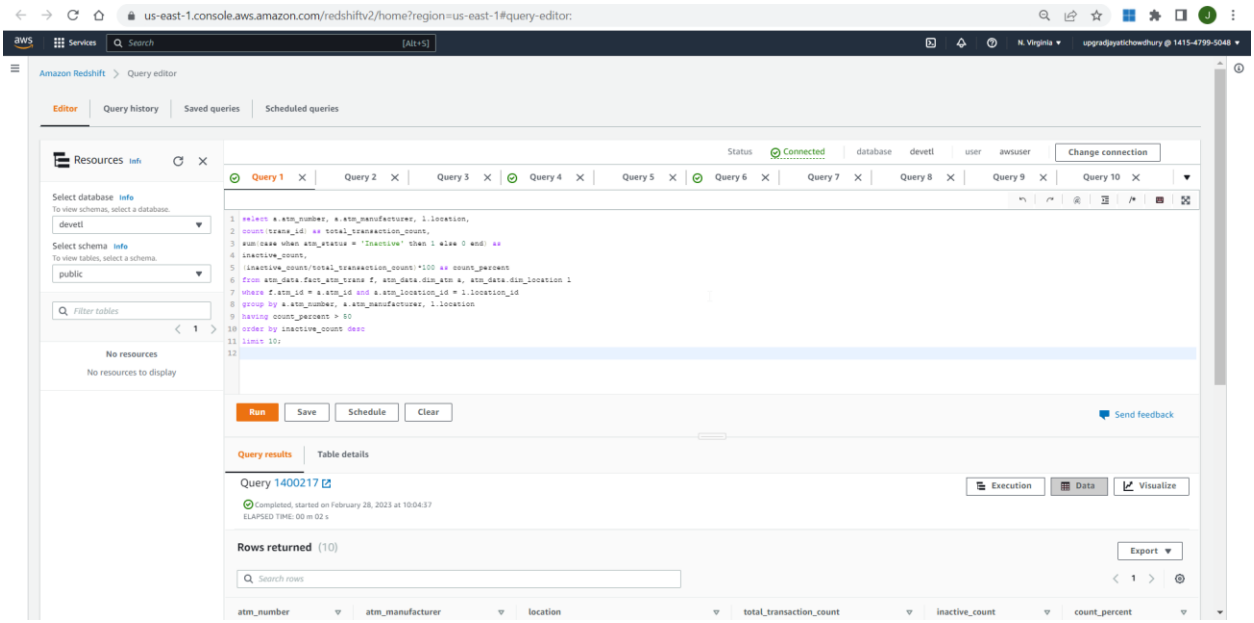
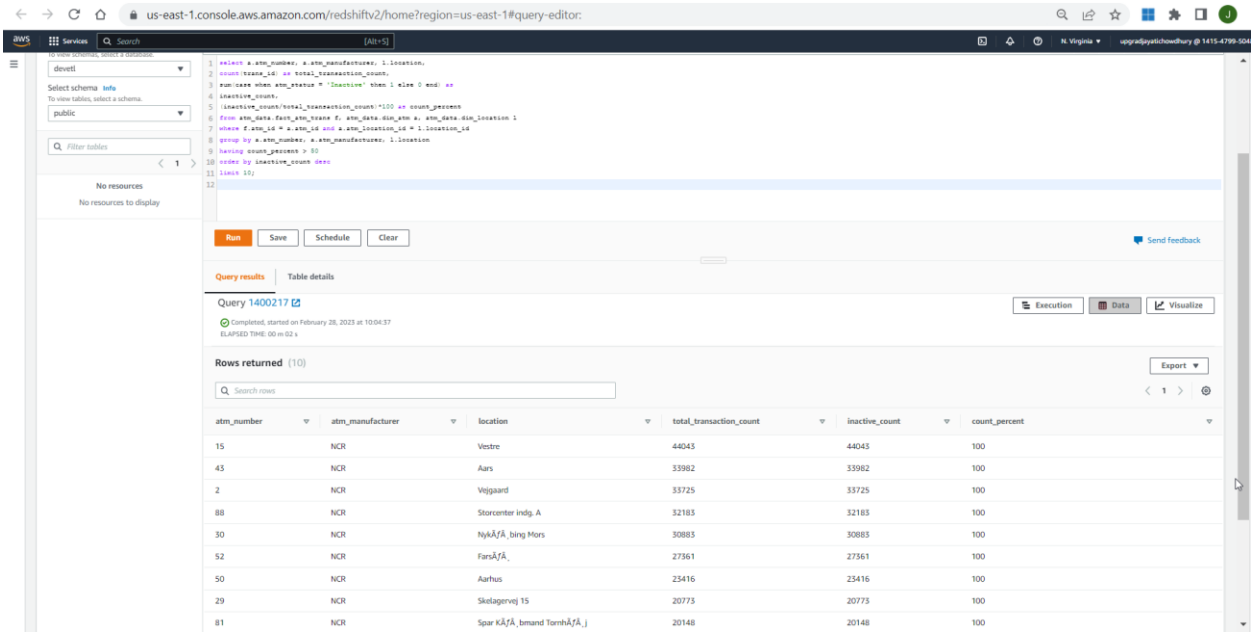
Solving analytical queries on Redshift Cluster

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**

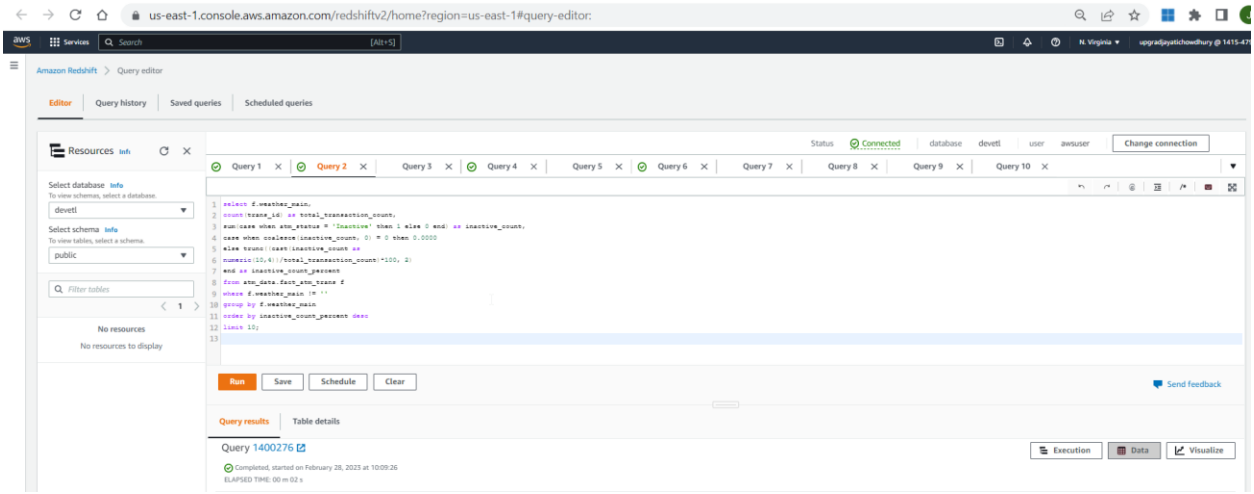
```
select a.atm_number, a.atm_manufacturer, l.location,
count(trans_id) as total_transaction_count,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as
inactive_count,
(inactive_count/total_transaction_count)*100 as count_percent
from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location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

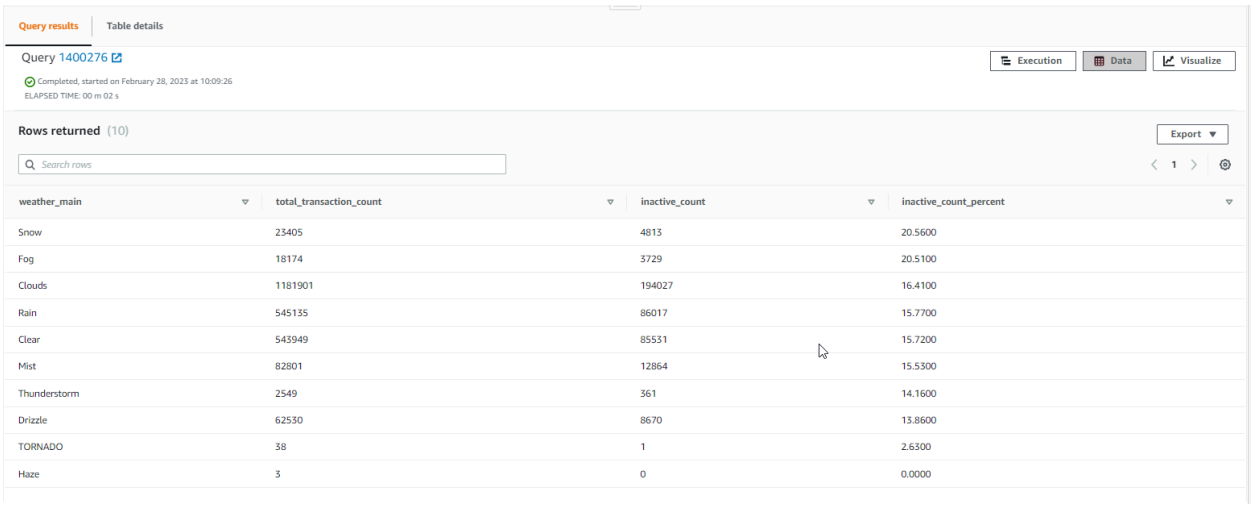



us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1#query-editor:
Services
Search
[Alt+S]
N. Virginia
devetl
Select schema Info
To view tables, select a schema.
public
Filter tables
No resources
No resources to display
select a.atm_number, a.atm_manufacturer, l.location,
count(trans_id) 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 > 80
order by inactive_count desc
limit 10;
Run
Save
Schedule
Clear
Send feedback
Query results
Table details
Query 1400217
Completed, started on February 28, 2023 at 10:04:57
ELAPSED TIME: 00 m 02 s
Execution
Data
Visualize
Rows returned (10)
Export
Search rows
atm_number
atm_manufacturer
location
total_transaction_count
inactive_count
count_percent
15 NCR Vestre 44043 44043 100
43 NCR Aars 33982 33982 100
2 NCR Vejgaard 33725 33725 100
88 NCR Storcenter indg. A 32183 32183 100
30 NCR NykÃƒÂ¸bing Mors 30883 30883 100
52 NCR FarsÃƒÂ¸ 27361 27361 100
50 NCR Aarhus 23416 23416 100
29 NCR Skelagervej 15 20773 20773 100
81 NCR Spar KÃƒÂ¸bmand TornhÃƒÂ¸j 20148 20148 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;
```





Query results | Table details

Query 1400276

Completed, started on February 28, 2023 at 10:09:26
ELAPSED TIME: 00 m 02 s

Execution | Data | Visualize

Rows returned (10)

Export

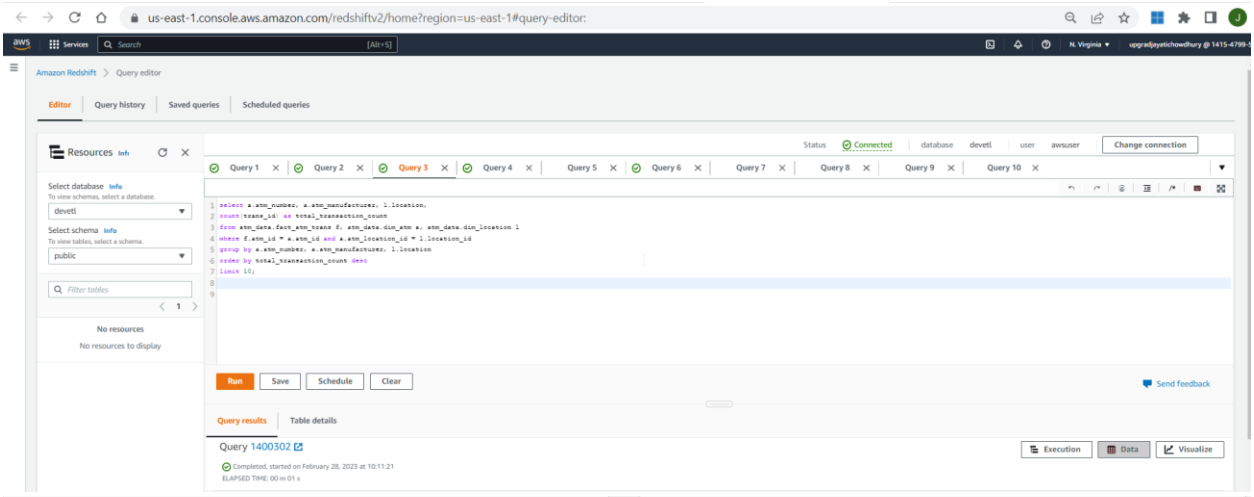
Search rows

1

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;
```



Query results | Table details

Query 1400302

Completed, started on February 28, 2023 at 10:11:21
ELAPSED TIME: 00 m 01 s

Execution | Data | Visualize

Rows returned (10)

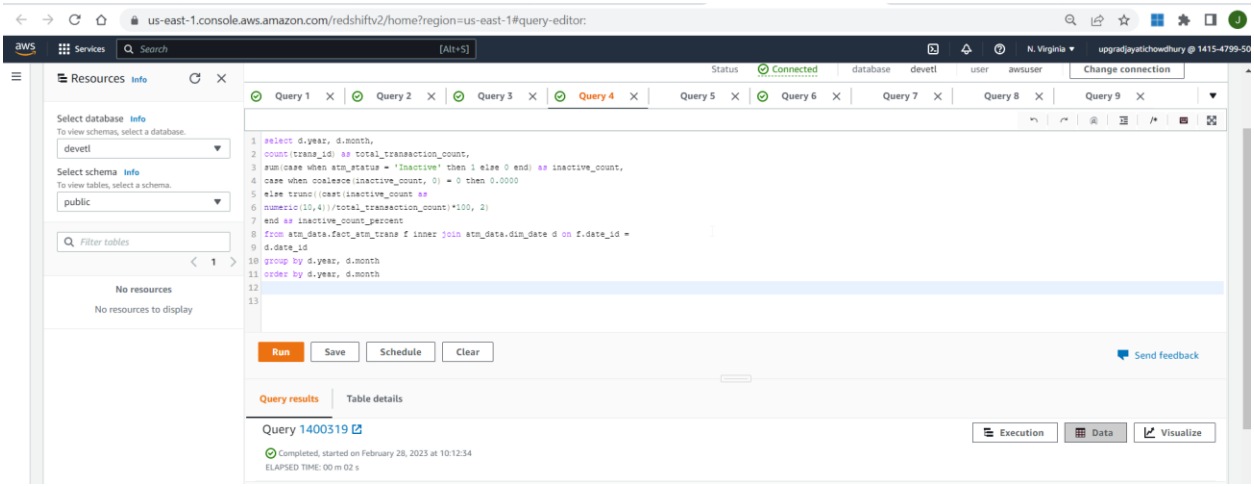
Export

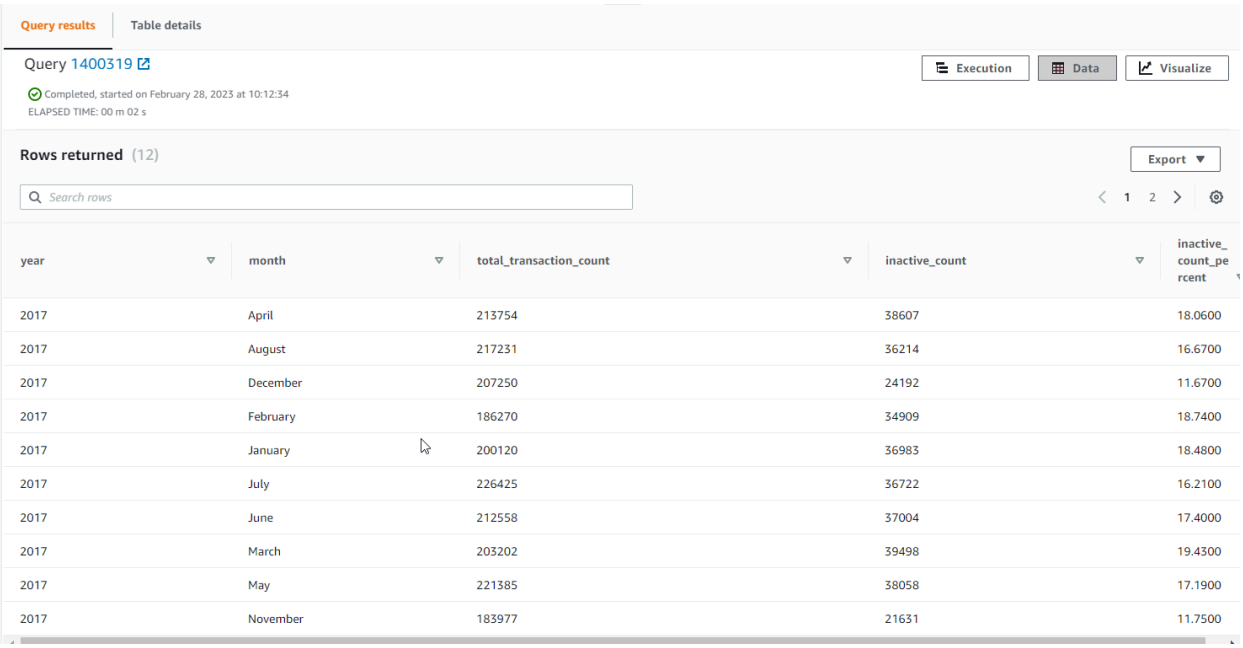
Search rows

atm_number	atm_manufacturer	location	total_transaction_count
31	NCR	Slagelse	83320
39	NCR	Svenstrup	55380
12	NCR	ÃƒÉœsterÃƒÂ¥ Duus	54211
24	NCR	Hobro	53378
34	NCR	Skipperen	53198
15	NCR	Vestre	44043
30	NCR	NykÃƒÂ¸bing Mors Lobby	43767
1	NCR	NÃƒÂ¦stved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÃƒÂ¸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
```

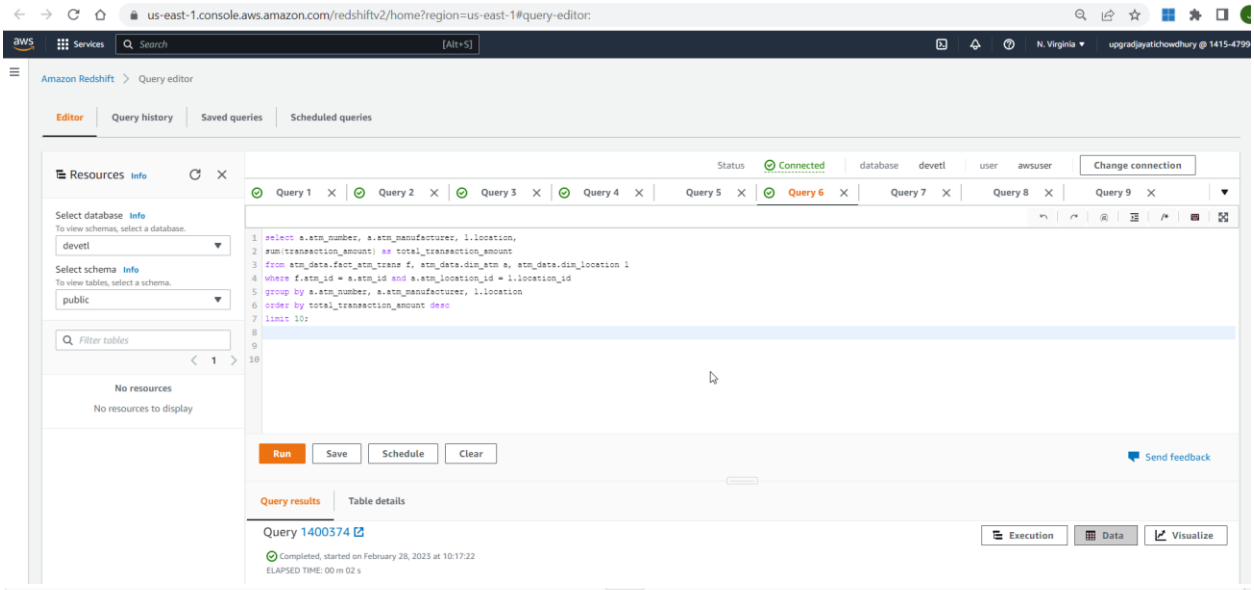


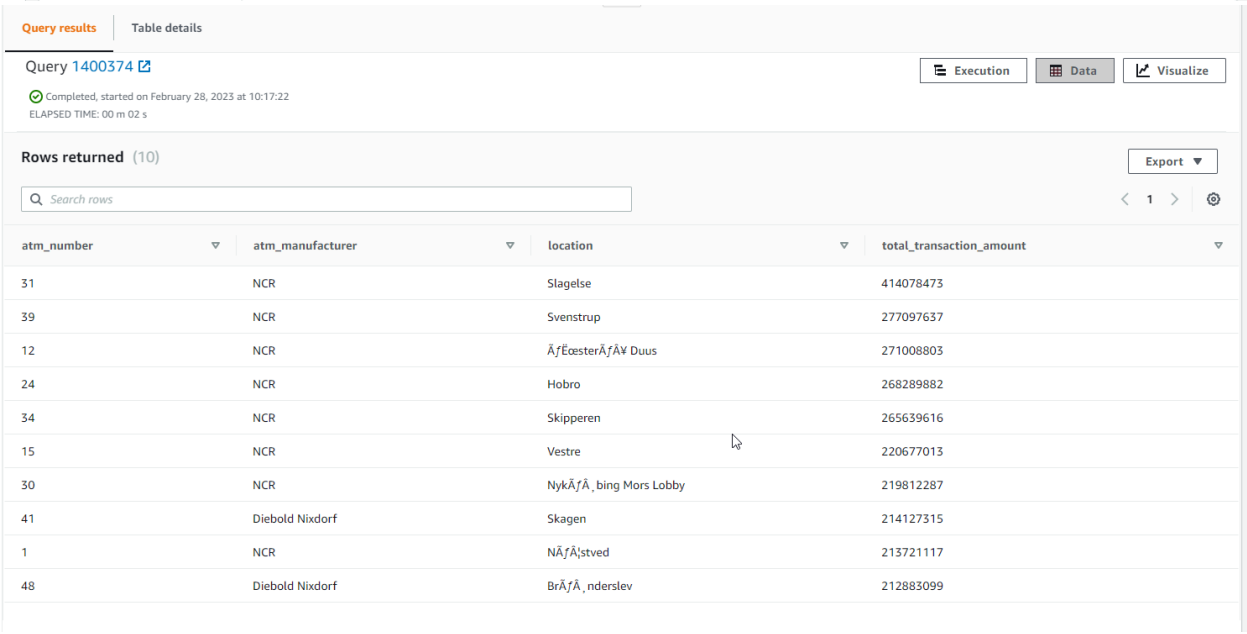


year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	213754	38607	18.0600
2017	August	217231	36214	16.6700
2017	December	207250	24192	11.6700
2017	February	186270	34909	18.7400
2017	January	200120	36983	18.4800
2017	July	226425	36722	16.2100
2017	June	212558	37004	17.4000
2017	March	203202	39498	19.4300
2017	May	221385	38058	17.1900
2017	November	183977	21631	11.7500

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;
```

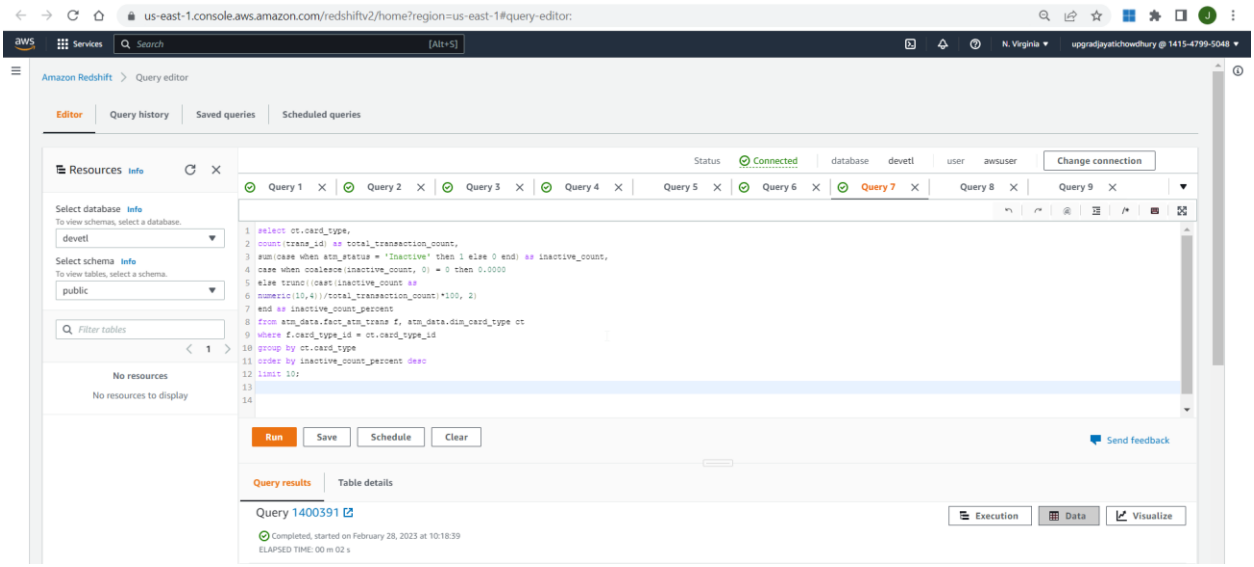


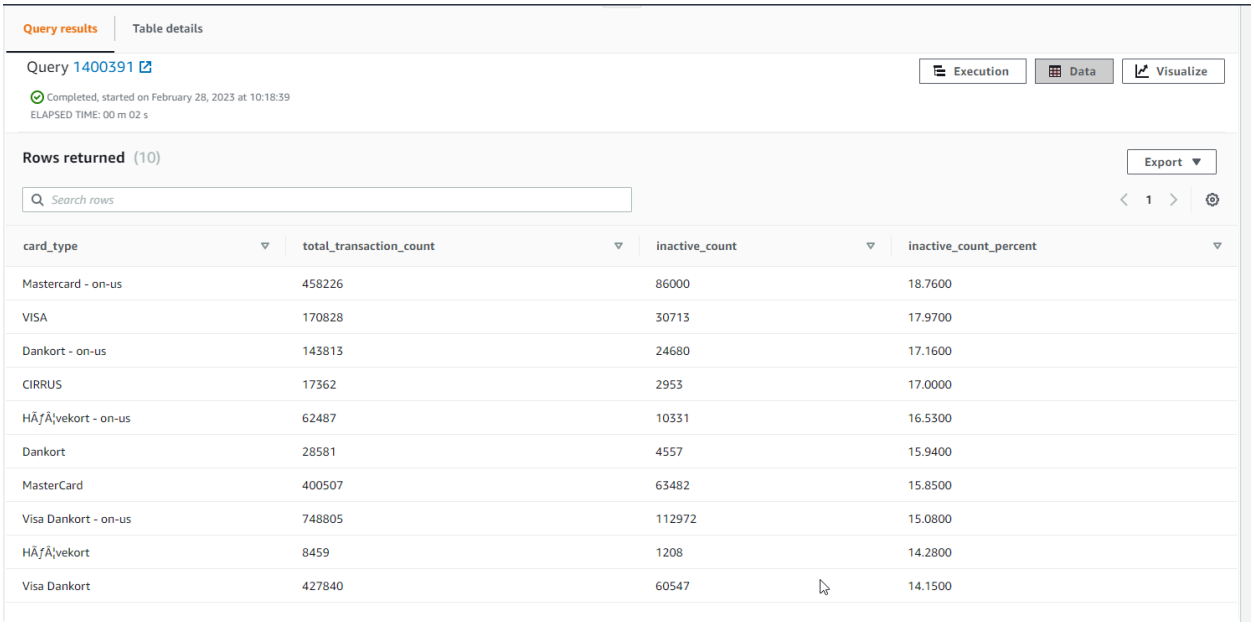


atm_number	atm_manufacturer	location	total_transaction_amount
31	NCR	Slagelse	414078473
39	NCR	Svenstrup	277097637
12	NCR	ÃƒËœsterÃƒÂ¥ Duus	271008803
24	NCR	Hobro	268289882
34	NCR	Skipperen	265639616
15	NCR	Vestre	220677013
30	NCR	NykÃƒÂ¸bing Mors Lobby	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	NÃƒÂ¦stved	213721117
48	Diebold Nixdorf	BrÃƒÂ¸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;
```





Query results | Table details

Query 1400391

Execution | Data | Visualize

Completed, started on February 28, 2023 at 10:18:39
ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export

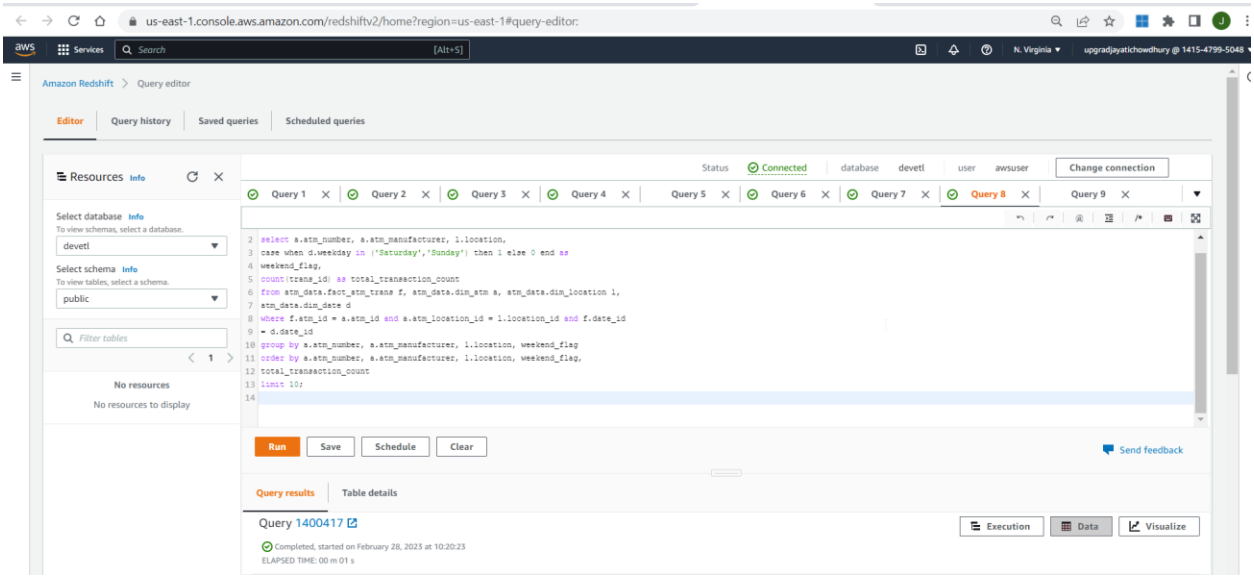
Search rows

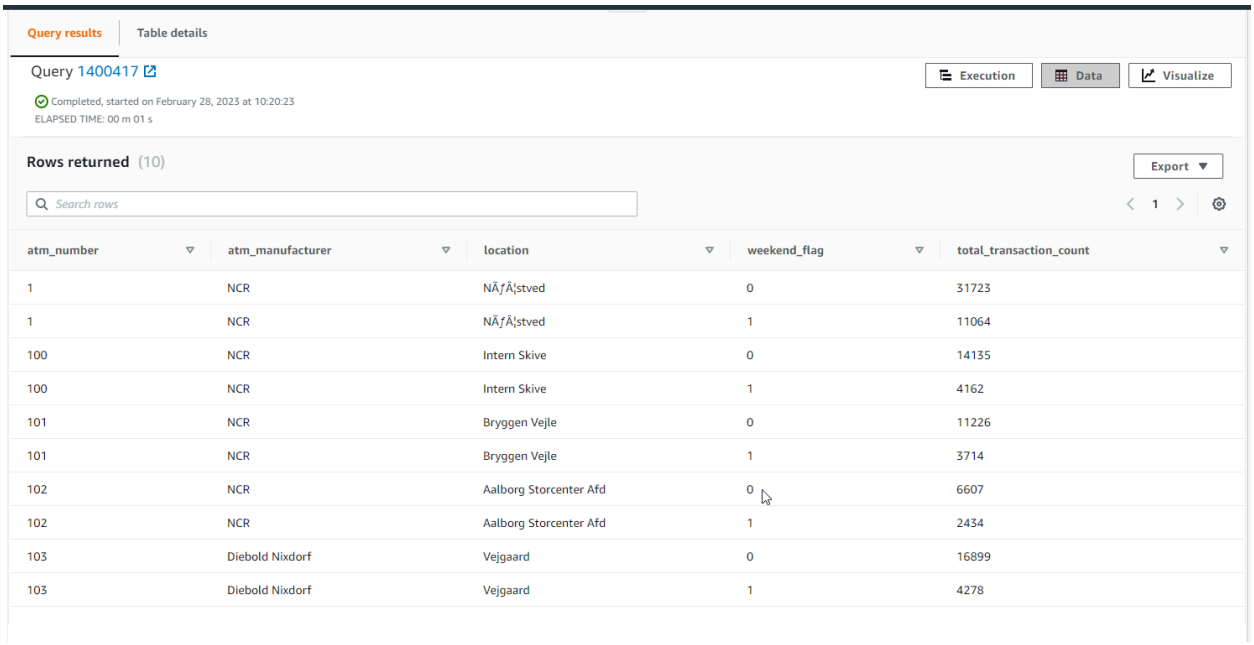
1

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
HÃƒÂ¦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
HÃƒÂ¦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;
```





Query results | Table details

Query 1400417

Execution | Data | Visualize

Completed, started on February 28, 2023 at 10:20:23
ELAPSED TIME: 00 m 01 s

Rows returned (10)

Export

Search rows

1

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃƒÂ¦stved	0	31723
1	NCR	NÃƒÂ¦stved	1	11064
100	NCR	Intern Skive	0	14135
100	NCR	Intern Skive	1	4162
101	NCR	Bryggen Vejle	0	11226
101	NCR	Bryggen Vejle	1	3714
102	NCR	Aalborg Storcenter Afd	0	6607
102	NCR	Aalborg Storcenter Afd	1	2434
103	Diebold Nixdorf	Vejgaard	0	16899
103	Diebold Nixdorf	Vejgaard	1	4278

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;
```