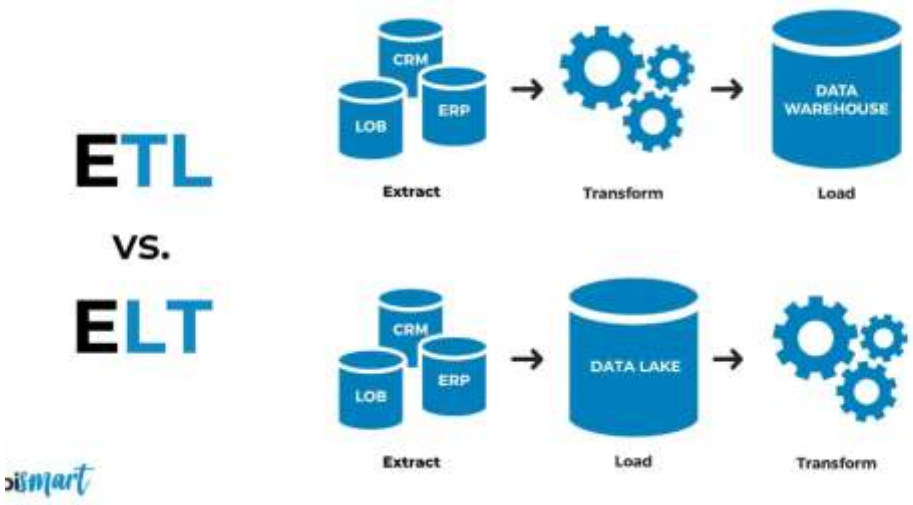
ELT vs ETL

ETL (Extract, Transform, Load):

1. **Extract:** Data is extracted from various sources (databases, files, APIs).
2. **Transform:** Data is cleaned, structured transformed into a desired format.
3. **Load:** Transformed data is loaded into a target database or data warehouse.

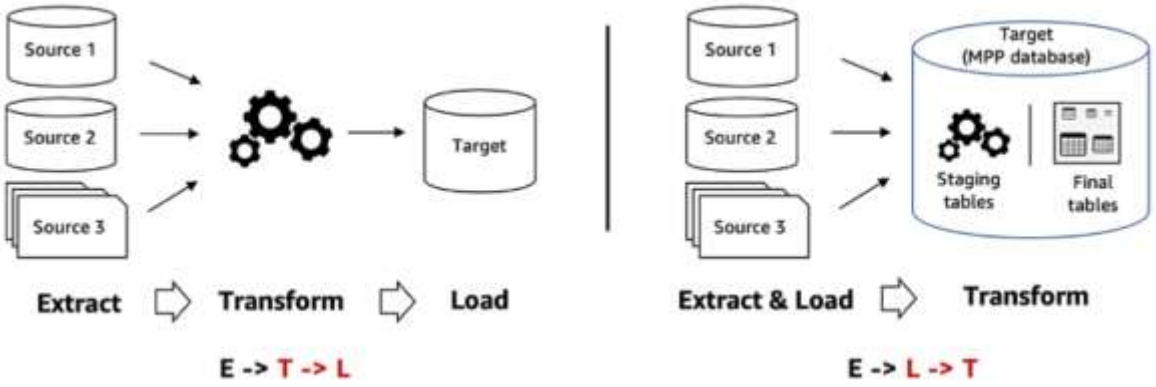
ELT (Extract, Load, Transform):

1. **Extract:** Data is extracted from various sources.
2. **Load:** Data is loaded into the target system (usually a data lake or a data warehouse).
3. **Transform:** Data is transformed within the target system.



Main Difference between ETL & ELT

ETL	ELT
Data Transformations happen **outside** the data warehouse	Data Transformations happen **inside** the data warehouse
Useful for small or medium sized datasets	Useful for large sized datasets
Can introduce delays due to transformation steps before loading	Typically faster as data is loaded first and transformed as needed



When to Use Each

- Use **ETL** when data needs significant preprocessing before loading, especially when dealing with legacy systems or when transformations are complex and must be completed before analysis.

- Use **ELT** when working with modern cloud-based data warehouses that can handle large-scale transformations efficiently, and when you want to load data quickly to enable fast querying and analysis.

ETL Use Case

Scenario:

A company needs to migrate its legacy on-premises data warehouse to a modern cloud-based data warehouse. The existing data requires extensive cleaning and transformation before it can be loaded into the new system.

Solution

As the transformation step is critical and complex, requiring significant preprocessing before loading the data into the target system therefore ETL process will be used.

ELT Use Case

Scenario:

A company wants to perform real-time analytics on its customer interaction data collected from various web and mobile applications.

Solution

Loading data quickly into the data lake enables fast access for real-time analytics. Transformations can leverage the power of the cloud-based processing engines.

Batch vs Streaming Pipeline

Batch Pipelines:

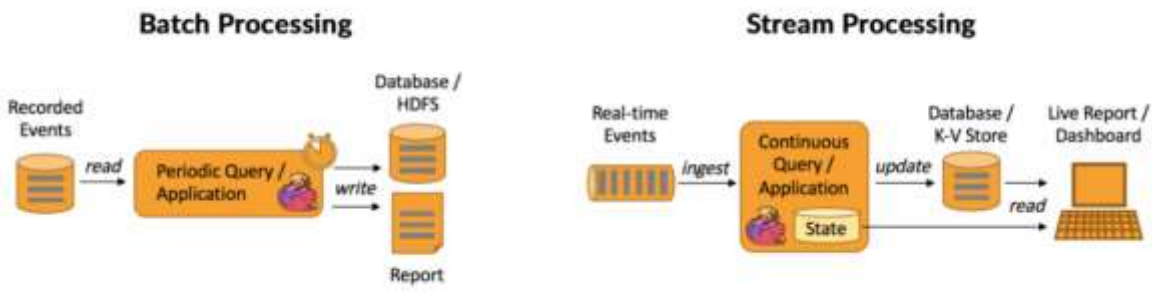
Processes data in large chunks at scheduled intervals (e.g., daily, hourly). Suitable for non-time-sensitive data processing tasks.

Streaming Pipelines:

Processes data in real-time as it arrives. Suitable for time-sensitive tasks where immediate insights are necessary.

Main Difference between Batch and Streaming Pipeline

Batch	Streaming
Processes a large volume of data at once.	Processes data continuously, record by record or in small chunks.
Higher latency due to the delay between data arrival and processing	Low latency as data is processed in near real-time.
Suitable for end-of-day reports, monthly aggregations, or data migrations.	Suitable for real-time analytics, monitoring systems, fraud detection, and live dashboards.



When to Use Each

Batch Pipelines Use Case

Scenario:

A company needs to generate monthly financial reports by aggregating transactional data from various sources.

Solution:

Financial reporting is typically performed at regular intervals and does not require real-time processing so Batch Processing will be the best choice.

Streaming Pipelines Use Case

Scenario:

A bank wants to detect fraudulent transactions in real-time to prevent potential losses.

Solution:

Real-time fraud detection is critical to prevent financial losses, requiring immediate processing of incoming transaction data therefore Streaming Pipelines will be the best choice.