	Henry Orange	1011	Product K	10	\$110.00	\$1100.00
112	Ivy Green	1012	Product L	15	\$120.00	\$1800.00
113	Jack Blue	1013	Product M	20	\$130.00	\$2600.00
114	Karen Red	1014	Product N	10	\$140.00	\$1400.00
115	Liam Purple	1015	Product O	15	\$150.00	\$2250.00
116	Mia Orange	1016	Product P	20	\$160.00	\$3200.00
117	Noah Green	1017	Product Q	10	\$170.00	\$1700.00
118	Olivia Blue	1018	Product R	15	\$180.00	\$2700.00
119	William Red	1019	Product S	20	\$190.00	\$3800.00
120	Ella Purple	1020	Product T	10	\$200.00	\$2000.00

DISTINCT

DISTINCT()

Returns a **single column** table of unique values when a **column name** is given.
If a **table** is supplied, DISTINCT returns all unique combinations of values.

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

=DISTINCT(Column**Name** or Table**Expression**)

The **column** you want to extract
unique values from

Examples:

- 'Sales by Store'[CustomerID]
- 'Sales by Store'[UnitPrice]

The **table** you want to extract unique
combinations of values from

Examples:

- 'Sales by Store'
- 'Food Inventory'



PRO TIP:

Trying to build a relational model from a single, blended table? Use DISTINCT to create new lookup/dimension tables by extracting unique values from fields in your data table!

File Home Help Table tools

Name: Calendar

Mark as date table Manage relationships New measure New measure column New table

Structure Calendars

Write a DAX expression to create a new table.

1) Click here to create a new table.

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	Dimensions	Date Format (Type: mm-yy)	Year_Half	Quarter & Year
01/07/2017	201707001	26	Week 26	7	July	1	Q3	2017	Week 26-2017	True	2017-07-01	2H	Q3-2017
02/07/2017	201707002	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-02	2H	Q3-2017
03/07/2017	201707003	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-03	2H	Q3-2017
04/07/2017	201707004	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-04	2H	Q3-2017
05/07/2017	201707005	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-05	2H	Q3-2017
06/07/2017	201707006	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-06	2H	Q3-2017
07/07/2017	201707007	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-07	2H	Q3-2017
08/07/2017	201707008	27	Week 27	7	July	1	Q3	2017	Week 27-2017	True	2017-07-08	2H	Q3-2017
09/07/2017	201707009	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-09	2H	Q3-2017
10/07/2017	201707010	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-10	2H	Q3-2017
11/07/2017	201707011	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-11	2H	Q3-2017
12/07/2017	201707012	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-12	2H	Q3-2017
13/07/2017	201707013	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-13	2H	Q3-2017
14/07/2017	201707014	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-14	2H	Q3-2017
15/07/2017	201707015	28	Week 28	7	July	1	Q3	2017	Week 28-2017	True	2017-07-15	2H	Q3-2017
16/07/2017	201707016	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-16	2H	Q3-2017
17/07/2017	201707017	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-17	2H	Q3-2017
18/07/2017	201707018	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-18	2H	Q3-2017
19/07/2017	201707019	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-19	2H	Q3-2017
20/07/2017	201707020	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-20	2H	Q3-2017
21/07/2017	201707021	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-21	2H	Q3-2017
22/07/2017	201707022	29	Week 29	7	July	1	Q3	2017	Week 29-2017	True	2017-07-22	2H	Q3-2017
23/07/2017	201707023	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-23	2H	Q3-2017
24/07/2017	201707024	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-24	2H	Q3-2017
25/07/2017	201707025	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-25	2H	Q3-2017
26/07/2017	201707026	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-26	2H	Q3-2017
27/07/2017	201707027	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-27	2H	Q3-2017
28/07/2017	201707028	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-28	2H	Q3-2017
29/07/2017	201707029	30	Week 30	7	July	1	Q3	2017	Week 30-2017	True	2017-07-29	2H	Q3-2017
30/07/2017	201707030	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True	2017-07-30	2H	Q3-2017
31/07/2017	201707031	31	Week 31	7	July	1	Q3	2017	Week 31-2017	True	2017-07-31	2H	Q3-2017
01/08/2017	201708001	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True	2017-08-01	2H	Q3-2017
02/08/2017	201708002	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True	2017-08-02	2H	Q3-2017
03/08/2017	201708003	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True	2017-08-03	2H	Q3-2017
04/08/2017	201708004	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True	2017-08-04	2H	Q3-2017
05/08/2017	201708005	31	Week 31	8	August	1	Q3	2017	Week 31-2017	True	2017-08-05	2H	Q3-2017
06/08/2017	201708006	32	Week 32	8	August	1	Q3	2017	Week 32-2017	True	2017-08-06	2H	Q3-2017
07/08/2017	201708007	32	Week 32	8	August	1	Q3	2017	Week 32-2017	True	2017-08-07	2H	Q3-2017

Fields

Search

- Measure Table
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store
- Store Lookup

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name: DISTINCT Example

Mark as date table - Manage relationships New measure Quick New measure column New table

Measure Calculations

1 DISTINCT Example =
2 DISTINCT(
3 'Product Lookup'[product_group]
4)

product_group ▾
Whole Bean/Tea
Beverages
Add-ons
Food
Merchandise

Fields

Search

Measure Table
Calendar
Customer Lookup
DISTINCT Example
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
Store Lookup

2) The code above will output the unique values within the product group field located in the Product Lookup table.

Maven Routes - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: product_group
Data type: Text

Summarization: Don't summarize
Data category: Uncategorized

Sort by column
Data groups
Manage relationships
New column

Structure
Formatting
Properties
Sum
Group
Relationship
Observations

Fields

Search

Measure Table
Calendar
Customer Lookup
DISTINCT Example
Employee Lookup
Food Inventory
Product Lookup

product_group
product_id
product_name
product_category
product_type
product
product_description
unit_of_measure
current_wholesale_price
current_retail_price
tax_exempt_y/n
promo_y/n
current_cost

3) The 5 distinct values within the product group field can be validated here.

product_id	product_group	product_category	product_type	product	product_description	unit_of_measure	current_wholesale_price	current_retail_price	tax_exempt_y/n	promo_y/n	current_cost
1	Whole Bean/Teas	Coffee beans	Organic Beans	Brasilian - Organic	It's like Carnival in a cup. Clean and smooth.	12 oz	24.4	28 Y	N	18	
2	Whole Bean/Teas	Coffee beans	House Blend Beans	Our Old Time Diner Blend	Our packed blend of beans that is reminiscent of the cup.	12 oz	24.4	28 Y	N	18	
3	Whole Bean/Teas	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.	1 lb	21.0	24.75 Y	N	2.95	
4	Whole Bean/Teas	Coffee beans	Espresso Beans	Promo Espresso Roast	Our premium single source of hand roasted beans.	1 lb	26.88	30.45 Y	N	4.68	
5	Whole Bean/Teas	Coffee beans	Gourmet Beans	Colombian Medium Roast	A smooth cup of coffee any time of day.	1 lb	27	33 Y	N	3	
6	Whole Bean/Teas	Coffee beans	Gourmet Beans	Ethiopia	From the home of coffee.	1 lb	29.0	33 Y	N	4.2	
7	Whole Bean/Teas	Coffee beans	Premium Beans	Jamaican Coffee River	Yea man, it will start your day off right.	1 lb	29.0	33.75 Y	N	3.95	
8	Whole Bean/Teas	Coffee beans	Premium Beans	Cover Cat	The most expensive coffee in the world. The cat's got all the beans.	1 lb	38	43 Y	N	8	
9	Whole Bean/Teas	Coffee beans	Organic Beans	Organic Decaf Blend	Our blend of hand picked organic beans that have been roasted.	1 lb	28	32.5 Y	N	4.5	
10	Whole Bean/Teas	Coffee beans	Green beans	Guatemalan Sustainably Grown	Green beans you can roast yourself.	1 lb	8	10 Y	N	2	
11	Whole Bean/Teas	Loose Tea	Herbal tea	Lemon Grass	You will think you are Thailand as you sip your cups.	9 oz	7.20	8.55 Y	N	0.6	
12	Whole Bean/Teas	Loose Tea	Herbal tea	Peppermint	Cool and refreshing to help calm your nerves.	9 oz	7.20	8.55 Y	N	0.6	
13	Whole Bean/Teas	Loose Tea	Black tea	English Breakfast	The traditional cup to start your day.	9 oz	7.20	8.55 Y	N	0.6	
14	Whole Bean/Teas	Loose Tea	Black tea	Earl Grey	A full leaf of Ceylon Peppermint blended with organic oil of bergamot.	9 oz	7.20	8.55 Y	N	0.6	
15	Whole Bean/Teas	Loose Tea	Green tea	Serenity Green Tea	Mountain grown and harvested at the optimal time.	3 oz	7.4	8.25 Y	N	0.6	
16	Whole Bean/Teas	Loose Tea	Chai tea	Traditional Blend Chai	A traditional blend.	9 oz	7.20	8.55 Y	N	0.6	
17	Whole Bean/Teas	Loose Tea	Chai tea	Morning Sunrise Chai	Fair trade and organic and has a warm finish.	9 oz	7.20	8.55 Y	N	0.6	
18	Whole Bean/Teas	Loose Tea	Chai tea	Spicy Eye Opener Chai	A zinger blend to awaken your taste buds.	9 oz	8.70	10.55 Y	N	0.6	
19	Whole Bean/Teas	Packaged Chocolate	Drinking Chocolate	Dark chocolate	This drinking chocolate is smooth and creamy.	1 lb	8.12	8.4 Y	N	1.25	
20	Whole Bean/Teas	Packaged Chocolate	Organic Chocolate	Sustainably Grown Organic	Certified organic containing the highest quality ingredients.	1 lb	8.00	7.6 Y	N	1.55	
21	Whole Bean/Teas	Packaged Chocolate	Drinking Chocolate	Chili Mayan	Pregnant with spices, this is the most flavourful drinking chocolate.	1 lb	20.00	22.55 Y	N	3.00	
22	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Sm	An honest cup o' coffee.	8 oz	1	2 Y	N	0.5	
23	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Lg	An honest cup o' coffee.	16 oz	2.25	2.5 Y	N	0.625	
24	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Lg	An honest cup o' coffee.	24 oz	2.5	3 Y	N	0.75	
25	Beverages	Coffee	Organic brewed coffee	Brasilian Lg	It's like Carnival in a cup. Clean and smooth.	8 oz	2.1	2.2 Y	N	0.55	
26	Beverages	Coffee	Organic brewed coffee	Brasilian Pg	It's like Carnival in a cup. Clean and smooth.	16 oz	2.5	3 Y	N	0.75	
27	Beverages	Coffee	Organic brewed coffee	Brasilian Pg	It's like Carnival in a cup. Clean and smooth.	24 oz	2.75	3.5 Y	N	0.875	
28	Beverages	Coffee	Gourmet brewed coffee	Colombian Medium Bean Sm	A smooth cup of coffee any time of day.	8 oz	1	2 Y	N	0.5	
29	Beverages	Coffee	Gourmet brewed coffee	Colombian Medium Bean Pg	A smooth cup of coffee any time of day.	16 oz	2.25	2.5 Y	N	0.625	
30	Beverages	Coffee	Gourmet brewed coffee	Colombian Medium Bean Lg	A smooth cup of coffee any time of day.	24 oz	2.5	3 Y	N	0.75	
31	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Sm	A bold cup when you want that something extra.	8 oz	2.1	2.2 Y	N	0.55	
32	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Pg	A bold cup when you want that something extra.	16 oz	2.5	3 Y	N	0.75	
33	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Lg	A bold cup when you want that something extra.	24 oz	2.75	3.5 Y	N	0.875	
34	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Sm	Smell a frosty number for good premium coffee.	8 oz	1.225	2.45 Y	N	0.6125	
35	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Pg	Smell a frosty number for good premium coffee.	16 oz	2.25	2.5 Y	N	0.75	
36	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Lg	Smell a frosty number for good premium coffee.	24 oz	2.625	3.75 Y	N	0.8875	

Table: Product Lookup (36 rows) Column: product_group (3 distinct values)

VALUES

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

VALUES()

Returns a **single column** table of unique values when a **column name** is given. If a table name is supplied, VALUES returns the entire table (including duplicates) plus a blank row

=VALUES(Table Name or Column Name)

The **table** you want to pull all columns
and all rows from (not unique)

Examples:

- 'Sales by Store'
- 'Food Inventory'

The **column** you want to extract
unique values from

Examples:

- 'Sales by Store'[CustomerID]
- 'Sales by Store'[UnitPrice]

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

Product Lookup

Product ID	Product
12	Coffee
13	Tea
14	Pastry

blank

blank

Sales by Store

Product ID	Quantity
12	2
13	6
14	1
14	3
15	5
16	6

In this example we have a Product Lookup table containing three product IDs (12, 13, 14)

But our fact table ('Sales by Store') contains additional product IDs NOT included in the Product Lookup (15, 16)

- Instead of throwing an error when a value is missing from a lookup table, the DAX engine adds a **blank row** (which will appear in visuals when missing values are present)
- Different table functions handle the presence of this blank row differently; for example, **VALUES** will always show the blank row but **DISTINCT** will not

PRO TIP:

If you think you might have missing values in your lookup tables (or aren't sure), use **VALUES** to check

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name: VALUES Example

Mark as data table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 VALUES Example =
2 VALUES(
3 'Product Lookup'[product_category]
4)

product_category

- Coffee beans
- Loose Tea
- Packaged Chocolate
- Coffee
- Tee
- Drinking Chocolate
- Flavours
- Bakery
- Branded

Fields

Search

- Measure Table
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
 - current_cost
 - current_retail_price
 - current_wholesale_product
 - product_category
 - product_descript...
 - product_group
 - product_id
 - product_type
 - prompt_yn
 - tax_exempt_yn
 - unit_of_measure
- Sales by Store
- Store Lookup
- VALUES Example

1) The 10 distinct values within the product category field have been provided including the last blank row.

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name: VALUES Example

Mark as data table -

Manage relationships

New measure

Quick

New measure column

New table

Calculated

Measure

1 VALUES Example =
2 VALUES(
3 'Product Lookup'[product_group]
4)

product_group

- Whole Bean/Tea
- Beverages
- Add-on
- Food
- Merchandise

Fields

Search

- Measure Table
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
 - current_cost
 - current_retail_price
 - current_wholesale_product
 - product_category
 - product_description
 - product_group
 - product_id
 - product_type
 - promo_yn
 - tax_exempt_yn
 - unit_of_measure
- Sales by Store
- Store Lookup
- VALUES Example

1) The 6 distinct values within the product group field have been provided including the last blank row. Using DISTINCT previously we had 5 values displayed.

Maven Roentgen - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Server Enter data sources Refresh data New visual Text box More visuals Transform data Quick measure Publish Calculations Share

product_category Customer Sales

product_category	Customer Sales
	\$49,949.63
Bakery	\$451,341.69
Branded	\$83,784.00
Coffee	\$1,651,861.15
Coffee beans	\$240,414.95
Drinking Chocolate	\$439,942.50
Flavours	\$51,060.00
Loose Tea	\$68,451.90
Packaged Chocolate	\$25,584.16
Tea	\$1,190,314.90
Total	\$4,252,704.88

Perform the below actions.

- 1) Add a matrix within the report
- 2) Add Product category field to the matrix
- 3) Add Customer Sales field to the matrix

Visualizations Fields

Search

Measure Table

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

VALUES Example

product_group

Add data fields here

Values

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Table & Filter Functions

Page 1 Advanced CALCULATE Keep Remove Filters

Page 4 of 4

ASSIGNMENT: VALUES & DISTINCT BLANK ROW

KEY OBJECTIVES

- 1) Check if the Ginger Scone product exists in the data model.
- 2) Use counting functions plus DISTINCT & VALUES to create a view that shows the blank row.
- 3) Update the data model to include the new product & use add a new visual to show the missing product id when the blank product is selected.

Maven Roasters - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: product Data type: Text Summarization: Don't summarize Data category: Uncategorized Sort by column Sort Data groups Group Manage relationships New column Calculations

product_id product_group product_category product_type product product_description unit_of_measure current_wholesale_price current_retail_price tax_exempt_yn promo_yn current_cost

1) Objective 1 to check if the Ginger Scone product exists in the data model. The below check confirms that the Ginger Scone product does not exist.

Fields

Search

Measure Table Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup current_cost current_retail_price current_wholesale_price product product_category product_description product_group product_id product_type promo_yn tax_exempt_yn unit_of_measure Sales by Store Store Lookup VALUES Example

Key objective 1 is now complete. Check if the Ginger Scone product exists in the data model.

Table: Product Lookup (86 rows) Column: product (86 distinct values)
1 Whole Bean/Tee Coffee beans Organic Beans Brazilian - Organic Our Old Time Diner Blend
2 Whole Bean/Tee Coffee beans House Blend Beans Espresso Beans Espresso Roast
3 Whole Bean/Tee Coffee beans Espresso Beans Prime Espresso Roast
4 Whole Bean/Tee Coffee beans Gourmet Beans Columbian Medium Roast
5 Whole Bean/Tee Coffee beans Gourmet Beans Ethiopia
6 Whole Bean/Tee Coffee beans Premium Beans Jamaican Coffee River
7 Whole Bean/Tee Coffee beans Premium Beans Civet Cat
8 Whole Bean/Tee Coffee beans Organic Beans Organic Decaf Blend
9 Whole Bean/Tee Coffee beans Green beans Guatemalan Sustainability Green
10 Whole Bean/Tee Loose Tea Herbal tea Lemon Grass
11 Whole Bean/Tee Loose Tea Herbal tea Peppermint
12 Whole Bean/Tee Loose Tea Black tea English Breakfast
13 Whole Bean/Tee Loose Tea Black tea Earl Grey A full leaf of Orange Pekoe blended with organic oil of bergamot.
14 Whole Bean/Tee Loose Tea Green tea Serenity Green Tea Mountain grown and harvested at the optimal time.
15 Whole Bean/Tee Loose Tea Chai tea Traditional Blend Chai A traditional blend.
16 Whole Bean/Tee Loose Tea Chai tea Morning Sunrise Chai Fair trade and organic and has a warm flavor.
17 Whole Bean/Tee Loose Tea Chai tea Spicy Eye Opener Chai A spicier blend to awaken your taste buds.
18 Whole Bean/Tee Packaged Chocolate Drinking Chocolate Dark chocolate This drinking chocolate is smooth and creamy.
19 Whole Bean/Tee Packaged Chocolate Organic Chocolate Sustainably Grown Organic Certified organic; containing the highest quality ingredients.
20 Whole Bean/Tee Packaged Chocolate Drinking Chocolate Chili Mayan Fragrant with spices, this is the most flavorful drinking chocolate.
21 Beverages Coffee Drip coffee Our Old Time Diner Blend Sm An honest cup o' coffee.
22 Beverages Coffee Drip coffee Our Old Time Diner Blend Md An honest cup o' coffee.
23 Beverages Coffee Drip coffee Our Old Time Diner Blend Lg An honest cup o' coffee.
24 Beverages Coffee Organic brewed coffee Brazilian Sm It's like Carnaval in a cup. Clean and smooth.
25 Beverages Coffee Organic brewed coffee Brazilian Md It's like Carnaval in a cup. Clean and smooth.
26 Beverages Coffee Organic brewed coffee Brazilian Lg It's like Carnaval in a cup. Clean and smooth.
27 Beverages Coffee Gourmet brewed coffee Colombian Medium Roast Sm A smooth cup of coffee any time of day.
28 Beverages Coffee Gourmet brewed coffee Colombian Medium Roast Md A smooth cup of coffee any time of day.
29 Beverages Coffee Gourmet brewed coffee Colombian Medium Roast Lg A smooth cup of coffee any time of day.
30 Beverages Coffee Gourmet brewed coffee Ethiopia Sm A bold cup when you want that something extra.
31 Beverages Coffee Gourmet brewed coffee Ethiopia Md A bold cup when you want that something extra.
32 Beverages Coffee Gourmet brewed coffee Ethiopia Lg A bold cup when you want that something extra.
33 Beverages Coffee Premium brewed coffee Jamaican Coffee River Sm Still a front runner for good premium coffee.
34 Beverages Coffee Premium brewed coffee Jamaican Coffee River Md Still a front runner for good premium coffee.
35 Beverages Coffee Premium brewed coffee Jamaican Coffee River Lg Still a front runner for good premium coffee.
36 Beverages Coffee Barista Espresso Espresso shot You will think you are in Venice when you sip this one.
37 Beverages Coffee Barista Espresso Latte You will think you are in Venice when you sip this one.

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Out data - Excel Power BI datasets SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

Dataset Data Queries Calculations Share

product_category Customer Sales

product_category	Customer Sales
Bakery	\$451,341.69
Branded	\$83,784.00
Coffee	\$1,651,861.15
Coffee beans	\$240,414.95
Drinking Chocolate	\$439,942.50
Flavours	\$51,050.00
Loose Tea	\$68,451.90
Packaged Chocolate	\$25,564.16
Tea	\$1,190,314.90
Total	\$4,252,704.88

product

- Almond Croissant
- Brazilian - Organic
- Brazilian Lg
- Brazilian Rg
- Brazilian Sm
- Cappuccino
- Cappuccino Lg
- Carmel syrup
- Chili Mayan
- Chocolate Chip Biscotti
- Chocolate Croissant
- Chocolate syrup
- Civet Cat
- Columbian Medium Roast
- Columbian Medium Roast Lg
- Columbian Medium Roast Rg
- Columbian Medium Roast Sm
- Cranberry Scone
- Croissant
- Dark chocolate
- Dark chocolate Lg

Visualizations Fields

product

Food Inventory

Product Lookup

- current_cost
- current_retail_price
- current_wholesale_price
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- promo_yn
- tax_exempt_yn
- unit_of_measure

Sales by Store

VALUES Example

Add drill-through fields here

Perform the below actions.

- 1) Add a matrix to the report
- 2) Add the product field from the Product Lookup table to the rows

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Count of Product ID Data category Uncategorized

Home table Measure Table \$ - % , . 0

New Quick measure measure Calculations

Structure Formatting Properties Calculations

Visualizations Fields

product

Baked Brand

Coffee Coffee

Drink Flavour

Loose Packag Tea

Total

1 Count of Product ID (DISTINCT) =
2 COUNTROWS(
3 DISTINCT(
4 'Product Lookup'[product_id]
5)
6)

Chocolate Croissant
Chocolate syrup
Civet Cat
Columbian Medium Roast
Columbian Medium Roast Lg
Columbian Medium Roast Sm
Cranberry Scone
Croissant
Dark chocolate
Dark chocolate Lg

1) Objective 2 to use counting functions plus DISTINCT & VALUES to create a view that shows the blank row. Create the Count of Product ID (DISTINCT) measure.

Maven Routers - Analysis - Power BI Desktop

Sign in

Home

Cut Copy Format painter

Out data - Excel Power BI datasets Server Enter data Recent sources - Transform Refresh data - New visual Text box More visuals - New measure measure Publish

Queries

Calculations Share

product_category Customer Sales

product_category	Customer Sales
Bakery	\$451,841.69
Branded	\$83,754.00
Coffee	\$1,651,861.15
Coffee beans	\$240,414.95
Drinking Chocolate	\$439,942.50
Flavours	\$51,060.00
Loose Tea	\$68,451.90
Packaged Chocolate	\$25,534.16
Tea	\$1,190,314.90
Total	\$4,252,704.88

product Count of Product ID (DISTINCT)

product	Count of Product ID (DISTINCT)
Almond Croissant	1
Brazilian - Organic	1
Brazilian Lg	1
Brazilian Rg	1
Brazilian Sm	1
Cappuccino	1
Cappuccino Lg	1
Carmel syrup	1
Chili Mayan	1
Chocolate Chip Biscotti	1
Chocolate Croissant	1
Chocolate syrup	1
Civet Cat	1
Columbian Medium Roast	1
Columbian Medium Roast Lg	1
Columbian Medium Roast Rg	1
Columbian Medium Roast Sm	1
Cranberry Scone	1
Croissant	1
Total	86

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT) **Selected**
- Customer Sales (Last Year)
- Customer Sales LY (COLUMNS)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)

Rows

product

Columns

Add data fields here

Values

Count of Product ID (DISTINCT)

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans —
- Store 3 Sales of whole Beans —
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Profit
- Tip & Best Practices
- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- DateID
- GNUMBER
- MonthID
- Month_Name
- Quarter & Year
- QuarterID
- Quarter_Name
- Transaction Date
- Week_Desc
- Week_ID

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions +

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Count of Product ID
Home table Measure Table \$ - % , . 0 Data category Uncategorized New measure Quick measure measure

Structure Formatting Properties Calculations

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans ...
- Store 3 Sales of whole Beans ...
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc

1 Count of Product ID (VALUES) =
2 COUNTROWS(
3 VALUES(
4 'Product Lookup'[product_id]
5)
6)

Chocolate Chip Biscotti 1
Chocolate Croissant 1
Chocolate syrup 1
Civet Cat 1
Columbian Medium Roast 1
Columbian Medium Roast Lg 1
Columbian Medium Roast Rg 1
Columbian Medium Roast Sm 1
Cranberry Scone 1
Croissant 1
Total 86

1) Create the Count of Product ID (VALUES) measure.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions +

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Get data - Excel Power BI datasets - SQL Server Enter data Recent sources Transform refresh data - New measure measure Quick Publish

Format painter

Customer Sales

product_category	Customer Sales
	\$49,949.63
Bakery	\$451,341.69
Branded	\$93,784.00
Coffee	\$1,651,661.15
Coffee beans	\$240,414.95
Drinking Chocolate	\$439,942.50
Flavours	\$51,060.00
Loose Tea	\$68,451.90
Packaged Chocolate	\$25,584.15
Tee	\$1,190,314.90
Total	\$4,232,704.88

product Count of Product ID (DISTINCT) Count of Product ID (VALUES)

product	Count of Product ID (DISTINCT)	Count of Product ID (VALUES)
Americano - Medium	1	1
Americano - Organic	1	1
Americano Lg	1	1
Americano Rg	1	1
Brazilian Sm	1	1
Cappuccino	1	1
Cappuccino Lg	1	1
Carmel syrup	1	1
Chili Mayan	1	1
Chocolate Chip Biscotti	1	1
Chocolate Croissant	1	1
Chocolate syrup	1	1
Civet Cat	1	1
Columbian Medium Roast	1	1
Columbian Medium Roast Lg	1	1
Columbian Medium Roast Rg	1	1
Columbian Medium Roast Sm	1	1
Cranberry Scone	1	1
Total	86	87

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (SEALAND)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans ..
- Store 3 Profit
- Store 3 Profit (KEEPPFILTERS)
- Store 3 Sales (KEEPPFILTERS)
- Store 8 Sales (KEEPPFILTERS)

Drill through

Cross-report

On Off

Keep all filters

Add drill-through fields here

Date Format (yyyy-mm-dd)

Date_ID

ISNUMBER

Month_ID

Month_Name

Quarter & Year

Quarter_ID

Quarter_Name

Transaction Date

Week_Day

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions +

Blank row shown via using the Count of Product ID (VALUES) measure in the matrix.

2) Add the Count of Product ID (VALUES) to the matrix i.e. values area. You will notice that the product has a extra row where the product entry is blank.

Maven Roasters - Analysis - Power BI Desktop

All products sold are contained within this table.

product_id

product

Count of Product ID (DISTINCT)

Count of Product ID (VALUES)

product_category Customer Sales

product_category	Customer Sales
Bakery	\$49,949.63
Branded	\$451,341.69
Coffee	\$33,784.00
Coffee beans	\$1,651,861.15
Drinking Chocolate	\$240,414.95
Flavours	\$439,942.50
Loose Tea	\$51,060.00
Packaged Chocolate	\$68,451.90
Tea	\$25,554.16
Total	\$4,252,704.88

Visualizations

Fields

product

Measure Table

- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)

Food Inventory

- product_id

Product Lookup

- current_cost
- current_retail_price
- current_wholesale_price
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- promo_yn
- tax_exempt_yn
- unit_of_measure

Sales by Store

- product_id

VALUES Example

- product_group

3) Perform the below actions.

- 1) Add a matrix to the report
- 2) Add the product id field from the Sales by Store table to the rows

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions

Key objective 2 is now complete. Use counting functions plus DISTINCT & VALUES to create a view that shows the blank row.

The screenshot shows a Power BI interface with three matrices and a Fields pane. The top matrix is titled 'Customer Sales' and has columns for 'product_category' and 'Customer Sales'. It shows a total value of '\$49,949.63'. The middle matrix is titled 'product' and has columns for 'product' and two measures: 'Count of Product ID (DISTINCT)' and 'Count of Product ID (VALUES)'. The bottom matrix is titled 'product_id' and has a single column for 'product_id' with a value of '72'. Blue arrows point from the highlighted cell in the 'Customer Sales' matrix to the 'product_id' matrix. The Fields pane on the right lists various fields under 'product', including 'product_id', 'product', 'product_category', etc.

When you click on this blank value here it shows that it is product id 72 on the matrix on the right hand side that is missing from the lookup table that's the missing product. Customer Sales \$49,949.63 value is also highlighted on the far left matrix.

File Home Transform Add Column View Tools Help

Close X New Recent Data source settings Manage Parameters Refresh Previous Manage Choose Column Remove Columns Keep Rows Remove Rows Split Column Group By Replace Values Data Type: Whole Number Use First Row as Headers Merge Queries Append Queries Combine Compose Filters Text Analytics AI Vision Azure Machine Learning

Queries (8) #1 product_id #2 product_group #3 product_category #4 product_type #5 product #6 product_description #7 unit_of_measure #8 current_cost

	#1 product_id	#2 product_group	#3 product_category	#4 product_type	#5 product	#6 product_description	#7 unit_of_measure	#8 current_cost
1	1	Whole Bean/Tess	Coffee beans	Organic Beans	Brazilian - Organic	It's like Carnival in a cup. Clean and smooth.	12 oz	
2	2	Whole Bean/Tess	Coffee beans	House Blend Beans	Our Old Time Diner Blend	Our packed blend of beans that is reminiscent of the cup of coffee you...	12 oz	
3	3	Whole Bean/Tess	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.	1 lb	
4	4	Whole Bean/Tess	Coffee beans	Espresso Beans	Primo Espresso Roast	Our premium single source of hand roasted beans.	1 lb	
5	5	Whole Bean/Tess	Coffee beans	Gourmet Beans	Columbian Medium Roast	A smooth cup of coffee any time of day.	1 lb	
6	6	Whole Bean/Tess	Coffee beans	Gourmet Beans	Ethiopia	From the home of coffee.	1 lb	
7	7	Whole Bean/Tess	Coffee beans	Premium Beans	Jamaican Coffee River	Yankee, it will start your day off right.	1 lb	
8	8	Whole Bean/Tess	Coffee beans	Premium Beans	Civet Cat	The most expensive coffee in the world, the cats do all the work.	5 lbs	
9	9	Whole Bean/Tess	Coffee beans	Organic Beans	Organic Decaf Blend	Our blend of hand picked organic beans that have been naturally deca...	1 lb	
10	10	Whole Bean/Tess	Coffee beans	Green beans	Guatemalan Sustainably Grown	Green beans you can roast yourself.	1 lb	
11	11	Whole Bean/Tess	Loose Tea	Herbal tea	Lemon Grass	You will think you are Thailand as you sip your cuppa.	.5 oz	
12	12	Whole Bean/Tess	Loose Tea	Herbal tea	Peppermint	Cool and refreshing to help calm your nerves.	.5 oz	
13	13	Whole Bean/Tess	Loose Tea	Black tea	English Breakfast	The traditional cup to start your day.	.5 oz	
14	14	Whole Bean/Tess	Loose Tea	Black tea	Karl Grey	A full leaf of Orange Pekoe blended with organic oil of bergamot.	.5 oz	
15	15	Whole Bean/Tess	Loose Tea	Green tea	Serenity Green Tea	Mountain grown and harvested at the optimal time.	1 oz	
16	16	Whole Bean/Tess	Loose Tea	Chai tea	Traditional Blend Chai	A traditional blend.	.5 oz	
17	17	Whole Bean/Tess	Loose Tea	Chai tea	Morning Sunrice Chai	Fair trade and organic and has a warm finish.	.5 oz	
18	18	Whole Bean/Tess	Loose Tea	Chai tea	Tulsi Eye Opener Chai	A spicier blend to awaken your taste buds.	.5 oz	
19	19	Whole Bean/Tess	Packaged Chocolate	Drinking Chocolate	Dark chocolate	This drinking chocolate is smooth and creamy.	1 lb	
20	20	Whole Bean/Tess	Packaged Chocolate	Organic Chocolate	Sustainably Grown Organic	Certified organic containing the highest quality ingredients.	1 lb	
21	21	Whole Bean/Tess	Packaged Chocolate	Drinking Chocolate	Chili Mayan	Fragrant with spices, this is the most flavourful drinking chocolate you...	1 lb	
22	22	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Sm	An honest cup a coffee.	8 oz	
23	23	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Mg	An honest cup a coffee.	16 oz	
24	24	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Lg	An honest cup a coffee.	24 oz	
25	25	Beverages	Coffee	Organic brewed coffee	Brazilian Sm	It's like Carnival in a cup. Clean and smooth.	8 oz	
26	26	Beverages	Coffee	Organic brewed coffee	Brazilian Mg	It's like Carnival in a cup. Clean and smooth.	16 oz	
27	27	Beverages	Coffee	Organic brewed coffee	Brazilian Lg	It's like Carnival in a cup. Clean and smooth.	24 oz	
28	28	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Sm	A smooth cup of coffee any time of day.	8 oz	
29	29	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Mg	A smooth cup of coffee any time of day.	16 oz	
30	30	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Lg	A smooth cup of coffee any time of day.	24 oz	
31	31	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Sm	A bold cup when you want that something extra.	8 oz	
32	32	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Mg	A bold cup when you want that something extra.	16 oz	
33	33	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Lg	A bold cup when you want that something extra.	24 oz	
34	34	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Sm	Still a front runner for good premium coffee.	8 oz	
35	35	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Mg	Still a front runner for good premium coffee.	16 oz	
36	36	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Lg	Still a front runner for good premium coffee.	24 oz	
37	37	Beverages	Coffee	Barista Espresso	Espresso shot	You will think you are in Venice when you sip this one.	1.5 oz	
38	38	Beverages	Coffee	Barista Espresso	Latte	You will think you are in Venice when you sip this one.	1.5 oz	
39	39	Beverages	Coffee	Barista Espresso	Latte Lg	You will think you are in Venice when you sip this one.	3.0 oz	

14 COLUMNS, 39 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 09/09

Query Settings

- PROPERTIES**
 - Name: Product Lookup
 - All Properties
- APPLIED STEPS**
 - Source
 - Promoted Headers
 - Changed Type

1) Objective 3 to update the data model to include the new product & use add a new visual to show the missing product id when the blank product is selected. From the query editor select the Product Lookup table.

2) Click on the source here.

3) Click on Browse to update the file path as we have an updated product lookup table.

Comma-Separated Values

Basic Advanced

File path

D:\Work\My Work\CV 2020 - Latest\Created Training Guides\Advanced DQ

Open file as

Csv Document

File origin

1252: Western European (Windows)

Line breaks

Apply all line breaks

Delimiter

Comma

OK Cancel

4) Select the Product Lookup (Updated) file & click on Open.

Screenshot of the Power Query Editor interface showing the 'Queries' list and a 'Product Lookup' query preview. A blue arrow points from the text instruction to the 'Open' dialog box, which is displayed over the main query preview area. The 'Open' dialog shows a file selection list with 'Product Lookup (Updated)' highlighted, and the 'Open' button is visible at the bottom right.

The 'Product Lookup' query preview shows data from a CSV file with columns: product_id, product_group, product_category, product_type, product, product_description, unit_of_measure, current_unit, and quantity_per_case. The preview displays rows 1 through 40, with the last few rows being Beverages.

Queries (8):

- Measure Table
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup**
- Sales by Store
- Store Lookup

Product Lookup (Preview):

product_id	product_group	product_category	product_type	product	product_description	unit_of_measure	current_unit
1	Whole Bean/Tea	Coffee beans	Organic Beans	Brazilian - Organic	It's like Carnival in a cup. Clean and smooth.	12 oz	3.6
2	Whole Bean/Tea	Coffee beans	Houseblend Beans	Our Old Time Diner Blend	Our packed blend of beans that is reminiscent of the days of coffee you...	12 oz	3.6
3	Whole Bean/Tea	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.	1 lb	2.95
4	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	4.5 oz	4.55
5	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
6	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
7	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
8	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
9	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
10	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
11	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
12	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
13	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
14	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
15	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
16	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
17	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
18	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
19	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
20	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
21	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
22	Whole Bean/Tea	Coffee beans	Espresso Beans	Italian Espresso Beans	Packed with aromatic, nutty, chocolatey notes of Italian coffee beans.	1 lb	2.95
23	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Lg	A smooth cup of coffee any time of day.	24 oz	0.75
24	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Sm	A bold cup when you want that something extra.	8 oz	0.55
25	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Ag	A bold cup when you want that something extra.	16 oz	0.75
26	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Lg	A bold cup when you want that something extra.	24 oz	0.875
27	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Sm	Still a front runner for good premium coffee.	8 oz	0.625
28	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Ag	Still a front runner for good premium coffee.	16 oz	0.775
29	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Lg	Still a front runner for good premium coffee.	24 oz	0.9375
30	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.75
31	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
32	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
33	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
34	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
35	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
36	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
37	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
38	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
39	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575
40	Beverages	Coffee	Espresso shot	Banista Espresso	You will think you are in Venice when you sip this one.	1.5 oz	0.8575

File name: Product Lookup (Updated)

Text File (*.txt; *.csv; *.xml)

Open Cancel

Query Settings:

- PROPERTIES
- Name: Product Lookup
- All Properties
- APPLIED STEPS
- Source
- Promoted Headers
- Changed Type

14 COLUMNS, 87 ROWS | Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 09:50

File Home Transform Add Column View Tools Help

Clear & Apply New Source Recent Data Enter Data Data source settings Manage Parameters Refresh Preview Message Properties Advanced Editor Choose Columns Remove Columns Keep Rows Remove Rows Split Column Group By Replace Values Data Type: Text Use First Row as Headers Merge Queries Append Queries Combine Filters Azure Machine Learning

Queries [8]

14 COLUMNS, 67 ROWS Column profiling based on top 1000 rows

5) Click on OK.

6) Click on Close & Apply.

Comma-Separated Values

Basic Advanced

File path: D:\Work\My Work\QY 2020 - Latest\Created Training Guides\Advanced Q4

Open file as: Csv Document

File origin: 1252: Western European (Windows)

Line breaks: Apply all line breaks

Delimiter: Comma

Query Settings

PROPERTIES

Name: Product Lookup
All Properties

APPLIED STEPS

Source Promoted Headers Changed Type

product_id	product_group	product_category	product_subcategory	product	product_description	unit_of_measurement	current_cost	
1	product_00	product_group	product_category	product_subcategory	product	product_description	unit_of_measurement	current_cost
2	1	Whole Bean/Tea	Coffee beans	Organic Beans	Brazilian - Organic	It's like Carnival in a cup. Clean and smooth.	12 oz	3.6
3	2	Whole Bean/Tea	Coffee beans	House blend Beans	Our Old Time Diner Blend	Our packed blend of beans that is reminiscent of the cup of coffee you...	12 oz	3.6
4	3	Whole Bean/Tea	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.	1 lb	2.99
5	4	Whole Bean/Tea	Coffee beans	Espresso Beans	Primo Espresso Roast	Our premium single source of hand roasted beans.	1 lb	4.09
6	5	Whole Bean/Tea	Coffee beans	Gourmet Beans	Columbian Medium Roast	A smooth cup of coffee any time of day.	1 lb	3
7	6	Whole Bean/Tea	Coffee beans	Gourmet Beans	Ethiopia	From the home of coffee.	1 lb	4.2
8	7	Whole Bean/Tea	Coffee beans				1 lb	3.99
9	8	Whole Bean/Tea	Coffee beans				1 lb	4.5
10	9	Whole Bean/Tea	Coffee beans				1 lb	3
11	10	Whole Bean/Tea	Coffee beans				9 oz	0.6
12	11	Whole Bean/Tea	Loose Tea				9 oz	0.6
13	12	Whole Bean/Tea	Loose Tea				9 oz	0.6
14	13	Whole Bean/Tea	Loose Tea				9 oz	0.6
15	14	Whole Bean/Tea	Loose Tea				9 oz	0.6
16	15	Whole Bean/Tea	Loose Tea				1 lb	0.9
17	16	Whole Bean/Tea	Loose Tea				9 oz	0.6
18	17	Whole Bean/Tea	Loose Tea				9 oz	0.6
19	18	Whole Bean/Tea	Loose Tea				9 oz	0.6
20	19	Whole Bean/Tea	Packaged Choc				1 lb	1.39
21	20	Whole Bean/Tea	Packaged Choc				1 lb	1.52
22	21	Whole Bean/Tea	Packaged Choc				9 oz	0.665
23	22	Beverages	Coffee				8 oz	0.5
24	23	Beverages	Coffee				16 oz	0.625
25	24	Beverages	Coffee				24 oz	0.79
26	25	Beverages	Coffee				8 oz	0.55
27	26	Beverages	Coffee	Organic brewed coffee	Brazilian Lg	It's like Carnival in a cup. Clean and smooth.	16 oz	0.75
28	27	Beverages	Coffee	Organic brewed coffee	Brazilian Lg	It's like Carnival in a cup. Clean and smooth.	24 oz	0.875
29	28	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Sm	A smooth cup of coffee any time of day.	8 oz	0.5
30	29	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Lg	A smooth cup of coffee any time of day.	16 oz	0.625
31	30	Beverages	Coffee	Gourmet brewed coffee	Columbian Medium Roast Lg	A smooth cup of coffee any time of day.	24 oz	0.75
32	31	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Sm	A bold cup when you want that something extra.	8 oz	0.55
33	32	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Lg	A bold cup when you want that something extra.	16 oz	0.75
34	33	Beverages	Coffee	Gourmet brewed coffee	Ethiopia Lg	A bold cup when you want that something extra.	24 oz	0.875
35	34	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Sm	Still a front runner for good premium coffee.	8 oz	0.6125
36	35	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Lg	Still a front runner for good premium coffee.	16 oz	0.775
37	36	Beverages	Coffee	Premium brewed coffee	Jamaican Coffee River Lg	Still a front runner for good premium coffee.	24 oz	0.9375
38	37	Beverages	Coffee	Barista Espresso	Espresso shot	You will think you are in Venice when you sip this one.	1.5 oz	0.75
39	38	Beverages	Coffee	Barista Espresso	Latte	You will think you are in Venice when you sip this one.	1.5 oz	0.875

PROVIEW DOWNLOADED AT 09:00

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Calculations Share

product_category Customer Sales

product

	Count of Product ID (DISTINCT)	Count of Product ID (VALUES)
Almond Croissant	1	1
Brazilian - Organic	1	1
Brazilian Ig	1	1
Brazilian Ig	1	1
Brazilian Sm	1	1
Cappuccino	1	1
Cappuccino Ig	1	1
Carmel Latte	1	1
Chai Mocha	1	1
Chocolate Chip Cookies	1	1
Chocolate Croissant	1	1
Chocolate Ig	1	1
Coffee Cat	1	1
Columbian Medium Roast	1	1
Columbian Medium Roast Ig	1	1
Columbian Medium Roast Ig	1	1
Columbian Medium Roast Sm	1	1
Cranberry Ig	1	1
Cream	1	1
Total	72	72

product_id

Key objective 3 is now complete. Update the data model to include the new product & use add a new visual to show the missing product id when the blank product is selected.

7) Once the updates are applied the 72 product id disappears from the product id from our Sales by Stores table because there is no missing values anymore both table now contain the same count of products once we add in that updated value.

Visualizations

Fields

Search

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFIELD)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LY (CALCSCD)
- Customer Sales LY (SBLANK)
- Orders by Females (Wrong)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans
- Store 3 Sales of whole Beans
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year
- Year_ID

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions

SELECTEDVALUE

SELECTEDVALUE()

Returns a value when there's **only one value** in a specified column, otherwise returns an (optional) alternate result

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

=**SELECTEDVALUE**(ColumnName, [AlternateResult])

The column you want to return a single value from

Examples:

- 'Customer Lookup'[customer_first_name]
- 'Product Lookup'[product]

(Optional) The value returned when there is either no value or more than one value in the specified column (if omitted, blank is returned)

Examples:

- “_”
- “NA”

product_group	Customer Sales	Quantity Sold (SELECTEDVALUE)	Retail Price (SELECTEDVALUE)
Add-ons	\$51,060.00		
Carmel syrup	\$12,366.40	15,458.00	\$0.80
Chocolate syrup	\$12,840.80	16,895.79	\$0.76
Hazelnut syrup	\$11,641.60	14,925.13	\$0.78
Sugar Free Vanilla syrup	\$14,211.20	17,330.73	\$0.82
Beverages	\$3,282,118.55		
Food	\$501,291.32		
Merchandise	\$83,784.00		
Whole Bean/Teas	\$334,451.01		
Total	\$4,252,704.88		

Because the expression doesn't evaluate to a **single** (scalar) result, nothing is returned. In this case, only when a single retail price can be determined is a result returned.

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

SELECTEDVALUE is another example of “syntax sugar” in DAX

- The DAX engine internally processes **SELECTEDVALUE** as a combination of **IF**, **HASONEVALUE**, and **VALUES**:

How it's written:

```
1 SELECTEDVALUE =  
2   SELECTEDVALUE(  
3     Customer[customer_first-name],  
4     "xx"  
5   )
```

How it's interpreted:

```
1 SELECTEDVALUE =  
2   IF(  
3     HASONEVALUE(  
4       Customer[customer_first-name]),  
5       VALUES(  
6         Customer[customer_first-name]),  
7         "xx"  
8   )
```

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Paste Cut Copy Format painter Get data from Excel Power BI datasets SQL Server Enter data from recent sources Transform Refresh data New visual Text box More visuals Insert New measure measure Quick Calculations Share

Visualizations Fields

product

Measure Table

- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)

Food Inventory

- product_id

Product Lookup

- Column1
- current_cost
- current_retail_price
- current_wholesale_price
- new_product_yn
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- promo_yn
- tax_exempt_yn
- unit_of_measure

Sales by Store

- product_id

VALUES Example

- product_group

product_group Customer Sales

product_group	Customer Sales
Add-ons	\$51,060.00
Beverages	\$3,282,118.55
Food	\$501,291.32
Merchandise	\$43,764.00
Whole Bean/Teas	\$334,451.01
Total	\$4,252,704.88

1) Perform the below actions.

- 1) Add a matrix to the report
- 2) Add the product group & product fields to the Rows
- 3) Add the Customer Sales to the values

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2

Maven Roaders - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Calculations Publish

Open Data Queries Import Calculations Share

Visualizations Fields

product

Measure Table

- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)

Food Inventory

- product_id

Product Lookup

- Column
- current_cost
- current_retail_price
- current_wholesale_price
- new_product_y/n
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- private_y/n
- tax_exempt_y/n
- unit_of_measure

Sales by Store

- product_id

VALUES Example

- product_group

product_group

Customer Sales

product_group	Customer Sales
Add-ins	\$51,060.00
Caramel syrup	\$12,366.40
Chocolate syrup	\$12,840.80
Hazelnut syrup	\$11,641.60
Sugar Free Vanilla syrup	\$14,211.20
Beverages	\$3,282,118.55
Brazilian Lg	\$91,003.50
Brazilian Rg	\$80,556.00
Brazilian Sm	\$58,676.20
Cappuccino	\$98,831.25
Cappuccino Lg	\$109,603.00
Columbian Medium Roast Lg	\$77,910.00
Columbian Medium Roast Rg	\$68,785.00
Columbian Medium Roast Sm	\$50,472.00
Dark chocolate Lg	\$126,588.50
Dark chocolate Rg	\$85,211.00
Total	\$4,252,704.88

2) Click here to collapse the product group & Customer Sales values.

3) Click here to drill back up.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Retail Price (SELECTEDVALUE)

\$ - % # .0 : Data category: Uncategorized

New Quick measure measure Calculations

1 Retail Price (SELECTEDVALUE) =
2 SELECTEDVALUE(
3 'Product Lookup'[current_retail_price]
4)

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL, Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPPFILTERS)
- Store 5 Sales (KEEPPFILTERS)
- Store 8 Sales (KEEPPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 +

2) Change the formatting accordingly over here.

1) Create the Retail Price (SELECTEDVALUE) measure

Category	Value	Current Retail Price
Food	\$501,291.32	
Merchandise	\$83,784.00	
Whole Bean/Tea	\$334,451.01	
Total	\$4,252,704.88	

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Documents Data Queries Calculations Share

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL.Modifier)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 3 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desk
- Week_ID
- Week_Year

product_group Customer Sales Retail Price (SELECTEDVALUE)

product_group	Customer Sales	Retail Price (SELECTEDVALUE)
Add-ons	\$51,060.00	
Beverages	\$1,382,110.55	
Food	\$301,291.32	
Merchandise	\$83,784.00	
Whole Bean/Tea	\$234,451.01	
Total	\$4,252,704.88	

3) What you will notice here is that there is no results as expected. What the SELECTEDVALUE function is doing is it's saying that based on the current filter context. Current filter context here is for Add-ons give me the current retail price for that. There isn't a current retail price at the product group level it can't evaluate to a unique value so it doesn't return anything

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 +

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Enter data Refresh Recent sources Transform New visual Text box More Quick measure measure Publish

Queries Data Calculations Share

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LV (COALESCE)
- Customer Sales LV (ISBLANK)
- Orders by Females (Wrong)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL, Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID
- Week_Year

product_group Customer Sales - Retail Price (SELECTEDVALUE)

product_group		
Add-ons	\$51,960.00	\$0.80
Caramel syrup	\$12,366.40	\$0.78
Chocolate syrup	\$12,840.80	\$0.78
Hazelnut syrup	\$11,641.60	\$0.78
Sugar Free Vanilla Syrup	\$14,211.20	\$0.82
Beverages	\$3,282,118.55	
Food	\$101,291.32	
Merchandise	\$89,544.00	
Whole Bean/Teas	\$334,451.91	
Total	\$4,252,704.88	

4) But when you drill down you can see values as they evaluate to a single retail price. All other product groups can be evaluated to a single retail price when drilling down i.e. Beverages, Food etc.

Maven Rooters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Quantity Sold (SEL... Format: Decimal number Data category: Uncategorized

Home table: Measure Table \$ - % . , : New Quick measure measure Calculations Properties

2) Change the formatting accordingly over here.

1 Quantity Sold (SELECTEDVALUE) =
2 DIVIDE(
3 [Customer Sales],
4 SELECTEDVALUE(
5 'Product Lookup'[current_retail_price]
6)
7)

1) Create the Quantity Sold (SELECTEDVALUE) measure.

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LV (OAL2SC)
- Customer Sales LV (SBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPMFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL.Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 +

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dashboard Data Queries Insert Calculations Share

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL Modifer)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPPFILTERS)
- Store 5 Sales (KEEPPFILTERS)
- Store 6 Sales (KEEPPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc
- Week_ID

3) Add the Quantity Sold (SELECTEDVALUE) measure to the matrix.

product_group Customer Sales Quantity Sold (SELECTEDVALUE) Retail Price (SELECTEDVALUE)

product_group	Customer Sales	Quantity Sold (SELECTEDVALUE)	Retail Price (SELECTEDVALUE)
Add-ons	\$51,060.00	15,458.00	\$3.38
Caramel syrup	\$12,366.40	16,895.79	\$0.75
Chocolate syrup	\$12,840.80	16,895.79	\$0.75
Hazelnut syrup	\$11,641.60	14,925.13	\$0.78
Sugar Free Vanilla syrup	\$14,211.20	17,330.73	\$0.82
Beverages	\$3,282,118.53		\$2.50
Brazilian Lg	\$91,003.50	26,001.00	\$3.50
Brazilian Pg	\$80,536.00	26,832.00	\$3.00
Brazilian Sm	\$58,676.20	26,671.00	\$2.20
Cappuccino	\$96,831.25	26,355.00	\$3.75
Cappuccino Lg	\$109,800.00	25,816.00	\$4.25
Columbian Medium Roast Lg	\$77,910.00	25,970.00	\$1.00
Columbian Medium Roast Pg	\$66,785.00	27,514.00	\$2.50
Columbian Medium Roast Sm	\$50,472.00	25,236.00	\$2.00
Dark chocolate Lg	\$126,596.50	28,131.00	\$4.50
Total	\$4,293,704.88		

What we were able to do with that function is we were able to return the quantities sold for the individual products. Able to divide the retail price based on the customer sales in the instance where a single value can be determined that's why we are still not seeing the quantities sold at retail price at the group level we are seeing those at the individual product level.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 +

ALLEXCEPT

ALLEXCEPT()

Removes all report context filters in the table **except** the filters applied to the specified columns in the query

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

=**ALLEXCEPT**(**TableName**, [**ColumnName**],[**ColumnName**], [...])

Name of an **existing table** (the use of table expressions is not allowed here)

Examples:

👍 'Sales by Store'

👍 'Returns'

👎 FILTER(

'Sales by Store',
'Sales by Store'[store_id] IN {2,5,7,9,10})

Additional column references within the **same referenced table** or a table that is on the **one-side of the relationship** (adding additional columns is optional)

Examples:

- 'Sales by Store'[product_group]
- 'Sales by Store'[product_category]
- 'Calendar'[Transaction_Date]



PRO TIP:

ALLEXCEPT is typically used as a **CALCULATE** modifier and not a stand-alone table function

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

```
1 ALLEXCEPT Demo =  
2 CALCULATE(  
3     [Customer Sales],  
4     ALLEXCEPT(  
5         'Sales by Store',  
6         'Calendar'[Transaction_Date], 'Store Lookup'[sales_outlet_id]  
7     )  
8 )
```



store_id	Customer Sales	Percent of All Selected Sales	ALLEXCEPT Demo
3	\$295,632.11	33.30%	\$295,632.11
Add-ons	\$2,142.40	0.24%	\$295,632.11
Beverages	\$234,583.35	26.42%	\$295,632.11
Food	\$33,571.50	3.78%	\$295,632.11
Bakery	\$33,571.50	3.78%	\$295,632.11
Merchandise	\$6,727.00	0.76%	\$295,632.11
Whole Bean/Teas	\$18,607.86	2.10%	\$295,632.11
5	\$293,310.74	33.04%	\$293,310.74
8	\$298,812.44	33.66%	\$298,812.44
Total	\$887,755.29	100.00%	\$887,755.29

Here we're using **ALLEXCEPT** as a **CALCULATE** modifier to remove all initial filter context from the '**Sales by Store**' table, except for filters on the following columns:

- 'Calendar'[Transaction_Date]
- 'Store Lookup'[sales_outlet_id]

The measure above returns Customer Sales by **store ID** and **Date**, but ignores filter context created by other fields (i.e. Product Category)

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Customer Sales (AL...
Home table: Measure Table
Format: Currency
Data category: Uncategorized

Structure Formating Properties Calculations

2) Change the formatting accordingly over here.

1 Customer Sales (ALLEXCEPT) =
2 CALCULATE(
3 [Customer Sales],
4 ALLEXCEPT(
5 'Sales by Store',
6 'Product Lookup'[product_group],
7 'Calendar'[Transaction_Date]
8)
9)

Visualizations Fields

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT) **(Selected)**
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL Modifier)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Duplicate of Table & Filter Functions - 3

Maven Reporters - Analysis - Power BI Desktop

Sign In

File Home Insert Modeling View Help

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Power Copy Format painter Datasheet Data Queries Insert Calculations Share

Visualizations Fields

Search

Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVED/FILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT)
- Customer Sales (Last Year)
- Customer Sales LY (CALCULATE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL Modifer)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)
- Tip & Best Practices
- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc

product_group Customer Sales Customer Sales (ALLEXCEPT)

product_group	Customer Sales	Customer Sales (ALLEXCEPT)
Add-ons	\$51,060.00	\$51,060.00
Caramel syrup	\$12,386.40	\$51,060.00
Chocolate syrup	\$12,640.00	\$51,060.00
Hazelnut syrup	\$11,641.60	\$51,060.00
Sugar Free Vanilla syrup	\$14,211.20	\$51,060.00
Beverages	\$3,282,118.55	\$3,282,118.55
Brazilian Lg	\$91,003.50	\$3,282,118.55
Brazilian Pg	\$88,556.00	\$3,282,118.55
Brazilian Sm	\$58,676.20	\$3,282,118.55
Cappuccino	\$90,831.25	\$3,282,118.55
Cappuccino Lg	\$109,803.40	\$3,282,118.55
Columbian Medium Roast Lg	\$77,100.00	\$3,282,118.55
Columbian Medium Roast Pg	\$52,785.00	\$3,282,118.55
Columbian Medium Roast Sm	\$50,472.00	\$3,282,118.55
Dark chocolate Lg	\$126,586.50	\$3,282,118.55
Dark chocolate Pg	\$88,211.00	\$3,282,118.55
Total	\$4,252,704.88	\$4,252,704.88

3) Add the Customer Sales (ALLEXCEPT) measure to the matrix. You will notice that each of the product groups have the same repeating totals.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dashboard Data Queries Insert Calculations Share

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVED FILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT)
- Customer Sales (Last Year)
- Customer Sales LY (OALCESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPPFILTERS)
- Store 5 Sales (KEEPPFILTERS)
- Store 6 Sales (KEEPPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc

Transaction Date

01/01/2017 23/10/2018

product_group Customer Sales Customer Sales (ALLEXCEPT)

product_group	Customer Sales	Customer Sales (ALLEXCEPT)
Add-ons	\$38,258.40	\$38,258.40
Caramel syrup	\$9,303.20	\$38,258.40
Chocolate syrup	\$9,632.00	\$38,258.40
Hazelnut syrup	\$8,644.00	\$38,258.40
Sugar Free Vanilla Syrup	\$10,679.20	\$38,258.40
Beverages	\$2,447,228.65	\$2,447,228.65
Brazilian Lg	\$67,865.00	\$2,447,228.65
Brazilian M	\$58,796.00	\$2,447,228.65
Brazilian Sm	\$45,637.00	\$2,447,228.65
Cappuccino	\$73,387.50	\$2,447,228.65
Cappuccino Lg	\$82,156.75	\$2,447,228.65
Columbian Medium Roast Lg	\$57,996.00	\$2,447,228.65
Columbian Medium Roast M	\$51,110.00	\$2,447,228.65
Columbian Medium Roast Sm	\$37,602.00	\$2,447,228.65
Dark chocolate Lg	\$94,504.50	\$2,447,228.65
Dark chocolate M	\$65,318.50	\$2,447,228.65
Total	\$3,172,762.00	\$3,172,762.00

4) Add transaction date to a slicer. You notice both columns in the matrix have the exact same values in each row when the data range are adjusted.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Duplicate of Table & Filter Functions - 3

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More

New measure measure Quick

Transform Refresh data New visual Text box More

Share

Visualizations Fields

Search

Measure Table

- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL, Modifier)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Top & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date
- Week_Desc

Transaction Date: 01/01/2017 - 21/06/2018

product_group Customer Sales Customer Sales (ALLEXCEPT)

product_group	Customer Sales	Customer Sales (ALLEXCEPT)
Beverages	\$294,833.25	\$2,203,141.05
Dark chocolate Lg	\$85,196.00	\$2,203,141.05
Dark chocolate lg	\$56,634.00	\$2,203,141.05
Happy Holidays hot chocolate		\$2,203,141.05
Snow Day Hot Chocolate		\$2,203,141.05
Sustainably Grown Organic Lg	\$86,536.75	\$2,203,141.05
Sustainably Grown Organic lg	\$66,180.00	\$2,203,141.05
Total	\$294,833.25	\$2,858,271.70

product_category

- Bakery
- Branded
- Coffee
- Coffee beans
- Drinking Chocolate
- Flavours
- Loose Tea
- Packaged Chocolate
- Tea

5) Add product category to a slicer. The Customer Sales values will adjust based on values selected from the product category slicer however the Customer Sales (ALLEXCEPT) values in the matrix will still remain the same as all filters are being cleared except for product group & transaction date the column will not update on any additional filter selection.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Duplicate of Table & Filter Functions - 3

ASSIGNMENT: ALLEXCEPT

KEY OBJECTIVES

- 1) create a matrix with the store id, product group & customer name on rows. Including filters for 2018 for the top 10 customers excluding non members & store 8.
- 2) Use **ALLEXCEPT** to create a measure that only accepts filters for the date, store ID product group & customer.
- 3) Create a **% of store-level sales** measure to calculate the percent of sales by the top 10 customers at the store level.

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI SQL Enter data Recent sources - Transform Refresh data - New visual Text box More visual - New measure measure Quick Publish

Queries Calculations Share

Transaction Date 01/01/2018 31/12/2018

store_id

- 8
 - Add-ons
 - Allen
 - Buffy
 - Chanda
 - Dean
 - Ferdinand
 - Ganeth
 - Raven
 - Summer
- Beverages
 - Allen
 - Buffy
 - Chanda
 - Dean
 - Ferdinand
 - Ganeth

Filters

Search

Customer Sales in (All)

customer_first-name top 10 by Customer Sales

Filter type: Top N

Show items: Top 10

By value: Customer Sales

Apply filter

customer_first-name is not Non-Member

Filter type: Basic filtering

Select all

- Aaron
- Abbot
- Abbot Charles
- Abel Hood
- Abra
- Alva Dyrson

Require single selection

product_group in (All)

store_id

Visualizations Fields

Measure Table

Calendar

Customer Lookup

Employee Lookup

Food Inventory

Product Lookup

Sales by Store

Store Lookup

VALUES Example

Rows

Columns

Add data fields here

Values

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

1) Objective 1 to create a matrix with the store id, product group & customer name on rows. Including filters for 2018 for the top 10 customers excluding non members & store 8.

2) Open up the filter pane filter for 2018 for the top 10 customers excluding non members & store 8.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI SQL Enter data Recent sources - Transform Refresh data - New visual Text box More... New Publish Measure measure Calculations Share

transaction_date
01/01/2018 31/12/2018

store_id
Add-ons Allen Buffy Chanda Dean Ferdinand Gareth Raven Summer Beverages Allen Buffy Chanda Dean Ferdinand Gareth

Filters Visualizations Fields

Search

Customer Sales in (All)

customer_first-name Top 10 by Customer S... Filter type Top N Show items Top 10 By value Customer Sales Apply filter

customer_first-name is not Non-Member Filter type Basic filtering Search Select all Aaron 1 Abbot 1 Abbot Charles 1 Abel Hood 1 Abra 1 Alva Dyrsson 1 Require single selection

product_group in (All)

store_id

Measure Table Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup VALUES Example

Rows Add data fields here

Columns Add data fields here

Values Add data fields here

Drill through

Cross-report Off Keep all filters On

Add drill-through fields here

4) Transaction date set parameters to 01/01/2018 - 31/12/2018.

3) Expand customer sales here & filter against a top N result here i.e. top 10 click apply filter to set it.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI SQL Enter data Recent sources Transform Refresh data New visual Text box More... Publish

Queries Data Calculations Share

Transaction Date 01/01/2018 31/12/2018

store_id

Add-ons Allen Buffy Chanda Dean Ferdinand Gareth Raven Summer Beverages Allen Buffy Chanda Dean Ferdinand Gareth

5) Filter for store 8 using a slicer based on the store id field. Change the selection settings to a drop down & choose 8.

6) Last thing to do is to exclude non members. Add customer first name to the filters on visual select all & filter de-select non members.

Filters Visualizations Fields

Search

Customer Sales in (All)

customer_first-name top 10 by Customer Sales

Filter type Top N

Show items Top 10

By value Customer Sales

Apply filter

customer_first-name is not Non-Member

Filter type Basic filtering

Search

Select all Aaron 1 Abbott 1 Abbott Charles 1 Abel Hood 1 Abra 1 Alva Dyrson 1

Require single selection

product_group in (All)

store_id in (All)

Measure Table Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store Store Lookup VALUES Example

Rows

Columns Add data fields here

Values Add data fields here

Drill through

Cross-report Off Keep all filters On

Add drill-through fields here

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

Maven Reactors - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Customer Sales (All...) Format: Currency Data category: Uncategorized

Home table: Measure Table \$ - % 9.0 2

Measure tools: New Quick measure measure

Structure, Formatting, Properties, Calculations

Visualizations > Fields >

Search: Measure Table

Row:

- store_id
- product_group
- customer_first_name

Columns: Add data fields here

Values: Add data fields here

Scalar Measures:

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Te...
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Drill through:

Cross-report: Off

Keep all filters: On

Add drill-through fields here

Tip & Best Practices

Total Profit

Calendar:

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name
- Transaction_Date

1 Customer Sales (ALLEXCEPT Assignment) =
2 CALCULATE(
3 [Customer Sales],
4 ALLEXCEPT(
5 'Sales by Store',
6 'Calendar'[Transaction_Date],
7 'Store Lookup'[store_id],
8 'Customer Lookup'[customer_first-name],
9 'Product Lookup'[product_group]
10)
11)

7) Objective 2 to use ALLEXCEPT to create a measure that only accepts filters for the date, store ID product group & customer.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 +

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Calculations Publish

Dataset Data Queries Insert Calculations Share

Transaction Date
01/01/2018 01/15/2018

store_id Customer Sales (ALLEXCEPT Assignment)

store_id	Customer Sales (ALLEXCEPT Assignment)
Add-ons	\$17,950.59
Allen	\$219.20
Buffy	\$13.60
Chanda	\$12.00
Dean	\$24.00
Ferdinand	\$29.60
Gareth	\$24.00
Raven	\$28.80
Summer	\$38.40
Beverages	\$12,351.55
Allen	\$1,421.90
Buffy	\$1,448.00
Chanda	\$1,280.95
Dean	\$1,260.40
Total	\$17,950.59

store_id

Filters

Search

Filters on this visual
Customer Sales is (All)

customer_first-name
Top 10 by Customer S...
Filter type Top N
Show items Top 10
By value Customer Sales
Apply filter

customer_first-name is not Non-Member
Filter type Basic filtering
Select all
Aaron 1
Abbott 1
Abbot Charles 1
Abel Hood 1
Abra 1
Alice Dawson 1
Require single selection

product_group_id (All)

More_id

Visualizations

Search

Measure Table
% Quantity Sold to Females
% Store Sales (REMOVEFILTERS)
Cost (CURRENCY)
Count of Product ID (DISTINCT)
Count of Product ID (VALUES)
Customer Sales (ALLEXCEPT Assignment)
Customer Sales (ALLEXCEPT)
Customer Sales (Last Year)
Customer Sales LY (COALESCE)
Customer Sales LT (ISBLANK)
Orders by Females (Wrong)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Scalar Measures
Store 3 Sales (KEEPFILTERS)
Store 3 Sales of whole Beans & Teas (ALL MODELS)
Store 3 Sales of whole Beans & Teas (CALCULATE)
Store 5 Profit
Store 5 Profit (KEEPFILTERS)
Store 5 Sales (KEEPFILTERS)
Store 8 Sales (KEEPFILTERS)
Tip & Best Practices
Total Profit

Fields

Search

Measure Table
% Quantity Sold to Females
% Store Sales (REMOVEFILTERS)
Cost (CURRENCY)
Count of Product ID (DISTINCT)
Count of Product ID (VALUES)
Customer Sales (ALLEXCEPT Assignment)
Customer Sales (ALLEXCEPT)
Customer Sales (Last Year)
Customer Sales LY (COALESCE)
Customer Sales LT (ISBLANK)
Orders by Females (Wrong)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Scalar Measures
Store 3 Sales (KEEPFILTERS)
Store 3 Sales of whole Beans & Teas (ALL MODELS)
Store 3 Sales of whole Beans & Teas (CALCULATE)
Store 5 Profit
Store 5 Profit (KEEPFILTERS)
Store 5 Sales (KEEPFILTERS)
Store 8 Sales (KEEPFILTERS)
Tip & Best Practices
Total Profit

Calendar
Date_ID
Month_ID
Month_Name
Quarter_ID
Quarter_Name
Transaction Date

8) Add the Customer Sales (ALLEXCEPT Assignment) measure to the matrix expected results are provided.

Key objective 2 is now complete. Use ALLEXCEPT to create a measure that only accepts filters for the date, store ID product group & customer.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name % of Store-Level Sa... % Format Percentage Data category Uncategorized

Home table Measure Table \$ - % 9.0 2

Measure Formatting Properties Calculations

1 % of Store-Level Sales =
2 VAR StoreLevelSales =
3 CALCULATE(
4 [Customer Sales],
5 REMOVEFILTERS(
6 'Customer Lookup'
7),
8 KEEPFILTERS(
9 'Calendar'
10)
11)
12 VAR Ratio =
13 DIVIDE(
14 [Customer Sales (ALLEXCEPT Assignment)],
15 StoreLevelSales
16)
17 RETURN
18 Ratio

Visualizations Fields

Search

Measure Table

- % of Store-Level Sales
- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (Last Year)
- Customer Sales (YTD (CALCULATE))
- Customer Sales (YTD (ISBLANK))
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL MODELS)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

Page 7 of 7

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dashboard Data Queries Insert Calculations Share

Transaction Date
01/01/2018 01/15/2018

store_id Customer Sales (% of Store-Level Sales)

	Customer Sales	% of Store-Level Sales
All	\$17,950.59	2.81%
Add-ons	\$219.20	0.00%
Allen	\$13.60	0.04%
Buffy	\$12.00	0.06%
Chanda	\$28.80	0.38%
Dean	\$24.00	0.32%
Ferdinand	\$29.60	0.39%
Gareth	\$24.00	0.32%
Raven	\$28.80	0.38%
Summer	\$28.80	0.31%
Beverages	\$12,351.9	2.51%
Allen	\$14.40	0.29%
Buffy	\$14.40	0.30%
Chanda	\$28.80	0.28%
Dean	\$28.80	0.28%
Total	\$17,950.59	2.81%

store_id

customer_first_name

customer_last_name

product_group

More

Filters

Visualizations

Fields

Search

Customer Sales (All)

customer_first_name

Top 10 by Customer Sales

Filter type

Top N

Show items

Top 10

By value

Customer Sales

Apply filter

customer_first_name

is not Non-Member

Filter type

Basic filtering

Search

Select all

Aaron

Abbott

Abbot Charles

Abel Hood

Abra

Abus Duran

Require single selection

product_group

id (All)

More

Scalar Measures

Store 3 Sales (KEEPMFILTERS)

Store 3 Sales of whole Beans & Teas (ALL MODELS)

Store 3 Sales of whole Beans & Teas (CALCULATE)

Store 5 Profit

Store 5 Profit (KEEPMFILTERS)

Store 5 Sales (KEEPMFILTERS)

Store 8 Sales (KEEPMFILTERS)

Tip & Best Practices

Total Profit

Calendar

Date_ID

Month_ID

Month_Name

Quarter_ID

Quarter_Name

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

Page 7 of 7

Maven Roadmap - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name % of Store-Level Sales
Home table Measure Table \$ - % 9.90

Format Whole number Data category Uncategorized

New Quick measure measure

Source Formatting Properties Calculators

1 % of Store-Level Sales =
2 VAR StoreLevelSales =
3 CALCULATE(
4 [Customer Sales],
5 REMOVEFILTERS(
6 'Customer Lookup'
7),
8 KEEPFILTERS(
9 'Calendar'
10)
11)
12 VAR Ratio =
13 DIVIDE(
14 [Customer Sales (ALLEXCEPT Assignment)],
15 StoreLevelSales
16)
17 RETURN
18 StoreLevelSales

Visualizations Fields

Measure Table

- % of Store-Level Sales
- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (Last Year)
- Customer Sales UV (COALESCE)
- Customer Sales UV (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Rows

- store_id
- product_group
- customer_firstname

Columns

- Add data fields here

Values

- Customer Sales (ALLEXCEPT)
- % of Store-Level Sales

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL MODELS)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPPATTERNS)

Drill through

- Cross-report
- Off
- Keep all filters
- On

Add drill-through fields here

product_group is (All)

store_id

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

11) Replace Ratio with the StoreLevelSales Variable so that we can validate the % of Store-Level Sales output.

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

New measure measure

Transform Refresh data New visual Text box More visuals Quick measure measure

Share

Transaction Date
01/01/2018 - 01/12/2018

store_id

Customer Sales (ALLEXCEPT Assignment)

Customer First Name	Customer Last Name	Customer Sales (ALLEXCEPT Assignment)	% of Store-Level Sales
Allen	Wong	\$17,950.59	638,014
Buffy	Yost	\$11,60	7,598
Chanda	Yost	\$12,00	7,598
Dean	Yost	\$12,80	7,598
Ferdinand	Yost	\$12,40	7,598
Gareth	Yost	\$12,40	7,598
Hanna	Yost	\$12,40	7,598
Josephine	Yost	\$12,40	7,598
Raven	Yost	\$12,40	7,598
Summer	Yost	\$12,40	7,598
Total		\$17,950.59	638,014

Filters

Visualizations

Fields

Measure Table

- Customer Sales (ALLEXCEPT Assignment)
- % of Store-Level Sales
- % Quantity Sold to Females
- % Store Sales (REMOVED FILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT)
- Customer Sales (Last Year)
- Customer Sales (LY) (CALCULATE)
- Customer Sales (LY) (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTIONVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL, Model)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPPFILTERS)
- Store 5 Sales (KEEPPFILTERS)
- Store 8 Sales (KEEPPFILTERS)
- Tip & Best Practices
- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID
- Quarter_Name

12) If you look \$638,014 in sales & if you were to take \$17,950.59 divide it by \$638,014 you would get 2.81% this is a good example of how you use variables to test the outputs of few different measures.

Key objective 3 is now complete. Create a % of store-level sales measure to calculate the percent of sales by the top 10 customers at the store level.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4

ALLSELECTED

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

ALLSELECTED()

Returns all rows in a table or values in a column, ignoring filters specified in the query but keeping any other existing filter context

=ALLSELECTED(**TableNameOrColumnName**, [**ColumnName**],[**ColumnName**], [...])

Name of a table or column that you want to remove filters from (**NOTE:** This input is required unless being used as a CALCULATE modifier)

Examples:

- ALLSELECTED
- 'Sales by Store'
- 'Sales by Store'[product_group]

Additional column references within the **same** referenced table (adding additional columns is optional)

Examples:

- 'Sales by Store'[product_group]
- 'Sales by Store'[product_category]



HEY THIS IS IMPORTANT!

ALLSELECTED respects existing filter context **except** row and column filters within a visual. This function can be used to obtain visual totals or subtotals in queries.

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

```
1 Selected Sales (ALLSELECTED) =  
2 CALCULATE(  
3     [Customer Sales],  
4     ALLSELECTED(  
5     )  
6 )
```

This measure calculates sales
(quantity * price), in a modified filter
context based on external filters

product_group	Customer Sales	All Sales	Percent of All Sales	Selected Sales (ALLSELECTED)	Percent of All Selected Sales
Beverages	\$3,282,118.55	\$4,252,704.88	77.18%	\$3,867,193.87	84.87%
Food	\$501,291.32	\$4,252,704.88	11.79%	\$3,867,193.87	12.96%
Merchandise	\$83,784.00	\$4,252,704.88	1.97%	\$3,867,193.87	2.17%
Total	\$3,867,193.87	\$4,252,704.88	90.93%	\$3,867,193.87	100.00%

When we select specific product groups in a report slicer (**Beverages**, **Food** & **Merchandise**) ALLSELECTED respects the filter context from the selection (slicer) but not from the rows and columns of the visual.

This function is commonly used to obtain visual totals or subtotals in queries

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name Customer Sales (ALLSELECTED) Format Currency Data category Uncategorized New Quick measure measure

Home table Measure Table \$ - % 9.9 Properties Calculations

Customer Sales (ALLSELECTED) =
CALCULATE([Customer Sales], ALLSELECTED())

Visualizations Fields

CUSTOMER

Measure Table

- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)**
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (SBLANK)

Scalar Measures

- Total Customers

Tip & Best Practices

- Customer Sales**

Customer Lookup

- birth_year
- birthdate
- customer_email
- customer_first_name
- customer_id
- customer_since
- gender
- home_store
- loyalty_card_number

Sales by Store

- customer_id

1) Create the Customer Sales (ALLSELECTED) measure.

Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED +

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED +

Page 6 of 8

Maven Reporters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Queries Data Calculations Share

Transaction Date

01/01/2017 30/04/2019

product_group Customer Sales Customer Sales (ALLSELECTED)

product_group	Customer Sales	Customer Sales (ALLSELECTED)
Add-ons	\$51,060.00	\$4,252,704.88
Beverages	\$3,282,118.55	\$4,252,704.88
Food	\$501,291.32	\$4,252,704.88
Merchandise	\$83,784.00	\$4,252,704.88
Whole Bean/Tees	\$334,451.01	\$4,252,704.88
Total	\$4,252,704.88	\$4,252,704.88

Visualizations Fields

Measure Table

- % of Store-Level Sales
- % Quantity Sold to Females
- % Store Sales (REMOVED FILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 2 Sales (KEEPFILTERS)
- Store 2 Sales of whole Bean & Tees (ALL Mod...)
- Store 2 Sales of whole Bean & Tees (CALCULAT...
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

- Total Profit

Calendar

- Date_ID
- Month_ID
- Month_Name
- Quarter_ID

2) Add the Customer Sales (ALLSELECTED) measure to the matrix expected results are provided.

You can see here that we have the same repeating total as there's no other context right there's nothing else that the ALLSELECTED measure can look to. To change the filter context right because we know that it invalids or doesn't pay attention to the row or any sort of column filter context.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED +

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Queries

Calculated

Share

Transaction Date

01/01/2017 19/04/2019

product_group Customer Sales Customer Sales (ALLSELECTED)

product_group	Customer Sales	Customer Sales (ALLSELECTED)
Beverages	\$3,282,119.87	\$3,867,193.87
Food	\$501,291.32	\$3,867,193.87
Merchandise	\$83,784.00	\$3,867,193.87
Total	\$3,867,193.87	\$3,867,193.87

Visualizations Fields

product_group

Product Lookup

product_group

VALUES Example

product_group

Field

product_group

Drill through

Cross-report

On

Keep all filters

On

Add drill-through fields here

3) Add a slicer with the product group field from the Product Lookup table.

4) When you make a selection with Beverages, Food & Merchandise you can see that the total is updating based on that selection. This is the portion of what the ALLSELECTED means where it ignores any of the filters from the initial context of the visual the row filter or the column filter & it takes into account external filter contexts so the slicer is updating the filter context when making new selections additionally the date slicer will influence this Customer Sales (ALLSELECTED) total.

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED +

Page 6 of 8

ASSIGNMENT: ALLSELECTED

KEY OBJECTIVES

- 1) Using the Food Inventory table, create two measures for **Total Baked** (quantity start of day) & **Total Sold** (quantity sold).
- 2) Create measures for **% of Total Baked** & **% of Total Sold** that total 100% based on a slicer selection of product type.
- 3) Build a visual to compare the **% of All Baked** to **% of Total Baked**. Which food item is baked the most for store 8?

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Total Baked Data category: Uncategorized

Home table: Measure Table \$ - % # . 0

Measure Formatting Properties Calculations

2) Change the formatting accordingly over here.

1 Total Baked =
2 SUMX(
3 'Food Inventory',
4 'Food Inventory'[quantity_start_of_day]
5)

Beverages
Food
Merchandise
Whole Bean/Teas

1) Objective 1 to use the Food Inventory table, create two measures for Total Baked (quantity start of day) & Total Sold (quantity sold)

Visualizations Fields

Search

Measure Table

- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (DAILYSCD)
- Customer Sales LY (SBLANQ)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 1 Sales (KEEPFILTERS)
- Store 1 Sales of whole Beans & Teas (ALL...)
- Store 1 Sales of whole Beans & Teas (CAL...
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 6 Sales (KEEPFILTERS)
- Tip & Best Practices
- Total Baked
- Total Profit
- Total Sold

Calendar

Date_ID

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED +

Page 8 of 8

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: Total Sold Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % , . 0

Measure Formatting Properties Calculations

2) Change the formatting accordingly over here.

1 Total Sold =
2 SUMX(
3 'Food Inventory',
4 'Food Inventory'[quantity_sold]
5)

Beverages
Food
Merchandise
Whole Bean/Teas

1) Objective 1 part two to use the Food Inventory table, create measure for Total Sold (quantity sold).

Key objective 1 is now complete. Using the Food Inventory table, create two measures for Total Baked (quantity start of day) & Total Sold (quantity sold).

Visualizations Fields

Search

Measure Table

- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (DALSCE)
- Customer Sales LY (SBLANO)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL...)
- Store 3 Sales of whole Beans & Teas (CAL...
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)
- Store 6 Sales (KEEPFILTERS)
- Tip & Best Practices
- Total Baked
- Total Profit
- Total Sold

Calendar

Data_ID

Page 1 Advanced CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED +

Page 6 of 6



File Home Insert Modeling View Help Table tools Measure tools

Name: % of Total Baked (A...) Type: % (0.00%, 0.00%) Data category: Undesignated New Quick Measure measure Calculations

Home table: Measure Table \$ - % , , 2 Properties Calculations

Measure Formatting Properties Calculations

1 % of Total Baked (ALLSELECTED) =
2 VAR SelectedQuantityBaked =
3 CALCULATE(
4 [Total Baked],
5 ALLSELECTED(
6)
7)
8 VAR Ratio =
9 DIVIDE(
10 [Total Baked],
11 SelectedQuantityBaked,
12 "0"
13)
14 RETURN
15 Ratio

Visualizations Fields

Search

Measure Table

- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (COALESC)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 1 Sales (KEEPFILTERS)
- Store 1 Sales of whole Beers & Teas (ALL...)
- Store 1 Sales of whole Beers & Teas (CAL...
- Store 1 Profit
- Store 1 Profit (KEEPFILTERS)
- Store 1 Sales (KEEPFILTERS)
- Store 1 Sales (KEEPFILTERS)

Top & Best Practices

- Total Baked
- Total Profit
- Total Sold

Calendar

- Date ID

2) Change the formatting accordingly over here.

1) Create the Total Sold measure.

Ultimately want to divide the quantity baked by the ALLSELECTED quantity baked.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: % of Total Sold (ALLSELECTED) Data category: Unassigned New Quick measure measure

Home table: Measure table: \$ - % , . : ;

Structure Formatting Properties Calculations

1 % of Total Sold (ALLSELECTED) =
2 VAR SelectedQuantitySold =
3 CALCULATE(
4 [Total Sold],
5 ALLSELECTED(
6)
7)
8 VAR Ratio =
9 DIVIDE(
10 [Total Sold],
11 SelectedQuantitySold,
12 "0"
13)
14 RETURN
15 Ratio

Visualizations Fields

Search: % of Store-Level Sales, % of Total Baked (ALLSELECTED), % of Total Sold (ALLSELECTED), % Quantity Sold to Females, % Store Sales (REMOVEFILTERS), Cost (CURRENCY), Count of Product ID (DISTINCT), Count of Product ID (VALUES), Customer Sales (ALLEXCEPT Assignment), Customer Sales (ALLEXCEPT), Customer Sales (ALLESELECTED), Customer Sales (Last Year), Customer Sales LY (ODALESC), Customer Sales LY (ISBLANK), Orders by Females (Wrong), Quantity Sold (SELECTEDVALUE), Retail Price (SELECTEDVALUE), Scalar Measures, Store 3 Sales (KEEPFILTERS), Store 3 Sales of whole Beans & Teas (ALL), Store 3 Sales of whole Beans & Teas (CALCULATE), Store 3 Profit, Store 5 Profit (KEEPFILTERS), Store 5 Sales (KEEPFILTERS), Store 5 Sales (XKEEPFILTERS), Store 8 Sales (KEEPFILTERS), Tip & Best Practices, Total Baked, Total Profit, Total Sold, Calendar, Date_ID

2) Change the formatting accordingly over here.

1) Objective 2 to create measures for % of Total Baked & % of Total Sold that total 100% based on a slicer selection of product type.

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data > Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Calculations Share

Transaction Date

01/01/2017 30/06/2019

product_group	Customer Sales	Customer Sales (ALLSELECTED)
Add-ons	\$51,060.00	\$4,252,704.88
Beverages	\$3,282,118.55	\$4,252,704.88
Food	\$501,291.32	\$4,252,704.88
Merchandise	\$83,784.00	\$4,252,704.88
Whole Bean/Teas	\$334,451.01	\$4,252,704.88
Total	\$4,212,704.88	\$4,252,704.88

product_group, product_type, product

- Add-ons
- Beverages
- Food
- Merchandise
- Whole Bean/Teas

1) Within this slicer add product type & product.

Visualizations Fields

product

Measure Table

- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)

Food Inventory

- product_id

Product Lookup

- Column
- current_cost
- current_retail_price
- current_wholesale_price
- new_product_fn
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- promo_fn
- tax_exempt_fn
- unit_of_measure

Sales by Store

- product_id

VALUES Example

- product_group

File CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED +

Maven Router - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Calculations Publish

Document Data Queries Insert Calculations Share

Transaction Date

product_group Customer Sales Customer Sales (ALLSELECTED)

product_group	Customer Sales	Customer Sales (ALLSELECTED)
Add-ons	\$11,000.00	\$4,252,704.88
Beverages	\$3,282,116.88	\$4,252,704.88
Food	\$301,291.32	\$4,252,704.88
Merchandise	\$83,784.00	\$4,252,704.88
Whole Bean/Teas	\$334,431.01	\$4,252,704.88
Total	\$4,252,704.88	\$4,252,704.88

product_group, product_type, product

Food

Filters

Search

Filters on this visual

product is (All)

product_group is Food

Filter type Basic filtering

Search

Select all

Add-ons

Beverages

Food

Merchandise

Whole Bean/Teas

Require single selection

product_type is (All)

Add data fields here

Drill through

Cross-report Off

Keep all filters On

Add drill-through fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Visualizations

Fields

Product

Measure Table

Count of Product ID (DISTINCT)

Count of Product ID (VALUES)

Food Inventory

product_id

Product Lookup

Column

current_cost

current_retail_price

current_wholesale_price

new_product_mn

product

product_category

product_description

product_group

product_id

product_type

promo_yrh

tax_exempt_yrh

unit_of_measure

Sales by Store

product_id

VALUES Example

product_group

ASSIGNMENT ALLSELECTED

Page 9 of 9

2) Expand the filters.

3) Go to the product group options & filter for Food from the list.

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More

New measure measure Quick

Transform Refresh data New visual Text box More

Measure Calculations Share

Transaction Date

01/01/2017 10/04/2019

product_group Customer Sales Customer Sales (ALLSELECTED)

Food	\$157,805.02	\$157,805.02
Total	\$157,805.02	\$157,805.02

4) By clicking here you can expand all the categories within the slicer.

product_group, product_type, product

- Food
 - Biscotti
 - Chocolate Chip Biscotti
 - Ginger Biscotti
 - Hazelnut Biscotti
 - Pastry
 - Almond Croissant
 - Chocolate Croissant
 - Croissant
 - Scone
 - Cranberry Scone
 - Ginger Scone
 - Jumbo Savory Scone
 - Oatmeal Scone
 - Scottish Cream Scone

Visualizations Fields

Product

Measure Table

- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)

Food Inventory

- product_id

Product Lookup

- Column
- current_cost
- current_retail_price
- current_wholesale_price
- new_product_tm
- product
- product_category
- product_description
- product_group
- product_id
- product_type
- promo_tm
- tax_exempt_tm
- unit_of_measure

Sales by Store

- product_id

VALUES Example

- product_group

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Data Queries Calculations Share

Transaction Date
01/01/2017 30/04/2019

product_group, product_type, product

- Food
 - Biscotti
 - Chocolate Chip Biscotti
 - Ginger Biscotti
 - Hazelnut Biscotti
 - Pastry
 - Almond Croissant
 - Chocolate Croissant
 - Croissant
 - Scone
 - Cranberry Scone
 - Ginger Scone
 - Jumbo Savory Scone
 - Oatmeal Scone
 - Scottish Cream Scone

5) Add the product type & product fields to the matrix.

6) Clear out previous values in values in matrix & add the below values into the matrix.

- Total Baked
- Total Sold
- % of Total Baked (ALLSELECTED)
- % Total Sold (ALLSELECTED)

product_type	Total Baked	Total Sold	% of Total Baked (ALLSELECTED)	% of Total Sold (ALLSELECTED)
Biscotti	136,350	34,921	21.83%	24.69%
Chocolate Chip Biscotti	45,630	11,616	7.04%	8.21%
Ginger Biscotti	45,360	11,007	7.00%	7.78%
Hazelnut Biscotti	45,360	12,298	7.00%	8.70%
Pastry	135,486	42,819	29.99%	30.28%
Almond Croissant	44,656	11,761	6.89%	6.32%
Chocolate Croissant	45,630	10,217	7.04%	13.59%
Croissant	45,198	11,841	6.97%	8.37%
Scone	376,446	63,693	58.07%	45.63%
Cranberry Scone	45,198	12,554	6.97%	6.88%
Ginger Scone	120,480	15,849	18.58%	11.21%
Jumbo Savory Scone	45,630	12,544	7.04%	8.87%
Oatmeal Scone	120,480	11,034	18.58%	7.80%
Scottish Cream Scone	44,656	11,712	6.89%	8.28%
Total	648,282	141,433	100.00%	100.00%

Visualizations
Filters
Fields

Search

Measure Table

- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOTECALC)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales (YTD (CALC))
- Customer Sales LY (SBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL...)
- Store 3 Sales of whole Beans & Teas (CAL...
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Top & Best Practices
- Total Baked
- Total Profit
- Total Sold

Calendar

CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSEL

Page 9 of 9

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name: % of All Baked
Home table: Measure Table
Format: \$ - % , . : Data category: Unassigned

Structure Formatting Properties Calculations

1 % of All Baked =
2 VAR AllBaked =
3 CALCULATE(
4 [Total Baked],
5 REMOVEFILTERS(
6 'Food Inventory'
7)
8)
9 VAR Ratio =
10 DIVIDE(
11 [Total Baked],
12 AllBaked,
13 "0"
14)
15 RETURN
16 Ratio

Visualizations Fields

Search: % of All Baked

- % of All Baked
- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALL) (EXCEPT Assignment)
- Customer Sales (ALL) (EXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Tea (ALL)
- Store 3 Sales of whole Beans & Tea (CALCULATE)
- Store 3 Profit
- Store 3 Profit (X) (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Top & Best Practices
- Total Baked
- Total Profit
- Total Sold

1) Create the % of All Baked measure.
2) Change the formatting accordingly over here.

CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSEL + Calendar

Maven Roasters - Analysis - Power BI Desktop

File Home Insert Modeling View Help

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Insert More visuals New measure measure Calculations Share

Dashboard Data Queries Visualizations Fields

Transaction Date
01/01/2017 30/04/2019

product_group, product_type, product

- Food
 - Biscotti
 - Chocolate Chip Biscotti
 - Ginger Biscotti
 - Hazelnut Biscotti
 - Pastry
 - Almond Croissant
 - Chocolate Croissant
 - Croissant
 - Scone
 - Cranberry Scone
 - Ginger Scone
 - Jumbo Savory Scone
 - Oatmeal Scone
 - Scottish Cream Scone
- Pastry
 - Almond Croissant
 - Chocolate Croissant
 - Croissant
- Scone
 - Cranberry Scone
 - Ginger Scone
 - Jumbo Savory Scone
 - Oatmeal Scone
 - Scottish Cream Scone

Key objective 2 is now complete. Create measures for % of Total Baked & % of Total Sold that total 100% based on a slicer selection of product type.

product_type	Total Baked	Total Sold	% of Total Sold (ALLSELECTED)	% of Total Baked (ALLSELECTED)	% of All Baked
Biscotti	136,350	34,921	24.49%	21.03%	21.03%
Chocolate Chip Biscotti	45,630	11,816	8.21%	7.64%	7.64%
Ginger Biscotti	45,360	11,007	7.78%	7.00%	7.00%
Hazelnut Biscotti	45,360	12,298	8.70%	7.00%	7.00%
Pastry	135,486	42,819	30.28%	29.90%	29.90%
Almond Croissant	44,634	11,761	8.32%	6.89%	6.89%
Chocolate Croissant	45,630	19,217	13.59%	7.04%	7.04%
Croissant	45,198	11,841	8.37%	6.97%	6.97%
Scone	376,446	63,693	45.03%	54.07%	54.07%
Cranberry Scone	45,198	12,354	8.86%	6.97%	6.97%
Ginger Scone	120,480	15,649	11.21%	18.58%	18.58%
Jumbo Savory Scone	45,630	12,544	8.57%	7.04%	7.04%
Oatmeal Scone	120,480	11,034	7.12%	18.58%	18.58%
Scottish Cream Scone	44,658	11,712	8.28%	6.89%	6.89%
Total	648,282	141,433	100.00%	100.00%	100.00%

Visualizations Fields

Search

% of All Baked
 % of Store-Level Sales
 % of Total Baked (ALLSELECTED)
 % of Total Sold (ALLSELECTED)
 % Quantity Sold to Females
 % Store Sales (REMOVEFILTERS)
 Cost (CURRENCY)
 Count of Product ID (DISTINCT)
 Count of Product ID (VALUES)
 Customer Sales (ALLEXCEPT Assignment)
 Customer Sales (ALLEXCEPT)
 Customer Sales (ALLSELECTED)
 Customer Sales (Last Year)
 Customer Sales (V(DATE))
 Customer Sales (V(SALARIO))
 Orders by Females (Wrong)
 Quantity Sold (SELECTEDVALUE)
 Retail Price (SELECTEDVALUE)

Scalar Measures

Store 3 Sales (KEEPFILTERS)
 Store 3 Sales of whole Beans & Teas (ALL)
 Store 3 Sales of whole Beans & Teas (CAL)
 Store 3 Profit
 Store 5 Profit (KEEPFILTERS)
 Store 5 Sales (KEEPFILTERS)
 Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

Total Baked
 Total Profit
 Total Sold

Assignment ALLSEL

Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSEL +

Page 9 of 9

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help

Power Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Dashboard Data Queries Calculations Share

Transaction Date
01/01/2017 30/04/2019

product_group, product_type, product

- Food
 - Biscotti
 - Chocolate Chip Biscotti
 - Ginger Biscotti
 - Hazelnut Biscotti
 - Pastry
 - Almond Croissant
 - Chocolate Croissant
 - Croissant
 - Scone
 - Cranberry Scone
 - Ginger Scone
 - Jumbo Savory Scone
 - Oatmeal Scone
 - Scottish Cream Scone

Total Baked Total Sold % of Total Sold % of Total Baked % of All Baked

	Total Baked	Total Sold	% of Total Sold	% of Total Baked	% of All Baked
<input type="checkbox"/> Biscotti	45,000	11,042	23.65%	20.86%	6.94%
Chocolate Chip Biscotti	15,120	3,156	6.78%	7.01%	2.33%
Ginger Biscotti	14,540	3,849	8.24%	6.52%	2.30%
Hazelnut Biscotti	14,340	4,037	8.63%	6.92%	2.30%
<input type="checkbox"/> Pastry	45,304	14,193	30.40%	21.80%	6.99%
Almond Croissant	15,210	3,924	8.40%	7.05%	2.35%
Chocolate Croissant	15,120	6,488	13.90%	7.01%	2.33%
Croissant	14,976	3,781	8.10%	6.94%	2.31%
<input type="checkbox"/> Scone	125,466	21,458	45.95%	38.15%	10.35%
Cranberry Scone	14,976	4,423	9.48%	6.94%	2.31%
Ginger Scone	40,000	5,394	11.55%	10.58%	6.18%
Jumbo Savory Scone	15,120	3,812	8.14%	7.01%	2.33%
Oatmeal Scone	40,000	5,501	10.7%	10.58%	6.18%
Scottish Cream Scone	15,210	4,516	9.69%	7.03%	2.33%
Total	215,772	46,693	100.00%	100.00%	33.28%

store_id

- 2
- 3
- 5
- 8

Visualizations

Fields

store_id

Food Inventory

- Biscotti

Selected by Store

- store_id

Store Lookup

- store_id

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Assignment ALLSEL

CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSEL

Page 9 of 9

Key objective 3 is now complete. Build a visual to compare the % of All Baked to % of Total Baked. Which food item is baked the most for store 8?

4) Objective 3 to build a visual to compare the % of All Baked to % of Total Baked. Which food item is baked the most for store. Add a slicer, assign store id to visual & select store 8. The expected values appear in the matrix above i.e. duplicate percentages now fixed.

SELECTCOLUMNS

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

SELECTCOLUMNS()

Returns a table with selected columns from the table plus any new columns specified by the DAX expressions

=SELECTCOLUMNS(Table, Name, Expression, [...])

Any DAX expression that returns a table

Name of the new column to be added, must be wrapped with double quotes. Repeatable

Any expression that returns a scalar value (i.e. column reference, integer or string). Repeatable

Examples:

- 'Sales by Store'
- 'Food Inventory'
- FILTER(
 'Employees',
 'Employees'[ID] IN {2,5,7,9,10})

Examples:

- "Employee Name & ID"
- "Employee Full Name"

Examples:

- 'Employees'[ID]
- [First Name] & " " & [Last Name]



PRO TIP:

SELECTCOLUMNS is an iterator function that's useful when you need to reduce the number of columns in a table for calculation

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name SELECTCOLUMNS...

Mark as date table - Measures Manage relationships New measure Quick New measure column New table Calculations

1 SELECTCOLUMNS Demo =
2 SELECTCOLUMNS(
3 'Employee Lookup',
4 "Employee ID",
5 'Employee Lookup' [staff_id]
6)

Employee ID

1) Create a new table to output employee id's from the data view.

SELECTCOLUMNS starts off with a blank table & returns the requested table.
"Employee ID" personal column name defined.

Fields

Search

Measure Table
Calendar
Date_ID
Month_ID
Month_Name
Quarter_ID
Quarter_Name
Transaction_Date
Week_Desc
Week_ID
Week_Year
Year_ID
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
SELECTCOLUMNS Demo
Store Lookup
VALUES Example

Table: SELECTCOLUMNS Demo (33 rows)

Maven Roasters - Analysis - Power BI Desktop

File Home Help Table tools

Name SELECTEDCOLUMNS...

Mark as date table - Manage relationships New measure Quick New measure column New table

Structure

1 SELECTEDCOLUMNS Demo 2 =
2 SELECTCOLUMNS(←
3 FILTER(
4 'Employee Lookup',
5 'Employee Lookup'[staff_id] IN {6,16,31} ←
6),
7 "Employee ID",
8 'Employee Lookup'[staff_id],
9 "Manager Name & Store",
10 'Employee Lookup'[first_name] & " " & 'Employee Lookup'[last_name] & "-" & 'Employee Lookup'[location]
11)

Fields

Search

Measure Table
Calendar
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
SELECTEDCOLUMNS Demo
Employee ID
SELECTEDCOLUMNS Demo 2
Store Lookup
VALUES Example

Employee ID Manager Name & Store
1 Karen Copper-1
2F Darren Ku-3
32 Lisa Latta-8

SELECTEDCOLUMNS starts off with a blank table & returns the requested table.

Staff id's that will be filtered in the table output below.

Concatenation used to generate the Manager Name & Store in a single line output.

1) Create a new table to output employee id's & Manager Name & Store from the data view. The SELECTEDCOLUMNS function is typically used for other functions.

ADDCOLUMNS

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

ADDCOLUMNS()

Returns a table with selected columns from the table plus any new columns specified by the DAX expressions

=ADDCOLUMNS(Table, Name, Expression, [...])

Any DAX expression that returns a table

Name of the new column to be added, must be wrapped with double quotes. Repeatable

Any expression that returns a scalar value (i.e. column reference, integer or string). Repeatable

Examples:

- 'Sales by Store'
- 'Food Inventory'
- FILTER(
 'Employees',
 'Employees'[ID] IN {2,5,7,9,10})

Examples:

- "Employee Name & ID"
- "Employee Full Name"

Examples:

- 'Employees'[ID]
- [First Name] & " " & [Last Name]



HEY THIS IS IMPORTANT!

ADDCOLUMNS & SELECTCOLUMNS are nearly identical and behave similarly with an exception, SELECTCOLUMNS starts with a blank table whereas ADDCOLUMNS starts with the entire original table and tacks on columns

Maven Reactors - Analysis - Power BI Desktop

File Home Help Table tools

Name ADDCOLUMNS Demo

Mark as date table Manage relationships New measure Quick New measure column New table

Structure Calculations

1 ADDCOLUMNS Demo =
2 ADDCOLUMNS(
3 FILTER(
4 'Employee Lookup',
5 'Employee Lookup'[staff_id] IN {6,16,31}
6),
7 "Employee ID",
8 'Employee Lookup'[staff_id],
9 "Manager Name & Store",
10 'Employee Lookup'[first_name] & " " & 'Employee Lookup'[last_name] & "-" & 'Employee Lookup'[location]
11)

Fields

Search

Measure Table
ADDCOLUMNS Demo
Calendar
Customer Lookup
Employee Lookup
Food Inventory
Product Lookup
Sales by Store
SELECTEDCOLUMNS Demo
SELECTEDCOLUMNS Demo 2
Store Lookup
VALUES Example

staff_id	first_name	last_name	position	start_date	location	Employee ID	Manager Name & Store
6	Karen	Cupper	Store Manager	24/07/2008 00:00:00	3	6	Karen Cupper-3
16	Damen	Xu	Store Manager	20/08/2008 00:00:00	3	16	Damen Xu-3
31	Lita	Littner	Store Manager	02/07/2009 00:00:00	18	31	Lita Littner-18

Staff id field is now redundant & can be removed as currently duplicated in output.

1) Entire filtered table is returned using the ADDCOLUMNS function. All of the Employee Lookup table columns have been returned with the staff id filters that of 6, 16 & 31. Two additional columns (1)Employee ID & (2) Manager Name & Store are returned in the output.

Maven Reactors - Analysis - Power BI Desktop

File Home Help Table tools ADDCOLUMNS Demo

Mark as date table Manage relationships New measure Quick New measure column New table

Structure Calculations Relationships Calculations

1 ADDCOLUMNS Demo =
2 ADDCOLUMNS(
3 FILTER(
4 'Employee Lookup',
5 'Employee Lookup'[staff_id] IN {6,16,31}
6),
7 "Manager Name & Store",
8 'Employee Lookup'[first_name] & " " & 'Employee Lookup'[last_name] & "-" & 'Employee Lookup'[location]
9)

Fields Search Measure Table ADDCOLUMNS Demo Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store SELECTEDCOLUMNS Demo SELECTEDCOLUMNS Demo 2 Store Lookup VALUES Example

staff_id	first_name	last_name	position	start_date	location	Manager Name & Store
6	Karen	Cupper	Store Manager	24/07/2004 00:00:00	3	Karen.Cupper-3
16	Darren	Xu	Store Manager	30/03/2006 00:00:00	5	Darren.Xu-5
31	Luna	Lam	Store Manager	02/07/2009 00:00:00	8	Luna.Lam-8

2) The second Staff id field is now removed.

SUMMARIZE

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

SUMMARIZE()

Creates a summary of the input table grouped by the specified columns

=SUMMARIZE(Table, GroupBy_ColumnName, [Name], [Expression])



**Deprecated and
not recommended**

Any DAX expression that returns a table of data

Examples:

- 'Sales by Store'
- FILTER(
 'Sales by Store',
 'Sales by Store'[product_id] = 16)

*Name of an existing column to be used to create summary groups based on the values found in the column.
Cannot be an expression*

Examples:

- 'Sales by Store'[store_id]
- 'Customer Lookup'[customer_id]



HEY THIS IS IMPORTANT!

SUMMARIZE isn't an aggregation function. Instead it returns all unique combinations of values based on the columns selected

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: customer_id Format: Whole number \$ - % , 0 Summarization: Sum Data category: Uncategorized Sort by column Data groups Manage relationships New column Properties

Layout Formatting

1 SUMMARIZE Demo =
2 SUMMARIZE(
3 'Sales by Store',
4 'Sales by Store'[customer_id]
5)

customer_id

5463
5893
5412
5878
5479
5541
5195
5211
5489
5373
5162
5394
5491
5180
5477
5450
5592
5291
5481
5290
5691
5403
5278
5400

1) Example of the SUMMARIZE function outputting the customer id details.

2) Additional details provided here when selecting the customer id column.

Fields

Search

- Measure Table
- ADDCOLUMNS Demo
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store
- SELECTCOLUMNS Demo
- SELECTCOLUMNS Demo 2
- Store Lookup
- SUMMARIZE Demo
- VALUES Example

Table: SUMMARIZE Demo (2,230 rows) Column: customer_id (2,230 distinct values)

Maven Rooters - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: quantity_sold Format: Whole number \$ - % , . 0 Summarization: Sum Data category: Uncategorized Sort by column: Sort Data groups: Group Manage relationships: Relationships New column: Calculations

Fields

Search

Measure Table ADDCOLUMNS Demo Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Sales by Store SELECTEDCOLUMNS Demo SELECTEDCOLUMNS Demo 2 Store Lookup SUMMARIZE Demo VALUES Example

1 SUMMARIZE Demo =
2 SUMMARIZE(
3 FILTER(
4 'Sales by Store',
5 'Sales by Store'[quantity_sold] >=3
6),
7 'Sales by Store'[customer_id],
8 'Sales by Store'[quantity_sold]
9)

customer_id quantity_sold

Sort ascending
Sort descending
Clear sort
Clear filter
Clear all filters
Number filters
(Select all)
3
4
5
6
7
OK Cancel

1) Example of the SUMMARIZE function outputting the customer id details for sales by store quantity sold ≥ 3 . 4 distinct values in the quantity sold column & in the customer id column we have 311 rows.

ASSIGNMENT: SUMMARIZE

KEY OBJECTIVES

- 1) Use **SUMMARIZE** to create a table that only shows the days with unsold pastries & columns needed to calculate sold, unsold & lost revenue.
- 2) Build a matrix that shows the results by product & store.

File Home Help Table tools

Name: Unsold Pastries

Mark as date table Relationships New measure Quick New measure column New table Calculations

1 Unsold Pastries =

2 SUMMARIZE(

3 FILTER(

 'Food Inventory',
 'Food Inventory'[quantity_sold] <> 'Food Inventory'[quantity_start_of_day]
6),
 'Food Inventory'[transaction_date],
 'Food Inventory'[store_id],
 'Food Inventory'[quantity_sold],
 'Food Inventory'[quantity_start_of_day],
 'Product Lookup'[current_retail_price],
 'Product Lookup'[product]

13)

transaction_date | store_id | quantity_sold | quantity_start_of_day | current_retail_price | product
 01/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 02/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 03/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 04/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 05/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 06/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 07/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 08/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 09/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 10/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 11/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 12/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 13/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 14/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 15/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 16/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 17/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 18/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 19/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 20/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti
 21/01/2017 00:00:00 | 1 | 1 | 10 | 3.3 | Chocolate Chip Biscotti

Fields

Search

- Measure Table
- ADDCOLUMNS Demo
- Calendar
- Customer Lookup
- Employee Lookup
- Food Inventory
- Product Lookup
- Sales by Store
- SELECTCOLUMNS Demo
- SELECTCOLUMNS Demo 2
- Store Lookup
- Unsold Pastries
- VALUES Example

1) Objective 1 to use SUMMARIZE to create a table that only shows the days with unsold pastries & columns needed to calculate sold, unsold & lost revenue. Example of the SUMMARIZE function outputting the Unsold Pastries.

Food quantity sold does not equal the food baked or food quantity start of day. This will show us the unsold pastries.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Dataset Data Queries Insert Calculations Share

Transaction Date
01/01/2017 30/04/2019

Biscotti Chocolate Chip Biscotti Ginger Biscotti Hazelnut Biscotti Pastry Almond Croissant Chocolate Croissant Croissant Scone Cranberry Scone Total 136,350 45,630 45,360 45,360 135,486 44,658 45,630 45,198 376,446 45,198 648,282 34,92 11,61 11,00 12,29 42,81 11,76 19,21 11,84 63,69 12,55 141,43

store_id

Almond Croissant Chocolate Chip Biscotti Chocolate Croissant Cranberry Scone Croissant Ginger Biscotti Ginger Scone Hazelnut Biscotti Jumbo Savory Scone Oatmeal Scone Scottish Cream Scone

Almond Croissant Chocolate Chip Biscotti Chocolate Croissant Cranberry Scone Croissant Ginger Biscotti

2) Perform the below actions.

a) Add a new matrix to the report.
b) Add the store_id & product row fields to the rows.

Visualizations

Filters

Fields

UNSOLO

Unsold Pastries

- Σ current_retail_price
- product
- Σ quantity_sold
- Σ quantity_start_of_day
- store_id
- transaction_date

Row: store_id, product

Column: Add data fields here

Values: Add data fields here

Drill through

Cross-report: Off

Keep all filters

On

Add drill-through fields here

ASSIGNMENT ALLSELECTED

ASSIGNMENT SU

Page 10 of 10

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Food Sold (SUMMARY...) Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % .0

Measure: Formatting Properties Calculations

2) Change the formatting accordingly over here.

1 Food Sold (SUMMARIZE Table) =
01/02
2 SUM(
3 'Unsold Pastries'[quantity_sold]
4)

Ginger Biscotti 45,360 11,00 Chocolate Croissant
Hazelnut Biscotti 45,360 12,29 Cranberry Scone
Pastry 135,486 42,81 Croissant
Almond Croissant 44,658 11,76 Ginger Biscotti
Chocolate Croissant 45,630 19,21 Ginger Scone
Croissant 45,198 11,84 Hazelnut Biscotti
Scone 376,446 63,69 Jumbo Savory Scone
Cranberry Scone 45,198 12,55 Oatmeal Scone
Total 648,282 141,43 Scottish Cream Scone

5 Almond Croissant
Chocolate Chip Biscotti
Chocolate Croissant
Cranberry Scone
Croissant
Ginger Biscotti
Oatmeal Scone

Visualizations Fields

UN SOLD

Unsold Pastries

- current_retail_price
- product
- quantity_sold
- quantity_start_of_day
- store_id
- transaction_date

Rows: store_id, product

Columns: Add data fields here

Values: Add data fields here

Drill through: Off

Cross-report: Off

Keep all filters: On

Add drill-through fields here

1) Create the Food Sold (SUMMARIZE Table) measure.

CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED ASSIGNMENT SU

Maven Roastem - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Food Unsold (SUM...)
Home table: Measure Table
Format: % Format Whole number
Data category: Uncategorized

Structure Properties Calculations

2) Change the formatting accordingly over here.

1 Food Unsold (SUMMARIZE Table) =
1 SUMX(
2 'Unsold Pastries',
3 'Unsold Pastries'[quantity_start_of_day] - 'Unsold Pastries'
4 [quantity_sold]

5

	Quantity	Price
Hazelnut Biscotti	45,360	12,29
Pastry	135,486	42,81
Almond Croissant	44,650	11,76
Chocolate Croissant	45,630	19,21
Croissant	45,198	11,84
Scone	376,446	63,69
Cranberry Scone	45,190	12,55
Total	648,282	141,43

Unsold Pastries

- Cranberry Scone
- Croissant
- Ginger Biscotti
- Ginger Scone
- Hazelnut Biscotti
- Jumbo Savory Scone
- Oatmeal Scone
- Scottish Cream Scone

- Almond Croissant
- Chocolate Chip Biscotti
- Chocolate Croissant
- Cranberry Scone
- Croissant
- Ginger Biscotti
- Pastry

Visualizations Fields

Search

Measure Table

- % of All Baked
- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Count (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)

Scalar Measures

- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales of whole Beans & Tea (All ...)
- Store 3 Sales of whole Beans & Tea (CAL...
- Store 3 Profit
- Store 3 Profit (KEEPPFILTERS)
- Store 3 Sales (KEEPPFILTERS)
- Store 8 Sales (KEEPPFILTERS)

Tip & Best Practices

- Total Baked

File Home Insert Modeling View Help Calculate Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED +

Page 10 of 10

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

Openers Data Queries Calculations Share

Transaction Date
01/01/2017 30/04/2019

	Count	Total
Biscotti	136,350	14,92
Chocolate Chip Biscotti	45,600	11,61
Ginger Biscotti	45,360	11,00
Hazelnut Biscotti	45,360	12,29
Pastry	136,486	42,81
Almond Croissant	44,658	11,76
Chocolate Croissant	45,630	10,21
Croissant	45,198	11,84
Scone	376,446	63,69
Cranberry Scone	45,198	12,53
Total	648,282	141,43

store_id Food Sold (SUMMARIZE Table) Food Unsold (SUMMARIZE Table)

store_id	Food Sold (SUMMARIZE Table)	Food Unsold (SUMMARIZE Table)
3	45454	169,934
3	3798	10,764
3	3893	11,333
3	6152	9,804
3	4123	10,979
3	3537	11,543
3	3868	11,305
3	4153	15,877
3	4094	11,206
3	4421	10,843
3	3918	36,116
3	3567	10,941
3	4216	167,836
5	4,079	10,847
5	4,469	10,697
5	641	8,777
5	4001	11,114
5	140893	506,849
Total	140893	506,849

Visualizations Fields

Rows: store_id, product

Columns: Add data fields here

Values: Food Sold (SUMMARIZE Table), Food Unsold (SUMMARIZE Table)

Drill through: Off

Cross-report: Off

Keep all filters: On

Add drill-through fields here

Measure Table

- % of All Baked
- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LY (COALESCE)
- Customer Sales LY (ISBLANK)
- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL -)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 8 Sales (KEEPFILTERS)
- Tip & Best Practices
- Total Baked

CALCULATE Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED +

3) Add the below measures to the matrix within the values area.

- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name Lost Revenue (SUM... P% Format \$ - % # . Auto Data category Uncategorized New Quick measure measure

Home table Measure Table Structure Formatting Properties Calculations

1 Lost Revenue (SUMMARIZE Table) =
2 SUMX(
3 'Unsold Pastries',
4 [Food Unsold (SUMMARIZE Table)] * 'Unsold Pastries'[current_retail_price]
5)

Chocolate Chip Biscotti 45,630 11,61
Ginger Biscotti 45,360 11,00
Hazelnut Biscotti 45,360 12,29
Pastry 135,486 42,81
Almond Croissant 44,658 11,76
Chocolate Croissant 45,630 10,21
Croissant 45,198 11,84
Scone 376,446 63,69
Cranberry Scone 45,198 12,53
Total 648,282 141,43

3 Almond Croissant 45454 169,934
Chocolate Chip Biscotti 3893 11,333
Chocolate Croissant 4152 9,804
Cranberry Scone 4123 10,979
Croissant 1557 11,543
Ginger Biscotti 3868 11,306
Ginger Scone 4155 35,877
Hazelnut Biscotti 4094 11,206
Jumbo Savory Scone 4421 10,843
Oatmeal Scone 3916 36,116
Scottish Cream Scone 3567 10,941
4 48916 167,836
Almond Croissant 4039 10,847
Chocolate Chip Biscotti 4549 10,697
Chocolate Croissant 6413 8,777
Cranberry Scone 4036 11,114
Total 140893 506,849

Visualizations Fields

Search

Count of Product ID (DISTINCT)
Count of Product ID (VALUES)
Customer Sales (ALL(Except Assignment))
Customer Sales (ALL(Except))
Customer Sales (AllSelected)
Customer Sales (Last Year)
Customer Sales (LY (COALESCE))
Customer Sales (LY (ISBLANK))
Food Sold (SUMMARIZE Table)
Food Unsold (SUMMARIZE Table)
Lost Revenue (SUMMARIZE Table)
Orders by Females (Wrong)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Scalar Measures
Store 3 Sales (KEEPFILTERS)
Store 3 Sales of whole Beans & Teas (All)
Store 3 Sales of whole Beans & Teas (CALCULATE)
Store 3 Profit
Store 3 Profit (KEEPFILTERS)
Store 3 Sales (KEEPFILTERS)
Store 8 Sales (KEEPFILTERS)
Tip & Best Practices
Total Baked
Total Profit
Total Sold
ADDCOLUMNS Demo
First_name
Last_name
Location
Manager Name & Store

1) Objective 1 part two to use SUMMARIZE to create a table that only shows the days with unsold pastries & columns needed to calculate lost revenue.

2) Change the formatting accordingly over here.

Key objective 1 is now complete. Use SUMMARIZE to create a table that only shows the days with unsold pastries & columns needed to calculate sold, unsold & lost revenue.

Key objective 2 is now complete. Build a matrix that shows the results by product & store.

The screenshot shows the Microsoft Power BI Data View interface. On the left, there's a card for 'Transaction Date' with filters for '01/01/2017' and '30/04/2019'. Below it is a matrix visualization with three columns: 'store_id', 'Food Sold (SUMMARIZE Table)', and 'Lost Revenue (SUMMARIZE Table)'. The matrix displays data for various products across different stores. A blue arrow points from the text below to the 'Lost Revenue (SUMMARIZE Table)' column header in the matrix. On the right, the 'Fields' pane is open, showing the 'Measure Table' section with numerous measures listed under 'Values', such as '% of All Baked', 'Customer Sales (ALLEXCEPT Assignment)', and 'Lost Revenue (SUMMARIZE Table)'. Another blue arrow points from the text below to the 'Lost Revenue (SUMMARIZE Table)' entry in the 'Values' list.

3) Add the Lost Revenue (SUMMARIZE Table) measures to the matrix within the values area. You can see there is over \$1.7m in lost revenue because of various products gone unsold.

3) Add the Lost Revenue (SUMMARIZE Table) measures to the matrix within the values area. You can see there is over \$1.7m in lost revenue because of various products gone unsold.

Generating New Data

In this section, we'll review **four methods** of manufacturing data from scratch:

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

ROW()

Returns a single row table with new columns specified by the DAX expression(s)

=**ROW**(Name, Expression, [...]])

DATABASE()

Returns a table containing new, static data

=**DATABASE**(Name, Type, [...]], Data)

GENERATESERIES()

Returns a single column table containing sequential values, based on a given increment

=**GENERATESERIES**(StartValue, EndValue, [IncrementValue])

Table Constructor { }

Uses curly brackets to return a table containing one or more columns and records

= { scalarExpr1, scalarExpr2, ... }

OR

= { (scalarExpr1, scalarExpr2, ...),
 (scalarExpr1, scalarExpr2, ...),
 }

ROW

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

ROW()

Returns a table with a single row containing values that result from the expressions given to each column

=ROW(Name, Expression, [Name, Expression, [...]])

Name of the new column, enclosed in double quotes. Repeatable.

Any DAX expression that returns a single scalar value

Examples:

- "Profit"
- "Cost"

Examples:

- [Customer Sales]
- SUM('Food Inventory'[quantity_start_of_day])

```
1 ROW Example =  
2 ROW(  
3     "Customer Sales", [Customer Sales],  
4     "Items Ordered", [Quantity Sold]  
5 )
```



Customer Sales	Items Ordered
4252704.88	1305637

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name: Row Demo

Mark as data table - Manage relationships New measure Quick New measure column New table

Measure Calculations

1 Row Demo =
2 ROW(
3 "Customer Sales", [Customer Sales]
4)

Customer Sales 4.212704 M

Fields

Search

Measure Table ADDCOLUMNS D... Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Unsold Products VALUES Example

1) Go to the Data view & create a New Table call it Row Demo.

2) The result is a one row, one column table that contains our total customer sales total over 4.2m.

Maven Rooters - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name Row Demo

Mark as data table - Manage relationships New measure Quick New measure column New table

Dimensions Relationships Calculations

Row Demo

1 Row Demo =
2 ROW(
3 "Customer Sales", [Customer Sales],
4 "Items Ordered",
5 SUMX(
6 'Sales by Store',
7 'Sales by Store'[quantity_sold]
8)
9
10)

Customer Sales Items Ordered
4232704.88 1005617

Fields

Measure Table ADDCOLUMNS D... Calendar Customer Lookup Employee Lookup Food Inventory Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Unsold Pastries VALUES Example

3) You can continue to add more pairs of names & expressions to add additional columns. Items ordered column will be added in this example. The SUMX function will return a value & not a measure, The sales by store table is used to obtain the quantity sold amount.

DATABASE

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

DATABASE()

Returns a table with a fixed set of values

=DATABASE(Name, Type, [Name, Type], [...], Data)

Column name (in quotes)

Column data type

Data points to add to table, called and separated with curly brackets ({})

Examples:

- "Test Number"
- "Value"

Examples:

- STRING
- DOUBLE
- INT
- CURRENCY

Examples:

- {"First", 1},
- {"Second", 2},
- {"Third", 3}



HEY THIS IS IMPORTANT!

DATABASE can only accept **fixed data inputs** (no table or column references, expressions or calculations)

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name DATABASE Demo

Mark as data table = Manage relationships New measure Quick New measure column New table

Calendars Relationships Calculations

1 DATABASE Demo =
2 DATABASE(
3 "Test Number", INTEGER,
4 {
5 {1} ← This is the data point for the "Test Number" column.
6 }
7)

Test Number

1) Go to the Data view & create a New Table call it DATABASE Demo. Curly bracket is used to define the table, the curly bracket used again to enter the data for the first column, close out the row, close out the table using the curly brackets & close out the function using the bracket.

2) Test Number is the column name the value "1" is provided based on the above code.

Fields

Measure Table ADDCOLUMN D... Calendar Customer Lookup DATABASE Demo Employee Lookup Food Inventory Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Unpivot Palettes VALUES Example

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name DATABASE Demo

Mark as data table - Manage relationships New measure Quick New measure column New table

Structure

1 DATABASE Demo =
2 DATABASE(
3 "Test Number", INTEGER,
4 "Heads | Tails", STRING,
5 "Trial", INTEGER,
6 {
7 {1, "Heads", 1},
8 {2, "Tails", 1},
9 {3, "Tails", 1}
10 }
11)

Fields

Search

Measure Table
ADDCOLUMNS D...
Calendar
Customer Lookup
DATABASE Demo
Employee Lookup
Food Inventory
Product Lookup
Row Demo
Sales by Store
SELECTEDCOLUMN...
SELECTEDCOLUMN...
Store Lookup
Unsold Records
VALUES Example

1) Another example of adding data to a table.

Test Number Heads | Tails Trial
1 Heads 1
2 Tails 1
3 Tails 1

GENERATESERIES

Calculated Tables

Filtering Tables

Manipulating
Tables

Generating Data

GENERATESERIES()

Returns a one column table populated with sequential values

=**GENERATESERIES(StartValue, EndValue, [IncrementValue])**

Start & End value for the series
(can be positive or negative)

Examples:

- 0, 1, 10.25, 50.234, etc.
- -1, -10.75, -45.5, etc.

Value used to increment between Start & End value (must be **non-zero** and positive)

Examples:

- .25, 1, 1.7745, 5, etc.



PRO TIP:

GENERATESERIES is a great way to build a custom range of data to be used as a parameter input

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name GENERATESERIES.D...

Mark as date table - Counters Manage relationships New measure Quick New measure column New table Calculations

Structure

1 GENERATESERIES Demo =
2 GENERATESERIES(
3 -50.5,
4 50.5,
5 10
6)

Value

-50.5
-40.5
-30.5
-20.5
-10.5
-5.5
5.5
15.5
25.5
35.5
45.5

Fields

Search

Measure Table
ADDCOLUMNS.D...
Calendar
Customer Lookup
DATATABLE Demo
Employee Lookup
Food Inventory
GENERATESERIES...
Product Lookup
Rovi Demo
Sales by Store
SELECTEDCOLUMN...
SELECTEDCOLUMN...
Store Lookup
Unsold Products
VALUES Example

1) Create a New Table call it GENERATESERIES Demo.
2) We have a start value at -50.5, ending value at 49.5 in increments of 10.

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name GENERATESERIES D...

Mark as date table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 GENERATESERIES Demo =
2 GENERATESERIES(
3 -50.5,
4 50.5
5)

Value

-50.5

Fields

Search

Measure Table ADDCOLUMNS D... Calendar Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Unsold Products VALUES Example

1) When removing the 10 increment value the numbers will increment by 1 automatically as shown in the below table.

Value
-50.5
-49.5
-48.5
-47.5
-46.5
-45.5
-44.5
-43.5
-42.5
-41.5
-40.5
-39.5
-38.5
-37.5
-36.5
-35.5
-34.5
-33.5
-32.5
-31.5
-30.5
-29.5
-28.5
-27.5
-26.5
-25.5
-24.5
-23.5

Table GENERATESERIES Demo (762 rows)

The Table Constructor

Calculated Tables

Filtering Tables

Manipulating Tables

Generating Data

The **table constructor { }** can be used to build tables containing one or more columns or rows, based on fixed values or DAX expressions which return scalar values

- *NOTE: Column headers can't be defined using table constructor syntax*

1 Single column with multiple rows:

```
1 Table Constructor Single Column Table =  
2 {  
3     [Customer Sales],  
4     [Quantity Sold]  
5 }
```



Value	4252704.88
Value	1305637

3 Multi-row, multi-column table:

```
1 Table Constructor Example =  
2 {  
3     ("This Is the First Column", -- Column 1 Row 1  
4     "This Is the Second Column" -- Column 2 Row 1  
5 ),  
6     ([Customer Sales], -- Column 1 Row 2  
7     [Quantity Sold] -- Column 2 Row 2  
8 ),  
9     ([Sales to Females], -- Column 1 Row 3  
10    [Count of Store Sales] -- Column 2 Row 3  
11 )  
12 }
```



2 Single row with multiple columns:

```
1 Table Constructor Single Row Table =  
2 {  
3     (  
4         [Customer Sales],  
5         [Quantity Sold]  
6     )  
7 }
```



Value1	4252704.88
Value2	1305637

Value1	Value2
This Is the First Column	This Is the Second Column
4252704.88	1305637
224	30

Maven Rooters - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name Table Constructor...

Mark as date table... Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Table Constructor Demo =
2 {
3 1
4 }

Value

Fields

Search

Measure Table ADDCOLUMNS D... Calendar Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Table Constructor Unvoid Parent VALUES Example

1) Create a New Table call it Table Constructor Demo. We will start off with single row, single column table start off using the curly bracket return a single value & close out with a curly bracket.

2) Receive a value of 1. Column header name defaults to Value.

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name Table Constructor ...

Mark as date table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Table Constructor Demo =
2 {
3 1,
4 2,
5 3,
6 4
7 }

Fields

Search

Measure Table ADDCOLUMNS D... Calendar Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES... Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Table Constructor... Unfold Passives VALUES Example

Value

2
3
4

3) To add more rows just use commas with other data inputs. Same single column table now with 4 rows of data.

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools

Name Table Constructor ...

Mark as data table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Table Constructor Demo =
2 {
3 (1,2,3,4)
4 }

Value1 Value2 Value3 Value4
1 1 1 1

Fields

Search

Measure Table ADDCOLUMNS D... Calendar Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIAL... Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Table Constructor... Unsold Pastries VALUES Example

4) Create a table with multiple columns & with 1 row.

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name Table Constructor ...

Mark as data table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Table Constructor Demo =
2 {
3 (1,2,3,4),
4 (5,6,7,8),
5 (9,10,11,12)
6 }

Value1 Value2 Value3 Value4

Value1	Value2	Value3	Value4
1	2	3	4
5	6	7	8
9	10	11	12

Fields

Search

- Measure Table
- ADDCOLUMNS D...
- Calendar
- Customer Lookup
- DATATABLE Demo
- Employee Lookup
- Food Inventory
- GENERATESERIES...
- Product Lookup
- Row Demo
- Sales by Store
- SELECTEDCOLUMN...
- SELECTEDCOLUM...
- Store Lookup
- Table Constructor...
- Unsold Pastries
- VALUES Example

5) To add additional data inputs for multiple rows.



```
1 Table Constructor Demo =  
2 {  
3     ("Customer 1",FORMAT(CALCULATE([Customer Sales],'Customer Lookup'[customer_id] = 1), "CURRENCY"))  
4 }
```

Value1 Value2
Customer 1: \$556.80

 ("Customer 1",FORMAT(CALCULATE([Customer Sales],'Customer Lookup'[customer_id] = 1), "CURRENCY"))

6) What we have here is single row two column table with sales for customer 1 i.e. \$556.80.

- Fields >
- Search
 - Measure Table
 - ADDCOLUMNS D...
 - Calendar
 - Customer Lookup
 - DATATABLE Demo
 - Employee Lookup
 - Food Inventory
 - GENERATESERIES...
 - Product Lookup
 - Row Demo
 - Sales by Store
 - SELECTEDCOLUMN...
 - SELECTEDCOLUMN...
 - Store Lookup
 - Table Constructor
 - Unsold Pastries
 - VALUES Example

Maven Reactors - Analysis - Power BI Desktop

File Home Help Table tools

Name Table Constructor...

Mark as data table Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Table Constructor Demo =
2 {
3 ("Customer 1",FORMAT(CALCULATE([Customer Sales],'Customer Lookup'[customer_id] = 1), "CURRENCY")),
4 ("Customer 79",FORMAT(CALCULATE([Customer Sales],'Customer Lookup'[customer_id] = 79), "CURRENCY"))
5 }

Value1 Value2
Customer 1 \$236.80
Customer 79 \$421.23

Fields >
Search
Measure Table ADDCOLUMNS D... Calendar Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Table Constructor... Unsold Pastries VALUES Example

7) What we have here is multiple row two column table with sales data for customer 1 & 79. Biggest drawback here is you can't add in the column header names.

ASSIGNMENT Generating Data

KEY OBJECTIVES

- 1) Use **DATATABLE** that's needed for forecasts.

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name Target Sales - April ...

Mark as clean table Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Target Sales - April 2019 (DATATABLE) =
2 DATATABLE(
3 "Store ID", INTEGER,
4 "Year", STRING,
5 "Month", STRING,
6 "Bean/Teas Goal", INTEGER,
7 "Beverage Goal", INTEGER,
8 "Food Goal", INTEGER,
9 "Merchandise Goal", INTEGER,
10 {
11 {3, "2019", "April", 268, 15608, 1964, 80},
12 {5, "2019", "April", 277, 14687, 2020, 91},
13 {8, "2019", "April", 377, 15011, 1973, 34}
14 }
15)

Fields

Search

Measure Table
ADOCOLUMNS Demo
Calendar
Customer Lookup
DATATABLE Demo
Employee Lookup
Food Inventory
GENERATESERIES Demo
Product Lookup
Row Demo
Sales by Store
SELECTEDCOLUMNS Demo
SELECTEDCOLUMNS Demo 2
Store Lookup
Table Constructor Demo
Target Sales - April 2019 (DA...
Unsold Pastries
VALUES Example

1) Objective 1 to use DATATABLE that's needed for forecasts. Create a New Table call it Target Sales - April 2019 (DATATABLE). Add the required columns shown in this example associated with formats for generating data.

Key objective 1 is now complete. Use DATATABLE that's needed for forecasts.

Store ID	Year	Month	Bean/Teas Goal	Beverage Goal	Food Goal	Merchandise Goal
3	2019	April	268	15608	1964	80
5	2019	April	277	14687	2020	91
8	2019	April	377	15011	1973	34

Table: Target Sales - April 2019 (DATATABLE) (3 rows)

7) Calculated Table Joins

Introduction - Calculated Table Joins



Calculated table joins are used to combine two or more tables of data; common examples include **CROSSJOIN**, **UNION**, **EXCEPT** and **INTERSECT**

TOPICS WE'LL COVER:

CROSSJOIN

UNION

EXCEPT

INTERSECT

COMMON USE CASES:

- *Blending or combining data across multiple tables*
- *Creating advanced calculations like new vs. returning users or repeat purchase behavior*
- *Querying tables to troubleshoot errors or better understand relationships in a data model*

CROSSJOIN

CROSSJOIN

UNION

EXCEPT

INTERSECT

CROSSJOIN()

Returns a table that contains the cartesian product of the specified tables

=CROSSJOIN(Table, Table, [...]])

List of table expressions to include in the crossjoin

Examples:

- 'Product Lookup'
- VALUES('Sales by Store'[store_id])

IMPORTANT NOTES:

- ✓ Column names must all be **different** in all table arguments
- ✓ The # of rows returned equals the **product of rows** in all tables
- ✓ The # of columns returned equals the **sum of columns** in all tables

Product Category	Product Group
Coffee	Beverages
Tea	Beverages
Bakery	Food
Chocolate	Food

Product Category	Product Group	Store ID
Coffee	Beverages	3
Tea	Beverages	3
Bakery	Food	3
Chocolate	Food	3
Coffee	Beverages	5
Tea	Beverages	5
Bakery	Food	5
Chocolate	Food	5
Coffee	Beverages	8
Tea	Beverages	8
Bakery	Food	8
Chocolate	Food	8

Resulting table contains 12 rows (4*3) and 3 columns (2+1)

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: product_category Data type: Text Summarization: Don't summarize Data category: Uncategorized Sort by column: Sort Data groups: Group Manage relationships: Relationships New column: Calculations

Fields

Search

Measure Table
 % of All Baked
 % of Store-Level Sales
 % of Total Baked (ALLSELECTED)
 % of Total Sold (ALLSELECTED)
 % Quantity Sold to Females
 % Store Sales (K)MOVEDFIRST
 Cost (CURRENCY)
 Count of Product ID (DISTINCT)
 Count of Product ID (VALUES)
 Customer Sales (ALLEXCEPT)
 Customer Sales (ALLEXCEPT)
 Customer Sales (ALLSELECTED)
 Customer Sales (Last Year)
 Customer Sales LY (COALESCE)
 Customer Sales LY (ISBLANK)
 Food Sold (SUMMARIZE Table)
 Food Unsold (SUMMARIZE Table)
 Lost Revenue (SUMMARIZE Table)
 Orders by Females (Wrong)
 Quantity Sold (SELECTEDVALUE)
 Retail Price (SELECTEDVALUE)
 Scalar Measures
 Store 3 Sales (K)(FILTERS)
 Store 3 Sales of whole Beans...
 Store 3 Sales of whole Beans...
 Store 5 Profit
 Store 5 Profit (K)(FILTERS)
 Store 5 Sales (K)(FILTERS)
 Store 6 Sales (KEEPFILTERS)
 Tip & Best Practices

1 CROSSJOIN Demo =
2 CROSSJOIN(
3 'Product Lookup',
4 'Store Lookup'
5)

1) Create a New Table call it CROSSJOIN Demo. We will use CROSSJOIN to join two table together.

2) Combination output of Product Lookup & Store Lookup data tables has been created below.

product_id	product_group	product_category	product_type	product	product_description	unit_of_measures	current_cost	current_wholesale_price	current_retail_price	tax_exempt_ye	process_ye	new_prods
1	Whole Bean/Tess	Coffee beans	Organic Beans	Brazilian Organic	It's like Carnival in a cup. Clean and smooth.	12 oz	3.8	14.4	28.7	N	N	
2	Whole Bean/Tess	Coffee beans	House Blend Beans	Our Old Time Diner Blend	Our packed blend of beans that is reminiscent of the old-t	12 oz	3.8	14.4	28.7	N	N	
3	Whole Bean/Tess	Coffee beans	Espresso Beans	Espresso Roast	Our house blend for a good espresso shot.	1 lb	7.95	21.8	54.75	N	N	
4	Whole Bean/Tess	Coffee beans	Espresso Beans	Primo Espresso Roast	Our premium single source of hand-roasted beans.	1 lb	4.09	20.98	50.49	N	N	
5	Whole Bean/Tess	Coffee beans	Gourmet Beans	Columbian Medium Roast	A smooth cup of coffee anytime of day.	1 lb	3	12	23.7	N	N	
6	Whole Bean/Tess	Coffee beans	Gourmet Beans	Ethiopia	From the home of coffee.	1 lb	4.2	26.8	28.7	N	N	
7	Whole Bean/Tess	Coffee beans	Premium Beans	Jamaican Coffee River	Yum, it will start your day off right.	1 lb	3.95	15.8	29.75	N	N	
8	Whole Bean/Tess	Coffee beans	Premium Beans	Orbit Cat	The most expensive coffee in the world, the cats do all the	5 lb	9	36	45	N	N	
9	Whole Bean/Tess	Coffee beans	Organic Beans	Organic Decaf Blend	Our blend of hand-picked organic beans that have been ro	1 lb	4.5	18	22.5	N	N	
10	Whole Bean/Tess	Coffee beans	Green Beans	Guatemalan Sustainability Blend	Green beans you can roast yourself.	1 lb	2	8	22	N	N	
11	Whole Bean/Tess	Loose Tea	Herbal tea	Lemon Grass	You will think you are Thailand as you sip your tea.	5 oz	0.6	7.16	8.89	N	N	
12	Whole Bean/Tess	Loose Tea	Herbal tea	Peppermint	Cool and refreshing to help calm your nerves.	5 oz	0.6	7.16	8.89	N	N	
13	Whole Bean/Tess	Loose Tea	Black tea	English Breakfast	The traditional cup to start your day.	5 oz	0.6	7.16	8.89	N	N	
14	Whole Bean/Tess	Loose Tea	Black tea	Earl Grey	A full leaf of Orange Pekoe blended with organic oil of berg	5 oz	0.6	7.16	8.89	N	N	
15	Whole Bean/Tess	Loose Tea	Green tea	Serenity Green Tea	Mountain grown and harvested at the optimal time.	5 oz	0.9	7.4	9.29	N	N	
16	Whole Bean/Tess	Loose Tea	Chai tea	Traditional Blend Chai	A traditional blend.	5 oz	0.6	7.16	8.89	N	N	
17	Whole Bean/Tess	Loose Tea	Chai tea	Morning Sunrise Chai	Fair trade and organic and has a warm finish.	5 oz	0.6	7.6	8.5	N	N	
18	Whole Bean/Tess	Loose Tea	Chai tea	Spicy Eye Opener Chai	A spicier blend to awaken your taste buds.	5 oz	0.6	8.76	10.99	N	N	
19	Whole Bean/Tess	Packaged Chocolate	Drinking Chocolate	Dark Chocolate	This drinking chocolate is smooth and creamy.	1 lb	3.29	9.12	6.4	N	N	
20	Whole Bean/Tess	Packaged Chocolate	Organic Chocolate	Sustainably Grown Organic	Certified organic containing the highest quality ingredient	1 lb	3.52	8.08	7.8	N	N	
21	Whole Bean/Tess	Packaged Chocolate	Drinking Chocolate	Chili Mayan	Pairing with spices, this is the most flavorful drinking ch	1 lb	2.69	10.94	13.37	N	N	
22	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Sm	An honest cup of coffee.	8 oz	0.5	1	2.1	N	N	
23	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Pg	An honest cup of coffee.	16 oz	0.823	2.25	2.5	N	N	
24	Beverages	Coffee	Drip coffee	Our Old Time Diner Blend Lg	An honest cup of coffee.	24 oz	0.75	2.5	3.1	N	N	
25	Beverages	Coffee	Organic Brewed coffee	Brazilian Sm	It's like Carnival in a cup. Clean and smooth.	8 oz	0.55	1.8	3.2	N	N	
26	Beverages	Coffee	Organic Brewed coffee	Brazilian Pg	It's like Carnival in a cup. Clean and smooth.	16 oz	0.98	4.8	5.8	N	N	

Table: CROSSJOIN Demo (348 rows) Column: product_category (3 distinct values)

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: product_group
Data type: Text
Summarization: Don't summarize
Data category: Uncategorized
Sort by column
Data groups
Manage relationships
New column
Properties
Formatting
Search

1 CROSSJOIN Demo =
2 CROSSJOIN(
3 VALUES(
4 'Product Lookup'[product_category]
5),
6 VALUES(
7 'Product Lookup'[product_group]
8)
9)

product_group

product_group	product_category
Whole Bean/Tear	Coffee beans
Whole Bean/Tear	Loose Tea
Whole Bean/Tear	Packaged Chocolate
Whole Bean/Tear	Coffee
Whole Bean/Tear	Tee
Whole Bean/Tear	Drinking Chocolate
Whole Bean/Tear	Flavour
Whole Bean/Tear	Bakery
Whole Bean/Tear	Branded
Beverages	Coffee beans
Beverages	Loose Tea
Beverages	Packaged Chocolate
Beverages	Coffee
Beverages	Tee
Beverages	Drinking Chocolate
Beverages	Flavour

Fields

Measure Table
% of All Baked
% of Store-Level Sales
% of Total Baked (ALLSELECTED)
% of Total Sold (ALLSELECTED)
% Quantity Sold to Females
% Store Sales (RMOVEDFROM)
Cost (CURRENCY)
Count of Product ID (VALUES)
Count of Product ID (VALUES)
Customer Sales (ALLEXCEPT -)
Customer Sales (ALLEXCEPT)
Customer Sales (ALLSELECTED)
Customer Sales (Last Year)
Customer Sales LY (COALESCE)
Customer Sales LY (ISBLANK)
Food Sold (SUMMARIZE Table)
Food Unsold (SUMMARIZE Table)
Lost Revenue (SUMMARIZE Table)
Orders by Females (Wrong)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Scalar Measures
Store 3 Sales (KEEPMFILTERS)
Store 3 Sales of whole Beans...
Store 3 Sales of whole Beans...
Store 5 Profit
Store 5 Profit (KEEPMFILTERS)
Store 5 Sales (KEEPMFILTERS)
Store 8 Sales (KEEPFILTERS)
Tip & Best Practices

1) Example of a CROSSJOIN query to combine two specified columns from the Product Lookup table i.e. product category & product group.

Table: CROSSJOIN Demo (45 rows) Column product_group (3 distinct values)

Maven Reactors - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: product_group
Data type: Text
Summarization: Don't summarize
Data category: Uncategorized
Sort by column
Data groups
Manage relationships
New column
Structure
Formatting
Properties
Sort
Group
Relationships
Calculations

Fields

Search

Measure Table

- % of All Baked
- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (R) [MOVEDFROM]
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales LV (COALESCE)
- Customer Sales LV (ISBLANK)
- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)
- Lost Revenue (SUMMARIZE Table)
- Orders by Females (Wrong)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Scalar Measures
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans
- Store 3 Sales of whole Beans
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Tip & Best Practices

1 CROSSJOIN Demo =
2 CROSSJOIN(
3 VALUES(
4 'Product Lookup'[product_category]
5),
6 VALUES(
7 'Product Lookup'[product_group]
8),
9 VALUES(
10 'Store Lookup'[store_id]
11)
12)

product_group product_category store_id

product_group	product_category	store_id
Whole Bean/Tea	Coffee beans	2
Whole Bean/Tea	Loose Tea	2
Whole Bean/Tea	Packaged Chocolate	2
Whole Bean/Tea	Coffee	2
Whole Bean/Tea	Tea	2
Whole Bean/Tea	Dringing Chocolate	2
Whole Bean/Tea	Flavours	2
Whole Bean/Tea	Bakery	2
Whole Bean/Tea	Branded	2
Beverages	Coffee beans	2
Beverages	Loose Tea	2
Beverages	Packaged Chocolate	2
Beverages	Coffee	2
Beverages	Tea	2
Beverages	Dringing Chocolate	2
Beverages	Flavours	2

Table: CROSSJOIN Demo (180 rows) Column: product_group (3 distinct values)

1) In this CROSSJOIN example the Store Lookup table has been added with its associated column store id.

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: store_id Format: Whole number \$ - % , .00 Data type: Whole number Summarization: Sum Data category: Uncategorized Sort by column Data groups Manage relationships New column

Structure Formatting Properties

Sort Group Relationship Calculations

1 CROSSJOIN Demo =
2 CROSSJOIN(
3 VALUES(
4 'Product Lookup'[product_category]
5),
6 VALUES(
7 'Product Lookup'[product_group]
8),
9 FILTER(
10 VALUES(
11 'Store Lookup'[store_id]
12),
13 'Store Lookup'[store_id] = 3
14)
15)

product_group product_category store_id
Whole Bean/Tea Coffee beans 3
Whole Bean/Tea Loose Tea 3
Whole Bean/Tea Packaged Chocolate 3
Whole Bean/Tea Coffee 3
Whole Bean/Tea Tea 3
Whole Bean/Tea Drinking Chocolate 3
Whole Bean/Tea Flavoured 3
Whole Bean/Tea Bakery 3
Whole Bean/Tea Branded 3
Beverages Coffee beans 3

Fields

Search

Measure Table
% of All Sales
% of Store-Level Sales
% of Total Sales (ALLSELECTED)
% of Total Sold (ALLSELECTED)
% Quantity Sold to Females
% Store Sales (REMOVE(DRILLDOWN))
Cost (CURRENCY)
Count of Product ID (DISTINCT)
Count of Product ID (VALUES)
Customer Sales (ALLEXCEPT)
Customer Sales (ALLEXCEPT)
Customer Sales (ALLSELECTED)
Customer Sales (Last Year)
Customer Sales LY (CALENDAR)
Customer Sales LY (SCHEDULE)
Food Sold (SUMMARIZE Table)
Food Unsold (SUMMARIZE Table)
Lost Revenue (SUMMARIZE Table)
Orders by Females (Wrong)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Scalar Measures
Store 3 Sales (KEEPFILTERS)
Store 3 Sales of whole Beans
Store 3 Sales of whole Beans
Store 3 Profit
Store 3 Profit (KEEPFILTERS)
Store 3 Sales (KEEPFILTERS)
Store 3 Sales (KEEPFILTERS)
Store 3 Sales (KEEPFILTERS)
Tip & Best Practices

1) In this CROSSJOIN example the Store Lookup table has been added with its associated column store id. Using the filter function now store id is filtered to 3. Filter functions are great for reducing the size of the data output & slicing of information.

UNION

UNION()

Combines or "stacks" rows from two or more tables sharing the same column structure

CROSSJOIN

UNION

EXCEPT

INTERSECT

=UNION(Table, Table, [...])

Accepts any DAX expression for two (or more) tables with *identical column structure*

Examples:

- 'Sales Target 2019',
'Sales Target 2020'
- 'Sales Target 2019', DATATABLE()

IMPORTANT NOTES:

- ✓ All tables must contain **the same number of columns**
- ✓ Columns are combined by **position** in their respective tables
- ✓ Column names are determined by the **first table expression**
- ✓ Duplicate rows are retained

Date	ABC	XYZ	123
5/1/19	A	X	1
5/1/19	B	Y	2
5/1/19	C	Z	3

Date	ABC	XYZ	123
5/1/20	A	X	1
5/1/20	B	Y	2
5/1/20	C	Z	3

Date	ABC	XYZ	123
5/1/19	A	X	1
5/1/19	B	Y	2
5/1/19	C	Z	3
5/1/20	A	X	1
5/1/20	B	Y	2
5/1/20	C	Z	3

UNION stacks tables together,
just like append!

Maven Reactors - Analysis - Power BI Desktop

File Home Help Table tools

Name UNION Demo

Mark as date table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 UNION Demo =
2 UNION(
3 'Sales by Store',
4 'Product Lookup'
5)

Each table argument of 'UNION' must have the same number of columns.

Fields

Search

Measure Table ADDCOLUMNS D... Calendar CROSSJOIN Demo Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES... Product Lookup Row Demo Sales by Store SELECTEDCOLUMN... SELECTEDCOLUMN... Store Lookup Table Constructor... Target Sales - Apr... UNION Demo Unfold Paste... VALUES Example

Table: UNION Demo (0 rows)

Maven Rooters - Analysis - Power BI Desktop

File Home Help Table tools

Name UNION Demo

Mark as date table Manage relationships New measure Quick New measure column New table

Structure Calculations

1 UNION Demo =
2 UNION(
3 'Target Sales - March 2019 (DATATABLE)',
4 'Target Sales - April 2019 (DATATABLE)'
5)

Fields

Search

Measure Table ADDCOLUMNS Demo Calendar CROSSTAB Demo Customer Lookup DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES Demo Product Lookup Row Demo Sales by Store SELECTEDCOLUMN3 Demo SELECTEDCOLUMN3 Demo 2 Store Lookup Table Constructor Demo Target Sales - April 2019 (DATATABLE) Bean/Teas Goal Beverage Goal Food Goal Merchandise Goal Month Store ID Year Month Bean/Teas Goal Beverage Goal Food Goal Merchandise Goal

1) The Union query has combined the data held in the Target Sales – March 2019 (DATATABLE) & Target Sales – April 2019 (DATATABLE).

Store ID Year Month Bean/Teas Goal Beverage Goal Food Goal Merchandise Goal
3 2019 March 211 12703 1595 53
5 2019 March 254 12451 1696 81
8 2019 March 306 12474 1633 17

2) Example of the data stored in the Target sales for March 2019 & April 2019.

Store ID Year Month Bean/Teas Goal Beverage Goal Food Goal Merchandise Goal
3 2019 April 268 15608 1964 80
5 2019 April 277 14687 2020 91
8 2019 April 377 15011 1973 34

Table UNION Demo (8 rows)

EXCEPT

CROSSJOIN

UNION

EXCEPT

INTERSECT

EXCEPT()

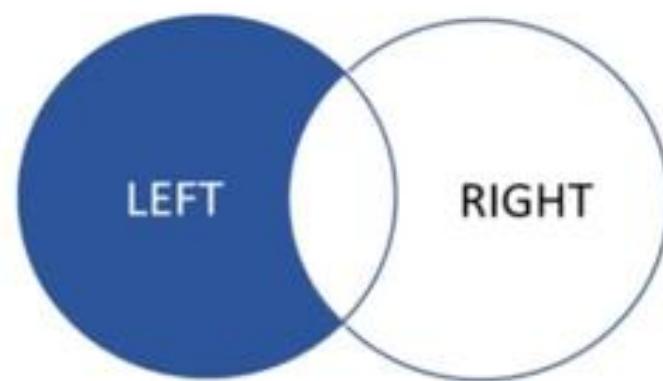
Returns all rows from the left table which do not appear in the right table

=EXCEPT(LeftTable, RightTable)

The left and right tables that will be used for joining
(NOTE: First table must be a table within the data model)

Example (churned customers):

=EXCEPT('Customers 2019','Customers 2020')



Resulting table contains rows which
ONLY appear in the **left** table

IMPORTANT NOTES:

- ✓ Both tables must contain **the same number of columns**
- ✓ Columns are compared based on **positioning** in their respective tables
- ✓ Column names are determined by the **left table**
- ✓ The resulting table does NOT retain **relationships** to other tables (*can't be used as an expanded table*)

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: customer_id Format: Whole number Data type: Whole number Summarization: Sum Sort by column: Sort Data category: Uncategorized Data groups: Group Manage relationships: Relationships New column: Calculations

1 EXCEPT Demo =
 2 EXCEPT(
 3 'Customer Lookup',
 4 FILTER(
 5 VALUES(
 6 'Customer Lookup'
 7),
 8 'Customer Lookup'[customer_since] >DATE(2017,02,01)
 9)
 10)

Fields

Search

- Measure Table
- ADOCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- DATAFILE Demo
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Demo
- Product Lookup
- Row Demo
- Sales by Store
- SELECTEDCOLUMNS Demo
- SELECTEDCOLUMNS Demo 2
- Store Lookup
- Table Constructor Demo
- Target Sales - April 2019 (DATATABLE)
- Target Sales - March 2019 (DATATABLE)
- UNION Demo
- Bean/Teas Goal
- Beverage Goal
- Food Goal
- Merchandise Goal
- Month
- Store ID
- Year
- Unsold Pastries
- VALUES Example

customer_id	home_store	customer_first_name	customer_email	customer_since	loyalty_card_number	birthdate	gender	birth_year
801	J. Aliza Rivers	Yvonne@alizur.com.us	04/02/2017 00:00:00	021-443-0682	13/05/1950 00:00:00	F	1950	
802	J. Seacia Wall	Yvette@seawall.org	06/03/2017 00:00:00	433-122-9277	29/06/1950 00:00:00	F	1950	
803	J. Baya Hampton	Martha@maitreyaedu.com	08/01/2017 00:00:00	478-782-8864	14/08/1950 00:00:00	F	1950	
804	J. Bella Reyes	Tierre@goodtates.net	20/02/2017 00:00:00	885-000-5889	30/09/1950 00:00:00	F	1950	
805	J. Brooke Munoz	William@monotec.com	22/02/2017 00:00:00	579-552-0754	18/11/1950 00:00:00	F	1950	
806	J. Tanisha Wolf	Clinton@varian.edu	24/02/2017 00:00:00	389-238-7140	01/02/1951 00:00:00	F	1951	
807	J. Culintessa Franklin	Audrey@mattos.net	26/02/2017 00:00:00	977-781-8833	17/02/1951 00:00:00	F	1951	
808	J. Charrissa Cobb	Ragan@viamus.net	28/02/2017 00:00:00	618-483-1618	04/04/1951 00:00:00	F	1951	
809	J. Susan Mathis	Cathy@thindunit.edu	20/03/2017 00:00:00	725-654-4328	21/05/1951 00:00:00	F	1951	
810	J. Cynthia Hurley	Miriam@artivita.edu	21/03/2017 00:00:00	305-571-4564	05/07/1951 00:00:00	F	1951	
811	J. Melodie Dawson	Stacy@hoster.us	24/03/2017 00:00:00	884-557-8983	23/08/1951 00:00:00	F	1951	
812	J. Charlotte Weiss	Phelan@brutum.org	26/03/2017 00:00:00	543-135-7142	08/10/1951 00:00:00	F	1951	
813	J. Melvynne Jones	Dellon@Quix.edu	28/03/2017 00:00:00	829-851-0634	25/12/1951 00:00:00	F	1951	

Table: EXCEPT Demo (31 rows) Column: customer_id (31 distinct values)

INTERSECT

INTERSECT()

Returns all the rows from the left table which also appear in the right table

CROSSJOIN

UNION

EXCEPT

INTERSECT

=INTERSECT(LeftTable, RightTable)

The left and right tables that will be used for joining
(NOTE: First table must be a table within the data model)

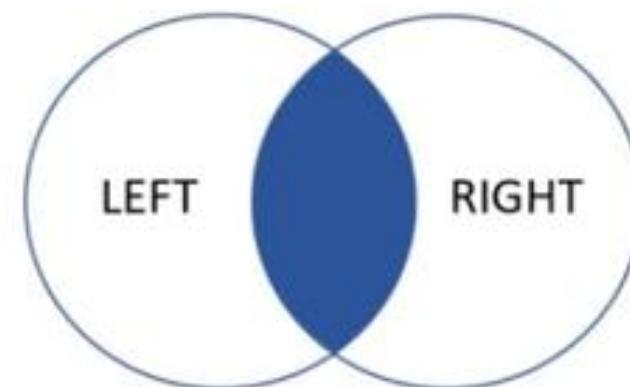
Example (previous month active customers):

LeftTable:

VALUES('Sales'[Customer ID])

RightTable:

CALCULATETABLE(
VALUES("Sales'[Customer ID]"),
DATEADD('Calendar'[Date],-1, MONTH))



Resulting table contains rows
which appear in BOTH tables

IMPORTANT NOTES:

- ✓ Order matters! (T_1, T_2) may have a different result set than (T_2, T_1)
- ✓ Columns are compared based on positioning in their respective tables
- ✓ Duplicate rows are retained
- ✓ Column names are determined by the left table
- ✓ The resulting table does NOT retain relationships to other tables (*can't be used as an expanded table*)

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

Name New Employees (INTERSECT)

Mark as date table Manage relationships New measure Quick New measure column New table

Measure Calculations Relationships Calculations

1 New Employees (INTERSECT) =
2 INTERSECT(
3 'Employee Lookup',
4 FILTER(
5 'Employee Lookup',
6 'Employee Lookup'[start_date] > DATE(2016,12,31)
7)
8)

Fields

Search

Measure Table
ADDCOLUMNS Demo
Calendar
CROSSJOIN Demo
Customer Lookup
DATATABLE Demo
Employee Lookup
EXCEPT Demo
Food Inventory
GENERATESERIES Demo
New Employees (INTERSECT)
Product Lookup
Row Demo
Sales by Store
SELECTEDCOLUMN1 Demo
SELECTEDCOLUMN3 Demo 2
Store Lookup
Table Constructor Demo
Target Sales - April 2019 (DATATABLE)
Target Sales - March 2019 (DATATABLE)
UNION Demo
X - Bean/Tees Goal
X - Beverage Goal
X - Food Goal
X - Merchandise Goal
Month
X - Store ID
Year
X - Unsold Pastries

1) Create a New Table call it New Employees (INTERSECT). Gives us an understanding of what we can do to use INTERSECT to combine two tables.

staff_id	first_name	last_name	position	start_date	location
21	Melodie	Mercedes	Store Manager	28/08/2018 00:00:00	S
22	Bythe	Arsenio	Coffee Wrangler	22/11/2018 00:00:00	S
23	Aime	Melanie	Coffee Wrangler	14/09/2017 00:00:00	S
47	Pendors	Neville	Coffee Wrangler	21/08/2019 00:00:00	S

Table: New Employees (INTERSECT) (4 rows)

Maven Routers - Analysis - Power BI Desktop

File Home Help Table tools

New Employee (N...)

Mark as date table ✓ Manage relationships New measure Quick New measure column New table

Measure Calculations

1 New Employees (INTERSECT) =
2 INTERSECT(
3 ADDCOLUMNS(
4 'Employee Lookup',
5 "Revenue",
6 [Customer Sales]
7),
8 ADDCOLUMNS(
9 FILTER(
10 'Employee Lookup',
11 'Employee Lookup'[start_date] > DATE(2016,12,31)
12),
13 "Revenue",
14 [Customer Sales]
15)
16)

Fields

Search

Measure Table
ADDCOLUMNS Demo
Calendar
CROSSJOIN Demo
Customer Lookup
DATATABLE Demo
Employee Lookup
EXCEPT Demo
Food Inventory
GENERATESERIES Demo
New Employees (INTERSECT)
Product Lookup
Row Demo
Sales by Store
SELECTEDCOLUMN3 Demo
SELECTEDCOLUMN3 Demo 2
Store Lookup
Table Constructor Demo
Target Sales - April 2019 (DATATABLE)
Target Sales - March 2019 (DATATABLE)
UNION Demo
Bean/Tees Goal
Beverage Goal
Food Goal
Merchandise Goal
Month
Store ID
Year
Unsold Pastries

staff_id first_name last_name position start_date location Revenue

22 Mercedes Mercedes Store Manager 29/09/2012 00:00:00 8
23 Blythe Amanio Coffee Wrangler 22/12/2012 00:00:00 8 20117.8100000001
24 Alina Melania Coffee Wrangler 14/04/2017 00:00:00 8
43 Pandora Neville Coffee Wrangler 22/05/2019 00:00:00 19 295658.49

1) Example of INTERSECT with the ADDCOLUMNS function to add in the Revenue column using the customer sales expression.

ASSIGNMENT Calculated Table Joins

KEY OBJECTIVES

- 1) Use CALCULATETABLE & INTERSECT to understand returning customers between **11/18/2018** & **12/1/2018**.
- 2) Add additional columns for Revenue & Profit.
- 3) Determine the total Revenue & Profit for repeat customers on week 46.

Maven Router - Analysis - Power BI Desktop

File Home Help Table tools Column tools

Name: customer_id Format: Whole number Data type: Whole number Summarization: Sum Data category: Uncategorized Sort by column: Sort Data groups: Data groups Manage relationships: Relationships New column: Calculations

1 Repeat Customer Sales (INTERSECT Assignment) =
2 VAR PurchasedTwoWeekPeriod =
3 CALCUTABLE(
4 VALUES(
5 'Sales by Store'[customer_id]
6),
7 'Calendar'[Week_ID] IN {45,46},
8 'Calendar'[Year_ID] = 2018
9)
10 VAR PurchasedOneWeekPeriod =
11 CALCUTABLE(
12 VALUES(
13 'Sales by Store'[customer_id]
14),
15 'Calendar'[Week_ID] = {46},
16 'Calendar'[Year_ID] = 2018
17)
18 VAR ReturningCustomers =
19 INTERSECT(
20 PurchasedTwoWeekPeriod,
21 PurchasedOneWeekPeriod
22)
23 RETURN
24 ReturningCustomers

customer_id

1536

1) Objective 1 to use CALCUTABLE & INTERSECT to understand returning customers between 11/18/2018 & 12/1/2018. Create a New Table call it Repeat Customer Sales (INTERSECT Assignment).

2) Key objective one met here 1536 distinct customer id's this represents the number of customers that are the same between these two groups i.e. PurchasedTwoWeekPeriod & PurchasedOneWeekPeriod.

Fields

Search

Measure Table ADDCOLUMNS Demo Calendar CROSSTAB Demo Customer Lookup DATABLE Demo Employee Lookup Food Inventory GENERATESERIES Demo Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store SELECTEDCOLUMNS Demo SELECTEDCOLUMNS Demo 2 Store Lookup Table Constructor Demo Target Sales - April 2019 (DATABLE) Target Sales - March 2019 (DATABLE) UNION Demo Unsold Products VALUES Example

Key objective 1 is now complete. Use CALCUTABLE & INTERSECT to understand returning customers between 11/18/2018 & 12/1/2018.

Maven Router - Analysis - Power BI Desktop

Sign in

File Home Help Table tools Column tools

Name: customer_id Format: Whole number \$ - % , . 0 Summarization: Sum Data category: Uncategorized Sort by column: Sort Data groups: Group Manage relationships: Relationships New column: Calculations

Key objective 2 is now complete. Add additional columns for Revenue & Profit.

Fields

Search

1. Repeat Customer Sales (INTERSECT Assignment) *

2. VAR PurchasedTwoWeekPeriod =

3. CALCULATETABLE(

4. ADDCOLUMNS(

5. VALUES(

6. 'Sales by Store'[customer_id]

7.),

8. "Revenue",

9. [Customer Sales],

10. "Profit",

11. [Profit]

12.),

13. 'Calendar'[Week_ID] IN {45,46},

14. 'Calendar'[Year_ID] = 2018

15.)

16. VAR PurchasedOneWeekPeriod =

17. CALCULATETABLE(

18. ADDCOLUMNS(

19. VALUES(

20. 'Sales by Store'[customer_id]

21.),

22. "Revenue",

23. [Customer Sales],

24. "Profit",

25. [Profit]

26.),

27. 'Calendar'[Week_ID] = {46},

28. 'Calendar'[Year_ID] = 2018

29.)

30. VAR ReturningCustomers =

31. INTERSECT(

32. PurchasedTwoWeekPeriod,

33. PurchasedOneWeekPeriod

34.)

35. RETURN

36. ReturningCustomers

customer_id Revenue Profit

5542 8.3 6.25

3212 10 7.5

3394 8 4.5

4293 3 2.75

5891 7 2.25

5892 4 1

... 4 0.5

Table: Repeat Customer Sales (INTERSECT Assignment) (178 rows) Columns: customer_id (178 distinct values)

3) Objective 2 to add additional columns for Revenue & Profit. Second objective between two sets we only have 178 customers that have actually purchased something in our second week they've returned & purchased something within that second week.

Maven Routers - Analysis - Power BI Desktop

File Home Insert Modeling View Help Table tools Measure tools

Name Profit (INTERSECT Assignment) Format Currency Data category Uncategorized

Home table Measure Table \$ - % .00 2 New Quick measure measure Properties Calculations

1 Profit (INTERSECT Assignment) =
2 SUM(
3 'Repeat Customer Sales (INTERSECT Assignment)'[Profit]
4)

2) Change the formatting accordingly over here.

Build visuals with your data
Select or drag fields from the Fields pane onto the report canvas.

1) Objective 3 to determine the total Revenue & Profit for repeat customers on week 46. The best way to do this is to create a measure go over to the report view & create the "Profit (INTERSECT Assignment)" measure.

Visualizations Fields

Search

Measure Table

- % of All Baked
- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLINCLUDEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales BY (COALESCE)
- Customer Sales BY (ISBLANK)
- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)
- Lost Revenue (SUMMARIZE Table)
- Orders by Females (Wrong)
- Profit (INTERSECT Assignment) ■
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Revenue (INTERSECT Assignment)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (ALL)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 5 Profit
- Store 5 Profit (KEEPFILTERS)
- Store 5 Sales (KEEPFILTERS)

Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED ASSIGNMENT

Page 11 of 11

File Home Insert Modeling View Help Table tools Measure tools

Name Revenue (INTERSE...
Home table Measure Table

Format Currency \$ - % .0 2 Data category Uncategorized New measure Quick measure measure Properties Calculations

2) Change the formatting accordingly over here.

1 Revenue (INTERSECT Assignment) =
2 SUM(
3 'Repeat Customer Sales (INTERSECT Assignment)' [Revenue]
4)

Build visuals with your data
Select or drag fields from the Fields pane onto the report canvas.

1) Third key objective part two create the “Revenue (INTERSECT Assignment)” measure.

Visualizations > Fields

Search

- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales BY (COALESCE)
- Customer Sales BY (SBLANK)
- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)
- Last Revenue (SUMMARIZE Table)
- Orders by Females (Wrong)
- Profit (INTERSECT Assignment)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Revenue (INTERSECT Assignment)

Scalar Measures

- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales of whole Beans & Teas (All)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)
- Store 4 Sales (KEEPFILTERS)

Tip & Best Practices

- Total Baked
- Total Profit
- Total Sold

ADOCOLUMNS Demo

- first_name
- last_name
- location
- Manager Name & Store

Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED ASSIGNMENT

Page 11 of 11

Maven Routers - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals

New measure measure Quick Publish

Dataset Data Queries Insert Calculations Share

Key objective 3 is now complete. Determine the total Revenue & Profit for repeat customers on week 46.

Visualizations Fields

Search

Measure Table

- % of All Baked
- % of Store-Level Sales
- % of Total Baked (ALLSELECTED)
- % of Total Sold (ALLSELECTED)
- % Quantity Sold to Females
- % Store Sales (REMMOVEDFILTERS)
- Cost (CURRENCY)
- Count of Product ID (DISTINCT)
- Count of Product ID (VALUES)
- Customer Sales (ALLEXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)
- Customer Sales (Last Year)
- Customer Sales (V COALESCE)
- Customer Sales (V ISBLANK)
- Food Sold (SUMMARIZE Table)
- Food Unsold (SUMMARIZE Table)
- Last Revenue (SUMMARIZE Table)
- Orders by Females (Wrong)
- Profit (INTERSECT Assignment)
- Quantity Sold (SELECTEDVALUE)
- Retail Price (SELECTEDVALUE)
- Revenue (INTERSECT Assignment)

Scalar Measures

- Store 3 Sales (K) (PPLTERS)
- Store 3 Sales of whole Beans & Teas (ALL Modifier)
- Store 3 Sales of whole Beans & Teas (CALCULATE)
- Store 3 Profit
- Store 3 Profit (KEEPFILTERS)
- Store 3 Sales (KEEPFILTERS)

Keep Remove Filters Table & Filter Functions Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED AS +

3) Within the matrix above add the below measure in.
a) Revenue (INTERSECT Assignment)
b) Profit (INTERSECT Assignment)

Revenue (INTERSECT Assignment)	Profit (INTERSECT Assignment)
\$1,817.68	\$1,203.31

8) Relationship Functions

Introduction - Relationship Functions



Relationship functions allow you to access fields within DAX measures or calculated columns through either *physical* or *virtual* relationships between tables

TOPICS WE'LL COVER:

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

COMMON USE CASES:

- Defining calculated columns or measures using fields from related tables
- Handling relationships between calendars and multiple date fields (transaction date, stock date, due date, etc.)
- Traversing inactive relationships in the model, or modifying relationship filter behavior
- Defining calculated, virtual relationships when physical relationships aren't possible due to table granularity

Physical vs. Virtual Relationships

There are two key types of table relationships: **PHYSICAL** and **VIRTUAL**

- **Physical** relationships are manually created, and visible in your data model
- **Virtual** relationships are temporary, and defined using DAX expressions



These are **physical** relationships:

- *Visible links between tables (typically 1:** cardinality)
- *Can be active or inactive*
- *Can be accessed using DAX functions like RELATED, RELATEDTABLE or USERELATIONSHIP*
- *Best way to connect tables (but not always possible)*

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

```
1 Bean Goal (TREATAS) =  
2 CALCULATE(  
3     SUM(  
4         'Target Sales UNION Example'[Bean/Teas Goal]  
5     ),  
6     TREATAS(  
7         VALUES(  
8             'Calendar'[Year_ID]  
9         ),  
10        'Target Sales UNION Example'[Year]  
11    ),  
12    TREATAS(  
13        VALUES(  
14            'Calendar'[Month_Name]  
15        ),  
16        'Target Sales UNION Example'[Month]  
17    )  
18 )
```

This is a **virtual relationship**:

- *Defined using DAX expressions*
- *Used when a physical relationship doesn't exist, or cannot be created directly*
- *Often used to connect tables with different levels of granularity (i.e. daily sales vs. monthly budgets/goals)*
- *Can be accessed using DAX functions like TREATAS*

RELATED

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

RELATED()

Returns a value from a related table in the data model

=RELATED(Column**Name**)



Name of the column you want to retrieve values from (must reference a table on the "one" side of a many-to-one relationship)

Examples:

- 'Product Lookup'[current_cost]
- 'Customer Lookup'[home_store]



PRO TIP

The RELATED function doesn't really perform an operation, it just "opens the door" to access columns from an expanded table

Maven Roasters - Analysis - Power BI Desktop

File Home Help External Tools Table tools Column tools

Name Product Group Data type Text Summarization Don't summarize Data category Uncategorized Sort by column Data groups Manage relationships New column

Structure Properties Sort Group Relationships Calculations

1 Product Group =
2 RELATED()
3 'Product Lookup'[product_group]
4)

transaction_id transaction_date store_id staff_id customer_id instance_yo order line_item_id product_id quantity_sold unit_price printed_item_yo Product Group

181 19 January 2017 06:45:47 8 42 8600 Y 1 82 1 3 N Beverages
182 21 January 2017 09:37:34 8 42 8600 Y 1 82 1 3 N Beverages
460 28 January 2017 09:29:07 8 42 8600 Y 1 82 1 3 N Beverages
21 42 January 2017 07:25:43 8 42 8600 Y 1 82 1 3 N Beverages
181 22 February 2017 06:45:47 8 42 8600 Y 1 82 1 3 N Beverages
21 22 February 2017 07:23:43 8 42 8600 Y 1 82 1 3 N Beverages
373 23 February 2017 08:58:59 8 42 8600 Y 1 82 1 3 N Beverages
21 12 March 2017 07:53:45 8 42 8600 Y 1 82 1 3 N Beverages
211 14 March 2017 09:12:37 8 42 8600 Y 1 82 1 3 N Beverages
728 17 March 2017 09:05:44 8 42 8600 Y 1 82 1 3 N Beverages
373 19 March 2017 08:58:59 8 42 8600 Y 1 82 1 3 N Beverages
290 21 March 2017 11:26:39 8 42 8600 Y 1 82 1 3 N Beverages
24 26 March 2017 06:34:09 8 42 8600 Y 1 82 1 3 N Beverages
460 28 March 2017 09:28:07 8 42 8600 Y 1 82 1 3 N Beverages
277 31 March 2017 08:29:16 8 42 8600 Y 1 82 1 3 N Beverages
181 09 April 2017 09:40:46 8 42 8600 Y 1 82 1 3 N Beverages
728 17 April 2017 09:05:44 8 42 8600 Y 1 82 1 3 N Beverages
292 21 April 2017 09:37:34 8 42 8600 Y 1 82 1 3 N Beverages
728 22 April 2017 06:47:33 8 42 8600 Y 1 82 1 3 N Beverages
26 26 April 2017 06:34:09 8 42 8600 Y 1 82 1 3 N Beverages
211 30 April 2017 09:17:27 8 42 8600 Y 1 82 1 3 N Beverages
181 30 May 2017 06:45:47 8 42 8600 Y 1 82 1 3 N Beverages
21 12 May 2017 07:23:43 8 42 8600 Y 1 82 1 3 N Beverages
2660 16 May 2017 06:58:10 8 42 8600 Y 1 82 1 3 N Beverages
1320 16 May 2017 08:17:38 8 42 8600 Y 1 82 1 3 N Beverages
728 17 May 2017 09:05:44 8 42 8600 Y 1 82 1 3 N Beverages
373 19 May 2017 08:58:59 8 42 8600 Y 1 82 1 3 N Beverages
180 23 May 2017 11:26:39 8 42 8600 Y 1 82 1 3 N Beverages
277 28 May 2017 08:39:16 8 42 8600 Y 1 82 1 3 N Beverages

Fields

Search

Measure Table ADDCOLUMNS Demo Calendar CROSSJOIN Demo Customer Lookup DAXTABLE Demo Employee Lookup Food Inventory GENDIMENSIONS Demo Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store

(Select All) Add-ons Beverages Food Merchandise Whole Bean Teas

OK Cancel

1) Add the Product Group column to the Sales by Store table using this code.

2) Product Groups now available in the Sales by Store table.

Maven Roastem - Analysis - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Wholesale Cost

Home table Measure Table

Format Currency

Data category Uncategorized

New Quick measure measure

Properties Calculations

1 Wholesale Cost =
2 SUMX(
3 'Sales by Store',
4 'Sales by Store'[quantity_sold] *
5 RELATED(
6 'Product Lookup'[current_wholesale_price]
7)
8)

Visualizations >

Fields >

Search

% Quantity sold to females
% Store Sales (REMOVEDFILTERS)
Cost (CURRENCY)
Count of Product ID (DISTINCT)
Count of Product ID (VALUES)
Customer Sales (ALLEXCEPT Assignment)
Customer Sales (ALLSELECTED)
Customer Sales (Last Year)
Customer Sales LY (COALESCE)
Customer Sales LY (ISBLANK)
Food Sold (SUMMARIZE Table)
Food Unsold (SUMMARIZE Table)
Lost Revenue (SUMMARIZE Table)
Orders by Females (Wrong)
Profit (INTERSECT Assignment)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Revenue (INTERSECT Assignment)

Scalar Measures

Store 3 Sales (KEEPFILTERS)
Store 3 Sales of whole Beans & Tea (ALL...
Store 3 Sales of whole Beans & Tea (CAL...
Store 3 Profit
Store 3 Profit (KEEPFILTERS)
Store 3 Sales (KEEPFILTERS)
Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

Total Baked
Total Profit
Total Sold
Wholesale Cost

2) Change the formatting accordingly over here.

Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED ASSIGNMENT SUMMARIZE Row Relationship Functions +

Page 12 of 12 Update available (click to download)

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Out data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

Queries Calculations Share

Over \$2.2m whole cost highlighted in the matrix.

Wholesale Cost
\$2,285,208.27

Visualizations Fields

Search

Customer Sales (ALLSELECTED)
Customer Sales (Last Year)
Customer Sales LV (COALESCE)
Customer Sales LV (CBLANK)
Food Sold (SUMMARIZE Table)
Food Unsold (SUMMARIZE Table)
Lost Revenue (SUMMARIZE Table)
Orders by Females (Wrong)
Profit (INTERSECT Assignment)
Quantity Sold (SELECTEDVALUE)
Retail Price (SELECTEDVALUE)
Revenue (INTERSECT Assignment)

Scalar Measures

Store 3 Sales (KEEPFILTERS)
Store 3 Sales of whole Beans & Teas (ALL ...
Store 3 Sales of whole Beans & Teas (CAL...
Store 5 Profit
Store 5 Profit (KEEPFILTERS)
Store 5 Sales (KEEPFILTERS)
Store 8 Sales (KEEPFILTERS)

Tip & Best Practices

Total Baked
Total Profit
Total Sold

Whole Cost

ADOCOLUMNS Demo

first_name
last_name
location
Manager Name & Store
position

Table & Filter Functions - 2 Table & Filter Functions - 3 Table & Filter Functions - 4 ALLSELECTED ASSIGNMENT ALLSELECTED ASSIGNMENT SUMMARIZE Row Relationship Functions +

Page 12 of 12 Update available (click to download)

RELATEDTABLE

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

RELATEDTABLE()

Returns a related table, filtered so that it only includes the related rows

=RELATEDTABLE(Table)



Physical table you want to retrieve rows from (must reference a table on the “many” side of a many-to-one relationship)

NOTE: RELATEDTABLE is commonly used with aggregators like COUNTROWS, SUMX, AVERAGEX, etc.

Examples:

- COUNTROWS(
 RELATEDTABLE('Food Inventory'))
- SUMX(
 RELATEDTABLE('Food Inventory'),
 [Quantity Sold] * [Retail Price])



HEY THIS IS IMPORTANT!

RELATEDTABLE is a shortcut for CALCUTATABLE (with no logical expression) and performs a **context transition** from row context to filter context, in order to return only the rows which satisfy the filter condition(s)

Maven Roasters - Analysis - Power BI Desktop

Sign in

File Home Help External Tools Table tools Column tools

Name: Number of Food It... Format: Whole number Summarization: Sum Data category: Uncategorized Sort by column: Sort Data groups: Group Manage relationships: Relationships New column Calculations:

1 Number of Food Items Made =

2 SUMX(

3 RELATEDTABLE(

4 'Food Inventory'

5),

6 'Food Inventory'[quantity_start_of_day]

7)

1) Add a new column called Number of Food Items Made within the Product Lookup table. To calculate this aggregation we will use SUMX.

2) We have returned the quantity we had made or baked for each one of these food items using RELATEDTABLE function.

Fields

Search

- Measure Table
- ADDCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- DATATABLE Demo
- Employee Lookup
- Food Inventory
- GENERATESERIES Demo
- Product Lookup

current_cost

current_retail_price

Sort ascending

Sort descending

Clear sort

Clear filter

Clear all filters

Number filters

- (Selected all)
- (Blank)
- 44653
- 45193
- 45360
- 45630
- 120480

OK Cancel

item_type	product	product_description	unit_of_measurement	current_cost	current_wholesale_price	current_retail_price	tax_exempt_ye	gross_ye	new_product_ye	Number of Food Items Made
rc Beans	Brazilian - Organic	It's like Carnival in a cup. Clean and smooth.	12 oz	1.8	14.4	18	N	N	N	
rb Blend Beans	Our Old Time Diner Blend	Our packed blend of beans that is reminiscent of the cup.	12 oz	1.8	14.4	18	N	N	N	
iso Beans	Espresso Roast	Our house blend for a good espresso shot.	1 lb	2.85	12.8	16.75	N	N	N	
iso Beans	Primo Espresso Roast	Our premium single source of hand roasted beans.	1 lb	4.09	18.48	22.45	N	N	N	
met Beans	Columbian Medium Roast	A smooth cup of coffee any time of day.	1 lb	3	12	15	N	N	N	
met Beans	Ethiopia	From the home of coffee.	1 lb	4.2	18.8	22	N	N	N	
rum Beans	Jamaican Coffee River	Ya man, it will start your day off right.	1 lb	3.85	15.8	18.75	N	N	N	
rum Beans	Civet Cat	The most expensive coffee in the world, the cat do all the work.	5 lb	9	36	45	N	N	N	
rc Beans	Organic Decaf Blend	Our blend of hand picked organic beans that have been decaf.	1 lb	4.5	18	22.5	N	N	N	
l Beans	Guatemalan Sustainably Grown	Green beans you can roast yourself.	1 lb	2	8	10	N	N	N	
tl tea	Lemon Grass	You will think you are Thailand as you sit your cuppa.	9 oz	0.6	7.14	8.28	N	N	N	
tl tea	Peppermint	Cool and refreshing to help calm your nerves.	9 oz	0.6	7.14	8.28	N	N	N	
te	English Breakfast	The traditional cup to start your day.	9 oz	0.6	7.14	8.28	N	N	N	
te	Earl Grey	A full leaf of Orange Pekoe blended with organic oil of bergamot.	9 oz	0.6	7.14	8.28	N	N	N	
tl tea	Serenity Green Tea	Mountain grown and harvested at the optimal time.	1 oz	0.3	7.4	9.28	N	N	N	
tl tea	Traditional Blend Chai	A traditional blend.	9 oz	0.6	7.14	8.28	N	N	N	
tl tea	Morning Sunrise Chai	Fair trade and organic and has a warm finish.	9 oz	0.6	7.14	8.28	N	N	N	
tl tea	Spicy Eye Opener Chai	A spicier blend to awaken your taste buds.	9 oz	0.6	8.76	10.88	N	N	N	
ing Chocolate	Dark chocolate	This drinking chocolate is smooth and creamy.	1 lb	2.28	5.12	6.4	N	N	N	
rc Chocolate	Sustainably Grown Organic	Certified organic containing the highest quality ingredients.	1 lb	2.32	6.08	7.6	N	N	N	

Table: Product Lookup (87 rows) Column: Number of Food Items Made (5 distinct value)

Update available (click to download)

KEY OBJECTIVES

- 1) Add a calculated column to the Customer Lookup table & create a variable for "Total Sales".
- 2) Add another variable called "AllCustomers" to count total customers.
- 3) Add another variable that defines average sales.
- 4) Use RELATEDTABLE to create a variable that computes sales at the customer level.
- 5) Return "High" if customer-level sales are above the average.
Otherwise, return "Low".

Maven Roastem - Analysis - Power BI Desktop

Sign in

Key objective 1 is now complete. Add a calculated column to the Customer Lookup table & create a variable for "Total Sales".

1 Customer Sales (High/Low) =

2 VAR TotalSales = 

3 SUMX(

4 'Sales by Store',

5 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]

6)

1) Objective 1 to add a calculated column to the Customer Lookup table & create a variable for "Total Sales". Head over to the customer lookup table add a new column called Customer Sales (High/Low). Create the first variable called TotalSales.

customer_id	home_store	customer_first_name	customer_email	customer_phone	customer_street	loyalty_card_number	birthdate	gender	birth_year	Column
301	3	Alice Rivers	Yoshi@aliquam.us	04-January-2017 021-443-0642	13-May-1950	F	1950			
302	3	Bachir Wall	Yvette@erat.org	08-January-2017 439-302-9777	29-June-1950	F	1950			
303	3	Reya Hampton	Martina@metaculus.com	08-January-2017 478-762-8984	14-August-1950	F	1950			
304	3	Belle Reyes	Sierre@sodales.net	23-January-2017 930-000-5889	09-September-1950	F	1950			
305	3	Brooke Munoz	William@montes.com	22-January-2017 575-552-0754	26-November-1950	F	1950			
306	3	Roxana Wolf	Clinton@varius.edu	24-January-2017 349-238-7140	01-January-1951	F	1951			
307	3	Quintessa Franklin	Asdney@mattamus.net	18-January-2017 877-311-3335	17-February-1951	F	1951			
308	3	Charrissa Cobb	Ragan@vivamus.net	18-January-2017 818-483-1438	04-April-1951	F	1951			
309	3	Susan Morris	Cody@thincidunt.edu	20-January-2017 726-654-4338	21-May-1951	F	1951			
310	3	Cynthia Murley	Miniam@arcu.edu	22-January-2017 305-573-4584	07-July-1951	F	1951			
311	3	Melodie Dawson	Stacy@indicta.us	24-January-2017 894-557-6183	22-August-1951	F	1951			
312	3	Charlotte Webb	Phelan@nutrum.org	26-January-2017 540-155-7142	08-October-1951	F	1951			
313	3	Malyssa Jones	Dellan@duis.edu	28-January-2017 029-651-0904	23-November-1951	F	1951			
314	3	Yuri Jacobson	Shea@ante.gov	30-January-2017 871-921-0474	09-January-1952	F	1952			
315	3	Keller Barton	Hudson@lurum.edu	02-February-2017 236-217-0354	25-February-1952	F	1952			
316	3	Caroline Glass	Hedley@adipiscing.gov	04-February-2017 448-528-0188	11-April-1952	F	1952			
317	3	Reagan Roberts	Echo@integer.edu	06-February-2017 324-062-4587	28-May-1952	F	1952			
318	3	Chelsea Gray	Willow@veni.net	08-February-2017 268-181-8624	13-July-1952	F	1952			
319	3	Lesley Benson	Laura@efficit.net	10-February-2017 683-365-3130	29-August-1952	F	1952			
320	3	Devon Matthews	Hannah@efficit.org	12-February-2017 212-982-5326	15-October-1952	F	1952			
321	3	Lillian Lucas	Uria@serena.com	14-February-2017 945-752-7433	20-November-1952	F	1952			
322	3	Brittany Bradshaw	Rina@duc.us	16-February-2017 827-758-6053	16-January-1953	F	1953			
323	3	Cecilia Davidson	Donovan@posuere.org	18-February-2017 764-621-4399	03-March-1953	F	1953			
324	3	Fiona Connersmith	Vernon@laurum.org	20-February-2017 491-145-8184	19-April-1953	F	1953			

Fields

Search

- Measure Table
- ADDCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- birth_year
- birthdate
- Column
- customer_email
- customer_first_name
- customer_id
- customer_since
- gender
- home_store
- loyalty_card_number
- DATATABLES Demo
- Employee Lookup
- Food Inventory
- GENERATEVALUES Demo
- Product Lookup
- Repeat Customer Sales (INTERSECT Assignment)
- Row Demo
- Sales by Store
- SELECTEDCOLUMNS Demo
- SELECTEDCOLUMNS Demo 2
- Store Lookup
- Table Constructor Demo
- Target Sales - April 2019 (DATATABLE)
- Target Sales - March 2019 (DATATABLE)

Maven Roosters - Analysis - Power BI Desktop

Sign in

Key objective 2 is now complete. Add another variable called "AllCustomers" to count total customers.

1 Customer Sales (High/Low) =

2 VAR TotalSales =

3 SUMX(

4 'Sales by Store',

5 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]

6)

7 VAR AllCustomers = ← I

8 COUNTROWS('Customer Lookup')

9

1) Objective 2 to add another variable called "AllCustomers" to count total customers.

customer_id	name_first	customer_email	customer_since	loyalty_card_number	birthdate	gender	birth_year	Column
301	Alice Rivers	Yoshi@outlook.us	04 January 2017	521-443-0982	13 May 1970	F	1990	
302	Sasha Wall	Yvette@ent.org	08 January 2017	433-322-9277	29 June 1990	F	1990	
303	Raya Hampton	Martina@microsoft.com	08 January 2017	478-782-8984	24 August 1980	F	1990	
304	Belle Hayes	Birrie@sociales.net	10 January 2017	810-000-5489	20 September 1980	F	1990	
305	Brooke Munroe	William@monroes.com	12 January 2017	575-552-0754	18 November 1980	F	1990	
306	Tanisha Wolff	Clinton@varius.edu	14 January 2017	369-136-7140	01 January 1981	F	1991	
307	Quintessa Franklin	Audrey@mathis.net	15 January 2017	877-581-8885	17 February 1981	F	1991	
308	Charissa Coop	Regan@vivamus.net	18 January 2017	818-483-1626	04 April 1981	F	1991	
309	Susan Matthes	Cody@indubius.edu	20 January 2017	756-654-0238	21 May 1981	F	1991	
310	Cynthia Murray	Miriam@actu.edu	22 January 2017	809-173-4564	07 July 1981	F	1991	
311	Merodie Dawson	Stacey@notora.us	24 January 2017	894-557-8183	23 August 1981	F	1991	
312	Charlotte Webb	Phelan@nutrum.org	26 January 2017	542-155-7242	08 October 1981	F	1991	
313	Malyssa Jones	Dellish@out.edu	28 January 2017	029-831-0108	23 November 1981	F	1991	
314	Huri Jacobson	Shea@ante.gov	30 January 2017	371-921-0474	09 January 1982	F	1992	
315	Katia Barton	Hudson@uma.edu	02 February 2017	238-217-0156	23 February 1982	F	1992	
316	Caroline Glass	Hedley@advancing.gov	04 February 2017	448-528-0198	22 April 1982	F	1992	

Table: Customer Lookup (2,251 rows) Column: Column (1 distinct values)

Update available (click to download)

Maven Reactors - Analysis - Power BI Desktop

Sign in

Key objective 3 is now complete. Add another variable that defines average sales.

Fields

Search

- Measure Table
- ADOCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- birth_year
- birthdate
- Column
- customer_email
- customer_first_name
- customer_id
- customer_since
- gender
- home_store
- loyalty_card_number
- DATATABLE Demo
- Employee Lookup
- Food Inventory
- GENERATESERIES Demo
- Product Lookup
- Repeat Customer Sales (INTERSECT Assignment)
- Row Demo
- Sales by Store
- SELECTEDCOLUMNS Demo
- SELECTEDCOLUMNS Demo 2
- Store Lookup
- Table Constructor Demo
- Target Sales - April 2019 (DATATABLE)
- Target Sales - March 2019 (DATATABLE)

File Home Help External Tools Table tools Column tools

Name: Column Format: \$ - % Summarization: Don't summarize Data category: Uncategorized Sort by column: Sort Data groups: Groups Manage relationships: Relationships New column: Calculations

1 Customer Sales (High/Low) =
 2 VAR TotalSales =
 3 SUMX(
 4 'Sales by Store',
 5 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
 6)
 7 VAR AllCustomers =
 8 COUNTROWS('Customer Lookup')
 9 VAR AvgSales =
 10 DIVIDE(TotalSales,AllCustomers)
 11

customer_id home_store customer_first-name customer_email customer_since loyalty_card_number birthdate gender birth_year Column
 301 J. Alba Flores fofo@aliquem.us 04-January-2017 021-443-0682 13-May-1980 F 1980
 302 J. Sacha Wall twtts@hot.org 06-January-2017 413-102-9277 29-June-1980 F 1980
 303 J. Raye Hampton Marmon@marleysuds.com 08-January-2017 478-762-8984 14-August-1980 F 1980
 304 J. Belle Rivers Stern@socialsales.net 10-January-2017 830-000-5559 10-September-1980 F 1980
 305 J. Brooke Munoz William@monicas.com 12-January-2017 573-052-0754 18-November-1980 F 1980
 306 J. Tamisha Wolff Clinton@overdue.edu 14-January-2017 389-256-7140 01-January-1981 F 1981
 307 J. Quintessa Franklin Audrey@mattress.net 16-January-2017 877-781-9855 17-February-1981 F 1981
 308 J. Charissa Coble Regan@vivamus.net 18-January-2017 618-482-0104 04-April-1981 F 1981
 309 J. Sutton Mathis Cody@thinkdunk.edu 20-January-2017 726-654-4316 22-May-1981 F 1981
 310 J. Cynthia Hurley Miriam@entu.edu 22-January-2017 309-573-4564 07-July-1981 F 1981
 311 J. Madeline Dawson Stacey@nostra.us 24-January-2017 684-557-8381 22-August-1981 F 1981

Table: Customer Lookup (2,251 rows) Column: Column (1 distinct values)

Update available (click to download)

Maven Reactors - Analysis - Power BI Desktop

File Home Help External Tools Table tools Column tools

Name: Column Format: \$ - % Summarization: Don't summarize Data category: Uncategorized Sort by column: Sort Data groups: Groups Manage relationships: Relationships New column Calculations

Key objective 4 is now complete. Use RELATEDTABLE to create a variable that computes sales at the customer level.

```

1 Customer Sales (High/Low) =
2 VAR TotalSales =
3 SUMX(
4     'Sales by Store',
5     'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
6 )
7 VAR AllCustomers =
8 COUNTROWS('Customer Lookup')
9 VAR AvgSales =
10 DIVIDE(TotalSales,AllCustomers)
11 VAR CustomerSales ←
12 SUMX(
13     RELATEDTABLE(
14         'Sales by Store'
15     ),
16     'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
17 )

```

Fields

Search

- Measure Table
- ADOCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
 - birth_date
 - birthdate
- Column
 - customer_email
 - customer_first_name
 - customer_id
 - customer_since
 - gender
 - home_store
 - loyalty_card_number
- DATATABLE Demo
- Employee Lookup
- Food Inventory
- GENERATESERIES Demo
- Product Lookup
- Repeat Customer Sales (INTERSECT Assignment)
- Row Demo
- Sales by Store
- SELECTDUNIQS Demo
- SELECTEDCOLUMNS Demo 2
- Store Lookup
- Table Constructor Demo
- Target Sales - April 2019 (DATATABLE)
- Target Sales - March 2019 (DATATABLE)

Customer ID Name_Short customer_firstname customer_email customer_since loyalty_card_number Birthdate gender birth_year Column

Customer ID	Name_Short	Customer Firstname	Customer Email	Customer Since	Loyalty Card Number	Birthdate	Gender	Birth Year	Column
101	J. Alice Rivers	Yvonne@angrymen.us	04 January 2017	023-443-0462	17 May 1950	F	1950		
102	J. Sacha Wall	Yvette@erat.org	06 January 2017	433-102-9777	29 June 1950	F	1950		
103	J. Baya Hampton	Martina@maleuada.com	08 January 2017	478-762-8884	14 August 1950	F	1950		
104	J. Boeby Reyes	Nicola@codicil.net	10 January 2017	930-000-5689	02 September 1950	F	1950		
105	J. Brooke Mundz	William@montes.com	12 January 2017	573-551-0754	16 November 1950	F	1950		
106	J. Tenisha Wolf	Clinton@verius.edu	14 January 2017	368-234-7140	01 January 1951	F	1951		
107	J. Quintessa Franklin	Audrey@mattt.net	16 January 2017	877-781-9899	27 February 1951	F	1951		
...

Table: Customer Lookup (2,231 rows) Column: Column (1 distinct values)

Update available (click to download)

Maven Roasters - Analysis - Power BI Desktop

File Home Help External Tools Table tools Column tools

Name: Customer Sales (High/Low) Data type: Text Summarization: Don't summarize Sort by column: Data category: Uncategorized Data groups: Manage relationships New column

Structure Filtering Properties

1 Customer Sales (High/Low) =
 2 VAR TotalSales =
 3 SUMX(
 4 'Sales by Store',
 5 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
 6)
 7 VAR AllCustomers =
 8 COUNTROWS('Customer Lookup')
 9 VAR AvgSales =
 10 DIVIDE(TotalSales,AllCustomers)
 11 VAR CustomerSales =
 12 SUMX(
 13 RELATEDTABLE(
 14 'Sales by Store'
 15),
 16 'Sales by Store'[quantity_sold] * 'Sales by Store'[unit_price]
 17)
 18 VAR HighLow = ←
 19 IF(
 20 CustomerSales > AvgSales,
 21 "High",
 22 "Low"
 23)
 24 RETURN
 25 HighLow

Fields

Measure Table ADDCOLUMNS Demo Calendar CROSSJOIN Demo Customer Lookup birth_year birthdate Customer Sales (High/Low) customer_email customer_first_name customer_id customer_since gender home_store loyalty_card_number DATATABLE Demo Employee Lookup Food Inventory GENERATESERIES Demo Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store SELECTEDCOLUMN Demo SELECTEDCOLUMN Demo 2 Store Lookup Table Constructor Demo Target Sales - April 2019 (DATATABLE) Target Sales - March 2019 (DATATABLE)

customer_id	home_store	customer_first_name	customer_email	customer_since	loyalty_card_number	Birthdate	gender	birth_year	Customer Sales (High/Low)
301	3	Alex Ross	ross@alexr.com.us	04 January 2017	021-443-0482	23 May 1990	F	1990	Low
302	3	Sasha Wolf	yvonne@event.org	08 January 2017	433-162-9277	29 June 1990	F	1990	Low
303	3	Raya Hampton	Martina@malevada.com	08 January 2017	478-763-6984	24 August 1990	F	1990	Low
304	3	Beth Payne	Sirma@sociale.net	10 January 2017	830-000-9488	10 September 1990	F	1990	Low
305	3	Brooke Munoz	William@monbed.com	12 January 2017	573-552-0754	18 November 1990	F	1990	Low
306	3	Tanisha Wolf	Clinton@varius.edu	14 January 2017	869-238-7140	02 January 1991	F	1991	Low
307	3	Quintessa Franklin	Audrey@mattus.net	16 January 2017	877-181-0835	17 February 1991	F	1991	Low

Table: Customer Lookup (2,251 rows) Column: Customer Sales (High/Low) (2 distinct values) Update available (click to download)

Maven Rooster - Analysis - Power BI Desktop

Sign in

File Home Help External Tools Table tools Column tools

Name Customer Sales (High/Low) Data type Text

Summarization Don't summarize Data category Uncategorized Sort by column Data groups Manage relationships New column

Structure Formatting Properties Sort Group Relationships Calculations

1 Customer Sales (High/Low) =

customer_id	home_store	customer_first_name	customer_email	customer_since	loyalty_card_number	birthdate	gender	birth_year	Customer Sales (High/Low)
301	3	Alice Rivers	trelo@aliquam.us	04 January 2007	031-443-0482	13 May 1980	F	1990	Low
302	3	Sasha Wall	Yvonne@arist.org	08 January 2007	833-122-9777	29 June 1980	F	1990	Low
303	3	Kaya Hampton	Martha@magnifica.com	08 January 2007	478-762-8884	14 August 1980	F	1990	Low
304	3	Bene Bayne	Sierra@sodales.net	23 January 2007	830-000-3489	30 September 1980	F	1990	Low
305	3	Brooke Munro	William@montes.com	12 January 2007	573-551-0794	18 November 1980	F	1990	Low
306	3	Tashika Wolf	Conson@charis.edu	14 January 2007	849-238-7140	01 January 1981	F	1991	Low
307	3	Quintessa Franklin	Audrey@matthes.net	28 January 2007	877-781-0833	17 February 1981	F	1991	Low
308	3	Chantelle Cobb	Eagan@ulvamas.net	28 January 2007	618-483-1638	04 April 1981	F	1991	Low
309	3	Jordan Mattox	Cody@inductus.edu	20 January 2007	726-654-4038	21 May 1981	F	1991	Low
310	3	Cynthia Hurley	Miriam@arcu.edu	22 January 2007	209-573-4564	07 July 1981	F	1991	Low
311	3	Melodie Dawson	Stacey@modera.us	24 January 2007	894-557-4083	22 August 1981	F	1991	Low
312	3	Charlotte Webb	Phelan@nutrum.org	26 January 2007	542-155-7242	08 October 1981	F	1991	Low
313	3	Maryssa Jones	Dellan@tibus.edu	28 January 2007	623-851-0936	28 November 1981	F	1991	Low
314	3	Kurt Jackson	Shara@nra.gov	30 January 2007	371-921-0474	09 January 1982	F	1992	Low
315	3	Katina Benson	Hudson@unia.edu	02 February 2007	239-217-0154	23 February 1982	F	1992	Low
316	3	Camille Glass	Heddy@adipiscing.gov	04 February 2007	448-528-3198	11 April 1982	F	1992	Low
317	3	Reagan Roberts	Echo@integer.edu	06 February 2007	324-062-4547	28 May 1982	F	1992	Low
318	3	Chelsea Bray	Wilbow@omni.net	08 February 2007	288-181-8804	13 July 1982	F	1992	Low
319	3	Lesley Benson	Laura@elefond.net	10 February 2007	683-365-3330	29 August 1982	F	1992	High
320	3	Devon Matthews	Hannah@fachira.org	12 February 2007	213-992-6428	13 October 1982	F	1992	Low
321	3	Liam Lucas	Urho@verus.com	14 February 2007	865-762-7433	09 November 1982	F	1992	Low
322	3	Brittany Bradshaw	Rina@bus.us	16 February 2007	827-758-6053	16 January 1983	F	1993	Low
323	3	Cecilia Davidson	Donovan@potere.org	18 February 2007	768-601-4090	03 March 1983	F	1993	Low
324	3	Emma Cunningham	Vera@eugue.org	20 February 2007	495-345-8184	19 April 1983	F	1993	Low
325	3	Pamela Hay	Willard@jones.net	22 February 2007	160-784-1258	05 June 1983	F	1993	Low
326	3	Sandra Andrews	Sheaugh@mauria.net	24 February 2007	871-879-4220	21 July 1983	F	1993	Low
327	3	Shafra Mosa	Leah@socialeju.com	26 February 2007	249-422-5454	06 September 1983	F	1993	Low
328	3	Carla Douglas	Regie@test.com	28 February 2007	239-265-9162	12 October 1983	F	1993	High
329	3	Angela Strickland	Alma@rl.edu	02 March 2007	878-857-9342	08 December 1983	F	1993	Low
330	3	Chanya Marie	Stewart@cooperative.org	05 March 2007	658-713-8490	24 January 1984	F	1994	Low
331	3	Amelia Aravado	Timon@neurus.us	07 March 2007	656-404-4159	21 March 1984	F	1994	Low
332	3	Samantha Riddle	Alvin@purus.net	09 March 2007	114-078-3429	27 April 1984	F	1994	Low
333	3	Chava Guzman	Igor@condimentum.us	11 March 2007	875-302-8831	22 June 1984	F	1994	Low
334	3	Nita Golden	Brynn@handwerk.org	13 March 2007	719-055-8234	29 July 1984	F	1994	Low
335	3	Denitra Powell	Dense@teabus.edu	15 March 2007	197-880-1848	14 September 1984	F	1994	Low
336	3	Albert Zamora	Declan@aul.com	17 March 2007	162-823-6779	30 October 1984	F	1994	Low
337	3	Dana Curtis	Burton@msu.com	19 March 2007	836-180-4251	16 December 1984	F	1994	Low
338	3	Leslie Acevedo	Yuli@Curse.us	21 March 2007	306-952-3492	22 January 1985	F	1995	Low

2) Click here to collapse the DAX formula area.

Fields

Search

(Select all) High Low

OK Cancel

3) Customer Sales (High/Low) sales average calculated here.

Key objective 5 is now complete.
Return "High" if customer-level sales are above the average.
Otherwise, return "Low".

Table: Customer Lookup (2,251 rows) Column: Customer Sales (High/Low) (2 distinct values)

Update available (click to download)

Maven Roatton - Analysis - Power BI Desktop

Sign in

File Home Help External Tools Table tools Column tools

Name Customer Sales (High/Low) Summarization Don't summarize Data category Uncategorized Sort by column Data groups Manage relationships New column

Data type Text S - % Auto

Structure Formatting Properties Sort Group Relationships Calculations

1 Customer Sales (High/Low)

customer_id	home_store	customer_first_name	customer_email	customer_since	loyalty_card_number	birthdate	gender	birth_year	Customer Sales (High/Low)
8600	B Non-Member	sales@mavencenterduo	02 January 2017	000-000-0008	20 July 1983	Not Specific	1983	High	
8423	B Marissa	Rigel@unibulum.edu	30 August 2017	719-259-1585	15 August 1997	Not Specific	1997	High	
8395	B Allen	Clio@horummy.com	30 August 2018	849-028-8101	03 September 1999	Not Specific	1999	High	
8373	B Ted	Desiree@vuscpt.net	23 May 2018	411-736-2128	01 December 1999	Not Specific	1999	High	
8332	B Callie	Kai@comstud.gov	20 February 2017	318-623-4047	02 November 1994	Not Specific	1994	High	
8283	B Francesca	Audrey@mights.us	26 October 2016	174-904-8055	22 September 1990	Not Specific	1990	High	
8279	B Wile	Erasmus@diem.org	03 July 2016	129-175-9034	22 November 1998	Not Specific	1998	High	
8147	B Bianca	Xanthus@erst.us	17 April 2018	778-657-9279	20 May 1995	Not Specific	1995	High	
8138	B Barclay	Autumn@daphus.net	05 May 2017	281-465-8066	18 January 1998	Not Specific	1998	High	
8048	B Buckminster	Macy@coosqu.us	22 August 2018	129-868-9852	04 November 1998	Not Specific	1998	High	
8036	B Wyke	Steven@present.net	06 December 2018	171-771-7131	17 January 1992	Not Specific	1992	High	
8009	B Germaine	Jennifer@dotu.com	11 June 2018	241-684-6251	20 January 1993	Not Specific	1993	High	
7999	B Non-Member	sales@mavencenterduo	02 January 2017	000-000-0009	20 July 1983	Not Specific	1983	High	
797	B Greg Junior	john@mori.com	28 February 2017	254-423-7592	22 October 1993	Not Specific	1993	High	
6000	B Non-Member	sales@mavencenterduo	02 January 2017	000-000-0005	20 July 1983	Not Specific	1983	High	
1799	B Jamal	Lewis@it.gov	03 January 2019	799-181-9370	05 November 1999	Not Specific	1999	High	
5091	B Meghan	Stade@ln.org	11 January 2017	887-174-8230	24 June 1987	Not Specific	1987	High	
5084	B Randi	Chancy@nion.net	03 June 2018	551-458-9431	23 May 1995	Not Specific	1995	High	
214	B Burke Moss	Liberty@polocat.org	14 August 2018	511-460-1845	22 August 1986	M	1986	High	
130	B Dale Marquez	Armand@missa.net	26 December 2017	656-130-6731	08 May 1972	M	1972	High	
89	B Halimah Boulets	Calidene@pacconet.net	19 April 2017	813-671-8437	14 November 1998	M	1998	High	
26	B Harding Davis	Caroline@monastic.org	20 March 2017	423-412-1809	31 December 1984	M	1984	High	
26	B Samuel Gallagher	Paloma@bomber.org	14 March 2017	237-344-8792	29 August 1994	M	1994	High	
52	B Channing Lanson	Marny@doe.edu	10 March 2017	258-537-1032	23 December 1993	M	1993	High	
8	B Luis Cardenas	Brianus@relaxus.edu	22 January 2017	859-175-9187	30 September 1992	M	1992	High	
5827	B Stephen	Hedley@vastibulum.net	05 February 2018	854-967-1279	01 November 1974	M	1974	High	
5026	B Kiko	Justin@pivinher.edu	20 October 2017	493-077-1790	31 March 1968	M	1968	High	
6448	B Johnn	Kane@nbn.net	01 October 2018	760-987-7116	04 October 1991	M	1991	High	
8383	B Flynn	Ansel@henderson.gov	26 June 2018	512-253-1887	08 March 1972	M	1972	High	
8384	B Duncan	Zoile@color.gov	21 October 2017	645-921-7429	16 February 1963	M	1963	High	
8342	B Ferdinand	Abel@pyuvinorgov	04 June 2017	238-448-5604	22 February 1973	M	1973	High	
8310	B Tegan	Jelano@juma.com	09 October 2017	899-480-1446	29 July 1994	M	1994	High	
8289	B Gauke	Vernon@ormus.gov	01 June 2018	836-930-0481	22 December 1998	M	1998	High	
8287	B Jim	Verna@Hemera.us	19 September 2018	758-767-0309	24 February 1983	M	1983	High	
8237	B Trevor	Ursula.gov	07 April 2018	374-188-1890	06 April 1952	M	1952	High	
8226	B Kerell	Amelia@natoque.com	06 April 2018	882-548-4597	19 August 1952	M	1952	High	
8132	B Germ	Dorey@mutta.org	04 January 2018	239-258-9172	27 February 2000	M	1990	High	
8137	B Bo	Roary@Pellentesque.us	18 April 2018	301-565-1469	12 January 1990	M	1990	High	

Table: Customer Lookup (2,251 rows, 77 filtered rows) Column: Customer Sales (High/Low) (2 distinct values, 1 filtered distinct value)

Fields

Search

Measure Table ADDCOLUMNS Demo Calendar CROSSJOIN Demo Customer Lookup birth_year birthdate Customer Sales (High/Low) customer_email customer_first_name customer_id gender home_store loyalty_card_number DATABLES Demo Employee Lookup Food Inventory GENERALSERIES Demo Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store SELECTEDCOLUMNS Demo SELECTEDCOLUMNS Demo 2 Store Lookup Table Constructor Demo Target Sales - April 2019 (DATATABLE) Target Sales - March 2019 (DATATABLE)

Update available (click to download)

Maven Reactors - Analysis - Power BI Desktop

File Home Help External Tools Table tools Column tools

Name Customer Sales (High/Low) Data type Text Summarization Don't summarize Data category Uncategorized Sort by column Data groups Manage relationships New column

Structure Formatting Properties

1 Customer Sales (High/Low) =

customer_id	home_state	customer_first_name	customer_email	customer_since	loyalty_card_number	birthdate	gender	birth_year	Customer Sales (High/Low)
8177	I	Elijah	Kane@conque.us	08 April 2019	804-870-9086	06 February 1993	M	1993	Low
8178	I	Garrison	Stacie@conque.org	07 January 2017	468-461-4353	04 November 1997	M	1997	Low
8179	I	Johann	Pamela@dotcom.edu	23 May 2017	994-921-4580	21 July 1998	M	1998	Low
8180	I	Deean	Madeline@nra.gov	24 March 2019	879-123-2641	16 January 1993	M	1993	Low
8181	I	Anastasia	Lesley@ext.gov	06 June 2018	124-388-6640	18 March 1990	M	1990	Low
8182	I	Cameron	Krispen@lucy.net	20 March 2019	710-834-3908	20 July 1997	M	1997	Low
8183	I	Tobie	Adena@lum.gov	09 December 2018	739-909-9988	10 January 1992	M	1992	Low
8184	I	Burke	Dora@menetrix.net	22 November 2017	776-069-5251	15 September 1993	M	1993	Low
8185	I	Beck	Elton@disibus.net	26 August 2018	816-425-4926	17 November 1991	M	1991	Low
8186	I	Doreen	Keve@fisus.us	21 October 2017	991-272-9254	24 April 1993	M	1993	Low
8187	I	Felix	Peter@fisus.us	24 April 2017	994-377-0496	13 March 1997	M	1997	Low
8188	I	Hector	Casmara@elementum.edu	26 February 2019	899-014-6274	29 April 1997	M	1997	Low
8189	I	Burnan	Nigel@lipus.net	07 August 2017	826-624-4485	11 December 1992	M	1992	Low
8190	I	Ray	James@venenadiso.net	03 May 2017	582-870-2402	02 July 1992	M	1992	Low
8191	I	Upton	Reese@Clezz.edu	21 May 2017	022-825-4786	26 July 1993	M	1993	Low
8192	I	Benjamin	Madeline@egressus.gov	08 February 2017	513-828-3474	04 February 1992	M	1992	Low
8193	I	Benjamin	Leontine@metaculus.ph	04 June 2017	344-253-3898	07 October 1990	M	1990	Low
8194	I	Rooney	Audrey@Athean.gov	23 March 2018	271-860-3984	21 December 1994	M	1994	Low
8195	I	Hall	Raymond@mis.edu	08 January 2017	875-742-2774	17 March 1998	M	1998	Low
8196	I	Timothy	Dominique@accumulo.org	21 January 2018	468-724-5219	28 May 1992	M	1992	Low
8197	I	Oren	Lance@diem.edu	29 April 2017	453-814-5129	01 March 1993	M	1993	Low
8198	I	Mariann	Heidi@habitat.org	04 September 2017	287-195-3158	06 September 1992	M	1992	Low
8199	I	Ryan	Quamar@bedrock.org	21 August 2017	113-820-3568	05 May 1994	M	1994	Low
8200	I	Oliver	Portia@porta.com	21 October 2017	658-670-1200	20 January 1991	M	1991	Low
8201	I	Karen	Wesley@ellipz.it	30 May 2019	313-924-5292	14 June 1998	M	1998	Low
8202	I	Kareem	Brianna@temer.org	02 November 2019	789-936-1429	04 September 1990	M	1990	Low
8203	I	Damon	Lorraine@omere.org	20 March 2019	629-401-1165	12 October 1993	M	1993	Low
8204	I	Horace	Gi@ext.gov	18 January 2019	935-617-0120	04 August 1997	M	1997	Low
8205	I	Zachery	Clayton@receptus.org	17 August 2017	538-469-3770	26 April 1992	M	1992	Low
8206	I	Hayward	Victor@mauris.edu	01 June 2017	842-590-8896	20 October 1993	M	1993	Low
8207	I	Hammett	Timothy@front.gov	29 July 2017	686-457-1951	09 January 1997	M	1997	Low
8208	I	Burnt	Madison@ovaciones.us	03 March 2019	107-762-0460	14 October 1993	M	1993	Low
8209	I	Magoo	Aaron@nra.us	28 June 2018	548-344-8184	08 July 1999	M	1999	Low
8210	I	Rashim	M. Alireza@minis.net	04 March 2017	529-580-6257	18 August 1997	M	1997	Low
8211	I	Denton	Dominic@tempo.org	29 July 2018	826-474-6256	29 June 1994	M	1994	Low
8212	I	Nano	Kennos@mus.net	29 May 2019	831-320-4250	29 November 1995	M	1995	Low
8213	I	Kyle	Erich@metaculus.us	26 October 2018	126-317-8254	24 July 1999	M	1999	Low
8214	I	Melvin	Rafael@scots.org	01 October 2018	219-981-1899	20 September 1994	M	1994	Low

Table: Customer Lookup (2,251 rows, 2,174 filtered rows) Column: Customer Sales (High/Low) (2 distinct values, 1 filtered distinct value)

5) When filtering Low you will notice that 2251 sales are below the average sales.

USERELATIONSHIPS

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

USERELATIONSHIP()

Specifies an existing relationship to be used in the evaluation of a DAX expression, defined by naming, as arguments, the two columns that serve as endpoints.

=USERELATIONSHIP(Column**Name1**, Column**Name2**)

Foreign (or primary) key of the relationship

Examples:

- Food Inventory[Baked_Date]
- Calendar[Transaction Date]
- Calendar[Transaction Date]
- Food Inventory[Baked_Date]

Primary (or foreign) key of the relationship

Examples:

- Food Inventory[Baked_Date]
- Calendar[Transaction Date]
- Food Inventory[Baked_Date]



HEY THIS IS IMPORTANT!

USERELATIONSHIPS can only be used in functions which accept a filter parameter (**CALCULATE**, **TOTALYTD**, etc.)



PRO TIP:

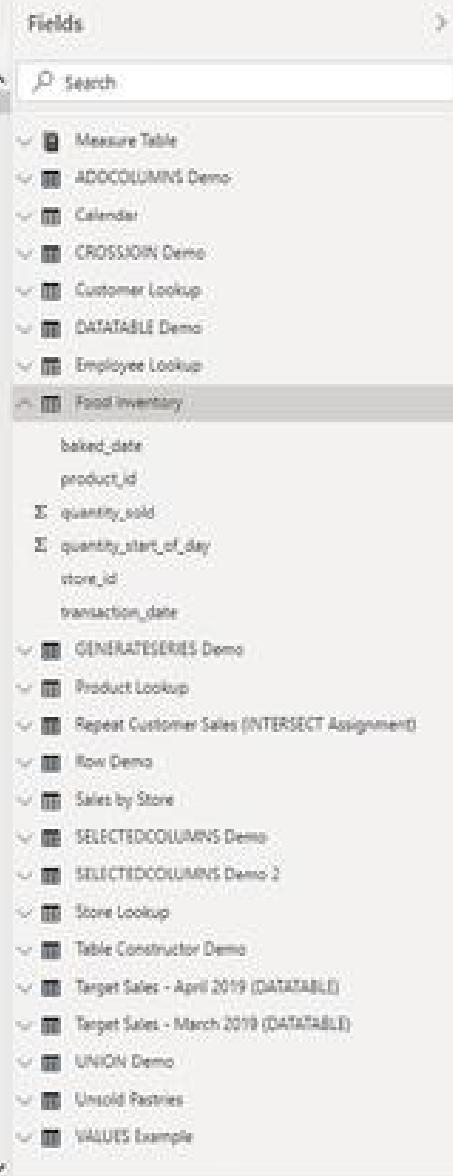
If you have **multiple date columns** connected to a single calendar table, **USERELATIONSHIP** is a great way to force measures to use **inactive relationships** without having to manually activate them in your model



Table: Food Inventory (21,649 rows)

store_id	baked_date	transaction_date	product_id	quantity_start_of_day	quantity_sold	
1	23 December 2016	06 January 2017	89	18	14	
1	23 December 2016	07 January 2017	89	18	14	
1	28 December 2016	08 January 2017	89	18	14	
1	31 December 2016	12 January 2017	89	18	14	
1	02 January 2017	13 January 2017	89	18	14	
1	07 January 2017	13 January 2017	89	18	14	
1	04 January 2017	18 January 2017	89	18	14	
1	18 January 2017	09 January 2017	89	18	14	
1	23 January 2017	07 February 2017	89	18	14	
1	01 February 2017	12 February 2017	89	18	14	
1	07 February 2017	18 February 2017	89	18	14	
1	11 February 2017	20 February 2017	89	18	14	
1	17 February 2017	25 February 2017	89	18	14	
1	20 March 2017	17 March 2017	89	18	14	
1	28 March 2017	29 March 2017	89	18	14	
1	03 April 2017	13 April 2017	89	18	14	
1	03 April 2017	14 April 2017	89	18	14	
1	03 April 2017	18 April 2017	89	18	14	
1	14 April 2017	23 April 2017	89	18	14	
1	19 April 2017	02 May 2017	89	18	14	
1	27 April 2017	07 May 2017	89	18	14	
1	30 April 2017	09 May 2017	89	18	14	
1	14 May 2017	21 May 2017	89	18	14	
1	14 May 2017	28 May 2017	89	18	14	
1	21 May 2017	02 June 2017	89	18	14	
1	24 May 2017	05 June 2017	89	18	14	
1	31 May 2017	10 June 2017	89	18	14	
1	07 June 2017	14 June 2017	89	18	14	
1	08 June 2017	19 June 2017	89	18	14	
1	07 June 2017	21 June 2017	89	18	14	
1	26 June 2017	09 July 2017	89	18	14	
1	26 June 2017	08 July 2017	89	18	14	
1	02 July 2017	11 July 2017	89	18	14	
1	04 July 2017	13 July 2017	89	18	14	
1	27 July 2017	06 August 2017	89	18	14	
1	06 August 2017	14 August 2017	89	18	14	
1	20 August 2017	31 August 2017	89	18	14	
1	28 August 2017	02 September 2017	89	18	14	

1) Look at the difference between baked date & transaction date for the quantity sold lets say compare those two.



Maven Reactors - Analysis - Power BI Desktop

Sign in

File Home Help External Tools

Get data -> Excel -> Power BI datasets -> SQL Server -> Enter data -> Recent sources ->

Transform Refresh data -> Manage relationships -> QSA Language setup -> DAX Language schema -> Publish

Dashboard Data Queries Relationships Security DAX Share

Properties

Select one or more model objects to set their properties.

Fields

Search

- ADDCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- DATABLE Demo
- Employee Lookup
- Food Inventory
- GENERATESERIES Demo
- Measure Table
- Product Lookup
- Repeat Customer Sales (INTERSECT Assign)
- Row Demo
- Sales by Store
- SELECTEDCOLUMNS Demo
- SELECTEDCOLUMNS Demo 2
- Store Lookup
- Table Constructor Demo
- Target Sales - April 2019 (DATATABLE)
- Target Sales - March 2019 (DATATABLE)
- UNION Demo
- Unsold Pastries
- VALUES Example

2) Need to make sure that the relationships are setup in the data model i.e. Calendar (Transaction Date) linked to Food Inventory (Baked Date)

All tables +

Update available (click to download)

Maven Brothers - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name: Quantity Sold (USERELATIONSHIP)
Home table: Measure Table
Format: Whole number
Data category: Uncategorized

4) Change the formatting accordingly over here.

1 Quantity Sold (USERELATIONSHIP) =
2 CALCULATE(
3 [SUM Quantity Sold],
4 USERELATIONSHIP(
5 'Food Inventory'[baked_date],
6 'Calendar'[Transaction_Date]
7)
8)

Visualizations >
Fields >
Search: Measure Table
Advanced CALCULATE
Calculated Table Joins
Relationship Functions
Average Order Value (CVP)
Bean % to Goal
Bean Goal (TREATAS)
Beverage % to Goal
Beverage Goal (TREATAS)
Customers who Purchased
Food % to Goal
Food Goal (TREATAS)
Merchandise % to Goal
Merchandise Goal (TREATAS)
Number of Employees (CROSSFILTER)
Quantity Sold (USERELATIONSHIP)
Wholesale Cost
Scalar Measures
Table & Filter Functions
Tips & Best Practices
ADDCOLUMNS Demo
Calendar
CROSSJOIN Demo
Customer Lookup
DATABLE Demo
Employee Lookup
EXCEPT Demo
Food Inventory
baked_date
product_id

3) Create the Quantity Sold (USERELATIONSHIP) measure.

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get Data = Excel Power BI datasets SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals Insert New measure measure Quick Publish

Datasources Datasets Data Queries Calculations Share

Wholesale Cost: \$2,268,299.27

Quantity Sold (USERRELATIONSHIP) SUM Quantity Sold: 141,433 141,433

Visualizations Fields

sum qua

Measure Table Advanced CALCULATE

CALCULATE('sum Quantity Sold')

SUM Quantity Sold

Values Add data fields here

Drill through

Cross-report

On Off

Keep all filters On

Add drill-through fields here

5) Perform the below actions.

- Add a matrix to the report.
- Add the Quantity Sold (USERRELATIONSHIP) field to the matrix.
- Add the SUM Quantity Sold field to the matrix.

6) Same values shown as expected makes sense without any other filter context

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7

Update available (click to download)

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Queries

Calculations

Share

Wholesale Cost
\$2,268,208.27

Year_ID Quantity Sold (USERELATIONSHIP) SUM Quantity Sold

Year_ID	Quantity Sold (USERELATIONSHIP)	SUM Quantity Sold
	478	
2017	56,242	55,943
2018	63,668	63,670
2019	21,045	21,820
Total	141,433	141,433

Visualizations Fields

Search

Measure Table Advanced CALCULATE Calculated Table Joins Relationship Functions Scalar Measures Table & Filter Functions Tips & Best Practices ADDCOLUMNS Demo Calendar Date Format (yyyy-mm-dd) Date_ID ISNUMBER Month_ID Month_Name Quarter & Year Quarter_ID Quarter_Name Transaction Date Week_Desc Week_ID Week_Year Year_Half Year_ID CROSSJOIN Demo Customer Lookup DATATABLE Demo Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo

Filters

Rows: Year_ID

Columns: Add data fields here

Values: Quantity Sold (USERELATIONSHIP)

SUM Quantity Sold

Drill through

Cross-report: Off

Keep all filters: On

Add drill-through fields here

7) When the year id is added to the matrix from the calendar table the following happens.

- Blank row appear at the top that has 478.
- Values for each year are different.

Page 7 of 7

Update available (click to download)

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals

New measure measure Insert Calculations Share

Visualizations Fields

Search

Measure Table Advanced CALCULATE Calculated Table Joins Relationship Functions Scalar Measures Table & Filter Functions Tips & Best Practices ADDCOLUMNS Demo Calendar Date Format (yyyy-mm-dd) Date_ID ISNUMBER Month_ID Month_Name Quarter & Year Quarter_ID Quarter_Name Transaction Date Week_Desc Week_ID Week_Year Year Half Year_ID CROSSJOIN Demo Customer Lookup DAXTABLE Demo Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo

Wholesale Cost
\$2,268,299.27

Quantity Sold (USERELATIONSHIP)

Year_ID	Quantity Sold (USERELATIONSHIP)	SUM Quantity Sold
2017	478	478
January	2,692	2,742
February	2,707	2,589
March	3,326	3,394
April	4,124	3,956
May	3,254	3,236
June	3,431	3,449
July	3,193	3,229
August	3,120	3,149
September	4,973	4,884
October	3,763	3,649
November	3,853	3,976
December	3,601	3,699
2018	63,648	63,670
January	4,638	4,684
February	4,145	4,102
March	4,831	4,829
April	3,091	3,066
May	3,827	3,759
June	3,709	3,620
July	3,433	3,480
August	3,337	3,319
Total	141,433	141,433

9) Click here so that you can drill down to more detail.

10) The 478 number is interesting to see.

8) Add Month Name from the calendar table to the rows you get to see deeper level of the differences between the Quantity Sold (USERELATIONSHIP) & SUM Quantity Sold.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7

Update available (click to download)

CROSSFILTER

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

CROSSFILTER()

Specifies cross filtering direction to be used for the duration of the DAX expression.
The relationship is defined by naming the two columns that serve as endpoints

=CROSSFILTER(LeftColumnName, RightColumnName2, CrossFilterType)

The two columns you want to use. **Left column** is typically the “many” side and **right column** is typically the “one” side

Examples:

- 'Sales by Store'[customer_id]
- 'Customer Lookup'[customer_id]

Specifies the direction of the CROSSFILTER

Examples:

- OneWay, Both, None



PRO TIP

Instead of bi-directional relationships, use CROSSFILTER to enable two-way filtering **ONLY in specific cases**



Let's say we wanted to look at the customer sales by the number of employees working on a specific day & specific store.

Year_ID	Quantity Sold (USERELATIONSHIP)	SUM Quantity Sold
	478	
	478	
2017	56,242	55,843
January	2,692	2,742
February	2,707	2,589
March	3,526	3,394
April	4,124	3,956
May	5,254	5,236
June	5,433	5,440
July	5,195	5,229
August	5,120	5,149
September	4,973	4,884
October	5,765	5,649
November	5,852	5,976
December	5,601	5,699
2018	63,668	63,670
January	4,638	4,654
February	4,145	4,102
March	4,831	4,829
April	5,091	4,966
May	5,827	5,759
June	5,700	5,820
July	5,433	5,480
August	5,337	5,316
Customer Sales	4,697	4,646
Total	141,433	141,433

store_id
All
Transaction_Date
Customer Sales
01/01/2017
\$2,508.20
02/01/2017
\$2,403.35
03/01/2017
\$2,565.00
04/01/2017
\$2,220.10
05/01/2017
\$2,418.85
06/01/2017
\$2,273.85
07/01/2017
\$2,787.00
08/01/2017
\$2,638.53
09/01/2017
\$2,676.61
10/01/2017
\$2,685.65
11/01/2017
\$2,555.75
12/01/2017
\$2,327.70
13/01/2017
\$3,033.60
14/01/2017
\$2,682.51
15/01/2017
\$3,167.71
16/01/2017
\$2,829.16
17/01/2017
\$3,285.80
18/01/2017
\$2,735.96
19/01/2017
\$2,913.68
20/01/2017
\$2,603.73
21/01/2017
\$3,082.85
22/01/2017
\$2,367.33
23/01/2017
\$2,853.15
24/01/2017
\$2,868.95
25/01/2017
\$2,846.55
Total
\$4,252,704.88

1) Add a matrix to the report.

2) Perform the below actions.

- Add Transaction Date to the rows.
- Add Customer Sales to values.
- Add slicer to the report
- Add store id to the slicer.
- Change slicer to dropdown view.

Visualizations

Fields

Measure Table

ADDCOLUMNS Demo

Calendar

CROSSJOIN Demo

Customer Lookup

DATABLE Demo

Employee Lookup

EXCEPT Demo

Food Inventory

GENERATESERIES Demo

New Employees (INTERSECT)

Product Lookup

Repeat Customer Sales (INTERSECT Assign..)

Row Demo

Sales by Store

SELECTCOLUMNS Demo

Store Lookup

SUMMARIZE Demo

Table Constructor Demo

Target Sales - April 2019 (DATATABLE)

Target Sales - March 2019 (DATATABLE)

TREATAS Demo

UNION Demo

Unsold Pastries

VALUES Example

Maven Reporters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name Number of Employees
Home table Measure Table \$ - % 9.0 0

Data category Uncategorized New measure Quick measure measure

Structure Formatting Properties Calculations

Visualizations Fields

Number of Employees (CROSSFILTER)

All Transaction Date Customer Sales Number of Employees (CROSSFILTER)

				55
2017	56,242	55,943	01/01/2017	55
January	2,692	2,742	02/01/2017	55
February	2,707	2,589	03/01/2017	55
March	3,526	3,394	04/01/2017	55
April	4,124	3,956	05/01/2017	55
May	5,254	5,236	06/01/2017	55
June	5,433	5,440	07/01/2017	55
July	5,195	5,229	08/01/2017	55
August	5,120	5,149	09/01/2017	55
September	4,973	4,884	10/01/2017	55
October	5,765	5,649	11/01/2017	55
November	5,852	5,976	12/01/2017	55
December	5,601	5,699	13/01/2017	55
2018	63,648	63,670	14/01/2017	55
January	4,638	4,684	15/01/2017	55
February	4,145	4,102	16/01/2017	55
March	4,831	4,829	17/01/2017	55
April	5,091	4,966	18/01/2017	55
May	5,827	5,759	19/01/2017	55
June	5,700	5,820	20/01/2017	55
July	5,433	5,480	21/01/2017	55
August	5,337	5,316	22/01/2017	55
September	4,769	4,699	23/01/2017	55
Total	141,413	141,433	24/01/2017	55
			Total	\$4,252,704.88

4) You will notice that the 55 value is repeating for the date range as there is no relationship between Employees & Date table.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7 Update available (click to download)

Maven Bousters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name: Number of Employees
Home table: Measure Table
Format: Whole number
Data category: Uncategorized

New Quick Measure measure
Measure Calculations

1 Number of Employees (CROSSFILTER) =
2 CALCULATE(
3 COUNTROWS(
4 'Employee Lookup'
5),
6 CROSSFILTER(
7 'Sales by Store'[staff_id],
8 'Employee Lookup'[staff_id],
9 Both
10)
11)

Visualizations > Fields >

Number of Employees (CROSSFILTER)
Measure Table
Relationship Functions
Number of Employees (CROSSFILTER)
Calendar
Customer Lookup
loyalty_card_number
DATATARU Demo
Test Number
EXCEPT Demo
loyalty_card_number
Product Lookup
Number of Food Items Made

Rows: Transaction Date
Columns: Add data fields here
Values: Customer Sales
Number of Employees
Drill through:
Cross-report: Off
Keep all filters: On

24/07/2017 24,000.37
Total \$4,252,704.88 23 55

Tips & Best Practices | Scalar Functions | Advanced CALCULATE | Table & Filter Functions | Joins | Relationship Functions | Duplicate of Relationship Functions | + | Update available (click to download)



6) Repeating employees values are now fixed total number of employees are 55.

The screenshot shows a Power BI report interface with the following details:

- Visualizations:** A matrix visualization titled "Transaction_Date - Customer Sales - Number of Employees (CROSSFILTER)" showing sales data for January 2017.
- Fields:** A sidebar showing the available fields in the model, including "Number of Employees (CROSSFILTER)" which is highlighted in yellow.
- Annotations:** A large blue arrow points from the text "Repeating employees values are now fixed total number of employees are 55." to the value "55" in the bottom-right corner of the matrix visualization.

	Customer Sales	Number of Employees
Total	\$4,252,704.88	55

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Insert Text box More visuals New measure measure Quick Publish

Queries

Visualizations Fields

Measure Table Relationship Functions Number of Employees (CROSSFILTER) Customer Lookup Loyalty_card_number DATAPRINT Demo Product Lookup Number of Food Items Made

7) Slicer also refreshes the data correctly now for store id 3.

store_id

3

Transaction_Date Customer Sales Number of Employees (CROSSFILTER)

Transaction_Date	Customer	Sales	Number of Employees (CROSSFILTER)
01/01/2017		\$868.40	2
02/01/2017		\$925.50	2
03/01/2017		\$902.75	3
04/01/2017		\$808.25	2
05/01/2017		\$903.05	2
06/01/2017		\$716.05	2
07/01/2017		\$807.30	3
08/01/2017		\$793.15	3
09/01/2017		\$774.01	3
10/01/2017		\$947.00	3
11/01/2017		\$740.15	3
12/01/2017		\$860.70	3
13/01/2017		\$705.80	3
14/01/2017		\$763.70	3
15/01/2017		\$1,348.93	3
16/01/2017		\$1,027.33	3
17/01/2017		\$1,075.40	3
18/01/2017		\$1,064.03	3
19/01/2017		\$1,079.38	3
20/01/2017		\$871.43	3
21/01/2017		\$992.35	3
22/01/2017		\$742.80	3
23/01/2017		\$928.30	3
24/01/2017		\$922.05	3
25/01/2017		\$942.00	3
Total		\$1,420,721.10	12

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7 Update available (click to download)

File Home Help External Tools Table tools Column tools

Name staff_id
Data type Whole number

Format Whole number
\$ - % , . 0

Summarization Don't summarize
Data category Uncategorized

Sort by column Sort

Data groups Group

Manage relationships Relationships

New column Column

transaction_id	transaction_date	transaction_time	store_id	customer_id	instore_y_n	order	line_item_id	product_id	quantity_sold	unit_price	promo_item_y_n	Time Group	Total Revenue	Product Group
801	03 January 2017	18:11:11	1					70	1	3.75	N	18:00-20	\$3.75	Food
1284	03 January 2017	18:11:40	1					70	1	3.75	N	18:00-20	\$3.75	Food
1340	03 January 2017	18:45:31	1					70	1	3.75	N	18:00-20	\$3.75	Food
21	03 January 2017	18:52:00	1					70	1	3.75	N	18:00-20	\$3.75	Food
577	03 January 2017	18:44:43	1					70	1	3.75	N	14:00-16	\$3.75	Food
420	03 January 2017	19:02:09	1					70	1	3.75	N	19:00-01	\$3.75	Food
1897	03 January 2017	19:53:30	1					70	1	3.75	N	19:00-01	\$3.75	Food
2043	03 January 2017	19:58:56	1					70	1	3.75	N	19:00-01	\$3.75	Food
106	03 January 2017	12:45:37	1					70	1	3.75	N	12:00-00	\$3.75	Food
1358	03 January 2017	19:09:18	1					69	1	3.25	N	15:00-00	\$3.25	Food
2550	03 January 2017	17:13:46	1					69	1	3.25	N	17:00-00	\$3.25	Food
2497	03 January 2017	19:37:29	1					69	1	3.25	N	15:00-00	\$3.25	Food
1890	03 January 2017	19:52:13	1	12	Y			70	1	3.75	N	19:00-00	\$3.75	Food
1127	03 January 2017	18:18:51	1	12	Y			70	1	3.75	N	11:00-00	\$3.75	Food
1282	03 January 2017	20:17:17	1	12	Y			70	1	3.75	N	16:00-00	\$3.75	Food
231	03 January 2017	22:03:18	1	12	Y			70	1	3.75	N	11:00-00	\$3.75	Food
2171	03 January 2017	14:47:10	1	17	Y			70	1	3.75	N	14:00-20	\$3.75	Food
1238	03 January 2017	14:58:17	1	17	Y			70	1	3.75	N	14:00-20	\$3.75	Food
2464	03 January 2017	11:46:29	1	17	Y			70	1	3.75	N	11:00-00	\$3	Food
879	03 January 2017	11:47:48	1	17	Y			70	1	3	N	11:00-00	\$3	Food
181	03 January 2017	19:07:24	1	17	Y			70	1	4.5	N	18:00-00	\$4.5	Food
991	03 January 2017	11:26:40	1	17	Y			70	1	4.5	N	11:00-00	\$4.5	Food
1784	03 January 2017	15:02:54	1	12	Y			24	1	3	N	13:00-00	\$3	Beverages
1737	03 January 2017	15:16:04	1	12	Y			51	1	3	N	13:00-00	\$3	Beverages
2479	03 January 2017	16:51:10	1	12	Y			50	1	3	N	16:00-00	\$3	Beverages
2662	03 January 2017	16:18:52	1	12	Y			49	1	3	N	16:00-00	\$3	Beverages
581	03 January 2017	14:07:15	1	12	Y			57	1	3	N	14:00-00	\$3	Beverages
1882	03 January 2017	16:18:17	1	12	Y			47	1	3	N	16:00-00	\$3	Beverages
436	03 January 2017	16:22:03	1	17	Y			50	1	3	N	16:00-00	\$3	Beverages
187	03 January 2017	14:45:14	1	17	Y			57	1	3	N	14:00-00	\$3	Beverages
369	03 January 2017	15:52:01	1	17	Y			40	1	3	N	15:00-00	\$3	Beverages
2440	03 January 2017	19:33:27	1	17	Y			52	1	3	N	18:00-00	\$3	Beverages
2023	03 January 2017	18:20:48	1	17	Y			47	1	3	N	18:00-00	\$3	Beverages
106	03 January 2017	12:45:07	1	17	Y			51	1	3	N	12:00-00	\$3	Beverages
2014	03 January 2017	12:28:03	1	17	Y			51	1	3	N	12:00-00	\$3	Beverages
1357	03 January 2017	18:01:57	1	17	Y			40	1	3	N	18:00-00	\$3	Beverages
1837	03 January 2017	18:37:29	1	17	Y			40	1	3	N	18:00-00	\$3	Beverages
2084	03 January 2017	19:58:14	1	17	Y			40	1	3	N	11:00-00	\$3	Beverages

8) Perform the below actions to reconcile the store id 3 value

- Go to the data view & select the Sales by Store table.
- Go to the transaction date column & filter to 1/3/2017.
- Go to store id column filter to 3.

9) 2 employees correctly validated for 1/3/2017.

Fields
Measure Table
ADDCOLUMNS Demo
Calendar
CROSSJOIN Demo
Customer Lookup
DATABLE Demo
Employee Lookup
EXCEPT Demo
Food Inventory
GENERATESERIES Demo
New Employees (INTERSECT A..)
Product Lookup
Repeat Customer Sales (INTERSECT A..)
Row Demo
Sales by Store
customer_id
instore_y_n
line_item_id
order
Product Group
product_id
promo_item_y_n
quantity_sold
staff_id
store_id
Time Group
Total Revenue
transaction_date
transaction_id
transaction_time

KEY OBJECTIVES

- 1) Create a measure called "Customers who Purchased" that uses **COUNTROWS & CROSSFILTER** to calculate the number of customers who made a purchase in a given time period.
- 2) Create a measure that calculates the average order value for "Customers who Purchased".
- 3) Create a matrix that shows the previous two measures broken down.

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Average Order Value... Format: Currency Data category: Uncategorized New measure Quick measure

Home table: Measure Table \$ - % 9 0 2 Properties Calculations

2) Objective 2 to create a measure that calculates the average order value for "Customers who Purchased".

Average Order Value (CWP) =
DIVIDE(
[Customer Sales],
[Customers who Purchased],
BLANK()

Visualizations Fields

Search

Measure Table

- Advanced CALCULATE
- Calculated Table Joins
- Relationship Functions
- Average Order Value (CWP)
- Blanks % to Goal
- Blanks Goal (TREATAS)
- Beverage % to Goal
- Beverage Goal (TREATAS)
- Customers who Purchased
- Food % to Goal
- Food Goal (TREATAS)
- Merchandise % to Goal
- Merchandise Goal (TREATAS)
- Number of Employees (CROSSFILTER)
- Quantity Sold (USERELATIONSHIP)
- Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tip & Best Practices
- ADOCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- DATATABLE Demo
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Demo
- New Employees (INTERSECT)

Values

Add data fields here

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Key objective 2 is now complete. Create a measure that calculates the average order value for "Customers who Purchased".

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7

Update available (click to download)

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get Data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Queries

Calculations

Share

Visualizations

Fields

Search

Measure Table

Advanced CALCULATE

Calculated Table / Join

Relationship Functions

- Average Order Value (CWP)**
- Bean % to Goal**
- Bean Goal (TREATAS)**
- Beverage % to Goal**
- Beverage Goal (TREATAS)**
- Customer who Purchased**
- Food % to Goal**
- Food Goal (TREATAS)**
- Merchandise % to Goal**
- Merchandise Goal (TREATAS)**
- Number of Employees (CROSSPI...**
- Quantity Sold (USERELATIONSHIP)**
- Wholesale Cost**
- Scalar Measures**
- Table & Filter Functions**
- Tips & Best Practices**
- ADDCOLUMNS Demo**

Filters

Row

Year_ID

Month_Name

product_group

Columns

Add data fields here

Values

Customers who Purchased

Average Order Value (CWP)

Keep all filters

On

Add drill-through fields here

Calendar

- Date Format (yyyy-mm-dd)**
- Date_ID**
- isNUMBER**
- Month_ID**
- Month_Name**
- Quarter & Year**
- Quarter_ID**
- Quarter_Name**
- Transaction Date**

3) Objective 3 to create a matrix that shows the previous two measures broken down.

Key objective 3 is now complete. Create a matrix that shows the previous two measures broken down.

4) Perform the below actions.

- Add the year id, month name & product group fields to the matrix to the rows.
- Add the Customers who Purchased field to the values.
- Add the Average Order Value (CWP) field to the values.
- Within the store id slicer clear all filters.

Year_ID	Customer who Purchased	Average Order Value (CWP)
2017	2,250	\$745.81
January	2,116	\$38.68
Add-ons	253	\$3.85
Beverages	2,113	\$29.49
Food	932	\$10.25
Merchandise	38	\$49.74
Whole Bean/Teas	143	\$49.77
February	2,094	\$36.48
Add-ons	246	\$3.79
Beverages	2,089	\$28.35
Food	911	\$9.93
Merchandise	30	\$41.17
Whole Bean/Teas	149	\$39.16
March	2,175	\$45.59
Add-ons	297	\$3.96
Beverages	2,173	\$34.80
Total	2,251	\$1,689.25

Physical vs. Virtual Relationships

RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS

TREATAS()

Applies the result of a table expression to filter columns in an unrelated table
(essentially creating a new, virtual relationship)

=TREATAS(TableExpression, ColumnName, [ColumnName], [...])

A table expression which generates the set of columns to be mapped. Table expression must be based on physical table in data model.

Examples:

- `TREATAS(
VALUES('Calendar'[Year_ID]...)`
- `TREATAS(
SUMMARIZE('Calendar',
'Calendar'[Year_ID], 'Calendar[Month]...)`

The list of output columns (cannot be an expression)

NOTE: The number of columns specified must match the **number of columns** in the table expression and be in the **same order**

Examples:

- `'Target Sales'[Year]`
- `'Target Sales'[Year],
'Target Sales'[Month]`

PRO TIP:

Use **physical relationships** (or **USERELATIONSHIP** functions) whenever possible, and only rely on TREATAS if you are unable to create a direct relationship between tables

Maven Rowsets - Analysis - Relationship Functions - Power BI Desktop

File Home Help External Tools Table tools

Name TREATAS Demo

Mark as data table - Manage relationships New measure Quick New measure column New table Calculations

Structure X ✓

```
1 TREATAS Demo =  
2 VAR StoreID =  
3 {  
4     "1",  
5     "2",  
6     "3",  
7     "4",  
8     "5",  
9     "6"  
10 }  
11 RETURN  
12 TREATAS(  
13     StoreID,  
14     'Store Lookup'[store_id]  
15 )
```

Fields

Search

- Measure Table
- ADOCOLUMNS Demo
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- DATATABLE Demo
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Demo
- New Employees (INTERSECT)
- Product Lookup
- Repeat Customer Sales (INTERSEC...
- Row Demo
- Sales by Store
- SELECTCOLUMNS Demo
- Store Lookup
- SUMMARIZE Demo
- Table Constructor Demo
- Target Sales - April 2019 (DATATAB...
- Target Sales - March 2019 (DATATAB...
- TREATAS Demo
- store_id
- UNION Demo
- Unsold Products
- VALUES Example

Table: TREATAS Demo (3 rows)

Update available (click to download)

1) Go to the data view & create a new table called TREATAS Demo.

2) We created a virtual table with values between 1-6 using the StoreID variable doesn't have a relationship with the Store Lookup table. Taken the virtual table values & relate them to the Store Lookup table.

3) The results returned all of the rows within the Store Lookup store id column also exist in the StoreID virtual table are returned reason why 1-6 values are not returned.

TREATAS (EXAMPLE)

Physical vs. Virtual Relationships

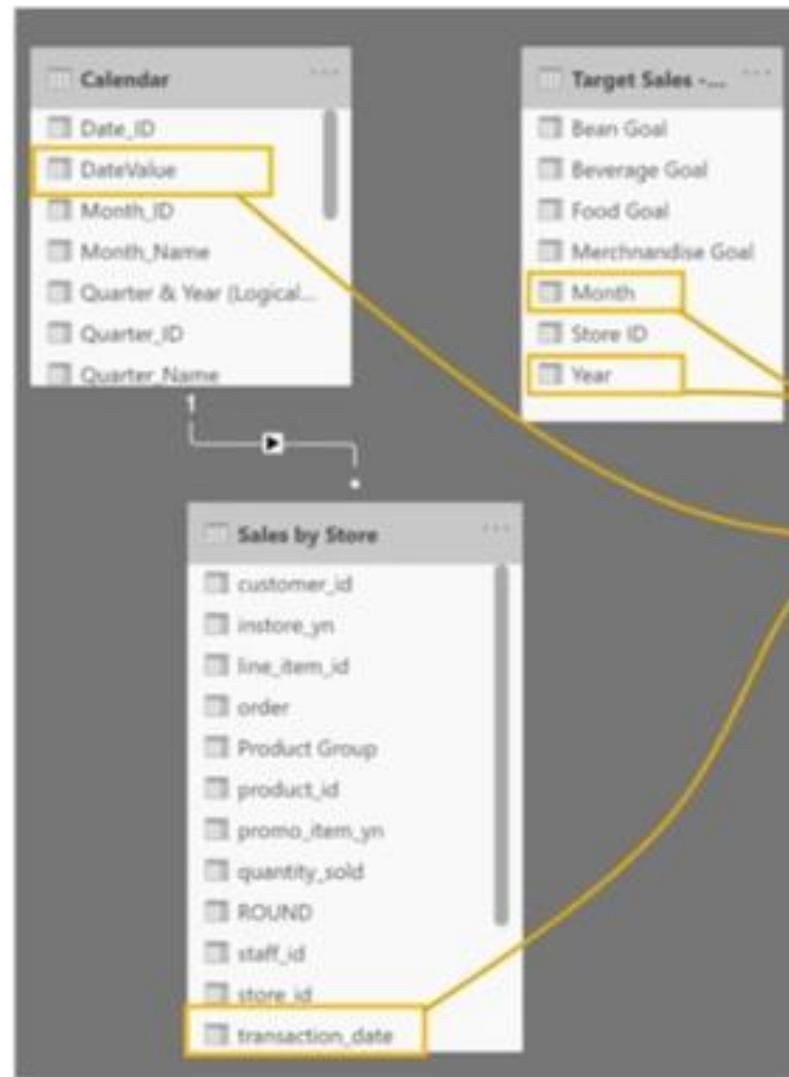
RELATED

RELATEDTABLE

USERELATIONSHIP

CROSSFILTER

TREATAS



In this case, we can't create physical relationships to **Target Sales** since it's at a *different level of granularity* than the other tables in the model

Target Sales is at the **Month & Year** level

Both **Sales by Store** and **Calendar** are at the **Date (daily)** level

TREATAS allows us to create *virtual, summarized* versions of our tables to match the granularity that we need to form a valid relationship

ASSIGNMENT Sales Targets (TREATAS)

KEY OBJECTIVES

- 1) Based on the Target Sales Union table, use **TREATAS** & create measures for Bean/Tea, Beverage, Merchandise & Food sales goals.
- 2) Create **% of goal** measures that compare quantity sold to the goal amount.
- 3) Add the above measures to a matrix broken down by store & target months.

The screenshot shows the Power BI Data Model view. On the left, there is a grid of tables including Sales, Customer, Product, Store, and various Lookups. Relationships between tables are indicated by lines connecting their primary keys. A blue arrow points from the text below to the relationship line between the Store and Store Lookups tables. The Store table has its 'Store ID' field selected. The Store Lookup table also has its 'Store ID' field selected. The right side of the screen shows the Properties and Fields panes.

Ensure that there is a relationship between Store Lookup & Union Demo tables with the store id field so that the matrix visual does not output repeating values.

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Bean Goal (TREATAS) Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % , . : New measure Quick measure measure

Structure Properties Calculations

```

1 Bean Goal (TREATAS) =
2 CALCULATE(
3     SUM(
4         'UNION Demo'[Bean/Teas Goal]
5     ),
6     TREATAS(
7         SUMMARIZE(
8             'Calendar',
9             'Calendar'[Year_ID],
10            'Calendar'[Month_Name]
11        ),
12        'UNION Demo'[Year],
13        'UNION Demo'[Month]
14    )
15 )

```

- 1) Objective 1 based on the Target Sales Union table, use TREATAS & create measures for Bean/Tea, Beverage, Merchandise & Food sales goals. Create the Bean Goal (TREATAS) measure.

Visualizations Fields

Measure Table

- Advanced CALCULATE
- Calculated Table Joins
- Relationship Functions
 - Average Order Value (CWP)
 - Bean % to Goal
 - Bean Goal (TREATAS)
 - Beverage % to Goal
 - Beverage Goal (TREATAS)
 - Customer who Purchased
 - Food % to Goal
 - Food Goal (TREATAS)
 - Merchandise % to Goal
 - Merchandise Goal (TREATAS)
 - Number of Employees (CROSSP...)
 - Quantity Sold (USERELATIONSHIP)
 - Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices
- ADDCOLUMNS Demo
- Calendar
 - Date Format (yyyy-mm-dd)
 - Date_ID
 - ISNUMBER
 - Month_ID
 - Month_Name
 - Quarter & Year
 - Quarter_ID
 - Quarter_Name
 - Transaction Date

Add data fields here

Drill through

Cross-report

OR —

Keep all filters

On —

Add drill-through fields here

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name: Beverage Goal (TREATAS)
Home table: Measure Table
Format: Whole number
Data category: Uncategorized

Structure Properties Calculations

1 Beverage Goal (TREATAS) =

2 CALCULATE(

3 SUM(

4 'UNION Demo'[Beverage Goal]

5),

6 TREATAS(

7 SUMMARIZE(

8 'Calendar',

9 'Calendar'[Year_ID],

10 'Calendar'[Month_Name]

11),

12 'UNION Demo'[Year],

13 'UNION Demo'[Month]

14)

15)

2) Create the Beverage Goal (TREATAS) measure.

Visualizations Fields

Search

Measure Table

- Advanced CALCULATE
- Calculated Table/Join
- Relationship Functions**
 - Average Order Value (CWP)
 - Beverage % to Goal
 - Beverage Goal (TREATAS)
 - Customers who Purchased**
 - Food % to Goal
 - Food Goal (TREATAS)
 - Merchandise % to Goal
 - Merchandise Goal (TREATAS)
 - Number of Employees (CROSSPNT)
 - Quantity Sold (USERELATIONSHIP)
 - Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices

Add data fields here

Row

Year_ID Month_Name product_group

Columns

Add data fields here

Values

Customers who Purchased Average Order Value (CWP)

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- isNUMBER
- Month_ID
- Month_Name**
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction Date

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7

Update available (click to download)

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Food Goal (TREATAS) Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % .0 : New measure Quick measure measure

Structure Formatting Properties Calculations

Visualizations Fields

Search

Measure Table

- Advanced CALCULATE
- Calculated Table Joins
- Relationship Functions
 - Average Order Value (CWP)
 - Bean % to Goal
 - Bean Goal (TREATAS)
 - Beverage % to Goal
 - Beverage Goal (TREATAS)
 - Customers who Purchased
 - Food % to Goal
 - Food Goal (TREATAS)
 - Merchandise % to Goal
 - Merchandise Goal (TREATAS)
 - Number of Employees (CROSSPAG...
 - Quantity Sold (USRELATIONSHIP)
 - Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices
- ADDCOLUMNS Demo

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- KNUMBER
- MAHNUID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction Date

1 Food Goal (TREATAS) =
2 CALCULATE(
3 SUM(
4 'UNION Demo'[Food Goal]
5),
6 TREATAS(
7 SUMMARIZE(
8 'Calendar',
9 'Calendar'[Year_ID],
10 'Calendar'[Month_Name]
11),
12 'UNION Demo'[Year],
13 'UNION Demo'[Month]
14)
15)

3) Create the Food Goal (TREATAS) measure.

Tip & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7 Update available (click to download)

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Merchandise Goal L... Format: Whole number Data category: Uncategorized

Home table: Measure Table \$ - % , ; : New measure Quick measure measure Calculations

Structure Formatting Properties Calculations

Key objective 1 is now complete. Based on the Target Sales Union table, use TREATAS & create measures for Bean/Tea, Beverage, Merchandise & Food sales goals.

```

1 Merchandise Goal (TREATAS) =
2 CALCULATE(
3     SUM(
4         'UNION Demo'[Merchandise Goal]
5     ),
6     TREATAS(
7         SUMMARIZE(
8             'Calendar',
9             'Calendar'[Year_ID],
10            'Calendar'[Month_Name]
11        ),
12        'UNION Demo'[Year],
13        'UNION Demo'[Month]
14    )
15 )

```

4) Create the Merchandise Goal (TREATAS) measure.



Visualizations Fields

Search

Measure Table

- Advanced CALCULATE
- Calculated Table Joins
- Relationship Functions
 - Average Order Value (CWP)
 - Bean % to Goal
 - Bean Goal (TREATAS)
 - Beverage % to Goal
 - Beverage Goal (TREATAS)
 - Customers who Purchased
 - Food % to Goal
 - Food Goal (TREATAS)
 - Merchandise % to Goal
 - Merchandise Goal (TREATAS)
 - Number of Employees (CROSSPIT...)
 - Quantity Sold (USERELATIONSHIP)
 - Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices
- ADOCOLUMNS Demo
- Calendar
 - Date Format (yyyy-mm-dd)
 - Day_ID
 - ISNUMBER
 - Month_ID
 - Month_Name
 - Quarter & Year
 - Quarter_ID
 - Quarter_Name
 - Transaction Date

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Bean % to Goal Format: Percentage Data category: Uncategorized

Home table: Measure Table \$ - % 9.8 2 New Quick measure measure

Structure Formatting Properties Calculators

1 Bean % to Goal =

2 DIVIDE(

3 [Bean Goal (TREATAS)], ← 1) Objective 2 to create % of goal measures
that compare quantity sold to the goal
amount.

4 CALCULATE(

5 SUM(

6 'Sales by Store'[quantity_sold]

7),

8 'Product Lookup'[product_group] = "Whole Bean/Teas"

9)

10)

Transaction Date	Quantity Sold	Merchandise	Count	Total
01/01/2017	\$2,403.35	Merchandise	30	\$41.17
02/01/2017	\$2,565.00	Whole Bean/Teas	149	\$39.16
03/01/2017	\$2,220.10			
04/01/2017	\$2,418.85	March	2,175	\$45.59
05/01/2017	\$2,418.85	Add-ons	297	\$3.95
06/01/2017	\$2,273.85	Accessories	3,173	\$35.35
07/01/2017	\$2,787.00	Total	2,251	\$1,689.25
08/01/2017	\$2,635.53			
09/01/2017	\$2,676.61			
10/01/2017	\$2,685.65			
Total	\$4,252,704.88			

Visualizations Fields

Search

Measure Table

- Advanced CALCULATE
- Calculated Table Joins
- Relationship Functions
- Average Order Value (CWP)
- Bean % to Goal
- Bean Goal (TREATAS)
- Beverage % to Goal
- Beverage Goal (TREATAS)
- Customers who Purchased
- Food % to Goal
- Food Goal (TREATAS)
- Merchandise % to Goal
- Merchandise Goal (TREATAS)
- Number of Employees (CROSSP...
- Quantity Sold (USRELATIONSHIP)
- Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices
- ADDCOLUMNS Demo

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- ISNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction Date

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7

Update available (click to download)

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Beverage % to Goal Format: Percentage Data category: Uncategorized

Home table: Measure Table \$ - % , ; : New Quick measure measure

Structure Formatting Properties Calculations

1 Beverage % to Goal =

2 DIVIDE(

3 [Beverage Goal (TREATAS)], 

4 CALCULATE(

5 SUM(

6 'Sales by Store'[quantity_sold]

7),

8 'Product Lookup'[product_group] = "Beverages"

9)

10)

Transaction Date	Quantity Sold	Unit Price	Total
01/01/2017	\$2,403.35	10	\$41.17
02/01/2017	\$2,565.00	149	\$39.16
03/01/2017	\$2,220.10	2,175	\$45.59
04/01/2017	\$2,415.85	Add-ons	\$3.95
05/01/2017	\$2,273.85	Merchandise	\$164.85
06/01/2017	\$2,787.00	Total	\$1,889.25
07/01/2017	\$2,638.53		
08/01/2017	\$2,676.61		
09/01/2017	\$2,689.65		
Total	\$4,252,704.88		

Visualizations Fields

Search:

- Measure Table
- Advanced CALCULATE
- Calculated Table Joins
- Relationship Functions
 - Average Order Value (CWP)
 - Bean % to Goal
 - Bean Goal (TREATAS)
 - Beverage % to Goal
 - Beverage Goal (TREATAS)
 - Customers who Purchased
 - Food % to Goal
 - Food Goal (TREATAS)
 - Merchandise % to Goal
 - Merchandise Goal (TREATAS)
 - Number of Employees (CROSSPIT...)
 - Quantity Sold (USERELATIONSHIP)
 - Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices
- ADOCOLUMNS Demo
- Calendar
 - Date Format (yyyy-mm-dd)
 - Date_ID
 - ISNUMBER
 - Month_ID
 - Month_Name
 - Quarter & Year
 - Quarter_ID
 - Quarter_Name
 - Transaction Date

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions

Page 7 of 7 Update available (click to download)

Maven Roasters - Analysis - Relationship Functions - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Food % to Goal Format: Percentage Data category: Uncategorized

Home table: Measure Table \$ - % , . : New measure Quick measure measure

Structure Properties Calculations

Visualizations Fields

Search

Measure Table

Advanced CALCULATE

Calculated Table Joins

Relationship Functions

- Average Order Value (CWR)
- Bean % to Goal
- Bean Goal (TREATAS)
- Beverage % to Goal
- Beverage Goal (TREATAS)
- Customers who Purchased
- Food % to Goal
- Food Goal (TREATAS)
- Merchandise % to Goal
- Merchandise Goal (TREATAS)
- Number of Employees (CROSSFILTER)
- Quantity Sold (USERELATIONSHIP)
- Wholesale Cost

Scalar Measures

Table & Filter Functions

Tips & Best Practices

ADOCOLUMNS Demo

Calendar

- Date Format (yyyy-mm-dd)
- Date_ID
- isNUMBER
- Month_ID
- Month_Name
- Quarter & Year
- Quarter_ID
- Quarter_Name
- Transaction Date

1 Food % to Goal =
2 DIVIDE(
3 [Food Goal (TREATAS)],
4 CALCULATE(
5 SUM(
6 'Sales by Store'[quantity_sold]
),
7 'Product Lookup'[product_group] = "Food"
8)
9)
10)

01/01/2017 \$2,403.35 Merchandise 30 \$41.17
03/01/2017 \$2,565.00 Whole Bean/Teas 149 \$39.16
04/01/2017 \$2,220.10 March 2,175 \$45.59
05/01/2017 \$2,418.85 Add-ons 297 \$1.96
06/01/2017 \$2,273.85 Beverages 2,173 \$34.89
07/01/2017 \$2,787.00 Total 2,251 \$1,689.25
08/01/2017 \$2,635.53
09/01/2017 \$2,676.61
10/01/2017 \$2,685.65
Total \$4,252,704.88

Tip & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Duplicate of Relationship Functions +

Page 7 of 7

Update available (click to download)

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Merchandise % to ... Format: Percentage Data category: Uncategorized

Home table: Measure Table \$ - % , . : New measure Quick measure

Structure Formatting Properties Calculations

Key objective 2 is now complete. Create % of goal measures that compare quantity sold to the goal amount.

```

1 Merchandise % to Goal =
2 DIVIDE(
3   [Merchandise Goal (TREATAS)],
4   CALCULATE(
5     SUM(
6       'Sales by Store'[quantity_sold]
7     ),
8     'Product Lookup'[product_group] = "Merchandise"
9   )
10 )

```

4) Create the Merchandise % to Goal measure.

Transaction Date	Quantity Sold	Product Group	Unit Price	Total Value
01/01/2017	\$2,403.35	Merchandise	30	\$41.17
01/01/2017	\$2,565.00	Whole Bean/Teas	149	\$39.16
04/01/2017	\$2,220.10	March	2,175	\$45.59
05/01/2017	\$2,418.85	Add-ons	297	\$3.96
06/01/2017	\$2,273.85	Accessories	2,172	\$39.30
07/01/2017	\$2,787.00	Total	2,251	\$1,089.25
08/01/2017	\$2,638.53			
09/01/2017	\$2,676.61			
10/01/2017	\$2,685.65			
Total	\$4,282,704.88			

Visualizations > Fields >

Search

Measure Table

- Advanced CALCULATE
- Calculated Table Icons
- Relationship Functions
 - Average Order Value (CWP)
 - Bean % to Goal
 - Bean Goal (TREATAS)
 - Beverage % to Goal
 - Beverage Goal (TREATAS)
 - Customers who Purchased
 - Food % to Goal
 - Food Goal (TREATAS)
 - Merchandise % to Goal
 - Merchandise Goal (TREATAS)
 - Number of Employees (CROSSFILTER)
 - Quantity Sold (USERELATIONSHIP)
 - Wholesale Cost
- Scalar Measures
- Table & Filter Functions
- Tips & Best Practices
- ADDCOLUMNS Demo
- Calendar
 - Date Format (yyyy-mm-dd)
 - Date_ID
 - NUMBER
 - Month_ID
 - Month_Name
 - Quarter & Year
 - Quarter_ID
 - Quarter_Name
 - Transaction Date



Key objective 3 is now complete. Add the above measures to a matrix broken down by store & target months.

The screenshot shows the Power BI Desktop interface with a report containing two tables and a visualizations pane.

Table 1:

Year_ID	Quantity Sold (USERELATIONSHIP)	SUM Quantity Sold	store_id
All			
2019	21,045	21,820	
January	4,843	4,853	
February	4,305	4,274	
March	5,449	4,935	
April	6,445	7,758	
Total	21,045	21,820	

Table 2:

store_id	Bean Goal (TREATAS)	Bean % to Goal	Beverage % to Goal	Beverage Goal (TREATAS)	Food Goal (TREATAS)	Food % to Goal	Merchandise Goal (TREATAS)	Merchandise % to Goal
3	479	51.01%	49.99%	28311	3159	50.28%	133	49.68%
March	211	94.62%	97.66%	12703	1595	97.43%	53	79.10%
April	268	85.62%	78.04%	15608	1964	78.91%	80	85.11%
5	531	51.45%	49.99%	27138	3716	49.40%	172	49.86%
March	254	109.96%	101.73%	12451	1696	100.77%	81	106.55%
April	277	76.52%	76.17%	14687	2020	74.98%	91	72.80%
8	683	49.93%	49.79%	27485	3606	49.83%	81	50.50%
March	306	97.45%	99.72%	12474	1633	99.33%	17	60.71%
Total	1693	50.70%	49.88%	82934	10681	49.83%	356	49.65%

Visualizations pane:

- Fields:** year, store_id, Month_Name
- Filters:** Year_ID (2019 selected)
- Row:** store_id, Month_Name
- Values:** Bean Goal (TREATAS), Bean % to Goal, Beverage % to Goal, Beverage Goal (TREATAS), Food Goal (TREATAS), Food % to Goal, Merchandise Goal (TREATAS), Merchandise % to Goal
- Drill through:** Cross-report, Off —, On —, Add drill-through fields here

Actions:

- 5) Perform the below actions.
 - a) Add a new matrix to the report.
 - b) Add the month name field to the rows.
 - c) Add store id field to rows.
 - d) Add all the Goal & % to Goal measures to the matrix.
 - e) Add a slicer filter for 2019 using the year id field.

9) Iterator Functions

Introduction - Iterator Functions



Iterator functions allow you to loop through the same expression on every row of a table in order to evaluate a single scalar value (*i.e. max, min, average*) or derive a new table

TOPICS WE'LL COVER:

Iterator Review

Iterator Cardinality

CONCATENATEX

AVERAGEX

RANKX

COMMON USE CASES:

- Aggregating a column into a single value (*i.e. average customer age, max product price, count of orders, etc.*)
- Returning a table of data (*i.e. ADDCOLUMNS and SELECTCOLUMNS*)

Review: Basic Iterators

Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEGEX

RANKX

SUMX()

Returns the sum of an expression evaluated for each row in a table

=SUMX(Table, Expression)

Aggregation to apply to
calculated rows

Examples:

- SUMX
- COUNTX
- AVERAGEGEX
- RANKX
- MAXX/MINX

Table in which the expression will be
evaluated

Examples:

- 'Sales by Store'
- FILTER(Sales,
RELATED(
'Products'[Category])="Clothing")

Expression to be evaluated for
each row of the given table

Examples:

- [Total Orders]
- 'Sales'[RetailPrice] * 'Sales'[Quantity]



PRO TIP:

Imagine the function **adding a temporary new column** to the table, calculating the value in each row (based on the expression) and then applying the aggregation to that new column

Iterator Cardinality

Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEX

RANKX

Iterator cardinality is the number of rows in the table(s) being iterated; the more unique rows, the *higher* the cardinality (*this is different from relationship cardinality*)

Sales by Store				
Index	Date	Price	Product ID	Quantity
1	1/1/2017	\$18	1	2
2	1/1/2017	\$18	2	6
3	1/1/2017	\$14.75	3	1
...
912,422	4/30/2019	\$10	10	1
912,423	4/30/2019	\$3.75	79	2
912,424	4/30/2019	\$3.75	36	1

When iterators reference a single table, cardinality is simply the number of unique rows (*in this case 912,424*)

When using nested iterators, cardinality depends on whether you are using **physical or virtual** relationships:

- For **physical relationships**, cardinality is defined as the max number of unique rows in the *largest* table
- For **virtual relationships**, cardinality is defined as the number of unique rows in each table *multiplied together*

HEY THIS IS IMPORTANT!

When using nested iterators, only the *innermost "X"* function can be optimized by the DAX engine. Nested iterators aren't always a bad thing, but they can have significant performance implications



Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEEX

RANKX

```
1 Customer Sales 2 =  
2 SUMX(  
3     'Product Lookup',  
4     SUMX(  
5         RELATEDTABLE( 'Sales by Store'),  
6         'Sales by Store'[Unit_Price] * 'Sales by Store'[Quantity_Sold]  
7     )  
8 )
```

Physical relationship

- Relationship between 'Product Lookup' and 'Sales by Store'
- 'Product Lookup' contains **88 rows**
- 'Sales by Store' contains **907,841 rows**

Cardinality = **907,841**

```
1 Customer Sales 3 =  
2 SUMX(  
3     VALUES('Product Lookup'),  
4     SUMX(  
5         'Sales by Store',  
6         IF(  
7             'Product Lookup'[product_id] = 'Sales by Store'[Product_ID],  
8             'Sales by Store'[Quantity_Sold] * 'Sales by Store'[Unit_Price],  
9             "-"  
10        )  
11    )  
12 )
```

Virtual relationship

- No physical relationship between 'Product Lookup' and 'Sales by Store'
- 'Product Lookup' contains **88 rows**
- 'Sales by Store' contains **907,841 rows**

Cardinality = **79,890,008**

CONCATENATEX

Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEX

RANKX

CONCATENATEX()

Evaluates an expression for each row of the table and returns the concatenation of those values in a single string, separated by a delimiter

=CONCATENATEX(Table, Expression, [Delimiter], [OrderBy_Expression], [Order])

Table or table expression that contains the rows you want to return

Examples:

- 'Product Lookup'
- **CONCATENATEX(**
VALUES('Employee Lookup')...

Column that contains values to concatenate or an expression that returns a value

Examples:

- 'Product'[Category]
- 'Employee'[Name]
- [Customer Sales]
- 7

Optional arguments:

- **Delimiter:** Used with concatenated expression
 - Examples: ;, "& ", " ; " etc.
- **OrderBy Expression:** Expression used to sort the table
 - Examples: Product Lookup[Product Category]
- **Order:** Order of results applied
 - Examples: ASC, DESC

Maven Readers - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Employee Full Name... Data category Uncategorized

Home table Measure Table

Structure Formatting Properties Calculations

2) Format here defaults to text.

1 Employee Full Name (CONCATENATEX) =
2 CONCATENATEX(
3 'Employee Lookup',
4 'Employee Lookup'[first_name] & " " & 'Employee Lookup'[last_name],
5 " ",
6 'Employee Lookup'[first_name],
7 ASC
8)

Select or drag fields from the Fields pane onto the report canvas.

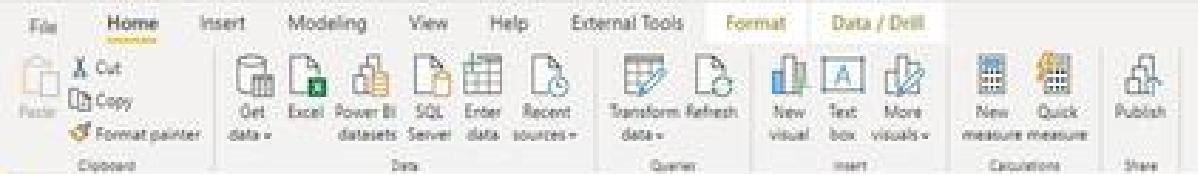
1) Look at example where we concatenate employee first name & there last name first thing to do is to create a new measure call it Employee Full Name (CONCATENATEX).

Visualizations Fields

Measure Table Advanced CALCULATE Calculated Table Joins Employee F... Iterators Relationship Functions Scalar Measures Table & Filter Functions Tips & Best Practices ADDCOLUMNS D... first_name last_name location Manager N... position staff_id start_date Average Days Calendar CROSSJOIN Demo Customer Lookup DATATABLE Demo Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES... New Employees (L... Product Lookup Repeat Customer...

Properties Calculations

Page 8 of 8



Employee Full Name (CONCATENATEX)
Adam.Songs,Adrian.May,Anthony.Evelyn,Aline
Melanie,Alice,Lysandra,Amela
Chadwick,Anthony,Kaitlin,Berk,Derek,Blythe
Arsenio,Brent,Herman,Britanni,Jorden,Cairo
Vaughn,Caldwell,Veda,Chelesa,Claudia,Clare
Oscar,Coby,Shelly,Dale,Joshua,Damon
Sasha,Daren,Xu,Depree,Ariana,Ezekiel
Rashad,Ezekiel,Griffin,Ezekiel,Bertha,Garrett
Doris,Gemma,Eaton,Hamilton,Emi,Hop
Bianca,Hope,Sheila,Ima,Winfred,Jamie
Toast,Jean,LeBean,Joelle,Christen,Joseph
Byron,Karen,Cuppi,Kelsey,Cameron,Kylie
Candace,Lawrence,Roberts,Lisa,Latte,Mark
Brewer,Marry,Dennis,Melinda,Zeus,Melodie
Mercedes,Orson,Benedict,Pandora,Neville,Peter
Paloma,Quail,Octavia,Remedios,Mari,Ronan
Magee,Ruth,Leslie,Sawyer,Casper,Tamekah
Maya,Tatum,Laurel,Vance,Samuel,Xavier
Zachary,Yair,Lilith

3) Add the Employee Full Name (CONCATENATEX) measure to a table.

Visualizations Fields

Filters

Measure Table

Advanced CALCULATE

Calculated Table Joins

Employee Full Name (CONCATENATEX)

Iterators

Relationship Functions

Scalar Measures

Table & Filter Functions

Tips & Best Practices

ADDCOLUMNS Demo

- first_name
- last_name
- location
- Manager Name & Store
- position
- staff_id
- start_date

Average Days

Calendar

CROSSJOIN Demo

Customer Lookup

DATATABLE Demo

Employee Lookup

EXCEPT Demo

Food Inventory

GENDERSERIES Demo

New Employees (INTERSECT)

Product Lookup

Repeat Customer Sales (INTERSECT Assignm...

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

You will notice CONCATENATEX has created just one list of all of our employees. This output is not really useful. What would be beneficial is to look at this by perhaps by employee id.

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More New measure measure Quick measure Publish

Queries Data Calculations Share

Visualizations Fields

Filters

Count of staff_id Employee Full Name (CONCATENATE)

55 Adam Song;Adrian Macon;Ansley Evelyn;Aline Melania;Alisa Tysandra;Amala Arsenio;Brent Herman;Britanni Jordyn;Cairo Vaughan;Calidwell Veda;Chelsea Ci Sasha;Darren Xu;Desiree Anita;Ezekiel Rashad;Ezekiel Griffin;Ezekiel Bertha;Gur Sheila;Imra Winifred;Jamie Toast;Jean LeJean;Joelle Christen;Joseph Byron;Kare Roberts;Lisa Lutte;Mark Brewer;Marny Dennis;Melinda Zeus;Melodie Mercedes;Octavia Remedios;Man Ronan Magee;Ruth Leslie Sawyer;Kasper;Tamekah May

Remove field Rename Move Conditional formatting Remove conditional formatting Don't summarize Sum Average Minimum Maximum Count (Distinct) Count Standard deviation Variance Median Show value as New quick measure SELECTCOLUMNS Demo Employee ID

4) Add the Staff id field to the table & amend options to Don't summarise.

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Dataset Data Queries Calculations Share

Visualizations Fields

employee

Measure Table

- Employee Full Name (CONCATENATE)

Iterations

- Sales by Employee Name (CONCATENATE)

Relationship Functions

- Number of Employees (CROSSFILTER)

Scalar Measures

- Total Employees

Employee Lookup

- first_name
- last_name
- location
- position
- staff_id
- start_date

New Employees (INTERSECT)

- first_name
- last_name
- location
- position
- Σ Revenue
- Σ staff_id
- Σ start_date

SELECTCOLUMNS Demo

- Σ Employee ID

staff_id Employee Full Name (CONCATENATE)

- 1 Mark Brewer
- 2 Jean LeBeau
- 3 Jamie Tost
- 4 Chelsea Claudia
- 5 Adam Songs
- 6 Karen Cupps
- 7 Kelsey Cameron
- 8 Hamilton Emi
- 9 Caldwell Veda
- 10 Ima Winifred

Total: Adam Songs, Adrian Macon, Ainsley Evelyn, Aline Melanie, Alisa Lysandra, Amela Chadwick, Anthony Kaitlin, Berk Derek, Blythe Arsenio, Brent Herman, Britanni Joey Vaughan, Caldwell Veda, Chelsea Claudia, Clare Oscar, Coby Shelly, Dale Joshua, Dan Sasha, Darren Xu, Desiree Anika, Ezekiel Rashad, Ezekiel Griffin, Ezekiel Bertha, Garn Doris, Gemma Eaton, Hamilton Emi, Hop Blanca, Hope Sheila, Ima Winifred, Jamie T LeBeau, Joelle Christen, Joseph Byron, Karen Cupps, Kelsey Cameron, Kylie Candace, Lawrence Roberts, Luca Latte, Mark Brewer, Marny Dennis, Melinda Zeus, N Mercedes, Orson Benedict, Pandora Neville, Peter Paloma, Quail Octavia, Remedios Mari, Ronan Magee, Ruth Leslie, Sawyer Kasper, Tamekah Maya, Tatsumi Laurel, Vanox Samuel, Xavier Zachary, Yasir Lillith

5) The above output generated from the previous step.

Maven Readers - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Employee Full Name... Data category Uncategorized New measure Quick measure measure

Home table Measure Table Structure Formatting Properties Calculations

Visualizations Fields

P Employee

Measure Table

- Employee Full Name (CONCATENATEX)
- Sales by Employee Name (CONCATENATEX)
- Number of Employees (CROSSFILTER)
- Total Employees

Relationship Functions

- Employee Lookup
- first_name
- last_name
- location
- position
- staff_id
- start_date

Scalar Measures

- New Employees (INTERSECT)
- first_name
- last_name
- location
- position
- Revenue
- staff_id
- start_date

Cross-report

Drill through

Keep all filters

Add drill-through fields here

Employee Full Name (CONCATENATEX) =
IF(
HASONEVALUE(
'Employee Lookup'[first_name]
)
,
CONCATENATEX(
'Employee Lookup',
'Employee Lookup'[first_name] & " " & 'Employee Lookup'[last_name],
,
'Employee Lookup'[first_name],
ASC
,
BLANK()
)

6) Amend the Employee Full Name (CONCATENATEX) measure so that Total row does not output any values using the IF & HASONEVALUE statements.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators +

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get Data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

Openers Data Queries Calculations Share

Visualizations Fields

Filters

Measure Table

Employees Employee Full Name (CONCATENATE)

Sold by Employee Name (CONCATENATE)

Relationship Functions

Number of Employees (CROSSFILTER)

Scalar Measures

Total Employees

Employee Lookup

First_name

Last_name

location

position

staff_id

start_date

New Employees (INTERACT)

first_name

last_name

location

position

E Revenue

E staff_id

E start_date

SELECTCOLUMNS Demo

E Employee ID

7) Here you can see that Total row values displays nothing now we just have employees by first & last names & staff id.

staff_id Employee Full Name (CONCATENATED)

1	Mark Brewer
2	Jean LeBeau
3	Jamie Toast
4	Chelsea Claudia
5	Adam Song
6	Karen Cupps
7	Kelsey Cameron
8	Hamilton Emi
9	Caldwell Veda
10	Ima Winifred
11	Ruth Leslie
12	Britanni Jorden
13	Berk Derek
14	Damon Sasha
15	Remedios Mari
16	Darren Xu
17	Quail Octavia
18	Ezekiel Rashad
19	Peter Paloma
20	Ronan Magee
21	Melodie Mercedes
22	Manny Dennis
23	Blythe Arsenio
24	Garnett Doris
25	Aline Melanie
26	Joelle Christen
	Total

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators

Page 6 of 8

PRO TIP: Adding Dynamic Labels

Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEX

RANKX



Product Category slicer is
filtered to **Coffee, Coffee Beans**
& **Drinking Chocolate**

```
1 Selected Product Category (CONCATENATEX) =  
2 "Showing Sales For: " &  
3 CONCATENATEX(  
4     VALUES(  
5         'Product Lookup'[product_category]  
6     ),  
7     'Product Lookup'[product_category],  
8     ", ",  
9     'Product Lookup'[product_category],  
10    ASC  
11 )
```

The **Selected Product Category** measure
uses **CONCATENATEX** to capture the
selections from the slicer...

store_id	Customer Sales
3	\$732,308.45
Astoria	\$732,308.45
5	\$713,956.00
Lower Manhattan	\$713,956.00
8	\$758,641.75
Hell's Kitchen	\$758,641.75
Total	\$2,204,906.20

Showing Sales For: Coffee, Coffee Beans, Drinking Chocolate

...which can be displayed as a **dynamic label**
using a card to show selected items

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data > Excel Power BI datasets Server Enter data Recent sources Transform Refresh data New visual Text box Insert New measure measure Calculations Publish

Queries

product_category

- Bakery
- Branded
- Coffee
- Coffee beans
- Drinking Chocolate
- Flavours
- Loose Tea
- Packaged Chocolate
- Tea

store_id Customer Sales

store_id	Customer Sales
3	\$774,596.30
Long Island City	\$774,596.30
5	\$755,401.95
New York	\$755,401.95
8	\$802,226.35
New York	\$802,226.35
Total	\$2,132,218.60

Visualizations Fields

product_category

CROSSJOIN Demo

product_category

Product Looker

product_category

Rows: store_id, storeCity

Columns: Add data fields here

Values: Customer Sales

Drill through: Off

Cross-report: Off

Keep all filters: On

Add drill-through fields here

Lets say we want to add in a slicer for product category update slicer to just show Coffee, Coffee beans & Drinking Chocolate. We want to create a more intuitive way of for report consumers to totally understand that the visual is also being filtered by three product categories.

1) Perform the below actions.
a) Add store id & store city fields to rows within the table.
b) Add Customer Sales to the values within the table.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators +

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Selected Product C... Data category: Uncategorized

Home table: Measure Table New Quick measure measure

Structure Formatting Properties Calculations

1 Selected Product Category (CONCATENATEX) =
2 "Showing Sales For: " &
3 CONCATENATEX(
4 VALUES(
5 'Product Lookup'[product_category]
6),
7 'Product Lookup'[product_category],
8 ", ",
9 'Product Lookup'[product_category],
10 ASC
11)

Visualizations Fields

Selected

Measure Table

Selected Product Category (CONCATENATEX)

Table & Filter Functions

% of Total Baked (ALL SELECTED)
% of Total Sold (ALL SELECTED)
Customer Sales (ALL SELECTED)
Quantity Sold (SELECTED VALUE)
Retail Price (SELECTED VALUE)

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

1) Create a new measure call it Selected Product Category (CONCATENATEX)

2) Format here defaults to text.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators +

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets - SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals

New measure Quick measure Publish

product_category

- Bakery
- Branded
- Coffee**
- Coffee beans**
- Drinking Chocolate**
- Flavours
- Lovely Tea
- Packaged Chocolate
- Tea

store_id Customer Sales

store_id	Customer Sales
3	\$774,596.30
Long Island City	\$774,596.30
5	\$755,401.95
New York	\$755,401.95
8	\$802,220.35
New York	\$802,220.35
Total	\$2,332,218.60

Select or drag fields to populate this visual.

3) Add a card visual to the report.

Visualizations

Fields

Selected

Measure Table

Iterations

- Selected Product Category (CONCATENATED)**

Table & Filter Functions

- % of Total Baked (ALL) **SELECTED**
- % of Total Sold (ALL) **SELECTED**
- Customer Sales (ALL) **SELECTED**
- Quantity Sold **SELECTED VALUE**
- Retail Price **SELECTED VALUE**

Add data fields here

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Tips & Best Practices

Scalar Functions

Advanced CALCULATE

Table & Filter Functions

Joins

Relationship Functions

Iterators

Duplicate of Iterators

Page 6 of 8

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

Queries Calculations Share

product_category

- Bakery
- Branded
- Coffee**
- Coffee beans**
- Drinking Chocolate**
- Flavours
- Loose Tea
- Packaged Chocolate
- Tea

store_id Customer Sales

store_id	Customer Sales
3	\$774,596.30
Long Island City	\$774,596.30
5	\$751,401.95
New York	\$751,401.95
8	\$802,220.35
New York	\$802,220.35
Total	\$2,332,218.60

Showing Sales For: Coffee, Coffee beans, Drinking Choc...

Selected Product Category (CONCATENATEX)

4) Add Selected Product Category (CONCATENATEX) to the card huge label is now displayed.

Visualizations Fields

Selected

Measure Table Iterators Selected Product Category (CONCATENATEX)

Table & Filter Functions % of Total Baked (ALL) % of Total Sold (ALL) Customer Sales (ALL) Quantity Sold (SELECTEDVALUE) Retail Price (SELECTEDVALUE)

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators +

Page 8 of 8

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

Queries Calculations Share

product_category

- Bakery
- Branded
- Coffee**
- Coffee beans**
- Drinking Chocolate**
- Flavours
- Loose Tea
- Packaged Chocolate
- Tea

store_id Customer Sales

store_id	Customer Sales
3	\$774,596.30
Long Island City	\$774,596.30
5	\$755,401.95
New York	\$755,401.95
8	\$802,220.35
New York	\$802,220.35
Total	\$2,332,218.60

Showing Sales For: Coffee, Coffee beans, Drinking Chocolate
Selected Product Category (CONCATENATEX)

Visualizations Fields

Selected

Measure Table Iterators Selected Product Category (CONCATENATED) Table & Filter Functions % of Total Baked (ALL) % of Total Sold (ALL) Customer Sales (ALL) Quantity Sold RETAIL PRICE

Color: Display units: Auto Value decimal places: Auto Text size: 16 pt Font family: DIN Source spacing: On Revert to default

Category label: On Word wrap: On Title: Off

5) Click here to format the card accordingly i.e. font size etc.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators

Page 8 of 8

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

Queries

Calculated measures

Share

product_category

- Bakery
- Branded
- Coffee
- Coffee beans
- Drinking chocolate
- Flavours
- Loose Tea
- Packaged Chocolate
- Tea

store_id Customer Sales

store_id	Customer Sales
3	\$33,614.00
Long Island City	\$33,614.00
5	\$39,372.00
New York	\$39,372.00
8	\$10,798.00
New York	\$10,798.00
Total	\$83,784.00

Showing Sales For: Branded
Selected Product Category (CONCATENATEX)

Visualizations Fields

Selected

Measure Table

Relations

- Selected Product Category (CONCATENATEX)

Table & Filter Functions

- % of Total Baked (ALL) **SELECTED**
- % of Total Sold (ALL) **SELECTED**
- Customer Sales (ALL) **SELECTED**
- Quantity Sold **SELECTED VALUE**
- Retail Price **SELECTED VALUE**

Slicer header

Items

Title

Background

Lock aspect

Border

Shadow

Visual header

Tips & Best Practices

Scalar Functions

Advanced CALCULATE

Table & Filter Functions

Joins

Relationship Functions

Iterators

Duplicate of Iterators

Page 8 of 8

ASSIGNMENT: CONCATENATEX

KEY OBJECTIVES

- 1) Create a visual for Store 5 that shows **customer sales** by **store & employee id**.
- 2) Create a measure to show **% of Total Sales** for Store 5.
- 3) Use **CONCATENATEX** to define a measure that shows the employee name(s) selected & the % of Total Sales for Store 5.

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box Insert More visuals New measure measure Publish

Queries Calculations Share

Customer Sales

store_id: Customer Sales
1 \$1,409,796.81
Total: \$1,409,796.81

Filters

Search

Filters on this visual

Customer Sales
is (All)

store_id
is 5

Filter type: Basic filtering

Select all

3 308817

3 291658

8 107136

Require single selection

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Fields

Selected

Measure Table

Iterators

Selected Product Category (CONCATENATED) **SELECTED**

Table & Filter Functions

% of Total Baked (All) **SELECTED**

% of Total Sold (All) **SELECTED**

Customer Sales (All) **SELECTED**

Quantity Sold **SELECTED VALUE**

Retail Price **SELECTED VALUE**

Customer Sales

Add drill-through fields here

Add data fields here

Add data fields here

ASSIGNMENT: CONCATENATEX

1) Objective 1 to create a visual for Store 5 that shows customer sales by store & employee id. Select the matrix above, click on the filters go to store id select the Basic filtering option filter to 5.

Page 9 of 9

Maven Readers - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Out data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box Insert More visuals New measure measure Calculations Publish

Queries

Share

Customer Sales

store_id	Customer Sales
0	\$1,409,796.51
1	\$14,425.05
12	\$142,120.30
14	\$20,219.56
21	\$125,117.81
26	\$113,660.40
27	\$112,714.49
28	\$162,208.80
29	\$115,399.40
30	\$204,348.60
Total	\$5,409,796.51

Filters

Search

Customer Sales is (All)

staff_id is (All)

store_id is 5

Filter type: Basic filtering

Select all

1 108817

1 291888

8 307136

Require single selection

Add data fields here

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Visualizations

Fields

staff_id

ADOCOLUMNS Demo

Employee Lookup

New Employees (INTERSECT)

Sales by Store

Customer Sales

Notes

Scalar Functions

Advanced CALCULATE

Table & Filter Functions

Join

Relationship Functions

Iterators

Duplicate of Iterators

ASSIGNMENT: CONCATENATEX

Page 9 of 9

2) Add in the staff id field to the matrix.

Key objective 1 is now complete. Create a visual for Store 5 that shows customer sales by store & employee id.

Maven Readers - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name % of Customer Sales
Home table Measure Table

Format Percentage \$ - % , . ;

Data category Uncategorized New Quick measure measure Properties Calculations

1 % of Customer Sales (CONCATENATEX Assignment) =
2 VAR AllExceptSales =
3 CALCULATE(
4 [Customer Sales],
5 ALLEXCEPT(
6 'Sales by Store',
7 'Sales by Store'[store_id]
8)
9)
10 VAR Ratio =
11 DIVIDE(
12 [Customer Sales],
13 AllExceptSales,
14 BLANK()
15)
16 RETURN
17 Ratio

Visualizations Fields
Measure Table
Iterations
% of Customer Sales (CONCATENATEX Assignment)

Row
store_id
staff_id

Columns
Add data fields here

Values
Customer Sales

Drill through
Cross-report
On Off
Keep all filters
On Off
Add drill-through fields here

Filters on all pages
Add data fields here

ASSIGNMENT: CONCATENATEX

Key objective 2 is now complete. Create a measure to show % of Total Sales for Store 5.

2) Change the formatting accordingly over here.

1) Objective 2 to create a measure to show % of Total Sales for Store 5. The best way to do this is to create a measure go over to the report view & create the "% of Customer Sales (CONCATENATEX Assignment)" measure.

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Sales by Employee... Data category Uncategorized New measure Quick measure measure

Home table Measure Table Structure Properties Calculations

1 Sales by Employee Name (CONCATENATEX) =
2 IF(
3 HASONEVALUE(
4 'Employee Lookup'[first_name]
5),
6 "Employee: " &
7 CONCATENATEX(
8 VALUES(
9 'Employee Lookup'[first_name]
10),
11 'Employee Lookup'[first_name] &
12 "-" &
13 FORMAT([% of Customer Sales (CONCATENATEX Assignment)], "Percent"),
14 ", ",
15 'Employee Lookup'[first_name],
16 ASC
17),
18 "Select a Single Employee"
19)

2) Change the formatting accordingly over here.

Visualizations Fields

sales by

Measure Table Iterators

Sales by Employee Name (CONCATENATEX)

Values Add data fields here

Drill through

Cross-import

Off —

Keep all filters

On —

Add drill-through fields here

customer_id
item_id
line_item_id
order
Product Group
product_id
promo_item_id
quantity_id
staff_id
store_id
Time Group
Total Revenue
transaction_date
transaction_id
transaction_time
unit_price

ASSIGNMENT: CONCATENATEX

Page 9 of 9



Key objective 3 is now complete. Use CONCATENATEX to define a measure that shows the employee name(s) selected & the % of Total Sales for Store 5.

Customer Sales	
store_id	Customer Sales
1	123456789
2	123456789
3	123456789
4	123456789
5	123456789
6	123456789
7	123456789
8	123456789
9	123456789
10	123456789
11	123456789
12	123456789
13	123456789
14	123456789
Total	123,456,789.01

When different store id's are selected the label within the card below will be updated dynamically.

**Employee:
Damon - 1.43%**

Sales by Employee Name (CONCATENATED)

3) Perform the below actions.

- Add a card to the report.
- Within the card add in the Sales by Employee Name (CONCATENATEX) field.

Filters

Search

Filters on this visual

- Customer Sales is (All)
- staff_id is (All)
- store_id is 5

Filter type

Select all:

- 1 108817
- 2 291888
- 3 307136

Require single selection

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Visualizations

Grid

Column headers

Row headers

Values

Subtotals

Grand total

Field formatting

Conditional formatting

Title

Background

Lock aspect

Border

Shadow

Tooltip

Visual header

Fields

sales by

Measure Table

Iterators

Employee name (CONCATENATED)

Sales by Store

- customer_id
- store_id
- line_item_id
- order
- Product Group
- product_id
- promo_item_id
- quantity_sold
- staff_id
- store_id
- Time Group
- Total Revenue
- transaction_date
- transaction_id
- transaction_time
- unit_price

ASSIGNMENT: CONCATENATEX

AVERAGEX

Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEX

RANKX

AVERAGEX()

Calculates the average (arithmetic mean) of a set of expressions evaluated over a table

=AVERAGEX(Table, Expression)

Table, or table expression, that contains the rows to evaluate

Examples:

- 'Calendar'
- 'Product Lookup'

The expression that you want to evaluate

Examples:

- [Customer Sales]
- SUM[quantity_sold]

HEY THIS IS IMPORTANT!

AVERAGE & AVERAGEX do **NOT** count days with zero sales when computing an average. To evaluate an average over a date range that includes dates with no sales, use **DIVIDE & COUNTROWS** instead

Maven Readers - Analysis - Iterators - Power BI Desktop Sign in

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Average Daily Sales... Format Currency Data category Uncategorized New Quick measure measure

Home table Measure Table \$ - % , . : Structure Formatting Properties Calculations

1 Average Daily Sales (AVERAGEX) =
2 AVERAGEX(
3 'Calendar',
4 [Customer Sales]
5)

Employee: Britanni-20.01%
Sales by Employee Name (CONCATENATE)

2) Change the formatting accordingly over here.

Visualizations Fields

Average

Measure Table Iterators

- Average Daily Sales (AVERAGE)
- Average Profit (AVERAGE)
- Moving Average (AVERAGE)
- Moving Average Profit (AVERAGE)

Relationship Functions

- Average Order Value (CVP)

Average Days

- Average Days
- Average Days Value

Add data fields here

Filter on all pages Add drill-through fields here

On —— On ——

Keep all filters

On ——

Add data fields here

On ——

Add drill-through fields here

1) Create the Average Daily Sales (AVERAGE) measure.

Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMNT: CONCATENATE

The screenshot shows the Microsoft Power BI desktop interface. On the left, there is a card visual titled "Customer Sales" showing a list of items with their values, and a table visual titled "Sales by Employee Name" showing monthly sales data. The table has columns for Year_ID, Customer Sales, and Average Daily Sales (AVERAGE). A blue arrow points from the text below to the table visual.

Select a Single Employee

Sales by Employee Name (CONCATENATEX)

Year_ID	Customer Sales	Average Daily Sales (AVERAGE)
2017	\$1,678,074.11	\$4,597.46
January	\$31,845.99	\$2,640.16
February	\$76,273.99	\$2,724.07
March	\$96,154.43	\$3,198.83
April	\$118,328.21	\$3,871.26
May	\$157,208.99	\$3,871.26
June	\$168,891.73	\$3,543.33
July	\$157,998.53	\$3,295.76
August	\$154,493.33	\$4,863.45
September	\$149,821.19	\$4,863.71
October	\$148,119.44	\$4,458.67
Total	\$4,252,704.88	\$3,099.09

3) Now we will create a matrix to be able to look at the monthly total sales & the average daily sales.

- Add a matrix to the report.
- Add the year id field to the rows.
- Add month name field to the rows.
- Add the Average Daily Sales (AVERAGE) measure to the values.
- Add the customer sales measure to the values.

Filters

Search:

Filters on this visual

- Average Daily Sales (AVERAGE) is (All)
- Customer Sales is (All)
- Month_Name is (All)
- Year_ID is (All)

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Visualizations

Matrix

Fields

customer sales

- Measure Table
- Iterations
 - % of Customer Sales (CONCATENATEX Assignment)
 - Rank of Customer Sales (RANKX)
 - Rank of Rounded Customer Sales (RANKX)
 - Rounded Customer Sales
- Scalar Measures
 - Customer Sales (V) (COALESCE)
- Table & Filter Functions
 - Customer Sales (ALLEXCEPT Assignment)
 - Customer Sales (ALLEXCEPT)
 - Customer Sales (ALLSELECTED)
- Tips & Best Practices
 - Customer Sales
 - Average Daily Sales (AVERAGE)
 - Customer Sales (Last Year)
 - Customer Sales (V) (ISBLANK)
- Drill through
- Customer Lookup
 - Customer Sales (High/Low)
- EXCEPT Demo
- Customer Sales (High/Low)
- Repeat Customer Sales (INTERSECT Assignment)
 - customer_id
 - Profit
 - Revenue
- Row Demo
 - Customer Sales

3) Now we will create a matrix to be able to look at the monthly total sales & the average daily sales.

- a) Add a matrix to the report.
 - b) Add the year id field to the rows.
 - c) Add month name field to the rows.
 - d) Add the Average Daily Sales (AVERAGE) measure to the values.
 - e) Add the customer sales measure to the values.

PRO TIP: Moving Averages

Iterator Review

Iterator Cardinality

CONCATENATEX

AVERAGEX

RANKX

```
1 AVERAGEX Sales =
2 VAR LastTransactionDate = MAX('Calendar'[Transaction_Date])
3 VAR AverageDays = 30
4 VAR PeriodInVisual =
5 FILTER(
6   ALL(
7     'Calendar'[Transaction_Date]
8   ),
9   AND(
10    'Calendar'[Transaction_Date] > LastTransactionDate - AverageDays,
11    'Calendar'[Transaction_Date] <= LastTransactionDate
12  )
13 )
14 VAR Output =
15 CALCULATE(
16   AVERAGEX(
17     'Calendar',
18     [Customer Sales]
19   ),
20   PeriodInVisual
21 )
22 RETURN
23 Output
```

Here we're using **MAX**, **FILTER** & **ALL** to create a 30-day rolling time period (*based on the latest transaction date*)

Once the rolling window is defined, we can use **AVERAGEX** to calculate the average sales over that period

- Note that we could also use **SUMX** or **COUNTX** to calculate the rolling total/count



PRO TIP:

Use a **parameter** to create a dynamic, user-defined period!

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Moving Average (A...) Format Currency Data category Uncategorized New Quick measure measure

Home table Measure Table \$ - % , . : Properties Calculations

1 Moving Average (AVERAGEX) =
2 VAR LastTransactionDate = MAX('Calendar'[Transaction_Date])
3 VAR AverageDay = 30
4 VAR PeriodInVisual =
5 FILTER(
6 ALL(
7 'Calendar'[Transaction_Date]
8),
9 AND(
10 'Calendar'[Transaction_Date] > LastTransactionDate - AverageDay,
11 'Calendar'[Transaction_Date] <= LastTransactionDate
12)
13)
14 VAR OutPut =
15 CALCULATE(
16 AVERAGEX(
17 'Calendar',
18 [Customer Sales]
19),
20 PeriodInVisual
21)
22 RETURN
23 OutPut

2) Change the formatting accordingly over here.

Visualizations Fields

Add data fields here

Values Drill through

Keep all filters Cross-report

On — Add drill-through fields here

Iterations

- Average Daily Sales (AVERAGEX)
- Average Profit (AVERAGEX)
- Moving Average (AVERAGEX)
- Moving Average Profit (AVERAGEX)

Relationship Functions

- Average Order Value (CVP)

Average Days

- Average Days
- Average Days Value

Icons Scalar Functions Advanced CALCULATE Table & Filter Functions Join Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATEX

Page 9 of 9

Maven Readers - Analysis - Iterators - Power BI Desktop

Sign in

File **Home** **Insert** **Modeling** **View** **Help** **External Tools** **Format** **Data / Drill**

Cut Copy Format painter **Get data** Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box Insert New measure measure Quick measure Publish

Customer Sales

1	\$1,469,798.51
2	\$34,405.05
3	\$32,120.35
4	\$20,215.56
5	\$20,317.81
6	\$113,960.85
7	\$102,714.69
8	\$102,998.83
9	\$133,399.64
10	\$208,946.04
Total	\$1,469,798.51

Select a Single Employee
Sales by Employee Name (CONCATENATEX)

Customer Sales Average Daily Sales AVERAGEX

Year	Customer Sales	Average Daily Sales
2017	\$1,478,074.11	\$4,597.44
January	\$91,640.09	\$2,460.16
February	\$74,273.89	\$2,224.07
March	\$99,154.43	\$3,165.00
April	\$119,358.01	\$3,171.97
May	\$117,208.89	\$3,071.21
June	\$156,099.71	\$3,983.33
July	\$157,968.53	\$3,985.71
August	\$154,403.32	\$4,003.40
Total	\$4,252,764.88	\$5,009.68

Customer Sales and Moving Average (AVERAGEX) by Transaction Date

Customer Sales Moving Average (AVERAGEX)

Transaction Date: 01/01/2018
Moving Average (AVERAGEX): \$5,054.87

Filters

Search:

Filters on this visual:

- Customer Sales (All)
- Moving Average (AVERAGEX) (All)
- Transaction Date (All)

Shared axis: Transaction Date

Column series: Add data fields here

Column values: Customer Sales

Line values: Moving Average (AVERAGEX)

Filters on all pages: Add data fields here

Tooltips: Add data fields here

Drill through: Off

Cross-report: Off

Keep all filters: On

Add drill-through fields here

Visualizations

Fields

moving

Measure Table

Iterators

- Moving Average (AVERAGEX) (Selected)
- Moving Average (AVERAGEX)

ASSIGNMENT: CONCATENATEX

Page 9 of 9

ASSIGNMENT: Moving Averages

KEY OBJECTIVES

- 1) Create a matrix to show total & daily average profit by month, for Jan 2018 — Apr 2019.
- 2) Use **GENERATESERIES** & **SELECTEDVALUE** to create a parameter with increments of 7 days over a 9-week period.
- 3) Create a measure using **AVERAGEX** & the parameter you defined to calculate the moving average profit.

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Power Get Data Excel Power BI SQL Enter Recent Transform Refresh New Quick Publish

Datasets Server data sources data Query New measure measure Text box More visuals Calculations Share

store_id Customer Sales

store_id	Customer Sales
5	\$1,469,796.51
7	\$34,405.00
12	\$282,120.88
14	\$20,215.56
25	\$20,517.81
26	\$913,665.83
27	\$182,714.89
28	\$162,008.83
29	\$185,399.66
30	\$208,548.66
Total	\$1,469,796.51

Customer Sales and Moving Average (AVERAGEX) by Transaction_Date

Select a Single Employee

Sales by Employee Name (CONCATENATEX)

Year_ID Customer Sales Average Daily Sales (AVERAGEX)

Year_ID	Customer Sales	Average Daily Sales (AVERAGEX)
2017	\$1,678,074.11	\$4,597.46
January	\$81,845.09	\$2,640.16
February	\$76,273.99	\$2,724.07
March	\$99,154.43	\$3,198.53
April	\$119,309.01	\$3,976.97
May	\$137,208.99	\$3,071.26
June	\$166,899.73	\$5,563.33
July	\$157,968.53	\$5,095.76
August	\$154,493.32	\$4,903.40
Total	\$4,252,754.88	\$5,009.68

Year_ID Profit

Year_ID	Profit
2017	\$1,242,522.69
January	\$60,662.93
February	\$56,449.75
March	\$73,373.74
April	\$68,264.99
May	\$116,370.94
June	\$123,642.33
July	\$116,991.33
August	\$114,384.04
September	\$107,986.97
October	\$176,637.68
Total	\$3,148,872.56

Visualizations

Fields

profit

- Measure Table**
 - Store 3 Profit
 - Store 3 Profit (KEEPFILTERS)
 - Total Profit
- Calculated Table Joins**
 - Profit (INTERSECT Assignment)
- Iterators**
 - Average Profit (AVERAGED)
 - Moving Average Profit (AVERAGEX)
 - Top 5 Products by Profit (RANKED)
- Tip & Best Practices**
 - Profit
- Repeat Customer Sales (INTERSECT Assignment)**
 - Profit

Drill through

Cross-report

On

Keep all filters

On

Add drill-through fields here

ASSIGNMENT: CONCATENATEX

1) Perform the below actions.

- Add a matrix to the report.
- Add the year id field to the rows.
- Add the month name field to the rows.
- Add the profit field to the values.

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Average Profit (AVERAGEX)

Home table Measure Table \$ - % , ; : Data category Uncategorized New Quick measure measure Properties Calculations

1 Average Profit (AVERAGEX) =
 2 AVERAGEX(
 3 'Calendar',
 4 [Profit]
 5)

Total \$1,409,796.51

Select a Single Employee

Sales by Employee Name (CONCATENATEX)

Year_ID Customer Sales Average Daily Sales (AVERAGEX)

Year_ID	Customer Sales	Average Daily Sales (AVERAGEX)
2017	\$1,678,074.11	\$4,597.46
January	\$81,845.09	\$2,640.16
February	\$76,273.99	\$2,724.07
March	\$99,154.43	\$3,196.53
April	\$119,056.01	\$3,976.87
May	\$137,206.99	\$3,671.26
June	\$168,899.76	\$5,563.33
July	\$157,068.53	\$5,095.76
August	\$154,483.32	\$4,903.40
Total	\$4,252,794.88	\$5,009.08

3) Change the formatting accordingly over here.

2) Objective 1 to create a matrix to show total & daily average profit by month, for Jan 2018 — Apr 2019.. Create the Average Profit (AVERAGEX) measure.

Visualizations Fields

average profit

Measure Table Iterators Average Profit (AVERAGEX) Moving Average Profit (AVERAGEX)

Row: Year_ID Month_Name

Columns: Add data fields here

Values: Profit

Drill through: Off — Keep all filters: On — Add drill-through fields here

Table tools

Icons Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATEX

Page 9 of 9

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Out data - Excel Power BI datasets SQL Server Enter data Recent sources

Transform Refresh data - New visual Text box More

New measure measure

Calculations

Publish Share

store_id Customer Sales

6	\$1,469,796.51
7	\$34,405.09
12	\$282,120.38
14	\$20,215.96
25	\$20,517.81
26	\$313,665.85
27	\$182,714.80
28	\$162,008.83
29	\$185,599.66
30	\$208,548.68
Total	\$1,469,796.51

Select a Single Employee

Sales by Employee Name (CONCATENATEX)

Year_ID	Customer Sales	Average Daily Sales (AVERAGEX)
2017	\$1,678,074.11	\$4,597.46
January	\$31,845.09	\$2,640.16
February	\$76,278.99	\$2,724.07
March	\$99,154.43	\$3,198.53
April	\$119,209.01	\$3,076.97
May	\$137,208.99	\$3,071.26
June	\$166,099.78	\$3,563.33
July	\$157,968.53	\$3,095.76
August	\$154,415.32	\$4,003.40
Total	\$4,252,704.88	\$5,009.98

Customer Sales and Moving Average (AVERAGEX) by Transaction_Date

Visualizations

Fields

- average profit
- Measure Table
- Iterators
 - Average Profit (AVERAGEX)
 - Moving Average Profit (AVERAGEX)

Filters

Rows

- Year_ID
- Month_Name

Columns

Add data fields here

Values

- Profit
- Average Profit (AVERAGEX)

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

4) Add the Average Profit (AVERAGEX) measure to the matrix.

Notes **Scalar Functions** **Advanced CALCULATE** **Table & Filter Functions** **Join** **Relationship Functions** **Iterators** **Duplicate of Iterators** **ASSIGNMENT: CONCATENATEX** **+**

Key objective 1 is now complete. Create a matrix to show total & daily average profit by month, for Jan 2018 — Apr 2019.

The screenshot shows the Power BI desktop interface with three visualizations:

- Customer Sales:** A card showing "Customer Sales" with a value of \$854,809.34.
- Customer Sales and Moving Average (AVERAGEX) by Transaction_Date:** A line chart with a blue area showing "Customer Sales" over time from Jan 2018 to Mar 2019, with a moving average line overlaid.
- Sales by Employee Name (CONCATENATEX):** A card showing sales by employee name.

A blue arrow points from the date range selector in the line chart to the date range selector in the matrix visualization below.

Matrix Visualization Data:

Year_ID	Profit	Average Profit (AVG(AVG))
2018	\$1,916,544.75	\$5,250.81
2019	\$658,086.02	\$5,530.13
Total	\$2,574,630.77	\$5,319.49

Matrix Visualization Data (Detailed View):

Month	Profit	Average Profit (AVG(AVG))
January	\$104,615.00	\$3,374.68
February	\$91,885.10	\$3,281.61
March	\$108,171.47	\$3,424.89
April	\$110,925.38	\$3,697.31
May	\$128,293.47	\$4,128.50
June	\$129,046.22	\$4,301.54
July	\$122,195.18	\$3,941.78
August	\$119,321.13	\$3,848.07
September	\$111,393.30	\$3,713.11
October	\$126,664.18	\$4,165.94
Total	\$1,904,349.87	\$5,319.49

Maven Readers - Analysis - Iterators - Power BI Desktop

Sign in

File Home Help External Tools Table tools

Name Average Days

Mark as date table - Manage relationships New measure Quick New measure column New table

Structure Calculations

1 Average Days =
2 GENERATESERIES(
3 7,
4 63,
5 7
6)

Fields

Search

Measure Table ADDCOLUMNS Demo Average Days Average Days Value Calendar CROSSJOIN Demo Customer Lookup DATATABLE Demo Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo New Employees (INTERSECT) Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store SELECTCOLUMNS Demo Store Lookup SUMMARIZE Demo Table Constructor Demo Target Sales - April 2019 (DATATABLE) Target Sales - March 2019 (DATATABLE) TREATAS Demo UNION Demo Unsold Pastries VALUES Example

6) Objective 2 to use GENERATESERIES & SELECTEDVALUE to create a parameter with increments of 7 days over a 9-week period. Create a new table & call it Average Days.

Series created for 9 weeks.

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Help External Tools Table tools Measure tools

Name Average Days Value

Home table Average Days

Format Whole number

Data category Uncategorized

New Quick measure measure Calculations Properties

1 Average Days Value =
2 SELECTEDVALUE(
3 'Average Days' [Average Days],
4 30
5)

Average Days

7
14
21
28
35
42
49
56
63

8) Change the formatting accordingly over here.

7) Create the Average Days Value measure under the Average Days table.

Key objective 2 is now complete. Use GENERATESERIES & SELECTEDVALUE to create a parameter with increments of 7 days over a 9-week period.

Fields

- Measure Table
- ADOCOLUMNS Demo
- Average Days
 - Average Days
 - Average Days Value
- Calendar
- CROSSJOIN Demo
- Customer Lookup
- DATABLE Demo
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Demo
- New Employees (INTERSECT)
- Product Lookup
- Repeat Customer Sales (INTERSECT Assignment)
- Row Demo
- Sales by Store
- SELECTCOLUMNS Demo
- Store Lookup
- SUMMARIZE Demo
- Table Constructor Demo
- Target Sales - April 2019 (DATABLE)
- Target Sales - March 2019 (DATABLE)
- TREATAS Demo
- UNION Demo
- Unsold Pastries
- VALUES Example

Table: Average Days (9 rows) Column: Average Days Value (0 distinct values)

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Moving Average Pr... Format: General Data category: Uncategorized

Home table: Measure Table \$ - % Auto New Quick measure measure

Structure, Formatting, Properties, Calculators

Visualizations Fields

moving average

Measure Table

Iteration

Moving Average (AVERAGEX)

Moving Average Profit (AVERAGEX)

Values

Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

1 Moving Average Profit (AVERAGEX) =
2 VAR LastTransactionDate = MAX('Calendar'[Transaction_Date])
3 VAR AverageDay = [Average Days Value]
4 VAR PeriodInVisual =
5 FILTER(
6 ALL(
7 'Calendar'[Transaction_Date]
8),
9 AND(
10 'Calendar'[Transaction_Date] > LastTransactionDate - AverageDay,
11 'Calendar'[Transaction_Date] <= LastTransactionDate
12)
13)
14 VAR OutPut =
15 CALCULATE(
16 AVERAGEX(
17 'Calendar',
18 [Profit]
19),
20 PeriodInVisual
21)
22 RETURN
23 OutPut

9) Objective 3 to create a measure using AVERAGEX & the parameter you defined to calculate the moving average profit. Create the Moving Average Profit (AVERAGEX) measure using variables.

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools

Power Get Data Excel Power BI datasets SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure Quick measure Publish

Customer Sales

store_id	Customer Sales
6	\$854,899.34
7	\$20,675.80
12	\$170,789.02
14	\$12,221.39
25	\$12,379.64
26	\$190,729.31
27	\$110,157.14
28	\$98,115.04
29	\$112,912.53
30	\$126,830.45
Total	\$854,899.34

Customer Sales and Moving Average (AVERAGEX) by Transaction_Date

Customer Sales

Transaction Date

01/01/2018 30/04/2019

Select a Single Employee

Sales by Employee Name (CONCATENATEX)

Year_ID Customer Sales Average Daily Sales (AVERAGEX)

Year_ID	Customer Sales	Average Daily Sales (AVERAGEX)
2018	\$1,916,544.75	\$5,250.81
2019	\$658,086.02	\$5,530.19
Total	\$2,574,630.77	\$5,310.49

Year_ID Profit Average Profit (AVERAGEX) Moving Average Profit (AVERAGEX)

Year_ID	Profit	Average Profit (AVERAGEX)	Moving Average Profit (AVERAGEX)
2018	\$1,410,047.46	\$3,887.89	4,191.35
January	\$104,615.09	\$3,374.68	3,380.35
February	\$91,885.10	\$3,281.61	3,258.17
March	\$106,171.47	\$3,424.89	3,430.77
April	\$110,925.38	\$3,697.51	3,697.51
May	\$128,293.47	\$4,138.50	4,151.58
June	\$129,046.22	\$4,301.54	4,301.54
July	\$122,195.18	\$3,941.78	3,944.58
August	\$119,321.13	\$3,849.07	3,860.00
September	\$111,393.30	\$3,713.11	3,713.11
Total	\$1,906,349.87	\$3,938.74	5,931.15

Visualizations Fields

Filters

Measure Table Iterators

- Moving Average (AVERAGEX)
- Moving Average Profit (AVERAGEX)

Values

Add data fields here

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

10) Add the Moving Average Profit (AVERAGEX) measure to the matrix.

Home Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATEX

Page 9 of 9

Maven Roasters - Analysis - Iterators - Power BI Desktop

The average profit values will remain static when the slicer is changed for the average days selections.

Select a Single Employee

Sales by Employee Name (CONCATENATEX)

Year_ID	Customer Sales	Average Daily Sales (AVERAGEX)
2017	\$1,678,074.11	\$4,597.46
January	\$81,843.09	\$2,640.16
February	\$76,273.99	\$2,724.07
March	\$99,154.43	\$3,198.53
April	\$119,359.01	\$3,197.67
May	\$137,200.99	\$3,071.26
June	\$166,899.78	\$3,561.33
July	\$157,988.55	\$3,695.76
August	\$154,485.32	\$4,983.40
Total	\$4,252,704.88	\$3,009.00

Average Days

Average Days	Year_ID	Profit	Average Profit (AVERAGEX)	Moving Average Profit (AVERAGEO)
7	2017	\$1,242,522.69	\$3,404.17	4,109.63
14	January	\$60,662.93	\$1,935.87	1,945.00
21	February	\$58,449.75	\$2,016.06	2,110.58
28	March	\$70,377.76	\$3,548.80	3,371.25
35	April	\$88,364.99	\$2,545.50	2,521.42
42	May	\$116,370.94	\$3,733.90	3,819.74
49	June	\$123,680.33	\$4,122.74	4,205.77
56	July	\$118,991.33	\$3,773.92	3,770.49
63	August	\$114,384.04	\$3,693.81	3,730.63
Total		\$3,148,872.56	\$3,708.92	6,055.38

Visualizations

Fields

average

Measure Table

Iterators

- Average Daily Sales (AVERAGEX)
- Average Profit (AVERAGEX)
- Moving Average (AVERAGEO)
- Moving Average Profit (AVERAGEX)

Relationship Functions

- Average Order Value (CVP)

Average Days

- Average Days
- Average Days Value

11) Last task is to be able to provide a way to update the moving days average currently defaulting to 30 days average we had set.

- Add a slicer to the matrix.
- Add the average days field to the slicer.

Key objective 3 is now complete. Create a measure using AVERAGEX & the parameter you defined to calculate the moving average profit.

Icons Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATEX

RANKX

RANKX()

Returns the ranking of a number in a list of numbers for each row in the table argument

Iterator Review

Iterator
Cardinality

CONCATENATEX

AVERAGEX

RANKX

=RANKX(Table, Expression, [Value], [Order], [Ties])

Optional arguments:

- **Value:** Any DAX expression that returns a single scalar value whose rank is to be found (by default, the value in the current row is used)
- **Order:** Specifies how to rank (low-high vs. high-low)
 - Examples: ASC or DESC
- **Ties:** Determines how ties and following ranks are treated:
 - SKIP (default): Skips ranks after ties
 - DENSE: Shows the next rank, regardless of ties



PRO TIP:

If your "Total" row is showing a rank of 1, use IF & HASONEVALUE with RANKX to exclude it from the rank!

SKIP:

Rounded Customer Sales	Rank on Rounded Customer Sales
\$1,700,000.00	1
\$1,200,000.00	2
\$100,000.00	3
\$400,000.00	4
\$300,000.00	5
\$100,000.00	6
\$100,000.00	7
\$0.00	8
\$4,300,000.00	9

DENSE:

Rounded Customer Sales	Rank on Rounded Customer Sales
\$1,700,000.00	1
\$1,200,000.00	2
\$100,000.00	3
\$400,000.00	4
\$300,000.00	5
\$100,000.00	6
\$100,000.00	7
\$0.00	8
\$4,300,000.00	9

-[**table**, **expression**, [**value**], [**order**], [**ties**])

Optional arguments:

- **Value:** Any DAX expression that returns a single scalar value whose rank is to be found (by default, the value in the current row is used)
- **Order:** Specifies how to rank (low-high vs. high-low)
 - **Examples:** ASC or DESC
- **Ties:** Determines how ties and following ranks are treated:
 - **SKIP** (default): Skips ranks after ties
 - **DENSE:** Shows the next rank, regardless of ties

SKIP:

Rounded Customer Sales	Rank on Rounded Customer Sales
\$1,700,000.00	1
\$1,200,000.00	2
\$500,000.00	3
\$400,000.00	4
\$300,000.00	5
\$100,000.00	6
\$100,000.00	6
\$100,000.00	6
\$0.00	9
\$4,300,000.00	

DENSE:

Rounded Customer Sales	Rank on Rounded Customer Sales
\$1,700,000.00	1
\$1,200,000.00	2
\$500,000.00	3
\$400,000.00	4
\$300,000.00	5
\$100,000.00	6
\$100,000.00	6
\$100,000.00	6
\$0.00	7
\$4,300,000.00	

The screenshot shows the Microsoft Power BI Data Editor interface. On the left, there is a data grid titled "Customer Sales" with columns "product_category" and "Customer Sales". The data includes rows for Bakery, Branded, Coffee, Coffee beans, Drinking Chocolate, Flavours, Loose Tea, Packaged Chocolate, Tea, and a Total row. A blue arrow points from the text below to this grid. On the right, there is a "Visualizations" pane with various icons and a "Fields" pane listing various Power BI functions and demos. The "Fields" pane has sections for Rows, Columns, Values, Drill through, Cross-report, and Keep all filters.

1) Create a new matrix. So that the customer sales can be ranked.

- Add the product category field to the row.
- Add the customer sales field to the row.

Maven Roadster - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name: Rank of Customer Sales
Home table: Measure Table

Format: Whole number
\$ - % , . : Data category: Uncategorized

New Quick measure measure Calculations Properties

1 Rank of Customer Sales (RANKX) =
2 IF(
3 HASONEVALUE(
4 'Product Lookup'[product_category]
5),
6 RANKX(
7 ALL(
8 'Product Lookup'[product_category]
9),
10 [Customer Sales]
11)
12)

3) Change the formatting accordingly over here.

Visualizations > Fields

Rank

Measure Table Iterators

Rank of Customer Sales (RANKX)

Rank of Rounded Customer Sales (RANKX)

Top 5 Products by Profit (RANKX)

Rows: product_category

Columns: Customer Sales

Drill through: Off

Cross-report: Off

Keep all filters: On

Assignment: CONCATENATE

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATE

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign In

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get Data - Excel Power BI SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Calculations Share

Queries

product_category Customer Sales Rank of Customer Sales (RANKX)

product_category	Customer Sales	Rank of Customer Sales (RANKX)
Coffee	\$1,651,861.15	1
Tea	\$1,190,314.90	2
Bakery	\$501,291.32	3
Drinking Chocolate	\$439,942.50	4
Coffee beans	\$240,414.95	5
Branded	\$83,784.00	6
Loose Tea	\$68,451.90	7
Flavours	\$51,060.00	8
Packaged Chocolate	\$25,584.16	9
Total	\$4,252,764.88	

Visualizations Fields

Search

Measure Table ADDCOLUMNS Demo Average Days Calendar CROSSJOIN Demo Customer Lookup DATATABLE Demo Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo New Employees (INTERSECT) Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store SELECTCOLUMNS Demo Store Lookup SUMMARIZE Demo Table Constructor Demo Target Sales - April 2019 (DATATABLE) Target Sales - March 2019 (DATATABLE) TREATAS Demo UNION Demo Unsold Pastries VALUES Example

Filters

Rows product_category

Columns Add data fields here

Values Customer Sales Rank of Customer Sales (RANKX)

Drill through

Cross-report

On Off

Keep all filters

On

Add drill-through fields here

4) The rank of sales are displayed here & the total displayed no values from performing the steps from the previous slide.

The screenshot shows the Power BI Measure Editor interface. The top navigation bar includes File, Home, Insert, Modeling, View, Help, External Tools, Table tools, and Measure tools. The Measure tools tab is selected. The left pane displays the measure definition:

```
1 Rounded Customer Sales =  
2 MROUND(  
3     [Customer Sales],  
4     100000  
5 )
```

A blue arrow points from the text "5) Example of using MROUND to round the Customer Sales to the nearest 100 thousand." to the number "100000" in the measure definition.

The right pane shows the Visualizations and Fields sections. In the Fields section, there is a list of measures, with "Rounded Customer Sales" highlighted in yellow. The table below shows sample data:

	Loose Tea	\$68,451.90	7
	Flavours	\$51,060.00	8
	Packaged Chocolate	\$25,584.16	9
Total		\$4,252,704.00	

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get Data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Datasets Data Queries Calculations Share

product_category Customer Sales Rank of Customer Sales (RANKID) Rounded Customer Sales

Coffee	\$1,651,861.15	1	\$1,700,000.00
Tea	\$1,190,314.90	2	\$1,200,000.00
Bakery	\$501,291.32	3	\$500,000.00
Drinking Chocolate	\$439,942.50	4	\$400,000.00
Coffee beans	\$240,414.95	5	\$200,000.00
Branded	\$83,784.00	6	\$100,000.00
Loose Tea	\$68,451.90	7	\$100,000.00
Flavours	\$51,060.00	8	\$100,000.00
Packaged Chocolate	\$25,584.16	9	\$0.00
Total	\$4,252,704.88		\$4,300,000.00

6) Rounded the Customer Sales to the nearest 100 thousand.

7) 3 way tie of rounding customer sales. Skip & dense comes in handy in these situations.

Visualizations Fields

Measure Table Iterators

Rank of Rounded Customer Sales (RANKID) Rounded Customer Sales

Values Add data fields here

Drill through

Cross-report

Off On Keep all filters

Add drill-through fields here

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATE

Page 10 of 10

Maven Readers - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Rank of Rounded C... Format: Whole number Data category: Uncategorized New Quick measure measure

Home table: Measure Table \$ - % , . 0 Structure Formatting Properties Calculations

1 Rank of Rounded Customer Sales (RANKX) =
2 IF(
3 HASONEVALUE(
4 'Product Lookup'[product_category])
5),
6 RANKX(
7 ALL(
8 'Product Lookup'[product_category])
9),
10 [Rounded Customer Sales],
11 ,DESC
12 ,Dense ← 8) Create the Rank of Rounded Customer Sales
13 (RANKX) measure.
14)

Visualizations Fields
Search
Measure Table ADDCOLUMNS Demo Average Days Calendar CROSSJOIN Demo Customer Lookup DATATABLE Demo Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo New Employees (INTERSECT) Product Lookup Repeat Customer Sales (INTERSECT Assignment) Row Demo Sales by Store SELECTCOLUMNS Demo Store Lookup SUMMARIZE Demo Table Constructor Demo Target Sales - April 2019 (DATATABLE) Target Sales - March 2019 (DATATABLE) TRIMAS Demo UNION Demo Unsold Products VALUES Example

Add data fields here
Drill through
Cross-report
Off On
Keep all filters
On
Add drill-through fields here

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATE

Maven Readers - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Out data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Queries Calculations Share

product_category Customer Sales Rank of Customer Sales (RANKX) Rounded Customer Sales Rank of Rounded Customer Sales (RANKX)

product_category	Customer Sales	Rank of Customer Sales (RANKX)	Rounded Customer Sales	Rank of Rounded Customer Sales (RANKX)
Coffee	\$1,651,861.15	1	\$1,700,000.00	1
Tea	\$1,190,314.90	2	\$1,200,000.00	2
Bakery	\$501,291.32	3	\$500,000.00	3
Drinking Chocolate	\$439,942.50	4	\$400,000.00	4
Coffee beans	\$240,414.95	5	\$200,000.00	5
Branded	\$83,784.00	6	\$100,000.00	6
Loose Tea	\$68,451.90	7	\$100,000.00	6
Flavours	\$51,060.00	8	\$100,000.00	6
Packaged Chocolate	\$25,534.16	9	\$0.00	7
Total	\$4,252,704.88		\$4,300,000.00	

Visualizations Fields

Filters

Measure Table Iterators

- Rank of Customer Sales (RANKX)
- Rank of Rounded Customer Sales (RANKX)
- Top 3 Products by Profit (RANKX)

Rows product_category

Columns Add data fields here

Values Customer Sales, Rank of Customer Sales (RANKX), Rounded Customer Sales (RANKX), Rank of Rounded Customer Sales (RANKX)

Drill through

Cross-report Off

Keep all filters On

Add drill-through fields here

9) Add the Rank of Rounded Customer Sales (RANKX) measure to the matrix.

10) When you get to 100 thousand these 3 values are ranked the same i.e. 6.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT CONCATENATE

Page 10 of 20 pgAdmin 4 v1

Maven Readers - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Rank of Rounded C... Format: Whole number Data category: Uncategorized New measure measure

Home table: Measure Table \$ - % , . 0 Structure Formatting Properties Calculations

1 Rank of Rounded Customer Sales (RANKX) =
2 IF(
3 HASONEVALUE(
4 'Product Lookup'[product_category])
5),
6 RANKX(
7 ALL(
8 'Product Lookup'[product_category])
9),
10 [Rounded Customer Sales],
11 ,DESC
12 ,Skip
13)
14)

Visualizations > Fields
rank
Measure Table
Iterations
Rank of Customer Sales (RANKX)
Rank of Rounded Customer Sales (RANKX)
Top 3 Products by Profit (RANKX)

Add data fields here
Values
Add drill-through fields here

Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields here

11) Amend statement here to skip from previously dense.



12) When using skip the opposite happens here & skips to 9.

A screenshot of the Power BI Desktop interface. On the left, there's a data grid showing sales data for various products across different categories. The columns include product category, customer sales, rank of customer sales, rounded customer sales, and rank of rounded customer sales. A blue arrow points from the text above to the 'Rank of Rounded Customer Sales' column, highlighting the value '9'. To the right, the 'Fields' pane is open, showing a measure table with three iterators: 'Rank of Customer Sales', 'Rank of Rounded Customer Sales', and 'Top 3 Products by Profit'. The 'Values' section of the iterator pane shows the top 3 products by profit, which includes the product shown in the grid.

product_category	Customer Sales	Rank of Customer Sales (RANKX)	Rounded Customer Sales	Rank of Rounded Customer Sales (RANKX)
Coffee	\$1,651,001.15	1	\$1,701,000.00	1
Tea	\$1,190,314.90	2	\$1,200,000.00	2
Bakery	\$501,291.32	3	\$500,000.00	3
Drinking Chocolate	\$439,942.50	4	\$400,000.00	4
Coffee beans	\$240,414.95	5	\$200,000.00	5
Branded	\$83,784.00	6	\$100,000.00	6
Loose Tea	\$68,451.90	7	\$100,000.00	6
Flavours	\$51,060.00	8	\$100,000.00	6
Packaged Chocolate	\$25,534.16	9	\$0.00	9
Total	\$4,252,704.88		\$4,300,000.00	

KEY OBJECTIVES

- 1) Use **variables** & **RANKX** to create a single measure for Top 5 Products.
- 2) Create a slicer for Product Category to be able to analyse the top 5 across different categories.

Maven Roasters - Analysis - Iterators - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: Top 5 Products by ... Format: Whole number Data category: Uncategorized New measure Quick measure

Home table: Measure Table \$ - % , . : Properties Descriptions

1 Top 5 Products by Profit (RANKX) =
2 VAR ProfitRank =
3 IF(
4 HASONEVALUE(
5 'Product Lookup'[product_category])
6),
7 RANKX(
8 ALL(
9 'Product Lookup'[product])
10),
11 [Customer Sales] - [Cost]
12)
13)
14 VAR Top5Products =
15 IF(
16 ProfitRank <= 5,
17 [Profit]
18)
19 Return
20 Top5Products

2) Change the formatting accordingly over here.

Visualizations Fields

Key objective 1 is now complete. Use variables & RANKX to create a single measure for Top 5 Products.

1) Objective 1 to use variables & RANKX to create a single measure for Top 5 Products. Create the Top 5 Products by Profit (RANKX) measure using variables.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CONCATENATE

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter Delete Data Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

product_category Customer Sales Rank of Customer Sales (RANKX) Rounded Customer Sales Rank of Rounded Customer Sales (RANKX)

product_category	Customer Sales	Rank of Customer Sales (RANKX)	Rounded Customer Sales	Rank of Rounded Customer Sales (RANKX)
Coffee	\$1,651,861.13	1	\$1,700,000.00	1
Tea	\$1,190,314.90	2	\$1,200,000.00	2
Bakery	\$301,291.32	3	\$300,000.00	3
Drinking Chocolate	\$439,340.50	4	\$400,000.00	4
Coffee beans	\$240,414.95	5	\$200,000.00	5
Branded	\$80,784.00	6	\$100,000.00	6
Loose Tea	\$68,451.90	7	\$100,000.00	6
Flavours	\$51,060.00	8	\$100,000.00	6
Packaged Chocolate	\$23,584.16	9	\$0.00	9
Total	\$4,252,704.88		\$4,300,000.00	

product_category product Top 5 Products by Profit (RANKX)

product	Top 5 Products by Profit (RANKX)
Cappuccino lg	82352
Dark chocolate lg	94949
Latte lg	66312
Morning Sunrise Chai lg	60811
Sustainably Grown Organic lg	56622
Total	3148873

product_category

3) Objective 2 to create a slicer for Product Category to be able to analyse the top 5 across different categories. Perform the below tasks.

- Add in a new matrix to the report.
- Add in the Top 5 Products by profit (RANKX) field to the values.
- Add in the product field to the rows.
- Add in a slicer to the report.
- Add product category to the slicer.

Visualizations Fields

product

product_category

Top 5 Products by Profit (RANKX)

On

Assignment: CO

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Duplicate of Iterators ASSIGNMENT: CO

Page 10 of 10

Maven Roasters - Analysis - Iterators - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box Insert More visuals New measure measure Publish

Queries Calculations Share

product,category Customer Sales Rank of Customer Sales (RANKX) Rounded Customer Sales (RANKX) Rank of Rounded Customer Sales (RANKX)

Coffee beans	\$240,474.93	5	\$200,000.00	5
Branded	\$83,784.00	6	\$100,000.00	6
Flavours	\$51,060.00	8	\$100,000.00	6
Total	\$375,258.93		\$400,000.00	

4) Multiply product categories can be selected for a top 5 rank calculation.

product_category

- Bakery
- Branded
- Coffee
- Coffee beans
- Drinking Chocolate
- Flavours
- Loose Tea
- Packaged Chocolate
- Tea

product

Top 5 Products by Profit (RANKX)

Civet Cat	53244
Ethiopia	22277
I Need My Bean! Latte cup	22330
I Need My Bean! T-shirt	31278
Organic Decaf Blend	22523
Total	298425

Visualizations Fields

product_category

CROSSJOIN Demo

product_category

Product Lookup

product_category

Drill through

Cross-report

Keep all filters

Add drill-through fields here

Tips & Best Practices

Scalar Functions

Advanced CALCULATE

Table & Filter Functions

Join

Relationship Functions

Iterators

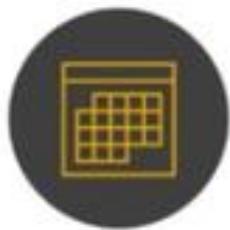
Duplicate of Iterators

ASSIGNMENT CO

Page 10 of 20

10) Advanced Time Intelligence

Introduction - Advanced Time Intelligence



DAX offers a range of powerful **time intelligence** functions, which allow you to build custom calendars, define dynamic date ranges, and compare performance across specific periods

TOPICS WE'LL COVER:

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence
Functions

Week-Based
Calculations

COMMON USE CASES:

- *Building custom calendar tables (rather than using default, auto-generated versions)*
- *Calculating period-over-period, moving average, or running total calculations*
- *Managing non-traditional time periods like week-based retail calendars (5-4-5 or 5-4-4)*

Automatic Date Tables

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

By default, Power BI automatically creates a **hidden date table** for any table that contains a **Date** or **DateTime** column on the one-side of a relationship

- Auto-generated calendars include *all* dates through the end of the year, regardless of the actual date range in the table

The diagram illustrates a relationship between two tables. On the left, the 'Employee Lookup' table is shown with columns: first_name, last_name, location, position, staff_id, and start_date. The 'start_date' column is highlighted with a yellow border. A curly brace on the right side of the table indicates a relationship to another table. To the right of the brace is a screenshot of a date table with the following data:

Date	Day	Month	Quarter	Year
1/01/2017 00:00:00	1	January	Q1	2017
1/02/2017 00:00:00	2	January	Q1	2017
1/03/2017 00:00:00	3	January	Q1	2017
1/04/2017 00:00:00	4	January	Q1	2017
1/05/2017 00:00:00	5	January	Q1	2017
1/06/2017 00:00:00	6	January	Q1	2017
1/07/2017 00:00:00	7	January	Q1	2017
1/08/2017 00:00:00	8	January	Q1	2017
1/09/2017 00:00:00	9	January	Q1	2017
1/10/2017 00:00:00	10	January	Q1	2017
1/11/2017 00:00:00	11	January	Q1	2017
1/12/2017 00:00:00	12	January	Q1	2017

Automatically creates a **hidden date table** containing all these columns

Automatic Date
Tables

Calendar
Functions

Date Formatting

Time Intelligence
Functions

Week-Based
Calculations



PROS:

- Automatically generated
- Enables (some) time intelligence functionality by default
- Simplifies data model creation and management
- Does not require an advanced understanding of DAX



CONS:

- Hidden from view, cannot be modified/customized
- Generated for every date field across every lookup/dimension table (*bloats model size*)
- Can't be enabled or disabled at the table-level
- Hierarchies aren't automatically generated (*if grouped by month, would summarize that month across ALL years*)
- Each automatic date table can *only* filter the table it corresponds to (*cannot traverse table relationships*)



PRO TIP:

Turn OFF the **auto date/time** feature in Power BI Desktop and either import a date dimension table or create your own using **CALENDAR** functions (more on this soon!)

Date Table Requirements

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

If you import or create your own date table, it must meet these requirements:

- ✓ Must contain *all* the days for all years represented in your fact tables
- ✓ Must have at least one field set as a **Date** or **Datetime** datatype
- ✓ Cannot contain duplicate dates or datetime values
- ✓ If using a time component within a date column, all times must be identical (*i.e.* 12:00)
- ✓ Should be marked as a **date table** (*not required but a best practice*)

HEY THIS IS IMPORTANT!

If **Time** is present in your date field, split the time component into a new column
(this adheres to relationship requirements and decreases column cardinality)

Maven Roasters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get Data - Excel Power BI datasets SQL Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Dataset Data Query Insert Calculations Share

Build visuals with your data

Select or drag fields from the Fields pane onto the report canvas.

Visualizations Fields

Add data fields here

Drill through

Cross-report

On Off

Keep all filters

On Off

Add drill-through fields here

Mark as date table

Data table settings

Mark as date table

Mark as date table

Unhide all

Collapse all

Expand all

INTERSECT Assignment

1) Go to the calendar table & perform the below actions.

- Select Mark as date table.
- Choose option Mark as date table.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of

Page 9 of 9

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More New measure measure Quick Publish

Queries Calculations Share

Visualizations Fields

Search

Add data fields here

Date column

Transaction Date

Validated successfully

When you mark this as a date table, the built-in date tables that were associated with this table are removed. Visuals or DAX expressions referring to them may break.

Learn how to fix visuals and DAX expressions

OK Cancel

Build visuals

Select or drag fields from the Fields pane to the canvas

Mark as date table

2) Select transaction date from the drop down menu & click OK.
Power BI has automatically calculated that transaction date is the only field where the date values are unique.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of +

CALENDAR

Automatic Date
Tables

Calendar
Functions

Date Formatting

Time Intelligence
Functions

Week-Based
Calculations

CALENDAR()

Returns a table with one column of all dates between start and end date

=CALENDAR(StartDate, EndDate)

Start and End dates of your calendar table. Can be explicitly assigned or defined using DAX functions (i.e. MIN/MAX)

Examples:

- `CALENDAR(
DATE(2019,01,01), DATE(2020,12,31))`
- `CALENDAR(
DATE(YEAR (MIN(Sales by Store[Transaction Date])), 1, 1),
DATE(YEAR (MAX(Sales by Store[Transaction Date])), 12, 31))`



Date
1/1/2017 12:00:00 AM
1/2/2017 12:00:00 AM
1/3/2017 12:00:00 AM
1/4/2017 12:00:00 AM
1/5/2017 12:00:00 AM
1/6/2017 12:00:00 AM
1/7/2017 12:00:00 AM
1/8/2017 12:00:00 AM
1/9/2017 12:00:00 AM
1/10/2017 12:00:00 AM
1/11/2017 12:00:00 AM
1/12/2017 12:00:00 AM
1/13/2017 12:00:00 AM
1/14/2017 12:00:00 AM
1/15/2017 12:00:00 AM
1/16/2017 12:00:00 AM
1/17/2017 12:00:00 AM

Maven Roaders - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Help External Tools Table tools

Name CALENDAR Demo

Mark as date table = Manage relationships New measure Quick New measure column New table

Measure Calculations

1 CALENDAR Demo =
2 CALENDAR(
3 DATE(YEAR(MIN('Calendar'[Transaction_Date])),1,1),
4 DATE(YEAR(MAX('Calendar'[Transaction_Date])),12,31)
5)

Date
01/01/2017 00:00:00
02/01/2017 00:00:00
03/01/2017 00:00:00
04/01/2017 00:00:00
05/01/2017 00:00:00
06/01/2017 00:00:00
07/01/2017 00:00:00
08/01/2017 00:00:00
09/01/2017 00:00:00
10/01/2017 00:00:00
11/01/2017 00:00:00
12/01/2017 00:00:00
13/01/2017 00:00:00
14/01/2017 00:00:00
15/01/2017 00:00:00
16/01/2017 00:00:00
17/01/2017 00:00:00
18/01/2017 00:00:00
19/01/2017 00:00:00
20/01/2017 00:00:00
21/01/2017 00:00:00
22/01/2017 00:00:00
23/01/2017 00:00:00
24/01/2017 00:00:00
25/01/2017 00:00:00
26/01/2017 00:00:00

Fields

Search

Measure Table
4-5-4 Calendar
ADOCOLUMNS.DL
Average Days
Calendar
CALENDAR Demo
Date
CALENDARAUTO...
CROSSJOIN Demo
Customer Lookup
DATABLE Demo
Data Table (CALL...
Employee Lookup
EXCEPT Demo
Food Inventory
GENERATESERIES...
New Employees (L...
Product Lookup
Repeat Customer...
Row Demo
Sales by Store
SELECTCOLUMNSL...
Store Lookup
SUMMARIZE Demo
Table Constructor...
Target Sales - Apr...
Target Sales - Ma...
TREATAS Demo

1) Using the CALENDAR function to specify the start & end dates.

CALENDARAUTO

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

CALENDARAUTO()

Returns a table with one column of dates based on a fiscal year end month. The Range of dates is calculated automatically based on data in the model

=CALENDARAUTO(FiscalYearEndMonth)

An integer from 1 to 12 that represents the last month of the fiscal year.
Can be explicitly assigned or created using DAX functions (i.e. MIN/MAX)

Examples:

- CALENDARAUTO(6)
- Calendar Table =

```
VAR MinYear = YEAR( MIN( 'Sales by Store'[Transaction Date]))
```

```
VAR MaxYear= YEAR( MAX( 'Sales by Store'[Transaction Date]))
```

```
RETURN
```

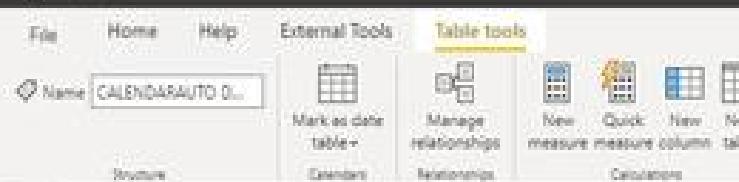
```
FILTER( CALENDARAUTO(),
```

```
YEAR( [Date] ) >= MinYear &&
```

```
YEAR( [Date] ) <= MaxYear )
```

1 CALENDARAUTO =
2 CALENDARAUTO(6)

Date
7/1/2016 12:00:00 AM
7/2/2016 12:00:00 AM
7/3/2016 12:00:00 AM
7/4/2016 12:00:00 AM
7/5/2016 12:00:00 AM
7/6/2016 12:00:00 AM
7/7/2016 12:00:00 AM
7/8/2016 12:00:00 AM
7/9/2016 12:00:00 AM
7/10/2016 12:00:00 AM



1 CALENDARAUTO Demo =

2 CALENDARAUTO(6)

Table CALENDARAUTO Demo (44,560 rows)

1) Create a new table called CALENDARAUTO Demo (6 is the Fiscal Year End Month). July 01 will be returned as a starting point. Results are provided however look relatively surprising start date & month are accurate but the year is incorrect. What is actually happening here is a interesting when mixing between how CALENDARAUTO works because a specific table is not defined or column for CALENDARAUTO to reference. CALENDARAUTO actually scans every column within the data model looking for date data types & in our model we have dates like transaction date, customer birth date, customer since date, baked date none of those columns start with those dates that early. The output is not incorrect in this example.

Fields

Search

- Measure Table
- 4-5-4 Calendar
- ADDCOLUMNS Demo
- Average Days
- Calendar
- CALENDAR Demo
- Date
- CALENDARAUTO Demo
- Date
- CROSSJOIN Demo
- Customer Lookup
- DATATABLE Demo
- Date Table (CALENDARAUTO)
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Demo
- New Employee (INTERSECT)
- Product Lookup
- Report Customer Sales (INTERSECT Along)
- Row Demo
- Sales by Store
- SELECTCOLUMNS Demo
- Store Lookup
- SUMMARIZE Demo
- Table Constructor Demo
- Target Sales - April 2019 (DATATABLE)
- Target Sales - March 2019 (DATATABLE)

PRO TIP: Building a Reusable Date Table

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

```
1 CALENDARAUTO Date Table =
2 VAR MinYear = YEAR(MIN('Sales by Store'[transaction_date]))
3 VAR MaxYear = YEAR(MAX('Sales by Store'[transaction_date]))
4
5 RETURN
6 ADDCOLUMNS(
7     FILTER(
8         CALENDARAUTO(),
9         YEAR([Date]) >= MinYear &&
10        YEAR([Date]) <= MaxYear
11    ),
12    "Year", YEAR([Date]),
13    "Quarter Number", INT(FORMAT([Date], "q")),
14    "Quarter", "Q" & INT(FORMAT([Date], "q")),
15    "Month Number", MONTH([Date]),
16    "Month Short", FORMAT([Date], "mm"),
17    "Week Number", WEEKNUM([Date])
18 )
```

This code can be repurposed and recycled across multiple data models!

Date	Year	Quarter Number	Quarter	Month Number	Month Short	Week Number
1/1/2017 12:00:00 AM	2017	1 Q1	1 Jan	1	JAN	1
1/2/2017 12:00:00 AM	2017	1 Q1	1 Jan	2	JAN	1
1/3/2017 12:00:00 AM	2017	1 Q1	1 Jan	3	JAN	1
1/4/2017 12:00:00 AM	2017	1 Q1	1 Jan	4	JAN	1
1/5/2017 12:00:00 AM	2017	1 Q1	1 Jan	5	JAN	1
1/6/2017 12:00:00 AM	2017	1 Q1	1 Jan	6	JAN	1
1/7/2017 12:00:00 AM	2017	1 Q1	1 Jan	7	JAN	1
1/8/2017 12:00:00 AM	2017	1 Q1	1 Jan	8	JAN	2
1/9/2017 12:00:00 AM	2017	1 Q1	1 Jan	9	JAN	2
1/10/2017 12:00:00 AM	2017	1 Q1	1 Jan	10	JAN	2
1/11/2017 12:00:00 AM	2017	1 Q1	1 Jan	11	JAN	2
1/12/2017 12:00:00 AM	2017	1 Q1	1 Jan	12	JAN	2

Maven Reactors - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Help External Tools Table tools

Name Date Table (CALENDAR)

Mark as date table+ Manage relationships New measure Quick New measure column New table

Structure Calculations

```
1 Date Table (CALENDARAUTO) =  
2 VAR MinYear = YEAR(MIN('Calendar'[Transaction_Date]))  
3 VAR MaxYear = YEAR(MAX('Calendar'[Transaction_Date]))  
4  
5 RETURN  
6 ADDCOLUMNS(  
7     FILTER(  
8         CALENDARAUTO(),  
9         YEAR([Date]) >= MinYear &&  
10        YEAR([Date]) <= MaxYear  
11    ),  
12    "Year", YEAR([Date]),  
13    "Quarter Number", INT(FORMAT([Date], "q")),  
14    "Quarter", "Q" & INT(FORMAT([Date], "q")),  
15    "Month Number", MONTH([Date]),  
16    "Month Short", FORMAT([Date], "mmm"),  
17    "Week Number", WEEKNUM([Date])  
18 )
```

a) This portion of the code only needs to be changed i.e. table & column references which are in the variables over here.

b) This portion of the code i.e. FILTER allows you to generate a table which we can than add columns to, table is generated by the dates between two YEAR's based on CALENDARAUTO.

c) Columns created with specified data types & format specified.

1) Created reusable calendar code outputs these results.

Date	Year	Quarter Number	Quarter	Month Number	Month Short	Week Number
01/01/2017 00:00:00	2017	1 Q1	1 Q1	1 Jan	J Jan	20
02/01/2017 00:00:00	2017	1 Q1	1 Q1	2 Jan	J Jan	21
03/01/2017 00:00:00	2017	1 Q1	1 Q1	3 Jan	J Jan	22
04/01/2017 00:00:00	2017	1 Q1	1 Q1	4 Jan	J Jan	23
05/01/2017 00:00:00	2017	1 Q1	1 Q1	5 Jan	J Jan	24
06/01/2017 00:00:00	2017	1 Q1	1 Q1	6 Jan	J Jan	25
07/01/2017 00:00:00	2017	1 Q1	1 Q1	7 Jan	J Jan	26
08/01/2017 00:00:00	2017	1 Q1	1 Q1	8 Jan	J Jan	27
09/01/2017 00:00:00	2017	1 Q1	1 Q1	9 Jan	J Jan	28
10/01/2017 00:00:00	2017	1 Q1	1 Q1	10 Jan	J Jan	29

Table: Date Table (CALENDARAUTO) (1095 rows)

Fields

- Measure Table
- 4-5-4 Calendar
- ADDCOLUMNS Demo
- Average Days
- Calendar
- CALENDAR Demo
- CALENDARAUTO Demo
- CROSSJOIN Demo
- Customer Lookup
- DATABLE Demo
- Date Table (CALENDARAUTO)
- Month Number
- Month Short
- Quarter
- Quarter Number
- Week Number
- Year
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Demo
- New Employees (INTERSECT)
- Product Lookup
- Repeat Customer Sales (INTERSECT Anag.)
- Row Demo

Date Formatting

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

Use the **FORMAT** function to specify date/time formatting. Common examples include:

Code	Description	Example
dddd	Display the full-day name	Sunday, Monday, etc.
ddd	Display the short-day name	Sun, Mon, Tue, etc.
mmmm	Show full month name	July, August, etc.
mmm	Show abbreviated month name	Jul, Aug, Sept, etc.
mm	Display the month as a two-digit number	08, 09, 10, 11, etc.
q	Display the quarter of the year	1, 2, 3, 4
yyyy	Show the year as a four-digit number	2018, 2019, 2020, etc.

<https://docs.microsoft.com/en-us/dax/custom-date-and-time-formats-for-the-format-function>

ASSIGNMENT: Date Formatting

KEY OBJECTIVES

- 1) Add columns to the calendar table to capture the weekday number & name.
- 2) Add a binary Y/N column to flag weekend dates.
- 3) Use a Matrix visual to show the percent of total sales for weekdays vs weekends.

1) Objective 1 to add columns to the calendar table to capture the weekday number & name. Create a new table called Weekday Number.

The screenshot shows the 'Weekday Number' column being added to the 'Calendar' table. The 'Weekday Number' column is highlighted in yellow. A blue arrow points from the 'Weekday Name' field in the Fields pane to the 'Weekday Number' column in the table preview.

Transaction_Date	Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	RNUMBER	Date Format (yyyy-mm-dd)	Year Half	Quarter & Year	Weekday Number	Weekday Name
01/07/2017	20170701	26	Week 26	7	July	3	Q3	2017	Week 26-2017	True	2017-07-01	2H	Q3-2017	6	Saturday
02/07/2017	20170702	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-02	2H	Q3-2017	7	Sunday
03/07/2017	20170703	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-03	2H	Q3-2017	1	Monday
04/07/2017	20170704	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-04	2H	Q3-2017	2	Tuesday
05/07/2017	20170705	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-05	2H	Q3-2017	3	Wednesday
06/07/2017	20170706	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-06	2H	Q3-2017	4	Thursday
07/07/2017	20170707	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-07	2H	Q3-2017	5	Friday
08/07/2017	20170708	27	Week 27	7	July	3	Q3	2017	Week 27-2017	True	2017-07-08	2H	Q3-2017	6	Saturday
09/07/2017	20170709	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-09	2H	Q3-2017	7	Sunday
10/07/2017	20170710	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-10	2H	Q3-2017	1	Monday
11/07/2017	20170711	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-11	2H	Q3-2017	2	Tuesday
12/07/2017	20170712	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-12	2H	Q3-2017	3	Wednesday
13/07/2017	20170713	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-13	2H	Q3-2017	4	Thursday
14/07/2017	20170714	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-14	2H	Q3-2017	5	Friday
15/07/2017	20170715	28	Week 28	7	July	3	Q3	2017	Week 28-2017	True	2017-07-15	2H	Q3-2017	6	Saturday
16/07/2017	20170716	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-16	2H	Q3-2017	7	Sunday
17/07/2017	20170717	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-17	2H	Q3-2017	1	Monday
18/07/2017	20170718	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-18	2H	Q3-2017	2	Tuesday
19/07/2017	20170719	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-19	2H	Q3-2017	3	Wednesday
20/07/2017	20170720	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-20	2H	Q3-2017	4	Thursday
21/07/2017	20170721	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-21	2H	Q3-2017	5	Friday
22/07/2017	20170722	29	Week 29	7	July	3	Q3	2017	Week 29-2017	True	2017-07-22	2H	Q3-2017	6	Saturday
23/07/2017	20170723	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-23	2H	Q3-2017	7	Sunday
24/07/2017	20170724	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-24	2H	Q3-2017	1	Monday
25/07/2017	20170725	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-25	2H	Q3-2017	2	Tuesday
26/07/2017	20170726	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-26	2H	Q3-2017	3	Wednesday
27/07/2017	20170727	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-27	2H	Q3-2017	4	Thursday
28/07/2017	20170728	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-28	2H	Q3-2017	5	Friday
29/07/2017	20170729	30	Week 30	7	July	3	Q3	2017	Week 30-2017	True	2017-07-29	2H	Q3-2017	6	Saturday
30/07/2017	20170730	31	Week 31	7	July	3	Q3	2017	Week 31-2017	True	2017-07-30	2H	Q3-2017	7	Sunday

File Home Help External Tools Table tools Column tools

Name Weekday Name
Data type Text
Structure
Formatting Properties

Summarization Don't summarize
Data category Uncategorized
Sort by column Sort
Data groups Group
Manage relationships Relationships
New column Calculations

```
1 Weekday Name =
2 FORMAT(
3     'Calendar'[Transaction_Date],
4     "dddd"
5 )
```

Fields

weekday
Calendar
Weekday Name
Weekday Number

Date_ID	Week_ID	Week_Desc	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Year	ISNUMBER	Date Format (yyyy-mm-dd)	Year Half	Quarter & Year	Weekday Number	Weekday Name	Weekends
2017	20170701	26 Week 26	7	July	1	Q3	2017	2017 Week 26-2017	True	2017-07-01	2H	Q3-2017	6	Saturday	1
2017	20170702	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-02	2H	Q3-2017	7	Sunday	1
2017	20170703	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-03	2H	Q3-2017	1	Monday	0
2017	20170704	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-04	2H	Q3-2017	2	Tuesday	0
2017	20170705	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-05	2H	Q3-2017	3	Wednesday	0
2017	20170706	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-06	2H	Q3-2017	4	Thursday	0
2017	20170707	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-07	2H	Q3-2017	5	Friday	0
2017	20170708	27 Week 27	7	July	1	Q3	2017	2017 Week 27-2017	True	2017-07-08	2H	Q3-2017	6	Saturday	1
2017	20170709	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-09	2H	Q3-2017	7	Sunday	1
2017	20170710	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-10	2H	Q3-2017	1	Monday	0
2017	20170711	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-11	2H	Q3-2017	2	Tuesday	0
2017	20170712	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-12	2H	Q3-2017	3	Wednesday	0
2017	20170713	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-13	2H	Q3-2017	4	Thursday	0
2017	20170714	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-14	2H	Q3-2017	5	Friday	0
2017	20170715	28 Week 28	7	July	1	Q3	2017	2017 Week 28-2017	True	2017-07-15	2H	Q3-2017	6	Saturday	1
2017	20170716	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-16	2H	Q3-2017	7	Sunday	1
2017	20170717	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-17	2H	Q3-2017	1	Monday	0
2017	20170718	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-18	2H	Q3-2017	2	Tuesday	0
2017	20170719	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-19	2H	Q3-2017	3	Wednesday	0
2017	20170720	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-20	2H	Q3-2017	4	Thursday	0
2017	20170721	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-21	2H	Q3-2017	5	Friday	0
2017	20170722	29 Week 29	7	July	1	Q3	2017	2017 Week 29-2017	True	2017-07-22	2H	Q3-2017	6	Saturday	1
2017	20170723	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-23	2H	Q3-2017	7	Sunday	1
2017	20170724	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-24	2H	Q3-2017	1	Monday	0
2017	20170725	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-25	2H	Q3-2017	2	Tuesday	0
2017	20170726	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-26	2H	Q3-2017	3	Wednesday	0
2017	20170727	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-27	2H	Q3-2017	4	Thursday	0
2017	20170728	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-28	2H	Q3-2017	5	Friday	0
2017	20170729	30 Week 30	7	July	1	Q3	2017	2017 Week 30-2017	True	2017-07-29	2H	Q3-2017	6	Saturday	1
2017	20170730	31 Week 31	7	July	1	Q3	2017	2017 Week 31-2017	True	2017-07-30	2H	Q3-2017	7	Sunday	1

2) First objective is to add columns to the calendar table to capture the weekday number & name. Create a new table called Weekday Name.

Key objective 1 is now complete. Add columns to the calendar table to capture the weekday number & name.

File Home Help External Tools Table tools Column tools

Name: Weekends Data type: Text Summarization: Don't summarize Data category: Uncategorized Sort by column Data groups Manage relationships New column

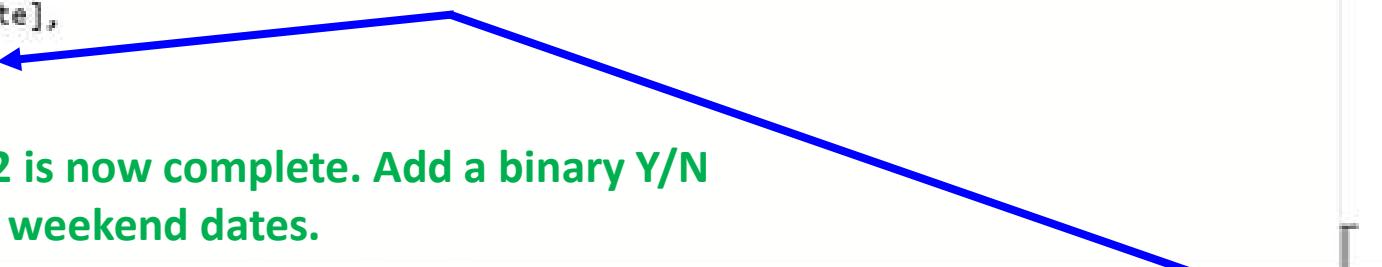
```

1 Weekends =
2 IF(
3   WEEKDAY(
4     'Calendar'[Transaction_Date],
5     2
6   ) IN {6,7},
7   "Y",
8   "N"
9 )

```

3) Objective 2 to add a binary Y/N column to flag weekend dates. Create a column called Weekends.

Key objective 2 is now complete. Add a binary Y/N column to flag weekend dates.



#	Date_ID	Week_ID	Week_Day	Month_ID	Month_Name	Quarter_ID	Quarter_Name	Year_ID	Week_Year	IsBusinessDay	Date Formatted (yyyy-mm-dd)	Year_Half	Quarter & Year	Sunday_Number	Weekday_Name	Weekends
1/2017	20170701	26	Week 26		July	1	Q3	2017	Week 26-2017	True	2017-07-01	2H	Q3-2017		Saturday	Y
1/2017	20170702	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-02	2H	Q3-2017		Sunday	Y
1/2017	20170703	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-03	2H	Q3-2017		Monday	N
1/2017	20170704	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-04	2H	Q3-2017		Tuesday	N
1/2017	20170705	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-05	2H	Q3-2017		Wednesday	N
1/2017	20170706	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-06	2H	Q3-2017		Thursday	N
1/2017	20170707	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-07	2H	Q3-2017		Friday	N
1/2017	20170708	27	Week 27		July	1	Q3	2017	Week 27-2017	True	2017-07-08	2H	Q3-2017		Saturday	Y
1/2017	20170709	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-09	2H	Q3-2017		Sunday	Y
1/2017	20170710	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-10	2H	Q3-2017		Monday	N
1/2017	20170711	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-11	2H	Q3-2017		Tuesday	N
1/2017	20170712	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-12	2H	Q3-2017		Wednesday	N
1/2017	20170713	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-13	2H	Q3-2017		Thursday	N
1/2017	20170714	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-14	2H	Q3-2017		Friday	N
1/2017	20170715	28	Week 28		July	2	Q3	2017	Week 28-2017	True	2017-07-15	2H	Q3-2017		Saturday	Y
1/2017	20170716	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-16	2H	Q3-2017		Sunday	Y
1/2017	20170717	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-17	2H	Q3-2017		Monday	N
1/2017	20170718	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-18	2H	Q3-2017		Tuesday	N
1/2017	20170719	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-19	2H	Q3-2017		Wednesday	N
1/2017	20170720	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-20	2H	Q3-2017		Thursday	N
1/2017	20170721	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-21	2H	Q3-2017		Friday	N
1/2017	20170722	29	Week 29		July	3	Q3	2017	Week 29-2017	True	2017-07-22	2H	Q3-2017		Saturday	Y
1/2017	20170723	30	Week 30		July	4	Q3	2017	Week 30-2017	True	2017-07-23	2H	Q3-2017		Sunday	Y
1/2017	20170724	30	Week 30		July	4	Q3	2017	Week 30-2017	True	2017-07-24	2H	Q3-2017		Monday	N

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data from Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box Insert Measure measure Quick visual Publish

Queries Calculations Share

4) Objective 3 to use a Matrix visual to show the percent of total sales for weekdays vs weekends.

Perform the below actions.

- Add a matrix to the report.
- Bring in the weekend field into the rows.

Maven Roasters - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name % of Total Sales % Format Percentage Data category Uncategorized \$ - % , . : New Quick measure measure

Home table Measure Table Layout Properties Calculators

1 % of Total Sales =
2 VAR AllSales =
3 CALCULATE(
4 [Customer Sales],
5 ALL(
6 'Sales by Store'
7)
8)
9 VAR Ratio =
10 DIVIDE(
11 [Customer Sales],
12 AllSales
13)
14 Return
15 Ratio

Visualizations > Fields >
Measure Table
Advanced CALCULATE
Advanced Time Intelligence
% of Total Sales
Customer Sales MoM Change
Customer Sales YoY Change
WkWk Change (4-5-6)
Iterations
Customer Sales (CONCATENATEX...
Relationship Functions
Bean to Goal
Beverage to Goal
Food to Goal
Merchandise to Goal
Table & Filter Functions
% of All Baked
% of Store-Level Sales
% of Total Baked (ALLSELECTED)
% of Total Sold (ALLSELECTED)
Tips & Best Practices
Quantity Sold to Females

6) Change the formatting accordingly over here.

5) Create a new measure & call it % of Total Sales.

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals

New measure measure Calculations

Publish

Visualizations Fields

Filters

Weekends: % of Total Sales

	Weekends: % of Total Sales
N	71.41%
Y	28.59%
Total	100.00%

7) Add the % of Total Sales to the matrix i.e. values area.

Key objective 3 is now complete. Use a Matrix visual to show the percent of total sales for weekdays vs weekends.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time

Page 9 of 9

Common Time Intelligence Functions

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

Time Intelligence functions allow you to define and compare custom time periods:

Performance-to-Date

Functions commonly used to calculate performance through the current date

Common Examples:

- DATESYTD
- DATESQTD
- DATESMTD

Time Period Shift

Functions commonly used to compare performance between specific periods

Common Examples:

- SAMEPERIODLASTYEAR
- DATEADD
- PARALLELPERIOD
- PREVIOUSYEAR
(QUARTER/MONTH/DAY)
- NEXTYEAR
(QUARTER/MONTH/DAY)

Running Total

Functions commonly used to calculate running totals or moving averages

Common Examples:

- DATESINPERIOD

Use these common time intelligence patterns for YTD, period-over-period, or running total calculations:

Performance
To-Date

= CALCULATE(Measure, DATESYTD(Calendar[Date]))

 Use DATESQTD for Quarters or DATESMTD for Months

Previous
Period

= CALCULATE(Measure, DATEADD(Calendar[Date], -1, MONTH))



Select an interval (DAY, MONTH, QUARTER, or YEAR) and the
of intervals to compare (i.e. previous month, rolling 10-day)

Running
Total

= CALCULATE(Measure,
DATESINPERIOD(Calendar[Date], MAX(Calendar[Date]), -10, DAY))





PRO TIP:

To calculate a **moving average**, use a running total calculation and divide by the number of intervals

PARALLELPERIOD

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

PARALLELPERIOD()

Returns a column of dates from a parallel period, by shifting the dates specified forward or backward in time based on a given interval (month/quarter/year)

=PARALLELPERIOD(Date, NumberOfIntervals, Interval)

The name of a column containing dates or a one column table containing dates

Example:

- Calendar[Transaction_Date]

Number of Intervals (positive or negative);
If a decimal is supplied, number is rounded to nearest whole integer

Example:

- 1, 2, 3, etc.
- -1,-2,-3 etc.
- -1.4 (-1) -1.5 (-2)

Interval value can only be:

- Month
- Quarter
- Year



HEY THIS IS IMPORTANT!

PARALLELPERIOD computes the *entire* period in the interval (i.e. entire year, quarter, etc.). Values in total rows may not reflect the expected total if partial periods are present!

Automatic Date
Tables

Calendar
Functions

Date Formatting

Time Intelligence
Functions

Week-Based
Calculations

```
1 Last Quarter's Sales (PARALLELPERIOD) =  
2 CALCULATE(  
3     [Customer Sales],  
4     PARALLELPERIOD(  
5         'Calendar'[Transaction_Date],  
6         -1,  
7         QUARTER  
8     )  
9 )
```

Year_ID	Customer Sales	Last Quarter's Sales (PARALLELPERIOD)
2017	\$1,678,074.11	1,158,966.35
January	\$81,645.09	
February	\$76,273.99	
March	\$99,154.43	
April	\$119,309.01	257,273.51
May	\$157,208.99	257,273.51
June	\$166,899.78	257,273.51
July	\$157,968.55	443,417.78
August	\$154,485.32	443,417.78
September	\$145,821.19	443,417.78
October	\$169,223.54	458,275.06
November	\$179,999.30	458,275.06
December	\$169,884.92	458,275.06
2018	\$1,916,544.75	1,901,863.54
January	\$141,284.63	\$19,107.76
February	\$124,030.84	\$19,107.76
March	\$143,475.17	\$19,107.76
April	\$149,780.08	408,790.64
May	\$173,257.84	408,790.64
June	\$174,349.34	408,790.64
July	\$164,985.34	497,387.26
August	\$161,108.60	497,387.26
September	\$150,483.94	497,387.26
October	\$174,293.02	476,577.88
November	\$184,401.31	476,577.88
December	\$175,094.64	476,577.88
2019	\$658,086.02	959,528.19
January	\$146,863.61	533,788.97
February	\$129,473.65	533,788.97
March	\$149,401.96	533,788.97
April	\$232,346.80	425,739.22
Total	\$4,252,704.88	4,020,358.08

Total returned for the **entire previous quarter** on each row

Year-level total = Q4 2018 + Q1 2019

Overall total is "missing" April 2019

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Last Quarter's Sales... Format Currency Data category Uncategorized New Quick measure measure

Home table Measure Table \$ - % , ; : Properties Calculations

1 Last Quarter's Sales (PARALLELPERIOD) =
2 CALCULATE(
3 [Customer Sales],
4 PARALLELPERIOD(
5 'Calendar'[Transaction_Date],
6 -1,
7 QUARTER
8)
9)

2) Change the formatting accordingly over here.

Visualizations Fields

Last Quarter's Sales (PARALLELPERIOD)

Measure Table Advanced Time Intelligence Last Quarter's Sales (PARALLELPERIOD)

Values Add data fields here

Drill through

Cross-report Off

Keep all filters On

Add drill-through fields here

1) Create a new measure & call it Last Quarter's Sales (PARALLELPERIOD). -1 used for previous quarter sales.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

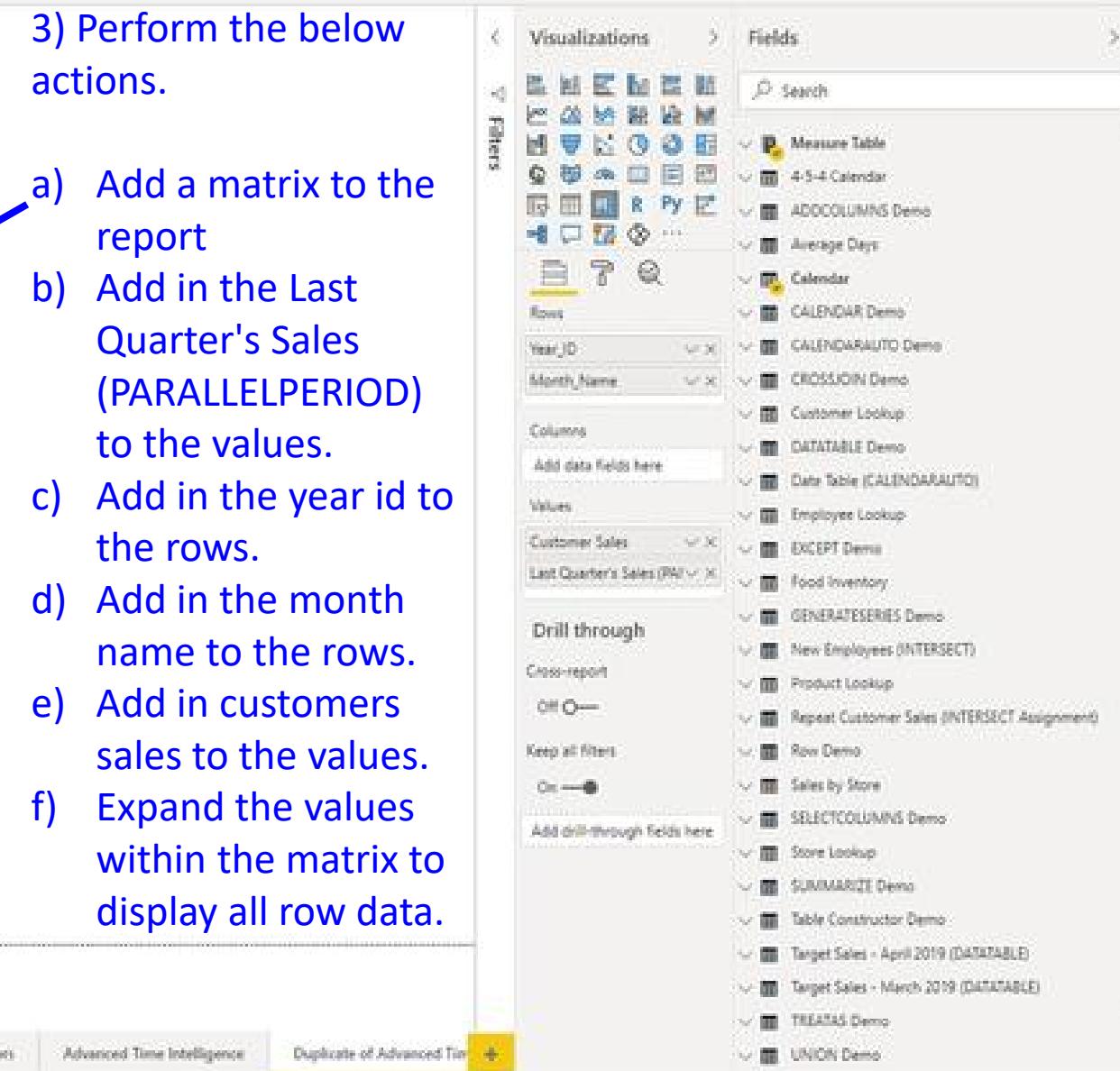
Sign in    

- File**
- Home** selected
- Insert**
- Modeling**
- View**
- Help**
- External Tools**
- Format**
- Data / Drill**

 Cut
 Copy
 Format painter
 Get data
 Excel datasets
 Power BI Server
 SQL Server
 Enter data
 Recent sources
 Transform refresh data
 New visual
 Text box
 More
 New measure
 Quick measure
 Publish

3) Perform the below actions.

- Add a matrix to the report
- Add in the Last Quarter's Sales (**PARALLELPERIOD**) to the values.
- Add in the year id to the rows.
- Add in the month name to the rows.
- Add in customers sales to the values.
- Expand the values within the matrix to display all row data.



The screenshot shows the Power BI Desktop interface. On the left, there is a matrix visualization titled "Customer Sales: Last Quarter's Sales (PARALLELPERIOD)". The matrix has "Year_ID" as the column header and "Month_Name" as the row header. The values represent sales figures for each month in a given year. To the right of the matrix is the "Visualizations" pane, which lists various data types and functions such as Measure Table, 4-5-4 Calendar, ADDCOLUMNS Demo, etc. Below the Visualizations pane is the "Fields" pane, which shows the current fields in the model: Year_ID, Month_Name, Customer Sales, and Last Quarter's Sales (PA). The "Customer Sales" field is currently selected. At the bottom of the screen, there is a navigation bar with links like "Tips & Best Practices", "Scalar Functions", "Advanced CALCULATE", "Table & Filter Functions", "Join", "Relationship Functions", "Iterators", "Advanced Time Intelligence", and "Duplicate of Advanced Time Intelligence".

PREVIOUSQUARTER

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

PREVIOUSQUARTER

Returns a table containing a column of all dates from the previous quarter, based on the first date in the date range specified

=PREVIOUSQUARTER(Dates)

The name of a column containing dates or a one column table containing dates

Example:

- Calendar[Transaction_Date]

HEY THIS IS IMPORTANT!

PREVIOUSQUARTER works similarly to PARALLELPERIOD (QUARTER) and computes the entire prior period (but handles totals differently)

Year_ID	Customer Sales	Last Quarter's Sales (PREVIOUSQUARTER)
2017	\$1,258,224.88	\$257,273.51
May	\$115,942.28	\$257,273.51
June	\$106,899.78	\$257,273.51
July	\$157,968.55	\$443,417.76
August	\$154,485.32	\$443,417.76
September	\$145,821.19	\$443,417.76
October	\$169,221.54	\$458,275.06
November	\$179,999.30	\$458,275.06
December	\$169,884.92	\$458,275.06
2018	\$1,918,544.75	\$518,107.76
January	\$141,264.63	\$518,107.76
February	\$124,030.84	\$518,107.76
March	\$143,475.17	\$518,107.76
April	\$148,780.08	\$408,790.64
May	\$173,257.84	\$408,790.64
June	\$174,349.34	\$408,790.64
July	\$164,905.34	\$497,387.26
August	\$161,108.60	\$497,387.26
September	\$150,483.94	\$497,387.26
October	\$174,293.02	\$476,577.88
November	\$184,401.31	\$476,577.88
December	\$175,094.64	\$476,577.88
2019	\$458,084.02	\$533,788.97
January	\$146,363.81	\$533,788.97
February	\$129,473.65	\$533,788.97
March	\$146,401.96	\$533,788.97
April	\$232,346.80	\$425,799.22
Total	\$3,832,855.65	\$257,273.51

Transaction_Date
is after 5/9/2017

When first date specified is 5/9/2017, totals for May and June reflect Jan-Mar (Q1) 2017

2018 Total row reflects the total for Q4 2017

With no date context, total is based on first visible date in the table (5/9/2017)

Maven Roadmap - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Last Quarter's Sales... Format Currency Data category Uncategorized New measure Quick measure measure

Home table Measure Table \$ - % , ; Properties Calculations

1 Last Quarter's Sales (PREVIOUSQUARTER) =
2 CALCULATE(
3 [Customer Sales],
4 PREVIOUSQUARTER(
5 'Calendar'[Transaction_Date]
6)
7)

October \$169,223.54 \$458,275.06
November \$179,999.30 \$458,275.06
December \$169,884.92 \$458,275.06
2018 \$1,916,544.75 \$1,901,863.54
January \$141,284.63 \$519,107.76
February \$124,030.84 \$519,107.76
March \$143,475.17 \$519,107.76
April \$149,780.08 \$408,790.64
May \$173,257.84 \$408,790.64
June \$174,349.34 \$408,790.64
July \$164,985.34 \$497,387.26
August \$161,108.60 \$497,387.26
September \$150,483.94 \$497,387.26
October \$174,293.02 \$476,577.88
November \$184,401.31 \$476,577.88
December \$175,094.64 \$476,577.88
2019 \$658,086.02 \$959,528.19
January \$146,883.61 \$533,788.97
February \$129,473.65 \$533,788.97
March \$149,401.96 \$533,788.97
April \$232,346.00 \$425,739.22
Total \$4,252,704.88 \$4,020,358.08

Visualizations Fields
Last Quarter's Sales (PREVIOUSQUARTER)
Measure Table Advanced Time Intelligence Last Quarter's Sales (PREVIOUSQUARTER)

Drill through
Cross-report Off
Keep all filters On
Add drill-through fields here

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time

2) Change the formatting accordingly over here.

1) Create a new measure & call it Last Quarter's Sales (PREVIOUSQUARTER).



3) Add the Last Quarter's Sales (PREVIOUSQUARTER) measure to the matrix.

Weekends % of Total Sales

N	71.41%
Y	28.59%
Total	100.00%

Year_ID	Customer Sales	Last Quarter's Sales (PARALLELPERIOD)	Last Quarter's Sales (PREVIOUSQUARTER)
2017	\$1,678,074.11	\$1,158,966.35	
January	\$81,845.09		
February	\$76,273.99		
March	\$99,154.43		
April	\$119,309.01	\$257,273.51	\$257,273.51
May	\$157,208.99	\$257,273.51	\$257,273.51
June	\$166,899.78	\$257,273.51	\$257,273.51
July	\$157,968.55	\$443,417.78	\$443,417.78
August	\$154,485.32	\$443,417.78	\$443,417.78
September	\$145,821.19	\$443,417.78	\$443,417.78
October	\$169,223.54	\$458,275.06	\$458,275.06
November	\$179,999.30	\$458,275.06	\$458,275.06
December	\$169,884.92	\$458,275.06	\$458,275.06
2018	\$1,916,544.75	\$1,901,863.54	\$519,107.76
January	\$141,284.63	\$519,107.76	\$519,107.76
February	\$124,030.84	\$519,107.76	\$519,107.76
March	\$143,475.17	\$519,107.76	\$519,107.76
April	\$149,700.00	\$408,790.64	\$408,790.64
May	\$173,257.84	\$408,790.64	\$408,790.64
June	\$174,349.34	\$408,790.64	\$408,790.64
July	\$164,985.34	\$497,387.26	\$497,387.26
August	\$161,108.60	\$497,387.26	\$497,387.26
September	\$150,483.94	\$497,387.26	\$497,387.26
October	\$174,293.02	\$476,577.88	\$476,577.88
November	\$184,401.31	\$476,577.88	\$476,577.88
December	\$175,094.64	\$476,577.88	\$476,577.88
2019	\$658,086.02	\$859,528.19	\$533,788.97
January	\$146,863.61	\$533,788.97	\$533,788.97
February	\$129,473.65	\$533,788.97	\$533,788.97
March	\$149,401.96	\$533,788.97	\$533,788.97
April	\$232,346.80	\$425,739.22	\$425,739.22
Total	\$4,252,704.88	\$4,020,358.08	

Visualizations Fields

Filters

Measure Table Advanced Time Intelligence Last Quarter's Sales (PREVIOUSQUARTER)

Row

Year_ID Month_Name

Columns

Add data fields here

Values

Customer Sales Last Quarter's Sales (PAI) Last Quarter's Sales (PBI)

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

actions Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence +

Page 9 of 9



5) These quarter values are refreshed the entire Q1 total is provided in June 2017 & so fourth including the total quarter row.

4) Add transaction date to the filters & apply these parameters.

The screenshot shows a Power BI desktop interface with several components:

- Left Panel:** Displays a table titled "Weekends % of Total Sales" with three rows: X (60.81%), Y (24.40%), and Total (85.25%).
- Middle Panel:** A data grid table with columns: Year_ID, Customer_Sales, Last Quarter's Sales (PARALLELPERIOD), and Last Quarter's Sales (PREVIOUSQUARTER). The table spans from 2017 to 2019, listing months and their sales figures. The last row is a "Total" row.
- Right Panel:** The Power BI visualizations pane, which includes sections for Filters, Visualizations, Fields, and Drill through.
- Filters Section:** Contains filters for "Customer_Sales" (is null), "Last Quarter's Sales (PA)" (is null), "Last Quarter's Sales (PR)" (is null), "Year_ID" (is null), and "Month_Name" (is null).
- Advanced Filtering Parameters:** A detailed view of the "Transaction_Date" filter, set to "is on or after 18/06/2017" at 12:00 AM. It includes options for "And" and "Or" logic.
- Visualizations Section:** Shows various visualization icons like charts and maps.
- Fields Section:** Lists fields such as "transaction_date", "transaction_id", "transaction_time", and "Unsold_Partner".

SAMEPERIODLASTYEAR

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

SAMEPERIODLASTYEAR()

Returns a set of dates in the current selection from the previous year

= SAMEPERIODLASTYEAR(Dates)

The name of a column containing dates or a one column table containing dates

Example:

- Calendar[Transaction_Date]

HEY THIS IS IMPORTANT!

Pay attention to totals and filter context!
SAMEPERIODLASTYEAR is equivalent to
DATEADD('Dates', -1, YEAR) and may return values which aren't always intuitive or straightforward to understand



Year_ID	YTD Sales	Last Year's Sales (SAMEPERIODLASTYEAR)
2018	\$1,916,544.75	\$1,445,900.03
March	\$408,790.64	\$25,099.43
April	\$558,570.72	\$118,309.01
May	\$731,828.56	\$157,208.99
June	\$906,177.90	\$166,899.78
July	\$1,071,163.24	\$157,968.55
August	\$1,232,271.84	\$154,485.32
September	\$1,382,755.78	\$145,821.19
October	\$1,557,048.80	\$169,223.54
November	\$1,741,450.11	\$179,999.30
December	\$1,916,544.75	\$169,804.92
2019	\$658,086.02	\$558,570.72
January	\$146,863.61	\$141,284.63
February	\$276,337.26	\$124,030.84
March	\$425,739.22	\$143,475.17
April	\$658,086.02	\$149,780.08
Total	\$658,086.02	\$2,004,470.75

Total Sales for 3/24/2017 – 3/31/2017

2019 total has filter context of 1/1/2019 – 4/30/2019, and returns 2018 YTD total through April

Automatic Date
Tables

Calendar
Functions

Date Formatting

Time Intelligence
Functions

Week-Based
Calculations

SAMEPERIODLASTYEAR is really just a shortcut for DATEADD and YEAR:

How it's written:

```
1 Store 3 Sales - Last Year =  
2 CALCULATE(  
3     [Total Sales],  
4     SAMEPERIODLASTYEAR('Calendar'[Transaction_Date])  
5 )
```

How it's interpreted:

```
1 Store 3 Sales - Last Year =  
2 CALCULATE(  
3     [Total Sales],  
4     DATEADD(  
5         'Calendar'[Transaction_Date],  
6         -1,  
7         YEAR)  
8 )
```

Maven Reactors - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name Last Years Sales (SA... Format Currency Data category Uncategorized New measure measure Quick measure

Home table Measure Table \$ - % , . : Properties Calculations

1 Last Years Sales (SAMEPERIODLASTYEAR) =
2 CALCULATE([Customer Sales], SAMEPERIODLASTYEAR('Calendar'[Transaction_Date]))
3
4
5
6
7)

February \$124,030.84 \$519,107.76 \$519,107.76
March \$143,475.17 \$519,107.76 \$519,107.76
April \$149,780.08 \$408,790.64 \$408,790.64
May \$173,257.84 \$408,790.64 \$408,790.64
June \$174,349.34 \$408,790.64 \$408,790.64
July \$164,905.34 \$497,387.26 \$497,387.26
August \$161,108.60 \$497,387.26 \$497,387.26
September \$150,483.94 \$497,387.26 \$497,387.26
October \$174,293.02 \$476,577.88 \$476,577.88
November \$184,401.31 \$476,577.88 \$476,577.88
December \$175,094.64 \$476,577.88 \$476,577.88
2019 \$658,086.02 \$959,528.19 \$533,788.97
January \$146,863.61 \$533,788.97 \$533,788.97
February \$129,473.65 \$533,788.97 \$533,788.97
March \$149,401.96 \$533,788.97 \$533,788.97
April \$232,346.80 \$425,739.22 \$425,739.22
Total \$3,629,332.21 \$4,020,158.08 \$267,273.51

Visualizations Fields
Last Years Sales
Measure Table
Advanced Time Intelligence
Last Years Sales (SAMEPERIODLASTYEAR)

1) Create a new measure & call it Last Years Sales (SAMEPERIODLASTYEAR).
2) Change the formatting accordingly over here.

Actions Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence +

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in - X

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter Transform Refresh data New visual Text box More Publish

Openers Data Queries Insert Calculations Share

Weekends % of Total Sales

	Weekends % of Total Sales
W	60.85%
Y	24.40%
Total	85.25%

Year_ID Customer Sales Last Quarter's Sales (SAMEPERIODLASTYEAR) Last Quarter's Sales (PREVIOUSQUARTER) Last Years Sales (SAMEPERIODLASTYEAR)

Year_ID	Customer Sales	Last Quarter's Sales (SAMEPERIODLASTYEAR)	Last Quarter's Sales (PREVIOUSQUARTER)	Last Years Sales (SAMEPERIODLASTYEAR)
2017	\$1,050,701.44	\$1,158,946.35	\$257,273.51	
June	\$13,318.62	\$257,273.51	\$257,273.51	
July	\$157,968.53	\$443,417.78	\$443,417.78	
August	\$154,485.32	\$443,417.78	\$443,417.78	
September	\$143,821.19	\$443,417.78	\$443,417.78	
October	\$169,223.54	\$458,275.06	\$458,275.06	
November	\$179,999.30	\$458,275.06	\$458,275.06	
December	\$169,884.92	\$458,275.06	\$458,275.06	
2018	\$1,916,544.75	\$1,901,863.54	\$519,107.76	\$1,678,074.11
January	\$141,204.61	\$519,107.76	\$519,107.76	\$81,845.09
February	\$124,000.84	\$519,107.76	\$519,107.76	\$76,273.99
March	\$143,475.17	\$519,107.76	\$519,107.76	\$99,154.43
April	\$149,780.08	\$408,790.64	\$408,790.64	\$118,309.01
May	\$173,257.84	\$408,790.64	\$408,790.64	\$157,208.99
June	\$174,349.34	\$408,790.64	\$408,790.64	\$166,899.78
July	\$164,985.34	\$497,387.26	\$497,387.26	\$157,988.55
August	\$161,100.60	\$497,387.26	\$497,387.26	\$154,485.32
September	\$156,483.94	\$497,387.26	\$497,387.26	\$145,821.19
October	\$174,291.02	\$476,577.88	\$476,577.88	\$169,223.54
November	\$184,401.31	\$476,577.88	\$476,577.88	\$179,999.30
December	\$173,094.64	\$476,577.88	\$476,577.88	\$169,884.92
2019	\$658,696.02	\$959,528.19	\$533,788.97	\$558,570.72
January	\$145,823.61	\$533,788.97	\$533,788.97	\$141,204.61
February	\$129,473.63	\$533,788.97	\$533,788.97	\$124,000.84
March	\$149,401.96	\$533,788.97	\$533,788.97	\$143,475.17
April	\$122,346.80	\$423,739.22	\$423,739.22	\$149,780.08
Total	\$3,625,332.21	\$4,020,358.08	\$257,273.51	\$2,236,644.83

Visualizations Fields

Last Years Sales

Measure Table Advanced Time Intelligence Last Years Sales (SAMEPERIODLASTYEAR)

Row: Year_ID, Month_Name

Column: Add data fields here

Values: Customer Sales, Last Quarter's Sales (SAMEPERIODLASTYEAR), Last Quarter's Sales (PREVIOUSQUARTER), Last Years Sales (SAMEPERIODLASTYEAR)

Drill through: Off

Cross-report: Off

Keep all filters: On

Add drill-through fields here

3) Add the Last Years Sales (SAMEPERIODLASTYEAR) measure to the matrix.

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data from Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More Quick measure measure

Queries Data Calculations Share

Weekends % of Total Sales

	% of Total Sales
W	42.85%
F	24.45%
Total	65.25%

Year_ID Customer Sales Last Quarter's Sales (PALLELPERIOD) Last Quarter's Sales (PREVIOUSQUARTER) Last Year's Sales (SAMEPERIODLASTYEAR)

Year_ID	Customer Sales	Last Quarter's Sales (PALLELPERIOD)	Last Quarter's Sales (PREVIOUSQUARTER)	Last Year's Sales (SAMEPERIODLASTYEAR)
2017	\$1,899,791.44	\$1,738,996.35	\$217,273.51	
January	\$73,273.51	\$217,273.51	\$257,273.51	
February	\$157,961.11	\$443,417.78	\$443,417.78	
March	\$154,485.55	\$443,417.78	\$443,417.78	
April	\$143,821.19	\$443,417.78	\$443,417.78	
May	\$169,223.54	\$498,275.06	\$498,275.06	
June	\$179,999.10	\$498,275.06	\$498,275.06	
July	\$169,884.92	\$498,275.06	\$498,275.06	
August	\$1,896,344.73	\$1,901,860.54	\$319,107.76	\$1,678,074.11
September	\$241,254.63	\$519,107.76	\$519,107.76	\$51,045.09
October	\$124,203.54	\$519,107.76	\$519,107.76	\$76,273.99
November	\$143,475.17	\$519,107.76	\$519,107.76	\$99,154.43
December	\$149,782.08	\$400,790.64	\$400,790.64	\$118,328.01
2018	\$1,899,791.44	\$1,899,791.44	\$498,275.06	
January	\$73,273.51	\$1,738,996.35	\$217,273.51	
February	\$157,961.11	\$443,417.78	\$443,417.78	
March	\$154,485.55	\$443,417.78	\$443,417.78	
April	\$143,821.19	\$443,417.78	\$443,417.78	
May	\$169,223.54	\$498,275.06	\$498,275.06	
June	\$179,999.10	\$498,275.06	\$498,275.06	
July	\$169,884.92	\$498,275.06	\$498,275.06	
August	\$1,896,344.73	\$1,901,860.54	\$319,107.76	\$1,678,074.11
September	\$241,254.63	\$519,107.76	\$519,107.76	\$51,045.09
October	\$124,203.54	\$519,107.76	\$519,107.76	\$76,273.99
November	\$143,475.17	\$519,107.76	\$519,107.76	\$99,154.43
December	\$149,782.08	\$400,790.64	\$400,790.64	\$118,328.01
2019	\$1,898,086.02	\$993,788.19	\$333,788.97	\$958,576.72
January	\$146,881.81	\$533,788.97	\$533,788.97	\$741,284.83
February	\$129,473.69	\$533,788.97	\$533,788.97	\$124,030.84
March	\$140,451.96	\$533,788.97	\$533,788.97	\$143,473.17
April	\$232,346.80	\$425,739.22	\$425,739.22	\$148,785.08
Total	\$3,625,332.21	\$4,620,158.09	\$257,273.51	\$2,236,644.83

Filters

Visualizations

Fields

Last Years Sales

Measure Table

Advanced Time Intelligence

Last Years Sales (SAMEPERIODLASTYEAR)

Search

Filters on this visual

Customer Sales is (All)

Last Quarter's Sales (PA... is (All)

Last Quarter's Sales (PR... is (All)

Last Year's Sales (SAM... is (All)

Month_Name

Columns Add data fields here

Values Customer Sales is (All)

Last Quarter's Sales (PALLELPERIOD) is (All)

Last Quarter's Sales (PREVIOUSQUARTER) is (All)

Last Year's Sales (SAMEPERIODLASTYEAR) is (All)

Add data fields here

Drill through

Cross-report

On Off

Remove filter

Keep all filters

On

Transaction Date is on or after 18/06/2017

Time type Advanced filtering

Show items when the value is on or after 18/06/2017

12 00 AM

And Or

Duplicate of Advanced

4) Click here to clear the filters:

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure Quick measure Publish

Weekends % of Total Sales

W	71.41%
Y	28.59%
Total	100.00%

Year_ID Customer Sales Last Quarter's Sales (PARALLELPERIOD) Last Quarter's Sales (PREVIOUSQUARTER) Last Years Sales (SAMEPERIODLASTYEAR)

Year_ID	Customer Sales	Last Quarter's Sales (PARALLELPERIOD)	Last Quarter's Sales (PREVIOUSQUARTER)	Last Years Sales (SAMEPERIODLASTYEAR)
2017	\$1,678,074.11	\$1,158,966.35		
January	\$81,845.09			
February	\$91,273.99			
March	\$99,154.43			
April	\$119,309.01	\$257,273.51	\$257,273.51	
May	\$117,208.99	\$257,273.51	\$257,273.51	
June	\$116,899.78	\$257,273.51	\$257,273.51	
July	\$117,968.55	\$443,417.78	\$443,417.78	
August	\$114,483.32	\$443,417.78	\$443,417.78	
September	\$115,821.19	\$443,417.78	\$443,417.78	
October	\$119,223.54	\$458,275.06	\$458,275.06	
November	\$118,999.30	\$458,275.06	\$458,275.06	
December	\$119,884.92	\$458,275.06	\$458,275.06	
2018	\$1,186,544.75	\$1,901,863.54	\$519,107.76	\$1,678,074.11
January	\$141,284.63	\$519,107.76	\$519,107.76	\$81,845.09
February	\$124,039.84	\$519,107.76	\$519,107.76	\$76,273.99
March	\$143,473.17	\$519,107.76	\$519,107.76	\$99,154.43
April	\$149,789.08	\$408,790.64	\$408,790.64	\$119,309.01
May	\$173,257.84	\$408,790.64	\$408,790.64	\$157,208.99
June	\$174,349.34	\$408,790.64	\$408,790.64	\$166,899.78
July	\$164,943.34	\$497,387.26	\$497,387.26	\$157,968.55
August	\$161,108.60	\$497,387.26	\$497,387.26	\$154,483.32
September	\$150,483.94	\$497,387.26	\$497,387.26	\$143,821.19
October	\$174,293.02	\$476,577.88	\$476,577.88	\$169,223.54
November	\$184,401.31	\$476,577.88	\$476,577.88	\$179,899.30
December	\$173,054.64	\$476,577.88	\$476,577.88	\$169,884.92
2019	\$658,086.02	\$959,521.19	\$533,788.97	\$558,570.72
January	\$146,863.61	\$533,788.97	\$533,788.97	\$141,284.63
February	\$129,473.65	\$533,788.97	\$533,788.97	\$124,039.84
March	\$149,401.95	\$533,788.97	\$533,788.97	\$143,473.17
April	\$232,346.80	\$425,739.22	\$425,739.22	\$149,789.08
Total	\$4,252,704.81	\$4,020,358.08		\$2,236,644.83

Visualizations Fields

Last Years Sales

Measure Table Advanced Time Intelligence Last Years Sales (SAMEPERIODLASTYEAR)

Filters Rows Year_ID Month_Name

Columns Add data fields here

Values Customer Sales Last Quarter's Sales (PARALLELPERIOD) Last Quarter's Sales (PREVIOUSQUARTER) Last Years Sales (SAMEPERIODLASTYEAR)

Drill through Cross-report Off Keep all filters On Add drill-through fields here

5) 2018 January is looking at last years sales i.e. \$81,845.09

Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced

Page 9 of 9

ASSIGNMENT: Time Periods

KEY OBJECTIVES

- 1) Using time intelligence functions to calculate total sales for the previous month, as well as the percent change month-over-month.
- 2) Calculate the year-over-year change in sales from April 2018 vs. April 2019.

Maven Roasters - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Customer Sales Last Month

Home table Measure Table

Format: Currency Data category: Uncategorized

\$ - % 9.99 2

Structure Properties Calculations

2) Change the formatting accordingly over here.

1 Customer Sales Last Month (PARALLELPERIOD) =
2 CALCULATE(
3 [Customer Sales],
4 PARALLELPERIOD(
5 'Calendar'[Transaction_Date],
6 -1,
7 MONTH
8)
9)

Visualizations Fields

Customer Sales Last Month (PARALLELPERIOD)

Measure Table Advanced Time Intelligence Customer Sales Last Month (PARALLELPERIOD)

Values Add data fields here

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

1) Objective 1 is to use time intelligence functions to calculate total sales for the previous month, as well as the percent change month-over-month. Create a new measure & call it Customer Sales Last Month (PARALLELPERIOD).

Month	Sales
April	\$149,780.00
May	\$173,257.84
June	\$174,343.34
July	\$164,985.34
August	\$161,108.60
September	\$150,483.94
October	\$174,293.02
November	\$184,401.31
December	\$175,094.64
2019	\$658,086.62
January	\$146,663.61
February	\$129,473.65
March	\$149,401.96
April	\$232,346.80
Total	\$4,252,704.88

ATE Table & Filter Functions Aggregations Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time Intelligence +

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get Data - Excel Power BI datasets - SQL Server Enter date Recent sources - Transform Refresh data - New visual Text box More visuals - New measure measure Quick Publish

Queries Data Calculations Share

Weekends % of Total Sales

N	71.41%
Y	28.59%
Total	100.00%

Customer Sales Customer Sales Last Month (PARALLELPERIOD)

Year_ID	Customer Sales	Customer Sales Last Month (PARALLELPERIOD)
2017	\$1,673,074.11	\$1,598,189.19
January	\$81,045.09	
February	\$76,273.99	\$81,045.09
March	\$99,154.43	\$76,273.99
April	\$119,309.01	\$99,154.43
May	\$157,208.99	\$119,309.01
June	\$164,099.78	\$157,208.99
July	\$157,968.55	\$164,099.78
August	\$154,485.32	\$157,968.55
September	\$145,821.19	\$154,485.32
October	\$169,223.54	\$145,821.19
November	\$179,999.30	\$169,223.54
December	\$169,084.92	\$179,999.30
2018	\$1,916,544.75	\$1,911,335.03
January	\$141,284.63	\$169,084.92
February	\$124,030.84	\$141,284.63
March	\$143,475.17	\$124,030.84
April	\$149,780.00	\$143,475.17
May	\$173,257.84	\$149,780.00
June	\$174,349.34	\$173,257.84
July	\$164,985.34	\$174,349.34
August	\$161,108.60	\$164,985.34
September	\$150,401.94	\$161,108.60
October	\$174,293.02	\$150,401.94
November	\$184,401.31	\$174,293.02
December	\$175,094.00	\$184,401.31
2019	\$658,011.02	\$650,833.86
January	\$173,863.61	\$175,094.00
February	\$129,473.65	\$173,863.61
March	\$149,401.96	\$129,473.65
April	\$122,146.80	\$149,401.96
Total	\$4,252,704.88	\$4,020,358.08

The previous month values appear here correctly.

Visualizations

Filters

Fields

Customer Sales Last Month (PARALLELPERIOD)

Measure Table

Advanced Time Intelligence

Customer Sales Last Month (PARALLELPERIOD)

Rows

Year_ID

Month_Name

Columns

Add data fields here

Values

Customer Sales

Customer Sales Last Month (PARALLELPERIOD)

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

3) Add the Customer Sales Last Month (PARALLELPERIOD) measure to the matrix.

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name Customer Sales Mo... % Format Percentage \$ - % . , : Data category Uncategorized New Quick measure measure

Home table Measure Table Structure Formatting Properties Calculations

1 Customer Sales MoM % Change =
 2 DIVIDE([Customer Sales] - [Customer Sales Last Month (PARALLELPERIOD)],
 [Customer Sales Last Month (PARALLELPERIOD)],
 BLANK(),
)

4) First objective part two is to use time intelligence functions to calculate the percent change month-over-month. Create a new measure & call it Customer Sales MoM % Change.

5) Change the formatting accordingly over here.

Month	Year	Customer Sales	Customer Sales Last Month (PARALLELPERIOD)	MoM % Change
October	2018	\$169,223.54	\$145,821.19	-14.1%
November	2018	\$179,999.30	\$169,223.54	+6.4%
December	2018	\$169,884.92	\$179,999.30	-6.0%
	2019	\$1,816,544.75	\$1,911,335.03	-5.2%
January	2019	\$141,284.63	\$169,884.92	-19.4%
February	2019	\$124,030.84	\$141,284.63	-15.7%
March	2019	\$143,475.17	\$124,030.84	+15.3%
April	2019	\$149,780.08	\$143,475.17	+9.0%
May	2019	\$173,237.84	\$149,780.08	+18.8%
June	2019	\$174,589.34	\$173,237.84	+1.9%
July	2019	\$164,963.34	\$174,589.34	-6.2%
August	2019	\$161,108.60	\$164,963.34	-6.8%
September	2019	\$150,483.84	\$161,108.60	-6.7%
October	2019	\$174,293.02	\$150,483.84	+13.9%
November	2019	\$184,401.31	\$174,293.02	+6.0%
December	2019	\$175,094.64	\$184,401.31	-5.8%
	2020	\$658,986.02	\$499,833.86	+31.6%
January	2020	\$146,963.61	\$658,986.02	-79.8%
February	2020	\$129,473.65	\$146,963.61	-17.4%
March	2020	\$149,401.66	\$129,473.65	+14.6%
April	2020	\$232,346.00	\$149,401.66	+53.8%
Total		\$4,252,704.88	\$4,020,358.08	+5.8%

Visualizations Fields

Customer Sales

Measure Table

Advanced Time Intelligence

Customer Sales Last Month (PARALLELPERIOD)

Customer Sales MoM % Change

Customer Sales YoY % Change

Iterators

% of Customer Sales (CONCATENATEX)

Rank of Customer Sales (RANKX)

Rank of Rounded Customer Sales (RANKEQX)

Rounded Customer Sales

Scalar Measures

Customer Sales LY (COALESCE)

Table & Filter Functions

Customer Sales (ALL(Except Assignment))

Customer Sales (ALL(Except))

Customer Sales (AllSelected)

Tips & Best Practices

Customer Sales

Customer Sales (Last Year)

Customer Sales LY (BLANK)

Customer Lookup

Customer Sales (High/Low)

EXCEPT Demo

Customer Sales (High/Low)

Report Customer Sales (INTERSECT Assignment)

customer_id

Profit

Revenue

Row Demo

Customer Sales

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Queries Data Calculations Share

Weekends % of Total Sales

N	71.41%
Y	28.59%
Total	100.00%

Year_ID Customer Sales Customer Sales Last Month (PARALLELPERIOD) Customer Sales MoM % Change

Year_ID	Customer Sales	Customer Sales Last Month (PARALLELPERIOD)	Customer Sales MoM % Change
2017	\$1,678,974.11	\$1,598,189.19	11.26%
January	\$1,045.09		
February	\$76,273.99	\$81,843.09	-6.81%
March	\$99,154.43	\$76,273.99	30.00%
April	\$119,309.01	\$99,154.43	20.39%
May	\$157,208.99	\$119,309.01	31.77%
June	\$166,899.78	\$137,208.99	6.18%
July	\$157,968.55	\$166,899.78	-5.29%
August	\$154,485.32	\$157,968.55	-2.21%
September	\$145,821.19	\$154,485.32	-5.81%
October	\$169,223.54	\$145,821.19	16.05%
November	\$179,999.30	\$169,223.54	6.07%
December	\$169,884.92	\$179,999.30	-5.62%
2018	\$1,914,544.75	\$1,911,335.03	0.37%
January	\$141,284.63	\$160,884.92	-16.84%
February	\$124,030.84	\$141,284.63	-12.21%
March	\$143,473.17	\$124,030.84	15.68%
April	\$149,760.08	\$143,473.17	4.39%
May	\$173,237.84	\$149,760.08	15.67%
June	\$174,546.34	\$173,237.84	0.63%
July	\$164,985.34	\$174,546.34	-5.27%
August	\$161,108.60	\$164,985.34	-2.35%
September	\$150,483.94	\$161,108.60	-6.58%
October	\$174,293.03	\$150,483.94	15.02%
November	\$184,401.31	\$174,293.03	5.80%
December	\$173,094.64	\$184,401.31	-5.00%
2019	\$658,086.02	\$620,833.86	9.53%
January	\$146,853.61	\$173,094.64	-16.12%
February	\$129,473.23	\$146,853.61	-11.84%
March	\$134,401.98	\$129,473.23	15.29%
April	\$232,346.80	\$134,401.98	53.52%
Total	\$4,252,704.88	\$4,020,358.08	5.78%

Visualizations Fields

Customer Sales

Measure Table

Advanced Time Intelligence

- Customer Sales Last Month (PARALLELPERIOD)
- Customer Sales MoM % Change
- Customer Sales YoY % Change

Iterators

- % of Customer Sales (CONCATENATE)
- Rank of Customer Sales (RANK)
- Rank of Rounded Customer Sales (RANKEQ)
- Rounded Customer Sales

Scalar Measures

- Customer Sales LY (COALESCE)

Table & Filter Functions

- Customer Sales (ALL/EXCEPT Assignment)
- Customer Sales (ALLEXCEPT)
- Customer Sales (ALLSELECTED)

Tips & Best Practices

- Customer Sales
- Customer Sales (Last Year)
- Customer Sales LY (ISBLANK)

Customer Lookup

- Customer Sales (High/Low)

EXCEPT Demo

- Customer Sales (High/Low)

Repeat Customer Sales (INTERSECT Assignment)

- customer_id
- Profit
- Revenue

Row Demo

- Customer Sales

Month over month % change display correctly here.

Key objective 1 is now complete. Using time intelligence functions to calculate total sales for the previous month, as well as the percent change month-over-month.

6) Add the Customer Sales MoM % Change measure to the matrix.

Maven Readers - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name Customer Sales YoY % Change
Home table Measure Table

Structure Formatting Properties Calculations

8) Change the formatting accordingly over here.

1 Customer Sales YoY % Change =
2 VAR LastYearsSales =
3 CALCULATE(
4 [Customer Sales],
5 SAMEPERIODLASTYEAR(
6 'Calendar'[Transaction_Date]
7)
8)
9 VAR Ratio =
10 DIVIDE(
11 ([Customer Sales] - LastYearsSales),
12 LastYearsSales,
13 "
14)
15 Return
16 Ratio

Visualizations Fields

Customer Sales

Measure Table

Advanced Time Intelligence

- Customer Sales Last Month (PARALLEL)
- Customer Sales MoM % Change
- Customer Sales YoY % Change

Iterators

- % of Customer Sales (CONCATENATE)
- Rank of Customer Sales (RANK)
- Rank of Rounded Customer Sales (RANL)
- Rounded Customer Sales

Scalar Measures

- Customer Sales LY (COALESCE)

Table & Filter Functions

- Customer Sales (ALL/EXCEPT Assignment)
- Customer Sales (ALL/EXCEPT)
- Customer Sales (ALLSELECTED)

Tips & Best Practices

- Customer Sales
- Customer Sales (Last Year)
- Customer Sales LY (SEALN)

Customer Lookup

- Customer Sales (High/Low)

EXCEPT Demo

- Customer Sales (High/Low)

Repeat Customer Sales (INTERSECT Assignment)

- customer_id
- Profit
- Revenue

Row Demo

Customer Sales

Page 10 of 10

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Queries Data Calculations Share

Weekends % of Total Sales

	Year_ID	Customer Sales	Customer Sales - Last Month (PARALLELPERIOD)	Customer Sales - MoM % Change	Customer Sales - YOY % Change
N	2017	\$1,678,074.11	\$1,598,189.19	11.26%	
Y	January	\$81,845.09			
	February	\$76,273.99	\$81,845.09	-6.81%	
	March	\$99,154.43	\$76,273.99	30.00%	
	April	\$119,309.01	\$99,154.43	20.33%	
	May	\$157,208.99	\$119,309.01	21.77%	
	June	\$166,899.78	\$157,208.99	6.16%	
	July	\$157,968.55	\$166,899.78	-5.85%	
	August	\$154,485.32	\$157,968.55	-4.21%	
	September	\$145,821.19	\$154,485.32	-5.81%	
	October	\$169,223.54	\$145,821.19	16.03%	
	November	\$179,999.30	\$169,223.54	6.37%	
	December	\$169,884.92	\$179,999.30	-5.62%	
T	2018	\$1,916,544.75	\$1,811,395.00	0.27%	14.21%
	January	\$141,284.63	\$169,884.92	-18.84%	72.62%
	February	\$124,030.64	\$141,284.63	-12.21%	62.61%
	March	\$143,475.17	\$124,030.64	15.68%	44.70%
	April	\$149,780.08	\$143,475.17	4.39%	25.54%
	May	\$173,237.84	\$149,780.08	15.67%	10.21%
	June	\$174,349.34	\$173,237.84	0.63%	4.46%
	July	\$164,985.34	\$174,349.34	-5.17%	4.44%
	August	\$161,108.60	\$164,985.34	-2.35%	4.29%
	September	\$150,483.94	\$161,108.60	-6.59%	3.20%
	October	\$174,293.03	\$150,483.94	15.87%	3.00%
	November	\$184,401.31	\$174,293.03	5.80%	2.45%
	December	\$175,094.04	\$184,401.31	-5.00%	3.07%
T	2019	\$658,085.02	\$600,831.94	9.53%	17.82%
	January	\$146,883.61	\$713,094.64	-18.12%	3.95%
	February	\$129,473.65	\$146,883.61	-11.64%	4.39%
	March	\$149,421.42	\$129,473.65	15.39%	4.13%
	April	\$149,421.96	\$149,421.42	0.32%	33.12%
T	Total	\$4,252,794.88	\$4,020,358.00	5.78%	90.14%

Key objective 2 is now complete. Calculate the year-over-year change in sales from April 2018 vs. April 2019.

Year over year % change display correctly here.

Visualizations

Fields

Customer Sales

Measure Table

Advanced Time Intelligence

- Customer Sales - Last Month (PARALLELPERIOD)
- Customer Sales - MoM % Change
- Customer Sales - YOY % Change

Iterators

- % of Customer Sales (CONCATENATE)
- Rank of Customer Sales (RANK)
- Rank of Rounded Customer Sales (RANRANK)
- Rounded Customer Sales

Scalar Measures

- Customer Sales LY (COALESCE)

Table & Filter Functions

- Customer Sales (ALL(Except Assignment))
- Customer Sales (ALL(Except))
- Customer Sales (AllSelected)

Tips & Best Practices

- Customer Sales
- Customer Sales (Last Year)
- Customer Sales LY (ISBLANK)

Customer Lookup

- Customer Sales (High/Low)

EXCEPT Demo

- Customer Sales (High/Low)

Report Customer Sales (INTERSECT Assignment)

- customer_id
- Profit
- Revenue

Row Demo

Customer Sales

9) Add the Customer Sales MoM % Change measure to the matrix.

Week-based Calculations

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

Week-based DAX calculations (i.e. previous week sales, same week last year) can be especially challenging for several reasons:

- Can't use standard DAX functions (i.e. `PARALLELPERIOD`, `SAMEPERIODLASTYEAR`)
- Weeks can start on various days (*Sunday, Monday, etc.*)
- There can be partial weeks in any given month, quarter or year (i.e. partial 53rd week)
- Weeks can be grouped differently based on different fiscal calendars (i.e. 5-4-4, 4-5-4)



PRO TIP:

Using a 5-4-4 or 4-5-4 fiscal calendar (common in the retail industry) can enable some week-based calculations, but you still can't use standard DAX time intelligence functions



Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

2020																							
February				May				August				November											
Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	14	3	4	5	6	7	8	9	21	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9	21	10	11	12	13	14	15	16	28	9	10	11	12	13	14	15
3	4	5	6	7	8	9	10	21	17	18	19	20	21	22	23	29	15	16	17	18	19	20	21
4	5	6	7	8	9	10	11	21	23	24	25	26	27	28	29	30	17	18	19	20	21	22	23
5	6	7	8	9	10	11	12	21	24	25	26	27	28	29	30	31	18	19	20	21	22	23	24
March				June				September				December											
Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	14	21	1	2	3	4	5	6	21	30	21	1	2	3	4	5
2	3	4	5	6	7	8	9	21	19	20	21	22	23	24	25	22	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	21	15	16	17	18	19	20	21	23	13	14	15	16	17	18	19
4	5	6	7	8	9	10	11	21	22	23	24	25	26	27	28	29	14	15	16	17	18	19	20
5	6	7	8	9	10	11	12	21	21	22	23	24	25	26	27	28	15	16	17	18	19	20	21
April				July				October				January 21											
Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	14	20	21	22	23	24	25	26	21	27	28	29	30	1	2	3
2	3	4	5	6	7	8	9	21	28	29	30	1	2	3	4	22	5	6	7	8	9	10	11
3	4	5	6	7	8	9	10	21	15	16	17	18	19	20	21	23	12	13	14	15	16	17	18
4	5	6	7	8	9	10	11	21	19	20	21	22	23	24	25	26	13	14	15	16	17	18	19
5	6	7	8	9	10	11	12	21	26	27	28	29	30	31	1	14	21	22	23	24	25	26	27
August				November				December				January 21											
Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S	Wk	S	M	T	W	T	F	S
1	2	3	4	5	6	7	8	14	21	22	23	24	25	26	27	21	1	2	3	4	5	6	7
2	3	4	5	6	7	8	9	21	28	29	30	1	2	3	4	22	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	21	16	17	18	19	20	21	22	23	13	14	15	16	17	18	19
4	5	6	7	8	9	10	11	21	19	20	21	22	23	24	25	26	14	15	16	17	18	19	20
5	6	7	8	9	10	11	12	21	25	26	27	28	29	30	31	1	15	16	17	18	19	20	21

In a 4-4-5 fiscal calendar, the months in each quarter contain exactly 4, 4, and 5 weeks



PROS:

- Standardizes the number of days in each week (**7**)
- Standardizes the number of weeks in each year (**52**)



CONS:

- Does not have standard start & end dates for years, quarters, or months
- Cannot be used with standard DAX time intelligence functions

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools

Get Data > Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Weekends % of Total Sales

	Year	Customer Sales	Customer Sales Last Month	Customer Sales MoM % Change	Customer Sales YoY % Change
N	2017	\$1,678,074.11	\$1,508,189.19	+11.34%	
Y					
Total		\$4,252,704.88	\$4,020,358.08	+5.78%	+9.14%

Visualizations Fields

Filters

1) Go to get data & select Text/CSV load in the 4-5-4 Calendar.csv file than click on open.

File Home Insert Modeling View Help External Tools

Get Data > Text/CSV

Open

Organizer New folder

This PC

3D Objects

Desktop

Documents

Downloads

Music

Pictures

Videos

Windows10Pro C:

Data1 (D)

Data3 (E)

Data2 (F)

Data4 (I)

wdVMW (O)

Name Date modified Type Size

4-5-4 Calendar 16/09/2020 14:15 Microsoft Excel CSV 64 KB

File name: 4-5-4 Calendar

Text files (*.txt; *.csv; *.prn)

Open Cancel

December \$175,094.64 \$184,401.31 -5.05% 3.07%

2019 \$658,086.02 \$600,833.86 9.53% 17.82%

January \$146,863.61 \$175,094.64 -16.12% 3.95%

February \$129,473.65 \$146,863.61 -11.84% 4.39%

March \$149,401.96 \$129,473.65 15.39% 4.13%

April \$232,346.80 \$149,401.96 55.52% 55.13%

Total \$4,252,704.88 \$4,020,358.08 5.78% 9.14%

Actions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I...

Sign in

Page 10 of 10

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools

Power Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Queries Data Calculations Share

Weekends: % of Total Sales

N	71.41%
Y	28.59%
Total	100.00%

4-5-4 Calendar.csv

File Origin: E:\0001.Unicode (UTF-8) Delimiter: Comma Data Type Detection: Based on first 200 rows

Date	FiscalYear	FiscalQuarter	FiscalMonthNumber	FiscalMonthOfQuarter	FiscalWeekOffset	DayOfWeek	FiscalMonthName
01/01/2017	2017	1	1	1	1	1	January
02/01/2017	2017	1	1	1	1	2	February
03/01/2017	2017	1	1	1	1	3	February
04/01/2017	2017	1	1	1	1	4	February
05/01/2017	2017	1	1	1	1	5	February
06/01/2017	2017	1	1	1	1	6	February
07/01/2017	2017	1	1	1	1	7	February
08/01/2017	2017	1	1	1	1	1	February
09/01/2017	2017	1	1	1	1	2	February
10/01/2017	2017	1	1	1	1	3	February
11/01/2017	2017	1	1	1	1	4	February
12/01/2017	2017	1	1	1	1	5	February
13/01/2017	2017	1	1	1	1	6	February
14/01/2017	2017	1	1	1	1	7	February
15/01/2017	2017	1	1	1	1	1	February
16/01/2017	2017	1	1	1	1	2	February
17/01/2017	2017	1	1	1	1	3	February
18/01/2017	2017	1	1	1	1	4	February
19/01/2017	2017	1	1	1	1	5	February
20/01/2017	2017	1	1	1	1	6	February
21/01/2017	2017	1	1	1	1	7	February
22/01/2017	2017	1	1	1	1	1	February
23/01/2017	2017	1	1	1	1	2	February
24/01/2017	2017	1	1	1	1	3	February

Load **Transform Data** **Cancel**

	January	February	March	April
January	\$190,000.00	\$173,036.04	-10.74%	3.03%
February	\$120,473.65	\$146,863.61	-11.84%	4.39%
March	\$149,401.96	\$129,473.65	15.39%	4.13%
April	\$232,346.80	\$149,401.96	55.52%	55.13%
Total	\$4,252,704.88	\$4,020,358.08	5.78%	90.14%

Visualizations **Fields**

Search

Filters

Measure Table
4-5-4 Calendar
ADDCOLUMNS D...
Average Days
Calendar
CALENDAR Demo
CROSSJOIN Demo
Customer Lookup
DATATABLE Demo
Date Table (CALE...
Employee Lookup
EXCEPT Demo
Food Inventory
GENERATESERI...
New Employees D...
Product Lookup
Repeat Customer...
Row Demo
Sales by Store
SELECTCOLUMNS...
Store Lookup
SUMMARIZE Demo
Table Constructor...
Target Sales - Apr...
Target Sales - Ma...
TREATAS Demo
UNION Demo

2) Click on Load data.

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Help External Tools

Get data v Excel Power BI datasets SQL Server data Recent sources Transform Refresh data Manage relationships Q&A Language schema Publish

Calendars Customer Lookup Employee Lookup Product Lookup Store Lookup 4-5-4 Calendar Sales by Store Food Inventory UNION Demo Measure Table

Properties Fields

Select one or more model objects to set their properties.

3) Connect date column within the 4-5-4 Calendar table to the transaction date column within the Sales by Store table.

The screenshot shows the Power BI Data Model view. On the left, there's a diagram of the data flow between various tables: 'Calendars' connects to 'Customer Lookup', 'Employee Lookup', 'Product Lookup', 'Store Lookup', and 'Sales by Store'. 'Customer Lookup' connects to 'Food Inventory'. 'Product Lookup' connects to 'Food Inventory'. 'Store Lookup' connects to 'Food Inventory'. 'Sales by Store' connects to 'Food Inventory'. 'Food Inventory' connects to 'UNION Demo' and 'Measure Table'. In the center, the '4-5-4 Calendar' table is selected, and its 'Date' column is highlighted with a yellow selection bar. In the bottom-left corner, the 'transaction_date' column in the 'Sales by Store' table is also highlighted with a yellow selection bar. A large blue arrow points from the 'transaction_date' column to the 'Date' column, indicating the connection being established. The right side of the screen shows the 'Properties' and 'Fields' panes, and the status bar at the bottom indicates 'All tables'.

Maven Roasters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Help External Tools

Get data Get data from Power BI datasets Server Enter data Recent sources Transform Refresh data Manage relationships Manage roles View Q&A Language setup Publish

Dashboard Data Queries Relationships Security DAX Share

Properties Fields

Search

Select one or more model objects to set their properties.

4) Connect date column within the 4-5-4 Calendar table to the transaction date column within the Food Inventory table.

The screenshot shows the Power BI Data Model view. On the left, there is a grid of tables: Calendar, Customer Lookup, Employee Lookup, Product Lookup, Store Lookup, 4-5-4 Calendar, Sales by Store, Food Inventory, and Measure Table. The 'transaction_date' column in the 'Food Inventory' table and the 'Date' column in the '4-5-4 Calendar' table are both selected, indicated by yellow selection bars. A large blue arrow points from the 'transaction_date' column towards the 'Date' column. The 'Properties' and 'Fields' panes are visible on the right side of the interface.

Previous Fiscal Week

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

To calculate performance for the previous fiscal week, you can use **DATEADD** with an interval of **-7 days**:

```
1 Last Week's Sales 4-5-4 (DATEDADD) =  
2 CALCULATE(  
3     [Customer Sales],  
4     DATEADD(  
5         '4-5-4 Calendar'[Date],  
6         -7,  
7         DAY  
8     )  
9 )
```

This works with 4-5-4 or 4-4-5 fiscal calendars, but can break down when applied to standard calendars where years often contain **partial weeks**



PRO TIP:

Use **DATEADD** with an interval of **-364 days** to compare the same week last year!

Maven Roasters - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name Last Week's Sales 4-5-4
Home table Measure Table

Format Currency \$ - % .0 2 Data category Uncategorized New measure Quick measure

Structure Properties Calculations

1 Last Week's Sales 4-5-4 (DATEADD) =
2 CALCULATE([Customer Sales],
DATEADD('4-5-4 Calendar'[Date],
-7,
DAY))

2) Change the formatting accordingly over here.

1) Go to the report view & create a measure called Last Week's Sales 4-5-4 (DATEADD).

Visualizations Fields

Last Week's Sales 4-5-4 (DATEADD)

Measure Table Advanced Time Intelligence Last Week's Sales 4-5-4 (DATEADD)

Add data fields here

Values Drill through Cross-report Off On Keep all filters On Add drill-through fields here

Page 11 of 11

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Insert Calculations Publish

Visualizations Fields

Search Measure Table 4-5-4 Calendar Date DayName DayOfMonthNumber

FiscalYear
3203656

Drill through Cross-report Off Keep all filters On Add drill-through fields here

Remove field Rename Conditional formatting Remove conditional formatting Don't summarize Sum Average Minimum Maximum Count (Distinct) Count Standard deviation Variance Median Show value as New quick measure Customer Lookup DATETIME Demo Date Table (CALENDARAUTO) Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo New Employees (INTERSECT)

FiscalYear field values don't show up correctly here as defaulted to a sum value.

3) Perform the below actions.

- Add a matrix to the report.
- Bring FiscalYear field into the values.
- For the FiscalYear field within the values are change the options to Don't summarise for correct output.

Maven Roadmap - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure

New Calculations Share

4) Values now appear correctly from previous slide steps applied.

FiscalYear

2017
2018
2019

Visualizations

Filters

Volumen

FiscalYear

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Fields

Measure Table

4-5-4 Calendar

- Date
- DayName
- DayOfMonthNumber
- DayOfWeek
- FiscalMonthName
- FiscalMonthNumber
- FiscalMonthOfQuarter
- FiscalMonthYear
- FiscalQuarter
- FiscalQuarterYear
- FiscalWeekOffset
- FiscalYear**

ADDCOLUMNS Demo

Average Days

Calendar

CALENDAR Demo

CALENDARAUTO Demo

CROSSJOIN Demo

Customer Lookup

DATABLE Demo

Date Table (CALENDARAUTO)

Employee Lookup

EXCEPT Demo

Food Inventory

GENERATESERIES Demo

New Employees (INTERSECT)

Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data = Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box Insert Measure measure Quick measure Publish

Queries Calculations Share

Visualizations Fields

Search

Measure Table

4-5-4 Calendar

Date DayName DayOfMonthNumber DayOfWeek

Remove field Rename Move Conditional formatting Remove conditional formatting

Drill through

Cross-report Off

Keep all filters On

Don't summarise

Sum Average Minimum Maximum Count (Distinct) Count Standard deviation Variance Median New quick measure Show items with no data

New group

Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo New Employees (INTERSECT)

FiscalYear FiscalWeekOfYear

FiscalYear	FiscalWeekOfYear
2017	1
2017	2
2017	3
2017	4
2017	5
2017	6
2017	7
2017	8
2017	9
2017	10
2017	11
2017	12
2017	13
2017	14
2017	15
2017	16
2017	17
2017	18
2017	19

Relationship functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period +

Page 11 of 31

The screenshot shows a Microsoft Power BI Desktop interface. In the center, there is a data grid titled 'FiscalYear FiscalWeekOfYear' containing 19 rows of data from 2017, numbered 1 to 19. A blue arrow points from the bottom-left towards this grid. To the right of the grid is a context menu for the 'FiscalWeekOfYear' field. The menu includes options like 'Remove field', 'Rename', 'Move', 'Conditional formatting', 'Remove conditional formatting', 'Drill through', 'Cross-report', 'Keep all filters', and 'Don't summarise'. The 'Don't summarise' option is highlighted with a yellow box and has a blue arrow pointing to it from the bottom-right. The background shows other Power BI components like 'Visualizations' and 'Fields'.

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Queries Calculations Share

Visualizations Fields

Measure Table Advanced Time Intelligence Customer Sales Last Month (P
Last QTD Sales (4-5-6) Last Quarter's Sales (P
Last Quarter's Sales (PREVIOUSQUARTER) Last Week's Sales 4-5-4 (DATEADD)
Last Years Sales (SAMEPERIODLASTYEAR) Tips & Best Practices Customer Sales Last Year ADDCOLUMNS Demo last_name Employee Lookup last_name New Employees (INTERSECT) last_name

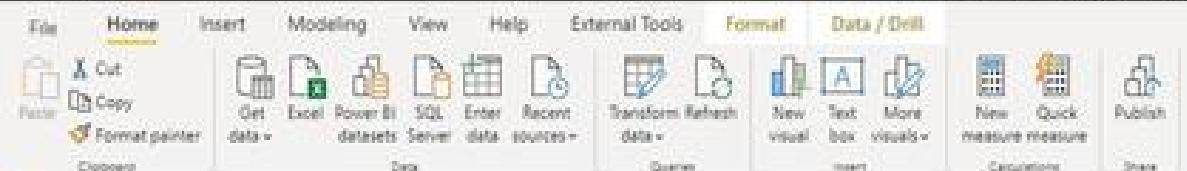
FiscalYear FiscalWeekOffYear Last Week's Sales 4-5-4 (DATEADD)

FiscalYear	FiscalWeekOffYear	Last Week's Sales 4-5-4 (DATEADD)
2017	1	\$17,872.83
2017	2	\$19,531.03
2017	3	\$20,325.92
2017	4	\$26,401.29
2017	5	\$21,381.41
2017	6	\$23,342.89
2017	7	\$21,424.50
2017	8	\$21,833.58
2017	9	\$25,925.84
2017	10	\$28,369.79
2017	11	\$29,395.41
2017	12	\$28,193.59
2017	13	\$31,128.58
2017	14	\$33,996.26
2017	15	\$34,158.39
2017	16	\$37,637.28
2017	17	\$33,297.71
2017	18	\$37,422.53
2017	19	\$4,166,743.44
Total		\$4,166,743.44

6) Perform the below actions.

- Add the Last Week's Sales 4-5-4 (DATEADD) field to the matrix.
- Add the Customer Sales field to the matrix.

Relationship functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I Previous Fiscal Period +



7) The %91,961.44 value appear due to it being based on a different calendar. Fiscal Week 1 of 2017 is actually in February we have January sales that don't exist in this calendar. You'd have to apply some filter here if you are running into that same case that's why the \$91k value is displaying here.

8) These values run through in a succinct pattern as the FiscalWeekOfYear column have standardised the values.

FiscalYear	FiscalWeekOfYear	Customer Sales	Last Week's sales 4-5-6 (DATEADD)
2017	1	\$91,961.44	\$17,872.85
2017	2	\$19,531.01	\$19,531.01
2017	3	\$20,325.92	\$19,531.01
2017	4	\$20,401.20	\$20,125.92
2017	5	\$22,381.41	\$20,401.20
2017	6	\$13,342.89	\$22,381.41
2017	7	\$23,424.50	\$23,342.89
2017	8	\$21,832.18	\$23,424.50
2017	9	\$25,925.94	\$21,832.18
2017	10	\$28,569.79	\$25,925.94
2017	11	\$29,396.61	\$28,569.79
2017	12	\$28,155.39	\$29,396.61
2017	13	\$31,128.38	\$28,155.39
2017	14	\$35,998.36	\$31,128.38
2017	15	\$38,158.29	\$35,998.36
2017	16	\$37,697.39	\$38,158.29
2017	17	\$33,297.71	\$37,697.39
Total:		\$4,252,704.88	\$4,160,743.44

Visualizations

Fields

customer sales

- Advanced Time Intelligence
 - Customer Sales Last Month (PARALLELPERIOD)
 - Customer Sales MoM % Change
 - Customer Sales YoY % Change
- Iterators
 - % of Customer Sales (CONCATENATEX)
 - Rank of Customer Sales (RANKX)
 - Rank of Rounded Customer Sales (RANKX)
 - Rounded Customer Sales
- Scalar Measures
 - Customer Sales LY (CALCSCD)
- Table & Filter Functions
 - Customer Sales (ALLEXCEPT Assignment)
 - Customer Sales (ALLEXCEPT)
 - Customer Sales (ALLSELECTED)
- Tips & Best Practices
 - Customer Sales
 - Customer Sales (Last Year)
 - Customer Sales LY (SBLANDO)
- Customer Lookup
 - Customer Sales (High/Low)
- EXCEPT Demo
 - Customer Sales (High/Low)
- Repeat Customer Sales (INTERSECT Assignment)
 - I: customer_id
 - I: Profit
 - I: Revenue
- Row Demo
 - I: Customer Sales



9) No issue from going from week 52 2017 to week 1 of 2018.

FiscalYear	FiscalWeekOffset	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)
2017	43	\$39,968.19	\$39,968.19
2017	44	\$39,834.88	\$39,968.19
2017	45	\$37,939.89	\$39,834.88
2017	46	\$29,866.23	\$37,939.89
2017	47	\$32,954.10	\$29,866.23
2017	48	\$34,237.50	\$32,954.10
2017	49	\$34,149.15	\$34,237.50
2017	50	\$27,611.35	\$34,149.15
2018	1	\$29,634.13	\$27,611.35
2018	2	\$31,272.64	\$29,634.13
2018	3	\$31,356.59	\$31,272.64
2018	4	\$30,823.53	\$31,356.59
2018	5	\$31,351.67	\$30,823.53
2018	6	\$33,149.27	\$31,351.67
2018	7	\$34,946.40	\$33,149.27
2018	8	\$30,652.90	\$34,946.40
2018	9	\$31,652.60	\$30,652.90
2018	10	\$36,128.37	\$31,652.60
Total:		\$4,252,704.88	\$4,160,743.44

Visualizations

Fields

customer sales

- Advanced Time Intelligence
 - Customer Sales Last Month (PARALLEL PERIOD)
 - Customer Sales MoM % Change
 - Customer Sales YoY % Change
- Iterators
 - % of Customer Sales (CONCATENATEX Assignment)
 - Rank of Customer Sales (RANKX)
 - Rank of Rounded Customer Sales (RANKX)
 - Rounded Customer Sales
- Scalar Measures
 - Customer Sales IF (COALESCE)
- Table & Filter functions
 - Customer Sales (ALLEXCEPT Assignment)
 - Customer Sales (ALLEXCEPT)
 - Customer Sales (ALLSELECTED)
- Tips & Best Practices
 - Customer Sales
 - Customer Sales (Last Year)
 - Customer Sales IF (SBLANK)
- Customer Lookup
 - Customer Sales (High/Low)
- EXCEPT Demo
 - Customer Sales (High/Low)
- Repeat Customer Sales (INTERSECT Assignment)
 - Sales
 - Profit
 - Revenue
- Row Demo
 - Customer Sales

Fiscal Period to Date

Automatic Date Tables

Calendar Functions

Date Formatting

Time Intelligence Functions

Week-Based Calculations

Instead of a standard DAX performance-to-date functions (i.e. `DATESYTD`, `DATESMTD`), you can use a combination of **CALCULATE**, **MAX**, and **HASONEVALUE**:

```
1 QTD Sales (4-5-4) =  
2 VAR MaxDate = MAX('4-5-4 Calendar'[Date])  
3 VAR MaxPeriod = MAX('4-5-4 Calendar'[FiscalQuarterYear])  
4 VAR Output =  
5 IF(  
6   HASONEVALUE(  
7     '4-5-4 Calendar'[FiscalQuarter]  
8   ),  
9   CALCULATE(  
10    [Customer Sales],  
11    '4-5-4 Calendar'[Date] <= MaxDate,  
12    '4-5-4 Calendar'[FiscalQuarterYear] = MaxPeriod  
13  ),  
14  "-"  
15 )  
16 RETURN  
17 Output
```

Here we're calculating *QTD sales* based on a 4-5-4 fiscal calendar

You can replace *FiscalQuarter* with the fiscal year or fiscal month for *YTD* and *MTD* calculations

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name: QTD Sales (4-5-4) Date category: Unassigned

Home table: Measure Table \$ - % ✓

Structure Formatting Properties Calculations

2) Change the formatting accordingly over here.

1 QTD Sales (4-5-4) =
2 VAR MaxDate = MAX('4-5-4 Calendar'[Date])
3 VAR MaxPeriod = MAX('4-5-4 Calendar'[FiscalQuarterYear])
4 VAR Output =
5 IF(
6 HASONEVALUE(
7 '4-5-4 Calendar'[FiscalQuarterYear])
8),
9 CALCULATE(
10 [Customer Sales],
11 '4-5-4 Calendar'[Date] <= MaxDate,
12 '4-5-4 Calendar'[FiscalQuarterYear] = MaxPeriod
13),
14 "
15)
16 RETURN
17 Output

Visualizations Fields

Values Fields

FiscalYear
FiscalWeekOffset
Customer Sales
Last Week's Sales 4-3-4 (DATEADD)

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Relationship functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period +

1) Create a new measure & call it QTD Sales (4-5-4).

Maven Roadmap - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools

Cut Copy Format painter

Get data - Excel Power BI datasets - SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Publish

Dashboard Data Query Insert Calculations Share

FiscalYear FiscalWeekOfficer Customer Sales Last Week's Sales: 4-5-4 (DATACADO)

2017 45 \$39,968.19 \$36,893.13
2017 46 \$39,836.88 \$39,968.19
2017 47 \$37,939.89 \$39,836.88
2017 48 \$39,866.23 \$37,939.89
2017 49 \$32,584.10 \$29,866.23
2017 50 \$34,237.58 \$32,584.10
2017 51 \$34,140.15 \$34,237.58
2017 52 \$37,611.33 \$34,140.15
2018 1 \$39,636.13 \$37,611.33
2018 2 \$31,272.64 \$29,636.13
2018 3 \$33,356.59 \$31,272.64
2018 4 \$30,823.53 \$33,356.59
2018 5 \$31,951.67 \$30,823.53
2018 6 \$31,149.27 \$31,951.67
2018 7 \$34,946.40 \$33,149.27
2018 8 \$33,652.98 \$34,946.40
2018 9 \$31,652.63 \$33,652.98
2018 10 \$36,128.37 \$31,652.63
2018 11 \$37,604.14 \$36,128.37
2018 12 \$35,803.52 \$37,604.14
2018 13 \$33,711.72 \$35,803.52
2018 14 \$39,020.56 \$33,711.72
2018 15 \$42,011.67 \$39,020.56
2018 16 \$41,317.48 \$42,011.67
2018 17 \$36,059.63 \$41,317.48
2018 18 \$38,145.87 \$36,059.63
2018 19 \$42,327.98 \$39,145.87
2018 20 \$42,420.39 \$42,327.98
2018 21 \$39,982.90 \$42,420.39
Total \$4,252,704.88 \$4,160,743.44

FiscalYear

3) Perform the below actions.

a) Add a new matrix to the report.

b) Create a date hierarchy with the Fiscal Periods; Add in FiscalMonthName & date to FiscalYear to generate the FiscalYear Hierarchy.

c) Add the FiscalYear Hierarchy to the rows.

Visualizations Fields

Measure Table

4-5-4 Calendar

Date DayName DayOfMonthNumber DayOfWeek FiscalMonthName FiscalMonthNumber FiscalMonthOfQuarter FiscalMonthYear FiscalQuarter FiscalQuarterYear FiscalWeekOfficer FiscalYear FiscalMonthName Date

FiscalYear Hierarchy

Name: '4-5-4 Calendar'[FiscalYear]

Keep all filters

Add drill-through fields here

Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter Delete Data Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Queries Data Calculations Share

FiscalYear FiscalWeekOffYear Customer Sales Last Week's Sales 4-5-4 (DATEADD)

FiscalYear	FiscalWeekOffYear	Customer	Sales	Last Week's Sales 4-5-4 (DATEADD)
2017	45	\$39,968.19	\$38,893.13	
2017	46	\$39,836.88	\$39,968.19	
2017	47	\$37,939.89	\$39,836.88	
2017	48	\$29,866.23	\$37,939.89	
2017	49	\$32,584.10	\$29,866.23	
2017	50	\$34,237.58	\$32,584.10	
2017	51	\$34,140.15	\$34,237.58	
2017	52	\$27,611.29	\$34,140.15	
2018	1	\$29,636.13	\$27,611.29	
2018	2	\$31,272.64	\$29,636.13	
2018	3	\$33,356.59	\$31,272.64	
2018	4	\$30,821.53	\$33,356.59	
2018	5	\$31,951.67	\$30,821.53	
2018	6	\$31,149.27	\$31,951.67	
2018	7	\$34,946.40	\$31,149.27	
2018	8	\$30,652.88	\$34,946.40	
2018	9	\$31,652.63	\$30,652.88	
2018	10	\$36,128.37	\$31,652.63	
2018	11	\$37,604.14	\$36,128.37	
2018	12	\$33,803.52	\$37,604.14	
2018	13	\$33,711.72	\$33,803.52	
2018	14	\$39,020.96	\$33,711.72	
2018	15	\$42,011.67	\$39,020.96	
2018	16	\$41,517.48	\$42,011.67	
2018	17	\$36,059.63	\$41,517.48	
2018	18	\$39,145.87	\$36,059.63	
2018	19	\$42,327.98	\$39,145.87	
2018	20	\$42,420.39	\$42,327.98	
2018	21	\$39,982.90	\$42,420.39	
Total		\$4252,704.88	\$4,160,743.44	

4) Add in QTD Sales (4-5-4) to the values.

Visualizations Fields

Filters

Row: FiscalYear Hierarchy, FiscalYear, FiscalMonthName, Date

Columns: Add data fields here

Values: QTD Sales (4-5-4)

Drill through: FiscalYear Hierarchy, FiscalYear, FiscalMonthName, Date

Cross-report: Off

Keep all filters: On

Add drill-through fields here

This value displays due to sales occurring before the fiscal year starting.

Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period +

Page 11 of 11

Maven Readers - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI SQL Enter data Recent sources Transform Refresh data New visual Text box More measure Quick measure Publish

Power BI datasets Server

Queries Calculations Share

FiscalYear FiscalWeekOfYear Customer Sales Last Week (4-5-4) (ADOCAL)

FiscalYear	FiscalWeekOfYear	Customer	Sales
2017	45	\$10,968.19	\$10,968.19
2017	46	\$10,888.01	\$10,888.01
2017	47	\$17,999.89	\$16,966.00
2017	48	\$19,888.23	\$19,888.23
2017	49	\$32,584.10	\$32,584.10
2017	50	\$34,237.33	\$32,964.10
2017	51	\$34,143.15	\$34,237.33
2017	52	\$27,611.35	\$34,143.15
2018	1	\$29,636.13	\$27,611.35
2018	2	\$31,372.64	\$31,372.64
2018	3	\$33,398.39	\$31,272.64
2018	4	\$30,623.33	\$33,398.39
2018	5	\$31,951.47	\$30,623.33
2018	6	\$33,149.27	\$31,951.47
2018	7	\$34,946.40	\$33,149.27
2018	8	\$30,482.98	\$34,946.40
2018	9	\$31,652.43	\$30,482.98
2018	10	\$34,128.37	\$31,652.43
2018	11	\$37,604.14	\$34,128.37
2018	12	\$33,803.52	\$31,604.14
2018	13	\$33,711.72	\$33,803.52
2018	14	\$39,023.98	\$33,711.72
2018	15	\$42,011.47	\$39,023.98
2018	16	\$41,317.48	\$42,011.47
2018	17	\$36,098.83	\$41,317.48
2018	18	\$28,145.87	\$36,098.83
2018	19	\$42,327.98	\$28,145.87
2018	20	\$42,429.39	\$42,327.98
2018	21	\$39,962.90	\$42,429.39
Total		\$4,252,704.88	\$4,168,743.44

FiscalYear QTD Sales (4-5-4)

FiscalYear	QTD Sales (4-5-4)
2017	\$71,135.89
2017	\$104,820.91
2017	\$31,168.48
2017	\$145,291.74
2017	\$33,212.58
2017	\$477,341.07
2017	\$145,185.82
2017	\$112,166.51
2017	\$468,856.89
2017	\$148,190.29
2017	\$33,254.82
2017	\$461,427.79
2017	\$112,166.51
2018	\$123,268.99
2018	\$207,041.84
2018	\$403,268.59
2018	\$138,626.74
2018	\$254,519.86
2018	\$358,683.47
2018	\$146,099.71
2018	\$322,470.64
2018	\$403,513.51
2018	\$172,379.06
2018	\$364,244.13
2018	\$408,950.26
2019	\$129,473.00
2019	\$213,334.51
2019	\$502,653.51
Total	

Filters

Visualizations

Fields

5) Apply these filters, will fix the blank value appearing in the matrix above.

Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period +

Page 11 of 11

6) Go to the data view select the FiscalMonthName column & sort by FiscalMonthNumber. Perform this task if the months are in mixed order in the matrix.

The screenshot shows the Power BI Desktop interface with the 'Column tools' ribbon tab selected. In the main area, a table view displays columns: Date, FiscalYear, FiscalQuarter, FiscalMonthNumber, FiscalMonthOfQuarter, FiscalWeekOfYear, and DayName. A context menu is open over the 'FiscalMonthName' column, listing various date-related columns like Dayname, DayOfMonthNumber, DayOfWeek, FiscalMonthNumber, FiscalMonthOfQuarter, FiscalMonthYear, FiscalQuarter, FiscalQuarterYear, FiscalWeekOfYear, and FiscalYear. To the right, the 'Fields' pane lists numerous demo tables, with a blue arrow pointing from the 'FiscalMonthName' entry to the 'FiscalMonthNumber' entry. The status bar at the bottom indicates 'Table: 4-5-4 Calendar (1,092 rows) Column: FiscalMonthName (12 distinct values)'.

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure Quick measure Publish

Queries Data Calculations Share

FiscalYear FiscalWeekOffer Customer Sales Last Week's Sales 4-3-4 (QTD Sales)

FiscalYear	FiscalWeekOffer	Customer Sales	Last Week's Sales 4-3-4	QTD Sales 4-3-4
2017	45	\$29,968.19	\$26,898.19	\$1,779,514.19
2017	46	\$29,828.08	\$26,898.19	\$78,130.99
2017	47	\$27,909.89	\$26,898.19	\$116,806.82
2017	48	\$29,866.23	\$26,898.19	\$117,250.37
2017	49	\$32,504.10	\$26,898.19	\$145,291.74
2018	50	\$34,237.33	\$26,898.19	\$145,041.07
2017	51	\$34,143.15	\$26,898.19	\$145,898.42
2017	52	\$27,671.25	\$26,898.19	\$171,632.89
2018	1	\$29,836.13	\$27,471.35	\$166,268.34
2018	2	\$31,172.64	\$29,656.19	\$166,350.23
2018	3	\$33,358.99	\$31,273.64	\$164,894.32
2018	4	\$33,823.53	\$33,358.99	\$125,068.89
2018	5	\$31,851.47	\$33,823.53	\$121,882.86
2018	6	\$33,149.27	\$31,851.47	\$143,247.75
2018	7	\$34,946.40	\$33,149.27	\$158,659.74
2018	8	\$30,452.98	\$34,946.40	\$177,910.12
2018	9	\$31,652.43	\$30,452.98	\$170,163.85
2018	10	\$34,128.37	\$31,652.43	\$146,098.75
2018	11	\$37,804.14	\$34,128.37	\$178,170.29
2018	12	\$35,823.52	\$37,804.14	\$165,652.87
2018	13	\$33,711.72	\$35,823.52	\$172,879.09
2018	14	\$39,023.98	\$33,711.72	\$171,175.11
2018	15	\$42,011.47	\$39,023.98	\$164,044.15
2018	16	\$41,517.48	\$42,011.47	\$166,080.26
2018	17	\$36,059.63	\$41,517.48	\$129,673.00
2018	18	\$29,145.87	\$36,059.63	\$123,651.31
2018	19	\$42,827.98	\$29,145.87	\$129,318.00
2018	20	\$42,420.39	\$42,827.98	\$32,651.31
2018	21	\$39,962.90	\$42,420.39	Total \$4,160,743.44
Total		\$4,252,704.88	\$4,160,743.44	

Filters

Visualizations

Fields

7) Add customer sales field to the values. The matrix now displays running totals for each quarters.

Fiscal Previous Period

Automatic Date
Tables

Calendar
Functions

Date Formatting

Time Intelligence
Functions

Week-Based
Calculations

Instead of a standard DAX time period shifting functions (i.e. PREVIOUSYEAR, PARALLELPERIOD), you can use CALCULATE, FILTER, ALL and SELECTEDVALUE:

Here we're calculating QTD sales for the previous quarter based on a 4-5-4 fiscal calendar

```
1 Last QTD Sales (4-5-4) =  
2 VAR LastPeriod =  
3 CALCULATE(  
4     [Customer Sales],  
5     FILTER(  
6         ALL(  
7             '4-5-4 Calendar'  
8         ),  
9         IF(  
10             SELECTEDVALUE('4-5-4 Calendar'[FiscalQuarter]) = 1,  
11             '4-5-4 Calendar'[FiscalQuarter] = 4 &&  
12             '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) -1,  
13             '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) &&  
14             '4-5-4 Calendar'[FiscalQuarter] = SELECTEDVALUE('4-5-4 Calendar'[FiscalQuarter]) -1  
15         )  
16     )  
17 )  
18 RETURN  
19 LastPeriod
```

You can replace FiscalQuarter with the fiscal month or fiscal week for MTD and WTD calculations

Just make sure to update the fiscal period to either 4, 12 or 52!

Maven Reactors - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Name Last QTD Sales(4-5-4) Format Currency Data category Uncategorized New Quick measure measure

Home table Measure Table \$ - % 9.0 2 Structure Formatting Properties Calculations

1 Last QTD Sales (4-5-4) =
2 VAR LastPeriod =
3 CALCULATE(
4 [Customer Sales],
5 FILTER(
6 ALL(
7 '4-5-4 Calendar'
8),
9 IF(
10 SELECTEDVALUE('4-5-4 Calendar'[FiscalQuarter]) = 1,
11 '4-5-4 Calendar'[FiscalQuarter] +4 && '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) -1,
12 '4-5-4 Calendar'[FiscalYear] = SELECTEDVALUE('4-5-4 Calendar'[FiscalYear]) +8
13 '4-5-4 Calendar'[FiscalQuarter] = SELECTEDVALUE('4-5-4 Calendar'[FiscalQuarter]) +1
14)
15)
16)
17 RETURN
18 LastPeriod

	Y	3Q(0728)	3Q(0828)	July	August	September	October	November	December	January	2019	February	March	April	Total	
2018	10	\$36,128.37	\$31,691.61		\$150,163.61	\$306,683.47	\$146,699.75	\$146,699.75		\$191,174.09	\$364,044.13				\$4,160,743.44	
2018	11	\$37,604.14	\$36,128.37		\$178,170.89	\$322,870.64				\$160,652.87	\$469,523.31					
2018	12	\$35,801.53	\$37,604.14			\$172,870.06	\$172,870.06			\$194,036.11	\$498,080.26					
2018	13	\$33,711.72	\$35,801.53				\$160,652.87	\$469,523.31								
2018	14	\$39,020.96	\$33,711.72					\$172,870.06	\$172,870.06							
2018	15	\$42,011.67	\$39,020.96						\$191,174.09	\$364,044.13						
2018	16	\$41,517.48	\$42,011.67							\$134,036.11	\$498,080.26					
2018	17	\$38,059.63	\$41,517.48								\$302,832.51					
2018	18	\$38,145.87	\$38,059.63									\$129,873.00				
2018	19	\$42,327.90	\$38,145.87										\$153,661.31	\$113,334.51		
2018	20	\$42,420.39	\$42,327.90											\$169,318.00	\$502,652.51	
2018	21	\$39,982.90	\$42,420.39													
	Total	\$4,252,704.88	\$4,160,743.44													

Visualizations Fields

Search: last qtd

Measure Table Advanced Time Intelligence Last QTD Sales(4-5-4)

FiscalYear Hierarchy FiscalYear FiscalMonthName Date

Rows Columns Add data fields here

Values Customer Sales QTD Sales (4-5-4)

Drill through Cross-report Off Keep all filters On Add drill-through fields here

Relationship functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous fiscal Period +

Sign in

2) Change the formatting accordingly over here.

1) Create a new measure & call it Last QTD Sales (4-5-4).

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Queries Data Calculations Share

FiscalYear FiscalWeekOfficer Customer Sales Last Week's Sales 4-5-4 (DATADOC)

FiscalYear	FiscalWeekOfficer	Customer Sales	Last Week's Sales 4-5-4 (DATADOC)
2017	45	\$39,968.19	\$38,893.13
2017	46	\$39,836.88	\$39,968.19
2017	47	\$37,939.89	\$39,836.88
2017	48	\$29,866.23	\$37,939.89
2017	49	\$32,584.10	\$29,866.23
2017	50	\$34,237.58	\$32,584.10
2017	51	\$34,140.15	\$34,237.58
2017	52	\$27,611.29	\$34,140.15
2018	1	\$29,636.13	\$27,611.29
2018	2	\$31,272.64	\$29,636.13
2018	3	\$33,356.59	\$31,272.64
2018	4	\$30,821.53	\$33,356.59
2018	5	\$31,951.67	\$30,821.53
2018	6	\$33,149.27	\$31,951.67
2018	7	\$34,946.40	\$33,149.27
2018	8	\$30,652.88	\$34,946.40
2018	9	\$31,652.63	\$30,652.88
2018	10	\$36,128.37	\$31,652.63
2018	11	\$37,604.14	\$36,128.37
2018	12	\$33,803.52	\$37,604.14
2018	13	\$33,711.72	\$33,803.52
2018	14	\$39,020.96	\$33,711.72
2018	15	\$42,011.67	\$39,020.96
2018	16	\$41,517.48	\$42,011.67
2018	17	\$36,059.63	\$41,517.48
2018	18	\$38,145.87	\$36,059.63
2018	19	\$42,327.98	\$38,145.87
2018	20	\$42,420.39	\$42,327.98
2018	21	\$39,982.90	\$42,420.39
Total		\$4,252,704.88	\$4,160,743.44

FiscalYear Customer Sales QTD Sales (4-3-4) Last QTD Sales (4-5-4)

FiscalYear	Customer Sales	QTD Sales (4-3-4)	Last QTD Sales (4-5-4)
2017	\$1,739,514.10	-	-
February	\$78,130.99	\$78,130.99	-
March	\$116,056.92	\$194,917.81	-
April	\$117,239.37	\$212,188.48	-
May	\$145,091.74	\$145,091.74	\$112,168.48
June	\$190,129.64	\$395,212.88	\$312,168.48
July	\$141,828.69	\$477,041.07	\$312,168.48
August	\$140,935.62	\$140,935.62	\$477,041.07
September	\$171,652.89	\$112,168.48	\$477,041.07
October	\$156,248.34	\$46,656.85	\$477,041.07
November	\$168,330.20	\$143,350.20	\$468,856.25
December	\$184,564.32	\$352,854.52	\$468,856.25
January	\$128,573.18	\$481,407.70	\$468,856.25
2018	\$1,918,576.80	-	-
February	\$123,048.07	\$121,048.07	\$481,407.70
March	\$161,952.13	\$287,041.84	\$481,407.70
April	\$143,247.75	\$430,289.39	\$481,407.70
May	\$159,029.74	\$158,609.74	\$480,289.39
June	\$171,910.12	\$156,319.86	\$480,289.39
July	\$150,163.61	\$596,683.47	\$480,289.39
August	\$148,699.75	\$148,699.75	\$596,683.47
September	\$176,170.89	\$322,870.84	\$596,683.47
October	\$160,652.87	\$483,323.51	\$596,683.47
November	\$172,870.06	\$172,870.06	\$483,323.51
December	\$191,174.09	\$364,044.15	\$483,323.51
January	\$134,036.11	\$496,000.26	\$483,323.51
2019	\$502,652.31	-	-
February	\$129,673.00	\$129,673.00	\$488,000.26
March	\$183,641.91	\$211,334.51	\$488,000.26
April	\$189,318.00	\$302,652.31	\$488,000.26
May	-	-	\$302,652.31
June	-	-	\$302,652.31
Total	\$4,160,743.44	-	-

Visualizations Fields

last qtd

Measure Table Advanced Time Intelligence Last QTD Sales (4-5-4)

Filters Row

FiscalYear Hierarchy

FiscalYear

FiscalMonthName

Date

Columns Add data fields here

Values Customer Sales

QTD Sales (4-3-4)

Last QTD Sales (4-5-4)

Drill through

Cross-report Off

Keep all filters On

Add drill-through fields here

3) Add the Last QTD Sales (4-5-4) field to the values so available in the matrix.

ASSIGNMENT: 4-5-4 Calendar

KEY OBJECTIVES

- 1) Using the 4-5-4 Calendar & **[Customer Sales]** create measures for YTD & MTD sales.
- 2) Create a measure to compute week-over-week % change.
- 3) Create a matrix (or two) to visualize your results.

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: MTD Sales (4-5-4) %: Data category: Unassigned New measure Quick measure measure

Home table: Measure Table \$ - % ✓ Properties Calculations

Measure

```

1 MTD Sales (4-5-4) =
2 VAR MaxDate = MAX('4-5-4 Calendar'[Date])
3 VAR MaxPeriod = MAX('4-5-4 Calendar'[FiscalMonthYear])
4 VAR OutPut =
5 IF(
6     HASONEVALUE(
7         '4-5-4 Calendar'[FiscalMonthName]
8     ),
9     CALCULATE(
10        [Customer Sales],
11        '4-5-4 Calendar'[Date] <= MaxDate,
12        '4-5-4 Calendar'[FiscalMonthYear] = MaxPeriod
13    ),
14    ""
15 )
16 RETURN
17 OutPut

```

Visualizations > Fields

P MTD

MeasureTable

Advanced Time Intelligence

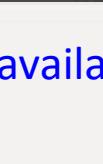
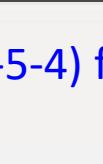
MTD Sales (4-5-4)

1) Objective 1 to use the 4-5-4 Calendar & [Customer Sales] create measures for YTD & MTD sales. Create the MTD Sales (4-5-4) measure.

	14	\$41,517.44	\$42,011.67	2019	\$502,652.51		
	15	\$41,517.44	\$42,011.67	February	\$129,673.00	\$129,673.00	\$498,080.26
	16	\$16,059.63	\$41,517.44	March	\$183,651.31	\$313,334.51	\$498,080.26
	17	\$79,145.87	\$36,009.63	April	\$189,318.00	\$302,652.51	\$498,080.26
	18	\$42,327.98	\$38,145.87	May			\$502,652.51
	19	\$42,420.39	\$42,327.98	June			\$502,652.51
	20	\$39,962.90	\$42,420.39	Total	\$4,160,743.44		
Total		\$4,252,704.88	\$4,160,743.44				

Join Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time I... Previous Fiscal Period

Page 11 of 11



FiscalYear	FiscalWeekOfYear	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)
2017	45	\$39,966.19	\$36,893.13
2017	46	\$39,836.88	\$39,966.19
2017	47	\$37,939.89	\$39,836.88
2017	48	\$29,866.23	\$37,939.89
2017	49	\$32,584.10	\$29,866.23
2017	50	\$34,237.58	\$32,584.10
2017	51	\$34,140.15	\$34,237.58
2017	52	\$27,611.35	\$34,140.15
2018	1	\$29,636.13	\$27,611.35
2018	2	\$31,272.64	\$29,636.13
2018	3	\$33,356.59	\$31,272.64
2018	4	\$30,823.53	\$33,356.59
2018	5	\$31,551.67	\$30,823.53
2018	6	\$33,149.27	\$31,551.67
2018	7	\$34,946.40	\$33,149.27
2018	8	\$30,652.98	\$34,946.40
2018	9	\$31,652.63	\$30,652.98
2018	10	\$36,128.37	\$31,652.63
2018	11	\$37,604.14	\$36,128.37
2018	12	\$35,803.52	\$37,604.14
2018	13	\$33,711.72	\$35,803.52
2018	14	\$39,020.96	\$33,711.72
2018	15	\$42,011.67	\$39,020.96
2018	16	\$41,517.48	\$42,011.67
2018	17	\$36,059.63	\$41,517.48
2018	18	\$39,145.87	\$36,059.63
2018	19	\$42,327.98	\$39,145.87
2018	20	\$42,420.39	\$42,327.98
2018	21	\$39,982.90	\$42,420.39
Total		\$4,252,704.88	\$4,160,743.44

FiscalYear	Customer Sales	MTD Sales (4-5-4)	QTD Sales (4-5-4)	Last QTD Sales (4-5-4)
2017	\$1,739,514.10	-	-	-
February	\$70,130.99	70,130.99	\$70,130.99	-
March	\$116,806.92	116,806.92	\$194,917.91	-
April	\$117,250.57	117,250.57	\$312,188.48	-
May	\$145,091.74	145,091.74	\$145,091.74	\$312,188.48
June	\$190,120.64	190,120.64	\$335,212.38	\$312,188.48
July	\$141,828.69	141,828.69	\$477,041.07	\$312,188.48
August	\$140,935.62	140,935.62	\$140,935.62	\$477,041.07
September	\$171,652.89	171,652.89	\$312,588.51	\$477,041.07
October	\$156,268.34	156,268.34	\$468,856.85	\$477,041.07
November	\$168,350.20	168,350.20	\$168,350.20	\$468,856.85
December	\$184,504.32	184,504.32	\$352,854.52	\$468,856.85
January	\$128,573.18	128,573.18	\$481,427.70	\$468,856.85
2018	\$1,918,576.83	-	-	-
February	\$125,058.89	125,058.89	\$125,058.89	\$481,427.70
March	\$161,952.95	161,952.95	\$287,041.84	\$481,427.70
April	\$143,247.75	143,247.75	\$430,289.59	\$481,427.70
May	\$158,609.74	158,609.74	\$158,609.74	\$430,289.59
June	\$197,910.12	197,910.12	\$356,519.86	\$430,289.59
July	\$150,163.61	150,163.61	\$506,683.47	\$430,289.59
August	\$146,699.75	146,699.75	\$146,699.75	\$506,683.47
September	\$176,170.89	176,170.89	\$322,870.64	\$506,683.47
October	\$160,652.87	160,652.87	\$483,523.51	\$506,683.47
November	\$172,870.06	172,870.06	\$172,870.06	\$483,523.51
December	\$191,174.09	191,174.09	\$364,044.15	\$483,523.51
January	\$134,036.11	134,036.11	\$498,080.26	\$483,523.51
2019	\$502,652.51	-	-	-
February	\$129,673.00	129,673.00	\$129,673.00	\$498,080.26
March	\$183,661.51	183,661.51	\$313,334.51	\$498,080.26
April	\$189,318.00	189,318.00	\$502,652.51	\$498,080.26
May	-	-	-	\$502,652.51
June	-	-	-	\$502,652.51
Total	\$4,160,743.44	-	-	-

2) Add the MTD Sales (4-5-4) field to the values so available in the matrix below.

3) The Customer Sales & MTD Sales (4-5-4) values so match each other.

Visualizations

Fields

MTD

Measure Table

Advanced Time Intelligence

MTD Sales (4-5-4)

Row

Customer Hierarchy

FiscalYear

FiscalMonthName

Date

Column

Add data fields here

Values

Customer Sales

MTD Sales (4-5-4)

QTD Sales (4-5-4)

Last QTD Sales (4-5-4)

Drill through

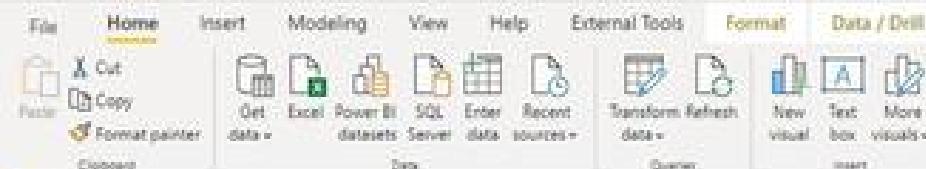
Cross-report

Off

Keep all filters

On

Add drill-through fields here



FiscalYear	FiscalWeekOfYear	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)
2017	45	\$39,968.19	\$36,893.13
2017	46	\$39,836.88	\$39,968.19
2017	47	\$37,939.89	\$39,836.88
2017	48	\$29,866.23	\$37,939.89
2017	49	\$32,584.10	\$29,866.23
2017	50	\$34,237.58	\$32,584.10
2017	51	\$34,140.15	\$34,237.58
2017	52	\$27,611.35	\$34,140.15
2018	1	\$29,636.13	\$27,611.35
2018	2	\$31,272.64	\$29,636.13
2018	3	\$33,356.59	\$31,272.64
2018	4	\$30,823.53	\$33,356.59
2018	5	\$31,551.67	\$30,823.53
2018	6	\$33,149.27	\$31,551.67
2018	7	\$34,946.40	\$33,149.27
2018	8	\$30,652.98	\$34,946.40
2018	9	\$31,652.63	\$30,652.98
2018	10	\$36,128.37	\$31,652.63
2018	11	\$37,604.14	\$36,128.37
2018	12	\$35,803.52	\$37,604.14
2018	13	\$33,711.72	\$35,803.52
2018	14	\$39,020.96	\$33,711.72
2018	15	\$42,011.67	\$39,020.96
2018	16	\$41,517.48	\$42,011.67
2018	17	\$36,059.63	\$41,517.48
2018	18	\$39,145.87	\$36,059.63
2018	19	\$42,327.98	\$39,145.87
2018	20	\$42,420.39	\$42,327.98
2018	21	\$39,982.90	\$42,420.39
Total		\$4,252,704.88	\$4,160,743.44

4) When drill in into the month the MTD sales (4-5-4) values are broken down into days. Days by day MTD sales are shown until we finally get to the total.

FiscalYear		Customer Sales	MTD Sales (4-5-4)	QTD Sales (4-5-4)	Last QTD Sales (4-5-4)
2017		\$1,739,514.10			
	February	\$78,130.99	78,130.99	\$78,130.99	
	05 February 2017	\$2,304.70	2,304.70	\$2,304.70	
	06 February 2017	\$2,203.40	4,508.10	\$4,508.10	
	07 February 2017	\$2,563.35	7,071.45	\$7,071.45	
	08 February 2017	\$2,762.43	9,833.88	\$9,833.88	
	09 February 2017	\$2,610.63	12,444.51	\$12,444.51	
	10 February 2017	\$2,901.60	15,346.11	\$15,346.11	
	11 February 2017	\$2,526.74	17,872.85	\$17,872.85	
	12 February 2017	\$2,894.00	20,766.85	\$20,766.85	
	13 February 2017	\$2,845.48	23,612.33	\$23,612.33	
	14 February 2017	\$2,673.93	26,286.26	\$26,286.26	
	15 February 2017	\$2,928.05	29,214.31	\$29,214.31	
	16 February 2017	\$3,023.33	32,237.64	\$32,237.64	
	17 February 2017	\$2,300.75	34,538.39	\$34,538.39	
	18 February 2017	\$2,865.48	37,403.87	\$37,403.87	
	19 February 2017	\$3,219.60	40,623.47	\$40,623.47	
	20 February 2017	\$2,883.63	43,507.10	\$43,507.10	
	21 February 2017	\$2,783.53	46,290.63	\$46,290.63	
	22 February 2017	\$2,928.70	49,219.33	\$49,219.33	
	23 February 2017	\$2,746.21	51,963.54	\$51,963.54	
	24 February 2017	\$2,940.70	54,906.24	\$54,906.24	
	25 February 2017	\$2,823.55	57,729.79	\$57,729.79	
	26 February 2017	\$2,956.75	60,686.54	\$60,686.54	
	27 February 2017	\$3,160.00	63,846.54	\$63,846.54	
	28 February 2017	\$2,311.10	66,157.64	\$66,157.64	
	01 March 2017	\$3,040.25	69,197.89	\$69,197.89	
	02 March 2017	\$2,996.05	72,193.94	\$72,193.94	
	03 March 2017	\$3,155.15	75,349.09	\$75,349.09	
	04 March 2017	\$2,781.90	78,130.99	\$78,130.99	
	05 March 2017	\$2,906.92	116,806.92	\$116,806.92	
	06 March 2017	\$117,250.57	117,250.57	\$117,250.57	
	Total	\$4,160,743.44	-	-	-

Visualizations

Fields

MTD

Measure Table

Advanced Time Intelligence

MTD Sales (4-5-4)

Row

FiscalYear Hierarchy

FiscalYear

FiscalMonthName

Date

Column

Add data fields here

Values

Customer Sales

MTD Sales (4-5-4)

QTD Sales (4-5-4)

Last QTD Sales (4-5-4)

Drill through

Cross-report

Off

Keep all filters

On

Add drill-through fields here

Maven Roosters - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Table tools Measure tools

Name: YTD Sales (4-5-4) Format: General Data category: Uncategorized

Home table: Measure Table \$ - % # . Auto New Quick measure measure

Structure Formatting Properties Calculations

6) Change the formatting accordingly over here.

5) First objective part 2 is to use the 4-5-4 Calendar & [Customer Sales] create measures for YTD sales. Create the YTD Sales (4-5-4) measure.

```

1 YTD Sales (4-5-4) =
2 VAR MaxDate = MAX('4-5-4 Calendar'[Date])
3 VAR MaxPeriod = MAX('4-5-4 Calendar'[FiscalYear])
4 VAR MaxSellDate = MAX('Sales by Store'[transaction_date])
5 VAR OutPut =
6 CALCULATE(
7     [Customer Sales],
8     '4-5-4 Calendar'[Date] <= MaxDate,
9     '4-5-4 Calendar'[FiscalYear] = MaxPeriod,
10    'Calendar'[Transaction Date] <= MaxSellDate
11 )
12 RETURN
13 OutPut
  
```

Date	Value	Value	Value
21 February 2017	\$2,783.53	45,290.63	\$46,290.63
22 February 2017	\$2,928.70	49,219.33	\$49,219.33
23 February 2017	\$2,746.21	51,965.54	\$51,965.54
24 February 2017	\$2,940.70	54,906.24	\$54,906.24
25 February 2017	\$2,823.55	57,729.79	\$57,729.79
26 February 2017	\$2,956.75	60,686.54	\$60,686.54
27 February 2017	\$3,160.00	63,846.54	\$63,846.54
28 February 2017	\$2,311.10	66,157.64	\$66,157.64
01 March 2017	\$3,040.25	69,197.89	\$69,197.89
02 March 2017	\$2,996.05	72,193.94	\$72,193.94
03 March 2017	\$3,155.15	75,149.09	\$75,149.09
04 March 2017	\$2,781.90	78,130.99	\$78,130.99
March	\$116,806.92	116,806.92	\$194,937.91
April	\$117,260.57	117,260.57	\$312,188.48
Total	\$4,160,743.44	-	-
Total	\$4,252,704.88	\$4,160,743.44	-

Visualizations Fields

YTD Sales (4-5-4)

Measure Table Advanced Time Intelligence YTD Sales (4-5-4)

Add data fields here Drill through Cross-report Off On Keep all filters Off Add drill-through fields here

Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time Intelligence Previous Fiscal Period

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill Data Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Publish

7) Add the YTD Sales (4-5-4) field to the values so available in the matrix below.

FiscalYear	FiscalWeekOfYear	Customer Sales	Last Week's Sales 4-5-4 (DATEADD)	Metric
2017	1	\$91,961.44	\$17,872.85	Customer Sales
2017	2	\$19,531.02	\$17,872.85	MTD Sales (4-5-4)
2017	3	\$20,325.92	\$19,531.02	YTD Sales (4-5-4)
2017	4	\$20,401.20	\$20,325.92	QTD Sales (4-5-4)
2017	5	\$22,281.41	\$20,401.20	Next QTD Sales (4-5-4)
2017	6	\$23,342.89	\$22,281.41	
2017	7	\$23,424.50	\$23,342.89	
2017	8	\$21,832.18	\$23,424.50	
2017	9	\$25,925.94	\$21,832.18	
2017	10	\$28,568.79	\$25,925.94	
2017	11	\$29,396.61	\$28,568.79	
2017	12	\$28,155.59	\$29,396.61	
2017	13	\$31,128.58	\$28,155.59	
2017	14	\$35,998.36	\$31,128.58	
2017	15	\$38,158.39	\$35,998.36	
2017	16	\$37,637.28	\$38,158.39	
2017	17	\$33,297.71	\$37,637.28	
2017	18	\$37,422.55	\$33,297.71	
2017	19	\$40,708.46	\$37,422.55	
2017	20	\$41,036.10	\$40,708.46	
2017	21	\$37,015.22	\$41,036.10	
2017	22	\$33,938.31	\$37,015.22	
2017	23	\$36,661.38	\$33,938.31	
2017	24	\$38,204.03	\$36,661.38	
2017	25	\$35,323.99	\$38,204.03	
2017	26	\$31,639.29	\$35,323.99	
2017	27	\$34,044.07	\$31,639.29	
2017	28	\$37,794.35	\$34,044.07	
Total		\$4,252,704.88	\$4,160,743.44	

Key objective 1 is now complete. Use the 4-5-4 Calendar & [Customer Sales] create measures for YTD & MTD sales.

Visualizations Fields

YTD Sales (4-5-4)

Measure Table Advanced Time Intelligence YTD Sales (4-5-4)

Rows

FiscalYear Hierarchy FiscalYear FiscalMonthName Date

Columns

Add data fields here

Values

Customer Sales MTD Sales (4-5-4) YTD Sales (4-5-4) QTD Sales (4-5-4) Last QTD Sales (4-5-4)

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Maven Readers - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill Table tools Measure tools

Measure Name: WoW % Change (4-5-4) Format: Percentage

Home table: Measure Table \$ - % , . :

Structure, Styling, Properties, Controls

Visualizations Fields

P WOW

Measure Table

Advanced Time Intelligence

WoW % Change (4-5-4)

Row

FiscalYear Hierarchy

FiscalYear

FiscalMonthName

Date

Columns

Add data fields here

Values

Customer Sales

MTD Sales (4-5-4)

YTD Sales (4-5-4)

QTD Sales (4-5-4)

Last QTD Sales (4-5-4)

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

8) Objective 2 to create a measure to compute week-over-week % change. Create the WoW % Change (4-5-4) measure.

1 WoW % Change (4-5-4) =
 2 DIVIDE(
 3 ([Customer Sales] - [Last Week's Sales 4-5-4 (DATEADD)]),
 4 [Last Week's Sales 4-5-4 (DATEADD)]),
 5 BLANK()
 6)

2017	8	\$21,832.18	\$23,424.50	November	\$172270.06	172,870.06	1,391,366.63	\$172,870.06	\$403,323.51
2017	9	\$25,925.94	\$21,832.18	October	\$160,652.87	160,652.87	1,420,496.57	\$483,523.51	\$506,683.47
2017	10	\$28,569.79	\$25,925.94	September	\$175,170.89	175,170.89	1,259,843.70	\$322,870.64	\$506,683.47
2017	11	\$29,396.61	\$28,569.79	August	\$146,699.75	146,699.75	1,083,672.81	\$146,699.75	\$506,683.47
2017	12	\$28,155.99	\$29,396.61	July	\$150,163.61	150,163.61	936,973.06	\$506,683.47	\$430,289.59
2017	13	\$31,128.58	\$28,155.99	June	\$197,910.12	197,910.12	786,809.45	\$356,519.86	\$430,289.59
2017	14	\$35,998.36	\$31,128.58	May	\$158,609.74	158,609.74	588,899.13	\$158,609.74	\$430,289.59
2017	15	\$38,158.39	\$35,998.36	April	\$143,247.75	143,247.75	430,289.59	\$430,289.59	\$481,427.70
2017	16	\$37,637.28	\$38,158.39	March	\$161,952.95	161,952.95	287,041.84	\$287,041.84	\$481,427.70
2017	17	\$33,297.71	\$37,637.28	February	\$125,088.89	125,088.89	125,088.89	\$125,088.89	\$481,427.70
2017	18	\$37,422.55	\$33,297.71	2017	\$1,739,514.10	-	1,739,514.10	-	-
2017	19	\$40,708.46	\$37,422.55	January	\$128,573.18	128,573.18	1,739,514.10	\$481,427.70	\$468,856.85
2017	20	\$41,036.10	\$40,708.46	December	\$104,504.32	104,504.32	1,610,940.52	\$352,854.52	\$468,856.85
2017	21	\$37,015.22	\$41,036.10	November	\$168,350.20	168,350.20	1,426,436.60	\$168,350.20	\$468,856.85
2017	22	\$33,938.31	\$37,015.22	October	\$156,268.34	156,268.34	1,258,086.40	\$468,856.85	\$477,041.07
2017	23	\$36,661.38	\$33,938.31	September	\$171,652.89	171,652.89	1,101,816.06	\$312,588.51	\$477,041.07
2017	24	\$38,204.03	\$36,661.38	August	\$140,935.62	140,935.62	930,166.17	\$140,935.62	\$477,041.07
2017	25	\$35,323.99	\$38,204.03	July	\$141,828.69	141,828.69	789,229.55	\$477,041.07	\$312,188.48
2017	26	\$31,639.29	\$35,323.99	June	\$190,120.64	190,120.64	647,400.86	\$335,212.38	\$312,188.48
2017	27	\$34,044.07	\$31,639.29	May	\$145,091.74	145,091.74	457,280.22	\$145,091.74	\$312,188.48
2017	28	\$37,794.35	\$34,044.07	April	\$117,250.57	117,250.57	312,188.48	\$312,188.48	
				March	\$116,806.92	116,806.92	194,937.91	\$194,937.91	
				Total	\$4,160,743.44	-	\$62,652.51	-	
					\$4,252,704.88	\$4,160,743.44			

Key objective 2 is now complete. Create a measure to compute week-over-week % change.

Table & Filter Functions, Joins, Relationship Functions, Iterators, Advanced Time Intelligence, Duplicate of Advanced Time Intelligence, Duplicate of Duplicate of Advanced Time I... Previous Next

Maven Roadster - Analysis - Advanced Time Intelligence - Power BI Desktop

Sign in

File Home Insert Modeling View Help External Tools Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure New calculation Publish

Data Query Insert Calculations Share

FiscalYear FiscalWeekOfYear Customer Sales Last Week's Sales 4-5-4 (DATEADD) WoW % Change (4-5-4)

FiscalYear Customer Sales MTD Sales (4-5-4) YTD Sales (4-5-4) QTD Sales (4-5-4) Last QTD Sales (4-5-4)

2019 \$502,652.51 502,652.51 \$502,652.51

July \$502,652.51

June \$502,652.51

May \$502,652.51

April \$502,652.51

March \$502,652.51

February \$502,652.51

2018 \$1,918,576.83 1,918,576.83 1,918,576.83

January \$1,918,576.83

December \$1,918,576.83

November \$1,918,576.83

October \$1,918,576.83

September \$1,918,576.83

August \$1,918,576.83

July \$1,918,576.83

June \$1,918,576.83

May \$1,918,576.83

April \$1,918,576.83

March \$1,918,576.83

February \$1,918,576.83

2017 \$1,739,514.10 1,739,514.10 1,739,514.10

January \$1,739,514.10

December \$1,739,514.10

November \$1,739,514.10

October \$1,739,514.10

September \$1,739,514.10

August \$1,739,514.10

July \$1,739,514.10

June \$1,739,514.10

May \$1,739,514.10

April \$1,739,514.10

March \$1,739,514.10

February \$1,739,514.10

Total \$4,252,704.88 \$4,160,743.44 2.21%

\$4,160,743.44 502,652.51

Visualizations Fields

WOW

Measure Table Advanced Time Intelligence WoW % Change (4-5-4)

Values

FiscalYear FiscalWeekOfYear Customer Sales Last Week's Sales 4-5-4 (DATEADD) WoW % Change (4-5-4)

Drill through

Cross-report

Off On

Keep all filters

On

Add drill-through fields here

Key objective 3 is now complete. Create a matrix (or two) to visualize your results.

Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Duplicate of Advanced Time Intelligence Duplicate of Duplicate of Advanced Time Intelligence Previous Fiscal Period

Page 11 of 11

11) Performance Tuning

Introduction - Performance Tuning



Tools like **DAX Studio** and Power BI's **Performance Analyzer** can help you troubleshoot issues, measure load times for visuals/DAX queries, and optimize your code

TOPICS WE'LL COVER:

Performance
Analyzer

Copy Query

DAX Studio

Optimization
Workflow

COMMON USE CASES:

- *Identify issues with slow-loading visuals or queries*
- *Test and compare measures to determine the impact on speed and performance*
- *Optimize measures or report configurations to minimize processing and load times*

Power BI Performance Analyzer

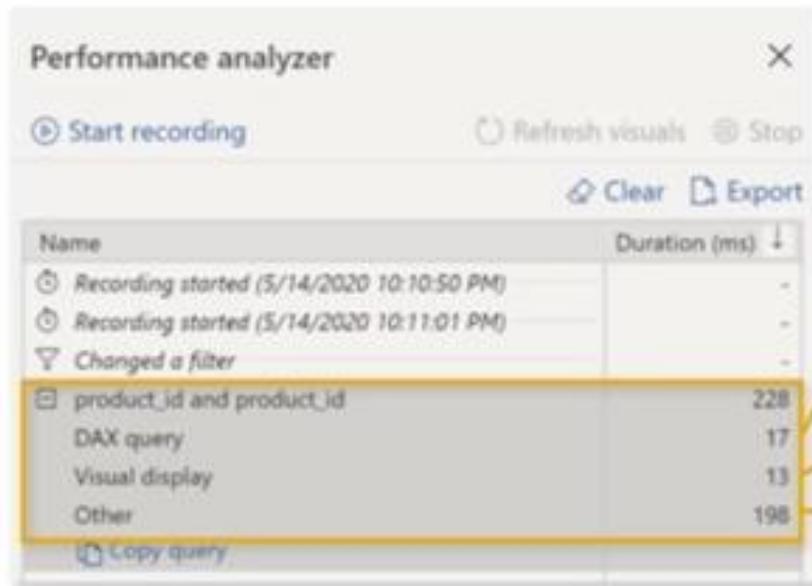
Power BI Desktop's **Performance Analyzer** records user actions (*like Excel's macro recorder*), and tracks the load time (*in milliseconds*) for each step in the process:

Performance Analyzer

Copy Query

DAX Studio

Optimization Workflow



DAX Query

- Shows the amount of time it takes for the visual to send the query to the engines, and for the engines to return the result (**Note:** DAX Studio can only help optimize this)

Visual Display

- Shows the amount of time it takes for the visual to populate, or "draw", on the screen. Includes time to retrieve web-based and geocoded images

Other

- Shows the amount of time required by the visual to prepare the query, wait for other visuals to complete their queries and perform other processing tasks

File Home Insert Modeling View Help

Themes

Page view Mobile layout Page options

Filters Bookmarks Selection Performance analyzer

Assess your report's performance and identify areas for improvement.

Build visuals with your data

Select or drag fields from the Fields pane onto the report canvas.

1) To access the Performance Analyser go to the View menu under the show panes section the Performance Analyser will be available click on to it.

The screenshot shows the Microsoft Power BI Desktop interface. At the top, there's a ribbon with tabs: File, Home, Insert, Modeling, View, and Help. Below the ribbon, there are several icons for themes, page view, mobile layout, and page options. A specific icon for 'Performance analyzer' is highlighted with a yellow arrow. The main workspace is titled 'Build visuals with your data' and contains instructions to 'Select or drag fields from the Fields pane onto the report canvas.' On the right side, there's a 'Visualizations' pane with various chart and table icons, and a 'Fields' pane listing numerous demo datasets like 'Measure Table', '4-5-4 Calendar', and 'ADDCOLUMNS Demo'. The 'Performance Tuning' tab is currently selected at the bottom of the screen.

Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help

Themes Scale to fit Mobile Page view = Layout Options Snap to grid Lock objects Page options Show panel Performance analyzer

Build visuals with your data
Select or drag fields from the Fields pane onto the report canvas.

Performance analyzer

Start recording Refresh values Stop

Start monitoring your report to see details about the time taken by each visual to query for its data and render the result.

Visualizations > Fields >

Search

Measure Table
4-5-4 Calendar
ADDCOLUMNS Demo
Average Days
Calendar
CALENDAR Demo
CALENDARAUTO Demo
CROSSJOIN Demo
Customer Lookup
DATATABLE Demo
Date Table (CALENDARAUTO)
Employee Lookup
EXCEPT Demo
Food Inventory
GENERATESERIES Demo
New Employees (INTERSECT)
Product Lookup
Repeat Customer Sales (INTER...
Row Demo
Sales by Store
SELECTCOLUMNS Demo
Store Lookup
SUMMARIZE Demo
Table Constructor Demo
Target Sales - April 2019 (DAT...
Target Sales - March 2019 (DA...
TRIMZAD Demo
UNION Demo

Values Add data fields here

Drill through

Cross-report

Off —

Keep all filters On —

Add drill-through fields here

2) These options will be made available after clicking on the Performance Analyser button.

Learn more about optimizing your report's performance on our [support site](#). Find help tuning your report from specialist Power BI partners on [AppSource](#).

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced 1 +

Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Closures Data Queries Calculations Share

Out Cut Copy Format painter Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure measure Quick Publish

Transaction Date Customer Sales 1

Transaction Date	Customer Sales 1
01/01/2017	\$2,368
02/01/2017	\$2,469
03/01/2017	\$2,349
04/01/2017	\$2,229
05/01/2017	\$2,479
06/01/2017	\$2,274
07/01/2017	\$2,767
08/01/2017	\$2,639
09/01/2017	\$2,677
10/01/2017	\$2,658
11/01/2017	\$2,548
12/01/2017	\$2,524
Total	\$4,258,795

3) Perform the below tasks.

- Click on Start recording.
- Add matrix to the report.
- Add Transaction Date field to the rows.
- Add Customer Sales 1 to the values.

4) The amount of time it takes the matrix & measures to generate in milliseconds i.e. steps to run.

5) Click on stop once the analysis is complete.

Performance analyzer

Start recording Refresh visuals Stop Clear Export

Name Duration (ms)

Recording started (26/10/2020 20:58:43)

Matrix 5

Other 3

Changed a filter 197

Matrix 10

DAX query 69

Visual display 118

Changed a filter 238

Matrix 6

DAX query 107

Visual display 125

Copy query

Rows

Columns

Values

Drill through

Cross-report

Off —

Keep all filters

On —

Add drill-through fields here

Visualizations Fields

Search

Measure Table

Advanced CALCULATE

Advanced Time Intelligence

Calculated Table Joins

Iterators

Performance Tuning

Customer Sales 1

Customer Sales 2

Customer Sales 3

Relationship Functions

Scalar Measures

Table & Filter Functions

Tips & Best Practices

4-1-4 Calendar

ADDCOLUMNS Demo

Average Days

Calendar

CALENDARAUTO Demo

CROSSJOIN Demo

Customer Lookup

DATATABLE Demo

Data Table (CALENDARAUTO)

Employee Lookup

EXCEPT Demo

Food Inventory

GENERATESERIES Demo

New Employees (INTERSECT)

Product Lookup

Learn more about optimizing your report's performance on our support site. Find help tuning your report from specialist Power BI partners on [AcadSource](#).

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators

PRO TIP: Copy Query

The **Performance Analyzer** includes a **Copy Query** option, which you can use to copy the actual code that DAX generates behind the scenes to produce specific visuals:

The screenshot shows the Power BI Performance analyzer interface. At the top, there are three buttons: 'Start recording', 'Refresh visuals', and 'Stop'. Below this is a table with two columns: 'Name' and 'Duration (ms)'. The table contains the following data:

Name	Duration (ms)
Recording started (6/11/2020...)	-
Slicer	51
DAX query	3
Visual display	6
Other	42
Copy query	-
Refreshed visual	-

A yellow arrow points from the 'Copy query' button in the table to a code editor window on the right. The code editor contains the following DAX query:

```
1 Slicer Query // DAX Query
2 EVALUATE
3 TOPN(
4    101,
5    VALUES('Product Lookup'[product_category]),
6    'Product Lookup'[product_category],
7    1
8 )
9
10 ORDER BY
11 'Product Lookup'[product_category]
12
```

Remember that **VALUES** will include a blank if it exists in the model? Well if you see a blank option in a slicer, this is why!

Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter Get data - Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals Quick measure measure Publish

Dashboard Data Queries Insert Calculations Share

Transaction Date Customer Sales 1

Transaction Date	Customer Sales 1
01/01/2011	\$2,500
02/01/2011	\$2,400
03/01/2011	\$2,300
04/01/2011	\$2,200
05/01/2011	\$2,100
06/01/2011	\$2,000
07/01/2011	\$2,000
08/01/2011	\$2,000
09/01/2011	\$2,000
10/01/2011	\$2,000
11/01/2011	\$2,000
12/01/2011	\$2,000
Total	\$4,750,000

Performance analyzer

(Start recording) Refresh values Stop Clear Export

Name	Duration (ms)
Recording started (26/10/2020 20:58:40)	-
Matrix	3
Other	3
(Copy query)	-
Changed a filter	-
Matrix	107
DAX query	10
Visual display	69
Other	113
(Copy query)	-
Changed a filter	-
Matrix	238
DAX query	6
Visual display	107
Other	123
(Copy query)	-

Visualizations Fields

Rows Transaction Date

Columns Add data fields here

Values Customer Sales 1

Drill through

Cross-report

On Off

Keep all filters

On Off

Add drill-through fields here

Learn more about optimizing your report's performance on our support site. Find help tuning your report from specialist Power BI partners on [AdSourcer](#).

1) Click here to Copy query note that the only recent actions can be copied other actions get greyed out.

Tips & Best Practices Scalar Functions Advanced CALCULATE Table & Filter Functions Joins Relationship Functions Iterators

Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: Measure Format: \$ - % Date category: Uncategorized

Home table: 4-5-4 Calendar

Measure tools: New Quick measure measure

Structure Properties Calculations

// DAX Query

```
1 // DAX Query
2 DEFINE
3     VAR __DSOCore =
4         SUMMARIZECOLUMNS(
5             ROLLUPADDSUBTOTAL('Calendar'[Transaction_Date], "[IsGrandTotalRowTotal]"),
6             "Customer_Sales_1", "Measure Table'[Customer Sales 1]"
7         )
8
9     VAR __DSOPrimaryMindrow =
10        TOPN(500, __DSOCore, [IsGrandTotalRowTotal], 0, 'Calendar'[Transaction Date], 1)
11
12 EVALUATE
13     __DSOPrimaryMindrow
14
15 ORDER BY
16     [IsGrandTotalRowTotal] DESC, 'Calendar'[Transaction Date]
17
```

Performance analyzer

Start recording Return visuals Stop Clear Export Duration (ms): 1

Name	Duration (ms)
Recording started (28/10/2020 21:29:01)	1
Macro	7
Other	7
DAX query	1396
Visual display	11
Changed a filter	152
Other	1231
Copy query	1
Changed a filter	172
Macro	296
DAX query	7
Visual display	77
Other	172
Copied	1

Visualizations Fields

Search

Measure Table Advanced CALCULATE Advanced Time Intelligence Calculated Table Joins Iterators Performance Tuning Customer Sales 1 Customer Sales 2 Customer Sales 3 Relationship Functions Scalar Measures Table & Filter Functions Tips & Best Practices 4-5-4 Calendar

Rows Transaction Date

Columns Add data fields here

Values Customer Sales 1

Drill through

Cross-report Off On Keep all items On Add drill-through fields here

Date DayName DayOfMonthNumber DayOfWeek FiscalMonthName FiscalMonthNumber FiscalMonthOfQuarter FiscalMonthYear FiscalQuarter FiscalQuarterYear FiscalWeekOfYear FiscalYear FiscalYearHierarchy Measure ADDCOLUMNS Demo Average Days

Learn more about optimizing your report's performance on our [support site](#). Find help tuning your report from specialist Power BI partners on [AppSource](#).

2) Paste the DAX code into a new measure you will notice that there are numerous errors picked up because Power BI doesn't understand this type of syntax i.e. double back slash & DEFINE statement that's when the tool like DAX Studio comes into play it does understand exactly what's going on here.

DAX Studio

DAX Studio is a free tool that allows you to connect to your Power BI data model to test and optimize your DAX queries

Performance Analyzer

Copy Query

DAX Studio

Optimization Workflow



www.Daxstudio.org



The screenshot displays the DAX Studio interface. The top navigation bar includes Home, Advanced, Help, Layout, Undo, Copy, Paste, DAX Format Query, To Upper, To Lower, Comment, Uncomment, Swap Definitions, Merge XML, To XML, Replace, Load Perf Data, Power BI, Connect, Server Timings, Refresh Metadata, and Connection. The main area shows a query editor with the following DAX code:

```
// DAX Query
DEFINE VAR __DSQFITterTable =
    FILTER(
        KEEPFILTERS(VALUES('4-5-4_Calendar'[Date])),
        NOT('4-5-4_Calendar'[Date] IN {BLANK()})
    )
EVALUATE
TOPN(
    SUMMARIZECOLUMNS(
        ROLLUPDROPSUMTOTAL(
            '4-5-4_Calendar'[FiscalYear], "EndAndTotalRowTotal"
        ),
        ROLLUPGROUP('4-5-4_Calendar'[FiscalMonthName]), '4-5-4_Calendar'[FiscalMonthNumber]
    ),
    __DSQFITterTable,
    [Customer_Sales].[measures].[Customer_Sales]
)
```

The bottom section is the Performance Analyzer, showing the following execution details:

Total	# CPU	Line	Subtask	Duration	CPU	Rows	#B	Query
150 ms	96 ms	2	Scan	16	1,259	22	WITH Blip0 = (SELECT	
48 ms	48 ms	4	Scan	0	0	2,095	9	SELECT * FROM [4-5-4_Calend
47 ms	47 ms	6	Scan	0	0	1,052	5	SELECT * FROM [4-5-4_Calend
47 ms	47 ms	8	Scan	1	0	35	2	SELECT * FROM [4-5-4_Calend
47 ms	47 ms	10	Scan	1	0	6	1	SELECT * FROM [4-5-4_Calend
47 ms	47 ms	12	Scan	17	0	814	10	WITH Blip0 = (SELECT
47 ms	47 ms	14	Scan	0	0	7	1	SELECT * FROM [4-5-4_Calend
47 ms	47 ms	16	Scan	17	0	814	11	WITH Blip0 = (SELECT
47 ms	47 ms	18	Scan	1	0	15	1	SELECT * FROM [4-5-4_Calend
47 ms	47 ms	20	Scan	17	0	814	12	WITH Blip0 = (SELECT
47 ms	47 ms	22	Scan	0	0	2,095	0	END)

Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help External Tools Format Data / Drill

ALM DAX Tabular Toolkit Studio Editor

External Tools

1) Click on External Tools & click on DAX Studio.

Performance analyzer

Start recording Refresh visuals Stop

Start monitoring your report to see details about the time taken by each visual to query for its data and render the result.

Visualizations

Fields

Search

Table & Filter Functions

Tips & Best Practices

4-5-4 Calendar

- Date
- DayName
- Σ DayOfMonthNumber
- Σ DayOfWeek
- FiscalMonthName
- Σ FiscalMonthNumber
- Σ FiscalMonthOfQuarter
- FiscalMonthYear
- Σ FiscalQuarter
- Σ FiscalQuarterYear
- Σ FiscalWeekOfYear
- Σ FiscalYear

Rows

Transaction_Date

Columns

Add data fields here

Values

Customer Sales 1

Drill through

Measurements

Functions Joins Relationship Functions Iterators +

Learn more about optimizing your report's performance on our [support site](#). Find help tuning your report from specialist Power BI partners on [AppSource](#).



A To Upper
B To Lower
Swap Delimiters

Comment
Uncomment
Merge XML



Query

Edit

Format

Find

Power BI

Traces

Connection

Query1.dax X

Metadata

eb7e3462-92f6-4c05-bd21

Model

- 4-5-4 Calendar
- ADDCOLUMNS Demo
- Average Days
- Calendar
- CALENDAR Demo
- CALENDARAUTO Demo
- CROSSJOIN Demo
- Customer Lookup
- DATATABLE Demo
- Date Table (CALENDAR/
- Employee Lookup
- EXCEPT Demo
- Food Inventory
- GENERATESERIES Democ
- Measure Table
- New Employees (INTER:
- Product Lookup
- Repeat Customer Sales

1

2) DAX Studio application will appear.



100 %

Output

	Start	Duration	Text
1	16:15:58		Establishing Connection

Metadata Functions DMV

Output Results Query History

File Home Advanced Help

Run Cancel Clear Cache Output

Cut Undo DAX Format Query To Upper Comment Find Load Perf Data

Copy Redo Paste To Lower Uncomment Replace All Queries Query Plan Server Timings

Swap Delimiters Merge XML Find Power BI Traces Connection

Query Edit Format

Query1.dax* X

Metadata eb7e3462-92f6-4c05-bd21 Model

4-5-4 Calendar ADDCOLUMNS Demo Average Days Calendar CALENDAR Demo CALENDARAUTO Demo CROSSJOIN Demo Customer Lookup DATATABLE Demo Date Table (CALENDAR) Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo Measure Table New Employees (INTER Product Lookup Repeat Customer Sales

```
// DAX QUERY
DEFINE
    VAR __DSOCore =
        SUMMARIZECOLUMNS(
            ROLLUPADDISSUBTOTAL('Calendar'[Transaction_Date], "[IsGrandTotalRowTotal]"),
            "Customer_Sales_1", "Measure Table'[Customer Sales 1]"
        )

    VAR __DSOPrimarywindowed =
        TOPN(SO2, __DSOCore, [IsGrandTotalRowTotal], 0, 'calendar'[Transaction_Date], 1)

EVALUATE
    __DSOPrimarywindowed
ORDER BY
    [IsGrandTotalRowTotal] DESC, 'Calendar'[Transaction_Date]
```

75 %

Output

Start	Duration	Text
16:15:58		Establishing Connection
16:20:07		Establishing Connection

3) From Power BI paste the query here into DAX Studio. No errors thrown as DAX Studio is able to read the code format.



Query1.dax* X

Metadata

eb7e3462-92f6-4c05-bd21

Model

- > 4-5-4 Calendar
- > ADDCOLUMNS Demo
- > Average Days
- > **Calendar**
- > CALENDAR Demo
- > CALENDARAUTO Demo
- > CROSSJOIN Demo
- > Customer Lookup
- > DATABASE Demo
- > Date Table (CALENDAR/
- > Employee Lookup
- > EXCEPT Demo
- > Food Inventory
- > GENERATESERIES Demc
- > Measure Table
- > New Employees (INTER:
- > Product Lookup
- > Repeat Customer Sales

```
1 // DAX QUERY
2 DEFINE
3     VAR __DSOCore =
4         SUMMARIZECOLLUMNS(
5             ROLLUPADDISSUBTOTAL('Calendar'[Transaction_Date], "IsGrandTotalRowTotal"),
6             "Customer_Sales_1", 'Measure Table'[Customer Sales 1]
7         )
8
9     VAR __DSOPrimarywindowed =
10        TOPN(502, __DSOCore, [IsGrandTotalRowTotal], 0, 'Calendar'[Transaction_Date], 1)
11
12 EVALUATE
13     __DSOPrimarywindowed
14
15 ORDER BY
16     [IsGrandTotalRowTotal] DESC, 'calendar'[Transaction_date]
```

75 %

Results

Transaction_Date	IsGrandTotalRowTotal	Customer_Sales_1
1/1/2017	True	4252704.879999999
1/2/2017	False	2508.2
1/3/2017	False	2403.35
1/4/2017	False	2220.1
1/5/2017	False	2418.85
1/6/2017	False	2273.85
1/7/2017	False	2787
1/8/2017	False	2638.53
1/9/2017	False	2638.53
1/10/2017	False	2638.53
1/11/2017	False	2638.53
1/12/2017	False	2638.53
1/13/2017	False	2638.53
1/14/2017	False	2638.53
1/15/2017	False	2638.53
1/16/2017	False	2638.53
1/17/2017	False	2638.53
1/18/2017	False	2638.53
1/19/2017	False	2638.53
1/20/2017	False	2638.53
1/21/2017	False	2638.53
1/22/2017	False	2638.53
1/23/2017	False	2638.53
1/24/2017	False	2638.53
1/25/2017	False	2638.53
1/26/2017	False	2638.53
1/27/2017	False	2638.53
1/28/2017	False	2638.53
1/29/2017	False	2638.53
1/30/2017	False	2638.53
1/31/2017	False	2638.53
2/1/2017	False	2638.53
2/2/2017	False	2638.53
2/3/2017	False	2638.53
2/4/2017	False	2638.53
2/5/2017	False	2638.53
2/6/2017	False	2638.53
2/7/2017	False	2638.53
2/8/2017	False	2638.53
2/9/2017	False	2638.53
2/10/2017	False	2638.53
2/11/2017	False	2638.53
2/12/2017	False	2638.53
2/13/2017	False	2638.53
2/14/2017	False	2638.53
2/15/2017	False	2638.53
2/16/2017	False	2638.53
2/17/2017	False	2638.53
2/18/2017	False	2638.53
2/19/2017	False	2638.53
2/20/2017	False	2638.53
2/21/2017	False	2638.53
2/22/2017	False	2638.53
2/23/2017	False	2638.53
2/24/2017	False	2638.53
2/25/2017	False	2638.53
2/26/2017	False	2638.53
2/27/2017	False	2638.53
2/28/2017	False	2638.53
2/29/2017	False	2638.53
3/1/2017	False	2638.53
3/2/2017	False	2638.53
3/3/2017	False	2638.53
3/4/2017	False	2638.53
3/5/2017	False	2638.53
3/6/2017	False	2638.53
3/7/2017	False	2638.53
3/8/2017	False	2638.53
3/9/2017	False	2638.53
3/10/2017	False	2638.53
3/11/2017	False	2638.53
3/12/2017	False	2638.53
3/13/2017	False	2638.53
3/14/2017	False	2638.53
3/15/2017	False	2638.53
3/16/2017	False	2638.53
3/17/2017	False	2638.53
3/18/2017	False	2638.53
3/19/2017	False	2638.53
3/20/2017	False	2638.53
3/21/2017	False	2638.53
3/22/2017	False	2638.53
3/23/2017	False	2638.53
3/24/2017	False	2638.53
3/25/2017	False	2638.53
3/26/2017	False	2638.53
3/27/2017	False	2638.53
3/28/2017	False	2638.53
3/29/2017	False	2638.53
3/30/2017	False	2638.53
3/31/2017	False	2638.53
4/1/2017	False	2638.53
4/2/2017	False	2638.53
4/3/2017	False	2638.53
4/4/2017	False	2638.53
4/5/2017	False	2638.53
4/6/2017	False	2638.53
4/7/2017	False	2638.53
4/8/2017	False	2638.53
4/9/2017	False	2638.53
4/10/2017	False	2638.53
4/11/2017	False	2638.53
4/12/2017	False	2638.53
4/13/2017	False	2638.53
4/14/2017	False	2638.53
4/15/2017	False	2638.53
4/16/2017	False	2638.53
4/17/2017	False	2638.53
4/18/2017	False	2638.53
4/19/2017	False	2638.53
4/20/2017	False	2638.53
4/21/2017	False	2638.53
4/22/2017	False	2638.53
4/23/2017	False	2638.53
4/24/2017	False	2638.53
4/25/2017	False	2638.53
4/26/2017	False	2638.53
4/27/2017	False	2638.53
4/28/2017	False	2638.53
4/29/2017	False	2638.53
4/30/2017	False	2638.53
5/1/2017	False	2638.53
5/2/2017	False	2638.53
5/3/2017	False	2638.53
5/4/2017	False	2638.53
5/5/2017	False	2638.53
5/6/2017	False	2638.53
5/7/2017	False	2638.53
5/8/2017	False	2638.53
5/9/2017	False	2638.53
5/10/2017	False	2638.53
5/11/2017	False	2638.53
5/12/2017	False	2638.53
5/13/2017	False	2638.53
5/14/2017	False	2638.53
5/15/2017	False	2638.53
5/16/2017	False	2638.53
5/17/2017	False	2638.53
5/18/2017	False	2638.53
5/19/2017	False	2638.53
5/20/2017	False	2638.53
5/21/2017	False	2638.53
5/22/2017	False	2638.53
5/23/2017	False	2638.53
5/24/2017	False	2638.53
5/25/2017	False	2638.53
5/26/2017	False	2638.53
5/27/2017	False	2638.53
5/28/2017	False	2638.53
5/29/2017	False	2638.53
5/30/2017	False	2638.53
5/31/2017	False	2638.53
6/1/2017	False	2638.53
6/2/2017	False	2638.53
6/3/2017	False	2638.53
6/4/2017	False	2638.53
6/5/2017	False	2638.53
6/6/2017	False	2638.53
6/7/2017	False	2638.53
6/8/2017	False	2638.53
6/9/2017	False	2638.53
6/10/2017	False	2638.53
6/11/2017	False	2638.53
6/12/2017	False	2638.53
6/13/2017	False	2638.53
6/14/2017	False	2638.53
6/15/2017	False	2638.53
6/16/2017	False	2638.53
6/17/2017	False	2638.53
6/18/2017	False	2638.53
6/19/2017	False	2638.53
6/20/2017	False	2638.53
6/21/2017	False	2638.53
6/22/2017	False	2638.53
6/23/2017	False	2638.53
6/24/2017	False	2638.53
6/25/2017	False	2638.53
6/26/2017	False	2638.53
6/27/2017	False	2638.53
6/28/2017	False	2638.53
6/29/2017	False	2638.53
6/30/2017	False	2638.53
7/1/2017	False	2638.53
7/2/2017	False	2638.53
7/3/2017	False	2638.53
7/4/2017	False	2638.53
7/5/2017	False	2638.53
7/6/2017	False	2638.53
7/7/2017	False	2638.53
7/8/2017	False	2638.53
7/9/2017	False	2638.53
7/10/2017	False	2638.53
7/11/2017	False	2638.53
7/12/2017	False	2638.53
7/13/2017	False	2638.53
7/14/2017	False	2638.53
7/15/2017	False	2638.53
7/16/2017	False	2638.53
7/17/2017	False	2638.53
7/18/2017	False	2638.53
7/19/2017	False	2638.53
7/20/2017	False	2638.53
7/21/2017	False	2638.53
7/22/2017	False	2638.53
7/23/2017	False	2638.53
7/24/2017	False	2638.53
7/25/2017	False	2638.53
7/26/2017	False	2638.53
7/27/2017	False	2638.53
7/28/2017	False	2638.53
7/29/2017	False	2638.53
7/30/2017	False	2638.53
7/31/2017	False	2638.53
8/1/2017	False	2638.53
8/2/2017	False	2638.53
8/3/2017	False	2638.53
8/4/2017	False	2638.53
8/5/2017	False	2638.53
8/6/2017	False	2638.53
8/7/2017	False	2638.53
8/8/2017	False	2638.53
8/9/2017	False	2638.53
8/10/2017	False	2638.53
8/11/2017	False	2638.53
8/12/2017	False	2638.53
8/13/2017	False	2638.53
8/14/2017	False	2638.53
8/15/2017	False	2638.53
8/16/2017	False	2638.53
8/17/2017	False	2638.53
8/18/2017	False	2638.53
8/19/2017	False	2638.53
8/20/2017	False	2638.53
8/21/2017	False	2638.53
8/22/2017	False	2638.53
8/23/2017	False	2638.53
8/24/2017	False	2638.53
8/25/2017	False	2638.53
8/26/2017	False	2638.53
8/27/2017	False	2638.53
8/28/2017	False	2638.53
8/29/2017	False	2638.53
8/30/2017	False	2638.53
8/31/2017	False	2638.53
9/1/2017	False	2638.53
9/2/2017	False	2638.53
9/3/2017	False	2638.53
9/4/2017	False	2638.53
9/5/2017	False	2638.53
9/6/2017	False	2638.53
9/7/2017	False	2638.53
9/8/2017	False	2638.53
9/9/2017	False	2638.53
9/10/2017	False	2638.53
9/11/2017	False	2638.53
9/12/2017	False	2638.53
9/13/2017	False	2638.53
9/14/2017	False	2638.53
9/15/2017	False	2638.53
9/16/2017	False	2638.53
9/17/2017	False	2638.53
9/18/2017	False	2638.53
9/19/2017	False	2638.53
9/20/2017	False	2638.53
9/21/2017	False	2638.53
9/22/2017	False	2638.53
9/23/2017	False	2638.53
9/24/2017	False	2638.53
9/25/2017	False	2638.53
9/26/2017	False	2638.53
9/27/2017	False	2638.53
9/28/2017	False	2638.53
9/29/2017	False	2638.53
9/30/2017	False	2638.53
10/1/2017	False	2638.53
10/2/2017	False	2638.53
10/3/2017	False	2638.53
10/4/2017	False	2638.53
10/5/2017	False	2638.53
10/6/2017	False	2638.53

DaxStudio - 2.11.2

File Home Advanced Help Layout

Run Cancel Clear Cache Output

Cut Undo DAX Format Query To Upper Comment Find Load Perf Data All Queries Server Timings

Copy Redo To Lower Uncomment Replace Power BI Query Plan Connect Refresh Metadata

Paste Swap Delimiters Merge XML

Query Edit Format Find Traces Connection

Query1.dax* X

Metadata eb7e3462-92f6-4c05-bd21 Model

4-5-4 Calendar ADDCOLUMNS Demo Average Days Calendar CALENDAR Demo CALENDARAUTO Demo CROSSJOIN Demo Customer Lookup DATATABLE Demo Date Table (CALENDAR/ Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo Measure Table New Employees (INTER Product Lookup Repeat Customer Sales

// DAX QUERY
DEFINE
VAR __DSOCore =
SUMMARIZECOLUMNS(
ROLLUPADDISSUBTOTAL('Calendar'[Transaction_Date], "IsGrandTotalRowTotal"),
"Customer_Sales_1", "Measure Table"[Customer Sales 1])

VAR __DSOPrimarywindowed =
TOPN(502, __DSOCore, [IsGrandTotalRowTotal], 0, 'Calendar'[Transaction_Date], 1)

EVALUATE
__DSOPrimarywindowed

ORDER BY
[IsGrandTotalRowTotal] DESC, 'Calendar'[Transaction_Date]

75 %

Total SE CPU Line Subclass Duration CPU Rows KB Query

Total	SE CPU	Line	Subclass	Duration	CPU	Rows	KB	Query
9 ms	0 ms	2	Scan	3	0	853	14	WITH \$Expr0:= (CAST (PFCAST ('Sales by Store'[qu
FE	SE							
6 ms	3 ms							
66.7%	33.3%							
SE Queries	SE Cache							
1	0							
	0.0%							

5) Server Timings available from here for optimisation purposes.

Output Results Query History Server Timings

Ready In 2, Col 5 localhost:50724 15.1.48.30 23058 502 rows 00:00:0

Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill Table tools Measure tools

Name: 4-5-4 Calendar

Measure Calculators Relationships Calculations

Transaction Date Customer Sales 1 Customer Sales 2 Customer Sales 3

Transaction Date	Customer Sales 1	Customer Sales 2	Customer Sales 3
21/01/2017	\$2,100		
22/01/2017	\$2,400	\$2,	
23/01/2017	\$2,800		
24/01/2017	\$2,200		
25/01/2017	\$2,410		
26/01/2017	\$2,274		
27/01/2017	\$2,707	\$2,	
28/01/2017	\$2,819	\$2,	
29/01/2017	\$2,877	\$2,	
30/01/2017	\$2,800	\$2,	
31/01/2017	\$2,510	\$2,	
Total	\$4,252,705	\$4,271	

Performance analyzer

Start recording Refresh visuals Stop
Clear Export

Name Duration (ms)

Recording started (26/10/2020 21:29:01) 1

Macro 1396

Other 7

Copy query 152

Changed a filter 1231

Macro 1396

DAX query 11

Visual display 152

Other 1231

Copy query 152

Changed a filter 296

Macro 296

DAX query 7

Visual display 77

Other 172

Copy query 77

Recording started (26/10/2020 22:18:34) 7145

Refreshed visual 275

Macro 275

Changed a filter 275

Macro 7145

Visualizations Fields

Search

Measure Table Advanced CALCULATE Advanced Time Intelligence Calculated Table Joins Iterators Performance Tuning Customer Sales 1 Customer Sales 2 Customer Sales 3 Relationship Functions Scalar Measures Table & Filter Functions Tips & Best Practices 4-5-4 Calendar // DAX Query/DEFINE VAR _D...

Rows Transaction Date

Columns Add data fields here

Values Customer Sales 1 Customer Sales 3

Drill through

Cross-report Off On

Keep all filters On

Add drill-through fields here

CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Performance Tuning ADDCOLUMNS Demo Average Days

Learn more about optimizing your report's performance on our [support site](#). Find help tuning your report from specialist Power BI partners on [Accounts](#).

6) Go back to Power BI & load Customer Sales 3 to the Values. You will notice that the query run significantly longer.



Maven Roasters - Analysis - Performance Tuning - Power BI Desktop

File Home Insert Modeling View Help Format Data / Drill

Cut Copy Format painter

Get data - Excel Power BI datasets - SQL Server Enter data Recent sources - Transform Refresh data - New visual Text box More visuals - Quick measure measure Publish

Calendars Data Queries Insert Calculations Share

Customer Sales 1 - Customer Sales 3

7) The DAX query took 6916 milliseconds to run significantly longer code potentially poorly written.

Performance analyzer

Start recording Refresh visuals Stop Clear Export

Name	Duration (ms)
Recording started (26/10/2020 21:29:01)	-
Matrix	-
Other	-
Copy query	-
Changed a filter	-
Matrix	1396
DAX query	11
Visual display	152
Other	1231
Copy query	-
Changed a filter	-
Matrix	296
DAX query	7
Visual display	77
Other	172
Copy query	-
Recording started (26/10/2020 22:18:34)	-
Matrix	-
Other	-
Copy query	-
Changed a filter	-
Matrix	275
DAX query	7145
Visual display	103
Other	127
Copy query	-

Visualizations Fields

Search

Measure Table

Advanced CALCULATE

Advanced Time Intelligence

Calculated Table Joins

Iterators

Performance Tuning

Customer Sales 1

Customer Sales 2

Customer Sales 3

Relationship Functions

Scalar Measures

Table & Filter Functions

Tips & Best Practices

4-5-4 Calendar

// DAX Query//DEFINE VAR _DS...

Data

DayName

DayOfMonthNumber

DayOfWeek

FiscalMonthName

FiscalMonthNumber

FiscalMonthOfQuarter

FiscalMonthYear

FiscalQuarter

FiscalQuarterYear

FiscalWeekOfYear

FiscalYear

FiscalYear Hierarchy

ADDCOLUMNS Demo

Average Days

Learn more about optimizing your report's performance on our [support site](#). Find help tuning your report from specialist Power BI partners on [Accounts](#).

CALCULATE Table & Filter Functions Joins Relationship Functions Iterators Advanced Time Intelligence Performance Tuning

Page 9 of 9

DaxStudio - 2.11.2

File Home Advanced Help Layout

Cut Undo DAX To Upper Comment Find Load Perf Data All Queries Server Timings

Copy Redo Format Query To Lower Uncomment Replace Query Plan Connect Refresh Metadata

Paste Swap Delimiters Merge XML

Run Cancel Clear Cache Output

Query Edit Format Find Power BI Traces Connection

Query1.dax X

Metadata eb7e3462-92f6-4c05-bd21 Model

4-5-4 Calendar ADDCOLUMNS Demo Average Days Calendar CALENDAR Demo CALENDARAUTO Demo CROSSJOIN Demo Customer Lookup DATATABLE Demo Date Table (CALENDAR/ Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo Measure Table New Employees (INTER Product Lookup Repeat Customer Sales

// DAX QUERY
DEFINE
VAR __DSOCore =
SUMMARIZECOLLUMNS(
ROLLUPADDISSUBTOTAL('Calendar'[Transaction_Date], "IsGrandTotalRowTotal"),
"Customer_Sales_1", "Measure Table'[Customer Sales 1]",
"Customer_Sales_3", "Measure Table'[Customer Sales 3]"
)

VAR __DSOPrimarywindowed =
TOPN(502, __DSOCore, [IsGrandTotalRowTotal], 0, 'Calendar'[Transaction_Date], 1)

EVALUATE
__DSOPrimarywindowed

ORDER BY
[IsGrandTotalRowTotal] DESC, 'Calendar'[Transaction_Date]

75 %

Total SE CPU Line Subclass Duration CPU Rows KB Query

0 ms 0 ms

FE SE

0 ms 0 ms 0.0% 0.0% 0.0% 0.0% 0.0% 0.0%

SE Queries SE Cache

0 0 0.0%

Still Processing!

8) Paste the Power BI code here & run it you will notice its taking long to obtain the results.

Output Results Query History Server Timings

Running Query..

Ln 4, Col 9 localhost:50724 15.1.48.30 23058 502 rows 00:07.5



Query1.dax* X

Metadata eb7e3462-92f6-4c05-bd21 Model

4-5-4 Calendar ADDCOLUMNS Demo Average Days Calendar CALENDAR Demo CALENDARAUTO Demo CROSSJOIN Demo Customer Lookup DATATABLE Demo Date Table (CALENDAR) Employee Lookup EXCEPT Demo Food Inventory GENERATESERIES Demo Measure Table New Employees (INTER) Product Lookup Repeat Customer Sales

```
1 // DAX QUERY
2 DEFINE
3     VAR __DSOCore =
4         SUMMARIZECOLUMNS(
5             ROLLUPADDISSUBTOTAL('Calendar'[Transaction_date], "IsGrandTotalRowTotal"),
6             "Customer_Sales_1", "Measure Table'[Customer Sales 1]",
7             "Customer_Sales_3", "Measure Table'[Customer Sales 3]
8         )
9
10    VAR __DSOPrimarywindowed =
11        TOPN(502, __DSOCore, [IsGrandTotalRowTotal], 0, 'Calendar'[Transaction_date], 1)
12
13 EVALUATE
14     __DSOPrimarywindowed
15
16 ORDER BY
17     [IsGrandTotalRowTotal] DESC, 'Calendar'[Transaction_date]
```

75 %

9,842 ms 500 ms 2 Scan 5 0 853 14 WITH \$Expr0:= (CAST (PFCAST ('Sales by Store'[que
9,341 ms 501 ms 4 Scan 265 266 907,841 7,093 SELECT 'Calendar'[Transaction Date], 'Sales by Store'[F
FE SE 6 Scan 0 0 8,100 64 SELECT 'Product Lookup'[RowNumber], 'Product Look
94.9% 5.1%

SE Queries: 4 SE Cache: 0 0.0%

Output Results Query History Server Timings

Ln 4, Col 9 localhost:50724 15.148.30 23058 502 rows 00:09:8

9) Almost 95% of the query is taking place within the formula engine. The next step would be to analyse what the measure is doing to be able to enhance the code performance for efficiency.

DAX Optimization Workflow



- 1 Start Performance Analyzer in Power BI Desktop**
 - Click "Start Recording", any actions will be logged and displayed in the Performance Analyzer pane
- 2 Analyze your query load times**
 - Each part of the query will likely take a different amount of time, and if the **DAX Query** load time is long (100-200ms or more) you can likely optimize your code using DAX Studio
 - **Visual display** load time may be decreased by changing the visual type or reducing data points
- 3 Copy & paste or import queries into DAX Studio**
 - Copy and paste an individual query and analyze its performance in DAX Studio, or export the entire performance analyzer results and use **Analyze Performance Data** to import a file containing all queries
- 4 Use DAX studio to optimize your code**
 - DAX Studio uses some specific functions which allow you to analyze your DAX queries (i.e. **DEFINE**, **EVALUATE** and **ORDER BY**)
 - In an optimized query, the storage engine (SE) should carry the majority of the CPU workload