**Comparison** between **Database**, **Data Warehouse**, **Data Lake**, and **Delta Lake**

| **Feature / Aspect** | **Database 🗂️** | **Data Warehouse 🏢** | **Data Lake 🌊** | **Delta Lake ⚡** |
| --- | --- | --- | --- | --- |
| **Purpose** | Stores **current operational data** used by applications | Stores **historical & aggregated data** for **analytics & reporting** | Stores **raw, unstructured, semi-structured, or structured data** at scale | Combines **Data Lake flexibility** with **Data Warehouse reliability** |
| **Data Type** | Mostly **structured** (e.g., tables) | Structured (cleaned, transformed data) | Structured, semi-structured (JSON, Parquet), unstructured (images, logs) | Structured & semi-structured (via Delta format on top of Data Lake) |
| **Data Processing** | **OLTP** (Online Transaction Processing) – frequent inserts/updates | **OLAP** (Online Analytical Processing) – complex queries on large data | Raw data stored; processing usually done externally using tools like Spark | Supports **both batch & streaming** processing with ACID transactions |
| **Schema** | **Schema-on-write** – predefined structure | **Schema-on-write** – data is transformed before loading | **Schema-on-read** – structure applied at query time | **Schema evolution & enforcement** supported (better governance) |
| **Latency** | Real-time / near real-time | Near real-time or scheduled batches | Data available once ingested, analysis depends on external tools | Low latency with support for real-time data updates |
| **Scalability** | Scales vertically (hardware upgrades) | Scales vertically, some modern systems scale horizontally | Highly **horizontally scalable**, designed for **big data** | Inherits scalability of underlying data lake with **transaction guarantees** |
| **Data Quality / Consistency** | High consistency (ACID transactions) | High consistency after ETL | Raw, may be inconsistent or duplicate | **ACID transactions**, time travel, versioning for reliability |
| **Use Cases** | App backends, user data, transactions, real-time apps | Business intelligence, reporting, dashboards, historical analysis | Storing massive amounts of varied data for data science, ML, IoT | Streaming + batch analytics, ML pipelines, incremental ETL |
| **Examples** | MySQL, PostgreSQL, Oracle, SQL Server | Snowflake, Amazon Redshift, Google BigQuery, Teradata | Hadoop HDFS, Amazon S3, Azure Data Lake, Google Cloud Storage | Databricks Delta Lake (on top of S3, ADLS, or HDFS) |
| **Cost** | Typically moderate | High (due to compute and ETL processes) | Lower storage cost, processing depends on external compute | Lower storage + better optimization, can reduce ETL cost |
| **Data Transformation** | Limited, mostly CRUD | Done during ETL before loading | Done at query time or using processing frameworks | Supports incremental upserts and merges natively |