

1. Database

Definition:

A **database** is a structured system used to store and manage data for day-to-day operations (also called **OLTP – Online Transaction Processing**).

Examples: MySQL, PostgreSQL, SQL Server, Oracle, MongoDB

Key Features:

- Stores **current operational data** like customers, orders, payments.
- Data is **highly structured** (tables with rows and columns).
- Ensures **ACID** properties (Atomicity, Consistency, Isolation, Durability).
- Supports **real-time reads and writes** for apps or websites.

Use Case:

- Banking transactions
- E-commerce order management
- Student management systems

2. Data Warehouse

Definition:

A **data warehouse (DW)** is a central repository used for **analytics and reporting**, not for real-time transactions.

It stores **historical and aggregated data** from multiple databases.

Examples: Amazon Redshift, Google BigQuery, Snowflake, Azure Synapse

Key Features:

- Data comes from multiple sources → **ETL (Extract, Transform, Load)** process.
- Data is **structured and cleaned** before loading.
- Optimized for **complex queries and analytics** (OLAP – Online Analytical Processing).
- Stores **historical data** for trend analysis and dashboards.

Use Case:

- Sales performance analysis
- Business intelligence (BI) dashboards
- Year-over-year revenue trend

3. Data Lake

Definition:

A **data lake** stores **raw data** (structured, semi-structured, and unstructured) from various sources — such as logs, videos, IoT data, or social media — in its **original format**.

Examples: Azure Data Lake Storage (ADLS), Amazon S3, Google Cloud Storage

Key Features:

- Handles **any type** of data: text, images, audio, CSV, JSON, etc.
- Stores **massive volumes** at low cost.
- Schema is applied **when reading** (Schema-on-Read) instead of when writing.
- Used in **machine learning (ML)**, **data science**, and **big data analytics**.

Use Case:

- Storing IoT sensor data
- Logs for AI/ML model training
- Social media analytics

4. Delta Lake

Definition:

A **Delta Lake** is an enhanced **data lake** built on top of storage like S3 or ADLS, adding **reliability**, **ACID transactions**, **versioning**, and **time travel**. It bridges the gap between **data lakes** (flexibility) and **data warehouses** (reliability).

Examples: Databricks Delta Lake (open-source format)

Key Features:

- **ACID transactions** on big data.
- **Schema enforcement** and **evolution**.
- **Time travel**: roll back or view old data versions.
- Compatible with **Apache Spark** for big data processing.
- Supports **batch and streaming** data together.

Use Case:

- Real-time data pipelines
- Large-scale analytics with consistent data
- Combining structured + unstructured data for ML and BI

Table

Feature	Database	Data Warehouse	Data Lake	Delta Lake
Purpose	Operational data storage	Analytical data storage	Store all types of raw data	Reliable, versioned data lake
Data Type	Structured	Structured	All (structured, semi, unstructured)	All (with ACID support)
Data Freshness	Real-time	Historical	Raw / near real-time	Both real-time & historical
Processing Type	OLTP	OLAP	Big Data	Batch + Streaming
Cost	Medium	High	Low	Moderate
Schema	Schema-on-Write	Schema-on-Write	Schema-on-Read	Schema-on-Read + Write

Feature	Database	Data Warehouse	Data Lake	Delta Lake
Best For	Apps, websites	BI, dashboards	AI/ML, data exploration	Unified data & analytics

Overview

- **Database:** For live apps — “run the business.”
- **Data Warehouse:** For analysis — “understand the business.”
- **Data Lake:** For storing everything — “explore the data.”
- **Delta Lake:** For reliable, versioned analytics — “trust the data.”

