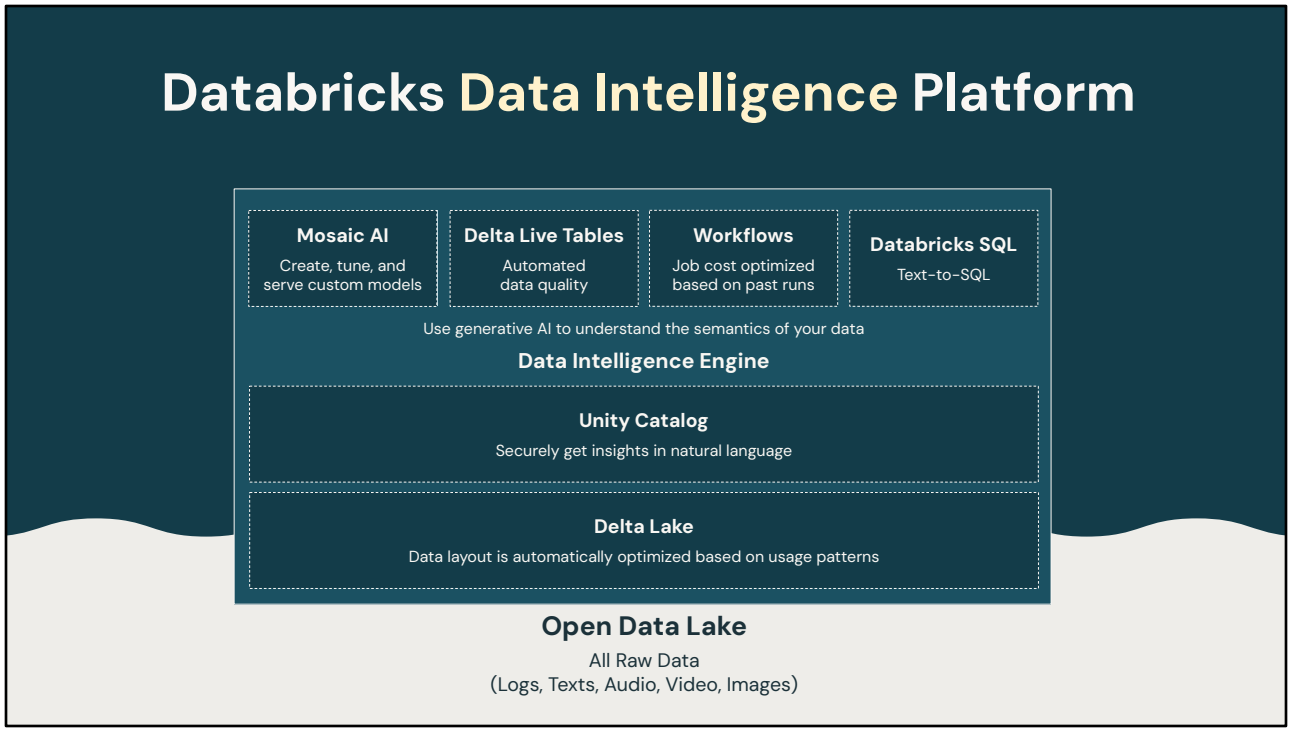
databricks

Managing and Exploring Data

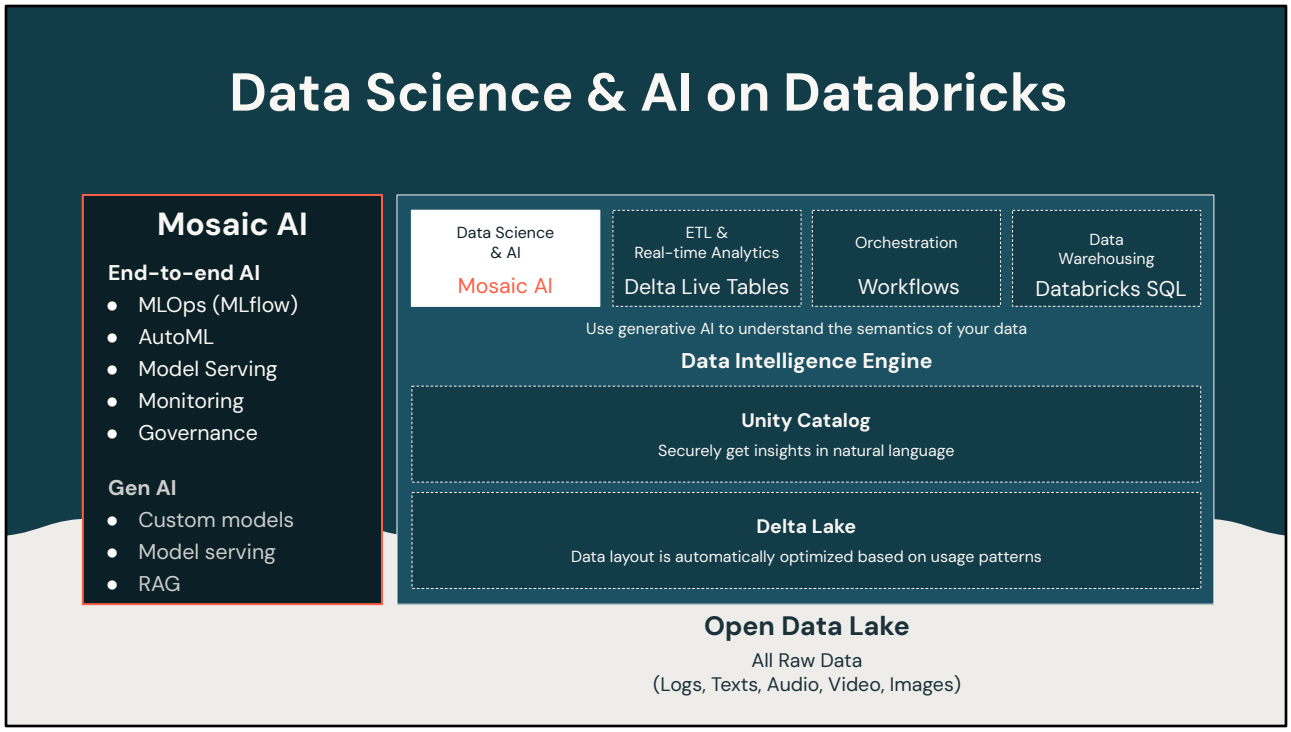
LECTURE

Managing and Exploring Data in the LakeHouse

In this lecture, we cover managing and exploring data in the Lakehouse, highlighting the Databricks Data Intelligence Platform, collaborative notebooks, quick exploratory analysis, tools for ML model development, and the challenges of modern data and AI governance.



If you've seen blog posts, other Databricks Academy courses, or even Databricks documentation pages, you've probably seen an image similar to the one shown on this slide.

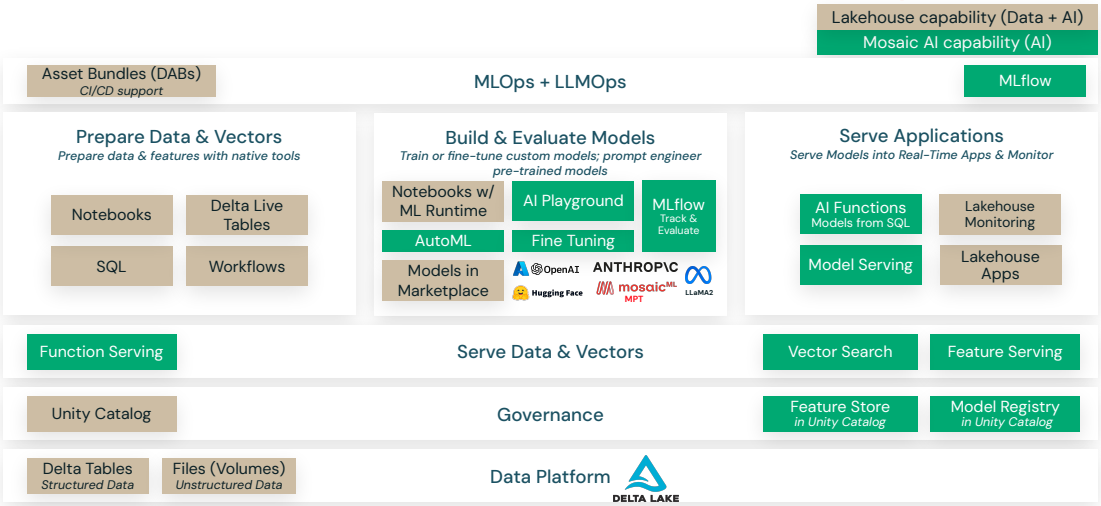


Today, we're going to focus in on the upper-left area - Mosaic AI

Databricks's lakehouse platform supports the complete AI workflow, assisting you in deploying and managing your models throughout their lifecycle in production. Our AutoML feature transparently helps in identifying the optimal model swiftly, without compromising your control over the process. We also offer model monitoring to provide insights into their performance over time, and governance features to ensure reproducibility and adherence to compliance standards. The icing on the cake? It's all built on the open foundation of MLflow, ensuring seamless integration with the broader AI ecosystem.

Databricks for Machine Learning

A data-native and collaborative solution for the full ML lifecycle



Expanding a little bit on the AI capabilities, here's a complete list of platform features which can be used throughout the ML development lifecycle

Features of Databricks for Machine Learning

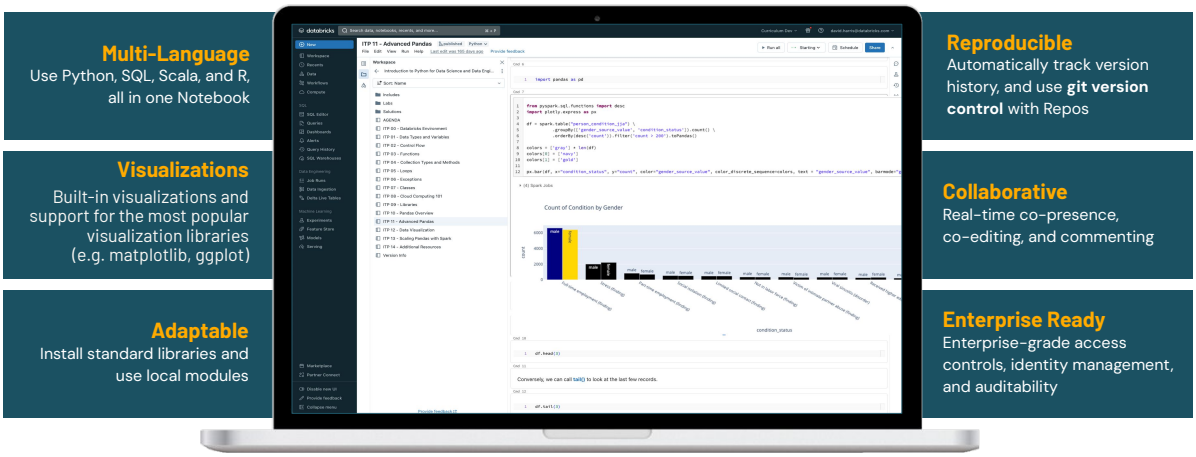
Non exhaustive list of features that will be used throughout this module

- **Collaborative notebooks**
- **ML Runtime**
- **Governance of Data & Models *(via Unity Catalog)***
- **Feature Store**
- Managed MLflow
- Model Serving
- AutoML

ML optimized runtime with pre-installed ML libraries and support for GPUs.

Collaborative Multi-Language Notebooks

Collaborative, reproducible, and enterprise ready

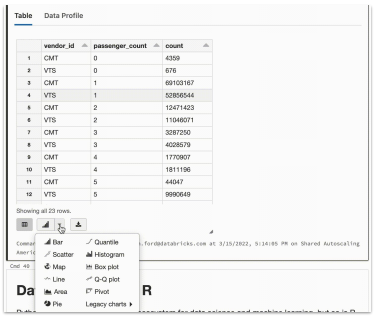


"Notebooks are a widely-used tool in data science and machine learning on Databricks, serving for tasks like data exploration and model training. These notebooks are a central tool for creating data science workflows and facilitating collaboration among team members. Databricks notebooks offer real-time coauthoring capabilities across multiple programming languages, automatic versioning, and built-in data visualization and profiling.

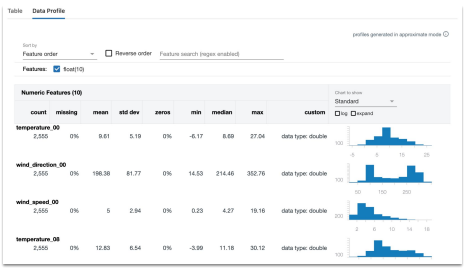
Key features of Databricks notebooks include:
1. **Interactive Coding with Multiple Languages**: Notebooks support interactive coding using various programming languages, including Python, Scala, R, and SQL.
2. **Visualizations and Data Profiling**: Easily conduct exploratory data analysis with the ability to create visualizations and access data profiling tools, enhancing your understanding of both data and model results.
3. **Collaborative Environment**: Databricks notebooks provide a collaborative workspace where multiple users can work simultaneously, fostering teamwork and efficiency.
4. **Enterprise-Ready**: These notebooks are enterprise-grade, offering robust access controls, identity management, and auditability, ensuring security and compliance.
5. **Structured Workflows**: Notebooks help structure data workflows by allowing users to combine code, visualizations, and narrative text, facilitating organized and understandable workflows."

Quick Exploratory Data Analysis

Native tools for visualizing and understanding data in ML workflow



Create **interactive charts** to visualize data in the Notebook with only two clicks



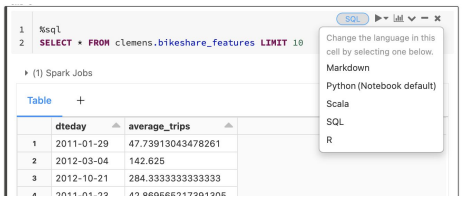
Summarize a data set's essential properties and statistics in a **data profile** with the push of a button

"Visualizations in Databricks notebooks provide a diverse set of tools to craft informative and interactive data representations. These visualizations empower users to reveal patterns, trends, and relationships within their data, enhancing the ability to make data-driven decisions and convey insights effectively.

The Data Profiler in notebooks simplifies data analysis by summarizing crucial properties and statistics of a dataset with a single click. It facilitates exploratory data analysis, making it easily accessible."

Tools for Quick ML Model Development

Multi-language support, use standard libraries and custom modules



Mix and match languages based on use case and preferred workflow, choosing from **Python, SQL, Scala, and R**

```
%pip install folium seaborn==0.11.1
```

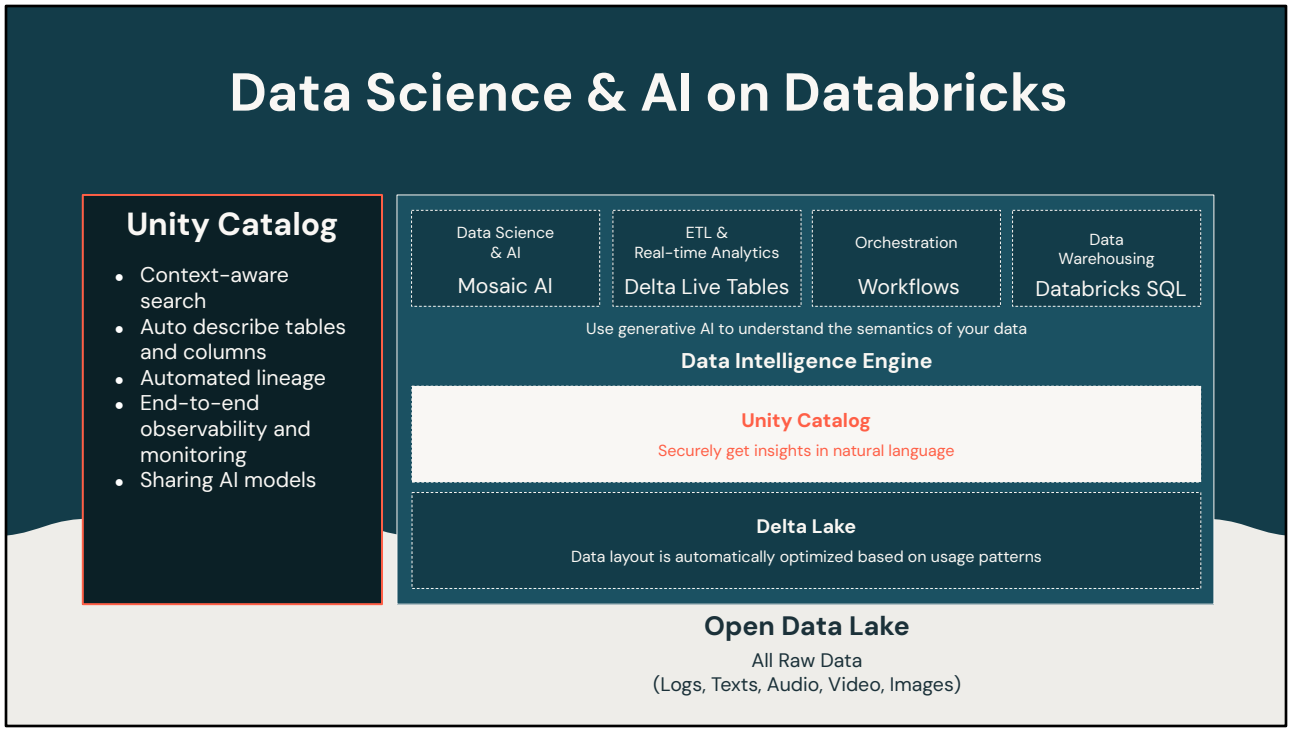
```
import utils

df2 = utils.scrub(df1, drop="num_columns")
```

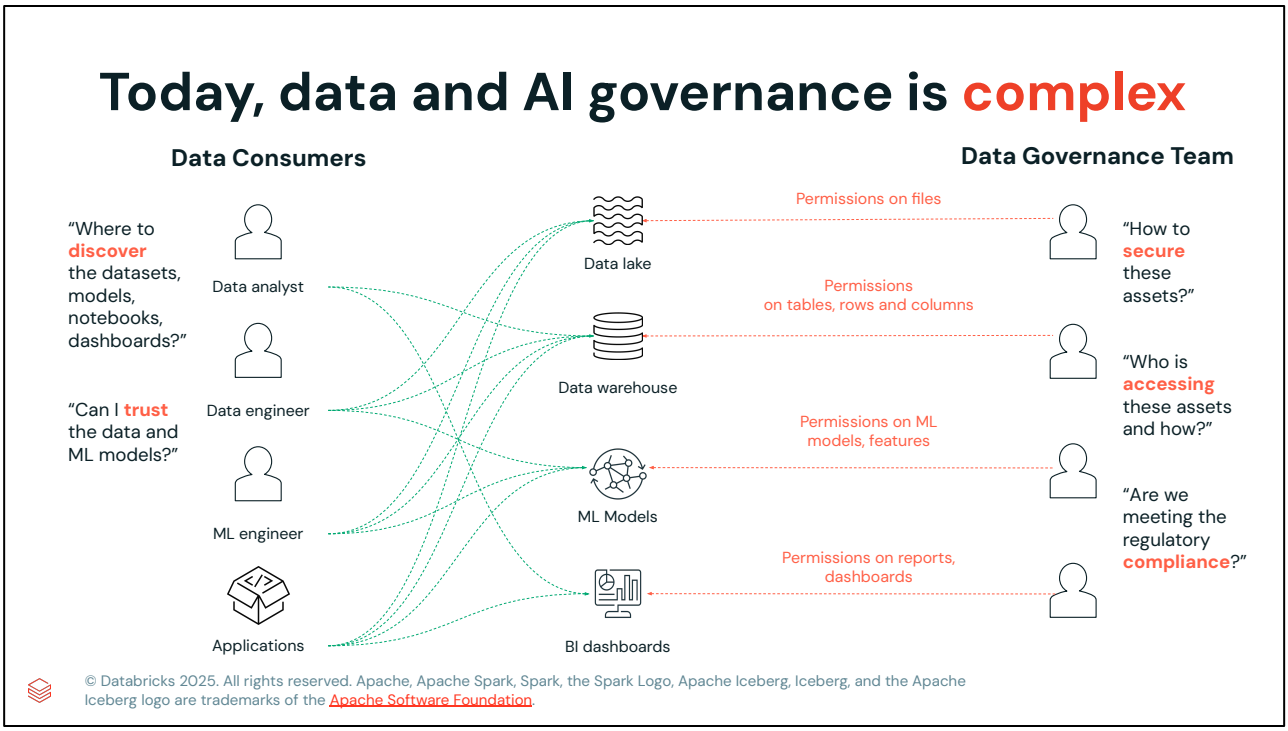
Install Python libraries for a notebook without affecting other users with *%pip*

Import local modules using **arbitrary file support** when working in Repos

"Notebooks are highly versatile, making them ideal for swift development, whether in the context of data engineering or ML tasks.
You have the flexibility to combine various supported languages depending on the specific use case and your preferred workflow.
Additionally, you can install and utilize third-party libraries from external sources and import and employ local libraries as needed."

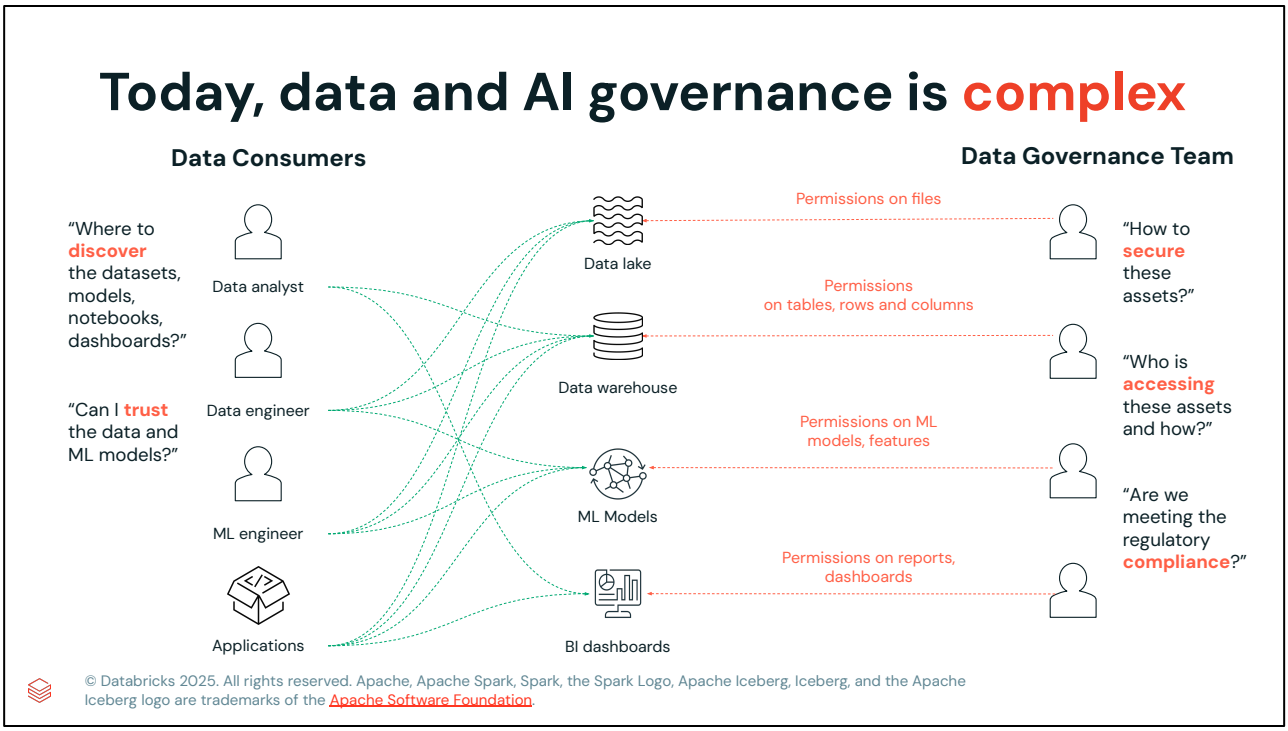


Now let's do a quick intro on why UC exists and what it consists of



In organizations, a diverse array of users collaborates on a wide range of data and AI assets, each requiring specialized tools. These tools often come with distinct governance frameworks, creating a complex web of permissions and security considerations.

Imagine a common machine learning use case: you start by needing permissions to access entire files, such as images or documents, within your data lake. This step seems straightforward; however, when your data scientists harness these files to build a machine learning model, a new layer of governance comes into play. Now, you must ensure the secure and compliant management of the model's lifecycle.



Moreover, the output generated by this model - its metrics and predictions - needs to be securely shared with regional analysts, each of whom may require fine-grained permissions to access specific rows and columns of data. As analysts curate key performance indicators (KPIs), they often seek to share these insights with business users, opening another realm of permissions related to dashboards and reports.

This multi-layered governance and security landscape can result in numerous open questions and uncertainty across the organization regarding the status of data and AI assets. On one side, data consumers face challenges accessing data scattered across various sources to meet their analytical and AI needs. Meanwhile, governance and security teams work tirelessly to ensure data integrity, security, and compliance across these disparate platforms, further complicating the picture.

Unity Catalog (UC)

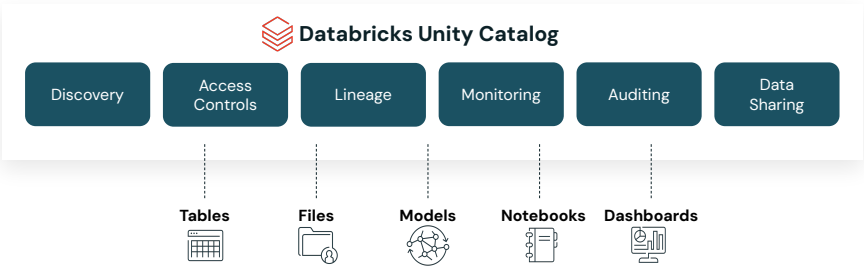
Unified governance for AI; data, code, models

Unified visibility into data and AI

Single permission model for data and AI

AI-powered monitoring and observability

Open data sharing



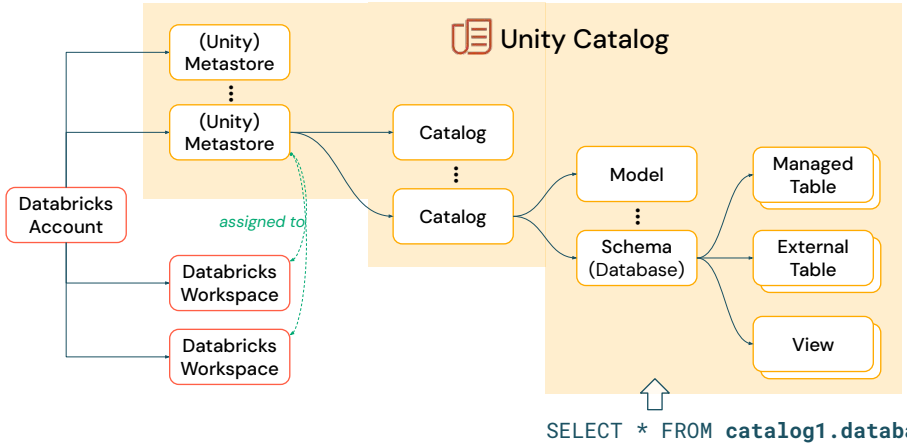
Databricks, through Unity Catalog (UC), offers a comprehensive solution to tackle these challenges, providing unified governance for all data and AI assets, including tables, files, notebooks, dashboards, features, and models.

UC offers significant advantages:

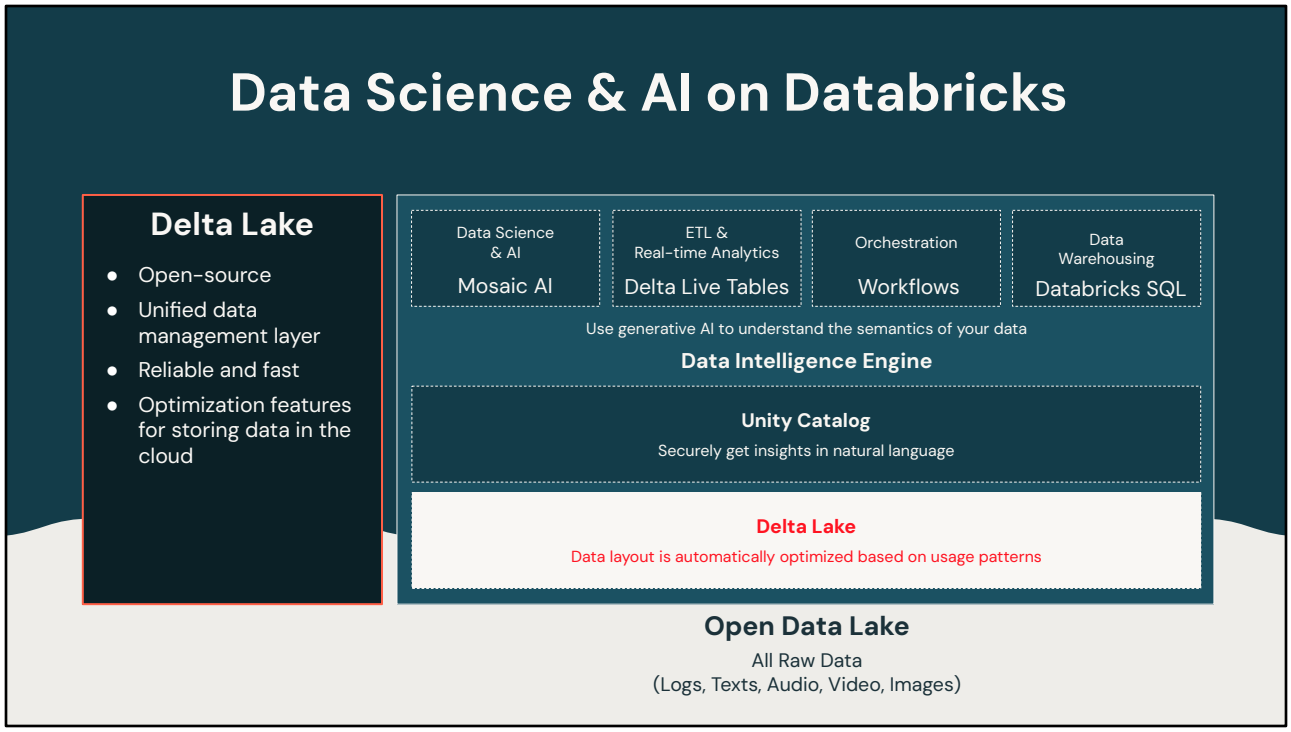
1. **Unified Visibility:** UC provides a centralized platform for managing both data and AI assets.
2. **Single Permission Model:** It simplifies permissions with a consistent model for all assets.
3. **AI-Powered Monitoring:** UC uses AI for robust asset monitoring and observability.
4. **Open Data Sharing:** It encourages collaborative sharing while maintaining governance and security.

The 3-level Namespace of UC

How to use UC



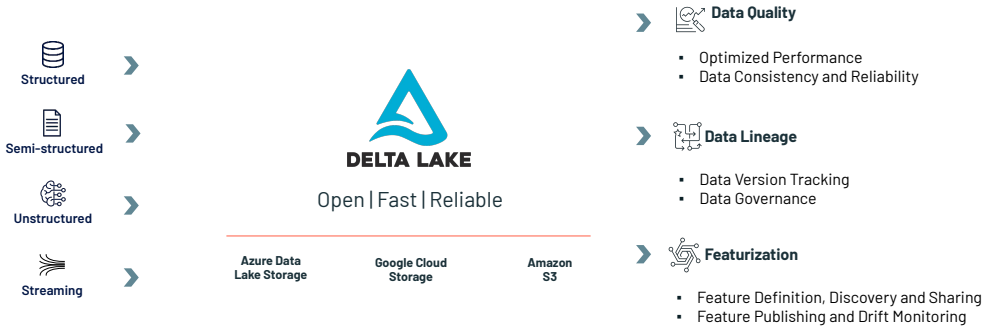
When using UC, it's essential to grasp the concept of a "Catalog" within a Metastore. All assets, including tables and models, are associated with a catalog. When accessing these assets, a three-level namespace is employed. For instance, to query data from a table, the syntax would be "SELECT * FROM catalog1.database1.table1," specifying the catalog and database to which it belongs.



The foundational building block is Delta Lake

What is Delta Lake?

- A **unified data management layer** that brings data reliability and fast analytics to cloud data lakes. It is the optimized storage layer that provides the foundation for storing data and tables in the Databricks DI Platform.



Delta Lake runs on top of existing data lakes and is fully compatible with Apache Spark™ APIs

Delta Lake and Its Features

Open-source, default storage format on Databricks

- Delta Lake is an **open-source** project.
- It is the **default format** for the tables created in Databricks.
- Delta Lake **optimizes performance** with large datasets, providing **ACID transactions** and **scalable metadata handling**.
- Designed to improve data reliability, quality, and performance in data lakes.

Let's break down the key features of Delta Lake and its significance:

- Delta Lake is an Open-Source Project:
  - Delta Lake is a community-driven open-source project, allowing users to leverage and contribute to its development. This collaborative approach fosters innovation and ensures transparency in its evolution.
- Default Storage Format for Databricks Tables:
  - Databricks, a popular analytics platform, has adopted Delta Lake as its default storage format for tables. This integration streamlines data management processes within the Databricks environment.
- Optimizes Performance with Large Datasets:
  - Delta Lake is designed to handle large datasets efficiently. By optimizing performance, it ensures faster query execution and data processing, which is crucial for organizations dealing with substantial data volumes.

Delta Lake and Its Features

Open-source, default storage format on Databricks

- Delta Lake is an **open-source** project.
- It is the **default format** for the tables created in Databricks.
- Delta Lake **optimizes performance** with large datasets, providing **ACID transactions** and **scalable metadata handling**.
- Designed to improve data reliability, quality, and performance in data lakes.

- Provides ACID Transactions:
  - ACID (Atomicity, Consistency, Isolation, Durability) transactions are a critical feature of Delta Lake. This ensures data consistency and reliability even in the face of failures, making it suitable for mission-critical applications.
- Scalable Metadata Handling:
  - Delta Lake addresses the challenges of managing metadata at scale. Its architecture allows for scalable and efficient handling of metadata, contributing to improved performance and reliability.
- Improves Data Reliability, Quality, and Performance in Data Lakes:
  - Delta Lake is specifically designed to enhance the overall data reliability, quality, and performance within the context of data lakes. It provides a robust foundation for managing and processing diverse data in a unified and reliable manner.

Understanding these features underscores the role of Delta Lake in providing a comprehensive solution for managing and optimizing data workflows, particularly in environments like Databricks, where seamless integration and performance are essential.

Delta Lake features

Key features

- Unified **batch** and **streaming**
- Automatic **schema validation**
- Support upserts using the merge operation
- Update your table schema without rewriting data.
- Track row-level changes with Change Data Feed
- **Time-travel**; querying previous versions of a table based on version number of timestamp
- **Performance optimization** with data skipping and liquid clustering
- Supports multiple programming languages like Python, Scala, and SQL.

Let's quickly go over some of the most important features of Delta Lake;
Unified Batch and Streaming: Delta Lake seamlessly handles both batch and streaming data processing, providing flexibility in your data pipelines.
Automatic Schema Validation: Delta Lake automatically validates schemas, ensuring data consistency and quality.
Upserts with Merge Operation: You can perform upsert operations on your data using the merge operation, simplifying data updates.
Schema Evolution: Update your table schema without the need to rewrite existing data, making schema changes more efficient.
Change Data Feed: Delta Lake enables tracking of row-level changes, allowing you to monitor data modifications effectively.
Query Previous Versions: You can query previous versions of a table based on version numbers or timestamps, facilitating historical data analysis.
Performance Optimization: Delta Lake offers performance optimization techniques like ZORDER and OPTIMIZE to enhance query speeds.
Multi-Language Support: It supports multiple programming languages such as Python, Scala, and SQL, making it accessible to a broad range of users.
These features make Delta Lake a powerful choice for managing and processing your data effectively.

Thank you for completing this lesson and continuing your journey to develop your skills with us.