LAMBTON COLLEGE

A Report on
[Lab 1,2,3 on AWS Academy Data Analytics]

121 Brunel Rd, Mississauga

ON L4Z 3E9

A Group assignment with screenshots of Lab 1, 2, and 3

on Aws academy

Big Data Analytics DSMM

Under the supervision
of
Professor Teresa Zhu

Submitted BY:
Aadarsha Chapagain (C0825975)
Roshan Acharya (C0831342)
Anjana Kuriakose (C0829580)
Onyinye Mbanefo (C0831578)

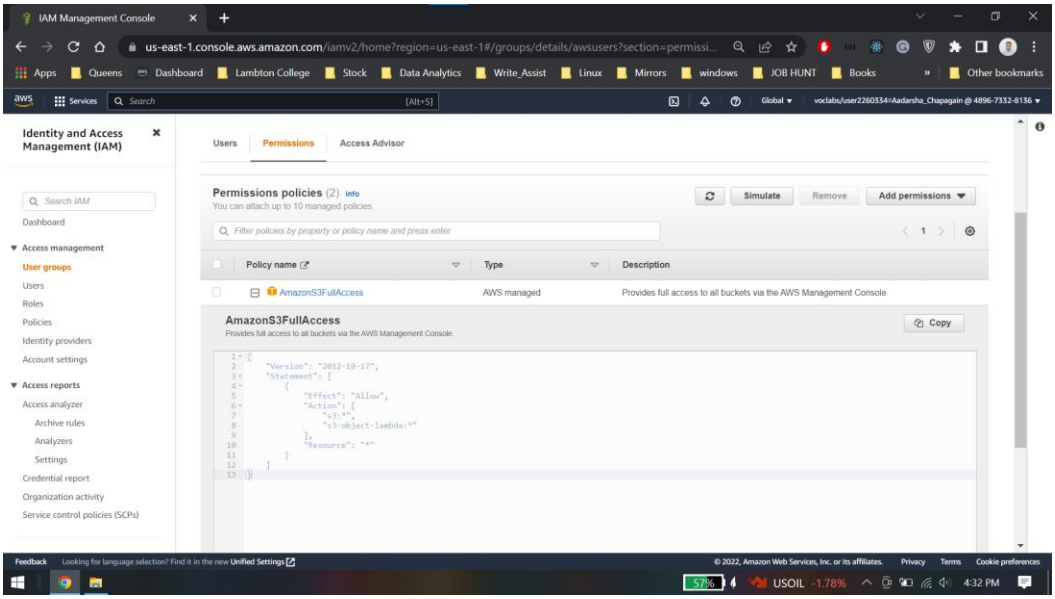
Submitted To:
Lambton College
Professor Teresa Zhu

Submission Date:
27th November 2022

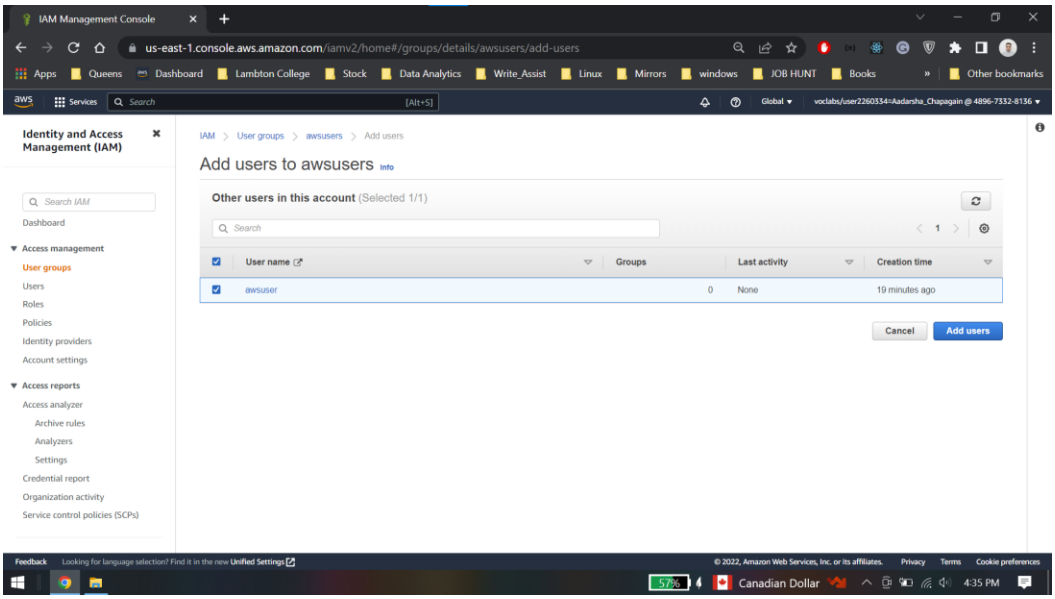
Lab1: Store data in Amazon S3

Task 1: Create an IAM user account

Task 1.1: Review users and group permissions in the IAM console



Task 1.2: Add awsuser to the awsusers group

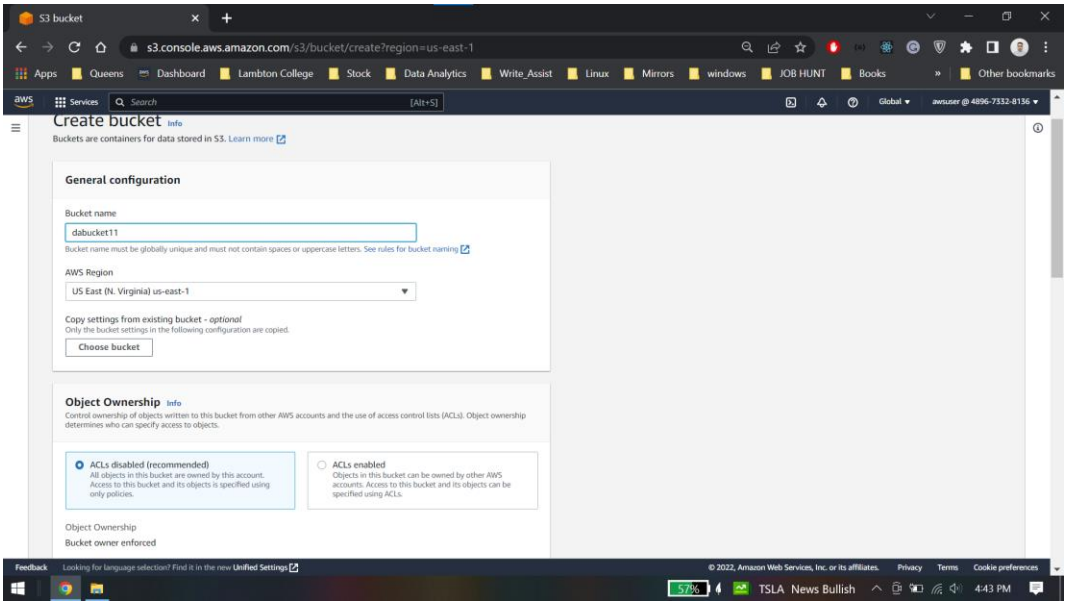


AccountId: 489673328136

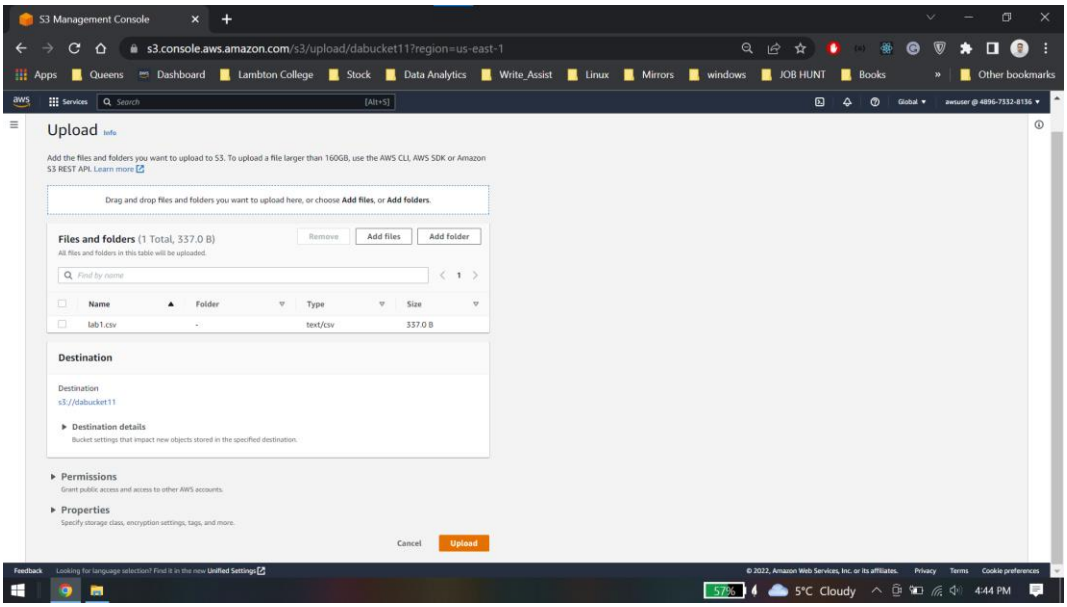
Task 2: Load data into Amazon S3

Task 2.1: Create an S3 bucket

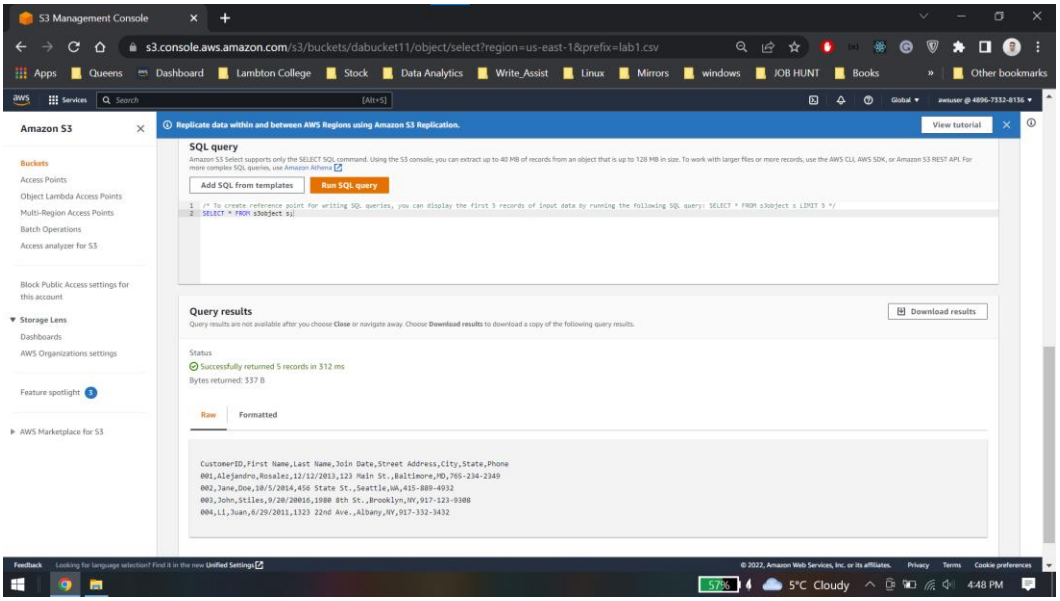
Bucket name: dabucket11



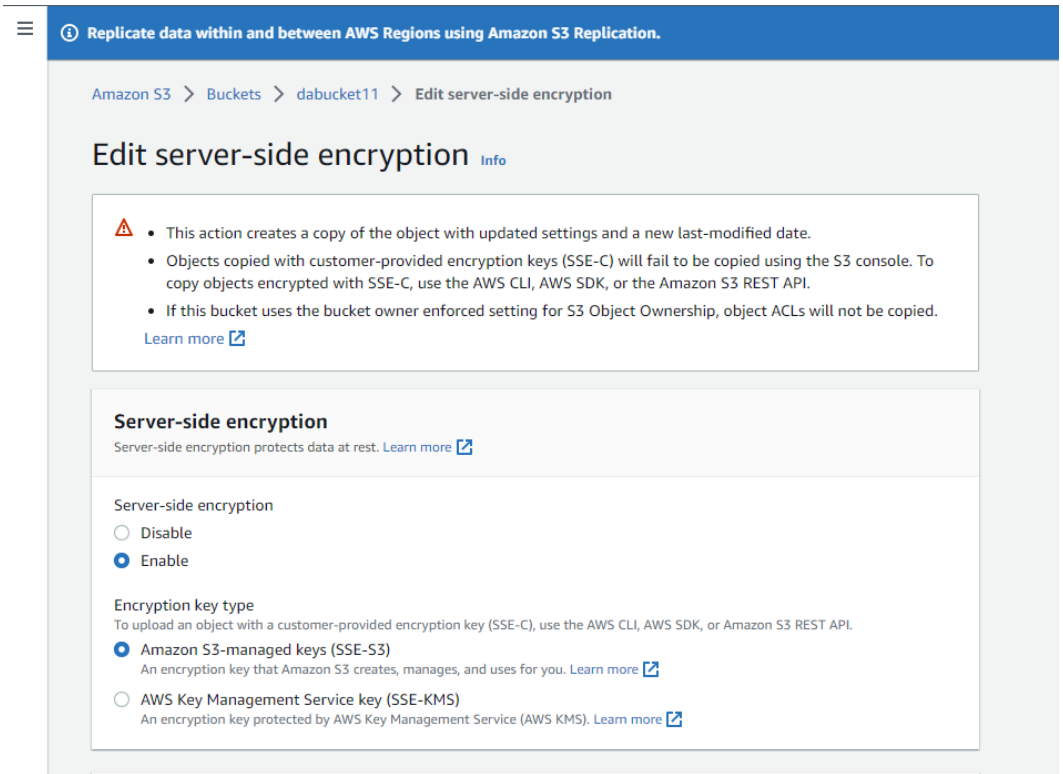
Task 2.2: Upload an object

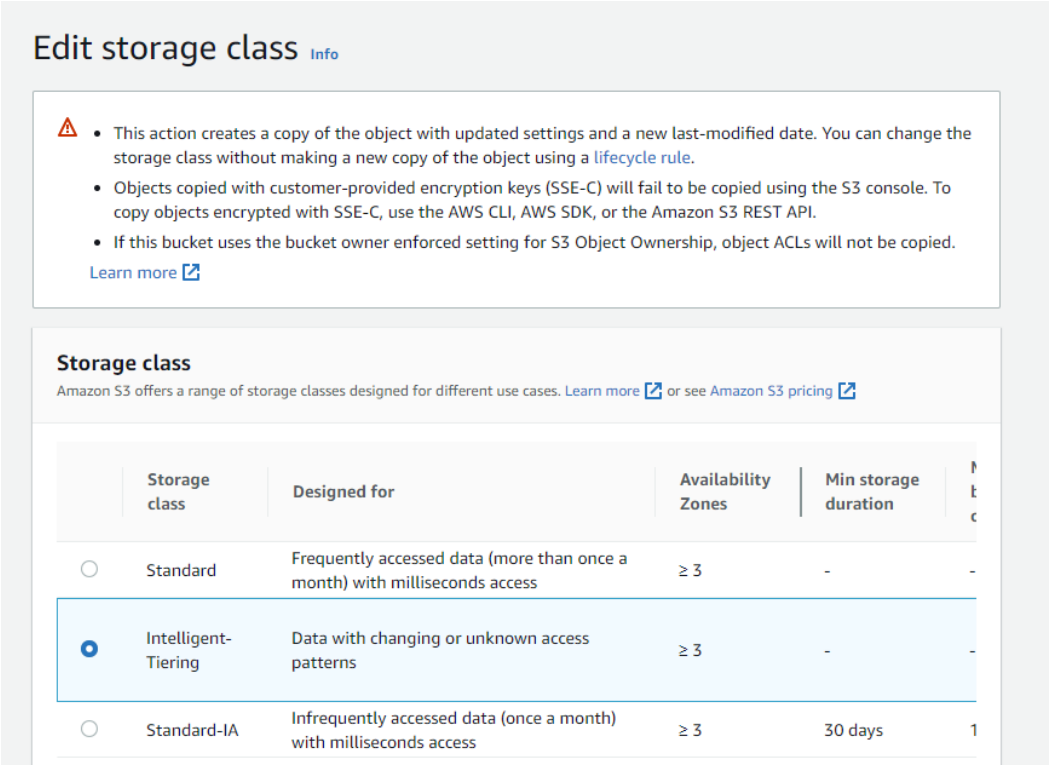


Task 2.3: Query the object you uploaded

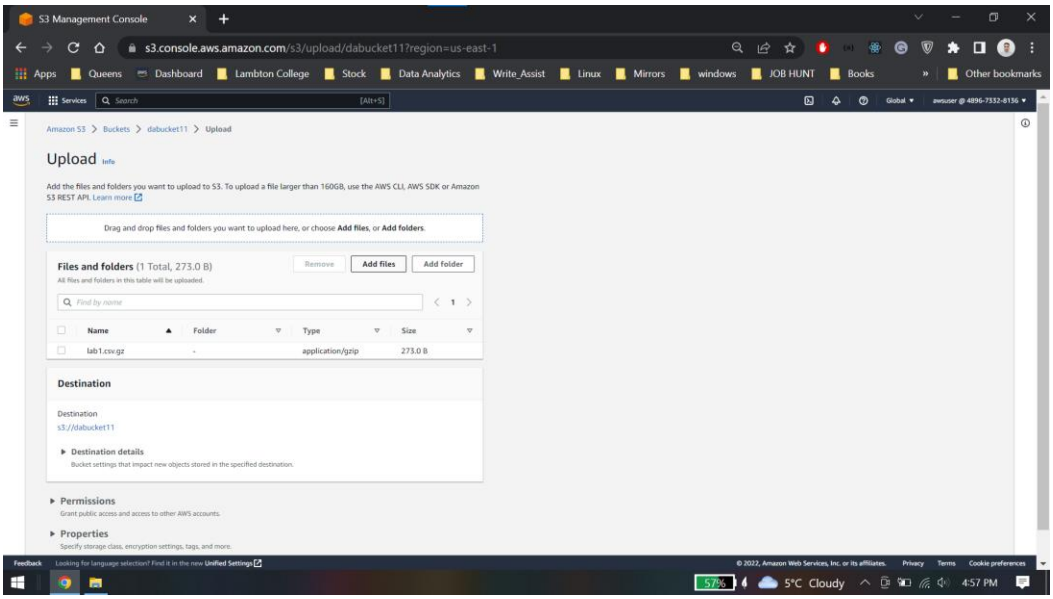


Task 2.4: Change the encryption properties and storage type

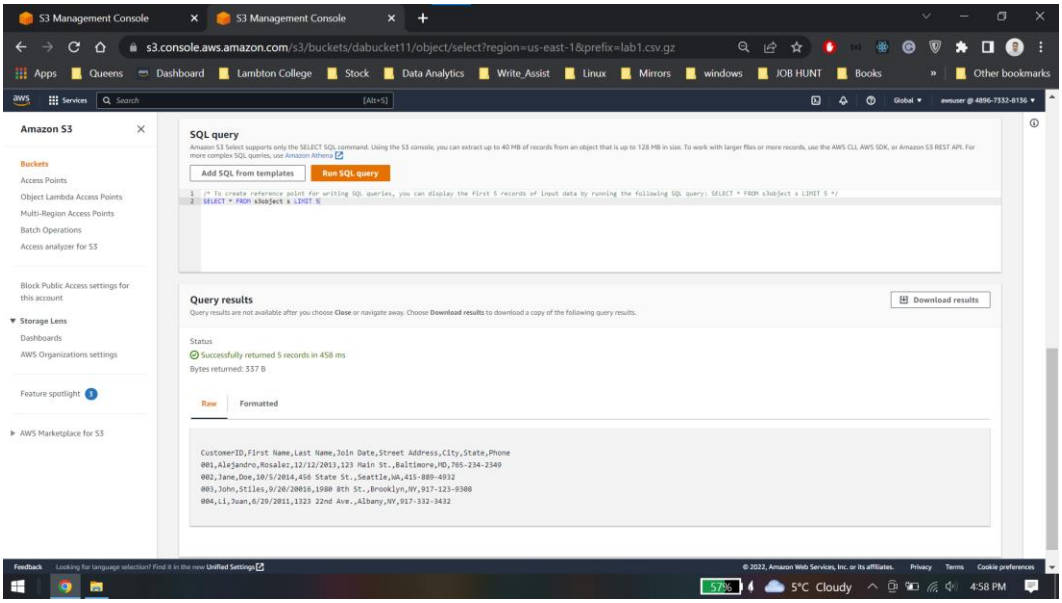




Task 2.5: Upload a compressed file



Compressed file can be queried in the same way as a non-compressed file.



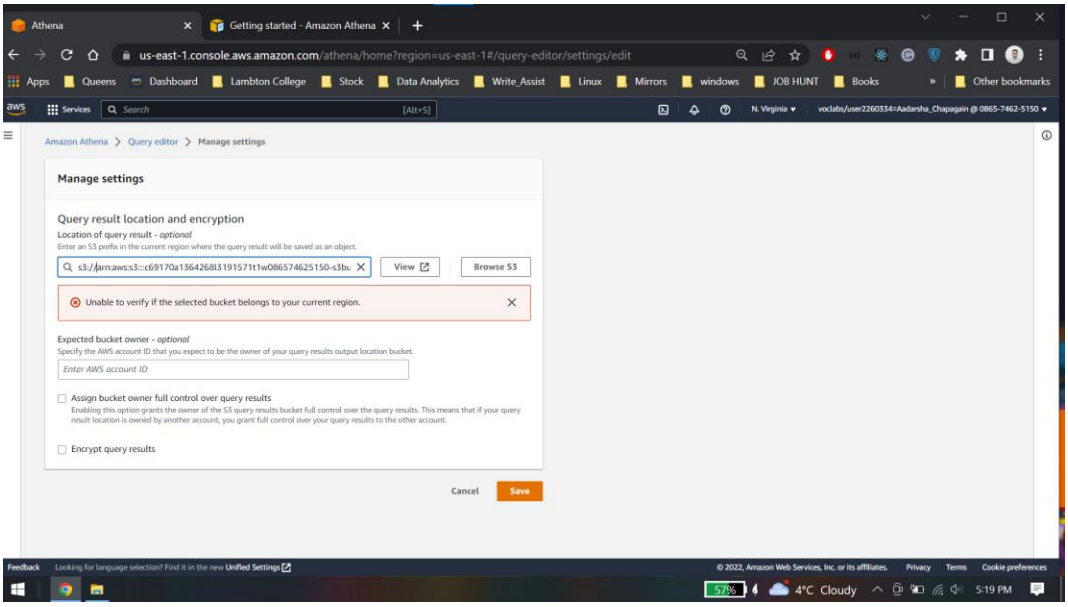
Lab 1 Conclusion.

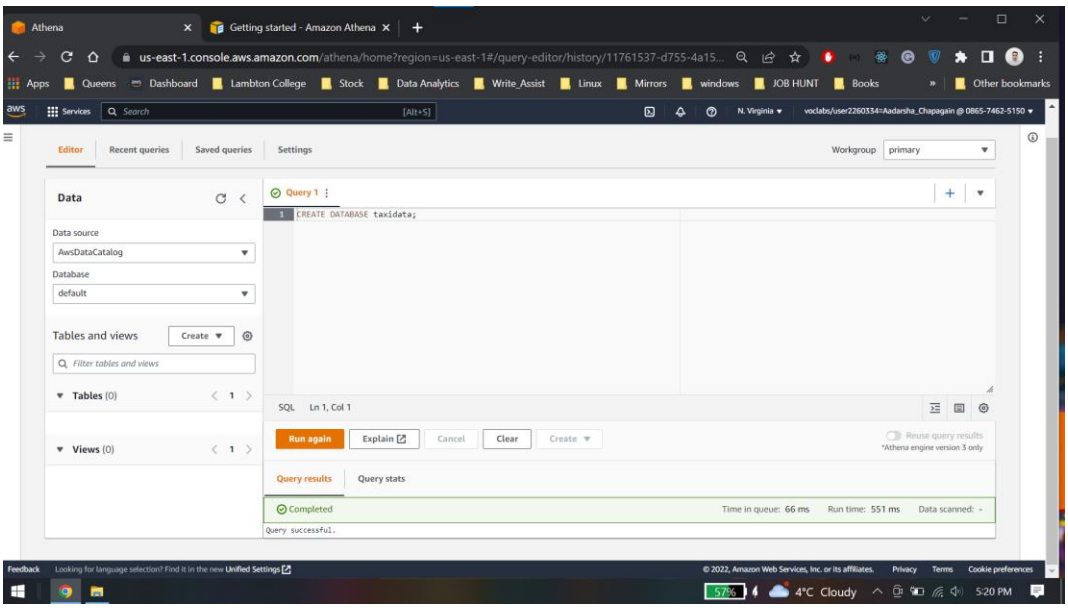
- Access Amazon S3 in the AWS Management Console
- Secure an S3 bucket with IAM
- Create a bucket with Amazon S3
- Load data into an S3 bucket
- Query an S3 bucket

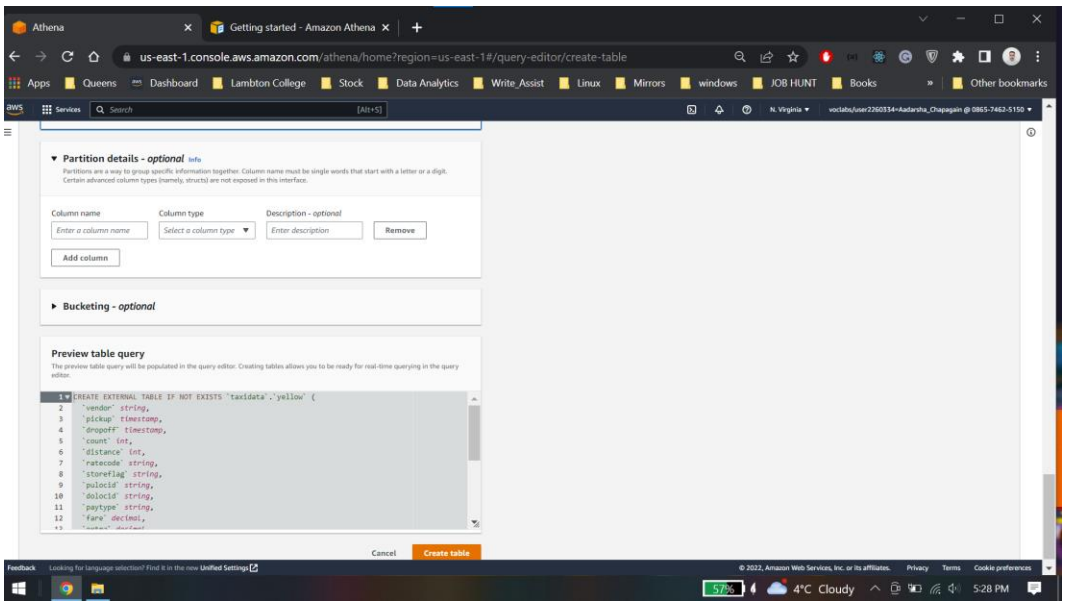
Lab 2: Query Data in Amazon Athena

Task 1 : Query Data in Amazon Athena

Bucket ARN: arn:aws:s3:::c69170a1364268l3191571t1w086574625150-s3bucket-q3gdz12cykyj





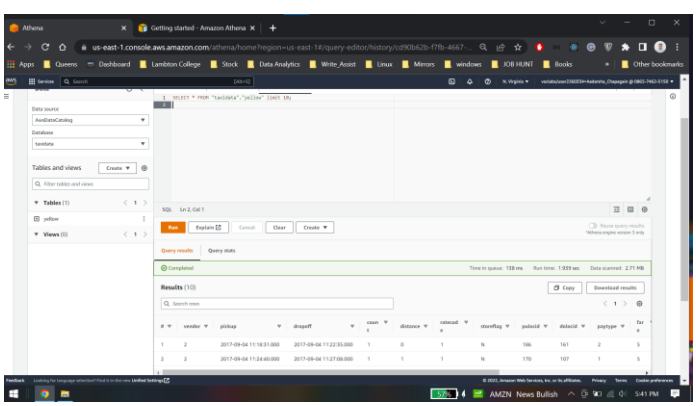


Preview table query

The preview table query will be populated in the query editor. Creating tables allows you to be ready for real-time querying in the query editor.

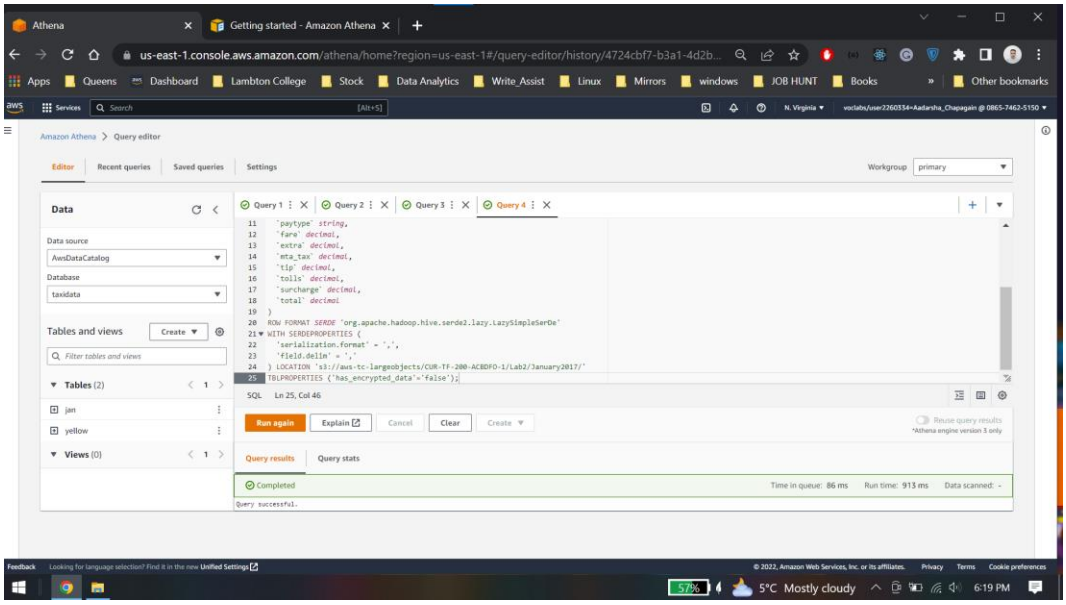
```
  `extra` decimal,
  `mta_tax` decimal,
  `tip` decimal,
  `tolls` decimal,
  `surcharge` decimal,
  `total` decimal
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim' = ',')
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat' OUTPUTFORMAT 'org.apache
    .hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://aws-tc-largeobjects/CUR-TF-200-ACBDFO-1/Lab2/yellow/'
TBLPROPERTIES ('classification' = 'csv');
```

Cancel **Create table**

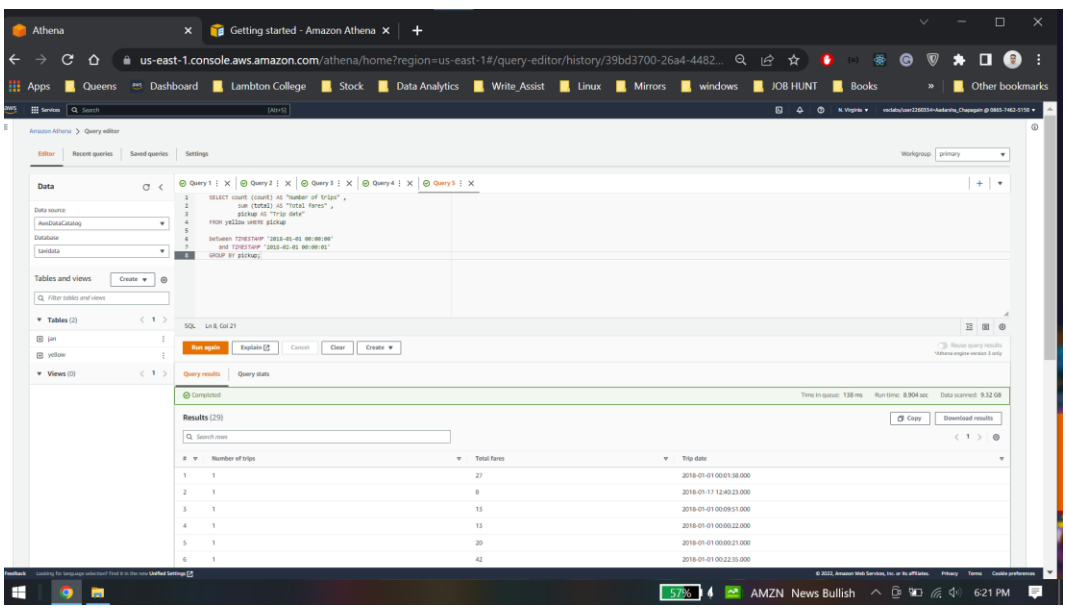


Task 2: Optimize the database

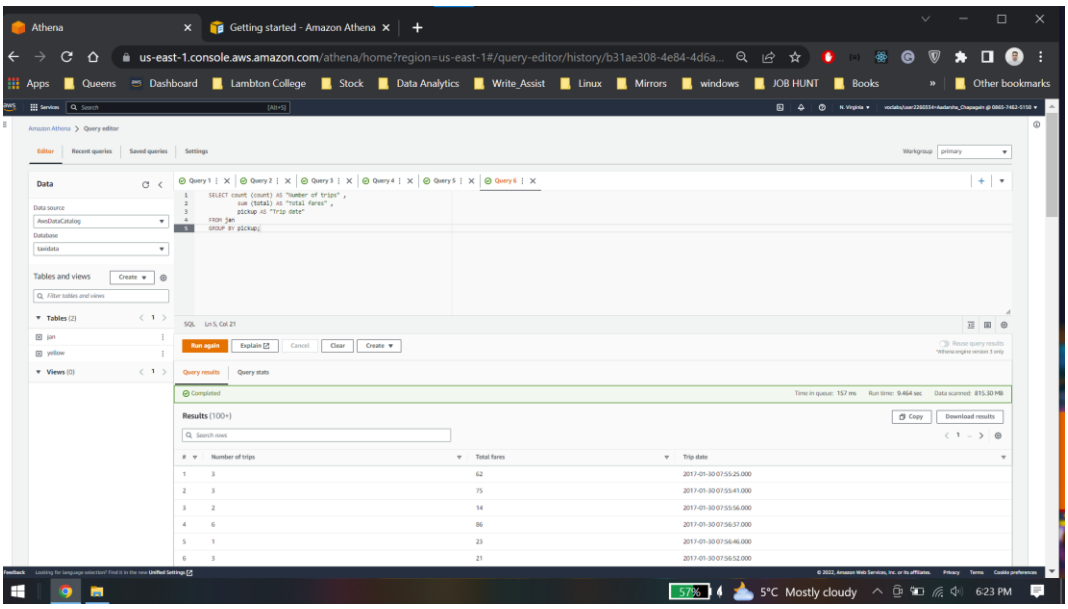
Task 2.1: Create a table for the January 2017 data



Task 2.2: Run a query using the data that is not divided into buckets



Task 2.3: Run a query using the data that is divided into buckets for each month

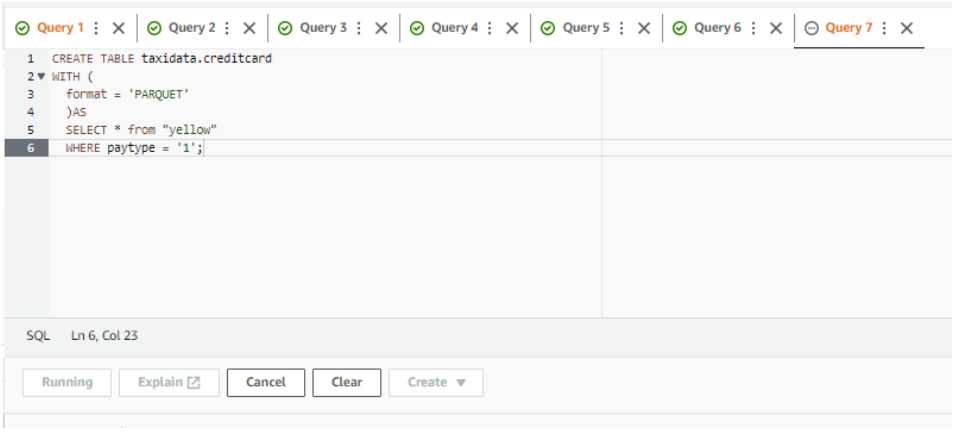


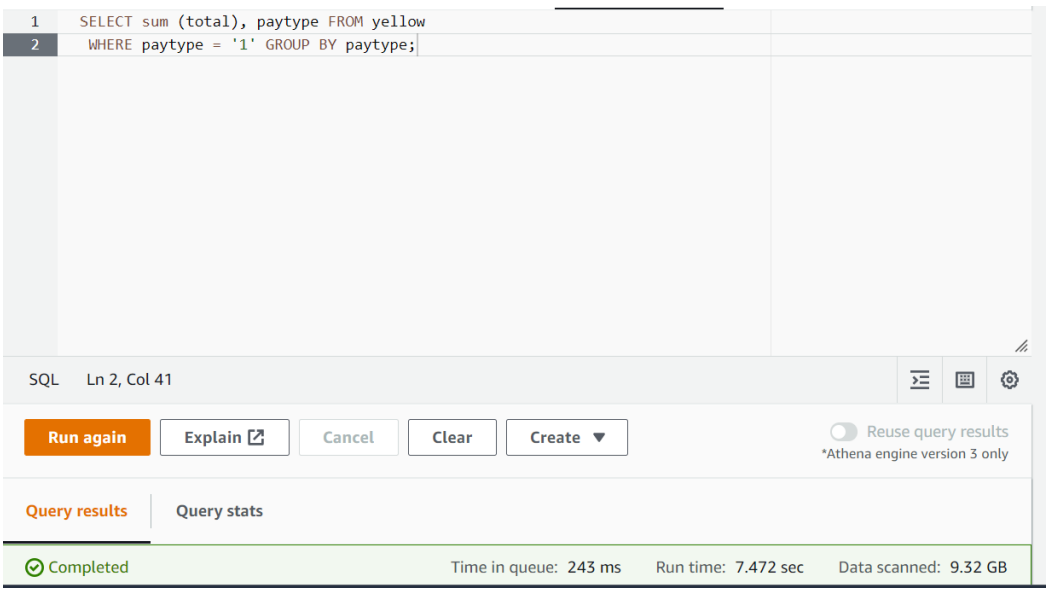
Following results was found

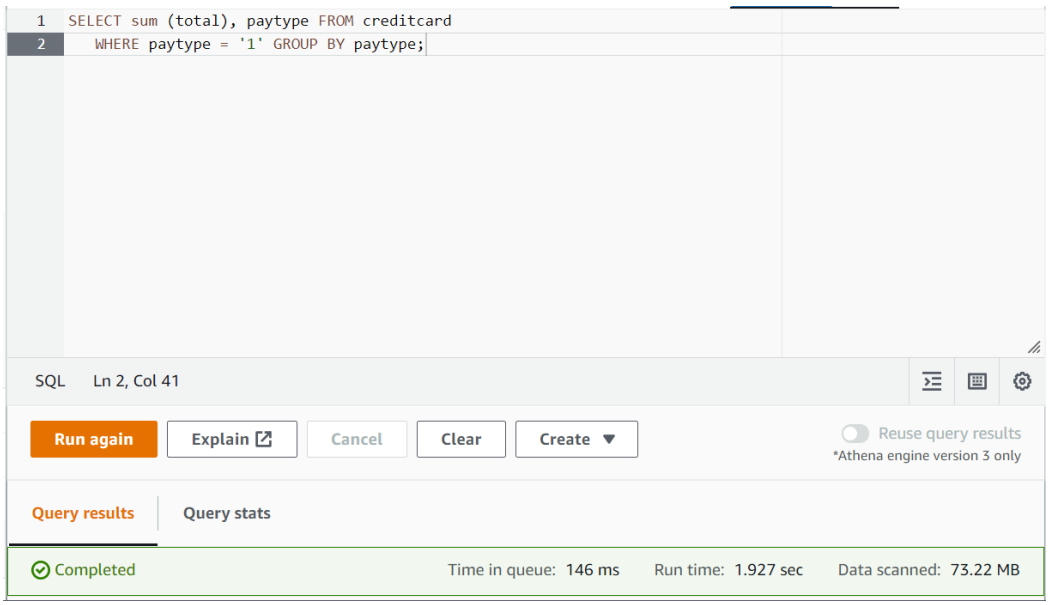
- No buckets:
    - Total data scanned: 9.32 GB
- Buckets:
    - Total data scanned: 815 MB

Task 2.4: Query partitioned data

Task 2.4.1: Partition the data







Yellow table:
Run time: 7.19 seconds & Data scanned: 9.32 GB
Credit card table:
Run time: 3.32 seconds & Data scanned: 71.8 MB

Task 3: Create and query views

Query 9

```
CREATE VIEW cctrips AS
SELECT "sum"("fare") "CreditCardFares"
FROM yellow
WHERE ("paytype"='1');
```

SQL Ln 4, Col 23

Run again | Explain | Cancel | Clear | Create

Reuse query results
*Athena engine version 3 only

Query results | Query stats

Completed Time in queue: 96 ms Run time: 458 ms Data scanned: -

Query 9

```
CREATE VIEW cashtrips AS
SELECT "sum"("fare") "CashFares"
FROM yellow
WHERE ("paytype"='2');
```

SQL Ln 4, Col 23

Run again | Explain | Cancel | Clear | Create

Reuse query results
*Athena engine version 3 only

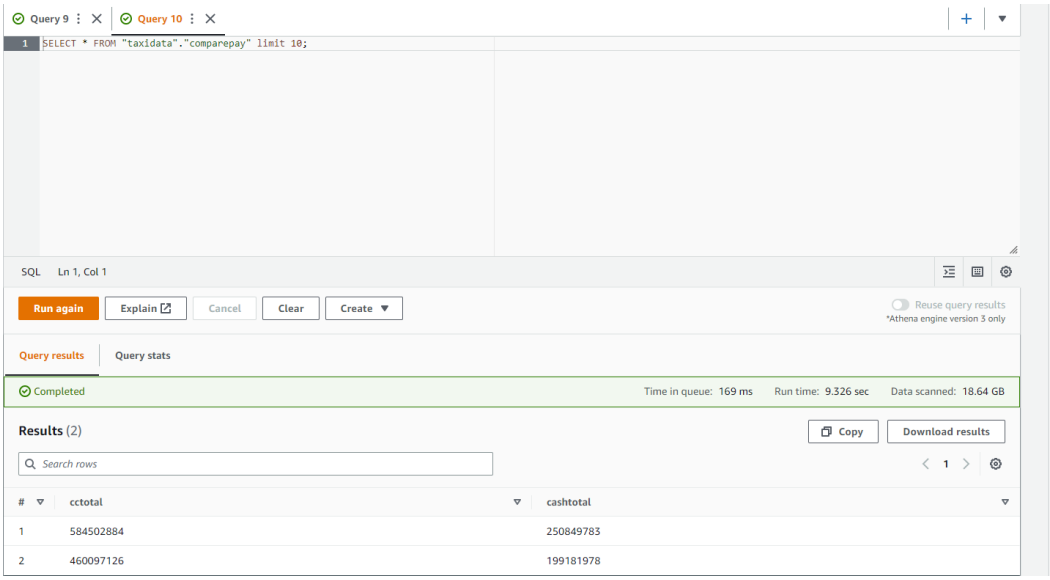
Query results | Query stats

Completed Time in queue: 53 ms Run time: 456 ms Data scanned: -

Views (2) 1

cashtrips

cctrips



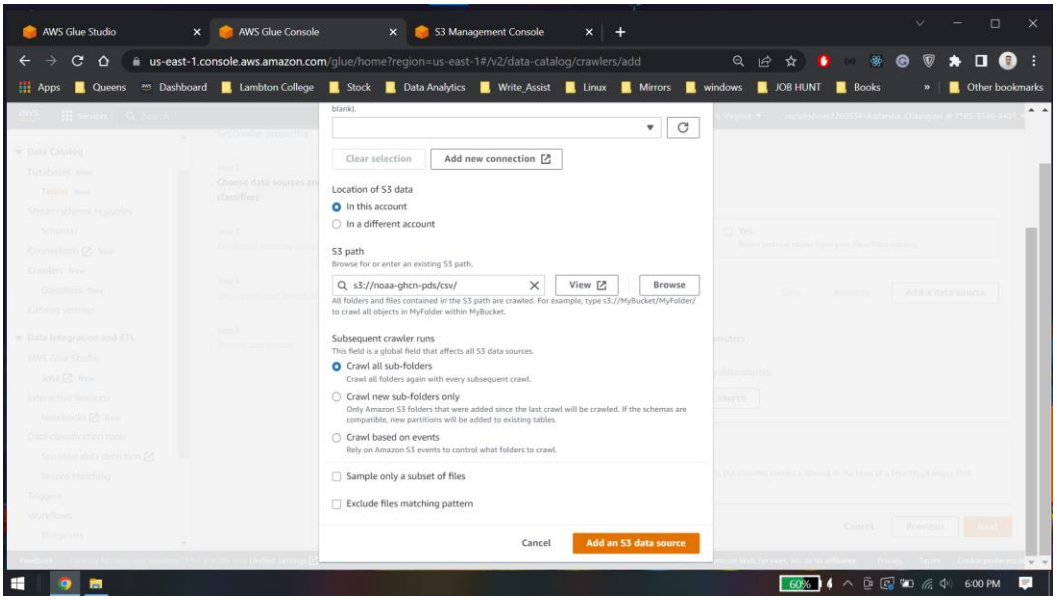
Lab2 Conclusion

- Accessed Athena in the AWS Management Console
- Created tables and define data types
- Queried data in Amazon S3 from Athena
- Optimized queries with partitioning

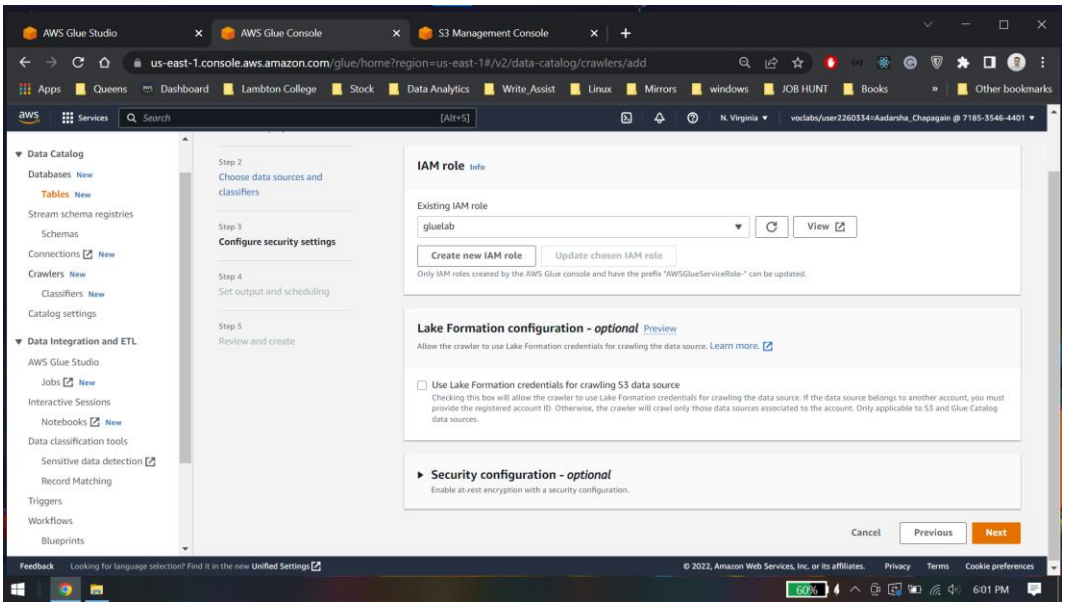
Lab 3: Query Data in Amazon S3 with Amazon Athena and AWS Glue

Task 1: Create a crawler for the GHCN-D dataset

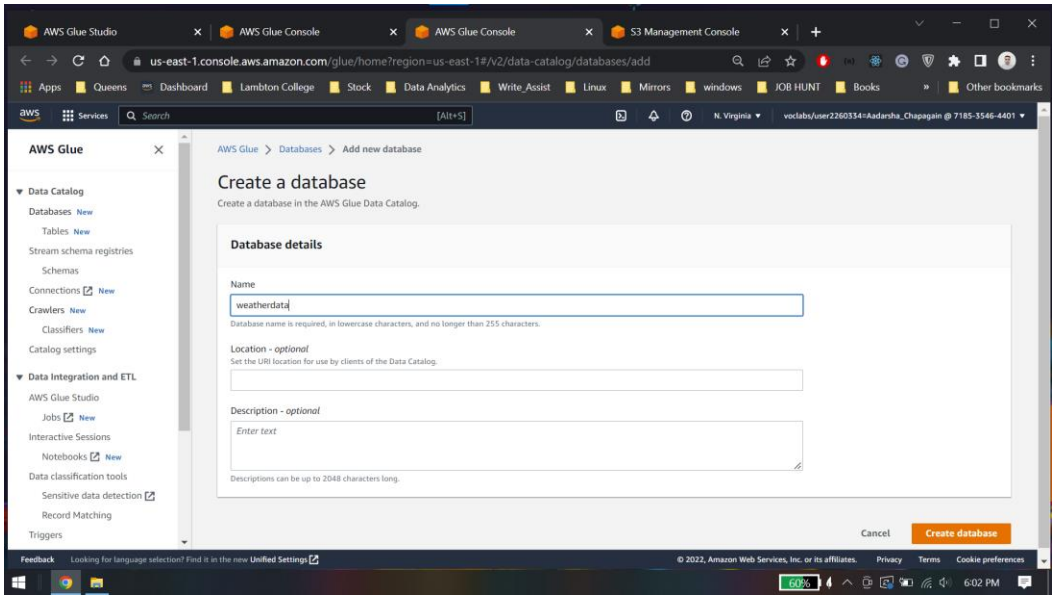
Create a Crawler



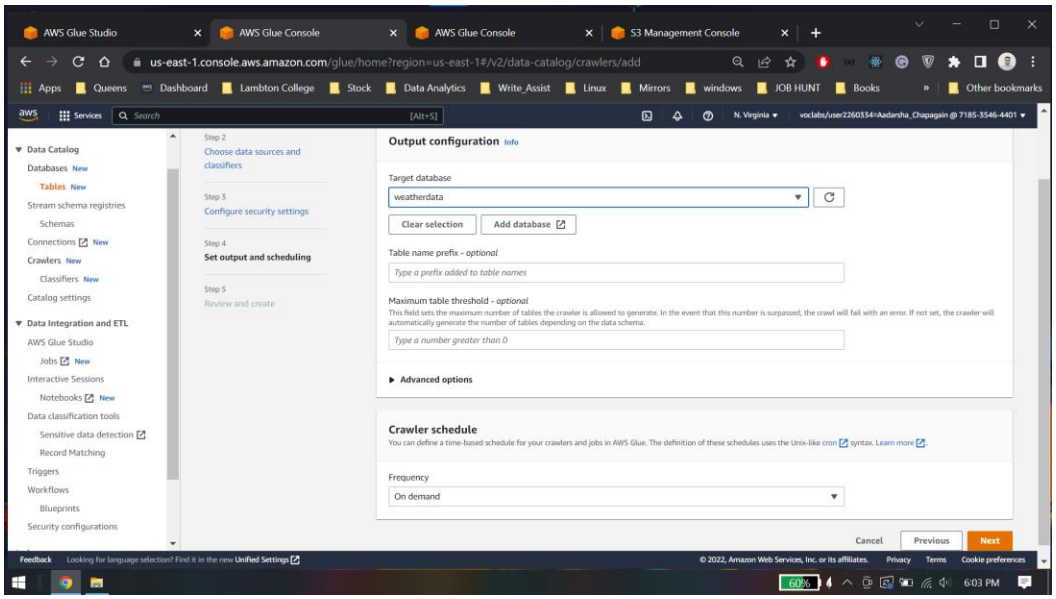
Choose Iam role



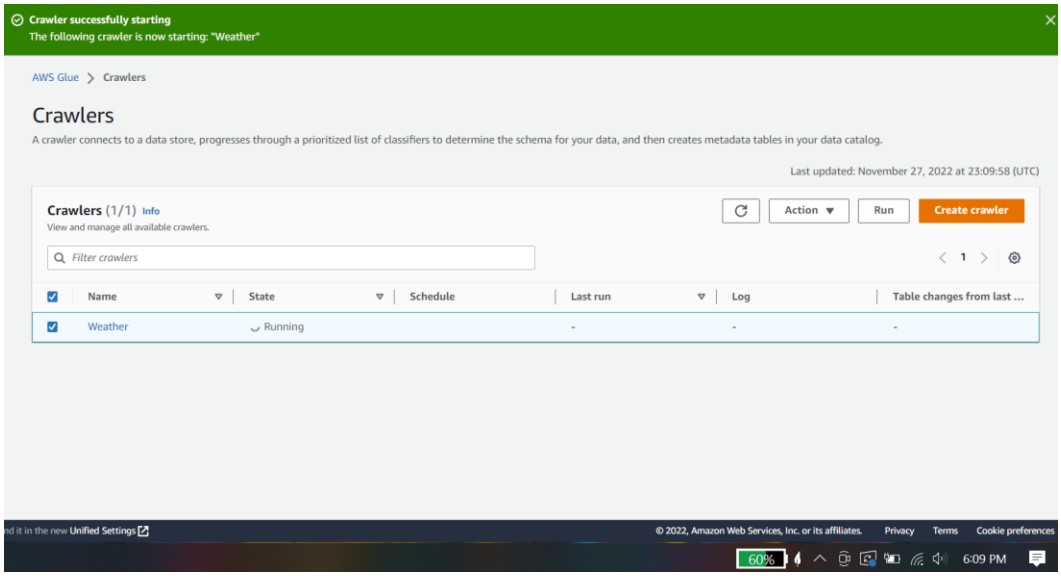
Create a database



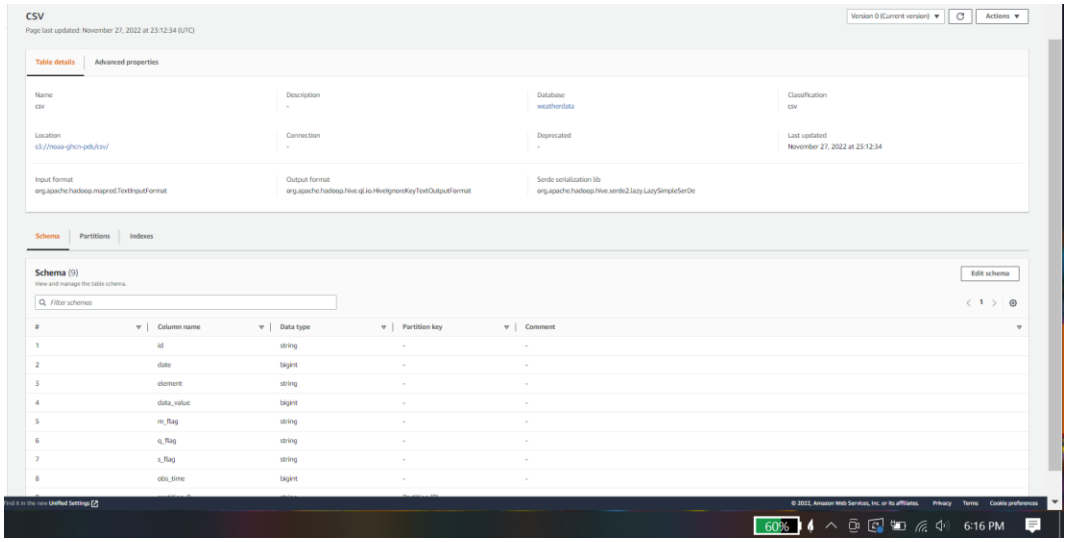
Output Configuration



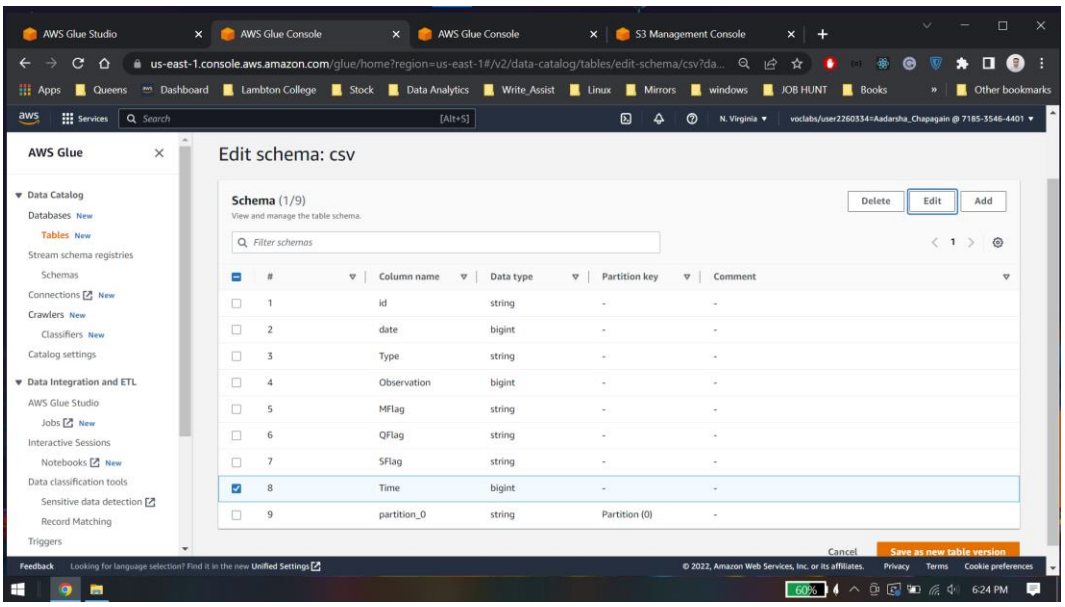
Task 1.1: Run the crawler



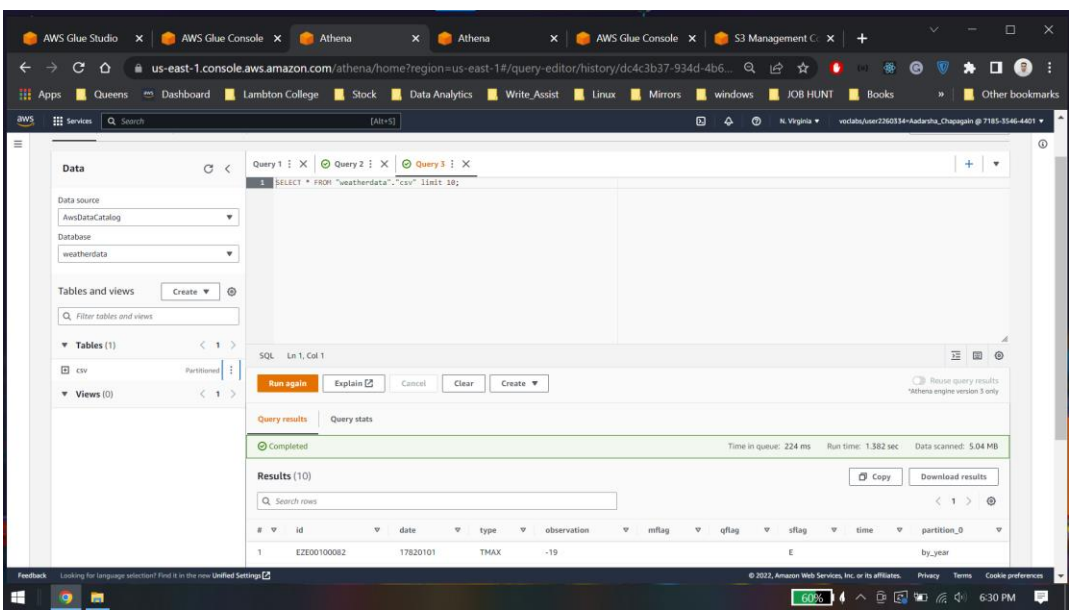
Task 1.2: Review the metadata created by AWS Glue



Task 1.3: Edit the schema

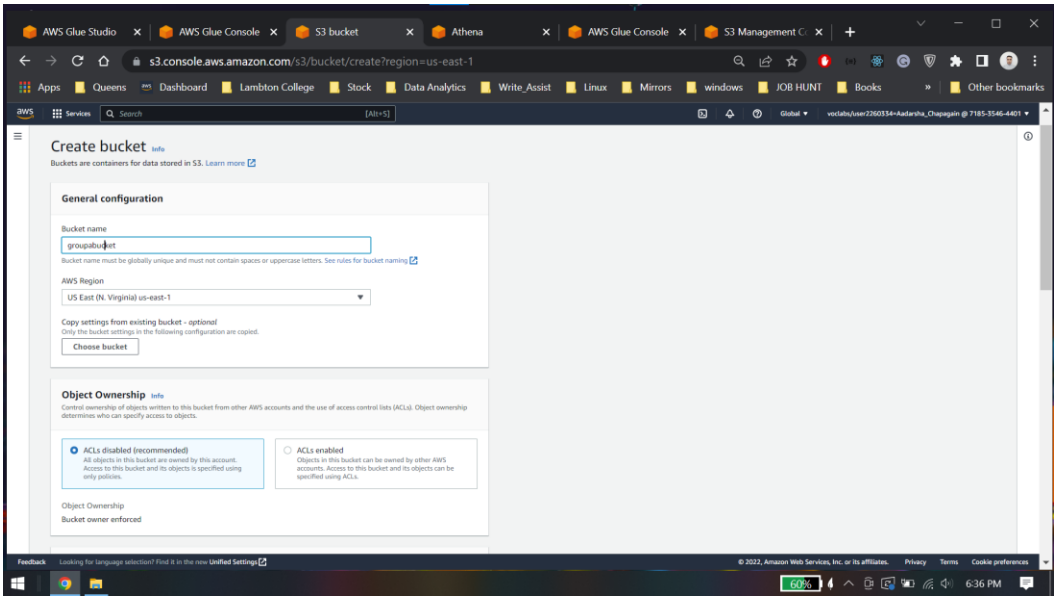


Task 2: Query the table using the AWS Glue Data Catalog



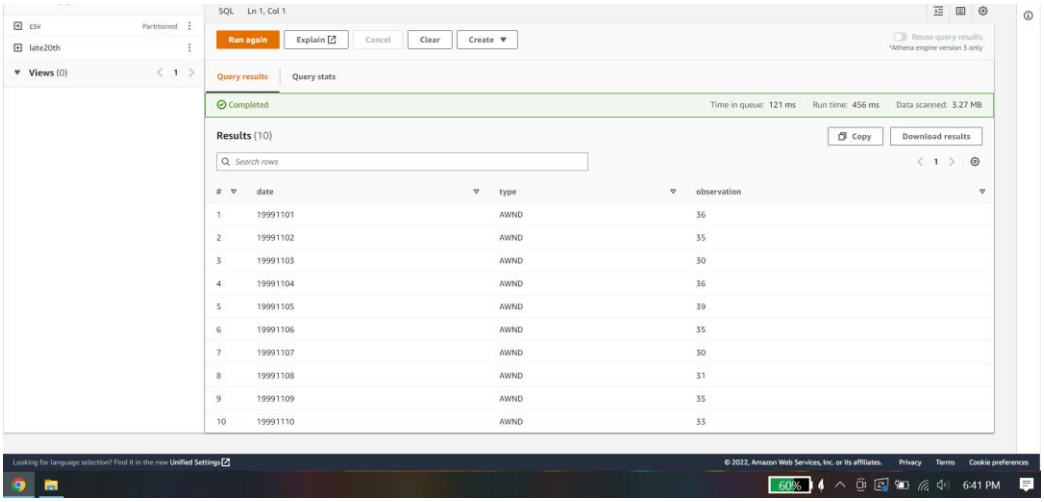
Task 2.1: Create a table for data after 1950

Create a bucket in same region



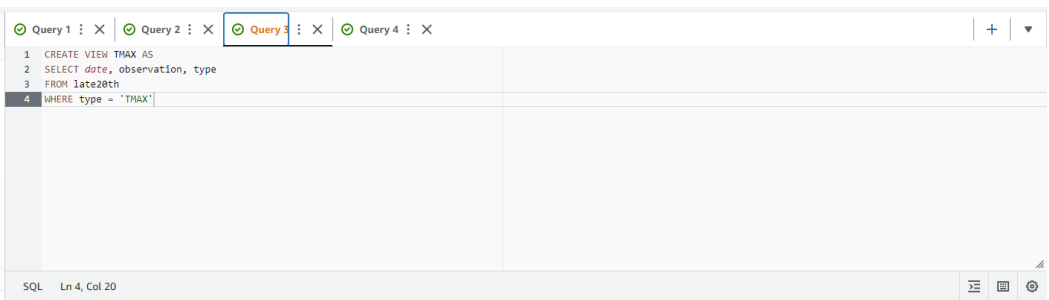
Create a table specifying the bucket location

Preview table

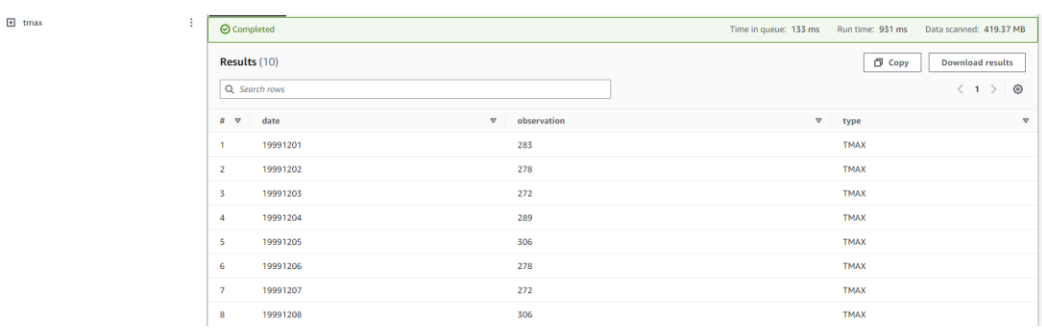


Task 2.2: Run a query from the selected data

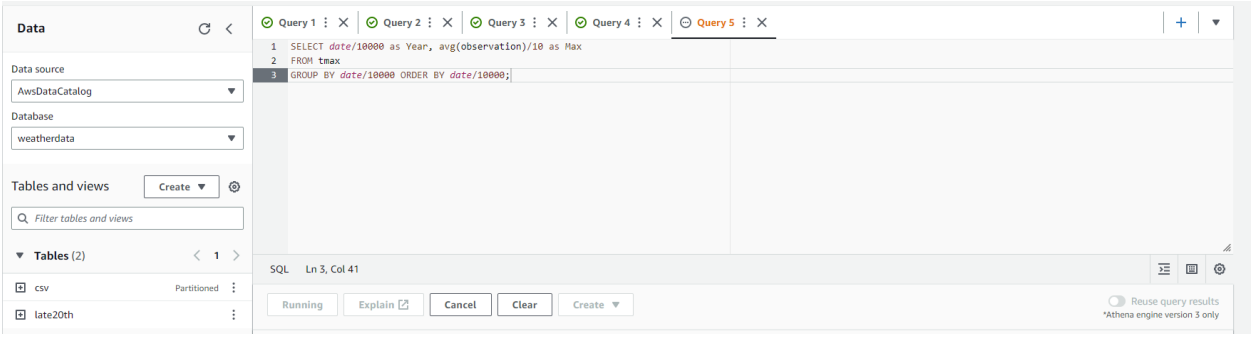
Create a view



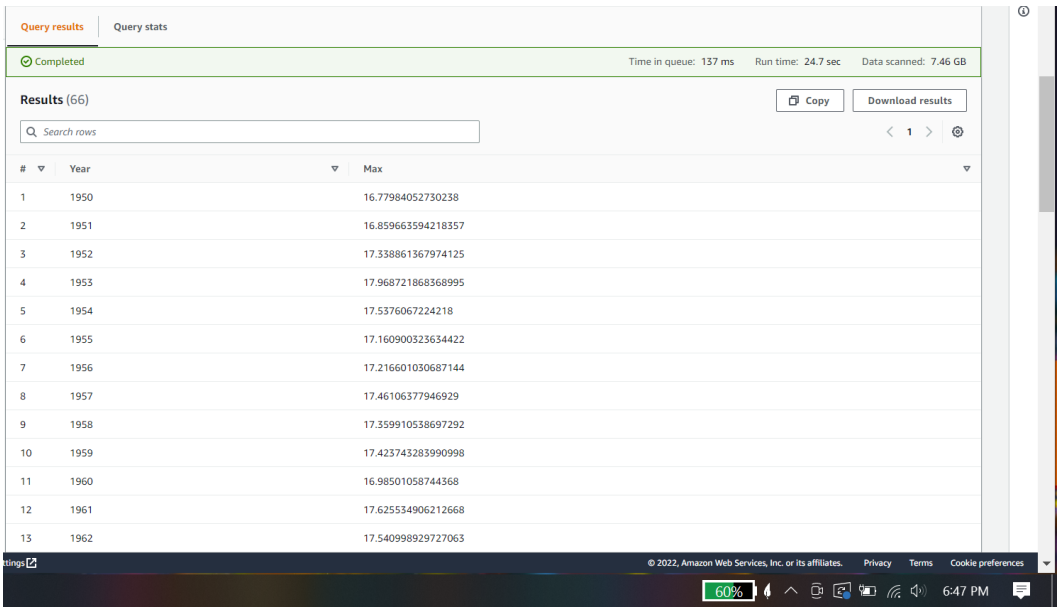
Preview the data



Average maximum temperature from 1950 to 2018



Results



#	Year	Max
1	1950	16.77984052730238
2	1951	16.859663594218357
3	1952	17.338861367974125
4	1953	17.968721868368995
5	1954	17.5376067224218
6	1955	17.160900323634422
7	1956	17.216601030687144
8	1957	17.46106377946929
9	1958	17.359910538697292
10	1959	17.423743283990998
11	1960	16.98501058744368
12	1961	17.625534906212668
13	1962	17.540998929727063

Lab 3 Conclusion

- Accessed AWS Glue in the AWS Management Console
- Created a crawler with AWS Glue
- Created tables and a schema with AWS Glue
- Queried data in Amazon Simple Storage Service (Amazon S3) from Amazon Athena with the AWS Glue data catalog