**Power Bi Assignment 5**

1. Explain DAX.

Ans:- DAX is a collection of functions, operators, and constants that can be used in a formula, or expression, to calculate and return one or more values. Stated more simply, DAX helps you create new information from data already in your model.

It’s easy to create a workbook and import some data into it. You can even create PivotTables or PivotCharts that display important information without using any DAX formulas. But, what if you need to analyze critical sales data across several product categories and for different date ranges? Or, you need combine important inventory data from several tables in different data sources? DAX formulas provide this capability and many other important capabilities as well. Learning how to create effective DAX formulas will help you get the most out of your data. When you get the information you need, you can begin to solve real business problems that affect your bottom line. This is Business Intelligence, and DAX will help you get there.

Data Analysis Expressions (DAX) is a formula expression language used in Analysis Services, Power BI, and Power Pivot in Excel. DAX formulas include functions, operators, and values to perform advanced calculations and queries on data in related tables and columns in tabular data models.

This article provides only a basic introduction to the most important concepts in DAX. It describes DAX as it applies to all the products that use it. Some functionality may not apply to certain products or use cases. Refer to your product's documentation describing its particular implementation of DAX.

**Calculations**

DAX formulas are used in measures, calculated columns, calculated tables, and row-level security.

### Measures

Measures are dynamic calculation formulas where the results change depending on context. Measures are used in reporting that support combining and filtering model data by using multiple attributes such as a Power BI report or Excel PivotTable or PivotChart. Measures are created by using the DAX formula bar in the model designer.

A formula in a measure can use standard aggregation functions automatically created by using the Autosum feature, such as COUNT or SUM, or you can define your own formula by using the DAX formula bar. Named measures can be passed as an argument to other measures.

When you define a formula for a measure in the formula bar, a Tooltip feature shows a preview of what the results would be for the total in the current context, but otherwise the results are not immediately output anywhere. The reason you cannot see the (filtered) results of the calculation immediately is because the result of a measure cannot be determined without context. To evaluate a measure requires a reporting client application that can provide the context needed to retrieve the data relevant to each cell and then evaluate the expression for each cell. That client might be an Excel PivotTable or PivotChart, a Power BI report, or a table expression in a DAX query in SQL Server Management Studio (SSMS).

Regardless of the client, a separate query is run for each cell in the results. That is to say, each combination of row and column headers in a PivotTable, or each selection of slicers and filters in a Power BI report, generates a different subset of data over which the measure is calculated. For example, using this very simple measure formula:

Total Sales = SUM([Sales Amount])

When a user places the TotalSales measure in a report, and then places the Product Category column from a Product table into Filters, the sum of Sales Amount is calculated and displayed for each product category.

Unlike calculated columns, the syntax for a measure includes the measure's name preceding the formula. In the example just provided, the name **Total Sales** appears preceding the formula. After you've created a measure, the name and its definition appear in the reporting client application Fields list, and depending on perspectives and roles is available to all users of the model.

The **Adventure Works DW 2020** Power BI Desktop sample model is designed to support your DAX learning. The model is based on the [Adventure Works data warehouse sample](https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure#data-warehouse-downloads) for AdventureWorksDW2017—however, the data has been modified to suit the objectives of the sample model.

The sample model does not contain any DAX formulas. It does however support hundreds or even thousands of potential calculation formulas and queries. Some function examples, like those in CALCULATE, DATESBETWEEN, DATESIN PERIOD, IF, and LOOKUPVALUE can be added to the sample model without modification. We're working on including more examples in other function reference articles that work with the sample model.

These functions are new or are existing functions that have been significantly updated.

[Aggregation functions](https://learn.microsoft.com/en-us/dax/aggregation-functions-dax) - These functions calculate a (scalar) value such as count, sum, average, minimum, or maximum for all rows in a column or table as defined by the expression.

[Date and time functions](https://learn.microsoft.com/en-us/dax/date-and-time-functions-dax) - These functions in DAX are similar to date and time functions in Microsoft Excel. However, DAX functions are based on the datetime data types used by Microsoft SQL Server.

[Filter functions](https://learn.microsoft.com/en-us/dax/filter-functions-dax) - These functions help you return specific data types, look up values in related tables, and filter by related values. Lookup functions work by using tables and relationships between them. Filtering functions let you manipulate data context to create dynamic calculations.

[Financial functions](https://learn.microsoft.com/en-us/dax/financial-functions-dax) - These functions are used in formulas that perform financial calculations, such as net present value and rate of return.

[Information functions](https://learn.microsoft.com/en-us/dax/information-functions-dax) - These functions look at a table or column provided as an argument to another function and returns whether the value matches the expected type. For example, the ISERROR function returns TRUE if the value you reference contains an error.

[Logical functions](https://learn.microsoft.com/en-us/dax/logical-functions-dax) - These functions return information about values in an expression. For example, the TRUE function lets you know whether an expression that you are evaluating returns a TRUE value.

[Math and Trig functions](https://learn.microsoft.com/en-us/dax/math-and-trig-functions-dax) - Mathematical functions in DAX are similar to Excel's mathematical and trigonometric functions. However, there are some differences in the numeric data types used by DAX functions.

[Other functions](https://learn.microsoft.com/en-us/dax/other-functions-dax) - These functions perform unique actions that cannot be defined by any of the categories most other functions belong to.

[Parent and Child functions](https://learn.microsoft.com/en-us/dax/parent-and-child-functions-dax) - These functions help users manage data that is presented as a parent/child hierarchy in their data models.

[Relationship functions](https://learn.microsoft.com/en-us/dax/relationship-functions-dax) - These functions are for managing and utilizing relationships between tables. For example, you can specify a particular relationship to be used in a calculation.

[Statistical functions](https://learn.microsoft.com/en-us/dax/statistical-functions-dax) - These functions calculate values related to statistical distributions and probability, such as standard deviation and number of permutations.

[Table manipulation functions](https://learn.microsoft.com/en-us/dax/table-manipulation-functions-dax) - These functions return a table or manipulate existing tables.

[Text functions](https://learn.microsoft.com/en-us/dax/text-functions-dax) - With these functions, you can return part of a string, search for text within a string, or concatenate string values. Additional functions are for controlling the formats for dates, times, and numbers.

[Time intelligence functions](https://learn.microsoft.com/en-us/dax/time-intelligence-functions-dax) - These functions help you create calculations that use built-in knowledge about calendars and dates. By using time and date ranges in combination with aggregations or calculations, you can build meaningful comparisons across comparable time periods for sales, inventory, and so on.

1. Explain datasets, reports, and dashboards and how they relate to each other?

Ans :-

## What are datasets?

A Power BI Dataset is a series of Power Query queries that have been shaped in a DAX model. Each dataset can combine different files, database tables and online services all into one tabular model.  In our cookie analogy, these are all different “ingredients”.

Unlike SSRS, a dataset in Power BI does not represent a single table or query of data. A dataset should be considered more like a “flavor” of data used to accomplish a specific type of reporting: financial, operational, HR, etc. So in our analogy, the dataset is the “raw dough”.

So in Power Query, you are going to have a set of queries which each combine a data source with a usually linear set of transformations.

Then, in DAX, you are going to take each of those outputs and combine them into a model. This consists of defining relationships between the outputted tables and adding business logic via calculated columns and measures.

## What are reports?

A power BI report is a series of visualizations, filters and static elements on a canvas. Power BI reports are saved as a single PBIX file and connect to a single dataset. Remember, a Power BI dataset can have many data sources.

Each report can have multiple sheets, just like an Excel workbook. In our analogy, this is us placing our “cookies” on multiple “cookie sheets” making one big batch, all of the same “flavor”.

### One report per dataset

A quick aside to something that used to confuse me. In most cases, a report and a dataset are going to have a one to one relationship. A dataset can have one report and a report can have one data set.

Recently this has changed, however. A while back, they added the ability to use an [existing dataset as a data source for a report](https://docs.microsoft.com/en-us/power-bi/desktop-report-lifecycle-datasets). and at Ignite they announced the ability to share datasets outside of the app workspace they were made in.

That being said, while you are still learning Power BI, it’s easier to remember that in many cases, your dataset and your report are going to have a one-to-one relationship and be tightly linked.

## What are dashboards?

In Power BI, dashboards are a way of pulling together visualizations from various reports. When you think dashboard, you are probably thinking something like Microsoft’s [definition](https://docs.microsoft.com/en-us/power-bi/consumer/end-user-dashboards): “A Power BI **dashboard** is a single page, often called a canvas, that uses visualizations to tell a story. Because it is limited to one page, a well-designed dashboard contains only the most-important elements of that story.”

However, if you look at the report example above, it probably fits that definition. It is not a Power BI Dashboard. In Power BI, a dashboard is tool for pinning visuals from different reports and other sources of data.

In my opinion, a Power BI Dashboard is as much a tool for **organization** and **navigation**, as it is for actual reporting. I think that’s the real value add with Power BI dashboards.

Dashboards are created from multiple datasets or reports. Dashboards always concentrate on building insights into the data by using graphs, attractive visuals, charts, etc. Reports are not concentrated on the visualization part of the data rather it looks to create summary pages.

**Dashboards vs reports**

| **Item** | **Dashboards** | **Reports** |
| --- | --- | --- |
| Pages | Dashboards are created on only one page | Can be created in one or more pages |
| Data sources | Dashboards are created from multiple datasets or **reports**. | Reports are created from a single dataset |
| Visualization | Dashboards always concentrate on building insights into the data by using graphs, attractive visuals, charts, etc. | Reports are not concentrated on the visualization part of the data rather it looks to create summary pages. |
| Available in Power BI Desktop | Dashboards can not be created in Power BI Desktop | Reports can be built and viewed in Power BI Desktop. |
| Filters and Slicers | You cant add Slicers and Filters as Dashboards are limited to a single page. | In reports, we can use many different ways to filter, highlight, and slice. |
| User Interactivity | **Dashboards** allow a user to pin visuals from different **reports** and datasets onto a single canvas, making it simple to group what’s essential to the user. | **Reports** are more focused on being able to visualize and apply transformations to a single dataset. |
| Favourite | Yes. Can set multiple dashboards as *favourites*. | Yes. Can set multiple reports as *favourites*. |
| Q&A Feature | Yes | Yes, provided you have edit permissions for the report and underlying dataset. |
| Alerts | In dashboards, alerts to emails are created, when specific condition or criteria is met or limit crossed. | We can’t create Alerts in **Reports.** |
| Subscribe | Yes. Can subscribe to a dashboard. | Yes. Can subscribe to a report page. |
| See underlying dataset tables and fields | In **dashboards**, you can’t see the underlying dataset tables but can export data. | While in **Reports** you can see a dataset under the **Data** tab in Power BI Desktop. |

1. How reports can be created in power BI, explain two ways with Navigation of each.

Ans:-

1. Create a data source
2. Run a transformation (Optional)
3. Create a query or data model
4. Pick a visualization
5. Select fields
6. Arrange visualization on dashboard

To illustrate, let’s move through each of these steps. First, we create a data source. That connects to a file, like a .csv, or a database.

Next, you have the option to run a **transformation**. In our example, we use financial data. We will:

* Apply a filter to select only negative values (payments)
* Drop and rename columns
* Optionally apply a function, such as an aggregation

Step 3 is the natural result of step 2, because you have built up a query in stages.

Alternatively, at this point, you could create a data model. For example, if you have sales and inventory movements in two data sources you can model that. You would create a model to show the common element between tables: product number. (But in the example we’re using, we only have a single data source.)

In step 4, you create a visualization. In this example, we will have a table of transactions. A table is a row and column display. We will also have a single **card** (like a text box) to show a single number, the maximum transaction amount.

Next, we’ll pick fields from step 4. Finally, in the last step, we’ll position the visualizations on the dashboard.

1. How to connect to data in Power BI? How to use the content pack to connect to google analytics? Mention the steps.

Ans:-

To connect to Google Analytics data:

1. Select **Get Data** from the **Home** ribbon in Power BI Desktop. Select **Online Services** from the categories on the left, and then select **Google Analytics**. Then select **Connect**.
2. If this is the first time you're getting data through the Google Analytics connector, a third-party notice is displayed. Select **Don't warn me again with this connector** if you don't want this message to be displayed again. Then select **Continue**.
3. To connect to Google Analytics data through the legacy "Universal Analytics" API, select Implementation 1.0. To connect to Google Analytics data through the new Google Analytics Data API with support for Google Analytics 4, select Implementation 2.0 (Beta).
4. To sign in to your Google Analytics account, select **Sign in**.

In the **Sign in with Google** window that appears, provide your credentials to sign in to your Google Analytics account. You can either supply an email address or phone number. Then select **Next**.

Enter your Google Analytics password and select **Next**.

When asked if you want Power BI Desktop to access your Google account, select **Allow**.

Once you've successfully signed in, select **Connect**.

Once the connection is established, you’ll see a list of the accounts you have access to. Drill through the account, properties, and views to see a selection of values, categorized in display folders.

You can **Load** the selected table, which brings the entire table into Power BI Desktop, or you can select **Transform Data** to edit the query, which opens Power Query Editor. You can then filter and refine the set of data you want to use, and then load that refined set of data into Power BI Desktop.

1. How to import Local files in Power BI? Mention the Steps.

Ans:-

If you want to import Analytics data manually through Power BI, follow these instructions.

1. In Power BI, click **Get Data** in the lower left screen.
2. Under **Import or Connect to Data** > **Files**, click **Get**.

Click Local File.

**If you want to import Analytics data manually through Power BI, follow these instructions.**

1. In Power BI, click Get Data in the lower left screen.
2. Under Import or Connect to Data > Files, click Get.
3. Click Local File.
4. Choose which file to upload and click Open.
5. Click Upload under Upload your Excel file to Power BI.

1. In Power BI visualization, what are Reading View and Editing view?

Ans:-

Editing view is used by report designers, who create the reports and share them with you. Reading view is your way to explore and interact with reports created by colleagues.

## Reading view

There are two modes for interacting with reports in the Power BI service: Editing view and Reading view. If you are a business user, then you are more likely to use Reading view to consume reports created by others. Editing view is used by report designers, who create the reports and share them with you. Reading view is your way to explore and interact with reports created by colleagues.