Table of contents

Introduction to Athena :

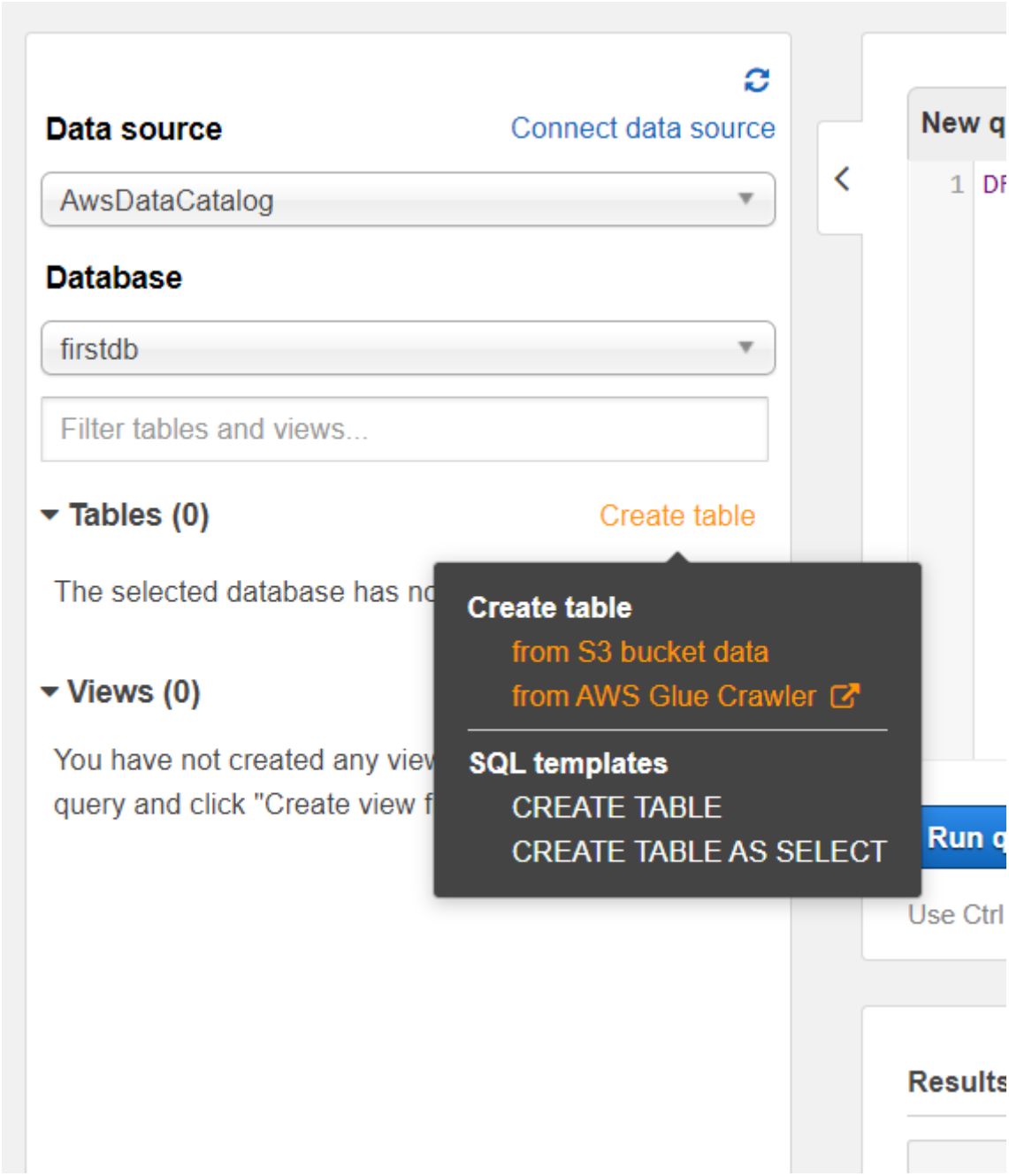
- Athena is one of the relatively newer service from AWS which has been gaining popularity .
- Athena is a serverless querying engine which helps us query data residing in S3
- Athena is fast and easy to setup . It takes away the need of loading data in database seperately .
- Since athena charges on per query (5$ for every TB scanned), it makes it perfect option for ad-hoc queries

Table in Athena

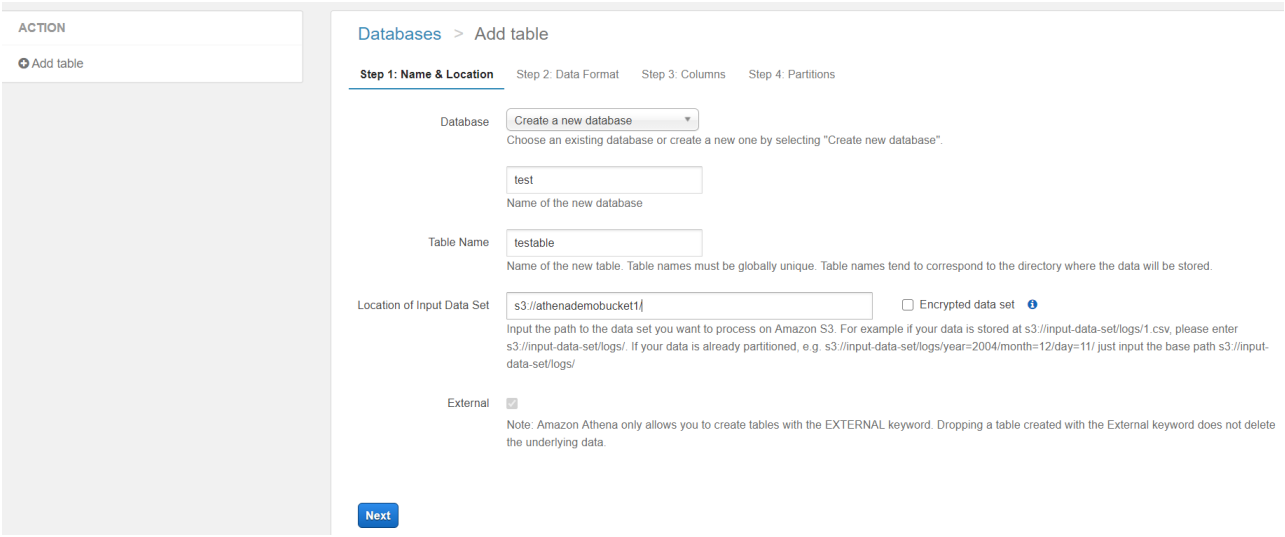
- All tables created in athena are external tables . This means the schema of the table and actual data will be loosely coupled. Even if the schema is dropped , it will not impact the data

Creating table in athena

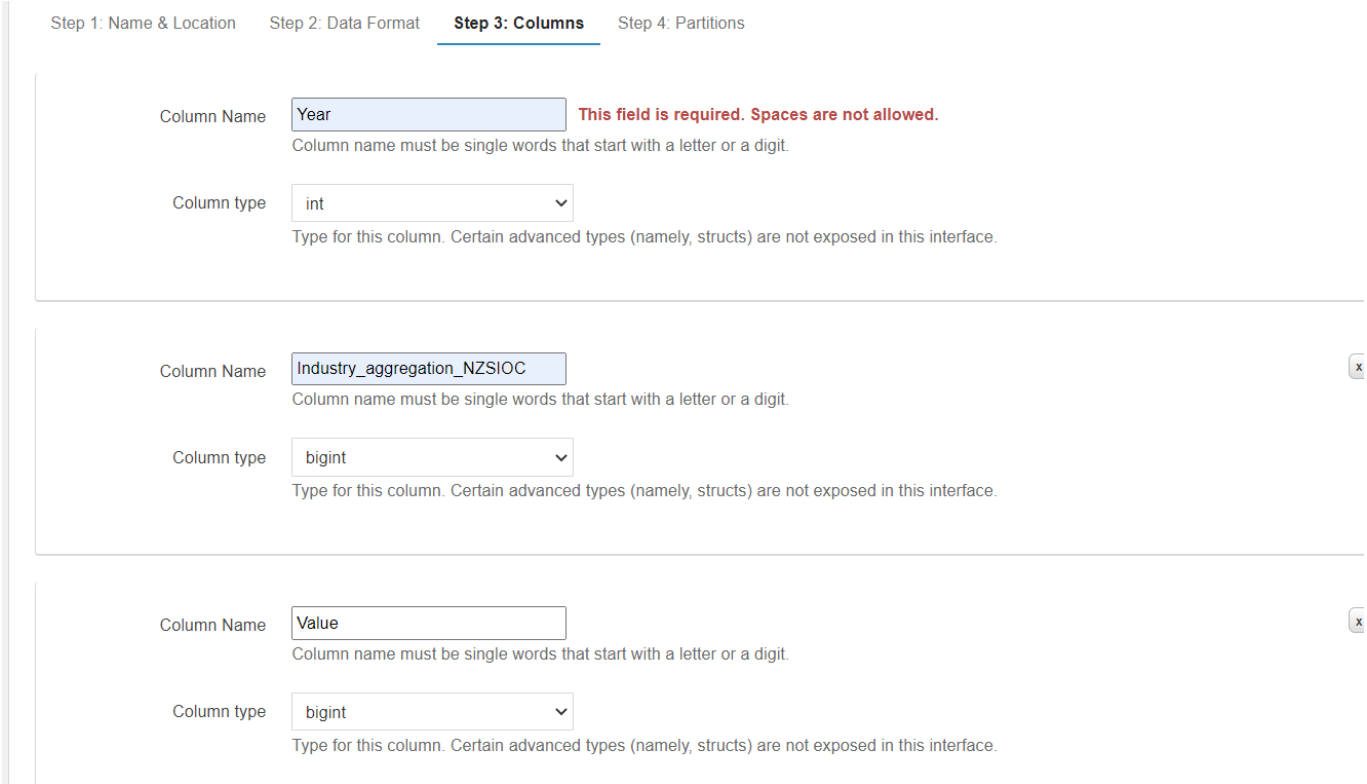
- There various ways using which we can create a table in Athena . For ex console UI, using a SQL query or using glue crawler .
- the console UI uses the query itself in the background .
- Before we create the table however , we need to have data in S3 . There are various file types that are supported , we'll use a sample csv file that we have . Upload it to any S3 bucket. Once uploaded , copy the object path . It will look something like "s3://athenademobucket1/annual-enterprise-survey-2017-financial-year-provisional-csv"
- Navigate to athena , click on get started .
- On left hand side , you'll see create table . Click on it and select create table from S3 bucket data



- It will pop out a UI which will help us create the table
- You can chose to create a new database or use an existing one . give the table name
- Under location paste the object path we just copied , and remove just the object name from it . It will look something like "s3://athenademobucket1/"



- Click next
- Select the data format as CSV as our file is in that format and click next
- In the 3rd setp we have to add column names and their data types . Open the sample csv where you can see there are 3 columns named as
    - Year int
    - Industry_code_NZSIOC bigint
    - Value bigint
- Give the column names and data types as input in the UI and click next



- As of now , we are not going to add partitions so we'll skip this step and click on create table
- This will create a query in the background and create the table
- Once created you can start querying
- Everytime you query start observing the amount of data scanned .

Creating partitions in athena

- Since athena charges on based on the amount of data that is scanned , cost optimization will include any methodology which reduces the amount of scanned data
- There are 2 approaches one can take for it
    - Using compressed file format
        - Generally the parquet and ORC file formats are smaller in size as compared to CSV . Using them will save money both at S3 storage level and Athena qeury level
        - Creating partitions
            - Partition helps us define chunks of data . During a qeury if whole dataset is getting scanned , it will be very expensive for large datasets
            - If we enable partition on a specific column for ex Year , and include that as a filter when we query . Then the athena engine will only search for that partition and will skip the rest of the data
            - Pre-requisite of to enable partition is the S3 folder structure should be present according to the partioned column
- We'll take the similar dataset , but this dataset will be divided on basis of year 2013,2014, 2015.
- Create folders in S3 as "year=2014","year=2013" etc
- Keep the files in appropriate folders
- Let us create a new table for this dataset .
- Click on create table from S3 bucket as usual . Give it table name
- While giving location , give it to similar what we did previously for ex : s3://bucketname/
- This is where your Year folders are residing
- Click next and select the datatype as csv and click next
- For columns just give all columns other than Year which is our partitioned column . Click next
- Click on add partition . Here let us add Year as the partitioned column
- Click on create table
- For partitioned tables , we cannot directly start querying , we need to load the partitions . There are two ways to load a partition
    - Click on the table name options and click on load partitions
    - Use below query

```
ALTER TABLE elb_logs_raw_native_part ADD PARTITION (Year=2015) location
's3://athena-examples/elb/plaintext/2015/'
```

- Once the partitions are loaded , you can start executing queries
- Notice the data scanned difference between normal select * queries and queries where you add the partitioned column as filter

```
select * from test where year=2014
```

- The data scanned on a partitioned table when the partitioned column is included in the query filter will be much lesser than non partitioned table

FAQ

- Not receiving any data in the query result
    - Verify the location specified while creating the table
- Getting garbage data in the query result
    - The data type mentioned while creating table or adding partition needs to be checked
- Unable to execute query as the query result location is not set
    - On the top right corner of the screen click on settings
    - In the setttings specify any S3 bucket location where athena will keep the result set