***Basics of Data Engineering***

***What is Data Engineering***

Data engineering is the process of planning, developing, and maintaining the infrastructure and tools needed to manage vast and complicated datasets. It involves duties such as data gathering, storage, processing, transformation, and analysis. To make sure that data is accurate, dependable, and simple to access for usage by data scientists, analysts, and other stakeholders, data engineers employ a number of tools and technology.

Data engineering's primary objective is to enable businesses to make data-driven decisions by offering dependable, effective, and scalable systems for gathering, storing, and processing data. This necessitates technological know-how in a variety of fields, including as database architecture, distributed systems, data modelling, and programming. A trained workforce is required to develop and construct the data infrastructure that supports modern enterprises as the demand for data engineering capabilities rises.

Here are some basic terms that everyone must know when entering this field.

***Big Data***

Big data refers to extremely large and complex data sets that cannot be easily processed, stored, or analyzed using traditional data management tools or methods. Big data typically refers to data sets that are too large to be handled by a single machine and require distributed computing frameworks and tools.

Big data is characterized by three key characteristics: **volume, velocity, and variety.**

***Volume*** refers to the large amount of data generated and collected from various sources, such as social media, sensors, and mobile devices. These data sets can range from terabytes to petabytes or even exabytes in size, making it difficult to store and process.

***Velocity*** refers to the high speed at which data is generated and must be processed in real-time or near real-time. This is particularly true for data streams, such as social media feeds, stock market data, or sensor data, which can generate huge amounts of data in a very short time.

***Variety*** refers to the different types of data that make up a big data set. These data types can be structured, semi-structured, or unstructured, and can come from a variety of sources, including text, images, video, and audio. This adds complexity to the processing and analysis of big data, as different tools and techniques may be required to handle each data type.

***Storing and processing big data*** requires specialized tools and technologies that can handle the large volume, high velocity, and diverse variety of data. Here are some of the common methods and technologies used to store and process big data:

Distributed file systems: These are designed to store and manage data across multiple nodes in a cluster. Examples include the Hadoop Distributed File System (HDFS) and Apache Cassandra.

Cloud computing platforms: Cloud-based storage and processing platforms like Amazon Web Services (AWS) and Microsoft Azure provide scalable and cost-effective solutions for storing and processing big data.

***Data Lake***

A data lake is a centralized repository that stores raw, unstructured, semi-structured, and structured data at scale.

Data lakes are designed to handle vast amounts of data, ranging from terabytes to petabytes, and are intended to support a wide range of use cases, such as data exploration, machine learning, and analytics.

In data engineering, data lakes are typically built using distributed file systems, such as Hadoop Distributed File System (HDFS), and are accompanied by a suite of tools for data ingestion, processing, and analysis, such as Apache Spark.

Data lakes differ from traditional data warehouses in that they are not pre-defined and rigidly structured, but rather flexible and adaptable to changing data requirements. This makes them an ideal solution for organizations that deal with large volumes of diverse and rapidly changing data.

***Database***

A database is a structured collection of data that is organized in a way that allows efficient storage, retrieval, and manipulation of information. Databases are used to store information for a wide range of applications, including business, scientific research, education, and more.

In a database, data is organized into tables, which consist of rows and columns. Each row in a table represents a single record or instance of an entity, while each column represents a specific attribute or characteristic of that entity. For example, in a database for an e-commerce website, a table might store customer information, with columns for each customer's name, address, email, and order history.

***Data Warehouse***

A data warehouse is a large-scale, centralized repository of data that is specifically designed to support business intelligence (BI) activities, such as data analysis, reporting, and decision-making.

Data warehouses are typically constructed from data that has been extracted, transformed, and loaded (ETL) from a variety of disparate sources, such as transactional databases, web services, and other data sources. The data is then transformed and organized into a common data model that is optimized for querying and analysis.

Overall, data warehouses are an essential tool for businesses and organizations that need to make data-driven decisions based on large volumes of complex data.

***Data Mart***

A smaller portion of a bigger data warehouse called a "data mart" is created to cater to the demands of a particular department or group inside an organization.

Data marts are often focused on a single business function, such as sales, marketing, finance, or human resources, and contain a portion of the data that is housed in the organization's data warehouse. The organization and optimization of the data in a data mart are tailored to the unique requirements of the department or group that uses it, making it simpler and more effective to access and analyze.

Data marts can be configured to be either autonomous or dependent on the bigger data warehouse. Independent data marts are freestanding entities that are built and maintained separately from the data warehouse, while dependent data marts are created from subsets of data that are extracted from the larger data warehouse.

***Data Lakehouse***

A data lakehouse is a modern data management architecture that combines the benefits of data lakes with data warehouses. It is a unified platform that allows users to store and analyze structured, semi-structured, and unstructured data in a single spot.

In a data lakehouse, data is absorbed into a data lake, which is a huge, centralized repository of raw, unstructured data. This data can then be changed and arranged using a combination of schema-on-read and schema-on-write techniques to make it more accessible and usable for analysis. The data warehouse provides users with fast and efficient access to the data, making it easier to run complex queries and generate insights.

***Data Mesh***

An architectural paradigm called "data mesh" aims to address the problems associated with scaling data management in big businesses.

In typical centralized data architectures, data is managed and controlled by a centralized team, which can cause bottlenecks and impede down the supply of data-driven insights. Since each domain or business unit is in charge of controlling its own data goods and services, data mesh advocates for decentralized data ownership.

***Extract, Transform, and Load (ETL) process***

A data pipeline called extract, transform, and load (ETL) is used to gather data from various sources. The data is subsequently transformed in accordance with business requirements before being loaded into a destination data repository. ETL's transformation process is carried out in a dedicated engine, and staging tables are frequently used to store data while it is being transformed and then loaded to its final destination.

The numerous procedures that are used to convert data often include filtering, sorting, aggregating, joining, cleaning, deduplicating, and validating data.

The three ETL processes are frequently carried out in simultaneously to save time. Instead of waiting for the entire extraction process to finish, for instance, a loading process could start working on the prepared data while the data is still being extracted by performing transformation on the data that has already been received.

Here's a breakdown of each step in the ETL process:

***Extract:*** The first step is to extract data from various sources, such as databases, flat files, APIs, or web services. The goal is to gather all the relevant data needed for analysis.

***Transform:*** Once the data is extracted, it needs to be transformed into a format that can be used for analysis. This includes cleaning the data, removing duplicates, filling in missing values, and converting data types. In addition, the data may need to be enriched with additional information, such as customer demographics or product categories.

***Load:*** The final step is to load the transformed data into a data warehouse or data mart. This involves creating tables and columns to store the data and ensuring that the data is loaded correctly and consistently. Once the data is loaded, it can be used for reporting, analysis, and visualization.

***Extract, Load, and Transform (ELT) process***

ELT (Extract, Load, Transform) is a variation of the ETL (Extract, Transform, Load) process used in data integration and business intelligence systems. While both ETL and ELT involve extracting data from various sources and loading it into a central repository, they differ in the way data is transformed.

In the ELT process, the data is first extracted and loaded into a target system, such as a data warehouse or data lake, without any transformation. Once the data is loaded, it can be transformed using tools within the target system, such as SQL queries or data pipelines.

Here's a breakdown of each step in the ELT process:

***Extract:*** The first step is to extract data from various sources, such as databases, flat files, APIs, or web services. The goal is to gather all the relevant data needed for analysis.

***Load:*** Once the data is extracted, it is loaded into a target system, such as a data warehouse or data lake. This involves creating tables and columns to store the data and ensuring that the data is loaded correctly and consistently.

***Transform:*** The final step is to transform the data within the target system. This can include cleaning the data, removing duplicates, filling in missing values, and converting data types. In addition, the data can be enriched with additional information, such as customer demographics or product categories, using SQL queries or data pipelines.

***3 Tier Architecture in DE***

In the context of data engineering, the 3-tier architecture is a common design pattern for building scalable and maintainable data processing systems.

The 3-tier architecture typically consists of the following layers:

***Presentation layer:*** This layer is responsible for presenting the data to the end-user. It can include various user interfaces such as web applications, mobile apps, or desktop applications.

***Application layer:*** This layer contains the business logic of the system. It is responsible for processing the data, applying transformations, and performing various computations. The application layer is typically implemented using programming languages such as Python, Java, or Scala.

***Data layer:*** This layer is responsible for storing and managing the data. It can include various data storage technologies such as relational databases, SQL databases, data lakes, and data warehouses.