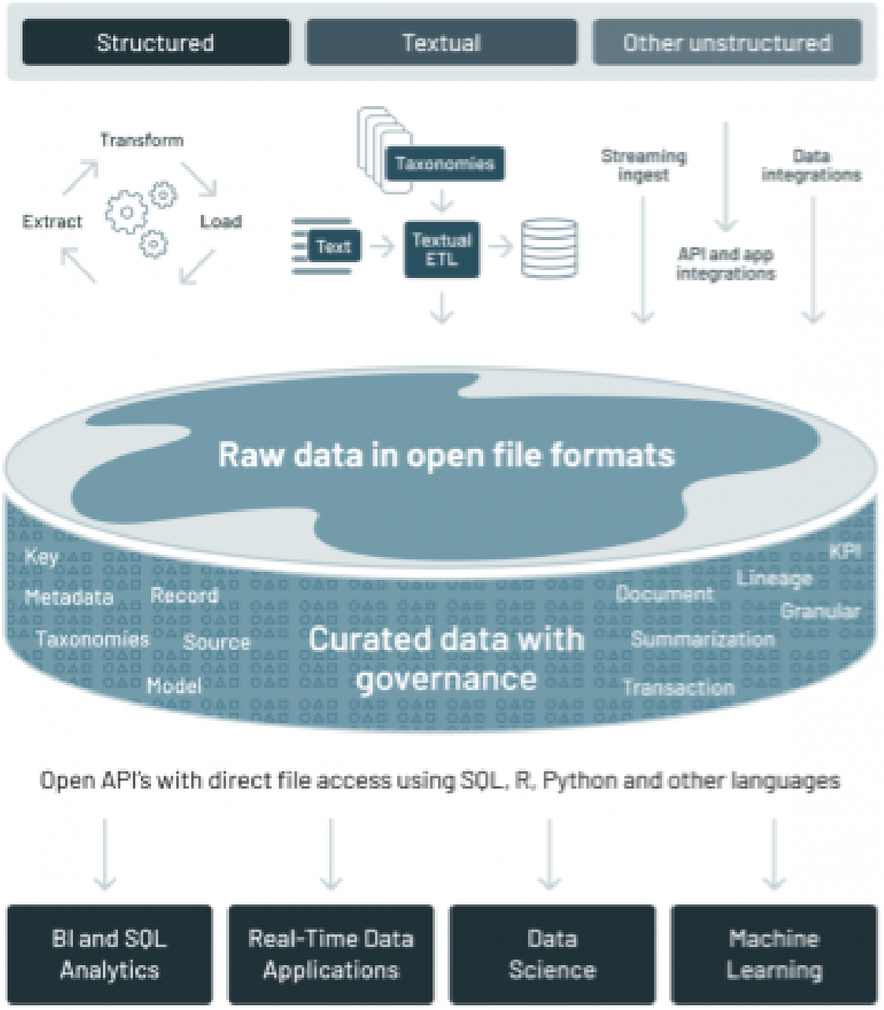
Overview of Databricks SQL

DATABRICKS CONCEPTS

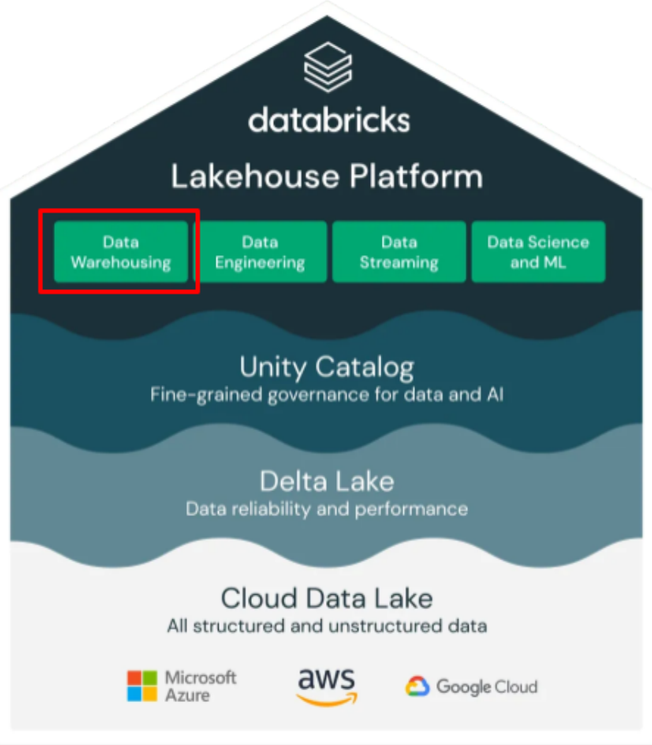
Kevin Barlow
Data Practitioner

datacamp

Data Lakehouse



Databricks for SQL Users



Databricks SQL

- Data Warehousing for the Lakehouse
- Familiar environment for SQL users
- SQL-optimized performance (Photon)
- Connect to your favorite BI tools

Comes built into the platform!

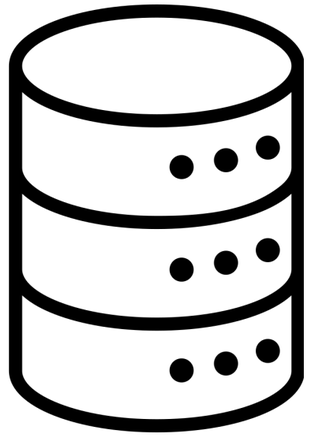
Databricks SQL vs. other databases

Databricks SQL

- Open file format (Delta)

Other Data Warehouses

- Proprietary data format



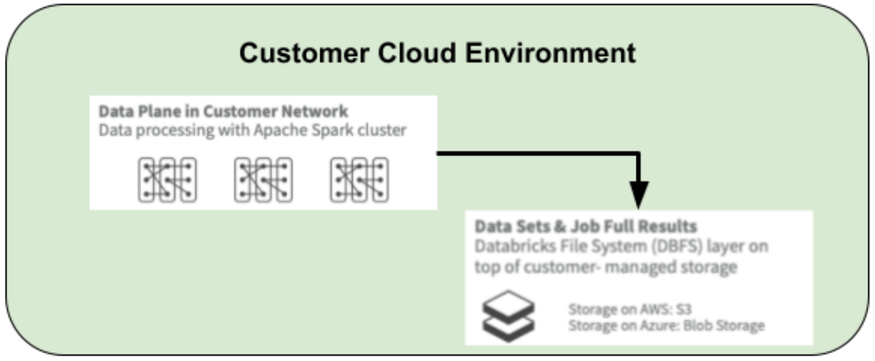
Databricks SQL vs. other databases

Databricks SQL

- Open file format (Delta)
- Separation of compute and storage

Other Data Warehouses

- Proprietary data format
- Storage often tied to compute



Databricks SQL vs. other databases

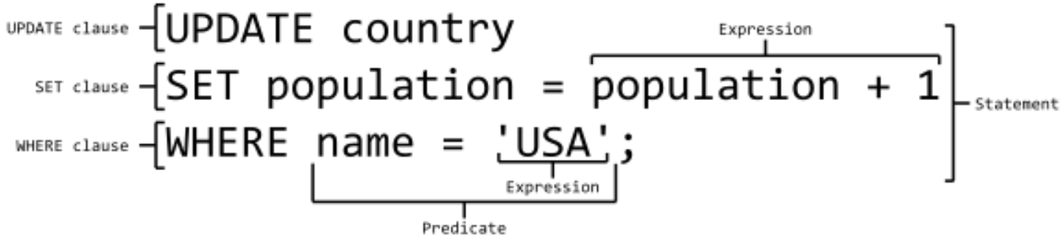
Databricks SQL

- Open file format (Delta)
- Separation of compute and storage
- ANSI SQL

Other Data Warehouses

- Proprietary data format
- Storage often tied to compute
- Tech-specific SQL

American National Standards Institute



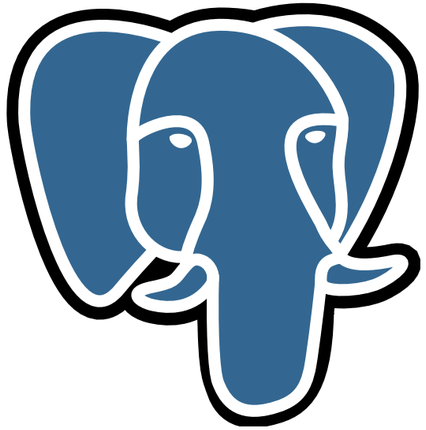
Databricks SQL vs. other databases

Databricks SQL

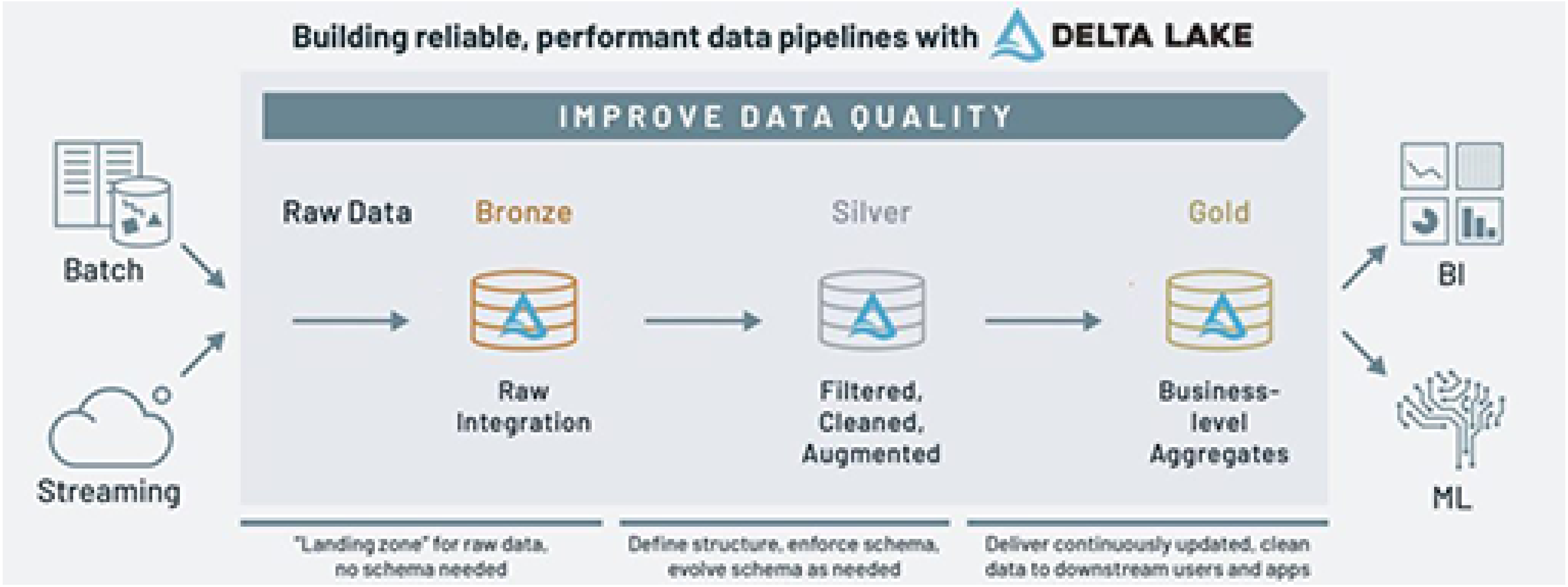
- Open file format (Delta)
- Separation of compute and storage
- ANSI SQL
- Integrated into other data workloads

Other Data Warehouses

- Proprietary data format
- Storage often tied to compute
- Tech-specific SQL
- Usually lacking advanced analytics



SQL in the Lakehouse Architecture



Let's review!

DATABRICKS CONCEPTS

Getting started with Databricks SQL

DATABRICKS CONCEPTS

Kevin Barlow
Data Practitioner

datacamp

SQL Compute vs. General Compute

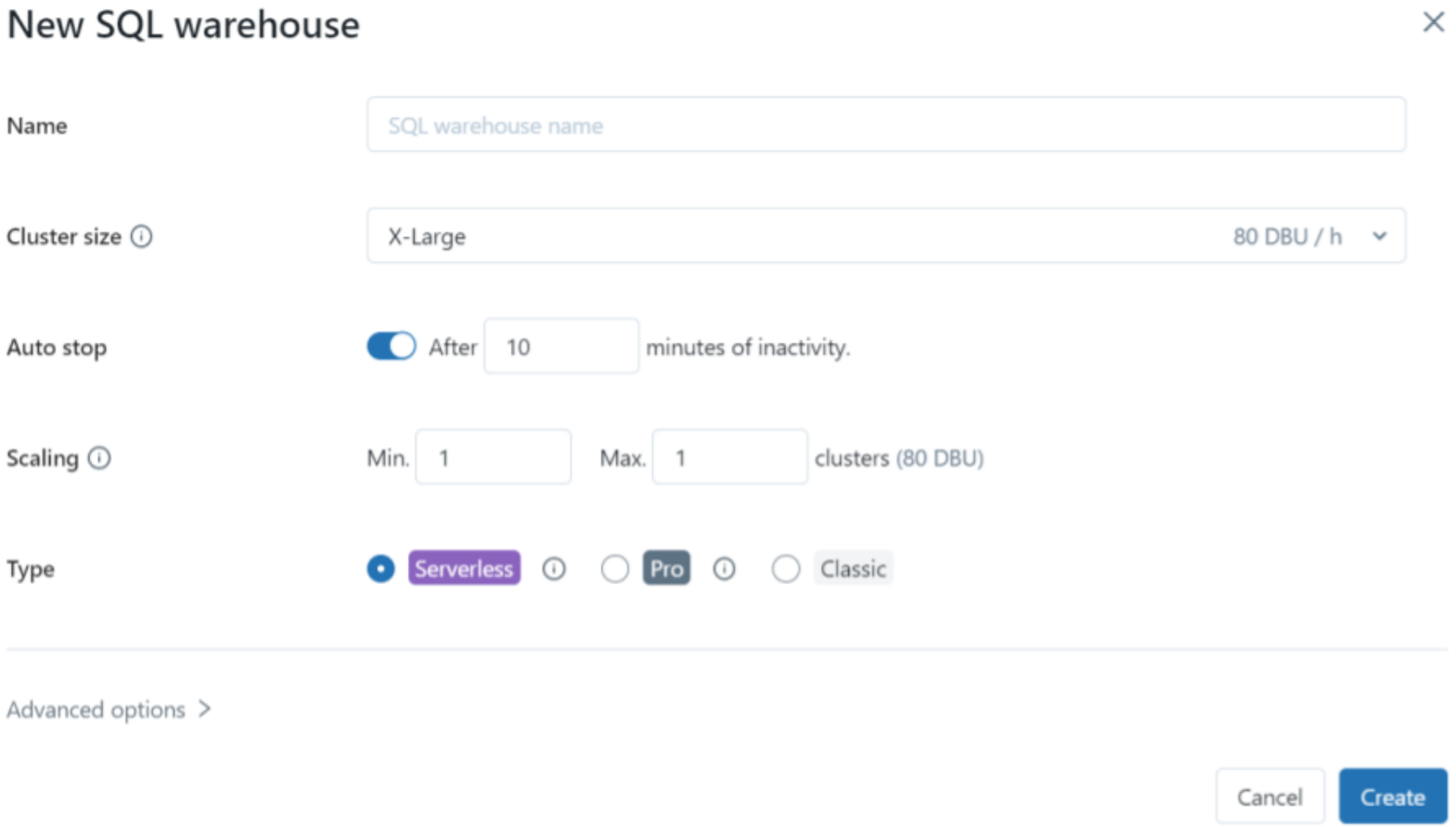
Designing compute clusters for data science or data engineering workloads...

```
import pyspark.sql.functions as F

spark_df = (spark
            .read
            .table('user_table'))

spark_df = (spark_df
            .withColumn('score',
                        F.flatten(...))
           )
```

is inherently different than designing compute for SQL workloads

```
SELECT *
FROM user_table u
LEFT JOIN product_use p
    ON u.userId = p.userId
WHERE country = 'USA'
AND utilization >= 0.6
```

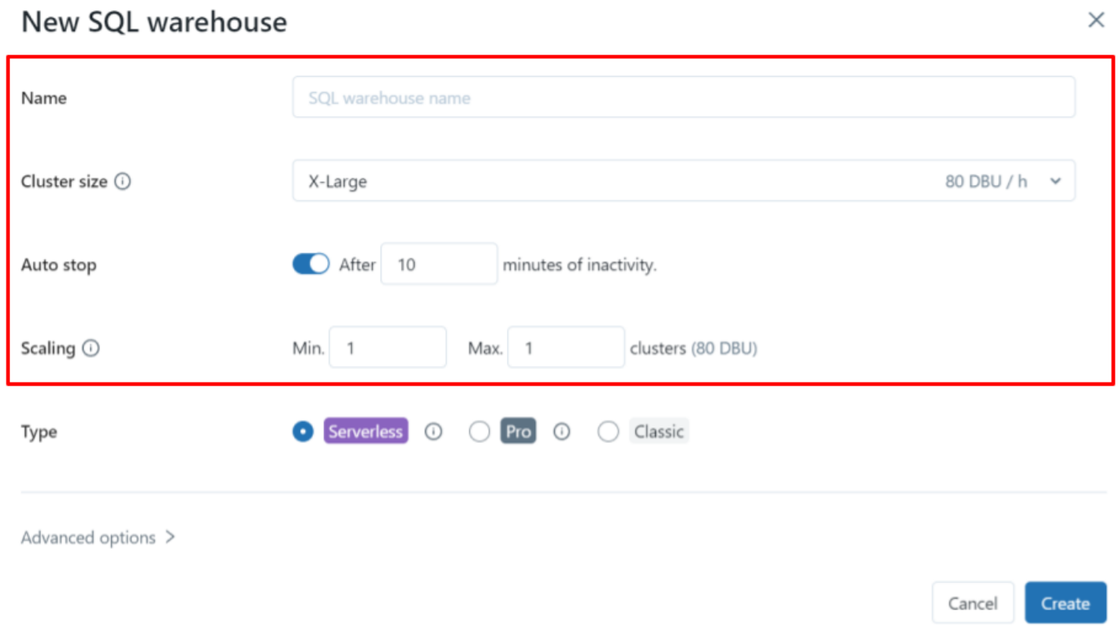
SQL Warehouse



SQL Warehouse

SQL Warehouse Configuration Options

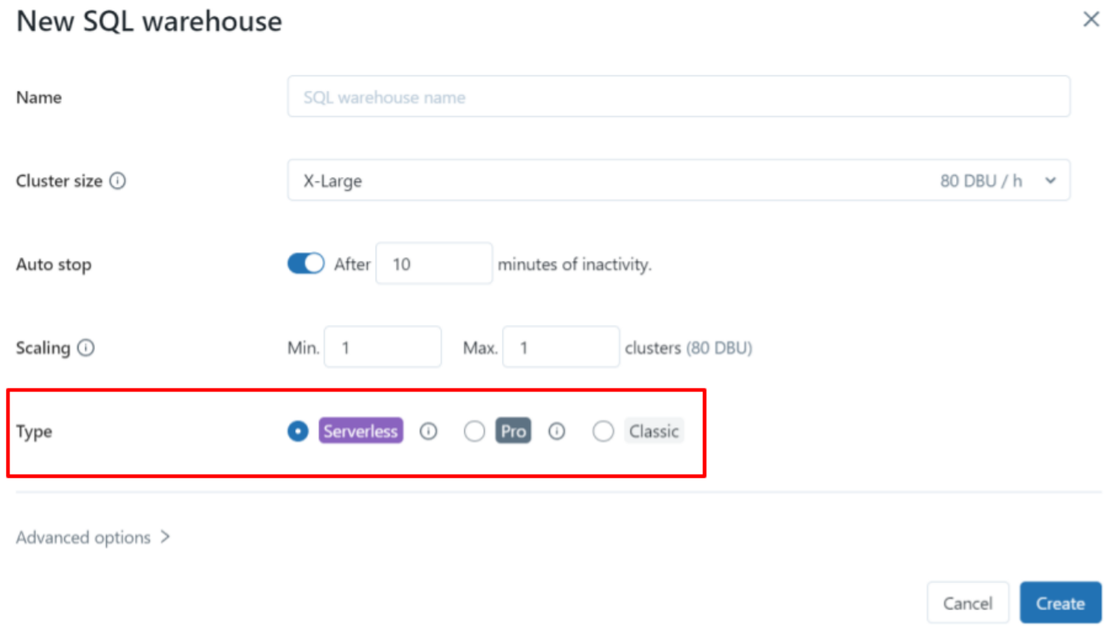
1. Cluster Name
2. Cluster Size (S, M, L, etc.)
3. Scaling behavior



SQL Warehouse

SQL Warehouse Configuration Options

1. Cluster Name
2. Cluster Size (S, M, L, etc.)
3. Scaling behavior
4. Cluster Type



SQL Warehouse Types

Different types provide different benefits

Classic

- Most basic SQL compute
- In customer cloud

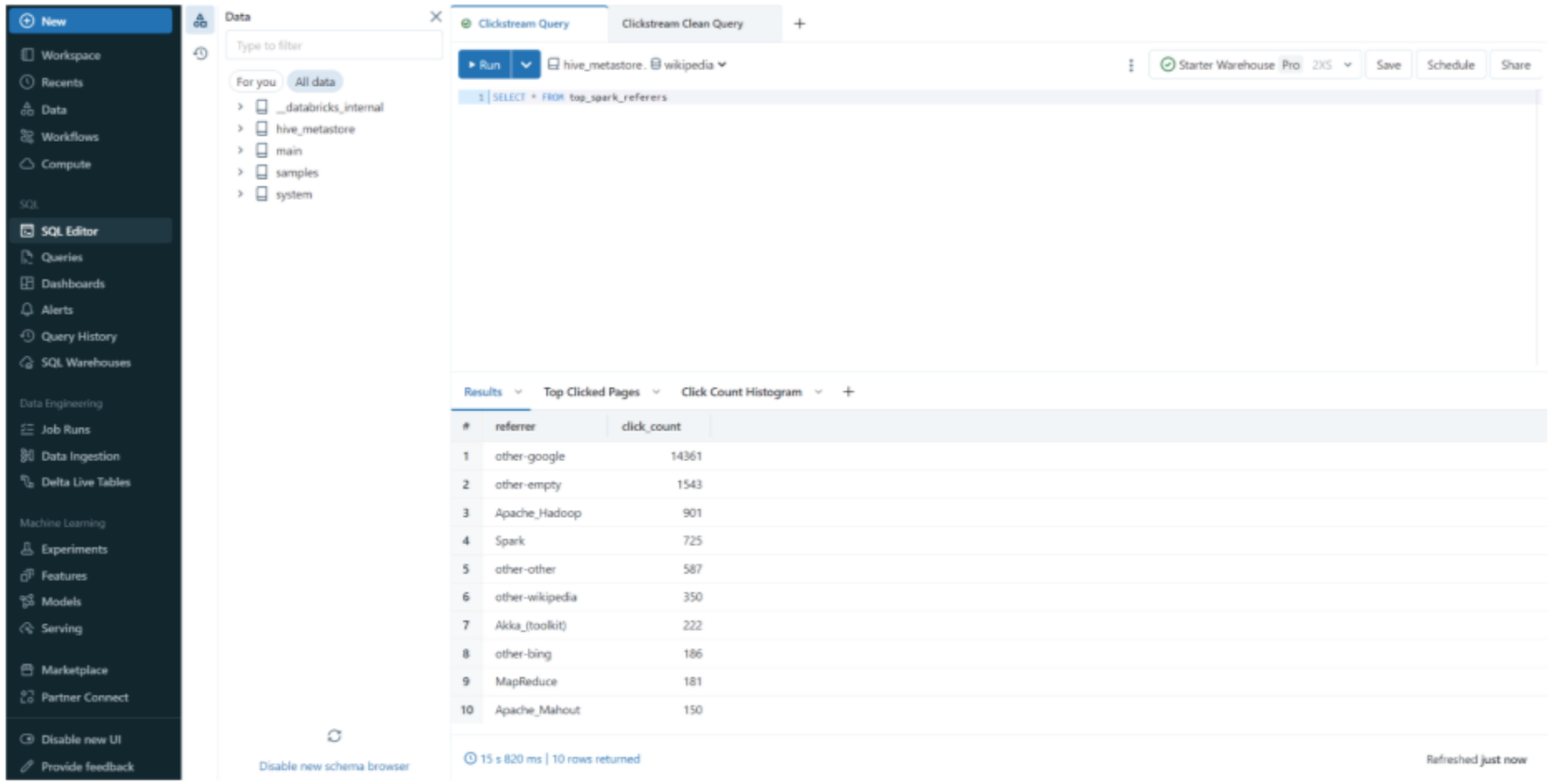
Pro

- More advanced features than Classic
- In customer cloud

Serverless

- Cutting edge features
- In Databricks cloud
- Most cost performant

SQL Editor


New
Workspace
Recents
Data
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Experiments
Features
Models
Serving
Marketplace
Partner Connect
Disable new UI
Provide feedback
Data
Type to filter
For you
All data
__databricks_internal
hive_metastore
main
samples
system
Disable new schema browser
Clickstream Query
Clickstream Clean Query
Run
hive_metastore . wikipedia
Starter Warehouse Pro 2XS
Save
Schedule
Share
1 SELECT * FROM top_spark_referers
Results
Top Clicked Pages
Click Count Histogram
referrer click_count
1 other-google 14361
2 other-empty 1543
3 Apache_Hadoop 901
4 Spark 725
5 other-other 587
6 other-wikipedia 350
7 Akka_(toolkit) 222
8 other-bing 186
9 MapReduce 181
10 Apache_Mahout 150
15 s 820 ms | 10 rows returned
Refreshed just now

Common SQL Commands

COPY INTO

- Grab raw data and put into Delta
- The Extract of ETL

```
COPY INTO my_table
FROM '/path/to/files'
FILEFORMAT = <format>
FORMAT_OPTIONS ('mergeSchema' = 'true')
COPY_OPTIONS ('mergeSchema' = 'true');
```

CREATE <entity> AS

- Create a Table or View
- The Transform in ETL

```
CREATE TABLE events
  USING DELTA
  AS (
      SELECT *
    FROM raw_events
    WHERE ...
  )
```

Let's practice!

DATABRICKS CONCEPTS

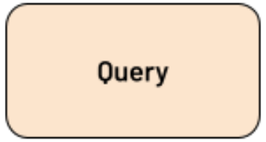
Databricks SQL queries and dashboards

DATABRICKS CONCEPTS

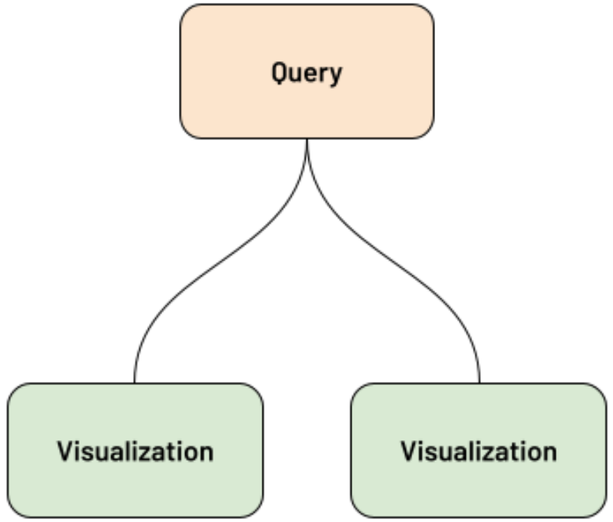
Kevin Barlow
Data Practitioner

datacamp

Databricks SQL Assets

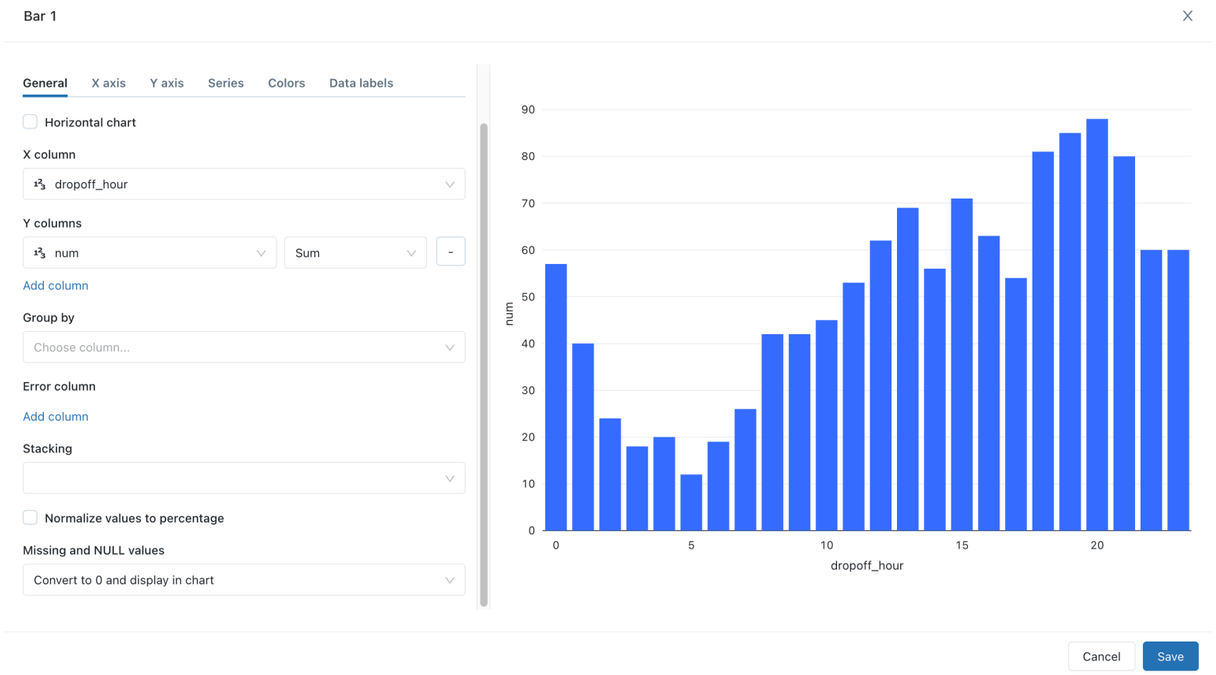


Databricks SQL Assets

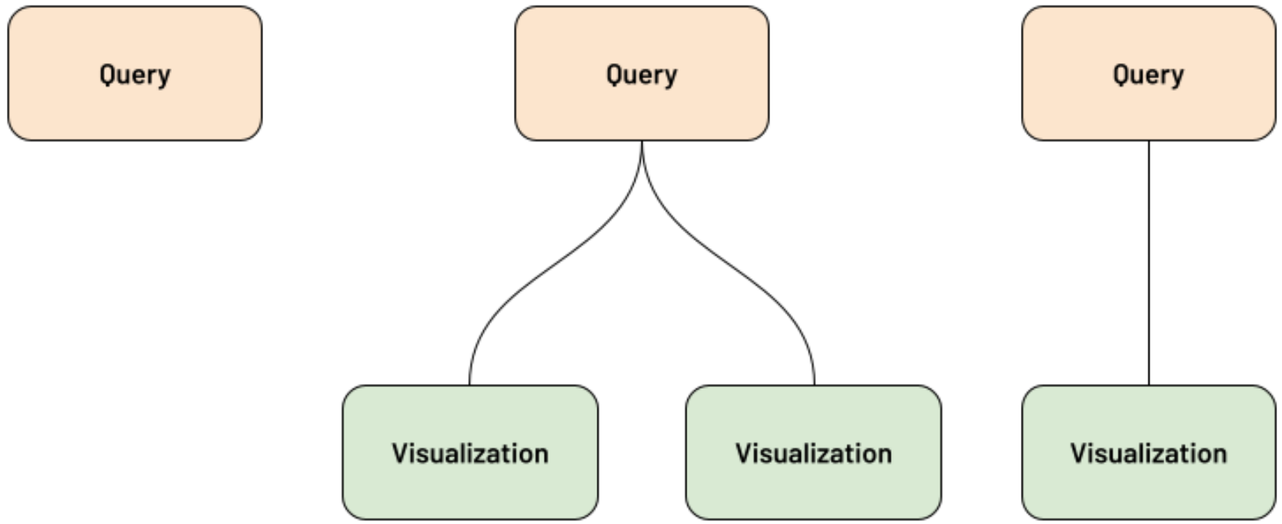


Visualizations

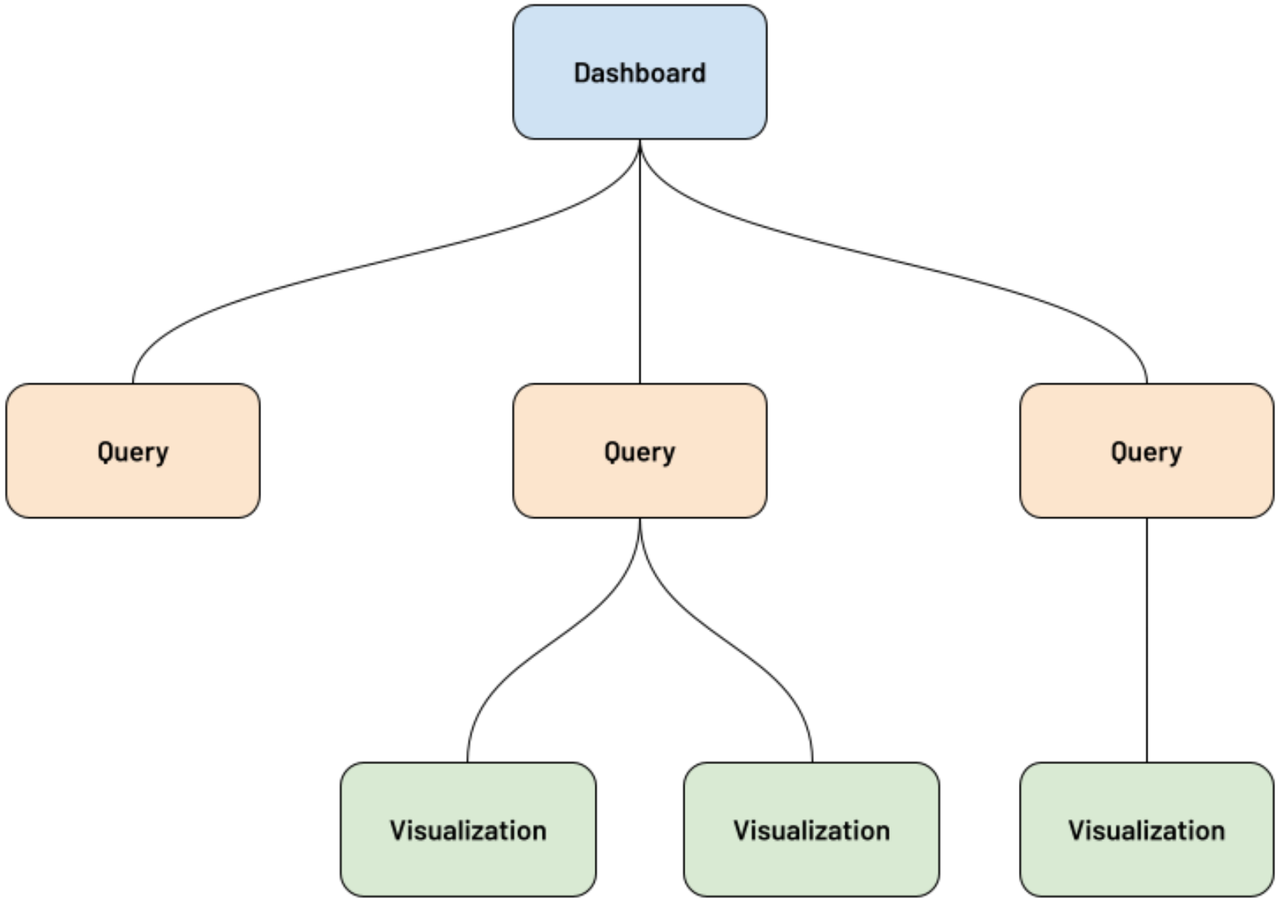
- Lightweight, in-platform visualizations
- Support for standard visual types
- Ability to quickly comprehend data in a graphical way



Databricks SQL Assets

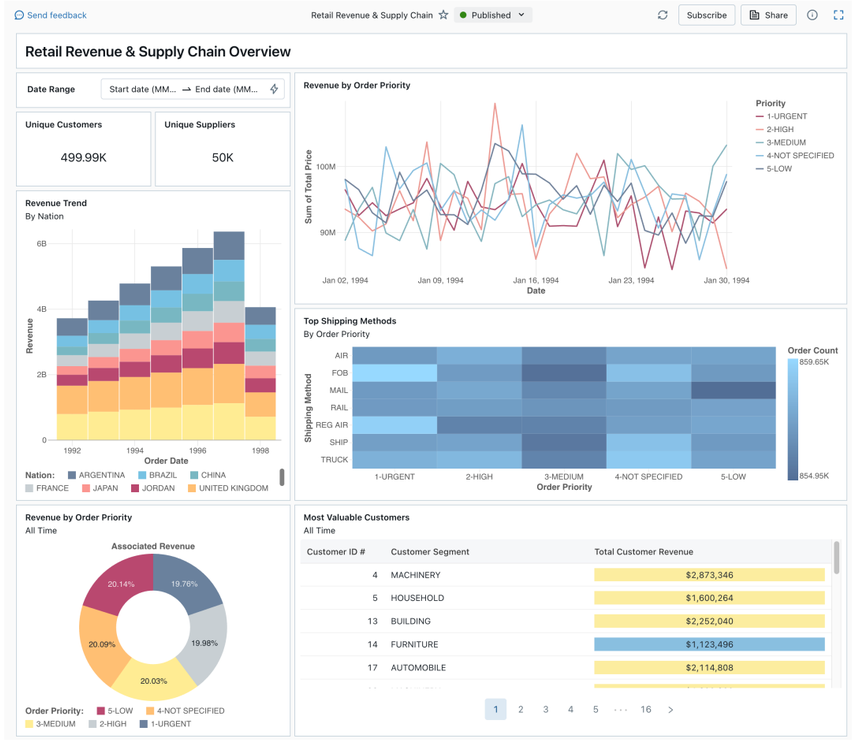


Databricks SQL Assets



Dashboards

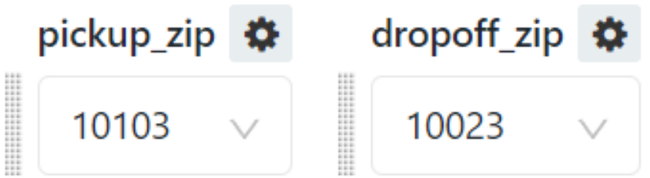
- Lightweight, easily created dashboards
- Ability to share and govern across your organization
- Scalable and performant



Query Filters

Filters

- Interactive query / dashboard components that allow the user to reduce the size of the result dataset
- Works on the client-side, so is very fast
- Supports single select, multi-select, text fields, and date / time pickers

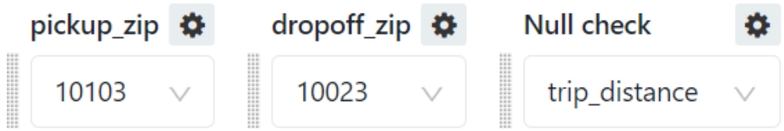



```sql
SELECT *
FROM nyctaxi.trips
WHERE pickup_zip = 10103
AND dropoff_zip = 10023
```

Query Parameters

Parameters

- More flexible than filters, and supports more kinds of selectors
- Allow the user to provide a value that is input into the underlying SQL query text
- Created in the query by using the `{{ }}` syntax




```
SELECT *
FROM nyctaxi.trips
WHERE pickup_zip = 10103
AND dropoff_zip = 10023
AND {{ nullCheck }} IS NOT NULL
```

Let's practice!

DATABRICKS CONCEPTS

Creating a Databricks SQL Dashboard

DATABRICKS CONCEPTS

Kevin Barlow
Data Practitioner

datacamp

Let's practice!

DATABRICKS CONCEPTS