

Brett Powell

Foreword by:

**Gilbert Quevauvilliers**

Microsoft MVP - *Power BI and Microsoft Power BI Consultant at Fourmoo*

# Microsoft Power BI Cookbook

Creating Business Intelligence Solutions of Analytical  
Data Models, Reports, and Dashboards



Packt

# 5

## Creating Power BI Dashboards

In this chapter, we will cover the following recipes:

- Building a Power BI dashboard
- Constructing an enterprise dashboard
- Developing dynamic dashboard metrics
- Preparing datasets and reports for Q&A natural language queries
- Embedding analytical context into visualizations
- Exposing what matters - top N and percentage of total visualizations
- Leveraging Power BI reports in Power BI dashboards
- Deploying content from Excel and SSRS to Power BI
- Adding data alerts and email notifications to dashboards

### Introduction

Power BI dashboards are collections of tiles created in the Power BI service, representing the visuals from one or many Power BI reports and optionally other sources, such as Excel and **SQL Server Reporting Services (SSRS)**. Dashboards are best used to centralize essential measures and trends into a visually and mobile optimized layout, and to provide an entryway to other dashboards or reports with additional details. Additionally, dashboards can be enhanced with URL links, streaming data, images, web content, and interactivity.

*"A dashboard is really a content aggregator. It lets you bring together lots of different data sources in one place so you can have a 360 degree view of your business on one dashboard."*

- Adam Wilson, group program manager for Power BI service

In this chapter, Power BI dashboards are constructed to provide simple at a glance monitoring of critical measures and high impact business activities. The unique features of dashboards, such as Q & A natural language queries, data alerts, and integration of other report types, such as Excel and SSRS, are also included.

## Building a Power BI dashboard

With a robust data model and multiple reports created in Power BI Desktop, dashboards can be created in the Power BI service to consolidate the essential visuals from these reports onto one canvas. Additionally, the dashboard will provide an access point to the detailed reports supporting the tiles and will be optimized for mobile access through the Power BI mobile application.

This recipe walks through all the essential components of building a Power BI dashboard, from creating an app workspace to hold the dashboard to enhancing the layout and settings of the dashboard.

## How to do it...

### Dashboard design process

1. Define dashboard consumers and requirements.
  - The report design planning process described in *Chapter 4, Authoring Power BI Reports* is directly applicable to dashboards as well
  - Confirm that the existing data model (or models) supports the required business questions
2. Map dashboard tiles to reports and datasets.
  - In this example, a sales dashboard is created for the North American sales management team:



North America sales dashboard structure: 4 reports and 1 dataset

- Four reports are created from the same dataset (model) and one or more visuals from each report are pinned as tiles to the sales dashboard

## Dashboard development process

1. Create an app workspace.
  - In the PBI service, click on **Create app workspace** from workspaces in the **Navigation** pane
  - Name the workspace and add team members with edit access who will be creating dashboard content
2. Import the dataset.
  - From the new app workspace, click on **Get Data** and then **Get** from the **File** menu to add the PBIX dataset
3. Create the dashboard.
  - From the app workspace in Power BI, click on dashboards and then click on **Create** in the top-right corner
  - Name the new dashboard and delete the dashboard that was created when the dataset was imported
    - Alternatively, a new dashboard can be created when a visual is pinned from a report

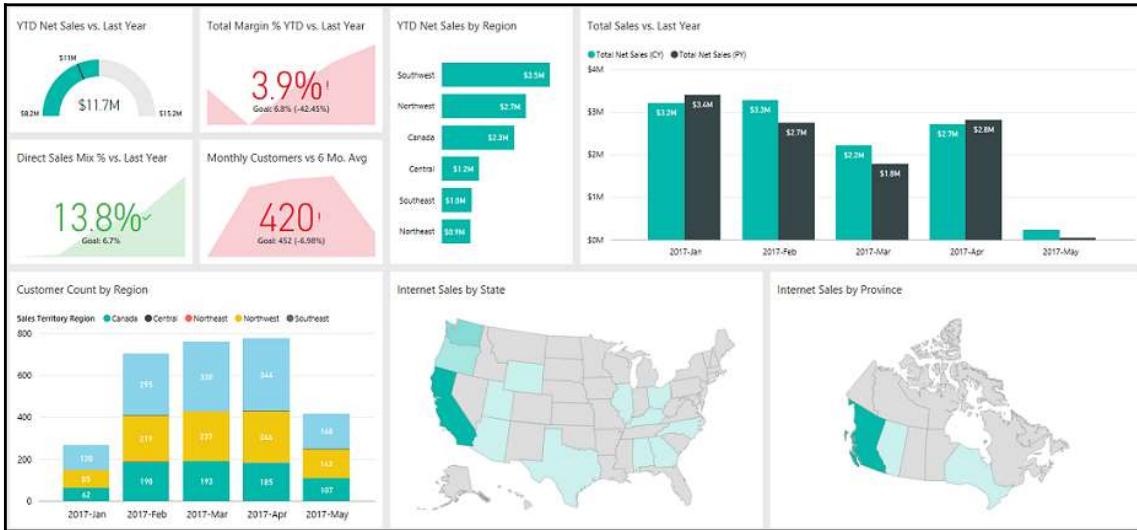
4. Create dashboard reports.
  - From a blank PBIX file, connect to the published dataset hosted in the app workspace
  - The Power BI service data source is available under online services
5. Copy the connected PBIX file for each report needed for the dashboard and develop the report visuals.

For better manageability and version control, the PBIX files can be stored and imported from OneDrive for business.



Design the report pages in context of the dashboard and app workspace. The report visuals should directly support the tiles of the dashboard, such that a user can instantly derive more useful details by clicking a dashboard tile.

6. Publish the reports.
  - From the **Home** tab in Power BI Desktop, click on **Publish** from each of the reports
  - The dashboard, reports, and dataset are now within the sales management app workspace in Power BI
7. Pin visuals to the dashboard.
  - In the app workspace, open a report and select a visual to be pinned to the dashboard
  - Click the pin icon and choose the existing dashboard; repeat this process for each report in the workspace
8. Refine dashboard layout.
  - Move and resize the dashboard tiles such that most important visuals are in the top and left corners



North America sales dashboard

- The dashboard provides the at a glance visibility to the measures most important to the North America sales team
- The user can access any of the four detail reports (country, customer, margin, and region) by clicking a dashboard tile

## Constructing an enterprise dashboard

Power BI dashboards are valuable assets for specific stakeholders and focused use cases, but their greatest strength is in consolidating important information from across an enterprise. These dashboards generally source from multiple datasets, such as SSAS tabular models, and often integrate on-premise with cloud-borne data. Enterprise dashboards typically utilize card and KPI visuals to focus on strategic objectives and maximize canvas space. Given the scale and breadth of data sources for a modern enterprise, a significant level of coordination is required to ensure that all datasets supporting the dashboard represent an appropriate level of data quality and governance.

In this recipe, an enterprise dashboard is constructed based off of four datasets (models) to include key measures across sales, inventory, general ledger, and customer service business processes.

## How to do it...

### Dashboard design process

1. Define dashboard requirements.
  - Map the required dashboard tiles to existing datasets (that is, data models) and source systems
  - For any new dataset to be created, evaluate readiness, scale, and data retrieval and modeling needed
  - The data warehouse bus matrix and model planning described in Chapter 3, *Building a Power BI Data Model*, can help guide this process to promote re-usability and version control
2. Map dashboard tiles to reports and datasets.



Enterprise dashboard supported by four datasets

- The design process results in a logical architecture of the components needed in Power BI to support the dashboard

As each report is tied to a single dataset, consolidating required schemas into fewer data models (datasets) can simplify manageability, and support richer analytics and visualizations.





An alternative dashboard architecture is for an executive dashboard to link to multiple dashboards (such as corporate divisions or product groups), which then link to individual reports. The option to link a dashboard tile to a separate dashboard is available in Tile details when set custom link is enabled.

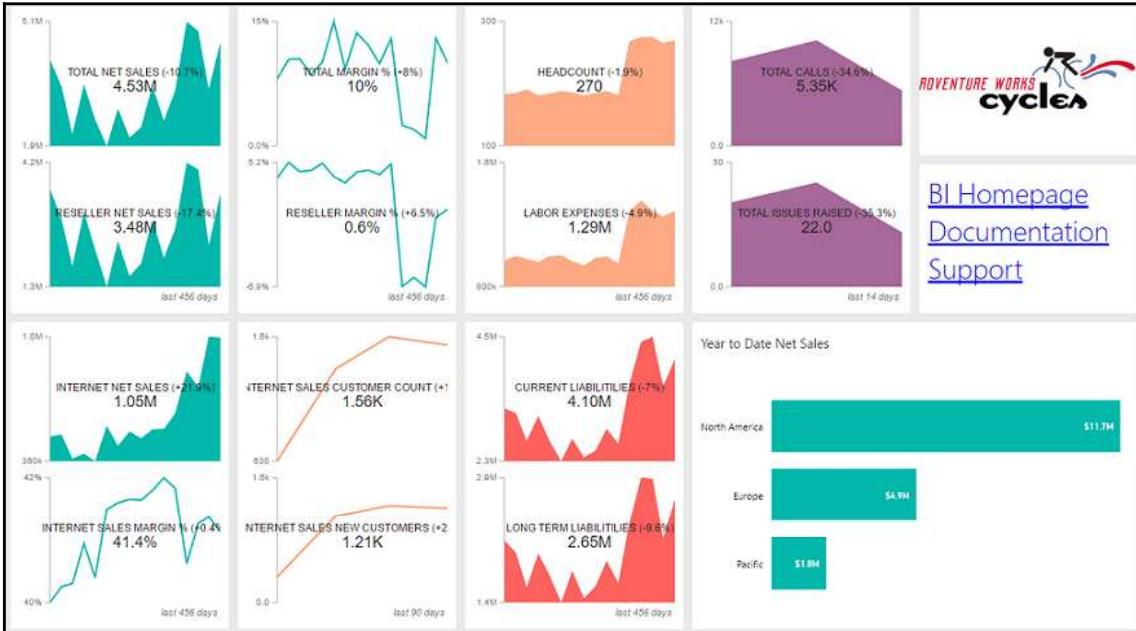
3. Create an app workspace.
  - The same steps as in creating a workspace and adding team members from the previous recipe apply here
  - In the workspace, a blank dashboard can be created, and existing datasets and reports can be added
4. Create or augment datasets.
  - This could be a new Power BI Desktop file or additional queries and measures to an existing model
  - Publish or import the completed dataset to the new app workspace in Power BI
5. Create dashboard reports.
  - Connect PBIX report files to the data models, such as the published dataset or an SSAS tabular model
  - The visuals for the enterprise dashboard, such as Cards and KPIs, should follow a standard formatting scheme



The individual PBIX report files can be stored in OneDrive for business (if available) or at a secure network location. A single report page may be sufficient for certain tiles, while others (such as, sales) require robust report details.

6. Publish reports.
  - Publish each report to the workspace and then pin the required visuals to the existing dashboard
7. Refine dashboard layout.
  - Organize and resize the tiles to prioritize KPIs and make the best use of the canvas
  - Apply a mobile layout to the dashboard by switching from Web view to Phone view in the top right corner

The dual KPI chart types and color options (for both bottom and top charts) are used to help distinguish the different dashboard metrics. For example, sales metrics are displayed with green area charts, while the liabilities metrics are red, and the margin metrics are presented as line charts.



Enterprise dashboard: dual KPIs, corporate logo and links

## 8. Customize tile settings.

- Click the ellipsis of the dashboard tiles and then the pencil icon to open the tile details form
- Optionally, enter a title and subtitle for the tile, as well as a custom URL link, such as a SharePoint site

## 9. Add supporting tiles.

- Optionally, click on **Add tile** to add text boxes, images, and other content to support the dashboard

## How it works...

### Dual KPI custom visual

- The dual KPI custom visual, developed by Microsoft, was used in this recipe to efficiently utilize dashboard canvas
- See Chapter 9, *Applying Advanced Analytics and Custom Visuals*, for additional details on this visual

### Supporting tiles

- The corporate logo was pinned to the dashboard from an image that was inserted into a report
- The URL links were created within the **Add tile** feature in the Power BI Service for a textbox

## Developing dynamic dashboard metrics

Dashboard tiles represent snapshots of report visuals. The values of the measures and columns used in the pinned report visual are refreshed, but modifications to the report, such as filters or formatting, are not reflected in the dashboard. Therefore, it's necessary to develop dynamic logic in the data model and measures that keep dashboard tiles current across time. Additionally, dynamic logic simplifies the user experience of interacting with reports and the dashboard report development process.

In this recipe, two columns are added to the date dimension table and used to drive dashboard report visuals. Additionally, a measure is created to provide a target value to support comparison with a KPI.

## How to do it...

### Dynamic date columns

1. Identify the grain(s).
  - Dashboards are usually focused on the current year, month, or week, and trends within the current year or recent history
  - Whether calendar or fiscal periods, columns associating dates to these values enhances and simplifies analysis
    - Time Intelligence measures, reviewed in Chapter 6, *Getting Serious with Date Intelligence*, enable the comparison of distinct time frames
2. Modify the date view.
  - Add columns to the SQL view supporting the date dimension table for the required grains with dynamic expressions

```
CASE
    WHEN YEAR(D.Date) = YEAR(CURRENT_TIMESTAMP) THEN 'Current
Calendar Year'
    WHEN YEAR(D.Date) = YEAR(CURRENT_TIMESTAMP)-1 THEN 'Prior
Calendar Year'
    WHEN YEAR(D.Date) = YEAR(CURRENT_TIMESTAMP)-2 THEN '2 Yrs Prior
Calendar Year'
    WHEN YEAR(D.Date) = YEAR(CURRENT_TIMESTAMP)-3 THEN '3 Yrs Prior
Calendar Year'
    ELSE 'Other Calendar Year'
END AS [Calendar Year Status]

CASE
    WHEN YEAR(D.Date) = YEAR(CURRENT_TIMESTAMP) AND MONTH(D.Date) =
MONTH(CURRENT_TIMESTAMP)
        THEN 'Current Calendar Month'
    WHEN YEAR(D.Date) = YEAR(DATEADD(MONTH, -1, CAST(CURRENT_TIMESTAMP
AS date))) AND
        MONTH(D.Date) = MONTH(DATEADD(MONTH, -1, CAST(CURRENT_TIMESTAMP
AS date)))
        THEN 'Prior Calendar Month'
    WHEN YEAR(D.Date) = YEAR(DATEADD(MONTH, -2, CAST(CURRENT_TIMESTAMP
AS date))) AND
        MONTH(D.Date) = MONTH(DATEADD(MONTH, -2, CAST(CURRENT_TIMESTAMP
AS date)))
        THEN '2 Mo Prior Calendar Month'
    WHEN YEAR(D.Date) = YEAR(DATEADD(MONTH, -3, CAST(CURRENT_TIMESTAMP
```

```
AS date))) AND  
MONTH(D.Date) = MONTH(DATEADD(MONTH, -3, CAST(CURRENT_TIMESTAMP  
AS date)))  
THEN '3 Mo Prior Calendar Month'  
ELSE 'Other Calendar Month'  
END AS [Calendar Month Status]
```

- Standard SQL syntax with CASE and CURRENT\_TIMESTAMP () create two columns, each with five distinct values (Current Calendar Month, Prior Calendar Month, 2 Mo Prior Calendar Month, 3 Mo Prior Calendar Month, and Other Calendar Month)
- Since the date table query is executed on a schedule to support dataset refresh (or at run time if DirectQuery mode), these columns will be updated and available to the report author, thus avoiding stale or hard coded reports:
  - As an example, a report level filter could be set to only include the Current Year and Prior Year values of the Calendar Year Status column, and a page of the report could be set to only include the Current Calendar Month and Prior Calendar Month values of the Calendar Month Status column
  - The tables and charts of the report would update to respect the dates these values refer to as the dataset or report is refreshed



Alternatively, if the date dimension table is updated daily via an ETL process, then the new dynamic columns could be included in this process and persisted in the table. Additionally, if both the source table and the SQL view cannot be modified, M queries with conditional expressions can be used to create the derived columns.

3. Implement in reports.

- The new dynamic date columns can be used as filters in dashboard reports in the following methods:
  - As report level, page level, and visual level filters within the reports used to support dashboards
  - As slicer visuals in report pages to allow for further analysis of the dashboard tile

- Optionally, as filter arguments in time intelligence measures; see Chapter 6, *Getting Serious with Date Intelligence* for greater detail

```
Internet Net Sales (CY YTD) = CALCULATE([Internet Net Sales],  
    FILTER(ALL('Date'), 'Date'[Calendar Year Status] = "Current  
    Calendar Year" &&  
    'Date'[Date] <= MAX('Date'[Date])))
```

When used as a measure filter, the Calendar Year Status column can avoid the need to apply other filters (that is, slicers and page level filters) to produce the current or prior year value. This can be helpful for the fixed requirements of dashboard visuals but limits the use of these measures in self-service analysis with other date dimension columns. However, the standard and recommended date intelligence practice is to set the year column equal to the max value of this column in the current filter context, such as `Date[Calendar Year] = MAX(Date[Calendar Year])`. See Chapter 6, *Getting Serious with Date Intelligence* for examples of standard and more advanced custom date intelligence expressions.



For many dashboards, it's necessary to exclude the current month, given a large transaction or GL entry to be applied at the end of the month. A visual level filter, based on the dynamic month column, can support this scenario. Using the Calendar Month Status column from this recipe as example, four of the five computed values would be included, but the Current Calendar Month value would be filtered out of the visual.

## KPI target measures

1. Identify the KPI.
  - These are dashboard measures which lack a relevant target to drive formatting (that is, red, yellow, green)
2. Define the target logic.
  - In the absence of a budget or plan, work with stakeholders to define upper and lower boundaries of the KPI
  - Without specific guidance or requirements, a trailing average measure of the KPI can be used

3. Develop the target measure.

- In this example, the KPI trend axis is at the month grain and thus the target is needed at the month grain

```
Internet Sales Customer Count Average of Trailing 6 =  
VAR LastNonBlankMonth =  
LASTNONBLANK('Date'[CalYearMonthIndex], [Internet Sales Customer  
Count])  
VAR SixMonthsPrior = LastNonBlankMonth - 6  
RETURN  
CALCULATE(AVERAGEX(VALUES('Date'[CalYearMonthIndex]), [Internet  
Sales Customer Count]),  
FILTER(ALL('Date'), 'Date'[CalYearMonthIndex] <= LastNonBlankMonth  
-1 && 'Date'[CalYearMonthIndex] >= SixMonthsPrior))
```

- The target measure identifies the last month with customers and then only uses completed months for evaluation

## How it works...

### Target measure - trailing 6 months

- With `CALCULATE()`, the target measure can include prior months that are filtered out in the report via filters or slicers
- `LASTNONBLANK()` identifies the last month that has customer count in the current filter context
- Per earlier recipes, a sequentially increasing column can be used to drive Time Intelligence measures.

# Preparing datasets and reports for Q & A natural language queries

Q & A can be a powerful method of enabling users to explore datasets, by directly submitting their own questions in both the Power BI service and through the Power BI mobile application. The tables and measures of each dataset, represented by a tile on the dashboard, are available to answer Q & A questions and per *Chapter 4, Authoring Power BI Reports, Enhancing exploration of reports* recipe, reports and featured questions can be configured to aid the Q & A experience.

This recipe provides data model design and metadata tips to prepare a dataset for Q & A. Additionally, synonyms are added to Power BI Desktop data model to improve the accuracy of natural language queries.

## Getting ready

### Determine use cases and feasibility



Q & A may not be appropriate for certain dashboards and datasets. For example, the Q & A search bar "ask a question about your data" may be a distraction to users of the enterprise dashboard, who only want to view the KPIs. Additionally, if the dataset requires a gateway, such as an on-premises SSAS server or a DirectQuery Power BI Desktop model to an on-premises source, Q & A may be avoided given the additional (and potentially inefficient) queries and performance considerations.

1. Enable or disable Q & A.
  - In the Power BI service, access the app workspace containing the dashboard
  - Click the Gear icon and select **Settings**. From the **Dashboards** tab, click **Show the Q & A search box**.



Currently, Q & A is not supported for Power BI Desktop Models in DirectQuery mode. Q & A is available for imported Power BI Desktop datasets and SSAS datasets. If supported in the future, this can be toggled on or off in the datasets' settings dialog. Additionally, all data models (Import or DirectQuery) with row level security roles applied cannot be used with Q & A.

## How to do it...

### Prepare a model for Q & A

#### Model metadata

1. Revise any columns with incorrect data types, such as dates or numbers that are stored as text data types.
2. Set the default summarization for dimension columns to do not summarize.
3. Associate geographical columns, such as states and zip codes, with a related data category.

#### Model design

1. Split columns containing multiple values into distinct columns.
2. Normalize tables such that entities within the tables are moved to their own distinct table.
  - For example, columns of a vendor in a products table can be moved to a vendor table
3. Q & A queries only work with active relationships of a model.
  - Consider dedicated role playing dimensions with active relationships
  - Alternatively, consider de-normalizing an inactive relationship dimension into a fact table

## Apply synonyms

### Analyze Q & A use cases

1. Define the top or most common natural language questions and test for accuracy in Power BI.
2. Identify the gaps between the names of data model entities and the names used in natural language queries.
  - Focus on entities with longer and less intuitive names, that aren't used casually by the users

## Apply synonyms

1. Open the Power BI Desktop data model (or models) supporting the dashboard locally.
2. Select the relationships window and click the synonyms icon from the modeling tab.
3. The synonyms window will open on the right. Select a table in the diagram view to access its synonyms:



Synonym added to the reseller table

- Table names are at the top, measures are associated with their home tables. Names of the entities are synonyms by default.
4. Click in the input box of the table name, column name, or measure name and add synonyms separated by commas.



Avoid reusing the same synonym across multiple entities, as this can lead to incorrect query results. Ensure that the primary synonym for each entity of the model is unique.

## Publish the dataset

1. Save and publish the dataset to the app workspace in Power BI:



Wholesaler term entered in Q & A associated with the reseller table via the synonym

2. Test the query behavior with the synonyms and optionally create featured Q & A questions per Chapter 4, *Authoring Power BI Reports*.

## Embedding analytical context into visualizations

Users of Power BI dashboards appreciate the linking of tiles to detailed reports to support further analysis. However, many users are not comfortable navigating through reports and would prefer that the dashboard itself provides all the essential information. Embedding Tooltips and conditional formatting logic into dashboard visuals are two powerful techniques to raise the insight to effort ratio mentioned in Chapter 4, *Authoring Power BI Reports*, while not compromising the performance or manageability of the solution.

In this recipe, a simple column chart of sales by month is enhanced with tooltip measures and conditional formatting logic. When pinned to a dashboard, the users instantly visualize a negative outcome and can hover over the bars for additional context.

### How to do it...

#### Design the visual

1. Identify the essential components of a dashboard measure, such as group contributions to a total value.
2. Determine what rule or measure should drive any color changes, such as a negative variance to plan.

#### Create the visual

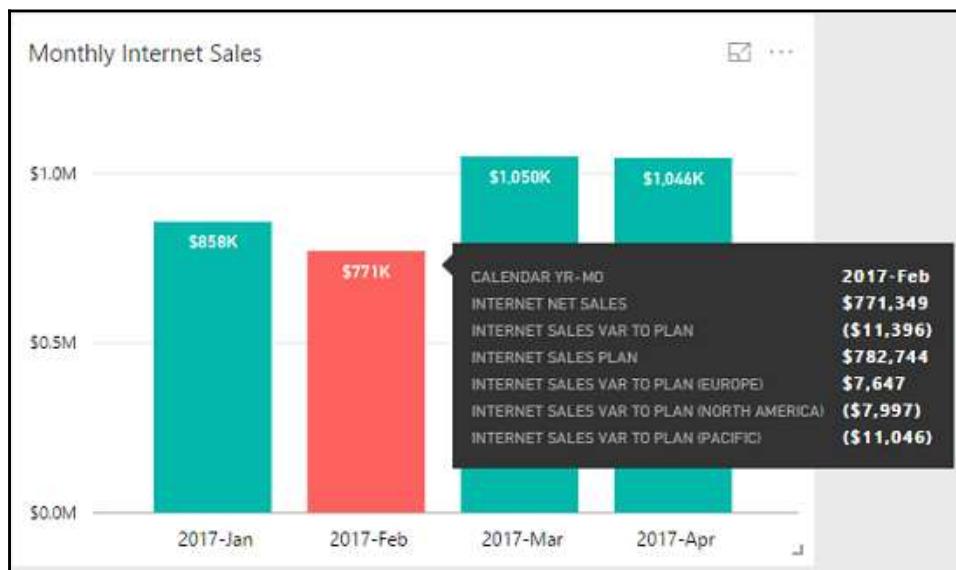
1. Open a Power BI Desktop report and create a clustered column chart visual with a measure and a dimension:
  - The measure will be added to the value field well and the dimension to the axis
2. Add a measure to the color saturation field well that will drive the color formatting of the value field.

3. From the formatting pane, open the data colors card and enter minimum and maximum values.
4. In the data colors formatting options, associate colors with these values, such as red with the minimum.



The color saturation field could be the same measure as the value field or a different measure that provides even further context. In this example, internet net sales is used as the value and internet sales var to plan is used for color saturation. Color saturation can only be applied against a single value field—this option is not available when multiple values are used.

5. Add measures to the tooltips field well, that give context to either the value or the color saturation field:
  - In this example, measures specific to the three sales regions are added to the tooltips field well:



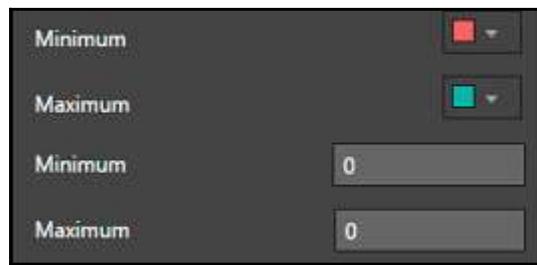
Dashboard tile with tooltips and conditional formatting

- Hovering over **2017-Feb** exposes the tooltip measures. The tooltips (by region) help explain the \$11K miss to plan.

## How it works...

### Color saturation rule

The rule applied in this example is to color the bars red if sales misses the plan. Otherwise, use the default green theme.



The internet sales var to plan measure used for color saturation is only negative for the 2017-Feb month.

### Tooltip measures

Simple CALCULATE() functions can be used to create the tooltip measures if they don't exist in the dataset.

```
Internet Sales Var to Plan (Europe) =  
CALCULATE([Internet Sales Var to Plan], 'Sales Territory'[Sales Territory  
Group] = "Europe")
```

If tooltip measures begin to clutter the fields list, they can be hidden or organized in dedicated measure group tables.

### There's more...

See the recipes in Chapter 4, *Authoring Power BI Reports*, for further examples of conditional formatting, including line charts, tables, and matrix visuals.

## Exposing what matters - top N and percentage of total visualizations

A common use case for dashboards is to highlight and monitor the drivers of significant events or business trends. For example, an enterprise dashboard may feature a reseller margin % KPI visualization, but a separate dashboard may identify the top and bottom 10 individual resellers and products by margin %.

In this recipe, dashboard visuals are created leveraging the top N filter type available to visual level filters and DAX measures to present focused, actionable information in Power BI dashboards.

### How to do it...

#### Top 25 resellers with below -3% margin

1. Create a table visual with the dimension name (reseller) and two measures (reseller margin % and reseller net sales).
2. Add the dimension key column (reseller key) to the visual level filters pane.
3. In visual level filters, click the drop-down for reseller key and select top N as the filter type.
4. Enter 25 for **Show Items** and drag the reseller net sales measure to the **By Value** input box. Click **Apply Filter**.
5. In visual level filters, open the drop-down for the reseller margin % measure.
6. In the **Show items** when the value: input box, use the is less than option and enter (- .03). Click **Apply Filter**.



Top N and measure filters applied at visual level

7. Sort the table by margin % ascending and apply a format to the table visual that will align with the dashboard.

Top 25 Resellers with Below -3% Margin		
Reseller	Reseller Margin % ▲	Reseller Net Sales
Golf and Cycle Store	-4.1 %	\$585,516
Fitness Toy Store	-3.6 %	\$649,392
Metropolitan Equipment	-3.4 %	\$542,660
<b>Total</b>	<b>-3.7 %</b>	<b>\$1,777,569</b>

Formatted table visual reflecting low margin, high impact resellers

- In this example, only three of the larger (top 25) resellers are included in the visual



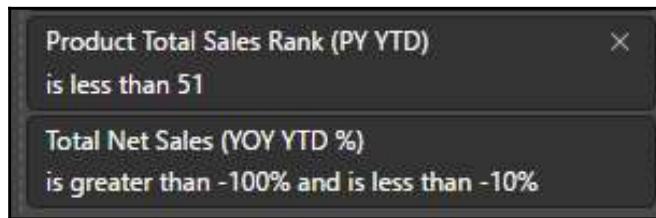
Only a single top N filter condition can be applied to a visual. As an alternative or supplement to top N, ranking measures can be created and used as visual level filters.

## Last year's top 50 products with below -10% growth

1. Create a table visual with the product name column and a year-to-date growth measure.
2. Add a prior year, year-to-date ranking measure to the visual level filters field well:
  - Enter 101 in the is less than condition and click on **Apply Filter**

3. In the visual level filters pane, click the dropdown for the year-to-date growth measure:

- Enter `-1` for an is greater than condition and `-10%` for an is less than condition. Click on **Apply filter**.



Visual level filters applied to ranking measure and YOY YTD % growth measure

- The greater than `-100%` condition accounts for products which aren't being sold this year.
4. Sort by growth measure ascending and optionally add additional supporting measures and apply a format.

The finished table visual displays only the top (50) products from last year which are still being sold in the current year but with declining sales revenue of 10% or more.

Last Year's Top 50 Products with Below -10% Growth			
Product Name	Total Net Sales (YOY YTD %)	Total Net Sales (CY YTD)	Total Net Sales (PY YTD)
Road-250 Red, 58	-15.9 %	\$298,089	\$354,286
ML Road Frame-W - Yellow, 38	-12.5 %	\$59,602	\$68,135
ML Road Frame-W - Yellow, 48	-12.4 %	\$59,959	\$68,460
<b>Total</b>	<b>-14.9 %</b>	<b>\$417,650</b>	<b>\$490,880</b>

Formatted table visual reflecting high value products from last year with declining sales



Look to leverage drill through report pages for deep analysis of individual dimension values such as products or vendors. In this example, after the user has accessed the underlying report from the dashboard, the user could right-click one of the three rows displaying the Product Name column and drill through to a report page that provides great detail about the specific product. See the *Enhancing the exploration of reports* recipe in Chapter 4, *Authoring Power BI Reports* for additional details.

5. Publish the report to the app workspace in the Power BI service and pin the exception visual to a dashboard tile.

## How it works...

### Prior year rank measure

- The ranking measure removes filters on the product name and the alternate key via ALL(), prior to the RANKX() evaluation.

```
Product Total Sales Rank (PY YTD) =  
IF(HASONEVALUE('Product'[Product Alternate Key]),  
RANKX(ALL('Product'[Product Name],'Product'[Product Alternate Key]),  
[Total Net Sales (PY YTD)],,DESC,Skip),BLANK())
```

- The HASONEVALUE() function is used to check if a single product is in the filter context. If not, a blank is returned.

## Visualizing performance relative to targets with KPIs and gauges

KPI and gauge visuals are frequently used to present the most critical performance measures in dashboards. Given their compact size and supporting context, such as trend graphs and target values, users can quickly obtain useful insights from these visuals alone, on any device. However, to derive the most value out of these visuals, it's often necessary to apply Visual level filters, create supporting target measures, and group related visuals.

In this recipe, a KPI and gauge visual are developed to present growth relative to planned growth. Groups of KPI visuals are then created to provide visibility to current period, prior period, and year-to-date.

## How to do it...

1. Identify the measure:
  - This measure will serve as the Indicator input for the KPI visual and the value input for the gauge visual
2. Define the grain:
  - For the KPI visual, the date dimension column used in the trend axis input will determine
  - For example, a period trend axis will result in the value of the indicator for the latest period



Per Chapter 3, *Building a Power BI Data Model*, it's essential that Sort By columns are applied to date dimension columns, such as month and year-month. Without this configuration, the default sort order will cause the KPI visual to select the value for the last alphabetical month rather than the last chronological month via the trend axis input.

3. Identify the target:
  - This measure will serve as the target goal for the KPI visual at the grain of the trend axis input
  - For the gauge visual, this measure will be the target value input, and optionally, the maximum value as well

## Create the visuals

1. Create a new Power BI report, connect to the dataset, and create two blank visuals: a KPI and a gauge.
2. Add the measures identified in steps 1 through 3 to their designated field wells.

3. Provide a custom title and apply any other helpful formatting, such as enlarging the target on the gauge visual.



Standard KPI and gauge visuals: current month growth versus planned growth



In this example, the trend axis on the KPI is monthly, and since the month isn't complete, the current growth of 250 percent is well below the 463 percent growth planned for the given month. As the month progresses (and data refreshes), the 250 percent will increase.

## Grouping KPIs

Create a prior month and a year-to-date KPI visual to supplement the current month KPI visual:



Year-to-date and prior month KPI visuals created to support the dashboard



The same monthly trend axis column (for example, 2017-May) is used for the two new KPI visuals. The year-to-date KPI simply substitutes year-to-date measures for the indicator and target goals. The prior month sales growth KPI uses the same inputs but the Calendar Month Status column is used as a visual level filter for both this visual and the year-to-date KPI.

## Publish KPIs to dashboard

1. When complete, publish the report with the KPI visuals to the App Workspace containing the dashboard.
2. Pin the visual(s) to the dashboard and re-organize the layout.



Per the *Data alerts* recipe in this chapter alerts can be configured for standard card, KPI, and gauge visuals published to dashboards. See this recipe for an example of setting a notification, and optionally an email message, based on these visuals.

## How it works...

### Current month filter

- For the gauge visual, the Calendar Month Status dynamic date dimension column, described earlier in this chapter, is used a Visual level filter. This filter is set to Current Calendar Month.
- For both the year-to-date and the prior month sales growth KPI visuals, the Calendar Month Status column is used as a visual level filter. This filter is set to Current Calendar Month.
  - The prior month sales growth KPI therefore defaults to the next latest month—the prior month
  - The YTD KPI provides an apples-to-apples comparison by only comparing completed months against plan

## Time intelligence measures

- Year over year, percentage measures are used for both the indicator and target input fields of the visuals
- These measures, in turn, reference current year, prior year, current year to date, and prior year to date measures
  - For example, the year-to-date target measure compares the sales plan (YTD) versus sales (prior YTD)
- See Chapter 6, *Getting Serious with Date Intelligence*, for additional detail on developing time intelligence measures

## Leveraging Power BI reports in Power BI dashboards

By default, the tiles of a dashboard are independent of each other and cannot be filtered or interacted with. Additionally, modifications to reports after visuals have been pinned to dashboards, such as filter and layout changes, are not automatically reflected in the dashboards. In many scenarios, the users consuming a dashboard want to retain the interactive filtering experience of Power BI reports from within their dashboard and it can be helpful to automatically synchronize reports with dashboards.

In this recipe, a fully interactive live page of visuals is pinned to a dashboard, along with additional supporting visuals.

## How to do it...

### Define live page requirements

1. Determine the top business questions and thus measures and visuals to include in the page:
  - As report pages (and reports) are limited to a single dataset, confirm feasibility with existing models

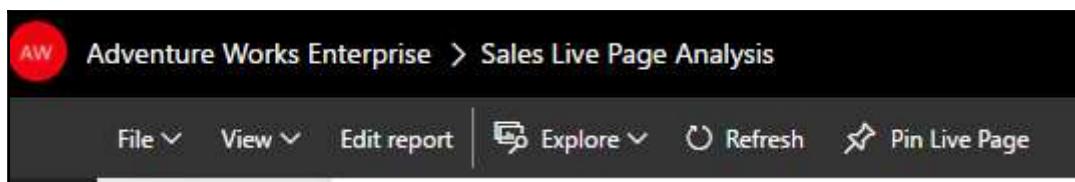
2. For a single report page, identify the essential dimensions that will be used as slicer visuals:
  - Additionally, identify report and page level filters to align the page with other dashboard visuals



In this example, the USA sales management team wants to easily view and filter high level sales measures and KPIs by sales region and time period, without having to navigate from the dashboard to detail level reports.

## Create and publish to the dashboard

1. Create a Power BI report and connect to the published dataset in the app workspace of the dashboard.
2. Construct the report page according to the requirements, and to maximize the user experience with the dashboard.
3. Publish the report with the live page to the App Workspace in Power BI.
4. In the workspace in Power BI, open the published report in the given workspace and select the page to pin.



The Sales Live Page Analysis Report in the Adventure Works Enterprise Workspace

5. Click the **Pin Live Page** option in the preceding image and choose either a new or existing dashboard.
6. Pin any additional visuals from other pages of this report or other reports to the dashboard.

## Refine dashboard layout

1. Position the live page at the top left of the dashboard, with separate visual tiles to the right:



Dashboard with pinned live page and additional visuals

- If a title and refresh message is included in the page, per this example, then those values can be excluded in tile details.
2. Switch from web view to phone view at the top right of the dashboard:
    - Given the size of the live page, it may be necessary to hide this tile from Phone view or move it to the bottom
    - The user of the dashboard can interact with the region and date slicers and other visuals on the live page



The ellipsis in the top right of the live page tile includes a go to report link to access all pages of the live page report. Modifications to the underlying report will be reflected in the pinned live page. The other visual tiles link to their report pages.

## How it works...

### Live page slicers

- The sales region slicer at the top has the orientation option in the formatting pane set to horizontal
- The date slicer is based on the calendar date from the date dimension and uses the relative slicer visual option

## Deploying content from Excel and SSRS to Power BI

Dashboards in Power BI can consolidate much more than Power BI report visuals. Microsoft Excel objects, such as pivot tables, charts, and workbook ranges, and SSRS report items can also be pinned as dashboard tiles. This integration with Power BI allows teams to utilize existing reports and skills, and to leverage the unique capabilities of these tools as part of overall BI solutions.

In this recipe, a pivot table and pivot chart from an Excel workbook are integrated into an existing Power BI dashboard for the Australian sales team. Additionally, an SSRS report item is also pinned to this dashboard. For more advanced integration examples, see Chapter 13, *Integrating Power BI with Other Applications*.

## Getting ready

1. Install Power BI Publisher for Excel:
  - See Chapter 1, *Configuring Power BI Development Tools*, for details on this process
  - If the Power BI tab is not visible in Excel, check that the COM Add-in in Excel options is visible and enabled
2. Configure Report Server for Power BI:
  - Open reporting services configuration manager
  - Click on the **Power BI Integration** tab and select **Register with Power BI**

## How to do it...

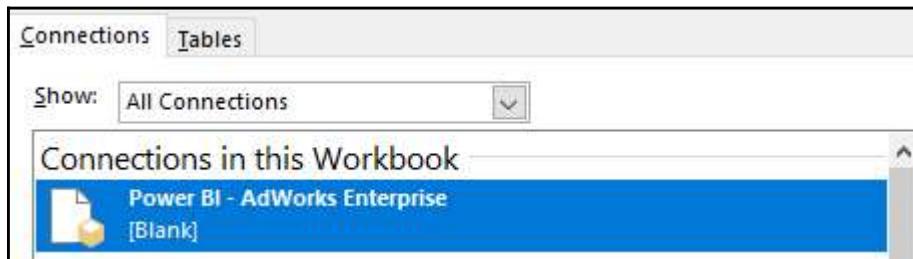
### Publish and pin excel objects

1. Open a new Excel workbook and select **Connect to Data** from the Power BI tab (Power BI Publisher for Excel).
2. Choose the Power BI workspace and dataset to be used as the source for the Excel report.
3. Click **Connect** and a blank pivot table will be created with the field list exposed on the right.
4. Create and format a pivot table. Apply a slicer based on a dynamic date column, such as current year.
5. Create additional slicers, such as Sales Territory Country = Australia:
  - To create a slicer, right-click a column in the field list and select **Add as Slicer**



Using slicers rather than the filters field well for pivot tables allows for a better presentation on the dashboard tile.

6. Select any cell outside of the pivot table, and from the Insert tab, click **PivotChart**:
  - Select **Use an External Data Source** from the **Create PivotChart** dialog. Click the Power BI connection.



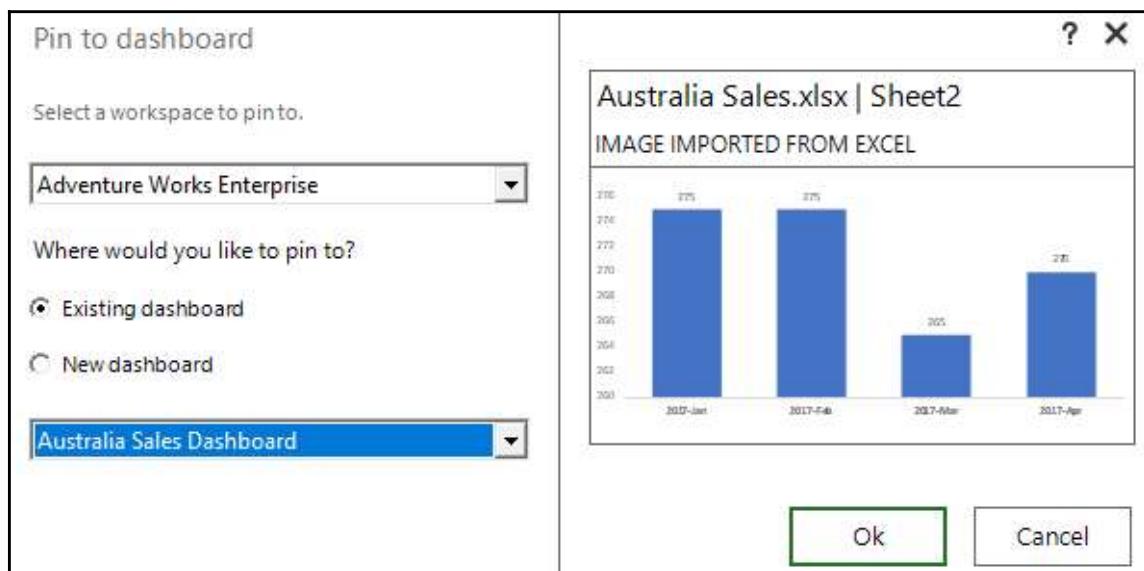
Existing Connections Window in Excel: Connection to the Power BI dataset

7. Select the Power BI connection from the existing connections window and click on **Open**.
8. Build and format the pivot chart. With the pivot chart selected, click on **Filter Connections** from the **Analyze** tab:
  - Ensure that the same slicers filtering the pivot table are also filtering the pivot chart



Click on **Options** under the **Analyze** tab for each new report object and enable evaluate calculated members from OLAP server in filters on the Totals and Filters tab. Additionally, from the Display tab of Options, enable show calculated members from OLAP server for both objects. It is necessary to apply the same slicer to both the chart and the table.

9. Save the Excel report file to either a OneDrive for business folder or at a secure network location.
10. Select the chart and click **Pin** from the Power BI tab. Choose the workspace and the dashboard, and then click **OK**.



Pinning a chart from an Excel workbook to a Power BI dashboard in an App Workspace of the Power BI Service

11. Now select the full range of cells of the pivot table and pin this to the dashboard as well.
12. In the Power BI service, navigate to the workspace and dashboard to adjust the size and layout of the tiles.
13. Optionally, adjust the tile details for the pinned Excel tiles, such as title and subtitle.



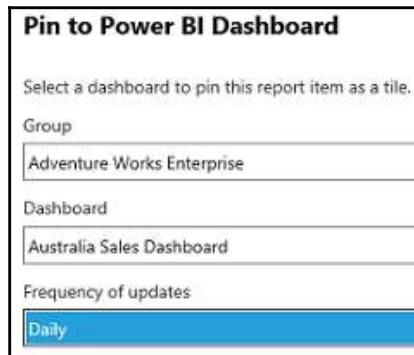
Excel online does not currently support the refresh of external data connections. Therefore, though it's possible to publish the workbook from Excel to Power BI and then pin items from the workbook report in the Power BI service, once published, the workbook would not be refreshed. By pinning items directly from the Excel workbook to the dashboard, the connection to the dataset hosted in the Power BI service must be periodically refreshed and the **Pin Manager** dialog in the Power BI Publisher for Excel can be used to update pinned tiles.

To avoid this manual and local refresh process, Excel report visuals can be built on top of an Excel data model, and this Excel workbook can be published to the Power BI Service. Published workbooks, containing data models, can be configured for scheduled refresh in the Power BI Service, and their dependent reports will be updated to reflect these refreshes.

## Pin SSRS report items

1. Create or identify the SSRS report to support the dashboard.
2. Publish this report to the SSRS report server or open this report on the report server.
3. From the report server browser window, click the Power BI icon and sign in with the appropriate Power BI account.

4. Click on the SSRS report item to pin:



Pin to Power BI from SSRS 2016

5. From the Pin dialog, choose the workspace and dashboard to pin the item to.
  - The update frequency creates an SSRS subscription to keep the tile updated in Power BI

In this example, the reseller freight expense tile (lower left) is from an SSRS report. The Australia headcount chart and Australia sales to plan pivot table tiles are both from an Excel workbook.



Excel objects and SSRS report items pinned to a Power BI dashboard



You can pin charts, gauge panels, maps, and images from SSRS reports to Power BI dashboards provided these items are within the report body (not page header or footer). You cannot currently pin tables, matrix, or list report items from SSRS reports.

## Adding data alerts and email notifications to dashboards

Alerts can be configured on the tiles of dashboards to provide notification that a specific condition has been met. Alert rules and options can be managed in the Power BI service, and notifications can be limited to the notification center in Power BI or shared via email. Data-driven alerts enhance the value of Power BI dashboards, as they immediately bring attention to significant events or outcomes as the dataset supporting the dashboard tile is refreshed.

In this recipe, an alert is configured on a KPI visual represented in a dashboard tile. Additionally, an example is provided of automating email delivery of notifications via Microsoft Flow based on a Power BI data alert.

### How to do it...

#### Configure data alert

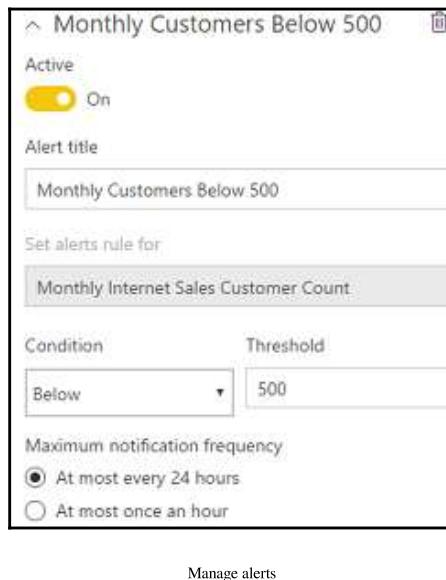
1. Open the App Workspace in Power BI and select the dashboard containing the tile to be used for the alert.
2. Click on the ellipsis in the top right of the tile and select the bell icon to open the **Manage Alerts** window.



Alerts can only be configured on dashboard tiles of standard gauge, KPI, and card visuals, and they only work with numeric data types. Custom visuals, streaming data tiles, and date datatypes are not currently supported.

Only the user who configures the alert can see the alerts in the Power BI Service.

3. Click on the **Add alert rule** button and enter a title for the alert that describes the measure and the condition.
4. Set the **Condition** and **Threshold** parameters for the alert:



Manage alerts

5. Repeat this process for other tiles in the dashboard or other dashboards.



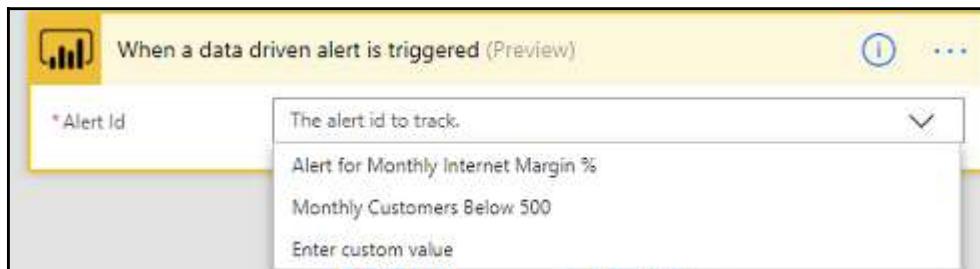
Multiple alerts can be configured for the same dashboard tile with each alert having a separate condition and/or threshold, such as a maximum and a minimum accepted value. Click the Gear icon in Power BI and select **Settings** to access all the alerts configured across the workspaces.

Data alerts can also be set and viewed in the Power BI mobile apps.

## Automate email notification

1. Open Microsoft Flow in Office 365.
2. Enter **Power BI** in the search bar and select **Trigger a flow with a Power BI data-driven alert**:
  - Click on **Continue** to use this template. Ensure your Power BI account is associated with the Power BI trigger.

3. In the **Flow name** input box at the top, provide a descriptive title (for example, email sales team based on margin % alert).
4. From the **Alert ID** dropdown, choose the specific Power BI data alert to trigger the Flow and click on **New Step**.



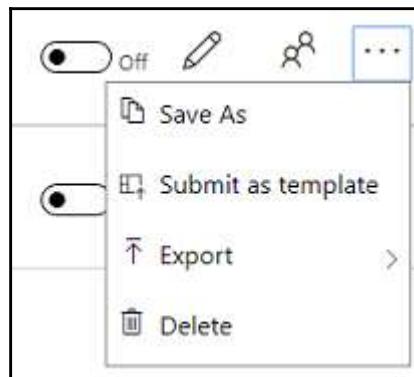
Associating the Power BI alert with the flow

5. Click on **Add an Action** and then select **Office 365 Outlook**. Click on the **Send an email** action available for this service:
  - Customize the send an email action to specify users or groups and what alert content and text to include:



Outlook send an email action step in MS Flow

6. Click on **Create flow** from the top menu. The flow will be saved and will begin working immediately. Click on **Done**.
  - From My flows, this new alert flow can be saved as to create additional flows which leverage the same triggers, logic, and actions



Save as option in my flows

- Save as will create a copy of the flow and add it to the my flows page



Given the value of alerts to stakeholders and the low effort required in creating and managing alerts and notifications in Power BI and Flow, dedicated alert dashboards can be developed to reduce the amount of analysis and monitoring required.

## How it works...

Power BI evaluates the alert rule when the dataset supporting the dashboard is refreshed.

If the alert is triggered, an icon and message will appear in the notification center in the Power BI service containing a link to the dashboard tile configured for the alert.



Alert notification from notification center in Power BI

Notifications are also visible in Power BI mobile. For example, a notification message is visible on the Apple watch.

[Buy the full eBook](#)

