Power Query Recipes

Dimensional Modeling Edition

Dimensional Modeling
for the Excel and Power BI Pro
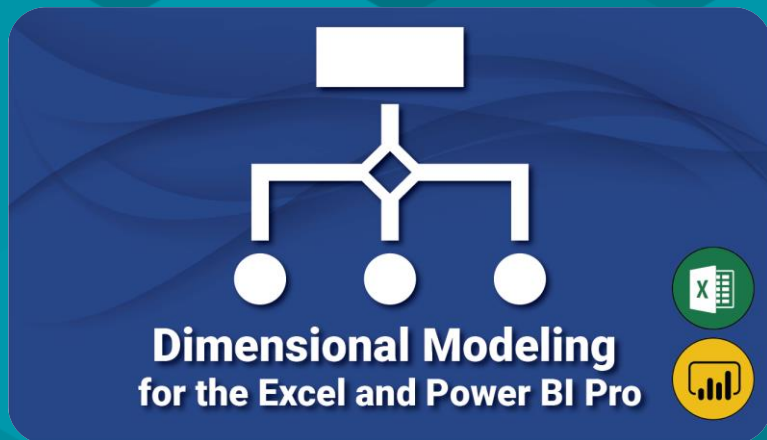
About Power Query Recipes

These cards are a set of ready-to-use Power Query Recipes to lead you step-by-step through a variety of data transformations and techniques. These cards are a useful reference where you can look up the exact steps needed for a particular data transformation pattern. We hope you find them as handy as we do ourselves!

Using the Recipe Cards

The Table of Contents and the recipe cards are fully linked so you can easily navigate between them. Clicking on the recipe number or name on the Table of Contents will take you directly to that recipe. Likewise, if a recipe card references another recipe, clicking that reference will take you directly to the referenced recipe.

These recipes are also linked to corresponding videos in our Dimensional Modeling for the Excel and Power BI Pro course (both the Excel and Power BI versions), and are indicated with the video icon shown here:

Because you are enrolled in both versions of the course, clicking the corresponding video icon will take you to that video on the course site. Please note that you will first need to login to the course site before being able to watch the video. *You will not be able to watch the video if you are not logged into the Skillwave Training site or if you are not enrolled in the course.*

Power Query Recipes — Table of Contents

00. Best Practice Architectures

Look for the Pattern Difficulty scale:

No Code   Lo Code   Pro Code

© www.skillwave.training

Power Query Recipes – Table of Contents

60. Date & Time Techniques

Link to Corresponding Course

Look for the Pattern Difficulty scale:

No Code · Lo Code · Pro Code

© www.skillwave.training

0. Best Practice Architectures

MASTER YOUR DATA IN EXCEL & POWER BI

Architecture Recipes: Options when using Files as a Data Source

Individual Files

Save the new data file with a different name

⬇

Open the Excel Workbook/ Power BI Report

⬇

Edit the Query

⬇

Update the Source path to the new file

Save the new data file over the original

All Files in a Folder

Save the new data file under a new name (in the same folder)

⬇

Open the Excel Workbook/ Power BI Report

⬇

The Benefits

Build your solution once and then re-use your existing business intelligence reports and logic!

* If you are using Power BI, you may be able to schedule the refresh

Close the Power Query Editor ➡ **Refresh the Data*** ➡ **Update the Visuals**

Master Your Data in Excel & Power BI

Architecture Recipes: Creating a Staging Query

Power BI Desktop

Step 1:   Prepare the Source Data

Connect to your Data source

Perform any desired transformations

Name your query

Step 2:   Load as a Staging query

Right click the Query name

Uncheck Enable Load

Queries [1]

A^B_C *MyData*

- Copy
- Paste
- ✕ Delete
- Rename
- Enable load
- ✓ Include in report refresh
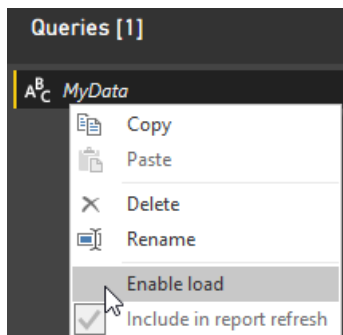
Staging queries do not consume any processor or RAM until they are called by another query. They can be very helpful in order to act as a starting point for other queries!

Excel

Step 1:   Prepare the Source Data

Connect to your Data source

Perform any desired transformations

Name your query

Step 2:   Load as a Staging query

Go to Home → Close & Load → Close & Load To...

Choose Only Create Connection

Uncheck "Add this data to the Data Model"
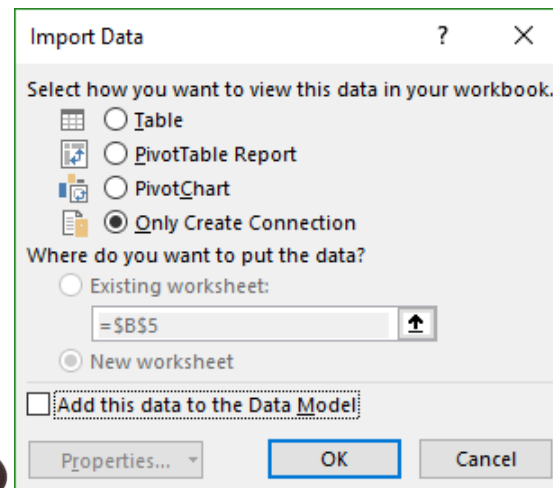
Import Data   ?   ✕

Select how you want to view this data in your workbook.

- ▦ ○ Table
- ○ PivotTable Report
- ○ PivotChart
- ⦿ Only Create Connection

Where do you want to put the data?

- ○ Existing worksheet:
  =$B$5
- ⦿ New worksheet

☐ Add this data to the Data Model

Properties...   ▾   OK   Cancel

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data in Excel & Power BI

Pattern Difficulty

Architecture Recipes: Modifying Query Load Destinations

Excel

Step 1: Display the Queries & Connections Pane

Excel 2010/2013	Power Query → Show Pane
Excel 2016	Data → Show Queries
Excel 2019/365	Data → Queries & Connections

Step 2: Modify the Load Destination

Right click the desired query → Load To...

Modify the load behavior as desired

Import Data ? ✕

Select how you want to view this data in your workbook.
- ▦ ○ Table
- ▣ ○ PivotTable Report
- ▤ ○ PivotChart
- ▤ ● Only Create Connection

Where do you want to put the data?
- ○ Existing worksheet:
  =$B$5 ⬆
- ○ New worksheet

☐ Add this data to the Data Model

Properties... ▾   OK   Cancel
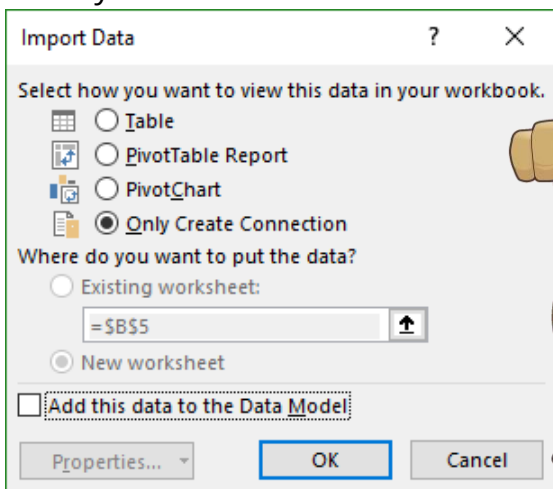
Power BI Desktop

Go to Home → Edit Queries

Right click the Query name

(Un-) Check "Enable Load"

Click Close & Apply

Go to Home → Refresh

Queries [1]

A^BC *MyData*

- 🗐 Copy
- 📋 Paste
- ✕ Delete
- 🖼 Rename
- Enable load
- ☑ Include in report refresh
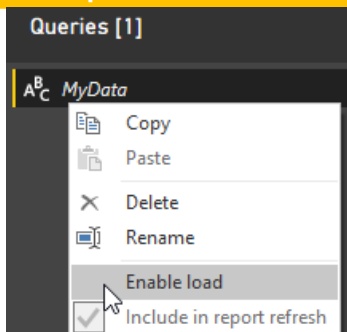
Excel Tips!

1) If you accidentally loaded a query to a table, you can convert it to a Connection Only query by simply deleting the worksheet!

2) Change your default Load Behaviour by modifying the Query Options found here:

Excel 2010/2013	Power Query → Options
Excel 2016	Data → New Query → Query Options
Excel 2019/365	Data → Get Data → Query Options

Dimensional Modeling Recipes: Creating Fact Tables

Date	Item	Category	Units
2011-01-02	Omelette	Breakfast	3
2011-01-02	Garlic Bread	Soup/Salad	2
2011-01-02	Chowder	Soup/Salad	2
2011-01-02	Omelette	Breakfast	5
2011-01-02	Pizza	Entrees	4
2011-01-02	Eggs Benny		

Date	Item	Units
2011-01-02	Omelette	3
2011-01-02	Garlic Bread	2
2011-01-02	Chowder	2
2011-01-02	Omelette	5
2011-01-02	Pizza	4
2011-01-02	Eggs Benny	5

> Since [Category] is contained in my Items table, I don't need it in my Fact table as well.

Why you care

Separating Fact and Dimension tables is an important component of building effective Power Pivot and Power BI models

Creating Fact Tables Recipe

If you do not already have a staging query for your data...

Create a staging query for the original data

Edit the original data query

Right click the final step → Extract Previous

Name the original "Staging-" <something descriptive>

Load the query as Connection Only *(Recipe 0.110)*

Create a new "fact table" query

Reference the (new) Staging query

Go to Home → Choose Columns

Un-check any column that doesn't meet one of these criteria:

Columns that will be aggregated (the Facts)

Columns that will act as Foreign Keys to link to Dimensions

(Re) set the Data Types for each column

Load the table to the Data model

Link the new Fact table to any related Dimension tables

For more data patterns check out our online shop
© www.skillwave.training

Recipe: 0.410.2008

Dimensional Modeling Recipes: Creating Dimension Tables

Date	Item	Category	Units
2011-01-02	Omelette	Breakfast	3
2011-01-02	Garlic Bread	Soup/Salad	2
2011-01-02	Chowder	Soup/Salad	2
2011-01-02	Omelette	Breakfast	5
2011-01-02	Pizza	Entrees	4
2011-01-02	Eggs Benny	Breakfast	

Item	Category
Omelette	Breakfast
Garlic Bread	Soup/Salad
Chowder	Soup/Salad
Pizza	Entrees
Eggs Benny	Breakfast

> Your dimension must have a Primary Key column (like **Item** in my new table.) As a hint, these columns always show the same number of Unique and Distinct values in Power Query's column profiling tools!

Why you care

Separating Fact and Dimension tables is an important component of building effective Power Pivot and Power BI models

Creating Dimension Tables Recipe

If you do not already have a staging query for your data...

Create a staging query for the original data

Edit the original data query

Right click the final step → Extract Previous

Name the original "Staging-" <something descriptive>

Load the query as Connection Only (*Recipe 0.110*)

Create a new "dimension" query

Reference the (new) Staging query

Select the column(s) you wish to keep in your Dimension

Right click any column → Remove other columns

Select All columns

Right click any column → Remove duplicates

Ensure you have a column that can act as a Primary Key

(Re) set the Data Types for each column

Load the table to the Data model

Link the new dimension to your table(s)

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data in Excel & Power BI

Dimensional Modeling Recipes: Solving Many to Many Joins

Sales

Date	Category	Amount
2018-01-01	Beer	175
2018-01-01	Beer	175
2018-01-01	Beer	175
2018-01-02	Beer	1785
2018-01-01	Wine	512
2018-01-01	Burgers	72
2018-01-02	Wine	873
2018-01-02	Wine	873
2018-01-02		244

Budgets

Date	Category	Amount
2018-01-31	Beer	43,683
2018-01-31	Wine	101,558
2018-01-31	Burgers	13,677
2018-02-28	Beer	104,637
2018-02-28	Wine	21,173
2018-02-28		10,884

Dimension
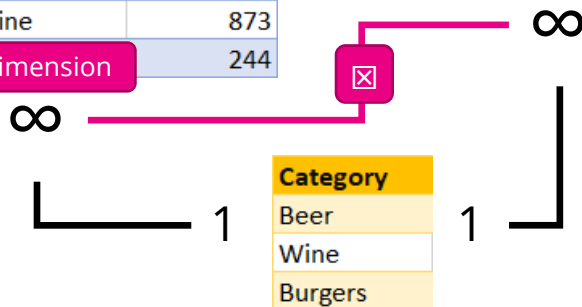
This recipe enforces "referential integrity", ensuring that items added to either table appear in the Bridge Table!

Dimension

∞

☒

∞

Category
Beer
Wine
Burgers

1   1

Getting a message that you can't create a relationship due to multiple distinct values in each column? This recipe may be part of the solution!

You may need to combine this recipe with the recipe for Creating Composite Keys *(Recipe 0.530.x)*
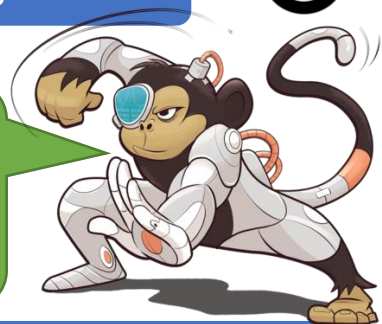
Creating Bridge Tables Recipe

Create staging queries for each table
- Reference each query
- Right click column → Remove other columns
- Right click column → Remove duplicates
- Load as Connection Only *(Recipe 0.110)*

Create a new "dimension" query
- Reference one of the new staging tables
- Append the second table
- Remove duplicates
- Load the new table to the data model

Link the new dimension to your tables

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data in Excel & Power BI

Pattern Difficulty

Dimensional Modeling Recipes: Solving Many to Many Joins

The "Many" Table

Date	Account	Dept	Amount
2008-01-31	61510	150	642.62
2008-01-31	61520	150	1,359.62
2008-01-31	61530	150	261.73
2008-01-31	61540	150	7.16
2008-01-31	61520	155	1,185.46
2008-01-31	61530	155	33.26
2008-01-31	61540	155	1,901.98
2008-02-29	61510	150	2,957.41
2008-02-29	61520	150	6,210.83
2008-02-29	61530	150	1,586.26

The "One" Table

Account	Dept	Category
61510	150	Liquor
61520	150	Wine
61530	150	Beer
61540	150	Cider
61520	155	Beer
61530	155	Wine
61540	155	Liquor

1

∞

The trick to this pattern is identifying that combining the fields in the "One" table results in unique values.

Don't forget to use a delimiter when joining the columns. Without it, 111 could mean 1-11 or 11-1.

You may need to combine this recipe with the recipe for Creating Bridge Tables *(Recipe 0.520.x)*

Creating Composite Keys Recipe

Edit the "Many" Table

Hold CTRL and select the columns to join

Go to the *Transform* tab → Merge Columns

Merge the columns with a delimiter

Edit the "One" Table

Hold CTRL and select the columns to join

Go to the *Add Column* tab → Merge Columns

Merge the columns with a delimiter

Load the tables to the Data model

Preserving the original columns in the "One" table allows you to use them in filters later!

Dimensional Modeling Recipes: Recommended Practices

A "Snowflaked" dimension is a dimensional table that only has a single relationship. This level of abstraction is unnecessary in Power BI and Power Pivot models.

"Star" Table

Accounts
- Account
- **Div**
- Category
- Lnk_AccDiv

Divisions
- **Div**
- Region
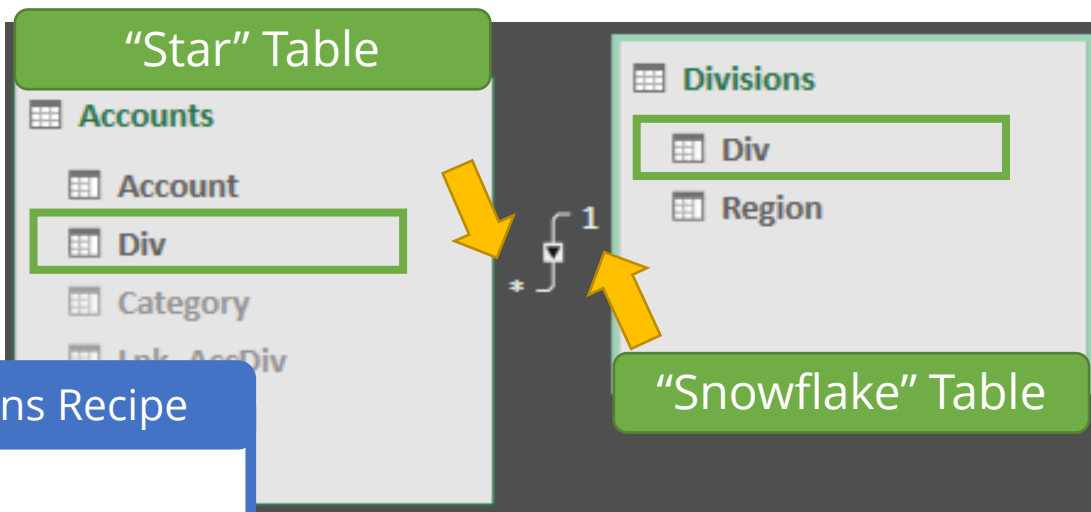
"Snowflake" Table

Flattening Snowflaked Dimensions Recipe

Edit the "Star" table
- Go to Home → Merge Queries
- Merge the Star table to the Snowflake table
- Select the common fields for the merge → OK
- Expand desired fields from the Snowflake column
- Load to the Data Model
- Create the table relationships

Edit the "Snowflake" table
- Change to load to Connection Only *(Recipe 0.120)*

Snowflakes belong on mountains, not in your data model

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

Excel   Power BI

For more data patterns check out our online shop
© www.skillwave.training

Recipe: 0.540.2007

ISBN 978-1-61547-246-8   US$19.95
9 781615 472468

50. Ranking

Master Your Data in Excel & Power BI

Number Grouped Rows Recipe

Initial Preparation

Sort any **[Dimension]** column(s) to group the data

Sort the **[Fact]** column → Descending

Re-apply the column data types

Group the Data

Go to Transform → Group → Advanced

Group by the **[Dimension]** column(s)

Add a new column called "Data" → All Rows

Add Row Numbers

Go to Add Column → Custom Column

    Name: Custom

    Formula: `=Table.AddIndexColumn([Data], "Index", 1, 1)`

Right click the **[Custom]** column → Remove Other Columns

Expand all columns from **[Custom]**

Re-order the columns if desired

DBAs know this output as using SQL's ROW_NUMBER with PARTITION
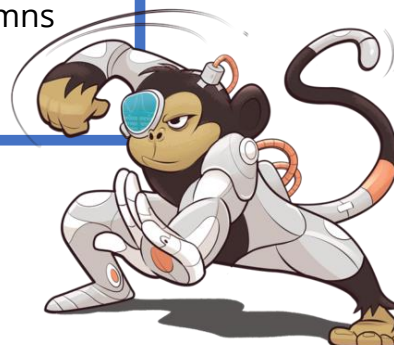
Source Table

Group	Category	Sales
Food	Breakfast	99,562.64
Food	Soups/Salads	113,642.33
Food	Sandwiches	217,510.36
Alcohol	Draft Beer	132,882.83
Alcohol	Wine	217,492.74
Alcohol	Liquor	50,939.94
Alcohol	Coolers/Ciders	22,628.14
Alcohol	Canned Beer	49,996.55

Dimension	Fact

Table With Numbered Rows

Group	Category	Sales	Index
Food	Sandwiches	217,510.36	1
Food	Soups/Salads	113,642.33	2
Food	Breakfast	99,562.64	3
Alcohol	Wine	217,492.74	1
Alcohol	Draft Beer	132,882.83	2
Alcohol	Liquor	50,939.94	3
Alcohol	Canned Beer	49,996.55	4
Alcohol	Coolers/Ciders	22,628.14	5

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

Excel   Power BI

For more data patterns check out our online shop
© www.skillwave.training

60. Date & Time Techniques

Master Your Data in Excel & Power BI

Pattern Difficulty

Overview: Creating Dynamic Calendar Tables Part 1

Step 1:   Create Calendar Boundaries

	12 Month Calendars Dec 31 Year End	12 Month Calendars Non-std Year Ends	364 Day Calendars 445, 454, 544, 13 Mth
Create Helper Queries		Create *YEMonth* Query based on Recipe 60.110.x	Create *Start364* Query based on Recipe 60.110.x
Generate Calendar Boundaries (Start & End of Year)	Create *StartDate* & *EndDate* based on Recipe 60.100.x	Create *StartDate* & *EndDate* based on Recipe 60.105.x	Create *StartDate* & *EndDate* based on Recipe 60.105.x
Adjust Boundaries for non-standard Year Ends		Shift the *StartDate* & *EndDate* based on Recipe 60.120.x	Shift the *StartDate* & *EndDate* based on Recipe 60.130.x

Step 2:   Create the Calendar Table

Fill In the Calendar depending on what type you need	Calendars that span StartDate to EndDate Recipe 60.210.x

Step 3:   Add Period ID's for non-standard year ends   *(see Recipe 60.002.x)*

Master Your Data in Excel & Power BI

Pattern Difficulty

Overview: Creating Dynamic Calendar Tables Part 2

Step 1: Create Calendar Boundaries *(see Recipe 60.001.x)*

Step 2: Create the Calendar Table *(see Recipe 60.001.x)*

Step 3: Create the Fiscal Period columns

	12 Month Calendars Non-std Year Ends	364 Day Calendars 445, 454, 544	364 Day Calendars 13 x 4-Weeks
Create PeriodID Columns		Create PeriodIDs based on Recipe 60.410.x	Create PeriodIDs based on Recipe 60.510.x
Create Fiscal Period Columns	Create Fiscal Period Columns based on Recipe 60.330.x	Create Fiscal Period Columns based on Recipe 60.430.x	Create Fiscal Period Columns based on Recipe 60.530.x

For more data patterns check out our online shop
© www.skillwave.training

Dynamic Calendar Boundaries:
Standard December 31 Year Ends

Month End	Category	Amount
2008-01-31	Beer	226.22
2008-01-31	Beer	1657.65
2008-01-31	Cider	10.55
2008-01-31	Liquor	473.05
2008-02-29	Beer	984.34
2008-02-29	Beer	2398.14
2008-02-29	Cider	64.01

Determine the StartDate

Reference table with _earliest_ date

Right click **[Date]** → Remove Other Columns

Filter **[Date]** → Date Filters → Is **Earliest**

Right click **[Date]** → Remove duplicates

Change the data type to a Date*****

Transform → Date → Year → **Start of Year**

Right click the date cell → Drill down **†**

Call query **StartDate** (no spaces)

Load as Connection Only

Determine the EndDate

Reference table with _latest_ date

Right click **[Date]** → Remove Other Columns

Filter **[Date]** → Date Filters → Is **Latest**

Right click **[Date]** → Remove duplicates

Change the data type to a Date*****

Transform → Date → Year → **End of Year**

Right click the date cell → Drill down **†**

Call query **EndDate** (no spaces)

Load as Connection Only

Date
1 2008-01-01

- Copy
- Date Filters
- Replace Values...
- 2 Drill Down
- Add as New Query

= #"Calculated Start of Year"{0}[Date]

2008-01-01 ← StartDate

Continue with your preferred Dynamic Calendar Recipe (60.2xx.x)

* This step is important if you are not forcing to start/end of period

† Ensure that you click the cell to drill down, not the header of the table. You need your new formula to read as follows:
`=#"<previous step name>"{0}[Date Column Name]`

Master Your Data in Excel & Power BI

Dynamic Calendar Boundaries: Non-Standard Year Ends

Generating Calendar *StartDate* (*EndDate*) Queries

1. Set up required "Helper" queries per Recipe 60.110.x

2. Reference table with *earliest* (*latest*) date

3. Right click **[Date]**→ Remove Other Columns

4. Filter **[Date]** → Date Filters → Is *Earliest* (*Latest*)

5. Right click **[Date]** → Remove duplicates

6. Transform date to *Start* (*End*) *of Year*

7. Apply "Year End Adjustment" steps
   a) For 12-Month Calendars, see Recipe 60.120.x
   b) 445, 454, 544 (ISO) or 13-Month Calendars, see Recipe 60.130.x

8. Change the data type to a Date*

9. Right click the date cell→ Drill down †

10. Call query *StartDate* (*EndDate*) – with no spaces

11. Load as Connection Only

Continue with your preferred Dynamic Calendar Recipe (60.2xx.x)

* Be Safe! Failure to apply Step 7 could lead to a hard-coded date when applying Step 8!

† When performing Step 8, ensure that you right click the cell to drill down (as shown at right), not the column header on the table!
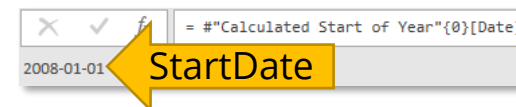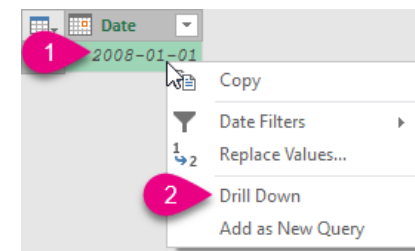
Date
1 2008-01-01
   Copy
   Date Filters
   Replace Values...
2 Drill Down
   Add as New Query

= #"Calculated Start of Year"{0}[Date]
2008-01-01   StartDate

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!
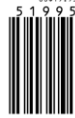Excel   Power BI

For more data patterns check out our online shop
© www.skillwave.training

Recipe: 60.105.2004

ISBN 978-1-61547-246-8   US$19.95
51995>
9 781615 472468

Master Your Data in Excel & Power BI

Pattern Difficulty

Helper Queries for Custom Calendars

12-Month Calendars
(Year Ends other than December 31)

1. Create a new Blank Query
2. Name the query **YEMonth**
3. In the formula bar, enter the calendar month number of your fiscal year end
4. Load as Connection Only

364-Day Year Ends
(445, 454, 544, & 13-Month)

1. Create a new Blank Query
2. Name the query **Start364**
3. In the formula bar, enter the start date of *any* valid 364-day fiscal year end for your company in the following format:

   =#date(yyyy,mm,dd)

   e.g.   =#date(2020,12,28)
4. Load as Connection Only

Return to Recipe 60.105.x – Step 2

Could I use a Parameter for this instead of a Query? Absolutely!

Why make these simple queries?

These recipes are intended to act as Helper Queries in order to make it easier to generate custom calendars for non-standard year ends. Refer to Recipe 60.105.x to review the steps that require these queries.

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data in Excel & Power BI

Year End Adjustment for Non-Standard Year Ends: 12-Month Year End other than December 31

Determine the StartDate

Follow Recipe 60.105.x Steps 1-6

Step 7:

Go to Add Column → Custom Column

Use the following formula: †

```
=Date.AddMonths(
   [Date] ,
   YEMonth -12
   )
```

Click OK

Right click **[Date]** † → Remove

Determine the EndDate

Follow Recipe 60.105.x Steps 1-6

Step 7:

Go to Add Column → Custom Column

Use the following formula: †

```
=Date.AddMonths(
   [Date] ,
   YEMonth
   )
```

Click OK

Right click **[Date]** † → Remove

Return to Recipe 60.105.x – Step 8

Precedents Required!

This pattern requires the **YEMonth** query, which is generated in Recipe 60.110.x

† No "Date" column ?

If your dates column is not called "Date", then replace both instances of **[Date]** with **[<your date column name>]**

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!
Excel   Power BI

For more data patterns check out our online shop
© www.skillwave.training

Recipe: 60.120.2008

ISBN 978-1-61547-246-8
US$19.95
51995>
9 781615 472468

Master Your Data in Excel & Power BI

Year End Adjustment for Non-Standard Year Ends: 364-Day Calendars (445, 454, 544, & 13 Month)

Determine the StartDate

Follow Recipe 60.105.x Steps 1-6

Step 7:

Go to Add Column → Custom Column

Use the following formula: †

```
=Date.AddDays(
Start364,
364 * Number.Round(
  Duration.Days(
   Duration.From([Date]-Start364)
  ) /364 ,
  0
  )
)
```

Click OK

Right click **[Date]** † → Remove

Determine the EndDate

Follow Recipe 60.105.x Steps 1-6

Step 7:

Go to Add Column → Custom Column

Use the following formula: †

```
=Date.AddDays(
Start364,
364 * Number.RoundUp(
  Duration.Days(
   Duration.From([Date]-Start364)
  ) /364 ,
  0
  ) - 1
)
```

Click OK

Right click **[Date]** † → Remove

Return to Recipe 60.105.x – Step 8

Precedents Required!

This pattern requires the *Start364* query, which is generated in Recipe 60.110.x

† No "Date" column ?

If your dates column is not called "Date", then replace both instances of **[Date]** in this recipe with:
[*<your date column name>*]

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

📹 Excel   📹 Power BI

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data in Excel & Power BI

Dynamic Calendar Recipe: From Start Date to End Date

Determine the Calendar boundaries
- Create **StartDate** query *(Recipe 60.10x.x)*
- Create **EndDate** query *(Recipe 60.10x.x)*

Create a Blank Query
- Enter the following in the formula bar:
  `={Number.From(StartDate)..Number.From(EndDate)}`
- Go to List Tools → To Table → OK
- Set **[Column1]** to a Date data type
- Rename **[Column1]** to Date
- Rename the table to Calendar

Add New Columns
1. Select the **[Date]** column
2. Go to Add Column → Date → <the format to add>
3. Repeat Steps 1-2 as many times as needed

Load the Calendar to the intended destination

Given only a **StartDate** and an **EndDate**...

	List
1	39448
2	39449
3	39450
4	39451
5	39452
6	39453

	Date
1	2008-01-01
2	2008-01-02
3	2008-01-03
4	2008-01-04
5	2008-01-05
6	2008-01-06
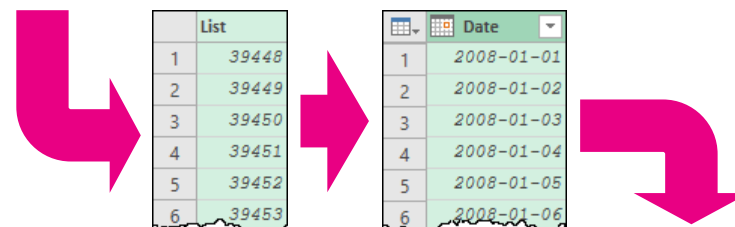
Fully Dynamic 'Calendar' Table

	Date	Year	Month	Month Name	Day
1	2008-01-01	2008	1	Jan	1
2	2008-01-02	2008	1	Jan	2
3	2008-01-03	2008	1	Jan	3
4	2008-01-04	2008	1	Jan	4
5	2008-01-05	2008	1	Jan	5
6	2008-01-06	2008	1	Jan	6

To shorten Month or Day names, go to Transform → Extract → First Characters

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!
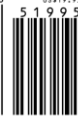Excel   Power BI

For more data patterns check out our online shop
© www.skillwave.training
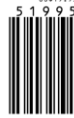
Master Your Data in Excel & Power BI

Fiscal Period Columns for 12-Mth Non-Std Year Ends

To add the period column(s):

Go to Add Column → Custom Column

Name:

Fiscal (Month, Quarter or Year)

Formula:

Enter the appropriate formula based on the column you need, as shown in the table on the right

Click OK

Set the column's data type

Fiscal...	Required Column(s)	Formula
Month	**[Date]**	```Date.Month(` ` Date.AddMonths(` ` [Date] , - YEMonth)` `)```
Quarter	**[Fiscal Month]**	```Number.RoundUp([Fiscal Month] / 3)```
Year	**[Date]**	```if Date.Month([Date]) > YEMonth` `then Date.Year([Date]) + 1` `else Date.Year([Date])```

Which data type should I choose?

Setting your Fiscal Year to text doesn't compromise the sort order and means that it will never be "summed" when putting it on a visual. But don't do this for Months, as it will then sort 1, 10, 11, 12, 2

These formulae, as written, rely on the **YEMonth** query created in Recipe 60.110.x. If you don't have this query, or would like to skip it, just replace it with the numeric value of the final month in your fiscal year.

For more data patterns check out our online shop
© www.skillwave.training

Recipe: 60.330.2008

PeriodID Column Recipes for 445 / 454 / 544 Calendars

Step 1: Add a DayID column

Go to Add Column → Index Column → From 1

Rename [Index] to DayID

Step 2: Add other ID columns

For each item in the *PeriodID Formulae* table:

Go to Add Column → Custom Column

Enter the column names and formulae as per the table

Click OK

PeriodID Formulae

Column	Formula
YearID	`= Number.RoundUp( [DayID]/364 )`
QuarterID	`= Number.RoundUp( [DayID]/91 )`
MonthID	See Recipe 60.420.x
WeekID	`= Number.RoundUp( [DayID]/7 )`

These columns have two uses:
1) Creating Time Intelligent DAX Measures: Implement Rob Collie's GFITW pattern for custom calendar time intelligence! Click here to learn more!
2) Creating Fiscal Date Columns for reporting: Create columns like "End of Fiscal Year" or "Fiscal Month of Year"

My ID columns don't reset at period end...

You are correct, and this is by design. These ID columns are intended to drive Power Query logic and DAX calculations, not necessarily fiscal reporting.

I only want YearID, so do I need DayID?

These Recipes make use of DayID to make things simpler. If you don't want the DayID, then you can either modify the formulae or delete it when you have the columns you need.

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

Excel   Power BI

MonthID Column Recipe for 445 / 454 / 544 Calendars

To add the MonthID column:

Go to Add Column → Custom Column

  Name: MonthID

  Formula:

      Enter the appropriate formula based on your calendar type, as shown in the table on the right

Click OK

Calendar Type	MonthID Formula

4-4-5	```= Number.RoundDown([DayID]/91)*3+```
	``` (if Number.Mod([DayID],91)=0 then 0```
	``` else if Number.Mod([DayID],91)<= 28 then 1```
	``` else if Number.Mod([DayID],91)<= 56 then 2```
	``` else 3```
	```)```

4-5-4	```= Number.RoundDown([DayID]/91)*3+```
	``` (if Number.Mod([DayID],91)=0 then 0```
	``` else if Number.Mod([DayID],91)<= 28 then 1```
	``` else if Number.Mod([DayID],91)<= 63 then 2```
	``` else 3```
	```)```

5-4-4	```= Number.RoundDown([DayID]/91)*3+```
	``` (if Number.Mod([DayID],91)=0 then 0```
	``` else if Number.Mod([DayID],91)<= 35 then 1```
	``` else if Number.Mod([DayID],91)<= 63 then 2```
	``` else 3```
	```)```

Don't forget that this formula relies on the **_DayID_** column created in Recipe 60.410.x

Master Your Data in Excel & Power BI

Fiscal Period Columns for 445 / 454 / 544 Calendars

To add the period column(s):

Go to Add Column → Custom Column

Name:

Fiscal <Period>

Formula:

Enter the appropriate formula based on the column you need, as shown in the table on the right

Click OK

Set the column's data type

Fiscal...	Required Column(s)	Formula

Year	**[YearID]** **[Date]** **[DayID]**	`= [YearID] +`
		`Date.Year(`
		`Date.AddDays( [Date], - [DayID] + 1 )`
		`)`

Quarter	**[QuarterID]**	`= Number.Mod( [QuarterID] - 1, 4 ) + 1`
Month	**[MonthID]**	`= Number.Mod( [MonthID] - 1 , 12 ) + 1`
Week	**[WeekID]**	`= Number.Mod( [WeekID] - 1, 52 ) + 1`

If your Fiscal Year is off by one in the first year, just add or subtract 1 at the end of the formula.

Which data type should I choose?

Setting your Fiscal Year to text doesn't compromise the sort order and means that it will never be "summed" when putting it on a visual. But don't do this for Months, as it will then sort 1, 10, 11, 12, 2

These formulae rely on:
The **Start364** query created in Recipe 60.110.x and
The **PeriodID** columns from Recipe 60.410.x.

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

Excel   Power BI

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data in Excel & Power BI

Pattern Difficulty

PeriodID Column Recipes for 13 x 4-Week Calendars

Step 1: Add a DayID column

Go to Add Column → Index Column → From 1

Rename [Index] to DayID

Step 2: Add other ID columns

For each item in the *PeriodID Formulae* table:

Go to Add Column → Custom Column
Enter the column names and formulae as per the table
Click OK

> These columns have two uses:
> 1) Creating Time Intelligent DAX Measures: Implement Rob Collie's GFITW pattern for custom calendar time intelligence! Click here to learn more!
> 2) Creating Fiscal Date Columns: Create columns like "End of Fiscal Year" or "Fiscal Month of Year"

> The QuarterID column relies on the MonthID column. Don't forget to create it first!

> Changing 3, 6, 9 (in bold) to 4, 7, 10 would include the 13th month in the 1st quarter instead of the 4th quarter.

> **My ID columns don't reset at period end…**
>
> You are correct, and this is by design. These ID columns are intended to drive Power Query logic and DAX calculations, not necessarily fiscal reporting.

Column	PeriodID Formulae
WeekID	`= Number.RoundUp( [DayID]/7 )`
MonthID	`= Number.RoundUp( [DayID] / 28 )`

QuarterID	`= Number.RoundDown([MonthID]/13)*4 +`
	`  (if Number.Mod([MonthID],13)=0 then 0`
	`   else if Number.Mod([MonthID],13)<= `**3**` then 1`
	`   else if Number.Mod([MonthID],13)<= `**6**` then 2`
	`   else if Number.Mod([MonthID],13)<= `**9**` then 3`
	`   else 4)`

YearID	`= Number.RoundUp( [DayID]/364 )`

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

📹 Excel   📹 Power BI
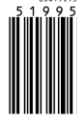
For more data patterns check out our online shop
© www.skillwave.training

Recipe: 60.510. 2008

ISBN 978-1-61547-246-8
US$19.95
51995>
9 781615 472468

Fiscal Period Columns for 13 x 4-Week Calendars

To add the period column(s):

Go to Add Column → Custom Column

  Name:

  Fiscal <Period>

  Formula:

  Enter the appropriate formula based on the column you need, as shown in the table on the right

Click OK

Set the column's data type

Fiscal…	Required Column(s)	Formula

Year	**[YearID]** **[Date]** **[DayID]**	`= [YearID] +`
		`Date.Year(`
		`Date.AddDays( [Date], - [DayID] + 1 )`
		`)`

Quarter	**[Fiscal Month]**	`= if [Fiscal Month] > `**`3`**` then 2`
		`   else if [Fiscal Month] > `**`6`**` then 3`
		`   else if [Fiscal Month] > `**`9`**` then 4`
		`   else 1`

Month	**[MonthID]**	`= Number.Mod( [MonthID] - 1 , 13 ) + 1`
Week	**[WeekID]**	`= Number.Mod( [WeekID] - 1, 52 ) + 1`

Which data type should I choose?

Setting your Fiscal Year to text doesn't compromise the sort order and means that it will never be "summed" when putting it on a visual. But don't do this for Months, as it will then sort 1, 10, 11, 12, 2

If your Fiscal Year is off by one in the first year, just add or subtract 1 at the end of the formula.

Changing 3, 6, 9 (in bold) to 4, 7, 10 would include the 13th month in the 1st quarter instead of the 4th quarter.

These formulae rely on:
The *Start364* query created in Recipe 60.110.x and
The *PeriodID* columns from Recipe 60.510.x.

Dimensional Modeling Subscribers:
Click icons to see this recipe in action!

Excel   Power BI

Recipe: 60.530.2008

Master Your Data in Excel & Power BI

Replace *null* with Current Date Recipe

Perform a simple text replacement

    Right click the desired date column → Replace Values

    Replace ***null*** with ***01-01-01***

Edit the formula in the Formula bar

    Select: `#date(2001, 1, 1)`

    Replace with: `Date.From(DateTime.LocalNow())`

> Need the current date AND time? Replace null with:
> `01-01-01 12:59`
> then replace
> `#datetime(2001,1,1,1,0,0)`
> in the formula with only
> `DateTime.LocalNow()`

	1²₃ CustID	ABC Name	ABC Address	From	To
1	1	Fred	123 Somewhere St	2019-01-01	null
2	2	Bob	456 Nowhere Close	2019-01-01	2019-02-15
3	3	John	4 The Win	2019-01-01	null
4	2	Bob	1 So Close	2019-02-16	null

```
= Table.ReplaceValue(#"Changed Type",null,#date(2001, 1, 1),Replacer.ReplaceValue,{"To"})
```

Replace *nulls* with current date

This recipe is very useful for preparing a slowly moving dimension table

	1²₃ D	ABC Name	ABC Address	From	To
1	1	Fred	123 Somewhere St	2019-01-01	2019-06-14
2	2	Bob	456 Nowhere Close	2019-01-01	2019-02-15
3	3	John	4 The Win	2019-01-01	2019-06-14
4	2	Bob	1 So Close	2019-02-16	2019-06-14

```
= Table.ReplaceValue(#"Changed Type",null,Date.From(DateTime.LocalNow()),Replacer.ReplaceValue,{"To"})
```

For more data patterns check out our online shop
© www.skillwave.training

Master Your Data

Purchase the full set of Power Query Recipes at skillwave.training/shop