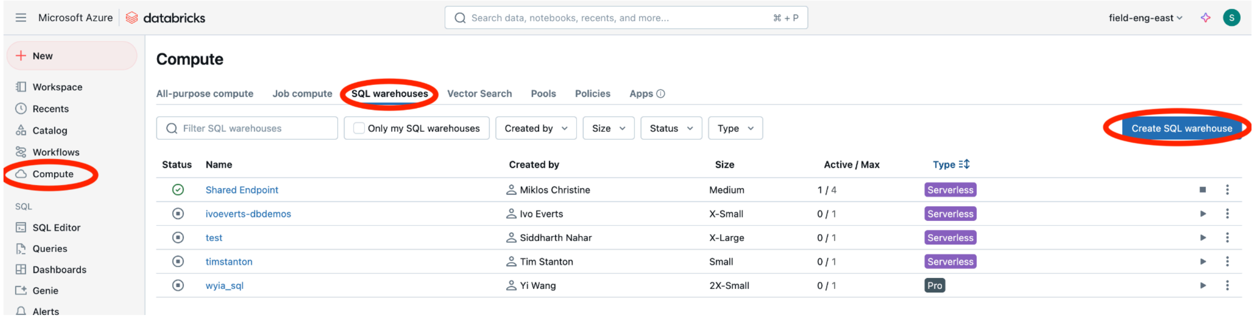
Prep Work for Workshop

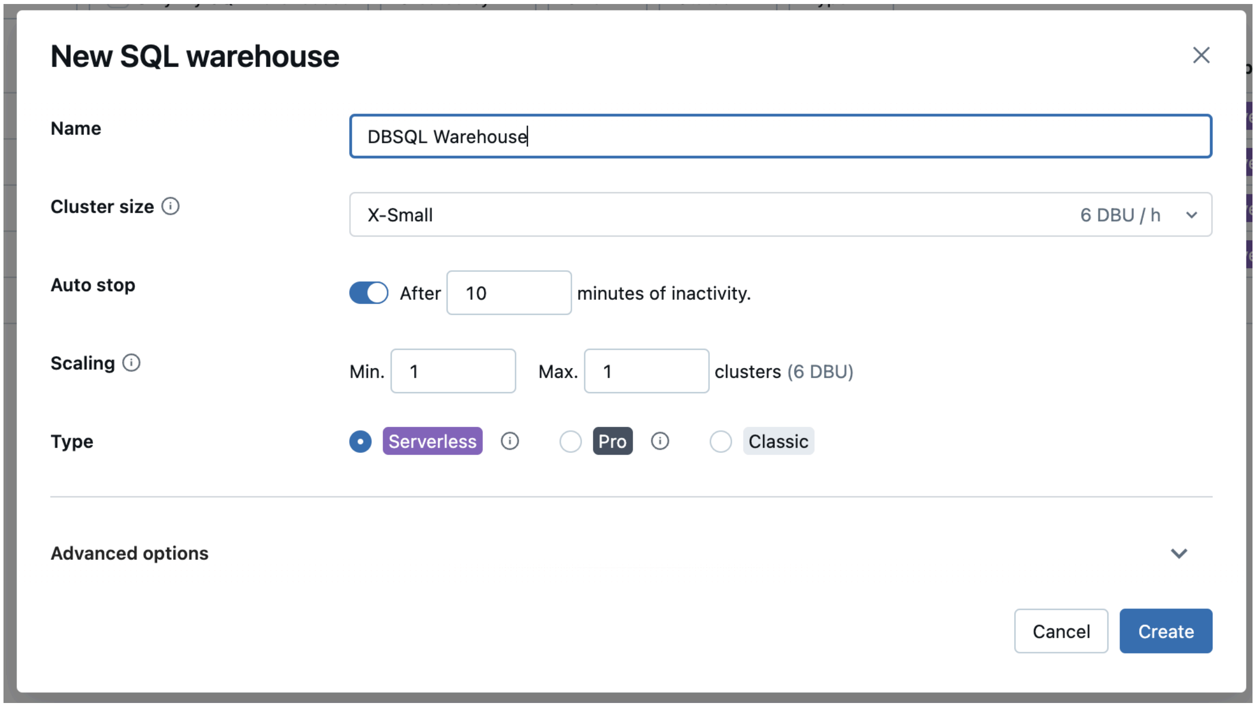
1. Let's create the SQL Warehouse we'll use. Click on Compute in the left menu. Choose the SQL Warehouses tab.



Compute

Status	Name	Created by	Size	Active / Max	Type
✓	Shared Endpoint	Miklos Christine	Medium	1 / 4	Serverless
○	ivoeverts-dbdemos	Ivo Everts	X-Small	0 / 1	Serverless
○	test	Siddharth Nahar	X-Large	0 / 1	Serverless
○	timstanton	Tim Stanton	Small	0 / 1	Serverless
○	wyia_sql	Yi Wang	2X-Small	0 / 1	Pro

2. Give it a name and choose Serverless if you have the option to.



New SQL warehouse

Name: DBSQL Warehouse

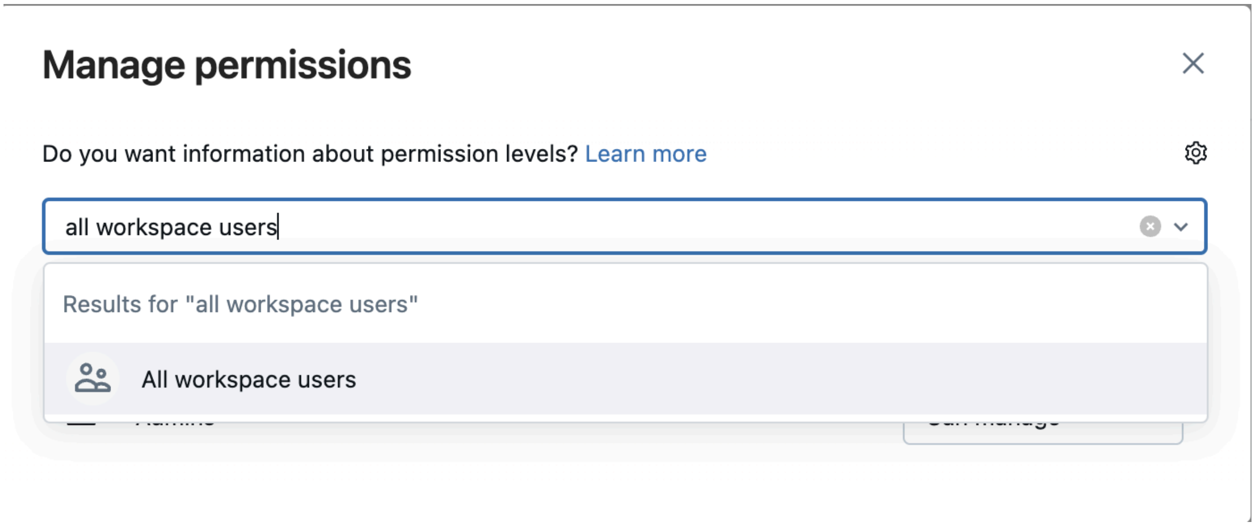
Cluster size: X-Small 6 DBU / h

Auto stop: After 10 minutes of inactivity.

Scaling: Min. 1 Max. 1 clusters (6 DBU)

Type: ● Serverless ○ Pro ○ Classic

3. Provide access. You may wish to choose All workspace users or just specific users.



Manage permissions

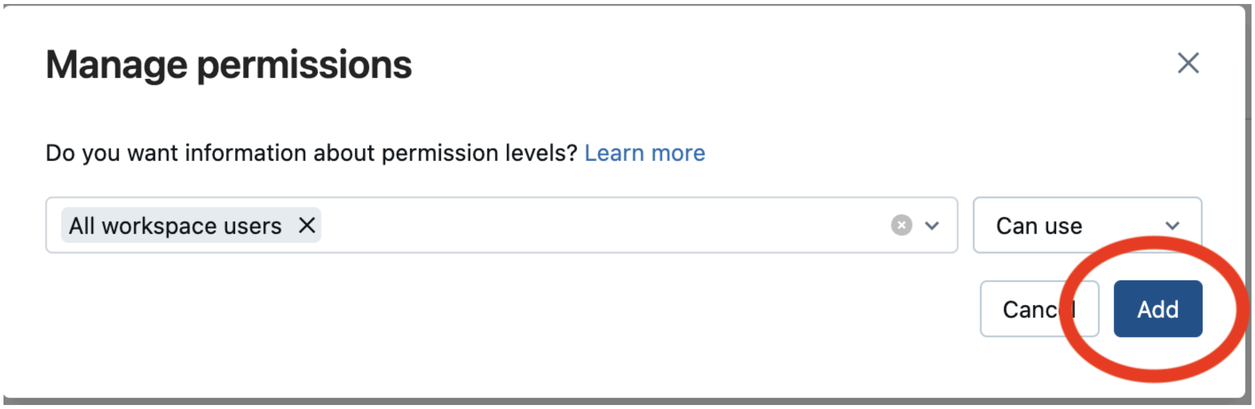
Do you want information about permission levels? Learn more

all workspace users

Results for "all workspace users"

All workspace users

Make sure to Click Add



Manage permissions

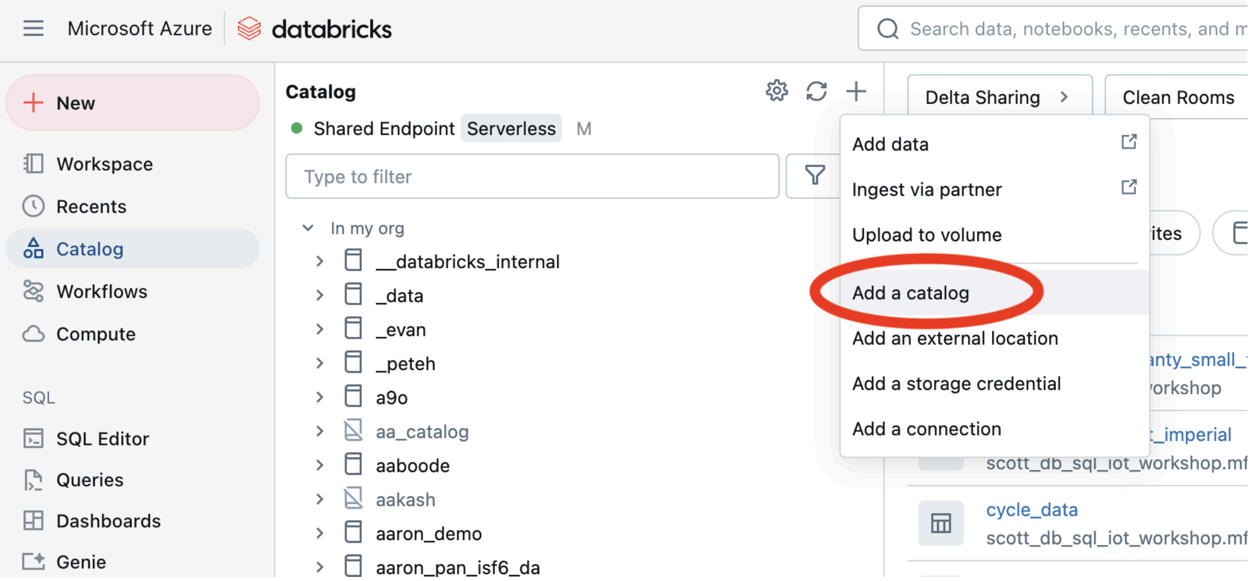
Do you want information about permission levels? Learn more

All workspace users

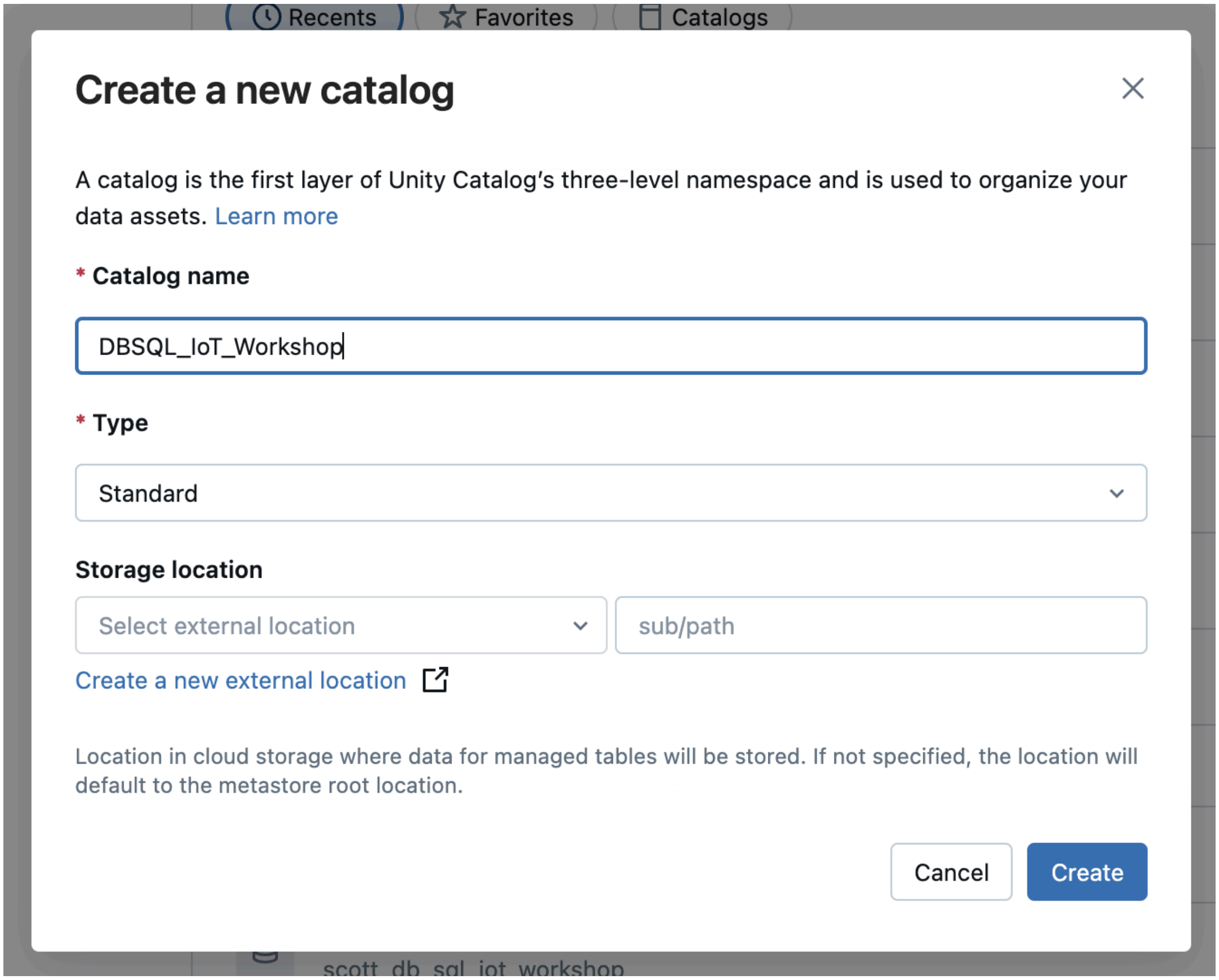
Can use

We will attach to this SQL Warehouse later after we import the dashboard.

4. Now, let’s create the Catalog. Click on the Catalog menu on the left, click the +, and select ‘Add a catalog’



5. Name and Create the Catalog



Create a new catalog

A catalog is the first layer of Unity Catalog's three-level namespace and is used to organize your data assets. Learn more

* Catalog name

DBSQL_IoT_Workshop

* Type

Standard

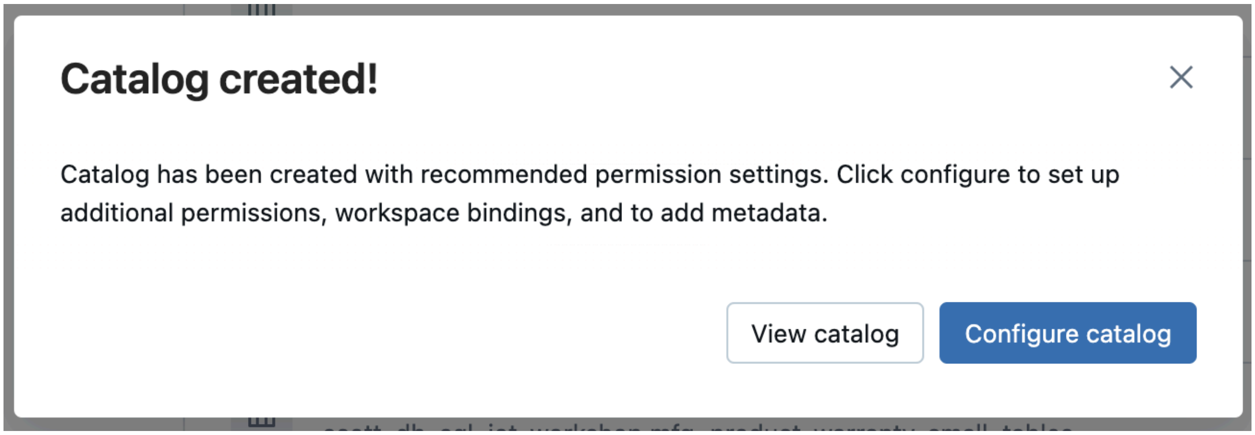
Storage location

Select external location | sub/path

Create a new external location

Location in cloud storage where data for managed tables will be stored. If not specified, the location will default to the metastore root location.

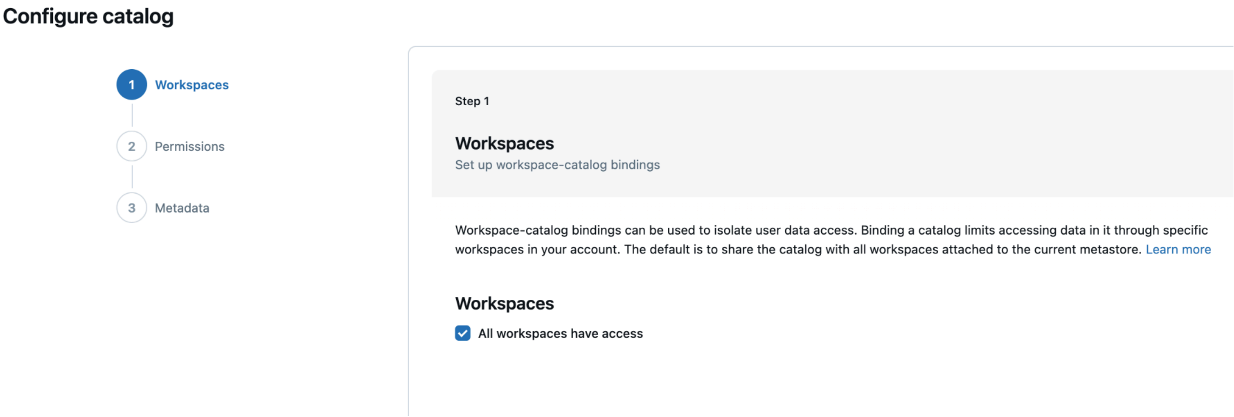
6. Your catalog is now created. Press the Configure catalog button.



Catalog created!

Catalog has been created with recommended permission settings. Click configure to set up additional permissions, workspace bindings, and to add metadata.

7. Walk through the steps granting access to Workspaces, Users, and just press Save on the Metadata screen for now.



Configure catalog

Step 1

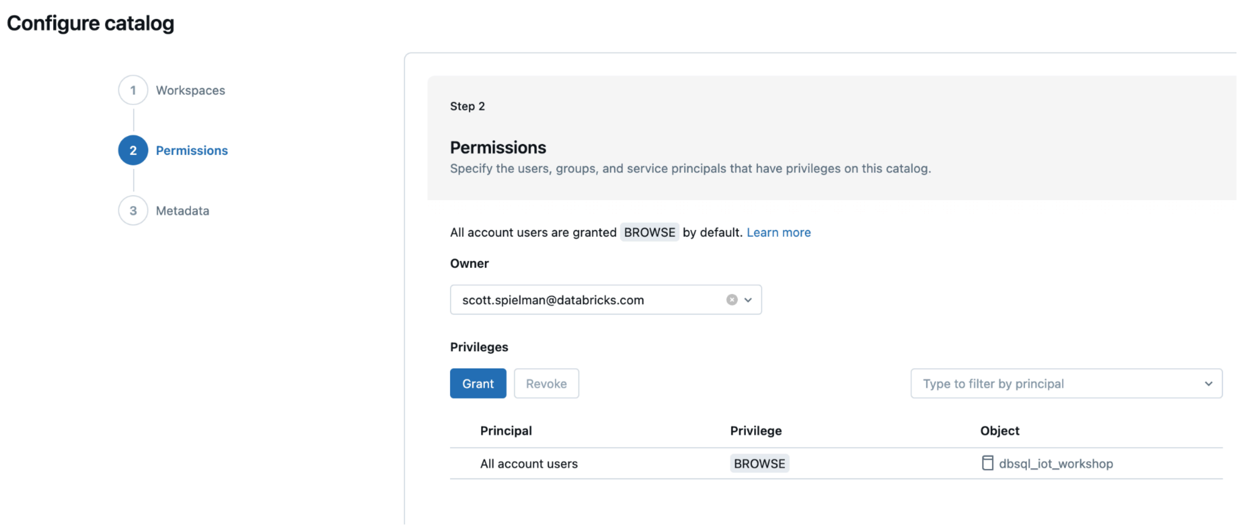
Workspaces

Set up workspace-catalog bindings

Workspace-catalog bindings can be used to isolate user data access. Binding a catalog limits accessing data in it through specific workspaces in your account. The default is to share the catalog with all workspaces attached to the current metastore. Learn more

Workspaces

☑ All workspaces have access



Configure catalog

1 Workspaces

2 Permissions

3 Metadata

Step 2

Permissions

Specify the users, groups, and service principals that have privileges on this catalog.

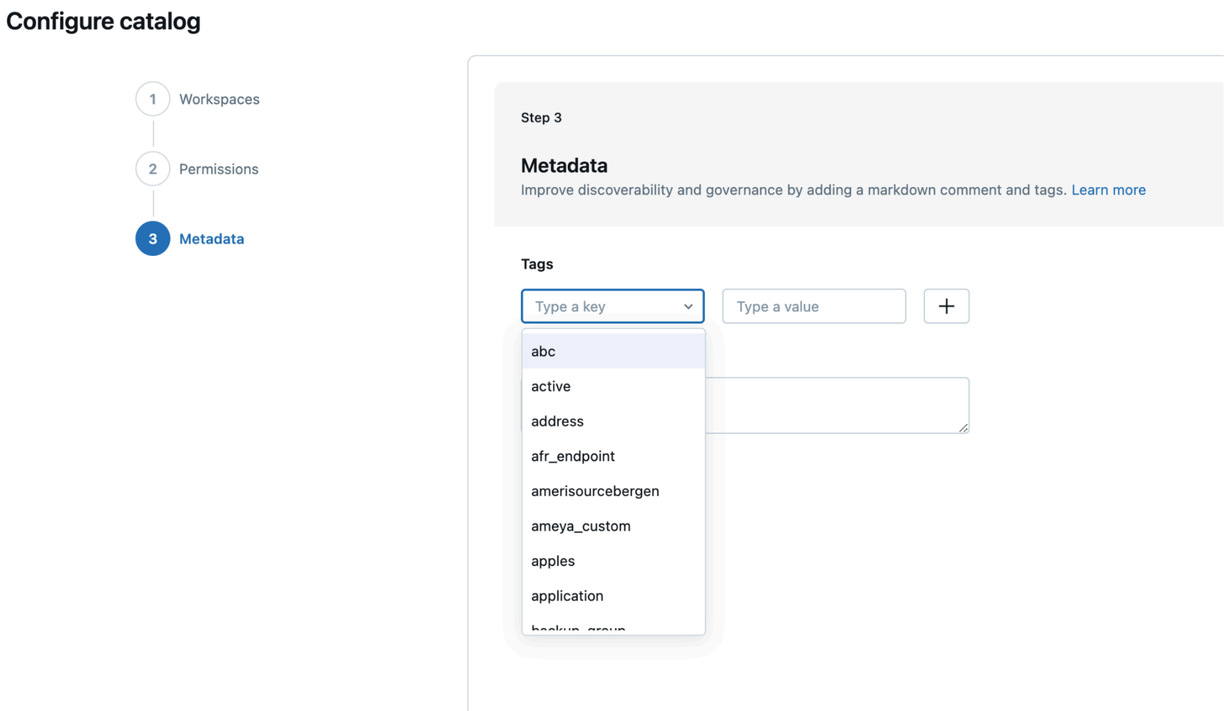
All account users are granted BROWSE by default. Learn more

Owner

scott.spielman@databricks.com

Privileges

Principal	Privilege	Object
All account users	BROWSE	dbsql_iot_workshop



Configure catalog

1 Workspaces

2 Permissions

3 Metadata

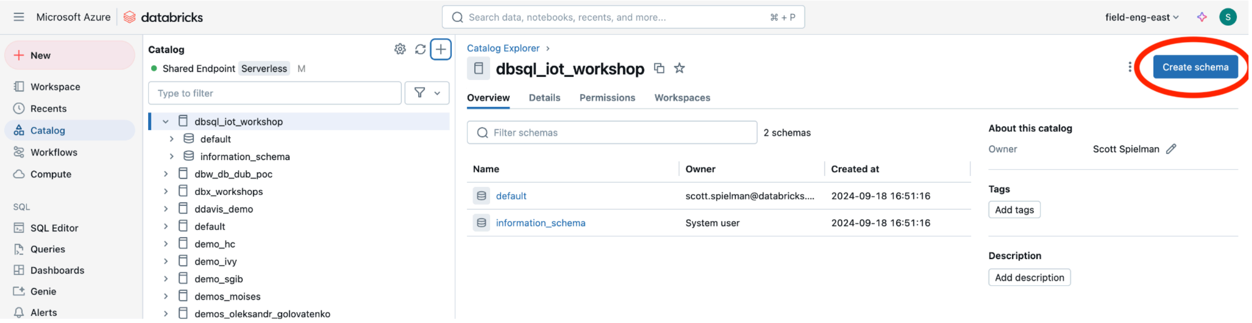
Step 3

Metadata

Improve discoverability and governance by adding a markdown comment and tags. Learn more

Tags

8. After saving you will see your Catalog listed on the left. Click on Create schema



dbsql_iot_workshop

2 schemas

Name	Owner	Created at
default	scott.spielman@databricks....	2024-09-18 16:51:16
information_schema	System user	2024-09-18 16:51:16

About this catalog

Owner Scott Spielman

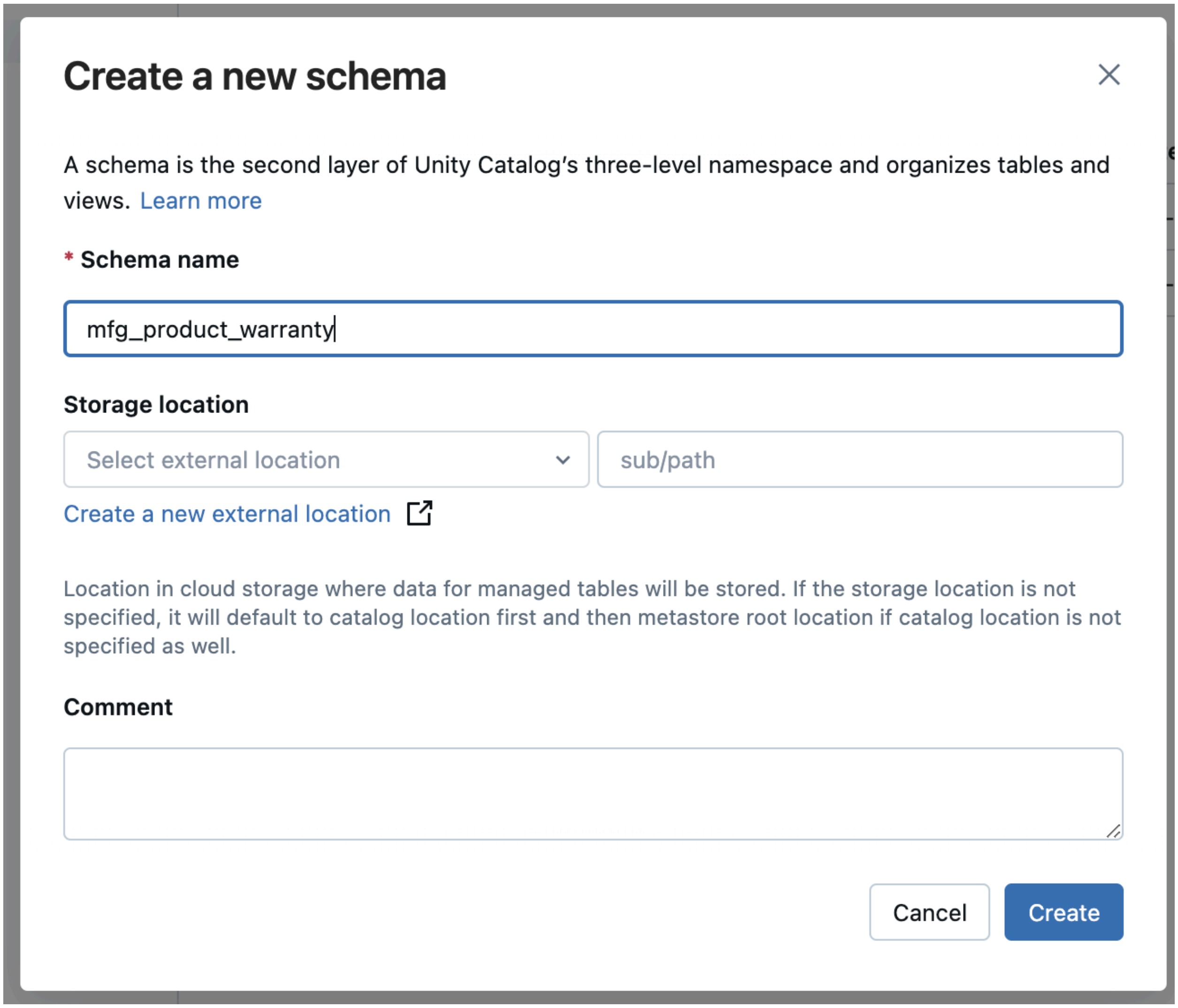
Tags

Add tags

Description

Add description

9. Specify the schema name: mfg_product_warranty and Click Create



Create a new schema

A schema is the second layer of Unity Catalog's three-level namespace and organizes tables and views. Learn more

* Schema name

mfg_product_warranty

Storage location

Select external location | sub/path

Create a new external location

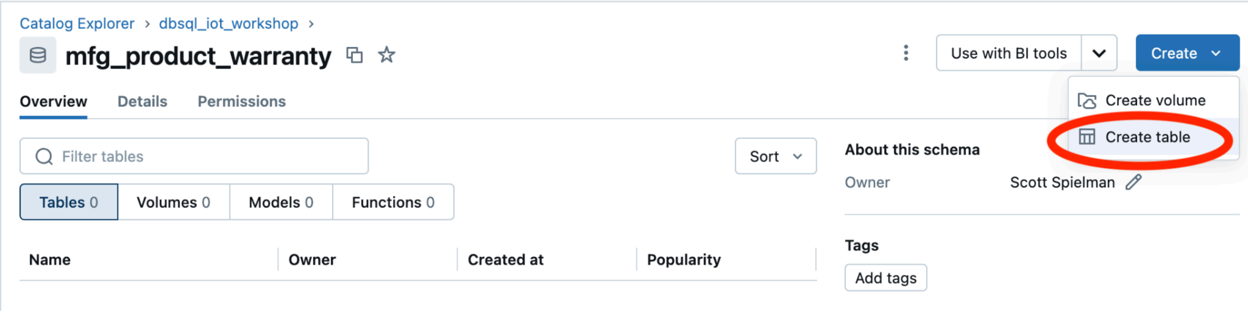
Location in cloud storage where data for managed tables will be stored. If the storage location is not specified, it will default to catalog location first and then metastore root location if catalog location is not specified as well.

Comment

Your schema is now created

10. Download and uncopress the workshop files archive “Databricks DBSQL IoT Workshop.zip”

11. Navigate to the schema and select create table



Catalog Explorer › dbsql_iot_workshop ›

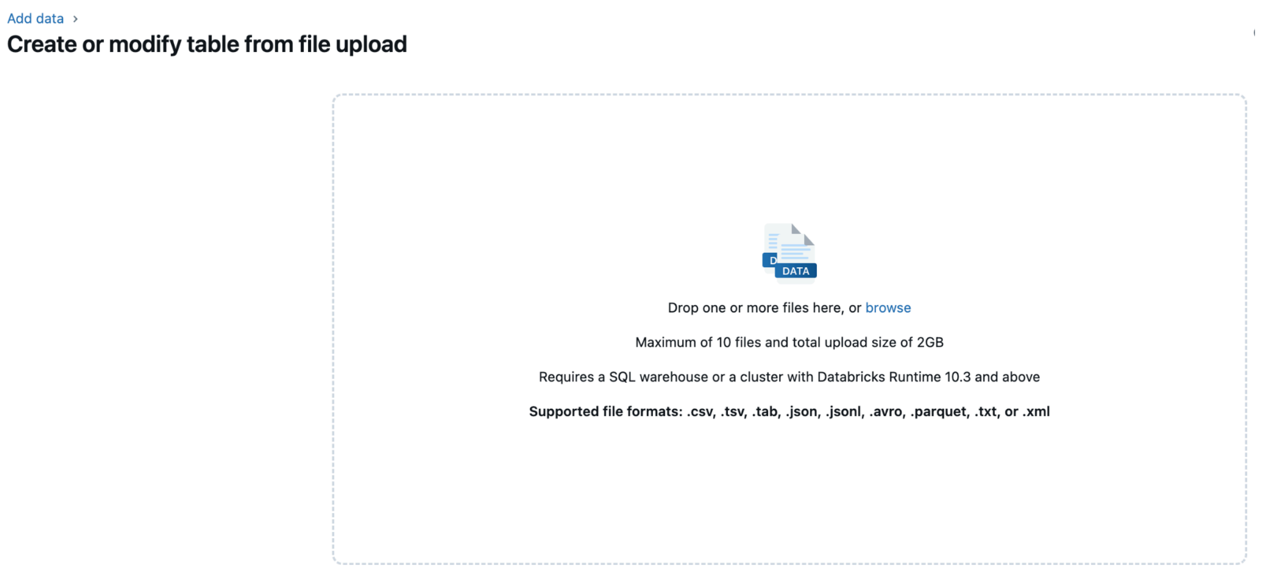
mfg_product_warranty

Name	Owner	Created at	Popularity

About this schema

Owner: Scott Spielman

Tags



Add data ›

Create or modify table from file upload

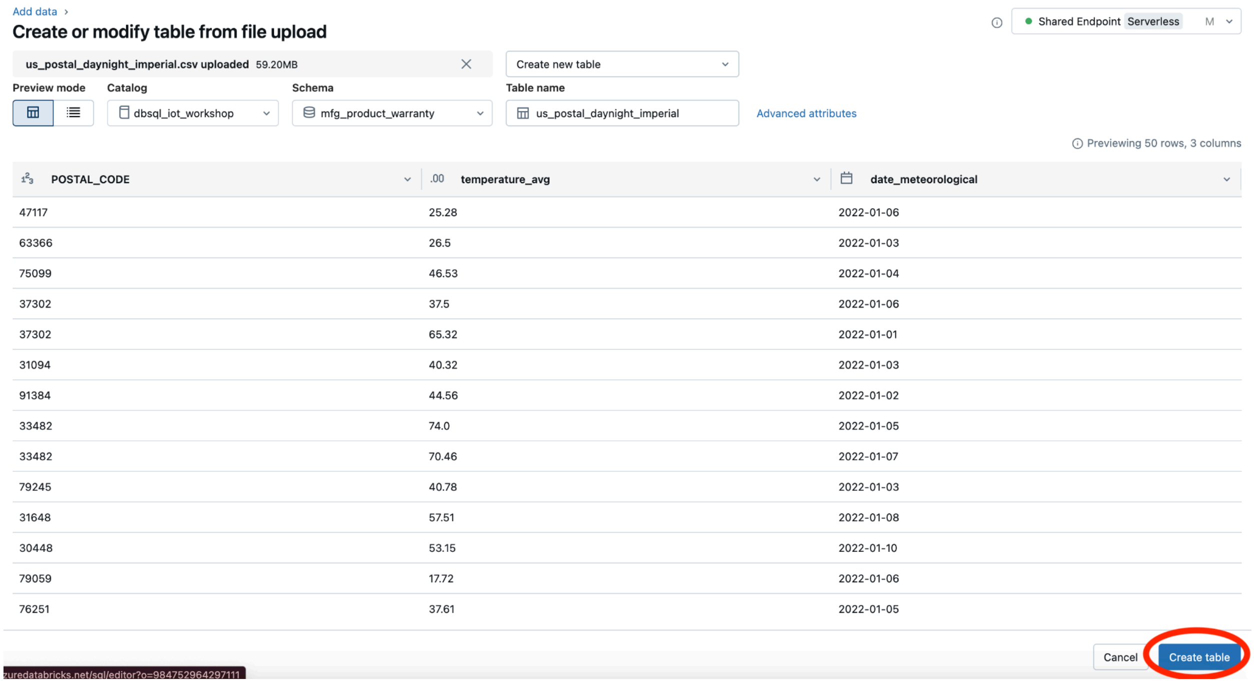
Drop one or more files here, or browse

Maximum of 10 files and total upload size of 2GB

Requires a SQL warehouse or a cluster with Databricks Runtime 10.3 and above

Supported file formats: .csv, .tsv, .tab, .json, .jsonl, .avro, .parquet, .txt, or .xml

12. Choose the file you downloaded from google drive



Add data ›

Create or modify table from file upload

us_postal_daynight_imperial.csv uploaded 59.20MB

Create new table

Preview mode

Catalog: dbsql_iot_workshop

Schema: mfg_product_warranty

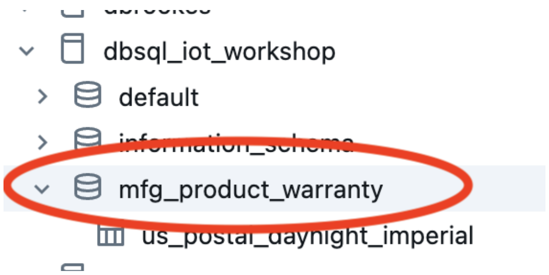
Table name: us_postal_daynight_imperial

Previewing 50 rows, 3 columns

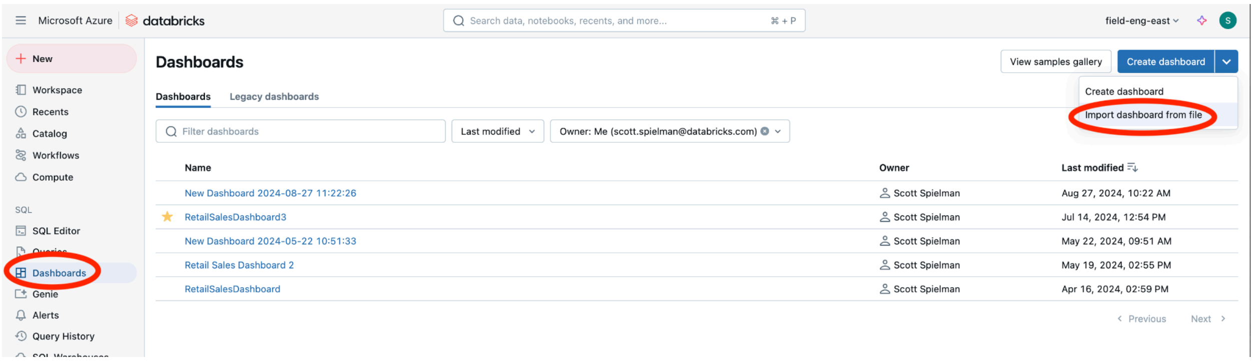
POSTAL_CODE	temperature_avg	date_meteorological
47117	25.28	2022-01-06
63366	26.5	2022-01-03
75099	46.53	2022-01-04
37302	37.5	2022-01-06
37302	65.32	2022-01-01
31094	40.32	2022-01-03
91384	44.56	2022-01-02
33482	74.0	2022-01-05
33482	70.46	2022-01-07
79245	40.78	2022-01-03
31648	57.51	2022-01-08
30448	53.15	2022-01-10
79059	17.72	2022-01-06
76251	37.61	2022-01-05

Click on create table as shown below

13. Table will be created. Click back on Schema to Add the next table.



14. Add all the other tables
15. Navigate to Dashboards and Import Dashboard from file

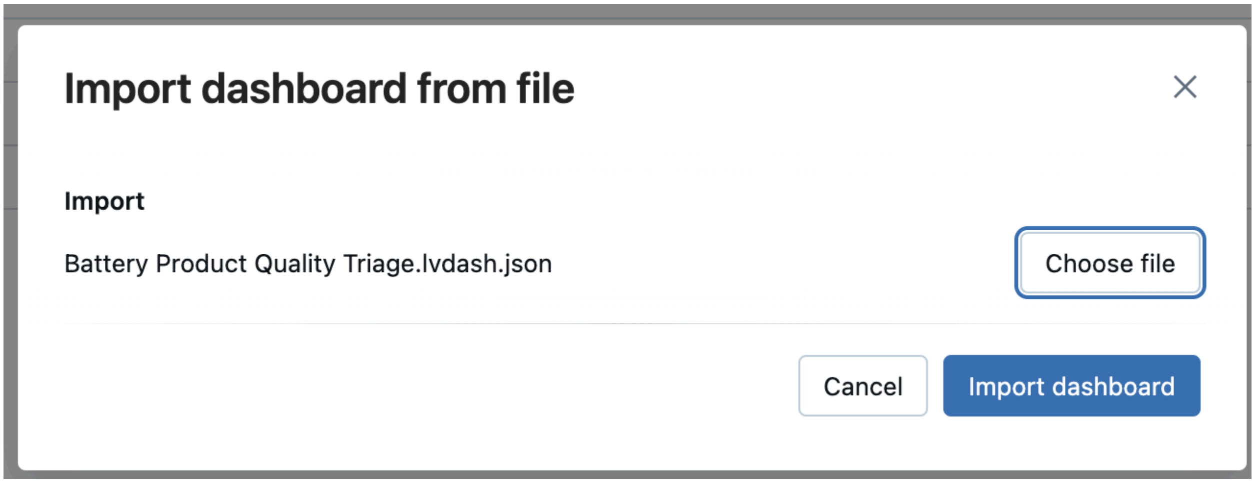


Dashboards

Owner: Me (scott.spielman@databricks.com)

	Name	Owner	Last modified
	New Dashboard 2024-08-27 11:22:26	Scott Spielman	Aug 27, 2024, 10:22 AM
★	RetailSalesDashboard3	Scott Spielman	Jul 14, 2024, 12:54 PM
	New Dashboard 2024-05-22 10:51:33	Scott Spielman	May 22, 2024, 09:51 AM
	Retail Sales Dashboard 2	Scott Spielman	May 19, 2024, 02:55 PM
	RetailSalesDashboard	Scott Spielman	Apr 16, 2024, 02:59 PM

16. Import the Battery Product Quality Triage.lvdash.json file

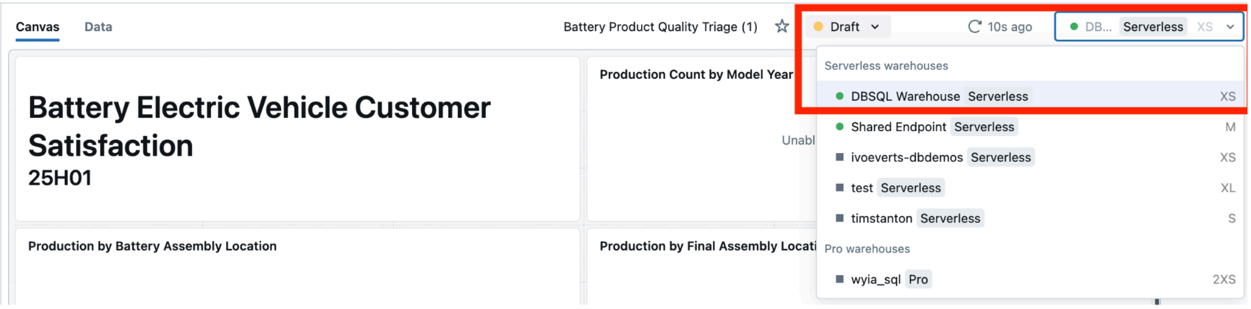


Import dashboard from file

Import

Battery Product Quality Triage.lvdash.json

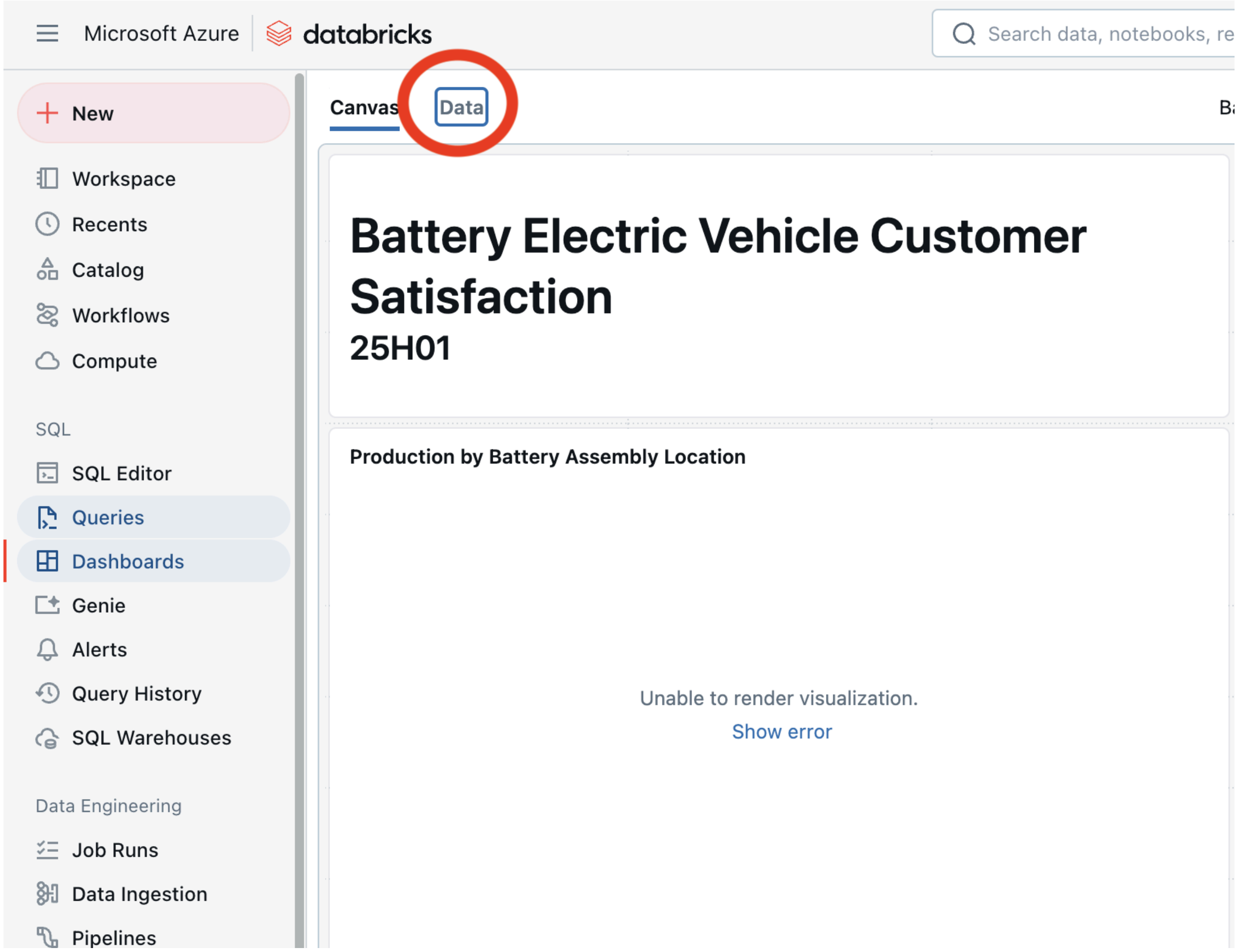
17. Let’s attach this dashboard to the SQL Warehouse we created.


Canvas
Data
Battery Product Quality Triage (1)
Draft
10s ago
DB... Serverless XS
Serverless warehouses
DBSQL Warehouse Serverless XS
Shared Endpoint Serverless M
ivoeverts-dbdemos Serverless XS
test Serverless XL
timstanton Serverless S
Pro warehouses
wyia_sql Pro 2XS
Production Count by Model Year
Unabl
Production by Battery Assembly Location
Production by Final Assembly Locati

Battery Electric Vehicle Customer Satisfaction

25H01

18. At first the dashboard won't fully render. We need to go to the Data tab



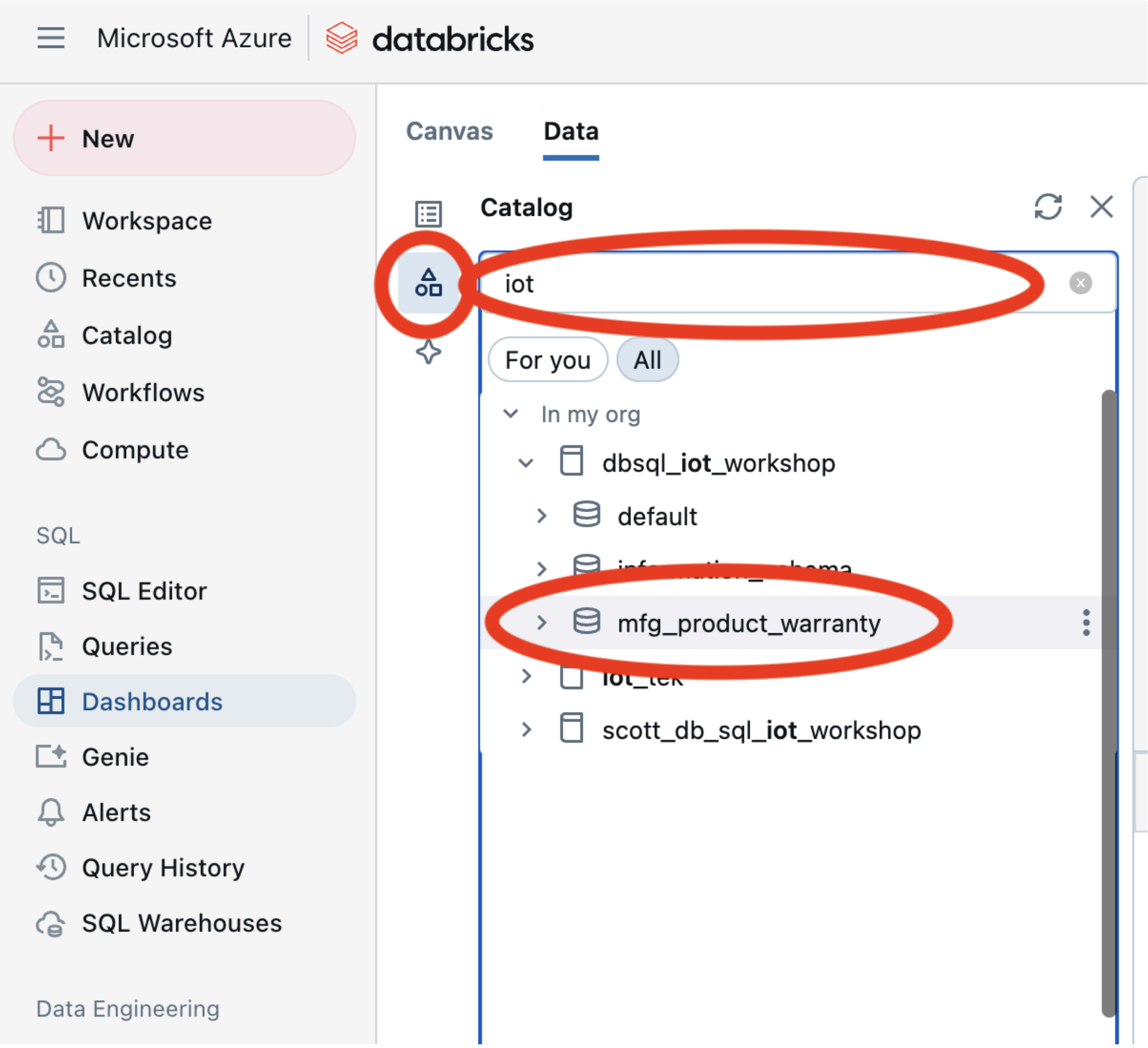
Battery Electric Vehicle Customer Satisfaction

25H01

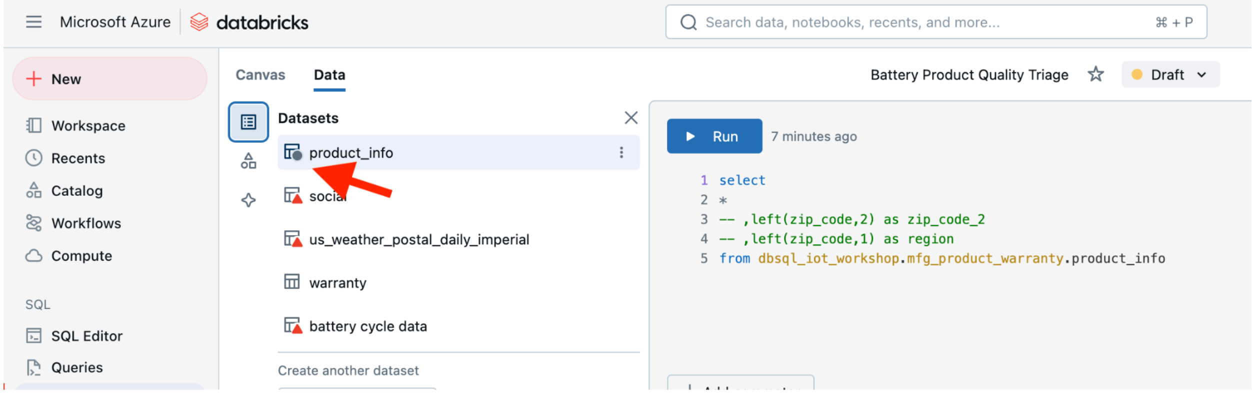
Unable to render visualization.

We need to modify the SQL code to use the catalog name you chose.

One way to do this is to open the Catalog pane, filter to the name you chose, and navigate to it.



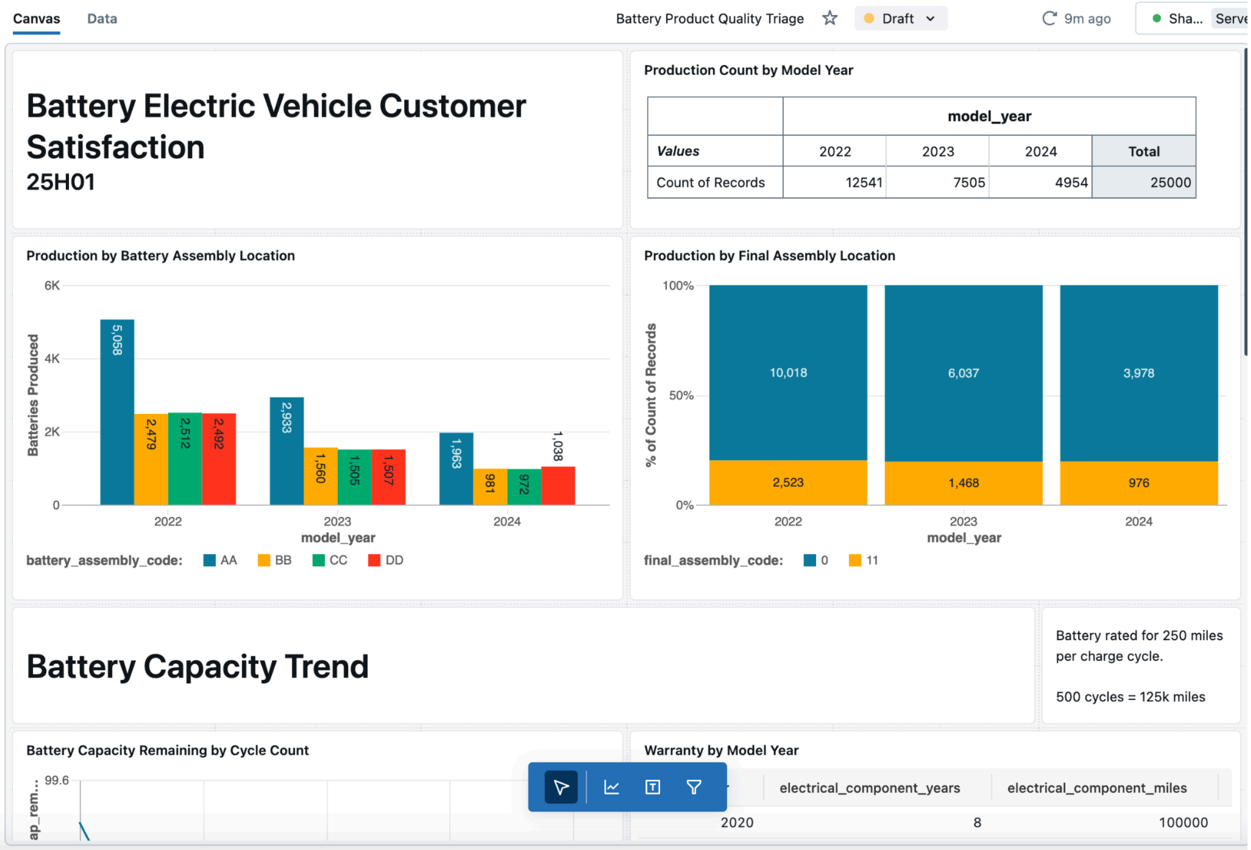
If you flip back to the Data tab you will see the table name has turned green.



```
select
*
-- ,left(zip_code,2) as zip_code_2
-- ,left(zip_code,1) as region
from dbsql_iot_workshop.mfg_product_warranty.product_info
```

Do this for the other tables. (You can leave warranty as is).

Flip back to the Canvas tab. Refresh the page, and your dashboard should populate and look like this:



Battery Electric Vehicle Customer Satisfaction

25H01

Production Count by Model Year

	model_year			
Values	2022	2023	2024	Total
Count of Records	12541	7505	4954	25000

Production by Battery Assembly Location

Production by Final Assembly Location

Battery Capacity Trend

Battery rated for 250 miles per charge cycle.

500 cycles = 125k miles

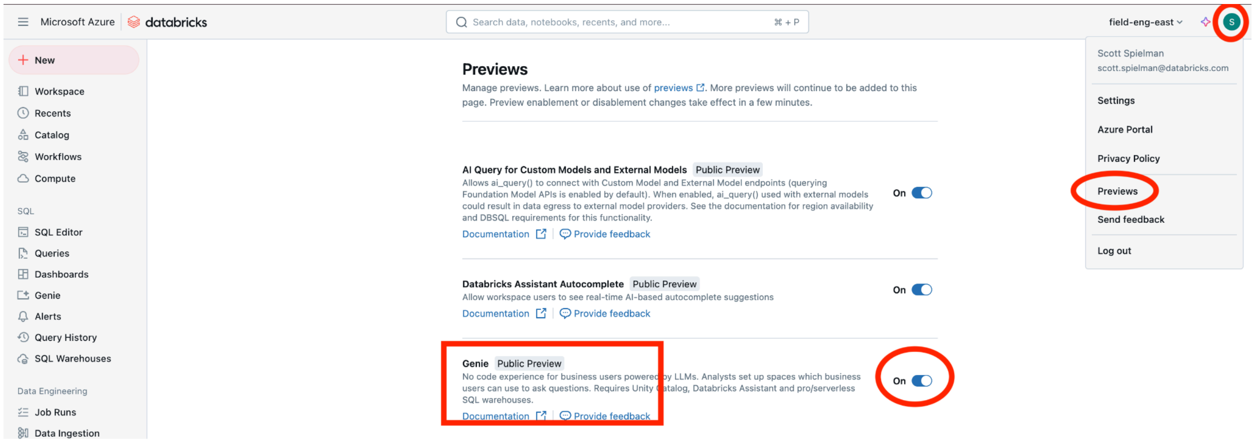
Battery Capacity Remaining by Cycle Count

Warranty by Model Year

	electrical_component_years	electrical_component_miles
2020	8	100000

Enable “Genie Spaces”

Click on your user icon in the top right corner of the workspace. Select Previews and Enable Genie.



Previews

Manage previews. Learn more about use of previews. More previews will continue to be added to this page. Preview enablement or disablement changes take effect in a few minutes.

AI Query for Custom Models and External Models Public Preview

Allows ai_query() to connect with Custom Model and External Model endpoints (querying Foundation Model APIs is enabled by default). When enabled, ai_query() used with external models could result in data egress to external model providers. See the documentation for region availability and DBSQL requirements for this functionality.

Documentation | Provide feedback

On ●

Databricks Assistant Autocomplete Public Preview

Allow workspace users to see real-time AI-based autocomplete suggestions

Documentation | Provide feedback

On ●

Genie Public Preview

No code experience for business users powered by LLMs. Analysts set up spaces which business users can use to ask questions. Requires Unity Catalog, Databricks Assistant and pro/serverless SQL warehouses.

Documentation | Provide feedback

On ●

You are now ready for the workshop!