Name: Abhishek Kanoujia

DATA ENGINEERING BATCH 1

DAY 1 ASSIGNMENT

Data Engineer's Role:

Data Engineers are responsible for designing, building, and scaling platforms that facilitate data collection, processing, and storage for applications in data science and business analytics.

Data Scientist's Role:

Data Scientists use mathematical concepts such as linear algebra and multivariable calculus to derive new insights from existing data.

Data Engineering Tasks:

Data Engineering involves designing, building, and scaling systems for organizing data for analytics, often accomplished through Extract, Transform, Load (ETL) processes.

ETL Basics:

Extract: Retrieve raw data from sources like JSON format.

Transform: Apply schema to raw data, producing processed data.

Load: Store processed data in event tables or pipeline destinations.

Data Classification:

Raw Data: Unprocessed data without a schema.

Processed Data: Raw data with a schema applied, stored in event tables.

Cooked Data: Summarized processed data.

Big Data Properties:

Volume: The amount of data.

Velocity: The speed at which data arrives.

Variety: The diversity of data types.

Veracity: The reliability of data.

Data Processing Methods:

Batch Processing: Handling data in scheduled, periodic batches.

Stream Processing: Processing data in real-time as it arrives.

Streaming Methods:

At Least Once: Ensuring data is processed at least once.

At Most Once: Processing data at most once to prevent duplication.

Exactly Once: Ensuring data is processed exactly once with no duplicates.

Processing Frameworks:

Map Reduce: Organizing data into key-value pairs, sorting by key, and combining data with matching keys.

Data Storage:

Relational Database (SQL): Structured storage with tables and relationships.

Document Store (NoSQL): Non-relational storage for flexible, schema-less data.

Conclusion:

Understanding the basics of data engineering involves grasping the roles of data engineers and scientists, ETL processes, data classification, big data properties, processing methods, and storage options. This knowledge forms the foundation for building scalable and efficient data systems.

Data Warehousing Definition:

Data Warehouse (DW): According to W.H. Inmon, a Data Warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data supporting management's system.

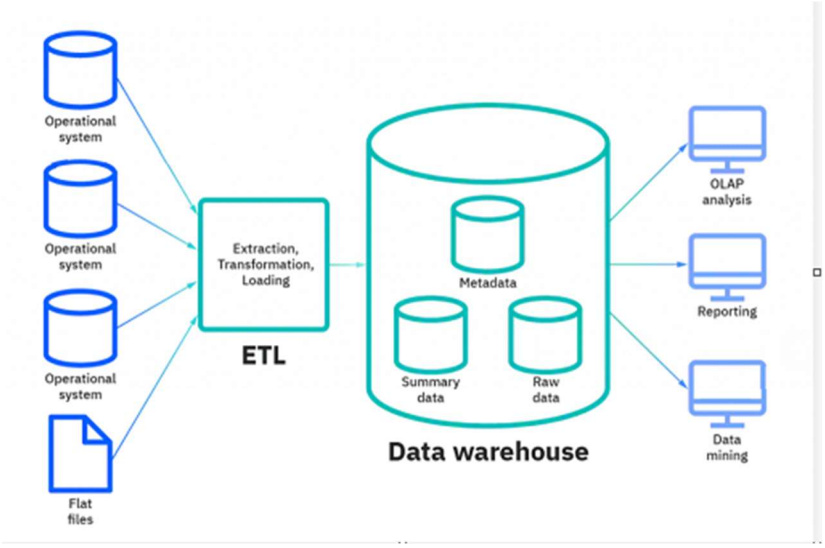
Data Warehousing Features:

Subject-Oriented: Organizes data based on subjects, focusing on analysis for decision-makers rather than daily operations.

Integrated: Integrates data from various sources, ensuring consistency in naming conventions and structures.

Time-Variant: Provides historical data over a longer time horizon, offering a historical perspective for analysis.

Non-Volatile: Once data is in the warehouse, no updates are allowed, preserving a company's historical records.



Need for Decision Support Systems (DSS):

In a competitive business environment, quick decision-making is crucial.

DSS assists in assessing and resolving everyday business questions by compiling information from various sources.

It supports both structured components, aiding direct decisions, and unstructured components, requiring human interaction.

Structured and Unstructured Components of DSS:

Structured Component: Involves quantifiable aspects like risk and performance.

Unstructured Component: Requires human intuition, such as the choice of stocks for a portfolio.

DSS Architectural Styles:

OLTP (Online Transaction Processing): Used by traditional operational systems for regular transaction processing.

OLAP (Online Analytical Processing): Used by Data Warehouses for analytical purposes.

OLTP:

OLTP databases are optimized for fast transaction processing, accessing data through operations like inserting, deleting, and updating.

Works intuitively for quick deductions based on investigative reasoning.

Benefits of OLTP:

Simplicity & Efficiency: Reduces paper trails and enables faster, more accurate revenue and expense forecasts.

Maintains Data Integrity: Provides fast query processing in environments with multiple access.

Pitfalls of OLTP:

Requires instant updates.

Data obtained is not suitable for in-depth data analysis.

Multiple table queries and joins are needed for even simple transactions with normalized structures.