Terms of Use

© 2014 Microsoft Corporation. All rights reserved.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the 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. Complying with all applicable copyright laws is the responsibility of the user. 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 the express written permission of Microsoft Corporation.

For more information, see Microsoft Copyright Permissions at http://www.microsoft.com/permission

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The Microsoft company name and Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

This document reflects current views and assumptions as of the date of development and is subject to change. Actual and future results and trends may differ materially from any forward-looking statements. Microsoft assumes no responsibility for errors or omissions in the materials.

THIS DOCUMENT IS FOR INFORMATIONAL AND TRAINING PURPOSES ONLY AND IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.
## Contents

TERMS OF USE ......................................................................................................................... 2

CONTENTS ................................................................................................................................. 3

ABOUT THE AUTHOR .................................................................................................................. 4

DOCUMENT REVISIONS ............................................................................................................. 4

LAB OVERVIEW .......................................................................................................................... 5

EXERCISE 1: CREATING A POWER PIVOT DATA MODEL ............................................................ 7

Task 1 – Creating the Excel Workbook ......................................................................................... 7
Task 2 – Enabling the Power Pivot Add-In .................................................................................... 8
Task 3 – Loading Tables from a SQL Server Database ................................................................. 9
Task 4 – Loading a Table from an OData Data Service ............................................................... 12
Task 5 – Loading a Table from an Excel Workbook .................................................................. 13
Task 6 – Relating Tables ............................................................................................................. 15
Task 7 – Exploring the Power Pivot Data .................................................................................... 16
Task 8 – Marking the Date Table ................................................................................................ 17
Task 9 – Configuring Column Value Sorting ............................................................................. 17
Task 10 – Creating Calculated Columns .................................................................................... 18
Task 11 – Enriching the Model with Hierarchies ...................................................................... 20
Task 12 – Defining Calculated Fields ......................................................................................... 23
Task 13 – Defining a KPI ........................................................................................................... 26
Task 14 – Creating a PivotTable Report ..................................................................................... 28
Task 15 – Finishing Up .............................................................................................................. 30

SUMMARY .................................................................................................................................... 30
About the Author

This lab was designed and written by Peter Myers.

Peter Myers has worked with Microsoft database and development products since 1997. Today, he specializes in all Microsoft BI products and provides mentoring, technical training, and education content authoring for SQL Server, Office, and SharePoint. Peter has a broad business background supported by a bachelor’s degree in applied economics and accounting, and he extends this with solid experience backed by current MCSE and MCT certifications. He has been a SQL Server MVP since 2007.

Document Revisions

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Author</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>24-AUG-2014</td>
<td>Peter Myers</td>
<td>Initial release</td>
</tr>
</tbody>
</table>
Lab Overview

Introduction

Note: This lab is the first in a series of seven labs, which explore self-service BI with Excel 2013 and Office 365 Power BI. If you plan to complete all of the labs, we recommend that you complete them in the order in which they were designed, although the labs can be completed in any order you choose.

In this lab, you will create a Power Pivot data model that imports data from a SQL Server database, an OData data service, and an Excel workbook. You will then enhance the data model by defining relationships, calculated columns and hierarchies. You will hide columns not required in the data model interface, and then you will define calculated fields and a key performance indicator (KPI). Finally, you will produce a PivotTable report to monitor regional product profitability.

The final report will look like the following.

Figure 1
Previewing the PivotTable Report
Objectives

The objectives of this exercise are to:

- Enable the Power Pivot Add-in
- Import external data into the Power Pivot data model
- Create table relationships
- Mark a “Date” table
- Create calculated columns
- Sort column values by a related column
- Create hierarchies
- Hide columns
- Create calculated fields
- Create a KPI
- Create a PivotTable report

Exercises

This hands-on lab comprises the following exercise:

1. Developing a Power Pivot Data Model

Estimated time to complete this lab: 45 minutes
Exercise 1: Creating a Power Pivot Data Model

In this exercise, you create a Power Pivot data model that will import data from a SQL Server database, an OData data service, and an Excel workbook. You will then complete the Power Pivot data model by defining relationships, calculated columns, hierarchies, calculated fields and a KPI. Finally, you will create a PivotTable report based on the data model.

Task 1 – Creating the Excel Workbook

In this task, you will create a new Excel workbook that will be used to develop the Power Pivot data model and create a report.

1. To open Excel, on the taskbar, click the Excel program shortcut.
2. In Excel, to create a blank workbook, click the Blank Workbook template.

![Blank workbook](image)

**Figure 2**
*Locating the Blank Workbook Template*

3. On the File ribbon tab (also known as the backstage view), select Save As, select Computer, and then click Browse.
4. In the Save As window, browse to the D:\PowerBI\Lab01 folder.
5. In the File Name box, replace the text with Sales Analysis, and then click Save.
Task 2 – Enabling the Power Pivot Add-In

In this task, if necessary, you will enable the Power Pivot Add-in. In Excel 2013, by default, the Power Pivot Add-in is disabled.

1. If the **PowerPivot** ribbon tab is not available, on the **File** ribbon tab, select **Options**.

   **Note:** If the **PowerPivot** ribbon tab is available, there is no need to complete the steps in this task; continue the lab from **Task 3**.

2. In the **Excel Options** window, select the **Add-Ins** page.

3. In the **Manage** dropdown list, select **COM Add-Ins**, and then click **Go**.

4. In the **COM Add-Ins** window, select the **Microsoft Office PowerPivot for Excel 2013** add-in, and then click **OK**.

5. Notice the addition of the **PowerPivot** ribbon tab.
Task 3 – Loading Tables from a SQL Server Database

In this task, you will open the Power Pivot window and load four tables from the TailspinToys SQL Server database. You will remove unnecessary columns to return data only relevant for analysis.

1. On the **PowerPivot** ribbon tab, click **Manage**.

   ![Figure 5](image)
   
   *Launching the Power Pivot Window*

2. When the Power Pivot window opens, if necessary, maximize the window.

3. On the **Home** ribbon tab, from inside the **Get External Data** group, click **From Database**, and then select **From SQL Server**.

4. In the Table Import Wizard, at the **Connect to a Microsoft SQL Server Database** step, in the **Friendly Connection Name** box, replace the text with **TailspinToys**.

5. In the **Server Name** box, enter **localhost**.

6. In the **Database Name** dropdown list, select **TailspinToys**.

   ![Figure 6](image)
   
   *Reviewing the Connection Properties*
7. Click **Next**.

8. At the **Choose How to Import Data** step, accept the default import method, and then click **Next**.

9. In the **Select Tables and View** step, check the **Sales** table.

10. Click **Select Related Tables**.

11. Notice that the **Product** and **State** tables have been selected.

   **Note:** All foreign key related tables to the **Sales** table are automatically selected.

12. Check the **Region** table.

13. Verify that the table selection matches the following.

   ![Figure 7](image)

   *Figure 7*

   **Reviewing the Selected TailspinToys Database Tables**

14. To filter the columns retrieved from the **Product** table, select the **Product** table row, and then click **Preview & Filter**.

15. Uncheck the **Photo** column.

   **Note:** This column contains binary image data. The data model cannot efficiently store this data, and so this technique should be avoided especially when the binary data is large, and there are many rows of data.

   In **Lab 04** you will learn of a technique to reference external images which is more efficient, and suitable for solutions that will be used on the desktop, or published to SharePoint on-premises.

16. Click **OK**.
17. For the **Product** table, click on the **Applied Filters** link, and then review the selected columns.

   ![Figure 8](image1)
   *Reviewing the Applied Filters for the Product Table*

18. Click **OK**.

19. Repeat the steps in this task to remove the **Population** column from the **State** table.

   **Note:** More recent US state population data will be retrieved from a web page by using Power Query in **Lab 02**.

20. Repeat the steps in this task to remove the **OrderNumber** column from the **Sales** table.

21. To create the data model tables and load the tables with data, click **Finish**.

22. Verify that the tables were successfully imported.

   ![Figure 9](image2)
   *Reviewing the Import Process*

23. To review the relationships created, click the **Details** link.

   **Note:** Relationships in the Power Pivot data model have been created for each existing foreign key relationship between all imported tables.
24. Click **OK**.
25. In the Table Import Wizard, click **Close**.

**Task 4 – Loading a Table from an OData Data Service**

In this task, you will import data from an OData data service. The data feed represents a resource made available by the IT department to business analysts to provide them with data to produce a “Date” table.

1. On the **Home** ribbon tab, from inside the **Get External Data** group, click **From Data Service**, and then select **From OData Data Feed**.
2. In the Table Import Wizard, in the **Friendly Connection Name** box, replace the text with **TailspinToysFeeds**.
3. In the **Data Feed URL** box, enter `http://localhost/TailspinToysFeeds/DataService.svc`.
   (Do not include the period.)
   
   _Note:_ For convenience, the URL can be copied from the `D:\PowerBI\Lab01\Assets\Snippets.txt` file.

4. Click **Next**.
5. To filter the columns retrieved from the **Date** table, click **Preview & Filter**.
6. Uncheck the **QuarterKey** and **YearKey** columns.
7. Click **OK**.
8. Click on the **Applied Filters** link, and then review the selected columns.

   ![Selected columns: DateKey, Date, Day, MonthKey, Month, Quarter, Year](image)

   _Figure 10_

   *Reviewing the Applied Filters for the Date Table*

9. Click **OK**.
10. To create the table and import the data, click **Finish**.
11. Verify that the table was successfully imported.

![Image of successful import process]

**Figure 11**
*Reviewing the Import Process*

12. Click **Close**.

**Task 5 – Loading a Table from an Excel Workbook**
In this task, you will import local data stored in an Excel workbook.

1. On the **Home** ribbon tab, from inside the **Get External Data** group, click the **From Other Sources**.

2. In the Table Import Wizard, at the **Connect to a Data Source** step, scroll to the bottom of the list, select **Excel File**, and then click **Next**.

3. In the **Friendly Connection Name** box, replace the text with **ProductCost**.

4. Click **Browse**.

5. In the **Open** window, navigate to the **D:\PowerBI\Lab01\Assets** folder.

6. Select the **ProductCost.xlsx** file, and then click **Open**.

7. Check the **Use First Row as Column Headers** checkbox.

![Image showing the 'Use first row as column headers' checkbox]

**Figure 12**
*Reviewing the Excel File Information*
8. Click **Next**.

9. Remove the **Product** column (the second column) from the **ProductCost** table.

   **Note:** The product name is available in the **Product** table.

10. Review the applied filters for the **ProductCost** table.

    ![](Figure13.png)

    **Figure 13**

    *Reviewing the Applied Filters for the ProductCost Table*

11. Click **OK**.

12. To create the table and import the data, click **Finish**.

13. Verify that the table was successfully imported.

    ![](Figure14.png)

    **Figure 14**

    *Reviewing the Import Process*

14. Click **Close**.
Task 6 – Relating Tables

In this task, you will relate the Sales table to the Date table, and the Product table to the ProductCost table.

1. To switch to Diagram View, on the Home ribbon tab, from inside the View group, click Diagram View.

   **Note:** You can also toggle between Diagram View and Data View by using the buttons located in the bottom right corner.

   If you cannot locate the Date or ProductCost table, scroll horizontally to the right. You can also use the zoom control located in the top right corner to reveal all tables.

2. To move the Date table, drag the table header to position the table nearer to the Sales table.

3. To create a relationship, from the Sales table, drag the Date column on top of the Date column from the Date table.

   ![Figure 15](image)

   **Figure 15**
   *Creating a Relationship by Using Drag-and-Drop*

4. Repeat the previous steps to create a relationship between the ProductSKU column of the Product table, and the SKU column of the ProductCost table.

5. To reorganize the diagram, click Reset Layout (located at the top left of the diagram).

6. When prompted to confirm resetting the layout, click Reset Layout.
Task 7 – Exploring the Power Pivot Data

In this task, you will explore the data in Sales table. The volume of data is not very large, but still you will have the opportunity to appreciate how fast the interactive filter and sort operations can be performed.

1. Right-click the Sales table, and then select Go To.

2. In the record navigator (located at the bottom left of the window), notice the record count of 16,370.

3. To apply a filter, in the StateID column header (second column), click , uncheck (Select All), and then check 1.

![Figure 16](image)

*Figure 16*

*Applying a Filter*

4. Click OK.

5. Notice the record count of 337.

6. In the Revenue column header, click , and then select Sort Largest to Smallest.

7. Notice that the filtered sales records are now sorted in descending order of revenue.

8. To clear the filter, on the Home ribbon tab, from inside the Sort and Filter group, click Clear All Filters.

9. To clear the sort, on the Home ribbon tab, from inside the Sort and Filter group, click Clear Sort.
Task 8 – Marking the Date Table

In this task, you will mark the Date table as a date table. This will ensure that time intelligence functions work correctly, and that certain reporting tools (like an Excel PivotTable) will enable date-related filter options.

1. Select the Date table.

![Figure 17](locating_the_date_table)

**Figure 17**

*Locating the Date Table*

2. On the Design ribbon tab, click Mark as Date Table, and then select Mark as Date Table.

3. In the Mark as Date Table window, in the Date dropdown list, ensure that the Date column is selected, and then click OK.

Task 9 – Configuring Column Value Sorting

In this task, you will configure columns in the Date table so that they will sort values chronologically.

1. In the Month column header, click , and in the filter list, notice that the month values are sorted alphabetically (the months of 2012 are sorted April, August, December, February, etc.).

![Figure 18](reviewing_the_month_value_sort_order)

**Figure 18**

*Reviewing the Month Value Sort Order*

2. Click Cancel.
3. Select the **Month** column by clicking the column header, and then on the **Home** ribbon tab, from inside the **Sort and Filter** group, click the **Sort by Column** image.

![Sort by Column Image](image.png)

**Figure 19**
*Locating the Sort By Column Image*

4. In the **Sort by Column** window, configure the following value.

![](image.png)

**Figure 20**
*Configuring Sort by Column*

**Note:** The **MonthKey** column includes numeric month values, based on the year multiplied by 100, with the month number (January=1, February=2, etc.) added. When sorted, these values will produce the correct chronological sequence of months.

5. Click **OK**.

6. In the **Month** column header, click ☐, and then in the filter list, review the order of the month values.

7. Repeat the steps in this task to sort the **Day** column (second column) values by the **DateKey** column values.

**Task 10 – Creating Calculated Columns**

In this task, you will create two calculated columns in the **Sales** table.

1. Select the **Sales** table.

2. On the **Design** ribbon tab, from inside the **Columns** group, click **Add**.
3. In the formula bar (located above the table grid), enter the following expression.

\[ \times \, f_{\text{c}} \]

**Figure 21**
*Locating the Formula Bar*

**Note:** When you type the *ProductCost* table name, the auto-complete feature will prompt you to complete the object reference. Simply keep typing to filter the available objects, then use the down arrow key to select the required object, and then press Tab.

**DAX**

\[ =\text{RELATED}(\text{ProductCost}[\text{ProductCost}]) \times [\text{Quantity}] \]

**Note:** This expression retrieves the related product cost of the sales row and multiplies it by the sales quantity. The **RELATED** function is navigating across two relationships, first from the *Sales* table to the *Product* table, and then from the *Product* table to the *ProductCost* table (sourced from the Excel workbook).

4. Press **Enter**.

5. Right-click the new column header, and then select **Rename Column**.

6. Replace the column name text with **COGS**, and then press **Enter**.

**Note:** COGS is an abbreviation for Cost of Goods Sold.

7. To set the data type of the **COGS** column, ensure the column is selected, and then on the **Home** ribbon tab, from inside the **Formatting** group, set the **Data Type** to **Currency**.

**Figure 22**
*Configuring the Data Type of the COGS Column*
8. In the State table, create a calculated column by using the following expression.

<table>
<thead>
<tr>
<th>DAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>=RELATED(Region[RegionName])</td>
</tr>
</tbody>
</table>

9. Rename the new column to RegionName.

Note: The RegionName column was added to allow the creation of a hierarchy in the next task. Hierarchies can only be based on columns from the same table.

**Task 11 – Enriching the Model with Hierarchies**

In this task, you will create three hierarchies, one each for the Date, Product and State tables. Hierarchies allow navigation and the summarization of calculations at different levels of aggregation (year, quarter, month, etc.). Note that hierarchies can only be created in Diagram View.

1. Switch to Diagram View.

2. Locate the Date table, then hover over the top right corner of the table, and then click the Maximize button.

   ![Figure 23](image)

   **Figure 23**

   Maximizing the Date Table

3. To multi-select columns, first select the Day column, and then while pressing the Control key, select the Month, Quarter and Year columns.

4. Right-click the selected columns, and then select Create Hierarchy.
5. Modify the name of the hierarchy to **Calendar**, and then press **Enter**.

   **Note:** The correct order of hierarchy levels (Year, Quarter, Month, Day) was automatically configured. The levels have been ordered by the cardinality (count of distinct column values) of each of the selected columns.

6. To hide all columns, select the **DateKey** column, and then while pressing the **Shift** key, select the **Year** column.

7. Right-click the selected columns, and then select **Hide From Client Tools**.

   **Note:** Business users exploring and querying the data model do not need to access the columns directly. Year, quarter and month values are available from the **Calendar** hierarchy which is now the only visible resource in the **Date** table. Also, the key columns were required to sort other columns, and should not be visible to business users.

8. Ensure that the **Date** table looks like the following.

   ![Figure 19](image)

   **Figure 19**

   *Reviewing the Date Table*
9. To minimize the table, at the top right corner, click Restore.

![Figure 24](image)

**Figure 24**

*Restoring the Date Table*

10. Use the steps in this task to implement the following design enhancements.

<table>
<thead>
<tr>
<th>Table</th>
<th>Hierarchy Name</th>
<th>Hierarchy Levels</th>
<th>Hide Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Products</td>
<td>ProductCategory</td>
<td>ProductID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProductName</td>
<td>ProductName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProductName</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>States</td>
<td>RegionName</td>
<td>StateID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StateName</td>
<td>RegionID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RegionName</td>
</tr>
</tbody>
</table>

11. To hide the entire **ProductCost** table, right-click the **ProductCost** table header, and then select **Hide From Client Tools**.

12. Repeat the last step to hide the **Region** table.

**Note:** Both the **ProductCost** and **Region** tables do not need to be accessed by the business user. Recall that the **RELATED** function was used to retrieve and store values from these tables in other related, and visible, tables.
**Task 12 – Defining Calculated Fields**

In this task, you will define six calculated fields in the Sales table. Note that calculations (calculated columns and calculated fields) can only be defined in Data View.

1. To switch to Data View, right-click the header of the Sales table, and then select Go To.
2. To add a calculated field based on the ProductID column, select the header of ProductID column header (third column), and then on the Home ribbon tab, from inside the Calculations group, click the AutoSum dropdown arrow.

![AutoSum Dropdown Arrow](Figure 25)

*Locating the AutoSum Dropdown Arrow*

3. In the dropdown list, select Count.
4. In the calculation area (located at the bottom of the table grid), notice the calculated field. You may need to widen the ProductID column to fully view the cell content, which consists of the calculated field name and value 16,370 (based on the count aggregation of the table rows).
5. In the formula bar, notice the DAX expression that was automatically generated.
6. To rename the calculated field to Product Count, in the formula bar, modify the first portion of the expression as follows.

```
DAX
Product Count:=COUNTA([ProductID])
```

7. Press Enter.
8. To format the calculated field, in the calculation area, right-click the calculated field, and then select Format.
9. In the Formatting window, in the Category list, select Number.
10. In the Format dropdown list, select Whole Number.
11. Check the **Use 1000 Separator** checkbox.

![Figure 26](image)

*Reviewing the Calculated Field Formatting*

12. Click **OK**.

13. Notice the formatting applied to the calculated field value.

14. To add a calculated field based on the **Quantity**, **Revenue** and **COGS** columns, first select the **Quantity** column, and then while pressing the **Shift** key, select the **COGS** column.

15. On the **Home** ribbon tab, from inside the **Calculations** group, click **AutoSum** dropdown arrow, and then select **Sum**.

   **Note:** The **Sum** function is the most commonly used aggregate function used by calculated fields.

16. Select the first of the new calculated fields.

17. To rename the first new calculated field, in the formula bar, modify the first portion of the expression as follows.

   **DAX**

   Units:=SUM([Quantity])

18. Press **Enter**.

19. Format the calculated field by using the **Number** category, and as a whole number by using the thousands separator.
20. Rename the other two new calculated fields.

<table>
<thead>
<tr>
<th>Existing Calculated Field Name</th>
<th>New Calculated Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum of Revenue</td>
<td>Sales</td>
</tr>
<tr>
<td>Sum of COGS</td>
<td>Cost</td>
</tr>
</tbody>
</table>

**Note:** A calculated field cannot have the name of a column in the same table.

21. In the calculations area, select the cell beneath the **Cost** calculated field.

22. In the formula bar, enter the following expression.

**DAX**

Profit:=[Sales] - [Cost]

23. Press **Enter**.

24. Format the calculated field by using the **Currency** category (with default symbol and decimal places).

25. Add one additional calculated field, directly beneath **Profit** calculated field, based on the following expression.

**DAX**

Profitability:=DIVIDE([Profit], [Sales])

**Note:** The **DIVIDE** function divides two expressions, providing that the second argument results in a non-zero number. If the second argument results in zero or blank (missing), then the function will return blank.

26. Format the calculated field by using the **Number** category, and the **Percentage** format (with default decimal places and by using the 1000 separator option).
27. To hide all columns in the Sales table, multi-select all columns, right-click the selection, and then select Hide From Client Tools.

**Note:** The Sales table consists of foreign key and measure columns. These do not need to be visible when browsing the data model. The table will now only expose the calculated fields added in this task.

In Lab 03 to prepare the data model for Power View reporting, you will learn about a technique to summarize numeric columns by unhiding them.

**Task 13 – Defining a KPI**

In this task, you will define a KPI to support the monitoring of profitability. The company wants to monitor profitability on all products, and expects to achieve a 10% ratio of profit to revenue. Profitability less than 5% is to be reported as critically off track.

1. To create a KPI, in the calculations area, right-click the Profitability calculated field, and then select Create KPI.
2. In the Key Performance Indicator (KPI) window, in the Define Target Value section, select the Absolute Value option.
3. In the corresponding box, replace the value with 1.
4. In the Define Status Thresholds boxes, modify the first value to 0.05, and the second value to 0.1, and then press Enter.

**Note:** As the values are so near each other, the user interface almost overlays them.
5. Verify the KPI configuration looks like the following.

![KPI Configuration](image)

**Figure 27**
*Reviewing the KPI Configuration*

6. Click **OK**.

7. Notice the updated icon against the calculated field.

<table>
<thead>
<tr>
<th>Product Count</th>
<th>Units</th>
<th>Sales</th>
<th>Cost</th>
<th>Profit</th>
<th>Profitability</th>
</tr>
</thead>
<tbody>
<tr>
<td>16,370</td>
<td>172,564</td>
<td>$41,000,301.00</td>
<td>$33,029,620.00</td>
<td>$7,970,681.00</td>
<td>19.44%</td>
</tr>
</tbody>
</table>

**Figure 28**
*Reviewing the Calculated Fields and KPI*
Task 14 – Creating a PivotTable Report

In this task, you will add a PivotTable report to the workbook. You will then develop the report layout by introducing fields to the Values, Slicers and Row Label drop zones. Finally, you will apply conditional formatting to the product sales values, and sort the products, within their subcategory, in descending sales order.

1. On the Home ribbon tab, click the dropdown arrow beneath PivotTable, and then select PivotTable.

2. In the Create PivotTable window, select the Existing Worksheet option.

3. In the Location box, modify the text to use ‘Sheet1’!$B$1.

4. Click OK.

5. Notice that a PivotTable has been added to an Excel worksheet, and that the PivotTable Fields pane is open (located at the right).
6. To rename the worksheet, right-click the Sheet1 worksheet tab, and then select Rename.

7. Rename the worksheet to Profitability Monitoring, and then press Enter.

8. In the PivotTable Fields pane, expand each of the four groups, and review the resources that are available to produce the PivotTable layout.

9. To add a report filter, in the Date table, drag the Calendar hierarchy into the Filters drop zone.

10. In the PivotTable report filter (cell C1), click ⬇️, expand the All member, select CY2013 | CY2013 Q3 | 2013-Sep, and then click OK.

11. In the PivotTable Fields pane, in the Sales table, check the Sales and Profit fields.

12. Expand the Profitability KPI, and then check the Value and Status fields.

   **Note:** The status icon in the Profitability Status column is displayed by using Excel conditional formatting. The status value is either the value -1 (off target), 0 (slightly off target) or 1 (on target).

13. To introduce row labels, in the PivotTable Fields pane, in the Product table, expand the More Fields folder, and then check the ProductSKU field.

   **Note:** Visible columns are always available in the More Fields folder, and are hierarchies consisting of a single level. Hierarchies created in the Diagram View, expressing multiple levels, are always available directly within the table.
14. To introduce a slicer, in the **PivotTable Fields** pane, in the **State** table, expand the **States** hierarchy, right-click the **RegionName** level, and then select **Add as Slicer**.

15. Resize column **A** to accommodate the width of the slicer.

16. To move the slicer, drag the slicer header and relocate it to the top of column **A**.

17. In the **RegionName** slicer, select **Pacific Northwest**.

18. To sort the product SKUs in descending order of profitability, right-click any cell in the **Profitability** column, and then select **Sort | Sort Smallest to Largest**.

**Task 15 – Finishing Up**

In this task, you will finish up by closing Excel.

1. To save the workbook, on the **File** ribbon tab, click **Save**.

2. To close Excel, click the X button in the top right corner.

**Summary**

In this lab, you created a Power Pivot data model that imported data from a SQL Server database, an OData data service, and an Excel workbook. You then enhanced the data model by defining relationships, calculated columns and hierarchies. You then hid columns not required in the data model interface, and then you defined calculated fields and a KPI. Finally, you produced a PivotTable report to monitor regional product profitability.