What is a data lake & why does it matter?

A data lake is a storage repository that can rapidly ingest large amounts of raw data in its native format. As a result, business users can quickly access it whenever needed and data scientists can apply analytics to get insights. Unlike its older cousin – the data warehouse – a data lake is ideal for storing unstructured big data like tweets, images, voice and streaming data. But it can be used to store all types of data – any source, any size, any speed, any structure.

Some of the formats for information stored in data lakes are:

- Structured data, such as rows and columns from relational database tables.
- Semistructured data, like delimited flat text files and schema-embedded files.
- Unstructured data – including social media content and data from the Internet of Things (IoT) – as well as documents, images, voice and video.

History of data lakes: Where it all started

The early versions of what we now call data lakes were pioneered by the watering holes of the yellow elephant – Hadoop. When it was first released, Hadoop was a collection of open source software used for the distributed storage, processing and analytics of big data. It was especially useful for emerging sources of semistructured and unstructured data that were becoming more prevalent at the time. Data lakes were also used to scale up for structured data whose volumes were rising rapidly.

Unfortunately, the early hype around Hadoop implied that you could arbitrarily throw any amount of data into a lake, then let users fend for themselves. Multiple publicized failures proved that approach was wrong. Some early adopters saw their data lakes quickly devolve into poorly managed and ungoverned dumping grounds akin to data swamps. This resulted in:

- Redundancy, which skewed analytical results.
- Non-auditable data, which no one would trust.
- Poor query performance, which killed the primary early purposes of the data lake – high-performance exploration and discovery.

These undocumented and disorganized early data lakes were nearly impossible to navigate. Metadata tagging evolved to become one of the most essential data lake management practices because it made the data in the lake easier to find. Data lake governance improved its auditability and trustworthiness, verifying its usability for more enterprise applications.

The technologies and methodologies used to implement a data lake have matured over time. Now they include not only Hadoop but also other traditional and big data technologies.

Faster Insights From Faster Data: A TDWI Best Practices Report

For competitive advantage, businesses need to make fast, data-driven decisions. Data lakes are flexible platforms useful for any type of data – including operational, time-series and near-real-time data. Learn how they work with other technologies to provide fast insights that lead to better decisions.

Moving past the early hype

As the hype of early data lakes faded away, a data lake stopped being mistaken for a data platform. Instead, it was recognized as a container for multiple collections of varied data coexisting in one convenient location.

Today, data lakes are formally included in enterprise data and analytics strategies. Organizations recognize that the term data lake refers to just one part of the enterprise ecosystem, which includes:

- Source systems.
- Ingestion pipelines.
- Integration and data processing technologies.
- Databases.
- Metadata.
- Analytics engines.
- Data access layers.

To be a comprehensive business intelligence platform that generates high business value, a data lake requires integration, cleansing, metadata management and governance. Leading organizations are now taking this holistic approach to data lake management. As a result, they can use analytics to correlate diverse data from diverse sources in diverse structures. This means more comprehensive insights for the business to call upon when making decisions.

Why are data lakes important?

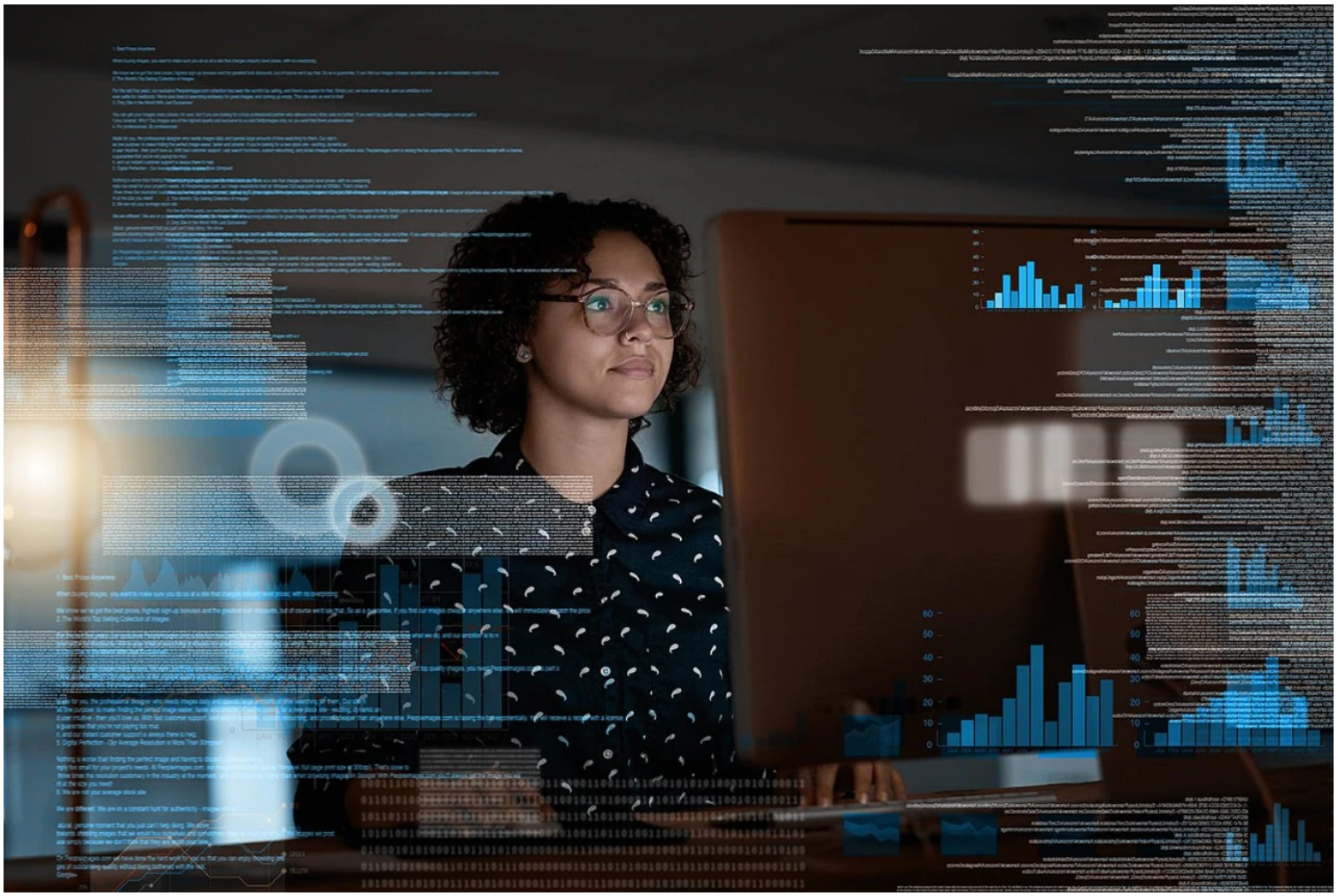
Because a data lake can rapidly ingest all types of new data – while providing self-service access, exploration and visualization – businesses can see and respond to new information faster. Plus, they have access to data they couldn't get in the past.

These new data types and sources are available for data discovery, proofs of concept, visualizations and advanced analytics. For example, a data lake is the most common data source for machine learning – a technique that's often applied to log files, clickstream data from websites, social media content, streaming sensors and data emanating from other internet-connected devices.

Many businesses have long wished for the ability to perform discovery-oriented exploration, advanced analytics and reporting. A data lake quickly provides the necessary scale and diversity of data to do so. It can also be a consolidation point for both big data and traditional data, enabling analytical correlations across all data.

Although it's typically used to store raw data, a lake can also store some of the intermediate or fully transformed, restructured or aggregated data produced by a data warehouse and its downstream processes. This is often done to reduce the time data scientists must spend on common data preparation tasks.

The same approach is sometimes used to obscure or anonymize personally identifiable information (PII) or other sensitive data that's not needed for analytics. This helps businesses comply with data security and privacy policies. Access controls are another method businesses can use to maintain security.



To be a comprehensive business intelligence platform that generates high business value, a data lake requires integration, cleansing, metadata management and governance. Many organizations are taking this holistic approach to data lake management.

Data lake versus data warehouse

Wondering what you need to consider when comparing a data lake and data warehouse? One of the top considerations relates to the design, or schema, of the data store.

Relational databases and other structured data stores use a schema-driven design. This means any data added to them must conform to, or be transformed into, the structure predefined by their schema. The schema is aligned with associated business requirements for specific uses. The easiest example of this type of design is a data warehouse.

A data lake, on the other hand, uses a data-driven design. This allows for rapid ingestion of new data before data structures and business requirements are defined for its use. Sometimes data lakes and data warehouses are differentiated by the terms schema on write (data warehouse) versus schema on read (data lake).

- **Schema on write (data warehouse)** limits or slows ingestion of new data. It is designed with a specific purpose in mind for the data, as well as specific associated metadata. However, most data can serve multiple purposes.
- **Schema on read (data lake)** retains the raw data, enabling it to be easily repurposed. It also allows multiple metadata tags for the same data to be assigned.

Since it's not restricted to a single structure, a data lake can accommodate multistructured data for the same subject area. For example, data lakes can blend structured sales transactions with unstructured customer sentiment. And since it's focused on storage, a data lake requires less processing power than a data warehouse. Data lakes are also much easier, faster and less expensive to scale over time.

One disadvantage of a lake is that its data is not standardized, unduplicated, quality-checked or transformed. In response, some people have adopted a trend to use data lakes differently. Lakes can provide a new, improved zone for landing and staging data before it's prepared, integrated and transformed for loading into the data warehouse.

These examples illustrate why a data lake does not replace a data warehouse – it complements the data warehouse. Besides being used as staging areas, lakes can also serve as archives. In this scenario, outdated data is archived but kept readily accessible for auditing and historical analysis.

Data lakes in the early days: A deeper dive

Early data lakes used the open source Hadoop distributed file system (HDFS) as a framework for storing data across many different storage devices as if it were a single file. HDFS worked in tandem with MapReduce as the data processing and resource management framework that split up large computational tasks – such as analytical aggregations – into smaller tasks. These smaller tasks ran in parallel on computing clusters of commodity hardware.

In its second release, Hadoop made an improvement that decoupled the resource management framework from MapReduce and replaced it with Yet Another Resource Negotiator (YARN). This essentially became Hadoop's operating system. Most important, YARN supported alternatives to MapReduce as the processing framework. This greatly expanded the applications (and data management and governance functions) that could be executed natively in Hadoop.

Check out some related content

Data lakes are formally included in many organizations' data and analytics strategies today.

Ready to learn more about some related topics? In the box to the right, learn how data integration has evolved and check out our tips for building better data lakes. Discover why governance is essential, and get the latest on data tagging best practices. Or, read all about the ins and outs of cloud computing.

-
- 3 tips for building a better data lake

  Do you know the most common pitfalls of building a data lake? Read this blog post to get great tips about how to skip the mistakes others have made.
- Cloud computing

  Cloud platforms are an integral part of many organizations' data strategies today, including decisions to place a data lake in the cloud. In this primer, you'll learn all about cloud

computing and why it's a major force for business innovation.

- 4 data tagging best practices

  Metadata tagging is an essential data lake management practice because it makes the data in the lake easier to find. In this blog post, read about data tagging best practices and why it's so important to tag your data correctly.

- Data integration: It ain't what it used to be

  As organizations ingest larger volumes of structured and unstructured data, many move data into a lake built using an underlying object store and custom metadata. Read this article to see how data integration techniques have evolved over time.

How data lakes work today

Rapid ingestion and the ability to store raw data in its native format have always been key benefits of data lakes. But what exactly does that mean? And how does it work?

- **Raw data** means that the data has not been processed or prepared for a particular use. Some data sources, however, have previously applied some amount of processing or preparation to their data. So, a data lake stores raw data in the sense that it does not process or prepare the data before storing it. One notable exception relates to formatting.
- **Native format** means data remains in the format of the source system or application that created it. However, this is not always the best option for data lake storage. In fact, rarely does rapid ingestion simply mean copying data as-is into a file system directory used by the lake.

For example, a Microsoft Excel spreadsheet is, by default, in its native XLS format. But most data lakes would prefer to store it as a delimited flat text file in comma-separated values (CSV) format. Transactional data from relational databases is also often converted to CSV files for storage in the lake.

Embedded schema and granular data

Another common alternative is to use a file format with embedded schema information, such as JavaScript Object Notation (JSON). For example, clickstream data, social media content and sensor data from the IoT are usually converted into JSON files for data lake storage. JSON files are also a good example of how data lake ingestion often involves converting data from its native format into a more granular format.

Granular data, especially with embedded schema information such as key-value pairs (KVP), enables faster read and write operations. This means that it:

- Does not waste storage on placeholders for optional, default or missing keys or values.
- Can be aggregated and disaggregated to meet the needs of different situations.
- Becomes easier to retrieve only the data applicable to a specific use.

Additional, and more optimized, data lake storage formats are also available. Some of these storage formats enable better scalability and parallel processing, while also embedding schema information. Data can be converted into:

- Column stores (e.g., Redshift, Vertica, Snowflake).
- Compressed column-oriented formats (e.g., Parquet for Spark or ORC for Hive).
- Conventional row-oriented formats (e.g., PostgreSQL, MySQL or other relational databases).
- Compressed row-oriented formats (e.g., Avro for Kafka).
- In-memory stores (e.g., SingleStore, Redis, VoltDB).
- NoSQL stores (e.g., MongoDB, Elasticsearch, Cassandra).

When to use different data storage options

Most data lakes use a variety of storage options, depending on the data sources and business cases. This is especially true for business cases related to access and analytics. For example:

- Column-oriented storage works best when rapid retrieval and accelerated aggregation are most important.
- Row-oriented storage works best when there is a lot of schema variability, as is often the case with streaming applications. It’s also ideal when the data lake is used as a staging area for a data warehouse.
- In-memory storage works best for real-time analytical use cases.
- NoSQL storage works best for analytics scenarios that require rapid generation of metrics across large data sets.

Data lakes and the importance of architecture

The bottom line is that a data lake is not just a massive repository – it requires a well-designed data architecture. It’s possible to use a broad range of tools for implementing rapid ingestion of raw data into the data lake. Such tools include the data integration and extract-transform-load (ETL) tools your enterprise likely already has. Certain new big data technologies (including some of the examples above) provide this functionality as well.

Regardless of how you choose to implement ingestion and storage, data lakes can involve deploying back-end technologies you’re less familiar with – especially nonrelational database management systems (non-RDBMSs). Luckily, many of these technologies include user-friendly front-end interfaces. For example, some provide SQL-like query functionality that many users expect and already know how to use.

Get to know SAS® Viya®

Data is constantly changing around us – and our decisions need to adapt quickly. To get started with turning raw data into meaningful decisions, organizations have to first access, explore, transform and prepare data for analysis.

The data lake gives business users and data scientists alike access to data they couldn’t get in the past – allowing them to explore and visualize the data from one convenient location. In turn, they can see and respond to new information faster.

Data lakes complement SAS Viya, which is an artificial intelligence, analytics and data management platform that’s used to transform raw data into operational insights. SAS Viya supports every type of decision an organization needs to make. Watch the video to learn more.

Data lake and cloud

A data lake can be used as a centralized repository through which all enterprise data flows. As such, it becomes an easily accessible staging area where all enterprise data can be sourced. This includes data consumed by on-site applications as well as cloud-based applications that can accommodate big data’s size, speed and complexity. All of which leads to the question of data lake versus cloud: Where should the data lake be located?

Cloud data lake

For some enterprises, the cloud may be the best option for data lake storage. That’s because it provides complementary benefits – elastic scalability, faster service delivery and IT efficiency – along with a subscription-based accounting model.

On-site data lake

Enterprises may opt for grounding their data lake within their own walls for reasons similar to arguments made for managing a private cloud on-site. This approach provides the utmost security and control while protecting intellectual property and business-critical applications. It can also safeguard sensitive data in compliance with government regulations.

But the disadvantages of managing a private cloud on-site also apply to a data lake. Both can lead to increased in-house maintenance of the data lake architecture, hardware infrastructure, and related software and services.

Hybrid data lake

Sometimes businesses choose a hybrid data lake, which splits their data lake between on-site and cloud. In these architectures, the cloud data lake typically does not store data that is business critical. And if it contains personally identifiable information (PII) or other sensitive data, it is obscured or anonymized. This helps the business comply with data security and privacy policies. To minimize cloud storage costs, the data stored in the cloud can be purged periodically or after pilot projects are completed.

About Jim Harris

Jim Harris is a recognized data quality thought leader with 25 years of enterprise data management industry experience. He is an independent consultant, speaker and freelance writer. Harris is the Blogger-in-Chief at Obsessive-Compulsive Data Quality, an independent blog offering a vendor-neutral perspective on data quality and its related disciplines, including data governance, master data management and business intelligence.

Recommended reading

Ready to subscribe to Insights now?

New enhancements. Powerful capabilities. Innovative results.