GROUP:

SAI DISHA – 231057026

ANJALI K M – 231057028

AISHWARAYA S YELI - 231057036

POWER BI ASSIGNMENT

1. LIST THE TASKS PERFORMED IN DATA ANALYSIS AND DEMONSTRATE FIVE CORE COMPONENTS OF ANALYTICS.

Data analysis involves various tasks and encompasses multiple components in order to gain insights and make data-driven decisions:

Tasks in Data Analysis:

(1) **Data Collection:** Gathering relevant data from various sources, which may include databases, files, surveys, APIs, or sensor data.

(2) **Data Cleaning:** Preprocessing and cleaning the data to handle missing values, outliers, and inconsistencies to ensure data quality.

(3) **Data Exploration:** Exploring the dataset to understand its structure, relationships, and potential patterns or trends.

(4) **Data Transformation:** Preparing data for analysis by performing operations like aggregating, encoding categorical variables, and feature engineering.

(5) **Data Visualization:** Creating visual representations of data, such as charts, graphs, and dashboards, to better understand and communicate findings.

(6) **Statistical Analysis:** Applying statistical techniques to uncover patterns, correlations, and make predictions.

(7) **Machine Learning:** Using machine learning models to build predictive and descriptive models based on the data.

(8) **Hypothesis Testing:** Formulating and testing hypotheses to draw conclusions about the data.

(9) **Model Evaluation:** Assessing the performance of models and ensuring they meet the desired criteria.

(10) **Report Generation:** Documenting and presenting analysis results, often in the form of reports, presentations, or dashboards.

Five Core Components of Analytics:

(1) **Data:** Data is the foundation of analytics. It includes raw data from various sources, cleaned and processed data, and historical data. The quality, volume, and relevance of data are essential.

(2) **Analytics Tools and Software:** These are the software and tools used to perform data analysis, from spreadsheet software like Excel to specialized analytics platforms like R, Python, or Power BI.

(3) **Analytics Models and Algorithms:** Models and algorithms are used to make sense of data, generate insights, and make predictions. These can range from simple statistical models to complex machine learning models.

(4) **Visualization and Reporting:** The ability to present data and findings visually through charts, graphs, dashboards, and reports. Effective visualization makes it easier to understand and communicate insights.

(5) **People and Expertise:** Skilled analysts, data scientists, and domain experts who can interpret data, choose the right models, and derive actionable insights. Domain knowledge is critical for understanding the context of data.

These core components work together to transform data into valuable insights and support informed decision-making. Data analysis leverages these components to address business challenges, improve processes, and discover opportunities for growth and optimization.

2. DESCRIBE FIVE JOB ROLES IN THE OVERALL SPECTRUM OF DATA DISCOVERY AND UNDERSTANDING.

In the field of data discovery and understanding, there are several job roles that play essential roles in various aspects of the data lifecycle. Here are five key job roles in this domain:

(1) **Data Analyst:**

Responsibilities: Data analysts are responsible for collecting, processing, and analyzing data to discover insights, trends, and patterns. They create reports, dashboards, and visualizations to communicate findings.

Skills: Proficiency in data analysis tools, database querying, statistical analysis, and data visualization. Strong analytical and problem-solving skills.

(2) Data Scientist:

Responsibilities: Data scientists go beyond data analysis to design and implement complex machine learning models and algorithms. They develop predictive and prescriptive models to solve business problems.

Skills: Strong programming skills (Python, R), machine learning expertise, data engineering, and domain knowledge in the industry they work in.

(3) Data Engineer:

Responsibilities: Data engineers focus on data acquisition, data transformation, and data warehousing. They design and maintain data pipelines, ensuring that data is collected and stored efficiently.

Skills: Proficiency in data engineering tools, ETL (Extract, Transform, Load) processes, database management, and programming. Knowledge of Big Data technologies is a plus.

(4) Business Intelligence (BI) Analyst:

Responsibilities: BI analysts are responsible for creating and maintaining dashboards, reports, and interactive visualizations. They provide business stakeholders with real-time insights for decision-making.

Skills: Proficiency in BI tools (e.g., Tableau, Power BI), SQL, data modeling, and data visualization. Strong communication and collaboration skills.

(5) Data Quality Analyst:

Responsibilities: Data quality analysts focus on ensuring data accuracy, consistency, and completeness. They create data quality standards, perform data profiling, and implement data cleansing and validation procedures.

Skills: Understanding of data quality frameworks, data profiling tools, SQL, and data validation techniques. Attention to detail and problem-solving skills.

These job roles collectively contribute to the data discovery and understanding process. They work in synergy to collect, clean, analyse, and communicate data-driven insights to support informed decision-making within an organization.

3. DESCRIBE FIVE KEY AREAS IN THE DATA ANALYSIS PROCESS.

The data analysis process involves several key areas, each of which is crucial for deriving meaningful insights from data. Here are five key areas in the data analysis process:

(1) Data Collection:

Description: Data collection is the initial step where raw data is gathered from various sources. This can include databases, spreadsheets, APIs, sensors, surveys, and more.

Tasks: Collect, extract, and ingest data from different sources into a centralized repository or data warehouse.

Challenges: Ensuring data accuracy, completeness, and data privacy compliance.

(2) Data Cleaning and Preparation:

Description: Raw data often contains errors, missing values, and inconsistencies. Data cleaning involves identifying and rectifying these issues to make the data suitable for analysis.

Tasks: Data cleaning, data transformation, handling missing values, outlier detection, and data normalization.

Challenges: Identifying and addressing data quality issues, maintaining data consistency.

(3) Exploratory Data Analysis (EDA):

Description: EDA is the process of visually and statistically exploring data to understand its characteristics, patterns, and relationships between variables.

Tasks: Data visualization, summary statistics, correlation analysis, and data profiling.

Challenges: Identifying meaningful patterns, insights, and outliers in the data.

(4) Data Analysis and Modeling:

Description: In this phase, data analysts use statistical techniques and machine learning algorithms to analyze and model the data. The goal is to gain insights, make predictions, and identify trends.

Tasks: Hypothesis testing, regression analysis, clustering, classification, and predictive modeling.

Challenges: Selecting the appropriate analysis techniques, model overfitting, and interpreting results.

(5) **Data Visualization and Communication:**

Description: Communicating data findings is crucial for decision-making. Data visualization is the process of creating charts, graphs, and dashboards to convey insights effectively.

Tasks: Data visualization design, report generation, and storytelling with data.

Challenges: Designing clear and informative visualizations, tailoring insights for the audience.

These key areas are interconnected and iterative, meaning that data analysts often cycle back and forth between them to refine their analysis. Effective data analysis requires a combination of technical skills, domain knowledge, and critical thinking to transform raw data into actionable insights.

4. DESCRIBE BUILDING BLOCKS OF POWER BI.

Power BI is a powerful data visualization and business intelligence tool that allows users to create interactive reports and dashboards. The basic building blocks in Power BI include: Visualizations, Dataset, Reports, Dashboards, Tiles.

- **Visualizations:**

  Description: Visualizations are the individual charts, graphs, and tables that display the data in a meaningful way. Power BI offers a wide range of visualization types, such as bar charts, line charts, maps, and tables.

  Use Case: You can create a bar chart to visualize sales data, a map to show the geographical distribution of customers, or a table to display detailed information.

- **Datasets:**

  Description: Datasets are collections of data that you import or connect to in Power BI. They serve as the foundation for your reports and dashboards. Datasets can include data from various sources and can be transformed and modeled within Power BI.

Use Case: You might create a dataset that combines data from your sales database, CRM system, and external sources for comprehensive analysis.

- **Reports:**

  Description: Reports are interactive pages or canvases where you organize and display your visualizations. You can create multiple reports within a single Power BI file, and each report can focus on different aspects of your data.

  Use Case: You could create a report that provides an overview of sales performance, another report for customer analysis, and a third report for product insights.

- **Dashboards:**

  Description: Dashboards are a way to display key visualizations and insights from one or more reports on a single page. Dashboards allow you to monitor your data at a glance and provide interactivity.

  Use Case: A dashboard might feature a sales overview, showing important KPIs, customer demographics, and product performance on one screen.

- **Tiles:**

  Description: Tiles are individual containers for visualizations, charts, or images. You can pin visualizations from reports to dashboards as tiles to create a custom layout.

  Use Case: You can pin a specific sales chart from a report to a dashboard as a tile for quick reference.

Power BI allows users to create, customize, and share these building blocks to generate interactive and insightful data-driven content. Whether you are analyzing sales data, tracking marketing metrics, or monitoring operational performance, Power BI's building blocks offer a flexible and powerful platform for data visualization and analysis.

5. DEMONSTRATE FACT AND DIMENSION TABLE WITH AN EXAMPLE AND LIST THEIR CHARACTERISTICS.

Fact Table:

- In a data warehouse, a fact table is a table that stores the measurements, metrics, or facts related to a business operation.
- It is located at the center of a star or snowflake schema and is surrounded by dimension tables.

- When multiple fact tables are used, they can be organized using a "fact constellation schema."
- A fact table has two types of columns: those that contain the facts and those that serve as foreign keys linking to dimension tables.
- The primary key of a fact table is often a composite key made up of all of the foreign keys in the table.
- Fact tables can hold various types of measurements, such as additive, non-additive, and partly additive measures, and store important information in the data warehouse.
- They are useful for evaluating dimensional attributes because they provide additive values that can act as independent variables.

Characteristics of a Fact Table

The following are some of the characteristics of a fact table:

- **Outrigger Dimensions**: Outrigger dimensions are dimensions that refer to any other dimension table.
- **Keys**: Each fact table has a key that is made up of the primary keys from all of the dimension tables connected to that fact table. A concatenated key is one such key that specifically identifies the row of the fact table.
- **Additive Measures**: The fact table's attributes may be entirely, partially, or not at all additive. Measures that are applied to all dimensions are referred to as fully addition or additive measures. Quasi-measures are those that store the fundamental unit of measurement for any business process, whereas semi-additive measures add measures to some dimensions but not to all.
- **Fact Table Grain**: The level of detail or depth of the information recorded in a fact table is referred to as the table's grain. A successful fact table must be designed at the highest level.
- **Degenerated Dimensions**: A degenerated dimension is any dimension or attribute that is available in the fact table but cannot be added to or is non-additive.
- **Sparse Data**: Some records in the fact table have characteristics with null values or measurements, which means that the information they contain is not given or provided.

- **Shrunken Rollup Dimensions**: The dimensions that are created by subdividing the base dimension's columns and rows are known as shrunken rollup dimensions.

Dimension table:

- Dimension tables contain descriptions of the objects in a fact table and provide information about dimensions such as values, characteristics, and keys.
- These tables are usually small, with a number of rows ranging from a few hundred to a few thousand.
- The term "dimension table" refers to a set of data related to any quantifiable event and is the foundation for dimensional modeling.
- Dimension tables have a column that serves as a primary key, allowing each dimension row or record to be uniquely identified. This key is used to link the dimension table to the fact tables. A surrogate key, which is a system-generated key, is often used to uniquely identify the rows in the dimension table.

Characteristics of a Dimension Table:

The following are some of the characteristics of a dimension table:

- **Relationship Between Attributes**: Although they are all included in the same dimension table, the attributes in it typically do not have a direct relationship with one another.
- **Records**: The dimension table contains more characteristics than records.
- **Keys**: The main key is required for every dimension table in order to help uniquely identify each entry.
- **Normalization**: The dimension table is not normalized because doing so splits the data into different tables and makes it more difficult for queries to execute quickly because they must go through these extra tables to retrieve measurements from the fact table for each corresponding attribute in the dimension table.
- **Attributes**: Because the dimension table has so many attributes, it looks to be expanding horizontally.
- **Drilling Down, Rolling Up**: The presence of attributes in a dimension table enables the extraction of information by drilling down from a higher level to a lower level or by rolling up from a lower level to a higher level of the attributes.

- **Attribute Values**: The majority of the values in the dimension table are expressed as text rather than numbers.

6. DISCUSS TEN DIFFERENCES BETWEEN FACT AND DIMENSION TABLE.

Basis of Distinction	Fact Table	Dimension Table
Definition	Facts about a business process, such as measurements or metrics.	Descriptive characteristics in the companion table to the fact table can be utilized as query constraints.
Characteristics	Positioned in the middle of a snowflake or star schema, surrounded by dimensions.	The edges of the snowflake or star schema, attached to the fact table,
Design	Defined by their grain or at the atomic level.	It must be extensive, in-depth, and of the highest calibre.
Task	A fact table is a quantifiable event for which data from a dimension table is gathered and used for reporting and analysis.	Gathering of background data about a company.
Data Type	Facts tables may include data about sales in relation to a number of parameters, such as Product and Date.	Each dimension table has attributes that provide information about the specifics of the dimension. For instance, product dimensions may include the product ID, category, etc.
Key	The fact table's primary key is mapped as a foreign key to dimensions.	Each dimension in a dimension table contains a primary key column that uniquely identifies it.
Storage	Helps to save report labels and filter domain values in dimension tables.	Load dimensional structures with thorough atomic data.

Hierarchy	Contains no hierarchy.	Hierarchies are present. For instance, Location could include a country, state, city, zip code, and more.

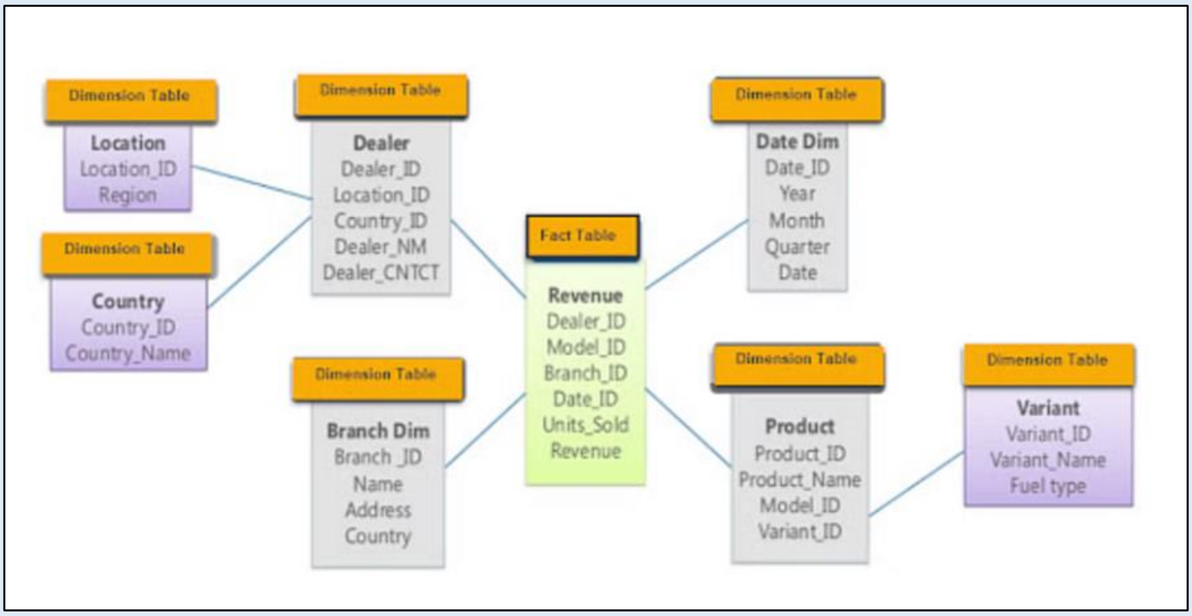
7. DEMONSTRATE STAR AND SNOW FLAKE SCHEMA WITH A USE CASE AND ILLUSTRATION IN DETAIL.

Snowflake schema:

- This particular kind of data warehouse schema is shaped like a snowflake. The snowflake schema aims to normalize the star schema's denormalized data.
- When the star schema's dimensions are intricate, highly structured, and have numerous degrees of connection, and the kid tables have several parent tables, the snowflake structure emerges. Some of the star schema's common issues are resolved by the snowflake schema.
- The snowflake schema can be thought of as a "multi-dimensional" structure. A snowflake schema's central component comprises Fact Tables that link the data inside the Dimension Tables, which then radiate outward like the Star Schema. The snowflake schema, on the other hand, divides the Dimension Tables into several tables, resulting in a snowflake pattern. Up until they are fully normalized, the Dimension Tables are split across multiple tables.

Example of Snowflake Schema

In the following example of a snowflake schema, the Country table has been further normalized into its own separate table:



Characteristics:

- The snowflake schema is an extension of the star schema, with dimension tables normalized into sub-dimensions.
- Normalization reduces data redundancy but can make queries more complex.
- In this example, DateDim has been normalized into Month, and ProductDim has been further normalized into Category.
- The RegionDim table is also added as a sub-dimension.
- While this approach saves storage space, it can require more complex joins when querying.

Star Schema:

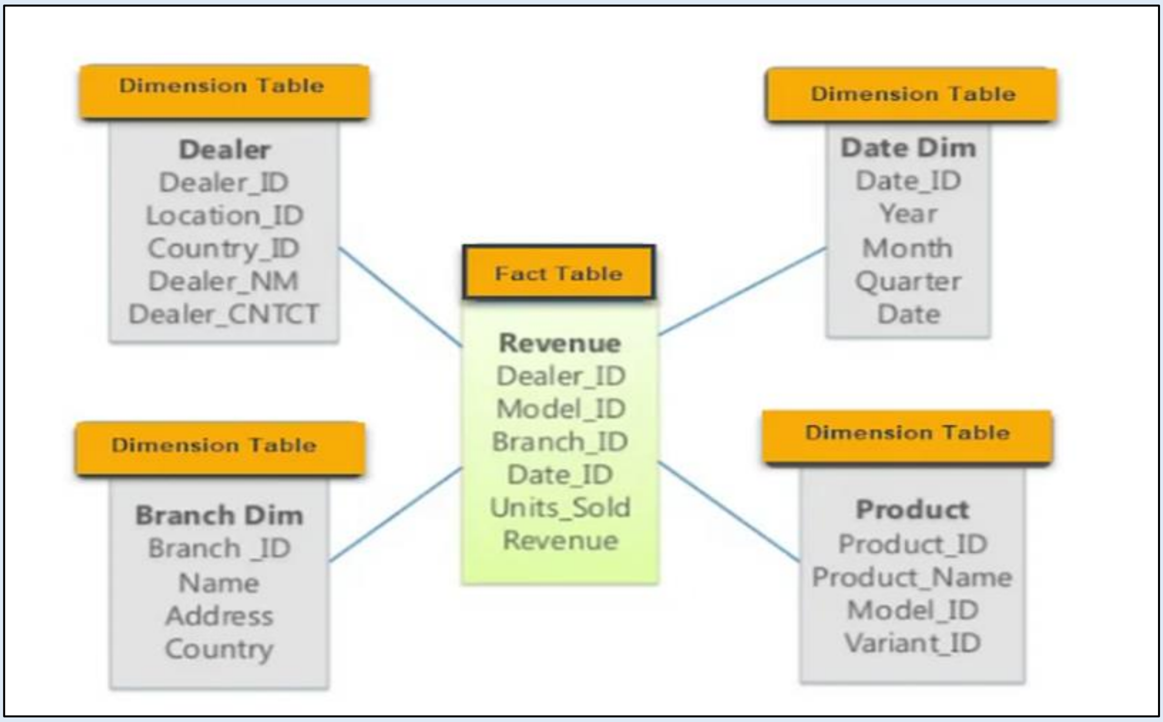
- The star schema is the most straightforward method for arranging data in the data warehouse. Any or even more Fact Tables that index a number of Dimension Tables may be present in the star schema's central area. Dimensions Keys, Values, and Attributes are found in Dimension Tables, which are used to define Dimensions.
- The star schema's objective is to distinguish between the descriptive or "DIMENSIONAL" data and the numerical "FACT" data that pertains to a business.
- The information displayed in a numerical format, such as cost, speed, weight, and quantity, might be considered fact data. Along with numbers, dimensional data can

also contain non-numerical elements like colors, places, names of salespeople and employees, etc.

- While the Dimension Data is contained inside the Dimension Tables, the Fact Data is arranged within the Fact Tables. In a star schema, the Fact Tables are the integrating points at the core of a star.

Example of Star Schema:

In the following example of a star schema, the central fact table contains keys to each dimension table (such as Dealer_ID, Model_ID, Date_ID, Product_ID, Branch_ID) and attributes like units sold and revenue:



Characteristics:

- The star schema consists of a central fact table connected to multiple dimension tables.
- Fact tables contain measures like sales revenue, quantity sold, and profit.
- Dimension tables contain descriptive attributes such as date details, product information, store attributes, and sales region.
- The schema is straightforward and easy to understand, making it efficient for querying and reporting.
- Each dimension table directly links to the central fact table, making data retrieval faster.

8. DEMONSTRATE DIFFERENT TYPES OF RELATIONSHIP PRESENT BETWEEN TABLES WITH AN EXAMPLE.

The different types of relationships can be represented in Power BI with examples as follows:

(1) **One-to-One (1:1)**

A one-to-one relationship is less common because it typically doesn't require separate tables. However, you might encounter it when dealing with data from different sources that need to be combined into a single dataset.

Ex: Suppose a Power BI report combining data from two sources: Sales data and Customer data. Each sales transaction has one customer, and each customer has one sales transaction. In this case, you can create a one-to-one relationship between the Sales table and the Customer table using a common field like CustomerID.

(2) **One-to-Many (1:N)**

This is the most common type of relationship. It's used when one table relates to multiple records in another table.

Ex: Imagine a report for a retail company. You have a Sales table and a Products table. Each sale can have multiple products, but each product belongs to one sale. The Sales table and Products table can be linked using a SaleID.

(3) **Many-to-Many (N:N)**

Power BI handles many-to-many relationships by using a bridge table, which acts as the intermediate junction table connecting the two tables.

Ex: Let's consider a scenario where we have a report for a university. You have a students table and a Courses table. Each student can enroll in multiple courses, and each course can have multiple students. To handle this many-to-many relationship, you create an Enrollments bridge table. This table contains StudentID and CourseID, allowing you to establish the relationships between Students, Courses, and Enrollments.

9. DEFINE DAX. DISCUSS THE FEATURES OF POWER BI, HOW DAX FORMULAS ARE WRITTEN.

DAX or Data Analysis Expressions drive all the calculations you can perform in Power BI. DAX formulas are versatile, dynamic, and very powerful – they allow you to create new fields and even new tables in your model. While DAX is most commonly associated with Power BI, you can also find DAX formulas in Power Pivot in Excel and SQL Server Analysis Services (SSAS).

DAX formulas are made up of 3 core components:

- **Syntax** – Proper DAX syntax is made up of a variety of elements, some of which are common to all formulas.
- **Functions** – DAX functions are predefined formulas that take some parameters and perform a specific calculation.
- **Context** – DAX uses context to determine which rows should be used to perform a calculation.

Key Features of Power BI:

The following features of Power BI make it so popular in today's market:

- **Data Selection:** Power BI provides you with options to filter the datasets and create relevant small chunks of data. This will allow you to focus on individual datasets and perform more insightful Data Analysis.
- **Responsiveness:** Power BI relies on highly responsive Navigation Trees and Dashboards on all platforms like Android, iOS, etc. This allows you to get a detailed insight into any Report or Graph without any glitches.
- **Seamless Functionalities:** Power BI offers you simple drag-and-drop functionalities to create customized reports. This way, even if you have zero to no technical experience, you can easily analyse & visualize your data. Furthermore, Power BI's platform works on Natural Language Queries. This safeguards you from putting time and resources into learning a complex query language.
- **Data Connectivity:** Power BI contains a multitude of connectors that allows it to integrate with various external data sources. This way you can directly connect your

preferred Data Sources to Power BI seamlessly and perform high-level Data Analytics and visualization.

Writing DAX Formulas in Power BI:

DAX formulas are intuitive and easy to read. This makes it easy to understand the basics of DAX so you can start writing your own formulas relatively quickly. Let's go over the building blocks of proper DAX syntax.

- The name of the measure or calculated columnThe equal-to operator ("=") indicates the start of the formula
- A DAX function
- Opening (and closing) parentheses ("()")
- Column and/or table references
- Note that each subsequent parameter in a function is separated by a comma (",")

DAX functions can also be nested inside each other to perform multiple operations efficiently. This can save a lot of time when writing DAX formulas. For example, it is often useful to have multiple nested IF statements or to use the IFERROR function to wrap around another function, so that any errors in the formula are represented by the value you specify.

Some of the most common DAX functions used in reports are:

- **Simple calculations:** COUNT, DISTINCTCOUNT, SUM, AVERAGE, MIN, MAX.
- **SUMMARISE:** Returns a table typically used to further apply aggregations over different groupings.
- **CALCULATE:** Performs an aggregation along with one or more filters. When you specify more than one filter, the function will perform the calculation where all filters are true.
- **IF:** Based on a logical condition, it will return a different value for if it is true or false. This is similar to the CASE WHEN operation in SQL.
- **IFERROR:** Looks for any errors for an inner function and returns a specified result
- **ISBLANK:** Checks if the rows in a column are blank and returns true or false. Useful to use in conjunction with other functions like IF.

- **EOMONTH:** Returns the last day of the month of a given date (column reference in a date format) for as many months in the past or the future.
- **DATEDIFF**: returns the difference between 2 dates (both as column references in date formats) in days, months, quarters, years, etc.

10. DEMONSTRATE THE FOLLOWING DAX FORMULAS WITH A CONTEXT: COUNT, DISTINCTCOUNT, SUM, AVERAGE, MIN, MAX, SUMMARISE, CALCULATE, IF, IFERROR, ISBLANK, EOMONTH, DATEDIFF.

To demonstrate various DAX formulas, we'll use a simple example involving sales data. Let's assume we have a dataset of sales transactions with information about sales amounts, product categories, and transaction dates.

Sample Data:

Transaction ID	Date	Product Category	Sales Amount
1	2023-01-01	Electronics	500
2	2023-01-05	Clothing	300
3	2023-01-10	Electronics	600
4	2023-01-12	Books	150
5	2023-01-15	Clothing	400

Let's demonstrate the DAX formulas:

(1) COUNT: Count the number of rows in a table.
Count of Transactions = COUNT(SalesData)
The result will be 5 because there are 5 rows in the "SalesData" table.

(2) DISTINCTCOUNT: Count the number of unique values in a column.
Distinct Product Categories = DISTINCTCOUNT(SalesData[Product Category])
The result will be 3 because there are three unique product categories in the "SalesData" table.

(3) SUM: Calculate the sum of values in a column.
Total Sales Amount = SUM(SalesData[Sales Amount])
The result will be 1950 because it sums up the sales amounts in the "SalesData" table.

(4) AVERAGE: Calculate the average of values in a column.

Average Sales Amount = AVERAGE(SalesData[Sales Amount])

The result will be 390 because it calculates the average sales amount in the "SalesData" table.

(5) MIN: Find the minimum value in a column.

Minimum Sales Amount = MIN(SalesData[Sales Amount])

The result will be 150 because it finds the minimum sales amount.

(6) MAX: Find the maximum value in a column.

Maximum Sales Amount = MAX(SalesData[Sales Amount])

The result will be 600 because it finds the maximum sales amount.

(7) SUMMARIZE: Create a summary table based on specified columns. SummaryTable = SUMMARIZE(SalesData, SalesData[Product Category], "Total Sales", SUM(SalesData[Sales Amount]))

This formula generates a summary table with product categories and their total sales.

(8) CALCULATE: Modify the filter context of a calculation.

For example, calculate total sales for the "Electronics" category.

Total Sales Electronics = CALCULATE([Total Sales Amount], SalesData[Product Category] = "Electronics")

This formula calculates the total sales amount for the "Electronics" category.

(9) IF: Perform conditional logic.

Calculate a bonus if sales amount is greater than 400.

Bonus = IF([Total Sales Amount] > 400, [Total Sales Amount] * 0.1, 0)

This formula calculates a 10% bonus on sales if the total sales amount is greater than 400.

(10) IFERROR: Handle errors.

If an error occurs, return 0.

Safe Sales Amount = IFERROR([Total Sales Amount], 0)

This formula returns the total sales amount but substitutes errors with 0.

(11) ISBLANK: Check if a value is blank.

Return true if sales amount is blank.

Is Sales Amount Blank = ISBLANK([Total Sales Amount])

This formula returns TRUE if the total sales amount is blank.

(12) **EOMONTH**: Calculate the last day of the month for a given date.

Last Day of the Month = EOMONTH(MAX(SalesData[Date]), 0)

This formula finds the last day of the month for the latest transaction date in the dataset.

(13) **DATEDIFF**: Calculate the number of days between two dates.

Days Since First Transaction = DATEDIFF(MIN(SalesData[Date]), TODAY(), DAY)

This formula calculates the number of days since the first transaction date up to today.

11. DESCRIBE THE TERMINOLOGY USED IN POWER BI WITH AN EXAMPLE/USE CASE:

1) **Table**: In Power BI, a table represents a dataset or a collection of related data. For example, a "Sales" table may contain information about sales transactions.
2) **Fact**: A fact table in Power BI typically contains numerical data or measures that you want to analyze. For example, a "Sales" fact table may include sales revenue and quantity sold.
3) **Dimension**: A dimension table in Power BI provides descriptive context to the data in the fact table. For example, a "Product" dimension table may include product names, categories, and descriptions.
4) **Calendar**: A calendar table is a dimension table that contains date-related attributes. It's often used to enable time-based analysis, such as year-over-year comparisons.
5) **Relationship**: Power BI allows you to define relationships between tables to establish connections between them. For example, you can create a relationship between the "Sales" fact table and the "Product" dimension table using the "Product ID" column.
6) One to Many, One to One, Many to Many

   One to Many (1:N):

   Description: In a one-to-many relationship, one record in the primary (one) table relates to multiple records in the related (many) table.

   Use Case: An example is the relationship between a "Customers" dimension table and a "Sales" fact table. Each customer (one) can have multiple sales transactions (many). This relationship enables you to analyze sales data per customer.

One to One (1:1):

Description: In a one-to-one relationship, one record in the primary table relates to one and only one record in the related table.

Use Case: Suppose you have a "Company Info" dimension table containing company details. You may have another "Company Financials" dimension table with financial data, and each company's financial data matches one company from the "Company Info" table.

Many to Many (N:N):

Description: In a many-to-many relationship, multiple records in the primary table relate to multiple records in the related table, and vice versa.

Use Case: A classic example is a database with "Students" and "Courses." Many students can enroll in multiple courses, and each course has multiple students. To establish a many-to-many relationship, a junction table or bridge table is often used to link the two tables.

7) **Keys - Primary & Foreign Keys:** Primary keys are unique identifiers in a table. For example, the "Product ID" column in the "Product" dimension table is a primary key. Foreign keys are references to primary keys in other tables, allowing for relationships.
8) **Star Schema:** A star schema is a data model in which a central fact table is connected to dimension tables. For instance, a "Sales" fact table at the center is linked to dimension tables like "Product," "Customer," and "Time."
9) **Snowflake Schema:** A snowflake schema is a variation of the star schema where dimension tables are normalized into sub-dimensions. For example, a "Location" dimension could be further broken down into "City" and "Country" sub-dimensions.
10) **Measures**: Measures in Power BI are calculations applied to the fact data. An example is a "Total Sales" measure calculated by summing the sales revenue from the "Sales" table.
11) **Values, Aggregation**: Values represent individual data points. Aggregation involves calculating totals, averages, or other summary statistics on these values. For instance, aggregating "Sales" data to find the total revenue for a specific product category.

12) **Data Modelling**: Data modeling in Power BI involves structuring your data tables, defining relationships, and creating calculated measures to facilitate analysis and reporting.

13) **Slicer**: A slicer is a visual filter that allows users to interactively filter data. For example, a slicer can be used to filter sales data by a specific date range.

14) **Filter**: Filters in Power BI limit the data displayed in visualizations. For instance, you can create a filter to display data for a specific region or product category.

15) **Query**: A query in Power BI is a request for data retrieval or transformation. For instance, you can create a query to filter and transform data from a data source.

16) **ETL (Extract, Transform, Load):** ETL processes in Power BI involve extracting data from source systems, transforming it (e.g., cleaning, reshaping), and loading it into Power BI for analysis.

17) **Transformations**: Transformations are steps applied to data during the ETL process. For example, transforming text data into lowercase or removing duplicates.

18) **Batch**: Batch processing in Power BI involves processing data in chunks or batches rather than real-time updates.

19) **Data Pipeline:** A data pipeline is a series of data processing steps that move and transform data from source to destination within Power BI.

20) **Source**: A source refers to the origin of data, such as a database, Excel file, or web service, from which data is imported into Power BI.

21) **Refresh**: Refreshing data in Power BI is the process of updating the dataset with new or updated data from the source.