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. That puts completely new demands on the DM system, and it’s not just about the capabilities; it’s about the architectural approach.is about using the principles of a well-designed platform that leverages the scalable resources of the cloud to manage all an organization’s data.

This book helps you understand more about how the lakehouse   
makes your DM efforts more effective and efficient in your company.

Data management (DM) consists of methods, architectural techniques, and tools for gaining access to and managing delivery of data in a consistent way across different data types in a company. The purpose of DM on an enterprise-wide   
scale is to fulfill all data requirements for use cases, applications, and business processes in a company.

In the early days of DM, the relational database was the primary method that companies used to collect, store, and analyze data. Relational databases offered a way for companies to store and analyze highly structured data about their customers using Structured Query Language (SQL). relational databases were simple and reliable.

Without a way to centralize and efficiently use their data, companies ended up with decentralized, fragmented stores of data, called data silos, across the organization. With so much data stored in different source systems, companies needed a way to integrate them. Data warehouses were born to meet this need and to unite disparate databases across the organization. As data volumes grew even larger (big data), and as the need to manage unstructured and more complex data became more important, data warehouses had limitations:   
» Data warehouses for a huge IT project can involve high   
maintenance costs.   
» Data warehouses only support business intelligence (BI) and   
reporting use cases.

» There’s no capability for supporting ML use cases.   
» Data warehouses lack scalability and flexibility when   
handling various sorts of data in a data warehouse.   
This started the push for yet another DM solution: data lakes   
that could offer repositories for raw data in a variety of   
formats.

Apache Hadoop emerged as an open-source distributed   
data processing technology. Apache Hadoop is a collection of   
open-source software for big data analytics that allowed large data   
sets to be processed with clusters of computers working in parallel.

Early data lakes built on Hadoop MapReduce and HDFS enjoyed   
varying degrees of success. Some early data lakes succeeded,   
while others failed due to Hadoop’s complexity and other factors.

Shortly after the introduction of Hadoop, Spark was introduced.   
Spark was the first unified analytics engine that facilitated large-  
scale data processing, SQL analytics, and ML. Spark was also 100   
times faster than Hadoop.

Today, many modern data architectures use Spark as the pro-  
cessing engine that enables data engineers and data scientists   
to perform ETL, refine their data, and train ML models. Cheap   
blob storage (AWS S3 and Microsoft Azure Data Lake Storage) is   
how the data is stored in the cloud, and Spark has become the   
processing engine for transforming data and making it ready for   
BI and ML.

While suitable for storing data, data lakes lack some critical   
features:   
» They don’t support transactions.   
» They don’t enforce data quality.   
» Their lack of consistency and isolation makes it almost   
impossible to mix appends and reads, and batch and   
streaming jobs.

More than ever, companies require systems   
for diverse data applications, including SQL analytics, real-time   
monitoring, and ML. Most of the recent advances in AI have been   
in better models to process unstructured data (text, images, video,   
audio).

However, having a multitude of systems   
introduces additional complexity and, more importantly, intro-  
duces delays as data professionals consistently need to move or   
copy data between different systems.

Lakehouses are enabled by a new system design using similar data   
structures and data management (DM) features to those in a data   
warehouse, directly on the kind of low-cost object storage used   
for data lakes.

A lakehouse is a new DM architecture that enables users to do   
everything from BI, SQL analytics, data science, and ML on a   
single platform.

Challenges of data lakes:

» Appending data is hard: Users want their changes to   
appear all at once. However, appending new data into the   
data lake while also trying to read it causes data consistency   
issues.   
» Modification of existing data is difficult: You need to be   
able to modify and delete specific records, especially with   
GDPR and CCPA. Unfortunately, it takes a rewrite of   
petabytes on the data lake to make specific changes.   
» Jobs failing mid-way: Job failures usually go undetected   
for weeks or months and aren’t discovered until later when   
you’re trying to access the data and find that some of it’s   
missing.   
» Real-time operations are hard: Combining real-time   
operations and batch leads to inconsistencies because data   
lakes don’t support transactions.   
» It’s costly to keep historical data versions: Regulated   
organizations need to keep many versions of their data for   
auditing and governance reasons. They manually make a lot   
of copies of the data, which is time intensive and costly.   
» Data lakes make it difficult to handle large metadata:   
If you have petabytes of data in the data lake, then the   
metadata itself becomes gigabytes and most data catalogs   
can’t support those sizes.   
» You have “too many files” problems: Because data lakes   
are file-based, you can end up with millions of tiny files or a   
few gigantic files. In either case, this impacts performance   
negatively.   
» Data lakes perform poorly: It’s hard to get great perfor-  
mance with big data. You have to use a number of manual   
techniques like partitioning that are error-prone.   
» You may have data quality issues: All the challenges   
eventually lead to data quality issues. It becomes harder   
to ensure that your data is correct and clean.

Real-time operations are consistent, and the historical data   
versions are automatically stored. The lakehouse also   
provides snapshots of data to allow developers to easily   
access and revert to earlier versions for audits, rollbacks, or   
experiment reproductions.

Lakehouse architecture treats   
metadata just like data, leveraging Apache Spark’s distrib-  
uted processing power to handle all its metadata. As a result,   
it can handle petabyte-scale tables with billions of partitions   
and files with ease.

Schema validation: All your data that goes into a table must   
adhere strictly to a defined schema. If data doesn’t satisfy   
the schema, it’s moved into a quarantine where you can   
examine it later and resolve the issues.

Scalability:

When analyzing the data warehouse approach, you soon realize   
that it comes with clustered or coupled storage, and the compute   
resources don’t scale. Clustered or coupled storage refers to the   
use of two or more storage servers working together to increase   
performance, capacity, or reliability.

Clustering distributes workloads to each server, manages the   
transfer of workloads between servers and provides access to   
all files from any server regardless of the physical location of the   
file. This should be compared to the data lake and the lakehouse,   
which are both highly scalable and use low-cost scalable storage   
and on-demand elastic compute.

The data warehouse approach   
isn’t future-proof because it’s missing support for predictions,   
real-time (streaming) data, flexible scalability, and managing   
raw data in any format.

The data lake approach, on the other hand, may, at a first glance,   
look almost the same as the lakehouse approach with its support   
for low operational cost, flexibility, scalability, and allowing stor-  
age of the raw data in any format needed for ML. But it has several   
drawbacks

A lakehouse enables business analytics and ML at a massive scale.

Solving Problems with a Lakehouse:

» Unifying data teams: One of the biggest benefits of a   
lakehouse is that it unifies all your data teams — data   
engineers, data scientists, and analysts — on one   
architecture.   
» Breaking data silos: A lakehouse approach facilitates   
breaking data silos by providing a complete and firm copy of   
all your data in a centralized location. This enables everyone   
in your organization to access and manage both structured   
and unstructured data.   
» Preventing data from becoming stale: In a continuous   
manner, the lakehouse approach can process batch and   
streaming data, updating tables and dashboards in near real   
time so your data is always generating value, staying   
updated, and never becoming stale.   
» Reducing the risk of vendor lock-in: The lakehouse approach   
uses open formats and open standards that allow your data to   
be stored independent of the tools you currently use to process   
it, making it easy at any time to move your data to a different   
vendor or technology.

With the lakehouse approach, you can build reliable data lakes   
by unifying data pipelines across both batch and streaming   
data.

In this setup, you experi-  
ence reduced compute times and costs with a scalable cloud run-  
time. This process is powered by highly optimized Spark clusters   
and elastic cloud resources that can intelligently auto-scale up   
with increased workloads and auto-scale down for cost savings.

The lakehouse approach to data pipelines offers modern data   
engineering best practices for improved productivity, system   
stability, and data reliability, including streaming data to enable   
reliable real-time analytics.

Delta Lake is an open-source storage layer that brings data reli-  
ability to your existing data lake by providing.   
» Atomicity, consistency, isolation, durability (ACID)   
transactions   
ACID transactions ensure that multiple data pipelines can   
simultaneously read and write data reliably on the same table.   
» Scalable metadata handling   
» Unified streaming and batch data processing   
FIGURE 3-2: An example of an efficient data pipeline setup as part of a   
lakehouse approach.

Everything you can do to simplify the ML life cycle is something   
to strive for. However, the reality is that most companies are   
stuck in organizational and technological silos that are difficult   
to break out of. The sheer amount and diversity of ML frame-  
works needed also makes it hard to manage ML environments.   
The disparate tools and process steps, from data preparation to

18 The Data Lakehouse Platform For Dummies, Databricks Special Edition These materials are © 2022 John Wiley & Sons, Inc. Any dissemination, distribution, or unauthorized use is strictly prohibited.   
experimentation and production, make handoffs difficult to man-  
age efficiently between teams. Due to the data dependency and   
sometimes lack of model transparency, there is also a built-in   
risk from a security and compliance perspective. In ML, it’s hard   
to track experiments, models, dependencies, and artifacts, which   
makes it hard to reproduce results.

Because today’s analytics use cases range from building simple   
SQL reports to more advanced machine learning (ML) predictions,   
you need to build a central data lake in an open format with data   
from all your data sources and make it accessible for various use   
cases.

A lakehouse utilizes inexpensive cloud object storage as the data   
storage layer, which is capable of scaling to virtually any size   
at low cost. For example, you can easily set this up by creating   
an AWS S3 Bucket or a Microsoft Azure Data Lake Storage Gen2   
repository. To move over your data from current applications,   
databases, data warehouses, and other data stores, you can use   
Databricks Ingest, a service that quickly and easily loads data into   
your lakehouse.

Delta Lake addresses the data reliability problems that have   
plagued data lakes, making them data swamps. The opensource   
storage layer that Delta Lake provides brings improved reliability   
to data lakes. Delta Lake on Databricks allows you to configure   
data lakes based on your workload patterns and provides opti   
mized layouts and indexes for fast, interactive queries and sits on   
top of object storage. The format and the compute layer help sim   
plify building big data pipelines and increase the overall efficiency   
of your pipelines.

This pattern of building a central, reliable, and efficient single   
source of truth for data in an open format for use cases ranging   
from BI to ML with decoupled storage and compute is the founda   
tion of the lakehouse approach.

However, it’s important to build data reliability into a lakehouse   
from the get-go to prevent downstream data corruption issues. In   
general, you need to manage two data ingestion scenarios:   
» Data ingestion from third-party sources: You typically   
have valuable user data in various internal data sources.   
Databricks Data Ingestion Network enables an automated   
way to populate your lakehouse from hundreds of data   
sources into Delta Lake.   
» Data ingestion from cloud storage: You already have a   
mechanism to pull data from your source into cloud storage.   
As new data arrives in cloud storage, you can load this new   
data by using the Delta Lake Auto-Loader capability in   
Databricks.

Users of a lakehouse enabled by Databricks Unified Data Analytics   
Platform also have access to a variety of standard tools (Spark,   
Python, R, ML libraries) for non-BI workloads like data science   
and ML.

To make ML management more efficient and get it under business   
control, your company needs a solution to orchestrate and manage   
its models in a way that may speed up model deployment without   
losing model governance. Databricks offers this support, which   
provides a business process management solution with support   
for operationalizing ML models. This includes support for model   
build, register, test, compare, approve, publish, monitor, and, if   
needed retraining those models in an automated and controlled

CHAPTER 4 Building a Modern Cloud Data Platform with Databricks 23 These materials are © 2022 John Wiley & Sons, Inc. Any dissemination, distribution, or unauthorized use is strictly prohibited.   
manner at the same time. This automated architecture is a build   
one, usemany solution that reduces manual human interven   
tion and accelerates customer capabilities of operationalizing ML   
models.

This automated architecture is a build   
one, usemany solution that reduces manual human interven   
tion and accelerates customer capabilities of operationalizing ML   
models. This is achieved through MLflow, an open-source plat   
form developed by Databricks to help manage the complete ML   
life cycle with enterprise reliability, security, and scale.

Reduces cost: With the lakehouse approach, you have one   
system for data warehousing and ML. Multiple systems for   
different analytics use cases are eliminated. You can store   
data in cheap object storage such as Amazon S3, Azure Blob   
Storage, and so on.

Handles security: Data related security challenges are   
easier to handle with a simplified data flow and single source   
of truth approach.