Data Architectures: Data Warehouse,

Data Lake and Data Lakehouse

Sanath Vijay Haritsa (11038004)  
*Applied Data Science and Analytics   
SRH Hochschule*Heidelberg, Germany  
sanath.vijayharitsa@stud.hochschule-heidelberg.de

*Abstract*— The diverse requirements of handling and analyzing vast amounts of data generated by organizations lead to the creation of *Data Warehouse* and *Data Lake*, which when used together in combination can store large volumes of raw data, including structured, semi-structured, and unstructured data. They provide high performance querying for big data and flexible, cost-effective scaling for high velocity data streams. However, this architecture results in creation of data silos where data is fragmented across different storage systems. Complex ETL processes cause increased operational overhead, latency in data availability and higher costs associated with data management. Ensuring consistent data governance, security, and compliance across both data lakes and warehouses can be incredibly challenging. A *Data Lakehouse* aims to solve these problems by providing a unified repository for all data types (structured, semi-structured, and unstructured). It combines data storage, processing, and management in one architecture and hence reduces the complexity of ETL processes. It provides a consistent framework for data governance, security, and compliance across all data. Data Lakehouse leverages low-cost storage solutions of data lakes while enhancing performance for analytical queries. It proposes a simplified architecture with better data accessibility and support for advanced analytics.

# Introduction

Data Warehouses are designed for the analysis of structured data, they are optimized for complex queries and reporting. Data warehouses integrate data from various sources into a unified schema (schema-on-write) and ensure data consistency and quality, essential for reliable business intelligence (BI) and decision-making. They store historical data for long-term analysis.

Data Lakes store large volumes of raw data, including structured, semi-structured, and unstructured data. They provide flexibility in data storage without the need to define a schema upfront (schema-on-read). They provide cost effective scaling for big data and serve as a repository for raw data to be used in advanced analytics and machine learning.

The emerging Data Lakehouse architecture aims to combine the best of both data warehouses and data lakes, by providing a unified platform for all data types and workloads. It leverages the low-cost storage of data lakes. It provides improved query performance with structured layers on top of raw data and enhanced data governance and management capabilities.

Setting up and maintaining a Data Lakehouse architecture can be complex. Ensuring consistent data governance and maintaining data quality, balancing performance for diverse workloads, managing costs effectively can be complex. Data Lakehouse require a diverse skill set to manage and operate.

# Related Work

## Data Marts

**Data Marts** are specialized subsets of data warehouses designed to serve the needs of specific business units or departments within an organization. They focus on a particular subject area or line of business, such as sales, finance, marketing, or HR.

## Operational Data Stores (ODS)

**Operational Data Stores (ODS)** are databases designed to integrate data from multiple sources and provide a consolidated view of current operational data. Unlike data warehouses, ODS are used for routine operational reporting and analysis, often in real-time or near-real-time.

# Technical Details

A **Data Warehouse** is a heterogeneous collection of different data sources organized under a unified schema. The essential components are:

## Data Sources

Different data sources can be connected to a data warehouse. This includes ETL systems, CRM systems, CSV files, and even other data warehouses.

## Staging Area

The staging area is a critical intermediate space where data is processed and prepared before being transferred to the warehouse. It serves as a buffer zone for cleansing, transforming, and consolidating data from various sources. An ETL tool is used to:

* **Extract** data from the sources.
* **Transform** the data into the standard format.
* **Load** the data into the warehouse.

## Storage Services

Storage services are at the core of a data warehouse. They are responsible for storage, partitioning, compression, replication, backup and recovery, life cycle management, and data integrity.

## Consumers

Various applications can consume data from a warehouse for different purposes like analytics, trend visualization, data mining, creating reports.

A **Data Lake** consists of the following crucial architectural components:

## A. Data Sources

Data sources can be broadly classified into three categories:

* Structured sources - SQL databases, RDBMS.
* Semi structured sources – HTML, JSON, XML.
* Unstructured sources – Text, video, audio streams, images, social media content.

*B. Data Ingestion*

Data ingestion is the process of importing data into the lake from various sources.

* Batch ingestion is a scheduled, interval-based method.
* Real-time ingestion immediately brings data into the data lake as it is generated.

*C. Data Storage and Processing*

* **Raw data store section**: Ingested data lands in the raw or landing zone. The data is in its native format. It acts as a repository where data is staged before any form of cleaning or transformation.
* **Transformation section**: Supports both batch and stream processing to clean, enrich, normalize and structure the data.
* **Processed data section**: The refined and conformed data is stored here. This is what analysts and data scientists interact with.

## D. Analytical Sandboxes

Analytical Sandboxes are isolated environments for data exploration, facilitating activities like discovery, machine learning, predictive modeling, and exploratory data analysis.

## Data Consumtion

The reliable data is ready for end users and is exposed via Business Intelligence tools.

## Security, governance, and monitoring

An overarching layer of governance, security, monitoring, and stewardship is typically implemented through a combination of configurations, third-party tools, and specialized teams.

A **Data Lakehouse** implements similar data structures and data management features to those found in a data warehouse but operating directly on the kind of low-cost storage used for data lakes.

## A. Lake first approach

It leverages the same architecture of a data lake to ingest structured, semi structured, and unstructured data.

## B. Making the lake reliable

* Leverages ACID transactions to ensure consistency.
* Supports star/snowflake schemas to provide robust governance and auditing mechanisms.

## C. Adding governance and security controls

* Support through Scala, Java, Python, and SQL APIs enables compliance with GDPR and CCPA.
* Data snapshots enable developers to access and revert to earlier versions of data.

*D. Support machine learning*

# Evaluation

1. Data Warehouse Evlauation

|  |  |  |
| --- | --- | --- |
| Key Features | Advantages | Disadvantages |
| Schema on write | Performance | Scalability |
| Performance and  consistency | Data Qiality | Flexibility |
| BI Integration | BI Integration | Cost |

1. Data Lake Evaluation

|  |  |  |
| --- | --- | --- |
| Key Features | Advantages | Disadvantages |
| Schema on read | Cost efficiency | Data Quality |
| Flexibility | Flexibility | Performance |
| Scalability | Scalability | Management |

1. Data Lakehouse Evaluation

|  |  |
| --- | --- |
| Key Features | Advantages |
| Unified storage | Unified platform |
| ACID transactions | Cost efficiency |
| Integrated analytics | Performance |
| Flexibility and  scalability | Simplified data  management |

Disadvantages:

### Complexity in implementation and maintenance

### Integration with various tools can be challenging

### Relatively new concept

# Conclusion

**Data Warehouses** are ideal for structured data, high-performance analytical queries, and business intelligence, but they can be expensive and less flexible.

**Data Lakes** offer flexibility and scalability for storing diverse data types but can suffer from data quality and performance issues if not properly managed.

**Data Lakehouses** provide a unified platform that combines the strengths of both data lakes and data warehouses, offering a balanced approach to modern data management needs. However, they come with their own set of complexities and require a skilled workforce to manage effectively.

Organizations should choose the architecture that best fits their specific needs, considering factors such as data types, use cases, performance requirements, and available expertise.

##### References

1. [Data warehouse architecture: A comprehensive guide (thoughtspot.com)](https://www.thoughtspot.com/data-trends/data-modeling/data-warehouse-architecture)
2. [Data Warehouse Architecture - GeeksforGeeks](https://www.geeksforgeeks.org/data-warehouse-architecture/)
3. [Data Lake Explained: Architecture and Examples (altexsoft.com)](https://www.altexsoft.com/blog/data-lake-architecture/)
4. [The Data Lakehouse Platform for Dummies | Databricks](https://www.databricks.com/resources/ebook/the-data-lakehouse-platform-for-dummies)
5. [What is a Data Lake? - Introduction to Data Lakes and Analytics - AWS (amazon.com)](https://aws.amazon.com/what-is/data-lake/)
6. Comparative evaluation of Data Warehouse, Lake and Lakehouse

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Data Warehouse | Data Lake | Data Lakehouse |
| Data Types | Structured | Any | Any |
| Schema | Schema on write | Schema on read | Both |
| Performance | High | Low to moderate | High |
| Scalability | Low | High | High |
| Cost | High | Low to moderate | Low to moderate |
| Flexibility | Low | High | High |
| Data Quality | High | Low to moderate | High |
| Use Cases | BI reporting,  historical data  analysis | Big data storage,  ML and analytics | Unified, real-time and historical  analysis |
| Complexity | Low to moderate | Low to moderate | High |