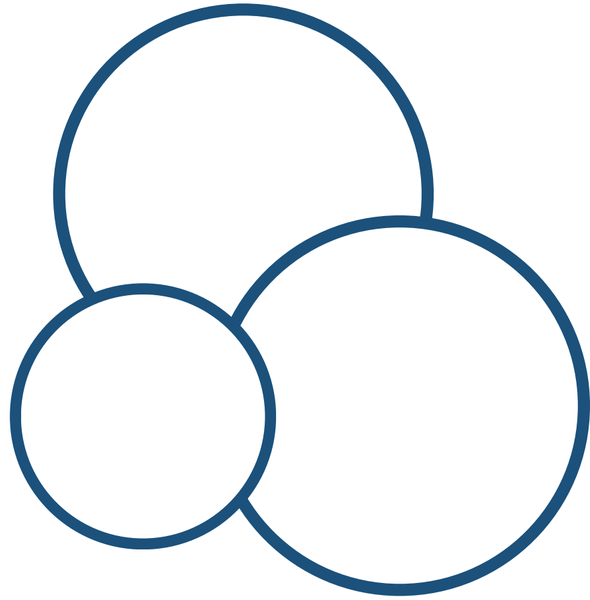
DATATECH LABS.

Data Engineering Fundamentals.

Week 1: Introduction to Data Engineering & Data Modeling

Outline: Week 1

- Overview of Data Engineering
- Role of a Data Engineer
- Data Lifecycle
- Data Modeling Concepts
- Relational vs. NoSQL databases
- Data modeling techniques for big data
- Introduction to distributed database systems



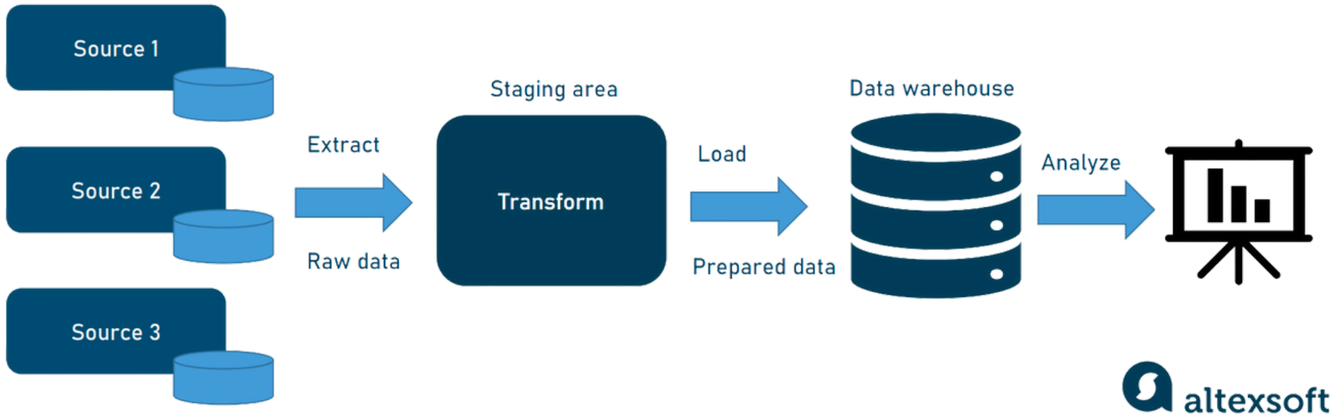
Introduction to Data Engineering & Data Modeling

Defining Data Engineering

What is Data Engineering?

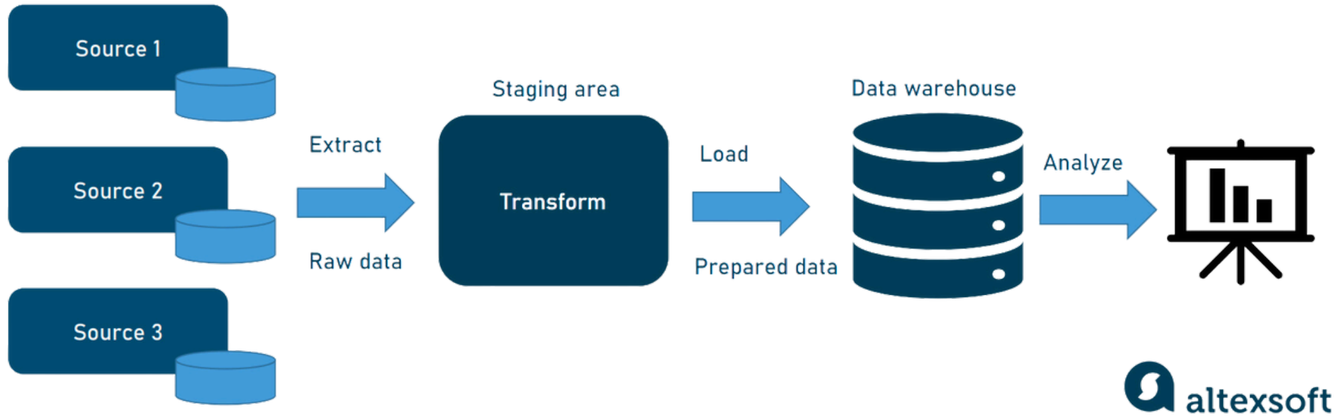
- **Definition**: Data Engineering is the process of designing and building systems that allow for the **collection**, **storage**, and **analysis** of large datasets.

- **Discussion Point**: Why is data engineering important in today's data-driven world?



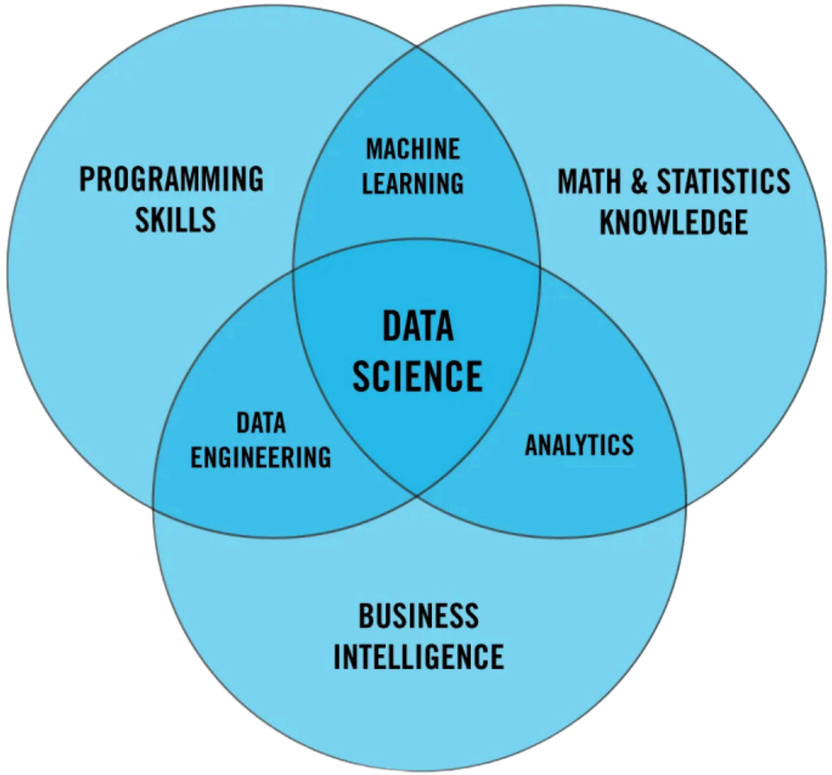
Data Engineering Terminologies

- **Data**: Is the new oil. Data is any collection of information that can be used to make decisions or improve understanding of a situation. It can be in any form, including numbers, text, images, audio, or video.
- **Data Engineering**: Is the discipline of designing, developing, and maintaining the architecture, tools, and systems that are used to **collect**, **store**, **process**, and **analyze** data.
- **Big Data**: Is the term used to describe the large and complex datasets that are too big to be processed using traditional data processing methods.



Roles in the Data World

- Data Engineer
- Data Scientist
- Data Analyst
- Machine Learning Engineer
- Database Administrator



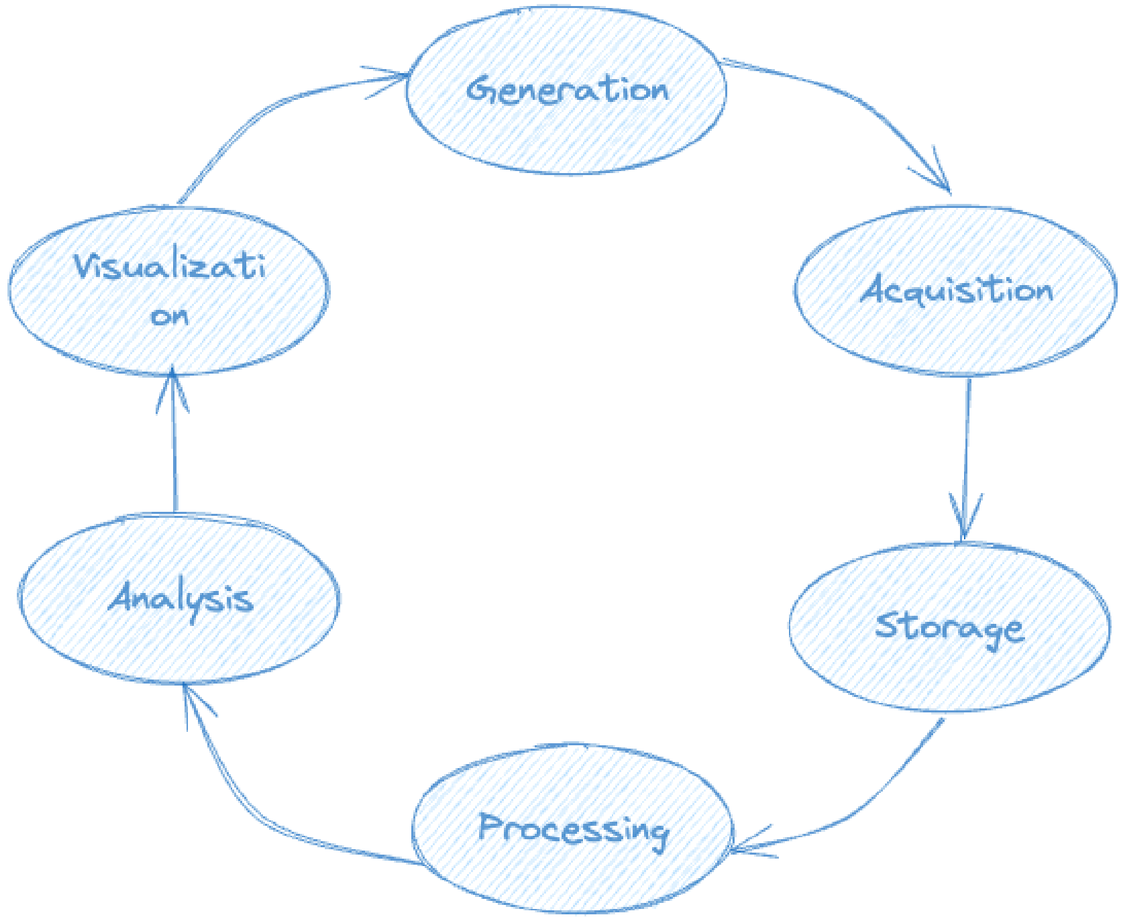
The Role of a Data Engineer

- **Responsibilities**:
  - Designing data pipelines
  - Building data infrastructure
  - Data collection and integration
  - Ensuring data quality and consistency

- **Discussion Point**: How does the role of a data engineer differ from a data scientist?

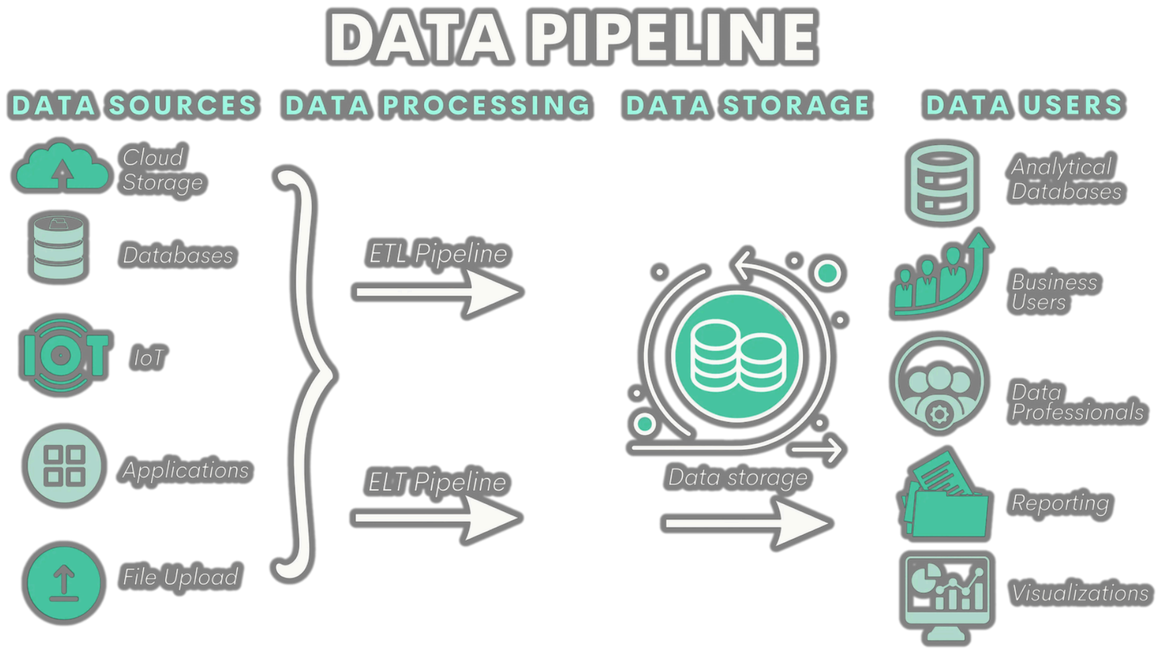
The Data Lifecycle

- Data Generation
- Data Acquisition
- Data Storage
- Data Processing
- Data Analysis
- Data Visualization



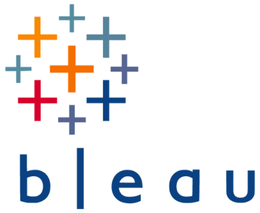
Key Concepts in Data Engineering

- **Data Pipeline**: A set of processes that move data from one system to another.
- **ETL (Extract, Transform, Load)**: The process of extracting data from sources, transforming it into a usable format, and loading it into storage.
- **Data Lake**: A storage repository that holds vast amounts of raw data in its native format.
- **Data Warehouse**: A system used for reporting (BI) and data analysis, storing integrated data from multiple sources.



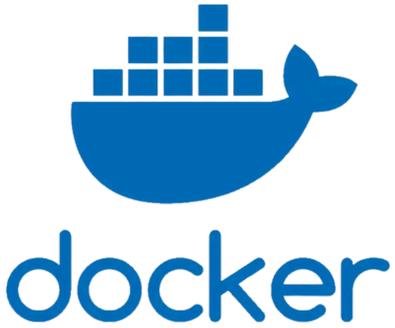
Tools and Technologies

- Apache Hadoop
- Apache Spark
- Apache Kafka
- Apache Flink
- Apache Airflow
- SQL Databases (e.g., PostgreSQL, MySQL)
- NoSQL Databases (e.g., MongoDB, Cassandra)
- Business Intelligence (BI) Tools



Course Tools and Setup

- Python
- Jupyter Notebook
- Docker
- GitHub
- Instructions for installing and setting up the tools



Hands-on: Setting Up Your Environment

- Download Python: python.org
- Download Docker Desktop: docker.com
- Create your python virtual environment with command:
  - `python3 -m venv <env_name>`
- Create your GitHub account: github.com
  - Create a new repository for the course.
  - Push your Jupyter Notebook to the repository.

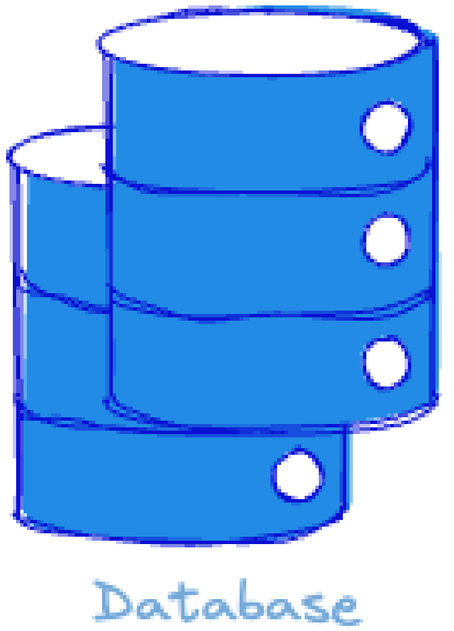
Hands-on:

Introduction to Python for Data Engineering.
Setting up your environment.

Data Modeling and Architecture

First: What is a Database?

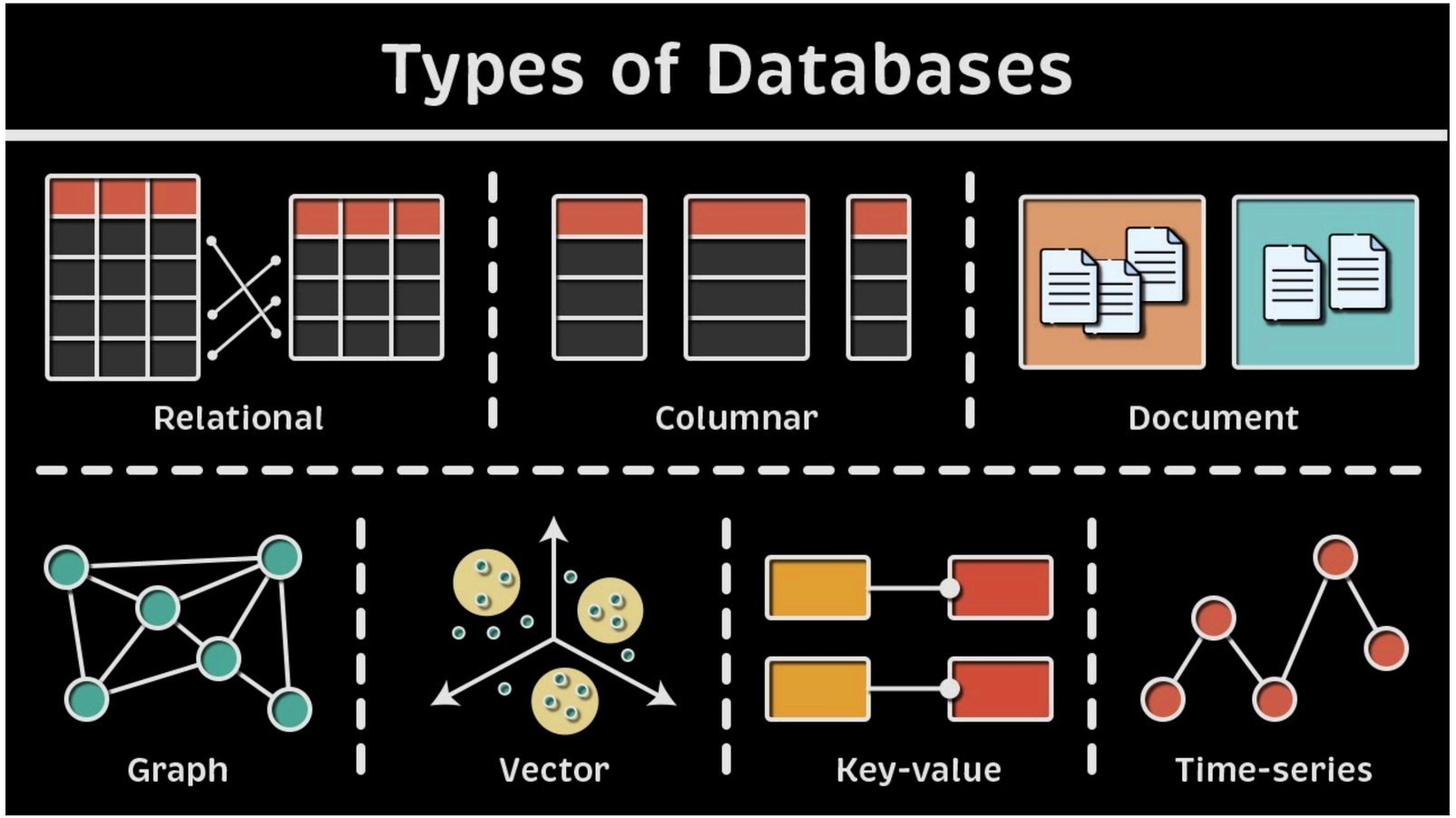
- A **database** is an organized collection of information, or data, typically stored electronically in a computer system.
- Key Components:
  - **Data**: Raw facts or figures, which might include numbers, text, or multimedia.
  - **Database Management System (DBMS)**: Software that interacts with users, applications, and the database itself to capture and analyze data.
- Purpose:
  - Databases are used to store, retrieve, and manage data efficiently.
  - They support operations such as data querying, updating, and administration.



Types of Databases

Majorly:

- Relational
- Non-Relational (NoSQL)



Relational Databases

- Based on the relational model of data
- Organizes data into tables (relations) with rows and columns
- Key concepts:
  - Tables, Columns, Rows
  - Primary Keys, Foreign Keys
  - Normalization
- Examples: MySQL, PostgreSQL, Oracle

NoSQL Databases

- "Not Only SQL" or non-relational databases
- Designed to handle various forms of data and large volumes of data
- Types of NoSQL databases:
  a. Document stores (e.g., MongoDB)
  b. Key-value stores (e.g., Redis)
  c. Column-family stores (e.g., Cassandra)
  d. Graph databases (e.g., Neo4j)

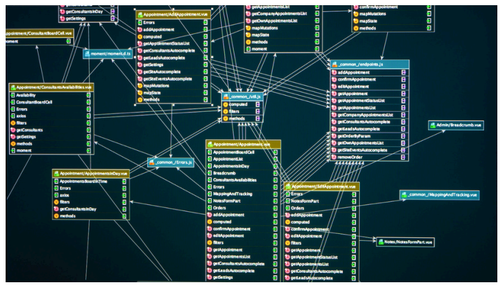
Choosing Between Relational and NoSQL

Aspect	Relational	NoSQL
Data Structure	Structured	Unstructured/Semi-structured
Schema	Fixed	Flexible
Scalability	Vertical	Horizontal
ACID Compliance (atomicity, consistency, isolation, and durability)	Yes	Often sacrificed for performance
Use Cases	Complex queries, transactions	High volume, high velocity data

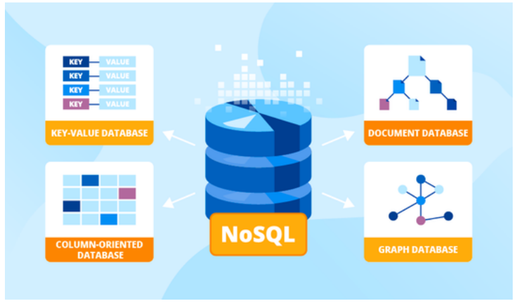
Data Storage Systems

- The type of data storage that is used depends on the **volume**, **velocity**, and **variety** of the data.
- Common data storage options include:

Relational Databases. Structured data

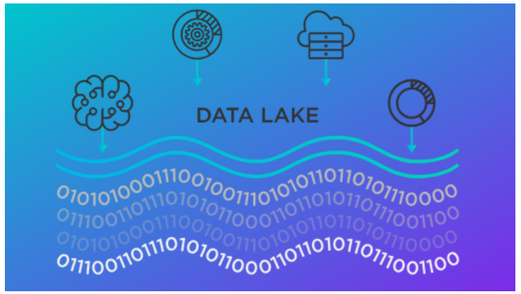


NoSQL Databases. Unstructured data



Data Warehouses. Structured Big data

Data Lakes. Big data

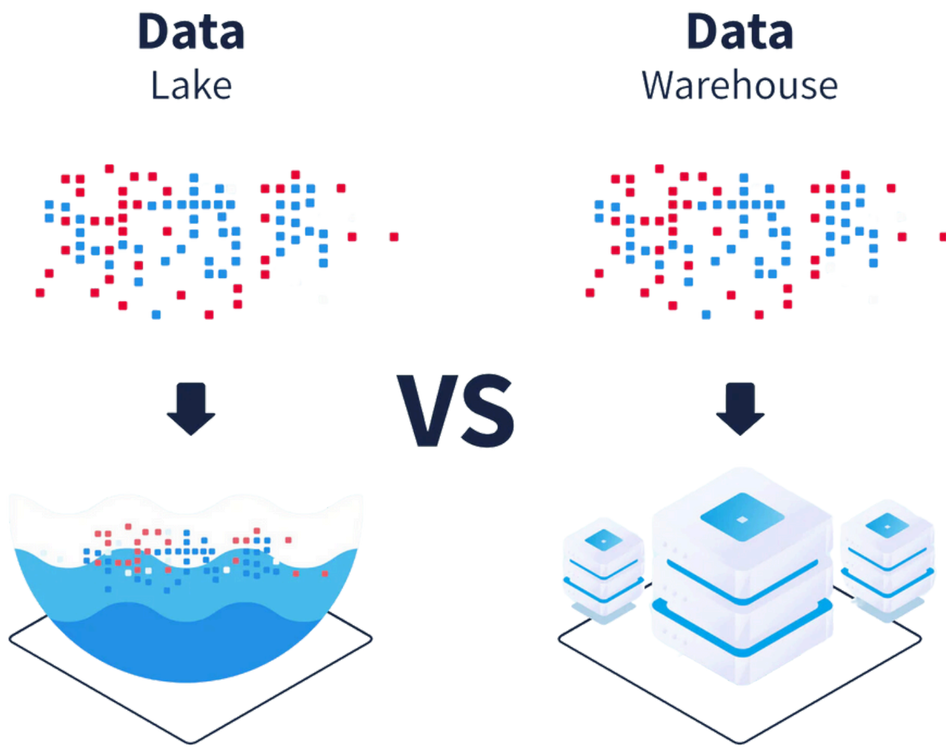


Defining Data Warehouses and Data Lakes

Both are storage tools, but:

- **Data Warehouse**: A centralized repository that stores structured integrated data from multiple sources, optimized for querying and analysis. Stores structured data, schema-on-write, optimized for read operations.
- **Data Lake**: Raw data, schema-on-read, optimized for storage and batch processing.

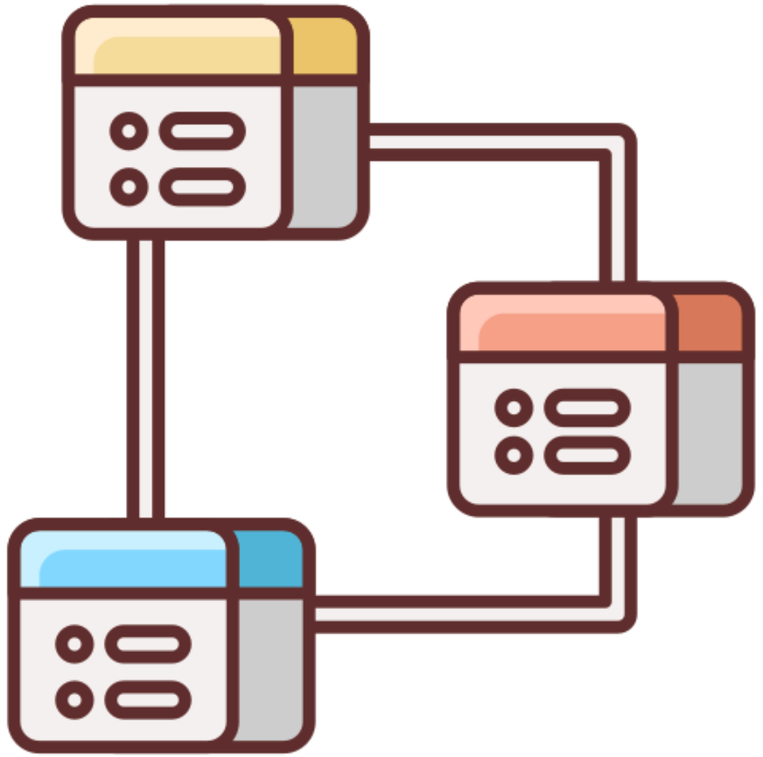
- **Discussion Point**: When would you choose a data lake over a data warehouse?



Data Modeling Concepts

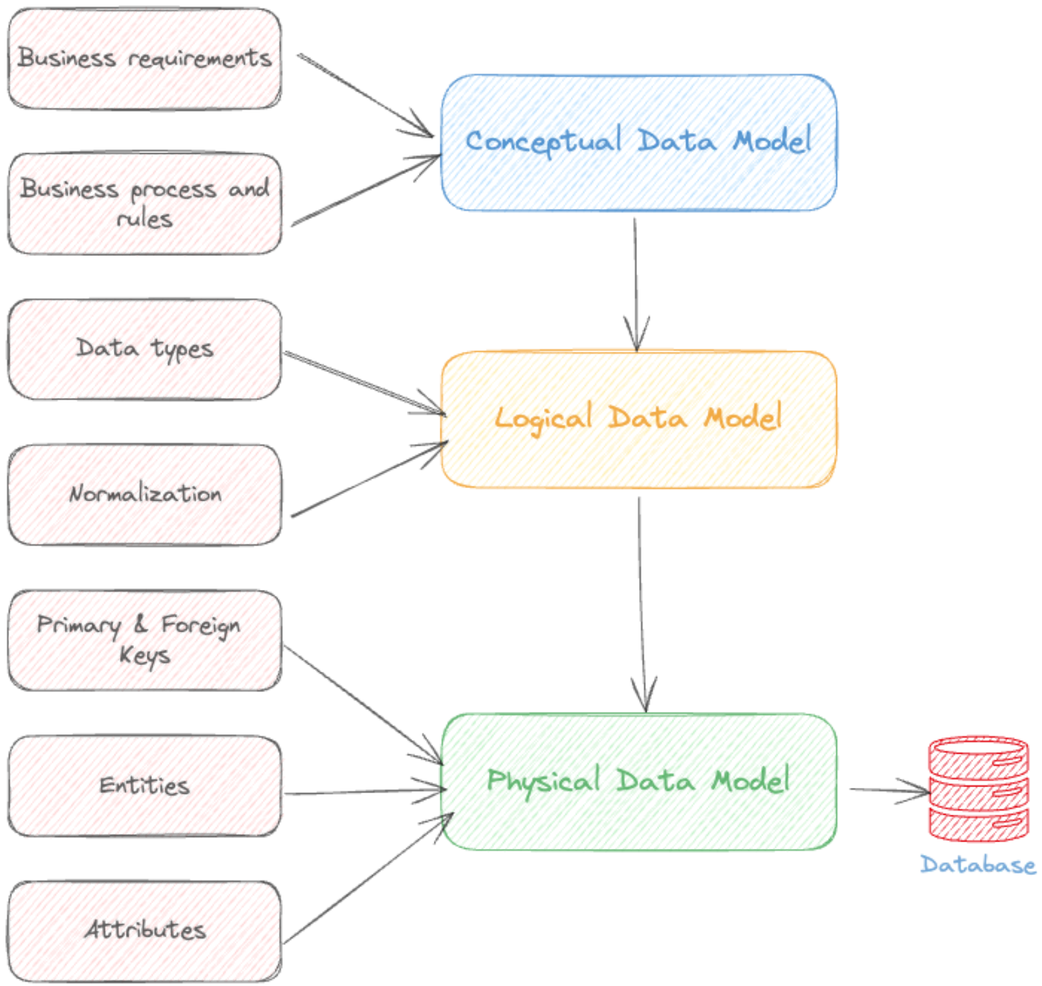
- **Definition**: Data Modeling is the process of creating a data model to visually represent the structure of data.

- **Discussion Point**: Why is data modeling important in database design?



Types of Data Models

- **Conceptual Data Model**: High-level overview of data entities and relationships.
- **Logical Data Model**: Detailed description of data entities, attributes, and relationships.
- **Physical Data Model**: Specification of data storage and access methods.

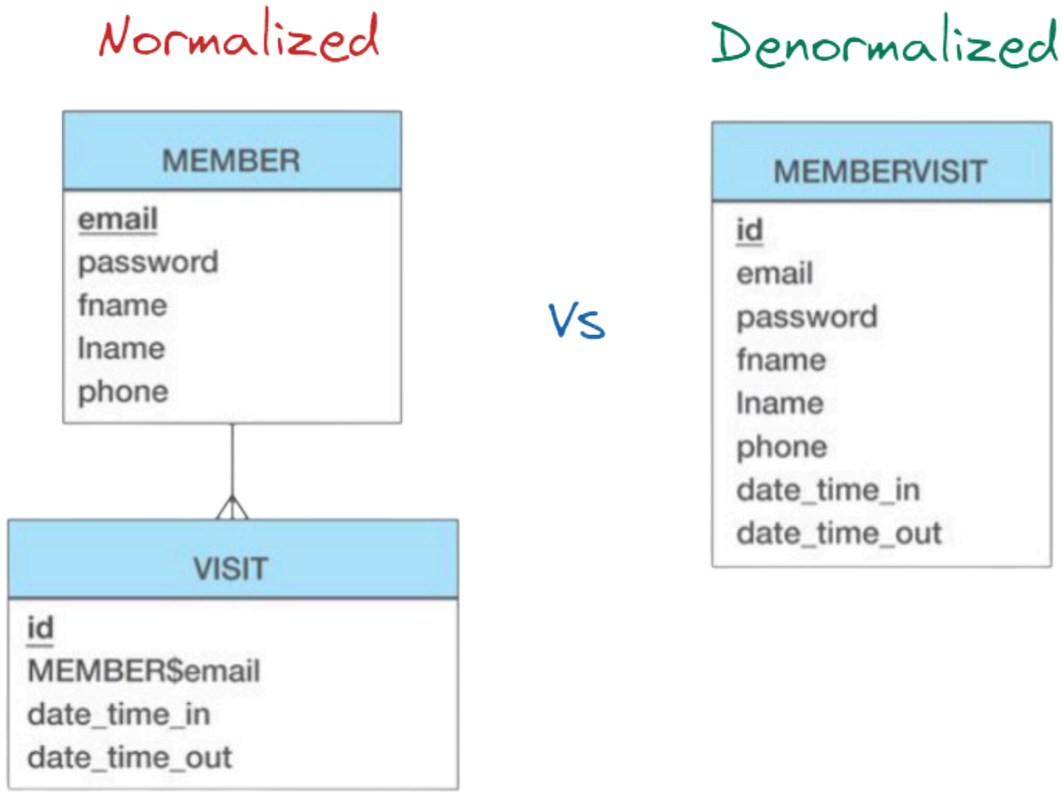


Tools for Data Modeling

- DBeaver
- Draw.io
- ER/Studio
- ERwin Data Modeler
- Microsoft Visio

Data Modeling Techniques

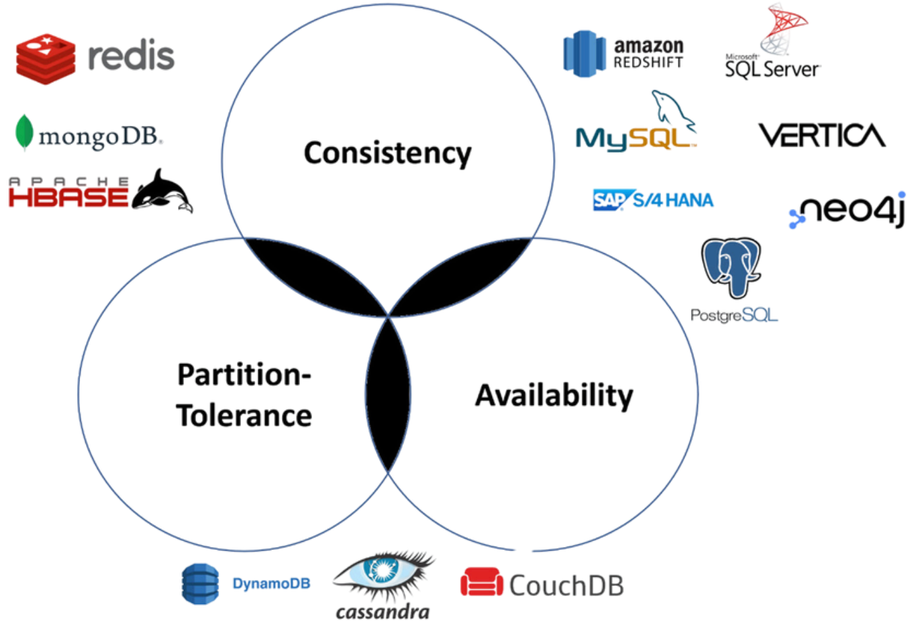
- **Denormalization**: Combining multiple tables to improve read performance
- **Normalization**: Splitting tables into Facts and Dimensions.
- Others:
  - Nested structures: For complex, hierarchical data
  - Polyglot persistence: Using multiple database types for different data needs
  - Wide-column design: For handling sparse data. names and formats of individual attributes can vary from row to row.



CAP Theorem

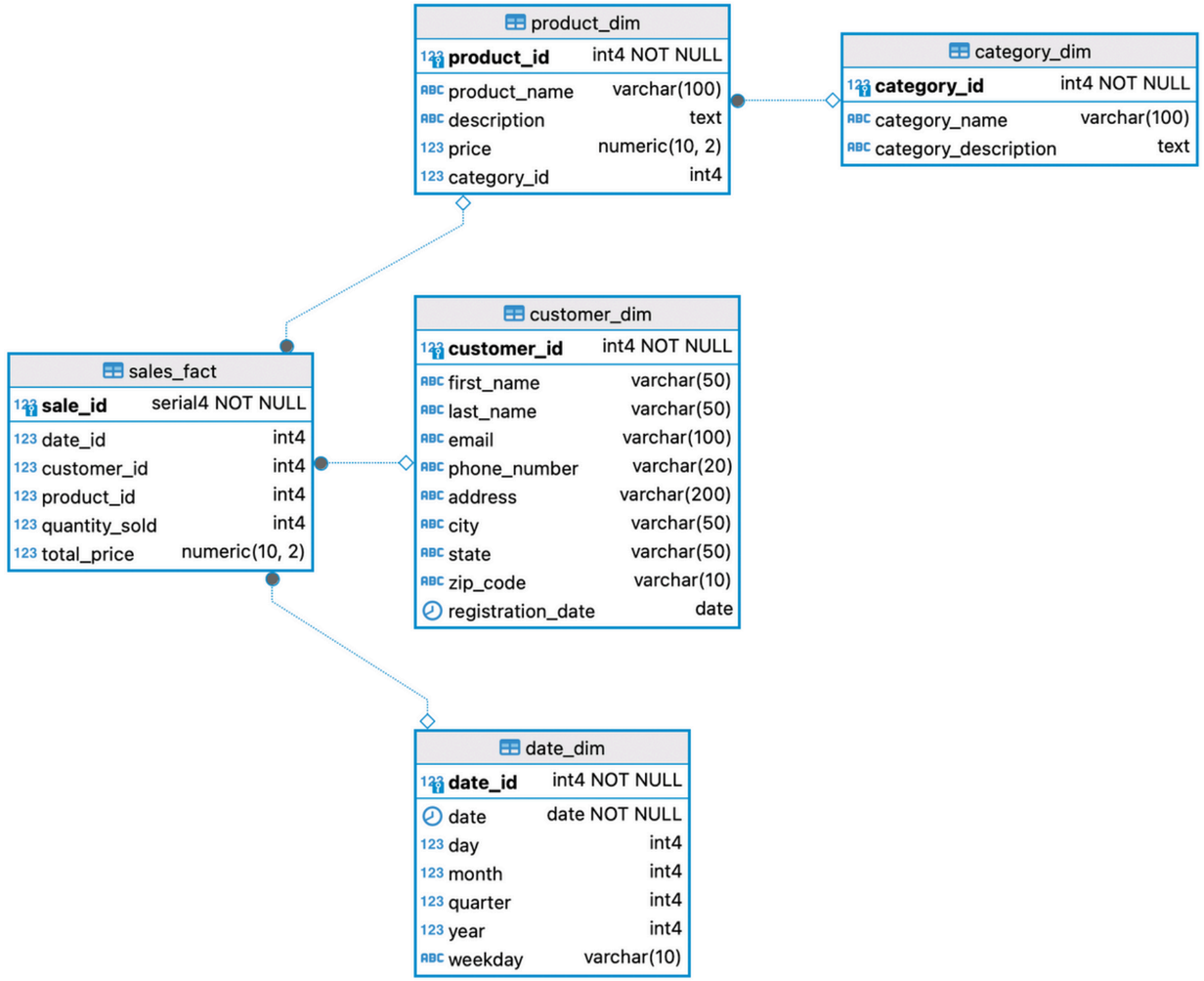
- **Definition**: A distributed database is a database in which storage devices are distributed across multiple nodes or servers, where they all work together

- **CAP Theorem** states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:
  - **Consistency**: all the nodes (databases) inside a network will have the same copies of a replicated data item visible for various transactions
  - **Availability**: Every request receives a response, without guarantee that it contains the most recent version of the information
  - **Partition tolerance**: The system continues to operate despite network partitions



Case Study: Designing a Data Model

- E-commerce business data model
- Identify entities, attributes, and relationships



SQL: Structured Query Language

- **Definition**: a standard language for accessing and manipulating databases.
- SQL stands for: **Structured Query Language**.
- **Query**: Is a question or inquiry about a set of data. We use SQL to retrieve meaningful and relevant information from databases.
- SQL lets you access and manipulate databases

What is SQL Used for?

- CRUD Operations: CREATE, READ, UPDATE, and DELETE records from databases.
- SQL can create new databases.
- SQL can create new tables in a database.
- and more...

Hands-on:

Designing a Data Model and Querying it

Q&A and Discussion

MEET OUR TEAM

Omar AlSaghier

Sr. Data Engineer

DATATECH LABS.

THANK YOU

OUR CONTACT

DataTechLabs

datatechlabs.ai@gmail.com

datatechlabs.ai

Amman, Jordan