

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

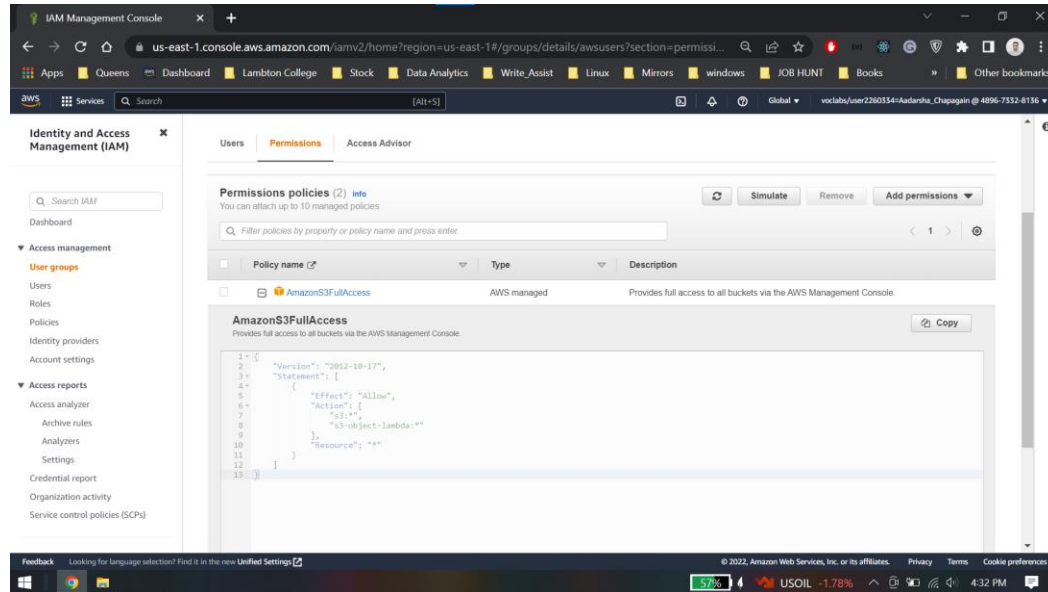
Submission Date:

27th November 2022

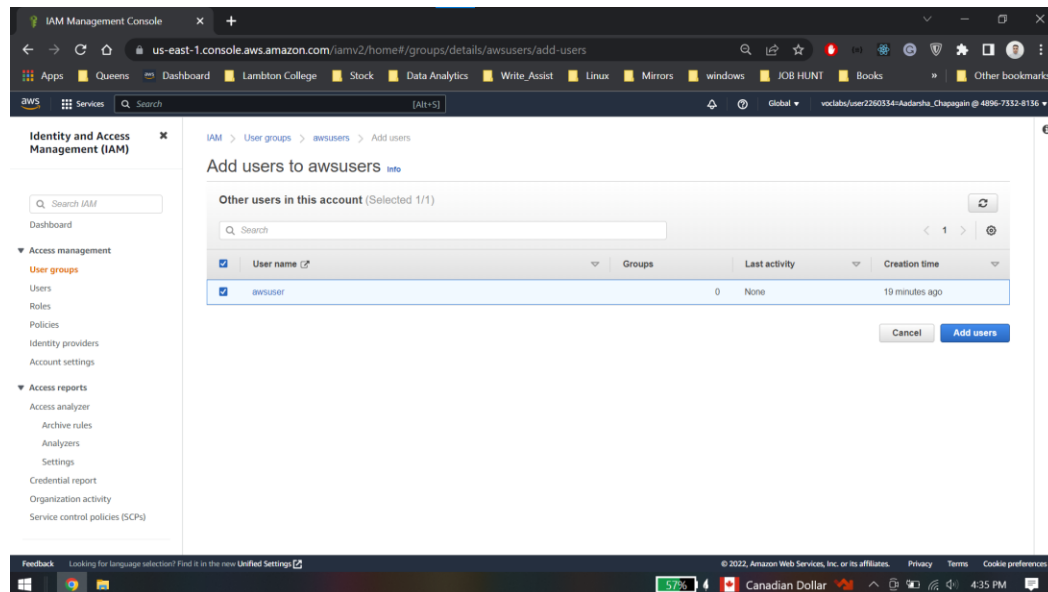
Lab1: Store data in Amazon S3

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

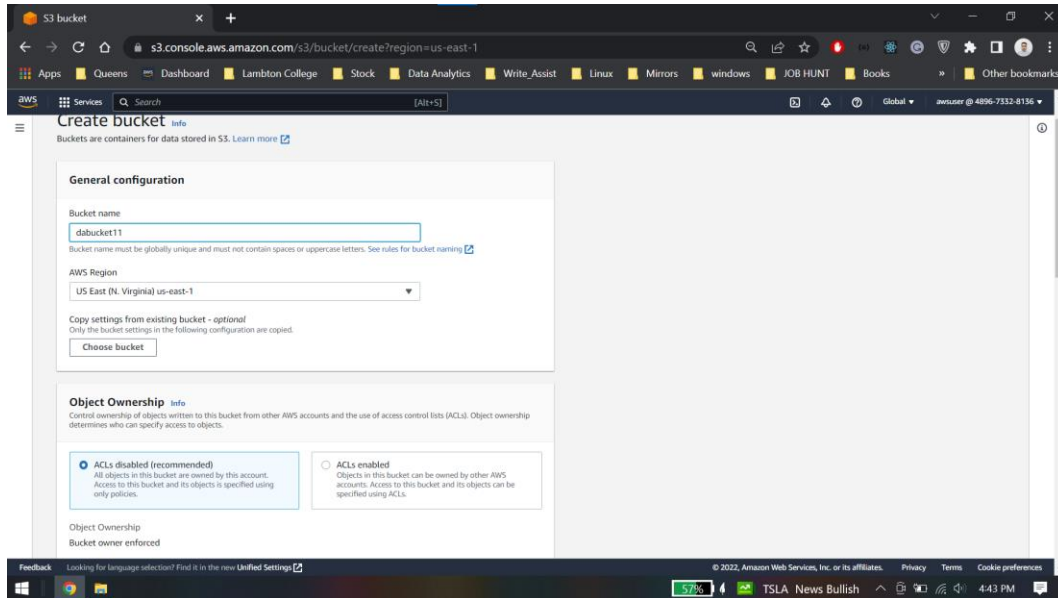


AccountId: 489673328136

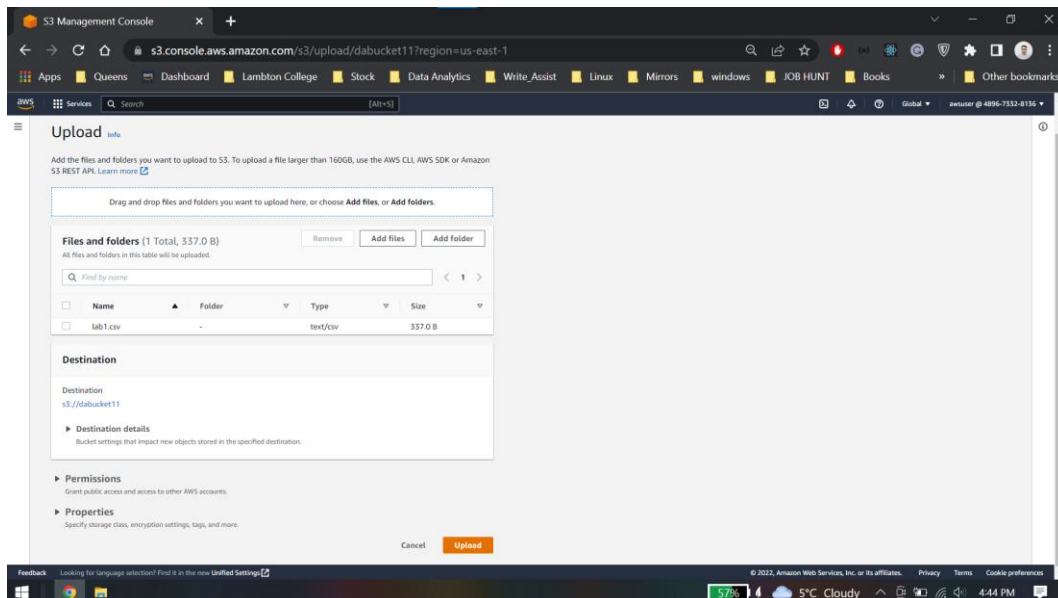
Task 2: Load data into Amazon S3

Task 2.1: Create an S3 bucket

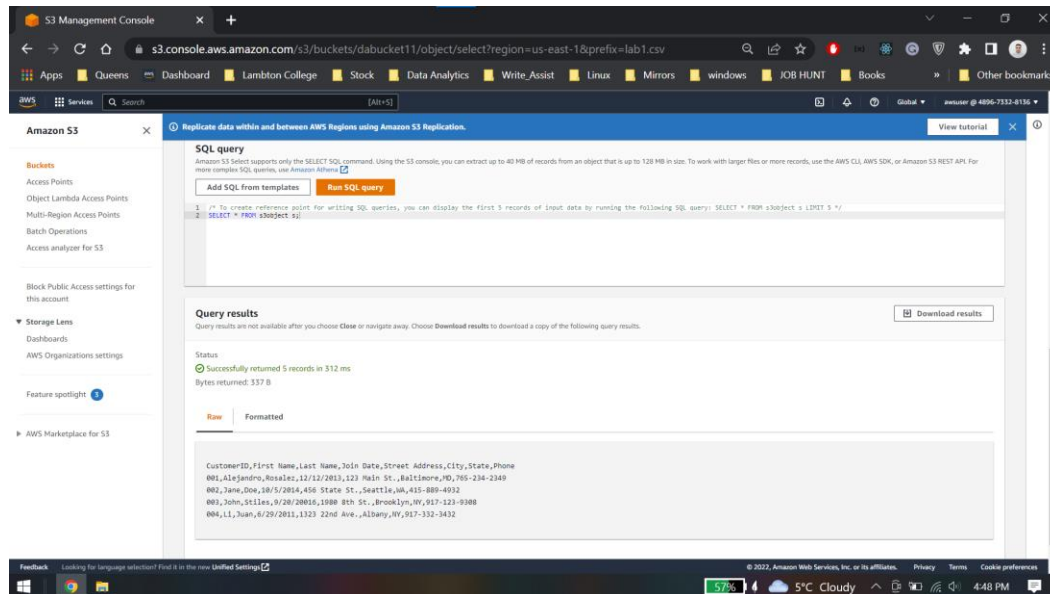
Bucket name: dabucket11



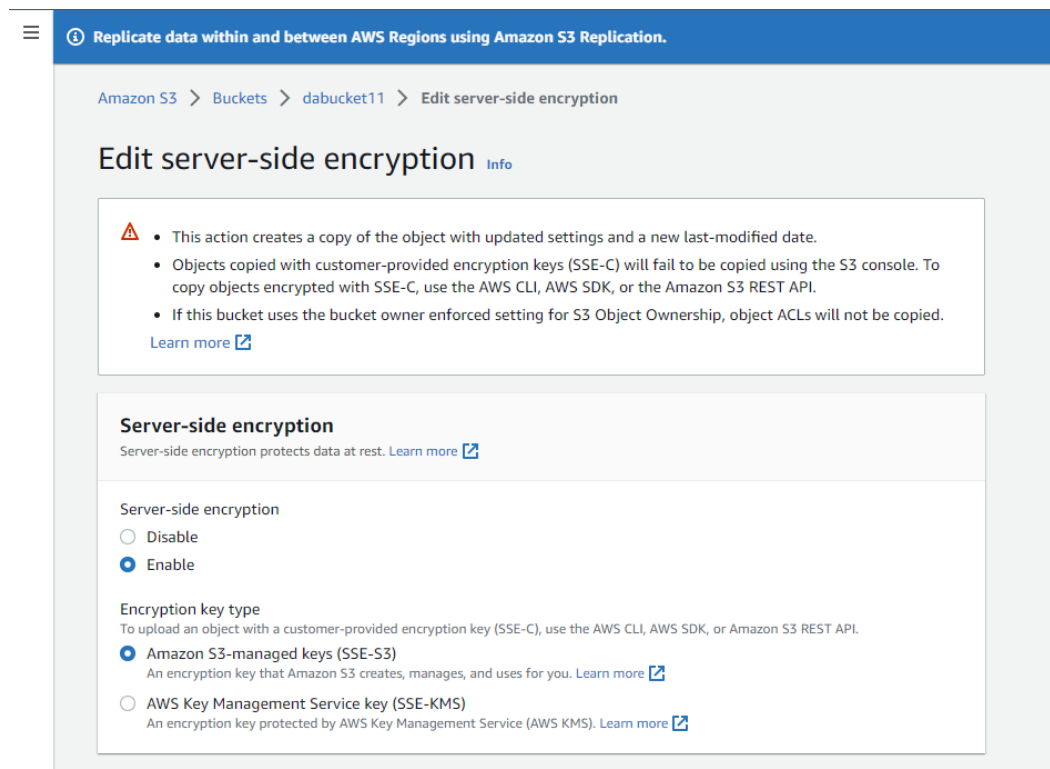
Task 2.2: Upload an object



Task 2.3: Query the object you uploaded



Task 2.4: Change the encryption properties and storage type



Edit storage class [Info](#)

- ⚠ This action creates a copy of the object with updated settings and a new last-modified date. You can change the storage class without making a new copy of the object using a [lifecycle rule](#).
- Objects copied with customer-provided encryption keys (SSE-C) will fail to be copied using the S3 console. To copy objects encrypted with SSE-C, use the AWS CLI, AWS SDK, or the Amazon S3 REST API.
- If this bucket uses the bucket owner enforced setting for S3 Object Ownership, object ACLs will not be copied.

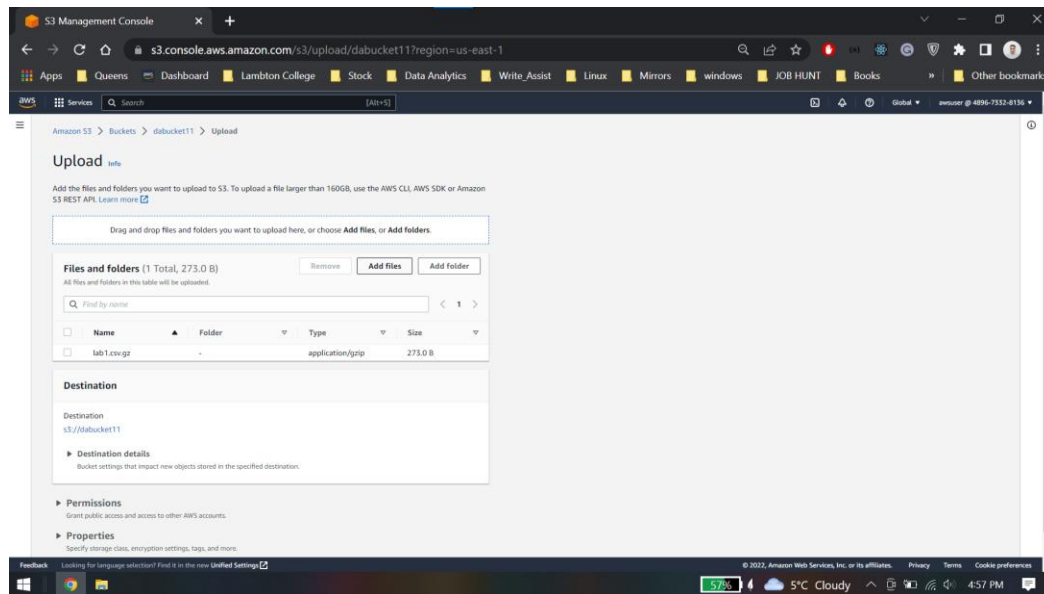
[Learn more](#)

Storage class

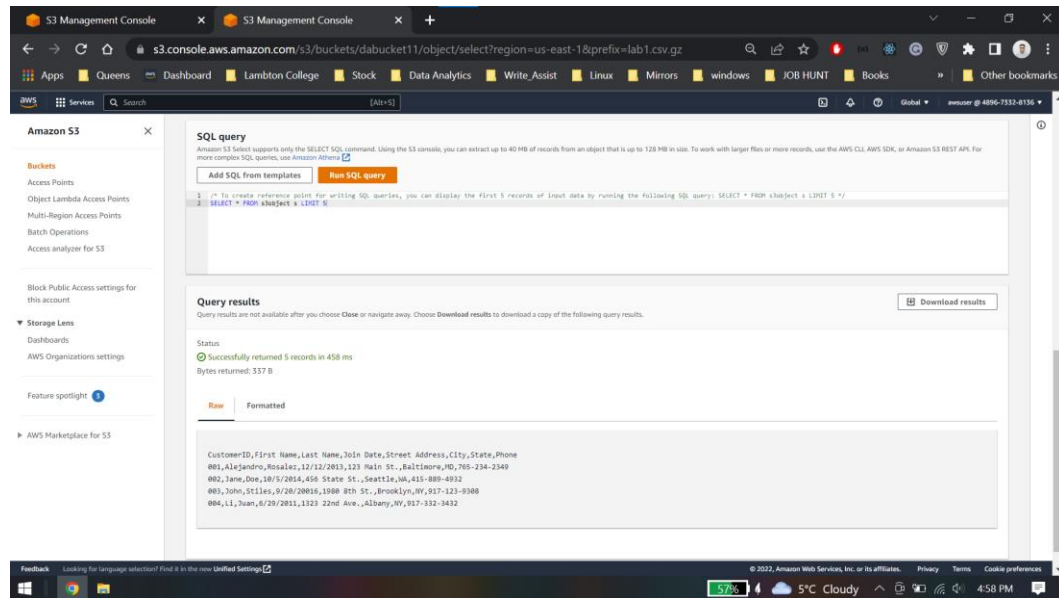
Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

	Storage class	Designed for	Availability Zones	Min storage duration	Transfer acceleration
<input type="radio"/>	Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-
<input checked="" type="radio"/>	Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-
<input type="radio"/>	Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1

Task 2.5: Upload a compressed file



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



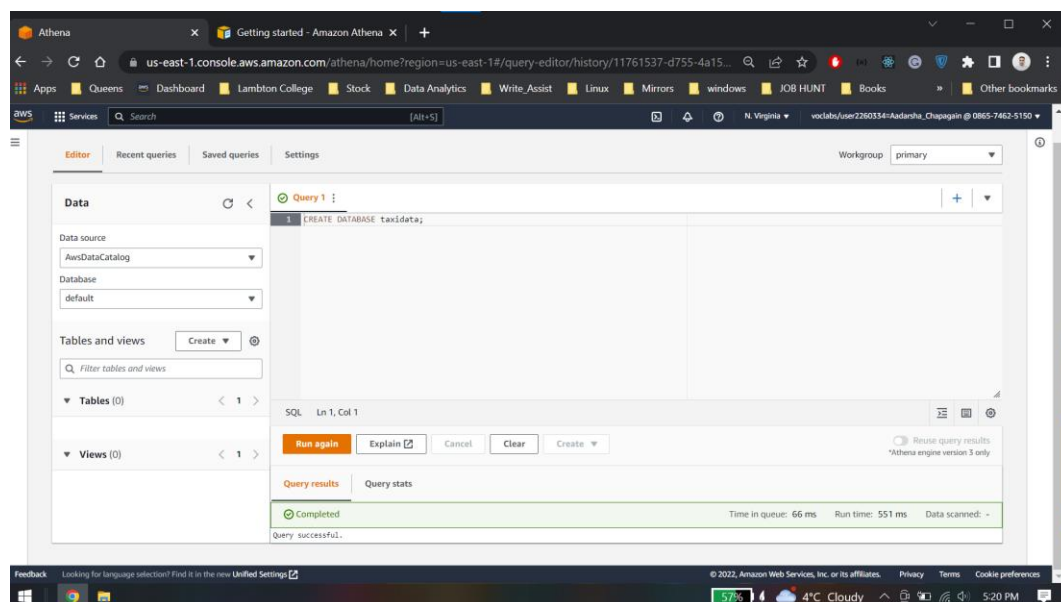
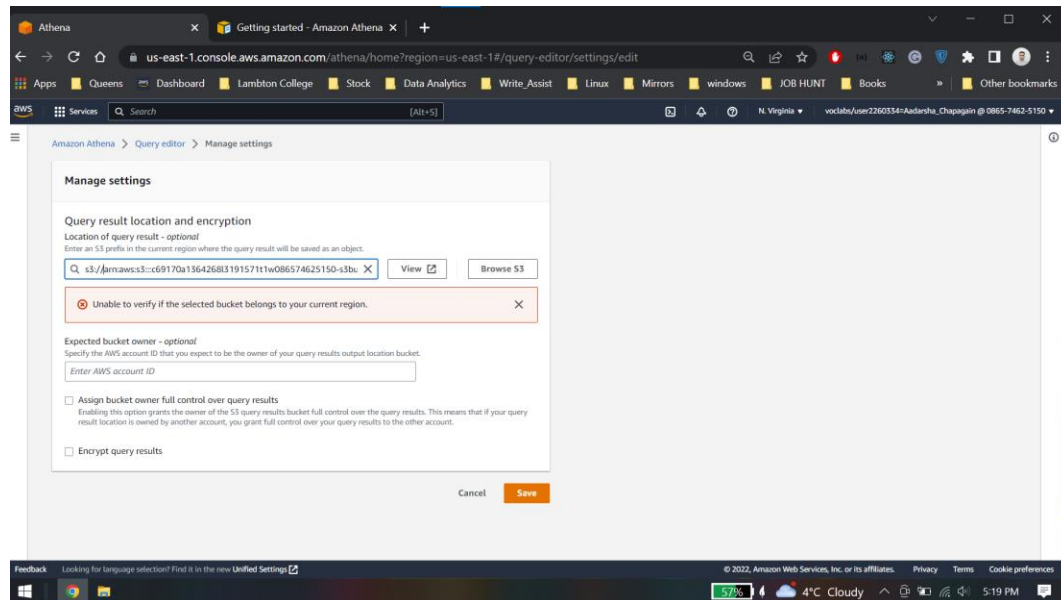
Lab 1 Conclusion.

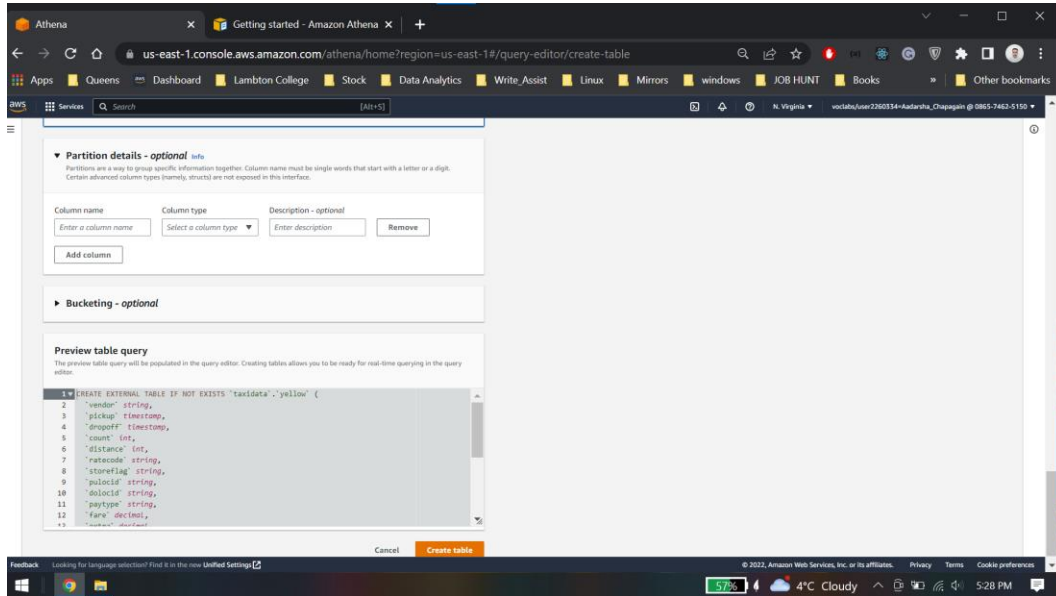
- Access Amazon S3 in the AWS Management Console
- Secure an S3 bucket with IAM
- Create a bucket with Amazon S3
- Load data into an S3 bucket
- Query an S3 bucket

Lab 2: Query Data in Amazon Athena

Task 1 : Query Data in Amazon Athena

Bucket ARN: arn:aws:s3:::c69170a136426813191571t1w086574625150-s3bucket-q3gdz12cykj





Preview table query

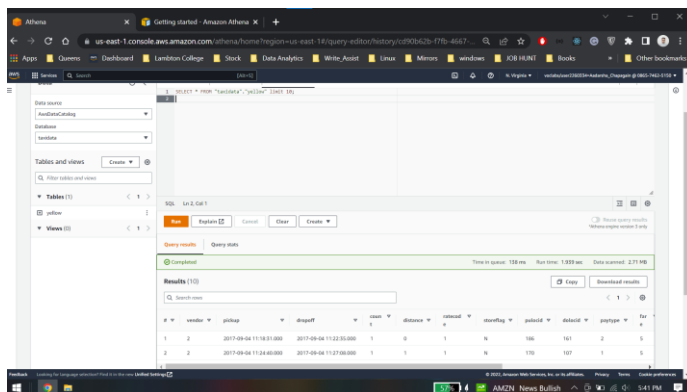
The preview table query will be populated in the query editor. Creating tables allows you to be ready for real-time querying in the query editor.

```

13
14 `mta_tax` decimal,
15 `tip` decimal,
16 `tolls` decimal,
17 `surcharge` decimal,
18 `total` decimal
19 )
20 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
21 WITH SERDEPROPERTIES ('field.delim' = ',')
22 STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat' OUTPUTFORMAT 'org.apache
23   .hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
24 LOCATION 's3://aws-tc-largeobjects/CUR-TF-200-ACBDF0-1/Lab2/yellow/'
25 TBLPROPERTIES ('classification' = 'csv');
  
```

Cancel

Create table



Task 2: Optimize the database

Task 2.1: Create a table for the January 2017 data

The screenshot shows the Amazon Athena Query Editor interface. The 'Data' panel on the left indicates the data source is 'AwsDataCatalog', the database is 'taxidata', and the table is 'taxi'. The 'Tables and views' panel shows a table named 'taxi' with columns 'jan' and 'yellow'. The 'Query editor' panel shows a SQL query that creates a table named 'taxi' with columns 'paytype', 'fare', 'extra', 'eta_tax', 'tip', 'tolls', 'surcharge', and 'total'. The query is executed successfully, and the 'Query results' panel shows the query status as 'Completed' with a run time of 913 ms.

```
11 'paytype' string,
12 'fare' decimal,
13 'extra' decimal,
14 'eta_tax' decimal,
15 'tip' decimal,
16 'tolls' decimal,
17 'surcharge' decimal,
18 'total' decimal
19 }
20 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
21 WITH SERDEPROPERTIES (
22   'serialization.format' = ',',
23   'field.delim' = ','
24 ) LOCATION 's3://aws-tc-largedataobjects/US-TF-200-ACBDF0-1/Lab2/January2017/'
25 TBLPROPERTIES ('has_encrypted_data'='false');
```

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

The screenshot shows the Amazon Athena Query Editor interface. The 'Data' panel on the left indicates the data source is 'AwsDataCatalog', the database is 'taxidata', and the table is 'taxi'. The 'Query editor' panel shows a SQL query that filters data by trip date and time. The query is executed successfully, and the 'Query results' panel shows the query status as 'Completed' with a run time of 9,532 ms. The results table shows 21 rows of data.

```
1 SELECT count(*) AS "Number of trips",
2       sum (total) AS "Total fares",
3       pickup AS "Trip date"
4 FROM yellow taxi
5 WHERE (pickup >= '2015-01-01 00:00:00'
6       and (pickup <= '2015-01-01 00:00:00'))
7 GROUP BY pickup;
```

#	Number of trips	Total fares	Trip date
1	1	27	2015-01-01 00:00:00
2	1	8	2015-01-01 00:00:00
3	1	13	2015-01-01 00:00:00
4	1	13	2015-01-01 00:00:00
5	1	20	2015-01-01 00:00:00
6	1	42	2015-01-01 00:00:00

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

The screenshot shows the Amazon Athena console interface. The query editor displays a SQL query that counts the number of trips for each pickup location, grouped by pickup date. The query results are shown in a table with columns: #, Number of trips, Total fares, and Trip date. The results are sorted by Total fares in descending order.

#	Number of trips	Total fares	Trip date
1	3	62	2017-01-01 07:51:25.000
2	3	75	2017-01-01 07:53:41.000
3	2	14	2017-01-01 07:55:56.000
4	6	86	2017-01-01 07:56:57.000
5	1	23	2017-01-01 07:56:46.000
6	3	27	2017-01-01 07:56:52.000

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

The screenshot shows the Amazon Athena console interface. The query editor displays a SQL query that creates a table named 'taxidata.creditcard' and partitions it by 'paytype'. The query results are shown in a table with columns: #, Number of trips, Total fares, and Trip date. The results are sorted by Total fares in descending order.

```
1 CREATE TABLE taxidata.creditcard
2 WITH (
3   format = 'PARQUET'
4 )AS
5 SELECT * from "yellow"
6 WHERE paytype = '1';
```

```
1 SELECT sum (total), paytype FROM yellow
2 WHERE paytype = '1' GROUP BY paytype;
```

SQL Ln 2, Col 41

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results
*Athena engine version 3 only

[Query results](#) | [Query stats](#)

✓ Completed Time in queue: 243 ms Run time: 7.472 sec Data scanned: 9.32 GB

```
1 SELECT sum (total), paytype FROM creditcard
2 WHERE paytype = '1' GROUP BY paytype;
```

SQL Ln 2, Col 41

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results
*Athena engine version 3 only

[Query results](#) | [Query stats](#)

✓ Completed Time in queue: 146 ms Run time: 1.927 sec Data scanned: 73.22 MB

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

1

CREATE VIEW cctrips AS

2

SELECT "sum"("fare") "CreditCardFares"

3

FROM yellow

4

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

1

CREATE VIEW cashtrips AS

2

SELECT "sum"("fare") "CashFares"

3

FROM yellow

4

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

:

Query 9 : X

Query 10 : X

+ ▼

1

SELECT * FROM "taxidata"."comparepay" limit 10;

SQLLn 1, Col 1

Run again

Explain

Cancel

Clear

Create ▼

Reuse query results
*Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 169 ms

Run time: 9.326 sec

Data scanned: 18.64 GB

Results (2)

Copy

Download results

Q Search rows

< 1 > ⌕

# ▼	ccttotal ▼	cashttotal ▼
1	584502884	250849783
2	460097126	199181978

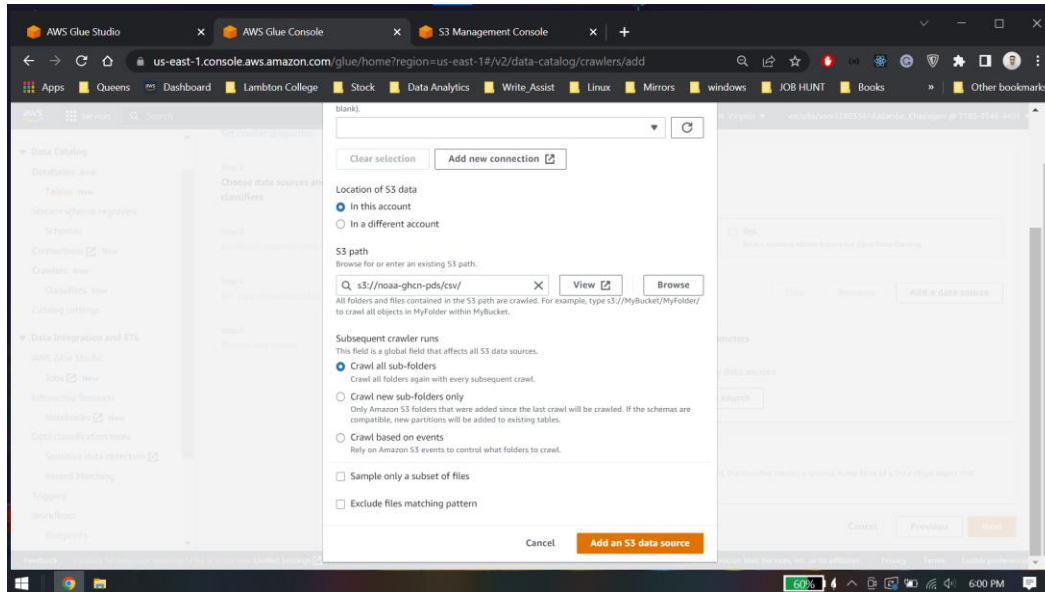
Lab2 Conclusion

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

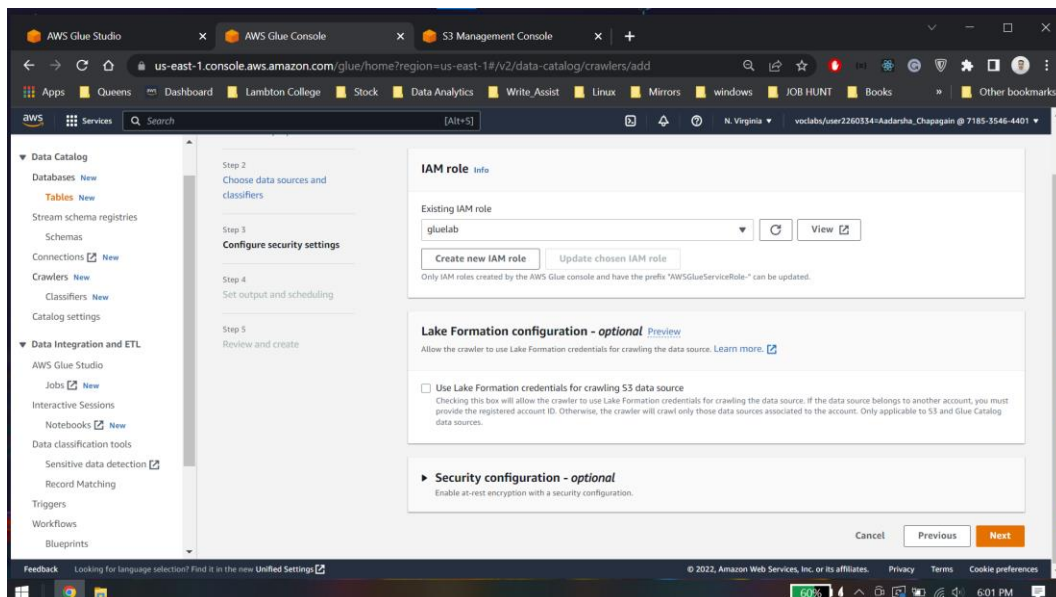
Lab 3: Query Data in Amazon S3 with Amazon Athena and AWS Glue

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

Create a Crawler



Choose Iam role



Create a database

The screenshot shows the AWS Glue console interface for creating a new database. The left sidebar contains navigation links for Data Catalog, Data Integration and ETL, and various AWS services. The main content area is titled 'Create a database' and includes a 'Database details' form. The form has three input fields: 'Name' (containing 'weatherdata'), 'Location - optional' (empty), and 'Description - optional' (containing 'Enter text'). Below the form are 'Cancel' and 'Create database' buttons. The browser's address bar shows the URL 'us-east-1.console.aws.amazon.com/glue/home?region=us-east-1/v2/data-catalog/databases/add'.

Create a database
Create a database in the AWS Glue Data Catalog.

Database details

Name:
Database name is required, in lowercase characters, and no longer than 255 characters.

Location - optional
Set the URI location for use by clients of the Data Catalog.

Description - optional
Enter text

Descriptions can be up to 2048 characters long.

[Cancel](#) [Create database](#)

Output Configuration

The screenshot shows the 'Output configuration' step in the AWS Glue console. The left sidebar is the same as in the previous image. The main content area is titled 'Output configuration' and includes a 'Target database' dropdown menu (set to 'weatherdata'), a 'Table name prefix - optional' field, and a 'Maximum table threshold - optional' field. Below these are 'Advanced options' and a 'Crawler schedule' section with a 'Frequency' dropdown menu (set to 'On demand'). At the bottom are 'Cancel', 'Previous', and 'Next' buttons. The browser's address bar shows the URL 'us-east-1.console.aws.amazon.com/glue/home?region=us-east-1/v2/data-catalog/crawlers/add'.

Output configuration

Target database:
[Clear selection](#) [Add database](#)

Table name prefix - optional
Type a prefix added to table names

Maximum table threshold - optional
This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.
Type a number greater than 0

Advanced options

Crawler schedule
You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron syntax. [Learn more](#)

Frequency:

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

Task 1.1: Run the crawler

The screenshot shows the AWS Glue Crawlers console. At the top, a green banner states: "Crawler successfully starting. The following crawler is now starting: 'Weather'". Below this, the breadcrumb "AWS Glue > Crawlers" is visible. The main heading is "Crawlers", followed by a description: "A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog." The last update timestamp is "November 27, 2022 at 23:09:58 (UTC)".

The "Crawlers (1/1) Info" section includes a link to "View and manage all available crawlers." and a search bar labeled "Filter crawlers". To the right are buttons for "Refresh", "Action" (with a dropdown arrow), "Run", and "Create crawler".

Below is a table listing the crawler:

<input checked="" type="checkbox"/>	Name	State	Schedule	Last run	Log	Table changes from last ...
<input checked="" type="checkbox"/>	Weather	Running		-	-	-

Task 1.2: Review the metadata created by AWS Glue

CSV
Version 0 (Current version) [🔄](#) [Actions ▼](#)

Page last updated: November 27, 2022 at 23:12:34 (UTC)

Table details
Advanced properties

Name <code>cw</code>	Description -	Database weatherdata	Classification <code>cw</code>
Location <code>s3://wsaa-gfcm-pub/us/</code>	Connection -	Deprecated -	Last updated November 27, 2022 at 23:12:34
Input format <code>org.apache.hadoop.mapred.TextInputFormat</code>	Output format <code>org.apache.hadoop.hive ql io.HiveHadoopKeyTextOutputFormat</code>	Serde serialization lib <code>org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe</code>	

Schema
Partitions
Indexes

Schema (0)
View and manage the table schema.

< 1 >

#	Column name	Data type	Partition key	Comment
1	id	string	-	-
2	date	bigint	-	-
3	element	string	-	-
4	data_value	bigint	-	-
5	m_flag	string	-	-
6	e_flag	string	-	-
7	s_flag	string	-	-
8	obs_time	bigint	-	-

The screenshot shows the AWS Glue console interface for editing a CSV schema. The main content area is titled 'Edit schema: csv' and displays a table schema with 9 columns. The 'Time' column is selected. The left sidebar contains navigation options for Data Catalog, Databases, Tables, Stream schema registries, Schemas, Connections, Crawlers, Classifiers, Catalog settings, Data Integration and ETL, AWS Glue Studio, Jobs, Interactive Sessions, Notebooks, Data classification tools, Sensitive data detection, Record Matching, and Triggers. The top navigation bar shows the AWS Glue Console and S3 Management Console tabs.

Edit schema: csv

Schema (1/9)

View and manage the table schema.

Filter schemas

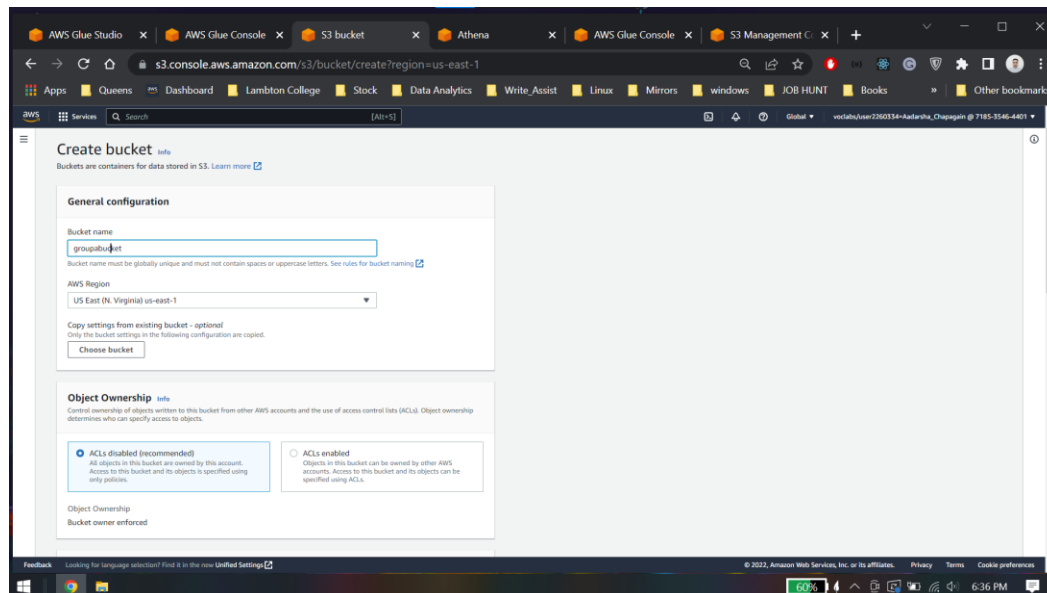
	#	Column name	Data type	Partition key	Comment
<input type="checkbox"/>	1	id	string	-	-
<input type="checkbox"/>	2	date	bigint	-	-
<input type="checkbox"/>	3	Type	string	-	-
<input type="checkbox"/>	4	Observation	bigint	-	-
<input type="checkbox"/>	5	MFlag	string	-	-
<input type="checkbox"/>	6	QFlag	string	-	-
<input type="checkbox"/>	7	SFlag	string	-	-
<input checked="" type="checkbox"/>	8	Time	bigint	-	-
<input type="checkbox"/>	9	partition_0	string	Partition (0)	-

Cancel Save as new table version

The screenshot displays the AWS Glue Console interface. At the top, there are tabs for various AWS services: AWS Glue Studio, AWS Glue Console, Athena, and S3 Management. The main content area shows the Athena query editor. The query being executed is: `SELECT * FROM "weatherdata"."csv" limit 10;`. The query status is 'Completed'. Below the query editor, the results are displayed in a table format. The table has 10 columns: #, id, date, type, observation, mflag, qflag, sflag, time, and partition_0. The first row of data is: 1, EEZ001000B2, 17820101, TMAX, -19, E, by_year. The console also shows the 'Tables and views' section on the left, with a search bar and a list of tables. The bottom of the screen shows the Windows taskbar with the time 6:30 PM and a battery level of 60%.

Task 2.1: Create a table for data after 1950

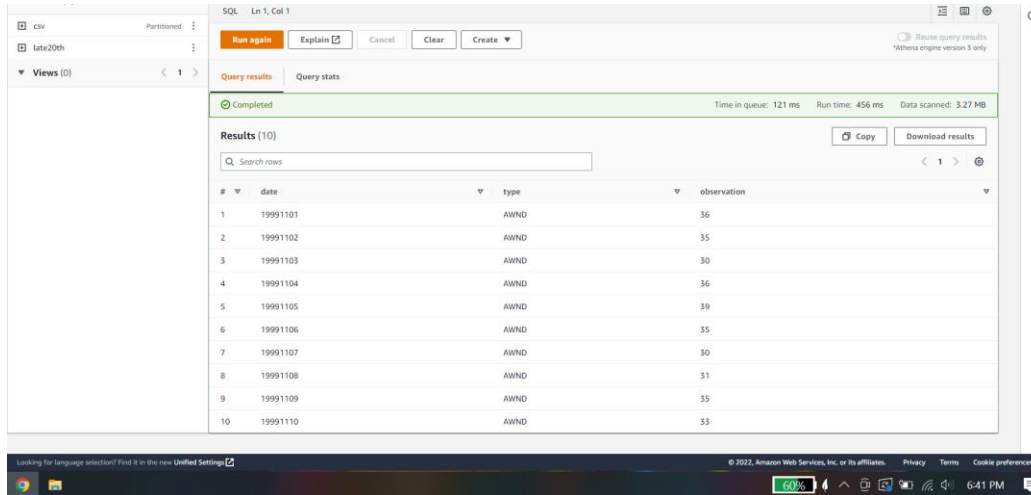
Create a bucket in same region



Create a table specifying the bucket location



Preview table

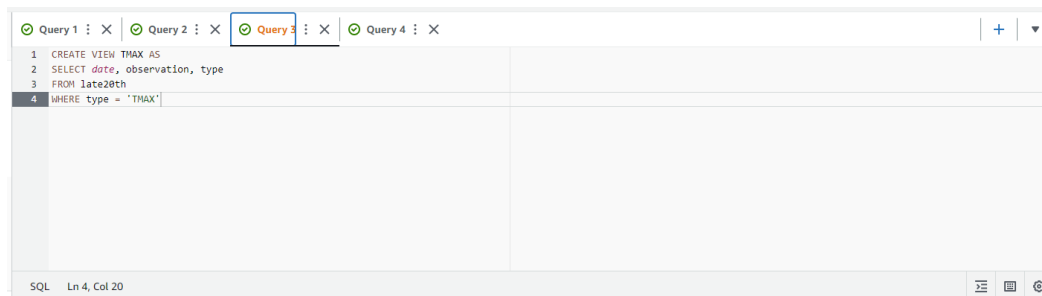


The screenshot shows the AWS Athena console interface. On the left, there's a sidebar with 'CSV' and 'late20th' datasets. The main area displays a query result for 'Ln 1, Col 1'. The query is 'SELECT * FROM late20th'. The results are shown in a table with 10 rows. The columns are '#', 'date', 'type', and 'observation'. The data shows dates from 19991101 to 19991110, all with 'type' 'AWND' and 'observation' values ranging from 30 to 39. The console also shows query statistics: 'Time in queue: 121 ms', 'Run time: 456 ms', and 'Data scanned: 3.27 MB'. There are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. A 'Query results' tab is active, and a 'Query stats' tab is also visible. A 'Copy' button and a 'Download results' button are present. A search bar for rows is also visible.

#	date	type	observation
1	19991101	AWND	36
2	19991102	AWND	35
3	19991103	AWND	30
4	19991104	AWND	36
5	19991105	AWND	39
6	19991106	AWND	35
7	19991107	AWND	30
8	19991108	AWND	31
9	19991109	AWND	35
10	19991110	AWND	33

Task 2.2: Run a query from the selected data

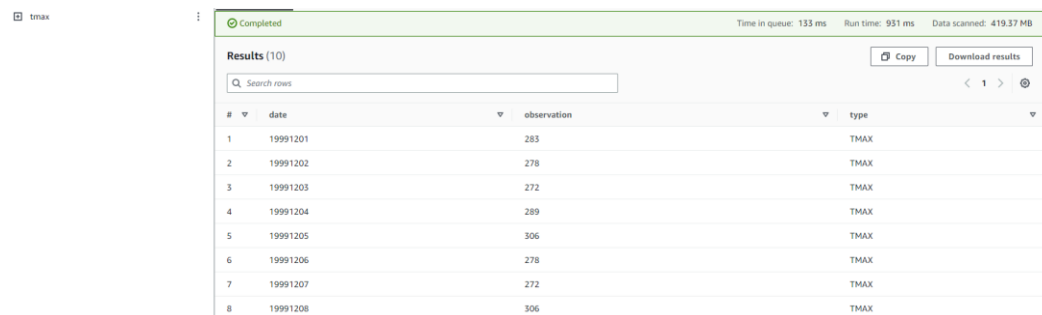
Create a view



The screenshot shows the AWS Athena console interface. The top bar shows 'Query 1', 'Query 2', 'Query 3', and 'Query 4'. The SQL editor is active, showing a query to create a view 'TMAX' from the 'late20th' dataset. The query is: 'CREATE VIEW TMAX AS SELECT date, observation, type FROM late20th WHERE type = 'TMAX''. The console also shows the query result: 'Ln 4, Col 20'.

```
1 CREATE VIEW TMAX AS
2 SELECT date, observation, type
3 FROM late20th
4 WHERE type = 'TMAX'
```

Preview the data



The screenshot shows the AWS Athena console interface. The top bar shows 'Query 1', 'Query 2', 'Query 3', and 'Query 4'. The SQL editor is active, showing a query to create a view 'TMAX' from the 'late20th' dataset. The query is: 'CREATE VIEW TMAX AS SELECT date, observation, type FROM late20th WHERE type = 'TMAX''. The console also shows the query result: 'Ln 4, Col 20'.

#	date	observation	type
1	19991201	283	TMAX
2	19991202	278	TMAX
3	19991203	272	TMAX
4	19991204	289	TMAX
5	19991205	306	TMAX
6	19991206	278	TMAX
7	19991207	272	TMAX
8	19991208	306	TMAX

Average maximum temperature from 1950 to 2018

The screenshot shows the AWS Athena console interface. On the left, the 'Data' sidebar is visible with 'Data source' set to 'AwsDataCatalog' and 'Database' set to 'weatherdata'. The 'Tables and views' section shows a list of tables, including 'csv' and 'late20th'. The main query editor displays a SQL query:

```
1 SELECT date/10000 as Year, avg(observation)/10 as Max
2 FROM tmax
3 GROUP BY date/10000 ORDER BY date/10000;
```

The query is currently in a 'Running' state. The bottom right corner indicates 'Reuse query results' and '*Athena engine version 3 only'.

Results

The screenshot shows the 'Query results' tab in the AWS Athena console. The query is 'Completed'. The results are displayed in a table with 66 rows. The table has columns for '#', 'Year', and 'Max'. The data shows the average maximum temperature for each year from 1950 to 1962.

#	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