



# Data Analysis with Databricks SQL



---

Databricks Academy

2024

# Get access to the lab

If you haven't done so already - please register yourself and get your username

## Registration

<https://bit.ly/3z8DPSC>

The screenshot shows a registration form with a blue header and a white body. The title 'Register Now' is at the top. The form fields include: First Name\*, Last Name\*, Email\*, Organization\*, Country\*, and a checkbox for agreeing to Terms of Service and Privacy Policy. A 'Submit' button is at the bottom.

Lab Description	Environment	Resources
Key	Value	
Databricks SQL Analytics URL	<a href="https://adb-3311722874107344.4.azuredatabricks.net/">https://adb-3311722874107344.4.azuredatabricks.net/</a>	
Username	@databrickslabs.com	
>Password		

WIFI

RESEAU : COMET-JARDIN

PASS : REALMEETINGS2024



# Agenda

Module Name	Duration
Ice breaker and org details	20 mins 13H00 -> 13h20
Intro – Databricks SQL Services and Capabilities	50 mins 13h20 -> 14h10
Unity Catalog in Databricks SQL	50 mins 14h10 -> 15h00
<b>Break / Networking</b>	40 mins 15h00 -> 15h40
Data Visualization and Dashboarding	50 mins 15h40 -> 16h30
Integration with AI	20 min 16h40 -> 17h00
Observability / Monitoring	20 mins 17h10 -> 17h30
<b>Networking</b>	30 mins 17h30 – 18h00

# Your Hosts



**Eugénie Vinet**  
Sr. Solutions Engineer



**Ali Azouz**  
Sr. Technical Service Engineer



**Jérôme Ivain**  
Solutions Architect



**Matthieu Lamairesse**  
Lead Solutions Architect

# Course goals

- 1** Describe how Databricks SQL:
  - Works in the Lakehouse architecture
  - Data layout / table performance best practices
  
- 2** Use Databricks SQL to:
  - Create tables and Query data
  - Create visualizations and dashboards
  - Integrate with AI models



# A few questions

- What do you do ?
  - Data Engineer
  - Data Analysts
  - DB Admin
  - Project Managers
  - Other

# A few questions

- How long have you been working with Databricks ?

Less than 6 months

6 months to a year

1 year to 2 years

2 years or more

# A few questions

- How many of you have worked with DB SQL before ?
- For how long?
  - Less than 6 months
  - 6 months to a year
  - 1 year to 2 years
  - 2 years or more

# A few questions

- What other data analysis tools do you also use ?

# Databricks SQL Services and Capabilities



---

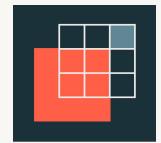
Databricks Academy 2023

# Databricks SQL Services and Capabilities

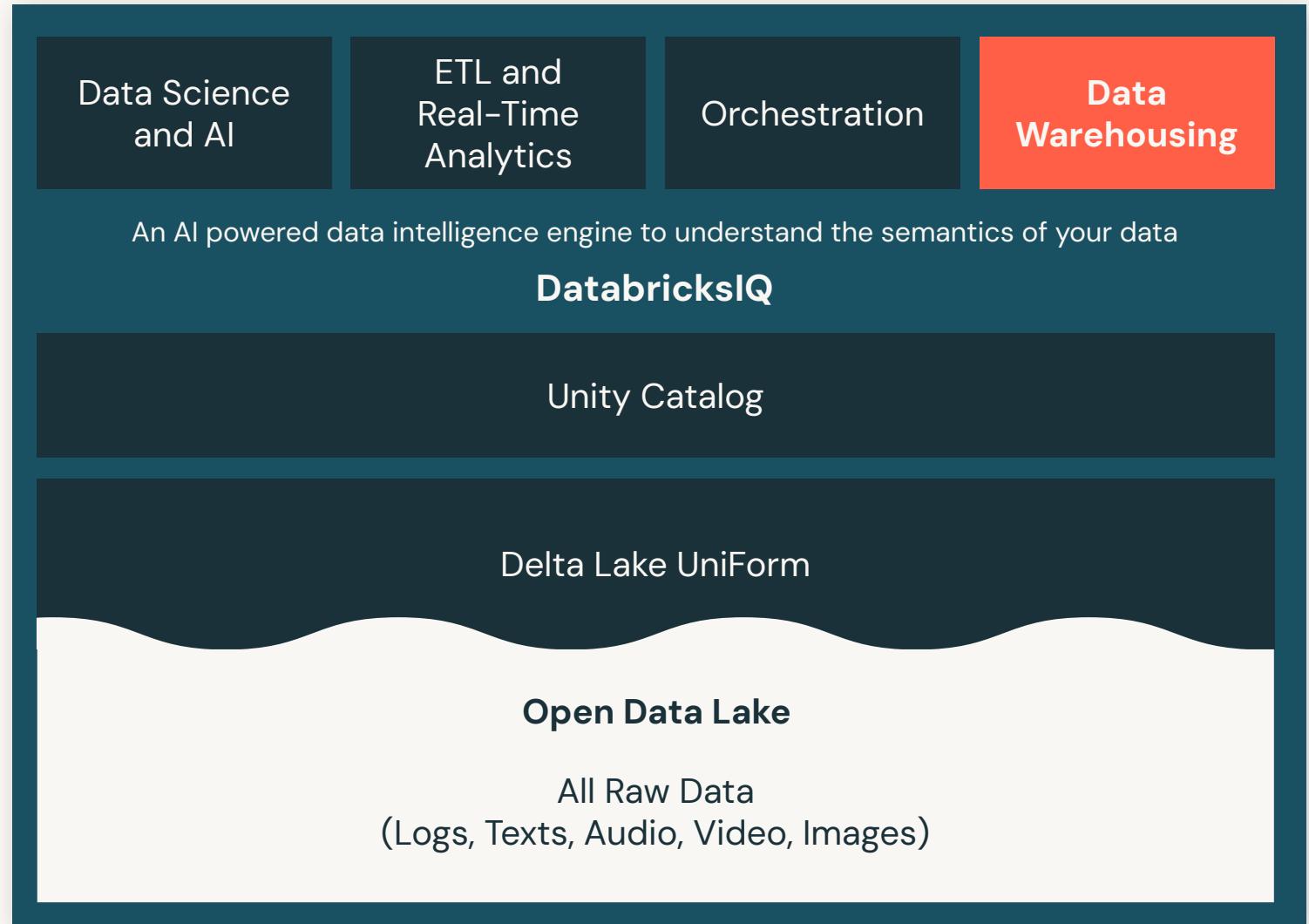
Lesson Name	Duration
Introduction to DB SQL	30 mins
Data Modeling and performance best practices	10 mins



# Introduction to Databricks SQL



**Databricks SQL**  
intelligent data  
warehousing on the  
Data Intelligence Platform



# Databricks SQL

Data warehousing on a lakehouse architecture is now standard

## What Is a Lakehouse?



by Ben Lorica, Michael Armbrust, Reynold Xin, Matei Zaharia and Ali Ghodsi

January 30, 2020 in [Engineering Blog](#)

Share this post



Over the past few years at Databricks, we've seen a new data management architecture that emerged independently across many customers and use cases: [the lakehouse](#). In this post we describe this new architecture and its advantages over previous approaches.

Data warehouses have a [long history](#) in decision support and business intelligence applications. Since its inception in the late 1980s, data warehouse technology continued to evolve and MPP architectures led to systems that were able to handle larger data sizes. But while warehouses were great for structured data, a lot of modern enterprises have to deal with unstructured data, semi-structured data, and data with high variety, velocity, and volume. Data warehouses are not suited for many of these use cases, and they are certainly not the most cost efficient.



2020

Databricks pioneered the lakehouse architecture

Today

**74% of global enterprises have adopted lakehouse**

MIT Technology Review Insights, 2023



# Trusted by organizations of all sizes

7,000+ Databricks SQL customers across industries



Abnormal



AKTIFY

AMGEN

INTUIT



ATLASSIAN



Barilla

ESTÉE LAUDER

yipitDATA



COMPASS



DEVSISTERS

CareSource

grammarly



ExxonMobil

Johnson & Johnson

CONDÉ NAST

Plume

Quartile

VIZIO

GSK

SEGA

T Mobile™

punchh.

FASTNED

edmunds

RIVIAN

vin

SNCF

flipp

DELL

WB

Shell

BAYER

bp

wejo



**AI is changing everything –  
including data warehousing**

# Introducing intelligent data warehousing



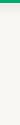
User experiences with  
natural language and  
data intelligence



Predictive optimizations  
for your infrastructure



World-class price  
performance

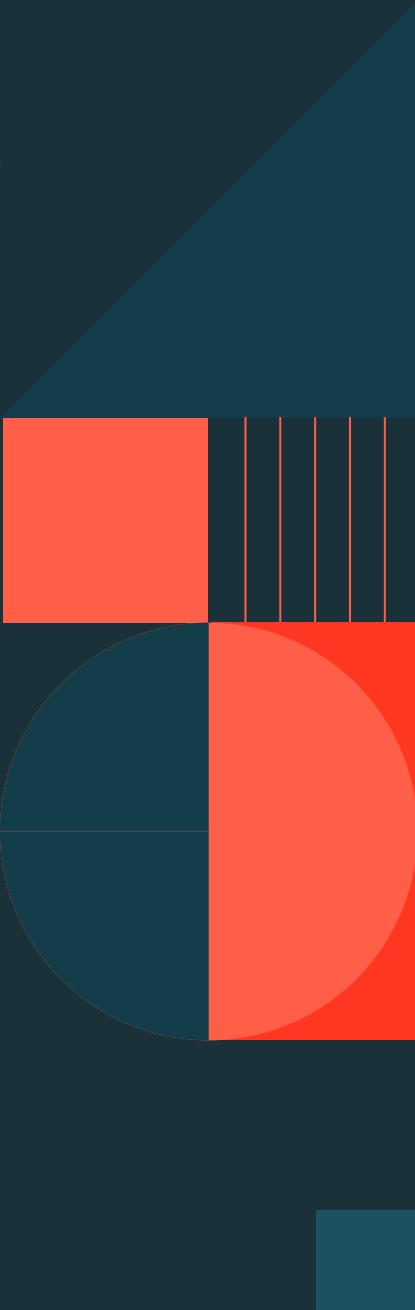


Access for everyone to  
ask questions of their data

Intelligent, automated  
management and tuning

The best TCO  
in the market

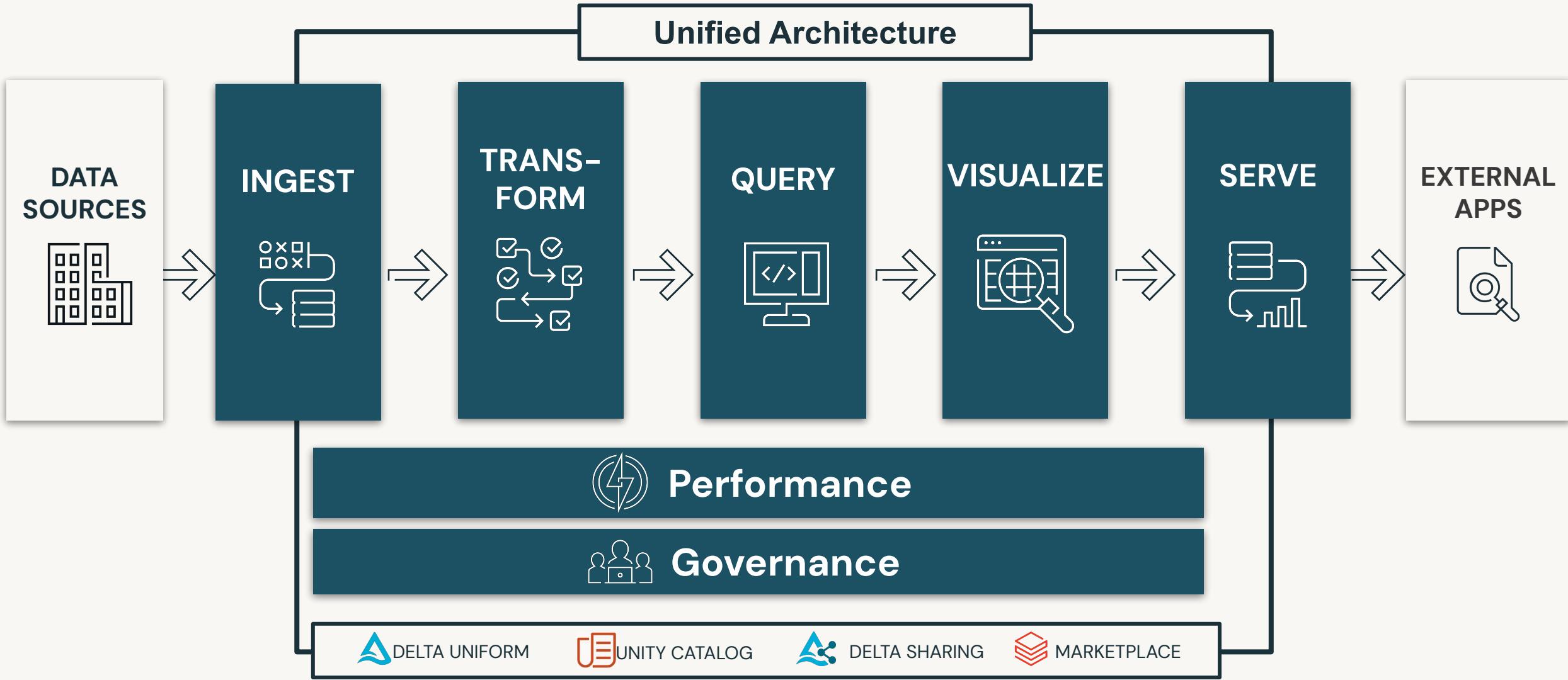
# Intelligent data warehousing with **Databricks SQL**



20



# Complete Data Warehousing Solution



# Complete Language functionality

✓ ANSI SQL

✓ Query **LLMs** and use built-in **AI functions** [Preview]

✓ **Spatial** SQL [Preview]

✓ Data Modeling With **pk/fk Constraints**

✓ **Programming**: SQL UDFs, Variables, Identifiers, Temp Views, Temp Tables\*, Scripting\*, Stored Procedures\*, **Python** UDFs,



# Lakehouse Federation

Discover, query, and govern all your data – no matter where it lives

- **Unified view** into all your data
- **Unified engine** for all your data and use cases
- **Unified governance** across all data sources



# Three reasons why customers love Databricks SQL

- 1 **Intelligent Experiences** with built-in understanding of your data
- 2 **Predictive Optimizations** for all your workloads
- 3 **Best Price/Performance**<sup>24</sup> for the lowest TCO



1

# Intelligent experiences with built-in understanding of your data

25



# Intelligent experiences with natural language

## Powered by DatabricksIQ

The screenshot shows the DatabricksSQL interface. On the left, there's an 'Assistant' panel with a message about accelerating work through diagnostics and suggestions. In the center, a query editor window displays a query to find the top 10 most expensive taxi trips. Below the editor is a results table showing fare amounts from 278.00 down to 105.00. At the bottom, there's a message input field and a note about the query's purpose.

**SQL Editor**

Data Science / Engineers

The screenshot shows the DatabricksIQ dashboard interface. It features a large central canvas area with a grid pattern, intended for placing data visualizations. Above the canvas, the title 'New Dashboard' is visible along with a timestamp and sharing options. A message at the bottom right says 'Select a widget to configure'.

**Dashboards**

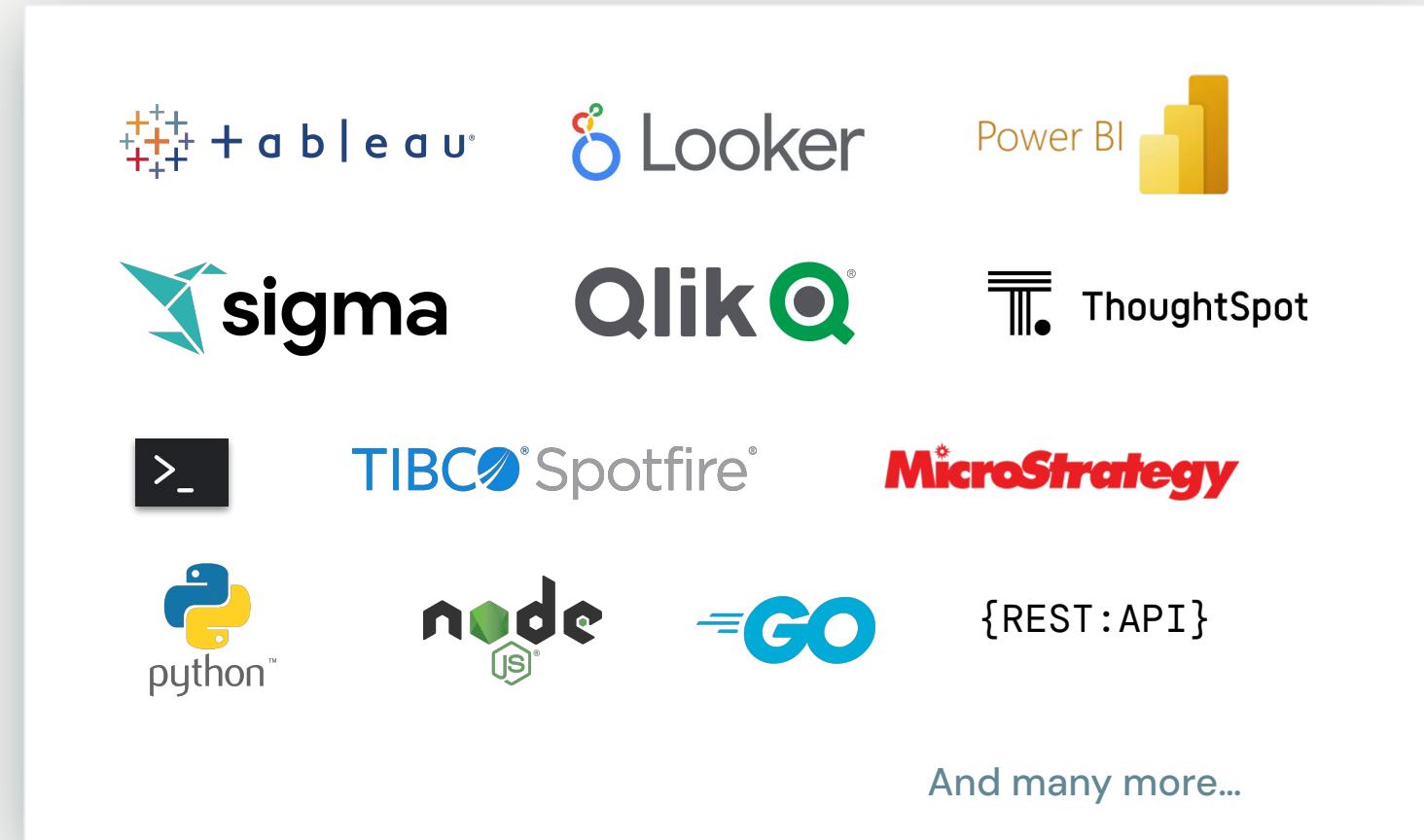
Analysts

The screenshot shows the DatabricksIQ Project Genie interface. It has sections for 'Quick actions' with buttons for 'Example questions', 'Explain data set', and 'Surprise me'. There are also tabs for 'Fast AI' and 'Smart AI'. The overall theme is AI integration into data analysis tasks.

**Project Genie**

Business Users

# First-class integration with leading BI tools

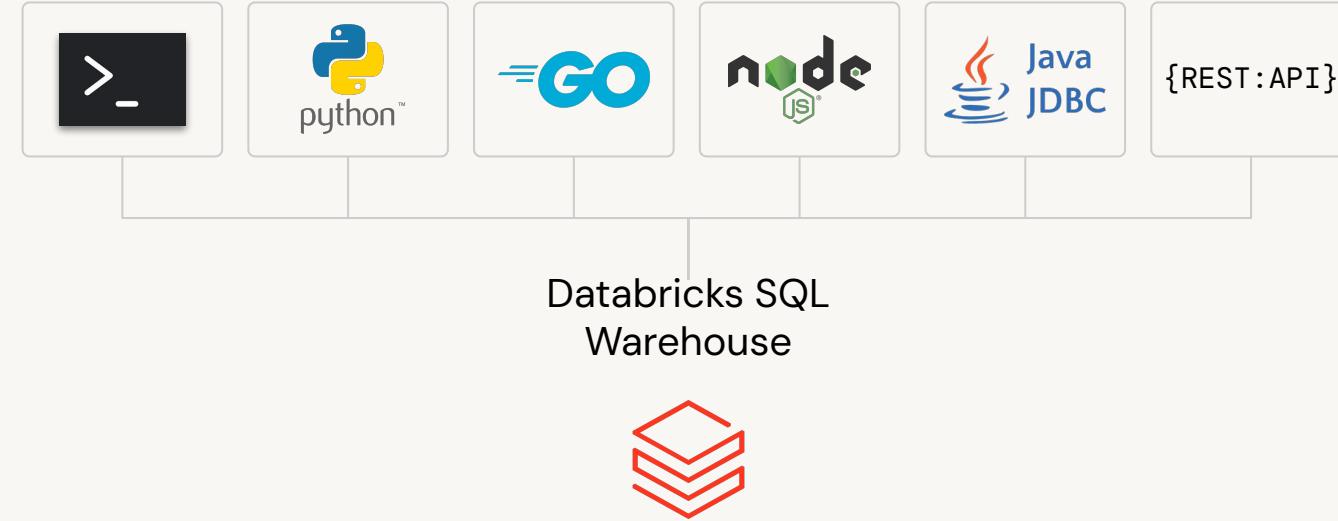


Use your existing BI Tools, SQL Workbenches, or IDEs to discover insights or build custom data apps with Databricks; using familiar tools and languages

# Build Data Apps on Databricks SQL!

Run SQL from anywhere and build custom data applications

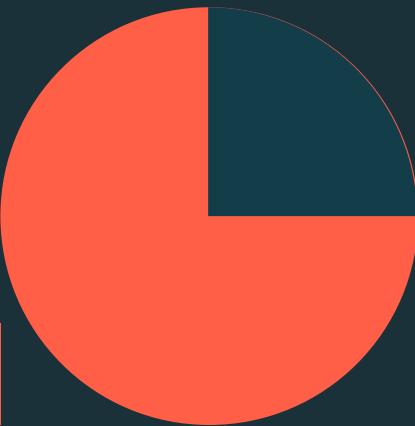
- Developers can interact with Databricks SQL programmatically by automating tasks, integrating with other systems and build custom applications leveraging compute power
- Ability to use Native Go, Node.js as well as CLI and **REST API** makes it accessible to wide range of developers



2

## Predictive optimizations for all your workloads

29



# Predictive optimizations in our engine

Powered by DatabricksIQ

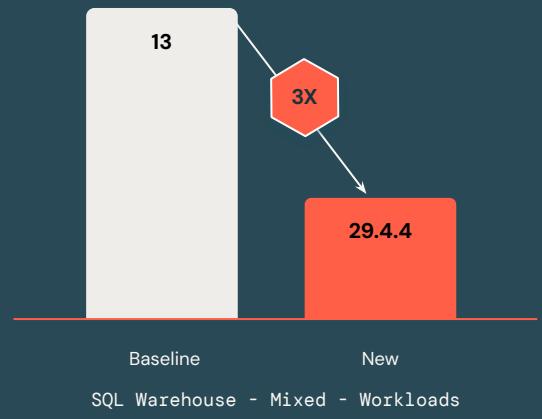


# Predictive optimizations in our engine

## Examples

### Intelligent Workload Management

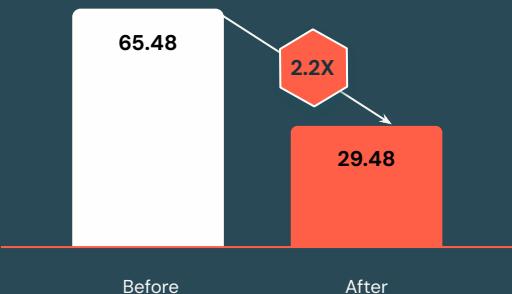
Leverages machine learning to efficiently route queries and scale clusters to maximize cost/performance



### Automatic Data Layout

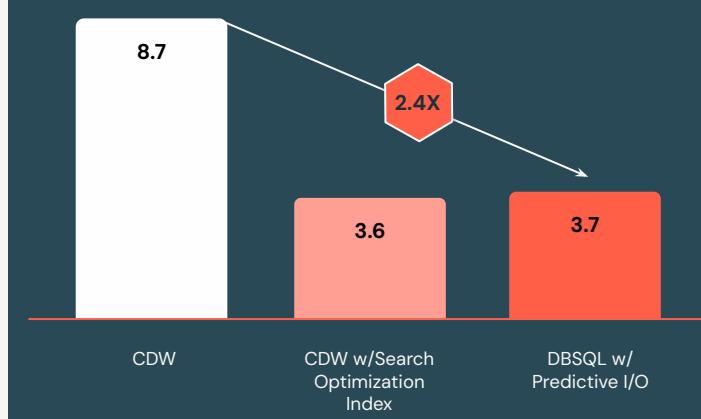
Eliminates knobs to optimize storage with ROI-based table maintenance algorithms

TPC-DCS 1TB – Query timings (minutes)  
Lower is better



### Indexless Indexing

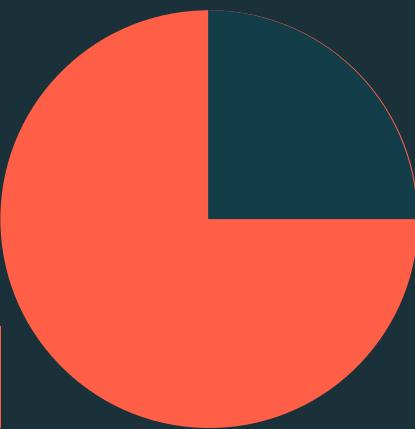
Predictive i/o delivers comparable performance **without** expensive search-optimized indexes



3

## Best price/performance for the lowest TCO

32



# Photon Query Engine

Next-gen, vectorized query engine with world-record performance

Written from the ground up in C++, Photon takes advantage of modern hardware for faster queries, providing TPC world-record price/performance compared to other cloud data warehouses — all natively on your data lake.



SIGMOD 2022  
Best Industry  
Paper Award

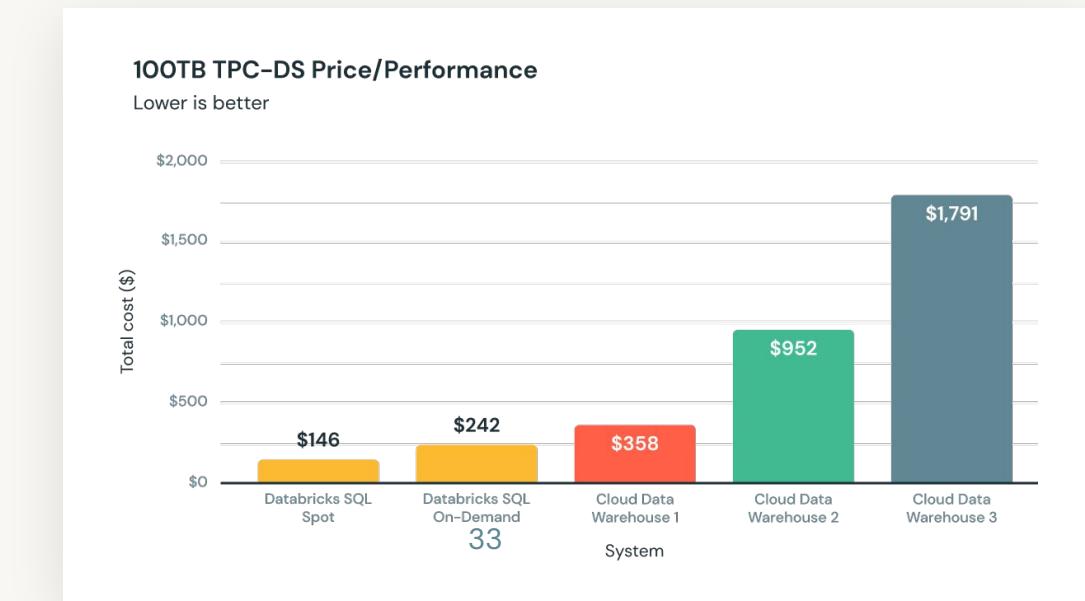
## Photon: A Fast Query Engine for Lakehouse Systems

Alexander Behm, Shounik Falkar, Utkarsh Agarwal, Timothy Armstrong, David Cashman, Ankur Luszczak, Prashanth Menon, Mostafa Mokhtar, Gene Pang, Sameer Paranjpye, Greg Rahn, Bart Samwel, Tom van Bussel, Herman van Hovell, Maryann Xue, Reynold Xin, Matei Zaharia  
photon-paper-authors@databricks.com  
Databricks Inc.

### ABSTRACT

Many organizations are shifting to a data management paradigm called the “Lakehouse” which implements the functionality of structured data warehouses on top of unstructured data lakes. This enables new use cases for query execution engines. The execution engines must support both high performance and raw unstructured data processing. Photon is a fast, efficient, and excellent system for executing queries on the Lakehouse. It uses a columnar storage approach and a vectorized query engine to achieve high performance and low latency. Photon is designed to be highly efficient and scalable, making it suitable for large-scale data processing tasks.

from SQL to machine learning. Traditionally, for the most demanding SQL workloads, enterprises have also moved a curated subset of their data into data warehouses to get high performance, governance and concurrency. However, this two-tier architecture is complex and expensive, as only a subset of data is available in the warehouse, and this data may be out of sync with the raw data due to issues in the extract, transform and load (ETL) process [19]. In response, many organizations are shifting to a data management system called the Lakehouse [19], which implements data storage and processing in a single system. The Lakehouse approach





**Databricks SQL**  
intelligent data  
warehousing on the data  
intelligence platform



**Intelligent experiences  
with built-in understanding of your data**



**Predictive optimizations  
for all your workloads**



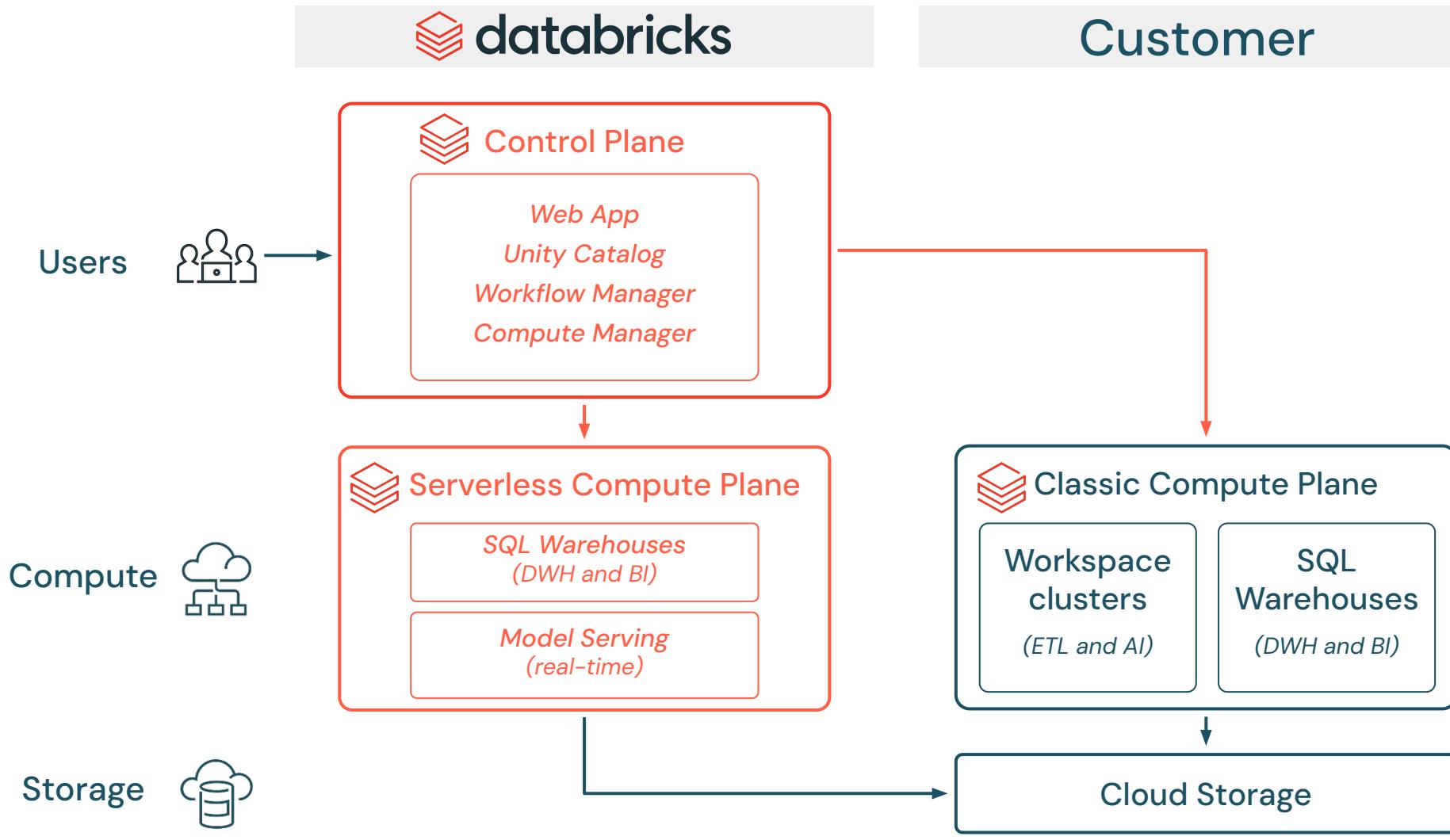
**Best price/performance  
for the lowest TCO**

34

# Databricks Datawarehouses Architecture

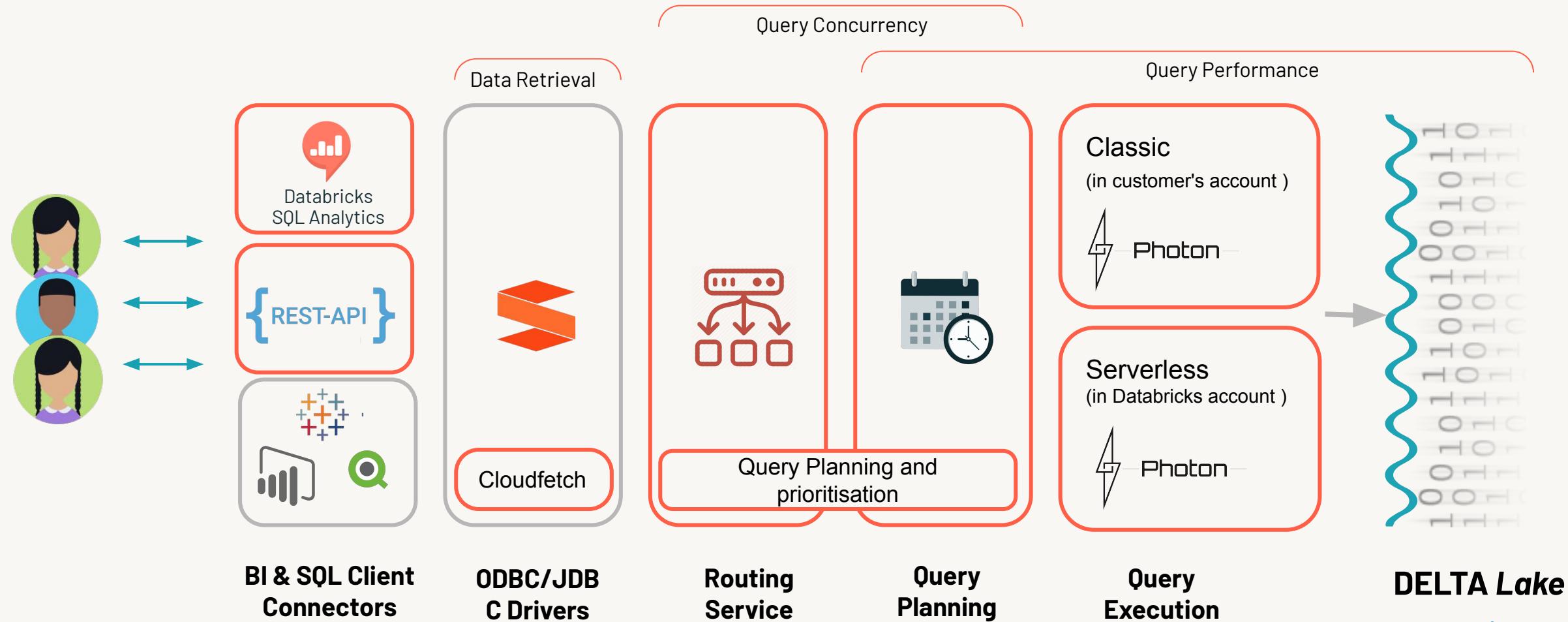


# Databricks Serverless – Overview



# Anatomy of a Query on Databricks SQL

Difference in architecture  
vs "standard" clusters



# Databricks SQL Service offers



# Databricks SQL Pricing / DW Offering

## SQL Classic

### Databricks SQL Classic

\$0.22/DBU  
(+infrastructure)

Self-managed, good for exploratory SQL workloads

“GOOD”

## SQL Pro

### Databricks SQL Pro

\$0.72/DBU  
(+infrastructure)

Self-managed, better performance, unlock new workloads

“BETTER”

## Serverless SQL

### Databricks SQL Serverless

\$0.91/DBU\*

Best for all workloads and performance, fully managed, elastic, best value

“BEST”

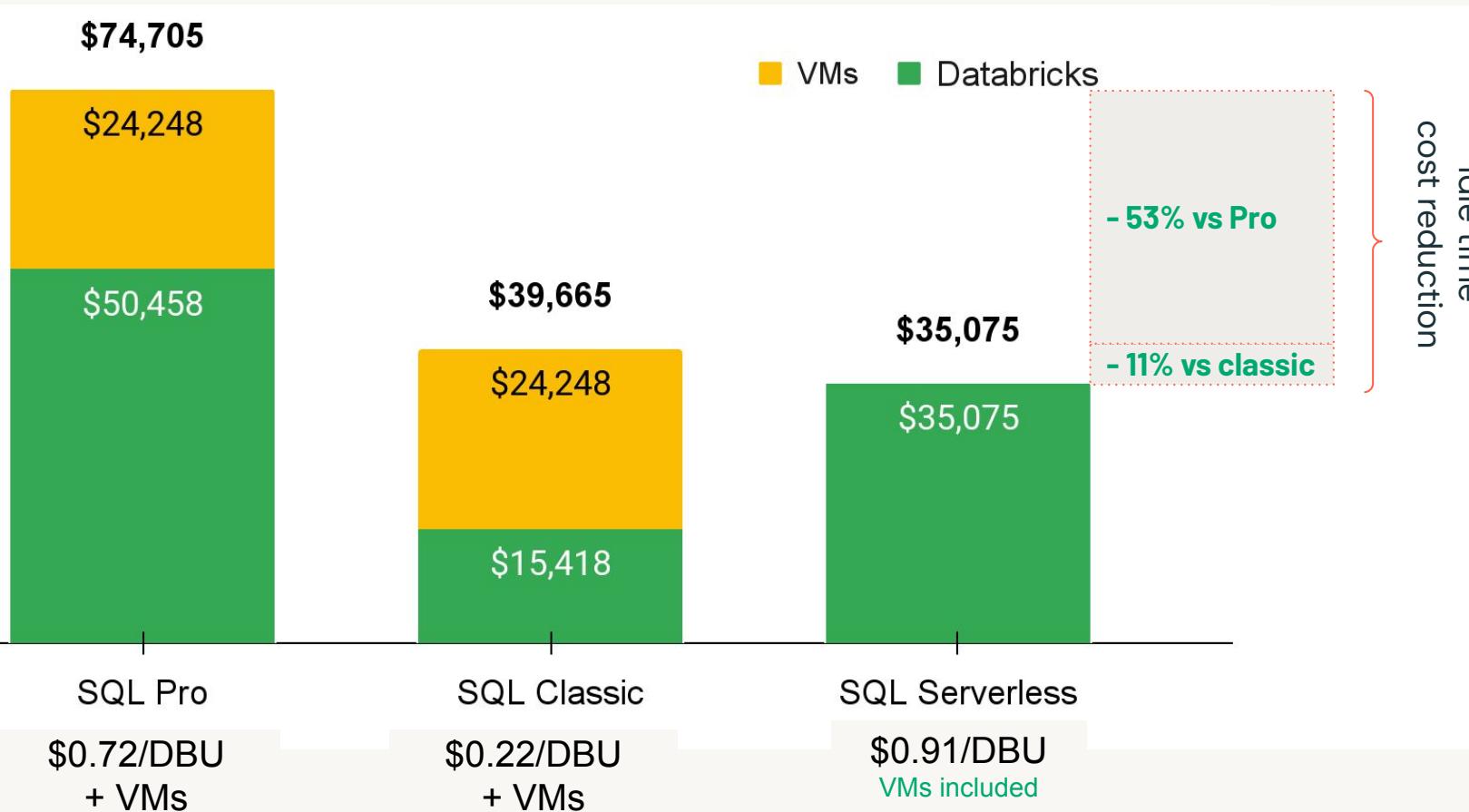
\* Infrastructure (VMs) costs included



Feature Matrix		Classic Self-Managed, Introductory SKU	Pro Self-Managed, Compute in your account	Serverless Fully Managed, Elastic, Best Value
		Good	Better	Best
Exploratory SQL	SQL Editor with intelligent auto complete, ANSI SQL	✓	✓	✓
Management & Governance	Query History & Profile, Data Explorer (Unity Catalog), Managed Data Sharing	✓	✓	✓
Connectivity	<a href="#">SQL Rest API</a> , <a href="#">Python</a> , <a href="#">Node.js</a> , <a href="#">Go</a> , <a href="#">Partner Connect</a>	✓	✓	✓
Performance	<a href="#">Photon Engine</a> (Massively Parallel Processing)	✓	✓	✓
	Predictive I/O	✗	✓	✓
SQL ETL/ELT	<a href="#">Query Federation</a> , <a href="#">Materialized Views</a> , <a href="#">Workflows Integration</a>	✗	✓	✓
Data Science & ML	<a href="#">Python UDFs</a> , <a href="#">Notebooks Integration</a> , <a href="#">Geospatial</a> , <a href="#">LLM AI Functions</a>	✗	✓	✓
Serverless Data Warehouse	<a href="#">Instant</a> , <a href="#">Elastic</a> , <a href="#">Fully Managed Compute</a>	✗	✗	✓
High Concurrency BI	<a href="#">Intelligent Workload Management</a>	✗	✗	✓
	Serverless Query Result Caching*	✗	✗	✓

# Databricks SQL classic vs. serverless – observed cost

Azure - West Europe - On Demand VMs



## Annual costs - Medium Warehouse

Classic/Pro warehouse up 8 h/day 365d/y  
Using on demand VMs

**Assuming a average 55% usage rate** for the Classic/Pro Warehouse

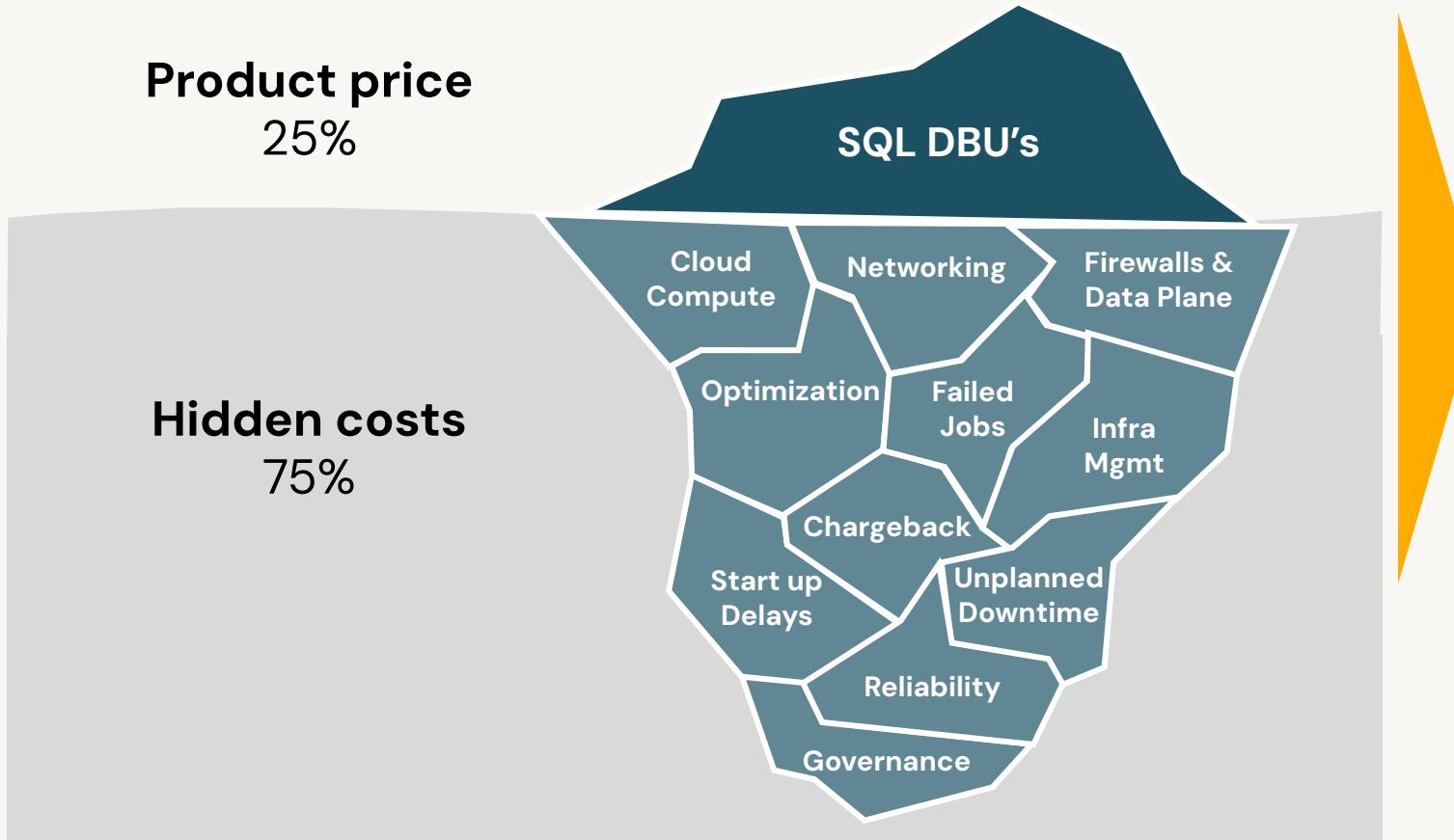
=> Observed average between ( 40 and 60 ) over 1 year





# Serverless SQL delivers more business value than simply reducing compute costs.

## Typical TCO Iceberg – SQL



## 4 Key Value Drivers

### Reduced Infrastructure Costs

- Reduced SQL Run Costs (Compute)
- Reduced Cloud Infra Costs

### Lower Management Overhead

- Lower Chargeback Support
- Reduced Optimisation Efforts
- Reduced Infrastructure Management

### Increased User Productivity

- Reduction in Failed Jobs
- Faster Start-up Time

### Greater Business Impact

- Reliability for Critical Workloads
- Streamlined Governance

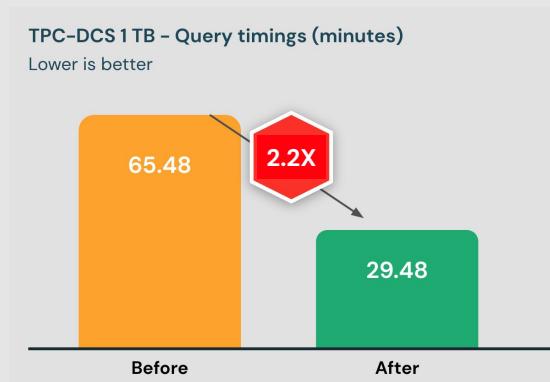


# Serverless focusing our R&D Efforts

Leaps in performance thanks to **AI-Powered features**

## Auto-tuning

automatically optimizes writes and compacts storage of managed tables to reduce latency and cost



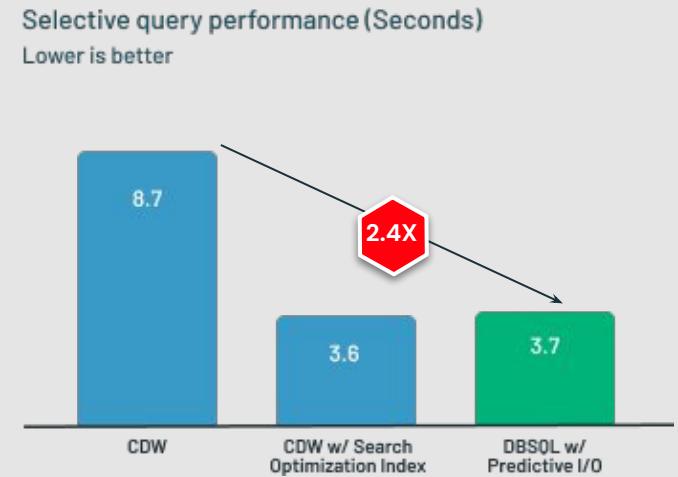
## Intelligent Workload Management

leverages machine learning to efficiently route queries and scale clusters to maximize cost / performance.



## Predictive I/O

delivers comparable performance to expensive search-optimized index automatically



# Roadmap



# New Features

## Intelligent Experiences

-  Dashboards Enhancements (GA)
-  AI Functions (Public Preview)
-  SQL Editor + Assistant Enhancements (Preview)
-  Vector search from SQL ( Public Preview )

## Predictive Optimizations

-  Automatic Statistics (Preview)
-  Intelligent Workload Management – Enhancements (GA)

## Core Warehousing

-  Lakehouse Federation (GA)
-  Materialized Views and Streaming Tables (GA)
-  SQL Scripting (Preview)
-  Publish to PowerBI and Tableau (Preview)
-  Serverless Warehouses on GCP (GA)
-  HIPAA, PCI, FedRamp for Serverless Warehouses on AWS (GA)



# Data Modeling and performance best practices



# Data warehouse sizing



# Data warehouse sizing

**Endpoint Size :** Compute capacity of a single cluster.

Adjust endpoint **Size** to adjust **single query Latency**

**Reco :** Start with **Medium** for typical BI style workloads

Increase size for low latency on larger dataset / complex queries

**X-Large**

1      2

**Endpoint Scaling :** Number of concurrent clusters behind a single warehouse

Adjust **scaling** to increase **query throughput / concurrency**

**Reco:** Start with **MIN = 1 and MAX = 5**

More Users – Smaller Data Size

**Medium**

1      2  
3      4



# Data Layout



# Performance

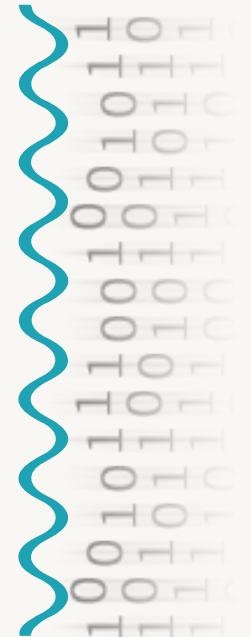
Query Performance

95 % of performances improvements can be addressed by :

- 1. Using an optimized data format for storage**
- 2. Optimized file size**
- 3. Layout adapted to the end user's query patterns**

The rest ( 5% ) can be addressed by :

- **Good SQL code ( majority of cases )**
- Advanced parameter tuning (advanced / edge cases)



# 1. Choose the right file format

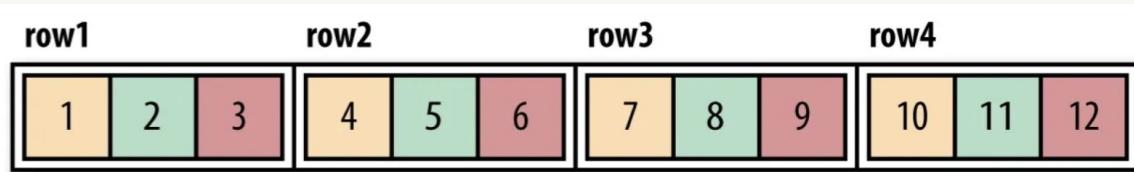
Column oriented offers the best performance for analytics use cases

## Row oriented

Example : Text ; Avro

Data is stored and retrieved one row at a time

- Very difficult to filter ;  
Must read every data point in every lines



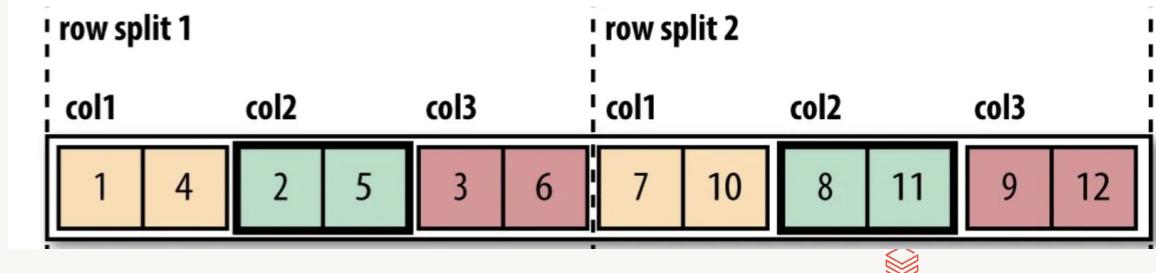
## Column oriented

Example : ORC ; Parquet; **Delta**



Data is stored and read by column

- + Read only the columns required for the lines requested



# Column oriented is not good enough... => Need Transactions



Open format **Delta Lake** is the foundation of the Lakehouse

- Databricks SQL stores and processes data using the **open source format** Delta Lake, based on Parquet
- Delta Lake adds **quality, reliability, and performance** to your existing data lakes
- Provides **one common data management framework** for data, analytics, and AI workloads.

- |                            |  |
|----------------------------|--|
| ✓ <b>ACID Transactions</b> | ✓ Caching                                  |
| ✓ Time Travel              | ✓ Auto-tuning                              |
| ✓ Schema Enforcement       | ✓ Fine-grained, Role-based access controls |
| ✓ Identity Columns         | ✓ Python, SQL, R, Scala Support            |
| ✓ Advanced Indexing        |  |



## 2. Optimize file size

### Fast Data Reading

Right sized files for ideal read parallelism

Too many small files => Large IO overhead

Too few large files => limited parallelism  
=> harder to filter

**Goal of Optimize**

=> Create ideal file size according to table data volume

# 3. Layout adapted to the end user query patterns

## Organize Data according to your query patterns

The less data you have to read the faster the query !

To work, data must be :

- Organised in a predictable manner

That is

- Flexible enough to allow different the use of predicate

**Goal of Z-ORDER / Liquid**

=> Equivalent to indexes on Databases

## Treat Z-Order as you would an index

- Frequent filter columns/predicates
- Foreign Keys

Reco : Limit ZOrder to 4 to 6 columns / dimensions

**With Delta Lake, partitioning is NOT recommended** (except in specific cases\*\*)

Partitioning has a physical reality  
=> Creates folder structure

- partition\_column (Folder)
  - file
  - file

=> Locks table on single effective predicate ( filter )  
and reduces performance on others



# Lab details

# Technical Environment Overview

## The Databricks SQL workspace

- Everyone:
  - is in the same workspace
  - has their own catalog ( you will create them )
  - is using the same SQL warehouse
- Only the instructor has administrator privileges in the workspace
  - only a select few tasks in this course require admin privileges
  - you will see these tasks in the slides in order to provide context
  - the labs do not require admin privileges

# Get access to the lab

If you haven't done so already – please register yourself and get your username

## Registration

<https://bit.ly/3z8DPSC>

Register Now

First Name\*

Last Name\*

Email\*

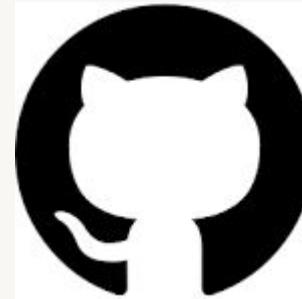
Organization\*  
 Full organization name

Country\*  
 Country

I agree to the Databricks [Terms of Service](#) and acknowledge the Databricks [Privacy Policy](#) (required).

Lab Description	Environment	Resources
Key	Value	
Databricks SQL Analytics URL	<a href="https://adb-3311722874107344.4.azure.databricks.net/">https://adb-3311722874107344.4.azure.databricks.net/</a>	
Username	<input type="text"/>	@databrickslabs.com
Password	<input type="password"/>	<input type="password"/>

## Lab Material

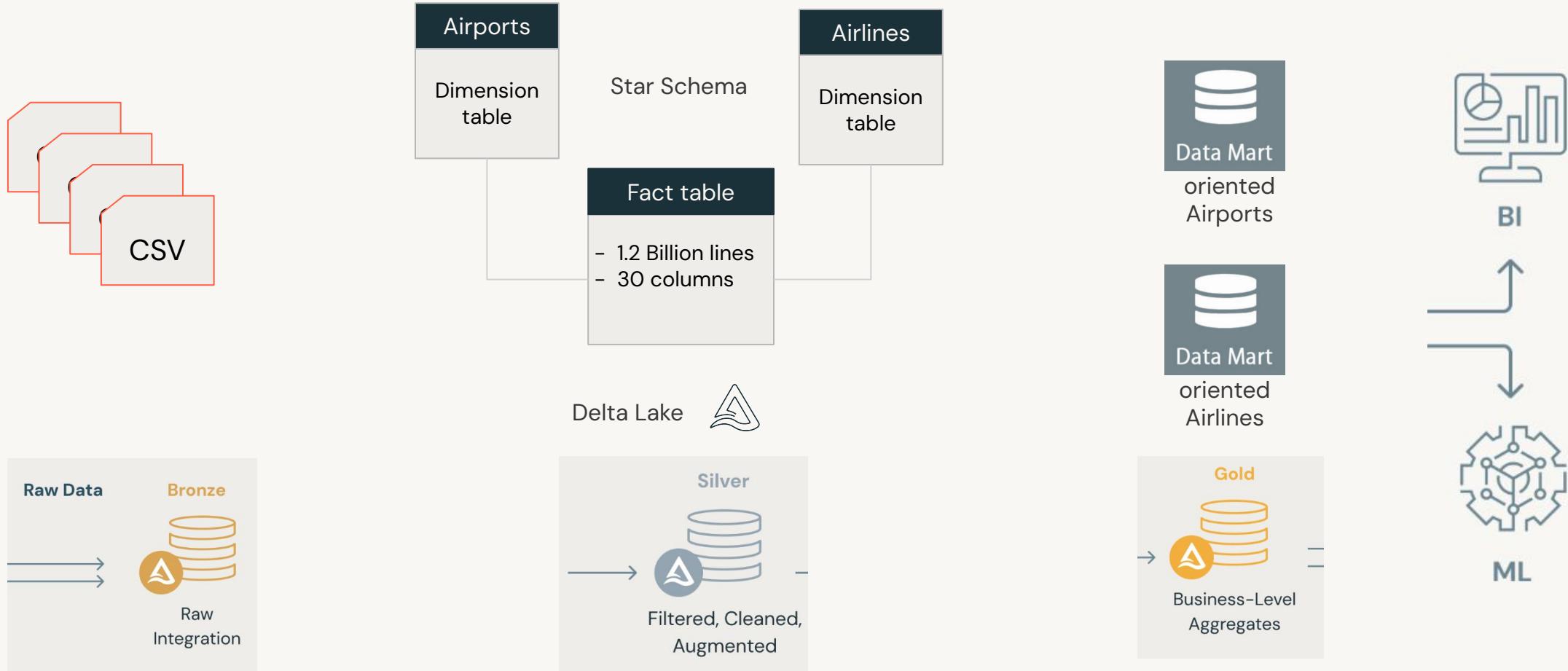


[https://github.com/frenchlam/DBKS\\_SQL\\_training\\_july3rd](https://github.com/frenchlam/DBKS_SQL_training_july3rd)



# Airline ontime Dataset ( ASA Data Expo 2009 )

US Domestic flights 1988 – 2008



# LAB – Getting started

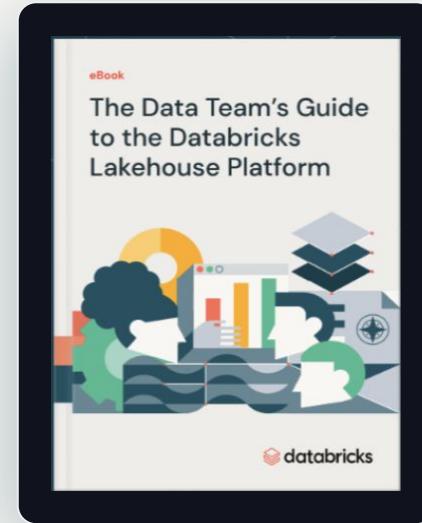
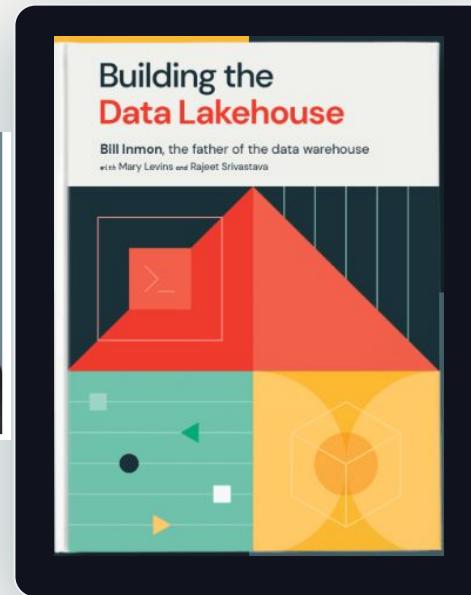
The screenshot shows the Databricks workspace interface. The left sidebar has a dark theme with various navigation options like New, Workspace, Recents, Catalog, Workflows, Compute, SQL, and Data Engineering. The 'Workspace' option is selected. The main area shows a folder named 'dbsql\_workshop' which contains a single item: 'DBSQL Workshop Lab Instructions' (Notebook) owned by 'odl\_instructor\_1135473@databrickslabs.com'. The top navigation bar shows the URL 'adb-3311722874107344.4.azuredatabricks.net/browse/folders/1091466089353624?o=3311722874107344'.

Name	Type	Owner
DBSQL Workshop Lab Instructions	Notebook	odl_instructor_1135473@databrickslabs.com

# Databricks' guide to the lakehouse



Bill Inmon



# Unity Catalog in Databricks SQL

# Data and AI governance drives business value

“Organizations are finally realizing the value of **data as an asset** that needs to be protected, managed and maintained to **increase asset value**”

—  
IDC

“Organizations seeing the **highest returns** from AI have a framework for **AI governance** to cover every step of the model development process”

—  
The State of AI in 2022, McKinsey & Co

“AI is now an enterprise essential, and as such, **AI governance** will join cybersecurity and compliance as a **board-level topic**”

—  
Forrester, 2023 AI Predictions report

# An Ideal Governance Solution Needs:

**Open  
Connectivity**  
Any data, any  
source, any format

**Unified  
Governance**  
Across data  
and AI

**Open  
Access**  
Any compute  
engine/client



# Databricks Unity Catalog

## Open Connectivity



Amazon Redshift



Azure Data Lake Storage



Amazon S3



AWS Glue



PostgreSQL



Cloud Storage



Google BigQuery



HIVE



Microsoft SQL Azure™



## Unified Governance for Data and AI

### Unity Catalog

Tables



Files



Models



AI Tools



Access control

Data sharing

Discovery

Lineage

Monitoring

Auditing

## Open Access



DuckDB



trino



presto



ICEBERG



Amazon EMR



Microsoft Fabric



Starburst



StarRocks



LangChain



PuppyGraph



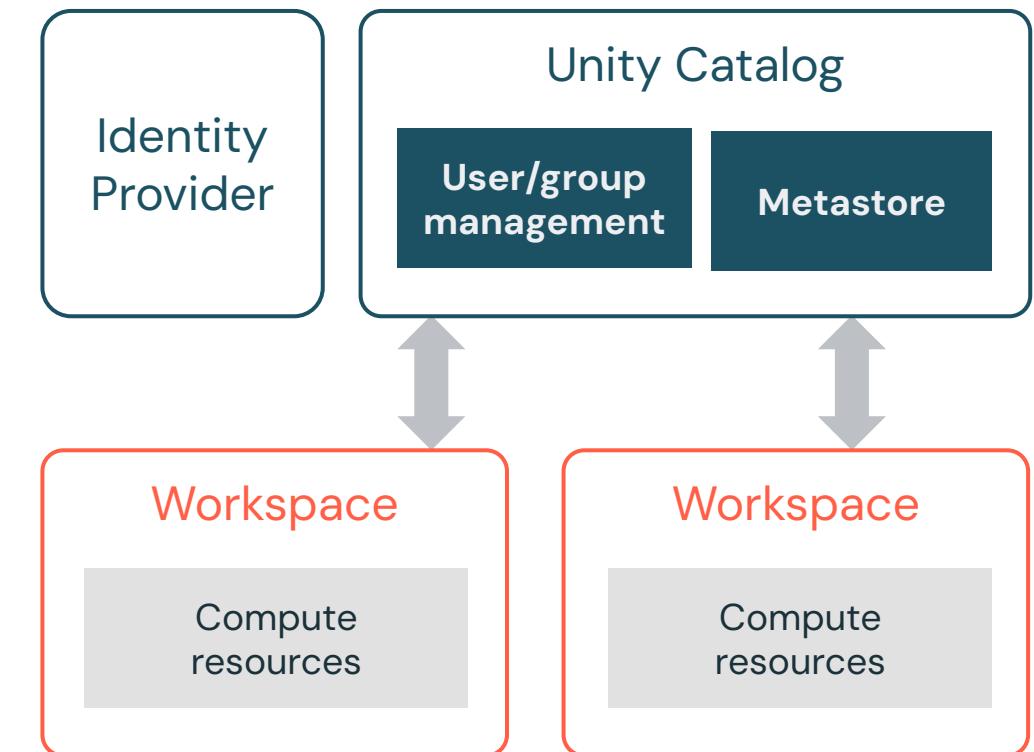
TECTON

daft

# Unity Catalog

## Architecture

- Implements access control on data
- Access control is always enabled
- Works across multiple workspaces
- Grants permissions to users at the account level



# Fundamental Concepts

## Working with file based data sources

- Credentials
  - Cloud provider credential to connect to storage
- External Locations
  - Storage location used for external tables or arbitrary files
- Managed Data Sources
  - External Location that is used exclusively for tabular data
- Volumes
  - Arbitrary file container inside an external location

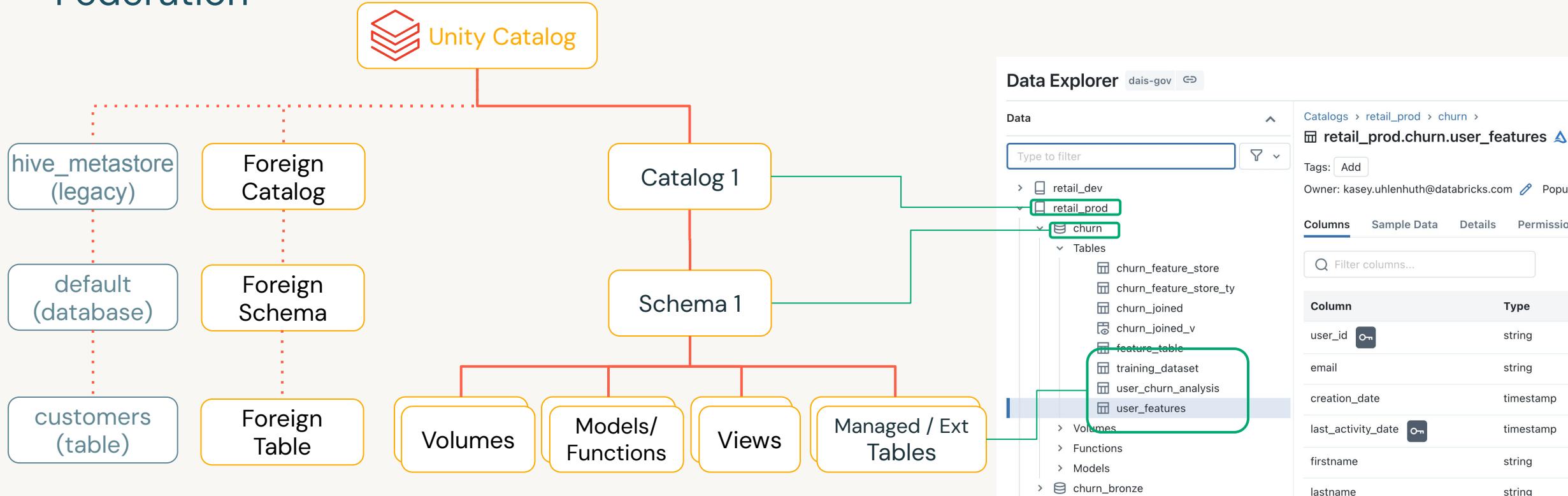
## Working with databases

- Connections
  - Credential and connection information to connect to an external database
- Foreign Catalogs
  - A catalog that represents an external database in UC and can be queried alongside managed data sources and file sources



# Governed namespace across file and database sources

Access legacy metastore and foreign databases powered by Lakehouse Federation



```
SELECT * FROM main.paul.red_wine; -- <catalog>.<database>.<table>
```

```
SELECT * FROM hive_metastore.default.customers;
```

```
SELECT * FROM snowflake_warehouse.some_schema.some_table;
```

# Unity Catalog Features

## Discovery

AI Generated Documentation (Preview)

Tagging (Preview)

Semantic Search

Data Usage Insights

## Access Control

Row and Column Security

Volumes

Scala in Shared Access Mode (Preview)

ABAC (Preview)

## AI Governance

Models in Unity Catalog

Feature Store

Vector Search

## Lineage

Volumes

Federated Sources

Models (Preview)

Bring Your Own Lineage (Preview)

## Sharing

Notebooks

Marketplace

Models (Preview)

Clean rooms (Preview)

## Monitoring

Lakehouse Monitoring

System Tables-Billable Usage



# Follow Along Demo: Setting Up a Catalog and Schema

# Follow along Demo

## Catalog Walkthrough

- Data Discovery
- AI Governance
- Data Access
- Data Lineage
- Data Sharing
- Monitoring

# LAB 1 – Unity Catalog

## Setting up your data assets

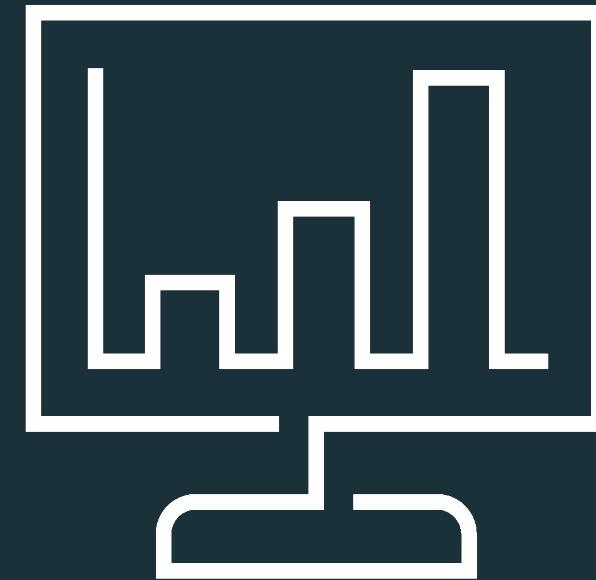
- Create a catalog
- Create a schema
- Create a volume
- Create a managed table
- Grant access to your catalog
- Query your data



TIME FOR  
A BREAK!



# Data Visualization and Dashboarding



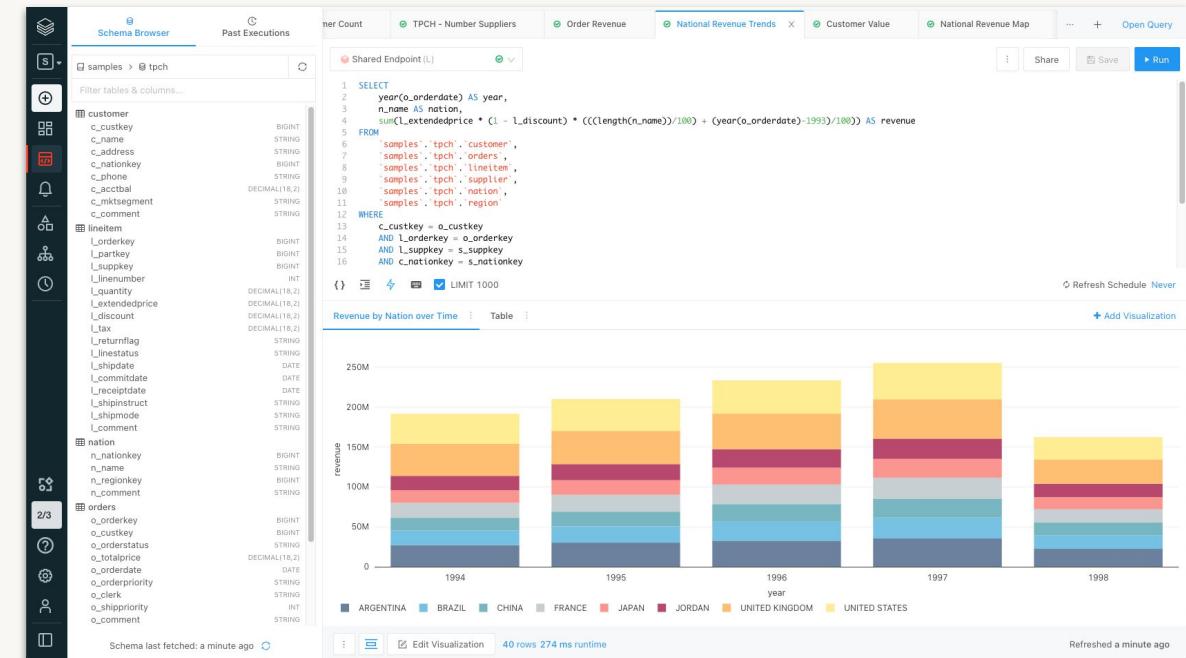
---

Databricks Academy 2023

# Databricks SQL Query Editor

Collaboratively query, explore, and transform data in-place

- Easily **ingest** data from cloud storage, local files, or business applications using Fivetran
- Discover data, explore database schema, and query data using **ANSI SQL**
- Save, share, and reuse queries across teams to get to results faster
- Orchestrate queries, alerts, and dashboards with automated **workflows**
- Stay up to date with alerts and automatic refresh schedules



# Databricks

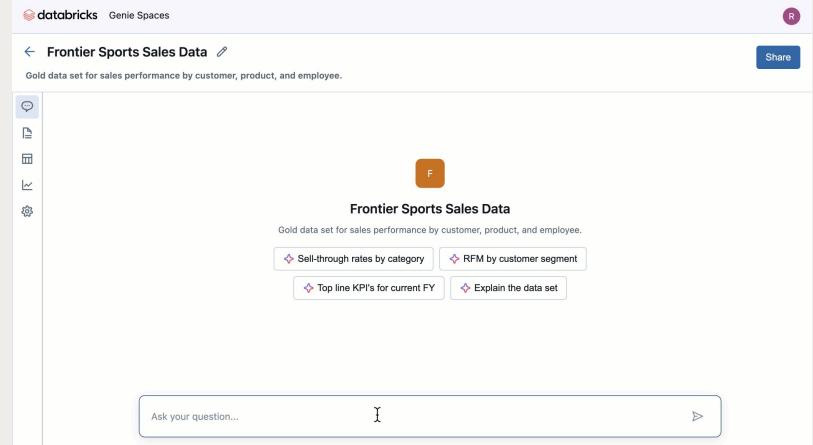
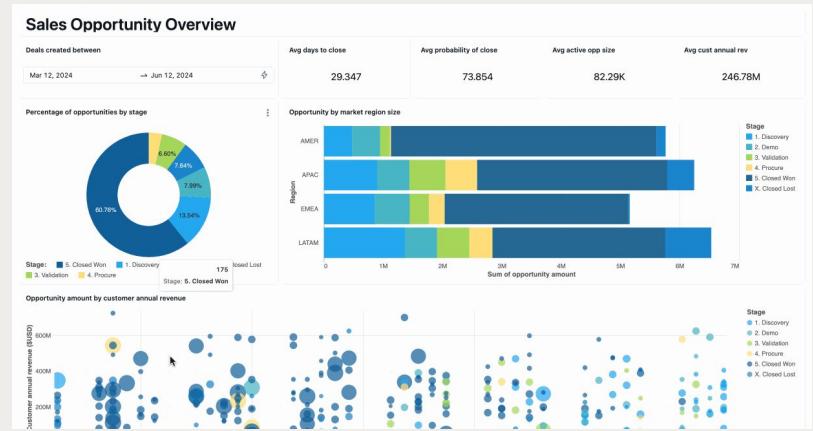
## AI/BI

### Intelligent analytics for real-world data

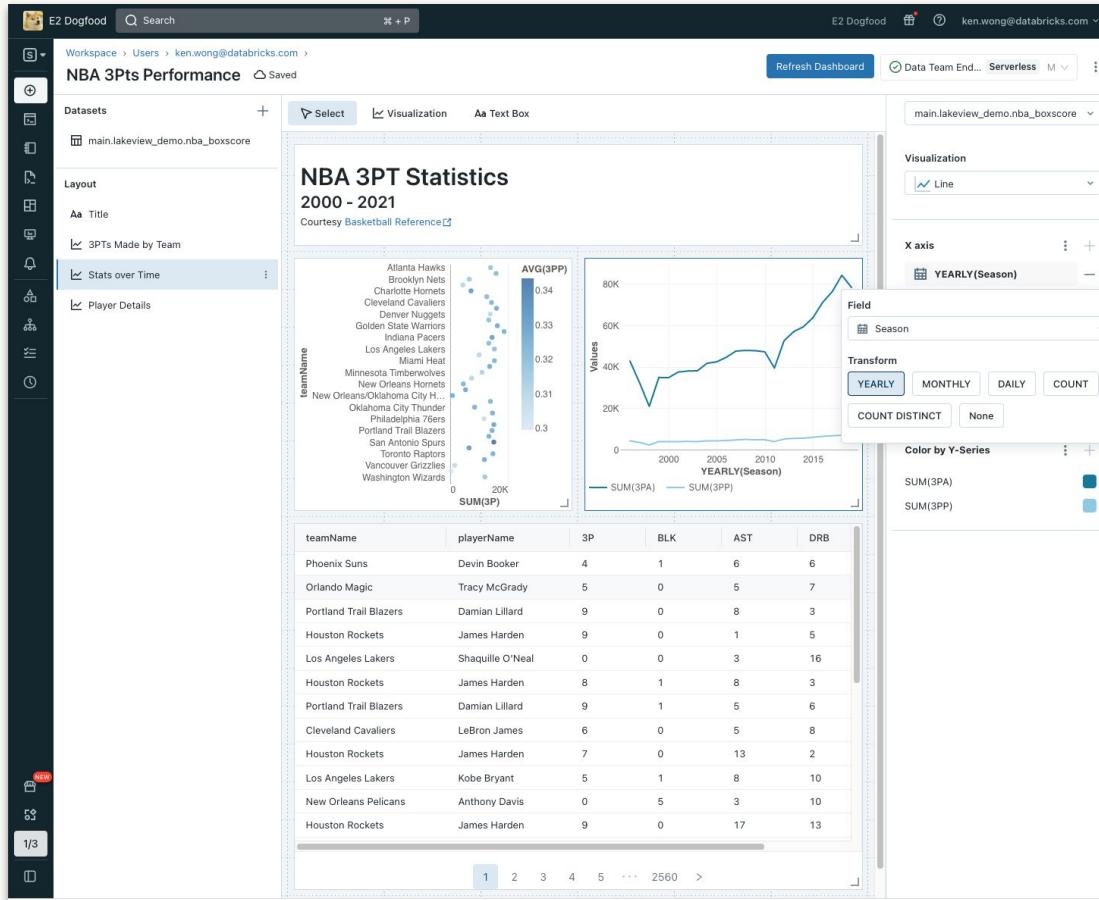
Generally Available  
Dashboards

Public Preview  
Genie

Governed and secured with Unity Catalog



# AI/BI Lakeview: Native BI for the Lakehouse



## Simple

New content model, new visualization library, and **SQL-optional UX** for simple use cases

## Shareable

Publish dashboards with optimizations for broad distribution **beyond the workspace**

## Integrated

Integrations into **Unity Catalog** and interoperability with Notebooks

# AI/BI Genie Spaces:

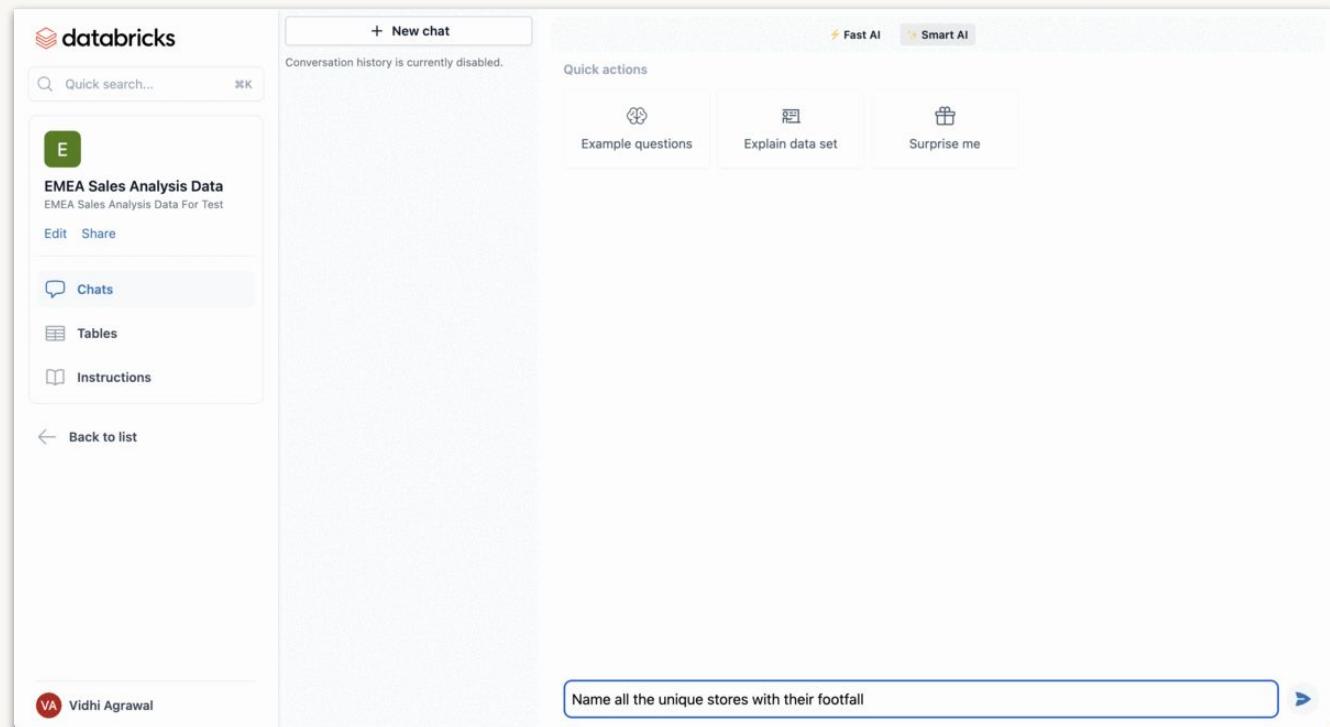
Data and AI for all with  
natural language

**Enable business users to  
interact with data with  
LLM-powered Q&A**

Ask questions in natural language and receive  
answers in text and visualizations

Curate dataset-specific experiences with  
custom instructions

Powered by Databricks SQL & DatabricksIQ



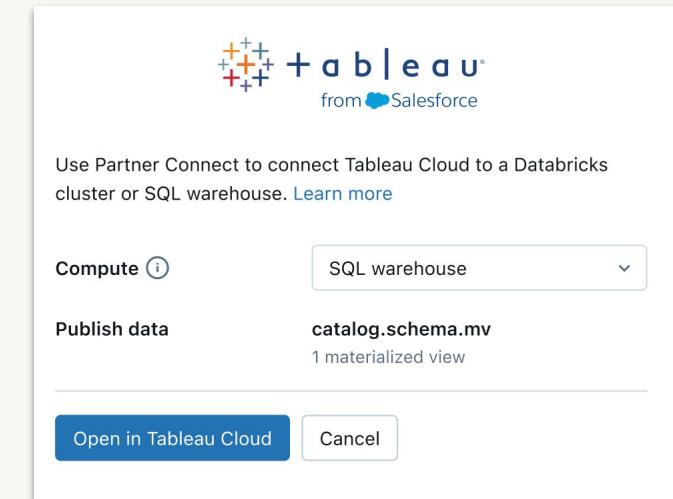
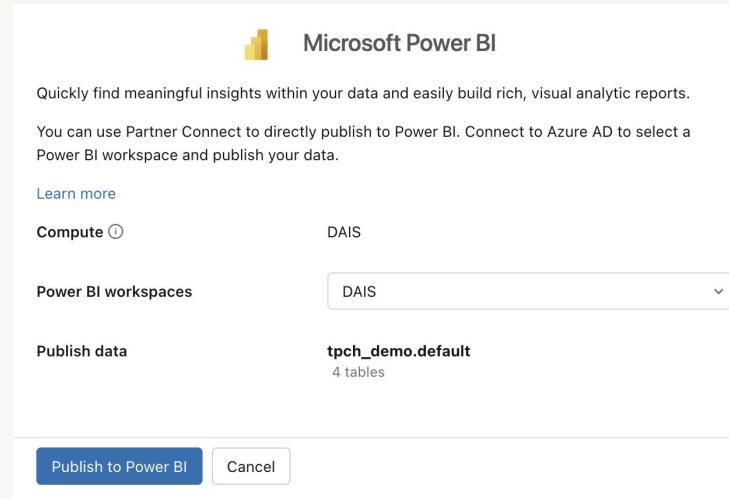
# Deep Power BI & Tableau Integrations

## Seamless catalog integration & data model sync

### Power BI Integration

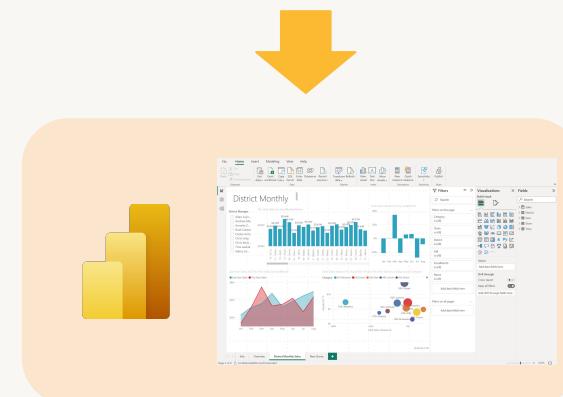
Publish UC datasets from Databricks UI, without PBI Desktop to Power BI Online.

Sync entire schemas including table relationships (PK/FK) to save time.



### Tableau Integration

Easily explore Unity Catalog datasets in Tableau Online with a single click from Data Explorer.



# Lab: Data Visualizations

# Follow along Demo

## Data Visualizations

- Create a Query in SQL Editor
- Create two visualizations

# Lab: Lakeview Dashboards

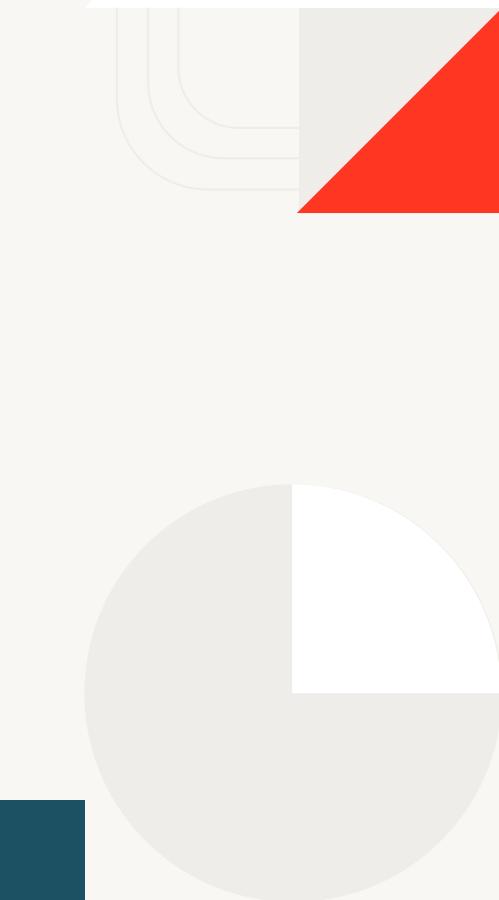
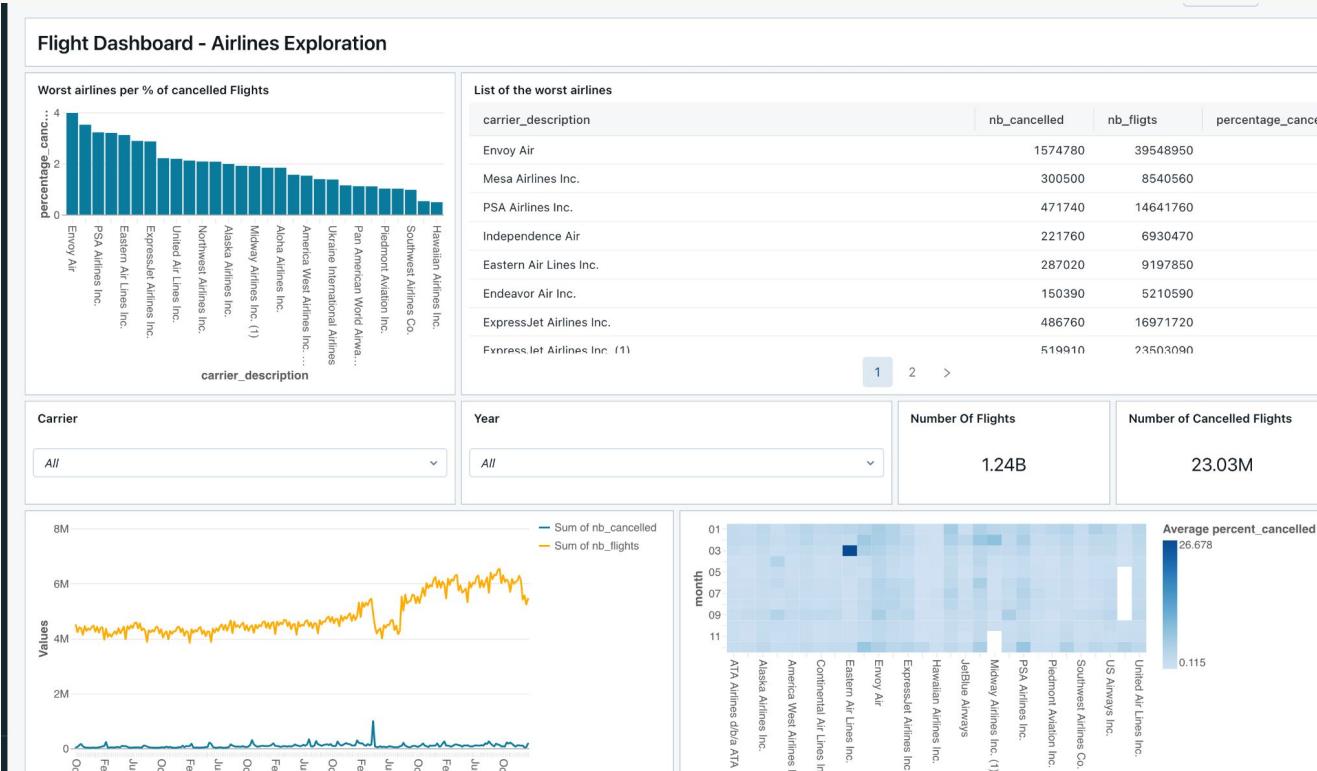
DAWD 01-5

# Lab



# Dashboards

- Create your own Lakeview dashboard



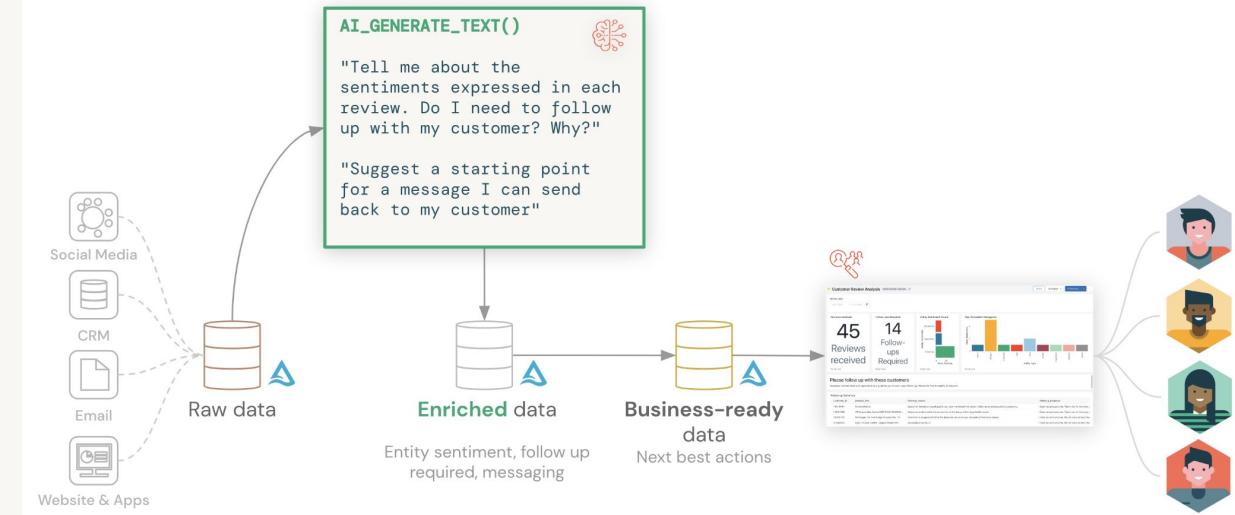
# Integration with AI

# AI Functions for DBSQL

Access Proprietary and Open Source LLMs directly within DBSQL



- Enable analysts to classify data with LLMs using **built-in functions**.
- Generate product descriptions, summaries, and more by simply adding the [ai\\_query\(\)](#) function.



Public Preview

# Use built-in AI Functions

## Analyze unstructured data with best-in-class models

Built-in functions invoke a state-of-the-art generative AI model to perform tasks like, sentiment analysis, classification and translation.

### Use cases

- Extract top product issues from call center transcripts—without manual tagging!
  - Tag customers as a potential churn risk based on customer support chat logs
  - Generate customized product descriptions for ad campaigns—automatically
  - Read product reviews to understand buying decision criteria
- ...many more...

- `ai_analyze_sentiment`
- `ai_classify`
- `ai_extract`
- `ai_fix_grammar`
- `ai_gen`
- `ai_mask`
- `ai_similarity`
- `ai_summarize`
- `ai_translate`



# Use ai\_query() function

The `ai_query()` function allows you to serve your machine learning models and large language models using [Mosaic AI Model Serving](#) and query them using SQL.

You can use `ai_query()` to:

- Query endpoints that serve custom models,
- Query foundation models made available using [Foundation Model APIs](#),
- Query [external models](#).

```
SELECT
  sku_id,
  product_name,
  ai_query(
    "my-external-openai-chat",
    "You are a marketing expert for a winter
holiday promotion targeting GenZ. Generate a
promotional text in 30 words mentioning a 50%
discount for product: " || product_name
  )
FROM
  uc_catalog.schema.retail_products
WHERE
  inventory > 2 * forecasted_sales
```



## How to leverage your AI Models in Databricks SQL?

1. Call an AI Foundation Model using SQL
2. Create a SQL function calling a Foundation Model
3. Leverage a custom AI Model in SQL
4. Leverage an external AI Model in SQL

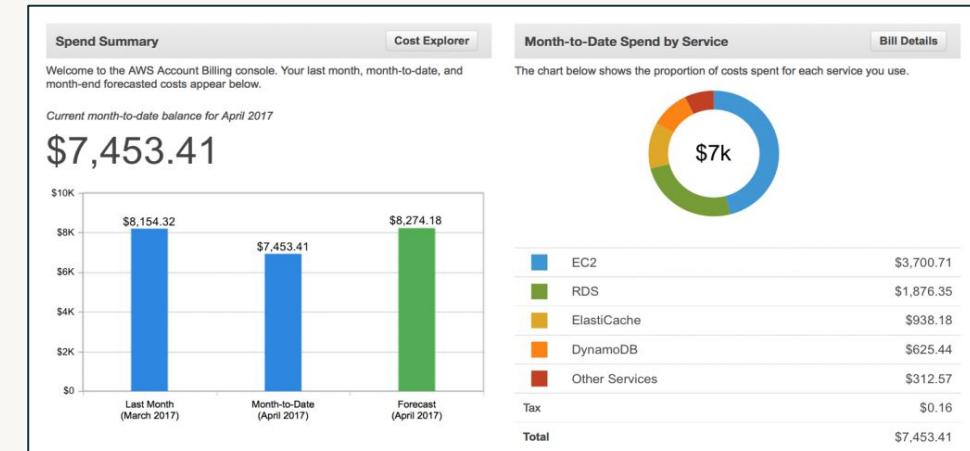
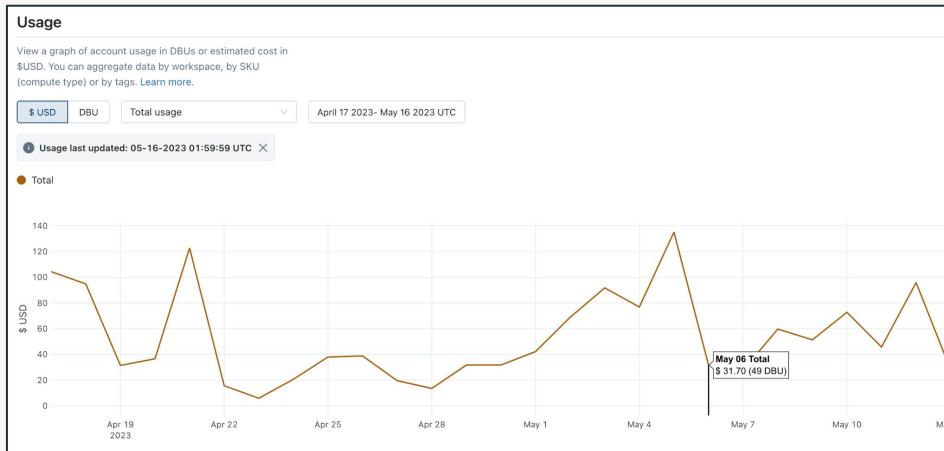
# Observability - using System Tables



# What is Observability?

Observability is a key aspect of a data platform which:

- Provides visibility into detailed platform usage
- Helps to correlate the spend/cost implications with different use cases
- Enables Security & Audit of data platform access



# What are some Observability questions we may want to answer?

- Monitoring Costs
  - Which are my top spending clusters?
  - Which are my top spending jobs?
  - Which are my top spending projects?
- Monitoring Security and Audit
  - Monitor and track Users and User Groups access to data of different classifications
  - Track Account Logins & modifications
- Monitoring Platform Usage / Pipeline states
  - What states are my pipelines in ?
  - How many new jobs are being added every month?
  - Which are my long running jobs/queries?
  - What are the cluster & job configs being used?
  - How many users are there in the workspace?

## Data Observability & Optimisation

- Which are my largest tables?
- Which tables have the most frequent writes?
- Which columns to z-order by?

## Performance/Resource Utilisation

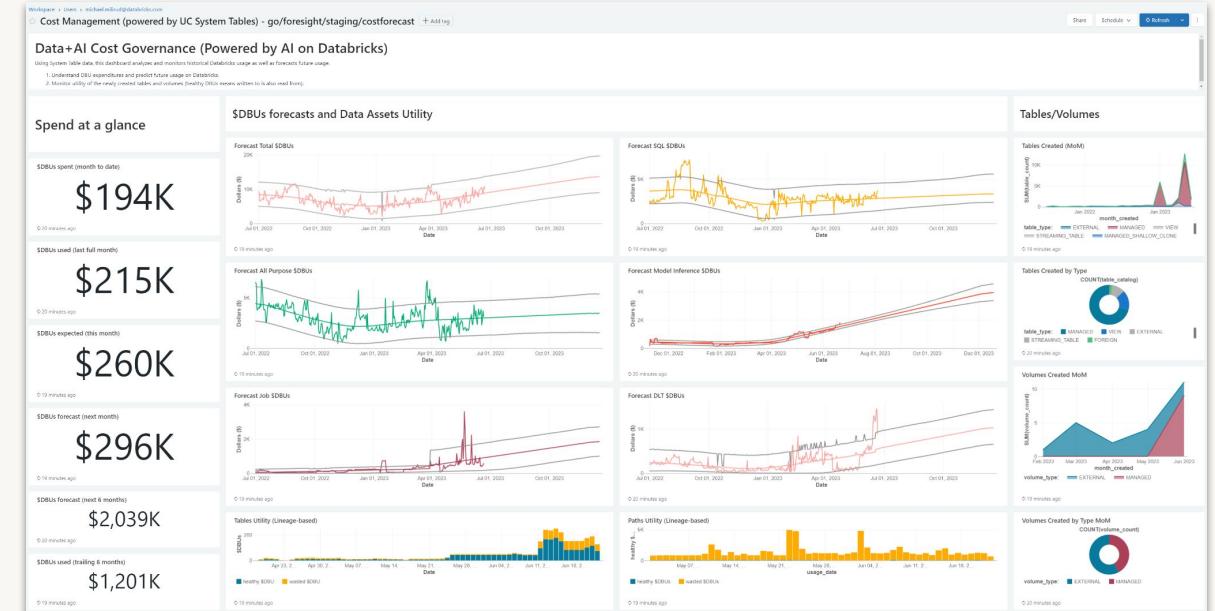
- Which are my lowest utilised clusters?
- How is utilisation trending at a hour/min level



# System Tables

System Tables are a Databricks-hosted analytical store for **all** of Databricks operational data's warm path used for historical customer observability, including:

- Audit/lineage analytics
- Cost/usage analytics
- Efficiency analytics
- SLO analytics
- Data Quality analytics



for jobs, notebooks, clusters, SQL/ML endpoints, etc.



# Audit Events

Audit Log events in System tables helps you to answer common account usage questions

- Who accessed this table?
- Which users accessed a table within the last day?
- Which tables did a user access?
- View all permissions changes
- View the most recently run notebook commands

Catalogs > system > access >  
system.access.audit

Tags: [Add](#)  
Owner: System user [Edit](#) Popularity: Size: Unknown

[Columns](#) [Sample Data](#) [Details](#) [Permissions](#) [History](#) [Lineage](#) [Insights](#) [Quality](#)

Filter columns...

Column	Type
account_id	string
workspace_id	string
version	string
event_time	timestamp
event_date	date
source_ip_address	string
user_agent	string
session_id	string
user_identity	struct
service_name	string
action_name	string



# Billable usage

## Analyze and optimize DBU consumption by leveraging usage tables

- What is the daily trend in DBU consumption?
- How many DBUs of each SKU have been used throughout this month?
- How much of each SKU did a workspace use on June 1?
- Which jobs consumed the most DBUs?
- How much usage can be attributed to resources with a certain tag?
- Show me the SKUs where usage is growing
- What is the usage trend of All Purpose Compute (Photon)?

Catalogs > system > billing >  
system.billing.usage

Tags: [Add](#)  
Owner: System user Popularity: Size: Unknown

Columns Sample Data Details Permissions History Lineage Insights Quality

Filter columns...

Column	Type
account_id	string
workspace_id	string
usage_record_id	string
sku_name	string
cloud	string
usage_start_time	timestamp
usage_end_time	timestamp
usage_date	date
custom_tags	map
usage_unit	string
usage_quantity	decimal
system_metadata	struct



# Table and Column Lineage

Build on Unity Catalog's data lineage feature, allowing you to programmatically query lineage data to fuel decision making and reports.

There are two lineage system tables:

- `system.access.table_lineage`
- `system.access.column_lineage`

The screenshot shows two side-by-side tables in the Databricks UI. Both tables are titled 'Catalogs > system > access >' followed by their respective names: 'system.access.table\_lineage' and 'system.access.column\_lineage'. Each table has columns for 'Owner', 'Popularity', 'Size', 'Tags', and buttons for 'Add tags', 'Sample Data', 'Details', 'Permissions', 'History', 'Lineage', 'Insights', and 'Quality'. The 'Columns' tab is selected for both tables. Below the tabs are two search bars labeled 'Filter columns...'. The first table ('table\_lineage') has columns: account\_id, metastore\_id, workspace\_id, entity\_type, entity\_id, entity\_run\_id, source\_table\_full\_name, source\_table\_catalog, source\_table\_schema, source\_table\_name, source\_path, source\_type, target\_table\_full\_name. The second table ('column\_lineage') has columns: account\_id, metastore\_id, workspace\_id, entity\_type, entity\_id, entity\_run\_id, source\_table\_full\_name, source\_table\_catalog, source\_table\_schema, source\_table\_name, source\_path, source\_type, source\_column\_name. All columns are of type 'string'.

Catalogs > system > access >											
system.access.table_lineage											
Owner: System user Popularity: Size: Unknown Tags: Add tags											
Columns Sample Data Details Permissions History Lineage Insights Quality											
Columns	Sample Data	Details	Perm	Columns	Sample Data	Details	Permissions	History	Lineage	Insights	Quality
<input type="text"/> Filter columns...			<input type="text"/> Filter columns...			<input type="text"/> Filter columns...			<input type="text"/> Filter columns...		
Column	Column	Type	Column	Column	Type	Column	Column	Type	Column	Column	Type
account_id	account_id	string	metastore_id	metastore_id	string	workspace_id	workspace_id	string	entity_type	entity_type	string
metastore_id	metastore_id	string	workspace_id	workspace_id	string	entity_type	entity_type	string	entity_id	entity_id	string
workspace_id	workspace_id	string	entity_type	entity_type	string	entity_id	entity_id	string	entity_run_id	entity_run_id	string
entity_type	entity_type	string	entity_id	entity_id	string	entity_run_id	entity_run_id	string	source_table_full_name	source_table_full_name	string
entity_id	entity_id	string	entity_run_id	entity_run_id	string	source_table_full_name	source_table_full_name	string	source_table_catalog	source_table_catalog	string
entity_run_id	entity_run_id	string	source_table_catalog	source_table_catalog	string	source_table_schema	source_table_schema	string	source_table_schema	source_table_schema	string
source_table_full_name	source_table_full_name	string	source_table_schema	source_table_schema	string	source_table_name	source_table_name	string	source_table_name	source_table_name	string
source_table_catalog	source_table_catalog	string	source_table_name	source_table_name	string	source_path	source_path	string	source_path	source_path	string
source_table_schema	source_table_schema	string	source_path	source_path	string	source_type	source_type	string	source_type	source_type	string
source_table_name	source_table_name	string	source_type	source_type	string	target_table_full_name	target_table_full_name	string	source_column_name	source_column_name	string

# Clusters

## Understand infrastructure usage and scaling events

- How long were clusters up ?
- What type of VM types were used ?
- How did they scale up / down

The screenshot shows the Databricks Catalog Explorer interface. The left sidebar displays a tree view of catalogs under 'sys'. The 'compute' catalog is expanded, showing sub-schemas like 'clusters', 'node\_timeline', 'node\_types', 'warehouse\_events', and 'information\_schema'. The right panel shows the details for the 'clusters' schema. It includes the owner ('System user'), popularity rating, and a table of columns with their types and comments.

Column	Type	Comment
account_id	string	(+)
workspace_id	string	(+)
cluster_id	string	(+)
cluster_name	string	(+)
owned_by	string	(+)
create_time	timestamp	(+)
delete_time	timestamp	(+)
driver_node_type	string	(+)
worker_node_type	string	(+)
worker_count	long	(+)
min_autoscale_workers	long	(+)
max_autoscale_workers	long	(+)

## Schemas

- **system.compute.clusters**
- **system.compute.node\_timeline**
- **system.compute.node\_types**
- **system.compute.warehouse\_events**



# Query

## Understand how data is being queried and by whom

- What tables/column are most frequently used ?
- How are tables being queries ?
- Who looks at what data ?

Catalog Explorer field-eng-east Send feedback

Catalog sys

In my org

- akshay\_system
- bd\_pharmasystems\_demo
- cdg\_sys\_tester
- dws\_syst\_eqd\_strg\_derived\_ffp

system

- access
- billing
- compute
- clusters
- node\_timeline
- node\_types
- warehouse\_events
- hms\_to\_uc\_migration
- information\_schema
- lineage
- marketplace
- query
- history
- storage
- predictive\_optimization\_operations...

Shared

- net\_one\_systems\_co\_ltd\_netone\_v...

Catalogs > system > query > system.query.history

Owner: System user Popularity: .all

Tags: Add tags

Columns Sample Data Details Permissions History Lineage Insights Quality

Filter columns...

Column	Type	Comment
account_id	string	(+)
workspace_id	string	(+)
statement_id	string	(+)
executed_by	string	(+)
session_id	string	(+)
execution_status	string	(+)
warehouse_id	string	(+)
executed_by_user_id	string	(+)
statement_text	string	(+)
statement_type	string	(+)
error_message	string	(+)
warehouse_channel	string	(+)



## **How to understand usage and costs**

1. Query the Billing table to get all serverless product usage cost by workspace
2. Import the prepared account monitor dashboard

# Summary and Next Steps

# What we've learnt

- What Databricks SQL is
- How Databricks SQL works in the Lakehouse architecture
- How to ensure the best performance with Databricks SQL
- How Databricks SQL can help you:
  - Create tables and Query data
  - Create visualizations and dashboards
  - Integrate with AI models

# Earn a Databricks certification!

Certification helps you gain industry recognition, competitive differentiation, greater productivity, and results.

- This course helps you prepare for the **Databricks Certified Data Analyst Associate exam**
- Recommended Self-Paced Courses
  - Ingesting Data for Databricks SQL
  - Integrating BI Tools with Databricks SQL
- Please see the Databricks Academy for additional prep materials



For more information visit:  
[databricks.com/learn/certification](https://databricks.com/learn/certification)



DATA INTELLIGENCE DAY

MARDI 23 AVRIL, PARIS

**Une demi-journée pour découvrir  
la Data Intelligence Platform :**

- Sessions Techniques
- Témoignages clients : RATP et Kiliba
- Networking
- Formations hands-on (*après-midi*)

AGENDA & INSCRIPTION :

<https://events.databricks.com/data-intelligence-day-paris>





DATA INTELLIGENCE DAY

EN PRÉSENTIEL

# Data Intelligence Day Paris

Découvrez la Data Intelligence Platform

MARDI 23 AVRIL, 8H30



# Thank you!



databricks