

# **Introduction To Databricks SQL Workshop**

## About this workshop

Thank you for attending this workshop. This workshop is an introductory hands-on workshop that has been designed for administrators and data consumers (data analyst) to give you a foundational understanding on what and how to leverage Databricks SQL. This workshop doesn't not go into implementation details or best practices of building a lakehouse or Databricks SQL. For more information about deep dive topics please visit Databricks Academy or contact your Databricks representative.

In this workshop, we will give you an overview of Databricks SQL, how administrators can setup Databricks SQL that can be used by data consumers to gain insight from the lake house.

## Agenda

- Overview of the Databricks SQL workspace
- Import sample dashboard
- Learn about SQL Endpoints
- Explore database and tables, and data access controls
- Create & execute queries and visualizations using Databricks SQL
- Monitor a SQL Endpoint, query history, query performance
- Configure alerts
- Create and share dashboards
- Demo: Connect BI tools to SQL endpoints (Tableau)

# Introduction to Databricks SQL

As organizations adopt cloud data lakehouses, which are the best of breed capabilities of data warehouses and data lakes, many non-technical users, such as data analysts, require expert support from data engineers or data administrators to gain access to lakehouse data to support analytics or BI initiatives. However, this form of collaboration is time-consuming and counterproductive as data engineers or data administrators can focus their efforts on other initiatives.

To enable quick access to the lakehouse, Databricks SQL allows the data analyst to perform BI and SQL workloads on a multi-cloud lakehouse architecture. This new service consists of four core components: A dedicated SQL-native workspace, built-in connectors to common BI tools, computation management, and governance and administration capabilities.

Before we get started, let's review some key concepts and common terminology you will need to understand as you work through this lab.

## Interface:

Users can access Databricks SQL using a graphical interface to administer, create queries, dashboards, visualizations, configure SQL Endpoints, and alerts. Databricks SQL has a REST API interface that allows users to automate tasks on Databricks SQL objects. Please note, REST API functionality will not be covered in this lab.

## Data Management – how to build visualizations, dashboards

**Visualization:** A graphical presentation of the result of running a query.

**Dashboard:** A presentation of query visualizations and commentary.

**Alert:** A notification that a field returned by a query has reached a threshold.

## Computation Management – how to run SQL queries in Databricks SQL

**Query:** A valid SQL statement.

**SQL endpoint:** A computation resource on which SQL queries execute.

**Serverless SQL:** instant compute for BI and SQL workloads, with minimal management required and capacity optimizations.

**Query history:** A list of executed queries and their performance characteristics.

**Governance and Administration** – how to manage Databricks SQL users, groups and access to data

**User and group:** A user is a unique individual who has access to the system. A group is a collection of users.

**Access control list:** A set of permissions attached to a principal that requires access to an object. An ACL entry specifies the object and the actions allowed on the object. Each entry in an ACL specifies a principal, action type, and object.

**Personal access token:** An opaque string is used to authenticate to the REST API and by Connect BI tools to Databricks to connect to SQL endpoints.

## Lab 1: Login Databricks SQL workspace

**Duration:** 5 minutes

**Objective:** In this lab, you will log in to Databricks SQL and get familiar with the workspace UI.

As we get started with Databricks SQL, first you will navigate the workspace UI.

**Quick Fact:** With native collaboration capabilities, the Databricks platform provides an integrated, unified, web-based user experience for data engineering, data science, machine learning and SQL analytics on a secure and scalable workspace. In this lab, our focus will be SQL analytics. The URL link provided below will automatically take you to the Databricks SQL workspace.

For this lab, we have provisioned a fully functional Databricks Platform, enabled with Data Engineering, Data Science, Machine Learning, and Databricks SQL. Feel free to navigate through the various user experiences after this workshop. We ask that you not configure, edit or access anything not related to the scope of this workshop.

**Note:** Databricks SQL supports all major public clouds. Your experience with Databricks SQL doesn't change if you're using AWS, Azure or GCP.

### Login to Databricks SQL

To access the lab environment:

**Databricks SQL Workspace:** <https://<databricks-instance>/sql>

**User: (instructor provided)**

**Password: (instructor provided)**

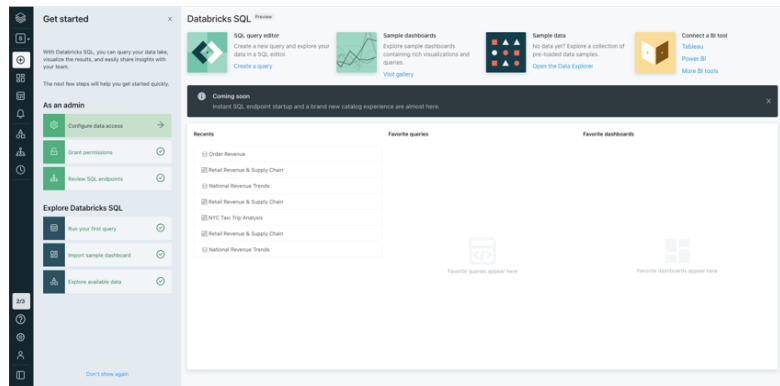
For this lab, you will log in as an Administrator, to get full exposure to the product. In a production or dev environment, an administrator will configure the appropriate groups with permission/privileges associated with your role.

Once you're logged in, Databricks SQL presents a landing page.

From this landing page, on the left, is a **getting started** guide for quick and easy onboarding for both administrator and data consumers. (**Do Not Close** this window – we will use this shortly).

At the top of the page, you can create a query, view sample dashboards, explore data or connect to a BI tool. We will explore these in the coming labs.

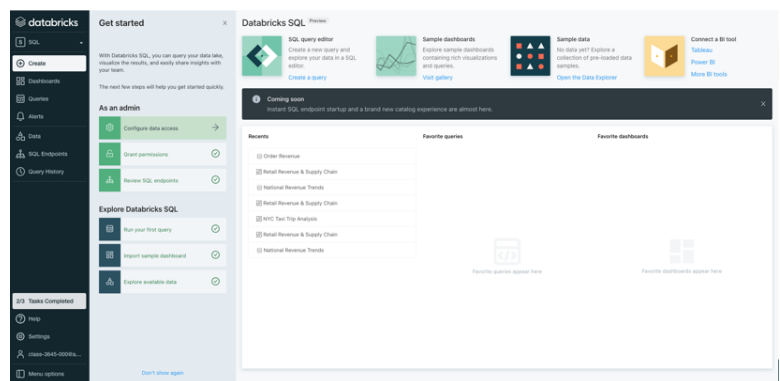
The middle of the page shows recently accessed objects (queries, visualizations, dashboards). Quickly access your favorite queries or dashboards.



**Hover your mouse** over the left navigation pane (it should automatically expand).

This navigation bar gets you to access dashboards, queries, alerts, data exploration, configure SQL Endpoints, view Query History. Towards the bottom, you will see help, additional settings.

Click on **Menu Options** and click on **Expand**



## Lab 2: Import a Sample Dashboard

**Duration:** 5 minutes

**Objective:** In this lab, you will import a sample dashboard. This newly provisioned environment has no sample data or dashboards to work with.

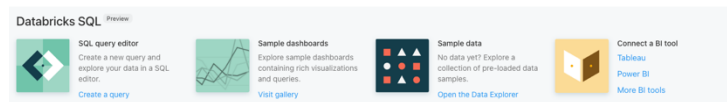
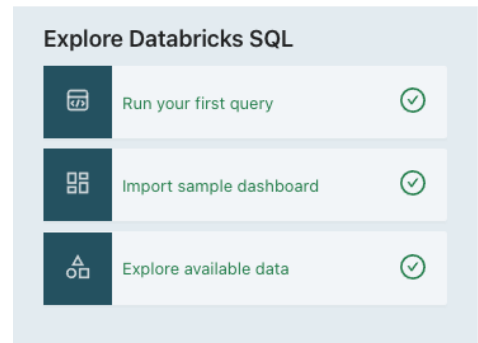
For this lab, we will import and interact with dashboards in Databricks SQL using the Sample Dashboard Gallery. These dashboards illustrate some of the rich visualizations that Databricks SQL provides for deriving insights. These dashboards use data that is already available in your workspace and rely on a compute resource (called a SQL endpoint) that is already configured.

There are two easy ways to import a sample dashboard. You may choose either:

From the get started navigation, under **Explore Databricks SQL** click on **"Import sample dashboard"**

Click on the **Import sample dashboard**

Alternatively, from the top of the screen, click **Visit Gallery**.



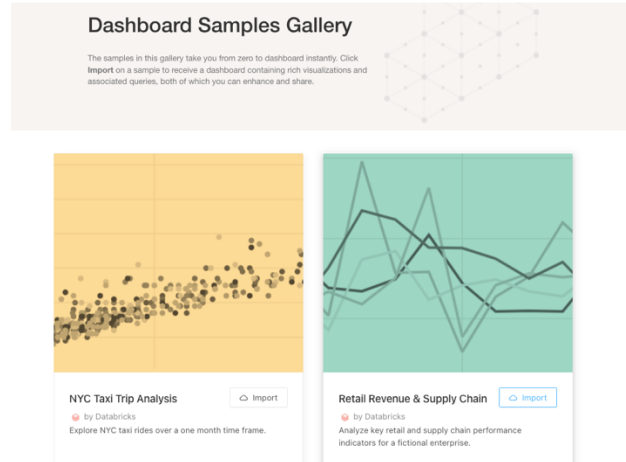
There are two sample dashboards to choose from. Let's import the **"Retail Revenue & Supply Chain"** sample.

This sample dashboard provides analysis and insight into retail and supply chain performance for a fictional enterprise.

Click "Import" (Highlighted in blue)

It should take a few minutes to import the retail dashboard.

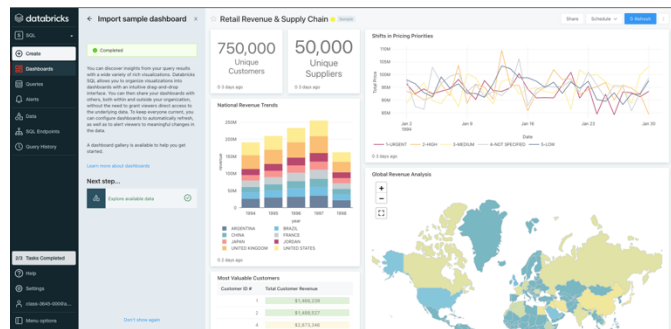
This task imports all the data, queries, visualization, and dashboards for this lab.



Once imported, you will see a dashboard with various visualizations imported.

Your screen should be like the image on the right.

We will explore more visualizations and dashboards in upcoming labs. For now, we have all the data we need to learn more about Databricks SQL features and functionalities.



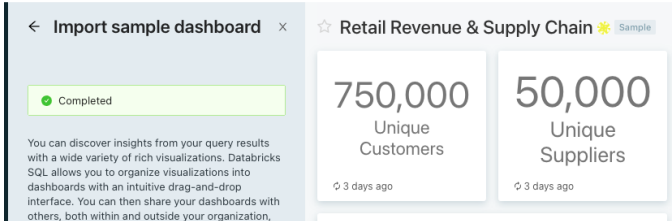
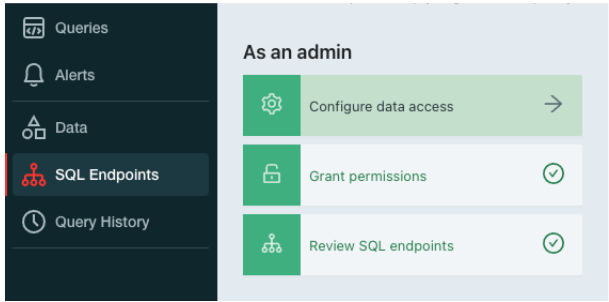
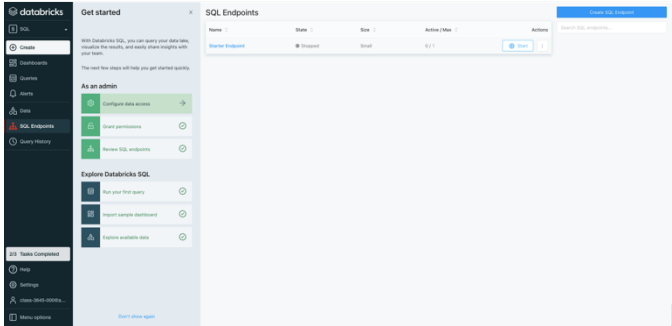
# Lab 3: Review SQL Endpoints & Understand Computation Resources

**Duration:** 10 minutes

**Objective:** In this lab, you will learn about computation resources.

A SQL endpoint is a computation resource on which SQL queries execute within the Databricks platform. When you create a query in Databricks SQL or use an existing BI tool (Tableau, PowerBI, Looker, Qlik), the data object is converted and optimized to run natively on the Databricks compute cluster.

For this lab, we have configured a small endpoint. To handle more complex workloads, you can easily increase its size (to reduce latency) or the number of underlying clusters (to handle more concurrent users). We will then review the configured SQL endpoint and start the endpoint.

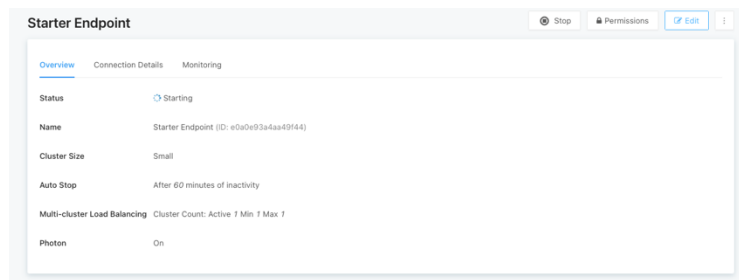
<p>To review SQL endpoints, in the onboarding, getting started navigation, click the back arrow ( ← )</p>	
<p>Under the “As an admin” click on <b>“Review SQL endpoints”</b>. This will take you to the SQL Endpoint configuration screen.</p> <p>Or from the navigation pane click SQL Endpoints</p>	
<p>You will notice a pre-configured SQL Endpoint—called “starter endpoint”—that is currently stopped. Go ahead and start the endpoint. Click “Start”. (Highlighted in blue)</p> <p>While this endpoint starts, hover over and click on “Starter Endpoint” to review the configuration.</p>	



When the configuration screen opens, you will notice the endpoint start.

Click “Edit” to see the pre-configured settings.

**!!**  
**Do NOT change any settings or configurations. Please ask !!**



When you edit (or create) a SQL Endpoint you specify the

**Name:** Starter Endpoint

**Cluster Size:** for this lab, we chose a **small** cluster size (keeps the cost down!)

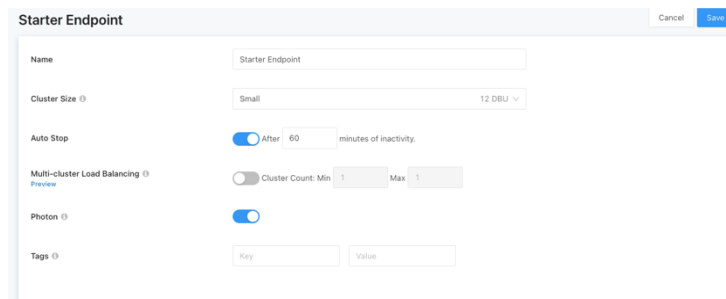
However, to reduce the latency of queries, you can increase the size. Larger sized clusters have a larger coordinator and doubles the number of cluster workers.

**(Please don't change the cluster size!)**

**Auto Stop:** whether the endpoint stops if it's idle for the specified number of minutes.

**Tags:** allow you to easily monitor the cost of cloud resources used by various groups in your organization. You can specify tags as key-value pairs when you create an endpoint, and Databricks applies these tags to cloud resources.

**Click “Cancel” to go back to Endpoint settings**



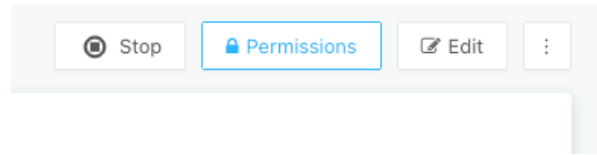
**Multi-cluster Load Balancing:** the minimum and a maximum number of clusters over which queries sent to the endpoint are distributed.

**Photon:** queries are executed on the Photon native vectorized engine that speeds up query execution. By default, this feature is turned on.

**About Photon:** the native vectorized query engine on Databricks, written to be directly compatible with Apache Spark APIs, so it works with your existing code. It is developed in C++ to take advantage of modern hardware and uses the latest techniques in vectorized query processing to capitalize on data- and instruction-level parallelism in CPUs, enhancing performance on real-world data and applications—all natively on your data lake. Photon is part of a high-performance runtime that runs your existing SQL and DataFrame API calls faster and reduces your total cost per workload.

As an administrator, you can control **SQL endpoint** access.

**Click on “Permissions”**



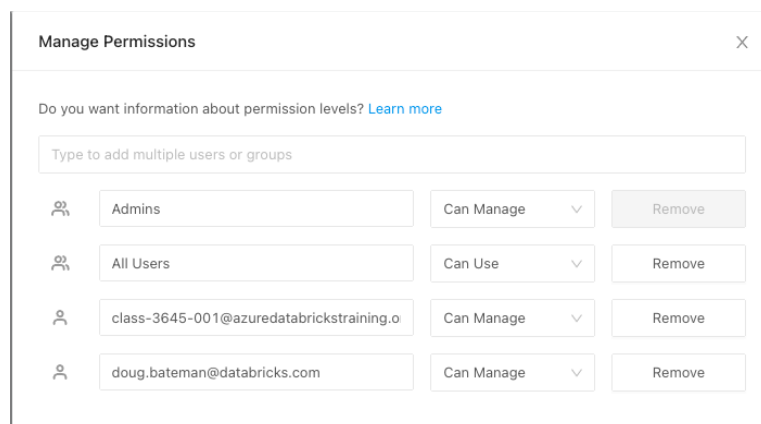
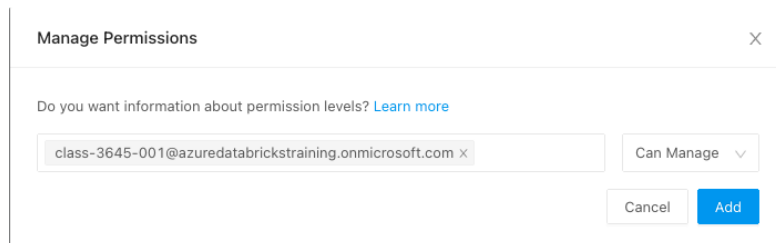
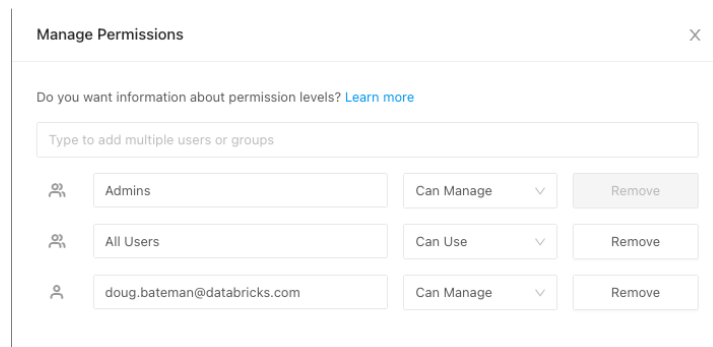
With SQL endpoint access control, individual permissions determine a user's abilities.

There are three permission levels for a SQL endpoint: **No Permissions**, **Can Use**, and **Can Manage**.

For this exercise, search for your **username** (used for login) and provide **“Can Manage”** permissions and click **“Add”**

You will see the new **permissions** assigned.

**Click (X)** to close out permissions and return to the SQL Endpoint Screen.



Click on the **Connection Details** tab.

The screenshot shows the 'Connection Details' tab for a Databricks Starter Endpoint. It displays the following configuration:

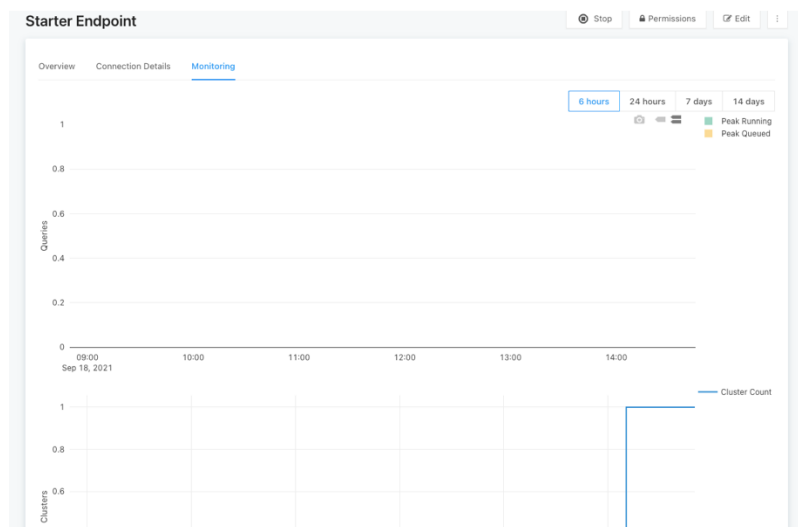
- Server Hostname: adb-626557336600866.6.azuredatabricks.net
- Port: 443
- Protocol: https
- HTTP Path: /sql/1.0/endpoints/e0a0e93a4aa49f44
- JDBC URL: jdbc:spark://adb-626557336600866.6.azuredatabricks.net:443/default;transportMode=http;ssl=1;AuthMech=3;httpPath=/sql/1.0/endpoints/e0a0e93a4aa49f44;

At the bottom, there are icons for Tableau, Power BI, and More BI Tools, along with a link to 'Create a personal access token'.

Click on the **Monitoring** tab.

Administrators can examine the number of queries handled by the endpoint and the number of clusters allocated to the endpoint.

When executing a query, we will navigate back to the SQL Endpoint monitoring tab to understand the runtime statistics.



Up to this point, you've seen and understood how SQL endpoints work. The endpoint you're working with in this lab uses a compute resource within a cloud provider associated with this account (AWS, Azure, GCP). At Databricks, we call this a Classic SQL Endpoint, akin to a customer managed compute resource.

We've recently introduced a Databricks-managed SQL endpoint, called **Serverless SQL Endpoints**. This type of endpoint is a compute resource that is managed in the Databricks cloud. As an administrator or a data consumer, all the complexity of managing the underlying compute cluster is abstracted away from the user. Therefore, data consumers can focus on creating insights with Databricks SQL. Serverless SQL endpoints simplify SQL endpoint management and accelerate launch times. You can use them to run Databricks SQL queries just like you do with a Classic SQL endpoint.

# Lab 4: Explore database and tables, and data access controls

**Duration:** 10 minutes

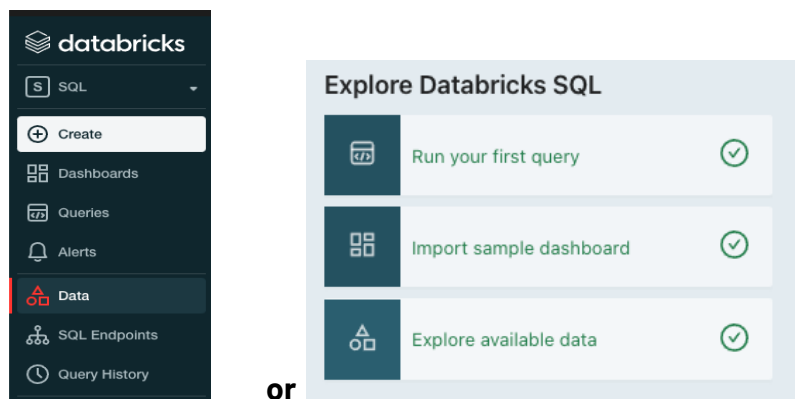
**Objective:** In this lab, you will learn to explore data and manage permissions on databases and tables.

Within Databricks SQL, the data explorer feature lets you easily explore and manage permissions on databases and tables. Data users can view schema details, preview sample data, and see table details and properties. Administrators can view and change owners, while admins and data object owners can grant and revoke permissions.

Data Exploration gives non-technical users a visual mechanism to view and understand the data object in the lakehouse.

To explore data, from the navigation sidebar, select **Data**.

Or if you're using the getting started quick start, under "Explore Databricks SQL" click on "Explore available data"



This will take you to the **Data Explorer** screen.

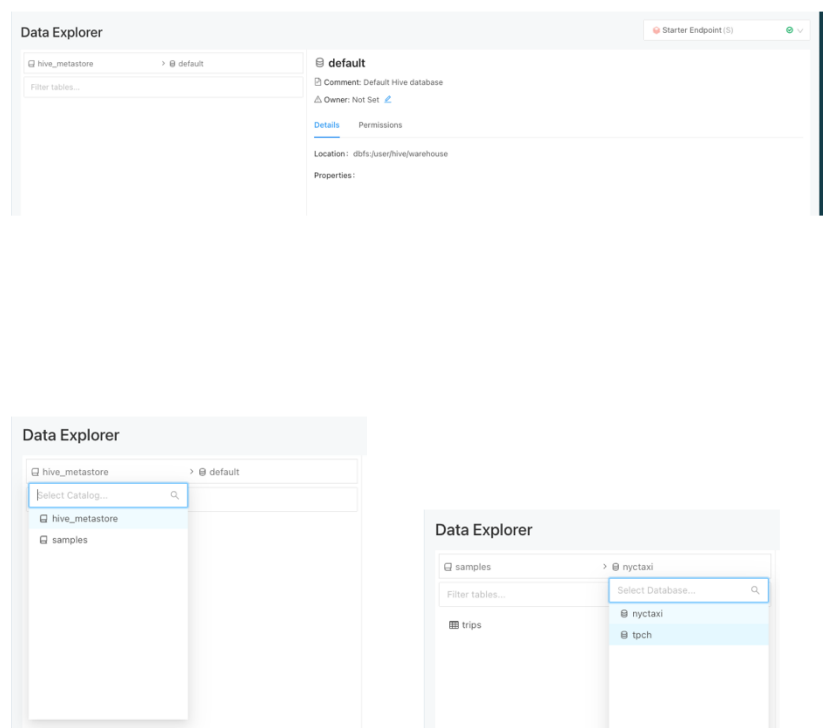
**Note:** make sure your SQL Endpoint is running. If not, start the endpoint.

By default, you will see the **hive\_metastore** and **default** database.

However, when we imported the data and visualizations, the process created a new database and tables. Let's change to the correct database.

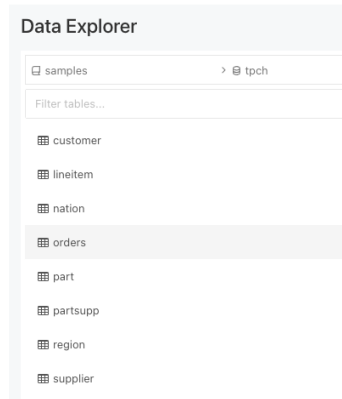
To change, hover over "**hive\_metastore**" and click. A drop-down menu appears with all databases in the catalog.

Choose the "**samples**" catalog and select the "**tcph**" database.



You should see **8 tables** in this database.

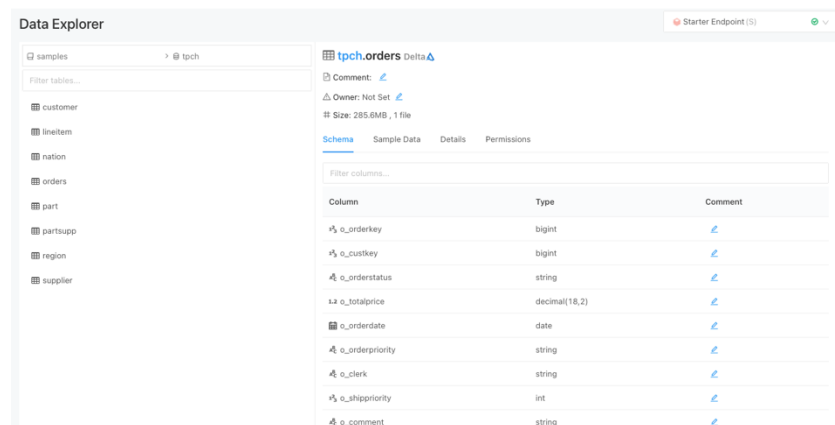
Select the **orders** table



On the right-hand side pane, you will see more details about the **orders** data object.

This **orders** table is a Delta Lake table, built on an open-source, open-format storage layer that brings ACID transactions, scalable metadata handling, and unified streaming and batch data processing on top of existing data lakes.

Let's explore some of the features of data exploration.



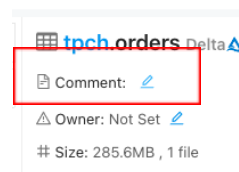
First, you can update the object with a **comment**. This provides the context of the object and makes it discoverable to other data consumers.

Let's **add a comment**.

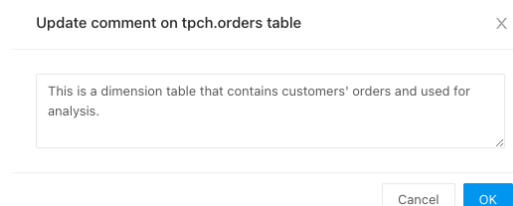
Click on the pencil icon next to the comment tag and update with the following text:

"This is a dimension table that contains customers' orders and is used for analysis."

Step 1: Click the **pencil icon** next to comment



Step 2: Add a comment and click "OK"



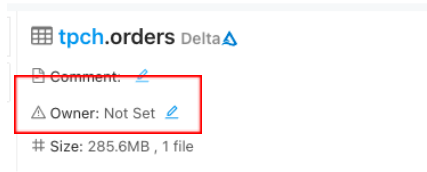
Next, we can set the “owner” of the data object.

By setting **ownership**, you’re providing governance that makes it easy for data consumers to contact the owner for any additional information about the data object.

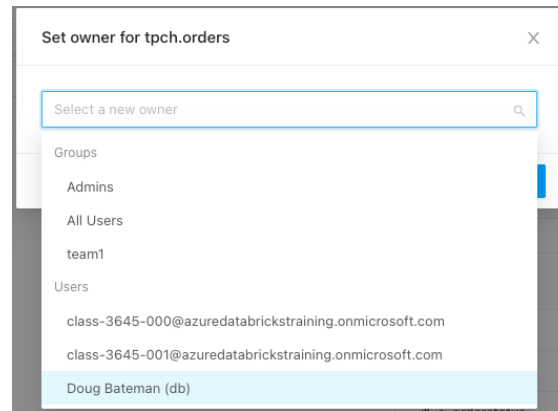
You can provide ownership to a single user or a group.

Click on the **pencil icon**, next to the **Owner**, and set the owner

Step 1: Click the **pencil icon** next to “Owner”



Step 2: Select the user “Doug Batemen”. Additionally, you can select a group or individual user. Then click “ok”



Next, let’s look at the additional details such as schema, sample data, details, and permissions.

**Schema** provides the user the structure of the data object, this includes the data types.

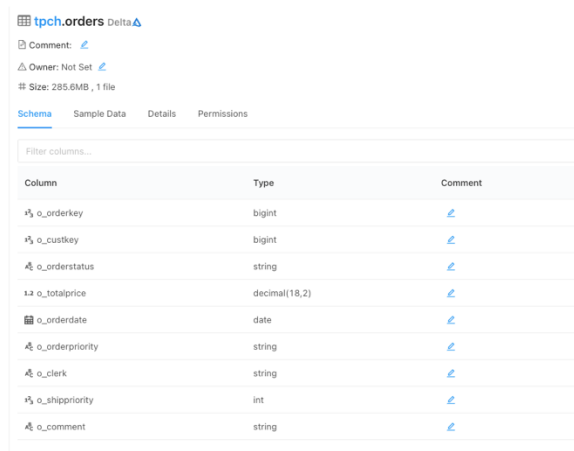
As a user, you can also provide a comment for each metadata column. For example, you can describe the contents of a column:

“Total price for the entire order per customer”

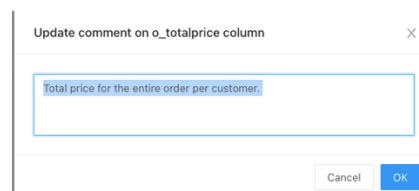
You can add comments to all the metadata columns within the data object.

As you’ve probably noticed, you’re enriching the data object with details making the data set discoverable and understandable.

Step 1: Click the **pencil icon** for o\_total\_price



Step 2: Add a comment and click “ok”



Next, click on the **Sample Data** tab.

Data stored in a data lake can be complex and difficult for analysts to view the data.

**Sample Data** provides a visual view of the data stored in the table. If the table contains a complex data type, such as nested columns, the data analyst will be able to view them in the sample data tab.

o_orderkey	o_custkey	o_orderstatus	o_totalprice	o_orderdate	o_orderpriority	o_clerk
11396166	179329	O	187683.1	1996-06-22	4-NOT SPECIFIED	Clerk#000004689
11396167	473245	F	117554.58	1994-07-20	1-URGENT	Clerk#000001856
11396192	77549	O	47611.56	1996-01-03	3-MEDIUM	Clerk#000001450
11396193	610651	F	121869.26	1994-11-29	5-LOW	Clerk#000003696

Click the **Details** tab to view the location of the table files, the type of table, and table properties.

Schema Sample Data **Details** Permissions

Location: dbfs:/databricks-datasets/tpch/delta-001/orders

Type: EXTERNAL

Table Properties: delta.checkpoint.writeStatsAsJson=false  
delta.checkpoint.writeStatsAsStruct=true  
delta.minReaderVersion=1  
delta.minWriterVersion=2

Lastly, click on the **Permissions** tab.

Here a data owner or administrator can grant or revoke permission and privileges for the data object.

Step 1: Click on the **Permissions** tab of the **tpch.orders** object

Schema Sample Data Details **Permissions**

Grant Revoke

☐ Principal ☐ Privilege ☐ Object

No permissions granted yet.

Step 2: From the drop-down menu add a user or group. Select "All Users" from the drop-down.

Grant on tpch.orders

Type to add multiple users or groups

☐ ALL gives all privileges (is translated into all the other privileges).  
☐ SELECT gives read access to an object.  
☐ MODIFY gives ability to add, delete, and modify data to or from an object.  
☐ READ\_METADATA gives ability to view an object and its metadata.

Cancel OK

Grant on tpch.orders

Type to add multiple users or groups

Groups  
Admins  
All Users  
team1  
Users  
class-3645-000@azuredatabrickstraining.onmicrosoft.com  
class-3645-001@azuredatabrickstraining.onmicrosoft.com  
Doug Bateman (db)

Step 3: Choose type of permission for the user or group. Choose Select and read\_metadata. Click "OK"

Grant on tpch.orders

X

All Users X

☐ ALL gives all privileges (is translated into all the other privileges).
 ☒ SELECT gives read access to an object.
 ☐ MODIFY gives ability to add, delete, and modify data to or from an object.
 ☒ READ\_METADATA gives ability to view an object and its metadata.

Cancel

OK

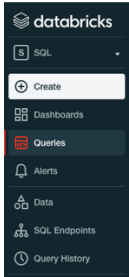
This concludes the section on data exploration. **Fun fact:** By completing the task of data exploration, you've had a sneak peek into a few features of a new product Databricks is releasing called **Unity Catalog!** **Unity Catalog** provides a fine-grained governance and centralized security model for data lakes across clouds – based on the ANSI SQL open standard.

## Lab 5: Create & execute queries and visualizations

**Duration:** 10 minutes

**Objective:** In this lab, you will learn to use SQL commands to perform ad-hoc and exploratory data analysis on your data lake and quickly develop agile dashboards to test and validate business requirements.

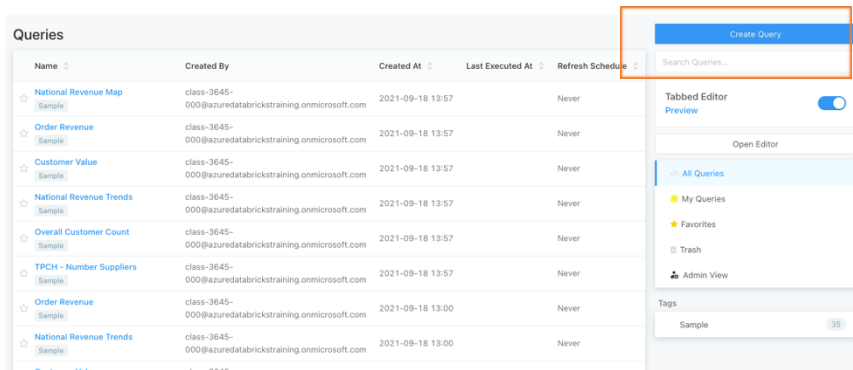
To perform ad-hoc and exploratory data analysis, hover to the left-hand navigation bar and click on **Queries**.



This takes you to the Queries page.

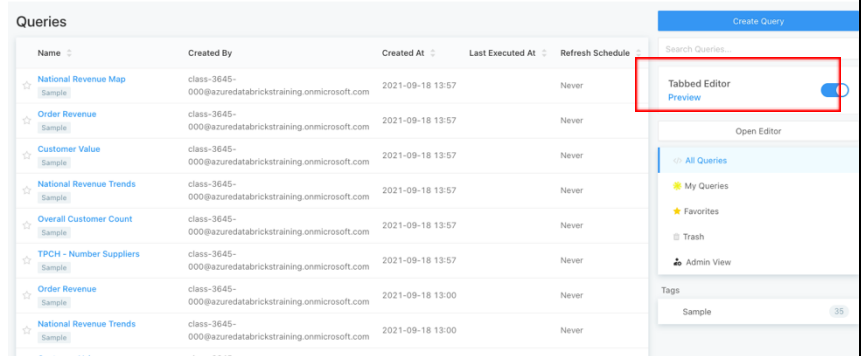
For each query listed, you can see who created it, when it was created, last executed or if there is a refresh schedule attached.

Within this view, you can **create a new query** or **search for existing queries**.

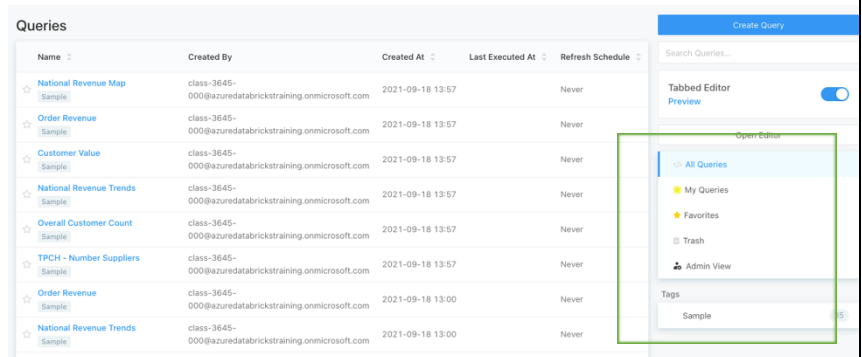




First, enable the **tabbed editor**. This is a preview feature that allows you to open multiple queries without opening a new query editor.



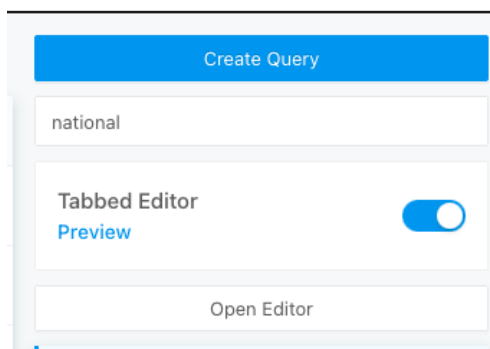
Just below this preview feature, you can view saved queries in various ways. You can view queries you've created by clicking **"My Queries"** or your **"favorite queries"** or starred queries or deleted queries (sent to **trash**). The **admin view** gives you access to see all queries created and delete queries in this workspace. However, an admin can't edit a query if it is not shared with the admin.



Lastly, you can filter the query view by **tags**. For example, if you've tagged your queries as **"Orders"**, you can filter for only those queries.

In the search bar, search for **"National Revenue Trends"**.

Databricks SQL automatically starts filtering for queries based on the search term.



Click the third query in the list, called **"National Revenue Trends"**

This opens the query editor with an already saved query.

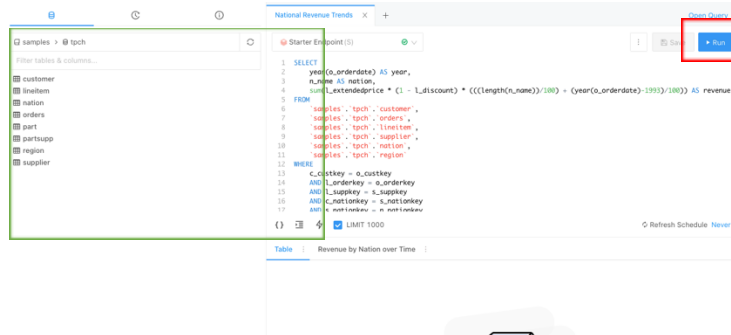
Queries				
Name	Created By	Created At	Last Executed At	Refresh Schedule
★ National Revenue Map Sample	class-3645-000@azuredatabrickstraining.onmicrosoft.com	2021-09-18 13:57		Never
★ National Revenue Map Sample	class-3645-000@azuredatabrickstraining.onmicrosoft.com	2021-09-16 07:20		Never
★ National Revenue Trends Sample	class-3645-000@azuredatabrickstraining.onmicrosoft.com	2021-09-18 13:00		Never
★ National Revenue Trends Sample	class-3645-000@azuredatabrickstraining.onmicrosoft.com	2021-09-16 07:20		Never
★ National Revenue Trends Sample	class-3645-000@azuredatabrickstraining.onmicrosoft.com	2021-09-18 13:57		Never
★ National Revenue Map Sample	class-3645-000@azuredatabrickstraining.onmicrosoft.com	2021-09-18 13:00		Never

Let's navigate through the query editor.

If you have metadata read permission, the **schema browser** displays the available databases and tables. This should open to the catalog, database you're working in. If not, change to open the **samples tpch** tables.

Go ahead and run the query shown – click **"Run"**, in the top right-hand corner.

(Note: the 3 dots, on the left next to the Save icon allow you to share, clone or trash the query)



When the query runs, it executes on a compute cluster leveraging the Photon engine.

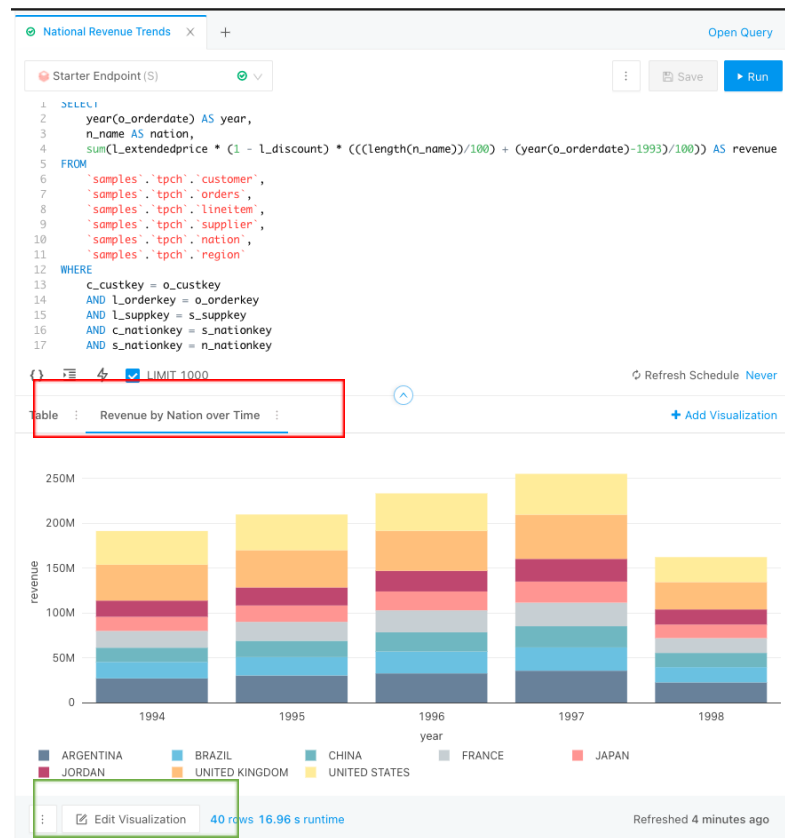
When the query returns, in the lower third of the screen, you can see the table results based on your query conditions.

The query editor has Auto Complete, which makes writing queries faster. Auto Complete can complete schema tokens, query syntax identifiers (like SELECT and JOIN), and the titles of query snippets.

Click on the “Revenue by Nation over Time” tab – this is the visualization created for this query.

Click to "Edit Visualization" at the bottom of the screen.

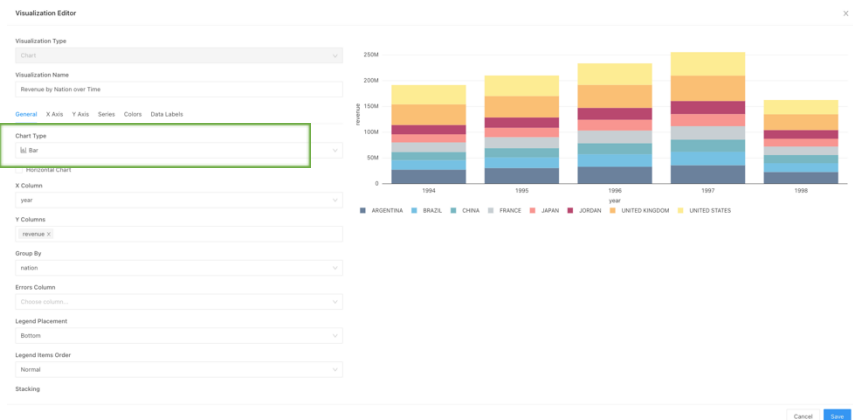
(Note: the 3 dots, on the left allow you to download the data as CSV, TSV or Excel)



The visualization helps you understand the different aspects of your queries.

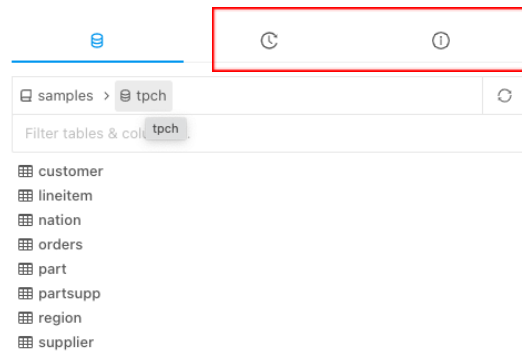
Databricks SQL supports several different types of visualizations. A table is the default. Spend a minute looking thru the various types under “**Chart Type**”. Based on the type of chart you need to create, there are various intuitive properties you can configure for your visualization.

Click (X) or Cancel to exit out of the visualization editor.



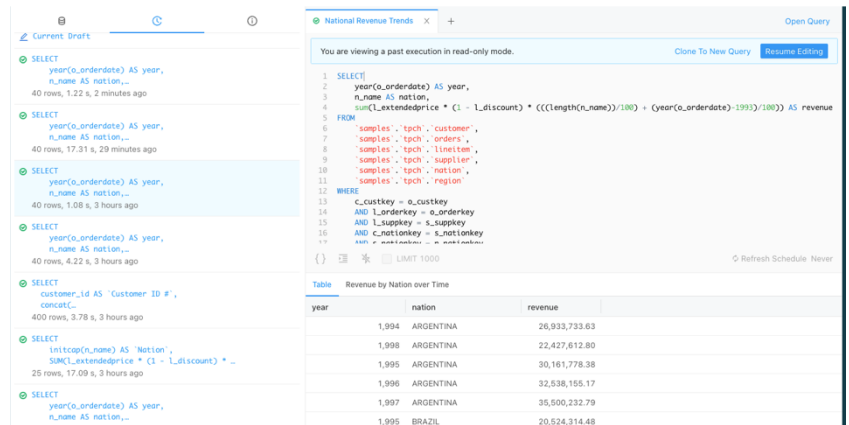
Returning to the editor, under the schema browser, you can view **past query executions** or **query info**.

Click on the first clock icon.



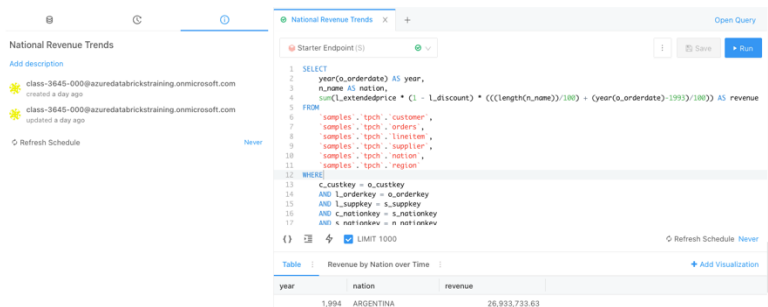
Past executions show you all the queries that ran in the query editor, in read-only mode. This is helpful to step back-in-time to see queries executed. You can clone to a new query or resume editing the query.

Try it out by clicking any query in the current view. To return to the current query, click on **Current Draft**.

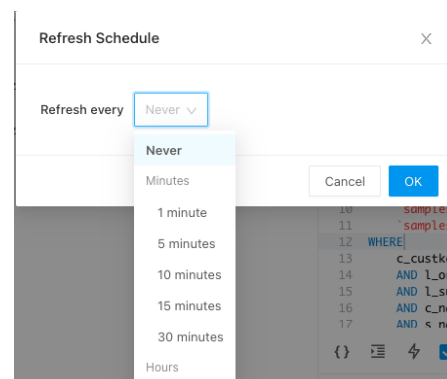


The query information view tells you when the query was created or updated, click the query info tab.

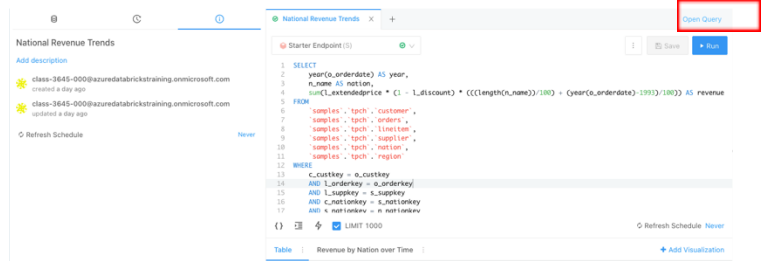
Here, the query can be scheduled to be refreshed.



Click on "Never" to bring up the schedule window. You can choose a schedule in minutes or hours. Click cancel to return to the query editor.

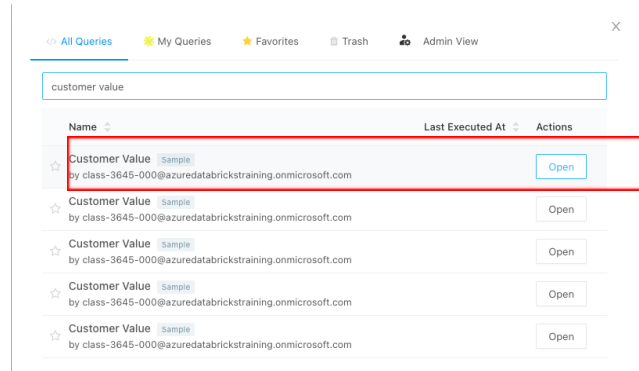


When the tabbed editor is enabled, you can easily switch between multiple queries. Click **Open Query** on the top left corner.



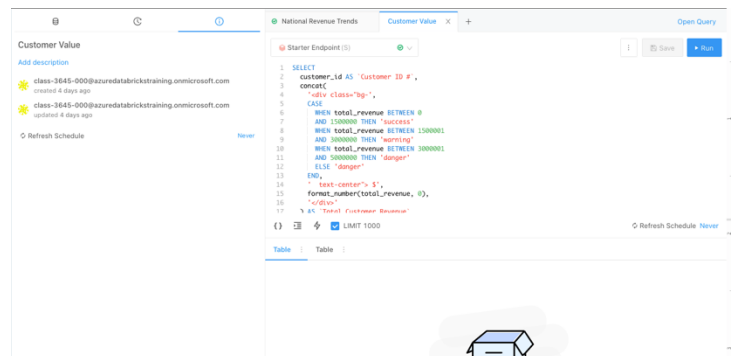
When the query viewer opens, you can select a query you'd like to view or edit.

Search for **"Customer Value"** and click open.



The customer value query will open. Notice, the revenue trends query on the left.

Go ahead and run the Customer Value query.



# Lab 6: Monitor a SQL Endpoint, query history, query performance

**Duration:** 10 minutes

**Objective:** Now that you've built and ran query, you will understand query history, monitor query performance, and monitor a SQL endpoint.

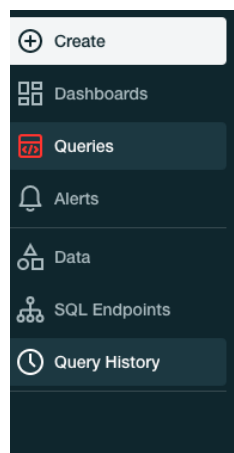
As SQL queries are critical to supporting business decisions, it's imperative that the queries run efficiently. Ineffective queries conditions, such as joins, can lead to performance problems.

To start, let's look at query history and performance.

From the navigation bar, click on **"Query History"**

The query history shows SQL queries performed using SQL endpoints. Within Query History, you can debug performance issues with the query.

Note: these steps are performed via the Databricks SQL UI but you (or your admin) can work with query history via the API.



You will arrive at the query history page – this will list all the queries executed on the SQL endpoint.

At the top of the screen, you can filter by user, date range, endpoint type or query status.

Filter by data range by hour. Click or hover over the lightning icon, and choose **"last hour"**. This will filter out all other queries.

A screenshot of the Databricks SQL Query History page. The page shows a table of queries with columns: Query, SQL Endpoint, Started At, Duration, and User. The table is filtered by 'Last hour' and 'All SQL endpoints'. The queries listed are various system queries like 'Listing columns', 'Listing tables', 'SELECT year', 'Listing columns', 'Listing tables', 'describe database extended', 'show tables in', 'show grant on DATABASE', 'show databases', 'Listing columns', 'Listing tables', 'SELECT year', and 'SELECT customer\_id'. The duration of each query is shown in milliseconds or seconds. The user for all queries is 'class-3645-000@azuredatabricks...'.

Query	SQL Endpoint	Started At	Duration	User
Listing columns 'catalog' : null, schemaPattern : default, tablePattern : null, c...	Starter Endpoint	2021-09-24 13:36	53 ms	class-3645-000@azuredatabricks...train...
Listing tables 'catalog' : null, schemaPattern : default, tableTypes : null, tabl...	Starter Endpoint	2021-09-24 13:36	270 ms	class-3645-000@azuredatabricks...train...
Listing columns 'catalog' : null, schemaPattern : default, tableTypes : null, tabl...	Starter Endpoint	2021-09-24 13:33	136 ms	class-3645-000@azuredatabricks...train...
Listing tables 'catalog' : null, schemaPattern : default, tableTypes : null, tabl...	Starter Endpoint	2021-09-24 13:33	75 ms	class-3645-000@azuredatabricks...train...
SELECT year(o_orderdate) AS year, n_name AS nation, sum(l_extendedprice * (1 - l...	Starter Endpoint	2021-09-24 13:32	996 ms	class-3645-000@azuredatabricks...train...
Listing columns 'catalog' : null, schemaPattern : default, tablePattern : null, c...	Starter Endpoint	2021-09-24 13:32	55 ms	class-3645-000@azuredatabricks...train...
Listing tables 'catalog' : null, schemaPattern : default, tableTypes : null, tabl...	Starter Endpoint	2021-09-24 13:32	89 ms	class-3645-000@azuredatabricks...train...
Listing columns 'catalog' : null, schemaPattern : default, tablePattern : null, c...	Starter Endpoint	2021-09-24 13:31	54 ms	class-3645-000@azuredatabricks...train...
Listing tables 'catalog' : null, schemaPattern : default, tableTypes : null, tabl...	Starter Endpoint	2021-09-24 13:31	109 ms	class-3645-000@azuredatabricks...train...
describe database extended 'default' -- user_id: {}	Starter Endpoint	2021-09-24 13:31	232 ms	class-3645-000@azuredatabricks...train...
show tables in 'default' -- user_id: {}	Starter Endpoint	2021-09-24 13:31	199 ms	class-3645-000@azuredatabricks...train...
show grant on DATABASE 'default' -- user_id: {}	Starter Endpoint	2021-09-24 13:31	228 ms	class-3645-000@azuredatabricks...train...
show databases -- user_id: {}	Starter Endpoint	2021-09-24 13:31	344 ms	class-3645-000@azuredatabricks...train...
Listing columns 'catalog' : null, schemaPattern : default, tablePattern : null, c...	Starter Endpoint	2021-09-24 13:31	71 ms	class-3645-000@azuredatabricks...train...
Listing tables 'catalog' : null, schemaPattern : default, tableTypes : null, tabl...	Starter Endpoint	2021-09-24 13:31	448 ms	class-3645-000@azuredatabricks...train...
SELECT year(o_orderdate) AS year, n_name AS nation, sum(l_extendedprice * (1 - l...	Starter Endpoint	2021-09-24 13:27	13.87 s	class-3645-000@azuredatabricks...train...
SELECT customer_id AS 'Customer ID #', concat('idiv class="bg-', CASE WHEN tota...	Starter Endpoint	2021-09-24 13:26	36.85 s	class-3645-000@azuredatabricks...train...

Since you're logged in as an admin, you can see the query history as it relates to getting database/table listings (metadata listing capabilities).

For this lab, click on the first "select" statement you see in the query history.

Query History				
Me (class-3645-000@azuredatabrickstraining.onmicrosoft.com)		Last hour	All SQL endpoints	Status
Query	SQL Endpoint	Started At	Duration	User
SELECT customer_id AS 'Customer ID #', concat('adv class="tp-') AS WHEN tota...	Starter Endpoint	2021-09-24 14:35	746 ms	class-3645-000@azuredatabrickstraining.onmicrosof...
SELECT year(o_orderdate) AS year, n_name AS nation, sum(l_extendedprice * (1 - l...	Starter Endpoint	2021-09-24 14:35	910 ms	class-3645-000@azuredatabrickstraining.onmicrosof...
SELECT year(o_orderdate) AS year, n_name AS nation, sum(l_extendedprice * (1 - l...	Starter Endpoint	2021-09-24 14:34	7.87 s	class-3645-000@azuredatabrickstraining.onmicrosof...
Listing columns 'catalog': null, schemaPattern: default, tablePattern: null, c...	Starter Endpoint	2021-09-24 14:34	38 ms	class-3645-000@azuredatabrickstraining.onmicrosof...
Listing tables 'catalog': null, schemaPattern: default, tableTypes: null, tabl...	Starter Endpoint	2021-09-24 14:34	165 ms	class-3645-000@azuredatabrickstraining.onmicrosof...
Listing columns 'catalog': null, schemaPattern: default, tablePattern: null, c...	Starter Endpoint	2021-09-24 13:36	53 ms	class-3645-000@azuredatabrickstraining.onmicrosof...
Listing tables 'catalog': null, schemaPattern: default, tableTypes: null, tabl...	Starter Endpoint	2021-09-24 13:36	270 ms	class-3645-000@azuredatabrickstraining.onmicrosof...

On the right, you will see the query details. The first tab gives you an overview of the query.

Additionally, you will see runtime statistics of the query - the status (finished, running, queued, failed), start and end time, duration, the user who ran the query and the SQL endpoint it ran against.

Additionally, Databricks SQL can generate a query plan. You will see in the **Details** field click the **Open link** to display the plan.

**Note: this opens in a new window**

The Open link is available only for queries executed on endpoints you have Can Manage permission for.

### Query Details

Overview

Execution Details

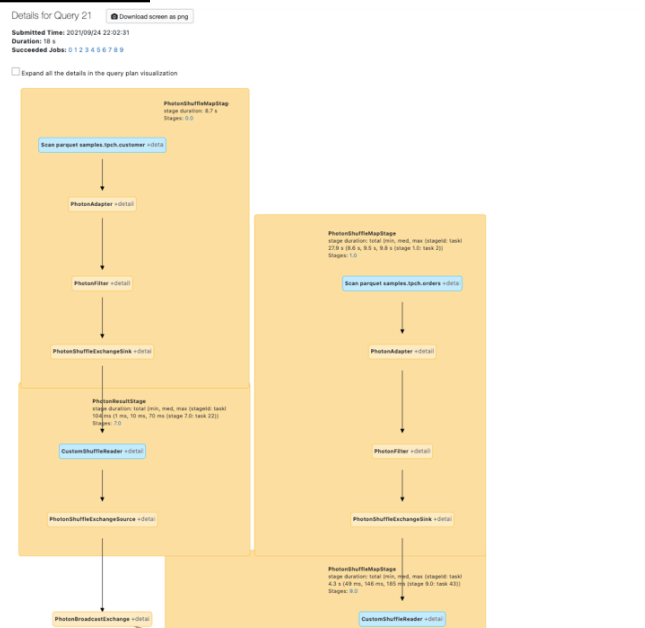
```
1 SELECT
2   year(o_orderdate) AS year,
3   n_name AS nation,
4   sum(
5     l_extendedprice * (1 - l_discount) * (
6       ((length(n_name)) / 100) + (year(o_orderdate) -
7     )
8   ) AS revenue
9 FROM
10  'samples`.`tpch`.`customer`,
11  'samples`.`tpch`.`orders`,
12  'samples`.`tpch`.`lineitem`,
13  'samples`.`tpch`.`supplier`,
14  'samples`.`tpch`.`nation`,
15  'samples`.`tpch`.`region`
```

ID	a87003b0-7f33-4219-a989-23430c75490f
Status	Finished
Start time	2021-09-24 15:02:06.315
End time	2021-09-24 15:02:50.223
Duration	42.24 s
User	class-3645-000@azuredatabrickstraining.onmicrosoft.com
SQL Endpoint	Starter Endpoint
Details	<a href="#">Open</a>

In the new window you will the full query plan. (The image shows part of the plan).

Scroll through the query plan to see the entire plan.

The query plan gives administrators useful information and provides insights about how the query was executed. This is important especially with large workloads or whenever the execution takes too long to process and becomes costly. Based on the information from the query plan you can find out what is not efficient and decide to rewrite part of the query to achieve better performance.



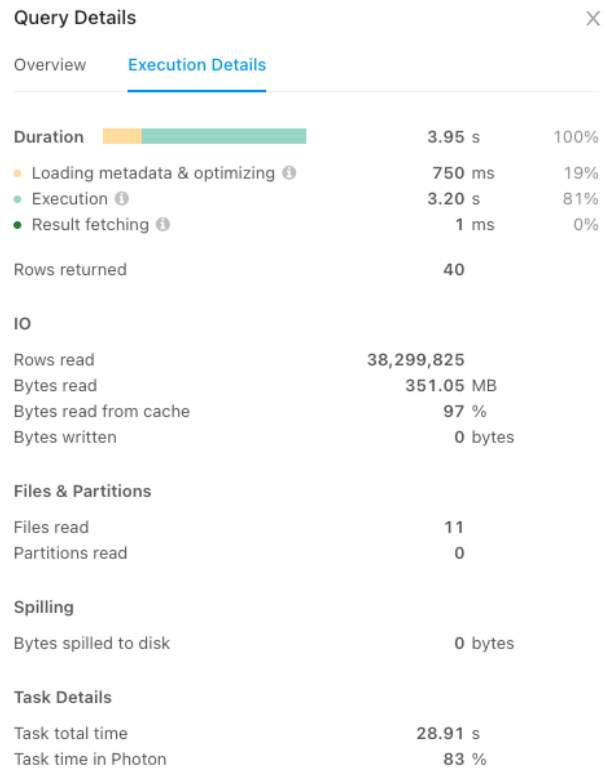
**Important: go back to the Databricks SQL tab in your browser. You should see the Query Details panel.**

Next, click the Execution Details tab.

Here, Databricks SQL displays the overall duration of the query execution. Below is the breakdown of time spent in:

- **loading metadata and optimization** (time spent fetching metadata, determining user access, and query optimization),
- **execution** (time spent on querying the data objects),
- **result fetching** for a query

You will get to see more details such as rows read/bytes, number of files/partitions, any spill to disk and task details.



Next, let's monitor the **SQL Endpoint**. Remember, you can monitor the number of queries handled by the endpoint and the number of clusters allocated to the SQL endpoint.

**Step 1:** To navigate to the SQL Endpoint, click on the Overview on the Query Details tab.

**Step 2:** Next to SQL Endpoint, click on "Starter Endpoint" link



*Note: This is a shortcut to SQL Endpoint Monitoring. You can also navigate to the monitoring by clicking SQL Endpoints in the side nav-bar, and click on Monitoring Tab.*



Click the **Monitoring** tab.

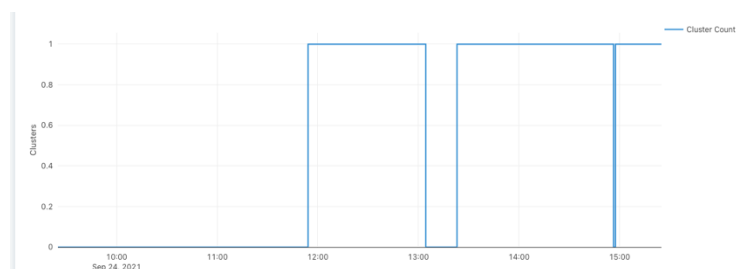
In the top chart, you will see a bar chart showing the number of queries handled by the SQL endpoint and the number of clusters allocated to the endpoint over the last 6 hours. The timescale allows you to view the information across 24 hours, 7 or 14 days.

If you hover over the chart, you can see at peak how many queries running or queued



The below chart shows how many clusters are allocated to this SQL endpoint.

If you have multi-cluster load balancing enabled and configured, the cluster count will show greater than one.



## Lab 7: Configure alerts

**Duration:** 5 minutes

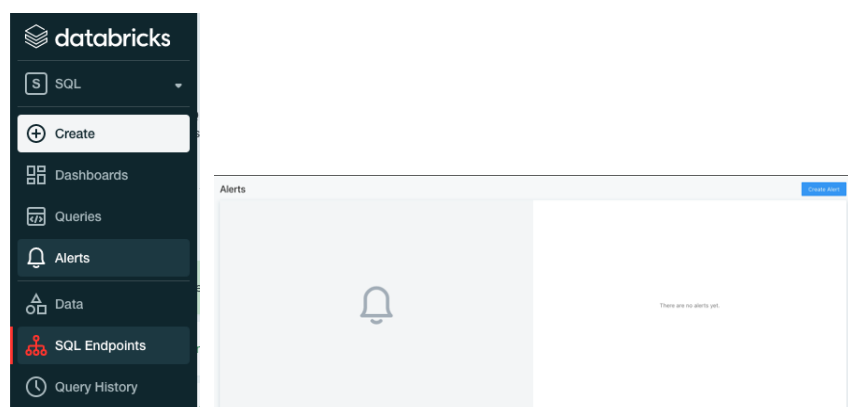
**Objective:** In this lab, you will learn how to configure alerts and understand various user settings.

With **alerts**, users define a set of conditions on fields returned by a scheduled query meets a threshold. Data teams can use alerts to monitor queries used for reporting or analytics or send alerts via email, Slack messages or custom webhooks.

Let's get started with creating an alert.

From the navigation bar, select **Alerts** (the bell icon). This will take you to the Alerts page.

Since this is a new workspace, there aren't any alerts set up yet. Go ahead and create one by clicking the **"Create Alert"** blue button in the top right corner.



When the new alert dialog box appears, hover over to the query box to select a query to monitor.

In this case, you can select the **“National Revenue Trends”** query.

Since we’re looking at national revenue, let’s set an alert when revenue drops to lower than \$10M, send an notification.

## New Alert

Start by selecting the query that you would like to monitor using the search bar. Keep in mind that Alerts do not work with queries that use parameters.

Query

- Copy of: National Revenue Trends [Sample](#)
- National Revenue Trends [Sample](#)
- National Revenue Map [Sample](#)
- Order Revenue [Sample](#)
- Customer Value [Sample](#)
- National Revenue Trends [Sample](#)
- Overall Customer Count [Sample](#)
- TPCH - Number Suppliers [Sample](#)

On the next screen, you can configure triggers to setup the alert.

A few things to mention, when you create an alert, it is advised to set up a schedule. An Alert without a query schedule will only send notifications if a user manually executes this query.

National Revenue Trends: year > 1

Start by selecting the query that you would like to monitor using the search bar. Keep in mind that Alerts do not work with queries that use parameters. [Setup Instructions](#)

Query

Value column  Condition  Threshold

Trigger when

When triggered, send notification

Template

[Create Alert](#)

When you set a trigger you specify the output column, condition and threshold the alert is based on.

In the Trigger **when field**, configure the alert.

- The **Value** column drop-down controls which field of your query result is evaluated.
- The **Condition** drop-down controls the logical operation to be applied.
- The **Threshold** text input is compared against the Value column using the Condition you specify.

Step 1: For this query, in the value column, select “revenue”. Notice, Databrick SQL gives you a list of all columns from the query.

Query

Value column  Condition  Threshold

Trigger when

When triggered, send notification

Step 2: Set the condition, in this case choose Less than.

Start by selecting the query that you would like to monitor using the search bar. Keep in mind that Alerts do not work with queries that use parameters.

Query

Value column  Condition  Threshold

Trigger when

When triggered, send notification

Template

[Create Alert](#)

Step 3: Set the threshold to 10,000,000

Value column  Condition  Threshold

Trigger when

In the **When triggered, send notification** field, select how many notifications are sent when your alert is triggered.

- **Just once:** Send a notification when the alert status changes from OK to TRIGGERED.
- **Each time alert is evaluated:** Send a notification whenever the alert status is TRIGGERED regardless of its status at the previous evaluation.
- **At most every:** Send a notification whenever the alert status is TRIGGERED at a specific interval. This choice lets you avoid notification spam for alerts that trigger often.

Step 1: In the **When triggered, send notification** field, choose **Just once**

The screenshot shows a dropdown menu for the 'When triggered, send notification' field. The menu is open, showing three options: 'Just once until back to normal', 'Each time alert is evaluated until back to normal', and 'At most every ... when alert is evaluated'. The 'Just once' option is selected, and a 'Create Alert' button is visible at the bottom of the menu.

Lastly, you can choose a template or customize a template for the alert.

There are two types of templates:

**Default template:** Alert notification is a message with links to the Alert configuration screen and the Query screen.

**Custom template:** Alert notification includes more specific information about the alert. With customized templates, a box display that consists of input fields for the subject and body, any static content is valid, or include built-in template variables such as `alert_status`, `alert_condition`, `query_name` etc.

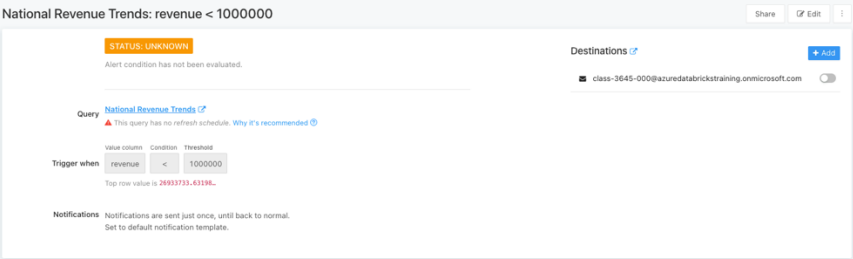
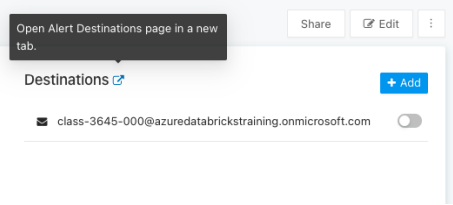
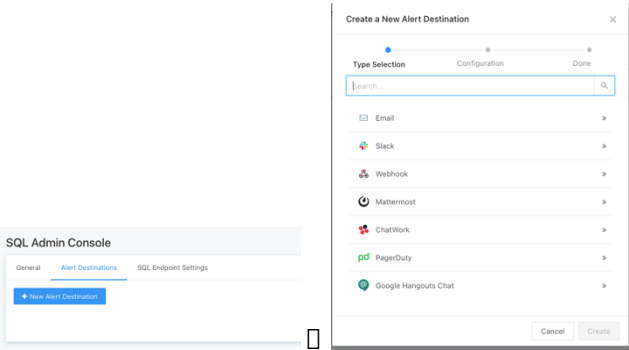
Now you've created your first alert.

Step 1: choose **Default** template and the final step to complete this task is click **"Create Alert"**

The screenshot shows a dropdown menu for the 'Template' field. The menu is open, showing two options: 'Default template' and 'Custom template'. The 'Default template' option is selected.

\*FYI – here's what a customized template would look like:

The screenshot shows the 'Use custom template' form. It has a 'Subject / Body' input field and a 'Preview' toggle switch. Below the input field is a 'Formatting guide' link. A 'Create Alert' button is visible at the bottom.

	
<p>Lastly, you can configure destinations. Databricks SQL sends notifications to your chosen alert destinations whenever it detects that the Alert status has changed from OK to TRIGGERED or vice versa.</p> <p>Destinations can be send emails, Slack messages, or custom webhooks.</p>	<p>( for this lab, we will not configure destinations, you can navigate thru the various options)</p> <p>Click on Destinations (which will open in a new window to create a new alert destination, you will notice this is in the SQL Admin Console). This is where you can define a new definition (email, slack, webhooks, PagerDuty)</p>  <p>In the new tab, you can click on New Alert Destinations.</p>  <p>Hit Cancel, Close the window and return to the previous Databricks SQL screen – the New Alert Screen.</p>

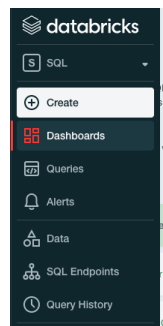
## Lab 8: Creating Dashboards

**Duration:** 10 minutes

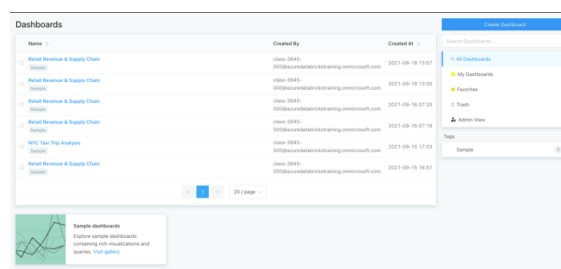
**Objective:** In this lab, you will explore Databricks SQL dashboards, which lets you combine visualizations and text boxes that provide context with your data.

To create dashboards, from the navigation, click “Dashboards”.

This will take you to the dashboard overview screen.

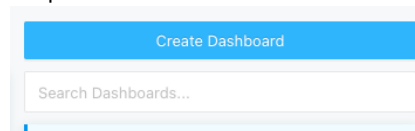


Here you can see or access existing dashboard or create a new dashboard.



Click “Create Dashboard”

Step 1: Create Dashboard

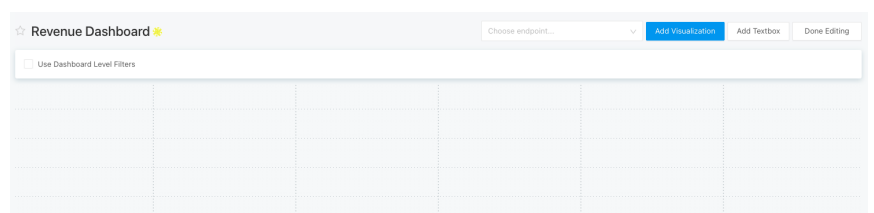


Step 2: Name the New Dashboard – call it **“Revenue Dashboard”** and click **“Save”**



The blank dashboard editor appears.

First, let's add a heading for this dashboard. Click the “Add Textbox” widget.



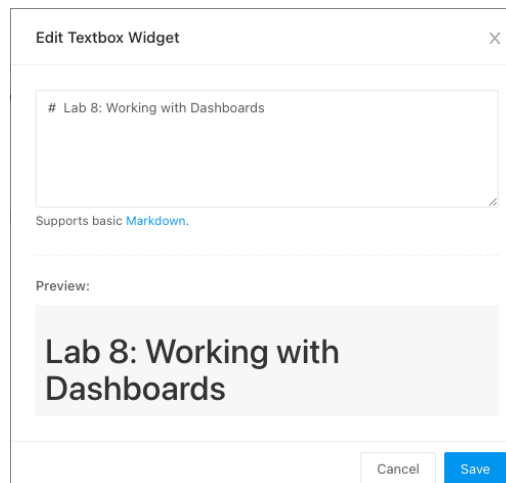
Add some text, call it “# Lab 8: Working with Dashboards”.

You can also use some basic Markdown for headings, text formatting, images, or links.

Below, you will see a preview of the text.

Click “Save”

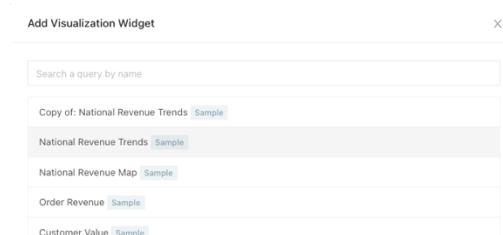
You can resize the textbox, move it around the canvas.



Next add a Visualization – click the “Add Visualization” widget.

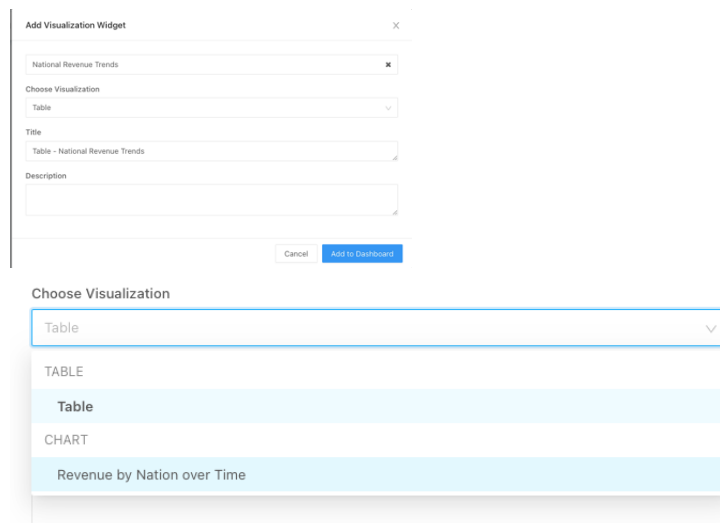
Then select the “**National Revenue Trends**”

Step 1: Select “**National Revenue Trends**”



You can choose the type of visualization you would like to include in the dashboard: add the table or a chart.

Step 2: Under “**Choose Visualization**”, change to chart – “**Revenue by Nation over Time**”



Step 3: You may want to update the title and add a description

Step 4: click: “**Add Visualization**”

Add Visualization Widget

National Revenue Trends

Choose Visualization

Revenue by Nation over Time

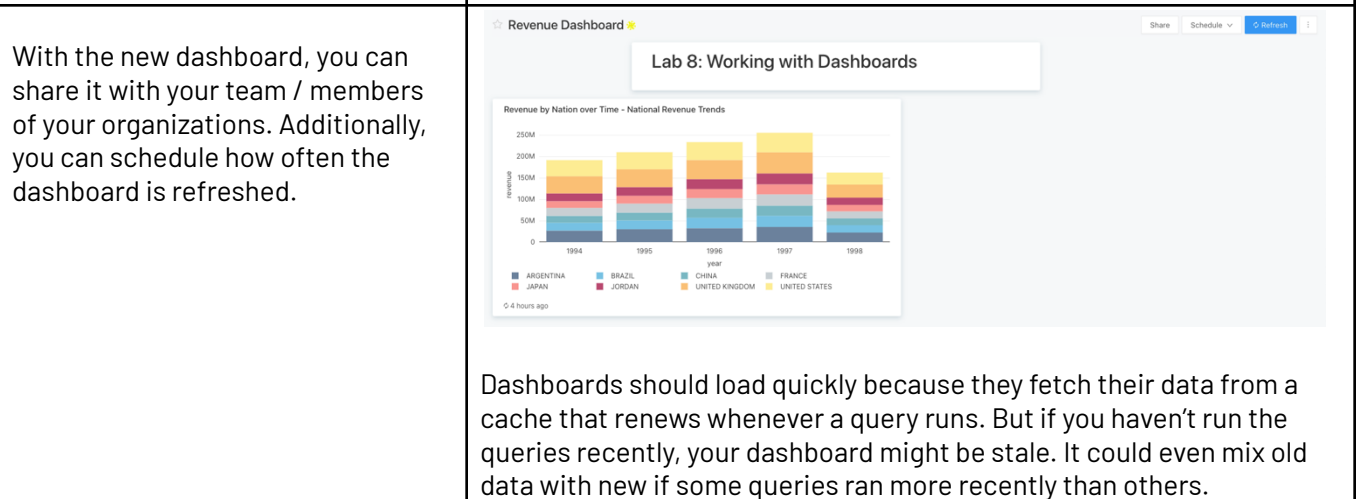
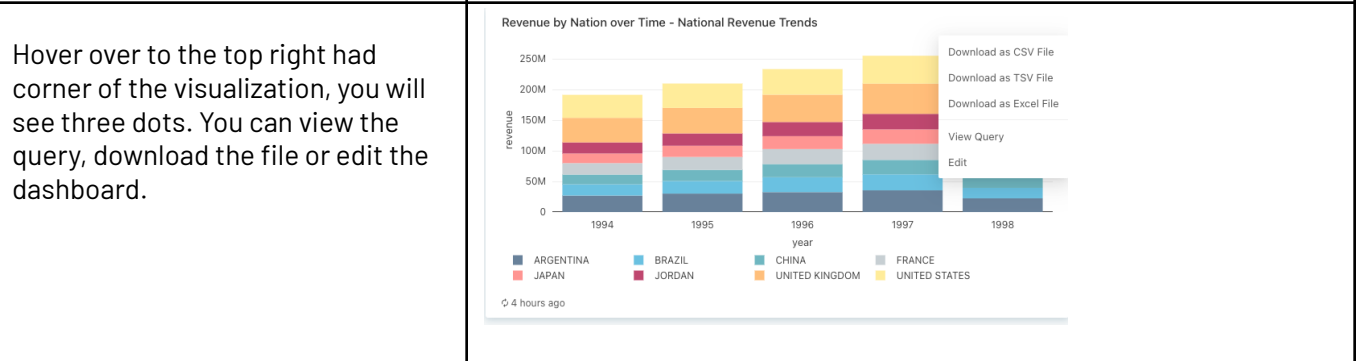
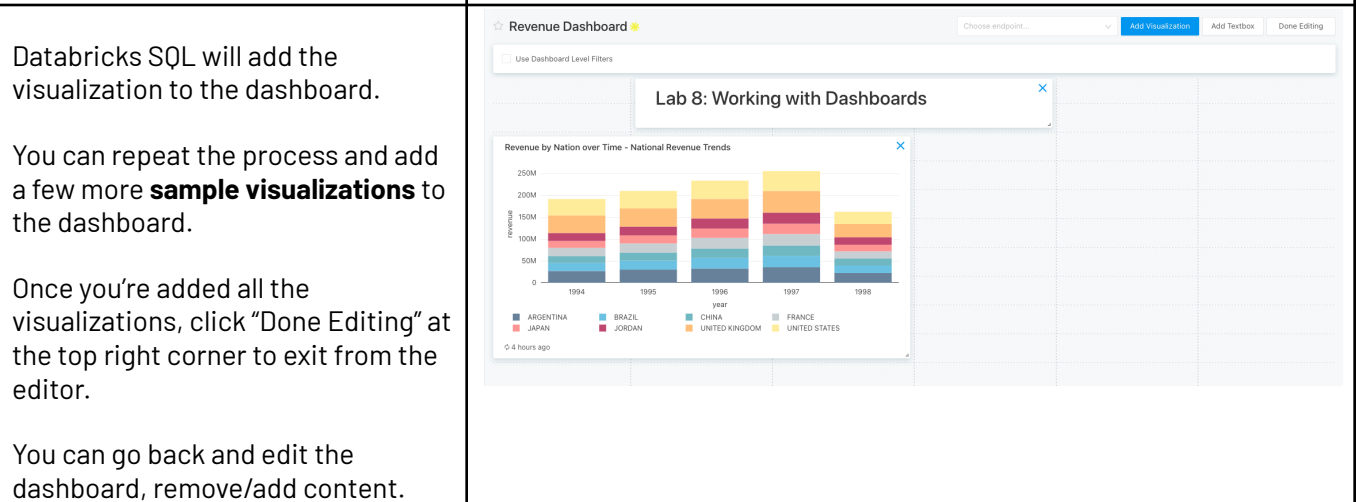
Title

Revenue by Nation over Time - National Revenue Trends

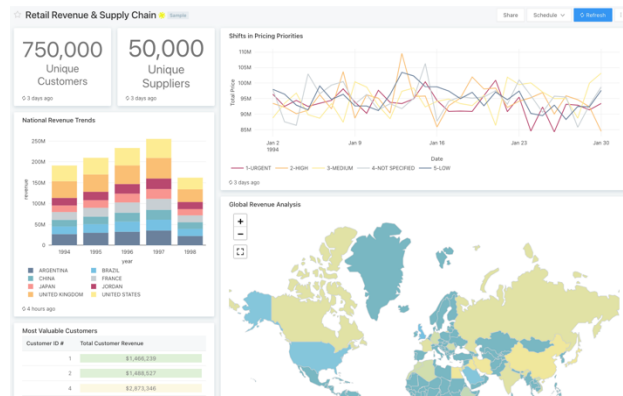
Description

Cancel

Add to Dashboard



You can go back to the dashboards to view and explore another sample dashboard. The image here shows the dashboard: Retail Revenue & Supply Chain.



**Congratulations! This concludes the introductory to Databricks SQL workshop!**