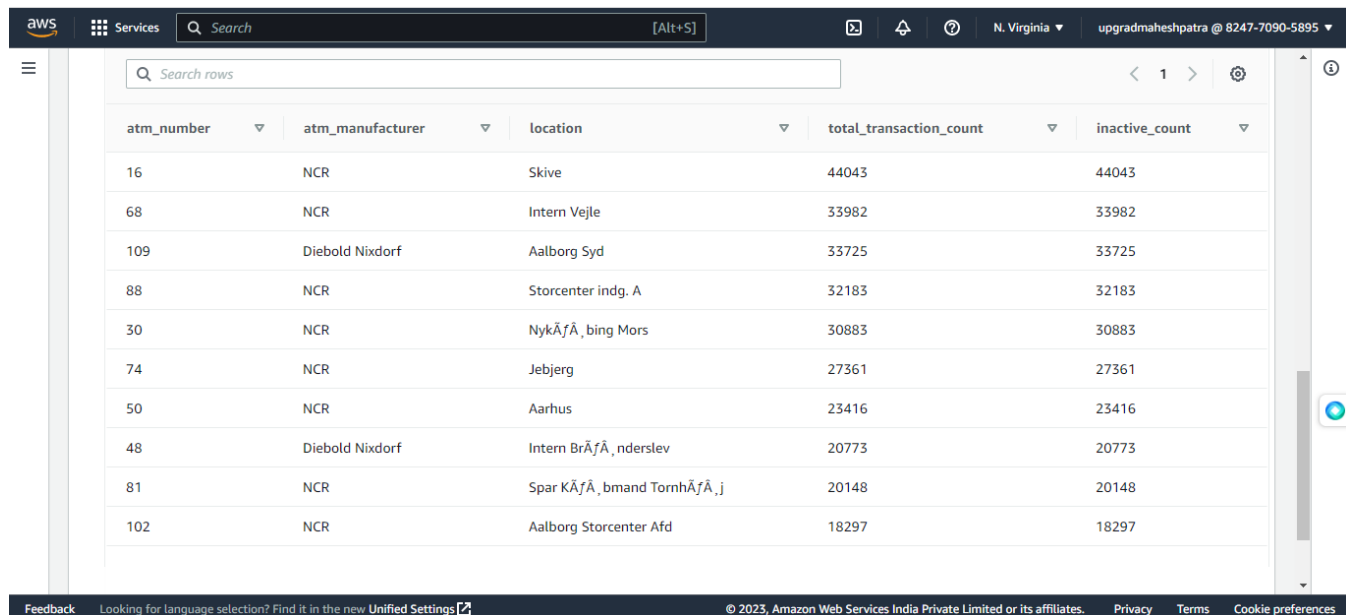


## Solving analytical queries on Redshift Cluster

Here, I have written 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

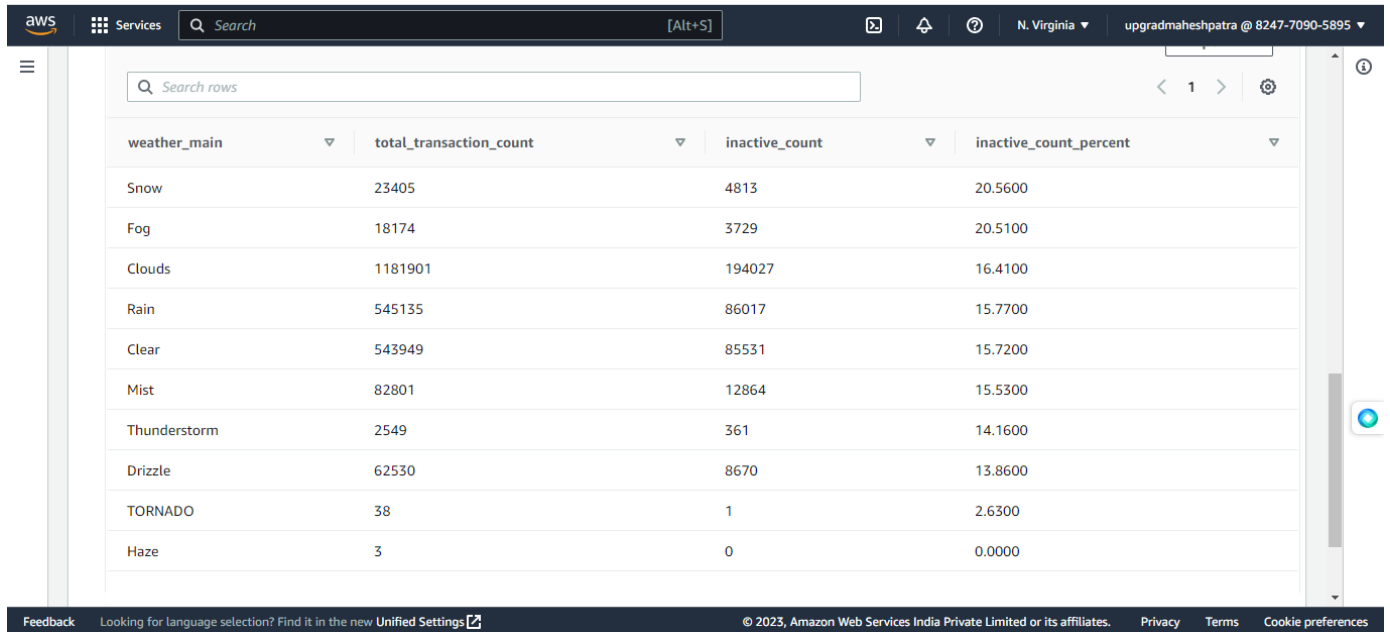
```
select a.atm_number, a.atm_manufacturer, l.location,
count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count
from bank_data.fact_atm_trans f, bank_data.dim_atm a, bank_data.dim_location l
where f.atm_id = a.atm_id and a.atm_location_id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



atm_number	atm_manufacturer	location	total_transaction_count	inactive_count
16	NCR	Skive	44043	44043
68	NCR	Intern Vejle	33982	33982
109	Diebold Nixdorf	Aalborg Syd	33725	33725
88	NCR	Storcenter indg. A	32183	32183
30	NCR	Nykjøbing Mors	30883	30883
74	NCR	Jebjerg	27361	27361
50	NCR	Aarhus	23416	23416
48	Diebold Nixdorf	Intern Brønderslev	20773	20773
81	NCR	Spar Kjøbenhavn, Tårnvej	20148	20148
102	NCR	Aalborg Storcenter Afd	18297	18297

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

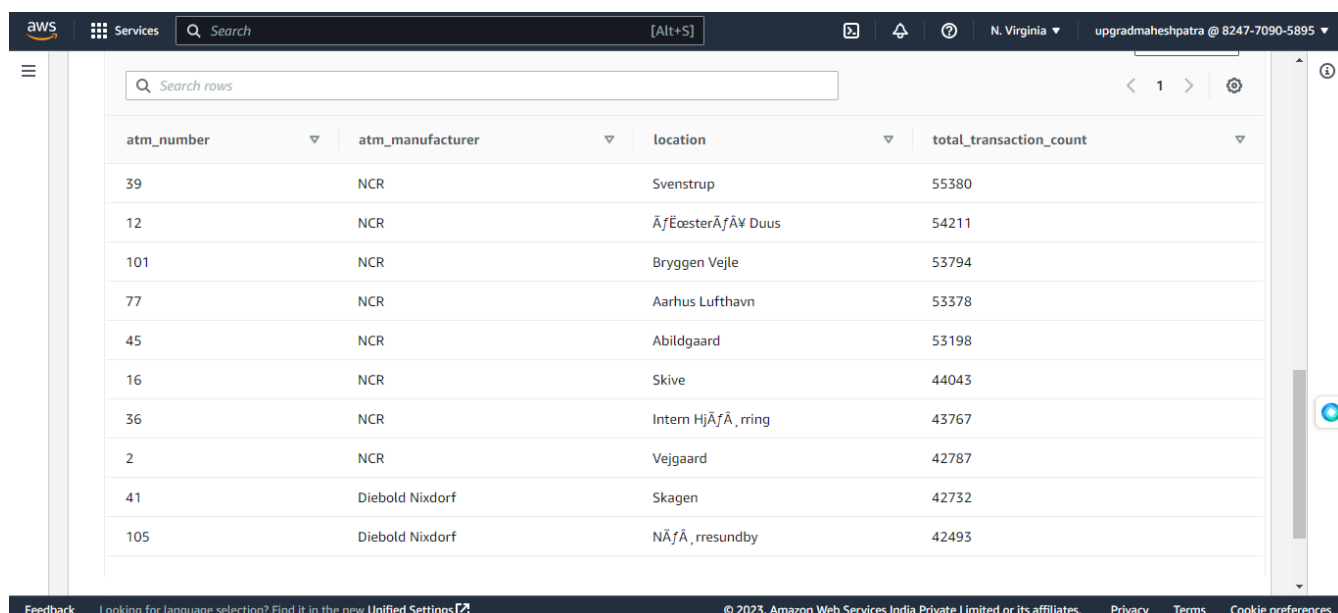
```
select f.weather_main, count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,
case when coalesce(inactive_count, 0) = 0 then 0.0000
else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2)
end as inactive_count_percent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



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

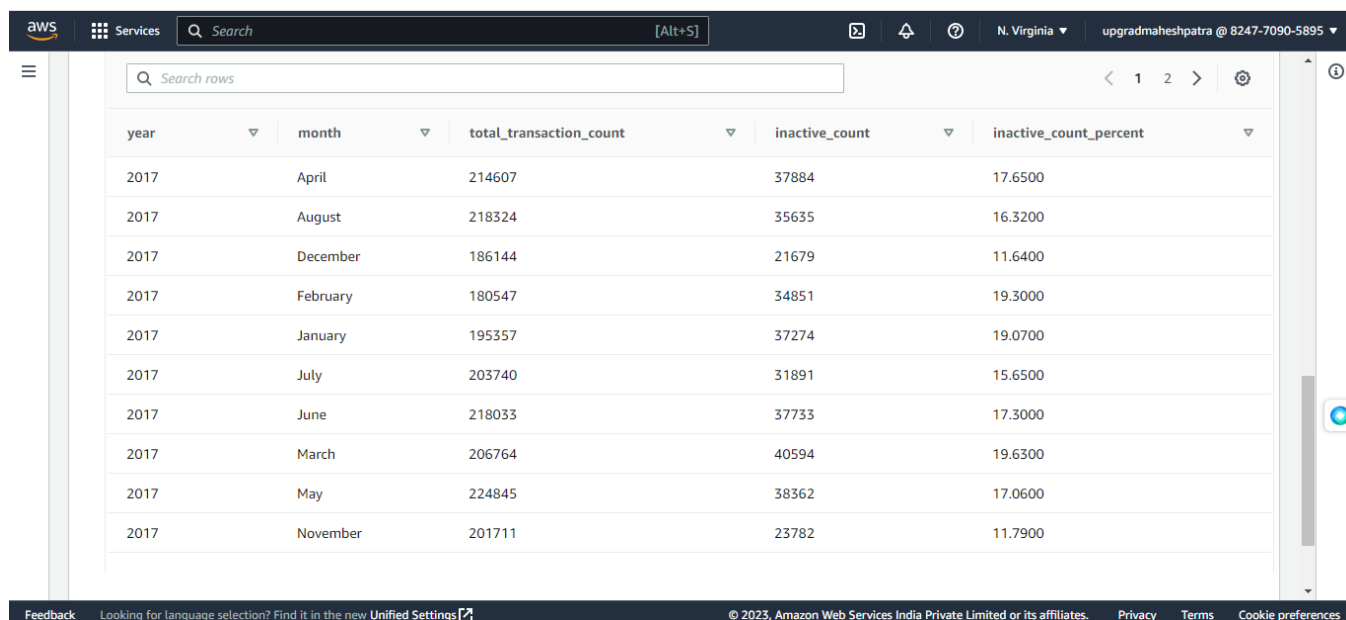
```
select a.atm_number, a.atm_manufacturer, l.location,
count(trans_id) as total_transaction_count
from bank_data.fact_atm_trans f, bank_data.dim_atm a, bank_data.dim_location l
where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
group by a.atm_number, a.atm_manufacturer, l.location
order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



atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
12	NCR	Århus Lufthavn	54211
101	NCR	Bryggen Vejle	53794
77	NCR	Aarhus Lufthavn	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
36	NCR	Intern Hjørring	43767
2	NCR	Vejgaard	42787
41	Diebold Nixdorf	Skagen	42732
105	Diebold Nixdorf	Næstved	42493

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

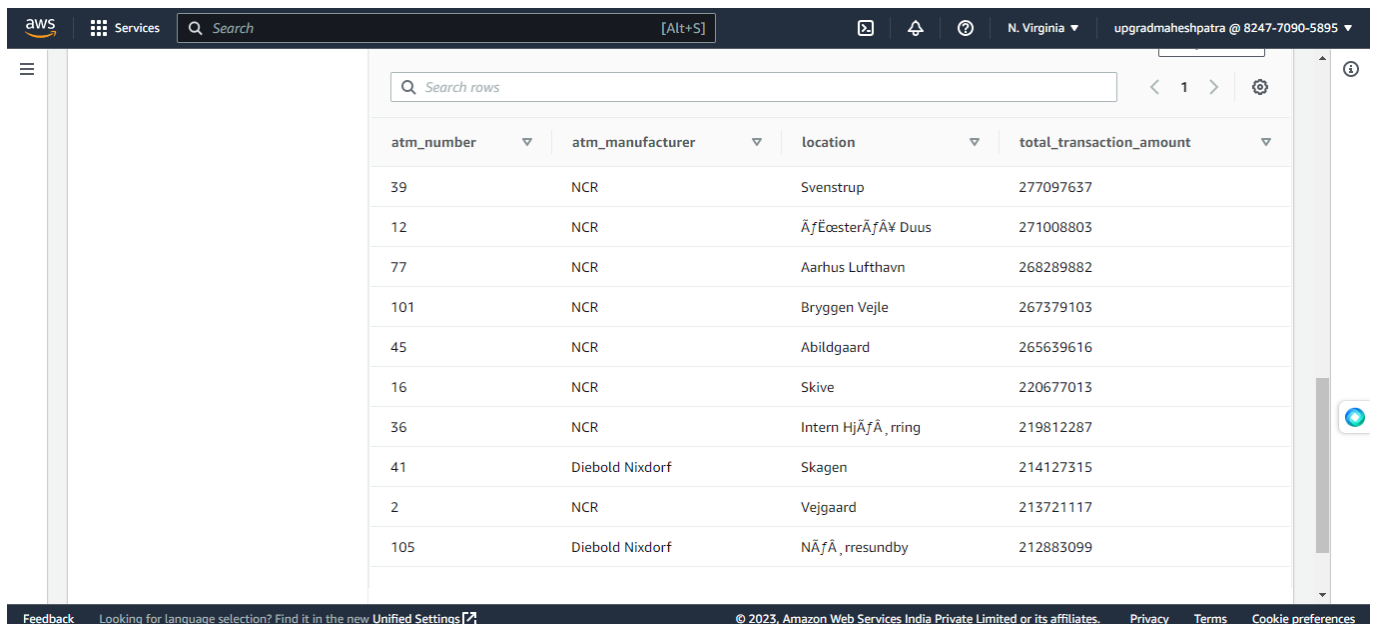
```
select d.year, d.month, count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,
case when coalesce(inactive_count, 0) = 0 then 0.0000
else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2)
end as inactive_count_percent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



year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	214607	37884	17.6500
2017	August	218324	35635	16.3200
2017	December	186144	21679	11.6400
2017	February	180547	34851	19.3000
2017	January	195357	37274	19.0700
2017	July	203740	31891	15.6500
2017	June	218033	37733	17.3000
2017	March	206764	40594	19.6300
2017	May	224845	38362	17.0600
2017	November	201711	23782	11.7900

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

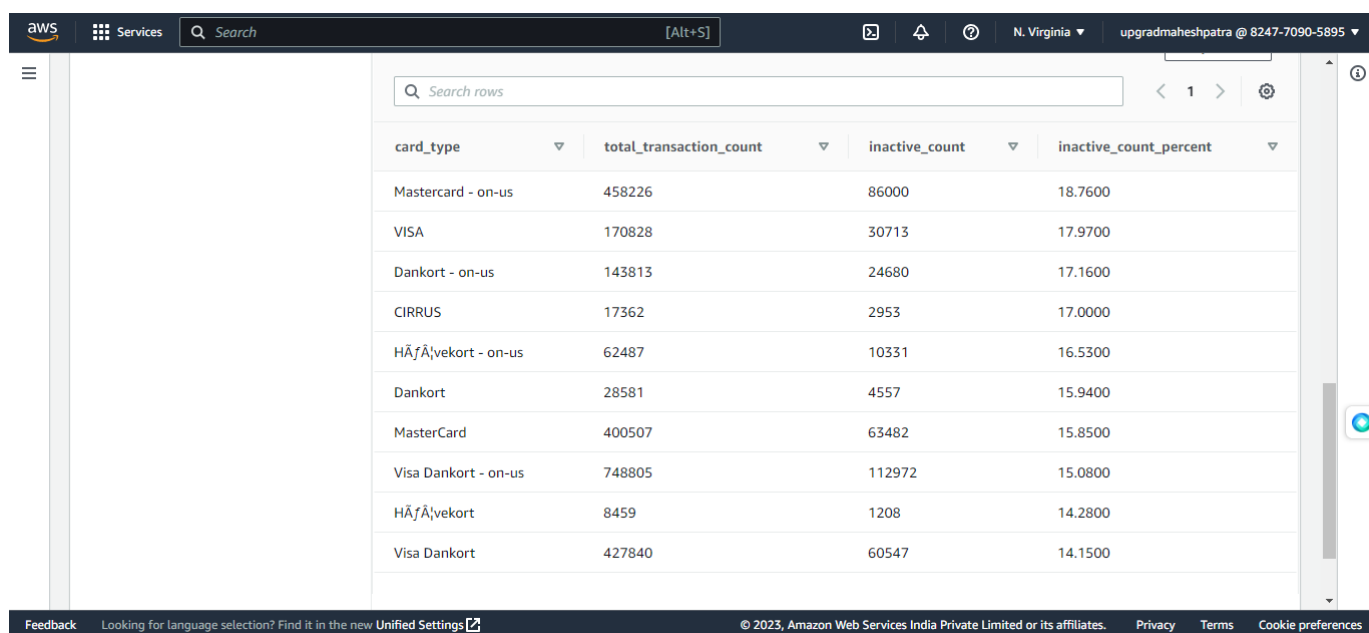
```
select a.atm_number, a.atm_manufacturer, l.location,
sum(transaction_amount) as total_transaction_amount
from bank_data.fact_atm_trans f, bank_data.dim_atm a, bank_data.dim_location l
where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
group by a.atm_number, a.atm_manufacturer, l.location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



atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
12	NCR	ÅfjæsterÅfÅ Duus	271008803
77	NCR	Aarhus Lufthavn	268289882
101	NCR	Bryggen Vejle	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
36	NCR	Intern HjøfÅ, rring	219812287
41	Diebold Nixdorf	Skagen	214127315
2	NCR	Vejgaard	213721117
105	Diebold Nixdorf	NÅfÅ, resundby	212883099

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

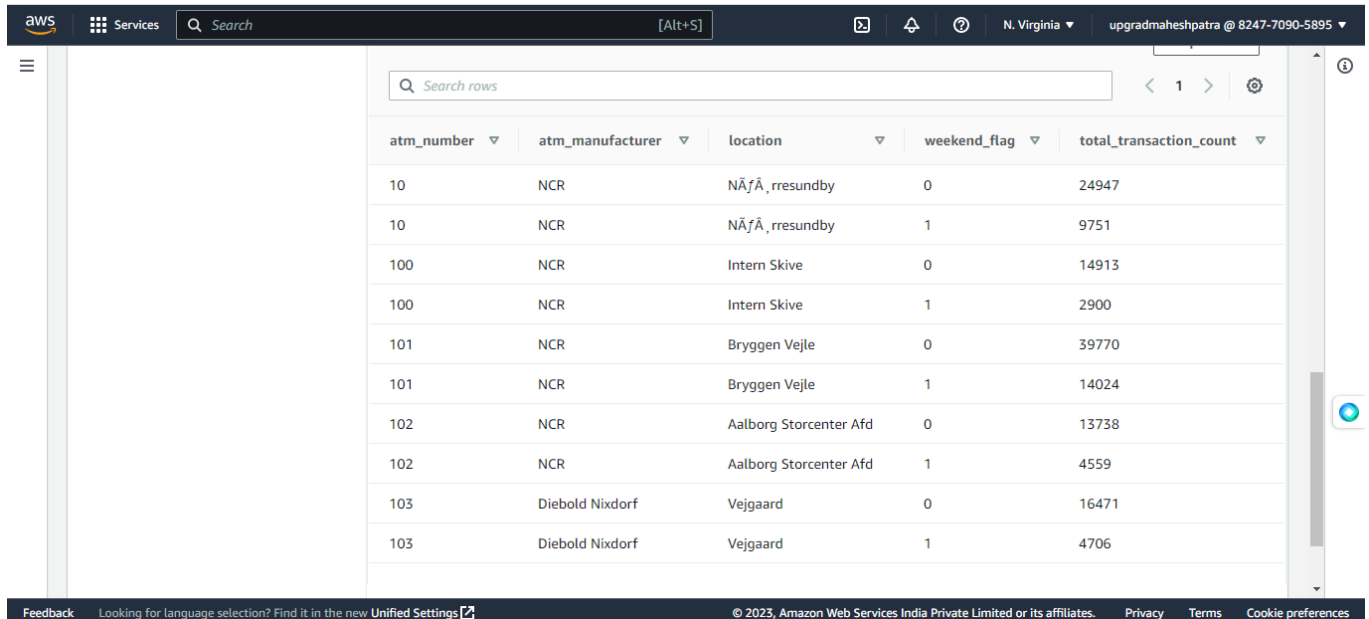
```
select ct.card_type, count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,
case when coalesce(inactive_count, 0) = 0 then 0.0000
else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2)
end as inactive_count_percent
from bank_data.fact_atm_trans f, bank_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
HÃ¶fÃ¶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
HÃ¶fÃ¶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 bank_data.fact_atm_trans f, bank_data.dim_atm a, bank_data.dim_location l,
bank_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;
```

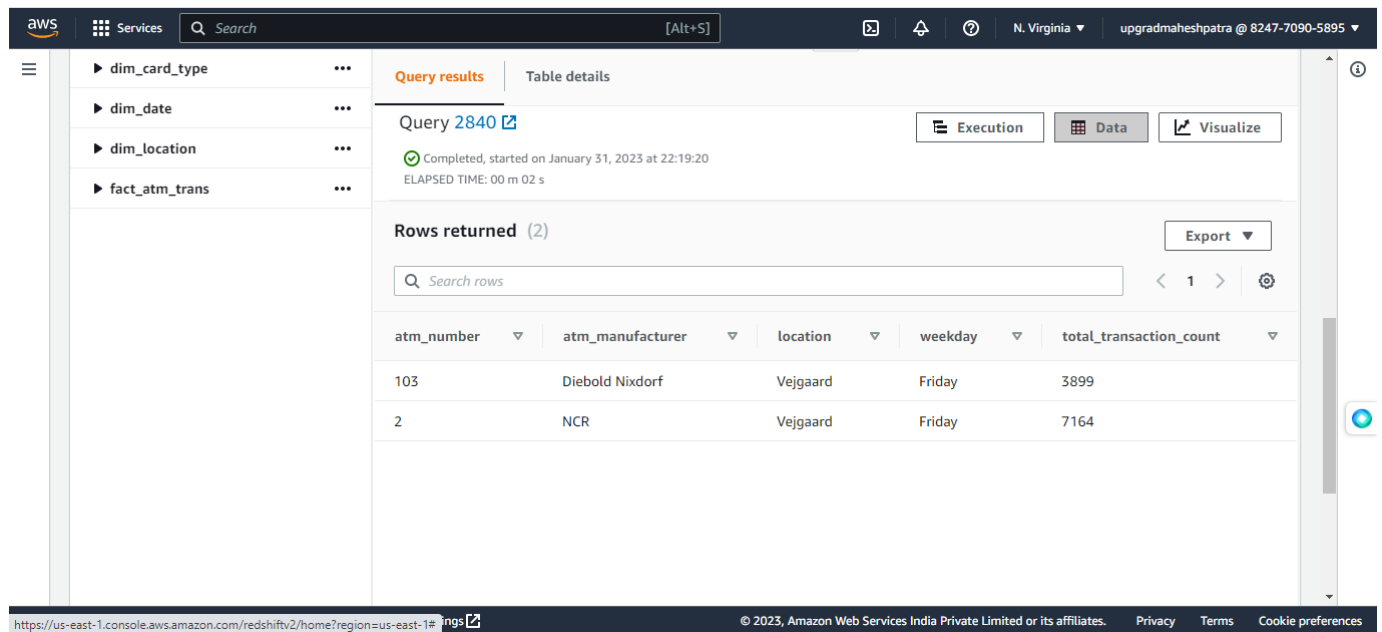


The screenshot shows the AWS Data console interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, a search bar, and user information. Below this is a table with 5 columns: atm\_number, atm\_manufacturer, location, weekend\_flag, and total\_transaction\_count. The table contains 10 rows of data, sorted by atm\_number, then atm\_manufacturer, then location, then weekend\_flag, and finally total\_transaction\_count in descending order.

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
10	NCR	NÅfÅ ,rresundby	0	24947
10	NCR	NÅfÅ ,rresundby	1	9751
100	NCR	Intern Skive	0	14913
100	NCR	Intern Skive	1	2900
101	NCR	Bryggen Vejle	0	39770
101	NCR	Bryggen Vejle	1	14024
102	NCR	Aalborg Storcenter Afd	0	13738
102	NCR	Aalborg Storcenter Afd	1	4559
103	Diebold Nixdorf	Vejgaard	0	16471
103	Diebold Nixdorf	Vejgaard	1	4706

## 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 bank_data.fact_atm_trans f inner join bank_data.dim_atm a on f.atm_id = a.atm_id
inner join bank_data.dim_location l on a.atm_location_id = l.location_id
inner join bank_data.dim_date d on f.date_id = d.date_id
where l.location = 'Vejgaard' and d.weekday in
( select d.weekday from bank_data.fact_atm_trans f inner join bank_data.dim_date d
on f.date_id = d.date_id
inner join bank_data.dim_location l on f.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;
```



The screenshot shows the AWS Redshift console interface. The left sidebar contains a navigation menu with items like 'dim\_card\_type', 'dim\_date', 'dim\_location', and 'fact\_atm\_trans'. The main panel displays the 'Query results' for a query with ID 2840. The query is completed, started on January 31, 2023 at 22:19:20, and took 00 m 02 s to execute. The results show 2 rows returned. The table has columns: atm\_number, atm\_manufacturer, location, weekday, and total\_transaction\_count. The data rows are:

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejgaard	Friday	3899
2	NCR	Vejgaard	Friday	7164

The bottom of the screenshot shows the URL: <https://us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#ings> and the footer text: © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences.