

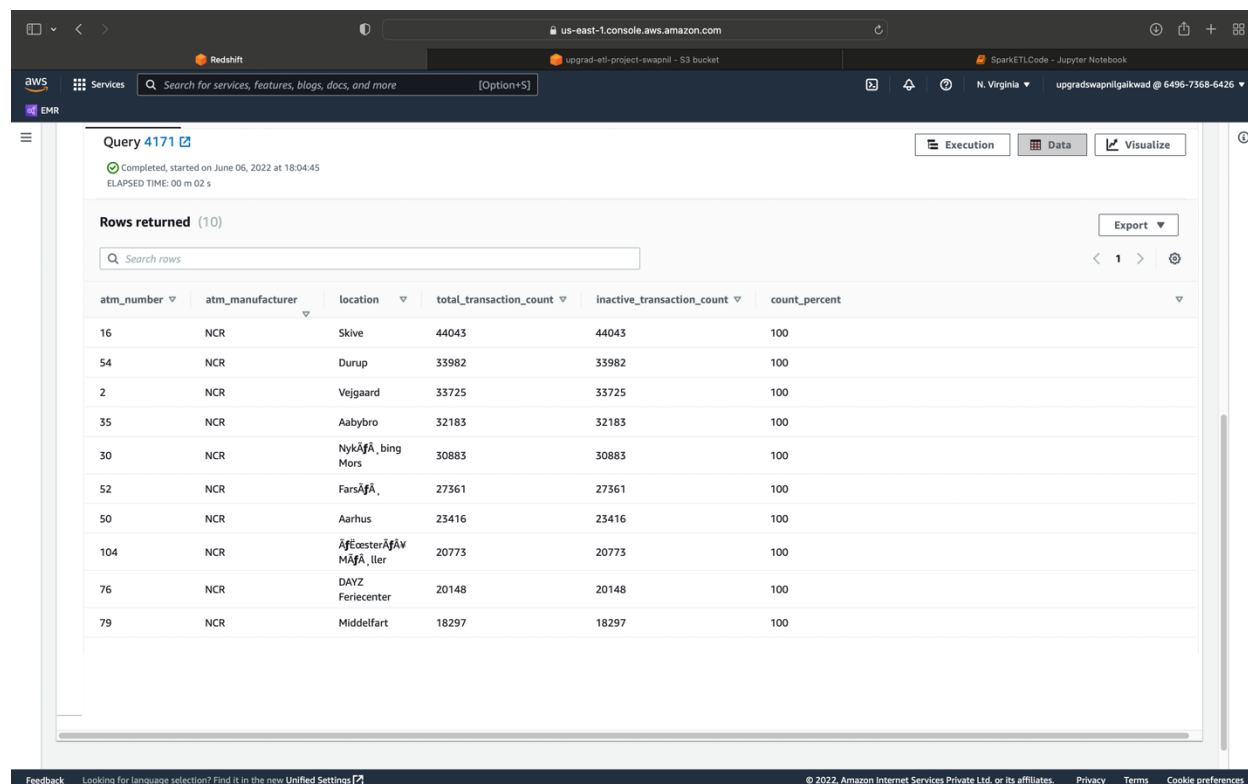
Solving analytical queries on Redshift Cluster

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_transaction_count ,
(inactive_transaction_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 incative_transaction_count desc
limit 10;
```

<Screenshot of the resultant table>

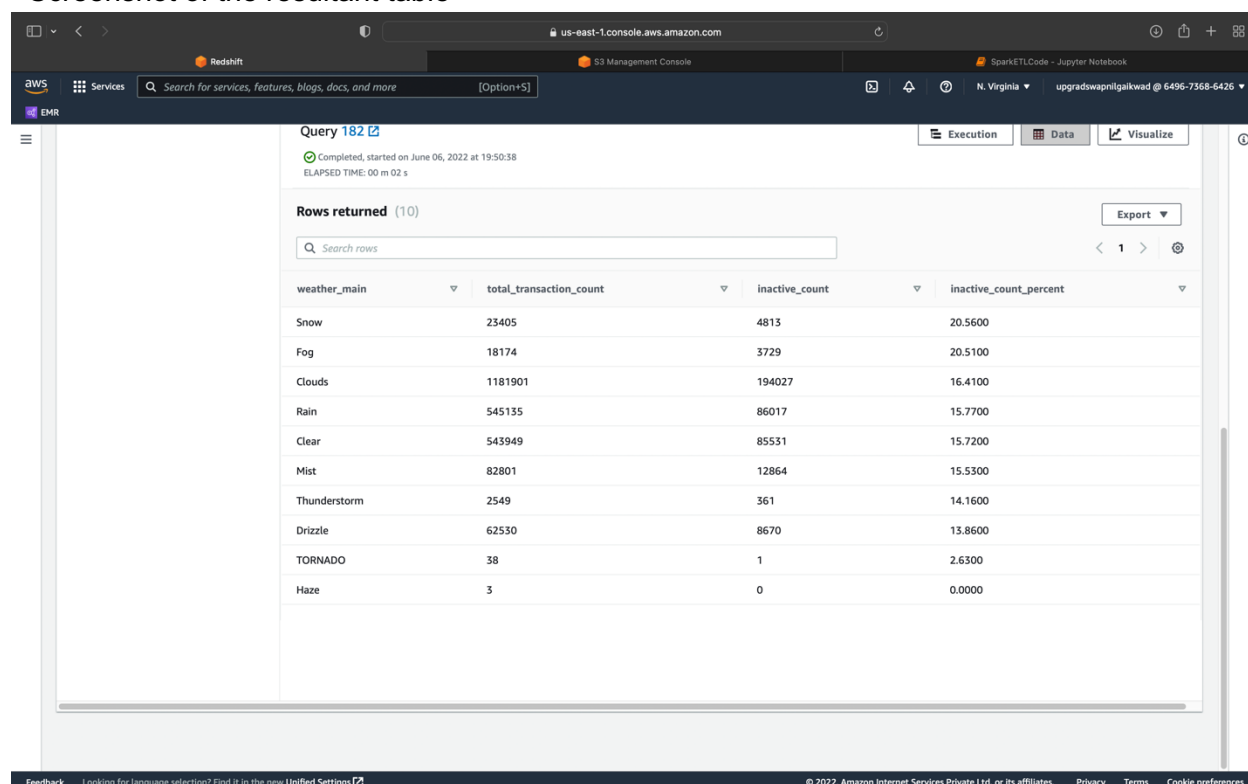


atm_number	atm_manufacturer	location	total_transaction_count	inactive_transaction_count	count_percent
16	NCR	Skive	44043	44043	100
54	NCR	Durup	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
35	NCR	Aabybro	32183	32183	100
30	NCR	Nykøbing Mors	30883	30883	100
52	NCR	Farsø	27361	27361	100
50	NCR	Aarhus	23416	23416	100
104	NCR	Århus	20773	20773	100
76	NCR	DAYZ Feriecenter	20148	20148	100
79	NCR	Middelfart	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;
```

<Screenshot of the resultant table>



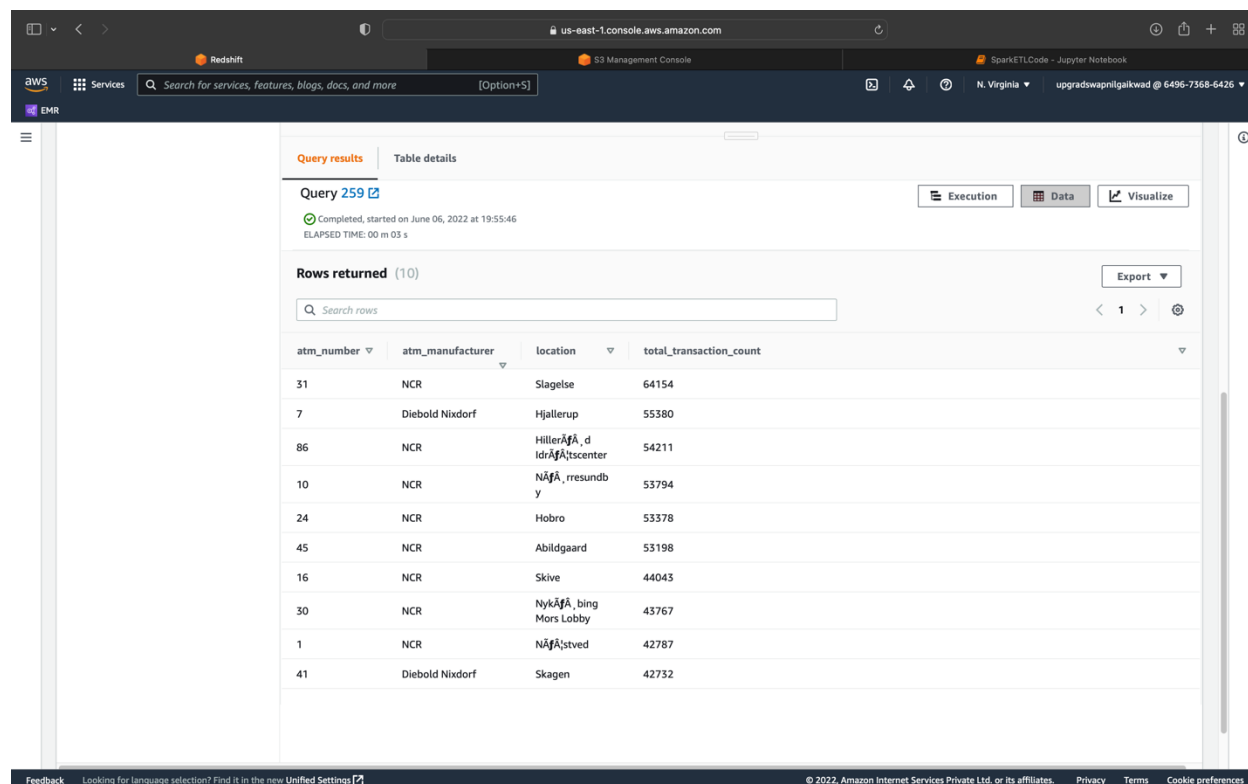
The screenshot shows the AWS Redshift console interface. At the top, there's a navigation bar with 'us-east-1 console.aws.amazon.com' and 'Redshift' selected. Below the navigation bar, there's a search bar and a 'Query 182' link. The main content area displays the query results for 'Query 182'. The query is completed, started on June 06, 2022 at 19:50:38, and elapsed time is 00 m 02 s. The results are shown in a table with 4 columns: 'weather_main', 'total_transaction_count', 'inactive_count', and 'inactive_count_percent'. The table contains 10 rows of data, sorted by 'inactive_count_percent' in descending order. The 'weather_main' column lists various weather conditions: Snow, Fog, Clouds, Rain, Clear, Mist, Thunderstorm, Drizzle, TORNADO, and Haze. The 'total_transaction_count' column shows the number of transactions for each weather condition. The 'inactive_count' column shows the number of ATM failures for each weather condition. The 'inactive_count_percent' column shows the percentage of ATM failures for each weather condition.

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

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.locoation
order by total_transaction_count desc
limit 10;
```

<Screenshot of the resultant table>



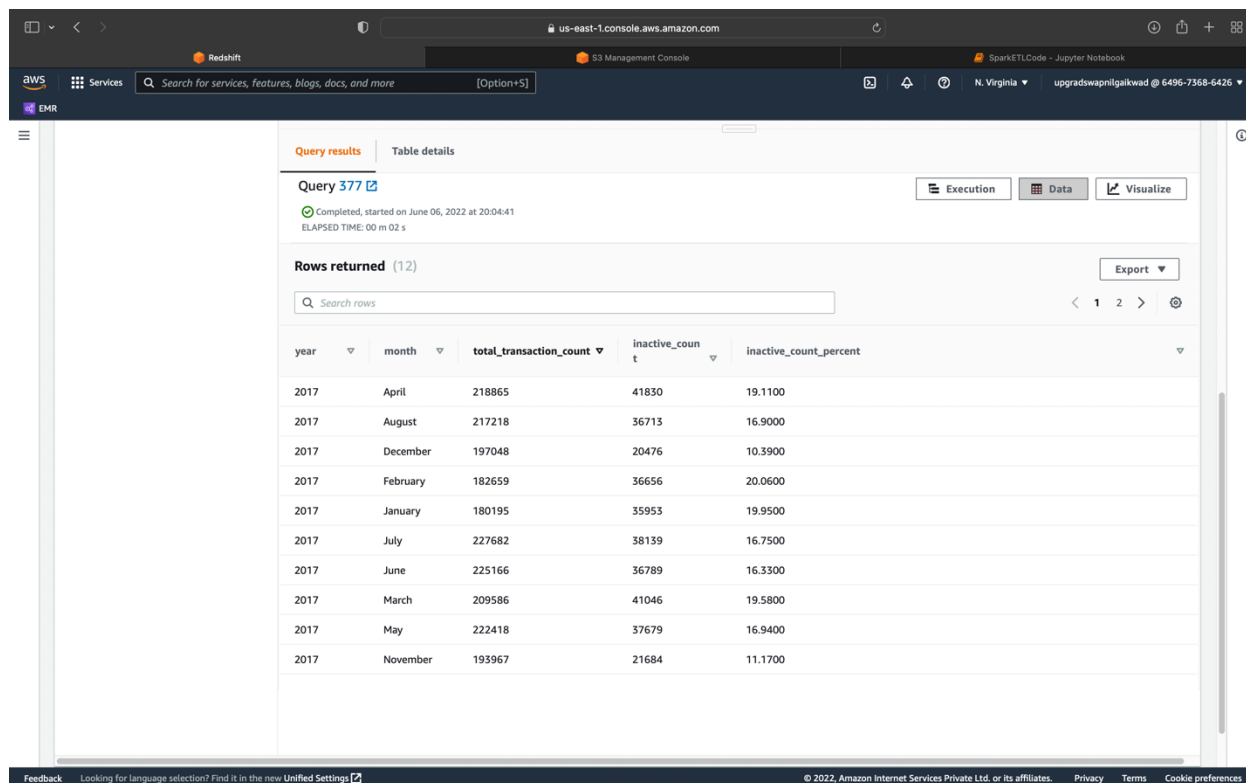
The screenshot shows the AWS Redshift console interface. The top navigation bar includes the AWS logo, a search bar, and the user's profile. The main content area displays the results of a SQL query (Query 259) executed on June 06, 2022, at 19:55:46. The query returned 10 rows, sorted by total transaction count in descending order. The table has four columns: atm_number, atm_manufacturer, location, and total_transaction_count. The data is as follows:

atm_number	atm_manufacturer	location	total_transaction_count
31	NCR	Slagelse	64154
7	Diebold Nixdorf	Hjallerup	55380
86	NCR	Hillerød, d Idrættscenter	54211
10	NCR	Næstved, resundb y	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
30	NCR	Nykøbing, bing Mors Lobby	43767
1	NCR	Næstved	42787
41	Diebold Nixdorf	Skagen	42732

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
eles 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
```

<Screenshot of the resultant table>



Query 377

Completed, started on June 06, 2022 at 20:04:41
ELAPSED TIME: 00 m 02 s

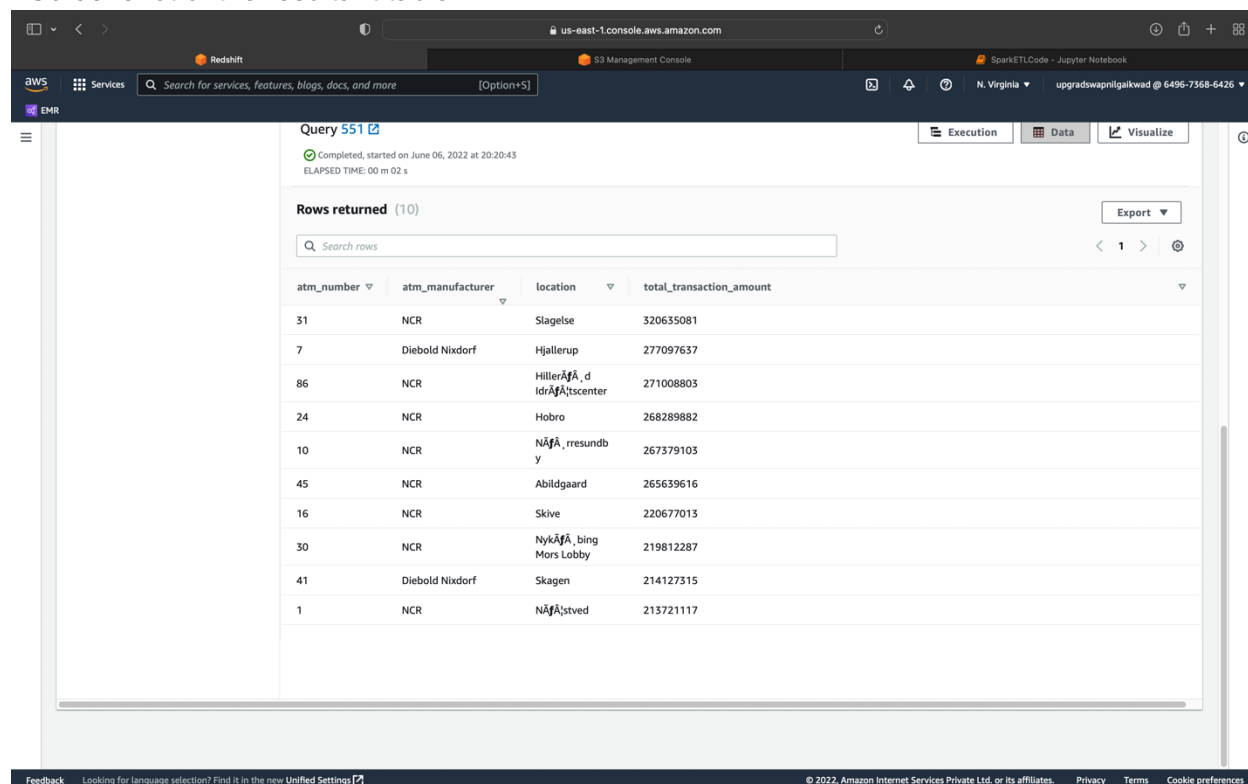
Rows returned (12)

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
2017	November	193967	21684	11.1700

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 ;
```

<Screenshot of the resultant table>



Query 551

Completed, started on June 06, 2022 at 20:20:43
ELAPSED TIME: 00 m 02 s

Rows returned (10)

Search rows

atm_number	atm_manufacturer	location	total_transaction_amount
31	NCR	Slagelse	320635081
7	Diebold Nixdorf	Hjallerup	277097637
86	NCR	Hillerød, d Idrættscenter	271008803
24	NCR	Hobro	268289882
10	NCR	Næstved, resundb y	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
30	NCR	Nykøbing, bing Mors Lobby	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	Næstved	213721117

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

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 0.000
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 ;
```

<Screenshot of the resultant table>

us-east-1.console.aws.amazon.com

Redshift

S3 Management Console

SparkETLCode - Jupyter Notebook

Services

Search for services, features, blogs, docs, and more

[Option+S]

N. Virginia

upgradswapnilgaikwad @ 6496-7368-6426

EMR

Query 653

Completed, started on June 06, 2022 at 20:30:15

ELAPSED TIME: 00 m 02 s

Execution

Data

Visualize

Rows returned (10)

Export

Search rows

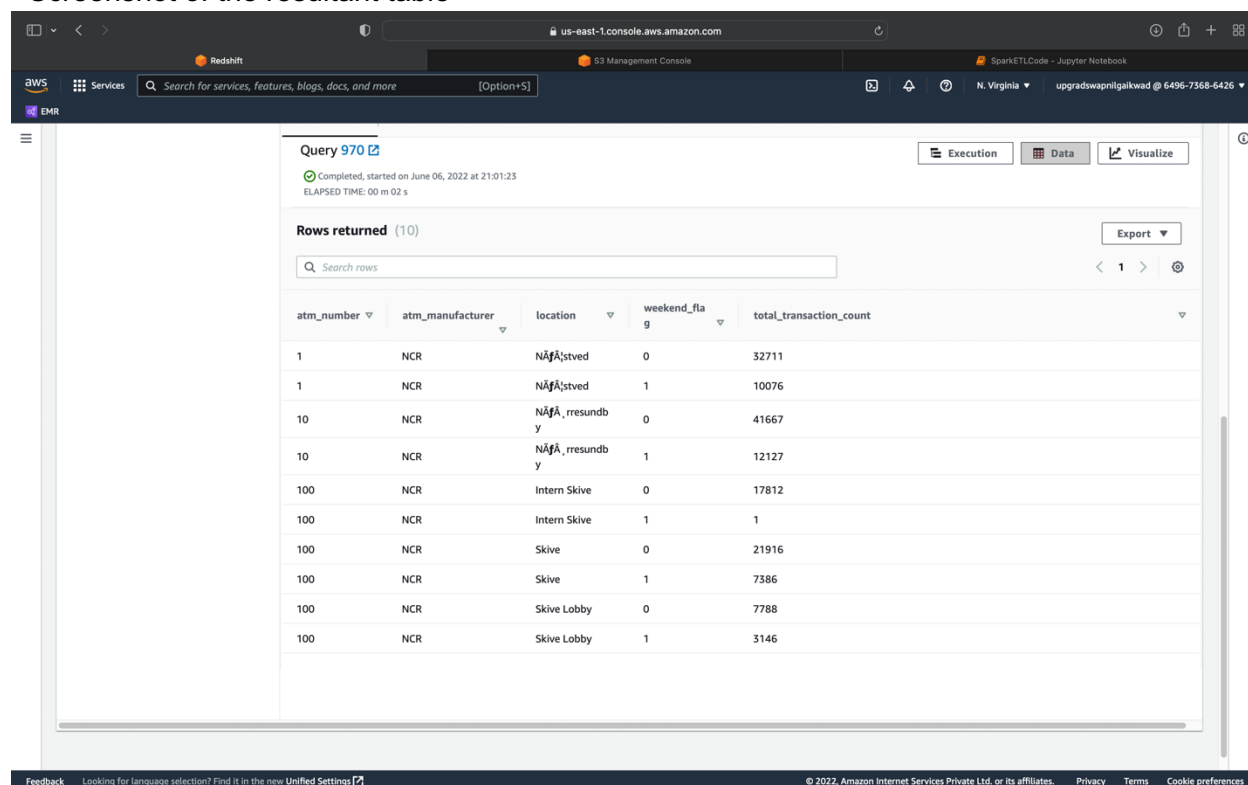
< 1 >

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
Hankort - on-us	62487	10331	16.5300
Dankort	28581	4557	15.9400
MasterCard	400507	63482	15.8500
Visa Dankort - on-us	748805	112972	15.0800
Hankort	8459	1208	14.2800
Visa Dankort	427840	60547	14.1500

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 ;
```

<Screenshot of the resultant table>



Query 970 [View](#)

Completed, started on June 06, 2022 at 21:01:23
ELAPSED TIME: 00 m 02 s

Execution Data Visualize

Rows returned (10)

Search rows

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NCR, resundb	0	32711
1	NCR	NCR, resundb	1	10076
10	NCR	NCR, resundb	0	41667
10	NCR	NCR, resundb	1	12127
100	NCR	Intern Skive	0	17812
100	NCR	Intern Skive	1	1
100	NCR	Skive	0	21916
100	NCR	Skive	1	7386
100	NCR	Skive Lobby	0	7788
100	NCR	Skive Lobby	1	3146

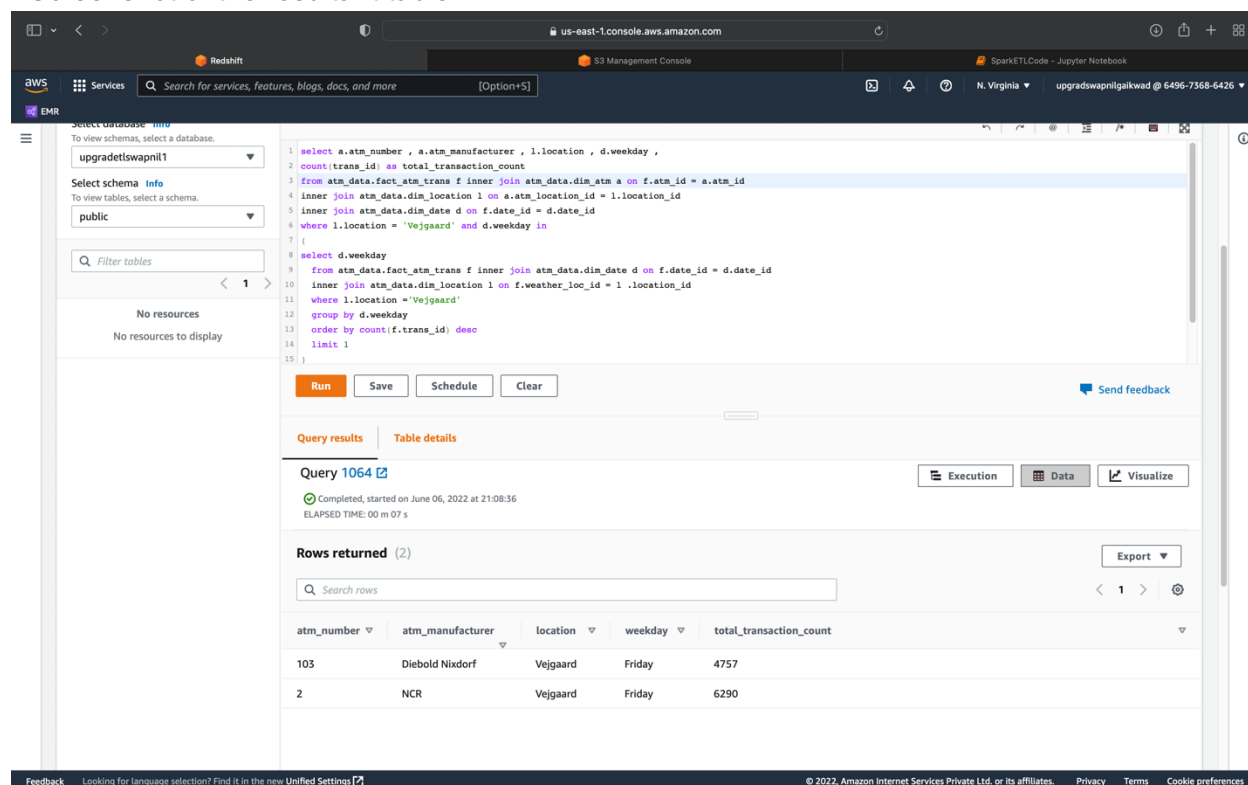
Feedback Looking for language selection? Find it in the new Unified Settings [View](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

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 weather_loc_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 ;
```

<Screenshot of the resultant table>



The screenshot shows the AWS Redshift console interface. On the left, there's a sidebar with navigation options like 'EMR', 'Redshift', and 'S3 Management Console'. The main area displays a SQL query in a text editor. The query is designed to find the most active day in each ATM from the location 'Vejgaard'. It uses a subquery to identify the most active day (Friday) and then joins this with the fact table to get the total transaction count for each ATM on that day.

Query 1064 (Completed, started on June 06, 2022 at 21:08:36, ELAPSED TIME: 00 m 07 s)

Rows returned (2)

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejgaard	Friday	4757
2	NCR	Vejgaard	Friday	6290