Data pipelines, views & data products [Python-based]

Data Universe 2024

Note: This is fundamentally the same as the **Data pipelines, views & data products [SQL-based]** workshop, but leveraging Python and the PyStarburst Dataframe API instead of direct SQL.

Table of Contents

Lab 1: Introduction and setup

Learning objectives

- Describe the lab scenarios and goals.
- Setup a free Starburst Galaxy account.
- Understand how to continue using Starburst Galaxy after the end of the lab.

Activities

1. Lab overview
2. Create a Starburst Galaxy account
3. Housekeeping items

Part 1: Overview of labs

You are a data engineer at Nintendo. You were asked to gather some data about Pokemon Go and help the marketing team figure out which Pokemon spawns are most common in the San Francisco Bay Area. You need to help both teams by discovering, transforming, and cleaning the data from multiple sources.

Step 1 - Purpose of labs

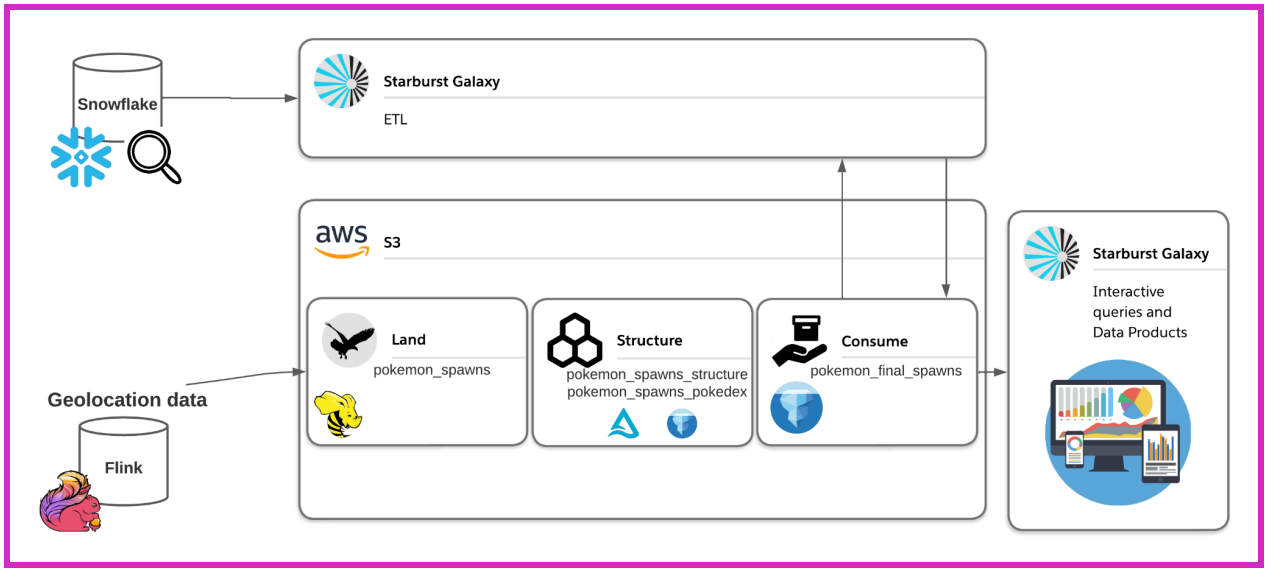
These labs use Pokemon Go data being ingested into S3, which contains the encounter information of the Pokemon including geolocation data of where the Pokemon spawned, and how long the Pokemon was at that location.

Importantly, you do not have any information about the Pokemon's abilities, that's all contained in the Pokedex in Snowflake. This has all the stats on your desired Pokemon including type_1, type_2, catch rate, and more.

Step 2 - Description of activities

To make sense of data from multiple sources, you will create a reporting structure in your data lake. First, you will use schema discovery to understand the data in your data lake. You will then use Starburst Galaxy to read the data in the land layer, then clean and optimize that data into more optimal ORC files in the structure layer. In the last step, you will join the geolocation information from AWS S3 with the Pokedex lookup table in Snowflake into a single table that is cleaned and ready to be utilized by our teams. After completing the discovery, location, governance, and query stages, you will end the lab by creating data products, which package the dataset in a curated way for easy consumption.

You will also be introduced to Gravity and Great Lakes connectivity in Starburst Galaxy, which are two awesome features that make it easy to run data lake analytics. Both these features will be demonstrated throughout the lab guide.



Step 3 - Data challenge

This data challenge involves two key missions:

- Create a final table output combining data from both structure tables.
- Create a data product answering two specific business questions from the marketing department.
    a. What are the easiest and most popular Pokemon to catch in San Francisco by Type_1?
    b. Find the total number of Pokemon caught for each Type_1 and Type_2 pairing. Also, find the average catch rate.

Note: Easiest is defined by having a high catch rate. A high catch rate is greater than or equal to 100. Also consider that in the structure layer, you filtered out data that did not exist in the San Francisco Bay Area.

Part 2: Create a Starburst Galaxy account

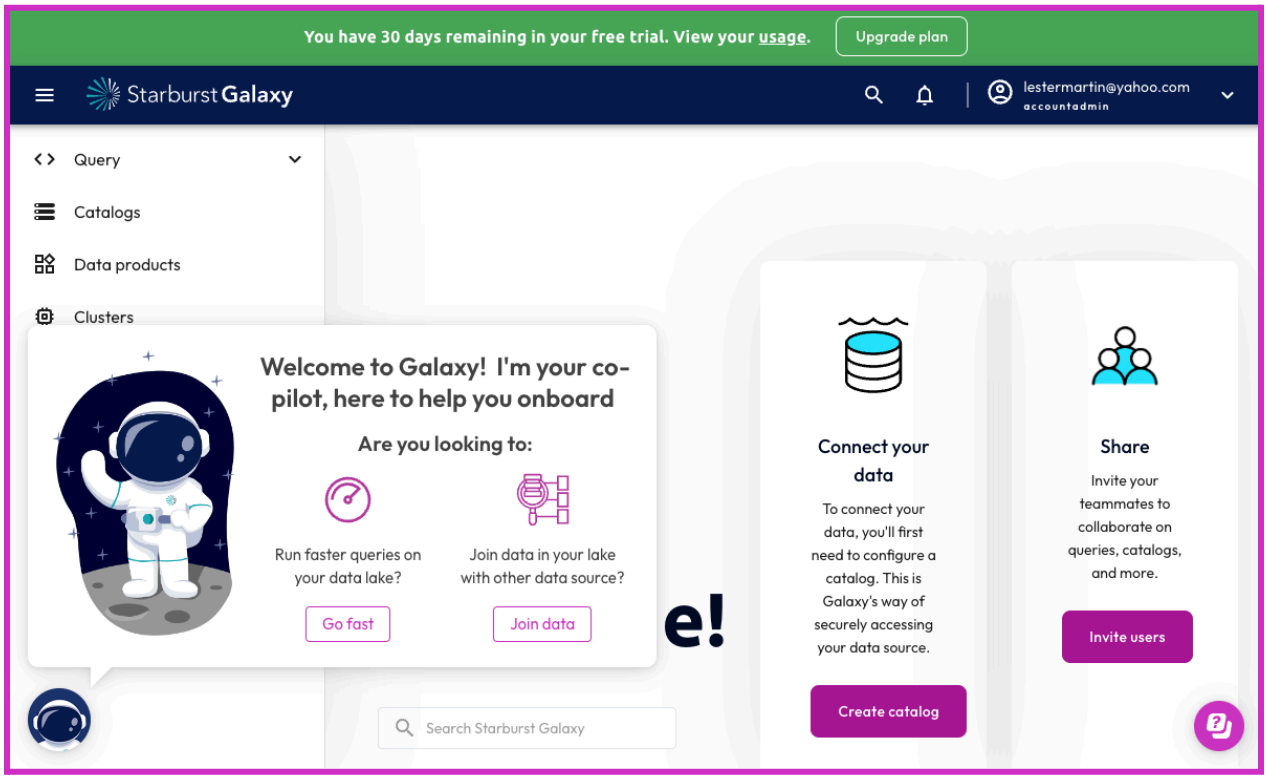
If you have already registered for Starburst Galaxy, you may skip the remainder of this Lab.

To sign up for Starburst Galaxy, follow the instructions on the free registration page at https://www.starburst.io/platform/starburst-galaxy/start/.

Note: You will receive an "invitation" email. Please check your spam or junk folder if it does not immediately arrive in your inbox.

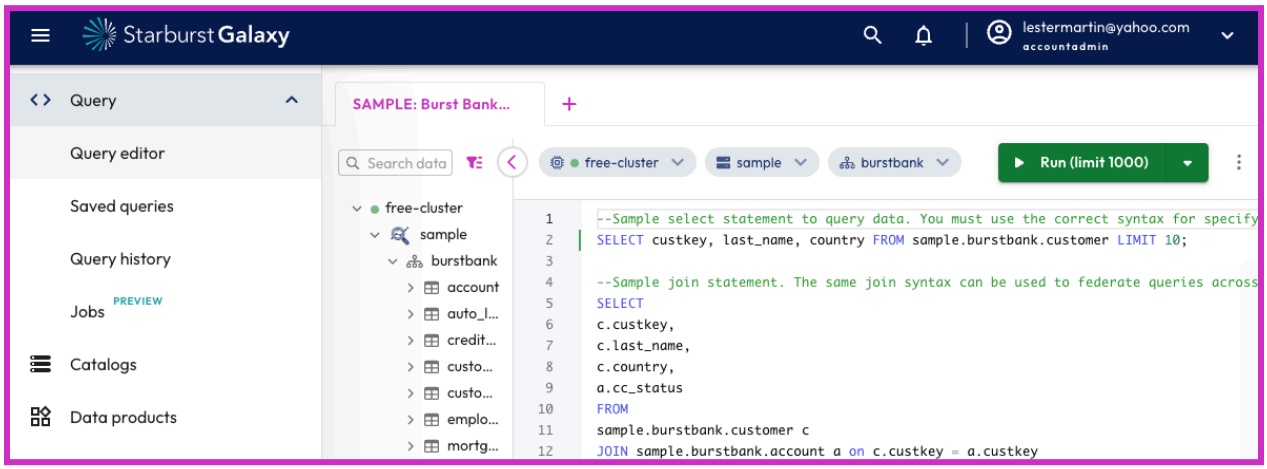
After you have entered your confirmation code, set a password, and selected your new domain name, you will be presented with a series of questions about your desired usage for Starburst Galaxy. Complete these with whatever you choose to share.

Eventually, you will likely be presented with a page similar to the following screenshot.



Click on the astronaut helmet icon in the lower-left to silence the pop-up coming from it.

At this point, you should see something similar to this to indicate you are fully configured.



Part 3: Housekeeping

As part of the Galaxy Lunch and Lab, the credentials for Amazon S3 and Snowflake will be available for up to 1 week. It is critical to understand that time, **YOU WILL BE UNABLE TO RUN ANY QUERIES AGAINST THE TWO CATALOGS CREATED IN THIS LAB**.

If you want to continue exploring Starburst Galaxy, here are some other free projects and helpful links you can utilize with your Starburst Galaxy account:

- Federate multiple data sources tutorial
- Starburst Academy
    - Starburst Galaxy courses
    - Data foundations courses
    - Learn SQL courses
    - Starburst foundations
- Starburst Galaxy documentation
- Near Real-Time Ingestion tutorial

Lab 2: Connect to data sources

Learning objectives

- Describe the process for creating catalogs that connect AWS S3 and Snowflake.
- Demonstrate how to connect a catalog to a cluster in Starburst Galaxy.

Activities

1. Create Amazon S3 catalog
2. Create Snowflake catalog

Part 1: Create Amazon S3 catalog

Objective

You're going to begin by setting up an AWS S3 catalog in Starburst Galaxy and connect the Pokemon spawns geolocation data.

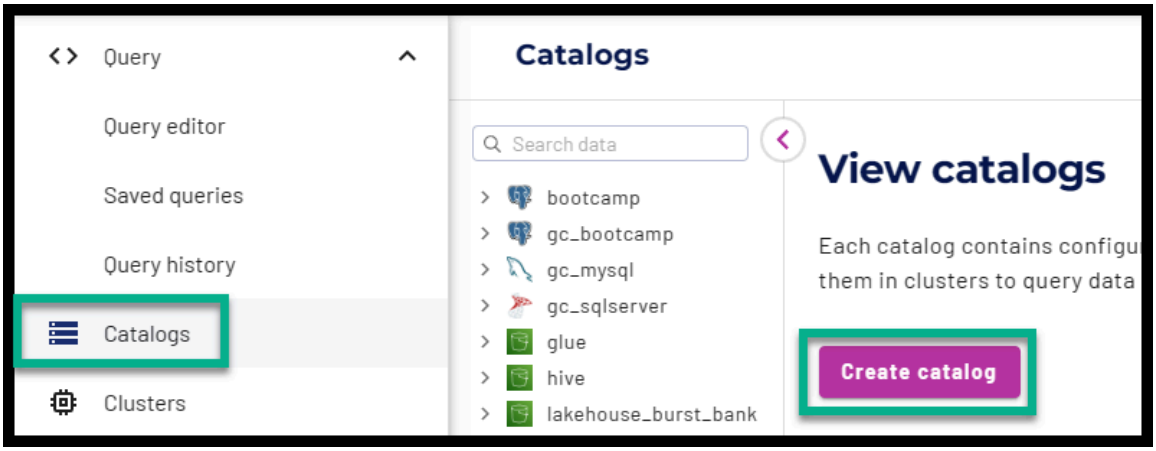
If you configured a data lake catalog in one of the other workshops, you may skip these steps in this Part and move on to Part #2 of this Lab.

Step 1 - Sign in and verify your role

Sign in to Starburst Galaxy. Use the account credentials you previously created. In the upper right corner of the screen, confirm that your role is set as `accountadmin`.

Step 2 - Create Amazon S3 catalog

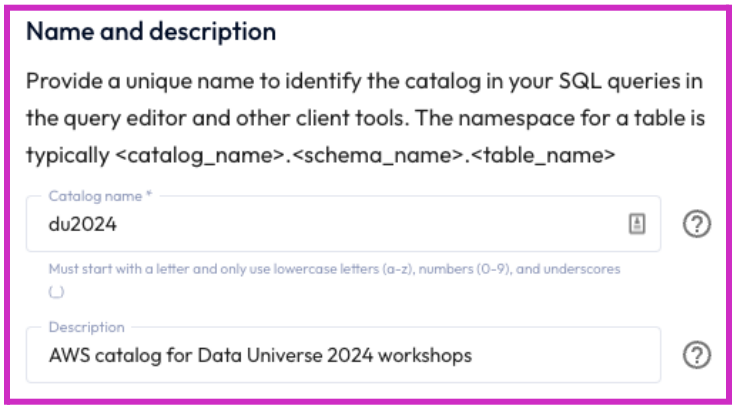
Click **Catalogs** in the menu on the left and then click the **Create catalog** button.



Click the **Amazon S3** tile.

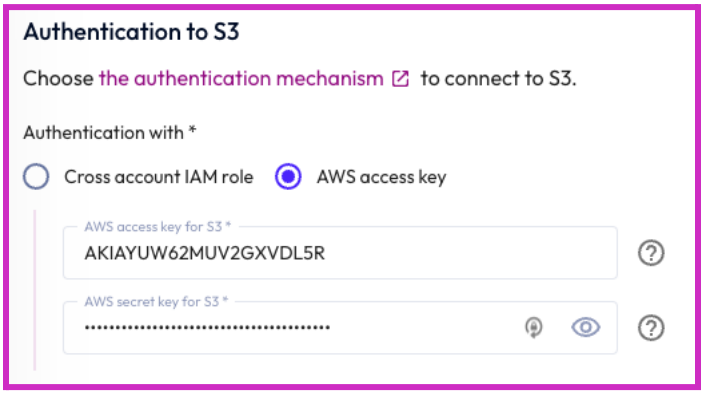
Use the information below to configure your catalog.

Catalog name: `du2024`
Description: `AWS catalog for Data Universe 2024 workshops`

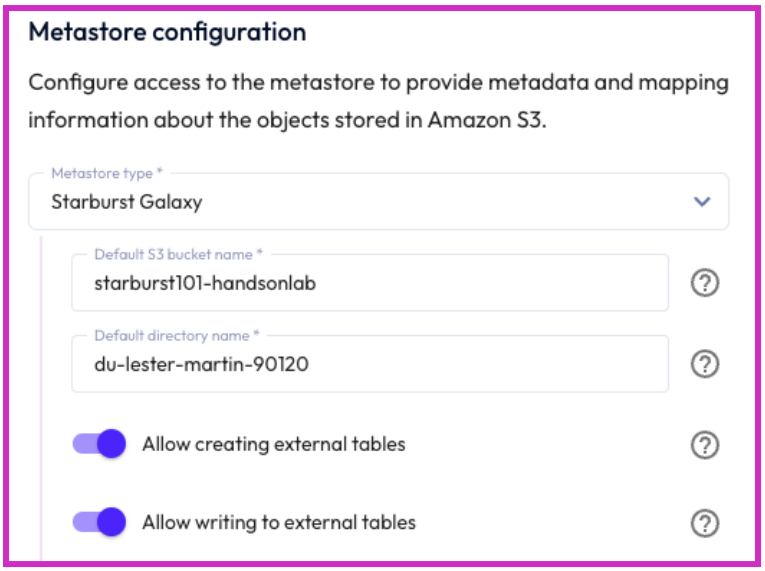


Authentication with: select the radio button **AWS access key**
AWS access key for S3: `AKIAYUW62MUV2GXVDL5R`
AWS secret key for S3: `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`

Note: These AWS credentials will only be operational through the weekend following the webinar. You will not be able to utilize this catalog beyond that point and should remove it from your Galaxy configuration.

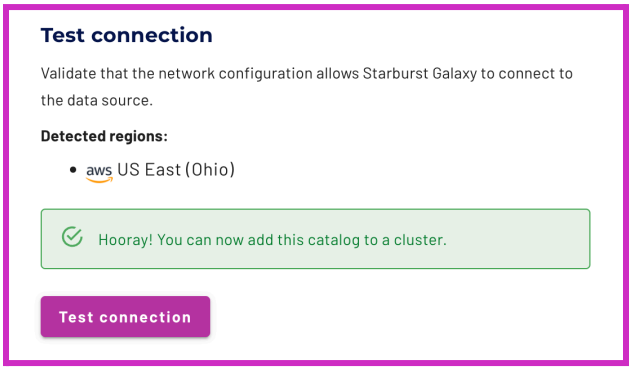


Metastore type: select the radio button **Starburst Galaxy**
Default S3 bucket name: `starburst101-handsonlab`
Default directory name: du-fname-lname-postalcode (ex: `du-lester-martin-90120`)
Allow creating external tables: enable the slider
Allow writing to external tables: enable the slider

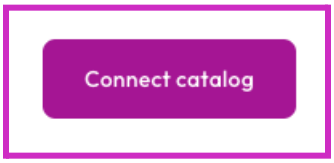


Default table format: ensure the radio button is selected to **Iceberg**

Validate the connection by hitting **Test connection.** Your catalog should return the same message indicating that you can now add the catalog. Confirm you see the **Hooray! You can now add this catalog to a cluster** message.

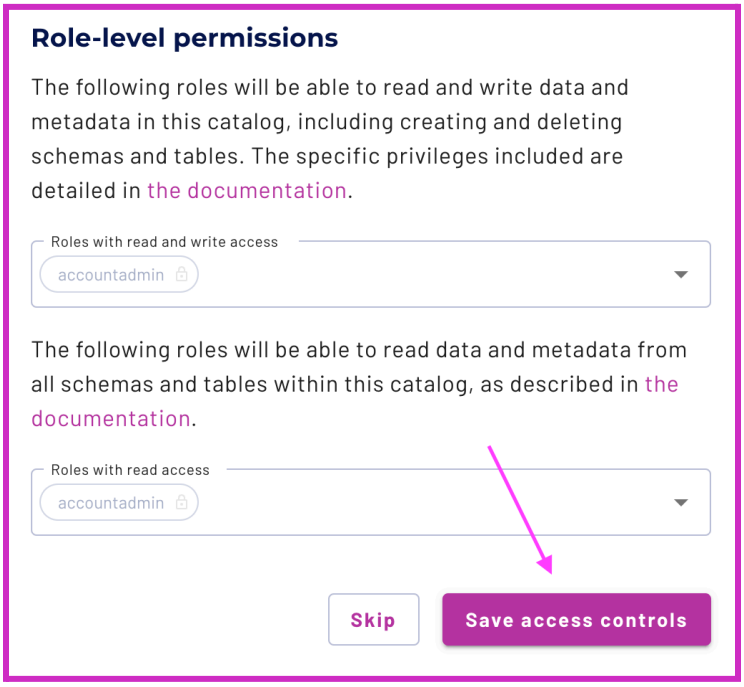


Select **Connect catalog**. This will save the credentials for your Amazon S3 catalog.



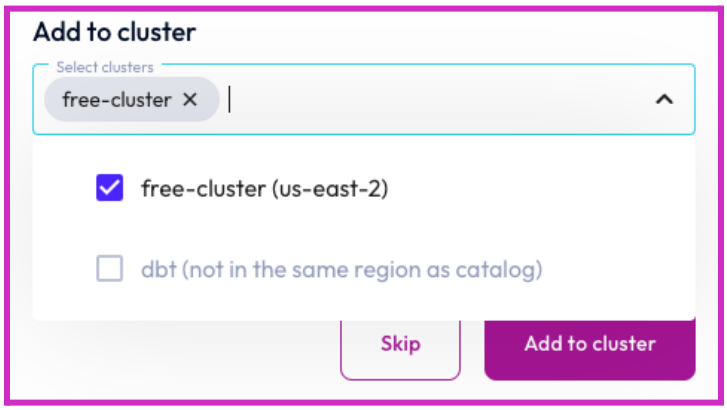
Step 3 - Set permissions

Next, accept the default permissions for your catalog by selecting the button **Save access controls**.

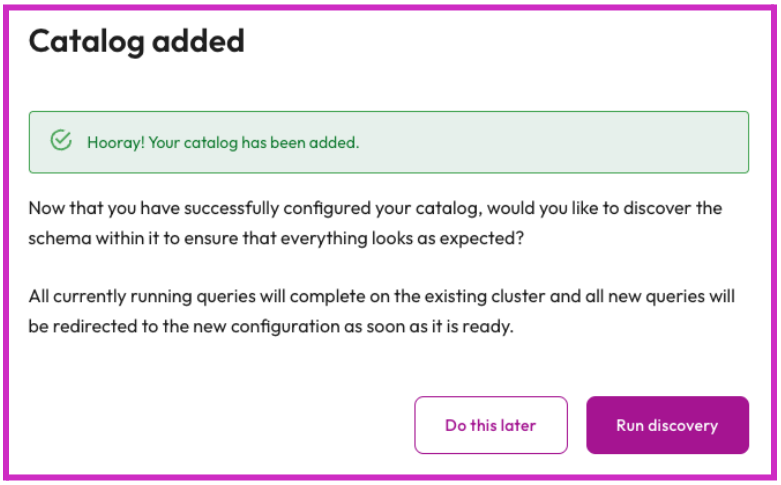


Step 4 - Add to cluster

Select `free-cluster` in the **Select clusters** pulldown and then click on **Add to cluster**.



Click **Do this later** in the **Catalog added** pop-up.



Part 2: Create Snowflake catalog

Objective

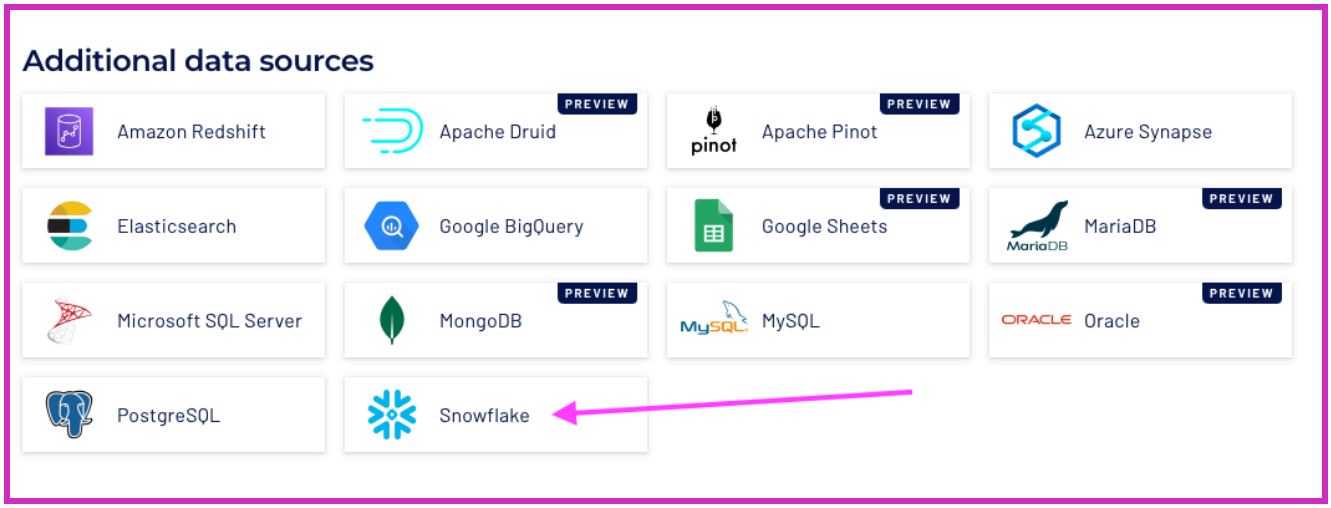
Now it's time to create a Snowflake catalog alongside your AWS S3 catalog. Later, this will allow us to federate across the two data sources.

If you configured a Snowflake catalog in one of the other workshops, you may skip the REMAINDER of the steps in this Part and move on to Lab #3.

Step 1 - Create Snowflake catalog

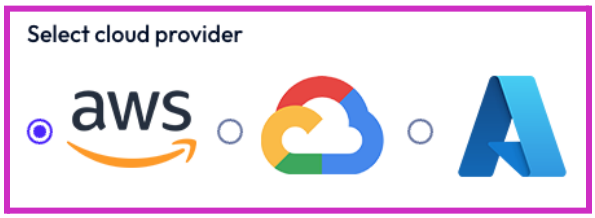
From the catalog page, select the **Create catalog** button to create the second catalog.

Choose **Snowflake**.

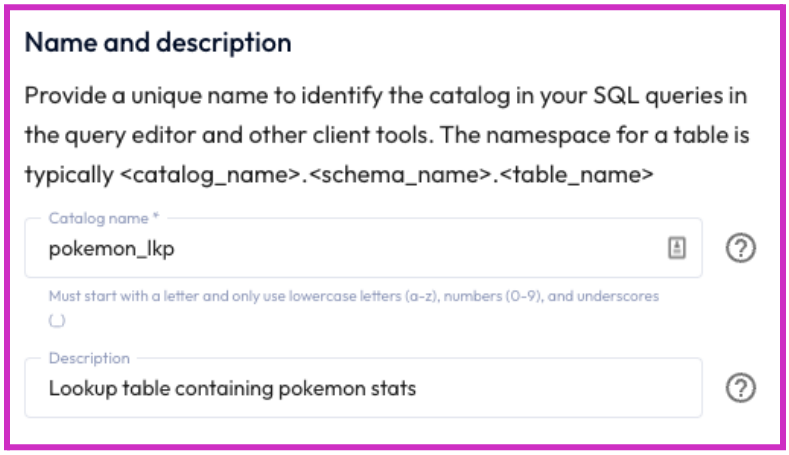


Using the list below as a guide, configure your catalog to query objects in Snowflake, specifically the Pokedex information. Provide the necessary credentials to authenticate the connection.

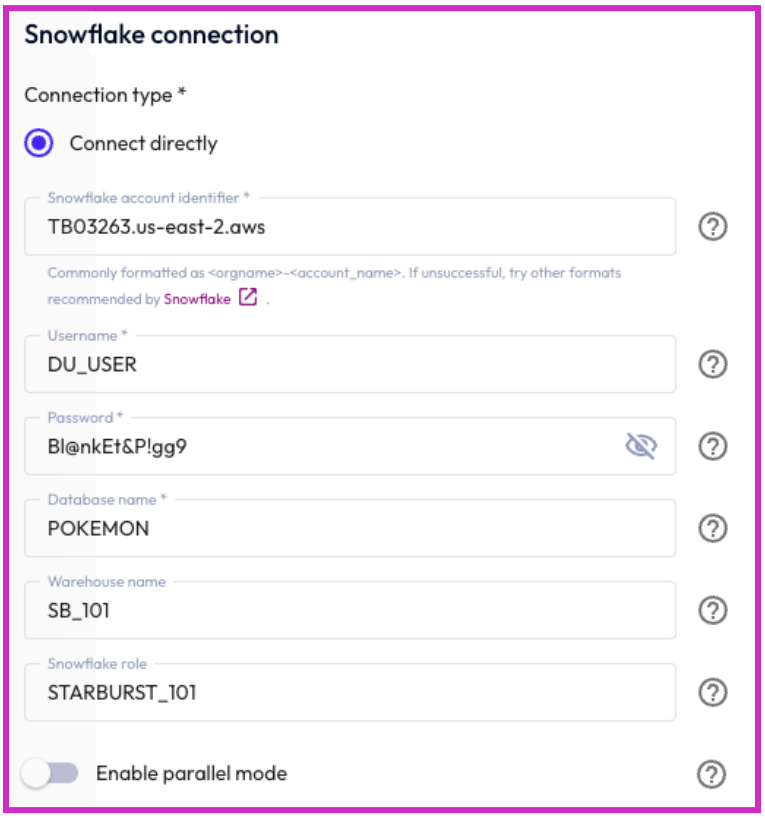
Cloud Provider: AWS



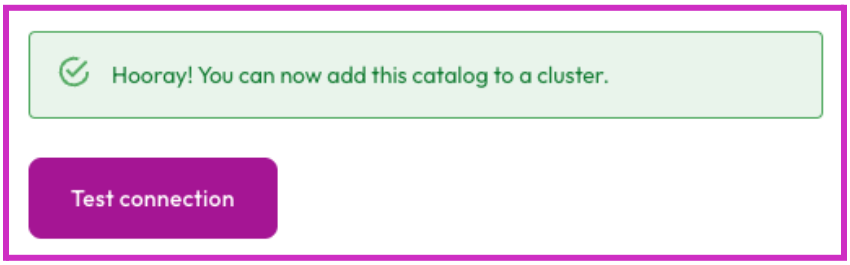
Catalog Name: pokemon_lkp
Description: Lookup table containing pokemon stats



Snowflake account identifier: TB03263.us-east-2.aws
Username: DU_USER
Password: Bl@nkEt&P!gg9
Database name: POKEMON
Warehouse name: SB_101
Snowflake role: STARBURST_101



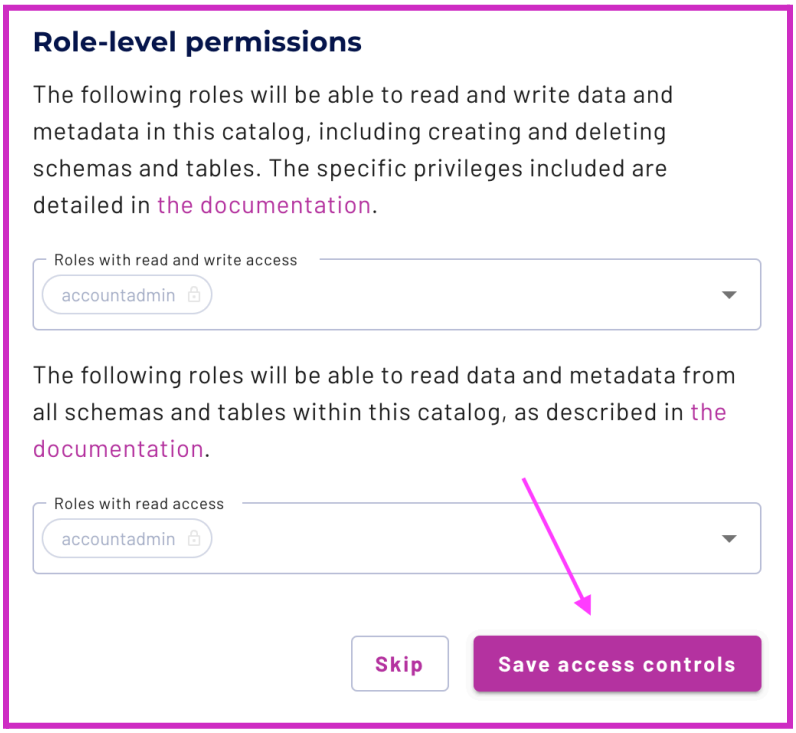
Test the connection to ensure that the setup is correct.



Select **Connect catalog** to save the credentials for your Snowflake catalog.

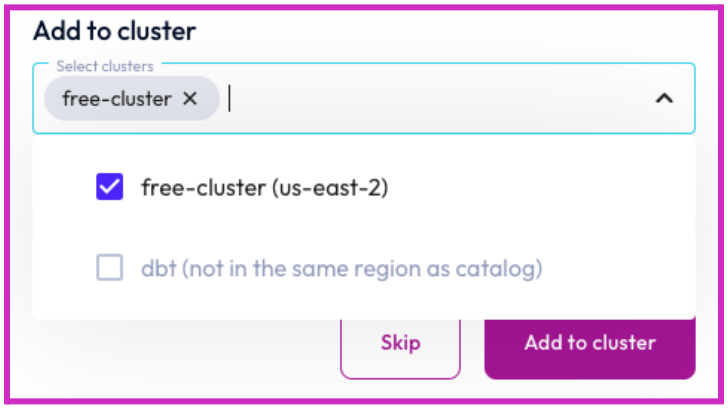
Step 2 - Save access controls

Next, set the default permissions for your catalog by selecting the **Save access controls** button.

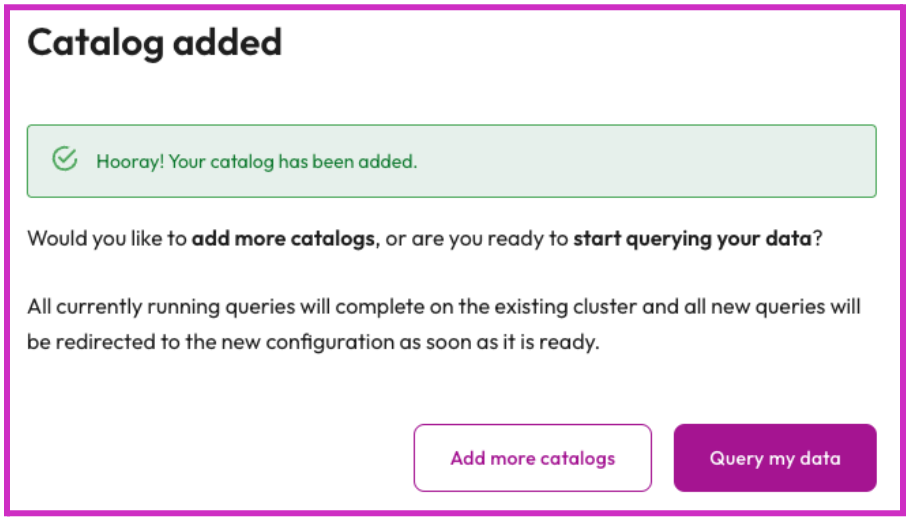


Step 3 - Add to cluster

Select `free-cluster` in the **Select clusters** pulldown and then click on **Add to cluster**.



Click **Query my data** in the **Catalog added** pop-up.



Lab 3: Build within your data lake

Learning objectives

- Demonstrate the process needed to run schema discovery to analyze a root object in an object storage location.
- Show how to use open table formats.
- Demonstrate the steps needed to build a reporting structure in your data lake, and secure your team's access.

Prerequisites

- Lab 1: Introduction and setup
- Lab 2: Connect to data sources

Activities

1. Use schema discovery
2. Discover the lookup data
3. Build the structure layer
4. Build the consume layer
5. Secure access to your consume layer

Part 1: Use schema discovery

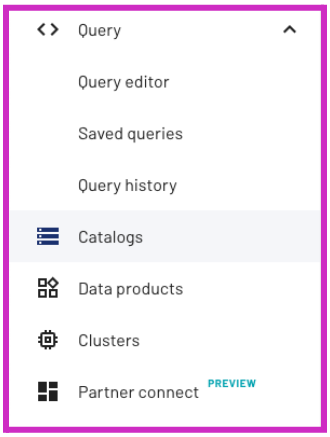
Objective

You're going to begin by utilizing schema discovery to create your schema and table. Schemas control the structure of the data inside them. Luckily for us, Starburst Gravity will take care of the discovery work.

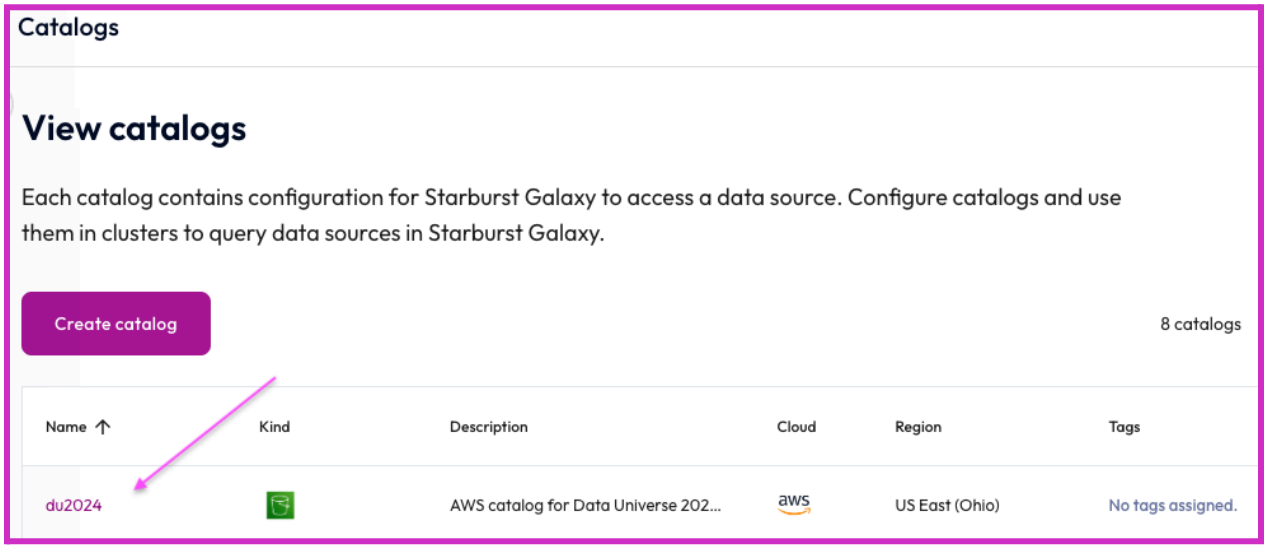
If you ran schema discovery in the SQL-based Workshop #3, you may skip to Part #2.

Step 1 - Navigate to the catalogs page

In the left hand navigation pane, select **Catalogs**.

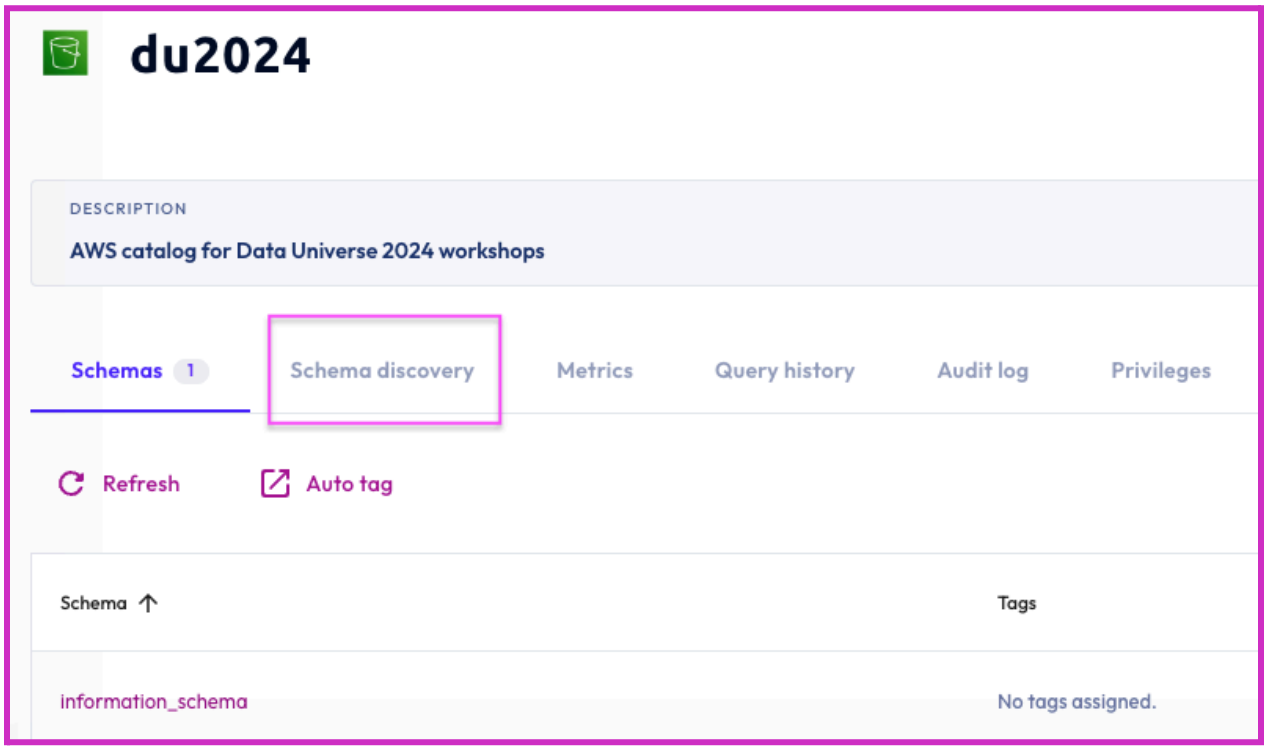


Select the **du2024** catalog to navigate within it.



Step 2 - Run Schema discovery

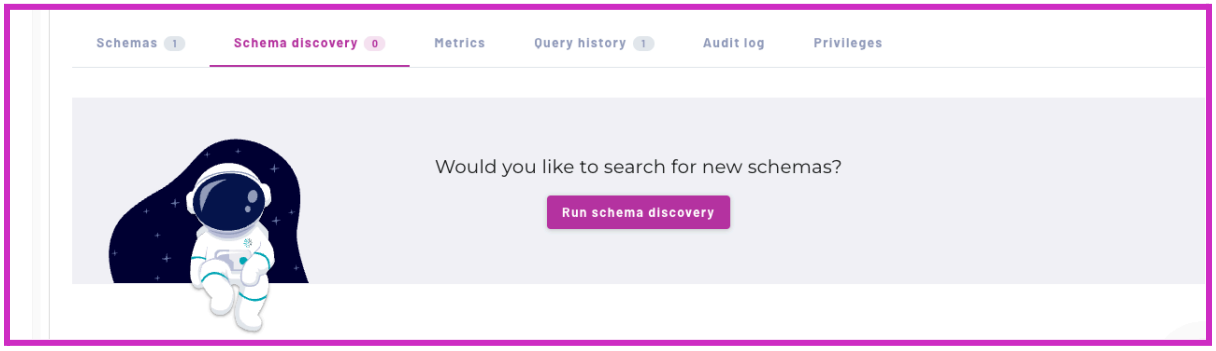
As part of Gravity, you can see all the metrics, schemas, query history, audit log, privileges, and more! Click on the **Schema discovery** tab.



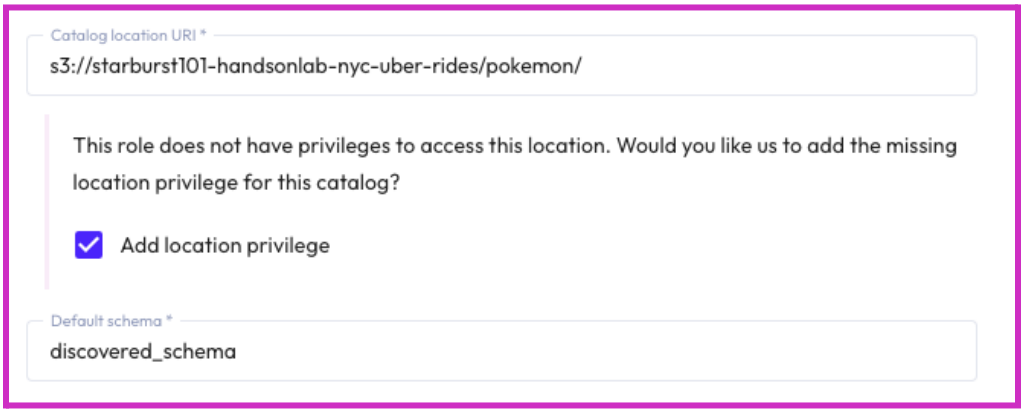
The Schema discovery pane lets you examine the metadata of the specified object storage location. Schema discovery is for catalogs in object storage data sources only.

Use schema discovery to identify and register tables or views that are newly added to a known schema location. For example, a logging process might drop a new log file every hour, rolling over from the previous hour's log file. The purpose of schema discovery is to find the newly added files to make sure Starburst Galaxy knows how to query them.

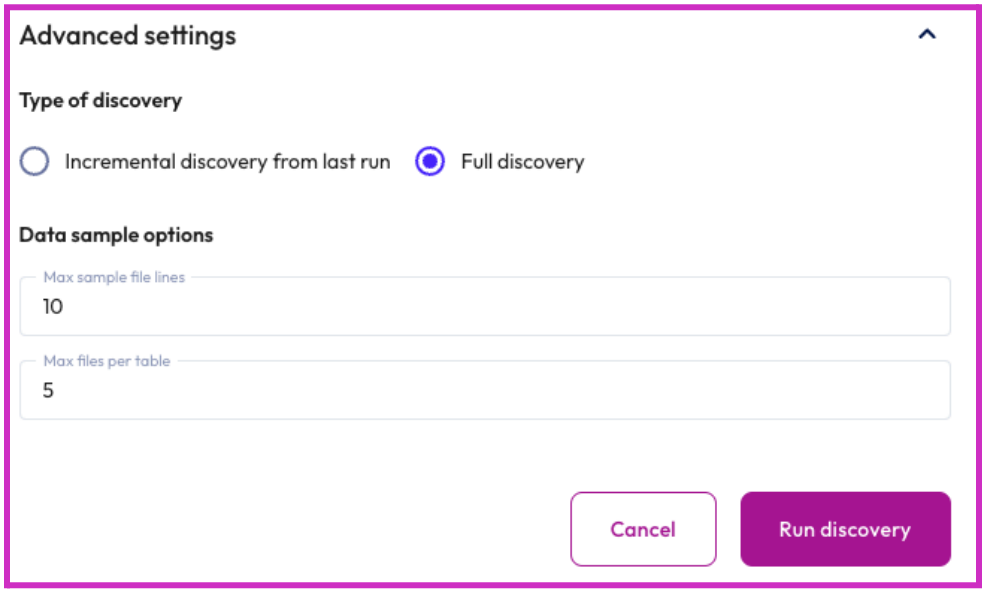
Select **Run schema discovery**.



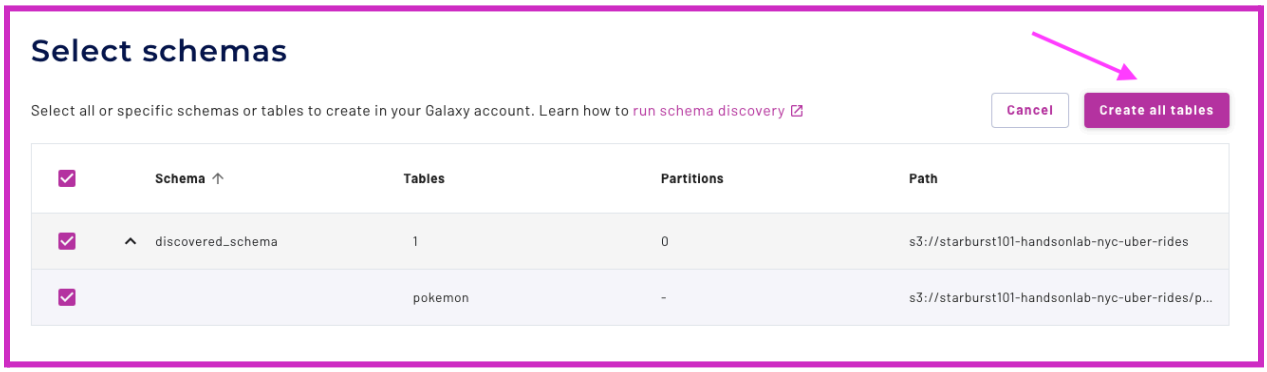
Add the following information:
Catalog location URL:
`s3://starburst101-handsonlab-nyc-uber-rides/pokemon/`
Add location privilege: Leave it checked
Default schema: `discovered_schema`



Toggle the **Advanced settings** arrow down and select **Full discovery** for the **Type of discovery**.



Click **Run discovery.** Starburst will start scanning for you. Then, it will return code to create your desired schema and table. Toggle open `discovered_schema` and then check the top checkbox to select the two below it. Select **Create all tables**.



Schema discovery has done the heavy lifting so you don't have to spend time trying to investigate what columns exist, or bother your AWS administrator to give you details about the file.

Click on the Query text to view the full queries. Your first query created the desired schema, and the second query created a Hive table. Hit **Close**.

Step 3 - Set up the Query editor

Navigate to the **Query editor**. If you already have queries, add a new tab + using the fuschia plus sign. Change the location drop-downs in the top left-hand corner to match the cluster and catalog previously created.

Cluster: free-cluster
Catalog: du2024

Run the following query to validate the table you created using schema discovery.

```
SELECT * from du2024.discovered_schema.pokemon LIMIT 100;
```

Your data sample should look something like the following:

s2_id	s2_token	num	name	lat	lng
-918579452294725...	8085808cc6d	13	Weedle	37.7935915752623	-122.408720633183
-9185794529389707...	8085808b51d	16	Pidgey	37.7947455405929	-122.406419649564
-918579452938970...	8085808b271	41	Zubat	37.794999066064	-122.404384122075
-9185794082713108...	808580f3587	16	Pidgey	37.7956444102582	-122.407127649888
-918579407627065...	808580f4b1d	60	Poliwag	37.7955915257874	-122.406331149188
-9182922218470900...	808fb4e54b3	50	Diglett	37.3011286952679	-122.048453380601

Run a command to view the CREATE TABLE statement:

```
SHOW CREATE TABLE discovered_schema.pokemon;
```

You should return the same code as run with Schema discovery. Notice that the columns are already utilizing proper data types. Also notice that the table format is HIVE. You will update this as you build your reporting structure in your data lake using Great Lakes connectivity.

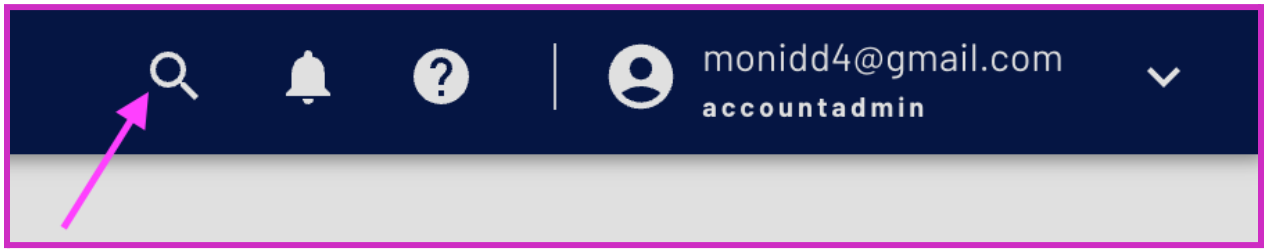
Part 2: Discover the Snowflake data source

Objective

Learn about the Pokedex lookup table stored in Snowflake using global search. Global search lets users find datasets quickly and intuitively. It is a powerful tool that helps keep better track of your data. Use global search to discover the Pokedex data in Snowflake and validate that connection.

Step 1 - Navigate Starburst UI

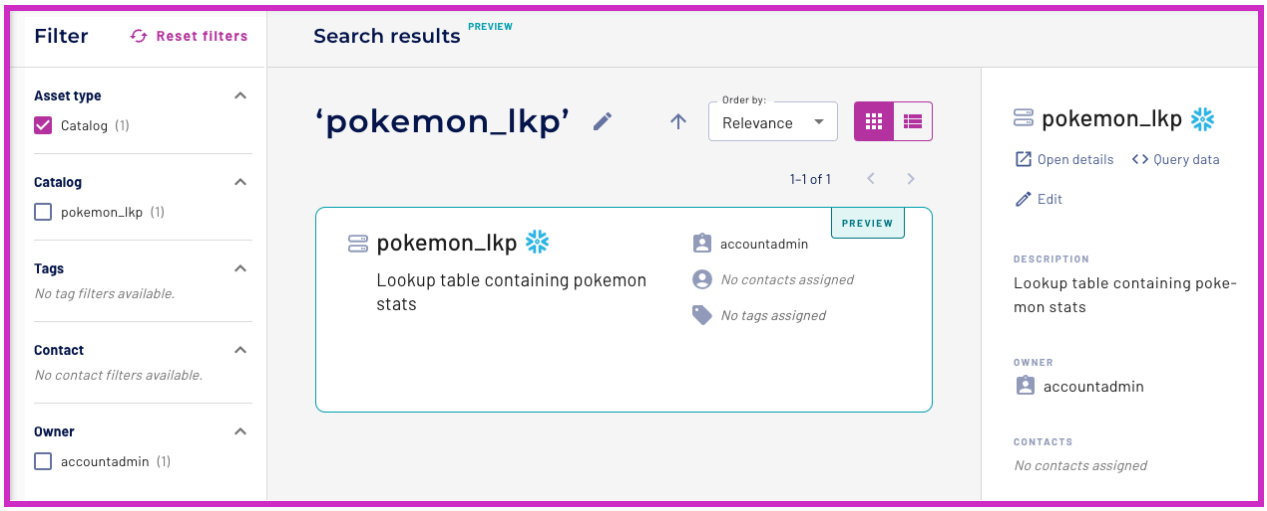
To use global search, select the magnifying glass icon in the upper-right corner.



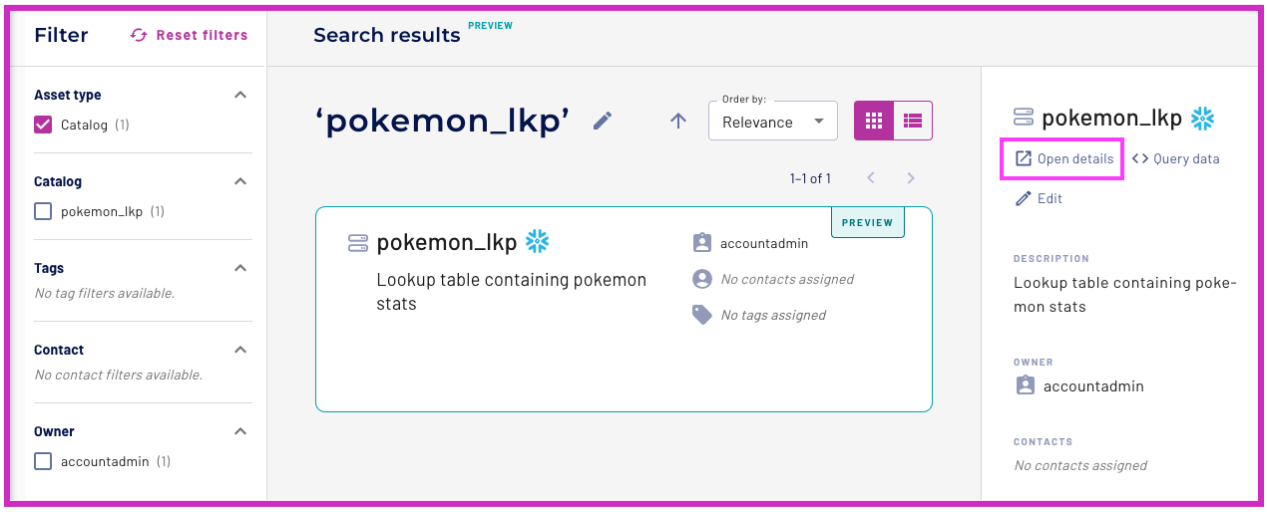
Step 2 - Execute global search

Enter `pokemon_lkp` and select **View all results** at the bottom of the pop-up window.

Starburst Galaxy lets you filter and organize your search results. This is handy when you have a bigger environment with more results.

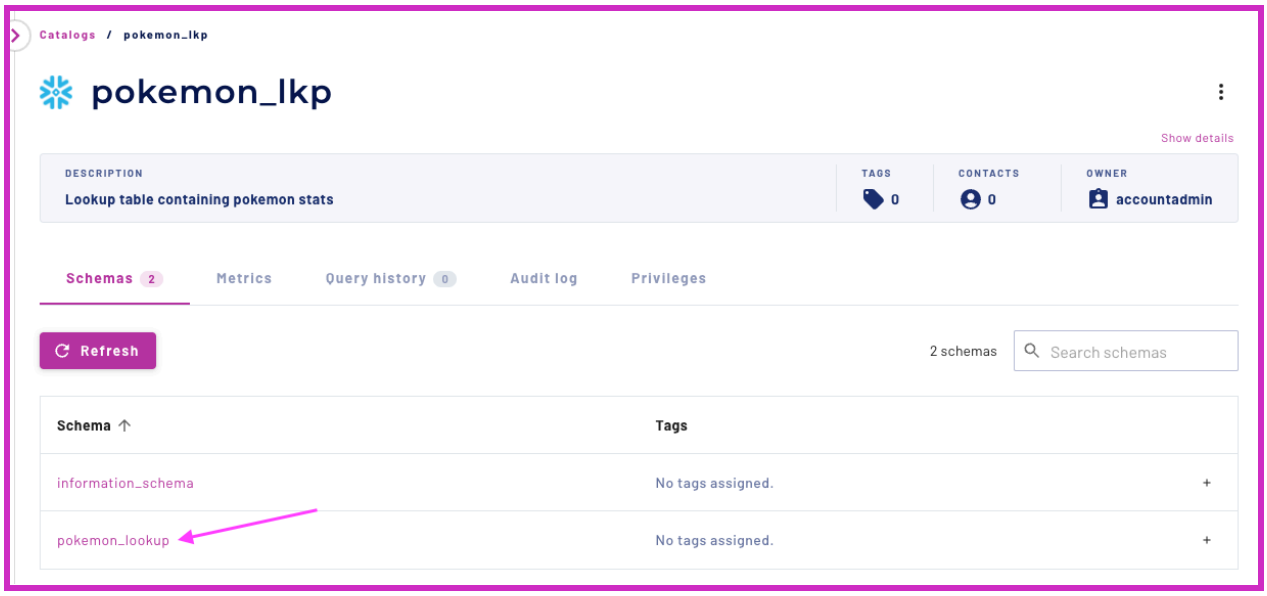


Click on the **Open details** in the top right corner.

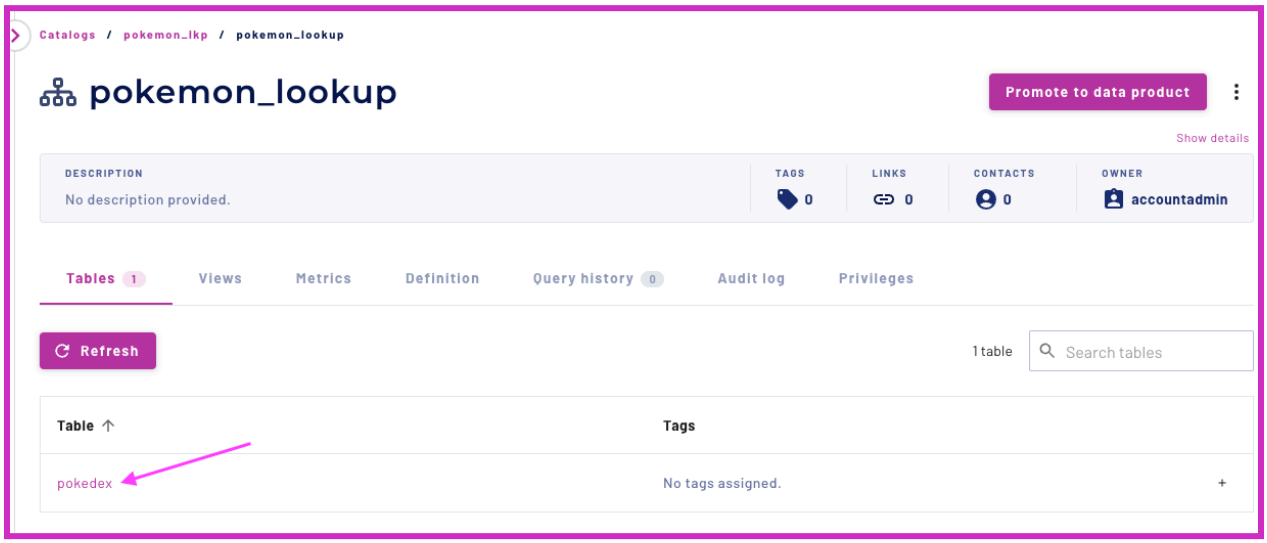


Step 3 - Explore your Snowflake data source

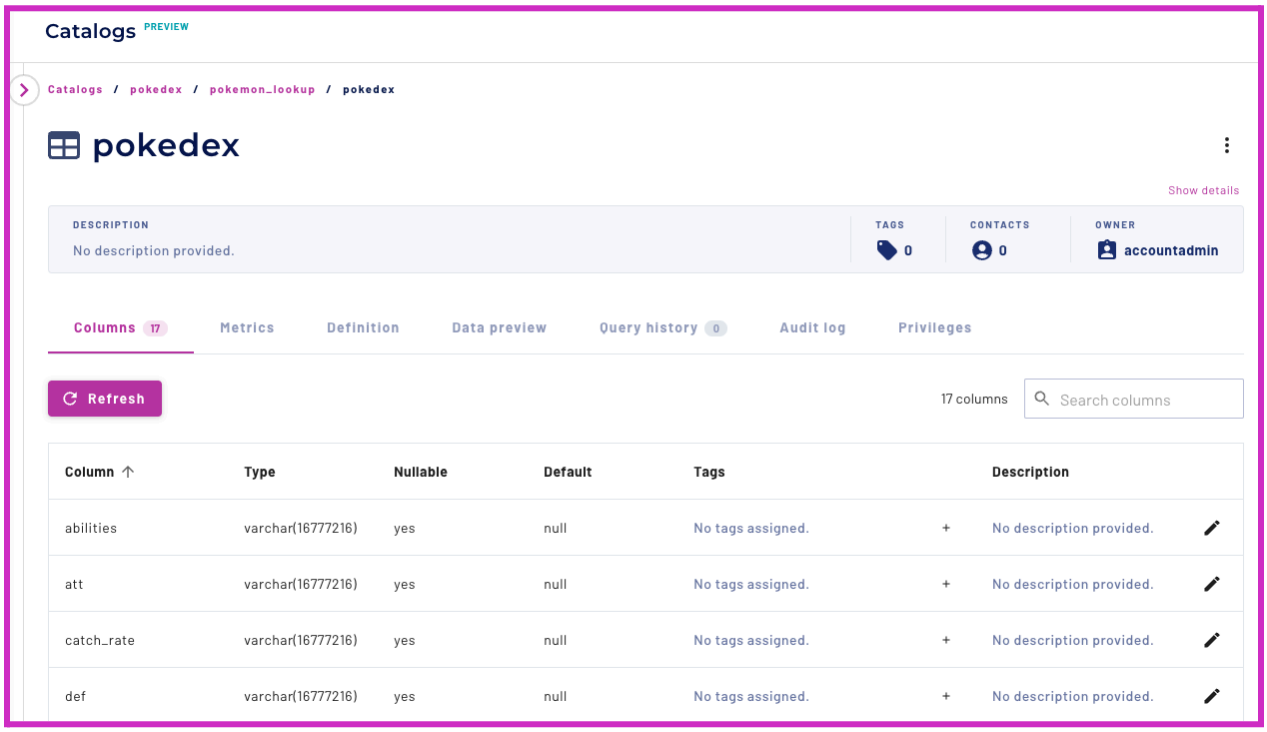
Notice that Starburst Galaxy automatically places you within the catalog page. You can see the catalog has a place to add additional details, as well as shared information regarding **Schemas**, **Metrics**, **Query history**, **Audit log**, and **Privileges.** Click within the `pokemon_lookup` schema to learn more.



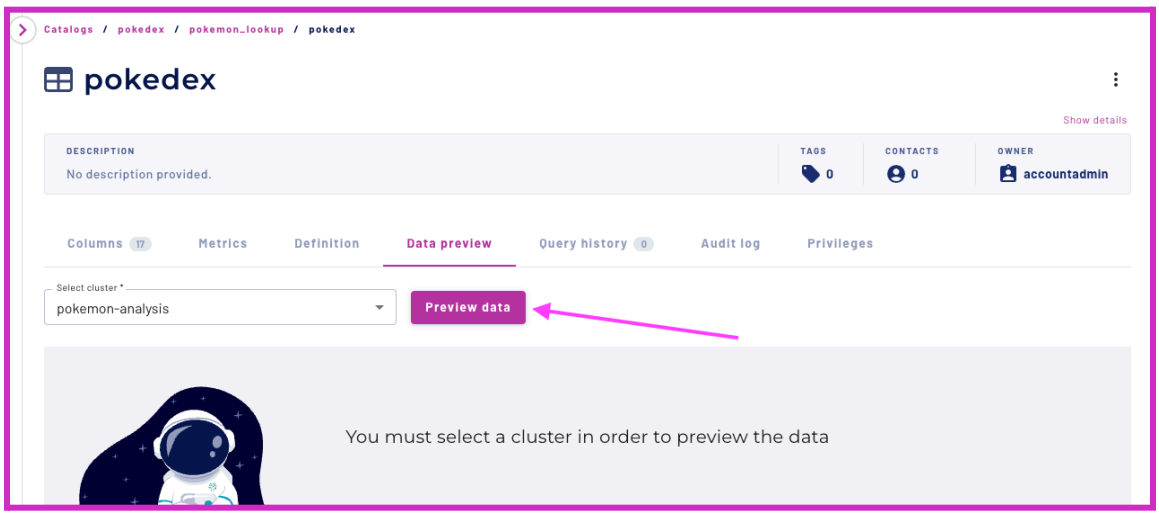
Within the schema, you can see more information available to you. Stay tuned throughout the lab as you will come back and utilize these features of Starburst Gravity. For now, click within the `pokedex` table.



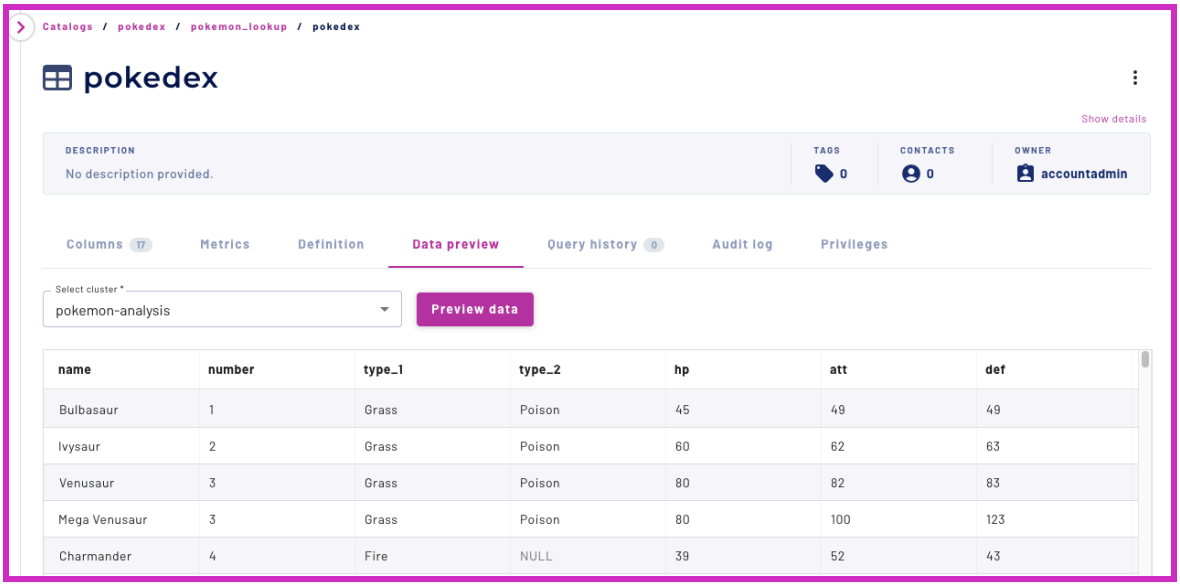
You can see a preview of the **Columns** in the table. You also see all the **Metrics**, the **Definition**, the **Data preview**, the **Query history**, the **Audit log**, and the **Privileges**.



Navigate to the **Data preview** tab. Make sure the `free-cluster` cluster is selected, then hit **Preview data**.

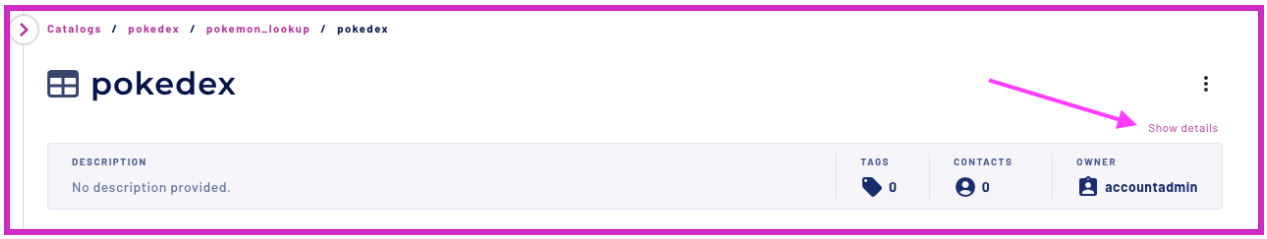


The data is available to be previewed without ever having to run a query. This is handy if you have any data consumers who want access to the data but do not want to use the **Query editor**.



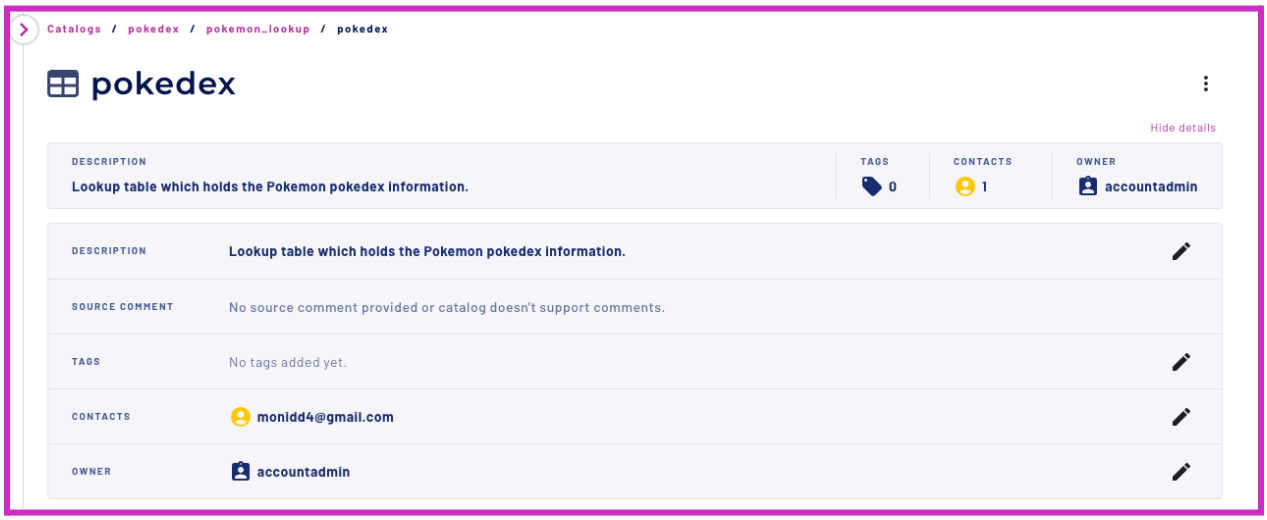
Step 4 - Enter in data definitions

Add the following information to your table so that anyone else who looks at the table has some basic understanding of the data. Hit **Show details**.



Click on the pencil icon on the right of the 2 entries below to enter the information provided below:

> **Description**: Lookup table which holds the Pokemon pokedex information.
> **Contacts:** yourself



Now those reading the dataset for the first time will have more context.

Part 3: Build your reporting structure in S3

Objective

Now it's time to use both your data sources and create a reporting structure in S3.

- **Land layer** - This is the raw data you were ingesting that's landing in S3. Thanks to schema discovery, this layer is already created.
- **Structure layer** - This is the enriched, cleaned, and cleansed data.
- **Consume layer** - This is the data that is ready to be queried and utilized by consumers.

Starburst is special because it allows you to build this reporting structure not just with data that already exists in your data lake, but also with data that exists in other data sources in your orbit - like our Snowflake pokedex data.

Step 1 - Connect to Jupyter

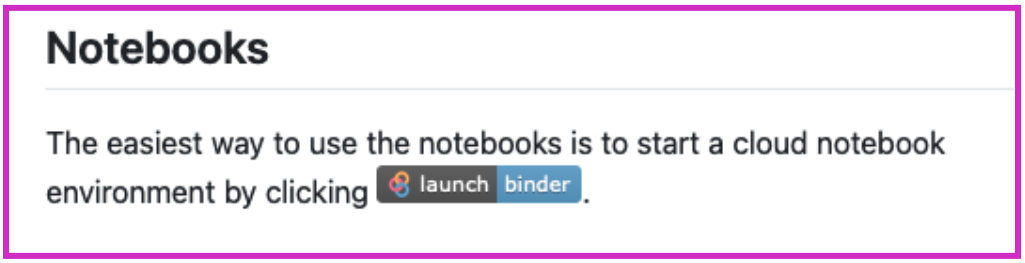
At this point, you will transition from using the Starburst web UI to running code in a Jupyter web-based notebook environment.

You have a few options at this point.

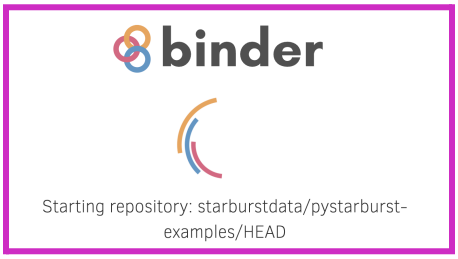
1. If you already have an instance of Jupyter that you can use, then you are all set to continue.
2. You are welcome to follow the Installing Jupyter instructions to set up an environment on your workstation.
3. Or, you can use a temporary Jupyter environment that Starburst can help you create.

If you are using Option 1 or 2 above, you may skip to Step #2.

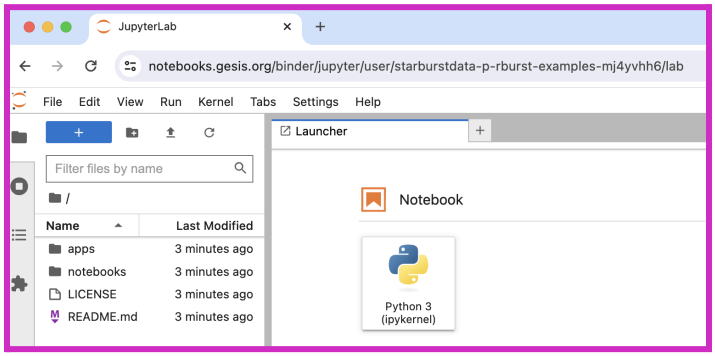
If selecting Option 3 above, visit https://github.com/starburstdata/pystarburst-examples, scroll down to the **Notebooks** section of the README, and click on the button labeled as "**launch binder**".



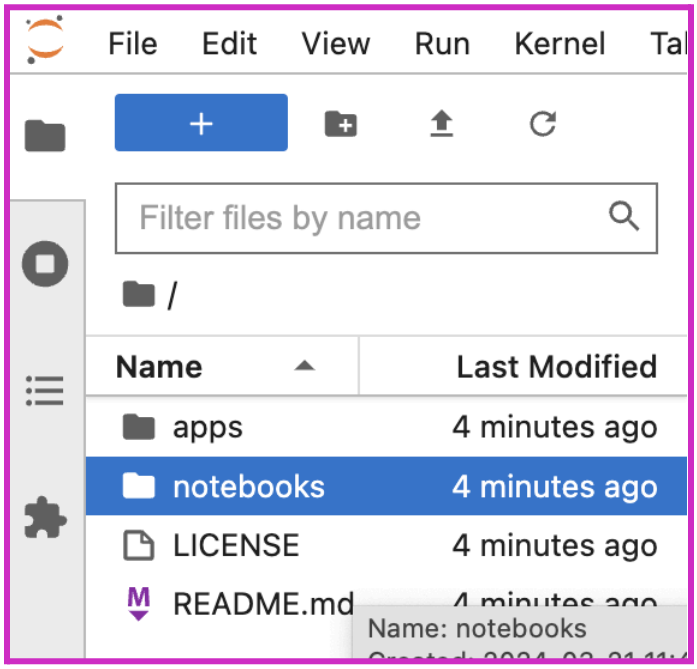
Alternatively, click on https://mybinder.org/v2/gh/starburstdata/pystarburst-examples/HEAD if you cannot access GitHub directly. Regardless of which route you choose, your browser will look like this once launched.



After a short time, a Jupyter web-based notebook system will load.



Now, double-click on **notebooks** in the explorer pane on the left.



Step 2 - Import Jupyter notebook

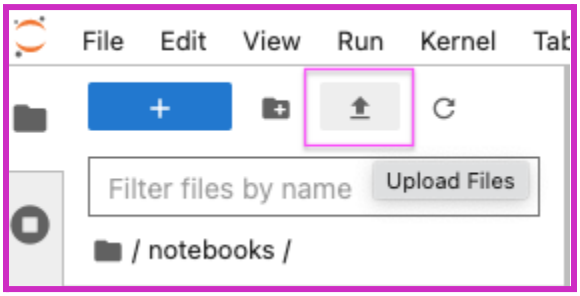
Download `Workshop4.ipynb` to your workstation.

https://github.com/lestermartin/du2024/blob/main/workshop4/Workshop4.ipynb

Import it into your Jupyter environment.

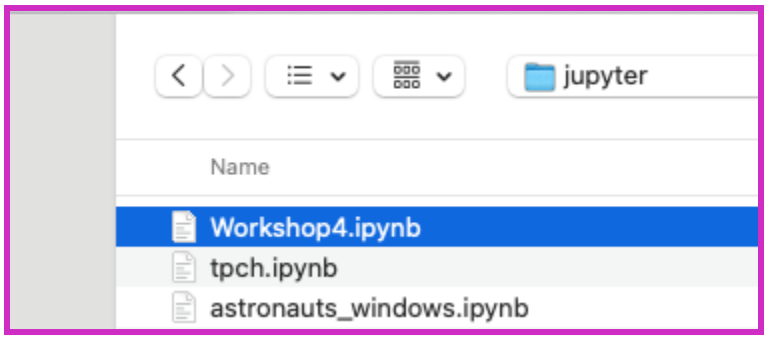
Note: The following screenshots are for the Option 3 setup from the prior Step. If you selected Option 1 or 2 your UI may be slightly different.

Click on the icon with the up-arrow in it.

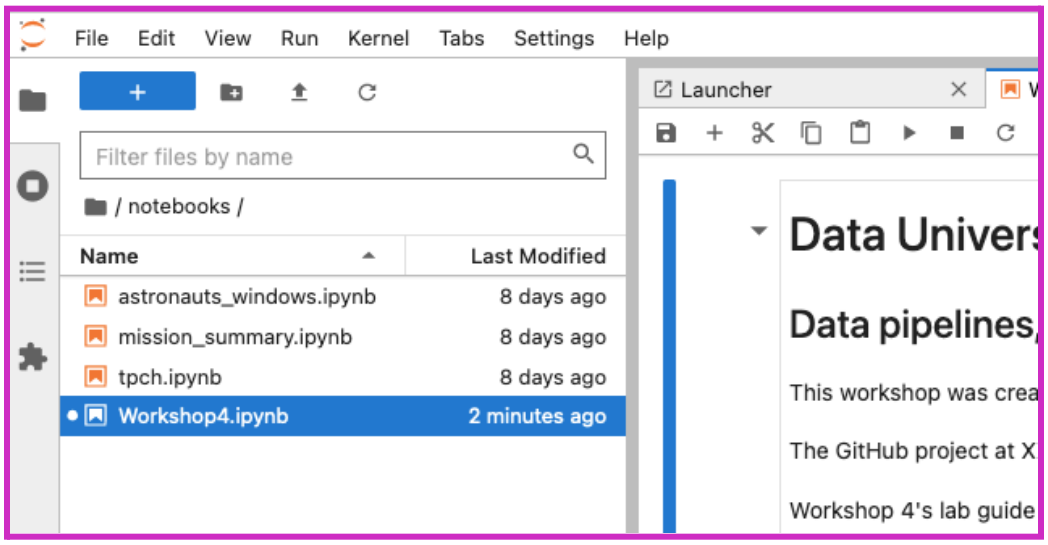


This will open a file-selector window.

Navigate to where you saved the `Workshop4.ipynb` file and choose the file as the example below shows.



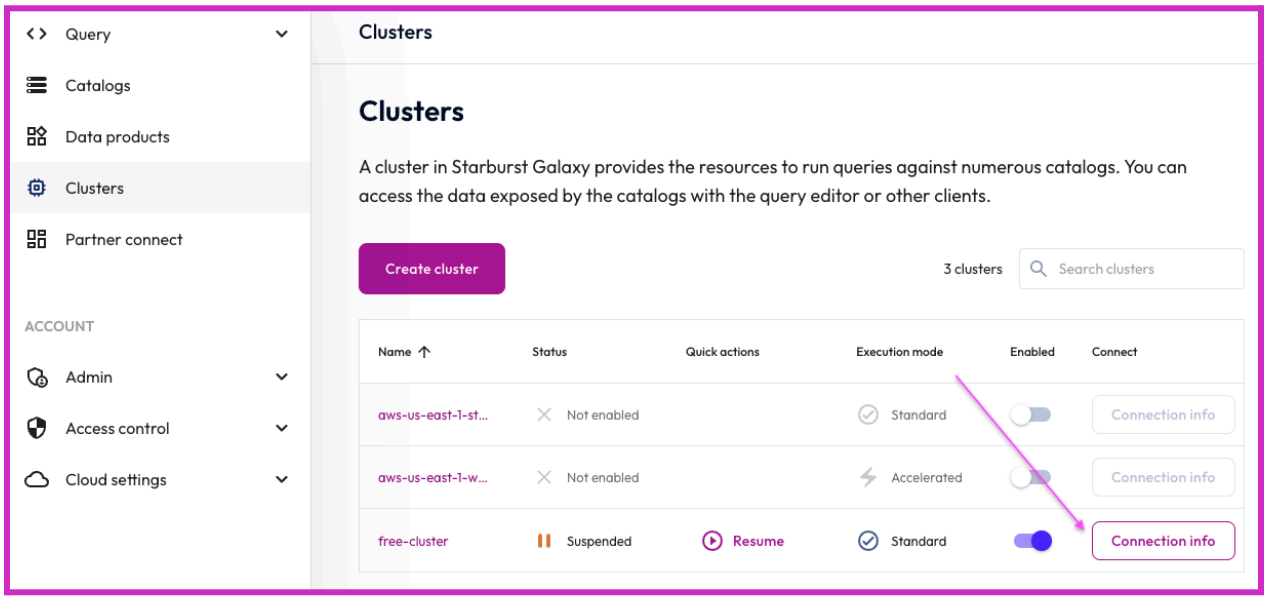
This will add it to the list of files that Jupyter has access to. Double-click on it to open the notebook up in the editor.



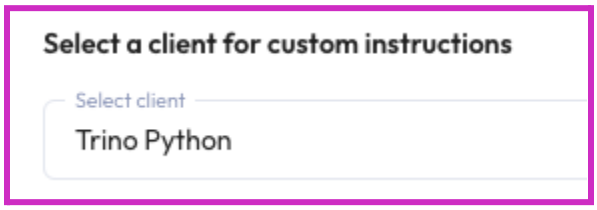
Step 3 - Find connection information

Once you start working in the imported notebook you will need to have the host name and username to connect to Starburst Galaxy.

To find these, select **Clusters** in the left nav and press the **Connection info** button to the far right of the free-cluster entry.



In the pop-up that surfaces, select **Trino Python** in the **Select a client** pulldown.



You can find the **User** and **Host** values at the bottom of the pop-up that you will need in the notebook.

Step 4 - Begin working with PyStarburst

Notice the section in the notebook that looks like the following.

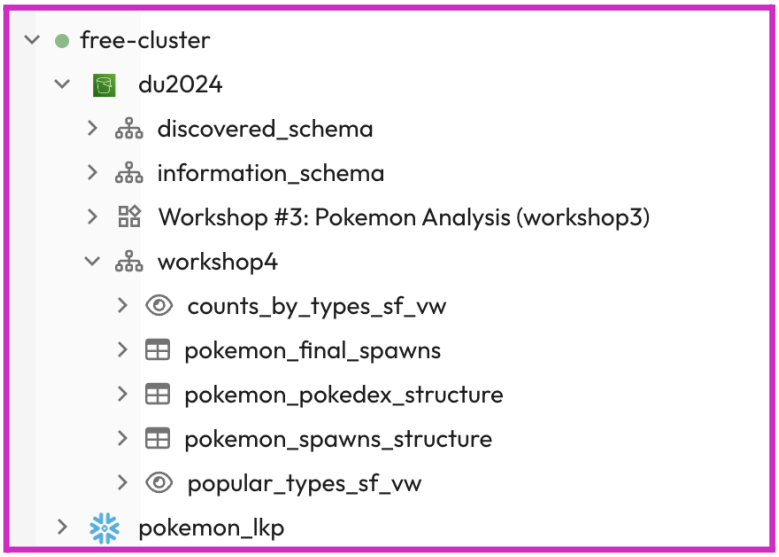
> **Lab 3 > Part 3 > Step 4: Begin working with PyStarburst**
>
> Follow the instructions and run the code in the cells until you read a **RETURN TO LAB GUIDE** message.
>
> Run the following cell to install the PyStarburst library.

As it says, continue working in Jupyter until you see the following **RETURN TO LAB GUIDE** message.

> **RETURN TO LAB GUIDE**
>
> **Resume at Lab 3 > Part 3 > Step 5: Return from working with PyStarburst**

Step 5 - Return from working with PyStarburst

Collapse and expand the `workshop4` schema to see the 3 structure tables and 2 consume views just as you have done in the Jupyter notebook.



Part 4: Secure access to your consume layer

Objective

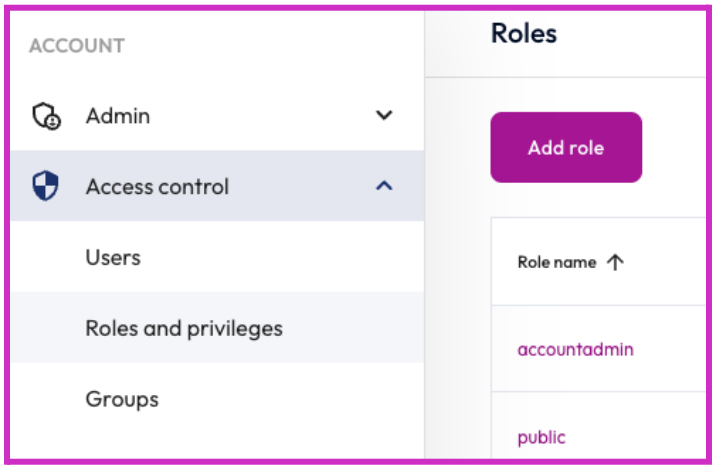
The consume layer has been created. Now it's time to ensure that access to this data is restricted to the appropriate users.

Skip this Part if you completed the same one in Workshop #3 as it is a repeat. If not, continue on and replace any reference to "workshop3" with "workshop4" that still may be present.

Step 1 - Create a marketing role

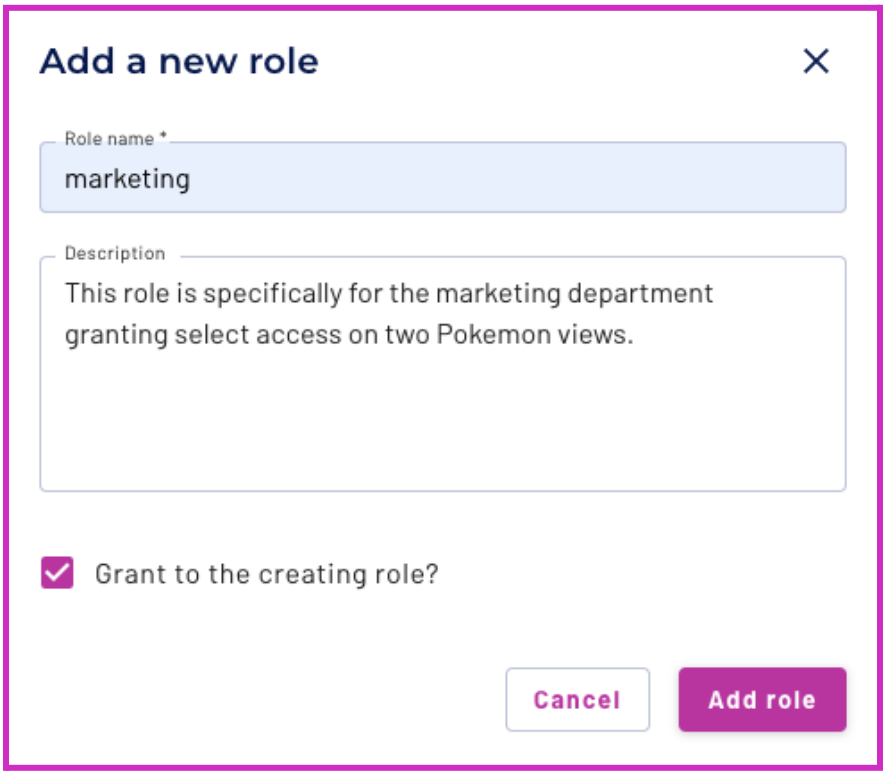
To restrict access to the consume layer, you're going to create a specific role for the marketing department. This will restrict access to the data to team members with the appropriate rights, and restrict their access to the two newly created views.

To do this, navigate to Access control near the bottom of the left navigation and select the **Roles and privileges** submenu item.



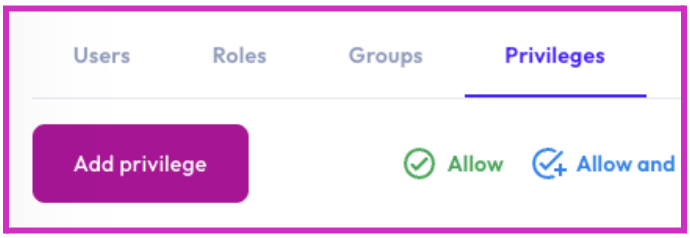
Select **Add role** and enter the following information:

Role name: marketing
Description: This role is specifically for the marketing department granting select access to two aggregated views.
Grant to the creating role? Yes

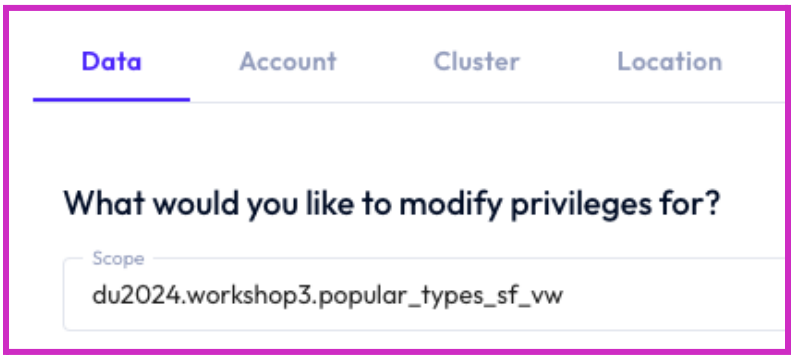


Next, select the newly created marketing role. This allows you to assign proper privileges.

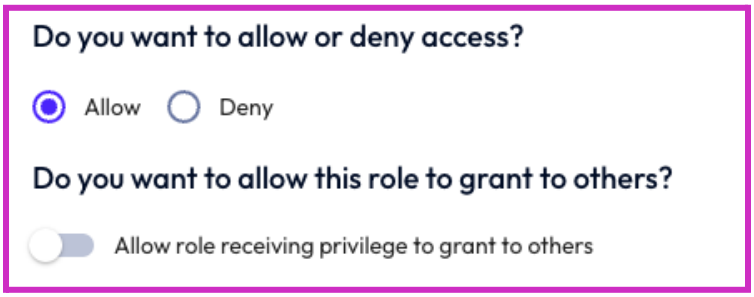
Navigate to the **Privileges** tab. Select **Add privilege**.



With the **Data** tab selected, use the explorer in the **What would you like to modify privileges for?** pulldown to be set to `du2024.workshop4.popular_types_sf_vw`.

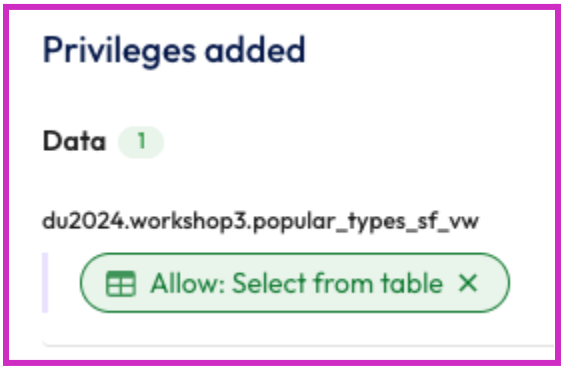


Leave the default values for **allow/deny access** and **allow role to grant to others**.

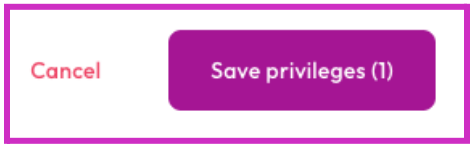


Scroll down if needed and then choose **Select from table** options in the bottom right of the **What can they do?** section.

The **Select from table** option will disappear once you select it. It will appear on the right, under **Privileges added**.



Press the **Save privileges** button below the information just presented.



Repeat this process for `counts_by_types_sf_vw`.

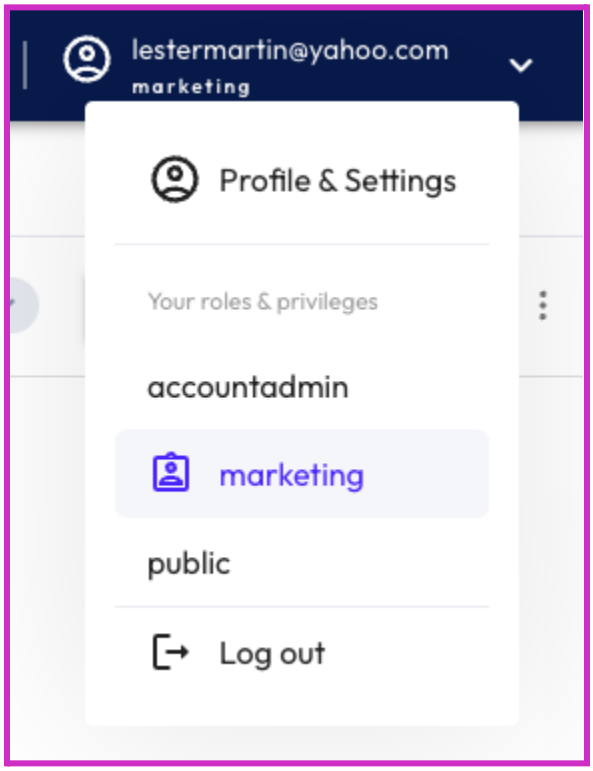
Once back to the **Privileges** tab, expand the **Catalogs** line in the list below.

Toggling `du2024` and then `workshop4` will show you that the **Select from table** privilege has been granted to both views.

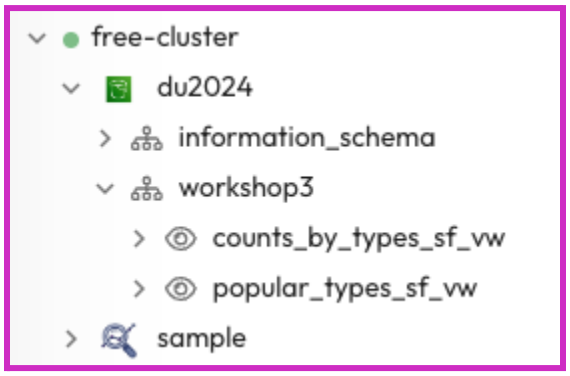
Step 2 - Test the marketing role

Now it's time to test that the new role is working correctly.

Navigate back to the **Query editor** and switch to the **marketing** role in the top right-hand corner.



Notice that the cluster explorer shows only 2 views in the `workshop4` schema. This is exactly what you would expect. Additionally, the `pokemon_lkp` Snowflake catalog is not present.



Run a select statement to validate the newly created role has access to view the tables.

```
SELECT * FROM counts_by_types_sf_vw;
```

Now, try recreating the `popular_types_sf_vw` view using the marketing role.

```
CREATE OR REPLACE VIEW popular_types_sf_vw AS
WITH
popular_types AS (
   SELECT
      type_1,
      name,
      COUNT(*) AS total_appearances,
      RANK() OVER (PARTITION BY type_1 ORDER BY COUNT(name) DESC
      ) AS rank_column
   FROM
      pokemon_final_spawns
   GROUP BY
      type_1,
      name
   ORDER BY
      type_1,
      COUNT(*) DESC
      )
SELECT
   type_1,
   name,
   total_appearances
FROM
   popular_types
WHERE
   rank_column = 1;
```

This will fail because you have only granted the marketing role select permissions.

Access Denied: Cannot create view aws_pokemon.webinar3.popular_types_sf_vw: Role marketing does not have the privilege CREATE_TABLE on the schema aws_pokemon.webinar3

Navigate back to the **accountadmin** role in the upper right-hand corner.

Lab 4: Create data products

Learning objectives

- Demonstrate how to execute a global search.
- Demonstrate how to create a data product.
- Demonstrate how to create tags.

Prerequisites

- Lab 1: Introduction and setup
- Lab 2: Connect to data sources
- Lab 3: Build within your data lake

Activities

1. Execute global search
2. Create a data product
3. Create tags

Skip this Lab if you completed the same one in Workshop #3 as it is a repeat. If not, continue on and replace any reference to "workshop3" with "workshop4" that still may be present.

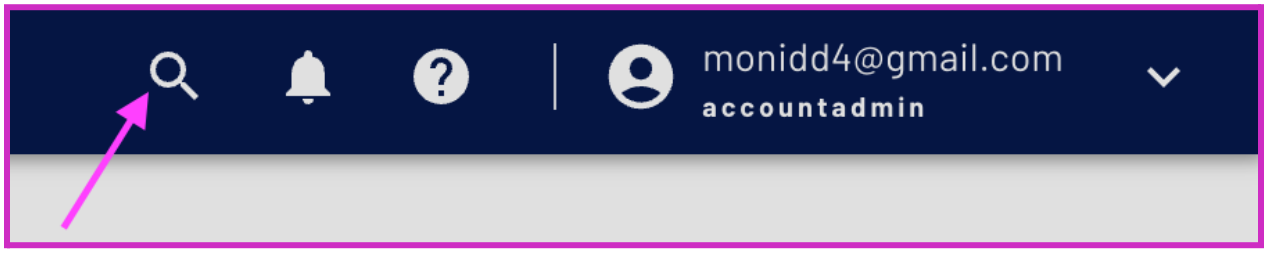
Part 1: Execute global search

Objective

Global search lets users find datasets quickly and intuitively. It is a powerful tool that helps keep better track of your data.

Step 1 - Navigate Starburst UI

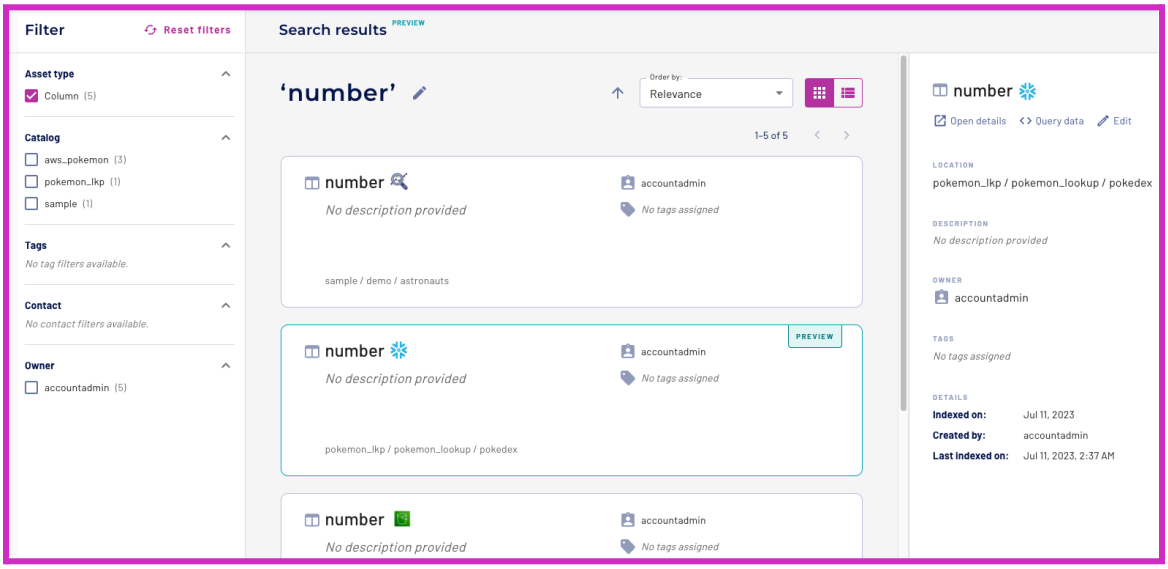
To use global search, select the magnifying glass icon in the upper-right corner.



Step 2 - Execute global search

Enter the word `number` and select **View all results** at the bottom of the pop-up window.

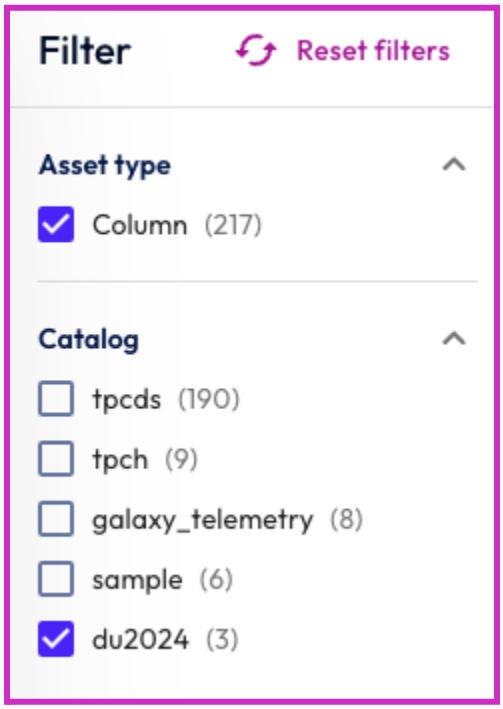
Starburst Galaxy displays multiple instances matching your search criteria drawn from multiple data sources. Some occur in tables and views you created throughout the lab. Others are simply populated through your catalog connection.



Step 3 - Filter global search

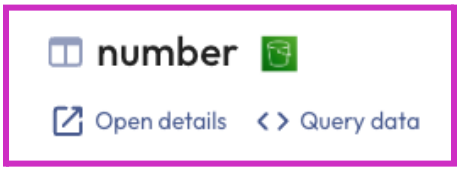
Global search can also be filtered to help refine your search.

Select the Catalog filter for `du2024` on the left to see how different search criteria impact the results.



Step 3 - Navigate to your newly created view

Select **Open details** on the far right for any of the `du2024` results.



Notice that Starburst Galaxy automatically routes you to the **Catalogs** page. Navigate to your `workshop4` schema by clicking on it.

The information for the entire schema is available to you, including **Tables**, **Views**, **Metrics**, **Definition**, **Query history**, and **Privileges**. Navigate through each tab to see the available features.

Part 2: Create a data product

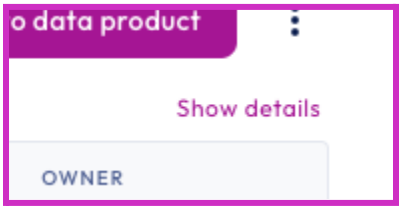
Objective

Now it's time to create a data product using your datasets. Data products curate data in a way that makes it more accessible and useful, and can be shared across teams.

Step 1 - Enter additional information

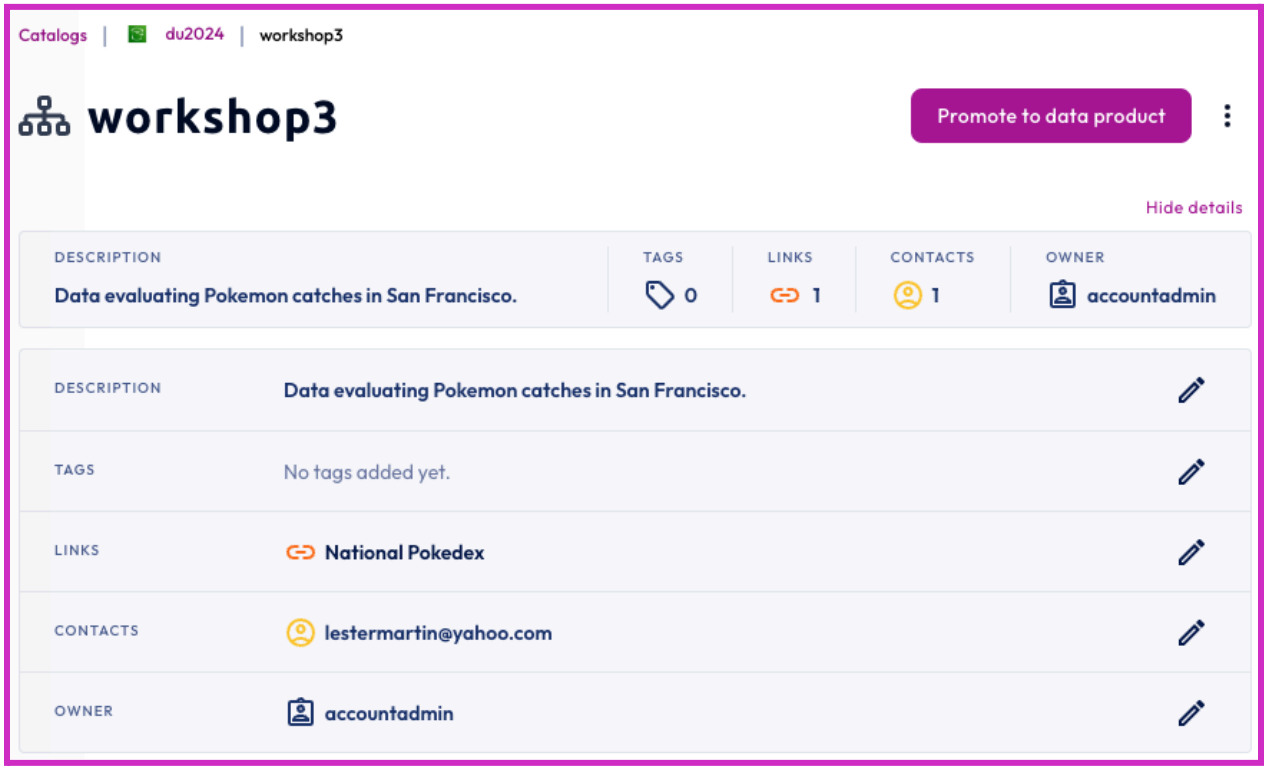
Click on the schema name so you are looking at its metadata.

Before creating a data product, enter some information into your catalog. Select **Show details** on the right-hand side.

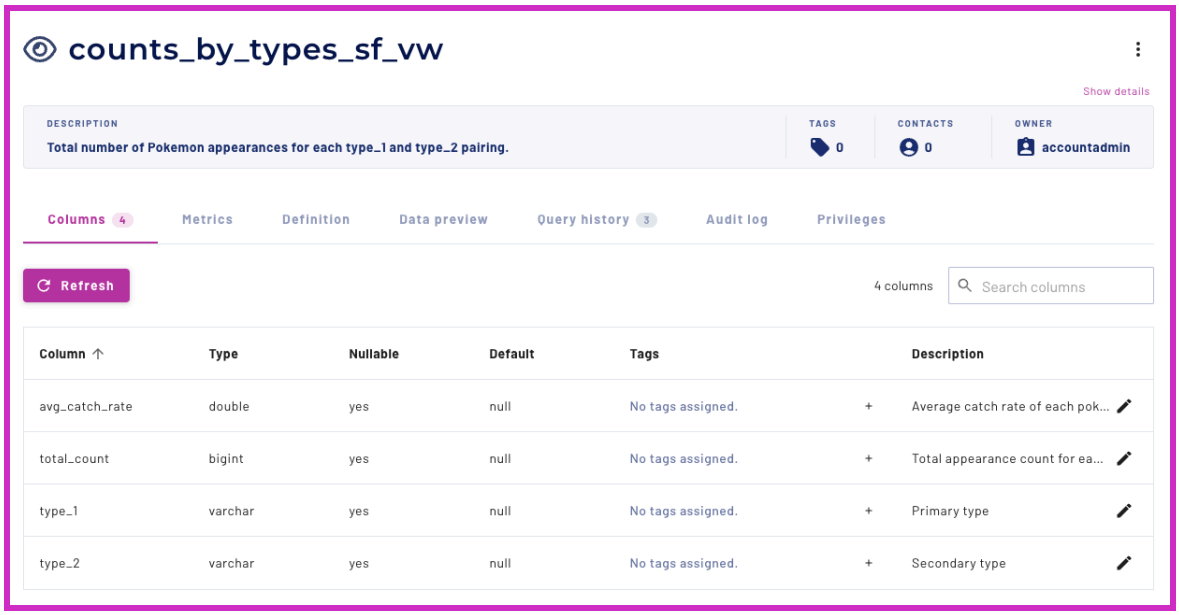


Edit the following information:

Description: Data evaluating Pokemon catches in San Francisco.
Links: Text to display: National Pokedex
Link URL: https://www.serebii.net/pokemon/nationalpokedex.shtml
Contacts: yourname

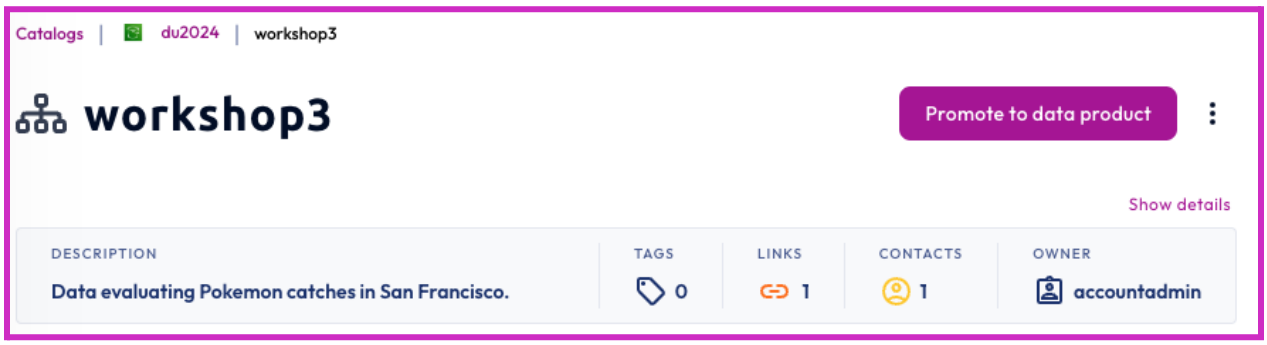


If you have extra time, go through the tables and views created and add meaningful descriptions to each table/view and the columns within them.

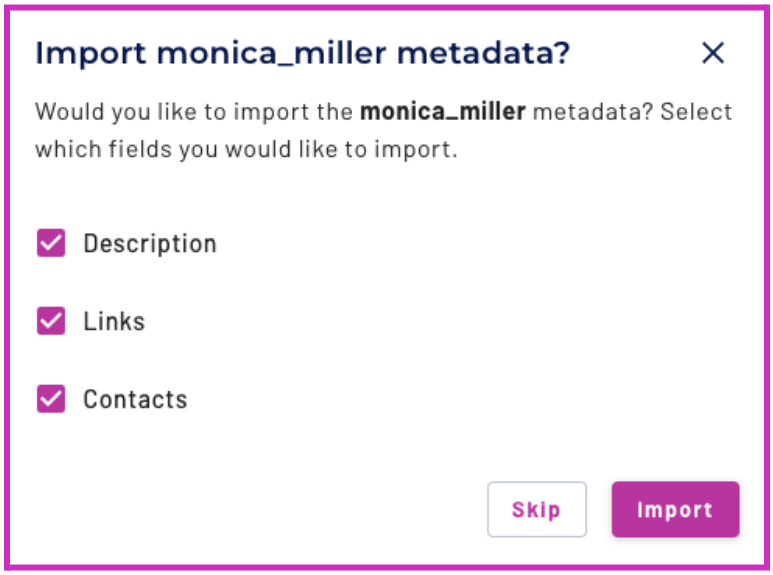


Step 2 - Promote your data product

Navigate back to your schema and select **Promote to data product**.



Import all the information you've already added to the schema. Select **Import**.



Add a descriptive name like `Workshop #4: Pokemon Analysis` for the **Data product name** input field.

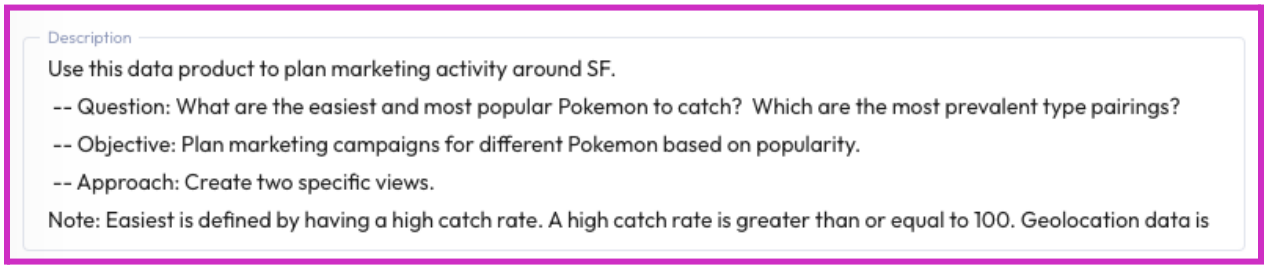
The summary has already been populated based on the information you added to the schema.

Add the following description:

Use this data product to plan marketing activity around SF.

- **Question:** What are the easiest and most popular Pokemon to catch? Which are the most prevalent type pairings?
- **Objective:** Plan marketing campaigns for different Pokemon based on popularity.
- **Approach:** Create two specific views.

Note: Easiest is defined by having a high catch rate. A high catch rate is greater than or equal to 100. Geolocation data is filtered to only be within the San Francisco Bay Area.



Select `free-cluster` as the **Default cluster**.

The **Contacts** and **Supporting information** have been automatically populated from the schema. Select **Promote to data product**.

Congratulations! You have created and promoted your first data product. You were routed to the Data products page where you can view your work.

Part 3: Create tags (Bonus)

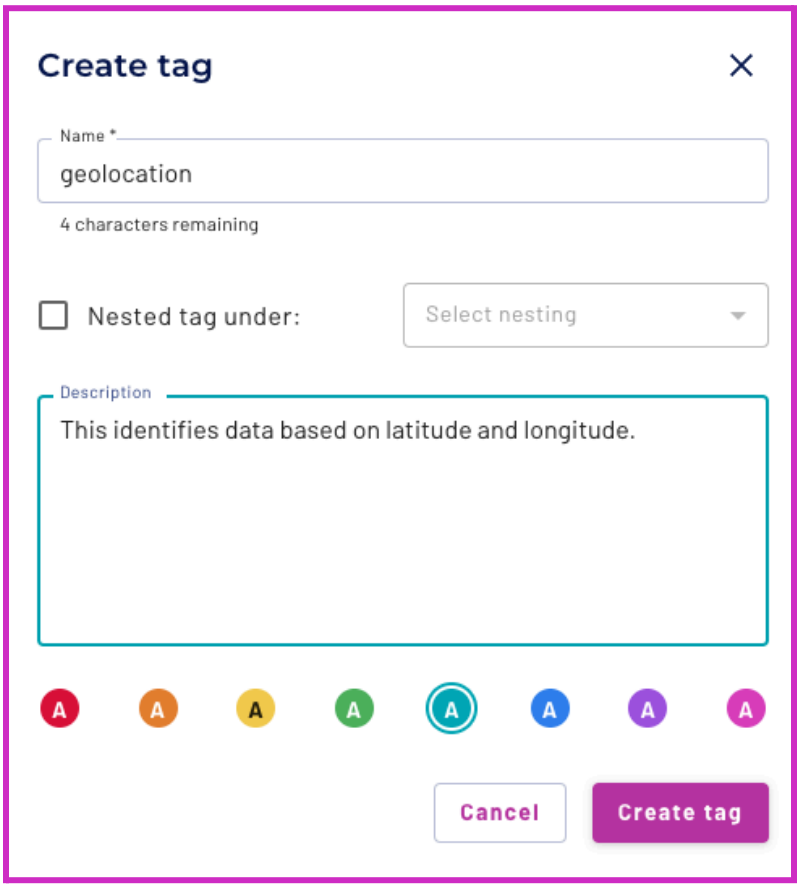
Objective

Last step. It's time to create tags. These can be used to identify the attributes of a dataset so they can be easily searched later. Tagging is flexible and allows you to create the level of granularity that works best for you. Tags can be assigned to the data product, the tables/views within the data product, or the columns within the tables/views.

Step 1 - Create a new tag

Navigate to Admin in the left nav and then select the **Tags** submenu item. Select **Create tag** then fill in the input fields as identified below.

Name: geolocation
Description: This identifies data based on latitude and longitude.
Color: Your choice

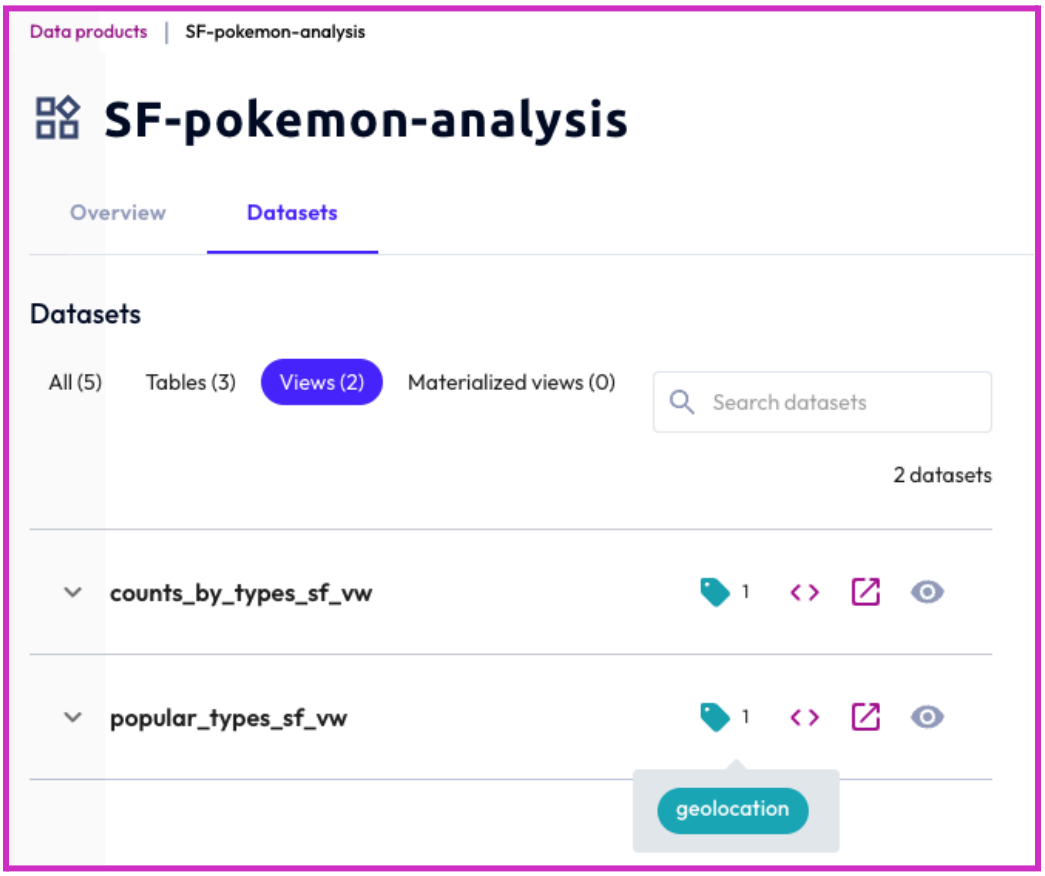


Step 2 - Assign the new tag appropriately

Your mission is to navigate to both views within the data product you created and correctly assign the tag to the created views. Can you figure it out on your own?

Hint: Assign the tags in the **Catalogs** page then verify they are present in the **Data products** page.

Your results should look like following. Ask for help if you need it! :)



If you have more time, add additional information to your data product to make it a more meaningful and curated dataset for your end users.