Databases, Data Lakes, Data Warehouses

Name: Aathirainathan P

Date: 27-11-2024

Databases:

A **database** is a collection of data that is stored and organized in a way that allows for easy access, management, and updating. It is typically used to support **Online Transaction Processing (OLTP)** workloads, which involve real-time operations such as inserting, updating, and deleting data. Databases can store **structured** or **semi-structured** data, and depending on the type (e.g., relational or NoSQL), they may support flexible or fixed schemas. Relational databases like **MySQL** or **PostgreSQL** store data in tables with fixed columns and rows, while NoSQL databases like **MongoDB** or **Cassandra** store data in formats such as JSON or key-value pairs. Databases are designed for fast query performance and data integrity through **ACID** (Atomicity, Consistency, Isolation, Durability) transactions. They are best suited for applications requiring **real-time** data access and updates, such as managing user accounts, order processing, or inventory management.

Data Warehouses:

A **data warehouse** is a specialized database optimized for **Online Analytical Processing (OLAP)** workloads, which focus on analysis and reporting rather than transactional processing. Data warehouses store large amounts of **current** and **historical** data from various sources in a **predefined, fixed schema**. The data is usually **cleansed, transformed, and aggregated** through an **ETL (Extract, Transform, Load)** process, often on a scheduled basis (e.g., daily or hourly). Data warehouses are ideal for business intelligence (BI), generating reports, and performing data analysis because they enable fast querying of large datasets. Examples of data warehouses include **Amazon Redshift**, **Google BigQuery**, and **Snowflake**. Data warehouses excel at making structured data easily accessible for business analysts and data scientists, but they are not designed for real-time transaction processing. The data in a data warehouse is not always up-to-date due to the ETL process, which may introduce latency.

Data Lakes:

A **data lake** is a storage system designed to hold large volumes of **raw data** in its **native format**, without the need for immediate structuring or transformation. Data lakes support a wide variety of data types, including **structured**, **semi-structured**, and **unstructured** data, such as JSON, CSV, log files, audio, video, and images. Unlike databases and data warehouses, data lakes do not require a **predefined schema** and instead use **schema-on-read**, meaning the structure is applied only when the data is read for analysis. This makes data lakes highly flexible and capable of storing vast amounts of raw data at a low cost. They are commonly used for big data analysis, machine learning, and **predictive analytics**. Popular data lake technologies include **AWS S3**, **Azure Data Lake Storage**, and **Google Cloud Storage**. While data lakes can store data efficiently and at scale, they require significant processing and management to transform raw data into usable insights, often necessitating expertise from data scientists and engineers.

Difference between the three:

	Database	Data Lake	Data Warehouse
Workloads	Operational and transactional	Analytical	Analytical
Data Type	Structured or semi-structured	Structured, semi-structured, and/or unstructured	Structured and/or semi-structur
Schema Flexibility	Rigid or flexible schema depending on database type	No schema definition required for ingest (schema on read)	Pre-defined and fixed schema definition for ingest (schema on and read)
Data Freshness	Real time	May not be up-to-date based on frequency of ETL processes	May not be up-to-date based on frequency of ETL processes
Users	Application developers	Business analysts, application developers, and data scientists	Business analysts and data scien

Pros	Fast queries for storing and updating data	Easy data storage simplifies ingesting raw data A schema is applied afterwards to make working with the data easy for business analysts Separate storage and compute	The fixed schema makes working with the data easy for business analysts
Cons	May have limited analytics capabilities	Requires effort to organize and prepare data for use	Difficult to design and evolve sch Scaling compute may require unnecessary scaling of storage, because they are tightly coupled

When to Use Each:

- **Database**: For real-time transactional data in apps.
- **Data Warehouse**: When structured data from multiple sources needs analysis and reporting.
- **Data Lake**: For storing raw data of any type for future analytics, machine learning, and big data processing.