

## AWS Glue and Athena

**AWS Glue:** AWS Glue is a serverless data integration service that makes it easy for analytics users to discover, prepare, move, and integrate data from multiple sources. You can use it for analytics, machine learning, and application development. It also includes additional productivity and data ops tooling for authoring, running jobs, and implementing business workflows.

With AWS Glue, you can discover and connect to more than 70 diverse data sources and manage your data in a centralized data catalog. You can visually create, run, and monitor extract, transform, and load (ETL) pipelines to load data into your data lakes. Also, you can immediately search and query cataloged data using Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum.

AWS Glue consolidates major data integration capabilities into a single service. These include data discovery, modern ETL, cleansing, transforming, and centralized cataloging. It's also serverless, which means there's no infrastructure to manage. With flexible support for all workloads like ETL, ELT, and streaming in one service, AWS Glue supports users across various workloads and types of users.

Also, AWS Glue makes it easy to integrate data across your architecture. It integrates with AWS analytics services and Amazon S3 data lakes. AWS Glue has integration interfaces and job-authoring tools that are easy to use for all users, from developers to business users, with tailored solutions for varied technical skill sets. [Learn More](#)

**Athena:** Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon Simple Storage Service (Amazon S3) using standard SQL. With a few actions in the AWS Management Console, you can point Athena at your data stored in Amazon S3 and begin using standard SQL to run ad-hoc queries and get results in seconds.

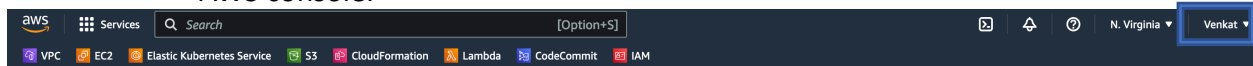
Amazon Athena also makes it easy to interactively run data analytics using Apache Spark without having to plan for, configure, or manage resources. When you run Apache Spark applications on Athena, you submit Spark code for processing and receive the results directly. Use the simplified notebook experience in Amazon Athena console to develop Apache Spark applications using Python or Athena notebook APIs.

Athena SQL and Apache Spark on Amazon Athena are serverless, so there is no infrastructure to set up or manage, and you pay only for the queries you run. Athena scales automatically—running queries in parallel—so results are fast, even with large datasets and complex queries. [Learn more.](#)

## AWS Glue and Athena

### ***Points to Remember before doing this Lab:***

1. Please ensure that you attach all the screenshots labeled with “**Note: This is a Deliverable**” under each of them.
2. When capturing each screenshot, be certain that your AWS account name (It should be your NETId) is visible. Locate the name at the top-right corner of your AWS console.



### ***Pre-requisite for this lab:***

- a.) Your file and folder names should start with <name>\_<resource\_name>.
- b.) All labs must be performed in US East (N.Virginia) us-east-1 region.

### **Learning Outcome:**

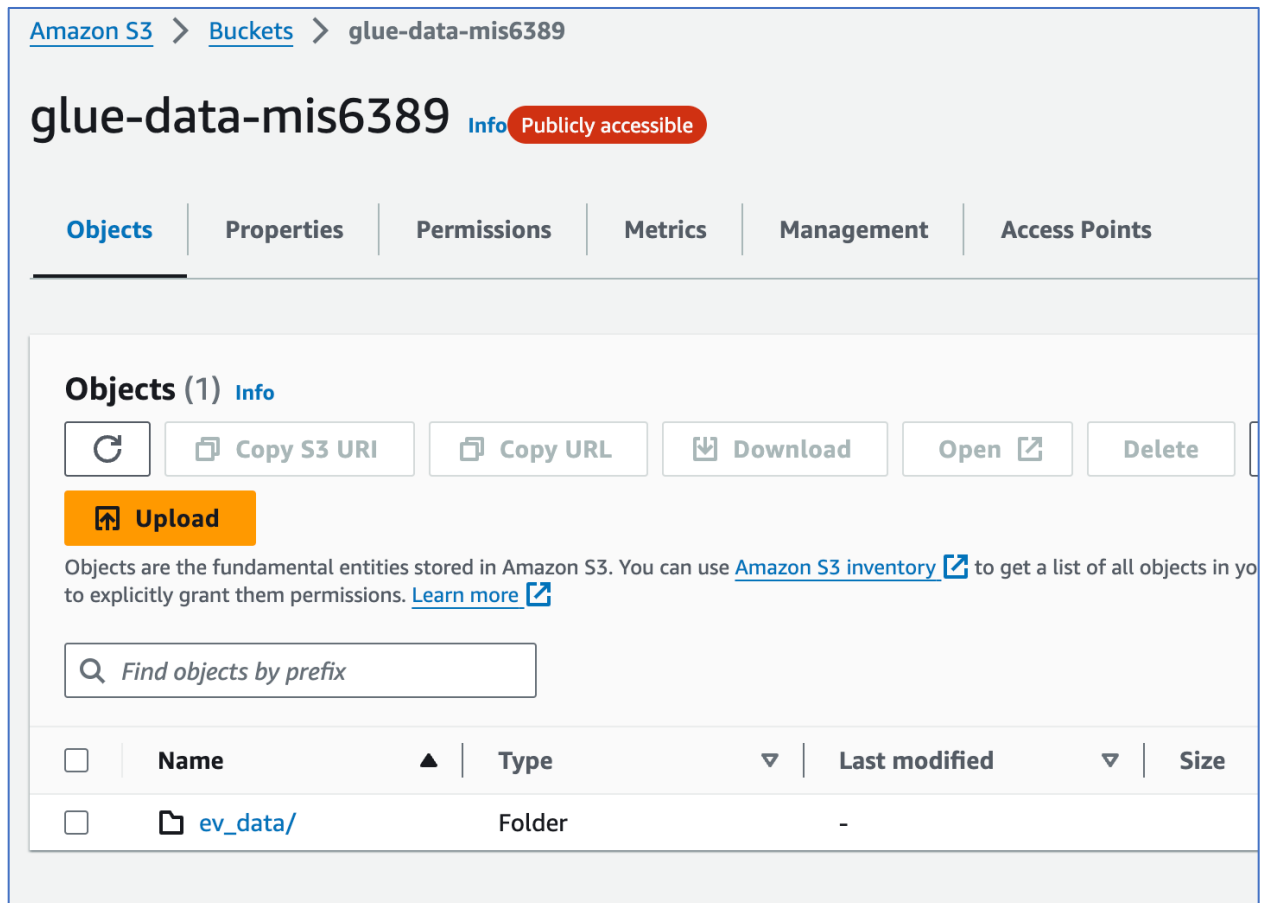
- How to define a database
- How to configure a crawler to explore data in Amazon S3 Bucket
- How to create tables
- How to query data with Amazon Athena

Let's get started...

Steps:

We are creating two buckets.


First one: Create the bucket name as given and then a folder inside it.



Upload the file in the ev\_data folder: [https://glue-data-mis6389.s3.amazonaws.com/ev\\_data/Electric\\_Vehicle\\_Population\\_Data.csv](https://glue-data-mis6389.s3.amazonaws.com/ev_data/Electric_Vehicle_Population_Data.csv)

Make sure the permissions tab looks as shown below. Unblock the public access and paste the policy:

## AWS Glue and Athena

**Block *all* public access**  
 Off  
► Individual Block Public Access settings for this bucket

**Bucket policy**

EditDelete

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

✔ Bucket policy copied

Copy

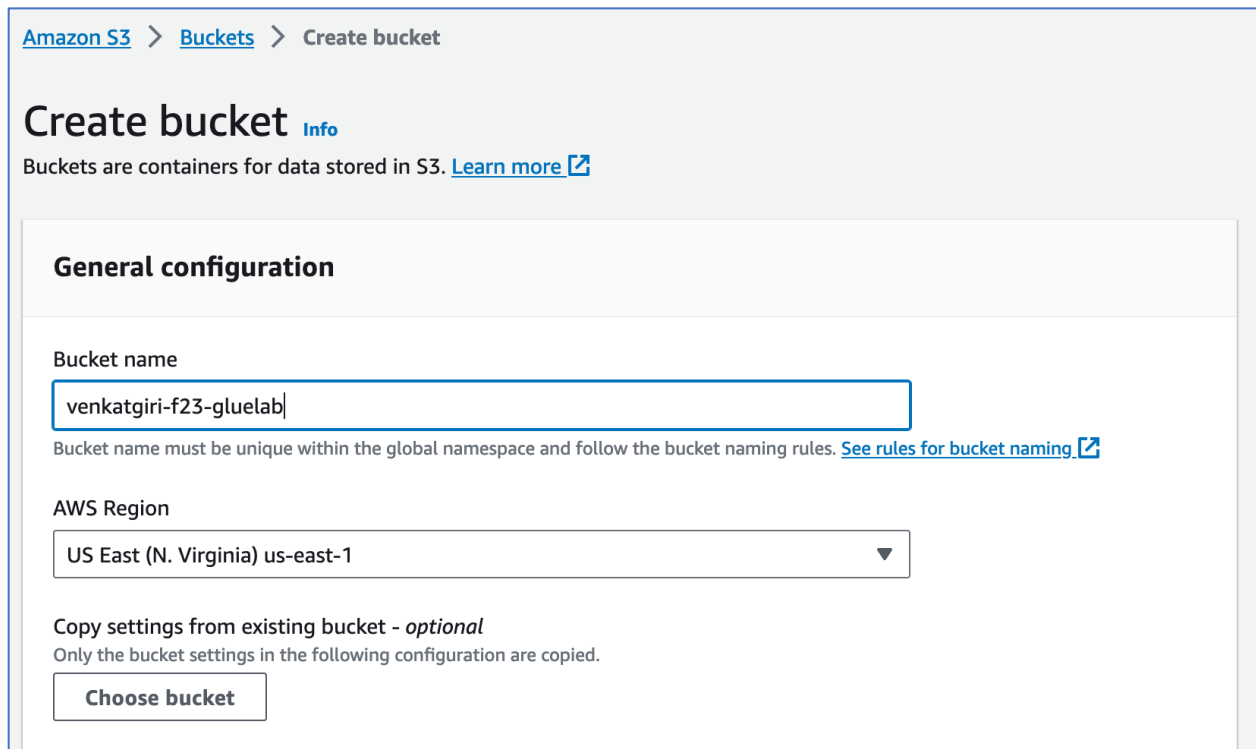
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessFromMultipleAccounts",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::glue-data-mis6389/*"
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessFromMultipleAccounts",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::glue-data-mis6389/*"
    }
  ]
}
```

### Second one:

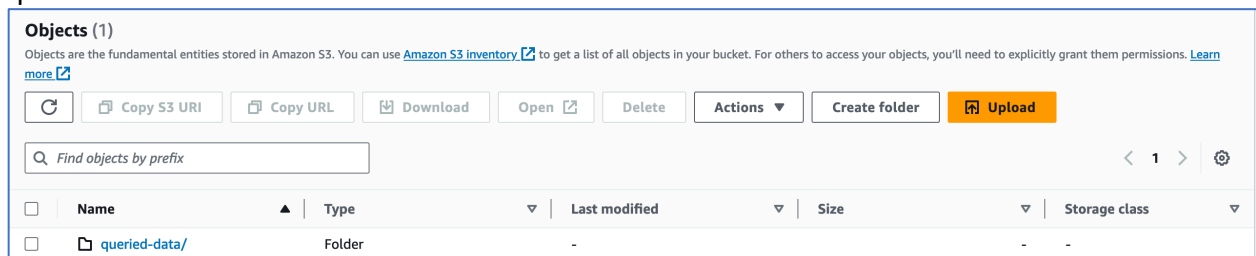
Create an S3 bucket called **<yourname>-f23-gluelab**.

1. Give the **Bucket name** as suggested. Leave the rest of the options to default and scroll down to choose **Create bucket**.



The screenshot shows the 'Create bucket' page in the AWS S3 console. The breadcrumb navigation at the top reads 'Amazon S3 > Buckets > Create bucket'. The main heading is 'Create bucket' with an 'Info' link. Below this, a note states 'Buckets are containers for data stored in S3.' with a 'Learn more' link. The 'General configuration' section contains three main fields: 'Bucket name' with the text 'venkatgiri-f23-gluelab', 'AWS Region' set to 'US East (N. Virginia) us-east-1', and a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button. A note indicates that only the bucket settings in the following configuration are copied.

Once the bucket is created, create a folder named **queried\_data**. We store the results of all the queries we run in the **Athena**.



The screenshot shows the 'Objects' page in the AWS S3 console. It displays a list of objects in a bucket. The table has columns for 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A single object is listed: 'queried-data/' of type 'Folder'. Above the table, there are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar is also present.

**Note: This is a Deliverable**

## Discovering the Data

## AWS Glue and Athena

In this section, we will create the Glue database, add a crawler, and populate the database table using a source CSV file.

2. Choose Services and search for **AWS Glue**. Choose **Databases**. Choose **Add Database**. Paste/type in the following for the Database name: **my\_ev\_database**

The screenshot shows the 'Create a database' form in the AWS Glue console. The breadcrumb navigation at the top reads 'AWS Glue > Databases > Add database'. The main heading is 'Create a database' with a subtitle 'Create a database in the AWS Glue Data Catalog.' Below this is a section titled 'Database details' containing three input fields: 'Name' (with 'my\_ev\_database' entered), 'Location - optional' (with a placeholder 'Set the URI location for use by clients of the Data Catalog.'), and 'Description - optional' (with a placeholder 'Enter text'). At the bottom right of the form are 'Cancel' and 'Create database' buttons.

3. Choose **Tables**. You can add a table manually or by using a crawler. A crawler is a program that connects to a data store and progresses through a prioritized list of classifiers to determine the schema for your data. AWS Glue provides classifiers for common file types like CSV, JSON, Avro, and others. You can also write your classifier using a grok pattern. Choose to **Add tables** using the **crawler**.

The screenshot shows the 'Tables' page in the AWS Glue console. The breadcrumb navigation at the top reads 'AWS Glue > Tables'. The main heading is 'Tables' with a subtitle 'A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.' Below this is a section titled 'Tables (0)' with a subtitle 'View and manage all available tables.' To the right of this section are buttons for 'Last updated (UTC) September 16, 2023 at 17:49:29', 'Delete', 'Add tables using crawler', and 'Add table'. Below these buttons is a search bar labeled 'Filter tables'. At the bottom of the page is a table with columns: 'Name', 'Database', 'Location', 'Classification', 'Deprecated', 'View data', and 'Data quality'. The table currently shows 'No available tables'.

4. Paste in **evdatacrawler** for the Crawler name. Choose **Next**.

## AWS Glue and Athena

AWS Glue > Crawlers > Add crawler

Step 1  
**Set crawler properties**

Step 2  
Choose data sources and classifiers

Step 3  
Configure security settings

Step 4  
Set output and scheduling

Step 5  
Review and create

### Set crawler properties

**Crawler details** [Info](#)

Name  
  
Name can be up to 255 characters long. Some character set including control characters are prohibited.

Description - *optional*  
  
Descriptions can be up to 2048 characters long.

► **Tags - optional**  
Use tags to organize and identify your resources.

Cancel **Next**

### 5. In the next step, choose **Add a data source**

AWS Glue > Crawlers > Add crawler

Step 1  
[Set crawler properties](#)

Step 2  
**Choose data sources and classifiers**

Step 3  
Configure security settings

Step 4  
Set output and scheduling

Step 5  
Review and create

### Choose data sources and classifiers

**Data source configuration**

Is your data already mapped to Glue tables?

☒ **Not yet**  
Select one or more data sources to be crawled.

☐ **Yes**  
Select existing tables from your Glue Data Catalog.

**Data sources (0)** [Info](#) [Edit](#) [Remove](#) [Add a data source](#)

The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
You don't have any data sources.		
<a href="#">Add a data source</a>		

⚠ Data source configuration cannot be empty.

► **Custom classifiers - optional**  
A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel [Previous](#) **Next**

### 6. Choose the Location of s3 data and select the options as shown below.

**Path:** s3://glue-data-mis6389/ev\_data/. This S3 bucket contains the data file ev\_data. Then, Choose to **Add an s3 data source**.

## Add data source

✕

### Data source

Choose the source of data to be crawled.

S3 ▼

### Network connection - *optional*

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

▼ ↻

Clear selection

Add new connection ↗

### Location of S3 data

☒ In this account

☐ In a different account

### S3 path

Browse for or enter an existing S3 path.

🔍 s3://glue-data-mis6389/ev\_data/ ✕

View ↗

Browse S3

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

### Subsequent crawler runs

This field is a global field that affects all S3 data sources.

☒ **Crawl all sub-folders**  
Crawl all folders again with every subsequent crawl.

☐ **Crawl new sub-folders only**  
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

☐ **Crawl based on events**  
Rely on Amazon S3 events to control what folders to crawl.

Cancel

Add an S3 data source





## AWS Glue and Athena

9. Select Crawlers and Select the **evdatacrawler** and choose **Run crawler**. When the crawler has finished, one table has been added. You will see that a table is added.

AWS Glue > Crawlers > evdatacrawler

### evdatacrawler

Last updated (UTC) March 14, 2024 at 04:14:49 [Refresh](#) [Run crawler](#) [Edit](#) [Delete](#)

#### Crawler properties

Name	IAM role	Database	State
evdatacrawler	<a href="#">AWSGlueServiceRole-f23-lab</a>	my_ev_database	READY
Description	Security configuration	Lake Formation configuration	Table prefix
-	-	-	-
Maximum table threshold			
-			

[Advanced settings](#)

[Crawler runs](#) | [Schedule](#) | [Data sources](#) | [Classifiers](#) | [Tags](#)

#### Crawler runs (0)

The list of crawler runs for this crawler.

[Refresh](#) [Stop run](#) [View CloudWatch logs](#) [View run details](#)

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
You don't have any crawler runs.					

AWS Glue > Tables

### Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

#### Tables (1)

Last updated (UTC) March 14, 2024 at 04:20:12 [Refresh](#) [Delete](#) [Add tables using crawler](#) [Add table](#)

View and manage all available tables.

[Refresh](#) [Stop run](#) [View CloudWatch logs](#) [View run details](#)

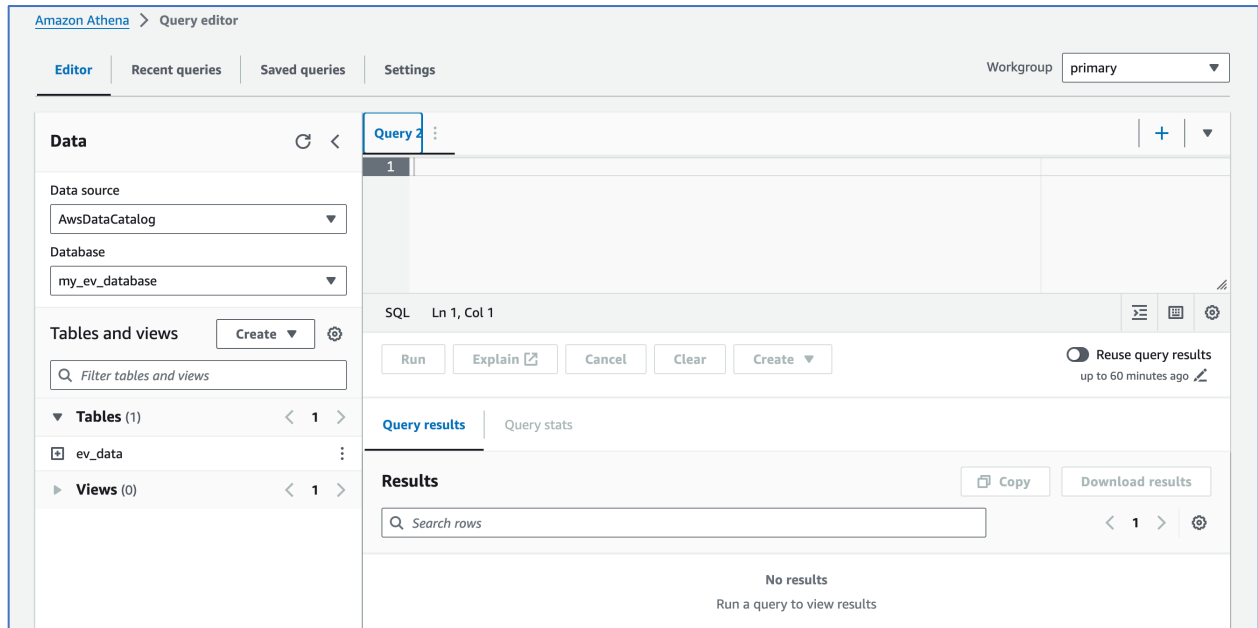
<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data	Data quality
<input type="checkbox"/>	ev_data	my_ev_database	s3://glue-data-mis6:	CSV	-	<a href="#">Table data</a>	<a href="#">View data quality</a>

Note: This is a deliverable (Select Tables from the left menu)

## AWS Glue and Athena

Now that we have the Data catalog and table created, we can start querying using Athena

10. Choose Services and search for Athena. You may need to choose **Launch query editor**. The database will show that **my\_ev\_database** has been selected. Make sure your screen looks as shown below.



11. Choose the **Settings** tab and then choose **Manage**. Here, you will add the target location. Select **Browse S3** and click on the bucket you created earlier in this lab. After that you will see the **queried-data** folder. Select the folder and then click **Choose**. Click on **the Save** option.

## AWS Glue and Athena

Amazon Athena > Query editor > Manage settings

### Manage settings

**Query result location and encryption**  
Location of query result - *optional*  
Enter an S3 prefix in the current region where the query result will be saved as an object.

**You can create and manage lifecycle rules for this bucket**  
Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time.  
[Learn more](#)

**Expected bucket owner - optional**  
Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

☐ **Assign bucket owner full control over query results**  
Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

☐ **Encrypt query results**

12. Come back to the **Editor** tab. Now, you can start querying. Paste the below command.  
Command1:

SELECT \* FROM "my\_ev\_database"."ev\_data" limit 10;

**Data**

Data source: AwsDataCatalog

Database: my\_ev\_database

Tables and views:

▼ **Tables (1)**

- ev\_data
  - vin (1-10) string
  - county string
  - city string
  - state string
  - postal code bigint
  - model year bigint
  - make string
  - model string
  - electric vehicle type string

Query 2 : X Query 3 : X

1 SELECT \* FROM "my\_ev\_database"."ev\_data" limit 10;

SQL Ln 1, Col 1

☐ Reuse query results up to 60 minutes ago

**Query results** | Query stats

Completed Time in queue: 76 ms Run time: 547 ms Data scanned: 812.80 KB

**Results (10)**

#	vin (1-10)	county	city	state	postal code	model year	make	model	electric vehicle type
1	5YJYGDEE1L	King	Seattle	WA	98122	2020	TESLA	MODEL Y	Battery Electric Vehicle
2	5YJYGDEE1L	King	Seattle	WA	98122	2020	TESLA	MODEL Y	Battery Electric Vehicle

Note: This is a deliverable

## AWS Glue and Athena

Here are a few more commands for you:

- `SELECT "vin (1-10)",make,"electric range" FROM "my_ev_database"."ev_data" where "electric range" > 200;`
- `select make from my_ev_database.ev_data where make != 'TESLA';;`
- `select make from my_ev_database.ev_data where "electric vehicle type" != 'Battery Electric Vehicle (BEV)';`

Run the 2 SQL commands of your choice, or choose from the above commands.

**(Note: This is a deliverable (screenshot of each command and respective output))**

**Deliverables: A total of 5 screenshots are expected from this lab.**

Lab Cleanup:

Make sure to delete the crawler, table, database, and s3 bucket.