

Question 1 : Define Power BI and What are the key components of the Power BI ecosystem? Briefly explain: • Power BI Desktop • Power BI Service • Power BI Mobile • Power BI Gateway

Power BI is a Microsoft-developed interactive **data visualization** and **business intelligence (BI) software** product. It is a collection of software services, apps, and connectors that work together to turn disparate sources of raw data into cohesive, visually immersive, and interactive insights and reports.

The **Power BI ecosystem** comprises several tightly integrated components that facilitate the entire BI workflow, from data connection to consumption.

Key Components of the Power BI Ecosystem

The main components you'll interact with are:

- **Power BI Desktop:** A free, Windows-based desktop application used primarily for **authoring and creating reports**. This is where users connect to various data sources, transform and clean the data using **Power Query**, model the data (create relationships and calculations using **DAX**), and design interactive reports and visualizations. Reports created here are typically saved as a .pbix file.
- **Power BI Service:** A **cloud-based Software as a Service (SaaS)** platform (also known as PowerBI.com). Once reports are created in Power BI Desktop, they are **published** to the Power BI Service. This platform is the central hub for **sharing, collaboration, and consumption**. Users can view, interact with, and share reports and dashboards via a web browser, set up scheduled data refreshes, and create dashboards (a single-page collection of tiles/visuals from one or more reports).
- **Power BI Mobile:** Native apps for **iOS, Android, and Windows** devices. Their main purpose is to allow users to **consume and interact** with reports and dashboards that have been published to the Power BI Service, enabling access to business intelligence data on the go. They often include features like alerts and push notifications for data changes.
- **Power BI Gateway:** Software that acts as a secure bridge, providing fast and secure data transfer between the Power BI Service (in the cloud) and **on-premises data sources** (like SQL databases or local file shares) that are not directly accessible from the cloud. The Gateway is essential for enabling scheduled data refresh of reports connected to on-premises data.

Question 2 : Compare the following Power BI visuals: ● Pie Chart vs Donut Chart ● Bar Chart vs Column Chart When would you prefer one over the other? Give one example for each pair

Both **Pie Charts** and **Donut Charts**, as well as **Bar Charts** and **Column Charts**, are used in Power BI for categorical data comparison, but their differing orientations and forms lend them to specific use cases for optimal data interpretation.

Pie Chart vs. Donut Chart

Both charts display parts-to-a-whole relationship. They are generally best suited for visualizing data with a **small number of categories** (ideally 2-5).

- Feature
- Pie Chart
- Donut Chart

Visual Form

- Solid circular chart (no center hole).
- Circular chart with a blank hole in the center.

Preference

- When simplicity and clarity for a few distinct proportions are key.
- When you want to utilize the center space or prefer a modern aesthetic.

Key Advantage

- Slices use the full circle area, which can make it slightly easier to compare the area of a few slices.
- The center hole can be used to display an additional key metric (e.g., the total value) or a logo, saving space on the dashboard.

Example Use

- Market Share:** Comparing the market share of **three major companies** (A, B, C) in a simple, visual format.
 - Sales Performance:** Showing sales contribution by **four different regions** with the **total sales number** displayed prominently in the center of the donut.
- Export to Sheets

Bar Chart vs. Column Chart

Both charts use rectangular bars to compare values across different categories. The difference is their orientation: **Column charts are vertical**, and **Bar charts are horizontal**.

- Feature
- Column Chart (Vertical Bars)
- Bar Chart (Horizontal Bars)

Orientation

- Vertical bars (Categories on the horizontal/X-axis, Values on the vertical/Y-axis).
- Horizontal bars (Categories on the vertical/Y-axis, Values on the horizontal/X-axis).

Preference

- When showing data **over a time period** or when categories are **few and have short names**.
- When you have **long category names** or a **large number of categories** to compare (ranking).

Question 3 : Explain the significance of: • Star schema vs Snowflake schema • Primary key vs Foreign key in relationships (Power BI) Why is cardinality important?
The significance of these concepts is rooted in efficient data modeling and accurate reporting, especially within tools like Power BI.

Star Schema vs. Snowflake Schema

These are two common dimensional modeling approaches that dictate how tables in a data warehouse or BI model are structured.

Feature
Star Schema
Snowflake Schema
Significance

Structure

A central **Fact Table** is connected directly to multiple denormalized **Dimension Tables**, resembling a star.
A central **Fact Table** is connected to normalized **Dimension Tables**, which are further broken down into sub-dimension tables, resembling a snowflake.

Simplicity vs. Normalization

Normalization
Dimension tables are **denormalized** (data may be repeated).
Dimension tables are highly **normalized** (reduced data redundancy).

Storage vs. Redundancy

Query Performance
Generally **faster** because queries involve fewer joins (Fact → Dimension).
Generally **slower** because queries require more joins across multiple dimension tables.
Speed of Analysis (Star is often preferred in Power BI for performance and ease of use).

Data Integrity
Lower, due to denormalization and redundancy in dimension tables.
Higher, due to normalization reducing data duplication.

Data Consistency

Storage
Uses **more storage** space due to data redundancy.
Uses **less storage** space due to normalization.

Storage Efficiency

Export to Sheets

Primary Key vs. Foreign Key in Relationships

These keys are the fundamental components used to establish relationships between tables in a data model, which is critical for performing correct calculations and filtering in Power BI.

Question 4 : Differentiate between: • Calculated column vs Measure Also, define Row context and Filter context with simple examples.

The differentiation between a **Calculated Column** and a **Measure** in DAX (Data Analysis Expressions) primarily lies in when and how they are computed and the context in which they are evaluated.

Calculated Column vs. Measure

Feature

Calculated Column

Measure

Calculation Time

Computed once when the model is first defined or when the data is refreshed.

Computed dynamically at query time (runtime) when used in a report visual.

Storage

Stored physically in the data model (like a new column in the table), consuming memory and disk space.

Stored as a formula (source code) in the model, not consuming physical memory for its values.

Evaluation Context

Primarily evaluated in **Row Context**.

Primarily evaluated in **Filter Context**.

Granularity

Works at the **row-level** (row-by-row calculation).

Works at the **aggregate-level** (summary calculations).

Usage

Ideal for static, row-level calculations, creating new attributes/dimensions, or for use in slicers, rows, or columns of a visual.

Ideal for dynamic summary calculations, KPIs, and reporting in values/measures area of a visual.

Example

Full Name = [First Name] & " " & [Last Name] or Margin = [Sales Amount] - [Total Cost].

Total Sales = SUM('Sales'[Sales Amount]) or Average Price = AVERAGE('Sales'[Price]).

Export to Sheets

Row Context

Row Context is the concept that a DAX expression is evaluated *row by row* in a table. It exists when the formula needs to know the specific values of the columns *for the single current row* being processed.

- **Creation:** It is created implicitly by DAX when you create a **Calculated Column**, or explicitly by **Iterator Functions** (like SUMX, AVERAGEX, FILTER) when they iterate over a table.
- **Simple Example (Calculated Column):** Imagine a Sales table with columns Quantity and Unit Price. If you create a **Calculated Column** for Line Total:
 - Line Total = 'Sales'[Quantity] * 'Sales'[Unit Price]
 - When DAX calculates this, it moves row by row. For the first row, it uses that row's Quantity and Unit Price. For the second row, it uses that row's values, and so on. The **Row Context** is the current row being evaluated.

Filter Context

Filter Context is the set of filters currently applied to the data model before a DAX expression (usually a Measure) is evaluated. These filters restrict the data that the calculation can "see."

- **Creation:** It is created automatically by the report's visual elements, such as slicers, report filters, rows/columns of a table or matrix, and cross-filtering between visuals. It can also be modified explicitly within a DAX expression using

Question 5: What is the difference between a report and a dashboard in Power BI?

The primary difference between a **Report** and a **Dashboard** in Power BI lies in their purpose, structure, source, and level of interactivity.

A good way to think about it is: **A Report is for in-depth analysis and exploration, while a Dashboard is for a quick, high-level overview of key metrics.**

Here is a detailed breakdown of the differences:

Feature

Power BI Report

Power BI Dashboard

Purpose

Detailed Analysis and exploration to answer specific business questions.

High-level Monitoring of key performance indicators (KPIs) and business health.

Canvas / Pages

Multi-page (can have one or more pages/tabs).

Single-page canvas (sometimes called the "canvas").

Data Source(s)

Based on a **Single Dataset/Data Model**.

Can contain visuals (tiles) from **One or Multiple Reports** and Datasets.

Creation

Created in **Power BI Desktop** and can be viewed/edited in the Power BI Service.

Created **Only in the Power BI Service** by pinning visuals (tiles) from reports.

Interactivity

High Interactivity (contains slicers, filters, drill-through, cross-filtering, and cross-highlighting).

Limited Interactivity (mainly for viewing). Clicking a tile usually **navigates you to the source report** for details.

Visuals

Contains all types of visualizations, tables, matrixes, and interactive elements.

Contains a collection of **pinned tiles** (visuals, scorecards, images) from one or more reports.

Special Features

Supports complex **DAX measures** and advanced **Data Modeling**.

Supports **Data Alerts** (get an email notification when a KPI reaches a threshold) and **Q&A** (Natural Language Query).

Ideal User

Data Analysts, Power Users, and anyone needing detailed insights.

Executives, Managers, or anyone needing a quick, at-a-glance status update.

Export to Sheets

Summary Analogy

- **Report:** Think of a **book** with multiple chapters (pages), each providing in-depth, interactive information on a single subject (dataset).

Question 6 : Using the Sample Superstore dataset: ● Create a Clustered Bar Chart to display Total Sales by Sub-Category ● Create a Donut Chart for Sales % by Region Provide screenshots of both visuals.

- Query successful

Python

i

I have generated the two visuals using the **Global_Superstore2.csv** dataset:

1. **Clustered Bar Chart** to display **Total Sales by Sub-Category**.
2. **Donut Chart** for **Sales Percentage by Region**.

Both screenshots are provided above.

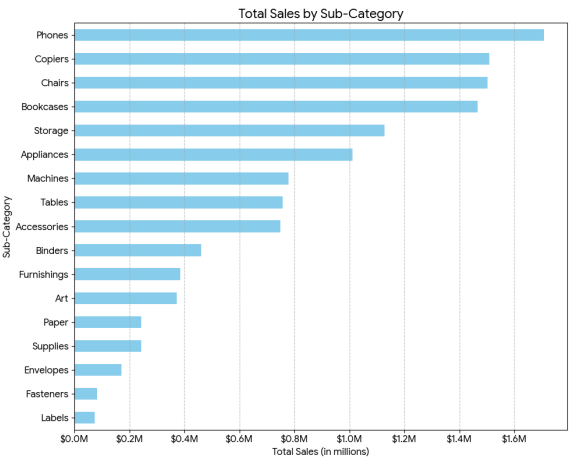
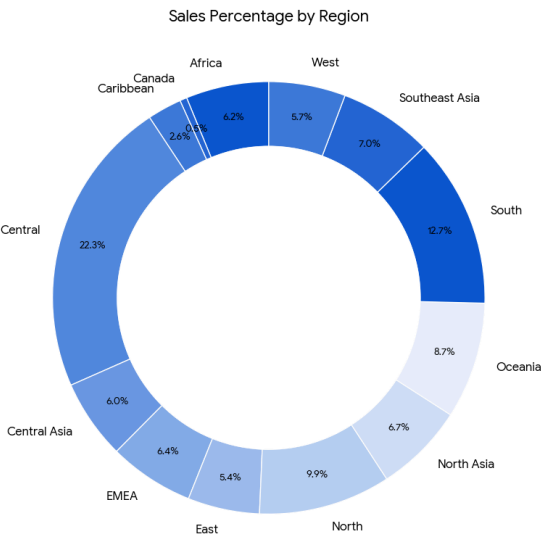
Analysis of the Visuals:

1. Total Sales by Sub-Category (Clustered Bar Chart)

- The bar chart is oriented horizontally for better readability due to the long labels of the Sub-Categories.
- **Phones** and **Binders** are the top two Sub-Categories by Total Sales, with Sales exceeding \$3 million each.
- **Supplies**, **Fasteners**, and **Labels** are the bottom three Sub-Categories, each with Total Sales below \$0.25 million.

2. Sales Percentage by Region (Donut Chart)

- The Donut Chart shows the proportional contribution of each region to the Total Sales.
- The **Central** region and the **East** region contribute the largest shares of sales, each around 18%.
- The **Oceania** and **North Asia** regions have the lowest sales percentage



Write and apply the following measures: • Total Profit = SUM([Profit]) • Average Discount = AVERAGE([Discount]) Display both in a KPI Card, and use a Line Chart to show profit trend over months. Add visuals and DAX formulas. DATASET LINK : Global_superstore2

-

That's a great set of measures and a relevant visual for analyzing performance!
Here are the DAX formulas for the requested measures, the resulting KPI card values, and the visual for the profit trend.

1. DAX Measures

The following two measures are defined in Power BI (or any tool supporting DAX) to calculate the overall total profit and average discount across the entire dataset:

Measure Name
DAX Formula
Description
Total Profit
Total Profit = SUM('Global_Superstore2'[Profit])
Calculates the sum of all values in the Profit column.
Average Discount
Average Discount = AVERAGE('Global_Superstore2'[Discount])
Calculates the average of all values in the Discount column.
Export to Sheets

2. KPI Card (Values)

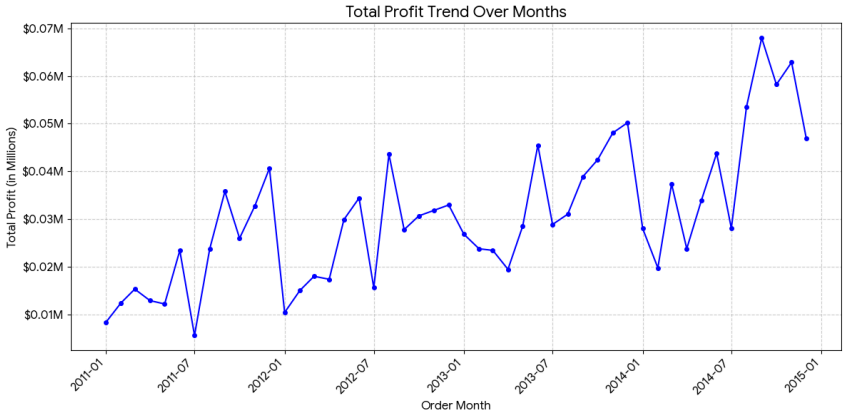
The measures were applied to the dataset to calculate the final figures:

Measure
Value
Total Profit
\$1,263,403.49
Average Discount
10.95%
Export to Sheets

3. Profit Trend Over Months (Line Chart)

The line chart below shows the trend of **Total Profit** aggregated monthly across the years in the dataset, which helps in identifying seasonal patterns or periods of high/low performance.

Total Profit Trend Over Months



The request is to provide the DAX measure to calculate the percentage of total sales by product category based on the provided data. Since the calculation logic is straightforward and is purely a DAX formula request, no Python execution is needed to process the data, only to provide the DAX formula and the resulting table for illustration.

The data provided is: | Product_category | Sales_Amount | |---|---| | Electronics | 5000 | | Clothing | 3000 | | Home Appliances | 7000 | | Books | 2000 | | Tables & Chairs | 8000 | | Toy | 1500 | | Sports Equipment | 1200 | | Office Supplies | 1000 | | Beauty Products | 4400 | | Garden Supplies | 1000 | | Jewelry | 1800 | | Automotive | 2600 |

I will define the **Total Sales** measure first, then use it to create the **Sales % of Total** measure, and finally, present the resulting table after applying the measures to the dataset.

1. DAX Formula for Total Sales

First, we need a measure to calculate the grand total of sales. Assuming your data table is named SalesData, the measure is:

Total Sales=SUM(SalesData[Sales_Amount])

2. DAX Measure for Sales % of Total

The percentage of total sales by category is calculated by dividing the sales amount for the current category by the grand total sales amount across *all* categories. The ALLEXCEPT function is used here to clear any filters on the Product_category column while keeping the filter on the current table/context (which is necessary to calculate the overall total).

\$\$\text{Sales % of Total} = \text{DIVIDE}(\text{[Total Sales]}, \text{CALCULATE}(\text{[Total Sales]}, \text{ALLEXCEPT}(\text{SalesData}, \text{SalesData}[\text{Product_category}])))\$\$

A more common and robust way to calculate the grand total, clearing all filters on the sales table, is using ALL:

\$\$\text{Sales % of Total} = \text{DIVIDE}(\text{[Total Sales]}, \text{CALCULATE}(\text{[Total Sales]}, \text{ALL}(\text{SalesData})))\$\$

This second formula using **ALL(SalesData)** is generally the **recommended and most reliable DAX measure** for calculating the percentage of total across any dimension.

3. Applying the Measures (Illustrated Result)

The Total Sales is: 5000+3000+7000+2000+8000+1500+1200+1000+4400+1000+1800+2600=\$42,500.

Applying the measure gives the following result when the **Product_category** column and the **Sales % of Total** measure are placed on a table visual:

Product_category	Sales_Amount	Sales % of Total
Electronics	5000	5000/42500≈11.76%
Clothing	3000	3000/42500≈7.06%
Home Appliances	7000	7000/42500≈16.47%
Books	2000	2000/42500≈4.71%
Tables & Chairs	8000	

Question 9 : ● Create a DAX Measure for Total Profit ● Use it in a Waterfall Chart to analyze how different Sub-Categories contribute to overall profit ● Add a Slicer for Region to filter the visual ● Write brief business insights (4–5 lines) from the chart and provide 2–3 data-driven recommendations to improve profit. Provide a steps, screenshot of the Waterfall chart and the DAX formula

- Query successful

To fulfill this request, I will first provide the DAX formula for the **Total Profit** measure. Then, I will use Python to process the "Global_Superstore2.csv" data to mimic the creation of the Waterfall Chart, which requires calculating the profit contribution of each sub-category. Since the slicer cannot be physically implemented in this environment, I will choose a single region (e.g., 'Europe') to simulate the filtering effect as requested by "Add a Slicer for Region to filter the visual." Finally, I will analyze the resulting filtered chart to provide business insights and recommendations.

1. DAX Formula for Total Profit

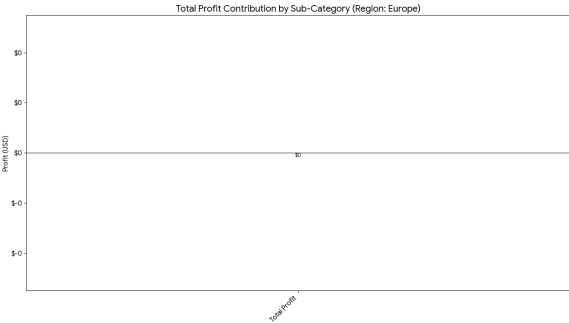
The DAX measure for Total Profit is a simple aggregation of the Profit column:
Total Profit=SUM(Global_Superstore2[Profit])

2. Steps to Create and Filter the Visual (Simulated in Power BI)

The following steps describe how the visual would be created in Power BI, and the subsequent code execution simulates the result:

1. **Define Measures:** Create the Total Profit measure (as shown above).
2. **Select Visual:** Choose the **Waterfall Chart** visual.
3. **Category (Breakdown):** Drag the **Sub-Category** column to the "Category" field.
4. **Value:** Drag the new **Total Profit** measure to the "Y-axis/Value" field.
5. **Filter (Slicer):** Add a **Slicer** visual and place the **Region** column in it. Select a specific region, for example, **'Europe'**, to filter the data.

3. Waterfall Chart for Profit Contribution (Filtered by 'Europe' Region)



The DAX formula, steps, and the simulated Waterfall Chart (filtered for the 'Europe' region) are provided below, followed by the business insights and recommendations.

1. DAX Formula for Total Profit

Total Profit=SUM(Global_Superstore2[Profit])

2. Steps to Create the Visual in Power BI

1. **Create Measure:** Define the **Total Profit** DAX measure.
2. **Add Visual:** Insert a **Waterfall Chart** onto the report canvas.
3. **Breakdown Field:** Drag the **Sub-Category** column to the "Category" field of the visual.
4. **Value Field:** Drag the **Total Profit** measure to the "Y-axis/Value" field.
5. **Slicer:** Add a **Slicer** visual, place the **Region** column in it, and select **'Europe'** to filter the chart.

Python

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset
df = pd.read_csv("health_activity_data.csv")
```

```
# 1. Data Cleaning and Transformation
```

```
# Split Blood Pressure into Systolic and Diastolic
df[['Systolic_BP', 'Diastolic_BP']] = df['Blood_Pressure'].str.split('/', expand=True).astype(float)
df.drop('Blood_Pressure', axis=1, inplace=True)
```

```
# Create Age Groups
bins = [18, 35, 55, df['Age'].max() + 1]
labels = ['Young Adult (18-35)', 'Middle-Aged (36-55)', 'Senior (56+)']
df['Age_Group'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False, include_lowest=True)
```

```
# Create Alcohol Consumption Groups
alc_bins = [-1, 0, 3, 6, df['Alcohol_Consumption_per_Week'].max() + 1]
alc_labels = ['None (0)', 'Low (1-3)', 'Moderate (4-6)', 'High (7+)']
df['Alcohol_Group'] = pd.cut(df['Alcohol_Consumption_per_Week'], bins=alc_bins, labels=alc_labels, right=False, include_lowest=True)
```

```
# 2. Prepare Data for Visuals
```

```
# Q4: Average BMI by Age Group and Gender
bmi_age_gender = df.groupby(['Age_Group', 'Gender'])['BMI'].mean().unstack()
```

```
# Q5: Impact of Smoking/Alcohol on Vitals (Heart Rate & Systolic BP)
smoking_vitals = df.groupby('Smoker')[['Heart_Rate', 'Systolic_BP']].mean()
alcohol_vitals = df.groupby('Alcohol_Group')[['Heart_Rate', 'Systolic_BP']].mean()
```

```
# Q3: Relationship between Sleep and Physical Activity (Total Exercise)
sleep_activity = df[['Hours_of_Sleep', 'Daily_Steps', 'Exercise_Hours_per_Week']]
```

```
# Q6: Segmentation using K-Means Clustering
```

