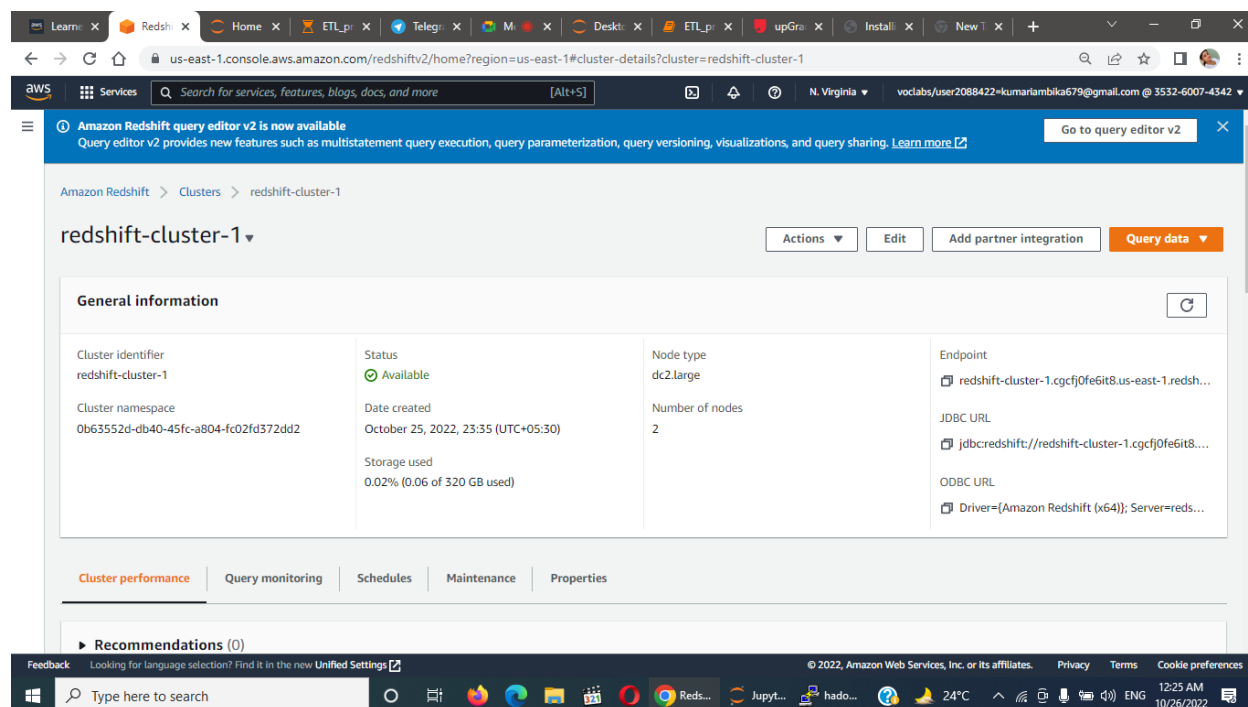


## Creation of a Redshift Cluster

### Screenshots of the configuration of the Redshift cluster that you have created:

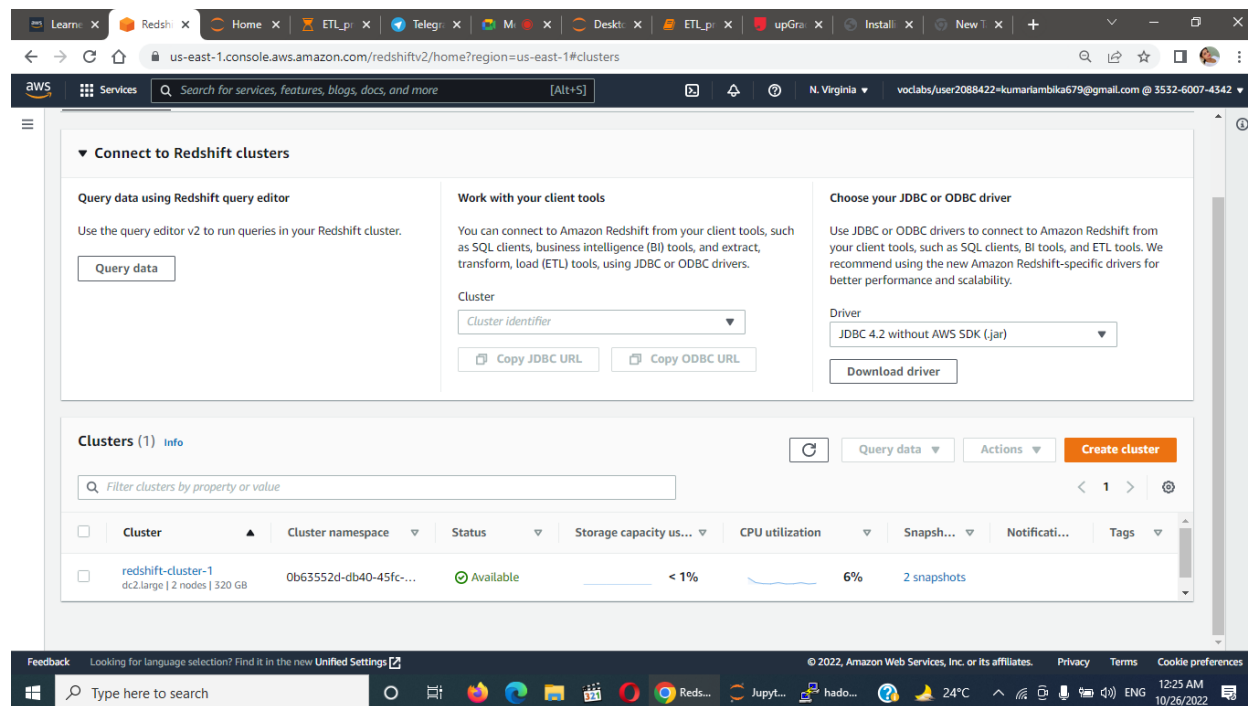
<Screenshot of the type of machine used along with number of nodes>



The screenshot shows the Amazon Redshift console interface. The top navigation bar includes the AWS logo, a search bar, and the user's profile. The main content area displays the details for the cluster 'redshift-cluster-1'. The 'General information' tab is selected, showing the following details:

Cluster identifier	Status	Node type	Endpoint
redshift-cluster-1	Available	dc2.large	redshift-cluster-1.cgcf0fe6it8.us-east-1.redsh...
Cluster namespace	Date created	Number of nodes	JDBC URL
0b63552d-db40-45fc-a804-fc02fd372dd2	October 25, 2022, 23:35 (UTC+05:30)	2	jdbc:redshift://redshift-cluster-1.cgcf0fe6it8...
Storage used			ODBC URL
0.02% (0.06 of 320 GB used)			Driver={Amazon Redshift (x64)}; Server=reds...

Below the general information, there are tabs for 'Cluster performance', 'Query monitoring', 'Schedules', 'Maintenance', and 'Properties'. A 'Recommendations' section is also visible at the bottom.



The screenshot shows the 'Connect to Redshift clusters' section of the Amazon Redshift console. It provides options to connect to the cluster using the Redshift query editor, client tools, or JDBC/ODBC drivers.

**Query data using Redshift query editor**

Use the query editor v2 to run queries in your Redshift cluster.

**Work with your client tools**

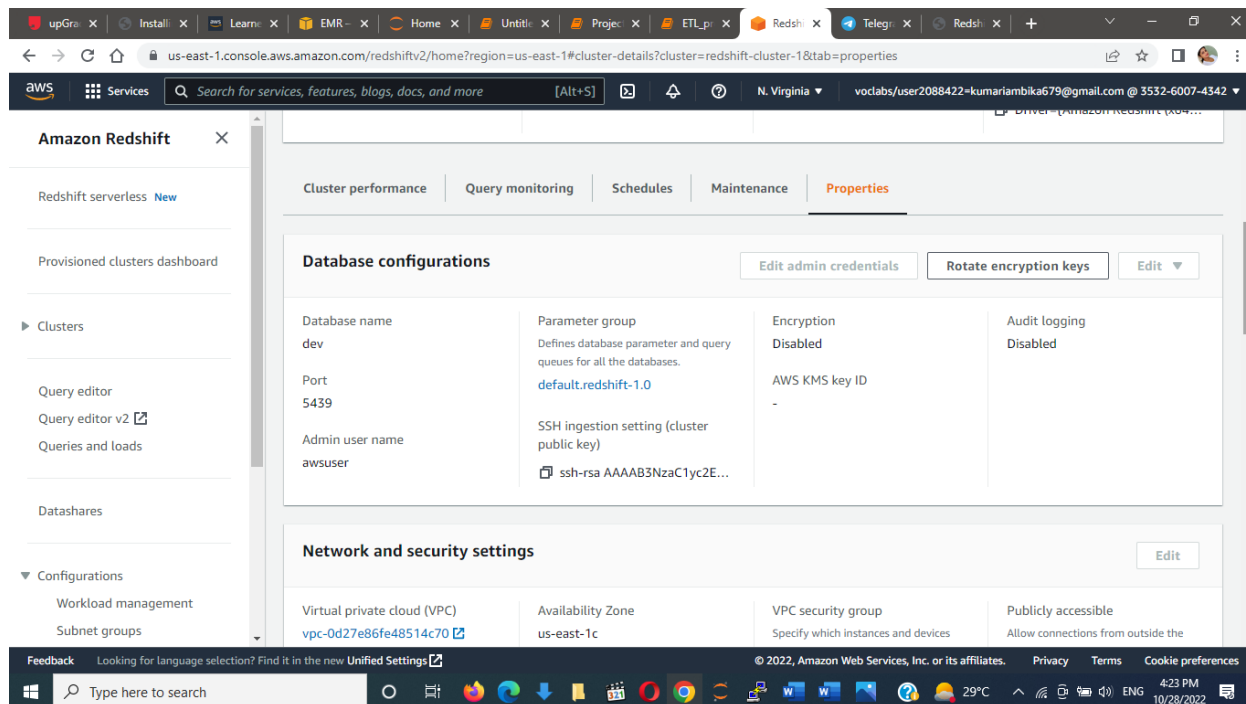
You can connect to Amazon Redshift from your client tools, such as SQL clients, business intelligence (BI) tools, and extract, transform, load (ETL) tools, using JDBC or ODBC drivers.

**Choose your JDBC or ODBC driver**

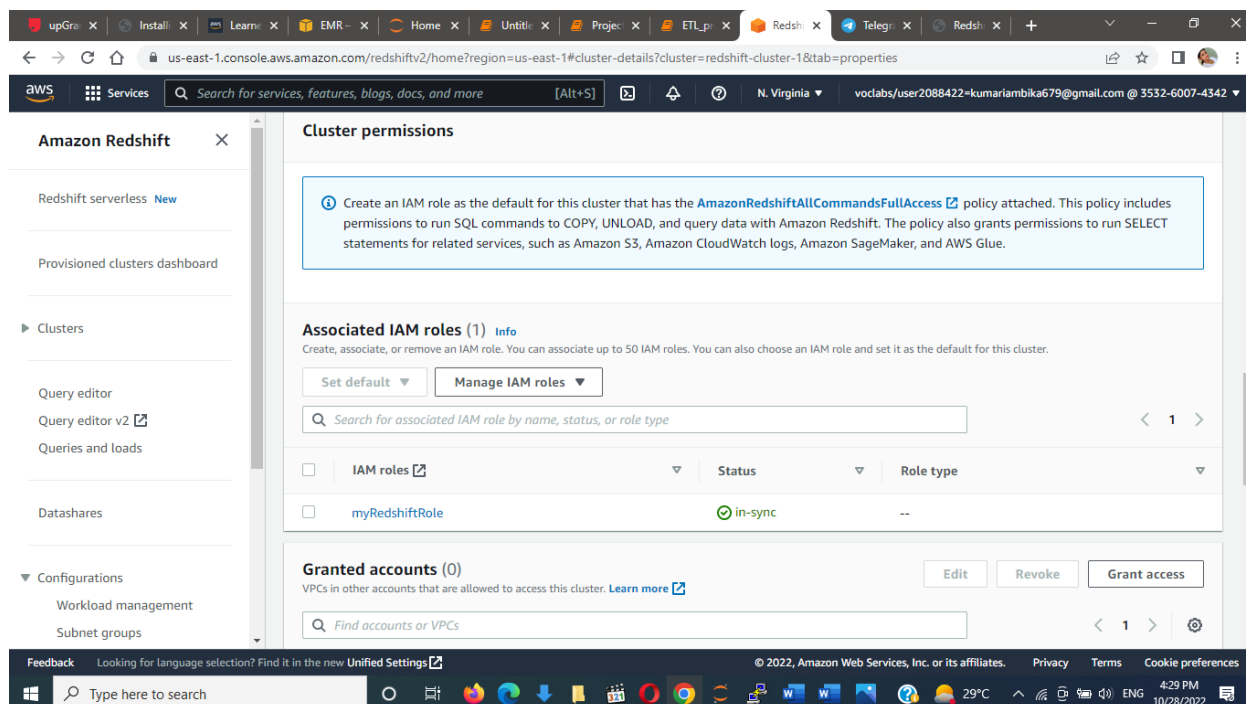
Use JDBC or ODBC drivers to connect to Amazon Redshift from your client tools, such as SQL clients, BI tools, and ETL tools. We recommend using the new Amazon Redshift-specific drivers for better performance and scalability.

The 'Clusters (1)' section shows a table with the following data:

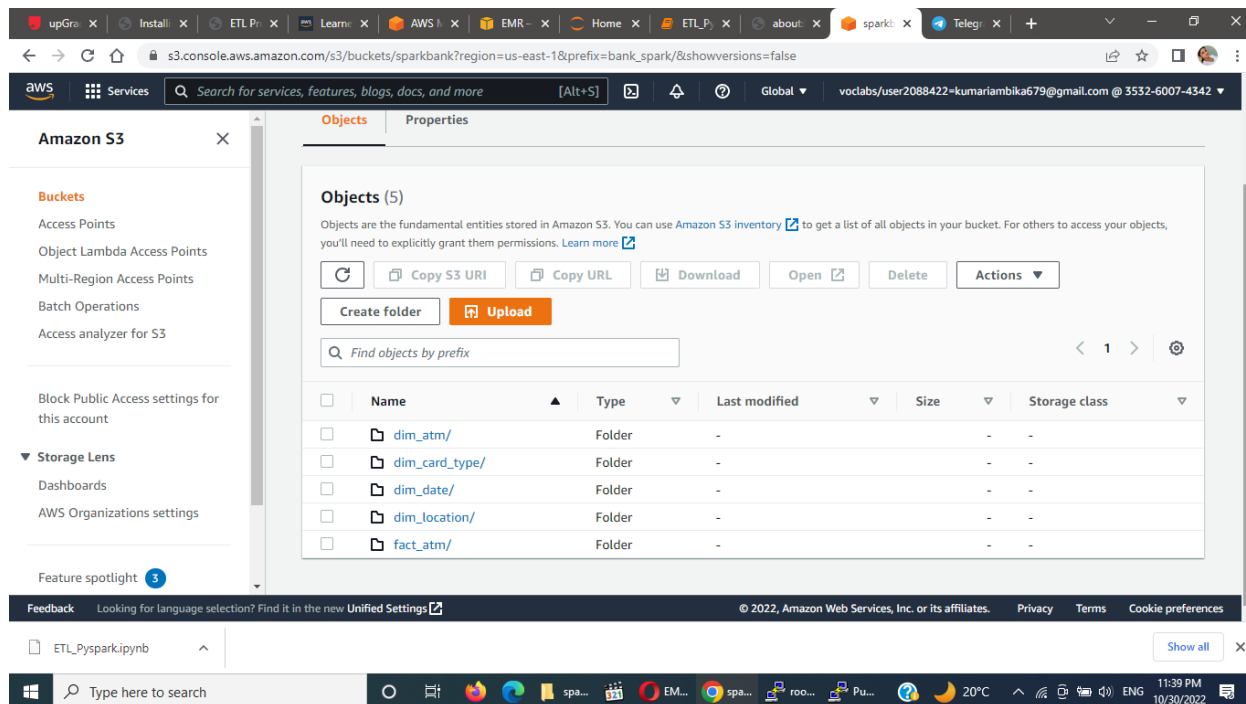
Cluster	Cluster namespace	Status	Storage capacity us...	CPU utilization	Snapsh...	Notificati...	Tags
redshift-cluster-1 dc2.large   2 nodes   320 GB	0b63552d-db40-45fc-...	Available	< 1%	6%	2 snapshots		



The screenshot shows the Amazon Redshift console interface. The left sidebar contains navigation options like 'Redshift serverless', 'Provisioned clusters dashboard', 'Clusters', 'Query editor', 'Query editor v2', 'Queries and loads', 'Datashares', 'Configurations', 'Workload management', and 'Subnet groups'. The main content area is titled 'Amazon Redshift' and shows the 'Properties' tab for a cluster. It includes sections for 'Database configurations' (Database name: dev, Port: 5439, Admin user name: awsuser, Parameter group: default.redshift-1.0, Encryption: Disabled, Audit logging: Disabled) and 'Network and security settings' (Virtual private cloud (VPC): vpc-0d27e86fe48514c70, Availability Zone: us-east-1c, VPC security group: Specify which instances and devices, Publicly accessible: Allow connections from outside the VPC).

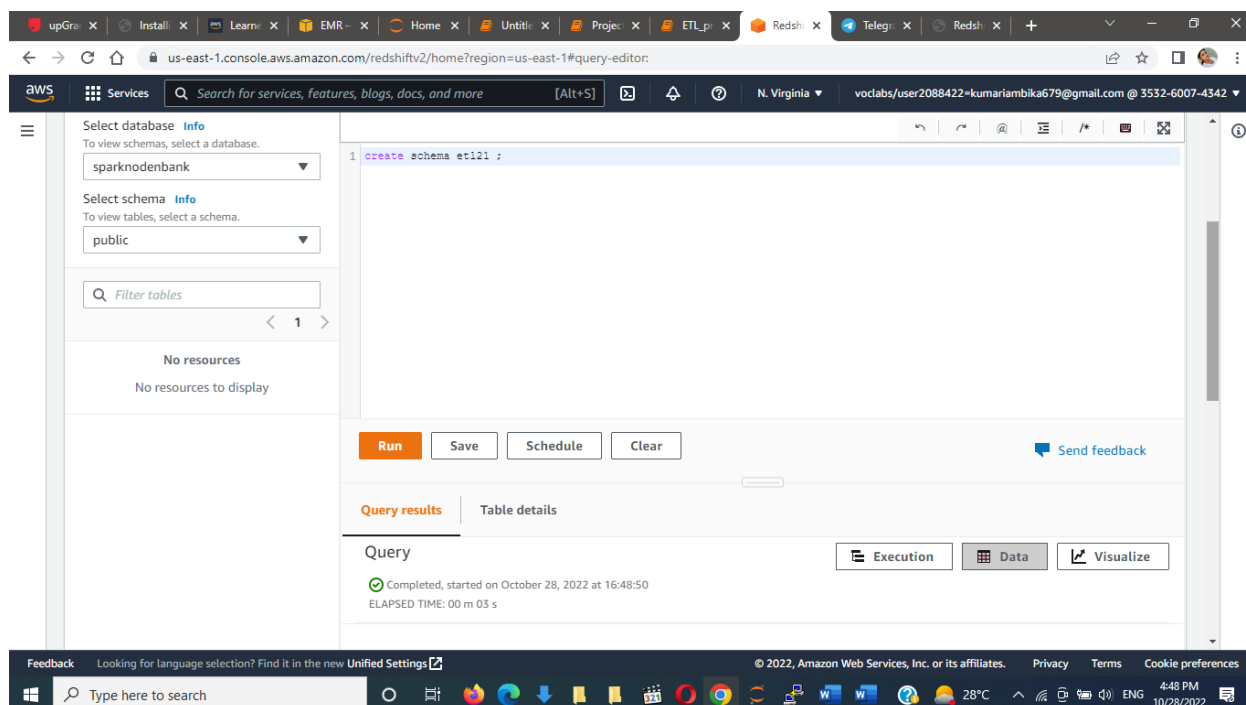


The screenshot shows the Amazon Redshift console interface, specifically the 'Cluster permissions' section. It includes a notification to create an IAM role with the 'AmazonRedshiftAllCommandsFullAccess' policy. Below this, there is a section for 'Associated IAM roles (1)' with a table showing the role 'myRedshiftRole' with status 'in-sync'. There is also a section for 'Granted accounts (0)' with a search bar and buttons for 'Edit', 'Revoke', and 'Grant access'.



Setting up a database in the Redshift cluster and running queries to create the dimension and fact tables

**Create schema etl21;**



Queries to create the various dimension and fact tables with appropriate primary and foreign keys:

- **Creating location dimension table**

create table etl21.DIM\_LOCATION

(

location\_id int not null DISTKEY SORTKEY,

location varchar(50),

streetname varchar(255),

street\_number int,

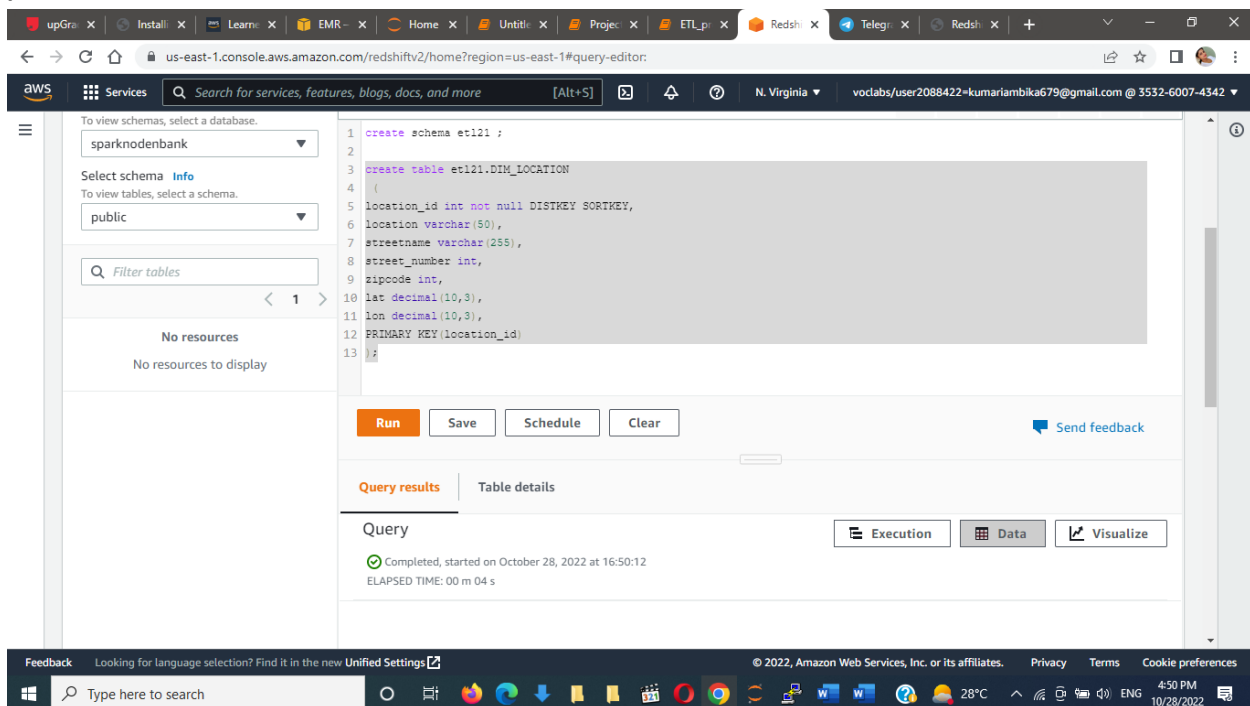
zipcode int,

lat decimal(10,3),

lon decimal(10,3),

PRIMARY KEY(location\_id)

);



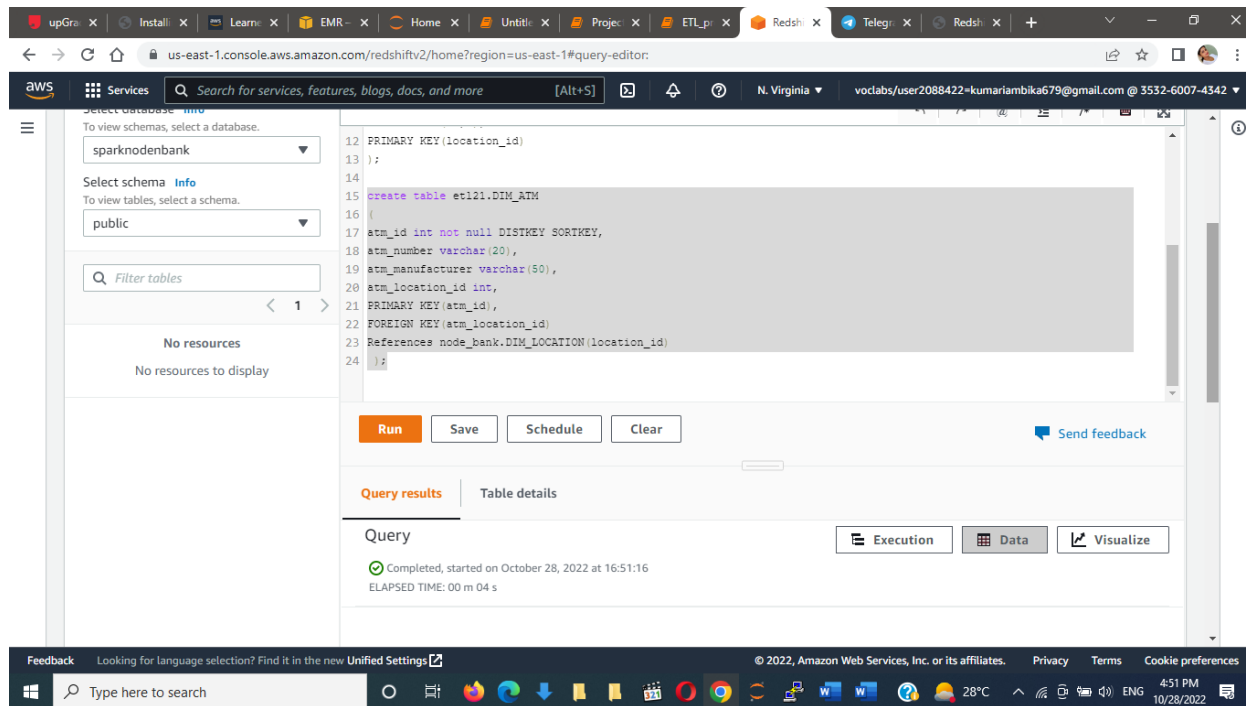
- **Creating atm dimension table**

create table etl21.DIM\_ATM

( atm\_id int not null DISTKEY SORTKEY,

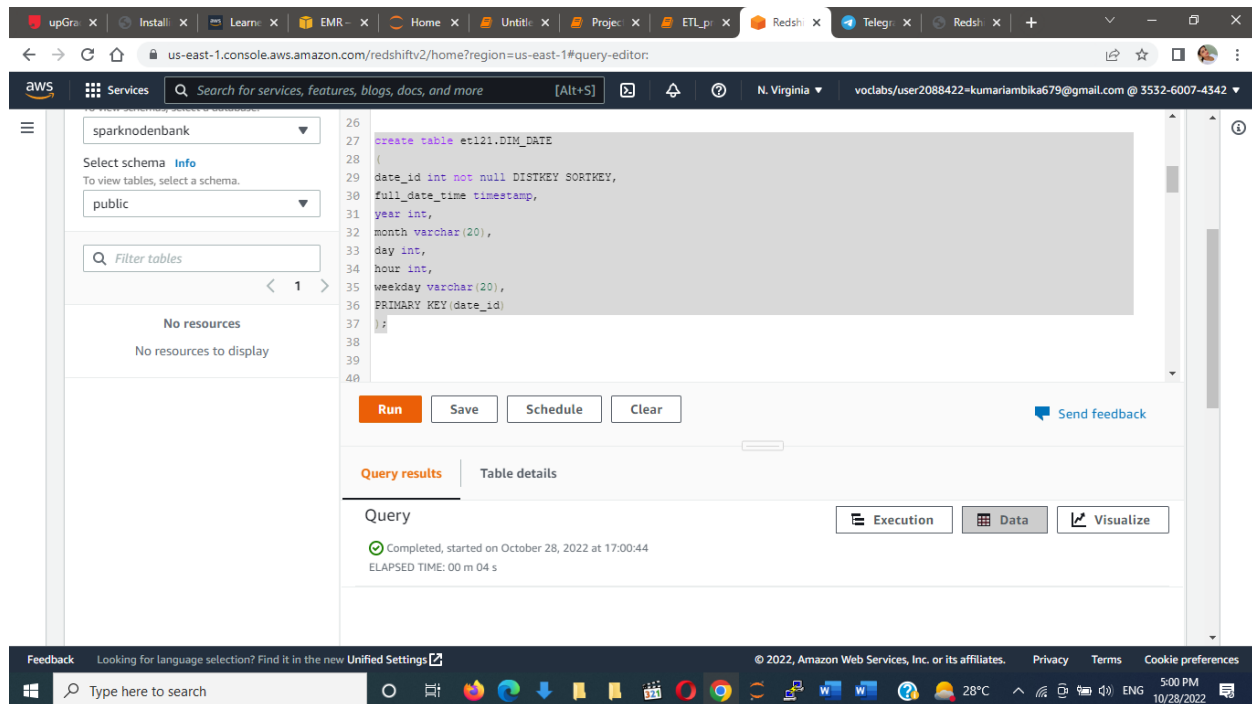
atm\_number varchar(20),

```
atm_manufacturer varchar(50),
atm_location_id int,
PRIMARY KEY(atm_id),
FOREIGN KEY(atm_location_id)
References nodebank.DIM_LOCATION(location_id)
);
```



## • Creating date dimension table

```
create table etl21.DIM_DATE
(
date_id int not null DISTKEY SORTKEY,
full_date_time timestamp,
year int,
month varchar(20),
day int,
hour int,
weekday varchar(20),
PRIMARY KEY(date_id));
```

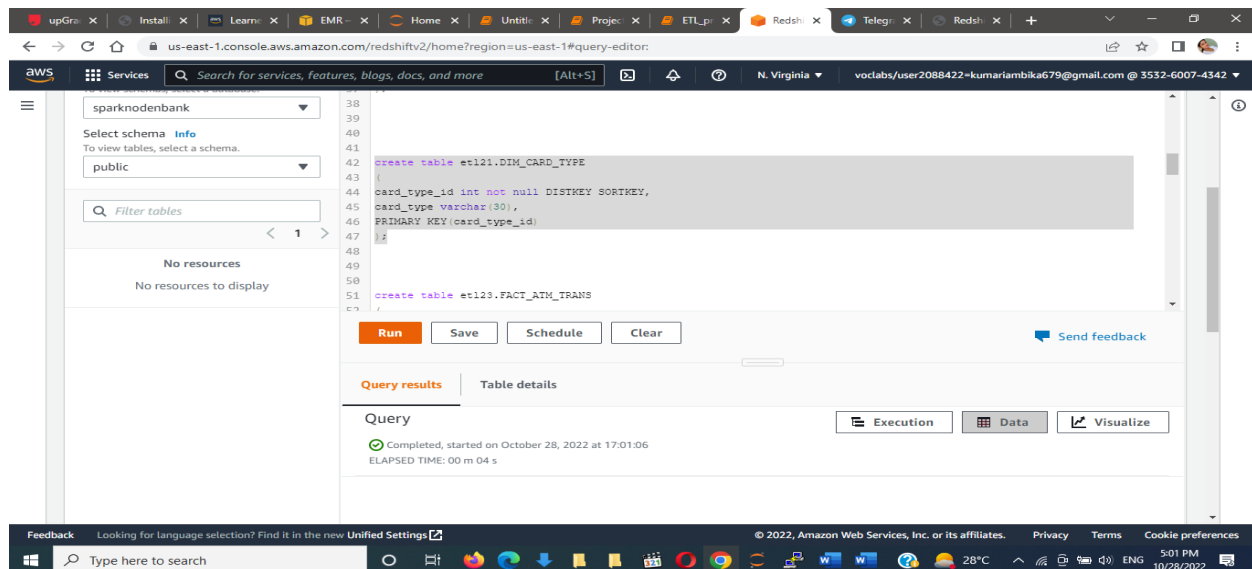


The screenshot shows the AWS Redshift console interface. On the left, there's a sidebar with 'sparknodenbank' selected and 'public' schema chosen. The main area displays a SQL query to create a table named 'etl21.DIM\_DATE' with columns: 'date\_id' (int, not null, DISTKEY SORTKEY, PRIMARY KEY), 'full\_date\_time' (timestamp), 'year' (int), 'month' (varchar(20)), 'day' (int), 'hour' (int), and 'weekday' (varchar(20)). The query is executed successfully, showing 'Completed, started on October 28, 2022 at 17:00:44' and 'ELAPSED TIME: 00 m 04 s'.

## • Creating card type dimension table

create table etl21.DIM\_CARD\_TYPE

```
(
card_type_id int not null DISTKEY SORTKEY,
card_type varchar(30),
PRIMARY KEY(card_type_id)
);
```



The screenshot shows the AWS Redshift console interface. On the left, there's a sidebar with 'sparknodenbank' selected and 'public' schema chosen. The main area displays a SQL query to create a table named 'etl21.DIM\_CARD\_TYPE' with columns: 'card\_type\_id' (int, not null, DISTKEY SORTKEY, PRIMARY KEY) and 'card\_type' (varchar(30)). Below this, there's a partial view of another query to create a table named 'etl21.FACT\_ATM\_TRANS'. The query is executed successfully, showing 'Completed, started on October 28, 2022 at 17:01:06' and 'ELAPSED TIME: 00 m 04 s'.

## • Creating atm transactions fact table

create table etl21.FACT\_ATM\_TRANS

(

trans\_id bigint not null DISTKEY SORTKEY,

atm\_id int,

weather\_loc\_id int,

date\_id int,

card\_type\_id int,

atm\_status varchar(20),

currency varchar(10),

service varchar(20),

transaction\_amount int,

message\_code varchar(225),

message\_text varchar(225),

rain\_3h decimal(10,3),

clouds\_all int,

weather\_id int,

weather\_main varchar(50),

weather\_description varchar(255),

PRIMARY KEY(trans\_id),

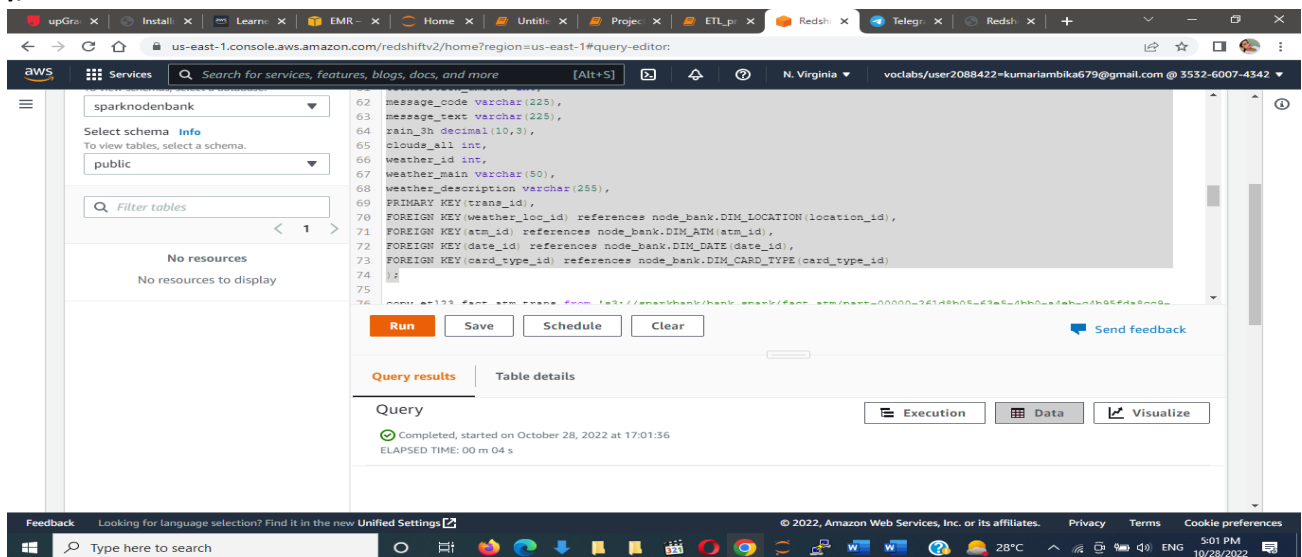
FOREIGN KEY(weather\_loc\_id) references etl21.DIM\_LOCATION(location\_id),

FOREIGN KEY(atm\_id) references etl21.DIM\_ATM(atm\_id),

FOREIGN KEY(date\_id) references etl21.DIM\_DATE(date\_id),

FOREIGN KEY(card\_type\_id) references etl21.DIM\_CARD\_TYPE(card\_type\_id)

);



The screenshot shows the AWS Redshift console interface. On the left, the 'sparknodbank' database is selected, and the 'public' schema is chosen. The main area displays the SQL query for creating the 'FACT\_ATM\_TRANS' table. The query is as follows:

```

62 message_code varchar(225),
63 message_text varchar(225),
64 rain_3h decimal(10,3),
65 clouds_all int,
66 weather_id int,
67 weather_main varchar(50),
68 weather_description varchar(255),
69 PRIMARY KEY(trans_id),
70 FOREIGN KEY(weather_loc_id) references node_bank.DIM_LOCATION(location_id),
71 FOREIGN KEY(atm_id) references node_bank.DIM_ATM(atm_id),
72 FOREIGN KEY(date_id) references node_bank.DIM_DATE(date_id),
73 FOREIGN KEY(card_type_id) references node_bank.DIM_CARD_TYPE(card_type_id)
74 );
75
76 copy etl21.fact_atm-trans from 's3://awsbank/bank-spark/fact_atm/part-00000-16148b06-63a6-4bb0-c4eb-c6b95fde80c9-
  
```

The query was executed successfully. The status is 'Completed, started on October 28, 2022 at 17:01:36' with an 'ELAPSED TIME: 00 m 04 s'. The 'Query results' tab is active, showing the execution details.

## Loading data into a Redshift cluster from Amazon S3 bucket

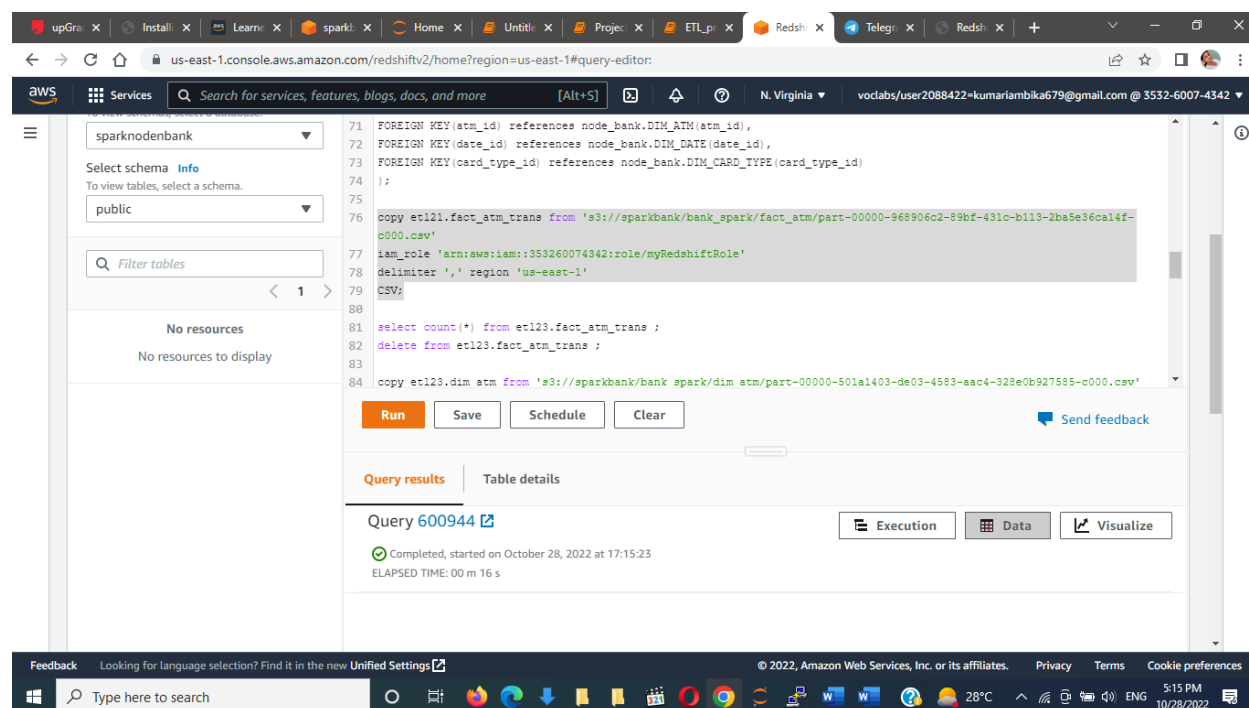
Queries to copy the data from S3 buckets to the Redshift cluster in the appropriate tables

**copy etl21.fact\_atm\_trans from 's3://sparkbank/bank\_spark/fact\_atm/part-00000-968906c2-89bf-431c-b113-2ba5e36ca14f-c000.csv'**

**iam\_role 'arn:aws:iam::353260074342:role/myRedshiftRole'**

**delimiter ',' region 'us-east-1'**

**CSV;**



The screenshot shows the AWS Redshift console interface. On the left, there's a sidebar with a search bar and a list of tables under the 'public' schema. The main area displays a SQL query being executed. The query is as follows:

```

71 FOREIGN KEY (atm_id) references node_bank.DIM_ATM(atm_id),
72 FOREIGN KEY (date_id) references node_bank.DIM_DATE(date_id),
73 FOREIGN KEY (card_type_id) references node_bank.DIM_CARD_TYPE(card_type_id)
74 );
75
76 copy etl21.fact_atm_trans from 's3://sparkbank/bank_spark/fact_atm/part-00000-968906c2-89bf-431c-b113-2ba5e36ca14f-c000.csv'
77 iam_role 'arn:aws:iam::353260074342:role/myRedshiftRole'
78 delimiter ',' region 'us-east-1'
79 CSV;
80
81 select count(*) from etl23.fact_atm_trans ;
82 delete from etl23.fact_atm_trans ;
83
84 copy etl23.dim_atm from 's3://sparkbank/bank_spark/dim_atm/part-00000-501a1403-de03-4583-aac4-328e0b927585-c000.csv'

```

Below the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button is highlighted. Below these buttons, there's a section for 'Query results' and 'Table details'. The 'Query results' section shows the query ID '600944' and its status: 'Completed, started on October 28, 2022 at 17:15:23' with an 'ELAPSED TIME: 00 m 16 s'.

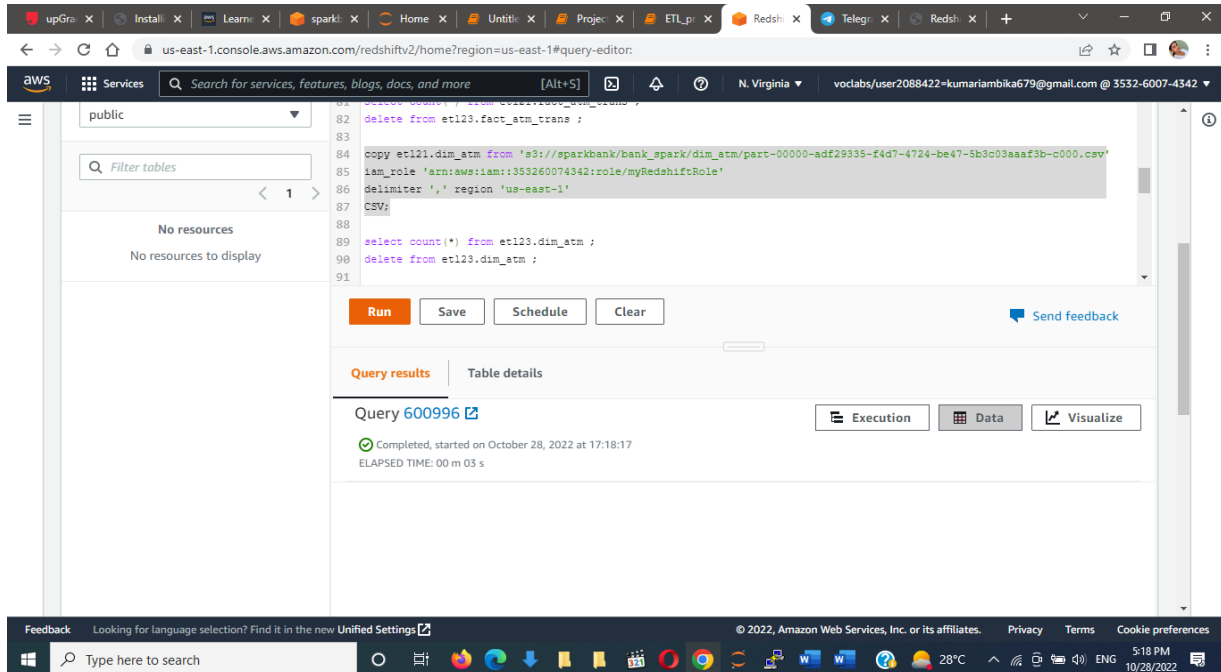
**copy etl21.dim\_atm from 's3://sparkbank/bank\_spark/dim\_atm/part-00000-adf29335-f4d7-4724-be47-5b3c03aaaf3b-c000.csv'**

**iam\_role 'arn:aws:iam::353260074342:role/myRedshiftRole'**

**delimiter ',' region 'us-east-1'**

**CSV;**





The screenshot shows the AWS Redshift Query Editor interface. The SQL query being executed is:

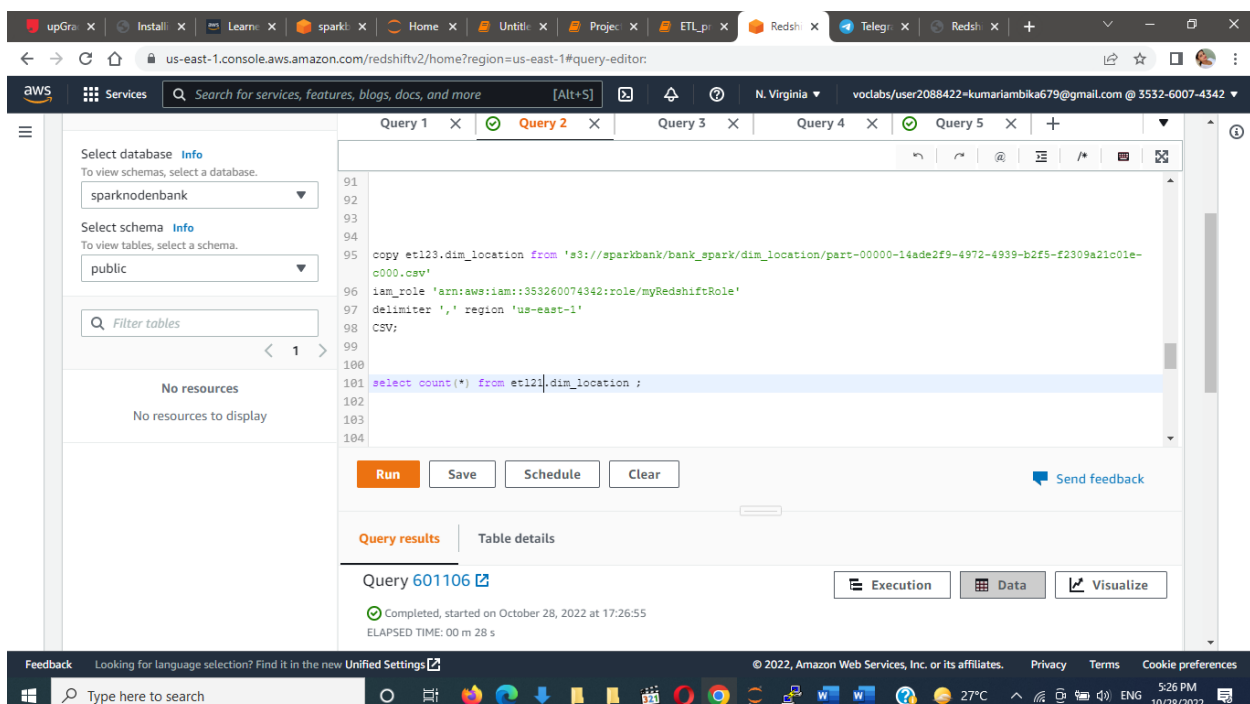
```

82 delete from etl23.fact_atm_trans ;
83
84 copy etl21.dim_atm from 's3://sparkbank/bank_spark/dim_atm/part-00000-adf29335-f4d7-4724-be47-5b3c03aaaf3b-c000.csv'
85 iam_role 'arn:aws:iam::353260074342:role/myRedshiftRole'
86 delimiter ',' region 'us-east-1'
87 CSV;
88
89 select count(*) from etl23.dim_atm ;
90 delete from etl23.dim_atm ;
91

```

The query results show that the query was completed successfully on October 28, 2022, at 17:18:17, with an elapsed time of 00 m 03 s.

copy etl23.dim\_location from 's3://sparkbank/bank\_spark/dim\_location/part-00000-14ade2f9-4972-4939-b2f5-f2309a21c01e-c000.csv'  
iam\_role 'arn:aws:iam::353260074342:role/myRedshiftRole'  
delimiter ',' region 'us-east-1'  
CSV;



The screenshot shows the AWS Redshift Query Editor interface. The SQL query being executed is:

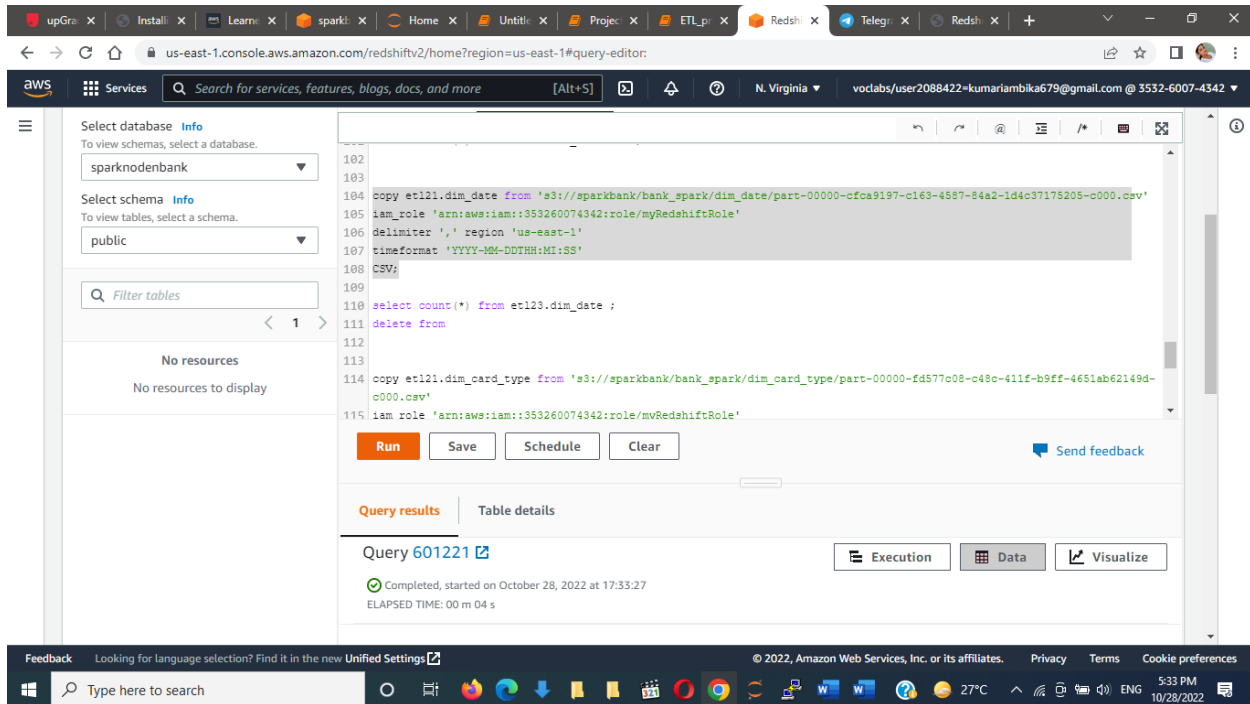
```

91
92
93
94
95 copy etl23.dim_location from 's3://sparkbank/bank_spark/dim_location/part-00000-14ade2f9-4972-4939-b2f5-f2309a21c01e-
96 c000.csv'
97 iam_role 'arn:aws:iam::353260074342:role/myRedshiftRole'
98 delimiter ',' region 'us-east-1'
99 CSV;
100
101 select count(*) from etl21.dim_location ;
102
103
104

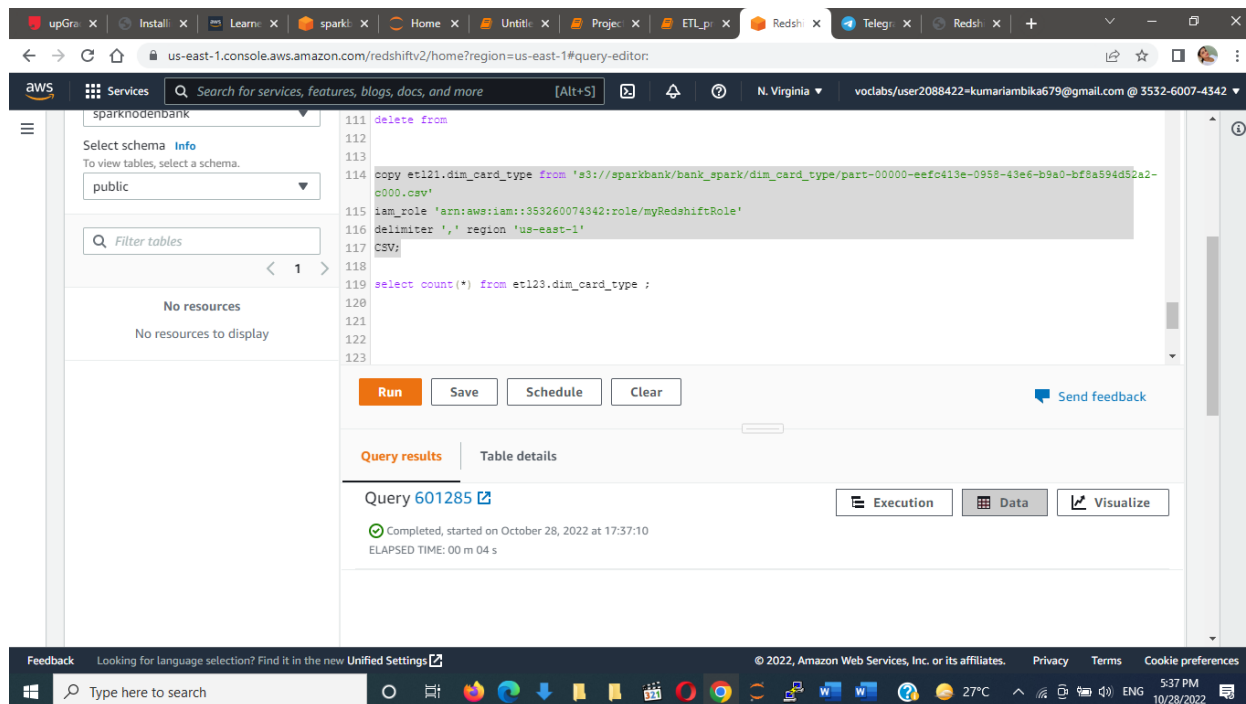
```

The query results show that the query was completed successfully on October 28, 2022, at 17:26:55, with an elapsed time of 00 m 28 s.

```
copy etl21.dim_date from 's3://sparkbank/bank_spark/dim_date/part-00000-cfca9197-
c163-4587-84a2-1d4c37175205-c000.csv'
iam_role 'arn:aws:iam::353260074342:role/myRedshiftRole'
delimiter ',' region 'us-east-1'
timeformat 'YYYY-MM-DDTHH:MI:SS'
CSV;
```



```
copy etl21.dim_card_type from 's3://sparkbank/bank_spark/dim_card_type/part-00000-
eefc413e-0958-43e6-b9a0-bf8a594d52a2-c000.csv'
iam_role 'arn:aws:iam::353260074342:role/myRedshiftRole'
delimiter ',' region 'us-east-1'
CSV;
```



The screenshot displays the AWS Redshift Query Editor interface. The left sidebar shows the 'sparknodenbank' database with a 'public' schema selected. The main area contains a SQL query:

```

111 delete from
112
113
114 copy etl21.dim_card_type from 's3://sparkbank/bank_spark/dim_card_type/part-00000-eefc413e-0958-43e6-b9a0-bf8a594d52a2-
115 c000.csv'
116 iam_role 'arn:aws:iam::353260074342:role/myRedshiftRole'
117 delimiter ',' region 'us-east-1'
118 CSV;
119
120 select count(*) from etl23.dim_card_type ;
121
122
123

```

Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button is highlighted. The 'Query results' tab is active, showing the query ID '601285' and its execution status: 'Completed, started on October 28, 2022 at 17:37:10' with an 'ELAPSED TIME: 00 m 04 s'.