

Solving analytical queries on RedShift Cluster

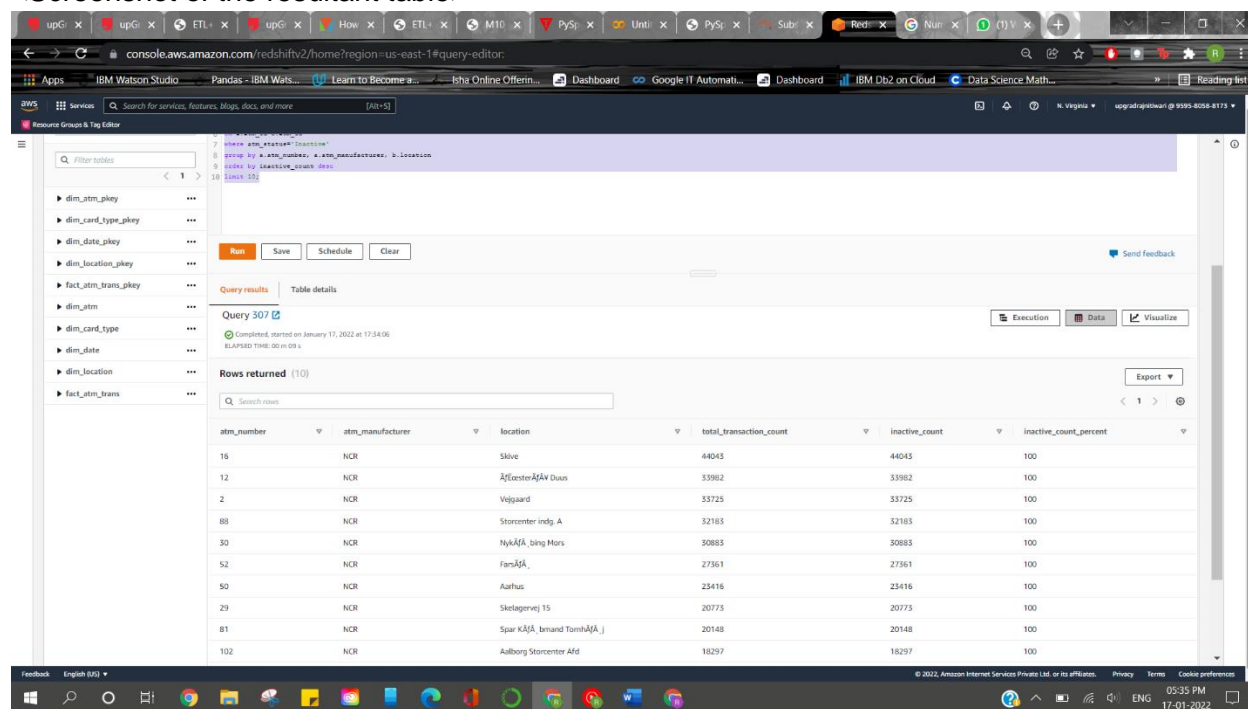
Here, you have to write the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS RedShift Query editor UI.

1. Top 10 ATMs where most transactions are in the 'inactive' state

<Query>

```
select a.atm_number, a.atm_manufacturer, b.location, count(c.trans_id) as
total_transaction_count, count(c.atm_status) as
inactive_count, (inactive_count*100.0/total_transaction_count) as inactive_count_percent
from atm_trans.dim_atm a
left join atm_trans.dim_location b
on a.atm_location_id=b.location_id
right join atm_trans.fact_atm_trans c
on a.atm_id=c.atm_id
where atm_status='Inactive'
group by a.atm_number, a.atm_manufacturer, b.location, c.atm_status
order by inactive_count desc
limit 10;
```

<Screenshot of the resultant table>



The screenshot shows the AWS RedShift Query Editor interface. The query has been executed successfully, and the results are displayed in a table. The table has 7 columns: atm_number, atm_manufacturer, location, total_transaction_count, inactive_count, and inactive_count_percent. The results are ordered by inactive_count in descending order, showing the top 10 ATMs.

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	inactive_count_percent
16	NCR	Slake	44043	44043	100
12	NCR	ÅlcentorÅÅÅV Ous	33982	33982	100
2	NCR	Vijgaard	33725	33725	100
88	NCR	Storcenter indy A	32183	32183	100
30	NCR	NyÅÅÅÅ Åing Mars	30883	30883	100
52	NCR	FamÅÅÅ	27361	27361	100
50	NCR	Aarhuc	23416	23416	100
29	NCR	Skelagorvej 15	20773	20773	100
81	NCR	Spar KÅÅÅ Årmand TomHÅÅÅ	20148	20148	100
102	NCR	Aalborg Storcenter ÅÅÅ	18297	18297	100

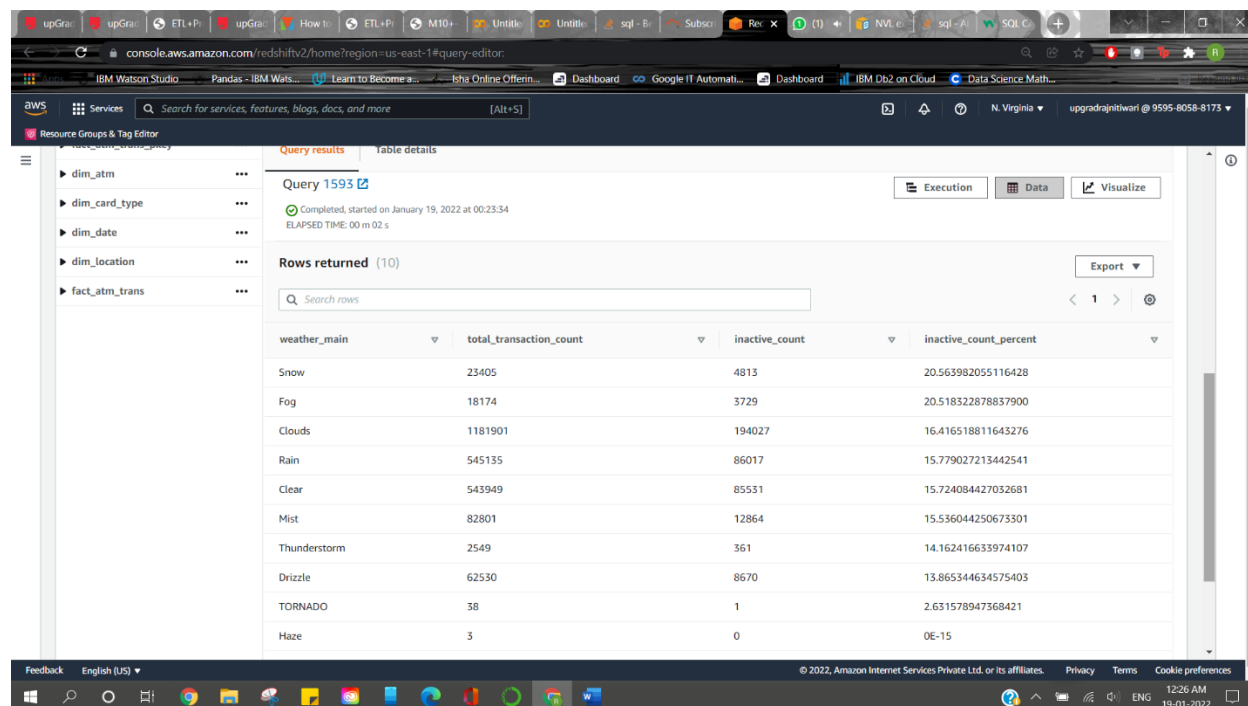
2. Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions

<Query>

```
select a.weather_main, a.total_transaction_count, case when c.inactive_count is null then 0 else
c.inactive_count end as inactive_count, case when (inactive_count*100.0 /
total_transaction_count) is null then 0 else (inactive_count * 100.0 / total_transaction_count)
end as inactive_count_percent
```

```
from (select b.weather_main, count(b.trans_id) as total_transaction_count
      from atm_trans.fact_atm_trans as b
      where b.weather_main <> "
      group by b.weather_main) a
left join (select b.weather_main, count(b.atm_status) as inactive_count
          from atm_trans.fact_atm_trans as b
          where b.weather_main <> " and b.atm_status='Inactive'
          group by b.weather_main) c
on a.weather_main=c.weather_main
order by inactive_count_percent desc;
```

<Screenshot of the resultant table>



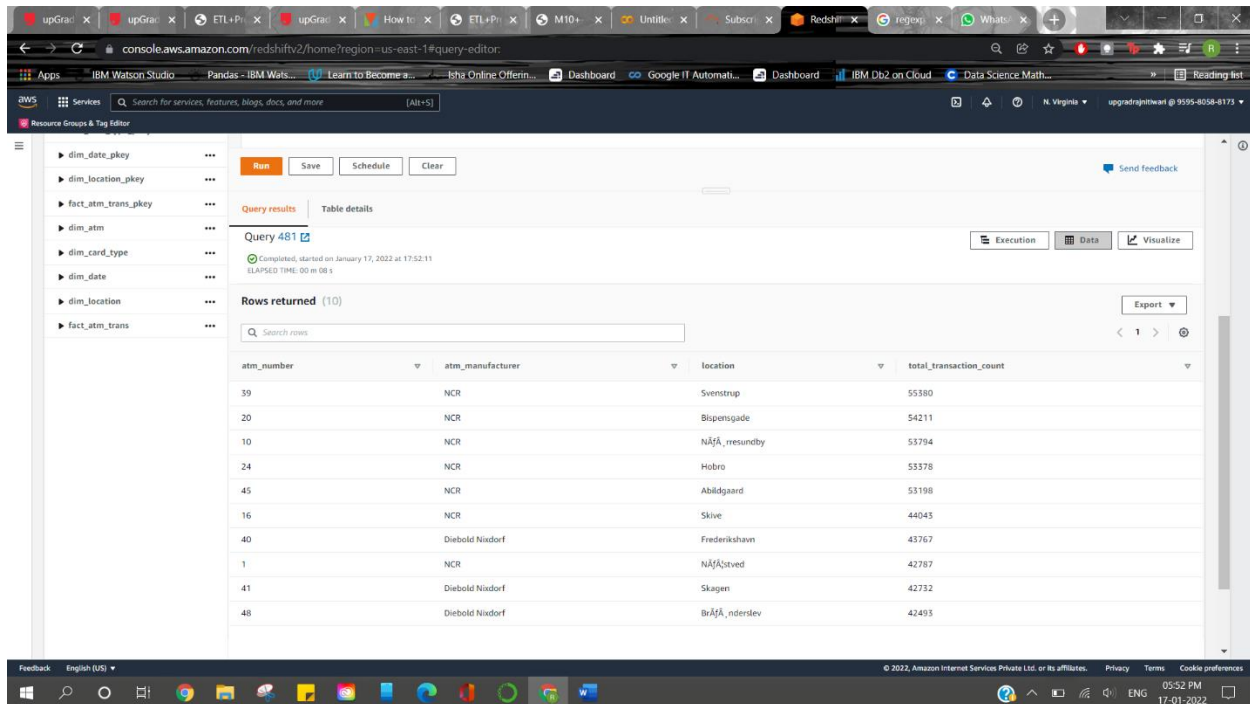
weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.563982055116428
Fog	18174	3729	20.518322878837900
Clouds	1181901	194027	16.416518811643276
Rain	545135	86017	15.779027213442541
Clear	543949	85531	15.724084427032681
Mist	82801	12864	15.536044250673301
Thunderstorm	2549	361	14.162416633974107
Drizzle	62530	8670	13.865344634575403
TORNADO	38	1	2.631578947368421
Haze	3	0	0E-15

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

<Query>

```
select a.atm_number, a.atm_manufacturer, b.location, count(c.trans_id) as
total_transaction_count
from atm_trans.dim_atm a
left join atm_trans.dim_location b
on a.atm_location_id=b.location_id
right join atm_trans.fact_atm_trans c
on a.atm_id=c.atm_id
group by a.atm_number, a.atm_manufacturer, b.location
order by total_transaction_count desc
limit 10;
```

<Screenshot of the resultant table>



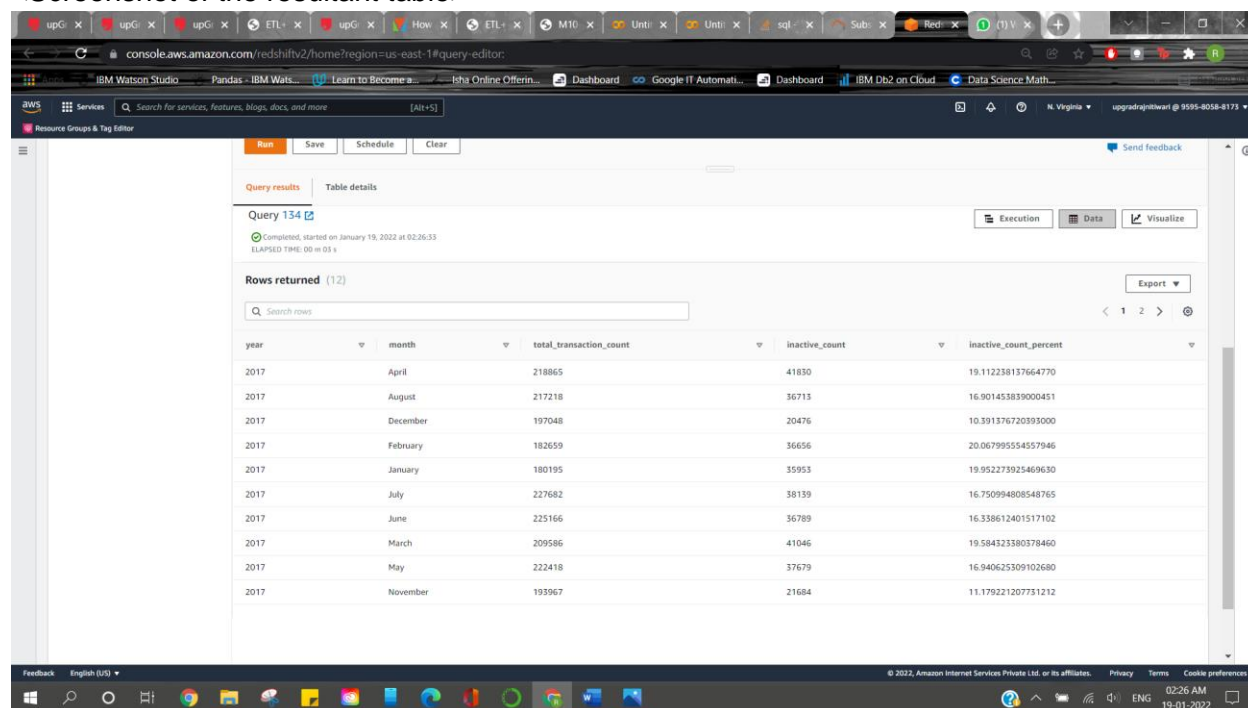
atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Blipensgade	54211
10	NCR	NÅgå, resundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44045
40	Diebold Niendorf	Frederikshavn	43767
1	NCR	NÅgåstved	42787
41	Diebold Niendorf	Skagen	42732
48	Diebold Niendorf	Brågå, nderlev	42493

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

<Query>

```
select dt1.year, dt1.month, dt1.total_transaction_count, dt2.inactive_count,
(dt2.inactive_count*100.0/dt1.total_transaction_count) as inactive_count_percent
from (select a.year, a.month, count(b.trans_id) as total_transaction_count
      from atm_trans.dim_date a right join atm_trans.fact_atm_trans b on a.date_id=b.date_id
      group by a.month, a.year) dt1 join
(select a.year, a.month, count(b.atm_status) as inactive_count
      from atm_trans.dim_date a right join atm_trans.fact_atm_trans b on a.date_id=b.date_id
      where atm_status='Inactive'
      group by a.month, a.year) dt2 on dt1.month=dt2.month and dt1.year=dt2.year
order by dt1.month;
```

<Screenshot of the resultant table>

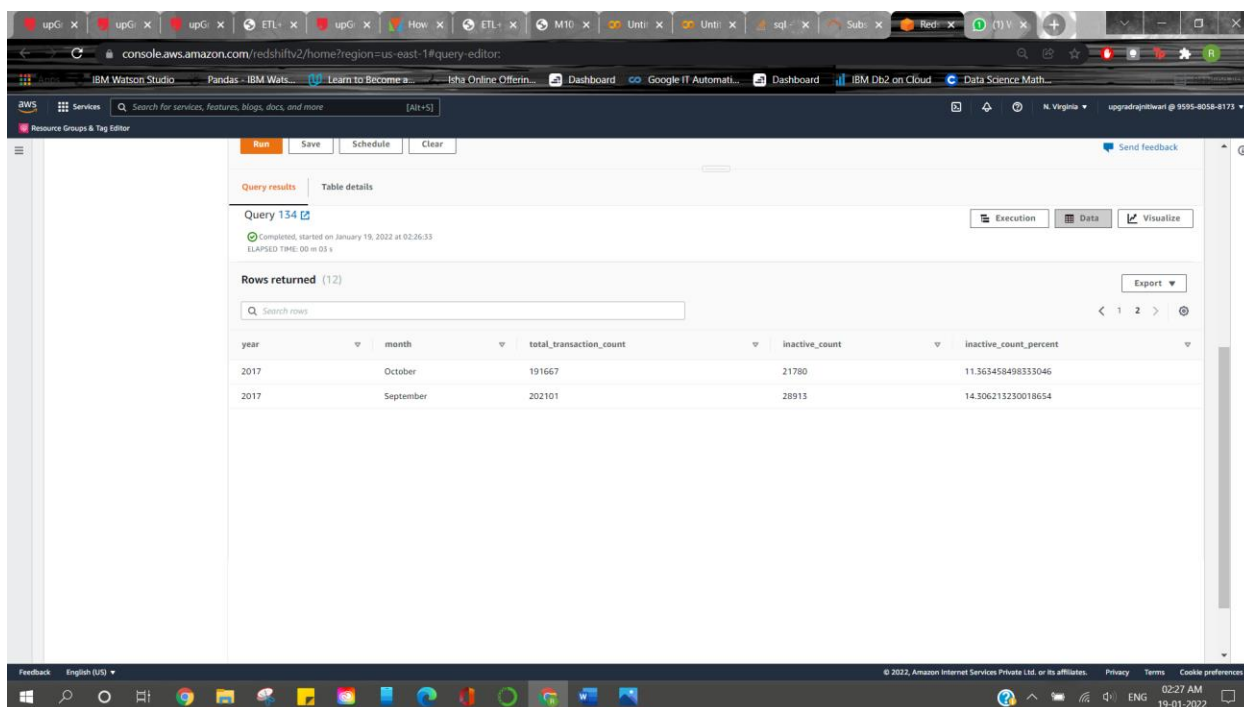


Query 134

Completed, started on January 19, 2022 at 02:26:33
ELAPSED TIME: 00 m 03 s

Rows returned (12)

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.112238137664770
2017	August	217218	36713	16.901453839000451
2017	December	197048	20476	10.391376720395000
2017	February	182659	36656	20.067995554557946
2017	January	180195	35953	19.952273925469630
2017	July	227682	38139	16.750994808548765
2017	June	225166	36789	16.338612401517102
2017	March	209586	41046	19.584323380378460
2017	May	222418	37679	16.940625309102680
2017	November	193967	21684	11.179221207731212



Query 134

Completed, started on January 19, 2022 at 02:26:33
ELAPSED TIME: 00 m 03 s

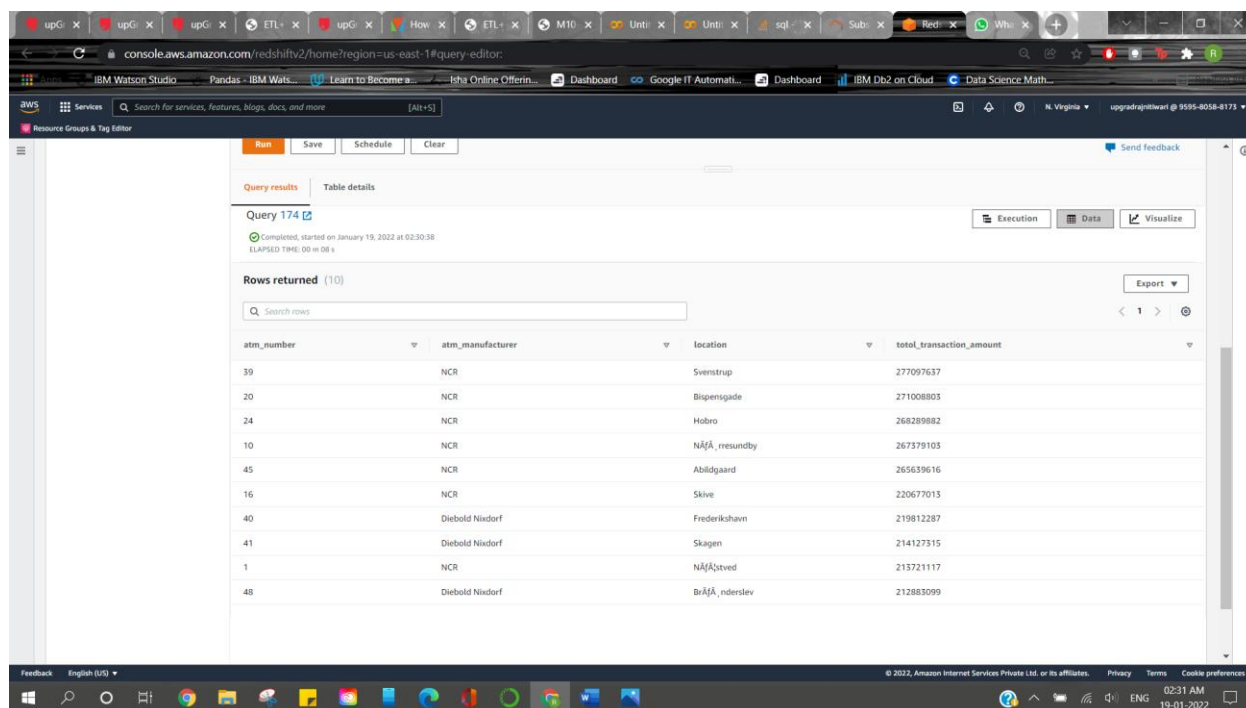
Rows returned (12)

year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	October	191667	21780	11.363458498333046
2017	September	202101	28913	14.306213230018654

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

```
select a.atm_number, a.atm_manufacturer, b.location, sum(c.transaction_amount) as
total_transaction_amount
from atm_trans.dim_atm a join atm_trans.dim_location b
on a.atm_location_id=b.location_id left join atm_trans.fact_atm_trans c
on a.atm_id=c.atm_id
group by a.atm_number, a.atm_manufacturer, b.location
order by total_transaction_amount desc
limit 10;
```

<Screenshot of the resultant table>



The screenshot shows the AWS Redshift console interface. The query results are displayed in a table with the following columns: atm_number, atm_manufacturer, location, and total_transaction_amount. The results are sorted in descending order of total_transaction_amount.

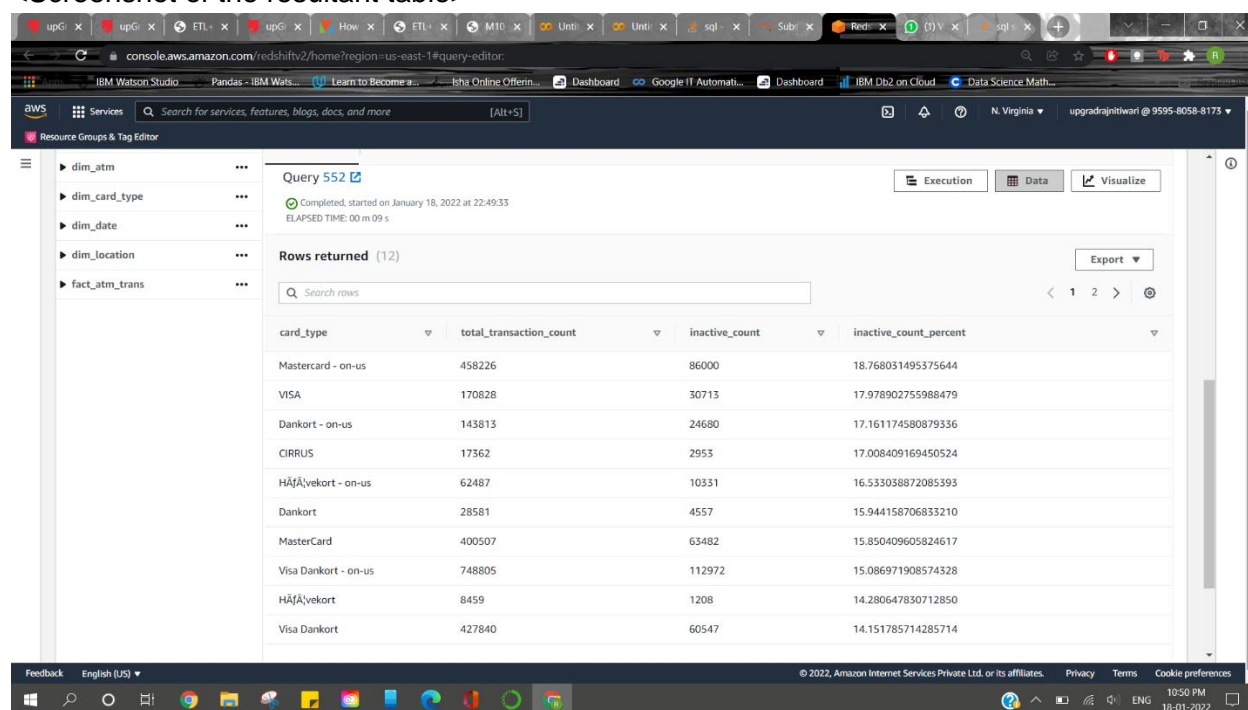
atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Blipensgade	271008803
24	NCR	Hubro	268289882
10	NCR	Nalå, resundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	Nalåstved	213721117
48	Diebold Nixdorf	Brålå, nderslev	212883099

6. Number of failed ATM transactions across various card types

<Query>

```
select dt1.card_type, dt1.total_transaction_count, dt2.inactive_count,
(dt2.inactive_count*100.0/dt1.total_transaction_count) as inactive_count_percent
from (select a.card_type, count(b.trans_id) as total_transaction_count
      from atm_trans.dim_card_type a join atm_trans.fact_atm_trans b on
a.card_type_id=b.card_type_id
      group by a.card_type) dt1 join
(select a.card_type, count(b.atm_status) as inactive_count
      from atm_trans.dim_card_type a join atm_trans.fact_atm_trans b on
a.card_type_id=b.card_type_id
      where b.atm_status='Inactive'
      group by a.card_type) dt2 on dt1.card_type=dt2.card_type
order by inactive_count_percent desc;
```

<Screenshot of the resultant table>



The screenshot shows the AWS Redshift console interface. On the left, there is a sidebar with a tree view containing folders for 'dim_atm', 'dim_card_type', 'dim_date', 'dim_location', and 'fact_atm_trans'. The main area displays the results for 'Query 552', which was completed on January 18, 2022, at 22:49:33. The query returned 12 rows. The table has four columns: 'card_type', 'total_transaction_count', 'inactive_count', and 'inactive_count_percent'. The data is sorted by 'inactive_count_percent' in descending order.

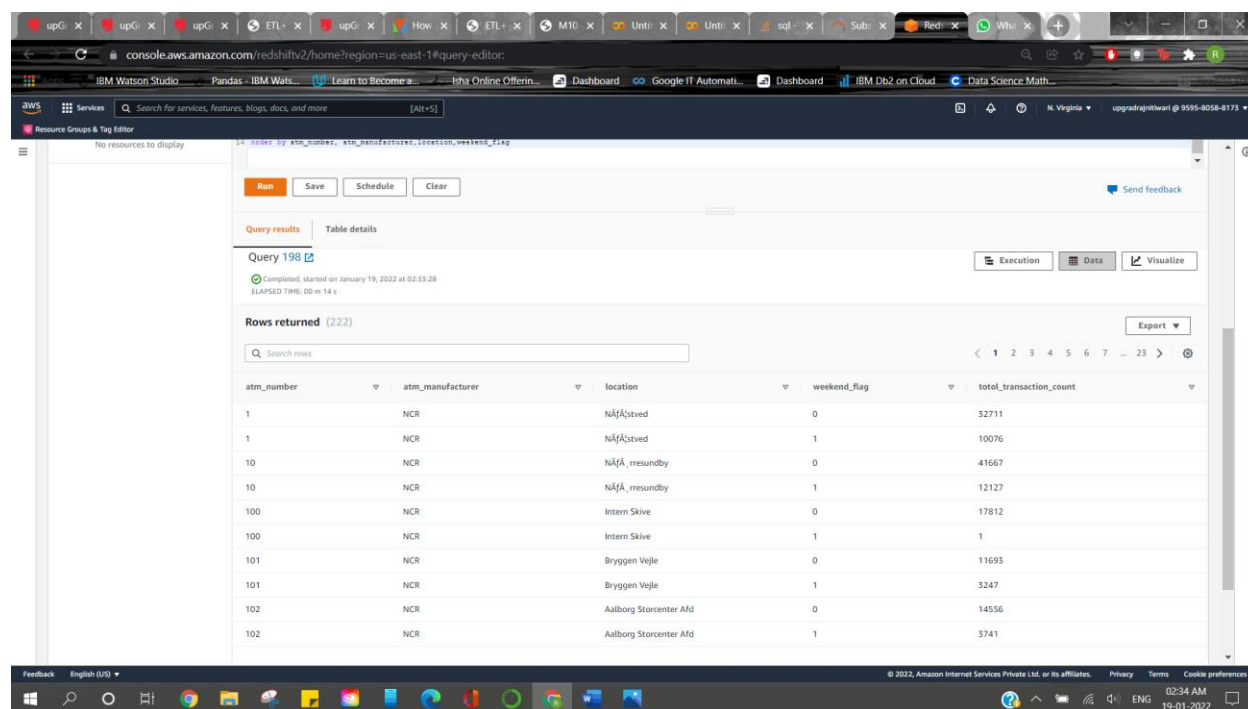
card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.768031495375644
VISA	170828	30713	17.978902755988479
Dankort - on-us	143813	24680	17.161174580879336
CIRRIUS	17362	2953	17.008409169450524
H&A\vekort - on-us	62487	10351	16.533038872085393
Dankort	28581	4557	15.944158706833210
MasterCard	400507	63482	15.850409605824617
Visa Dankort - on-us	748805	112972	15.086971908574328
H&A\vekort	8459	1208	14.280647830712850
Visa Dankort	427840	60547	14.151785714285714

- 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

<Query>

```
select atm_number, atm_manufacturer,location,weekend_flag ,sum(totol_transaction_count) as
totol_transaction_count from (
select a.atm_number, a.atm_manufacturer, b.location,
CASE
    WHEN weekday ='Sunday' THEN 1
    WHEN weekday ='Saturday' THEN 1
    ELSE 0
END AS weekend_flag,
count(c.trans_id)as totol_transaction_count
from atm_trans.dim_atm a join atm_trans.dim_location b
on a.atm_location_id=b.location_id join atm_trans.fact_atm_trans c
on a.atm_id=c.atm_id join atm_trans.dim_date d on d.date_id = c.date_id
group by a.atm_number, a.atm_manufacturer, b.location,d.weekday)
group by atm_number, atm_manufacturer,location,weekend_flag
order by atm_number, atm_manufacturer,location,weekend_flag
```

<Screenshot of the resultant table>



atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÅfÅstved	0	32711
1	NCR	NÅfÅstved	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 Åld	0	14556
102	NCR	Aalborg Storcenter Åld	1	3741

8. Most active day in each ATMs from location "Vejgaard"

<Query>

drop view if exists atm_trans.weekday_trans;

create view atm_trans.weekday_trans as

select dd.weekday, count(*)

from atm_trans.fact_atm_trans tf

join atm_trans.dim_date dd on dd.date_id=tf.date_id

join atm_trans.dim_atm da on tf.atm_id=da.atm_id

join atm_trans.dim_location dl on dl.location_id=da.atm_location_id

where dl.location='Vejgaard' group by dd.weekday;

select da.atm_id, da.atm_manufacturer, dd.weekday, dl.location, count(*) as

total_transaction_count

from atm_trans.fact_atm_trans tf

join atm_trans.dim_date dd on dd.date_id=tf.date_id

join atm_trans.dim_atm da on tf.atm_id=da.atm_id

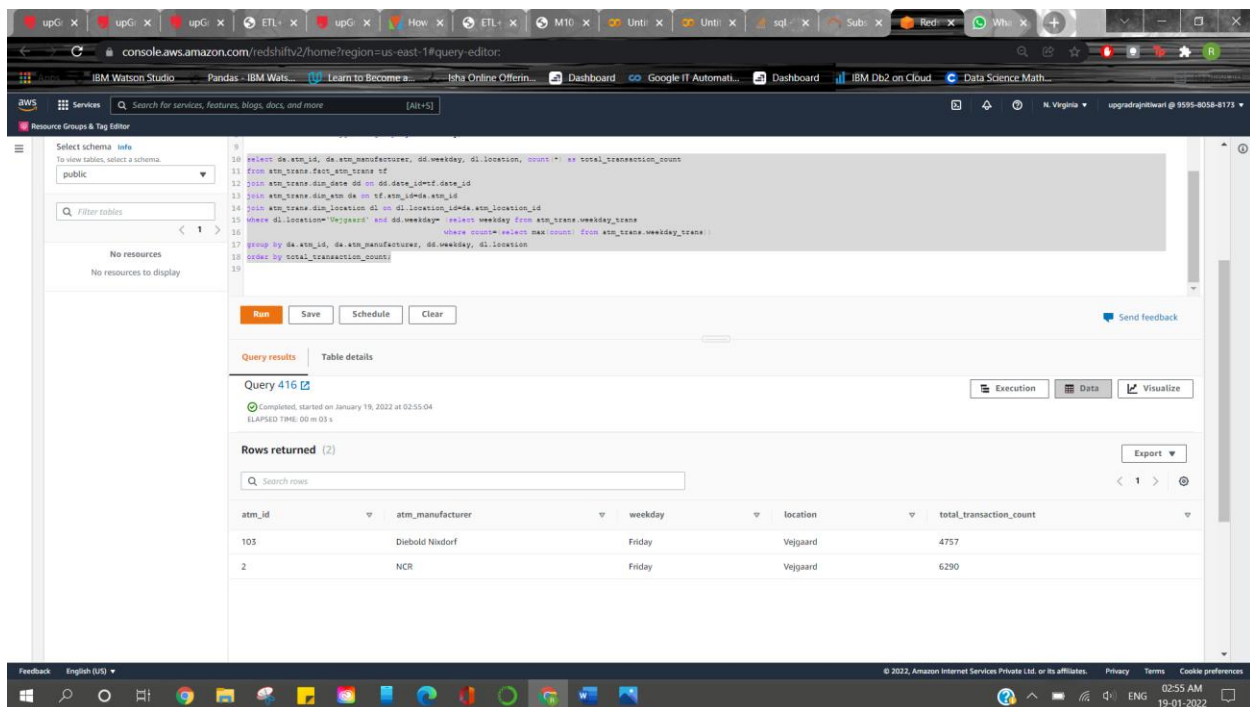
join atm_trans.dim_location dl on dl.location_id=da.atm_location_id

where dl.location='Vejgaard' and dd.weekday= (select weekday from atm_trans.weekday_trans
where count=(select max(count) from atm_trans.weekday_trans))

group by da.atm_id, da.atm_manufacturer, dd.weekday, dl.location

order by total_transaction_count;

<Screenshot of the resultant table>



The screenshot shows the AWS Redshift console interface. The query editor displays the SQL query used to find the most active day in each ATM from the location "Vejgaard". The query results are shown in a table with 5 columns: atm_id, atm_manufacturer, weekday, location, and total_transaction_count. The results show two rows: one for ATM 103 (Diebold Nindorf) on Friday with 4757 transactions, and one for ATM 2 (NCR) on Friday with 6290 transactions.

atm_id	atm_manufacturer	weekday	location	total_transaction_count
103	Diebold Nindorf	Friday	Vejgaard	4757
2	NCR	Friday	Vejgaard	6290