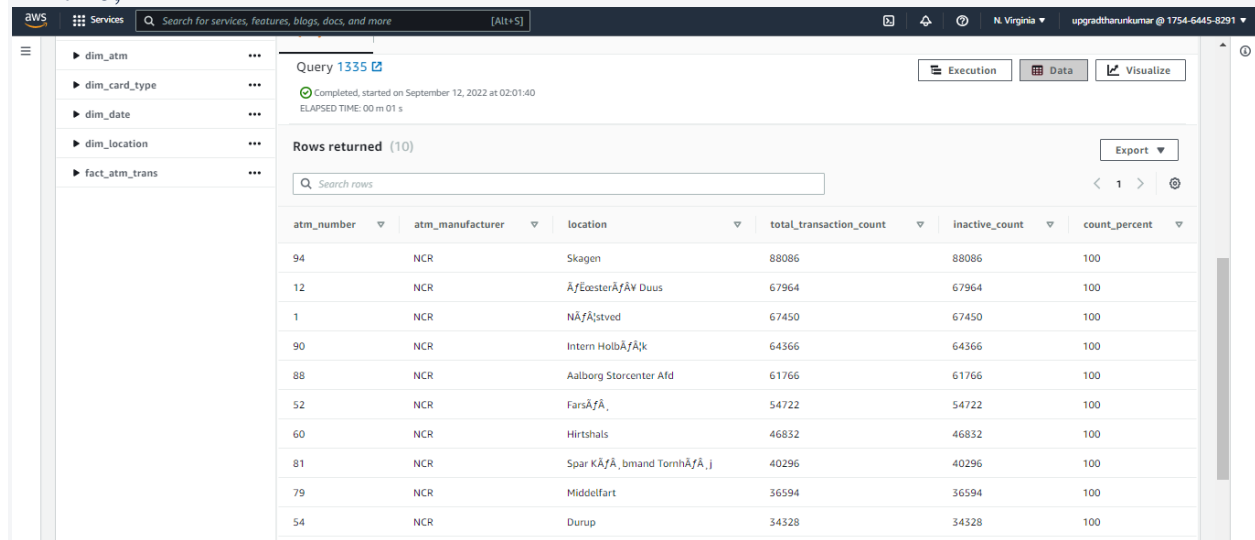


- There are the few analytical queries on a Redshift cluster to find the solutions.

## 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;
```



Query 1335

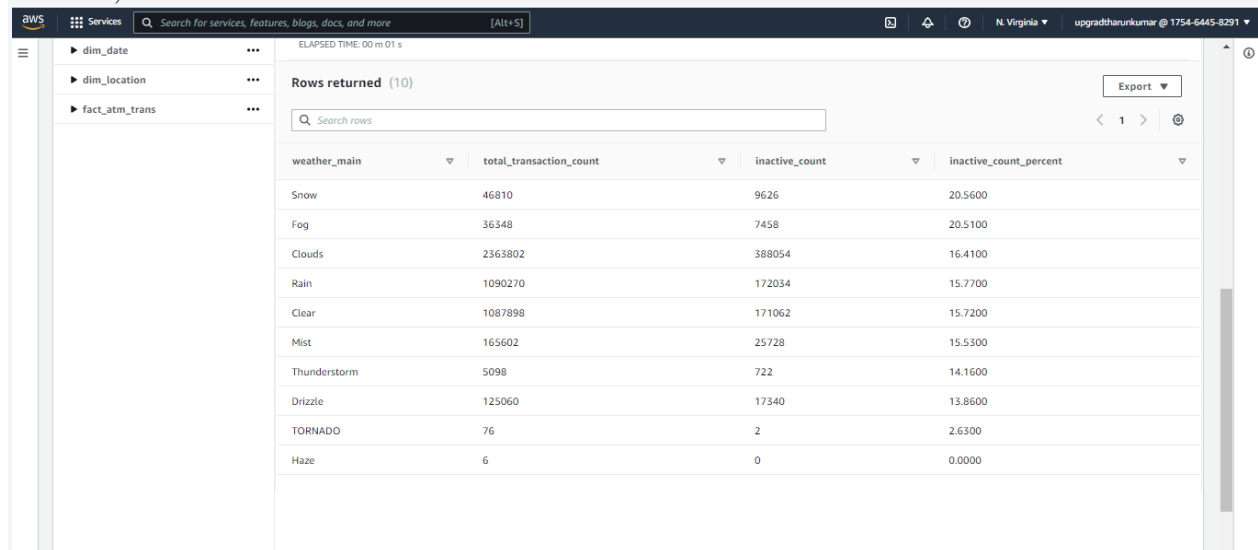
Completed, started on September 12, 2022 at 02:01:40  
ELAPSED TIME: 00 m 01 s

Rows returned (10)

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
94	NCR	Skagen	88086	88086	100
12	NCR	ÅfEøesterÅfÅv Duus	67964	67964	100
1	NCR	NÅfÅstved	67450	67450	100
90	NCR	Intern HolbÅfÅk	64366	64366	100
88	NCR	Aalborg Storcenter Afd	61766	61766	100
52	NCR	FarsÅfÅ	54722	54722	100
60	NCR	Hirtshals	46832	46832	100
81	NCR	Spar KÅfÅ, bmand TornhÅfÅ, j	40296	40296	100
79	NCR	Middelfart	36594	36594	100
54	NCR	Durup	34328	34328	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;
```



The screenshot shows the AWS Athena console interface. On the left, a sidebar lists the database schema: `dim_date`, `dim_location`, and `fact_atm_trans`. The main area displays the query results for the SQL statement provided above. The results are shown as 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 top row is 'Snow' with a total transaction count of 46810 and an inactive count of 9626, resulting in an inactive percentage of 20.5600. The bottom row is 'Haze' with a total transaction count of 6 and an inactive count of 0, resulting in an inactive percentage of 0.0000. The interface includes a search bar, a 'Rows returned (10)' indicator, and an 'Export' button.

weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	46810	9626	20.5600
Fog	36348	7458	20.5100
Clouds	2363802	388054	16.4100
Rain	1090270	172034	15.7700
Clear	1087898	171062	15.7200
Mist	165602	25728	15.5300
Thunderstorm	5098	722	14.1600
Drizzle	125060	17340	13.8600
TORNADO	76	2	2.6300
Haze	6	0	0.0000

### 3. Top 10 ATMs with the most number of transactions throughout the year

The screenshot shows a SQL query editor interface. The query is as follows:

```
1 select a.atm_number, a.atm_manufacturer, l.location,
2 count(trans_id) as total_transaction_count
3 from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l
4 where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
5 group by a.atm_number, a.atm_manufacturer, l.location
6 order by total_transaction_count desc
7 limit 10;
```

The query results are displayed in a table with the following columns: atm\_number, atm\_manufacturer, location, and total\_transaction\_count. The results are sorted in descending order of total\_transaction\_count.

atm_number	atm_manufacturer	location	total_transaction_count
65	NCR	Storvorde	110760
20	NCR	Bispensgade	108422
101	NCR	Bryggen Vejle	107588
24	NCR	Hobro	106756
45	NCR	Abildgaard	106396
94	NCR	Skagen	88086
40	Diebold Nixdorf	Frederikshavn	87534
109	Diebold Nixdorf	AalbåfÅk	85574
92	NCR	Intern HJÅfÅ, ring	85464
8	NCR	GlyngÅfÅ, re	84986

### 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

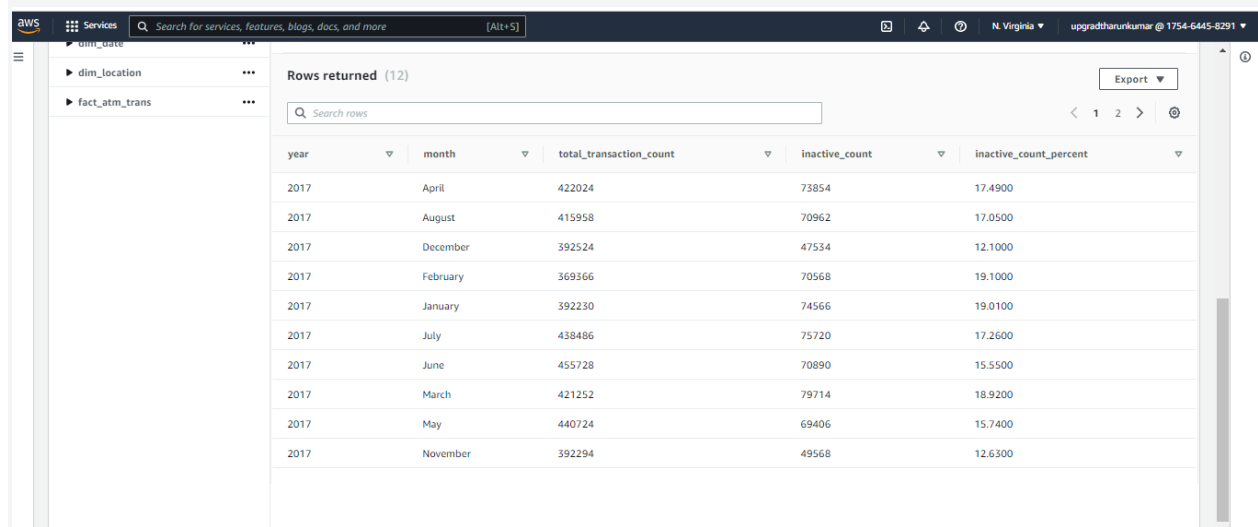
numeric(10,4))/total\_transaction\_count)\*100, 2)

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
```

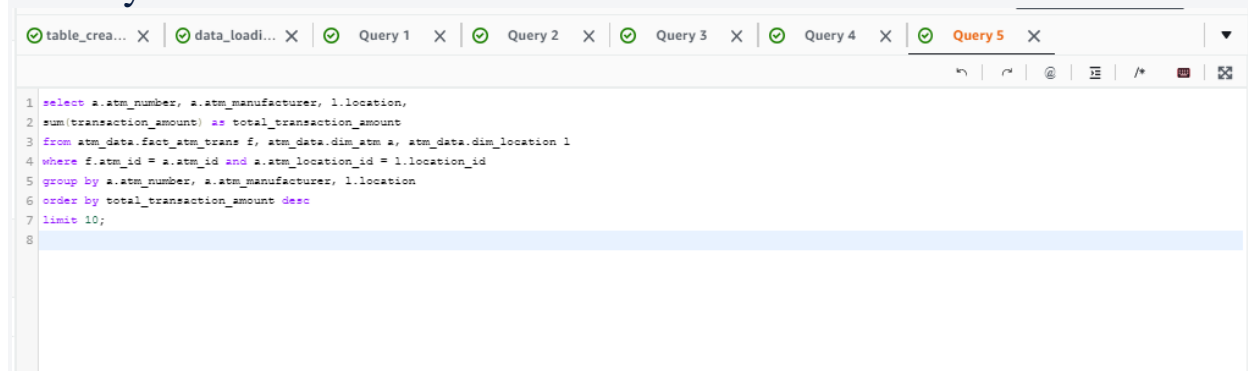
```
order by d.year, d.month
```



The screenshot shows the AWS Glue console interface. On the left, there is a sidebar with a search bar and a list of tables: 'dim\_location' and 'fact\_atm\_trans'. The main area displays the 'fact\_atm\_trans' table with 12 rows returned. The table has columns: 'year', 'month', 'total\_transaction\_count', 'inactive\_count', and 'inactive\_count\_percent'. The data is sorted by year (2017) and month (April, August, December, February, January, July, June, March, May, November).

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	422024	73854	17.4900
2017	August	415958	70962	17.0500
2017	December	392524	47534	12.1000
2017	February	369366	70568	19.1000
2017	January	392230	74566	19.0100
2017	July	438486	75720	17.2600
2017	June	455728	70890	15.5500
2017	March	421252	79714	18.9200
2017	May	440724	69406	15.7400
2017	November	392294	49568	12.6300

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



The screenshot shows the AWS Glue console interface with a SQL query editor. The query is as follows:

```
1 select a.atm_number, a.atm_manufacturer, l.location,  
2 sum(transaction_amount) as total_transaction_amount  
3 from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l  
4 where f.atm_id = a.atm_id and a.atm_location_id = l.location_id  
5 group by a.atm_number, a.atm_manufacturer, l.location  
6 order by total_transaction_amount desc  
7 limit 10;  
8
```

Rows returned (10)

atm_number	atm_manufacturer	location	total_transaction_amount
65	NCR	Storvorde	554195274
20	NCR	Bispensgade	542017606
24	NCR	Hobro	536579764
101	NCR	Bryggen Vejle	534758206
45	NCR	Abildgaard	531279232
94	NCR	Skagen	441354026
40	Diebold Nixdorf	Frederikshavn	439624574
92	NCR	Intern Hjørning	428254630
109	Diebold Nixdorf	Aalbæk	427442234
8	NCR	Glyngå, re	425766198

## 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;

```

The screenshot shows the AWS Data Catalog console. On the left, there is a sidebar with a menu containing 'dim\_date', 'dim\_location', and 'fact\_atm\_trans'. The main area displays a table with the following columns: 'card\_type', 'total\_transaction\_count', 'inactive\_count', and 'inactive\_count\_percent'. The table contains 10 rows of data, sorted by 'total\_transaction\_count' in descending order. An 'Export' button is visible in the top right corner of the table view.

card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	916452	172000	18.7600
VISA	341656	61426	17.9700
Dankort - on-us	287626	49360	17.1600
CIRRIUS	34724	5906	17.0000
HÅfÅvekort - on-us	124974	20662	16.5300
Dankort	57162	9114	15.9400
MasterCard	801014	126964	15.8500
Visa Dankort - on-us	1497610	225944	15.0800
HÅfÅvekort	16918	2416	14.2800
Visa Dankort	855680	121094	14.1500

7. Top 10 records with the number of transactions ordered by the ATM\_number, ATM\_manufacturer, location, weekend\_flag and then total\_transaction\_count, on weekdays and on weekends throughout the year

```
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;
```

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÅfÅstved	0	50234
1	NCR	NÅfÅstved	1	17216
100	NCR	Intern Skive	0	32986
100	NCR	Intern Skive	1	2640
101	NCR	Bryggen Vejle	0	81516
101	NCR	Bryggen Vejle	1	26072
102	NCR	Aalborg Storcenter Afd	0	13602
102	NCR	Aalborg Storcenter Afd	1	4480
103	Diebold Nixdorf	Vejgaard	0	34952
103	Diebold Nixdorf	Vejgaard	1	7402

## 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;
```

aws

Services

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

[Alt+S]

N. Virginia

upgradthanunkumar @ 1754-6445-8291

dim\_date\_pkey

dim\_location\_pkey

fact\_atm\_trans\_pkey

dim\_atm

dim\_card\_type

dim\_date

dim\_location

fact\_atm\_trans

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query 1460

Execution

Data

Visualize

Completed, started on September 12, 2022 at 02:08:14

ELAPSED TIME: 00 m 02 s

Rows returned (1)

Export

Search rows

< 1 >

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejgaard	Friday	16476