

Power Business Intelligence (BI)[p

Power BI:

Power BI is a Data Visualization and Business Intelligence tool that converts data from different data sources to interactive dashboards and BI reports.

Introduced by Microsoft and released in 2015.

Reporting:

- Rendering the business data in structured format using tables/Matrix's etc.

Visualization:

- Graphical representation of the business data using charts, Graphs, Maps etc.

Note: There are more than 200 Visualizations in Power BI.

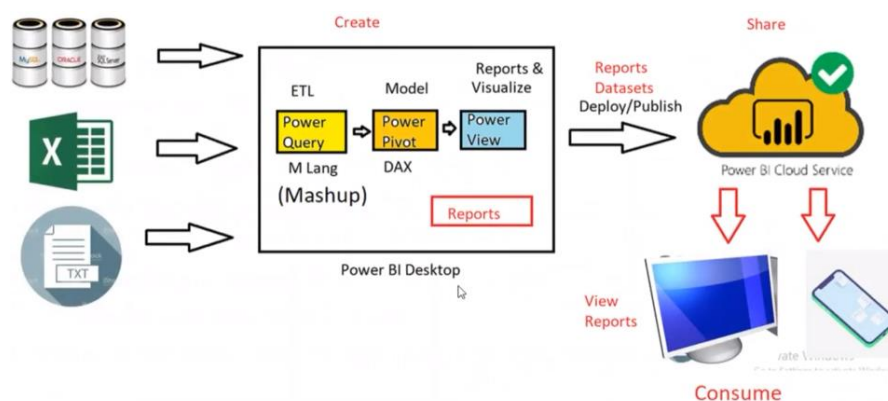
Dashboard:

- A Power BI dashboard is a single page, often called a canvas, which tells a story through visualizations.

BI VS AI:

- Report the Past data is called Business Intelligence.
- Predicting the Future is called as Artificial Intelligence.

Power BI Cloud Architecture:



Power BI Tools/Products:

- Power BI Desktop – Developers
 - Power Query - ETL(Mashup Language)- convert data from one format to another
 - Power Pivot – Stores the data in RAM (Data Analytical Expressions(DAX) Language)
 - Power View – here we can create Reports and Visualizations.
- Power BI Service (Azure/Cloud)-90% Use this
 - Deploying the reports to end user.
- Power BI Report Server(On Premises)- 10%use this
 - Deploying the reports to end user.
- Power BI Mobile – Mobile App(IOS, Android, Windows)

Power Query:

- In Power Query, we perform ETL operations.
- Here, we use Mashup Language which is Case Sensitive.
- It transforms the data from one format to another.

Power Pivot:

- It will compress 1/10 of the data and stores in RAM. It is called as Columnar Database.
- Data Modeling and Relationships
- Here, we use Data analytical Expression (DAX) Language.

Power View:

- We create reports, Dashboards and visualizations.

Building Blocks in Power BI:

- Data Sets
 - Power Query and Power Pivot
 - Tables, Relationships, Data Modeling, DAX, Enhancements
- Visualizations
 - Render the data in the forms of Images
- Reports
 - ETL+ Data Model+ Enhance Data Model+ Visualizations etc.
- Dashboards
 - High level information from Reports
 - Create dashboard from multiple reports.
- Tiles (Each object in the report)
 - Logo
 - Visualizations
 - Textboxes Etc.

Data Types:

- Numerical
- Text/String
- Date and Time
- Boolean(T/F)
- Map→ Latitude and longitude
- Decimal
- Binary

Note: PBI reads the first 1000 rows and assigns the data types

Filters:

Filters are used to load subset of data from data sources.

There are two types of filters

- Basic filter
- Advanced filter

We never use basic filters in power BI because M Lang issue. And we use always advanced filter.

Software development life cycle (SDLC):

- Scope → What need to be done
- Requirements → SRS document
- Analysis and design
- Coding (Developers/ Programmer/Engineer develop the code)
- Testing (Testers will test the code/app)
- Building

Column Transformations:

- Remove columns/remove other columns
- Choose column/Go to column
- Name/rename columns
- Reorder column/Sort column
- Add column/Custom column → Add year, Add Monthly salary
- Column from examples
- Duplicate column
- Split column
- Merge column
- Pivot → Rows to columns
- Unpivot → columns to rows
- Transpose → Rows to columns and columns to rows
- Replace values
- Remove empty.

Transform → it applies transformations on the existing/replacing column.

In add Column → It creates a new column and apply transformations

Row Transformations:

- Header rows/use first row as header
- Use headers as first row
- Keep top rows/Keep bottom rows/Keep range – complete table
- Keep errors – Complete table/selected Columns
- Remove error rows - Complete table/selected Columns
- Remove blank rows - Complete table
- Remove Empty rows - Complete table
- Keep Duplicates - Complete table/selected Columns
- Remove alternative rows
- Remove duplicates
- Remove blank rows
- Remove errors
- Group rows.

Joins:

Joins are used to retrieve/get the data from multiple tables

Note: To join the data we must have a common column in the both tables.

Maximum 2 can be joined in the Power BI.

There are 6 types of join

1. Inner Join/Simple Join/Natural Join /Join
2. Outer Join
 - a. Left Outer Join
 - b. Right Outer Join
 - c. Full Outer join
3. Cross join (Cartesian Product)
4. Self-join
5. Anti-left join (Left outer unmatched)
6. Anti-Right join (Right Outer unmatched)

Append Queries (Union All):

- Combine data from multiple data sets.
- Check order of columns in all the data sets
- Corresponding columns data types should be similar
- Keep duplicates by defaults
- Same no of columns in all the data sets
- Else, will get null values.

Case 1:

- Use folder with multiple files
 - Place Homogeneous files in the folder.

Note: Heterogeneous files in the folder will **NOT** work.

Case 2:

- Use folder with Excel with many sheets.

Union:

- To remove duplicates
- Sort the data
- Slow

Union All:

- It will keep the duplicates.
- Doesn't sort the data.
- Faster than Union.

Refresh Preview:

Refresh only one table data.

Refresh All:

Refresh all tables

Custom Functions:

- Create function using M code.
- Use it anywhere
- M code is case sensitive
- Ex: Add 2 numbers
 - Code: =(X as number, Y as number)=>(X+Y)

Power BI Service

Power BI Service:

Power BI service is a cloud-based business analytics and data visualization service that enables anyone to visualize and analyze data with greater speed, efficiency, and understanding.

- Azure Cloud based server
- To share and deploy the reports
- Manage the reports

There are two types of Licenses

- **Power BI Pro**
 - Small teams (30 users)
 - Per user \$10 per month
 - 1 GB Dataset
 - 8 times Refresh per day
 - 2 months free
- **Power BI Premium**
 - Bigger Organizations (1000 users)
 - \$5000
 - 10GB Dataset
 - 48 times refresh per a day.

Scenario:

- 20 Tables
- 10 reports
- Size is 20GB
 - Cannot store in Power Pivot
- What can be done
 - Store 10 tables and create 5 reports
 - Store another 10 tables and create another 5 reports
- We must know
 - What datasets we must create
 - What reports we must have on the given dataset.

Dashboard:

- Detail level info called as Reports
- High level info pages call it as Dashboards
- Dashboards will be created in PBI Service
- Dashboard is read only view of high level information
- Dashboards can be created on multiple reports and multiple datasets
- Cannot add slicer to the dashboard
 - Since dashboard will have multiple datasets
 - They do not have relationships
- Pin to a dashboard
 - Can add slicer in this page.
- When we share the dashboard with user, they will have access to underlying reports and datasets as well.
- Dashboards can be created using
 - Same dataset
 - Or different dataset

Data Refresh:

- To get the updated data from the data sources.
- Refresh Option available in Power Bi Desktop level

Data Gateways:

- It is a software
- It acts bridge between premise data sources to PBI service datasets
- It moves data from data sources to datasets in PBI Service
- Connectivity modes to get the data to Power BI Service
 - Import
 - Direct query
 - Live

Where to install the data gateway app

- Install in OLTP/Data Sources
- SQL Server
- Oracle
- Shared Folders etc.

Note:

- No gateway is required for Azure data sources
- Gateway is required only for on-Premises to cloud

Types of Gateways:

There are 2 types

- Personal gateways(Local)
 - One user can use it.
 - It supports only for import/Schedule refresh
- Standard/Enterprise gateways(Global)
 - It recommended
 - Share with Others
 - Re-usability with many users
 - Supports Import/Schedule refresh
 - Supports Direct query/Live Query.

Data Connectivity Modes:

There are three types of connectivity modes

- Import/Cached
- Direct Query (Databases)
- Live (Cube)

Import/Cached

- Less than 10GB of data
- Loads Tables Schema and data.
- You can see the report view, Data view, Relationships view

Direct Query (Databases)

- When we have more than 10GB of data
- Loads only Tables schema and it shows up to date data.
- Here You can see report view, Data view

Live (Cube)

- When we have TB of data
- Inherits the cube Structure (Tabular and Multi-Dimensional cube)
- We do not create any data model structure with in Power BI
- Here, you can see report view only.

When we refresh:

- Pro License
 - Truncate and Load the full data to PBI.
- Premium License
 - Incremental Data Loading.

Actions On the workspace:

- Update Workspace (Change the name of the workspace)
- Add members to the workspace
- Add/delete/update members from the workspace
- Add content (Dataset, Report and Dashboard) to the workspace
- Share the content with end user
- View the content in the work space.

Workspace access permissions roles:

- Admin
 - Rename the workspace
 - Add the members and delete the members
 - Add/delete/update the contents
 - Read the content and share to end users
- Member
 - Add/delete/update the contents
 - Read the content and share to end users
- Contributor
 - Add/delete/update the contents
 - Read the content
- Viewer
 - Read the content only.

Share the reports/dashboards:

- Basic Sharing/Direct sharing
 - End users will get an email with a link.
 - We can send one report/dashboard at a time.
 - End user cannot edit the report.
- Using apps
 - One app for one workspace
 - Here, we can send multiple reports and multiple dashboards.
 - Go to workspace.
 - Click on create app.
 - Publish the app.

Row Level Security:

Row-level security (RLS) with Power BI can be used to restrict data access for given users. Filters restrict data access at the row level, and you can define filters within roles.

There are two types of RLS

- Static Row level Security
- Dynamic Row level Security

UserPrincipalname() is a DAX function used in dynamical row level security.

Power Pivot with DAX Functions

Power Pivot:

Power Pivot is an in-memory data modeling component that provides highly compressed data storage and extremely fast aggregation and calculation.

- It will compress 1/10 of the data and stores in RAM. It is called as Columnar Database.
- It is 10 times faster than row databases like SQL Server, Oracle.
- Here, we use Data analytical Expression (DAX) Language.
- In Power Pivot we do data modeling in tables.

Data modeling (relationships):

- Giving relationships to retrieve data from multiple tables.
- 1 to 1
 - 2 tables
 - 2 PK
 - 2 FK (1 FK is enough – Any PK can become a FK in the opposite tables)
 - Ex: One country has only one president and a president represents one country only.
- 1 to M
 - PK of parent table can only become FK in the child Table.
 - 2 Tables
 - 2 PK
 - 1 FK (Parent PK to Child FK)
 - EX: One department has many employees.
- M to M
 - 3 Tables
 - 2 PK
 - 2 FK in bridge table

Note:

- SQL/DB does not support m to m relationship
- Whereas Power BI supports m to m (recently added).

ERD → Entity Relational Diagram

- We ERD to design OLTP

Data Analytical Expressions (DAX):

- It is a functional Language
- Create new column using existing column
- Using DAX we can create calculations
- We use Existing columns, Constants, Operators, Functions
- Calculations are done for each row.
- Space is needed in RAM

Revised Salary:

Increment the salary 1000 for DeptID=10 and 2000 for others

RevisedSalary = IF (Employee [DeptID]=10,Employee[Salary]+1000,Employee[Salary]+2000)

Tax = salary*18/100

Use Single Quotes:

- When the table name has space.
- Ex: 'Order Sales'[SalesAmount]

Features in Power Pivot:

- New column
 - When we need a calculation on each row
 - It will be added as part of table
 - Table size will increase
 - When we refresh new column values are re-calculated
 - Name must be unique in a table level.
- New Measure
 - When we need an aggregation and works on group of rows/all rows
 - E.g.: Percentages, Ratios, Aggregations, etc.
 - It will not be added in a table, shown in visualizations
 - Table size will not increase, stores formula
- New Table
- Quick Measure
 - Do not want to write the formula when you need
 - Power BI writes its own formula
 - Do not use this Function.

Measure:

- Measure is not a static value.
- Measure is a Calculation/Formula.
- Measure will not be added in the table.
- Measure does not need any space.
- Measure is displayed in visualizations.
- Measures are 2 types
 - Implicit
 - No formula
 - Basic Aggregations
 - Explicit
 - With Formula
 - Complex aggregations

DAX Operators:

- Arithmetical
- Comparison
- Logical
- Concatenation

Arithmetical Operators:

- Add(+)
- Subtract (-)
- Multiply(*)
- Division(/)

Comparison Operators:

- Equal to (=)
- Greater than(>)
- Greater than or Equal to (>=)
- Less than (<)
- Less than or equal to (<=)
- Not equal to (<>)

Logical Operator:

- && (And) – Double ampersand
- || (or) – Double pipe symbol.

- IN { }
- Eg: when the city = "BLR" and DeptID =10 then increase salary to 1000

IncreSalary = IF(Emp[DeptID]=10 &&Emp[City]="BLR",Emp[Salary]+1000,Emp[Salary])

Text Concatenation Operator:

- Ampersand (&)

DAX Functions:

Logical Functions (Condition Based)

- IF()
- IF. Else()
- Nested IF() → If within IF
 - **EmpType = IF(Emp[Salary]<2000,"Low",IF(Emp[salary]>=4000,"High",Medium))**
- True() or False() → Returns true or false
- Switch () → use instead of multiple if conditions
 - **RevSal = Switch(True(),Emp[Salary]<2000,"Low", Emp[salary]>=4000,"High",Medium)**
- IFError() → returns 0 when the error occurs else the value
 - **IFERROR(Sales/Qty,0)**

Text/String Functions:

- Len() → Counts no of letters in the columns
 - **Length =Len("Go online trainings")**
 - Result - 19
- Concatenate() → combine columns
 - **EIDwithName =Concatenate(Concatenate(Emp[EID], "-"), Emp[Ename])**
 - Result – 1-Srinivas
- Left() → To get letters from left side
 - **Left2Letter = Left("Srinivas",2)**
 - Result – Sr
- Right() → To get the letters from right side
 - **Right2Letters = Right("Srinivas",2)**
 - Result – AS
- Mid() → To get the middle text from string
 - **MIdletters = Mid("Srinivas",2,3)**
 - Result – RIN
- Upper() → To convert Lower to Uppercase
 - **UpperName= Upper("Srinivas")**
 - Result – SRINIVAS
- Lower() → To convert Upper case to lower case
 - **LowerName = Lower("SRINIVAS")**
 - Result – srinivas
- Trim () → Remove spaces from left and right and in between.
 - **InstName=Trim(" Go Online Trainings ")**
 - Result – Go Online Trainings
- Substitute () → It substitute the existing with new text
 - **SubstituteName = Substitute("Srinivas","Sri","Sree")**
 - Result – Sreenivas
 - Conditioned based.
- Replace() → replace the existing text with new text
 - **ReplacePhno=Replace(Emp[Phno],4,5,"*****")**
 - Result – 949*****61
 - Non conditional based.
- Blank()
 - Returns Blank
 - **DailyWage = IF(Emp[NoOfDays]=0,Blank(),Emp[Salary]/ Emp[NoOfDays])**
 - Result – 40

- ConcatenateX() → Iterate Function
 - To concatenate complete data
 - CitywiseEmps = ConcatenateX(Emp,Emp[EName],",")
 - Result – BLR – Rajesh,Kadir,Suresh,Naresh

Date and Time Functions:

- Year () → to get the year from the date field.
- Quarter () → to know the quarter from the date field.
- Month() → to get the month no from the date field
- Week () →to get the week no from the date field.
- Day() → to get the day
- EoMonth () → to know the last date of the month.
- DateDiff () → to find the difference between two dates.
- Calendar () → we need to pass the start date and End date to create calendar.
- CalenderAuto () → It Scan the date fields in the source and create the calendar from start date to End Date.
- Date() → It converts values to date
- Format() → we can format full date to Month, Day, Year
- Today() → To assign date and no time
- Now () → to assign Date and current time.

To get Year wise sales Growth:

We must have Dim.Date Table

- It must have Unique values
- Continue Values
- No nulls.

Math Functions:

- Int()
 - Convert into Integer
 - It removes decimals
 - Eg: SalaryInt = INT(Emp[Salary])
- Currency()
 - Converts into money
 - Eg: SalesAmt = Currency(Sales[Salesamount])
- Round()
 - It rounds the number to the given number of digits
 - If the decimal point is >=5 then it rounds the number to the next digit.
 - Eg: RoundSalary = Round(Emp[Salary],0)
- Roundup()
 - It will take the number to next level
 - Eg: RoundSalary = RoundUp(Emp[Salary],0)
 - Eg: RoundSalary = RoundUp(Emp[Salary],1)
- RoundDown()
 - Round down to nearest integer with decimal places
 - It is same as INT
 - Can Pass decimal or Integer
- Divide()
 - Divide(Numerator/Denominator,[AlternateResult])
 - Eg: Divsalary = Divide(Emp[Salary],Emp[Bonus],0)
 - If you not give 0, then shows null.
- Even()
 - Roundup the number to the next even number nearest integer
- Odd()
 - Roundup the number to the next odd number nearest integer

- Power()
 - Power(2,2)
- Sqrt()
 - Square root of the Number
- Factorial()
 - Factorial(5) → Output: 5*4*3*2*1
- Sign()
 - Returns 1 when the number is positive
 - Return 0, When the number is 0
 - Return -1, when the number is negative
 - Eg: ProfitSign = Sign(Sales[Profit])
- Requirement
 - Show no of positive profit orders
 - Show no of negative profit orders
 - Show no of zero(0) profit orders

DAX Statistical Functions:

- Sum()
 - Adds all the values in a given column
 - SUM(ColName)
 - Eg: SumOfSalary = Sum(Emp[Salary])
- Sumx(Table,Expression)
 - Returns sum of an expression evaluated values for row in table.
 - Sumx(Table,Expression)
 - SumofNetSalary = SUMx(Emp,Emp[Salary]+Emp[Bonus])
- Min()
 - Gets the min value in a given column
 - MinSalary = MIN(Emp[salary])
- MINx()
 - Returns the min value of expression evaluated for each
 - MinNetSalary = MINx(Emp, Emp[Salary]+Emp[Bonus])
- Max()
 - Gets the max value in a given column
 - MaxSalary = Max(Emp[salary])
- MAXx()
 - Returns the max value of expression evaluated for each
 - MaxNetSalary = MAXx (Emp, Emp[Salary]+Emp[Bonus])
- Count()
 - Counts no of rows in a given column
 - Count(ColName)
 - NoOfOrders = Count(Sales[SalesOrdername])
 - NoOfEmps = Count(Emp[Bonus])
- COUNTx()
 - Returns no of rows of an expression evaluated for each in table
 - NoOfOrders = Countx(Emp,Emp[Salary]+Emp[Bonus])
- Avg()
 - Gets the Average value in a given column
 - AvgOfSalary= Avg(Emp[Salary])
- Avgx()
 - Returns the Average value of expression evaluated for each
 - AvgOfSalary= Avg(Emp,Emp[Salary]]+Emp[Bonus])
- CountRows() → to count number of rows in a table
 - NoOfRowsEmp = CountRows(Emp)
- CountBlank() → Get the count of blank values in a column
 - NoOfBlankValues = CountBlank(Emp[Bonus])
- Rankx()
 - It assigns rank number based on given column
 - RankNo = Rankx(Emp,Emp[salary],,,Dense)

- Summarize()
 - It gives you aggregated values to speed up the PBI report.
 - Requirement
 - Get order year Sales Amount
 - Create an aggregated table and use it, Instead of aggregating values on the fly.

OrderYearwiseSales=Summarize(Sales.Sales[OrderYearNO],”TotalSales”,Sum(Sales[salesAmount]))

Filter Functions:

1. Calculate()

- Aggregation based on a filter
- Eg: Get total sales amount for black color products

BlckClrTotalSales = Calculate(Sum(Sales[Salesamount]),Products[Color]=”Black”)

- Eg: Get total sales amount for black color products

BlckandBlueClrTotalSales = Calculate(Sum(Sales[Salesamount]),Products[Color]=”Black” || Products[Color]=”Blue”)

- Filter is executes first and then it will sum
- Comma is AND Operator.

2. All()

- Percentage of sales using all()
- All() reads grand total.
- Formula = Sales/SumofSales*100

%Sales = SUM(sales[SalesAmount])/Calculate(Sum(Sales[SalesAmount]),All(Sales))*100

3. UseRelationship()

- Between 2 tables, Cannot have more than ONE Active relationship
- To calculate a measure based on in-active relationship

Solution 1:

- Get the DimDate Table once more
- DisAdv
 - It will occupy more space in Power Pivot

Solution 2:

- Use USERELATIONSHIP()
- Create a Measure

ShipDateSale = Calculate(SUM(Sales[SalesAmount]),UseRelationship(Sales[ShipDate],DimDate[FullDate]))

4. Related()

- Report requirement: EName, DeptID, DeptName
- When we create a visualization from multiple tables
- PBI Joins the tables on the fly, Hence it is slow.
- Instead of this
 - Get the DeptName to the Emp table
 - Create new column in Emp Table

DepName=related(Dept[DeptName])

DisAdv

When we refresh the data it may take little more time.

One more report required:

- Need a column with concatenate

EmpNamewithDeptName=Emp[EName]&"-"&Related(Dept[DeptName])

Time Intelligence Functions:

- It means doing calculations over time or date periods
- It needs date column to perform operations
- This date column must contains
 - Continuous values
 - No Nulls
 - Unique values
- To make intelligence functions work properly, mark it as Date table

1. TotalMTD → Month to Date

- Create a Measure

TotalMTDSales = TotalMTD(Sum(Sales[SalesAmount]),DimDate[Fulldate])

2. TotalQTD → Quarter to Date

- a. Create a measure

TotalQTDSales = TotalQTD (Sum(Sales[SalesAmount]),DimDate[Fulldate])

3. TotalYTD → Year to Date

- a. Create a measure

TotalYTDSales = TotalYTD (Sum(Sales[SalesAmount]),DimDate[Fulldate])

4. PreviousDay

PDSales = Calculate(Sum(Sales[SalesAmount]),PreviousDay(DimDate[Fulldate]))

5. PreviousMonth

PDSales = Calculate(Sum(Sales[SalesAmount]),PreviousMonth(DimDate[Fulldate]))

6. PreviousQuarter

PDSales = Calculate(Sum(Sales[SalesAmount]),PreviousQuarter(DimDate[Fulldate]))

7. PreviousYear

PDSales = Calculate(Sum(Sales[SalesAmount]),PreviousYear(DimDate[Fulldate]))

8. NextDay

9. NextMonth

10. NextQuarter

NextQtrSales = Calculate(Sum(Sales[SalesAmount]),NextQuarter(DimDate[Fulldate]))

11. NextYear

NextYrSales = Calculate(Sum(Sales[SalesAmount]),NextYear(DimDate[Fulldate]))

12. SamePeriodLastYear

SamePeriodLastYear = Calculate(Sum(Sales[SalesAmount]),SamePeriodLastYear (DimDate[Fulldate]))

13. %YoYGrowth

- Year on year growth
- Formula $\equiv (\text{Current Year Sales} - \text{Previous year sales}) / \text{Current year sales} * 100$

$$\%YoYSalesGrowth \equiv ([\text{SumOfSales}] - [\text{PySales}]) / [\text{SumOfSales}]$$

Power View

Power View:

Power View is a data visualization technology that lets you create interactive charts, graphs, maps, and other visuals that bring your data to life.

- To create visualizations using Datasets.

Power View Has

Fields Pane:

- Tables are organized in sorting order
- Search option: For column search

Visualization Pane:

- Common visualizations are displayed
- 200 visualizations we can use

Filters Pane: To show subset of data

Menu/Ribbon: it has all options to drag and drop

Views:

Report view, Data view and Model view

Report Canvas:

- Where we created the visualizations.

Pages:

To add pages, duplicate the page, delete page etc.

Visual Interactions/Edit Interactions:

- Highlight Action
- Filter Action
- None
- Whenever, we select anything, first priority is highlight action
- By default, either highlight or filter.
- Tables must have relationships to enable interactions.

Power View Filters:

To show subset of data

There are three types of filters in Power view

- Visual level Filter → if you want to filter the data on the visualization level
- Page level Filters/Filters on this page → Entire the page we can filter with this
- Report level Filters/Filters on all pages → We can use to filter all pages
- Drill through Filters

Filter Sub types

- Basic filtering
- Advance filtering
- Top n filtering
- Relative date filtering

Drill through Filters:

- Filter the data across the pages, we can use drill through filters
- Jumping from one page to another using filter.
- Create Parent and child report
- Eg: Country wise sales (Parent page) → Stage level sales(Child page)
- Drill through reports
 - Cross Report – Off(default)
 - Can be tested in power BI service only
 - Keep all filters – On (default)

Cross report drill through:

- Cross report must be on both reports
- Cross report must be on in PBI service as well
- Both reports must be in same workspace

Hierarchy:

- Ordering similar dimensions from Largest to smallest.
- Hierarchies are used to create drill down reports.

There are two types of Hierarchies

- System defined hierarchy
 - Created by the power BI automatically
 - Automatically created on Date and Time columns in Power BI
 - These hierarchies Cannot be modified
- User defined hierarchy
 - We must have related dimensions
 - User created these hierarchies
 - E.g.: Category, Subcategory, Product
 - Eg2: Country, State, City

Drill Down:

- Jumping reports from one level report to next level report.

Types of Visualizations:

- Basic data visualizations
- Visuals of filtering
 - Interactive reports/Dynamic reports
- Visualizing categorical data
- Visualizing Trend data/Time series data
- Visualizing KPI Data
- Visualizing Tabular data
- Visualizing geographical data
- Leveraging PBI Custom visuals

Slicer Visualization:

1. Purpose: To filter the dimension data in the report page by the end uses.
2. Field Wells: Where to Place Dimensions and facts(1Column)
3. Formatting Options

For Date Column:

- Between (Default)
- Before → Can change only end date
- After → Can change only the start date
- List
- Dropdown

- Relative date → to get last months, years data.

For Text Column

- List (Default)– Small amount of data
- Dropdown –Large amount data
- Search – Enable the search when we have large data

For Number Column:

- Between (Default)
- List
- Dropdown

Note: Always we can use single value

Categorical Visualization:

Pie Chart:

- To show categorical data.
- When we have less categorical data

Field wells

- Where to place dimension and facts
- There are four
 - Legend→If you drag more than one Dimension then it will create hierarchy and drill down report
 - Values→ Facts
 - Details→1 dimension only
 - Tooltips →fact values

Donut Chart:

- It is also like pie chart
- When we have less categorical data.

Treemap:

- It is used to visualize categorical data
- When we have medium level categorical data

Bar Chart:

- When we have only one dimension and one fact, then it is called as bar chart.
- When we have more than one Dimension and fact, then it is called Stacked bar chart
- Stacking means placing one on another
- It is also used to visualize the categorical data.
- When we have large categorical data
- We can't get the grand total by default but we can set the property total labels

Column Chart:

- When we have the large categorical data
- When we wanted to show negative values

Stacked VS Clustered:

Stacked: One value on the top of another Value

Clustered: One value next to the another value

Scattered Chart:

- When we have large categorical data
- To show the performance of sales.
- We have play x axis option.
- In this chart we can see the performance of year sales like a video.

Trend Visualizations(Date and time):

- **Line Chart**
 - Used to visualize time series data or trend data
 - Eg: Year wise sales
 - We can understand the trend
 - Increasing trend
 - Decreasing Trend
- **Area Chart**
 - Used to visualize time series data or trend data
 - Fill the color
 - It start with Zero
 - We can understand the trend
 - Increasing trend
 - Decreasing Trend
- **Stacked area chart**
 - We use this chart, When we need cumulative sum
 - It does not start with zero
- **Line and Stacked column chart(Combination Chart)**
 - Combining line and stacked column chart
 - It is also called as combination chart
- **Line and clustered column chart(Combination Charts)**
 - Combining line and clustered column chart
 - It is also called as combination chart
- **Ribbon chart**
 - It is same as stacked column chart
 - Sort the legend field well data based on ranking
- **Waterfall Chart**
 - Between two time periods, we wanted to see increasing and decreasing trend.
 - We wanted to see breakdown for given dimension

KPI Data Visualization:

- Card visualization→ To show the single total Nos
- Multi card visualization→ To show multiple total column values
- Gauge visualization
 - To understand sales and targets
 - Min value 0
 - Max Value(Sales*2)
- KPI Visualization→ to show the targets and sales in the form of percentage
- Customized tooltip
- Funnel chart
 - When the business contains multiple processes.
 - Identify bottlenecks any process.
 - Similar to bar chart.
 - Rarely, we get this type of data.

Visualizing Geographical data:

- **Map Visualization:**
 - Visualize the geographical data
- **Filled Map**
 - It fills the entire map
 - Categorize the geographical data

Visualizing table data:

- Table
 - To visualize Paginated report
 - Ex: Bank statements
 - Credit card month statements
- Matrix

Grouping and Binning:

- Customized groups on date and number data are called binning.
- Grouping of textual data is called a group
- Dimensions will be
 - Text data
 - Date and Time data
 - Number data
- Group
 - Text
- Bin
 - Date
 - Numbers

Date column custom Group:

- No of days to deliver to a product
- Add a new column
- **NoOfDaysDel= DateDiff(Sales[Orderdate],Sales[Shipdate],Day)**

Bookmarks:

- Saves the current state of the reports.
- Show you subset of data.

Selection pane:

Hide and show the visualizations in a given report.