***Database***

**Concept:**  
Stores current operational data used for day-to-day business transactions.

**Type:**  
OLTP (Online Transaction Processing)

**Characteristics:**

* Structured data only
* Supports real-time inserts, updates, and deletes
* Ensures data integrity and consistency (ACID)
* Used for frequent, small transactions
* Limited scalability (vertical scaling)

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

***Data Warehouse***

**Concept:**  
Centralized system for storing and analyzing large volumes of historical, structured data.

**Type:**  
OLAP (Online Analytical Processing)

**Characteristics:**

* Structured or semi-structured data
* Optimized for complex analytical queries
* Supports historical data analysis and reporting
* Periodically refreshed data
* Expensive due to compute and storage costs

**Examples:**  
Snowflake, Azure Synapse Analytics, Amazon Redshift, Google BigQuery.

***Data Lake***

**Concept:**  
A vast storage repository that holds raw data in its native format (structured, semi-structured, or unstructured).

**Type:**  
Big Data Storage

**Characteristics:**

* Stores data as-is (raw form)
* Can handle any data type (CSV, JSON, images, videos, logs, etc.)
* Low-cost storage (pay-per-use).
* Ideal for big data processing and machine learning.
* May lack schema enforcement and transactional integrity.

**Examples:**  
Azure Data Lake Storage (ADLS), Amazon S3, Hadoop HDFS.

***Delta Lake***

**Concept:**  
A layer built on top of a Data Lake that adds reliability and performance with ACID transactions and versioning.

**Type:**  
Transactional Data Lake (Hybrid OLAP + ACID)

**Characteristics:**

* Adds ACID transactions to Data Lakes.
* Supports schema enforcement and evolution.
* Enables time travel (data versioning).
* Handles batch and streaming data.
* Ensures reliable and high-performance analytics.

**Examples:**  
Databricks Delta Lake, Apache Hudi, Apache Iceberg.