

## Data & Information

### Data vs. Information

- **Data:** Raw, unprocessed facts or figures that do not carry any specific meaning on their own.
  - Can exist in various forms, such as numbers, text, or symbols.
  - Example: A list of temperatures: 23, 25, 28, 22.
- **Information:** Processed or organized data that provides meaning, context, or insights to the user.
  - Derived from data through processes such as analysis, sorting, or summarization.
  - Example: "The average temperature for the week is 25°C."

#### FAQ: What does it mean for data to be “transformed” or “processed”?



Data transformation refers to the manipulation and conversion of raw data into a more meaningful and useful format, which becomes **actionable information** (a.k.a. information that helps people make decisions). Therefore, any organizing, summarizing, analyzing, or interpreting of raw data into extract insights or knowledge is considered data transformation.

### Database Applications

Software systems that collect, process, and transform data into meaningful information. These applications enable decision-making by providing processed outputs that users need.

- **Process Flow:** Input Data => Database Application => Information
- **Example:** In a Student Information System, raw data such as grades and attendance records are entered into the system. The application processes this data and generates reports like "Top-performing students" or "Attendance rates per class."

### Database Architectures

- **Single Monolithic Application:**
  - A single application performs all tasks, including user interface, data processing, and data storage. All functionalities are tightly integrated, making the system simpler but less scalable.
  - Described as a data-based application, rather than a database application

- **Client-Server Architecture:**
  - **Client (Database Application):** Sends requests (e.g., queries) to the server.
  - **Server (Database Server):** Processes requests and returns results to the client.
  - This architecture separates the user interface (client) from the database processing (server), allowing multiple users to interact with the system simultaneously.
- **3-Tier Architecture:**
  - **Presentation Layer (Human Interaction):** User interface (e.g., web pages, mobile app UI).
  - **Logic Layer (Processing):** Application logic, processing, and rules (e.g., APIs, server-side code).
  - **Data Layer (Database Server):** Database and data management.
  - This architecture is modular and scalable, making it ideal for large, distributed applications.
  - Example: An e-commerce platform like Amazon.

### Characteristics of Quality Information

- **Accessible:** Information must be easily obtainable by authorized users in the required format and at the right time.
  - Example: A customer portal that allows users to view account balances anytime.
  - Usually seen if it is mentioned that the information is easy to access (i.e. reports are easy to generate)
- **Accurate:** Information should be free from errors. Errors often occur when incorrect data is input.
  - Example: A financial report showing the exact revenue for a quarter.
  - Usually seen if the developer/operator confirms the accuracy of things like formulas.
- **Complete:** All essential details should be present. Missing data can lead to incomplete or misleading conclusions.
  - Example: An employee record containing all fields like name, ID, address, and position.
- **Economical:** The cost of producing information should not outweigh its value.

- Example: Generating a summary of sales data instead of printing full transaction logs.
  - Usually seen if the cost of the method of obtaining data is explicitly mentioned.
- **Flexible:** Information should be usable for various purposes, not just for one task.
  - Example: Sales data being used to analyze both revenue trends and employee performance.
  - Usually seen if the database is to be used for multiple applications.
- **Relevant:** Information must address the user's specific needs and decision-making requirements.
  - Example: A marketing team using customer demographics to design targeted campaigns.
  - Usually seen if user needs are explicitly stated.
- **Reliable:** Users should trust the information based on its source and data collection methods.
  - Example: Survey results collected from verified participants instead of anonymous sources.
  - Usually seen in scenarios where users can mis-input data.
- **Secure:** Information must be protected from unauthorized access.
  - Example: A database with encryption and access control measures in place.
- **Simple:** Information should not be overly complex. Users should easily understand and act on it.
  - Example: A bar chart summarizing monthly sales instead of a complex multi-dimensional report.
- **Timely:** Information must be current and delivered when needed to support decisions.
  - Example: A daily stock market update for investors.
  - Usually seen in situations that involve a time dimension (i.e., the user immediately sees the information).
- **Verifiable:** Information should be cross-checked and validated for accuracy.
  - Example: Matching sales data with receipt records to ensure correctness.
  - Usually seen if the end-user can manually verify/check information.