**Data Storage and Optimization**

# **Introduction to module 7**

Hi, everyone. Welcome back to another module. At this point, you know how to use Spark and Spark SQL to access big data. You've also used both of those tools to work with big data. Now, we want to introduce one more concept, and that is Delta Engine. It's going to help you work faster and more efficiently, better, more optimized. You get the point. Delta Engine speeds up query performance on Data Lakes. It's even more powerful when combined with Delta Lake. We'll talk about all of that in this module. In this module, we'll also talk about data storage solutions because depending on where you work, your organization might store data in a number of ways. We want to make sure to cover some of that in this module. By the end of this module, you'll be able to list differences between data storage solution, specifically Data Warehouses and Data Lakes. You'll be able to describe how a Lakehouse relates to data management system, and also describe elements of Delta Lake and Delta Engine. When you're ready, click onto the next video to get started with this module.

# **A quick refresher**

Before we learn about delta lake and delta Engine, we'll review some of the most common ways, that companies store big data and how data storage can impact analysts, like here. As you read, keep in mind all of the big data characteristics that we went over earlier, in this course. We'll be looking for storage options that work well with the following characteristics. Big Data means volume. Remember that, our storage solution has to be able to handle massive volume. Companies are constantly collecting data, and they need a flexible place to store it. Velocity. Data's coming in super-fast and we need to be able to deal with it. Variety. Data's coming in from lots of different sources and in lots of different formats, including structured, unstructured, and semi-structured data. Veracity. The quality of data-driven decisions depends strictly on the accuracy and reliability of our data. Value. Analytics brings value to Big Data. The storage solution we choose must be able to support quick access to data and be able to deliver insights. As you go through this next reading, keep these characteristics in mind to help you evaluate the relative benefits of each storage solution.

# **Data Warehouses**

First appearing in the 1980s, data warehouses evolved as big data emerged and became one of the first solutions to accommodate its needs. Before the data warehouse, companies were storing data in lots of different systems with no way to unite the disparate databases. For analysts, a disparate system can make it impossible to synthesize a total view on which to base actionable business insights.

Data warehouses brought the collection of databases all under a single umbrella and allowed the data to be queried and viewed as a whole.

### Advantages

In the late 1990's, data warehouses were the most dominant data architecture for big companies. The primary advantages include:

* Standard structured query language (SQL) for access
* Integration of many data sources
* Data optimized for fast reads
* Ability to run quick ad-hoc analytical queries

These advantages enable data analysts to use an intuitive and powerful query language, SQL, to easily access data from a variety of sources. Also, the data is stored such that it can be delivered quickly to show results of analytical queries.

### Challenges

As the speed and scale of data really exploded though, some deficiencies became clear:

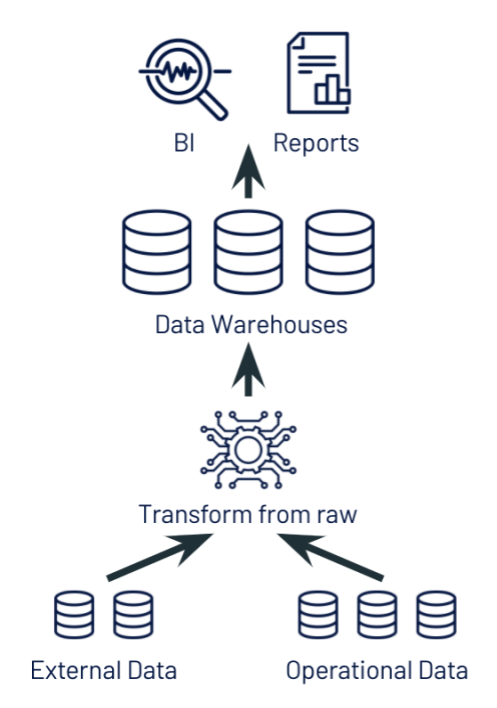
* Inability to store unstructured, raw data
* Difficult to scale
* Requires significant investment in a proprietary system

All of these add up to bottlenecks and roadblocks for data analysts.

Since warehouses are unable to store unstructured, raw data, data teams who collect both are automatically resigned to work within different systems. As big data gets bigger, it continues to push the limits of a legacy data warehouse. Scaling up requires engineers dedicated strictly to managing the infrastructure. More time on systems engineering translates into delays in analytics.

Once a company builds expensive, proprietary hardware and software into their system, it can be almost impossible to leave it.

Now, there are some cloud-based data warehouse solutions that have been built to address some of these challenges, but still - we are missing support for unstructured data and streaming data. As a result, many organizations employ data warehouses only for smaller subsets of their data, as in the diagram below. You can see that much of their data is flowing into data storage outside of the data warehouse, and only subsets are available at the data warehouse level.



# **Data Lakes**

Data Lakes are newer to the landscape, emerging around 2010 and developing over the past decade,  to answer some of the problems organizations were facing with their data warehouses. As big data continues to evolve, the data we collect is increasingly unstructured, fast-moving, and high-volume.

Data lakes are often used to consolidate all of an organization’s data in a single, central location, where it can be saved “as is,” without the need to impose a schema or structure on it up front. Data in all stages of the refinement process can be stored in a data lake: raw data can be ingested and stored right alongside an organization’s structured, tabular data sources (like database tables), as well as intermediate data tables generated in the process of refining raw data. Unlike most databases, data lakes can process all data types including images, video, audio and text.

Today, companies have lots of data, but it’s often isolated and siloed away in different storage systems: data warehouses, databases, and other storage systems across the enterprise. A data lake breaks down these data silos, centralizing and consolidating all of your organization’s data assets into a complete and authoritative data store for analytics that is always up to date. Unifying all data in a data lake is the first step for companies that aspire to harness the power of machine learning and data analytics to win in the next decade.

A data lake’s flexible, unified architecture opens up a wide range of new use cases for cross-functional enterprise scale analytics, BI, and machine learning projects that can unlock massive business value. Data analysts can harvest rich insights by querying the data lake using SQL, data scientists can join and enrich data sets to generate ML models with ever greater accuracy, data engineers can build automated ETL pipelines, and business intelligence analysts can create visual dashboards and reporting tools faster and easier than before. These use cases can all be performed on the data lake simultaneously, without lifting and shifting the data, even while new data is streaming in.

### Advantages

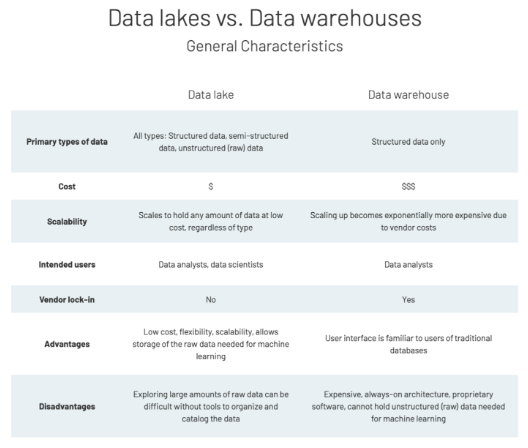
* Can hold all of an organization's data: structured, unstructured, and semi-structured
* Centralizes data for access by the whole data team
* Storage is relatively inexpensive, and the quantity of storage can be easily increased for scalability

### Challenges

* Data reliability - without proper tools, it can be difficult to maintain data lakes which can threaten the veracity of an organizations data.
* Query performance - in many systems, as data lakes grow larger, query performance suffers. This can be due to a variety of factors including slowdowns around metadata management and improper data partitioning.

# **Data Lakes vs Data Warehouses**

Use this chart to help compare Data lakes and Data Warehouses



# **Introducing a new data management paradigm**

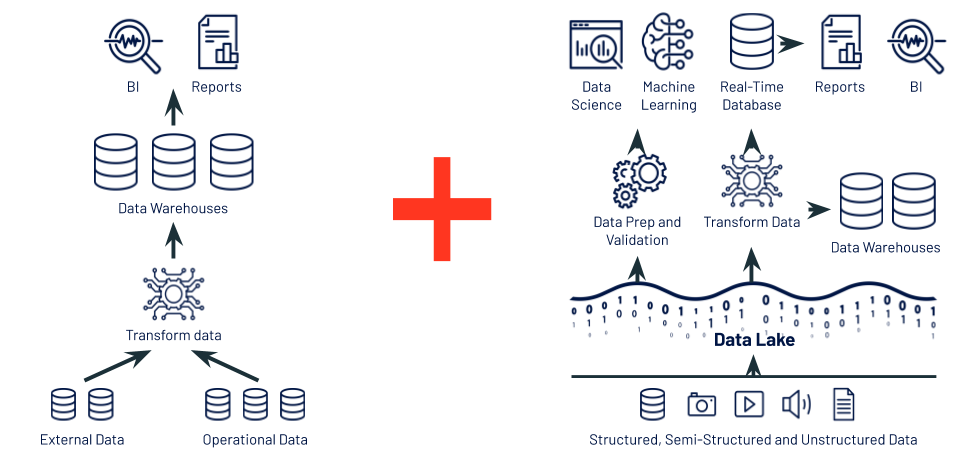
The next reading is a debit from a blog post about a new Data Management paradigm that has emerged over the past few years, The lake house. It is meant to merge the benefits of a data warehouse and a data lake into a single system. That means you get all the benefits of working with the Data Warehouse, access via SQL, super-fast reads, fast ad-hoc queries, and all the benefits of working with a data lake. Access to multiple data sources, open Storage, to files, money format, and the ability to work with a single source of truth. Continue into the reading to learn more about the system.

# **The Lakehouse**

Over the past few years at Databricks, we’ve seen a new data management paradigm that emerged independently across many customers and use cases: the lakehouse. In this reading we describe this new paradigm and its advantages over previous approaches.

We've identified both benefits and challenges for two types of popular data storage solutions available for working with big data. The lists of benefits and challenges in each system do tend to align - that is, where one system falters, the other is proficient and vice versa.

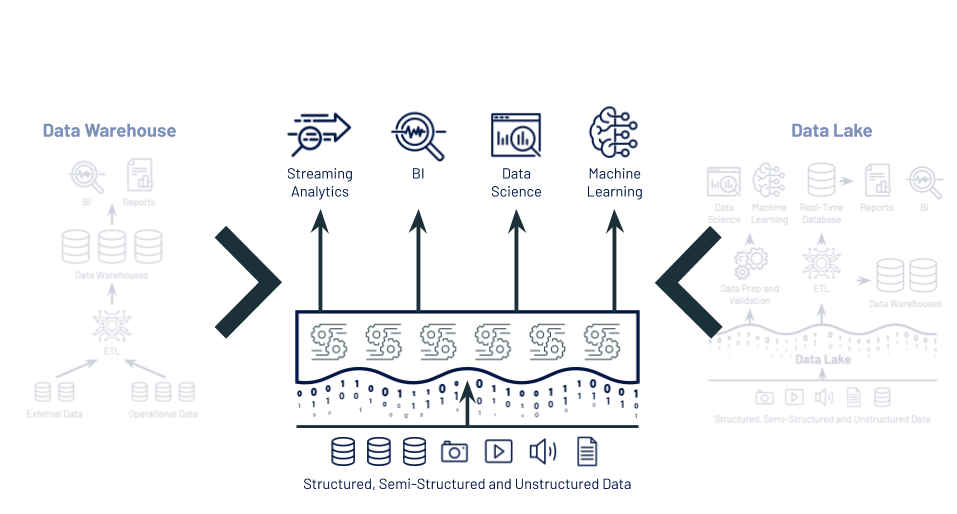
As a result, many organizations have taken to using some form of both, ostensibly hoping to develop a solution that can offer the best from both systems.



Following this path, we quickly find out that maintaining two systems is difficult at best, and at worst impossible. It is high cost, complex, and not reliable as a single source of truth.

### The Lakehouse

A Lakehouse combines the best elements of data lakes and data warehouses. They use similar data management features as those at work in data warehouses, thereby delivering the fast query speeds and reliable, manageable data. But, they are built directly on the low-cost and flexible storage used for data lakes. They are what you would get if you had to redesign data warehouses in the modern world, now that cheap and highly reliable storage (in the form of object stores) are available.



A lakehouse has the following key features, among others:

* support for diverse data types and formats
* ability to use BI tools directly on source data
* support for diverse workloads (BI, data science, machine learning, and analytics)
* data reliability and consistency

### From BI to AI

The lakehouse is a new data management paradigm that radically simplifies enterprise data infrastructure and accelerates innovation in an age when machine learning is poised to disrupt every industry. In the past most of the data that went into a company’s products or decision making was structured data from operational systems, whereas today, many products incorporate AI in the form of computer vision and speech models, text mining, and others. Why use a lakehouse instead of a data lake for AI? A lakehouse gives you data versioning, governance, security and [ACID](https://en.wikipedia.org/wiki/ACID) properties that are needed even for unstructured data.

Current lakehouses reduce cost but their performance can still lag specialized systems (such as data warehouses) that have years of investments and real-world deployments behind them. Users may favor certain tools (BI tools, IDEs, notebooks) over others so lakehouses will also need to improve their UX and their connectors to popular tools so they can appeal to a variety of personas. These and other issues will be addressed as the technology continues to mature and develop. Over time lakehouses will close these gaps while retaining the core properties of being simpler, more cost efficient, and more capable of serving diverse data applications.

# **Using Delta Lake**

# **Introduction to the lesson**

You just learned how data storage is evolving and improving to meet the needs of the data community. We left off describing a lake house, an emerging data management model that takes the best of what data warehouses have to offer. And combines it with the best of what data lakes have to offer, while resolving the aforementioned challenge. For Spark, Delta Lake is what makes the lake house possible. Delta Lake is an open source storage layer that brings reliability to data lakes. You'll be working with data you can count on, plus there's a built in optimization engine that can drastically improve your query performance on a data lake. It runs right on top of your existing data lake, and it's fully compatible with all of Apache's Spark's APIs. Now, you learn about Delta Lake and Delta engine and how you can use these technologies to optimize query performance in Spark.

# **What is Delta Lake**

By the end of this video, you'll be able to describe elements of Delta Lake and Delta Engine. Best place to start here is probably to figure out what exactly is Delta Lake. Delta Lake is a technology for building robust data lakes and is a core component of building your data lakehouse. Offers guaranteed consistency because it's ACID compliant. ACID transactions are typically a thing that engineers might talk about. Basically, for us, this speaks to the data reliability we need to make sure that our analytics are based on correct and up-to-date data. It is because of this consistency and reliability that we consider Delta Lake a robust data store where traditional data lake is not. Also, it's designed to work with Apache Spark. You get all the benefits that we've talked about throughout this course, working with Apache Spark plus reliable transactions on a data lake. We can break down the elements of Delta Lake into these four pieces which we'll go over in more detail in the next few slides. There's Delta architecture, the Delta storage layer, the Delta Engine, and Delta tables. The Delta architecture design pattern describes how raw data will be transformed and loaded into successively cleaner Delta Lake tables. Bronze tables hold raw data, and then we apply transformations to make clean silver tables. Once that data is cleaned, it can be aggregated into gold level tables as shown in this graphic. In this system, you're the end user and you'll be working with a gold or silver tables that come through this pipeline. What's really important to notice here is that you'll be working along with the rest of your data team, in a single unified system that is reliable and up-to-date. You will be accessing that elusive single source of truth on which all organizations want to base their decisions. The next piece we're want to talk about is the Delta storage layer. This is really what transforms a data lake into a Delta Lake. It runs on top of the cloud storage that holds all of your data. Let's remember that keeping all of your data in object storage is the central design pattern, that defines a data lake. We've got this data lake that holds all of your data, but as we know, traditional data lakes are facing challenges around data reliability, and query performance. The Delta storage layer provides this structured transactional layer, that allows you to easily query your data lake. Plus on top of that, it provides a layer that we can access for all possible applications. Whether that be in business intelligence, data science, machine learning, or streaming analytics. The Delta storage layer tracks and indexes all the files in your data lake so that organizations can guarantee the data is consistent. They can track metadata. You'd automatically handle variations and schema over time, which built-in version control so you can go back to your data at a certain point in time and role all the changes back to them. Also, their support easily merge and update data as it arrives. Which could be a huge paying point for a traditional data lake. If you're used to working with a traditional data lake, the Delta Engine is where you will see the most performance improvements. The Delta Engine is a high-performance query engine. Is obviously Apache Spark compatible and provides an efficient way to process data and data lakes. These optimizations accelerate data lake operations and support a variety of workloads from engineering to analytics. A lot of these optimizations take place automatically. You simply get the benefits of these Delta Engine capabilities just by using databricks for your data lakes. We will talk more about some particular available optimizations including file management optimizations. We'll have a hands-on practice with that. Performance optimization with Delta caching, and you can read more about dynamic file pruning and adaptive query execution. The fourth component, Delta tables, just means that you will have the ability to write a table as a Delta table so that you can create Delta files, register a table in the metastore, and start a transaction log for that table. It's really simple. As you can see from the syntax here, all course long, we have been creating tables using a certain file format. Instead of doing that, we can now create a table and specify that we want to use Delta. When we do that, that will create those delta files and start a transaction log. There is one additional step that we'll take to register the table and the metastore. In the next lesson, you'll get some hands-on practice actually transforming data as it moves from raw to gold, so that you can see that process, and you'll also get to see some performance optimizations in action.