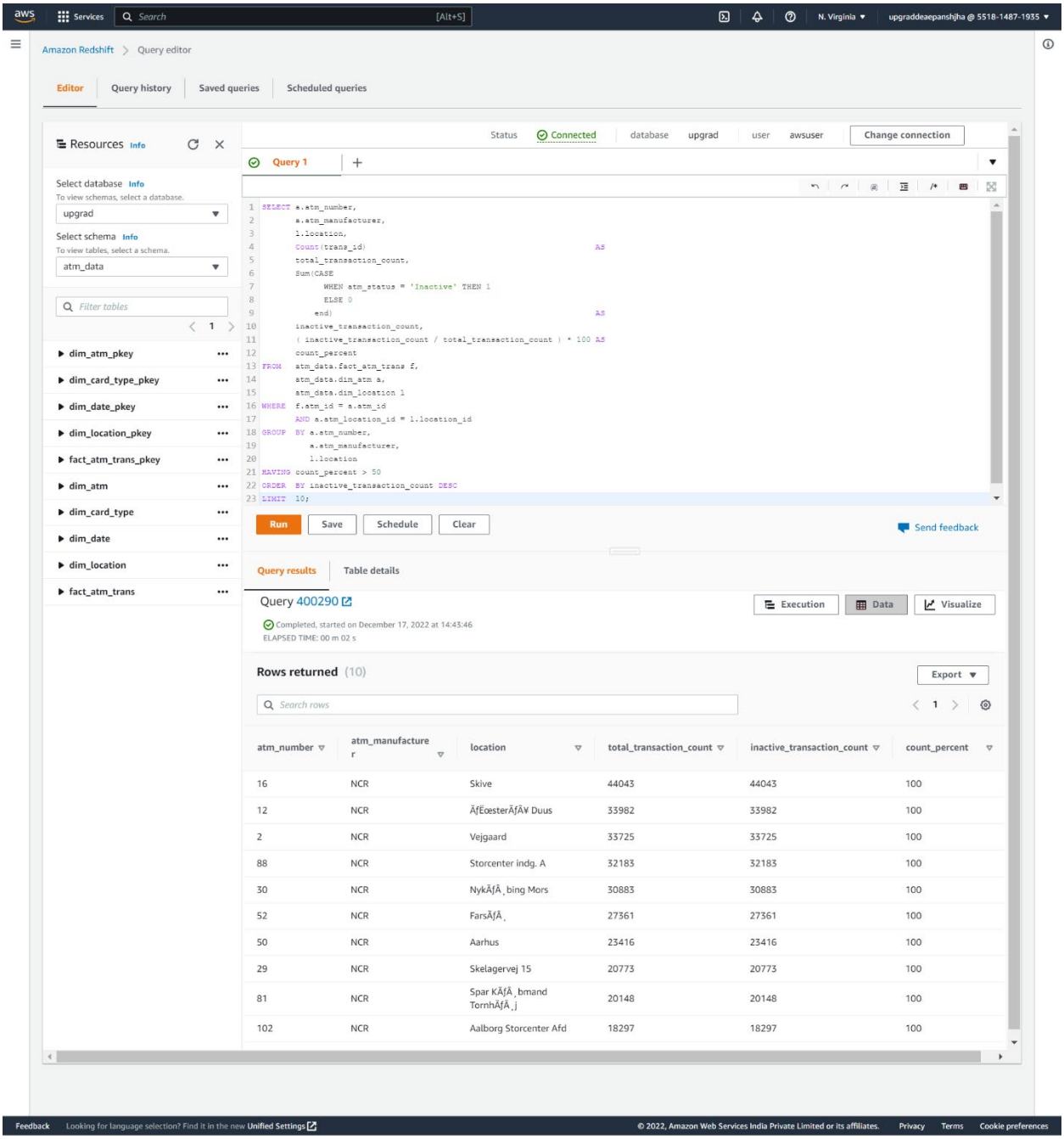
Solving analytical queries on Redshift Cluster

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**

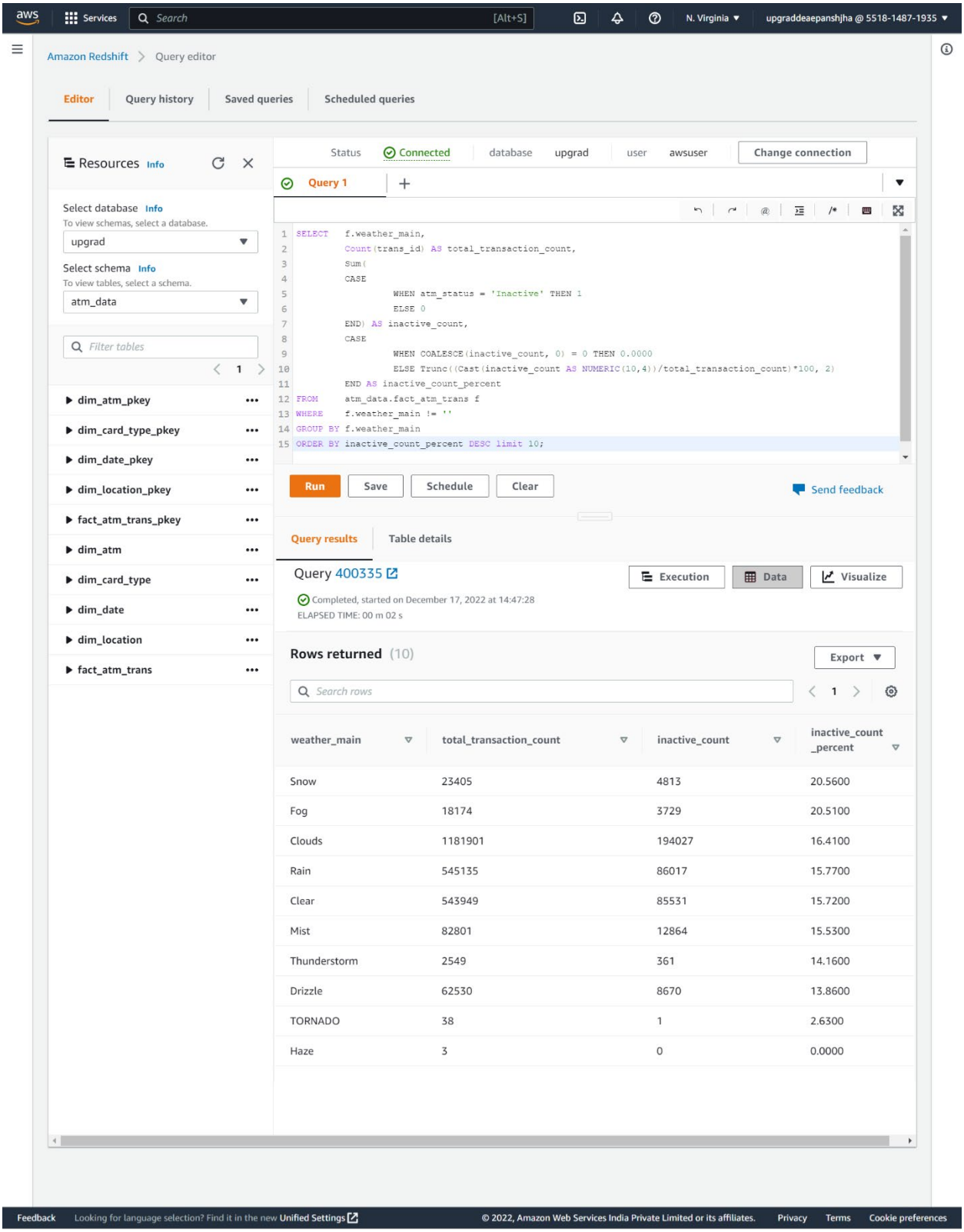
```
SELECT a.atm_number,
       a.atm_manufacturer,
       l.location,
       Count(trans_id)
        AS
       total_transaction_count,
       Sum(CASE
             WHEN atm_status = 'Inactive' THEN 1
             ELSE 0
           end)
        AS
       inactive_transaction_count,
       ( inactive_transaction_count / 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_transaction_count DESC
LIMIT  10;
```



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;
```

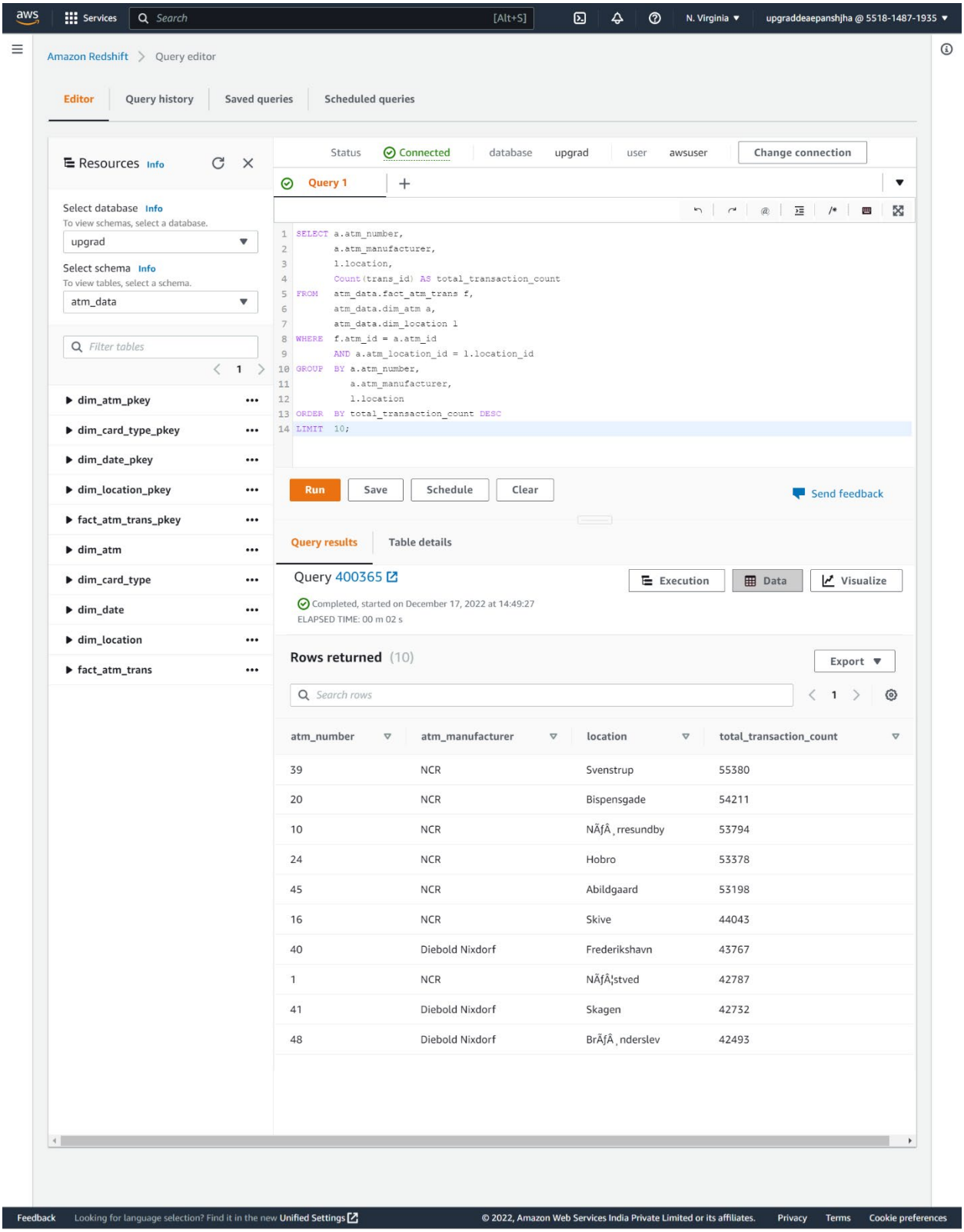
Question 2 Screenshot Below ↓



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;
```

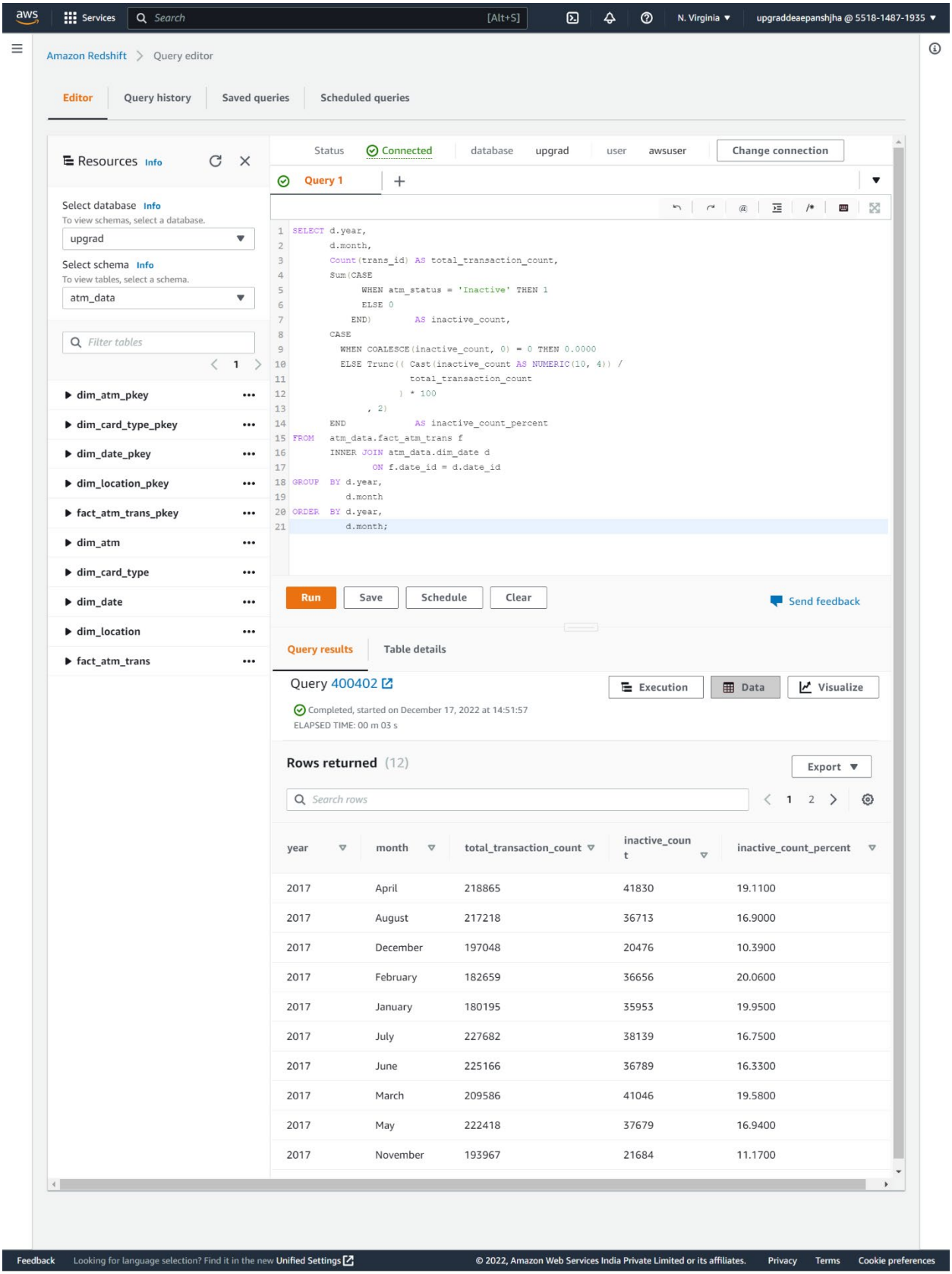
Question 3 Screenshot Below ↓



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;
```

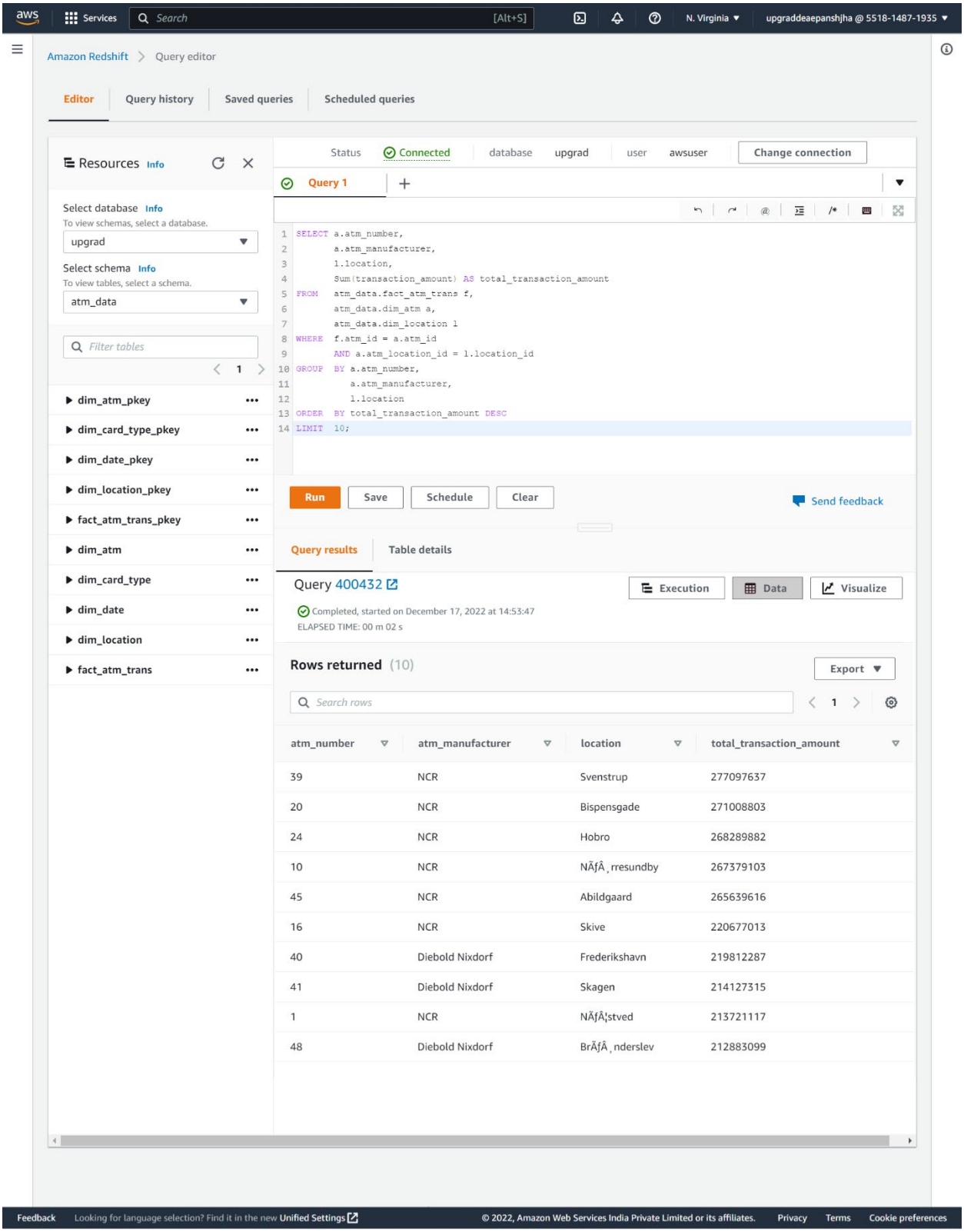
Question 4 Screenshot Below ↓



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;
```

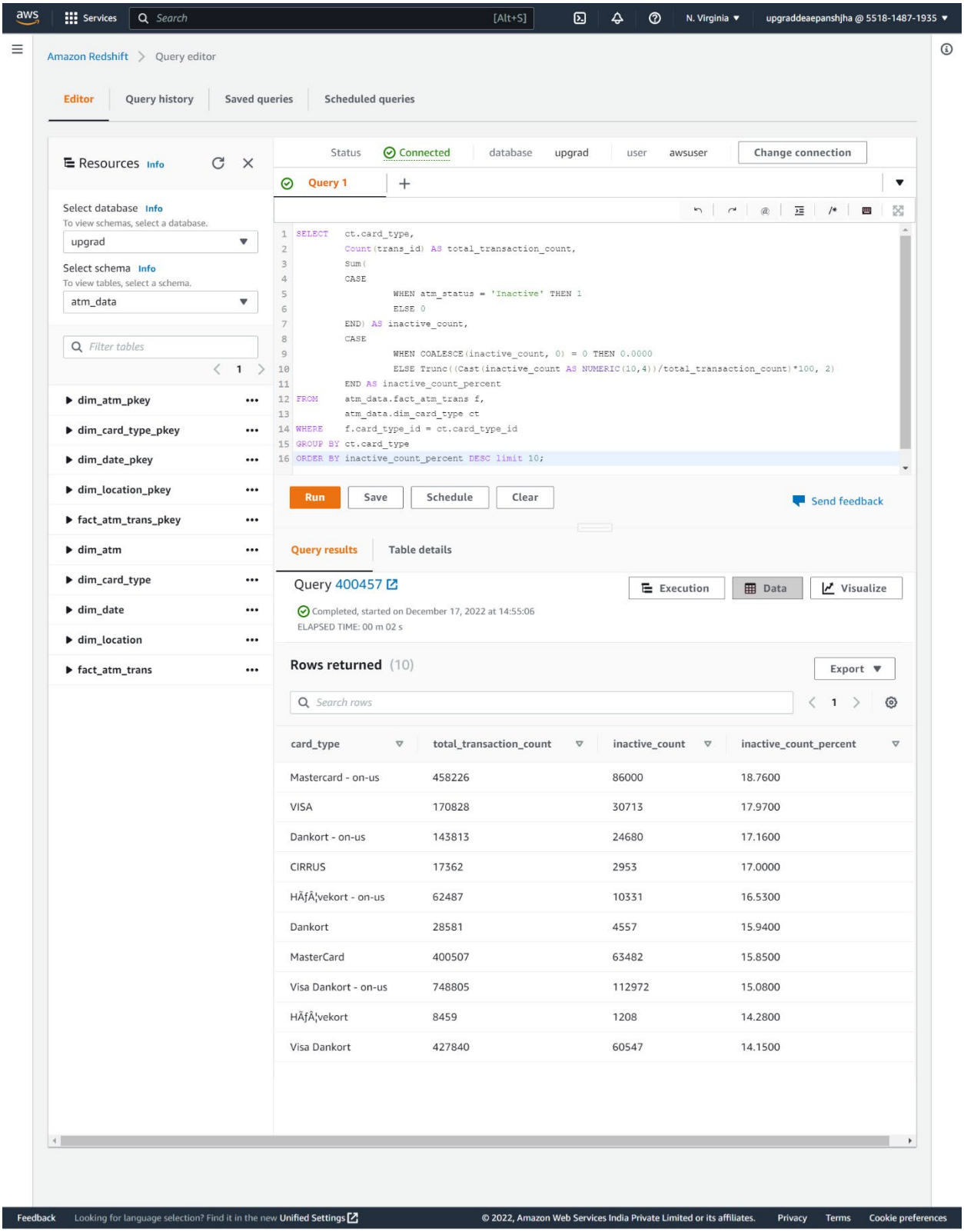
Question 5 Screenshot Below ↓



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;
```

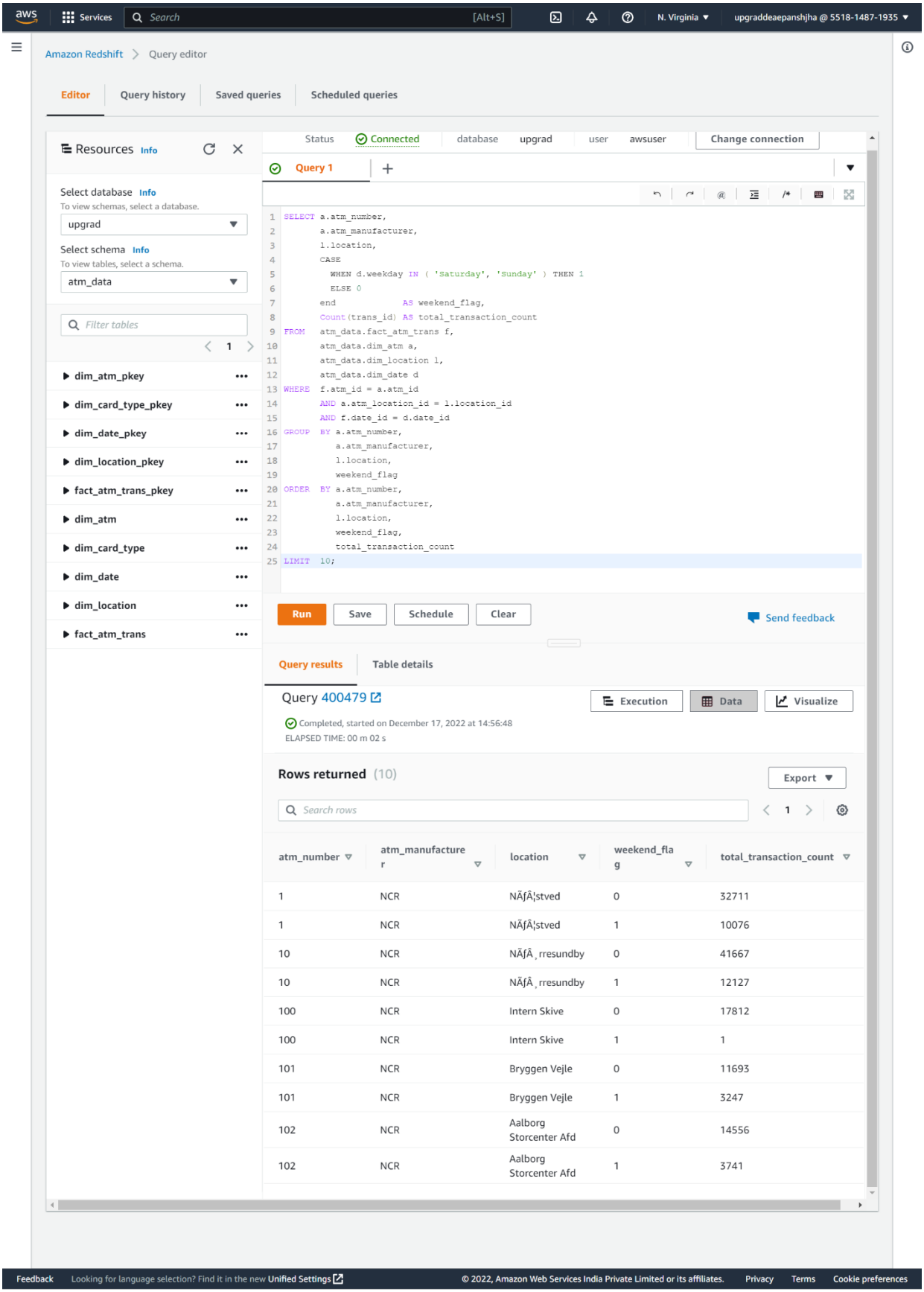
Question 6 Screenshot Below ↓



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;
```

Question 7 Screenshot Below ↓



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.loca
tion_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;
```

Question 8 Screenshot Below ↓