

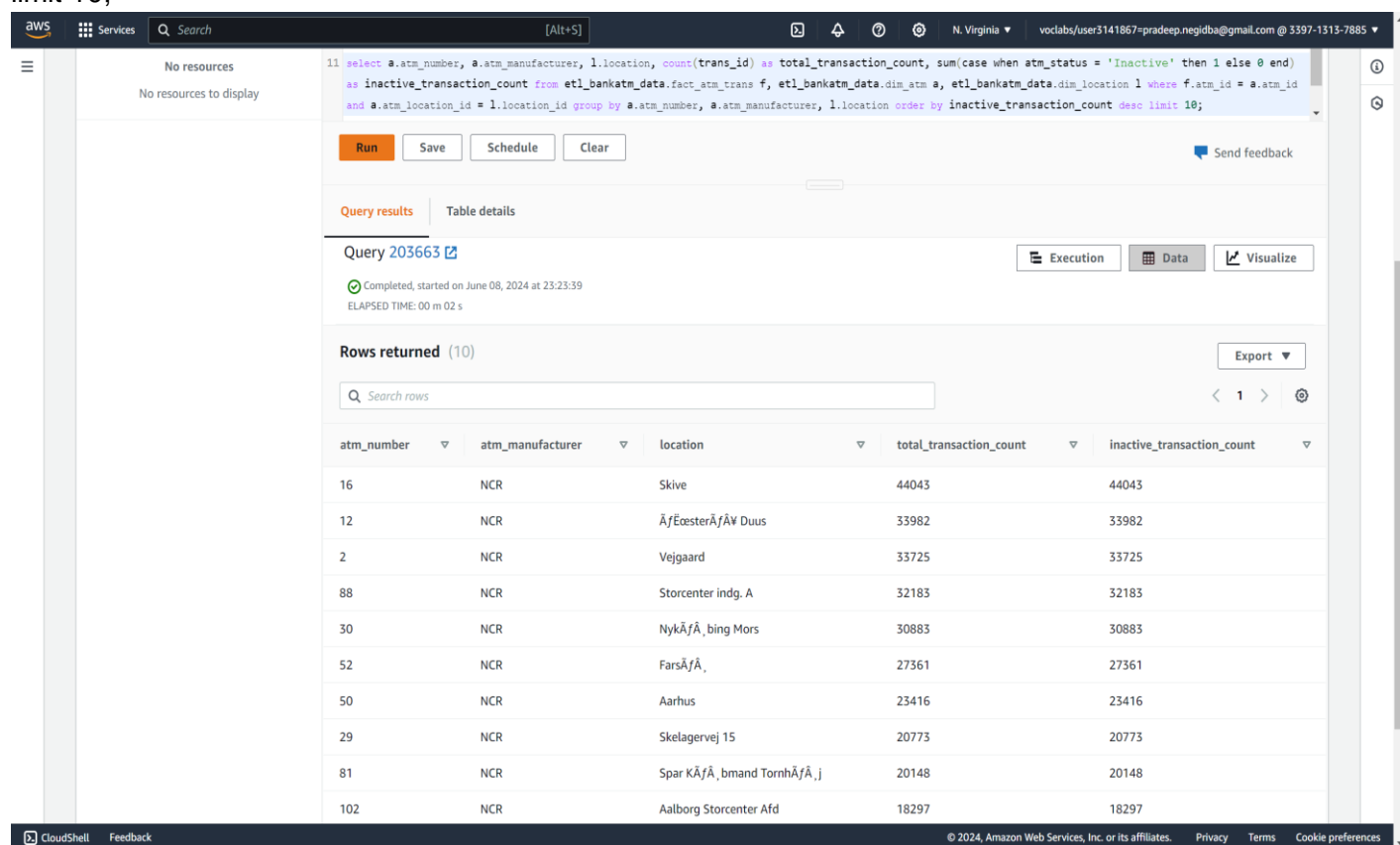
# Solving analytical queries on Redshift Cluster

## 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
From etl_bankatm_data.fact_atm_trans f,
etl_bankatm_data.dim_atm a,
etl_bankatm_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_transaction_count desc
limit 10;

```



The screenshot shows the AWS CloudShell interface with a SQL query executed on a Redshift cluster. The query results are displayed in a table with 10 rows, showing ATM details and transaction counts.

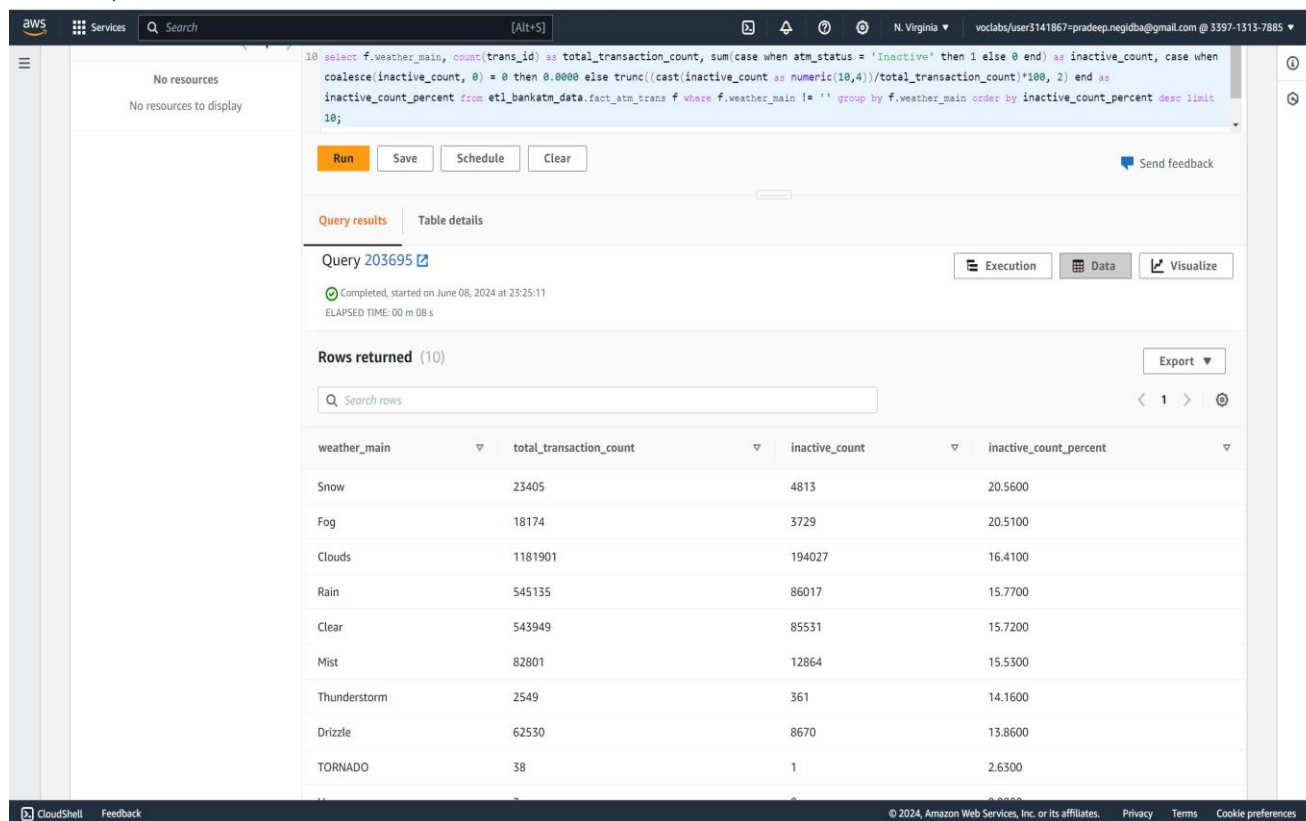
atm_number	atm_manufacturer	location	total_transaction_count	inactive_transaction_count
16	NCR	Skive	44043	44043
12	NCR	ÅfjæsterÅfjæ Duus	33982	33982
2	NCR	Vejgaard	33725	33725
88	NCR	Storcenter indg. A	32183	32183
30	NCR	NykÅfÅ, bing Mors	30883	30883
52	NCR	FarsÅfÅ,	27361	27361
50	NCR	Aarhus	23416	23416
29	NCR	Skelagervej 15	20773	20773
81	NCR	Spar KÅfÅ, bmand TornhÅfÅ, j	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 etl_ankatm_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 CloudShell interface with the query execution results. The query is labeled "Query 203695" and is completed. The results are displayed in a table with 4 columns: weather\_main, total\_transaction\_count, inactive\_count, and inactive\_count\_percent. The table shows 10 rows of data, ordered by inactive\_count\_percent in descending order.

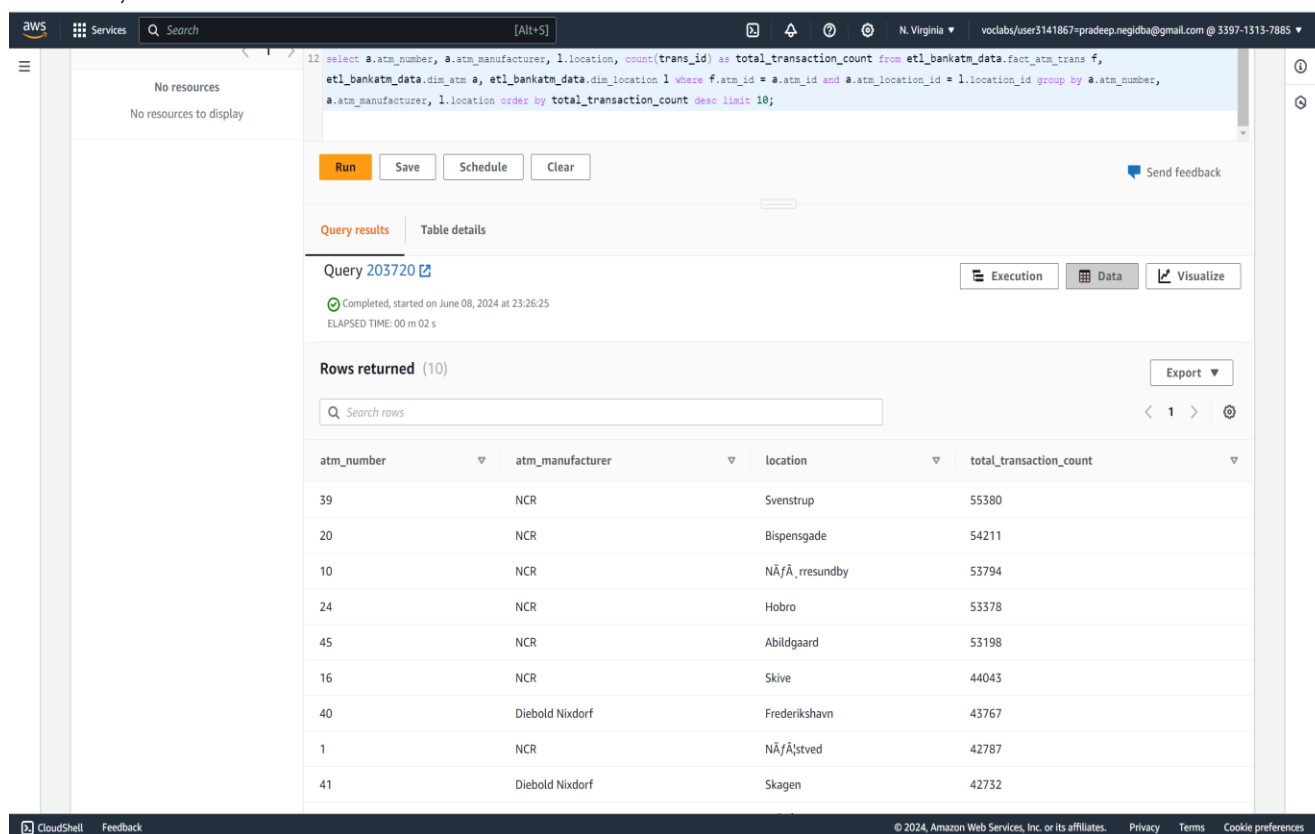
weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.5600
Fog	18174	3729	20.5100
Clouds	1181901	194027	16.4100
Rain	545135	86017	15.7700
Clear	543949	85531	15.7200
Mist	82801	12864	15.5300
Thunderstorm	2549	361	14.1600
Drizzle	62530	8670	13.8600
TORNADO	38	1	2.6300

### 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
etl_bankatm_data.fact_atm_trans f,
etl_bankatm_data.dim_atm a,
etl_bankatm_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;

```



The screenshot shows the AWS CloudShell interface with a SQL query executed successfully. The query results are displayed in a table with 10 rows, showing the top 10 ATMs by transaction count.

atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÅfÅ, resundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÅfÅstved	42787
41	Diebold Nixdorf	Skagen	42732

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

```
Select c.year,
       c.month,
       c.transaction_count,
       d.inactive_count,
       CAST(trunc(100.0 * d.inactive_count / c.transaction_count,2) AS NUMERIC(10, 4)
       ) as inactive_count_percent
from
  (select a.year,
         a.month,
         count(b.trans_id) as transaction_count
  from
    etl_bankatm_data.dim_date a,
    etl_bankatm_data.FACT_ATM_TRANS b
  where a.date_id = b.date_id
  group by
    a.month,
    a.year
  ) c
left join (select a.year,
                 a.month,
                 count(b.atm_status) as inactive_count
  from
    etl_bankatm_data.dim_date a,
    etl_bankatm_data.FACT_ATM_TRANS b
  where
    a.date_id = b.date_id
    and b.atm_status = 'Inactive'
  group by
    a.month,
    a.year
  ) d on c.year = d.year
and c.month = d.month
order by
  c.year,
  c.month;
```

Services
Search
[Alt+S]

---

```

11 select c.year, c.month, c.transaction_count, d.inactive_count, CAST(trunc(100.0*d.inactive_count/c.transaction_count,2) AS NUMERIC(10,4)) as
   inactive_count_percent from (select a.year, a.month, count(b.trans_id) as transaction_count from etl_bankatm_data.dim_date a,etl_bankatm_data.FACT_ATM_TRANS
   b where a.date_id = b.date_id group by a.month, a.year) c left join (select a.year, a.month, count(b.atm_status) as inactive_count from
   etl_bankatm_data.dim_date a,etl_bankatm_data.FACT_ATM_TRANS b where a.date_id = b.date_id and b.atm_status='Inactive' group by a.month, a.year) d on
   c.year=d.year and c.month=d.month order by c.year, c.month;
```

Run
Save
Schedule
Clear

Send feedback

Query results
Table details

---

Query [203748](#)

Completed, started on June 08, 2024 at 23:27:58  
ELAPSED TIME: 00 m 14 s

Execution
Data
Visualize

Rows returned (12)

Export ▼

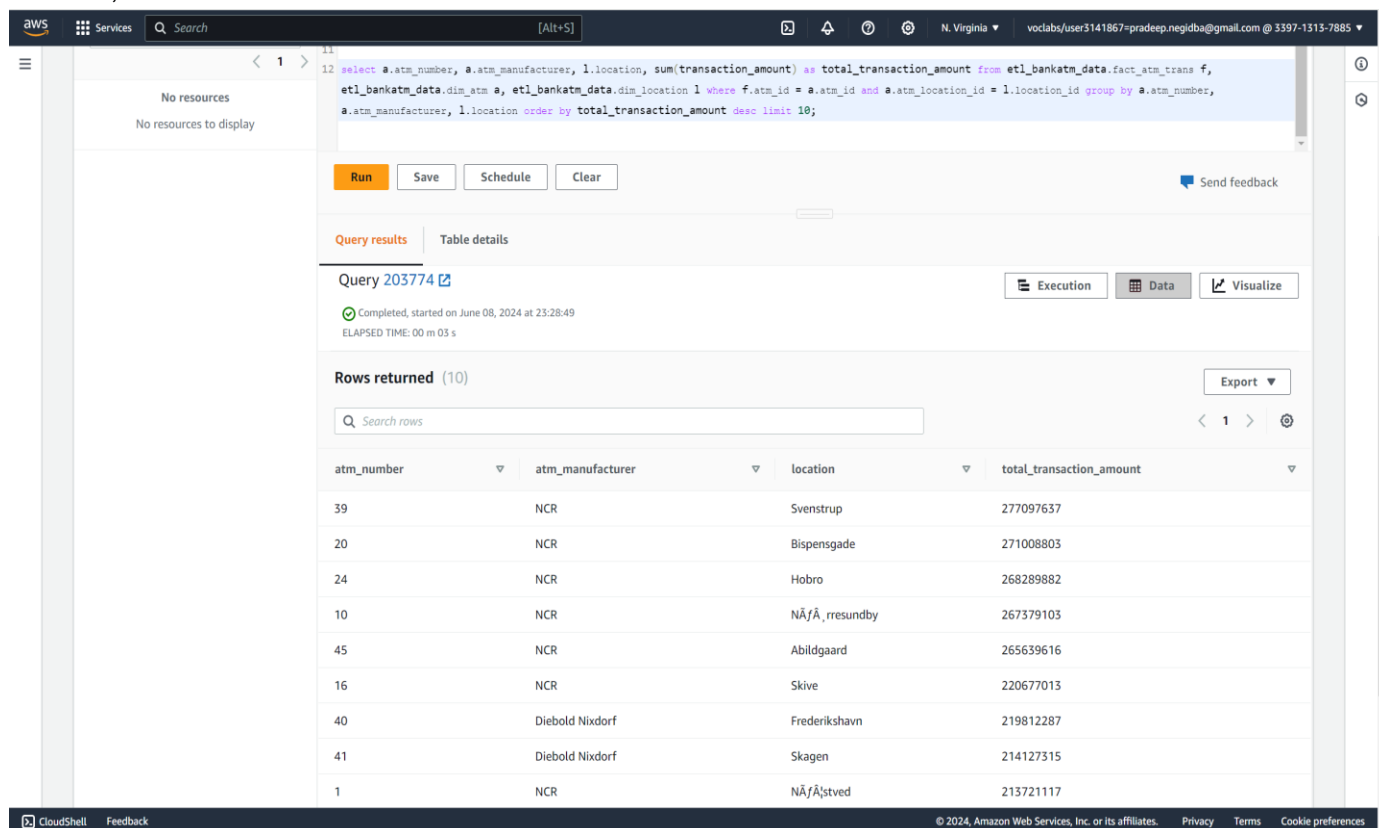
[<](#)
[1](#)
[2](#)
[>](#)

year	month	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

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy Terms Cookie preferences

## 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  
etl\_bankatm\_data.fact\_atm\_trans f,  
etl\_bankatm\_data.dim\_atm a,  
etl\_bankatm\_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;

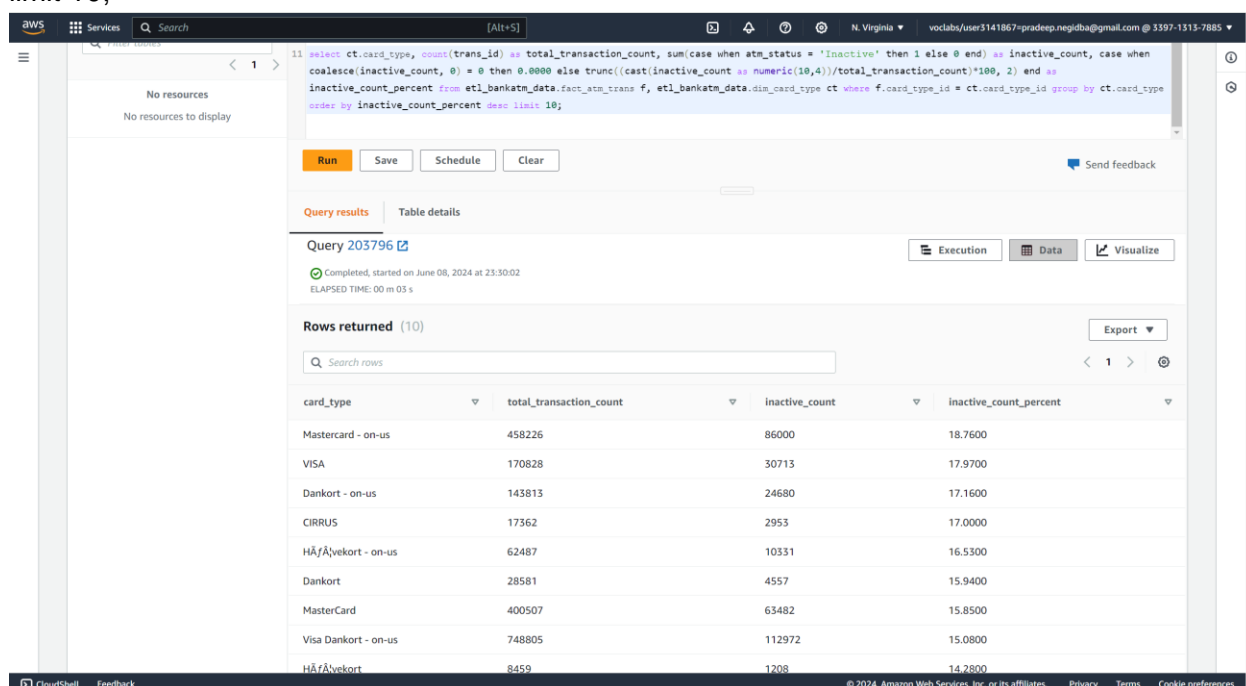


The screenshot shows the AWS CloudShell interface. The SQL query is entered in the editor and has been executed successfully. The results are displayed in a table with 4 columns: atm\_number, atm\_manufacturer, location, and total\_transaction\_amount. The results are ordered by total\_transaction\_amount in descending order, showing the top 10 ATMs.

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÅfÅ, rresundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	NÅfÅstved	213721117

## 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
etl_bankatm_data.fact_atm_trans f,
etl_bankatm_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 AWS CloudShell interface with the query execution results. The query is labeled "Query 203796" and was completed on June 08, 2024, at 23:30:02. The results are displayed in a table with 4 columns: card\_type, total\_transaction\_count, inactive\_count, and inactive\_count\_percent. The table shows the top 10 card types with the highest percentage of failed transactions.

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

**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  
  etl_bankatm_data.fact_atm_trans f,  
  etl_bankatm_data.dim_atm a,  
  etl_bankatm_data.dim_location l,  
  etl_bankatm_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;
```



aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3141867-pradeep.negidba@gmail.com @ 3397-1313-7885

No resources

No resources to display

12

```
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 etl_bankatm_data.fact_atm_trans f, etl_bankatm_data.dim_atm a, etl_bankatm_data.dim_location l, etl_bankatm_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;
```

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query 203821

Execution

Data

Visualize

Completed, started on June 08, 2024 at 23:31:27

ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export

Search rows

< 1 >

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÅfÅ\stved	0	32711
1	NCR	NÅfÅ\stved	1	10076
10	NCR	NÅfÅ, resundby	0	41667
10	NCR	NÅfÅ, resundby	1	12127
100	NCR	Intern Skive	0	17812
100	NCR	Intern Skive	1	1
101	NCR	Bryggen Vejle	0	11693
101	NCR	Bryggen Vejle	1	3247
102	NCR	Aalborg Storcenter Afd	0	14556

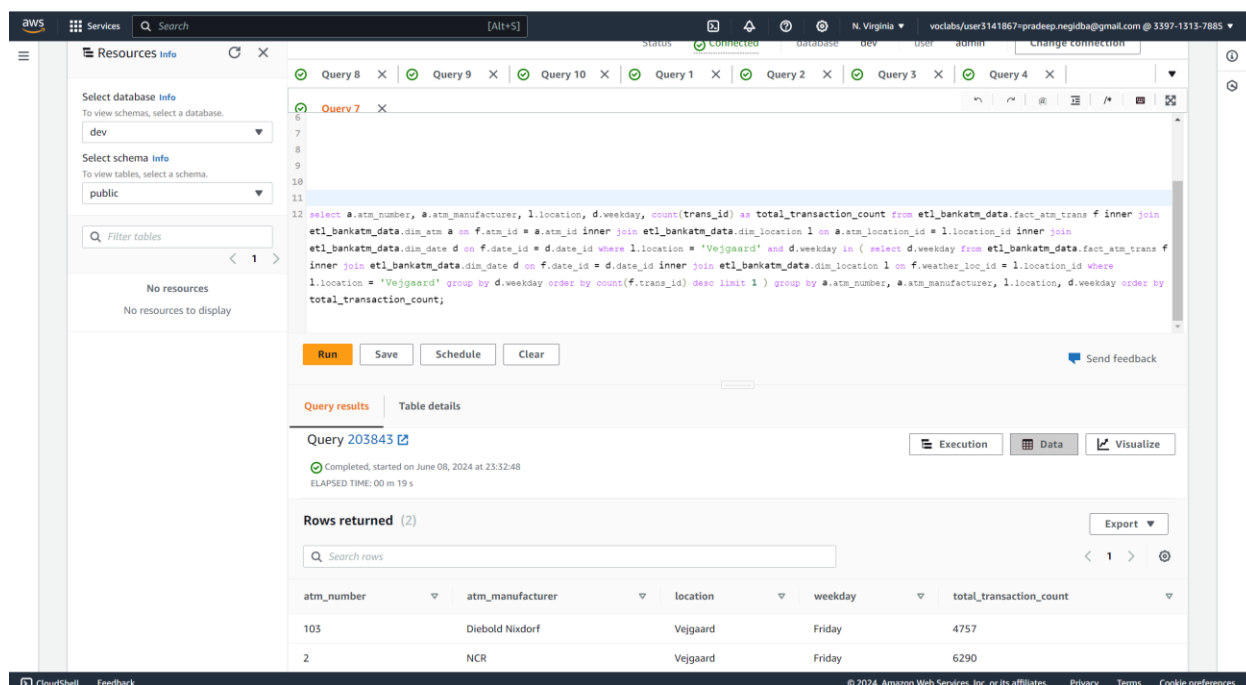
CloudShell

Feedback

© 2024, Amazon Web Services, Inc. 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   etl_bankatm_data.fact_atm_trans f
       inner join etl_bankatm_data.dim_atm a on f.atm_id = a.atm_id
       inner join etl_bankatm_data.dim_location l on a.atm_location_id = l.location_id
       inner join etl_bankatm_data.dim_date d on f.date_id = d.date_id
where  l.location = 'Vejgaard'
and d.weekday in (select d.weekday
                  from etl_bankatm_data.fact_atm_trans f
                  inner join etl_bankatm_data.dim_date d on f.date_id = d.date_id
                  inner join etl_bankatm_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. On the left, there's a sidebar with 'Resources' and 'Info' sections. The main area displays a SQL query (Query 7) that is identical to the one in the previous block. Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Query results' section shows that the query was completed successfully on June 08, 2024, at 23:32:48, with an elapsed time of 00 m 19 s. The results are displayed in a table with 2 rows returned. The table has columns: atm\_number, atm\_manufacturer, location, weekday, and total\_transaction\_count.

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