



Starburst Galaxy 101 Workshop

APRIL 27TH AT 09.30AM BST
10.30AM CEST

Handout

**The fastest path from
data to insight**

The wait is over. Bring all of your data to every critical decision.

Table of Contents

About	1
Sign In into Starburst Galaxy	1
Create Amazon S3 Catalog	2
Create catalog and select data source	2
Configure the connection	3
Optional Steps	5
Create a cluster	6
Query S3 Data	8
Open Query Editor	8
Create data structures	8
Query the data	10
Create a PostgreSQL Catalog	11
Create catalog and select data source	11
Configure the connection	12
Optional Steps	14
Federate Amazon S3 and PostgreSQL	16
Query PostgreSQL alone	16
Join Amazon S3 data with PostgreSQL data	16

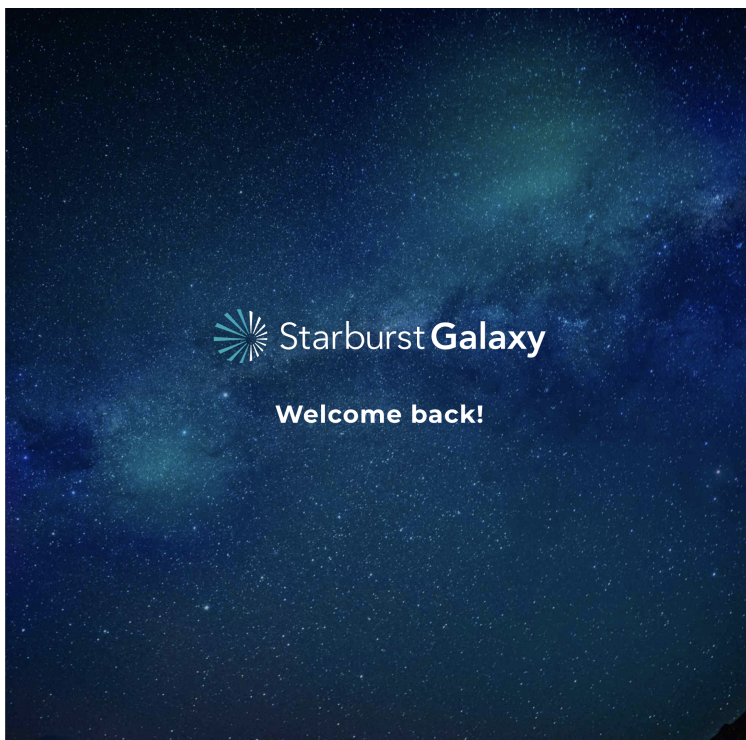
About

This guide demonstrates how to get started with Starburst Galaxy and follow along the hands-on webinar hosted by Starburst. You'll connect to your Starburst Galaxy account, create datasources, define schemas as well as tables, and query them using SQL.

Sign In into Starburst Galaxy

Before today's webinar you were asked to create your very own Starburst Galaxy Account. You will use this account to follow along with the hands-on parts of the webinar. So the very first step is to actually log into your account.

Go to https://<your_identifier>.galaxy.starburst.io/login and you will be greeted with the login form



Sign in to Starburst Galaxy

Email 1
douglas.adams@starburst.io

Password 2

[Forgot your password?](#) 3 **Sign in to Starburst Galaxy**

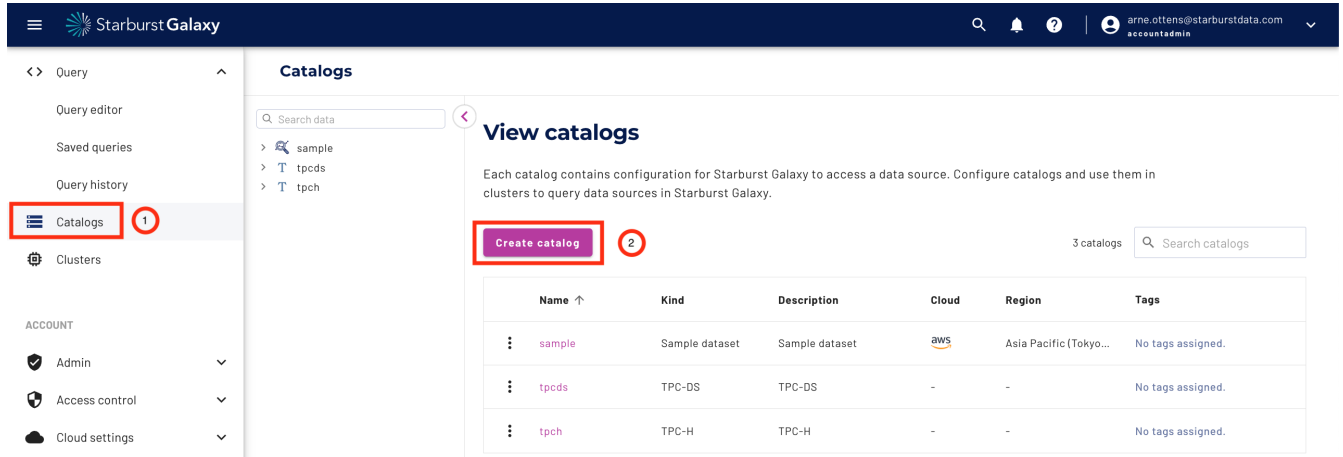
In order to log in, do the following:

1. Enter the E-Mail address you used for signing up with Starburst Galaxy
2. Enter the password you set during account creation
3. Click the **[Sign in to Starburst Galaxy]** Button

Create Amazon S3 Catalog

Now you will add your first data source, an Amazon S3 Bucket. You will be provided with a S3 bucket and respective Access/Secret Keys by Starburst for the duration of one week. The following steps will walk you through the setup of an Amazon S3 Catalog.

Create catalog and select data source



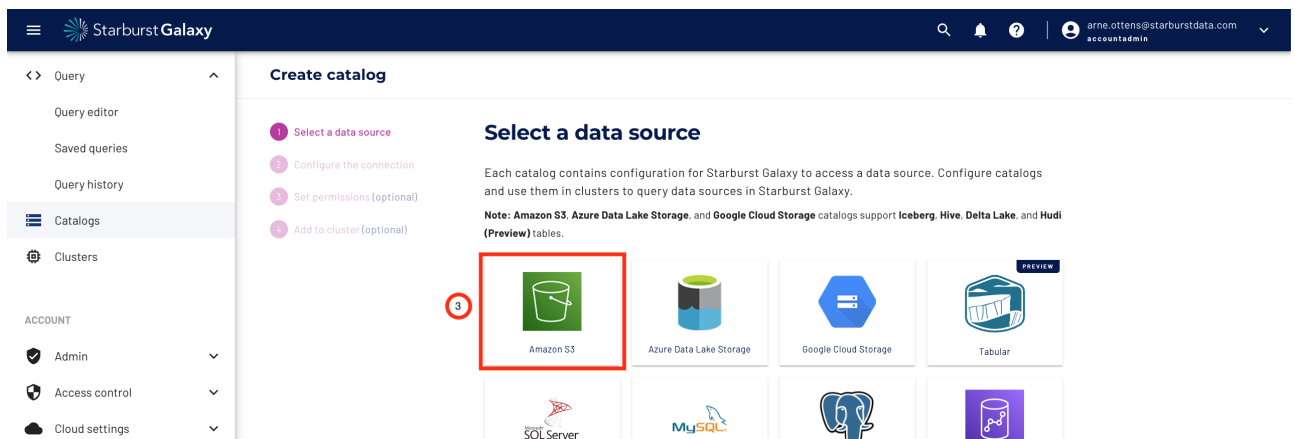
View catalogs

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.

3 catalogs

Name ↑	Kind	Description	Cloud	Region	Tags
sample	Sample dataset	Sample dataset	aws	Asia Pacific (Tokyo...	No tags assigned.
tpcds	TPC-DS	TPC-DS	-	-	No tags assigned.
tpch	TPC-H	TPC-H	-	-	No tags assigned.

1. Click Catalogs on the left
2. Click **[Create catalog]**



Create catalog

1. Select a data source
2. Configure the connection
3. Set permissions (optional)
4. Add to cluster (optional)

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.

3 **Amazon S3** Azure Data Lake Storage Google Cloud Storage Tabular

SQL Server MySQL ElephantDB

3. Select **Amazon S3** as a data source

Configure the connection

Starburst Galaxy

<> Query

Catalogs

Clusters

ACCOUNT

Admin

Access control

Cloud settings

Create catalog

- ✓ Select a data source
- 2 Configure the connection
- 3 Set permissions (optional)
- 4 Add to cluster (optional)

Amazon S3

Configure your catalog to query objects in Amazon S3. Learn more about [connecting to S3](#).

Name and description

Provide a unique name to identify the catalog in your SQL queries in the query editor and other client tools. The namespace for a table is typically <catalog_name>.<schema_name>.<table_name>

4 Catalog name *

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

Description

4. Give it a **name**. During the workshop we will refer to it as **s3**

Authentication to S3

Choose the **authentication mechanism** to connect to S3.

Authentication with *

☐ Cross account IAM role ☒ AWS access key 5

6 AWS access key for S3 *

7 AWS secret key for S3 *

5. Select **AWS access key** as Authentication Method

6. Access Key: **AKIASX5N4NQVJ4THP2QR**

7. Secret Key: **ch7qPNhczmAKR2+ks6XUlgfUx8r1EGZG8dyPsaK8**

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 8

9 Default S3 bucket name * ?

10 Default directory name * ?

11 ☒ Allow creating external tables ?

☐ Allow writing to external tables ?

8. Select **Starburst Galaxy** as the Metastore Configuration

9. Default S3 bucket name: **ao-emea-workshop**

10. Default S3 bucket name: **firstname_lastname**

11. Activate **Allow creating external tables**

Test connection

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

Detected regions:

-  Europe (Frankfurt)

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

Test connection 12

< Back

14 **Connect catalog**

12. Scroll to the end and Click **[Test Connection]**

13. You should see the **Hooray** message, otherwise contact the instructor

14. Click **[Connect catalog]**

Optional Steps

Create catalog

- ✓ Select a data source
- ✓ Configure the connection
- 3 Set permissions (optional)
- 4 Add to cluster (optional)

Set permissions

Now that your **s3** catalog has been created, assign users access with roles. [Learn how to create roles here.](#)

Catalog-level permissions

☐ Read-only catalog

Prohibits all users, **including the catalog owner**, from modifying data or metadata in this catalog.

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

15 Skip **Save access controls**

15. Click **[Save access controls]**

Create catalog

- ✓ Select a data source
- ✓ Configure the connection
- ✓ Set permissions (optional)
- 4 Add to cluster (optional)

Add to cluster

Attach your **s3** 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

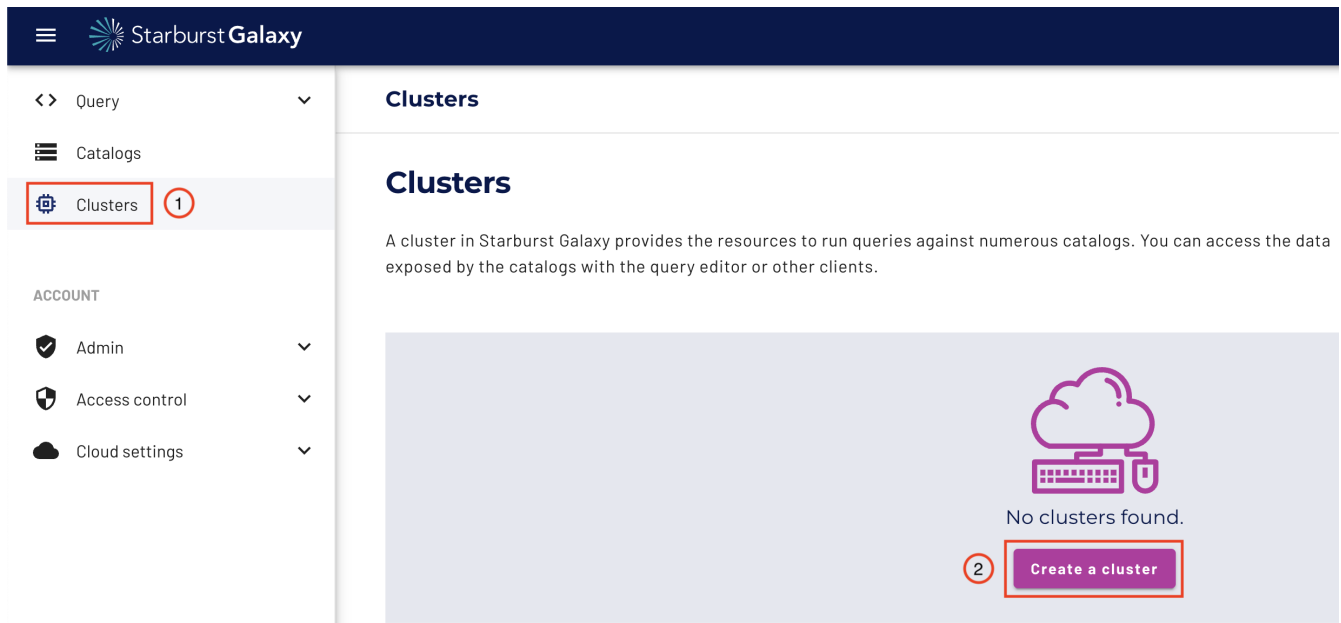
16 Skip Add to cluster

16. Click **[Skip]**

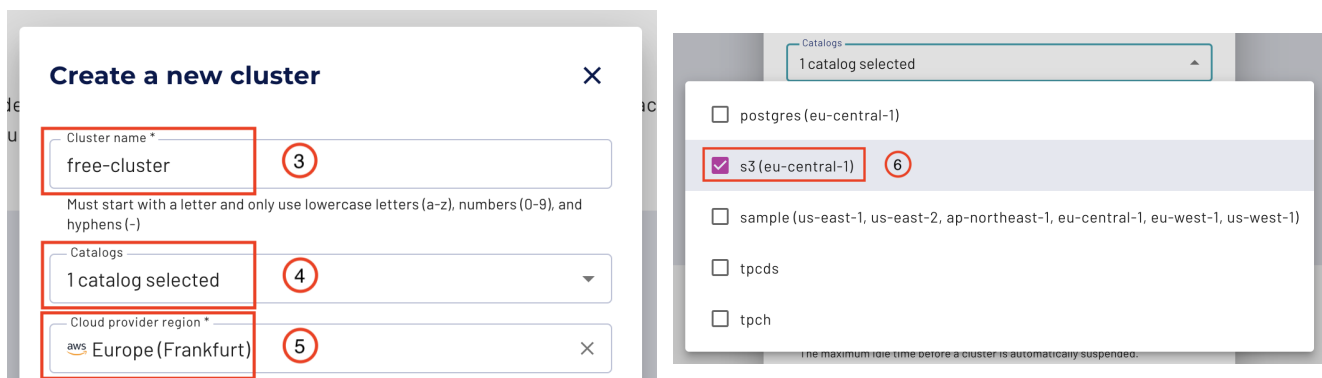
Congratulations! You created your first data source in Starburst Galaxy.

Create a cluster

Now you will add your first Starburst Galaxy Cluster. The account you created comes with free credits, so you can test everything out. For the webinar we will work with a free cluster, so your credits won't be used. The following steps will walk you through the setup of a Cluster.



1. Click **Clusters** on the left
2. Click **[Create a cluster]** and a pop up dialog opens



3. Define a **Cluster name**. During the workshop we will refer to it as **free-cluster**
4. Open the **Catalogs** dropdown menu and select your **s3** catalog (see 6.)
5. Select the **Cloud provider region: Europe (Frankfurt)**

Cluster type

Execution mode *

☒ Standard

7

Cluster size *

Free

8

Idle shutdown time

30 Minutes

9

The maximum idle time before a cluster is automatically suspended.

Advanced settings

Canc 10

Create cluster

- Define the **Execution mode** as **Standard**
- Define the **Cluster Size** as **Free**
- Define the **Idle shutdown time** as **30 Minutes**
- Click **[Create cluster]**

The pop up dialog will close and the newly created cluster will be listed.

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

1 cluster

Search clusters

Name ↑	Status	Quick actions	Catalogs	Region	Cluster type	Size	Auto suspend	Connect
free-cluster	Starting 11		s3	aws Europe (Frankfurt)	Standard	Free	30 minutes	Connection i

- The cluster you just created will now start.

Name ↑	Status ↑	Quick actions	Catalogs	Region	Cluster type	Size	Auto suspend	Connect
free-cluster	Running 12	Stop	s3	aws Europe (Frankfurt)	Standard	Free	30 minutes	Connection i

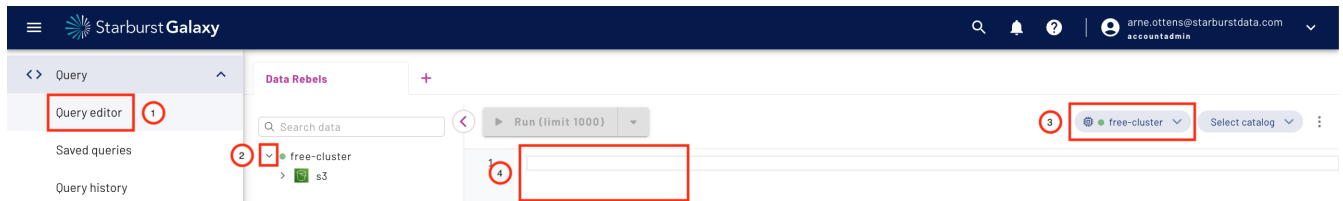
- After a couple minutes the cluster will be up and running.

Congratulations! You created your first cluster in Starburst Galaxy.

Query S3 Data

The following steps will walk you through the creation of schemas and tables on top of the data in your Amazon S3 Catalog as well as querying the data.

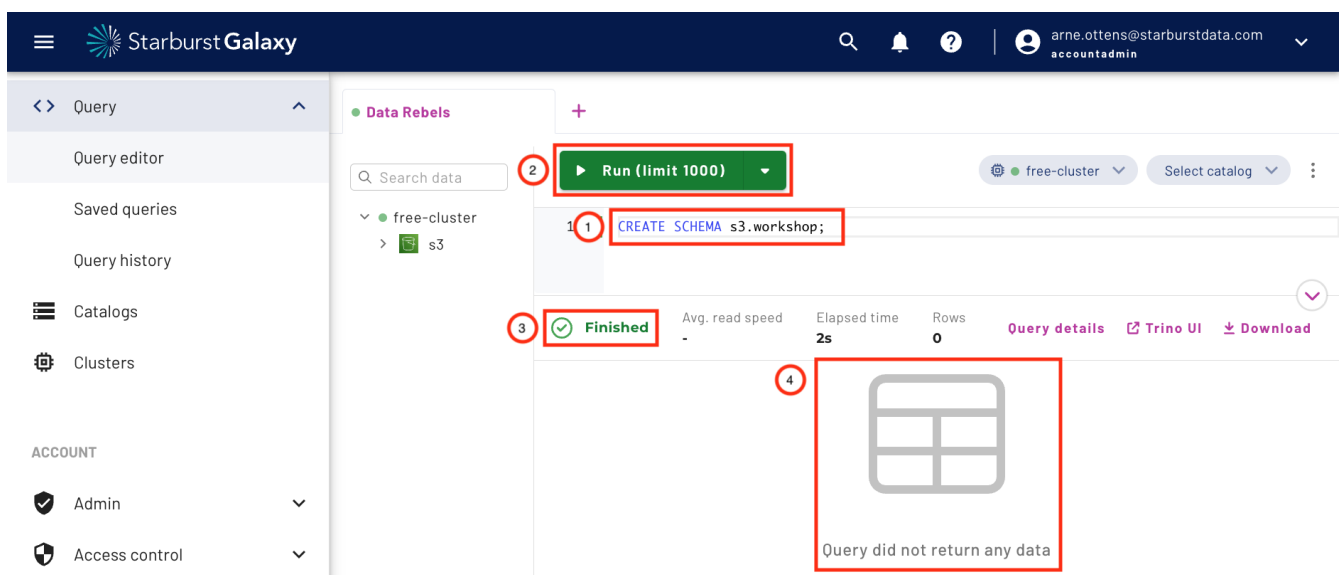
Open Query Editor



1. Click Query Editor on the left (you might have to expand the **Query** menu)
2. Expand your cluster and you should see your Amazon S3 catalog
3. Select your cluster for query execution
4. Click into the editor to write queries

Create data structures

For this workshop we wanted you to start with a clean slate. That is why before we can query the data, you need to create a schema and the table structure first.



1. Enter the **CREATE SCHEMA** statement into the query editor:

```
CREATE SCHEMA s3.workshop;
```

2. Click **[Run (limit 1000)]**. This will execute the query.
3. After the query executed successfully you will see that the query is finished
4. In this case the query did not return any data

Now you can create a table inside this schema. We prepared data for you to use during this workshop. Please enter the below statement in the query editor and execute it the same way you did before.


```

CREATE TABLE s3.workshop.taxi_trips (
  vendor_id bigint,
  pickup_datetime timestamp(3),
  dropoff_datetime timestamp(3),
  passenger_count double,
  trip_distance double,
  rate_code_id bigint,
  store_and_fwd_flag boolean,
  pickup_location_id bigint,
  dropoff_location_id bigint,
  payment_type bigint,
  fare_amount double,
  extra double,
  mta_tax double,
  tip_amount double,
  tolls_amount double,
  improvement_surcharge double,
  total_amount double,
  congestion_surcharge double,
  airport_fee double
)
WITH (
  format = 'ORC',
  type = 'HIVE',
  external_location = 's3://ao-emea-workshop/data/raw'
);

```

This statement creates the table `s3.workshop.taxi_trips_clean` with the defined fields and data types and provides information of the file format (`ORC`), the location where to find the data and the table format (`HIVE`).

Query the data

Now that the data structures are in place you can query the data itself. Here are some sample queries, but feel free to run your own queries as well.

The screenshot shows the Starburst Galaxy web interface. On the left is a sidebar with navigation options: Query, Query editor, Saved queries, Query history, Catalogs, and Clusters. The main area is divided into a top bar with a search icon, a bell icon, a help icon, and a user profile (arne.ottens@starburstdata.com). Below the top bar is a 'Data Rebels' section with a search bar and a list of clusters (free-cluster, s3). A red box highlights the 'Run (limit 1000)' button. Another red box highlights the query editor containing the text 'SELECT * FROM s3.workshop.taxi_trips;'. A third red box highlights the query results table, which shows a 'Finished' status, 'Avg. read speed', 'Elapsed time' of 4s, and 'Rows Limited to 1,000'. The table has columns: vendor_id, pickup_datetime, dropoff_datetime, passenger_count, and trip_distance. The results show 5 rows of taxi trip data.

vendor_id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance
2	2022-12-01 11:51:56.0...	2022-12-01 11:59:33...	1	0.91
2	2022-12-01 11:10:28.0...	2022-12-01 11:27:12.0...	1	3.52
2	2022-12-01 11:06:30.0...	2022-12-01 11:10:54.0...	1	0.31
2	2022-12-01 11:14:35.0...	2022-12-01 11:25:21.0...	1	1.45
2	2022-12-01 11:37:13.0...	2022-12-01 11:50:18.0...	1	1.47

1. Enter your query into the editor
2. Click **[Run (limit 1000)]** to execute the query.
3. See the results.

Sample Queries

```
SELECT * FROM s3.workshop.taxi_trips;

SELECT count(*) FROM s3.workshop.taxi_trips;

SELECT sum(total_amount) FROM s3.workshop.taxi_trips;

SELECT distinct payment_type FROM s3.workshop.taxi_trips;

SELECT *
FROM s3.workshop.taxi_trips
WHERE passenger_count > 1;
```

Create a PostgreSQL Catalog

Now you will add your second data source, a PostgreSQL database. You will be provided with a running instance and respective credentials by Starburst for the duration of one week. The following steps will walk you through the setup of a PostgreSQL Catalog.

Create catalog and select data source

The screenshot shows the Starburst Galaxy interface. On the left sidebar, the 'Catalogs' menu item is highlighted with a red box and a circled '1'. The main panel displays the 'View catalogs' page. At the top of this panel, the 'Create catalog' button is highlighted with a red box and a circled '2'. Below the button, there is a table listing existing catalogs.

Name ↑	Kind	Description	Cloud	Region	Tags
sample	Sample dataset	Sample dataset	aws	Asia Pacific (Tokyo...	No tags assigned.
tpcds	TPC-DS	TPC-DS	-	-	No tags assigned.
tpch	TPC-H	TPC-H	-	-	No tags assigned.

1. Click Catalogs on the left
2. Click [Create catalog]

The screenshot shows the Starburst Galaxy interface with the 'Create catalog' page. On the left sidebar, the 'Catalogs' menu item is highlighted. The main panel displays the 'Select a data source' page. Below the 'Select a data source' heading, there is a grid of data source options. The 'PostgreSQL' option is highlighted with a red box and a circled '3'.

Select a data source

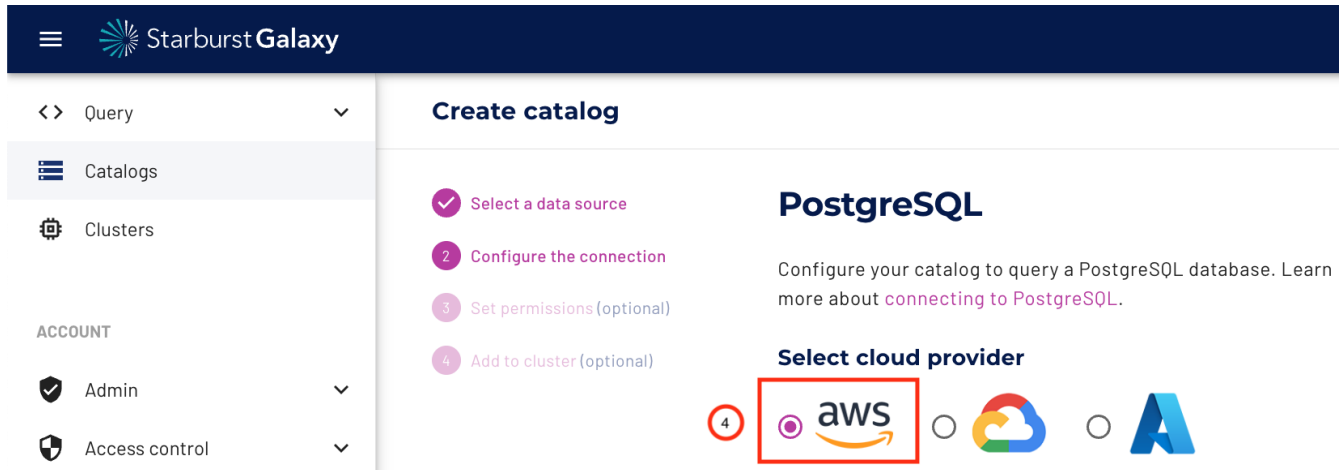
Each catalog contains configuration for Starburst Galaxy to access a data source. Configure catalogs and use them in clusters to query data sources in Starburst Galaxy.

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

Amazon S3	Azure Data Lake Storage	Google Cloud Storage	Tabular
Microsoft SQL Server	MySQL	PostgreSQL	Amazon Redshift

3. Select **PostgreSQL** as a data source

Configure the connection



4. Select **AWS** as the cloud provider

Name and description

Provide a unique name to identify the catalog in your SQL queries in the query editor and other client tools. The namespace for a table is typically <catalog_name>.<schema_name>.<table_name>

Catalog name *

5

?

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

Description

?

5. Give it a **name**. During the workshop we will refer to it as **postgres**

PostgreSQL connection

Connection type *

☒ Connect directly 6 ☐ Connect via SSH tunnel

RDS database host * 7

Port * 8

Database name * 9 ?

RDS master database username * 10 ?

RDS master database password * 11 ?

☒ Use TLS 12 ?


6. Select **Connect directly** as the Connection type
7. RDS database host: `ao-emea-101-workshop.ctjl8sdgduuf.eu-central-1.rds.amazonaws.com`
8. Port: `5432`
9. Database name: `starburst`
10. RDS master database username: `starburst`
11. RDS master database password: `StarburstR0cks!`
12. Activate **Use TLS**. It should be active by default

Test connection

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

Detected regions:

-  Europe (Frankfurt)

 Hooray! You can now add this catalog to a cluster. 14

Test connection 13

[< Back](#)

15 **Connect catalog**

13. Scroll to the end and Click **[Test Connection]**
14. You should see the **Hooray** message, otherwise contact the instructor
15. Click **[Connect catalog]**

Optional Steps

The screenshot shows the 'Create catalog' page in Starburst Galaxy. The left sidebar has 'Catalogs' selected. The main content area shows a progress list with four steps: 'Select a data source', 'Configure the connection', 'Set permissions (optional)', and 'Add to cluster (optional)'. The 'Set permissions' step is active. The 'Catalog-level permissions' section has a toggle for 'Read-only catalog' which is turned on. Below it, the 'Role-level permissions' section shows a dropdown for 'Roles with read access' with 'accountadmin' selected. At the bottom right, there are two buttons: 'Sk 17' and 'Save access controls'.

Starburst Galaxy

<> Query

Catalogs

Clusters

ACCOUNT

Admin

Access control

Cloud settings

Create catalog

- ✓ Select a data source
- ✓ Configure the connection
- 3 Set permissions (optional)
- 4 Add to cluster (optional)

Set permissions

Now that your **postgres** catalog has been created, assign users access with roles. [Learn how to create roles here.](#)

Catalog-level permissions

16 ☒ Read-only catalog

Prohibits all users, **including the catalog owner**, from modifying data or metadata in this catalog.

Role-level permissions

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

Sk 17 Save access controls

16. Enable **Read-only catalog**

17. Click **[Save access controls]**

The screenshot shows the 'Create catalog' page in Starburst Galaxy. The left sidebar has 'Catalogs' selected. The main content area shows a progress list with four steps: 'Select a data source', 'Configure the connection', 'Set permissions (optional)', and 'Add to cluster (optional)'. The 'Add to cluster' step is active. The 'Add to cluster' section has a dropdown menu for 'Select clusters' with 'free-cluster' selected. Below the dropdown, there is a checkbox for 'free-cluster (eu-central-1)' which is checked. At the bottom right, there are two buttons: 'Sk 19' and 'Add to cluster'.

Starburst Galaxy

<> Query

Catalogs

Clusters

ACCOUNT

Admin

Access control

Cloud settings

Create catalog

- ✓ Select a data source
- ✓ Configure the connection
- ✓ Set permissions (optional)
- 4 Add to cluster (optional)

Add to cluster

Attach your **postgres** 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

18 Select clusters

free-cluster

☒ free-cluster (eu-central-1)


Sk 19 Add to cluster

18. Select your cluster from the **Add to cluster** dropdown menu

19. Click **[Add to Cluster]**

Catalog added

×

 Hooray! Your catalog has been added.

Would you like to **add more catalogs**, or are you ready to **start querying your data**?

All currently running queries will complete on the existing cluster and all new queries will be redirected to the new configuration as soon as it is ready.

[Add more catalogs](#)[Query my data](#)

20. You should see the **Hooray** message, otherwise contact the instructor

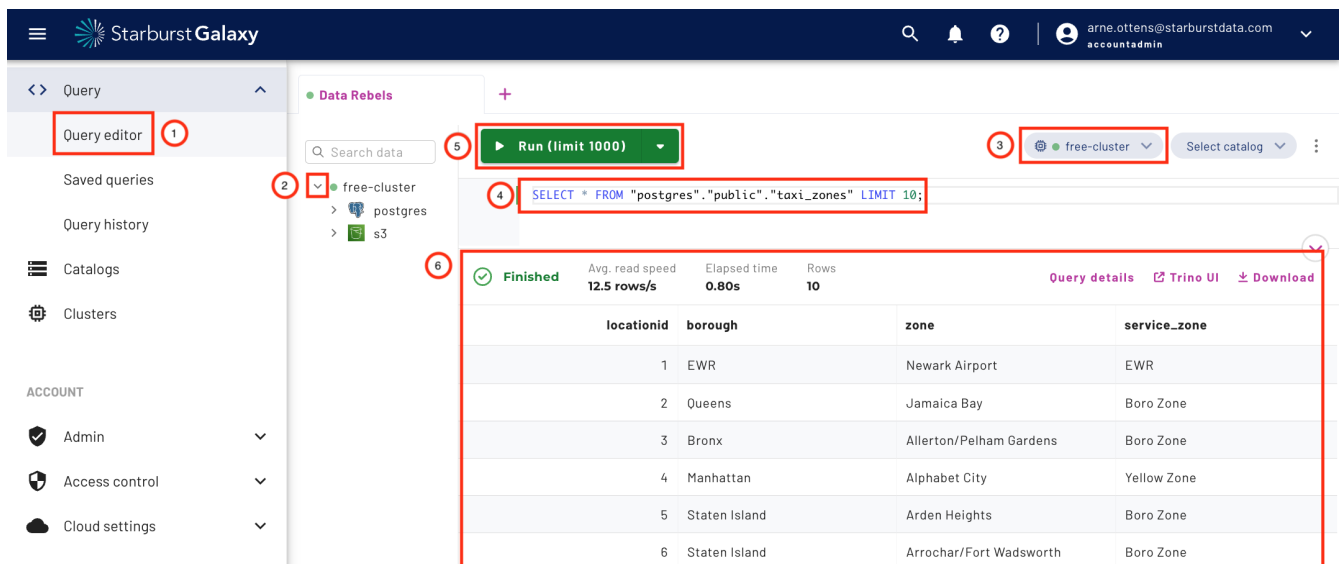
Congratulations! You created your second data source in Starburst Galaxy.

Federate Amazon S3 and PostgreSQL

You can query the data inside PostgreSQL right away, there is no need to create the structures as they are exposed by the database directly. The cool thing is that you can now use both data sources as part of the same query. The following steps will guide you through first querying PostgreSQL itself and then join it with the data in Amazon S3.

Query PostgreSQL alone

As there is no need to create data structures like you did for the Amazon S3 catalog, you can query the data in the database straight away. You will find some sample queries down below.



The screenshot shows the Starburst Galaxy interface. On the left sidebar, 'Query editor' is highlighted with a red box and number 1. In the center, 'Data Rebels' is expanded with a red box and number 2, showing 'postgres' and 's3' catalogs. The 'free-cluster' is selected with a red box and number 3. The 'Run (limit 1000)' button is highlighted with a red box and number 5. The query editor contains the query 'SELECT * FROM \"postgres\".\"public\".\"taxi_zones\" LIMIT 10;' highlighted with a red box and number 4. The results table, labeled 'Finished', shows 10 rows of data with columns: locationid, borough, zone, and service_zone. The table is highlighted with a red box and number 6.

locationid	borough	zone	service_zone
1	EWB	Newark Airport	EWB
2	Queens	Jamaica Bay	Boro Zone
3	Bronx	Allerton/Pelham Gardens	Boro Zone
4	Manhattan	Alphabet City	Yellow Zone
5	Staten Island	Arden Heights	Boro Zone
6	Staten Island	Arrochar/Fort Wadsworth	Boro Zone

1. Click Query Editor on the left (you might have to expand the **Query** menu)
2. Expand your cluster and you should see your PostgreSQL catalog
3. Select your cluster for query execution
4. Click into the editor to write queries
5. Click **[Run (limit 1000)]** to execute the query
6. See the results

Sample Queries

```
SELECT * FROM postgres.public.taxi_zones;

SELECT DISTINCT borough FROM postgres.public.taxi_zones;

SELECT borough, count(*)
FROM postgres.public.taxi_zones
GROUP BY 1
ORDER BY 2 DESC;
```

Join Amazon S3 data with PostgreSQL data

Now let's get into query federation. Query Federation let's you join data that lives in different systems as if they are part of the same data source.

The screenshot shows the Starburst Galaxy web interface. On the left is a navigation sidebar with options like 'Query editor' (highlighted with a red box and a circled '1'), 'Saved queries', 'Query history', 'Catalogs', 'Clusters', and 'ACCOUNT' settings. The main area is divided into a 'Data Rebels' section with a search bar and a 'Run (limit 1000)' button (highlighted with a red box and a circled '3'). Below this is a SQL query editor (highlighted with a red box and a circled '2') containing the following query:

```
SELECT tz.borough, sum(tt.passenger_count)
FROM s3.workshop.taxi_trips_clean tt,
postgres.public.taxi_zones tz
WHERE tt.pickup_location_id = tz.locationid
GROUP BY 1
ORDER BY 2 DESC;
```

Below the query editor, the results of the query are displayed (highlighted with a red box and a circled '4'). The status is 'Finished' with an average read speed of '4.6M rows/s' and an elapsed time of '8s'. The results are shown in a table with columns 'borough' and '_col1'.

borough	_col1
Manhattan	47612507
Queens	4983823
Unknown	690133
Brooklyn	284413

1. Go to the Query Editor
2. Enter a query that joins both Amazon S3 and PostgreSQL (sample below)
3. Click **[Run (limit 1000)]** to execute the query
4. See the results

Sample Query

```
SELECT tz.borough, sum(tt.passenger_count)
FROM s3.workshop.taxi_trips tt,
postgres.public.taxi_zones tz
WHERE tt.pickup_location_id = tz.location_id
GROUP BY 1
ORDER BY 2 DESC;
```