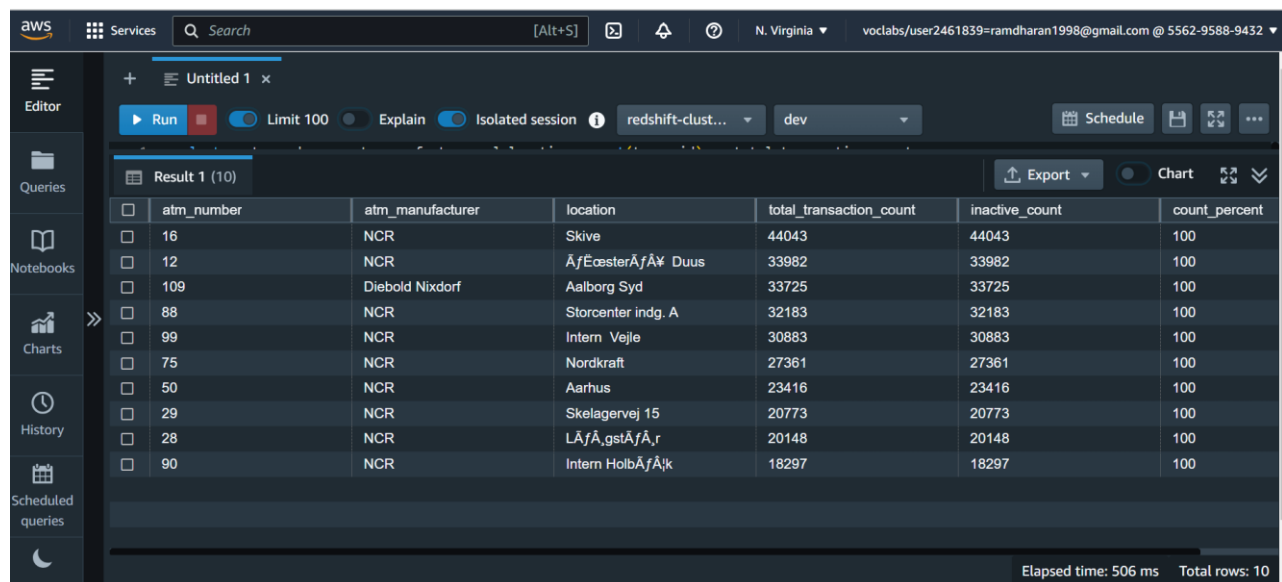


## Solving analytical queries on Redshift Cluster

Queries 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;
```



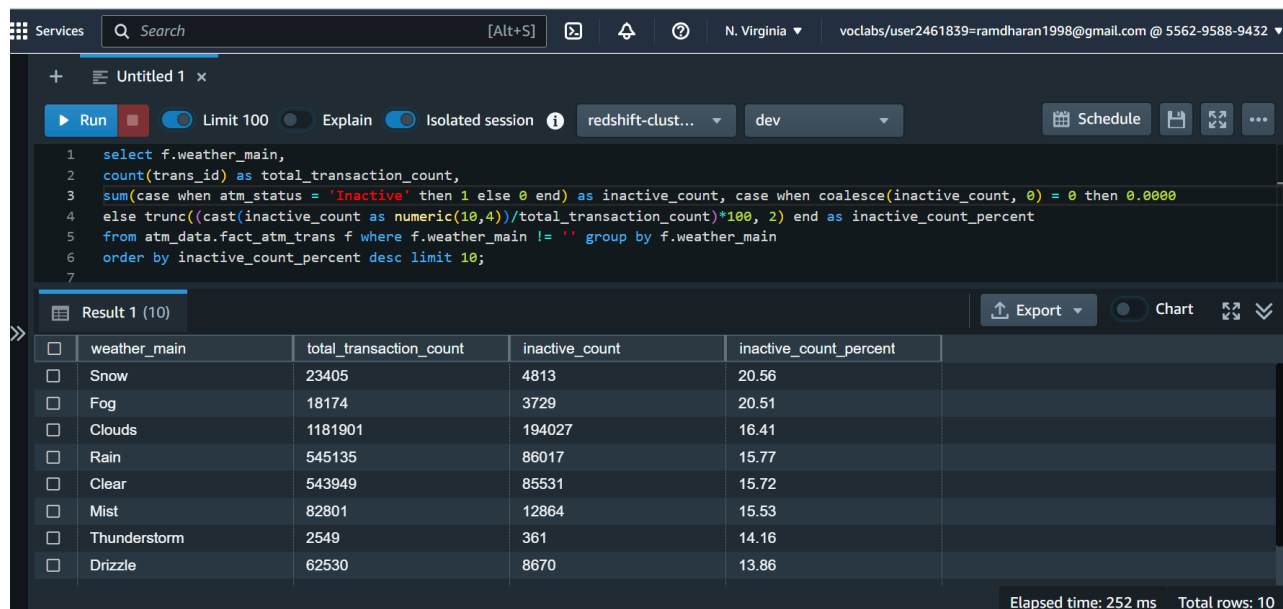
The screenshot shows the AWS Redshift Query Editor interface. The query results are displayed in a table with the following columns: atm\_number, atm\_manufacturer, location, total\_transaction\_count, inactive\_count, and count\_percent. The results are sorted by inactive\_count in descending order.

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
16	NCR	Skive	44043	44043	100
12	NCR	ÅfËcesterÅfÅr Duus	33982	33982	100
109	Diebold Nixdorf	Aalborg Syd	33725	33725	100
88	NCR	Storcenter indg. A	32183	32183	100
99	NCR	Intern Vejle	30883	30883	100
75	NCR	Nordkraft	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
28	NCR	LÅfÅgstÅfÅr	20148	20148	100
90	NCR	Intern HolbÅfÅk	18297	18297	100

Elapsed time: 506 ms Total rows: 10

## 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 a SQL IDE interface with a query editor and a results pane. The query is the same as the one provided in the previous block. The results pane shows a table with 10 rows of data, ordered by the percentage of inactive transactions in descending order.

weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.56
Fog	18174	3729	20.51
Clouds	1181901	194027	16.41
Rain	545135	86017	15.77
Clear	543949	85531	15.72
Mist	82801	12864	15.53
Thunderstorm	2549	361	14.16
Drizzle	62530	8670	13.86

Elapsed time: 252 ms Total rows: 10

### 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;
```

Services Search [Alt+S] N. Virginia voclabs/user2461839=ramdharan1998@gmail.com @ 5562-9588-9432

Untitled 1 x

Run Limit 100 Explain Isolated session redshift-clust... dev Schedule

```
1 select a.atm_number, a.atm_manufacturer, l.location, count(trans_id) as total_transaction_count
2 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
3 group by a.atm_number, a.atm_manufacturer, l.location order by total_transaction_count desc
4 limit 10;
5
```

Result 1 (10) Export Chart

	atm_number	atm_manufacturer	location	total_transaction_count
<input type="checkbox"/>	31	NCR	Slagelse	88076
<input type="checkbox"/>	39	NCR	Svenstrup	55380
<input type="checkbox"/>	20	NCR	Bispensgade	54211
<input type="checkbox"/>	71	NCR	AalbÅfÅk	53794
<input type="checkbox"/>	45	NCR	Abildgaard	53198
<input type="checkbox"/>	16	NCR	Skive	44043
<input type="checkbox"/>	40	Diebold Nixdorf	Frederikshavn	43767
<input type="checkbox"/>	2	NCR	Vejgaard	42787

Elapsed time: 226 ms Total rows: 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
```

Run Limit 100 Explain Isolated session redshift-clust... dev Schedule

```
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, case when coalesce(inactive_count, 0) = 0 then 0.0000
4 else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2) end as inactive_count_percent
5 from atm_data.fact_atm_trans f inner join atm_data.dim_date d on f.date_id = d.date_id
6 group by d.year, d.month order by d.year, d.month
7
```

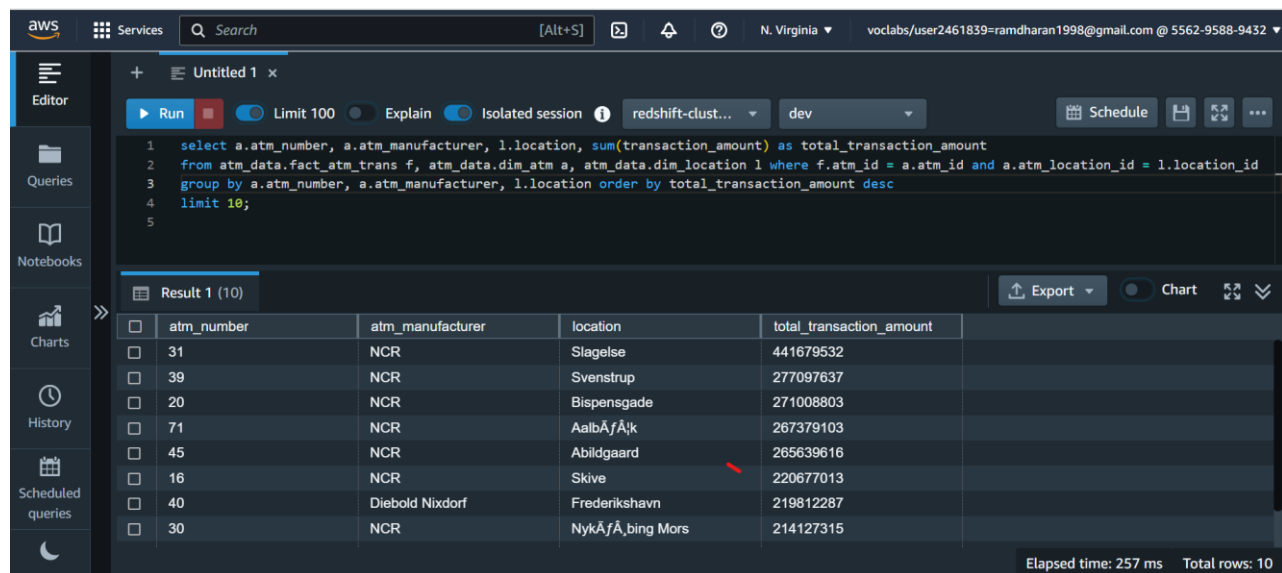
Result 1 (12)

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218750	38218	17.47
2017	August	220164	35110	15.94
2017	December	202711	23034	11.36
2017	February	174692	33812	19.35
2017	January	208576	40745	19.53
2017	July	215792	33970	15.74
2017	June	214781	37404	17.41
2017	March	200838	38169	19

Elapsed time: 8453 ms Total rows: 12

## 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 SQL query is executed, and the results are displayed in a table. The table has 5 columns: atm\_number, atm\_manufacturer, location, total\_transaction\_amount, and an empty column. The results are sorted by total\_transaction\_amount in descending order.

atm_number	atm_manufacturer	location	total_transaction_amount	
31	NCR	Slagelse	441679532	
39	NCR	Svenstrup	277097637	
20	NCR	Bispensgade	271008803	
71	NCR	AalbÃfÃk	267379103	
45	NCR	Abildgaard	265639616	
16	NCR	Skive	220677013	
40	Diebold Nixdorf	Frederikshavn	219812287	
30	NCR	NykÃfÃbing Mors	214127315	

Elapsed time: 257 ms Total rows: 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;
```

Run Limit 100 Explain Isolated session redshift-clust... dev Schedule

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

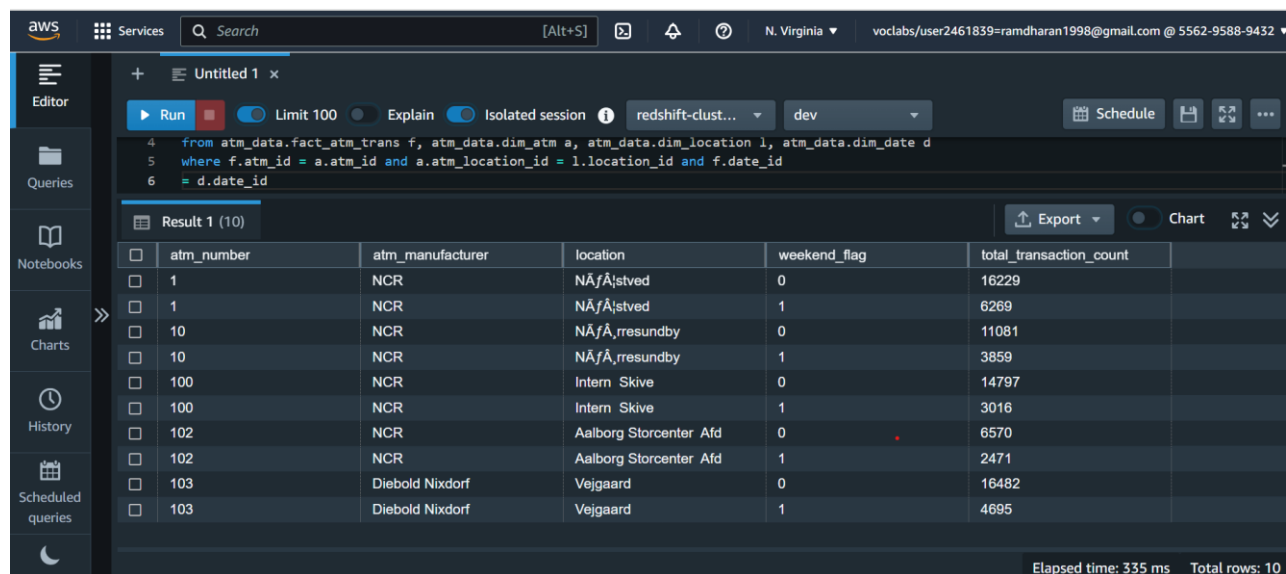
Result 1 (10) Export Chart

card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.76
VISA	170828	30713	17.97
Dankort - on-us	143813	24680	17.16
CIRRUS	17362	2953	17
HÅfÅ\vekort - on-us	62487	10331	16.53
Dankort	28581	4557	15.94
MasterCard	400507	63482	15.85
Visa Dankort - on-us	748805	112972	15.08

Elapsed time: 246 ms Total rows: 10

**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;
```



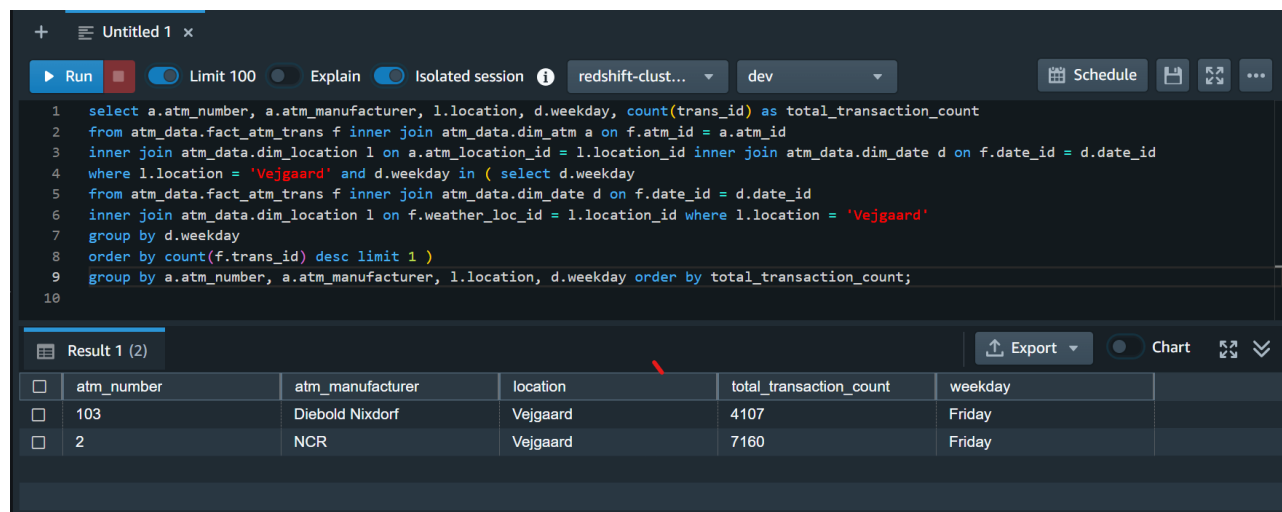
The screenshot shows the AWS Redshift console interface. The SQL query is executed, and the results are displayed in a table. The table has 6 columns: atm\_number, atm\_manufacturer, location, weekend\_flag, and total\_transaction\_count. The results are ordered by atm\_number, atm\_manufacturer, location, weekend\_flag, and total\_transaction\_count.

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃfÃstved	0	16229
1	NCR	NÃfÃstved	1	6269
10	NCR	NÃfÃresundby	0	11081
10	NCR	NÃfÃresundby	1	3859
100	NCR	Intern Skive	0	14797
100	NCR	Intern Skive	1	3016
102	NCR	Aalborg Storcenter Afd	0	6570
102	NCR	Aalborg Storcenter Afd	1	2471
103	Diebold Nixdorf	Vejgaard	0	16482
103	Diebold Nixdorf	Vejgaard	1	4695

Elapsed time: 335 ms Total rows: 10

## 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.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 a SQL IDE interface with a query editor and a results pane. The query is the same as the one provided in the previous block. The results pane shows the output of the query, which is a table with 6 columns: atm\_number, atm\_manufacturer, location, total\_transaction\_count, weekday, and an empty column. The results are sorted by total\_transaction\_count in descending order.

atm_number	atm_manufacturer	location	total_transaction_count	weekday	
103	Diebold Nixdorf	Vejgaard	4107	Friday	
2	NCR	Vejgaard	7160	Friday	