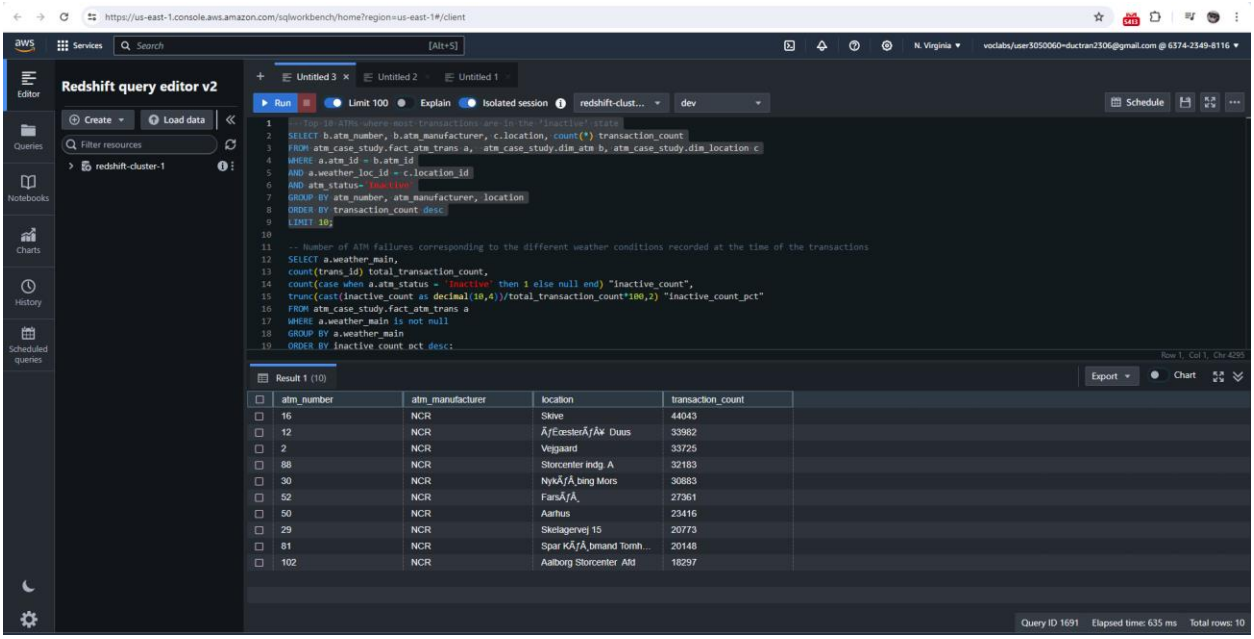
Solving analytical queries on Redshift Cluster

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.

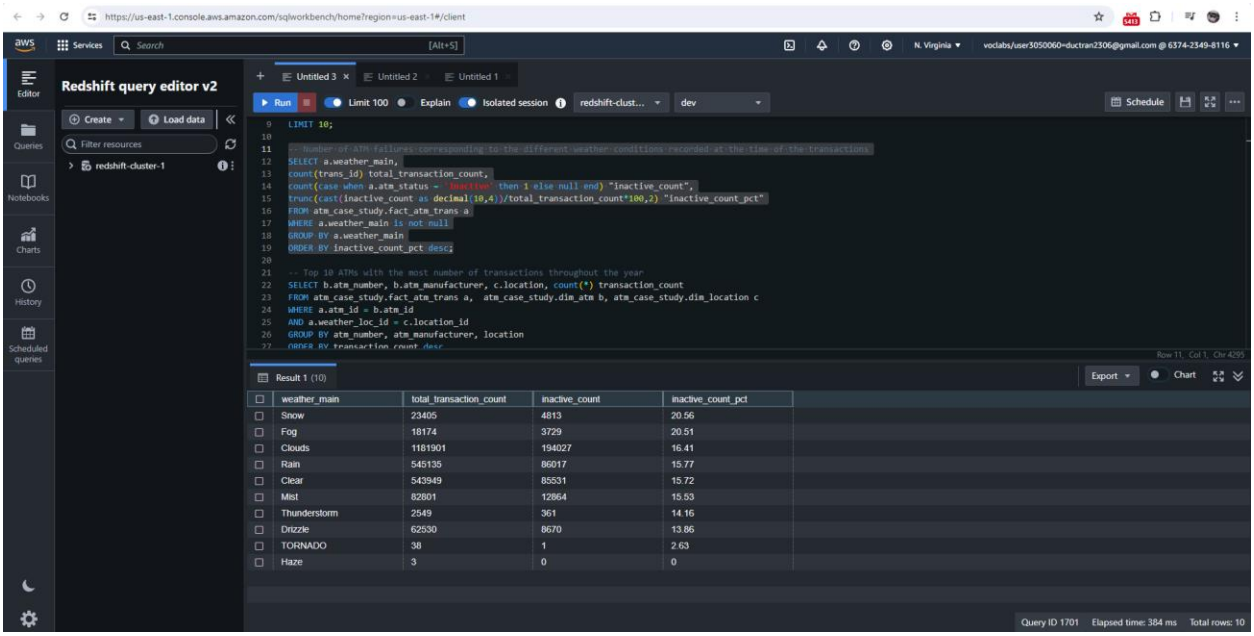
1. **Top 10 ATMs where most transactions are in the 'inactive' state**

```
-- Top 10 ATMs where most transactions are in the 'inactive' state
SELECT b.atm_number, b.atm_manufacturer, c.location, count(*) transaction_count
FROM atm_case_study.fact_atm_trans a,  atm_case_study.dim_atm b,
atm_case_study.dim_location c
WHERE a.atm_id = b.atm_id
AND a.weather_loc_id = c.location_id
AND atm_status='Inactive'
GROUP BY atm_number, atm_manufacturer, location
ORDER BY transaction_count desc
LIMIT 10;
```



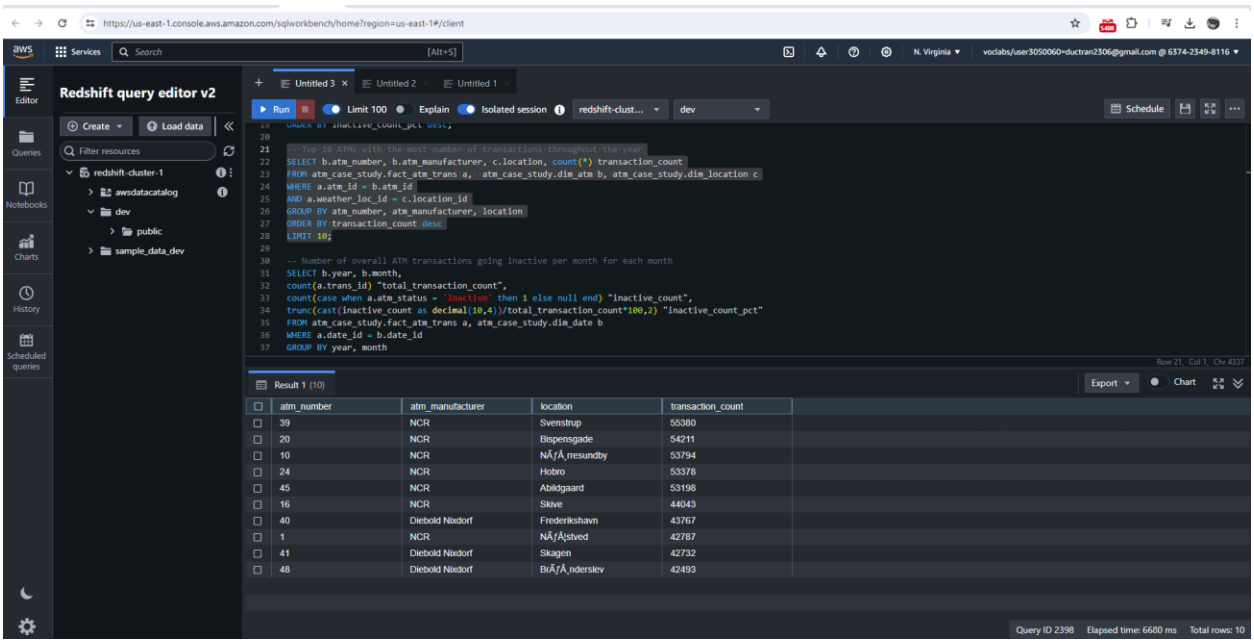
2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

```
-- Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions
SELECT a.weather_main,
count(trans_id) total_transaction_count,
count(case when a.atm_status = 'Inactive' then 1 else null end) "inactive_count",
trunc(cast(inactive_count as decimal(10,4))/total_transaction_count*100,2) "inactive_count_pct"
FROM atm_case_study.fact_atm_trans a
WHERE a.weather_main is not null
GROUP BY a.weather_main
ORDER BY inactive_count_pct desc;
```



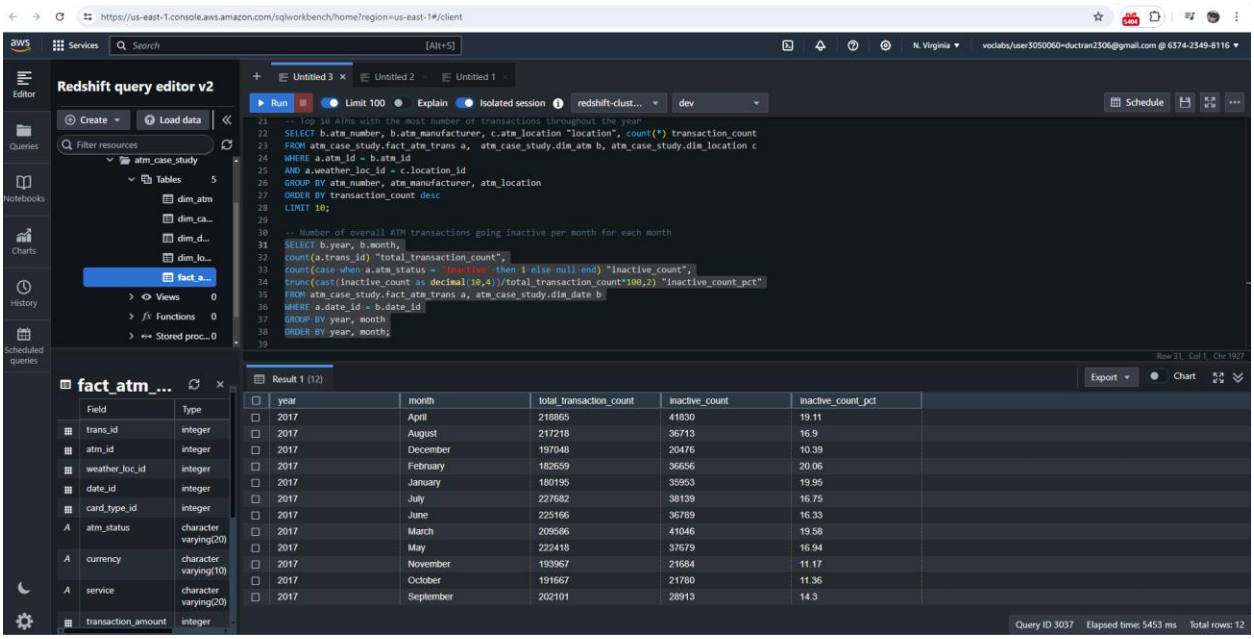
3. Top 10 ATMs with the most number of transactions throughout the year

```
-- Top 10 ATMs with the most number of transactions throughout the year
SELECT b.atm_number, b.atm_manufacturer, c.location, count(*) transaction_count
FROM atm_case_study.fact_atm_trans a,  atm_case_study.dim_atm b,
atm_case_study.dim_location c
WHERE a.atm_id = b.atm_id
AND a.weather_loc_id = c.location_id
GROUP BY atm_number, atm_manufacturer, location
ORDER BY transaction_count desc
LIMIT 10;
```



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

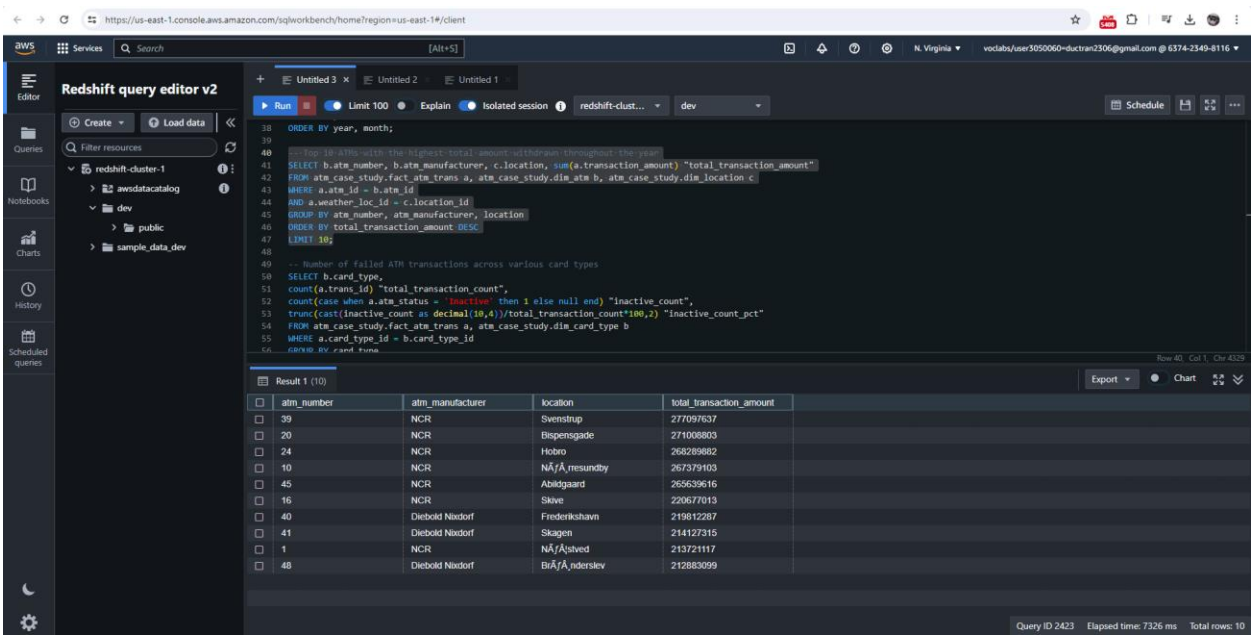
```
-- Number of overall ATM transactions going inactive per month for each month
SELECT b.year, b.month,
count(a.trans_id) "total_transaction_count",
count(case when a.atm_status = 'Inactive' then 1 else null end) "inactive_count",
trunc(cast(inactive_count as decimal(10,4))/total_transaction_count*100,2) "inactive_count_pct"
FROM atm_case_study.fact_atm_trans a, atm_case_study.dim_date b
WHERE a.date_id = b.date_id
GROUP BY year, month
ORDER BY year, month;
```



year	month	total_transaction_count	inactive_count	inactive_count_pct
2017	April	218865	41830	19.11
2017	August	217218	36713	16.9
2017	December	197048	20476	10.39
2017	February	182659	36656	20.06
2017	January	180195	35953	19.95
2017	July	227682	38139	16.75
2017	June	225166	36789	16.33
2017	March	209586	41046	19.58
2017	May	222418	37679	16.94
2017	November	193967	21684	11.17
2017	October	191667	21780	11.36
2017	September	202101	28913	14.3

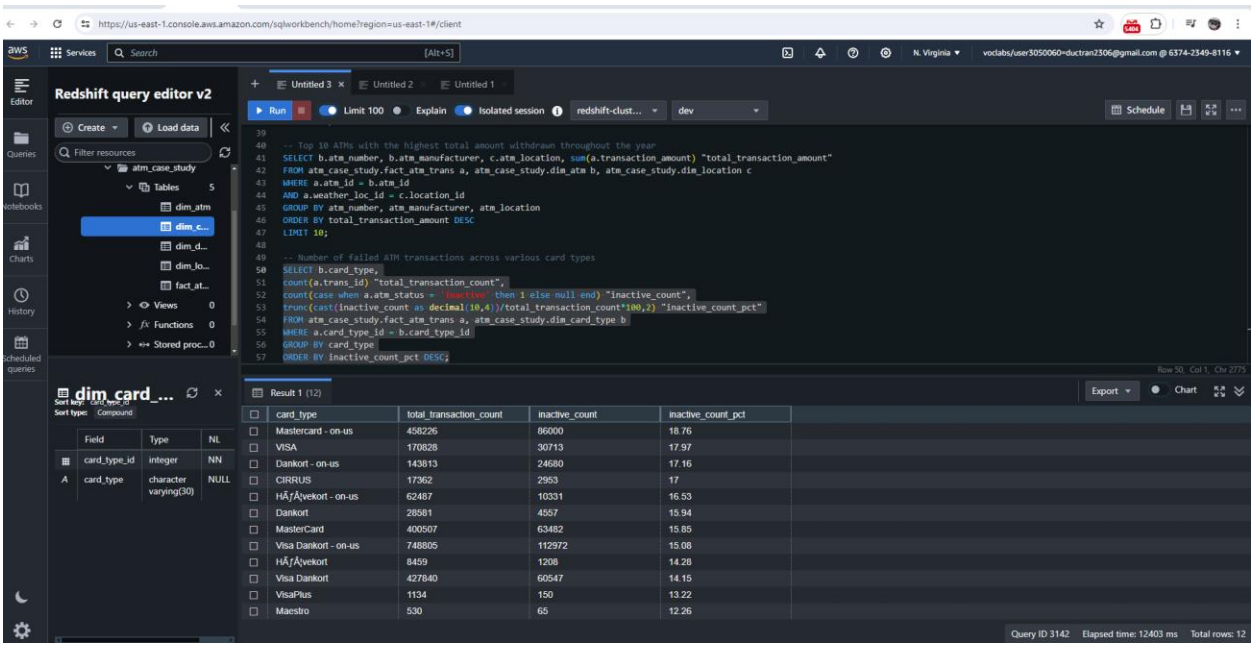
5. **Top 10 ATMs with the highest total withdrawn amount throughout the year**

```
-- Top 10 ATMs with the highest total amount withdrawn throughout the year
SELECT b.atm_number, b.atm_manufacturer, c.location, sum(a.transaction_amount)
"total_transaction_amount"
FROM atm_case_study.fact_atm_trans a, atm_case_study.dim_atm b,
atm_case_study.dim_location c
WHERE a.atm_id = b.atm_id
AND a.weather_loc_id = c.location_id
GROUP BY atm_number, atm_manufacturer, location
ORDER BY total_transaction_amount DESC
LIMIT 10;
```



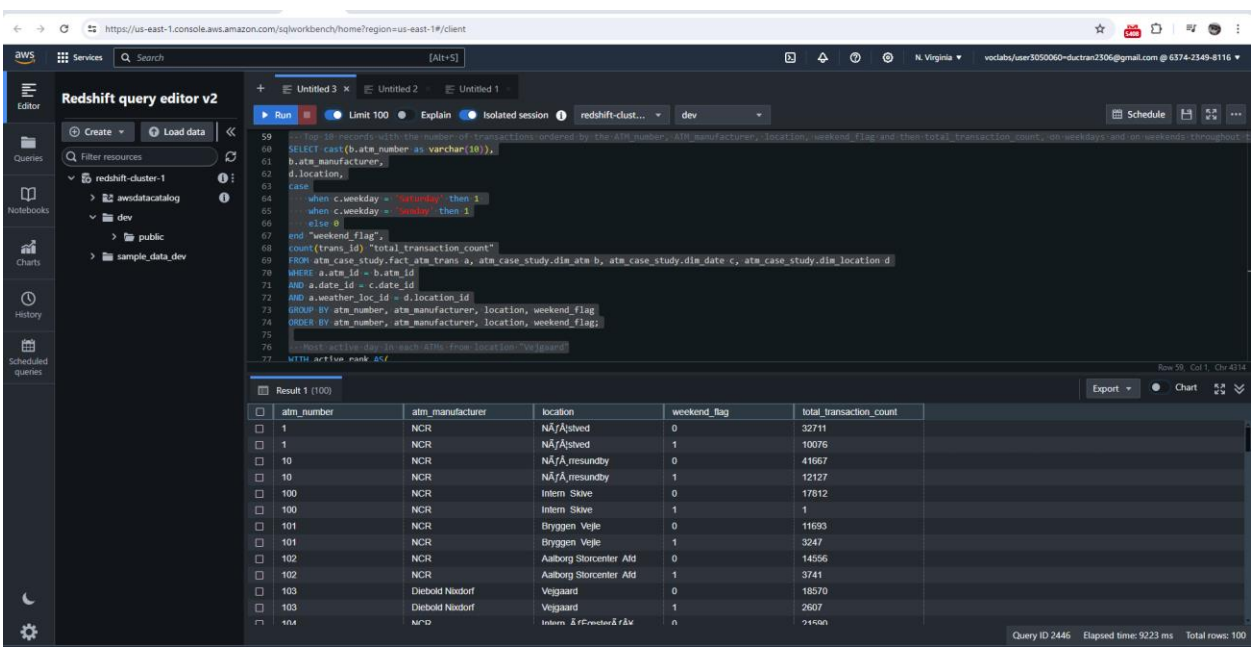
6. Number of failed ATM transactions across various card types

```
-- Number of failed ATM transactions across various card types
SELECT b.card_type,
count(a.trans_id) "total_transaction_count",
count(case when a.atm_status = 'Inactive' then 1 else null end) "inactive_count",
trunc(cast(inactive_count as decimal(10,4))/total_transaction_count*100,2) "inactive_count_pct"
FROM atm_case_study.fact_atm_trans a, atm_case_study.dim_card_type b
WHERE a.card_type_id = b.card_type_id
GROUP BY card_type
ORDER BY inactive_count_pct DESC;
```



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 cast(b.atm_number as varchar(10)),
b.atm_manufacturer,
d.location,
case
   when c.weekday = 'Saturday' then 1
   when c.weekday = 'Sunday' then 1
   else 0
end "weekend_flag",
count(trans_id) "total_transaction_count"
FROM atm_case_study.fact_atm_trans a, atm_case_study.dim_atm b,
atm_case_study.dim_date c, atm_case_study.dim_location d
WHERE a.atm_id = b.atm_id
AND a.date_id = c.date_id
AND a.weather_loc_id = d.location_id
GROUP BY atm_number, atm_manufacturer, location, weekend_flag
ORDER BY atm_number, atm_manufacturer, location, weekend_flag;
```



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

```
-- Most active day in each ATMs from location "Vejgaard"
WITH active_rank AS(
SELECT b.atm_number,
b.atm_manufacturer,
d.location,
c.weekday,
count(trans_id) "total_transaction_count",
RANK() OVER(PARTITION BY atm_number, atm_manufacturer, location ORDER BY
total_transaction_count DESC) "Rank"
FROM atm_case_study.fact_atm_trans a, atm_case_study.dim_atm b,
atm_case_study.dim_date c, atm_case_study.dim_location d
WHERE a.atm_id = b.atm_id
AND a.date_id = c.date_id
AND a.weather_loc_id = d.location_id
AND d.location = 'Vejgaard'
GROUP BY  atm_number, atm_manufacturer, location, weekday
ORDER BY atm_number, atm_manufacturer, location, weekday
)
SELECT atm_number, atm_manufacturer, location, weekday, total_transaction_count
FROM active_rank
WHERE rank = 1;
```