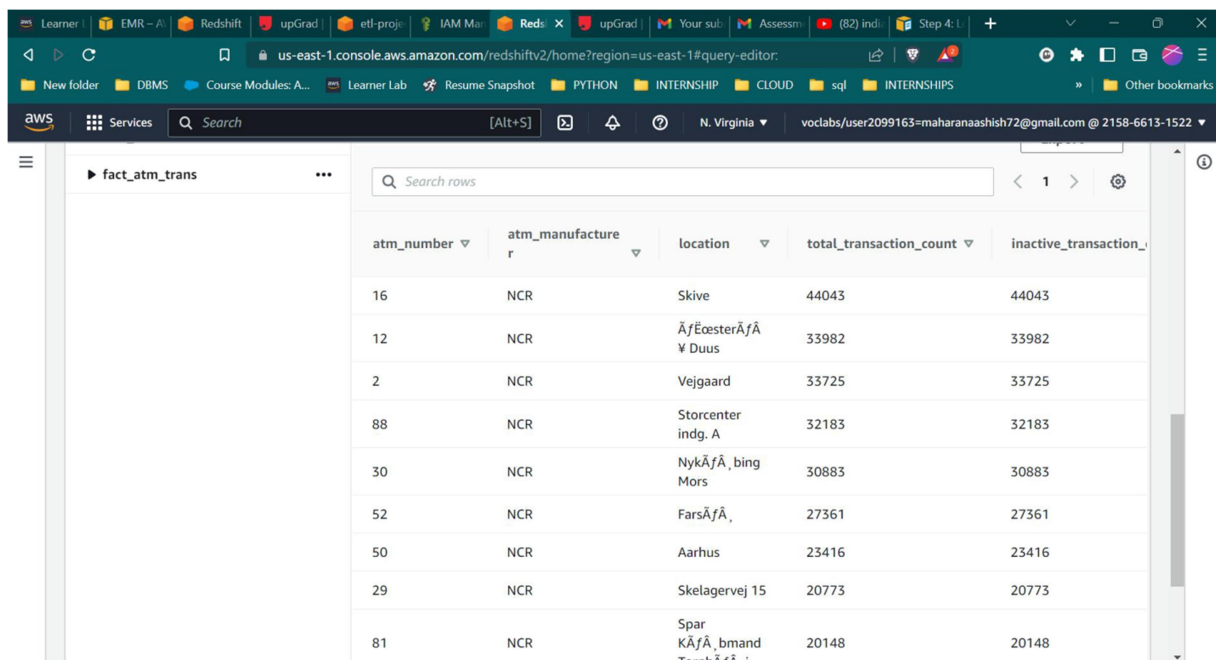


Solving analytical queries on Redshift Cluster

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

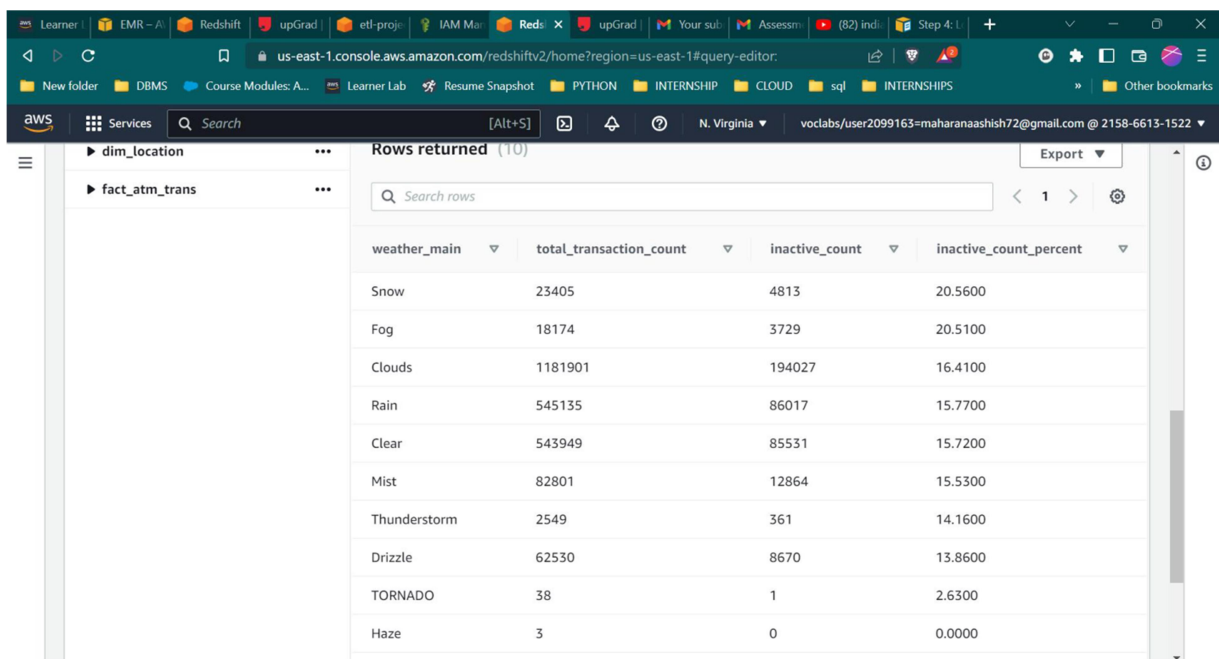
```
select a.atm_number, a.atm_manufacturer, l.location,
count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as
inactive_transaction_count,
(inactive_transaction_count/total_transaction_count)*100 as count_percent
from atm_data.fact_atm_trans f, atm_data.dim_atm 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 inactive_transaction_count desc
limit 10;
```



atm_number	atm_manufacturer	location	total_transaction_count	inactive_transaction_count
16	NCR	Skive	44043	44043
12	NCR	Åfjærøster Åfjær ¥ Duus	33982	33982
2	NCR	Vejgaard	33725	33725
88	NCR	Storcenter indg. A	32183	32183
30	NCR	Nyk Åfjær, bing Mors	30883	30883
52	NCR	Fars Åfjær,	27361	27361
50	NCR	Aarhus	23416	23416
29	NCR	Skelagervej 15	20773	20773
81	NCR	Spar K Åfjær bmand Tørnh Åfjær i	20148	20148

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;
```

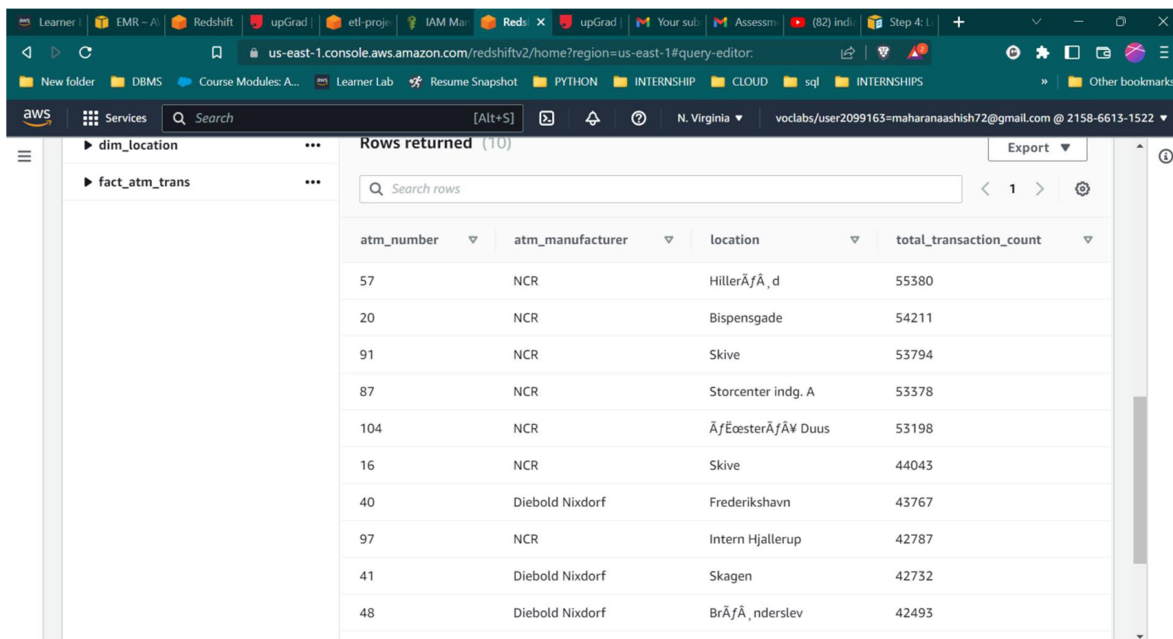


The screenshot shows the AWS Redshift console interface. The query results are displayed in a table with the following columns: weather_main, total_transaction_count, inactive_count, and inactive_count_percent. The results are ordered by inactive_count_percent in descending order.

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

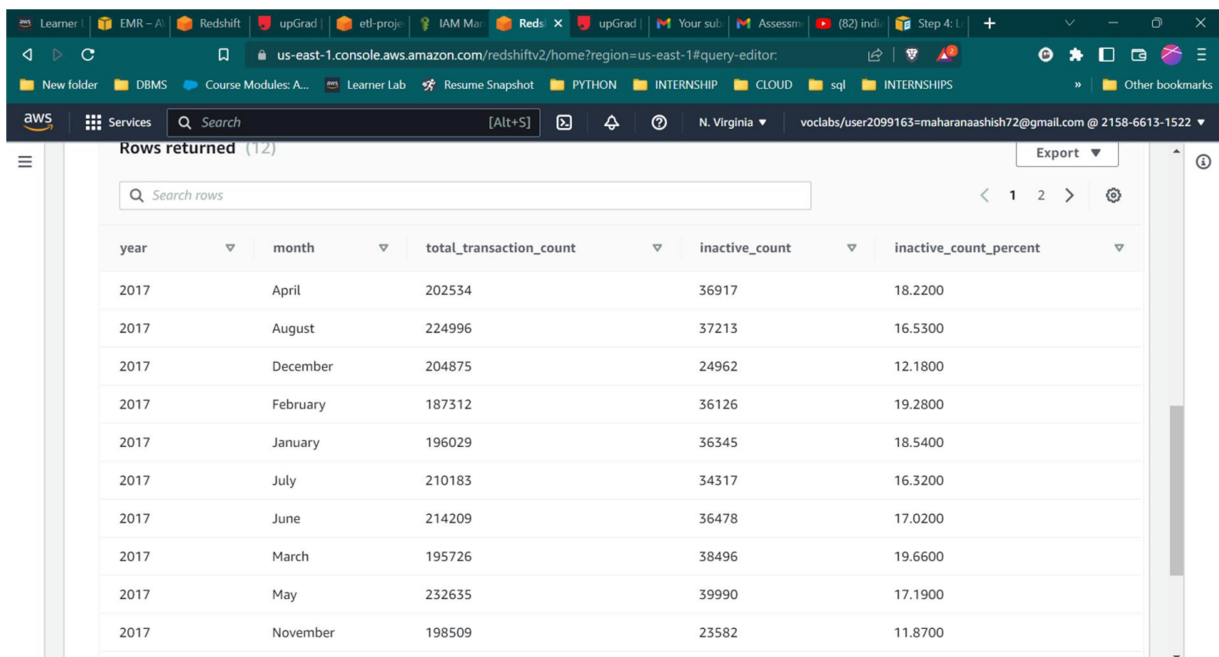


Rows returned (10)

atm_number	atm_manufacturer	location	total_transaction_count
57	NCR	Hillerfjeld	55380
20	NCR	Bispensgade	54211
91	NCR	Skive	53794
87	NCR	Storcenter indg. A	53378
104	NCR	Åfjellerød Duus	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
97	NCR	Intern Hjallerup	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	Brfjellerød, nderslev	42493

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;
```

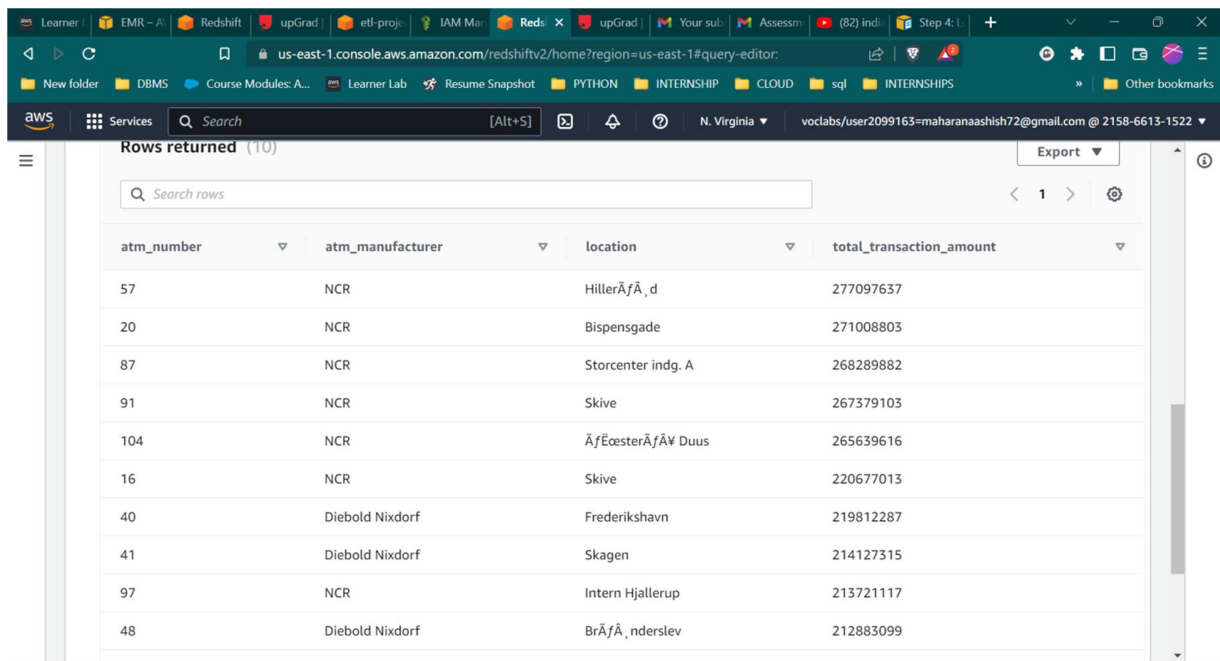


Rows returned (12)

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	202534	36917	18.2200
2017	August	224996	37213	16.5300
2017	December	204875	24962	12.1800
2017	February	187312	36126	19.2800
2017	January	196029	36345	18.5400
2017	July	210183	34317	16.3200
2017	June	214209	36478	17.0200
2017	March	195726	38496	19.6600
2017	May	232635	39990	17.1900
2017	November	198509	23582	11.8700

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;
```

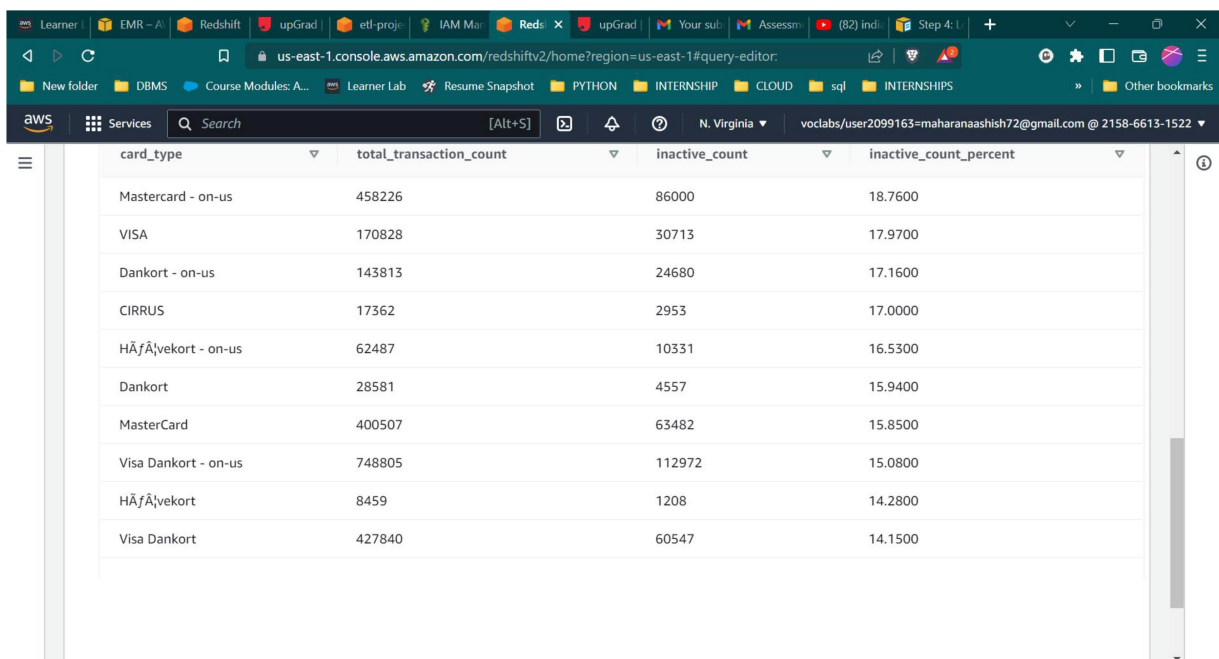


The screenshot shows the AWS Redshift console interface. The top navigation bar includes the AWS logo, a search bar, and various service links. The main content area displays the results of a SQL query, showing 10 rows of data. The columns are: atm_number, atm_manufacturer, location, and total_transaction_amount. The data is sorted in descending order of total_transaction_amount.

atm_number	atm_manufacturer	location	total_transaction_amount
57	NCR	HillerÃfÃ, d	277097637
20	NCR	Bispensgade	271008803
87	NCR	Storcenter indg. A	268289882
91	NCR	Skive	267379103
104	NCR	ÃfÃeesterÃfÃ Duus	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
97	NCR	Intern Hjallerup	213721117
48	Diebold Nixdorf	BrÃfÃ, nderslev	212883099

6. Number of failed ATM transactions across various card types

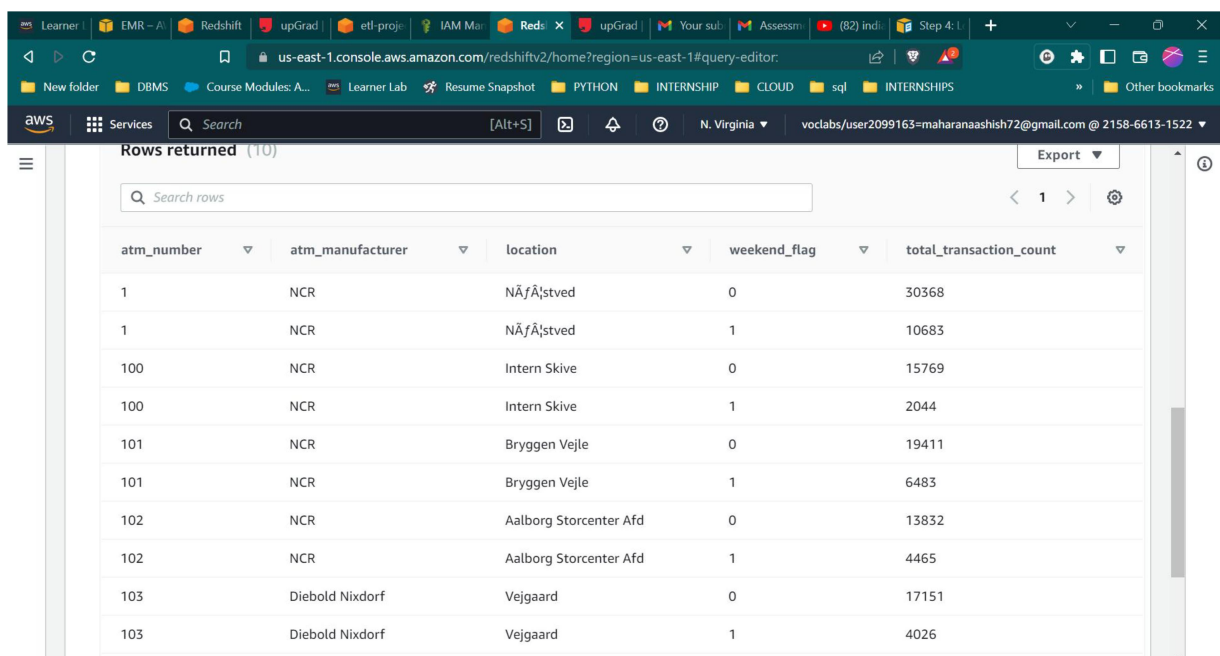
```
select ct.card_type,
count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as
inactive_count,
case when coalesce(inactive_count, 0) = 0 then 0.0000
else trunc((cast(inactive_count as
numeric(10,4))/total_transaction_count)*100, 2)
end as inactive_count_percent
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;
```



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
HÃfÃ!vekort - on-us	62487	10331	16.5300
Dankort	28581	4557	15.9400
MasterCard	400507	63482	15.8500
Visa Dankort - on-us	748805	112972	15.0800
HÃfÃ!vekort	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;
```

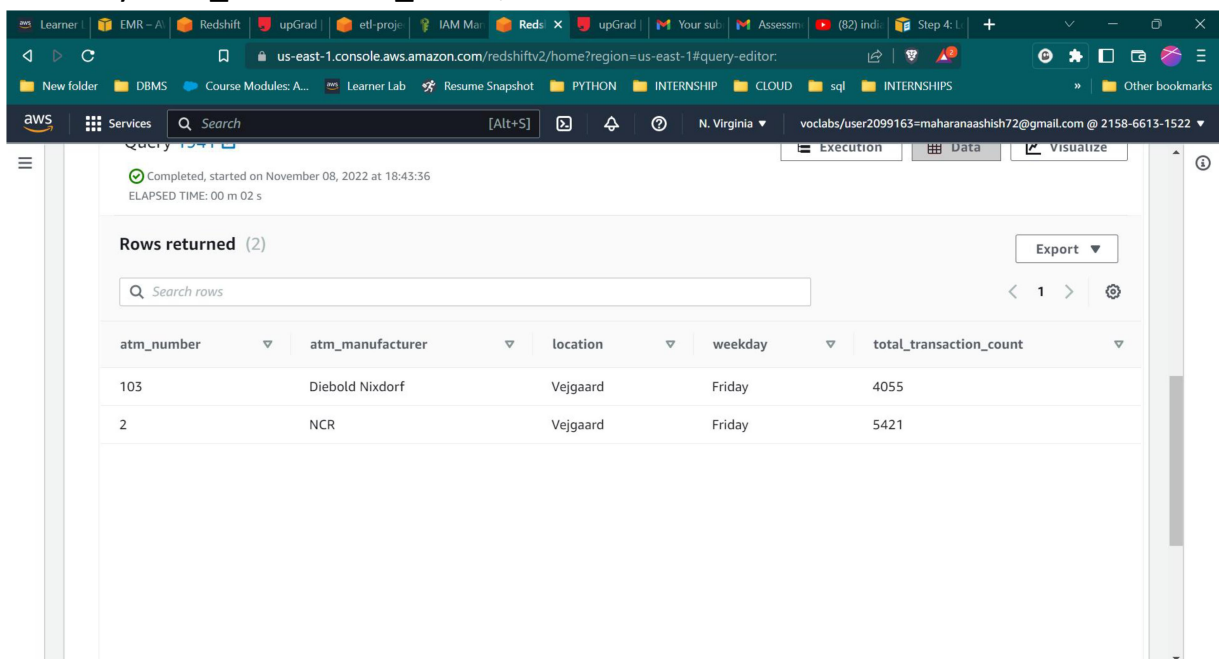


The screenshot shows the AWS Redshift console interface. The top navigation bar includes various services like EMR, Redshift, IAM, and others. The main content area displays the results of a SQL query. The table has 5 columns: atm_number, atm_manufacturer, location, weekend_flag, and total_transaction_count. The results are ordered by atm_number, atm_manufacturer, location, weekend_flag, and total_transaction_count. The first 10 rows are shown, with a search bar and pagination controls at the top of the table.

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÄfÄ'stved	0	30368
1	NCR	NÄfÄ'stved	1	10683
100	NCR	Intern Skive	0	15769
100	NCR	Intern Skive	1	2044
101	NCR	Bryggen Vejle	0	19411
101	NCR	Bryggen Vejle	1	6483
102	NCR	Aalborg Storcenter Afd	0	13832
102	NCR	Aalborg Storcenter Afd	1	4465
103	Diebold Nixdorf	Vejgaard	0	17151
103	Diebold Nixdorf	Vejgaard	1	4026

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;
```



Query completed, started on November 08, 2022 at 18:43:36
ELAPSED TIME: 00 m 02 s

Rows returned (2)

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejgaard	Friday	4055
2	NCR	Vejgaard	Friday	5421