

# **Lunch and Lab hands-on Lab**

## **Pokemon Lab**

# Pokemon Lunch and Lab Guide

## Lab 1: Introduction and setup

### Learning objectives

- Describe the lab scenario and goals.
- Demonstrate how to up 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: Lab overview

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 lab

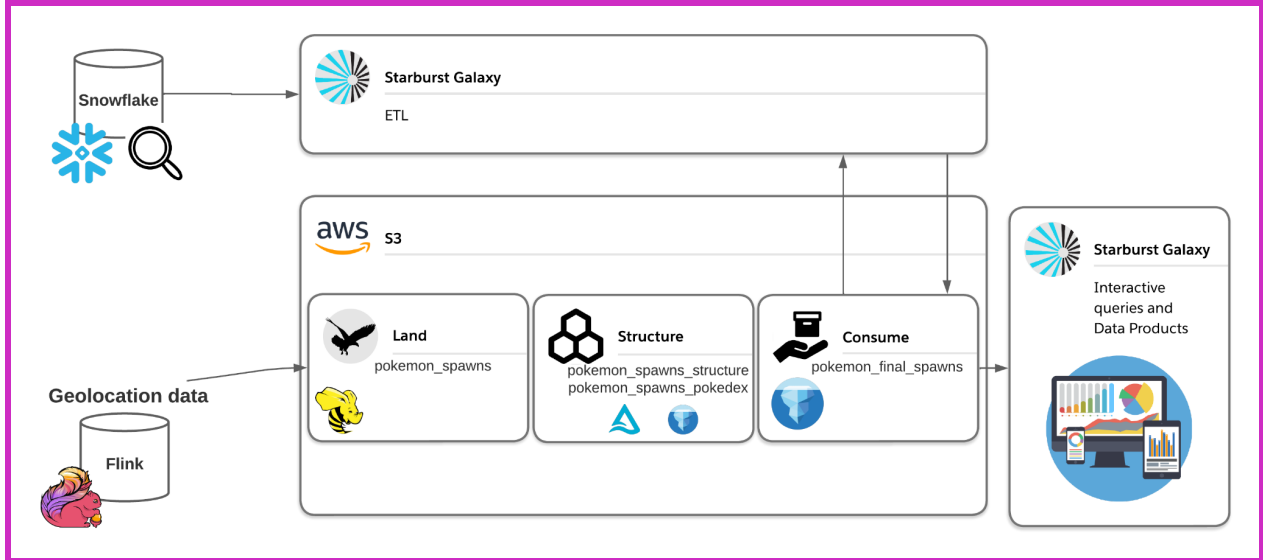
This lab uses Pokemon Go data being ingested into S3, which contains the encounter information of the Pokemon including geolocation data of where the Pokemon spawned, and how long the Pokemon was at that location.

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 performance 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.

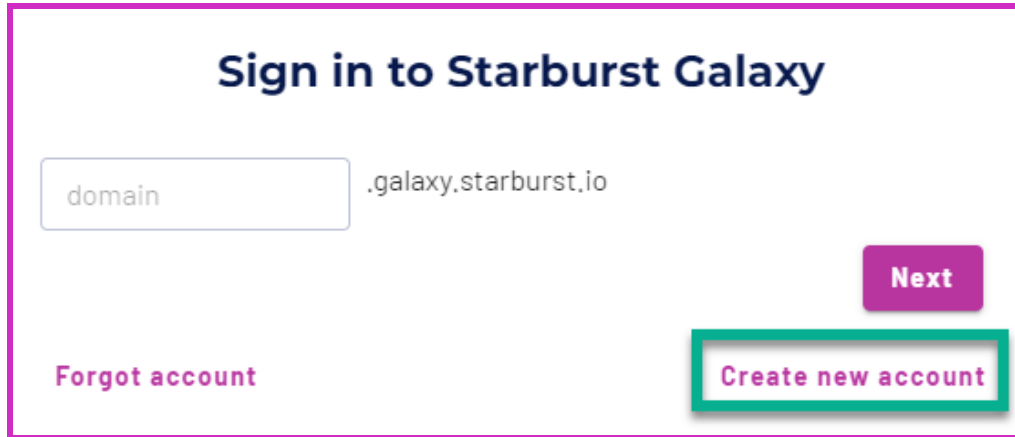
*Note: Easiest is defined by having a high catch rate. A high catch rate is greater than or equal to 100. Also consider that in the structure layer, you filtered out data that did not exist in the San Francisco Bay Area.*

## Part 2: Create a Starburst Galaxy account

Now it's time to create a Starburst Galaxy account. If you already have one, you can use that. If not, follow the steps below.

### Step 1 - Create a Starburst Galaxy domain

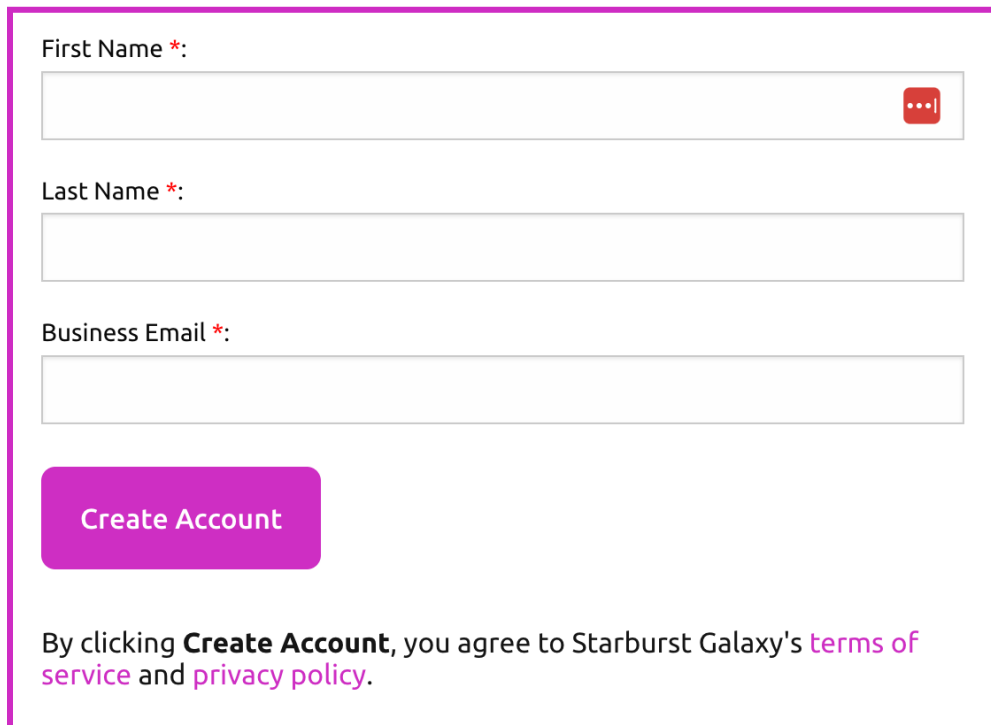
Navigate to the [Starburst Galaxy homepage](#). Click **Create new account**.



The image shows a sign-in page for Starburst Galaxy. At the top, it says "Sign in to Starburst Galaxy". Below this is a text input field containing "domain" and a dropdown menu showing ".galaxy.starburst.io". To the right of the input field is a purple "Next" button. At the bottom left, there is a link for "Forgot account". At the bottom right, there is a purple "Create new account" button, which is highlighted with a green rectangular border.

## Step 2 - Provide your information

Enter your First Name, Last Name, and Email address. Click the **Create Account** button.



The image shows a form for creating a new account. It has three input fields: "First Name \*:", "Last Name \*:", and "Business Email \*:". Each field has a red "X" icon in the top right corner. Below the input fields is a purple "Create Account" button. At the bottom of the form, there is a disclaimer: "By clicking **Create Account**, you agree to Starburst Galaxy's [terms of service](#) and [privacy policy](#)."

## Step 3 - Enter your confirmation code

Go to your email inbox and get the code you were just sent from the subject line of the email.

## Starburst Galaxy

### Confirm your email address

Your confirmation code is below, enter it in your open browser window and we'll help you get signed in.

977-612

Go back to the Starburst Galaxy login, and enter the code. As soon as you type the last digit the verification process will start.



### Enter your email address



### Enter the code sent to your email

We sent a 6-character code to **kyle.payne@starburstdata.com**.  
The code expires shortly, so enter it soon.

9

7

7

—

6

1

2

Verify



### Choose a domain

### Step 4 - Choose a domain name

Select a meaningful domain name according to the guidelines provided. Click the **Create account** button.

A screenshot of a web form for account creation. It features a vertical list of three steps, each with a purple circle icon. The first two steps are marked with a white checkmark, and the third step is marked with the number '3'. The third step is titled 'Choose a domain'. Below the steps, there is a text instruction: 'Select a meaningful domain for your account, this cannot be changed later. Domains can only be 4-32 characters long, must start with a letter, and may only contain letters or numbers.' Below this text is a text input field containing 'kylepaynelab01' and a dropdown menu showing '.galaxy.starburst.io'. A purple 'Create account' button is located at the bottom of the form.

✓ Enter your email address

✓ Enter the code sent to your email

3 Choose a domain

Select a meaningful domain for your account, this cannot be changed later. Domains can only be 4-32 characters long, must start with a letter, and may only contain letters or numbers.

kylepaynelab01 .galaxy.starburst.io

Create account

### Step 5 - Create a password

Enter a memorable and meaningful password. Click the **Create account** button.

A screenshot of a web form titled 'Create password for account'. The form asks the user to finish creating an account for the username 'kyle.payne@starburstdata.com'. It states that the password must be at least 8 characters long. There is a password input field with a green border, containing a series of dots. To the right of the input field is an eye icon. A purple 'Create account' button is located at the bottom right of the form.

## Create password for account

Finish creating account for username:  
**kyle.payne@starburstdata.com**

Password must be at least 8 characters long.

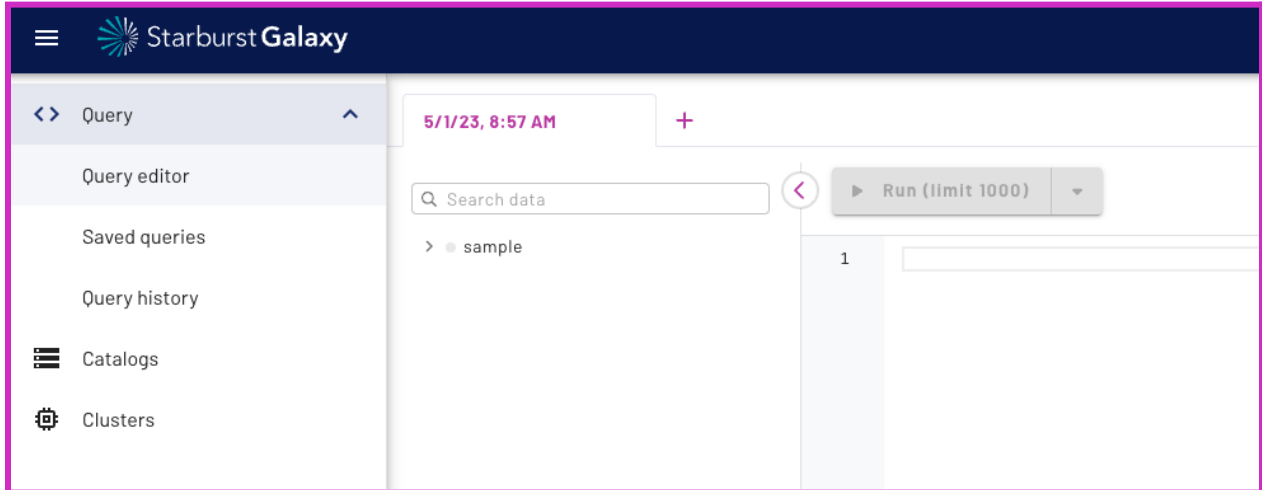
Password \*

.....

Create account

## Step 6 - Read through the Starburst Galaxy information

Read through the introductory onboarding information. This section is complete when you see the Query editor.



## Part 3: Housekeeping

As part of the Galaxy Lunch and Lab, the credentials for Amazon S3 and Snowflake will be available for 1 week. It is critical to understand that after 48 hours, **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](#)

For questions about the lab material, please reach out to [monica.miller@starburstdata.com](mailto:monica.miller@starburstdata.com).

## Lab 2: Connect to data sources

### Learning objectives

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

### Activities

1. Create Amazon S3 catalog
2. Create Snowflake catalog
4. Create a cluster
5. Grant location-based access control

## Part 1: Create Amazon S3 catalog

### Objective

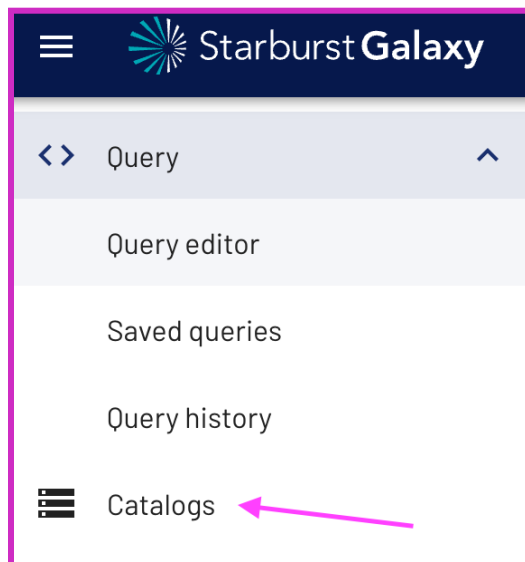
You're going to begin by setting up an AWS S3 catalog in Starburst Galaxy and connect the Pokemon spawns geolocation data.

### 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

Navigate to the **Catalogs** pane on the left-hand side of Starburst Galaxy. Select the **Create catalog** button to create your first catalog.



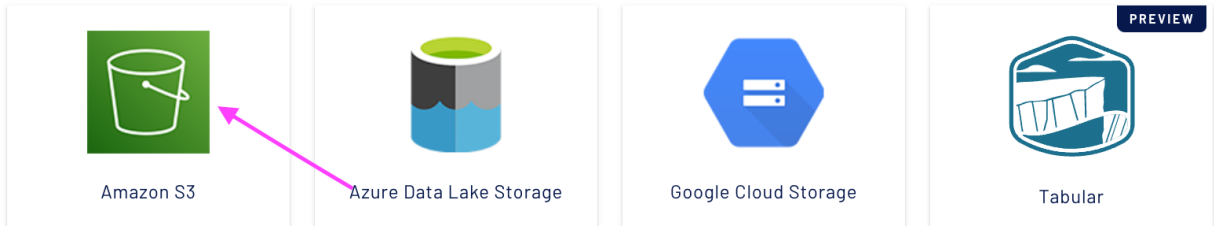


Select **Amazon S3** as your first data source.

## Select a data source

Each catalog contains configuration for Starburst Galaxy to access a data source. Configure catalogs and use them in clusters to query data sources in Starburst Galaxy.

**Note:** **Amazon S3**, **Azure Data Lake Storage**, and **Google Cloud Storage** catalogs support **Iceberg**, **Hive**, **Delta Lake**, and **Hudi (Preview)** tables.



The screenshot shows a 'Select a data source' interface with four options in a row. From left to right: 1. Amazon S3: A green square icon with a white bucket. Below it is the text 'Amazon S3'. 2. Azure Data Lake Storage: A blue and green cylinder icon. Below it is the text 'Azure Data Lake Storage'. 3. Google Cloud Storage: A blue hexagon icon with a white equals sign. Below it is the text 'Google Cloud Storage'. 4. Tabular: A blue icon of a table with a 'PREVIEW' label in a dark blue box above it. Below it is the text 'Tabular'. A pink arrow points from the Amazon S3 icon to the Azure Data Lake Storage icon.

Use the information below to configure your catalog. It will query objects in Amazon S3, specifically the Pokemon spawn data in your data lake. Provide the necessary credentials to authenticate the connection.

**Catalog Name:** aws\_pokemon

**Description:** Pokemon spawns across the country

**Authentication with:** AWS ACCESS KEY (select the radio button AWS access key)

Access Key: AKIAYUW62MUVXIB4HXHN

Secret Key: eNpyg9DjOs3j3Y4Lpk10Mbdp+rRJ35G9FqyzFyaB

**Metastore type:** Starburst Galaxy

**Default S3 bucket name:** starburst101-handsonlab

**Default directory name:** your-name (ex: monica-miller)

Allow creating external tables - YES

Allow writing to external tables - YES

**Default table format:** Iceberg

# Pokemon Lunch and Lab Guide

### Metastore configuration

Configure access to the metastore to provide metadata and mapping information about the objects stored in Amazon S3.

Metastore type \*

AWS Glue  Hive Metastore  Starburst Galaxy

Default S3 bucket name \*  ?

Default directory name \*  ?

Allow creating external tables ?

Allow writing to external tables ?

### Default table format

Select the default table format used for creating new tables. The catalog will be able to read from any type. [Check out our docs](#) to learn more.

Default table format \*

Iceberg  Hive  Delta Lake


Validate the connection by hitting **Test connection**. Your catalog should return the same message indicating that you can now add the catalog.

### Test connection

Validate that the network configuration allows Starburst Galaxy to connect to the data source.

**Detected regions:**

- aws US East (Ohio)

 Hooray! You can now add this catalog to a cluster.

**Test connection**

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**.

#### Role-level permissions

The following roles will be able to read and write data and metadata in this catalog, including creating and deleting schemas and tables. The specific privileges included are detailed in [the documentation](#).

Roles with read and write access

accountadmin

The following roles will be able to read data and metadata from all schemas and tables within this catalog, as described in [the documentation](#).

Roles with read access

accountadmin

**Skip** **Save access controls**

### Step 4 - Skip Add to cluster

Finally, you will be asked if you want to add your newly created catalog to a cluster. Select **Skip**. You will create a cluster later in the lab exercise.

#### Add to cluster

Attach your **aws\_pokemon** catalog to a cluster in order to query your data. You may add it to an existing cluster in the same region, or create a new cluster.

#### Add to cluster

Select clusters

+ Create a new cluster

**Skip** Add to cluster

## Part 2: Create Snowflake catalog

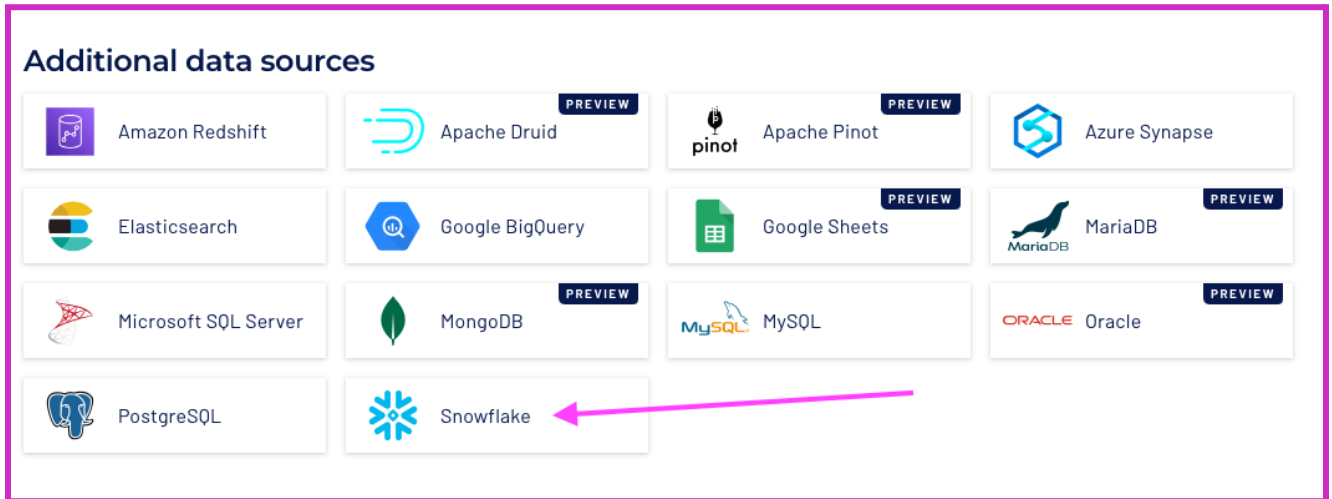
### Objective

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.

### Step 1 - Create Snowflake catalog

From the catalog page, select the **Create catalog** button to create your 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:** GALAXY\_LAB

**Password:** m45n!wiTH8B&ghp

**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.

### Role-level permissions

The following roles will be able to read and write data and metadata in this catalog, including creating and deleting schemas and tables. The specific privileges included are detailed in [the documentation](#).

Roles with read and write access

accountadmin

The following roles will be able to read data and metadata from all schemas and tables within this catalog, as described in [the documentation](#).

Roles with read access

accountadmin

**Skip** **Save access controls**

## Step 3 - Skip Add to cluster

Finally, you will be asked if you want to add your newly created catalog to a cluster. This is not needed, so select **Skip**. You will create a cluster later in this lab.

### Add to cluster

Attach your **taxi\_zone\_lookup** catalog to a cluster in order to query your data. You may add it to an existing cluster in the same region, or create a new cluster.

#### Add to cluster

Select clusters

+ Create a new cluster

**Skip** Add to cluster

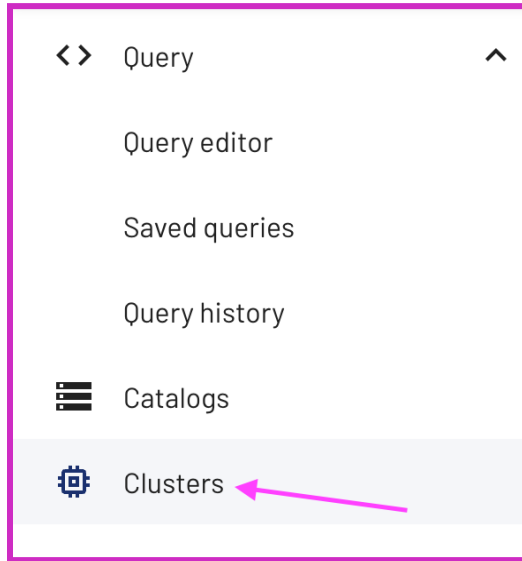
## Part 3: Create a cluster

### Objective

Now it's time to create a cluster to help you execute queries against your catalogs. To federate data from multiple sources, you need to add catalogs to the same cluster.

### Step 1 - Navigate to the Clusters pane

To begin, navigate to the Clusters pane on the left-hand side of your Galaxy browser window.



### Step 2 - Create cluster

Select **Create cluster**. Populate the cluster with the information below. Use the check boxes to select the catalogs. When complete, select **Create cluster**.

**Cluster name:** pokemon-analysis

**Catalogs:** aws\_pokemon & pokemon\_lkp

**Cloud provider region:** US East (Ohio)

**Execution mode:** Standard

**Cluster type:** Free

**Idle Shutdown time:** 30 minutes

## Pokemon Lunch and Lab Guide

### Create a new cluster ✕

Cluster name \*  
pokemon-analysis

Must start with a letter and only use lowercase letters (a-z), numbers (0-9), and hyphens (-)

Catalogs  
2 catalogs selected

Cloud provider region \*  
aws US East (Ohio) ✕

### Cluster type

Learn more about [execution modes](#) or [cluster sizes](#)

Execution mode \*  
 Standard

Cluster size \*  
Free

Auto suspend  
30 Minutes

The maximum idle time before a cluster is automatically suspended.

### Advanced settings ∨

Cancel Create cluster

You will now see your newly created cluster, similar to the image below.

## Clusters

A cluster in Starburst Galaxy provides the resources to run queries against numerous catalogs. You can access the data exposed by the catalogs with the query editor or other clients.

Create cluster 2 clusters

Name ↑	Status	Quick actions	Execution mode	Enabled	Connect
<a href="#">pokemon-analysis</a>	✓ Running		<input checked="" type="radio"/> Standard	<input checked="" type="checkbox"/>	<span>Connection Info</span> <span>⋮</span>
<a href="#">sample</a>	✕ Not enabled		<input checked="" type="radio"/> Standard	<input type="checkbox"/>	<span>Connection Info</span> <span>⋮</span>

## Part 4: Grant location-based access control

### Objective

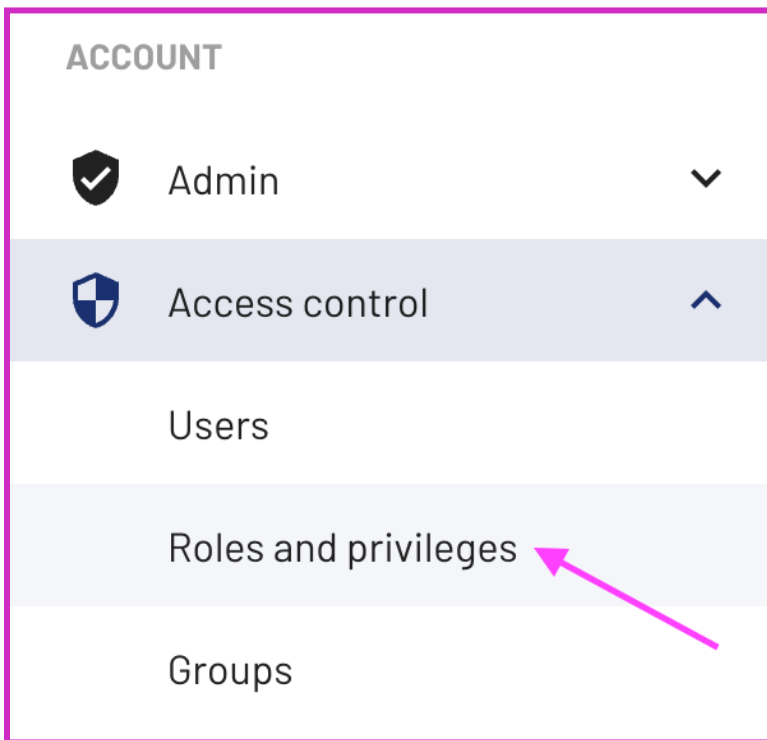
Starburst Galaxy provides built-in access control using customizable levels of granularity. You can control this feature using both role-based access control (RBAC) and attribute-based access control (ABAC).

In this section, you will apply the appropriate permissions to run queries on the S3 object storage location so that you can execute queries within the data lake.

Apply the permissions necessary to create the land table in your data lake.

### Step 1 - Add location privileges for the data lake

Navigate to the **Roles and privileges** section under **Access control** on the left-hand side of Starburst Galaxy.

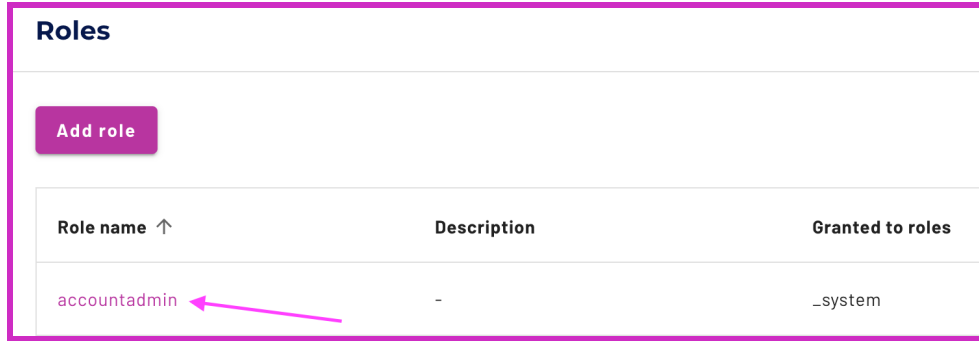


Starburst Galaxy's built-in access control allows you to define multiple users, roles, groups, privileges, and policies. These access options encourage businesses to create security policies that make sense for their organization.

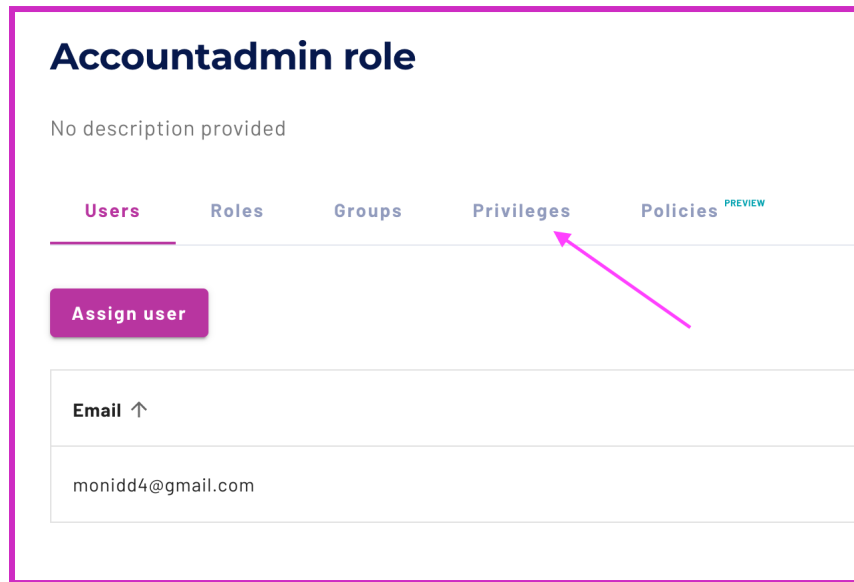
Click the **accountadmin** role link to add additional privileges.



## Pokemon Lunch and Lab Guide



Navigate to the **Privileges** tab to add location-based access.



Select the **Add privilege** button. Notice the various levels of granularity available, including: account, catalog, cluster, schema, table, column, location, and function.

Select the **Location** radio button.

Add the storage location listed below. This provides access to an external location in S3 and allows you to create an external table from this location.

Storage location name:

`s3://starburst101-handsonlab-nyc-uber-rides/pokemon/*`

Check the **Create schema and table in location** box and click **Add privileges**.

## Assign to: accountadmin

Allow this role to access all or specific catalogs, tables, or clusters within your organization. Refer to our detailed [documentation on assigning privileges to roles](#) .

### What would you like to modify privileges for?

- Account  Catalog  Cluster  Schema  Table  Column  Location
- Function

#### Enter a storage location.

A storage location starts with **s3://**, **gs://**, or **abfs://** and ends with **/\***

Storage location name \*

- Create schema and table in location

Cancel

Add privileges

# 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

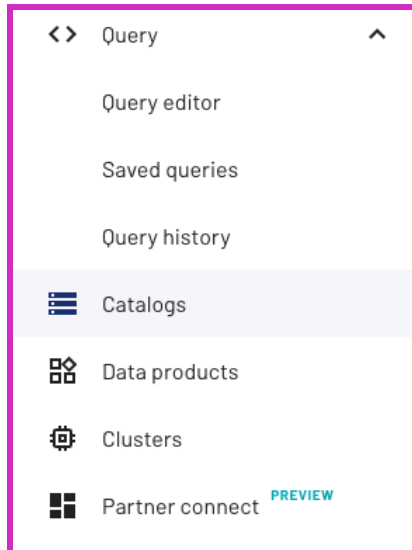
## Part 1: Use schema discovery

### Objective

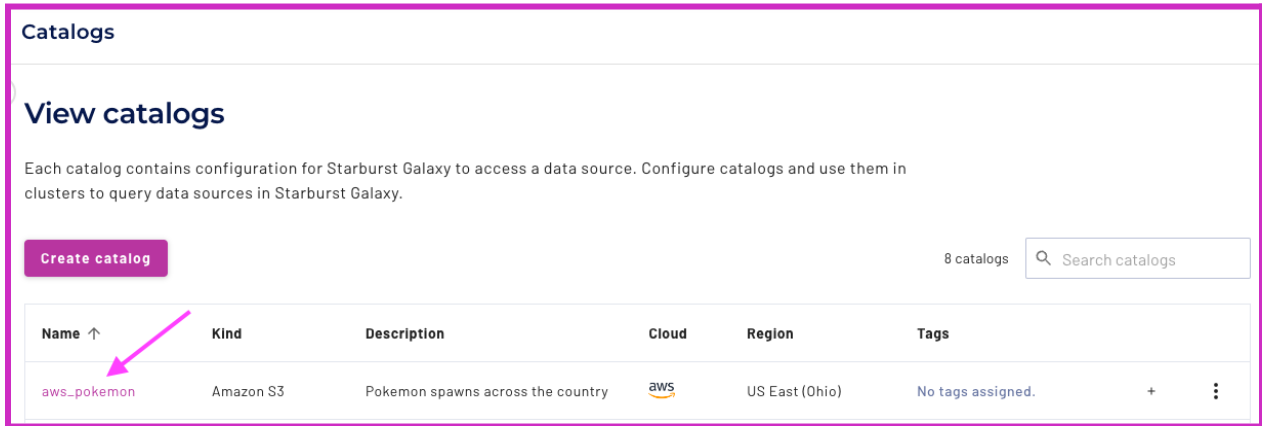
You’re going to begin by utilizing schema discovery to create your schema and table. Schemas control the structure of the data inside them. Luckily for us, Starburst Gravity will take care of the discovery work.

### Step 1 - Navigate to the catalogs page

In the left hand navigation pane, select Catalogs.

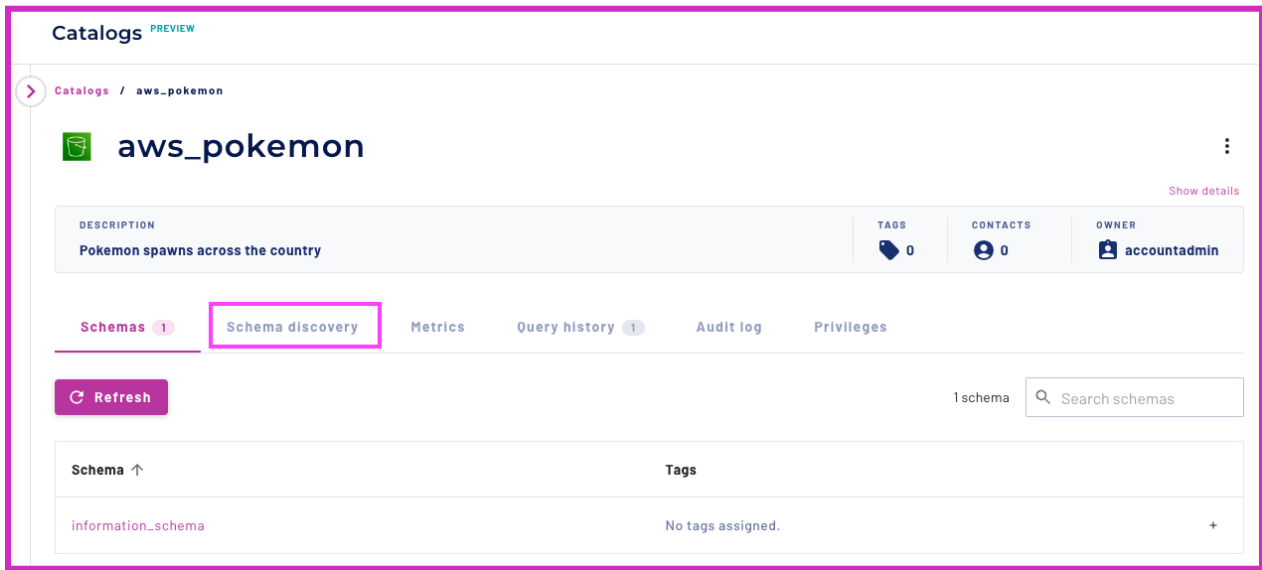


Select the **aws\_pokemon** catalog to navigate within it.



### 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.

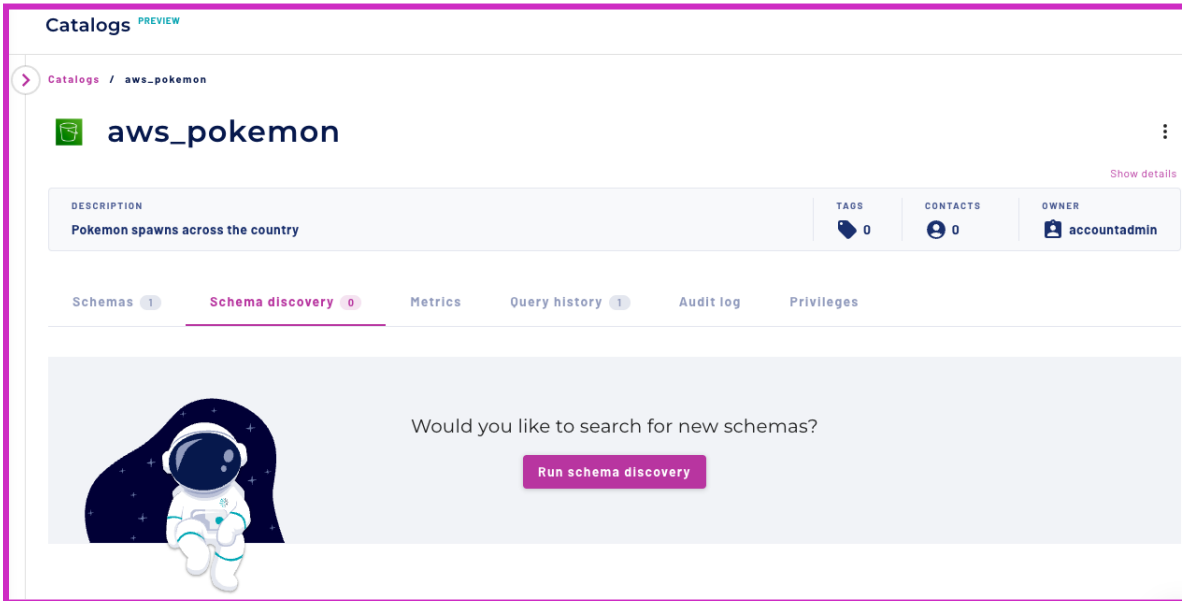


The Schema discovery pane lets you examine the metadata of the specified object storage location. Schema discovery is for catalogs in object storage data sources only.

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.

# Pokemon Lunch and Lab Guide

Select **Run schema discovery**.



Add the following information:

**Catalog location URL:**

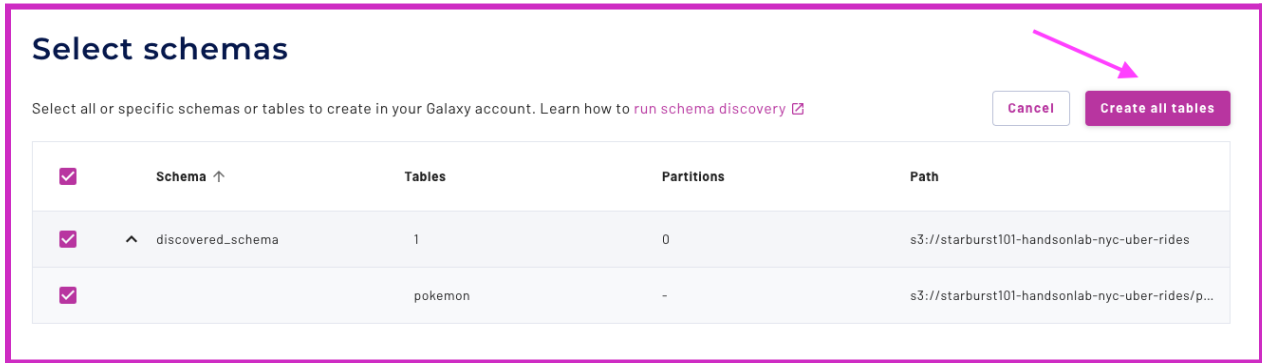
s3://starburst101-handsonlab-nyc-uber-rides/pokemon/

**Default schema:** discovered\_schema

**Type of discovery:** Full discovery

A screenshot of the 'Run schema discovery' dialog box. It contains the following fields and options: 'Catalog location URL \*' with the value 's3://starburst101-handsonlab-nyc-uber-rides/pokemon/'; 'Default schema \*' with the value 'discovered\_schema'; an 'Advanced settings' section with a chevron icon; 'Type of discovery' with two radio button options: 'Incremental discovery from last run' (unselected) and 'Full discovery' (selected); 'Data sample options' with two input fields: 'Max sample file lines' with the value '10' and 'Max files per table' with the value '5'. At the bottom, there are 'Cancel' and 'Run discovery' buttons.

Hit **Run discovery**. Starburst will start scanning for you. Then, it will return code to create your desired schema and table. 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 right hand corner to match the cluster and catalog previously created.

**Cluster:** pokemon-analysis

**Catalog:** aws\_pokemon



## Pokemon Lunch and Lab Guide

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

```
SELECT * from aws_pokemon.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.

### Step 4 - Create your schema

You have already created one schema, named `discovered_schema`. Now, create another schema so that you can separate out the table that was created from an external location and your own work.

Create a new schema with the following command. Highlight and run the command below. Replace `<yourname>` with your actual name, or another identifier that you prefer.

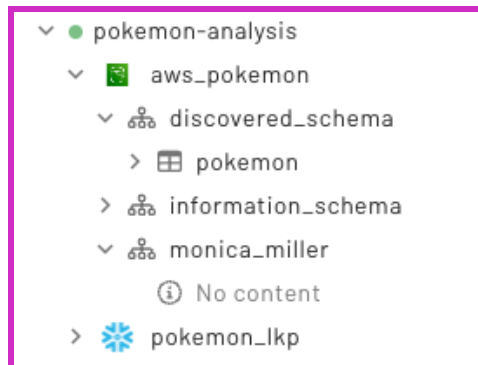
**Note: Only use lowercase characters and numbers; special characters or spaces are not allowed, so remember to replace the `<` `>` characters in the example below. You'll use this throughout the lab, so choose something easy to remember and type.**

```
CREATE SCHEMA aws_pokemon.<yourname>;  
--example: CREATE SCHEMA aws_pokemon.monica_miller;
```

### Step 5 - Verify schema in catalog

In the catalog explorer, expand the `aws_pokemon` catalog to verify your new schema and table are present. There will be other schemas listed.

**Note:** If you named your schema as `yourname`, create the schema again with the instructions above.



## Step 6 - Select schema after creation

In the top right hand corner, select your schema. Validate the location drop-downs in the top right hand corner match the cluster and catalog previously created.

**Cluster:** nyc-ride-info

**Catalog:** nyc\_uber\_rides

**Schema:** `<yourname>` (ex: `monica_miller`)



Keep this tab ready to go, so you can run your queries here.



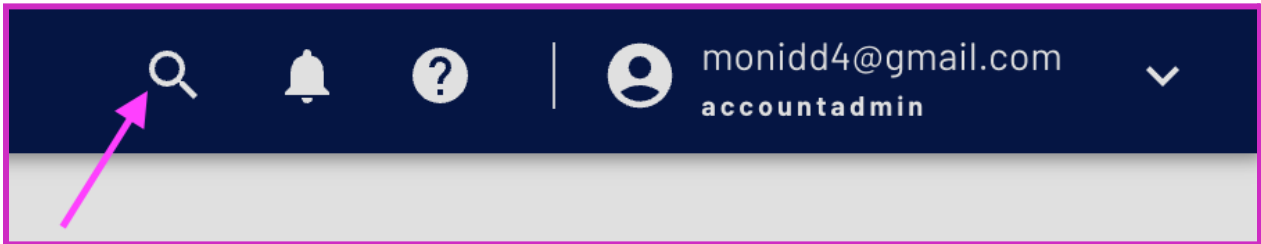
## Part 2: Discover the Snowflake data source

### 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

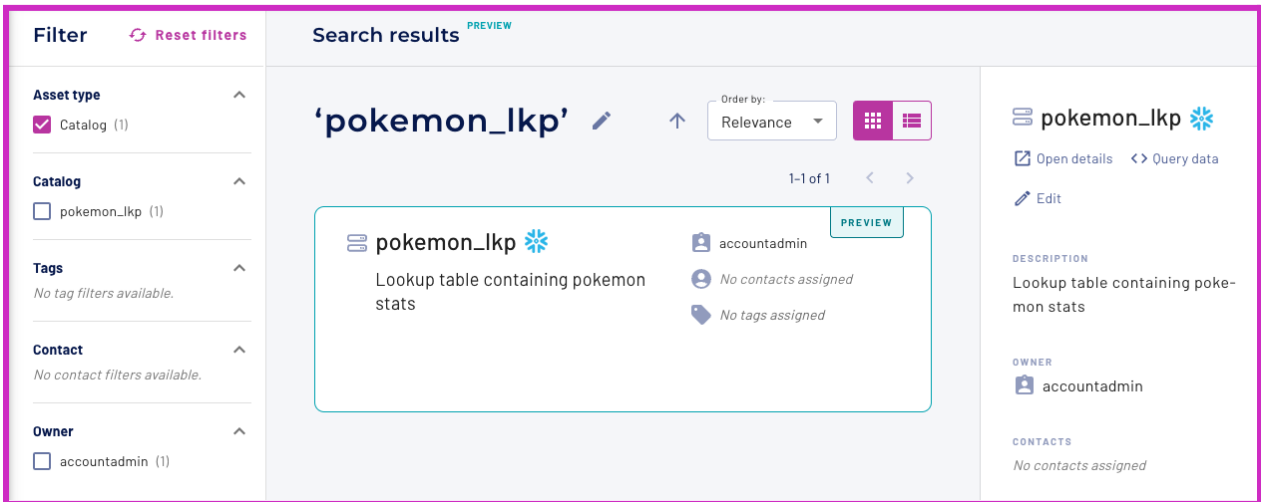
To use global search, select the magnifying glass icon in the upper-right corner.



### Step 2 - Execute global search

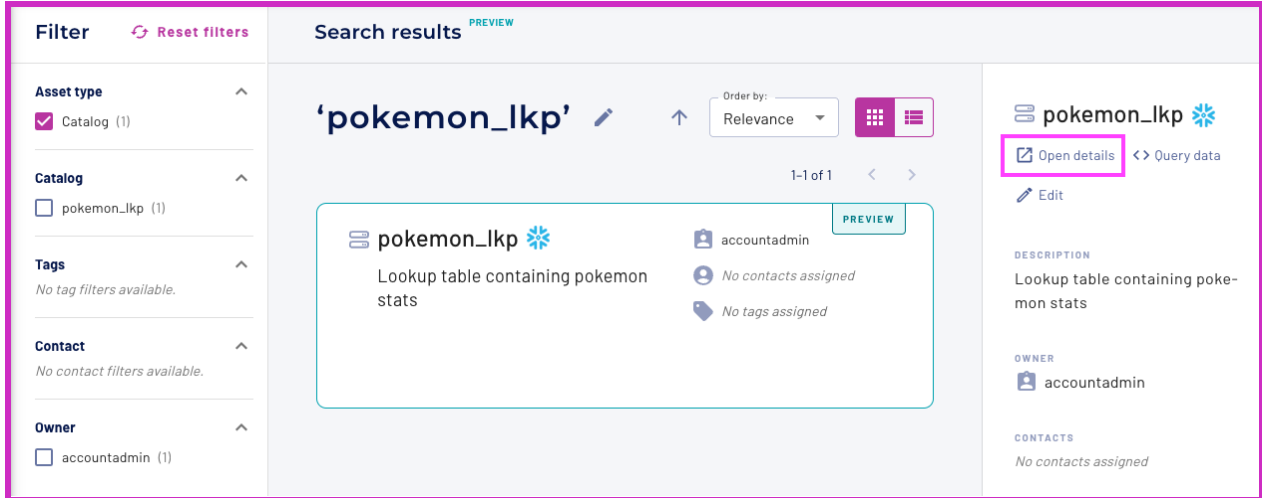
Enter `pokemon_lkp` and select **View all results**.

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



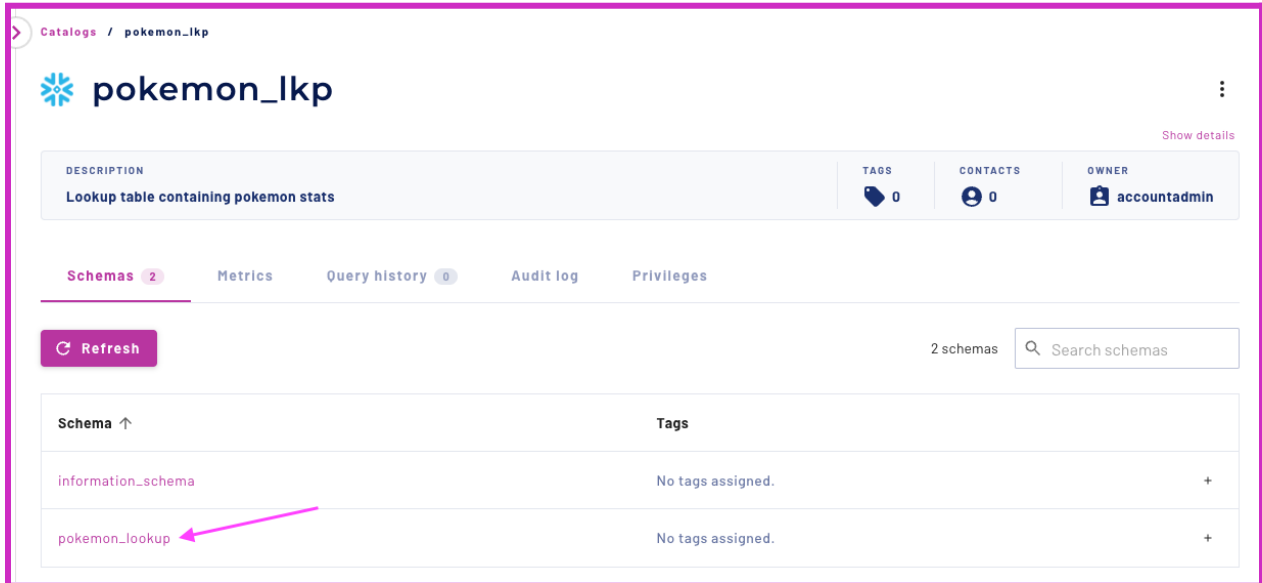
Click on the **Open details** in the top right corner.

# Pokemon Lunch and Lab Guide



## 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.

## Pokemon Lunch and Lab Guide

The screenshot shows the Databricks catalog page for the 'pokemon\_lookup' catalog. The breadcrumb trail is 'Catalogs / pokemon\_lkp / pokemon\_lookup'. The page title is 'pokemon\_lookup' with a 'Promote to data product' button and a 'Show details' link. The 'DESCRIPTION' section contains 'No description provided.' Below this are tabs for 'TAGS' (0), 'LINKS' (0), 'CONTACTS' (0), and 'OWNER' (accountadmin). A navigation bar includes 'Tables 1', 'Views', 'Metrics', 'Definition', 'Query history 0', 'Audit log', and 'Privileges'. A 'Refresh' button is on the left, and '1 table' is shown on the right with a search box. The table list shows one table named 'pokedex' with 'No tags assigned.' and a '+' icon. A pink arrow points to the 'pokedex' table name.

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**.

The screenshot shows the Databricks catalog page for the 'pokedex' table. The breadcrumb trail is 'Catalogs / pokedex / pokemon\_lookup / pokedex'. The page title is 'pokedex' with a 'Show details' link. The 'DESCRIPTION' section contains 'No description provided.' Below this are tabs for 'TAGS' (0), 'CONTACTS' (0), and 'OWNER' (accountadmin). A navigation bar includes 'Columns 17', 'Metrics', 'Definition', 'Data preview', 'Query history 0', 'Audit log', and 'Privileges'. A 'Refresh' button is on the left, and '17 columns' is shown on the right with a search box. The columns table is displayed with the following data:

Column ↑	Type	Nullable	Default	Tags	Description
abilities	varchar(16777216)	yes	null	No tags assigned.	+ No description provided. ✎
att	varchar(16777216)	yes	null	No tags assigned.	+ No description provided. ✎
catch_rate	varchar(16777216)	yes	null	No tags assigned.	+ No description provided. ✎
def	varchar(16777216)	yes	null	No tags assigned.	+ No description provided. ✎

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

## Pokemon Lunch and Lab Guide

The screenshot shows the Pokedex interface with the 'Data preview' tab selected. A pink arrow points to the 'Preview data' button. The interface includes a breadcrumb trail 'Catalogs / pokedex / pokemon\_lookup / pokedex', a 'Show details' link, and a 'DESCRIPTION' section with 'No description provided.'. Below the breadcrumb trail, there are tabs for 'Columns', 'Metrics', 'Definition', 'Data preview', 'Query history', 'Audit log', and 'Privileges'. A dropdown menu for 'Select cluster \*' is set to 'pokemon-analysis'. Below the dropdown is a message: 'You must select a cluster in order to preview the 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**.

The screenshot shows the Pokedex interface with the 'Data preview' tab selected. The 'Preview data' button is highlighted. Below the button is a table with the following data:

name	number	type_1	type_2	hp	att	def
Bulbasaur	1	Grass	Poison	45	49	49
Ivysaur	2	Grass	Poison	60	62	63
Venusaur	3	Grass	Poison	80	82	83
Mega Venusaur	3	Grass	Poison	80	100	123
Charmander	4	Fire	NULL	39	52	43

### 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**.

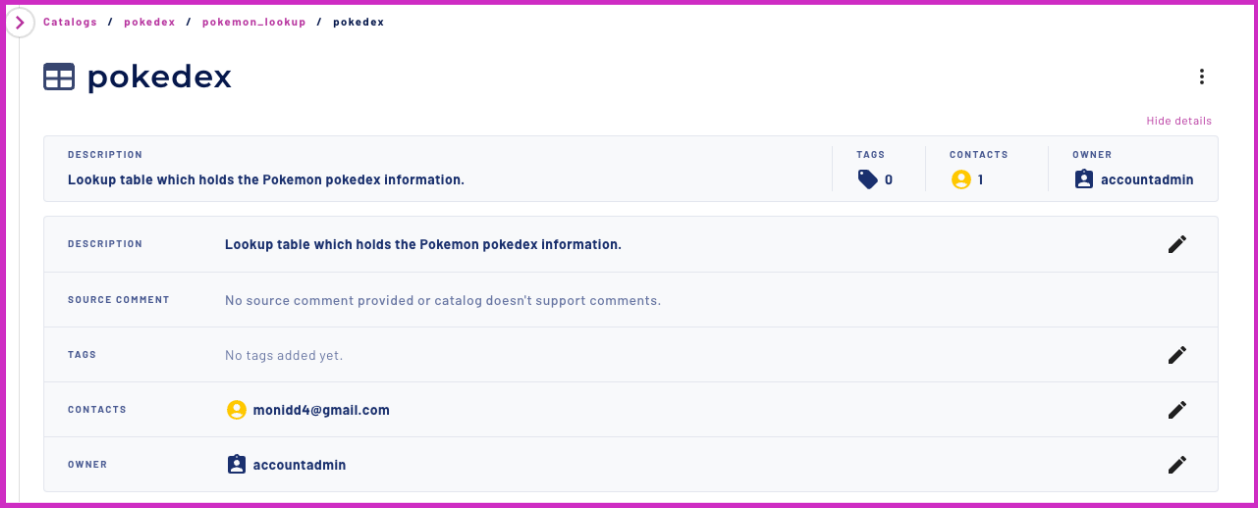
The screenshot shows the Pokedex interface with the 'Show details' button highlighted by a pink arrow. The interface includes a breadcrumb trail 'Catalogs / pokedex / pokemon\_lookup / pokedex', a 'Show details' link, and a 'DESCRIPTION' section with 'No description provided.'. Below the breadcrumb trail, there are tabs for 'Columns', 'Metrics', 'Definition', 'Data preview', 'Query history', 'Audit log', and 'Privileges'. A dropdown menu for 'Select cluster \*' is set to 'pokemon-analysis'. Below the dropdown is a message: 'You must select a cluster in order to preview the data'.

# Pokemon Lunch and Lab Guide

Enter in 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

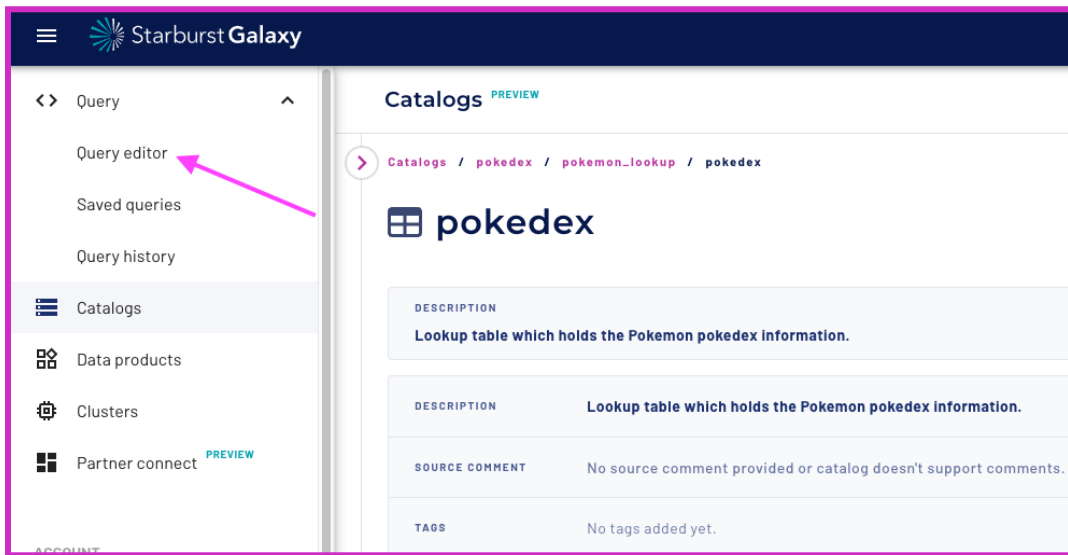
Now it's time to use both your data sources and create a reporting structure in S3.

- **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 an end consumer.

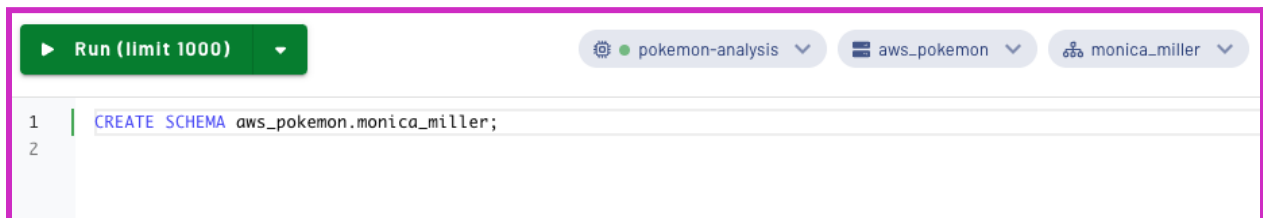
Starburst is special because it allows you to build this reporting structure not just with data that already exists in your data lake, but also with data that exists in other data sources in your orbit - like our Snowflake pokedex data.

### Step 1 - Set up the query editor

Using the left-hand pane, navigate back to the query editor.



Make sure you are in the tab you previously created, or use a new tab that has the cluster, catalog, and schema selected like below.



## Step 2 - Create the pokemon\_spawns structure table

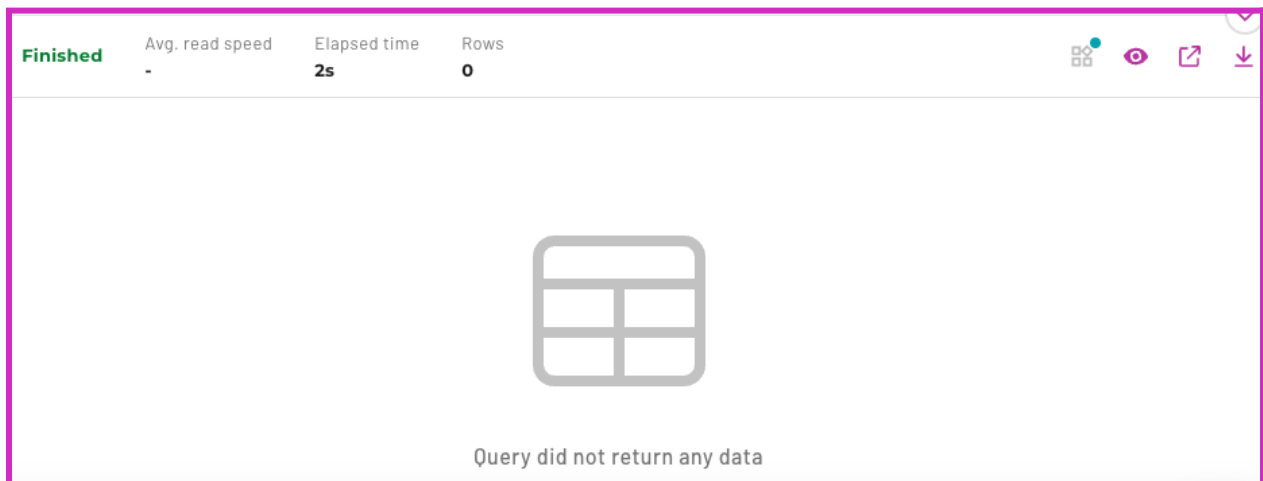
Recall that the default table format you set for this catalog was Iceberg. However, you are still able to create a table with the Hive table format due to [Great Lakes connectivity](#) in Starburst Galaxy. Great Lakes connectivity abstracts the details of using different table formats and file types when using certain write access statements for object storage systems.

Part of your cleansing in the structure layer is to update the table format from Hive to Iceberg and convert the text file to an ORC file. The table you just created with schema discovery is not yet fully optimized and transformed.

To fix this, you must build a structure layer, creating a new table and casting the necessary values into more accurate data types. To help with this, you will also create the table using the ORC file format. This improves performance when using the Iceberg table format.

Run the following command to create the structure layer table.

```
CREATE TABLE pokemon_spawns_structure
(
  number INTEGER,
  name VARCHAR,
  latitude DOUBLE,
  longitude DOUBLE,
  encounter_seconds BIGINT
)
WITH (
  format = 'ORC',
  type = 'ICEBERG'
);
```



### Step 3 - Insert data into the pokemon\_spawns structure table

Once the table is created, you need to insert the data into it.

**Note:** After doing this manually the first time, you can automate the process in the future using an orchestration tool like Airflow, Prefect, or Dagster.

```
INSERT INTO
    pokemon_spawns_structure
SELECT
    CAST(num AS INTEGER) AS number,
    name,
    round(lat, 2) AS latitude,
    round{lng, 2) AS longitude,
    CASE
        WHEN encounter_ms = -1 THEN encounter_ms
        ELSE (encounter_ms / 1000)
    END AS encounter_seconds
FROM aws_pokemon.discovered_schema.pokemon
WHERE lat >= 37.62 and lat <= 37.86 AND lng >= -122.51 and lng <=
-122.12;
```

You should insert 95197 rows.

### Step 3 - Validate the pokemon\_spawns\_structure table

It is considered best practice to validate that the new table has been created. To do this, run a simple select statement.

```
SELECT * FROM pokemon_spawns_structure LIMIT 10;
```

num	name	latitude	longitude	encounter_ms
13	Weedle	37.79	-122.41	1469520187
16	Pidgey	37.79	-122.41	1469520297
41	Zubat	37.79	-122.4	1469520709
16	Pidgey	37.8	-122.41	-1
60	Poliwag	37.8	-122.41	1469520741
46	Paras	37.76	-122.43	1469520422



### Step 4 - Preview the lookup table

The data in the lookup table also needs to be cleaned and optimized. It's best practice to clean up the data in the structure layer before creating the consume layer. First, run a quick join to show the federation capabilities available within Starburst Galaxy and join the newly created S3 `pokemon_spawns_structure` table with the unoptimized Snowflake lookup table `pokemon_lkp`.

Run the following command:

```
SELECT
    S.number,
    S.name,
    S.latitude,
    S.longitude,
    P.type_1,
    P.type_2,
    P.catch_rate,
    p.generation
FROM pokemon_lkp.pokemon_lookup.pokedex p
JOIN pokemon_spawns_structure s ON CAST(p.number AS INTEGER) =
s.number;
```

number	name	latitude	longitude	type_1	type_2	catch_rate	generation
13	Weedle	37.79	-122.41	Bug	Poison	255	1.0
16	Pidgey	37.79	-122.41	Normal	Flying	255	1.0
41	Zubat	37.79	-122.4	Poison	Flying	255	1.0
16	Pidgey	37.8	-122.41	Normal	Flying	255	1.0
60	Poliwag	37.8	-122.41	Water	NULL	255	1.0
46	Paras	37.76	-122.43	Bug	Grass	190	1.0
41	Zubat	37.76	-122.42	Poison	Flying	255	1.0
25	Pikachu	37.76	-122.42	Electric	NULL	190	1.0

You can see that data from two different data sources (catalogs) are returned with one interactive query. This is beneficial when performing interactive analytics, specifically for data consumers who could not otherwise get this information without an engineer.

## Step 5 - Create a pokedex structure table

Now, create a table which will store an optimized version of this table. Create this table as a delta lake table. Why would you create a delta lake table? Because Starburst Galaxy's [Great Lakes connectivity](#) gives you the ability to query multiple table formats at once. You will test this out later in the lab. Run the following command:

```
CREATE TABLE pokemon_pokedex_structure
(
    name VARCHAR,
    number INTEGER,
    type_1 VARCHAR,
    type_2 VARCHAR,
    catch_rate INTEGER,
    final_evolution DOUBLE,
    generation DOUBLE,
    abilities DOUBLE
)
WITH (
    format = 'ORC',
    type = 'DELTA'
);
```

## Step 6 - Insert data into the pokemon\_pokedex structure table

Once the table is created, you need to insert the data into it. Run the following command.

```
INSERT INTO pokemon_pokedex_structure
SELECT
    name,
    CAST(number AS INTEGER) as number,
    type_1,
    type_2,
    CAST(catch_rate AS INTEGER) as catch_rate,
    CAST(final_evolution AS DOUBLE) as final_evolution,
    CAST(generation AS DOUBLE) as generation,
    CAST(abilities AS DOUBLE) as abilities
FROM pokemon_lkp.pokemon_lookup.pokedex;
```

You should insert 1032 rows.

## Step 7 - Validate the pokemon\_pokedex\_structure table

It is considered best practice to validate that the new table has been created. To do this, run a simple select statement.

```
SELECT * FROM pokemon_pokedex_structure LIMIT 10;
```

name	number	type_1	type_2	catch_rate	final_evolution	generation	abilities
Bulbasaur	1	Grass	Poison	45	0	1	6.9
Ivysaur	2	Grass	Poison	45	0	1	13
Venusaur	3	Grass	Poison	45	1	1	100
Mega Venusaur	3	Grass	Poison	45	1	6	155.5
Charmander	4	Fire	NULL	45	0	1	8.5
Charmeleon	5	Fire	NULL	45	0	1	19
Charizard	6	Fire	Flying	45	1	1	90.5

## Part 4: Build the consume layer

### Objective

Now it's time to construct the last of the three layers of your data lake reporting structure, the consume layer. This will ready the layer for final consumption by data consumers.

### Step 1 - Revisit the business requirements

Take this opportunity to refamiliarize yourself with the business case. This will inform the kinds of questions that you will ask about your dataset.

In this scenario, the business case required you to:

1. Create a final table output combining data from both structure tables.
2. Create a data product answering two specific business questions from the marketing department.
  - c. What are the easiest and most popular Pokemon to catch in San Francisco by Type\_1?
  - d. 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.*

### Step 2 - Create the consume table

Run the following query to create the consume table. This constructs a new table from two separate tables in S3. Note that the `pokemon_spawns_structure` table is an Iceberg table and the `pokemon_pokedex_structure` table is a Delta Lake table.

```
CREATE TABLE pokemon_final_spawns AS
SELECT
    s.number,
    s.name,
    s.latitude,
    s.longitude,
    p.type_1,
    p.type_2,
    p.catch_rate
FROM pokemon_pokedex_structure p
JOIN pokemon_spawns_structure s ON p.number = s.number
WHERE catch_rate > 100;
```

The results should show 94626 rows. Run a `SELECT` statement to validate that the table's values are similar to the image below.

```
SELECT * FROM pokemon_final_spawns LIMIT 10;
```

## Pokemon Lunch and Lab Guide

number	name	latitude	longitude	type_1	type_2	catch_rate
13	Weedle	37.79	-122.41	Bug	Poison	255
16	Pidgey	37.79	-122.41	Normal	Flying	255
41	Zubat	37.79	-122.4	Poison	Flying	255
16	Pidgey	37.8	-122.41	Normal	Flying	255
60	Poliwag	37.8	-122.41	Water	NULL	255

### Step 3 - Find the easiest and most popular Pokemon in San Francisco

Now, you need to derive two different views for the marketing department. These will be used to answer the business questions. Remember, easiest is defined by a catch rate of greater than or equal to 100. The most popular is defined as the most number of appearances for a certain Pokemon for each Type\_1.

Create a [window function](#) to rank the most popular Pokemon for each Type\_1.

```
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;
```

## Pokemon Lunch and Lab Guide

Next, utilize a WITH statement to run a subquery and only select the most popular Pokemon types.

```
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
ORDER BY
  total_appearances DESC;
```

type_1	name	total_appearances
Poison	Zubat	17444
Normal	Pidgey	10636
Bug	Weedle	5352
Dark	Rattata	5198
Water	Magikarp	3090
Fairy	Clefairy	1909

### Step 4 - Create the first marketing view

Finally, you need to create a view to make the data visible to the marketing team. Use the SQL statement below.

```
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;
```

Run a select statement to validate the view was created properly.

```
SELECT * FROM popular_types_sf_vw;
```

type_1	name	total_appearances
Bug	Weedle	5352
Water	Magikarp	3090
Poison	Zubat	17444
Grass	Bellsprout	1278

## Step 5 - Create the second marketing view

Use a [Grouping set](#) to find the total number of Pokemon caught for each Type\_1 and Type\_2 pairing.

```
CREATE OR REPLACE VIEW counts_by_types_sf_vw AS
SELECT
    type_1,
    type_2,
    ROUND(AVG(catch_rate), 2) AS avg_catch_rate,
    COUNT(name) AS total_count
FROM
    pokemon_final_spawns
GROUP BY
    GROUPING SETS ((type_1, type_2))
ORDER BY
    type_1,
    total_count DESC;
```

Run a select statement to validate the view was created properly. The results should be similar to the image below.

```
SELECT * FROM counts_by_types_sf_vw;
```

type_1	type_2	avg_catch_rate	total_count
Bug	Poison	235.58	7180
Normal	Flying	236.84	19739
Poison	Flying	255	17444
Bug	Grass	190	1582
Electric	NULL	190	979
Fighting	NULL	187.32	1127
Fairy	NULL	150	1909
Poison	NULL	233.24	5884
Grass	Poison	249.86	2627
Dark	Normal	251.34	5351



## Part 5: 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.

### Step 1 - Create a marketing role

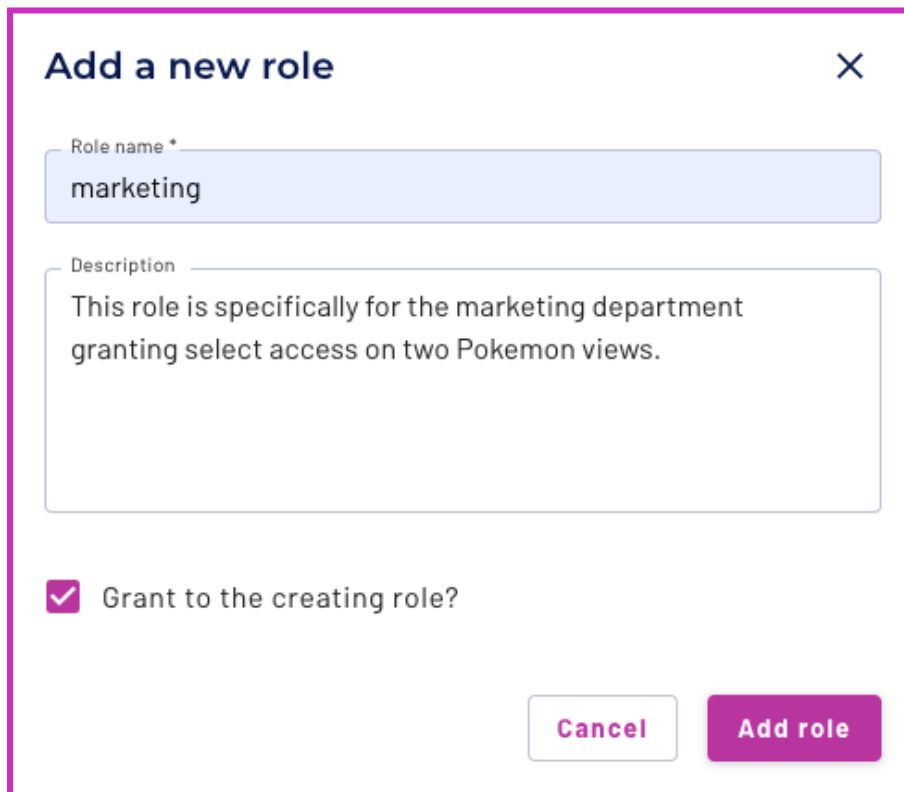
To restrict access to the consume layer, you're going to create a specific role for the marketing department. This will restrict access to the data to team members with the appropriate rights, and restrict their access to the two newly created views.

To do this, navigate to the **Roles and privileges** tab. Select **Add role**. 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



**Add a new role** ✕

Role name \*  
marketing

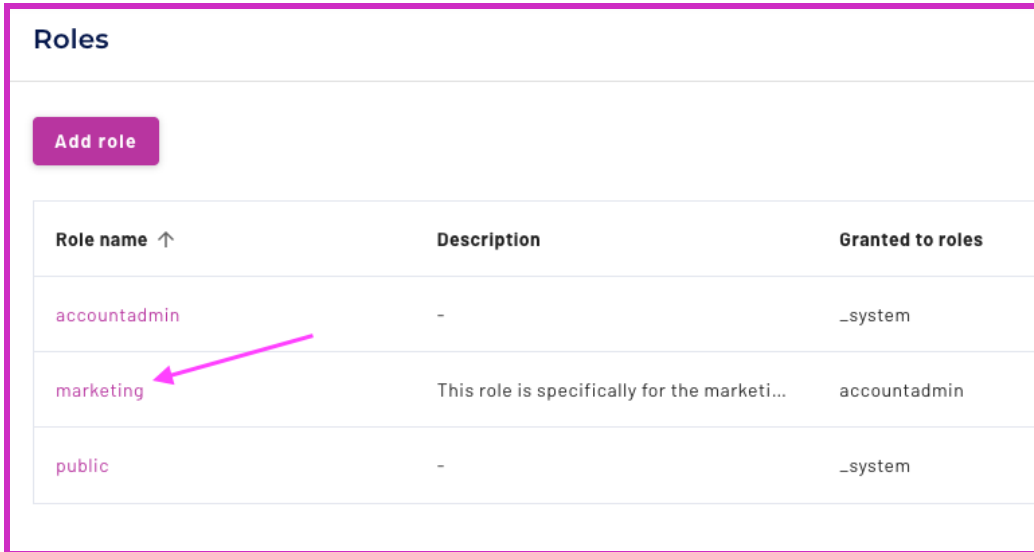
Description  
This role is specifically for the marketing department granting select access on two Pokemon views.

Grant to the creating role?

Cancel Add role

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

## Pokemon Lunch and Lab Guide



**Roles**

**Add role**

Role name ↑	Description	Granted to roles
accountadmin	-	_system
marketing	This role is specifically for the marketi...	accountadmin
public	-	_system

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

**What would you like to modify privileges for?** Table

**Which catalog contains the tables?** aws\_pokemon

**Which schemas can this role access?** <yourname> (ex: monica\_miller)

**Which tables can this role access?** popular\_types\_sf\_vw

**Do you want to allow or deny access?** Allow

**What can they do?** Select from table

## Pokemon Lunch and Lab Guide

The screenshot shows the 'Add permissions' page in the AWS IAM console. The 'Schema name' field is filled with 'monica\_miller'. Under 'Which tables can this role access?', the 'All tables' checkbox is unchecked, and the 'Table name' field contains 'popular\_types\_sf\_vw'. Under 'Do you want to allow or deny access?', the 'Allow' radio button is selected. A light blue information box states: 'Role **accountadmin** will also inherit these privileges. Refer to our detailed [documentation on role inheritance](#).' Under 'What can they do?', the 'All privileges below' checkbox is unchecked. The 'Select from table' checkbox is checked, while 'Delete from table', 'Insert into table', and 'Update table rows' are unchecked.

Select **Add privileges**.

Now it's time to do the same process again to address the second business requirement.

Repeat the process for the table `counts_by_types_sf_vw`.

**What would you like to modify privileges for?** Table

**Which catalog contains the tables?** `aws_pokemon`

**Which schemas can this role access?** `<yourname>` (ex: `monica_miller`)

**Which tables can this role access?** `counts_by_types_sf_vw`

**Do you want to allow or deny access?** Allow

**What can they do?** Select from table

# Pokemon Lunch and Lab Guide

Schema name \*  
monica\_miller

### Which tables can this role access?

Either select all tables within this catalog or specify access to a single table. To give access to multiple tables, use the SQL interface.

All tables

Table name \*  
counts\_by\_types\_sf\_vw

### Do you want to allow or deny access?

Allow  Deny

*i* Role **accountadmin** will also inherit these privileges. Refer to our detailed [documentation on role inheritance](#).

### What can they do?

Allow or deny users the ability to perform the following actions on tables.

All privileges below

---

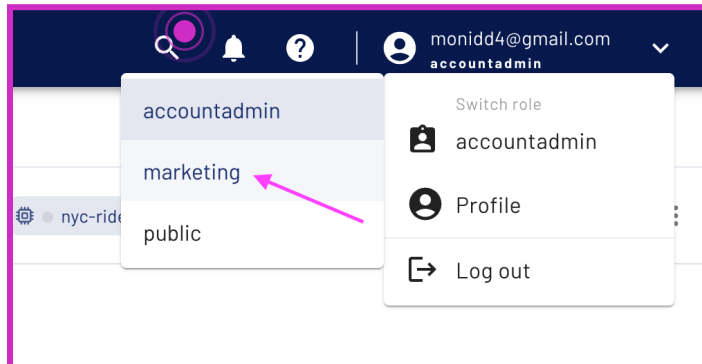
Delete from table  Insert into table

Select from table  Update table rows

## Step 2 - Test the marketing role

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

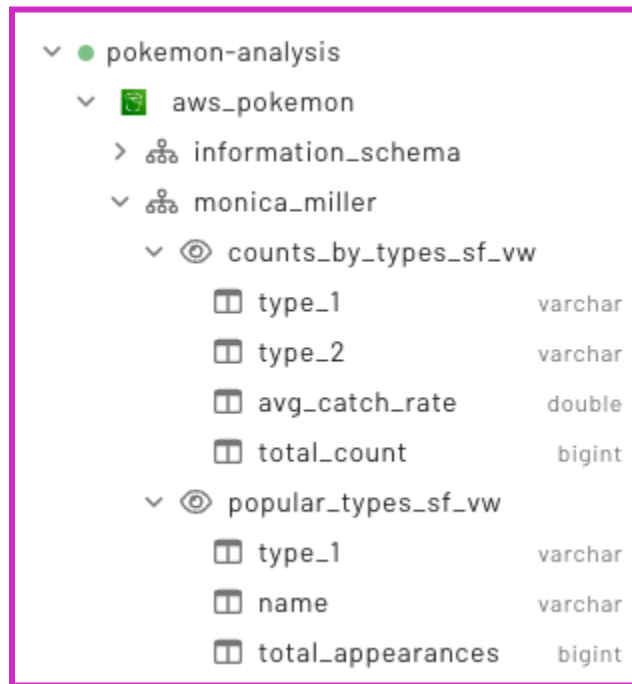
Navigate back to the **Query editor** and switch to the **marketing** role in the top right-hand corner.



## Pokemon Lunch and Lab Guide

Notice that the marketing role view has fewer capabilities available in the left-hand navigation bar.

Also, the cluster explorer shows only two views in the `pokemon-analysis` cluster. This is exactly what you would expect.



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 view using the marketing role. Run the command shared in [Lab 3 Part 4 Step 4](#). This will fail because you have only granted the marketing role select permissions.



Access Denied: Cannot create view `nyc_uber_rides.monica_miller.borough_most_pop_month_vw`: Role `marketing` does not have the privilege `CREATE_TABLE` on the schema `nyc_uber_rides.monica_miller`

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 (Bonus)

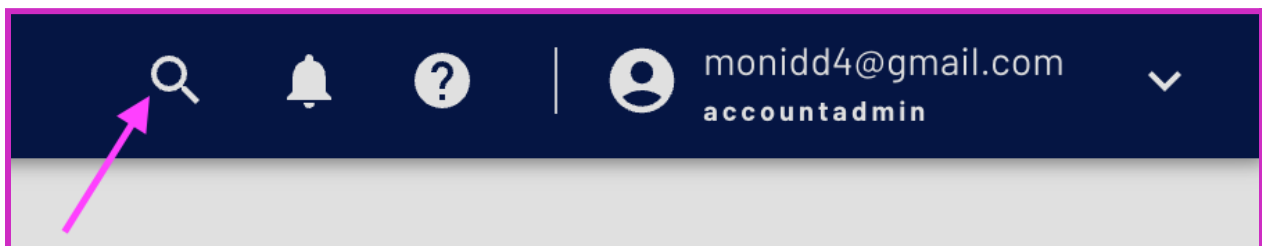
## Part 1: Execute global search

### 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.

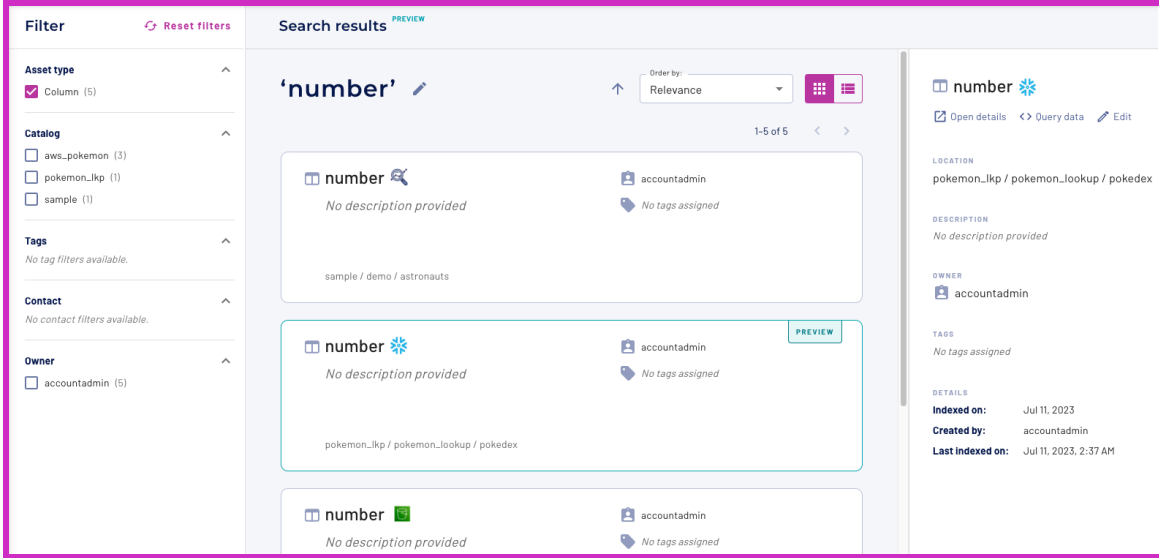


### Step 2 - Execute global search

Enter the word `number` and select **View all results**.

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.

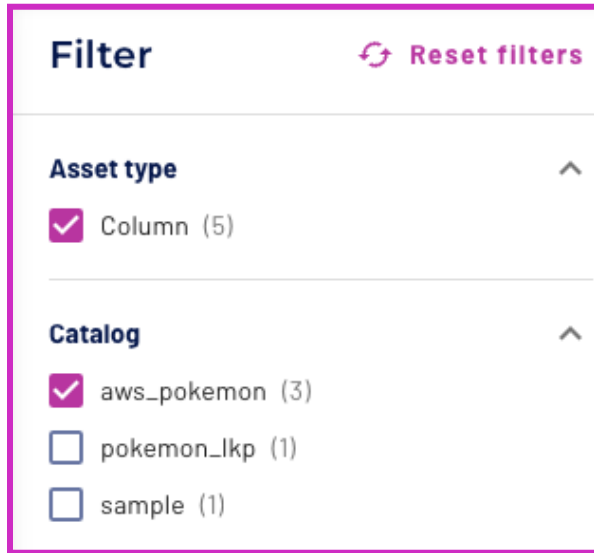
# Pokemon Lunch and Lab Guide



### Step 3 - Filter global search

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

Select the Catalog filter for aws\_pokemon on the left to see how different search criteria impact the results.



### Step 3 - Navigate to your newly created view

Open the details for any of the aws\_pokemon results. Notice that Starburst Galaxy automatically places you within the catalog page. Navigate to your schema by clicking within your defined schema name.



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

Make sure you are in your schema as mentioned in the previous step. Before creating a data product, enter some information into your catalog.

Select **Show details** on the right hand side.

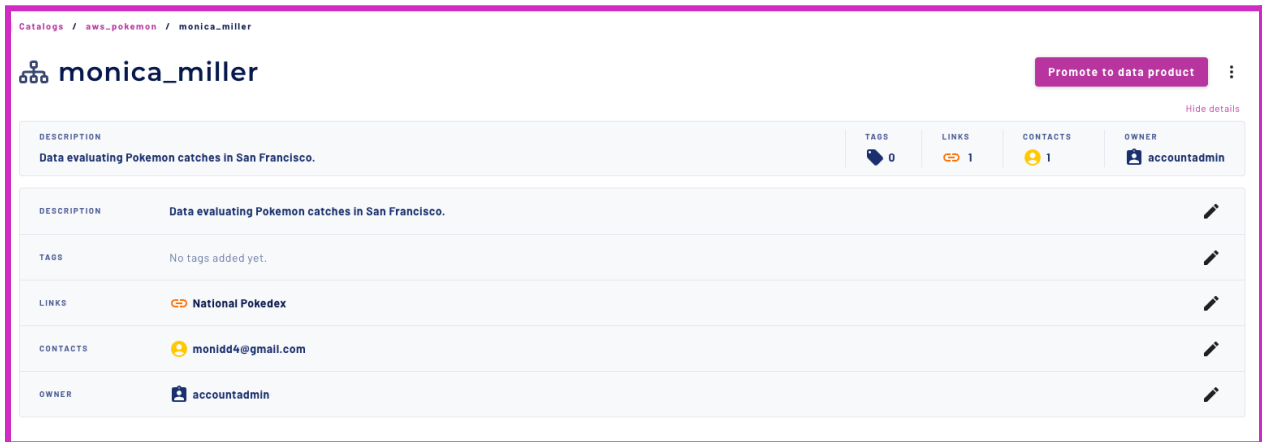
Edit the following information:

**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.



# Pokemon Lunch and Lab Guide

The screenshot shows a data catalog entry for a view named 'counts\_by\_types\_sf\_vw'. The view's description is 'Total number of Pokemon appearances for each type\_1 and type\_2 pairing.' The interface includes tabs for 'Columns', 'Metrics', 'Definition', 'Data preview', 'Query history', 'Audit log', and 'Privileges'. A 'Refresh' button is visible. Below the navigation is a table with 4 columns: 'Column', 'Type', 'Nullable', 'Default', 'Tags', and 'Description'. The table lists four columns: 'avg\_catch\_rate' (double), 'total\_count' (bigint), 'type\_1' (varchar), and 'type\_2' (varchar). Each column has a '+' icon and a description.

Column ↑	Type	Nullable	Default	Tags	Description
avg_catch_rate	double	yes	null	No tags assigned.	+ Average catch rate of each pok...
total_count	bigint	yes	null	No tags assigned.	+ Total appearance count for ea...
type_1	varchar	yes	null	No tags assigned.	+ Primary type
type_2	varchar	yes	null	No tags assigned.	+ Secondary type

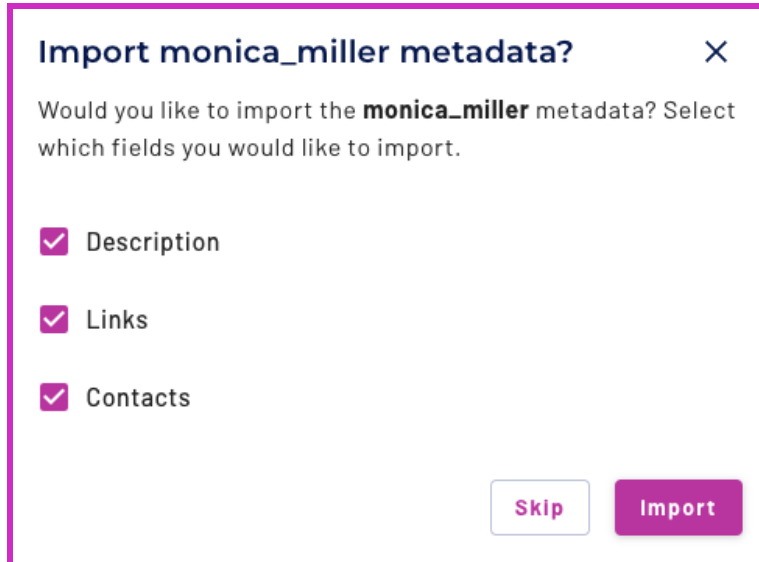
## Step 2 - Promote your data product

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

The screenshot shows a data catalog entry for a schema named 'monica\_miller'. The interface includes a 'Promote to data product' button, which is highlighted with a pink arrow. Below the navigation are tabs for 'Tables', 'Views', 'Metrics', 'Definition', and 'Query'. A 'Refresh' button is visible. Below the navigation is a table with 3 tables: 'pokemon\_final\_spawns', 'pokemon\_pokedex\_structure', and 'pokemon\_spawns\_structure'. Each table has a '+' icon and a description.

Table ↑	Tags
pokemon_final_spawns	No tags assigned. +
pokemon_pokedex_structure	No tags assigned. +
pokemon_spawns_structure	No tags assigned. +

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



**Import monica\_miller metadata?** X

Would you like to import the **monica\_miller** metadata? Select which fields you would like to import.

- Description
- Links
- Contacts

Skip Import

Add a descriptive name like **SF-pokemon-analysis**. 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 the **pokemon-analysis** cluster as the default cluster. Your contacts and supporting information have been automatically populated.

Select **Promote to data product**.

## Pokemon Lunch and Lab Guide

### Default cluster

Select cluster


### Supporting information

These links can be added and edited for the data product. They will not affect the schema links.

Text to display \*  Link URL \*

+ Add another



### Contacts

 monidd4@gmail.com

Contacts \*

Congratulations! You have created and promoted your first data product. Navigate to the Data products tab to view your work.


Learn more about [data products](#)

  1 data product

### SF-pokemon-analysis

Data evaluating Pokemon catches in San Francisco.

---

CONTACTS:  1

## 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 the **Tags** pane on the left-hand side navigation. Select **Create tag**.

**Name:** geolocation

**Description:** This identifies data based on latitude and longitude.

**Color:** Your choice

The screenshot shows a 'Create tag' modal window. At the top left is the title 'Create tag' and a close button (X). Below the title is a text input field for 'Name \*' containing the text 'geolocation'. Underneath the name field, it says '4 characters remaining'. Below the name field is a checkbox labeled 'Nested tag under:' and a dropdown menu labeled 'Select nesting'. Below that is a text area for 'Description' containing the text 'This identifies data based on latitude and longitude.'. At the bottom of the modal, there are seven circular color selection buttons in a row: red, orange, yellow, green, teal, blue, and purple. The teal button is currently selected. To the right of the color buttons are two buttons: 'Cancel' and 'Create tag'.

## 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.

The screenshot shows a data product page for 'SF-pokemon-analysis (monica\_mi...)'. At the top, there are tabs for 'DESCRIPTION', 'TAGS', 'LINKS', 'CONTACTS', and 'OWNER'. The 'TAGS' tab shows 0 tags. Below this is a navigation bar with 'Tables 3', 'Views 2', 'Metrics', 'Definition', and 'Query history 21'. A 'Refresh' button is on the left, and a search box for views is on the right. The main content area shows a table with two columns: 'Table' and 'Tags'. Two views are listed: 'counts\_by\_types\_sf\_vw' and 'popular\_types\_sf\_vw', both with a 'geolocation' tag assigned to them.

Table	Tags
counts_by_types_sf_vw	geolocation +
popular_types_sf_vw	geolocation +

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