1

Evolution of Data

Data surrounds us. Every cell of every lifeform holds data of some sort. From the moment we are born, our senses send signals (data) to our brain that we continually process, creating yet more data as part of our thought processes. It's a never-ending cycle. To provide a context, it may be helpful to briefly explore how modern data processing evolved.

Since the beginning of earliest civilizations, humankind has sought to share its thoughts and experiences with others through symbols, such as drawings on cave walls, hieroglyphs in tombs, ancient scrolls, papers, and books. As a species, our passion to learn and progress has led to the desire and need to capture all this data, to store and share it for posterity, and to pass our collective knowledge on to others as a means of building a civilization. The establishment of education delivered through scholastic programs and institutions helped formalize what we learn and how we learn. Educational, governmental, medical, public, and other organizations established their own libraries (the earliest forms dating back to 2600 BC), holding vast quantities of information, accessible for reference or for lending to patrons. Catalogs of this information have helped provide an indexed virtual representation of what is available, how it is stored, and where to find it. Library patrons have also benefitted from the expert assistance offered by librarians or library technicians.

Early Data Storage and Management

Decades ago, analog recordings of audio, photographs, and videos presented new dimensions of capturing data. Punched cards for gathering and processing early census data using tabulating machines appeared. Recorded music on 78 rpm platters and "wire recorders" became a mainstay of radio. Magnetic tape emerged from the laboratory.

Information storage in most people's minds at the end of the World War II era meant books, filing cabinets, or, to those at the leading edge of data-processing technology, paper punch cards. At that time, reels of tape, tape cartridges, and programmable computers were the stuff of science fiction. In 1952, IBM announced the IBM 726, its first magnetic-tape unit, as shown in Figure 1.1. It shipped with the IBM 701 Defense Calculator. This innovation was significant because it was the first IBM large-scale electronic computer manufactured in quantity and was:

- IBM's first commercially available scientific computer
- The first IBM machine in which programs were stored in an internal, addressable, electronic memory
- Developed and produced in record time (less than two years from "first pencil on paper" to installation)
- Key to IBM's transition from punched-card machines to electronic computers with tape storage

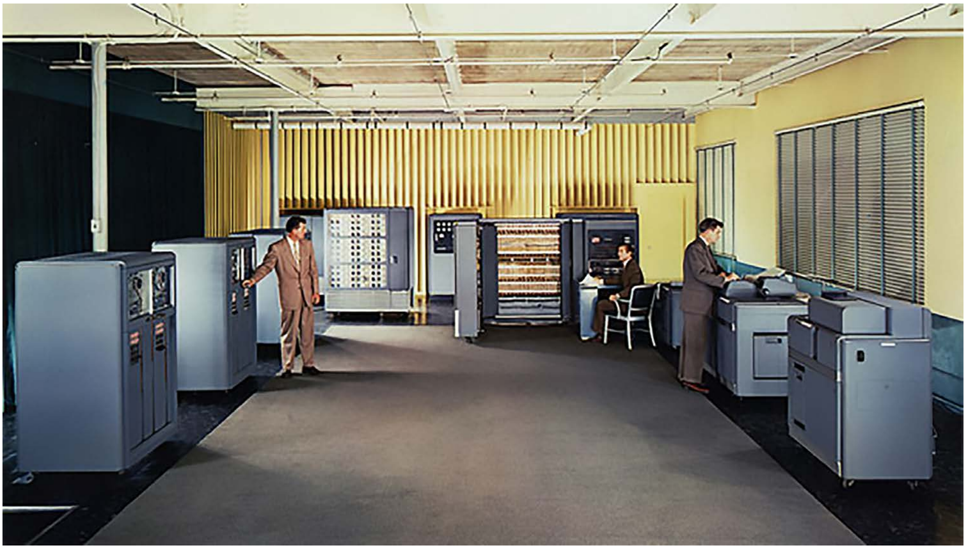

Figure 1.1: An IBM 700 Series

The IBM 701 Electronic Data Processing System included the IBM 701 electronic analytical control unit, IBM 706 electrostatic storage unit, IBM 711 punched-card reader, IBM 716 printer, IBM 721 punched-card recorder, IBM 726 magnetic-tape reader/recorder, IBM 727 magnetic-tape unit, IBM 731 magnetic-drum reader/recorder, IBM 736 power frame #1, IBM 737 magnetic-core storage unit, IBM 740 cathode-ray-tube output recorder, IBM 741 power frame #2, IBM 746 power distribution unit, and IBM 753 magnetic-tape control unit.

What followed was the advent of digital disk storage, which enabled organizations to collect and process more data faster than ever. In 1968, IBM launched the world's first commercial database-management system, called Information Control System and Data Language/Interface (ICS/DL/I). In 1969, it was renamed as Information Management System (IMS).

IBM's Database 2 traces its roots back to the beginning of the 1970s when Edgar F. Codd, a researcher working for IBM, described the theory of relational databases and, in June 1970, published the model for data manipulation.

In 1974, the IBM San Jose Research Center developed a relational Database Management System (DBMS), called System R, to implement Codd's concepts. A key development of the System R project was Structured Query Language (SQL), although it was initially named Structured English Query Language (SEQUEL). The SQL data-management language for relational databases is still in use today.

The name "DB2" was first given to the DBMS in 1983 when IBM released DB2 on its MVS mainframe platform. More information on the history of Db2 is available at https://www.ibm.com/blog/the-hidden-history-of-db2/.

IBM and many other vendors continue to invest in relational and other forms of databases as they are one of the key technologies in online transactional processing (OLTP). IBM Db2, as it is known today, is also used for transaction analytics processing.

Relational databases have become the core technology for data warehouses and Master Data Management (MDM) systems (MDM systems are described below). In parallel to relational databases, other forms of data stores appeared, such as object-oriented, NoSQL, key value, wide-column store, and graph databases, to name but a few.

From Centralized to Distributed

For many years, data storage and processing were centralized. People had to take their work to the computer or access it through "dumb" terminals. With the advent of more-affordable computers, processing and data became decentralized, putting computing power in the hands of individuals. However, this led to a problem of data being replicated in an uncontrolled manner.

With data being created, stored, and processed across many personal devices, it became increasingly difficult to control the sprawl of versions of data sets and apply quality, security, and other controls. It didn't take long for individual departments in various enterprises to start organizing and storing just the data they needed, which gradually resulted in the problem of creating many data silos that usually didn't communicate with each other across an organization.

Master Data Management (MDM) is the discipline by which business and information technology work together to ensure the uniformity, accuracy, stewardship, semantic consistency, and accountability of the enterprise's official shared master-data assets. Combined with data warehousing, MDM helps provide a 360-degree view of a data entity, such as a person or product. (The reference to 360-degree view implies users should be able to look at an entity from many different perspectives to form a more complete understanding of it.) In a sense, MDM's creation was an attempt to recentralize some of the key data that was being held in disparate silos so it could be used across the whole organization as a trusted source of data—a single version of the truth, if you will. However, it still left the problem that there were often prime copies and distributed secondary copies of the data that needed to be kept synchronized to provide a truly trustworthy data source.

Data Stores, Data Integration, and Data Management Tools

Subsequently, numerous solutions appeared for managing and integrating data in order to enable reporting, analysis, and discovery of insights as data volumes grew. All of them were data stores given names such as database, online transactional processing (OLTP), online analytical processing (OLAP), data warehouse, MDM system, data mart, data lake, data lakehouse, and Hadoop. These terms tend to be used somewhat interchangeably at times, but while the terms are similar, important differences exist that are explained below. Each provides certain capabilities and values to different groups of users, but none is a panacea for all data management challenges, as originators hoped for when each was created. However, technology follows a maturity curve or cycle, and these technologies eventually found their own niches as they matured.

Many forms of data stores and data servers are being used across the enterprise today. More variations of these, and new paradigms, will evolve in the future because technology is constantly advancing. The authors of this book believe a data fabric (an architectural approach that simplifies data access in an organization and facilitates self-service data consumption, as discussed in our *Data Fabric* book at MC Press Bookstore (mc-store.com)) can offer enough longevity and flexibility to be able to integrate an organization's current and future data assets and enable them for AI applications.

Data Warehouse

The data warehouse, or enterprise data warehouse (EDW), is a system that aggregates data from different sources, integrating it into a single, central, consistent data store to support data analysis, data mining, AI, and machine learning (ML). A data warehouse system enables an organization to run analytics on huge volumes (terabytes and petabytes) of historical data in ways that a standard database cannot. Aggregating data from different sources into a single warehouse also enables transactional systems to retain sub-millisecond performance while data analysis is concurrently being performed on the warehouse.

Data warehousing systems have been a part of business intelligence (BI) solutions for more than three decades, but their evolution has continued with the recent emergence of new data types and data-hosting methods. Traditionally, a data warehouse was hosted on premises—often on a mainframe computer—and its functionality was focused on extracting data from other sources, cleansing and preparing that data, and loading and maintaining the data within a relational database. More recently, a data warehouse might be hosted on a dedicated appliance or in the cloud, and most data warehouses have added analytics capabilities and data visualization and presentation tools.

A data warehouse provides a foundation for the following:

- **Better data quality:** A data warehouse centralizes data from a variety of data sources, such as transactional systems, operational databases, and flat files. It then cleanses the data set (by fixing incorrect, incomplete, or otherwise erroneous data and eliminating duplicate records) and standardizes it to create a single source of the truth.
- **Faster business insights:** Data from disparate sources limits the ability of decision-makers to set business strategies with confidence. Data warehouses enable data integration across such differing sources, enabling business users to leverage all of a company's data into each business decision.
- **Smarter decision-making:** A data warehouse supports large-scale BI functions such as data mining (finding unseen patterns and relationships in data), AI, and ML—tools data professionals and business leaders can use to get hard evidence for making smarter decisions in virtually every area of the organization.
- **Gaining and growing competitive advantage:** All the above combine to help an organization by finding more data opportunities more quickly than is possible from older methods using disparate data stores.

Data Warehouses and Online Analytical Processing (OLAP)

In a data warehouse environment like the one shown in Figure 1.2, relational databases can be optimized for OLAP to facilitate analysis, enable queries on large numbers of records, and summarize data in many ways. Data stored in the data warehouse can also come from multiple sources.

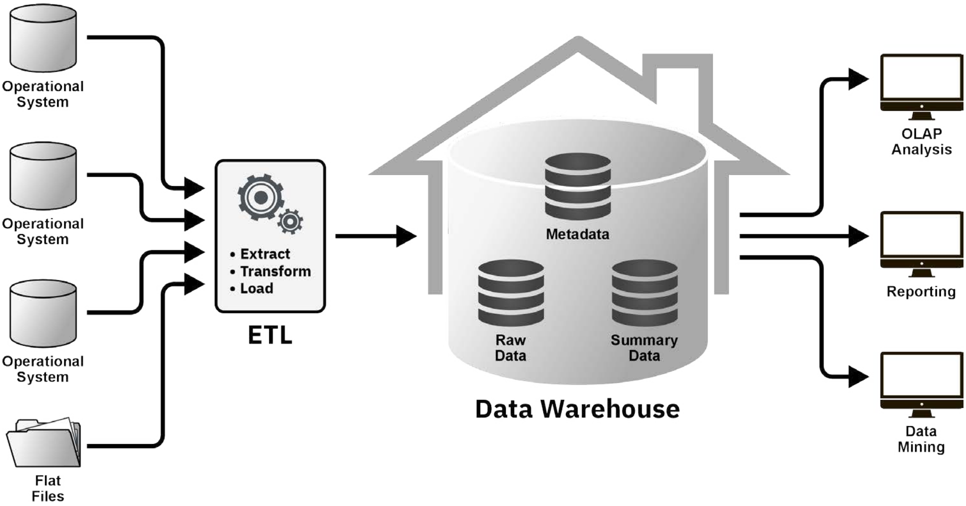

Figure 1.2: A typical data warehouse architecture

Online Analytical Processing (OLAP) vs. Online Transactional Processing (OLTP)

The main distinction between OLAP and OLTP is reflected in their names: analytical vs. transactional. Each system is optimized for that type of processing. OLAP is optimized to conduct complex data analysis for smarter decision-making. OLAP systems are designed for use by data scientists, business analysts, and knowledge workers, and they support BI, data mining, and other decision-support applications. OLTP, on the other hand, is optimized to process a massive number of transactions. OLTP systems are designed for use by frontline workers (e.g., cashiers, bank tellers, hotel desk clerks) or for customer self-service applications (e.g., online banking, e-commerce, travel reservations).

Other key differences between OLAP and OLTP include:

- **Focus:** OLAP systems enable users to extract data for complex analysis. To drive business decisions, the queries often involve large numbers of records. In contrast, OLTP systems are ideal for making simple updates, insertions, and deletions in databases. The queries typically involve just one or a few records.

- **Data source:** An OLAP database has a multidimensional schema, so it can support complex queries of multiple data facts from current and historical data. Different OLTP databases can be the source of aggregated data for OLAP, and they may be organized as a data warehouse. OLTP, on the other hand, uses a traditional DBMS to accommodate a large volume of real-time transactions.
- **Processing time:** In OLAP, response times are orders of magnitude slower than in OLTP. Workloads are read-intensive, involving enormous data sets. For OLTP transactions and responses, every millisecond counts, so OLTP workloads involve simple read and write operations via Structured Query Language (SQL), requiring less time and less storage space.
- **Availability:** Because they don't modify current data, OLAP systems can be backed up less frequently. However, OLTP systems modify data frequently. It is the nature of transactional processing to require frequent or concurrent backups to help maintain data integrity.

Data Warehouse vs. Transactional Database

As mentioned, a database is built primarily for fast queries and transaction processing rather than analytics. A database typically serves as the focused data store for a specific application, whereas a data warehouse stores data from any number (or even all) of the applications in an organization. A database focuses on updating real-time data while a data warehouse typically has a broader scope, capturing current and historical data for predictive analytics, ML, and other advanced types of analyses.

Data warehouses are good foundations for a data system that uses AI and a data-fabric architecture for several reasons:

- Transactional databases are typically smaller and grow to only a few terabytes of data. The larger they grow, the larger the performance impact. Data warehouses are typically several hundred terabytes and can grow to petabytes.

- There is a need to access and analyze data from many different sources to provide data scientists or business analysts with the ability to make better decisions by leveraging different types of data.

- Running analytical queries has a performance impact on any computer system and can take anywhere from multiple seconds to many minutes (and in some cases longer) to execute. While this is acceptable for reporting and BI, it is not acceptable for real-time transactions. Taking a banking transaction as an example, a user could be frustrated by having to wait several minutes to withdraw cash or deposit a check. Because of the impact on performance by analytical queries on a transactional database, moving data to a warehouse becomes the norm and hence the need for these two systems.

Disadvantages of a Data Warehouse

The disadvantages of a data warehouse are centered around the multiple complexities that can result when data needs to be moved or replicated regularly, as data warehouses often require. These include costs, the fact that data is typically out of date or out of synch, and slower performance. Security issues can include the need to provide data protection to multiple environments and the problem that, as additional users gain access to the data warehouse, that access creates security risks to personally identifiable, confidential, and sensitive data.

Data Warehouse vs. Data Lake

A data warehouse gathers raw data from multiple sources into a central repository, structured using predefined schemas (for example, predefined tables, each having a set of defined columns or fields) designed for data analytics. A data lake can be considered a data warehouse without the predefined schemas and often is much larger than warehouses, reaching multi-petabyte scale and higher. As a result, a data lake enables more types of analytics than a data warehouse, as it supports the storage of data in many different formats as and when new data needs to be stored. Data lakes are commonly but not exclusively built on big-data platforms such as Apache Hadoop (defined below).

Data Warehouse vs. Data Mart

A data mart is a subset of a data warehouse that contains data specific to a particular business line or department. Because they contain a smaller subset of data, data marts enable a department or business line to discover insights focused on their users' specific needs more quickly than possible when working with the broader data warehouse data set.

Data Warehouse Appliance

A data warehouse appliance is a pre-integrated bundle of hardware and software (CPUs, storage, operating system, and data warehouse software) that a business can connect to its network and start using as is. A data warehouse appliance sits somewhere between the cloud and on-premises implementations in terms of upfront cost, speed of deployment, ease of scalability, and management control.

Apache Hadoop

Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. The ecosystem is shown in Figure 1.3. It provides a software framework for distributed storage and processing of large data sets using the MapReduce programming model.

Hadoop was originally designed for use with computer clusters built from commodity hardware, which is still the most common use. It has since also found use on clusters of higher-end hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

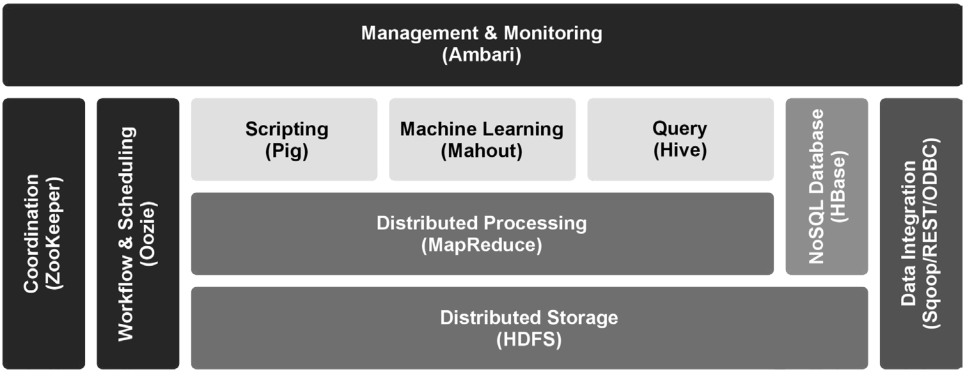

Figure 1.3: Apache Hadoop ecosystem

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part, which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This facilitates concurrent processing by splitting petabytes of data into smaller chunks and processing them in parallel on Hadoop commodity servers. Once processing is complete, Hadoop aggregates all the data from multiple servers to return a consolidated output back to the application.

This approach takes advantage of data locality, under which nodes manipulate the data they can access. This enables the data set to be processed faster and more efficiently than it would be within a more conventional supercomputer architecture that relies instead on a parallel file system in which computation and data are distributed via high-speed networking. Hadoop is considered by many to be a form of data lake.

Data Lake vs. Lakehouse

A data lake, as shown in Figure 1.4, is a centralized data repository for management of extremely large data volumes and serves as a foundation for collecting and analyzing data in its native format(s), whether that data is structured (such as relational database records), semi-structured (records with some structure, while enabling storage of different data types and sources), or unstructured (which may include multimedia and conversational free text). This can help organizations derive new insights, make better predictions, and achieve improved optimization. Unlike traditional data warehouses, data lakes can process video, audio, logs, texts, social media, and sensor data, as well as data and documents to power apps, analytics, and AI.

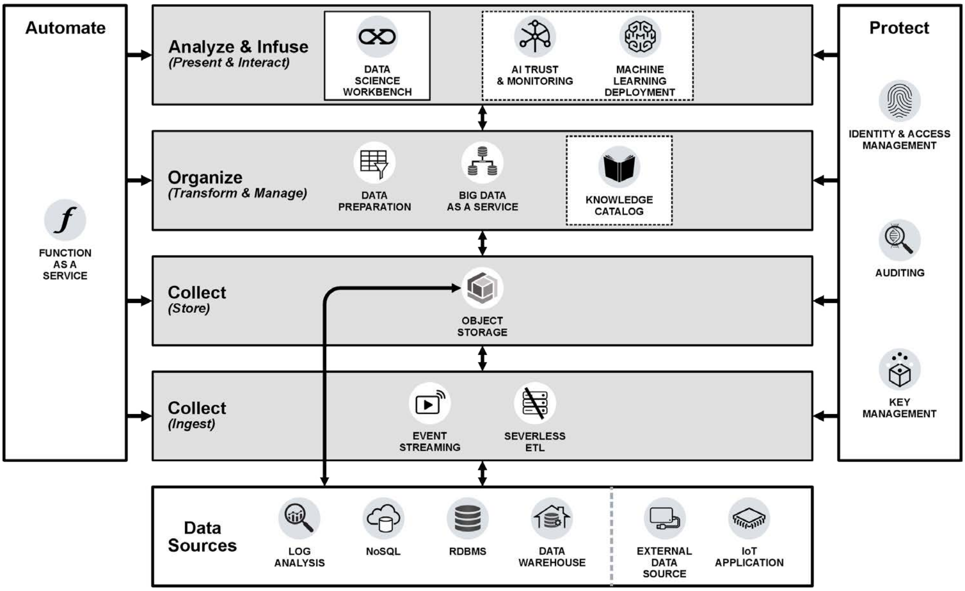

Figure 1.4: Example data lake architecture

Data warehouses and data lakes each evolved to meet a set of specific technology and business needs and values. As organizations often need both, there has been increasing demand for a convergence of both technologies. Thus, the lakehouse was born. A lakehouse couples the cost benefits of a data lake with the data structure and data management capabilities of a data warehouse. It is an evolution of the analytic data repository that supports acquisition to refinement, delivery, and storage with an open table format. Without going into detail, it is defined as part of Apache Iceberg (see https://iceberg.apache.org) and was designed for handling huge analytic data sets. It is used in production where a single table can contain tens of petabytes of data and the data can be read without a distributed SQL engine.

Lakehouses are designed to help organizations get more from their existing investment in data warehouses and data lakes. It supports the existence of both through access to and management of a larger variety of combined data for increased flexibility. Lakehouses can provide users with the following abilities:

- Understand and anticipate customer behaviors with more complete, governed (validated) insights.
- Spot patterns and trends to reduce waste and overhead through more diverse analytic and AI techniques.
- Promote auditability and transparency with metadata-powered, native data access in a governed data lake.
- Speed time to value with self-service data exploration and discovery for users.
- Increase collaboration in an integrated environment and reduce the time and cost of managing disparate systems and tools.
- Turn open-source and ecosystem investments into innovation opportunities with enterprise-ready, secure data lakes.
- Provide significant cost savings.

Lakehouses can:

- Reuse the data lake for 360-degree customer and operational intelligence, governance, and risk and compliance reporting.

- Ingest and integrate with transactional, operational, and analytical data to promote a complete insight.
- Extend information architectures to provide the right data at the right time on a common foundation for staging, storage, and access.
- Build and maintain a data foundation that powers data cataloging, curation, exploration, and discovery needs.
- Take a hybrid approach to access any data from any location, spanning real-time data to databases containing years of records.
- Integrate and expand analytics across multiple data repositories, revealing deeper, more holistic insights to help drive broader innovation and optimization across the enterprise to meet business needs ("at scale").

As organizations continue to move parts of their data estates and processing into hybrid multi-clouds, data lakes and lakehouses help provide optimum value, building on the following principles:

- Secure data-sharing across multiple teams accessing enterprise data: Organizations should be able to rely on data lake governance that houses raw structured and unstructured data—trusted, secured, and governed—with automated privacy and security anywhere.
- Presence of data-integration tools that combine data from disparate sources into valuable data sets: Such tools include those that provide extract, transform, load (ETL) function; enable controlled data replication and data virtualization (creating an integrated view of data spread across different data repositories, regardless of how they are physically stored and represented); and can extract large volumes of data from source systems and load it where applicable to a data warehouse.
- Via data virtualization, the ability to query data directly in the data lake without duplication or movement

In summary, a lakehouse is an emerging data-management architecture that converges data warehouse and data lake capabilities to meet modern data challenges such as ballooning costs, data silos, rapid data growth, and sprawling data, which otherwise often prevent organizations from getting the most out of their data. A lakehouse architecture is designed to help legacy warehouse customers tackle new problems and workloads by handling larger and more varied data sets, leading to new decision-making capabilities. The lakehouse architecture, if implemented correctly, can help reduce the need for complex data pipelines by enabling different analytic engines to operate on the same data store. These new abilities will ultimately help customers reduce the time needed to generate insights and optimize costs.