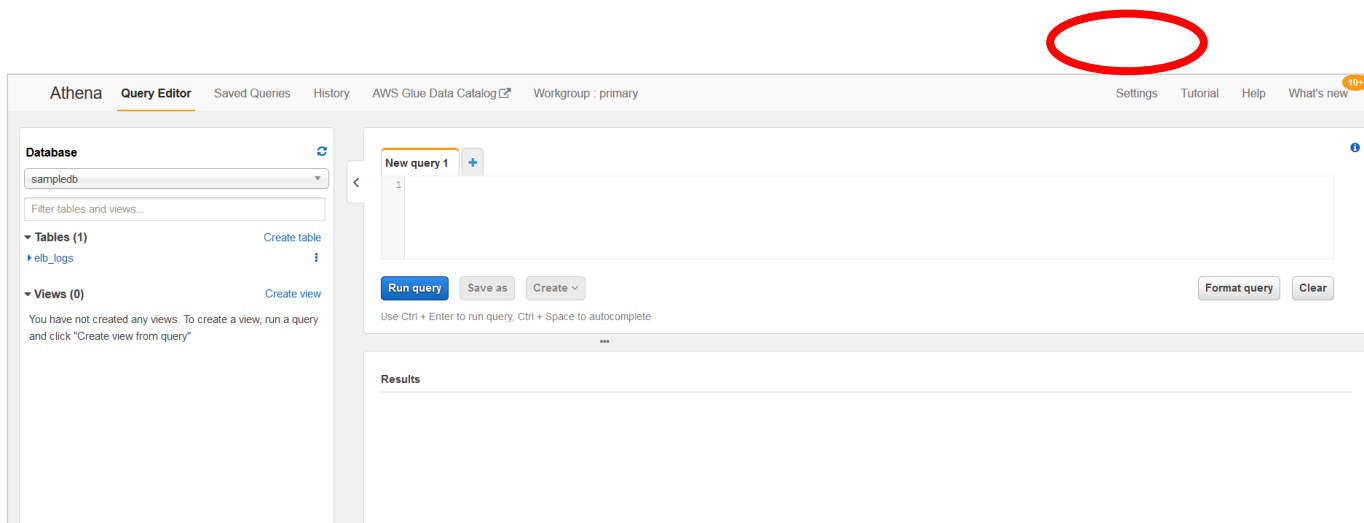


Section 2

Connecting Athena to the NOAA data repository

In this section, you will create an Athena table to hold data from the NOAA data repository on Registry of Open Data on AWS. Some of these directions require the use of SQL statements. These statements can be found in the workshop GitHub repo in the [sql-statements.sql](#) file. In that file you will see queries divided by section number (2 in this case) followed by a decimal number. The decimal number represents the specific step in the instructions that aligns with that statement.

- 1) Log on to the AWS console and change your region to N. Virginia.
- 2) Search for Athena and select that service.
- 3) If you encounter the “Get Started” page, click the “Get Started” button to go to the Athena Dashboard page.
- 4) First, you need to specify the location that Athena can store the query results. Click on the Settings tab:



- 5) Enter the bucket name and folder you created for the query results in previous section. Because bucket names must be globally unique, you'll have to use the exact name of the one you created. If you followed the directions in Section 1, the folder should be called “query-results”.

Settings

Settings apply by default to all new queries. [Learn more](#)

Workgroup: primary

Query result location

Example: s3://query-results-bucket/folder/

Encrypt query results ☐

Autocomplete ☐

Cancel

Save

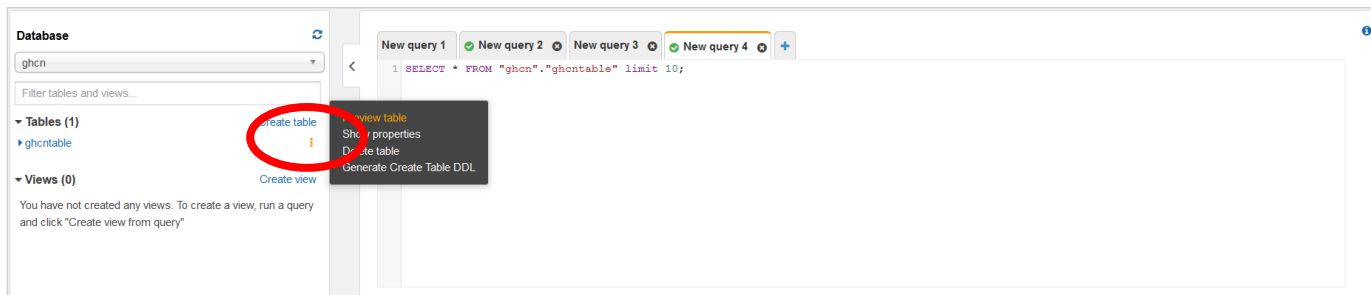
- 6) Create a database to put your tables in. In the query editor, type:

```
CREATE DATABASE ghcn
```

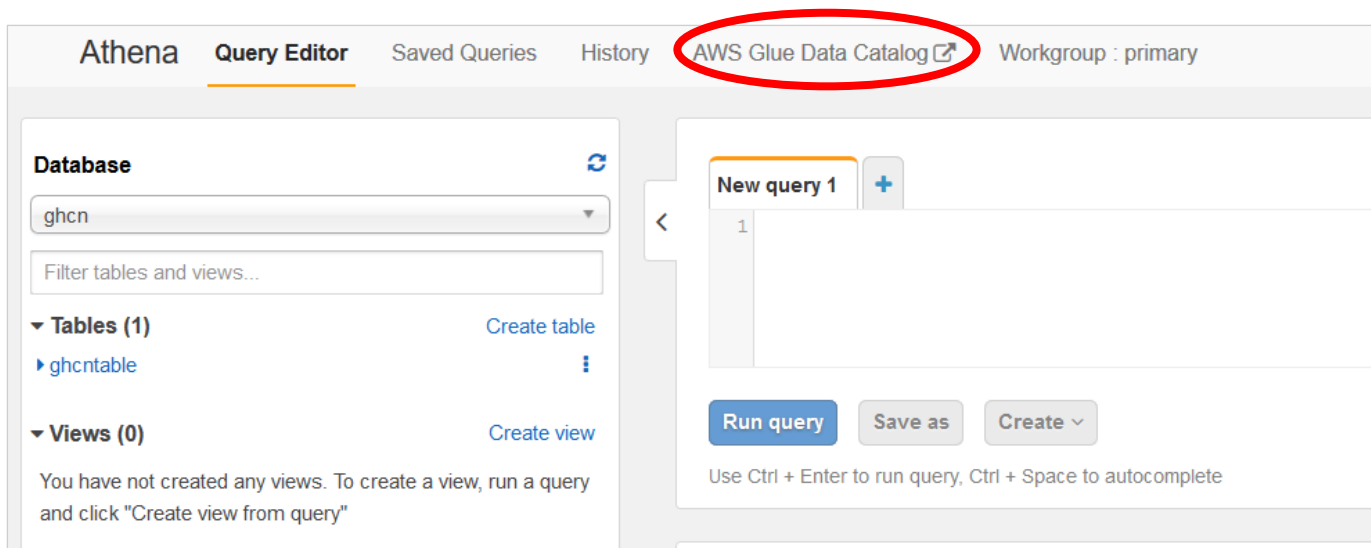
- 7) Click “Run Query”. You should see a confirmation in the Results section of “Query successful”. You can also press Ctrl+Enter to run the query.
- 8) On the left-hand side, change the database to the database you just created (“ghcn”), click on the “+” sign to create a new query. Paste this query into the query editor:

```
CREATE EXTERNAL TABLE ghcntable (
  id string,
  year_date string,
  element string,
  data_value string,
  m_flag string,
  q_flag string,
  s_flag string,
  obs_time string)
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://noaa-ghcn-pds/csv'
TBLPROPERTIES (
  'has_encrypted_data'='false',
  'transient_lastDdlTime'='1572285230')
```

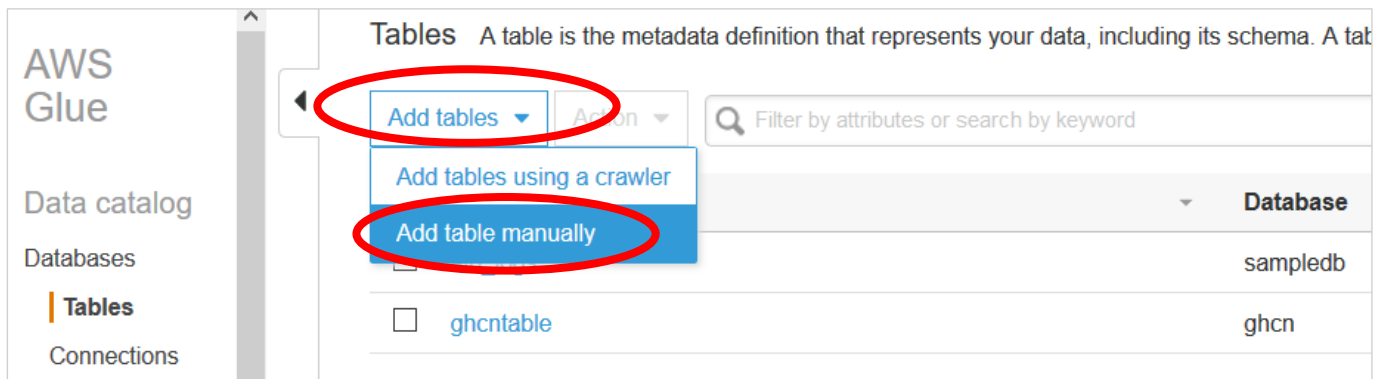
- 9) You will see the new table “ghcntable” on the left-hand side under Tables. Click on the three vertical dots, and select “Preview table”. This will create a new query to show you the first 10 results in the table. Notice you did not have to copy or manipulate the data, prior to executing a SQL query against the repository in S3. You are connecting to the noaa-ghcn-pds dataset that resides in another account, using the Athena SQL interface.



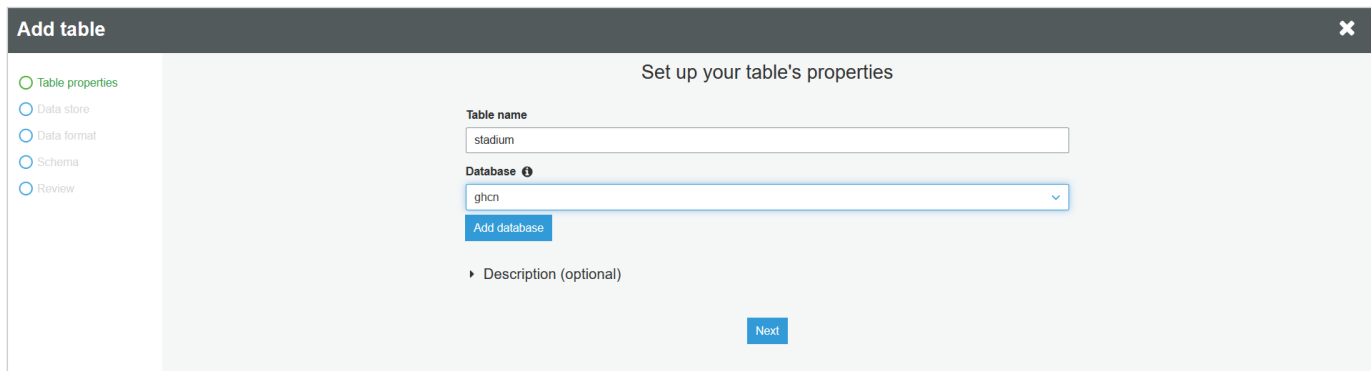
- 10) In the previous section, we copied the file [stadiums_with_stations_global.csv](#) into the S3 bucket. Now we’re going to create a table using this data as the source data. This file contains the list of potential sites for the Deep Racer event, the latitude and longitude of the site as well as sensor id at that location. From the Athena page, click on AWS Glue Data Catalog:



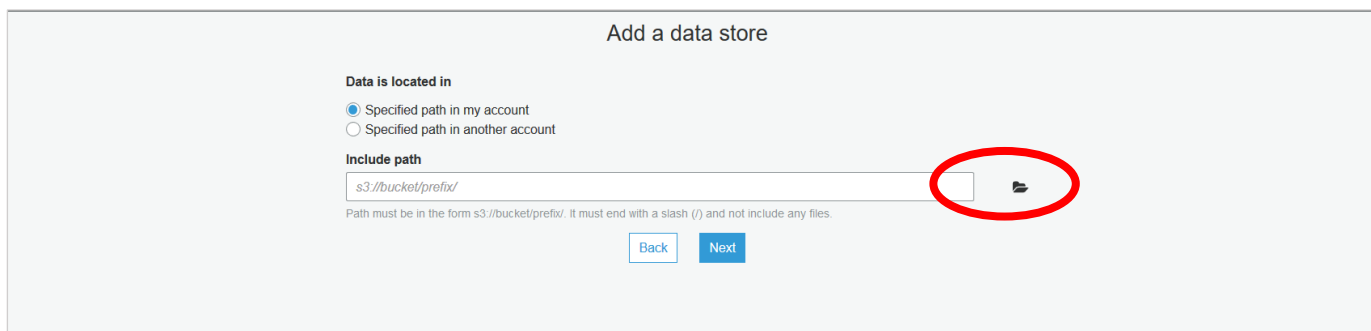
- 11) From the AWS Glue console, click “Add tables”, then “Add table manually”:



12) Enter the table name “stadium”, select the “ghcn” database, and click “Next”:



13) Select “Specified path in my account” and either type in the location of your S3 bucket and folder where you saved the stadium data OR click on the folder to browse to that folder, then click “Next”.





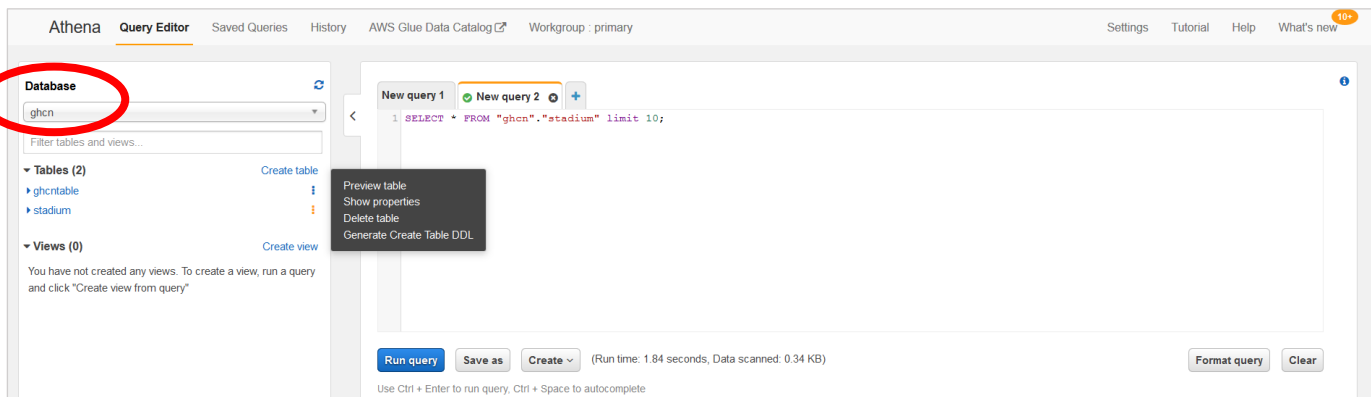
- 14) Specify the Classification as “CSV” and the delimiter as “Comma”. Click “Next”.
- 15) Click on the “Add column” button and add `city` as a column ,with column type “String”. Repeat the process for the following columns, all as column type “String”, then click “Next”.

`station_id`

`lat`

`lon`

- 16) Click “Finish”.
- 17) Change the database in the dropdown on the left-hand side to “ghcn”. Navigate back to the Athena console, locate the “stadium” table, click the three vertical dots and select “Preview Table”.



18) In the Query Editor, enter the following text to test a join across two S3 buckets in two different accounts:

```
SELECT city, station_id,element, data_value
FROM stadium
INNER JOIN ghcntable
ON stadium.station_id = ghcntable.id
WHERE ghcntable.year_date >= '20191029'
AND (
    ghcntable.element = 'TMIN' OR ghcntable.element = 'TMAX')
```

19) Click “Run Query”. You should see the results of the query.

In this section you created an Athena database, with two tables. One, the stadium table, was built using Glue and a CSV file in an S3 bucket you created. The other table was created with a SQL statement, referencing another S3 bucket, in a different account. That S3 bucket is updated daily with new weather data for its global sensors.

You’ve now completed section 2 of the workshop and can move on to the next session, [“Creating the endpoint and querying Athena.”](#)