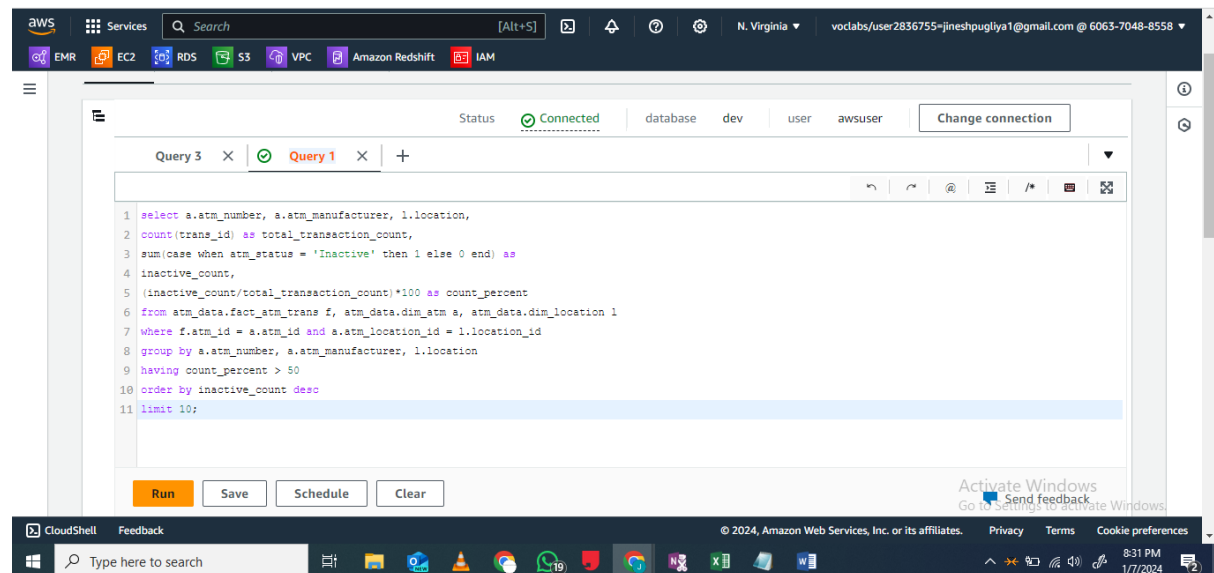


## Solving analytical queries on Redshift Cluster

Here, you have to write the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS Redshift Query editor UI.

### 1. Top 10 ATMs where most transactions are in the 'inactive' state

```
select a.atm_number, a.atm_manufacturer, l.location, count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as
inactive_count, (inactive_count/total_transaction_count)*100 as count_percent
from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l
where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
group by a.atm_number, a.atm_manufacturer, l.location
having count_percent > 50 order by inactive_count desc limit 10;
```



The screenshot shows the results of the SQL query in the AWS Redshift Query Editor. The results are displayed in a table with the following columns: atm\_number, atm\_manufacturer, location, total\_transaction\_count, inactive\_count, and count\_percent. The table contains 10 rows of data, sorted by inactive\_count in descending order.

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
16	NCR	Skive	44043	44043	100
12	NCR	ØsterØy Duus	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
88	NCR	Storcenter indg. A	32183	32183	100
30	NCR	NykØbing Mors	30883	30883	100
52	NCR	FarsØ	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
81	NCR	Spar KØbmand TornhØj	20148	20148	100
102	NCR	Aalborg Storcenter Afd	18297	18297	100

## 2. Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions

```
select f.weather_main, count(trans_id) as total_transaction_count,  
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,  
case when coalesce(inactive_count, 0) = 0 then 0.0000  
else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2)  
end as inactive_count_percent from atm_data.fact_atm_trans f  
where f.weather_main != '' group by f.weather_main  
order by inactive_count_percent desc  
limit 10;
```

The screenshot shows the AWS Redshift console interface. The top navigation bar includes the AWS logo, a search bar, and various service icons (EMR, EC2, RDS, S3, VPC, Amazon Redshift, IAM). The main content area displays a SQL query in the 'Query Editor' tab. The query is as follows:

```
1 select f.weather_main,  
2 count(trans_id) as total_transaction_count,  
3 sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,  
4 case when coalesce(inactive_count, 0) = 0 then 0.0000  
5 else trunc((cast(inactive_count as  
6 numeric(10,4))/total_transaction_count)*100, 2)  
7 end as inactive_count_percent  
8 from atm_data.fact_atm_trans f  
9 where f.weather_main != ''  
10 group by f.weather_main  
11 order by inactive_count_percent desc  
12 limit 10;
```

Below the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also present. The bottom of the console shows a Windows taskbar with the date and time (8:34 PM, 1/7/2024).

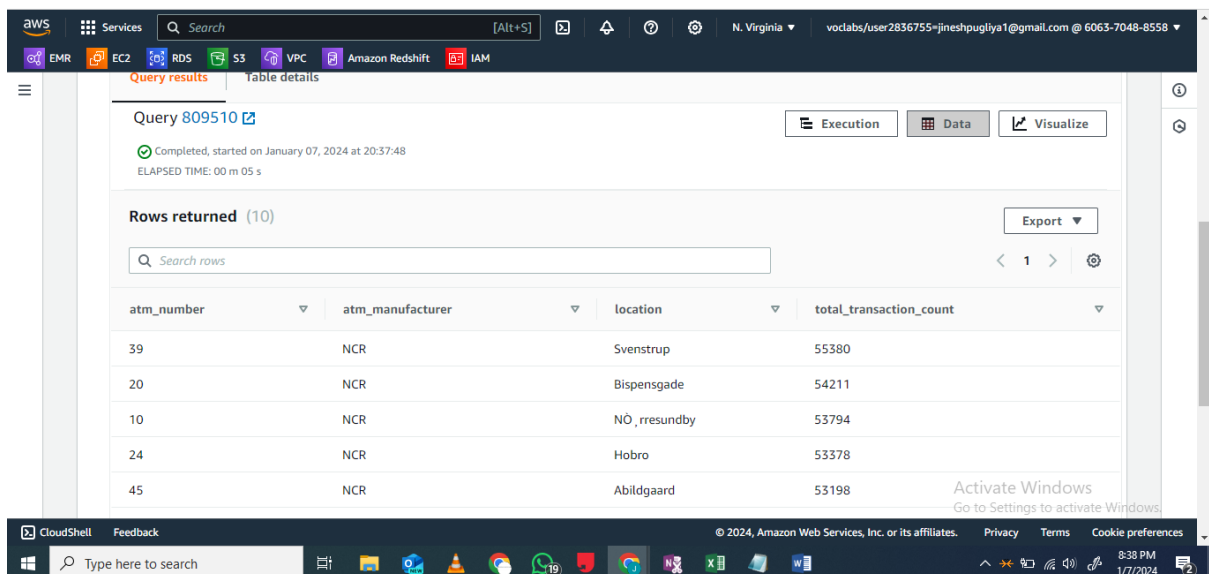
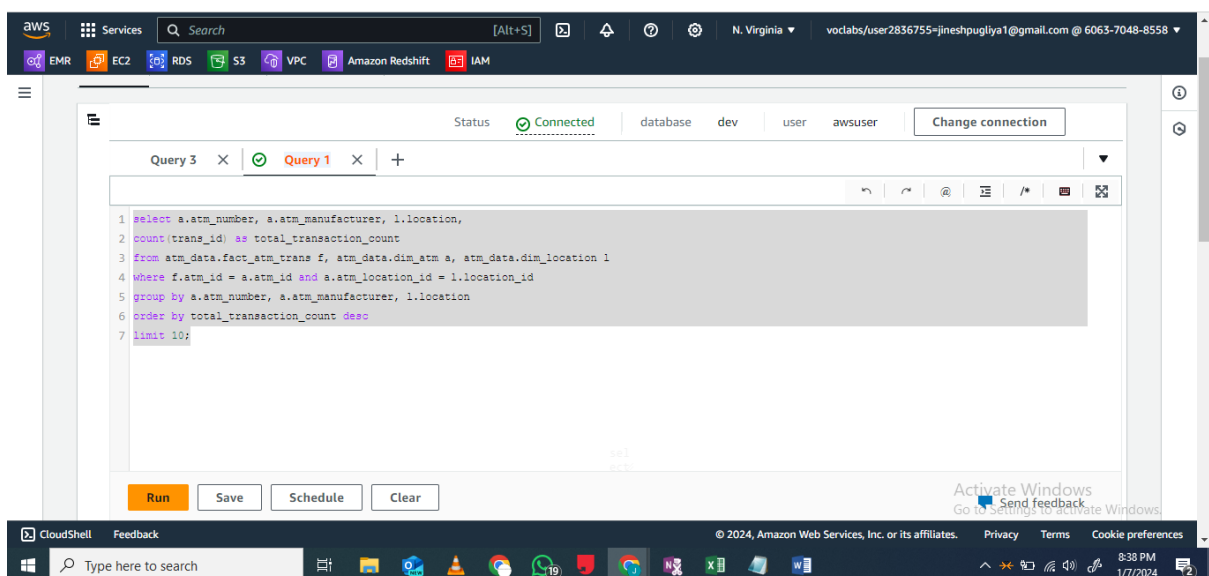
The screenshot shows the AWS Redshift console interface displaying the results of the SQL query in a table format. The table has four columns: 'weather\_main', 'total\_transaction\_count', 'inactive\_count', and 'inactive\_count\_percent'. The results are as follows:

weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.5600
Fog	18174	3729	20.5100
Clouds	1181901	194027	16.4100
Rain	545135	86017	15.7700
Clear	543949	85531	15.7200
Mist	82801	12864	15.5300
Thunderstorm	2549	361	14.1600
Drizzle	62530	8670	13.8600
TORNADO	38	1	2.6300
Haze	3	0	0.0000

The bottom of the console shows a Windows taskbar with the date and time (8:35 PM, 1/7/2024).

### 3. Top 10 ATMs with the most number of transactions throughout the year

```
select a.atm_number, a.atm_manufacturer, l.location,  
count(trans_id) as total_transaction_count  
from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l  
where f.atm_id = a.atm_id and a.atm_location_id = l.location_id  
group by a.atm_number, a.atm_manufacturer, l.location  
order by total_transaction_count desc  
limit 10;
```



#### 4. Number of overall ATM transactions going inactive per month for each month

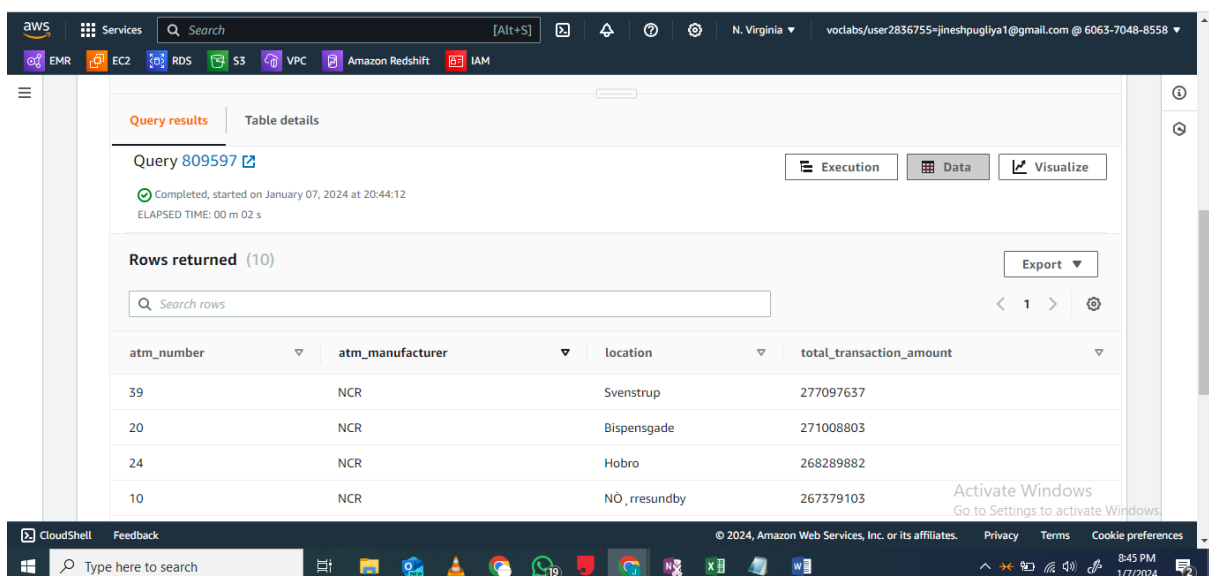
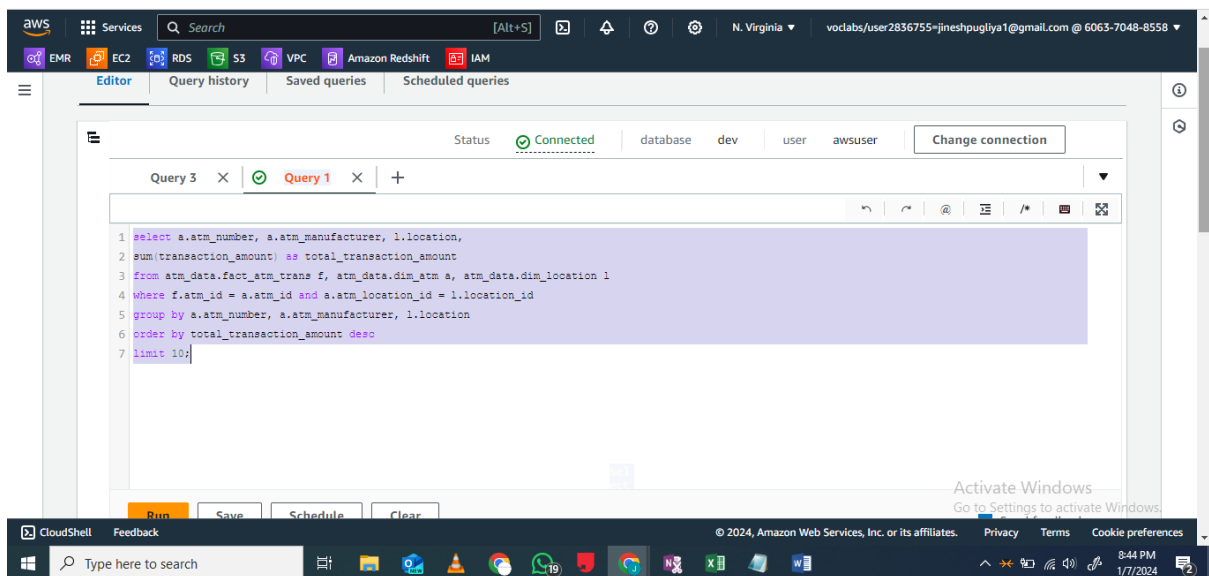
```
select d.year, d.month, count(trans_id) as total_transaction_count,  
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,  
case when coalesce(inactive_count, 0) = 0 then 0.0000  
else trunc((cast(inactive_count as  
numeric(10,4))/total_transaction_count)*100, 2)  
end as inactive_count_percent  
from atm_data.fact_atm_trans f inner join atm_data.dim_date d on f.date_id =  
d.date_id group by d.year, d.month order by d.year, d.month
```

```
1 select d.year, d.month,  
2 count(trans_id) as total_transaction_count,  
3 sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,  
4 case when coalesce(inactive_count, 0) = 0 then 0.0000  
5 else trunc((cast(inactive_count as  
6 numeric(10,4))/total_transaction_count)*100, 2)  
7 end as inactive_count_percent  
8 from atm_data.fact_atm_trans f inner join atm_data.dim_date d on f.date_id =  
9 d.date_id  
10 group by d.year, d.month  
11 order by d.year, d.month
```

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.1100
2017	August	217218	36713	16.9000
2017	December	197048	20476	10.3900
2017	February	182659	36656	20.0600
2017	January	180195	35953	19.9500
2017	July	227682	38139	16.7500
2017	June	225166	36789	16.3300
2017	March	209586	41046	19.5800
2017	May	222418	37679	16.9400

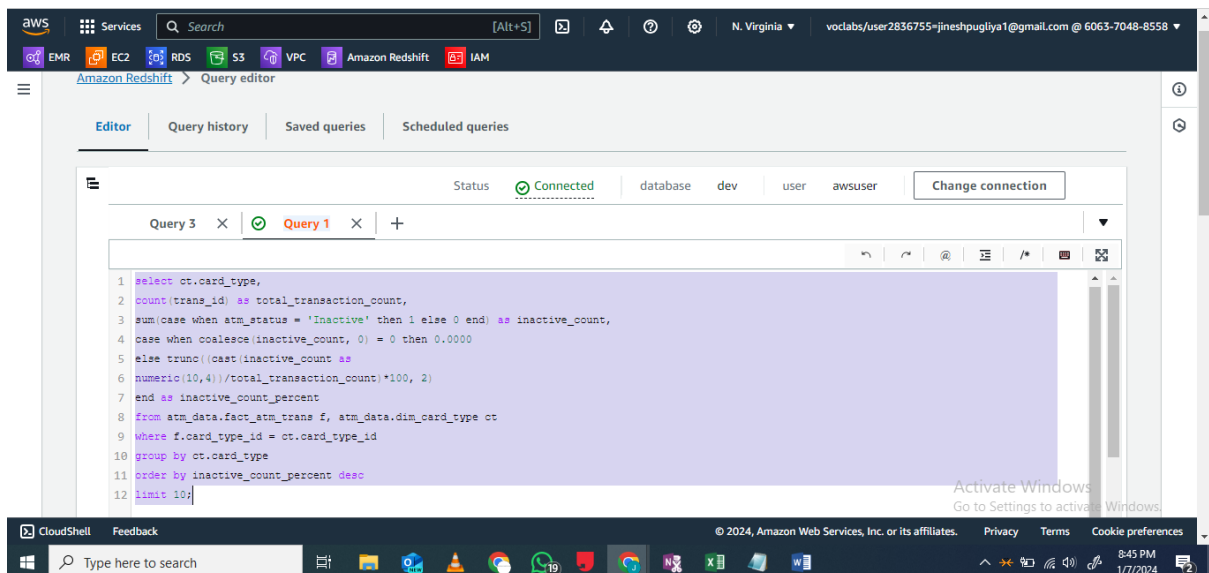
## 5. Top 10 ATMs with the highest total withdrawn amount throughout the year

```
select a.atm_number, a.atm_manufacturer, l.location,  
sum(transaction_amount) as total_transaction_amount  
from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l  
where f.atm_id = a.atm_id and a.atm_location_id = l.location_id  
group by a.atm_number, a.atm_manufacturer, l.location  
order by total_transaction_amount desc  
limit 10;
```



## 6. Number of failed ATM transactions across various card types

```
select ct.card_type, count(trans_id) as total_transaction_count,  
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,  
case when coalesce(inactive_count, 0) = 0 then 0.0000  
else trunc((cast(inactive_count as  
numeric(10,4))/total_transaction_count)*100, 2)  
end as inactive_count_percent  
from atm_data.fact_atm_trans f, atm_data.dim_card_type ct  
where f.card_type_id = ct.card_type_id  
group by ct.card_type  
order by inactive_count_percent desc limit 10;
```

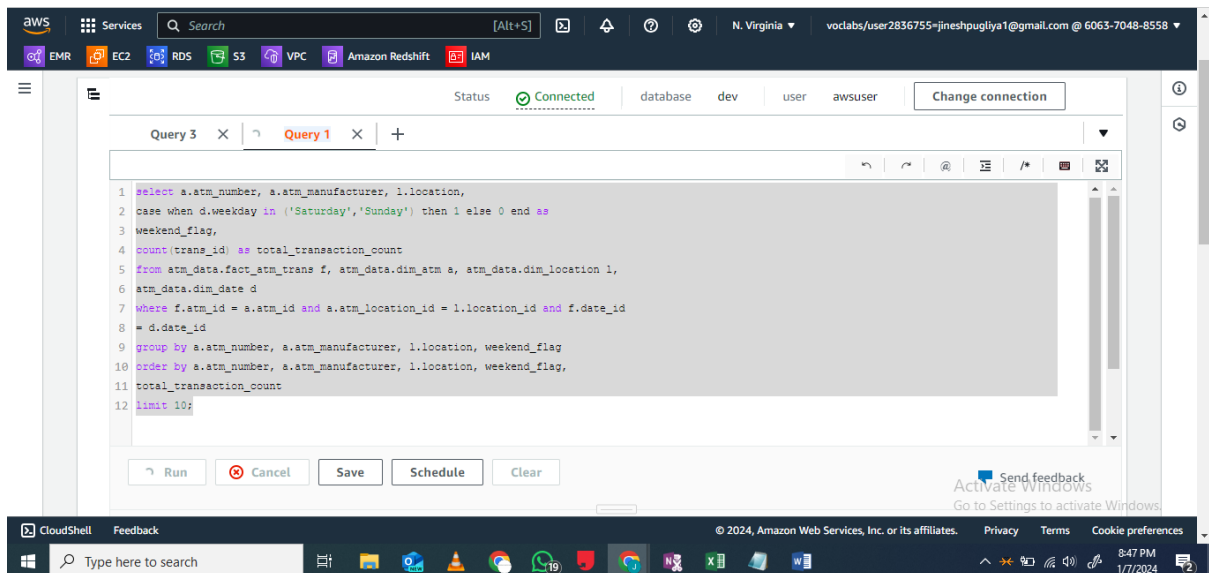


The screenshot shows the Amazon Redshift Query Results interface. The query is identified as Query 809621. The results are displayed in a table with 4 columns: card\_type, total\_transaction\_count, inactive\_count, and inactive\_count\_percent. The results are sorted by inactive\_count\_percent in descending order.

card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.7600
VISA	170828	30713	17.9700
Dankort - on-us	143813	24680	17.1600
CIRRUS	17362	2953	17.0000
HOJvekort - on-us	62487	10331	16.5300

**7. Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order this by the ATM\_number, ATM\_manufacturer, location, weekend\_flag and then total\_transaction\_count**

```
select a.atm_number, a.atm_manufacturer, l.location,  
case when d.weekday in ('Saturday','Sunday') then 1 else 0 end as  
weekend_flag, count(trans_id) as total_transaction_count  
from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l,  
atm_data.dim_date d  
where f.atm_id = a.atm_id and a.atm_location_id = l.location_id and f.date_id = d.date_id  
group by a.atm_number, a.atm_manufacturer, l.location, weekend_flag  
order by a.atm_number, a.atm_manufacturer, l.location, weekend_flag,  
total_transaction_count limit 10;
```

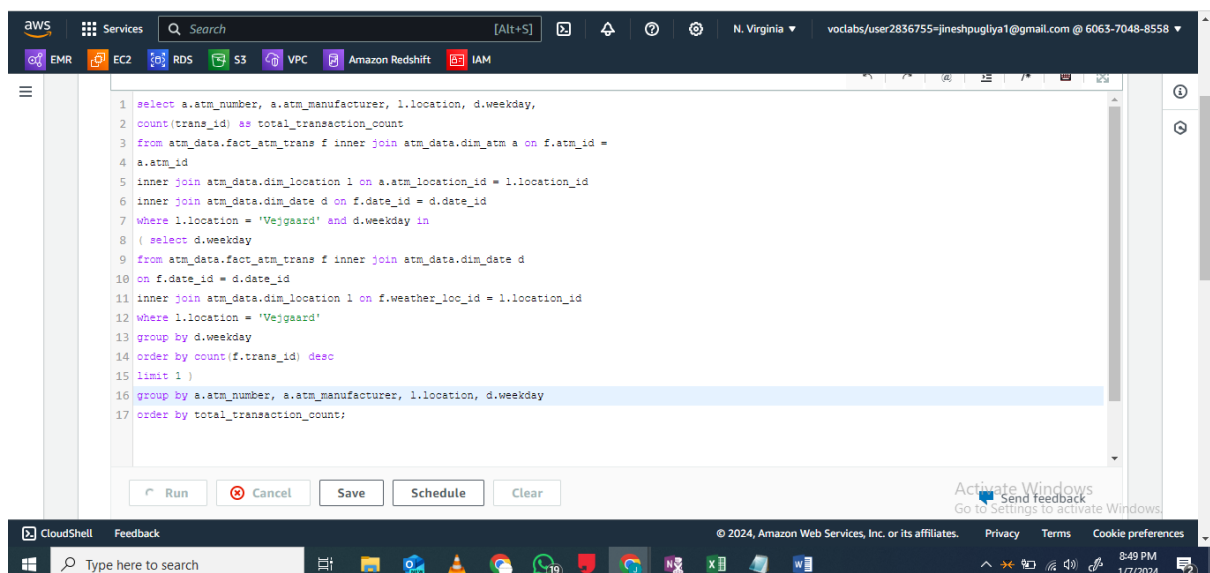


The screenshot shows the AWS Redshift console interface displaying the results of Query 809638. The query is completed, started on January 07, 2024 at 20:47:45, with an elapsed time of 00 m 13 s. The results are shown in a table with 10 rows returned.

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NO stved	0	32711
1	NCR	NO stved	1	10076
10	NCR	NO_rresundby	0	41667
10	NCR	NO_rresundby	1	12127
100	NCR	Intern Skive	0	17812

## 8. Most active day in each ATMs from location "Vejgaard"

```
select a.atm_number, a.atm_manufacturer, l.location, d.weekday,  
count(trans_id) as total_transaction_count  
from atm_data.fact_atm_trans f inner join atm_data.dim_atm a on f.atm_id =  
a.atm_id  
inner join atm_data.dim_location l on a.atm_location_id = l.location_id  
inner join atm_data.dim_date d on f.date_id = d.date_id  
where l.location = 'Vejgaard' and d.weekday in  
( select d.weekday  
from atm_data.fact_atm_trans f inner join atm_data.dim_date d  
on f.date_id = d.date_id  
inner join atm_data.dim_location l on f.location_id = l.location_id  
where l.location = 'Vejgaard'  
group by d.weekday  
order by count(f.trans_id) desc  
limit 1 )  
group by a.atm_number, a.atm_manufacturer, l.location, d.weekday  
order by total_transaction_count;
```





aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user2836755=jineshpugliya1@gmail.com @ 6063-7048-8558

EMR

EC2

RDS

S3

VPC

Amazon Redshift

IAM

Completed, started on January 07, 2024 at 20:49:51

ELAPSED TIME: 00 m 02 s

Rows returned (2)

Export

Search rows

< 1 >

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejgaard	Friday	9514
2	NCR	Vejgaard	Friday	12580

Activate Windows  
Go to Settings to activate Windows.

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Type here to search

8:50 PM 1/7/2024