Solving analytical queries on RedShift Cluster

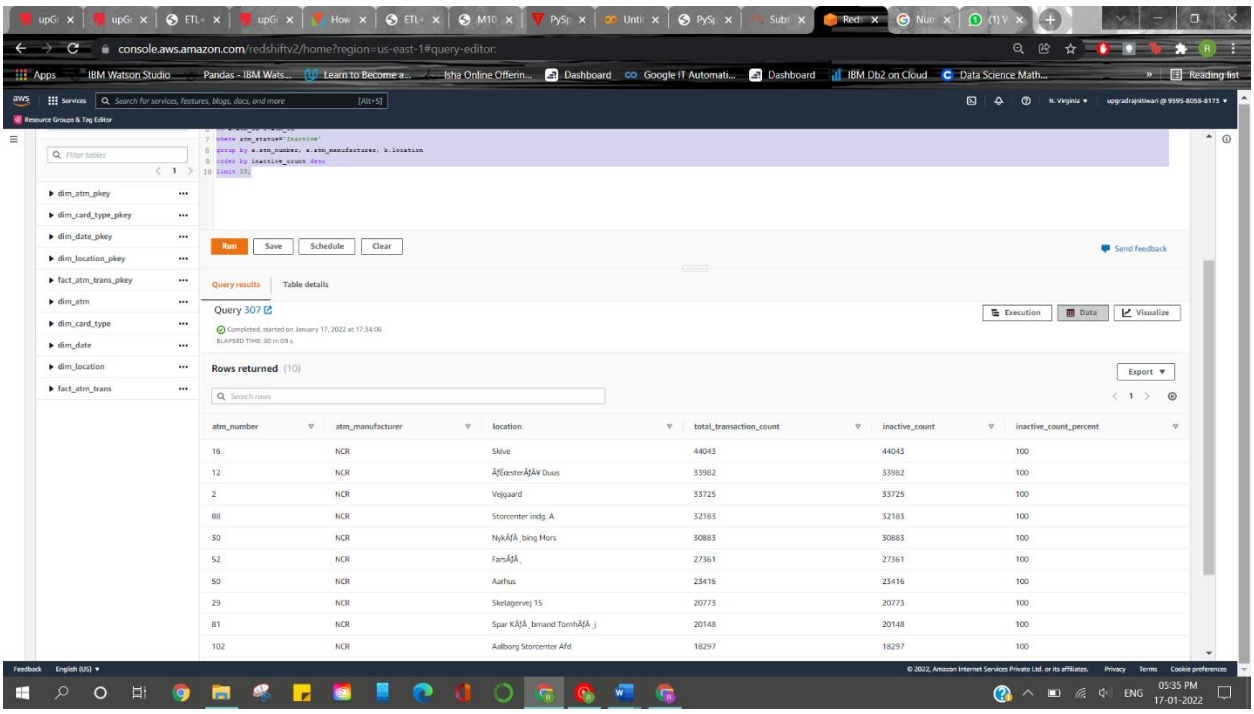
Here, you have to write the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS RedShift Query editor UI.

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>



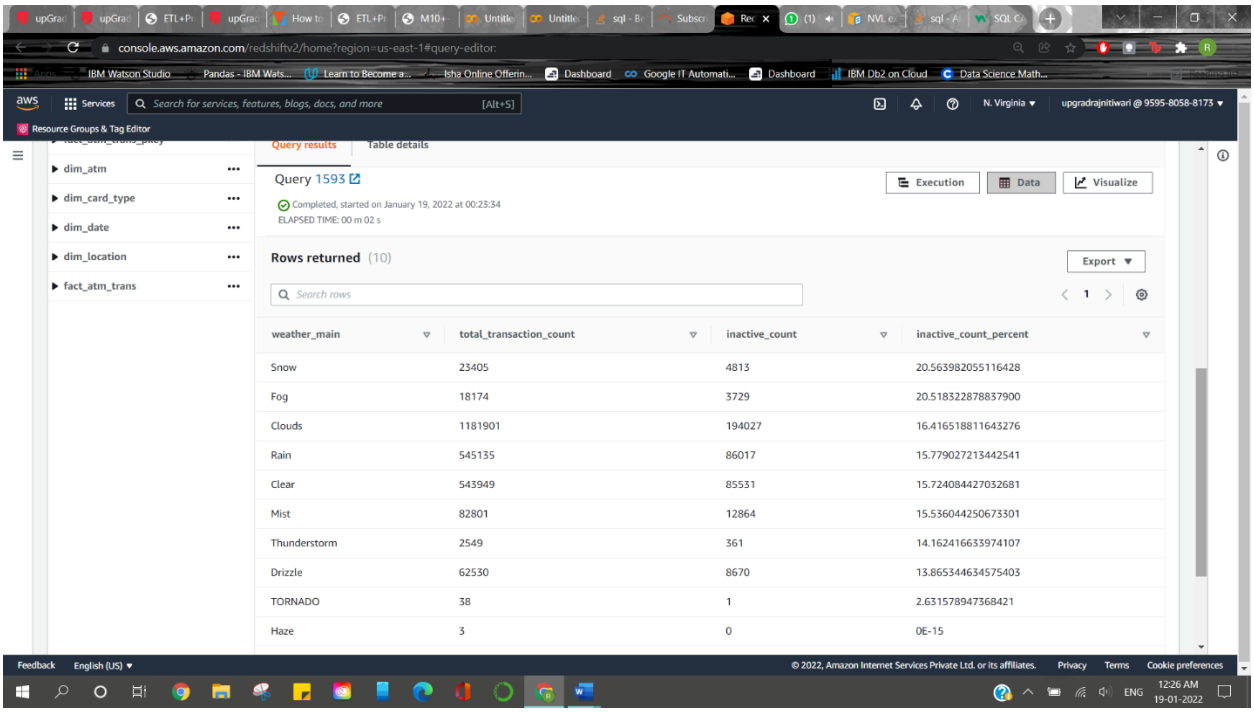
2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

<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>

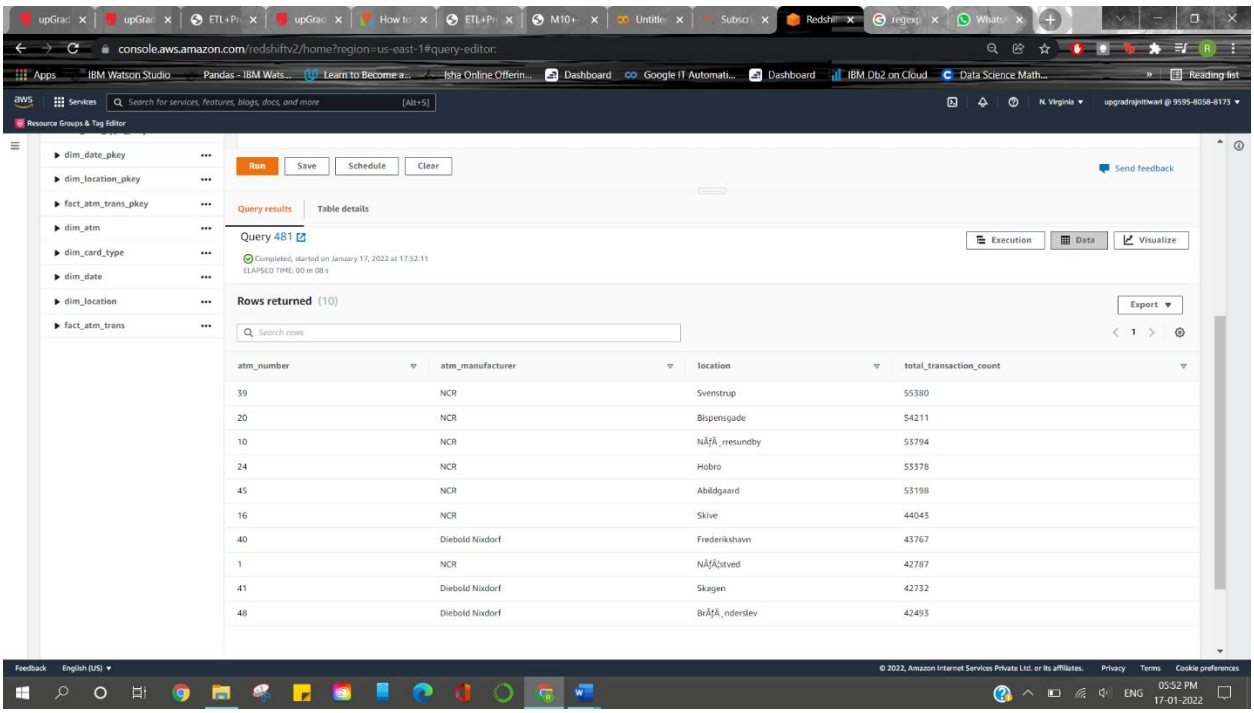


3. Top 10 ATMs with the most number of transactions throughout the year

<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>

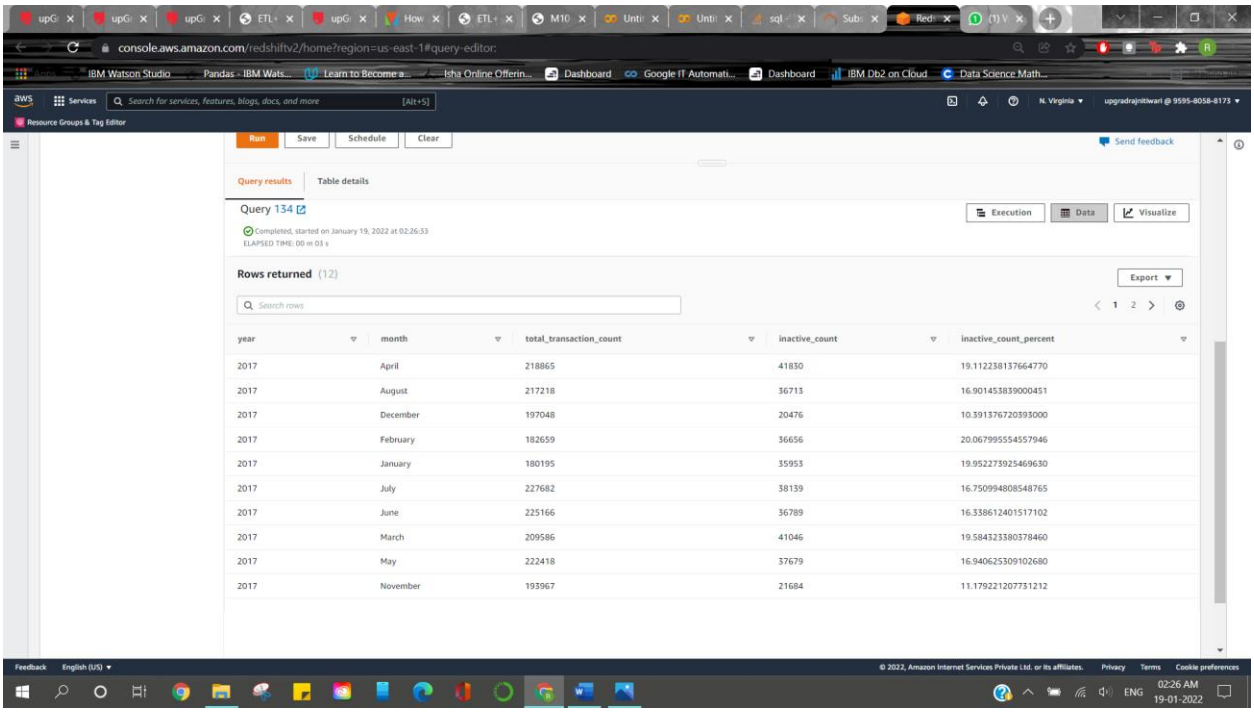


4. Number of overall ATM transactions going inactive per month for each month

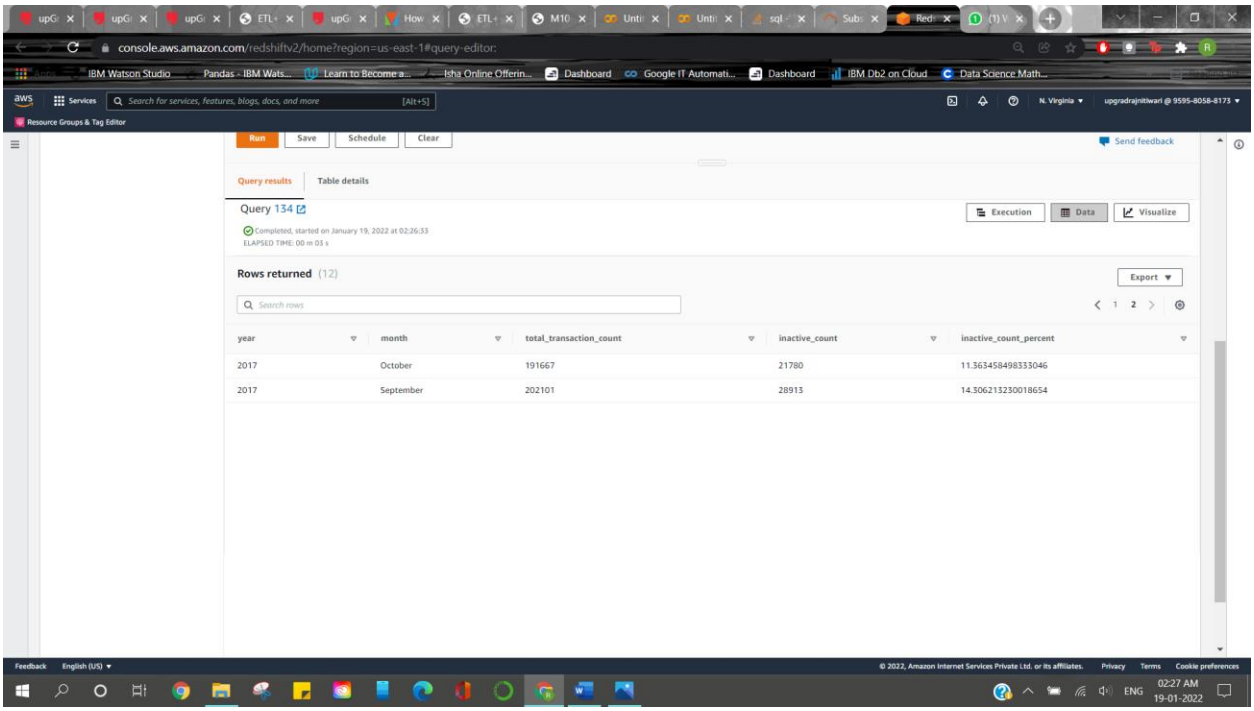
<Query>

```
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>



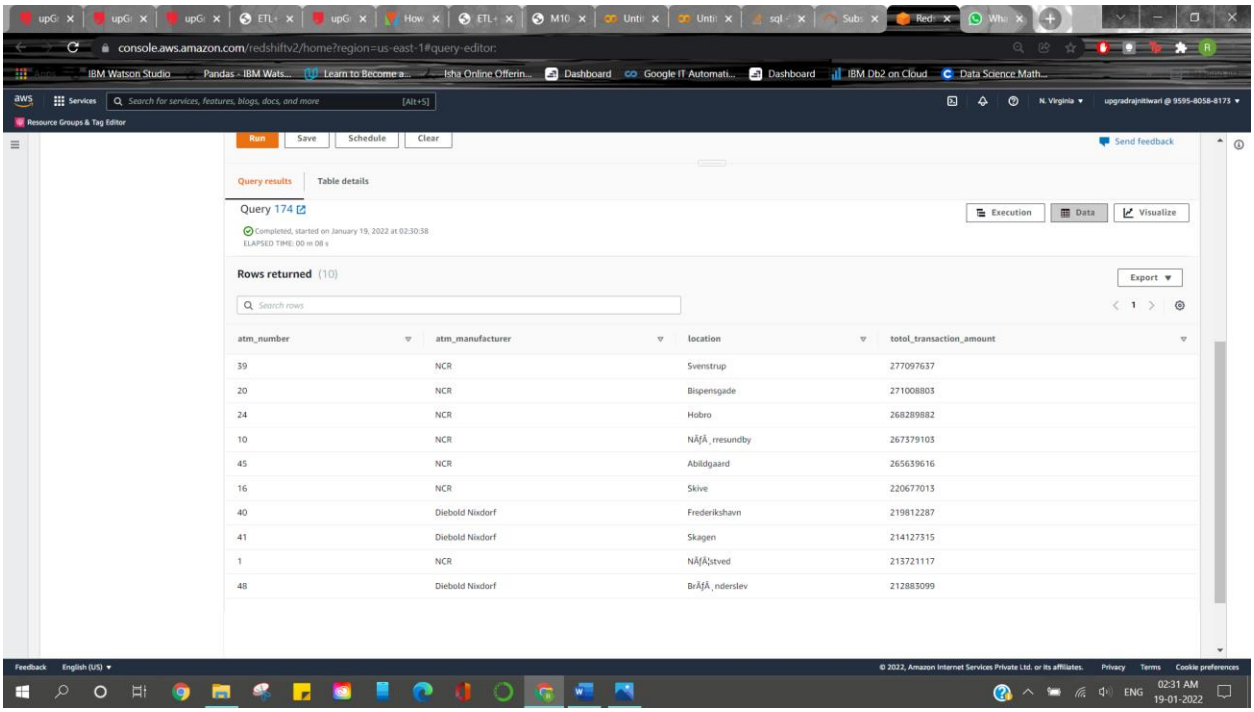
year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.112238137664770
2017	August	217218	36713	16.901453839000451
2017	December	197048	20476	10.391376720393000
2017	February	182659	36656	20.067995554557946
2017	January	180195	35953	19.952273925469630
2017	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



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
totol_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 totol_transaction_amount desc
limit 10;
```

<Screenshot of the resultant table>

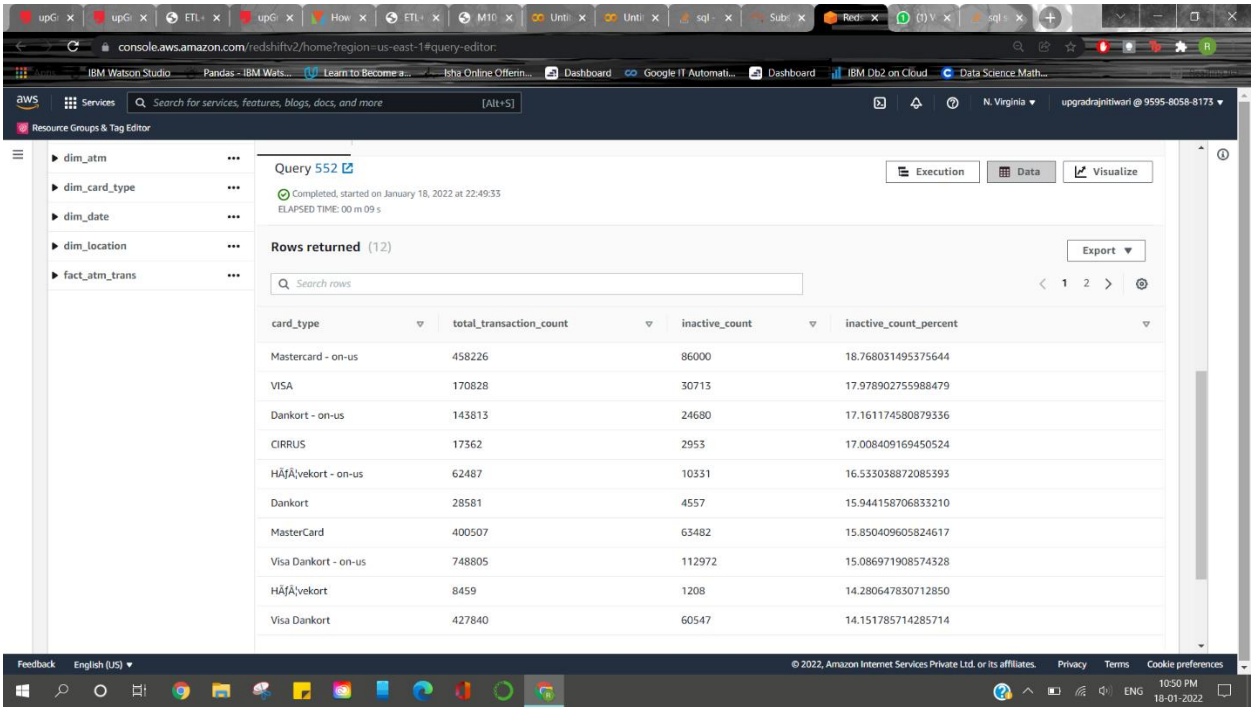


6. Number of failed ATM transactions across various card types

<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>



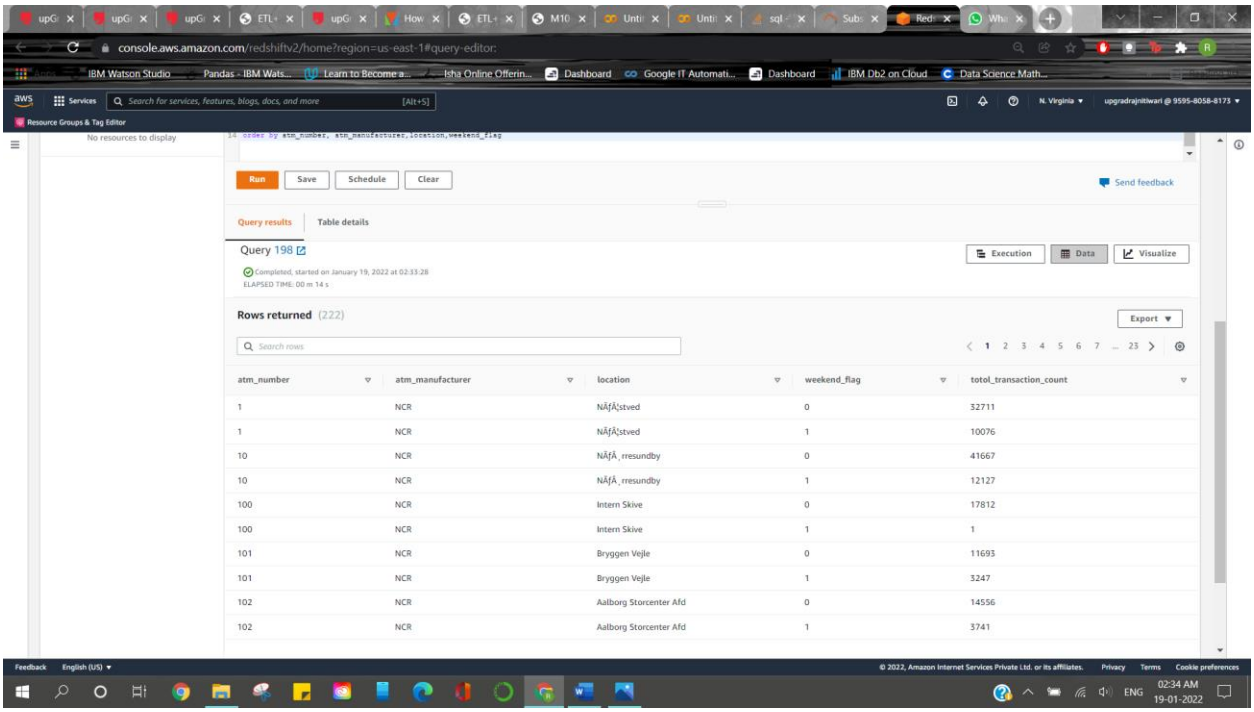
card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.768031495375644
VISA	170828	30713	17.978902755988479
Dankort - on-us	143813	24680	17.161174580879336
CIRRUS	17362	2953	17.008409169450524
HÃƒÂ¦vekort - on-us	62487	10331	16.533038872085393
Dankort	28581	4557	15.944158706833210
MasterCard	400507	63482	15.850409605824617
Visa Dankort - on-us	748805	112972	15.086971908574328
HÃƒÂ¦vekort	8459	1208	14.280647830712850
Visa Dankort	427840	60547	14.151785714285714

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**

<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>



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>