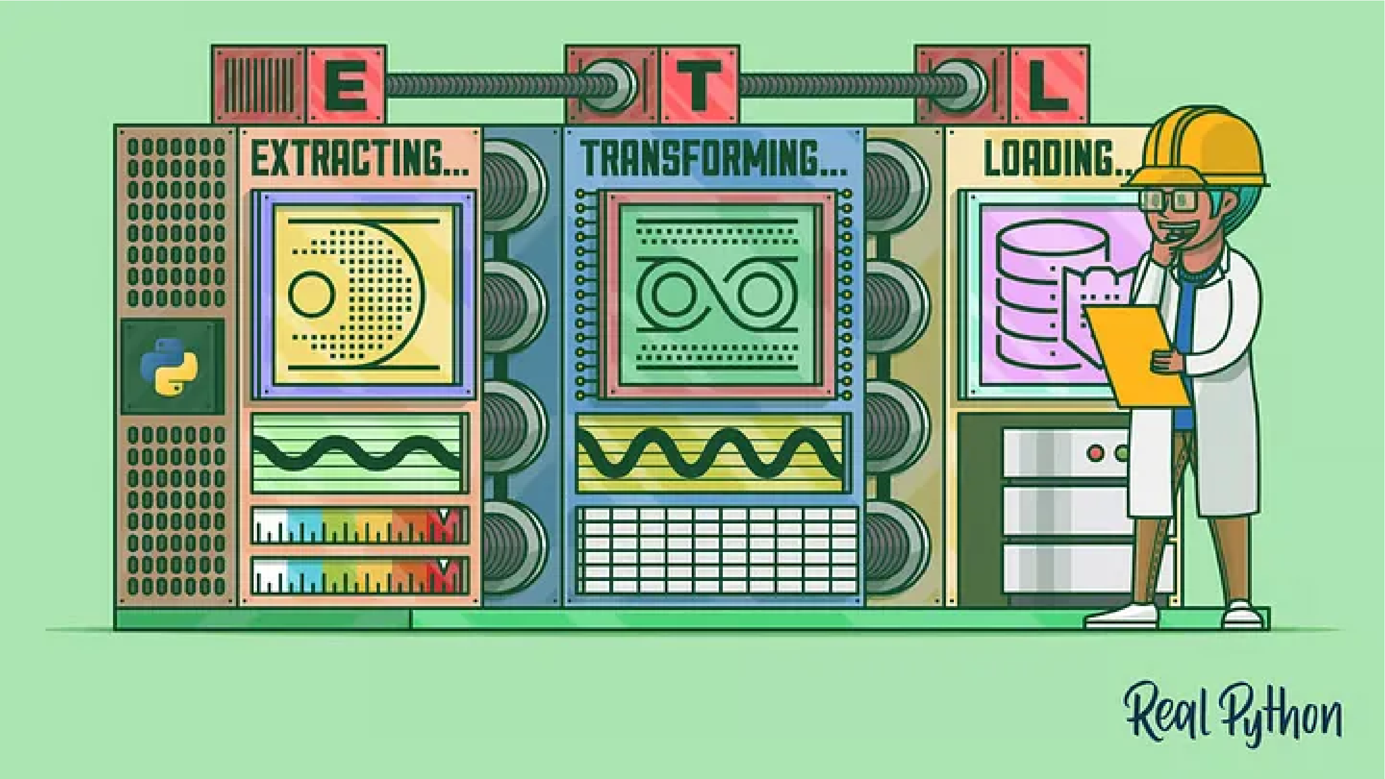
Muhammad Fahad
Mar 17 · 10 min read · Listen

Exploring the Fundamentals of Data Engineering

Data Engineering refers to the process of developing and creating the infrastructure to manage, store, process and analyze massive amounts of data. It encompasses various tasks including acquiring data, transforming data, preserving data, and retrieving data and requires special skills and knowledge in areas such as designing databases, integrating data, constructing data pipelines and developing data warehousing.

Data engineering helps organizations make sense of the vast amounts of data they collect by extracting valuable insights from it. This enables organizations to make informed decisions, optimize business processes, and grow. Industries like finance, healthcare and e-commerce that deal with large amounts of data rely on data engineering to stay competitive in today's data-driven world. By using data engineering techniques, these industries can stay ahead of the curve in a world that increasingly depends on data.

Responsibilities of Data Engineer



- **Design and Develop Data Architectures:** Data Engineers design and develop the architecture for data systems, ... nes, data warehouse and data lakes. They ensure that the ar... , reliable and optimized.

- **Developing Data Pipelines:** Data engineers build and maintain data pipelines that transport data from source to target systems. This includes creating connectors to different data sources performing data transformation and ensuring data quality and consistency.
- **Ensure Data Quality**: Data engineers are responsible for ensuring that data is consistent, accurate and trustworthy. They implement data validation and quality checks. They work with data scientists and analyst to resolve any issues arise.
- **Implementing Data Governance Policies**: Data engineers implement data governance policies to ensure that data is managed in compliance with legal and regulatory requirements and ensure the security of sensitive data by implementing access controls, encryption, and other security measures
- **Data Integration**: Collect, clean and transform data from different sources so that it can be used for analysis. This may involve designing and implementing ETL (Extract, Transform, Load) processes that move data from one system to another.
- **Performance Optimization**: Optimize data processing and query performance by tuning databases, data pipelines and other systems. This includes monitoring system performance and implementing optimizations to improve system efficiency.

— —

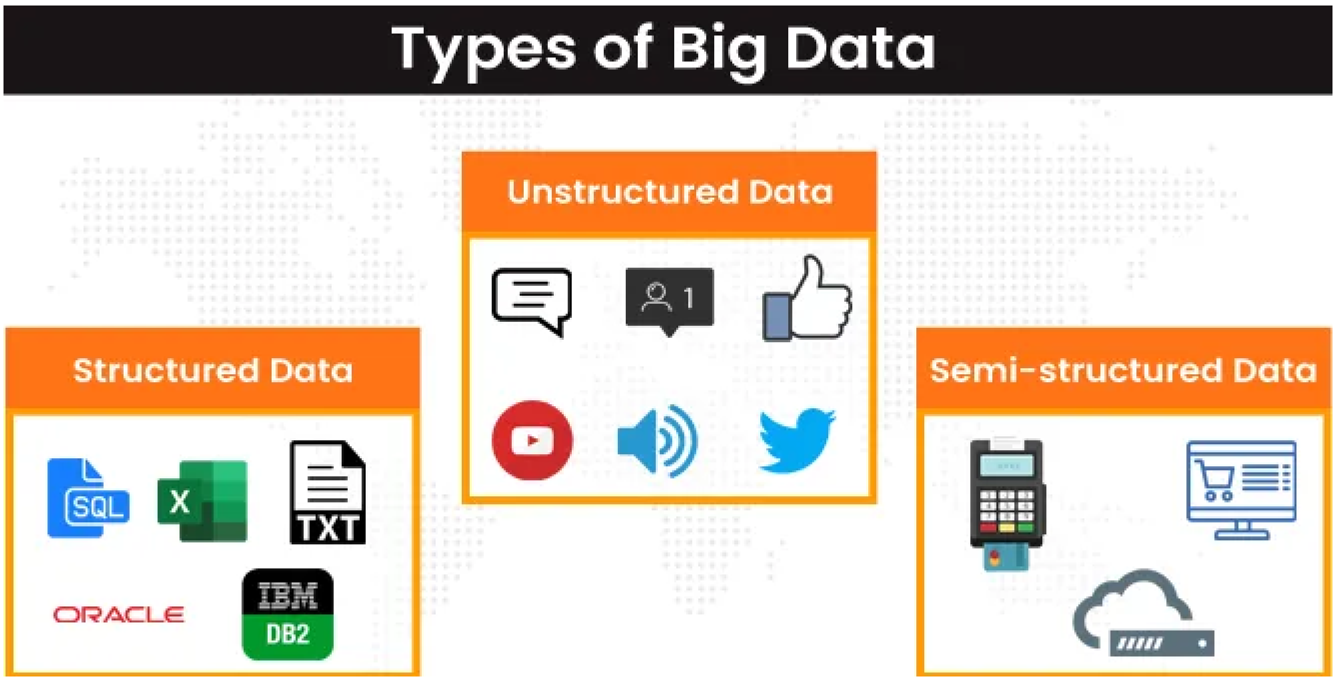
It’s important to understand the basics of Data Engineering. These concepts provide the foundation for managing, processing, and analyzing large amounts of data. By having a good grasp of these core concepts, individuals are better prepared to take on data engineering projects and handle any issues that may come up.

> **Big Data**

BIG DATA
Volume
Value
Veracity
Visualization
Variety
Velocity
Virality

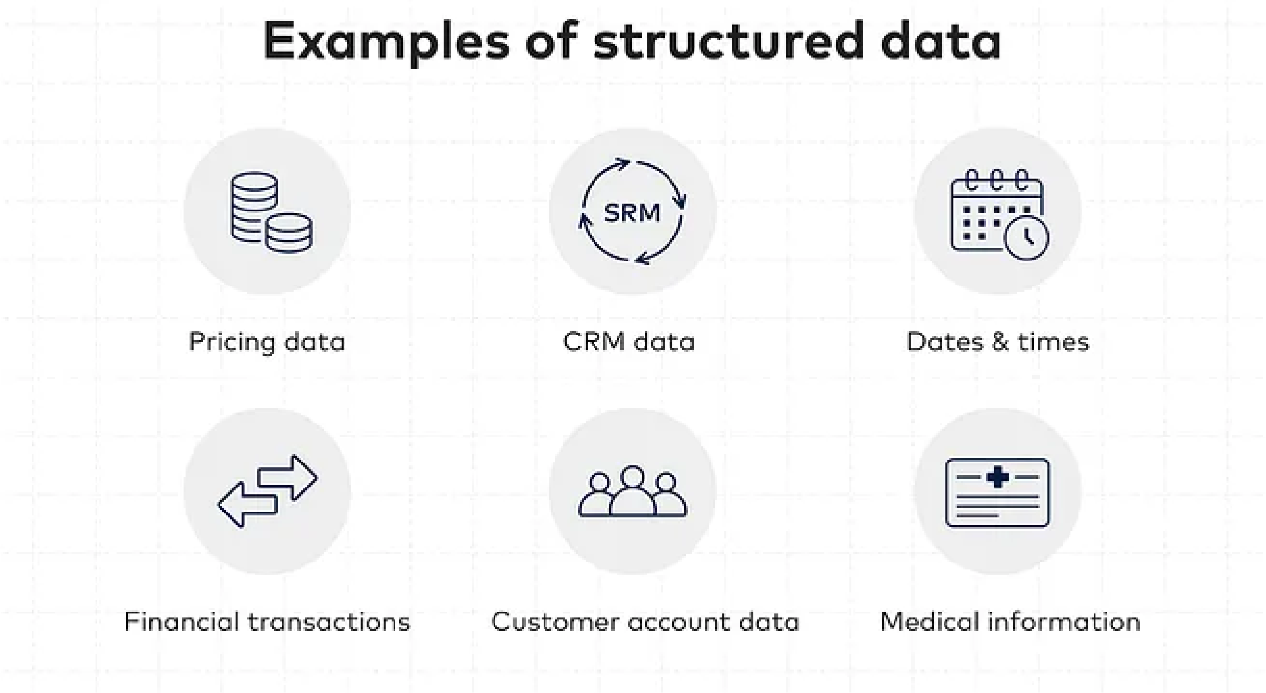
Big Data is the term used to describe extremely big and complex datasets that require advanced processing techniques in order to successfully analyze and extract knowledge from the data. This data can come from a wide range of sources, including social media, online transactions, sensors and other electronic devices. Big data can be structured, unstructured or semi-structured and ranges from terabytes (TB) to petabytes (PT) or even more in size.

Types of Data

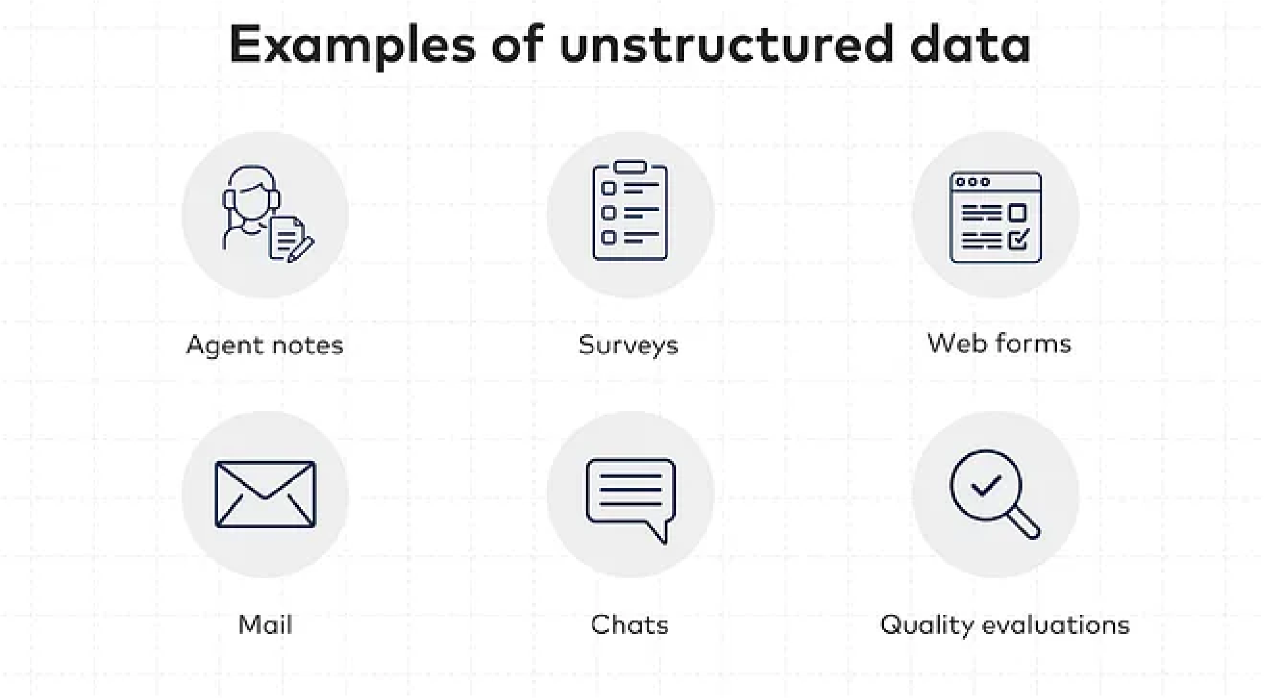


Data can be categorized into 3 types based on their characteristics and properties:

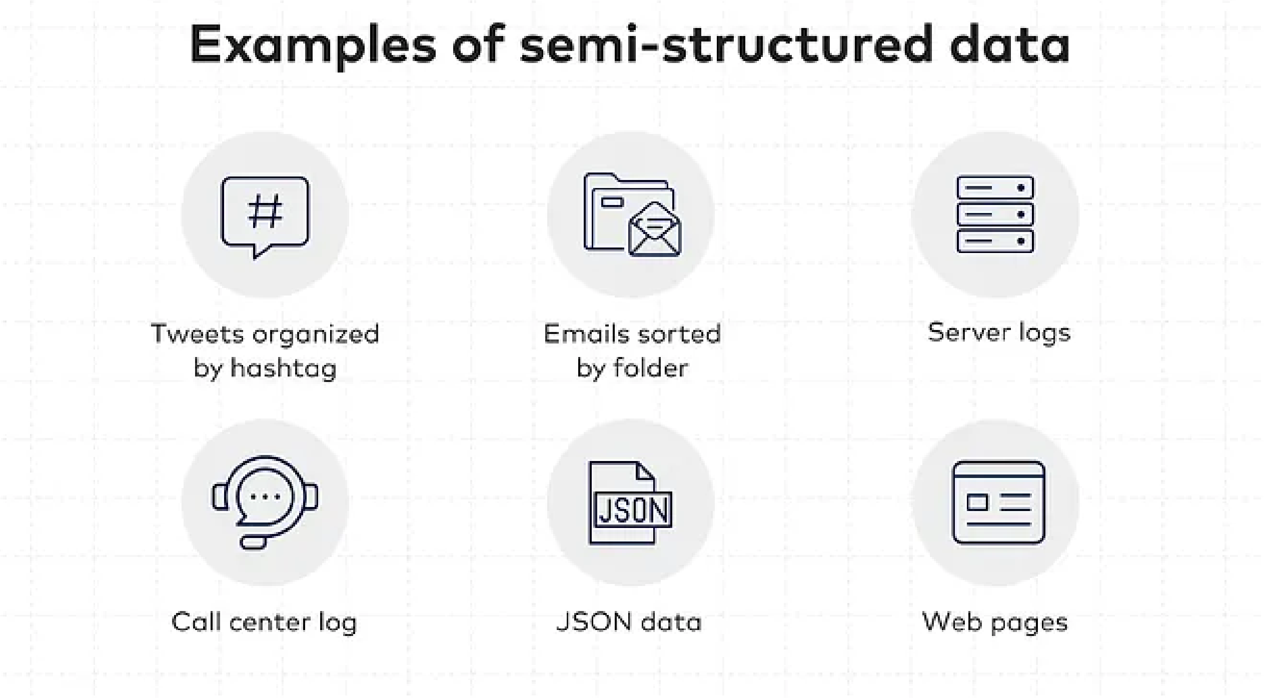
1. **Structured Data:** Structured data is referred to the data that is highly organized and formatted in a specific way that makes it easy to process and analyze data. Structured data is typical stored in Databases, Spreadsheets or any other formats that allow efficient storing and querying of data.



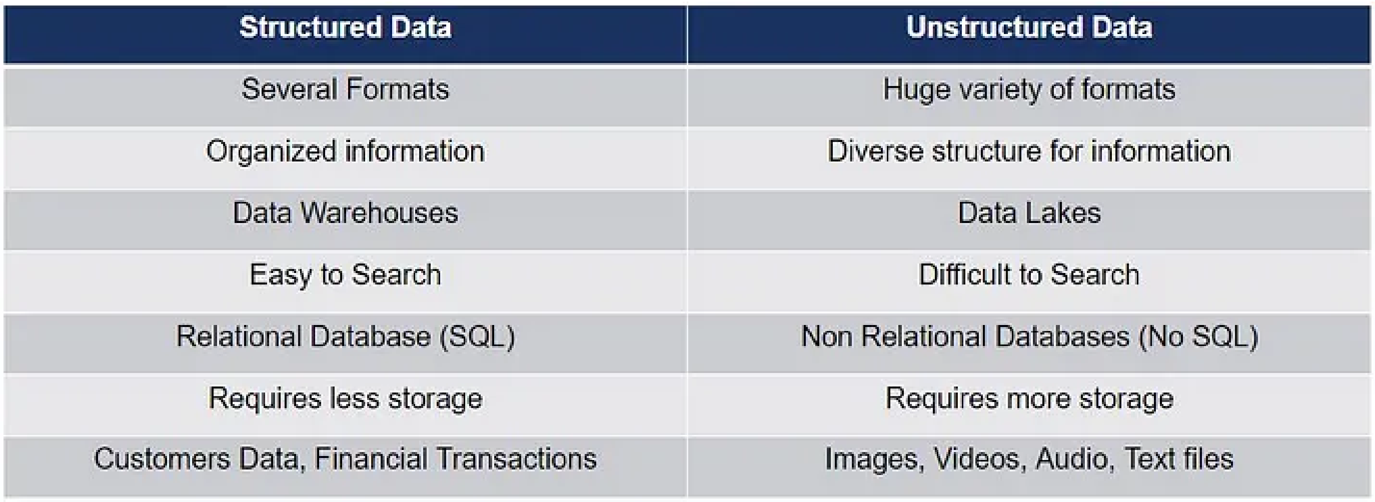
2. Unstructured Data: Unstructured data is referred to the data that does not have a specific format or structure which makes it difficult to analyze and process the data with traditional data processing tools.



3. Semi-Structured Data: Semi structured data refers to data that have some structure, but not organized as structured data. Semi structured data contain tags, metadata or other markers that help to organize and structure the data.



Structured Data vs Unstructured Data

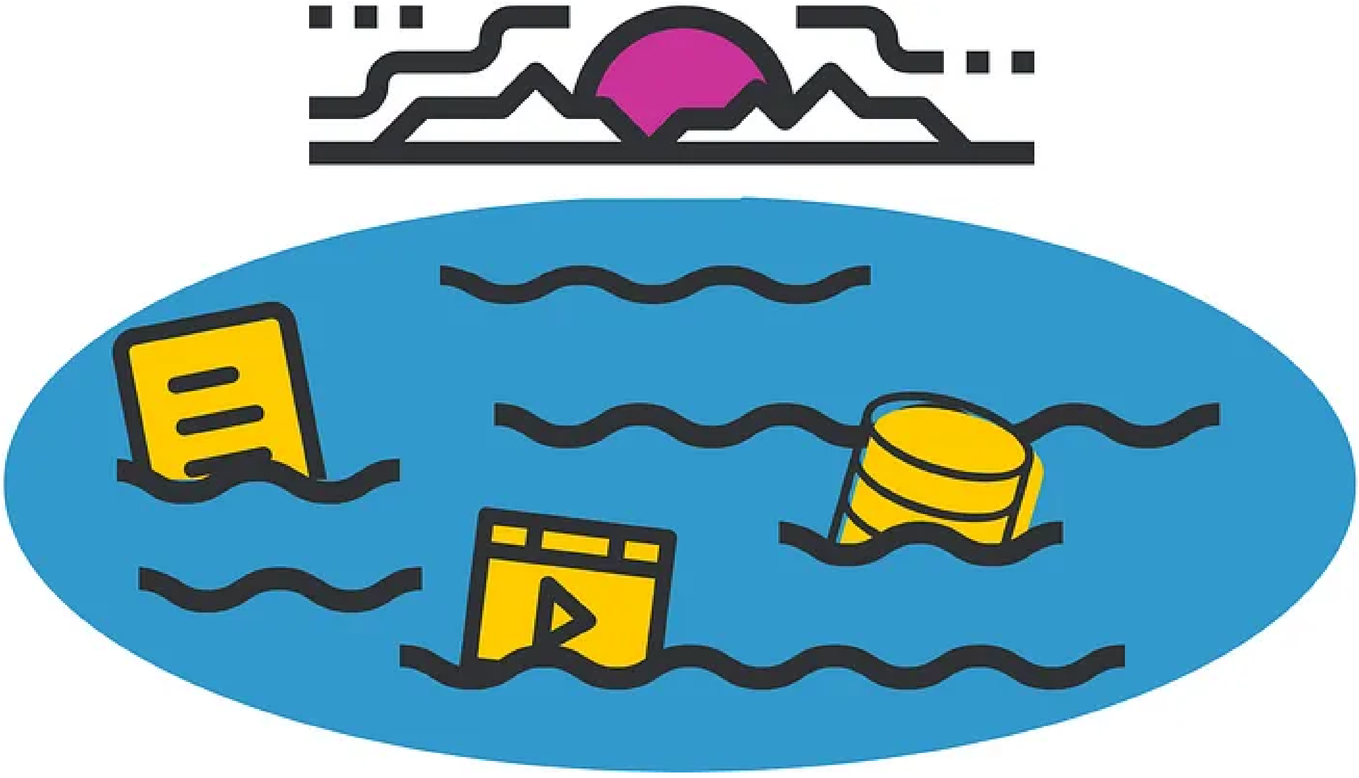


Structured Data	Unstructured Data
Several Formats	Huge variety of formats
Organized information	Diverse structure for information
Data Warehouses	Data Lakes
Easy to Search	Difficult to Search
Relational Database (SQL)	Non Relational Databases (No SQL)
Requires less storage	Requires more storage
Customers Data, Financial Transactions	Images, Videos, Audio, Text files

Structured vs Unstructured Data Comparison

Data Lake

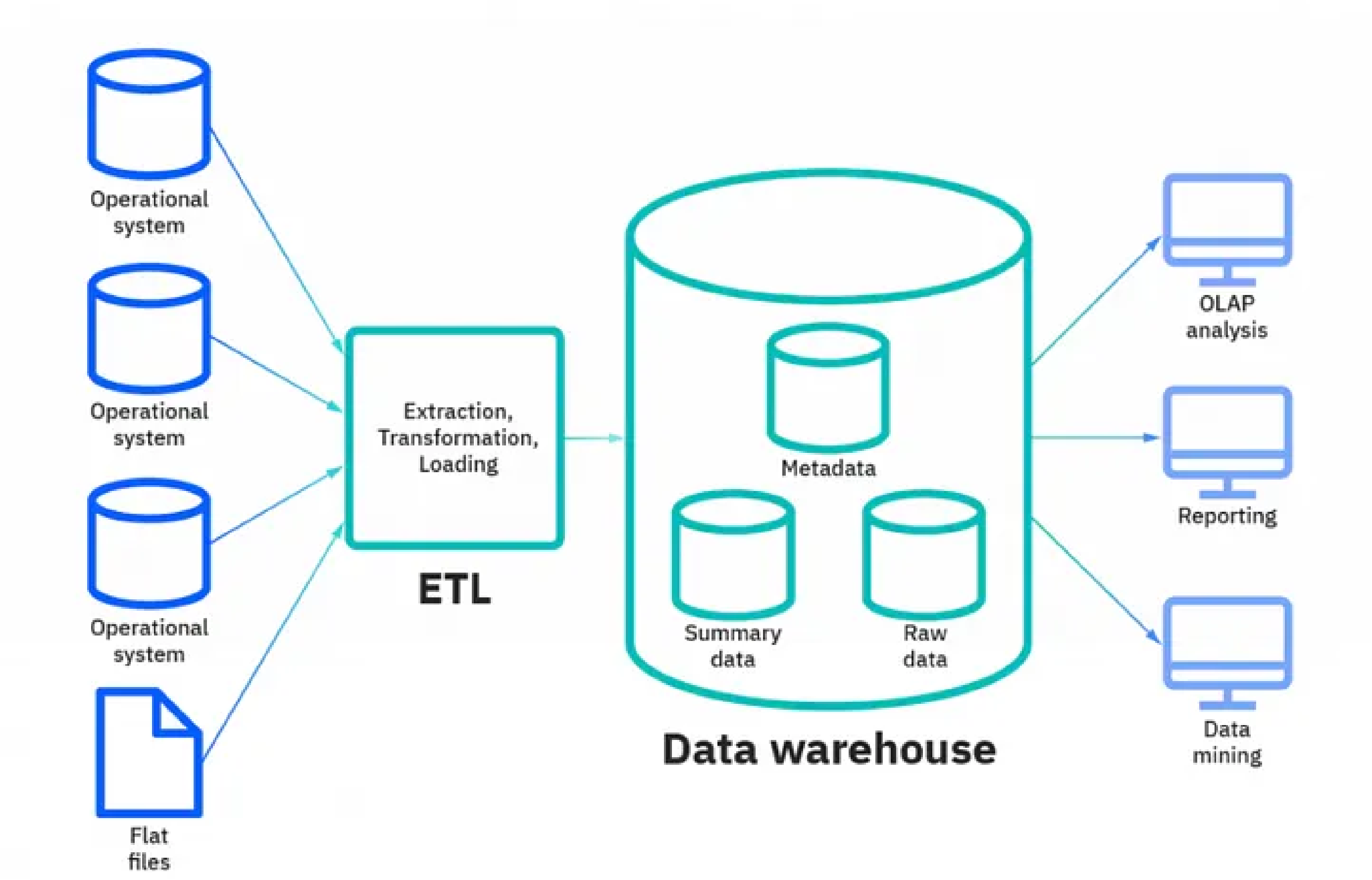
A Data Lake is a centralized repository that allows to store all structured, unstructured and unstructured data at any scale. It can store data in its native format and process any variety of it, ignoring the size limits. It provides secure and scalable platform that allows enterprises to ingest any data from any system at any speed even if the data comes from cloud or edge-computing systems. It stores any type or volume of data in full fidelity and process data in real time or batch mode.



Data Lake

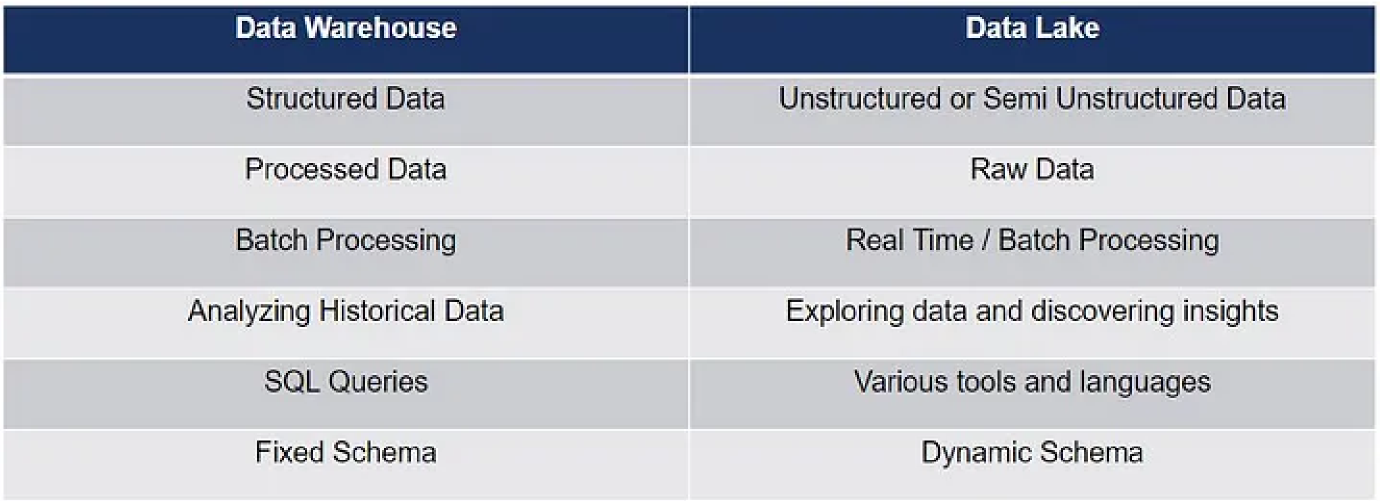
Data Warehouse

Data warehouse is a large central repository of information/data that can be used for reporting and analysis to make more informed decisions. It is designed to support business intelligence activities, such as data mining, reporting and Online Analytical Processing (OLAP). Data flows into a data warehouse from transactional system, relational databases and other sources. Business analysts, Data engineers, data scientists and decision makers access the data from data warehouse. Data warehouses provide a structured way to store and analyze large volumes of data, making it easier for organizations to make data-driven decisions and gain insights.



Data Warehouse

Difference between Data Lake and Data Warehouse



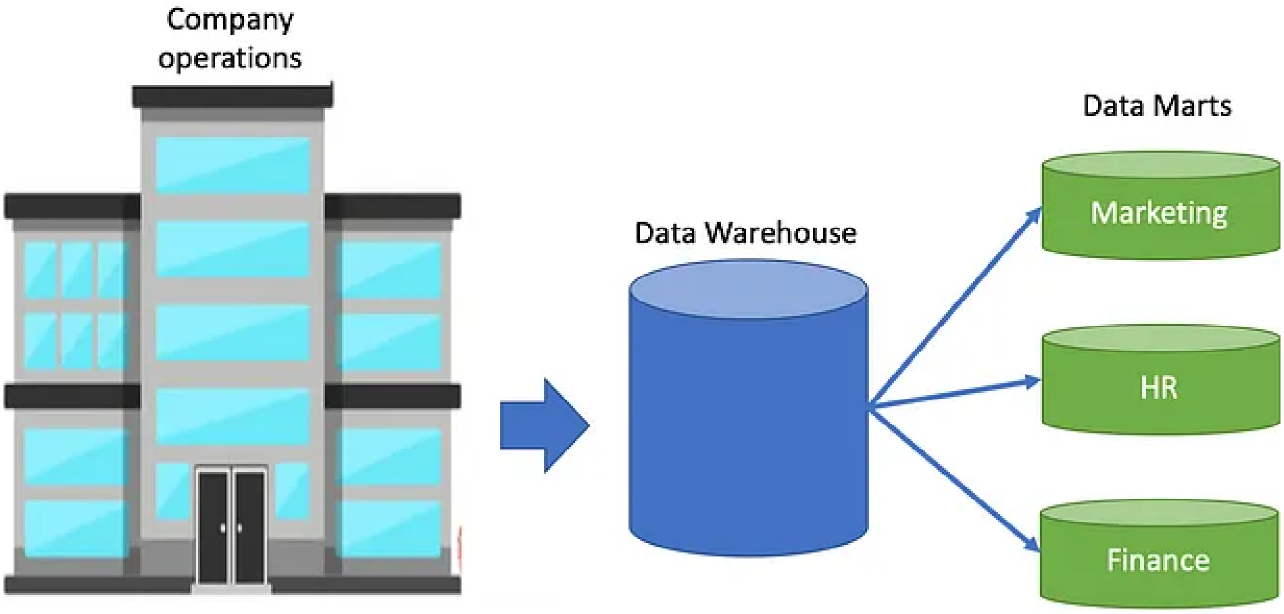
Data Warehouse	Data Lake
Structured Data	Unstructured or Semi Unstructured Data
Processed Data	Raw Data
Batch Processing	Real Time / Batch Processing
Analyzing Historical Data	Exploring data and discovering insights
SQL Queries	Various tools and languages
Fixed Schema	Dynamic Schema

Database

A database is an organized collection of structured data / information typically stored in a computer system and accessed electronically. It is essentially a structured way of organizing and managing data so that it can be easily accessed, managed and updated. Databases are typically organized into table that contain related data. For example, a customer database might contain tables for customer information, purchase history, and billing information. Each table contains columns that define the data that is stored, such as a customer's name, address, phone number, and email address.

Data Marts

A data mart is a data storage system that contains information specific to an organization's business unit. It contains a small and selected part of the data that the company stores in a larger storage system. Companies use a data mart to analyze department specific information more efficiently. It provides summarized data that key stakeholders can use to quickly make informed decisions.



Data Marts

Data Lakehouse

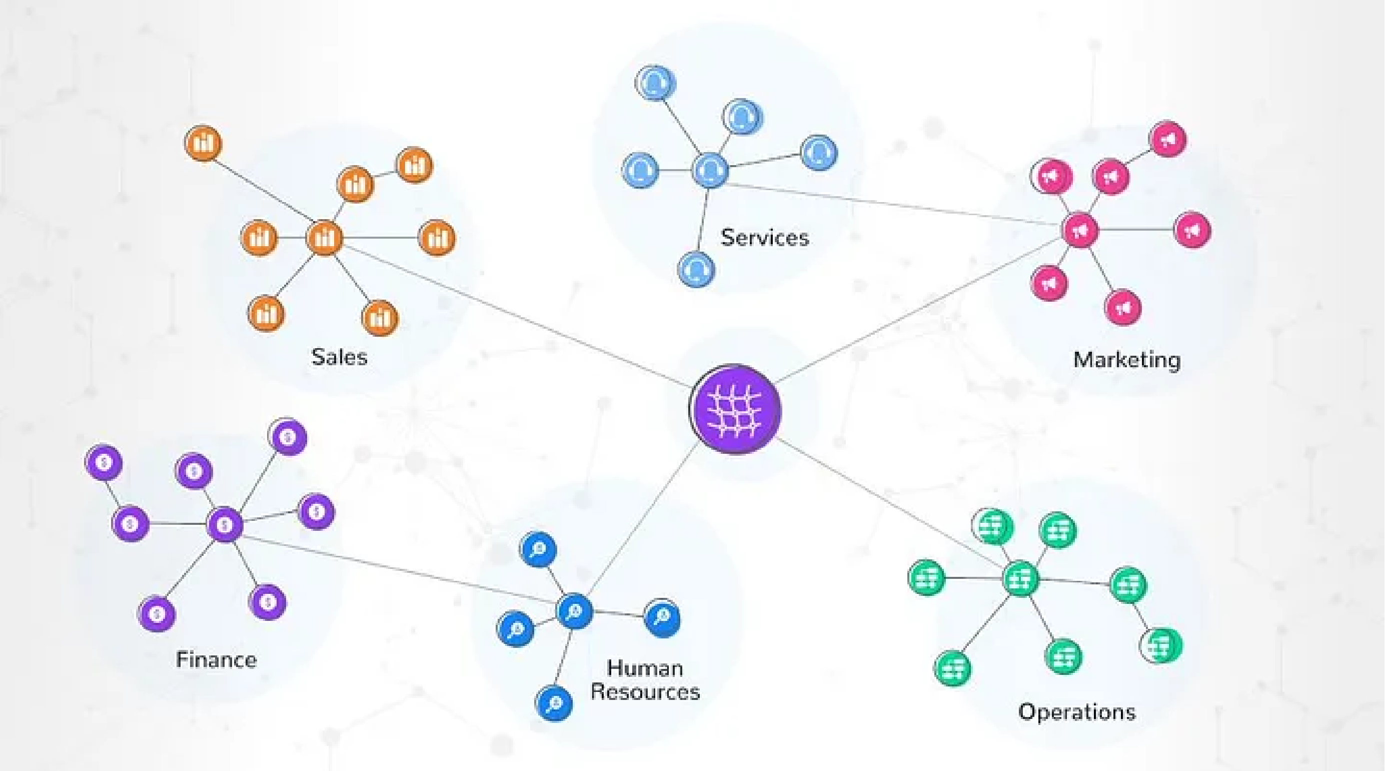
A data lakehouse is an integrated data management architecture that combines the strengths of both data warehouses and data lakes. It is designed to store, manage, and analyze large volumes of structured and unstructured data from multiple sources in a highly efficient and scalable way. Data from various sources like databases, applications, social media, IoT devices etc., are collected in a single repository that acts as a central source of data. The advantage of a data lakehouse is its ability to provide a unified platform that can handle both structured and unstructured data, which eliminates the need for separate data warehouses and data lakes. This reduces data duplication, improves data quality, and simplifies data management, making it easier to extract insights from data.

Data Warehouse
+
Data Lake
=
Lake House

Lake House

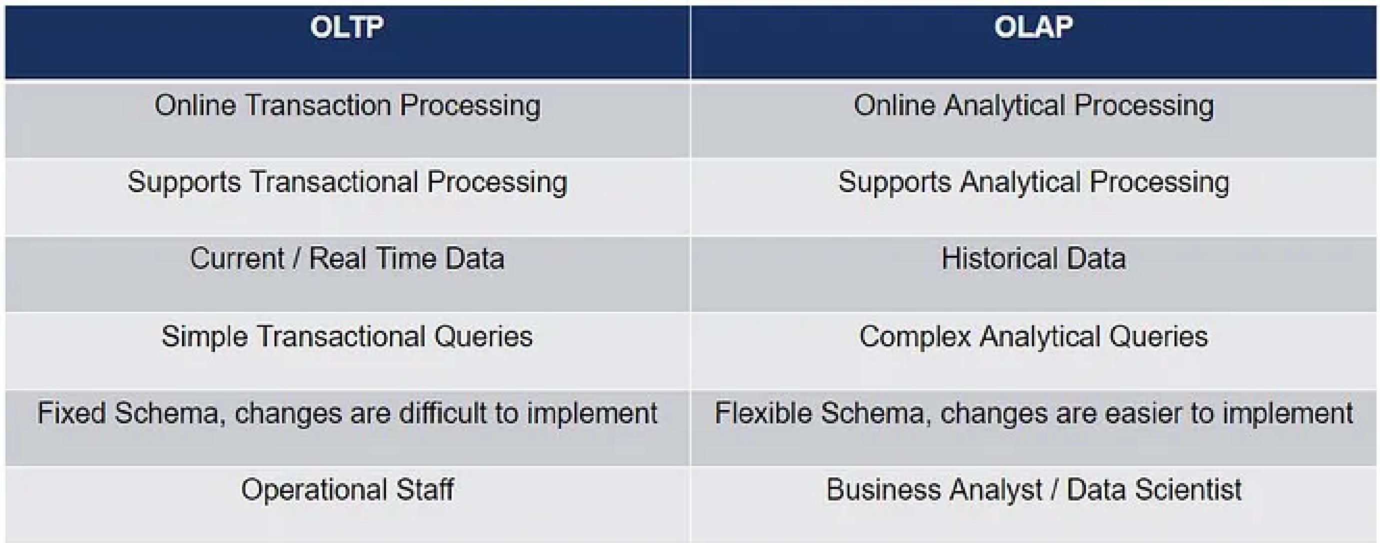
Data Mesh

Data mesh is a way of organizing data in a decentralized manner according to business domains, such as marketing, sales, and customer service. This approach gives more control and ownership to the teams responsible for producing each dataset. By distributing data ownership, bottlenecks and silos can be minimized, allowing for scalable growth without sacrificing data governance. In essence, the concept behind data mesh is that business domains should have the ability to create, access, and manage their own data products.



Data Mesh

Difference between OLTP and OLAP



OLTP	OLAP
Online Transaction Processing	Online Analytical Processing
Supports Transactional Processing	Supports Analytical Processing
Current / Real Time Data	Historical Data
Simple Transactional Queries	Complex Analytical Queries
Fixed Schema, changes are difficult to implement	Flexible Schema, changes are easier to implement
Operational Staff	Business Analyst / Data Scientist

OLTP vs OLAP

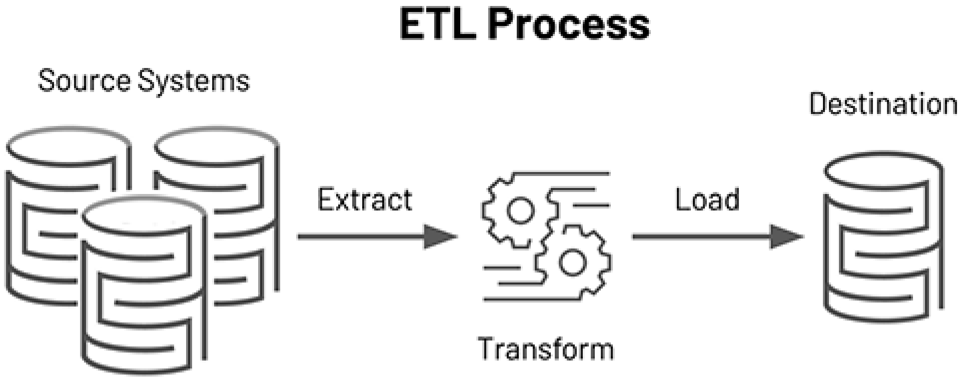
— —

What is ETL?

ETL stands for **Extract**, **Transform** and **Load**.

- **Extract** data from its original source
- **Transform** data by deduplicating it, combining it and ensuring quality
- **Load** data into the target database

ETL is a traditional approach that involves extracting data from various sources, transforming it into a consistent format and then loading it into the target data warehouse/data mart. ETL is best suited for batch processing and large volumes of data and is often used in situations where data needs to be cleaned, normalized and aggregated before it's loaded into target data warehouse or data mart.



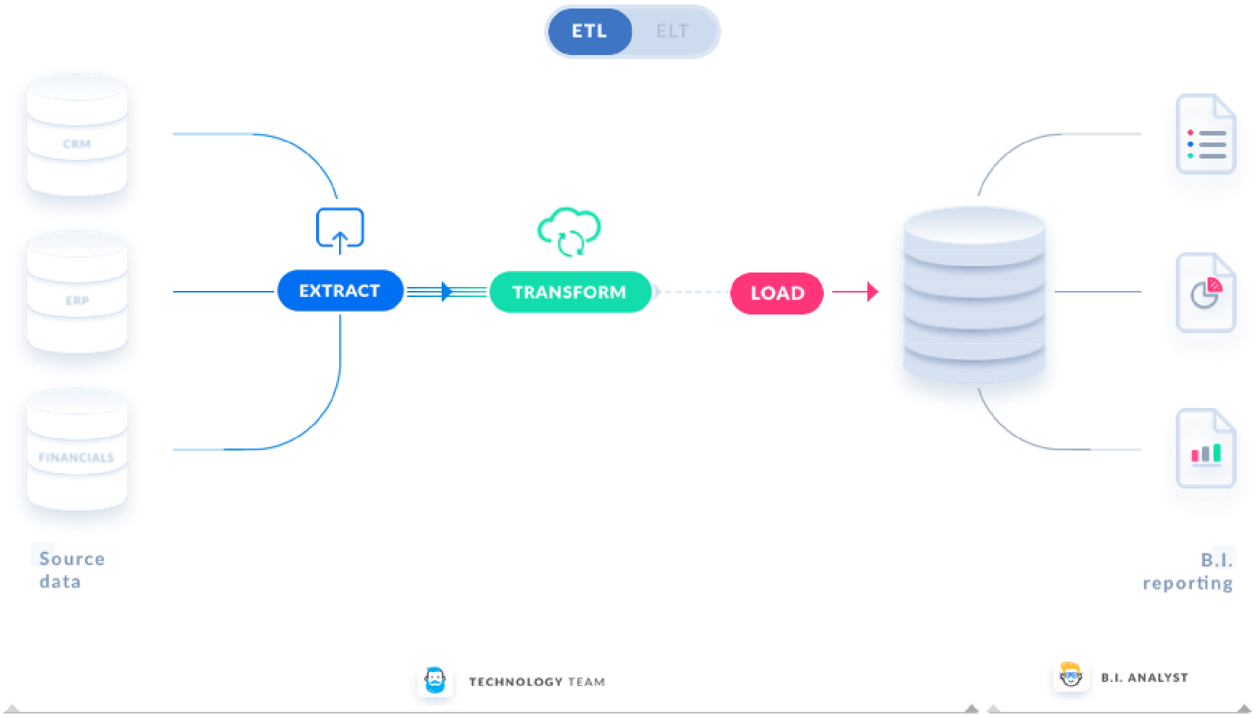
ETL Process

Steps in ETL

1. **Extract**: This is the first step of ETL, in this process the data is extracted from the target sources that are usually heterogeneous, such as business systems, APIs, transaction database, No SQL, XML and flat files into the staging area As some of the data types are likely to be structured outputs of widely used systems while other are semi structured JSON server logs.
2. **Transform**: The data extracted from the source server is typically raw and not useful in its original form. Therefore, it must be processed through a series of steps to cleanse, map, and transform it to make it useful for generating BI reports. This transformation step is a crucial part of the ETL process that enhances the value of the data. Some data requires no transformation and can be directly moved to the target system. Transformation in ETL includes *Cleaning, Deduplication, Filtering, Joining, Aggregation, Splitting, Derivation, Integration* etc.
3. **Load**: Loading data into the target Data Warehouse database is the last step of the ETL process. In a typical Data warehouse, huge volume of data needs to be loaded in a relatively short period. Hence, load process should be optimized for performance. In case of load failure, recover mechanisms should be configured to restart from the point of failure without data integrity loss. Data Warehouse admins need to monitor, resume, cancel loads as per prevailing server performance.

What is ELT?

ELT is an acronym for Extract, Load, and Transform. ELT is a modern variation on the older process of extract, transform, and load (ETL), in which transformations take place before the data is loaded. It's a process that extracts raw data from a source system to a target system, and the information is then transformed into the source or destination system for downstream applications. Unlike ETL, where data transformation processes occur on a staging area before being loaded into the target system, in ELT, data is loaded directly into the target system and converted there. In this way, ELT is most useful for handling enormous datasets and using them for business intelligence and data analytics.



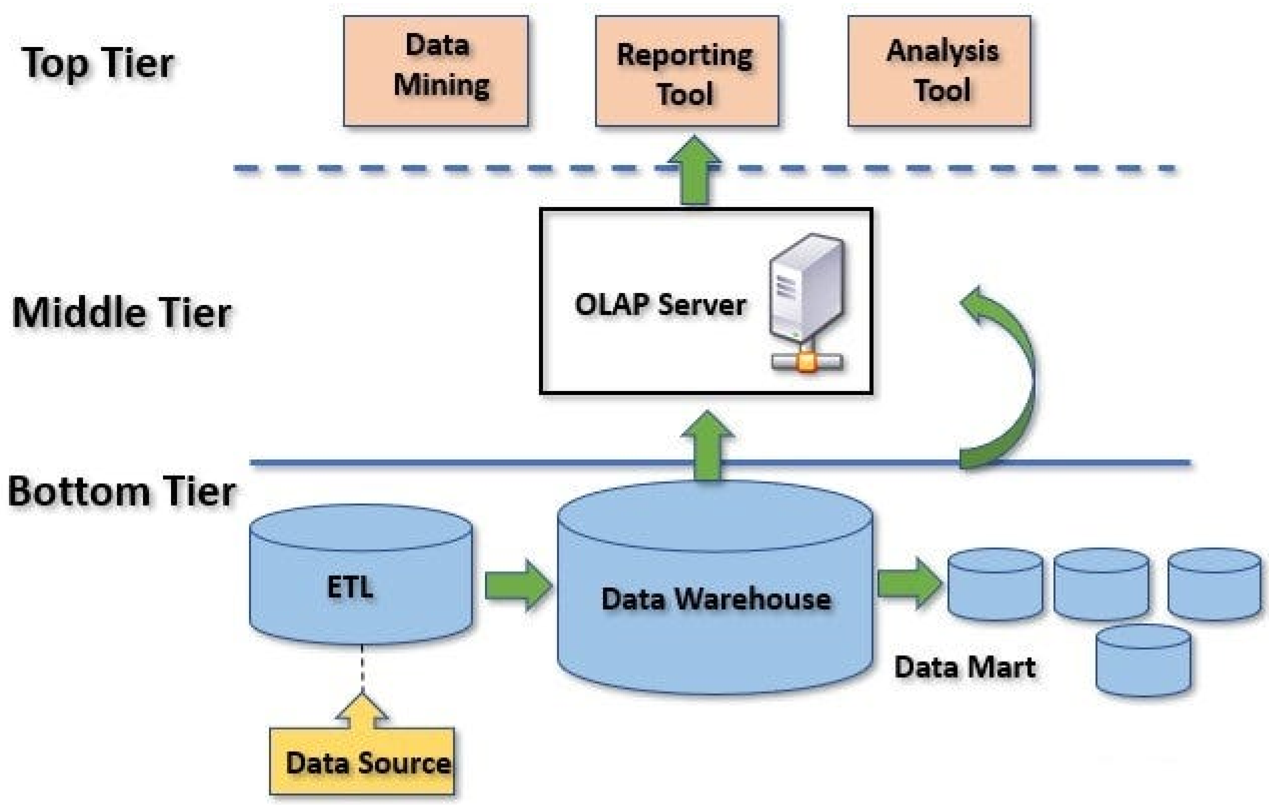
ETL vs ELT

3 Tier Architecture

The most widely used 3 tier architecture consists of:

- **Presentation Tier (Top):** The presentation layer is the top layer of the architecture, also known as the user interface layer. This layer is responsible for presenting the data to end-users, typically through a web or mobile application. The presentation layer interacts with the middle tier to retrieve the data needed to display the user interface.
- **Application Tier (Middle):** The middle tier in Data warehouse is an OLAP server which is implemented using either ROLAP or MOLAP model. For a user, this application tier presents an abstracted view of the database. This layer also acts as a mediator between the end-user and the database
- **Data Tier (Bottom):** The data layer is the bottom layer and is responsible for storing and managing the data. This layer contains the databases, data warehouses, or

other data storage solutions used to store the data. The data layer is responsible for retrieving the data requested by the application layer and returning it for processing.



Popular Tools for ETL

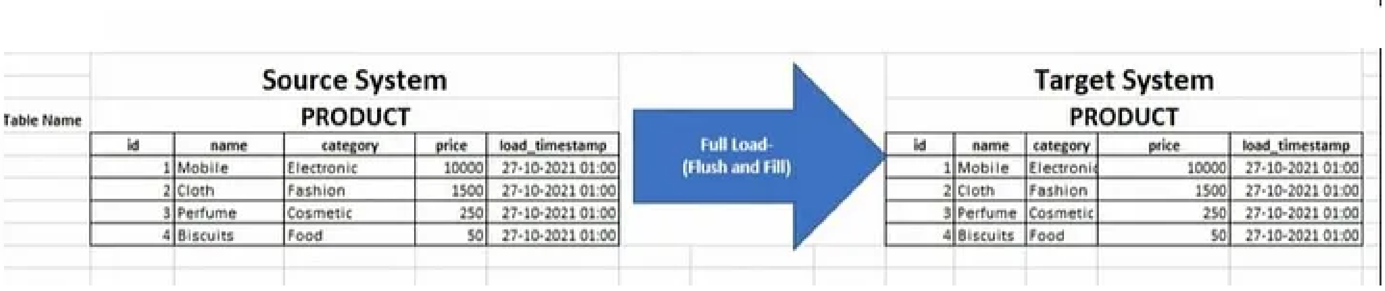
- Informatica Power Center

- Apache Airflow

- AWS Glue

Data Loading Strategies

- **Full Load:** This involves loading all the data from the source system into the target system, regardless of whether it has changed or not. This is typically used when the target system needs to be completely refreshed, and all the data gets updated. This can be time-consuming and resource-intensive, especially if the data volume is large.



Source System

Table Name PRODUCT

id	name	category	price	load_timestamp
1	Mobile	Electronic	10000	27-10-2021 01:00
2	Cloth	Fashion	1500	27-10-2021 01:00
3	Perfume	Cosmetic	250	27-10-2021 01:00
4	Biscuits	Food	50	27-10-2021 01:00

Full Load-
(Flush and Fill)

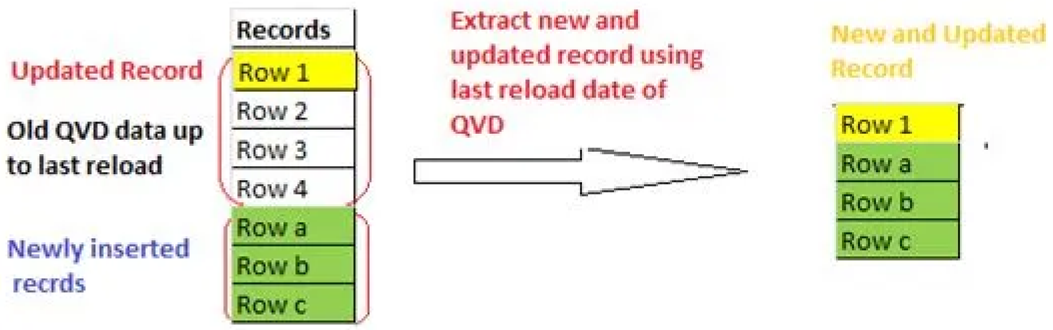
Target System

PRODUCT

id	name	category	price	load_timestamp
1	Mobile	Electroni	10000	27-10-2021 01:00
2	Cloth	Fashion	1500	27-10-2021 01:00
3	Perfume	Cosmetic	250	27-10-2021 01:00
4	Biscuits	Food	50	27-10-2021 01:00

Full Load

- **Incremental Load:** This involves loading only the data that has changed since the last load into the target system. This is typically used when the target system needs to be kept up-to-date and only the changes in the source system need to be reflected in the target system. Incremental loads are more efficient than full loads, as they reduce the amount of data that needs to be processed and loaded.



Incremental Load

- **Historical Load:** Historical load involves loading all the historical data available from the source system into the target system. This is typically used when building a data warehouse for the first time or when starting a new project. The historical load can be a large amount of data and it may take longer to load and transform the data.

Data Engineering Data Data Pipeline