# Lab 4: Working with Amazon Athena

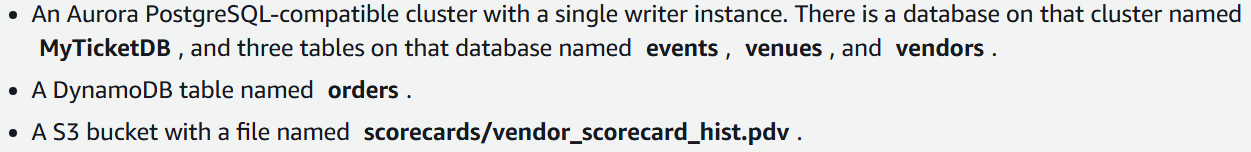
Objs-

1. Create databases and tables in Athena to query S3 objects.
2. Query data stored in S3 objects using Athena..
3. Query data from Aurora using Athena.
4. Query data from DynamoDB using Athena federated queries.
5. Access, join, and analyze data across Amazon S3, Aurora, and DynamoDB sources.

Athena features-

* Serverless: Analysts don't have to worry about managing servers.
* SQL Support: It allows running SQL queries on data from various sources like Amazon S3, DynamoDB, and relational databases.
* Data Integration: It can combine data from different sources to provide useful insights.

Pre-set:



**Task 1 - create a database and table that uses a file stored in an S3**

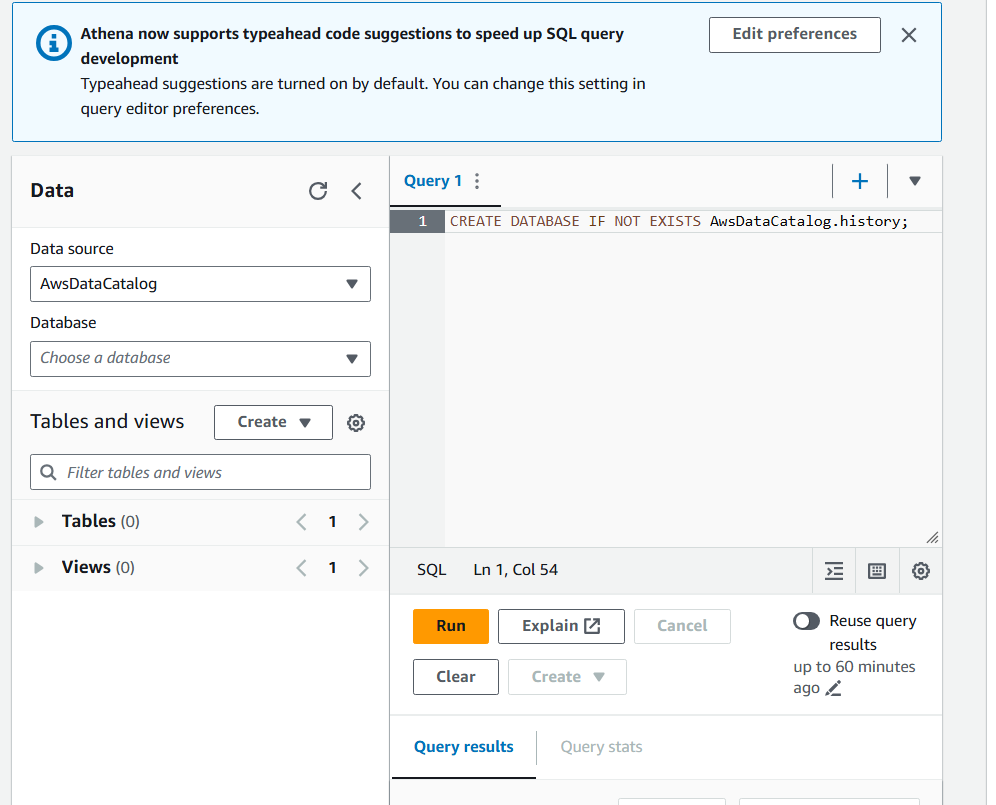
We also validate that the data is accessible via Athena

1.1 Open athena in console and launch query editor option.

1.2 Screenshot missed -

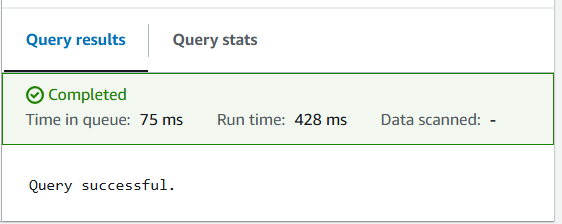
A pop-up will appear showing the path of S3 bucket, acknowledge it.

This location is also given in lab manual

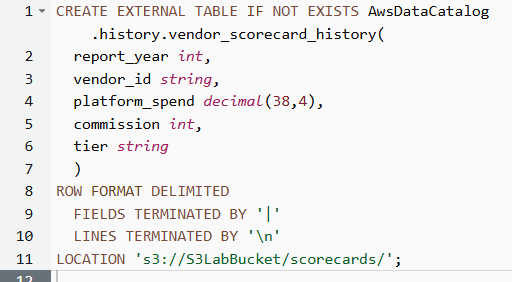


1.3 run this query

CREATE DATABASE IF NOT EXISTS AwsDataCatalog.history;

1.4 

1.5 replace bucket value and run this

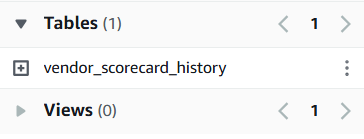


external table in Athena means the data isn't stored in Athena itself but remains in S3. Athena only stores the metadata (the schema).

's3://S3LabBucket/scorecards/':

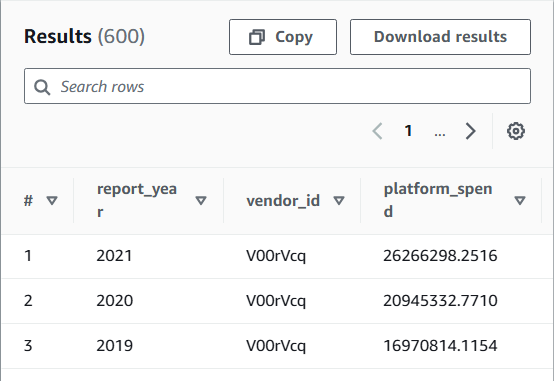
Specifies the location in Amazon S3 where the data files are stored. In this case, the data files are in the scorecards directory of the S3LabBucket bucket.

1.6 new table created



1.7 view the table using

SELECT \* FROM AwsDataCatalog.history.vendor\_scorecard\_history;

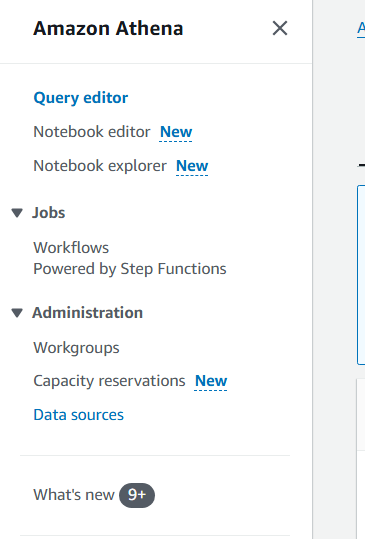


**Task 2: Setup Athena Federated Query access for the Event Management database**

**From aurora postgreSQL**

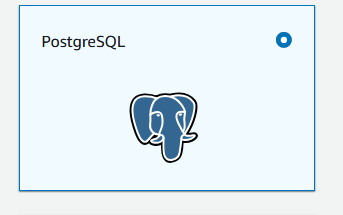
In this task, you create an Athena data source that connects to an the AnyCompany event management database, which consists of multiple tables in a Aurora PostgreSQL database.

2.1 click on data sources

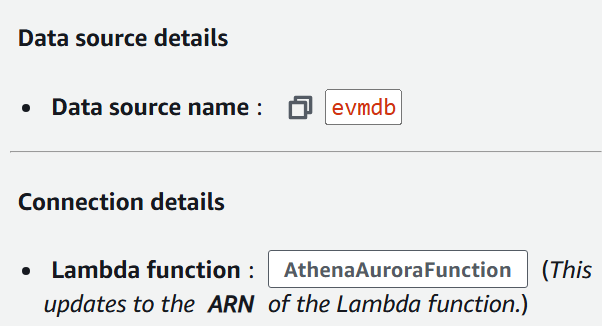


2.2 click on create data source

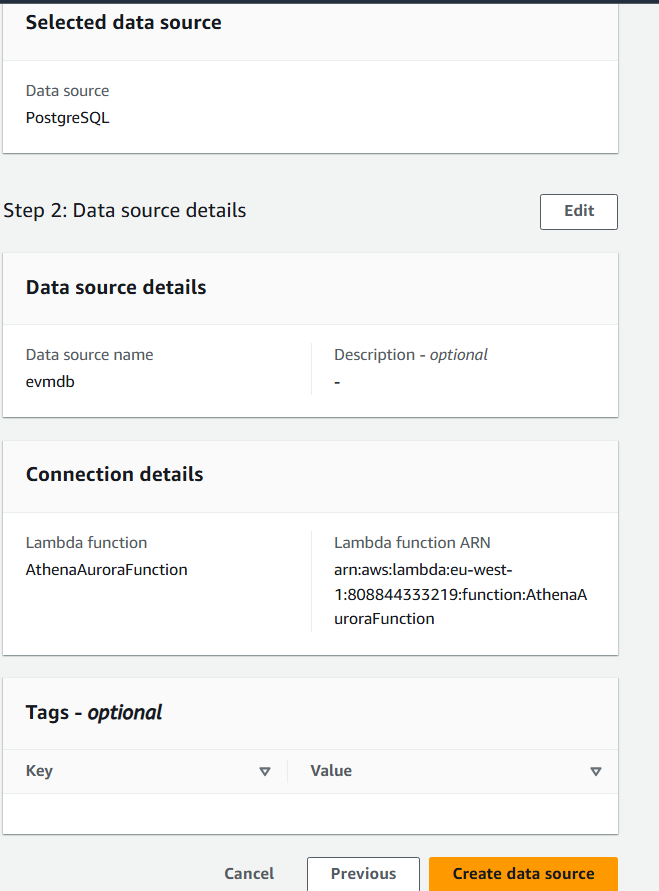
2.3 choose this



2.4 fill these data source details

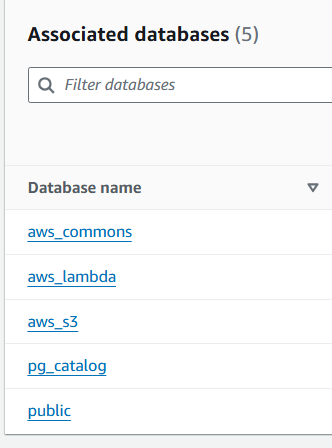


2.5 choose next

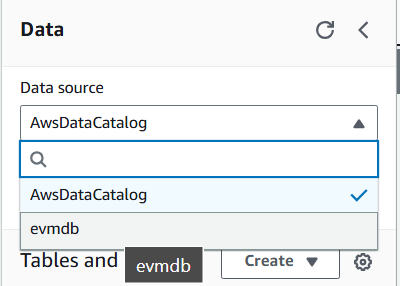


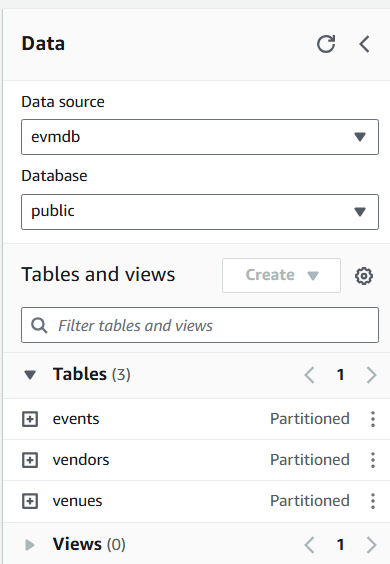
2.6 

2.7 in the data source -



2.8 return to query editor, and change data source

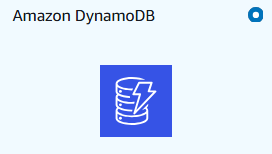


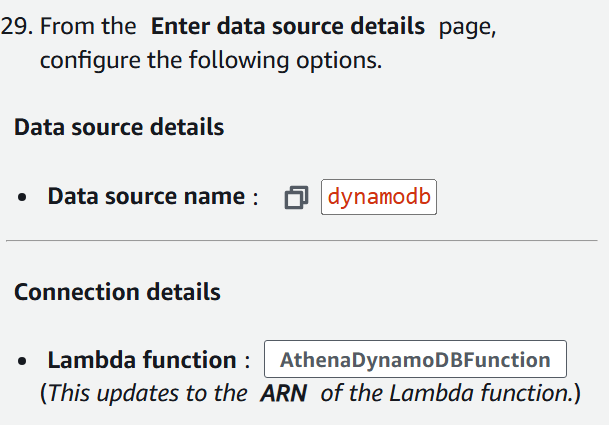
2.9 these tables show up in the public DB of this DS

**Task 3: Setup Athena Federated Query access for the Order Management table**

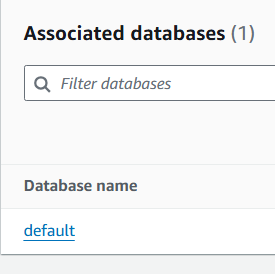
**From DynamoDB**

3.1 repeat steps and create a new data source:

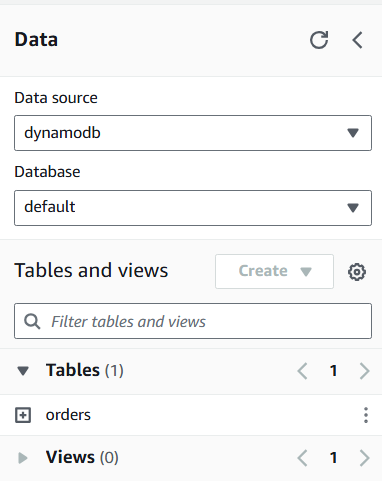




3.2 databases in this



3.3 orders table added



**Task 4: Run SQL analytical query**

4.1 run this command

| WITH  vendor\_scorescard as (  select \*,  case  when report\_year = 2021 then platform\_spend  end as "spend(Y-1)",  case  when report\_year = 2021 then commission  end as "comm(Y-1)",  case  when report\_year = 2020 then platform\_spend  end as "spend(Y-2)",  case  when report\_year = 2020 then commission  end as "comm(Y-2)",  case  when report\_year = 2019 then platform\_spend  end as "spend(Y-3)",  case  when report\_year = 2019 then commission  end as "comm(Y-3)"  from AwsDataCatalog.history.vendor\_scorecard\_history  where report\_year between 2019 and 2021  ),  vendor\_scorescard\_last\_3y as (  select vendor\_id,  max("spend(Y-1)") as "spend(Y-1)",  max("spend(Y-2)") as "spend(Y-2)",  max("spend(Y-3)") as "spend(Y-3)",  max("comm(Y-1)") as "comm(Y-1)",  max("comm(Y-2)") as "comm(Y-2)",  max("comm(Y-3)") as "comm(Y-3)"  from vendor\_scorescard  group by vendor\_id  ),  current\_year\_spend as (  select  vendors."vendor\_id",  coalesce(sum(orders."salePrice" \* orders."orderQty"),0.00) as "spend(CurrYear)",  vendors.commission as "comm(CurrYear)" ,  vendors."vendor\_name" as "name",  vendors."pterm" as "terms",  vendors."status" as "status"  from  ( evmdb.public.vendors as vendors  left outer join evmdb.public.events as events ON (vendors."vendor\_id" = events."vendor\_id") )  left outer join dynamodb.default.orders as orders ON (orders."eventSku" = events."event\_sku")  group by vendors."vendor\_id", vendors.commission, vendors."vendor\_name",  vendors."pterm", vendors."status"  )  select  curr."vendor\_id",  curr."spend(CurrYear)",  vendors\_hist."spend(Y-1)",  vendors\_hist."spend(Y-2)",  vendors\_hist."spend(Y-3)",  curr."comm(CurrYear)",  vendors\_hist."comm(Y-1)",  vendors\_hist."comm(Y-2)",  vendors\_hist."comm(Y-3)",  curr."name",  curr."terms",  curr."status"  from current\_year\_spend as curr  left outer join vendor\_scorescard\_last\_3y as vendors\_hist ON (vendors\_hist."vendor\_id" = curr."vendor\_id"); |
| --- |

This query combines historical spend and commission data from

1. AwsDataCatalog.history.vendor\_scorecard\_history,
2. evmdb.public.vendors, evmdb.public.events, and
3. dynamodb.default.orders

into a comprehensive report with details on spend, commission, vendor name, payment terms, and status.  
  
Basically, it combines data from all three data sources, to run the query

