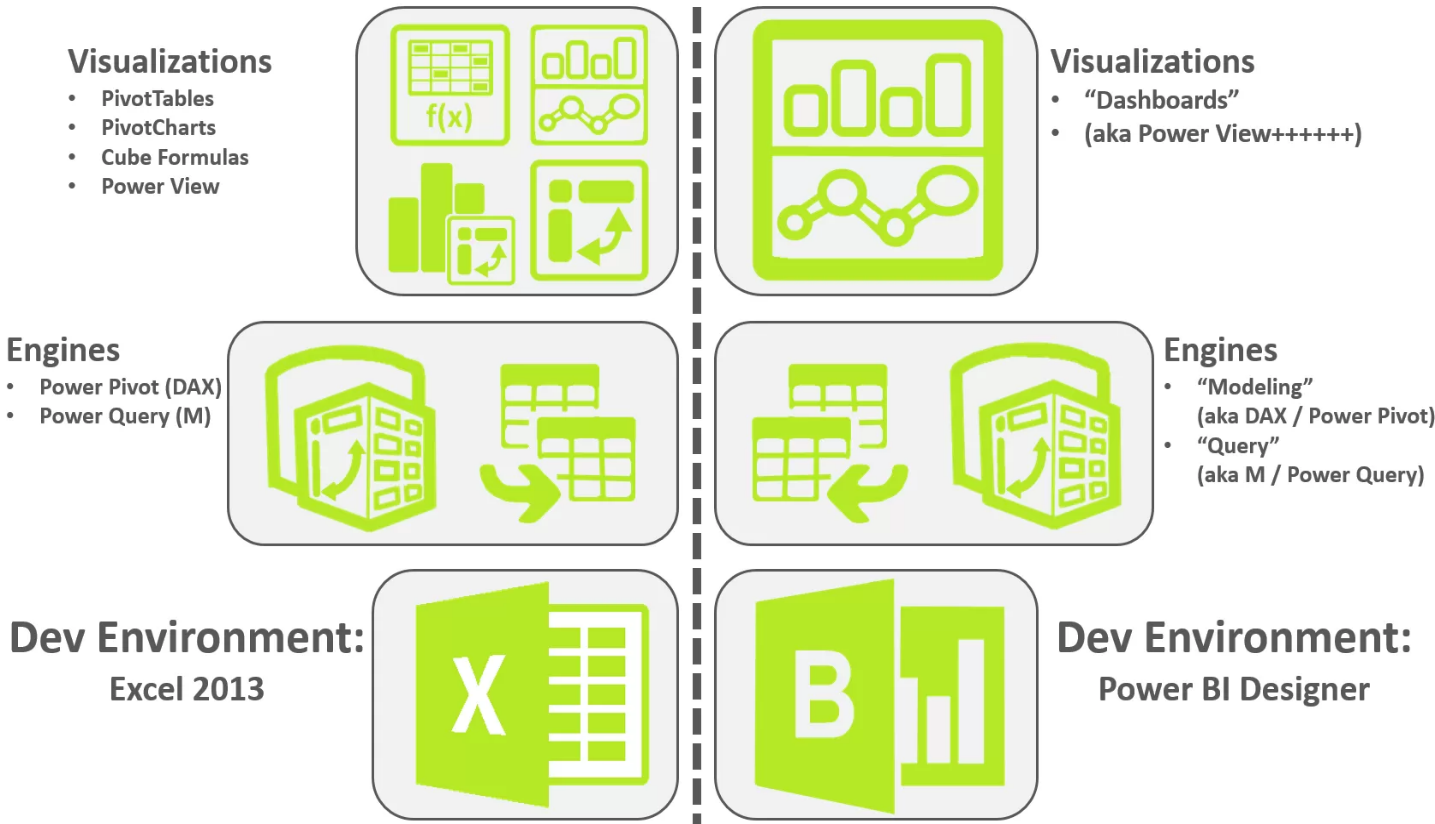
**Question 1. What is Power BI and how does it differ from Excel?**

**Answer -** Power BI is a business intelligence platform that provides nontechnical business users with tools for aggregating, analysing, visualizing and sharing data. it is used to find insights within an organization's data. It can help connect disparate data sets, transform and clean the data into a data model and create charts or graphs to provide visuals of the data.

**Differences between Power BI and Excel-**

1. **Functionality and Data Handling:**

Excel is a versatile spreadsheet software that allows users to manipulate and analyze data using formulas, functions, and pivot tables. It provides a wide range of features for data organization, calculations, and basic visualizations. On the other hand, Power BI is a dedicated business intelligence tool that focuses on data modeling, interactive visualizations, and advanced analytics. It enables users to connect to various data sources, create complex data models, and build interactive dashboards and reports.

1. **Scalability and Performance:**

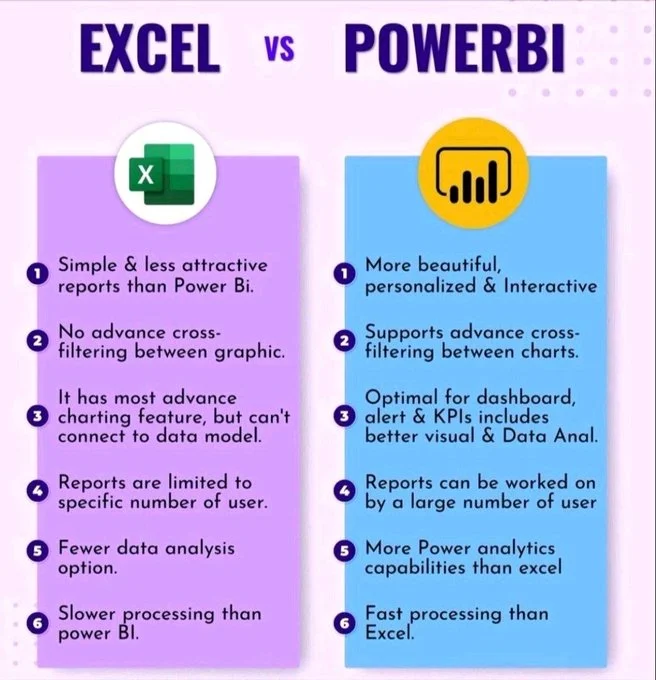
Excel is suitable for small to medium-sized datasets and can handle basic data analysis tasks efficiently. However, as the dataset and complexity grow, Excel’s performance may start to decline. Power BI, designed for larger datasets and enterprise-level reporting, excels in handling big data and provides robust performance even with complex calculations and visualizations. It leverages in-memory technology and optimized data models to deliver faster and more scalable results.

1. **Collaboration and Sharing:**

Excel allows for easy sharing of workbooks via email or cloud storage platforms like OneDrive. While multiple users can collaborate on a workbook simultaneously, it may lead to version control issues. Power BI, on the other hand, offers enhanced collaboration features. It enables users to create and share interactive dashboards and reports on the Power BI service, allowing real-time collaboration, data refresh, and access control. Power BI also provides the ability to publish reports to the web or embed them in other applications for broader sharing.

1. **Visualizations and Reporting:**

Excel provides basic charting and graphing capabilities, allowing users to create static visuals. Power BI, however, offers a wide range of customizable and interactive visualizations, including charts, maps, and gauges. It enables users to create dynamic reports that respond to user interactions, drill down into data, and provide a rich visual experience. Power BI also provides advanced data exploration features like natural language queries and Q&A, making it easier to find insights quickly.



**Question 2. Explain the concept of data modeling in Power BI?**

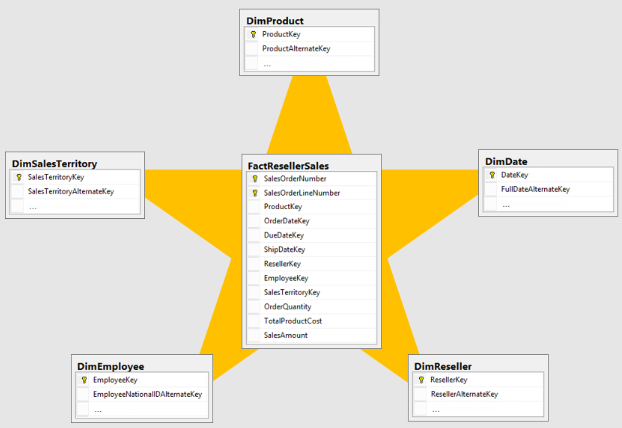
**Answer -** Data modeling is the process of defining the data structure, properties, and relationships within a data model basically data model in Power BI is a logical representation of how data is structured and related within the tool. It is a collection of tables and relationships between them that are used to create reports and visualizations.

Data models help organize, group, and assess data from different dimensions to better understand the causal and correlational relationships between different elements, identify repeating patterns, and draw conclusions. Power BI data models help business leaders explore data at different levels of abstraction, using pre-defined business rules for analysis.

A data model typically consists of one or more data sources, which can be anything from Excel spreadsheets to cloud-based databases and one or more tables that represent the data in those sources. The relationships that connect these tables are the cornerstone of data modelling.

Here are some data modeling concepts in Power BI:

**Star schema :-** A star schema is a data organization strategy that involves having fact tables at the center of the star, with dimension tables radiating from it. This layout provides a more intuitive and user-friendly experience for end-users, enabling them to easily navigate through the data model and explore data at varying degrees of detail.



**Optimizing data models :-** A well-developed data model can help optimize tables and columns and save time.

**Creating relationships** :- Power BI's data modeling capabilities allow users to create relationships between different data sets. This can be useful for large organizations with complex data structures.

Other concepts related to data modeling in Power BI include: Importing data into the Power BI file, Direct Query, and Aggregating data in volume.

**Question 3. What are the different types of connections available in Power BI?**

**Answer -** These are the approaches through which Power BI interacts with data and generates responses to visualization queries. Every visualization within a Power BI report relies on a dataset, which in turn utilizes one of the following connection types:**-**

**1. Import Mode: -**

In this mode, data from the source is imported into Power BI Desktop or Power BI Service. This mode is suitable for small to medium-sized datasets that do not frequently change. It provides fast performance as the data is stored within the Power BI file or dataset.

This type of connection imports the whole dataset into the memory. This memory will be the memory of the machine that hosts the Power BI dataset. If you have a Power BI dataset opened in Power BI Desktop, then it will be the memory of the machine that Power BI Desktop is running on it. When you publish your Power BI file on the Power BI Service, it will be the memory of that machine in the cloud.

**2. DirectQuery Mode: -**

With DirectQuery mode, Power BI sends queries directly to the data source each time a visualization or report is refreshed or interacted with. This mode is suitable for large datasets or scenarios where you need real-time or near real-time access to data. However, it's important to note that performance may be slower compared to Import mode, especially for complex queries or large datasets.

DirectQuery, as a connection type in Power BI, does not involve loading data into the Power BI model. DirectQuery means that Power BI establishes a direct connection to the data source. Whenever a visualization appears in a report, the data is sourced directly from a query sent to the data source.

**3. Live Connection Mode: -**

This mode allows you to connect directly to an Analysis Services model, either in Azure Analysis Services or SQL Server Analysis Services. With this mode, Power BI reports connect live to the underlying data model, enabling interactive analysis while leveraging the processing power and optimizations of Analysis Services.

**Question 4. How do you handle data transformation in Power BI?**

**Answer -** To handle data transformation in Power BI, we can use the Power Query tool to import, clean, and transform data from various sources. Here are some steps we can take:

**1. Load data -**

Open Power BI Desktop, click Get Data, and select your data source. You can import data from sources like Excel, SQL databases, and web data.

**2. Edit data -**

Use the Power Query Editor to change and reshape data. You can access the Power Query Editor by selecting Home > Transform data in Power BI Desktop.

**3. Perform query operations -**

You can perform operations on queries, such as combining, referencing, duplicating, or grouping them.

**4. Create data models -**

Use Power Query to create data models by loading and transforming data.

**5. Create data visualizations -**

Use Power Query to convert data into desired formats, such as pivoting and unpivoting data.

**6. Replace values -**

Use the Replace Values feature to substitute specific values within a column with new values.

**Question 5. What is DAX (Data Analysis Expressions) and why is it important in Power BI?**

**Answer –** DAX is a collection of functions, operators, and constants that can be used to create formulas and expressions. DAX is used in Power BI, Analysis Services, and Power Pivot in Excel.

**Importance of DAX function in Power BI -**

 Companies today create and consume data at dizzying speeds. Furthermore, to outperform one another, these companies rely on Data Analytics to interpret their massive amounts of data and extract some meaning from it. Organisations then use the outputs of Data Analytics to formulate and execute data-driven #business choices. This procedure, however, is more complex than it seems because businesses keep their data in numerous sources in a decentralized fashion.

In such cases, organisations look to Business Intelligence (BI) products like Power BI to help them manage their data more effectively. Power BI delivers simple solutions for activities involving data analytics and #visualisation. Furthermore, it has capabilities that organisations may use to optimise their data-related operations. DAX is the whole list of such functions in Power BI.

 DAX (Data Analysis Expressions) is a significant function package that helps Power BI users simplify their jobs. Furthermore, DAX allows users to construct dynamic Data Management and do complex mathematical and logical activities with a few mouse clicks. In this way, the #DAX library improves users' Power BI experience.

 This article describes Power BI and DAX and their essential features. It also goes through the different sorts of DAX Functions for Power BI. It will also show you the advantages of utilising DAX Functions. DAX is an abbreviation for Data Analysis Expressions, expressions or formulae used for data analysis and computations. These expressions are composed of functions, operators, and constants evaluated as one formula to produce results (value or values). DAX formulae are highly valuable in BI solutions like Power BI because they allow data analysts to make the best use of their data sets. Analysts may use the DAX programming language to explore new ways to calculate data values and get new insights.

 Let us go through some crucial elements regarding DAX to assist you in comprehending the concept better:

* DAX is a functional language whose whole code is always a function. A DAX expression that may be executed can include conditional expressions, nested functions, value references, etc.
* DAX formulas have two basic data kinds: numeric and non-numeric or Others. Integers, decimals, currencies, and other numerical data types are examples of numeric data types. Strings and binary objects are examples of non-numeric data.
* DAX expressions are evaluated starting with the innermost function and progressing to the outermost one at the end. This emphasises the need to develop a DAX formula.

It is normal to wonder why DAX is crucial to grasp and operate correctly with Power BI. Creating reports in Power BI utilising data importing, processing, and visualising features is accessible. A user must have a basic understanding of #PowerBI Desktop to build a good report with all the access data. However, DAX is required if you wish to develop and employ complex calculations in your Power BI reports.

 Assume you want to create a graphic to compare year-over-year growth/sales or to evaluate growth percentages across different states of a country. It would help if you made new measurements in the DAX language for this. This allows you to construct new measures, utilise them to generate innovative visualisations, and get new insights into data. With such unique insights into data, you might have appropriate answers for business challenges you could overlook using the traditional analysis method. As a result, DAX makes data analysis with Power BI smart and intelligent.

 Power BI itself has some fantastic features. Implementing DAX Functions in Power BI, on the other hand, will take you a step further and give you the following benefits:

Many complex Power BI functions may be used using the DAX library. For example, you may dynamically change your data with DAX by utilising select, join, filter, and many other functions. Users may add data to your Power BI dashboard in this manner, and it will dynamically produce data, including Calculated Columns, Tables, and Measures.

 You can make your Power BI #Dashboards savvier by using DAX. This is accomplished by constraining the data your Dashboard will retrieve and show using Calculated Columns and Measures in Power BI. A well-written DAX expression can reduce latency and finish data processing while working with restricted resources.

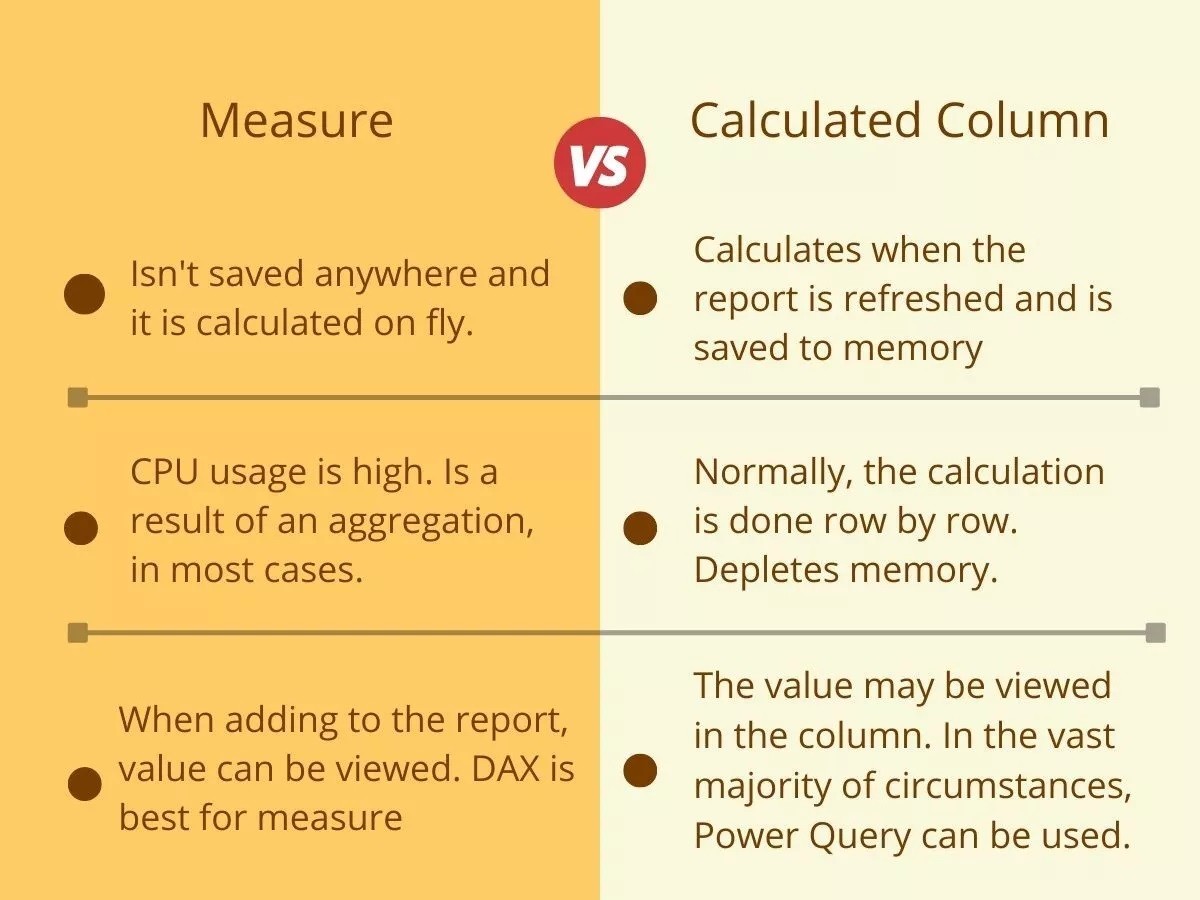
 DAX works with a set of nested filter contexts to maximise efficiency. Working with DAX Functions will therefore affect your thinking pattern when it comes to tables and data filtering. Your DAX experience will improve your thinking and help you identify better ways to integrate, filter, and manipulate data with Power BI and other tools.

**Question 6. Can you explain the difference between calculated columns and measures in Power BI?**

**Answer -** Calculated columns are computed based on data that has already been loaded into your data model. When you write a calculated column formula, it is automatically applied to the whole table and evaluated individually for each row. The values in calculated columns are evaluated when you first define them and when you refresh your dataset. Once evaluated, the values are stored in your data model, which means your data model size increases and it consumes more RAM. The more calculated columns you have the larger your model will become. On the other hand, measures are computed at query time. A measure is stored in the model as source code, but it is computed only when it is used in the report.

Another important difference between measures and calculated columns is that measures are evaluated in the filter context of the visual in which they are applied. The filter context is defined by the filters applied in the report such as row selection, column selection, report filters and slicers applied. Measures are only evaluated at the level of granularity they are plotted at. As calculated columns are computed when you first define them/ when you refresh your dataset, they do not have access to the filter context. Calculated columns are calculated outside of the filter context and do not depend on user interaction in the report.

When you write a calculated column, you need to rely only on the row context. The row context, is simply the notion of a current row. It specifies which row we are calculating the values for, so that DAX can look at other values in the same row. In a calculated column, the row context is implied. When you write an expression in a calculated column, the expression is evaluated for each row of the table. The calculated column has knowledge of the current row.

By contrast, measures implicitly do not have a row context. This is because, by default, they work at the aggregate level. So, you cannot refer to columns directly in a DAX measure, you will get an error because no row context exists. This is because the measure will not know which row to choose in the table. In order to reference a column in a measure, you must wrap the column in an aggregation function. As we are aggregating values, it doesn't matter which row we are using, we do not need a row context, as we are just aggregating all of the values in a single column to return a single value. Alternatively, we can create a row context in a measure by using an iterator function.

**Question 7. How do you handle relationships between tables in Power BI?**

**Answer -** A relationship described between two or more tables via a common attribute is termed a table relationship. It is very crucial as it enables users to access data from two separate tables with ease.

To handle relationships between tables in Power BI, we can create and manage them in the following ways:

* Create a relationship

In the Data view, select Table tools > Manage relationships. You can also create a relationship by clicking and dragging a column from one table onto a column in another table.

* Manage a relationship

In the Model view, you can select Home > Manage relationships. You can also manage a relationship by double-clicking a line between two tables, right-clicking a line between two tables, or selecting Open relationship editor in the Properties pane.

* Use AutoDetect

Delete a relationship, then click AutoDetect. In the dialog, select New, choose a table from the first dropdown, and choose the other table from the second dropdown.

* Handle referential integrity violations

Unknown members indicate referential integrity violations, where the "many" side value has no corresponding "one" side value. You can eliminate these by cleansing or repairing the source data.

For automatic detection of relationships, the data types in all related columns should be compatible. The number of unique keys in the lookup column must also be greater than the values in the table on the many side.

**Question 8. What is the purpose of a Power BI Gateway?**

**Answer -** The purpose of Power BI Gateway is to facilitate secure data transfer between on-premises data sources and Power BI services. It acts as a bridge, enabling seamless and secure communication between cloud-based Power BI reports and dashboards and data stored within an organization's local network.

**The key purposes and benefits of using Power BI Gateway:**

1. **Secure Data Transfer**: Power BI Gateway ensures that data transfer between on-premises data sources and Power BI services is secure. It uses encryption and other security protocols to protect data during transit, thereby maintaining data confidentiality and integrity.
2. **Real-Time Data Access:** By using Power BI Gateway, users can access and visualize real-time data from on-premises sources directly in their Power BI reports and dashboards. This is crucial for scenarios where timely insights are necessary, and static data snapshots are insufficient.
3. **Scheduled Data Refresh:** Power BI Gateway allows for scheduled data refreshes, ensuring that the data in Power BI reports and dashboards is up-to-date. Users can configure refresh schedules to pull the latest data from on-premises sources at regular intervals without manual intervention.
4. **Support for Multiple Data Sources:** Power BI Gateway supports a wide range of on-premises data sources, including SQL Server, Oracle, SAP, and various other relational and non-relational databases. This flexibility allows organizations to leverage existing data infrastructure while utilizing Power BI's analytical and visualization capabilities.
5. **Hybrid Deployment Scenarios:** Power BI Gateway is essential for hybrid deployment scenarios where organizations have a mix of cloud and on-premises data sources. It enables seamless integration of these disparate data environments, allowing users to create comprehensive and unified reports and dashboards that draw from both local and cloud-based data sources.

By enabling secure and efficient data connectivity, Power BI Gateway plays a crucial role in extending the capabilities of Power BI to include on-premises data, thus providing a holistic and integrated data analytics solution.

**Question 9. How can you schedule data refresh in Power BI Service?**

**Answer -** To schedule data refresh in Power BI service, we can:

1. Log in to Microsoft Power BI Service
2. Go to the workspace where your dataset is located
3. Select the dataset you want to schedule for refresh
4. Open the dataset's settings and configuration options
5. Select Refresh > Schedule refresh
6. Enable refresh and configure the schedule
7. Click Apply to confirm your selection

We can also schedule data refresh without a gateway by:

1. Opening your recently published report
2. Going to More options => View semantic model
3. Clicking the Refresh drop-down menu and selecting Schedule refresh

**Question 10. Explain the concept of row-level security in Power BI.**

**Answer -** Row-Level Security (RLS), in the context of Power BI, refers to limiting the rows of data that a specific user can see when viewing a report. This allows you to better control what users can see in the published report based on their Power BI login account. For example, using this technique, a regional manager can view data for his region but not data for other regional managers. Thanks to this approach to data protection, it is possible to publish a single report or dashboard (we manage and maintain only one report then) instead of several identical ones supplied with a different set of data.

RLS Management in Power BI involves effectively implementing and maintaining Row-Level Security to control data access based on user roles and attributes. With RLS, organizations can ensure that users only see the data that is relevant to them, while maintaining a high level of data protection and confidentiality.

For example, consider a scenario where a Power BI report contains data for multiple regions. By implementing RLS, a regional manager can be granted access to view data specific to their region, while being restricted from accessing data for other regional managers. This approach streamlines data protection by enabling the publication of a single report or dashboard that can be managed and maintained centrally, eliminating the need for multiple reports with different datasets.

Power BI provides the necessary tools and features to manage RLS efficiently. Administrators can define security roles and associated filters to control data visibility. The Power BI service allows for the centralized administration of security roles, providing a comprehensive overview of the roles implemented within the organization.

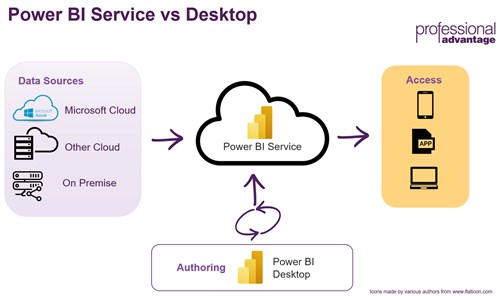
Additionally, Power BI offers capabilities for auditing and monitoring RLS usage and effectiveness. This helps administrators track and analyze access patterns, identify any potential security gaps, and ensure compliance with data protection regulations.

To effectively manage RLS, administrators can leverage resources provided by Microsoft, such as the Power BI documentation on RLS management. The documentation offers guidance on implementing RLS, creating security roles, and managing access levels based on user attributes and roles.

By implementing RLS management in Power BI, organizations can enforce data access controls, ensure the appropriate level of security, and provide users with access to the data they need to make informed decisions while maintaining data confidentiality.

**Question 11. What is the Power BI Desktop and how does it differ from Power BI Service?**

**Answer -** Power BI Desktop - Power BI Desktop is a Windows application that empowers users to create rich and interactive data visualizations, reports, and dashboards. It serves as a local development environment where you can connect to various data sources, transform and model data, and design compelling visuals.



When to Use Power BI Desktop:

1. Data Preparation and Modeling: Power BI Desktop is your go-to tool when you need to import, clean, and transform data from different sources. You can create relationships between tables, define calculated columns, and customize the data model to suit your needs.

2. Report Design: If you want to design pixel-perfect reports and dashboards with precise control over layout, formatting, and visual elements, Power BI Desktop is your best friend. It offers a wide range of data visualization options, including charts, maps, tables, and more.

3. Offline Work: Power BI Desktop allows you to work on your reports without an internet connection. This is especially useful for scenarios where data privacy or compliance requirements restrict cloud access.

4. Advanced Calculations: For complex calculations, custom measures, and DAX (Data Analysis Expressions) formulas, Power BI Desktop provides a comprehensive environment for creating and testing these calculations.

When to Use Power BI Service:

1. Collaboration: Power BI Service is designed for collaboration. Once you publish your reports to the service, you can share them with colleagues and collaborate in real-time. Multiple users can access and work on the same report simultaneously.

2. Data Refresh and Live Connections: Power BI Service provides scheduled data refresh options for keeping your reports up to date. It also supports live connections to on-premises data sources through the Power BI Gateway, ensuring your reports always reflect the latest data.

3. Accessibility: Since Power BI Service is cloud-based, it can be accessed from anywhere with an internet connection. This makes it an ideal choice for distributed teams or organizations with remote work setups.

4. Embedding: If you want to embed your Power BI reports and dashboards into external websites, applications, or SharePoint, Power BI Service offers embedding capabilities, making it possible to share insights with a broader audience.

5. Content Management: Power BI Service provides content packs, workspaces, and sharing features that allow you to organize and manage your reports and dashboards efficiently.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Power BI Desktop** | **Power BI Service** |
| What is it? | Also known as Power BI Designer, Power BI Desktop is a free on-premise version of Power BI. | Also called Power BI Portal and Power BI Online, Power BI service is a cloud-based service. |
| Purpose | It is designed for data analysis and report creation. | It is designed for light report editing and collaboration. |
| Features | Power BI Desktop’s features include Python and DAX, calculated columns, data transformation, modeling and shaping and RLS creation. | Some unique features include dashboards, sharing, gateway connections and RSL management. |
| Data Sharing and Collaboration | Not available in Desktop | Available in service |

**Question 12. Explain the concept of Direct Query in Power BI.**

**Answer -** Power BI Direct Query: DirectQuery is an alternative method of querying data in Power BI. Find out everything you need to know about DirectQuery.

When working with data in [Power BI](https://datascientest.com/en/tutorial-power-bi-learn-how-to-use-the-bi-tool-by-microsoft), it is essential to choose the connectivity mode. Import mode is generally the most commonly used, in combination with the Tabular model. This allows you to run queries and generate reports very quickly. However, a less well-known alternative is DirectQuery.

[Power BI](https://datascientest.com/en/power-bi-license-how-much-does-the-microsoft-solution-cost) Direct Query is a method for retrieving data directly from its source, at the time of the query. Whereas Import mode stores a snapshot of the data in memory, Power BI Direct Query (DQ) does not store any data.

With each query, the data is extracted directly from the source. The data therefore resides within the original source before, during and after the query is executed.

With each query, users generate a query or a set of queries that need to be executed in a specific order. The Tabular model is made up of the Formula Engine (FE) and the Storage Engine (SE).

The FE accepts queries, creates a query plan, and generates the query based on the choice between Import and DirectQuery modes to target the corresponding data source.

If Power BI Directy Query mode is chosen, the Formula Engine translates the DAX into [SQL](https://datascientest.com/en/sql-joins-everything-you-need-to-know-about) and sends the query directly to the data source. The data is not imported by PowerBI from the underlying tables. Only the metadata is retained.

Using Power BI’s Composite models, it is possible to combine Import and DirectQuery modes within the data model. The preferred option can be chosen for each table. Note that it is possible to switch from DirectQuery to Import mode, but not vice versa.

Finally, Dual mode is a combination of Import and DirectQuery modes. Data from the table is loaded into memory, but can also be retrieved directly from the source at query time.

**Question 13. What are Power BI templates and how are they useful?**

**Answer -** Power BI Templates are a huge topic, partially because different people will give you different definitions of what they consider to be a Power BI template. A Power BI Developer would say that a Power BI template is a Power BI file which contains all meta-data for a report but doesn’t contain the data itself. Then, the developer could add that you can also use theme templates in Power BI to apply colour schemes to your Power BI reports.

On the other hand, a Power BI business user would say that a Power BI template is an example of a Power BI application. An example of a report which you might use to start not from scratch when trying to think of a sales or financial forecast dashboard. You can change, improve and tailor such a template to the needs of your business.

**Benefits :-**

[1. Extensive Data Analysis and Sharing](https://www.ccslearningacademy.com/top-benefits-power-bi/#1-extensive-data-analysis-and-sharing)

[2. Enable Artificial Intelligence](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-2)

[3. Personalized Dashboards and Interactive Reports](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-3)

[4. Integration with a Wide Array of Data Sources](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-4)

[5. No Geographical Limits](https://www.ccslearningacademy.com/top-benefits-power-bi/#5-no-geographical-limits)

[6. Time-saving Models](https://www.ccslearningacademy.com/top-benefits-power-bi/#6-time-saving-models)

[7. No Need for Specialized Technical Support](https://www.ccslearningacademy.com/top-benefits-power-bi/#7-no-need-for-specialized-technical-support)

[8. Cost Efficiency](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-8)

[9. Interactive UI/UX Features](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-9)

[10. Added Functionality with Other Microsoft Apps](https://www.ccslearningacademy.com/top-benefits-power-bi/#10-added-functionality-with-other-microsoft-apps)

[11. Scalability](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-11)

[12. Easy to Migrate](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-12)

[13. Multiple Dashboard Deployment](https://www.ccslearningacademy.com/top-benefits-power-bi/#13-multiple-dashboard-deployment)

[14. Regular Updates](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-14)

[15. Secured and Convenient Filters](https://www.ccslearningacademy.com/top-benefits-power-bi/#15-secured-and-convenient-filters)

[16. Question and Answer Functionality](https://www.ccslearningacademy.com/top-benefits-power-bi/#16-question-and-answer-functionality)

[17. Quick Retrieval of Business Information](https://www.ccslearningacademy.com/top-benefits-power-bi/#eb-table-content-17)

**Question 14. How do you handle incremental data refresh in Power BI?**

**Answer -** Incremental refresh refers to loading only the new or updated data sets added to a data table after the previous loading cycle has been completed.  
It helps to extend the scheduled refresh operations for such tables to ensure only the most recent sections are uploaded instead of the entire table.

To successfully set up an incremental refresh in Power BI, the explained requirements below are necessary to have:  
**Table with Date Field(s)**  
In implementing incremental refresh, your dataset must have a table with a DateTime or timestamp field. This column will mark new or updated information that needs to be refreshed. You can tell by looking at this column if you have made any edits to a table since PowerBI last refreshed.  
**A Data Source That Supports Query Folding**Power BI's Query Folding functionality optimizes queries by offloading some processing work to the data source. A data source that supports query folding is required for an incremental refresh. Based on the date column in your database, the data source must be able to conduct queries that return just the rows that have changed since the last refresh.  
**Supported Data Sources**  
Power BI supports incremental refresh for the following data sources: SQL Server, Oracle, PostgreSQL, MySQL, and Teradata. You can also utilize additional data sources that enable query folding; however, you may have to construct the queries to use incremental refresh manually. In addition, specific data sources may necessitate the installation of a gateway to enable communication between Power BI and the data source.

**Supported Plans**  
Whether or not you can use Incremental refresh also depends on the Power BI plan you're on. You can only use Power BI incremental refresh with Power Bi Pro, Power BI Embedded datasets, and Premium per user plan. Additionally, you want to get the latest data in real-time using DirectQuery. In that case, you must be on a Power BI Premium, Power BI Embedded datasets, or Premium per user plan.

**Create the RangeStart and RangeEnd Parameters**   
The first step is to use the Power Query Editor to create RangeStart and RangeEnd parameters and set their default values. These parameters allow you to filter the number of rows to be initially loaded into the Power BI model. To do this, select Manage Parameters in the Power Query Editor. In the window that opens, create the parameter by typing the name type, e.g., RangeEnd (case sensitive), selecting the Date/Time, and entering the date/time value in the box for Current Value.

**Filter The Data Using The Parameters**   
Still, in the Query editor, select the Custom Filter option for the data column you're trying to set a filter for. Here you will be prompted to choose two conditions. For the first condition, select the *After* or *Equal to* option; for the second condition, select the *Is before*option. Alternatively, you can choose the *After* option for the first condition, in which case the second condition will be *Is before* or *Equal to*. Click the *Ok*button to close this dialog then click *Close & Apply* to apply the filter.

**Question 15. What is the role of Power Query in Power BI?**

**Answer –** Role of Power Query:

* Connecting to data sources: Power Query can connect to hundreds of data sources.
* Transforming data: Power Query offers over 350 different types of data transformations for each data source.
* Cleaning data: Power Query can clean, reshape, and consolidate data.

| **Existing challenge** | **How does Power Query help?** |
| --- | --- |
| Finding and connecting to data is too difficult | Power Query enables connectivity to a wide range of data sources, including data of all sizes and shapes. |
| Experiences for data connectivity are too fragmented | Consistency of experience, and parity of query capabilities over all data sources. |
| Data often needs to be reshaped before consumption | Highly interactive and intuitive experience for rapidly and iteratively building queries over any data source, of any size. |
| Any shaping is one-off and not repeatable | When using Power Query to access and transform data, you define a repeatable process (query) that can be easily refreshed in the future to get up-to-date data. In the event that you need to modify the process or query to account for underlying data or schema changes, you can use the same interactive and intuitive experience you used when you initially defined the query. |
| Volume (data sizes), velocity (rate of change), and variety (breadth of data sources and data shapes) | Power Query offers the ability to work against a subset of the entire data set to define the required data transformations, allowing you to easily filter down and transform your data to a manageable size. Power Query queries can be refreshed manually or by taking advantage of scheduled refresh capabilities in specific products (such as Power BI) or even programmatically (by using the Excel object model). Because Power Query provides connectivity to hundreds of data sources and over 350 different types of data transformations for each of these sources, you can work with data from any source and in any shape. |

* Refreshing data: Power Query queries can be refreshed manually, by schedule, or programmatically.
* Sharing queries: Users can share queries by copying the query code to their clipboard.

**Question 16. Explain the difference between calculated columns and calculated tables in Power BI.**

**Answer -** Calculated Columns :- A [calculated column](https://go.christiansteven.com/bi-blog/power-bi/automate-your-company-reports-using-power-bi) is a column of data that is added to an existing table in your model. This is done either in report view or data view using a [DAX](https://go.christiansteven.com/bi-blog/what-is-dax-what-are-the-benefits-of-using-variables-in-dax) formula to determine the data that is displayed. The column will be shown under Field as usual, but you will notice a symbol indicating that a formula was used.

Calculated columns are like other columns in your report, including that they are stored in memory. When you refresh your report, the column will calculate row by row, so it cannot give you real-time updates.

When to use a calculated column:

A calculated column can be a useful tool if you are looking to combine data from multiple columns in a table or across tables. You might also want to extract specific sub-data from existing columns.

To create a calculated column:

* Right-click on the table name
* Click “New Column”
* Enter your DAX formula

Calculated Tables :- A calculated table differs from other tables you might have in your report because it is created by pulling from data that is already in your model, whereas most other tables are imported from an outside source.

A calculated table can have a connection with other tables. You may choose to join two tables, or extract data from multiple tables to be displayed together. Like calculated columns, these tables are stored in memory and will update whenever the source data is updated.

When to use a calculated table :

Calculated tables are great for data that you want to be stored in the model, as opposed to having real-time calculations. If you need to join two tables or combine data from multiple tables, this function will work well for you. Many report creators enjoy it for creating calendar tables as well.

You may also choose to use a calculated table for calculations that you do not want in your end report, or that will not be shared with others. This option is good for modeling different scenarios. Modelers often use calculated tables to debug a lengthy DAX expression to make sure it is running smoothly.

To create a calculated table:

* From [Report View](https://go.christiansteven.com/bi-blog/power-bi/what-are-the-different-views-in-power-bi-desktop) or Data View, go to the Modeling tab
* Within Modeling, find the section for Calculations
* Click “New Table”

If you create the table in Data View, you will be able to see it right away, so this can be a good starting point.

**Question 17. How do you create custom visuals in Power BI?**

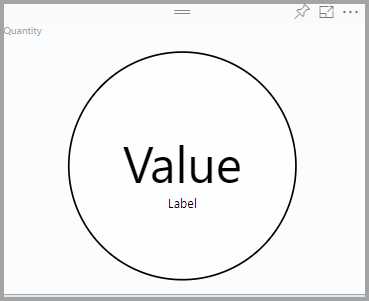
**Answer -** Power BI Custom Visuals -

**1. Break Down The Design Elements :**

As with any project, you first need to break down the elements of what you want to build. What may seem simple at a moment’s glance can quickly turn complex when you factor in all the variables.

Let’s look at an example.

Below is a [circle card visual](https://learn.microsoft.com/en-us/power-bi/developer/visuals/develop-circle-card) that Microsoft includes in their Power BI report tutorial on how to build custom visuals. Let’s break it down to its visual elements:

[](https://f.hubspotusercontent10.net/hubfs/20618610/Imported_Blog_Media/Circle-with-a-border-Dec-17-2021-07-20-28-61-PM.png)

1. Circle with a border
2. White background color in the circle
3. Value text
4. Label text

Okay, not too bad. So, let’s say you install this and get everything working on your laptop. Once you start playing with the visual, you start to ask questions: How does resizing the visual work? Does it auto-scale? Should the data labels be in a fixed position or allow the user to choose formatting options? How do you change the background of the circle based on the data? How can I add more data to this visual? What about tooltips? Drill-though?

So, after giving it some thought and talking with your team, your breakdown starts to grow:

* 1. Data Wells:
  + Value
  + Label
  + Color
* 2. Formatting Options:
  + Border/No Border
  + Font
  + Font Size
  + Font Color
  + Value Positioning
  + Label Positioning
* 3. Tooltips
* 4. Drill-through

Now, this is just the beginning. After you and your team have the night to sleep about it, you come back with all kinds of ideas: icons, arrows, sparklines, etc.

Like with all projects, narrowing down the scope to get a minimum viable product (MVP) is critical to getting acceptance, feedback, and championing

**2. Get The Visual Studio Code Editor :**

[Visual Studio Code](https://code.visualstudio.com/) is probably the hottest code editor out there on the market today(At least for Microsoft development). It’s fast to develop, fast to extend, and continually updated every month just like Power BI! If you don’t already have this tool in your development toolbelt, try it out. It’s great for text editing, but you will ultimately need to learn it if you plan to do anything with Power BI reporting custom visuals.

**3. Get Source Control And Check In Your Work :**

If you don’t already have source control, get some. There are plenty of providers out there that you can use for free like GitHub, Bitbucket, or Azure DevOps. If you don’t already have a source control provider (SCP), we recommend GitHub, a provider known for its speed. To integrate GitHub with Visual Studio Code, all you need is to download and install Git. Checking your code into GitHub is easy and there’s no excuse not to do it this way.

**4. Learn: Trial Basic Power BI Reporting Visualization Packages To Build Your Skill Set :**

Probably the easiest way to learn how to build Power BI custom visuals is to start with a basic visualization package like D3. Take advantage of online tutorials and examples to help you along the way as you grow your skill set. Try them out and impress your friends.

**5. Test Live Or Test In Power BI Desktop :**

If you go through the Microsoft tutorial online, they walk you through how to set up a development environment which is beneficial if you’re making small incremental changes to your code. In the end, be sure you can package your custom visual with the pbiviz package command. That produces the magical pbiviz file that you can then import into Power BI Desktop, tie to your data and publish to the Power BI Service to see your work in all its grandeur.

**Question 18. What are the best practices for optimizing performance in Power BI?**

**Answer -**

1. **Limit the number of visuals in dashboards and reports** - Using too many visuals in a single report slows report performance as each visual requires data processing and rendering. Limit widget visuals to eight per report page and grids to one per page. Limit tiles to 10 per dashboard.
2. **Use on-premises data gateway standard mode instead of personal mode** - The personal mode imports data into Power BI, which can cause resource limitations and performance issues (particularly with large databases). On the other hand, the standard mode retains data in its original location.
3. **Use separate gateways for Power BI service live connection and scheduled data refresh** - A Power BI service live connection (DirectQuery) maintains a real-time connection with data sources. Scheduled data refresh updates imported data at set times. Using the same gateway for both live connections and scheduled data refresh can overload it during refreshes, slowing live connections.

Separating the gateways ensures that both functions can work efficiently without affecting each other.

1. **Use calculated measures and filters to limit complex measures and aggregations in data models** - Complex measures and aggregations in data models can slow down query performance. Using calculated measures, which is computed during query execution, are more efficient than calculated columns. Applying filters can also aid in this process.
2. **Push calculations to the source** - Pushing calculations to the source is recommended as it offloads the processing to the data source, optimizing query processing. Being closer to the source can increase performance speed.
3. **Use Star schema instead of the snowflake schema when possible** - The star schema is a simpler and more denormalized data model compared to the snowflake schema. The snowflake schema involves more complex queries due to multiple related tables.

The star schema accelerates query execution and eases maintenance by reducing required joins. It improves query efficiency, reduces data redundancy, and simplifies report development.

1. **Use slicers sparingly** - Slicers are a great way of allowing users to navigate data, but they come at a performance cost. Each slicer generates two queries: one fetches the data and the other fetches selection details. Adding too many slicers drastically slows performance. Use the Filter pane to evaluate and remove unnecessary slicers.
2. **Host reports and data sources in the same region** - A Power BI tenant is the dedicated environment that houses an organization's data, reports, and dashboards. Microsoft operates data centers worldwide to provide and hosts its services. Each data center is in a specific region.
3. **Partition data and process multiple partitions in parallel for large datasets** - Partitioning is a technique that divides large tables into smaller subsets called partitions based on specific criteria. A columnar index is a type of database index that stores and organizes data by columns rather than rows.
4. **Use templates (.PBIT files) instead of starting with an empty .PBIX file**

Templates enable you to develop standardized and branded reports faster. With templates, you can save custom color palettes and themes, ensuring corporate branding is pre-applied to all pages. Templates automatically connect to commonly used data sources and offer commonly used DAX measures as well. These all help streamline the report creation process and allow more time for analysis.

1. **Use the query reduction settings in Power BI** - Use Power BI's query reduction settings to lower query frequency.

For slicers, select the **Add an Apply button to each slicer to apply changes when you’re ready** option. With this, changes to slicer selections are only applied when the user clicks the **Apply** button. This reduces the frequency of queries sent.

For filters, the **Add a single Apply button to the filter pane to apply changes at once (preview)** option consolidates filter changes and applies them all at once, further minimizing query volume.

1. **Avoid bi-directional and many-to-many relationships against high cardinality columns** - Bi-directional relationships refer to the two-way data flow between tables. Many-to-many relationships refer to multiple records in one table being related to multiple records in another. High cardinality columns are columns in a table with many unique values relative to the total number of rows.
2. **Avoid using floating point data types** - Floating point data types, such as float and double, can lead to round-off errors and less predictable calculations. They also often require more processing power, which can slow down report performance.

By avoiding these data types, you ensure more accurate data representation and improve report efficiency.

1. **Replace the auto-generated date table with a custom date table in your model** - Replacing auto-generated date tables with custom ones reduces model size and increases performance. The auto-generated date table creates a separate table for each date column, which can bloat the model. A singular custom date table, with date and time split for better compression, streamlines the model and allows for more efficient use of time-series functions in Power BI.
2. **Set IsAvailableinMdx to false on non-attribute columns** - Disabling attribute hierarchy for non-attribute columns, like measures, streamlines the data model. This setting reduces the overall data size and load time, increasing report performance by focusing on essential data and calculations.
3. **Reduce the amount of data loaded on the page**

Using bookmarks, drillthrough pages, and tooltips can significantly reduce the amount of data loaded on a page. This approach improves page load times and improved the user experience, especially on landing pages where quick access to information is crucial.

1. **Use report backgrounds for static images** - Employing report backgrounds for static images instead of multiple visuals reduces the performance cost. This strategy uses fewer resources than individual visuals, ensuring the report remains responsive and efficient.
2. **Choose the ideal storage mode for tables** - Choosing the right [storage mode](https://learn.microsoft.com/en-us/power-bi/transform-model/desktop-storage-mode) (Import, DirectQuery, or Dual) based on your data aggregation needs and visualization requirements is crucial. This decision impacts where your data is stored and how queries are processed, directly affecting the report's responsiveness and efficiency.
3. **Cross-check referential integrity check for relationships** - In DirectQuery sources with enforced primary keys, verify the **Assume Referential Integrity** setting in relationships. If set to **Off**, Power BI defaults to slower outer joins instead of inner joins. Adjusting this setting can optimize query performance and speed.
4. **Enable and optimize your**[**Row-Level Security**](https://docs.microsoft.com/en-us/sql/relational-databases/security/row-level-security?view=sql-server-2017)**(RLS)** - Row-level security limits data access based on user roles. With RLS, Power BI only imports data the user is authorized to view.
5. **Use**[**Power BI-Certified custom visuals**](https://docs.microsoft.com/en-us/power-bi/developer/visuals/power-bi-custom-visuals-certified) - Power BI-certified visuals are custom visuals on AppSource that have passed rigorous quality testing. Certified custom visuals are also the only custom visuals that can be viewed in [Export to PowerPoint mode](https://docs.microsoft.com/en-us/power-bi/consumer/end-user-powerpoint) and email subscriptions.
6. **Categorize report data by business impact using sensitivity labels** - Using [sensitivity labels](https://learn.microsoft.com/en-us/power-bi/enterprise/service-security-sensitivity-label-overview) in Power BI to categorize data (high, medium, low impact) raises user awareness about data security. High Business Impact (HBI) data requires users to request a policy exception to share the data externally. Low Business Impact (LBI) and Medium Business Impact (MBI) data do not require exceptions.
7. **Align cache update frequency with data source refresh** - By default, the Power BI cache update frequency is set to one hour. Synchronizing the cache update frequency with the data source refresh frequency ensures data accuracy and optimizes report performance. This alignment prevents unnecessary data processing and keeps the information up-to-date
8. **Use white or light background colors for reports** - White or light backgrounds in reports make them printer-friendly and easier to share. This approach ensures reports are accessible both online and offline, improving their utility.
9. **Shorten displayed numbers** - Limiting displayed numbers to four numerals and decimal points to two decimal points improves readability and consistency across reports. This practice makes reports easier to understand and analyze, especially when scaling numbers for thousands or millions.
10. **Use tooltips to provide additional information** - Tooltips provide extra context in a compact form, making efficient use of report space. Careful selection of visuals in tooltips ensures information is helpful without being overwhelming, elevating the user experience.
11. **Use meaningful names for report objects** - Assigning clear, business-friendly names to columns and measures, and hiding unused columns, reduces user confusion. This clarity improves the usability of the reports, making them more intuitive and effective.
12. **Enable personalization of visuals** - Enable the [Personalize visuals](https://learn.microsoft.com/en-us/power-bi/consumer/end-user-personalize-visuals) option in report settings when sharing reports with users. This fosters deeper engagement and exploration. Combining this feature with personal bookmarks empowers users, facilitating self-service BI.
13. **Avoid visual/page scrolls and using non-standard report sizes** - Minimizing scrolls within visuals and on the page improves user experience. Having a lot of scrolls can be cumbersome for users, especially when trying to understand a complex visualization or dataset. Scrolling can disrupt the flow of data analysis and make it difficult to get a comprehensive view of the information presented.
14. **Use drillthrough buttons instead of expecting users to right-click on data points** - Drillthrough buttons, with context-driven conditional formatting, offer an intuitive way for users to explore data. This approach is more user-friendly than right-click navigation, improving the report.
15. **Optimize visual interactions** - Every visual interacts with other visuals on the page either by cross-filtering or cross-highlighting. Evaluate the interactions which are relevant to the report audience and remove the unnecessary ones. For DirectQuery reports, use **Optimize Ribbon** to optimize reports for interactivity.

**Question 19. How can you integrate Power BI with other Microsoft products like Azure and Office 365?**

**Answer -** Power BI integrates seamlessly with various Microsoft tools, making it an incredibly efficient platform for businesses already using Microsoft products. It connects directly with Excel, allowing users to import and analyze data with just a few clicks. With Power BI, you can also publish your Excel workbooks directly to the cloud, enabling real-time collaboration and data analysis.

For businesses using Azure, Power BI can connect to data stored in Azure SQL Database, Data Lake, and other Azure services. This integration ensures that all your data, whether in the cloud or on-premises, is easily accessible and can be visualized in real-time dashboards.

Power BI also works well with Microsoft Teams, enabling team members to share reports and dashboards effortlessly. You can embed Power BI dashboards directly into Teams channels, facilitating real-time discussions and decision-making based on live data. Moreover, integration with Microsoft Dynamics 365 enables users to combine CRM data with business analytics, offering in-depth insights into customer behavior, sales trends, and more.

Additionally, Power BI connects with SharePoint, allowing organizations to publish reports on their SharePoint sites for easy access. It also integrates with Power Automate (formerly Microsoft Flow), which allows users to create automated workflows based on data changes or set conditions within Power BI.

**Question 20. Explain the concept of aggregations in Power BI.**

**Answer -** Using aggregations, we can store the values of a (large) table aggregated by, for example, a date, a product category, a store location, This way we can visualize these aggregated values on our report. However, the need to enable deeper insight on these aggregated numbers can soon arise. Therefore, we can also go in detail on the data after filtering, without the user noticing anything different. In this blog, we’ll provide you all you need to know to get started with aggregations in Power BI, some pitfalls and some very interesting, newly released, preview features.

Here are some aggregate functions in Power BI:

* **Sum**: Calculates the total of numerical values in a column or expression. It's often used to aggregate sales, revenue, quantities, and other numerical data.
* **Count**: Counts the number of rows in a table or column that contain a value. It's useful for determining the number of occurrences.
* **DISTINCTCOUNT**: Counts the number of distinct values in a column.
* **MIN**: Returns the lowest value in a set of non-NULL values.
* **SUMX**: Takes two arguments: the first is the table or filter context to iterate over, and the second is the expression to evaluate for each row.
* **COUNTA**: Counts all the values in a column, including both non-blank and blank cells. It's useful for assessing the total number of entries in a dataset.
* **Groupby**: Returns a table with records grouped together based on the values in one or more columns.

**Question 21. How do you handle error handling and data quality in Power BI?**

**Answer -** Here are some ways to handle errors and data quality in Power BI:

* **Separate errors from other records**

Create two tables, one for errors and one for good records. You can also direct errors to a hidden table called Error Logs.

* **Use role management**

Deny access to error pages for clients who shouldn't see the information.

* **Remove, replace, or keep errors**

In Power Query, you can remove errors from a column by selecting the column, then choosing Home > Remove Rows > Remove Errors. You can also replace errors by selecting the column, then Transform > Replace Values > Replace Errors.

* **Use the IFERROR function**

In DAX, you can use the IFERROR function to trap and handle errors in an expression.

* **Check data quality**

You can check data quality in Power BI to see things like the min and max value, average, standard deviation, and how many are null.

* **Ensure you have the correct permissions**

Make sure you're logged in with the correct credentials to access your data.

**Question 22. What is the purpose of Power BI Embedded and when would you use it?**

**Answer –**

Purpose-

Developers use Power BI Embedded to visualize application data without having to build it themselves.

* When to use it

Developers and software companies, also known as independent software vendors (ISVs), use Power BI Embedded to:

* + Provide customer-facing reports, dashboards, and analytics
  + Reduce developer resources by automating the monitoring, management, and deployment of analytics
  + Deliver compelling data experiences for end users