databricks

0 In this workshop we will experience specific capabilities with the Databricks Data Intelligence Platform in the areas of governance, data engineering, task orchestration, SQL analytics, and data visualization. Along the way you will experience DatabricksIQ which is a first-of-its-kind Data Intelligence Engine that uses AI to power all parts of the Databricks Data Intelligence Platform. DatabricksIQ enables natural language interfaces so we can be more productive with data.

SETUP

Prior to the workshop, work with a workspace administrator to ensure that everyone has access to the Databricks workspace and that Unity Catalog is enabled. UC is **required** for this workshop.. As an administrator, create a catalog entitled "databricks_workshop" and add permissions to it for everyone that will participate in the workshop. Give them "all privileges".

Check connectivity to data.vermont.gov:443 before the workshop.

NOTE: The word "workspace" in Databricks has two meanings. There is the Databricks Workspace which is the UI that you log into with all components. There is also your personal workspace which is where your code for this workshop will be stored.

Have each user log into the Databricks workspace, navigate to catalog on the left pane and then to the databricks_workshop catalog. Have each user create their own schema with a naming convention that is easy to remember such as "userid_schema".

Email the files for the workshop to each user and have them save the files into their Downloads directory. The "Welcome to the Databricks" workshop may also contain code snippets which we'll use later as time permits.

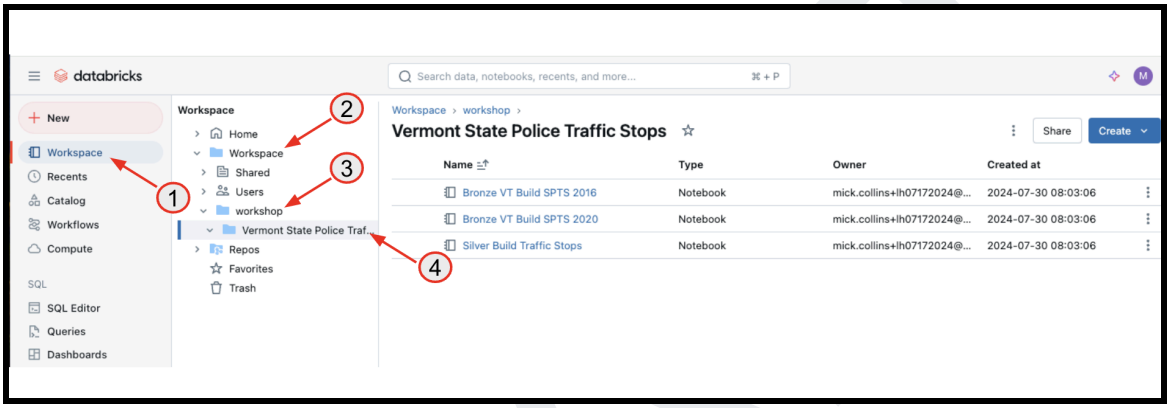
Module 1: Data Engineering and Task Orchestration

1. Each user will need to navigate to their personal workspace and create a folder entitled "workshop". Open that folder by double clicking it and choose the 3 dots in the upper right hand corner and choose to "import". Navigate to your Downloads folder and choose the file "Vermont State Police Traffic Stops.dbc".

NOTE: A "dbc" is a Databricks archive and is a mechanism for sharing notebooks between workspaces and users.

2. Review existing notebooks for ingesting and transforming data from the State of Vermont's open data portal. Follow the menu navigation to review the notebooks for ingesting the 2016, and 2020 traffic stop data, along with the notebook to consolidate the two results to a single dataset.

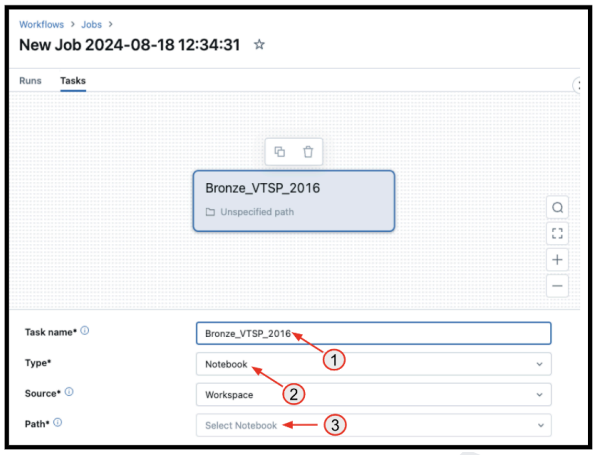
   Workspace ① > **Workspace** ② > **workshop** ③ > **Vermont State Police Traffic Stops** ④

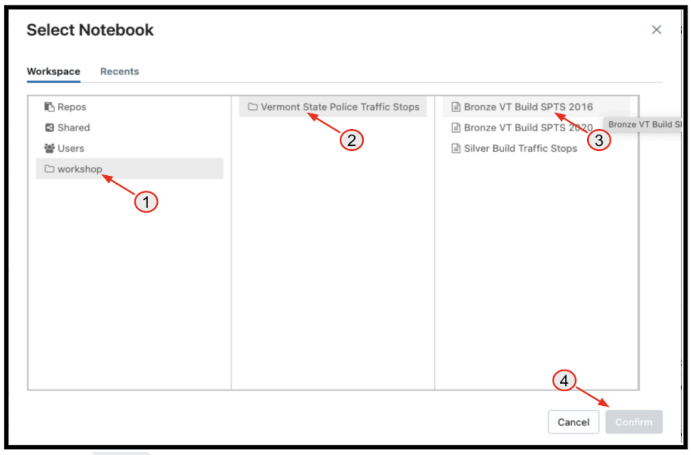


1. Open each of the *Bronze VT Build SPTS 2016*, *Bronze VT Build SPTS 2020*, and *Silver Build Traffic Stops* notebooks.
   a. Notice both notebooks producing bronze data use a combination of Markdown (for documentation), SQL and Python to complete their tasks. These notebooks are defaulted to using SQL (see the default setting next to your notebook title)
   b. Both *bronze* notebooks also receive parameters specifying which catalog and database (schema) to use when persisting data as tables using the default Delta Lake file format. Command cell 2 gets the `catalog` parameter, and command cell 3 gets the `schema` parameter.
   c. The *silver* notebook creates a consolidated table that is a combination of both *bronze* tables while standarding a few column names.

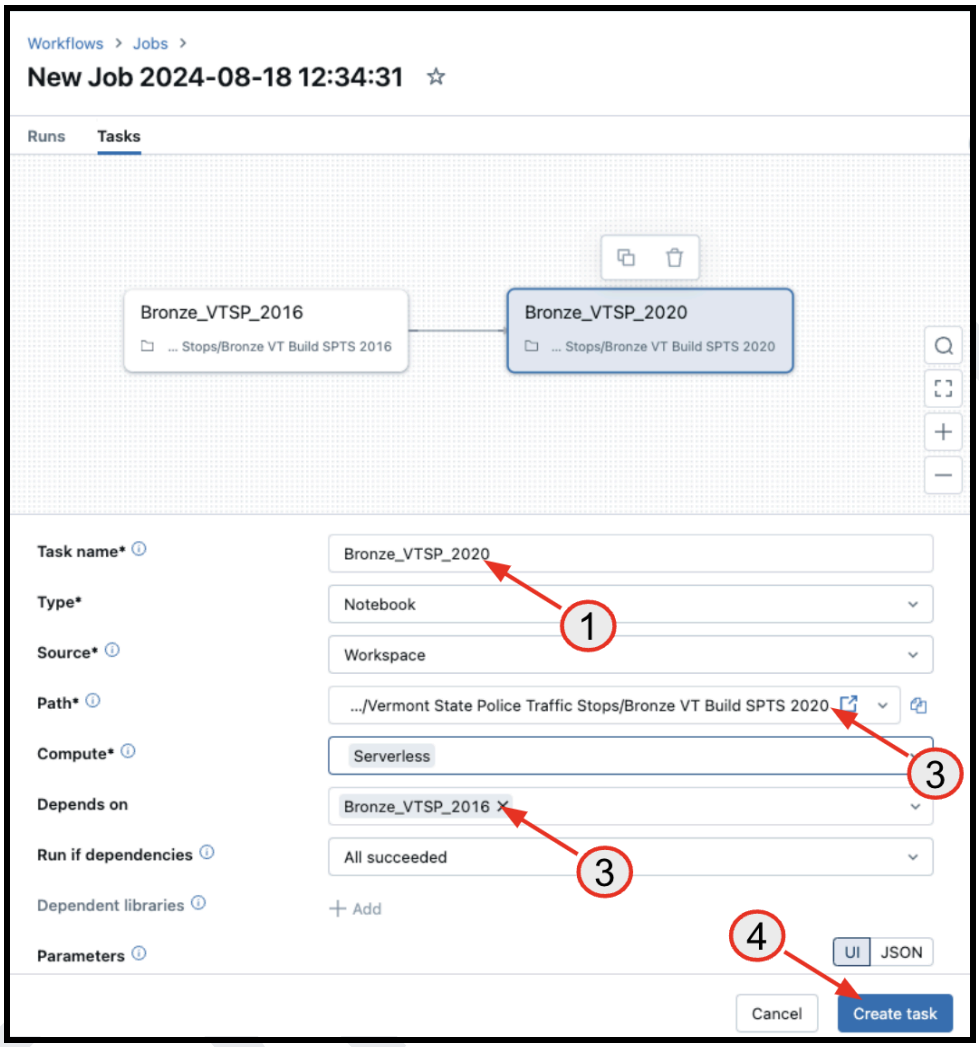
2. Create a new workflow, leveraging the notebooks just reviewed, to persist data in your own database (schema) using the following instructions.

a. To create your workflow navigate to Workflows primary page. Click Create job. The workflow canvas will have a blank task to configure.
Give the task an appropriate name, leave the Type as *Notebook*, and select the **`Bronze VT Build SPTS 2016`** notebook for this task based on what was reviewed in step 3 above, and as represented in the images below.
After filling in the task details, click [Create task].

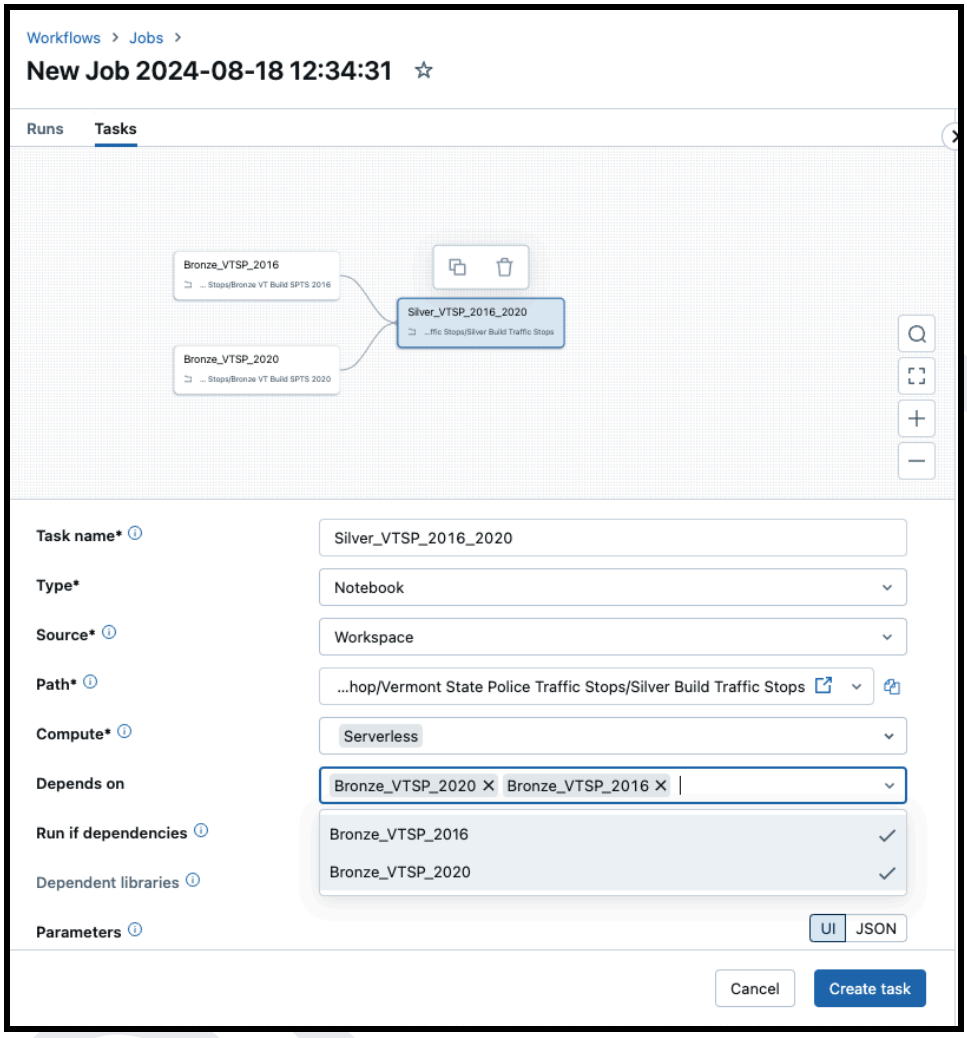




b. Add a second task by selecting the + Add task button, fill in the details to reference the notebook for the 2020 data, remove the task dependency, and click Create task

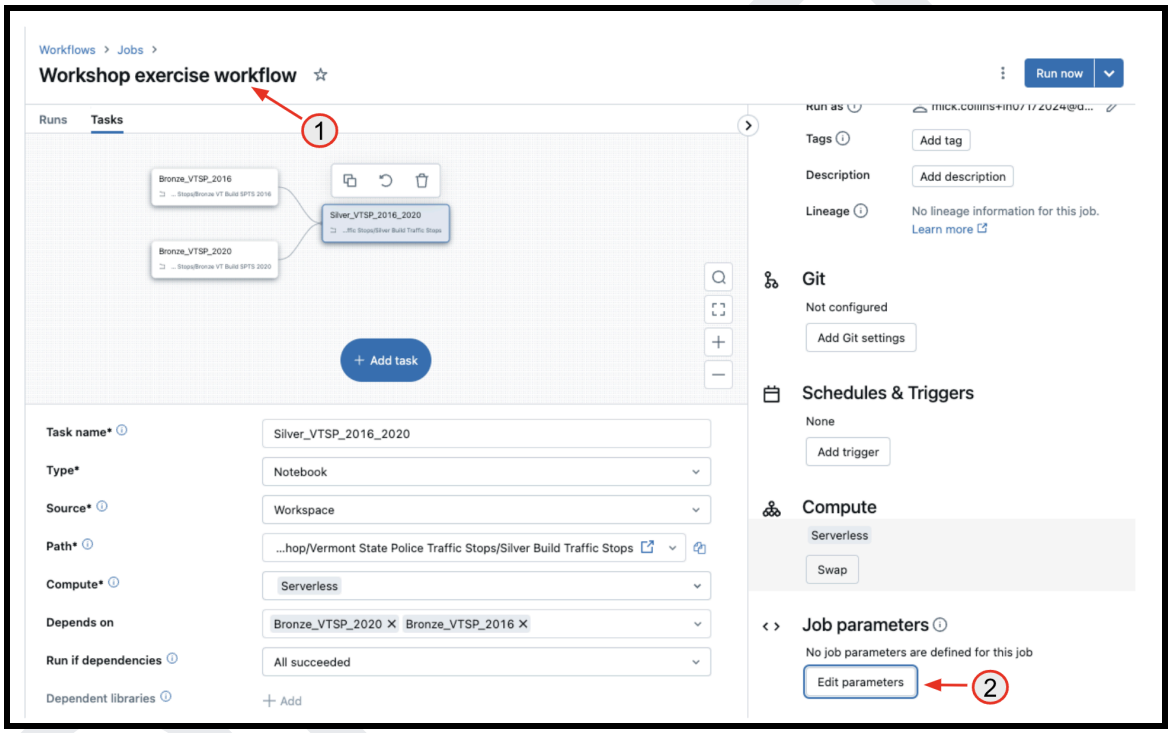


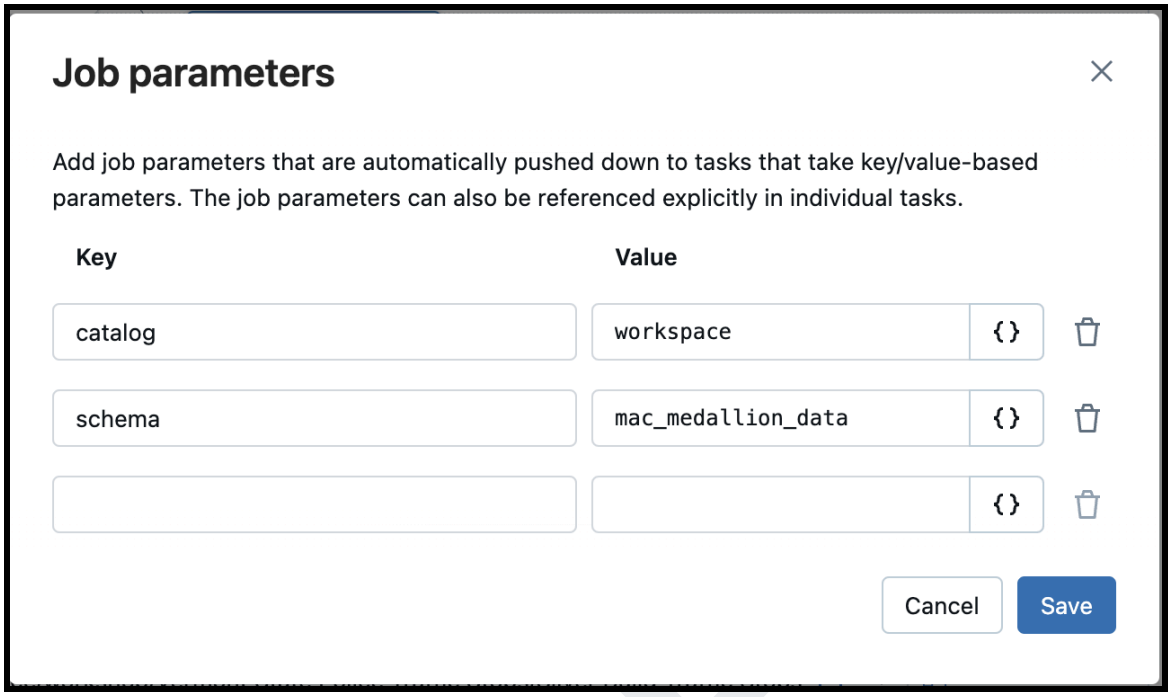
c. Add the final task creating the silver table, ensure the dependency for both the 2016 and 2020 tasks are selected as are represented in the images below, and click Create task.



d. (See images below) Give your workflow an appropriate name (1), for workshop purposes use your initials to differentiate from other participants. Also set the parameters (2) so the results will be saved in your new database.
Important! The parameter keys must be exactly as represented in the image below as the values match the notebook parameters. The value for the `catalog` parameter is the "databricks_workshop" which we set up earlier. However, the value for the `schema` parameter should be set to **your** schema. E.g. *`userid_schema`*.

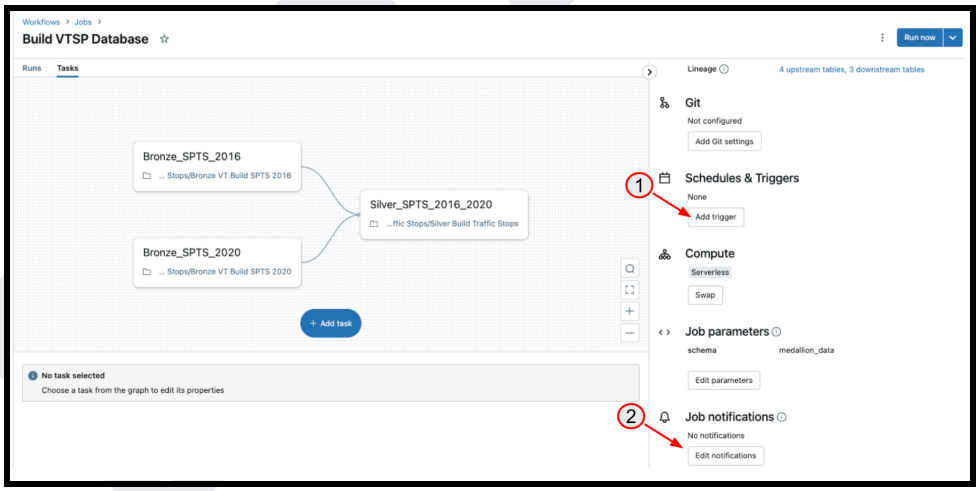
Save the parameter setting and click **Run now**





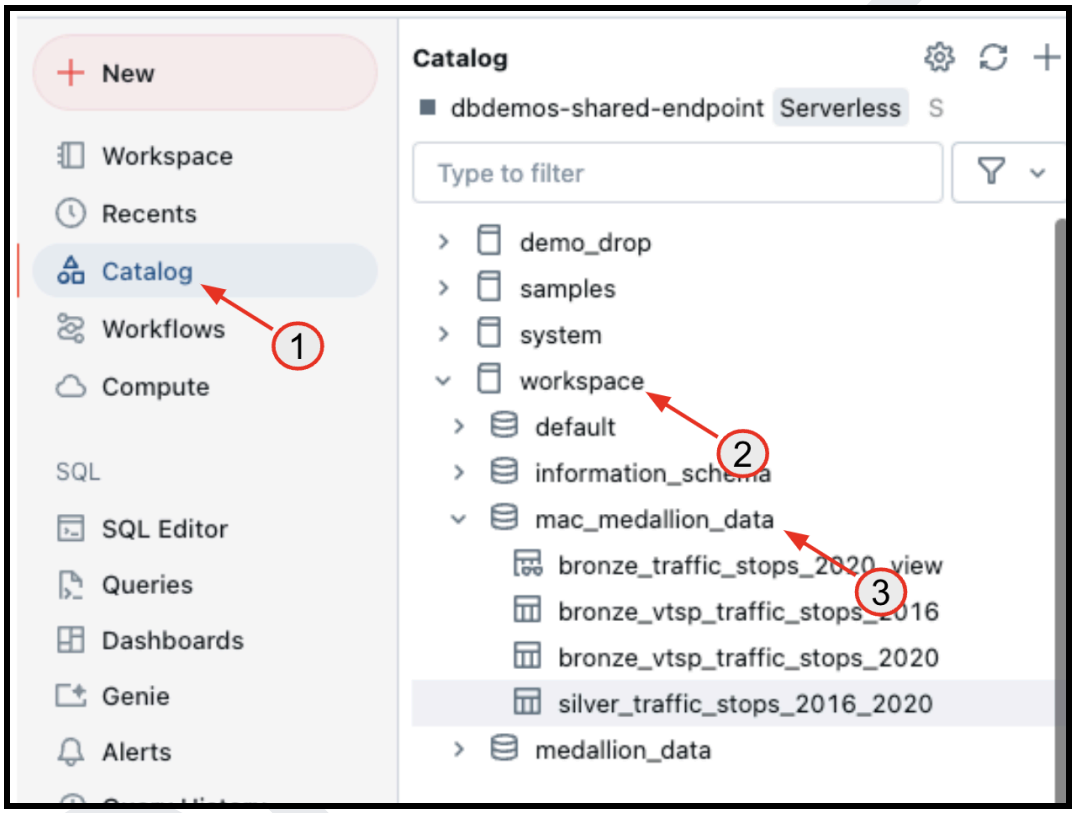
e. After submitting the job for execution you can follow the progress by selecting *View run*, or select the run from the Workflow Runs page.

f. Optionally, you may set a schedule for the job (1), or set notifications for the job start, success, failure, or duration warning (2).



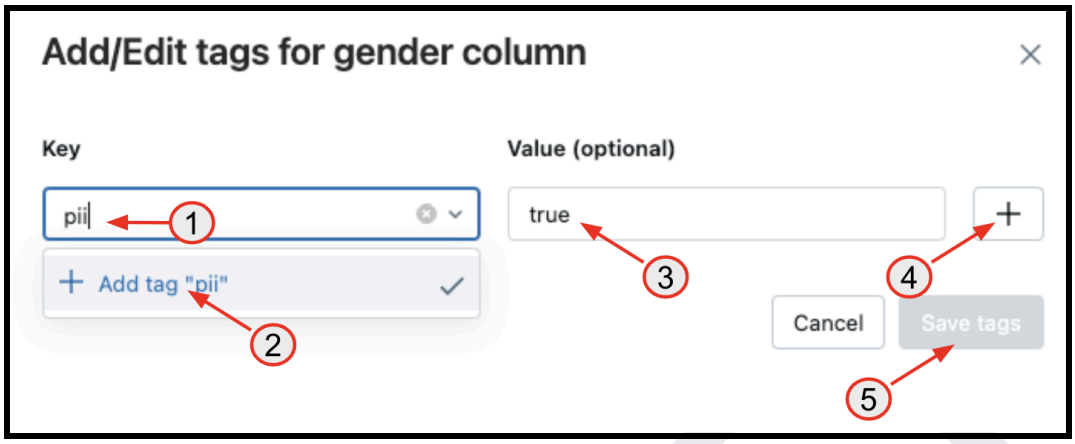
Module 2: Governance and Security

1. Open Unity Catalog (Catalog) on the main menu to review your new database (schema).
   You will notice that at this point, unless you are an administrator, you will only have visibility to your database, and none of the other newly created databases.
   Navigate from the main menu following Catalog ① > workspace ② > *your_database* ③



2. Review tables created by your workflow
   Select the ***silver_traffic_stops_2016_2020*** table.
3. Generate table and column descriptions
   Notice that the Databricks Data Intelligence Platform's AI engine automatically suggests a table description if one does not already exist.
   Select AI generate to generate column descriptions for the columns missing descriptions.
   Accept, edit, or reject descriptions per your preference.
4. Add table and column tags
   Click the button to add a tag to either the *gender* or *race* columns. Tags are a

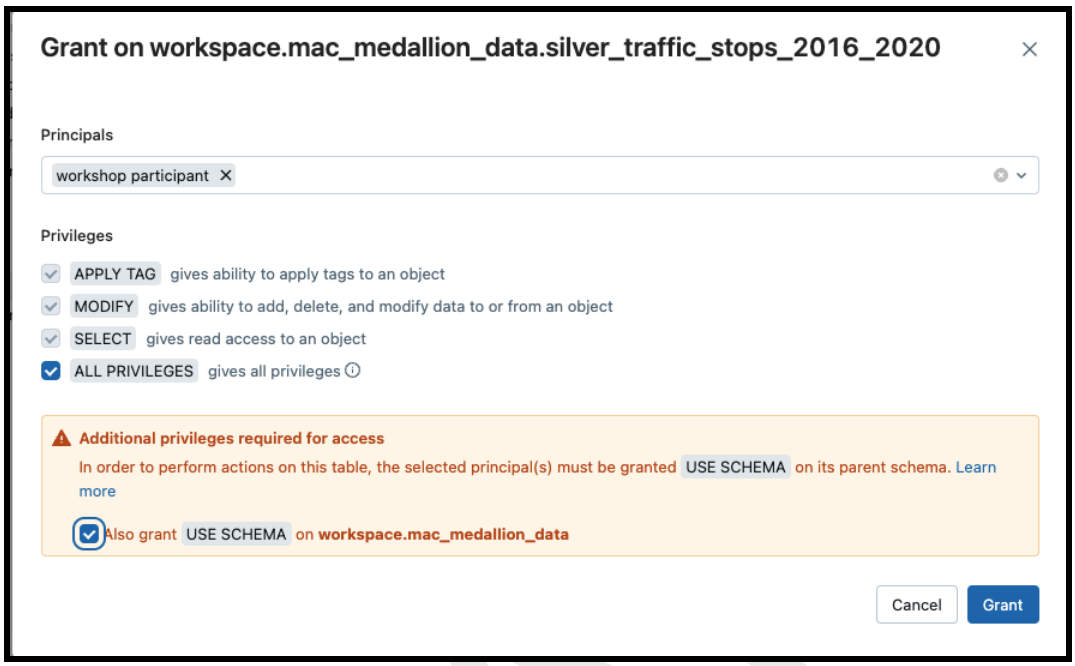
key:value pair. Enter the Key (1). If the Key does not yet exist, select the Add tag option (2), enter the value (3), accept the setting (4), and Save tags (5).



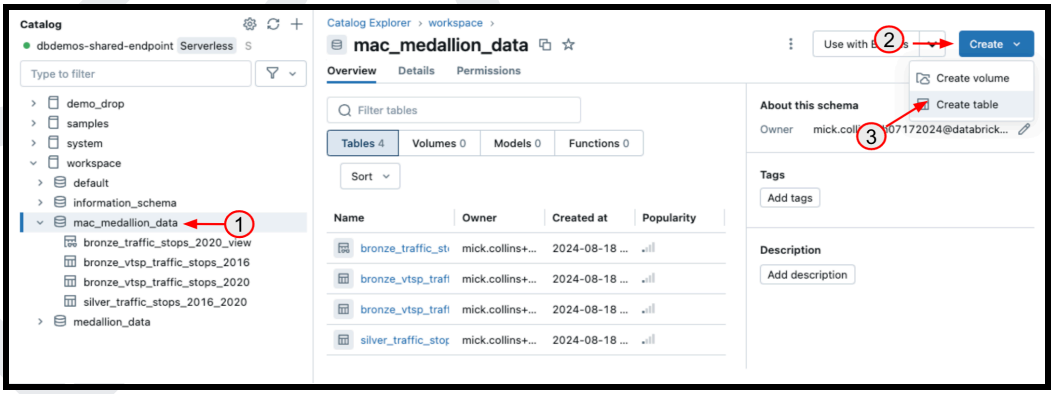
5. Review each of the tabs providing access to the table metadata (Overview, Sample Data, Details, Permissions, History, Lineage, Insights, and Quality (you may be prompted to select a compute cluster to see details behind some of the tabs).
    a. Sample Data: allows you to review the data without executing a separate query.
    b. Details: is a technical view of the data including the storage location, creation timestamp, file type (Delta is the default), and etc.
    c. Permission: provides a view of current permissions for users and groups, and allows privileged users to grant access. You will grant access in step 6 below.
    d. History: displays version history for the table. You may see multiple history levels for your newly created table due to the API building the table in batches.
    e. Lineage: provides both a tabular and graphical view of both upstream and downstream lineage. Select the See lineage graph button for the graphical view. Select `Show more columns` for one of the tables, and select a column to see the column mapping. Table lineage was automatically captured from your workflow.
    f. Insights: show recent table activity.
    g. Quality: provides visibility to data quality for the table. This function needs to be explicitly enabled at the table level and will capture data quality over time.

6. Grant access to the '**workshop participant**' group from the `Permissions` tab. Refer to the image below. Table level access also requires USE SCHEMA on the

parent schema. Accept the option to also grant this privilege.
As access is granted to the group you will begin having visibility to other schemas and content.



7. Add a new table (manually from CSV) following the instructions below.
    a. Navigate back to the database overview page for your newly created schema (1), click Create (2) > Create table (3), and follow the process to import the nyc_boroughs.csv that was downloaded earlier.



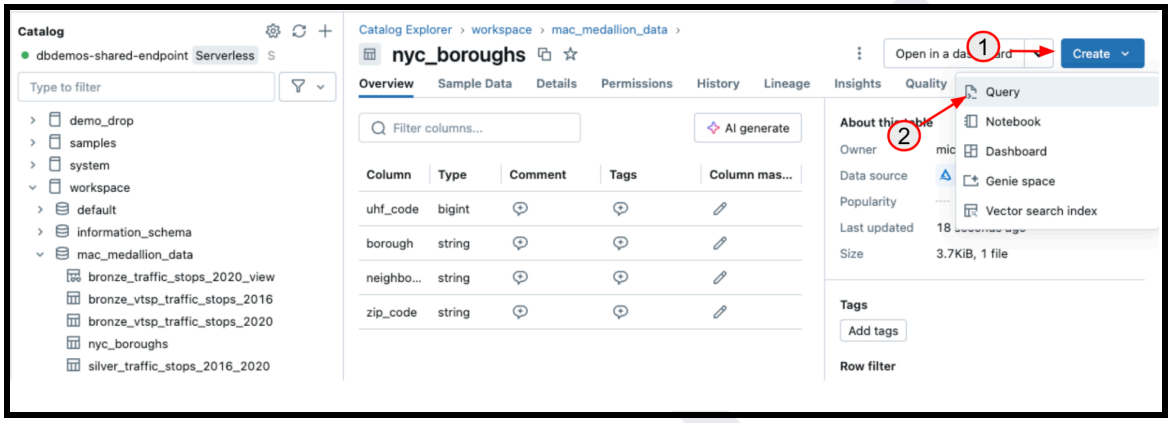
    b. Change the data type of the zip_code column from `Int` to `String` by selecting the  button, and then *string*. Save the table 

.

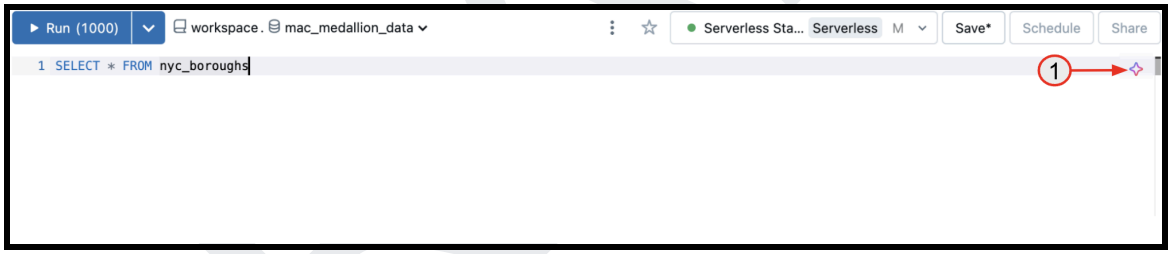
Module 3: SQL Analytics

1. Open the new table in SQL

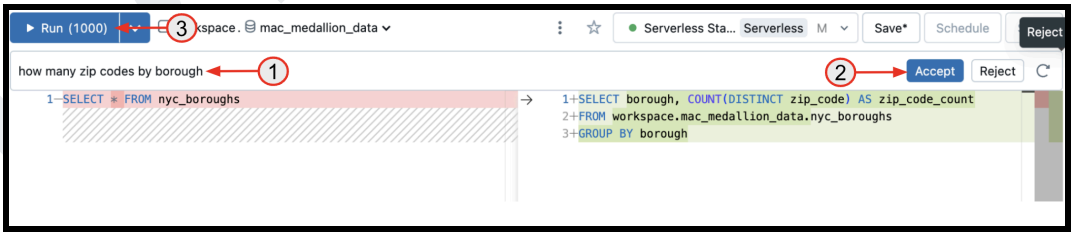
   From the table overview page create a query of this new table by selecting the Create button (1), and Query (2). Notice that you will navigate to the **SQL Editor** screen. Select *Run* if you want to see the results of the **`Select *`** command.



2. Open Databricks Assistant to modify this query as is represented in the image below.



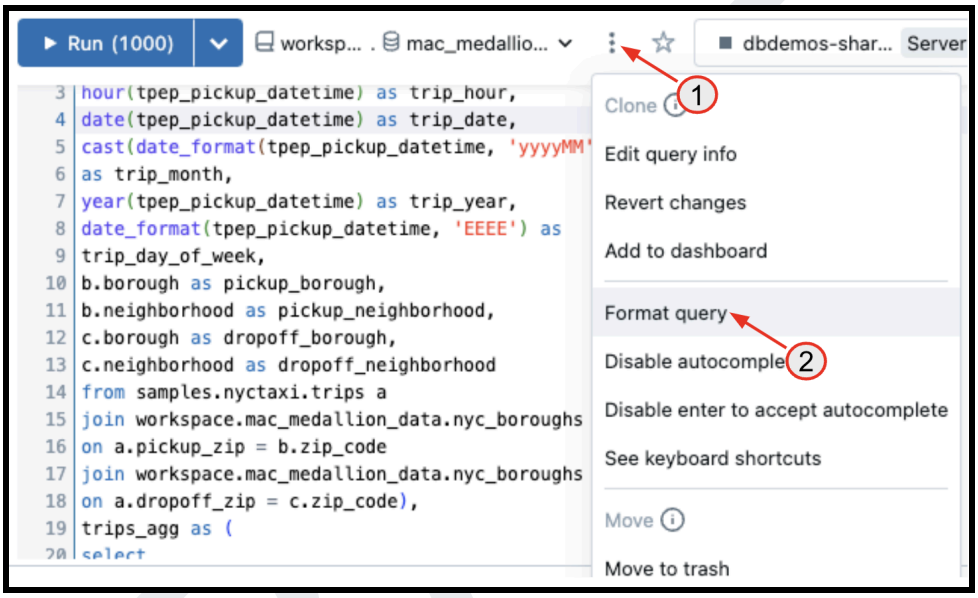
   a. Type **`how many zip codes by borough`** in the dialog box (1), hit return, accept the results (2), and run (3) the command.



3. Create a new SQL statement

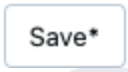
   We will create a query using a Common Table Expression (CTE)

a. Open a new tab in the SQL Editor by selecting the + icon.
b. On your workstation, open the CTE_query.txt document. Copy the full contents (^A, ^C). Paste the text in the new query tab.
c. In the CTE query replace **`MY_SCHEMA`** with the name of the schema you created. This is on two lines (lines 15, and 17). Run the query and review the results.
d. Format the query to improve readability by selecting the kabob menu (1), and **Format query** (2)



4. Save query

Double click **New query** in the query tab, and give your query a name, and click



Your query will be saved in your default workspace folder. You can see the results by selecting **Workspace** in the main menu.

5. Add visualization

   Add a visualization to this query. Select + and Visualization in the query results pane. Provide a title (1), set the visualization type as **Line** (2), select `trip_day_of_week` for the X column (3), add a column for the Y axis and set the value to `trips` (4).

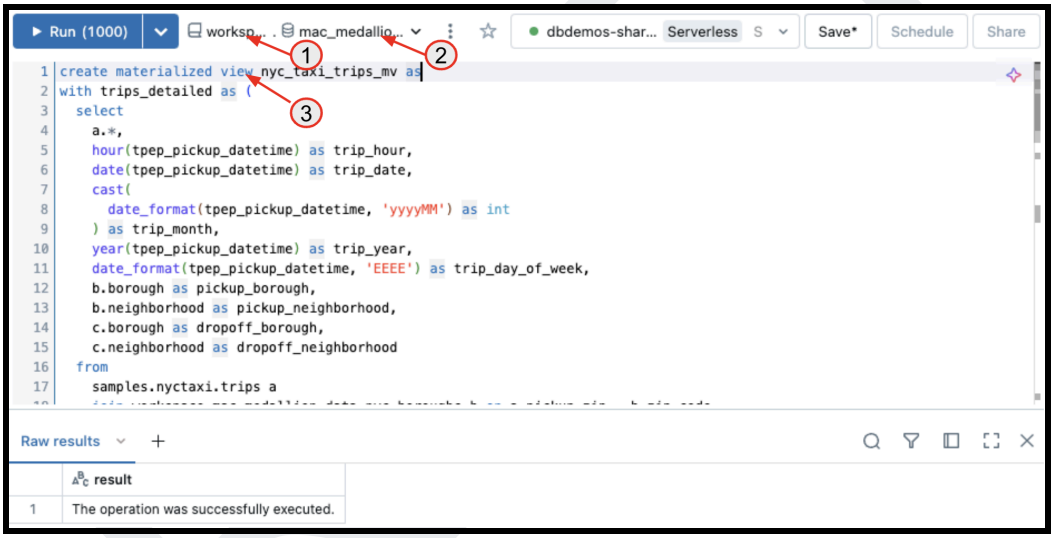
6. Schedule query (optional)

   Click Schedule to schedule your query.

7. Create a materialized view from the CTE query
    a. Select and Copy the entire query text
    b. Create a new query by selecting + on the query tabs.
    c. Set the catalog (1) and database (schema) (2) to reference your database.
    d. Paste the CTE as the new query.
    e. Add a new line at the top of the query with the following text:

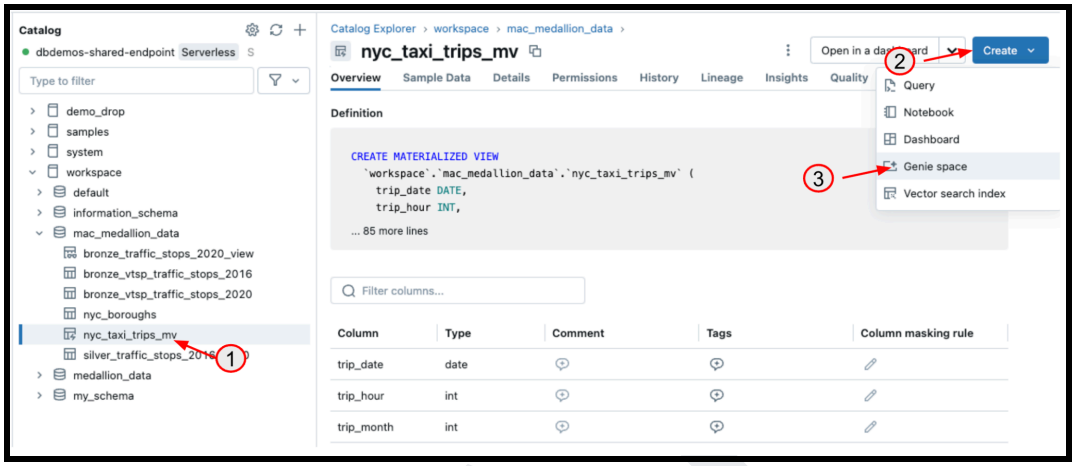
    `create materialized view nyc_taxi_trips_mv as` (3)

    (this text is also in the mv.txt file from the workshop resources).
    f. Select ▶ Run (1000) to execute the CREATE statement (this operation will take approximately 90 seconds)

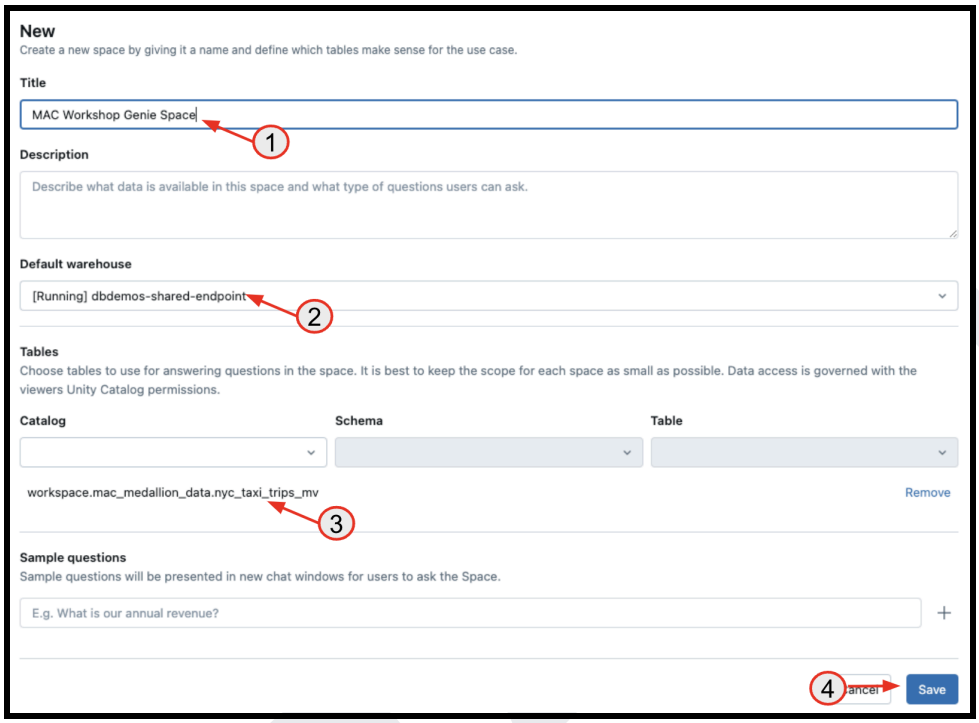


Module 4: Converse with the data by creating a *Genie Space*

1. In Unity Catalog, navigate to the schema where you created your materialized view (MV) ①
2. Select the MV ②, and create a Genie Space ③

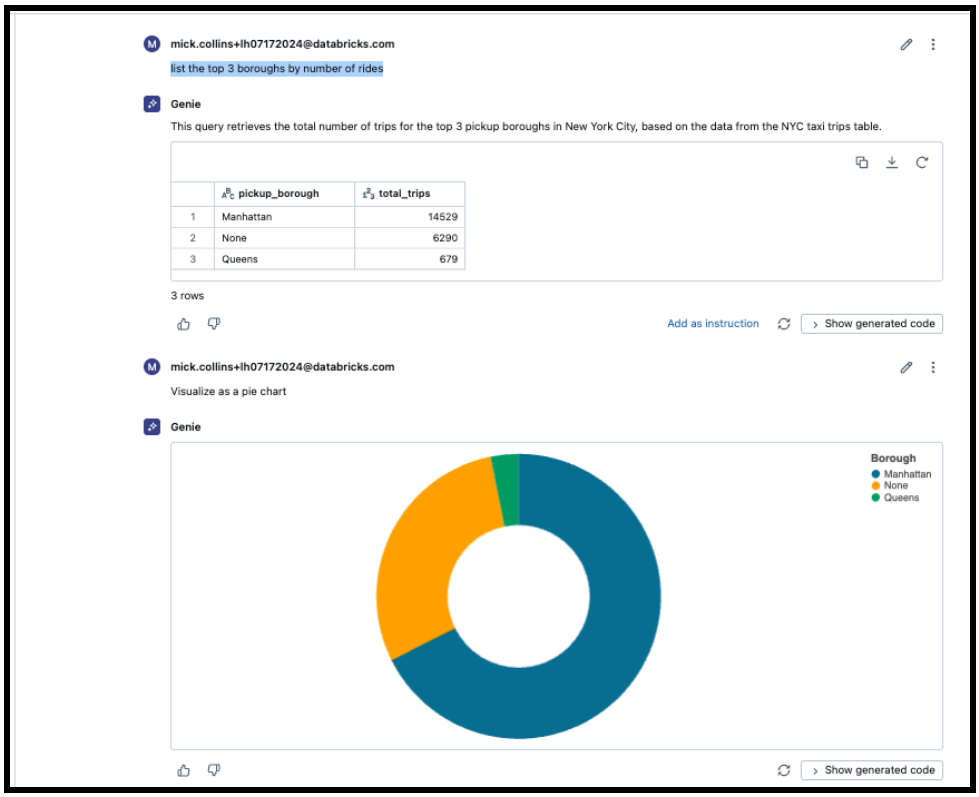


3. We will minimally configure the Genie Space. Provide a title (1), select the default warehouse as the compute (2), the MV is already included (3). Optionally you can add other related tables (skip this), and Save (4)



4. Ask a question of the data (e.g. ***list the top 3 boroughs by number of rides***)

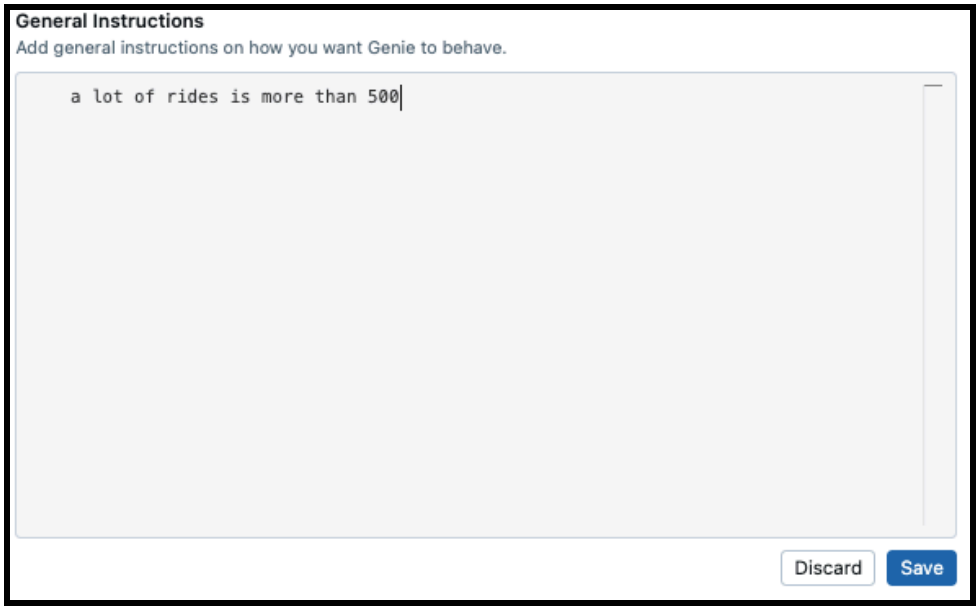
5. Graph the results as a pie chart (click the Pie Chart button)



6. Provide instructions for Genie to understand terms that may be organizationally specific, such as fiscal year, etc. For our example we will tell Genie that **a lot of rides is more than 200**.(Use 200, not 500 as in the image below)

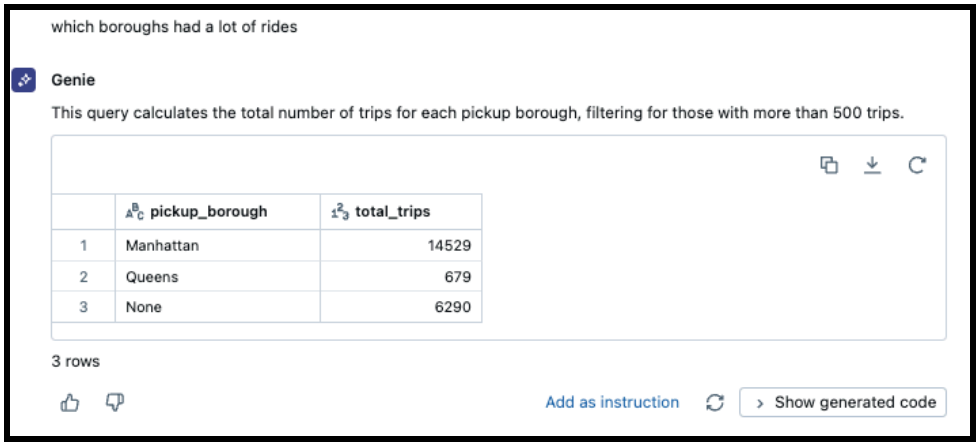
   Select the Instructions button 

. Enter the instruction and Save



7. Return to the chat 

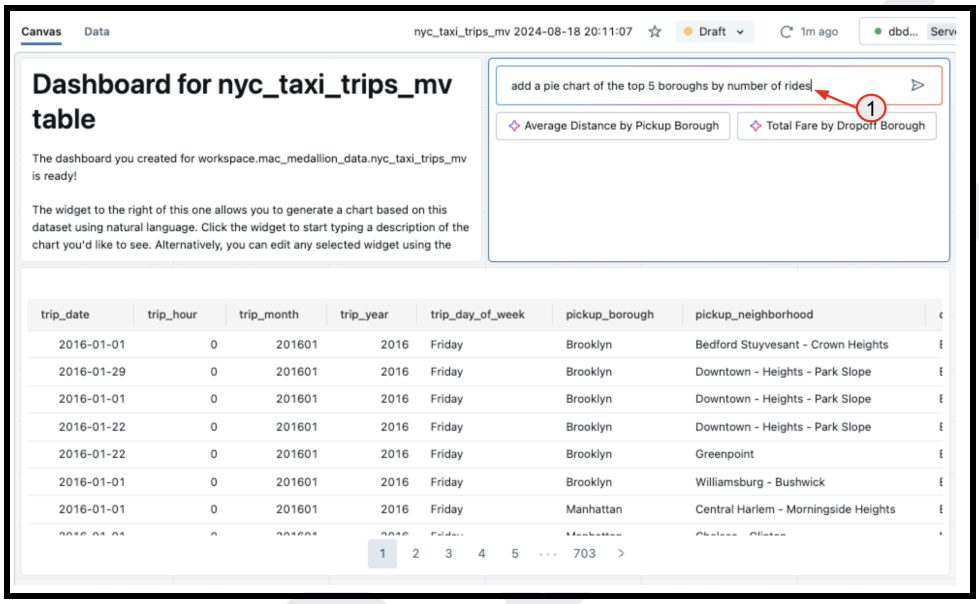
and ask a new question, such as **which boroughs had a lot of rides**

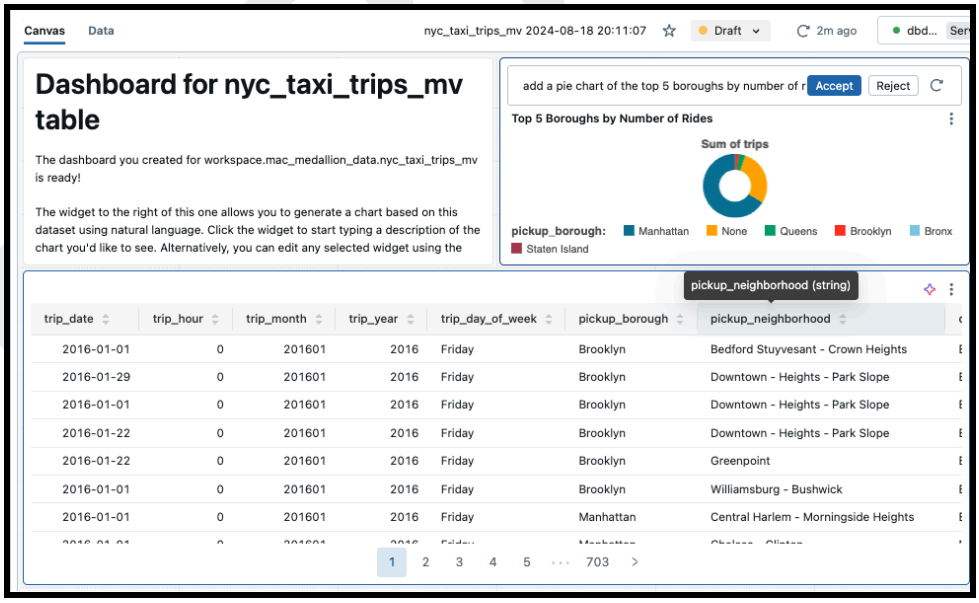


8. Notice, you can show the generated code if desired

Module 5: Data Visualization

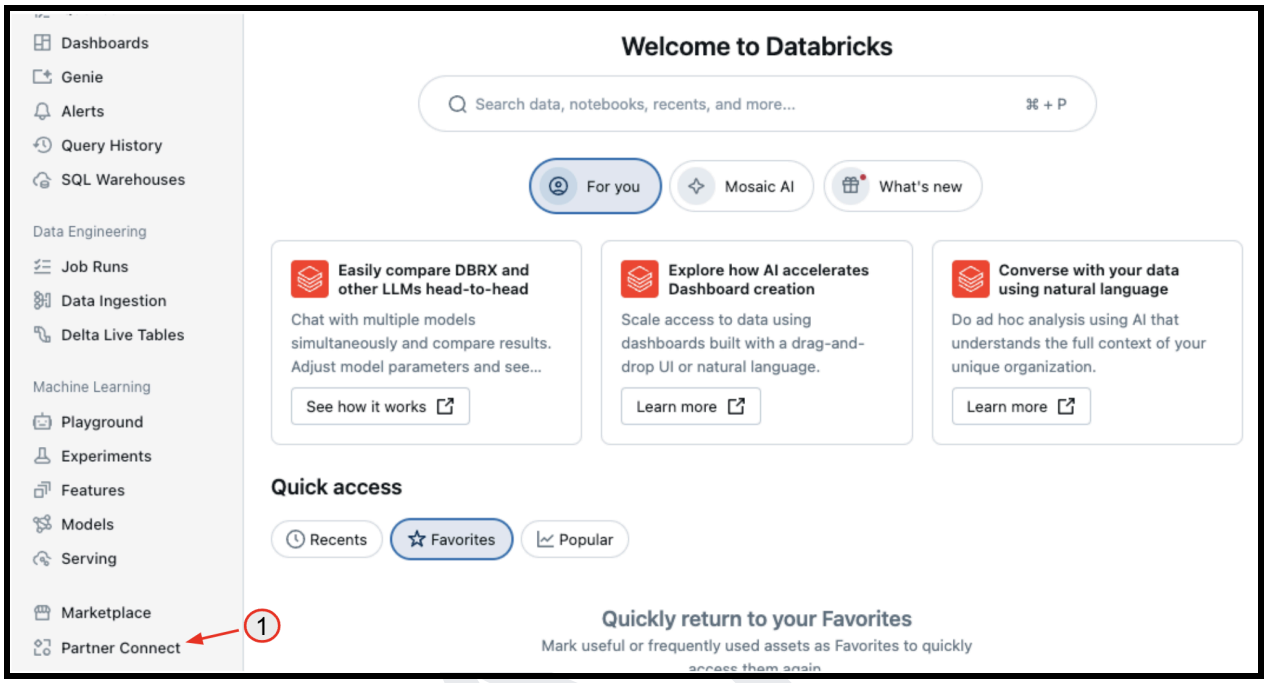
1. Return to your MV in Unity Catalog
2. Use the Create button, and select `Dashboard`
3. Use Databricks Assistant to add a pie chart to the dashboard as represented in the image below (1).





Module 6: Connecting partner tools (optional)

1. Select Partner Connect at the bottom of the main menu (1).



2. Explore the options for leveraging external tooling for data visualization, and other categories.