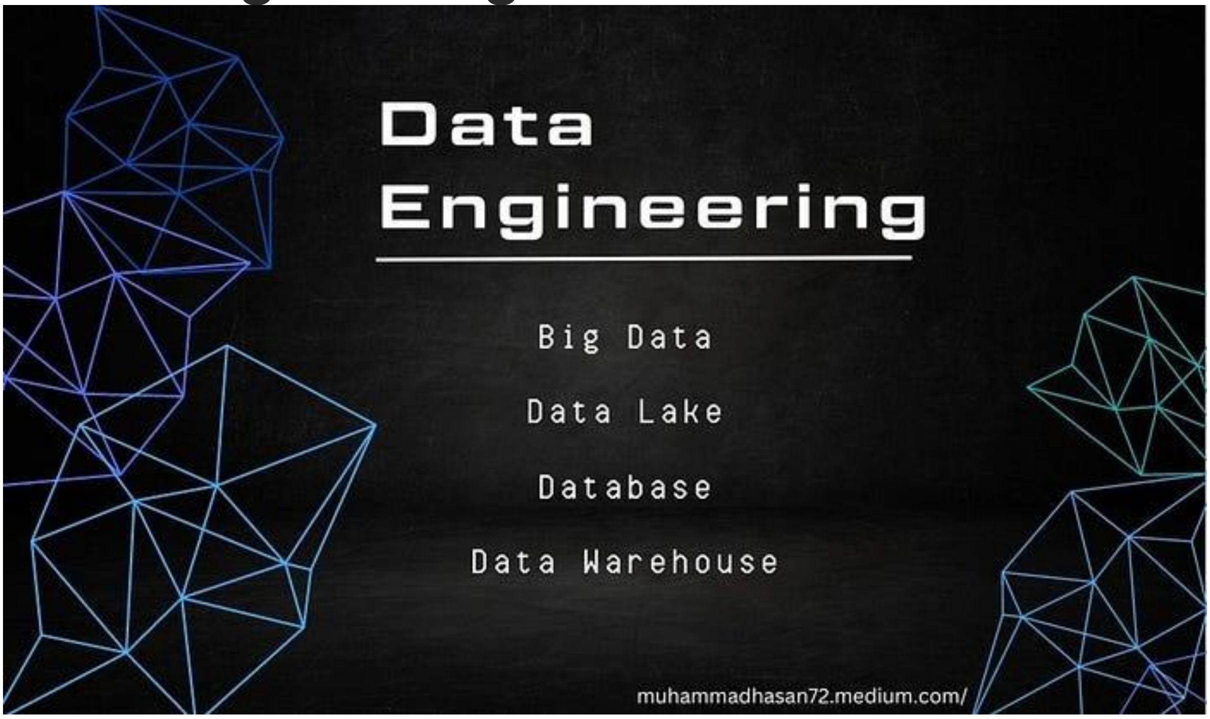
Data Engineering



Data Engineering

Data Engineering

Data engineering is the process of designing, building, and maintaining the infrastructure and systems that enable organizations to collect, store, process, and analyze large volumes of data. It involves the use of technologies such as Hadoop, Spark, and NoSQL databases to develop reliable, scalable, and efficient data pipelines. Data engineers work with programming languages such as Python, Java, and SQL to manipulate and transform data. Data engineering plays a crucial role in modern data-driven organizations, allowing them to make informed decisions based on accurate and up-to-date data.

Big Data

Big Data refers to extremely large and complex data sets that cannot be easily managed, processed, or analyzed using traditional data processing techniques. Big Data includes a variety of data types, such as structured, unstructured, and semi-structured data, and is typically characterized by its volume, velocity, and variety.

Social media data, IoT sensor data, and scientific research data are examples of Big Data. These data sets are typically large, and complex, and require specialized tools and techniques for analysis.

Data Lake

A data lake is a large centralized repository that stores all types of raw, unprocessed data in its native format. Unlike a data warehouse, a data lake is designed to store vast amounts of data, including both structured and unstructured data, and allows for flexible and on-demand processing of data. A data lake can be used to support advanced analytics, machine learning, and other data-intensive applications.

A company may create a data lake to store all its customer interactions across various channels such as emails, chatbots, social media, and phone calls. The data could include raw text, images, audio, and other types of unstructured data. This data could be used to develop more personalized marketing campaigns, improve customer service, or identify potential product improvements.

Database

A database is a structured collection of data that is organized and stored in a specific way to enable efficient storage, retrieval, and manipulation of data. A database typically consists of one or more tables, each containing rows and columns of data, and is managed using a database management system (DBMS).

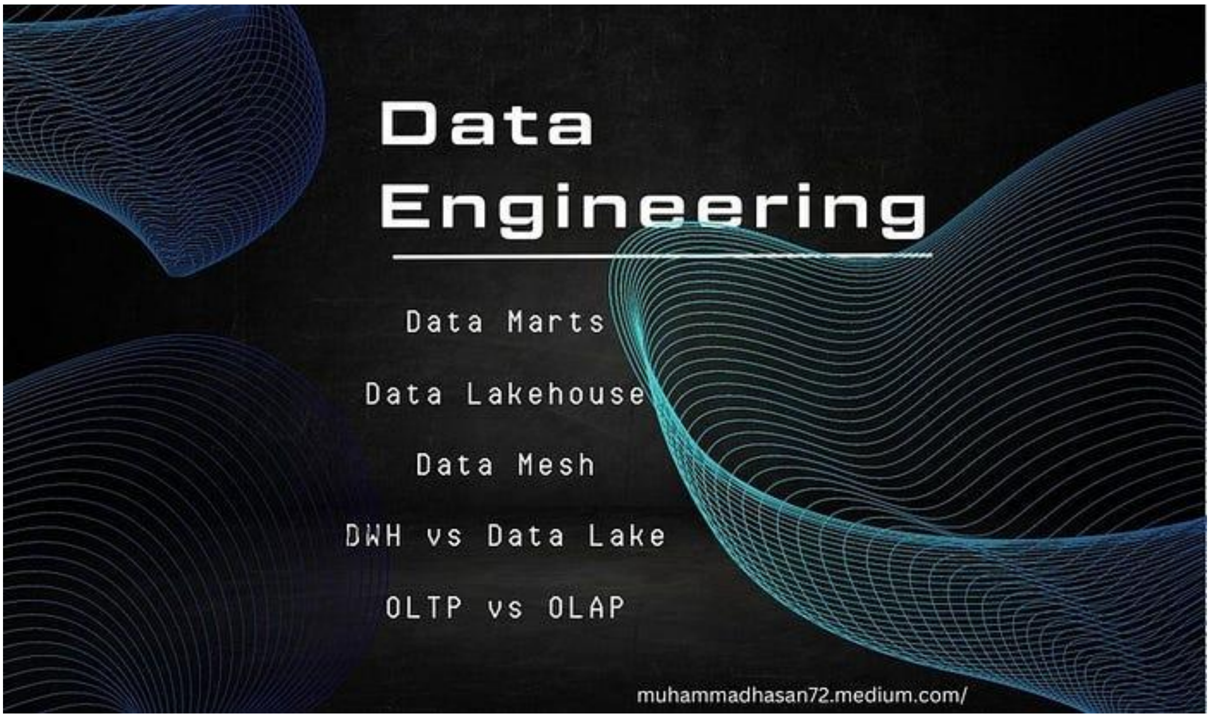
A company may use a database to store its customer data, such as names, addresses, and purchase history. The database would likely include one or more tables, with each row representing a single customer and columns containing specific customer attributes.

Data Warehouse

A data warehouse is a centralized repository that stores historical data from multiple sources in a structured way to support business intelligence, reporting, and data analysis. A data warehouse is typically optimized for fast query performance and is designed to facilitate complex analytical queries on large volumes of data. Data warehouses often use Extract, Transform, Load (ETL) processes to extract data from source systems, transform it into a consistent format, and load it into the warehouse.

A retailer may create a data warehouse to store historical sales data from various sources, such as point-of-sale systems, online orders, and customer loyalty programs. The data warehouse would be optimized for fast querying and would allow analysts to gain insights into sales trends, customer behavior, and inventory management.

ETL processes could be used to transform and integrate the data from these sources into a consistent format.



Data Engineering

Data Marts

A data mart is a subset of a data warehouse that is designed to serve a specific business unit, department, or function within an organization. Unlike a data warehouse, a data mart contains a smaller subset of data that is tailored to the needs of specific users. Data marts can be created by extracting data from a data warehouse or by integrating data from multiple sources.

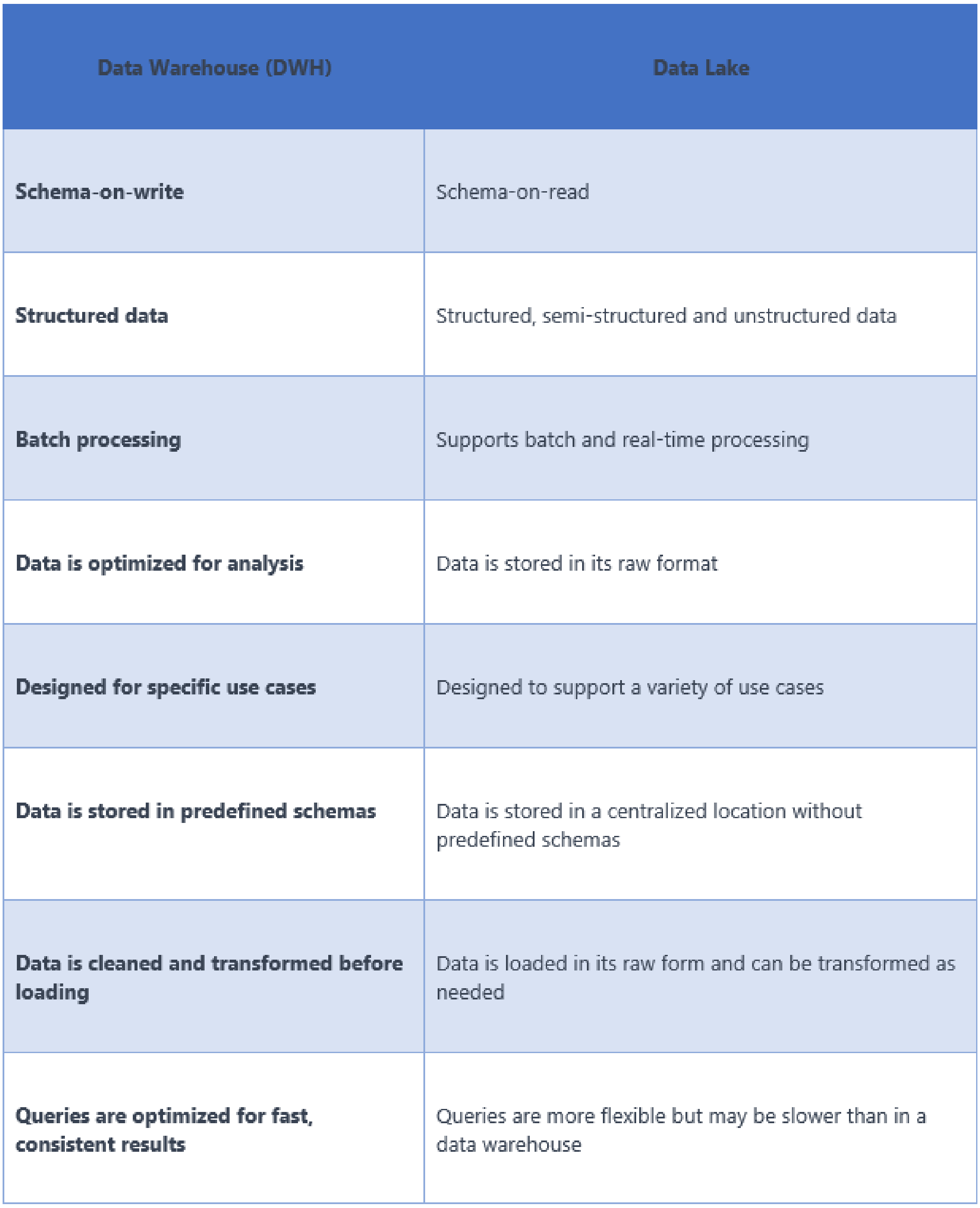
Data Lakehouse

A data lakehouse is a hybrid architecture that combines the flexibility and scalability of a data lake with the performance and reliability of a data warehouse. A data lakehouse architecture allows organizations to store raw data in a centralized location and use it for a variety of analytics use cases. The data is structured using schema-on-read, which means that the schema is applied to the data only when it is read or queried, allowing for more flexible and agile data analysis.

Data Mesh

Data mesh is a new approach to organizing and managing data within an organization. It emphasizes the creation of decentralized, domain-specific data teams that are responsible for managing the data for their specific domains. Data mesh advocates for a self-serve data platform that provides data teams with the tools and services they need to manage their data independently. This approach aims to address the challenges of traditional centralized data management, such as data silos and bottlenecks, and to promote a more agile and collaborative approach to data management.

Data Warehouse vs Data Lake

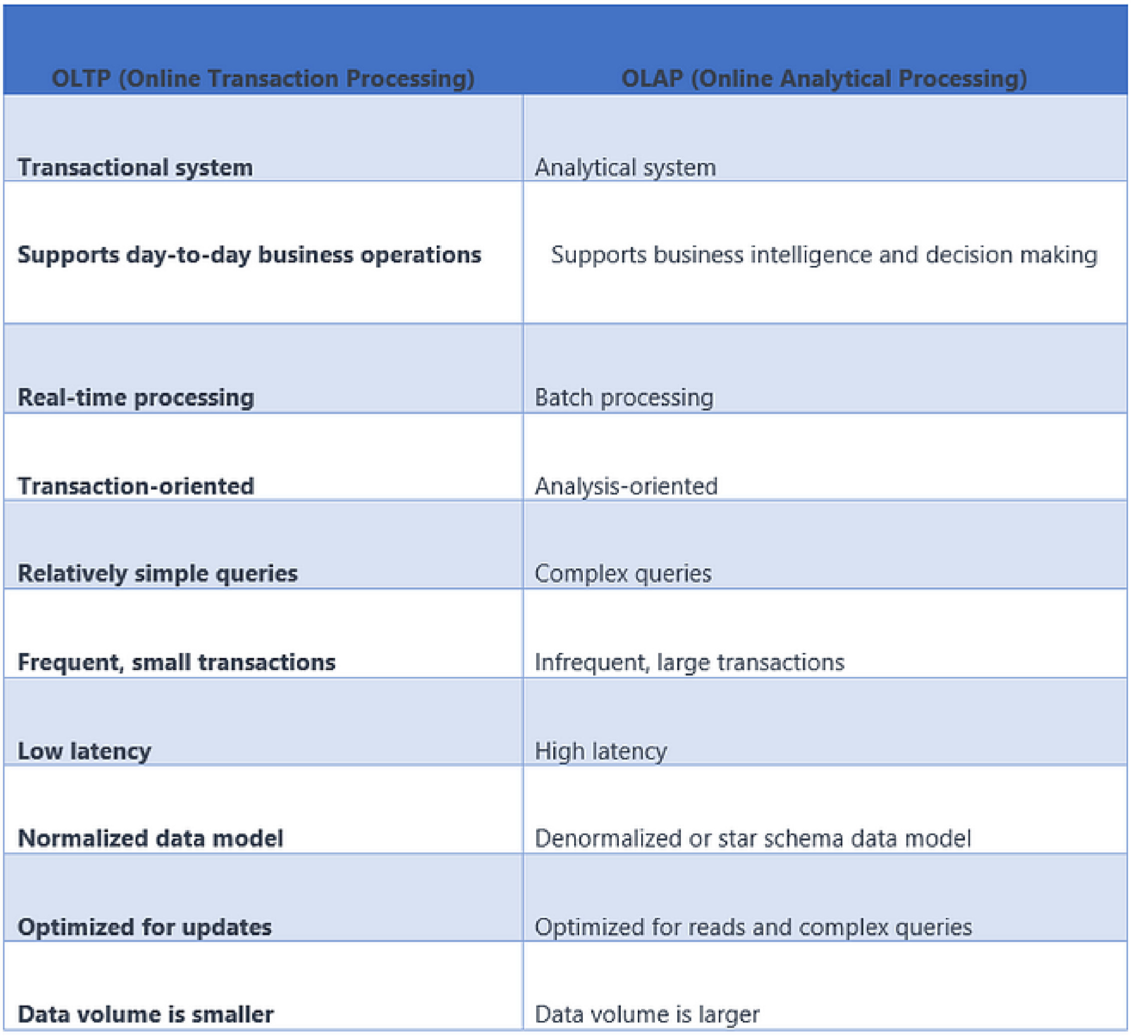


Data Warehouse (DWH)	Data Lake
Schema-on-write	Schema-on-read
Structured data	Structured, semi-structured and unstructured data
Batch processing	Supports batch and real-time processing
Data is optimized for analysis	Data is stored in its raw format
Designed for specific use cases	Designed to support a variety of use cases
Data is stored in predefined schemas	Data is stored in a centralized location without predefined schemas
Data is cleaned and transformed before loading	Data is loaded in its raw form and can be transformed as needed
Queries are optimized for fast, consistent results	Queries are more flexible but may be slower than in a data warehouse

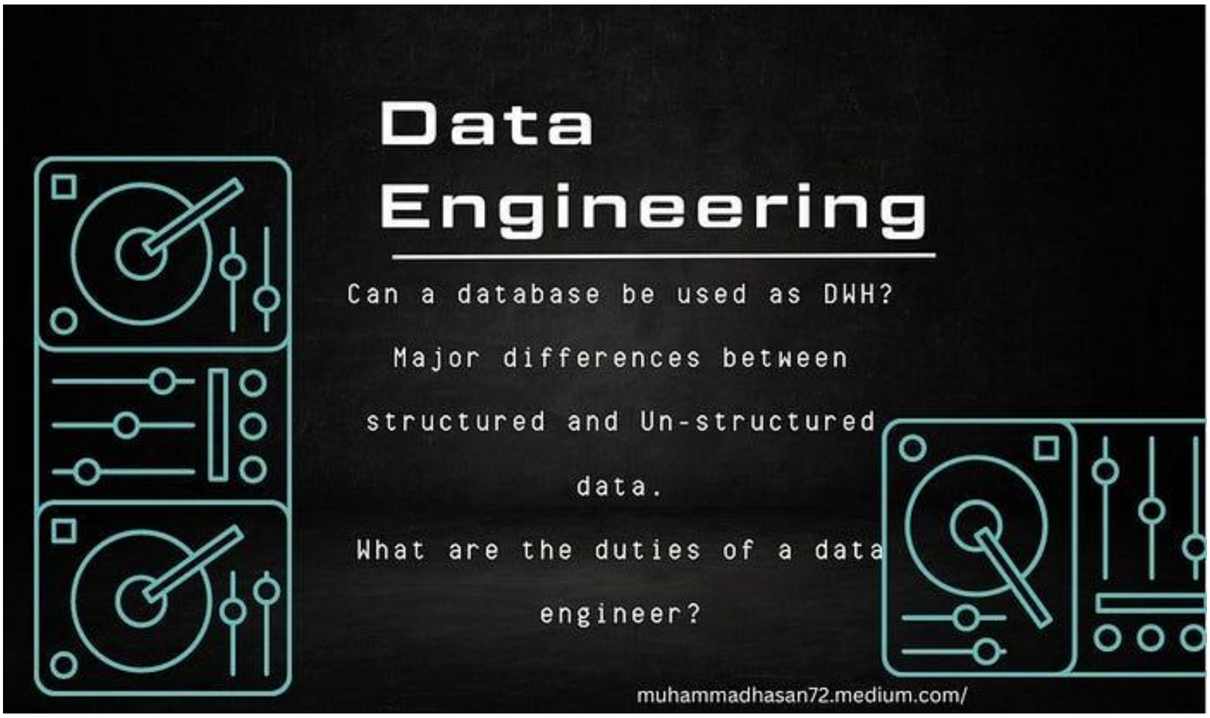
OLTP vs OLAP

OLTP (Online Transaction Processing) is a type of data processing that is used to manage day-to-day business operations. OLTP systems are typically transaction-oriented and handle small, frequent transactions in real time. These systems are designed to support high levels of concurrency and low-latency processing. Examples of OLTP systems include point-of-sale systems, online banking systems, and airline reservation systems.

OLAP (Online Analytical Processing) is a type of data processing that is used to support business intelligence and decision-making. OLAP systems are typically analysis-oriented and handle large, infrequent transactions in batch-processing mode. These systems are designed to support complex queries and provide fast, interactive access to large volumes of data. OLAP systems often use denormalized or star schema data models to optimize for reads and complex queries. Examples of OLAP systems include data warehouses, data marts, and business intelligence tools.



OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
Transactional system	Analytical system
Supports day-to-day business operations	Supports business intelligence and decision making
Real-time processing	Batch processing
Transaction-oriented	Analysis-oriented
Relatively simple queries	Complex queries
Frequent, small transactions	Infrequent, large transactions
Low latency	High latency
Normalized data model	Denormalized or star schema data model
Optimized for updates	Optimized for reads and complex queries
Data volume is smaller	Data volume is larger


Data
Engineering
Can a database be used as DWH?
Major differences between
structured and Un-structured
data.
What are the duties of a data
engineer?
muhammadhasan72.medium.com/

Data Engineering

Can a database be used as DWH?

Yes, a database can be used as a data warehouse, especially for smaller-scale data warehousing needs. Some databases are designed specifically for data warehousings, such as Amazon Redshift, Google BigQuery, and Microsoft Azure SQL Data Warehouse. These databases offer features such as columnar storage, distributed processing, and advanced analytics capabilities, which make them well-suited for data warehousing use cases.

However, as data volumes and complexity increase, a separate data warehouse may be necessary to ensure optimal performance, scalability, and flexibility. A data warehouse typically offers additional features and functionality that are not available in a traditional database, such as support for complex ETL processes, data modeling, data quality, and data governance. Additionally, a data warehouse can be optimized specifically for analytics and reporting, which can result in faster query performance and more accurate insight.

Major differences between structured and Unstructured data.

The major differences between structured and unstructured data are:

Structure: Structured data is highly organized and formatted in a specific way, whereas unstructured data has no predefined structure or format.

Storage: Structured data is typically stored in a database or spreadsheet, while unstructured data can be stored in various formats such as text files, images, videos, and audio files.

Analysis: Structured data can be easily analyzed using traditional analytical methods, such as SQL queries and statistical models. Unstructured data, on the other hand, requires more advanced techniques such as natural language processing (NLP), machine learning, and deep learning to extract insights.

Volume: Structured data is usually smaller in volume than unstructured data. Unstructured data can come in large volumes and can be more difficult to manage and analyze.

Accuracy: Structured data is typically more accurate than unstructured data, as it is often subject to strict data entry requirements and validation. Unstructured data, on the other hand, can be subject to errors and inconsistencies.

Use cases: Structured data is well-suited for use cases such as transactional processing, business reporting, and data analysis. Unstructured data is better suited for use cases such as sentiment analysis, image and video recognition, and text analytics.

What are the duties of a data engineer?

The duties of a data engineer can vary depending on the organization and the specific needs of the data infrastructure, but generally include the following:

Design and build data pipelines: Data engineers design, build, and maintain data pipelines that extract data from various sources, transform it into a usable format, and load it into a data warehouse or data lake. This involves writing ETL (extract, transform, load) scripts and working with various data integration tools and technologies.

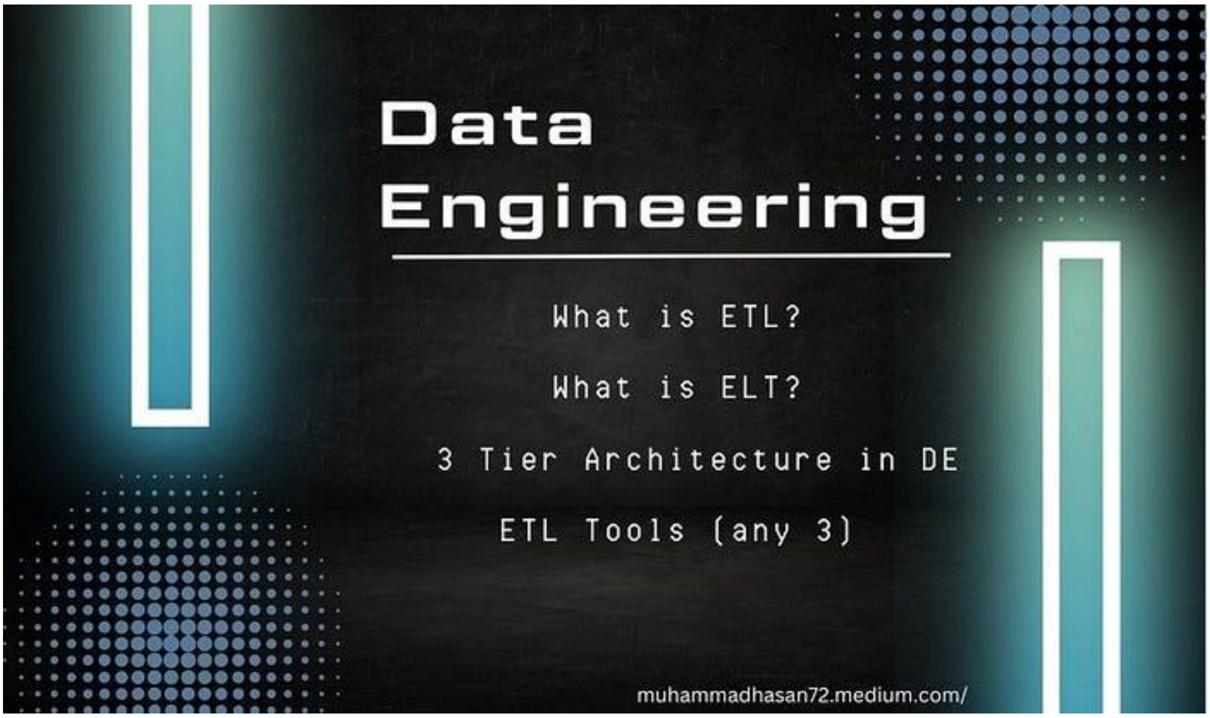
Maintain data infrastructure: Data engineers are responsible for ensuring that the data infrastructure is available, reliable, and scalable. This includes monitoring the performance of data systems, troubleshooting issues, and implementing solutions to improve performance and availability.

Data modeling and architecture: Data engineers work with data architects and data scientists to design and implement data models and data architectures that meet the needs of the organization. They must have a solid understanding of data modeling concepts and be able to translate business requirements into effective data models.

Data quality and governance: Data engineers are responsible for ensuring that data is accurate, complete, and consistent. They must develop and implement data quality standards and ensure compliance with data governance policies and regulations.

Collaborate with cross-functional teams: Data engineers work closely with data scientists, data analysts, software engineers, and other stakeholders to understand their needs and develop data solutions that meet their requirements.

Stay up-to-date with industry trends: Data engineers must stay up-to-date with the latest technologies and industry trends in data engineering, and continuously improve their skills and knowledge to keep pace with changing demands.



Data Engineering

What is ETL?

ETL stands for Extract, Transform, Load, which is a data integration process used in data warehousing and business intelligence.

The ETL process involves three main steps:

Extract: The first step is to extract data from various sources, such as databases, files, APIs, and web services. The data is typically extracted into a staging area or a temporary location, where it can be processed and cleaned before being loaded into the target data warehouse or data lake.

Transform: The extracted data is then transformed into a format that is suitable for analysis and reporting. This involves cleaning, filtering, aggregating, and enriching the data using various data processing techniques, such as sorting, joining, deduplicating, and formatting.

Load: The final step is to load the transformed data into the target data warehouse or data lake. This involves mapping the data to the target schema and performing a data load process, which can be an either incremental or full load.

The ETL process is critical for data warehousing and business intelligence, as it ensures that data is accurate, complete, and consistent across multiple sources. It also enables organizations to perform complex data analysis and reporting, which can help them make informed business decisions.

What is ELT?

LT stands for Extract, Load, Transform, which is a data integration process used in data warehousing and business intelligence.

The ELT process involves the following steps:

Extract: The first step is to extract data from various sources, such as databases, files, APIs, and web services. The data is then loaded directly into the target data warehouse or data lake without any transformation.

Load: Once the data is loaded into the target data warehouse or data lake, it is stored in its raw form without any processing. This allows organizations to store large volumes of data without incurring the cost of transforming the data upfront.

Transform: The final step is to transform the data as needed, such as cleaning, filtering, aggregating and enriching the data using various data processing techniques. This transformation can be performed on-demand as required, rather than upfront during the ETL process.

The ELT process differs from the traditional ETL process in that it flips the order of the “T” and “L” steps. Instead of transforming the data before loading it into the target data warehouse or data lake, ELT loads the raw data first and transforms it on demand as needed. This approach can be more efficient and cost-effective, as it reduces the need for upfront transformation and can provide more flexibility in the data processing. However, it also requires more powerful data processing capabilities in the target data warehouse or data lake to support the on-demand transformation

ETL vs ELT

ETL and ELT are two different approaches to data integration used in data warehousing and business intelligence.

ETL stands for Extract, Transform, Load, and involves extracting data from various sources, transforming it into a format that is suitable for analysis and reporting, and then loading it into the target data warehouse or data lake. ETL is a batch-oriented approach, meaning that it processes data in batches at regular intervals, such as nightly or weekly. ETL is suitable for situations where data needs to be transformed upfront before being loaded into the target system, and when the target system is not able to handle large volumes of data.

ELT stands for Extract, Load, Transform, and involves extracting data from various sources, loading it directly into the target data warehouse or data lake without any transformation, and then transforming the data on-demand as needed. ELT is a more real-time approach, meaning that data can be processed as soon as it is available. ELT is suitable for situations where the target system can handle large volumes of data and where there is a need for flexibility in data processing.

The main differences between ETL and ELT are:

Data transformation: ETL transforms data upfront before loading it into the target system, while ELT loads the raw data and transforms it on-demand as needed.

Data volume: ETL is suitable for handling smaller volumes of data, while ELT is better suited for larger volumes of data.

Processing speed: ETL is a batch-oriented approach that processes data at regular intervals, while ELT is a more real-time approach that can process data as soon as it is available.

Overall, the choice between ETL and ELT depends on the specific requirements of the data integration project, such as the data volume, processing speed, and target system capabilities.

3 Tier Architecture in DE

The three-tier architecture is a common architecture used in data engineering to design and deploy data systems. The three tiers are:

Presentation tier: The presentation tier, also known as the client tier or front-end tier, is the topmost layer in the architecture. It is responsible for presenting data to the user in a user-friendly format, such as a web interface or a dashboard. The presentation tier interacts with the application tier to retrieve and display the data.

Application tier: The application tier, also known as the middle tier or server tier, is the layer that sits between the presentation tier and the data storage tier. It contains the business logic and processing logic that determines how the data is retrieved, processed, and delivered to the presentation tier. The application tier interacts with both the presentation tier and the data storage tier.

Data storage tier: The data storage tier, also known as the back-end tier or database tier, is the layer that stores the data. It can consist of one or more databases, data warehouses, data lakes, or other storage systems. The data storage tier is responsible for managing the data, ensuring data integrity and security, and providing efficient data access.

The three-tier architecture is a modular and scalable architecture that allows data engineers to separate concerns and manage each tier independently. It also enables organizations to easily add or *remove components as needed to adapt to changing business needs.*

ETL Tools

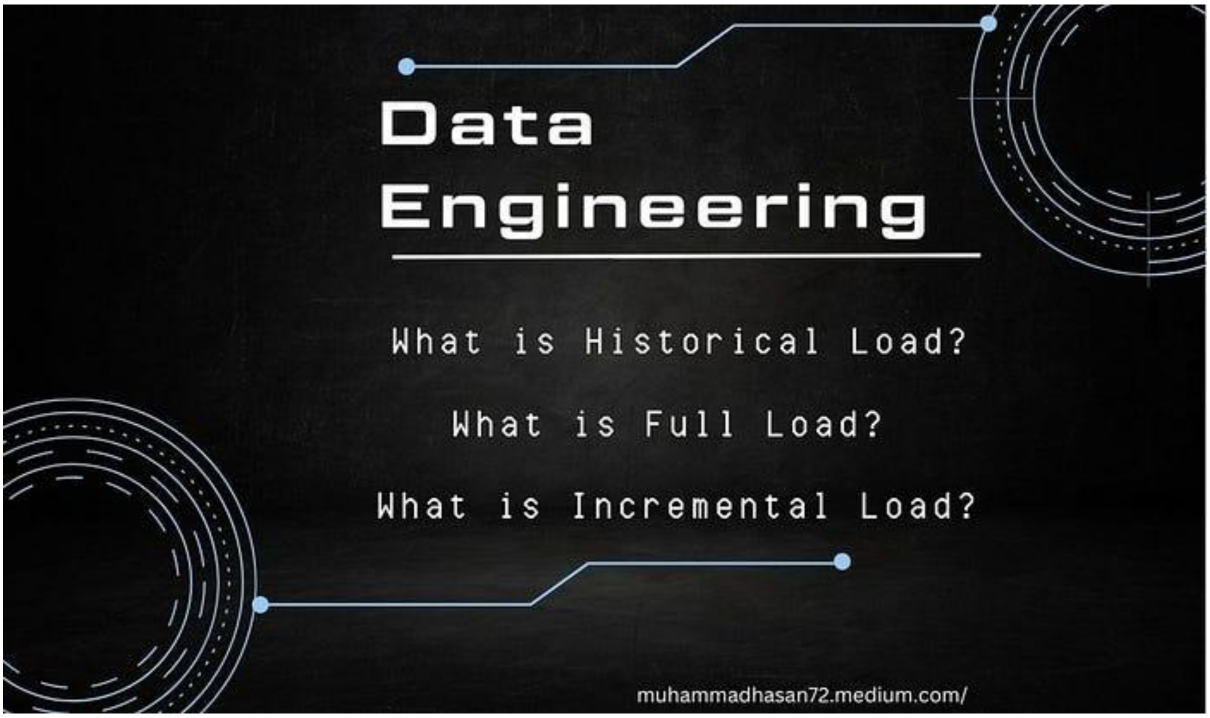
There are many ETL (Extract, Transform, Load) tools available in the market, but three popular ones are:

Apache NiFi: Apache NiFi is an open-source data integration tool that provides a web-based user interface to design, control, and monitor data flows. It supports a wide range of data sources and destinations, and its drag-and-drop interface makes it easy to create complex data integration pipelines.

Talend: Talend is a popular open-source ETL tool that provides a comprehensive set of data integration and data management tools. It supports a wide range of data sources and destinations, and its visual interface makes it easy to design and deploy data integration workflows.

Microsoft SSIS: Microsoft SQL Server Integration Services (SSIS) is a popular ETL tool that is included with Microsoft SQL Server. It provides a visual interface for building data integration workflows and supports a wide range of data sources and destinations.

These tools are widely used in data integration projects and have a large community of users and developers, making them reliable and well-supported options for ETL.



Data Engineering

What is a Historical Load?

Historical load refers to the process of loading historical data into a data warehouse or data lake. This involves extracting data from source systems, transforming it into a format that is suitable for analysis and reporting, and then loading it into the target system.

The purpose of the historical load is to provide a comprehensive view of historical data, allowing analysts and decision-makers to understand trends, patterns, and insights that can inform business decisions. Historical data can be used to analyze performance, track customer behavior, identify market trends, and more.

Historical load is typically a one-time process, where data from past periods is loaded into the data warehouse or data lake to create a complete historical record. This process can be time-

consuming and resource-intensive but is essential for organizations that need to analyze historical data to gain insights into their business operations.

What is a Full Load?

Full load refers to the process of loading all of the data from source systems into a target data warehouse or data lake. This involves extracting all of the data from source systems, transforming it into a format that is suitable for analysis and reporting, and then loading it into the target system.

A full load is typically performed during the initial implementation of a data warehouse or data lake, or when there are significant changes to the source data that require a complete refresh of the target system. A full load can be a time-consuming and resource-intensive process, but it is necessary to ensure that the target system contains all of the necessary data for accurate reporting and analysis.

After the initial full load, incremental loads can be used to update the target system with only the changes that have occurred since the last load, which is a more efficient way of keeping the target system up to date. However, periodic full loads may still be required to ensure that the target system remains in sync with the source systems and contains a complete and accurate set of data.

What is an Incremental Load?

Incremental load is the process of updating a target data warehouse or data lake with only the data that has changed since the last load. This approach allows for more efficient data integration and reduces the amount of time and resources required to keep the target system up to date.

Incremental load involves identifying the changes that have occurred in the source data since the last load, and then extracting, transforming, and loading only those changes into the target system. This can be accomplished using various techniques, such as comparing timestamps or tracking changes using flags or markers.

Incremental load is typically performed regularly, such as daily or hourly, to keep the target system up to date with the latest data changes. This approach allows organizations to keep up with changes in their business operations and to provide timely and accurate reporting and analysis.