**Data Engineering Fundamentals**

**What is data engineering?**

**Data Engineering** is the process of designing and building systems that allow organizations to collect, store, process, and analyse large amounts of data. Data engineers work to make sure data is accessible, clean, and ready to be analysed by data scientists, analysts, or other stakeholders in an organization.

In other words, converting raw data into information.

**Difference between OLTP and OLAP.**

|  |  |
| --- | --- |
| **OLTP** | **OLAP** |
| Online Transaction Processing | Online Analytical Processing |
| OLTP systems are designed to manage real-time transaction data. They handle large numbers of short, simple transactions and are optimized for quick write and read operations | OLAP systems are designed for analysing historical data. They provide insights by enabling complex queries and aggregations across large datasets |
| Faster for transactions | Slower but optimized for analysis |
| Highly normalized | Denormalized schemas |
| Current, operational data | Historical, aggregated data |
| Ex: Databases like MySQL | Ex: Data warehouses like RedShift |

**What is what is staging area?**

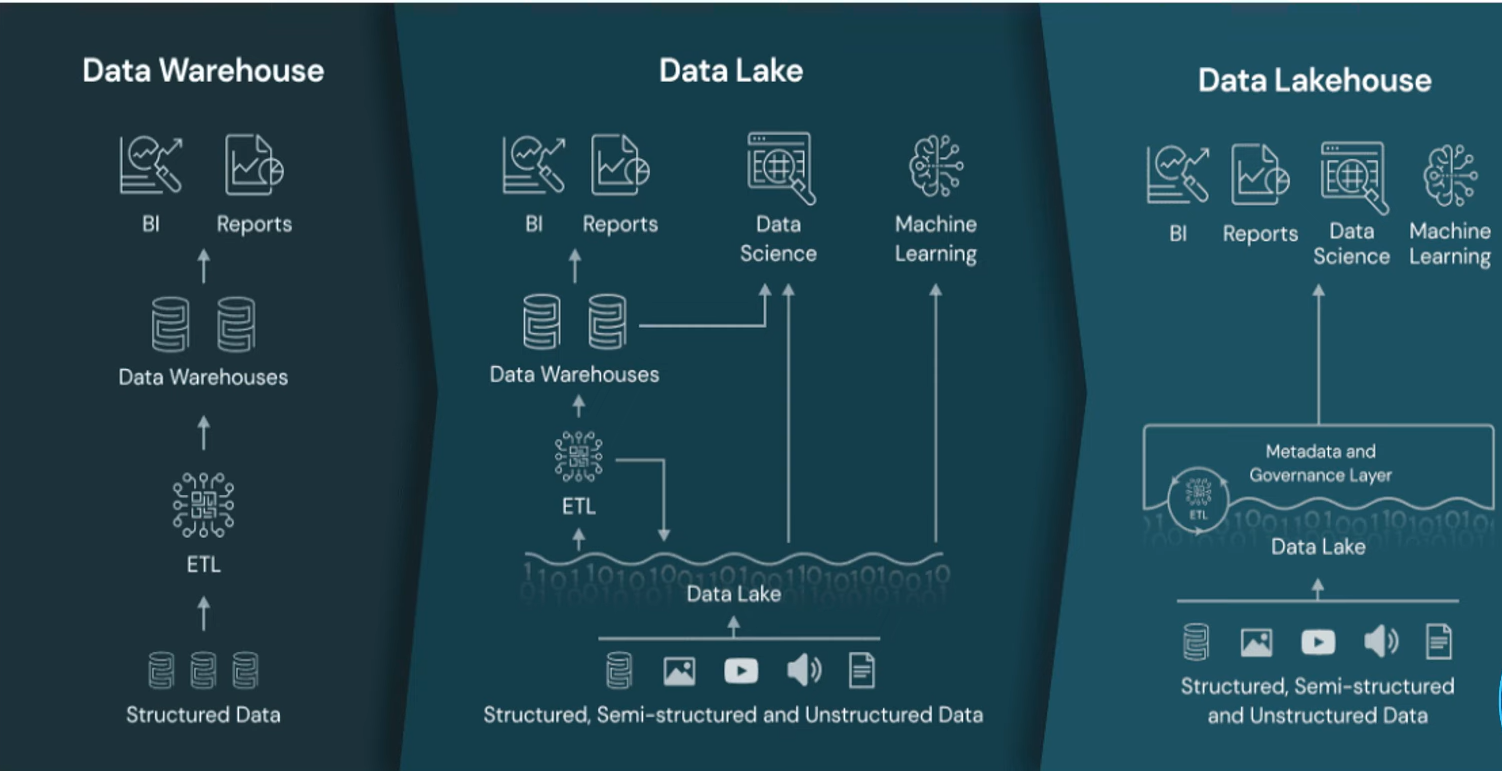
Staging refers to the process of temporarily storing and preparing data before it is moved into a final destination. The staging area is often used to handle raw or intermediate data from various sources before it's transformed, cleaned, and loaded into a structured data repository.

Example: Data lake, warehouse, database, s3 etc.

**What is incremental loading?**

Incremental Loading is a data integration technique used to efficiently load only the new or modified data into a target system (e.g., a data warehouse) instead of reloading the entire dataset.

The main motivation behind incremental loading is to improve performance and efficiency. This reduces the load on the database, optimizes processing time, and minimizes resource usage, which is especially important when dealing with large datasets.



**Difference between row based and column-based file formats.**

Row-based: [CSV, JSON]

In row-based file formats, the data is stored one row at a time. This means that all the data values for a single record (or row) are stored together in a sequence before moving to the next row.

Advantages of Row-Based Formats:

* Good for transactional systems: If you are accessing or modifying individual rows (e.g., in OLTP systems), row-based formats are ideal.
* Efficient for row-level operations: Since all the data for a row is stored together, accessing an entire record (like reading the "name" and "age" for a person) is quick.

Column-based: [Parquet, ORC]

Advantages of Column-Based Formats:

* Efficient for analytical queries: If you're querying specific columns (e.g., finding the average value of "age" across millions of records), columnar formats are much faster and more efficient.
* Better compression: Since the data in a column is often of the same type and structure, column-based formats can compress the data much more effectively, reducing storage requirements.
* Column pruning: When querying a dataset, you can choose to load only the columns you're interested in, which speeds up processing and reduces memory usage.