Data Modeling with DAX

Data Modeling with DAX - Overview

The decision makers in all the organizations have identified the need for analyzing the historical data of their organization in specific, and of the industry in general. This is becoming crucial day-by-day in the present competitive world, to meet the ever-changing business challenges.

Big Data and Business Intelligence have become the buzzwords in the business world. Data sources have become huge and data formats have become variant. The need of the hour is to have simple-to-use tools to handle the ever-flowing vast data in less time to gain insight and make relevant decisions at the appropriate time.

Data analysts can no longer wait for the required data to be processed by the IT department. They require a handy tool that enables them to quickly comprehend the required data and make it available in a format that helps the decision makers take required action at the right time.

Microsoft Excel has a powerful tool called as Power Pivot that was available as an add-in in the prior versions of Excel and is built-in feature in Excel 2016. The database of Power Pivot, called the data model and the formula language that works on the data model, called DAX (**D**ata **A**nalysis **E**xpressions) enables an Excel user to perform tasks such as data modeling and analysis in no time.

In this tutorial, you will learn data modeling and analysis using DAX, based on the Power Pivot data model. A sample Profit and Analysis database is used for the illustrations throughout this tutorial.

Data Modeling and Analysis Concepts

The data that you obtain from different variety of sources, termed as raw data, needs processing before you can utilize it for analysis purposes. You will learn about these in the chapter − Data Modeling and Analysis Concepts.

Data Modeling and Analysis with Excel Power Pivot

Since the tool that you will be mastering in this tutorial is Excel Power Pivot, you need to know how the data modeling and analysis steps are carried out in Power Pivot. You will learn these at a broader level in the chapter - Data Modeling and Analysis with Excel Power Pivot.

As you proceed with the subsequent chapters, you will learn about the different facets of Power Pivot, DAX and DAX functions in data modeling and analysis.

By the end of the tutorial, you will be able to perform data modeling and analysis with DAX for any context at hand.

Data Modeling with DAX

Data Modeling with DAX - Concepts

Business Intelligence (BI) is gaining importance in several fields and organizations. Decision making and forecasting based on historical data have become crucial in the ever-growing competitive world. There is huge amount of data available both internally and externally from diversified sources for any type of data analysis.

However, the challenge is to extract the relevant data from the available big data as per the current requirements, and to store it in a way that is amicable for projecting different insights from the data. A data model thus obtained with the usage of key business terms is a valuable communication tool. The data model also needs to provide a quick way of generating reports on an as needed basis.

Data modeling for BI systems enables you to meet many of the data challenges.

Prerequisites for a Data Model for BI

A data model for BI should meet the requirements of the business for which data analysis is being done. Following are the minimum basics that any data model has to meet −

The data model needs to be Business Specific

A data model that is suitable for one line of business might not be suitable for a different line of business. Hence, the data model must be developed based on the specific business, the business terms used, the data types, and their relationships. It should be based on the objectives and the type of decisions made in the organization.

The data model needs to have built-in Intelligence

The data model should include built-in intelligence through metadata, hierarchies, and inheritances that facilitate efficient and effective Business Intelligence process. With this, you will be able to provide a common platform for different users, eliminating repetition of the process.

The data model needs to be Robust

The data model should precisely present the data specific to the business. It should enable effective disk and memory storage so as to facilitate quick processing and reporting.

The data model needs to be Scalable

The data model should be able to accommodate the changing business scenarios in a quick and efficient way. New data or new data types might have to be included. Data refreshes might have to be handled effectively.

Data Modeling with DAX

Data Modeling for BI

Data modeling for BI consists of the following steps −

- Shaping the data
- Loading the data
- Defining the relationships between the tables
- Defining data types
- Creating new data insights

Shaping the Data

The data required to build a data model can be from various sources and can be in different formats. You need to determine which portion of the data from each of these data sources is required for specific data analysis. This is called Shaping the Data.

For example, if you are retrieving the data of all the employees in an organization, you need to decide what details of each employee are relevant to the current context. In other words, you need to determine which columns of the employee table are required to be imported. This is because, the lesser the number of columns in a table in the data model, the faster will be the calculations on the table.

Loading the Data

You need to load the identified data – the data tables with the chosen columns in each of the tables.

Defining the Relationships Between Tables

Next, you need to define the logical relationships between the various tables that facilitate combining data from those tables, i.e. if you have a table – Products - containing data about the products and a table - Sales - with the various sales transactions of the products, by defining a relationship between the two tables, you can summarize the sales, product wise.

Defining Data Types

Identifying the appropriate data types for the data in the data model is crucial for the accuracy of calculations. For each column in each table that you have imported, you need to define the data type. For example, text data type, real number data type, integer data type, etc.

Creating New Data Insights

This is a crucial step in date modeling for BI. The data model that is built might have to be shared with several people who need to understand data trends and make the required decisions in a very short time. Hence, creating new data insights from the source data will be effective, avoiding rework on the analysis.

The new data insights can be in the form of metadata that can be easily understood and used by specific business people.

Data Analysis

Once the data model is ready, the data can be analyzed as per the requirement. Presenting the analysis results is also an important step because the decisions will be made based on the reports.

Data Modeling Using Excel Power Pivot

Microsoft Excel Power Pivot is an excellent tool for data modeling and analysis.

- Data model is the Power Pivot database.
- DAX is the formula language that can be used to create metadata with the data in the data model by means of DAX formulas.
- Power PivotTables in Excel created with the data and metadata in the data model enables you to analyze the data and present the results.

In this tutorial, you will learn data modeling with Power Pivot data model and DAX and data analysis with Power Pivot. If you are new to Power Pivot, please refer to the Excel Power Pivot tutorial.

You have learnt the data modeling process steps in the previous chapter - Data Modeling and Analysis Concepts. In this chapter, you will learn how to execute each of those steps with Power Pivot data model and DAX.

In the following sections, you will learn each of these process steps as applied to Power Pivot data model and how DAX is used.

Shaping the Data

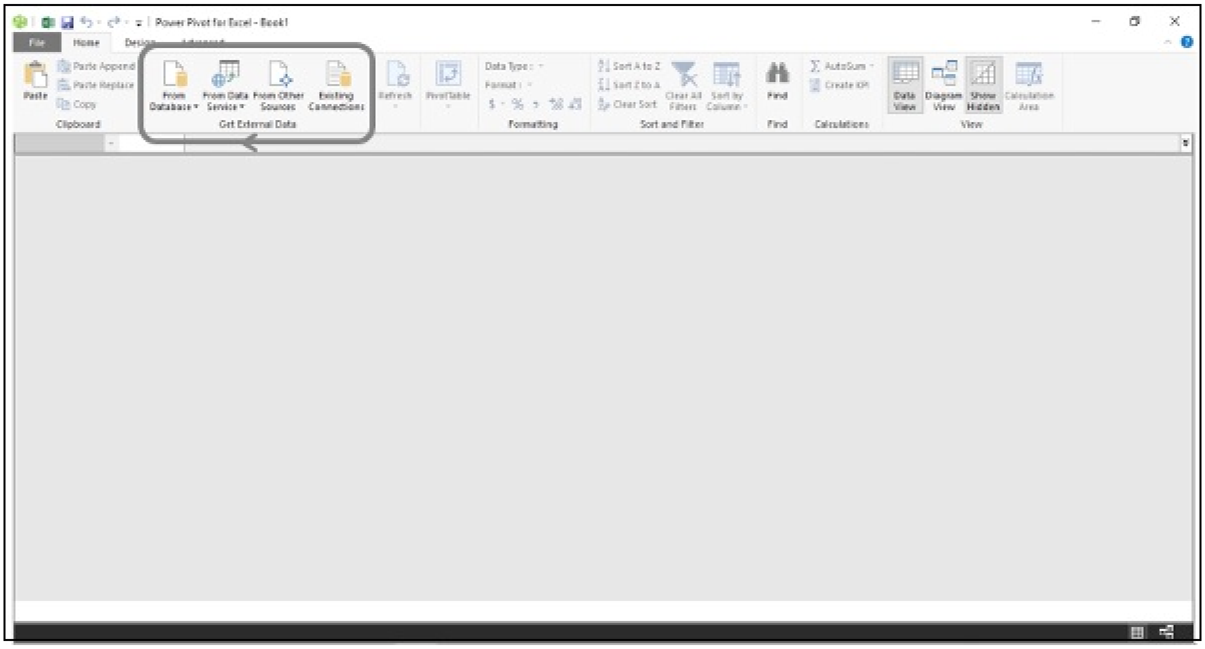
In Excel Power Pivot, you can import data from various types of data sources and while importing, you can view and choose the tables and columns that you want to import.

- Identify the data sources.
- Find the data source types. For example, database or data service or any other data source.
- Decide on what data is relevant in the current context.
- Decide on the appropriate data types for the data. In Power Pivot data model, you can have only one data type for the entire column in a table.
- Identify which of the tables are the fact tables and which are the dimensional tables.

- Decide on the relevant logical relationships between the tables.

Loading Data into the Data Model

You can load data into the data model with several options provided in the Power Pivot window on the Ribbon. You can find these options in the group, Get External Data.



You will learn how to load data from an Access database into the data model in the chapter – Loading Data into the Data Model.

For illustration purposes, an Access database with Profit and Loss data is used.

Defining Data Types in the Data Model

The next step in the data modeling process in Power Pivot is defining data types of the columns in the tables that are loaded into the data model.

You will learn how to define data types of the columns in the tables in the chapter – Defining Data Types in the Data Model.

Creating Relationships Between the Tables

The next step in the data modeling process in Power Pivot is creating relationships between the tables in the data model.

You will learn how to create relationships between the tables in the chapter – Extending the Data Model.

Data Modeling with DAX

Creating New Data Insights

In the data model, you can create metadata necessary for creating new data insights by −

- Creating Calculated Columns
- Creating Date Table
- Creating Measures

You can then analyze the data by creating dynamic Power PivotTables that are based on the columns in the tables and measures that appear as fields in the PivotTable Fields list.

Adding Calculated Columns

Calculated columns in a table are the columns that you add to a table by using DAX formulas.

You will learn how to add calculated columns in a table in the data model in the chapter - Extending the Data Model.

Creating Date Table

To use Time Intelligence Functions in DAX formulas to create metadata, you require a Date table. If you are new to Date tables, please refer to the chapter – Understanding Date Tables.

You will learn how to create a Date table in the data model in the chapter – Extending the Data Model.

Creating Measures

You can create various measures in the Data table by using the DAX functions and DAX formulas for different calculations as required for data analysis in the current context.

This is the crucial step of data modeling with DAX.

You will learn how to create the measures for various purposes of profit and loss analysis in the subsequent chapters.

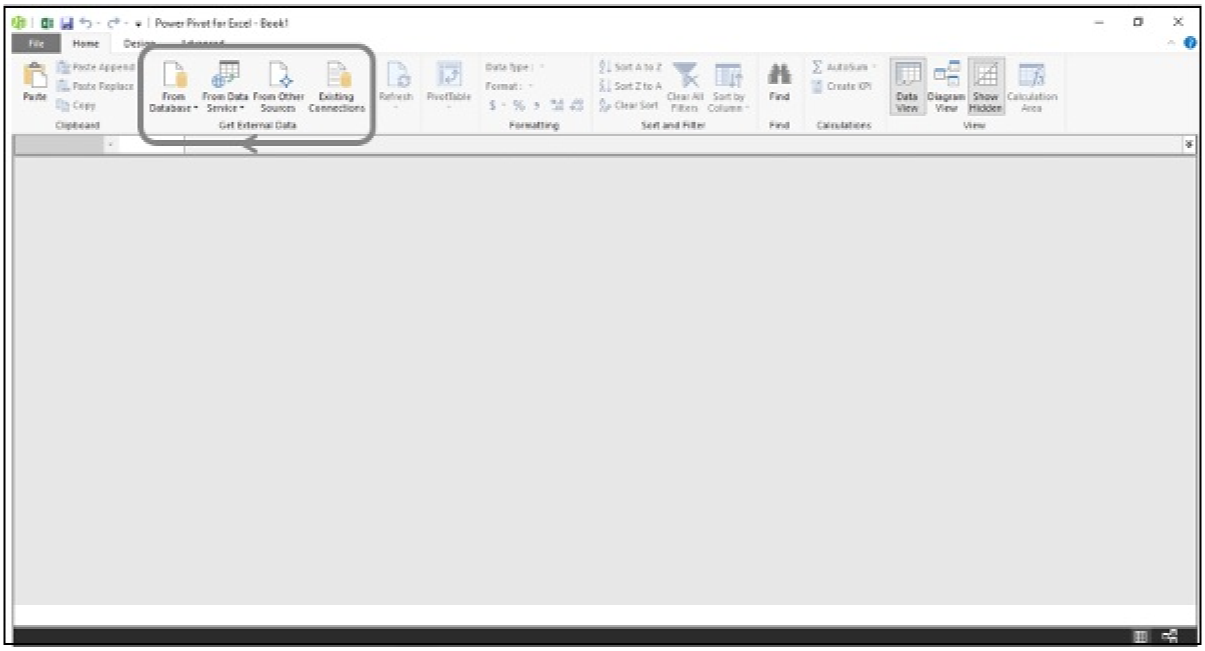
Analyzing Data with Power PivotTables

You can create Power PivotTables for each of the facets of profit and loss analysis. As you learn how to create measures using DAX in the subsequent chapters, you will also learn how to analyze data with these measures using Power PivotTables.

Data Modeling with DAX

Loading Data into the Data Model

You can load data from different types of data sources into the data model. For this, you can find various options in the Get External Data group on the Ribbon in the Power Pivot window.



As you can observe, you can load data from databases, or from data services or several other types of data sources.

When you load data from a data source into the data model, a connection will be established with the data source. This enables data refresh when the source data changes.

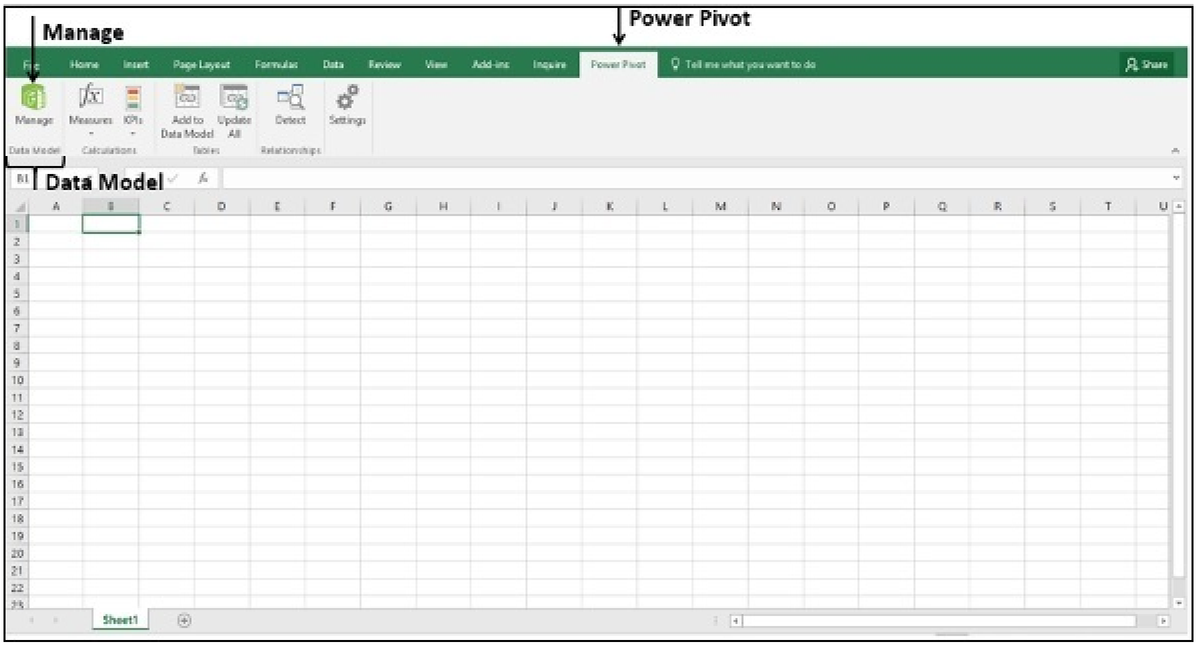
Initiating with a New Data Model

In this section, you will learn how to model the data for profit and loss analysis. The data for analysis is in a Microsoft Access database.

You can initiate a new data model as follows −

- Open a new Excel workbook
- Click the PowerPivot tab on the Ribbon
- Click Manage in the Data Model group

Data Modeling with DAX



The Power Pivot window appears. The window will be blank as you have not yet loaded any data.

Loading Data from Access Database into the Data Model

To load the data from the Access database, carry out the following steps –

- Click From Database in the Get External Data group on the Ribbon.
- Click From Access in the dropdown list.

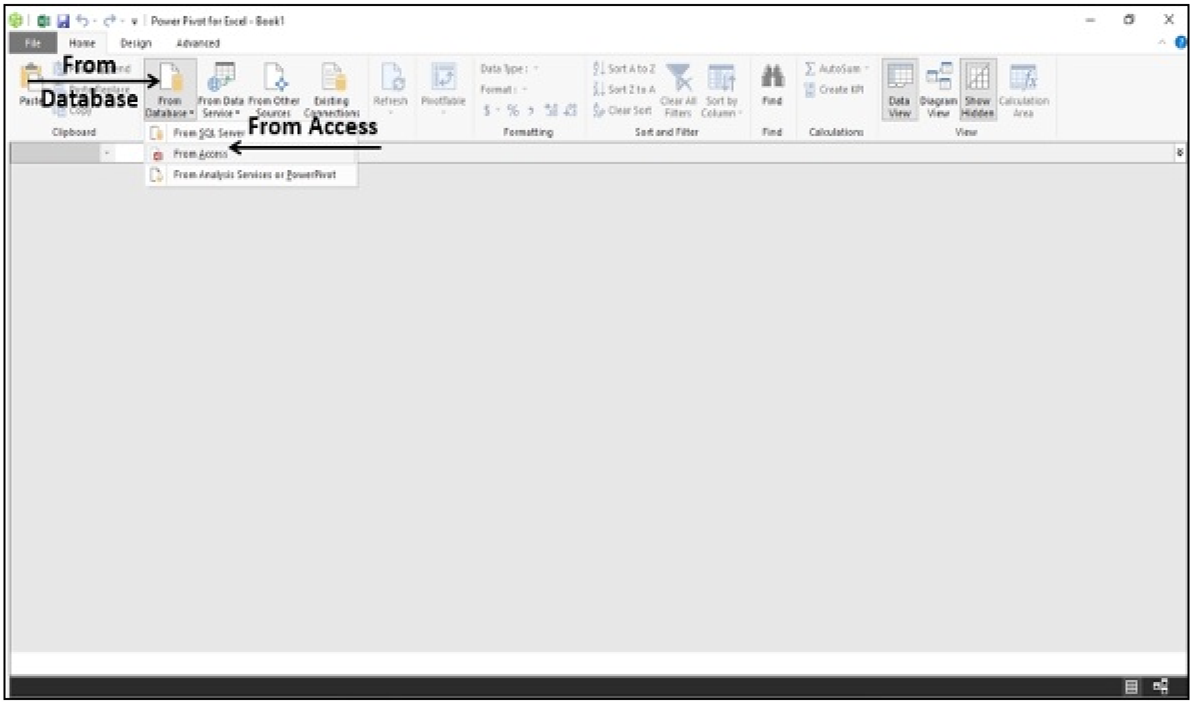


Table Import Wizard dialog box appears.

- Browse to the Access file.
- Give a friendly name for the connection.

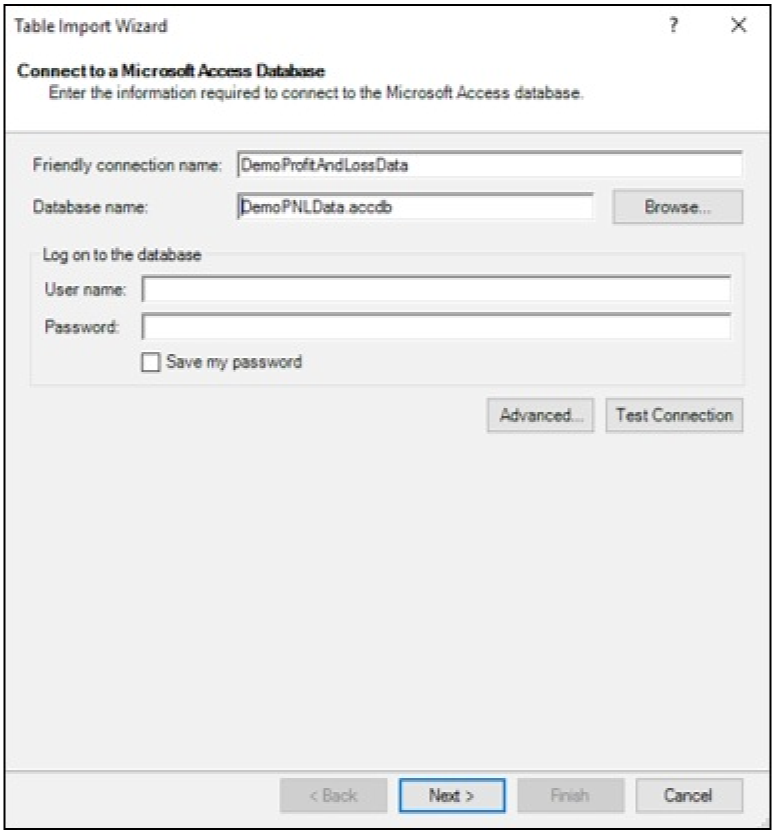


Table Import Wizard

Connect to a Microsoft Access Database

Enter the information required to connect to the Microsoft Access database.

Friendly connection name: DemoProfitAndLossData

Database name: DemoPNLData.accdb

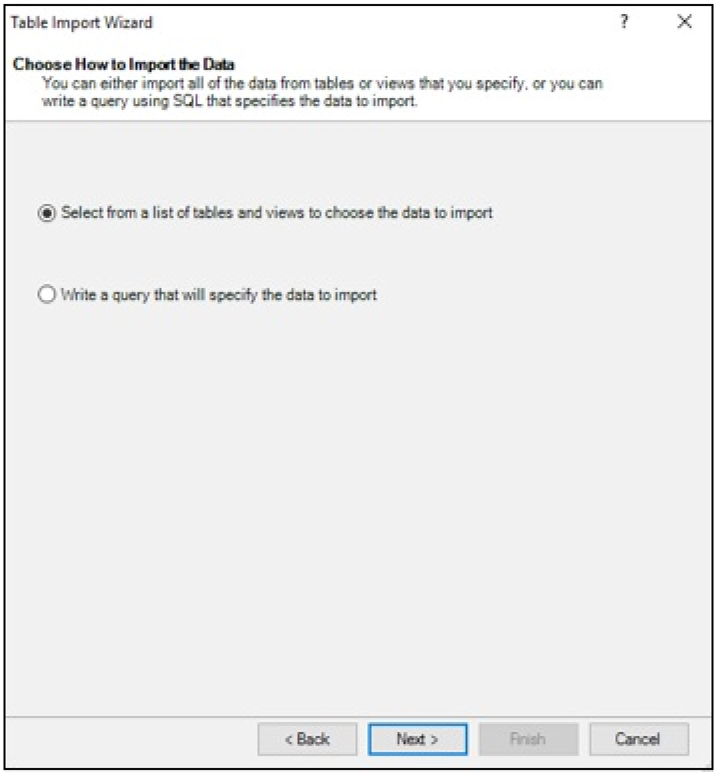
Log on to the database

User name:

Password:

□ Save my password

- Click the Next button. The next part of the Table Import Wizard appears.
- In the Table Import Wizard, select the option – Select from a list of tables and views to choose the data to import.



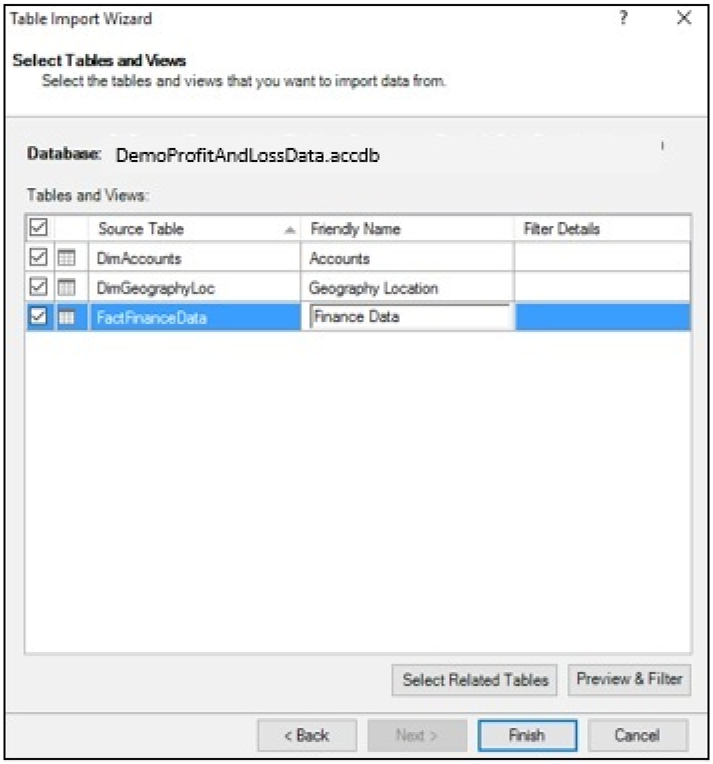
Choose How to Import the Data

You can either import all of the data from tables or views that you specify, or you can write a query using SQL that specifies the data to import.

● Select from a list of tables and views to choose the data to import

○ Write a query that will specify the data to import

- Click the Next button. The next part of the Table Import Wizard appears as shown in the following screenshot.
- Select all the tables.
- Give friendly names to the tables. This is necessary because these names appear in the Power PivotTables and hence should be understood by everyone.



Select Tables and Views

Select the tables and views that you want to import data from.

Database: DemoProfitAndLossData.accdb

Tables and Views:

☑		Source Table	Friendly Name	Filter Details
☑	▦	DimAccounts	Accounts	
☑	▦	DimGeographyLoc	Geography Location	
☑	▦	FactFinanceData	Finance Data	

Choosing the Columns in the Tables

You might not require all the columns in the selected tables for the current analysis. Hence, you need to select only those columns that you selected while shaping the data.

- Click the Preview & Filter button. The next part of the Table Import Wizard -Preview of the selected table - appears.

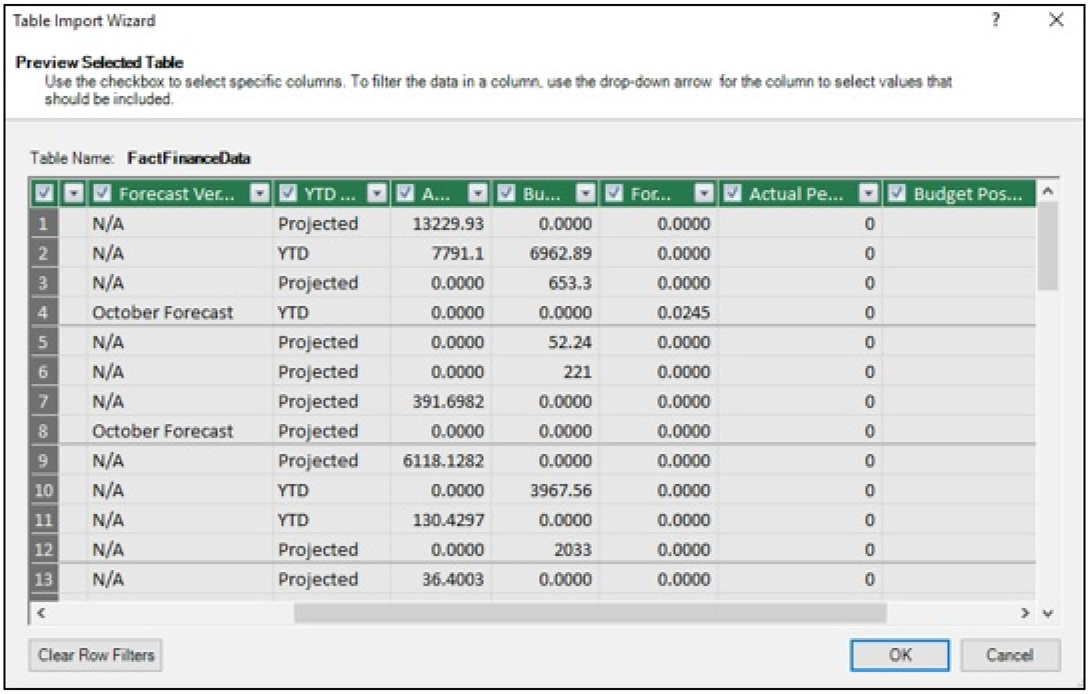


Table Import Wizard

Preview Selected Table

Use the checkbox to select specific columns. To filter the data in a column, use the drop-down arrow for the column to select values that should be included.

Table Name: **FactFinanceData**

☑	▾	☑ Forecast Ver... ▾	☑ YTD ... ▾	☑ A... ▾	☑ Bu... ▾	☑ For... ▾	☑ Actual Pe... ▾	☑ Budget Pos...
1		N/A	Projected	13229.93	0.0000	0.0000	0	
2		N/A	YTD	7791.1	6962.89	0.0000	0	
3		N/A	Projected	0.0000	653.3	0.0000	0	
4		October Forecast	YTD	0.0000	0.0000	0.0245	0	
5		N/A	Projected	0.0000	52.24	0.0000	0	
6		N/A	Projected	0.0000	221	0.0000	0	
7		N/A	Projected	391.6982	0.0000	0.0000	0	
8		October Forecast	Projected	0.0000	0.0000	0.0000	0	
9		N/A	Projected	6118.1282	0.0000	0.0000	0	
10		N/A	YTD	0.0000	3967.56	0.0000	0	
11		N/A	YTD	130.4297	0.0000	0.0000	0	
12		N/A	Projected	0.0000	2033	0.0000	0	
13		N/A	Projected	36.4003	0.0000	0.0000	0	

- As seen in the above screenshot, the column headers have check boxes. Select the columns you want to import in the selected table.
- Click OK. Repeat the same for the other tables.

Importing Data into the Data Model

You are at the last stage of loading data into the data model. Click the Finish button in the Table Import Wizard. The next part of the Table Import Wizard appears.

The importing status will be displayed. The status finally displays Success when data loading is complete.

Data Modeling with DAX

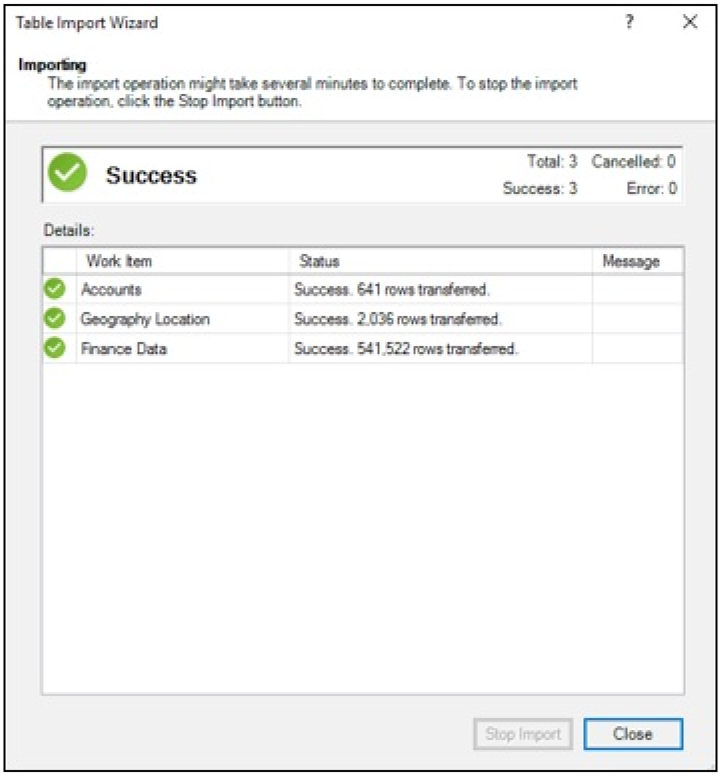


Table Import Wizard

Importing
The import operation might take several minutes to complete. To stop the import operation, click the Stop Import button.

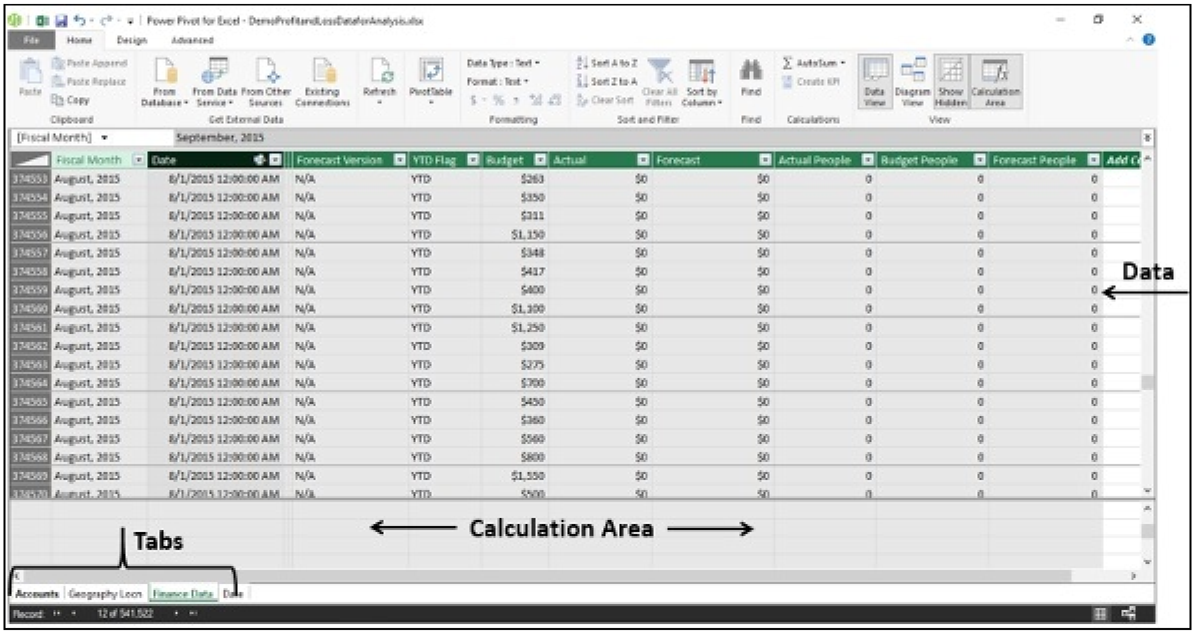
Success — Total: 3 Cancelled: 0 Success: 3 Error: 0

Details:

Work Item	Status	Message
Accounts	Success. 641 rows transferred.	
Geography Location	Success. 2,036 rows transferred.	
Finance Data	Success. 541,522 rows transferred.	

Viewing the Data in the Data Model

The imported tables appear in the Power Pivot window. This is the view of the data model



Power Pivot for Excel - DemoProfitandLossDataforAnalysis.xlsx

[Fiscal Month] September, 2015

	Fiscal Month	Date	Forecast Version	YTD Flag	Budget	Actual	Forecast	Actual People	Budget People	Forecast People	Add C
374553	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$263	$0	$0	0	0	0	
374554	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$350	$0	$0	0	0	0	
374555	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$311	$0	$0	0	0	0	
374556	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$1,150	$0	$0	0	0	0	
374557	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$348	$0	$0	0	0	0	
374558	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$417	$0	$0	0	0	0	
374559	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$400	$0	$0	0	0	0	
374560	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$1,100	$0	$0	0	0	0	
374561	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$1,250	$0	$0	0	0	0	
374562	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$309	$0	$0	0	0	0	
374563	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$275	$0	$0	0	0	0	
374564	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$700	$0	$0	0	0	0	
374565	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$450	$0	$0	0	0	0	
374566	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$360	$0	$0	0	0	0	
374567	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$560	$0	$0	0	0	0	
374568	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$800	$0	$0	0	0	0	
374569	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$1,550	$0	$0	0	0	0	
374570	August, 2015	8/1/2015 12:00:00 AM	N/A	YTD	$500	$0	$0	0	0	0	

Record: 12 of 541,522

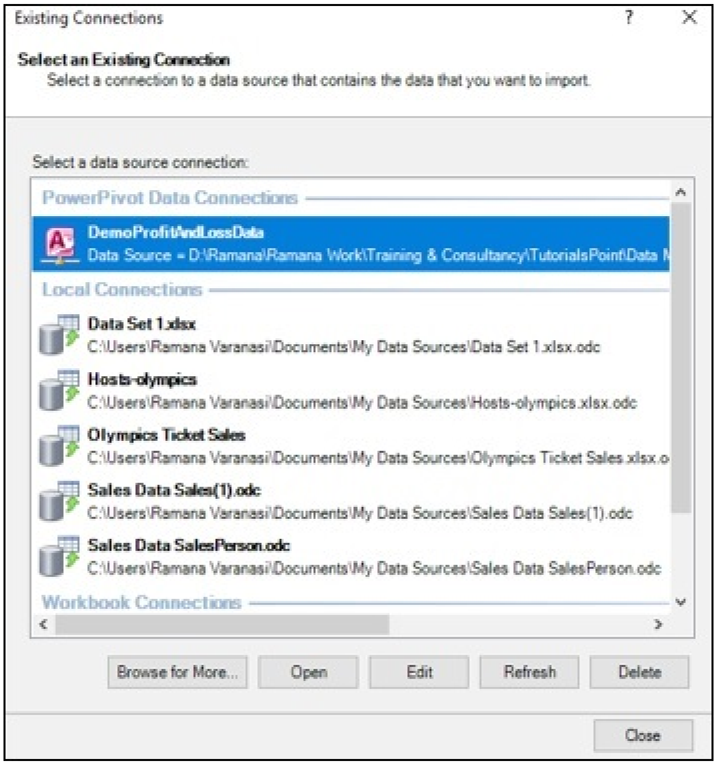
You can observe the following –

Data Modeling with DAX

- Each of the tables appear in a separate tab.
- The tab names are the respective table names.
- The area below the data is for the calculations.

Viewing the Connection Name

Click the Existing Connections in the Get External Data group. Existing Connections dialog box appears as shown in the following screenshot.


Existing Connections

Select an Existing Connection
Select a connection to a data source that contains the data that you want to import.

Select a data source connection:

PowerPivot Data Connections

DemoProfitAndLossData
Data Source = D:\Ramana\Ramana Work\Training & Consultancy\TutorialsPoint\Data N

Local Connections

Data Set 1.xlsx
C:\Users\Ramana Varanasi\Documents\My Data Sources\Data Set 1.xlsx.odc

Hosts-olympics
C:\Users\Ramana Varanasi\Documents\My Data Sources\Hosts-olympics.xlsx.odc

Olympics Ticket Sales
C:\Users\Ramana Varanasi\Documents\My Data Sources\Olympics Ticket Sales.xlsx.o

Sales Data Sales(1).odc
C:\Users\Ramana Varanasi\Documents\My Data Sources\Sales Data Sales(1).odc

Sales Data SalesPerson.odc
C:\Users\Ramana Varanasi\Documents\My Data Sources\Sales Data SalesPerson.odc

Workbook Connections

As seen in the above screenshot, the connection name given appears under PowerPivot Data Connections.

Defining Data Types in the Data Model

In the Power Pivot data model, the entire data in a column must be of the same data type. To accomplish accurate calculations, you need to ensure that the data type of each column in each table in the data model is as per requirement.

Tables in the Data Model

In the data model created in the previous chapter, there are 3 tables −

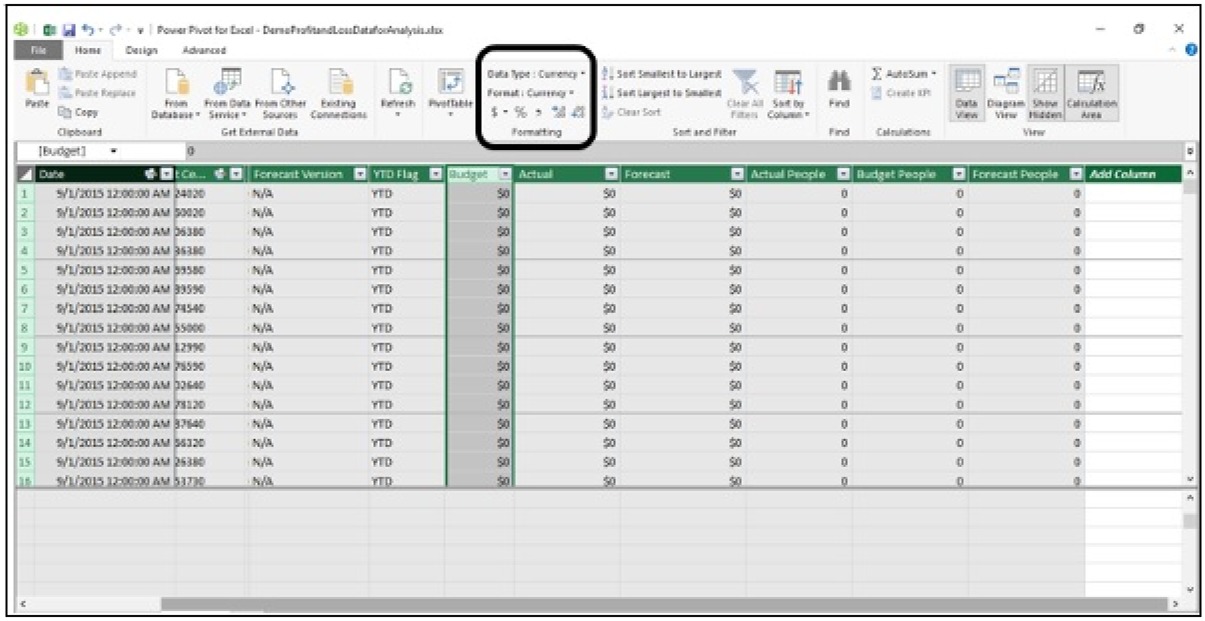
- Accounts

- Geography Locn
- Finance Data

Ensuring Appropriate Data Types

To ensure that the columns in the tables are as required, you need to check their data types in the Power Pivot window.

- Click a column in a table.
- Note the data type of the column as displayed on the Ribbon in the Formatting group.



Data Type : Currency

Format : Currency

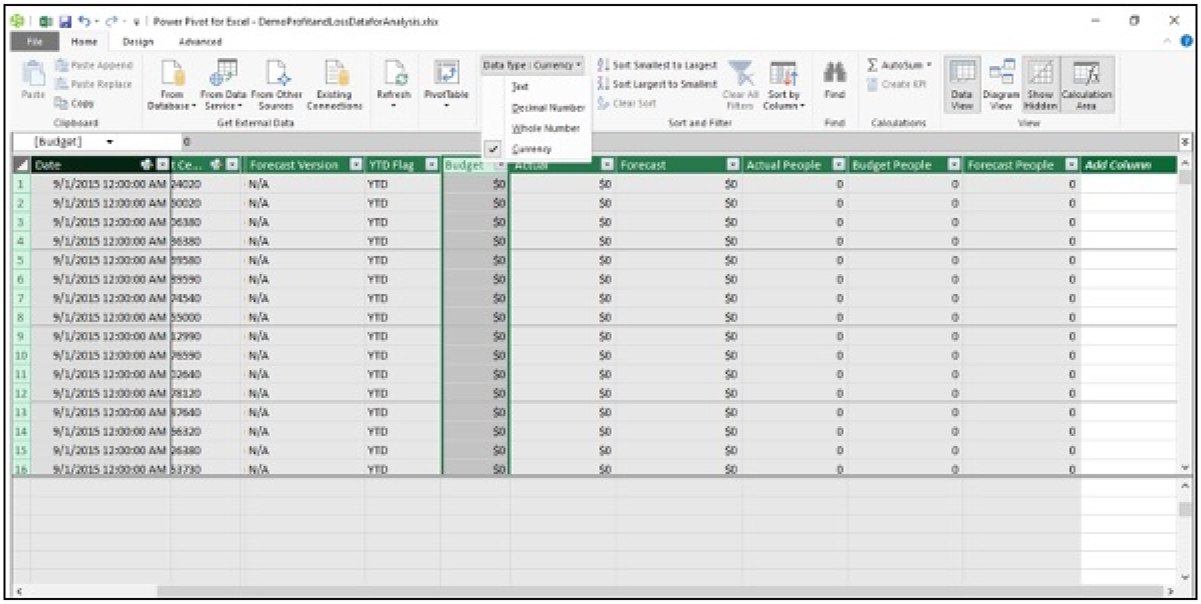
[Budget] 0

	Date	t Co...	Forecast Version	YTD Flag	Budget	Actual	Forecast	Actual People	Budget People	Forecast People	Add Column
1	9/1/2015 12:00:00 AM	24020	N/A	YTD	$0	$0	$0	0	0	0	
2	9/1/2015 12:00:00 AM	50020	N/A	YTD	$0	$0	$0	0	0	0	
3	9/1/2015 12:00:00 AM	06380	N/A	YTD	$0	$0	$0	0	0	0	
4	9/1/2015 12:00:00 AM	36380	N/A	YTD	$0	$0	$0	0	0	0	
5	9/1/2015 12:00:00 AM	39580	N/A	YTD	$0	$0	$0	0	0	0	
6	9/1/2015 12:00:00 AM	39590	N/A	YTD	$0	$0	$0	0	0	0	
7	9/1/2015 12:00:00 AM	74540	N/A	YTD	$0	$0	$0	0	0	0	
8	9/1/2015 12:00:00 AM	55000	N/A	YTD	$0	$0	$0	0	0	0	
9	9/1/2015 12:00:00 AM	12990	N/A	YTD	$0	$0	$0	0	0	0	
10	9/1/2015 12:00:00 AM	76590	N/A	YTD	$0	$0	$0	0	0	0	
11	9/1/2015 12:00:00 AM	02640	N/A	YTD	$0	$0	$0	0	0	0	
12	9/1/2015 12:00:00 AM	78120	N/A	YTD	$0	$0	$0	0	0	0	
13	9/1/2015 12:00:00 AM	37640	N/A	YTD	$0	$0	$0	0	0	0	
14	9/1/2015 12:00:00 AM	06120	N/A	YTD	$0	$0	$0	0	0	0	
15	9/1/2015 12:00:00 AM	26380	N/A	YTD	$0	$0	$0	0	0	0	
16	9/1/2015 12:00:00 AM	53730	N/A	YTD	$0	$0	$0	0	0	0	

If the data type of the selected column is not appropriate, change the data type as follows.

- Click the down arrow next to the data type in the Formatting group.
- Click the appropriate data type in the dropdown list.
- Repeat for every column in all the tables in the data model.

Data Modeling with DAX



Power Pivot for Excel - DemoProfitandLossDataforAnalysis.xlsx

[Budget] 0

	Date	t Ce...	Forecast Version	YTD Flag	Budget	Actual	Forecast	Actual People	Budget People	Forecast People	Add Column
1	9/1/2015 12:00:00 AM	24020	N/A	YTD	$0	$0	$0	0	0	0	
2	9/1/2015 12:00:00 AM	00020	N/A	YTD	$0	$0	$0	0	0	0	
3	9/1/2015 12:00:00 AM	06380	N/A	YTD	$0	$0	$0	0	0	0	
4	9/1/2015 12:00:00 AM	36380	N/A	YTD	$0	$0	$0	0	0	0	
5	9/1/2015 12:00:00 AM	99580	N/A	YTD	$0	$0	$0	0	0	0	
6	9/1/2015 12:00:00 AM	89590	N/A	YTD	$0	$0	$0	0	0	0	
7	9/1/2015 12:00:00 AM	24540	N/A	YTD	$0	$0	$0	0	0	0	
8	9/1/2015 12:00:00 AM	55000	N/A	YTD	$0	$0	$0	0	0	0	
9	9/1/2015 12:00:00 AM	12990	N/A	YTD	$0	$0	$0	0	0	0	
10	9/1/2015 12:00:00 AM	76590	N/A	YTD	$0	$0	$0	0	0	0	
11	9/1/2015 12:00:00 AM	02640	N/A	YTD	$0	$0	$0	0	0	0	
12	9/1/2015 12:00:00 AM	78120	N/A	YTD	$0	$0	$0	0	0	0	
13	9/1/2015 12:00:00 AM	87640	N/A	YTD	$0	$0	$0	0	0	0	
14	9/1/2015 12:00:00 AM	56320	N/A	YTD	$0	$0	$0	0	0	0	
15	9/1/2015 12:00:00 AM	26380	N/A	YTD	$0	$0	$0	0	0	0	
16	9/1/2015 12:00:00 AM	53730	N/A	YTD	$0	$0	$0	0	0	0	

Columns in the Accounts Table

In the Accounts table, you have the following columns −

Sr.No	Column & Description
1	**Account** Contains one account number for each row. The column has unique values and is used in defining the relationship with the Finance Data table.
2	**Class** The class associated with each account. Example - Expenses, Net Revenue, etc.
3	**Sub Class** Describes the type of expense or revenue. Example – People.

All the columns in the Accounts table are of descriptive in nature and hence are of Text data type.

Columns in the Geography Locn Table

The Geography Locn table contains data about each Profit Center.

The column Profit Center contains one profit center identity for each row. This column has unique values and is used in defining the relationship with the Finance Data table.

Columns in the Finance Data Table

Data Modeling with DAX

In the Finance Data table, you have the following columns −

Column	Description	Data type
Fiscal Month	Month and Year	Text
Profit Center	Profit Center identity	Text
Account	Account number. Each account can have multiple Profit Centers.	Text
Budget	Monthly budget amounts for each Profit Center.	Currency
Actual	Monthly actual amounts for each Profit Center.	Currency
Forecast	Monthly forecast amounts for each profit center.	Currency
Actual People	Month end actual number of employees for each Profit Center of each people Account.	Whole Number
Budget People	Month end budget number of employees for each Profit Center of each people Account.	Whole Number
Forecast People	Month end forecast number of employees for each Profit Center of each people Account.	Whole Number

Types of Tables in the Data Model

Both Accounts and Geography Locn tables are the dimensional tables, also called as **lookup tables**.

Finance Data table is the fact table, also known as the data table. Finance Data table contains the data required for the profit and analysis calculations. You will also create metadata in the form of measures and calculated columns in this Finance Data table, so as to model the data for various types of profit and loss calculations, as you proceed with this tutorial.

Understanding Data Tables

Data Analysis involves browsing data over time and making calculations across time periods. For example, you might have to compare the current year's profits with the previous year's profits. Similarly, you might have to forecast the growth and profits in the coming years. For these, you need to use grouping and aggregations over a period of time.

DAX provides several Time Intelligence functions that help you perform most of such calculations. However, these DAX functions require a Date table for usage with the other tables in the data model.

You can either import a Date table along with other data from a data source or you can create a Date table by yourself in the data model.

Data Modeling with DAX

In this chapter, you will understand different aspects of Date tables. If you are conversant with Date tables in the Power Pivot data model, you can skip this chapter and proceed with the subsequent chapters. Otherwise, you can understand the Date tables in the Power Pivot data model.

What is a Date Table?

A Date Table is a table in a data model, with at least one column of contiguous dates across a required duration. It can have additional columns representing different time periods. However, what is necessary is the column of contiguous dates, as required by the DAX Time Intelligence functions.

For example,

- A Date table can have columns such as Date, Fiscal Month, Fiscal Quarter, and Fiscal Year.
- A Date table can have columns such as Date, Month, Quarter, and Year.

Date Table with Contiguous Dates

Suppose you are required to make calculations in the range of a calendar year. Then, the Date table must have at least one column with a contiguous set of dates, including all the dates in that specific calendar year.

For example, suppose the data you want to browse has dates from April 1st, 2014 through November 30th, 2016.

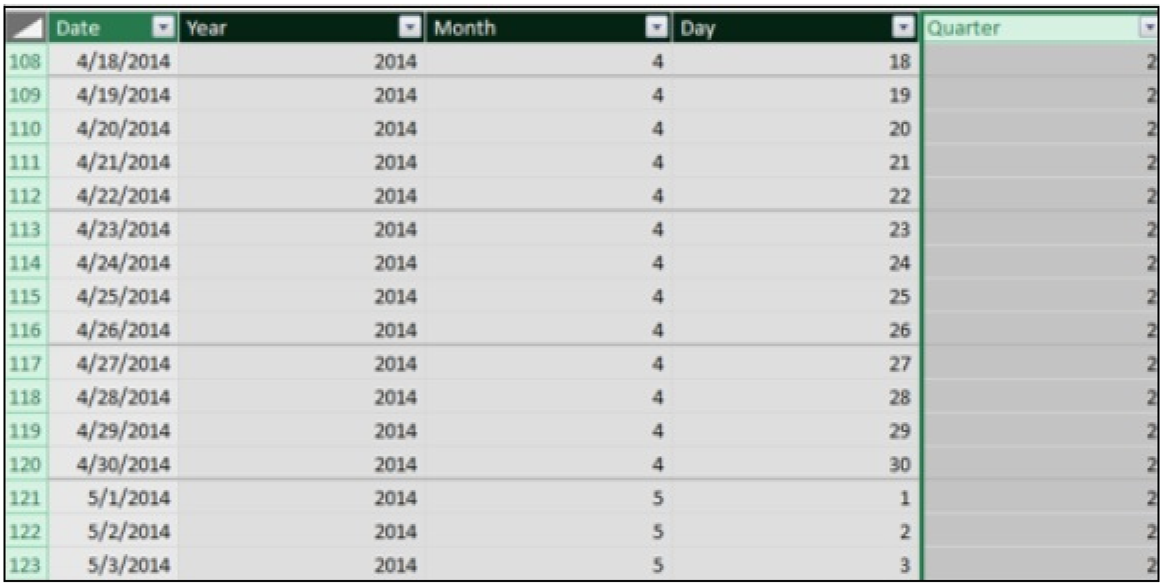
- If you have to report on a calendar year, you need a Date table with a column – Date, which contains all the dates from January 1st, 2014 to December 31st, 2016 in a sequence.
- If you have to report on a fiscal year, and your fiscal year end is 30th June, you need a Date table with a column – Date, which contains all the dates from July 1st, 2013 to June 30th, 2017 in a sequence.
- If you have to report on both calendar and fiscal years, then you can have a single Date table spanning the required range of dates.

Your Date table must contain all of the days for the range of every year in the given duration. Thus, you will get contiguous dates within that period of time.

If you regularly refresh your data with new data, you will have the end date extended by a year or two, so that you do not have to update your Date table often.

A Date table looks like the following screenshot.

Data Modeling with DAX



	Date	Year	Month	Day	Quarter
108	4/18/2014	2014	4	18	2
109	4/19/2014	2014	4	19	2
110	4/20/2014	2014	4	20	2
111	4/21/2014	2014	4	21	2
112	4/22/2014	2014	4	22	2
113	4/23/2014	2014	4	23	2
114	4/24/2014	2014	4	24	2
115	4/25/2014	2014	4	25	2
116	4/26/2014	2014	4	26	2
117	4/27/2014	2014	4	27	2
118	4/28/2014	2014	4	28	2
119	4/29/2014	2014	4	29	2
120	4/30/2014	2014	4	30	2
121	5/1/2014	2014	5	1	2
122	5/2/2014	2014	5	2	2
123	5/3/2014	2014	5	3	2

Adding a Date Table to the Data Model

You can add a Date table to the data model in any of the following ways −

- Importing from a relational database, or any other data source.
- Creating a Date table in Excel and then copying or linking to a new table in Power Pivot.
- Importing from Microsoft Azure Marketplace.

Creating a Date Table in Excel and Copying to the Data Model

Creating a Date table in Excel and copying to the data model is the easiest and most flexible way of creating a Data table in the data model.

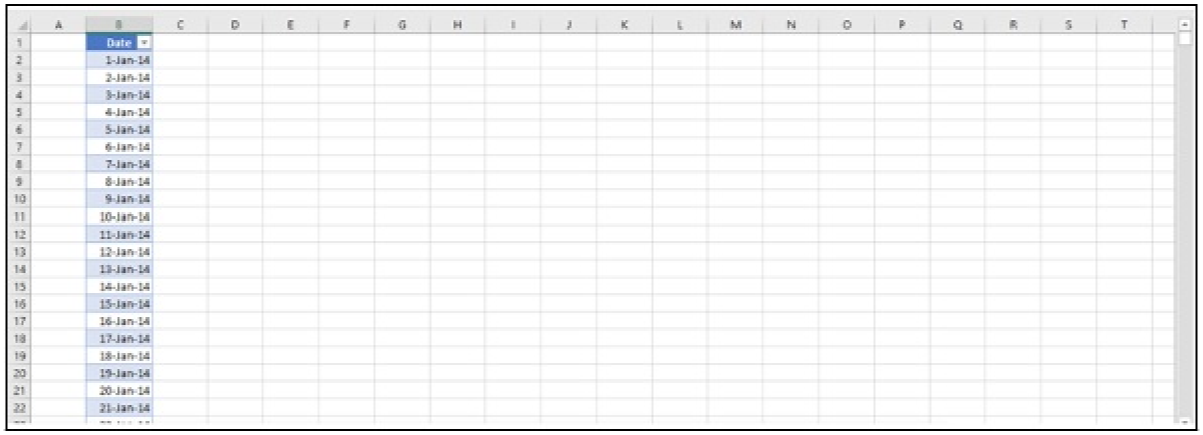
- Open a new worksheet in Excel.
- Type – Date in the first row of a column.
- Type the first date in the date range that you want to create in the second row in the same column.
- Select the cell, click the fill handle and drag it down to create a column of contiguous dates in the required date range.

For example, type 1/1/2014, click the fill handle and drag down to fill the contiguous dates up to 31/12/2016.

- Click the Date column.
- Click the INSERT tab on the Ribbon.
- Click Table.
- Verify the table range.

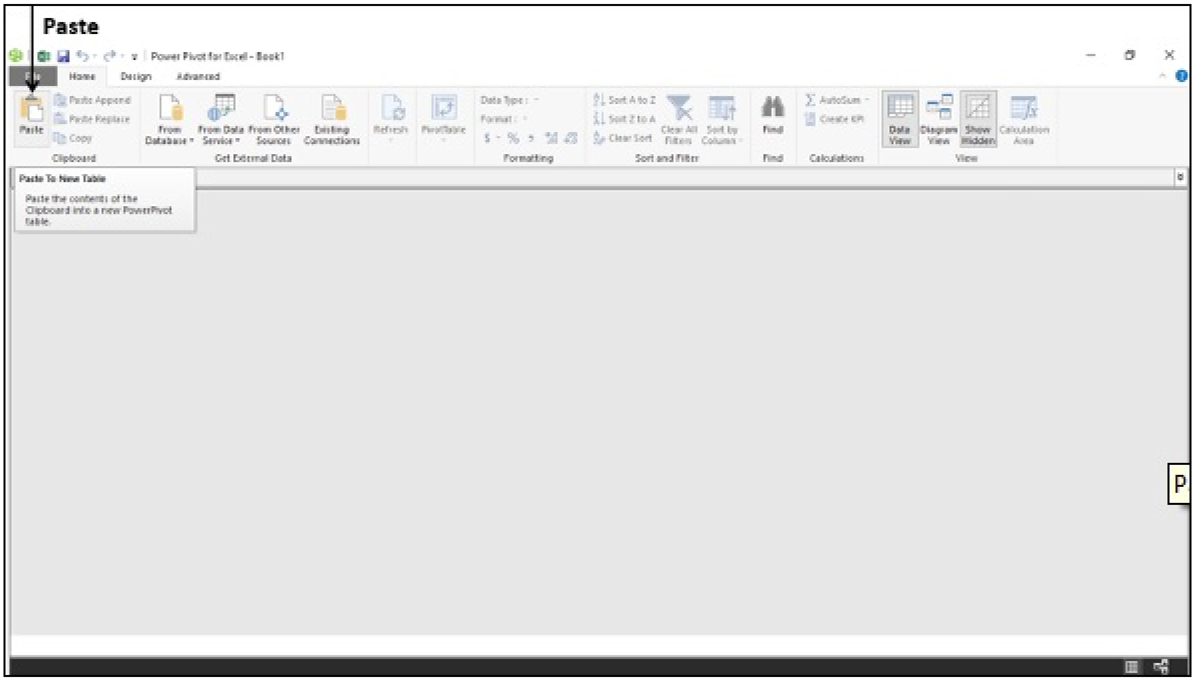
- Click OK.

The table of a single column of dates is ready in Excel.



Date
1-Jan-14
2-Jan-14
3-Jan-14
4-Jan-14
5-Jan-14
6-Jan-14
7-Jan-14
8-Jan-14
9-Jan-14
10-Jan-14
11-Jan-14
12-Jan-14
13-Jan-14
14-Jan-14
15-Jan-14
16-Jan-14
17-Jan-14
18-Jan-14
19-Jan-14
20-Jan-14
21-Jan-14

- Select the table.
- Click Copy on the Ribbon.
- Click the Power Pivot window.
- Click Paste on the Ribbon.



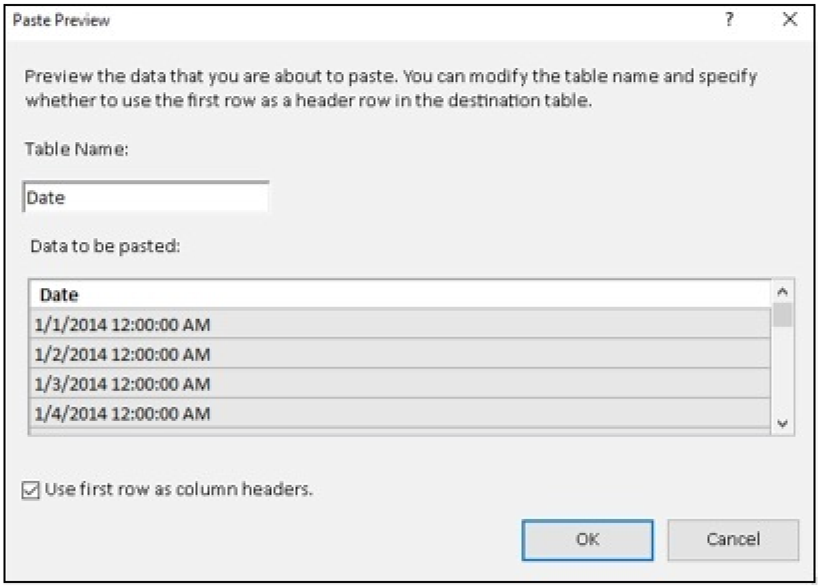
Paste To New Table

Paste the contents of the Clipboard into a new PowerPivot table.

This will add the contents of the clipboard to a new table in the data model. Hence, you can use the same method to create a Date table in an existing data model also.

Paste preview dialog box appears as shown in the following screenshot.

- Type Date in the Table Name box.
- Preview the data.



Preview the data that you are about to paste. You can modify the table name and specify whether to use the first row as a header row in the destination table.

Table Name: Date

Data to be pasted:

Date
1/1/2014 12:00:00 AM
1/2/2014 12:00:00 AM
1/3/2014 12:00:00 AM
1/4/2014 12:00:00 AM

☑ Use first row as column headers.

- Check the box – Use first row as column headers.
- Click OK.

This copies the contents of the clipboard to a new table in the data model.

Now, you have a Date table in the data model with a single column of contiguous dates. The header of the column is Date as you had given in the Excel table.

Adding New Date Columns to the Date Table

Next, you can add calculated columns to the Date table as per the requirement for your calculations.

For example, you can add columns – Day, Month, Year, and Quarter as follows −

- Day

  =DAY('Date'[Date])

- Month

  =MONTH('Date'[Date])

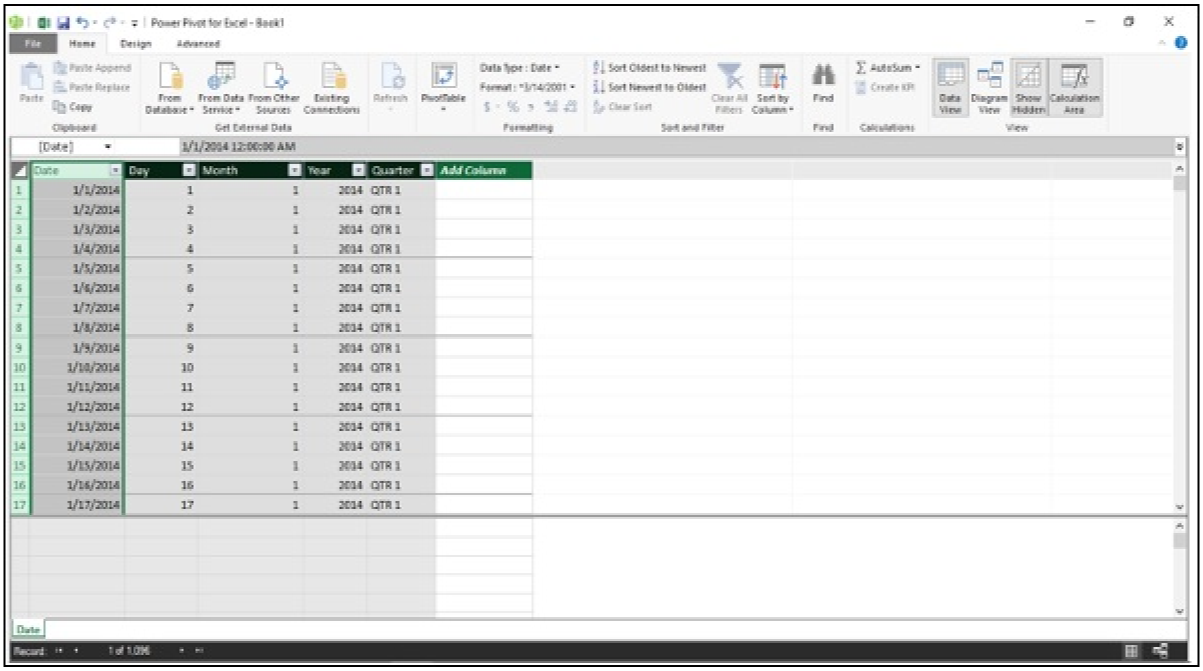
- Year

  =YEAR('Date'[Date])

- Quarter

  =CONCATENATE ("QTR ", INT (('Date'[Month]+2)/3))

The resulting Date table in the data model looks like the following screenshot.



[Date] 1/1/2014 12:00:00 AM

	Date	Day	Month	Year	Quarter	Add Column
1	1/1/2014	1	1	2014	QTR 1	
2	1/2/2014	2	1	2014	QTR 1	
3	1/3/2014	3	1	2014	QTR 1	
4	1/4/2014	4	1	2014	QTR 1	
5	1/5/2014	5	1	2014	QTR 1	
6	1/6/2014	6	1	2014	QTR 1	
7	1/7/2014	7	1	2014	QTR 1	
8	1/8/2014	8	1	2014	QTR 1	
9	1/9/2014	9	1	2014	QTR 1	
10	1/10/2014	10	1	2014	QTR 1	
11	1/11/2014	11	1	2014	QTR 1	
12	1/12/2014	12	1	2014	QTR 1	
13	1/13/2014	13	1	2014	QTR 1	
14	1/14/2014	14	1	2014	QTR 1	
15	1/15/2014	15	1	2014	QTR 1	
16	1/16/2014	16	1	2014	QTR 1	
17	1/17/2014	17	1	2014	QTR 1	

Thus, you can add any number of calculated columns to the Date table. What is important and is required is that the Date table must have a column of contiguous dates that spans the duration of time over which you perform calculations.

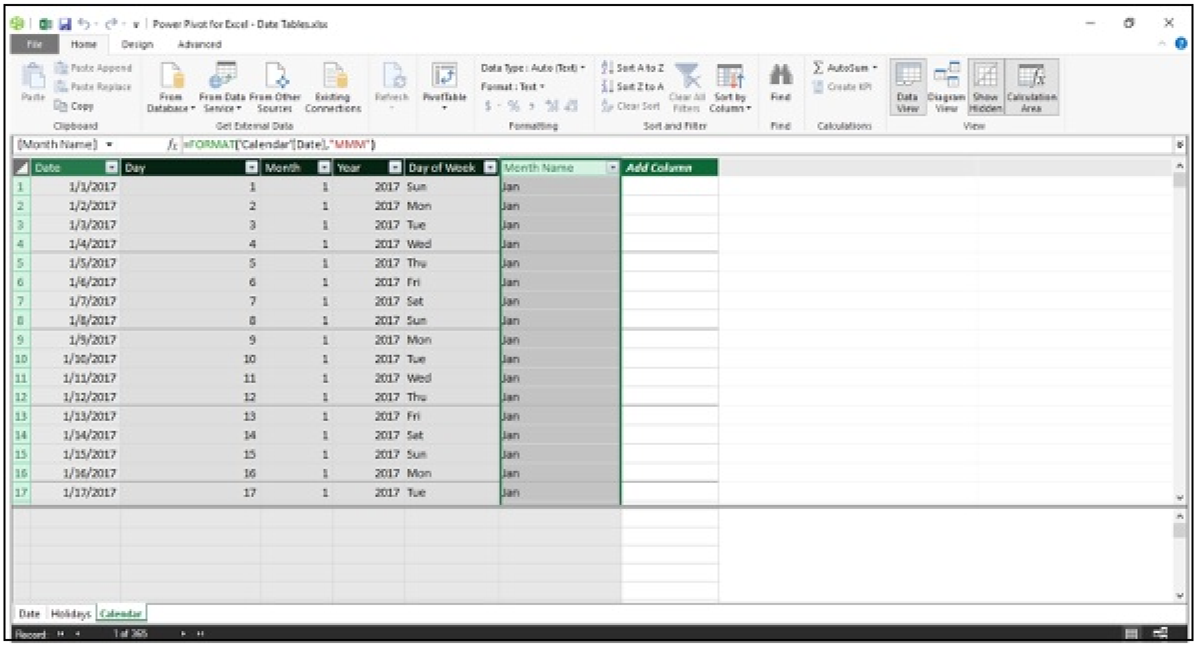
Creating a Date Table for a Calendar Year

A calendar year typically includes the dates from 1st January to 31st December of a year and also includes the holidays marked for that particular year. When you perform calculations, you might have to take into account only the working days, excluding weekends and holidays.

Suppose, you want to create a Date table for the calendar year 2017.

- Create an Excel table with a column Date, consisting of contiguous dates from 1st January 2017 to 31st December 2017. (Refer to the previous section to know how to do this.)
- Copy the Excel table and paste it into a new table in the data model. (Refer to the previous section to know how to do this.)
- Name the table as Calendar.
- Add the following calculated columns −
  - Day =DAY('Calendar'[Date])
  - Month =MONTH('Calendar'[Date])
  - Year =YEAR('Calendar'[Date])

- Day of Week =FORMAT('Calendar'[Date],"DDD")
- Month Name =FORMAT('Calendar'[Date],"MMM")



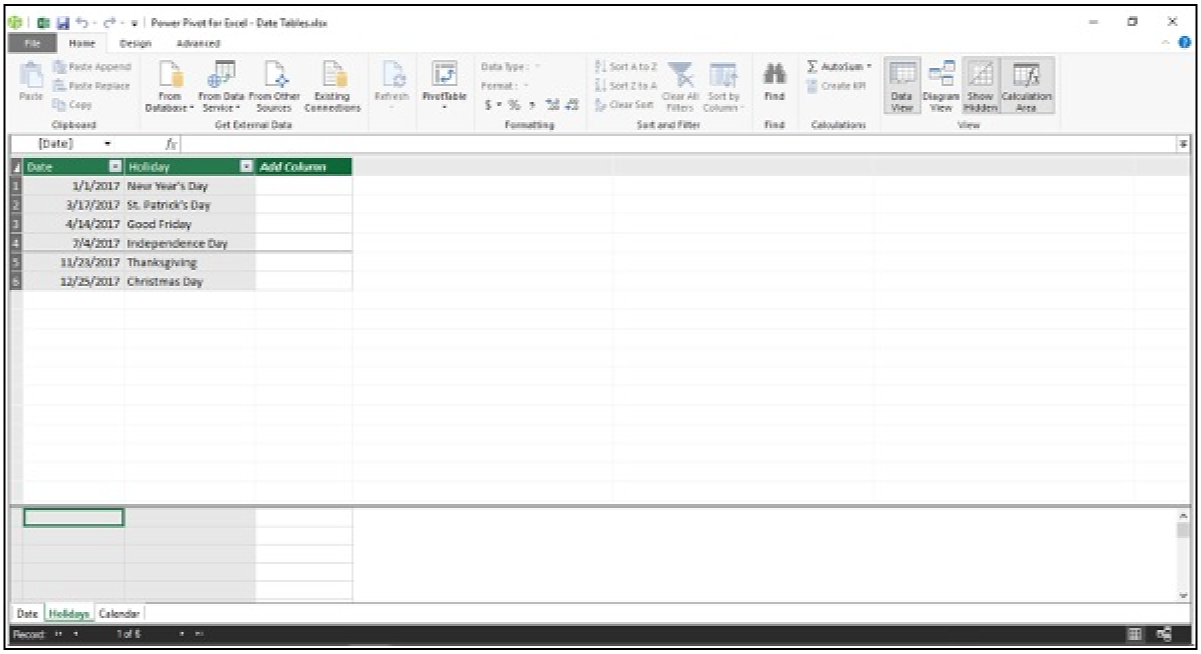
[Month Name] =FORMAT('Calendar'[Date],"MMM")

	Date	Day	Month	Year	Day of Week	Month Name	Add Column
1	1/1/2017	1	1	2017	Sun	Jan	
2	1/2/2017	2	1	2017	Mon	Jan	
3	1/3/2017	3	1	2017	Tue	Jan	
4	1/4/2017	4	1	2017	Wed	Jan	
5	1/5/2017	5	1	2017	Thu	Jan	
6	1/6/2017	6	1	2017	Fri	Jan	
7	1/7/2017	7	1	2017	Sat	Jan	
8	1/8/2017	8	1	2017	Sun	Jan	
9	1/9/2017	9	1	2017	Mon	Jan	
10	1/10/2017	10	1	2017	Tue	Jan	
11	1/11/2017	11	1	2017	Wed	Jan	
12	1/12/2017	12	1	2017	Thu	Jan	
13	1/13/2017	13	1	2017	Fri	Jan	
14	1/14/2017	14	1	2017	Sat	Jan	
15	1/15/2017	15	1	2017	Sun	Jan	
16	1/16/2017	16	1	2017	Mon	Jan	
17	1/17/2017	17	1	2017	Tue	Jan	

Adding Holidays to the Calendar Table

Add holidays to the Calendar Table as follows −

- Get the list of declared holidays for the year.
- For example, for the US, you can get the list of holidays for any required year from the following link http://www.calendar-365.com/.
- Copy and paste them into an Excel worksheet.
- Copy the Excel table and paste it into a new table in the data model.
- Name the table as Holidays.



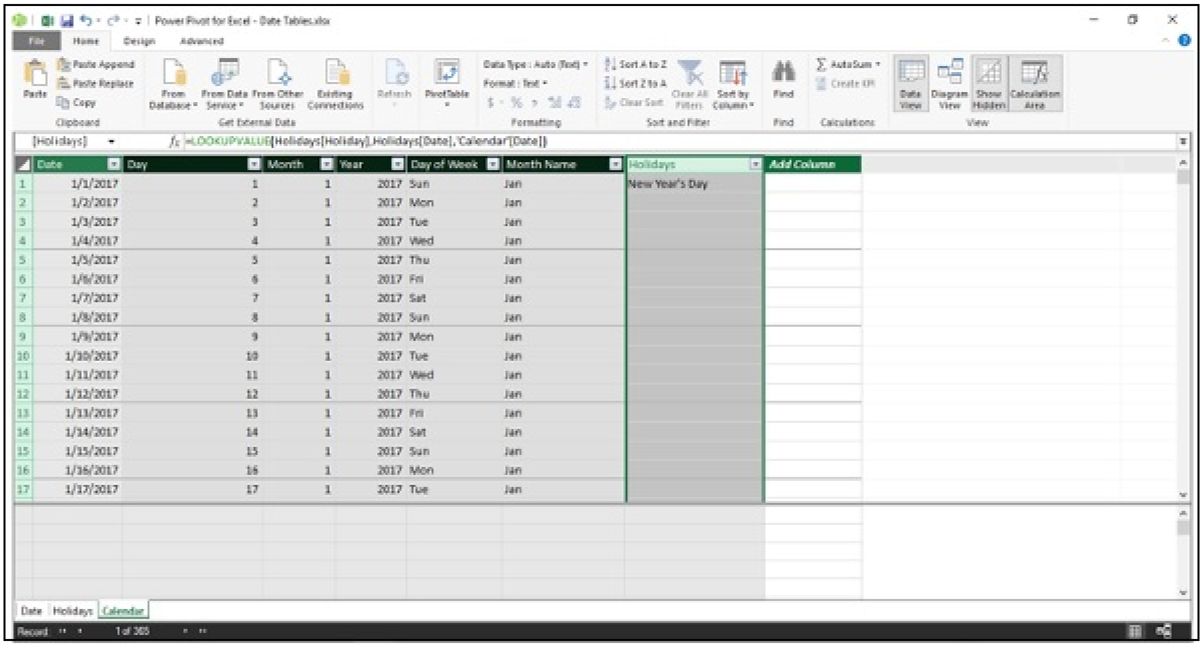
[Date]

	Date	Holiday	Add Column
1	1/1/2017	New Year's Day	
2	3/17/2017	St. Patrick's Day	
3	4/14/2017	Good Friday	
4	7/4/2017	Independence Day	
5	11/23/2017	Thanksgiving	
6	12/25/2017	Christmas Day	

- Next, you can add a calculated column of holidays to the Calendar table using DAX LOOKUPVALUE function.

=LOOKUPVALUE(Holidays[Holiday],Holidays[Date],'Calendar'[Date])

DAX LOOKUPVALUE function searches the third parameter, i.e. Calendar[Date] in the second parameter, i.e. Holidays[Date] and returns the first parameter, i.e. Holidays[Holiday] if there is a match. The result will look like what is shown in the following screenshot.



[Holidays] =LOOKUPVALUE(Holidays[Holiday],Holidays[Date],'Calendar'[Date])

	Date	Day	Month	Year	Day of Week	Month Name	Holidays	Add Column
1	1/1/2017	1	1	2017	Sun	Jan	New Year's Day	
2	1/2/2017	2	1	2017	Mon	Jan		
3	1/3/2017	3	1	2017	Tue	Jan		
4	1/4/2017	4	1	2017	Wed	Jan		
5	1/5/2017	5	1	2017	Thu	Jan		
6	1/6/2017	6	1	2017	Fri	Jan		
7	1/7/2017	7	1	2017	Sat	Jan		
8	1/8/2017	8	1	2017	Sun	Jan		
9	1/9/2017	9	1	2017	Mon	Jan		
10	1/10/2017	10	1	2017	Tue	Jan		
11	1/11/2017	11	1	2017	Wed	Jan		
12	1/12/2017	12	1	2017	Thu	Jan		
13	1/13/2017	13	1	2017	Fri	Jan		
14	1/14/2017	14	1	2017	Sat	Jan		
15	1/15/2017	15	1	2017	Sun	Jan		
16	1/16/2017	16	1	2017	Mon	Jan		
17	1/17/2017	17	1	2017	Tue	Jan		

Data Modeling with DAX

Adding Columns to a Fiscal Year

A fiscal year typically includes the dates from 1st of the month after the fiscal year end to the next fiscal year end. For example, if the fiscal year end is 31st March, then the fiscal year ranges from 1st April to 31st March.

You can include the fiscal time periods in the calendar table using the DAX formulas −

- Add a measure for FYE

  FYE:=3

- Add the following calculated columns −
  - **Fiscal Year**

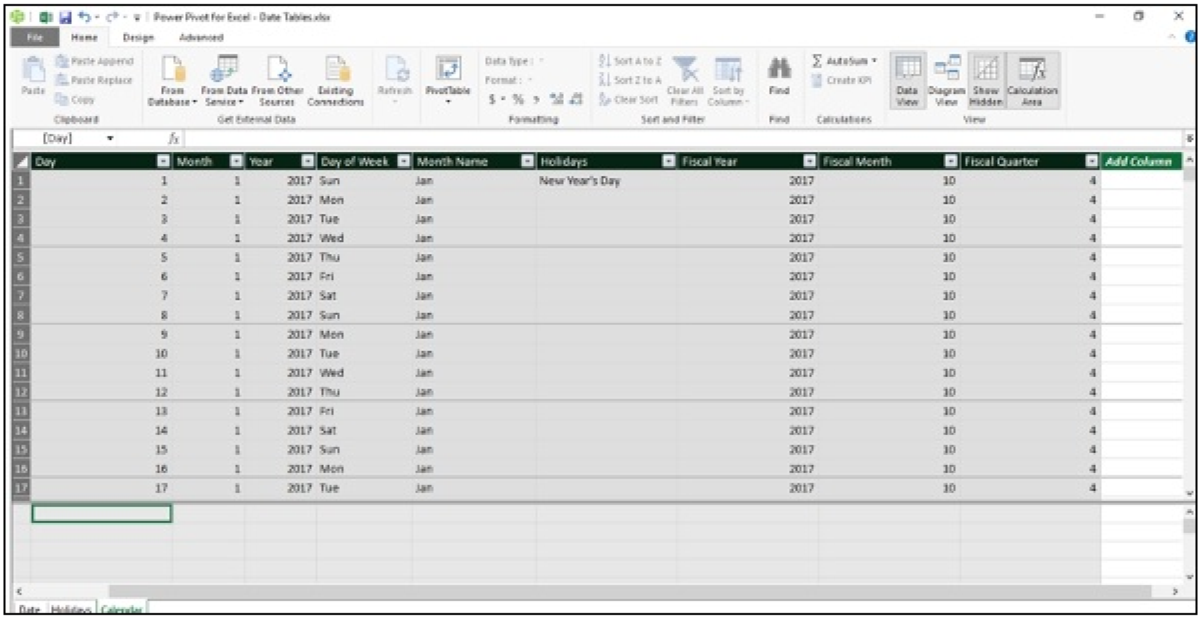
    =IF('Calendar'[Month]<='Calendar'[FYE],'Calendar'[Year],'Calendar'[Year]+1)

  - **Fiscal Month**

    =IF('Calendar'[Month]<='Calendar'[FYE],12-'Calendar'[FYE]+'Calendar'[Month],'Calendar'[Month]-'Calendar'[FYE])

  - **Fiscal Quarter**

    =INT(('Calendar'[Fiscal Month]+2)/3)



	Day	Month	Year	Day of Week	Month Name	Holidays	Fiscal Year	Fiscal Month	Fiscal Quarter	Add Column
1	1	1	2017	Sun	Jan	New Year's Day	2017	10	4	
2	2	1	2017	Mon	Jan		2017	10	4	
3	3	1	2017	Tue	Jan		2017	10	4	
4	4	1	2017	Wed	Jan		2017	10	4	
5	5	1	2017	Thu	Jan		2017	10	4	
6	6	1	2017	Fri	Jan		2017	10	4	
7	7	1	2017	Sat	Jan		2017	10	4	
8	8	1	2017	Sun	Jan		2017	10	4	
9	9	1	2017	Mon	Jan		2017	10	4	
10	10	1	2017	Tue	Jan		2017	10	4	
11	11	1	2017	Wed	Jan		2017	10	4	
12	12	1	2017	Thu	Jan		2017	10	4	
13	13	1	2017	Fri	Jan		2017	10	4	
14	14	1	2017	Sat	Jan		2017	10	4	
15	15	1	2017	Sun	Jan		2017	10	4	
16	16	1	2017	Mon	Jan		2017	10	4	
17	17	1	2017	Tue	Jan		2017	10	4	

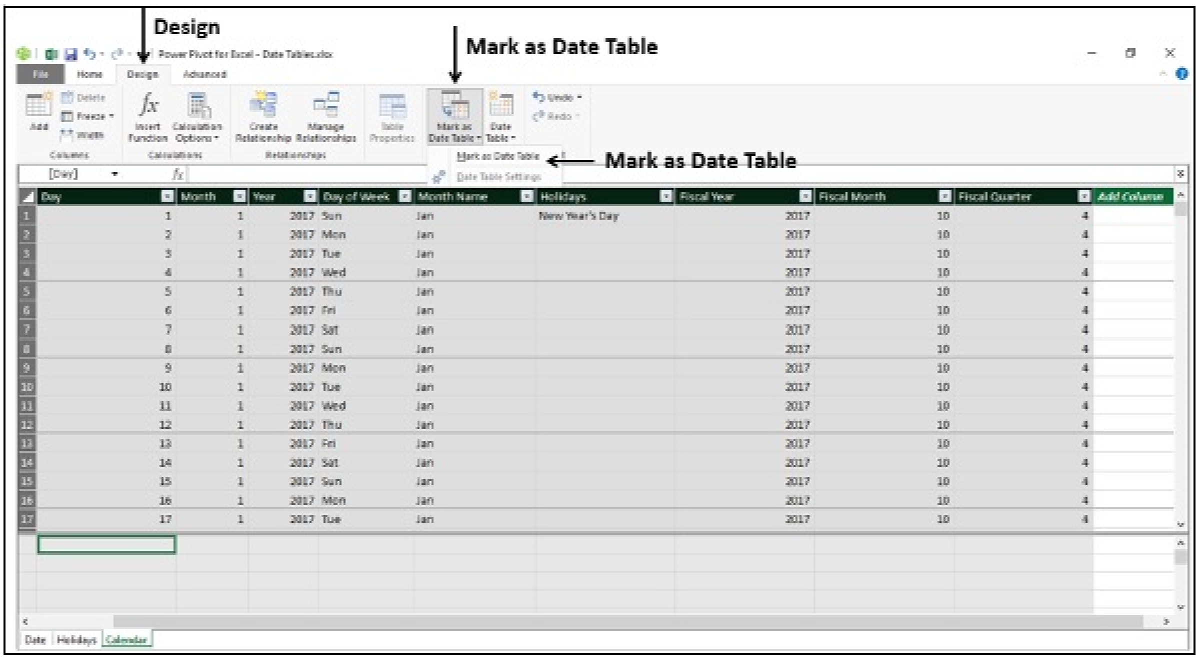
Setting the Date Table Property

Data Modeling with DAX

When you use DAX Time Intelligence functions such as TOTALYTD, PREVIOUSMONTH, and DATESBETWEEN, they require metadata to work correctly. Date Table Property sets such metadata.

To set the Date Table property −

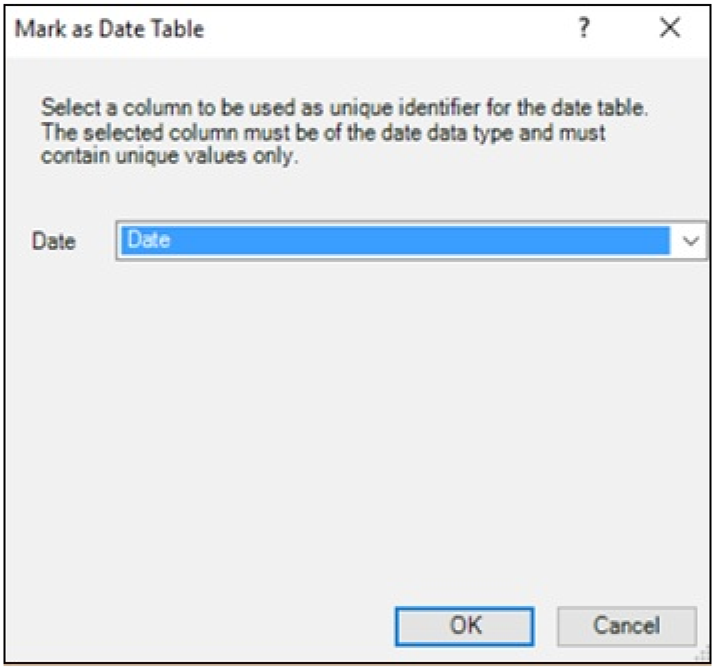
- Select Calendar table in the Power Pivot window.
- Click the Design tab on the Ribbon.
- Click Mark as Date Table in the Calendars group.
- Click Mark as Date Table in the dropdown list.



	Day	Month	Year	Day of Week	Month Name	Holidays	Fiscal Year	Fiscal Month	Fiscal Quarter	Add Column
1	1	1	2017	Sun	Jan	New Year's Day	2017	10	4	
2	2	1	2017	Mon	Jan		2017	10	4	
3	3	1	2017	Tue	Jan		2017	10	4	
4	4	1	2017	Wed	Jan		2017	10	4	
5	5	1	2017	Thu	Jan		2017	10	4	
6	6	1	2017	Fri	Jan		2017	10	4	
7	7	1	2017	Sat	Jan		2017	10	4	
8	8	1	2017	Sun	Jan		2017	10	4	
9	9	1	2017	Mon	Jan		2017	10	4	
10	10	1	2017	Tue	Jan		2017	10	4	
11	11	1	2017	Wed	Jan		2017	10	4	
12	12	1	2017	Thu	Jan		2017	10	4	
13	13	1	2017	Fri	Jan		2017	10	4	
14	14	1	2017	Sat	Jan		2017	10	4	
15	15	1	2017	Sun	Jan		2017	10	4	
16	16	1	2017	Mon	Jan		2017	10	4	
17	17	1	2017	Tue	Jan		2017	10	4	

Mark as Date Table dialog box appears. Select the Date column in the Calendar table. This has to be the column of Date data type and has to have unique values. Click OK.

Data Modeling with DAX



Select a column to be used as unique identifier for the date table. The selected column must be of the date data type and must contain unique values only.

Date: Date

Extending the Data Model

In this chapter, you will learn how to extend the data model created in the previous chapters. Extending a data model includes −

- Addition of tables
- Addition of calculated columns in an existing table
- Creation of measures in an existing table

Of these, creating the measures is crucial, as it involves providing new data insights in the data model that will enable those using the data model avoid rework and also save time while analyzing the data and decision making.

As Profit and Loss Analysis involves working with time periods and you will be using DAX Time Intelligence functions, you require a Date table in the data model.

If you are new to Date tables, go through the chapter – Understanding Date Tables.

You can extend the data model as follows −

- To create a relationship between the data table, i.e. Finance Data table and the Date table, you need to create a calculated column Date in the Finance Data table.
- To perform different types of calculations, you need to create relationships between the data table - Finance Data and the lookup tables – Accounts and Geography Locn.
- You need to create various measures that help you perform several calculations and carry out the required analysis.

Data Modeling with DAX

These steps essentially constitute the data modeling steps for Profit and Loss Analysis using the data model. However, this is the sequence of steps for any type of data analysis that you want to perform with Power Pivot data model.

Further, you will learn how to create the measures and how to use them in the Power PivotTables in the subsequent chapters. This will give you sufficient understanding of data modeling with DAX and data analysis with Power PivotTables.

Adding a Date Table to the Data Model

Create a Date table for the time periods spanning the fiscal years as follows −

- Create a table with a single column with header – Date and contiguous dates ranging from 7/1/2011 to 6/30/2018 in a new Excel worksheet.
- Copy the table from Excel and paste it into the Power Pivot window. This will create a new table in the Power Pivot data model.
- Name the table as Date.
- Ensure that the Date column in the Date table is of data type - Date (DateTime).

Next, you need to add the calculated columns – Fiscal Year, Fiscal Quarter, Fiscal Month and Month to the Date table as follows −

Fiscal Year

Suppose the fiscal year end is June 30th. Then, a fiscal year spans from 1st July to 30th June. For example, the period July 1st, 2011 (7/1/2011) to June 30th, 2012 (6/30/2012) will be the fiscal year 2012.

In the Date table, suppose you want to represent the same as FY2012.

- You need to first extract the financial year part of the Date and append it with FY.
    - For the dates in the months July 2011 to December 2011, the financial year is 1+2011.
    - For the dates in the months January 2012 to June 2012, the financial year is 0+2012.
    - To generalize, if the Month of Financial Year End is FYE, do the following −

      Integer Part of ((Month – 1)/FYE) + Year

    - Next, take the rightmost 4 characters to obtain the Financial Year.
- In DAX, you can represent the same as −

  RIGHT(INT((MONTH('Date'[Date])-1)/'Date'[FYE])+YEAR('Date'[Date]),4)

- Add the calculated column Fiscal Year in the Date table with the DAX formula −

="FY"&RIGHT(INT((MONTH('Date'[Date])-1)/'Date'[FYE])+YEAR('Date'[Date]),4)

Fiscal Quarter

If FYE represents the month of financial year end, the financial quarter is obtained as

Integer Part of ((Remainder of ((Month+FYE-1)/12) + 3)/3)

- In DAX, you can represent the same as −

  INT((MOD(MONTH('Date'[Date])+'Date'[FYE]-1,12)+3)/3)

- Add the calculated column Fiscal Quarter in the Date table with the DAX formula −

  ='Date'[FiscalYear]&"-Q"&FORMAT(INT((MOD(MONTH('Date'[Date]) + 'Date'[FYE]-1,12) + 3)/3), "0")

Fiscal Month

If FYE represents the financial year end, the financial month period is obtained as

(Remainder of (Month+FYE-1)/12) + 1

- In DAX, you can represent the same as −

  MOD(MONTH('Date'[Date])+'Date'[FYE]-1,12)+1

- Add the calculated column Fiscal Month in the Date table with the DAX formula −

  ='Date'[Fiscal Year]&"-P" & FORMAT(MOD(MONTH([Date])+[FYE]-1,12)+1,"00")

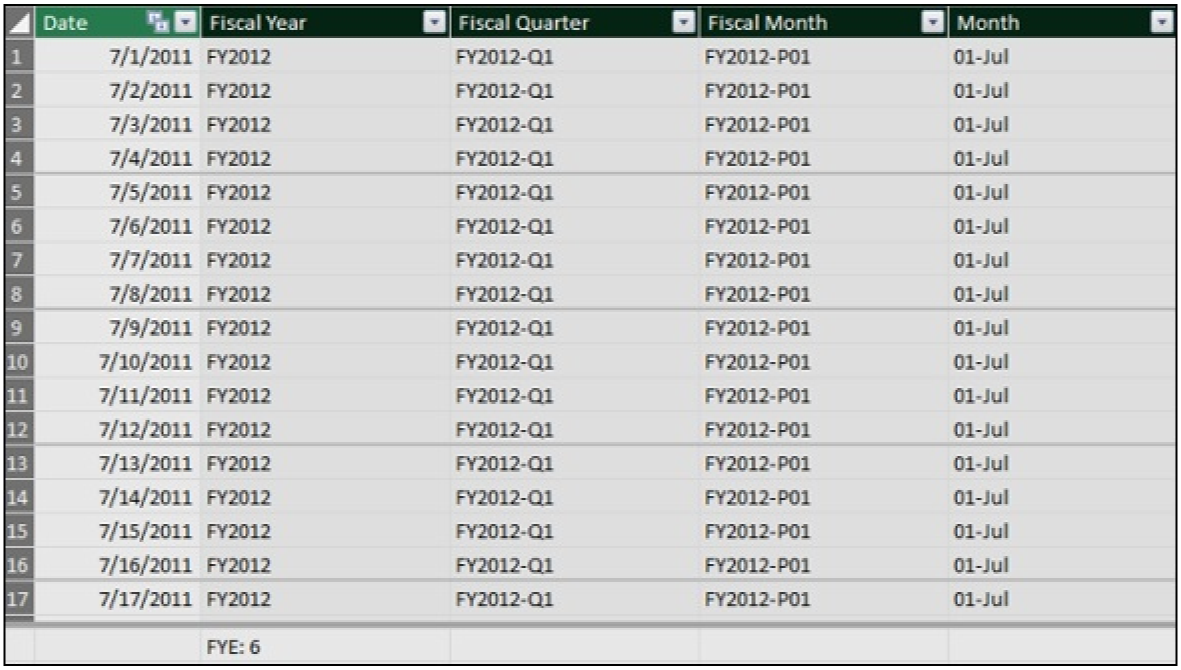
Month

Finally, add the calculated column Month that represents the month number in a financial year as follows −

=FORMAT(MOD(MONTH([Date])+[FYE]-1,12)+1,"00") & "-" & FORMAT([Date],"mmm")

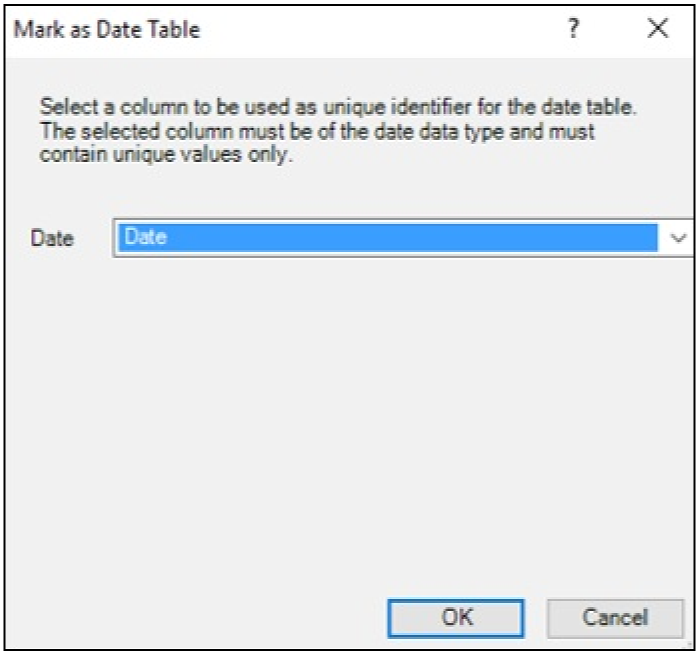
The resulting Date table looks like the following screenshot.

Data Modeling with DAX



	Date	Fiscal Year	Fiscal Quarter	Fiscal Month	Month
1	7/1/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
2	7/2/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
3	7/3/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
4	7/4/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
5	7/5/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
6	7/6/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
7	7/7/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
8	7/8/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
9	7/9/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
10	7/10/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
11	7/11/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
12	7/12/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
13	7/13/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
14	7/14/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
15	7/15/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
16	7/16/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
17	7/17/2011	FY2012	FY2012-Q1	FY2012-P01	01-Jul
		FYE: 6			

Mark the table – Date as Date Table with the column - Date as the column with unique values as shown in the following screenshot.



Mark as Date Table

Select a column to be used as unique identifier for the date table. The selected column must be of the date data type and must contain unique values only.

Date: Date

Adding Calculated Columns

To create a relationship between the Finance Data table and the Date table, you require a column of Date values in the Finance Data table.

- Add a calculated column Date in the Finance Data table with the DAX formula −

= DATEVALUE ('Finance Data'[Fiscal Month])

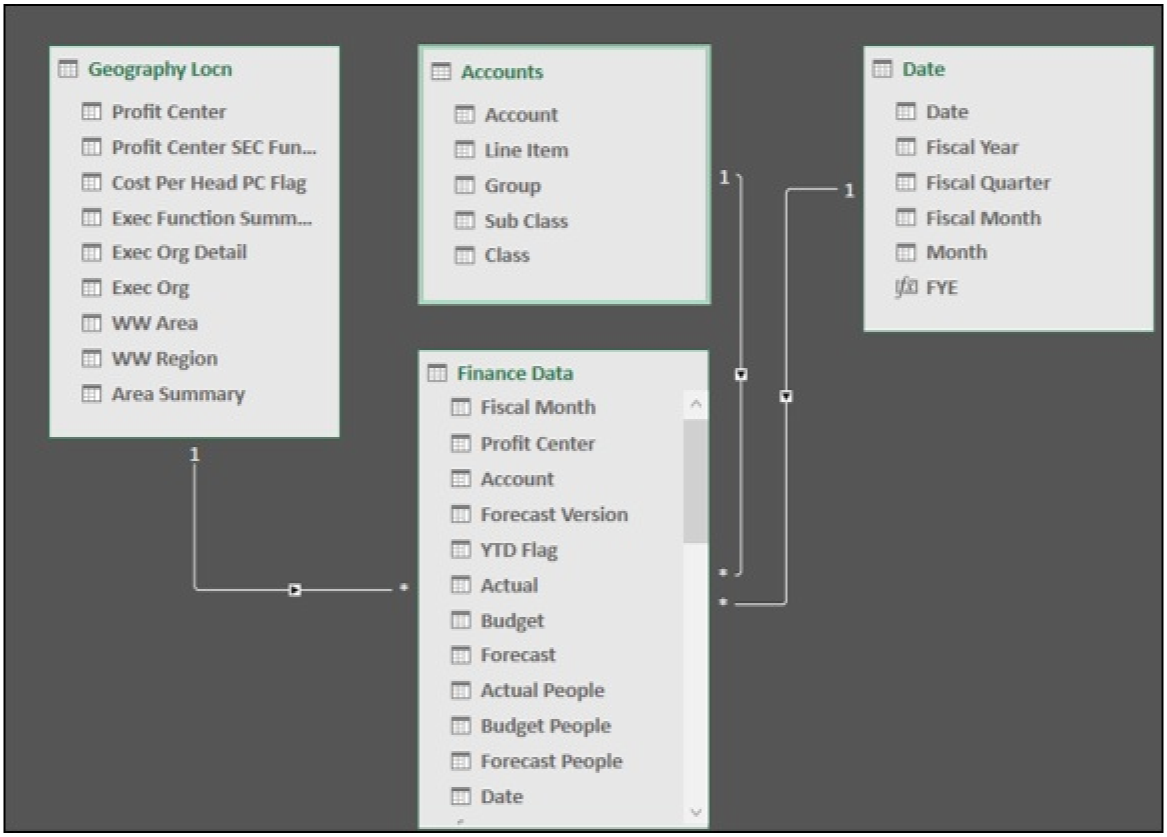
Defining Relationships Between Tables in the Data Model

You have the following tables in the data model −

- Data table - Finance Data
- Lookup tables - Accounts and Geography Locn
- Date table - Date

To define Relationships between the tables in the data model, following are the steps −

- View the tables in the Diagram View of the Power Pivot.
- Create the following relationships between the tables −
  - Relationship between Finance Data table and Accounts table with the column Account.
  - Relationship between Finance Data table and Geography Locn table with the column Profit Center.
  - Relationship between Finance Data table and Date table with the column Date.



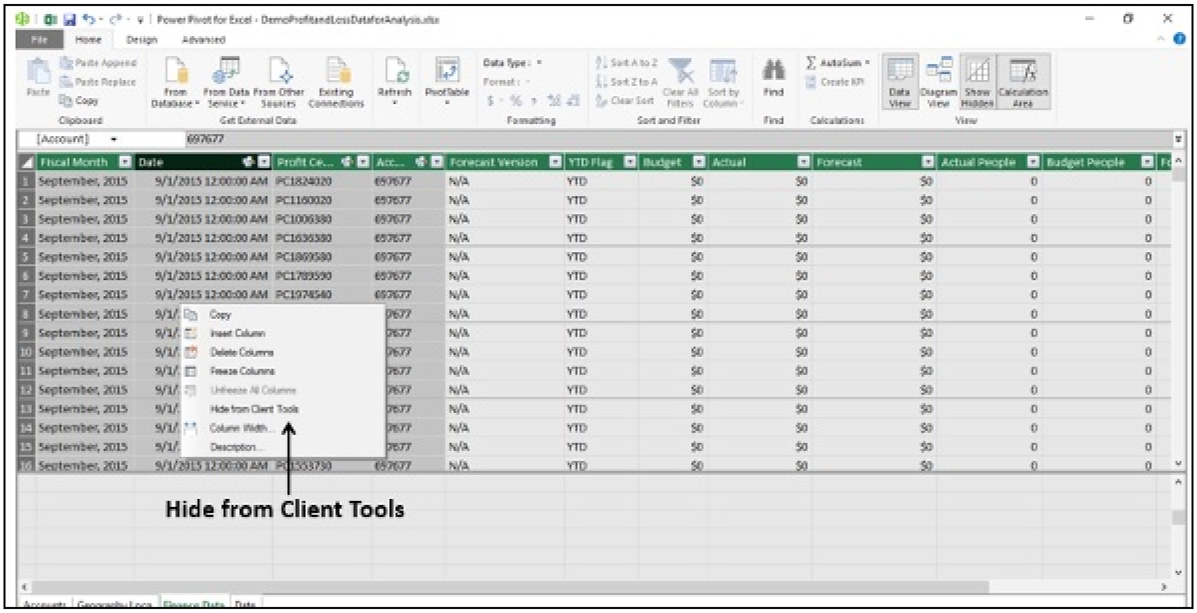
Data Modeling with DAX

Hiding Columns from Client Tools

If there are any columns in a data table that you won’t be using as fields in any PivotTable, you can hide them in the data model. Then, they will not be visible in the PivotTable Fields list.

In the Finance Data table, you have 4 columns – Fiscal Month, Date, Account and Profit Center that you won’t be using as fields in any PivotTable. Hence, you can hide them so that they do not appear in the PivotTable Fields list.

- Select the columns - Fiscal Month, Date, Account, and Profit Center in the Finance Data table.
- Right-click and select Hide from Client Tools in the dropdown list.



	Fiscal Month	Date	Profit Ce...	Acc...	Forecast Version	YTD Flag	Budget	Actual	Forecast	Actual People	Budget People
1	September, 2015	9/1/2015 12:00:00 AM	PC1824020	697677	N/A	YTD	$0	$0	$0	0	0
2	September, 2015	9/1/2015 12:00:00 AM	PC1160020	697677	N/A	YTD	$0	$0	$0	0	0
3	September, 2015	9/1/2015 12:00:00 AM	PC1006380	697677	N/A	YTD	$0	$0	$0	0	0
4	September, 2015	9/1/2015 12:00:00 AM	PC1636380	697677	N/A	YTD	$0	$0	$0	0	0
5	September, 2015	9/1/2015 12:00:00 AM	PC1869580	697677	N/A	YTD	$0	$0	$0	0	0
6	September, 2015	9/1/2015 12:00:00 AM	PC1789590	697677	N/A	YTD	$0	$0	$0	0	0
7	September, 2015	9/1/2015 12:00:00 AM	PC1974540	697677	N/A	YTD	$0	$0	$0	0	0
8	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
9	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
10	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
11	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
12	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
13	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
14	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
15	September, 2015	9/1/		7677	N/A	YTD	$0	$0	$0	0	0
16	September, 2015	9/1/2015 12:00:00 AM	PC1553730	697677	N/A	YTD	$0	$0	$0	0	0

Creating Measures in the Tables

You are all set for data modeling and analysis with DAX using the data model and Power PivotTables.

In the subsequent chapters, you will learn how to create measures and how to use them in Power PivotTables. You will create all the measures in the data table, i.e. Finance Data table.

You will create measures using DAX formulas in the data table – Finance Data, which you can use in any number of PivotTables for the data analysis. The measures are essentially the metadata. Creating measures in the data table is part of data modeling and summarizing them in the Power PivotTables is part of data analysis.

Base Finance Measures and Analysis

Data Modeling with DAX

You can create various measures in the data model to be used in any number of Power PivotTables. This forms the data modeling and analysis process with the data model using DAX.

As you have learnt earlier in the previous sections, data modeling and analysis is dependent on specific business and context. In this chapter, you will learn data modeling and analysis based on a sample Profit and Loss database to understand how to create the required measures and use them in various Power PivotTables.

You can apply the same method for data modeling and analysis for any business and context

Creating Measures Based on Finance Data

To create any financial report, you need to make calculations of amounts for a particular time period, organization, account, or geographical location. You also need to perform the headcount and cost per headcount calculations. In the data model, you can create base measures that can be reused in creating other measures. This is an effective way of data modeling with DAX.

In order to perform calculations for profit and loss data analysis, you can create measures such as sum, year-over-year, year-to-date, quarter-to-date, variance, headcount, cost per headcount, etc. You can use these measures in the Power PivotTables to analyze the data and report the analysis results.

In the following sections, you will learn how to create the base finance measures and analyze data with those measures. The measures are termed as base measures as they can be used in creating other financial measures. You will also learn how to create measures for the previous time periods and use them in the analysis.

Creating Base Finance Measures

In the finance data analysis, budget and forecast play a major role.

Budget

A budget is an estimate of a company's revenues and expenses for a financial year. The budget is calculated at the beginning of a financial year keeping in view the company's goals and targets. Budget measures need to be analyzed from time to time during the financial year, as the market conditions may change and the company may have to align its goals and targets to the current trends in the industry.

Forecast

A financial forecast is an estimate of a company's future financial outcomes by examining the company's historical data of revenues and expenses. You can use financial forecasting for the following −

Data Modeling with DAX

- To determine how to allocate budget for a future period.
- To track the expected performance of the company.
- To take timely decisions to address shortfalls against the targets, or to maximize an emerging opportunity.

Actuals

To perform the budgeting and forecasting calculations, you require the actual revenue and expenses at any point in time.

You can create the following 3 base finance measures that can be used in creating other financial measures in the data mode −

- Budget Sum
- Actual Sum
- Forecast Sum

These measures are the aggregation sums over the columns – Budget, Actual, and Forecast in the Finance Data table.

Create the base finance measures as follows −

Budget Sum

Budget Sum:=SUM('Finance Data'[Budget])

Actual Sum

Actual Sum:=SUM('Finance Data'[Actual])

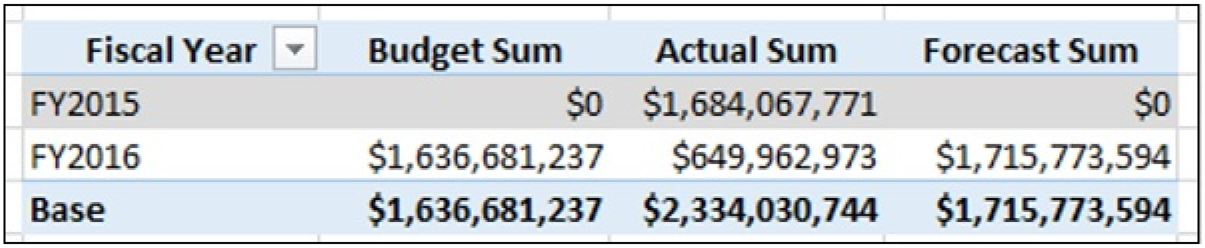
Forecast Sum

Forecast Sum:=SUM('Finance Data'[Forecast])

Analyzing Data with Base Finance Measures

With the base finance measures and the Date table, you can perform your analysis as follow −

- Create a Power PivotTable.
- Add the field Fiscal Year from the Date table to Rows.
- Add the measures Budget Sum, Actual Sum, and Forecast Sum (that appear as fields in the PivotTable Fields list) to Values.



Fiscal Year	Budget Sum	Actual Sum	Forecast Sum
FY2015	$0	$1,684,067,771	$0
FY2016	$1,636,681,237	$649,962,973	$1,715,773,594
Base	**$1,636,681,237**	**$2,334,030,744**	**$1,715,773,594**

Creating Finance Measures for Previous Periods

With the three base finance measures and the Date table, you can create other finance measures.

Suppose you want to compare the Actual Sum of a Quarter with the Actual Sum of previous Quarter. You can create the measure - Prior Quarter Actual Sum.

Prior Quarter Actual Sum:=CALCULATE([Actual Sum],
DATEADD('Date'[Date],1,QUARTER))

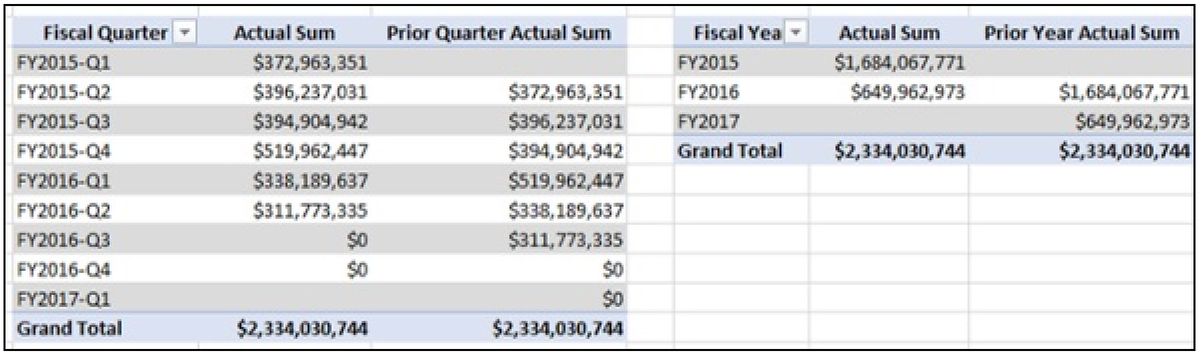
Similarly, you can create the measure - Prior Year Actual Sum.

Prior Year Actual Sum:=CALCULATE([Actual Sum], DATEADD('Date'[Date],1,YEAR))

Analyzing Data with Finance Measures for Previous Periods

With the base measures, measures for previous periods and the Date table, you can perform your analysis as follows −

- Create a Power PivotTable.
- Add the field Fiscal Quarter from the Date table to Rows.
- Add the measures Actual Sum and Prior Quarter Actual Sum to Values.
- Create another Power PivotTable.
- Add the field Fiscal Year from the Date table to Rows.
- Add the measures Actual Sum and Prior Year Actual Sum to Values.



Fiscal Quarter	Actual Sum	Prior Quarter Actual Sum
FY2015-Q1	$372,963,351	
FY2015-Q2	$396,237,031	$372,963,351
FY2015-Q3	$394,904,942	$396,237,031
FY2015-Q4	$519,962,447	$394,904,942
FY2016-Q1	$338,189,637	$519,962,447
FY2016-Q2	$311,773,335	$338,189,637
FY2016-Q3	$0	$311,773,335
FY2016-Q4	$0	$0
FY2017-Q1		$0
Grand Total	**$2,334,030,744**	**$2,334,030,744**

Fiscal Yea	Actual Sum	Prior Year Actual Sum
FY2015	$1,684,067,771	
FY2016	$649,962,973	$1,684,067,771
FY2017		$649,962,973
Grand Total	**$2,334,030,744**	**$2,334,030,744**

Data Modeling with DAX

YoY Finance Measures and Analysis

Year-over-Year (YoY) is a measure of growth. It is obtained by subtracting the actual sum of the previous year from the actual sum.

If the result is positive, it reflects an increase in actual, and if it is negative, it reflects a decrease in actual, i.e. if we calculate year-over-year as −

year-over-year = (actual sum –prior year actual sum)

- If the actual sum > the prior year actual sum, year-over-year will be positive.
- If the actual sum < the prior year actual sum, year-over-year will be negative.

In the financial data, accounts such as the expense accounts will have debit (positive) amounts and the revenue accounts will have credit (negative) amounts. Hence, for the expense accounts, the above formula works fine.

However, for the revenue accounts, it should be the reverse, i.e.

- If the actual sum > the prior year actual sum, year-over-year should be negative.
- If the actual sum < the prior year actual sum, year-over-year should be positive.

Hence for the revenue accounts, you have to calculate year-over-year as −

year-over-year = -(actual sum – prior year actual sum)

Creating Year-over-Year Measure

You can create Year-over-Year measure with the following DAX formula −

YoY:=IF(CONTAINS(Accounts, Accounts[Class],"Net Revenue"),-([Actual Sum]-[Prior Year Actual Sum]), [Actual Sum]-[Prior Year Actual Sum])

In the above DAX formula −

- DAX CONTAINS function returns TRUE, if a row has "Net Revenue" in the column Class in the Accounts table.
- DAX IF function then returns –([Actual Sum]-[Prior Year Actual Sum]).
- Otherwise, DAX IF function returns [Actual Sum]-[Prior Year Actual Sum].

Creating Year-over-Year Percentage Measure

You can represent Year-over-Year as a percentage with the ratio −

(YoY) / (Prior Year Actual Sum)

You can create the Year-over-Year Percentage measure with the following DAX formula −

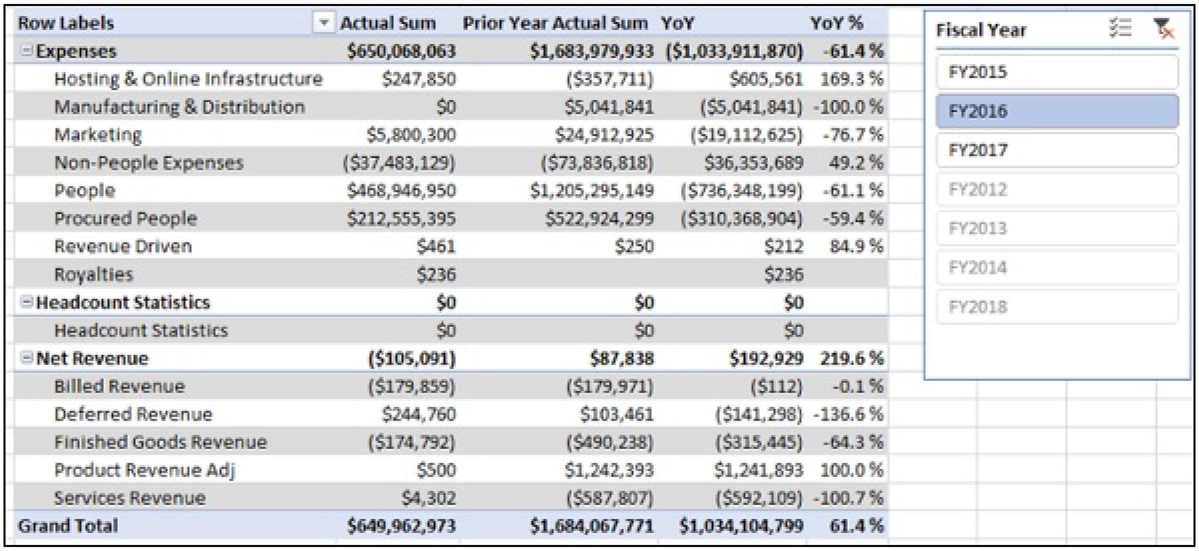
YoY %:=IF([Prior Year Actual Sum], [YoY] / ABS([Prior Year Actual Sum]),BLANK())

DAX IF function is used in the above formula to ensure that there is no division by zero.

Analyzing Data with Year-over-Year Measures

Create a Power PivotTable as follows −

- Add the fields Class and Sub Class from the Accounts table to Rows.
- Add the measures – Actual Sum, Prior Year Actual Sum, YoY and YoY % to Values.
- Insert a Slicer on the field Fiscal Year from the Date table.
- Select FY2016 in the Slicer.



Row Labels	Actual Sum	Prior Year Actual Sum	YoY	YoY %
⊟Expenses	$650,068,063	$1,683,979,933	($1,033,911,870)	-61.4 %
Hosting & Online Infrastructure	$247,850	($357,711)	$605,561	169.3 %
Manufacturing & Distribution	$0	$5,041,841	($5,041,841)	-100.0 %
Marketing	$5,800,300	$24,912,925	($19,112,625)	-76.7 %
Non-People Expenses	($37,483,129)	($73,836,818)	$36,353,689	49.2 %
People	$468,946,950	$1,205,295,149	($736,348,199)	-61.1 %
Procured People	$212,555,395	$522,924,299	($310,368,904)	-59.4 %
Revenue Driven	$461	$250	$212	84.9 %
Royalties	$236		$236	
⊟Headcount Statistics	$0	$0	$0	
Headcount Statistics	$0	$0	$0	
⊟Net Revenue	($105,091)	$87,838	$192,929	219.6 %
Billed Revenue	($179,859)	($179,971)	($112)	-0.1 %
Deferred Revenue	$244,760	$103,461	($141,298)	-136.6 %
Finished Goods Revenue	($174,792)	($490,238)	($315,445)	-64.3 %
Product Revenue Adj	$500	$1,242,393	$1,241,893	100.0 %
Services Revenue	$4,302	($587,807)	($592,109)	-100.7 %
Grand Total	$649,962,973	$1,684,067,771	$1,034,104,799	61.4 %

Creating Budget Year-over-Year Measure

You can create Budget Year-over-Year measure as follows −

Budget YoY: = IF(CONTAINS(Accounts,Accounts[Class],"Net Revenue"), - ([Budget Sum] - [Prior Year Actual Sum]), [Budget Sum] - [Prior Year Actual Sum])

Creating Budget Year-over-Year Percentage Measure

You can create Budget Year-over-Year Percentage measure as follows −

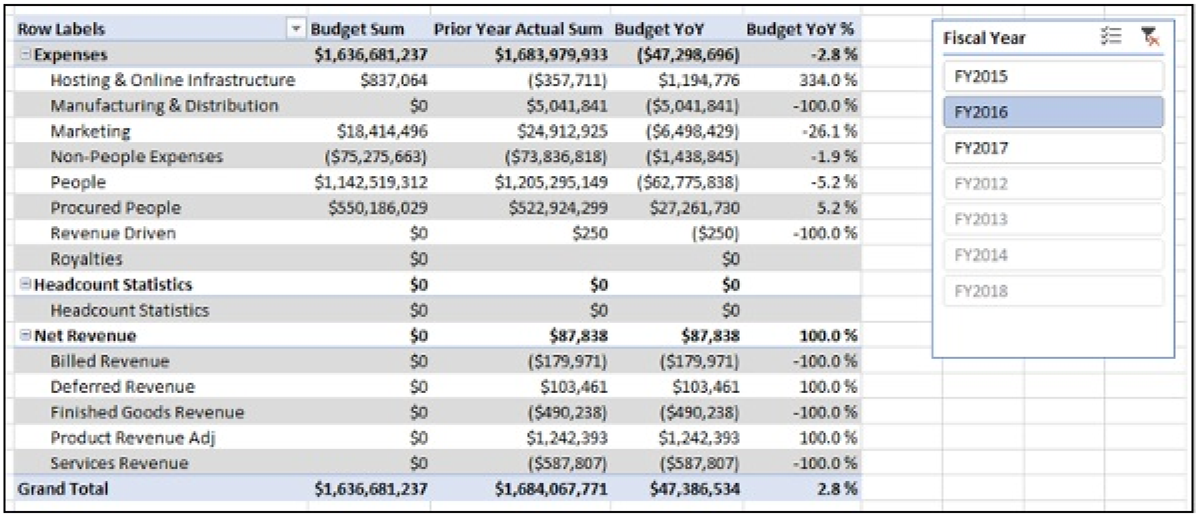
Data Modeling with DAX

Budget YoY %:=IF([Prior Year Actual Sum] , [Budget YoY] / ABS ([Prior Year Actual Sum]) , BLANK())

Analyzing Data with Budget Year-over-Year Measures

Create a Power PivotTable as follows −

- Add the fields Class and Sub Class from the Accounts table to Rows.
- Add the measures – Budget Sum, Prior Year Actual Sum, Budget YoY and Budget YoY % to Values.
- Insert a Slicer on the field Fiscal Year from the Date table.
- Select FY2016 in the Slicer.



Row Labels	Budget Sum	Prior Year Actual Sum	Budget YoY	Budget YoY %
⊟Expenses	$1,636,681,237	$1,683,979,933	($47,298,696)	-2.8 %
Hosting & Online Infrastructure	$837,064	($357,711)	$1,194,776	334.0 %
Manufacturing & Distribution	$0	$5,041,841	($5,041,841)	-100.0 %
Marketing	$18,414,496	$24,912,925	($6,498,429)	-26.1 %
Non-People Expenses	($75,275,663)	($73,836,818)	($1,438,845)	-1.9 %
People	$1,142,519,312	$1,205,295,149	($62,775,838)	-5.2 %
Procured People	$550,186,029	$522,924,299	$27,261,730	5.2 %
Revenue Driven	$0	$250	($250)	-100.0 %
Royalties	$0		$0	
⊟Headcount Statistics	$0	$0	$0	
Headcount Statistics	$0	$0	$0	
⊟Net Revenue	$0	$87,838	$87,838	100.0 %
Billed Revenue	$0	($179,971)	($179,971)	-100.0 %
Deferred Revenue	$0	$103,461	$103,461	100.0 %
Finished Goods Revenue	$0	($490,238)	($490,238)	-100.0 %
Product Revenue Adj	$0	$1,242,393	$1,242,393	100.0 %
Services Revenue	$0	($587,807)	($587,807)	-100.0 %
Grand Total	$1,636,681,237	$1,684,067,771	$47,386,534	2.8 %

Creating Forecast Year-over-Year Measure

You can create Forecast Year-over-Year measure as follows −

Forecast YoY:=IF(CONTAINS(Accounts,Accounts[Class],"Net Revenue"), - ([Forecast Sum] - [Prior Year Actual Sum]), [Forecast Sum] - [Prior Year Actual Sum])

Creating Forecast Year-over-Year Percentage Measure

You can create Forecast Year-over-Year Percentage measure as follows −

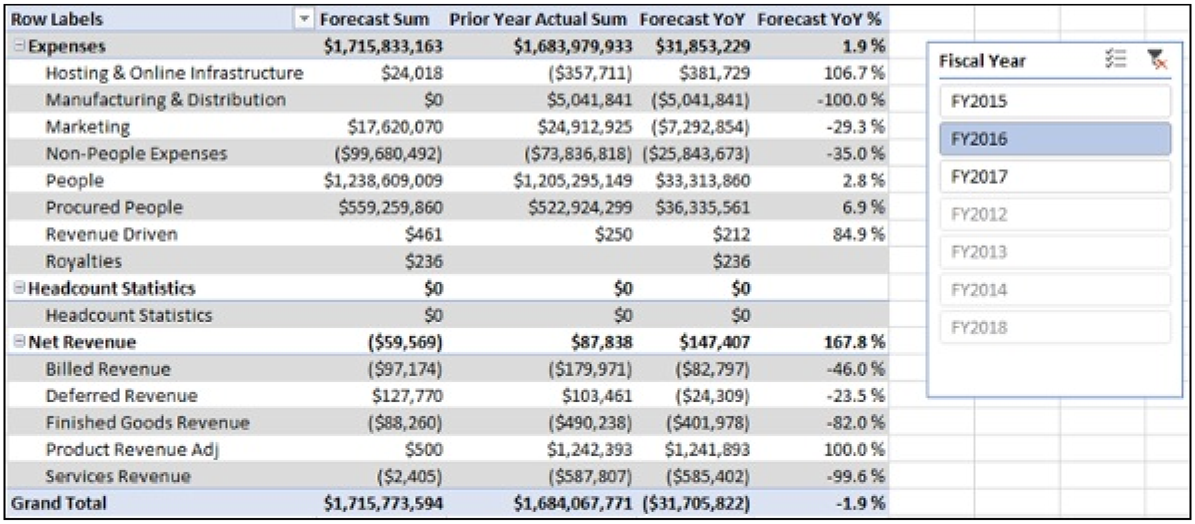
Forecast YoY %:=IF([Prior Year Actual Sum],[Forecast YoY]/ABS([Prior Year Actual Sum]),BLANK())

Analyzing Data with Forecast Year-over-Year Measures

Data Modeling with DAX

Create a Power PivotTable as follows −

- Add the fields Class and Sub Class from the Accounts table to Rows.
- Add the measures – Forecast Sum, Prior Year Actual Sum, Forecast YoY and Forecast YoY % to Values.
- Insert a Slicer on the field Fiscal Year from the Data table.
- Select FY2016 in the Slicer.



Row Labels	Forecast Sum	Prior Year Actual Sum	Forecast YoY	Forecast YoY %
Expenses	$1,715,833,163	$1,683,979,933	$31,853,229	1.9 %
Hosting & Online Infrastructure	$24,018	($357,711)	$381,729	106.7 %
Manufacturing & Distribution	$0	$5,041,841	($5,041,841)	-100.0 %
Marketing	$17,620,070	$24,912,925	($7,292,854)	-29.3 %
Non-People Expenses	($99,680,492)	($73,836,818)	($25,843,673)	-35.0 %
People	$1,238,609,009	$1,205,295,149	$33,313,860	2.8 %
Procured People	$559,259,860	$522,924,299	$36,335,561	6.9 %
Revenue Driven	$461	$250	$212	84.9 %
Royalties	$236		$236	
Headcount Statistics	$0	$0	$0	
Headcount Statistics	$0	$0	$0	
Net Revenue	($59,569)	$87,838	$147,407	167.8 %
Billed Revenue	($97,174)	($179,971)	($82,797)	-46.0 %
Deferred Revenue	$127,770	$103,461	($24,309)	-23.5 %
Finished Goods Revenue	($88,260)	($490,238)	($401,978)	-82.0 %
Product Revenue Adj	$500	$1,242,393	$1,241,893	100.0 %
Services Revenue	($2,405)	($587,807)	($585,402)	-99.6 %
Grand Total	$1,715,773,594	$1,684,067,771	($31,705,822)	-1.9 %

Variance Measures and Analysis

You can create variance measures such as variance to budget, variance to forecast, and forecast variance to budget. You can also analyze the finance data based on these measures.

Creating Variance to Budget Sum Measure

Create Variance to Budget Sum measure (VTB Sum) as follows −

VTB Sum:=[Budget Sum]-[Actual Sum]

Creating Variance to Budget Percentage Measure

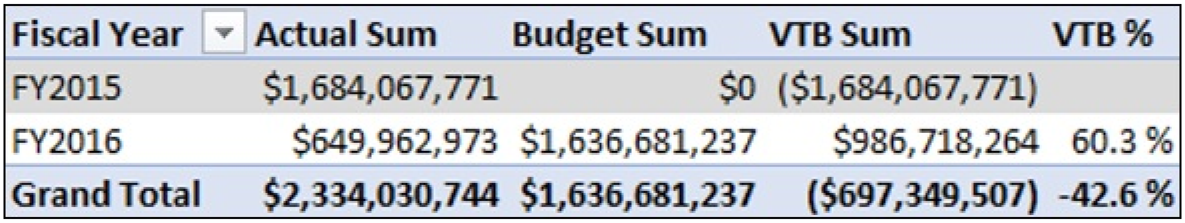
Create Variance to Budget Percentage measure (VTB %) as follows −

VTB %:=IF([Budget Sum],[VTB Sum]/ABS([Budget Sum]),BLANK())

Analyzing Data with Variance to Budget Measures

Create a Power PivotTable as follows −

- Add Fiscal Year from the Date table to Rows.
- Add the measures Actual Sum, Budget Sum, VTB Sum, VTB % from the Finance Data table to Values.



Fiscal Year	Actual Sum	Budget Sum	VTB Sum	VTB %
FY2015	$1,684,067,771	$0	($1,684,067,771)	
FY2016	$649,962,973	$1,636,681,237	$986,718,264	60.3 %
Grand Total	$2,334,030,744	$1,636,681,237	($697,349,507)	-42.6 %

Creating Variance to Forecast Sum Measure

Create Variance to Forecast Sum (VTF Sum) measure as follows −

VTF Sum:=[Forecast Sum]-[Actual Sum]

Creating Variance to Forecast Percentage Measure

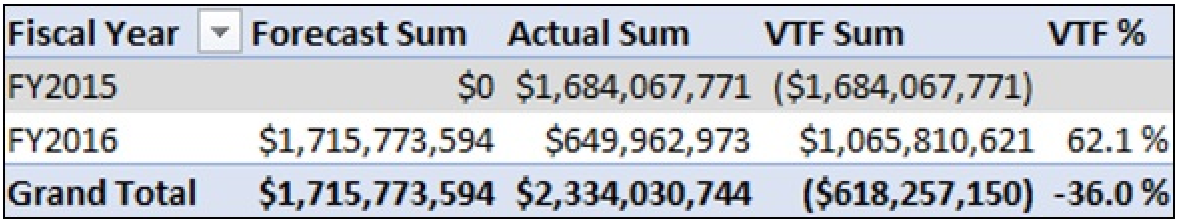
Create Variance to Forecast Percentage measure (VTF %) as follows −

VTF %:=IF([Forecast Sum],[VTF Sum]/ABS([Forecast Sum]),BLANK())

Analyzing Data with Variance to Forecast Measures

Create a Power PivotTable as follows −

- Add Fiscal Year from the Date table to Rows.
- Add the measures Actual Sum, Forecast Sum, VTF Sum, VTF % from the Finance Data table to Values.



Fiscal Year	Forecast Sum	Actual Sum	VTF Sum	VTF %
FY2015	$0	$1,684,067,771	($1,684,067,771)	
FY2016	$1,715,773,594	$649,962,973	$1,065,810,621	62.1 %
Grand Total	$1,715,773,594	$2,334,030,744	($618,257,150)	-36.0 %

Creating Forecast Variance to Budget Sum Measure

Create Forecast Variance to Budget Sum (Forecast VTB Sum) measure as follows −

Forecast VTB Sum:=[Budget Sum]-[Forecast Sum]

Creating Forecast Variance to Budget Percentage Measure

Data Modeling with DAX

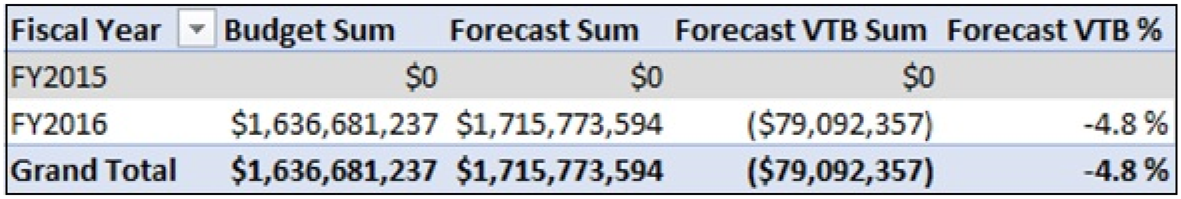
Create Forecast Variance to Budget Percentage (Forecast VTB Percentage) measure as follows −

Forecast VTB %:=IF([Budget Sum],[Forecast VTB Sum]/ABS([Budget Sum]),BLANK())

Analyzing Data with Forecast Variance to Budget Measures

Create a Power PivotTable as follows −

- Add Fiscal Year from the Date table to Rows.
- Add the measures Budget Sum, Forecast Sum, Forecast VTB Sum, Forecast VTB % from the Finance Data table to Values.



Fiscal Year	Budget Sum	Forecast Sum	Forecast VTB Sum	Forecast VTB %
FY2015	$0	$0	$0	
FY2016	$1,636,681,237	$1,715,773,594	($79,092,357)	-4.8 %
Grand Total	**$1,636,681,237**	**$1,715,773,594**	**($79,092,357)**	**-4.8 %**

Year-to-Date Measures and Analysis

To calculate a result that includes a starting balance from the beginning of a period, such as a fiscal year, up to a specific period in time, you can use DAX Time Intelligence functions. This will enable you to analyze data on a month level.

In this chapter, you will learn how to create Year-to-Date measures and how to carry out data analysis with the same.

Creating Year-to-Date Actual Sum Measure

Create Year-to-Date Actual Sum measure as follows −

YTD Actual Sum:=TOTALYTD([Actual Sum], 'Date'[Date], ALL('Date'), "6/30")

Creating Year-to-Date Budget Sum Measure

Create Year-to-Date Budget Sum measure as follows −

YTD Budget Sum:=TOTALYTD([Budget Sum], 'Date'[Date], ALL('Date'), "6/30")

Creating Year-to-Date Forecast Sum Measure

Create Year-to-Date Forecast Sum measure as follows −

YTD Forecast Sum:=TOTALYTD([Forecast Sum], 'Date'[Date], ALL('Date'), "6/30")

Data Modeling with DAX

Creating Prior Year-to-Date Actual Sum Measure

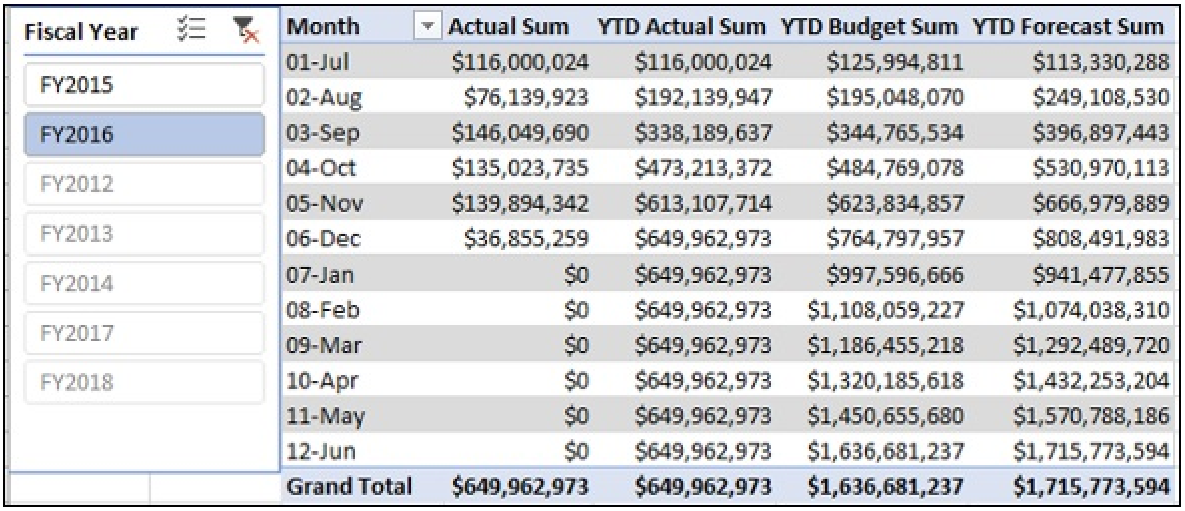
Create Prior Year-to-Date Actual Sum measure as follows −

Prior YTD Actual Sum:=TOTALYTD([Prior Year Actual Sum], 'Date'[Date], ALL('Date'), "6/30")

Analyzing Data with Year-to-Date Measures

Create a Power PivotTable as follows −

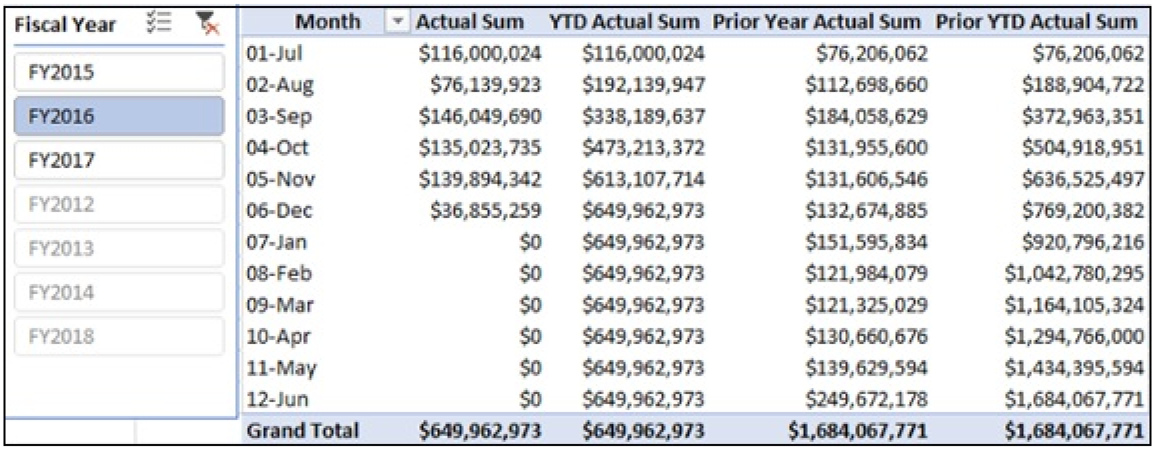
- Add Month from Date table to Rows.
- Add the measures Actual Sum, YTD Actual Sum, YTD Budget Sum, and YTD Forecast Sum from the Finance Data table to Values.
- Insert a Slicer on the Fiscal Year from the Date table.
- Select FY2016 in the Slicer.



Month	Actual Sum	YTD Actual Sum	YTD Budget Sum	YTD Forecast Sum
01-Jul	$116,000,024	$116,000,024	$125,994,811	$113,330,288
02-Aug	$76,139,923	$192,139,947	$195,048,070	$249,108,530
03-Sep	$146,049,690	$338,189,637	$344,765,534	$396,897,443
04-Oct	$135,023,735	$473,213,372	$484,769,078	$530,970,113
05-Nov	$139,894,342	$613,107,714	$623,834,857	$666,979,889
06-Dec	$36,855,259	$649,962,973	$764,797,957	$808,491,983
07-Jan	$0	$649,962,973	$997,596,666	$941,477,855
08-Feb	$0	$649,962,973	$1,108,059,227	$1,074,038,310
09-Mar	$0	$649,962,973	$1,186,455,218	$1,292,489,720
10-Apr	$0	$649,962,973	$1,320,185,618	$1,432,253,204
11-May	$0	$649,962,973	$1,450,655,680	$1,570,788,186
12-Jun	$0	$649,962,973	$1,636,681,237	$1,715,773,594
Grand Total	$649,962,973	$649,962,973	$1,636,681,237	$1,715,773,594

Create a Power PivotTable as follows −

- Add Month from Date table to Rows.
- Add the measures Actual Sum, YTD Actual Sum, Prior Year Actual Sum, and Prior Year YTD Actual Sum from the Finance Data table to Values.
- Insert a Slicer on the Fiscal Year from the Date table.
- Select FY2016 in the Slicer.



Month	Actual Sum	YTD Actual Sum	Prior Year Actual Sum	Prior YTD Actual Sum
01-Jul	$116,000,024	$116,000,024	$76,206,062	$76,206,062
02-Aug	$76,139,923	$192,139,947	$112,698,660	$188,904,722
03-Sep	$146,049,690	$338,189,637	$184,058,629	$372,963,351
04-Oct	$135,023,735	$473,213,372	$131,955,600	$504,918,951
05-Nov	$139,894,342	$613,107,714	$131,606,546	$636,525,497
06-Dec	$36,855,259	$649,962,973	$132,674,885	$769,200,382
07-Jan	$0	$649,962,973	$151,595,834	$920,796,216
08-Feb	$0	$649,962,973	$121,984,079	$1,042,780,295
09-Mar	$0	$649,962,973	$121,325,029	$1,164,105,324
10-Apr	$0	$649,962,973	$130,660,676	$1,294,766,000
11-May	$0	$649,962,973	$139,629,594	$1,434,395,594
12-Jun	$0	$649,962,973	$249,672,178	$1,684,067,771
Grand Total	$649,962,973	$649,962,973	$1,684,067,771	$1,684,067,771

Quarter-to-Date Measures and Analysis

To calculate a result that includes a starting balance from the beginning of a period, such as a fiscal quarter, up to a specific period in time, you can use DAX Time Intelligence functions. This will enable you to analyze data on a month level.

In this chapter, you will learn how to create Quarter-to-Date measures and how to carry out data analysis with the same.

Creating Quarter-to-Date Sum Measure

Create Quarter-to-Date Actual Sum measure as follows −

QTD Actual Sum:=TOTALQTD([Actual Sum],'Date'[Date],ALL('Date'))

Creating Quarter-to-Date Budget Sum Measure

Create Quarter-to-Date Budget Sum measure as follows −

QTD Budget Sum:=TOTALQTD([Budget Sum], 'Date'[Date], ALL('Date'))

Creating Quarter-to-Date Forecast Sum Measure

Create Quarter-to-Date Budget Sum measure as follows −

QTD Budget Sum:=TOTALQTD([Budget Sum], 'Date'[Date], ALL('Date'))

Creating Quarter-to-Date Forecast Sum Measure

Create Quarter-to-Date Forecast Sum measure as follows −

QTD Forecast Sum:=TOTALQTD([Forecast Sum], 'Date'[Date], ALL('Date'))

Data Modeling with DAX

Creating Prior Quarter-to-Date Actual Sum Measure

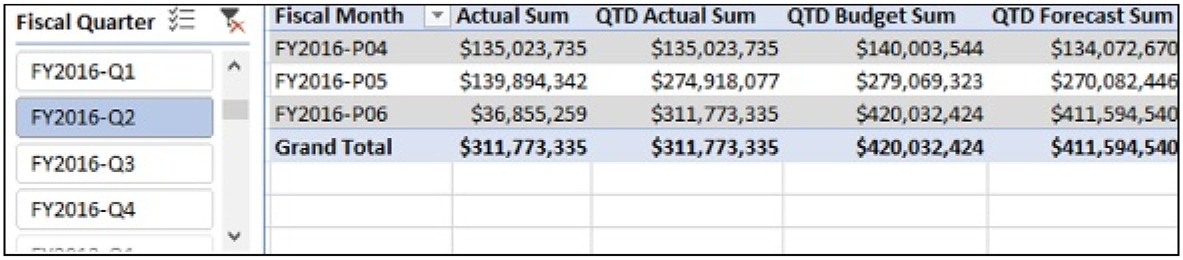
Create Prior Quarter-to-Date Actual Sum measure as follows −

Prior QTD Actual Sum:=TOTALQTD([Prior Quarter Actual Sum], 'Date'[Date], ALL('Date'))

Analyzing Data with Quarter-to-Date Measures

Create a Power PivotTable as follows −

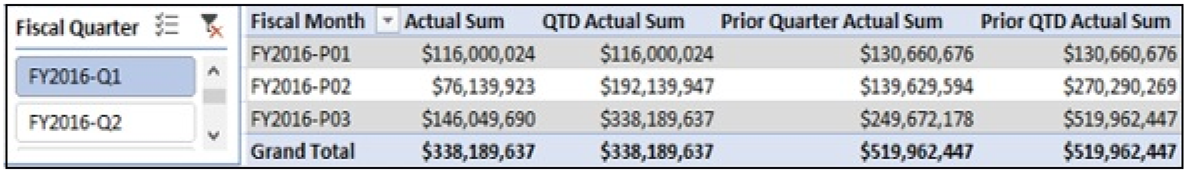
- Add Fiscal Month from Date table to Rows.
- Add the measures Actual Sum, QTD Actual Sum, QTD Budget Sum, and QTD Forecast Sum from Finance Data table to Values.
- Insert a Slicer on the Fiscal Quarter from the Date table.
- Select FY2016-Q2 in the Slicer.



Fiscal Month	Actual Sum	QTD Actual Sum	QTD Budget Sum	QTD Forecast Sum
FY2016-P04	$135,023,735	$135,023,735	$140,003,544	$134,072,670
FY2016-P05	$139,894,342	$274,918,077	$279,069,323	$270,082,446
FY2016-P06	$36,855,259	$311,773,335	$420,032,424	$411,594,540
Grand Total	**$311,773,335**	**$311,773,335**	**$420,032,424**	**$411,594,540**

Create a Power PivotTable as follows −

- Add Fiscal Month from Date table to Rows.
- Add the measures Actual Sum, QTD Actual Sum, Prior Quarter Actual Sum, and Prior QTD Actual Sum from Finance Data table to Values.
- Insert a Slicer on the Fiscal Quarter from Date table.
- Select FY2016-Q1 in the Slicer.



Fiscal Month	Actual Sum	QTD Actual Sum	Prior Quarter Actual Sum	Prior QTD Actual Sum
FY2016-P01	$116,000,024	$116,000,024	$130,660,676	$130,660,676
FY2016-P02	$76,139,923	$192,139,947	$139,629,594	$270,290,269
FY2016-P03	$146,049,690	$338,189,637	$249,672,178	$519,962,447
Grand Total	$338,189,637	$338,189,637	$519,962,447	$519,962,447

Budget Measures and Analysis

Budgeting involves estimating the cash flows of a company over a financial year. The financial position of the company, its goals, expected revenues, and expenses are taken into account in budgeting.

However, the market conditions may change during the financial year and the company may have to reset its goals. This requires analyzing the financial data with the budget estimated at the beginning of the financial year (Budget Sum) and the actual expended sum from the beginning of the financial year to date (YTD Actual Sum).

At any time during a financial year, you can calculate the following −

Unexpended Balance

Unexpended Balance is the budget remaining after the actual expenses, i.e.

Unexpended Balance = YTD Budget Sum – YTD Actual Sum

Budget Attainment %

Budget Attainment % is the percentage of the budget that you have spent to date, i.e.

Budget Attainment % = YTD Actual Sum/YTD Budget Sum

These calculations help those companies that use budgeting to make decisions.

Creating Unexpended Balance Measure

You can create Unexpended Balance measure as follows −

Unexpended Balance:=CALCULATE([YTD Budget Sum],ALL('Finance Data'[Date]))-[YTD Actual Sum]

Creating Budget Attainment Percentage Measure

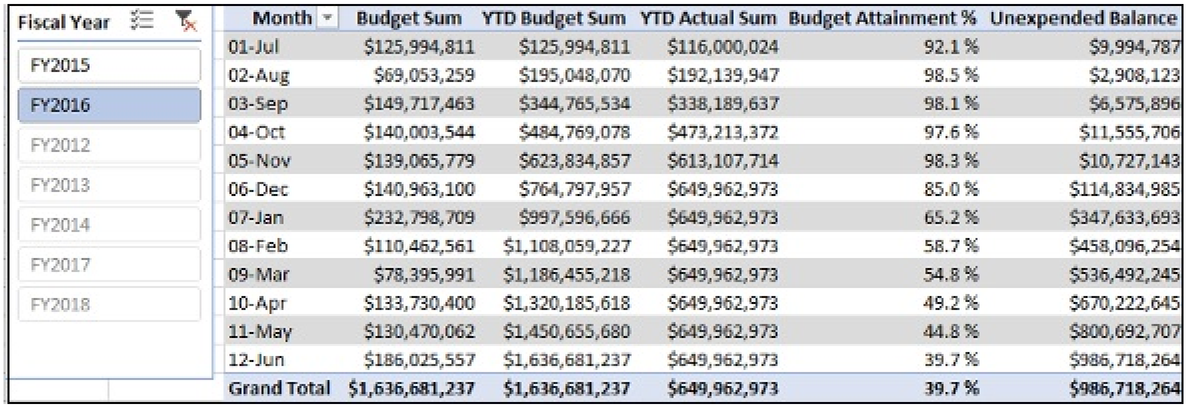
You can create Budget Attainment Percentage measure as follows −

Budget Attainment %:=IF([YTD Budget Sum],[YTD Actual Sum]/CALCULATE([YTD Budget Sum],ALL('Finance Data'[Date])),BLANK())

Analyzing Data with Budget Measures

Create a Power PivotTable as follows −

- Add Month from the Date table to Rows.
- Add the measures Budget Sum, YTD Budget Sum, YTD Actual Sum, Budget Attainment % and Unexpended Balance from Finance Data table to Values.
- Insert a Slicer on the Fiscal Year field.
- Select FY2016 in the Slicer.



Month	Budget Sum	YTD Budget Sum	YTD Actual Sum	Budget Attainment %	Unexpended Balance
01-Jul	$125,994,811	$125,994,811	$116,000,024	92.1 %	$9,994,787
02-Aug	$69,053,259	$195,048,070	$192,139,947	98.5 %	$2,908,123
03-Sep	$149,717,463	$344,765,534	$338,189,637	98.1 %	$6,575,896
04-Oct	$140,003,544	$484,769,078	$473,213,372	97.6 %	$11,555,706
05-Nov	$139,065,779	$623,834,857	$613,107,714	98.3 %	$10,727,143
06-Dec	$140,963,100	$764,797,957	$649,962,973	85.0 %	$114,834,985
07-Jan	$232,798,709	$997,596,666	$649,962,973	65.2 %	$347,633,693
08-Feb	$110,462,561	$1,108,059,227	$649,962,973	58.7 %	$458,096,254
09-Mar	$78,395,991	$1,186,455,218	$649,962,973	54.8 %	$536,492,245
10-Apr	$133,730,400	$1,320,185,618	$649,962,973	49.2 %	$670,222,645
11-May	$130,470,062	$1,450,655,680	$649,962,973	44.8 %	$800,692,707
12-Jun	$186,025,557	$1,636,681,237	$649,962,973	39.7 %	$986,718,264
Grand Total	$1,636,681,237	$1,636,681,237	$649,962,973	39.7 %	$986,718,264

Forecast Measures and Analysis

You can use Forecast measures to analyze the finance data and help an organization make necessary adjustments in its goals and targets for the year, to align the company's performance to the changing business requirements.

You need to update the forecasts regularly to keep up with the changes. You can then compare the most recent forecast to the budget for the rest of the period in the financial year so that the company can make the required adjustments to meet the business changes.

At any time during a financial year, you can calculate the following −

Forecast Attainment %

Forecast Attainment % is the percentage of the forecast sum that you have spent to date, i.e.

Forecast Attainment % = YTD Actual Sum/YTD Forecast Sum

Forecast Unexpended Balance

Forecast Unexpended Balance is the Forecast Sum remaining after the actual expenses, i.e

Forecast Unexpended Balance = YTD Forecast Sum – YTD Actual Sum

Budget Adjustment

Budget Adjustment is the adjustment in the budget sum an organization needs to make (an increase or decrease) based on the forecast.

Budget Adjustment = Forecast Unexpended Balance - Unexpended Balance

The budget needs to be increased if the resulting value is positive. Otherwise, it can be adjusted for some other purpose.

Data Modeling with DAX

Creating Forecast Attainment Percentage Measure

You can create Forecast Attainment Percentage measure as follows −

Forecast Attainment Percentage:= IF([YTD Forecast Sum], [YTD Actual Sum]/[YTD Forecast Sum], BLANK())

Creating Forecast Unexpended Balance Measure

You can create Forecast Unexpended Balance measure as follows −

Forecast Unexpended Balance:=[YTD Forecast Sum]-[YTD Actual Sum]

Creating Budget Adjustment Measure

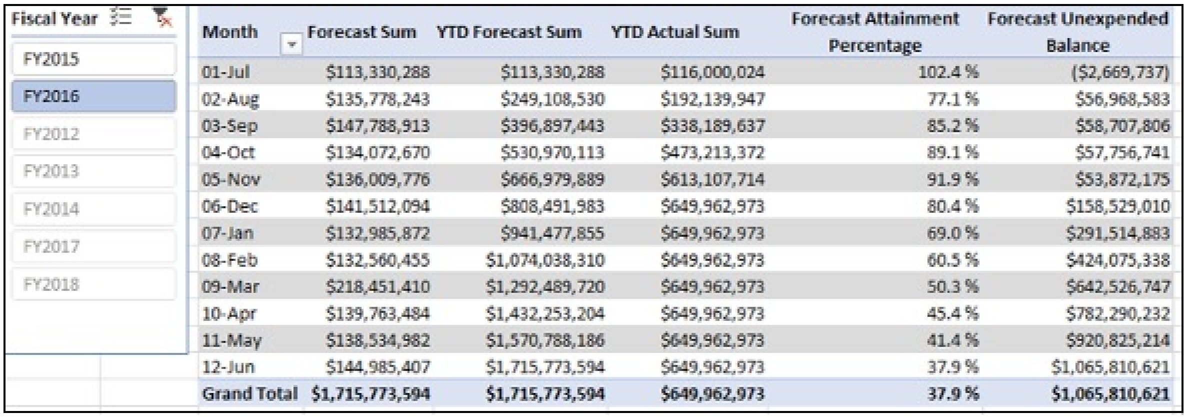
You can create Budget Adjustment measure as follows −

Budget Adjustment:=[Forecast Unexpended Balance]-[Unexpended Balance]

Analyzing Data with Forecast Measures

Create a Power PivotTable as follows −

- Add Month from Date table to Rows.
- Add the measures Budget Sum, YTD Budget Sum, YTD Actual Sum, Budget Attainment % and Unexpended Balance from Finance Data table to Values.
- Insert a Slicer on Fiscal Year.
- Select FY2016 in the Slicer.



Month	Forecast Sum	YTD Forecast Sum	YTD Actual Sum	Forecast Attainment Percentage	Forecast Unexpended Balance
01-Jul	$113,330,288	$113,330,288	$116,000,024	102.4 %	($2,669,737)
02-Aug	$135,778,243	$249,108,530	$192,139,947	77.1 %	$56,968,583
03-Sep	$147,788,913	$396,897,443	$338,189,637	85.2 %	$58,707,806
04-Oct	$134,072,670	$530,970,113	$473,213,372	89.1 %	$57,756,741
05-Nov	$136,009,776	$666,979,889	$613,107,714	91.9 %	$53,872,175
06-Dec	$141,512,094	$808,491,983	$649,962,973	80.4 %	$158,529,010
07-Jan	$132,985,872	$941,477,855	$649,962,973	69.0 %	$291,514,883
08-Feb	$132,560,455	$1,074,038,310	$649,962,973	60.5 %	$424,075,338
09-Mar	$218,451,410	$1,292,489,720	$649,962,973	50.3 %	$642,526,747
10-Apr	$139,763,484	$1,432,253,204	$649,962,973	45.4 %	$782,290,232
11-May	$138,534,982	$1,570,788,186	$649,962,973	41.4 %	$920,825,214
12-Jun	$144,985,407	$1,715,773,594	$649,962,973	37.9 %	$1,065,810,621
Grand Total	$1,715,773,594	$1,715,773,594	$649,962,973	37.9 %	$1,065,810,621

Count of Months Measures

Data Modeling with DAX

You can create the Count of Months measures that can be used in creating Headcount measures and Cost Per Head measures. These measures count the distinct values of Fiscal Month column where the Actual column / Budget column / Forecast column has non-zero values in the Finance Data table. This is required because the Finance Data table contains zero values in the Actual column and those rows are to be excluded while calculating Headcount and Cost per Head.

Creating Count of Actual Months Measure

You can create Count of Actual Months measure as follows −

CountOfActualMonths:=CALCULATE(DISTINCTCOUNT('FinanceData' [Fiscal Month]),'Finance Data'[Actual]<>0)

Creating Count of Budget Months Measure

You can create Count of Budget Months measure as follows −

CountOfBudgetMonths:=CALCULATE(DISTINCTCOUNT('FinanceData' [Fiscal Month]),'Finance Data'[Budget]<>0)

Creating Count of Forecast Months Measure

You can create Count of Forecast Months measure as follows −

CountOfForecastMonths:=CALCULATE(DISTINCTCOUNT('FinanceData' [Fiscal Month]),'Finance Data'[Forecast]<>0)

Ending Headcount Measures

You can create Ending Headcount measures for a specific period of time. The Ending Headcount is the sum of the people as on the last date in the specified period for which we have a non-blank sum of people.

The Ending Headcount is obtained as follows −

- For a Month − Sum of People at the end of the specific Month.
- For a Quarter − Sum of People at the end of the last Month of the specific Quarter.
- For a Year − Sum of People at the end of the last Month of the specific Year.

Creating Actual Ending Headcount Measure

You can create Actual Ending Headcount measure as follows −

Actual Ending Head Count:=CALCULATE(SUM('Finance Data'[Actual People]),LASTNONBLANK('Finance Data'[Date], IF(CALCULATE(SUM('Finance

Data'[Actual People]), ALL(Accounts))=0, BLANK(), CALCULATE(SUM('Finance Data'[Actual People]), ALL(Accounts)))), ALL(Accounts))

DAX LASTNONBLANK function as used above returns the last date for which you have a non-blank sum of people so that you can calculate the sum of people on that date.

Creating Budget Ending Headcount Measure

You can create Budget Ending Headcount measure as follows −

Budget Ending Head Count: = CALCULATE(SUM('Finance Data'[Budget People]),LASTNONBLANK('Finance Data'[Date], IF(CALCULATE(SUM('Finance Data'[Budget People]), ALL(Accounts))=0, BLANK(), CALCULATE(SUM('Finance Data'[Budget People]), ALL(Accounts)))), ALL(Accounts))

Creating Forecast Ending Headcount Measure

You can create Forecast Ending Headcount measure as follows −

Forecast Ending Head Count:= CALCULATE(SUM('Finance Data'[Forecast People]), LASTNONBLANK('Finance Data'[Date], IF(CALCULATE(SUM('Finance Data'[Forecast People]), ALL(Accounts))=0, BLANK(),CALCULATE(SUM('Finance Data'[Forecast People]), ALL(Accounts)))), ALL(Accounts))

Creating Prior Year Actual Ending Headcount Measuree

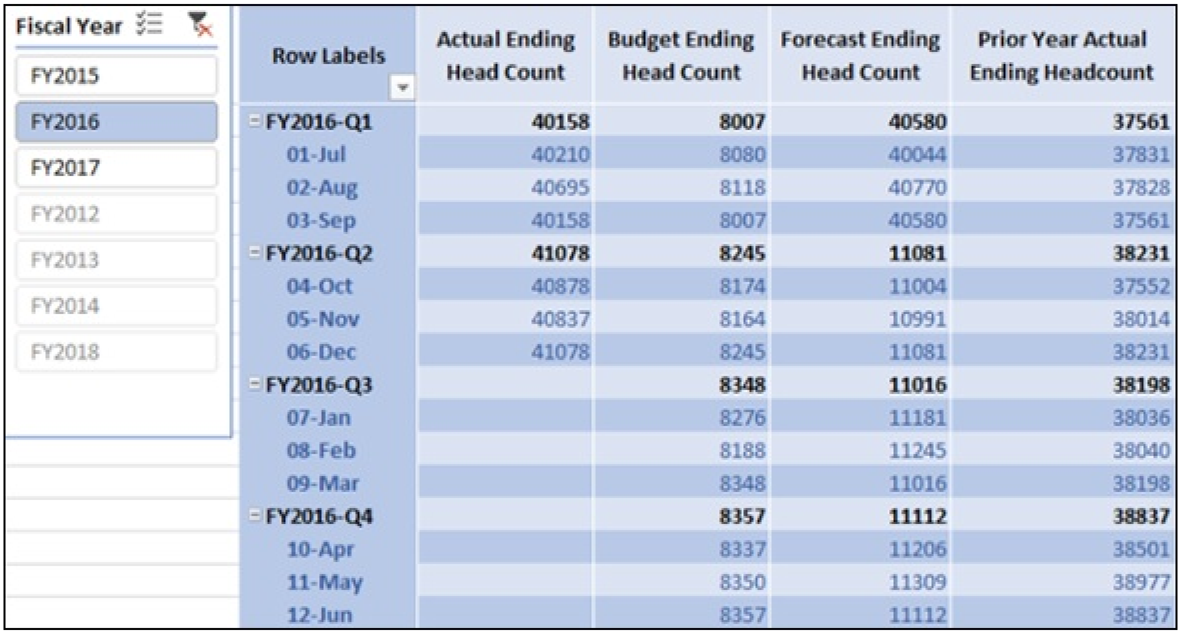
You can create Prior Year Actual Ending Headcount measure as follows −

Prior Year Actual Ending Headcount:=CALCULATE('Finance Data'[Actual Ending Head Count], DATEADD('Date'[Date],-1,YEAR))

Analyzing Data with Ending Headcount Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Year and Month from the Date table to Rows.
- Add the measures Actual Ending Headcount, Budget Ending Headcount, Forecast Ending Headcount, Prior Year Actual Ending Headcount from Finance Data table to Values.
- Insert a Slicer on the Fiscal Year field.
- Select FY2016 in the Slicer.



Row Labels	Actual Ending Head Count	Budget Ending Head Count	Forecast Ending Head Count	Prior Year Actual Ending Headcount
FY2016-Q1	**40158**	**8007**	**40580**	**37561**
01-Jul	40210	8080	40044	37831
02-Aug	40695	8118	40770	37828
03-Sep	40158	8007	40580	37561
FY2016-Q2	**41078**	**8245**	**11081**	**38231**
04-Oct	40878	8174	11004	37552
05-Nov	40837	8164	10991	38014
06-Dec	41078	8245	11081	38231
FY2016-Q3		**8348**	**11016**	**38198**
07-Jan		8276	11181	38036
08-Feb		8188	11245	38040
09-Mar		8348	11016	38198
FY2016-Q4		**8357**	**11112**	**38837**
10-Apr		8337	11206	38501
11-May		8350	11309	38977
12-Jun		8357	11112	38837

Average Headcount Measures

In the previous chapter, you have learnt how to calculate ending headcounts for a specific period. Likewise, you can create the average monthly headcount for any given selection of months.

The Average Monthly Headcount is the sum of the monthly headcounts divided by the number of months in the selection.

You can create these measures using DAX AVERAGEX function.

Creating Actual Average Headcount Measure

You can create Actual Average Headcount measure as follows −

Actual Average Headcount:=AVERAGEX(VALUES('Finance Data'[Fiscal Month]), [Actual Ending Head Count])

Creating Budget Average Headcount Measure

You can create Actual Average Headcount measure as follows −

Budget Average Headcount:=AVERAGEX(VALUES('Finance Data'[Fiscal Month]), [Budget Ending Head Count])

Creating Forecast Average Headcount Measure

You can create Forecast Average Headcount measure as follows −

Forecast Average Headcount:=AVERAGEX(VALUES('Finance Data'[Fiscal Month]), [Actual Ending Head Count])

Creating Prior Year Actual Average Headcount Measure

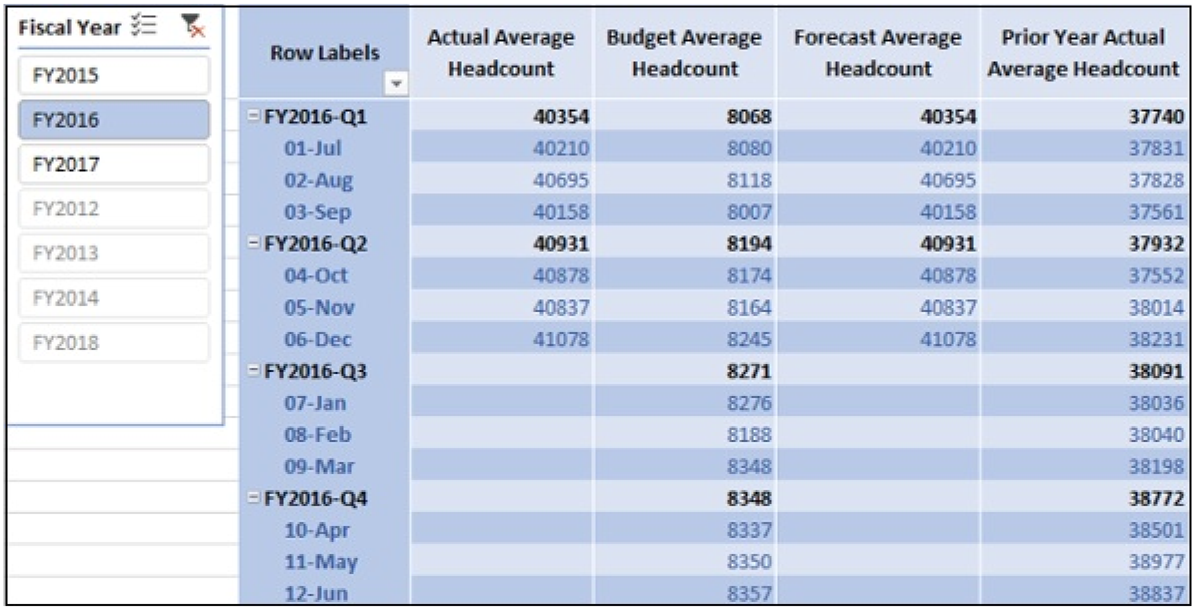
You can create Prior Year Actual Average Headcount measure as follows −

Prior Year Actual Average Headcount:=CALCULATE('Finance Data'[Actual Average Headcount], DATEADD('Date'[Date], -1, YEAR))

Analyzing Data with Average Headcount Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Year and Month from the Date table to Rows.
- Add the measures Actual Average Headcount, Budget Average Headcount, Forecast Average Headcount, Prior Year Actual Average Headcount from Finance Data table to Values.
- Insert a Slicer on the Fiscal Year field.
- Select FY2016 in the Slicer.



Row Labels	Actual Average Headcount	Budget Average Headcount	Forecast Average Headcount	Prior Year Actual Average Headcount
FY2016-Q1	**40354**	**8068**	**40354**	**37740**
01-Jul	40210	8080	40210	37831
02-Aug	40695	8118	40695	37828
03-Sep	40158	8007	40158	37561
FY2016-Q2	**40931**	**8194**	**40931**	**37932**
04-Oct	40878	8174	40878	37552
05-Nov	40837	8164	40837	38014
06-Dec	41078	8245	41078	38231
FY2016-Q3		**8271**		**38091**
07-Jan		8276		38036
08-Feb		8188		38040
09-Mar		8348		38198
FY2016-Q4		**8348**		**38772**
10-Apr		8337		38501
11-May		8350		38977
12-Jun		8357		38837

Total Headcount Measures

In the previous chapters, you have learnt how to create Count of Months measures and Average Headcount measures. You can use these measures to calculate the base Headcount Measures −

- Actual Total Headcount

- Budget Total Headcount
- Forecast Total Headcount

In the subsequent chapters, you will learn how to use these base Headcount measures in other calculations such as YoY Headcount and Variance measures.

Creating Actual Total Headcount Measure

You can create Actual Total Headcount Measure as follows −

Actual Total Head Count:= 'Finance Data'[Actual Average Headcount]*'Finance Data'[CountOfActualMonths]

Creating Budget Total Headcount Measure

You can create Budget Total Headcount Measure as follows −

Budget Total Headcount:= 'Finance Data'[Budget Average Headcount]*'Finance Data'[CountOfBudgetMonths]

Creating Forecast Total Headcount Measure

You can create Forecast Total Headcount Measure as follows −

Forecast Total Headcount:= 'Finance Data'[Forecast Average Headcount]*'Finance Data'[CountOfForecastMonths]

YoY Headcount Measures and Analysis

In the previous chapter, you have learnt how to create base Headcount measures – i.e. Actual Total Headcount, Budget Total Headcount, and Forecast Total Headcount.

In this chapter, you will learn how to create Year-Over-Year Headcount measures and how you can analyze the data with these measures.

Creating Year-over-Year Actual Ending Headcount Measure

You can create Year-over-Year Actual Ending Headcount Measure as follows −

YoY Actual Ending Headcount:=[Actual Ending Head Count]-[Prior Year Actual Ending Headcount]

Creating Year-over-Year Actual Average Headcount Measure

You can create Year-over-Year Actual Average Headcount Measure as follows −

YoY Actual Average Headcount:= [Actual Average Headcount]-[Prior Year Actual Average Headcount]

Creating Year-over-Year Actual Total Headcount Measure

You can create Year-over-Year Actual Total Headcount Measure as follows −

YoY Actual Total Headcount:=[Actual Total Head Count]-[Prior Year Actual Total Headcount]

Analyzing Data with Year-over-Year Actual Headcount Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Quarter and Month from the Date table to Rows.
- Add the measures – Actual Ending Head Count, Prior Year Actual Ending Head Count, YoY Actual Ending Head Count to Values.
- Insert a Slicer on the field Fiscal Year.
- Select FY2016 in the Slicer.

Create another Power PivotTable on the same worksheet as follows −

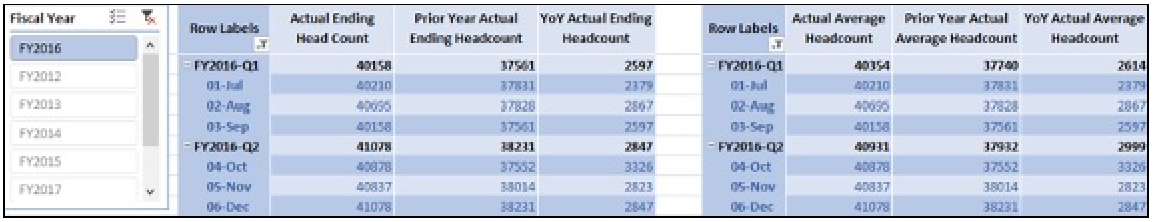
- Add the fields Fiscal Quarter and Month from the Date table to Rows.
- Add the measures – Actual Average Head Count, Prior Year Actual Average Head Count, YoY Actual Average Head Count to Values.

Connect the Slicer to this PivotTable as follows −

- Click the Slicer.
- Click the Options tab under Slicer Tools on the Ribbon.
- Click Report Connections.

Report Connections dialog box appears.

- Select the above two PivotTables.
- Click OK.



Fiscal Year

FY2016
FY2012
FY2013
FY2014
FY2015
FY2017

Row Labels	Actual Ending Head Count	Prior Year Actual Ending Headcount	YoY Actual Ending Headcount
FY2016-Q1	**40158**	**37561**	**2597**
01-Jul	40210	37831	2379
02-Aug	40695	37828	2867
03-Sep	40158	37561	2597
FY2016-Q2	**41078**	**38231**	**2847**
04-Oct	40878	37552	3326
05-Nov	40837	38014	2823
06-Dec	41078	38231	2847

Row Labels	Actual Average Headcount	Prior Year Actual Average Headcount	YoY Actual Average Headcount
FY2016-Q1	**40354**	**37740**	**2614**
01-Jul	40210	37831	2379
02-Aug	40695	37828	2867
03-Sep	40158	37561	2597
FY2016-Q2	**40931**	**37932**	**2999**
04-Oct	40878	37552	3326
05-Nov	40837	38014	2823
06-Dec	41078	38231	2847

Creating Year-over-Year Budget Ending Headcount Measure

You can create Year-over-Year Budget Ending Headcount Measure as follows −

YoY Budget Ending Headcount:= [Budget Ending Head Count]-[Prior Year Actual Ending Headcount]

Creating Year-over-Year Budget Average Headcount Measure

You can create Year-over-Year Budget Average Headcount Measure as follows −

YoY Budget Average Headcount:= [Budget Average Headcount]-[Prior Year Actual Average Headcount]

Creating Year-over-Year Budget Total Headcount Measure

You can create Year-over-Year Budget Total Headcount Measure as follows −

YoY Budget Total Headcount:=[Budget Total Headcount]-[Prior Year Actual Total Headcount]

Creating Year-over-Year Forecast Ending Headcount Measure

You can create Year-over-Year Forecast Ending Headcount Measure as follows −

YoY Forecast Ending Headcount:= [Forecast Ending Head Count]-[Prior Year Actual Ending Headcount]

Creating Year-over-Year Forecast Average Headcount Measure

You can create Year-over-Year Forecast Average Headcount Measure as follows −

YoY Forecast Average Headcount:= [Forecast Average Headcount]-[Prior Year Actual Average Headcount]

Creating Year-over-Year Forecast Total Headcount Measure

You can create Year-over-Year Forecast Total Headcount Measure as follows −

YoY Forecast Total Headcount:=[Forecast Total Headcount]-[Prior Year Actual Total Headcount]

Variance Headcount Measures

You can create the Variance Headcount measures based on the Headcount measures that you have created so far.

Creating Variance to Budget Ending Headcount Measure

You can create Variance to Budget Ending Headcount Measure as follows −

VTB Ending Head Count:= 'Finance Data'[Budget Ending Head Count]-'Finance Data'[Actual Ending Head Count]

Creating Variance to Budget Average Headcount Measure

You can create Variance to Budget Average Headcount Measure as follows −

VTB Average Head Count:= 'Finance Data'[Budget Average Headcount]-'Finance Data'[Actual Average Headcount

Creating Variance to Budget Total Headcount Measure

You can create Variance to Budget Total Headcount Measure as follows −

VTB Total Head Count:= 'Finance Data'[Budget Total Headcount]-'Finance Data'[Actual Total Head Count]

Creating Variance to Forecast Ending Headcount Measure

You can create Variance to Forecast Ending Headcount Measure as follows −

VTF Ending Head Count:= 'Finance Data'[Forecast Ending Head Count]-'Finance Data'[Actual Ending Head Count]

Creating Variance to Forecast Average Headcount Measure

You can create Variance to Forecast Average Headcount Measure as follows −

VTF Average Head Count:= 'Finance Data'[Forecast Average Headcount]-'Finance Data'[Actual Average Headcount]

Creating Variance to Forecast Total Headcount Measure

You can create Variance to Forecast Total Headcount Measure as follows −

VTF Total Head Count:= 'Finance Data'[Forecast Total Headcount]-'Finance Data'[Actual Total Head Count]

Data Modeling with DAX

Creating Forecast Variance to Budget Ending Headcount Measure

You can create Forecast Variance to Budget Ending Headcount Measure as follows −

Forecast VTB Ending Head Count:= 'Finance Data'[Budget Ending Head Count]-'Finance Data'[Forecast Ending Head Count]

Creating Forecast Variance to Budget Average Headcount Measure

You can create Forecast Variance to Budget Average Headcount Measure as follows −

Forecast VTB Average Head Count:='Finance Data'[Budget Average Headcount]-'Finance Data'[Forecast Average Headcount]

Creating Forecast Variance to Budget Total Headcount Measure

You can create Forecast Variance to Budget Total Headcount Measure as follows −

Forecast VTB Total Head Count:= 'Finance Data'[Budget Total Headcount]-'Finance Data'[Forecast Total Headcount

Cost Per Headcount Measures and Analysis

You have learnt about the two major categories of Measures −

- Finance Measures.
- Headcount Measures.

The third major category of measures that you will learn is People Cost Measures. Any organization will be interested to know the annualized cost per head. Annualized cost per head represents the cost to the company of having one employee on a full year basis.

To create Cost Per Head measures, you need to first create certain preliminary People Cost Measures. In the Accounts table, you have a column – Sub Class that contains People as one of the values. Hence, you can apply a filter on the Accounts table on the Sub Class column to obtain the filter context onto the Finance Data table to obtain People Cost.

You can use thus obtain People Cost measures and Count of Months measures to create Annualized People Cost measures. You can finally create Annualized Cost Per Head measures from Annualized People Cost measures and Average Head Count measures.

Creating Actual People Cost Measure

You can create Actual People Cost measure as follows −

Actual People Cost:=CALCULATE('Finance Data'[Actual Sum], FILTER('Finance Data', RELATED(Accounts[Sub Class])="People"))

Creating Budget People Cost Measure

You can create Budget People Cost measure as follows −

Budget People Cost:=CALCULATE('Finance Data'[Budget Sum], FILTER('Finance Data', RELATED(Accounts[Sub Class])="People"))

Creating Forecast People Cost Measure

You can create Forecast People Cost measure as follows −

Forecast People Cost:=CALCULATE('Finance Data'[Forecast Sum], FILTER('Finance Data', RELATED(Accounts[Sub Class])="People"))

Creating Annualized Actual People Cost Measure

You can create Annualized Actual People Cost measure as follows −

*Annualized Actual People Cost:=IF([CountOfActualMonths],[Actual People Cost]*12/[CountOfActualMonths],BLANK())*

Creating Annualized Budget People Cost Measure

You can create Annualized Budget People Cost measure as follows −

*Annualized Budget People Cost:=IF([CountOfBudgetMonths], [Budget People Cost]*12/[CountOfBudgetMonths],BLANK())*

Creating Annualized Forecast People Cost Measure

You can create Annualized Forecast People Cost measure as follows −

*Annualized Forecast People Cost:=IF([CountOfForecastMonths],[Forecast People Cost]*12/[CountOfForecastMonths],BLANK())*

Creating Actual Annualized Cost Per Head Measure

You can create Actual Annualized Cost Per Head (CPH) measure as follows −

Actual Annualized CPH:=IF([Actual Average Headcount], [Annualized Actual People Cost]/[Actual Average Headcount],BLANK())

Data Modeling with DAX

Creating Budget Annualized Cost Per Head Measure

You can create Budget Annualized Cost Per Head (CPH) measure as follows −

Budget Annualized CPH:=IF([Budget Average Headcount],[Annualized Budget People Cost]/[Budget Average Headcount],BLANK())

Creating Forecast Annualized Cost Per Head Measure

You can create Forecast Annualized Cost Per Head (CPH) measure as follows −

Forecast Annualized CPH:=IF([Forecast Average Headcount],[Annualized Forecast People Cost]/[Forecast Average Headcount], BLANK())

Creating Prior Year Actual Annualized Cost Per Head Measure

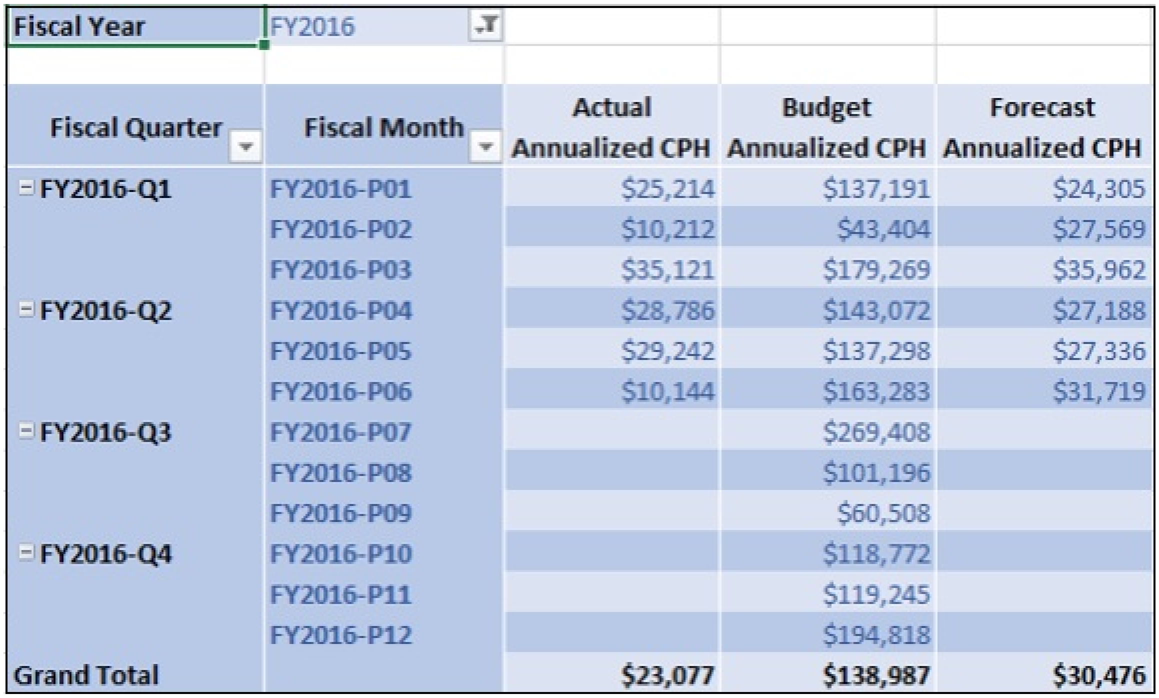
You can create Prior Year Actual Annualized Cost Per Head (CPH) measure as follows −

Prior Year Actual Annualized CPH:=CALCULATE([Actual Annualized CPH], DATEADD('Date'[Date],-1,YEAR))

Analyzing Data with Cost Per Head Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Quarter and Fiscal Month from Date table to Rows.
- Add the measures Actual Annualized CPH, Budget Annualized CPH, and Forecast Annualized CPH to Columns.
- Add the field Fiscal Year from Date table to Filters.
- Select FY2016 in the Filter.

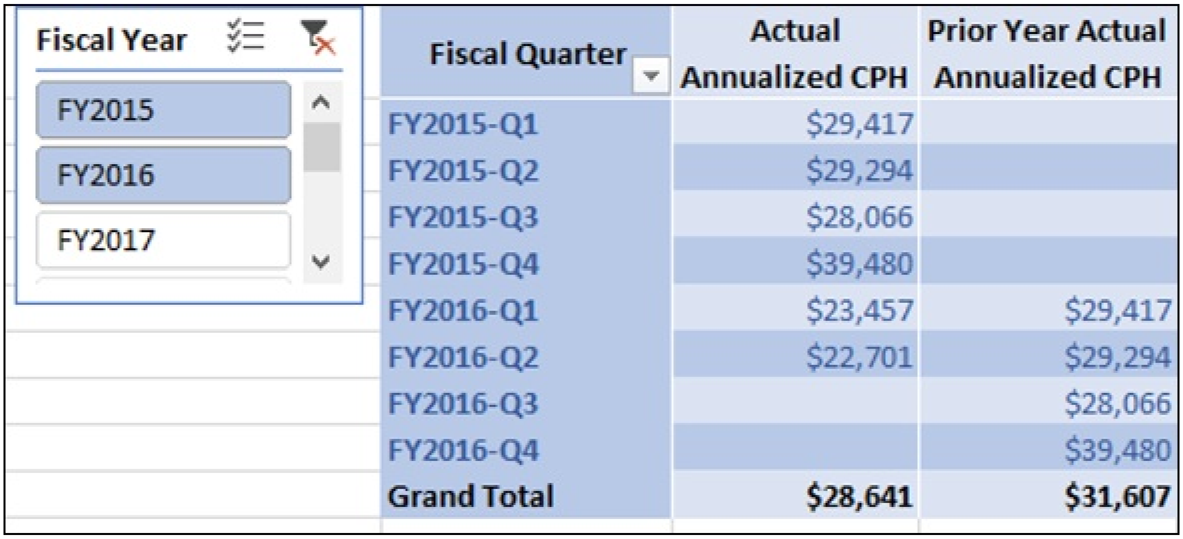


Fiscal Year | FY2016

Fiscal Quarter	Fiscal Month	Actual Annualized CPH	Budget Annualized CPH	Forecast Annualized CPH
FY2016-Q1	FY2016-P01	$25,214	$137,191	$24,305
	FY2016-P02	$10,212	$43,404	$27,569
	FY2016-P03	$35,121	$179,269	$35,962
FY2016-Q2	FY2016-P04	$28,786	$143,072	$27,188
	FY2016-P05	$29,242	$137,298	$27,336
	FY2016-P06	$10,144	$163,283	$31,719
FY2016-Q3	FY2016-P07		$269,408	
	FY2016-P08		$101,196	
	FY2016-P09		$60,508	
FY2016-Q4	FY2016-P10		$118,772	
	FY2016-P11		$119,245	
	FY2016-P12		$194,818	
Grand Total		$23,077	$138,987	$30,476

Create another Power PivotTable as follows −

- Add the field Fiscal Quarter from Date table to Rows.
- Add the measures Actual Annualized CPH, and Prior Year Actual Annualized CPH to Columns.
- Insert a Slicer on the field Fiscal Year from Date table.
- Select FY2015 and FY2016 on the Slicer.



Fiscal Year

FY2015

FY2016

FY2017

Fiscal Quarter	Actual Annualized CPH	Prior Year Actual Annualized CPH
FY2015-Q1	$29,417	
FY2015-Q2	$29,294	
FY2015-Q3	$28,066	
FY2015-Q4	$39,480	
FY2016-Q1	$23,457	$29,417
FY2016-Q2	$22,701	$29,294
FY2016-Q3		$28,066
FY2016-Q4		$39,480
Grand Total	$28,641	$31,607

Rate Variance and Volume Variance

Data Modeling with DAX

You have learnt how to create measures for Annualized Cost Per Head and Total Headcount. You can use these measures to create Rate Variance and Volume Variance measures.

- Rate Variance measures calculate what portion of a Currency Variance is caused by differences in Cost Per Head.
- Volume Variance measures calculate how much of the Currency Variance is driven by fluctuation in Headcount.

Creating Variance to Budget Rate Measure

You can create Variance to Budget Rate measure as follows −

VTB Rate:=([Budget Annualized CPH]/12-[Actual Annualized CPH]/12)[Actual Total Head Count]*

Creating Variance to Budget Volume Measure

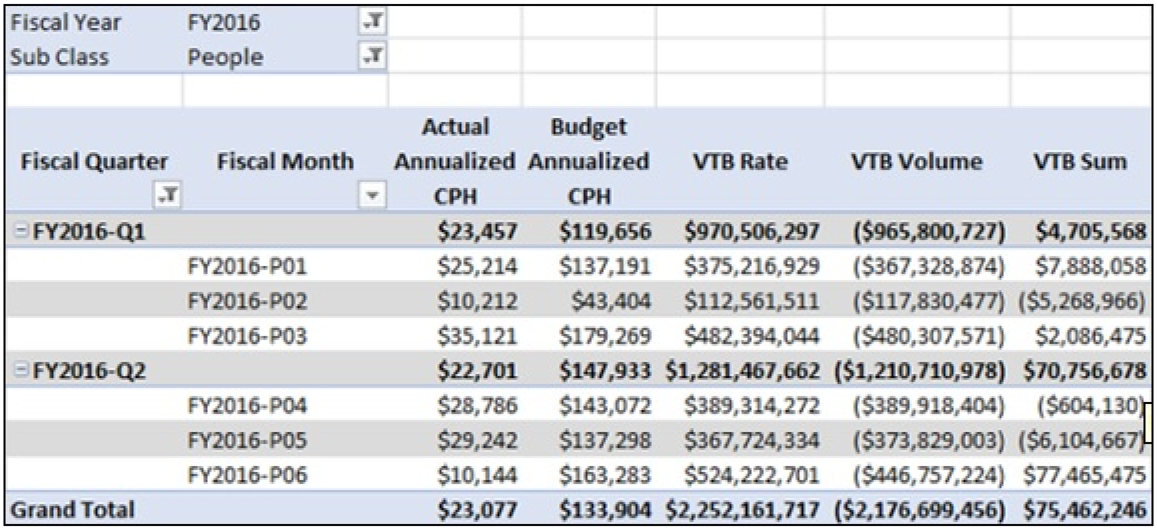
You can create Variance to Budget Volume measure as follows −

VTB Volume:=[VTB Total Head Count][Budget Annualized CPH]/12*

Analyzing Data with Variance to Budget Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Quarter and Fiscal Month from Date table to Rows.
- Add the measures Actual Annualized CPH, Budget Annualized CPH, VTB Rate, VTB Volume, VTB Sum to Values.
- Add the fields Fiscal Year from Date table and Sub Class from Accounts table to Filters.
- Select FY2016 in the Fiscal Year Filter.
- Select People in the Sub Class Filter.
- Filter Row Labels for Fiscal Quarter values FY2016-Q1 and FY2016-Q2.



Fiscal Year	FY2016					
Sub Class	People					

Fiscal Quarter	Fiscal Month	Actual Annualized CPH	Budget Annualized CPH	VTB Rate	VTB Volume	VTB Sum
⊟FY2016-Q1		**$23,457**	**$119,656**	**$970,506,297**	**($965,800,727)**	**$4,705,568**
	FY2016-P01	$25,214	$137,191	$375,216,929	($367,328,874)	$7,888,058
	FY2016-P02	$10,212	$43,404	$112,561,511	($117,830,477)	($5,268,966)
	FY2016-P03	$35,121	$179,269	$482,394,044	($480,307,571)	$2,086,475
⊟FY2016-Q2		**$22,701**	**$147,933**	**$1,281,467,662**	**($1,210,710,978)**	**$70,756,678**
	FY2016-P04	$28,786	$143,072	$389,314,272	($389,918,404)	($604,130)
	FY2016-P05	$29,242	$137,298	$367,724,334	($373,829,003)	($6,104,667)
	FY2016-P06	$10,144	$163,283	$524,222,701	($446,757,224)	$77,465,475
Grand Total		**$23,077**	**$133,904**	**$2,252,161,717**	**($2,176,699,456)**	**$75,462,246**

You can observe the following in the above PivotTable −

- VTB Sum value shown is only for Sub Class – People.
- For Fiscal Quarter FY2016-Q1, VTB Sum is $4,705,568, VTB Rate is $970,506,297, and VTB Volume is $-965,800,727.
- VTB Rate measure calculates that $970,506,297 of the Variance to Budget (VTB Sum) is caused by the difference in Cost per Head, and $-965,800,727 is caused by the difference in Headcount.
- If you add VTB Rate and VTB Volume, you will get $4,705,568, the same value as returned by VTB Sum for Sub Class People.
- Similarly, for Fiscal Quarter FY2016-Q2, VTB Rate is $1,281,467,662, and VTB Volume is $-1,210,710,978. If you add VTB Rate and VTB Volume, you will get $70,756,678, which is the VTB Sum value shown in the PivotTable.

Creating Year-Over-Year Rate Measure

You can create Year-Over-Year Rate measure as follows −

YoY Rate:=([Actual Annualized CPH]/12-[Prior Year Actual Annualized CPH]/12)[Actual Total Head Count]*

Creating Year-Over-Year Volume Measure

You can create Year-Over-Year Volume measure as follows −

YoY Volume:=[YoY Actual Total Headcount][Prior Year Actual Annualized CPH]/12*

Creating Variance to Forecast Rate Measure

You can create Variance to Forecast Rate measure as follows −

VTF Rate:=([Forecast Annualized CPH]/12-[Actual Annualized CPH]/12)[Actual Total Head Count]*

Creating Variance to Forecast Volume Measure

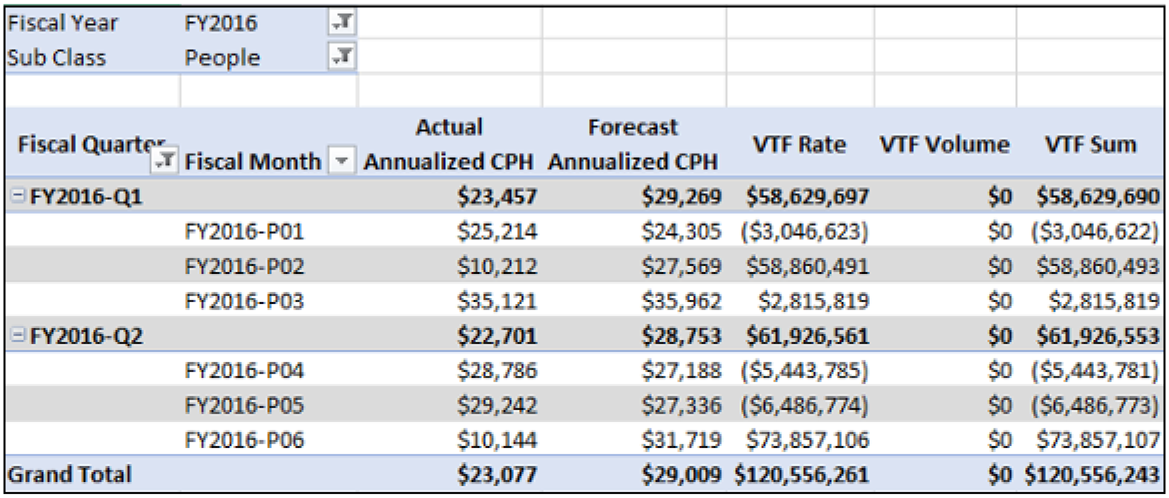
You can create Variance to Forecast Volume measure as follows −

VTF Volume:=[VTF Total Head Count][Forecast Annualized CPH]/12*

Analyzing Data with Variance to Forecast Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Quarter and Fiscal Month from Date table to Rows.
- Add the measures Actual Annualized CPH, Forecast Annualized CPH, VTF Rate, VTF Volume, VTF Sum to Values.
- Add the fields Fiscal Year from Date table and Sub Class from Accounts table to Filters.
- Select FY2016 in the Fiscal Year Filter.
- Select People in the Sub Class Filter.
- Filter Row Labels for Fiscal Quarter values FY2016-Q1 and FY2016-Q2.



Fiscal Year	FY2016
Sub Class	People

Fiscal Quarter	Fiscal Month	Actual Annualized CPH	Forecast Annualized CPH	VTF Rate	VTF Volume	VTF Sum
FY2016-Q1		$23,457	$29,269	$58,629,697	$0	$58,629,690
	FY2016-P01	$25,214	$24,305	($3,046,623)	$0	($3,046,622)
	FY2016-P02	$10,212	$27,569	$58,860,491	$0	$58,860,493
	FY2016-P03	$35,121	$35,962	$2,815,819	$0	$2,815,819
FY2016-Q2		$22,701	$28,753	$61,926,561	$0	$61,926,553
	FY2016-P04	$28,786	$27,188	($5,443,785)	$0	($5,443,781)
	FY2016-P05	$29,242	$27,336	($6,486,774)	$0	($6,486,773)
	FY2016-P06	$10,144	$31,719	$73,857,106	$0	$73,857,107
Grand Total		$23,077	$29,009	$120,556,261	$0	$120,556,243

Creating Forecast Variance to Budget Rate Measure

You can create Forecast Variance to Budget Rate measure as follows −

Forecast VTB Rate:=([Budget Annualized CPH]/12-[Forecast Annualized CPH]/12)[Forecast Total Headcount]*

Creating Forecast Variance to Budget Volume Measure

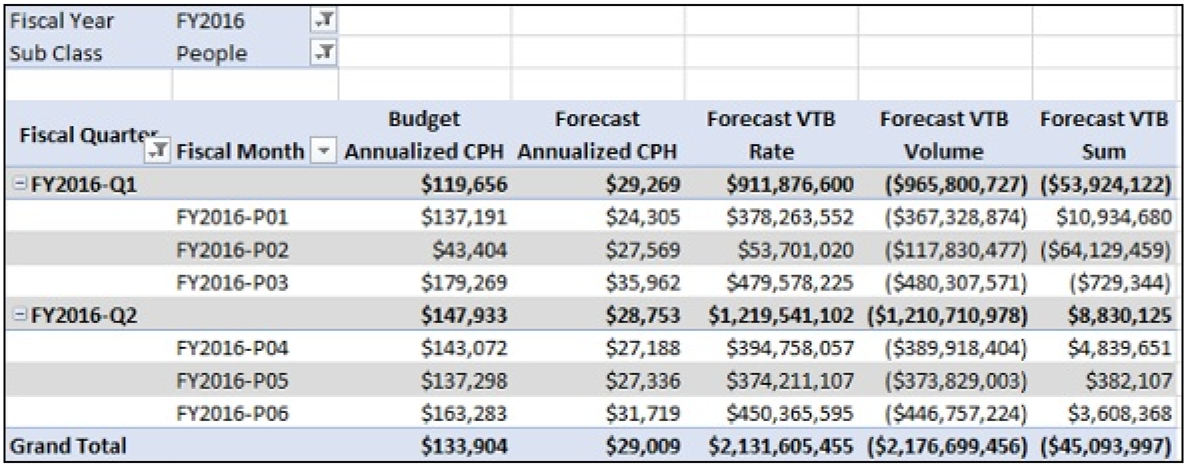
You can create Forecast Variance to Budget Volume measure as follows −

Forecast VTB Volume:=[Forecast VTB Total Head Count][Budget Annualized CPH]/12*

Analyzing Data with Forecast Variance to Budget Measures

Create a Power PivotTable as follows −

- Add the fields Fiscal Quarter and Fiscal Month from Date table to Rows.
- Add the measures Budget Annualized CPH, Forecast Annualized CPH, Forecast VTB Rate, Forecast VTB Volume, Forecast VTB Sum to Values.
- Add the fields Fiscal Year from Date table and Sub Class from Accounts table to Filters.
- Select FY2016 in the Fiscal Year Filter.
- Select People in the Sub Class Filter.
- Filter Row Labels for Fiscal Quarter values FY2016-Q1 and FY2016-Q2.



Fiscal Year	FY2016
Sub Class	People

Fiscal Quarter	Fiscal Month	Budget Annualized CPH	Forecast Annualized CPH	Forecast VTB Rate	Forecast VTB Volume	Forecast VTB Sum
FY2016-Q1		**$119,656**	**$29,269**	**$911,876,600**	**($965,800,727)**	**($53,924,122)**
	FY2016-P01	$137,191	$24,305	$378,263,552	($367,328,874)	$10,934,680
	FY2016-P02	$43,404	$27,569	$53,701,020	($117,830,477)	($64,129,459)
	FY2016-P03	$179,269	$35,962	$479,578,225	($480,307,571)	($729,344)
FY2016-Q2		**$147,933**	**$28,753**	**$1,219,541,102**	**($1,210,710,978)**	**$8,830,125**
	FY2016-P04	$143,072	$27,188	$394,758,057	($389,918,404)	$4,839,651
	FY2016-P05	$137,298	$27,336	$374,211,107	($373,829,003)	$382,107
	FY2016-P06	$163,283	$31,719	$450,365,595	($446,757,224)	$3,608,368
Grand Total		**$133,904**	**$29,009**	**$2,131,605,455**	**($2,176,699,456)**	**($45,093,997)**