Architecture Design
BIRD STRIKE ANALYSIS

Written By	Onkar Arjunwade
Document Version	1.0
Last Revised Date	18 MARCH 2024

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS
1.0	18 MARCH 2024	Onkar	

Reviews:

VERSION	DATE	REVIEWER	COMMENTS

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY	APPROVED BY	COMMENTS

Contents:

1. Introduction

1.1 What is Architecture design document?

Any software needs the architectural design to represents the design of software. IEEE defines architectural design as "the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system." The software that is built for computer-based systems can exhibit one of these many architectures.

Each style will describe a system category that consists of :

• A set of components (e.g.: a database, computational modules) that will perform a function

required by the system.

• The set of connectors will help in coordination, communication, and cooperation between

the components.

• Conditions that how components can be integrated to form the system.

• Semantic models that help the designer to understand the overall properties of the

System.

1.2 Scope

Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

2. Power BI Architecture
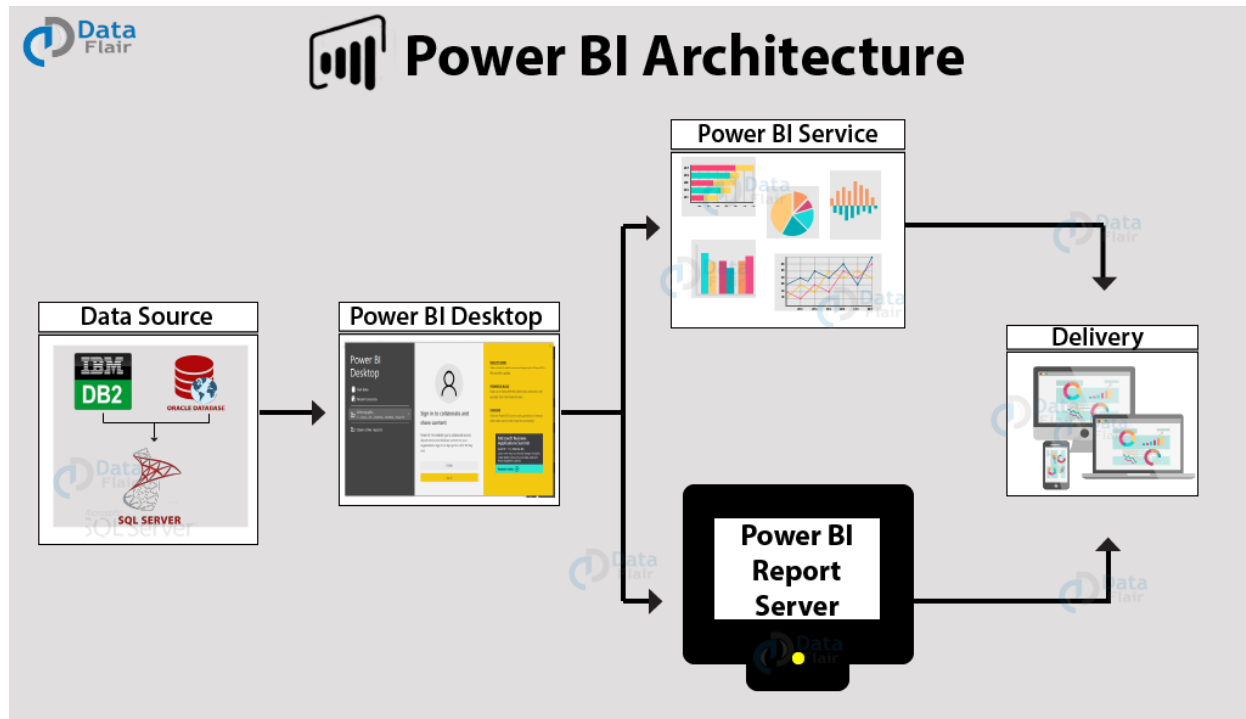
2.1 Data Sources

An important component of Power BI is its vast range of data sources. You can import data from files in your system, cloud-based online data sources or connect directly to live connections. If you import from data on-premise or online services there is a limit of 1 GB. Some commonly used data sources in Power BI are:

- Excel
- Text/CSV
- XML
- JSON
- Oracle Database
- IBM DB2 Database
- MySQL Database
- PostgreSQL Database
- Sybase Database
- Teradata Database
- SAP HANA Database
- SAP Business Warehouse server
- Amazon Redshift

- Impala
- Google BigQuery (Beta)
- Azure SQL Database
- Salesforce Reports
- Google Analytics
- Facebook
- GitHub

2.2. Power BI Desktop

Power BI Desktop is a client-side tool known as a companion development and authoring tool. This desktop-based software is loaded with tools and functionalities to connect to data sources, transform data, data modeling and creating reports.

You can download and install Power BI Desktop in your system for free. Using Power BI Desktop features, one can do data cleansing, create business metrics and data models, define the relationship between data, define hierarchies, create visuals and publish reports.

2.3 Power BI Service

Power BI Service is a web-based platform from where you can share reports made on Power BI Desktop, collaborate with other users, and create dashboards.
It is available in three versions:

- Free version
- Pro version
- Premium version

Power BI Service is also known as, "Power BI.com", "Power BI Workspace", "Power BI Site" and "Power BI Web Portal". This component also offers advanced features like natural language Q&A and alerts.

2.4 Power BI Report Server

The Power BI Report Server is similar to the Power BI Service. The only difference between these two is that Power BI Report Server is an on-premise platform. It is used by organizations who do not want to publish their reports on the cloud and are concerned about the security of their data.

Power BI Report Server enables you to create dashboards and share your reports with other users following proper security protocols. To use this service, you need to have a Power BI Premium license.

2.5 Power BI Gateway

This component is used to connect and access on-premise data in secured networks. Power BI Gateways are generally used in organizations where data is kept in security and watch. Gateways help to extract out such data through secure channels to Power BI platforms for analysis and reporting.

2.6 Power BI Mobile

Power BI Mobile is a native Power BI application that runs on iOS, Android, and Windows mobile devices. For viewing reports and dashboards, these applications are used.

2.7 Power BI Embedded

Power BI Embedded offers APIs which are used to embed visuals into custom applications.

3. Working of Power BI Architecture

Now that we have understood the individual components of Power BI, let us learn how do all of these components work in tandem. We will understand the Power BI architecture with the help of this diagram.

POWER BI ARCHITECTURE
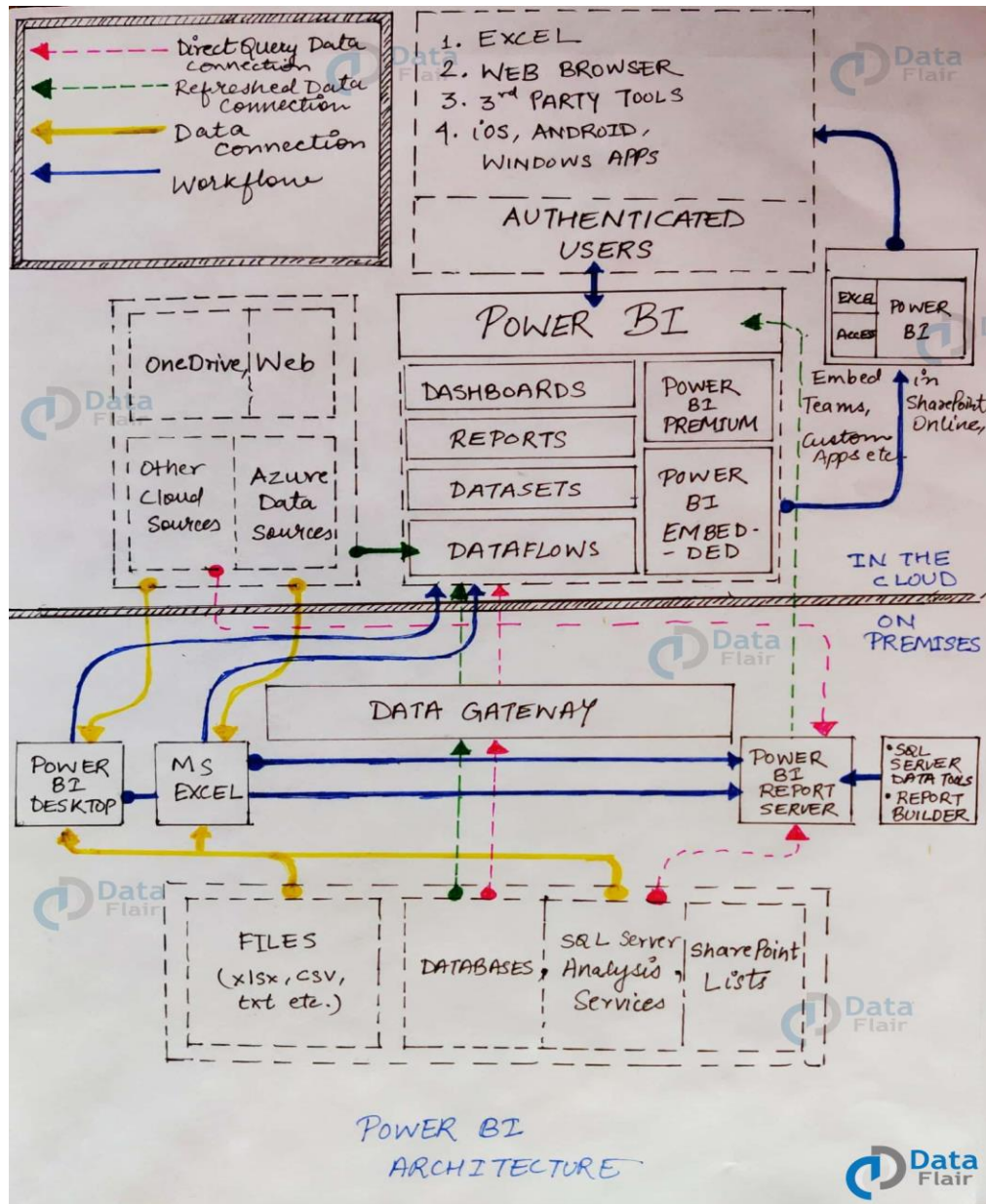
If you look closely, the diagram has numbering done on each component in the architecture. Also, note that the lower half part is the on-premise part and the upper half part depicts the on-cloud services.

To begin with, what forms the starting point or source of all the data flowing into Power BI components are the data sources. Power BI has the get data feature using which you can connect to different kinds of data sources like *files, on-premise or on-cloud databases, direct connections,* etc. Data connections are established from these data sources to authoring tools such as Power BI Desktop.

3.1 On-Premise

Power BI Desktop is a companion development, authoring, and publishing tool. You can import data from data sources to Power BI Desktop and use it to create reports and then publish them on a Power BI Service or Power BI Report Server.

You can also publish Excel workbooks directly using Power BI Publisher for Excel to the Power BI Report Server. The SQL Server Data tools and Report Publisher help in *creating datasets, KPIs, mobile reports, paginated reports,* etc. The reports from all kinds of reports are published to the Power BI Report Server from where they are distributed to the end-users.

3.2 On-Cloud

An important component in Power BI architecture is the Power BI Gateway. The Power BI Gateway acts as a secure channel to transport data from on-premise data sources to on-cloud apps or sites.

On the cloud side of the architecture, resides a lot of components. Like a complete Power BI suite having *dataflows, datasets, dashboards, reports, Power BI Embedded, Power BI Premium,* etc. You can embed your reports and dashboards into *Teams, SharePoint, custom applications,* etc. There are on-cloud data sources as well that connects to Power BI tools via direct connections.

At last, there is a layer of authenticated users who share the published reports and dashboard and collaborate with one another to make educated decisions based on the insights. There are different kinds of users who consume Power BI reports and dashboards and connect through *web browsers, Excel, third-party tools, and mobile devices* (iOS, Windows, Android apps).

4. Power BI Service

As we have learned in the earlier sections, all the reports that you create in Power BI Desktop are published on a cloud platform known as Power BI Service.

Users can access the reports and dashboards from Power BI Service using client platforms like websites, mobile devices, etc. This means that every client who wants to access content created on Power BI needs to interact with Power BI Service. And so, we must take a look under the hood and learn how Power BI Service works.

Power BI Service's architecture consists of two parts:

- A front end
- A back end

4.1 Front End cluster

The front end also called the web front-end cluster acts as an intermediary between clients and the back end. The front-end services are used for establishing an initial connection and authenticating clients using Azure Active Directory. The Azure Active Directory stores user identities.

Along with this, Azure Traffic Manager is used to direct user requests to the nearest data center after authentication. Once a client/user is authenticated, the **Azure Content Delivery Network (CDN)** distributes static Power BI content/files to users.
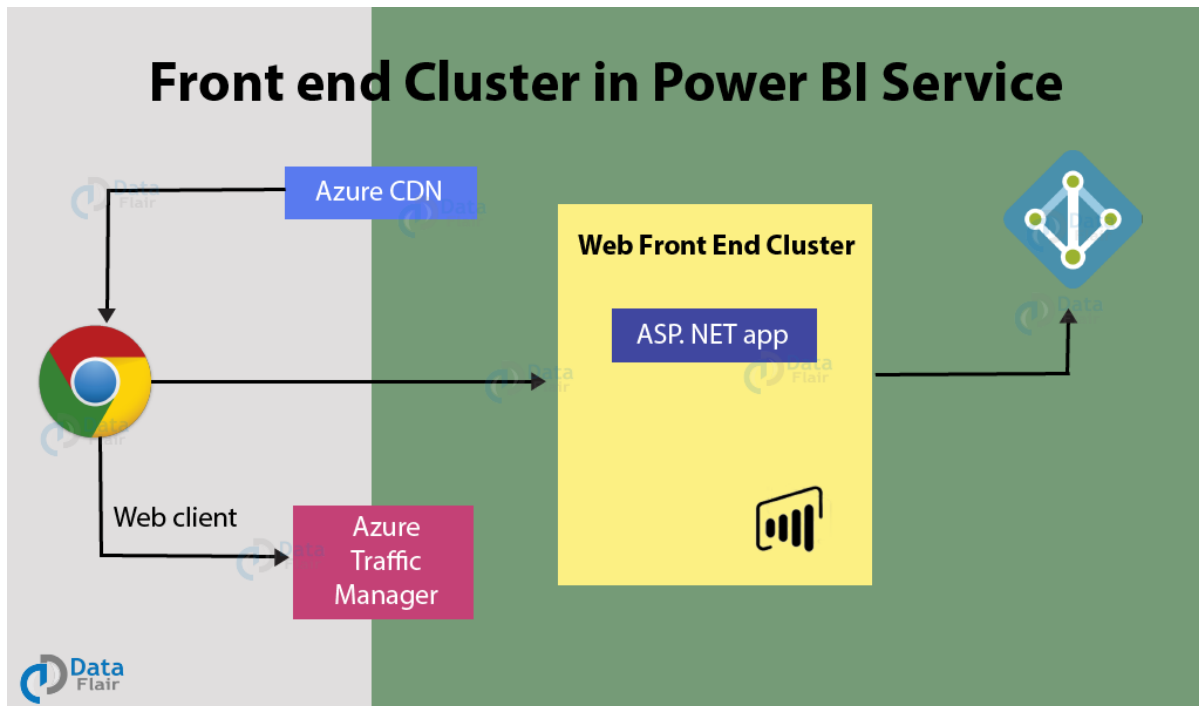
4.2 Back End Cluster

The Power BI services at the back end take care of visualizations, datasets, storage, reports, data connections, data refreshing, and other interactions with Power BI. At the back-end, a web client has only two direct points of interaction, **Azure API Management**, and **Gateway Role**. These two components are responsible for load balancing, authentication, authorization, routing, etc.


Back end Cluster in Power BI Service
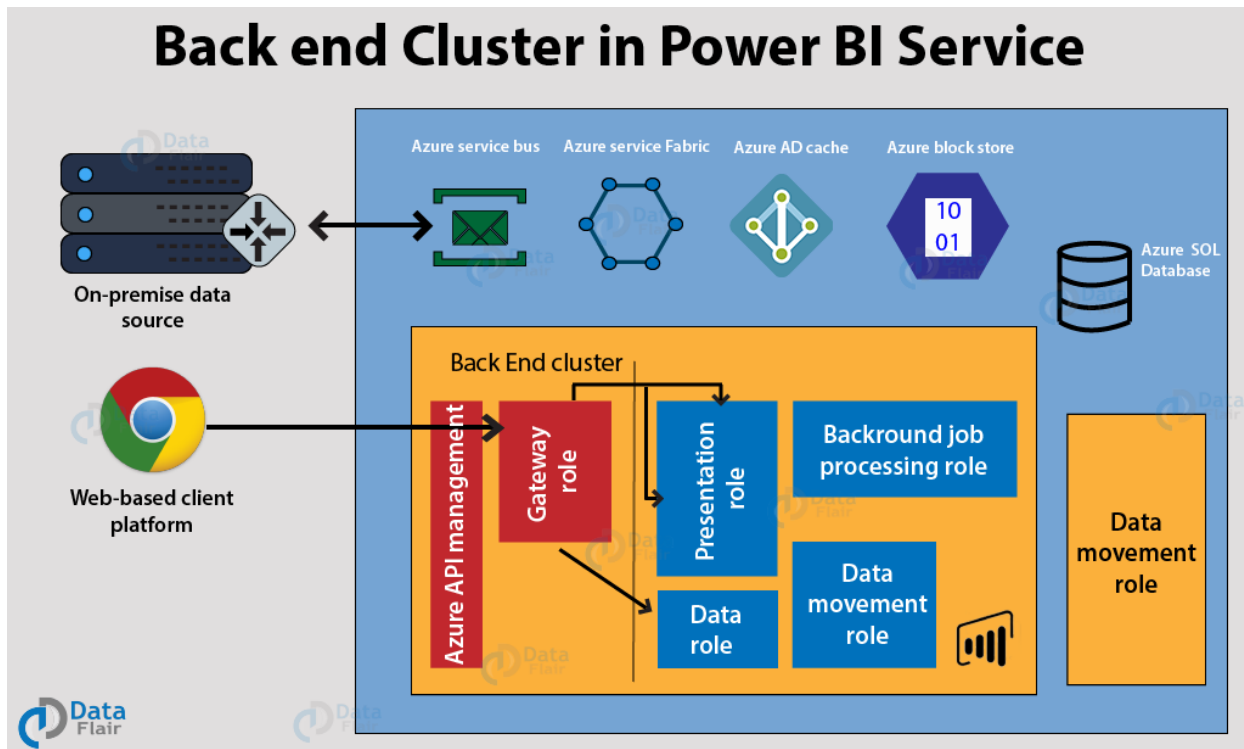
4.3. Working of Power BI Service

- Power BI stores its data in two main repositories; **Azure block storage** and **Azure SQL database**. Azure block storage stores the datasets uploaded by users and all the metadata and system-related data is stored in the Azure SQL database.
- After Azure API Management authenticates a user request, it is sent to the Gateway Role. The Gateway Role processes the requests and directs them to suitable components like *Presentation Role, Background Job Processing Role, Data Role, and Data Movement Role.*
- For instance, the Presentation Role handles all the visualization related queries like for dashboards and reports.
- For all the data related queries, the request is sent by the Gateway Role to the Data Role or Data Movement Role.
- Power BI Service back end uses Azure Service Bus to connect on-premise data sources with the cloud. Azure Service Bus receives all the requests to fetch data from the on-premise data source. Then it processes the request and executes the query on the on-premise data source to retrieve data from it to the cloud service.
- The Azure Service Fabric manages all the microservices and components associated with running Power BI.
- Azure AD Cache helps in real-time reporting using the data stored in the in-memory of the Power BI system.

6. Power BI Components

The components of Power BI are shown as below:
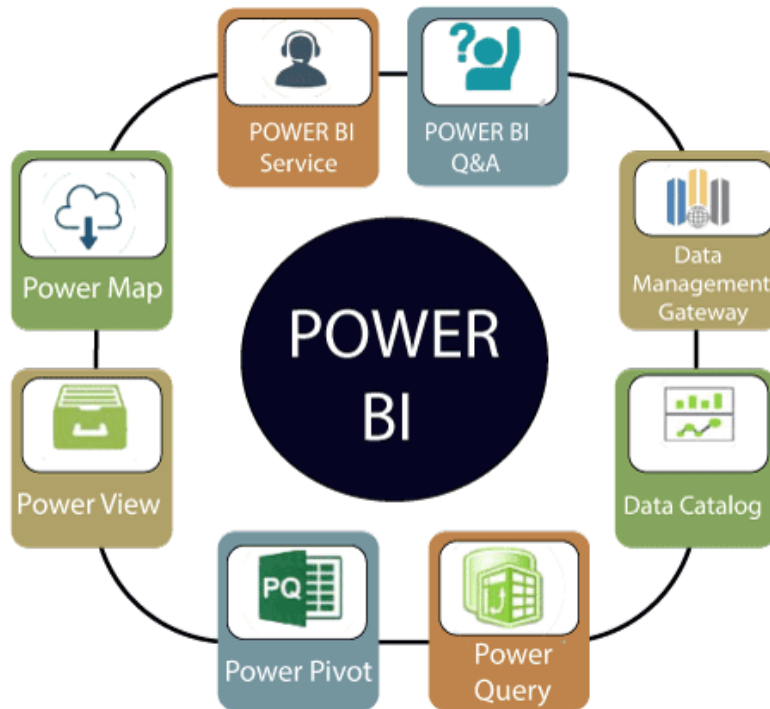
1. Power Query: It is used to access, search, and transform public and internal data sources.

2. Power Pivot: Power pivot is used in data modeling for in-memory analytics.

3. Power View: By using the power view, you can analyze, visualize, and display the data as an interactive data visualization.

4. Power Map: It brings the data to life with interactive geographical visualization.

5. Power BI Service: You can share workbooks and data views which are restored from on-premises and cloud-based data sources.

6. Power BI Q&A: You can ask any questions and get an immediate response with the natural language query.

7. Data Management Gateway: You get periodic data refreshers, expose tables, and view data feeds.

8. Data Catalog: By using the data catalog, you can quickly discover and reuse the queries.

7. Difference between Dashboards and Reports

| Capabilities	Dashboards	Reports
Pages	It has only one page.	It can have one or more pages.
Data Sources	It has one or more reports and datasets per dashboard.	It has only a single dataset per report.
Pinning	It can pin existing visualizations only from the current dashboard to your other dashboards.	It can pin visualizations to any of the dashboards. And also, can pin entire report pages to any of the dashboards.
Filtering	It can't filter or slice.	It has many different ways to filter, highlight, and slice.
Feature	It can set one dashboard as the featured dashboard.	It cannot create a feature report.
Set alerts	No	Yes, it can set alerts.
Subscribe	We can't subscribe to a dashboard.	We can subscribe to report pages.
Available in Power BI desktop	No	Yes, it can create and view reports in desktop.
Change visualization type	No, if a report owner changes the visualization type in the report, the pinned visualization on the dashboard does not update.	Yes, it can change the visualization type.

Dashboard and reports both terms are used interchangeably, but they are not synonymous.

8. Power BI DAX

DAX (Data Analysis Expressions) is a formula expression language. It can be used in different BI and visualization tools. DAX is also known as function language in which the full code is kept inside a function. DAX programming formula contains two data types such as Numeric and Other.
Numeric includes currency, integers, and decimals, where Other includes string and a binary object.

Power BI DAX

- Syntax
- Context
- Functions
- Calculated columns & Measures
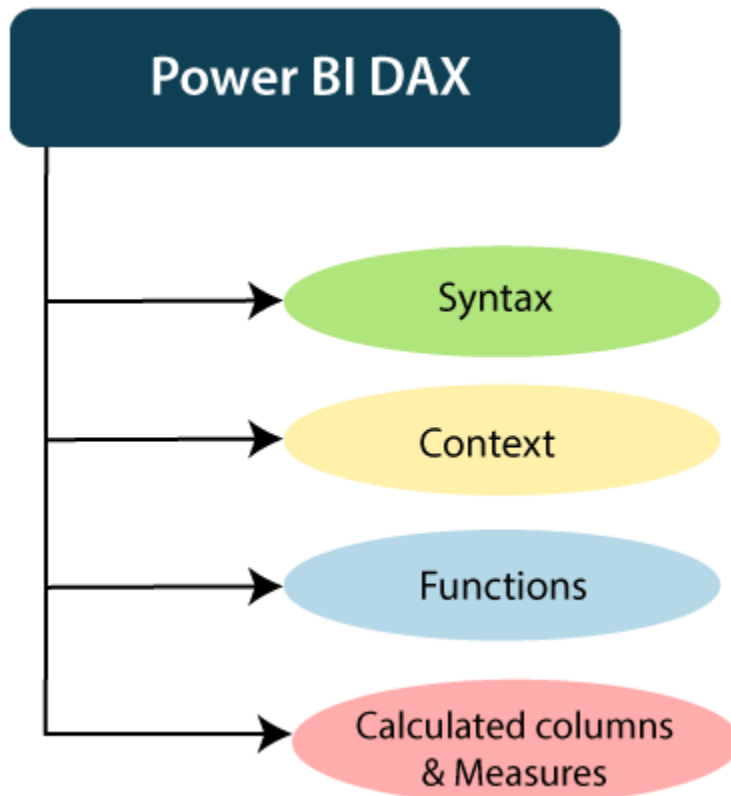
How does it work?
For understanding the Power BI DAX, it has main three fundamental concepts such as:
  o Syntax
  o Context
  o Functions

1. Syntax

the syntax consists of various components that make up a formula.

1. Total Sales = SUM (Sales [SalesAmount])
- o Total Sales is the measure name.
- o The equal sign (=) operator indicates the beginning of the formula.
- o The DAX function SUM adds up all the numbers in the Sales [SalesAmount] column.
- o Parentheses () surround an expression containing one or more arguments. And all function requires at least one argument. An argument passes a value to a function.
- o The reference table Sales.
- o The referenced column [SalesAmount] in the Sales table. With this argument, the SUM function knows on which column to aggregate a SUM.

2. Context

Context is one of the essential concepts of DAX. It is categorized into two parts; Row context and Filter context.

The **Row-Context** is the easiest thought of as the current row. It applies whenever a formula has a function which uses the filters to identify a single row in a table.

The **Filter context** is a little more challenging to understand than the Row context. You can most easily think of the Filter-Context as one or more filters applied in a calculation. The Filter-Context doesn't exist in the Row-context's stead. Instead, it uses in addition to the former. Look at the following DAX formula.

3. Functions

Functions are predefined and ordered formula. They can perform calculations using arguments passed on to them. These arguments can be text, numbers, logical values, or other functions.

Types of Functions

Here are some important DAX functions:

1. Aggregate Functions

- o **MIN**
  This DAX function returns the minimum numeric value in a column, or between the two scalar expressions.
  Syntax
  MIN(<column>)
- o **MAX**
  This DAX function returns the maximum value in a column, including any logical values and numbers represented as text.
  Syntax
  MAX(<column>)
- o **AVERAGE**
  This DAX function returns the arithmetic mean of the values in a column.
  Syntax
  AVERAGE(<column>)
- o **SUM**
  This DAX function adds all the numbers in a column.
  Syntax
  SUM(<column>)

2. Count Function

- **COUNT**

  This DAX function is used to return the count of items in a column. If there are multiple numbers of the same thing, this function will count it as separate items and not a single item.

  Syntax

  COUNT(<column>)

- **DISTINCTCOUNT**

  This DAX function is used to return the distinct count of items in a column. If there are multiple numbers of the same thing, this function will count it as a single item.

  Syntax

  DISTINCTCOUNT(<column>)

3. Date time Function

- **DATE**

  This DAX function returns the specified date in Date-Time format.

  Syntax

  DATE(<year>, <month>, <day>)

- **HOUR**

  This DAX function returns the specified hour as a number from 0 to 23 (12:00 A.M. to 11:00 P.M.).

  Syntax

  HOUR(>datetime<)

4. Logical Function

- **AND**

  This DAX function performs logical AND(conjunction) on two expressions. For AND to return true, both conditions specified have to be fulfilled.

  Syntax

  AND(<logical argument1>,<logical argument2>)

- **OR**

  This DAX function performs logical OR(disjunction) on two expressions. For OR to return true, either of the two conditions specified has to be fulfilled.

  Syntax

  OR(<logical argument1>,<logical argument2>)

- **NOT**

  This DAX function performs logical NOT (negation) on given expression.

  Syntax

  NOT(<logical argument>)

5. Text function

- **CONCATENATE**

  This DAX function joins two text strings into one text string.

  Syntax

  CONCATENATE(<text1>, <text2>)

- **FIXED**

  This DAX function rounds a number to the specified number of decimals and returns the result as text.

  Syntax

  FIXED(<number>, <decimals>, <no_commas>)

- **REPLACE**

  This DAX function replaces part of a text string, based on the number of characters you specify, with a different text string.

  Syntax

  REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)

4. Calculated Columns and Measures

The Power BI DAX formulae are used in calculations, in Measures and Calculated Columns.

- Calculated Columns

When you create a data model on the Power BI Desktop, you can extend a table by creating new columns. The content of the columns is defined by a DAX expression, evaluated row by row or in the context of the current row across that table.

- Measures

There is another way of defining calculations in a DAX model, useful if you need to operate on aggregate values instead of on a row-by-row basis. These calculations are measures. One of the requirements of DAX is a measure that needs to be defined in a table. However, the action does not belong to the table. So, you can move a measure from one table to another one without losing its functionality.