Chapter 10:

The Evolution of Big Data Architectures: From Traditional Data Management to Data Lakehouses

Hüseyin Can Barutcu
İstanbul Topkapı University

doi: https://doi.org/10.66414/mls.978-1-0666009-1-5_10

Abstract

The rapid growth in data volume, variety, and generation speed has led to a significant transformation in data management architectures. This study examines the evolution of Big Data architectures from traditional file-based systems and relational databases to Data Warehouses, Data Lakes, and Data Lakehouses. The main characteristics, data processing approaches, advantages, and limitations of these architectures are discussed comparatively. The study also examines cloud-based Big Data platforms, distributed data processing technologies such as Apache Spark and Apache Kafka, and the integration of Big Data with Business Intelligence and Artificial Intelligence applications. In addition, current challenges such as data quality, metadata management, data governance, and the risk of Data Lakes becoming Data Swamps are discussed. Finally, the study highlights the management of data structures required by Machine Learning applications and high-dimensional data as important research topics for the future development of Data Lake and Lakehouse architectures.

Keywords: big data, data lakehouse, data warehouse, data lake, data management

1. Introduction

The acceleration of digitalization processes, the diversification of data generation sources, and industrial transformations have created the need for a major change in approaches to data storage, processing, and analytical interpretation. In the early stages of information systems, traditional data management structures were based on file-based systems and Relational Database Management Systems (RDBMS). These systems stored data in structured forms under strict schemas and supported continuous daily operational processes (OLTP) with high consistency and Atomicity, Consistency, Isolation, Durability (ACID) guarantees. However, when organizations needed to integrate large volumes of data to support historical analysis and strategic decision-making, operational databases faced performance limitations when handling complex analytical queries (OLAP). This limitation led to the development of the Data Warehouse architecture, in which data is cleaned through ETL (Extract, Transform, Load) processes, collected in a centralized structure, and used to support Business Intelligence (BI) applications.

After the 2000s, especially with the widespread use of the Internet of Things (IoT), sensors, social media, and mobile systems, the era of "Big Data" began, being defined by the dimensions of Volume, Velocity, Variety, Veracity and Value (5Vs). Traditional Data Warehouses started to become insufficient because of their high costs, predefined schema requirements, and limitations in processing unstructured data. To overcome these structural limitations, Data Lakes emerged and enabled heterogeneous and raw data to be transferred at high speed to low-cost cloud object storage or distributed file systems (DFS) without any prior transformation, based on the schema-on-read principle. However, the lack of ACID transaction guarantees, schema management, and data governance in Data Lakes led over time to a decline in data quality, performance losses caused by the "small file problem," and the transformation of these systems into uncontrolled *"Data Swamps"*.

This methodological limitation in today's data ecosystem has led to the emergence of the Data Lakehouse architecture, which combines the ACID reliability, high SQL performance, and strong data governance of Data Warehouses with the flexible, scalable, and low-cost structure of Data Lakes. By using Open Table Formats such as Delta Lake, Apache Iceberg, and Apache Hudi, this new approach builds logical transaction logs on the object storage layer. Within the Medallion Architecture, it provides a gradual transition from raw data to processed data and supports both traditional Business Intelligence reporting and advanced analytics and machine learning processes on the same infrastructure. This study aims to examine the evolution of Big Data technologies, starting from traditional file-based and relational data management systems, as well as the technical characteristics and operational capabilities of Data Warehouse, Data Lake, and Data Lakehouse architectures and the transformation processes among these architectures from a comprehensive perspective.

2. Evolution of Big Data

2.1. Traditional Data Management

Data management includes all processes carried out to store, organize, access, and maintain the usability of data. In the early stages of information systems development, these processes were mainly performed through file-based systems. In the file-based approach, each application or user group stores the required data in its own files and performs operations on these files (Elmasri & Navathe, 2008). Although traditional file-based data management provides a practical approach, especially for storing and sharing limited amounts of data, various management problems arise as the amount of data and the number of users working with the data increase (Weidmann, 2023).

The relational data model, proposed by Codd in 1970, introduced an important transformation in data management by enabling data to be represented based on relationships. In the relational model, data is represented in tables consisting of rows and columns, and logical

relationships between data are established through common values. This approach contributed to reducing the dependency between the physical storage structure of data and the logical view of data presented to users (Codd, 1970).

With the widespread use of relational database management systems (RDBMS), functions such as data integrity, data security, concurrent access, and querying became centrally manageable. Relational databases structure data in tables consisting of rows and columns. The system maintains data integrity through primary key and foreign key relationships, while ACID properties support the reliable and consistent execution of transactions (Coronel & Morris, 2015; Silberschatz et al., 2020).

The limitations of traditional file-based data management approaches in handling large data volumes, concurrent data processing, and data integrity created the need to move toward Database Management Systems and modern Data Warehouse architectures.

Since relational databases are mainly used to support daily operational transactions, new requirements have emerged over time for analyzing large volumes of historical data and using them in decision-support processes (Kimball & Ross, 2002). RDBMS structures, which are designed to record daily transactions quickly, frequently, and securely, have shown significant performance limitations when handling long-term and complex analytical queries across an organization (Coronel & Morris, 2015; Elmasri & Navathe, 2008). These architectural limitations of operational databases led to the development of the Data Warehouse approach, which integrates data from different organizational systems into a central structure, provides historical depth, and is optimized for analytical queries (Inmon, 2002).

Another important issue in file-based data management is maintaining data quality. In file formats that lack strict validation and constraint rules at the schema level, violations of data types, value ranges, or logical dependencies may result in *"data pollution"* (Saha & Srivastava, 2014). For example, detecting and correcting errors such as entering a

textual value into a variable that should contain numerical values may require additional checks by users. This situation makes it difficult to maintain data integrity, especially in large datasets created by different users.

File-based data management systems have significant limitations in terms of concurrent access mechanisms. When multiple users work on the same data file at the same time, updates may conflict with each other, data may be overwritten, and uncontrolled file versions may be created over time (Silberschatz et al., 2020; Weidmann, 2023). This situation leads to important consistency and reliability problems, such as determining which file contains the most recent data, who made the changes, and which version of the data should be used in analytical processes (Elmasri & Navathe, 2008; Coronel & Morris, 2015).

The increasing volume of data and the growing variety of data sources have made the limitations of traditional relational databases and Data Warehouse architectures more visible, particularly in terms of scalability, data processing capacity, and data variety. The predefined schema-based structure of relational systems provides important advantages for managing structured data in an organized and consistent way. However, it may create various difficulties in managing unstructured or semi-structured data such as text, images, videos, sensor data, and log records. In addition, storing and processing large volumes of data in centralized systems requires more computing resources and may challenge the scalability limits of traditional systems (Gandomi & Haider, 2015; Stonebraker et al., 2010).

In the Data Warehouse approach, integrating data from different sources into a centralized structure by transforming it into a specific data model has been an important development for organizational reporting and decision-support processes. However, the rapid increase in data volume, the generation of data from different sources and in different formats, and the need to process data in real time or near real time in some use cases have created new requirements that traditional data management approaches alone cannot fully meet. This transformation, particularly

described by the dimensions of volume, velocity, and variety, has increased the need for methods such as distributed storage and parallel processing in data management (Gandomi & Haider, 2015; Laney, 2001) .

The limitations of file-based data management, such as data redundancy, inconsistency, concurrent access, and data integrity issues, increased the need to manage data in a more systematic structure. In response to this need, relational database management systems enabled data to be stored and queried within defined schemas and relationships and to be managed reliably by multiple users (Codd, 1970; Silberschatz et al., 2020). However, the increasing number of operational systems used in organizations and the growing amount of data generated by these systems created new requirements for integrating data from different sources and performing analytical queries on historical data. These requirements supported the development of centralized data structures for decision support and analytical purposes beyond operational data management and contributed to the growing importance of the Data Warehouse approach (Inmon, 2002; Kimball & Ross, 2002).

2.2. Data Warehouse Architecture and Key Features

The increasing variety of operational systems and data sources used in organizations has increased the need to integrate and analyze data generated by different systems. A Data Warehouse is a data management structure in which data obtained from different and heterogeneous sources is integrated, stored, and analyzed for decision-support purposes (Dhaouadi et al., 2022). Data Warehouses have developed as data management systems that support the integration of structured organizational data and its use for analytical purposes (Nambiar & Mundra, 2022). These systems enable data obtained from different sources to be used in decision-support and analytical processes, making them one of the main data infrastructures for Business Intelligence (BI) applications (Antunes et al., 2022; Nambiar & Mundra, 2022). In other words, the main purpose of a Data Warehouse is not to perform daily operations, but to

bring together data generated from different sources and support the analysis of an organization's past and current situation.

At this point, the main difference between operational systems and Data Warehouses can be explained through the concepts of OLTP (Online Transaction Processing) and OLAP (Online Analytical Processing). OLTP systems focus on performing daily operations such as creating orders, recording payments, or updating inventory, while OLAP systems support queries for analyzing and comparing data collected over different periods (Chaudhuri & Dayal, 1997). For example, recording a sales transaction made by a customer is an operational process, whereas comparing sales over the last five years by product, customer, or region is an analytical process. Therefore, Data Warehouses enable data generated in operational systems to be evaluated beyond the context of daily transactions, analyzed over a broader time period, and used for decision-support purposes (Inmon, 2002).

To use data from different operational systems in a Data Warehouse, the data must first be transformed into a common and consistent structure. This process is called ETL (Extract, Transform, Load) and is one of the main components of Data Warehouse architecture (Dhaouadi et al., 2022). In the Extract stage, data is collected from different source systems. In the Transform stage, cleaning, transformation, and integration processes are applied, while in the Load stage, the prepared data is transferred to the Data Warehouse. In this way, an integrated data structure is created, allowing data from different formats or systems to be used together for analytical purposes (Vassiliadis, 2009).

In a traditional Data Warehouse architecture, data flow generally consists of sequential layers, including data sources, the ETL process, the central Data Warehouse, and analytical applications. Data obtained from operational systems is processed through ETL and then integrated into the central Data Warehouse. The stored data is made available to users for analysis through OLAP, Business Intelligence (BI), and reporting tools (Chaudhuri & Dayal, 1997). Smaller data sets created from the organization-wide Data Warehouse to meet the needs of a specific

department or business area are called Data Marts (Inmon, 2002). This structure makes it possible to analyze organizational data distributed across different systems within a common analytical environment. The data flow and the main components of a traditional Data Warehouse architecture are illustrated in Figure 1.

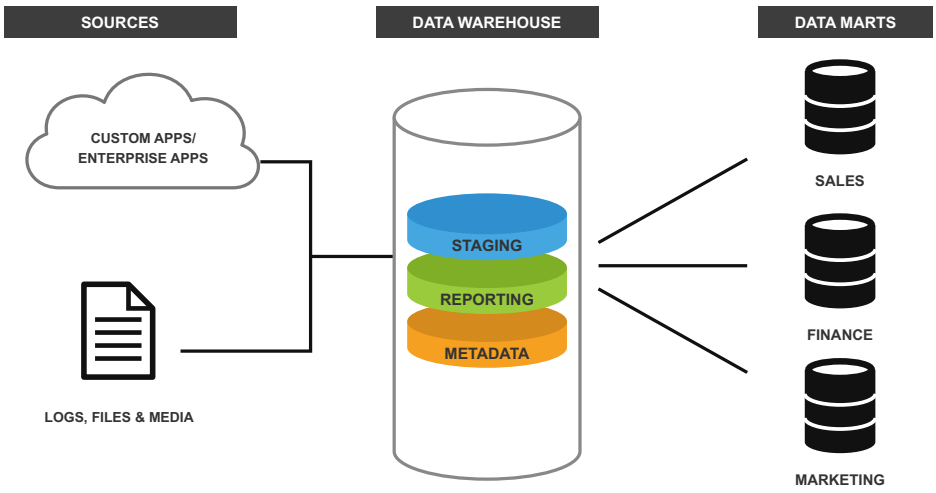


Figure 1. Data flow in traditional data warehouse architecture
Source: Created by the author

The main advantage of this architecture is that data from different sources can be integrated into a common structure and analyzed together with historical data. In this way, organizations can evaluate not only their current operational situation but also changes that occur over time, supporting data-driven decision-making processes. However, since traditional Data Warehouses mainly rely on predefined schemas and structured data, they have faced new requirements as data volume and variety have increased (Nambiar & Mundra, 2022).

In particular, the increasing use of semi-structured and unstructured data, such as text, log, and sensor data, has made the limitations of traditional Data Warehouses in managing different data types more visible. In addition, traditional ETL processes are generally performed at predefined time intervals, which may cause delays in transferring newly generated data from operational systems to analytical environments (de Assis Vilela et al., 2023). As data volume and variety increased and the need for faster analysis became more important, the

demand for more flexible and scalable data management approaches alongside the traditional Data Warehouse approach also increased. These requirements contributed to the later development of Big Data technologies and new data architectures such as Data Lakes.

2.3. Big Data and Data Lakes

Technological developments have also brought major industrial transformations. In particular, with the industry 4.0 movement, which gained momentum from the 2010s onward, the transformation of technological infrastructure became more visible (Nergiz & Barutcu, 2020). One of the most critical components of this transformation has been Big Data. As a result of technological advances and the trends created by these advances across different sectors, approaches to data and methods of processing data have changed significantly (Harby & Zulkernine, 2022). Especially after the 2000s, the increasing amount of data and efforts to analyze this data to obtain meaningful information led to the emergence of the concept of Big Data. In this context, the Big Data approach and related technologies have enabled an important transformation in the production, collection, and processing of data (Mayer-Schönberger & Cukier, 2013).

Considering the developments over the last two decades, it can be stated that this transformation has mainly resulted from the limitations of traditional databases in handling data types that require different processing methods, such as structured, semi-structured, and unstructured data, as well as the need to manage data arriving at different speeds (Hariri et al., 2019). Another critical factor is the limited physical hardware capacity for traditional data analytics processes such as ETL and the inability of established database systems to effectively support high-speed data ingestion (Armbrust et al., 2021). In this context, the concept of Big Data is defined through the 5V model, which addresses these technical needs and challenges: Volume refers to the large amount of data, Velocity represents the speed at which data is generated and processed, Variety refers to the presence of different data formats, Veracity represents the

reliability of data, and Value refers to the strategic benefit obtained from these processes (Demchenko et al., 2014).

The Big Data era has introduced new challenges for Data Warehouses. High-speed and complex data structures continuously generated by industrial facilities, smart production lines, and Internet of Things (IoT/IIoT) ecosystems have significantly limited the processing capacity of traditional Data Warehouse and ETL architectures (Harby & Zulkernine, 2022). As a result, real-time data processing has become necessary for industrial decision-support mechanisms and autonomous process control. However, it can be stated that even technological advances in hardware infrastructure have remained insufficient to enable traditional Data Warehouse systems, which are mainly designed to store filtered and structured data, to manage such dynamic data efficiently.

To overcome these limitations and structural constraints of traditional warehouse architectures, the Data Lake (DL) concept emerged over time. Data Lakes became one of the first flexible storage solutions of the Big Data era by allowing heterogeneous and hybrid unstructured data to be stored rapidly in its raw form without prior transformation, based on the schema-on-read approach. This solution is fundamentally based on an approach in which data is stored without first being converted into a fixed schema, and its structure (schema) is determined only when the data is accessed and analyzed (Janković et al., 2018). The Schema-on-Read approach and its role in big data processing are illustrated in Figure 2.

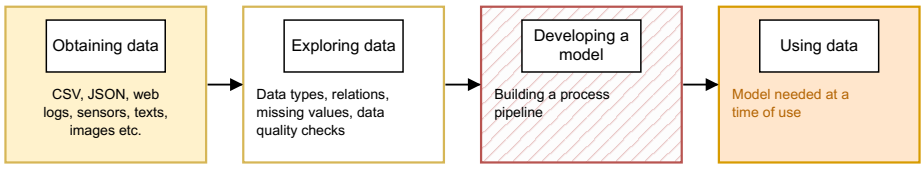


Figure 2. Schema on Read in Big Data Processing
Source: Adapted from Janković et al., 2018

This approach can be illustrated through a global coffee chain or a manufacturing facility. For example, heterogeneous data generated from the point-of-sale systems, mobile application, and IoT sensors in coffee machines of a global coffee chain can be transferred to a Data Lake in its

raw form without a predefined schema requirement. While a supply chain manager defines an analytical schema when accessing this data for inventory forecasting, a regional manager reads the same raw data through a different schema that focuses only on sales performance. Similarly, in a manufacturing facility, vibration, temperature, and camera data generated by machines can be stored in the Data Lake in their raw form. While a predictive maintenance specialist creates a schema suitable for time-series data, a quality control analyst analyzes the same raw data by defining a different schema that includes only visual features.

In both scenarios, heterogeneous raw data continuously generated from different sources and geographical locations is transferred through the network layer without any prior processing or transformation. The data is then written and stored in its raw form at high speed on the physical disk blocks of low-cost commodity hardware clusters operating in a distributed architecture or in highly scalable cloud object storage systems.

Data Lakes allow data from a wide variety of sources, such as OLTP systems, Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), cloud services, social media, and IoT sensor records, to be stored in their original raw formats without requiring any prior transformation or formatting (Janssen et al., 2024). The removal of complex and time-consuming ETL processes before data ingestion significantly reduces initial storage costs (Khine & Wang, 2018). In particular, distributed file system architectures such as Hadoop, which are based on low-cost infrastructure, are used to store data in its raw form for later processing (HDFS Architecture Guide, n.d.). In addition, the ability to query data using specific schemas according to analytical needs at the time of reading (schema-on-read) makes it easier for authorized users and data scientists within an organization to directly access data through different analytical tools for a wide range of use cases, including machine learning models (Azzabi et al., 2024; Walker & Alrehamy, 2015). The conceptual architecture of a Data Lake, including its data ingestion, storage, and analytical components, is illustrated in Figure 3.

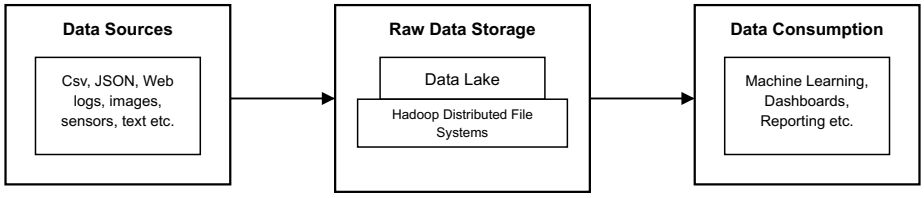


Figure 3. Data Lake Conceptual Architecture
Source: Created by the author

However, this high level of storage flexibility and ease of raw data ingestion also brings several methodological and operational challenges. Storing data without a predefined schema may lead to lower data quality, make it more difficult to track changing data structures, and create a continuous need for metadata extraction to transform the data. At the same time, millions of small files created in storage by continuous and fragmented data streams, such as IoT sensor data and logs, may cause the small file problem. This increases file system metadata overhead, slows down read/write (I/O) operations, and significantly reduces analytical query performance. Data Lakes that are not supported by effective metadata management, strong data governance, and security measures may result in delays in raw data processing, the accumulation of stale data, and inefficient use of resources. In such an uncontrolled environment, the reliability of data analytics processes may decrease, causing Data Lakes to lose their functionality and turn into "Data Swamps" (Gali et al., 2022; Janssen et al., 2024; Khine & Wang, 2018).

These limitations experienced in Data Lakes have become one of the main factors leading to the emergence of the Data Lakehouse architecture, which combines Atomicity, Consistency, Isolation, and Durability (ACID) reliability with the flexibility of Data Lakes within a single architecture.

2.4 From Data Lakes to Data Lakehouses

In today's data architectures, particularly in large organizations, traditional Data Warehouses and Data Lakes alone may be insufficient to meet the requirements of modern data analytics, machine learning, and real-time data processing due to the previously discussed limitations. This

limitation has led to the emergence of the Data Lakehouse architecture, which combines the strengths of both approaches within a single structure.

Open Table Formats form the technical foundation of the transition from Data Lake to Data Lakehouse architecture. Open Table Formats such as Delta Lake, Apache Iceberg, and Apache Hudi aim to provide relational database capabilities directly at the object storage level by building an ACID-compliant logical transaction log layer on top of cloud object storage (Parimi, 2025).

At the center of the transition architecture is the Medallion Architecture, in which data is processed gradually according to its quality. The data flow starts in the Bronze layer, where raw data is accepted in its original source format with schema-on-read flexibility. It then moves to the Silver layer, where data is cleaned, data types are validated, and schema enforcement rules are applied. Finally, the data is transferred to the Gold layer, where it is aggregated into star schemas for Business Intelligence and advanced analytical models, supporting end-to-end data quality (Schneider et al., 2024). Figure 4 presents this layered architecture, highlighting how data is progressively transformed from raw inputs into refined datasets suitable for analytical applications.

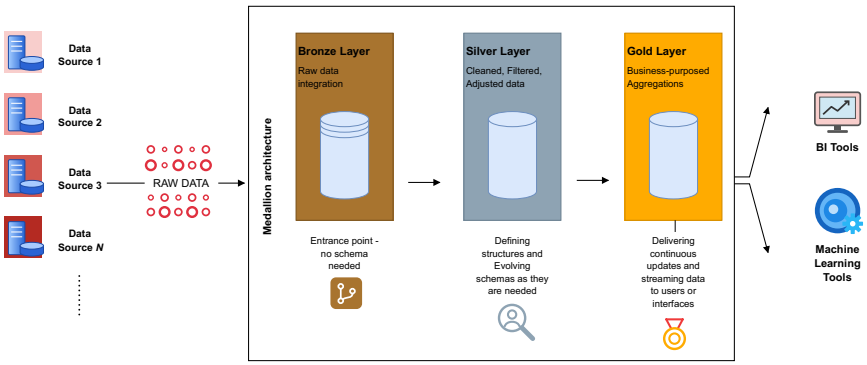


Figure 4. Lakehouse and Medallion Architecture
Source: Created by the author

In the implementation process, the transformation begins by handling text-based raw data that initially arrives in the Data Lake, such as CSV, JSON, logs, and sensor data, together with data that has already been

converted into columnar formats such as Parquet and ORC (Optimized Row Columnar), which provide high compression rates to improve analytical performance and reduce storage requirements (Armbrust et al., 2021). Unlike traditional text files, these columnar structured formats store data column by column rather than row by row. This allows only the relevant columns to be read from very large datasets and significantly reduces analytical query times.

Data already stored in the system in Parquet or ORC format can be quickly converted to next-generation table formats, such as Delta Lake and Apache Iceberg, by updating only the metadata layer through in-place conversion, without the cost of copying the data from the beginning. Raw streaming data, such as CSV and JSON, is directly ingested into the raw (Bronze) layer of the architecture and connected to automated transformation pipelines through ACID-supported table formats.

After this integration is completed, the *"small file problem",* one of the main challenges of Data Lakes, is addressed by combining fragmented files into appropriate sizes through compaction/optimization. By physically organizing data according to frequently queried columns using Z-Ordering or multi-dimensional clustering, the system can directly skip unrelated data blocks during query execution through data skipping, which significantly improves read performance. In addition, the storage overhead created over time by the transaction history, which records each change, and the time travel feature, which provides access to previous versions, is managed by periodically removing old versions through vacuuming (Databricks, n.d.; Delta Lake, n.d.).

As a result, the evolution from Data Warehouse to Data Lake and, more recently, to Data Lakehouse is not simply a change in storage methods or file formats. This transformation represents a comprehensive architectural development that provides end-to-end data quality across both raw and processed data flows, row- and column-level access security, flexible schema changes (schema evolution), and control over the source and movement of data (data lineage). The main characteristics, key capabilities, and operational differences of these three architectural

approaches are presented in detail in the comparison table (Table 1), which summarizes the main points discussed in this section. The main characteristics, key capabilities, and operational differences of these four architectural approaches are summarized in Table 1, highlighting the key distinctions discussed in this section.

Table 1. Comparison of all architectures

	Traditional Data Management	**Data Warehouse**	**Data Lake**	**Data Lakehouse**
Main Purpose	Managing daily operational transactions	Reporting and data analysis	Storing large volumes of raw data	Integrating data management, BI, and advanced analytics
Schema Approach	Schema-on-Write	Schema-on-Write	Schema-on-Read	Schema-on-Read and Schema-on-Write
Data Processing	Database operations	ETL/ELT and analytical processing (OLAP)	Data is stored first and processed later	Step-by-step processing with Medallion Architecture (Bronze–Silver–Gold)
Data Reliability	High	High	Can be difficult to manage	Supports ACID transactions
Performance	Effective for operational transactions	Suitable for analytical queries	Depends on processing engines, data formats, and optimization techniques	Can be optimized for analytical queries
Storage	Databases and file systems	On-premises, cloud, or distributed data warehouse storage	Distributed or cloud storage	Commonly cloud object storage with managed table formats
Cost	Increases as data volume grows	Relatively high	Relatively low	More flexible in terms of storage costs

3. Big Data Applications and Technologies

In today's digital ecosystem, the effective storage, processing, and analysis of rapidly growing data volumes have caused traditional database systems and hardware infrastructures to face various limitations in terms

of capacity and scalability. In this context, Big Data Management (BDM) approaches are becoming increasingly integrated with cloud computing infrastructures that support scalable and flexible resource use (Qi et al., 2023). Cloud computing platforms provide infrastructure, platform, and analytical services through service-based models, allowing organizations to respond dynamically to changing data and processing requirements. With these features, cloud-based Big Data solutions have become an important technological infrastructure in both academic and industrial applications for managing data-intensive processes and supporting data-driven decision-making mechanisms (Berisha et al., 2022).

3.1. Cloud-Based Big Data Platforms

Cloud-based Big Data platforms provide scalable storage and parallel processing capabilities for businesses, healthcare organizations, smart cities, and educational infrastructures. The increasing use of Big Data analytics in cloud environments has also increased the adoption of the ELT approach alongside the traditional ETL approach (Berisha et al., 2022).

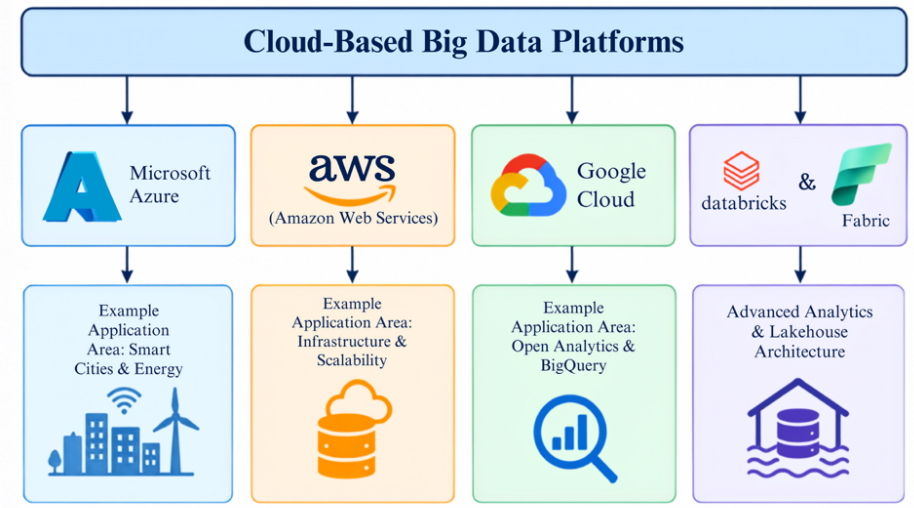


Figure 5. Cloud-base big data platforms
Source: Created by the author

As shown in Figure 5, today's cloud-based Big Data ecosystem includes large-scale cloud service providers such as Microsoft Azure,

Amazon Web Services (AWS), and Google Cloud, as well as integrated data and analytics platforms such as Databricks and Microsoft Fabric. These platforms support scalable data storage, distributed data processing, and analytical services in cloud environments. The ability to scale cloud-based resources according to changing requirements allows organizations to manage Big Data workloads of different volumes and characteristics more flexibly (Berisha et al., 2022; Qi et al., 2023). BigQuery, which is part of the Google Cloud ecosystem, is one of the cloud-based Data Warehouse solutions that enables large-scale datasets to be analyzed through a serverless architecture (Berisha et al., 2022).

The development of cloud-based data platforms has transformed architectures not only for storing Big Data but also for processing and analyzing data from different sources in an integrated way. In particular, the Data Lake approach allows structured, semi-structured, and unstructured data to be stored in their original formats. This structure reduces the need to transform data into a predefined schema in advance and makes it easier to meet different analytical requirements using the same data source (Berisha et al., 2022).

One important result of this transformation is the emergence of the Lakehouse architecture, which combines the strengths of Data Lake and Data Warehouse approaches. The Lakehouse architecture aims to combine the flexible and scalable storage structure of Data Lakes with the data management and analytical capabilities of Data Warehouses. In this way, Business Intelligence applications, data science processes, and machine learning workloads can be performed on a common data infrastructure (Armbrust et al., 2021). While Databricks is one of the leading platforms in the development and adoption of this architecture, Microsoft Fabric is also a current platform that brings together data engineering, data science, real-time analytics, and Business Intelligence processes within an integrated analytics environment.

This integrated structure provided by cloud-based Big Data platforms allows organizations to manage large volumes of data from different sources more flexibly and to scale analytical processes according

to changing requirements. However, the effectiveness of Big Data infrastructure depends not only on where the data is stored but also on how quickly and efficiently it is processed using different technologies. Therefore, distributed and real-time data processing technologies such as Apache Spark and Apache Kafka represent another important component of the modern big data ecosystem.

3.2. Big Data Processing Technologies

Traditional batch processing methods and disk-based architectures such as MapReduce face various limitations, particularly in Big Data applications that require low latency and real-time analytics. For this reason, in-memory computing, distributed computing, and stream processing approaches have become increasingly important in modern Big Data systems. Apache Spark is one of the main technologies in this transformation and enables large datasets to be processed in parallel and distributed across computer clusters (Belcastro et al., 2022). Spark's in-memory processing capabilities can provide important advantages over the disk-based MapReduce approach, especially for iterative and interactive analytical workloads.

Resilient Distributed Dataset (RDD), one of Spark's main data abstractions, supports the distributed processing of data across cluster nodes in a parallel and fault-tolerant manner. Within the Spark ecosystem, Spark SQL enables structured data to be processed through SQL queries, Datasets, and DataFrames, while MLlib supports the scalable application of machine learning methods such as classification, regression, and clustering on large datasets. Spark Streaming enables the same ecosystem to process streaming data (Belcastro et al., 2022).

Real-time stream processing technologies are especially important for analyzing continuously generated data such as IoT sensor data, system logs, financial transactions, and social media streams. In these architectures, Apache Kafka is a distributed messaging and event-streaming platform that supports the collection of high-volume data streams from different sources and their transfer to distributed systems.

Data transferred through Kafka can be processed by distributed processing engines such as Spark Structured Streaming or Apache Flink and included in real-time analytical processes (Lalaoui et al., 2025).

The integrated use of these technologies enables data collection, transmission, distributed processing, and analytics to be carried out within an end-to-end data pipeline. In this way, high-volume data streams can be processed with low latency and used in time-sensitive applications such as real-time analytics and decision support (Isah et al., 2019). However, the ability of Big Data to create value for organizations depends not only on processing data quickly and in a scalable way but also on transforming the results into understandable information and using them in decision-making processes. At this point, Business Intelligence systems play an important role in presenting analytical results generated by Big Data infrastructures to decision-makers through dashboards, reporting tools, real-time decision-support systems, and enterprise analytics applications.

3.3. Big Data and Business Intelligence

The development of Big Data technologies has enabled organizations not only to store large volumes and various types of data obtained from different sources but also to analyze this data and use it in decision-making processes. In this context, Business Intelligence (BI) includes the methods, processes, and technologies used to analyze data and transform it into meaningful and useful information for managers and decision-makers. The integration of Big Data and Business Intelligence supports moving beyond traditional reporting approaches and enables more comprehensive analytical processes (Phillips-Wren et al., 2021; Wang et al., 2022).

One of the important components of Business Intelligence applications is dashboards. Dashboards are Business Intelligence tools that support the monitoring of Key Performance Indicators (KPIs), trends, and analytical results through charts and tables. Reporting systems contribute to presenting operational and managerial data in a form that can be used in decision-making processes (Martins et al., 2022).

The rapid data processing capabilities provided by Big Data infrastructures have contributed to the transformation of Business Intelligence applications from systems based only on the analysis of historical data into more dynamic systems. In particular, transferring continuously generated data to analytical systems enables the development of real-time decision support processes. Through these systems, organizations can monitor operational changes more quickly, identify unusual situations, and use up-to-date analytical results in decision-making processes. This development has contributed to the expansion of Business Intelligence beyond the reporting of historical data to include analytical applications that support decision-making processes using more current data (Phillips-Wren et al., 2021; Wang et al., 2022).

Another result of this development is the increasing use of enterprise analytics applications. Enterprise analytics refers to the integrated analysis of different organizational data sources and their use in performance evaluation and decision-support processes (Martins et al., 2022; Wang et al., 2022).

However, the increasing volume and variety of data have made it necessary for organizations not only to analyze current and historical situations but also to predict future situations and identify complex data patterns. This need has accelerated the integration of Big Data analytics with machine learning and artificial intelligence methods. As a result, the Big Data ecosystem is expanding from reporting and visualization toward predictive analytics, automated decision-making, and generative artificial intelligence applications. This transformation makes it necessary to examine the relationship between Big Data and artificial intelligence.

3.4. Big Data and Artificial Intelligence

Big Data and Artificial Intelligence (AI) are considered two key technology areas that complement each other in today's data-driven systems. Large volumes of diverse and continuously generated data provide an important resource for developing and training AI models, while AI methods make it possible to identify complex relationships and

patterns within these datasets. In this context, machine learning is used for analytical tasks such as classification, clustering, prediction, and anomaly detection, while Deep Learning (DL) is particularly important for analyzing high-dimensional and unstructured data such as text, images, and audio. The integration of Big Data processing infrastructures with AI methods supports the scalable application of these models to larger datasets (Rahmani et al., 2021).

One of the important application areas where Big Data and Artificial Intelligence are integrated is predictive analytics. Predictive analytics aims to predict future events, trends, or behaviors based on patterns obtained from historical and current data. This approach can be used in data-intensive applications such as demand forecasting, risk assessment, customer behavior prediction, anomaly and fraud detection, and predictive maintenance. Therefore, AI-supported Big Data analytics goes beyond traditional descriptive analytics by providing a more advanced analytical structure that enables predictions to be generated from data and supports decision-making processes (Rahmani et al., 2021).

In recent years, the development of generative artificial intelligence and large language models has expanded the use of Big Data beyond prediction and classification toward the generation of new content and information (Banh & Strobel, 2023). One important application of this development is the retrieval-augmented generation approach, which supports generative models with relevant content obtained from external information sources. In a RAG architecture, information related to a user query is first obtained through retrieval from organizational documents, databases, or other information sources and is then provided to the generative model as context. In this way, connections can be established between organizational information sources within Big Data ecosystems and Generative AI models, supporting more context-aware information retrieval and generation (Gao et al., 2024).

4. Current State and Future Trends

Data Lakes provide flexibility for storing large-scale data with different structures and formats; however, they also involve various technical and operational limitations during implementation and operation. In particular, storing unstructured raw data without sufficient prior validation may make it more difficult to ensure data consistency and accuracy and may cause data quality problems (Laurent et al., 2020). In addition, the development and implementation of data governance policies require additional time and cost. The absence of standardized schemas or data models may make it more difficult to integrate data from heterogeneous sources, while providing access to large user groups raises important considerations regarding data security and privacy (Marjani et al., 2017).

One of these challenges is the risk of Data Lakes turning into a *"Data Swamp"* over time. Insufficient information about data types, data quality, and data lineage may make stored data more difficult to understand and use (Azzabi et al., 2024). In this context, metadata management is considered an important component. Associating metadata with data during data ingestion is important for identifying and tracking data and applying access policies. However, effective metadata management may require the integration of the Data Lake with existing metadata tools and the tracking of data transformations performed at different stages (Marz & Warren, 2015). At the organizational level, integrating different databases, systems, and data silos, automating processes, and applying governance standards is a process that requires technical expertise (Azzabi et al., 2024).

In addition to these technical and management challenges in Data Lakes, recent studies also examine the relationship between data management and Artificial Intelligence and Machine Learning applications. In this context, improving existing Data Lake functions and supporting Machine Learning applications in these environments are considered among the topics that can be explored in future research. While examining the existing functions and open research problems of

Data Lake systems, Hai et al. (2023) identify supporting Machine Learning in Data Lakes as one of the areas for future research. In parallel, there are also studies focusing on the storage of high-dimensional data structures used in Artificial Intelligence and Machine Learning applications. Bao et al. (2024) propose the Delta Tensor approach for more efficient storage of vector and tensor data in a Delta Lake-based Lakehouse architecture and show through experiments that improvements in time and storage space can be achieved compared with traditional tensor serialization methods. These studies indicate that, in addition to existing challenges related to data quality, governance, metadata management, and integration, the management of data structures required by Machine Learning applications is also among the current research topics for Data Lakes and related data architectures.

As a result, in addition to existing issues such as data governance, metadata management, and data quality, the management of data structures required by Machine Learning applications is also among the topics that can be addressed in future research on Data Lakes and Lakehouse architectures.

5. Conclusion

The evolution of big data architectures reflects the growing demand for scalable, flexible, and integrated data management approaches capable of accommodating diverse data types and analytical requirements. The transition from traditional data warehouses to data lakes and subsequently to Data Lakehouse architectures represents a significant development in managing large-scale and heterogeneous datasets. By combining the structured data management capabilities of data warehouses with the flexibility of data lakes, the Data Lakehouse approach provides a unified architectural framework for supporting different data formats and analytical workloads (Armbrust et al., 2021; Harby & Zulkernine, 2022).

The integration of distributed data processing technologies into modern data architectures represents another important advancement in managing and analyzing complex datasets. In particular, scalable

processing frameworks support large-scale data analytics by addressing increasing data volumes and computational demands (Belcastro et al., 2022). Within this context, Data Lakehouse architectures offer opportunities to integrate data storage, processing, and analytical capabilities within a common environment. However, their successful implementation requires careful consideration of system design, technological compatibility, and infrastructure requirements (Schneider et al., 2024).

Despite these developments, several challenges remain, particularly in relation to data governance, interoperability, architectural complexity, and the management of heterogeneous data environments. Ensuring data security, maintaining data integrity, and integrating diverse data sources within a unified infrastructure continue to be important considerations for the sustainability of modern data architectures (Janssen et al., 2024). Accordingly, future research should further investigate the applicability of Data Lakehouse architectures across different operational environments, with particular emphasis on comparative evaluations of scalability, processing efficiency, data reliability, and organizational requirements.

Overall, the transition toward Data Lakehouse architectures represents an important stage in the ongoing evolution of big data management technologies. Rather than completely replacing traditional data management approaches, these architectures provide an alternative model that integrates the strengths of existing systems with contemporary analytical requirements. The integration of data storage and processing capabilities within a unified architectural framework offers opportunities for developing more flexible, reliable, and interconnected data ecosystems. In this regard, the continued development of Data Lakehouse architectures is expected to play an important role in supporting increasingly complex data-driven applications and shaping the future of big data infrastructure (Armbrust et al., 2021; Schneider et al., 2024).

Declarations

Ethics Committee Approval: This chapter does not fall within the scope of research requiring ethics committee approval, such as studies involving the collection of data from participants through surveys, interviews, focus groups, observations, experiments, or other qualitative and quantitative approaches; the experimental or scientific use of humans or animals (including materials/data); clinical studies conducted on humans; studies conducted on animals; or retrospective studies subject to personal data protection legislation. Therefore, ethics committee approval was not deemed necessary.

Use of Artificial Intelligence: In this chapter, ChatGPT (OpenAI) was used for language review in selected sections. The ChatGPT–Canva integration was also used during the preparation of Figure 5, with subsequent visual editing and final adjustments carried out by the author in Canva. The author retains full responsibility for the content of the chapter.

Conflict of Interest: The author declares that there is no conflict of interest.

Funding: This study and the doctoral dissertation on which it is based did not receive support from any institution, project, fund, or other funding source.

Third-Party Materials and Permissions: Third-party materials used in this study were obtained from publicly accessible sources and appropriately cited. No separate permissions were obtained.

References

Antunes, A. L., Cardoso, E., & Barateiro, J. (2022). Incorporation of ontologies in data warehouse/business intelligence systems - A systematic literature review. *International Journal of Information Management Data Insights, 2*(2), 100131. https://doi.org/10.1016/j.jjimei.2022.100131

Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics. In *11th Annual Conference on Innovative Data Systems Research (CIDR '21)*. https://vldb.org/cidrdb/2021/lakehouse-a-new-generation-of-open-platforms-that-unify-data-warehousing-and-advanced-analytics.html

Azzabi, S., Alfughi, Z., & Ouda, A. (2024). Data lakes: A survey of concepts and architectures. *Computers, 13*(7), 183. https://doi.org/10.3390/computers13070183

Banh, L., & Strobel, G. (2023). Generative artificial intelligence. *Electronic Markets*, *33*(1), 63. https://doi.org/10.1007/s12525-023-00680-1

Bao, Z., Liu, L.-L., Wu, Z., Zhou, Y., Fan, D., Aibin, M., Coady, Y., & Brownsword, A. (2024). *Delta Tensor: Efficient vector and tensor storage in Delta Lake* (arXiv:2405.03708). arXiv. https://doi.org/10.48550/arXiv.2405.03708

Belcastro, L., Cantini, R., Marozzo, F., Orsino, A., Talia, D., & Trunfio, P. (2022). Programming big data analysis: Principles and solutions. *Journal of Big Data*, *9*(1), 4. https://doi.org/10.1186/s40537-021-00555-2

Berisha, B., Mëziu, E., & Shabani, I. (2022). Big data analytics in cloud computing: An overview. *Journal of Cloud Computing, 11*(1), 24. https://doi.org/10.1186/s13677-022-00301-w

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record, 26*(1), 65-74. https://doi.org/10.1145/248603.248616

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, 13*(6), 377-387. https://doi.org/10.1145/362384.362685

Coronel, C., & Morris, S. (2015). *Database systems: Design, implementation, & management* (11th ed.). Cengage Learning.

Databricks. (n.d.). *VACUUM*. https://docs.databricks.com/aws/en/sql/language-manual/delta-vacuum

de Assis Vilela, F., Times, V. C., de Campos Bernardi, A. C., de Paula Freitas, A., & Ciferri, R. R. (2023). A non-intrusive and reactive architecture to support real-time ETL processes in data warehousing environments. *Heliyon*, *9*(5), e15728. https://doi.org/10.1016/j.heliyon.2023.e15728

Delta Lake. (n.d.). *Optimizations*. Delta Lake documentation. https://docs.delta.io/optimizations-oss/

Demchenko, Y., de Laat, C., & Membrey, P. (2014). Defining architecture components of the big data ecosystem. In *2014 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 104-112). IEEE. https://doi.org/10.1109/CTS.2014.6867550

Dhaouadi, A., Bousselmi, K., Gammoudi, M. M., Monnet, S., & Hammoudi, S. (2022). Data warehousing process modeling from classical approaches to new trends: Main features and comparisons. *Data, 7*(8), 113. https://doi.org/10.3390/data7080113

Elmasri, R., & Navathe, S. B. (2008). *Fundamentals of database systems*. Pearson Education.

Gali, S., Bangarusamy, S., & Mahmoud, M. (2022, July). *The small files issue in big data platforms, problem and solutions* [Paper presentation]. 2022 World Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, United States. https://www.researchgate.net/publication/362278375_The_Small_Files_Issue_in_Big_Data_Platforms_Problem_and_Solutions

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management, 35*(2), 137-144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). *Retrieval-augmented generation for large language models: A survey* (arXiv:2312.10997). arXiv. https://doi.org/10.48550/arXiv.2312.10997

Hai, R., Koutras, C., Quix, C., & Jarke, M. (2023). Data lakes: A survey of functions and systems. *IEEE Transactions on Knowledge and Data Engineering, 35*(12), 12571-12590. https://doi.org/10.1109/TKDE.2023.3270101

Harby, A. A., & Zulkernine, F. (2022). From data warehouse to lakehouse: A comparative review. In *2022 IEEE International Conference on Big Data (Big Data)* (pp. 389-395). IEEE. https://doi.org/10.1109/BigData55660.2022.10020719

Hariri, R. H., Fredericks, E. M., & Bowers, K. M. (2019). Uncertainty in big data analytics: Survey, opportunities, and challenges. *Journal of Big Data*, *6*(1), 44. https://doi.org/10.1186/s40537-019-0206-3

HDFS architecture guide [User guide]. (n.d.). Apache Hadoop. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Inmon, W. H. (2002). *Building the data warehouse* (3rd ed.). John Wiley & Sons.

Isah, H., Abughofa, T., Mahfuz, S., Ajerla, D., Zulkernine, F., & Khan, S. (2019). A survey of distributed data stream processing frameworks. *IEEE Access, 7*, 154300-154316. https://doi.org/10.1109/ACCESS.2019.2946884

Janković, S., Mladenović, S., Mladenović, D., Vesković, S., & Glavić, D. (2018). Schema on read modeling approach as a basis of big data analytics integration in EIS. *Enterprise Information Systems, 12*(8-9), 1180-1201. https://doi.org/10.1080/17517575.2018.1462404

Janssen, N., Ilayperuma, T., Jayasinghe, J., Bukhsh, F., & Daneva, M. (2024). The evolution of data storage architectures: Examining the secure value of the data lakehouse. *Journal of Data, Information and Management, 6*(4), 309-334. https://doi.org/10.1007/s42488-024-00132-1

Khine, P. P., & Wang, Z. S. (2018). Data lake: A new ideology in big data era. *ITM Web of Conferences*, *17*, 03025. https://doi.org/10.1051/itmconf/20181703025

Kimball, R., & Ross, M. (2002). *The data warehouse toolkit: The complete guide to dimensional modeling* (2nd ed.). Wiley.

Lalaoui, I. L., El Haji, E., & Kounaidi, M. (2025). The evolution and challenges of real-time big data: A review. *Computer Sciences & Mathematics Forum, 10*(1), 11. https://doi.org/10.3390/cmsf2025010011

Laney, D. (2001). *3D data management: Controlling data volume, velocity, and variety* (Application Delivery Strategies, File 949). META Group. https://diegonogare.net/wp-content/uploads/2020/08/3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

Laurent, A., Laurent, D., & Madera, C. (Eds.). (2020). *Data lakes*. ISTE; John Wiley & Sons. https://doi.org/10.1002/9781119720430

Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob, I. (2017). Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access, 5*, 5247-5261. https://doi.org/10.1109/ACCESS.2017.2689040

Martins, A., Abbasi, M., Martins, P., & Sá, F. (2022). BigData oriented to business decision making: A real case study in Constructel. *Computational and Mathematical Organization Theory, 28*(3), 271-291. https://doi.org/10.1007/s10588-021-09330-3

Marz, N., & Warren, J. (2015). *Big data: Principles and best practices of scalable realtime data systems*. Manning Publications.

Mayer-Schönberger, V., & Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.

Nambiar, A., & Mundra, D. (2022). An overview of data warehouse and data lake in modern enterprise data management. *Big Data and Cognitive Computing, 6*(4), 132. https://doi.org/10.3390/bdcc6040132

Nergiz, E., & Barutcu, H. C. (2020). The impact of Industry 4.0 applications on production processes: The case of Bosch Industry and Trade Corporation. *Econder Uluslararası Akademik Dergi, 4*(1), 47-71. https://doi.org/10.35342/econder.666369

Parimi, S. (2025). A comparative performance & metadata study of open table formats: Iceberg vs Delta vs Hudi at scale. *Journal of Computer Science and Technology Studies, 7*(12), 513-520. https://doi.org/10.32996/jcsts.2025.7.12.56

Phillips-Wren, G., Daly, M., & Burstein, F. (2021). Reconciling business intelligence, analytics and decision support systems: More data, deeper insight. *Decision Support Systems*, *146*, 113560. https://doi.org/10.1016/j.dss.2021.113560

Qi, W., Sun, M., & AghaSeyedHosseini, S. (2023). Facilitating big-data management in modern business and organizations using cloud computing: A comprehensive study. *Journal of Management & Organization, 29*(4), 697-723. https://doi.org/10.1017/jmo.2022.17

Rahmani, A. M., Azhir, E., Ali, S., Mohammadi, M., Ahmed, O. H., Ghafour, M. Y., Ahmed, S. H., & Hosseinzadeh, M. (2021). Artificial intelligence approaches and mechanisms for big data analytics: A systematic study. *PeerJ Computer Science*, *7*, e488. https://doi.org/10.7717/peerj-cs.488

Saha, B., & Srivastava, D. (2014). Data quality: The other face of big data. In *2014 IEEE 30th International Conference on Data Engineering* (pp. 1294-1297). IEEE. https://doi.org/10.1109/ICDE.2014.6816764

Schneider, J., Gröger, C., Lutsch, A., Schwarz, H., & Mitschang, B. (2024). The lakehouse: State of the art on concepts and technologies. *SN Computer Science, 5*(5), 449. https://doi.org/10.1007/s42979-024-02737-0

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database system concepts* (7th ed.). McGraw-Hill Education.

Stonebraker, M., Abadi, D., DeWitt, D. J., Madden, S., Paulson, E., Pavlo, A., & Rasin, A. (2010). MapReduce and parallel DBMSs: Friends or foes? *Communications of the ACM, 53*(1), 64-71. https://doi.org/10.1145/1629175.1629197

Vassiliadis, P. (2009). A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM), 5*(3), 1-27. https://doi.org/10.4018/jdwm.2009070101

Walker, C., & Alrehamy, H. (2015). Personal data lake with data gravity pull. In *2015 IEEE Fifth International Conference on Big Data and Cloud Computing* (pp. 160-167). IEEE. https://doi.org/10.1109/BDCloud.2015.62

Wang, F., Raisinghani, M. S., Mora, M., & Forrest, J. (2022). Effective decision support in the big data era: Optimize organizational performance via BI&A. *International Journal of Decision Support System Technology (IJDSST), 14*(1), 1-16. https://doi.org/10.4018/IJDSST.286683

Weidmann, N. B. (2023). *Data management for social scientists: From files to databases*. Cambridge University Press. https://doi.org/10.1017/9781108990424