# Data ingestion

* Is the process of gathering,managing and utilizing data efficiently
* Plays a foundational step in the data processing pipeline
* It involves the seamless important,transfer or loading of row data from diverse external sources into centralized system

### 3 type ingestion

* Batch – Data is collected and processed at scheduled intervals — daily reports,payroll,backups
* Real-Time – Data is ingested as its generated,enabling instant processing — fraud detection,live dash boards
* Micro-Batching – A hybrid approach-small batches processed frequently — IoT data, semi-live analytics

### The data ingestion work flow

* Data source identification – understand the data formate,structure and access method
* Data extraction –
* Data staging
* Data validation
* Data transformation – perform necessary transformations including cleaning,normalizing and enrichment
* Data loading – load the transformed into the target storage
* Data monitoring

### Batch Processing

* Data is collected,stored over a period of time
* Processed together

### 

### Characteristics

* Latency
* Data size
* Processing
* Scheduling - usually triggered by time
* Fault tolerance - easier to retry
* Coast - generally cheaper

#### 

#### Examples

1. Payroll systems – salaries for thousands of employees
2. Bank statement generation - monthly bank statements are generated
3. E-commerce order Processing - all unshipped orders are processed together
4. Credit card billing - At the end of the billing cycle

### 

### Advantages

* Efficiently at scale
* Automated scheduling
* error handling and retry logic
* Reduced manual intervention
* Resource optimization
* Consistency and standardization

### Limitations

* Deleayed result
* Resource spikes processing large volumes at once can sudden surges in CPU,memory.
* Complex debugging
* Limitted Flexibility
* Data staleness - the insights are only as fresh as the last batch run

DAta source(Database,logs,csv) → data ingestion (Scheduled jobes)→ Storage layer(HDFS,S3) → Batch processing → processed data → BI Tools/REports

#### BP tools

Appache spark

### Stream processing

* Continuous and real time processing f data as flaws in -one record or event at a time
* There is no delay b/w data and processing

### Characteristic

Latancy =low

Data size -desined for ongoing unbounded streams of data

### 

### Examples

Fraud detection in banking

REal time analytics of e-commerce

Social media feeds& notifications

Log monitoring and alerting

Online gaming

Stock market data &alerts and Video live streaming

Advantages: data handled as a continuous , unbounded flow

### 

### Srteam processing architecture

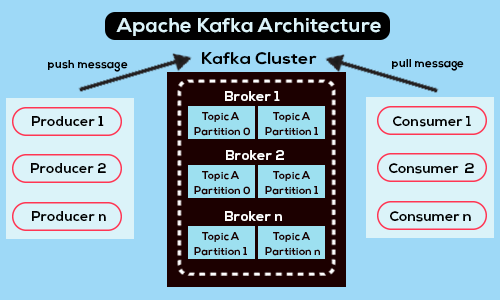
Continuous data sources → data ingestion

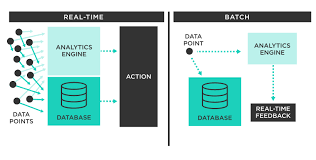
## Kafka Supporting streaming

### Apache Kafka is a **distributed event streaming platform**

* Producer
* Consumer
* Broker

Product details customer details





### ELT AND ETL

| **Feature** | **ETL** | **ELT** |
| --- | --- | --- |
| **Transformation Location** | Before loading (in staging server) | After loading (in warehouse) |
| **Speed** | Slower with large data | Faster with modern cloud systems |
| **Storage** | Only transformed data is stored | Raw + transformed data is stored |
| **Flexibility** | Less flexible | More flexible and scalable |
| **Cost** | Higher for on-prem processing | Lower with cloud usage-based cost |

* Data warelake -cleaned data (OLAP)
* Data lake

### ETL use cases

1. BAnking and financing
2. Healthcare
3. Retail POS systems
4. Insurance

Tools apache airflow

Talend open studio

### 

### ELT use cases

1. Streaming and analytics/ioT
2. E-commerce
3. Social media analytics
4. Real time fraud detection

# Data warehousing

Data warehousing refers to the process of collecting, storing , and managing large amounts of data from various sources in centralized repository, called a data warehouse. It enables organizations to make informed decisions.

### 

### Characteristics

* Integrated data from multiple sources
* Stores historical data for long term analysis
* Non-volatile

### Concepts

* OLTP
* OLAP
* DIMENSIONAL MODELING

OLTP:

OLTP stands for **Online Transaction Processing** — it's the backbone of systems that handle real-time, day-to-day transactions. Think of it as the engine behind banking apps, airline bookings, retail checkouts, and more.

Key Features of OLTP

* **Real-time processing**: Handles thousands of short, quick transactions like inserts, updates, and deletes.
* **ACID compliance**: Ensures data integrity through Atomicity, Consistency, Isolation, and Durability.
* **Normalized databases**: Data is structured to reduce redundancy and improve efficiency.
* **High concurrency**: Supports many users accessing and modifying data simultaneously.

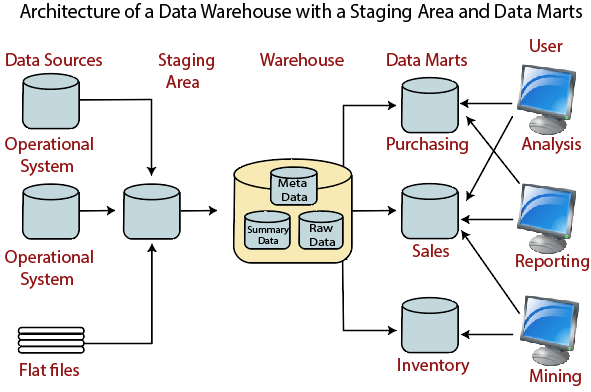
Examples of OLTP Systems

* ATM withdrawals and deposits
* Online shopping carts
* Airline ticket reservations
* Mobile banking

OLAP

OLAP stands for **Online Analytical Processing**, and it's all about helping businesses make sense of large volumes of data by enabling fast, multidimensional analysis.

What OLAP Does

* **Multidimensional analysis**: Lets you explore data across multiple dimensions like time, geography, product, etc.
* **Complex queries**: Supports advanced calculations, trend analysis, and forecasting.
* **Interactive exploration**: Users can drill down, roll up, slice, dice, and pivot data to uncover insights.
* 

### OLAP OPERATIONS

#### **1. Roll-up (Drill-up)**

* **What it does**: Aggregates data to a higher level.
* **Example**: From **day → month → quarter → year** or from **city → state → country**.
* **Use Case**: Summarizing sales from individual stores to region level.

#### **2. Drill-down**

* **What it does**: Opposite of roll-up; goes from higher-level summary data to more detailed data.
* **Example**: From **year → quarter → month → day**.
* **Use Case**: Analyzing low-performing days within a quarter.

#### **3. Slice**

* **What it does**: Extracts a single layer (slice) of the cube by fixing one dimension.
* **Example**: Viewing sales data for **2024 only** (fixing year dimension).
* **Use Case**: Analyzing one time period or one product category.

#### **4. Dice**

* **What it does**: Selects a **sub-cube** by choosing specific values of multiple dimensions.
* **Example**: Sales for **products A and B**, in **Q1 and Q2**, in the **East and West regions**.
* **Use Case**: Comparing multiple combinations of attributes.

#### **5. Pivot (Rotate)**

* **What it does**: Re-orients the data cube to view it from a different perspective.
* **Example**: Swapping rows and columns — showing **regions as columns** instead of rows.
* **Use Case**: Easier data comparison or visualization.

#### **6. Drill-across**

* **What it does**: Performs operations across **multiple fact tables** that share dimensions.
* **Use Case**: Comparing **sales vs. inventory** over the same period.

### Types of OLAP Systems

| **Feature** | **MOLAP** | **ROLAP** | **HOLAP** |
| --- | --- | --- | --- |
| Storage | Multidimensional | Relational Database | Both |
| Performance | Very fast | Moderate | Balanced |
| Scalability | Limited | High | High |
| Flexibility | Low (predefined) | High (ad-hoc queries) | Medium |
| Best For | Summary analysis | Large detail queries | Mixed-use scenarios |

### Where OLAP Shines

* Business Intelligence (BI)
* Financial reporting
* Sales forecasting
* Market trend analysis

### Dimensional Modeling

Dimensional Modeling is a powerful technique used in **data warehouse design** to make data easier to understand and analyze. It was popularized by **Ralph Kimball**, and it's especially useful for supporting **OLAP** systems and business intelligence tools.

### 

### Core Concepts

* **Fact Tables**: Store measurable, quantitative data (e.g., sales amount, profit).
* **Dimension Tables**: Contain descriptive attributes (e.g., product name, customer location) that provide context to facts.
* **Star Schema**: A central fact table connected to dimension tables — simple and fast for querying.
* **Snowflake Schema**: A more normalized version of the star schema — dimension tables are split into sub-dimensions.

### Why Use Dimensional Modeling?

* **Simplifies complex data**: Easier for analysts and business users to navigate.
* **Improves query performance**: Optimized for SELECT operations and aggregations.
* **Supports historical analysis**: Tracks changes over time using techniques like Slowly Changing Dimensions (SCDs).
* **Scales well**: Can handle large volumes of data with consistent structure.

### Steps to Build a Dimensional Model

1. **Identify the business process** (e.g., sales, inventory, customer support).
2. **Declare the grain** — the level of detail (e.g., daily sales per store).
3. **Define dimensions** — who, what, where, when.
4. **Define facts** — numeric measures tied to the business process.
5. **Design schema** — usually a star or snowflake schema