# PBI365

# Data Analytics and Reporting with Power BI and Office 365

***Power BI for Business Analyists and Power Users***

Complying with all applicable copyright laws is the responsibility of the user. All rights reserved. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without express written permission of Critical Path Training, LLC.

Printed and bound in Canada.

Microsoft, Microsoft Press, ActiveX, Excel, FrontPage, Internet Explorer, PowerPoint, SharePoint, Webdings, Windows, and Windows 7 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Course Manual Version 1.0

**About This Training Course** ..... **5**
**About the Hands-on Labs** ..... **7**
GitHub Repository for this course ..... 7
Student Workstation Prerequisites ..... 7
Copying the Student Lab Files to the Student VM ..... 7
**Module 01: Introduction to the Power BI Platform** ..... **9**
**Module 01 Lab: Getting Up and Running with the Power BI Service** ..... **39**
Exercise 1: Create a new Office 365 Trial Tenant ..... 39
Exercise 2: Upload a Workbook with Sample Data to OneDrive for Business ..... 44
Exercise 3: Configure Multiple Users with Power BI Pro Licenses ..... 46
Exercise 4: Use the Power BI Service to Import a New Dataset ..... 50
Exercise 5: Create a New Power BI Report with Multiple Pages ..... 52
Exercise 6: Create and Share a Power BI Dashboard ..... 63
**Module 02: Getting Started with Power BI Desktop** ..... **69**
Module 02 Lab: Working with Power BI Desktop ..... 87
Exercise 1: Installing and Configuring Power BI Desktop ..... 87
Exercise 2: Importing Data into Power BI Desktop using Power Query ..... 90
Exercise 3: Modeling Tables in a Dataset using Power Pivot ..... 97
Exercise 4: Create a Report using Power BI Desktop ..... 105
Exercise 5: Publishing a PBIX File to the Power BI Service ..... 115
**Module 03: Using the Power Query Features of Power BI Desktop** ..... **119**
Module 03 Lab: Using the Power Query Features of Power BI Desktop ..... 151
Exercise 1: Importing Data from a SQL Azure Database ..... 151
Exercise 2: Using Power Query to Transform and Reshape Customer Data ..... 155
Exercise 3: Using Power Query to Transform and Reshape Product Data ..... 163
Exercise 4: Using Power Query to Transform and Reshape Product Data ..... 166
Exercise 5: Using Power Query to Transform and Reshape Purchase Data ..... 171
**Module 04: Using the Power Pivot Features of Power BI Desktop** ..... **177**
Module 04 Lab: Using the Power Pivot Features of Power BI Desktop ..... 203
Exercise 1: Hiding and Formatting Columns in the Data Model ..... 203
Exercise 2: Extending the Data Model by Creating Calculated Columns ..... 210
Exercise 3: Extending the Data Model with a Lookup Table ..... 218
Exercise 4: Extending the Data Model by Creating Measures ..... 224
**Module 05: Extending Datasets to Support Time Intelligence** ..... **231**
Module 05 Lab: Extending a Data Model with Time Intelligence ..... 257
Exercise 1: Adding a Calendar Dimension Table ..... 257
Exercise 2: Creating Visuals using the Calendar Table ..... 264
Exercise 3: Creating Measures using DAX Time Intelligence Functions ..... 268
Exercise 4: Creating Measures to Monitor Growth in Sales Revenue ..... 272
**Module 06: Designing Reports in Power BI Desktop** ..... **279**
Module 06 Lab: Designing and Publishing Reports with Power BI Desktop ..... 297
Exercise 1: Create the Sales Revenue Breakdown Report ..... 297
Exercise 2: Create the Product Revenue Breakdown Report ..... 302
Exercise 3: Create the Top 5 Products Report ..... 306
Exercise 4: Create the Top 10 Customers Report ..... 313
Exercise 5: Publish Your Project and Its Reports to the Power BI Service ..... 323
**Module 07: Creating Dashboards for Mobile Devices** ..... **325**
Module 07 Lab: Creating and Deploying Dashboards in Power BI ..... 349
Exercise 1: Create the Product Sales Analysis Dashboard ..... 349
Exercise 2: Share the Product Sales Analysis Dashboard ..... 360
Exercise 3: Create the Regional Sales 2015 Dashboard ..... 363
Exercise 4: Deploy a Dashboard using an Organizational Content Pack ..... 379
**Module 08: Leveraging the BI Features in Excel 2016** ..... **385**
Module 08 Lab: Leveraging the BI Features in Excel 2016 ..... 389
Exercise 1: Creating a New Excel Workbook and Importing Data into the Data Model ..... 389

© Critical Path Training. 2015. All Rights Reserved

Exercise 2: Using the Power Pivot Features of Excel 2016 ..... 405

© Critical Path Training. 2015. All Rights Reserved

# About This Training Course

| | |
|---|---|
| **Course Title:** | Data Analytics and Reporting with Power BI and Office 365 |
| **Course Code:** | PBI365 |
| **Audience:** | Business Analysts, Technical Specialists and BI Enthusiasts |
| **Format:** | Instructor-led training with hands-on labs |
| **Length** | 3 Days |

## Course Description

**Data Analytics and Reporting with Power BI** is 3-day, hands-on training course for business analysts, technical specialist and BI enthusiasts who want to learn about the fundamental concepts and tools used to create custom business solutions using the Business Intelligence (BI) features of Power BI, Office 365, Excel Online, SharePoint Online, Microsoft Excel 2016.

Students will learn about the architecture of the Power BI platform as well as how to get up and running with Power BI Desktop. The course focuses on teaching techniques and a methodology for creating and implementing BI projects containing datasets, reports and dashboards. Students will learn how to use the Power Query features of Power BI Desktop to import and reshape data as it's loaded into the data model of a project. The course also teaches students how to use the Power Pivot features of Power BI Desktop and DAX (Data Analysis Expression language) to implement a project with a data model that has been designed to support rich data analytics and reporting.

After teaching students how to design and implement a rich data model in a Power BI Desktop project using Power Query and Power Pivot, the course then focuses on creating dashboard and reports. Students will learn about the rich set of visualization types that are built into the Power BI reporting platform as well as how to design Power BI reports and dashboards using tables, matrices, graphs, charts and maps.

Along the way, students will learn how to create reports and dashboards for mobile devices. Students will also learn how to design and deploy a custom BI solution to an Office 365 tenancy using Organization Content Packs. The third day of the course introduces the new BI features that have been added to Microsoft Excel 2016 and Excel Online and compares the relative strengths and weaknesses of Excel 2016 when compared to Power BI Desktop as a tool for creating custom BI solutions. The course concludes by examining common approaches for designing and implementing real-world BI solutions using the Power BI service.

## Student Prerequisites

Students should have prior experience working with Microsoft Excel and be familiar with navigating around within the Office 365 and Excel Online environment. It is recommended that students also have prior experience working with relational databases, tabular databases and other related BI technologies.

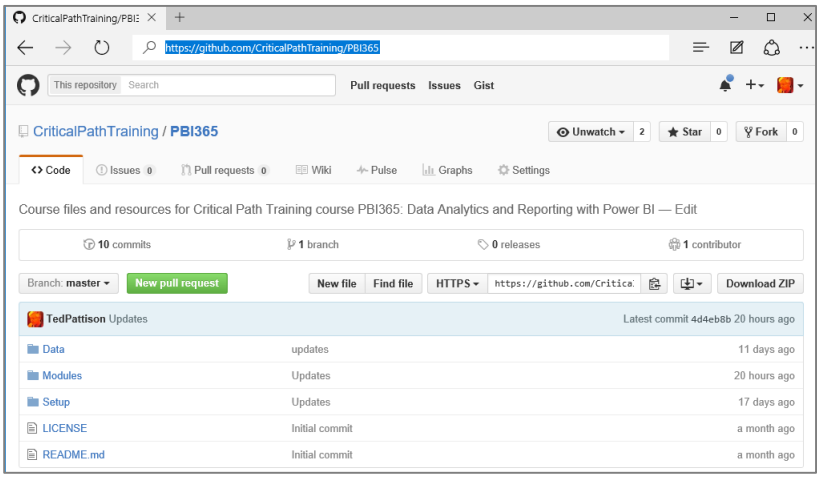
© Critical Path Training. 2015. All Rights Reserved

# About the Hands-on Labs

The courseware for PBI365 includes a rich set of hands-on lab exercises that are designed to complement the learning material that is presented in the lectures. The lab exercises were designed to run on a student workstation running a modern version of Windows such as Windows 8.1 or Windows 10. The student workstation requires the installation or prerequisite software which is listed below on this page. As a student, you may be supplied with a virtual machine running a pre-configured build of the student workstation by the training company that is hosting this training class. Alternatively, you can configure your own computer with the prerequisite software if you are running Windows 8.1 or Windows 10 if you would rather complete the lab exercises on your own computer.

## GitHub Repository for this course

Student files for this course are maintained in a GitHub fil repository at https://github.com/CriticalPathTraining/PBI365.




## Student Workstation Prerequisites

All lab exercises should be completed on a student workstation that has the following software.

1. Windows 8.1 or Windows 10.
2. One or more Internet browsers (Chrome, Edge, IE, etc.)
3. Microsoft Office 2016 Professional including Microsoft Excel 2016
4. Power BI Desktop – which is installed during lab 2.

## Copying the Student Lab Files to the Student VM

There is a set of student lab files that must be copied to our student VM before you can begin to work on the student lab exercises. These student files are distributed in a single ZIP archive known as the **"student.zip"** file which will have a file name that also contains the course code and the version (e.g. **CPT_PBI365_Student_v1.0.zip**). Make sure the version number (e.g. v1.0, v1.1, v2.0, etc.) of the student.zip file matches the version number of this manual.

© Critical Path Training. 2015. All Rights Reserved

# Module 01: Introduction to the Power BI Platform

## Module Description

This module introduces the basic terminology and Business Intelligence features supported by Office 365 and the Power BI platform. Students will learn about the enhanced BI features offered by the Power BI service and the self-service BI tools including Power BI Desktop, Power Query, PowerPivot and Data Analysis Expressions (DAX) Language. The module also discusses Microsoft's strategic vision for BI which includes support for traditional BI as well as new support for the self-service BI model where power users are able to complete BI projects involving data analysis and reporting in a more timely fashion. The module concludes by demonstrating how to get a BI project up and running by creating a Power BI group, a dataset, a report and a dashboard.

## Module Agenda

- Introduction to the Power BI Platform
- Creating a Power BI Test Environment
- Introduction to the Power BI Service
- Creating Datasets, Reports and Dashboards
- Loading a Sample Content Pack

## Topics Covered

What is Business Intelligence?
Brief History of Microsoft's BI Platform
Microsoft BI Technology Winners
Microsoft BI Technology Losers
What is Power BI?
Power BI Workspaces
Central Power BI Concepts
Power BI Tools
Creating a Power BI Testing Environment
Your First Stop - Office 365 admin center
Add a Secondary User for Testing
Accessing Your Office 365 Mailbox
Uploading Data Files to OneDrive for Office 365
Power BI Service
The Navigation Pane
Dashboards and Tiles
Dashboards and Reports
Reports and Pages
Report Authoring
Visuals (aka Visualizations)
Editing Visual Properties
Report and Datasets
Getting Started with the Power BI Service
Importing a Dataset
Designing Reports
Creating Dashboards
Power BI Sample Content Packs
Customer Profitability Dashboard Sample

## Instructor Demos

- Managing Users and Subscriptions in the Office 365 admin center
- Getting started with Datasets, Reports and Dashboards

© Critical Path Training. 2015. All Rights Reserved

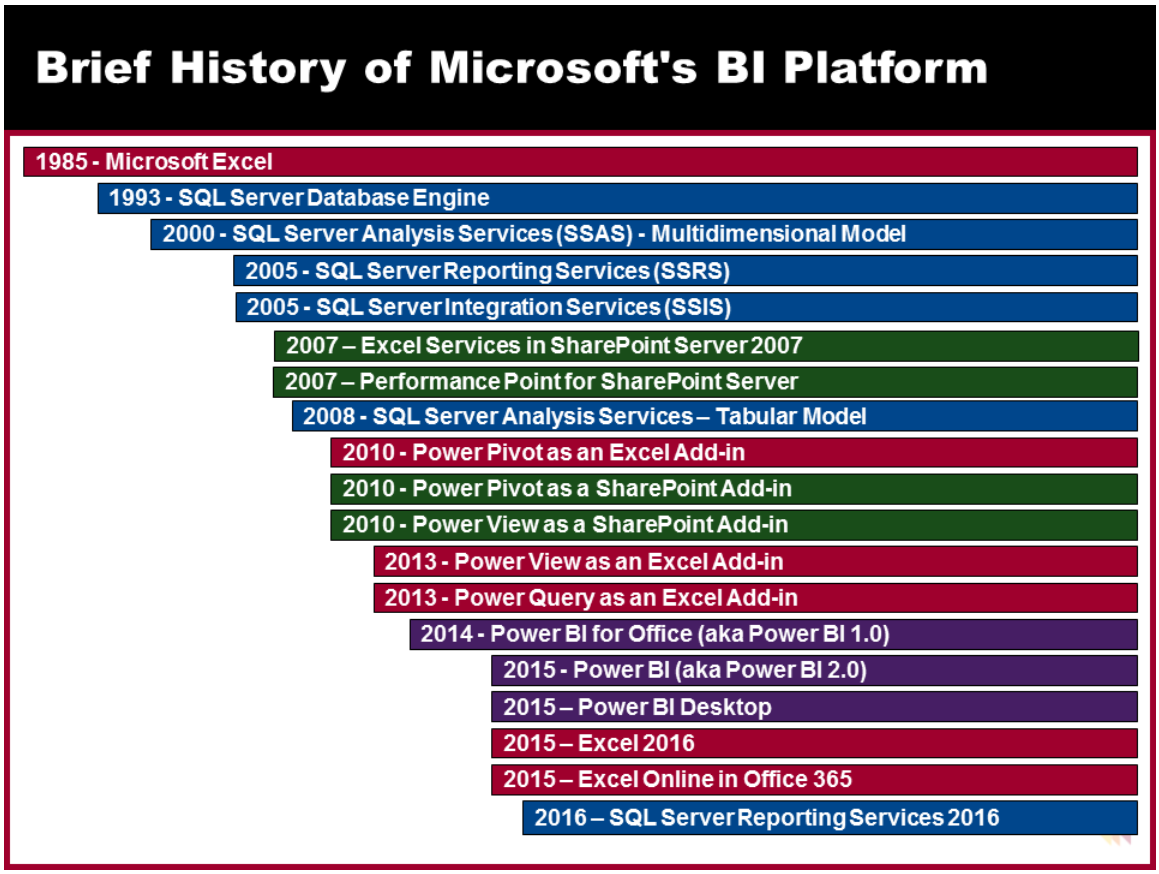
## What is Business Intelligence?

- BI focuses on business health and performance
  - Gather insights into health of business processes
  - Take corrective actions in a timely fashion
- BI projects typically involve the following…
  - Integrate data from multiple sources
  - Perform data transformation and complex calculations
  - Analyze large data sets with rapid query response
  - Create insightful reports and visualizations

"Business Intelligence" is a widely used term that can different things to different people. However, BI generally refers to the practice of monitoring business health and performance. People who perform BI roles in an organization have the job of gathering insights into health of business processes and helping the organization take corrective actions in a timely fashion.

Many BI projects share a common set of chores. There include importing and integrate data from multiple sources and performing data transformation and complex calculations on that data. Once the data has been imported and transformed, there is then the need to analyze large data sets with rapid query response and to create insightful reports and visualizations for business executives and managers.

© Critical Path Training. 2015. All Rights Reserved




© Critical Path Training. 2015. All Rights Reserved

## Microsoft BI Technology Winners

- Which BI products are strategic moving forward?
  - Microsoft Excel
  - SQL Server, SSAS, SSIS, SSRS
  - Tabular Database Model
  - Power Pivot and DAX language
  - Power Query
  - Power BI Service (aka Power BI 2.0 or what they call the "New Experience")
  - Power BI Desktop
  - Excel Online (no longer considered part of SharePoint)
  - SQL Server Reporting Services 2016

© Critical Path Training. 2015. All Rights Reserved

## Microsoft BI Technology Losers

- Technologies in which Microsoft is no longer investing
  - Power BI for Office 365 (aka Power BI v1.0)
  - Power View
  - Performance Point Services
  - Excel Services in SharePoint Server
  - SharePoint as place to surface reports & dashboards

© Critical Path Training. 2015. All Rights Reserved

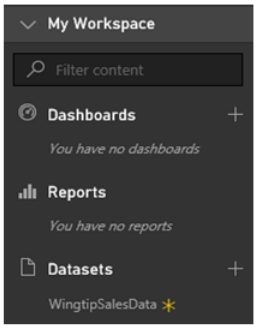
## What is Power BI?

- What is Power BI?
  - Collection of online services and features
  - Environment created to promote self-service BI *to the end user*
  - Environment created to support Power Query and Power Pivot
  - Something to assists with data import, analysis and visualization
- The Power BI service
  - accessible to subscribed users through https://app.powerbi.com
  - user has access to personal workspace and group workspaces
  - workspaces contain datasets, reports and dashboards
  - supports Power Query features for data import and transformation
  - supports Power Pivot and DAX used for data modeling
  - provides authoring environment for reports and dashboards

© Critical Path Training. 2015. All Rights Reserved

# Power BI Workspaces

- Power BI manages user context using workspaces
  - User is always running within the scope of a specific workspace
  - Personal workspace (i.e. My Workspace) created for each user
  - Group workspaces can be created for team-based authoring
- Workspace is container with three collections
  - Dashboards
  - Reports
  - Datasets




© Critical Path Training. 2015. All Rights Reserved

## Central Power BI Concepts

- Workspace
  - Provides user context (personal workspaces & group workspaces)
- Dashboard
  - Dashboard provides consolidated view into reports and datasets
  - Provides entry point in custom solution for Power BI consumers
- Report
  - Report is a collection of pages containing visuals
  - Report visual can be pinned to a dashboard
- Dataset
  - A data model containing one or more tables
  - Can be very simple or very complex

© Critical Path Training. 2015. All Rights Reserved

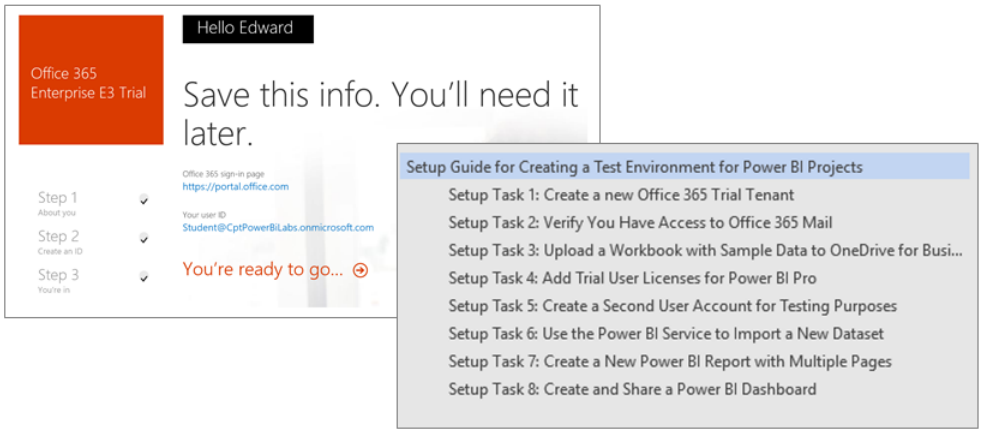
# Power BI Tools

- Power BI service using a standard web browser
  - Import datasets
  - Design reports and dashboards
  - Create content packs
  - No dataset authoring tools
- Power BI Desktop
  - Create datasets using Power Query
  - Model datasets using Power Pivot and DAX
  - Design reports
  - Publish PBIX files from local desktop to Power BI service
- Microsoft Excel 2016
  - Similar feature set to Power BI Desktop for Power Query and Power Pivot
  - Contains extra features, charts and visuals not supported Power BI
  - An complete BI solution can be embedded in a Excel workbook file

© Critical Path Training. 2015. All Rights Reserved

# Creating a Power BI Testing Environment

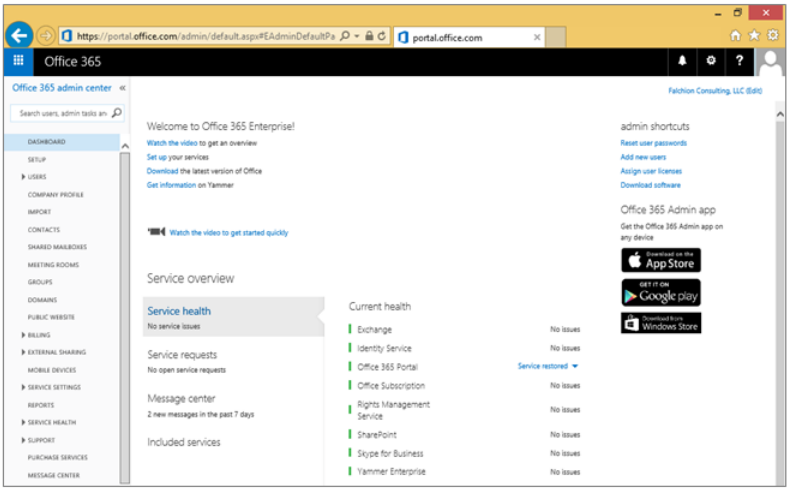
- Sign up for an Office 365 trial account
  - Creates a new Office 365 tenant
  - Sign up account you use becomes tenant administrator
  - You can add up to 25 user accounts
  - User can be configured trial licenses for Office 365 and Power BI



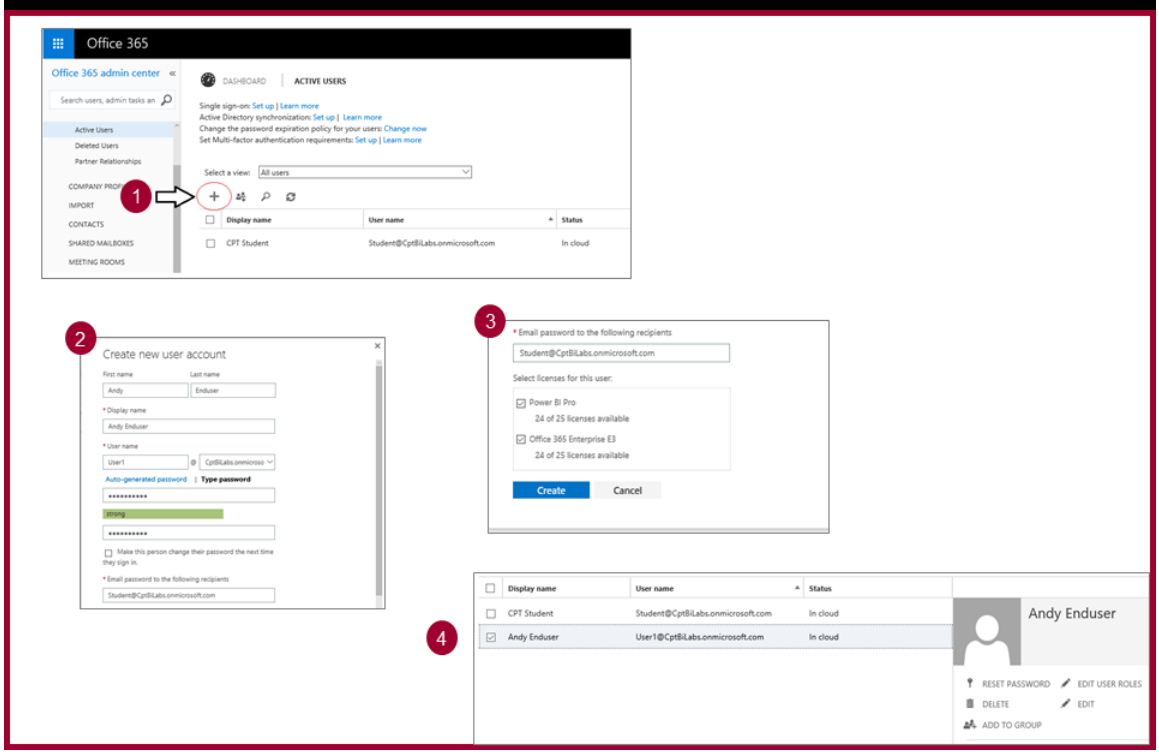

© Critical Path Training. 2015. All Rights Reserved

# Your First Stop - Office 365 admin center

- What will you need to do in Office 365 admin center?
  - Purchase 30 trial for Power BI Pro
  - Configure your Power BI License
  - Add a secondary user account for testing

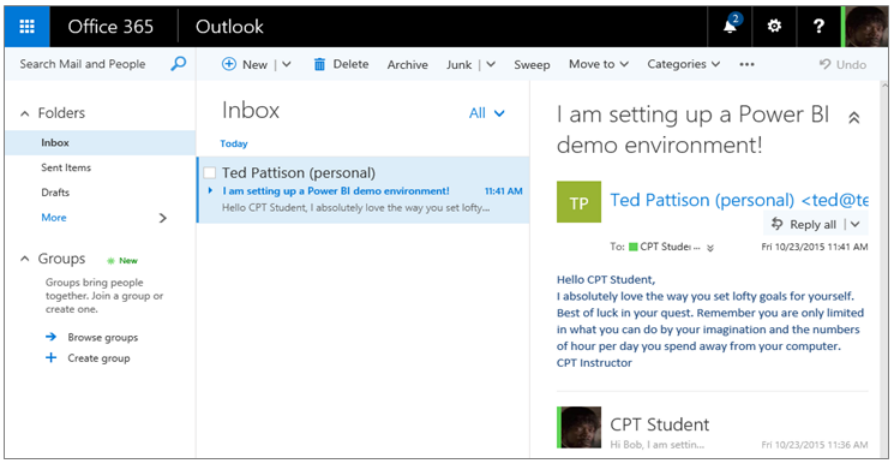


# Add a Secondary User for Testing



# Accessing Your Office 365 Mailbox

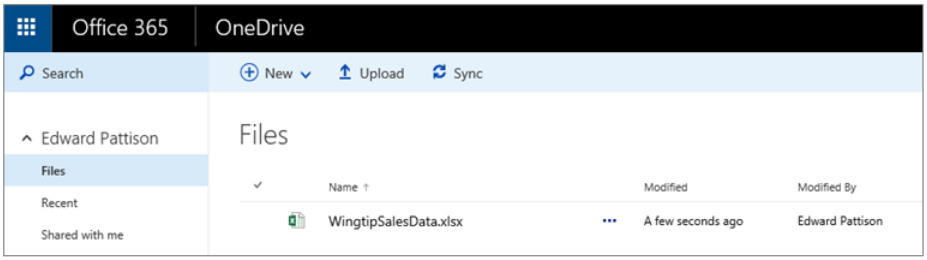
- Make sure you can access your Office 365 Inbox
  - View messages sent by Power BI service




© Critical Path Training. 2015. All Rights Reserved

# Uploading Data Files to OneDrive for Office 365

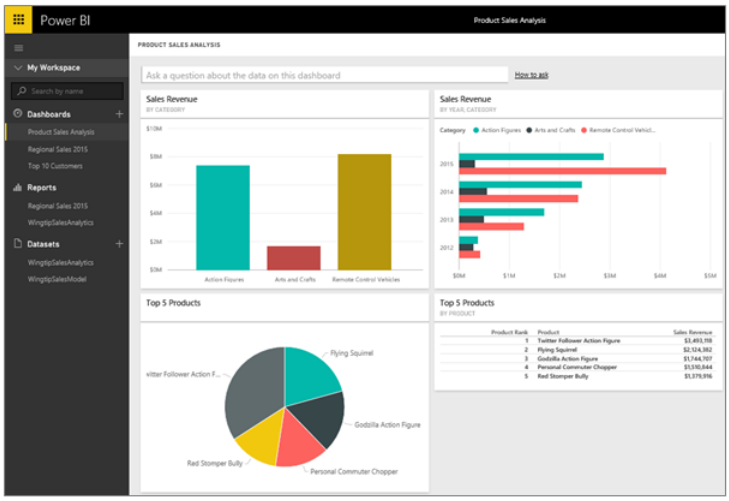
- Preferred location for data files consumed by Power BI service
  - Excel workbooks
  - CSV files
  - PBIX files created using Power BI Desktop




© Critical Path Training. 2015. All Rights Reserved

# Power BI Service

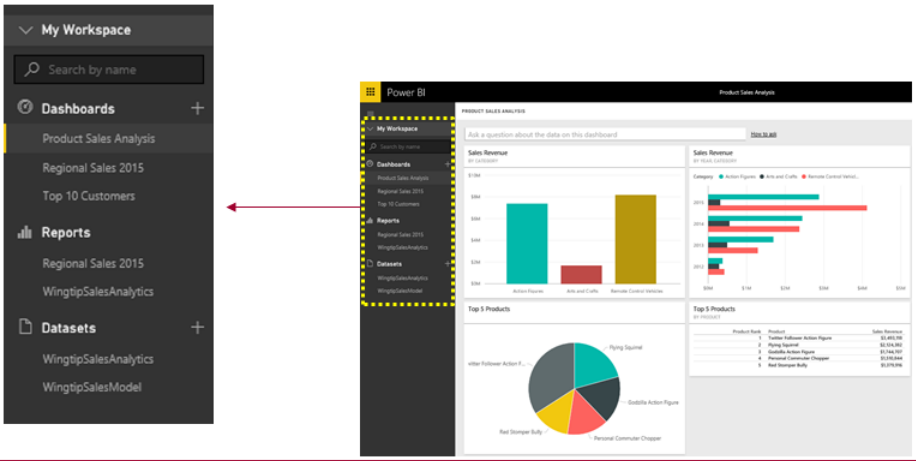
- Power BI service is accessible through browser
  - Logged in user runs in context of a specific workspace
  - Provides user with access to dashboards, reports and datasets




© Critical Path Training. 2015. All Rights Reserved

# The Navigation Pane

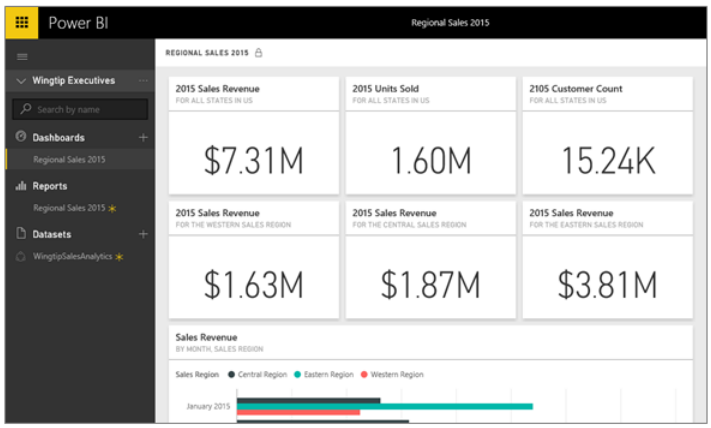
- Navigation pane is central to user experience
  - Displays links to visible dashboards, reports and datasets
  - Allows user to move between workspaces
  - Allows user to create new group workspace




© Critical Path Training. 2015. All Rights Reserved

# Dashboards and Tiles

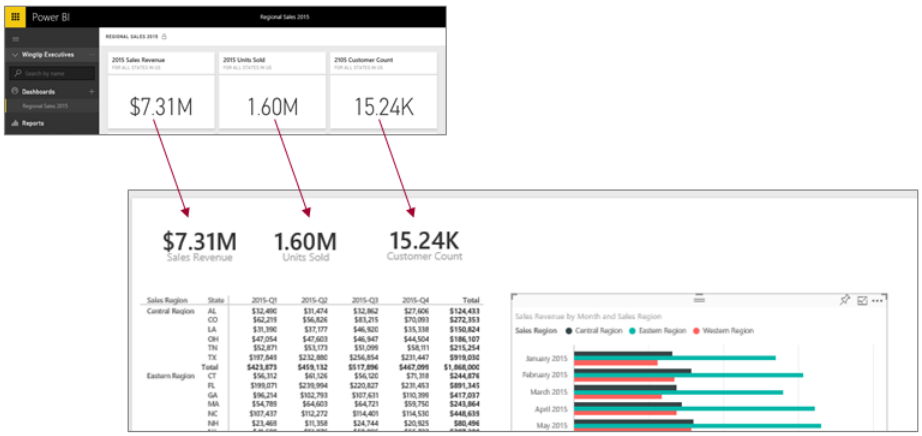
- Dashboard is a collection of tiles
  - Tile can be created by pinning visual from a report
  - Tile can be created by pinning query result from dataset




© Critical Path Training. 2015. All Rights Reserved

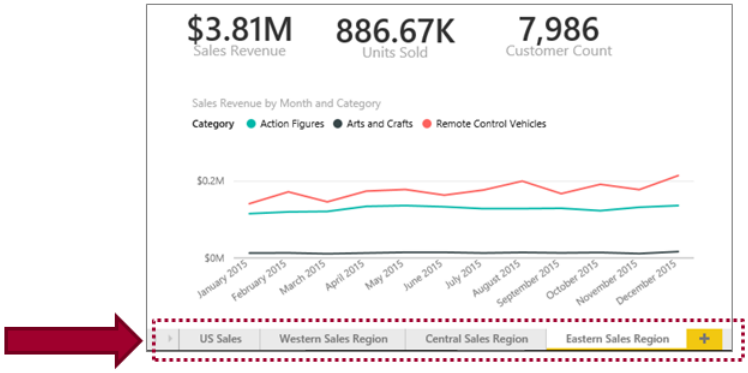
# Dashboards and Reports

- Dashboards link users to reports
  - Dashboard tiles designed to provide high-level view
  - Clicking tiles drills down into report to see more detail



# Reports and Pages

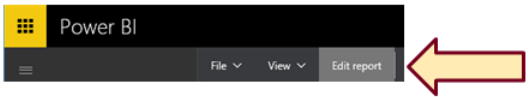
- Reports contain one or more pages
  - A report can be designed with a single page
  - A report can be designed with many pages
  - Tabbed navigation located at bottom of report view



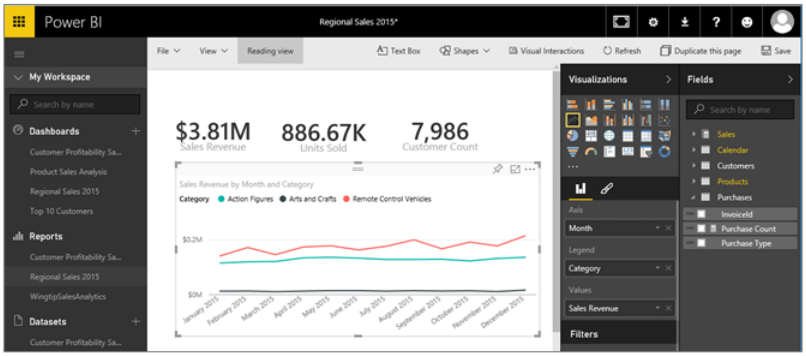

© Critical Path Training. 2015. All Rights Reserved

# Report Authoring

- Report initially opens in reading view
  - Click Edit report to switch to edit mode

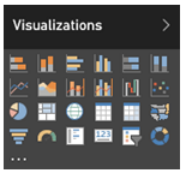



  - Report design tools appear on right side of page

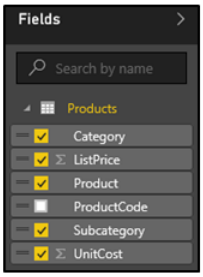


# Visuals (aka Visualizations)

- Reports are designed using visual (aka visualizations)
  - Each visuals is based on an underlying visualization type
  - Visualization type can be changed using **Visualizations** pane



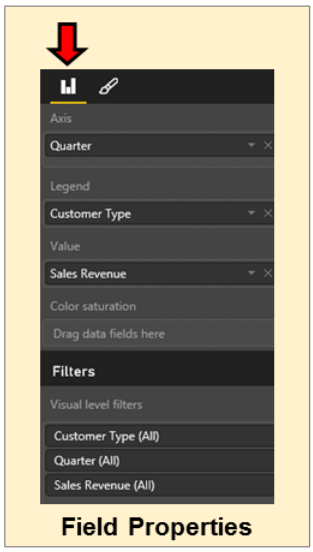

  - Visuals creating by using fields from tables inside **Fields** list



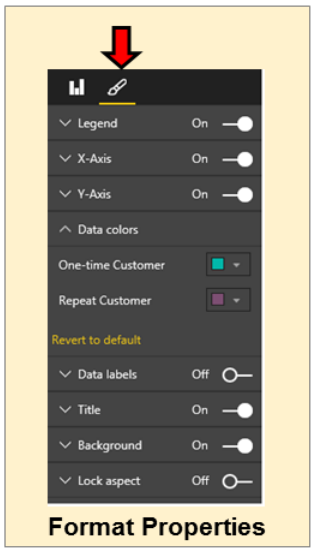

© Critical Path Training. 2015. All Rights Reserved

# Editing Visual Properties

- Visuals have Field properties and Format properties
  - Visual properties vary depending on visualization type




Field Properties

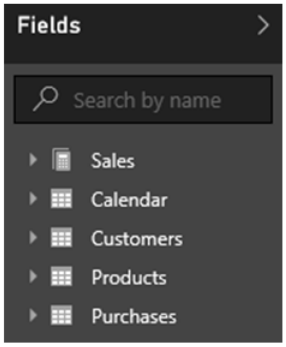


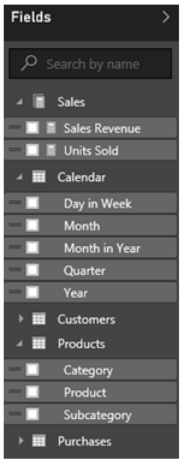

Format Properties

© Critical Path Training. 2015. All Rights Reserved

# Report and Datasets

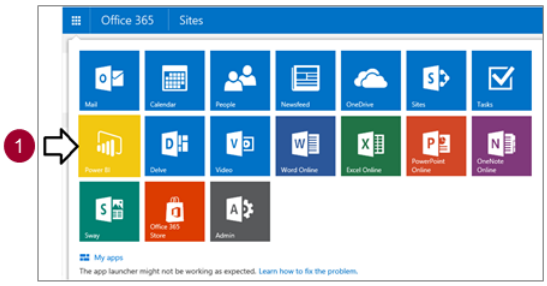
- Each report is based on an underlying dataset
  - **Fields** list in report designer shows tables and fields
  - Report author sees tables & fields as dataset consumer

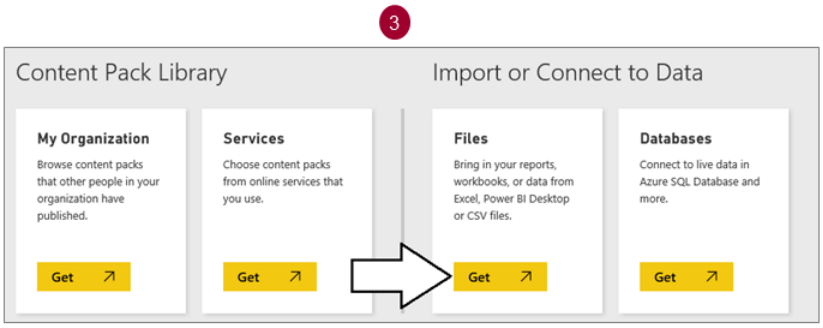



© Critical Path Training. 2015. All Rights Reserved

# Getting Started with the Power BI Service

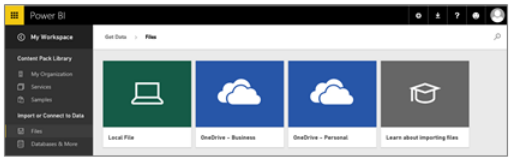




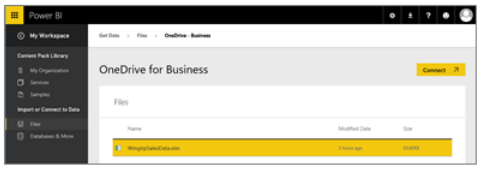

© Critical Path Training. 2015. All Rights Reserved

# Importing a Dataset

- Connect to files in OneDrive




- Select the file and click Connect

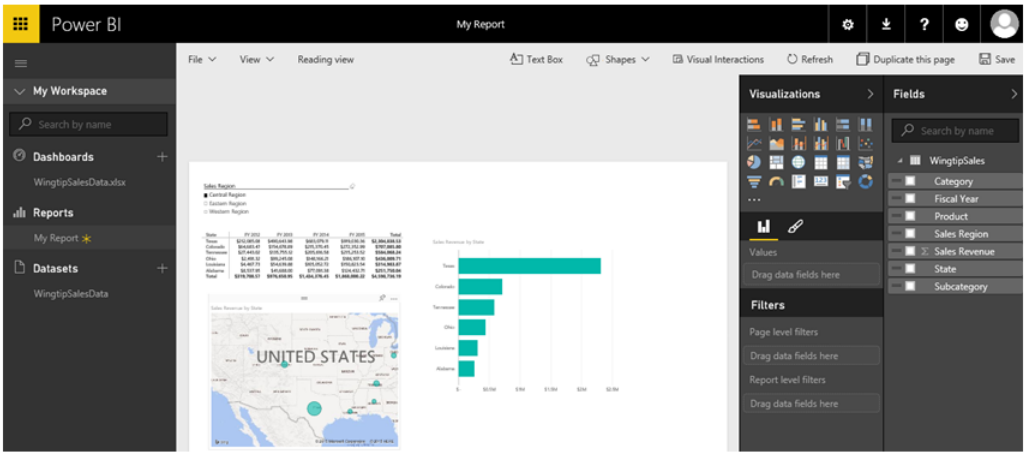



- Choose between Import or Connect

© Critical Path Training. 2015. All Rights Reserved

# Designing Reports

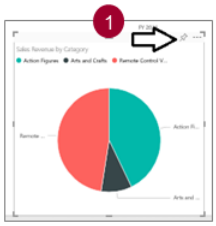
- Power BI service provides browser-based report designer
  - Report contains one o more pages
  - Pages contain visuals (aka visualizations)
  - Visuals display fields from underling data model

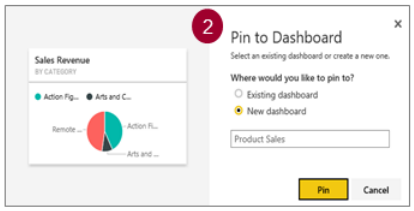


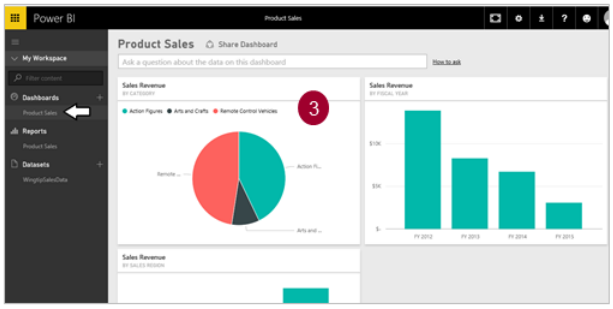

© Critical Path Training. 2015. All Rights Reserved

# Creating Dashboards

- Dashboards contain tiles
  - Tiles created from visuals on report pages

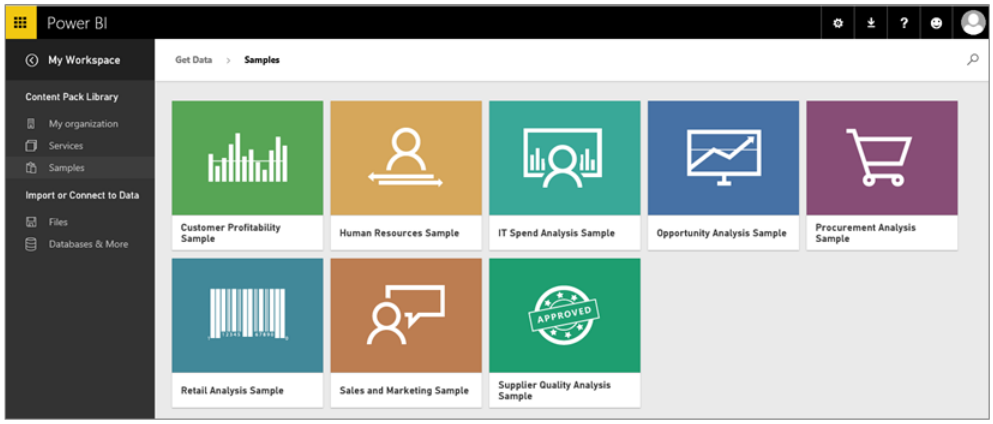


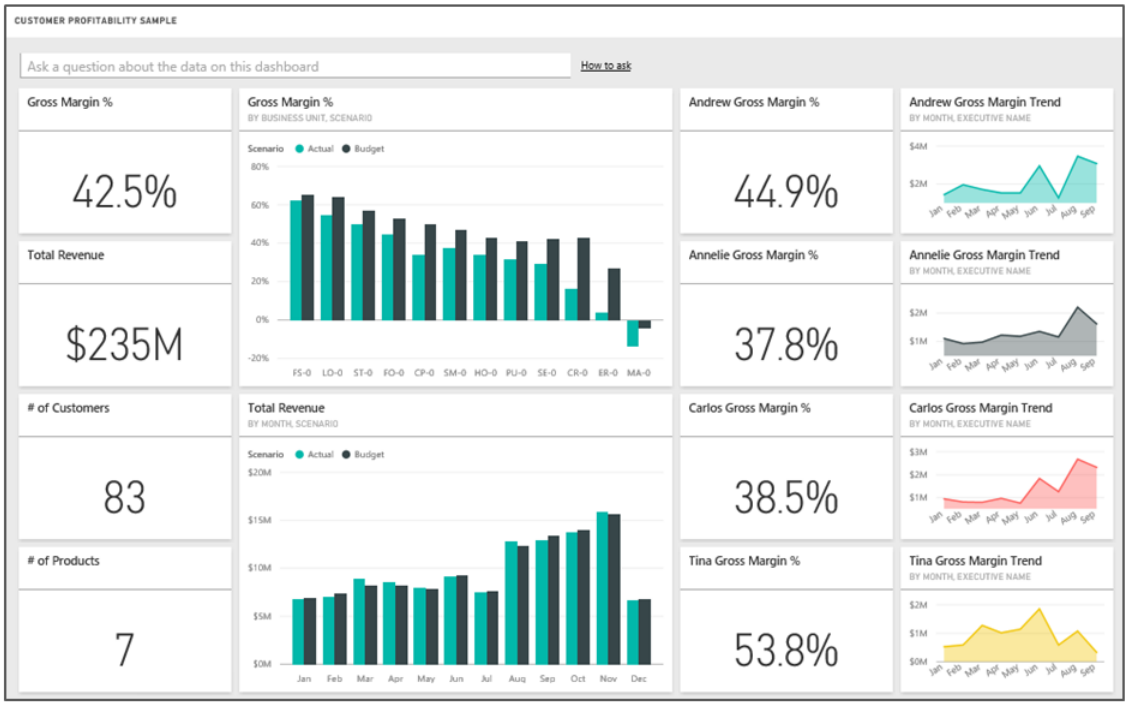
# Power BI Sample Content Packs

- Power BI service offers sample BI solutions
  - Each sample installs in seconds
  - Samples provide examples of dashboards, reports and datasets
  - Provides sample solutions for common BI solution scenarios




© Critical Path Training. 2015. All Rights Reserved

# Customer Profitability Dashboard Sample




© Critical Path Training. 2015. All Rights Reserved

# Module 01 Lab: Getting Up and Running with the Power BI Service

**Setup Time**: 45-60 minutes

**Overview**: This lab covers how to get up and running with Power BI by creating a new Office 365 tenant with trial subscriptions to Office 365 and Power BI Pro. The act of creating and configuring this new Office 365 tenant will yield an isolated testing and development environment for working on projects with the Power BI service and using Microsoft's latest self-service BI tools such as Power BI Desktop and Microsoft Excel 2016. One valuable aspect of creating a new and isolated Office 365 tenant is that you will have tenant-level administrative permissions allowing you to configure the tenant with multiple user accounts for testing your Power BI projects in isolation from any existing Office 365 tenancy.

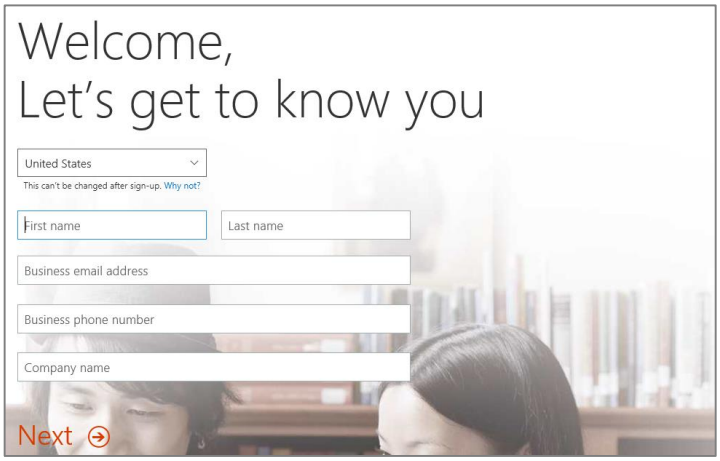
## Exercise 1: Create a new Office 365 Trial Tenant

In this exercise, you will create a new Office 365 tenant which allows you to create up to 25 user accounts with Enterprise E3 trial licenses. Being able to create multiple Office 365 user accounts in your Power BI testing environment will be important so that you can test the effects of sharing Power BI dashboards between users.

1. Navigate to the following URL:

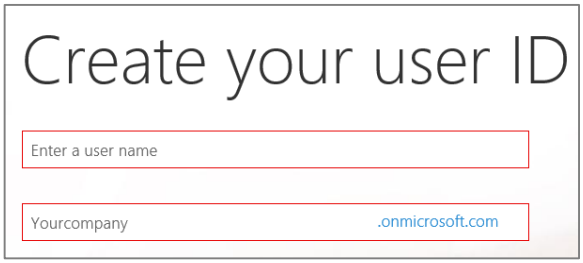
   https://go.microsoft.com/fwlink/p/?LinkID=403802&culture=en-US&country=US

2. Fill out the form with your personal information and click **Next**.




The information you provide here will be used throughout your tenant so if you do not wish to use your actual company name then provide humorous and fictitious company name.

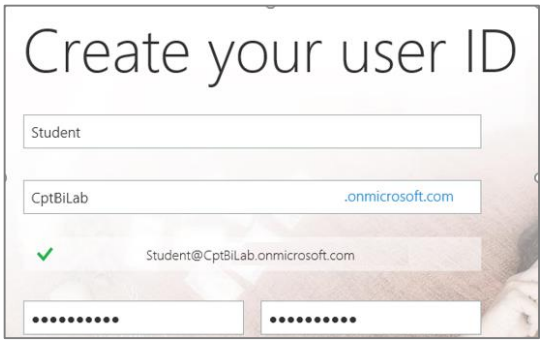
3. On the next page, you are prompted to provide a user ID, company name and password.




Note that the company name you enter on this page will be used to create the domain name for your new Office 365 trial tenant. For example, if you were to enter a company name of **CptBiLabs**, it would result in the creation of a new Office 365 tenant within a domain of **CptBiLabs.onMicrosoft.com**. The user name you enter will be used to create the first user account which will be given administrative rights within the trial tenant. If you enter a user name of **Student**, then the email address as well as user principal name for this account will be **Student@CptBiLabs.onMicrosoft.com**.

© Critical Path Training. 2015. All Rights Reserved

4. Enter a user name and a company name for your new Office 365 trial tenant. For the company name, you may wish to simply use your first and/or last name with a number which you can increment each time you have to create a new trial account (e.g. EricClapton1.onmicrosoft.com).

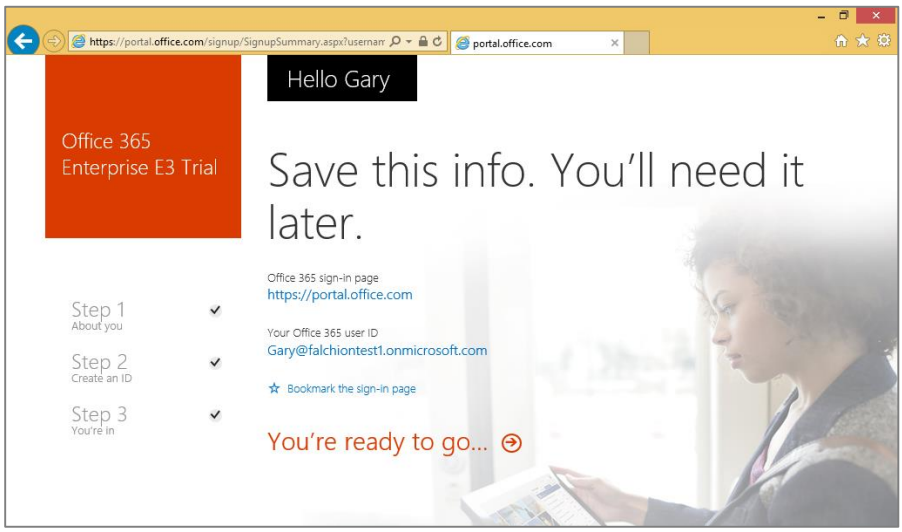



Don't use your actual company name as that may cause some conflict when your company decides to create their own official tenant. Throughout the remainder of this guide you will see a company domain name of **CptBiLab** which you should replace with the value specified for your company name.

5. Click **Next** to continue to step 3.
6. Complete the validation form in step 3 by proving you are not a robot.
   a) Select the **Text me** option and provide the number of your mobile phone.
   b) When you go through this process, a Microsoft service will send you a text message that contains an access code.
   c) You retrieve the access code form your mobile device and use it to complete the validation process.

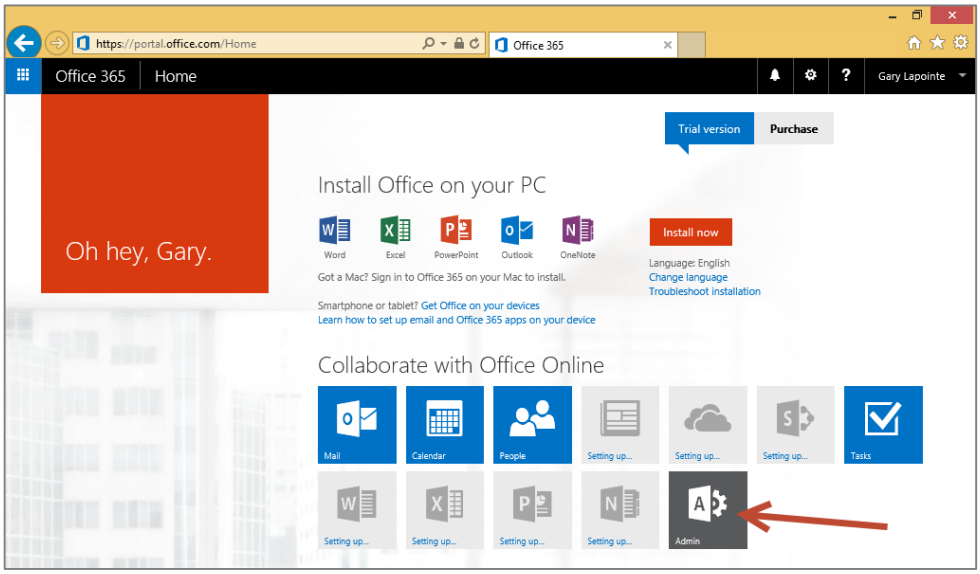
7. Once you have completed the validation process, click the **You're ready to go…** link to navigate to the portal welcome page for your new Office 365 trial tenant. Note that you should already be logged on using the user account that was created during the sign up process.

© Critical Path Training. 2015. All Rights Reserved



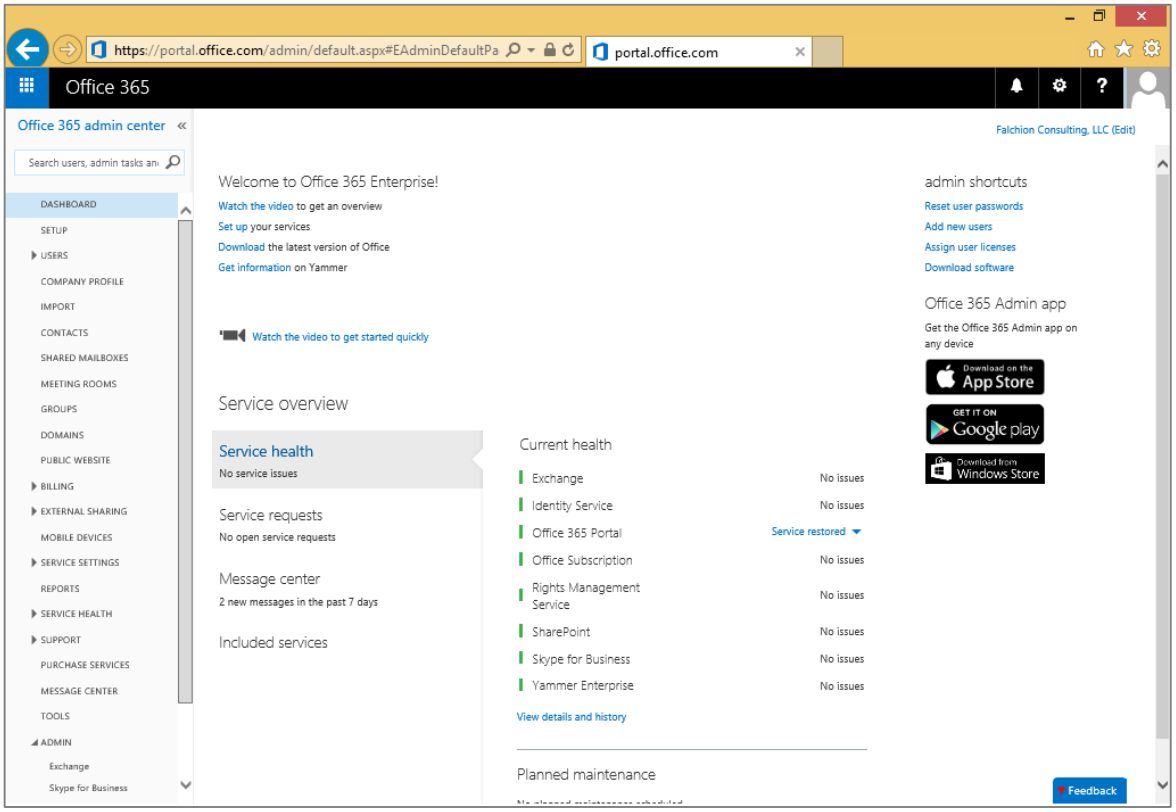

At this point, you have already created your new Office 365 tenant which can support creating up to 25 user accounts with Office 365 Enterprise E3 trial licenses. Note that some Office 365 services within your new Office 365 tenant such as the Office 365 admin center can be accessed immediately. Other services within your Office 365 tenant such as SharePoint Online are not ready immediately and will take some time to provision.

8. At this point, you should be located on the portal welcome page of Office 365. You will notice that this page shows the progress of the Office 365 environment in setting up each of the individual services that make up your new Office 365 tenant. Click the **Admin** tile to proceed to the Office **365 admin center**.



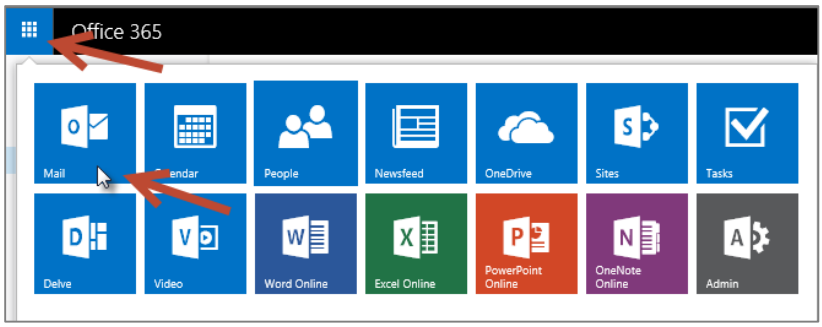

9. Verify that you are able to access the home page of the **Office 365 admin** center.

© Critical Path Training. 2015. All Rights Reserved




If you are interested in getting more familiar with the **Office 365 admin center**, take a minute to explore the administrative pages behind each of the links in the Office 365 admin center navigation menu on the left-hand side of the page.

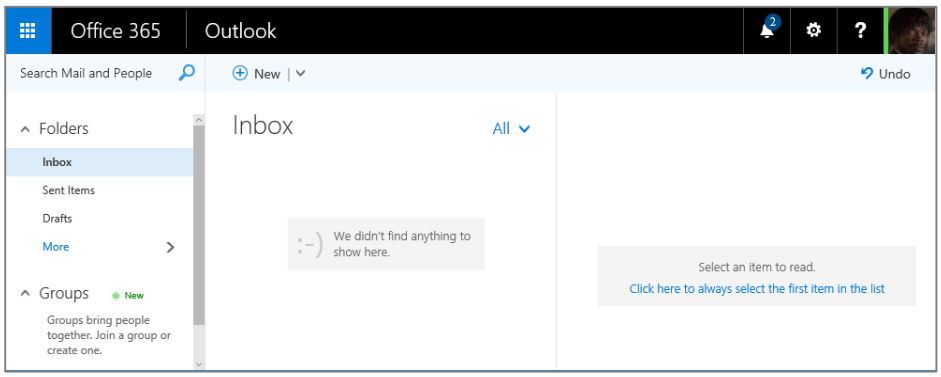
10. Make sure you can access mail and your Office 365 inbox.
    a) Open the Outlook web access client by navigating to **https://outlook.office365.com** or clicking the **Mail** icon in the Application Launcher accessible via the **Waffle Button**.



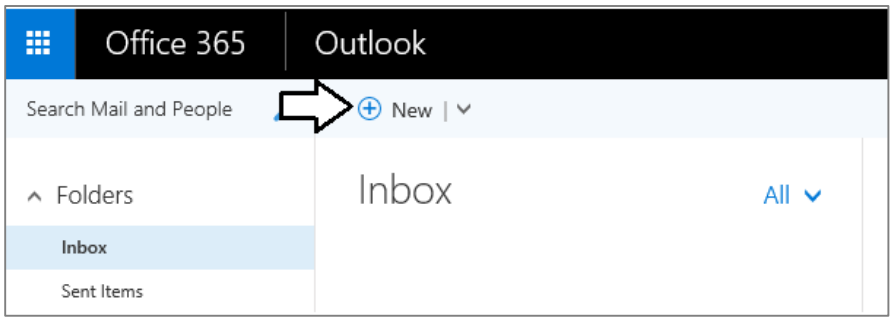

    b) If prompted, specify your language and time zone for Outlook.

© Critical Path Training. 2015. All Rights Reserved

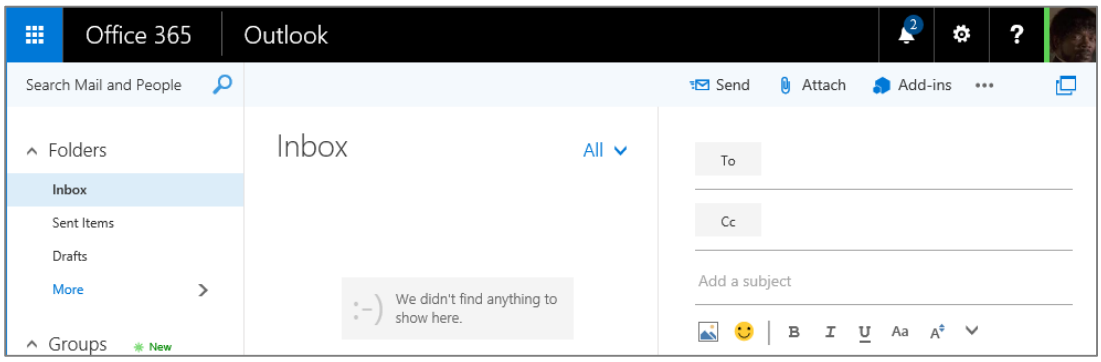
c) You should now see a web page with the Office 365 Outlook web access client and a view the Exchange inbox that is associated with the primary user account that was created when you created the Office 365 tenancy.




d) Test email by sending a message to one of your other email addresses, Display the form to create a new email clicking the **New** button. If the **New** button is not showing, it's probably because the form to create a new email is already showing.

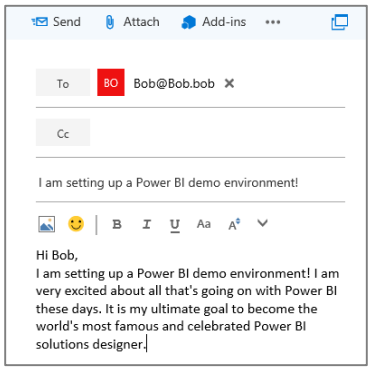



e) At this point, you should see the Outlook form to create new email on the right side of the page.




© Critical Path Training. 2015. All Rights Reserved

f) Fill out the new email form using sample data (see example below) to send a test message. Be sure to send the test message to an email address that is yours. Click **Send** when it's ready to go.

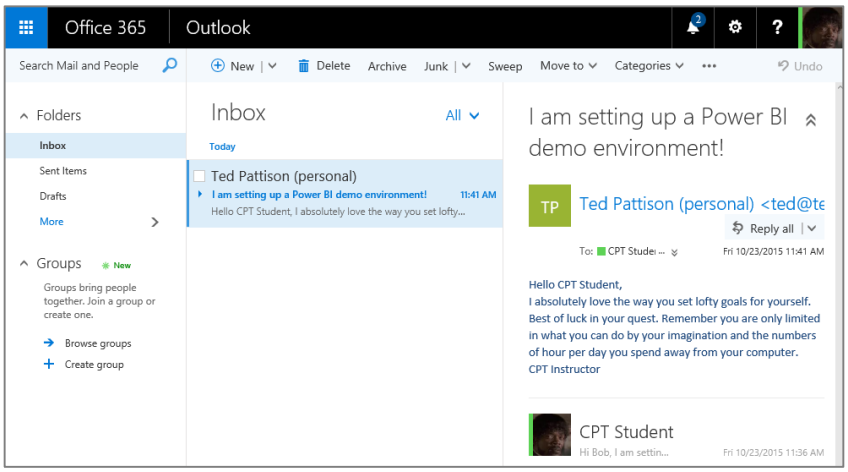



g) Click **Send** to send the email.

h) Check the email account you sent the email to and verify that you received the email.

i) Reply to the email to verify that you can send an email to your new account.

j) Return to the Outlook Web Client and verify receipt of your reply.




Having access to mail is valuable when you are working with Power BI. That's because the Office 365 and the Power BI service use email messages to send invitations and notification to users is response to user actions such as creating a new group workspace or sharing a dashboard.

## Exercise 2: Upload a Workbook with Sample Data to OneDrive for Business

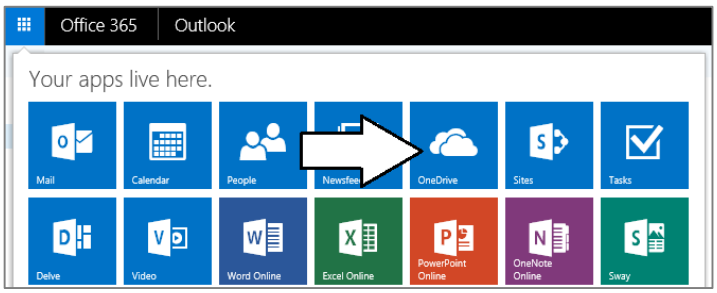
In this exercise, you will upload an Excel workbook file containing sample data to OneDrive for Business. However, the first step will to download a copy of the sample Excel workbook to your local hard drive.

1. Download the sample Excel workbook file from Critical Path Training's GitHub repository.
   a) In the browser, use the URL below to access and download the Excel workbook named **WingtipSalesData.xlsx** to the Downloads folder in your local machine.

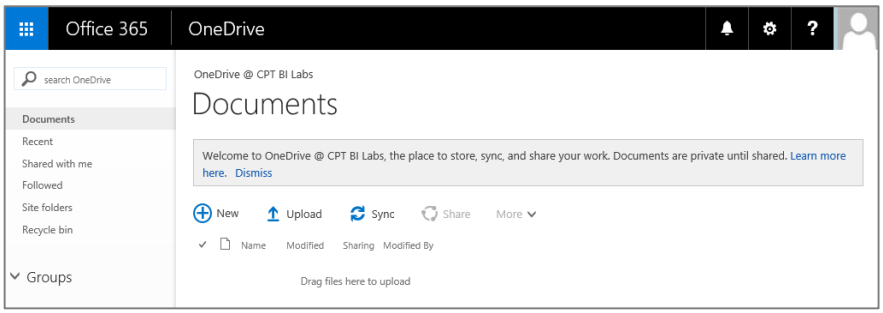
   https://github.com/CriticalPathTraining/PBI365/raw/master/Data/WingtipSalesData.xlsx

   b) Ensure that the Excel workbook file has been saved to your local hard drive.
2. Open OneDrive for Business by clicking the **OneDrive** icon in the Application Launcher accessible via the **Waffle Button**.

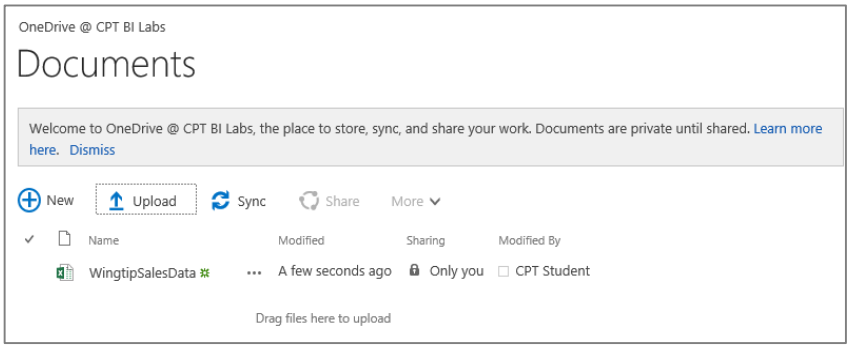
© Critical Path Training. 2015. All Rights Reserved



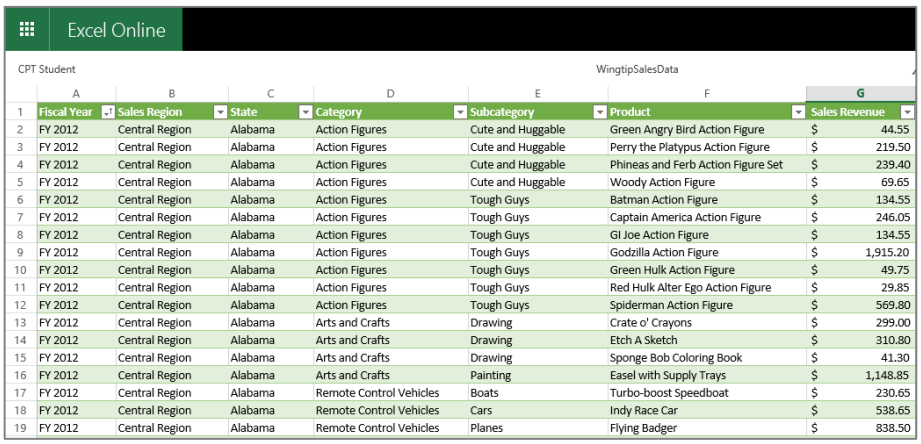

3. You should now be at the main landing page for **OneDrive for Business** which displays the **Documents** library.




4. Click the **Upload** button and go through the steps to upload the local copy of **WingtipSalesData.xlsx** to the **Documents** library. Once you have completed this step, you should be able to verify that the **Documents** library contains **WingtipSalesData.xlsx**.




5. Inside the **Documents** library, locate and click on the **WingtipSalesData** link to open the workbook in Excel Online. As you can see, the workbook contains a sample set of tabular data that will be used in a later exercise.




| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Fiscal Year | Sales Region | State | Category | Subcategory | Product | Sales Revenue |
| 2 | FY 2012 | Central Region | Alabama | Action Figures | Cute and Huggable | Green Angry Bird Action Figure | $ 44.55 |
| 3 | FY 2012 | Central Region | Alabama | Action Figures | Cute and Huggable | Perry the Platypus Action Figure | $ 219.50 |
| 4 | FY 2012 | Central Region | Alabama | Action Figures | Cute and Huggable | Phineas and Ferb Action Figure Set | $ 239.40 |
| 5 | FY 2012 | Central Region | Alabama | Action Figures | Cute and Huggable | Woody Action Figure | $ 69.65 |
| 6 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | Batman Action Figure | $ 134.55 |
| 7 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | Captain America Action Figure | $ 246.05 |
| 8 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | GI Joe Action Figure | $ 134.55 |
| 9 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | Godzilla Action Figure | $ 1,915.20 |
| 10 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | Green Hulk Action Figure | $ 49.75 |
| 11 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | Red Hulk Alter Ego Action Figure | $ 29.85 |
| 12 | FY 2012 | Central Region | Alabama | Action Figures | Tough Guys | Spiderman Action Figure | $ 569.80 |
| 13 | FY 2012 | Central Region | Alabama | Arts and Crafts | Drawing | Crate o' Crayons | $ 299.00 |
| 14 | FY 2012 | Central Region | Alabama | Arts and Crafts | Drawing | Etch A Sketch | $ 310.80 |
| 15 | FY 2012 | Central Region | Alabama | Arts and Crafts | Drawing | Sponge Bob Coloring Book | $ 41.30 |
| 16 | FY 2012 | Central Region | Alabama | Arts and Crafts | Painting | Easel with Supply Trays | $ 1,148.85 |
| 17 | FY 2012 | Central Region | Alabama | Remote Control Vehicles | Boats | Turbo-boost Speedboat | $ 230.65 |
| 18 | FY 2012 | Central Region | Alabama | Remote Control Vehicles | Cars | Indy Race Car | $ 538.65 |
| 19 | FY 2012 | Central Region | Alabama | Remote Control Vehicles | Planes | Flying Badger | $ 838.50 |

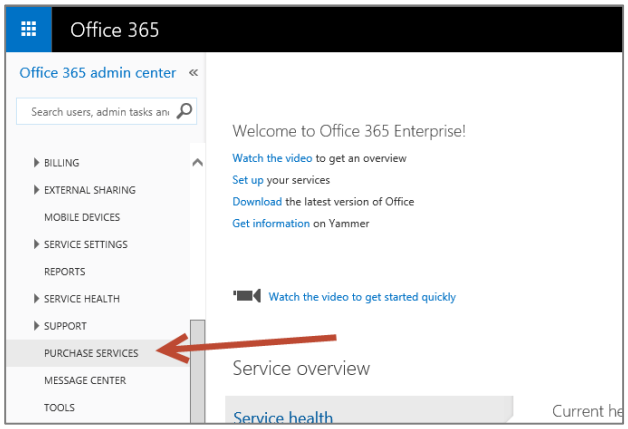
© Critical Path Training. 2015. All Rights Reserved

At this point, you have now uploaded an Excel workbook with sample data that you will use in later steps to create a dataset, a report and a dashboard.

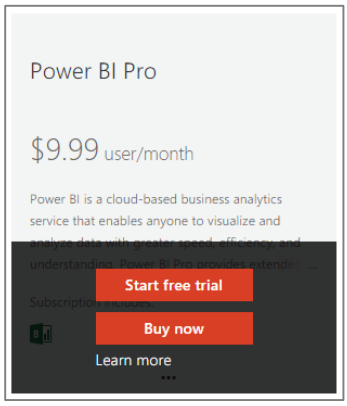
## Exercise 3: Configure Multiple Users with Power BI Pro Licenses

In this exercise, you will configure your new Office 365 tenant by adding trial subscriptions for Power BI Pro. Once you have completed this task, you will be able to begin working with the Power BI service.

1. Return to Office 365 admin center by clicking the **Admin** icon in the Application Launcher accessible via the **Waffle Button**.
2. Start a free trial of Power BI Pro.
   a) In the **Office 365 admin center**, click the **Purchase Services** link in the left column.

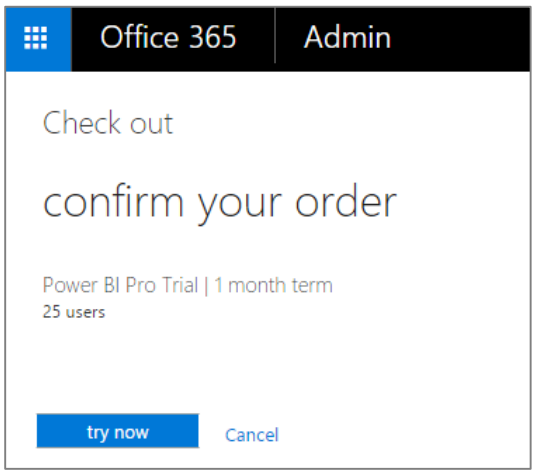



   b) Scroll down the page of available services and locate the **Power BI Pro** tile. Hover over the ellipses (…) in the bottom of the tile and click **Start free trial**.




   c) On the **Check out** page, click **try now** to add the subscription to your tenant.

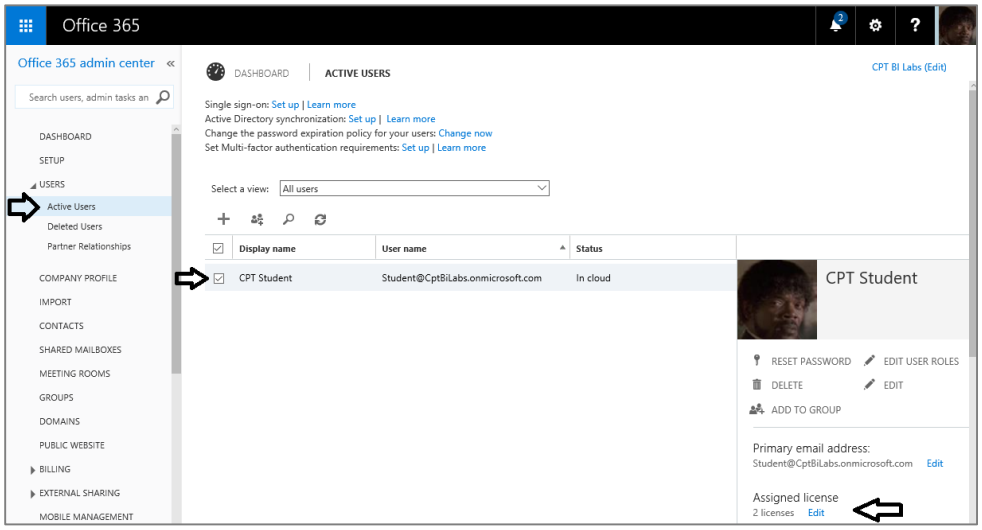
© Critical Path Training. 2015. All Rights Reserved




d) Click continue on the order receipt page.

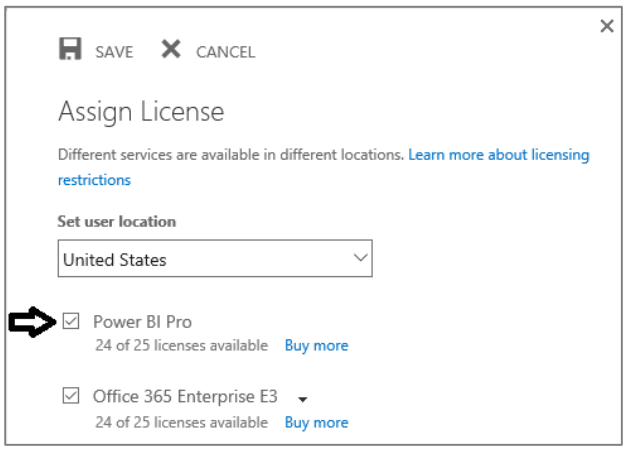
While you have just started the Power BI Pro trial, you cannot use it yet because no trial license has been assigned to a user account. The next step is to configure your user account with a trial license for Power BI Pro.

3. Assign the Power BI license to your account.
   a) In the Office 365 admin center, expand the **Users** node in the left column and click **Active Users**.
   b) Check the box next to your account and click **Edit** next to **Office 365 Enterprise E3** in the tool pane on the right.



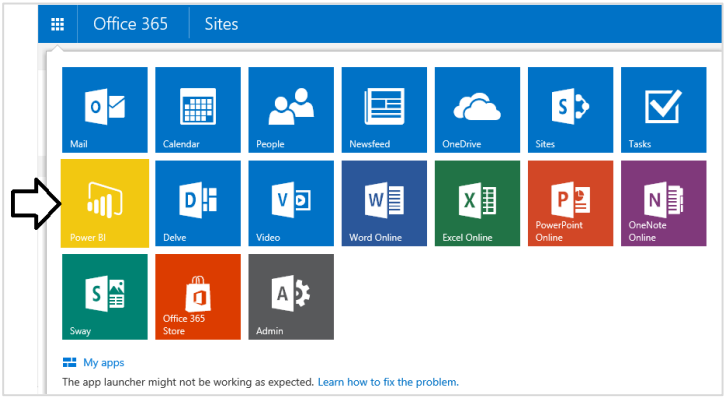

c) Expand **Microsoft Power BI for Office 365** and select the checkbox for the Power Bi pro license.

© Critical Path Training. 2015. All Rights Reserved




d) Click **Save** to complete the assignment.

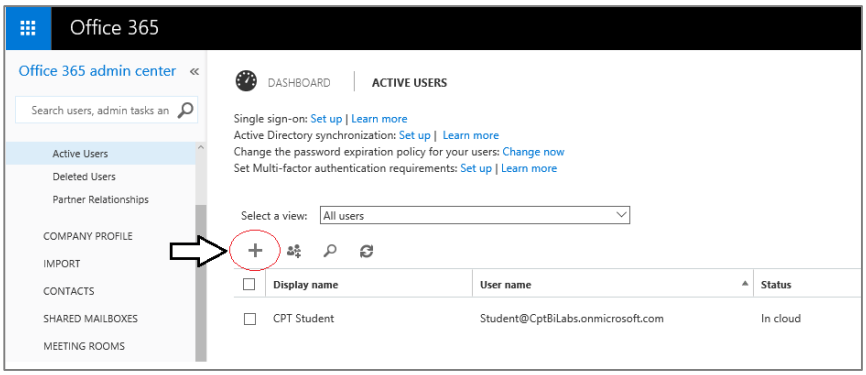
4. Check the Application Launcher accessible via the **Waffle** button and verify you can see a tile for **Power BI**.




Don't click on the tile for **Power BI** just yet. You will get to the Power BI environment in just a few more minutes. There is just one more setup task you must first complete in the Office 365 admin center which is to create a second user account. This second user account will be needed when you begin to test the effects of dashboard sharing.

5. In the Office 365 admin center, expand the **Users** node in the left column and click **Active Users**.

6. Click the button with the plus sign (**+**) to create a new user account



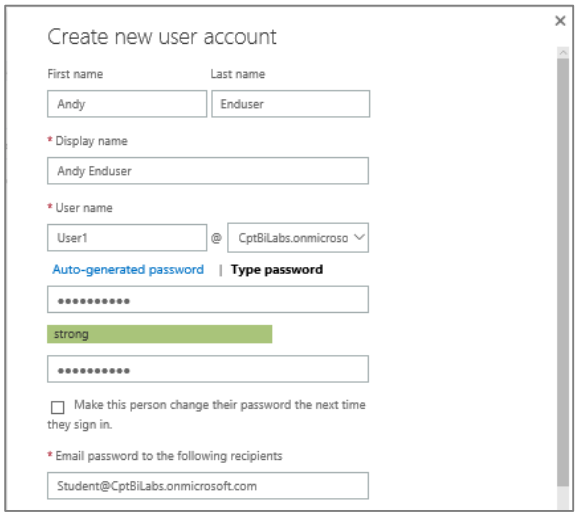

7. Fill in the **Create new user account** form with information for a new user account.

   a) When creating this account, you can use any name you would like.

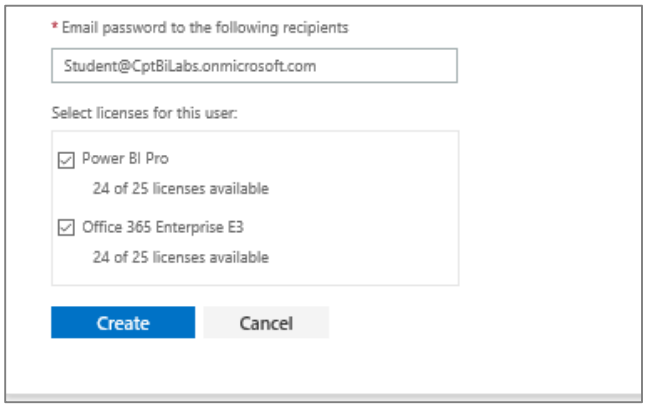
© Critical Path Training. 2015. All Rights Reserved

b) Click **Type password** and enter a password that you will remember such as **pass@word1**.

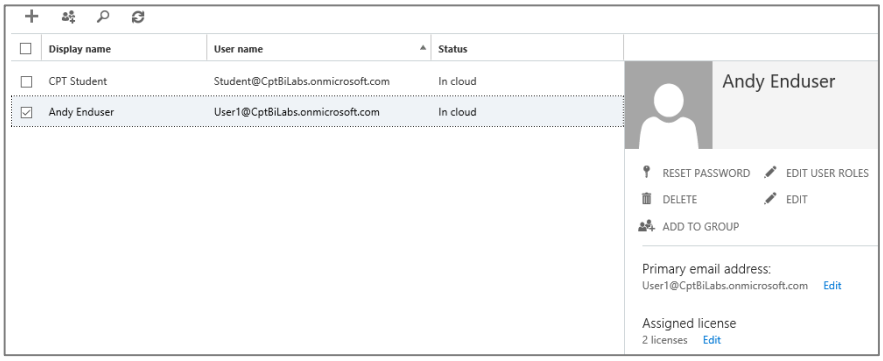
c) For your convenience, uncheck the option to Make this person change their password the next time they sign in.




d) Make sure to create the new user with trial licenses for **Office 365 Enterprise E3** and **Power BI Pro**. Click the **Create** button to create the new user account.




8. Verify that the new user account has been created.

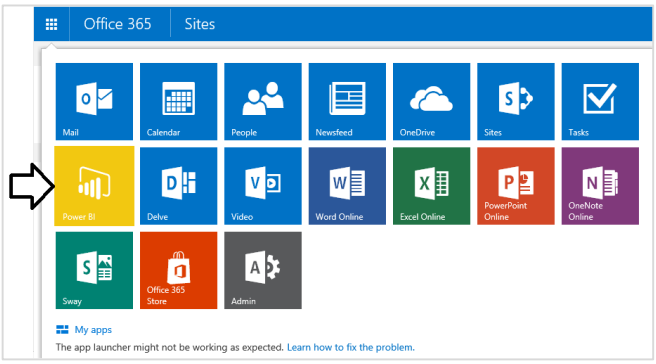



© Critical Path Training. 2015. All Rights Reserved

## Exercise 4: Use the Power BI Service to Import a New Dataset

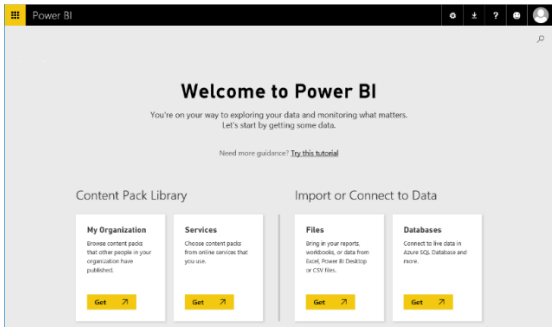
Now, after all that busy work you are finally ready to begin working with Power BI. In this setup task you will import data from an Excel workbook to create a new dataset. In the setup tasks that follow, you will create a report and a dashboard.

1. Navigate to the Power BI environment by clicking the **Power BI** icon in the Application Launcher accessible via the **Waffle Button**.

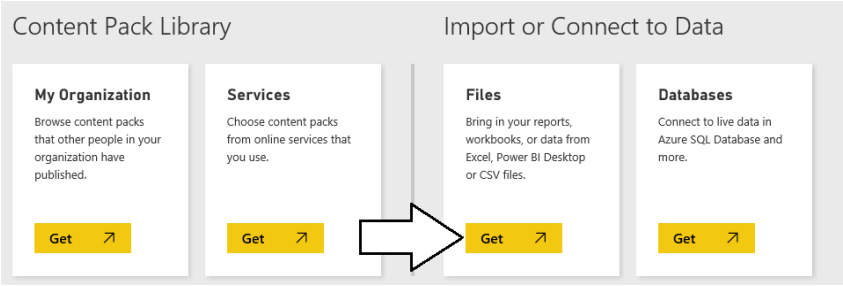



What usually happens when you click the **Power BI** tile in the Office 365 Application Launcher is that you will navigate to the page that shows the dashboards, reports and datasets in your personal workspace. However, your personal workspace is initially empty so it doesn't contain any dashboards, reports or datasets yet. Therefore, the Power BI service display a special welcome page that allows you to get started by linking to or importing data.

2. At this point, you should be at the Welcome to Power BI page as seen in the following screenshot.

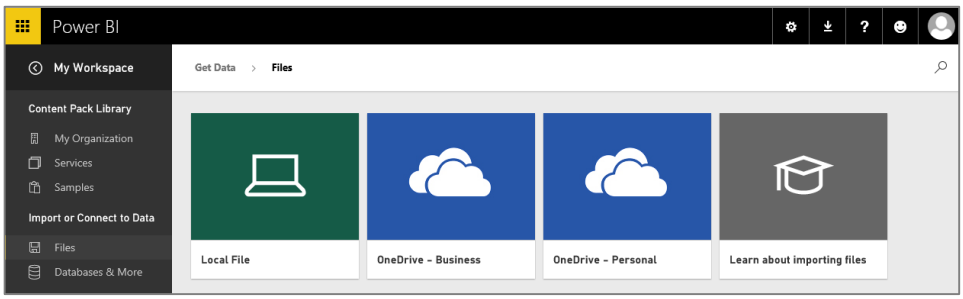



3. Click in the **Get** button in the **Files** tile under the **Import or Connect to Data** section header.

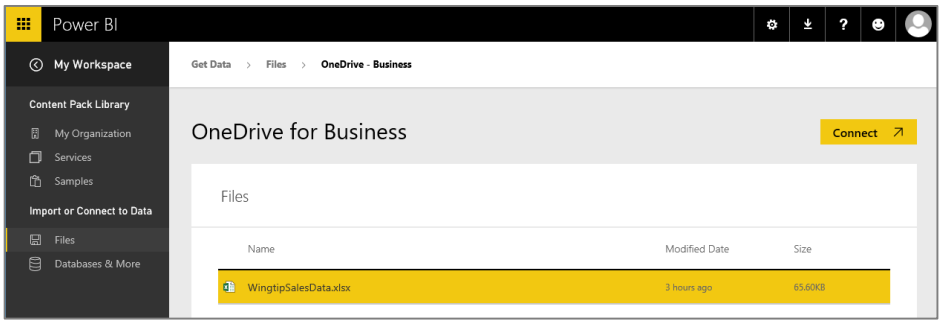



4. On the next page you should see several tiles which indicate your choices for the location of the file you would like to connect to or import. Click on the tile with the caption **OneDrive – Business** so you can import data from the Excel workbook you uploaded to your OneDrive site in a previous setup task.

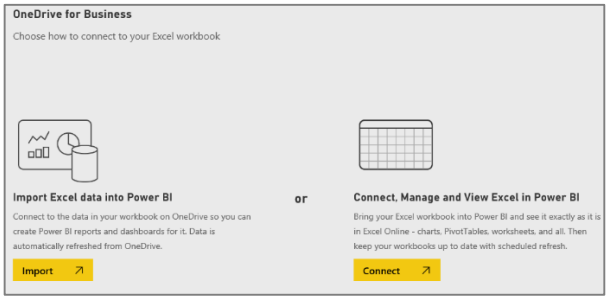
© Critical Path Training. 2015. All Rights Reserved




5. One the **OneDrive for Business** page, select the workbook named **WingtipSalesData.xslx** and then click the **Connect** button on the top right-hand side of the page.

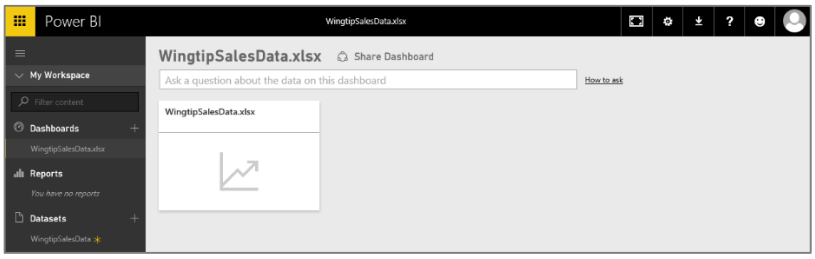



6. After clicking the **Connect** button in the previous step, you are taken to a page which prompts you to **Choose how to connect to your Excel workbook**. Click the **Import** button on the bottom left-hand side of the page to import data from the Excel workbook into the Power BI service to create a new dataset.




At this point you might make the observation that Microsoft has invested to streamline the user experience in Power BI of working with data files that have been uploaded to OneDrive sites. Once you upload your data files to a OneDrive site, they are very easy to access and integrate into your Power BI workspaces.

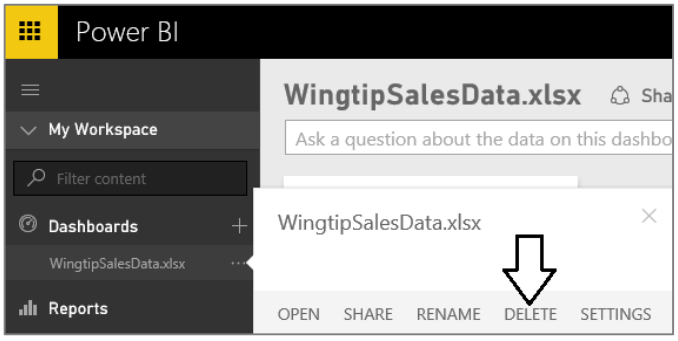
7. After the import process has completed, the Power BI service will display the main page that shows the Personal workspace named **My Workspace** for the current user. You should be able to verify that the import process has created a dataset named **WingtipSalesData** and a dashboard named **WingtipSalesData.xslx**.



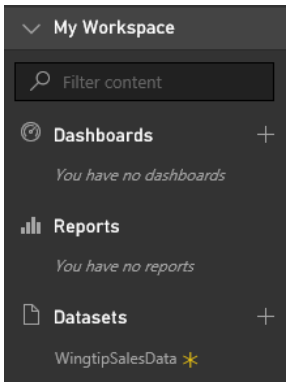

© Critical Path Training. 2015. All Rights Reserved

Note that when importing data from an Excel workbook that the Power BI service creates both a new dataset and a new dashboard. However, you might want just the dataset but not the dashboard. Feel free to delete the dashboard if you do not need it.

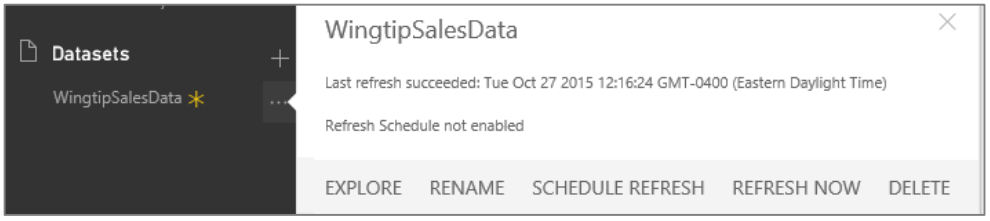
8. Delete the dashboard named **WingtipSalesData.xslx**. Accomplish this by expanding the ellipse menu to the right of the **WingtipSalesData.xlsx** dashboard and selecting the **DELETE** command.




9. Now your Power BI workspace should contain the **WingtipSalesData** dataset but there should not be any dashboards or reports.




10. Expand the ellipse menu (**…**) to the right of the **WingtipSalesData** dataset link. There is no need to execute any of the commands at this time. However, you should observe the set of commands that you can execute on a dataset that's been created by importing data from an excel workbook. For example, you have the ability to rename the dataset or to refresh its underlying data in a scenario in which the data in the underlying Excel workbook has changed.

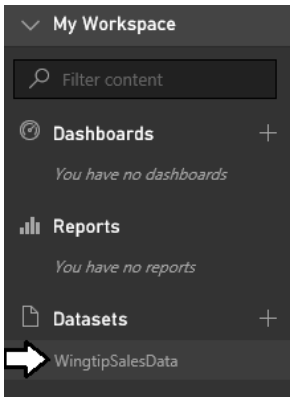



## Exercise 5: Create a New Power BI Report with Multiple Pages

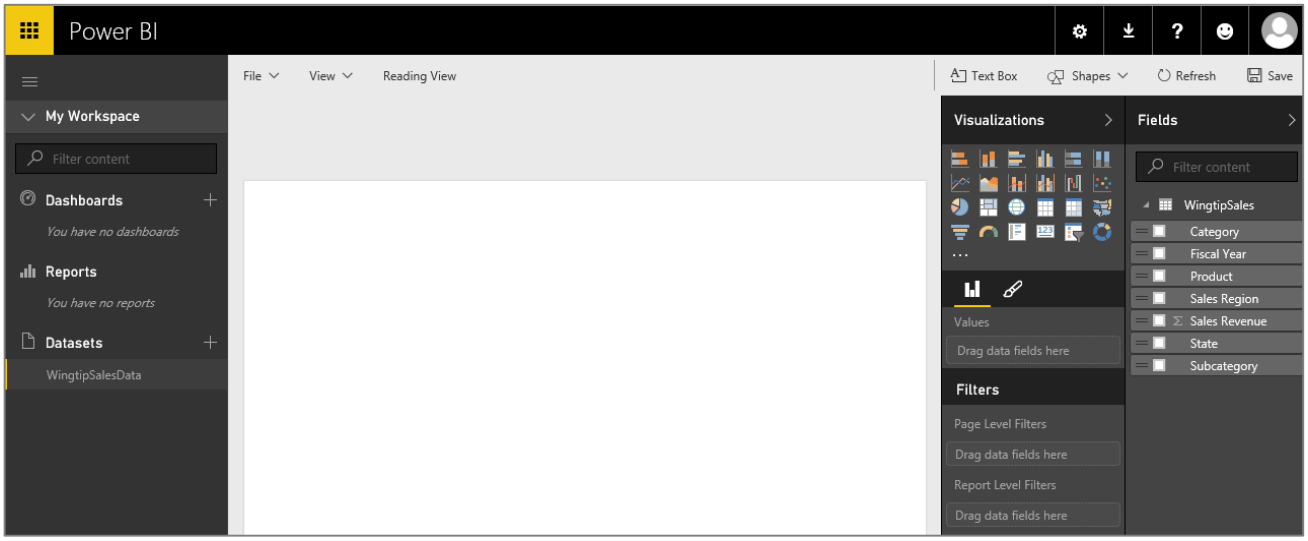
Now that you have created a dataset, the next setup step involves creating a new report with two pages of visualizations.

1. Click the **WingtipSalesData** link in the **Datasets** section of the navigation menu.

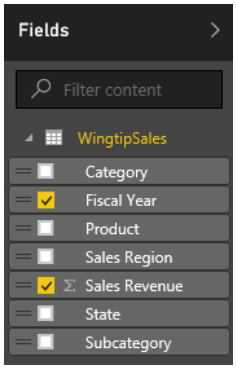
© Critical Path Training. 2015. All Rights Reserved




2. When you navigate to a dataset such as **WingtipSalesData**, the Power BI service display a page in report design mode as shown in the following screenshot. Locate the **Fields** list for the dataset on the right-hand side of the page.

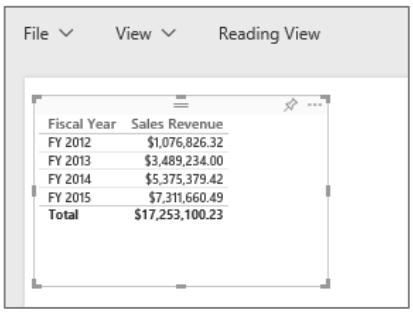



3. In the **Fields** list on the right-hand side of the page, click the checkbox beside **Fiscal Year** and then select the checkbox beside **Sales Revenue**.

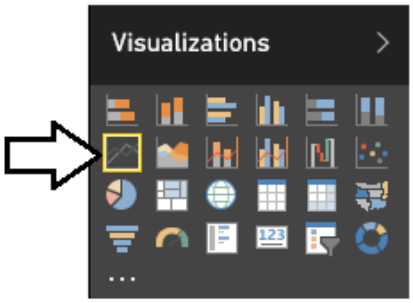



4. This should create a table visualization in the new report as shown in the following screenshot.

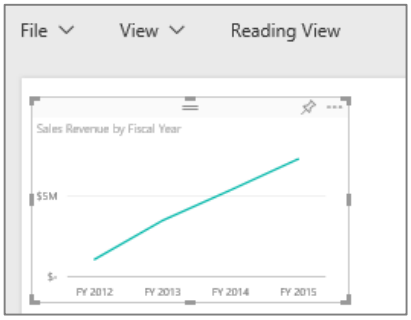
© Critical Path Training. 2015. All Rights Reserved



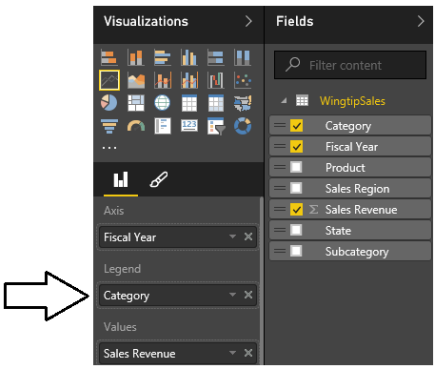

5. Change the visualization from a table to a line chart by clicking the line chart button in the **Visualizations** list.




6. At this point, you should see that the visualization on the report should now display a line chart.



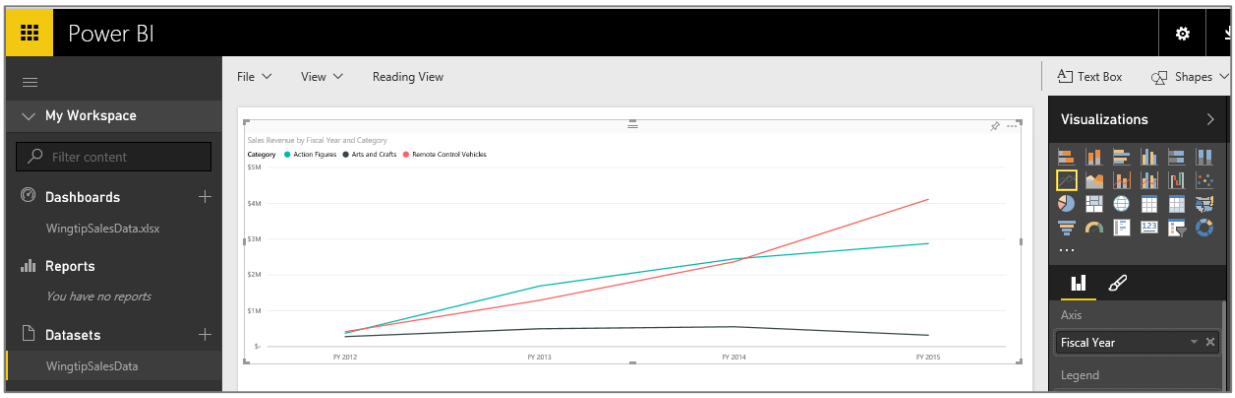

7. Next, you will add a new dimension to your visualization to show how sales revenue is distributed across product categories. First, make sure the visualization with the line chart is selected and then drag-and-drop the **Category** field from the **Fields** list into the **Legend** well in the **Visualizations** pane as shown in the following screenshot.



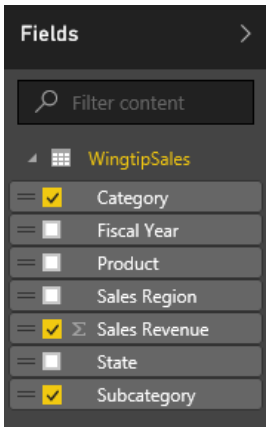

8. At this point, your visualization should match the one shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

9. Select the handle at the bottom-right corner of the visualization and resize it so it takes up the width of the current report page.

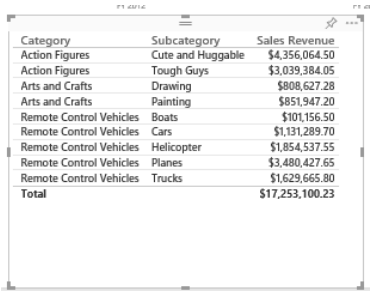



10. Now you will add a second visualization to the current report page. Begin by clicking the white space under the visualization so that the visualization is no longer selected. Next, return to the **Fields** list and select the checkbox beside the **Category** field. Next, select the checkbox beside the **Subcategory** field and then select the checkbox beside the **Sales Revenue** field.



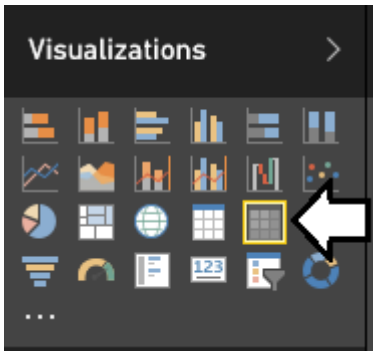

11. Now, the current report page should display a second visualization like the one shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

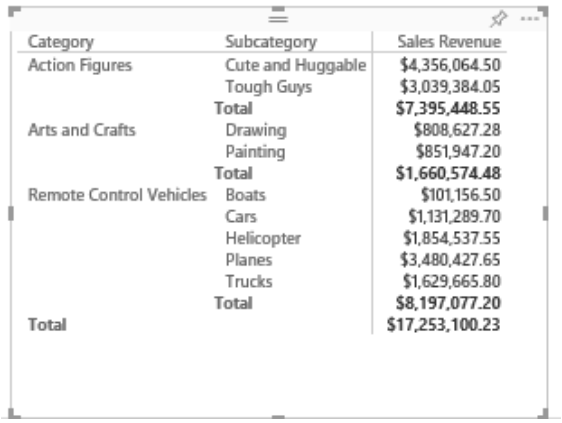


| Category | Subcategory | Sales Revenue |
|---|---|---|
| Action Figures | Cute and Huggable | $4,356,064.50 |
| Action Figures | Tough Guys | $3,039,384.05 |
| Arts and Crafts | Drawing | $808,627.28 |
| Arts and Crafts | Painting | $851,947.20 |
| Remote Control Vehicles | Boats | $101,156.50 |
| Remote Control Vehicles | Cars | $1,131,289.70 |
| Remote Control Vehicles | Helicopter | $1,854,537.55 |
| Remote Control Vehicles | Planes | $3,480,427.65 |
| Remote Control Vehicles | Trucks | $1,629,665.80 |
| **Total** | | **$17,253,100.23** |

12. Change the type of visualization from table to matrix by clicking the **Matrix** button in the **Visualizations** list.




13. Now, the second visualization should display as a matrix instead of a standard table.



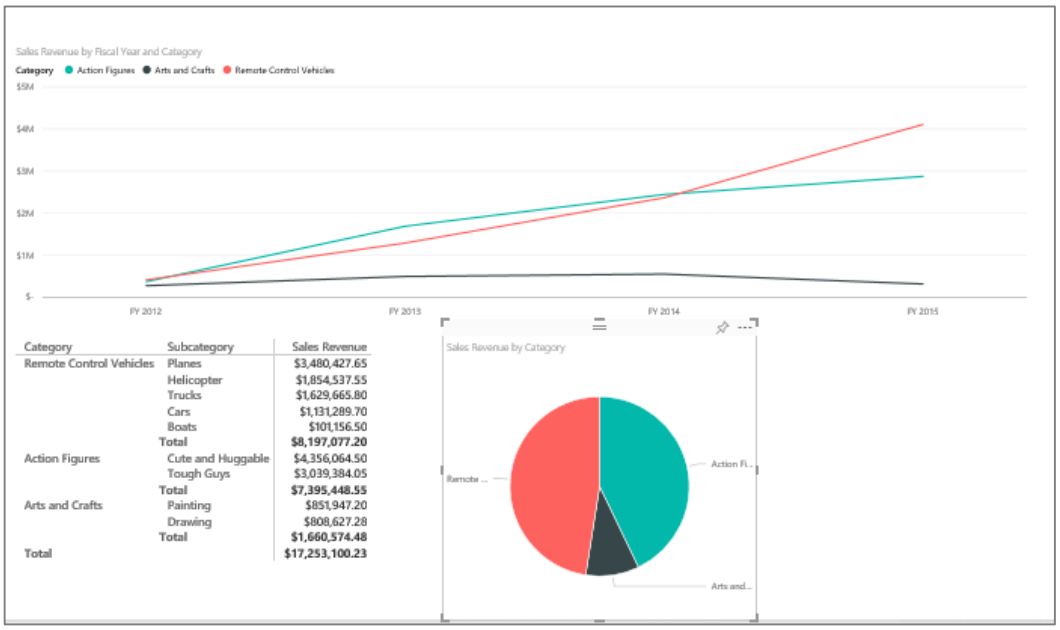
| Category | Subcategory | Sales Revenue |
|---|---|---|
| Action Figures | Cute and Huggable | $4,356,064.50 |
| | Tough Guys | $3,039,384.05 |
| | **Total** | **$7,395,448.55** |
| Arts and Crafts | Drawing | $808,627.28 |
| | Painting | $851,947.20 |
| | **Total** | **$1,660,574.48** |
| Remote Control Vehicles | Boats | $101,156.50 |
| | Cars | $1,131,289.70 |
| | Helicopter | $1,854,537.55 |
| | Planes | $3,480,427.65 |
| | Trucks | $1,629,665.80 |
| | **Total** | **$8,197,077.20** |
| **Total** | | **$17,253,100.23** |

14. Inside the matrix, click on the **Sales Revenue** column header to resort the data in the matrix so that the product categories and subcategories with the highest amounts of sales revenue are sorted to the top of the matrix.

© Critical Path Training. 2015. All Rights Reserved

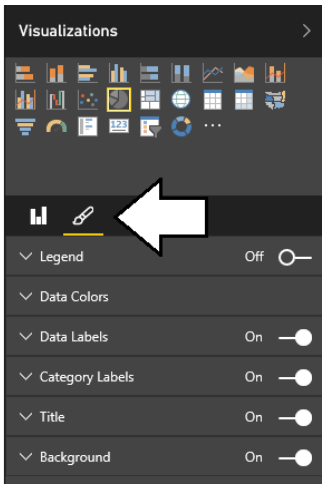
| Category | Subcategory | Sales Revenue |
|---|---|---|
| Remote Control Vehicles | Planes | $3,480,427.65 |
| | Helicopter | $1,854,537.55 |
| | Trucks | $1,629,665.80 |
| | Cars | $1,131,289.70 |
| | Boats | $101,156.50 |
| | **Total** | **$8,197,077.20** |
| Action Figures | Cute and Huggable | $4,356,064.50 |
| | Tough Guys | $3,039,384.05 |
| | **Total** | **$7,395,448.55** |
| Arts and Crafts | Painting | $851,947.20 |
| | Drawing | $808,627.28 |
| | **Total** | **$1,660,574.48** |
| **Total** | | **$17,253,100.23** |

15. Now you will add a third visualization to the current report page. Begin by clicking the white space on the report page outside of the two existing visualizations so that neither visualization is selected. Next, return to the **Fields** list and select the checkbox beside the **Category** field. Next, select the checkbox beside the **Sales Revenue** field. After creating the visualization, change it to a pie chart by clicking the Pie chart button in the **Visualizations** list.

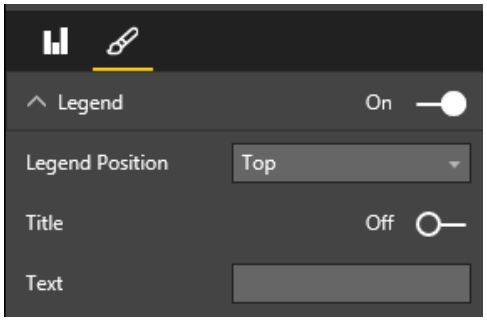



16. With the pie chart visualization selected, click on the pen icon in the **Visualizations** task pane so you can edit the visual properties of the new pie chart visualization.

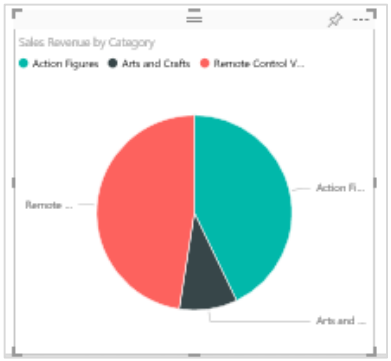
© Critical Path Training. 2015. All Rights Reserved




17. Turn on the **Legend** option and then expand the **Legend** section. Set the **Legend Position** to **Top** and turn off the **Title** option as shown in the following screenshot.



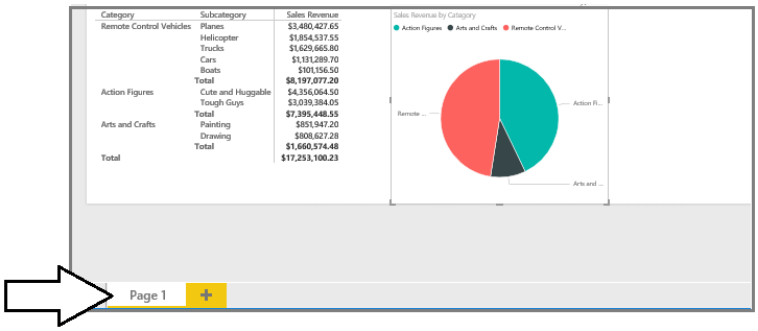

18. At this point, you should see the visual changes you have made to the pie chart.




If you have time, you might explore the other options available for editing the appearance of a visualization by examining the other options that are available on the Visualizations task pane when a visualization is selected. Note that the set of available options change depending on what type of visualization is selected.

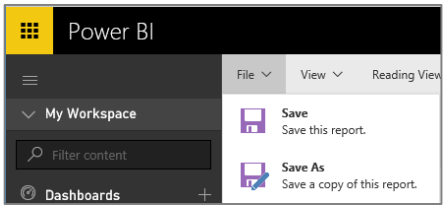
19. Now it is time to save the report. Begin by changing the name of the current page. Locate the report page name section at the bottom left of the current page and observe that the page has been given an initial name of **Page 1**.

© Critical Path Training. 2015. All Rights Reserved




20. Double click on the page name of **Page 1** to enter edit mode and then update the page name to **Sales by Product Category**,

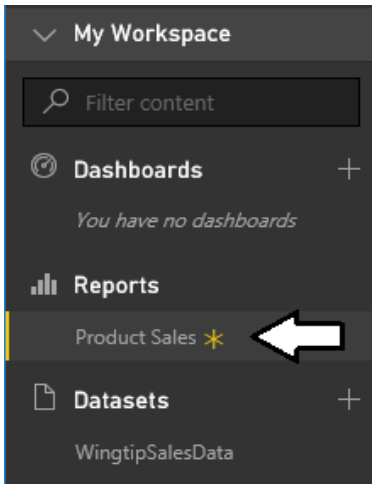
21. Save the report by dropping down the reports **File** menu and selecting the **Save As** menu command.



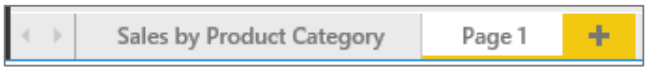

22. When prompted, enter a report name of **Product Sales** and click the **Save** button.

23. After saving the **Product Sales** report, you should be able to see a link for it in the **Reports** section of the left-hand navigation.

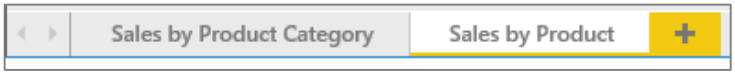
© Critical Path Training. 2015. All Rights Reserved



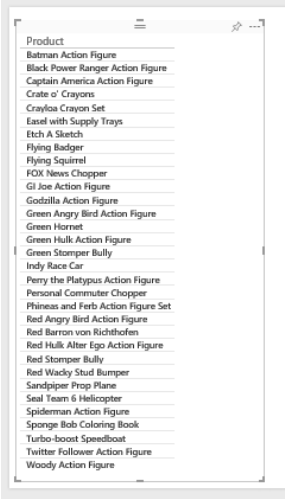

24. Now, add a second page to the **Product Sales** report. Accomplish this by clicking the button with the plus (+) sign to the right of the page name. The Power BI service will respond by creating a second page named **Page 1**.




25. Change the name of the second page from **Page 1** to **Sales by Product**.

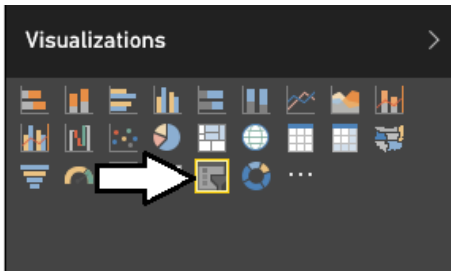



26. On the new **Sales by Product** page, add a new visualization by selecting the checkbox beside the **Product** field from the **Fields** list. This should create a simple table visualization with a list of products. Resize the height of the visualization to display all products at once without the need for a scrollbar.

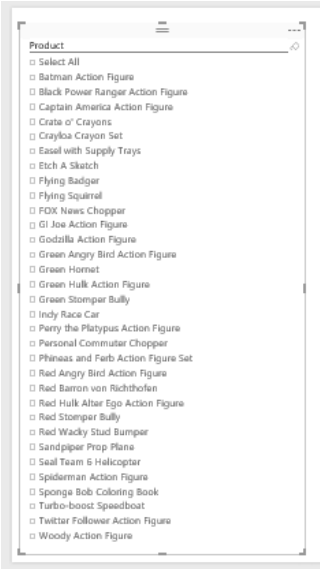



27. Change the type of visualization from a table to a slicer by clicking the Slicer button in the **Visualizations** list.

© Critical Path Training. 2015. All Rights Reserved

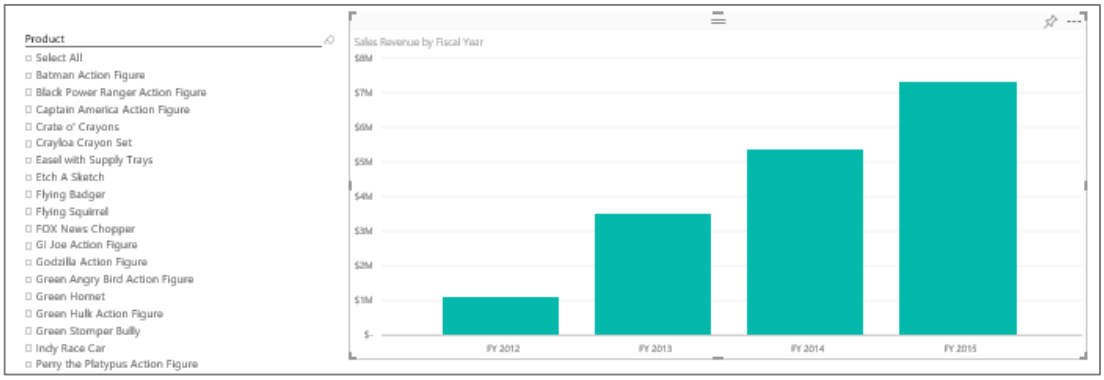



28. Now that the visualization has been changed to a slicer, you should see that each product has an associated checkbox.



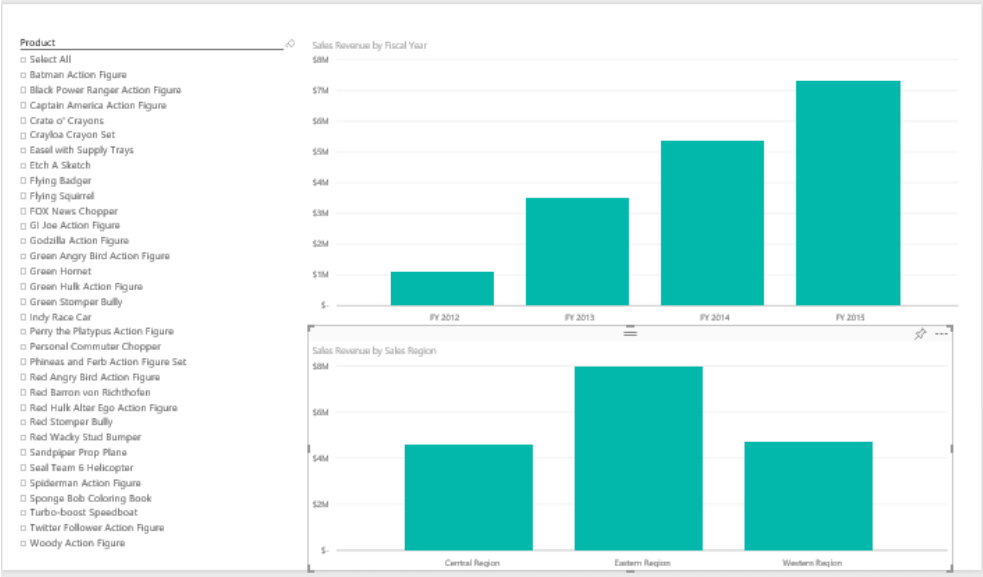

Keep in mind that this slicer visualization adds the ability for the current user to intact with this report by selecting one or more products using these checkboxes. When a user changes the selection of products, the Power BI service will automatically refresh the other visualizations on the page by filtering the results using the selected product or products. Learning how to make reports interactive is a key to creating effective BI solutions with Power BI.

29. Next, you will add a second visualization to **Sales by Product** page. Click whitespace in the report to ensure the first visualization is not selected. Next, create a new visualization by selecting the checkbox for the **Sales Revenue** field and then selecting the checkbox for the **Fiscal Year** field. This should create the second visualization on the page as a bar chart as shown in the following screenshot.

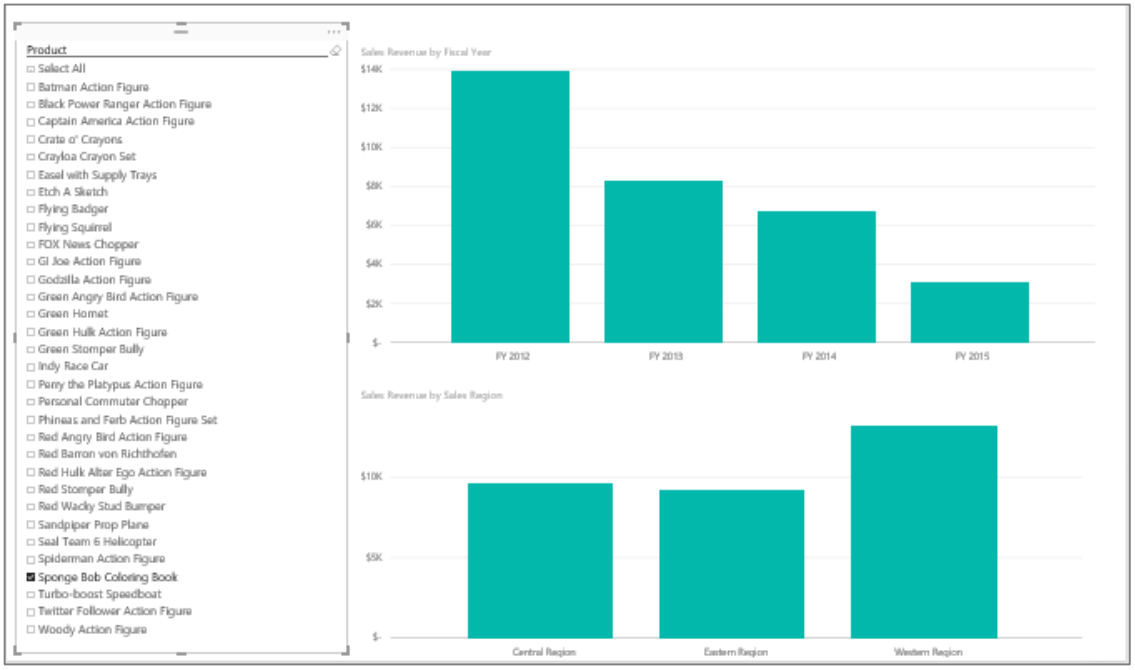



© Critical Path Training. 2015. All Rights Reserved

30. Now you will add a third visualization. Click whitespace in the report to ensure the neither of the two visualizations are currently selected. Next, create a third visualization by selecting the checkbox for the **Sales Revenue** field and then selecting the checkbox for the **Sales Region** field. This should create the third visualization as a bar chart as shown in the following screenshot.

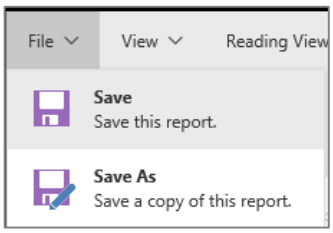



31. Test out the slicer by selecting one product at a time and observing how the two other visualizations on the page automatically refresh to show sales data for one product at a time. Play the role of a business analyst and determine which products have the most positive increases in sales revenue from year to year. Also, find the products with downward trending sales. If you examine the sales data for the **Sponge Bob Coloring Book**, you can sales revenue is trending in the wrong direction over the last four years.




32. Now you are done authoring the report. Save the work you have done authoring the report by dropping down the **File** menu and selecting to **Save** menu command.

© Critical Path Training. 2015. All Rights Reserved

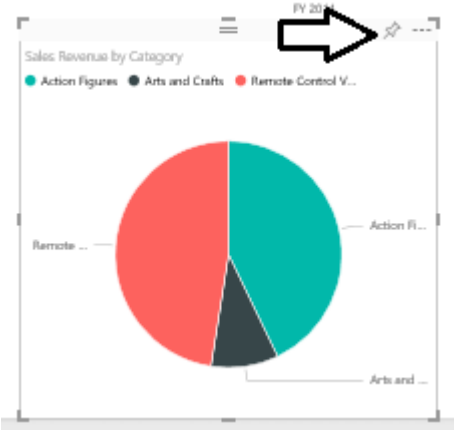



Now that you have created a report with multiple pages, it is time to move on to the last setup task where you will create a new dashboard and test sharing it with another user.

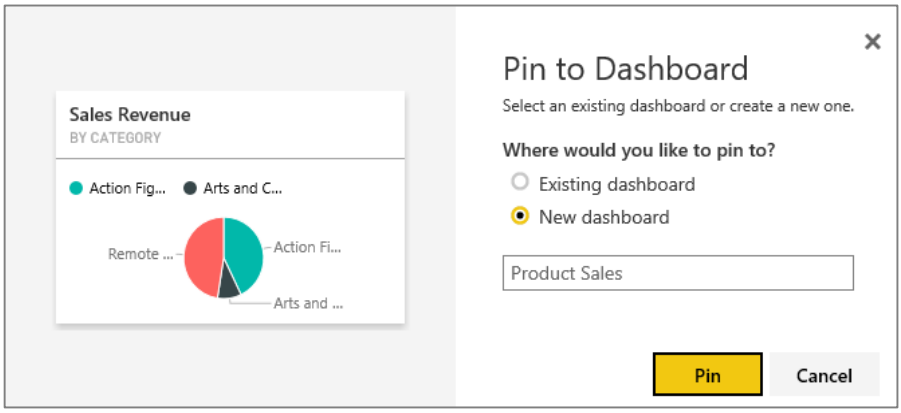
## Exercise 6: Create and Share a Power BI Dashboard

While you have already created a dataset and a report, you must create a dashboard to effectively share a customized BI solution with other users. This final setup task will walk you through the steps of creating and sharing a Power BI dashboard.

1. Navigate to the **Sales by Product Category** page of the **Product Sales** report.
2. Inspect the pie chart with product categories. Locate and click the button with the thumbtack icon which is used to pin a report visualization to a dashboard.

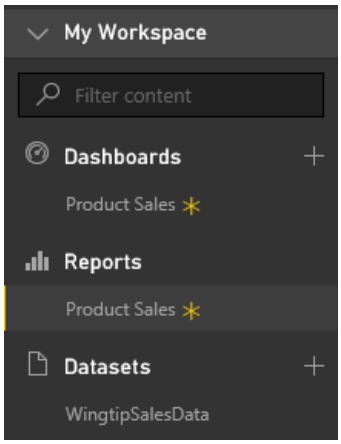



3. When you click the button with the thumbtack icon, you will be prompted with the dialog which asks you where to pin the visualization. Select the option to pin the visualization to a **New Dashboard** and give the new dashboard a name of **Product Sales**. When the **Pin to Dashboard** form is filled out like the one shown in the following screenshot, click the **Pin** button.

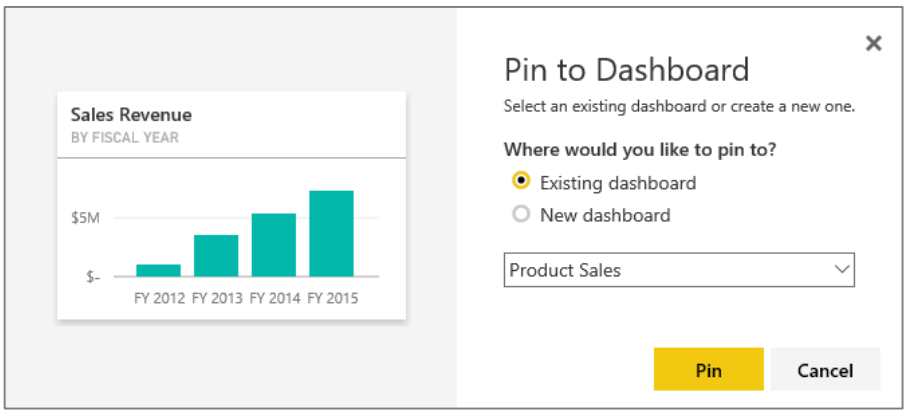



4. At this point, the new **Product Sales** dashboard should be created and a link to it should appear in the left navigation menu.

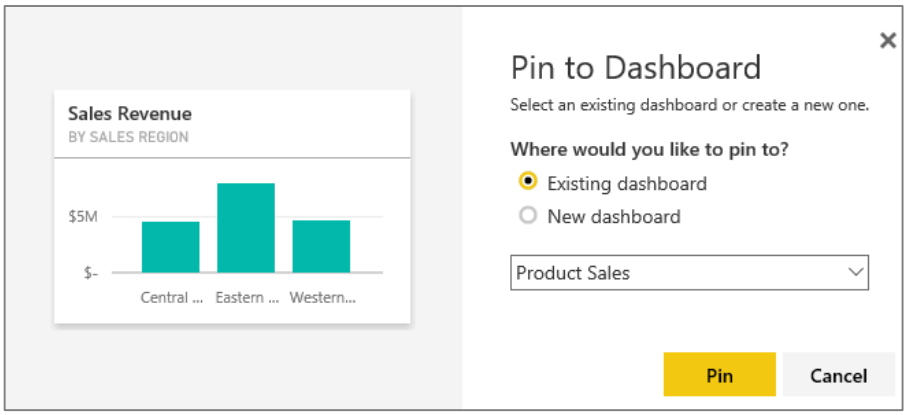
© Critical Path Training. 2015. All Rights Reserved




5. Navigate to the **Sales by Product** page of **Product Sales** report and follow the same steps to pin the bar chart visualization showing sales revenue by fiscal year to the **Product Sales** dashboard.

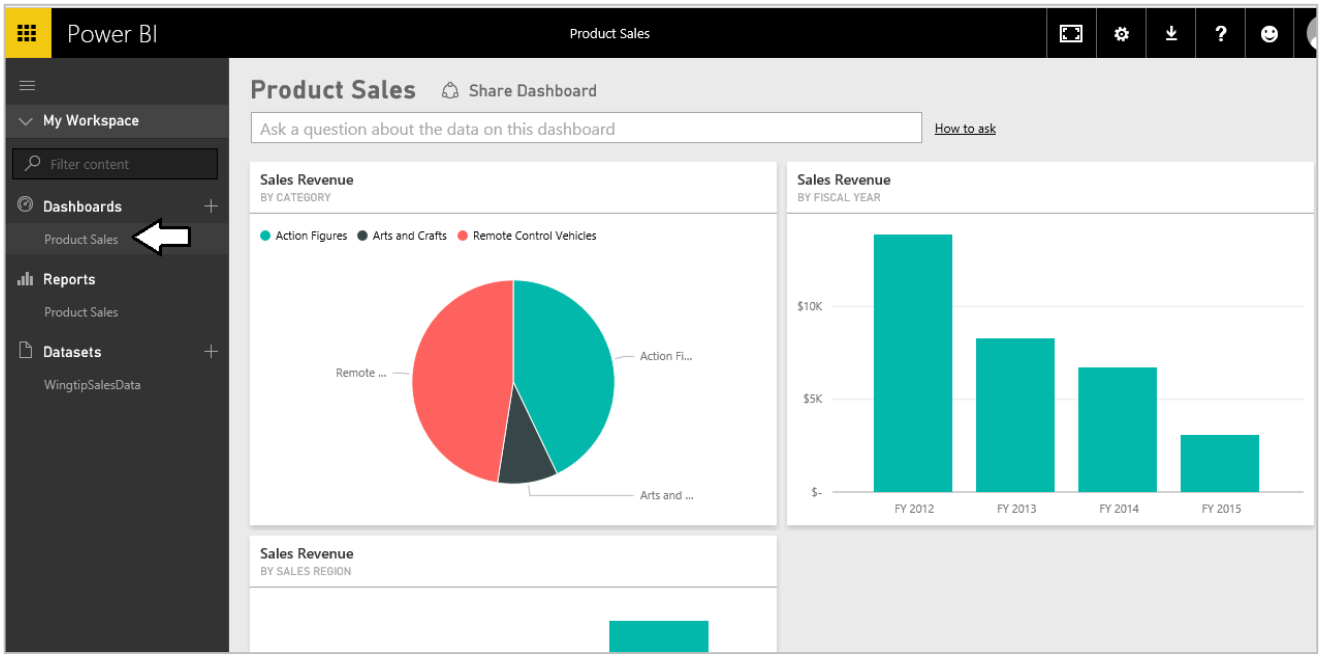



6. Remain on the **Sales by Product** page of **Product Sales** report and follow the same steps to pin the bar chart visualization showing sales revenue by sales region to the **Product Sales** dashboard.

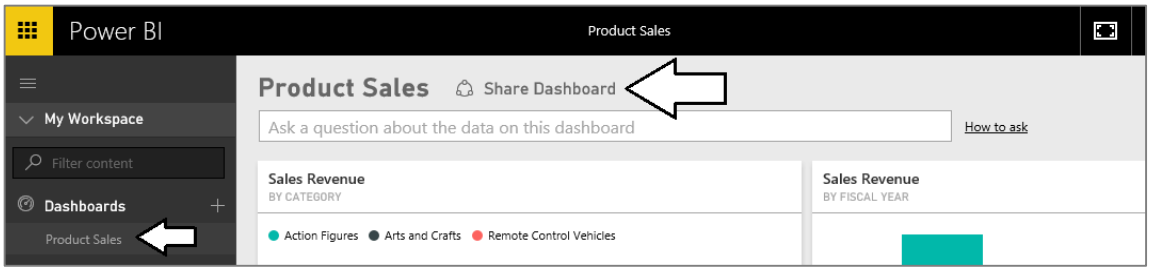



7. Click on the **Product Sales** link in the **Dashboards** section of the left navigation menu to display the **Product Sales** dashboard. You should be able to verify that you see three tiles that have been created from the three report visualization that you pinned to this dashboard.

© Critical Path Training. 2015. All Rights Reserved

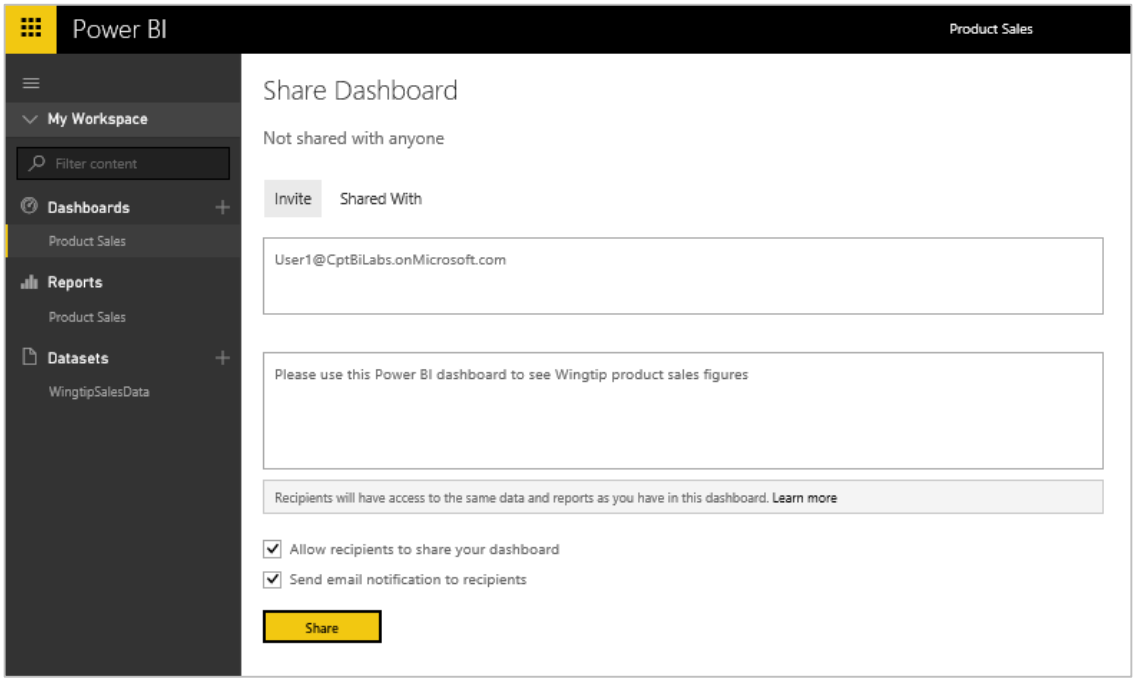



8. Note that you can move or resize the tiles inside the dashboard. This is due to the fact that you are the dashboard author and you are in dashboard edit mode.
9. Experiment by clicking on the tiles in the dashboard. You will find that clicking a tile will navigate the user to the report and page that contains the visualization which was pinned to the dashboard.
10. Now, it is time to test sharing a Power BI dashboard. Start by navigating to the **Product Sale** dashboard and clicking the **Share Dashboard** link at the top of the page.

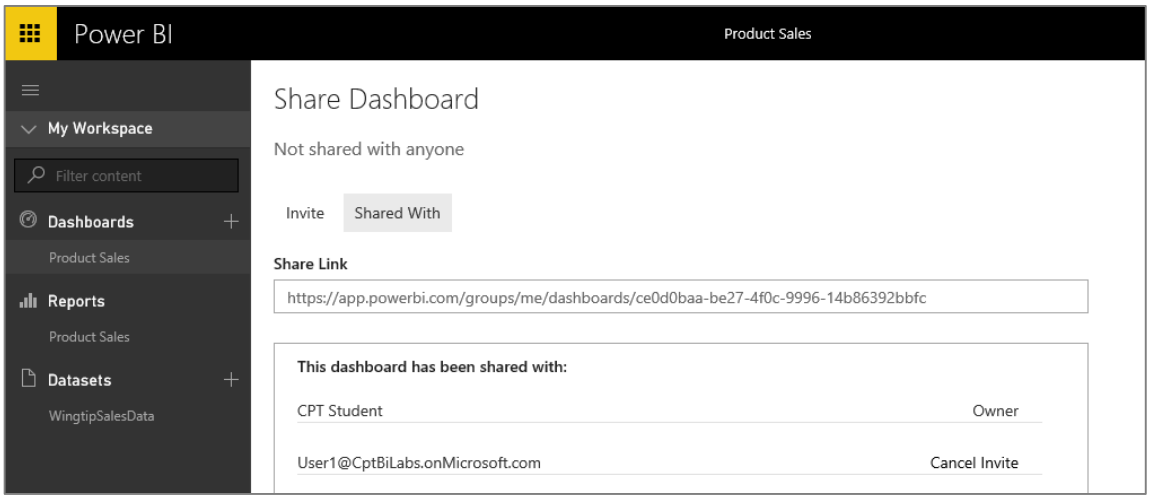



11. At this point, you are prompted with the **Share Dashboard** page where you can enter information about the users and/or groups with which you want to share your new dashboard. Enter the email address of the secondary user account that you created earlier in setup task five. After you have entered the email address, click the **Share** button at the bottom of the page.

© Critical Path Training. 2015. All Rights Reserved

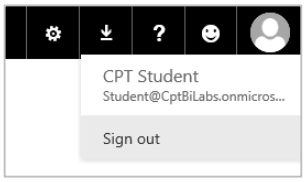



12. Once the dashboard has been successfully shared, you should be able to confirm that the secondary user has access to this dashboard by examining the **Shared With** tab of the **Share Dashboard** page as shown in the following screenshot.



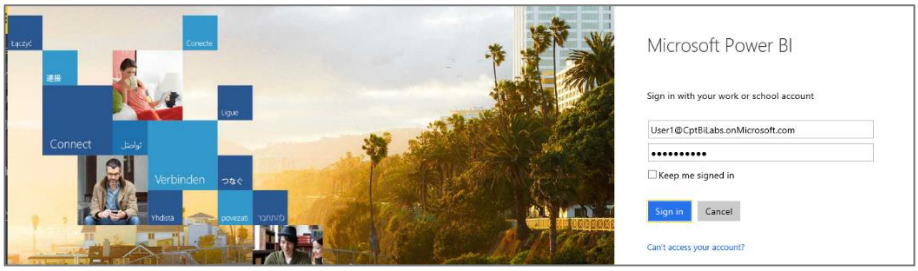

Now you have completed the steps to share the dashboard. The final step is to test the experience of a user who is not the dashboard author, but instead a dashboard consumer. This will require that you sign out of the Power BI service and then sign back in under the identity of the secondary user account. By accessing the shared dashboard in this fashion, you will be able to observe the typical experience of a dashboard consumer when accessing a dashboard that has been shared by another user.

13. Drop down the user menu from the top, right-hand corner of the page and click the Sign out command.

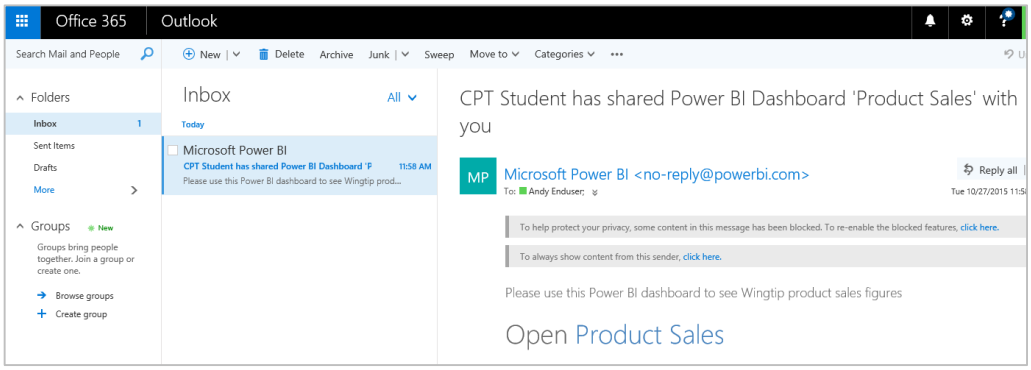



© Critical Path Training. 2015. All Rights Reserved

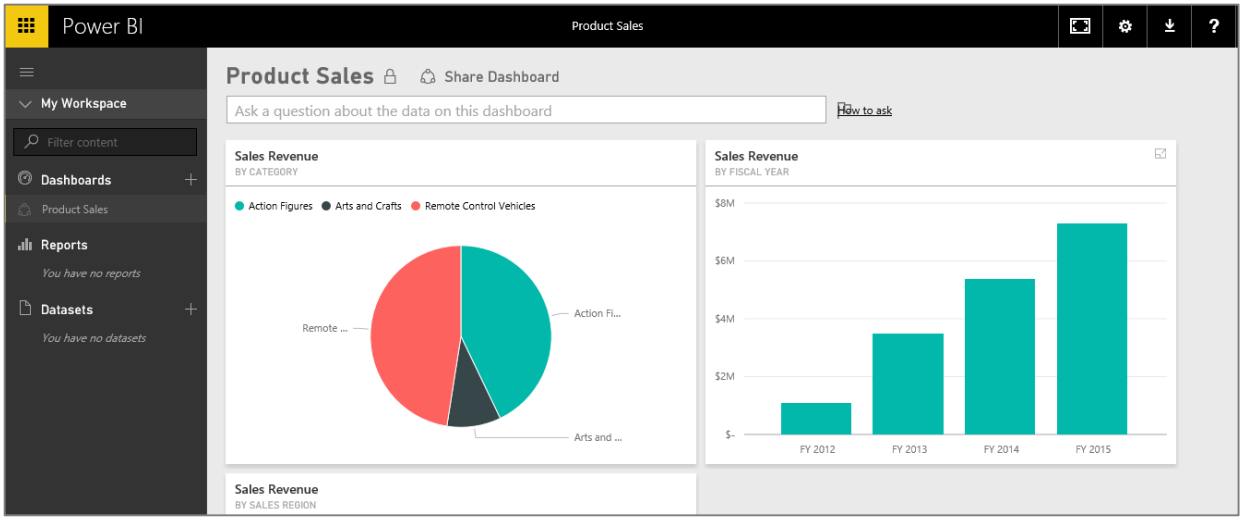
14. Now, sign back in using the account name and the password of the secondary user account you created earlier.




15. Once you have signed in, click the **Mail** tile in the Application Launcher accessible via the **Waffle Button**.
16. Once you are able to access the inbox for the secondary user, you should be able to verify that the Power BI service has sent this user a notification email message informing the user that the Product Sales dashboard that has been shared. In the body of the email message, click the **Open Product Sales** link to navigate to the new dashboard.

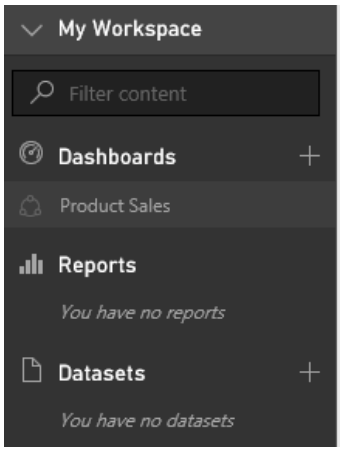



17. At this point, you will be taken to a page which shows the personal workspace for the current user and displays the page for the Product Sale dashboard.



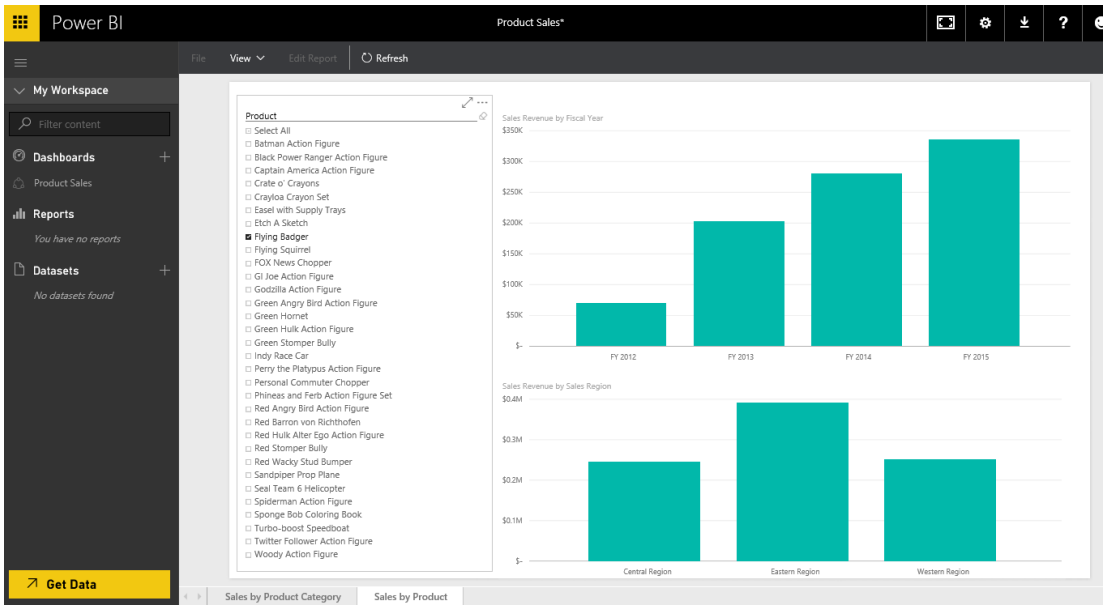

18. Take note that this user can see the **Product Sales** dashboard link in the left navigation menu but there are not links for any reports or datasets.

© Critical Path Training. 2015. All Rights Reserved




Note that Power BI does not include links to provide the dashboard consumer with direct access to the report or the dataset behind the dashboard. However, Power BI does supply the dashboard consumer with indirect access to the report and the dataset behind the dashboard. It's just that the dashboard consumer can only access the report and dataset by interacting with the dashboard. One key benefit is that this approach keeps the left navigation less cluttered when the user is accessing many different shared dashboards.

19. Navigate to the **Product Sales** dashboard and observe that the dashboard page is in read-only view. The user is not able to delete, move or resize any of the tiles. Only the dashboard author is able to edit this dashboard page.
20. On the **Product Sales** dashboard, experiment by clicking on the dashboard tile showing sales revenue by fiscal year. This will allow you to navigate to the **Sales by Product** page of the **Product Sales** report. Interact with the report page by using the slicer to filter and analyze the underlying data and see how individual products are selling from year to year..




Congratulations. You have made it to the end of this setup guide and you have now created and configured a test environment in which you can begin to create and implement custom BI solutions using the Power BI platform.

© Critical Path Training. 2015. All Rights Reserved

# Module 02: Getting Started with Power BI Desktop

## Module Description

This module provides an overview of Power BI Desktop, Microsoft's primary tool for creating custom BI solutions using the Power BI service. Students will learn how to install and configure Power BI Desktop as well as how to create datasets by importing data from external sources. The module provides students with an introductory tour of how to get around in Power BI Desktop as well as how to import data and to create datasets. Students will learn about the PBIX file format used by Power BI Desktop as well as the process used to create dashboards and reports and to publish them to the Power BI service.

## Module Agenda

- The BI Project Lifecycle
- Getting Started with Power BI Desktop
- Using Power Query to Import Data
- Using Power Pivot to Model Data
- Publishing Power BI Desktop Projects

## Topics Covered

Project Lifecycle for a Custom BI Solution
Working with Power BI Desktop
Installing Power BI Desktop
Launching Power BI Desktop
Projects and PBIX Files
Getting Around in Power BI Desktop
Data View
Relationship View
Report View
Creating Queries with Power Query
Query Editor Window
Example Query Step: Choose Columns
The Project's Data Model
Creating Calculated Columns
Creating Measures
Publishing a Power BI Desktop Project
A Published Power BI Desktop Project

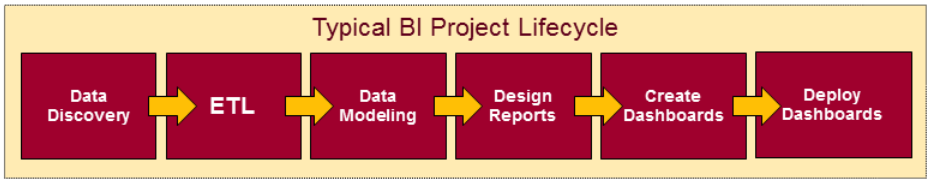
## Instructor Demos

- Getting Up and Running with Power BI Desktop
- Importing Data using Power Query
- Modeling Data using Power Pivot
- Publishing a Data Analysis Project using Power BI Desktop

© Critical Path Training. 2015. All Rights Reserved

# Project Lifecycle for a Custom BI Solution

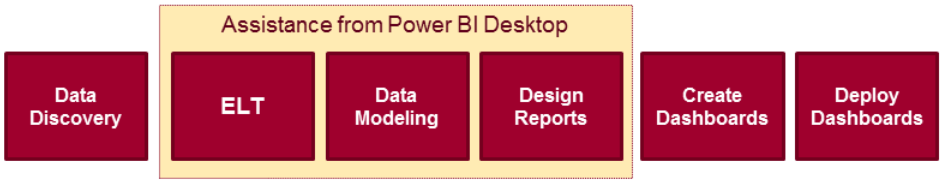
- Lifecycle of a typical BI project includes...
  - Discover where the data lives
  - Extract, transform and load (ETL) data
  - Model data to create dataset for analytics and reporting
  - Design and implement reports on top of dataset
  - Consolidate reports to one or more dashboards
  - Package project artifacts for deployment
  - Deploy to production environment




© Critical Path Training. 2015. All Rights Reserved

# Working with Power BI Desktop

- Power BI Desktop focuses on three phases
  - Power Query features used for ETL
  - Power Pivot and DAX used for data modeling
  - Report creation using the Power BI report designer



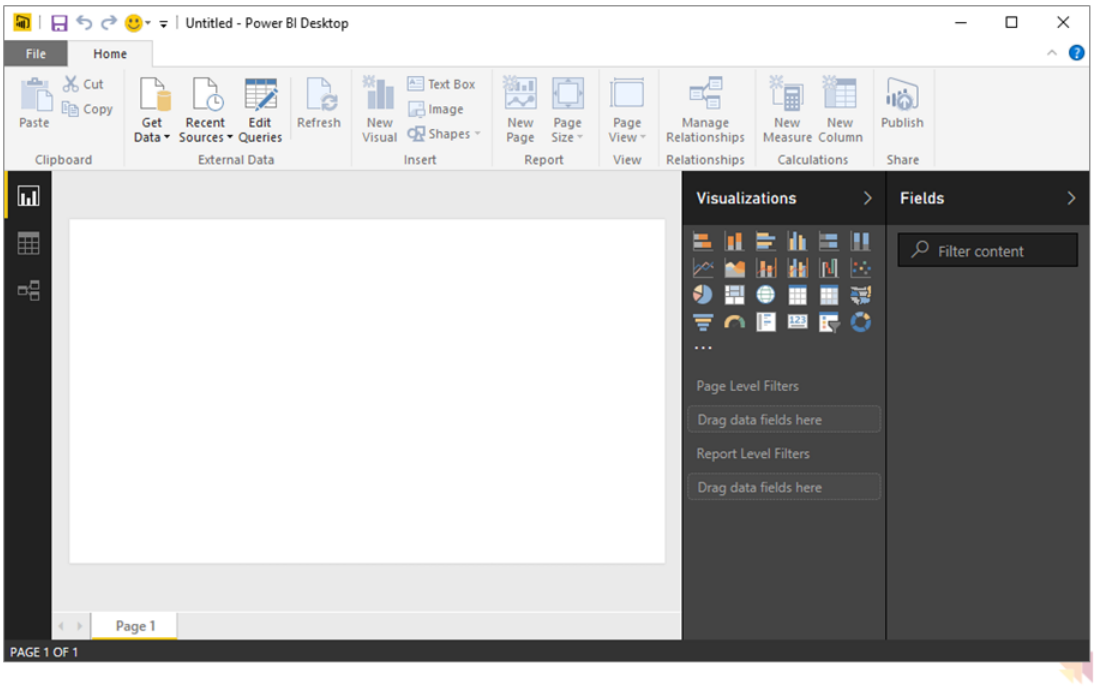

© Critical Path Training. 2015. All Rights Reserved

# Installing Power BI Desktop

- Power BI Desktop installs as click-once application
  - Can be installed when logged into Power BI service

© Critical Path Training. 2015. All Rights Reserved

# Launching Power BI Desktop




© Critical Path Training. 2015. All Rights Reserved

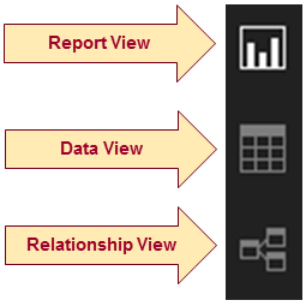
## Projects and PBIX Files

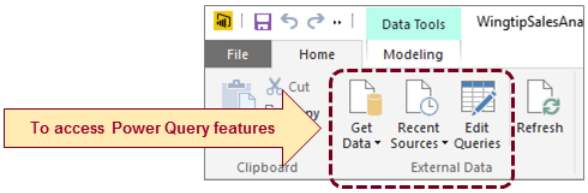
- Power BI Desktop projects saved using PBIX files
  - PBIX file contains queries created with Power Query
  - PBIX file contains data imported using queries
  - PBIX file contains data modeling definitions
  - PBIX file contains reports

© Critical Path Training. 2015. All Rights Reserved

# Getting Around in Power BI Desktop

- What do you need to learn to use Power BI Desktop?
  - Power query features for importing data
  - Power Pivot features for modeling data
  - Report designer for creating reports
- Navigating between view modes

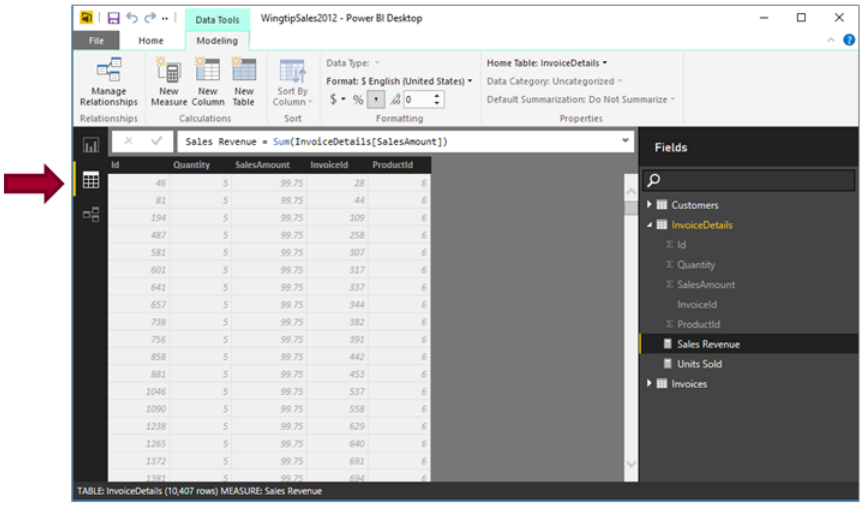



© Critical Path Training. 2015. All Rights Reserved

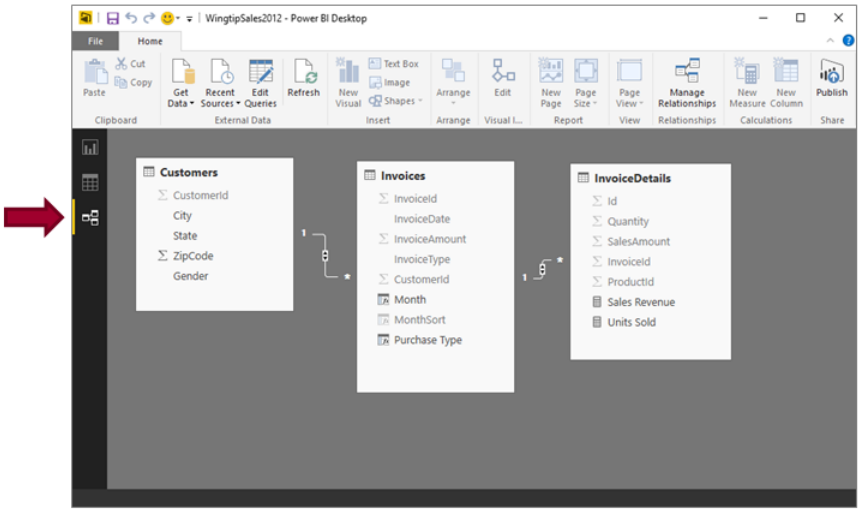
# Data View

- Primary view used when data modeling with Power Pivot
  - Data view displays columns and rows for each table
  - You can extend tables with calculated columns and fields



# Relationship View

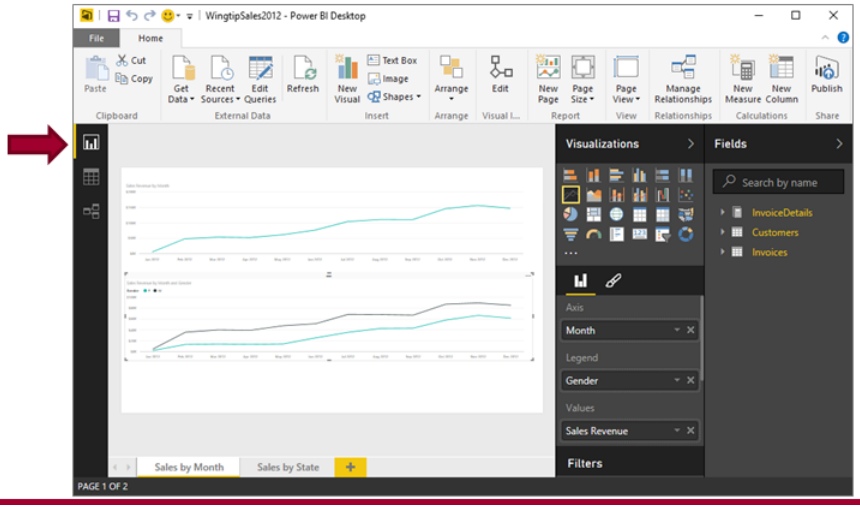
- Displays tables, fields and relationships
  - Used to view tables, fields and relationships in project's dataset
  - Used to create relationships when importing new tables




© Critical Path Training. 2015. All Rights Reserved

# Report View

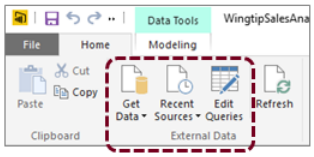
- Report view displays report of current project
  - Report designer is the same as in the Power BI service
  - Dataset appears in consumer mode - not author mode

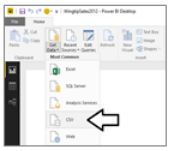



© Critical Path Training. 2015. All Rights Reserved

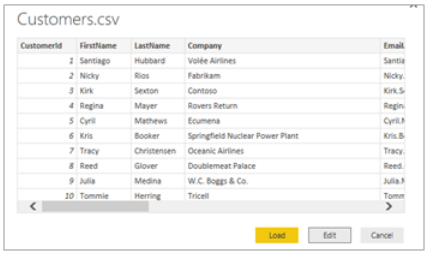
# Creating Queries with Power Query

- Power BI Desktop support Power Query features
  - Creating a new query



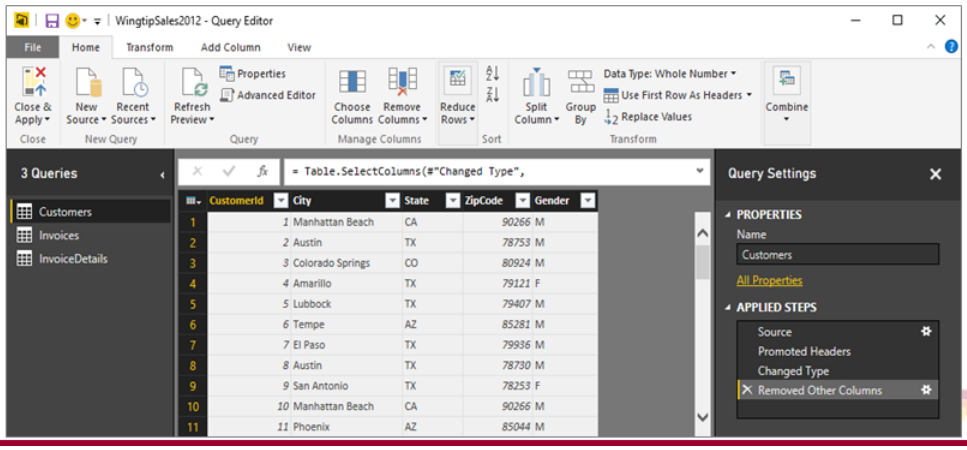
  - The difference between Load versus Edit
    - **Load** command creates the query and runs it as is
    - **Edit** command creates query & opens it in Query Editor window




© Critical Path Training. 2015. All Rights Reserved

# Query Editor Window

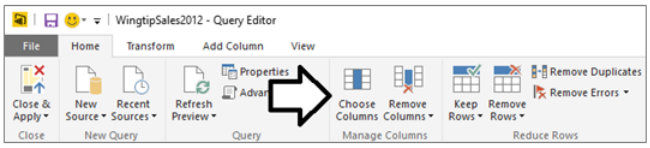
- Power BI Desktop provides Query Editor window
  - Provides Power Query features for designing queries
  - Displays list of all queries in project on the left
  - Displays properties and step for selected query on right

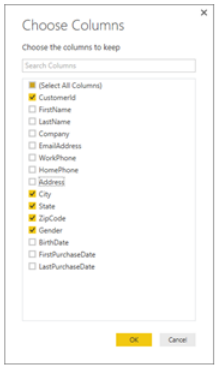


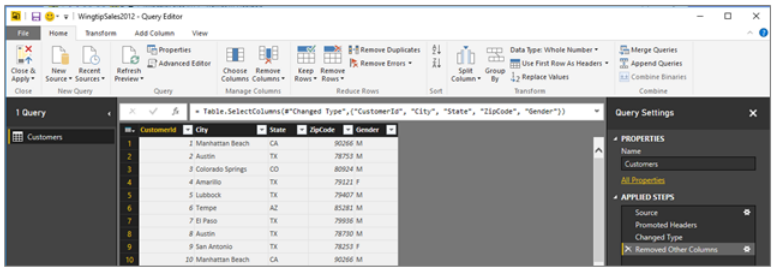

© Critical Path Training. 2015. All Rights Reserved

# Example Query Step: Choose Columns

- Select which columns to include in query output



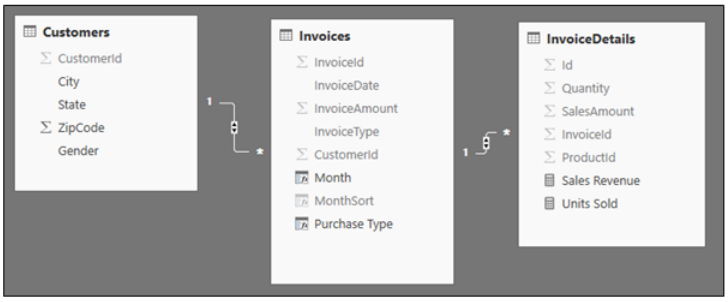




© Critical Path Training. 2015. All Rights Reserved

# The Project's Data Model

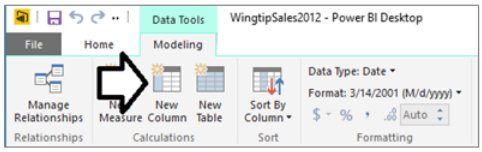
- Each PBIX project has a data model
  - Data model contains tables and relationships
  - Tables contain three different types of fields
    - Native columns
    - Calculated columns
    - Measures



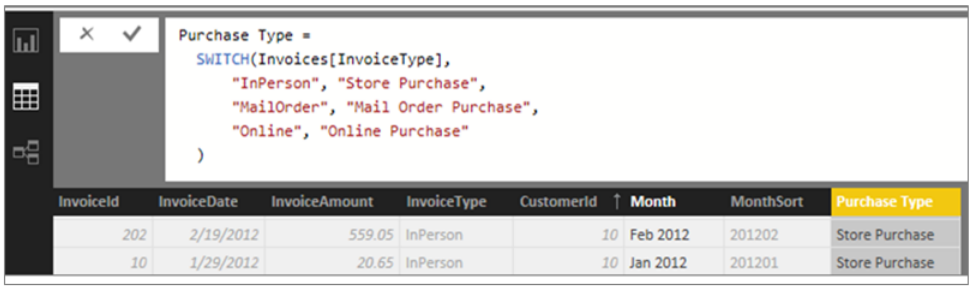

© Critical Path Training. 2015. All Rights Reserved

# Creating Calculated Columns

- Calculated column can be added to table



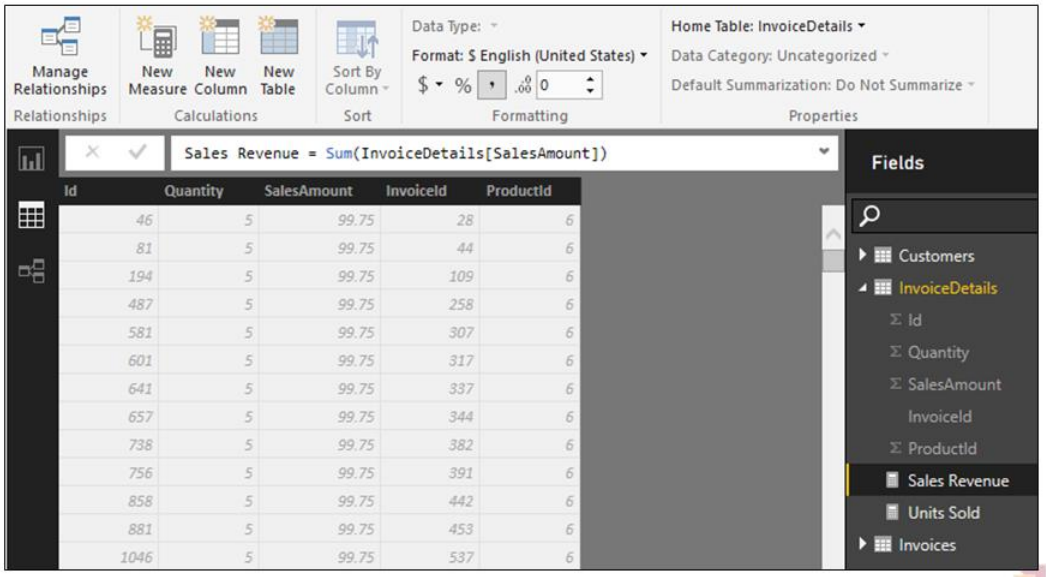

- Calculated columns created with DAX expressions




© Critical Path Training. 2015. All Rights Reserved

# Creating Measures

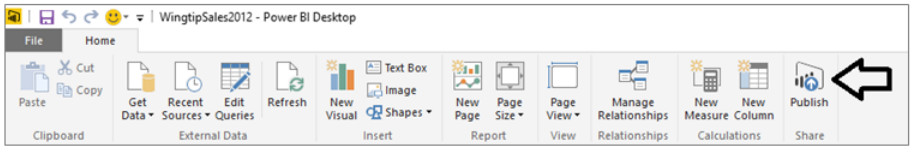
- Measure defined as DAX expression with format settings



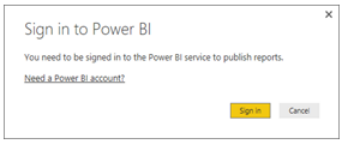

© Critical Path Training. 2015. All Rights Reserved

# Publishing a Power BI Desktop Project

- Power BI Desktop provides **Publish** command
  - Used to publish project to Power BI service



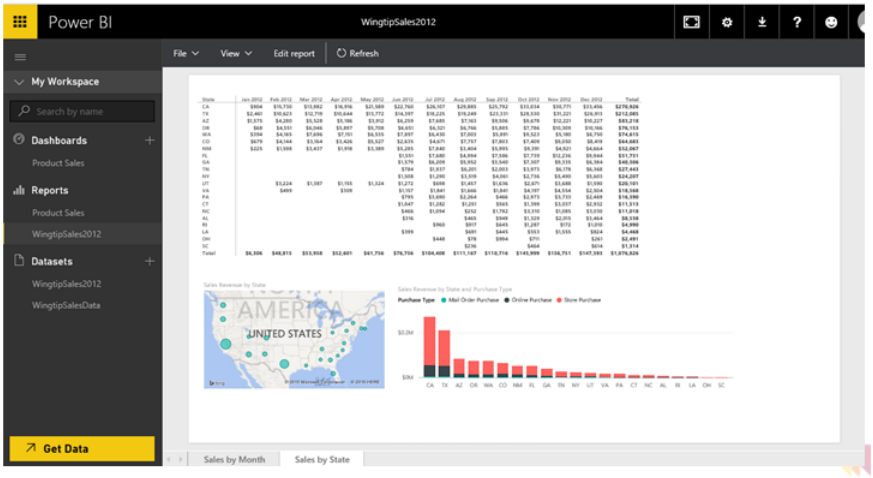

  - Requires logging into your Office 365 account



  - Published articles added to your personal workspace

© Critical Path Training. 2015. All Rights Reserved

# A Published Power BI Desktop Project



# Module 02 Lab: Working with Power BI Desktop

**Lab Time**: 60 minutes

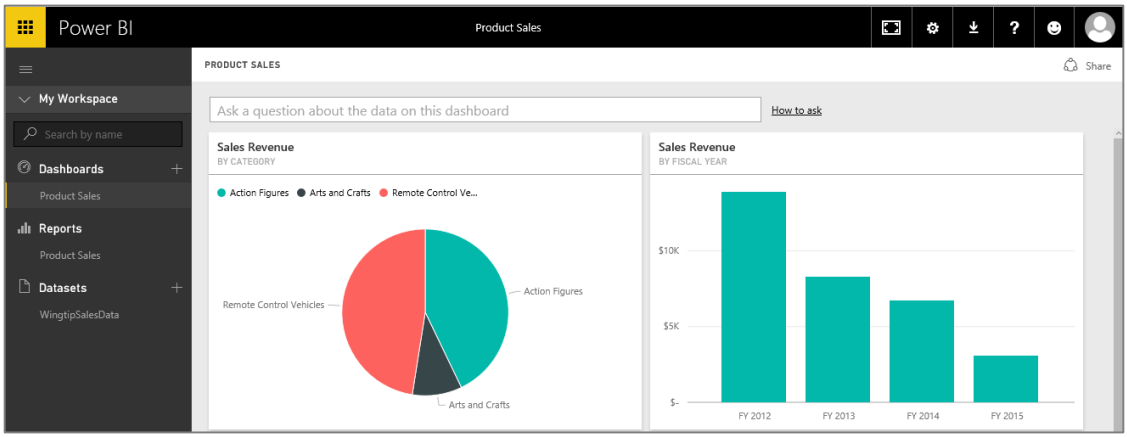
**Lab Folder**: C:\Student\Modules\PowerBIDesktop\Lab

**Lab Overview**: In this module you will learn how to install and begin working with Power BI Desktop. You will learn how to import data using Power Query to create a dataset and you will also learn how model the data within the dataset using Power Pivot. After modeling the data to make it better suited for analysis, you will create a report and publish it to the Power BI service.

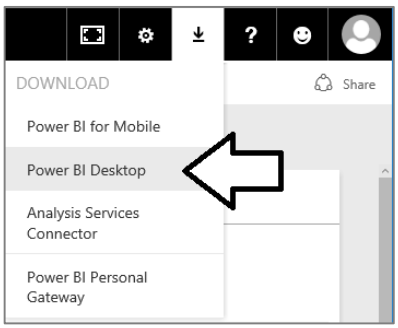
## Exercise 1: Installing and Configuring Power BI Desktop

In this exercise you will download and install Power BI Desktop. Note that if Power BI desktop is already installed on your student workstation, you can move ahead to exercise 2.

1. Launch the browser and navigate to the landing page of the Power BI service at https://app.powerbi.com. Make sure you log on to the Power BI service using the primary Office 365 account you used to create the Office 365 tenancy and not the secondary user account you used at the end of the previous lab to test dashboard sharing. Inside your personal workspace, you should see the dataset, report and dashboard you created in the previous lab.




2. On the top right of the Power BI service window, drop down the **Downloads** menu and click the **Power BI Desktop** menu command to begin the download of the installation file.

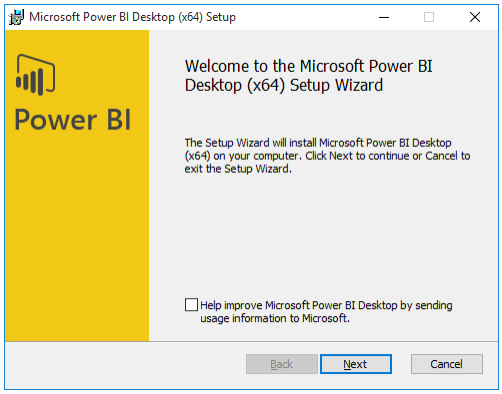



3. Wait for the MSI file to download.

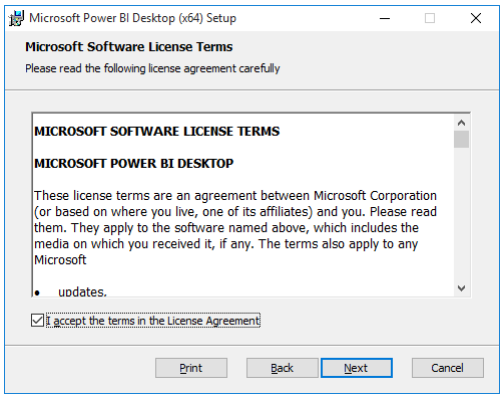
4. Once the file has downloaded, click the **Run** button to begin the installation of Power BI Desktop.

© Critical Path Training. 2015. All Rights Reserved

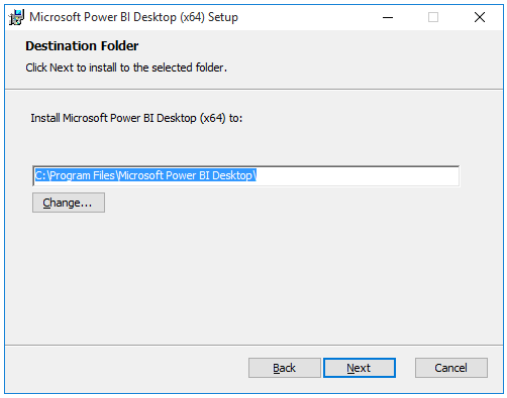
5. When you see the Welcome screen, click **Next** to continue with the installation.




6. Click the checkbox to accept the license agreement and click **Next**.

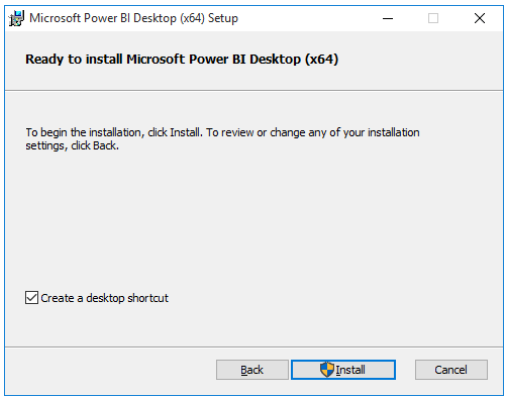



7. Accept the default location for the installation and click **Next**.



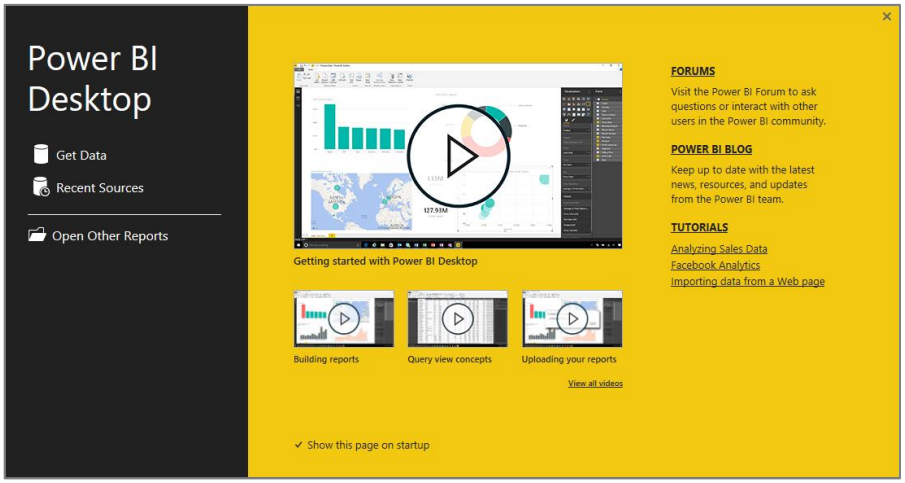

8. On the next screen, click **Install**.

© Critical Path Training. 2015. All Rights Reserved



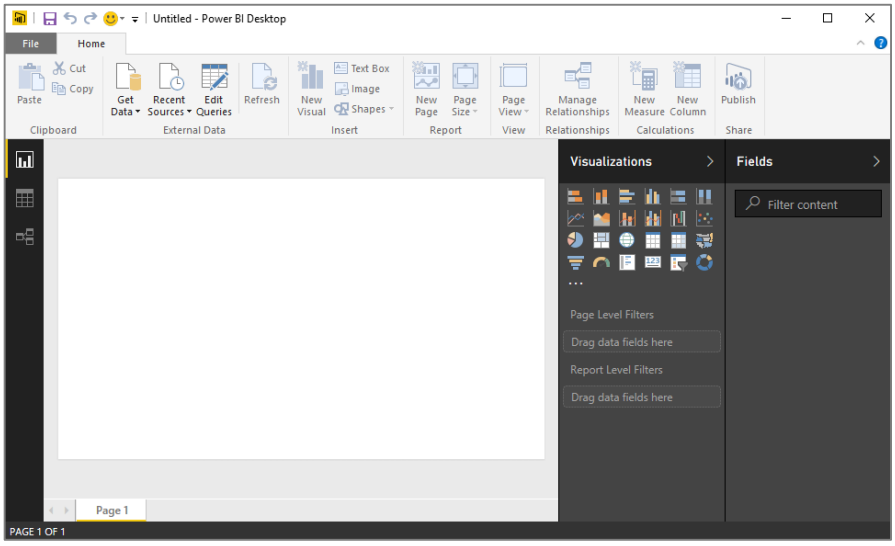

9. When you see the **Completed the Microsoft Power BI Desktop Setup Wizard** screen, click **Finish** to launch Power BI Desktop.

10. When Power BI Desktop launches for the first time, it displays a Welcome screen as shown in the following desktop. Click the (**X**) button in the upper right corner to close this window.




11. At this point, you should have Power BI Desktop running as shown in the following screenshot. You are now ready to move ahead to the next exercise.

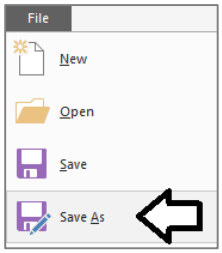
© Critical Path Training. 2015. All Rights Reserved




## Exercise 2: Importing Data into Power BI Desktop using Power Query

In this exercise you will use Power BI Desktop to import sales data from three separate CSV files. Moving through the steps of this exercise will give you a chance to begin working with the Power Query features that are built into Power BI Desktop.

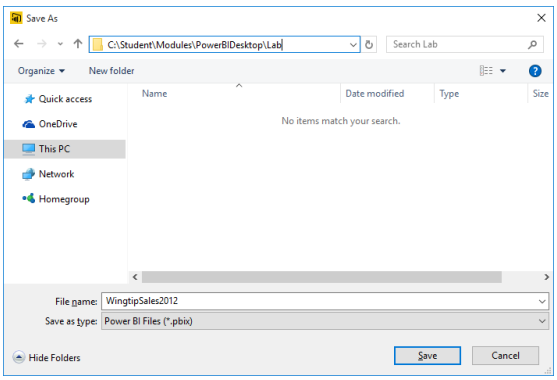
1. Open Power BI Desktop if it is not already open.
2. Power BI Desktop should now be open with a new and empty project.
3. Before beginning your work, save the empty project as a PBIX file.
   a) Drop down the **File** menu and click the **Save As** command.




   b) Save the PBIX file as **WingtipSales2012.pbix** using the following path location.

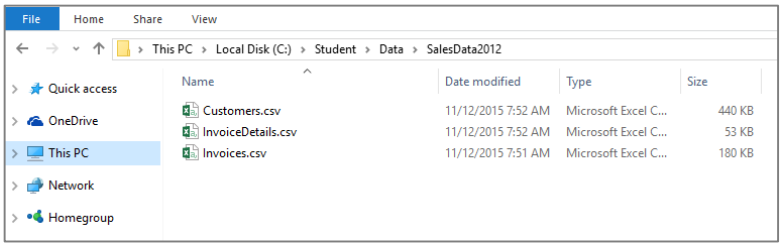
```
C:\Student\Modules\PowerBIDesktop\Lab\WingtipSales2012.pbix
```

   c) Click the **Save** button on the **Save As** dialog to save the PBIX file.

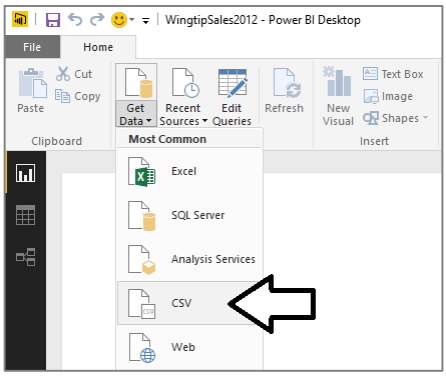



© Critical Path Training. 2015. All Rights Reserved

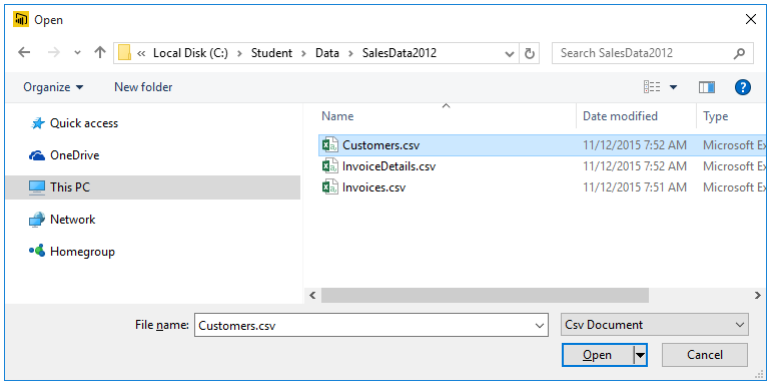
4. Using Windows Explorer, open the folder at **C:\Student\Data\SalesData2012** and examine what's inside. You should observe that there are three CSV files named **Customers.csv**, **Invoices.csv** and **InvoiceDetails.csv**. Over the next few steps, you will use these three files to import data into Power BI Desktop to create a new dataset.




5. Return to Power BI Desktop and examine the ribbon. Drop down the **Get Data** menu button and select the **CSV** command.

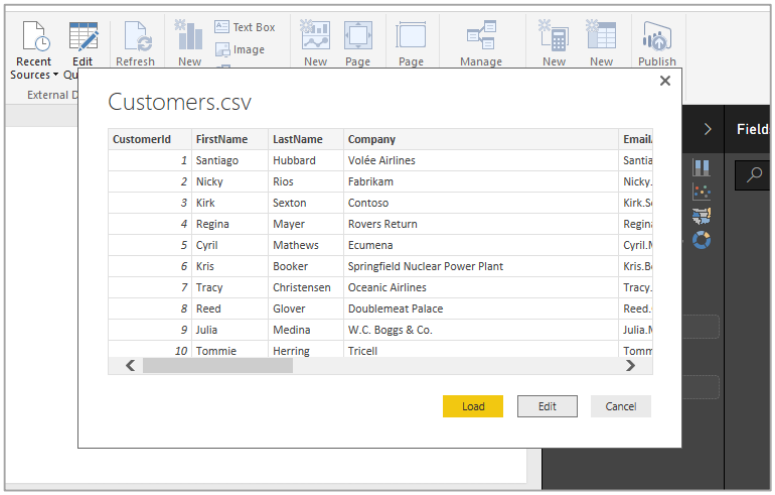



6. In the **Open** dialog, navigate to the folder at **C:\Student\Data\SalesData2012** and select the file named **Customers.csv**. Click the **Open** button in the bottom right of the dialog to begin the import process.

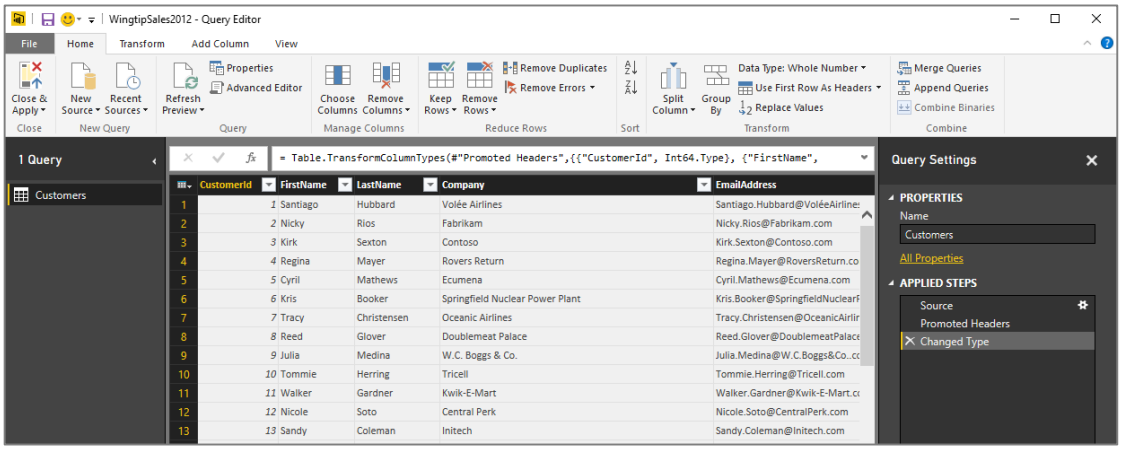



7. At this point you are presented with the **Customers.csv** dialog which displays a **Load** button, an **Edit** button and a **Cancel** button. Click the **Edit** button to view the data from **Customers.csv** in the Power Query Editor window.

© Critical Path Training. 2015. All Rights Reserved

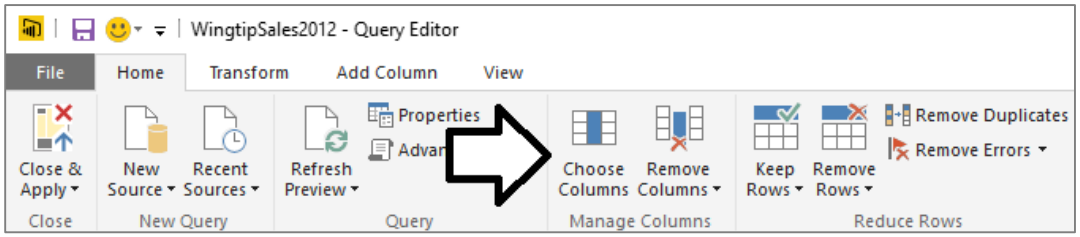



8. You should now be looking at the data from **Customers.csv** in the Query Editor window as shown in the following screenshot. If you examine the **Query Settings** task pane on the right-hand side, you can see that the query is named **Customers** and there are three steps in the **Applied Steps** list.




9. The **Customers** query contains several unneeded columns that will not be used to analyze sales data. In this step you will remove these unneeded columns to reduce the amount of data that is loaded into the data model.

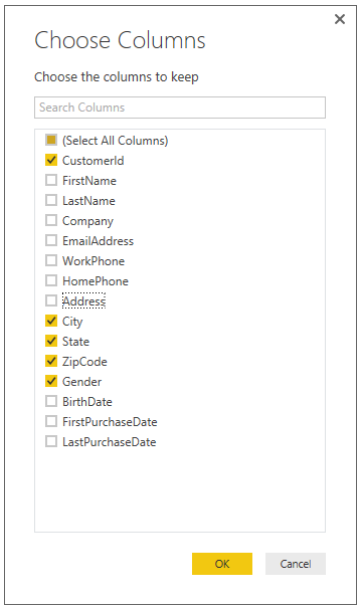
   a) In the ribbon of the **Query Editor** window, click the **Choose Columns** button.



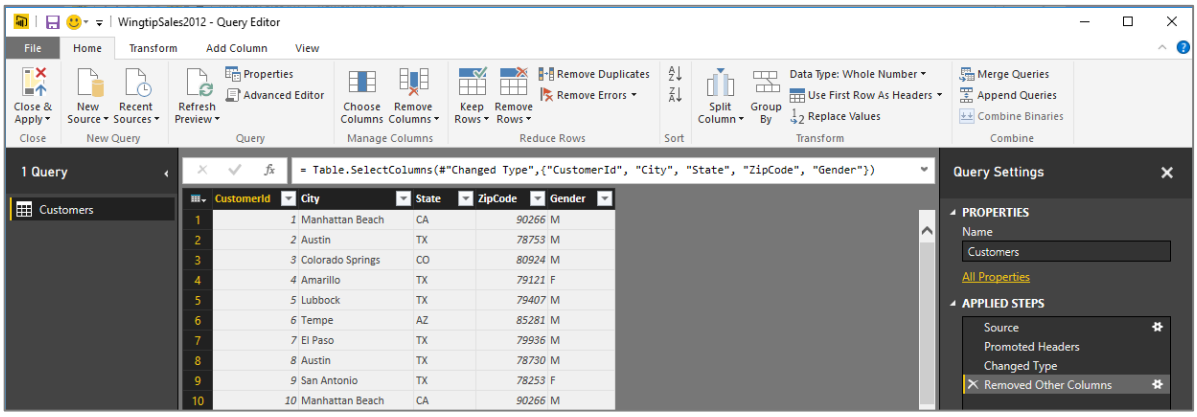

   b) In the **Choose Columns** dialog, begin be unselecting all columns.

   c) Select the columns named **CustomerId**, **City**, **State**, **ZipCode** and **Gender** as shown in the following dialog.

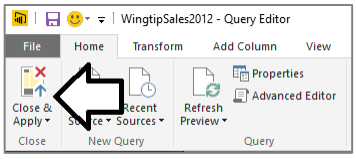
© Critical Path Training. 2015. All Rights Reserved



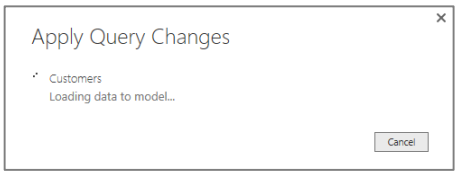

10. Once you have selected the desired columns, click the **OK** button to close the **Choose Columns** dialog. You should be able to observe that the Query Editor window now displays data for the columns you selected.




11. Now it's time to load the data from **Customer.csv** into the data model. Accomplish this by clicking the **Close & Apply** button in the ribbon of the Query Editor window to close the Query Editor window and import the data.

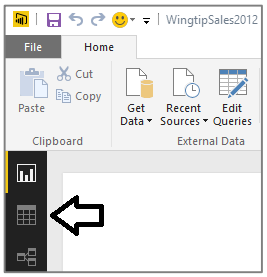



12. Power BI Desktop displays the **Apply Query Changes** dialog while it's executing the query and importing the data. Note that the **Apply Query Changes** dialog will close automatically after the data has been imported.

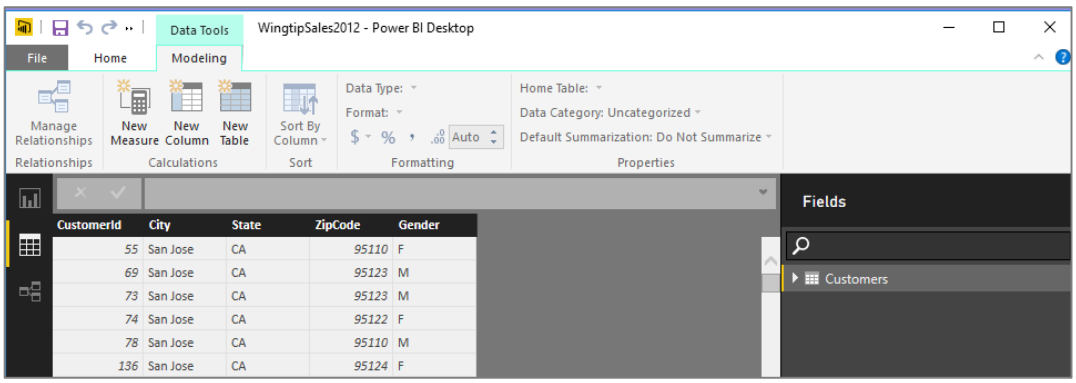



© Critical Path Training. 2015. All Rights Reserved

13. You should now be looking at the main window of Power BI Desktop in report view which displays a new and empty report. Locate the three buttons on the sidebar on the left-hand side of the screen. Click the button in the middle to move from report view to data view.

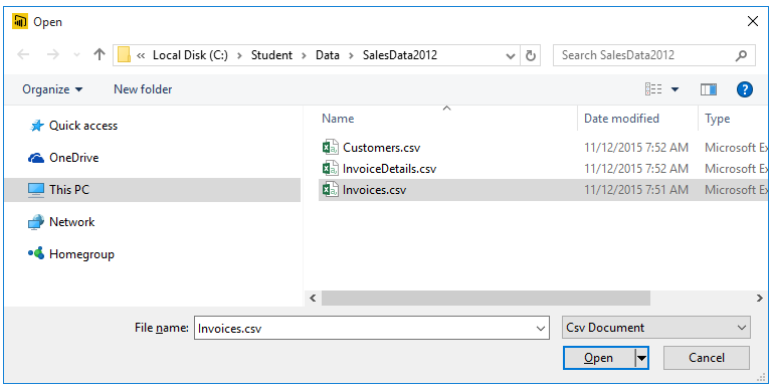



14. Once you have moved over to data view, you should be able to verify that the **Customers** table has been created. On the right side of the window, the **Fields** list shows the **Customers** table which is selected. The data from the **Customers** table is shown in a grid view in the middle of the screen.




You might observe that the data from the **Customers** table is not sorted in an intuitive fashion. For example, you might expect the default sort order to be the same as the underlying CSV file which orders customers by **CustomerId**. Instead, the table has a seemingly random sort order which is a side effect of the way data is stored in memory given the column-based nature of the tabular model. You can right-click the column header for the **CustomerId** column and run a command to sort the data in the table if it makes you feel better, but this action will not really have any effect because changes to the sort order are only temporary and they are not saved as part of the underlying data model.

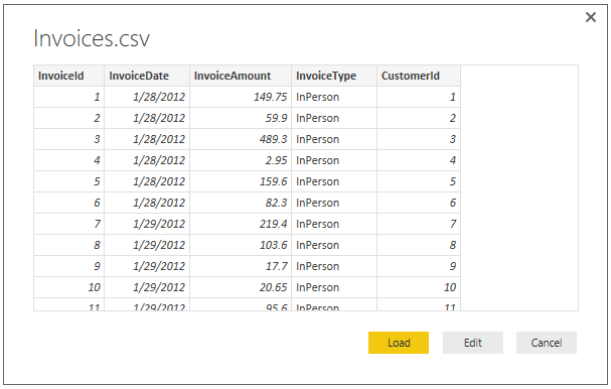
15. You have imported one table into the data model. Now you will import a second table to get additional data on invoices.
    a) In the Ribbon of the Power BI Desktop window, select the **Home** tab.
    b) Drop down the **Get Data** menu button and select the **CSV** command.
    c) In the **Open** dialog, navigate to the folder at **C:\Student\Data\SalesData2012** and select the file named **Invoices.csv**.




© Critical Path Training. 2015. All Rights Reserved

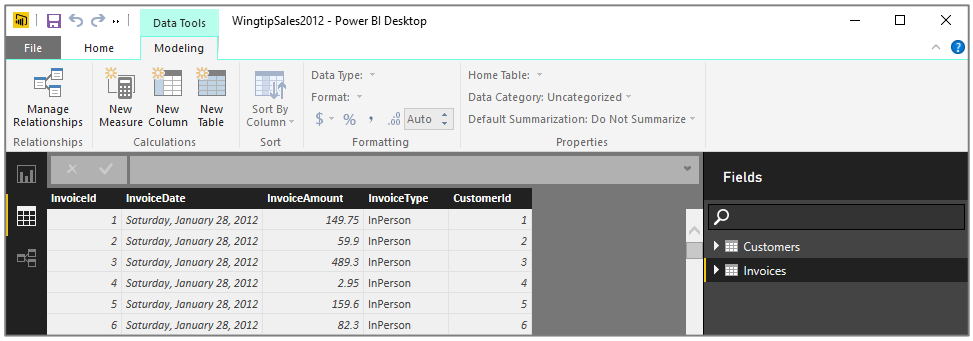
d) Click the **Open** button in the bottom right of the dialog to begin the import process.

e) In the **Invoices.csv** dialog, click the **Load** button to execute the query and import data to create the **Invoices** table.




| InvoiceId | InvoiceDate | InvoiceAmount | InvoiceType | CustomerId |
|---|---|---|---|---|
| 1 | 1/28/2012 | 149.75 | InPerson | 1 |
| 2 | 1/28/2012 | 59.9 | InPerson | 2 |
| 3 | 1/28/2012 | 489.3 | InPerson | 3 |
| 4 | 1/28/2012 | 2.95 | InPerson | 4 |
| 5 | 1/28/2012 | 159.6 | InPerson | 5 |
| 6 | 1/28/2012 | 82.3 | InPerson | 6 |
| 7 | 1/29/2012 | 219.4 | InPerson | 7 |
| 8 | 1/29/2012 | 103.6 | InPerson | 8 |
| 9 | 1/29/2012 | 17.7 | InPerson | 9 |
| 10 | 1/29/2012 | 20.65 | InPerson | 10 |
| 11 | 1/29/2012 | 95.6 | InPerson | 11 |

f) Make sure Power BI Desktop is in data view. You should be able to verify that the **Invoices** table has been created.



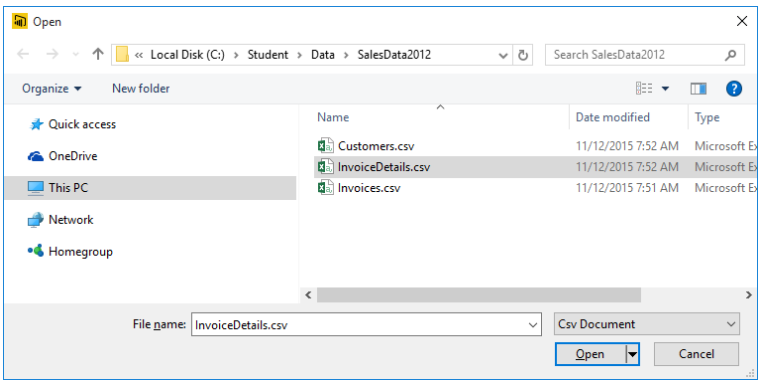

| InvoiceId | InvoiceDate | InvoiceAmount | InvoiceType | CustomerId |
|---|---|---|---|---|
| 1 | Saturday, January 28, 2012 | 149.75 | InPerson | 1 |
| 2 | Saturday, January 28, 2012 | 59.9 | InPerson | 2 |
| 3 | Saturday, January 28, 2012 | 489.3 | InPerson | 3 |
| 4 | Saturday, January 28, 2012 | 2.95 | InPerson | 4 |
| 5 | Saturday, January 28, 2012 | 159.6 | InPerson | 5 |
| 6 | Saturday, January 28, 2012 | 82.3 | InPerson | 6 |

16. Now, you will import one more table into the data model.

a) In the ribbon of the Power BI Desktop window, select the **Home** tab.

b) Drop down the **Get Data** menu button and select the **CSV** command.

c) In the **Open** dialog, navigate to the folder at **C:\Student\Data\SalesData2012** and select the file named **InvoiceDetails.csv**.

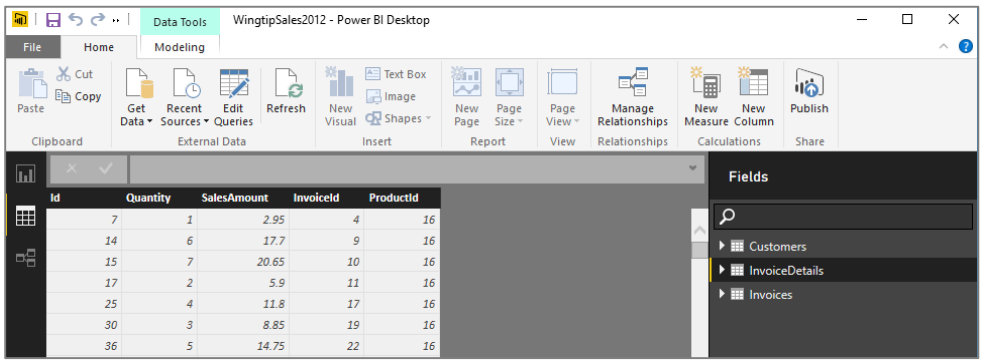



d) Click the **Open** button in the bottom right of the dialog to begin the import process.

e) In the **InvoiceDetails.csv** dialog, click **Load** to execute the query and import data to create the **InvoiceDetails** table.

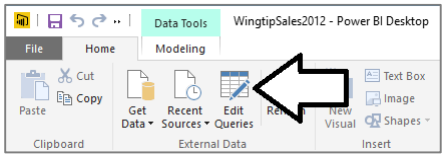
f) After the import process completes, you should be able to verify that the **InvoiceDetails** table has been created.

© Critical Path Training. 2015. All Rights Reserved

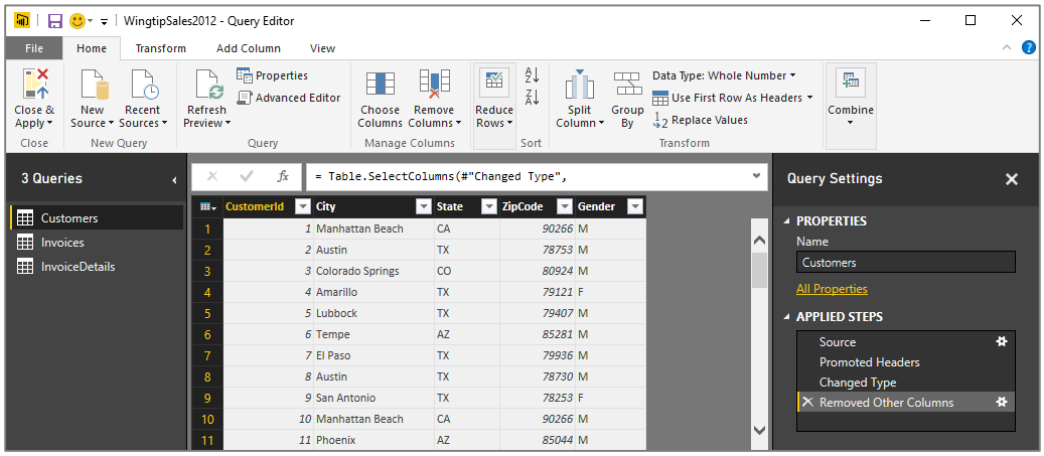



17. While you are done working with the query and data import aspects of Power BI Desktop for this particular lab exercise, it's still important that you know how to open and edit queries whenever it is required. You will now go through the steps to open the Query Editor window to see how to get back to query editing mode.

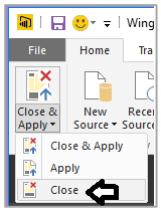
    a) In the ribbon of the Power BI Desktop window, click the **Edit Queries** button to open the Query Editor window.




    b) You should now see the Query Editor window displaying queries named **Customers**, **Invoices** and **InvoiceDetails** in the **Queries** list in the right-hand side of the screen. You can select any of these queries and continue to edit them if required. However, in this case there is no need to edit the queries any more.




    c) Close the Query Editor window by dropping down the **Close and Apply** menu button and clicking the **Close** command.




18. Save your work by clicking the Save button on the top left of the Power BI Desktop window.

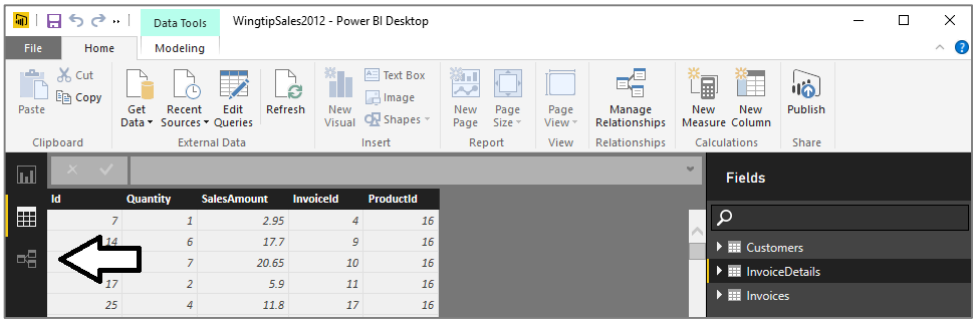
© Critical Path Training. 2015. All Rights Reserved

You have now finished importing data into your data model using Power Query. The next step is to begin modeling the data using the Power Pivot features of Power BI Desktop.

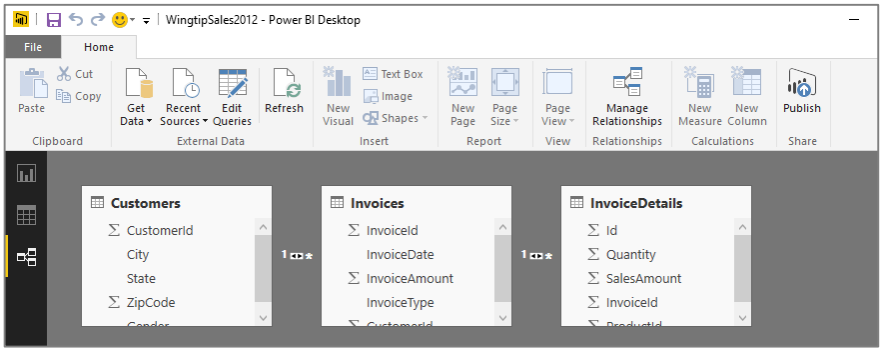
## Exercise 3: Modeling Tables in a Dataset using Power Pivot

In this exercise you will begin to model the data in the dataset you have created using the Power Pivot features of Power BI Desktop. You will begin by ensuring the appropriate relationships have been created between the three tables in the data model. After that, you will extend the tables within the data model by adding calculated columns and measures.

1. Currently, the Power BI Desktop window should be displaying in data view. Click on the bottom button in the sidebar on the left-hand side of the screen to move to relationship view.

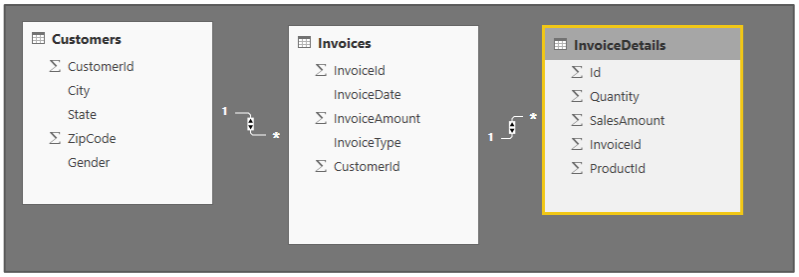



2. Once you navigate to relationship view, you should be able to see all three tables as shown in the following screenshot.




Note that Power BI desktop automatically created relationships between these tables because it determined that the **Customers** table and **Invoices** table both contain a common field named **CustomerId** and also that the **Invoices** tables and the **InvoicesDetails** table both contain a common field named **InvoiceId**.

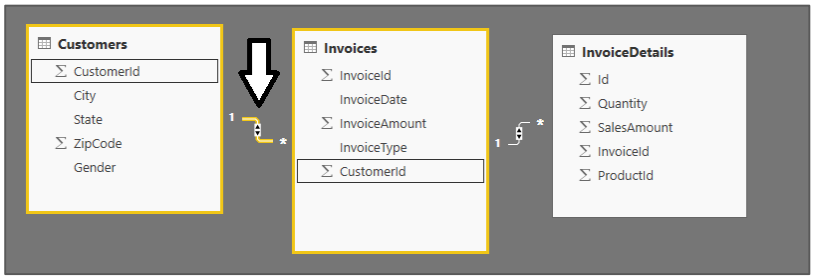
3. Using the mouse, resize each of the three tables in relationship view so you can see all the fields in each one. Also space the tables out a bit so it is easier to see the relationships between them. You can see that these relationship are displayed using a line that starts with a **1** and end with a (*****) to indicate a one-to-many relationship.



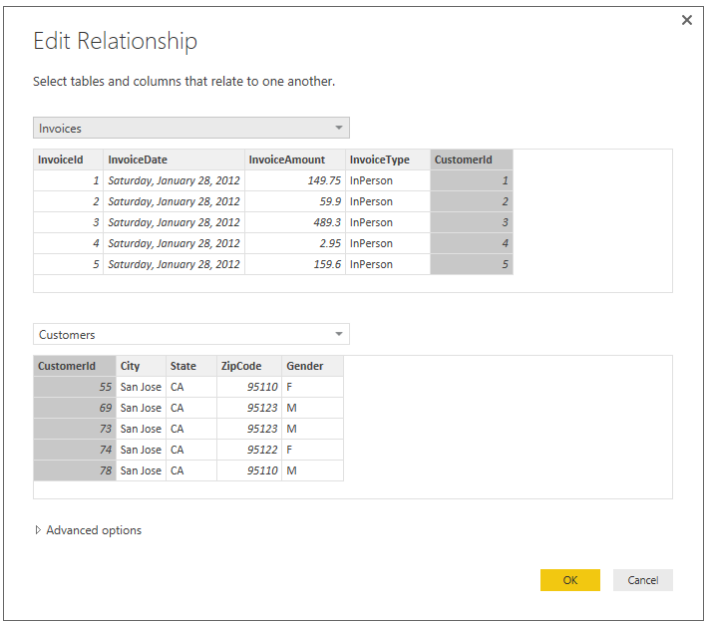

4. Take a moment to inspect the relationships that have been created.

© Critical Path Training. 2015. All Rights Reserved

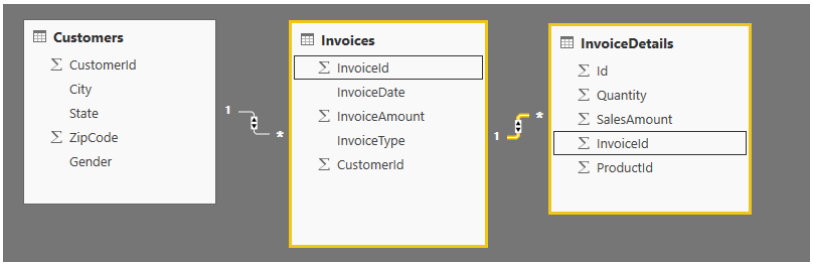
a) Click on the line that represents the relationship between the **Customers** table and the **Invoices** table. Once the relationship has been selected, you can see that the **CustomerId** field in each table becomes highlighted to indicate that those are the two fields used to define the one-to-many relationship.




b) Now, double-click on the line that represents the relationship between the **Customers** table and the **Invoices** table to display the **Edit Relationship** dialog. There is no need to change anything as you are just observing. Once you have seen what's inside this dialog, click the **OK** button to close it.

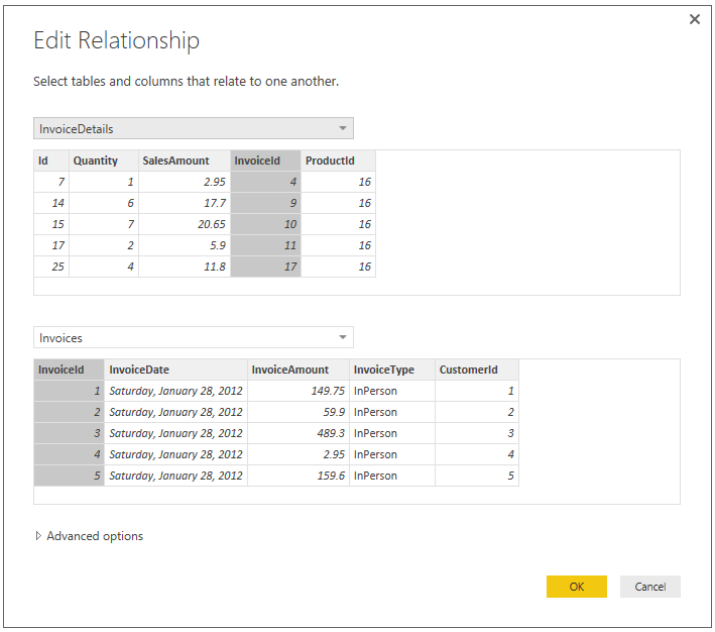



c) Click on the line that represents the relationship between the **Invoices** table and the **InvoiceDetails** table. When you do, you can see that the **InvoiceId** field in each table becomes highlighted to indicate that those are the two fields used to define the one-to-many relationship.

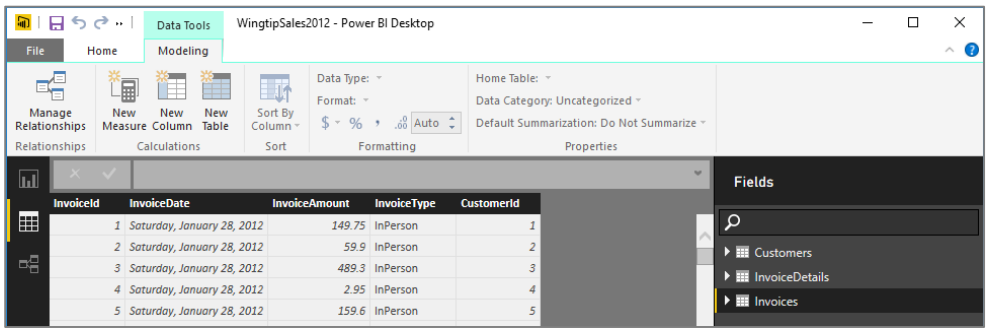



d) Now, double-click on the line that represents the relationship between the **Invoices** table and the **InvoiceDetails** table to display the **Edit Relationship** dialog. Once you have seen the dialog, click the **OK** button to close the dialog.

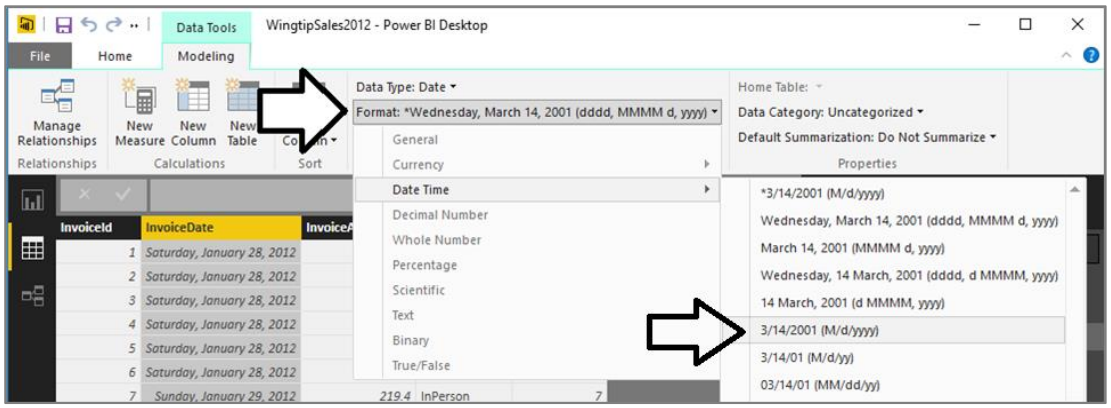
© Critical Path Training. 2015. All Rights Reserved




5. Modify the formatting on the **InvoiceDate** field of the **Invoices** table.
   a) Navigate back to data view and select the **Invoices** table from the **Fields** list.

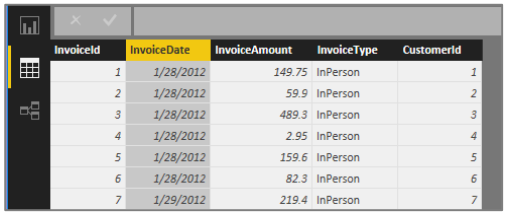



   b) Select the **InvoiceDate** column by clicking its column header.
   c) Drop down the **Format** menu in the ribbon and select the date format of **3/14/2001 (M/d/yyyy)**.



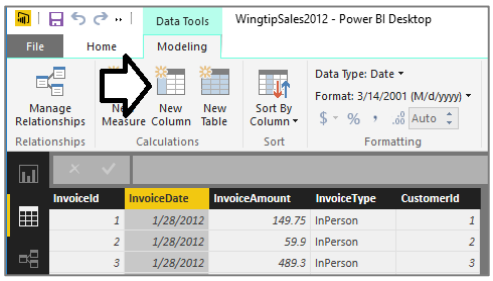

   d) The data in the **InvoiceDate** field should now be display in the new format.

© Critical Path Training. 2015. All Rights Reserved

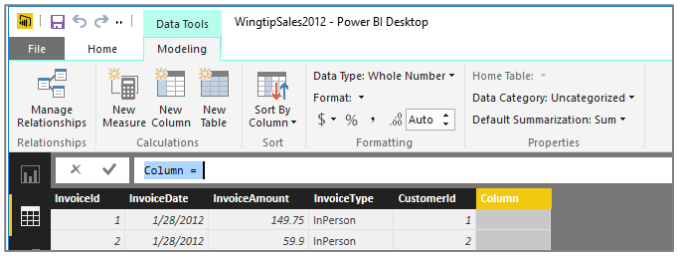


| InvoiceId | InvoiceDate | InvoiceAmount | InvoiceType | CustomerId |
|---|---|---|---|---|
| 1 | 1/28/2012 | 149.75 | InPerson | 1 |
| 2 | 1/28/2012 | 59.9 | InPerson | 2 |
| 3 | 1/28/2012 | 489.3 | InPerson | 3 |
| 4 | 1/28/2012 | 2.95 | InPerson | 4 |
| 5 | 1/28/2012 | 159.6 | InPerson | 5 |
| 6 | 1/28/2012 | 82.3 | InPerson | 6 |
| 7 | 1/29/2012 | 219.4 | InPerson | 7 |

6. Add a new calculated column to the **Invoice** table to display the current month.
   a) Click the **New Column** button in the ribbon to create a new column.



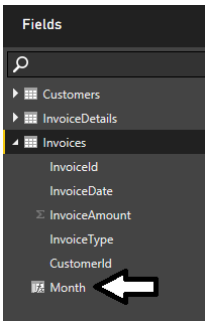

   b) The column is initially created with a generic name of **Column** and it has no formula in the formula bar.




   c) Give the new column a name of **Month** and provide a DAX formula for determining the month from the **InvoiceDate** column by replacing what's inside the formula bar with the following text. When you are done typing the formula, press the **Enter** key.

```
Month = FORMAT(Invoices[InvoiceDate], "MMM yyyy")
```

   d) At this point, you should be able to see the **Month** column in the **Fields** list. Note that the **Month** column is displayed with a special function icon (i.e. **Fx**) to indicate that it is a calculated column.




Now that you have created the **Month** column, you will be able to use it to display data on reports that is broken out by month. However, this can cause a sorting issue because month names will be sorted alphabetically by default. For example, February will be sorted before January which will cause confusion and produce strange results. In the next step you will create an extra column that will be used assist in sorting the Month column in a chronologic fashion which is what people expect.

7. Create an additional calculated column named **MonthSort** to supply the proper sorting behavior for the **Month** column.

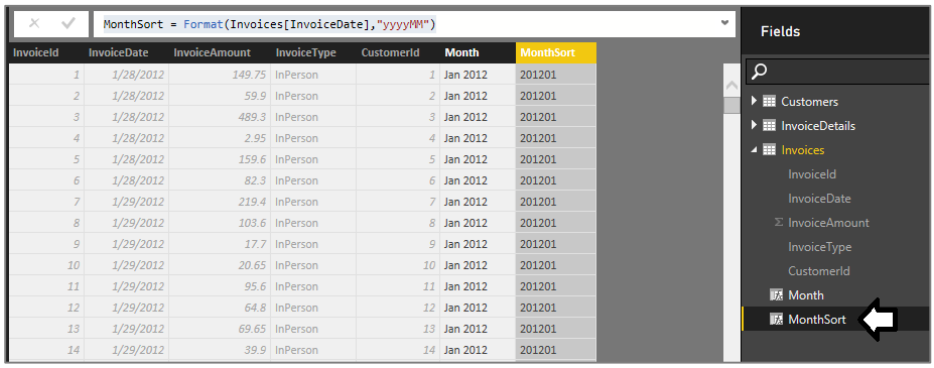
© Critical Path Training. 2015. All Rights Reserved

a) Click the **New Column** button in the ribbon to create a new column.

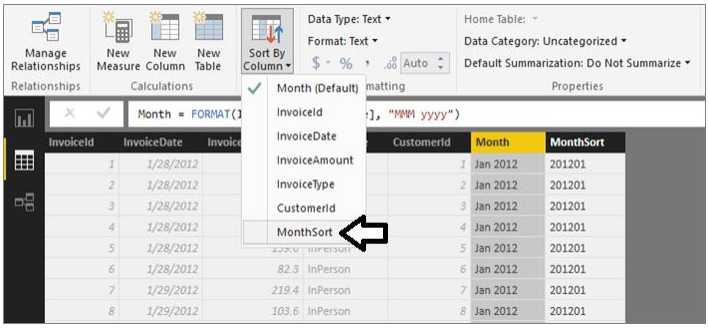
b) Give the new column a name of **MonthSort** and use the following DAX formula.

```
MonthSort = Format(Invoices[InvoiceDate],"yyyyMM")
```

c) Once you add the **MonthSort** column, you should be able to see it on the **Fields** list. You should also be able to see that the data in the **MonthSort** column contains a sortable text value which concatenates a 4-digit year value together with the 2-digit month value.




d) Now it is time to configure the **Month** column to use **MonthSort** as its sort column. Begin by clicking on the column header of the **Month** column to select it. Next, drop down the **Sort By Column** button and select the **MonthSort** column.




8. Create a calculated column named **Purchase Type** to provide a more human readable value for the **InvoiceType** column.

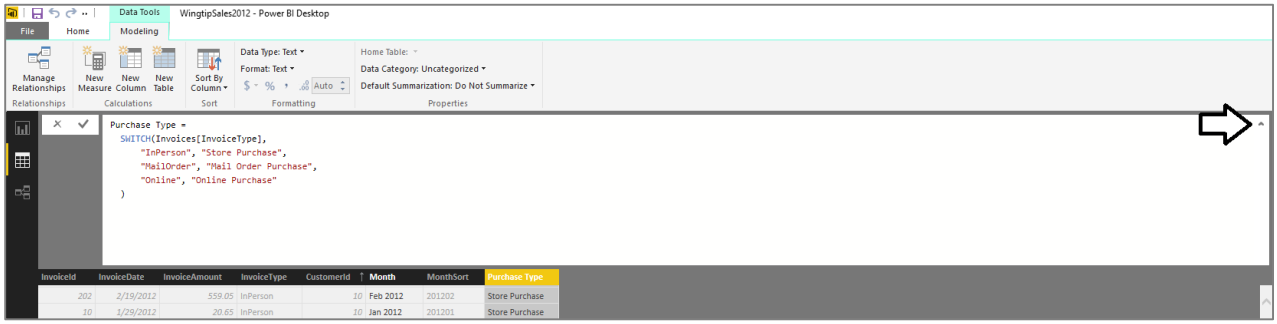
a) Click the **New Column** button in the ribbon to create a new column.

b) Given the new column a name of **Purchase Type** using the following formula.

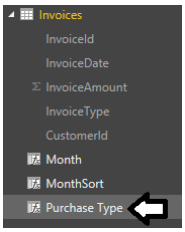
```
Purchase Type =
  SWITCH(Invoices[InvoiceType],
   "InPerson", "Store Purchase",
   "MailOrder", "Mail Order Purchase",
   "Online", "Online Purchase"
  )
```

c) Note that you can expand the height of the formula bar by clicking the downward facing button on the right-hand side. This will make it much easier to write multi-line DAX expressions which are far more readable and also more maintainable. Also remember that you can add a line break into a DAX formula by holding down the **SHIFT** key and then pressing the **ENTER** key. Once you have typed in the DAX formula for the **Purchase Type** field, press the **Enter** key.

© Critical Path Training. 2015. All Rights Reserved



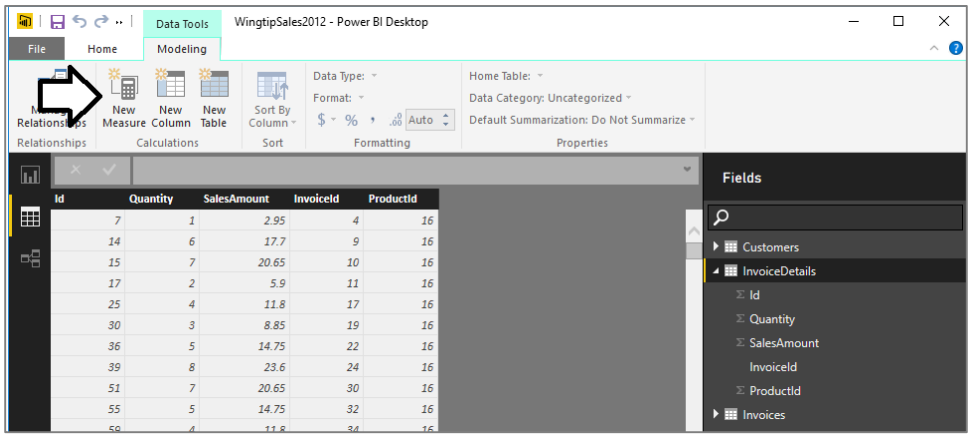

d) At this point, you should be able to see the **Purchase Type** column in the **Fields** list.




e) Inspect the values in the **Purchase Type** column to see the effect of the DAX formula you have written.

| InvoiceId | InvoiceDate | InvoiceAmount | InvoiceType | CustomerId | Month | MonthSort | Purchase Type |
|---|---|---|---|---|---|---|---|
| 5220 | 12/31/2012 | 25.85 | InPerson | 2653 | Dec 2012 | 201212 | Store Purchase |
| 5219 | 12/31/2012 | 319.3 | InPerson | 2652 | Dec 2012 | 201212 | Store Purchase |
| 5218 | 12/31/2012 | 162.6 | InPerson | 2651 | Dec 2012 | 201212 | Store Purchase |
| 5217 | 12/31/2012 | 183.6 | InPerson | 2650 | Dec 2012 | 201212 | Store Purchase |
| 5216 | 12/31/2012 | 114.6 | InPerson | 2649 | Dec 2012 | 201212 | Store Purchase |
| 5215 | 12/31/2012 | 59.8 | InPerson | 2648 | Dec 2012 | 201212 | Store Purchase |
| 5214 | 12/31/2012 | 41.8 | InPerson | 2647 | Dec 2012 | 201212 | Store Purchase |

9. Add a measure to the **InvoiceDetails** table to calculate total sales revenue.
   a) Select the **InvoiceDetails** table by clicking on it in the **Fields** list.
   b) Click the **New Measure** button in the ribbon to create a new measure.

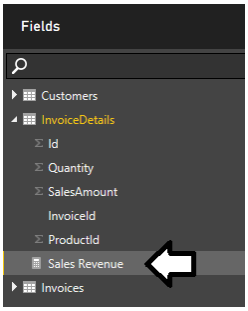



c) Create a new measure named **Sales Revenue** using the following DAX formula. Once you have typed the formula into the formula bar, press the **Enter** key to create the new measure.

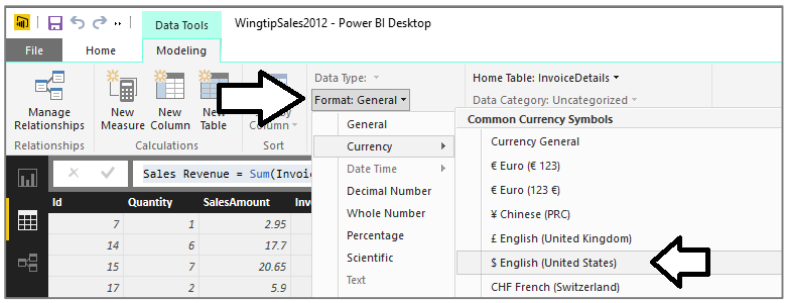
```
Sales Revenue = Sum(InvoiceDetails[SalesAmount])
```

© Critical Path Training. 2015. All Rights Reserved

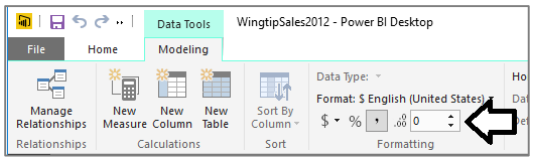
d) At this point you should be able to see the **Sales Revenue** measure in the **InvoiceDetails** table in the **Fields** list. Note that the **Sales Revenue** field is displayed with a special calculator icon to indicate that it is a measure.




e) Make sure the **Sales Revenue** measure is still selected in the **Fields** list. Modify the formatting of this measure by dropping down the **Format** menu on the ribbon and selecting a format type of **Currency > English (United States)**.




f) Enhance the formatting of the **Sales Revenue** column by reducing the number of decimal places shown from **2** to **0**.



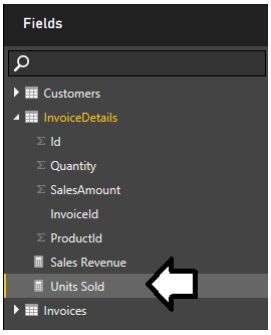

10. Create a second measure in the **InvoiceDetails** table to calculate the total number of units sold.

a) Click the **New Measure** button in the ribbon to create a new measure.

b) Create a new measure named **Units Sold** using the following DAX formula. Once you have type the formula into the formula bar, press the **Enter** key to create the new measure.

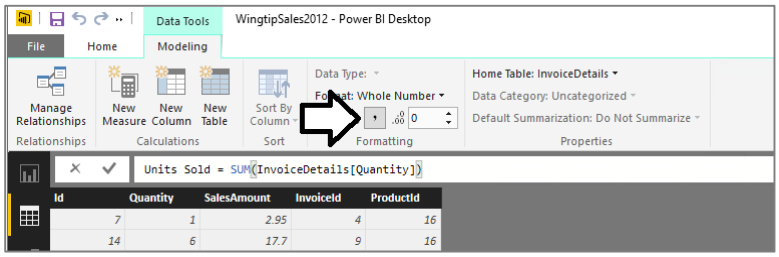
```
Units Sold = SUM(InvoiceDetails[Quantity])
```

c) You should be able to see the new **Units Sold** measure in the **Fields** list.




© Critical Path Training. 2015. All Rights Reserved

d) Make sure the **Units Sold** measure is selected in the **Fields** list and modify the formatting by clicking and selecting the button on the ribbon with the comma which adds a thousands separator so larger numbers are formatted with commas (e.g. 2,500).

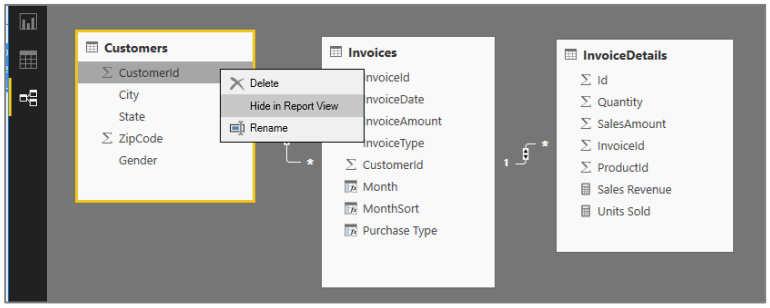



You are now done adding calculated columns and measures. You will now complete you modeling work by hiding fields that do not need to be seen when designing reports.

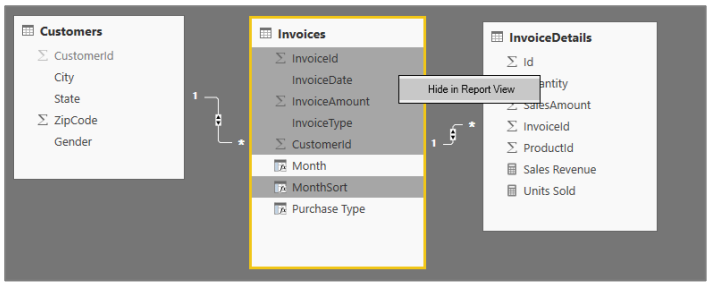
11. Hide all the fields that are not needed when designing reports.

a) Navigate to relationship view.

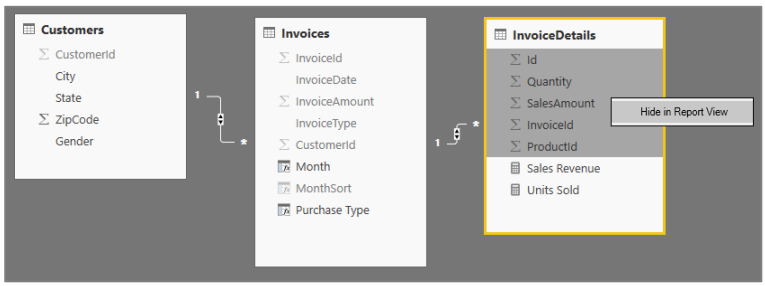
b) Right-click on the **CustomerId** field in the **Customers** table and click the **Hide in Report View** command.




c) Hold down the **Ctrl** key and select all the fields in the **Invoices** table except for the **Month** column and the **Purchase Type** column. Once you have the correct fields selected, right-click the selection and then click the **Hide in Report View** command.

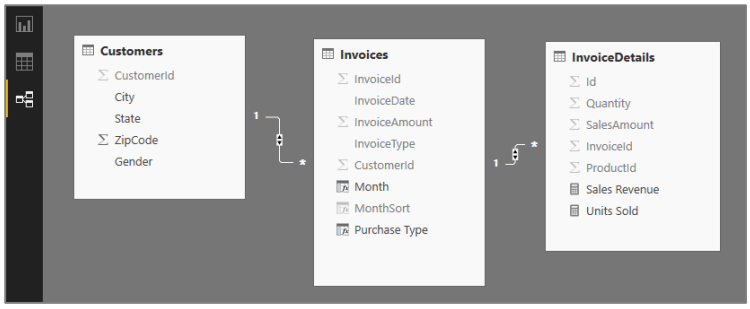



d) Hold down the **Ctrl** key and select all the fields in the **InvoiceDetails** table except for the **Sales Revenue** measure and the **Units Sold** measure. Once you have the correct fields selected, right-click the selection and then click the **Hide in Report View** command.




© Critical Path Training. 2015. All Rights Reserved

e) The tables in relationship view should appear as they do in the following screenshot. You are now done modeling the data.

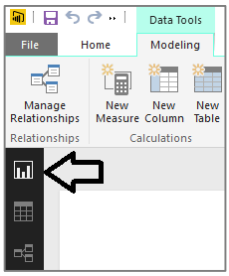



12. Save your work by clicking the **Save** button on the top left of the Power BI Desktop window.

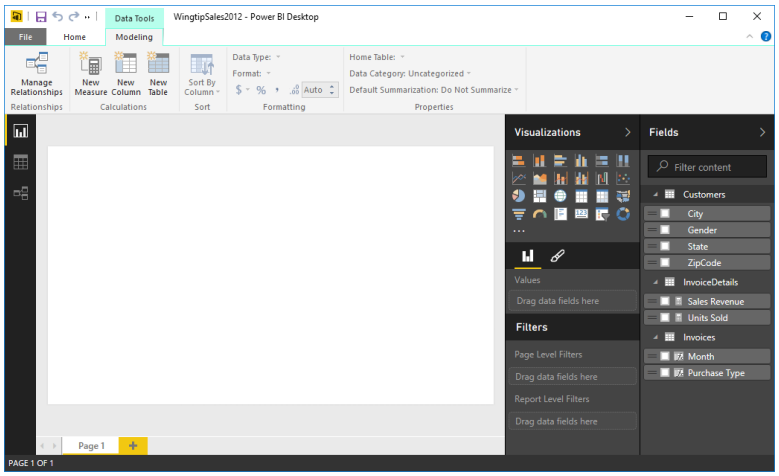
## Exercise 4: Create a Report using Power BI Desktop

In this exercise you will design a report with two pages of visualizations using the data model you created in the previous exercise.

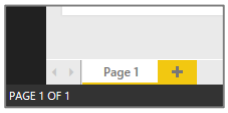
1. Make sure you still have the **Wingtipsales2012.pbix** file open that you created in the previous exercise.
2. Click on the top bottom in the sidebar to place Power BI Desktop into report view.




3. At this point, you should see the default report which contain an empty page named **Page 1**.

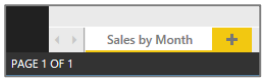



4. Change the title of the report's one and only page.
   a) Locate page title with a default value of **Page 1** on the bottom left corner of the report and double-click on it to edit its value.




© Critical Path Training. 2015. All Rights Reserved

b) Modify the page title to **Sales by Month** and press the **Enter** key to enter your changes

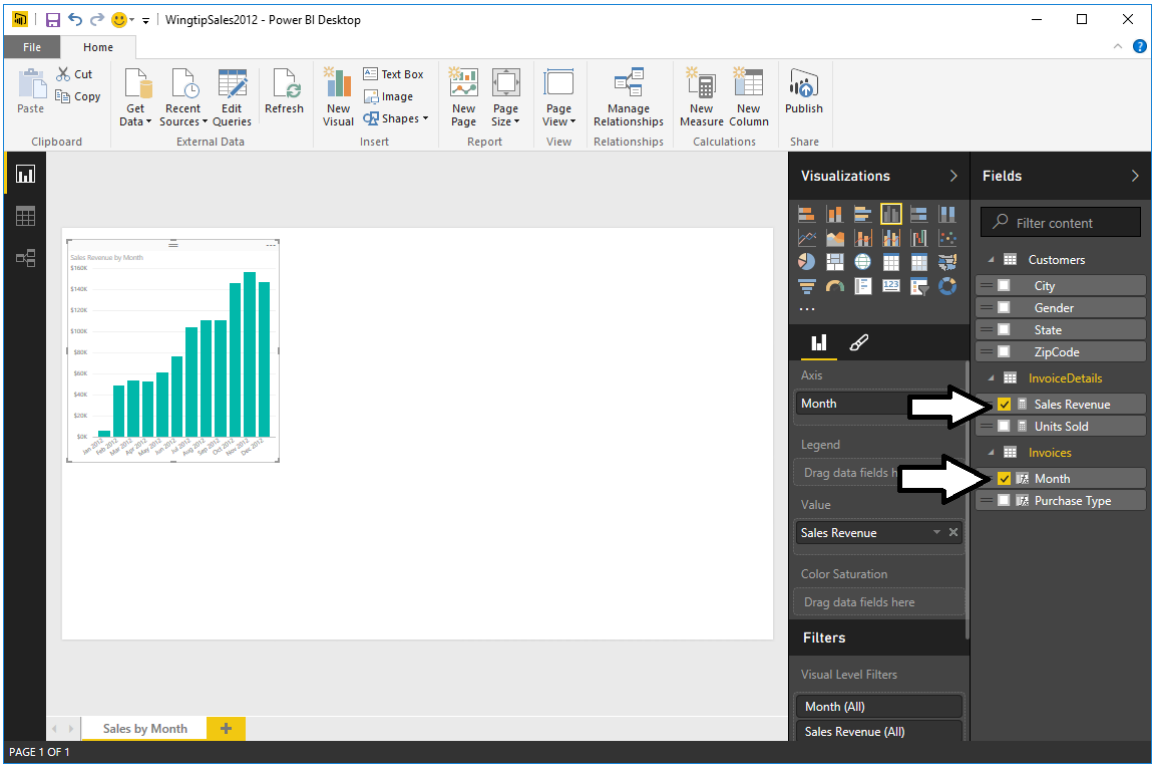



5. Add a new visualization to display sales revenue by month.

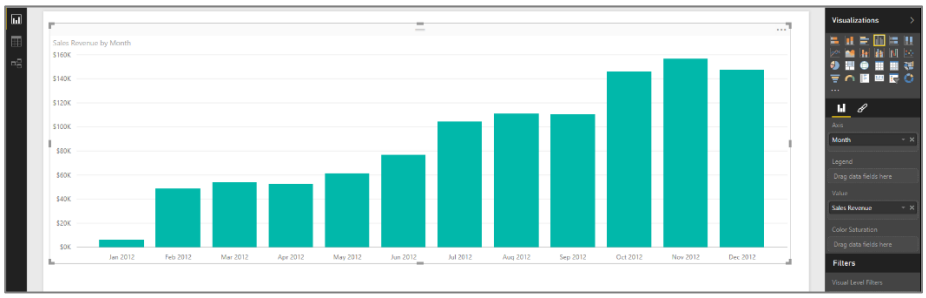
a) At this point the **Sales by Month** page should be empty without any visualizations.

b) In the **Fields** list, click the checkbox to select the **Sales Revenue** measure. Note that this action will add a new visualization to the page based on a clustered bar chart.

c) Next, click the checkbox to select the **Month** column which will automatically add that column into the **Axis** well and produce a bar chart like the one shown in the following screenshot.

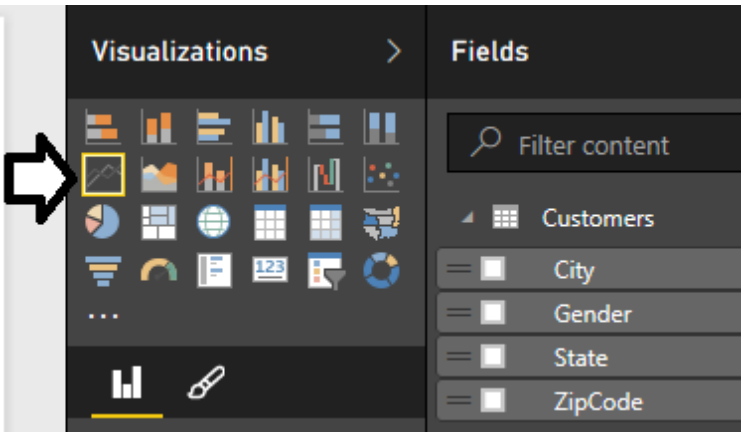



d) Use the mouse to grab the lower right corner of the visualization and resize it so it takes up the full width of the page.

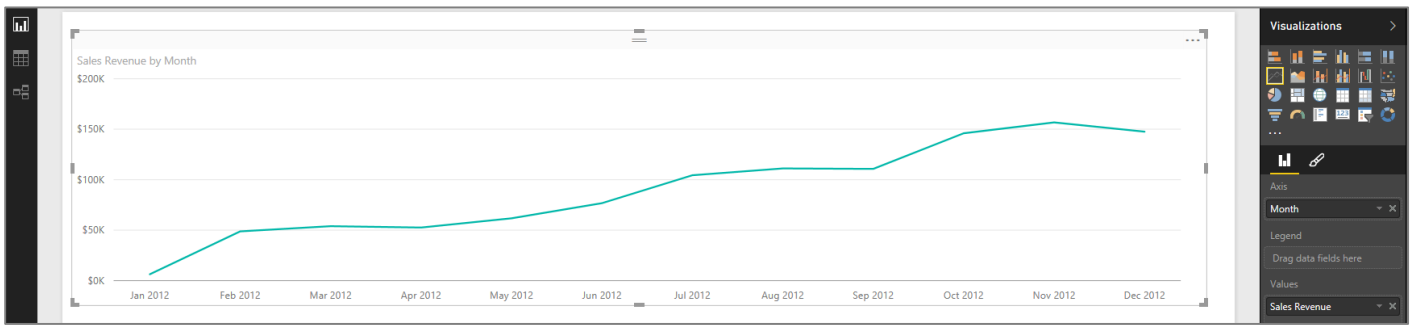


e) Ensure that the visualization is still selected. Click the **Line Chart** button in the **Visualizations** list to change the visualization type to display it as a line chart.

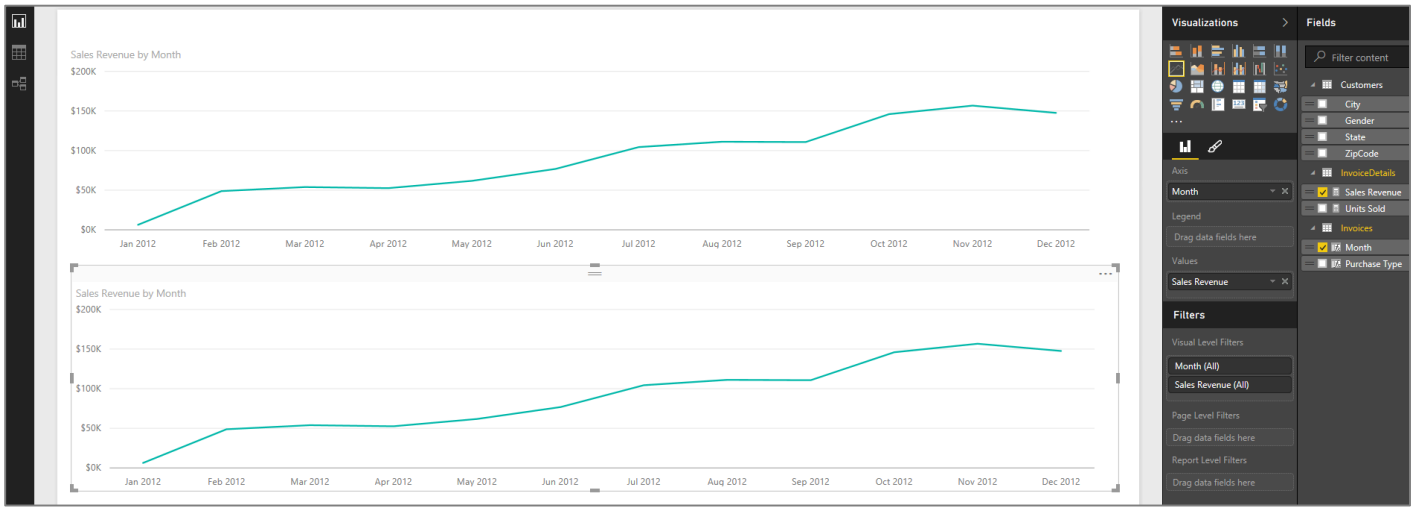
© Critical Path Training. 2015. All Rights Reserved




f) The visualization should now appear as a line chart like the one shown in the following screenshot.

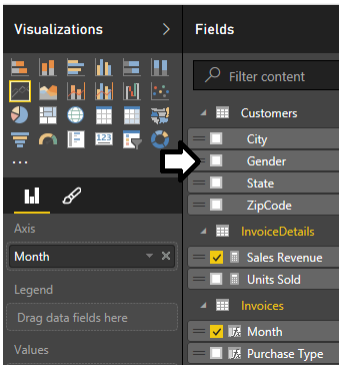



6. Create a second visualization to display sales revenue by month and gender.
   a) Select the visualization you created in the previous step and copy it to the Windows clpipboard.
   b) Use the Paste command to clone a copy of the first visualization.
   c) Use the mouse to move the second visualization so it appears under the first visualization.

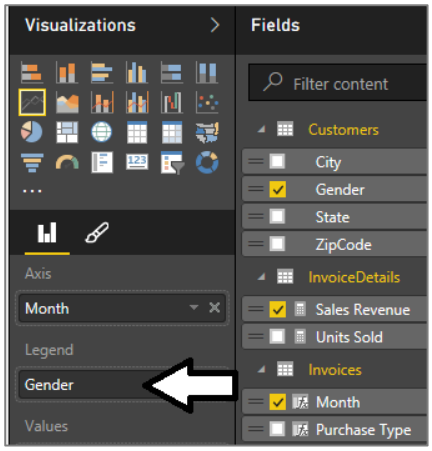



   d) Make sure the bottom visualization is selected.
   e) Locate the **Gender** field in the **Fields** list.

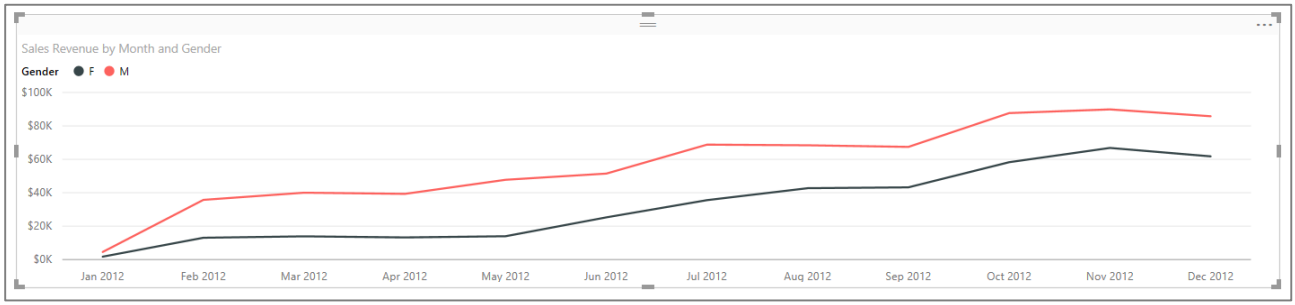
© Critical Path Training. 2015. All Rights Reserved



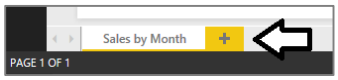

f) Using the mouse, drag the **Gender** field from the **Fields** list and drop it into the **Axis** well in the **Visualizations** pane.



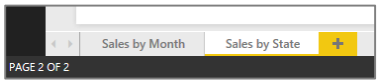

g) The bottom visualization should now appear as the one shown in the following screenshot.




7. Save your work by clicking the **Save** button on the top left of the Power BI Desktop window.
8. Add a second page to the report.
   a) Click the (**+**) button in the bottom left corner of report view to add a second page to the report.

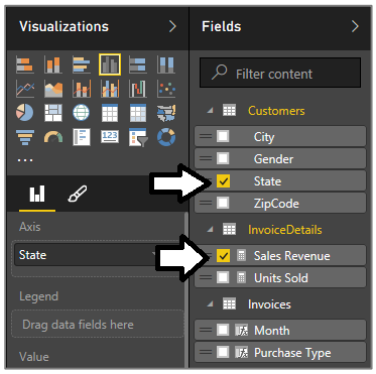



   b) Once the page has been created, modify its title to **Sales by States**.

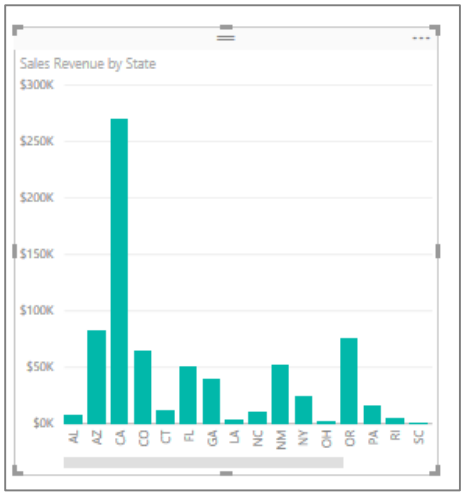



© Critical Path Training. 2015. All Rights Reserved

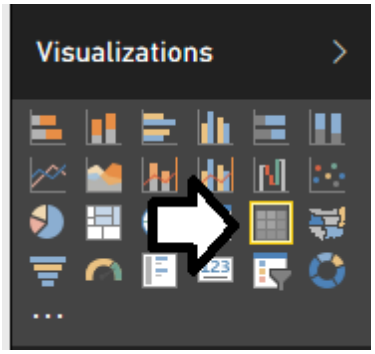
9. Add a new visualization to the **Sales by States** page to display numeric data in a matrix.
   a) In the **Fields** list, select the **Sales Revenue** measure to create a new clustered bar chart visualization.
   b) Next, select the **State** column.




   c) At this point, your visualization should appear like the one shown in the following screenshot.




   d) Change the visualization type by clicking the **Matrix** button in the **Visualizations** list.




   e) The visualization should now display numeric data in a tabular format as shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

| State | Sales Revenue |
|---|---|
| AL | $8,538 |
| AZ | $83,218 |
| CA | $270,926 |
| CO | $64,683 |
| CT | $11,513 |
| FL | $51,731 |
| GA | $40,306 |
| LA | $4,468 |
| NC | $11,018 |
| NM | $52,067 |
| NY | $24,207 |
| OH | $2,491 |
| OR | $76,153 |
| PA | $16,390 |
| RI | $4,990 |
| SC | $1,314 |
| TN | $27,443 |
| TX | $212,085 |
| UT | $20,101 |

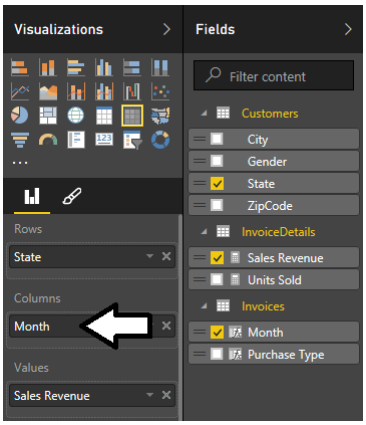
f) Click on the column header of the **Sales Revenue** column to resort the rows within the visualization so that the states with the largest amounts of sales revenue are sorted towards the top.

| State | Sales Revenue |
|---|---|
| CA | $270,926 |
| TX | $212,085 |
| AZ | $83,218 |
| OR | $76,153 |
| WA | $74,615 |
| CO | $64,683 |
| NM | $52,067 |
| FL | $51,731 |
| GA | $40,306 |
| TN | $27,443 |
| NY | $24,207 |
| UT | $20,101 |
| VA | $18,568 |
| PA | $16,390 |
| CT | $11,513 |
| NC | $11,018 |
| AL | $8,538 |
| RI | $4,990 |
| LA | $4,468 |
| OH | $2,491 |
| SC | $1,314 |
| **Total** | **$1,076,826** |

g) In the **Fields** list, locate the **Month** column.

h) Using the mouse, drag the **Month** field from the **Fields** list and drop it into the **Columns** well in the **Visualizations** pane.

© Critical Path Training. 2015. All Rights Reserved




i) You should now see the visualization expand to the right to display a column for each month.

j) Using the mouse, resize the visualization to display all of its contents.

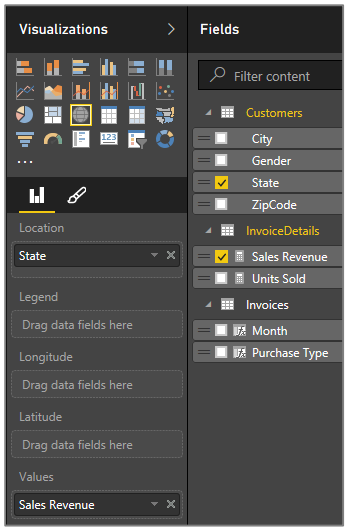
| State | Jan 2012 | Feb 2012 | Mar 2012 | Apr 2012 | May 2012 | Jun 2012 | Jul 2012 | Aug 2012 | Sep 2012 | Oct 2012 | Nov 2012 | Dec 2012 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CA | $904 | $15,730 | $13,982 | $16,916 | $21,589 | $22,760 | $26,107 | $29,885 | $25,792 | $33,034 | $30,771 | $33,456 | **$270,926** |
| TX | $2,461 | $10,623 | $12,719 | $10,644 | $13,772 | $14,397 | $18,225 | $19,249 | $23,331 | $28,530 | $31,221 | $26,913 | **$212,085** |
| AZ | $1,575 | $4,280 | $5,528 | $5,186 | $3,912 | $6,259 | $7,685 | $7,163 | $9,506 | $9,678 | $12,221 | $10,227 | **$83,218** |
| OR | $68 | $4,551 | $6,046 | $5,897 | $5,708 | $6,651 | $6,321 | $6,766 | $5,885 | $7,786 | $10,309 | $10,166 | **$76,153** |
| WA | $394 | $4,165 | $7,696 | $7,151 | $6,535 | $7,897 | $6,430 | $7,003 | $5,891 | $9,523 | $5,180 | $6,750 | **$74,615** |
| CO | $679 | $4,144 | $3,164 | $3,426 | $5,527 | $2,635 | $4,671 | $7,757 | $7,803 | $7,409 | $9,050 | $8,419 | **$64,683** |
| NM | $225 | $1,598 | $3,437 | $1,918 | $3,389 | $5,285 | $7,840 | $3,404 | $5,995 | $9,391 | $4,921 | $4,664 | **$52,067** |
| FL | | | | | | $1,551 | $7,680 | $4,994 | $7,586 | $7,739 | $12,236 | $9,944 | **$51,731** |
| GA | | | | | | $1,579 | $6,209 | $5,952 | $3,540 | $7,307 | $9,335 | $6,384 | **$40,306** |
| TN | | | | | | $784 | $1,937 | $6,201 | $2,003 | $3,973 | $6,178 | $6,368 | **$27,443** |
| NY | | | | | | $1,508 | $1,290 | $3,519 | $4,061 | $2,736 | $5,490 | $5,603 | **$24,207** |
| UT | | $3,224 | $1,387 | $1,155 | $1,324 | $1,272 | $698 | $1,457 | $1,636 | $2,671 | $3,688 | $1,590 | **$20,101** |
| VA | | $499 | | $309 | | $1,157 | $1,841 | $1,666 | $1,841 | $4,197 | $4,554 | $2,504 | **$18,568** |
| PA | | | | | | $795 | $3,690 | $2,264 | $466 | $2,973 | $3,733 | $2,469 | **$16,390** |
| CT | | | | | | $1,047 | $1,282 | $1,251 | $565 | $1,399 | $3,037 | $2,932 | **$11,513** |
| NC | | | | | | $466 | $1,094 | $252 | $1,782 | $3,310 | $1,085 | $3,030 | **$11,018** |
| AL | | | | | | $316 | | $465 | $949 | $1,329 | $2,015 | $3,464 | **$8,538** |
| RI | | | | | | | $960 | $917 | $645 | $1,287 | $172 | $1,010 | **$4,990** |
| LA | | | | | | $399 | | $691 | $445 | $553 | $1,555 | $824 | **$4,468** |
| OH | | | | | | | $448 | $78 | $994 | $711 | | $261 | **$2,491** |
| SC | | | | | | | | $236 | | $464 | | $614 | **$1,314** |
| **Total** | **$6,306** | **$48,815** | **$53,958** | **$52,601** | **$61,756** | **$76,756** | **$104,408** | **$111,167** | **$110,716** | **$145,999** | **$156,751** | **$147,593** | **$1,076,826** |

10. Add a second visualization to display a map visualization which shows how sales revenue is distributed across states.

a) Click on the white space below the matrix visualization to make sure no visualization is selected.

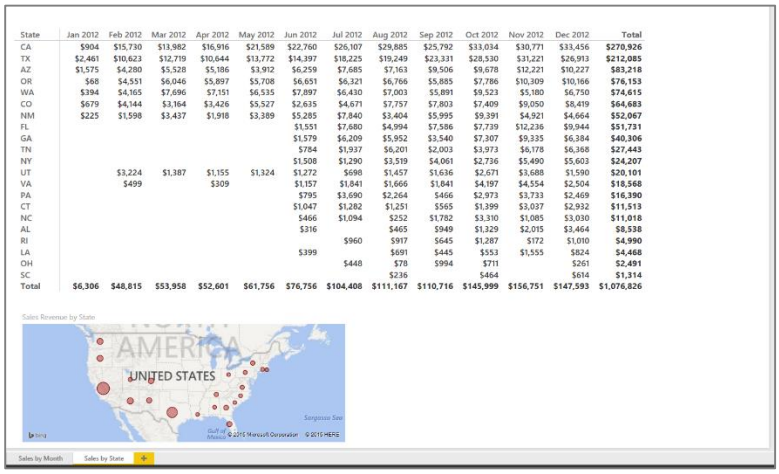
b) Select the **State** column in the **Fields** list to create a new visualization. Note that Power BI Desktop recognizes the **State** column as a geographic field so it creates the new visualization using the **Map** visualization type.

c) Using the mouse, drag the **Sales Revenue** measure from the **Fields** list into the **Values** well in the **Visualization** pane.




d) The Map visualization should now show dots of different sizes to visually depict how sales revenue is distributed across states.

e) Using the mouse, rearrange the visualizations on the page to match the following screenshot.



| State | Jan 2012 | Feb 2012 | Mar 2012 | Apr 2012 | May 2012 | Jun 2012 | Jul 2012 | Aug 2012 | Sep 2012 | Oct 2012 | Nov 2012 | Dec 2012 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CA | $904 | $15,730 | $13,982 | $16,916 | $21,589 | $22,760 | $26,107 | $29,885 | $25,792 | $33,034 | $30,771 | $33,456 | **$270,926** |
| TX | $2,461 | $10,623 | $12,719 | $10,644 | $13,772 | $14,397 | $18,225 | $19,249 | $23,331 | $28,530 | $31,221 | $26,913 | **$212,085** |
| AZ | $1,575 | $4,280 | $5,528 | $5,186 | $3,912 | $6,259 | $7,685 | $7,163 | $9,506 | $9,678 | $12,221 | $10,227 | **$83,218** |
| OR | $68 | $4,551 | $6,046 | $5,897 | $5,708 | $6,651 | $6,321 | $6,766 | $5,885 | $7,786 | $10,309 | $10,166 | **$76,153** |
| WA | $394 | $4,165 | $7,696 | $7,151 | $6,535 | $7,897 | $6,430 | $7,003 | $5,891 | $9,523 | $5,180 | $6,750 | **$74,615** |
| CO | $679 | $4,144 | $3,164 | $3,426 | $5,527 | $2,635 | $4,671 | $7,757 | $7,803 | $7,409 | $9,050 | $8,419 | **$64,683** |
| NM | $225 | $1,598 | $3,437 | $1,918 | $3,389 | $5,285 | $7,840 | $3,404 | $5,995 | $9,391 | $4,921 | $4,664 | **$52,067** |
| FL | | | | | | $1,551 | $7,680 | $4,994 | $7,586 | $7,739 | $12,236 | $9,944 | **$51,731** |
| GA | | | | | | $1,579 | $6,209 | $5,952 | $3,540 | $7,307 | $9,335 | $6,384 | **$40,306** |
| TN | | | | | | $784 | $1,937 | $6,201 | $2,003 | $3,973 | $6,178 | $6,368 | **$27,443** |
| NY | | | | | | $1,508 | $1,290 | $3,519 | $4,061 | $2,736 | $5,490 | $5,603 | **$24,207** |
| UT | | $3,224 | $1,387 | $1,155 | $1,324 | $1,272 | $698 | $1,457 | $1,636 | $2,671 | $3,688 | $1,590 | **$20,101** |
| VA | | $499 | | $309 | | $1,157 | $1,841 | $1,666 | $1,841 | $4,197 | $4,554 | $2,504 | **$18,568** |
| PA | | | | | | $795 | $3,690 | $2,264 | $466 | $2,973 | $3,733 | $2,469 | **$16,390** |
| CT | | | | | | $1,047 | $1,282 | $1,251 | $565 | $1,399 | $3,037 | $2,932 | **$11,513** |
| NC | | | | | | $466 | $1,094 | $252 | $1,782 | $3,310 | $1,085 | $3,030 | **$11,018** |
| AL | | | | | | $316 | | $465 | $949 | $1,329 | $2,015 | $3,464 | **$8,538** |
| RI | | | | | | | $960 | $917 | $645 | $1,287 | $172 | $1,010 | **$4,990** |
| LA | | | | | | $399 | | $691 | $445 | $553 | $1,555 | $824 | **$4,468** |
| OH | | | | | | | $448 | $78 | $994 | $711 | | $261 | **$2,491** |
| SC | | | | | | | | $236 | | $464 | | $614 | **$1,314** |
| **Total** | **$6,306** | **$48,815** | **$53,958** | **$52,601** | **$61,756** | **$76,756** | **$104,408** | **$111,167** | **$110,716** | **$145,999** | **$156,751** | **$147,593** | **$1,076,826** |



11. Add a third visualization to the page to display sales revenue by state and purchase type.

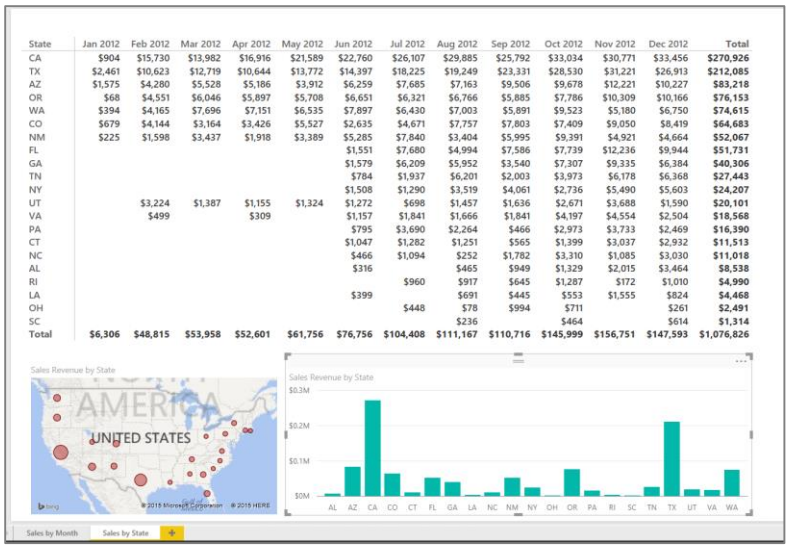
© Critical Path Training. 2015. All Rights Reserved

a) Click on the white space in the bottom right corner of the page to make sure no visualization is selected.

b) Select the **Sales Revenue** measure in the **Fields** list to create a new bar chart visualization.

c) Select the **State** column in the **Fields** list to add a legend to the bar chart.

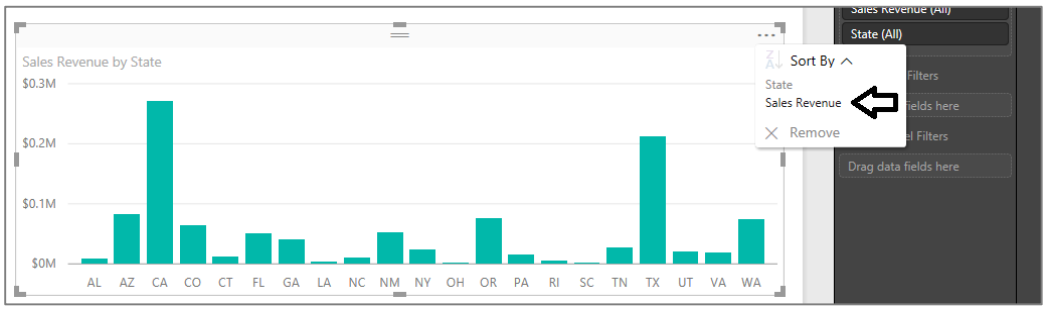
d) Using the mouse, rearrange the visualizations on the page to match the following screenshot.



| State | Jan 2012 | Feb 2012 | Mar 2012 | Apr 2012 | May 2012 | Jun 2012 | Jul 2012 | Aug 2012 | Sep 2012 | Oct 2012 | Nov 2012 | Dec 2012 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CA | $904 | $15,730 | $13,982 | $16,916 | $21,589 | $22,760 | $26,107 | $29,885 | $25,792 | $33,034 | $30,771 | $33,456 | $270,926 |
| TX | $2,461 | $10,623 | $12,719 | $10,644 | $13,772 | $14,397 | $18,225 | $19,249 | $23,331 | $28,530 | $31,221 | $26,913 | $212,085 |
| AZ | $1,575 | $4,280 | $5,528 | $5,186 | $3,912 | $6,259 | $7,685 | $7,163 | $9,506 | $9,678 | $12,221 | $10,227 | $83,218 |
| OR | $68 | $4,551 | $6,046 | $5,897 | $5,708 | $6,651 | $6,321 | $6,766 | $5,885 | $7,786 | $10,309 | $10,166 | $76,153 |
| WA | $394 | $4,165 | $7,696 | $7,151 | $6,535 | $7,897 | $6,430 | $7,003 | $5,891 | $9,523 | $5,180 | $6,750 | $74,615 |
| CO | $679 | $4,144 | $3,164 | $3,426 | $5,527 | $2,635 | $4,671 | $7,757 | $7,803 | $7,409 | $9,050 | $8,419 | $64,683 |
| NM | $225 | $1,598 | $3,437 | $1,918 | $3,389 | $5,285 | $7,840 | $3,404 | $5,995 | $9,391 | $4,921 | $4,664 | $52,067 |
| FL | | | | | | $1,551 | $7,680 | $4,994 | $7,586 | $7,739 | $12,236 | $9,944 | $51,731 |
| GA | | | | | | $1,579 | $6,209 | $5,952 | $3,540 | $7,307 | $9,335 | $6,384 | $40,306 |
| TN | | | | | | $784 | $1,937 | $6,201 | $2,003 | $3,973 | $6,178 | $6,368 | $27,443 |
| NY | | | | | | $1,508 | $1,290 | $3,519 | $4,061 | $2,736 | $5,490 | $5,603 | $24,207 |
| UT | | $3,224 | $1,387 | $1,155 | $1,324 | $1,272 | $698 | $1,457 | $1,636 | $2,671 | $3,688 | $1,590 | $20,101 |
| VA | | $499 | | $309 | | $1,157 | $1,841 | $1,666 | $1,841 | $4,197 | $4,554 | $2,504 | $18,568 |
| PA | | | | | | $795 | $3,690 | $2,264 | $466 | $2,973 | $3,733 | $2,469 | $16,390 |
| CT | | | | | | $1,047 | $1,282 | $1,251 | $565 | $1,399 | $3,037 | $2,932 | $11,513 |
| NC | | | | | | $466 | $1,094 | $252 | $1,782 | $3,310 | $1,085 | $3,030 | $11,018 |
| AL | | | | | | $316 | | $465 | $949 | $1,329 | $2,015 | $3,464 | $8,538 |
| RI | | | | | | | $960 | $917 | $645 | $1,287 | $172 | $1,010 | $4,990 |
| LA | | | | | | $399 | | $691 | $445 | $553 | $1,555 | $824 | $4,468 |
| OH | | | | | | | $448 | $78 | $994 | $711 | | $261 | $2,491 |
| SC | | | | | | | | $236 | | $464 | | $614 | $1,314 |
| Total | $6,306 | $48,815 | $53,958 | $52,601 | $61,756 | $76,756 | $104,408 | $111,167 | $110,716 | $145,999 | $156,751 | $147,593 | $1,076,826 |

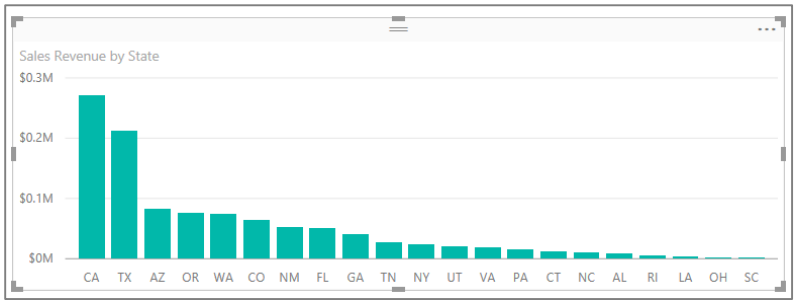


e) Click the (**…**) menu in the top left corner of the new visualization and click **Sort By > Sales Revenue**.





f) The visualization should now display states sorted by sales revenue.

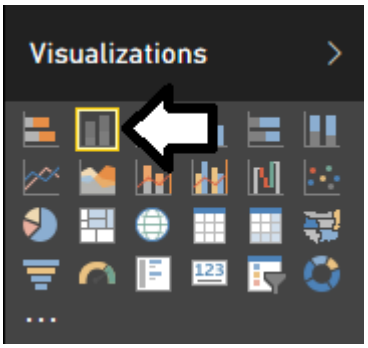




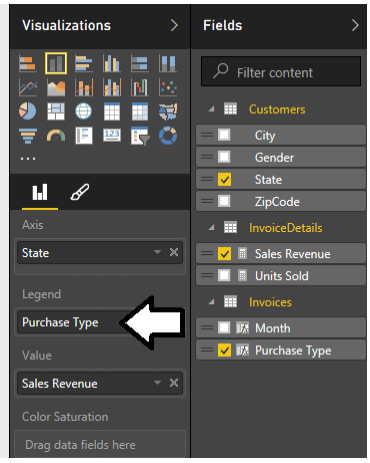
While the visualization appears to be a simple bar chart, it has actually been created as a clustered bar chart. You must modify the visualization type from a clustered bar chart to a standard bar chart to achieve the desired layout for this particular scenario.

g) Change the visualization type by clicking the **Bar chart** button in the **Visualizations** list. Note that you will not see the effects of this change until the next step.

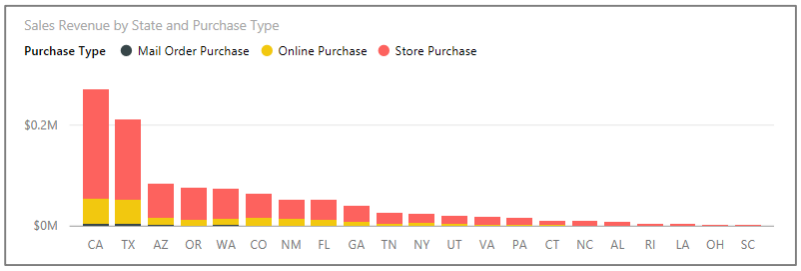
© Critical Path Training. 2015. All Rights Reserved




h) Using the mouse, drag the **Purchase Type** column from the **Fields** list to the **Legend** well in the **Visualization** pane.

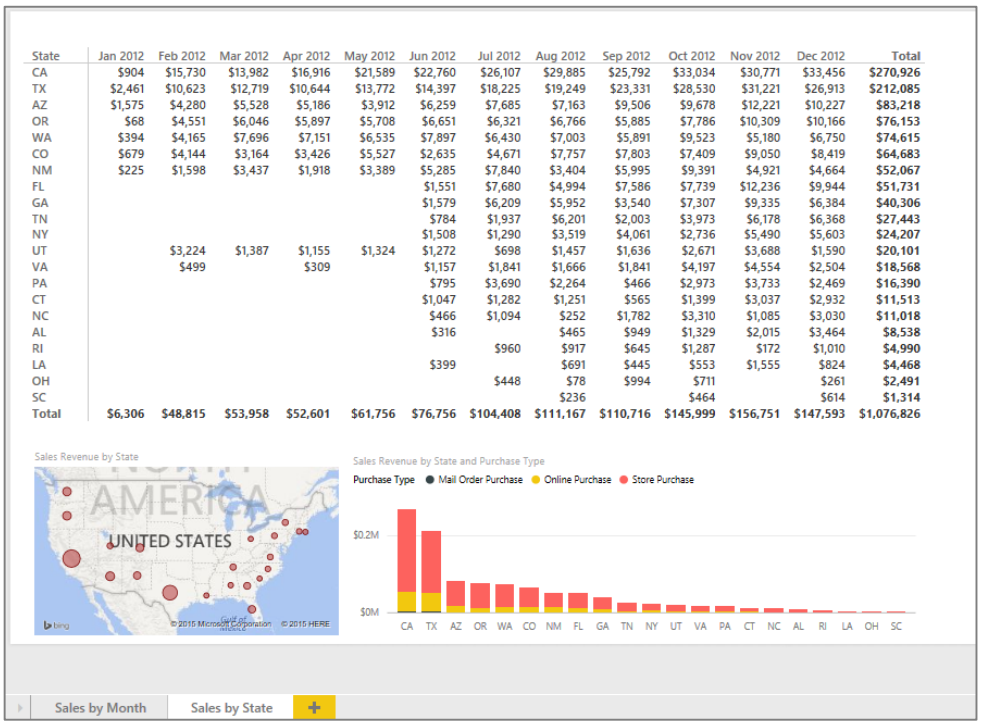



i) Your visualization should now show the per-state breakdown of sales revenue by purchase type.




12. At this point, you have now finished designing the **Sales by State** page.

© Critical Path Training. 2015. All Rights Reserved



| State | Jan 2012 | Feb 2012 | Mar 2012 | Apr 2012 | May 2012 | Jun 2012 | Jul 2012 | Aug 2012 | Sep 2012 | Oct 2012 | Nov 2012 | Dec 2012 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CA | $904 | $15,730 | $13,982 | $16,916 | $21,589 | $22,760 | $26,107 | $29,885 | $25,792 | $33,034 | $30,771 | $33,456 | $270,926 |
| TX | $2,461 | $10,623 | $12,719 | $10,644 | $13,772 | $14,397 | $18,225 | $19,249 | $23,331 | $28,530 | $31,221 | $26,913 | $212,085 |
| AZ | $1,575 | $4,280 | $5,528 | $5,186 | $3,912 | $6,259 | $7,685 | $7,163 | $9,506 | $9,678 | $12,221 | $10,227 | $83,218 |
| OR | $68 | $4,551 | $6,046 | $5,897 | $5,708 | $6,651 | $6,321 | $6,766 | $5,885 | $7,786 | $10,309 | $10,166 | $76,153 |
| WA | $394 | $4,165 | $7,696 | $7,151 | $6,535 | $7,897 | $6,430 | $7,003 | $5,891 | $9,523 | $5,180 | $6,750 | $74,615 |
| CO | $679 | $4,144 | $3,164 | $3,426 | $5,527 | $2,635 | $4,671 | $7,757 | $7,803 | $7,409 | $9,050 | $8,419 | $64,683 |
| NM | $225 | $1,598 | $3,437 | $1,918 | $3,389 | $5,285 | $7,840 | $3,404 | $5,995 | $9,391 | $4,921 | $4,664 | $52,067 |
| FL | | | | | | $1,551 | $7,680 | $4,994 | $7,586 | $7,739 | $12,236 | $9,944 | $51,731 |
| GA | | | | | | $1,579 | $6,209 | $5,952 | $3,540 | $7,307 | $9,335 | $6,384 | $40,306 |
| TN | | | | | | $784 | $1,937 | $6,201 | $2,003 | $3,973 | $6,178 | $6,368 | $27,443 |
| NY | | | | | | $1,508 | $1,290 | $3,519 | $4,061 | $2,736 | $5,490 | $5,603 | $24,207 |
| UT | | $3,224 | $1,387 | $1,155 | $1,324 | $1,272 | $698 | $1,457 | $1,636 | $2,671 | $3,688 | $1,590 | $20,101 |
| VA | | $499 | | $309 | | $1,157 | $1,841 | $1,666 | $1,841 | $4,197 | $4,554 | $2,504 | $18,568 |
| PA | | | | | | $795 | $3,690 | $2,264 | $466 | $2,973 | $3,733 | $2,469 | $16,390 |
| CT | | | | | | $1,047 | $1,282 | $1,251 | $565 | $1,399 | $3,037 | $2,932 | $11,513 |
| NC | | | | | | $466 | $1,094 | $252 | $1,782 | $3,310 | $1,085 | $3,030 | $11,018 |
| AL | | | | | | $316 | | $465 | $949 | $1,329 | $2,015 | $3,464 | $8,538 |
| RI | | | | | | | $960 | $917 | $645 | $1,287 | $172 | $1,010 | $4,990 |
| LA | | | | | | $399 | | $691 | $445 | $553 | $1,555 | $824 | $4,468 |
| OH | | | | | | | $448 | $78 | $994 | $711 | | $261 | $2,491 |
| SC | | | | | | | | $236 | | $464 | | $614 | $1,314 |
| Total | $6,306 | $48,815 | $53,958 | $52,601 | $61,756 | $76,756 | $104,408 | $111,167 | $110,716 | $145,999 | $156,751 | $147,593 | $1,076,826 |

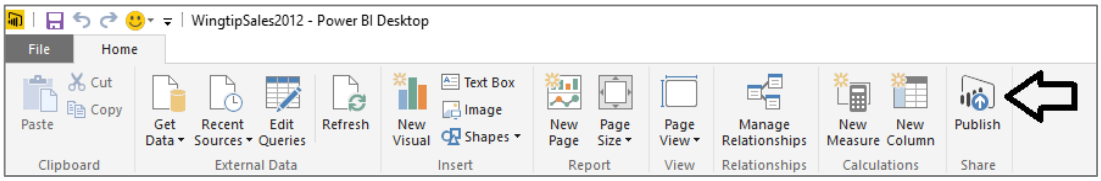


13. Save your work by clicking the Save button on the top left of the Power BI Desktop window.

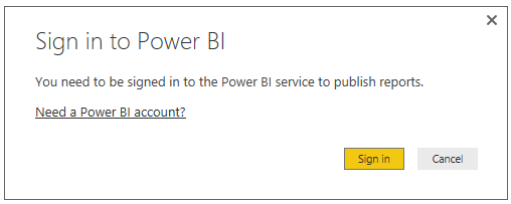
## Exercise 5: Publishing a PBIX File to the Power BI Service

In this exercise you will complete your work by publishing the PBIX file to your personal workspace in the Power BI service.

1. Make sure you still have the **Wingtipsales2012.pbix** file open that you created in the previous exercise.
2. Click the **Publish** button on the far right-hand side of the ribbon..

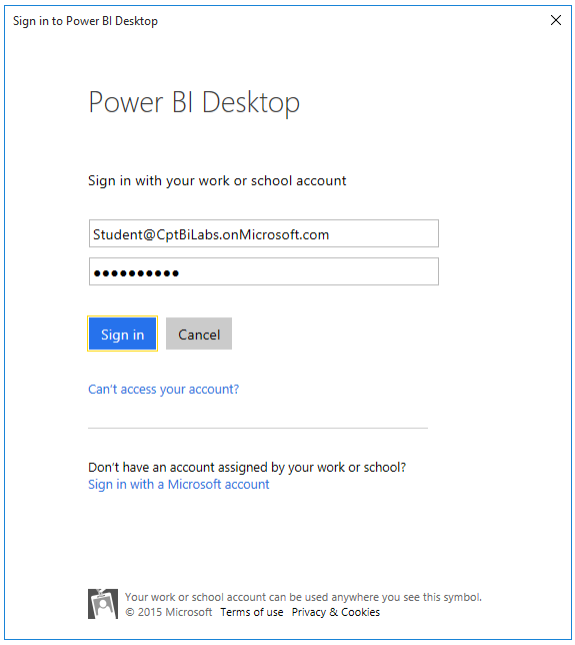



3. When promoted with the **Sign in to Power BI** dialog, click the **Sign In** button



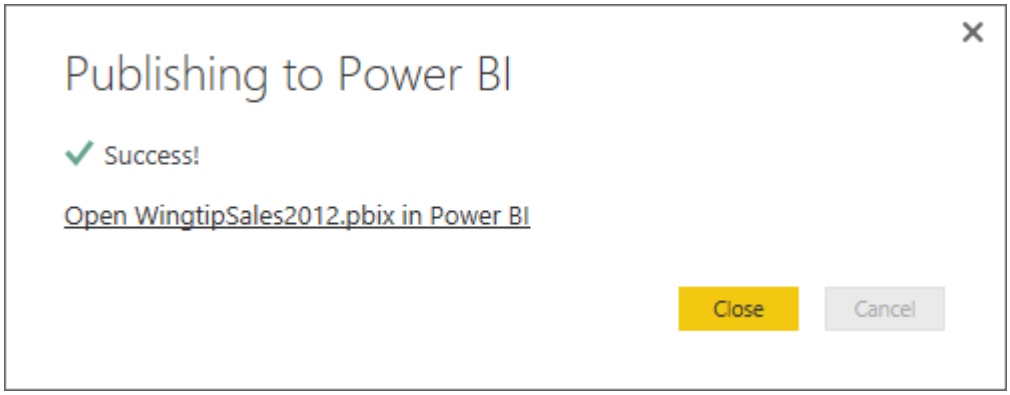

4. Sign into the Power BI service using your primary Office 365 account to give Power BI Desktop the access to publish the PBIX file.

© Critical Path Training. 2015. All Rights Reserved



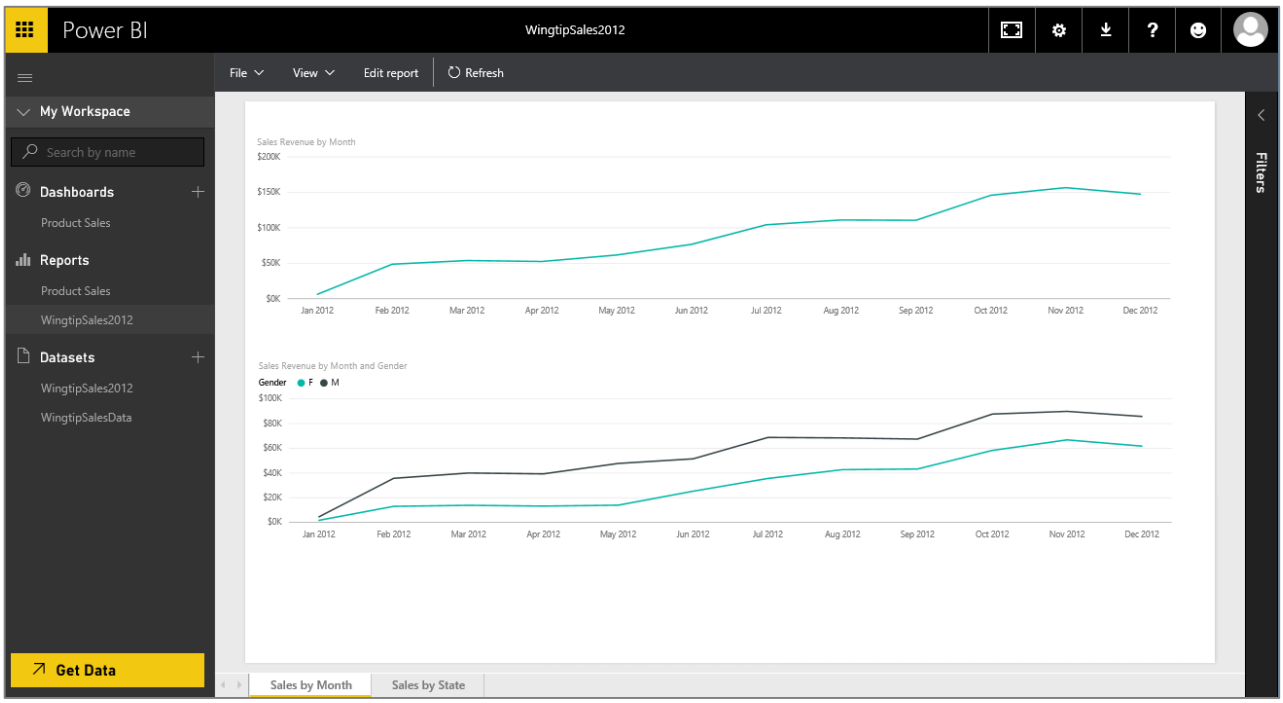

5. After you have signed in, Power BI Desktop will display the **Publishing to Power BI** dialog showing you that the publishing process is underway.

6. Once the publishing process has completed, the **Publishing to Power BI** dialog will display a success message and provide you with a link to **Open WingtipSales2012.pbix in Power BI**. Click on that link to navigate to the Power BI service using the browser.

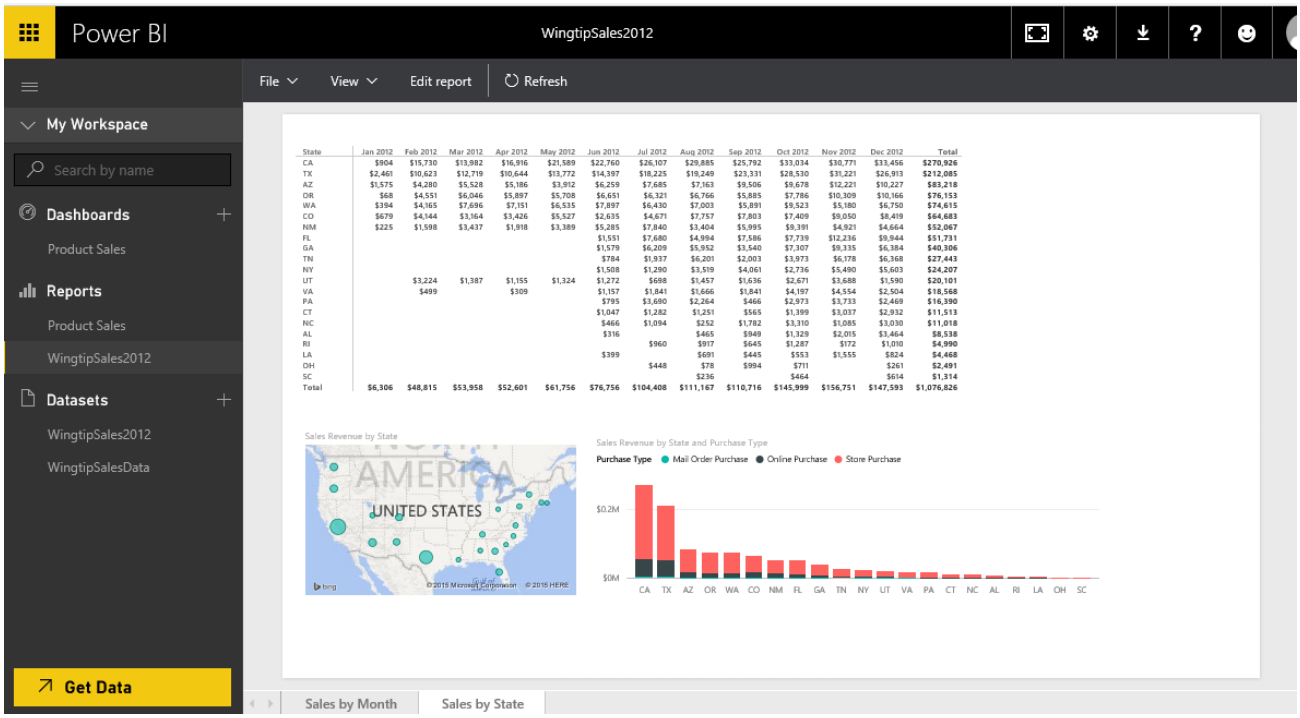



7. You should now be able to see the **Sales by Month** page of the report you just created.

© Critical Path Training. 2015. All Rights Reserved




8. Click on the **Sales by State** link at the bottom of the screen to see the second page of the report.




9. You have now successfully created and published a simple BI solution using Power BI Desktop.
10. If you finish early and have extra time, experiment by clicking the Edit report button and seeing how you can continue to modify the pages of the report using the browser.

© Critical Path Training. 2015. All Rights Reserved

# Module 03: Using the Power Query Features of Power BI Desktop

## Module Description

This module introduces the core concepts of creating and designing queries in Power BI Desktop using the features of Power Query. You will learn how to use Power Query to import data from a variety of sources including databases, Excel spreadsheets, web pages and SharePoint lists. You will also learn how Power Query can be used to reshape data during the import process by splitting a column with delimited values and combining multiple columns into a single column. You will also learn to perform powerful table operations such as merging columns and appending rows from multiple data sources as well as transposing, pivoting and un-pivoting the rows and columns in a source table.

## Module Agenda

- Data Sources and Data Discovery
- Power Query Fundamentals
- Working with the Query Editor Window
- Combining Queries
- Importing Data Into a Star Schema

## Topics Covered

Data Discovery
An Example OLTP Database
Deciding What To Measure
Defining Grain Statements
Power Query is an ETL Tool
Query Steps
Advanced Editor
Understanding Query Input and Output
Query Composition
Creating a New Query
Supported File-based Data Sources
Supported Databases
Azure Data Sources
Other Supported Data Sources
Selecting Tables from a Data Source
Query Editor Ribbon Tabs
Examples of Basic Power Query Steps
Replacing Values
Converting Column Types
Expanding Related Columns
Merging Columns
Splitting Columns
Adding a Custom Column
Combining Queries
Merging Columns
Appending Rows
Pivoting Columns
Unpivoting Columns
Data Modeling using a Star Schema
Building a Star Schema with Power Query

## Instructor Demos

- Creating a Query using a Web Page
- Using Power Query to Import Data from a SQL Azure Database
- Using Power Query to Transform Data During the Load Process
- Using Power Query to Import Data into a Star Schema

© Critical Path Training. 2015. All Rights Reserved

## Data Discovery

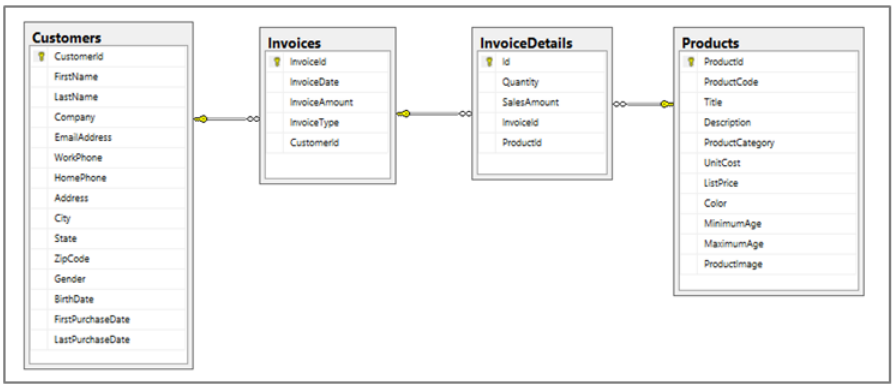
- Data can live in a variety of sources
  - Files (e.g. CSV file, Excel workbook)
  - OLTP Databases
  - OLAP Databases
  - SaaS Applications
  - Web services

As this slide shows, the data you want to analyze can live in many different places and can exist in many different formats. We will explore each of these types of data source in more detail over the next few slides.

© Critical Path Training. 2015. All Rights Reserved

# An Example OLTP Database

- Online Transaction Processing (OLTP) System
  - Used for real-time data access and transaction-based data entry
  - Optimized for faster transactions (e.g. inserts, updates & deletes)
  - Tables normalized to reduce/eliminate redundancies
  - Table schemas can be hard for business users to understand




Online Transaction Processing (OLTP) databases are used for databases that have high transaction rates. The data model of tables in this type of database is optimized to speed up transactions for operation such as inserting, updating and deleting records. Tables in an OLTP database are usually normalized to reduce or eliminate any redundancies.

Another noteworthy issue is that OLTP systems are often not designed to track historical data. For example, imagine a customer living in California makes a number of purchases over a 5 year period and then the customer moves to Florida. If you run a query to determine sales revenue by state, you will likely find that all the historic purchases for the customer are attributed to the new state Florida and not the state where the purchases were made which is California.

In a nutshell, there are several common issues that you deal with when try to build BI projects on top of an OLTP database. First, they are not designed for complex queries so you will have to join more tables together and performance might be poor since it will not have indexes to optimize these queries. Also, the table schema is often more complicated and not intuitive for the average business user.

© Critical Path Training. 2015. All Rights Reserved

## Deciding What To Measure

- You Must Determine Measurable Objectives
  - Financial (revenue, expenses, profit margin, etc.)
  - Business processes efficiency
  - Customer Satisfaction Levels

When you begin to create any type of BI system for monitoring the health and reporting on business processes, it is important to take a step back in the initial design phase and decide what your system is going to measure. This provides an opportunity to define what type of business analysis the results of the BI should provide. For example, one BI system might just need to track sales revenue in order to calculate commission for sales people. Another system with different requirements might need to analyze expenses and profit in addition to sales revenue. Determining high-level goals and capabilities is an important thing to do before getting into the nuts and bolts of beginning to extract and transform data.

© Critical Path Training. 2015. All Rights Reserved

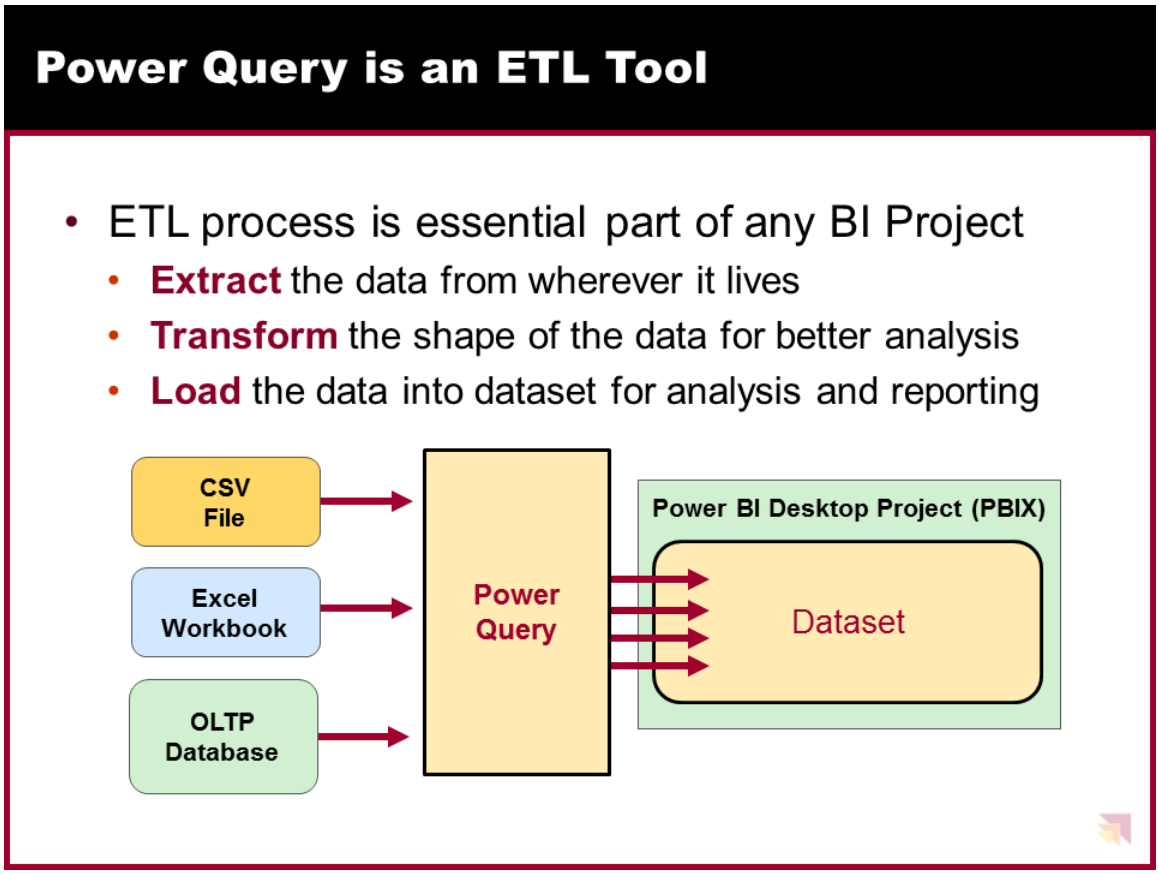
## Defining Grain Statements

- Grain statements should be defined in initial design phase
  - Grain statements helps determine requirements for BI queries
  - Grain statements can be created & understood by business users
- Example grain statements for BI project at Wingtip Toys
  - What was the total sales revenue over the last 4 years?
  - What was the sales revenue by year, quarter and month?
  - What was the sales revenue by region, state, city and zip code?
  - What was the sales revenue by category, subcategory and product?
  - What was the growth in sales revenue from month to month in 2013?
  - What was profit margin for each product by year, quarter and month?
  - Have their been any products with significantly decreasing profit margin?

A popular technique for establishing the requirements for a BI system is to define a set of grain statements. What's valuable about grain statements is that they can easily be created by business users with little or no technical experience.

It's a good idea to create a set of grain statements when you begin a BI project. From a set of grain statements you can determine what data needs to be imported as well as what queries and reports need to be generated as well.

© Critical Path Training. 2015. All Rights Reserved



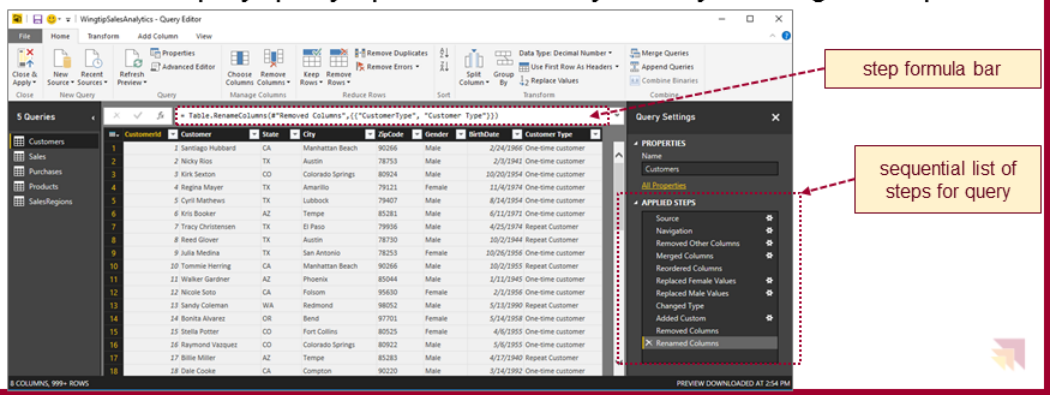

Every BI projects work with data of some type. Early in the project, you must discover where this data lives and decide how you are going to import it into a dataset that can be used for reporting and analysis. This is the phase of a BI project that is often called the "ETL" phase because it involves extracting data, transforming data and then loading data.

Power Query is a self-service ETL tool for business users. It has the ability to extract data from a variety of data source. However, the real value of Power Query is that you can perform many types of transforms to better shape the data just before it's loaded.

© Critical Path Training. 2015. All Rights Reserved

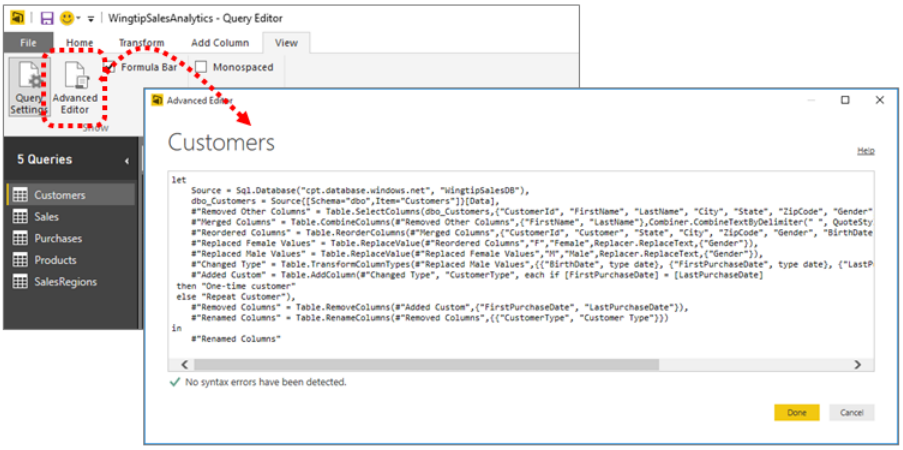
# Query Steps

- A query is created as a sequence of steps
  - Each step is a parameterized operation on the data
  - Each step has formula which can be viewed/edited in formula bar
  - Query starts with Source step to extract data from a data source
  - Additional steps added to perform transform operations on data
  - You can replay query operations one by one by clicking on steps



# Advanced Editor

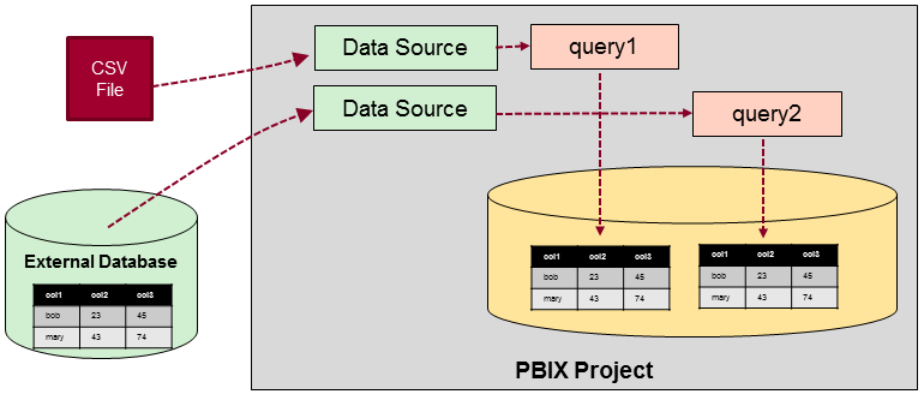
- Power Query based on "M" functional language
  - Query in Power Query saved as a set of M statements in code
  - Query Editor generates code in M behind the scenes
  - Advanced users can view & modify query code in Advanced Editor




© Critical Path Training. 2015. All Rights Reserved

# Understanding Query Input and Output

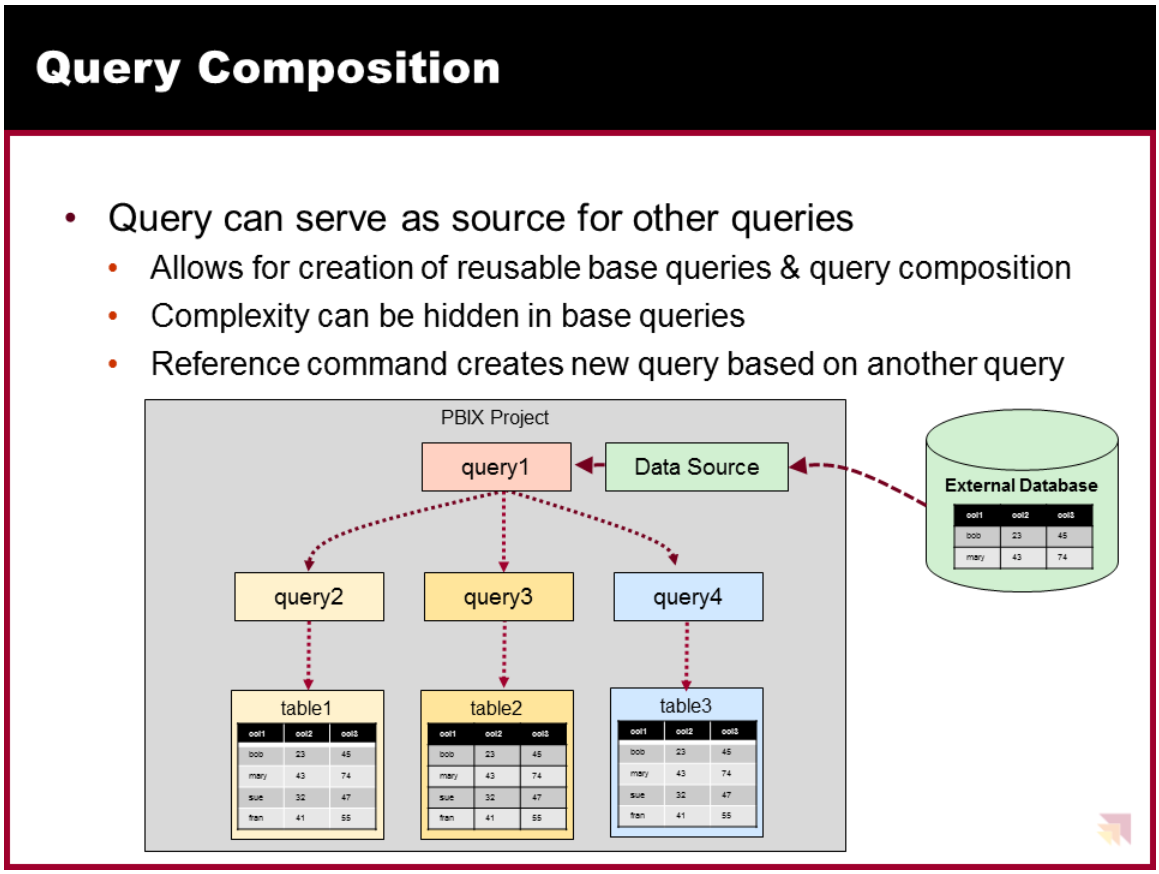
- PBIX project is container for data sources and queries
  - Queries created and saved within scope of Power BI project
  - Queries can pull data from local files
  - Queries can pull data from external content sources
  - Queries main purpose is to load imported data into data model




When working with Power BI Desktop, you create and save queries within the scope of the current project. Each query must have at least one source of data to use as its input. You can create a query from an external data source such as a CSV file, an Excel workbook, a SQL Server database or a SharePoint list.

By default, a query is configure to load its output into the project's data model. If you are creating a query whose sole purpose is to be used as a source for another query, you can disable loading and configure the query as a "connection-only" query.

© Critical Path Training. 2015. All Rights Reserved



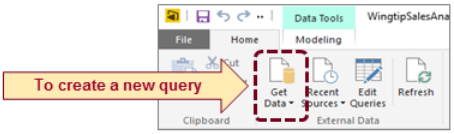

When working with Power Query in Excel 2013, you create and save queries within the scope of an Excel workbook. Each query must have at least one source of data to use as its input. To get started quickly, you can create a new query using a table in a worksheet inside the same workbook. Alternatively, you can create a query from an external data source such as an Access database, a SQL Server database or a SharePoint list.

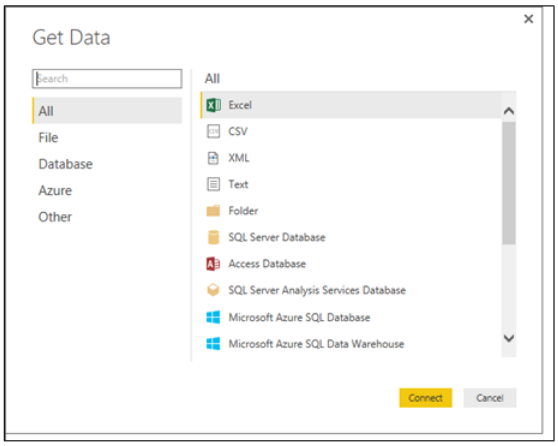
A query can also be configure to load its output into the Excel workbook. You can configure a query to load its output into a new table in an Excel worksheet or into the data model. If you are creating a query whose sole purpose is to be used as a source for another query, you can disable loading and configure the query as a "connection-only" query.

© Critical Path Training. 2015. All Rights Reserved

# Creating a New Query

- New query created using **Get Data** menu on ribbon
  - **Get Data** menu has many commands for creating new queries
  - You can create new query based on large variety of data sources
  - New query is opened in Power Query's Query Editor window

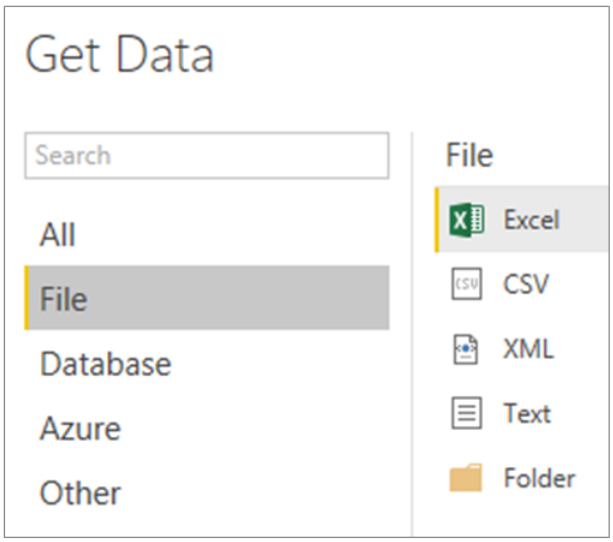



© Critical Path Training. 2015. All Rights Reserved

## Supported File-based Data Sources

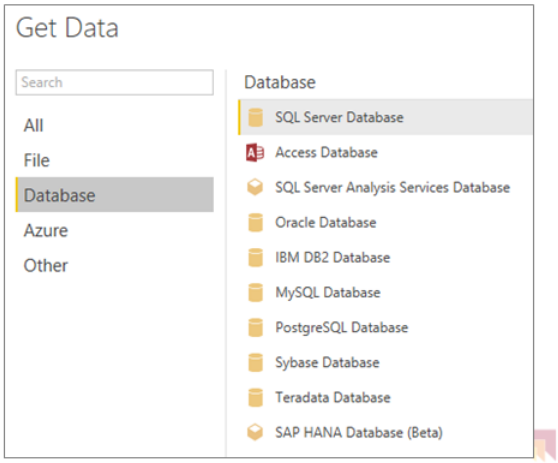
- Power Query supports common file types
  - Excel workbooks
  - CSV files
  - XML files
  - Text files
  - Folder structure




© Critical Path Training. 2015. All Rights Reserved

## Supported Databases

- Power Query supports many database systems
  - SQL Server
  - SQL Server Analysis Services
  - Access
  - Oracle
  - DB2
  - MySQL
  - PostgreSQL
  - Sybase
  - Teradata




Access Databases

SQL Server Databases

Oracle Databases

DB2 Database

MySQL

PostgreSQL

Sybase

Teradata

SQL Server Analysis Services (SSAS)

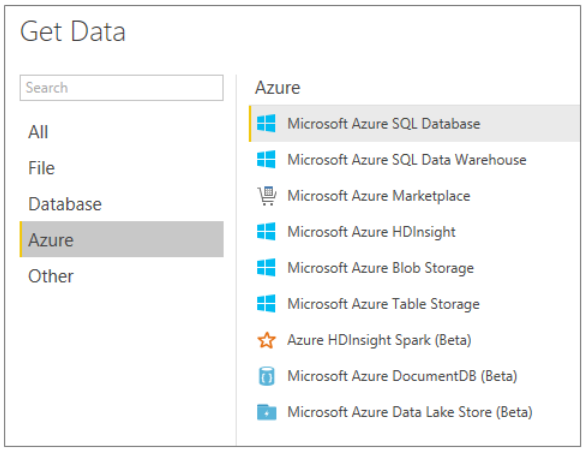
Tabular Databases

Multidimensional Databases with Cubes

© Critical Path Training. 2015. All Rights Reserved

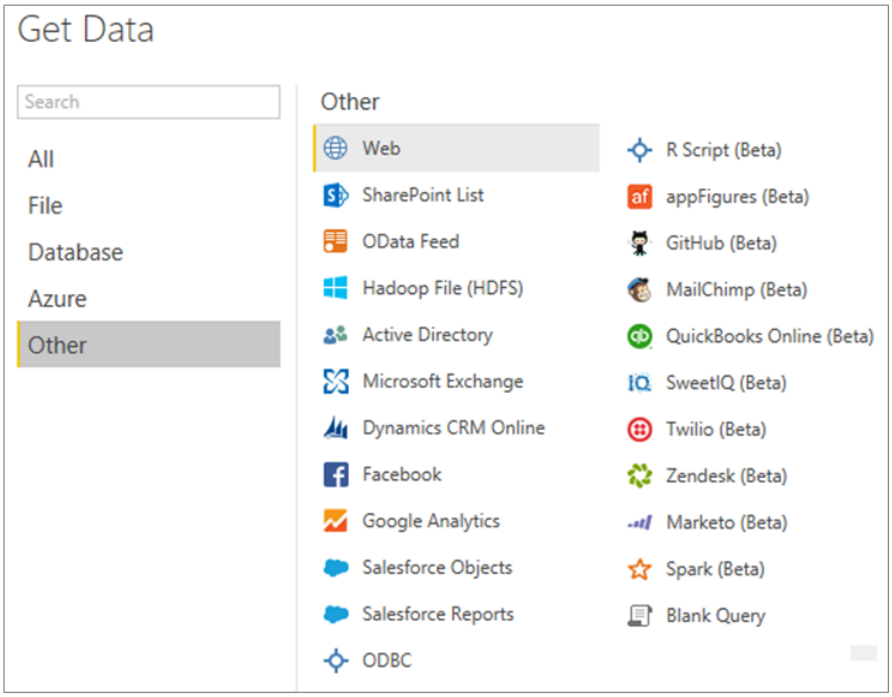
# Azure Data Sources

- Power Query supports many azure data sources




© Critical Path Training. 2015. All Rights Reserved

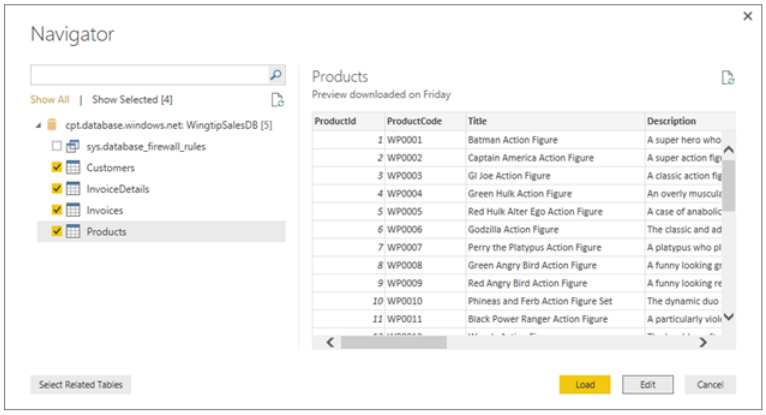
# Other Supported Data Sources




© Critical Path Training. 2015. All Rights Reserved

# Selecting Tables from a Data Source

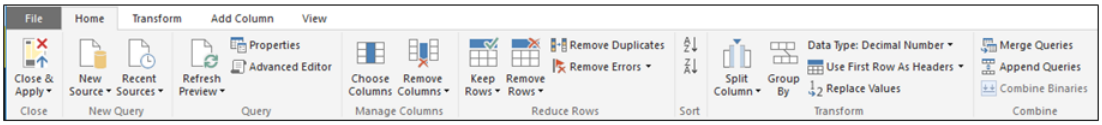
- Power Query provides Navigator dialog
  - Allows you to select tables
  - Navigator understands existing table relationships
  - Clicking **Load** will run query and import data
  - Clicking **Edit** will open queries in Query Editor window



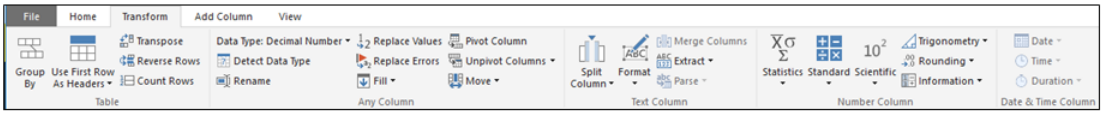

© Critical Path Training. 2015. All Rights Reserved

# Query Editor Ribbon Tabs

**Home** tab




**Transform** tab




**Add Columns** tab

**View** tab

© Critical Path Training. 2015. All Rights Reserved

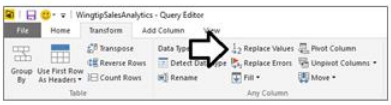
## Examples of Basic Power Query Steps

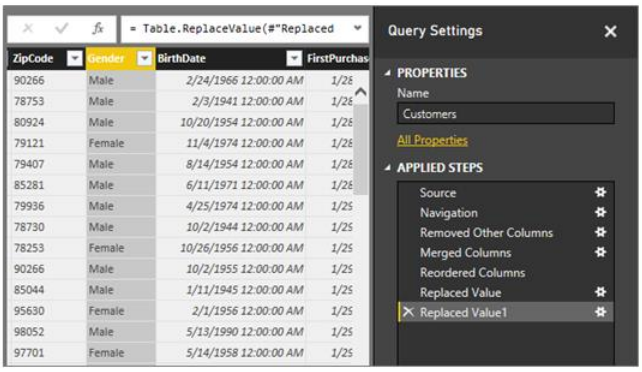
- Rename column
- Convert column type
- Format column values
- Reorder columns
- Replace column values
- Expanding related column
- Merging columns
- Splitting columns
- Adding custom column

© Critical Path Training. 2015. All Rights Reserved

# Replacing Values

- Used to substitute values during import

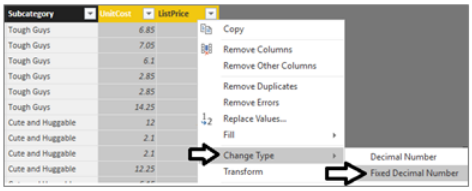





© Critical Path Training. 2015. All Rights Reserved

# Converting Column Types

- Transform data to make it more reliable
  - Convert date-time column to date column
- Transform data to make it more efficient
  - Convert decimal to fixed decimal number for currency

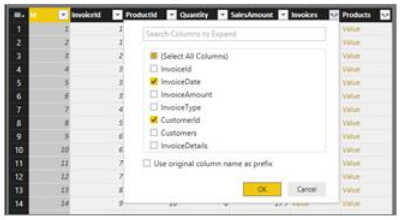



- Beware: Conversion can have destructive effect on data

© Critical Path Training. 2015. All Rights Reserved

# Expanding Related Columns

- Used to pull data from related tables
  - Saves you from performing SQL joins or VLOOKUP

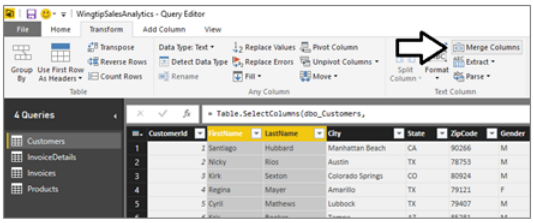


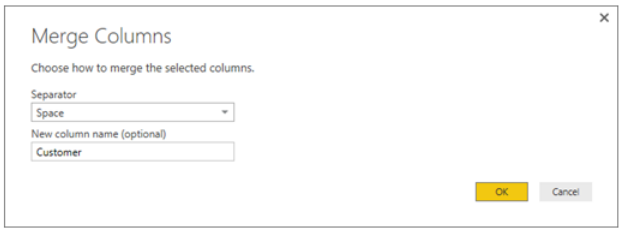

| | Id | InvoiceId | ProductId | Quantity | SalesAmount | InvoiceDate | CustomerId | Products |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 22 | 4 | 119.8 | 1/28/2012 12:00:00 AM | 1 | Value |
| 2 | 2 | 1 | 22 | 1 | 29.95 | 1/28/2012 12:00:00 AM | 1 | Value |
| 3 | 3 | 2 | 22 | 2 | 59.9 | 1/28/2012 12:00:00 AM | 2 | Value |
| 4 | 4 | 3 | 17 | 8 | 399.6 | 1/28/2012 12:00:00 AM | 3 | Value |
| 5 | 5 | 3 | 18 | 2 | 29.9 | 1/28/2012 12:00:00 AM | 3 | Value |
| 6 | 6 | 3 | 18 | 4 | 59.8 | 1/28/2012 12:00:00 AM | 3 | Value |
| 7 | 7 | 4 | 16 | 1 | 2.95 | 1/28/2012 12:00:00 AM | 4 | Value |

© Critical Path Training. 2015. All Rights Reserved

# Merging Columns

- Merge two columns into a single column



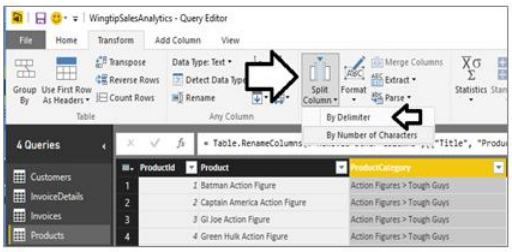




| | CustomerId | Customer |
|---|---|---|
| 1 | 1 | Santiago Hubbard |
| 2 | 2 | Nicky Rios |
| 3 | 3 | Kirk Sexton |
| 4 | 4 | Regina Mayer |
| 5 | 5 | Cyril Mathews |
| 6 | 6 | Kris Booker |
| 7 | 7 | Tracy Christensen |
| 8 | 8 | Reed Glover |

© Critical Path Training. 2015. All Rights Reserved

# Splitting Columns

- Split a single column up into two columns

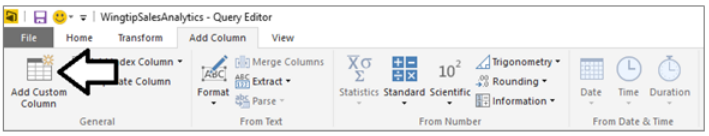


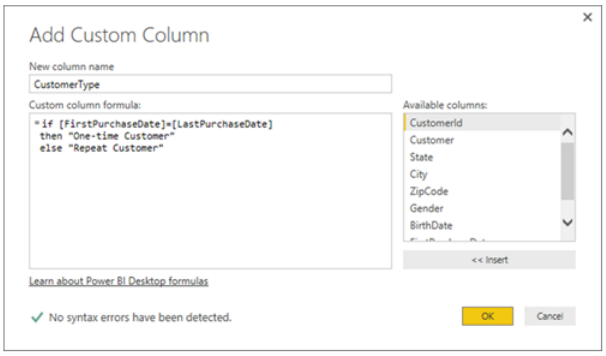
| | ProductId | Product | Category | Subcategory |
|---|---|---|---|---|
| 1 | 1 | Batman Action Figure | Action Figures | Tough Guys |
| 2 | 2 | Captain America Action Figure | Action Figures | Tough Guys |
| 3 | 3 | GI Joe Action Figure | Action Figures | Tough Guys |
| 4 | 4 | Green Hulk Action Figure | Action Figures | Tough Guys |
| 5 | 5 | Red Hulk Alter Ego Action Figure | Action Figures | Tough Guys |
| 6 | 6 | Godzilla Action Figure | Action Figures | Tough Guys |
| 7 | 7 | Perry the Platypus Action Figure | Action Figures | Cute and Huggable |
| 8 | 8 | Green Angry Bird Action Figure | Action Figures | Cute and Huggable |

© Critical Path Training. 2015. All Rights Reserved

# Adding a Custom Column

- Custom column provide custom logic
  - Logic must be written in M programming language



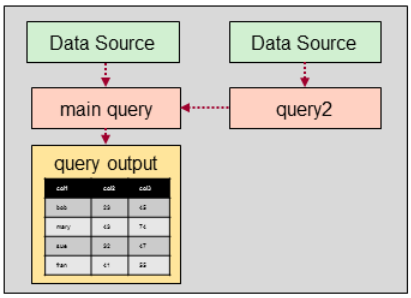



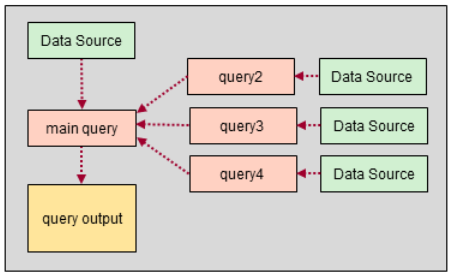

| FirstPurchaseDate | LastPurchaseDate | CustomerType |
|---|---|---|
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/29/2012 | 11/22/2015 | Repeat Customer |
| 1/29/2012 | 10/2/2015 | Repeat Customer |
| 1/29/2012 | 1/29/2012 | One-time Customer |
| 1/29/2012 | 5/6/2015 | Repeat Customer |
| 1/29/2012 | 1/29/2012 | One-time Customer |

© Critical Path Training. 2015. All Rights Reserved

# Combining Queries

- Query can be merged or appended with another query
  - Merge operation allows you combine columns from two tables
  - Append operation allows you to combine rows from two tables
- Two queries are combined into single output for loading
  - Load settings of main query determines where output is loaded
  - Secondary query acts as source for main query
  - Secondary query be can created with connection-only load setting




Power Query allows you to combine to queries. Combining queries means that you are combining two data sources into a single table as the queries output. Power Query supports two combine operations which are merge and append. A **merge** operation is used when you need to combine columns in from two tables into a single table. An append operation is used when you need to combine rows from two tables into a single table.

Consider that combining two queries produces a single output. Yet each of the queries involved has its own Power Query load setting. You configure the load setting of the main query to determine where to load the result of the merge/append operation. However, the secondary query which is only being used as the source in another query doesn't usually need to load its output anywhere in the workbook. When you are creating a query whose sole purpose is to be used as a source for a merge or append operation in another query, you can disable loading by configuring the load setting as a "connection-only" query.

What if you need to append together rows from more than one table? That's not a problem. A main query can include more than one append operation as shown in the diagram above on the right. This means a single query can append together rows from as many source tables as necessary.

© Critical Path Training. 2015. All Rights Reserved

# Merging Columns

Main Query Input

| FirstName | LastName | JobTitle | Department | Email |
|---|---|---|---|---|
| Matthew | McDermott | IT Manager | Information Technology | matthewmcdermott@wingtip.com |
| Keith | Moon | System Administrator | Information Technology | keithmoon@wingtip.com |
| Spencer | Reed | Records Manager | Information Technology | spencerreed@wingtip.com |
| Paul | Schaeflein | Software Developer | Information Technology | paulschaeflein@wingtip.com |
| Scot | Hillier | CFO | Finance | scothillier@wingtip.com |
| Bernie | Madeoff | Accountant | Finance | berniemadeoff@wingtip.com |
| Andrew | Connell | Sales Director | Sales | andrewconnell@wingtip.com |
| Chandler | Bing | Sales Representitive | Sales | chandlerbing@wingtip.com |
| Wilson | Picket | Sales Representative | Sales | wilsonpicket@wingtip.com |
| Christina | Wheeler | Marketing Director | Marketing | christinawheeler@wingtip.com |
| Katie | Bowman | Web Designer | Marketing | katiebowman@wingtip.com |
| Asif | Rehmani | Vice President of Product Development | Product Development | asifrehmani@wingtip.com |
| Ted | Pattison | Senior Toy Scientist | Product Development | tedpattison@wingtip.com |
| David | Mann | Toy Scientist | Product Development | davidmann@wingtip.com |
| Gary | Lapointe | Toy Scientist | Product Development | garylapointe@wingtip.com |

Join Column

Merge Query Input

| Email | City | State |
|---|---|---|
| matthewmcdermott@wingtip.com | Austin | TX |
| keithmoon@wingtip.com | Hollywood | CA |
| spencerreed@wingtip.com | Washington | DC |
| paulschaeflein@wingtip.com | St Paul | MN |
| scothillier@wingtip.com | Hartford | CT |
| berniemadeoff@wingtip.com | NY | NY |
| andrewconnell@wingtip.com | Jacksonville | FL |
| chandlerbing@wingtip.com | NY | NY |
| wilsonpicket@wingtip.com | San Fransisco | CA |
| christinawheeler@wingtip.com | Atlanta | GE |
| katiebowman@wingtip.com | Autsin | TX |
| asifrehmani@wingtip.com | Chicago | IL |
| tedpattison@wingtip.com | Tampa | FL |
| davidmann@wingtip.com | Philidephia | PA |
| garylapointe@wingtip.com | Dever | CO |

Join Column

Query Output

| First Name | Last Name | Email | Job Title | Department | City | State |
|---|---|---|---|---|---|---|
| Matthew | McDermott | matthewmcdermott@wingtip.com | IT Manager | Information Technology | Austin | TX |
| Keith | Moon | keithmoon@wingtip.com | System Administrator | Information Technology | Hollywood | CA |
| Spencer | Reed | spencerreed@wingtip.com | Records Manager | Information Technology | Washington | DC |
| Paul | Schaeflein | paulschaeflein@wingtip.com | Software Developer | Information Technology | St Paul | MN |
| Scot | Hillier | scothillier@wingtip.com | CFO | Finance | Hartford | CT |
| Bernie | Madeoff | berniemadeoff@wingtip.com | Accountant | Finance | NY | NY |
| Andrew | Connell | andrewconnell@wingtip.com | Sales Director | Sales | Jacksonville | FL |
| Chandler | Bing | chandlerbing@wingtip.com | Sales Representitive | Sales | NY | NY |
| Wilson | Picket | wilsonpicket@wingtip.com | Sales Representative | Sales | San Fransisco | CA |
| Christina | Wheeler | christinawheeler@wingtip.com | Marketing Director | Marketing | Atlanta | GE |
| Katie | Bowman | katiebowman@wingtip.com | Web Designer | Marketing | Autsin | TX |
| Asif | Rehmani | asifrehmani@wingtip.com | Vice President of Product Development | Product Development | Chicago | IL |
| Ted | Pattison | tedpattison@wingtip.com | Senior Toy Scientist | Product Development | Tampa | FL |
| David | Mann | davidmann@wingtip.com | Toy Scientist | Product Development | Philidephia | PA |
| Gary | Lapointe | garylapointe@wingtip.com | Toy Scientist | Product Development | Dever | CO |

© Critical Path Training. 2015. All Rights Reserved

# Appending Rows

Main Query Input

| FirstName | LastName | Company | Email | City | State | ZipCode |
|---|---|---|---|---|---|---|
| Joe | Stephens | Initech | Joe.Stephens@Initech.com | Sacramento | CA | 95823 |
| Basil | Frazier | Nordyne Defense Dynamics | Basil.Frazier@NordyneDefenseDynamics.com | Sacramento | CA | 95823 |
| Ignacio | Duran | Yoyodyne Propulsion Systems | Ignacio.Duran@YoyodynePropulsionSystems.com | Beaverton | OR | 97005 |
| Alvaro | Brock | W.C. Boggs & Co. | Alvaro.Brock@W.C.Boggs&Co..com | Beaverton | OR | 97005 |
| Alec | Miller | Krusty Burger | Alec.Miller@KrustyBurger.com | Vancouver | WA | 98662 |
| Maureen | Griffin | VersaLife Corporation | Maureen.Griffin@VersaLifeCorporation.com | Vancouver | WA | 98662 |
| Guillermo | Sykes | Soar Airlines | Guillermo.Sykes@SoarAirlines.com | Sacramento | CA | 95818 |
| Solomon | Warner | Wayne Enterprises | Solomon.Warner@WayneEnterprises.com | Seattle | WA | 98125 |
| Humberto | Petersen | W.C. Boggs & Co. | Humberto.Petersen@W.C.Boggs&Co..com | San Francisco | CA | 94118 |
| Keven | Griffith | Fabrikam | Keven.Griffith@Fabrikam.com | Bend | OR | 97701 |

Append Query Input

| FirstName | LastName | Company | Email | City | State | ZipCode |
|---|---|---|---|---|---|---|
| Rich | Pierce | The Regal Beagle | Rich.Pierce@TheRegalBeagle.com | Portland | OR | 97216 |
| Ronnie | Donaldson | Union Aerospace Corporation | Ronnie.Donaldson@UnionAerospaceCorporation.com | Vancouver | WA | 98684 |
| Willard | Frazier | Benthic Petroleum | Willard.Frazier@BenthicPetroleum.com | Portland | OR | 97220 |
| Marina | Caldwell | Ewing Oil | Marina.Caldwell@EwingOil.com | Portland | OR | 97205 |
| Milagros | Mercer | Hishii Industries | Milagros.Mercer@HishiiIndustries.com | Salem | OR | 97301 |
| Kirsten | Little | Binford | Kirsten.Little@Binford.com | Ventura | CA | 93003 |
| Terri | Ferrell | Black Mesa Research Facility | Terri.Ferrell@BlackMesaResearchFacility.com | Portland | OR | 97205 |
| Francine | Doyle | Peach Pit | Francine.Doyle@PeachPit.com | Vancouver | WA | 98662 |
| Dannie | Powers | Deon International | Dannie.Powers@DeonInternational.com | Issaquah | WA | 98027 |
| Noah | Best | Shinra Electric Power Company | Noah.Best@ShinraElectricPowerCompany.com | Portland | OR | 97217 |
| Keenan | Holmes | Biffco | Keenan.Holmes@Biffco.com | Portland | OR | 97220 |
| Douglas | Maynard | Cyberdyne Systems | Douglas.Maynard@CyberdyneSystems.com | San Francisco | CA | 94118 |
| Gerald | Harrington | Yoyodyne Propulsion Systems | Gerald.Harrington@YoyodynePropulsionSystems.com | Portland | OR | 97217 |
| Josue | Robinson | North Western Railway | Josue.Robinson@NorthWesternRailway.com | Beaverton | OR | 97005 |

Query Output

| FirstName | LastName | Company | Email | City | State | ZipCode |
|---|---|---|---|---|---|---|
| Joe | Stephens | Initech | Joe.Stephens@Initech.com | Sacramento | CA | 95823 |
| Basil | Frazier | Nordyne Defense Dynamics | Basil.Frazier@NordyneDefenseDynamics.com | Sacramento | CA | 95823 |
| Ignacio | Duran | Yoyodyne Propulsion Systems | Ignacio.Duran@YoyodynePropulsionSystems.com | Beaverton | OR | 97005 |
| Alvaro | Brock | W.C. Boggs & Co. | Alvaro.Brock@W.C.Boggs&Co..com | Beaverton | OR | 97005 |
| Alec | Miller | Krusty Burger | Alec.Miller@KrustyBurger.com | Vancouver | WA | 98662 |
| Maureen | Griffin | VersaLife Corporation | Maureen.Griffin@VersaLifeCorporation.com | Vancouver | WA | 98662 |
| Guillermo | Sykes | Soar Airlines | Guillermo.Sykes@SoarAirlines.com | Sacramento | CA | 95818 |
| Solomon | Warner | Wayne Enterprises | Solomon.Warner@WayneEnterprises.com | Seattle | WA | 98125 |
| Humberto | Petersen | W.C. Boggs & Co. | Humberto.Petersen@W.C.Boggs&Co..com | San Francisco | CA | 94118 |
| Keven | Griffith | Fabrikam | Keven.Griffith@Fabrikam.com | Bend | OR | 97701 |
| Rich | Pierce | The Regal Beagle | Rich.Pierce@TheRegalBeagle.com | Portland | OR | 97216 |
| Ronnie | Donaldson | Union Aerospace Corporation | Ronnie.Donaldson@UnionAerospaceCorporation.com | Vancouver | WA | 98684 |
| Willard | Frazier | Benthic Petroleum | Willard.Frazier@BenthicPetroleum.com | Portland | OR | 97220 |
| Marina | Caldwell | Ewing Oil | Marina.Caldwell@EwingOil.com | Portland | OR | 97205 |
| Milagros | Mercer | Hishii Industries | Milagros.Mercer@HishiiIndustries.com | Salem | OR | 97301 |
| Kirsten | Little | Binford | Kirsten.Little@Binford.com | Ventura | CA | 93003 |
| Terri | Ferrell | Black Mesa Research Facility | Terri.Ferrell@BlackMesaResearchFacility.com | Portland | OR | 97205 |
| Francine | Doyle | Peach Pit | Francine.Doyle@PeachPit.com | Vancouver | WA | 98662 |
| Dannie | Powers | Deon International | Dannie.Powers@DeonInternational.com | Issaquah | WA | 98027 |
| Noah | Best | Shinra Electric Power Company | Noah.Best@ShinraElectricPowerCompany.com | Portland | OR | 97217 |
| Keenan | Holmes | Biffco | Keenan.Holmes@Biffco.com | Portland | OR | 97220 |
| Douglas | Maynard | Cyberdyne Systems | Douglas.Maynard@CyberdyneSystems.com | San Francisco | CA | 94118 |
| Gerald | Harrington | Yoyodyne Propulsion Systems | Gerald.Harrington@YoyodynePropulsionSystems.com | Portland | OR | 97217 |
| Josue | Robinson | North Western Railway | Josue.Robinson@NorthWesternRailway.com | Beaverton | OR | 97005 |

© Critical Path Training. 2015. All Rights Reserved

# Pivoting Columns

- Pivot column adds its values are new columns
  - Create table layout like PivotTable

| City | Year | Population |
|---|---|---|
| New York, NY | 2010 | 8175133 |
| New York, NY | 2000 | 8008278 |
| New York, NY | 1990 | 7322564 |
| Los Angeles, CA | 2010 | 3792621 |
| Los Angeles, CA | 2000 | 3694820 |
| Los Angeles, CA | 1990 | 3485398 |
| Chicago, IL | 2010 | 2695598 |
| Chicago, IL | 2000 | 2896016 |
| Chicago, IL | 1990 | 2783726 |
| Houston, TX | 2010 | 2100263 |
| Houston, TX | 2000 | 1953631 |
| Houston, TX | 1990 | 1630553 |
| Philadelphia, PA | 2010 | 1526006 |
| Philadelphia, PA | 2000 | 1517550 |
| Philadelphia, PA | 1990 | 1585577 |
| Phoenix, AZ | 2010 | 1445632 |
| Phoenix, AZ | 2000 | 1321045 |
| Phoenix, AZ | 1990 | 983403 |

| City | 2010 | 2000 | 1990 |
|---|---|---|---|
| New York, NY | 8175133 | 8008278 | 7322564 |
| Los Angeles, CA | 3792621 | 3694820 | 3485398 |
| Chicago, IL | 2695598 | 2896016 | 2783726 |
| Houston, TX | 2100263 | 1953631 | 1630553 |
| Philadelphia, PA | 1526006 | 1517550 | 1585577 |
| Phoenix, AZ | 1445632 | 1321045 | 983403 |
| San Antonio, TX | 1327407 | 1144646 | 935933 |
| San Diego, TX | 1307402 | 1223400 | 1110549 |
| Dallas, TX | 1197816 | 1188580 | 1006877 |
| San Jose, CA | 945942 | 894943 | 782248 |

© Critical Path Training. 2015. All Rights Reserved

# Unpivoting Columns

- Unpivot columns to collapse them into single column
  - Removes PivotTable layout
  - Can be useful to prepare data for charting and analysis

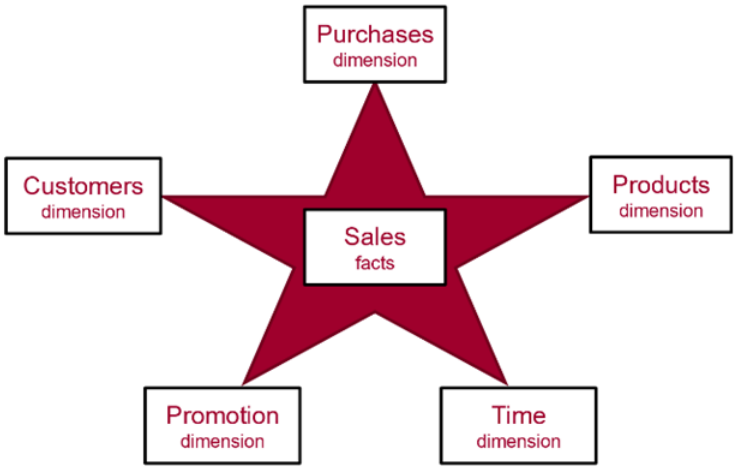
| City | 2010 | 2000 | 1990 |
|---|---|---|---|
| New York, NY | 8175133 | 8008278 | 7322564 |
| Los Angeles, CA | 3792621 | 3694820 | 3485398 |
| Chicago, IL | 2695598 | 2896016 | 2783726 |
| Houston, TX | 2100263 | 1953631 | 1630553 |
| Philadelphia, PA | 1526006 | 1517550 | 1585577 |
| Phoenix, AZ | 1445632 | 1321045 | 983403 |
| San Antonio, TX | 1327407 | 1144646 | 935933 |
| San Diego, TX | 1307402 | 1223400 | 1110549 |
| Dallas, TX | 1197816 | 1188580 | 1006877 |
| San Jose, CA | 945942 | 894943 | 782248 |

| City | Year | Population |
|---|---|---|
| New York, NY | 2010 | 8175133 |
| New York, NY | 2000 | 8008278 |
| New York, NY | 1990 | 7322564 |
| Los Angeles, CA | 2010 | 3792621 |
| Los Angeles, CA | 2000 | 3694820 |
| Los Angeles, CA | 1990 | 3485398 |
| Chicago, IL | 2010 | 2695598 |
| Chicago, IL | 2000 | 2896016 |
| Chicago, IL | 1990 | 2783726 |
| Houston, TX | 2010 | 2100263 |
| Houston, TX | 2000 | 1953631 |
| Houston, TX | 1990 | 1630553 |
| Philadelphia, PA | 2010 | 1526006 |
| Philadelphia, PA | 2000 | 1517550 |
| Philadelphia, PA | 1990 | 1585577 |
| Phoenix, AZ | 2010 | 1445632 |
| Phoenix, AZ | 2000 | 1321045 |
| Phoenix, AZ | 1990 | 983403 |

© Critical Path Training. 2015. All Rights Reserved

# Data Modeling using a Star Schema

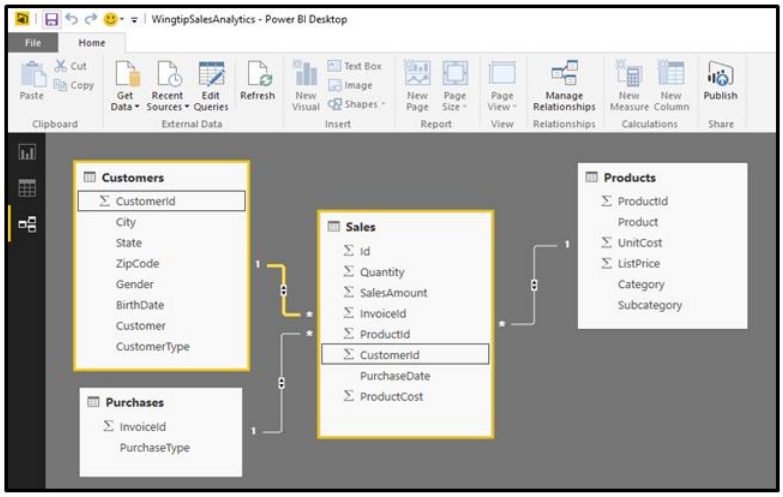
- OLAP Modeling often based on Star Schema
  - Tables defined as fact tables or dimension tables
  - Fact tables related to dimension table using 1-to-many relationships




© Critical Path Training. 2015. All Rights Reserved

# Building a Star Schema with Power Query

- Eliminate need to join three tables
  - Pull CustomerId column into Sales table



# Module 03 Lab: Using the Power Query Features of Power BI Desktop

**Lab Time**: 40 minutes

**Lab Folder**: C:\Student\Modules\PowerQuery\Lab\

**Lab Overview**: In this lab you will begin by creating a new Power BI Desktop project and saving it as a PBIX file. Next, you will learn to work with the Power Query features of Power BI Desktop to extract data from a SQL Azure database and to transform the data as it is loaded into the data model. This is the first lab in a sequence of labs that continue with the same PBIX file. In other words, the labs that follow will build upon the work you do in this lab.

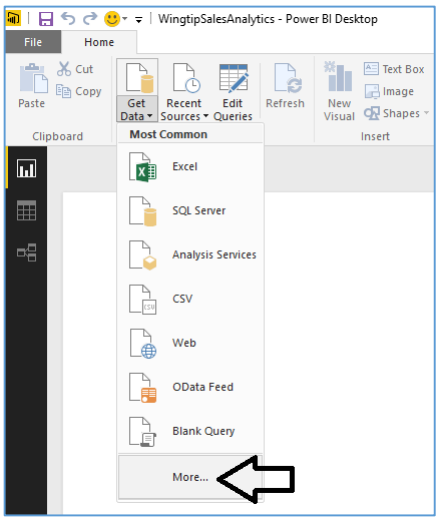
## Exercise 1: Importing Data from a SQL Azure Database

In this exercise you will create and save a new Power BI Desktop project. After that, you will connect to a SQL Azure database and import data into Power BI Desktop using its Power Query features.

1. Launch Power BI Desktop to start a new project.
2. Save the new project as **WingtipSalesAnalytics.pbix** using the following path.

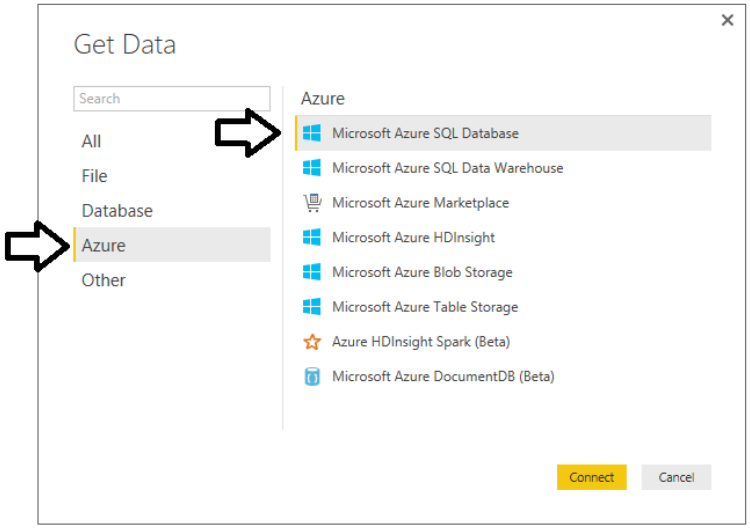
```
C:\Student\Projects\WingtipSalesAnalytics.pbix
```

3. Drop down the **Get Data** menu button on the ribbon and click **More…**.

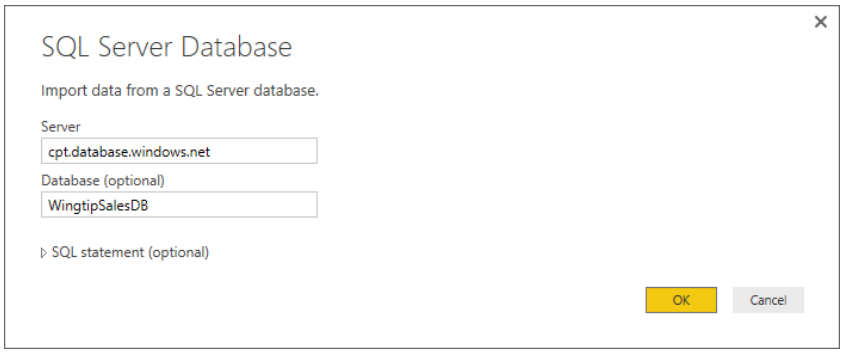



4. On the **Get Data** dialog, select **Azure** in the list on the left. Next, select **Microsoft SQL Azure Database** on the right and then click the **Connect** button.

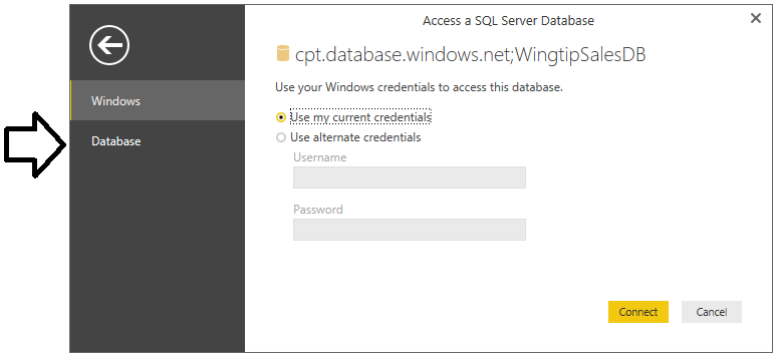
© Critical Path Training. 2015. All Rights Reserved




5. When you are prompted with the **SQL Server Database** dialog, enter the following values for the **Server** and **Database**.
   a) Server: cpt.database.windows.net
   b) Database: WingtipSalesDB
6. When the **SQL Server Database** dialog appears as it does in the following screenshot, click the **OK** button to continue.



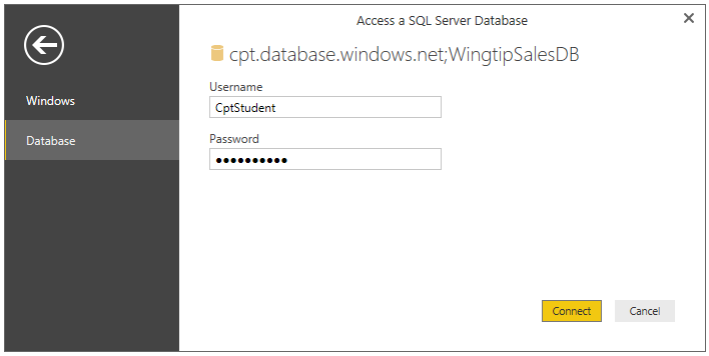

7. At this point, you will be prompted by the **Access a SQL Server Database** dialog. Click on **Database** on the left side of the dialog so that you can enter the credentials for a standard SQL account instead of using Windows authentication.



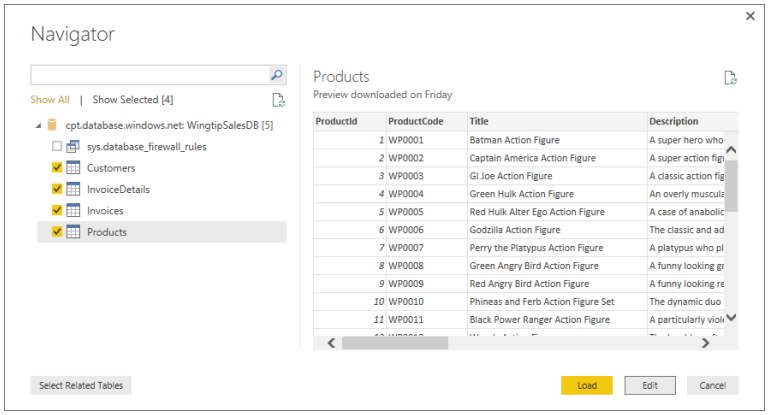

8. Enter the following credentials for a SQL user account that has been configured with read access to the database.
   a) Username: **CptStudent**
   b) Password: **pass@word1**

© Critical Path Training. 2015. All Rights Reserved

9. Once you have entered the credentials the **Access a SQL Server Database** dialog, click the **Connect** button to continue.

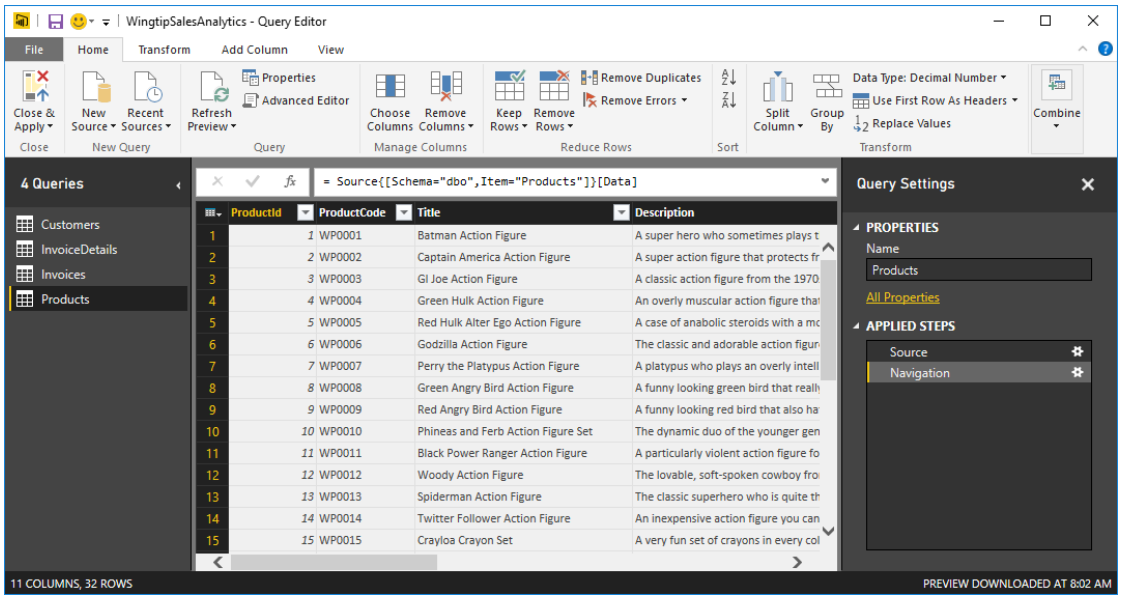



10. At this point, Power BI Desktop should be able to establish a connection to the database and then prompt you with the **Navigator** dialog. The **Navigator** dialog allows you to select the tables you would like to import into your PBIX project.
11. In the **Navigator** dialog, select the **Customers** table, the **InvoiceDetails** table, the **Invoices** table and the **Products** table as shown in the following screenshot. Once you have selected these four tables, click the **Edit** button to create a query for each of these tables and to open the **Query Editor** dialog.

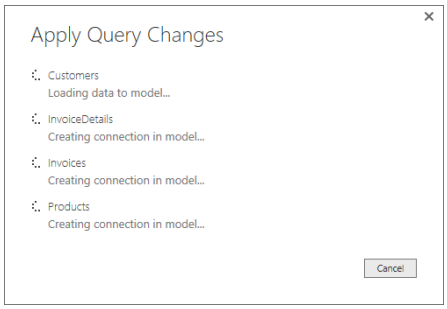



12. When you inspect the **Query Editor** window, you should now be able to observe that Power BI Desktop has created a new query for each of the four tables that you selected in the **Navigator** dialog. You will not begin to edit these queries until the next exercise. However, now you should take a moment to inspect each query by clicking on the query name in the **Queries** list on the left. You should be able to observe that each query is initially created to return the data just as it is defined in the underlying table.

© Critical Path Training. 2015. All Rights Reserved

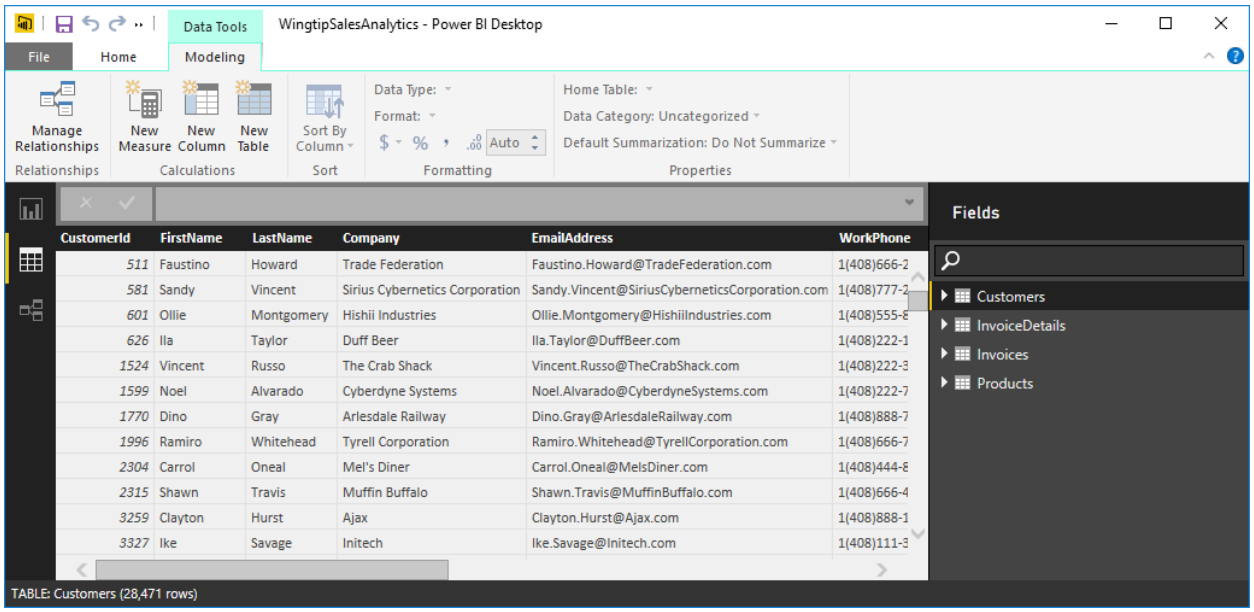



13. Click the **Close and Apply** button in the **Query Editor** window. This will close the **Query Editor** window and start the process of extracting the data from the SQL Azure database and importing it into the current project's data model using a local cache on your student workstation. The **Apply Query Changes** dialog will be displayed with spinning icons as Power BI Desktop imports that data.

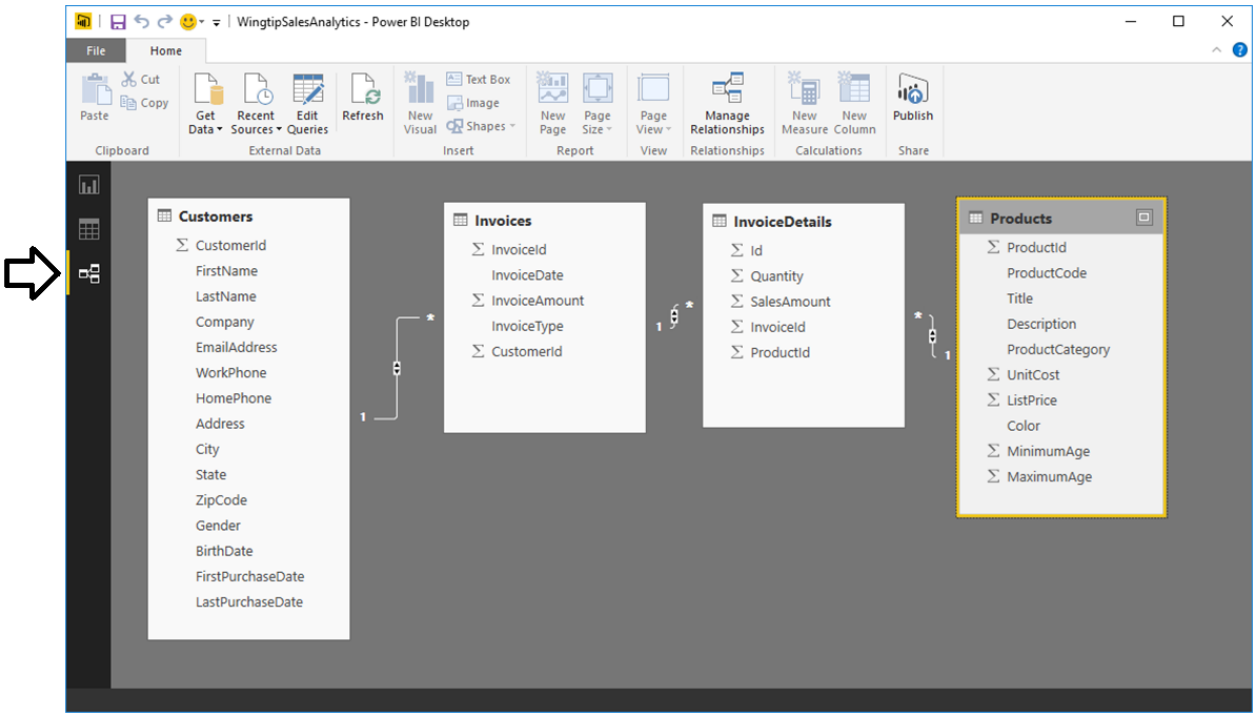



14. Once the data from all four queries has been imported, examine the **Fields** list on the right-hand side of the Power BI Desktop window. You should be able to see that four tables have been created from the four queries.

© Critical Path Training. 2015. All Rights Reserved




15. Click on the bottom icon in the sidebar to navigate to relationship view. You should be able to see each of the four tables. You should also be able to see that Power BI Desktop has automatically created relationships between the tables as they were imported into the data model. Use your mouse to move and resize the tables in relationship view so you can see all the fields in each table.



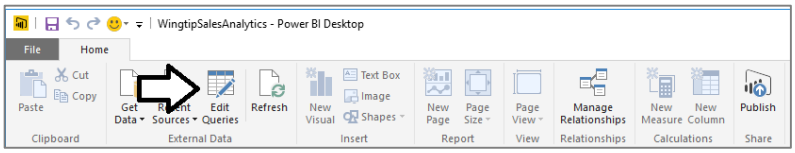

16. Save your work in the current project by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 2: Using Power Query to Transform and Reshape Customer Data

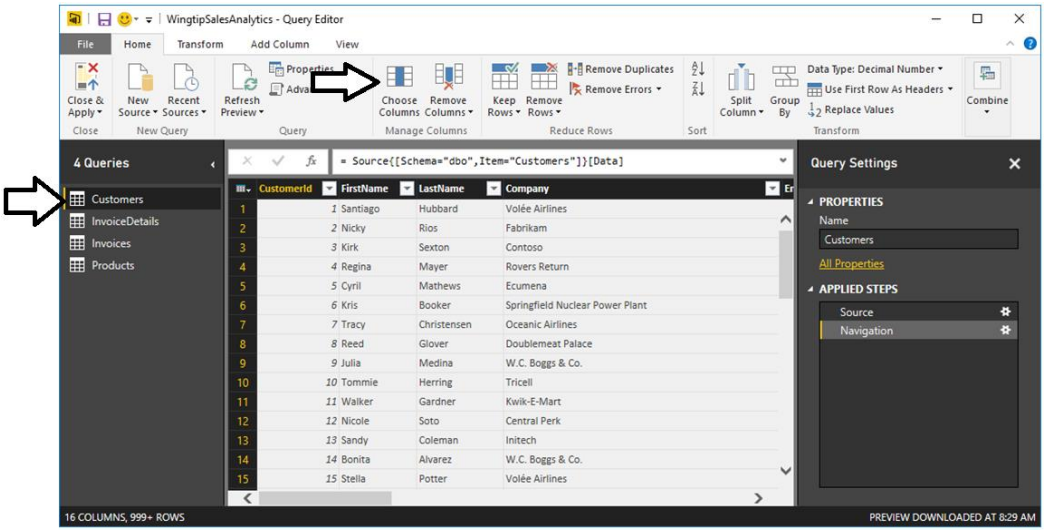
In the following exercise, you will use the **Query Editor** window to modify the **Customers** query to perform transforms on customer data as it is being loaded into the data model.

© Critical Path Training. 2015. All Rights Reserved

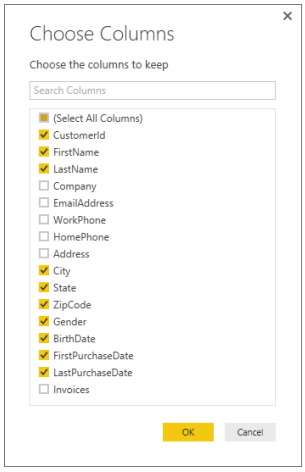
1. Make sure you have the **WingtipSalesAnalytics.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the Query Editor window.




3. Make sure the **Customers** query is selected in the **Queries** list on the left-hand side of the Query Editor window. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.




4. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all column. Next, select the checkboxes for **CustomerId**, **FirstName**, **LastName**, **City**, **State**, **ZipCode**, **Gender**, **BirthDate**, **FirstPurchaseDate** and **LastPurchaseDate** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.

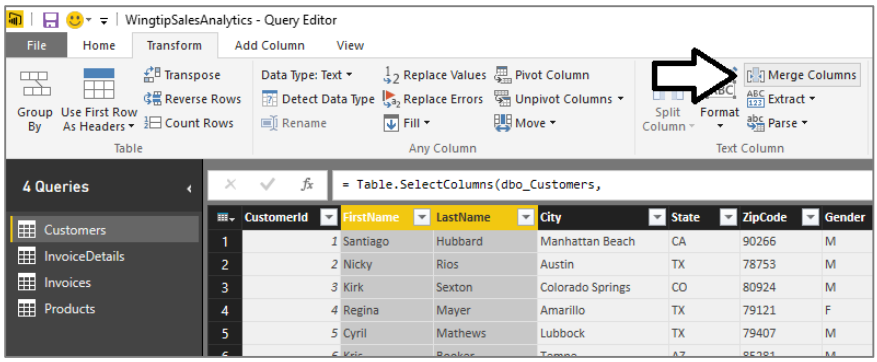



5. You should be able to see that the Query Editor window now only shows the columns that you selected.
6. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
   a) Begin by clicking on the **Transform** tab in the ribbon.
   b) Next, select the **FirstName** column by clicking on its column header.

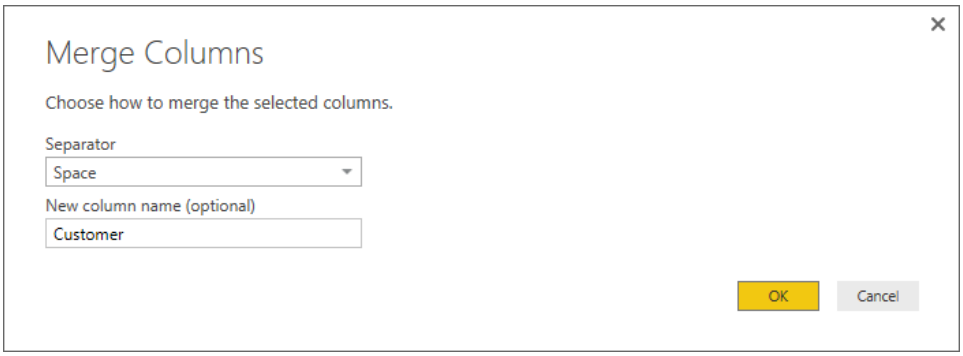
© Critical Path Training. 2015. All Rights Reserved

c) Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.

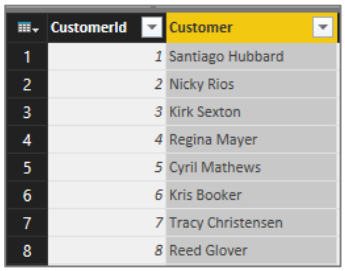
d) With both the **FirstName** column and the **LastName** column selected, click the **Merge Column** button in the ribbon to display the **Merge Column** dialog.




e) In the **Merge Column** dialog, drop down the **Separator** control and select a value of **Space**. Add a **New column name** value of **Customer** and click the **OK** button to modify the underlying query with your changes.



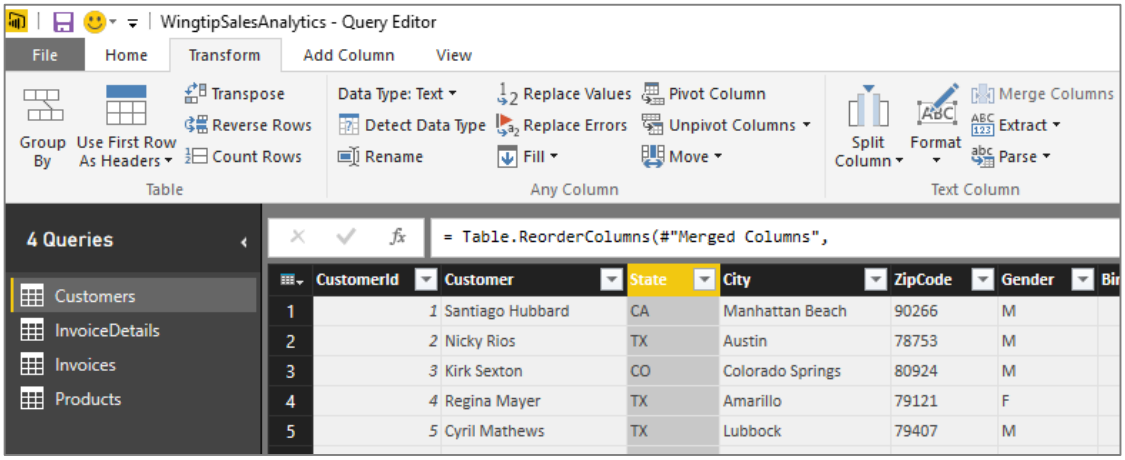

f) You should now be able to see that the **FirstName** column and the **LastName** column have been replaced with a single merged column named **Customer**.



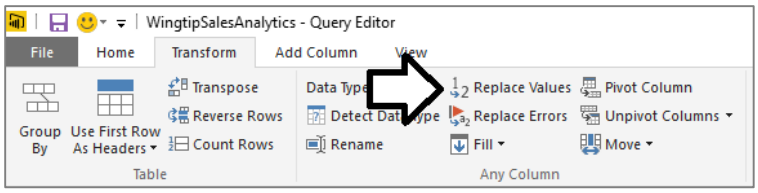
| | CustomerId | Customer |
|---|---|---|
| 1 | 1 | Santiago Hubbard |
| 2 | 2 | Nicky Rios |
| 3 | 3 | Kirk Sexton |
| 4 | 4 | Regina Mayer |
| 5 | 5 | Cyril Mathews |
| 6 | 6 | Kris Booker |
| 7 | 7 | Tracy Christensen |
| 8 | 8 | Reed Glover |

7. The **State** column is currently to the right of the **City** column. Move the **State** column so that it is repositioned to the left of the **City** column. Accomplish this by clicking on the column header for the **State** column and leaving the mouse button down. Drag the **State** column to the left of the **City** column and release the mouse button.

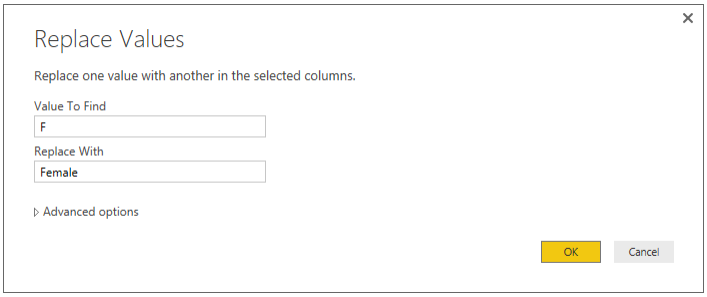
© Critical Path Training. 2015. All Rights Reserved



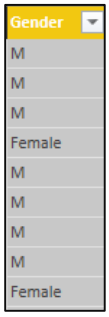

8. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
   a) Make sure the **Transform** tab is the active tab in the ribbon.
   b) Select the **Gender** column by clicking its column header.
   c) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.




   d) In the **Replace Value** dialog, enter a value of **F** in the **Value to Find** textbox and enter a value of **Female** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.




   e) You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.

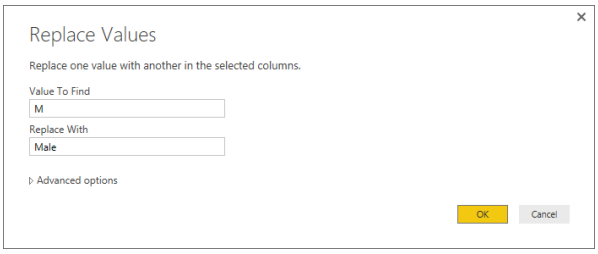



   f) Make sure the **Gender** column is still selected.

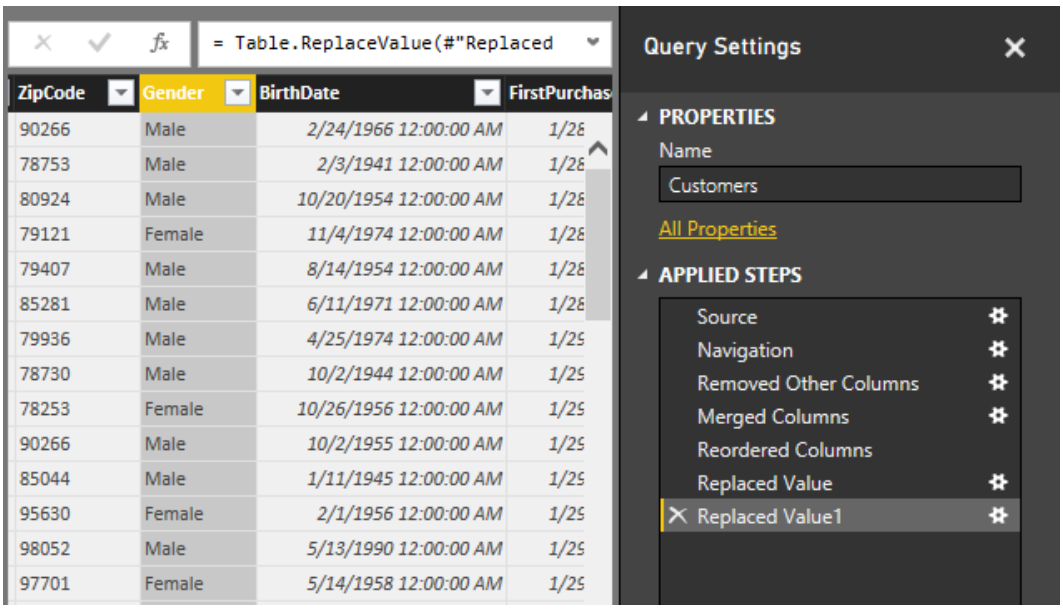
© Critical Path Training. 2015. All Rights Reserved

g) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.

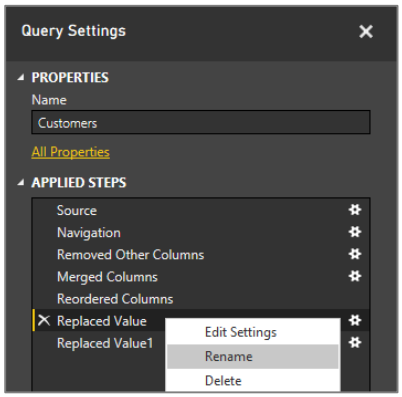
h) In the **Replace Value** dialog, enter a value of **M** in the **Value to Find** textbox and enter a value of **Male** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.




i) You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**. If you inspect the **APPLIED STEPS** list in the **Query Settings** pane, you should be able to see that there are two steps at the end that have been given the generic names of **Replaced Value** and **Replaced Value 1**.

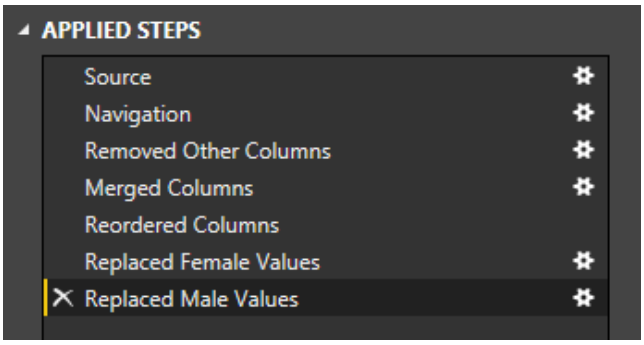



j) In order to promote higher levels of maintainability, it's often a good idea to rename steps with names such as of **Replaced Value** and **Replaced Value 1**. Rename the **Replaced Values** step by right-clicking it and clicking the **Rename** command to place the step name in edit mode. Modify the name of this step to **Replaced Female Values**.

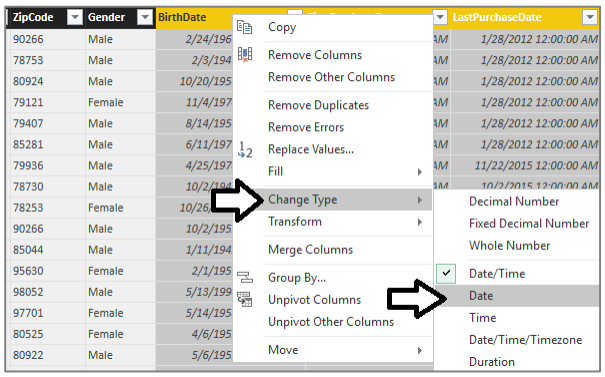



k) Using the same technique, rename the **Replaced Value 1** step to **Replaced Male Values**.

© Critical Path Training. 2015. All Rights Reserved




9. Change the column type of BirthDate, FirstPurchasedDate and LastPurchasedDate from Date/Time to Date.
   a) Select the **BirthDate** column by clicking its column header.
   b) Hold down the **SHIFT** key and click the column headers for **FirstPurchasedDate** and **LastPurchasedDate** so that all three columns are selected.
   c) Right-click on any one of the selected columns and select the **Change Type > Date** command.

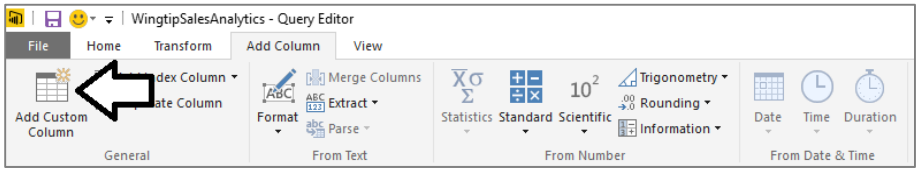



   d) You should see that the three columns now show values with a date but without a time.

| BirthDate | FirstPurchaseDate | LastPurchaseDate |
|---|---|---|
| 2/24/1966 | 1/28/2012 | 1/28/2012 |
| 2/3/1941 | 1/28/2012 | 1/28/2012 |
| 10/20/1954 | 1/28/2012 | 1/28/2012 |
| 11/4/1974 | 1/28/2012 | 1/28/2012 |
| 8/14/1954 | 1/28/2012 | 1/28/2012 |
| 6/11/1971 | 1/28/2012 | 1/28/2012 |
| 4/25/1974 | 1/29/2012 | 11/22/2015 |
| 10/2/1944 | 1/29/2012 | 10/2/2015 |
| 10/26/1956 | 1/29/2012 | 1/29/2012 |
| 10/2/1955 | 1/29/2012 | 5/6/2015 |

10. Add a new custom column named **CustomerType** to indicates whether the customer is a repeat customer or not. You will accomplish by writing a DAX a formula which compares the **FirstPurchaseDate** column to **LastPurchaseDate** column.
    a) Begin by activating the **Add Column** tab in the ribbon.
    b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.

© Critical Path Training. 2015. All Rights Reserved



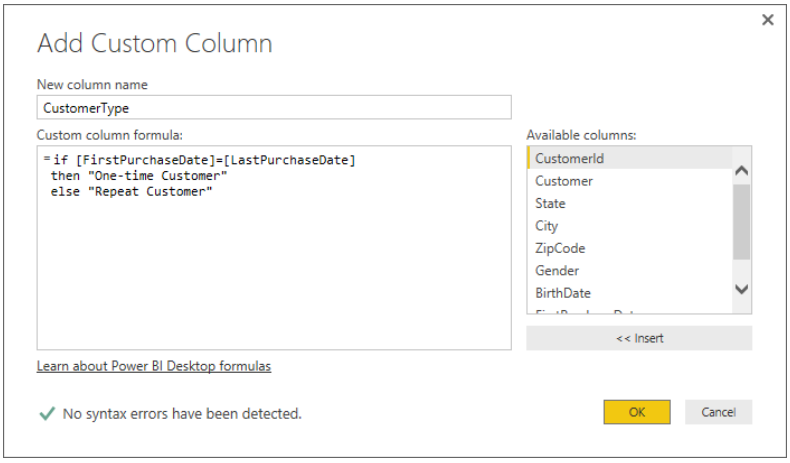

c) In the **Add Custom Column** dialog, add a value of **CustomerType** in the **New column name** textbox.

In this particular scenario, you are working under the assumption that the customer is a repeat customer when the **FirstPurchaseDate** column and the **LastPurchaseDate** column are not equal indicating the customer has made two or more purchases.

d) In the **Custom column formula** textbox, enter the following formula.

```
if [LastPurchaseDate]=[FirstPurchaseDate]
then "One-time Customer"
else "Repeat Customer"
```

e) When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.




f) You should be able to verify that the new **CustomerType** column has a value of **Repeat Customer** when the current customer has a **FirstPurchaseDate** column value that is not equal to the **LastPurchaseDate** column value. When these column values are equal, the **CustomerType** column has a value of **One-time Customer**.

| FirstPurchaseDate | LastPurchaseDate | CustomerType |
|---|---|---|
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/29/2012 | 11/22/2015 | Repeat Customer |
| 1/29/2012 | 10/2/2015 | Repeat Customer |
| 1/29/2012 | 1/29/2012 | One-time Customer |
| 1/29/2012 | 5/6/2015 | Repeat Customer |
| 1/29/2012 | 1/29/2012 | One-time Customer |

11. Now, that you have used the **FirstPurchaseDate** column and the **LastPurchaseDate** column to calculate the value of the **CustomerType** column, you can delete them because they are no longer needed.

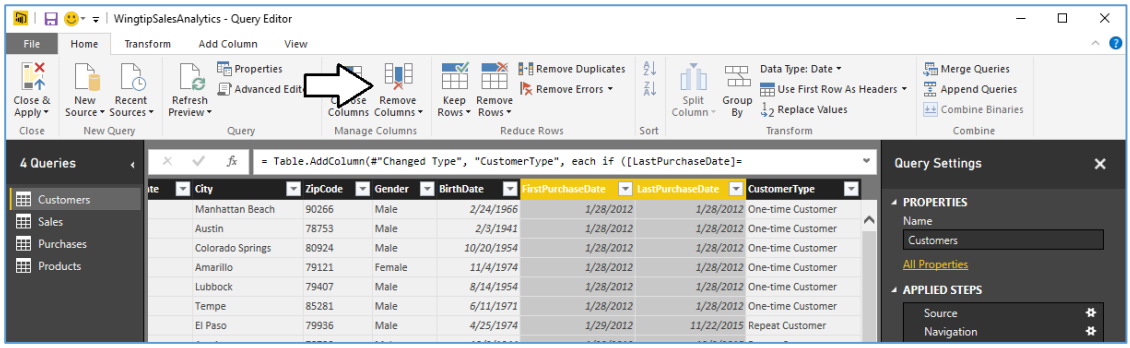
a) Activate the **Home** tab on the ribbon.

b) Select the **FirstPurchaseDate** column by clicking its column header.

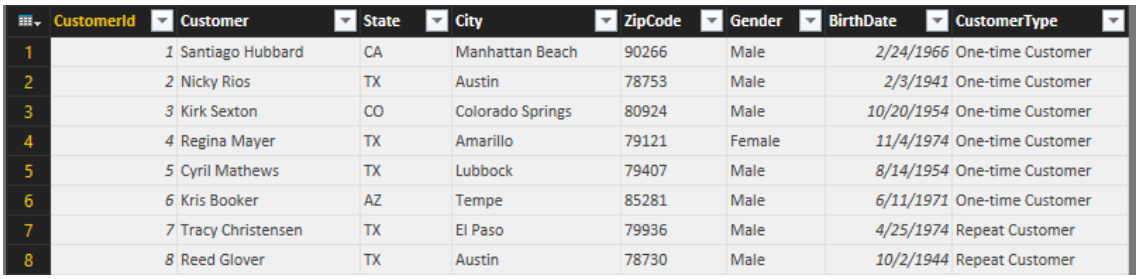
c) Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.

© Critical Path Training. 2015. All Rights Reserved

d) Click the **Remove Columns** button in the ribbon to remove both columns from the query results.



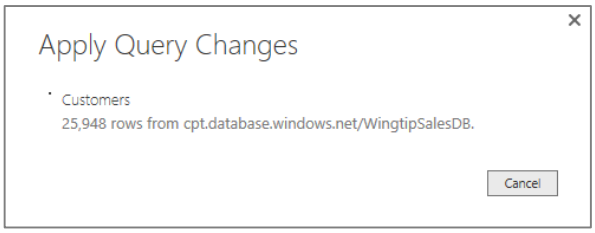

e) You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **CustomerType** column is still there.



| | CustomerId | Customer | State | City | ZipCode | Gender | BirthDate | CustomerType |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Santiago Hubbard | CA | Manhattan Beach | 90266 | Male | 2/24/1966 | One-time Customer |
| 2 | 2 | Nicky Rios | TX | Austin | 78753 | Male | 2/3/1941 | One-time Customer |
| 3 | 3 | Kirk Sexton | CO | Colorado Springs | 80924 | Male | 10/20/1954 | One-time Customer |
| 4 | 4 | Regina Mayer | TX | Amarillo | 79121 | Female | 11/4/1974 | One-time Customer |
| 5 | 5 | Cyril Mathews | TX | Lubbock | 79407 | Male | 8/14/1954 | One-time Customer |
| 6 | 6 | Kris Booker | AZ | Tempe | 85281 | Male | 6/11/1971 | One-time Customer |
| 7 | 7 | Tracy Christensen | TX | El Paso | 79936 | Male | 4/25/1974 | Repeat Customer |
| 8 | 8 | Reed Glover | TX | Austin | 78730 | Male | 10/2/1944 | Repeat Customer |

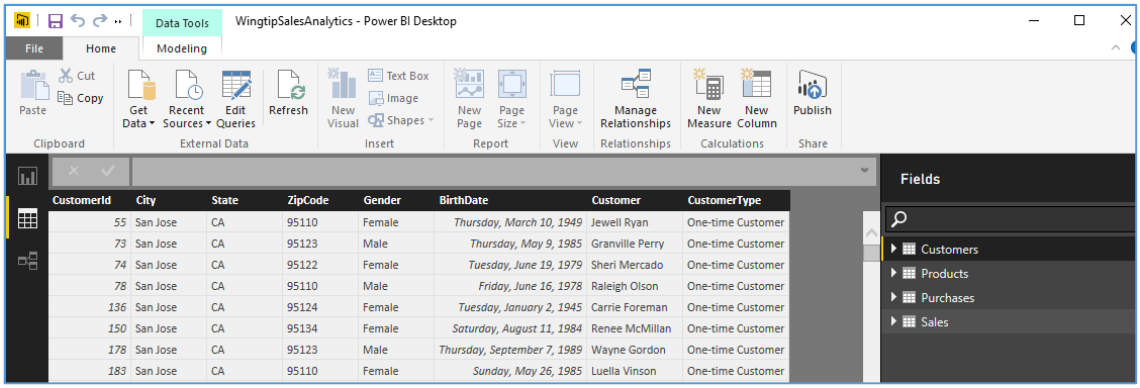
You are now done working with the **Customers** query.

12. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Customers** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data from the SQL Azure database and transforming it as it loads the customer data into the data model.




13. After the query changes have been applied, you should be able to see the results of your changes in the Customers table that has been loaded into the project's data model.

© Critical Path Training. 2015. All Rights Reserved

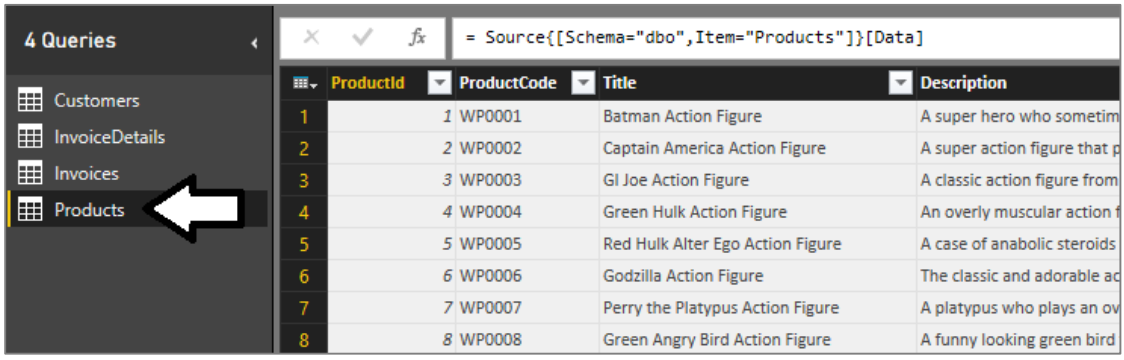



14. Save the work you have done to **WingtipSalesAnalytics.pbix** by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 3: Using Power Query to Transform and Reshape Product Data

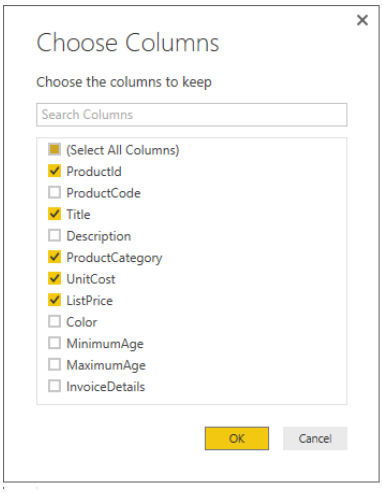
In the following exercise, you will use the **Query Editor** window to modify the **Products** query to perform transforms on product data as it is being loaded into the data model.

1. Make sure you have the **WingtipSalesAnalytics.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the **Query Editor** window.
3. Make sure the **Products** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.

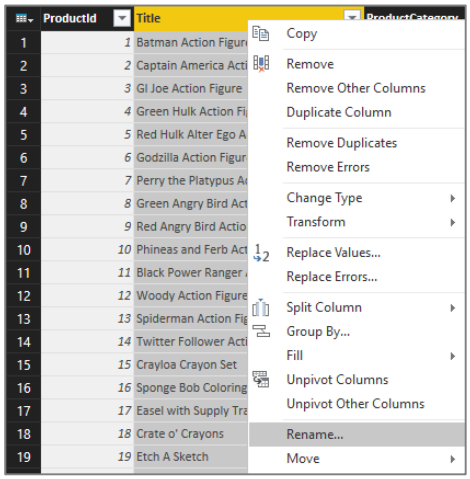



4. Remove the columns that are not required in the **Products** query results.
   a) Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
   b) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **ProductId**, **Title**, **ProductCategory**, **UnitCost** and **ListPrice** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the dialog and to modify the underlying query.

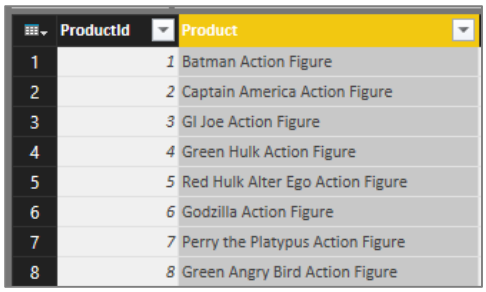
© Critical Path Training. 2015. All Rights Reserved




5. Rename the **Title** column to **Product**.
   a) Right-click on the **Title** column and click **Rename**.



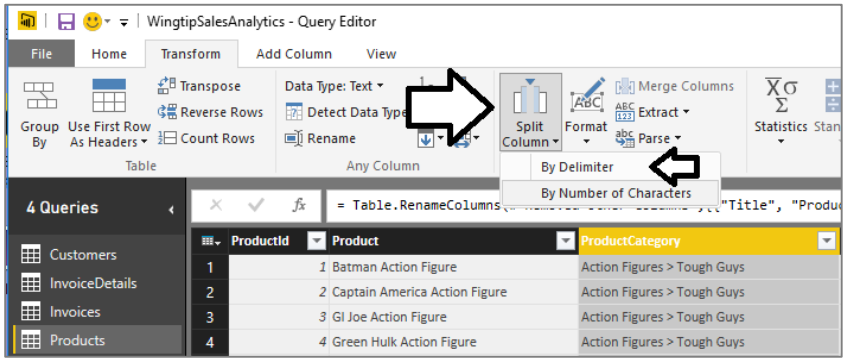

   b) Modify the column name to **Product**.



| | ProductId | Product |
|---|---|---|
| 1 | 1 | Batman Action Figure |
| 2 | 2 | Captain America Action Figure |
| 3 | 3 | GI Joe Action Figure |
| 4 | 4 | Green Hulk Action Figure |
| 5 | 5 | Red Hulk Alter Ego Action Figure |
| 6 | 6 | Godzilla Action Figure |
| 7 | 7 | Perry the Platypus Action Figure |
| 8 | 8 | Green Angry Bird Action Figure |

6. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
   a) Activate the **Transform** tab on the ribbon.
   b) Select the **ProductCategory** by clicking its column header.
   c) Drop down the **Split Column** menu button and click by **By Delimiter** to display the **Split Column By Column** dialog.

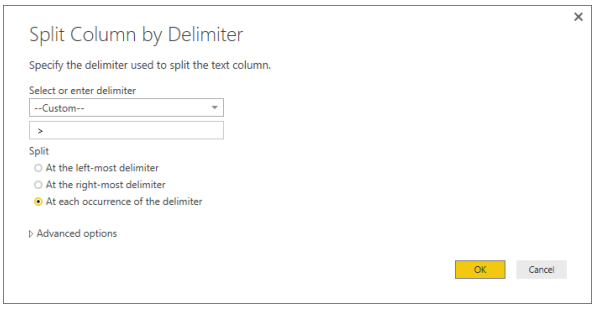
© Critical Path Training. 2015. All Rights Reserved




d) In the Split Column By Column dialog, drop down the Select or enter delimiter combo box and select --Custom--.

e) In the textbox below the combo box, enter a three-character text value which includes a space follow by the **>** character followed by another space.

f) When the **Split Column By Column** dialog appears as the one in the following screenshot, click the **OK** button to add the step to the **Products** query.




g) You should be able to confirm that Power BI Desktop has split the **ProductCategory** column into two separate columns named **ProductCategory.1** and **ProductCategory.2**.

| | ProductId | Product | ProductCategory.1 | ProductCategory.2 |
|---|---|---|---|---|
| 1 | 1 | Batman Action Figure | Action Figures | Tough Guys |
| 2 | 2 | Captain America Action Figure | Action Figures | Tough Guys |
| 3 | 3 | GI Joe Action Figure | Action Figures | Tough Guys |
| 4 | 4 | Green Hulk Action Figure | Action Figures | Tough Guys |
| 5 | 5 | Red Hulk Alter Ego Action Figure | Action Figures | Tough Guys |
| 6 | 6 | Godzilla Action Figure | Action Figures | Tough Guys |
| 7 | 7 | Perry the Platypus Action Figure | Action Figures | Cute and Huggable |
| 8 | 8 | Green Angry Bird Action Figure | Action Figures | Cute and Huggable |

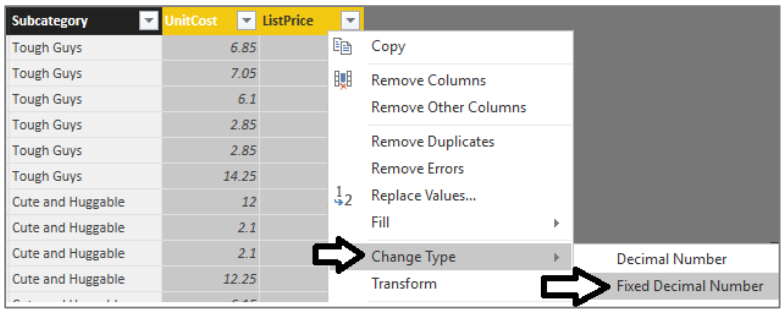
h) Rename the **ProductCategory.1** column to **Category.** Also rename the **ProductCategory.2** column to **Subcategory** as shown in the following screenshot.

| | ProductId | Product | Category | Subcategory |
|---|---|---|---|---|
| 1 | 1 | Batman Action Figure | Action Figures | Tough Guys |
| 2 | 2 | Captain America Action Figure | Action Figures | Tough Guys |
| 3 | 3 | GI Joe Action Figure | Action Figures | Tough Guys |
| 4 | 4 | Green Hulk Action Figure | Action Figures | Tough Guys |
| 5 | 5 | Red Hulk Alter Ego Action Figure | Action Figures | Tough Guys |
| 6 | 6 | Godzilla Action Figure | Action Figures | Tough Guys |
| 7 | 7 | Perry the Platypus Action Figure | Action Figures | Cute and Huggable |
| 8 | 8 | Green Angry Bird Action Figure | Action Figures | Cute and Huggable |

When you have query columns based on numeric currency values, it is best to change their column type to **Fixed Decimal Number**.

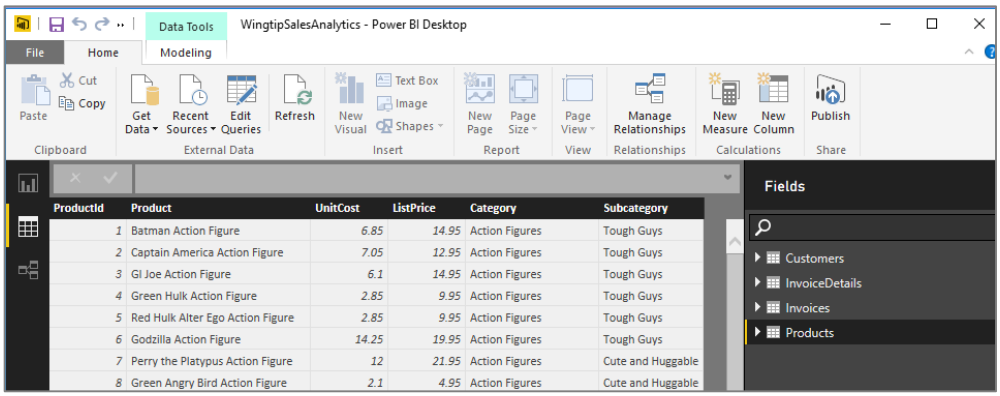
© Critical Path Training. 2015. All Rights Reserved

7. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Fixed Decimal Number** type.
   a) Select the **UnitCost** column by clicking its column header.
   b) Hold down the **SHIFT** key and click the ListPrice column so that both columns are selected.
   c) Right-click either one of the selected columns and click the **Change Type > Fixed Decimal Number** menu command.




You are now done working with the **Products** query.

8. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Products** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
9. After the query changes have been applied, you should be able to see the results of your changes in the **Products** table that has been loaded into the project's data model.




You might have noticed that the columns that are displayed in the **Products** table do not appear in the same order as they appear in the **Products** query. As of the November 2015 updates, Power BI Desktop still does not provide any ability to control and change the order of columns or fields as they are defined in a table. Microsoft is expected to address this limitation soon in a future update to the Power BI Designer.

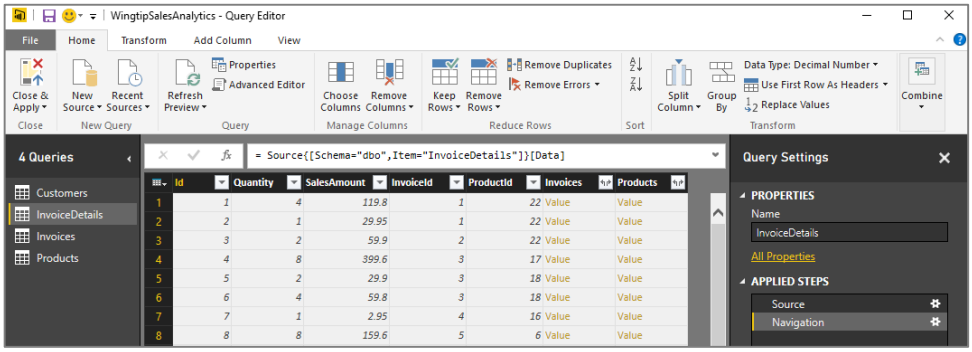
10. Save your work by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 4: Using Power Query to Transform and Reshape Product Data

In the following exercise, you will use the Query Editor to modify the **InvoiceDetails** query to transform data related to sales data before it is loaded into the data model.

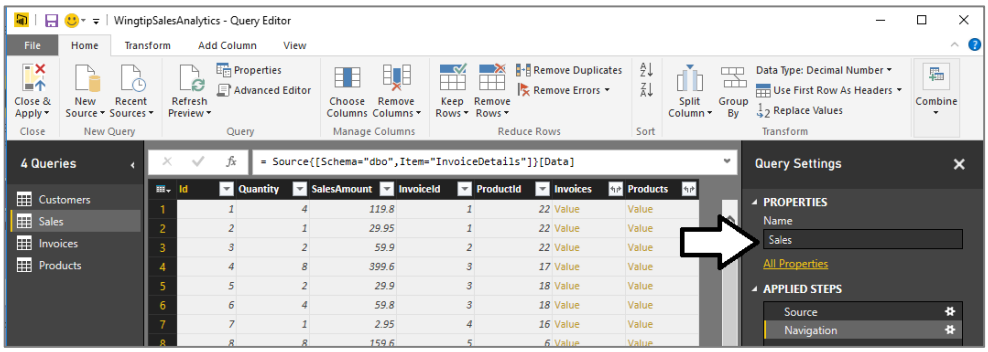
1. Make sure you have the **WingtipSalesAnalytics.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the Query Editor window.
3. Make sure the **InvoiceDetails** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.

© Critical Path Training. 2015. All Rights Reserved

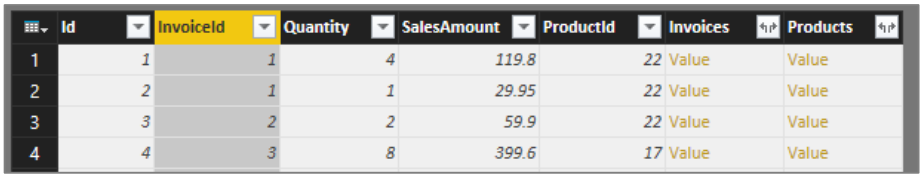



One important point of flexibility in the import process is that you can change the name of a query, and therefore the name of the resulting table to make the data model more intuitive and easier to understand. The **InvoiceDetails** query is returning data that will be used to calculate sales results at the most granular level. Therefore, the data model will be easier to understand if you change the name the name of the **InvoiceDetails** query to **Sales**. The will result in renaming the resulting table to the **Sales** table.

4. Update the name of the **InvoiceDetails** query to **Sales** by replacing the text in the **Name** textbox in the **Query Settings** pane.

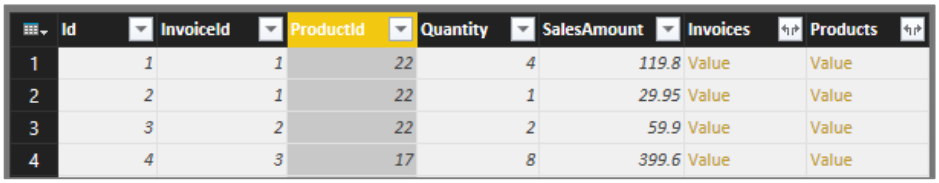



5. Using the mouse, drag and drop the column header for the **InvoiceId** column to move it to the immediate right of the **Id** column.



| | Id | InvoiceId | Quantity | SalesAmount | ProductId | Invoices | Products |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 4 | 119.8 | 22 | Value | Value |
| 2 | 2 | 1 | 1 | 29.95 | 22 | Value | Value |
| 3 | 3 | 2 | 2 | 59.9 | 22 | Value | Value |
| 4 | 4 | 3 | 8 | 399.6 | 17 | Value | Value |

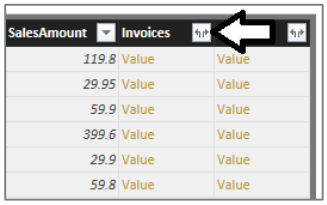
6. Using the mouse, drag and drop the column header for the **ProductId** column to move it to the right of the **InvoiceId** column.



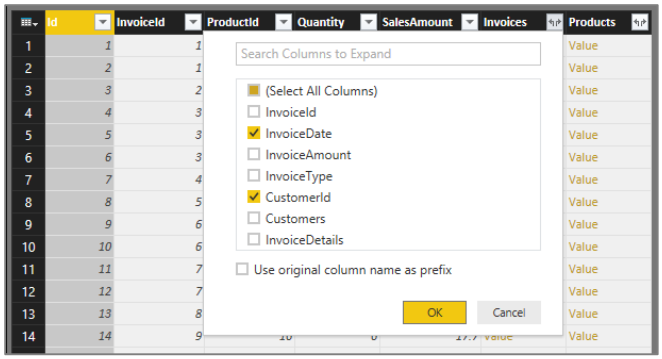
| | Id | InvoiceId | ProductId | Quantity | SalesAmount | Invoices | Products |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 22 | 4 | 119.8 | Value | Value |
| 2 | 2 | 1 | 22 | 1 | 29.95 | Value | Value |
| 3 | 3 | 2 | 22 | 2 | 59.9 | Value | Value |
| 4 | 4 | 3 | 17 | 8 | 399.6 | Value | Value |

7. Modify the column type of the **SalesAmount** column to the **Fixed Decimal Number** type.
   a) Select the **SalesAmoungt** column by clicking its column header.
   b) Right-click the selected column and click the **Change Type > Fixed Decimal Number** menu command.
8. Expand the **Invoices** column to add the **InvoiceDate** column and the **CustomerId** column to the **Sales** query.
   a) Click the Expand button inside the column header of the **Invoices** column to display the **Columns to Expand** dialog.

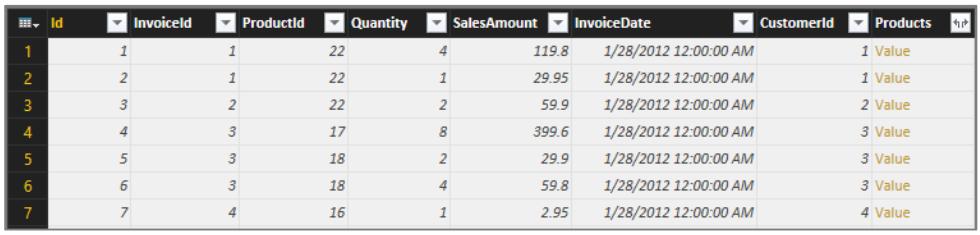
© Critical Path Training. 2015. All Rights Reserved




b) In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for the **InvoiceDate** column and the **CustomerId** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns to Expand** dialog looks like the one shown in the following screenshot, click the **OK** button to close the dialog and to modify the underlying query.

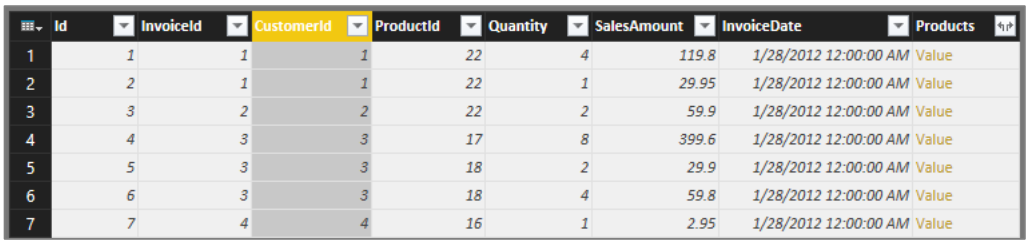



c) You should see that the **InvoiceDate** column and the **CustomerId** column have now been added to the **Sales** query results.



| | Id | InvoiceId | ProductId | Quantity | SalesAmount | InvoiceDate | CustomerId | Products |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 22 | 4 | 119.8 | 1/28/2012 12:00:00 AM | 1 | Value |
| 2 | 2 | 1 | 22 | 1 | 29.95 | 1/28/2012 12:00:00 AM | 1 | Value |
| 3 | 3 | 2 | 22 | 2 | 59.9 | 1/28/2012 12:00:00 AM | 2 | Value |
| 4 | 4 | 3 | 17 | 8 | 399.6 | 1/28/2012 12:00:00 AM | 3 | Value |
| 5 | 5 | 3 | 18 | 2 | 29.9 | 1/28/2012 12:00:00 AM | 3 | Value |
| 6 | 6 | 3 | 18 | 4 | 59.8 | 1/28/2012 12:00:00 AM | 3 | Value |
| 7 | 7 | 4 | 16 | 1 | 2.95 | 1/28/2012 12:00:00 AM | 4 | Value |

9. Use the mouse to drag and drop the **CustomerId** column to move it to the right of the **InvoiceId** column.



| | Id | InvoiceId | CustomerId | ProductId | Quantity | SalesAmount | InvoiceDate | Products |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 4 | 119.8 | 1/28/2012 12:00:00 AM | Value |
| 2 | 2 | 1 | 1 | 22 | 1 | 29.95 | 1/28/2012 12:00:00 AM | Value |
| 3 | 3 | 2 | 2 | 22 | 2 | 59.9 | 1/28/2012 12:00:00 AM | Value |
| 4 | 4 | 3 | 3 | 17 | 8 | 399.6 | 1/28/2012 12:00:00 AM | Value |
| 5 | 5 | 3 | 3 | 18 | 2 | 29.9 | 1/28/2012 12:00:00 AM | Value |
| 6 | 6 | 3 | 3 | 18 | 4 | 59.8 | 1/28/2012 12:00:00 AM | Value |
| 7 | 7 | 4 | 4 | 16 | 1 | 2.95 | 1/28/2012 12:00:00 AM | Value |

10. Use the mouse to drag and drop the **InvoiceDate** column to move it to the right of the **ProductId** column.

| | Id | InvoiceId | CustomerId | ProductId | InvoiceDate | Quantity | SalesAmount | Products |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 1/28/2012 12:00:00 AM | 4 | 119.8 | Value |
| 2 | 2 | 1 | 1 | 22 | 1/28/2012 12:00:00 AM | 1 | 29.95 | Value |
| 3 | 3 | 2 | 2 | 22 | 1/28/2012 12:00:00 AM | 2 | 59.9 | Value |
| 4 | 4 | 3 | 3 | 17 | 1/28/2012 12:00:00 AM | 8 | 399.6 | Value |
| 5 | 5 | 3 | 3 | 18 | 1/28/2012 12:00:00 AM | 2 | 29.9 | Value |
| 6 | 6 | 3 | 3 | 18 | 1/28/2012 12:00:00 AM | 4 | 59.8 | Value |
| 7 | 7 | 4 | 4 | 16 | 1/28/2012 12:00:00 AM | 1 | 2.95 | Value |

© Critical Path Training. 2015. All Rights Reserved

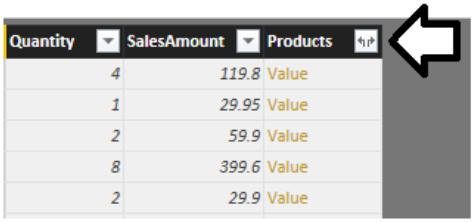
11. Change the column type of the **InvoiceDate** to the **Date** type by right-clicking its column header and selecting the **Change Type > Date** command. When you are done, the data in the column should show a date value without a time.

| | Id | InvoiceId | CustomerId | ProductId | InvoiceDate | Quantity | SalesAmount | Products |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 1/28/2012 | 4 | 119.8 | Value |
| 2 | 2 | 1 | 1 | 22 | 1/28/2012 | 1 | 29.95 | Value |
| 3 | 3 | 2 | 2 | 22 | 1/28/2012 | 2 | 59.9 | Value |
| 4 | 4 | 3 | 3 | 17 | 1/28/2012 | 8 | 399.6 | Value |
| 5 | 5 | 3 | 3 | 18 | 1/28/2012 | 2 | 29.9 | Value |
| 6 | 6 | 3 | 3 | 18 | 1/28/2012 | 4 | 59.8 | Value |
| 7 | 7 | 4 | 4 | 16 | 1/28/2012 | 1 | 2.95 | Value |

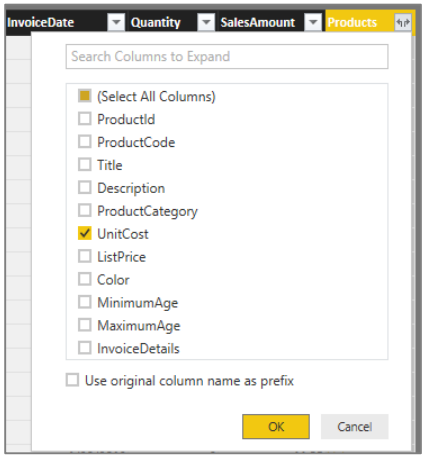
12. Change the name of the **InvoiceDate** column to **PurchaseDate**.

| | Id | InvoiceId | CustomerId | ProductId | PurchaseDate | Quantity | SalesAmount | Products |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 1/28/2012 | 4 | 119.8 | Value |
| 2 | 2 | 1 | 1 | 22 | 1/28/2012 | 1 | 29.95 | Value |
| 3 | 3 | 2 | 2 | 22 | 1/28/2012 | 2 | 59.9 | Value |
| 4 | 4 | 3 | 3 | 17 | 1/28/2012 | 8 | 399.6 | Value |
| 5 | 5 | 3 | 3 | 18 | 1/28/2012 | 2 | 29.9 | Value |
| 6 | 6 | 3 | 3 | 18 | 1/28/2012 | 4 | 59.8 | Value |

13. Expand the **Products** column to add the **UnitCost** column to the **Sales** query.
    a) Click the Expand button inside the column header of the **Products** column to display the **Columns to Expand** dialog.




    b) In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkbox for the **UnitCost** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns to Expand** dialog looks like the one shown in the following screenshot, click the **OK** button to close the dialog and to modify the underlying query.

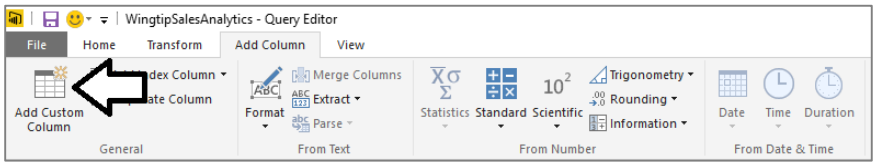



    c) You should see that the **UnitCost** column has now been added to the **Sales** query results.

© Critical Path Training. 2015. All Rights Reserved

| | Id | InvoiceId | CustomerId | ProductId | PurchaseDate | Quantity | SalesAmount | UnitCost |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 1/28/2012 | 4 | 119.8 | 21.5 |
| 2 | 2 | 1 | 1 | 22 | 1/28/2012 | 1 | 29.95 | 21.5 |
| 3 | 3 | 2 | 2 | 22 | 1/28/2012 | 2 | 59.9 | 21.5 |
| 4 | 4 | 3 | 3 | 17 | 1/28/2012 | 8 | 399.6 | 12.1 |
| 5 | 5 | 3 | 3 | 18 | 1/28/2012 | 2 | 29.9 | 10.5 |
| 6 | 6 | 3 | 3 | 18 | 1/28/2012 | 4 | 59.8 | 10.5 |

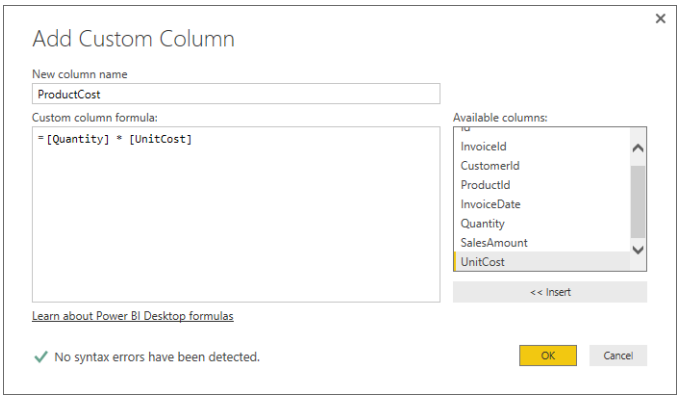
14. Add a new custom column named **ProductCost** to calculate the product of the **Quantity** field multiplied by the **UnitCost** field.
    a) Begin by activating the **Add Column** tab in the ribbon.
    b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.




    c) In the **Add Custom Column** dialog, add a value of **ProductCost** in the **New column name** textbox.
    d) In the **Custom column formula** textbox, enter the following formula.

```
[Quantity] * [UnitCost]
```

    e) When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.




    f) You should be able to verify that the new **ProductCost** column has a value calculated by multiplying the value of the **Quantity** column together with the value of the **UnitCost** column.

| | Id | InvoiceId | CustomerId | ProductId | PurchaseDate | Quantity | SalesAmount | UnitCost | ProductCost |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 1/28/2012 | 4 | 119.8 | 21.5 | 86 |
| 2 | 2 | 1 | 1 | 22 | 1/28/2012 | 1 | 29.95 | 21.5 | 21.5 |
| 3 | 3 | 2 | 2 | 22 | 1/28/2012 | 2 | 59.9 | 21.5 | 43 |
| 4 | 4 | 3 | 3 | 17 | 1/28/2012 | 8 | 399.6 | 12.1 | 96.8 |
| 5 | 5 | 3 | 3 | 18 | 1/28/2012 | 2 | 29.9 | 10.5 | 21 |
| 6 | 6 | 3 | 3 | 18 | 1/28/2012 | 4 | 59.8 | 10.5 | 42 |

Once the **UnitCost** column has been used to calculate the value for the **ProductCost** value, this column is no longer needed and can be removed from the results of the **Sales** query.

15. Remove the **UnitCost** column by selecting its column header and pressing the DELETE key on the keyboard.

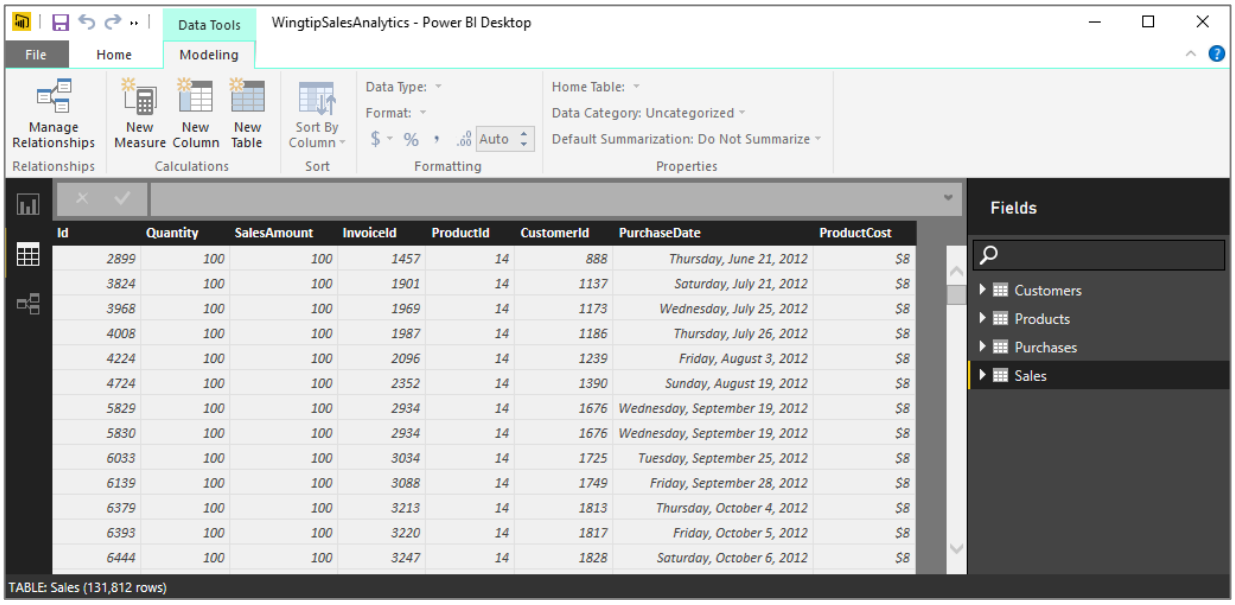
© Critical Path Training. 2015. All Rights Reserved

| | Id | InvoiceId | CustomerId | ProductId | PurchaseDate | Quantity | SalesAmount | ProductCost |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 22 | 1/28/2012 | 4 | 119.8 | 86 |
| 2 | 2 | 1 | 1 | 22 | 1/28/2012 | 1 | 29.95 | 21.5 |
| 3 | 3 | 2 | 2 | 22 | 1/28/2012 | 2 | 59.9 | 43 |
| 4 | 4 | 3 | 3 | 17 | 1/28/2012 | 8 | 399.6 | 96.8 |
| 5 | 5 | 3 | 3 | 18 | 1/28/2012 | 2 | 29.9 | 21 |
| 6 | 6 | 3 | 3 | 18 | 1/28/2012 | 4 | 59.8 | 42 |

16. Modify the column type of the **SalesAmount** column and the **ProductCost** column to the **Fixed Decimal Number** type.
    a) Select the **SalesAmount** column by clicking its column header.
    b) Hold down the **SHIFT** key and click the **ProductCost** column so that both columns are selected.
    c) Right-click either one of the selected columns and click the **Change Type > Fixed Decimal Number** menu command.

You are now done working with the **Sales** query.

17. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Sales** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
18. After the query changes have been applied, you should be able to see the results of your changes in the **Sales** table that has been loaded into the project's data model.




| Id | Quantity | SalesAmount | InvoiceId | ProductId | CustomerId | PurchaseDate | ProductCost |
|---|---|---|---|---|---|---|---|
| 2899 | 100 | 100 | 1457 | 14 | 888 | Thursday, June 21, 2012 | $8 |
| 3824 | 100 | 100 | 1901 | 14 | 1137 | Saturday, July 21, 2012 | $8 |
| 3968 | 100 | 100 | 1969 | 14 | 1173 | Wednesday, July 25, 2012 | $8 |
| 4008 | 100 | 100 | 1987 | 14 | 1186 | Thursday, July 26, 2012 | $8 |
| 4224 | 100 | 100 | 2096 | 14 | 1239 | Friday, August 3, 2012 | $8 |
| 4724 | 100 | 100 | 2352 | 14 | 1390 | Sunday, August 19, 2012 | $8 |
| 5829 | 100 | 100 | 2934 | 14 | 1676 | Wednesday, September 19, 2012 | $8 |
| 5830 | 100 | 100 | 2934 | 14 | 1676 | Wednesday, September 19, 2012 | $8 |
| 6033 | 100 | 100 | 3034 | 14 | 1725 | Tuesday, September 25, 2012 | $8 |
| 6139 | 100 | 100 | 3088 | 14 | 1749 | Friday, September 28, 2012 | $8 |
| 6379 | 100 | 100 | 3213 | 14 | 1813 | Thursday, October 4, 2012 | $8 |
| 6393 | 100 | 100 | 3220 | 14 | 1817 | Friday, October 5, 2012 | $8 |
| 6444 | 100 | 100 | 3247 | 14 | 1828 | Saturday, October 6, 2012 | $8 |

Unfortunately, the current version of Power BI Desktop does not retain the order of the columns that you configured in the **Sales** query.

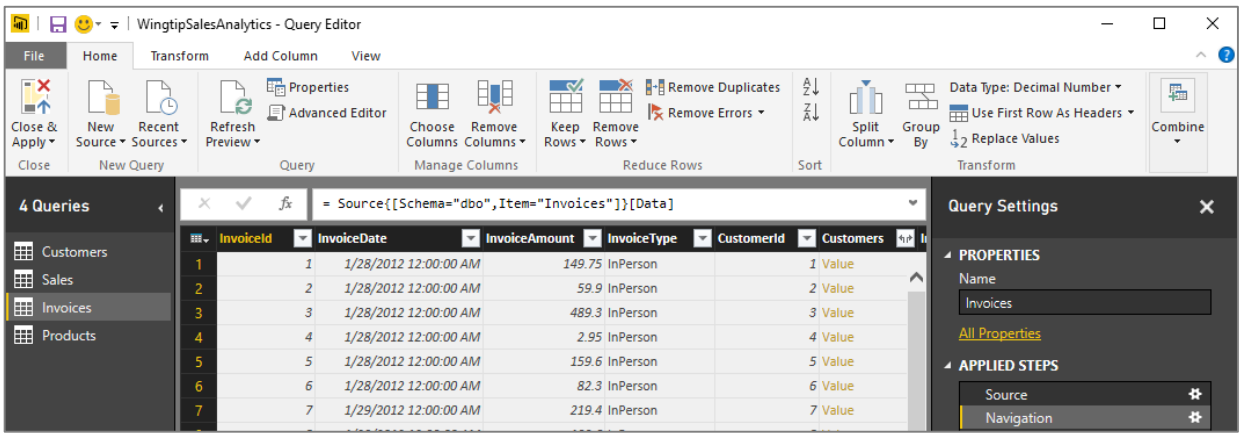
19. Save your work by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 5: Using Power Query to Transform and Reshape Purchase Data

In the following exercise, you will use the **Query Editor** window to modify the **Invoices** query to transform invoice data as it is being loaded into the data model.

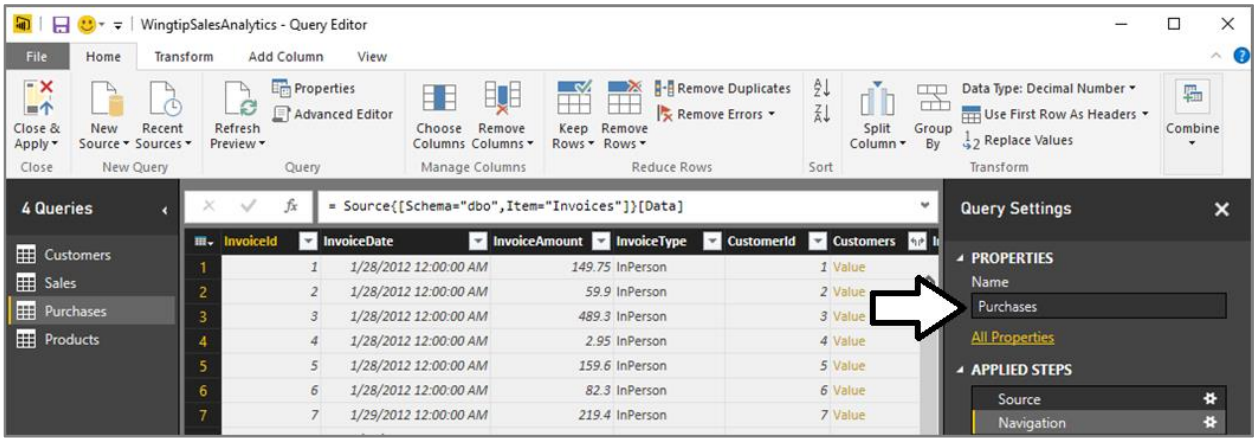
1. Make sure you have the **WingtipSalesAnalytics.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the **Query Editor** window.
3. Make sure the **Invoices** query is selected in the **Queries** list on the left-hand side of the Query Editor window.

© Critical Path Training. 2015. All Rights Reserved



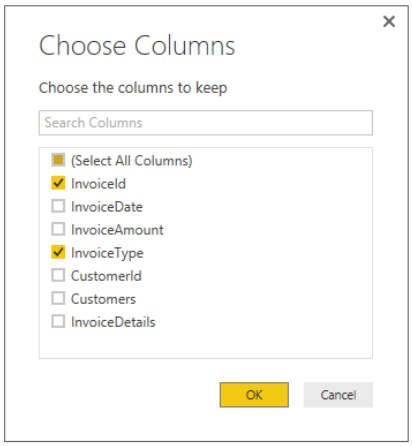

In the last exercise you changed the name of the **InvoiceDetails** to **Sales** to make the data model easier to understand. In this exercise you will change the name of the **Invoices** query to **Purchases** for the same reason.

4. Update the name of the **Invoices** query to **Purchases** by replacing the text in the **Name** textbox in the **Query Settings** pane.




5. Make sure the **Purchases** query is selected in the **Queries** list on the left-hand side of the Query Editor window. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
6. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **InvoiceId** and **InvoiceType** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.

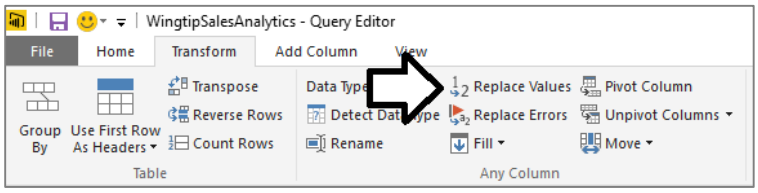
© Critical Path Training. 2015. All Rights Reserved



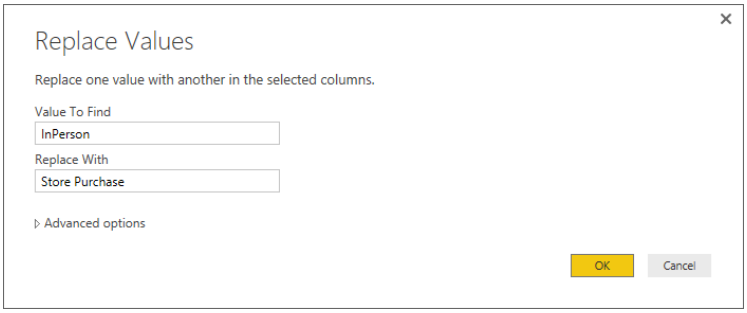

7. You should be able to see that the Query Editor window now only shows the columns that you selected.

| | InvoiceId | InvoiceType |
|---|---|---|
| 1 | 1 | InPerson |
| 2 | 2 | InPerson |
| 3 | 3 | InPerson |
| 4 | 4 | InPerson |
| 5 | 5 | InPerson |
| 6 | 6 | InPerson |

8. Modify the query so that the **InvoiceType** column returns values that are more human readable.
   a) Make sure the **Transform** tab is the active tab in the ribbon.
   b) Select the **InvoiceType** column by clicking its column header.
   c) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.



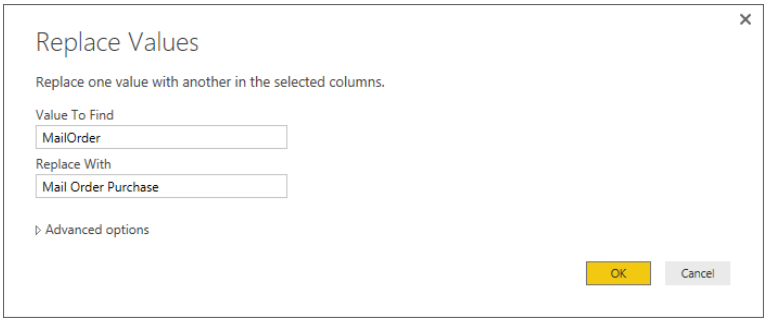

   d) In the **Replace Value** dialog, enter a value of **InPerson** in the **Value to Find** textbox and enter a value of **Store Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.




   e) Make sure the **InvoiceType** column is still selected.
   f) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.

© Critical Path Training. 2015. All Rights Reserved

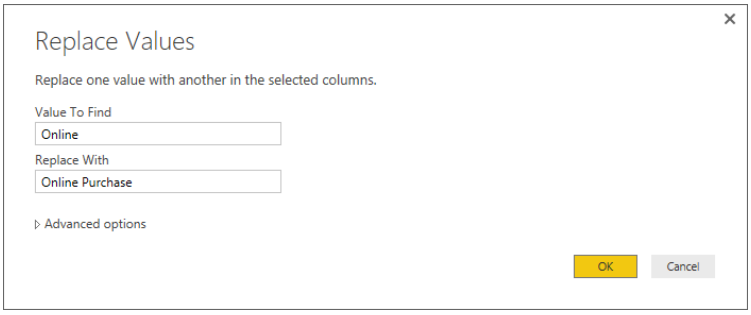
g) In the **Replace Value** dialog, enter a value of **MailOrder** in the **Value to Find** textbox and enter a value of **Mail Order Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.




h) Make sure the **InvoiceType** column is still selected.

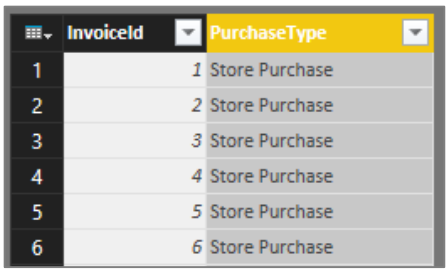
i) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.

j) In the **Replace Value** dialog, enter a value of **Online** in the **Value to Find** textbox and enter a value of **Online Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.




k) If you scroll down and look at all the rows within the **Purchases** table, you should be able to see that each row has a **InvoiceType** column value of either **Store Purchase**, **Mail Order Purchase** or **Online Purchase**.

9. Change the name of the **InvoiceType** column to **PurchaseType**.



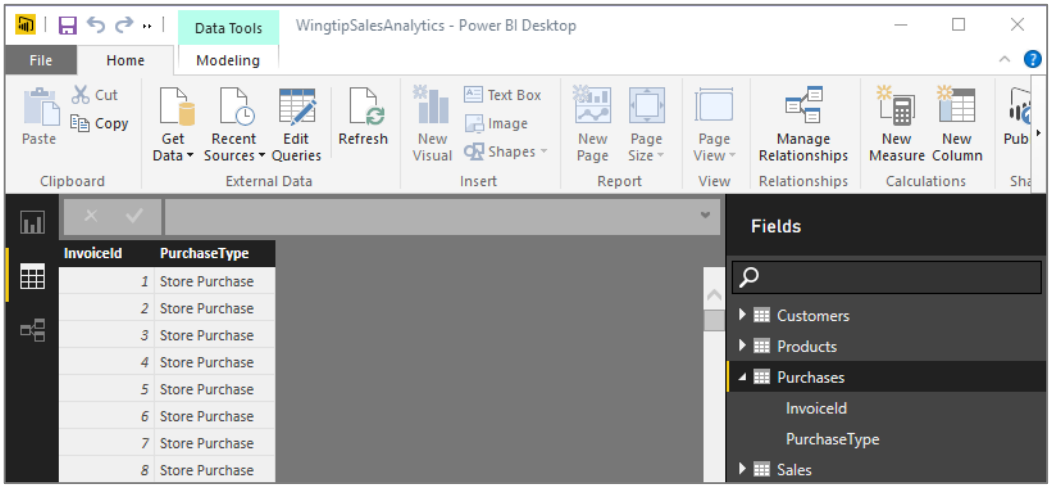
| | InvoiceId | PurchaseType |
|---|---|---|
| 1 | 1 | Store Purchase |
| 2 | 2 | Store Purchase |
| 3 | 3 | Store Purchase |
| 4 | 4 | Store Purchase |
| 5 | 5 | Store Purchase |
| 6 | 6 | Store Purchase |

You are now done working with the **Purchase** query.

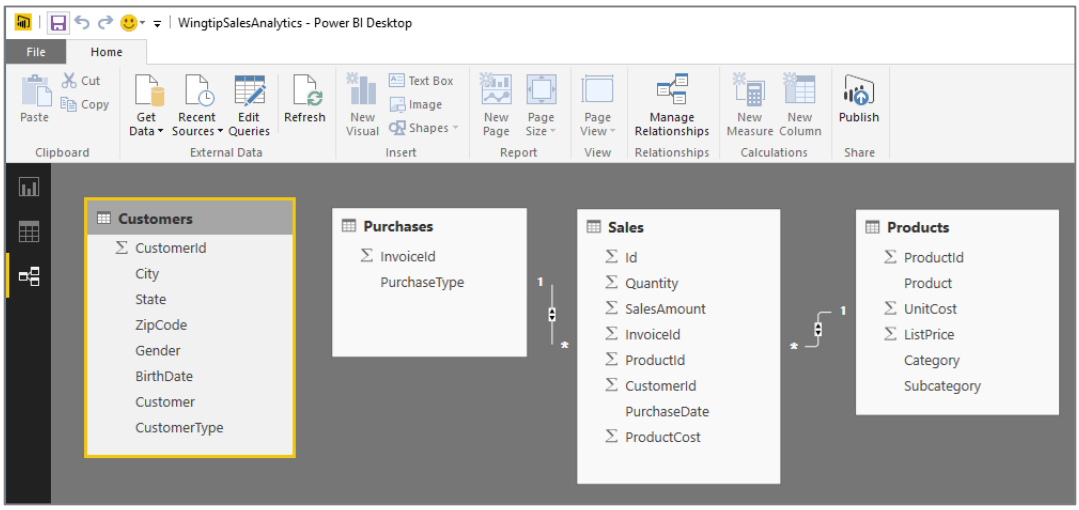
10. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Purchases** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.

11. After the query changes have been applied, you should be able to see the results of your changes in the **Purchase** table that has been loaded into the project's data model.

© Critical Path Training. 2015. All Rights Reserved

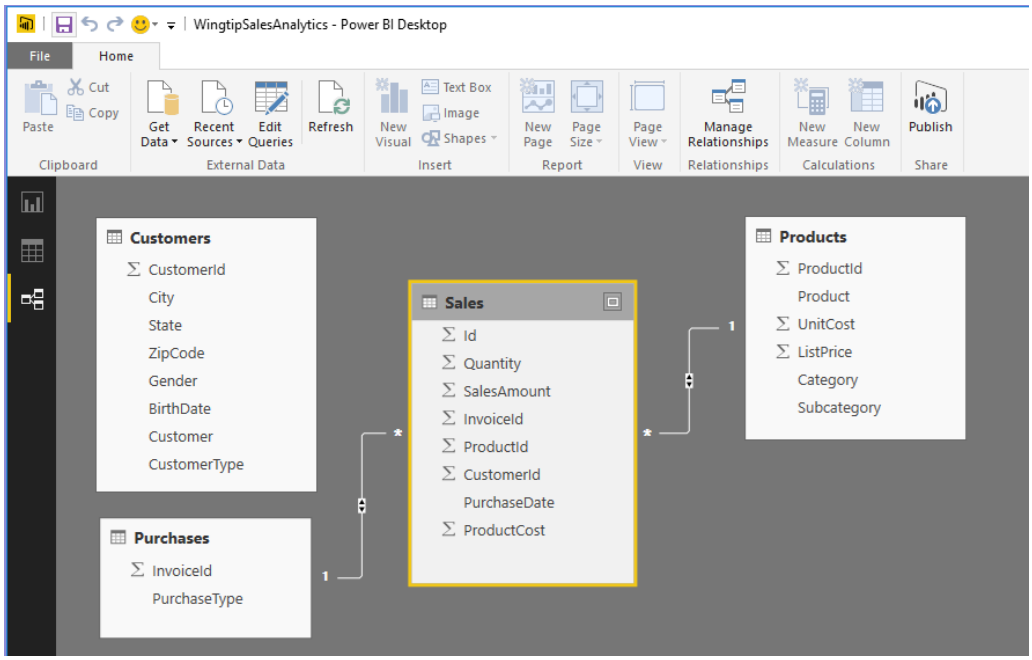



12. Navigate to relationship view to see how your updates to the queries in this project have affected the relationships between the tables. You should be able to see that there is no longer a relationship between the **Customers** table and the **Purchases** table. This is due to the fact that you removed the **CustomerId** column from the **Purchases** table.

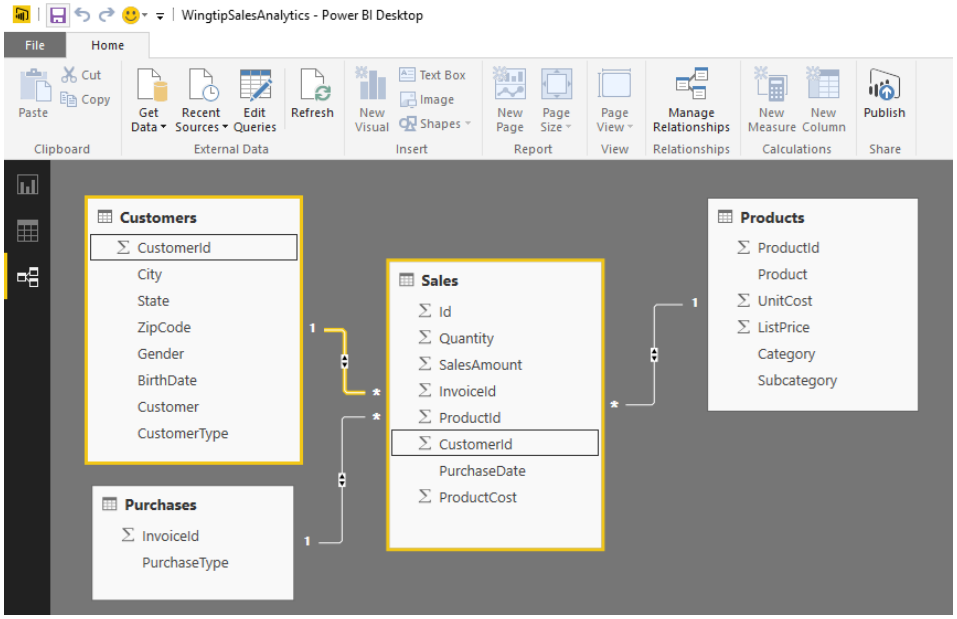



13. Use your mouse to move and resize the four tables in the data model to match the following screenshot.

© Critical Path Training. 2015. All Rights Reserved




14. Create a new relationship between the **Sales** table and the **Customers** table by clicking on the **CustomerId** column in the **Sales** table and then dragging and dropping it on the **CustomerId** column of the **Customers** table. Once you have successfully completed this step, you should see a visual line between these two tables to indicate that the data model now contains the new relationship.




The tables in the data model have now been refactored into a "star schema" that is commonly used in data modeling for BI projects.

15. Save your work by clicking the Save button in the upper left corner of the Power BI Desktop window.

You are now done with your work modifying queries to transform data as it is being loaded into the data model of your project. Note that you will continue to work on the PBIX file with this project in the next lab and the ones that follow.

© Critical Path Training. 2015. All Rights Reserved

# Module 04: Using the Power Pivot Features of Power BI Desktop

## Module Description

This module introduces students to the PowerPivot features and data modeling tools available when working with Power BI Desktop. Students will learn about the various ways that Power BI Desktop can be used to model data for richer analytics and reporting. The module discusses how to get up and running with the DAX programming language as well as how to extend tables in a dataset with Calculated Columns and Measures.

## Module Agenda

- Data Modeling with PowerPivot and DAX
- Creating Calculated Columns
- Integrating Lookup Tables
- Creating Measures

## Topics Covered

Building a Data Model with Power Pivot
Table Relationships
Relationship Properties
Converting Column Types
Formatting Columns
Working with DAX
Writing DAX Expressions
Calculated Columns vs Measures
Types of DAX Functions
When to Create Calculated Columns
Creating Calculated Columns
Calculated Column as a Column Label
Calculated Column for Customer Age Group
Calculated Column as a Row Label
Calculated Column used in a Slicer
Adding Lookup Tables to the Data Model
Importing the SalesRegions Table from Excel
Integrating the Lookup Table into the Data Model
The RELATED Function
Hiding the Lookup Table
Filtering on Sales Region
Benefits of Measures over Calculated Columns
Creating Measures
Formatting Measures

## Instructor Demos

- Creating Calculated Columns
- Creating Measures

© Critical Path Training. 2015. All Rights Reserved

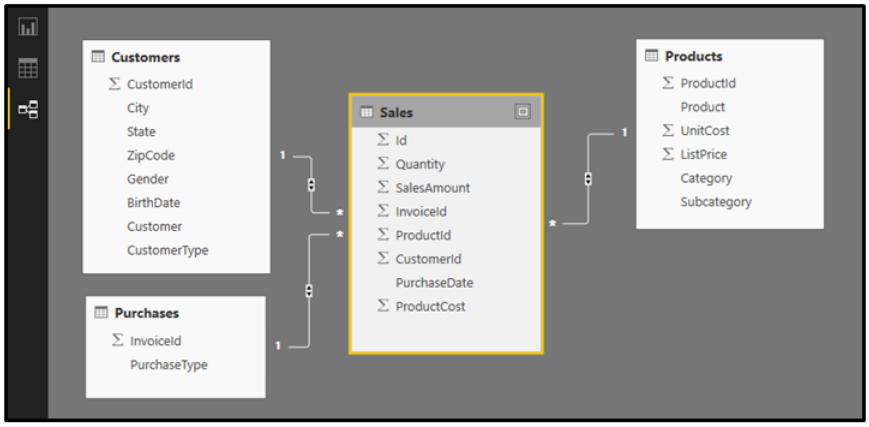
## Building a Data Model with Power Pivot

- Steps to create a data model with Power Pivot
  - Create relationships between tables
  - Modify columns (rename, set formatting, convert type)
  - Create calculated columns
  - Create measures
- Features not yet supported by Power BI Desktop
  - Creating dimensional hierarchies
  - Creating key performance indicators (KPIs)

© Critical Path Training. 2015. All Rights Reserved

# Table Relationships

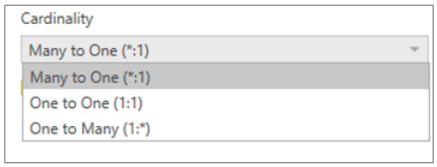
- Tables in data model associated with relationships
  - Relationships based on single columns
  - Tabular model supports [1-to-1] and [1-to-many] relationships
  - Relationships based on single column in each table



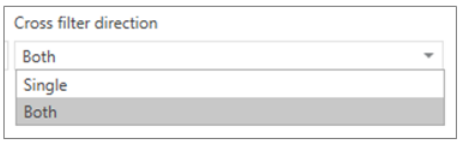

© Critical Path Training. 2015. All Rights Reserved

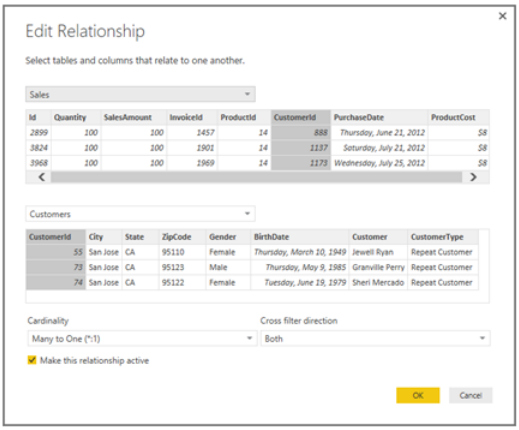
# Relationship Properties

- Cardinality



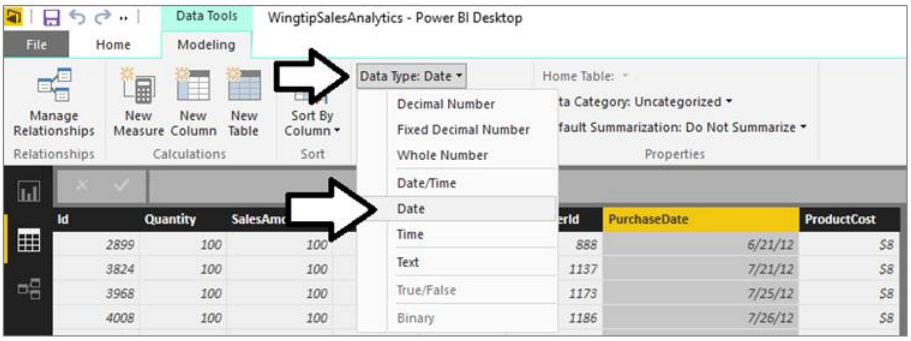

- Cross filter direction



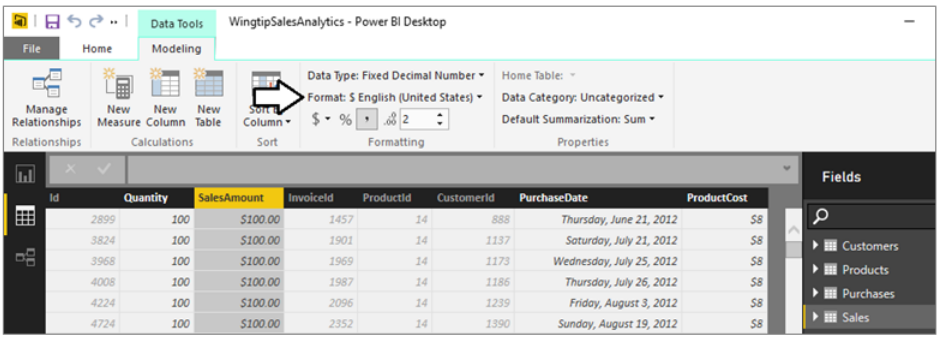
# Converting Column Types

- Power Pivot allows you to convert columns
  - Alternative to converting column with Power Query



# Formatting Columns

- Each column has its own formatting properties
  - Formatting propagated to reports and visuals
  - Makes it easier on data model consumers



## Working with DAX

- DAX is the language used to create data model
  - DAX stands for "**D**ata **A**nalysis E**x**pression Language"
- DAX expressions are similar to Excel formulas
  - They always start with an equal sign (=)
  - DAX provides many built-in functions similar to Excel
- DAX Expressions are unlike Excel formulas…
  - DAX expressions cannot reference cells (e.g. A1 or C4)
  - Instead DAX expressions reference columns and tables

```
=SUM('Sales'[SalesAmount])
```

© Critical Path Training. 2015. All Rights Reserved

# Writing DAX Expressions

- Some DAX expressions are simple

```
Sales Revenue = Sum(Sales[SalesAmount])
```

- Some DAX expressions are far more complex

```
Sales Growth PM = IF(
  ( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() ),
  DIVIDE(
    SUM(Sales[SalesAmount]) -
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    ),
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    )
  ),
  BLANK()
)
```

© Critical Path Training. 2015. All Rights Reserved

## Calculated Columns vs Measures

- Calculated Columns
  - Evaluated based on context of a single row
  - Evaluated when data is loaded into memory

```
Column1 = <DAX expression>
```

- Measures
  - Evaluated at query time based on current filter context
  - Commonly used for aggregations (e.g. SUM, AVG, etc.)
  - Used more frequently than calculated columns

```
Measure1 = <DAX expression>
```

© Critical Path Training. 2015. All Rights Reserved

## Types of DAX Functions

- Date and Time Functions
- Information Functions
- Logical Functions
- Mathematical and Trigonometric Functions
- Statistical Functions
- Filter Functions
- Text Functions
- Time Intelligence Functions

© Critical Path Training. 2015. All Rights Reserved

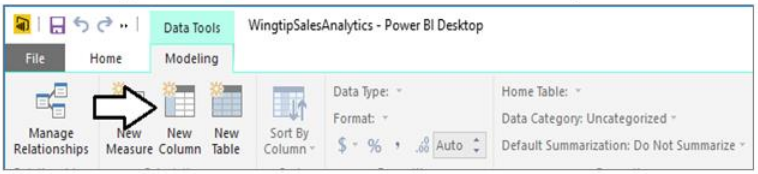
## When to Create Calculated Columns

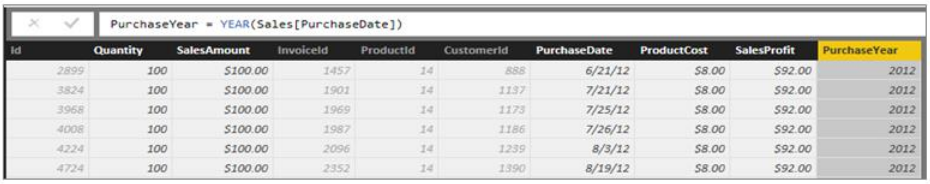
- Measures often better choice than calculate columns
  - Don't create calculated column when you need a measure
  - Prefer to create calculated columns only in specific scenarios
- When should you create calculated columns?
  - To create headers for row labels or column labels
  - To place calculated results in a slicer for filtering
  - Define an expression strictly bound to current row
  - Categories text or numbers (e.g. customer age groups)

© Critical Path Training. 2015. All Rights Reserved

# Creating Calculated Columns

- Edited in formula bar of Power Pivot data view
  - Start with name and then equals (=) sign
  - Enter a valid DAX expression
  - Clicking on column adds it into expression



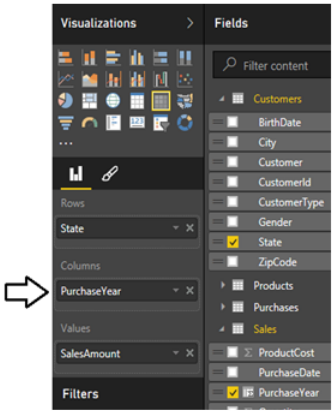




| Id | Quantity | SalesAmount | InvoiceId | ProductId | CustomerId | PurchaseDate | ProductCost | SalesProfit | PurchaseYear |
|---|---|---|---|---|---|---|---|---|---|
| 2899 | 100 | $100.00 | 1457 | 14 | 888 | 6/21/12 | $8.00 | $92.00 | 2012 |
| 3824 | 100 | $100.00 | 1901 | 14 | 1137 | 7/21/12 | $8.00 | $92.00 | 2012 |
| 3968 | 100 | $100.00 | 1969 | 14 | 1173 | 7/25/12 | $8.00 | $92.00 | 2012 |
| 4008 | 100 | $100.00 | 1987 | 14 | 1186 | 7/26/12 | $8.00 | $92.00 | 2012 |
| 4224 | 100 | $100.00 | 2096 | 14 | 1239 | 8/3/12 | $8.00 | $92.00 | 2012 |
| 4724 | 100 | $100.00 | 2352 | 14 | 1390 | 8/19/12 | $8.00 | $92.00 | 2012 |

© Critical Path Training. 2015. All Rights Reserved

# Calculated Column as a Column Label

- Calculate column can serve as...
  - Row labels
  - Column labels




| State | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| CA | $270,926.32 | $550,160.02 | $737,878.53 | $770,402.11 | $2,329,366.98 |
| TX | $212,085.08 | $490,643.98 | $683,079.11 | $919,030.36 | $2,304,838.53 |
| FL | $51,730.85 | $300,866.87 | $535,693.94 | $891,344.92 | $1,779,636.58 |
| NC | $11,018.02 | $164,804.24 | $315,139.92 | $448,638.72 | $939,600.90 |
| NY | $24,207.43 | $165,046.23 | $256,294.27 | $430,971.24 | $876,519.17 |
| GA | $40,305.80 | $152,807.51 | $239,451.05 | $417,037.28 | $849,601.64 |

© Critical Path Training. 2015. All Rights Reserved

# Calculated Column for Customer Age Group

1. Calculate customer age from birthdate

```
Age = Floor( (TODAY()-Customers[BirthDate])/365, 1)
```

| CustomerId | City | State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age |
|---|---|---|---|---|---|---|---|---|
| 55 | San Jose | CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 |
| 73 | San Jose | CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 |
| 74 | San Jose | CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 |
| 78 | San Jose | CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 |
| 136 | San Jose | CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 |
| 150 | San Jose | CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 |

2. Calculate age groups using calculated column

```
Age Group =
  SWITCH( TRUE(),
    [Age] >= 65, "Ages 65 and over",
    [Age] >= 50, "Ages 50 TO 65",
    [Age] >= 40, "Ages 40 TO 49",
    [Age] >= 30, "Ages 30 TO 39",
    [Age] >= 18, "Ages 18 TO 23",
    [Age] < 18, "Ages under 18"
  )
```

| CustomerId | City | State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age | Age Group |
|---|---|---|---|---|---|---|---|---|---|
| 55 | San Jose | CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 | Ages 65 and over |
| 73 | San Jose | CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 | Ages 30 TO 39 |
| 74 | San Jose | CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 | Ages 30 TO 39 |
| 78 | San Jose | CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 | Ages 30 TO 39 |
| 136 | San Jose | CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 | Ages 65 and over |
| 150 | San Jose | CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 | Ages 30 TO 39 |
| 178 | San Jose | CA | 95123 | Male | 9/7/89 | Wayne Gordon | Repeat Customer | 26 | Ages 18 TO 23 |
| 183 | San Jose | CA | 95110 | Female | 5/26/85 | Luella Vinson | Repeat Customer | 30 | Ages 30 TO 39 |
| 213 | San Jose | CA | 95123 | Male | 8/13/78 | Rosario Knight | Repeat Customer | 37 | Ages 30 TO 39 |

© Critical Path Training. 2015. All Rights Reserved

# Calculated Column as a Row Label

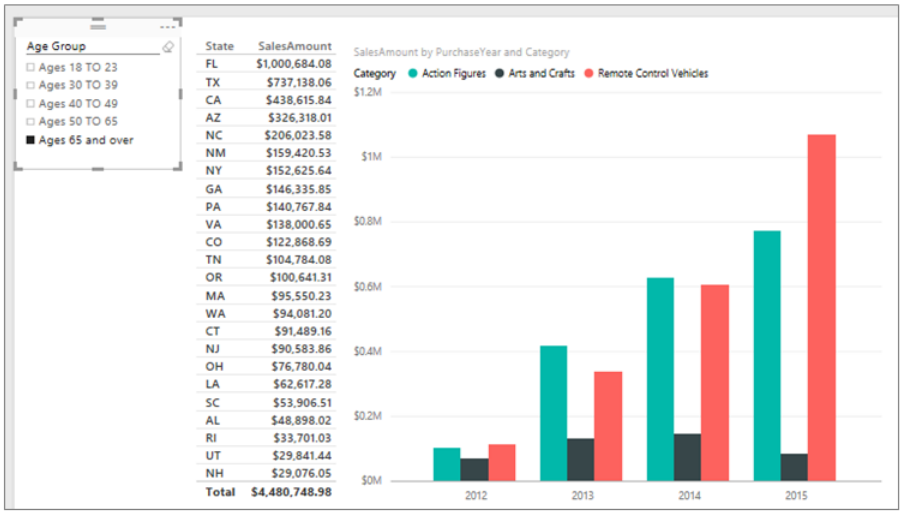
- Age Group can now be used as row label

| Age Group | Count of CustomerId | Quantity | SalesAmount |
|---|---|---|---|
| Ages 50 TO 65 | 7974 | 1,143,644 | $4,904,342.59 |
| Ages 65 and over | 7569 | 1,050,879 | $4,480,748.98 |
| Ages 40 TO 49 | 5103 | 713,453 | $3,083,964.23 |
| Ages 30 TO 39 | 4547 | 646,121 | $2,798,053.19 |
| Ages 18 TO 23 | 3278 | 478,405 | $1,985,991.24 |
| **Total** | **28471** | **4,032,502** | **$17,253,100.23** |

© Critical Path Training. 2015. All Rights Reserved

# Calculated Column used in a Slicer

- Calculated column can populate slicer values

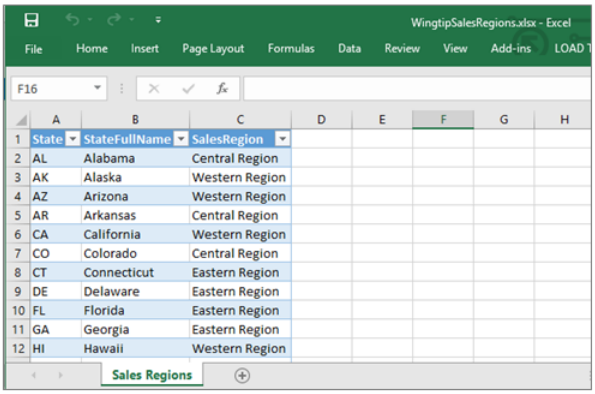



| State | SalesAmount |
|---|---|
| FL | $1,000,684.08 |
| TX | $737,138.06 |
| CA | $438,615.84 |
| AZ | $326,318.01 |
| NC | $206,023.58 |
| NM | $159,420.53 |
| NY | $152,625.64 |
| GA | $146,335.85 |
| PA | $140,767.84 |
| VA | $138,000.65 |
| CO | $122,868.69 |
| TN | $104,784.08 |
| OR | $100,641.31 |
| MA | $95,550.23 |
| WA | $94,081.20 |
| CT | $91,489.16 |
| NJ | $90,583.86 |
| OH | $76,780.04 |
| LA | $62,617.28 |
| SC | $53,906.51 |
| AL | $48,898.02 |
| RI | $33,701.03 |
| UT | $29,841.44 |
| NH | $29,076.05 |
| **Total** | **$4,480,748.98** |

© Critical Path Training. 2015. All Rights Reserved

# Adding Lookup Tables to the Data Model

- Data modeling might required adding lookup tables
  - Lookup tables inject extra related data into data model
- Example: Sales Regions table
  - Assign each state to specific sales region
  - Include full state name it required in reporting

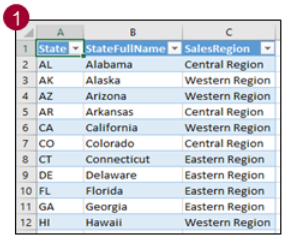



| State | StateFullName | SalesRegion |
|---|---|---|
| AL | Alabama | Central Region |
| AK | Alaska | Western Region |
| AZ | Arizona | Western Region |
| AR | Arkansas | Central Region |
| CA | California | Western Region |
| CO | Colorado | Central Region |
| CT | Connecticut | Eastern Region |
| DE | Delaware | Eastern Region |
| FL | Florida | Eastern Region |
| GA | Georgia | Eastern Region |
| HI | Hawaii | Western Region |

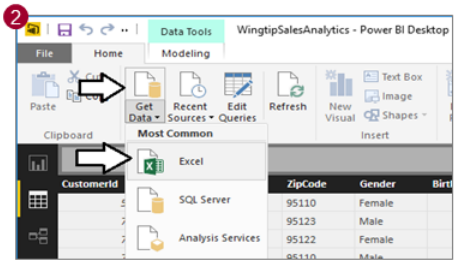
© Critical Path Training. 2015. All Rights Reserved

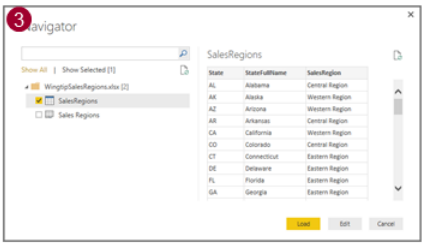
# Importing the SalesRegions Table from Excel

- Import table from Excel using Power Query



| State | StateFullName | SalesRegion |
|---|---|---|
| AL | Alabama | Central Region |
| AK | Alaska | Western Region |
| AZ | Arizona | Western Region |
| AR | Arkansas | Central Region |
| CA | California | Western Region |
| CO | Colorado | Central Region |
| CT | Connecticut | Eastern Region |
| DE | Delaware | Eastern Region |
| FL | Florida | Eastern Region |
| GA | Georgia | Eastern Region |
| HI | Hawaii | Western Region |

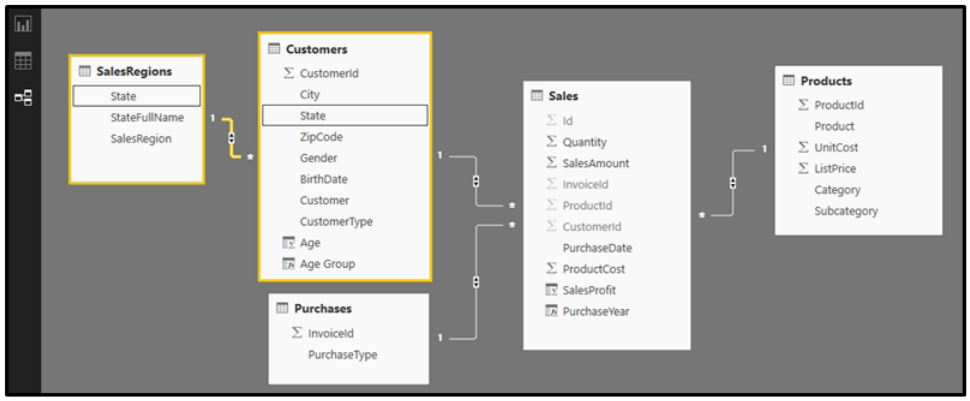



© Critical Path Training. 2015. All Rights Reserved

# Integrating the Lookup Table into the Data Model

- Lookup table must be integrated into data model
  - Accomplished by creating relationship to one or more tables




© Critical Path Training. 2015. All Rights Reserved

# The RELATED Function

- RELATED function performs cross-table lookup
  - Effectively replaces older VLOOKUP function
  - Used in many-side table to look up value from one-side
  - Used to pull data from lookup table into primary table

```
Sales Region = RELATED(SalesRegions[SalesRegion])
```

| CustomerId | City | State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age | Age Group | Sales Region |
|---|---|---|---|---|---|---|---|---|---|---|
| 55 | San Jose | CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 | Ages 65 and over | Western Region |
| 73 | San Jose | CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 | Ages 30 TO 39 | Western Region |
| 74 | San Jose | CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 | Ages 30 TO 39 | Western Region |
| 78 | San Jose | CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 | Ages 30 TO 39 | Western Region |
| 136 | San Jose | CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 | Ages 65 and over | Western Region |
| 150 | San Jose | CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 | Ages 30 TO 39 | Western Region |

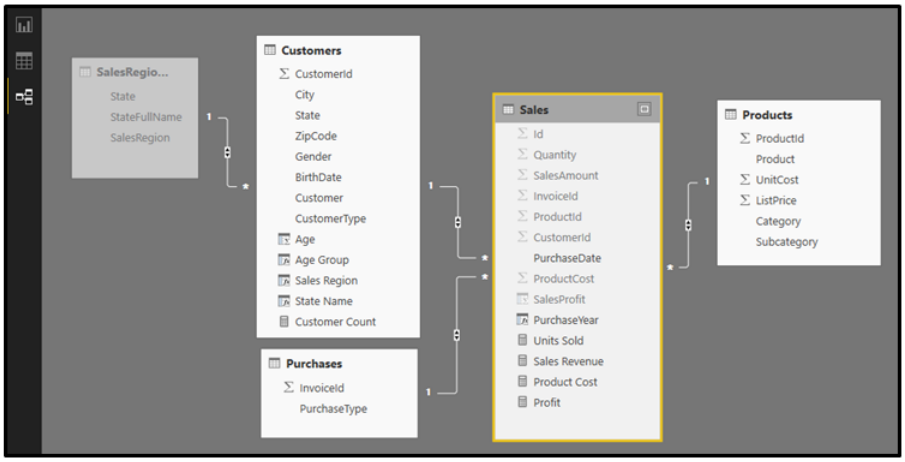
```
State Name = RELATED(SalesRegions[StateFullName])
```

| State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age | Age Group | Sales Region | State Name |
|---|---|---|---|---|---|---|---|---|---|
| CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 | Ages 65 and over | Western Region | California |
| CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 | Ages 30 TO 39 | Western Region | California |
| CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 | Ages 30 TO 39 | Western Region | California |
| CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 | Ages 30 TO 39 | Western Region | California |
| CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 | Ages 65 and over | Western Region | California |
| CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 | Ages 30 TO 39 | Western Region | California |

© Critical Path Training. 2015. All Rights Reserved

# Hiding the Lookup Table

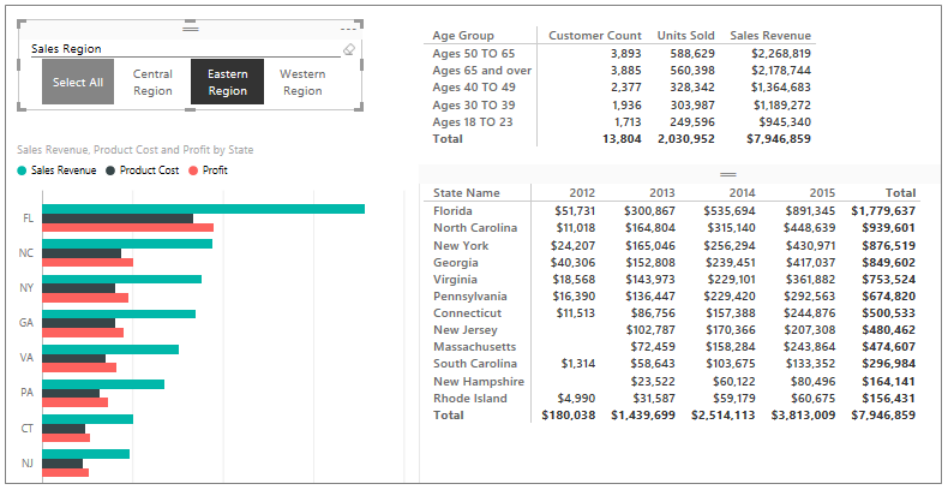
- Lookup table can often be hidden
  - simplifies reporting for data model consumers




© Critical Path Training. 2015. All Rights Reserved

# Filtering on Sales Region

- Calculated column used to calculate slicer values



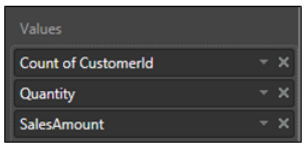

| Age Group | Customer Count | Units Sold | Sales Revenue |
|---|---|---|---|
| Ages 50 TO 65 | 3,893 | 588,629 | $2,268,819 |
| Ages 65 and over | 3,885 | 560,398 | $2,178,744 |
| Ages 40 TO 49 | 2,377 | 328,342 | $1,364,683 |
| Ages 30 TO 39 | 1,936 | 303,987 | $1,189,272 |
| Ages 18 TO 23 | 1,713 | 249,596 | $945,340 |
| **Total** | **13,804** | **2,030,952** | **$7,946,859** |

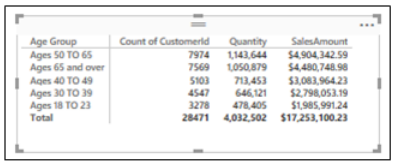
| State Name | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| Florida | $51,731 | $300,867 | $535,694 | $891,345 | $1,779,637 |
| North Carolina | $11,018 | $164,804 | $315,140 | $448,639 | $939,601 |
| New York | $24,207 | $165,046 | $256,294 | $430,971 | $876,519 |
| Georgia | $40,306 | $152,808 | $239,451 | $417,037 | $849,602 |
| Virginia | $18,568 | $143,973 | $229,101 | $361,882 | $753,524 |
| Pennsylvania | $16,390 | $136,447 | $229,420 | $292,563 | $674,820 |
| Connecticut | $11,513 | $86,756 | $157,388 | $244,876 | $500,533 |
| New Jersey | | $102,787 | $170,366 | $207,308 | $480,462 |
| Massachusetts | | $72,459 | $158,284 | $243,864 | $474,607 |
| South Carolina | $1,314 | $58,643 | $103,675 | $133,352 | $296,984 |
| New Hampshire | | $23,522 | $60,122 | $80,496 | $164,141 |
| Rhode Island | $4,990 | $31,587 | $59,179 | $60,675 | $156,431 |
| **Total** | **$180,038** | **$1,439,699** | **$2,514,113** | **$3,813,009** | **$7,946,859** |

© Critical Path Training. 2015. All Rights Reserved

# Benefits of Measures over Calculated Columns

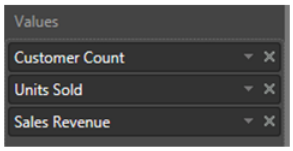
- Calculated columns can be aggregated in visual
  - However, aggregation details are stored in visual
  - Visual doesn't offer control over name and formatting

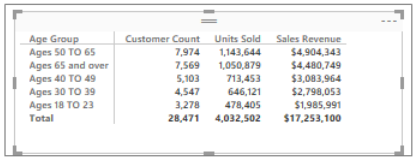


| Age Group | Count of CustomerId | Quantity | SalesAmount |
|---|---|---|---|
| Ages 50 TO 65 | 7974 | 1,143,644 | $4,904,342.59 |
| Ages 65 and over | 7569 | 1,050,879 | $4,480,748.98 |
| Ages 40 TO 49 | 5103 | 713,453 | $3,083,964.23 |
| Ages 30 TO 39 | 4547 | 646,121 | $2,798,053.19 |
| Ages 18 TO 23 | 3278 | 478,405 | $1,985,991.24 |
| Total | 28471 | 4,032,502 | $17,253,100.23 |

- Measure defines name, aggregation and formatting
  - Work is done once and reused across many visuals
  - Makes data model more fool-proof for report designers



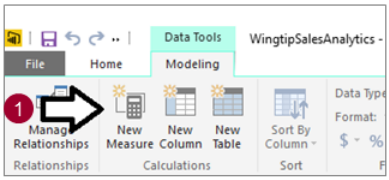



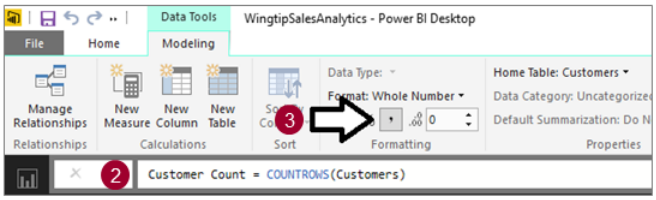
| Age Group | Customer Count | Units Sold | Sales Revenue |
|---|---|---|---|
| Ages 50 TO 65 | 7,974 | 1,143,644 | $4,904,343 |
| Ages 65 and over | 7,569 | 1,050,879 | $4,480,749 |
| Ages 40 TO 49 | 5,103 | 713,453 | $3,083,964 |
| Ages 30 TO 39 | 4,547 | 646,121 | $2,798,053 |
| Ages 18 TO 23 | 3,278 | 478,405 | $1,985,991 |
| Total | 28,471 | 4,032,502 | $17,253,100 |

© Critical Path Training. 2015. All Rights Reserved

# Creating Measures

- Measures have advantage over calculated columns
  - They are evaluated based on the current evaluation context
- Creating a measure with Power BI Desktop
  1. Click New Measure button
  2. Give measure a name and write DAX expressions
  3. Configure formatting

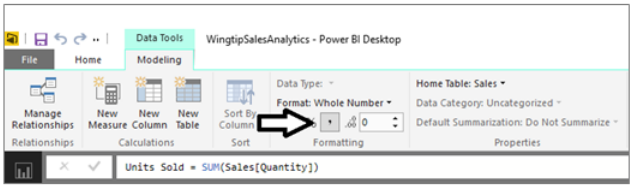


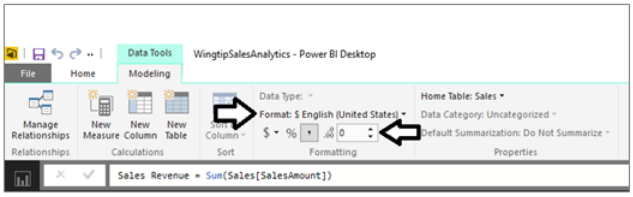

© Critical Path Training. 2015. All Rights Reserved

# Formatting Measures

- Format as whole number




- Format as currency




© Critical Path Training. 2015. All Rights Reserved

# Module 04 Lab: Using the Power Pivot Features of Power BI Desktop

**Lab Time**: 40 minutes

**Lab Folder**: C:\Student\Modules\PowerPivot\Lab

**Lab Overview**: In this set of lab exercises, you will continue to work on the Power BI Desktop project you started in the previous lab. At this point, you have already imported Wingtip sales data from a SQL Azure database and transformed it using the Power Query features of Power BI Desktop to create a starting point for the data model associated with your project. In this lab you will begin to leverage the Power Pivot features of Power BI Desktop to enhance this data model by adding calculated columns and calculated fields. Along the way, you will add pages and visuals to a Power BI Desktop report so you can see the effects of your modeling efforts.

**Important:** This lab assumes you have completed the previous lab titled **Using the Power Query Features of Power BI Desktop** in which you created a Power BI Desktop project named **WingtipSalesAnalytics.pbix**. In the previous lab you should have imported data into the data model using data from the **WingtipSalesDB** database in SQL Azure and transformed the data into a schema that is better suited for data modeling and analysis. If you would like to begin work on this lab without first completing the previous lab, use the Windows Explorer to copy the lab solution file at **C:\Student\Modules\PowerQuery\LabSolution\WingtipSalesAnalytics.pbix** into the folder at **C:\Student\Projects**.

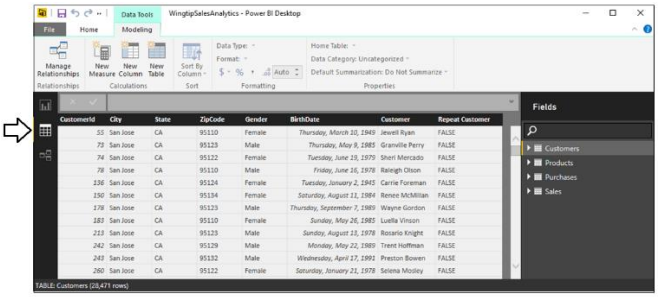
## Exercise 1: Hiding and Formatting Columns in the Data Model

In this exercise you will start with the data model you created in the previous lab and begin by hiding and formatting table columns inside the data model. After that, you will create a simple report to see the effects of your formatting changes.

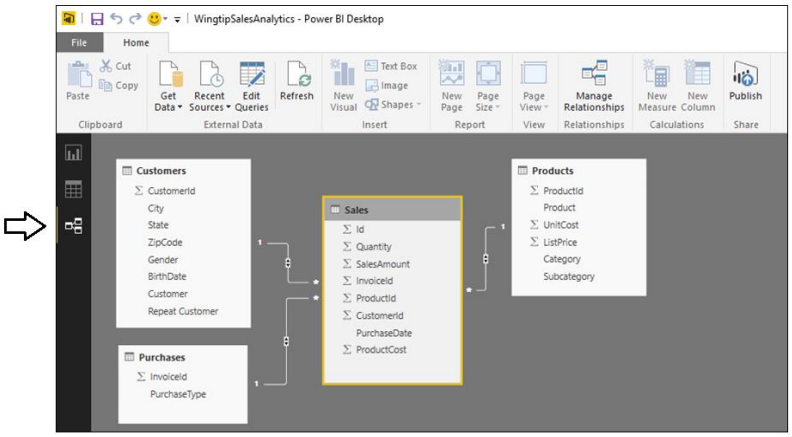
1. Launch Power BI Desktop to start a new project.
2. Open the Power BI Desktop project named **WingtipSalesAnalytics.pbix** from the previous lab located at the following path.

```
C:\Student\Projects\WingtipSalesAnalytics.pbix
```

3. When the project opens, click the table icon in the middle of the sidebar to enter data view mode.

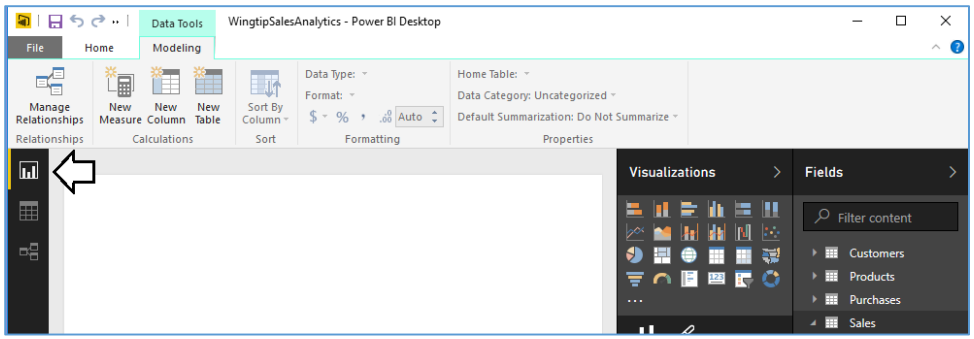



4. Take a moment to review the data in each of the four tables in the data model by clicking on the tables inside the **Fields** list.
5. Click the bottom button in the sidebar to navigate to relationship view. You should see that the four tables are arraigned in a star schema where the **Sales** table has a relationship established with each of the three other tables in the data model

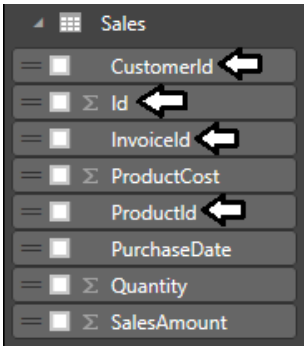



© Critical Path Training. 2015. All Rights Reserved

6. Now it's time to inspect the data model from a different perspective. More specifically, you will examine the data model from the perspective of a consumer who is designing reports and creating visuals using the Power BI Desktop report designer.
   a) Click the top button in the sidebar to navigate to report view.

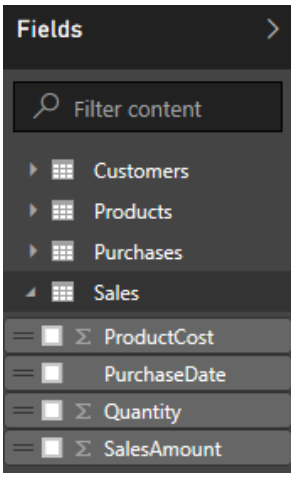



   b) Inside the **Fields** list, use the mouse to expand the fields inside the **Sales** table. You can see that there are several fields in the Sales table that will never be used when designing reports such as the four identifier columns. The data model will be easier for consumers such as report designers to understand if you hide these types of fields which add unnecessary clutter.



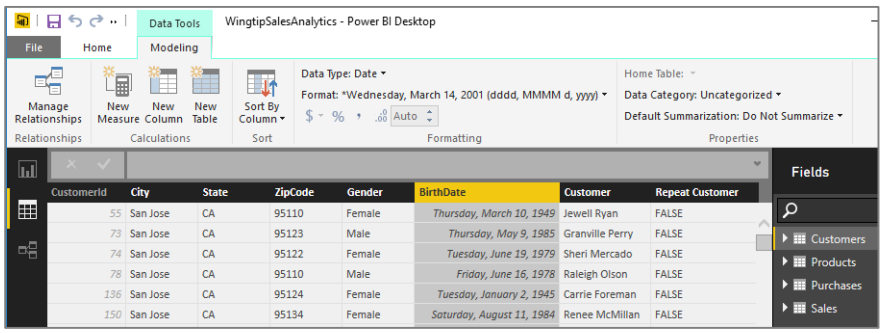

7. Use relationship view to update the data model by hiding fields in the **Sales** table that are unnecessary to display in report view.
   a) Navigate to relationship view in the Power BI Desktop window.
   b) Using the mouse, right-click on the **Id** column in the **Sales** table and select the **Hide in Report View** command.
   c) Right-click on the **InvoiceId** column in the **Sales** table and select the **Hide in Report View** command.
   d) Right-click on the **CustomerId** column in the **Sales** table and select the **Hide in Report View** command.
   e) Right-click on the **ProductId** column in the **Sales** table and select the **Hide in Report View** command.
   f) Now, navigate back to report view and examine the set of fields displayed for the **Sales** table.

© Critical Path Training. 2015. All Rights Reserved

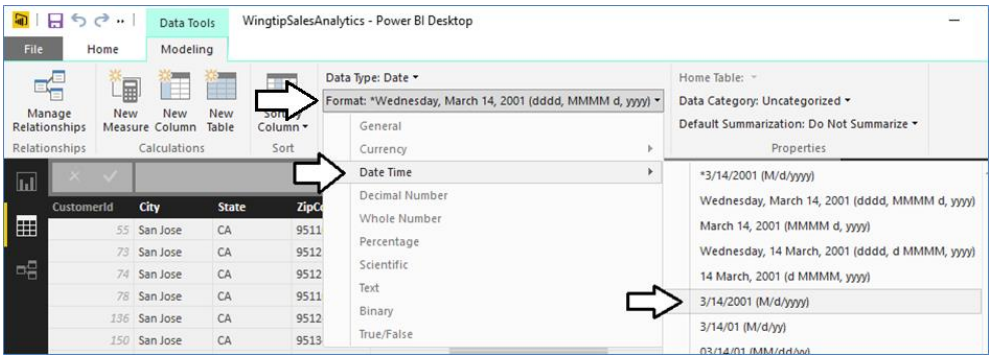



You should be able to see that hiding unnecessary columns from report view makes your data model easier to use. This is especially true in the scenario where you are creating a data model that other will be using to create reports and dashboards.

8. Modify the formatting of the **BirthDate** column in the **Customers** table.
   a) In the Power BI Desktop windows, navigate back to data view.
   b) In the **Fields** list on the right, select the **Customers** table to display its rows and columns.
   c) Select the **BirthDate** column by clicking on its column header.




   d) Modify the formatting of the **BirthDate** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Date Time > 3/14/2001 (M/d/yyyy)**.

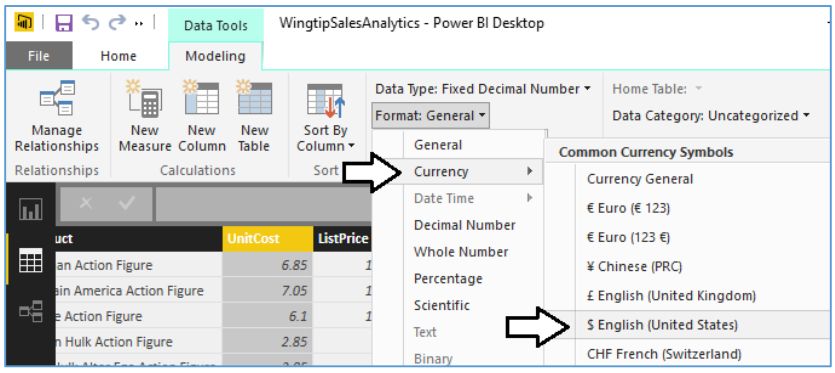



   e) The **BirthDate** column should now reflect the change in formatting.

© Critical Path Training. 2015. All Rights Reserved

| CustomerId | City | State | ZipCode | Gender | BirthDate | Customer | Repeat Customer |
|---|---|---|---|---|---|---|---|
| 55 | San Jose | CA | 95110 | Female | 3/10/1949 | Jewell Ryan | FALSE |
| 73 | San Jose | CA | 95123 | Male | 5/9/1985 | Granville Perry | FALSE |
| 74 | San Jose | CA | 95122 | Female | 6/19/1979 | Sheri Mercado | FALSE |
| 78 | San Jose | CA | 95110 | Male | 6/16/1978 | Raleigh Olson | FALSE |
| 136 | San Jose | CA | 95124 | Female | 1/2/1945 | Carrie Foreman | FALSE |

9. Modify the formatting of columns in the **Products** table.
   a) In the **Fields** list on the right, select the **Products** table to display its rows and columns.
   b) Select the **UnitCost** column by clicking on its column header.
   c) Use the **Format** menu button in the ribbon to update the format setting to **Currency > English (United States)**.

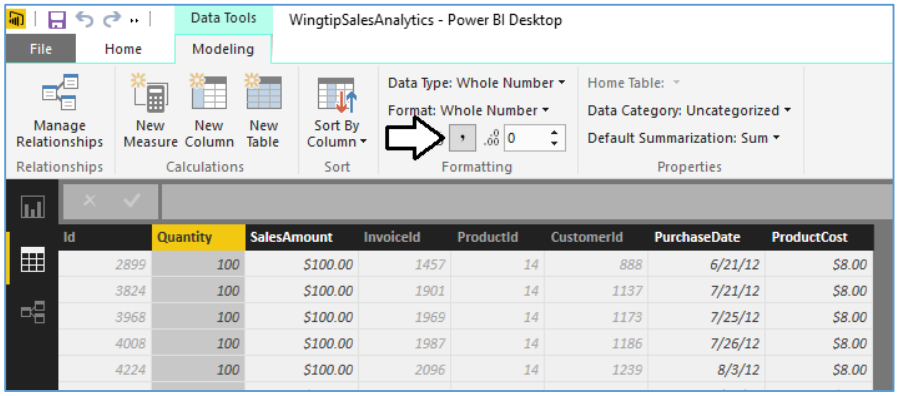



   d) Changing the format setting of the **ListPrice** column to **Currency > English (United States)** so it matches **UnitCost**.

| ProductId | Product | UnitCost | ListPrice |
|---|---|---|---|
| 1 | Batman Action Figure | $6.85 | $14.95 |
| 2 | Captain America Action Figure | $7.05 | $12.95 |
| 3 | GI Joe Action Figure | $6.10 | $14.95 |
| 4 | Green Hulk Action Figure | $2.85 | $9.95 |
| 5 | Red Hulk Alter Ego Action Figure | $2.85 | $9.95 |
| 6 | Godzilla Action Figure | $14.25 | $19.95 |

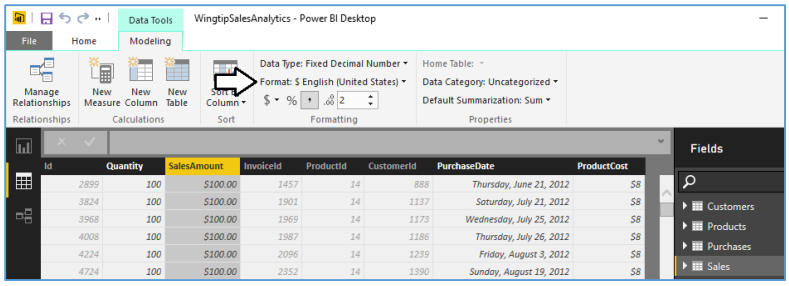
10. Modify the formatting of columns in the **Sales** table.
   a) In the **Fields** list on the right, select the **Sales** table to display its rows and columns.
   b) Select the **Quantity** column by clicking on its column header.
   c) Modify the **Quantity** column by clicking to select the comma button on the ribbon to add a comma separator.

© Critical Path Training. 2015. All Rights Reserved



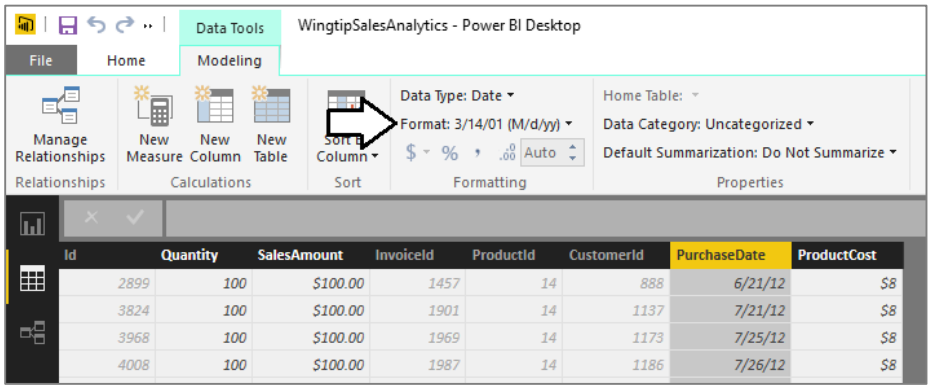

d) Select the **SalesAmount** column by clicking on its column header.

e) Modify the formatting of the **SalesAmount** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Currency > English (United States)**.



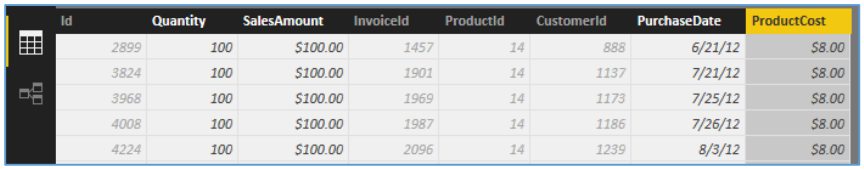

f) Select the **PurchaseDate** column by clicking on its column header.

g) Modify the formatting of the **PurchaseDate** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Date Time > 3/14/2001 (M/d/yyyy)**.



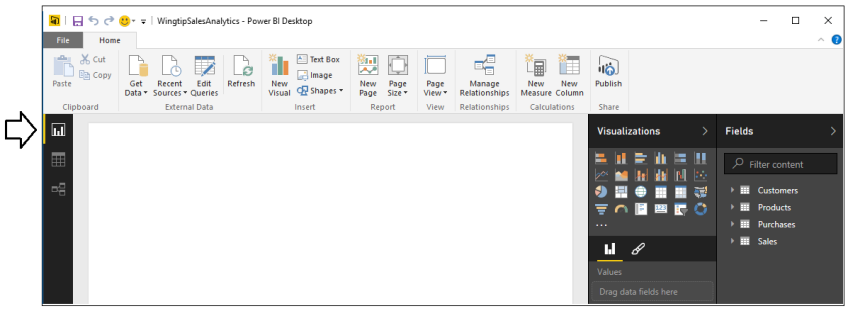

h) Select the **ProductCost** column by clicking on its column header.

i) Modify the formatting of the **ProductCost** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Currency > English (United States)**.

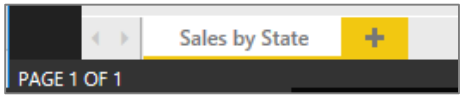



© Critical Path Training. 2015. All Rights Reserved

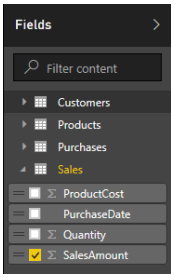
11. See the effect of your formatting by adding a visual to a report.
    a) Navigate to report view. There should be an empty report for the project with a single page named **Page 1**.



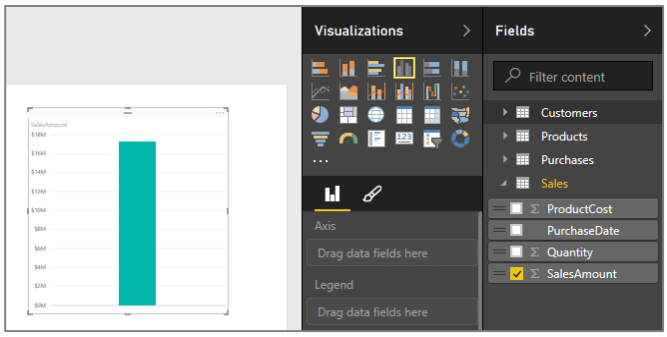

    b) Change the name of the page in the report from **Page 1** to **Sales by State**.



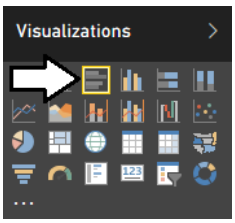

    c) Create a new visual in the report by selecting the checkbox for the **SalesAmount** column in the **Fields** list.




    d) When you select the **SalesAmount** column, Power BI Desktop will automatically add a new visual to the report based on the visualization type of **Clustered Column Chart**.

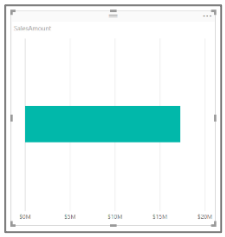



    e) Click the **Clustered Bar Chart** button in the **Visualizations** list to change the visualization type to a clustered bar chart.

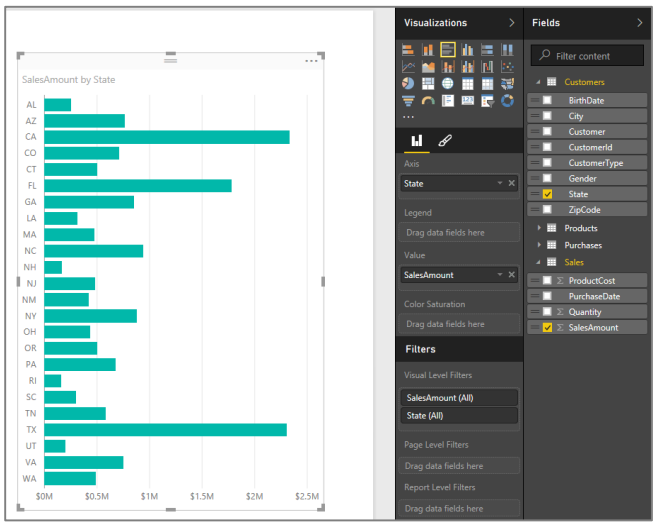



© Critical Path Training. 2015. All Rights Reserved

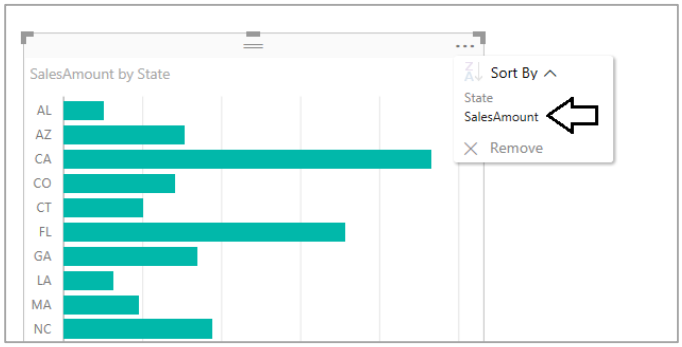
f) Since you haven't added any row labels yet, the clustered bar chart currently displays a single bar showing total sales.



g) Drag and drop the **States** field from the **Fields** list into the **Axis** well of the **Visualizations** pane.

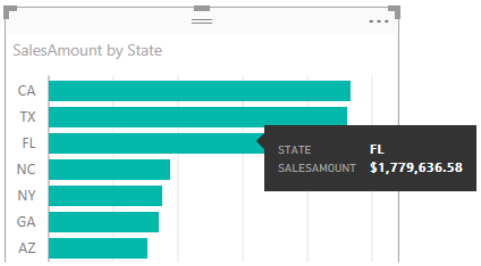



h) Sort the data in the visualization by clicking the ellipse (**…**) button in the top right corner of the visual to invoke the fly-out menu. From the fly-out menu, select **Sort By ^** > **SalesAmount** to sort by the aggregated **SalesAmount** values in descending order.

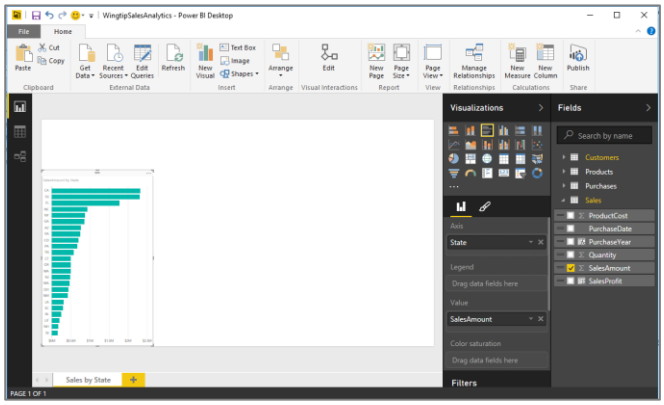



i) The visual should now sort states with the greatest amounts of sales revenue to the top.

© Critical Path Training. 2015. All Rights Reserved




j) Use your mouse to reposition the visual to the bottom left corner of the page as shown in the following screenshot.

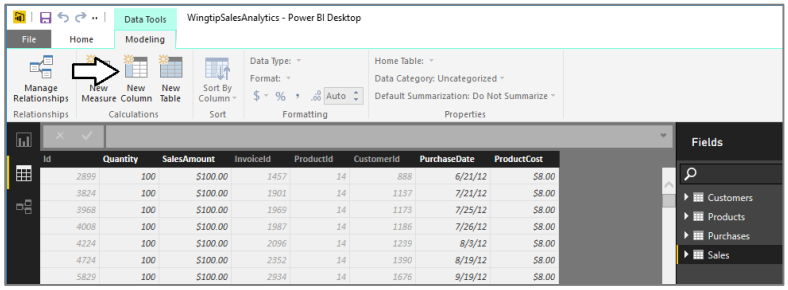


12. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 2: Extending the Data Model by Creating Calculated Columns

In this exercise you will create several calculated columns which will require you to write and test DAX expressions. After creating calculated columns, you will use them to enhance reports in the current project.

1. Add a calculated column to the **Sales** table named **SalesProfit** to determine profit by calculating the difference between **SalesAmount** and **ProductCost**.
   a) Navigate to data view.
   b) Select the **Sales** table in the **Fields** list.
   c) Create a new calculated column by clicking the **New Column** button in the ribbon.

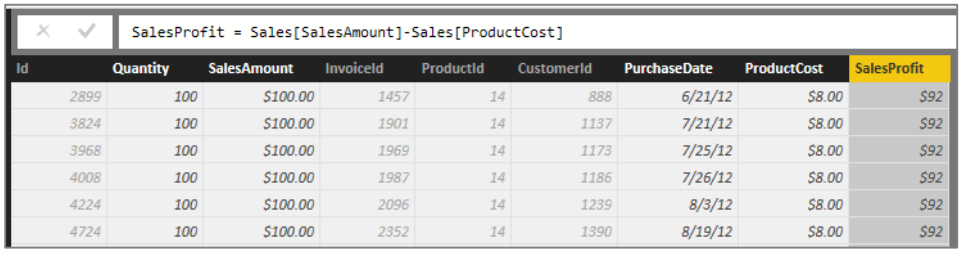



   d) Enter to following DAX expression into the formula bar to create the calculated column named **SalesProfit**.

```
SalesProfit = Sales[SalesAmount]-Sales[ProductCost]
```

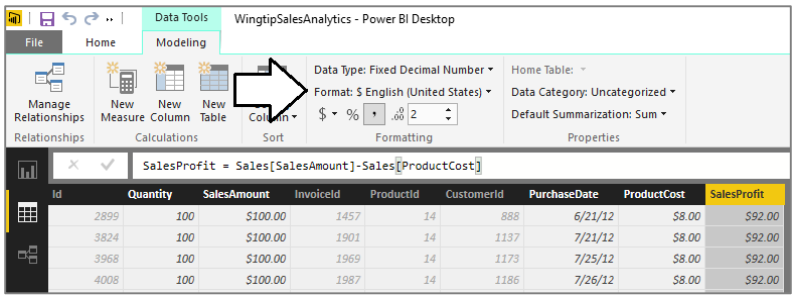
   e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a **SalesProfit** value for each row in the **Sales** table.

© Critical Path Training. 2015. All Rights Reserved

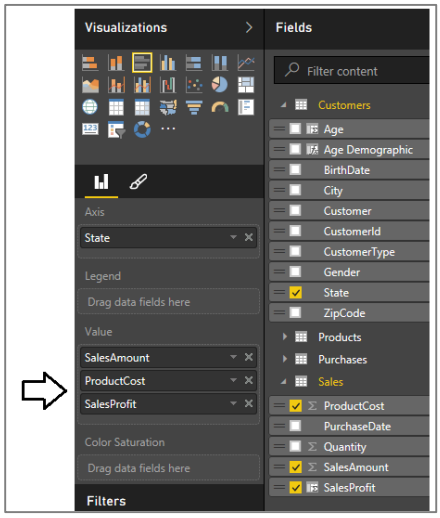



| Id | Quantity | SalesAmount | InvoiceId | ProductId | CustomerId | PurchaseDate | ProductCost | SalesProfit |
|---|---|---|---|---|---|---|---|---|
| 2899 | 100 | $100.00 | 1457 | 14 | 888 | 6/21/12 | $8.00 | $92 |
| 3824 | 100 | $100.00 | 1901 | 14 | 1137 | 7/21/12 | $8.00 | $92 |
| 3968 | 100 | $100.00 | 1969 | 14 | 1173 | 7/25/12 | $8.00 | $92 |
| 4008 | 100 | $100.00 | 1987 | 14 | 1186 | 7/26/12 | $8.00 | $92 |
| 4224 | 100 | $100.00 | 2096 | 14 | 1239 | 8/3/12 | $8.00 | $92 |
| 4724 | 100 | $100.00 | 2352 | 14 | 1390 | 8/19/12 | $8.00 | $92 |

f) Configure the column's formatting by using the **Format** menu on the ribbon to select **Currency > English (United States)**.

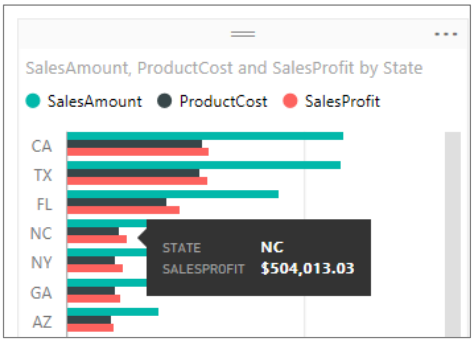



2. Update the clustered bar chart visual you created in the previous exercise.
   a) Navigate to report view.
   b) Select the visual you created in the previous exercise.
   c) Using your mouse, drag and drop the **ProductCost** column from the **Fields** list into the **Value** well in the **Visualizations** pane.
   d) Using your mouse, drag and drop the **SalesProfit** column from the **Fields** list into the **Value** well in the **Visualizations** pane.




   e) You should now see that the cluster bar chart is showing additional bars for product cost and profit.

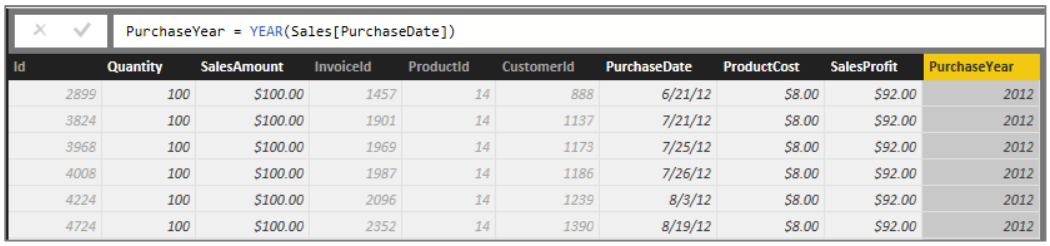
© Critical Path Training. 2015. All Rights Reserved




3. Add a calculated column to the **Sales** table named **PurchaseYear** to indicate the calendar year of each purchase.
   a) Navigate to data view.
   b) Select the **Sales** table in the **Fields** list.
   c) Create a new calculated column by clicking the **New Column** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the calculated column named **PurchaseYear**.

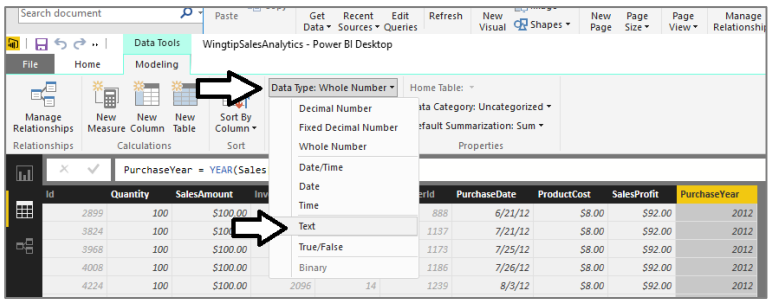
```
PurchaseYear = YEAR(Sales[PurchaseDate])
```

   e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a calendar year value (e.g. 2012) for each row in the **Sales** table




| Id | Quantity | SalesAmount | InvoiceId | ProductId | CustomerId | PurchaseDate | ProductCost | SalesProfit | PurchaseYear |
|---|---|---|---|---|---|---|---|---|---|
| 2899 | 100 | $100.00 | 1457 | 14 | 888 | 6/21/12 | $8.00 | $92.00 | 2012 |
| 3824 | 100 | $100.00 | 1901 | 14 | 1137 | 7/21/12 | $8.00 | $92.00 | 2012 |
| 3968 | 100 | $100.00 | 1969 | 14 | 1173 | 7/25/12 | $8.00 | $92.00 | 2012 |
| 4008 | 100 | $100.00 | 1987 | 14 | 1186 | 7/26/12 | $8.00 | $92.00 | 2012 |
| 4224 | 100 | $100.00 | 2096 | 14 | 1239 | 8/3/12 | $8.00 | $92.00 | 2012 |
| 4724 | 100 | $100.00 | 2352 | 14 | 1390 | 8/19/12 | $8.00 | $92.00 | 2012 |

   f) Once the **PurchaseYear** column has been created, use the **Data Type** dropdown menu on the ribbon to change the column type from **Whole Number** to **Text**.

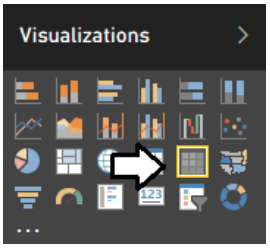



You might be wondering why it is important to modify the data type of the **PurchaseYear** column to be a **Text** column. If you leave the **Data Type** setting at its default value, the column type is set to the type **Whole Number**. This can cause confusion because in certain scenarios Power BI Desktop will assume that the **PurchaseYear** column values are numeric values that can be aggregated using operations such as SUM or AVERAGE. Configuring the **PurchaseYear** column with the **Text** data type is a technique that prevents Power BI Desktop from incorrectly assuming that it can aggregate its values.

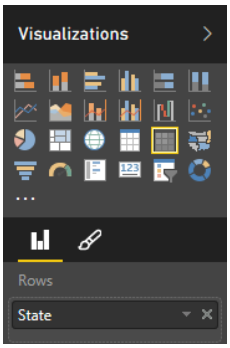
4. Use the new calculated column named **PurchaseYear** in a report
   a) Navigate to report view.
   b) Make sure that no visuals are selected on the page so that you can create a new visual.
   c) Select the checkbox for the **SalesAmount** column in the **Fields** list to create a new visual.

© Critical Path Training. 2015. All Rights Reserved

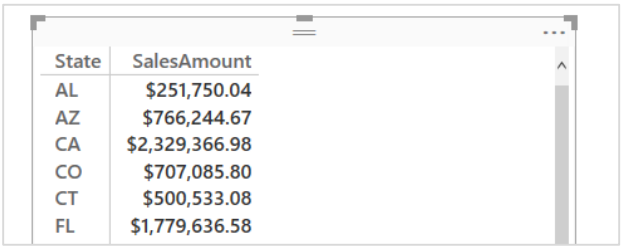
d) Click the **Matrix** button in the **Visualizations** list to change the visualization type to a matrix.




e) Add a second field to the visual by dragging the **State** column from the **Customers** table in the **Fields** list into the **Rows** well to displays a separate row for each state.

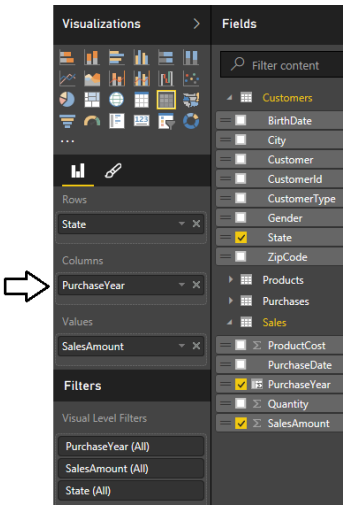



f) Now the matrix should display a row for each state that shows the aggregated sum of the **SalesAmount** column.



| State | SalesAmount |
|---|---|
| AL | $251,750.04 |
| AZ | $766,244.67 |
| CA | $2,329,366.98 |
| CO | $707,085.80 |
| CT | $500,533.08 |
| FL | $1,779,636.58 |

g) Extend the matrix visual to pivot on the **PurchaseYear** column. Accomplish this by using the mouse to drag the **PurchaseYear** column from the **Fields** list and drop it into the **Columns** well in the **Visualization** pane




© Critical Path Training. 2015. All Rights Reserved

h) The visual should now display a column for each year from 2012 through 2015. Resize the width of the visual so you can see all the columns.

| State | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| AL | $8,537.95 | $41,688.00 | $77,091.38 | $124,432.71 | **$251,750.04** |
| AZ | $83,217.79 | $176,586.42 | $224,727.85 | $281,712.61 | **$766,244.67** |
| CA | $270,926.32 | $550,160.02 | $737,878.53 | $770,402.11 | **$2,329,366.98** |
| CO | $64,683.47 | $154,678.89 | $215,370.45 | $272,352.99 | **$707,085.80** |
| CT | $11,512.65 | $86,755.99 | $157,387.97 | $244,876.47 | **$500,533.08** |
| FL | $51,730.85 | $300,866.87 | $535,693.94 | $891,344.92 | **$1,779,636.58** |
| GA | $40,305.80 | $152,807.51 | $239,451.05 | $417,037.28 | **$849,601.64** |
| LA | $4,467.73 | $54,639.88 | $105,052.72 | $150,823.54 | **$314,983.87** |
| MA | | $72,459.05 | $158,283.62 | $243,864.24 | **$474,606.91** |
| NC | $11,018.02 | $164,804.24 | $315,139.92 | $448,638.72 | **$939,600.90** |
| NH | | $23,522.41 | $60,121.72 | $80,496.42 | **$164,140.55** |
| NJ | | $102,786.77 | $170,366.35 | $207,308.44 | **$480,461.56** |
| NM | $52,066.79 | $99,342.05 | $131,171.78 | $139,195.07 | **$421,775.69** |
| NY | $24,207.43 | $165,046.23 | $256,294.27 | $430,971.24 | **$876,519.17** |
| OH | $2,491.32 | $99,245.08 | $148,166.21 | $186,107.10 | **$436,009.71** |
| OR | $76,153.05 | $108,898.70 | $142,070.40 | $172,165.22 | **$499,287.37** |
| PA | $16,389.99 | $136,447.11 | $229,419.50 | $292,563.22 | **$674,819.82** |
| RI | $4,990.25 | $31,586.78 | $59,179.35 | $60,674.54 | **$156,430.92** |
| SC | $1,314.35 | $58,643.31 | $103,674.55 | $133,352.10 | **$296,984.31** |
| TN | $27,443.02 | $135,755.12 | $205,616.58 | $215,253.52 | **$584,068.24** |
| TX | $212,085.08 | $490,643.98 | $683,079.11 | $919,030.36 | **$2,304,838.53** |
| UT | $20,101.20 | $36,832.07 | $59,272.82 | $83,136.96 | **$199,343.05** |
| VA | $18,568.32 | $143,972.83 | $229,100.97 | $361,881.60 | **$753,523.72** |
| WA | $74,614.94 | $101,064.69 | $131,768.38 | $184,039.11 | **$491,487.12** |
| To... | **$1,076,82...** | **$3,489,23...** | **$5,375,37...** | **$7,311,66...** | **$17,253,100...** |

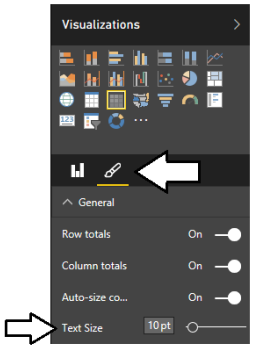
i) Now it's time to apply sorting on the matrix visual. Accomplish this by clicking on the **Total** column heading in the top right corner of the visual.

| 2014 | 2 | Total |
|---|---|---|
| $77,091.38 | $124,432.71 | **$251,750.04** |
| $224,727.85 | $281,712.61 | **$766,244.67** |
| $737,878.53 | $770,402.11 | **$2,329,366.98** |
| $215,370.45 | $272,352.99 | **$707,085.80** |
| $157,387.97 | $244,876.47 | **$500,533.08** |

j) The rows in the visual should now be sorted so that the states with the greatest amount of sales revenue are on top.

| State | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| CA | $270,926.32 | $550,160.02 | $737,878.53 | $770,402.11 | **$2,329,366.98** |
| TX | $212,085.08 | $490,643.98 | $683,079.11 | $919,030.36 | **$2,304,838.53** |
| FL | $51,730.85 | $300,866.87 | $535,693.94 | $891,344.92 | **$1,779,636.58** |
| NC | $11,018.02 | $164,804.24 | $315,139.92 | $448,638.72 | **$939,600.90** |
| NY | $24,207.43 | $165,046.23 | $256,294.27 | $430,971.24 | **$876,519.17** |
| GA | $40,305.80 | $152,807.51 | $239,451.05 | $417,037.28 | **$849,601.64** |

k) Increase the font size of the matrix visual. Accomplishing this by first clicking the Edit Pen button in the **Visualization** pane to view the appearance properties of the visual and then by modifying the **Text Size** property to a value of **10pt**.



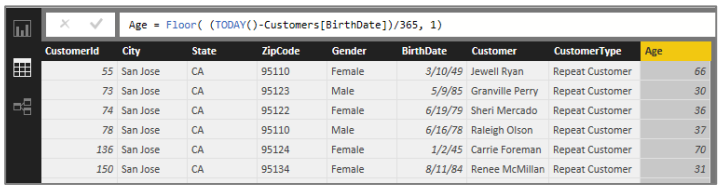

l) Using the mouse, rearrange the two visuals on that page to match the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

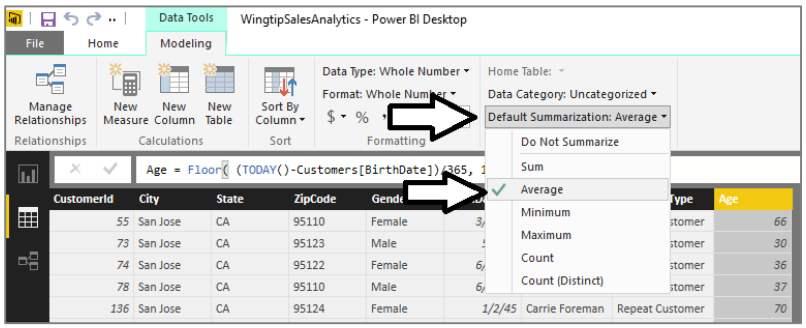
5. Add a calculated column to the **Customers** table named **Age** to indicate the age of the customer.
   a) Navigate to data view.
   b) Select the **Customers** table in the **Fields** list.
   c) Create a new calculated column by clicking the **New Column** button in the ribbon.
   d) Enter the following DAX expression into the formula bar to create the calculated column named **Age**.

```
Age = Floor( (TODAY()-Customers[BirthDate])/365, 1)
```

   e) Press the **ENTER** key to add the calculated column. You should be able to see a whole number for the age of each customer.




   f) Use the **Default Summarization** dropdown menu in the ribbon to change the default summarization setting for **Age** column from **Sum** to **Average**.

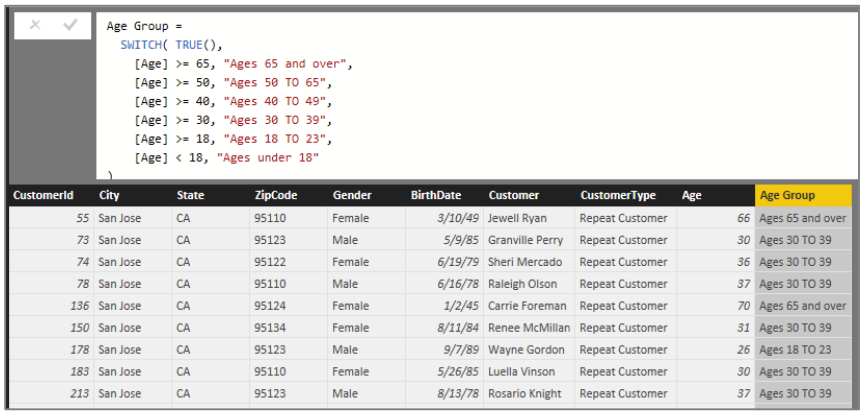



6. Add a calculated column to the **Customers** table named **Age Group** to break customers up into age-based sets.
   a) Create a new calculated column in the **Sales** table by clicking the **New Column** button in the ribbon.
   b) Enter to following DAX expression into the formula bar to create the calculated column named **Age Group**.

```
Age Group =
  SWITCH( TRUE(),
    [Age] >= 65, "Ages 65 and over",
    [Age] >= 50, "Ages 50 TO 65",
    [Age] >= 40, "Ages 40 TO 49",
    [Age] >= 30, "Ages 30 TO 39",
    [Age] >= 18, "Ages 18 TO 23",
    [Age] < 18, "Ages under 18"
)
```

© Critical Path Training. 2015. All Rights Reserved

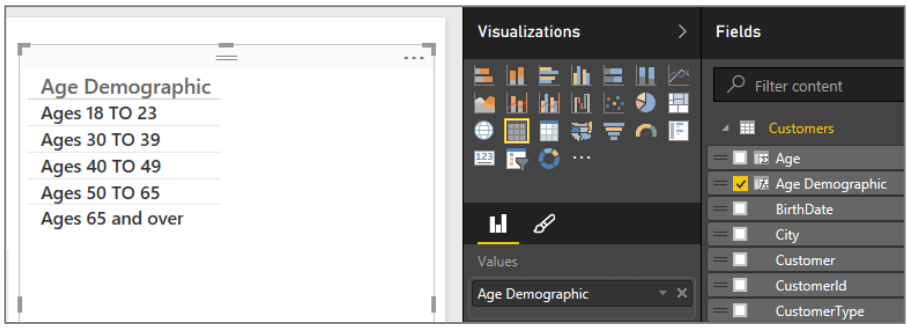
c) After creating the calculated column, you should be able to verify that the **Age Group** column calculates a value for each customer row which places that customer in a bucket for a particular age group.



```
Age Group =
  SWITCH( TRUE(),
    [Age] >= 65, "Ages 65 and over",
    [Age] >= 50, "Ages 50 TO 65",
    [Age] >= 40, "Ages 40 TO 49",
    [Age] >= 30, "Ages 30 TO 39",
    [Age] >= 18, "Ages 18 TO 23",
    [Age] < 18, "Ages under 18"
  )
```

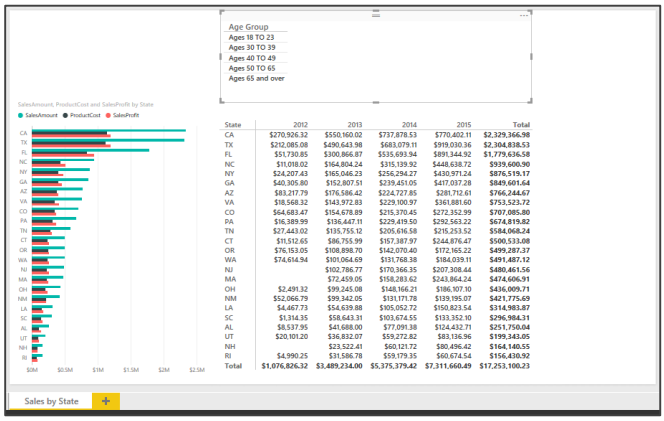
| CustomerId | City | State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age | Age Group |
|---|---|---|---|---|---|---|---|---|---|
| 55 | San Jose | CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 | Ages 65 and over |
| 73 | San Jose | CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 | Ages 30 TO 39 |
| 74 | San Jose | CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 | Ages 30 TO 39 |
| 78 | San Jose | CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 | Ages 30 TO 39 |
| 136 | San Jose | CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 | Ages 65 and over |
| 150 | San Jose | CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 | Ages 30 TO 39 |
| 178 | San Jose | CA | 95123 | Male | 9/7/89 | Wayne Gordon | Repeat Customer | 26 | Ages 18 TO 23 |
| 183 | San Jose | CA | 95110 | Female | 5/26/85 | Luella Vinson | Repeat Customer | 30 | Ages 30 TO 39 |
| 213 | San Jose | CA | 95123 | Male | 8/13/78 | Rosario Knight | Repeat Customer | 37 | Ages 30 TO 39 |

7. Use the **Age Group** calculated column as a row label in a matrix visual.
   a) Navigate to report view.
   b) Make sure that no visuals as selected on the page.
   c) Create a new visual in the report by selecting the checkbox for the **Age Group** column in the **Fields** list.



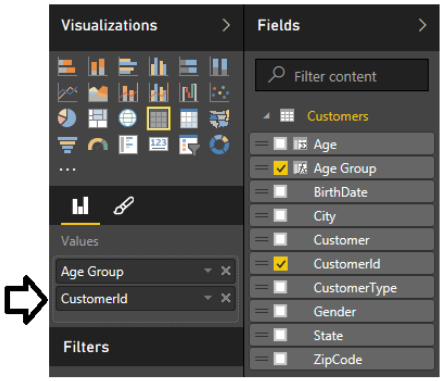

Note that this visual does not display a row for the age demographic value of **Age under 18** even thought you added that value to the DAX formula. The reason for this is that the Wingtip Sales database you are working with does not contain any customers under 18 years of age so the age demographic value of **Age under 18** is automatically filtered out of the visual.

   d) Change the visualization type of the new visual by clicking the **Matrix** button in the **Visualizations** list.
   e) Use the mouse to resize and reposition the visual so it appears on top of the matrix you created earlier.

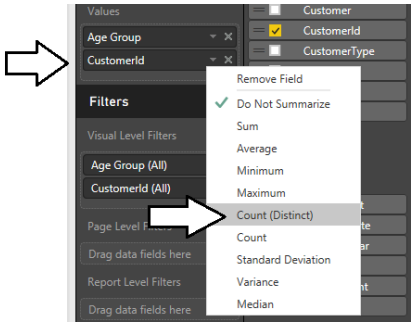



© Critical Path Training. 2015. All Rights Reserved

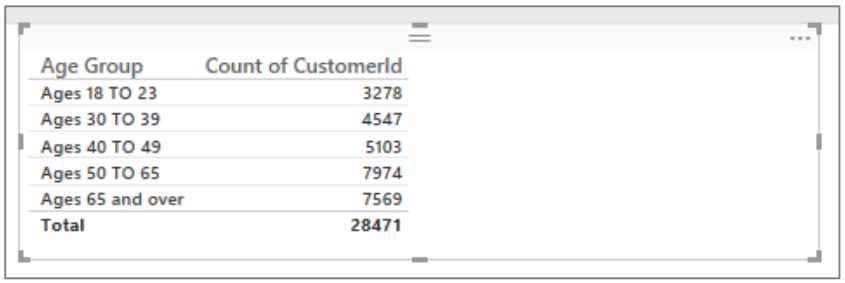
f) Now it's time to modify the matrix visual by adding a new column to show the count of customers in each age group. Accomplish this by dragging the **CustomerId** column from the **Customers** table in the **Fields** list and dropping it into the **Values** well in the **Visualizations** pane. When you drop the **CustomerId** column into the **Values** well, make sure to drop it so it appears underneath the **Age Group** column.




g) Click on the dropdown menu of the **CustomerId** field inside the **Values** well so you can see and change the aggregation that will be performed. Select the aggregation type that is **Count (Distinct)**.




h) Now you should see one row per age group which includes a total count of customers.



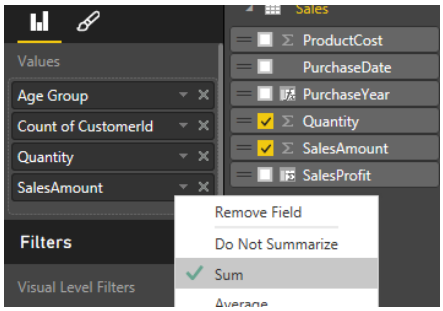
| Age Group | Count of CustomerId |
|---|---|
| Ages 18 TO 23 | 3278 |
| Ages 30 TO 39 | 4547 |
| Ages 40 TO 49 | 5103 |
| Ages 50 TO 65 | 7974 |
| Ages 65 and over | 7569 |
| **Total** | **28471** |

i) Drag the **Quantity** column from the **Sales** table and drop it into the **Values** well in the **Visualizations** pane.

j) Drag the **SalesAmount** column from the **Sales** table and drop it into the **Values** well in the **Visualizations** pane.

k) Configure the summarization type for the **Quantity** field and the **SalesAmount** field to perform a **Sum**.

© Critical Path Training. 2015. All Rights Reserved




l) The visual should now display a new **Quantity** column displaying the total number of units sold as well as a **SalesAmount** column displaying total sales revenue.

| Age Group | Count of CustomerId | Quantity | SalesAmount |
|---|---|---|---|
| Ages 18 TO 23 | 3278 | 478,405 | $1,985,991.24 |
| Ages 30 TO 39 | 4547 | 646,121 | $2,798,053.19 |
| Ages 40 TO 49 | 5103 | 713,453 | $3,083,964.23 |
| Ages 50 TO 65 | 7974 | 1,143,644 | $4,904,342.59 |
| Ages 65 and over | 7569 | 1,050,879 | $4,480,748.98 |
| **Total** | **28471** | **4,032,502** | **$17,253,100.23** |

m) Click on the column header for **SalesAmount** to sort the rows so that age groups with the greatest sales revenue to the top.

| Age Group | Count of CustomerId | Quantity | SalesAmount |
|---|---|---|---|
| Ages 50 TO 65 | 7974 | 1,143,644 | $4,904,342.59 |
| Ages 65 and over | 7569 | 1,050,879 | $4,480,748.98 |
| Ages 40 TO 49 | 5103 | 713,453 | $3,083,964.23 |
| Ages 30 TO 39 | 4547 | 646,121 | $2,798,053.19 |
| Ages 18 TO 23 | 3278 | 478,405 | $1,985,991.24 |
| **Total** | **28471** | **4,032,502** | **$17,253,100.23** |

At this point, you might be bothered by the fact that some of the column header names in this visual are not a pretty as they could be. For example, it would be less confusing to business users if the column displaying a count of customers had a column heading of **Customer Count** instead of **Count of CustomerId**. You will address this issue later in the lab exercise where you begin to extend the data model by creating measures.

8. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 3: Extending the Data Model with a Lookup Table

In this exercise, you will extend the data model by adding a new lookup table named **SalesRegions** which assigns a geographic sales region to each state. The work to accomplish this will involve importing data from an Excel workbook to create the **SalesRegions** table and also creating a relationship between the **SalesRegions** table and the **Customers** table. You will also create calculated columns in the **Customers** table to pull in related data from the **SalesRegions** table.

1. Locate the Excel workbook file which contains the **SalesRegions** table.
   a) Using Windows Explorer, open the folder at **c:\Student\Data** and locate the Excel workbook file named **SalesRegions.xlsx**.
   b) Double-click on named **SalesRegions.xlsx**. to open it inside Microsoft Excel 2016.
   c) Once the workbook file opens in Excel, take a moment to examine what's inside. You should be able to verify that there is a **SalesRegions** table which contains a row for each of the 50 states along with the state full name and an assigned sales region. As you can see from the following screenshot, the 50 states have been divided into three sales regions which are **Western Region**, **Central Region** and **Eastern Region**.

© Critical Path Training. 2015. All Rights Reserved

| | A | B | C |
|---|---|---|---|
| 1 | State | StateFullName | SalesRegion |
| 2 | AL | Alabama | Central Region |
| 3 | AK | Alaska | Western Region |
| 4 | AZ | Arizona | Western Region |
| 5 | AR | Arkansas | Central Region |
| 6 | CA | California | Western Region |
| 7 | CO | Colorado | Central Region |
| 8 | CT | Connecticut | Eastern Region |
| 9 | DE | Delaware | Eastern Region |
| 10 | FL | Florida | Eastern Region |
| 11 | GA | Georgia | Eastern Region |
| 12 | HI | Hawaii | Western Region |

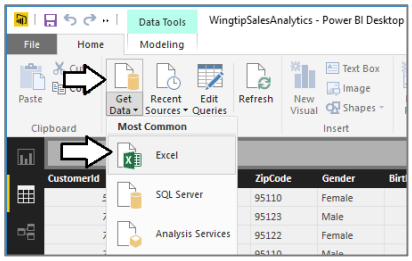
d) Close Microsoft Excel and make sure you do not save any chnages to **SalesRegions.xlsx**.

2. Import the **SalesRegions** table into the data model of the current project.

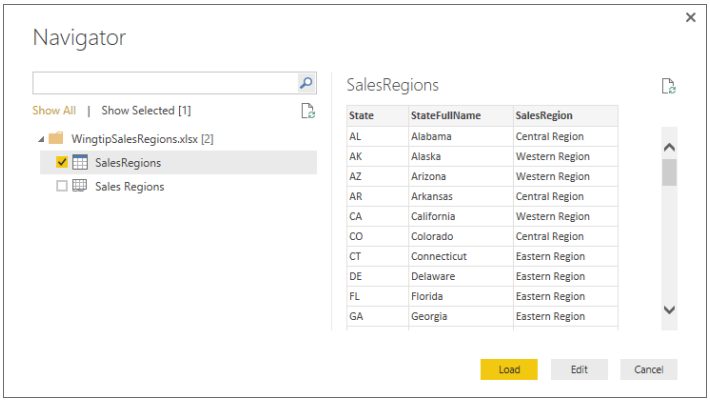
a) Navigate back to Power BI Desktop.

b) Click on the **Home** tab on the ribbon.

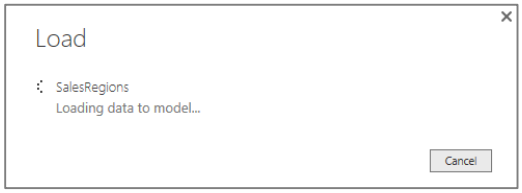
c) Drop down the **Get Data** menu in the ribbon and select the **Excel** command to display the **Navigator** dialog.




d) In the **Navigator** dialog, click the checkbox to select the **SalesRegion** table. Make sure to select the top checkbox for the **SalesRegion** table and not the bottom checkbox for the **Sales Region** worksheet. Once you have selected the **SalesRegion** table, you should be able to see a sample of its rows in the **Navigator** dialog as shown in the following screenshot.

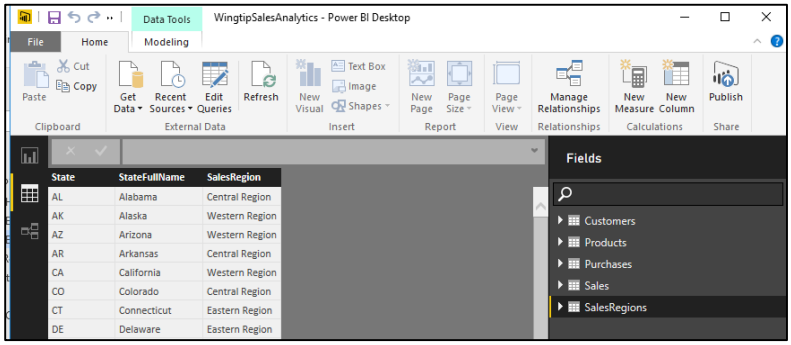



e) Click the **Load** button to load the data and to create the **SalesRegions** table. Power BI Desktop will display the **Load** dialog until the import process completed.

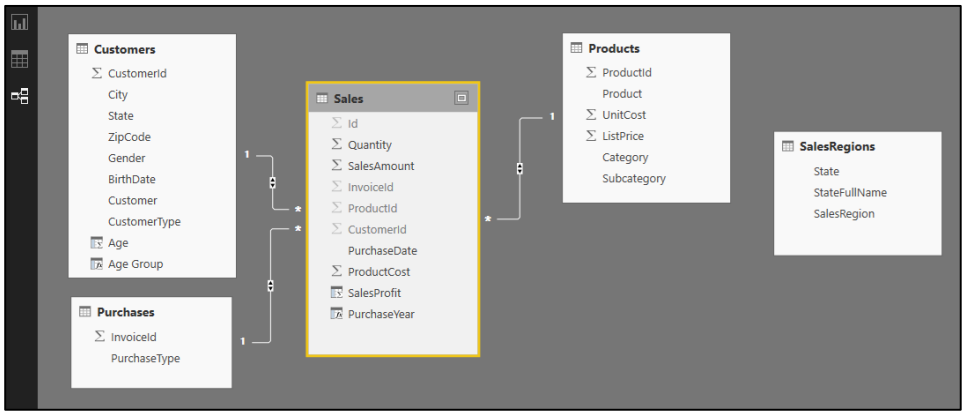



f) Once the import process completes, you should be able to see **SalesRegions** table in data view.

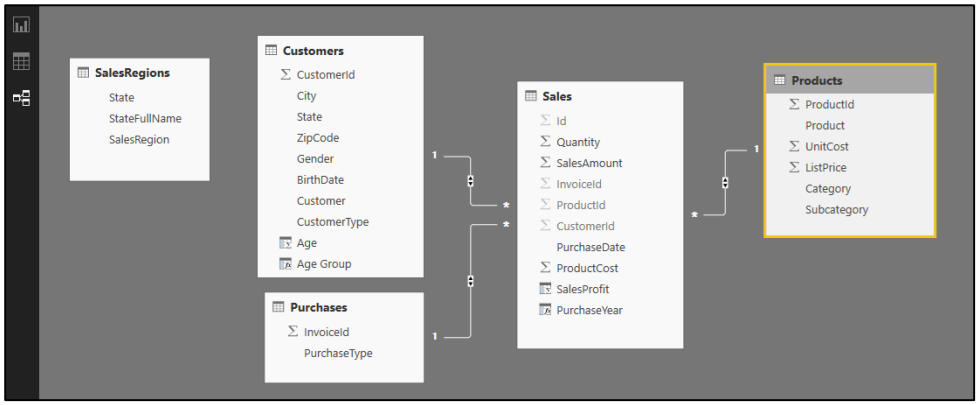
© Critical Path Training. 2015. All Rights Reserved




3. Define a relationship between the **Customers** table and the **SalesRegions** table.
   a) Navigate to relationship view.
   b) You should be able to see **SalesRegions** table. However, this table has been created without any relationships.

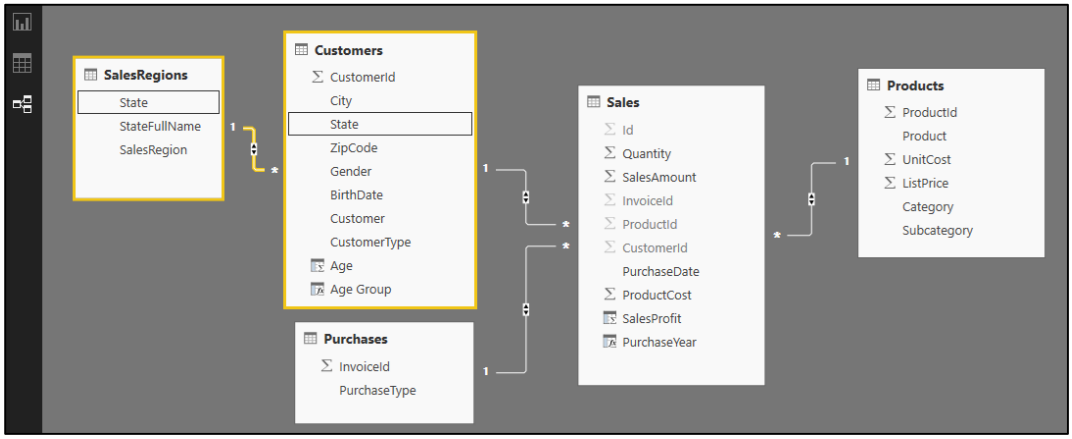



   c) Using your mouse, reorganize the layout of tables to match the following screenshot.




   d) Using your mouse, drag-and-drop the **State** column from the **Customers** table on top of the **State** column of the **SalesRegions** table to create a relationship between the two.

© Critical Path Training. 2015. All Rights Reserved

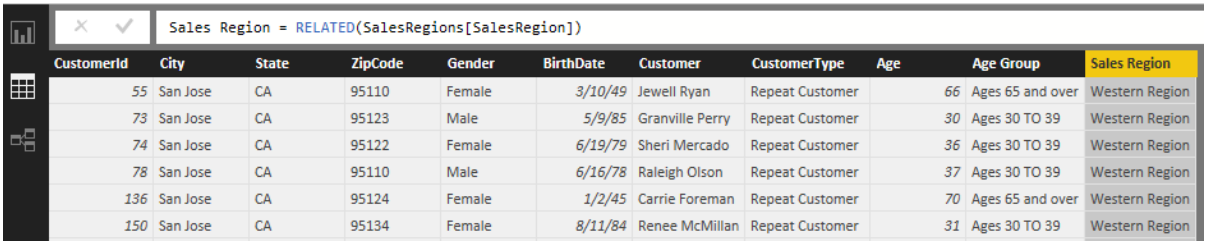



Now that you have created a relationship between the **SalesRegions** table and the **Customers** table, you can begin to use the **RELATED** function provided by DAX. More specifically, you can use the **RELATED** function to create calculated columns in the **Customers** table that are written to pull in data from the **SalesRegions** table.

4. Add a calculated column to the **Customers** table named **Sales Region** to display the sales region for each state.
   a) Navigate to data view.
   b) Select the **Customers** table in the **Fields** list.
   c) Create a new calculated column by clicking the **New Column** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the calculated column named **Sales Region**.

```
Sales Region = RELATED(SalesRegions[SalesRegion])
```

   e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a value in the **Sales Region** column for each row in the **Customers** table.



```
Sales Region = RELATED(SalesRegions[SalesRegion])
```

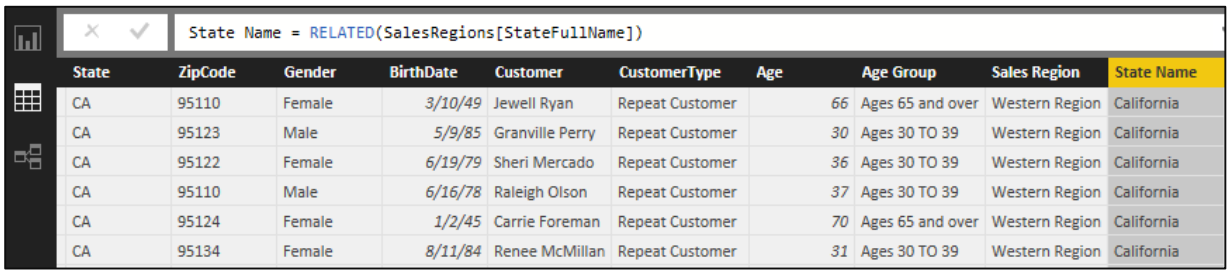
| CustomerId | City | State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age | Age Group | Sales Region |
|---|---|---|---|---|---|---|---|---|---|---|
| 55 | San Jose | CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 | Ages 65 and over | Western Region |
| 73 | San Jose | CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 | Ages 30 TO 39 | Western Region |
| 74 | San Jose | CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 | Ages 30 TO 39 | Western Region |
| 78 | San Jose | CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 | Ages 30 TO 39 | Western Region |
| 136 | San Jose | CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 | Ages 65 and over | Western Region |
| 150 | San Jose | CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 | Ages 30 TO 39 | Western Region |

5. Add a calculated column to the **Customers** table named **State Name** to display the full state name.
   a) Create a new calculated column in the **Customers** table by clicking the **New Column** button in the ribbon.
   b) Enter to following DAX expression into the formula bar to create the calculated column named **State Name**.

```
State Name = RELATED(SalesRegions[StateFullName])
```

   c) Press the **ENTER** key to add the calculated column to the table. You should be able to see a value in the **Sales Region** column for each row in the **Customers** table.

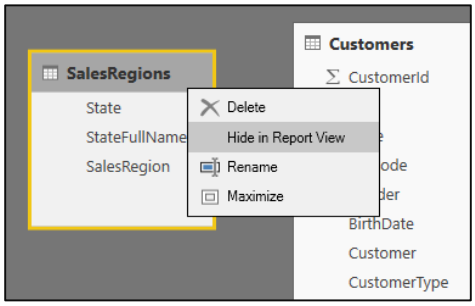
© Critical Path Training. 2015. All Rights Reserved



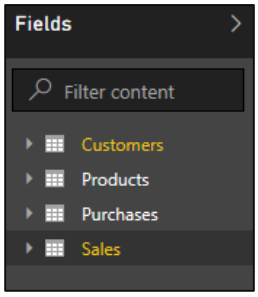

| State | ZipCode | Gender | BirthDate | Customer | CustomerType | Age | Age Group | Sales Region | State Name |
|---|---|---|---|---|---|---|---|---|---|
| CA | 95110 | Female | 3/10/49 | Jewell Ryan | Repeat Customer | 66 | Ages 65 and over | Western Region | California |
| CA | 95123 | Male | 5/9/85 | Granville Perry | Repeat Customer | 30 | Ages 30 TO 39 | Western Region | California |
| CA | 95122 | Female | 6/19/79 | Sheri Mercado | Repeat Customer | 36 | Ages 30 TO 39 | Western Region | California |
| CA | 95110 | Male | 6/16/78 | Raleigh Olson | Repeat Customer | 37 | Ages 30 TO 39 | Western Region | California |
| CA | 95124 | Female | 1/2/45 | Carrie Foreman | Repeat Customer | 70 | Ages 65 and over | Western Region | California |
| CA | 95134 | Female | 8/11/84 | Renee McMillan | Repeat Customer | 31 | Ages 30 TO 39 | Western Region | California |

Now that you have pulled all the important data from the **SalesRegions** table into the **Customers** table, there is no need to display the **SalesRegions** table in report view. In the next step, you will hide the **SaleRegions** table to simplify view of the data model that is shown in report view.

6. Hide the **SalesRegions** table from report view.
   a) Navigate to relationship view.
   b) Right-click on the the **SalesRegions** table

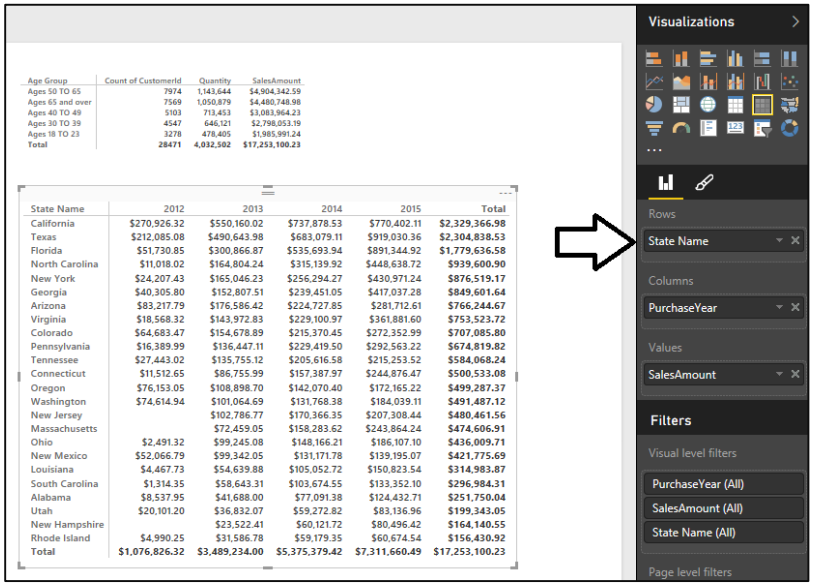



   c) Navigate to report view and verify that the **SalesRegions** table is not displayed as one of the tables in the **Fields** list.

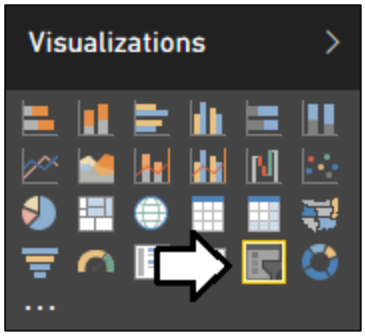



7. Update the matrix with row labels based on the **State** column to use the **State Name** column instead.
   a) Navigate back to report view.
   b) Select the matrix visual with the row labels based on the **State** column.
   c) Remove the **State** field from the **Rows** well in the **Visualization** pane and replace it with the **State Name** column. The visual should now show the full state name instead of the two-character state abbreviation.

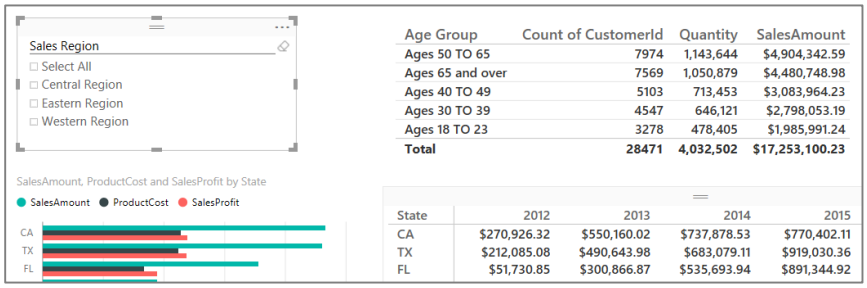
© Critical Path Training. 2015. All Rights Reserved




8. Add a slicer visual to the report to filter all displayed results by sales region.
   a) Navigate to report view if you are not already there.
   b) Make sure that no visuals are selected on the page so that you can create a new visual.
   c) Select the checkbox for the **Sales Region** column in the **Fields** list to create a new visual.
   d) Click the **Slicer** button in the **Visualizations** list to change the visual type to a slicer.

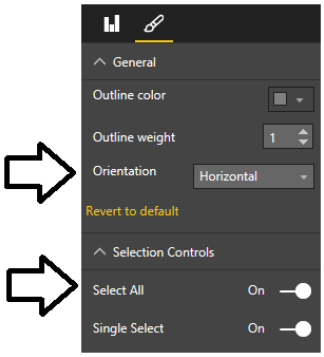



   e) Using the mouse, reposition the slicer visual so it appears in the upper, left-hand corner of the page as shown in the following screenshot.

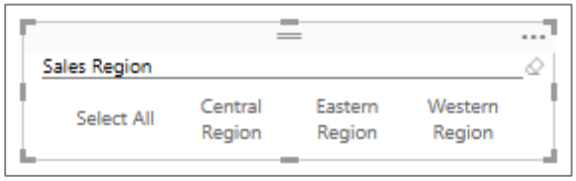



   f) Change the alignment of the slicer visual from vertical to horizontal. Accomplishing this by first clicking the Edit Pen button in the **Visualization** pane to view the appearance properties of the visual and then by modifying the **Orientation** property to a value of **Horizontal**. Also enable the **Select All** property by setting its value to **On**.

© Critical Path Training. 2015. All Rights Reserved




g) The slicer visual should now appear with a horizontal layout with an additional **Select All** node.




h) Click on the different nodes of the slicer visual to see its filtering effect. If you click on the **Eastern Region** node, all the other visuals apply a filter to only shows states assigned to the Eastern sales region.

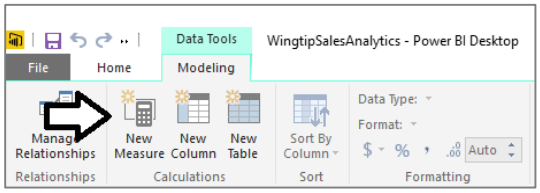
| State Name | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| Florida | $51,730.85 | $300,866.87 | $535,693.94 | $891,344.92 | $1,779,636.58 |
| North Carolina | $11,018.02 | $164,804.24 | $315,139.92 | $448,638.72 | $939,600.90 |
| New York | $24,207.43 | $165,046.23 | $256,294.27 | $430,971.24 | $876,519.17 |
| Georgia | $40,305.80 | $152,807.51 | $239,451.05 | $417,037.28 | $849,601.64 |
| Virginia | $18,568.32 | $143,972.83 | $229,100.97 | $361,881.60 | $753,523.72 |
| Pennsylvania | $16,389.99 | $136,447.11 | $229,419.50 | $292,563.22 | $674,819.82 |
| Connecticut | $11,512.65 | $86,755.99 | $157,387.97 | $244,876.47 | $500,533.08 |
| New Jersey | | $102,786.77 | $170,366.35 | $207,308.44 | $480,461.56 |
| Massachusetts | | $72,459.05 | $158,283.62 | $243,864.24 | $474,606.91 |
| South Carolina | $1,314.35 | $58,643.31 | $103,674.55 | $133,352.10 | $296,984.31 |
| New Hampshire | | $23,522.41 | $60,121.72 | $80,496.42 | $164,140.55 |
| Rhode Island | $4,990.25 | $31,586.78 | $59,179.35 | $60,674.54 | $156,430.92 |
| Total | $180,037.66 | $1,439,699.10 | $2,514,113.21 | $3,813,009.19 | $7,946,859.16 |

9. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 4: Extending the Data Model by Creating Measures

In this exercise you will create four measures named **Units Sold, Sales Revenue, Product Cost** and **Profit** that will perform sum aggregations on the **Sales** table. You will also create a measure named **Customer Count** that performs a distinct count aggregation on the **Customers** table. As you will see, creating measures to use in your report visuals will give you much greater control over the column names, formatting and aggregations that are displayed in your reports.

1. Create a measure named **Units Sold** to perform a sum aggregation on the **Quantity** column of the **Sales** table.
   a) Navigate to data view.
   b) Select the **Sales** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.

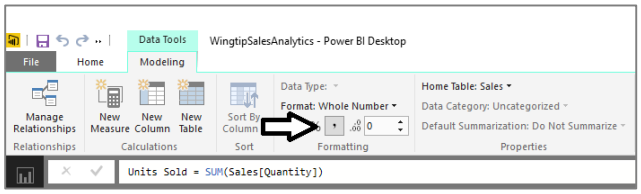


d) Enter to following DAX expression into the formula bar to create the calculated column named **Units Sold**.

```
Units Sold = SUM(Sales[Quantity])
```

e) Press the **ENTER** key to add the calculated column to data model.

f) Modify the formatting by clicking and selecting the Comma button on the ribbon to add a comma separator.




2. Create a measure named **Sales Revenue** to perform a sum aggregation on the **SalesAmount** column of the **Sales** table.

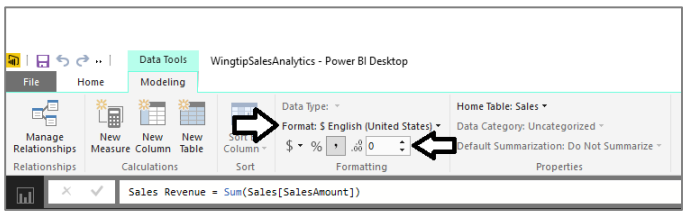
a) Create a new measure by clicking the **New Measure** button in the ribbon.

b) Enter to following DAX expression into the formula bar to create the calculated column named **Sales Revenue**.

```
Sales Revenue = Sum(Sales[SalesAmount])
```

c) Press the **ENTER** key to add the calculated column to data model.

d) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.




3. Create a measure named **Product Cost** to perform a sum aggregation on the **ProductCost** column of the **Sales** table.

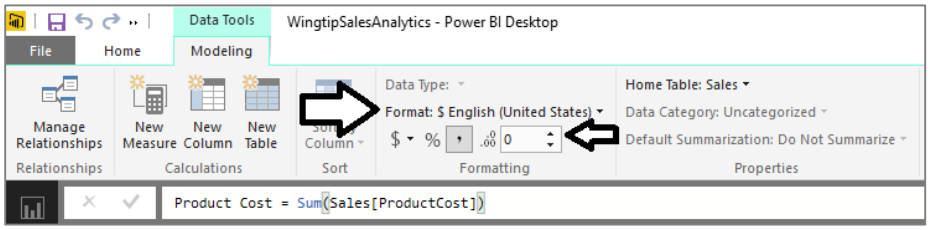
a) Create a new measure by clicking the **New Measure** button in the ribbon.

b) Enter to following DAX expression into the formula bar to create the calculated column named **Product Cost**.

```
Product Cost = Sum(Sales[ProductCost])
```

c) Press the **ENTER** key to add the calculated column to data model.

d) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.




4. Create a measure named **Profit** to perform a sum aggregation on the **SalesProfit** column of the **Sales** table.

a) Create a new measure by clicking the **New Measure** button in the ribbon.

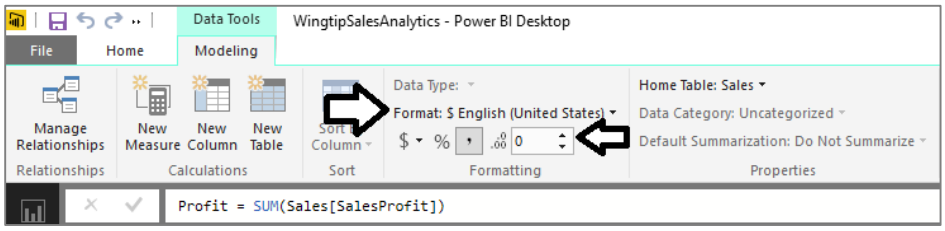
b) Enter to following DAX expression into the formula bar to create the calculated column named **Profit**.

```
Profit = SUM(Sales[SalesProfit])
```

c) Press the **ENTER** key to add the calculated column to data model.

© Critical Path Training. 2015. All Rights Reserved

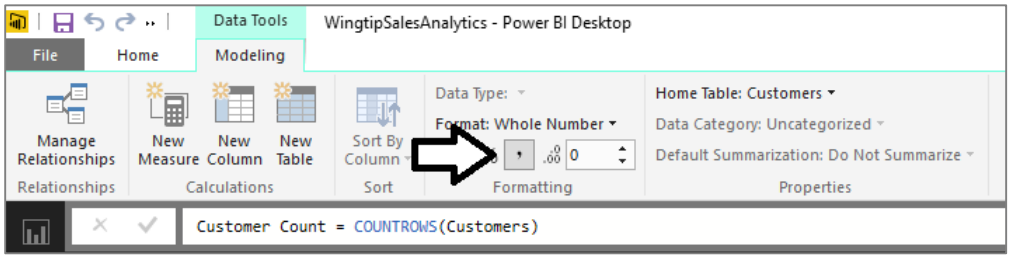
d) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.



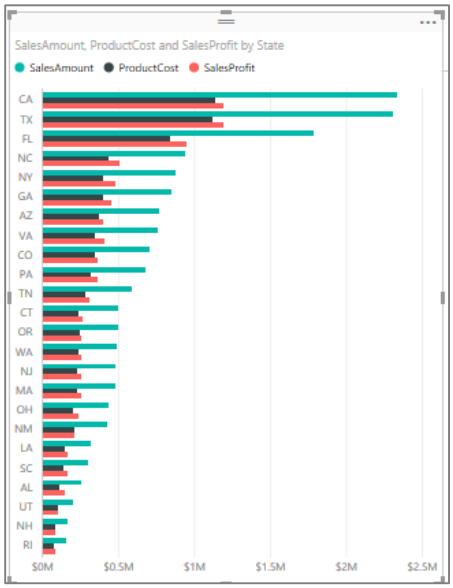

5. Create a measure named **Customer Count** to perform an aggregation to count the number of rows **in** the **Customers** table.
   a) Create a new measure by clicking the **New Measure** button in the ribbon.
   b) Enter to following DAX expression into the formula bar to create the calculated column named **Units Sold**.

```
Customer Count = COUNTROWS(Customers)
```

   c) Press the **ENTER** key to add the calculated column to data model.
   d) Modify the formatting by clicking and selecting the Comma button on the ribbon to add a comma separator.

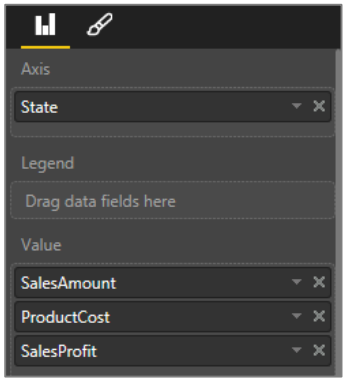



6. Update the visuals in the report to use the new measures you have just created.
   a) Navigate to report view.
   b) Select the clustered bar chart visual.

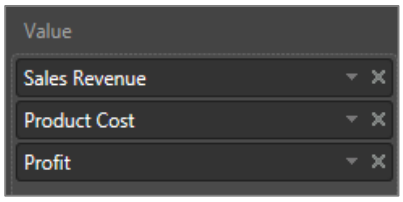



   c) Examine the properties of this visual in the **Visualizations** pane. The **Value** well should currently contain the three fields named **SalesAmount**, **ProductCost** and **SalesProfit**.

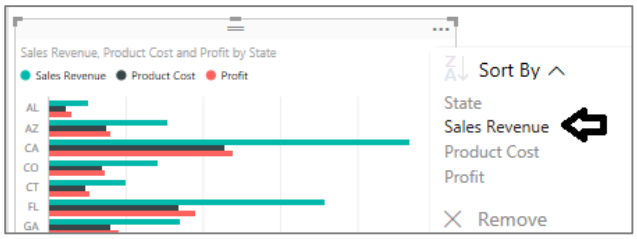
© Critical Path Training. 2015. All Rights Reserved



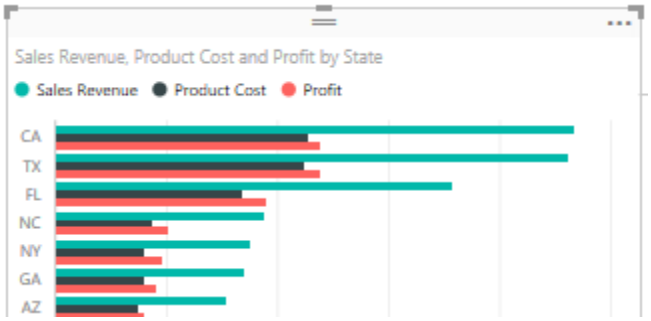

d) Remove those three fields from the **Value** well and replace them with the new measures named **Sales Revenue**, **Product Cost** and **Profit**.




e) You will notice that the sort order of this visual is now based on the **State** field because you removed the **SalesAmount** field that was being used for sorting. Resort the rows in the visual using **Sales Revenue**.



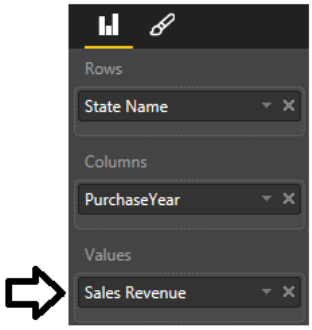

f) Now this visual looks more polished because it is using measures instead of aggregated columns to produce its values.




g) Select the matrix visual which displays sales revenue by state and year.

h) In the **Visualizations** pane, remove the **SalesAmount** column from the **Values** well and replace it with **Sales Revenue** measure.

© Critical Path Training. 2015. All Rights Reserved




i) The matrix visual should display its numbers in currency with no decimal places.

j) Click the **Total** column header to sort the state rows by total sales revenue.

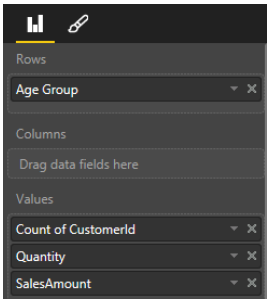
| State Name | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| California | $270,926 | $550,160 | $737,879 | $770,402 | $2,329,367 |
| Texas | $212,085 | $490,644 | $683,079 | $919,030 | $2,304,839 |
| Florida | $51,731 | $300,867 | $535,694 | $891,345 | $1,779,637 |
| North Carolina | $11,018 | $164,804 | $315,140 | $448,639 | $939,601 |
| New York | $24,207 | $165,046 | $256,294 | $430,971 | $876,519 |
| Georgia | $40,306 | $152,808 | $239,451 | $417,037 | $849,602 |
| Arizona | $83,218 | $176,586 | $224,728 | $281,713 | $766,245 |
| Virginia | $18,568 | $143,973 | $229,101 | $361,882 | $753,524 |

k) Select the matrix visual which displays a row for each age group.

l) Notice how some of the columns contain text (e.g. **Count of CustomerId**) that is not very user-friendly.

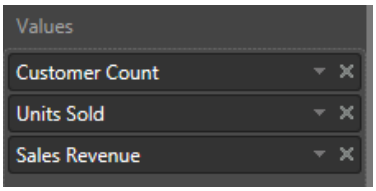
| Age Group | Count of CustomerId | Quantity | SalesAmount |
|---|---|---|---|
| Ages 50 TO 65 | 7974 | 1,143,644 | $4,904,342.59 |
| Ages 65 and over | 7569 | 1,050,879 | $4,480,748.98 |
| Ages 40 TO 49 | 5103 | 713,453 | $3,083,964.23 |
| Ages 30 TO 39 | 4547 | 646,121 | $2,798,053.19 |
| Ages 18 TO 23 | 3278 | 478,405 | $1,985,991.24 |
| **Total** | **28471** | **4,032,502** | **$17,253,100.23** |

m) Examine the properties of this visual in the **Visualizations** pane. The **Value** well should currently contain the three fields named **Count of CustomerId**, **Quantity** and **SalesAmount**.

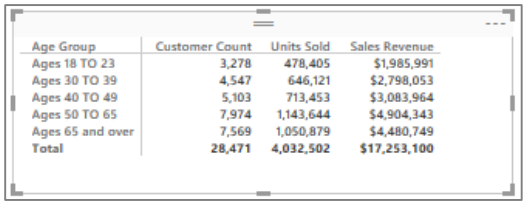



n) Remove those three fields from the **Value** well and replace them with the new measures named **Customer Count**, **Units Sold** and **Sales Revenue**.

© Critical Path Training. 2015. All Rights Reserved

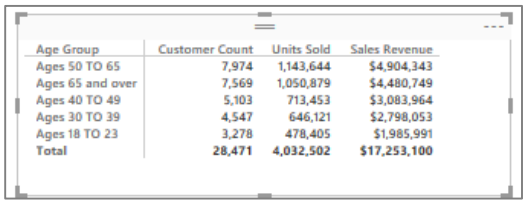



o) The columns of the matrix visual are now more user-friendly and the formatting of values looks better as well.



| Age Group | Customer Count | Units Sold | Sales Revenue |
|---|---|---|---|
| Ages 18 TO 23 | 3,278 | 478,405 | $1,985,991 |
| Ages 30 TO 39 | 4,547 | 646,121 | $2,798,053 |
| Ages 40 TO 49 | 5,103 | 713,453 | $3,083,964 |
| Ages 50 TO 65 | 7,974 | 1,143,644 | $4,904,343 |
| Ages 65 and over | 7,569 | 1,050,879 | $4,480,749 |
| **Total** | **28,471** | **4,032,502** | **$17,253,100** |

p) Click on the **Sales Revenue** column header to sort the age groups by sales revenue.



| Age Group | Customer Count | Units Sold | Sales Revenue |
|---|---|---|---|
| Ages 50 TO 65 | 7,974 | 1,143,644 | $4,904,343 |
| Ages 65 and over | 7,569 | 1,050,879 | $4,480,749 |
| Ages 40 TO 49 | 5,103 | 713,453 | $3,083,964 |
| Ages 30 TO 39 | 4,547 | 646,121 | $2,798,053 |
| Ages 18 TO 23 | 3,278 | 478,405 | $1,985,991 |
| **Total** | **28,471** | **4,032,502** | **$17,253,100** |

7. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

© Critical Path Training. 2015. All Rights Reserved

# Module 05: Extending Datasets to Support Time Intelligence

## Module Description

This module will begin by introducing concepts and terms used when modeling a dataset using time dimensions. Students will learn why and how to create a specialized date tables to track time dimensions as well as how to design a time-dimensional hierarchy which allows users to drill down into more specific regions of time while analyzing data. The second half of the module focuses on the support in the DAX language for creating calculated columns and measures within a dataset which provide rich analytic capabilities to calculate year-to-date totals and rolling averages and to provide drill down capabilities to the level of month, week or day.

## Module Agenda

- Understanding Evaluation Contexts
- Creating a Time Dimension Table
- Understanding DAX Time Intelligence Support
- Working with DAX Time Intelligence Functions

## Topics Covered

Evaluation Contexts
Understanding Filter Context
Understanding Row Context
Understanding Iterators Like SUMX
DAX Functions With Tables Parameter
DAX Functions that Return a Table
Using the CALCULATE Function
Creating Calendar Table as Calculated Table
Adding Columns to Calendar Table
Configuring Sort Columns
Columns for Month in Year and Day in week
Integrating Calendar Table into Data Model
Creating Visuals with a Calendar Table
Hierarchical Row Labels in a Matrix
Functions That Return a Single Date
To Date Functions
Function That Return a Table of Dates
Balance Functions
Calculated Fields for QTD and YTD Sales
Using the CALCULATE Function
Matrix Visual with To-Date Running Totals
Sales Growth PM Measure - First Attempt
Using the ISFILTERED Function
Simulating KPIs with Power BI Desktop

© Critical Path Training. 2015. All Rights Reserved

## Evaluation Contexts

- DAX expressions evaluated using two context
  - Filter context
  - Row context
- Row Context
  - The set of active rows in a calculation
- Filter Context
  - The current row in a table iteration

© Critical Path Training. 2015. All Rights Reserved

# Understanding Filter Context

- Visuals apply various filters in evaluation contexts

| Month in Year | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| January | $6,306 | $164,334 | $385,275 | $512,822 | $1,068,737 |
| February | $48,815 | $126,501 | $358,244 | $597,684 | $1,131,244 |
| March | $53,958 | $243,676 | $381,309 | $532,123 | $1,211,067 |
| April | $52,601 | $300,872 | $381,157 | $602,751 | $1,337,381 |
| May | $61,756 | $334,948 | $438,261 | $647,276 | $1,482,241 |
| June | $76,756 | $321,715 | $378,749 | $608,448 | $1,385,668 |
| July | $104,408 | $287,800 | $359,744 | $620,316 | $1,372,268 |
| August | $111,167 | $298,483 | $457,312 | $678,499 | $1,545,461 |
| September | $110,716 | $376,207 | $505,332 | $613,974 | $1,606,229 |
| October | $145,999 | $362,943 | $602,448 | $620,735 | $1,732,125 |
| November | $156,751 | $340,228 | $545,572 | $590,220 | $1,632,770 |
| December | $147,593 | $331,526 | $581,977 | $686,814 | $1,747,910 |
| Total | $1,076,826 | $3,489,234 | $5,375,379 | $7,311,660 | $17,253,100 |

**Filters on this evaluation**

[Year] = 2015

[Month in Year] = "October"

- Filter context also affected by slicers and other filters

Sales Region
- Select All
- Central Region
- Eastern Region
- Western Region

Customer Type
- One-time customer
- Repeat Customer

| Month in Year | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| January | $425 | $50,169 | $61,295 | $76,614 | $188,503 |
| February | $13,891 | $40,133 | $63,670 | $101,542 | $219,236 |
| March | $19,121 | $58,411 | $73,839 | $84,180 | $235,551 |
| April | $19,128 | $53,711 | $67,919 | $91,762 | $232,520 |
| May | $22,939 | $64,259 | $78,668 | $109,689 | $275,555 |
| June | $29,082 | $50,564 | $73,504 | $88,047 | $241,197 |
| July | $34,809 | $62,971 | $69,053 | $80,749 | $247,582 |
| August | $36,096 | $61,217 | $76,009 | $94,719 | $268,041 |
| September | $39,415 | $68,653 | $82,697 | $94,805 | $285,570 |
| October | $51,994 | $69,122 | $99,344 | $84,177 | $304,637 |
| November | $47,020 | $52,548 | $85,924 | $74,611 | $260,102 |
| December | $50,580 | $66,260 | $102,088 | $94,877 | $313,804 |
| Total | $364,500 | $698,018 | $934,009 | $1,075,771 | $3,072,298 |

**Filters on this evaluation**

[Year] = 2015

[Month in Year] = "October"

[Sales Region] = "Western Region"

[Customer Type] = "Repeat Customer"

© Critical Path Training. 2015. All Rights Reserved

# Understanding Row Context

- There are 2 cases that create active row context
  - Evaluation of a calculated column

```
=(TODAY()-[BirthDate])/365
```

```
=LEFT([ProductCategory], Find(" >", [ProductCategory]))
```

  - Evaluation of a DAX function that iterates over a table

```
=SUMX(RELATEDTABLE(Sales), Sales[SalesAmount])
```

```
=SUMX(Sales,
      Sales[Quantity] * RELATED(Products[UnitCost]))
```

© Critical Path Training. 2015. All Rights Reserved

# Understanding Iterators Like SUMX

- Standard aggregation functions (e.g. SUM) have no row context
  - You can sum an individual column but you cannot sum an expressions

```
Gross Margin = SUM( Sales[SalesAmount]-Sales[ProductCost] )
```

The SUM function only accepts a column reference as an argument.

- Iterator functions (e.g. SUMX) iterate through rows in a target table
  - First argument accepts an expressions evaluates to a table of rows
  - Second argument accepts expression that evaluated for each row in table

```
Gross Margin = SUMX(Sales, Sales[SalesAmount]-Sales[ProductCost] )
```

- Iterator functions can also be used in calculated columns

```
Product Sales Revenue = SUMX(RELATEDTABLE(Sales), Sales[SalesAmount])
```

| ProductId | Product | UnitCost | ListPrice | Category | Subcategory | Product Sales Revenue |
|---|---|---|---|---|---|---|
| 1 | Batman Action Figure | $6.85 | $14.95 | Action Figures | Tough Guys | $130,767.65 |
| 2 | Captain America Action Figure | $7.05 | $12.95 | Action Figures | Tough Guys | $499,714.6 |
| 3 | GI Joe Action Figure | $6.10 | $14.95 | Action Figures | Tough Guys | $168,905.1 |
| 4 | Green Hulk Action Figure | $2.85 | $9.95 | Action Figures | Tough Guys | $82,764.1 |
| 5 | Red Hulk Alter Ego Action Figure | $2.85 | $9.95 | Action Figures | Tough Guys | $13,691.2 |

© Critical Path Training. 2015. All Rights Reserved

## DAX Functions With Tables Parameter

- The following DAX functions create row context
  - AVERAGEX
  - COUNTAX
  - COUNTX
  - MAXX
  - MINX
  - SUMX

© Critical Path Training. 2015. All Rights Reserved

## DAX Functions that Return a Table

- ALL
- ALLEXCEPT
- CALCULATETABLE
- DISTINCT
- FILTER
- RELATEDTABLE
- VALUES

© Critical Path Training. 2015. All Rights Reserved

# Using the CALCULATE Function

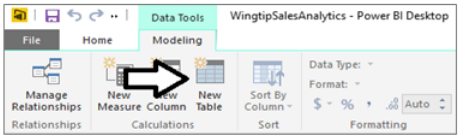
- CALCULATE provides greatest amount of control
  - You'll see more about Calculate later in this module

```
Sales Revenue RT =
CALCULATE(
    SUM(Sales[SalesAmount]),
    FILTER(
        ALL('Calendar'),
        'Calendar'[Date] <= MAX('Calendar'[Date])
    )
)
```

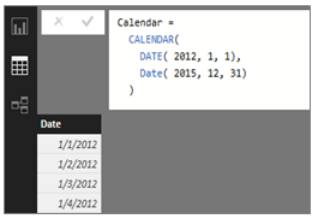
© Critical Path Training. 2015. All Rights Reserved

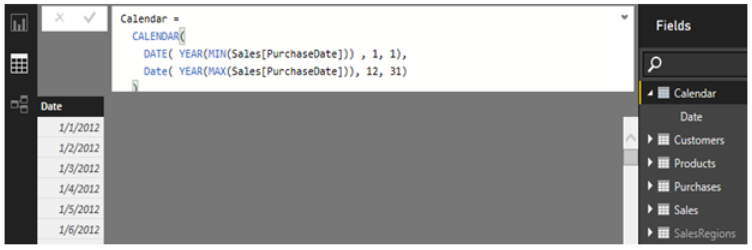
# Creating Calendar Table as Calculated Table

- Use **New Table** command in ribbon




- Create calculate table using CALENDAR function in DAX




© Critical Path Training. 2015. All Rights Reserved

# Adding Columns to Calendar Table

- Year

```
Year = YEAR('Calendar'[Date])
```

| Date | Year |
|---|---|
| 1/1/2012 | 2012 |
| 1/2/2012 | 2012 |
| 1/3/2012 | 2012 |

- Quarter

```
Quarter = YEAR('Calendar'[Date]) &
          "-Q" &
          QUOTIENT( (MONTH('Calendar'[Date])-1), 3) + 1
```

| Date | Year | Quarter |
|---|---|---|
| 3/28/2012 | 2012 | 2012-Q1 |
| 3/29/2012 | 2012 | 2012-Q1 |
| 3/30/2012 | 2012 | 2012-Q1 |
| 3/31/2012 | 2012 | 2012-Q1 |
| 4/1/2012 | 2012 | 2012-Q2 |
| 4/2/2012 | 2012 | 2012-Q2 |

- Month

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

| Date | Year | Quarter | Month |
|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 |

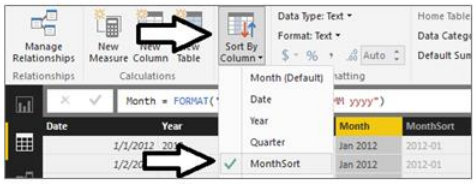
© Critical Path Training. 2015. All Rights Reserved

# Configuring Sort Columns

- Some columns do not sort in desired fashion by default
  - For example, April will sort before January, February and March
  - Column must be configured with sort column for desired sorting

```
MonthSort = FORMAT('Calendar'[Date], "yyyy-MM")
```

| Date | Year | Quarter | Month | MonthSort |
|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 |



# Columns for Month in Year and Day in week

```
Month in Year = FORMAT('Calendar'[Date], "MMMM")
```

| Date | Year | Quarter | Month | MonthSort | Month in Year |
|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January |

```
MonthInYearSort = MONTH('Calendar'[Date])
```

| Date | Year | Quarter | Month | MonthSort | Month in Year | MonthInYearSort |
|---|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 |

```
Day of Week = FORMAT('Calendar'[Date], "dddd")
```

| Date | Year | Quarter | Month | MonthSort | Month in Year | MonthInYearSort | Day of Week |
|---|---|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Sunday |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Monday |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Tuesday |
| 1/4/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Wednesday |
| 1/5/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Thursday |
| 1/6/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Friday |
| 1/7/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Saturday |

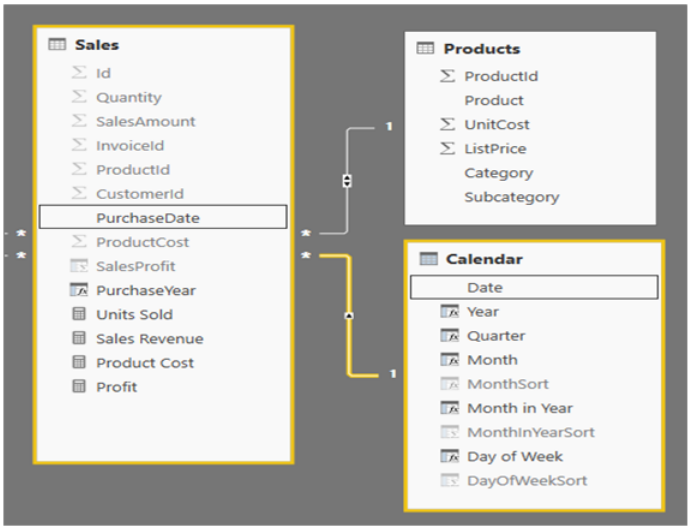
```
DayOfWeekSort = WEEKDAY('Calendar'[Date], 2)
```

| Date | Year | Quarter | Month | MonthSort | Month in Year | MonthInYearSort | Day of Week | DayOfWeekSort |
|---|---|---|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Sunday | 7 |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Monday | 1 |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Tuesday | 2 |
| 1/4/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Wednesday | 3 |
| 1/5/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Thursday | 4 |
| 1/6/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Friday | 5 |
| 1/7/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Saturday | 6 |

© Critical Path Training. 2015. All Rights Reserved

# Integrating Calendar Table into Data Model

- Calendar table needs relationship to one or more tables




© Critical Path Training. 2015. All Rights Reserved

# Creating Visuals with a Calendar Table

- **Year** for row labels and **Month in Year** as column labels

| Year | January | February | March | April | May | June | July | August | September | October | November | December | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2012 | $6,306 | $48,815 | $53,958 | $52,601 | $61,756 | $76,756 | $104,408 | $111,167 | $110,716 | $145,999 | $156,751 | $147,593 | **$1,076,826** |
| 2013 | $164,334 | $126,501 | $243,676 | $300,872 | $334,948 | $321,715 | $287,800 | $298,483 | $376,207 | $362,943 | $340,228 | $331,526 | **$3,489,234** |
| 2014 | $385,275 | $358,244 | $381,309 | $381,157 | $438,261 | $378,749 | $359,744 | $457,312 | $505,332 | $602,448 | $545,572 | $581,977 | **$5,375,379** |
| 2015 | $512,822 | $597,684 | $532,123 | $602,751 | $647,276 | $608,448 | $620,316 | $678,499 | $613,974 | $620,735 | $590,220 | $686,814 | **$7,311,660** |
| Total | **$1,068,737** | **$1,131,244** | **$1,211,067** | **$1,337,381** | **$1,482,241** | **$1,385,668** | **$1,372,268** | **$1,545,461** | **$1,606,229** | **$1,732,125** | **$1,632,770** | **$1,747,910** | **$17,253,100** |

- **Month in Year** for row labels and **Year** as column labels

| Month in Year | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| January | $6,306 | $164,334 | $385,275 | $512,822 | **$1,068,737** |
| February | $48,815 | $126,501 | $358,244 | $597,684 | **$1,131,244** |
| March | $53,958 | $243,676 | $381,309 | $532,123 | **$1,211,067** |
| April | $52,601 | $300,872 | $381,157 | $602,751 | **$1,337,381** |
| May | $61,756 | $334,948 | $438,261 | $647,276 | **$1,482,241** |
| June | $76,756 | $321,715 | $378,749 | $608,448 | **$1,385,668** |
| July | $104,408 | $287,800 | $359,744 | $620,316 | **$1,372,268** |
| August | $111,167 | $298,483 | $457,312 | $678,499 | **$1,545,461** |
| September | $110,716 | $376,207 | $505,332 | $613,974 | **$1,606,229** |
| October | $145,999 | $362,943 | $602,448 | $620,735 | **$1,732,125** |
| November | $156,751 | $340,228 | $545,572 | $590,220 | **$1,632,770** |
| December | $147,593 | $331,526 | $581,977 | $686,814 | **$1,747,910** |
| Total | **$1,076,826** | **$3,489,234** | **$5,375,379** | **$7,311,660** | **$17,253,100** |

- **Month in Year** for row labels and **Year** as column labels

| Day of Week | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| Monday | $163,919 | $486,428 | $802,943 | $1,090,519 | **$2,543,808** |
| Tuesday | $154,244 | $462,695 | $772,438 | $1,163,588 | **$2,552,965** |
| Wednesday | $162,811 | $425,869 | $752,274 | $1,029,697 | **$2,370,651** |
| Thursday | $169,687 | $513,948 | $837,905 | $1,066,526 | **$2,588,067** |
| Friday | $146,575 | $582,823 | $736,297 | $1,059,080 | **$2,524,774** |
| Saturday | $162,219 | $546,922 | $694,322 | $909,334 | **$2,312,797** |
| Sunday | $117,372 | $470,550 | $779,200 | $992,915 | **$2,360,038** |
| Total | **$1,076,826** | **$3,489,234** | **$5,375,379** | **$7,311,660** | **$17,253,100** |

© Critical Path Training. 2015. All Rights Reserved

# Hierarchical Row Labels in a Matrix

- Dimensional hierarchy can be represented with matrix
  - No support as of November 2016 to collapse and expand children

| Year | Quarter | Units Sold | Sales Revenue | Product Cost | Profit |
|---|---|---|---|---|---|
| 2012 | 2012-Q1 | 5,915 | $109,079 | $57,796 | $51,283 |
| | 2012-Q2 | 9,998 | $191,113 | $105,009 | $86,105 |
| | 2012-Q3 | 21,968 | $326,291 | $176,341 | $149,950 |
| | 2012-Q4 | 28,559 | $450,343 | $244,690 | $205,653 |
| | **Total** | **66,440** | **$1,076,826** | **$583,836** | **$492,991** |
| 2013 | 2013-Q1 | 67,144 | $534,511 | $274,583 | $259,928 |
| | 2013-Q2 | 313,031 | $957,536 | $401,557 | $555,979 |
| | 2013-Q3 | 276,121 | $962,490 | $420,217 | $542,273 |
| | 2013-Q4 | 296,759 | $1,034,697 | $451,751 | $582,946 |
| | **Total** | **953,055** | **$3,489,234** | **$1,548,108** | **$1,941,126** |
| 2014 | 2014-Q1 | 328,265 | $1,124,829 | $490,507 | $634,322 |
| | 2014-Q2 | 327,970 | $1,198,166 | $529,601 | $668,565 |
| | 2014-Q3 | 352,844 | $1,322,388 | $596,885 | $725,503 |
| | 2014-Q4 | 400,171 | $1,729,996 | $843,429 | $886,567 |
| | **Total** | **1,409,250** | **$5,375,379** | **$2,460,422** | **$2,914,957** |
| 2015 | 2015-Q1 | 380,321 | $1,642,629 | $803,675 | $838,954 |
| | 2015-Q2 | 423,115 | $1,858,474 | $912,540 | $945,934 |
| | 2015-Q3 | 420,095 | $1,912,789 | $946,017 | $966,772 |
| | 2015-Q4 | 380,226 | $1,897,768 | $958,635 | $939,133 |
| | **Total** | **1,603,757** | **$7,311,660** | **$3,620,868** | **$3,690,793** |
| **Total** | | **4,032,502** | **$17,253,100** | **$8,213,233** | **$9,039,867** |

© Critical Path Training. 2015. All Rights Reserved

## Functions That Return a Single Date

- FIRSTDATE / LASTDATE
- FIRSTNONBLANK / LASTNONBLANK
- STARTOFMONTH / ENDOFMONTH
- STARTOFQUARTER / ENDOFQUARTER
- STARTOFYEAR / ENDOFYEAR

© Critical Path Training. 2015. All Rights Reserved

## To Date Functions

- Convenience To Date functions
  - TOTALMTD
  - TOTALQTD
  - TOTALYTD
- Convenience To Date functions really do this
  - CALCULATE (Expression, DATESMTD(Date) [, SetFilter ] )
  - TOTALMTD (Expression, Date [, SetFilter ] )

© Critical Path Training. 2015. All Rights Reserved

## Function That Return a Table of Dates

- PREVIOUSDAY / NEXTDAY
- PREVIOUSMONTH / NEXTMONTH
- PREVIOUSQUARTER / NEXTQUARTER
- PREVIOUSYEAR / NEXTYEAR
- DATESMTD / DATESQTD / DATESYTD
- SAMEPERIODLASTYEAR
- DATEADD
- DATESBETWEEN
- DATESINPERIOD
- PARALLELPERIOD

© Critical Path Training. 2015. All Rights Reserved

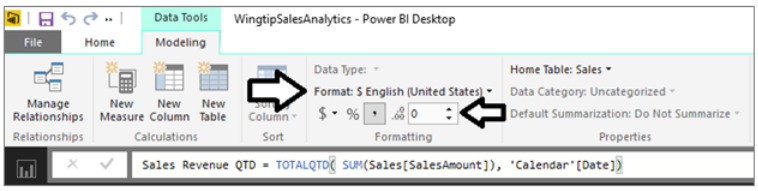
## Balance Functions

- DAX Functions return value at specific point in time
  - OPENINGBALANCEMONTH
  - OPENINGBALANCEQUARTER
  - OPENINGBALANCEYEAR
  - CLOSINGBALANCEMONTH
  - CLOSINGBALANCEQUARTER
  - CLOSINGBALANCEYEAR

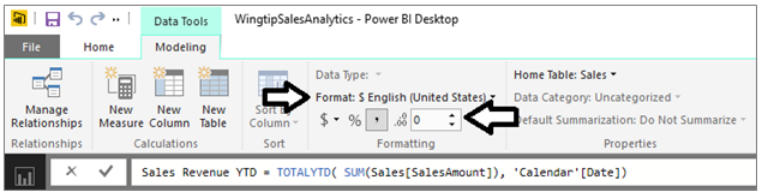
© Critical Path Training. 2015. All Rights Reserved

# Calculated Fields for QTD and YTD Sales

- TOTALQTD function calculates quarter-to-date totals



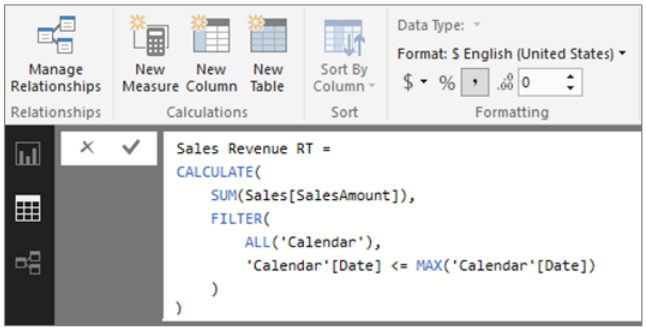

- TOTALYTD function calculates year-to-date totals




© Critical Path Training. 2015. All Rights Reserved

# Using the CALCULATE Function

- Calculate a running total of sales revenue across years




© Critical Path Training. 2015. All Rights Reserved

# Matrix Visual with To-Date Running Totals

- Running totals calculated using DAX

| Year | Quarter | Month | Sales Revenue | Sales Revenue QTD | Sales Revenue YTD | Sales Revenue RT |
|---|---|---|---|---|---|---|
| 2012 | 2012-Q1 | Jan 2012 | $6,306 | $6,306 | $6,306 | $6,306 |
| | | Feb 2012 | $48,815 | $55,121 | $55,121 | $55,121 |
| | | Mar 2012 | $53,958 | $109,079 | $109,079 | $109,079 |
| | 2012-Q2 | Apr 2012 | $52,601 | $52,601 | $161,680 | $161,680 |
| | | May 2012 | $61,756 | $114,357 | $223,436 | $223,436 |
| | | Jun 2012 | $76,756 | $191,113 | $300,192 | $300,192 |
| | 2012-Q3 | Jul 2012 | $104,408 | $104,408 | $404,600 | $404,600 |
| | | Aug 2012 | $111,167 | $215,575 | $515,767 | $515,767 |
| | | Sep 2012 | $110,716 | $326,291 | $626,483 | $626,483 |
| | 2012-Q4 | Oct 2012 | $145,999 | $145,999 | $772,483 | $772,483 |
| | | Nov 2012 | $156,751 | $302,750 | $929,234 | $929,234 |
| | | Dec 2012 | $147,593 | $450,343 | $1,076,826 | $1,076,826 |
| 2013 | 2013-Q1 | Jan 2013 | $164,334 | $164,334 | $164,334 | $1,241,161 |
| | | Feb 2013 | $126,501 | $290,835 | $290,835 | $1,367,661 |

- Question: when did company reach $1,000,000 in sales

| | | | | |
|---|---|---|---|---|
| Nov 2012 | $156,751 | $302,750 | $929,234 | $929,234 |
| Dec 2012 | $147,593 | $450,343 | $1,076,826 | $1,076,826 |
| Jan 2013 | $164,334 | $164,334 | $164,334 | $1,241,161 |
| Feb 2013 | $126,501 | $290,835 | $290,835 | $1,367,661 |

© Critical Path Training. 2015. All Rights Reserved

# Sales Growth PM Measure - First Attempt

- Create a measure named Sales Growth PM

```
Sales Growth PM =
DIVIDE(
  SUM(Sales[SalesAmount]) -
  CALCULATE(
    SUM(Sales[SalesAmount]),
    PREVIOUSMONTH(Calendar[Date])
  ),
  CALCULATE(
    SUM(Sales[SalesAmount]),
    PREVIOUSMONTH(Calendar[Date])
  )
)
```

- Use measure in matrix evaluating month and quarter
  - Measure returns correct value for Month
  - Measure returns incorrect and erroneous value for Quarter

| Year | Quarter | Month | Sales Revenue | Sales Growth PM |
|---|---|---|---|---|
| 2015 | 2015-Q1 | Jan 2015 | $512,822 | -11.88 % |
| | | Feb 2015 | $597,684 | 16.55 % |
| | | Mar 2015 | $532,123 | -10.97 % |
| | | **Total** | **$1,642,629** | **182.25 %** |
| | 2015-Q2 | Apr 2015 | $602,751 | 13.27 % |
| | | May 2015 | $647,276 | 7.39 % |
| | | Jun 2015 | $608,448 | -6.00 % |
| | | **Total** | **$1,858,474** | **249.26 %** |
| | 2015-Q3 | Jul 2015 | $620,316 | 1.95 % |
| | | Aug 2015 | $678,499 | 9.38 % |

© Critical Path Training. 2015. All Rights Reserved

# Using the ISFILTERED Function

- ISFILTERED function used to determine when perform evaluation

```
Sales Growth PM = IF(
  ( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() ),
  DIVIDE(
    SUM(Sales[SalesAmount]) -
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    ),
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    )
  ),
  BLANK()
)
```

- Expression returns Blank when evaluation context isn't appropriate

| Year | Quarter | Month | Sales Revenue | Sales Growth PM |
|---|---|---|---|---|
| 2015 | 2015-Q1 | Jan 2015 | $512,822 | -11.88 % |
| | | Feb 2015 | $597,684 | 16.55 % |
| | | Mar 2015 | $532,123 | -10.97 % |
| | | **Total** | **$1,642,629** | |
| | 2015-Q2 | Apr 2015 | $602,751 | 13.27 % |
| | | May 2015 | $647,276 | 7.39 % |
| | | Jun 2015 | $608,448 | -6.00 % |
| | | **Total** | **$1,858,474** | |
| | 2015-Q3 | Jul 2015 | $620,316 | 1.95 % |
| | | Aug 2015 | $678,499 | 9.38 % |

© Critical Path Training. 2015. All Rights Reserved

# Simulating KPIs with Power BI Desktop

- KPIs are not yet support
  - But you can create something similar using measures

```
Sales Growth PM Eval =
IF( ISNUMBER([Sales Growth PM]),
    SWITCH(TRUE(),
      ([Sales Growth PM] >= 0.2), "EXCELLENT",
      ([Sales Growth PM] >= 0.1), "GOOD",
      ([Sales Growth PM] >= 0), "OK",
      ([Sales Growth PM] <  0), "BAD"
  ))
```

```
Sales Growth PQ Eval =
IF( ISNUMBER([Sales Growth PQ]),
    SWITCH(TRUE(),
      ([Sales Growth PQ] >= 0.2), "EXCELLECT",
      ([Sales Growth PQ] >= 0.1), "GOOD",
      ([Sales Growth PQ] >= 0), "OK",
      ([Sales Growth PQ] <  0), "BAD"
  )
)
```

| Year | Quarter | Month | Sales Revenue | Sales Growth PM | Sales Growth PM Eval | Sales Growth PQ | Sales Growth PQ Eval |
|---|---|---|---|---|---|---|---|
| 2014 | 2014-Q1 | Jan 2014 | $385,275 | 16.21 % | GOOD | | |
| | | Feb 2014 | $358,244 | -7.02 % | BAD | | |
| | | Mar 2014 | $381,309 | 6.44 % | OK | | |
| | | **Total** | **$1,124,829** | | | **8.71 %** | **OK** |
| | 2014-Q2 | Apr 2014 | $381,157 | -0.04 % | BAD | | |
| | | May 2014 | $438,261 | 14.98 % | GOOD | | |
| | | Jun 2014 | $378,749 | -13.58 % | AWFUL | | |
| | | **Total** | **$1,198,166** | | | **6.52 %** | **OK** |
| | 2014-Q3 | Jul 2014 | $359,744 | -5.02 % | BAD | | |
| | | Aug 2014 | $457,312 | 27.12 % | EXCELLENT | | |
| | | Sep 2014 | $505,332 | 10.50 % | GOOD | | |
| | | **Total** | **$1,322,388** | | | **10.37 %** | **GOOD** |
| | 2014-Q4 | Oct 2014 | $602,448 | 19.22 % | GOOD | | |
| | | Nov 2014 | $545,572 | -9.44 % | BAD | | |
| | | Dec 2014 | $581,977 | 6.67 % | OK | | |
| | | **Total** | **$1,729,996** | | | **30.82 %** | **EXCELLENT** |

© Critical Path Training. 2015. All Rights Reserved

# Module 05 Lab: Extending a Data Model with Time Intelligence

**Lab Time**: 60 minutes

**Lab Folder**: C:\Student\Modules\TimeIntelligence\Lab

**Lab Overview**: In this lab, you will continue to work on the Power BI Desktop project you have been working with over the last two previous labs. At this point, you have already spent time designing the data model for Wingtip sales data that has been imported from a SQL Azure database. In this lab you will continue to extend the data model by adding a time dimension table named **Calendar** and by configuring this table so it can be used for time-based financial analysis. This will allow you to create measures using the DAX time intelligence functions.

**Important:** This lab assumes you have completed the previous lab titled **Using the Power Pivot Features of Power BI Desktop** in which you created a Power BI Desktop project named **WingtipSalesAnalytics.pbix**. If you would like to begin work on this lab without completing the earlier lab, copy the lab solution file at **C:\Student\Modules\PowerPivot\LabSolution\WingtipSalesAnalytics.pbix** into the folder at **C:\Student\Projects** using the Windows Explorer.

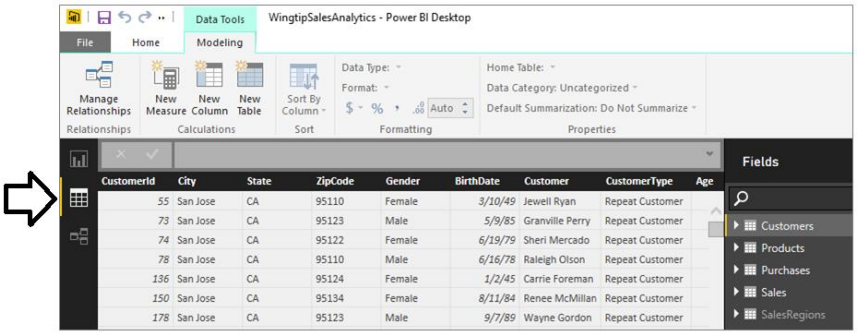
## Exercise 1: Adding a Calendar Dimension Table

In this exercise you will create a calculated table named **Calendar** which will play the role of a time dimension table in the data model. The motivation for adding a time dimension table to your data model is that it will allow you to take advantage of the time intelligence support that is built into Power Pivot and DAX.

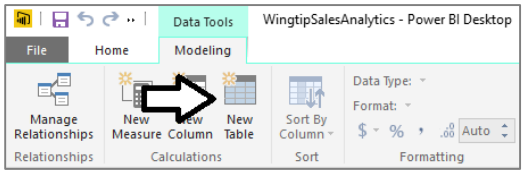
1. Launch Power BI Desktop to start a new project.
2. Open the Power BI Desktop project named **WingtipSalesAnalytics.pbix** from the previous lab located at the following path.

```
C:\Student\Projects\WingtipSalesAnalytics.pbix
```

3. When the project opens, click the table icon in the middle of the sidebar to enter data view mode.




4. Create a new calculated table named **Calendar**.
   a) Click the **New Table** button in the ribbon.

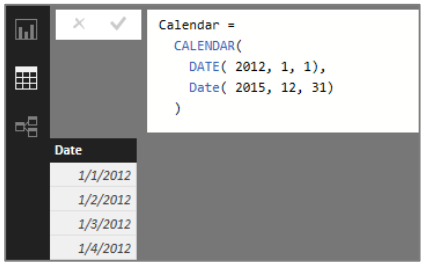



   b) Type the following DAX formula into the formula bar and press the **Enter** key to create the **Calendar** table.

```
Calendar =
  CALENDAR(
    DATE( 2012, 1, 1),
    Date( 2015, 12, 31)
  )
```

   c) You should be able to verify that the **Calendar** table has been created with a single column named **Date**. You should also be and to verify that there is one row in the **Calendar** table for each day in the calendar years of 2012, 2013, 2014 and 2015.

© Critical Path Training. 2015. All Rights Reserved




d) Modify the DAX expression for this calculated table so you do not have to hardcode literal values for the starting year or the ending year. First, replace the literal value for the starting year 2012 with the following DAX expression.

```
YEAR( MIN(Sales[PurchaseDate]) )
```

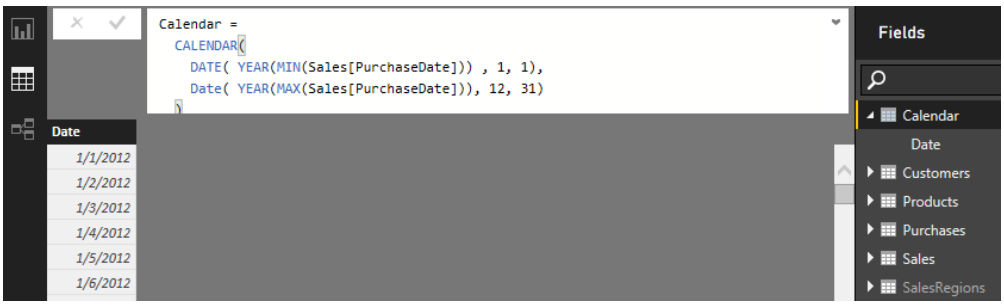
e) Next, replace the literal value for the ending year 2015 with the following DAX expressions.

```
YEAR( MAX(Sales[PurchaseDate]) )
```

f) At this point, your DAX expression to create the **Calendar** table should now match the following code listing.

```
Calendar =
  CALENDAR(
    DATE( YEAR( MIN(Sales[PurchaseDate]) ), 1, 1),
    Date( YEAR( MAX(Sales[PurchaseDate]) ), 12, 31)
  )
```

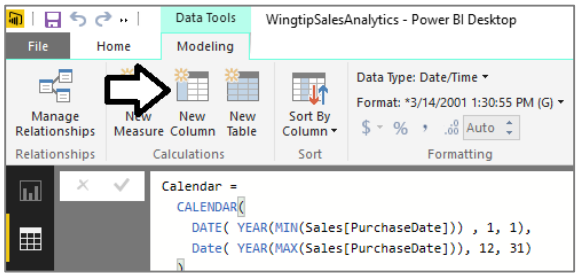
g) The **Calendar** table should continue to look as it did before starting at **1/1/20012** and running to **12/31/2015**.




The modification to the DAX expression to calculate the start date and end date dynamically provides flexibility. The new expression will now automatically add new rows for complete years if the **Sales** table is ever updated with purchases that occurred either before the start of 2012 or after the end of 2015.

5. Add a new column to the **Calendar** table named **Year** which display the financial year.

a) Create a new calculated column by clicking the **New Column** button in the ribbon.



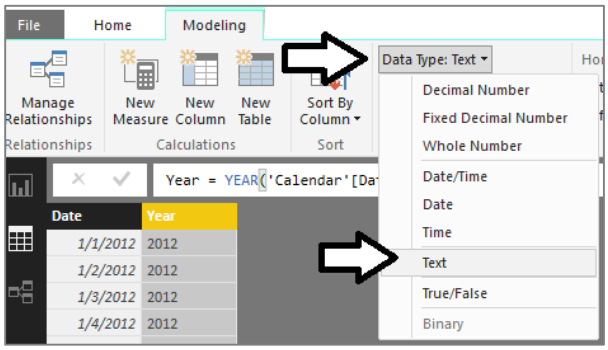

b) Type in the following DAX expression and press the Enter key.

```
Year = YEAR('Calendar'[Date])
```

© Critical Path Training. 2015. All Rights Reserved

c) You should see that the **Calendar** table now contains a **Year** column displaying the year.

d) Configure the **Data Type** of the **Year** column to be **Text**.



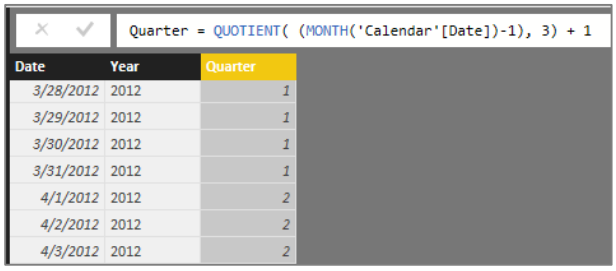

6. Add a new column to the **Calendar** table named **Quarter** which display the financial year and quarter.

a) Create a new calculated column by clicking the **New Column** button in the ribbon.

b) Type in the following DAX expression and press the Enter key.

```
Quarter = QUOTIENT( (MONTH('Calendar'[Date])-1), 3) + 1
```

c) The **Quarter** column now returns an integer value between 1 and 4 depending on what quarter the **Date** column value lies in.

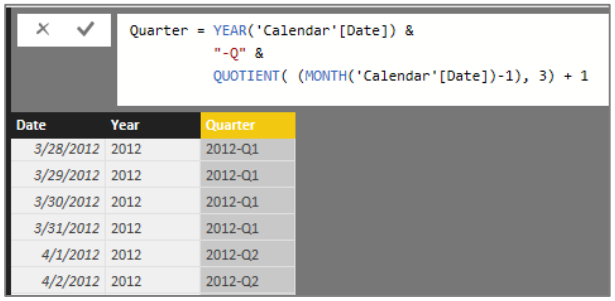



d) Modify the DAX expression for the **Quarter** column to include the year and a bit more text formatting by typing in the following DAX expression and pressing the **Enter** key.

```
Quarter = YEAR('Calendar'[Date]) &
          "-Q" &
          QUOTIENT( (MONTH('Calendar'[Date])-1), 3) + 1
```

e) Now the **Quarter** column should return a value for each financial quarter in the format of **2012-Q1**.

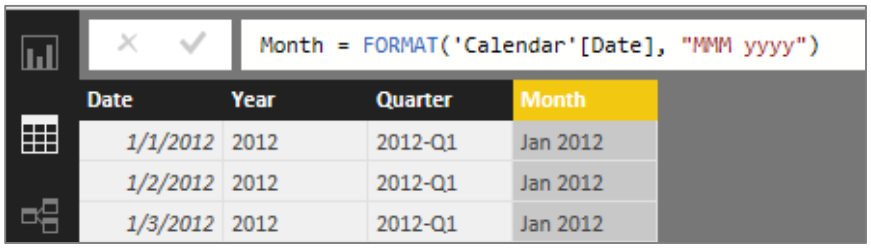
© Critical Path Training. 2015. All Rights Reserved




7. Add a new column to the **Calendar** table named **Month** which display the financial year and month.
   a) Create a new calculated column by clicking the **New Column** button in the ribbon.
   b) Type in the following DAX expression and press the Enter key.

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

   c) The column should now display

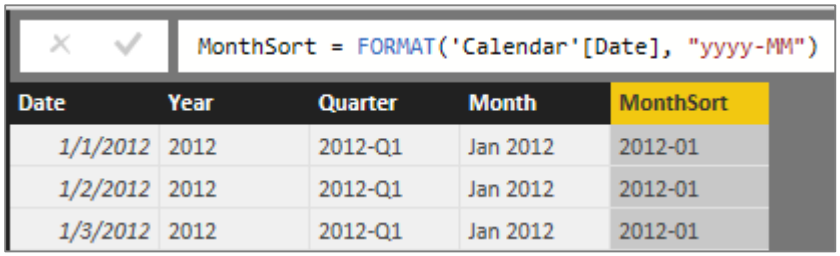



The **Month** column is a good example of a column whose value will not automatically be sorted in the chronological sort order. The default sort order of a text column like **Month** is to sort month names alphabetically so that April will sort before February, and February will sort before January. Therefore, you will now create an addition column named **MonthSort** whose sole purpose will be to provide assistance to the **Month** column to sort its values chronologically.

8. Add a new sort column to the **Calendar** table named **MonthSort** to control the sort order of the **Month** column.
   a) Create a new calculated column by clicking the **New Column** button in the ribbon.
   b) Type in the following DAX expression and press the Enter key.

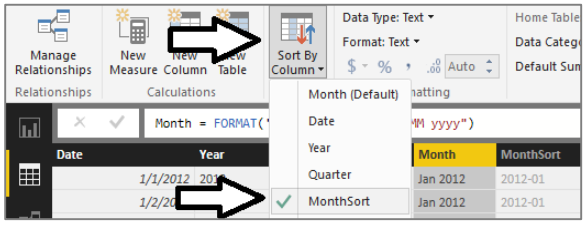
```
MonthSort = FORMAT('Calendar'[Date], "yyyy-MM")
```

   c) You should be able to see that the **MonthSort** column produces a text value for each date in the format of **2012-01**. The key aspect of this format is that **MonthSort** values are sorted chronologically when they are sorted alphabetically.

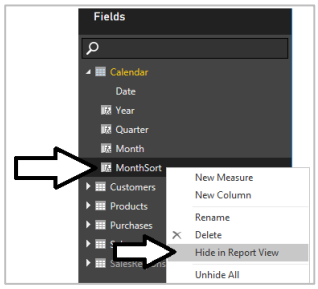



   d) Configure the **Month** column to use the **MonthSort** column as its sort column. Accomplish this by clicking the column header of the **Month** column to select it and then by dropping down the **Sort By Column** menu button in the ribbon and selecting the **MonthSort** column.

© Critical Path Training. 2015. All Rights Reserved



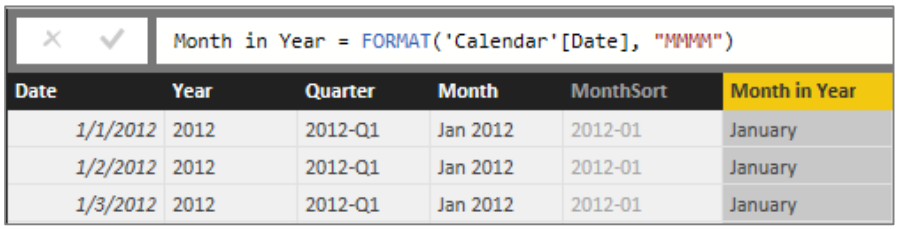

e) Hide the **MonthSort** column by right-clicking it on the **Fields** list and selecting the **Hide in Report View** menu command.




9. Add a new column to the **Calendar** table named **Month in Year** to display the financial month without the year.
   a) Create a new calculated column by clicking the **New Column** button in the ribbon.
   b) Type in the following DAX expression and press the Enter key.

```
Month in Year = FORMAT('Calendar'[Date], "MMMM")
```

   c) The **Month in Year** column should now display the month name for each date.



Month in Year = FORMAT('Calendar'[Date], "MMMM")

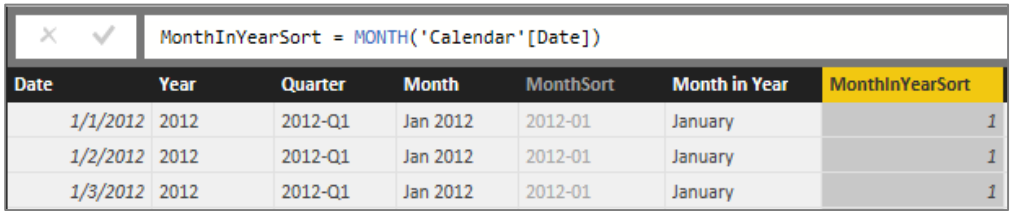
| Date | Year | Quarter | Month | MonthSort | Month in Year |
|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January |

Just like the **Month** column, the **Month in Year** column will need the assistance of a sort column.

10. Add a new sort column to the **Calendar** table named **MonthInYearSort** to control the sort order of the **Month in Year** column.
   a) Create a new calculated column by clicking the **New Column** button in the ribbon.
   b) Type in the following DAX expression and press the Enter key.

```
MonthInYearSort = MONTH('Calendar'[Date])
```

   c) As you can see, the **MonthInYearSort** column displays an integer value between 1 and 12 to indicate the month.

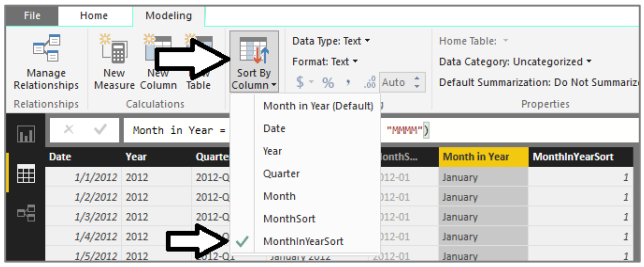


MonthInYearSort = MONTH('Calendar'[Date])

| Date | Year | Quarter | Month | MonthSort | Month in Year | MonthInYearSort |
|---|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 |

© Critical Path Training. 2015. All Rights Reserved

d) Configure the **Month in Year** column to use the **MonthInYearSort** column as its sort column. Accomplish this by clicking the column header of the **Month in Year** column to select it and then by dropping down the **Sort By Column** menu button in the ribbon and selecting the **MonthInYearSort** column.




e) Hide the **MonthInYearSort** column by right-clicking it in the **Fields** list and selecting the **Hide in Report View** command.

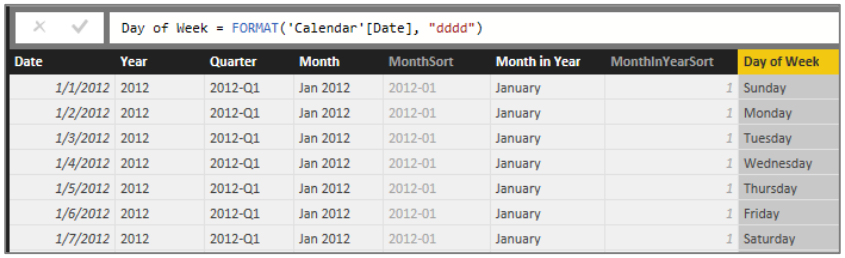
11. Add a new column to the **Calendar** table named **Day of Week** which display the day of the week.

a) Create a new calculated column by clicking the **New Column** button in the ribbon.

b) Type in the following DAX expression and press the Enter key.

```
Day of Week = FORMAT('Calendar'[Date], "dddd")
```

c) The **Day of Week** column should now display the name of the day (e.g. Monday) for each date.



Day of Week = FORMAT('Calendar'[Date], "dddd")

| Date | Year | Quarter | Month | MonthSort | Month in Year | MonthInYearSort | Day of Week |
|---|---|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Sunday |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Monday |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Tuesday |
| 1/4/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Wednesday |
| 1/5/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Thursday |
| 1/6/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Friday |
| 1/7/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Saturday |

12. Add a new sort column to the **Calendar** table named **DayOfWeekSort** to control the sort order of the **Day of Week** column.

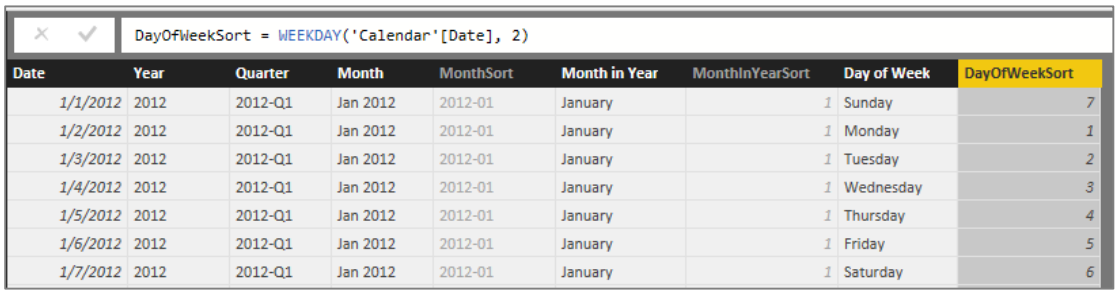
a) Create a new calculated column by clicking the **New Column** button in the ribbon.

b) Type in the following DAX expression and press the Enter key.

```
DayOfWeekSort = WEEKDAY('Calendar'[Date], 2)
```

The second argument passes to **WEEKDAY** function determines the starting day for the week. If you pass a value of 1, the starting day for the week will be Sunday. In this case you have passed a value of 2 which will make Monday the first day of the week.

c) Now the **DayOfWeekSort** column should return an integer value for each date indicating the day of the week. As you can see, each date that is a Monday has a value of 1.

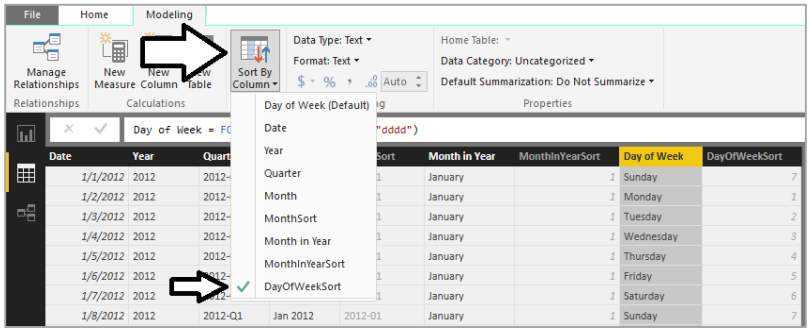


DayOfWeekSort = WEEKDAY('Calendar'[Date], 2)

| Date | Year | Quarter | Month | MonthSort | Month in Year | MonthInYearSort | Day of Week | DayOfWeekSort |
|---|---|---|---|---|---|---|---|---|
| 1/1/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Sunday | 7 |
| 1/2/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Monday | 1 |
| 1/3/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Tuesday | 2 |
| 1/4/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Wednesday | 3 |
| 1/5/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Thursday | 4 |
| 1/6/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Friday | 5 |
| 1/7/2012 | 2012 | 2012-Q1 | Jan 2012 | 2012-01 | January | 1 | Saturday | 6 |

d) Configure the **Day of Week** column to use the **DayInWeekSort** column as its sort column. Accomplish this by clicking the column header of the **Day of Week** column to select it and then by dropping down the **Sort By Column** menu button in the ribbon and selecting the **DayInWeekSort** column.

© Critical Path Training. 2015. All Rights Reserved



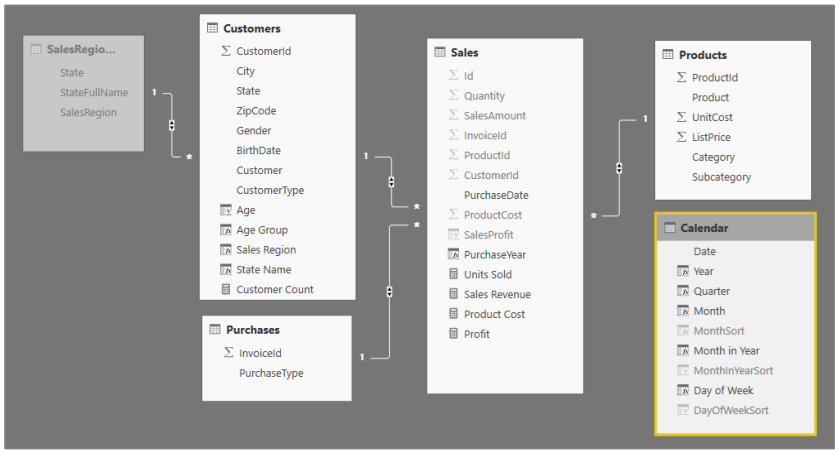

e) Hide the **DayInWeekSort** column by right-clicking it in the **Fields** list and selecting the **Hide in Report View** command.

At this point, you have created the **Calendar** table and added all the columns that it requires. The next step to integrate the **Calendar** table into the data model will be to create a relationship between the **Calendar** table and the **Sales** table.

13. Create a relationship between the **Calendar** table and the **Sales** table.

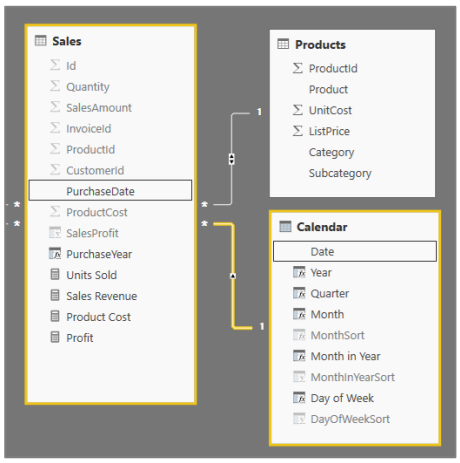
a) Navigate to relationship view. You should be able to see that the **Calendar** table is present. You should also be able to verify that the **Calendar** table does not have any existing relationships.

b) Using the mouse, rearrange the tables in relationship view to match the following screenshot where the **Calendar** table is positioned directly below the **Products** table and to the immediate right of the **Sales** table.




c) Create a new relationship by performing a drag and drop operation with the mouse to drag the **PurchaseDate** column from the **Sales** table and dropping it on the **Date** column of the **Calendar** table.

© Critical Path Training. 2015. All Rights Reserved



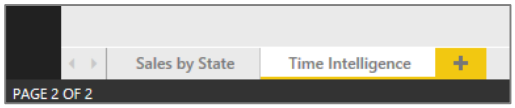

You have now completed the work of adding the **Calendar** table to the data model.

14. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

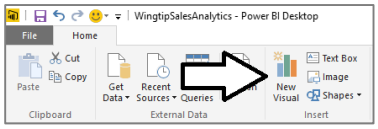
## Exercise 2: Creating Visuals using the Calendar Table

In this exercise, you will leverage the **Calendar** table that you created in the previous exercise by creating a few new visuals to display sales revenue totals aggregated over various time intervals.

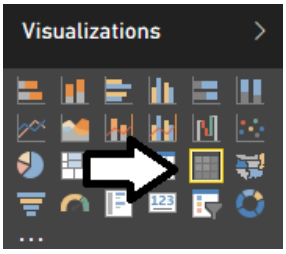
1. Create a new page in the project's report.
   a) Navigate to report view.
   b) Add a new page by clicking the (**+**) button to the right of the current page title.
   c) Once the new page has been created, modify its title to **Time Intelligence**.




   d) Now that you have created a new page, you can now add a few new visuals.
2. Create a new matrix visual to show sales revenue for specific time periods.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.

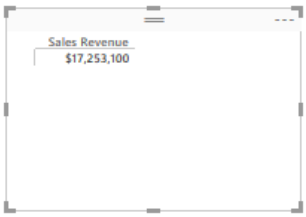



   b) Change the visual to a matrix by clicking the **Matrix** button in the **Visualizations** list.



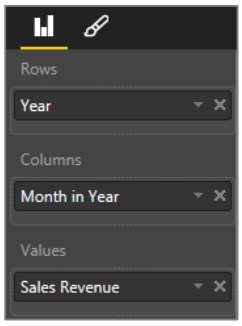

© Critical Path Training. 2015. All Rights Reserved

c) Select the checkbox next to the **Sales Revenue** measure in the **Fields** list. When you select the **Sales Revenue** measure, the report designer will add it to the **Values** well and the visual will show a single value for total sales revenue across the entire **Sales** table.



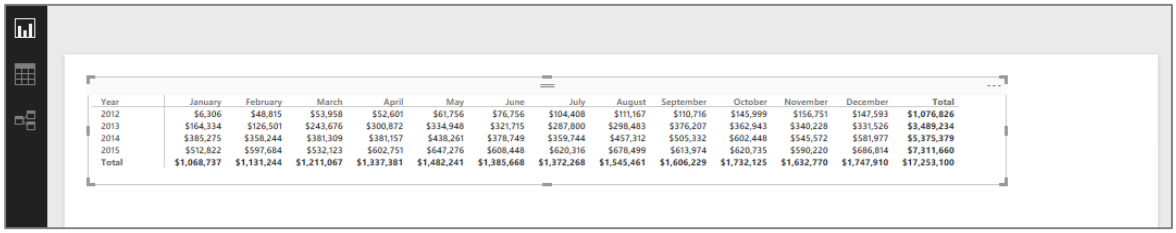

d) Now it's time to extend the matrix by adding row labels and column labels. First, drag and drop the **Year** column from the **Calendar** table in the **Fields** list into the **Rows** well in the **Visualizations** pane.

e) Now drag and drop the **Month in Year** column from the **Calendar** table in the **Fields** list into the **Rows** well in the **Visualizations** pane.



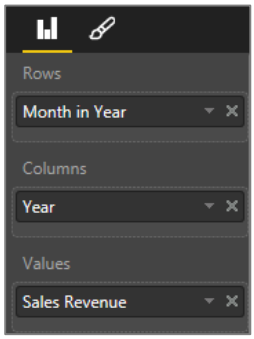

f) The matrix now has a column for each month and a row for each year.

g) Use your mouse to resize the matrix visual so you can see all the columns.



| Year | January | February | March | April | May | June | July | August | September | October | November | December | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2012 | $6,306 | $48,815 | $53,958 | $52,601 | $61,756 | $76,756 | $104,408 | $111,167 | $110,716 | $145,999 | $156,751 | $147,593 | $1,076,826 |
| 2013 | $164,334 | $126,501 | $243,676 | $300,872 | $334,948 | $321,715 | $287,800 | $298,483 | $376,207 | $362,943 | $340,228 | $331,526 | $3,489,234 |
| 2014 | $385,275 | $358,244 | $381,309 | $381,157 | $438,261 | $378,749 | $359,744 | $457,312 | $505,332 | $602,448 | $545,572 | $581,977 | $5,375,379 |
| 2015 | $512,822 | $597,684 | $532,123 | $602,751 | $647,276 | $608,448 | $620,316 | $678,499 | $613,974 | $620,735 | $590,220 | $686,814 | $7,311,660 |
| Total | $1,068,737 | $1,131,244 | $1,211,067 | $1,337,381 | $1,482,241 | $1,385,668 | $1,372,268 | $1,545,461 | $1,606,229 | $1,732,125 | $1,632,770 | $1,747,910 | $17,253,100 |

h) Now experiment by pivoting the matrix visual to display the exact same data using a different layout. Accomplish this by moving the **Month in Year** field into the **Rows** well and then moving the **Year** field into the **Columns** well. In effect, the **Month in Year** field and the **Year** field have just switched places.

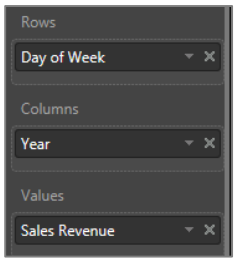



i) The matrix should now display a row for each month and a column for each year.

© Critical Path Training. 2015. All Rights Reserved

| Month in Year | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| January | $6,306 | $164,334 | $385,275 | $512,822 | $1,068,737 |
| February | $48,815 | $126,501 | $358,244 | $597,684 | $1,131,244 |
| March | $53,958 | $243,676 | $381,309 | $532,123 | $1,211,067 |
| April | $52,601 | $300,872 | $381,157 | $602,751 | $1,337,381 |
| May | $61,756 | $334,948 | $438,261 | $647,276 | $1,482,241 |
| June | $76,756 | $321,715 | $378,749 | $608,448 | $1,385,668 |
| July | $104,408 | $287,800 | $359,744 | $620,316 | $1,372,268 |
| August | $111,167 | $298,483 | $457,312 | $678,499 | $1,545,461 |
| September | $110,716 | $376,207 | $505,332 | $613,974 | $1,606,229 |
| October | $145,999 | $362,943 | $602,448 | $620,735 | $1,732,125 |
| November | $156,751 | $340,228 | $545,572 | $590,220 | $1,632,770 |
| December | $147,593 | $331,526 | $581,977 | $686,814 | $1,747,910 |
| Total | $1,076,826 | $3,489,234 | $5,375,379 | $7,311,660 | $17,253,100 |

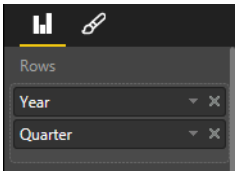
3. Create a new matrix visual to show sales revenue for specific time periods.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Make sure this new visual is positioned directly below the first visual that you created.
   c) Change the visual to a matrix by clicking the Matrix button in the **Visualizations** list.
   d) Drag and drop the **Day of Week** column from the **Calendar** table in the **Fields** list into the **Rows** well.
   e) Drag and drop the **Year** column from the **Calendar** table into the **Columns** well.
   f) Drag and drop the **Sales Revenue** measure from the **Calendar** table in the **Fields** list into the **Values** well.




   g) The matrix should now display a row for each day of the week and a column for each year.

| Day of Week | 2012 | 2013 | 2014 | 2015 | Total |
|---|---|---|---|---|---|
| Monday | $163,919 | $486,428 | $802,943 | $1,090,519 | $2,543,808 |
| Tuesday | $154,244 | $462,695 | $772,438 | $1,163,588 | $2,552,965 |
| Wednesday | $162,811 | $425,869 | $752,274 | $1,029,697 | $2,370,651 |
| Thursday | $169,687 | $513,948 | $837,905 | $1,066,526 | $2,588,067 |
| Friday | $146,575 | $582,823 | $736,297 | $1,059,080 | $2,524,774 |
| Saturday | $162,219 | $546,922 | $694,322 | $909,334 | $2,312,797 |
| Sunday | $117,372 | $470,550 | $779,200 | $992,915 | $2,360,038 |
| Total | $1,076,826 | $3,489,234 | $5,375,379 | $7,311,660 | $17,253,100 |

4. Add a third matrix visual to analyze sales data calculated at both the yearly level and the quarterly level.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a matrix by clicking the Matrix button in the Visualizations list.
   c) Drag and drop the **Year** column from the **Calendar** table into the **Rows** well.
   d) Drag and drop the **Quarter** column from the **Calendar** table into the **Rows** well.




© Critical Path Training. 2015. All Rights Reserved

e) At this point, the visual has row labels for the **Year** column and the **Quarter** column.

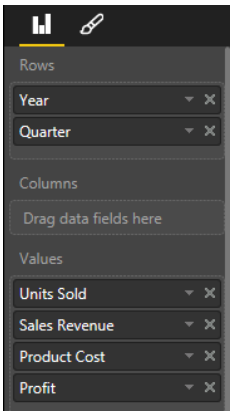
| Year | Quarter |
|---|---|
| 2012 | 2012-Q1 |
| | 2012-Q2 |
| | 2012-Q3 |
| | 2012-Q4 |
| 2013 | 2013-Q1 |
| | 2013-Q2 |
| | 2013-Q3 |
| | 2013-Q4 |
| 2014 | 2014-Q1 |
| | 2014-Q2 |
| | 2014-Q3 |
| | 2014-Q4 |
| 2015 | 2015-Q1 |
| | 2015-Q2 |
| | 2015-Q3 |
| | 2015-Q4 |

f) Drag and drop the **Units Sold** measure from the **Sales** table into the **Values** well.

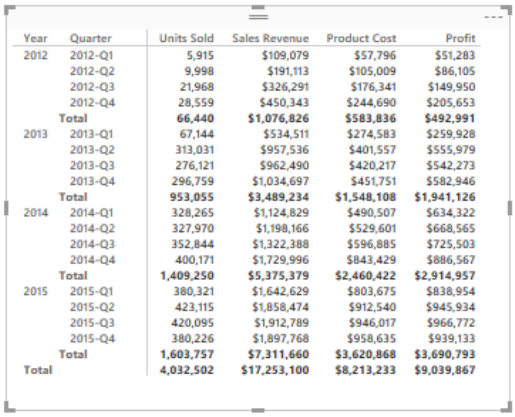
g) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Values** well.

h) Drag and drop the **Product Cost** measure from the **Sales** table into the **Values** well.

i) Drag and drop the **Profit** measure from the **Sales** table into the **Values** well.



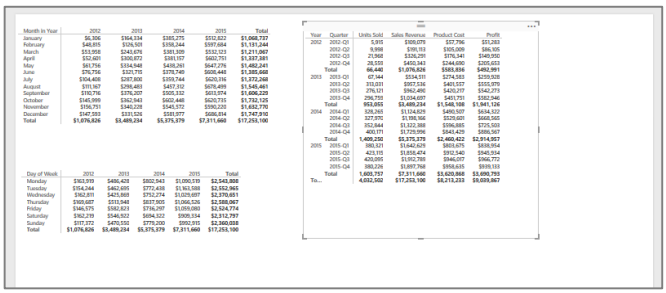

j) The matrix should now display values for each measure calculated at the quarterly level as well as at the yearly level.



| Year | Quarter | Units Sold | Sales Revenue | Product Cost | Profit |
|---|---|---|---|---|---|
| 2012 | 2012-Q1 | 5,915 | $109,079 | $57,796 | $51,283 |
| | 2012-Q2 | 9,998 | $191,113 | $105,009 | $86,105 |
| | 2012-Q3 | 21,968 | $326,291 | $176,341 | $149,950 |
| | 2012-Q4 | 28,559 | $450,343 | $244,690 | $205,653 |
| | **Total** | **66,440** | **$1,076,826** | **$583,836** | **$492,991** |
| 2013 | 2013-Q1 | 67,144 | $534,511 | $274,583 | $259,928 |
| | 2013-Q2 | 313,031 | $957,536 | $401,557 | $555,979 |
| | 2013-Q3 | 276,121 | $962,490 | $420,217 | $542,273 |
| | 2013-Q4 | 296,759 | $1,034,697 | $451,751 | $582,946 |
| | **Total** | **953,055** | **$3,489,234** | **$1,548,108** | **$1,941,126** |
| 2014 | 2014-Q1 | 328,265 | $1,124,829 | $490,507 | $634,322 |
| | 2014-Q2 | 327,970 | $1,198,166 | $529,601 | $668,565 |
| | 2014-Q3 | 352,844 | $1,322,388 | $596,885 | $725,503 |
| | 2014-Q4 | 400,171 | $1,729,996 | $843,429 | $886,567 |
| | **Total** | **1,409,250** | **$5,375,379** | **$2,460,422** | **$2,914,957** |
| 2015 | 2015-Q1 | 380,321 | $1,642,629 | $803,675 | $838,954 |
| | 2015-Q2 | 423,115 | $1,858,474 | $912,540 | $945,934 |
| | 2015-Q3 | 420,095 | $1,912,789 | $946,017 | $966,772 |
| | 2015-Q4 | 380,226 | $1,897,768 | $958,635 | $939,133 |
| | **Total** | **1,603,757** | **$7,311,660** | **$3,620,868** | **$3,690,793** |
| **Total** | | **4,032,502** | **$17,253,100** | **$8,213,233** | **$9,039,867** |

5. Using your mouse, arrange the new matrix visual on the right side of the page at the top to match the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

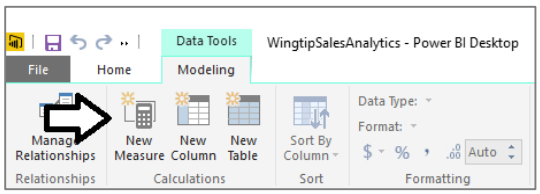


6. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

## Exercise 3: Creating Measures using DAX Time Intelligence Functions

In this exercise, you will leverage various the Time Intelligence functions in DAX to analyze sales revenue using quarter to date (QTD) totals and year to date (YTD) totals. You will also use DAX to write an expression which calculate a running total of sales revenue through the entire 4 years of sales activity.

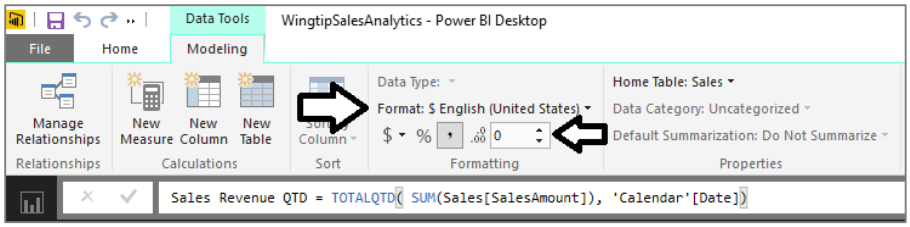
1. Create a measure named **Sales Revenue QTD** that calculates a quarter-to-date aggregate sum on the **SalesAmount** column of the **Sales** table.
   a) Navigate to data view.
   b) Select the **Sales** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.




   d) Enter to following DAX expression into the formula bar to create the new measure named **Sales Revenue QTD**.

```
Sales Revenue QTD = TOTALQTD( SUM(Sales[SalesAmount]), 'Calendar'[Date])
```

   e) Press the **ENTER** key to add the measure to data model.
   f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.



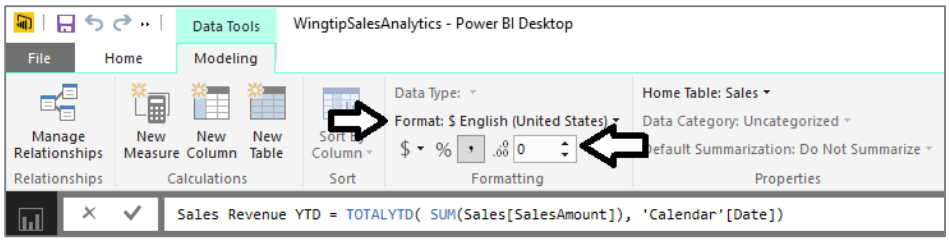

2. Create a measure named **Sales Revenue YTD** that calculates a year-to-date aggregate sum on the **SalesAmount** column of the **Sales** table.
   a) Navigate to data view.
   b) Select the **Sales** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the new measure named **Sales Revenue YTD**.

```
Sales Revenue QTD = TOTALYTD( SUM(Sales[SalesAmount]), 'Calendar'[Date])
```

© Critical Path Training. 2015. All Rights Reserved

e) Press the **ENTER** key to add the measure to data model.

f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.




3. Create a measure named **Sales Revenue RT** that calculates a running total aggregate sum on the **SalesAmount** column of the **Sales** table.

a) Navigate to data view.

b) Select the **Sales** table from the **Fields** list.

c) Create a new measure by clicking the **New Measure** button in the ribbon.

d) Enter to following DAX expression into the formula bar to create the new measure named **Sales Revenue RT**.

```
Sales Revenue RT =
CALCULATE(
 SUM(Sales[SalesAmount]),
 FILTER(
    ALL('Calendar'),
    'Calendar'[Date] <= MAX('Calendar'[Date])
 )
)
```

e) Press the **ENTER** key to add the measure to data model.

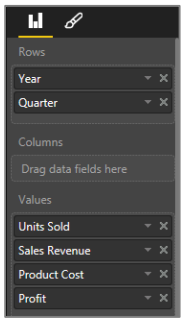
f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Currency > English (United States)**. Also use the spinner control below the format menu to set the number of decimal places shown to zero.

4. Use the three new measures in a matrix visual.

a) Navigate to report mode.

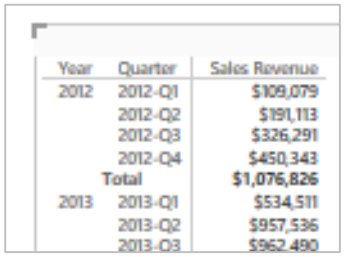
b) Select the matrix visual you created last that is currently positioned on the right-hand side of the page.

c) Examine the bottom of the **Visualizations** pane. Currently, the **Rows** well contains the **Year** column and the **Quarter** column. There are also four other measures in the **Values** well.



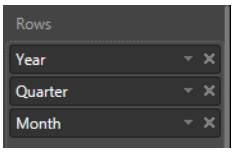

d) Remove all the measures from the **Values** well except for the **Sales Revenue** measure. At this point, your visual should match the one shown in the following screenshot where sales revenue totals as shown at the quarterly level and at the yearly level.

© Critical Path Training. 2015. All Rights Reserved

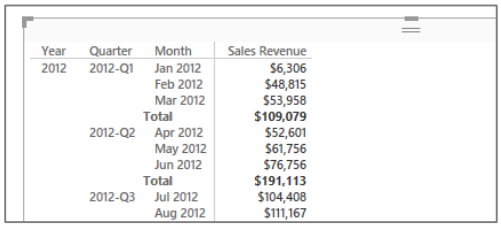


| Year | Quarter | Sales Revenue |
|---|---|---|
| 2012 | 2012-Q1 | $109,079 |
| | 2012-Q2 | $191,113 |
| | 2012-Q3 | $326,291 |
| | 2012-Q4 | $450,343 |
| | Total | $1,076,826 |
| 2013 | 2013-Q1 | $534,511 |
| | 2013-Q2 | $957,536 |
| | 2013-Q3 | $962,490 |

e) Drag and drop the **Month** column from the **Calendar** table into the **Rows** well below the two other columns.

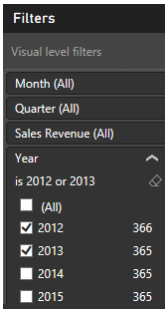



f) You should be able to see that now the visual now has a deeper level of granularity because it is show sales revenue broken out into a separate aggraegate value for each month.

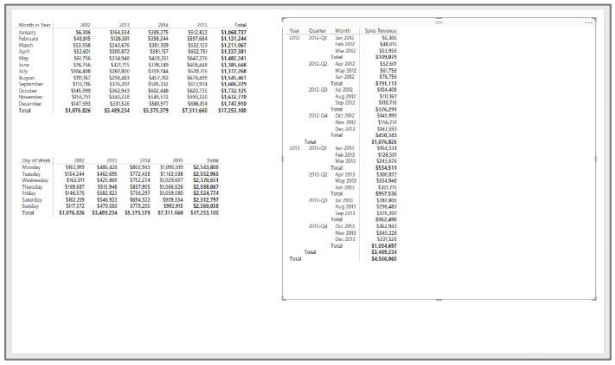


| Year | Quarter | Month | Sales Revenue |
|---|---|---|---|
| 2012 | 2012-Q1 | Jan 2012 | $6,306 |
| | | Feb 2012 | $48,815 |
| | | Mar 2012 | $53,958 |
| | | Total | $109,079 |
| | 2012-Q2 | Apr 2012 | $52,601 |
| | | May 2012 | $61,756 |
| | | Jun 2012 | $76,756 |
| | | Total | $191,113 |
| | 2012-Q3 | Jul 2012 | $104,408 |
| | | Aug 2012 | $111,167 |

g) Set a filter on the matrix visual so that is only displays sales revenue for the calendar years of 2012 and 2013. Accomplish this by selecting the checkboxes for **2012** and **2013** in the **Year** filter located inside the **Filters** section in the **Visualizations** pane.



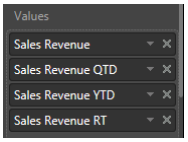

h) Resize the matrix visual to take up the entire right-hand side of the page.



i) Drag and drop the **Sales Revenue QTD** measure from the **Sales** table into the **Values** well.

j) Drag and drop the **Sales Revenue YTD** measure from the **Sales** table into the **Values** well.

k) Drag and drop the **Sales Revenue RT** measure from the **Sales** table into the **Values** well.



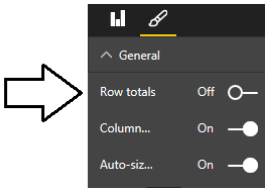

l) The matrix visual should now display three new columns for the three measure you added to the **Values** well.

| Year | Quarter | Month | Sales Revenue | Sales Revenue QTD | Sales Revenue YTD | Sales Revenue RT |
|---|---|---|---|---|---|---|
| 2012 | 2012-Q1 | Jan 2012 | $6,306 | $6,306 | $6,306 | $6,306 |
| | | Feb 2012 | $48,815 | $55,121 | $55,121 | $55,121 |
| | | Mar 2012 | $53,958 | $109,079 | $109,079 | $109,079 |
| | | **Total** | **$109,079** | **$109,079** | **$109,079** | **$109,079** |
| | 2012-Q2 | Apr 2012 | $52,601 | $52,601 | $161,680 | $161,680 |
| | | May 2012 | $61,756 | $114,357 | $223,436 | $223,436 |
| | | Jun 2012 | $76,756 | $191,113 | $300,192 | $300,192 |
| | | **Total** | **$191,113** | **$191,113** | **$300,192** | **$300,192** |
| | 2012-Q3 | Jul 2012 | $104,408 | $104,408 | $404,600 | $404,600 |
| | | Aug 2012 | $111,167 | $215,575 | $515,767 | $515,767 |

m) Currently the matrix is showing a **Total** row with a quarterly rollup. You should observe that the **Total** row can be confusing with measures like these because the it displays the exact same aggregate value as the row above it.

| Year | Quarter | Month | Sales Revenue | Sales Revenue QTD | Sales Revenue YTD | Sales Revenue RT |
|---|---|---|---|---|---|---|
| 2012 | 2012-Q1 | Jan 2012 | $6,306 | $6,306 | $6,306 | $6,306 |
| | | Feb 2012 | $48,815 | $55,121 | $55,121 | $55,121 |
| | | Mar 2012 | $53,958 | $109,079 | $109,079 | $109,079 |
| | | **Total** | **$109,079** | **$109,079** | **$109,079** | **$109,079** |
| | 2012-Q2 | Apr 2012 | $52,601 | $52,601 | $161,680 | $161,680 |
| | | May 2012 | $61,756 | $114,357 | $223,436 | $223,436 |

n) Look inside the **General** section for the visual in the **Visualizations** pane and set modify the **Row totals** property to **Off**.




o) The matrix visual is now better suited to display the three new measures you created in this exercise.

| Year | Quarter | Month | Sales Revenue | Sales Revenue QTD | Sales Revenue YTD | Sales Revenue RT |
|---|---|---|---|---|---|---|
| 2012 | 2012-Q1 | Jan 2012 | $6,306 | $6,306 | $6,306 | $6,306 |
| | | Feb 2012 | $48,815 | $55,121 | $55,121 | $55,121 |
| | | Mar 2012 | $53,958 | $109,079 | $109,079 | $109,079 |
| | 2012-Q2 | Apr 2012 | $52,601 | $52,601 | $161,680 | $161,680 |
| | | May 2012 | $61,756 | $114,357 | $223,436 | $223,436 |
| | | Jun 2012 | $76,756 | $191,113 | $300,192 | $300,192 |
| | 2012-Q3 | Jul 2012 | $104,408 | $104,408 | $404,600 | $404,600 |
| | | Aug 2012 | $111,167 | $215,575 | $515,767 | $515,767 |
| | | Sep 2012 | $110,716 | $326,291 | $626,483 | $626,483 |
| | 2012-Q4 | Oct 2012 | $145,999 | $145,999 | $772,483 | $772,483 |
| | | Nov 2012 | $156,751 | $302,750 | $929,234 | $929,234 |
| | | Dec 2012 | $147,593 | $450,343 | $1,076,826 | $1,076,826 |
| 2013 | 2013-Q1 | Jan 2013 | $164,334 | $164,334 | $164,334 | $1,241,161 |
| | | Feb 2013 | $126,501 | $290,835 | $290,835 | $1,367,661 |

© Critical Path Training. 2015. All Rights Reserved

p) Now imagine your boss asks you to determine in what month the company reached 1 million dollars in total sales revenue. By looking at down the list of values for the **Sales Revenue RT** measure, you can see that the company finally hit $1,000,000 in sales revenue in December of 2012.

| | | | | |
|---|---|---|---|---|
| Nov 2012 | $156,751 | $302,750 | $929,234 | $929,234 |
| Dec 2012 | $147,593 | $450,343 | $1,076,826 | $1,076,826 |
| Jan 2013 | $164,334 | $164,334 | $164,334 | $1,241,161 |
| Feb 2013 | $126,501 | $290,835 | $290,835 | $1,367,661 |

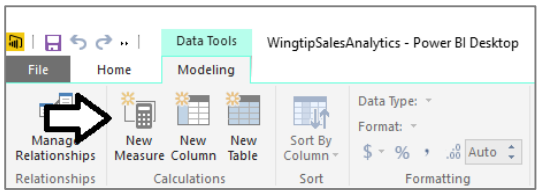
5. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

You have now learned how to use Time Intelligence functions in DAX together with a calendar table. Now you will move on to the final exercise where you will create additional measures to monitor sales growth.

## Exercise 4: Creating Measures to Monitor Growth in Sales Revenue

In this exercise you will create new measures to calculate the growth of sales revenue on a quarter-by-quarter basis as well as on a month-by-month basis. After that you will create additional measures that will act as KPIs to monitor the health of sales growth and provide visual indications as to how each month has done when compared to the previous month.

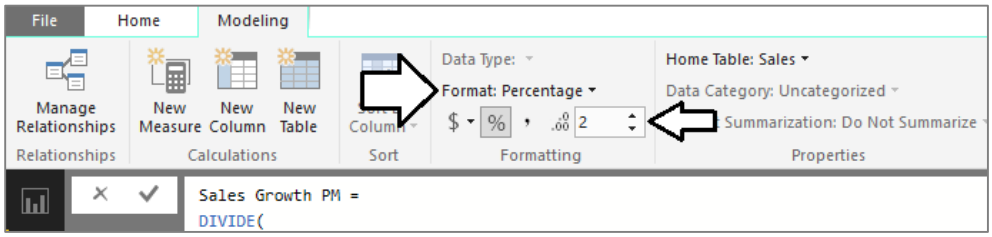
1. Create a measure named **Sales Growth PM** that calculates the percentage increase between sales revenue for the current month and sales revenue for the previous month.
   a) Navigate to data view.
   b) Select the **Sales** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.




   d) Enter to following DAX expression into the formula bar to create the measure named **Sales Growth PM**.

```
Sales Growth PM =
DIVIDE(
  SUM(Sales[SalesAmount]) -
  CALCULATE(
    SUM(Sales[SalesAmount]),
    PREVIOUSMONTH(Calendar[Date])
  ),
  CALCULATE(
    SUM(Sales[SalesAmount]),
    PREVIOUSMONTH(Calendar[Date])
  )
)
```

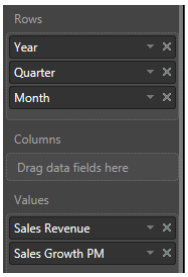
   e) Press the **ENTER** key to add the calculated column to data model.
   f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.



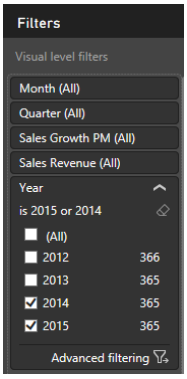

© Critical Path Training. 2015. All Rights Reserved

2. Create a new page in the project's report.
   a) Navigate to report view.
   b) Add a new page by clicking the (**+**) button to the right of the current page title.
   c) Once the new page has been created, modify its title to **Sales Revenue Growth**.

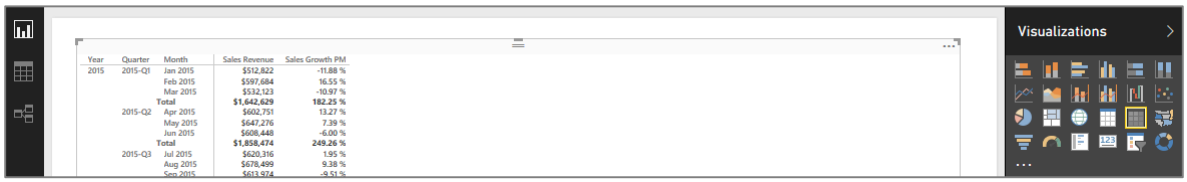
3. Create a new matrix visual to show month-to-month sales revenue growth in 2014 and 2015.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a matrix by clicking the Matrix button in the **Visualizations** list.
   c) Drag and drop the **Year** column from the **Calendar** table into the **Rows** well.
   d) Drag and drop the **Quarter** column from the **Calendar** table into the **Rows** well.
   e) Drag and drop the **Month** column from the **Calendar** table into the **Rows** well.
   f) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Value** well.
   g) Drag and drop the **Sales Revenue PM** measure from the **Sales** table into the **Value** well.
   h) The **Rows** well and the **Values** well for the matrix visual should match the following screenshot.




   i) Move down in the **Visualizations** page to the **Filters** section. Drop down the filter menu for **Year** and select the years of 2014 and 2015.




   j) Use your mouse the resize the matrix visual so you can see all the rows and columns. Give the matrix visual a width that is almost the same size of the page so you can add additional columns over the next few steps without having to resize the visual again.




© Critical Path Training. 2015. All Rights Reserved

k) Inspect the values produced by the **Sales Growth PM** measure. You can see that a value has been calculated for each month. You should also notice that the matrix currently displays values for the **Sales Growth PM** measure in the **Total** row. The values in the **Total** row are calculated at the quarterly level and not at the month level.

| Year | Quarter | Month | Sales Revenue | Sales Growth PM |
|---|---|---|---|---|
| 2015 | 2015-Q1 | Jan 2015 | $512,822 | -11.88 % |
| | | Feb 2015 | $597,684 | 16.55 % |
| | | Mar 2015 | $532,123 | -10.97 % |
| | | **Total** | **$1,642,629** | **182.25 %** |
| | 2015-Q2 | Apr 2015 | $602,751 | 13.27 % |
| | | May 2015 | $647,276 | 7.39 % |
| | | Jun 2015 | $608,448 | -6.00 % |
| | | **Total** | **$1,858,474** | **249.26 %** |
| | 2015-Q3 | Jul 2015 | $620,316 | 1.95 % |
| | | Aug 2015 | $678,499 | 9.38 % |

The **Sales Growth PM** measure was written to perform calculations on a month-to-month basis. However, there is currently a problem whenever this measure is evaluated in the context based on a larger time interval such as a quarter or a year. More specifically, the **Sales Growth PM** measure is currently producing a large and erroneous value when it is evaluated in the context of a quarter. Now that you have seen the problem, it's time to modify the DAX expression for the **Sales Growth PM** measure to return a blank value whenever the measure is evaluated in a context where the time interval is at a granularity other than at the month level.

4. Modify the DAX expression for the **Sales Revenue PM** measure.

a) Navigate to data view.

b) Select the **Sales Revenue PM** measure of the **Sales** table from the **Fields** list. When you select the **Sales Revenue PM** measure in the **Fields** list, you should then be able to see and modify its DAX expression in the formula bar.

c) Before you can modify the DAX expression for the **Sales Revenue PM** measure, you must be able to use the ISFILTERED function provided by DAX. You can write the following DAX expression to determine whether the current evaluation context is filtering at the month level.

```
ISFILTERED(Calendar[Month])
```

d) You can also write the following DAX expression to make sure that the current evaluation context is not filtering at a more granular level such as at the **Date** level.

```
ISFILTERED(Calendar[Date]) = FALSE()
```

e) You will need to ensure that both these expressions are true before the **Sales Revenue PM** measure evaluates to a value other than a blank value. You can write the following DAX expression using the DAX **&&** operator to return true when both inner conditions are true

```
( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() )
```

f) Update the DAX expression for the **Sales Revenue PM** measure to match the following code listing.

```
Sales Growth PM =
IF(
  ( ISFILTERED(Calendar[Month]) && ISFILTERED(Calendar[Date]) = FALSE() ),
  DIVIDE(
    SUM(Sales[SalesAmount]) -
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    ),
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSMONTH(Calendar[Date])
    )
  ),
  BLANK()
)
```

g) Navigate back to report view and inspect the effects of your changes to the visual on the **Sales Revenue Growth** page.

h) You should see that the **Sales Growth PM** measure is now returning blank values in the **Total** row for the quarterly evaluation.

© Critical Path Training. 2015. All Rights Reserved

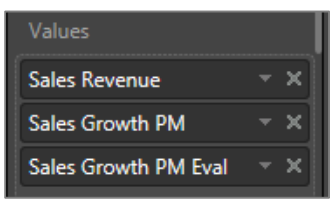
| Year | Quarter | Month | Sales Revenue | Sales Growth PM |
|---|---|---|---|---|
| 2015 | 2015-Q1 | Jan 2015 | $512,822 | -11.88 % |
| | | Feb 2015 | $597,684 | 16.55 % |
| | | Mar 2015 | $532,123 | -10.97 % |
| | | **Total** | **$1,642,629** | |
| | 2015-Q2 | Apr 2015 | $602,751 | 13.27 % |
| | | May 2015 | $647,276 | 7.39 % |
| | | Jun 2015 | $608,448 | -6.00 % |
| | | **Total** | **$1,858,474** | |
| | 2015-Q3 | Jul 2015 | $620,316 | 1.95 % |
| | | Aug 2015 | $678,499 | 9.38 % |

It is widely-accepted among BI experts and BI novices alike that a blank value is always better than large, erroneous value.

5. Create a measure named **Sales Growth PM Eval** that inspects the value of the **Sales Growth PM** measure and evaluates to a short string value to indicate the health of the sales growth value.
   a) Navigate to data view.
   b) Select the **Sales** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the measure named **Sales Growth PM Eval**.

```
Sales Growth PM Eval =
IF(
  ISNUMBER([Sales Growth PM]),
  SWITCH(TRUE(),
    ([Sales Growth PM] >= 0.2), "EXCELLENT",
    ([Sales Growth PM] >= 0.1), "GOOD",
    ([Sales Growth PM] >= 0), "OK",
    ([Sales Growth PM] >= -0.1), "BAD",
    ([Sales Growth PM] <  -0.1), "AWFUL"
  )
)
```

   e) Navigate to report view and select the matrix visual on the **Sales Revenue Growth** page.
   f) Drag and drop the **Sales Growth PM Eval** measure from the **Sales** table into the **Values** well in the **Visualizations** pane.




   g) You should now see values for the **Sales Growth PM Eval** measure which indicate the health of sales revenue growth.

| Year | Quarter | Month | Sales Revenue | Sales Growth PM | Sales Growth PM Eval |
|---|---|---|---|---|---|
| 2014 | 2014-Q1 | Jan 2014 | $385,275 | 16.21 % | GOOD |
| | | Feb 2014 | $358,244 | -7.02 % | BAD |
| | | Mar 2014 | $381,309 | 6.44 % | OK |
| | | **Total** | **$1,124,829** | | |
| | 2014-Q2 | Apr 2014 | $381,157 | -0.04 % | BAD |
| | | May 2014 | $438,261 | 14.98 % | GOOD |
| | | Jun 2014 | $378,749 | -13.58 % | AWFUL |
| | | **Total** | **$1,198,166** | | |
| | 2014-Q3 | Jul 2014 | $359,744 | -5.02 % | BAD |
| | | Aug 2014 | $457,312 | 27.12 % | EXCELLENT |
| | | Sep 2014 | $505,332 | 10.50 % | GOOD |

6. Create a measure named **Sales Growth PQ** that calculates the percentage increase between sales revenue for the current quarter and sales revenue for the previous quarter.
   a) Navigate to data view.
   b) Select the **Sales** table from the **Fields** list.

© Critical Path Training. 2015. All Rights Reserved

c) Create a new measure by clicking the **New Measure** button in the ribbon.

d) Enter to following DAX expression into the formula bar to create the measure named **Sales Growth PQ**.

```
Sales Growth PQ =
IF(
  ( ISFILTERED('Calendar'[Quarter]) &&
    (ISFILTERED('Calendar'[Month]) = FALSE() && ISFILTERED(Calendar[Date]) = FALSE())
  ),
  DIVIDE(
    SUM(Sales[SalesAmount]) -
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSQUARTER(Calendar[Date])
    ),
    CALCULATE(
      SUM(Sales[SalesAmount]),
      PREVIOUSQUARTER(Calendar[Date])
    )
  ),
  BLANK()
)
```

e) Press the **ENTER** key to add the calculated column to data model.

f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.

7. Create a measure named **Sales Growth PQ Eval** that inspects the value of the **Sales Growth PQ** measure and evaluates to a short string value to indicate the health of the sales growth value.

a) Navigate to data view.

b) Select the **Sales** table from the **Fields** list.

c) Create a new measure by clicking the **New Measure** button in the ribbon.

d) Enter to following DAX expression into the formula bar to create the measure named **Sales Growth PQ Eval**.

```
Sales Growth PQ Eval =
IF(
  ISNUMBER([Sales Growth PQ]),
  SWITCH(TRUE(),
    ([Sales Growth PQ] >= 0.2), "EXCELLENT",
    ([Sales Growth PQ] >= 0.1), "GOOD",
    ([Sales Growth PQ] >= 0), "OK",
    ([Sales Growth PQ] >= -0.1), "BAD",
    ([Sales Growth PQ] <  -0.1), "AWFUL"
  )
)
```

8. Add the **Sales Growth PQ** measure and the **Sales Growth PQ Eval** measure to the matrix visual.

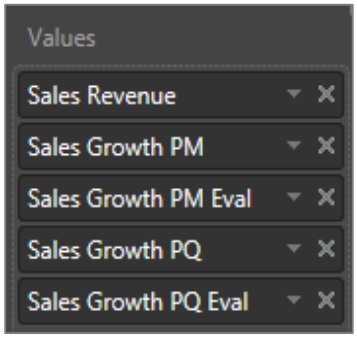
a) Navigate to report view and select the matrix visual on the **Sales Revenue Growth** page.

b) There should already be three fields in the **Values** well for this visual.

c) Drag and drop the **Sales Growth PQ** measure from the **Sales** table into the **Values** well in the **Visualizations** pane.

d) Drag and drop the **Sales Growth PQ Eval** measure from the **Sales** table into the **Values** well in the **Visualizations** pane.

© Critical Path Training. 2015. All Rights Reserved




e) You should now see values for the **Sales Growth PQ** measure and the **Sales Growth PQ Eval** measure in the **Total** rows which are evaluated at the quarterly level.

| Year | Quarter | Month | Sales Revenue | Sales Growth PM | Sales Growth PM Eval | Sales Growth PQ | Sales Growth PQ Eval |
|---|---|---|---|---|---|---|---|
| 2014 | 2014-Q1 | Jan 2014 | $385,275 | 16.21 % | GOOD | | |
| | | Feb 2014 | $358,244 | -7.02 % | BAD | | |
| | | Mar 2014 | $381,309 | 6.44 % | OK | | |
| | | **Total** | **$1,124,829** | | | **8.71 %** | **OK** |
| | 2014-Q2 | Apr 2014 | $381,157 | -0.04 % | BAD | | |
| | | May 2014 | $438,261 | 14.98 % | GOOD | | |
| | | Jun 2014 | $378,749 | -13.58 % | AWFUL | | |
| | | **Total** | **$1,198,166** | | | **6.52 %** | **OK** |
| | 2014-Q3 | Jul 2014 | $359,744 | -5.02 % | BAD | | |
| | | Aug 2014 | $457,312 | 27.12 % | EXCELLENT | | |
| | | Sep 2014 | $505,332 | 10.50 % | GOOD | | |
| | | **Total** | **$1,322,388** | | | **10.37 %** | **GOOD** |
| | 2014-Q4 | Oct 2014 | $602,448 | 19.22 % | GOOD | | |
| | | Nov 2014 | $545,572 | -9.44 % | BAD | | |
| | | Dec 2014 | $581,977 | 6.67 % | OK | | |
| | | **Total** | **$1,729,996** | | | **30.82 %** | **EXCELLENT** |

9. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

© Critical Path Training. 2015. All Rights Reserved

# Module 06: Designing Reports in Power BI Desktop

## Module Description

This module begins by explaining the fundamental concepts and the relationship between reports, visualizations, dashboard and tiles. Students will learn how to design reports using Power BI Desktop and the standard set of Power BI visualizations as well as how to publish reports to the Power BI service. The module also discusses how to make reports more interactive using filters and slicers. Students will also learn how to create dashboards in Power BI which display summary information as well as how to use dashboard tiles to link a dashboard to reports and underlying datasets which allow users to drill down in greater levels of detail and granularity.

## Module Agenda

- Understanding Visualization Types
- Working with Reports and Visuals
- Designing Interactive Reports
- Publishing Reports to the Power BI Service

## Topics Covered

Built-in Visualization Types
Creating Reports
Creating Visuals
Modifying Visual Properties
Reusing Visuals
Calculating Percentage of Product sales
The Product Revenue Breakdown Report
Creating the Top 5 Products Report
Next Steps
There a Problem with Measure Filters
Fixing the Problem
Display Top 5 Products with Two Visuals
Creating the Top 10 Customers Report
The Finished Report
Publishing a Power BI Desktop Project
Examining What's Been Published
Examining Dataset Properties

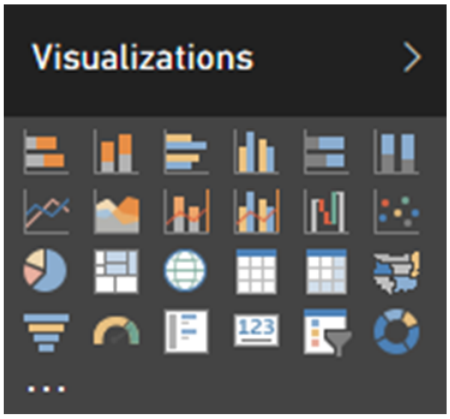
## Instructor Demos

- Creating Visuals with Power BI's Built-in Visualization Types

© Critical Path Training. 2015. All Rights Reserved

# Built-in Visualization Types

- Table and Matrix visuals
- Slicers
- Single number card tiles
- Bar charts and Column charts
- Line charts and Pie charts
- Basic (Layered) Area chart
- Combo charts
- Tree Maps
- Waterfall charts
- Doughnut charts
- Funnel charts
- Gauge charts
- Maps (basic)
- Scatter and bubble charts




© Critical Path Training. 2015. All Rights Reserved

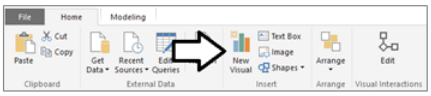
## Creating Reports

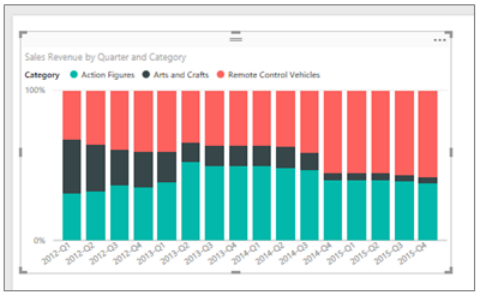
- Power BI Desktop project has one report
  - The report can contain many report pages
  - Report page contains visuals
- Reports can have filters
  - Visual-level filters
  - Page-level filters
  - Report-level filters

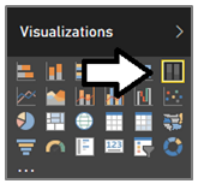
© Critical Path Training. 2015. All Rights Reserved

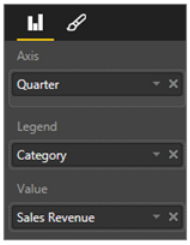
# Creating Visuals

- Working with visuals
  - Add visual to a page
  - Change its visualization type
  - Add fields to visual wells



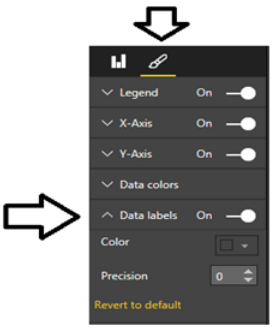



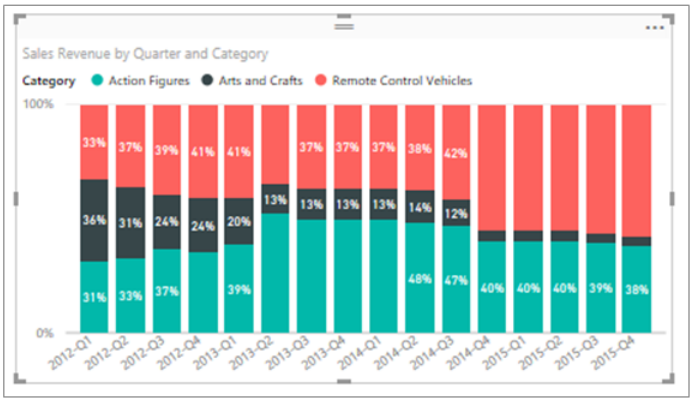




© Critical Path Training. 2015. All Rights Reserved

# Modifying Visual Properties

- Each visual has editable properties
  - Data properties
  - Appearance properties

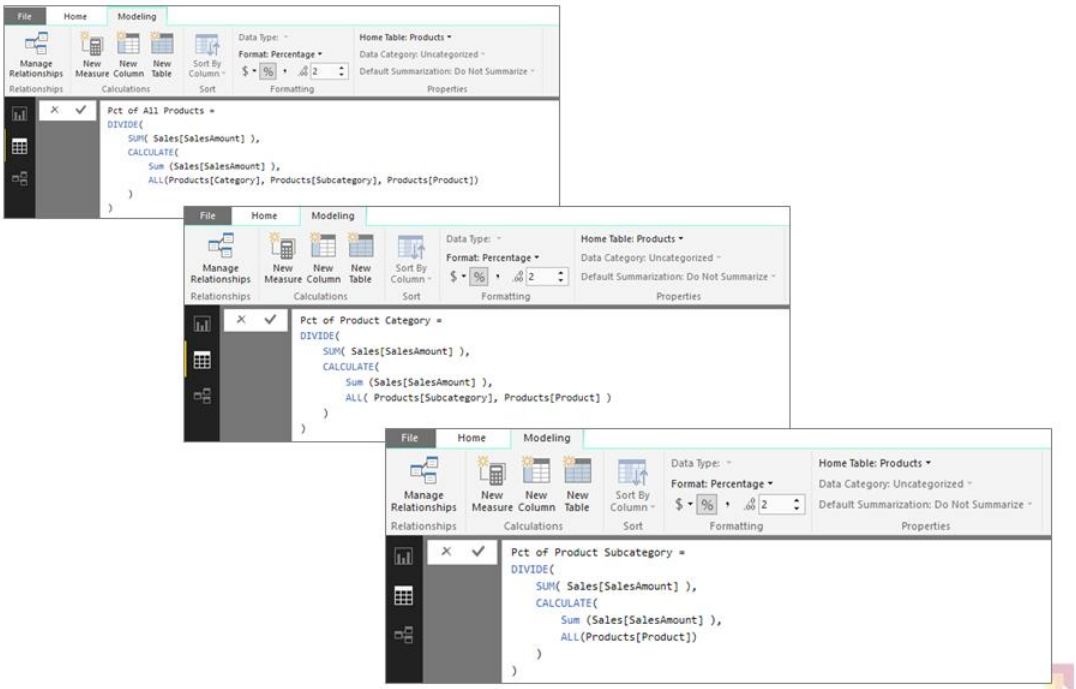


# Reusing Visuals

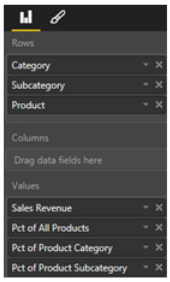
- Visuals support copy-and-paste
  - Quick way to clone a visual for reuse

# Calculating Percentage of Product sales




© Critical Path Training. 2015. All Rights Reserved

# The Product Revenue Breakdown Report




| Category | Subcategory | Product | Sales Revenue | Pct of All Products | Pct of Product Category | Pct of Product Subcategory |
|---|---|---|---|---|---|---|
| Action Figures | Cute and Huggable | Black Power Ranger Action Figure | $13,013 | 0.08 % | 0.18 % | 0.30 % |
| | | Green Angry Bird Action Figure | $23,295 | 0.14 % | 0.31 % | 0.53 % |
| | | Perry the Platypus Action Figure | $401,992 | 2.33 % | 5.44 % | 9.23 % |
| | | Phineas and Ferb Action Figure Set | $292,706 | 1.70 % | 3.96 % | 6.72 % |
| | | Red Angry Bird Action Figure | $61,654 | 0.36 % | 0.83 % | 1.42 % |
| | | Twitter Follower Action Figure | $3,493,118 | 20.25 % | 47.23 % | 80.19 % |
| | | Woody Action Figure | $70,287 | 0.41 % | 0.95 % | 1.61 % |
| | | **Total** | **$4,356,065** | **25.25 %** | **58.90 %** | **100.00 %** |
| | Tough Guys | Batman Action Figure | $130,768 | 0.76 % | 1.77 % | 4.30 % |
| | | Captain America Action Figure | $499,715 | 2.90 % | 6.76 % | 16.44 % |
| | | GI Joe Action Figure | $168,905 | 0.98 % | 2.28 % | 5.56 % |
| | | Godzilla Action Figure | $1,744,707 | 10.11 % | 23.59 % | 57.40 % |
| | | Green Hulk Action Figure | $82,764 | 0.48 % | 1.12 % | 2.72 % |
| | | Red Hulk Alter Ego Action Figure | $13,691 | 0.08 % | 0.19 % | 0.45 % |
| | | Spiderman Action Figure | $398,834 | 2.31 % | 5.39 % | 13.12 % |
| | | **Total** | **$3,039,384** | **17.62 %** | **41.10 %** | **100.00 %** |
| | **Total** | | **$7,395,449** | **42.86 %** | **100.00 %** | **100.00 %** |

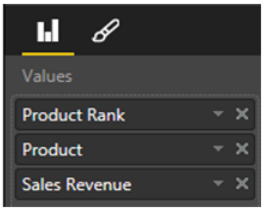
© Critical Path Training. 2015. All Rights Reserved

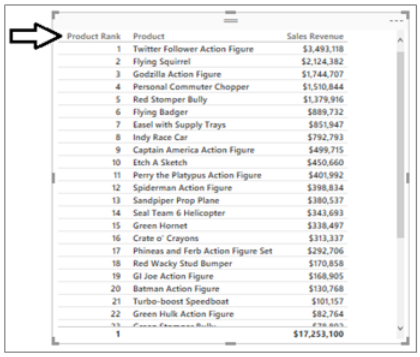
# Creating the Top 5 Products Report

- It starts with creating measure to rank products

```
Product Rank =
RANKX(
    ALL(Products),
    CALCULATE( SUM(Sales[SalesAmount]) )
)
```

- Next, add **Product Rank** measure to table visual



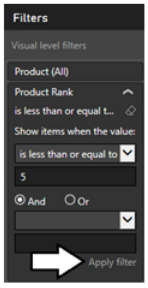



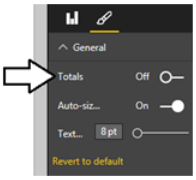
| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Twitter Follower Action Figure | $3,493,118 |
| 2 | Flying Squirrel | $2,124,382 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 5 | Red Stomper Bully | $1,379,916 |
| 6 | Flying Badger | $889,732 |
| 7 | Easel with Supply Trays | $851,947 |
| 8 | Indy Race Car | $792,793 |
| 9 | Captain America Action Figure | $499,715 |
| 10 | Etch A Sketch | $450,660 |
| 11 | Perry the Platypus Action Figure | $401,992 |
| 12 | Spiderman Action Figure | $398,834 |
| 13 | Sandpiper Prop Plane | $380,537 |
| 14 | Seal Team 6 Helicopter | $343,693 |
| 15 | Green Hornet | $338,497 |
| 16 | Crate o' Crayons | $313,337 |
| 17 | Phineas and Ferb Action Figure Set | $292,706 |
| 18 | Red Wacky Stud Bumper | $170,858 |
| 19 | GI Joe Action Figure | $168,905 |
| 20 | Batman Action Figure | $130,768 |
| 21 | Turbo-boost Speedboat | $101,157 |
| 22 | Green Hulk Action Figure | $82,764 |
| 1 | | $17,253,100 |

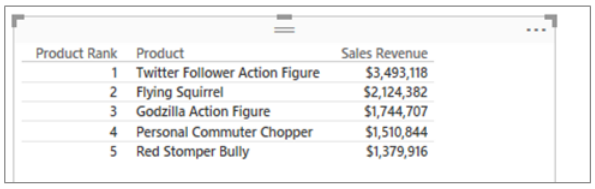
© Critical Path Training. 2015. All Rights Reserved

# Next Steps

- Apply filter and remove Totals row



| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Twitter Follower Action Figure | $3,493,118 |
| 2 | Flying Squirrel | $2,124,382 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 5 | Red Stomper Bully | $1,379,916 |

- Add slicer to make report interactive

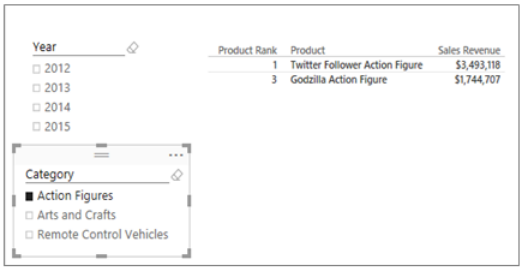
| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Easel with Supply Trays | $198,851 |
| 2 | Godzilla Action Figure | $156,268 |
| 3 | Red Stomper Bully | $125,251 |
| 4 | Flying Badger | $69,931 |
| 5 | Flying Squirrel | $64,564 |

© Critical Path Training. 2015. All Rights Reserved

# There a Problem with Measure Filters

- Add a Slicer to filter on product category

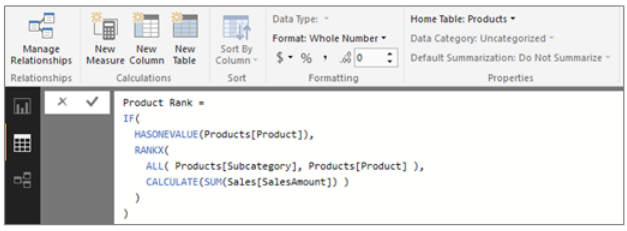
- Top 5 products ranking measure is now broken



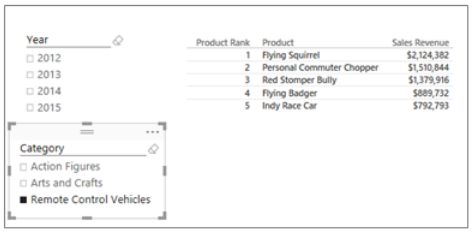

© Critical Path Training. 2015. All Rights Reserved

# Fixing the Problem

- Rewrite DAX using ALL function
  - Include columns which should not be filtered



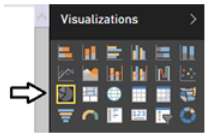

- Top 5 products ranking now works as expected

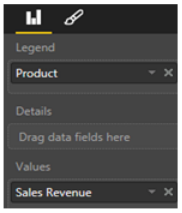


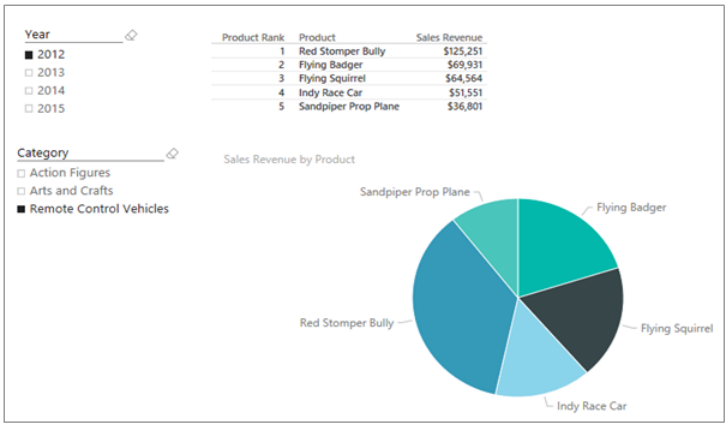

© Critical Path Training. 2015. All Rights Reserved

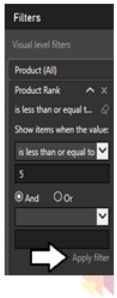
# Display Top 5 Products with Two Visuals

- Add pie chart
  - Filter by product rank



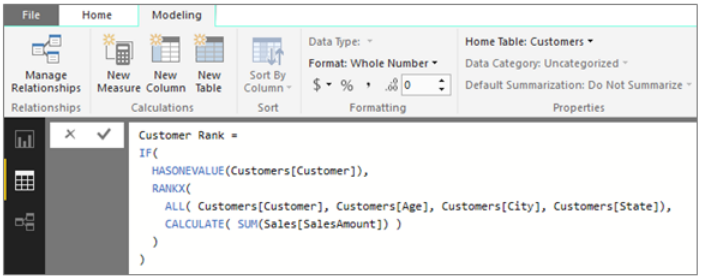



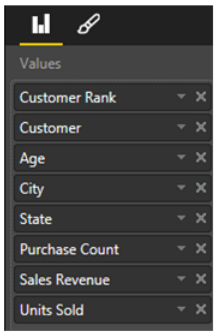



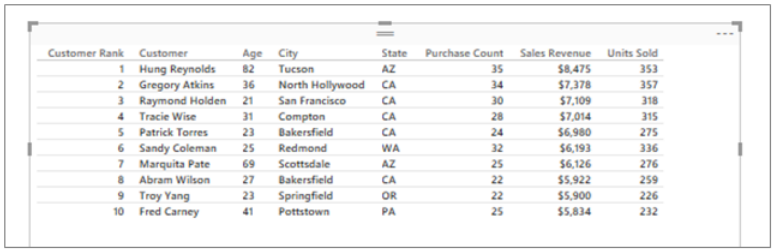




© Critical Path Training. 2015. All Rights Reserved

# Creating the Top 10 Customers Report



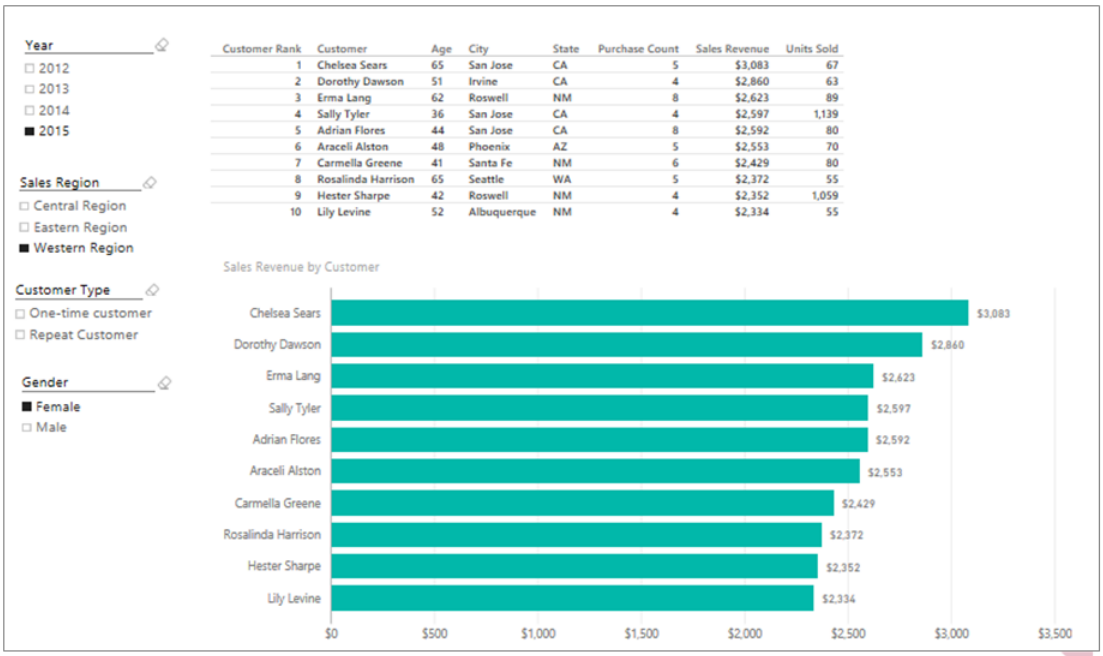



| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Hung Reynolds | 82 | Tucson | AZ | 35 | $8,475 | 353 |
| 2 | Gregory Atkins | 36 | North Hollywood | CA | 34 | $7,378 | 357 |
| 3 | Raymond Holden | 21 | San Francisco | CA | 30 | $7,109 | 318 |
| 4 | Tracie Wise | 31 | Compton | CA | 28 | $7,014 | 315 |
| 5 | Patrick Torres | 23 | Bakersfield | CA | 24 | $6,980 | 275 |
| 6 | Sandy Coleman | 25 | Redmond | WA | 32 | $6,193 | 336 |
| 7 | Marquita Pate | 69 | Scottsdale | AZ | 25 | $6,126 | 276 |
| 8 | Abram Wilson | 27 | Bakersfield | CA | 22 | $5,922 | 259 |
| 9 | Troy Yang | 23 | Springfield | OR | 22 | $5,900 | 226 |
| 10 | Fred Carney | 41 | Pottstown | PA | 25 | $5,834 | 232 |

© Critical Path Training. 2015. All Rights Reserved

# The Finished Report

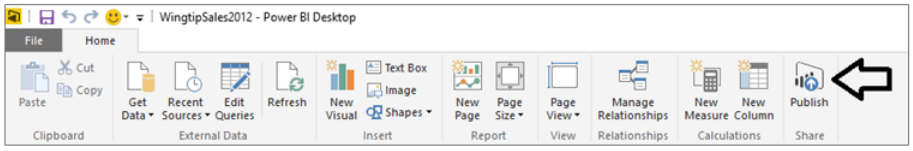



| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Chelsea Sears | 65 | San Jose | CA | 5 | $3,083 | 67 |
| 2 | Dorothy Dawson | 51 | Irvine | CA | 4 | $2,860 | 63 |
| 3 | Erma Lang | 62 | Roswell | NM | 8 | $2,623 | 89 |
| 4 | Sally Tyler | 36 | San Jose | CA | 4 | $2,597 | 1,139 |
| 5 | Adrian Flores | 44 | San Jose | CA | 8 | $2,592 | 80 |
| 6 | Araceli Alston | 48 | Phoenix | AZ | 5 | $2,553 | 70 |
| 7 | Carmella Greene | 41 | Santa Fe | NM | 6 | $2,429 | 80 |
| 8 | Rosalinda Harrison | 65 | Seattle | WA | 5 | $2,372 | 55 |
| 9 | Hester Sharpe | 42 | Roswell | NM | 4 | $2,352 | 1,059 |
| 10 | Lily Levine | 52 | Albuquerque | NM | 4 | $2,334 | 55 |

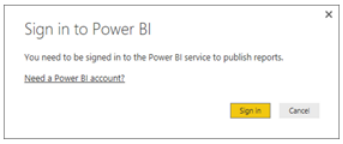
© Critical Path Training. 2015. All Rights Reserved

# Publishing a Power BI Desktop Project

- Power BI Desktop provides **Publish** command
  - Used to publish project to Power BI service




  - Requires logging into your Office 365 account

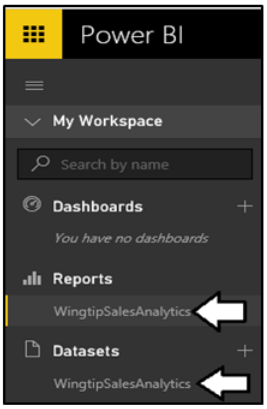


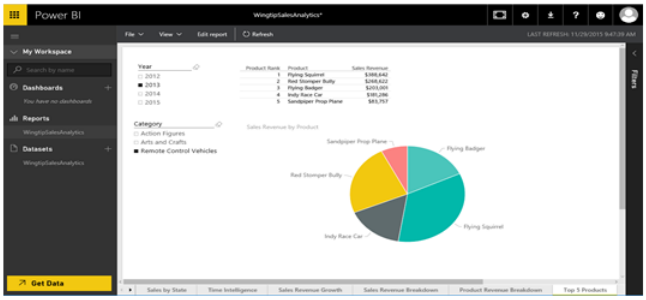
  - Published articles added to your personal workspace

© Critical Path Training. 2015. All Rights Reserved

# Examining What's Been Published

- What does project publishing add to workspace?
  - One dataset with same name as project
  - One report with same name as project

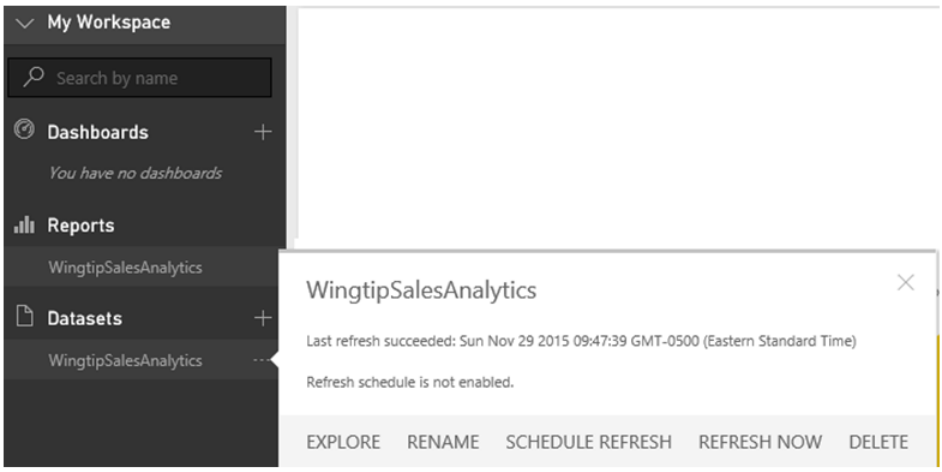



© Critical Path Training. 2015. All Rights Reserved

# Examining Dataset Properties

- Datasets support renaming and refreshing




© Critical Path Training. 2015. All Rights Reserved

# Module 06 Lab: Designing and Publishing Reports with Power BI Desktop

**Lab Time**: 60 minutes

**Lab Folder**: C:\Student\Modules\DesigningReports\Lab

**Lab Overview**: In this module you will continue to extend the Power BI Desktop project named **WingtipSalesAnalytics.pbix** that you have been with working with over the last few labs. In this lab you will focus on designing addition report pages. After creating and designing the several new report pages in the project, you will publish your reports and the underlying dataset to the Power BI service.

**Important:** This lab assumes you have completed the previous lab titled **Extending a Data Model with Time Intelligence** in which you extended the PBIX project with a calendar table and additional measures. If you would like to start this lab without completing the previous lab, copy the lab solution at **C:\Student\Modules\TimeIntelligence\LabSolution\WingtipSalesAnalytics.pbix** into the folder at **C:\Student\Projects** using the Windows Explorer.

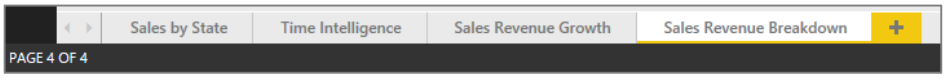
## Exercise 1: Create the Sales Revenue Breakdown Report

In this exercise you will create the **Sales Revenue Breakdown** report to design a new report page that shows how sales revenue breaks down over the last 4 years in areas such as product category, customer type, sales region and purchase type.

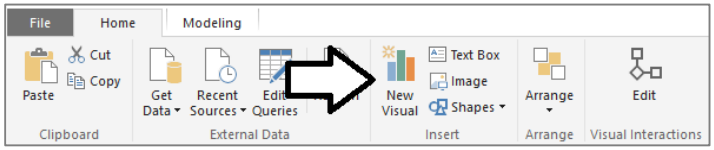
1. Launch Power BI Desktop.
2. Open the Power BI Desktop project named **WingtipSalesAnalytics.pbix** from the previous lab located at the following path.

```
C:\Student\Projects\WingtipSalesAnalytics.pbix
```

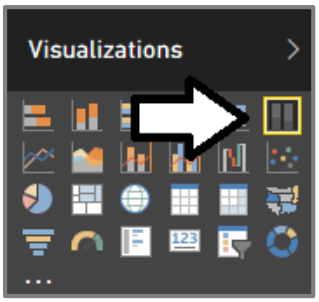
3. When the project opens, click the report icon on the top of the sidebar to enter report view mode.
   a) The project should already contain report pages named **Sales by State**, **Time Intelligence** and **Sales Revenue Growth**.
4. Create a new report page to the project and rename it to **Sales Revenue Breakdown**.




5. Add a new visual to the report to show sales revenue broken down by product category.
   a) Make sure the **Home** tab is active on the ribbon.
   b) Click on the **New Visual** button to add a new visual to the page.

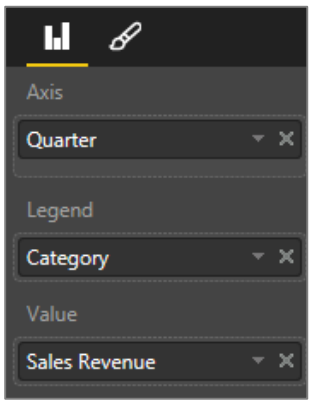



   c) Click the **100% Stacked column chart** button in the **Visualizations** list to change the visualization type.

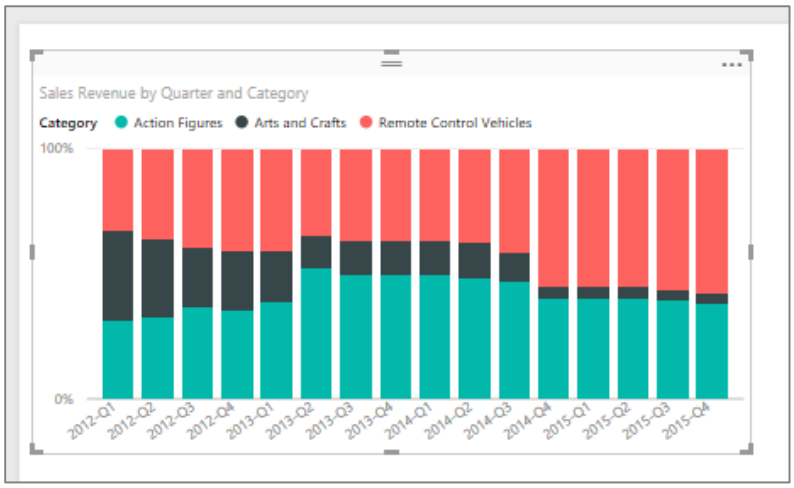



   d) Drag the **Quarter** column from the **Calendar** table in the **Fields** list and drop it into the **Axis** well in the **Visualizations** pane.
   e) Drag the **Category** column from the **Products** table and drop it into the **Legend** well in the **Visualizations** pane.
   f) Drag the **Sales Revenue** measure from the **Sales** table and drop it into the **Value** well in the **Visualizations** pane.

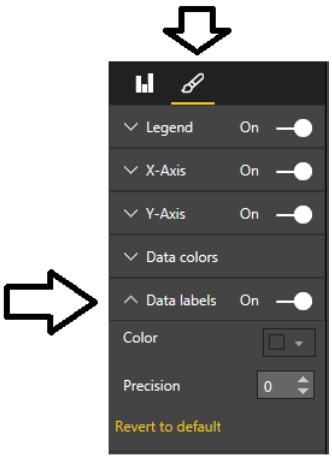
© Critical Path Training. 2015. All Rights Reserved




g) At this point, your visual should match the following screenshot.

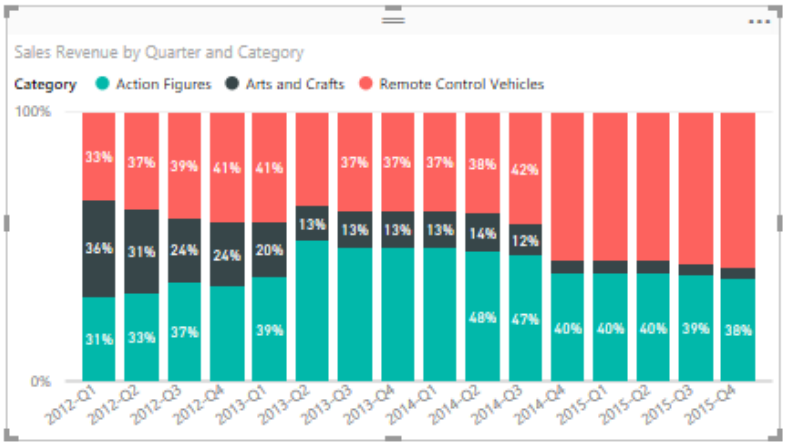



h) Click on the **Edit Brush** icon in the **Visualizations** pane to view the appearance properties for the visual. Locate the **Data labels** property and change its value to **On** as shown in the following screenshot.



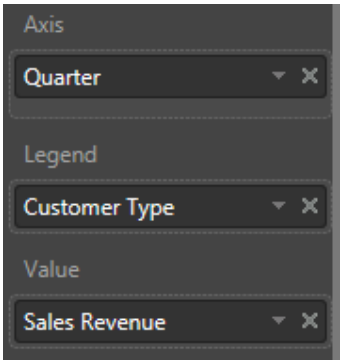

i) Now the visual should display percentage values showing revenue breakdown across categories for each quarter.

© Critical Path Training. 2015. All Rights Reserved

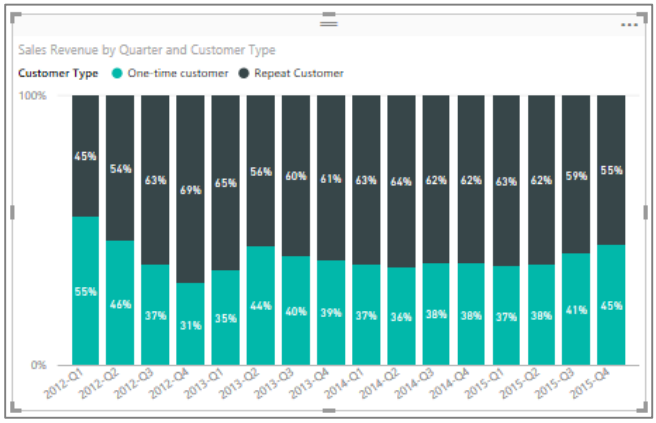



j) Reposition the visual so it takes up the entire upper, left-hand corner of the page.

6. Create a second visual to display a breakdown of sales revenue by customer type.
   a) Select the existing visual and copy it to the Windows clipboard.
   b) Perform a paste operation to add a second copy of the visual to the report page.
   c) Reposition the visual so it takes up the entire lower, left-hand corner of the page.
   d) Make sure the second visual is selected and examine its properties in the **Visualizations** pane.
   e) Remove the **Categories** column from the **Legend** well.
   f) Drag the **Customer Type** column from the **Customers** table and drop it into the **Legend** well in the **Visualizations** pane.

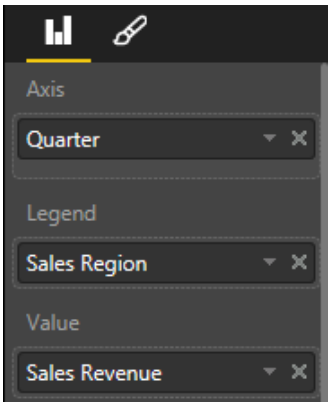



g) The new visual should now match the that is visual shown in the following screenshot.

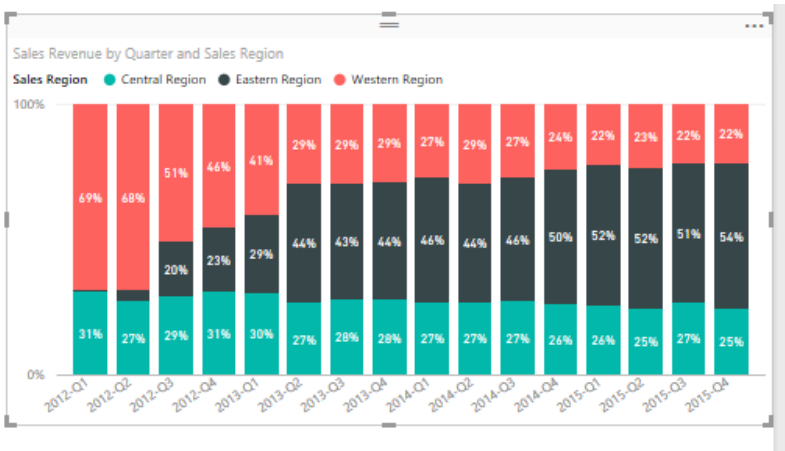



© Critical Path Training. 2015. All Rights Reserved

7. Create a third visual to display a breakdown of sales revenue by sales region.
   a) Select the first visual on the top, left of the page and copy it to the Windows clipboard.
   b) Perform a paste operation to add a new copy of the visual to the report page.
   c) Reposition the visual so it takes up the entire upper, right-hand corner of the page.
   d) Make sure the third visual is selected and examine its properties in the **Visualizations** pane.
   e) Remove the **Categories** column from the **Legend** well.
   f) Drag the **Sales Region** column from the **Customers** table and drop it into the **Legend** well in the **Visualizations** pane.



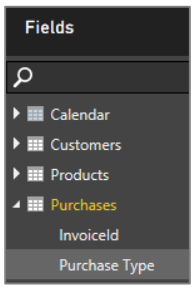

   g) The new visual should now match the visual shown in the following screenshot.



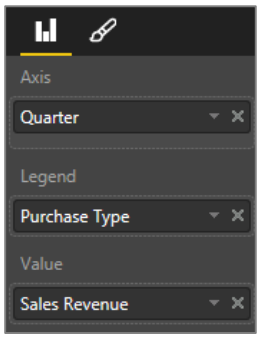

Over the next few steps you will use the **PurchaseType** column of the **Purchases** table to create a new visual. However, this column currently has a name that isn't as user-friendly as it could be. In the next step you will change the column name from **PurchaseType** to **Purchase Type** so the column name displays on report pages in a more user-friendly manner.

8. Change the name of the **PurchaseType** column to **Purchase Type**.
   a) Click the table icon in the middle of the sidebar to enter data view mode.
   b) Look at the **Fields** list and expand the columns inside the **Purchase** table.
   c) Locate the **PurchaseType** column.
   d) Right-click on the **PurchaseType** column in the Fields list and click the **Rename** command.
   e) Rename the column by adding a space so that the new column name is **Purchase Type**.

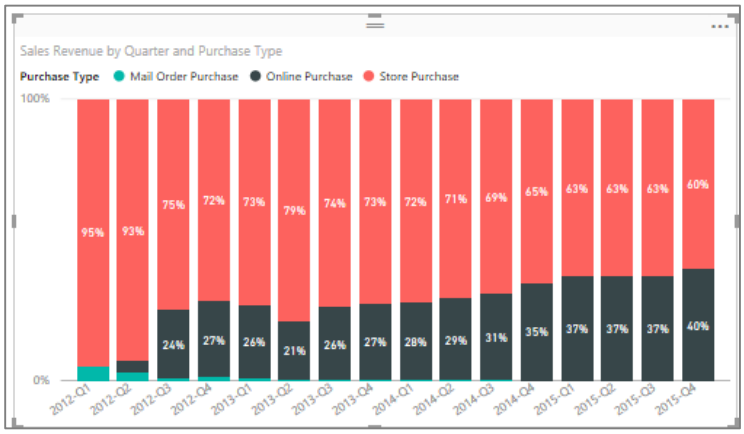
© Critical Path Training. 2015. All Rights Reserved




9. Create a fourth visual to display a breakdown of sales revenue by purchase type.
   a) Click the report icon on the top of the sidebar to navigate back to report view mode.
   b) Select the first visual on the top, left of the page and copy it to the Windows clipboard.
   c) Perform a paste operation to add a new copy of the visual to the report page.
   d) Reposition the visual so it takes up the entire lower, right-hand corner of the page.
   e) Make sure the new visual is selected and examine its properties in the **Visualizations** pane.
   f) Remove the **Categories** column from the **Legend** well.
   g) Drag the **Purchase Type** column from the **Purchases** table and drop it into the **Legend** well in the **Visualizations** pane.




   h) The new visual should now match the visual shown in the following screenshot.




10. Make sure that the four visuals are laid out on the page as shown in the following screenshot.

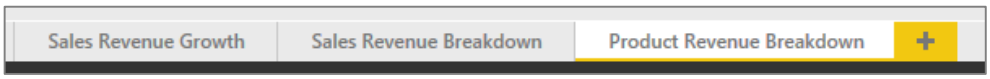
© Critical Path Training. 2015. All Rights Reserved

11. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

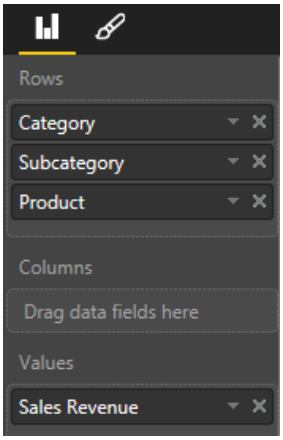
## Exercise 2: Create the Product Revenue Breakdown Report

In this exercise you will create a new report page named **Product Revenue Breakdown**. Creating this report will involve creating new measures to calculate percentages that each product category, subcategory and product contribute to overall sales revenue.

1. Create a new report page and rename it to **Sales Revenue Breakdown**.

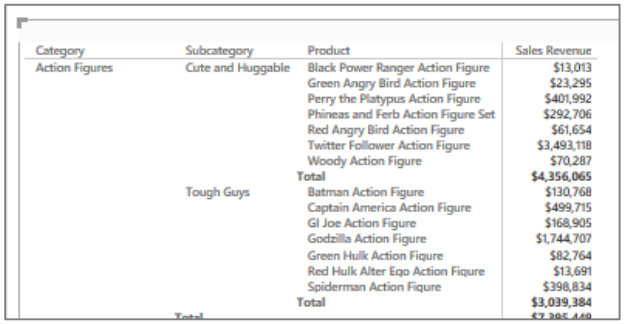



2. Create a new matrix visual to display sales revenue broken out by product category, subcategory and product.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a matrix by clicking the Matrix button in the **Visualizations** list.
   c) Drag and drop the **Category** column from the **Products** table into the **Rows** well.
   d) Drag and drop the **Subcategory** column from the **Products** table into the **Rows** well.
   e) Drag and drop the **Product** column from the **Products** table into the **Rows** well.
   f) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Value** well.



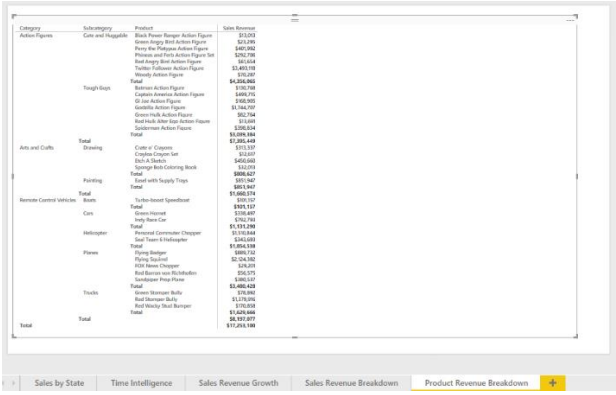

   g) The new visual should now match the visual shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved



| Category | Subcategory | Product | Sales Revenue |
|---|---|---|---|
| Action Figures | Cute and Huggable | Black Power Ranger Action Figure | $13,013 |
| | | Green Angry Bird Action Figure | $23,295 |
| | | Perry the Platypus Action Figure | $401,992 |
| | | Phineas and Ferb Action Figure Set | $292,706 |
| | | Red Angry Bird Action Figure | $61,654 |
| | | Twitter Follower Action Figure | $3,493,118 |
| | | Woody Action Figure | $70,287 |
| | | Total | $4,356,065 |
| | Tough Guys | Batman Action Figure | $130,768 |
| | | Captain America Action Figure | $499,715 |
| | | GI Joe Action Figure | $168,905 |
| | | Godzilla Action Figure | $1,744,707 |
| | | Green Hulk Action Figure | $82,764 |
| | | Red Hulk Alter Ego Action Figure | $13,691 |
| | | Spiderman Action Figure | $398,834 |
| | | Total | $3,039,384 |

h) Reposition the visual so it takes up the height and the width of the entire report page. You need extra width for this visual because you will be adding more columns to it later in this exercise.

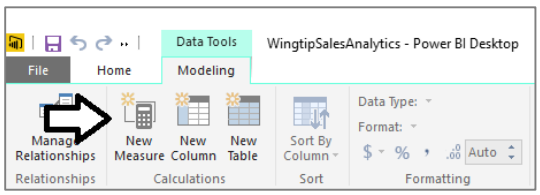



3. Create the **Pct of All Products** measure that calculates the percentage of sales revenue compared to that of all sales revenue.

a) Navigate to data view.

b) Select the **Products** table from the **Fields** list.

c) Create a new measure by clicking the **New Measure** button in the ribbon.



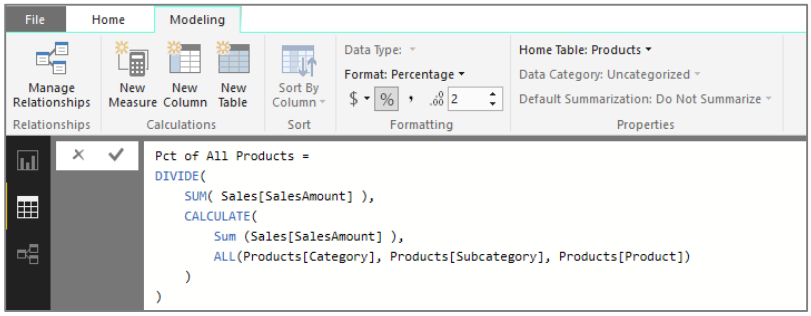

d) Enter to following DAX expression into the formula bar to create the measure named **Pct of All Products**.

```
Pct of All Products =
DIVIDE(
 SUM( Sales[SalesAmount] ),
 CALCULATE(
    Sum ( Sales[SalesAmount] ),
    ALL( Products[Category], Products[Subcategory], Products[Product] )
 )
)
```

e) Press the **ENTER** key to add the measure to the data model.

f) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.

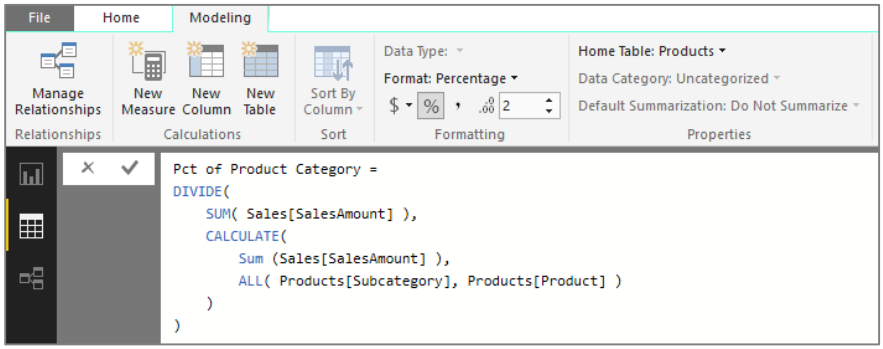
© Critical Path Training. 2015. All Rights Reserved




4. Create the **Pct of Product Category** measure that calculates the percentage of sales revenue within the current product category.
   a) Select the **Products** table from the **Fields** list.
   b) Create a new measure by clicking the **New Measure** button in the ribbon.
   c) Enter to following DAX expression into the formula bar to create the measure named **Pct of Product Category**.

```
Pct of Product Category =
DIVIDE(
 SUM( Sales[SalesAmount] ),
 CALCULATE(
    Sum ( Sales[SalesAmount] ),
    ALL( Products[Subcategory], Products[Product] )
 )
)
```

   d) Press the **ENTER** key to add the measure to the data model.
   e) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.

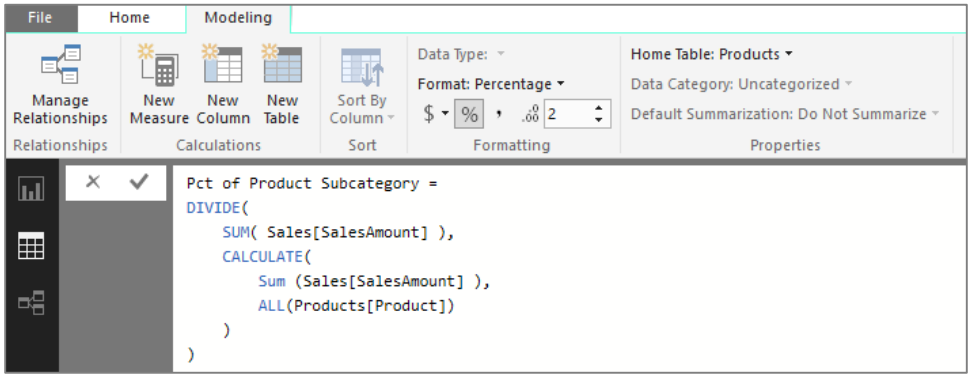



5. Create the **Pct of Product Subcategory** measure that calculates the percentage of sales revenue within the current subcategory.
   a) Select the **Products** table from the **Fields** list.
   b) Create a new measure by clicking the **New Measure** button in the ribbon.
   c) Enter to following DAX expression into the formula bar to create the measure named **Pct of Product Subcategory**.

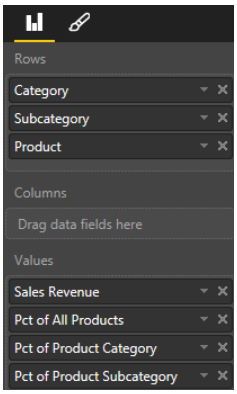
```
Pct of Product Subcategory =
DIVIDE(
 SUM( Sales[SalesAmount] ),
 CALCULATE(
    Sum ( Sales[SalesAmount] ),
    ALL( Products[Product] )
 )
)
```

   d) Press the **ENTER** key to add the measure to the data model.
   e) Modify the formatting by dropping down the **Format** menu on the ribbon and selecting **Percentage**. Also use the spinner control below the format menu to set the number of decimal places shown to **2**.

© Critical Path Training. 2015. All Rights Reserved




6. Modify the matrix visual on the **Sales Revenue Breakdown** report page to include the three new measures.
   a) Click the report icon on the top of the sidebar to enter report view mode.
   b) Make sure that the active report page is the **Sales Revenue Breakdown** report page.
   c) Select the matrix visual that you created earlier in this exercise.
   d) Drag and drop the **Pct of All Products** measure from the **Products** table into the **Values** well.
   e) Drag and drop the **Pct of Product Category** measure from the **Products** table into the **Values** well.
   f) Drag and drop the **Pct of Product Subcategory** measure from the **Products** table into the **Values** well.




   g) Your visual should now match the one shown in the following screenshot. If you examine the values of the **Pct of Product Subcategory** measure, you should see that the products in a subcategory have percentages that sum to 100% for that subcategory. If you examine the values of the **Pct of Product Category** measure, you should see that the products in a category have percentages that sum to 100% for that category.

| Category | Subcategory | Product | Sales Revenue | Pct of All Products | Pct of Product Category | Pct of Product Subcategory |
|---|---|---|---|---|---|---|
| Action Figures | Cute and Huggable | Black Power Ranger Action Figure | $13,013 | 0.08 % | 0.18 % | 0.30 % |
| | | Green Angry Bird Action Figure | $23,295 | 0.14 % | 0.31 % | 0.53 % |
| | | Perry the Platypus Action Figure | $401,992 | 2.33 % | 5.44 % | 9.23 % |
| | | Phineas and Ferb Action Figure Set | $292,706 | 1.70 % | 3.96 % | 6.72 % |
| | | Red Angry Bird Action Figure | $61,654 | 0.36 % | 0.83 % | 1.42 % |
| | | Twitter Follower Action Figure | $3,493,118 | 20.25 % | 47.23 % | 80.19 % |
| | | Woody Action Figure | $70,287 | 0.41 % | 0.95 % | 1.61 % |
| | | **Total** | **$4,356,065** | **25.25 %** | **58.90 %** | **100.00 %** |
| | Tough Guys | Batman Action Figure | $130,768 | 0.76 % | 1.77 % | 4.30 % |
| | | Captain America Action Figure | $499,715 | 2.90 % | 6.76 % | 16.44 % |
| | | GI Joe Action Figure | $168,905 | 0.98 % | 2.28 % | 5.56 % |
| | | Godzilla Action Figure | $1,744,707 | 10.11 % | 23.59 % | 57.40 % |
| | | Green Hulk Action Figure | $82,764 | 0.48 % | 1.12 % | 2.72 % |
| | | Red Hulk Alter Ego Action Figure | $13,691 | 0.08 % | 0.19 % | 0.45 % |
| | | Spiderman Action Figure | $398,834 | 2.31 % | 5.39 % | 13.12 % |
| | | **Total** | **$3,039,384** | **17.62 %** | **41.10 %** | **100.00 %** |
| | **Total** | | **$7,395,449** | **42.86 %** | **100.00 %** | **100.00 %** |

7. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

© Critical Path Training. 2015. All Rights Reserved

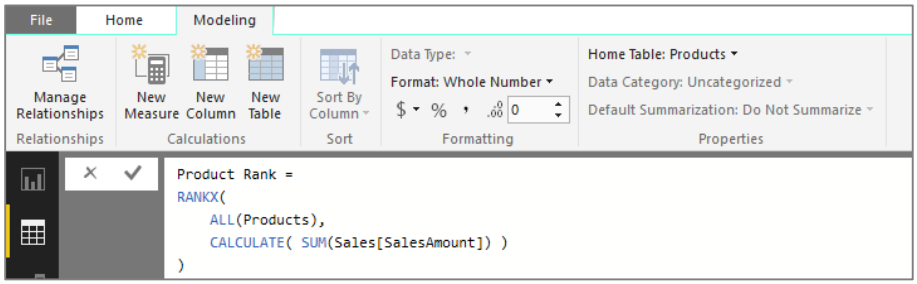
## Exercise 3: Create the Top 5 Products Report

In this exercise you will create a measure named **Product Rank** that ranks products according to their sales revenue. You will then work to create a report that displays the top 5 selling products. You will design this report to be interactive allowing the user to filter on a specific year or a specific product category to see what products are the best sellers.

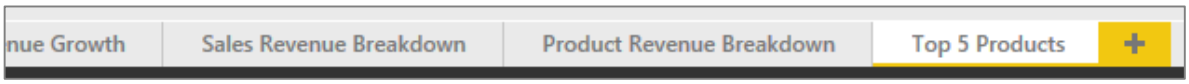
1. Create a new measure named **Product Rank** to determine the top selling products.
   a) Navigate to data view.
   b) Select the **Products** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the measure named **Product Rank**.

```
Product Rank =
RANKX(
 ALL(Products),
 CALCULATE( SUM(Sales[SalesAmount]) )
)
```

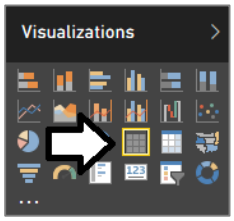
   e) Press the **ENTER** key to add the measure to the data model.
   f) Ensure the formatting for this measure is set to **Whole Number** as shown in the following screenshot.




2. Create a new report page named **Top 5 Products**
   a) Navigate to report view.
   b) Create a new report page and rename it to **Top 5 Products**.

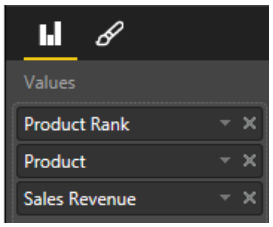



3. Add a new table visual to display the top 5 products.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a matrix by clicking the **Table** button in the **Visualizations** list.

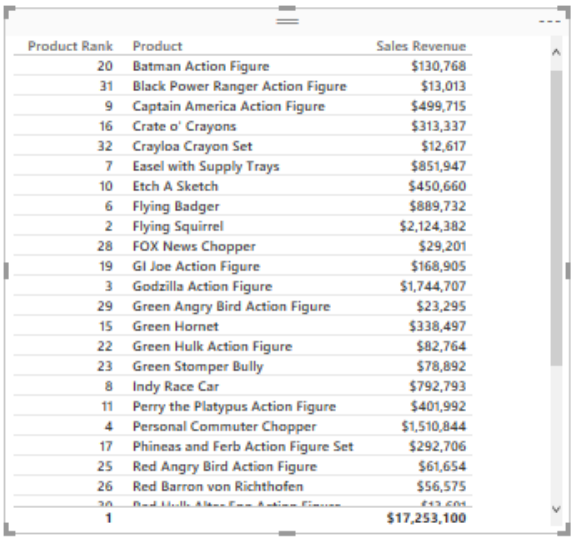



   c) Drag and drop the **Product Rank** measure from the **Products** table into the **Values** well.
   d) Drag and drop the **Product** column from the **Products** table into the **Rows** well.
   e) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Rows** well.

© Critical Path Training. 2015. All Rights Reserved

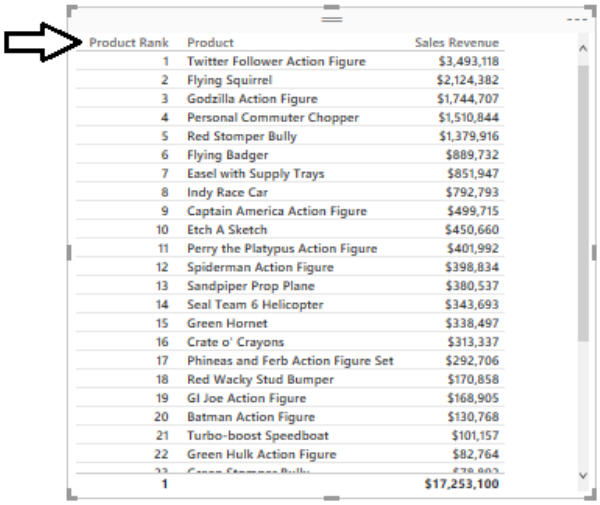



f) The new visual should now match the visual shown in the following screenshot.



| Product Rank | Product | Sales Revenue |
|---|---|---|
| 20 | Batman Action Figure | $130,768 |
| 31 | Black Power Ranger Action Figure | $13,013 |
| 9 | Captain America Action Figure | $499,715 |
| 16 | Crate o' Crayons | $313,337 |
| 32 | Crayloa Crayon Set | $12,617 |
| 7 | Easel with Supply Trays | $851,947 |
| 10 | Etch A Sketch | $450,660 |
| 6 | Flying Badger | $889,732 |
| 2 | Flying Squirrel | $2,124,382 |
| 28 | FOX News Chopper | $29,201 |
| 19 | GI Joe Action Figure | $168,905 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 29 | Green Angry Bird Action Figure | $23,295 |
| 15 | Green Hornet | $338,497 |
| 22 | Green Hulk Action Figure | $82,764 |
| 23 | Green Stomper Bully | $78,892 |
| 8 | Indy Race Car | $792,793 |
| 11 | Perry the Platypus Action Figure | $401,992 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 17 | Phineas and Ferb Action Figure Set | $292,706 |
| 25 | Red Angry Bird Action Figure | $61,654 |
| 26 | Red Barron von Richthofen | $56,575 |
| 1 | | $17,253,100 |

g) Click on the **Product Rank** column header twice to sort the visual so the products with the lowest ranks are sorted to the top.

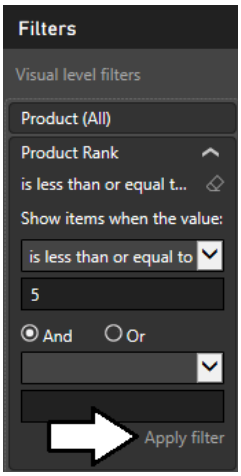


| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Twitter Follower Action Figure | $3,493,118 |
| 2 | Flying Squirrel | $2,124,382 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 5 | Red Stomper Bully | $1,379,916 |
| 6 | Flying Badger | $889,732 |
| 7 | Easel with Supply Trays | $851,947 |
| 8 | Indy Race Car | $792,793 |
| 9 | Captain America Action Figure | $499,715 |
| 10 | Etch A Sketch | $450,660 |
| 11 | Perry the Platypus Action Figure | $401,992 |
| 12 | Spiderman Action Figure | $398,834 |
| 13 | Sandpiper Prop Plane | $380,537 |
| 14 | Seal Team 6 Helicopter | $343,693 |
| 15 | Green Hornet | $338,497 |
| 16 | Crate o' Crayons | $313,337 |
| 17 | Phineas and Ferb Action Figure Set | $292,706 |
| 18 | Red Wacky Stud Bumper | $170,858 |
| 19 | GI Joe Action Figure | $168,905 |
| 20 | Batman Action Figure | $130,768 |
| 21 | Turbo-boost Speedboat | $101,157 |
| 22 | Green Hulk Action Figure | $82,764 |
| 1 | | $17,253,100 |

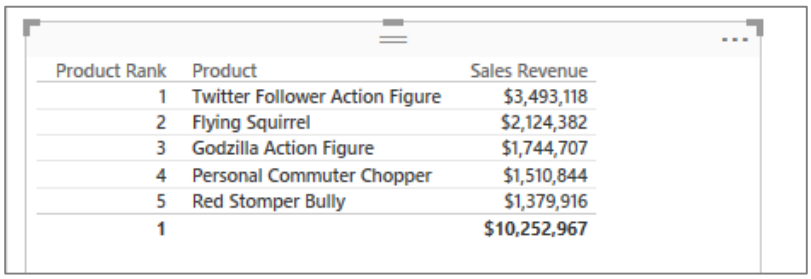
h) Drag and drop the **Product Rank** measure from the **Products** table into **Visual level filters** well of the **Filters** section.

i) Configure the **Product Rank** filter to only display products with a rank of 5 or lower as shown in the following screenshot and then click the **Apply Filter** link to apply the filter to the visual.

© Critical Path Training. 2015. All Rights Reserved

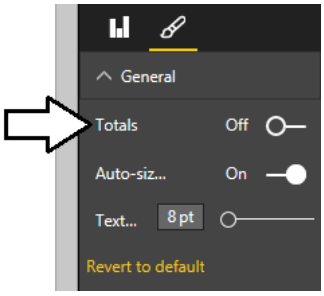



j) Your visual should now display the top 5 selling products as shown in the following screenshot. You should be able to observe that the visual is displaying the **Totals** row at the bottom which needs to be removed.

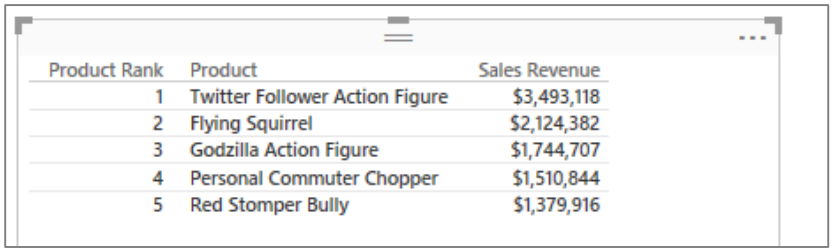


| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Twitter Follower Action Figure | $3,493,118 |
| 2 | Flying Squirrel | $2,124,382 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 5 | Red Stomper Bully | $1,379,916 |
| **1** | | **$10,252,967** |

k) Locate the **Totals** property for the table visual in the **General** section of the property sheet and set its value to **Off**.



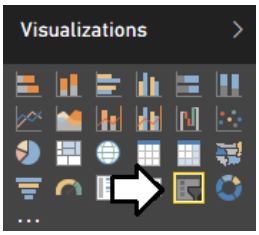

l) Your top 5 products visual should now look better when it is displayed without the **Totals** row.



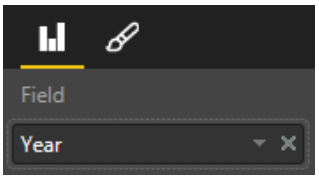
| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Twitter Follower Action Figure | $3,493,118 |
| 2 | Flying Squirrel | $2,124,382 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 5 | Red Stomper Bully | $1,379,916 |

4. Add a new slicer visual to the page to filter the top 5 products visual by **Year**.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.

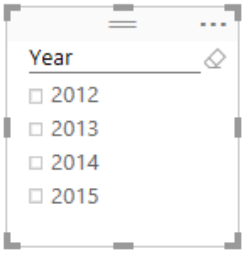
© Critical Path Training. 2015. All Rights Reserved



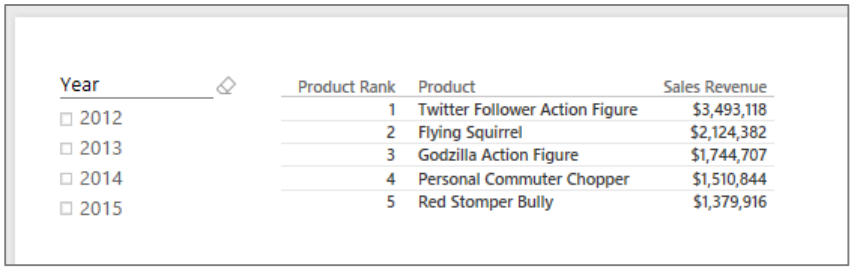

c) Drag and drop the **Year** column from the **Sales** table into the **Values** well.




d) You should now have a slicer visual that matches the following screenshot.




e) Reposition the two visuals on the page to match the following screenshot.



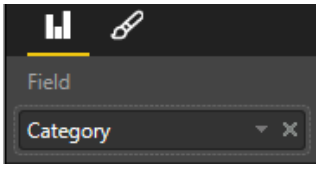

f) Try using the **Year** slicer by selecting individual years. You should see that the visual with the top 5 products changes whenever you select a different year.

5. Add a second slicer visual to the top five products visual by **Product Category**.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.

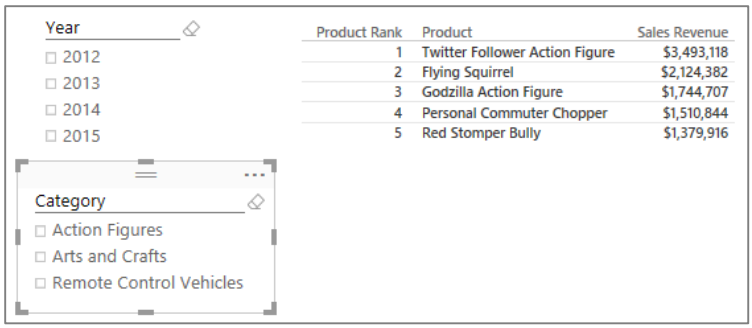
© Critical Path Training. 2015. All Rights Reserved

b) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.

c) Drag and drop the **Category** column from the **Products** table into the **Values** well.




d) Reposition the new visual to match the page layout shown in the following screenshot.




e) Try using the **Category** slicer by selecting individual product categories. You should see that there is now a problem with the report because the visual with the top 5 products doesn't show 5 products.

The problem you are facing here has to do with the manner in which the **Product Rank** measure is filtering during its evaluation. The problem is that the measure does not correctly filter by the product category column when determining the top 5 products. Therefore, you must modify the DAX expression for the **Product Rank** measure in order to calculate the top 5 selling products within a specific category when that category is selected in the slicer.

6. Modify the DAX expressions for the **Product Rank** measure to correct the filter problem with product category.

a) Navigate to data view.

b) Expand the **Products** table from the **Fields** list.

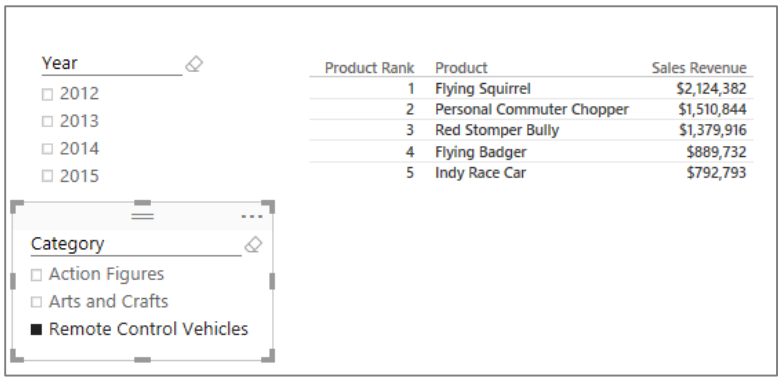
c) Select the **Product Rank** measure in the **Products** table so you can view and modify its DAX expression in the formula bar.

d) Modify the DAX expression for the **Product Rank** measure to match the following code listing.

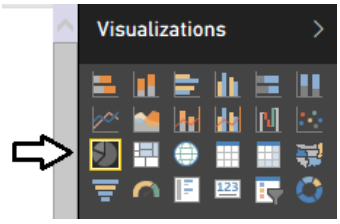
```
Product Rank =
IF(
  HASONEVALUE(Products[Product]),
  RANKX(
    ALL( Products[Subcategory], Products[Product] ),
    CALCULATE( SUM(Sales[SalesAmount]) )
  )
)
```

© Critical Path Training. 2015. All Rights Reserved

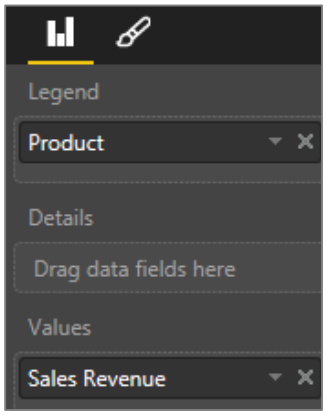
7. Test the changes you made to the **Product Rank** measure.
   a) Navigate to report view.
   b) Test the measure by selecting different categories using the **Category** slicer. At this point, the page filtering should be working correctly as you should see 5 top products when selecting a product category.




8. Add a new Pie chart visual to display the top 5 products.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a pie chart by clicking the Pie Chart button in the **Visualizations** list.

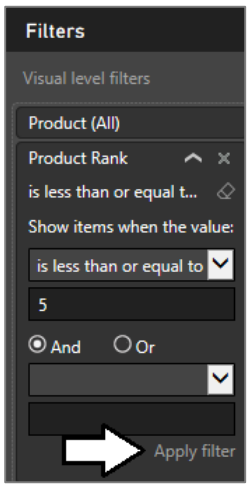



   c) Drag and drop the **Product** column from the **Products** table into the **Legend** well.
   d) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Values** well.

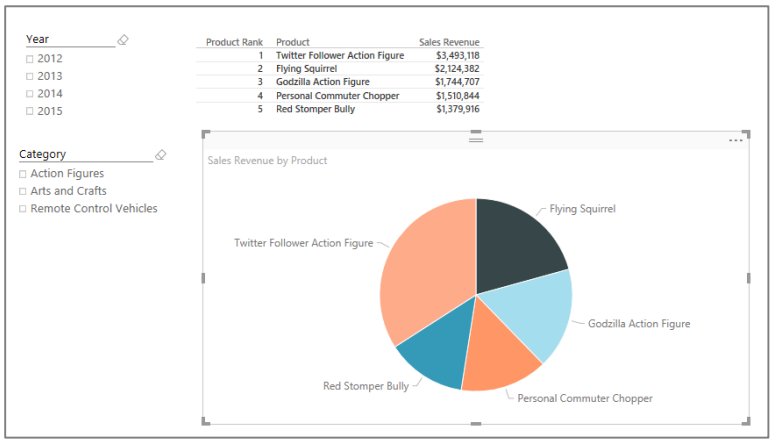



   e) Drag and drop the **Product Rank** measure from the **Products** table into **Visual level filters** well of the **Filters** section.
   f) Configure the **Product Rank** filter to only display products with a rank of 5 or lower as shown in the following screenshot and then click the **Apply Filter** link to apply the filter to the visual.

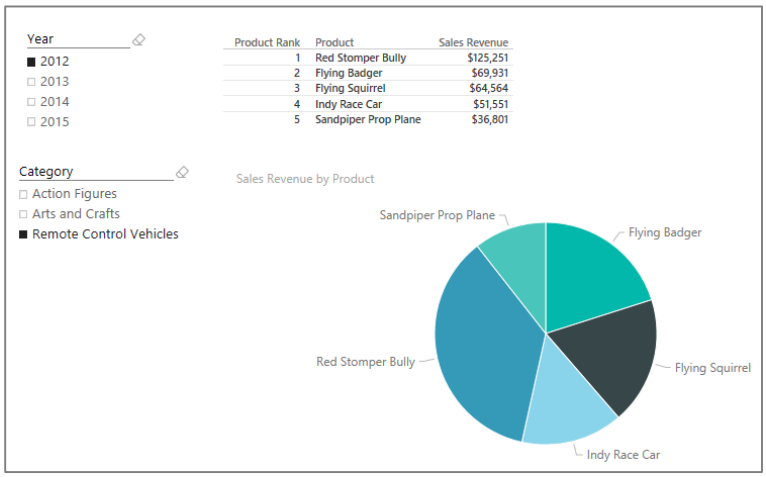
© Critical Path Training. 2015. All Rights Reserved




g) Reposition the new visual to match the page layout shown in the following screenshot.




h) Test your work by using the two slicer visuals on the page to select different combinations of years and product category. Both the table and the pie chart with the top 5 products should update together and stay in sync as you change the filter selection.




9. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

© Critical Path Training. 2015. All Rights Reserved

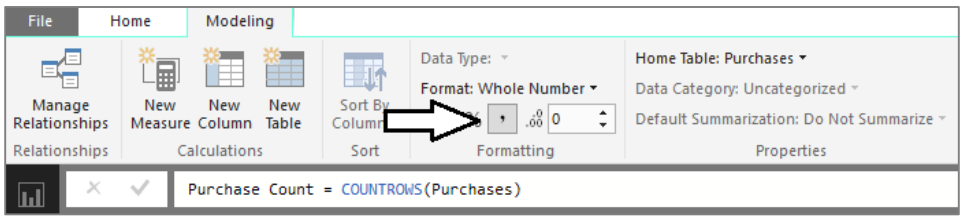
## Exercise 4: Create the Top 10 Customers Report

In this exercise you will create a measure named **Customer Rank** that ranks customers according to their sales revenue. You will then work to create a report that displays the top 10 customers. You will also design this report to be interactive allowing the user to filter on a specific year or a specific product category to see what products are the best sellers.

1. Create a new measure named **Purchase Count** to determine the number of purchases.
   a) Navigate to data view.
   b) Select the **Purchases** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the measure named **Purchase Count**.

```
Purchase Count = COUNTROWS(Purchases)
```

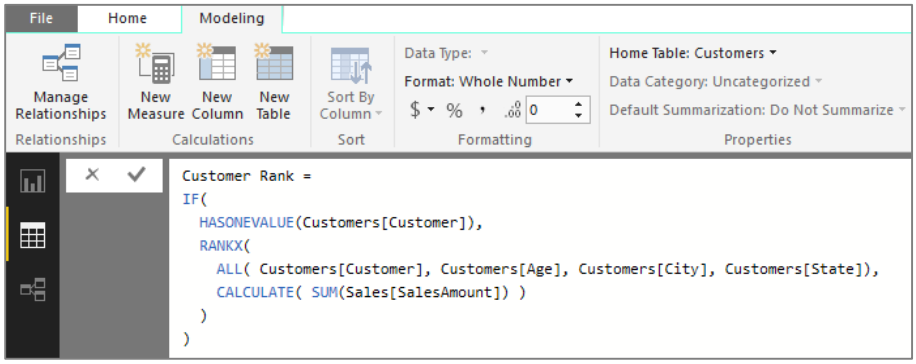
   e) Press the **ENTER** key to add the measure to the data model.
   f) Ensure the formatting for this measure is set to **Whole Number** as shown in the following screenshot. Also check the comma button to format values over 1000 with a comma separator.




2. Create a new measure named **Customer Rank** to determine the top ranked customers with respect to sales revenue.
   a) Navigate to data view if you are not already there.
   b) Select the **Customers** table from the **Fields** list.
   c) Create a new measure by clicking the **New Measure** button in the ribbon.
   d) Enter to following DAX expression into the formula bar to create the measure named **Customer Rank**.

```
Customer Rank =
IF(
  HASONEVALUE(Customers[Customer]),
  RANKX(
    ALL( Customers[Customer], Customers[Age], Customers[City], Customers[State]),
    CALCULATE( SUM(Sales[SalesAmount]) )
  )
)
```

   e) Press the **ENTER** key to add the measure to the data model.
   f) Ensure the formatting for this measure is set to **Whole Number** as shown in the following screenshot.

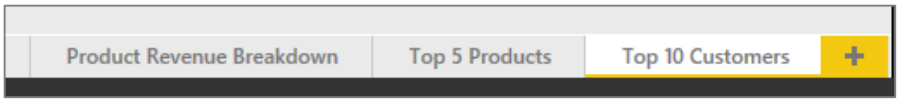



3. Create a new report page named **Top 10 Customers**.

© Critical Path Training. 2015. All Rights Reserved

a) Navigate to report view.

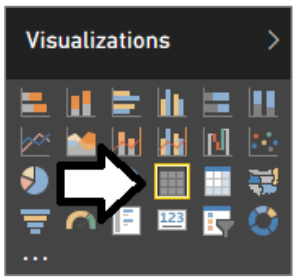
b) Create a new report page and rename it to **Top 10 Customers**.




4. Add a new table visual to display the top 10 customers.

a) Click the **New Visual** button on the ribbon to add a new visual to the page.

b) Change the visual to a table by clicking the **Table** button in the **Visualizations** list.




c) Drag and drop the **Customer Rank** measure from the **Customers** table into the **Values** well.

d) Drag and drop the **Customer** column from the **Customers** table into the **Values** well.

e) Drag and drop the **Age** column from the **Customers** table into the **Values** well.

f) Drag and drop the **City** column from the **Customers** table into the **Values** well.

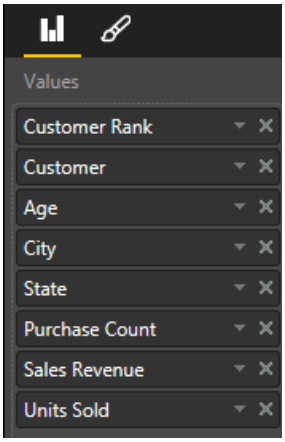
g) Drag and drop the **State** column from the **Customers** table into the **Values** well.

h) Drag and drop the **Purchase Count** measure from the **Products** table into the **Values** well.

i) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Values** well.

j) Drag and drop the **Units Sold** measure from the **Sales** table into the **Values** well.

k) The **Values** well for your visual should match the following screenshot.




l) The new visual should now display as the visual shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

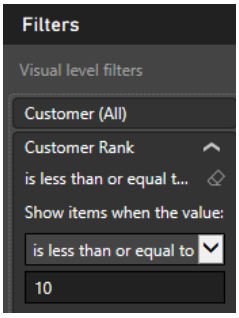
| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 11131 | Aaron Cardenas | 58 | Dorchester | MA | 1 | $439 | 22 |
| 12383 | Aaron Combs | 21 | Burbank | CA | 2 | $375 | 61 |
| 6276 | Aaron Gonzalez | 51 | Greenville | NC | 2 | $988 | 41 |
| 26722 | Aaron Moore | 71 | Houston | TX | 1 | $42 | 3 |
| 2977 | Aaron Mullins | 44 | Napa | CA | 3 | $1,477 | 1,015 |
| 4526 | Aaron Nelson | 41 | North Haven | CT | 1 | $1,172 | 1,011 |
| 16048 | Abby Benton | 59 | Tucson | AZ | 1 | $257 | 107 |
| 24778 | Abby Buckley | 75 | Glen Mills | PA | 1 | $85 | 28 |
| 25857 | Abby Burt | 23 | Colorado Springs | CO | 1 | $61 | 7 |
| 15884 | Abby Cook | 35 | Waterbury | CT | 1 | $260 | 108 |
| 6896 | Abby Doyle | 34 | Spokane | WA | 2 | $872 | 28 |
| 19636 | Abby Jarvis | 66 | Cincinnati | OH | 1 | $179 | 11 |
| 10260 | Abby Lindsey | 67 | Dallas | TX | 1 | $500 | 10 |
| 6899 | Abby Mooney | 35 | Braintree | MA | 1 | $871 | 13 |
| 16751 | Abby Olson | 71 | Tallahassee | FL | 1 | $242 | 114 |
| 24700 | Abdul Cote | 47 | Amarillo | TX | 1 | $88 | 4 |
| 21478 | Abdul Dillard | 25 | Bakersfield | CA | 1 | $140 | 7 |
| 23569 | Abdul Dudley | 70 | Raleigh | NC | 1 | $100 | 100 |
| 6969 | Abdul Ellison | 20 | Knoxville | TN | 7 | $859 | 55 |
| 24055 | Abdul Le | 50 | Allentown | PA | 1 | $100 | 5 |
| Total | | | | | 66,043 | $17,253,100 | 4,032,502 |

m) Click on the **Customer Rank** column header twice to sort the visual so the customers with the greatest amount of sales revenue are sorted to the top.

| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Hung Reynolds | 82 | Tucson | AZ | 35 | $8,475 | 353 |
| 2 | Gregory Atkins | 36 | North Hollywood | CA | 34 | $7,378 | 357 |
| 3 | Raymond Hol... | 21 | San Francisco | CA | 30 | $7,109 | 318 |
| 4 | Tracie Wise | 31 | Compton | CA | 28 | $7,014 | 315 |
| 5 | Patrick Torres | 23 | Bakersfield | CA | 24 | $6,980 | 275 |
| 6 | Sandy Coleman | 25 | Redmond | WA | 32 | $6,193 | 336 |
| 7 | Marquita Pate | 69 | Scottsdale | AZ | 25 | $6,126 | 276 |
| 8 | Abram Wilson | 27 | Bakersfield | CA | 22 | $5,922 | 259 |
| 9 | Troy Yang | 23 | Springfield | OR | 22 | $5,900 | 226 |
| 10 | Fred Carney | 41 | Pottstown | PA | 25 | $5,834 | 232 |
| 11 | Luisa Strickland | 38 | Bend | OR | 25 | $5,802 | 247 |
| 12 | Victor Puckett | 52 | Eureka | CA | 21 | $5,748 | 229 |
| 13 | Marco Bradsh... | 72 | Dallas | TX | 25 | $5,721 | 312 |
| 14 | Ernesto Bernard | 29 | Vancouver | WA | 28 | $5,703 | 320 |
| 14 | Max Simpson | 63 | Rochester | NY | 11 | $5,703 | 1,127 |
| 16 | Sandy Vincent | 35 | San Jose | CA | 25 | $5,571 | 281 |
| 17 | Neva Stephen... | 35 | Bellevue | WA | 25 | $5,564 | 288 |
| 18 | Chadwick Joh... | 33 | San Francisco | CA | 22 | $5,561 | 219 |
| 19 | Helga Simmons | 45 | Scottsdale | AZ | 19 | $5,445 | 233 |
| 20 | Bobbie Steph... | 56 | Denver | CO | 22 | $5,410 | 1,216 |
| Total | | | | | 66,043 | $17,253,100 | 4,032,502 |

n) Drag and drop the **Customer Rank** measure from the **Customers** table into **Visual level filters** well of the **Filters** section.

o) Configure the **Customer Rank** filter to only display customers with a rank of 10 or lower as shown in the following screenshot and then click the **Apply Filter** link to apply the filter to the visual.

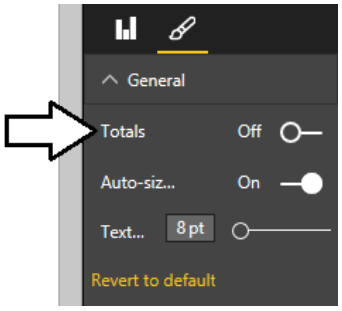



p) Your visual should now display the top 10 customers as shown in the following screenshot. Note that the visual is still showing the **Totals** row at the bottom which needs to be removed.

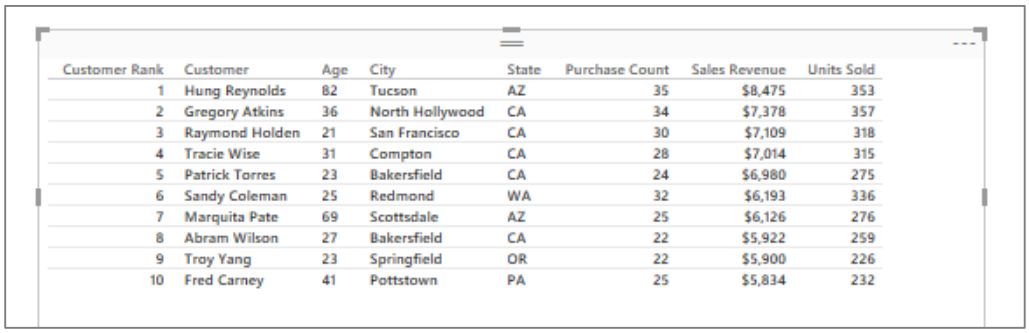
| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Hung Reynolds | 82 | Tucson | AZ | 35 | $8,475 | 353 |
| 2 | Gregory Atkins | 36 | North Hollywood | CA | 34 | $7,378 | 357 |
| 3 | Raymond Holden | 21 | San Francisco | CA | 30 | $7,109 | 318 |
| 4 | Tracie Wise | 31 | Compton | CA | 28 | $7,014 | 315 |
| 5 | Patrick Torres | 23 | Bakersfield | CA | 24 | $6,980 | 275 |
| 6 | Sandy Coleman | 25 | Redmond | WA | 32 | $6,193 | 336 |
| 7 | Marquita Pate | 69 | Scottsdale | AZ | 25 | $6,126 | 276 |
| 8 | Abram Wilson | 27 | Bakersfield | CA | 22 | $5,922 | 259 |
| 9 | Troy Yang | 23 | Springfield | OR | 22 | $5,900 | 226 |
| 10 | Fred Carney | 41 | Pottstown | PA | 25 | $5,834 | 232 |
| Total | | | | | 277 | $66,930 | 2,947 |

© Critical Path Training. 2015. All Rights Reserved

q) Locate the **Totals** property in the **General** section of the property sheet for the table visual and set it to a value of **Off**.




r) You visual should now look better when it is displayed without the **Totals** row.

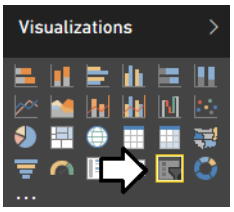


| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Hung Reynolds | 82 | Tucson | AZ | 35 | $8,475 | 353 |
| 2 | Gregory Atkins | 36 | North Hollywood | CA | 34 | $7,378 | 357 |
| 3 | Raymond Holden | 21 | San Francisco | CA | 30 | $7,109 | 318 |
| 4 | Tracie Wise | 31 | Compton | CA | 28 | $7,014 | 315 |
| 5 | Patrick Torres | 23 | Bakersfield | CA | 24 | $6,980 | 275 |
| 6 | Sandy Coleman | 25 | Redmond | WA | 32 | $6,193 | 336 |
| 7 | Marquita Pate | 69 | Scottsdale | AZ | 25 | $6,126 | 276 |
| 8 | Abram Wilson | 27 | Bakersfield | CA | 22 | $5,922 | 259 |
| 9 | Troy Yang | 23 | Springfield | OR | 22 | $5,900 | 226 |
| 10 | Fred Carney | 41 | Pottstown | PA | 25 | $5,834 | 232 |

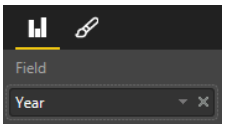
5. Add a new slicer visual to the page to filter the top 10 customers visual by **Year**.

a) Click the **New Visual** button on the ribbon to add a new visual to the page.

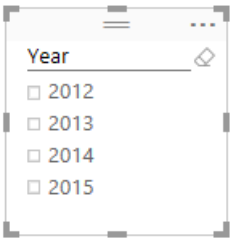
b) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.




c) Drag and drop the **Year** column from the **Sales** table into the **Values** well.




d) You should now have a slicer visual that matches the following screenshot.

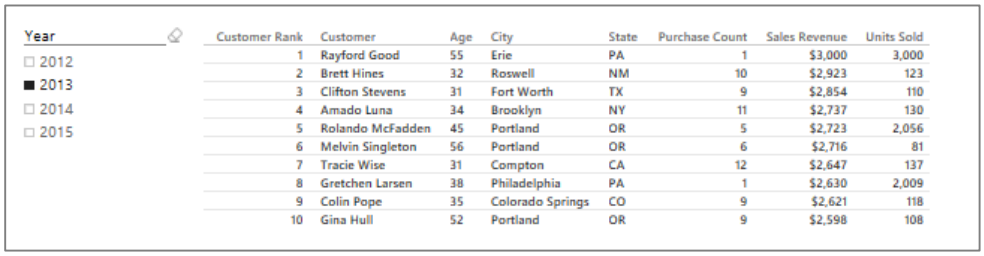



© Critical Path Training. 2015. All Rights Reserved

e) Reposition the two visuals on the page to match the following screenshot.

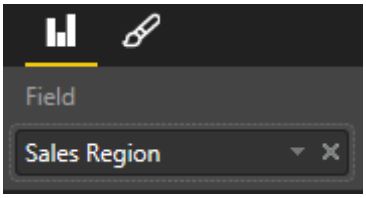
| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Hung Reynolds | 82 | Tucson | AZ | 35 | $8,475 | 353 |
| 2 | Gregory Atkins | 36 | North Hollywood | CA | 34 | $7,378 | 357 |
| 3 | Raymond Holden | 21 | San Francisco | CA | 30 | $7,109 | 318 |
| 4 | Tracie Wise | 31 | Compton | CA | 28 | $7,014 | 315 |
| 5 | Patrick Torres | 23 | Bakersfield | CA | 24 | $6,980 | 275 |
| 6 | Sandy Coleman | 25 | Redmond | WA | 32 | $6,193 | 336 |
| 7 | Marquita Pate | 69 | Scottsdale | AZ | 25 | $6,126 | 276 |
| 8 | Abram Wilson | 27 | Bakersfield | CA | 22 | $5,922 | 259 |
| 9 | Troy Yang | 23 | Springfield | OR | 22 | $5,900 | 226 |
| 10 | Fred Carney | 41 | Pottstown | PA | 25 | $5,834 | 232 |

f) Try using the slicer by selecting individual years. You should see that the visual with the top 10 customers list changes when you select a different year.

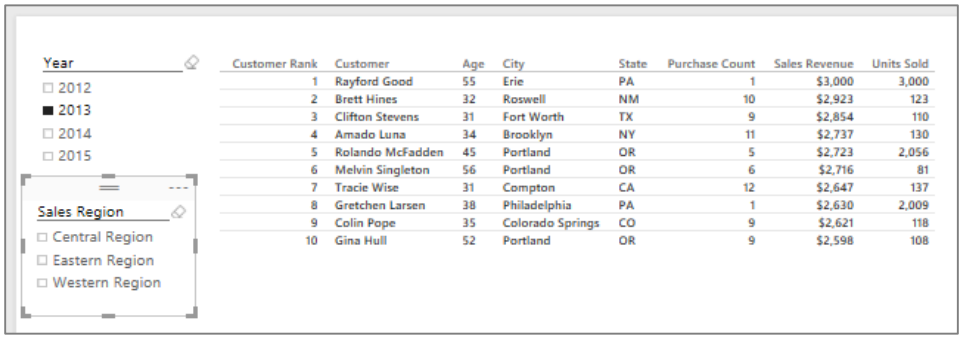



| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Rayford Good | 55 | Erie | PA | 1 | $3,000 | 3,000 |
| 2 | Brett Hines | 32 | Roswell | NM | 10 | $2,923 | 123 |
| 3 | Clifton Stevens | 31 | Fort Worth | TX | 9 | $2,854 | 110 |
| 4 | Amado Luna | 34 | Brooklyn | NY | 11 | $2,737 | 130 |
| 5 | Rolando McFadden | 45 | Portland | OR | 5 | $2,723 | 2,056 |
| 6 | Melvin Singleton | 56 | Portland | OR | 6 | $2,716 | 81 |
| 7 | Tracie Wise | 31 | Compton | CA | 12 | $2,647 | 137 |
| 8 | Gretchen Larsen | 38 | Philadelphia | PA | 1 | $2,630 | 2,009 |
| 9 | Colin Pope | 35 | Colorado Springs | CO | 9 | $2,621 | 118 |
| 10 | Gina Hull | 52 | Portland | OR | 9 | $2,598 | 108 |

6. Add a second slicer visual to filter the top 10 customers visual by **Sales Region**.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.
   c) Drag and drop the **Category** column from the **Products** table into the **Values** well.



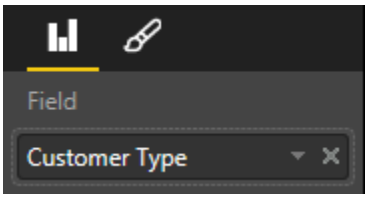

   d) Reposition the new visual to match the page layout shown in the following screenshot.




| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Rayford Good | 55 | Erie | PA | 1 | $3,000 | 3,000 |
| 2 | Brett Hines | 32 | Roswell | NM | 10 | $2,923 | 123 |
| 3 | Clifton Stevens | 31 | Fort Worth | TX | 9 | $2,854 | 110 |
| 4 | Amado Luna | 34 | Brooklyn | NY | 11 | $2,737 | 130 |
| 5 | Rolando McFadden | 45 | Portland | OR | 5 | $2,723 | 2,056 |
| 6 | Melvin Singleton | 56 | Portland | OR | 6 | $2,716 | 81 |
| 7 | Tracie Wise | 31 | Compton | CA | 12 | $2,647 | 137 |
| 8 | Gretchen Larsen | 38 | Philadelphia | PA | 1 | $2,630 | 2,009 |
| 9 | Colin Pope | 35 | Colorado Springs | CO | 9 | $2,621 | 118 |
| 10 | Gina Hull | 52 | Portland | OR | 9 | $2,598 | 108 |

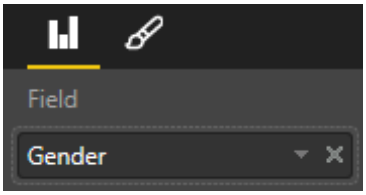
7. Add a third slicer visual to filter the top 10 customers visual by **Customer Type**.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.
   c) Drag and drop the **Customer Type** column from the **Customers** table into the **Values** well.

© Critical Path Training. 2015. All Rights Reserved

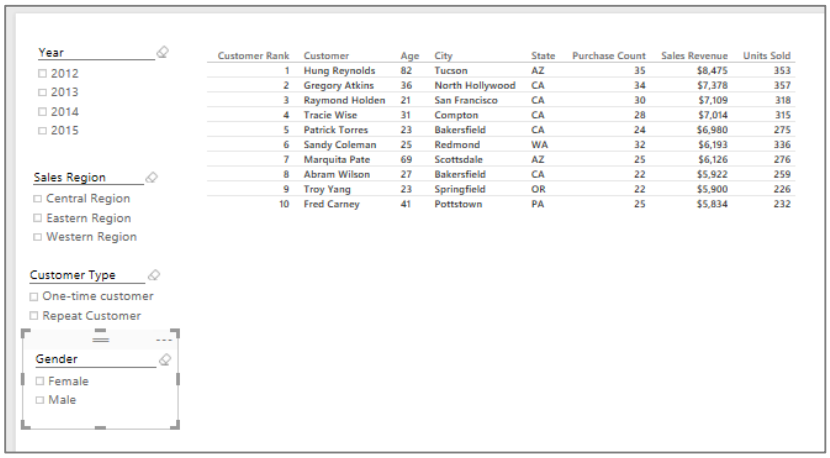



d) Reposition the new visual to match the page layout shown in the following screenshot.

8. Add a fourth slicer visual to filter the top 10 customers visual by **Gender**.
   a) Click the **New Visual** button on the ribbon to add a new visual to the page.
   b) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.
   c) Drag and drop the **Gender** column from the **Customers** table into the **Values** well.




d) Reposition the new visual to match the page layout shown in the following screenshot.

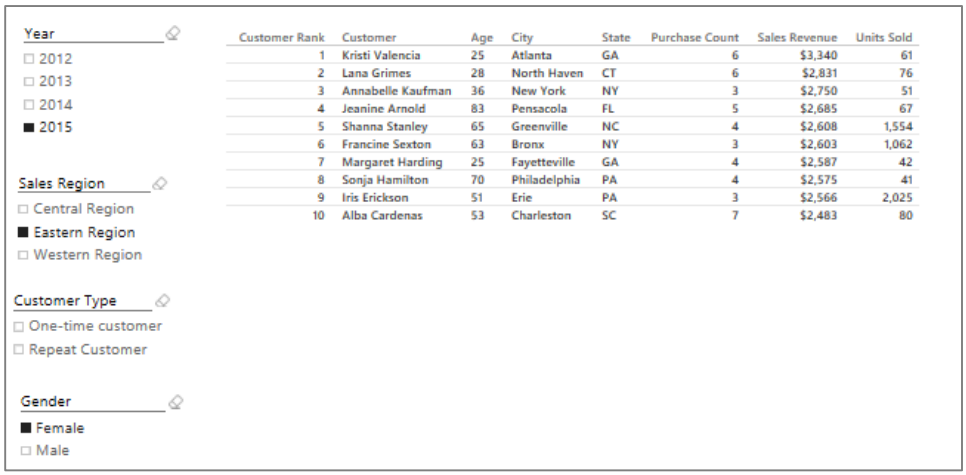



© Critical Path Training. 2015. All Rights Reserved

9. Now interact with the slicers on the page to answer the following questions.
   a) Who were the top 10 customers in 2014 who were one-time customers?

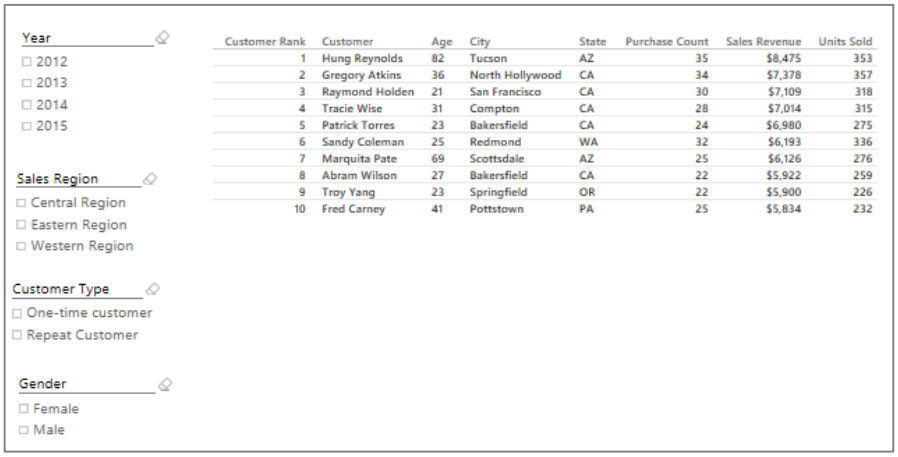
| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Mitzi Franco | 71 | Tucson | AZ | 1 | $2,540 | 2,009 |
| 2 | Earl Merrill | 63 | Grand Junction | CO | 1 | $2,400 | 2,004 |
| 3 | Tamra White | 32 | Riverdale | NJ | 1 | $2,270 | 2,009 |
| 4 | Nelson Bowers | 62 | Seattle | WA | 1 | $2,200 | 2,008 |
| 5 | Vonda Marks | 35 | Charlotte | NC | 1 | $2,140 | 2,002 |
| 6 | Bradley Burns | 47 | Tuscaloosa | AL | 1 | $2,117 | 2,009 |
| 7 | Juliet Bass | 62 | Clark | NJ | 1 | $2,104 | 2,008 |
| 8 | Quinn Wooten | 44 | Pittsburgh | PA | 1 | $2,100 | 2,100 |
| 9 | Vickie Wong | 48 | Birmingham | AL | 1 | $2,065 | 2,005 |
| 10 | Hassan Slater | 54 | Miami | FL | 1 | $2,040 | 2,002 |
| 10 | Liliana Prince | 43 | Scottsdale | AZ | 1 | $2,040 | 2,002 |

   b) Who were the top 10 female customers from the Eastern sales region in 2015?




| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Kristi Valencia | 25 | Atlanta | GA | 6 | $3,340 | 61 |
| 2 | Lana Grimes | 28 | North Haven | CT | 6 | $2,831 | 76 |
| 3 | Annabelle Kaufman | 36 | New York | NY | 3 | $2,750 | 51 |
| 4 | Jeanine Arnold | 83 | Pensacola | FL | 5 | $2,685 | 67 |
| 5 | Shanna Stanley | 65 | Greenville | NC | 4 | $2,608 | 1,554 |
| 6 | Francine Sexton | 63 | Bronx | NY | 3 | $2,603 | 1,062 |
| 7 | Margaret Harding | 25 | Fayetteville | GA | 4 | $2,587 | 42 |
| 8 | Sonja Hamilton | 70 | Philadelphia | PA | 4 | $2,575 | 41 |
| 9 | Iris Erickson | 51 | Erie | PA | 3 | $2,566 | 2,025 |
| 10 | Alba Cardenas | 53 | Charleston | SC | 7 | $2,483 | 80 |

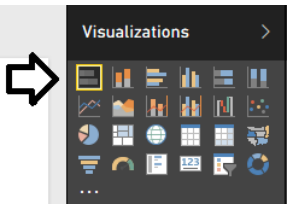
10. Clear the filter on all the slicers on the page so the table shows results for all sales.




| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Hung Reynolds | 82 | Tucson | AZ | 35 | $8,475 | 353 |
| 2 | Gregory Atkins | 36 | North Hollywood | CA | 34 | $7,378 | 357 |
| 3 | Raymond Holden | 21 | San Francisco | CA | 30 | $7,109 | 318 |
| 4 | Tracie Wise | 31 | Compton | CA | 28 | $7,014 | 315 |
| 5 | Patrick Torres | 23 | Bakersfield | CA | 24 | $6,980 | 275 |
| 6 | Sandy Coleman | 25 | Redmond | WA | 32 | $6,193 | 336 |
| 7 | Marquita Pate | 69 | Scottsdale | AZ | 25 | $6,126 | 276 |
| 8 | Abram Wilson | 27 | Bakersfield | CA | 22 | $5,922 | 259 |
| 9 | Troy Yang | 23 | Springfield | OR | 22 | $5,900 | 226 |
| 10 | Fred Carney | 41 | Pottstown | PA | 25 | $5,834 | 232 |

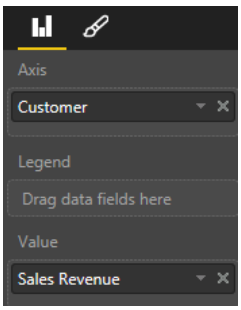
11. Add a new bar chart to show the sales revenue breakdown for the top 10 customers.
   a) Change the visual to a slicer by clicking the Slicer button in the **Visualizations** list.

© Critical Path Training. 2015. All Rights Reserved

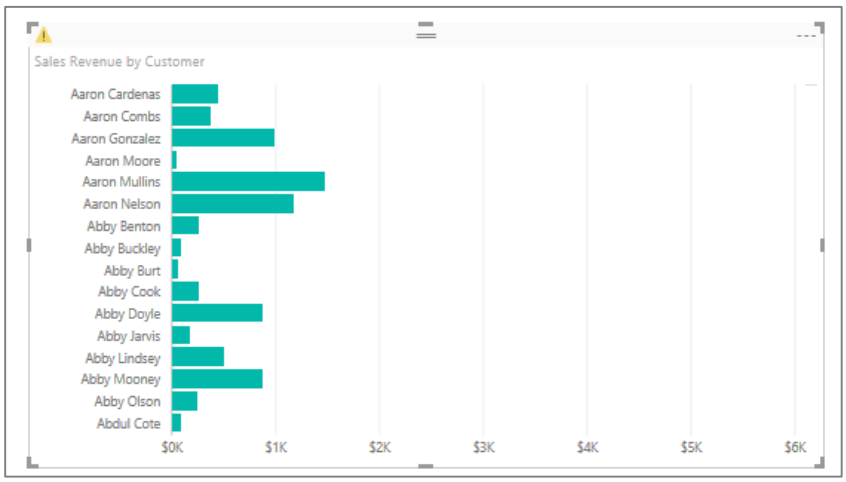



b) Drag and drop the **Customer** column from the **Customers** table into the **Axis** well.

c) Drag and drop the **Sales Revenue** measure from the **Sales** table into the **Values** well.

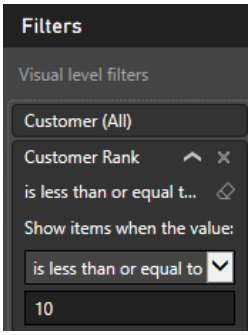



d) At this point, your visual should match the one shown in the following screenshot. Note that the visual currently shows an error because there are too many customers to show them all at once.



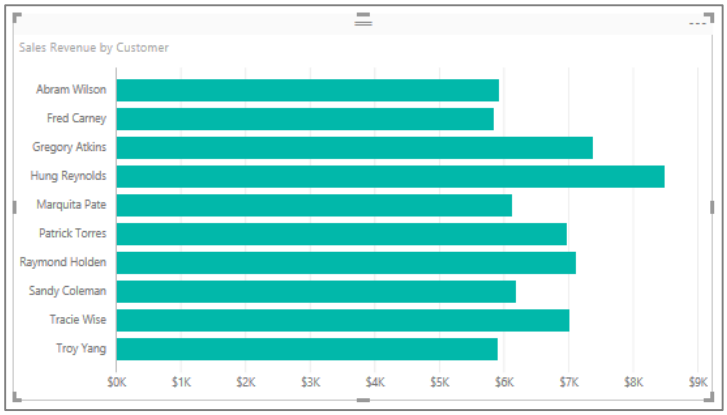

e) Drag and drop the **Customer Rank** measure from the **Customers** table into **Visual level filters** well of the **Filters** section.

f) Configure the **Customer Rank** filter to only display customers with a rank of 10 or lower as shown in the following screenshot and then click the **Apply Filter** link to apply the filter to the visual.

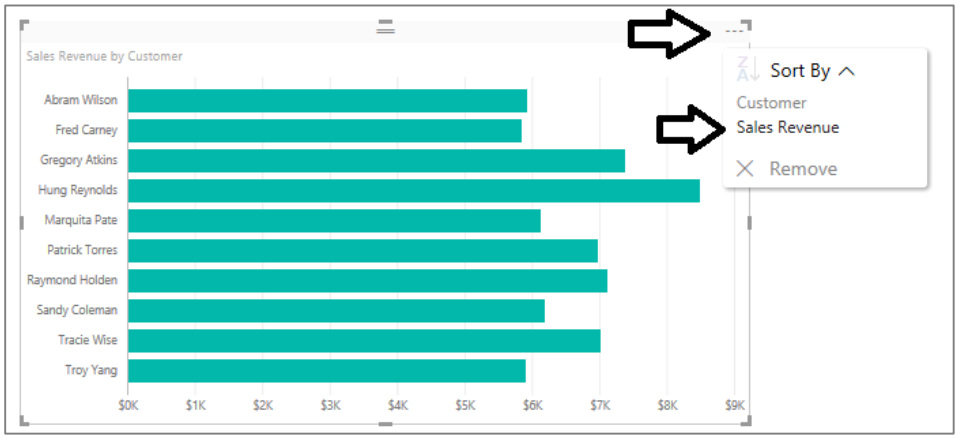



© Critical Path Training. 2015. All Rights Reserved

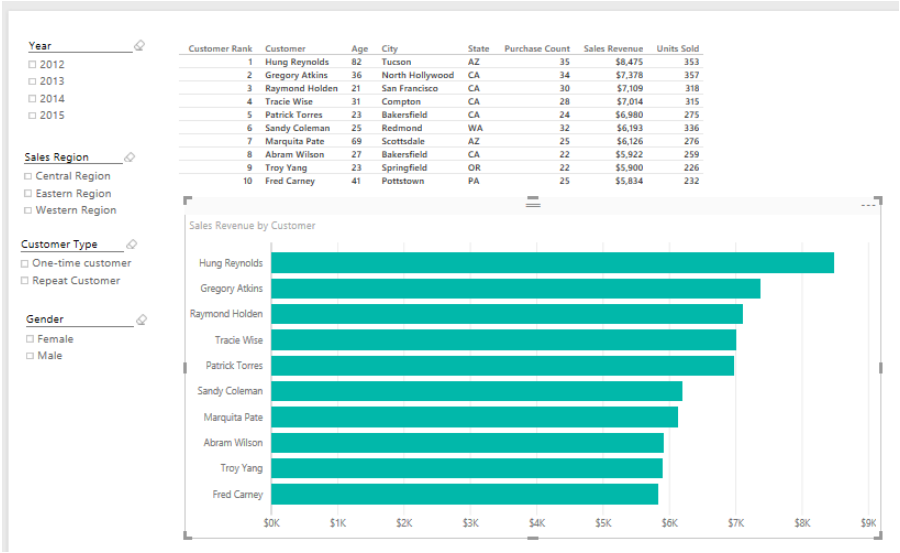
g) The visual should now match the following screenshot.




h) Click the ellipse (**...**) menu at the top, right corner of the visual and select Sort By ^ > Sales Revenue to sort the bars in the bar chart so that customers with the greatest revenue are at the top.

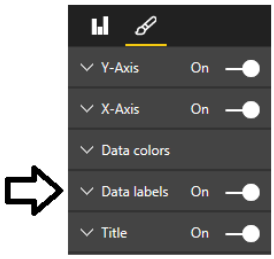



i) The customers in the visual should now be sorted with the greatest amounts of sales revenue at the top. Reposition the new visual to match the page layout shown in the following screenshot.

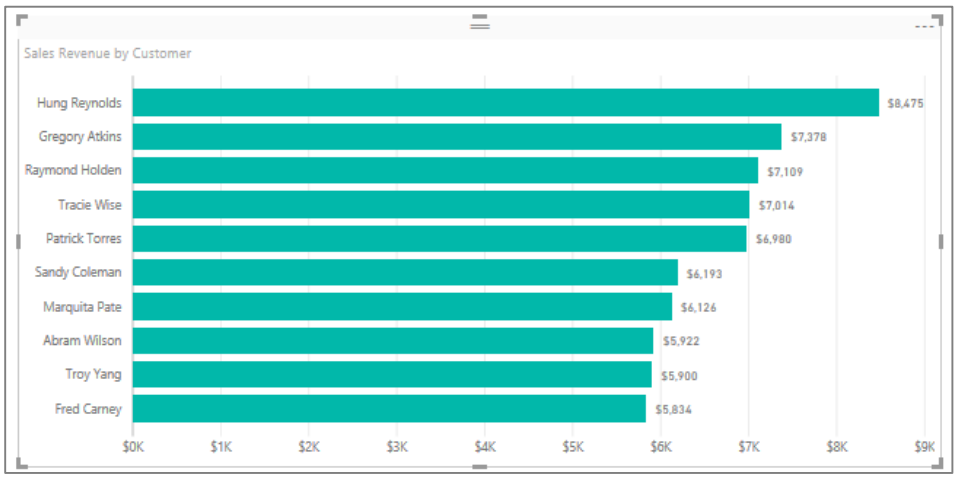



© Critical Path Training. 2015. All Rights Reserved

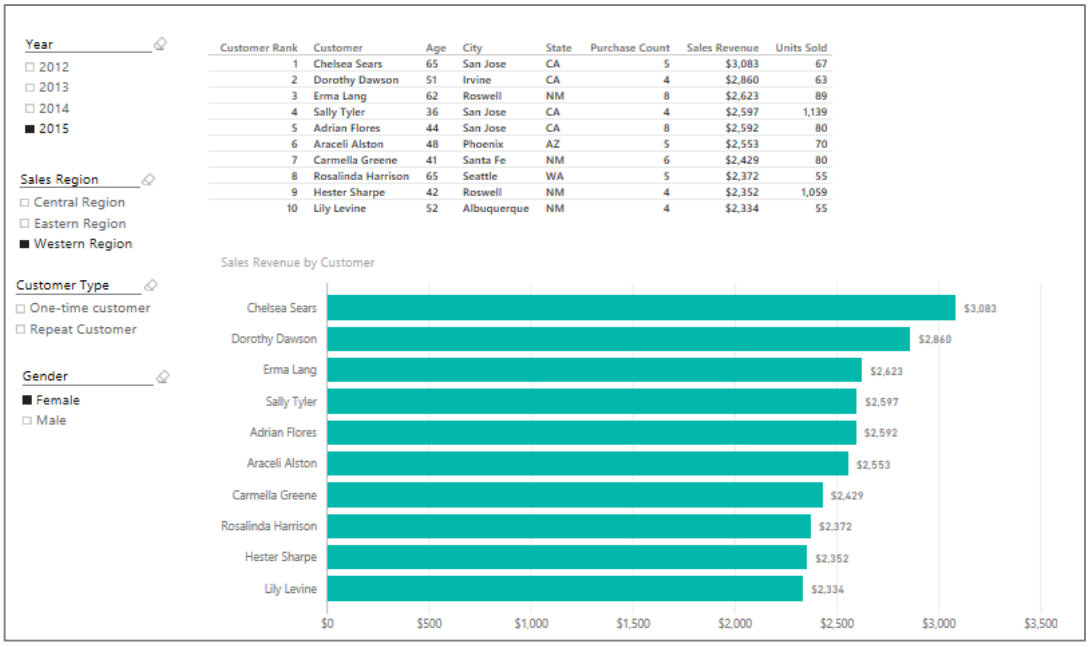
j) Add a small bit of formatting by selecting the bar chart and then changing the **Data labels** property setting from **Off** to **On**.




k) Now the visual should display an individual sales revenue total for each of the top 10 customers.




12. Test you work by using the slicer to select different combinations of years, sales region, customer type and gender. Both the table and the bar chart with the top 10 customers should update together and stay in sync as you change the filter selection.




| Customer Rank | Customer | Age | City | State | Purchase Count | Sales Revenue | Units Sold |
|---|---|---|---|---|---|---|---|
| 1 | Chelsea Sears | 65 | San Jose | CA | 5 | $3,083 | 67 |
| 2 | Dorothy Dawson | 51 | Irvine | CA | 4 | $2,860 | 63 |
| 3 | Erma Lang | 62 | Roswell | NM | 8 | $2,623 | 89 |
| 4 | Sally Tyler | 36 | San Jose | CA | 4 | $2,597 | 1,139 |
| 5 | Adrian Flores | 44 | San Jose | CA | 8 | $2,592 | 80 |
| 6 | Araceli Alston | 48 | Phoenix | AZ | 5 | $2,553 | 70 |
| 7 | Carmella Greene | 41 | Santa Fe | NM | 6 | $2,429 | 80 |
| 8 | Rosalinda Harrison | 65 | Seattle | WA | 5 | $2,372 | 55 |
| 9 | Hester Sharpe | 42 | Roswell | NM | 4 | $2,352 | 1,059 |
| 10 | Lily Levine | 52 | Albuquerque | NM | 4 | $2,334 | 55 |

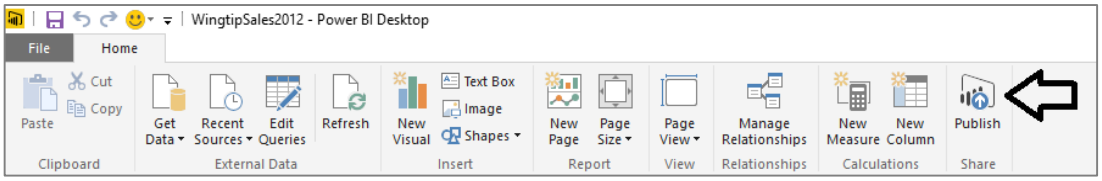
13. Save the work you have done by clicking the Save button in the upper left corner of the Power BI Desktop window.

© Critical Path Training. 2015. All Rights Reserved

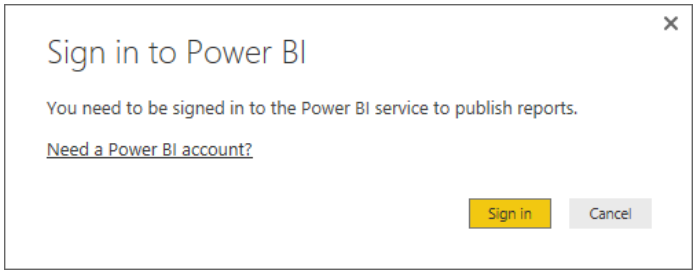
## Exercise 5: Publish Your Project and Its Reports to the Power BI Service

In this exercise you will complete your work by publishing the PBIX file to your personal workspace in the Power BI service.

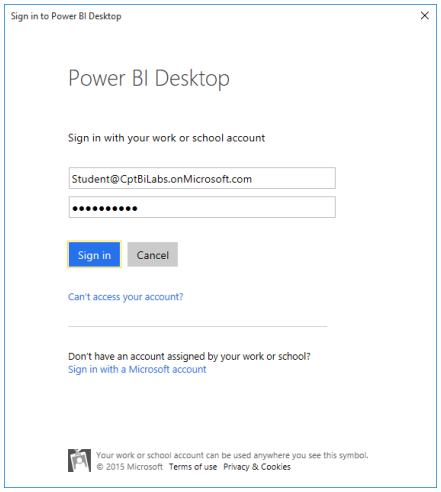
1. Make sure you still have the **WingtipSalesAnalytics.pbix** file open that you created in the previous exercise.
2. Click the **Publish** button on the far right-hand side of the ribbon..



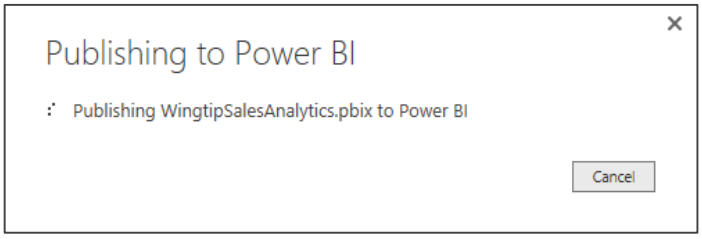

3. When promoted with the **Sign in to Power BI** dialog, click the **Sign In** button




4. Sign into the Power BI service using your primary Office 365 account to give Power BI Desktop the access to publish the PBIX file.

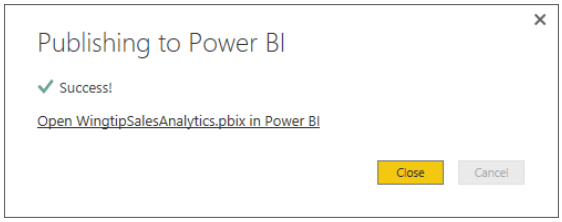



5. After you have signed in, Power BI Desktop will display the **Publishing to Power BI** dialog showing you that the publishing process is underway.

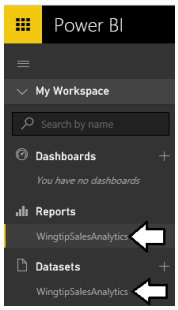



© Critical Path Training. 2015. All Rights Reserved

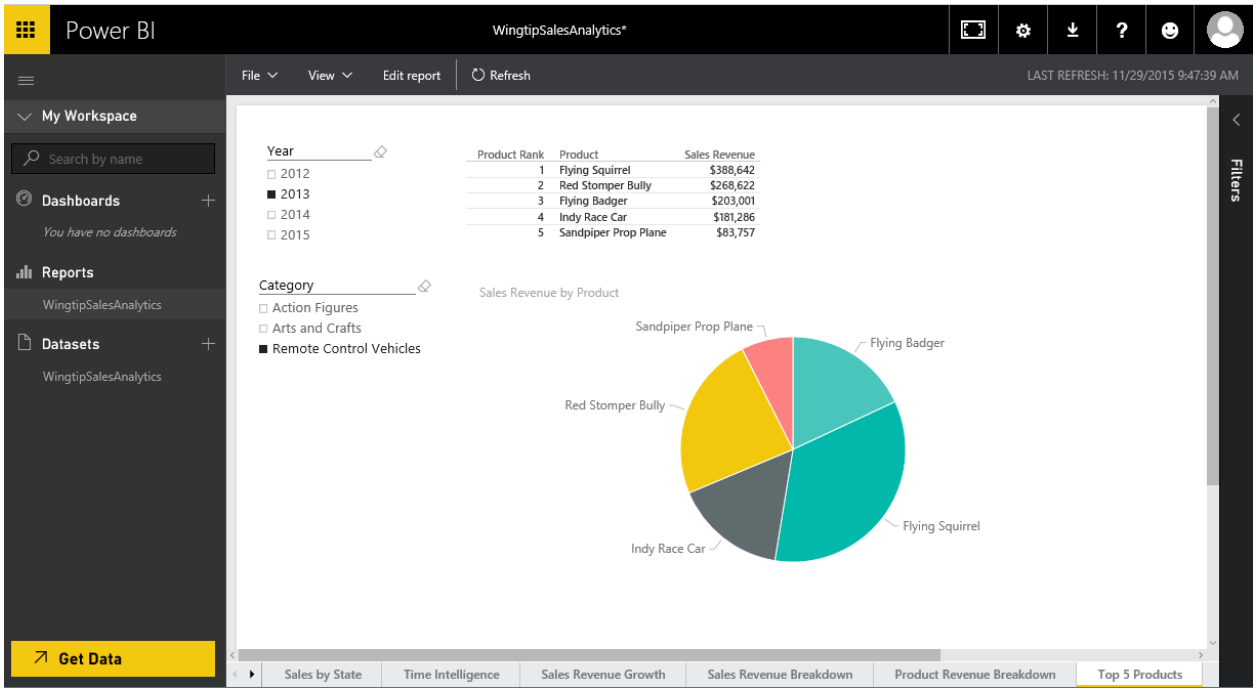
6. Once the publishing process has completed, the **Publishing to Power BI** dialog will display a success message and provide you with a link to **Open WingtipSalesAnalytics.pbix in Power BI**. Click on that link to navigate to the Power BI service using the browser.




7. Once you navigate to the Power BI service in the browser, you should be able to see that the publishing process added a dataset and a report named **WingtipSalesAnalysis** that appear in the left navigation along with any other datasets and reports that were already part of your personal workspace.




8. Inspect the various report pages that you created over the last few labs.




You have now successfully created and published your **WingtipsalesAnalysis.pbix** project using Power BI Desktop. In the next lab you will begin to consolidate the visuals you have created in these report pages into dashboards and you will also learn the various techniques you can use to deploy dashboards and share them with various groups of Office 365 users.

© Critical Path Training. 2015. All Rights Reserved

# Module 07: Creating Dashboards for Mobile Devices

## Module Description

One of Power BI's greatest strengths is the ease of which you can create dashboards and reports that are accessible to tablets and mobile devices such as iPhones, iPads, Android devices and Windows Phone. This module begins by explaining how the architecture of the Power BI services was designed to provide first class support for tablets and mobile devices. The module discusses which mobile platforms are currently supported as well as the steps one must go through to configure a tablet or a mobile device with the Power BI app. The module concludes with a discussion of dashboard design principles that are important to keep in mind when targeting the Power BI app on mobile devices.

## Module Agenda

- Creating Dashboards
- Sharing Dashboards
- Deploying Dashboards using Content Packs
- Accessing Dashboards using Mobile Devices

## Topics Covered

Dashboards in Power BI
Creating Dashboards
The Q&A Search Box
Pinning Search Result to Dashboard
Removing the Q&A Search Box
Modifying Tile Details
Sharing Dashboards
Sharing a Dashboard
Viewing the Shared With List
Testing the Dashboard as a Consumer
Drilling Down Into Reports
Creating a Content Pack
Viewing and Managing Content Packs
Creating a Group Workspace
Switching Over to the Group Workspace
Activating a Content Pack
Content Pack Activates Dashboard
Discovering and Installing the Power BI App
Launching the Power BI App
Viewing a Dashboard
Drilling Down on a Tile
Viewing a Report

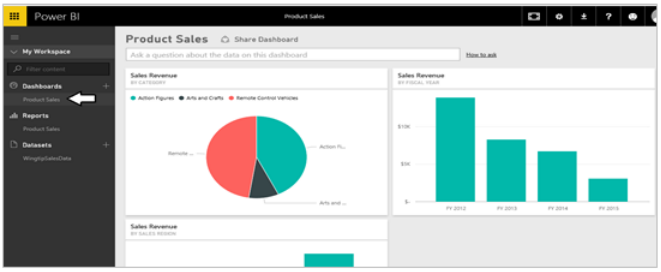
## Instructor Demos

- Creating a Dashboard
- Sharing a Dashboard
- Deploying a Dashboard using an Organizational Content Pack
- Viewing Dashboards and Reports using the Power BI App for Windows

© Critical Path Training. 2015. All Rights Reserved

# Dashboards in Power BI

- Dashboard is consolidated view of reports & datasets
  - Provides simpler, more intuitive entry point to reports and datasets
  - Simplicity and intuition key for users with tablets & mobile devices
- Dashboard represents unit of deployment and reuse
  - Dashboards can by deployed via dashboard sharing
  - Dashboards can by deployed using content packs
  - Dashboard represents security boundary to reports and datasets



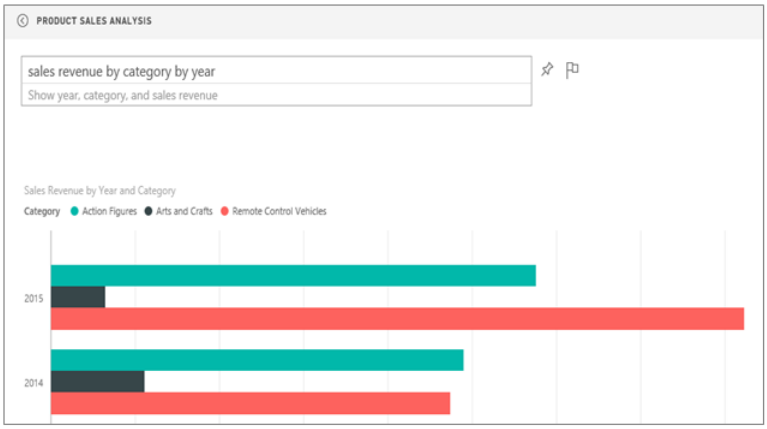
# Creating Dashboards

- Dashboard is a collection of tiles
  - Dashboard tiles can be created by pinning a visual from a report
  - Dashboard tiles can be created from Q&A search result
  - Each tile is snapshot of information from an underlying dataset
- Dashboard can be created two different ways
  - Click on (**+**) button in Dashboards section of navigation pane
  - Create new dashboard when pinning visual from report

© Critical Path Training. 2015. All Rights Reserved

# The Q&A Search Box

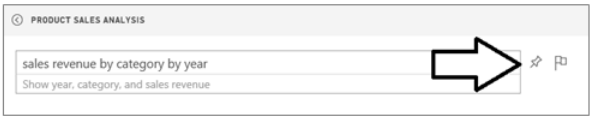
- Used to execute natural language queries
  - Queries are executed against underlying dataset in workspace
  - Query results display using some type of Power BI visual
  - Power BI service selects visualization type based on result type

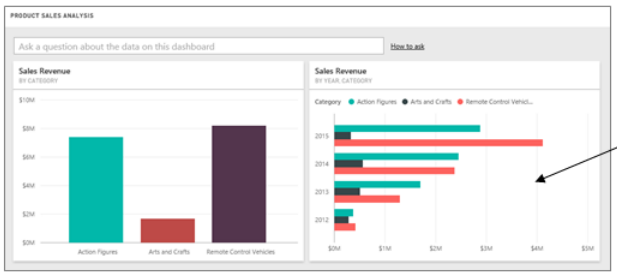



© Critical Path Training. 2015. All Rights Reserved

# Pinning Search Result to Dashboard

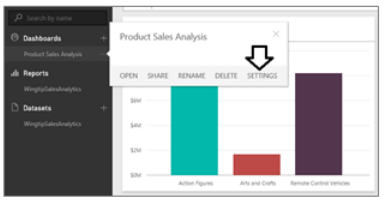
- Q&A search results can be pinned to dashboard
  - Allows users to easily save visuals from query results

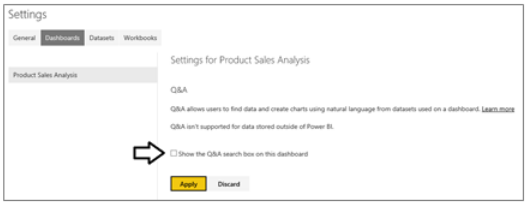


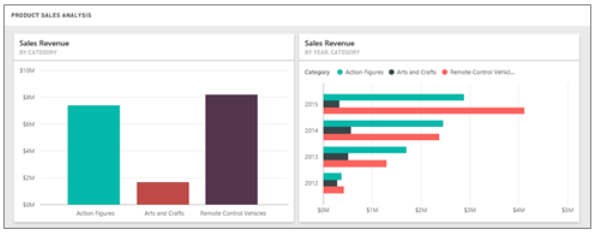




This tile was created from a visual on a report

# Removing the Q&A Search Box



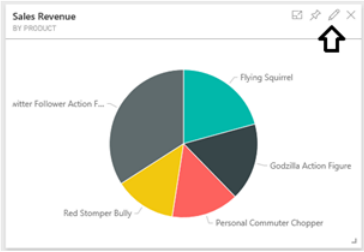



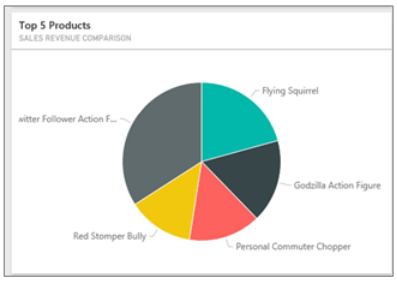



# Modifying Tile Details

- You can customize Title and Subtitle of tile
  - Default title and subtitle doesn't always provide enough clarity

  - Click on Edit title edit icon button





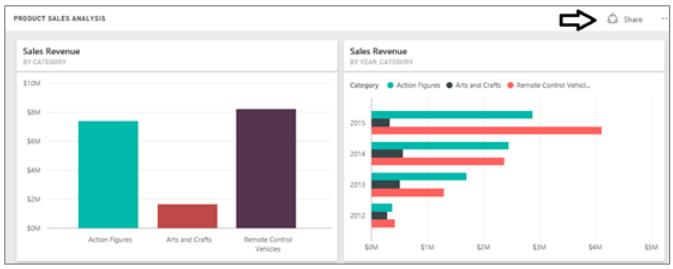

© Critical Path Training. 2015. All Rights Reserved

# Sharing Dashboards

- There are two types of Power BI workspaces
  - Personal workspaces
  - Group Workspaces
- To deploy reports and dashboards from a personal workspace
  - Create one or more reports in your personal workspace
  - Create a dashboard and pin report tiles to it
  - Share dashboard with other users
  - Dashboard sharing involves accepting invitation
  - Dashboard owner can see who has accepted invitation
- Deploying reports and dashboards from group workspace
  - Group workspace created using Unified Group in Office 365 tenant
  - Allows team of authors to work on a single BI project
  - Users can be added with authoring permission or read-only permissions

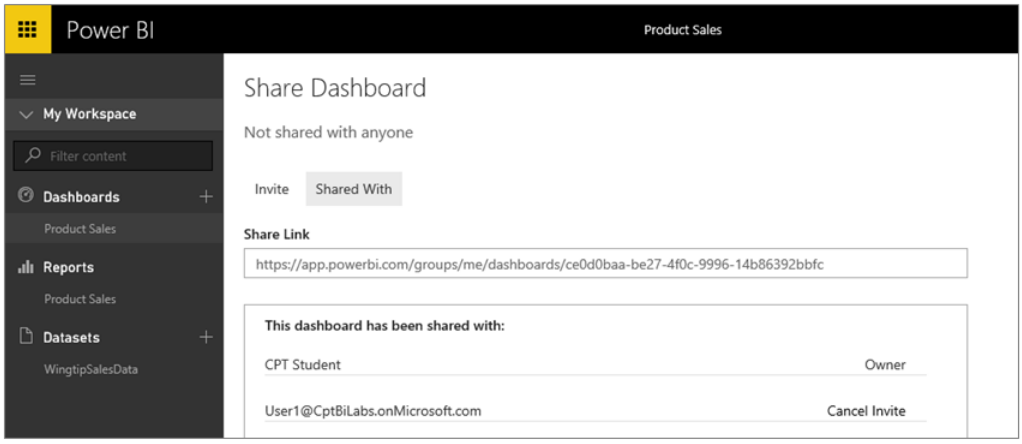
© Critical Path Training. 2015. All Rights Reserved

# Sharing a Dashboard



© Critical Path Training. 2015. All Rights Reserved

# Viewing the Shared With List

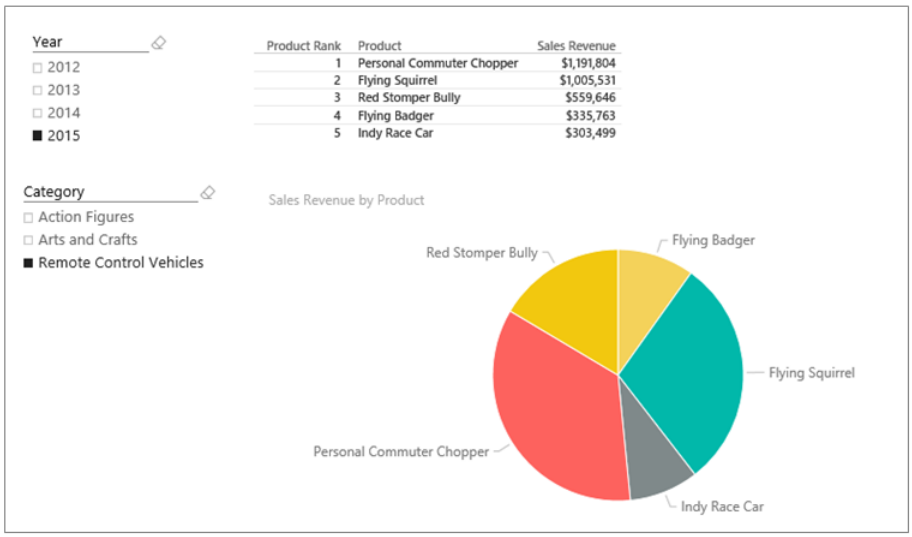



© Critical Path Training. 2015. All Rights Reserved

# Testing the Dashboard as a Consumer

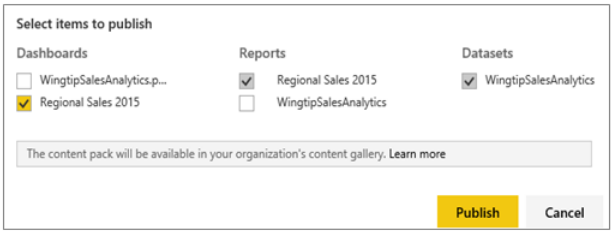
© Critical Path Training. 2015. All Rights Reserved

# Drilling Down Into Reports




© Critical Path Training. 2015. All Rights Reserved

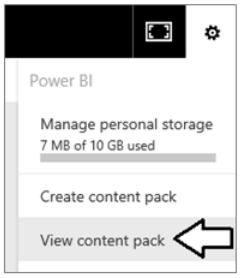
# Creating a Content Pack




© Critical Path Training. 2015. All Rights Reserved

# Viewing and Managing Content Packs

- You can view existing content packs



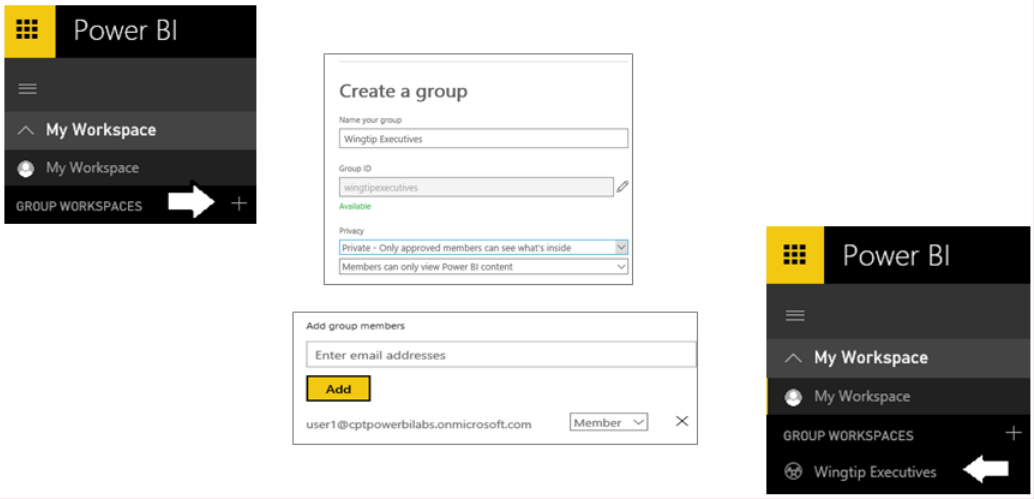

- You can delete a content pack or edit its properties

| Name | Published To | Date published | Actions |
|---|---|---|---|
| Regional Sales 2015 | My Organization | Dec 02, 2015 | Edit \| Delete |

© Critical Path Training. 2015. All Rights Reserved

# Creating a Group Workspace

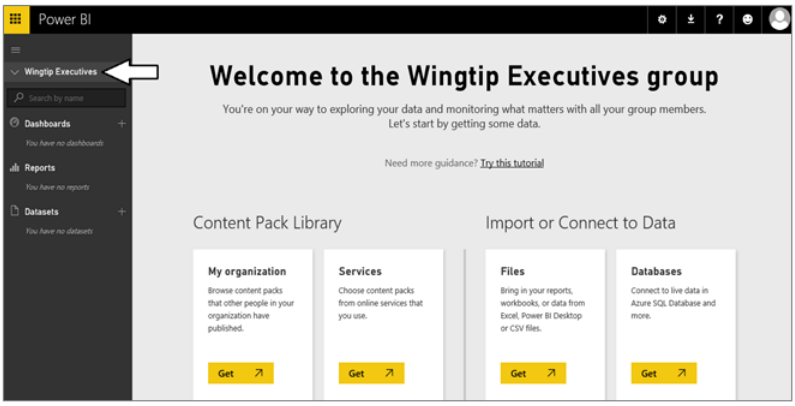
- Group Workspace can be created in Power BI service
  - This action created Office 365 unified group behind the scenes
  - Allow for shared authoring scenarios




© Critical Path Training. 2015. All Rights Reserved

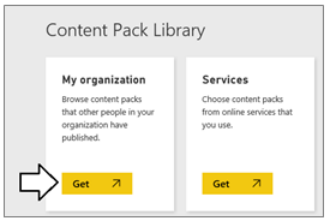
# Switching Over to the Group Workspace

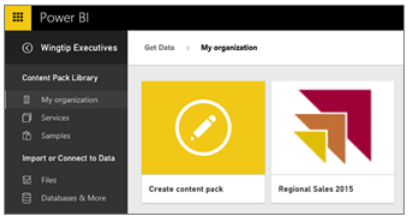
- Group workspace has its own collections
  - Dashboards
  - Reports
  - Datasets

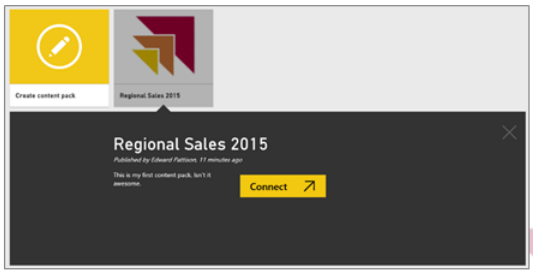



© Critical Path Training. 2015. All Rights Reserved

# Activating a Content Pack

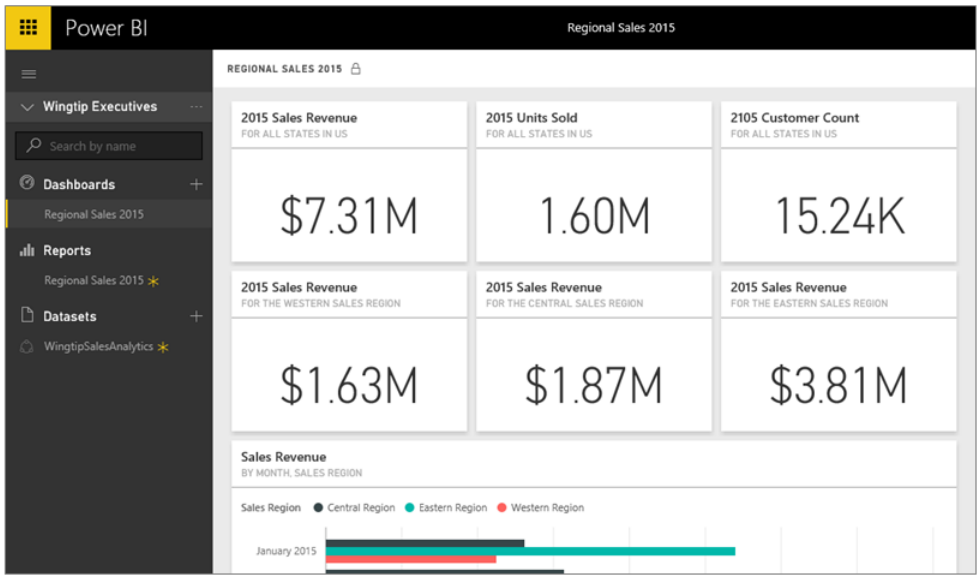


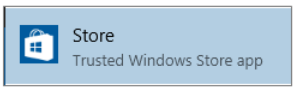

© Critical Path Training. 2015. All Rights Reserved

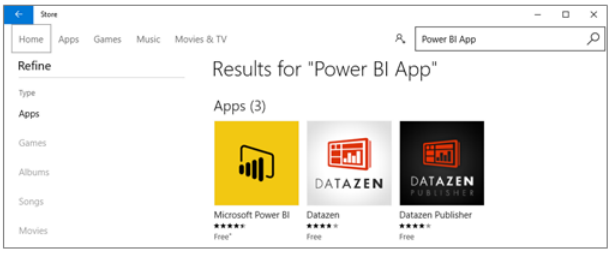
# Content Pack Activates Dashboard

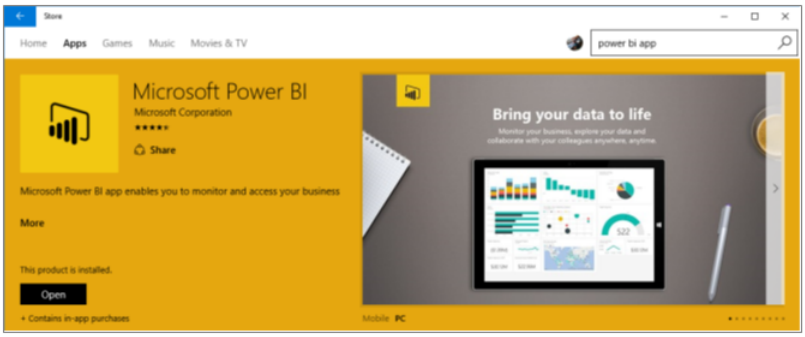



© Critical Path Training. 2015. All Rights Reserved

# Discovering and Installing the Power BI App

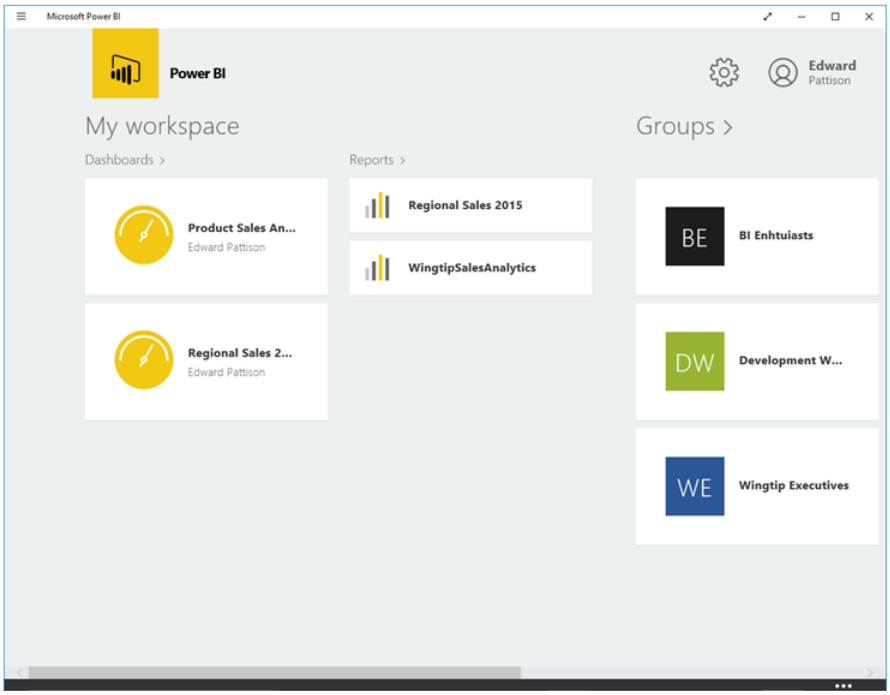


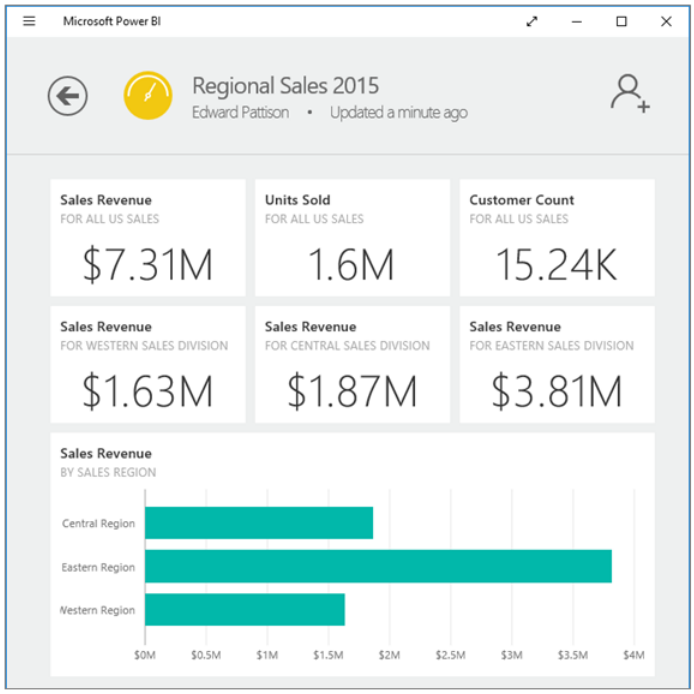

© Critical Path Training. 2015. All Rights Reserved

# Launching the Power BI App



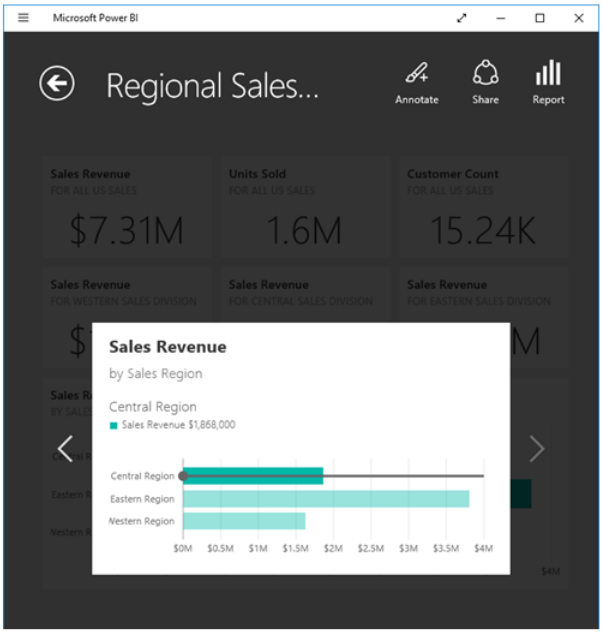

© Critical Path Training. 2015. All Rights Reserved

# Viewing a Dashboard

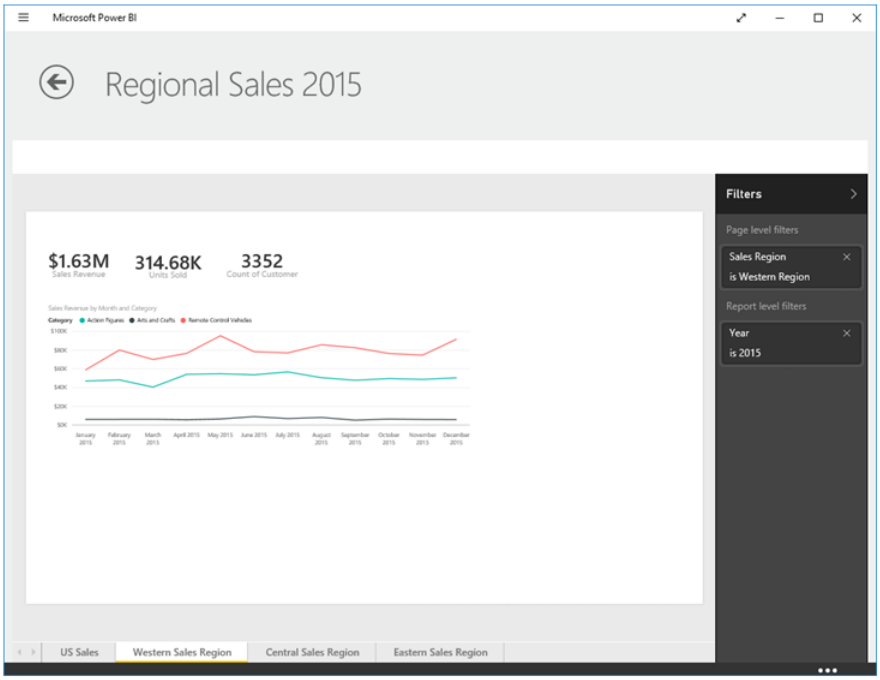



© Critical Path Training. 2015. All Rights Reserved

# Drilling Down on a Tile



# Viewing a Report




© Critical Path Training. 2015. All Rights Reserved

# Module 07 Lab: Creating and Deploying Dashboards in Power BI

**Lab Time**: 60 minutes

**Lab Folder**: C:\Student\Modules\Dashboards\Lab

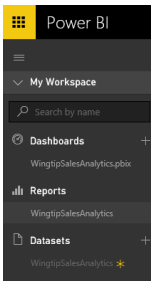
**Lab Overview**: In this lab you will continue to work with the dataset and the report that you have been designing in the Power BI Desktop project named **WingtipSalesAnalytics.pbix** in earlier labs. You will focus on the techniques required to create and deploy dashboards to an audience of business users.

**Important:** This lab assumes you have completed the previous lab titled **Designing and Publishing Reports with Power BI Desktop** in which you completed your data modeling work in the **WingtipSalesAnalytics.pbix** project and then published its report and dataset to the Power BI Service. If you would like to start this lab without completing the previous lab, copy the lab solution at **C:\Student\Modules\TimeIntelligence\LabSolution\WingtipSalesAnalytics.pbix** into the folder at **C:\Student\Projects** using the Windows Explorer. After that, you must complete **Exercise 5** from the previous lab titled **Publish Your Project and Its Reports to the Power BI Service** in order to reach the starting point for this lab.

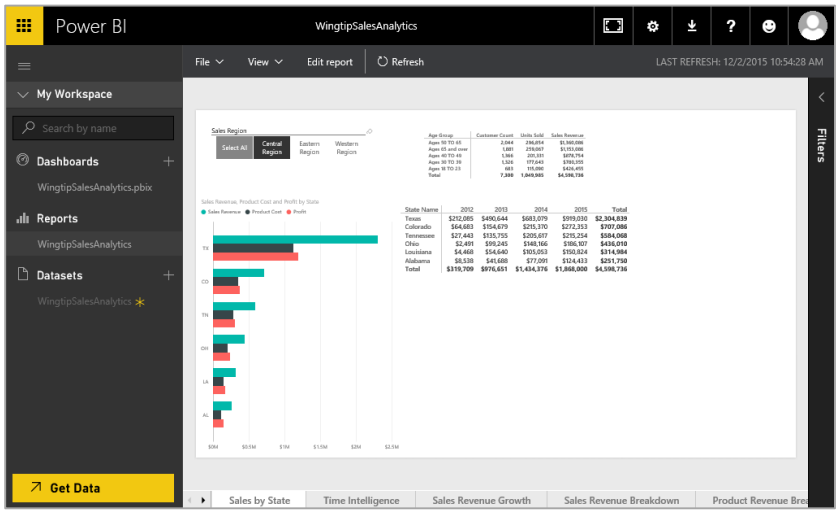
## Exercise 1: Create the Product Sales Analysis Dashboard

In this exercise you will create a new dashboard using the dataset and report you created in early labs using Power BI Desktop.

1. Log in to the Power BI service.
   a) Open a browser and navigate to the Power BI service at https://app.powerbi.com.
   b) Login with your primary Office 365 account.
   c) Make sure you are running in the context of your personal workspace.
   d) You should be able to see the dataset and report named **WingtipSalesAnalysis** that you published at the end of the last lab using Power BI Desktop. There is also a minimal dashboard named **WingtipSalesAnalysis.pbix** that was automatically created during the PBIX publishing process for the Power BI Desktop project file named **WingtipSalesAnalysis.pbix**.




2. Click on the report named **WingtipSalesAnalysis** in the **Reports** section. Examine the pages in the report and verify that these are the same report pages that you have been designing over the last several labs.

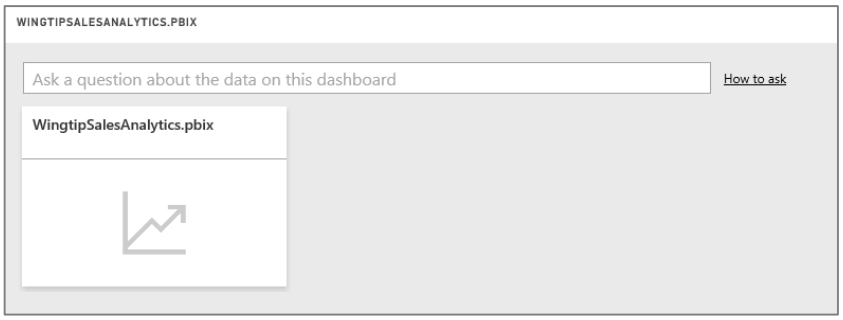



© Critical Path Training. 2015. All Rights Reserved

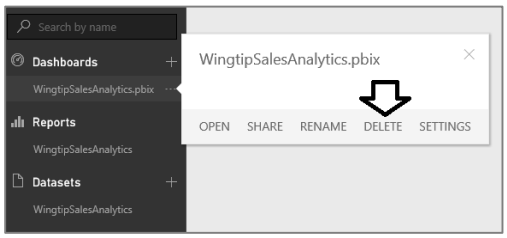
3. Click on the dataset named **WingtipSalesAnalysis** in the **Datasets** section. The Power BI service responds by displaying a new report that allows you to begin adding visuals.

When you navigate to a dataset in the Power BI service, it provides a different experience compared to when in Power BI Desktop. That's because Power BI Desktop allows you to customize and extend a dataset while the browser-based experience of the Power BI service only allows you to consume datasets but not to modify them. Given the fact that a dataset is a read-only object, the Power BI service responds to user's request to navigate to a dataset by opening a new report and showing the **Fields** list for that dataset.

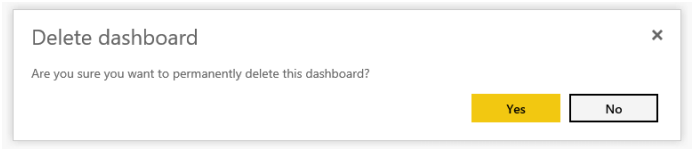
4. Click on the dashboard named **WingtipSalesAnalysis.pbix** in the **Dashboards** section. You should be able to verify that the dashboard isn't very interesting as it only contains the Q&A search box and a tile with the name of the PBIX project file.




5. Delete the WingtipSalesAnalysis.pbix dashboard.
   a) Click the ellipse (**…**) button to the right of the dashboard to display the dashboard fly out menu and then selecting the **DELETE** command.

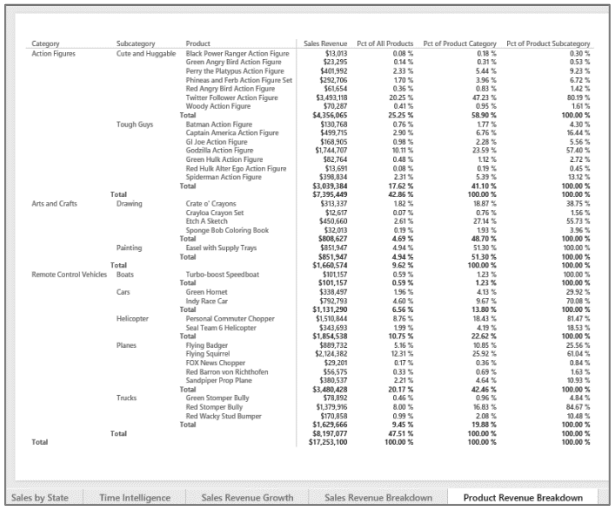



   b) Confirm you want to delete the dashboard by clicking the **Yes** button.




   c) You should be able to verify that the dashboard has been deleted.
6. Navigate to the **Product Revenue Breakdown** report page and add a new visual.
   a) Click on the report named **WingtipSalesAnalysis** in the **Reports** section.
   b) Navigate to the Product Revenue Breakdown page.

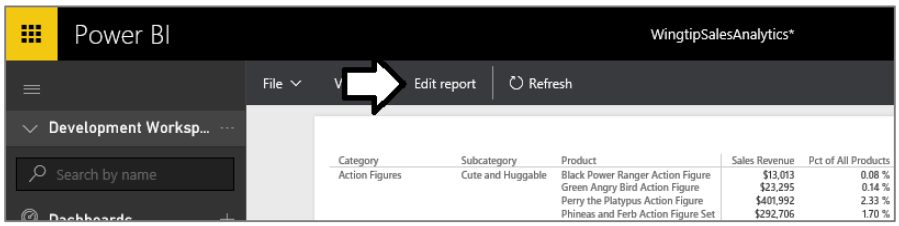
© Critical Path Training. 2015. All Rights Reserved



| Category | Subcategory | Product | Sales Revenue | Pct of All Products | Pct of Product Category | Pct of Product Subcategory |
|---|---|---|---|---|---|---|
| Action Figures | Cute and Huggable | Black Power Ranger Action Figure | $13,013 | 0.08 % | 0.18 % | 0.30 % |
| | | Green Angry Bird Action Figure | $23,295 | 0.14 % | 0.31 % | 0.53 % |
| | | Perry the Platypus Action Figure | $401,992 | 2.33 % | 5.44 % | 9.23 % |
| | | Phineas and Ferb Action Figure Set | $292,706 | 1.70 % | 3.96 % | 6.72 % |
| | | Red Angry Bird Action Figure | $61,654 | 0.36 % | 0.83 % | 1.42 % |
| | | Twitter Follower Action Figure | $3,493,118 | 20.25 % | 47.23 % | 80.19 % |
| | | Woody Action Figure | $70,287 | 0.41 % | 0.95 % | 1.61 % |
| | | **Total** | **$4,356,065** | **25.25 %** | **58.90 %** | **100.00 %** |
| | Tough Guys | Batman Action Figure | $130,768 | 0.76 % | 1.77 % | 4.30 % |
| | | Captain America Action Figure | $499,715 | 2.90 % | 6.76 % | 16.44 % |
| | | GI Joe Action Figure | $168,905 | 0.98 % | 2.28 % | 5.56 % |
| | | Godzilla Action Figure | $1,744,707 | 10.11 % | 23.59 % | 57.40 % |
| | | Green Hulk Action Figure | $82,764 | 0.48 % | 1.12 % | 2.72 % |
| | | Red Hulk Alter Ego Action Figure | $13,691 | 0.08 % | 0.19 % | 0.45 % |
| | | Spiderman Action Figure | $398,834 | 2.31 % | 5.39 % | 13.12 % |
| | | **Total** | **$3,039,384** | **17.62 %** | **41.10 %** | **100.00 %** |
| | **Total** | | **$7,395,449** | **42.86 %** | **100.00 %** | **100.00 %** |
| Arts and Crafts | Drawing | Crate o' Crayons | $313,337 | 1.82 % | 18.87 % | 38.75 % |
| | | Crayloa Crayon Set | $12,617 | 0.07 % | 0.76 % | 1.56 % |
| | | Etch A Sketch | $450,660 | 2.61 % | 27.14 % | 55.73 % |
| | | Sponge Bob Coloring Book | $32,013 | 0.19 % | 1.93 % | 3.96 % |
| | | **Total** | **$808,627** | **4.69 %** | **48.70 %** | **100.00 %** |
| | Painting | Easel with Supply Trays | $851,947 | 4.94 % | 51.30 % | 100.00 % |
| | | **Total** | **$851,947** | **4.94 %** | **51.30 %** | **100.00 %** |
| | **Total** | | **$1,660,574** | **9.62 %** | **100.00 %** | **100.00 %** |
| Remote Control Vehicles | Boats | Turbo-boost Speedboat | $101,157 | 0.59 % | 1.23 % | 100.00 % |
| | | **Total** | **$101,157** | **0.59 %** | **1.23 %** | **100.00 %** |
| | Cars | Green Hornet | $338,497 | 1.96 % | 4.13 % | 29.92 % |
| | | Indy Race Car | $792,793 | 4.60 % | 9.67 % | 70.08 % |
| | | **Total** | **$1,131,290** | **6.56 %** | **13.80 %** | **100.00 %** |
| | Helicopter | Personal Commuter Chopper | $1,510,844 | 8.76 % | 18.43 % | 81.47 % |
| | | Seal Team 6 Helicopter | $343,693 | 1.99 % | 4.19 % | 18.53 % |
| | | **Total** | **$1,854,538** | **10.75 %** | **22.62 %** | **100.00 %** |
| | Planes | Flying Badger | $889,732 | 5.16 % | 10.85 % | 25.56 % |
| | | Flying Squirrel | $2,124,382 | 12.31 % | 25.92 % | 61.04 % |
| | | FOX News Chopper | $29,201 | 0.17 % | 0.36 % | 0.84 % |
| | | Red Barron von Richthofen | $56,575 | 0.33 % | 0.69 % | 1.63 % |
| | | Sandpiper Prop Plane | $380,537 | 2.21 % | 4.64 % | 10.93 % |
| | | **Total** | **$3,480,428** | **20.17 %** | **42.46 %** | **100.00 %** |
| | Trucks | Green Stomper Bully | $78,892 | 0.46 % | 0.96 % | 4.84 % |
| | | Red Stomper Bully | $1,379,916 | 8.00 % | 16.83 % | 84.67 % |
| | | Red Wacky Stud Bumper | $170,858 | 0.99 % | 2.08 % | 10.48 % |
| | | **Total** | **$1,629,666** | **9.45 %** | **19.88 %** | **100.00 %** |
| | **Total** | | **$8,197,077** | **47.51 %** | **100.00 %** | **100.00 %** |
| **Total** | | | **$17,253,100** | **100.00 %** | **100.00 %** | **100.00 %** |

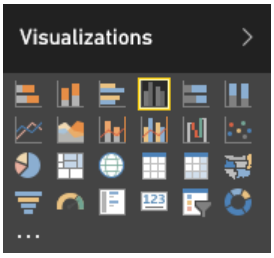


c) The report is initially displayed in **Reading view**. Click on the **Edit report** button to enter edit view.





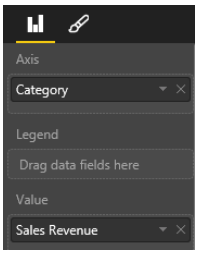
d) Add a new visual to the page based on the **Clustered column chart** visualization type.





e) Drag the **Categories** column from the **Products** table and drop it into the **Axis** well.

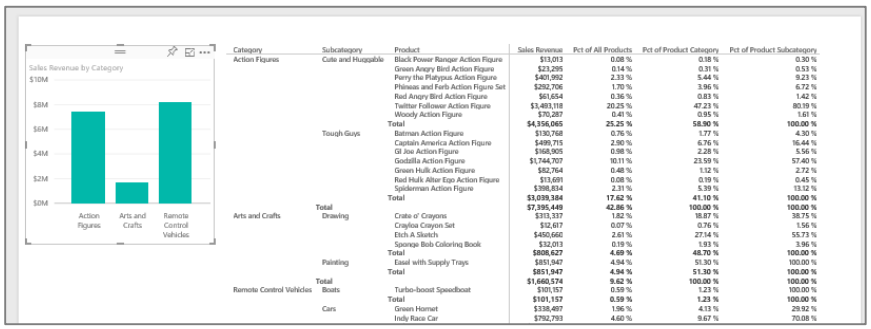
f) Drag the **Sales Revenue** measure from the **Sales** table and drop it into the **Value** well.





g) Rearrange the two visuals on the page to match the visuals shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

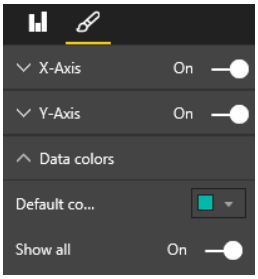



| Category | Subcategory | Product | Sales Revenue | Pct of All Products | Pct of Product Category | Pct of Product Subcategory |
|---|---|---|---|---|---|---|
| Action Figures | Cute and Huggable | Black Power Ranger Action Figure | $13,013 | 0.08 % | 0.18 % | 0.30 % |
| | | Green Angry Bird Action Figure | $23,295 | 0.14 % | 0.31 % | 0.53 % |
| | | Perry the Platypus Action Figure | $401,992 | 2.33 % | 5.44 % | 9.23 % |
| | | Phineas and Ferb Action Figure Set | $292,706 | 1.70 % | 3.96 % | 6.72 % |
| | | Red Angry Bird Action Figure | $61,654 | 0.36 % | 0.83 % | 1.42 % |
| | | Twitter Follower Action Figure | $3,493,118 | 20.25 % | 47.23 % | 80.19 % |
| | | Woody Action Figure | $70,287 | 0.41 % | 0.95 % | 1.61 % |
| | | **Total** | **$4,356,065** | **25.25 %** | **58.90 %** | **100.00 %** |
| | Tough Guys | Batman Action Figure | $130,768 | 0.76 % | 1.77 % | 4.30 % |
| | | Captain America Action Figure | $499,715 | 2.90 % | 6.76 % | 16.44 % |
| | | GI Joe Action Figure | $168,905 | 0.98 % | 2.28 % | 5.56 % |
| | | Godzilla Action Figure | $1,744,707 | 10.11 % | 23.59 % | 57.40 % |
| | | Green Hulk Action Figure | $82,764 | 0.48 % | 1.12 % | 2.72 % |
| | | Red Hulk Alter Ego Action Figure | $13,691 | 0.08 % | 0.19 % | 0.45 % |
| | | Spiderman Action Figure | $398,834 | 2.31 % | 5.39 % | 13.12 % |
| | | **Total** | **$3,039,384** | **17.62 %** | **41.10 %** | **100.00 %** |
| | **Total** | | **$7,395,449** | **42.86 %** | **100.00 %** | **100.00 %** |
| Arts and Crafts | Drawing | Crate o' Crayons | $313,337 | 1.82 % | 18.87 % | 38.75 % |
| | | Crayloa Crayon Set | $12,617 | 0.07 % | 0.76 % | 1.56 % |
| | | Etch A Sketch | $450,660 | 2.61 % | 27.14 % | 55.73 % |
| | | Sponge Bob Coloring Book | $32,013 | 0.19 % | 1.93 % | 3.96 % |
| | | **Total** | **$808,627** | **4.69 %** | **48.70 %** | **100.00 %** |
| | Painting | Easel with Supply Trays | $851,947 | 4.94 % | 51.30 % | 100.00 % |
| | | **Total** | **$851,947** | **4.94 %** | **51.30 %** | **100.00 %** |
| | **Total** | | **$1,660,574** | **9.62 %** | **100.00 %** | **100.00 %** |
| Remote Control Vehicles | Boats | Turbo-boost Speedboat | $101,157 | 0.59 % | 1.23 % | 100.00 % |
| | | **Total** | **$101,157** | **0.59 %** | **1.23 %** | **100.00 %** |
| | Cars | Green Hornet | $338,497 | 1.96 % | 4.13 % | 29.92 % |
| | | Indy Race Car | $792,793 | 4.60 % | 9.67 % | 70.08 % |

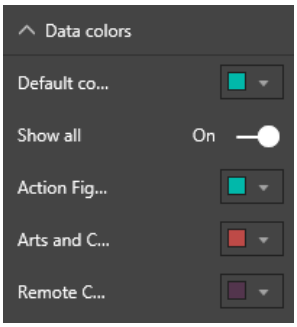
The new column chart visual contains three columns that are all the same color by default. In the next step, you will make the visual a little more fancy by configuring each a column with a different color.

h) With the column chart visual selected, click the brush icon in the **Visualization** pane to display the visual's **Format** properties.

i) Locate the **Show All** property inside the **Data colors** section and set its value to **On**.



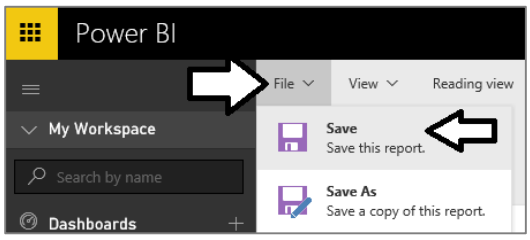

j) Modify the colors so that each category is a different color.



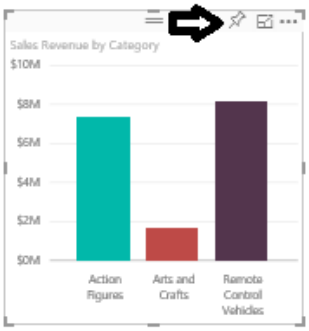

k) The visual should now display in a multicolored format.

l) Save the report by dropping down the **File** menu and clicking **Save**.

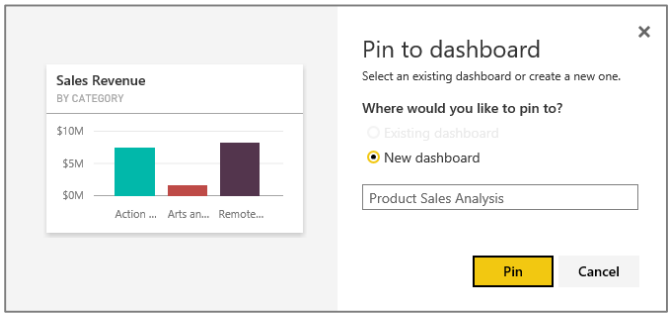
© Critical Path Training. 2015. All Rights Reserved



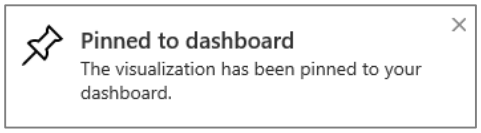

7. Create the **Product Sales Analysis** dashboard.
   a) At this point, you be looking at the **Product Revenue Breakdown** report page in edit view.
   b) If you select the column chart visual, you will notice several icon buttons in the upper-right corner. Click on the thumbtack icon button to pin the visual to display the **Pin to dashboard** dialog.




   c) The **Pin to dashboard dialog** prompts you enter a name for a new dashboard name. Enter a value of **Product Sales Analysis** as the new dashboard name and then click the **Pin** button to create the new dashboard and pin the visual to it.




   d) At this point, the Power BI service displays a notification indicating success.

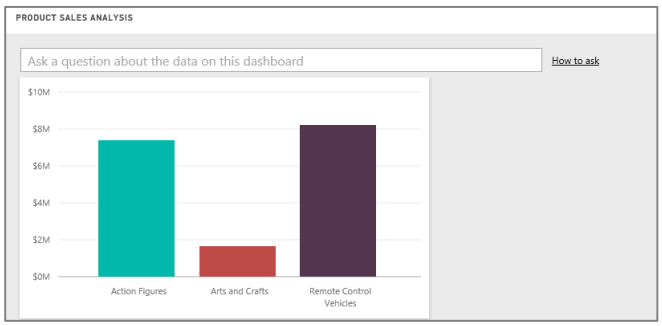



Your day's always a bit brighter when someone gives you a little positive reinforcement!

   e) At this point, you should be able to see the new **Product Sales Analysis** dashboard in the **Dashboards** section

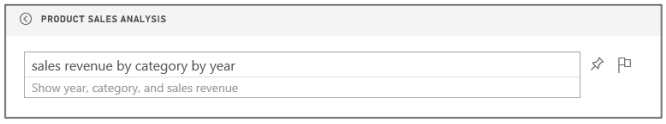
© Critical Path Training. 2015. All Rights Reserved

8. Examine the **Product Sales Analysis** dashboard.
   a) Click on the link for the **Product Sales Analysis** dashboard in the **Dashboards** section.
   b) Examine the dashboard. It currently contains the Q&A search box and a tile for the column chart visual.

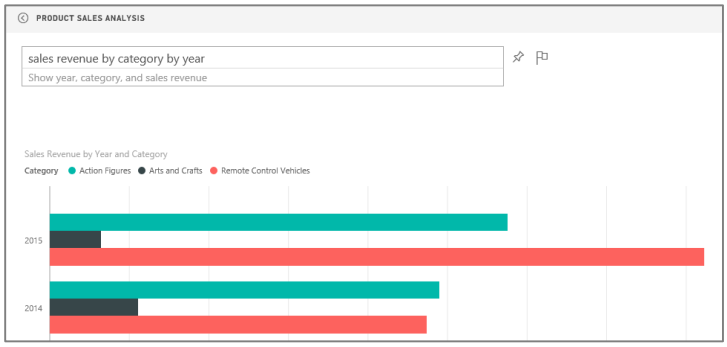



Note that when a user clicks on a dashboard tile created from a report visual, the user is then redirected to underlying report page where the original visual was created.

9. Add a new dashboard tile using the Q&A search box.
   a) Navigate to the **Product Sales Analysis** dashboard if you are not already there.
   b) Type ***sales revenue by category by year*** into the Q&A search box and press the ENTER key to run the query.

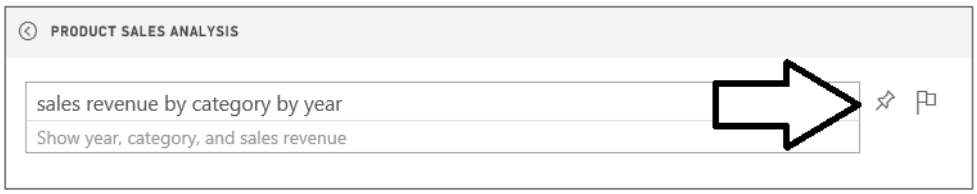



   c) The Power BI service should respond by running the query and then creating a bar chart visual displaying the query results.

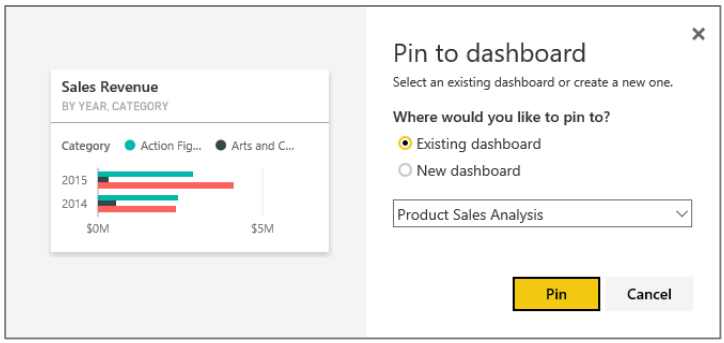



   d) Click on the thumbtack icon button to display the **Pin to dashboard** dialog.

© Critical Path Training. 2015. All Rights Reserved

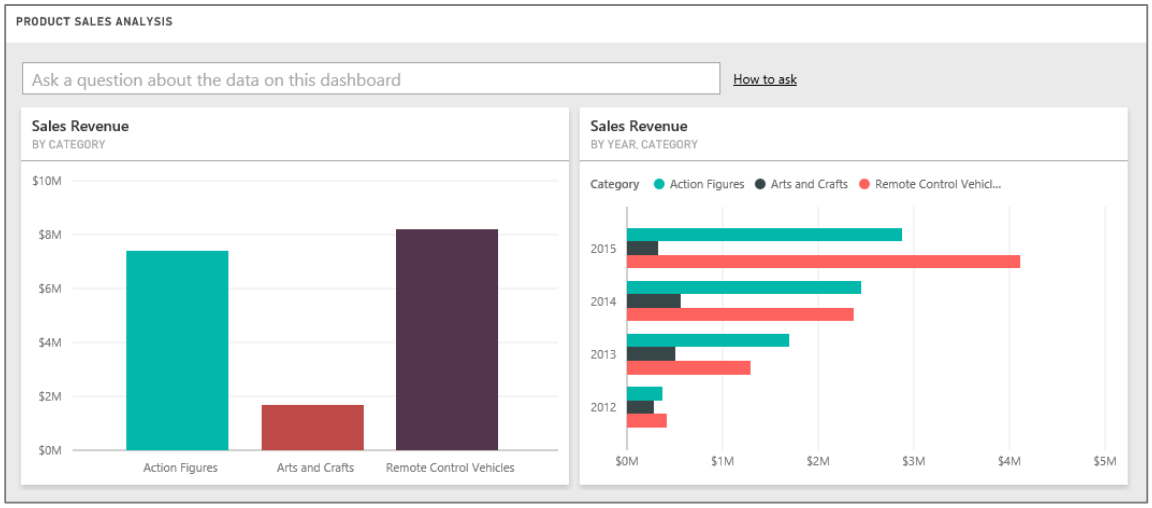



e) In the **Pin to dashboard** dialog, ensure that you are pinning the visual to the **Product Sales Analysis** dashboard and click the **Pin** button.



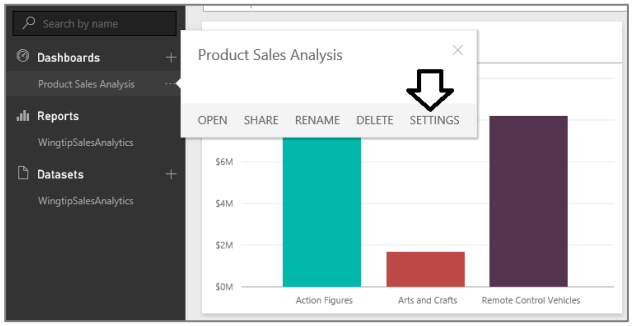

f) Click on the link to navigate back to the **Product Sales Analysis** dashboard.

g) You should now see the new dashboard tile that's been added by pinning the visual created using the Q&A search box.




10. Remove the Q&A search box from the dashboard.
    a) Use the fly out menu of the for the **Product Sales Analysis** dashboard and click the **SETTINGS** menu.

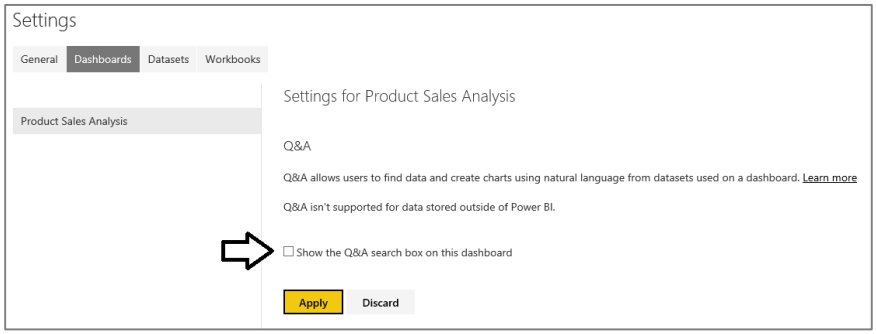
© Critical Path Training. 2015. All Rights Reserved



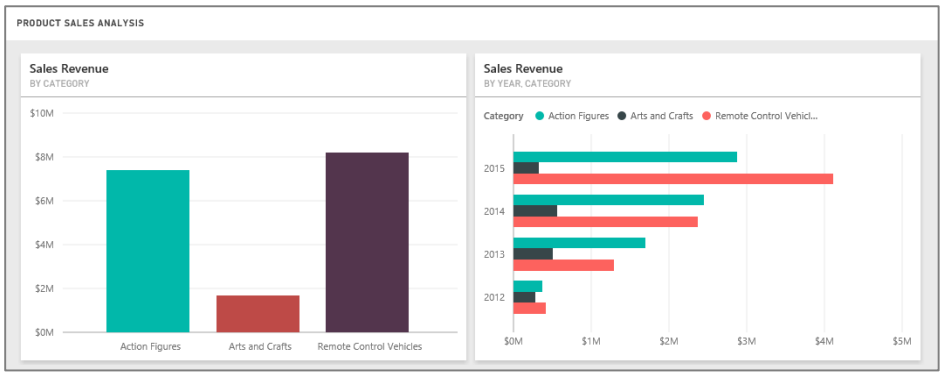

b) On the **Settings** page for the dashboard, navigate the **Dashboards** tab.

c) Uncheck the checkbox with the caption of **Show the Q&A search box**.

d) Click the **Apply** button.




e) You should be able to verify that the dashboard no longer displays the Q&A search box.



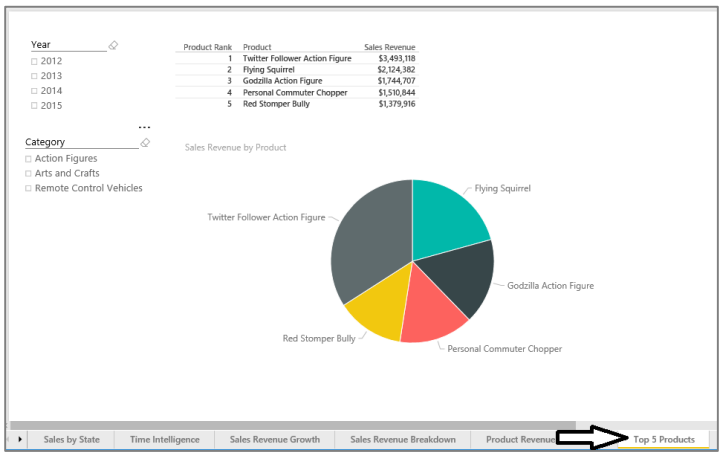

11. Add two more tiles to the dashboard by pinning visuals from the **Top 5 Product** report page.

a) Click on the report named **WingtipSalesAnalysis** in the **Reports** section.

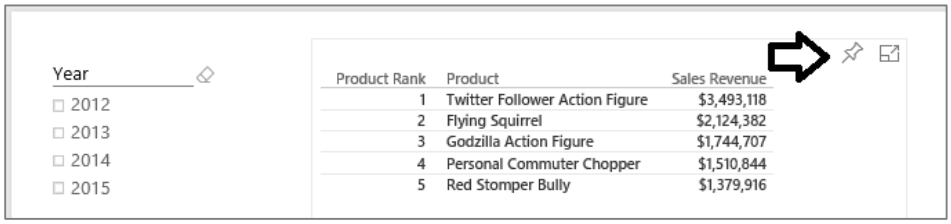
b) Navigate to the **Top 5 Product** page.

c) Make sure that both the slicers on the page are cleared so that the other visuals reflect sales revenue for all sales.

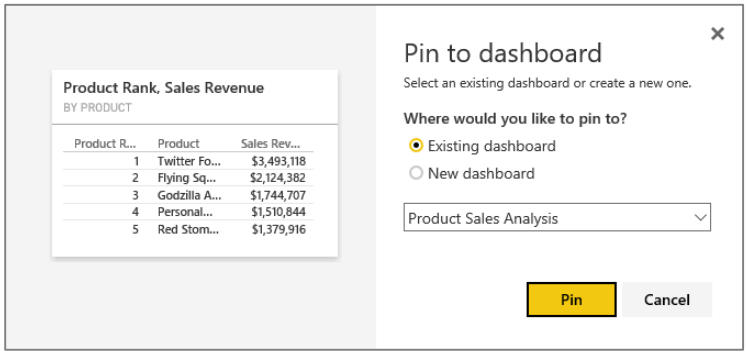
© Critical Path Training. 2015. All Rights Reserved



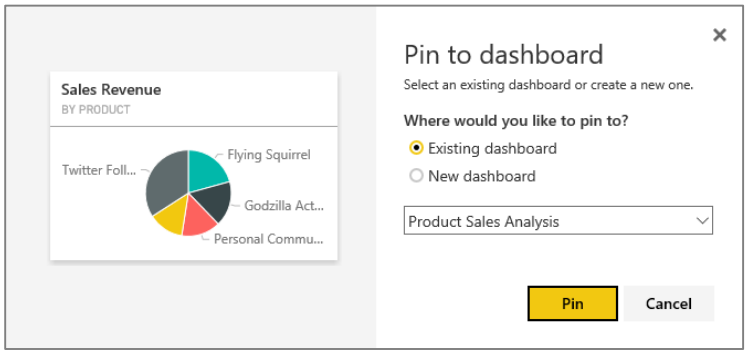

d) Select the table visual with the top five products. Click the thumbtack icon button to display the **Pin to dashboard** dialog.




e) On the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Product Sales Analysis** dashboard.

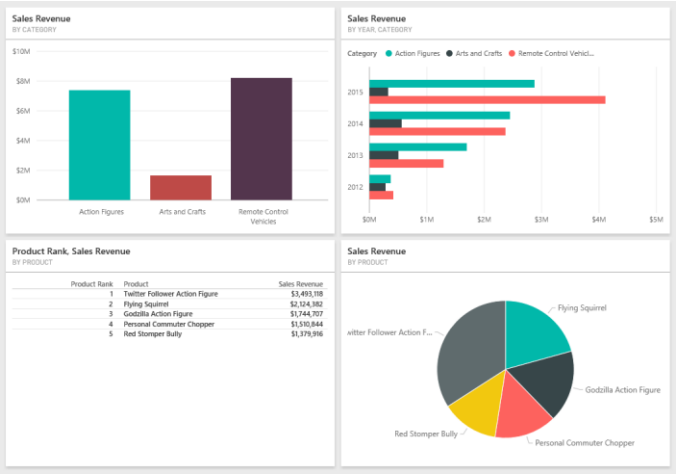



f) Go through the same steps to pin the other visual which shows a pie chart of the top 5 products.

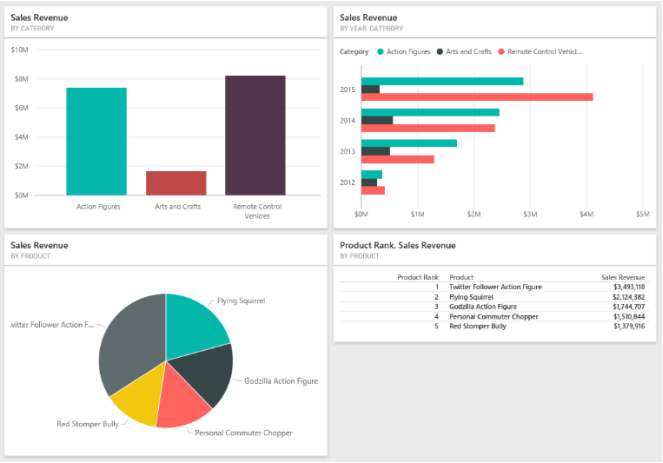



g) Navigate to the **Product Sales Analysis** dashboard and verify you see tiles for the two visuals on the **Top 5 Products** page.

© Critical Path Training. 2015. All Rights Reserved

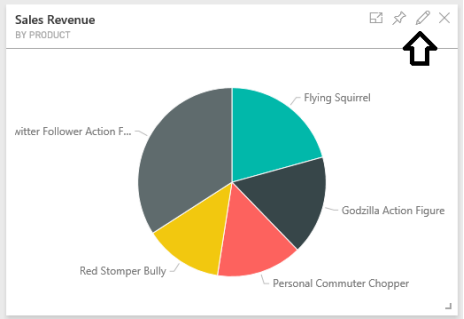



h) Rearrange the two tiles by reversing their positions and decreasing the height of tile for the table visual.



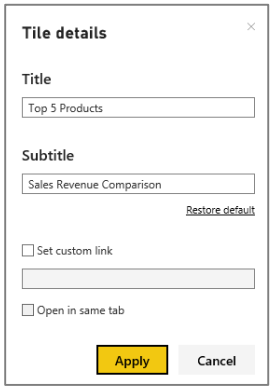

12. Modify the title and subtitle for the tile with the pie chart visual.

a) Select the tile with the pie chart visual and click on the pencil icon button to edit the tile details.

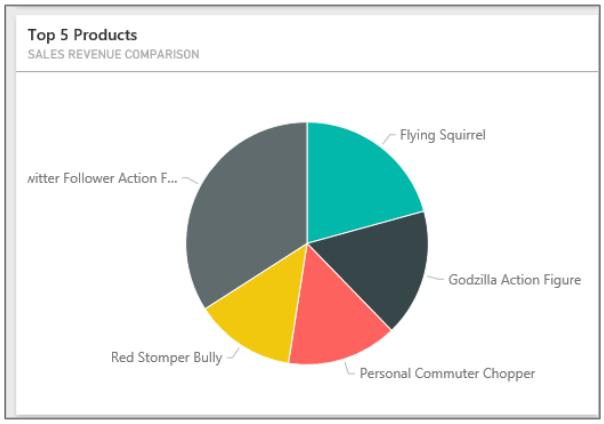



b) Enter a Title of Top 5 Products and a Subtitle of Sales Revenue Comparison and then click Apply.

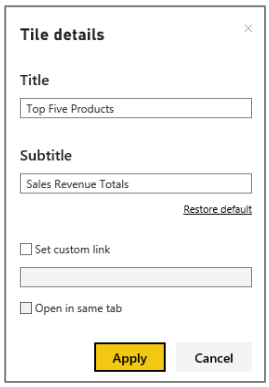
© Critical Path Training. 2015. All Rights Reserved




c) You should be able to see that the tile has been updated with a new title and subtitle.

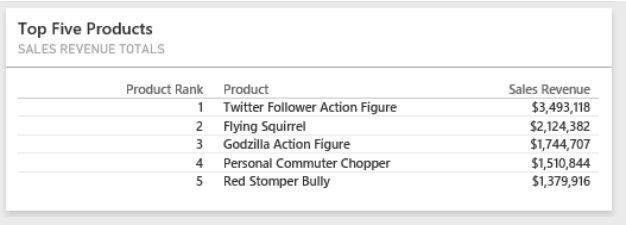



13. Modify the title and subtitle for the tile with the pie chart visual.
    a) Select the tile with the top 5 products table and click on the pencil icon button to edit the tile details.
    b) Enter a Title of Top 5 Products and a Subtitle of Sales Revenue Totals and then click Apply.




c) You should be able to see that the tile has been updated with a new title and subtitle.

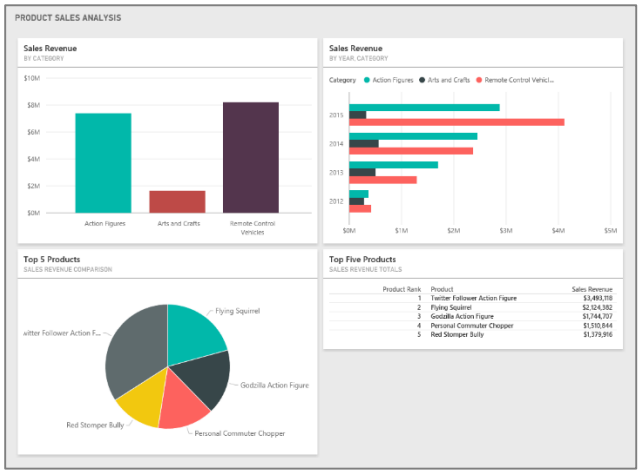
© Critical Path Training. 2015. All Rights Reserved



**Top Five Products**
SALES REVENUE TOTALS

| Product Rank | Product | Sales Revenue |
|---|---|---|
| 1 | Twitter Follower Action Figure | $3,493,118 |
| 2 | Flying Squirrel | $2,124,382 |
| 3 | Godzilla Action Figure | $1,744,707 |
| 4 | Personal Commuter Chopper | $1,510,844 |
| 5 | Red Stomper Bully | $1,379,916 |

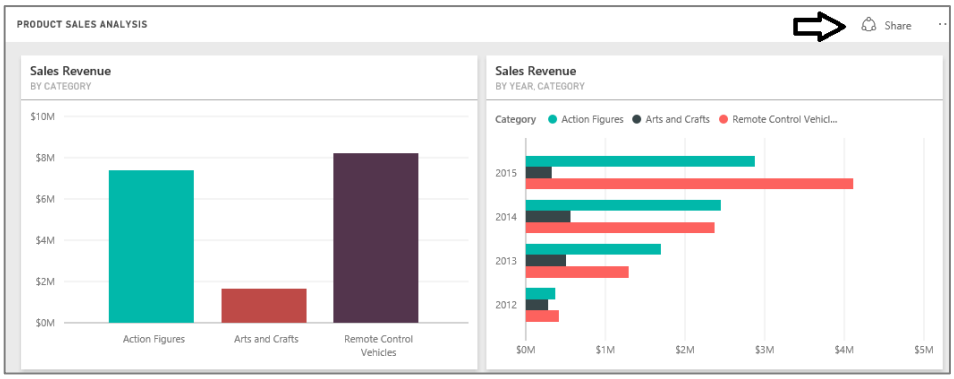
14. At this point, you are now done creating the **Product Sales Analysis** dashboard.




## Exercise 2: Share the Product Sales Analysis Dashboard

In this exercise you will share the Products Sales Analysis dashboard and test it out from the perspective of a user who is a dashboard consumer as opposed to a dashboard author.

1. Navigate to the **Product Sales Analysis** dashboard.
2. Click the **Share** link at the top, right-hand side of the page.

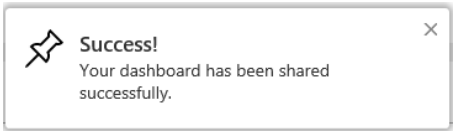



3. At this point, you are prompted with the **Share Dashboard** page where you can enter information about the users and/or groups with which you want to share your new dashboard. Enter the email address of the secondary user account that you created earlier in the very first lab.

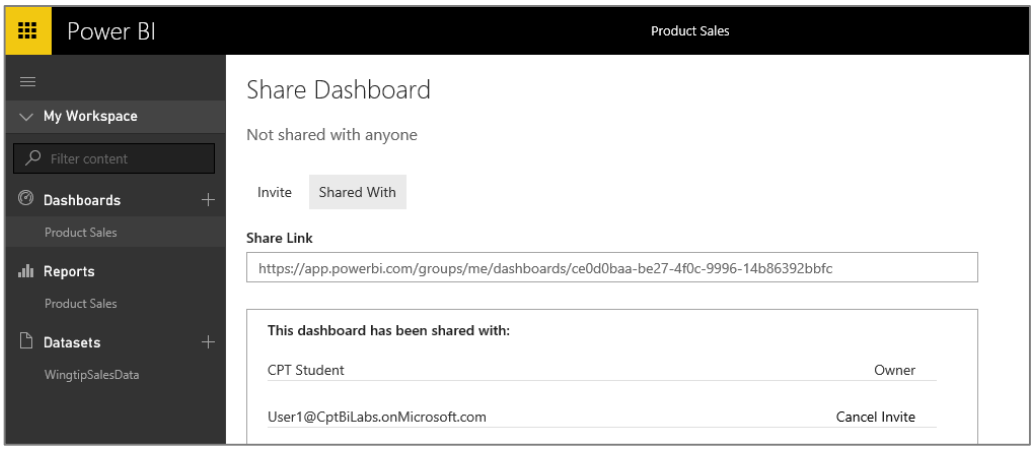
© Critical Path Training. 2015. All Rights Reserved

4. After you have entered the email address, click the **Share** button at the bottom of the page.

5. The Power BI service responds with even more positive reinforcement.

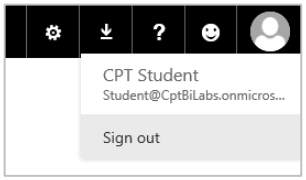



6. Once the dashboard has been successfully shared, you should be able to confirm that the secondary user has access to this dashboard by examining the **Shared With** tab of the **Share Dashboard** page as shown in the following screenshot.



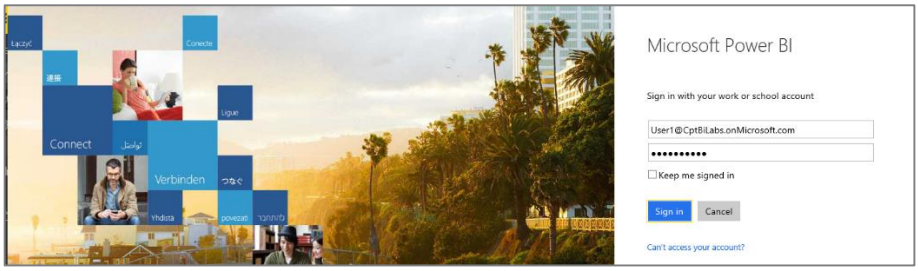

Now you have completed the steps to share the dashboard, the next step is to test out the dashboard experience when logged in as a user who is not the dashboard author, but instead a dashboard consumer. This will require that you sign out of the Power BI service and then sign back in under the identity of the secondary user account. By accessing the shared dashboard in this fashion, you will be able to observe the typical experience of a dashboard consumer when accessing a dashboard that has been shared by another user.

7. Drop down the user menu from the top, right-hand corner of the page and click the **Sign out** command.

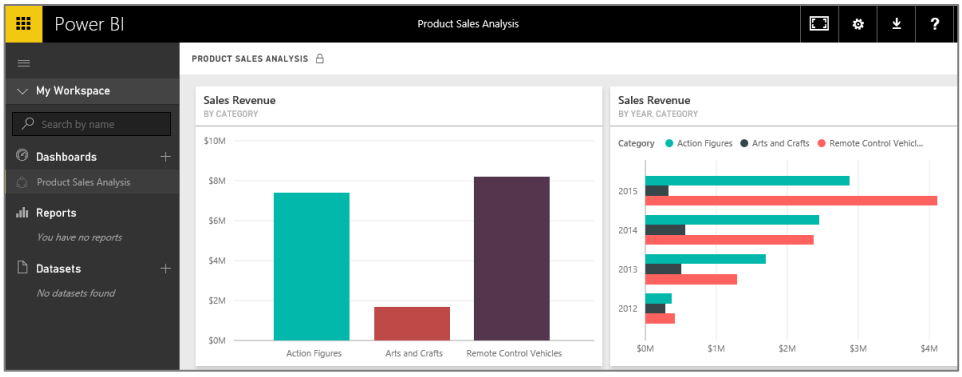



© Critical Path Training. 2015. All Rights Reserved

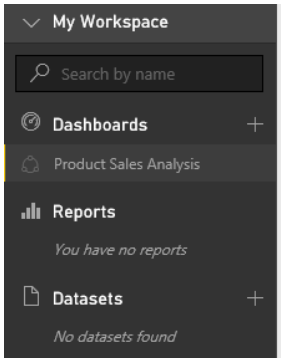
8. Now, sign back in using the account name and the password of the secondary user account you created earlier.




9. Once you have signed in, navigate to the Power BI service at https://app.powerbi.com. You should be able to see that the Product Sales Analysis dashboard is accessible to you through your personal workspace.



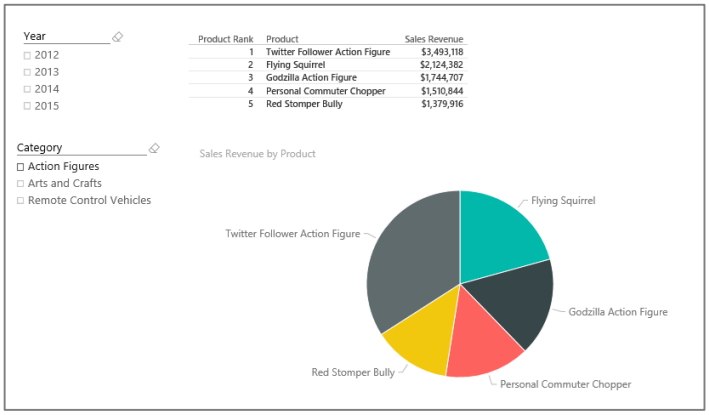

10. You should be able to verify that the current user can see the **Product Sales Analysis** dashboard link in the left navigation menu but the current user cannot see links to any of the reports or datasets behind the dashboard.



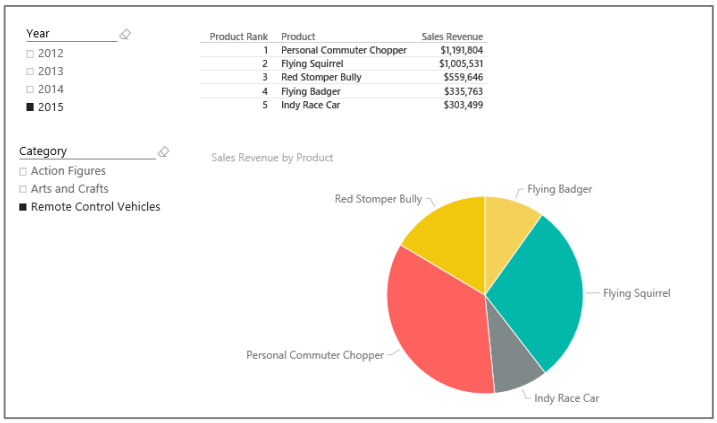

Note that Power BI does not include links to provide the dashboard consumer with direct access to the report or the dataset behind the dashboard. However, Power BI does supply the dashboard consumer with indirect access to the report and the dataset behind the dashboard. It's just that the dashboard consumer can only access the report and dataset by interacting with the dashboard. A key benefit is that this approach keeps the left navigation less cluttered when the user is accessing many different shared dashboards. Another benefit as that the dashboard can serve as a security boundary on which you can configure permissions to grant users access to reports and datasets.

11. Navigate to the **Product Sales Analysis** dashboard and observe that the dashboard page is in read-only view. The user is not able to delete, move or resize any of the tiles. Only the dashboard author is able to edit this dashboard page.
12. On the **Product Sales Analysis** dashboard, experiment by clicking on the dashboard tile with the three columns at the top right corner. This will allow you to navigate to the **Product Revenue Breakdown** page of the report behind the dashboard.
13. Navigate back to the **Product Sales Analysis** dashboard.
14. Click one of the two tiles in the bottom row to navigate the **Top 5 Products** report page.

© Critical Path Training. 2015. All Rights Reserved




15. Once you see the **Top 5 Products** report page, interact with the report by selecting different years and product categories to see how individual products are selling from year to year and across product categories.




You are now finished with your dashboard testing.

16. Sign out the current user and then log back in using your primary Office 365 account.

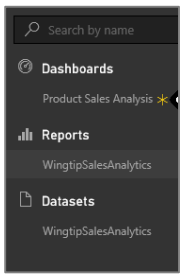
## Exercise 3: Create the Regional Sales 2015 Dashboard

In this exercise you will begin by creating a new report named **2015 Regional Sales** which pulls its data from the **WingtipSalesAnalysis** dataset that that you created over the last few labs. After creating the new report, you will then pin visuals from this report to a new dashboard named **Regional Sales 2015**.

**IMPORTANT**: In case you somehow missed the last step of the previous exercise, make sure you are no longer running as the secondary user. You should have logged out at the end of the previous exercise and then you should have logged back in with the primary Office 365 account you have been using to published your Power BI Desktop project to the Power BI service.

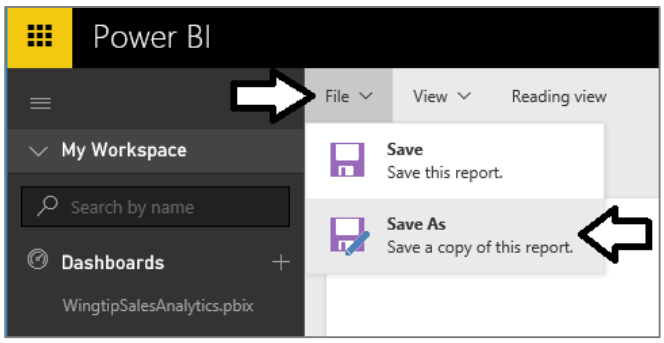
1. Navigate to the Power BI service at https://app.powerbi.com if you are not already there.
2. Make sure you are running in the context of your personal workspace.
3. Create the **Regional Sales 2015** report.
   a) Click the **WingtipSalesAnalysis** link in the **Datasets** section.

© Critical Path Training. 2015. All Rights Reserved

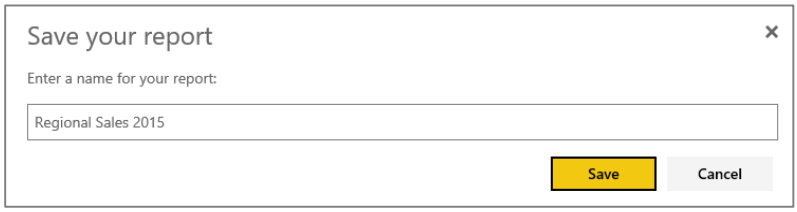



b) When you navigate to the **WingtipSalesAnalysis** dataset, the Power BI service displays a new empty report.

c) Save the report by dropping down the **File** menu and select the **Save** menu.

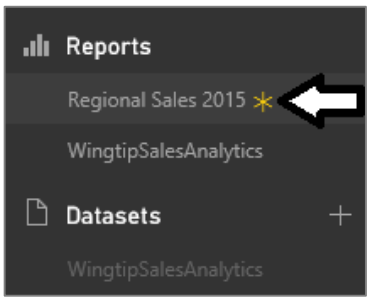



d) In the **Save your report** dialog, enter a report name of **Regional Sales 2015** and click **Save**.



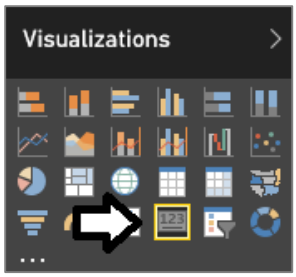

e) You should now be able to see the new report in the **Reports** section.

© Critical Path Training. 2015. All Rights Reserved

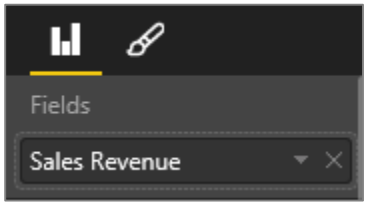



4. The report current has one page name **Page 1**. Modify the page name to **US Sales**.

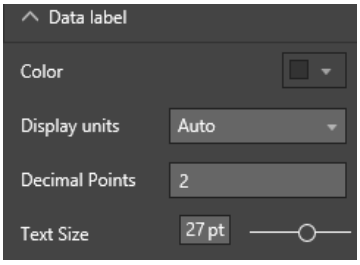
5. Add a card visual to the page to display to sales revenue for the year **2015**.
   a) Add a new Card visual to the page.




   b) Drag the **Sales Revenue** measure from the **Sales** table into the **Fields** well.

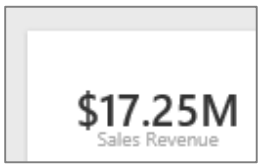



   c) Click the brush icon button in the **Visualizations** page to display the visual's Format properties.
   d) In the **Data label** section, set the **Decimal Point** properties to **2** and the **Text Size** property to **27 pt**.



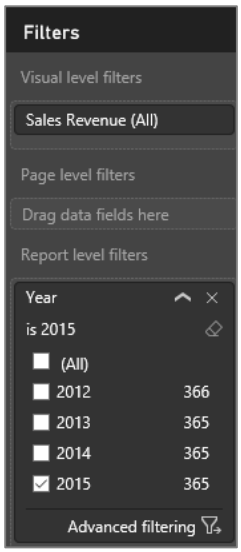

   e) The visual should appear like the visual shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved



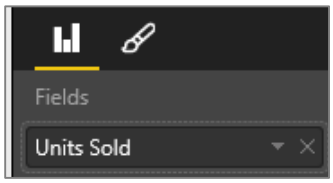

There is a problem because the card visual is displays a sum of sales revenue across all years instead of a total for the year 2015.

6. Set a report-level filter for the year 2015.
   a) Drag the **Year** column from the **Calendar** table in the **Fields** list and drop it into **Report level filters** section at the bottom of the **Visualization** pane. Set the report level filter to the year 2015 as shown in the following screenshot.



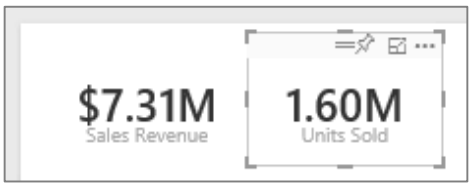

   b) The Card visual should now refresh display the sales revenue for just the year 2015.

7. Add a second card visual to the page to display the number units sold for the year **2015**.
   a) Select the card visual which displays sales revenue.
   b) Copy and paste the visual to make a cloned copy.
   c) Select the copy and modify it so it uses the Units Sold measure instead of the Sales Revenue measure.

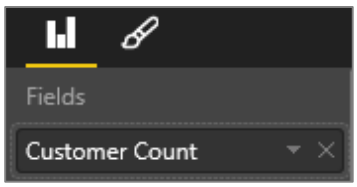



   d) Arrange the two visuals so they are side by side on the page.

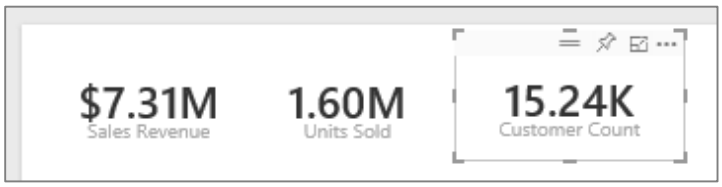
© Critical Path Training. 2015. All Rights Reserved



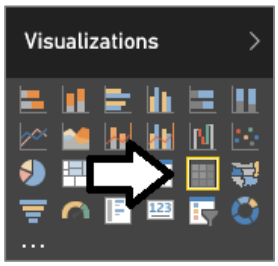

8. Add a third card visual to the page to display the total number of customers for the year **2015**.
   a) Select the card visual which displays sales revenue.
   b) Copy and paste the visual to make a cloned copy.
   c) Select the copy and modify it so it uses the **Customer Count** measure instead of the **Sales Revenue** measure.




   d) Arrange the three visuals so they are side by side on the page.

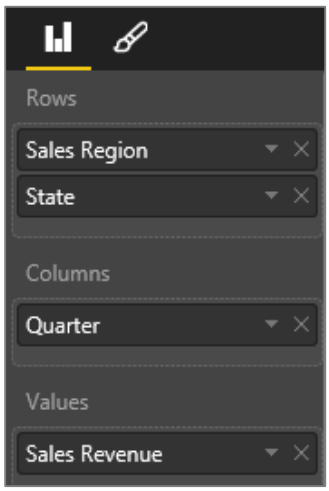



9. Add a new matrix visual to the page to visualize how 2015 sales revenue breaks down across sales region, state and quarter.
   a) Add a new matrix visual to the page.




   b) Drag the **Sales Regions** column from the **Customers** table into the **Rows** well.
   c) Drag the **State** column from the **Customers** table into the **Rows** well.
   d) Drag the **Quarter** column from the **Calendar** table into the **Columns** well.
   e) Drag the **Sales Revenue** measure from the **Sales** table into the **Values** well.

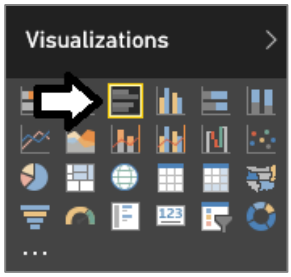
© Critical Path Training. 2015. All Rights Reserved




f) The visual should now appear as the visual shown in the following screenshot.

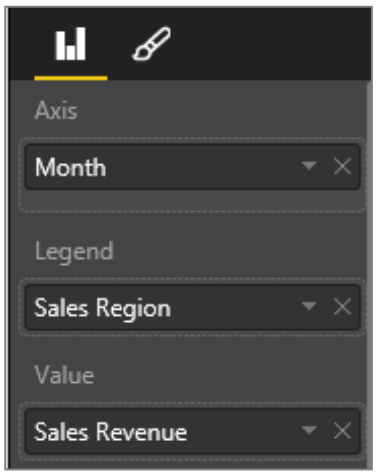
| Sales Region | State | 2015-Q1 | 2015-Q2 | 2015-Q3 | 2015-Q4 | Total |
|---|---|---|---|---|---|---|
| Central Region | AL | $32,490 | $31,474 | $32,862 | $27,606 | $124,433 |
| | CO | $62,219 | $56,826 | $83,215 | $70,093 | $272,353 |
| | LA | $31,390 | $37,177 | $46,920 | $35,338 | $150,824 |
| | OH | $47,054 | $47,603 | $46,947 | $44,504 | $186,107 |
| | TN | $52,871 | $53,173 | $51,099 | $58,111 | $215,254 |
| | TX | $197,849 | $232,880 | $256,854 | $231,447 | $919,030 |
| | Total | $423,873 | $459,132 | $517,896 | $467,099 | $1,868,000 |
| Eastern Region | CT | $56,312 | $61,126 | $56,120 | $71,318 | $244,876 |
| | FL | $199,071 | $239,994 | $220,827 | $231,453 | $891,345 |
| | GA | $96,214 | $102,793 | $107,631 | $110,399 | $417,037 |
| | MA | $54,789 | $64,603 | $64,721 | $59,750 | $243,864 |
| | NC | $107,437 | $112,272 | $114,401 | $114,530 | $448,639 |
| | NH | $23,469 | $11,358 | $24,744 | $20,925 | $80,496 |
| | NJ | $41,600 | $51,876 | $58,096 | $55,737 | $207,308 |
| | NY | $85,329 | $108,225 | $112,466 | $124,951 | $430,971 |
| | PA | $65,456 | $62,837 | $71,976 | $92,294 | $292,563 |
| | RI | $10,550 | $16,834 | $15,003 | $18,287 | $60,675 |
| | SC | $28,253 | $39,834 | $28,912 | $36,352 | $133,352 |
| | VA | $86,496 | $92,707 | $98,455 | $84,224 | $361,882 |
| | Total | $854,977 | $964,460 | $973,352 | $1,020,220 | $3,813,009 |
| Western Region | AZ | $56,910 | $81,762 | $76,120 | $66,921 | $281,713 |
| | CA | $179,554 | $210,179 | $187,467 | $193,201 | $770,402 |
| | NM | $33,405 | $38,447 | $32,404 | $34,939 | $139,195 |
| | OR | $39,729 | $35,868 | $45,376 | $51,192 | $172,165 |
| | UT | $18,010 | $18,002 | $26,547 | $20,578 | $83,137 |
| | WA | $36,172 | $50,623 | $53,627 | $43,618 | $184,039 |
| | Total | $363,779 | $434,882 | $421,541 | $410,449 | $1,630,651 |
| Total | | $1,642,629 | $1,858,474 | $1,912,789 | $1,897,768 | $7,311,660 |

10. Add a new clustered bar chart visual to the page to show how 2015 sales revenue breaks down across sales region and month.
    a) Add a new clustered bar chart visual to the page.

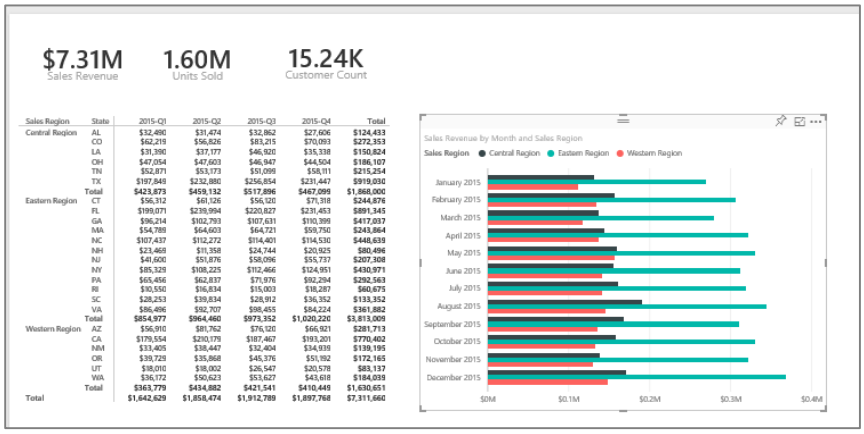



    b) Drag the **Month** column from the **Calendar** table into the **Axis** well.
    c) Drag the **Sales Regions** column from the **Customers** table into the **Legend** well.
    d) Drag the **Sales Revenue** measure from the **Sales** table into the **Value** well.

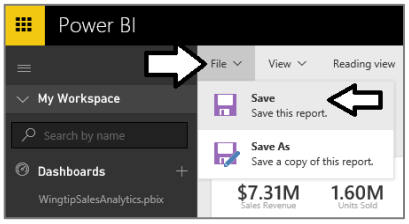
© Critical Path Training. 2015. All Rights Reserved



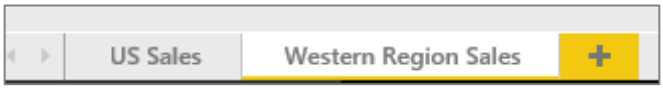

e) The visual should now appear as the visual shown in the following screenshot.




11. Save the work you have done by drop down the **File** menu and select the **Save** command.

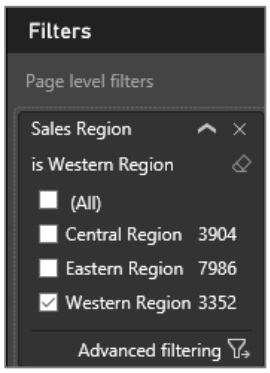



12. Add a new page and rename it to **Western Sales Region**.

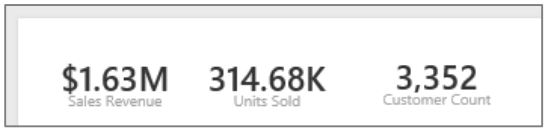



13. Set a page-level filter for the wester sales region.
    a) Drag the **Sales Region** column from the **Customers** table in the **Fields** list and drop it into **Page level filters** section at the bottom of the **Visualization** pane.
    b) Set the page level filter to **Western Region** as shown in the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

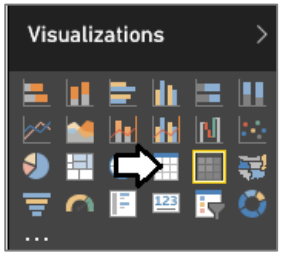



14. Copy and paste the three card visuals from the **US Sales** page to the **Western Regional Sales** page.
    a) Navigate to the **US Sales** page.
    b) Select the card visual on the left.
    c) Copy the card visual to the Windows clipboard.
    d) Navigate the Western Regional Sales page.
    e) Paste the visual in the windows clipboard to the **Western Regional Sales** page.
    f) Repeat the same steps to copy the second and third card visual as well.



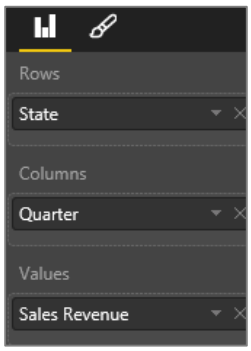

You can observe that the card visuals on the **Western Regional Sales** page do not return the same results as the card visuals on the **US Sales** page. That's because of the page-level filter you applied after creating the **Western Regional Sales** page.

15. Add a new matrix visual to the page to show how sales revenue is broken out across state and quarter.
    a) Add a new matrix visual to the page.




    b) Drag the **State** column from the **Customers** table into the **Rows** well.
    c) Drag the **Quarter** column from the **Calendar** table into the **Columns** well.
    d) Drag the **Sales Revenue** measure from the **Sales** table into the **Values** well.

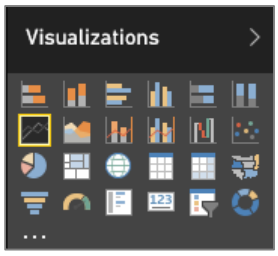
© Critical Path Training. 2015. All Rights Reserved




e) The visual should now appear as the visual shown in the following screenshot.

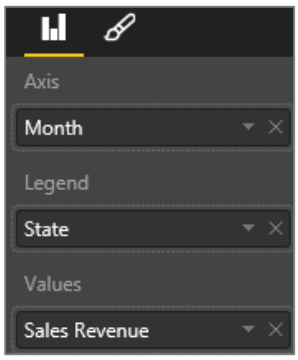
| State | 2015-Q1 | 2015-Q2 | 2015-Q3 | 2015-Q4 | Total |
|---|---|---|---|---|---|
| AZ | $56,910 | $81,762 | $76,120 | $66,921 | $281,713 |
| CA | $179,554 | $210,179 | $187,467 | $193,201 | $770,402 |
| NM | $33,405 | $38,447 | $32,404 | $34,939 | $139,195 |
| OR | $39,729 | $35,868 | $45,376 | $51,192 | $172,165 |
| UT | $18,010 | $18,002 | $26,547 | $20,578 | $83,137 |
| WA | $36,172 | $50,623 | $53,627 | $43,618 | $184,039 |
| Total | $363,779 | $434,882 | $421,541 | $410,449 | $1,630,651 |

16. Add a new line chart visual to the page to show sales revenue by quarter split out across sales regions.
    a) Add a new line chart visual to the page.

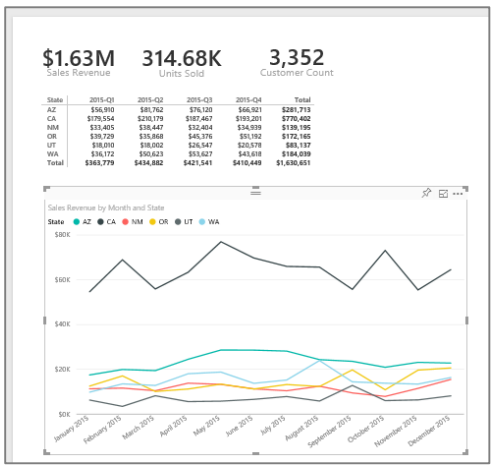



    b) Drag the **Month** column from the **Calendar** table into the **Axis** well.
    c) Drag the **Sales Regions** column from the **Customers** table into the **Legend** well.
    d) Drag the **Sales Revenue** measure from the **Sales** table into the **Value** well.

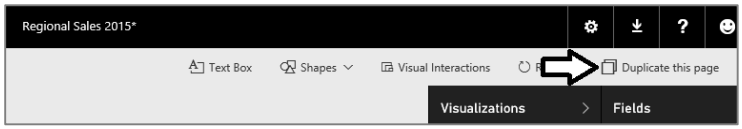
© Critical Path Training. 2015. All Rights Reserved




e) The line chart visual should now appear as the visual shown in the following screenshot.



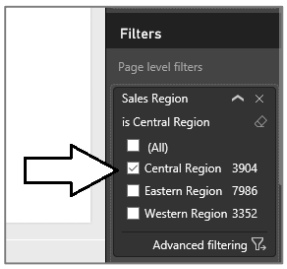

17. Save your work by dropping down the **File** menu and selecting the **Save** command.
18. Duplicate the Western Regional Sales page to create the Central Regional Sales page.
    a) Click the Duplicate this page button.



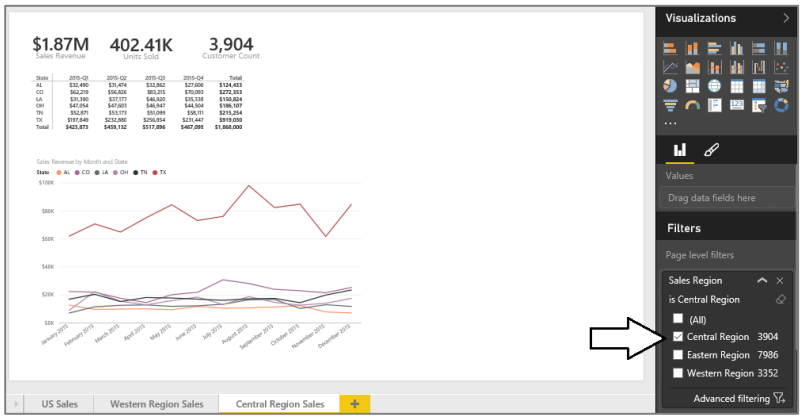

    b) Rename the new page to **Central Regional Sales**.

    c) Modify the page-level filtering to filter on **Central Region**.

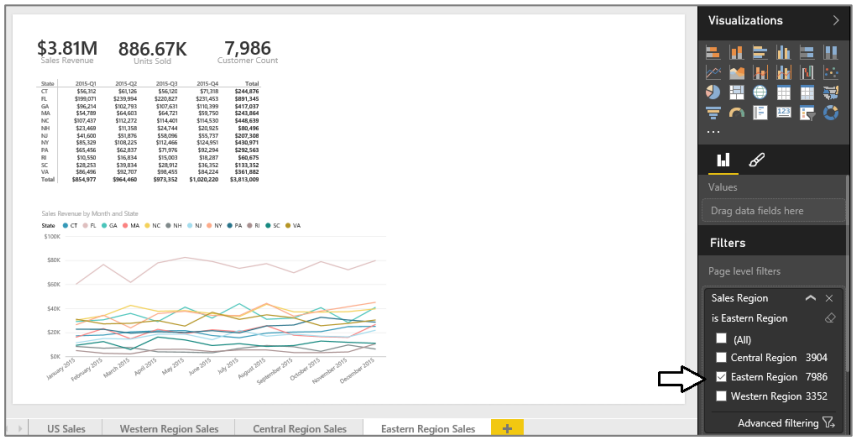
© Critical Path Training. 2015. All Rights Reserved




d) Now the **Western Regional Sales** page and the **Central Regional Sales** page display different results even though they are using the exact same visuals.

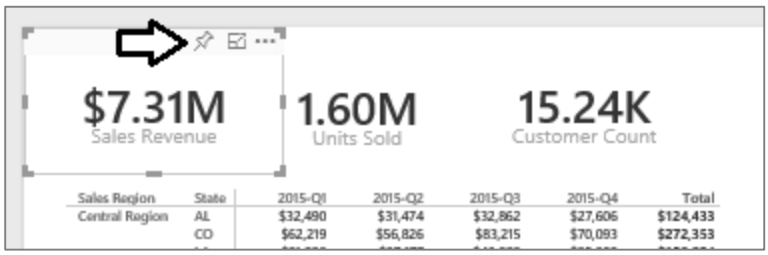



19. Duplicate the Central Regional Sales page to create the Eastern Regional Sales page.
    a) Click the Duplicate this page button.
    b) Rename the new page to **Eastern Regional Sales**.
    c) Modify the page-level filtering to filter on **Eastern Region**.

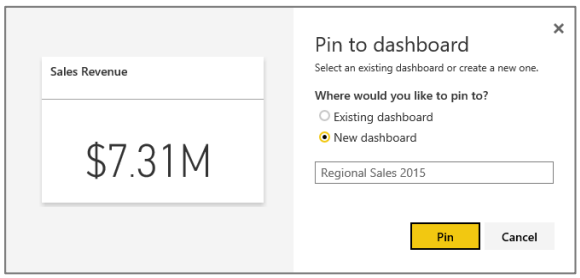



20. Save your work by dropping down the **File** menu and selecting the **Save** command.
21. Create the **Regional Sales 2015** dashboard.
    a) Navigate to the **US Sales** page.
    b) If you select the card visual on the left, you will notice several icon buttons in the upper-right corner. Click on the thumbtack icon button to pin the visual to display the **Pin to dashboard** dialog.

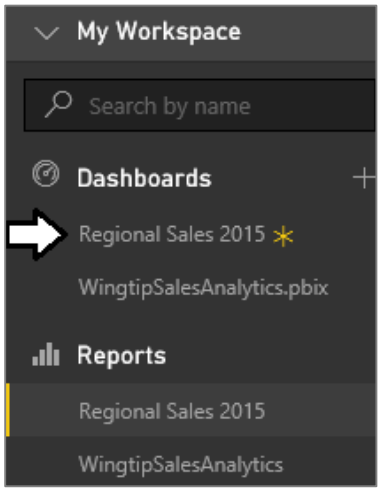
© Critical Path Training. 2015. All Rights Reserved



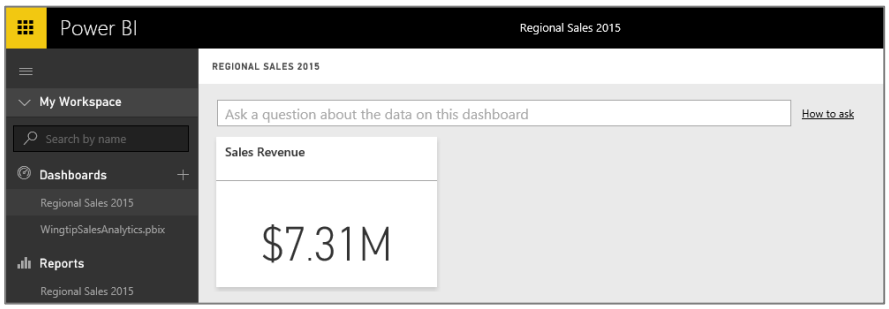

c) The **Pin to dashboard dialog** prompts you enter name for a new dashboard name. Enter a value of **Regional Sales 2015** as the new dashboard name and then click the **Pin** button to create the new dashboard and pin the visual to it.




d) At this point, you should be able to see the new **Regional Sales 2015** dashboard in the **Dashboards** section

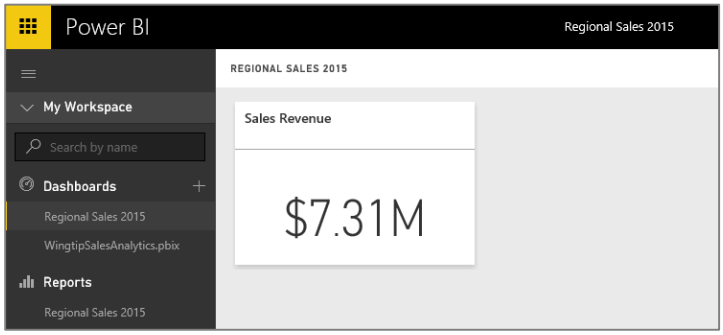



22. Examine the **Regional Sales 2015** dashboard.
    a) Click on the link for the **Regional Sales 2015** dashboard in the **Dashboards** section.
    b) Examine the dashboard. It currently contains the Q&A search box and a tile for the card visual.

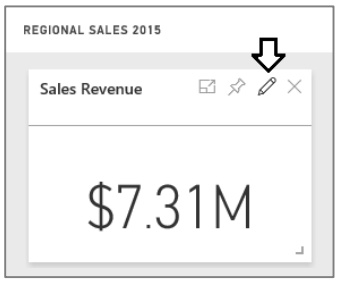



© Critical Path Training. 2015. All Rights Reserved

23. Remove the Q&A search box from the dashboard.
    a) Use the fly out menu of the for the **Regional Sales 2015** dashboard and click the **SETTINGS** menu.
    b) On the **Settings** page for the dashboard, navigate the **Dashboards** tab.
    c) Uncheck the checkbox with the caption of **Show the Q&A search box**.
    d) Click the **Apply** button.
    e) You should be able to verify that the dashboard no longer displays the Q&A search box.

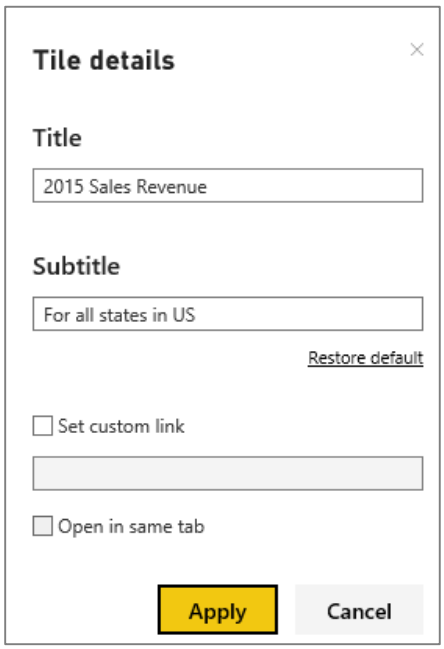



24. Modify the title and subtitle for the tile with the **Sales Revenue** card visual.
    a) Select the tile for the **Sales Revenue** card visual and click on the pencil icon button to edit the tile details.



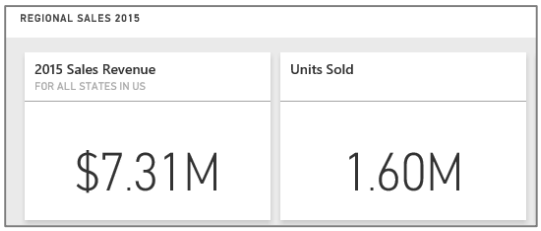

    b) Enter a Title of 2015 Sales Revenue and a Subtitle of For all states in US and then click Apply.

© Critical Path Training. 2015. All Rights Reserved



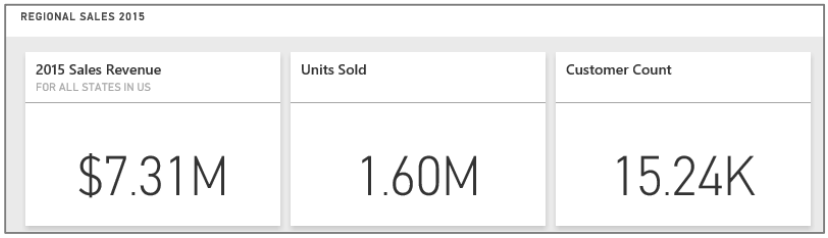

c) You should be able to see that the tile has been updated with a new title and subtitle.

25. Pin the **Units Sold** visual from the **US Sales** page to the **Regional Sales Revenue** dashboard.
    a) Navigate to the **US Sales** page.
    b) Select the card visual with Units Sold and click on the thumbtack icon button to display the **Pin to dashboard** dialog.
    c) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Regional Sales 2015** dashboard.
    d) Navigate to the **Regional Sales 2015** dashboard and confirm a new tile has been added for the **Units Sold** card visual.

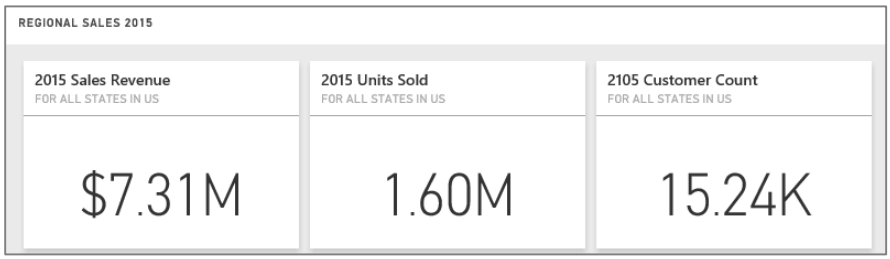



26. Pin the **Customer Count** visual from the **US Sales** page to the **Regional Sales Revenue** dashboard.
    a) Navigate to the **US Sales** page.
    b) Select the card visual with Customer Count and click on the thumbtack icon button to display the **Pin to dashboard** dialog.
    c) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Regional Sales 2015** dashboard.
    d) Navigate to the **Regional Sales 2015** dashboard and confirm a new tile has been added for the **Customer Count** card visual.

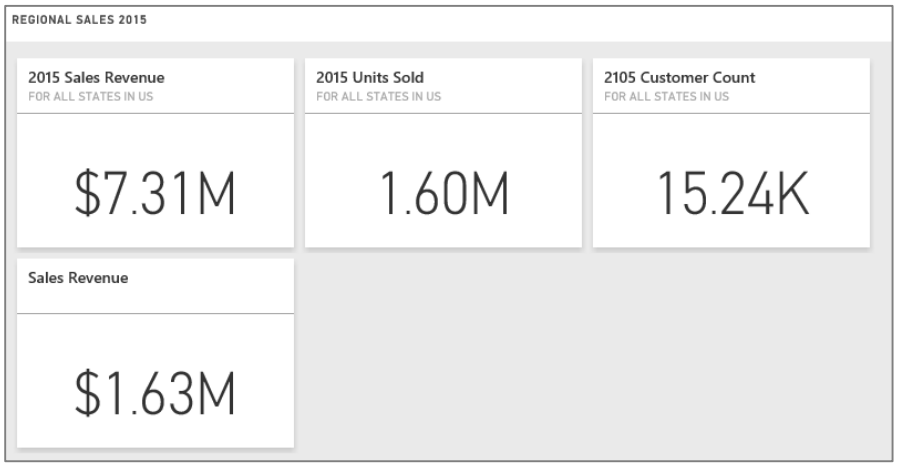
© Critical Path Training. 2015. All Rights Reserved




27. Modify the title and subtitle of the two new tiles.
    a) Select the **Units Sold** card tile and click on the pencil icon button to edit the tile details.
    b) Enter a Title of 2015 Units Sold and a Subtitle of For all states in US and then click Apply.
    c) Select the **Customer Count** card tile and click on the pencil icon button to edit the tile details.
    d) Enter a Title of 2015 Customer Count and a Subtitle of For all states in US and then click Apply.




28. Pin the Sales Revenue visual from the Western Region Sales page to the Regional Sales Revenue dashboard.
    a) Navigate to the **Western Region Sales** page.
    b) Select the card visual with Sales Revenue and click on the thumbtack icon button to display the **Pin to dashboard** dialog.
    c) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Regional Sales 2015** dashboard.
    d) Navigate to the **Regional Sales 2015** dashboard and confirm a new tile has been added for the **Sales Revenue** card visual.

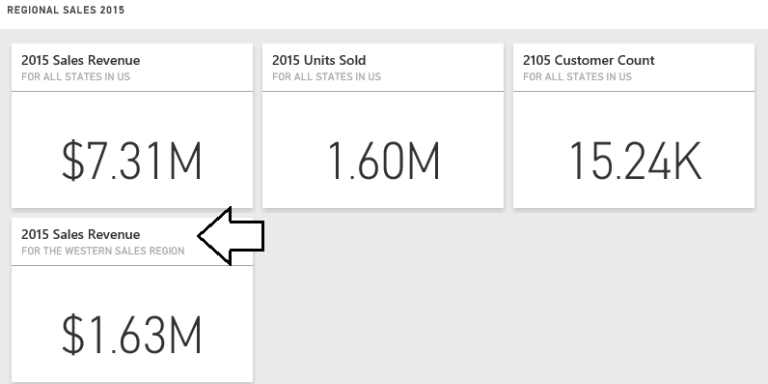



You can see there is a potential problem because the tile only displays a title of **Sales Revenue** but the tile does not indicate that this revenue figure has been calculated for the western sales region. Therefore, you must modify the title of the tile to make this clear.

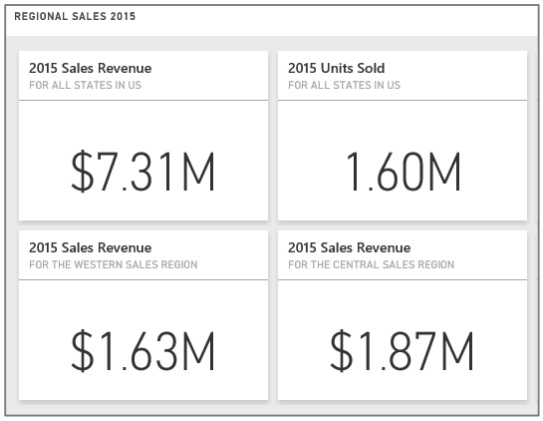
29. Modify the title and subtitle for the new **Sales Revenue** tile.
    a) Select the new tile for western sales revenue and click on the pencil icon button to edit the tile details.
    b) Enter a Title of 2015 Sales Revenue and a Subtitle of For the Western Sales Division and then click Apply.

© Critical Path Training. 2015. All Rights Reserved

c) You should be able to see that the tile has been updated with a new title and subtitle.

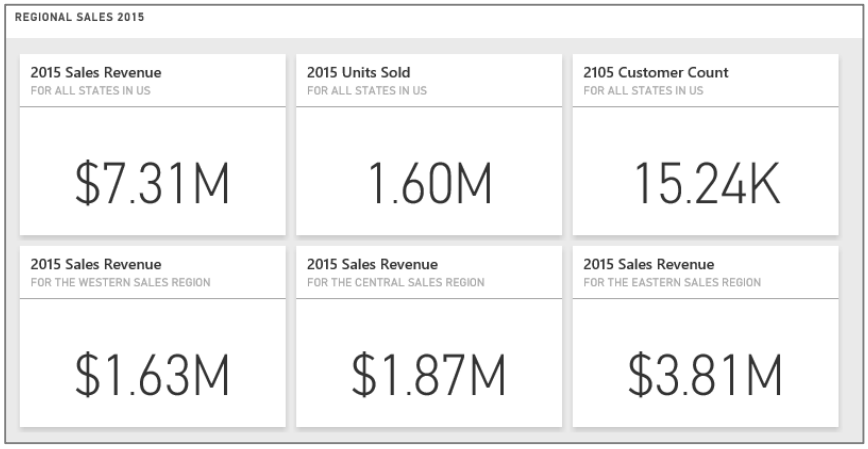



30. Pin the Sales Revenue visual from the Central Region Sales page to the Regional Sales Revenue dashboard.
    a) Navigate to the **Central Region Sales** page.
    b) Select the card visual with Sales Revenue and click on the thumbtack icon button to display the **Pin to dashboard** dialog.
    c) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Regional Sales 2015** dashboard.
    d) Navigate to the **Regional Sales 2015** dashboard and confirm a new tile has been added for the **Sales Revenue** card visual.
31. Modify the title and subtitle for the new **Sales Revenue** tile.
    a) Select the new tile for western sales revenue and click on the pencil icon button to edit the tile details.
    b) Enter a Title of 2015 Sales Revenue and a Subtitle of For the Central Sales Division and then click Apply.

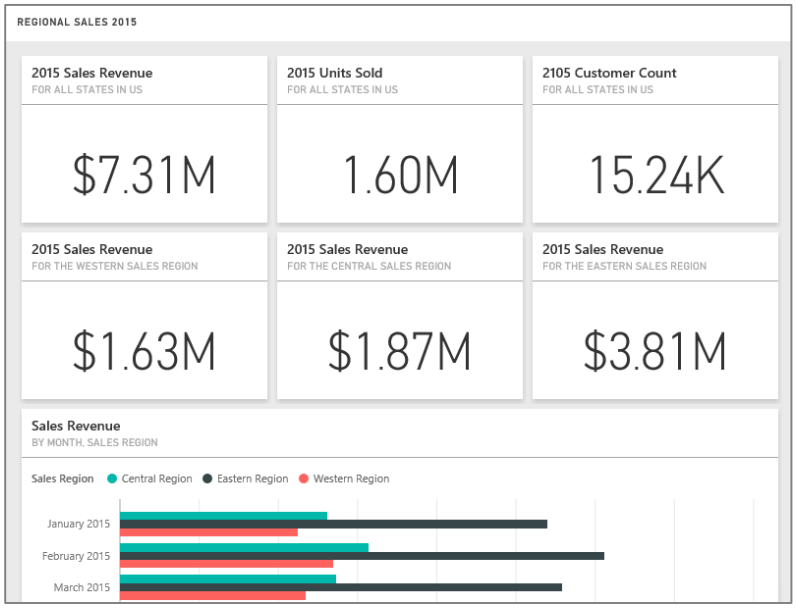



32. Pin the Sales Revenue visual from the Eastern Region Sales page to the Regional Sales Revenue dashboard.
    a) Navigate to the **Eastern Region Sales** page.
    b) Select the card visual with Sales Revenue and click on the thumbtack icon button to display the **Pin to dashboard** dialog.
    c) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Regional Sales 2015** dashboard.
    d) Navigate to the **Regional Sales 2015** dashboard and confirm a new tile has been added for the **Sales Revenue** card visual.
33. Modify the title and subtitle for the new **Sales Revenue** tile.
    a) Select the new tile for western sales revenue and click on the pencil icon button to edit the tile details.
    b) Enter a Title of 2015 Sales Revenue and a Subtitle of For the Eastern Sales Division and then click Apply.

© Critical Path Training. 2015. All Rights Reserved




34. Pin the **Bar Chart** visual from the **US Sales** page to the **Regional Sales Revenue** dashboard.
    a) Navigate to the **US Sales** page.
    b) Select the bar chart visual with sales revenue spit out across sales region and click on the thumbtack icon button to display the **Pin to dashboard** dialog.
    c) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Regional Sales 2015** dashboard.
    d) Navigate to the **Regional Sales 2015** dashboard and confirm a new tile has been added for the **bar chart** visual.



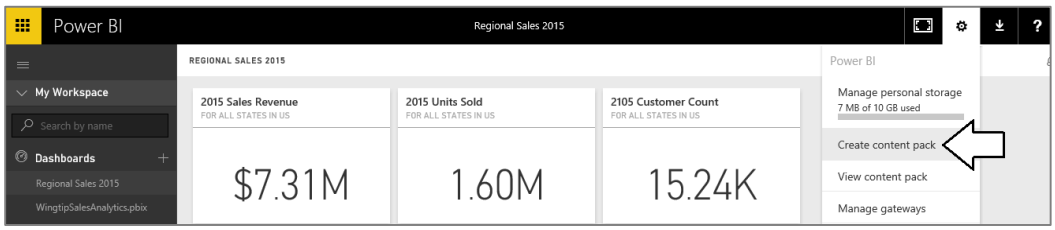

You have now finished creating the dashboard. You will now move on to the final exercise where you will deploy the dashboard using an organizational content pack.

## Exercise 4: Deploy a Dashboard using an Organizational Content Pack

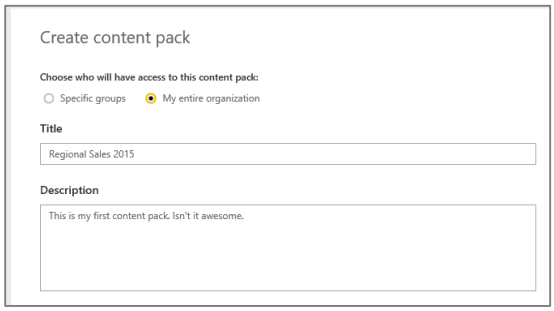
In this exercise you will create an organizational content pack to deploy the **Regional Sales 2015** dashboard. After creating the content pack for the **Regional Sales 2015** dashboard, you use the content pack to activate the **Regional Sales 2015** dashboard in a group workspace.

1. Create a content pack named the **Regional Sales 2015** to deploy the **Regional Sales 2015** dashboard.
    a) Drop down the **Settings** menu from the top right corner of the page and select the **Create content pack** command to display the **Create content pack** page.

© Critical Path Training. 2015. All Rights Reserved




b) On the **Create content pack** page, select an access level of **My entire organization** and enter a **Title** of **Regional Sales 2015**. Enter a creative description as well.




c) Below on the **Create content pack** page you should see a section to upload a custom image.

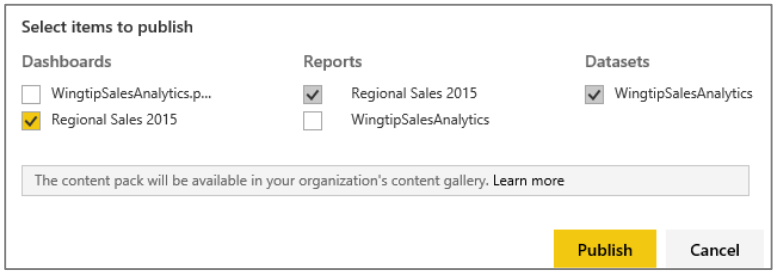
d) Click the **Upload** link and then upload the image file located at the following path.

```
C:\Student\Extras\Images\CPT\CPTAppIcon96x96.png
```

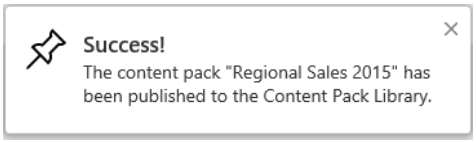
e) You should see the custom image on the **Create content pack** page.

f) At the bottom of the **Create content pack** page select the checkbox for the **Regional Sales 2015** dashboard. When you do this, the underlying report and dataset are also automatically selected.

© Critical Path Training. 2015. All Rights Reserved

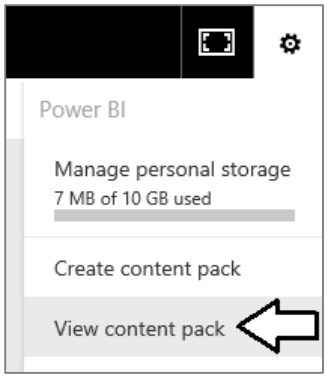



g) Click the **Publish** button. You should receive another notification of your ongoing and unstoppable success.




2. View the content pack you just created which is named **Regional Sales 2015**.

a) Drop down the **Settings** menu from the top right corner of the page and select the **View content pack** command.

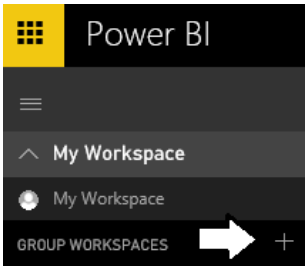



b) You should be able to see that the **Regional Sales 2015** content pack which has been published to **My Organization**.

| Name | Published To | Date published | Actions |
|---|---|---|---|
| Regional Sales 2015 | My Organization | Dec 02, 2015 | Edit \| Delete |

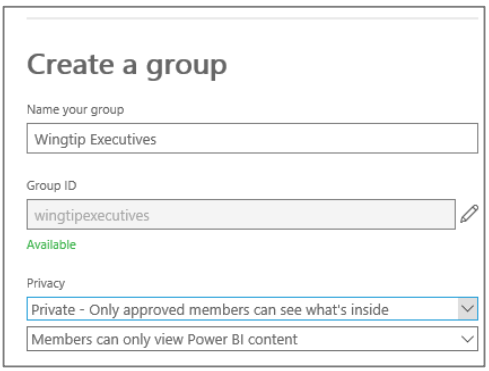
3. Create a new group workspace named **Wingtip Executives**.

a) Click on the icon button with the plus (**+**) sign to the right of the **Group Workspaces** header.

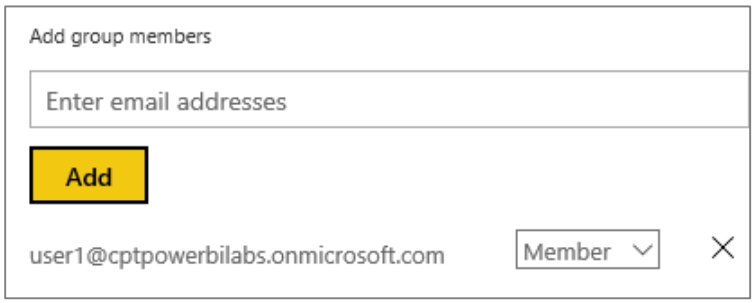



b) On the **Create a group** page, fill out the form using the data shown in the following screenshot.

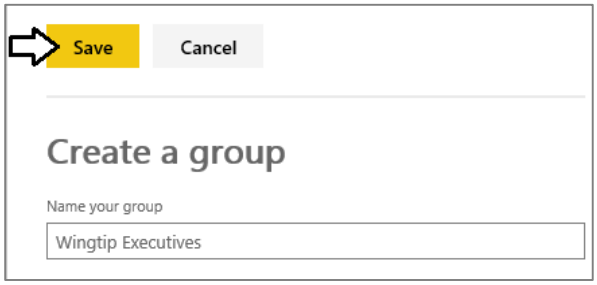
© Critical Path Training. 2015. All Rights Reserved



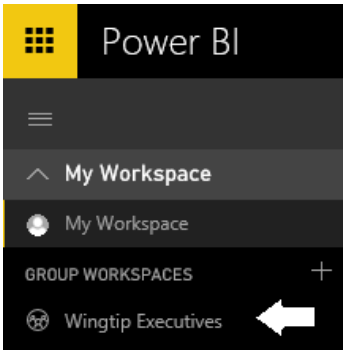

c) Enter the email address of the secondary user and click the **Add** button to add that user to the new group.




d) Click the Save button at the top of the page to save your changes.



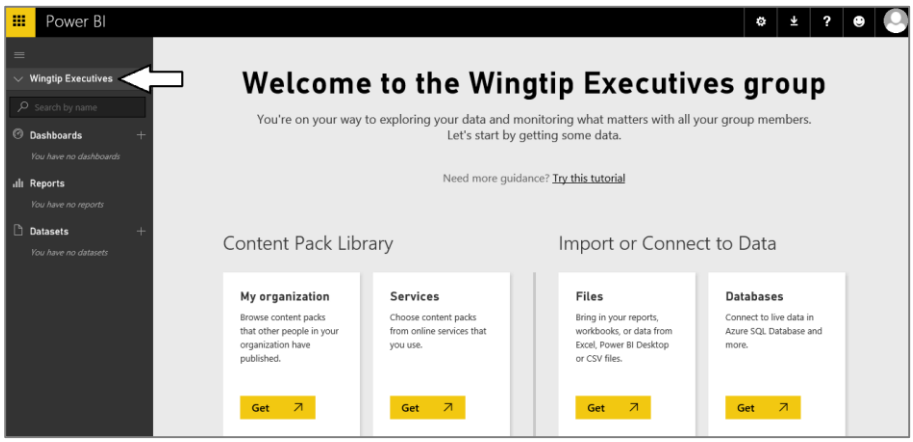

e) You should now be able to see a link to the **Wingtip Executives** group workspace in the **Group Workspaces** section of the left navigation menu.



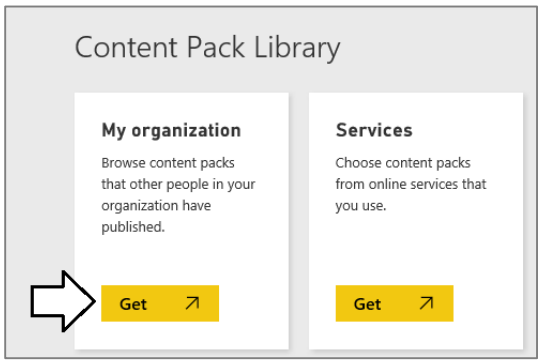

f) Click on the link for the **Wingtip Executives** group workspace.

g) The Power BI service will switch the user context from your personal workspace over to the **Wingtip Executives** workspace.

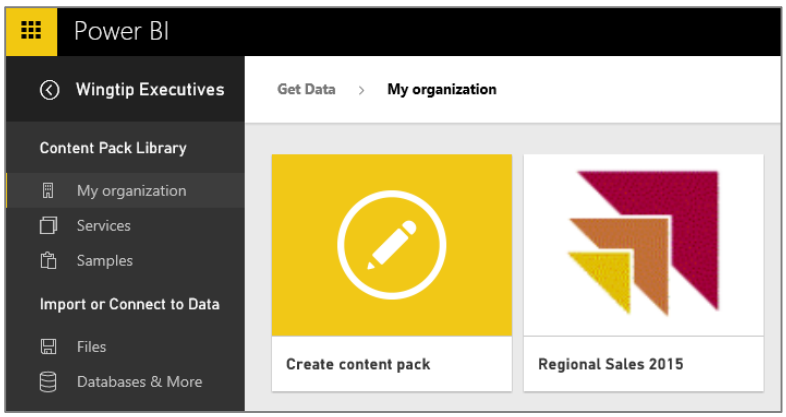
© Critical Path Training. 2015. All Rights Reserved



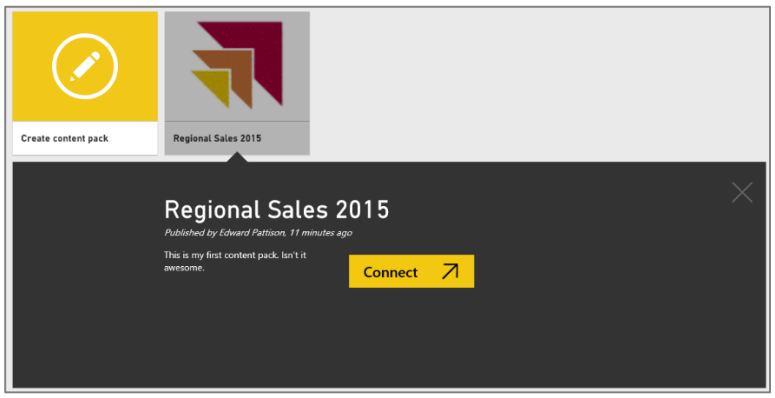

4. Activate the **Regional Sales 2015**.content pack in the **Wingtip Executives** group workspace.
   a) Click on the **Get** button the **My organization** section.



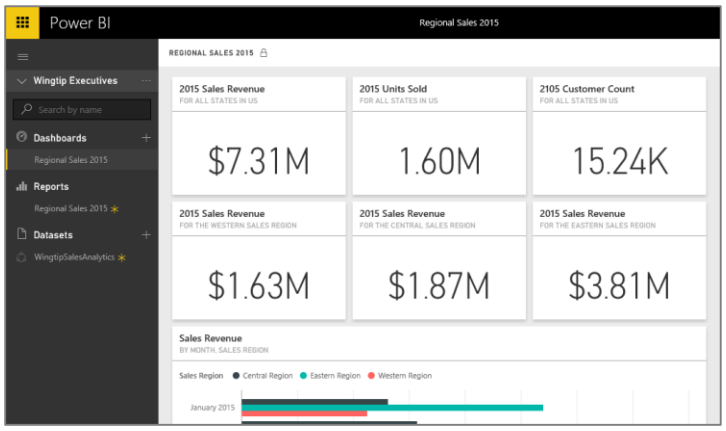

   b) Click on the **Regional Sales 2015**.content pack to invoke the **Connect** fly out menu.




   c) Click the **Connect** button.




d) Once the content pack has been activated, you should be able to access its dashboard.




You have now reach the end of this lab.

© Critical Path Training. 2015. All Rights Reserved

# Module 08: Leveraging the BI Features in Excel 2016

## Module Description

This module begins with an overview of the enhancements and BI support in Microsoft Excel 2016 along with a discussion of its strengths and weaknesses when compared to Power Bi Desktop. Students will learn how to use the enhanced support in Excel 2016 for Power Query, PowerPivot and the DAX language. The module also includes a discussion of creating charts and graphs on top of an excel data model as well as common practices in creating Excel workbooks which provides interactive dashboards.

## Module Agenda

- Data Model Support in Excel 2016
- Power Query Support in Excel 2016
- Power Pivot Support in Excel 2016
- Creating Graphs and Charts from a Data Model
- Publishing Excel Workbooks to Excel Online

## Topics Covered

Revisiting The Excel Data Model
Working with the Power Query Add-In

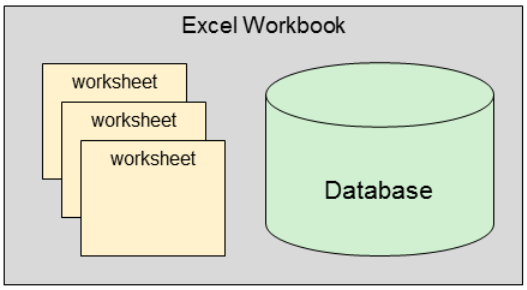
## Instructor Demos

- Working with Power Query in Excel 2016
- Working with Power Pivot in Excel 2016
- Creating Excel PivotTables from a Data Model
- Publishing an Excel workbook with a Data Model
- Working with Power Pivot in Excel 2016

© Critical Path Training. 2015. All Rights Reserved

# Revisiting The Excel Data Model

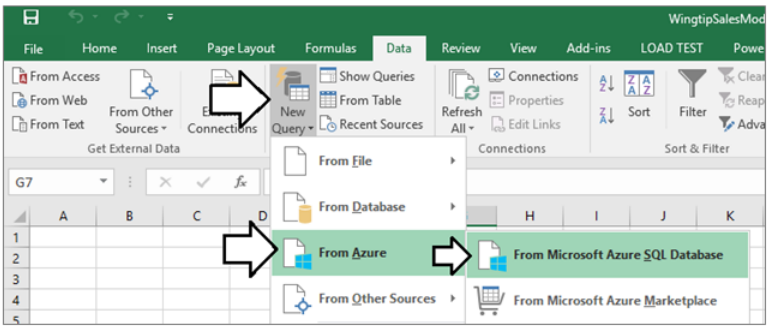
- Every Excel Workbook can have a data model
  - Provides a foundation for BI project solution
- When importing data with Power Query…
  - You can add the data as a table in a worksheet
  - You can add the data as a table in the data model




© Critical Path Training. 2015. All Rights Reserved

# Working with the Power Query Add-In

- Power Query Add-In adds a tab to Excel ribbon
  - Click **Manage** button display Power Query window
  - You must switch back and forth between Excel application window




© Critical Path Training. 2015. All Rights Reserved

# Module 08 Lab: Leveraging the BI Features in Excel 2016

**Lab Time**: 60 minutes

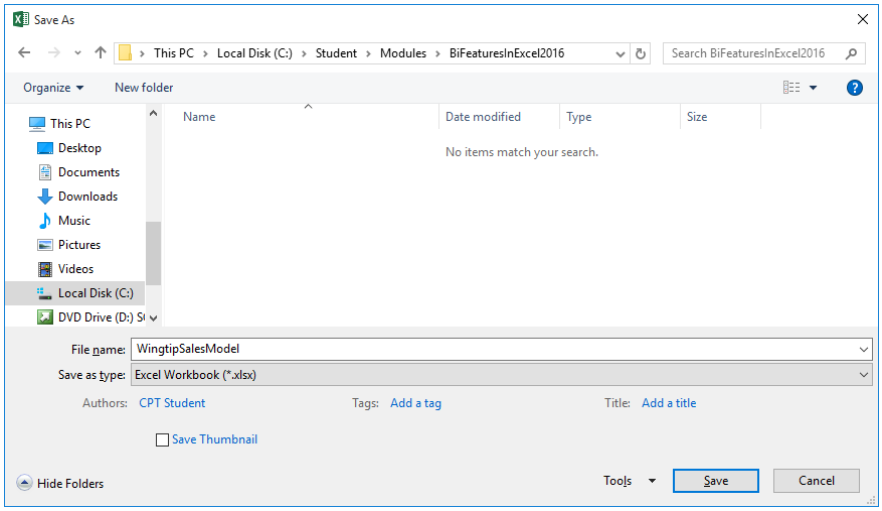
**Lab Folder**: C:\Student\Modules\BiFeaturesInExcel2016

**Lab Overview**: In this lab, you will create a new Excel workbook and use the Power Pivot features to create a new data model that is identical to the data model in the **WingtipSalesAnalysis.pbix** project that you created with Power BI Desktop. As you go through this lab, you will notice that using the Power Query and Power Pivot features in Excel 2016 is a bit different from using the same features in Power BI Desktop. However, your basic knowledge of how Power Query and Power Pivot work should allow you to work much faster.

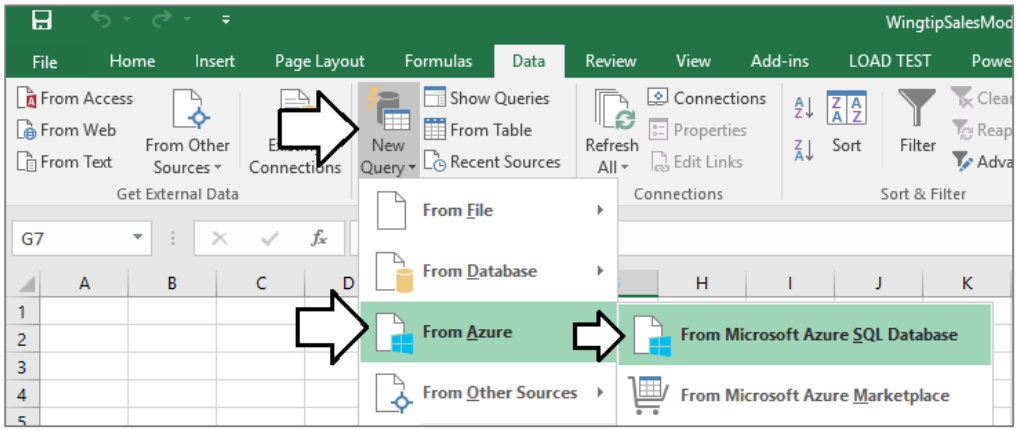
## Exercise 1: Creating a New Excel Workbook and Importing Data into the Data Model

In this exercise you will launch Microsoft Excel 2013 and create a new workbook file. After that, you will use the Power Query features in Excel 2016 to import data from the Wingtip Sales database into the workbook's data model.

1. Launch Microsoft Excel 2016.
2. Create a new empty workbook.
3. Save the workbook file as WingtipSalesModel.xlsx to the path of C:\Student\Modules\BiFeaturesInExcel2016.




4. Import Wingtip sales data using Power Query.
   a) Activate the **Data** tab on the ribbon.
   b) Drop down the New Query menu and select From Azure > From Microsoft Azure SQL Database.

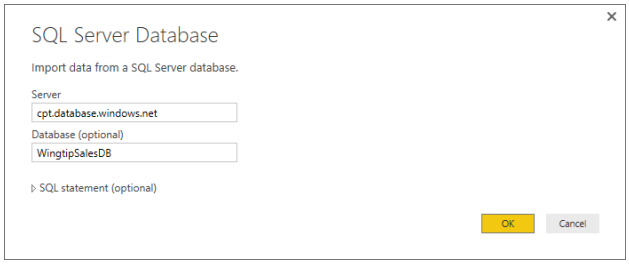



5. When you are prompted with the **SQL Server Database** dialog, enter the following values for the **Server** and **Database**.
   a) Server: cpt.database.windows.net

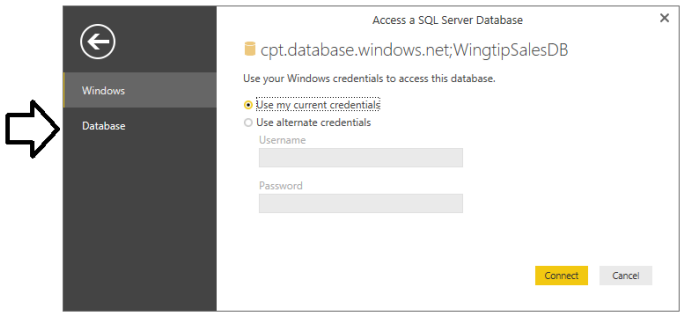
© Critical Path Training. 2015. All Rights Reserved

b) Database: WingtipSalesDB

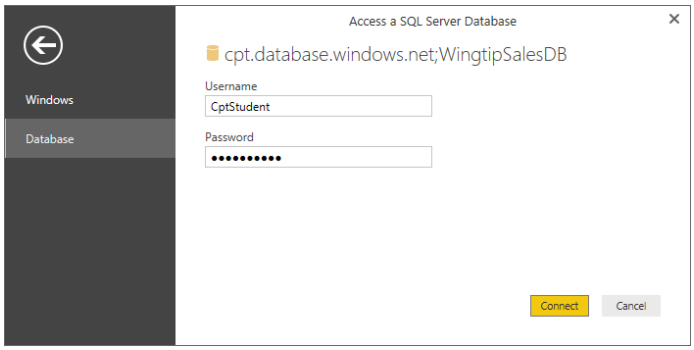
6. When the **SQL Server Database** dialog appears as it does in the following screenshot, click the **OK** button to continue.




7. At this point, you will be prompted by the **Access a SQL Server Database** dialog. Click on **Database** on the left side of the dialog so that you can enter the credentials for a standard SQL account instead of using Windows authentication.

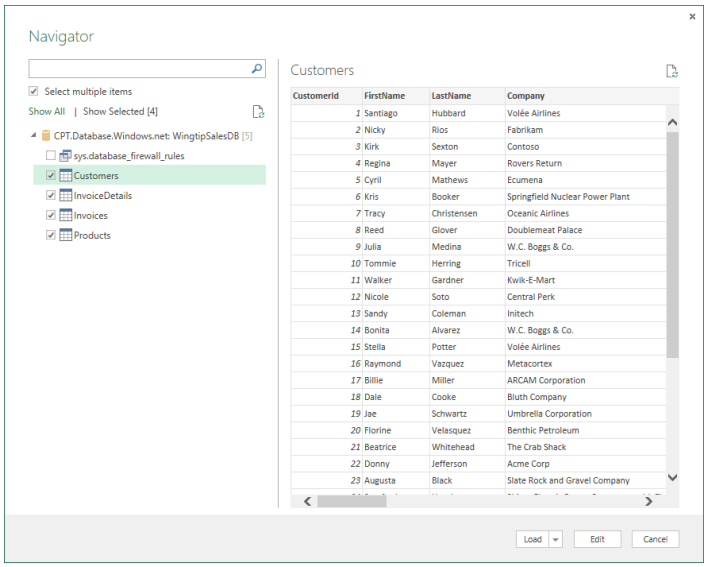



8. Enter the following credentials for a SQL user account that has been configured with read access to the database.
   a) Username: **CptStudent**
   b) Password: **pass@word1**
9. Once you have entered the credentials the **Access a SQL Server Database** dialog, click the **Connect** button to continue.

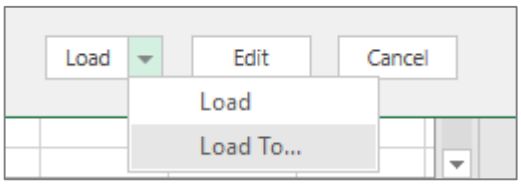



10. At this point, Excel should be able to establish a connection to the database and then prompt you with the **Navigator** dialog. The **Navigator** dialog allows you to select the tables you would like to import into your Excel workbook..
11. In the **Navigator** dialog, select the **Customers** table, the **InvoiceDetails** table, the **Invoices** table and the **Products** table as shown in the following screenshot.

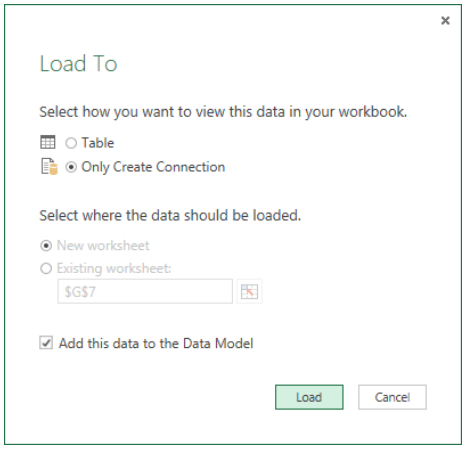
© Critical Path Training. 2015. All Rights Reserved




12. Look in the bottom right corner of the **Navigator** dialog and locate the **Load** drop down menu.
13. Drop down the **Load** menu and select the **Load To** command to display the **Load To** dialog.

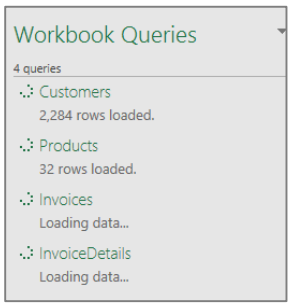



14. In the **Load To** dialog, select Only Create Connection and make sure the checkbox for the **Add this data to the Data Model** option is selected.

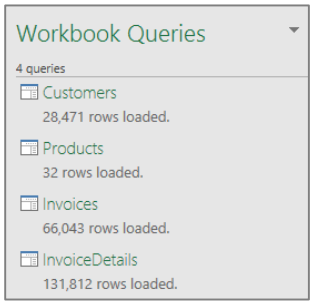



15. At this point, Excel will execute the four queries and load the data into the data model in the workbook.

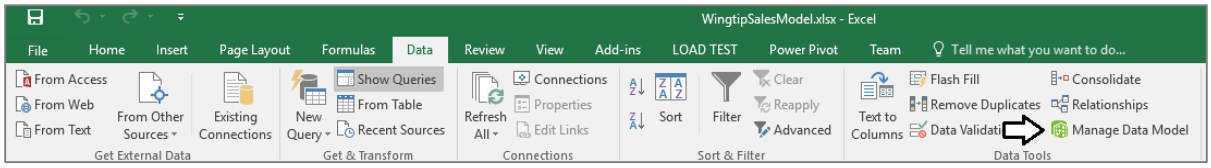
© Critical Path Training. 2015. All Rights Reserved



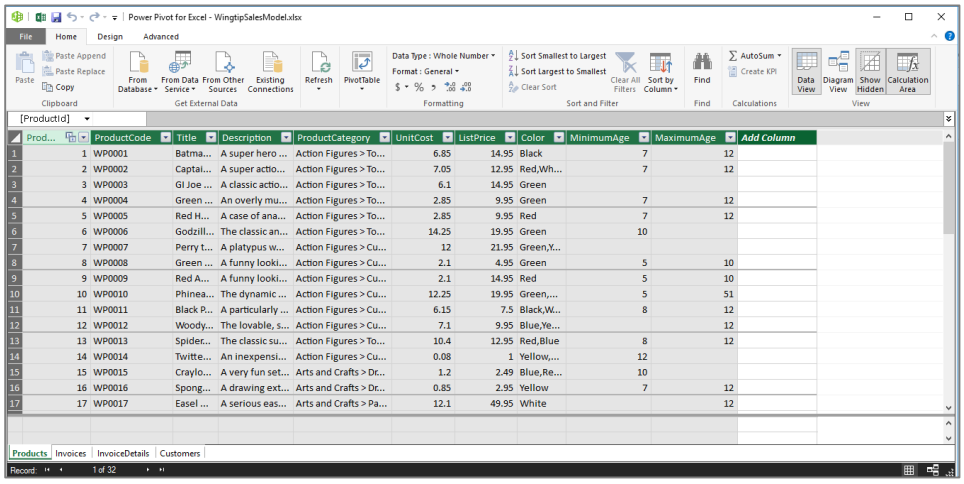

16. Once the queries have loaded their data into the data model, you can see how many rows were added to each table.




17. Inspect the tables that were added to the data model.
    a) Activate the **Data** tab in the ribbon.
    b) Click on the **Manage Data Model** button at the far right-hand side of the ribbon to display the **Power Pivot for Excel** window.



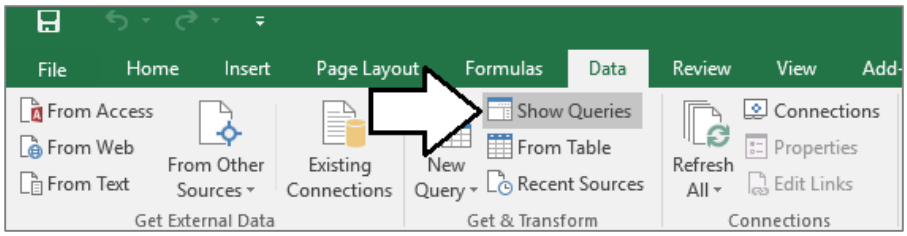

    c) When the Power Pivot windows opens, you should be able to see the four tables inside it. You can switch back and forth between the table in the data model by using the tabs at the bottom, left-hand corner of the page.



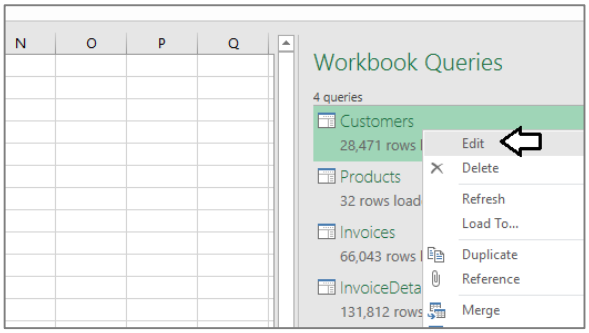

    d) Now, that you have seen the Power Pivot window, close it for now because you will need to spend more time modifying your queries and transform the imported data with Power Query before you begin to use Power Pivot to model the data .

© Critical Path Training. 2015. All Rights Reserved

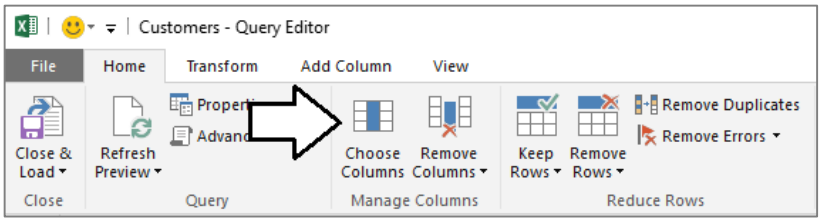
18. Make sure the **Workbook Queries** pane is showing on the right-hand side of the Excel application window.
    a) If the **Workbook Queries** pane is not showing, you can toggle it between displaying and hiding by clicking the **Show Queries** button in the **Data** tab of the ribbon.




19. Right-click on the **Customers** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.

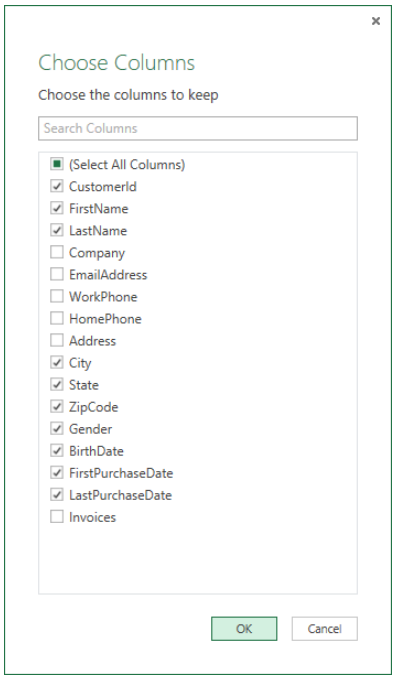



20. In the Customers – Query Editor window, click the Choose Columns button.

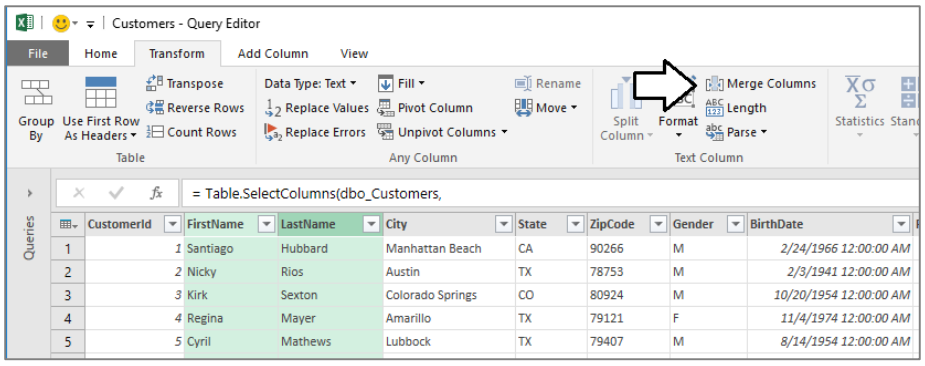



21. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all column. Next, select the checkboxes for **CustomerId**, **FirstName**, **LastName**, **City**, **State**, **ZipCode**, **Gender**, **BirthDate**, **FirstPurchaseDate** and **LastPurchaseDate** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.

© Critical Path Training. 2015. All Rights Reserved

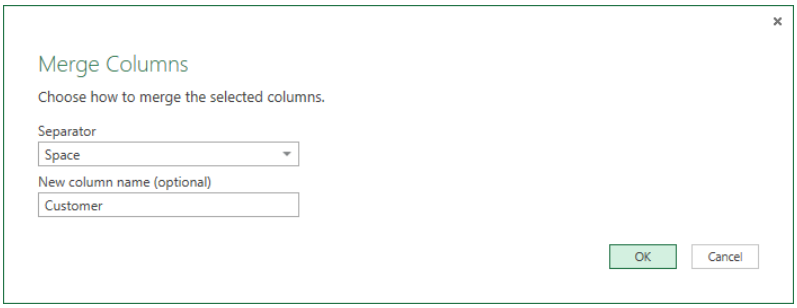



22. You should be able to see that the Query Editor window now only shows the columns that you selected.
23. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
    a) Begin by clicking on the **Transform** tab in the ribbon.
    b) Next, select the **FirstName** column by clicking on its column header.
    c) Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.
    d) With both the **FirstName** column and the **LastName** column selected, click the **Merge Column** button in the ribbon to display the **Merge Column** dialog.




    e) In the **Merge Column** dialog, drop down the **Separator** control and select a value of **Space**. Add a **New column name** value of **Customer** and click the **OK** button to modify the underlying query with your changes.

© Critical Path Training. 2015. All Rights Reserved



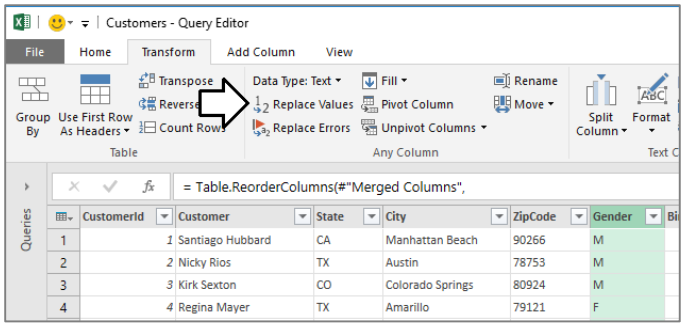

f) You should now be able to see that the **FirstName** column and the **LastName** column have been replaced with a single merged column named **Customer**.

| | CustomerId | Customer | City |
|---|---|---|---|
| 1 | 1 | Santiago Hubbard | Manhattan Beach |
| 2 | 2 | Nicky Rios | Austin |
| 3 | 3 | Kirk Sexton | Colorado Springs |
| 4 | 4 | Regina Mayer | Amarillo |
| 5 | 5 | Cyril Mathews | Lubbock |
| 6 | 6 | Kris Booker | Tempe |
| 7 | 7 | Tracy Christensen | El Paso |

24. The **State** column is currently to the right of the **City** column. Move the **State** column so that it is repositioned to the left of the **City** column. Accomplish this by clicking on the column header for the **State** column and leaving the mouse button down. Drag the **State** column to the left of the **City** column and release the mouse button.

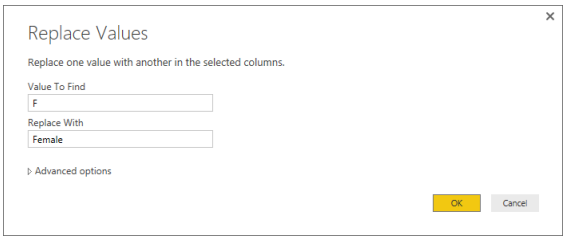
| | CustomerId | Customer | State | City | ZipCode |
|---|---|---|---|---|---|
| 1 | 1 | Santiago Hubbard | CA | Manhattan Beach | 90266 |
| 2 | 2 | Nicky Rios | TX | Austin | 78753 |
| 3 | 3 | Kirk Sexton | CO | Colorado Springs | 80924 |
| 4 | 4 | Regina Mayer | TX | Amarillo | 79121 |

25. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
   a) Make sure the **Transform** tab is the active tab in the ribbon.
   b) Select the **Gender** column by clicking its column header.
   c) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.

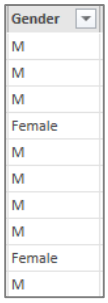



   d) In the **Replace Value** dialog, enter a value of **F** in the **Value to Find** textbox and enter a value of **Female** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.

© Critical Path Training. 2015. All Rights Reserved



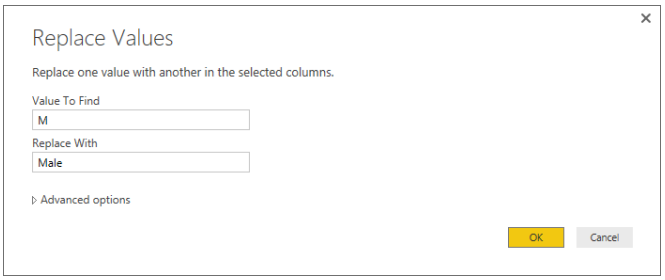

e) You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.



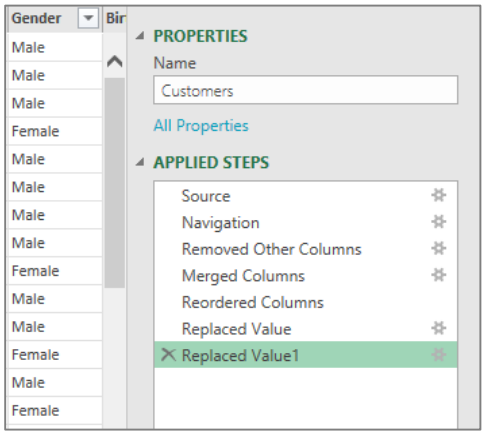

f) Make sure the **Gender** column is still selected.

g) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.

h) In the **Replace Value** dialog, enter a value of **M** in the **Value to Find** textbox and enter a value of **Male** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.

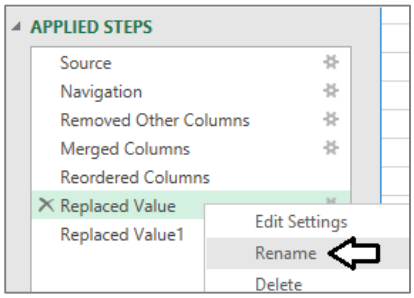



i) You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**. If you inspect the **APPLIED STEPS** list in the **Query Settings** pane, you should be able to see that there are two steps at the end that have been given the generic names of **Replaced Value** and **Replaced Value 1**.

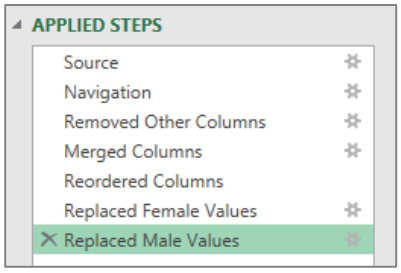



© Critical Path Training. 2015. All Rights Reserved

j) In order to promote higher levels of maintainability, it's often a good idea to rename steps with names such as of **Replaced Value** and **Replaced Value 1**. Rename the **Replaced Values** step by right-clicking it and clicking the **Rename** command to place the step name in edit mode. Modify the name of this step to **Replaced Female Values**.




k) Using the same technique, rename the **Replaced Value 1** step to **Replaced Male Values**.

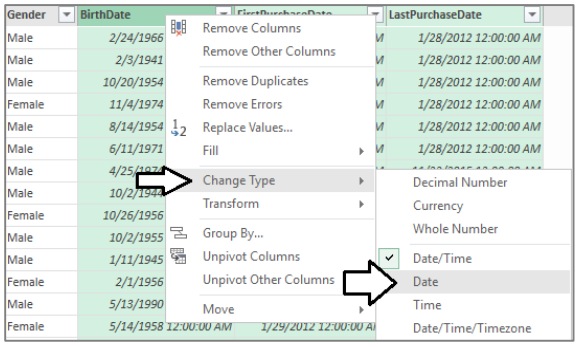



26. Change the column type of BirthDate, FirstPurchasedDate and LastPurchasedDate from Date/Time to Date.

a) Select the **BirthDate** column by clicking its column header.

b) Hold down the **SHIFT** key and click the column headers for **FirstPurchasedDate** and **LastPurchasedDate** so that all three columns are selected.

c) Right-click on any one of the selected columns and select the **Change Type > Date** command.

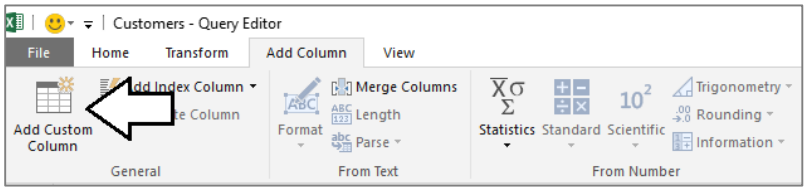



d) You should see that the three columns now show values with a date but without a time.

© Critical Path Training. 2015. All Rights Reserved

| BirthDate | FirstPurchaseDate | LastPurchaseDate |
|---|---|---|
| 2/24/1966 | 1/28/2012 | 1/28/201 |
| 2/3/1941 | 1/28/2012 | 1/28/201 |
| 10/20/1954 | 1/28/2012 | 1/28/201 |
| 11/4/1974 | 1/28/2012 | 1/28/201 |
| 8/14/1954 | 1/28/2012 | 1/28/201 |
| 6/11/1971 | 1/28/2012 | 1/28/201 |
| 4/25/1974 | 1/29/2012 | 11/22/201 |
| 10/2/1944 | 1/29/2012 | 10/2/201 |
| 10/26/1956 | 1/29/2012 | 1/29/201 |
| 10/2/1955 | 1/29/2012 | 5/6/201 |

27. Add a new custom column named **Customer Type** to indicates whether the customer is a repeat customer or not. You will accomplish by writing a DAX a formula which compares the **FirstPurchaseDate** column to **LastPurchaseDate** column.
    a) Begin by activating the **Add Column** tab in the ribbon.
    b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.



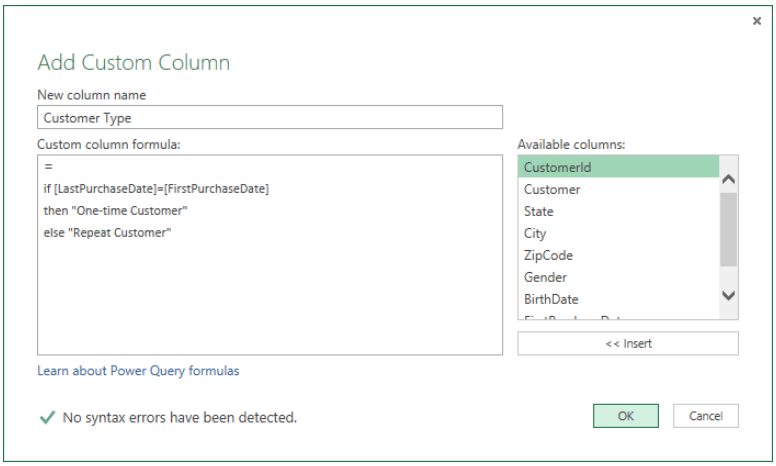

    c) In the **Add Custom Column** dialog, add a value of **Customer Type** in the **New column name** textbox.

In this particular scenario, you are working under the assumption that the customer is a repeat customer when the **FirstPurchaseDate** column and the **LastPurchaseDate** column are not equal indicating the customer has made two or more purchases.

    d) In the **Custom column formula** textbox, enter the following formula.

```
if [LastPurchaseDate]=[FirstPurchaseDate]
then "One-time Customer"
else "Repeat Customer"
```

    e) When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.




    f) You should be able to verify that the new **Customer Type** column has a value of **Repeat Customer** when the current customer has a **FirstPurchaseDate** column value that is not equal to the **LastPurchaseDate** column value. When these column values are equal, the **CustomerType** column has a value of **One-time Customer**.

© Critical Path Training. 2015. All Rights Reserved

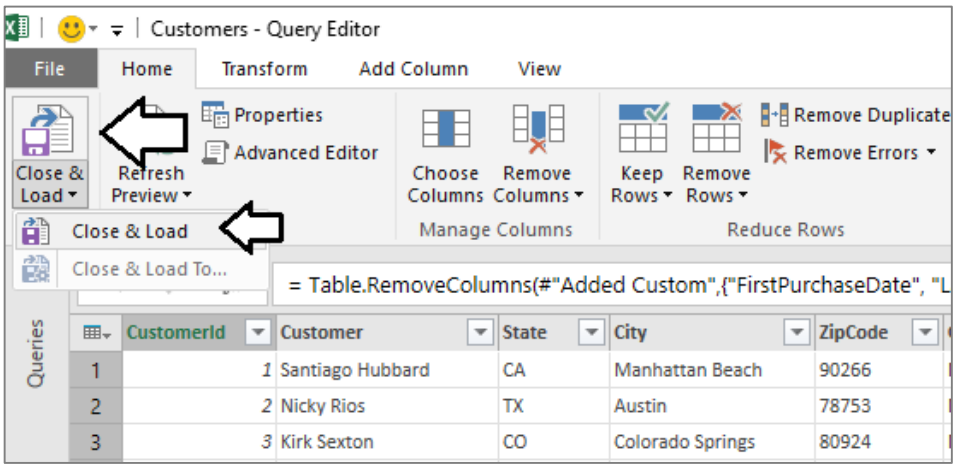
| FirstPurchaseDate | LastPurchaseDate | Customer Type |
|---|---|---|
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/28/2012 | 1/28/2012 | One-time Customer |
| 1/29/2012 | 11/22/2015 | Repeat Customer |
| 1/29/2012 | 10/2/2015 | Repeat Customer |
| 1/29/2012 | 1/29/2012 | One-time Customer |

28. Now, that you have used the **FirstPurchaseDate** column and the **LastPurchaseDate** column to calculate the value of the **CustomerType** column, you can delete them because they are no longer needed.
    a) Activate the **Home** tab on the ribbon.
    b) Select the **FirstPurchaseDate** column by clicking its column header.
    c) Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
    d) Click the **Remove Columns** button in the ribbon to remove both columns from the query results.
    e) You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **CustomerType** column is still there.

| | CustomerId | Customer | State | City | ZipCode | Gender | BirthDate | Customer Type |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Santiago Hubbard | CA | Manhattan Beach | 90266 | Male | 2/24/1966 | One-time Customer |
| 2 | 2 | Nicky Rios | TX | Austin | 78753 | Male | 2/3/1941 | One-time Customer |
| 3 | 3 | Kirk Sexton | CO | Colorado Springs | 80924 | Male | 10/20/1954 | One-time Customer |
| 4 | 4 | Regina Mayer | TX | Amarillo | 79121 | Female | 11/4/1974 | One-time Customer |
| 5 | 5 | Cyril Mathews | TX | Lubbock | 79407 | Male | 8/14/1954 | One-time Customer |
| 6 | 6 | Kris Booker | AZ | Tempe | 85281 | Male | 6/11/1971 | One-time Customer |
| 7 | 7 | Tracy Christensen | TX | El Paso | 79936 | Male | 4/25/1974 | Repeat Customer |
| 8 | 8 | Reed Glover | TX | Austin | 78730 | Male | 10/2/1944 | Repeat Customer |

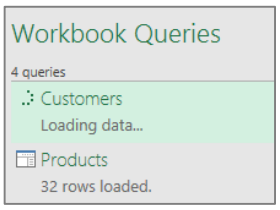
You are now done working with the **Customers** query.

29. Click the **Close & Load** button on the **Home** tab of the ribbon to execute the updated **Customers** query.




30. Excel will display an animated gif next to the **Customers** query while the query is executing.

© Critical Path Training. 2015. All Rights Reserved




31. After the query changes have been applied, you should be able to see the results of your changes in the Customers table that has been loaded into the project's data model.
32. Save the work you have done to **WingtipSalesModel.xlsx** by clicking the Save button in the upper left corner of the Excel window.
33. Right-click on the **Products** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.
34. Remove the columns that are not required in the **Products** query results.
    a) Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
    b) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **ProductId**, **Title**, **ProductCategory**, **UnitCost** and **ListPrice**. Once you have these columns selected, click the **OK** button to close the dialog and to modify the underlying query.
35. Rename the **Title** column to **Product**.
    a) Right-click on the **Title** column and click **Rename**.
    b) Modify the column name to **Product**.
36. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
    a) Activate the **Transform** tab on the ribbon.
    b) Select the **ProductCategory** by clicking its column header.
    c) Drop down the **Split Column** menu button and click by **By Delimiter** to display the **Split Column By Column** dialog.
    d) In the Split Column By Column dialog, drop down the Select or enter delimiter combo box and select --Custom--.
    e) In the textbox below the combo box, enter a three-character text value which includes a space follow by the **>** character followed by another space.
    f) When the **Split Column By Column** dialog appears as the one in the following screenshot, click the **OK** button to add the step to the **Products** query.
    g) You should be able to confirm that Excel has split the **ProductCategory** column into two separate columns named **ProductCategory.1** and **ProductCategory.2**.

| ProductId | Product | ProductCategory.1 | ProductCategory.2 | UnitCost | ListPrice |
|---|---|---|---|---|---|
| 1 | Batman Action Figure | Action Figures | Tough Guys | 6.85 | 14.95 |
| 2 | Captain America Action Figure | Action Figures | Tough Guys | 7.05 | 12.95 |
| 3 | GI Joe Action Figure | Action Figures | Tough Guys | 6.1 | 14.95 |
| 4 | Green Hulk Action Figure | Action Figures | Tough Guys | 2.85 | 9.95 |
| 5 | Red Hulk Alter Ego Action Figure | Action Figures | Tough Guys | 2.85 | 9.95 |
| 6 | Godzilla Action Figure | Action Figures | Tough Guys | 14.25 | 19.95 |

    h) Rename the **ProductCategory.1** column to **Category.** Also rename the **ProductCategory.2** column to **Subcategory** as shown in the following screenshot.

| | ProductId | Product | Category | Subcategory |
|---|---|---|---|---|
| 1 | 1 | Batman Action Figure | Action Figures | Tough Guys |
| 2 | 2 | Captain America Action Figure | Action Figures | Tough Guys |
| 3 | 3 | GI Joe Action Figure | Action Figures | Tough Guys |
| 4 | 4 | Green Hulk Action Figure | Action Figures | Tough Guys |
| 5 | 5 | Red Hulk Alter Ego Action Figure | Action Figures | Tough Guys |
| 6 | 6 | Godzilla Action Figure | Action Figures | Tough Guys |
| 7 | 7 | Perry the Platypus Action Figure | Action Figures | Cute and Huggable |
| 8 | 8 | Green Angry Bird Action Figure | Action Figures | Cute and Huggable |

When you have query columns based on numeric currency values, it is best to change their column type to **Fixed Decimal Number**.

© Critical Path Training. 2015. All Rights Reserved

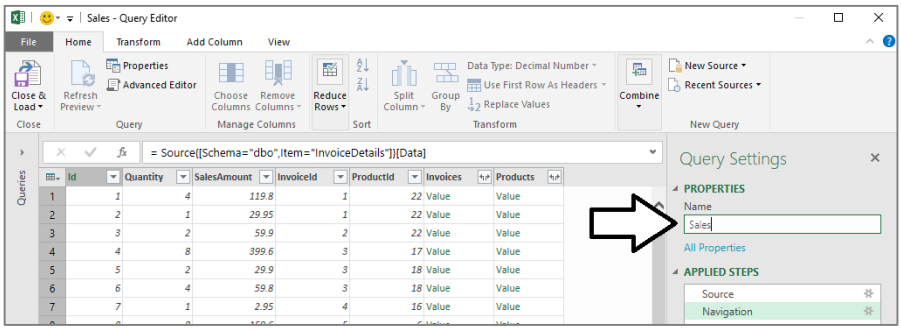
37. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Currency** type.
    a) Select the **UnitCost** column by clicking its column header.
    b) Hold down the **SHIFT** key and click the ListPrice column so that both columns are selected.
    c) Right-click either one of the selected columns and click the **Change Type > Currency** menu command.

You are now done working with the **Products** query.

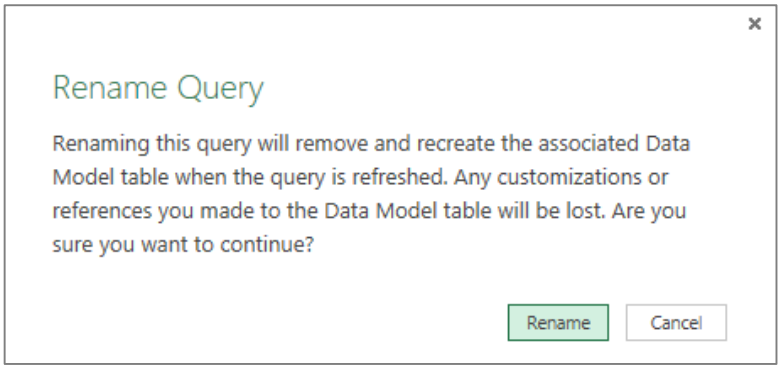
38. Click the **Close & Load** button on the **Home** tab of the ribbon to execute the updated **Products** query.
39. Save your work by clicking the Save button in the upper left corner of the Excel window.
40. Right-click on the **InvoiceDetails** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.

One important point of flexibility in the import process is that you can change the name of a query, and therefore the name of the resulting table to make the data model more intuitive and easier to understand. The **InvoiceDetails** query is returning data that will be used to calculate sales results at the most granular level. Therefore, the data model will be easier to understand if you change the name the name of the **InvoiceDetails** query to **Sales**. The will result in renaming the resulting table to the **Sales** table.

41. Update the name of the **InvoiceDetails** query to **Sales** by replacing the text in the **Name** textbox in the **Query Settings** pane.




42. When prompted to rename the query, read the warning and then click the **Rename** button.

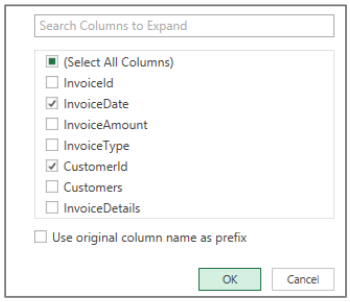



43. Using the mouse, drag and drop the column header for the **InvoiceId** column to move it to the immediate right of the **Id** column.
44. Using the mouse, drag and drop the column header for the **ProductId** column to move it to the right of the **InvoiceId** column.

© Critical Path Training. 2015. All Rights Reserved

| Id | InvoiceId | ProductId | Quantity | SalesAmount | Invoices | Products |
|---|---|---|---|---|---|---|
| 1 | 1 | 22 | 4 | 119.8 | Value | Value |
| 2 | 1 | 22 | 1 | 29.95 | Value | Value |
| 3 | 2 | 22 | 2 | 59.9 | Value | Value |
| 4 | 3 | 17 | 8 | 399.6 | Value | Value |
| 5 | 3 | 18 | 2 | 29.9 | Value | Value |
| 6 | 3 | 18 | 4 | 59.8 | Value | Value |
| 7 | 4 | 16 | 1 | 2.95 | Value | Value |

45. Modify the column type of the **SalesAmount** column to the **Currency** type.
    a) Select the **SalesAmoungt** column by clicking its column header.
    b) Right-click the selected column and click the **Change Type > Currency** menu command.
46. Expand the **Invoices** column to add the **InvoiceDate** column and the **CustomerId** column to the **Sales** query.
    a) Click the Expand button inside the column header of the **Invoices** column to display the **Columns to Expand** dialog.
    b) In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for the **InvoiceDate** column and the **CustomerId** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns to Expand** dialog looks like the one shown in the following screenshot, click the **OK** button to close the dialog and to modify the underlying query.




    c) You should see that the **InvoiceDate** column and the **CustomerId** column have now been added to the **Sales** query results.
47. Use the mouse to drag and drop the **CustomerId** column to move it to the right of the **InvoiceId** column.
48. Use the mouse to drag and drop the **InvoiceDate** column to move it to the right of the **ProductId** column.
49. Change the column type of the **InvoiceDate** to the **Date** type by right-clicking its column header and selecting the **Change Type > Date** command. When you are done, the data in the column should show a date value without a time.
50. Change the name of the **InvoiceDate** column to **PurchaseDate**.
51. Expand the **Products** column to add the **UnitCost** column to the **Sales** query.
    a) Click the Expand button inside the column header of the **Products** column to display the **Columns to Expand** dialog.
    b) In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkbox for the **UnitCost** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns to Expand** dialog looks like the one shown in the following screenshot, click the **OK** button to close the dialog and to modify the underlying query.
    c) You should see that the **UnitCost** column has now been added to the **Sales** query results.

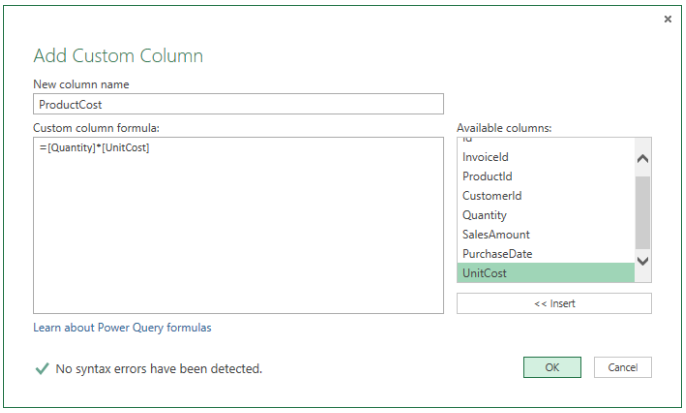
© Critical Path Training. 2015. All Rights Reserved

| | Id | InvoiceId | ProductId | CustomerId | Quantity | SalesAmount | PurchaseDate | UnitCost |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 22 | 1 | 4 | 119.8 | 1/28/2012 | 21.5 |
| 2 | 2 | 1 | 22 | 1 | 1 | 29.95 | 1/28/2012 | 21.5 |
| 3 | 3 | 2 | 22 | 2 | 2 | 59.9 | 1/28/2012 | 21.5 |
| 4 | 22 | 14 | 22 | 14 | 1 | 29.95 | 1/29/2012 | 21.5 |
| 5 | 10 | 6 | 1 | 6 | 4 | 59.8 | 1/28/2012 | 6.85 |
| 6 | 18 | 12 | 1 | 12 | 3 | 44.85 | 1/29/2012 | 6.85 |
| 7 | 27 | 17 | 1 | 17 | 4 | 59.8 | 1/29/2012 | 6.85 |
| 8 | 11 | 7 | 3 | 7 | 4 | 59.8 | 1/29/2012 | 6.1 |

52. Add a new custom column named **ProductCost** to calculate the product of the **Quantity** field multiplied by the **UnitCost** field.
    a) Begin by activating the **Add Column** tab in the ribbon.
    b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.
    c) In the **Add Custom Column** dialog, add a value of **ProductCost** in the **New column name** textbox.
    d) In the **Custom column formula** textbox, enter the following formula.

```
[Quantity] * [UnitCost]
```

    e) When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.




    f) You should be able to verify that the new **ProductCost** column has a value calculated by multiplying the value of the **Quantity** column together with the value of the **UnitCost** column.

Once the **UnitCost** column has been used to calculate the value for the **ProductCost** value, this column is no longer needed and can be removed from the results of the **Sales** query.

53. Remove the **UnitCost** column by selecting its column header and pressing the DELETE key on the keyboard.
54. Modify the column type of the **SalesAmount** column and the **ProductCost** column to the **Currency** type.
    a) Select the **SalesAmount** column by clicking its column header.
    b) Hold down the **SHIFT** key and click the **ProductCost** column so that both columns are selected.
    c) Right-click either one of the selected columns and click the **Change Type > Currency** menu command.
55. You are now done working with the **Sales** query and its output should match the following screenshot.

© Critical Path Training. 2015. All Rights Reserved

| | Id | InvoiceId | ProductId | CustomerId | Quantity | SalesAmount | PurchaseDate | ProductCost |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 22 | 1 | 4 | 119.8 | 1/28/2012 | 86 |
| 2 | 2 | 1 | 22 | 1 | 1 | 29.95 | 1/28/2012 | 21.5 |
| 3 | 3 | 2 | 22 | 2 | 2 | 59.9 | 1/28/2012 | 43 |
| 4 | 22 | 14 | 22 | 14 | 1 | 29.95 | 1/29/2012 | 21.5 |
| 5 | 10 | 6 | 1 | 6 | 4 | 59.8 | 1/28/2012 | 27.4 |
| 6 | 18 | 12 | 1 | 12 | 3 | 44.85 | 1/29/2012 | 20.55 |

56. Click the **Close & Load** button on the **Home** tab of the ribbon to execute the updated **Sales** query. Excel will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
57. After the query changes have been applied, you should be able to see the results of your changes in the **Sales** table that has been loaded into the project's data model.
58. Save your work by clicking the Save button in the upper left corner of the Excel window.
59. Right-click on the **Invoices** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.

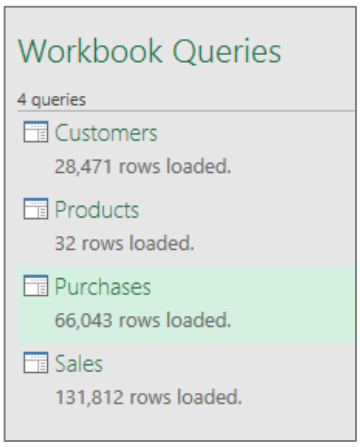
In an earlier step you changed the name of the **InvoiceDetails** query to **Sales** to make the data model easier to understand. In this exercise you will change the name of the **Invoices** query to **Purchases** for the same reason.

60. Update the name of the **Invoices** query to **Purchases** by replacing the text in the **Name** textbox in the **Query Settings** pane.
61. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
62. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **InvoiceId** and **InvoiceType** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.
63. You should be able to see that the Query Editor window now only shows the columns that you selected.
64. Modify the query so that the **InvoiceType** column returns values that are more human readable.
    a) Make sure the **Transform** tab is the active tab in the ribbon.
    b) Select the **InvoiceType** column by clicking its column header.
    c) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
    d) In the **Replace Value** dialog, enter a value of **InPerson** in the **Value to Find** textbox and enter a value of **Store Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.
    e) Make sure the **InvoiceType** column is still selected.
    f) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
    g) In the **Replace Value** dialog, enter a value of **MailOrder** in the **Value to Find** textbox and enter a value of **Mail Order Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.
    h) Make sure the **InvoiceType** column is still selected.
    i) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
    j) In the **Replace Value** dialog, enter a value of **Online** in the **Value to Find** textbox and enter a value of **Online Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.
    k) If you scroll down and look at all the rows within the **Purchases** table, you should be able to see that each row has a **InvoiceType** column value of either **Store Purchase**, **Mail Order Purchase** or **Online Purchase**.
65. Change the name of the **InvoiceType** column to **Purchase Type**.

| | InvoiceId | Purchase Type |
|---|---|---|
| 1 | 1 | Store Purchase |
| 2 | 2 | Store Purchase |
| 3 | 3 | Store Purchase |
| 4 | 4 | Store Purchase |
| 5 | 5 | Store Purchase |
| 6 | 6 | Store Purchase |

66. You are now done working with the **Purchase** query.

© Critical Path Training. 2015. All Rights Reserved



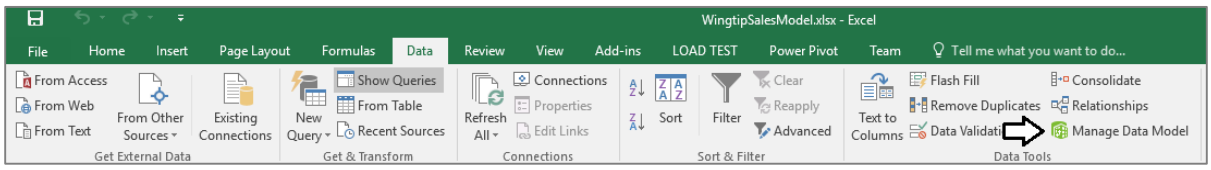

67. Save your work by clicking the Save button in the upper left corner of the Excel window.

You have now finished your work with Power Query and ready to begin working with Power Pivot.

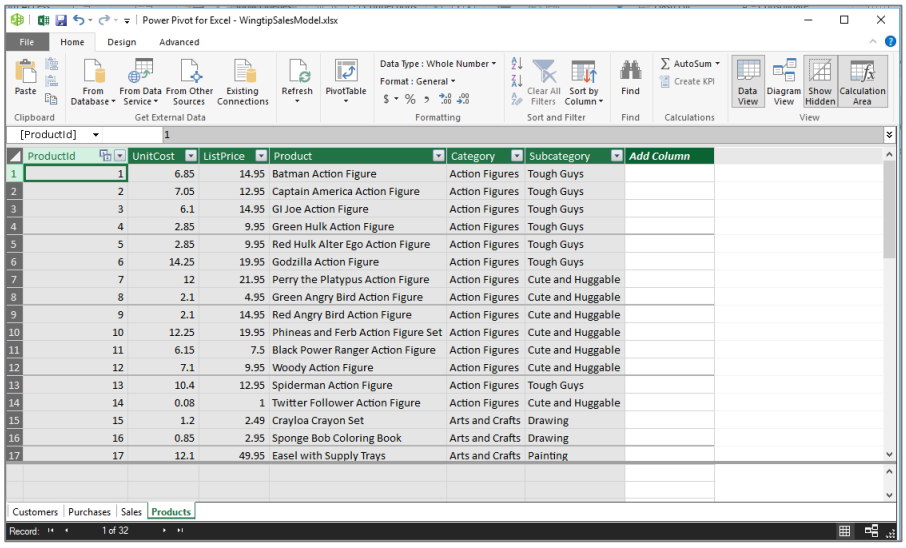
## Exercise 2: Using the Power Pivot Features of Excel 2016

In this exercise you will model the data you just imported using Power Pivot.

1. Open the Power Pivot windows to view and edit the data model.
   a) Activate the **Data** tab in the ribbon.
   b) Click on the **Manage Data Model** button at the far right-hand side of the ribbon to display the **Power Pivot for Excel** window.




   c) Examine the tables and verify that they have the Power Query transformations you applied in the previous exercise.

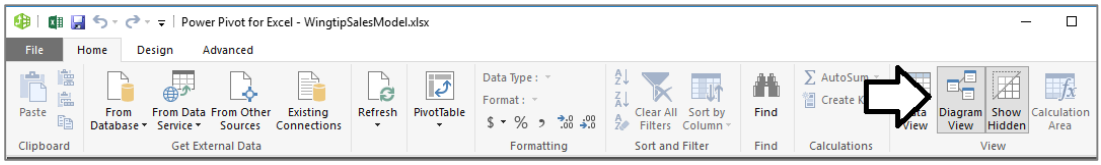



| ProductId | UnitCost | ListPrice | Product | Category | Subcategory | Add Column |
|---|---|---|---|---|---|---|
| 1 | 6.85 | 14.95 | Batman Action Figure | Action Figures | Tough Guys | |
| 2 | 7.05 | 12.95 | Captain America Action Figure | Action Figures | Tough Guys | |
| 3 | 6.1 | 14.95 | GI Joe Action Figure | Action Figures | Tough Guys | |
| 4 | 2.85 | 9.95 | Green Hulk Action Figure | Action Figures | Tough Guys | |
| 5 | 2.85 | 9.95 | Red Hulk Alter Ego Action Figure | Action Figures | Tough Guys | |
| 6 | 14.25 | 19.95 | Godzilla Action Figure | Action Figures | Tough Guys | |
| 7 | 12 | 21.95 | Perry the Platypus Action Figure | Action Figures | Cute and Huggable | |
| 8 | 2.1 | 4.95 | Green Angry Bird Action Figure | Action Figures | Cute and Huggable | |
| 9 | 2.1 | 14.95 | Red Angry Bird Action Figure | Action Figures | Cute and Huggable | |
| 10 | 12.25 | 19.95 | Phineas and Ferb Action Figure Set | Action Figures | Cute and Huggable | |
| 11 | 6.15 | 7.5 | Black Power Ranger Action Figure | Action Figures | Cute and Huggable | |
| 12 | 7.1 | 9.95 | Woody Action Figure | Action Figures | Cute and Huggable | |
| 13 | 10.4 | 12.95 | Spiderman Action Figure | Action Figures | Tough Guys | |
| 14 | 0.08 | 1 | Twitter Follower Action Figure | Action Figures | Cute and Huggable | |
| 15 | 1.2 | 2.49 | Crayloa Crayon Set | Arts and Crafts | Drawing | |
| 16 | 0.85 | 2.95 | Sponge Bob Coloring Book | Arts and Crafts | Drawing | |
| 17 | 12.1 | 49.95 | Easel with Supply Trays | Arts and Crafts | Painting | |

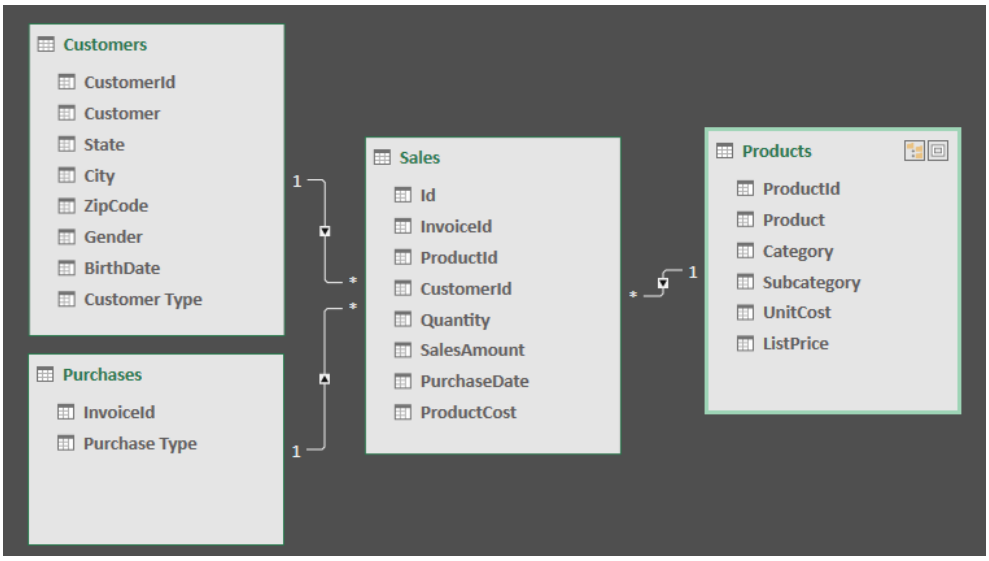
2. Modify the relations in the data model.

© Critical Path Training. 2015. All Rights Reserved

a) Navigate to diagram view to see the relationships between the tables.




b) Reorganize the tables so the view looks cleaner.




3. Modify the formatting of the **BirthDate** column in the **Customers** table.
   a) In the Power BI Desktop windows, navigate back to data view.
   b) In the **Fields** list on the right, select the **Customers** table to display its rows and columns.
   c) Select the **BirthDate** column by clicking on its column header.
   d) Modify the formatting of the **BirthDate** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Date Time > 3/14/2001 (M/d/yyyy)**.
   e) The **BirthDate** column should now reflect the change in formatting.
4. Modify the formatting of columns in the **Products** table.
   a) In the **Fields** list on the right, select the **Products** table to display its rows and columns.
   b) Select the **UnitCost** column by clicking on its column header.
   c) Use the **Format** menu button in the ribbon to update the format setting to **Currency > English (United States)**.
   d) Changing the format setting of the **ListPrice** column to **Currency > English (United States)** so it matches **UnitCost**.
5. Modify the formatting of columns in the **Sales** table.
   a) In the **Fields** list on the right, select the **Sales** table to display its rows and columns.
   b) Select the **Quantity** column by clicking on its column header.
   c) Modify the **Quantity** column by clicking to select the comma button on the ribbon to add a comma separator.
   d) Select the **SalesAmount** column by clicking on its column header.
   e) Modify the formatting of the **SalesAmount** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Currency > English (United States)**.
   f) Select the **PurchaseDate** column by clicking on its column header.
   g) Modify the formatting of the **Purchase Date** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Date Time > 3/14/2001 (M/d/yyyy)**.

© Critical Path Training. 2015. All Rights Reserved

h) Select the **ProductCost** column by clicking on its column header.

i) Modify the formatting of the **ProductCost** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Currency > English (United States)**.

6. Add a calculated column to the **Sales** table named **SalesProfit** to determine profit by calculating the difference between **SalesAmount** and **ProductCost**.

   a) Navigate to data view.

   b) Select the **Sales** table in the **Fields** list.

   c) Create a new calculated column by clicking the **New Column** button in the ribbon.

   d) Enter to following DAX expression into the formula bar to create the calculated column named **SalesProfit**.

```
SalesProfit = Sales[SalesAmount]-Sales[ProductCost]
```

   e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a **SalesProfit** value for each row in the **Sales** table.

   f) Configure the column's formatting by using the **Format** menu on the ribbon to select **Currency > English (United States)**.

7. Add a calculated column to the **Sales** table named **PurchaseYear** to indicate the calendar year of each purchase.

   a) Navigate to data view.

   b) Select the **Sales** table in the **Fields** list.

   c) Create a new calculated column by clicking the **New Column** button in the ribbon.

   d) Enter to following DAX expression into the formula bar to create the calculated column named **PurchaseYear**.

```
PurchaseYear = YEAR(Sales[PurchaseDate])
```

   e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a calendar year value (e.g. 2012) for each row in the **Sales** table

   f) Once the **PurchaseYear** column has been created, use the **Data Type** dropdown menu on the ribbon to change the column type from **Whole Number** to **Text**.

8. At this point, you are on your own to continue extending the data model for the workbook file to make it identical to the data model you created earlier with Power BI Desktop.

© Critical Path Training. 2015. All Rights Reserved