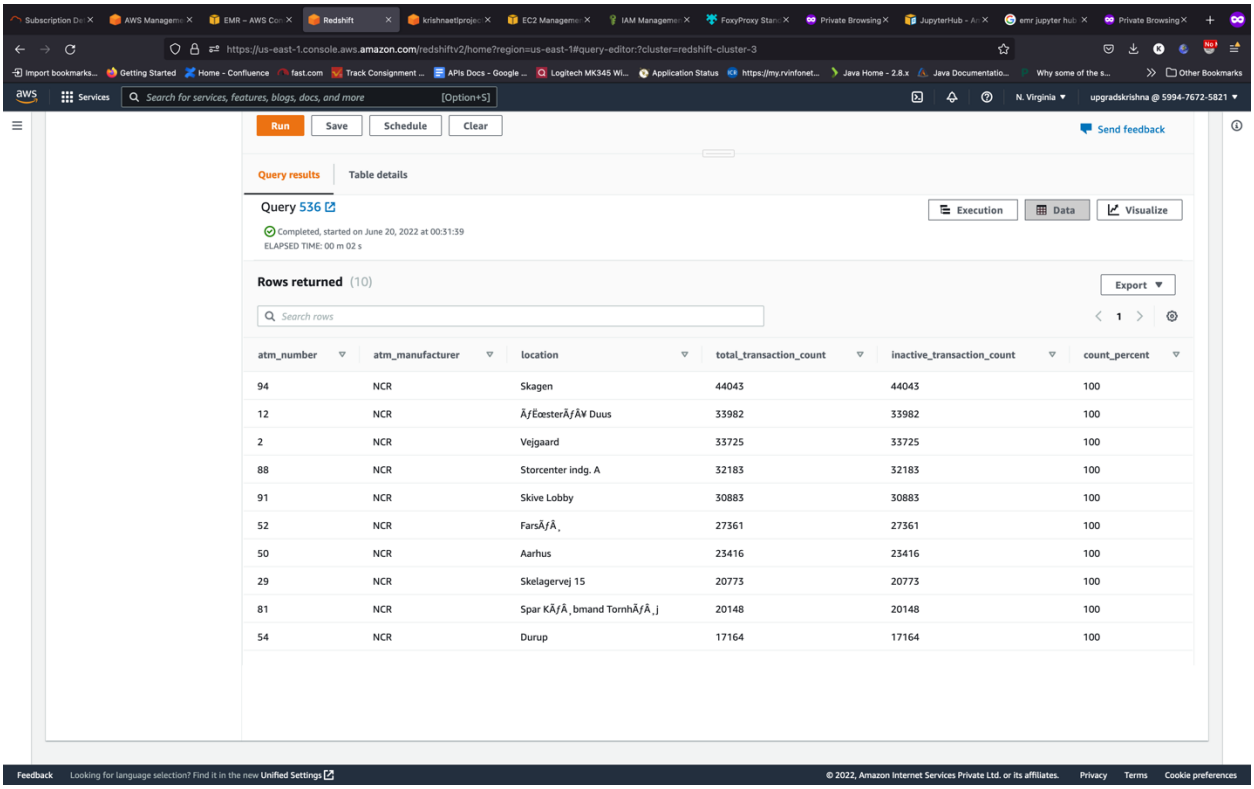
Solving analytical queries on Redshift Cluster

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.

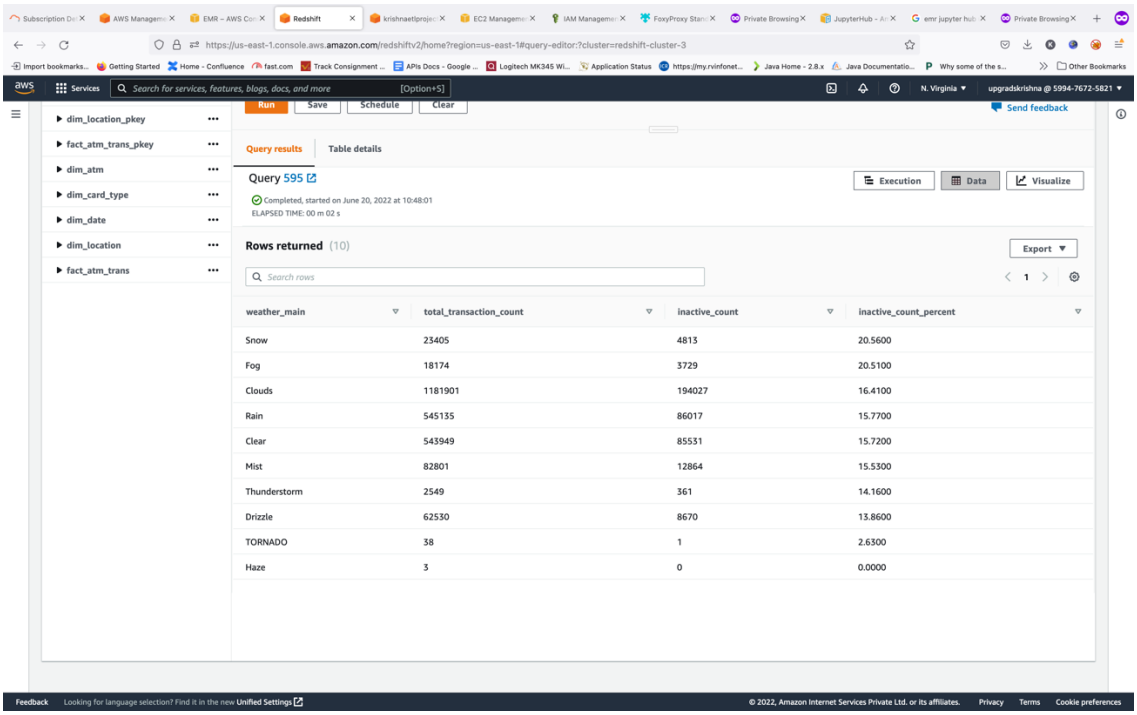
1. **Top 10 ATMs where most transactions are in the 'inactive' state**

```
--Top 10 ATMs where most transactions are in the 'inactive' state
select
    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;
```



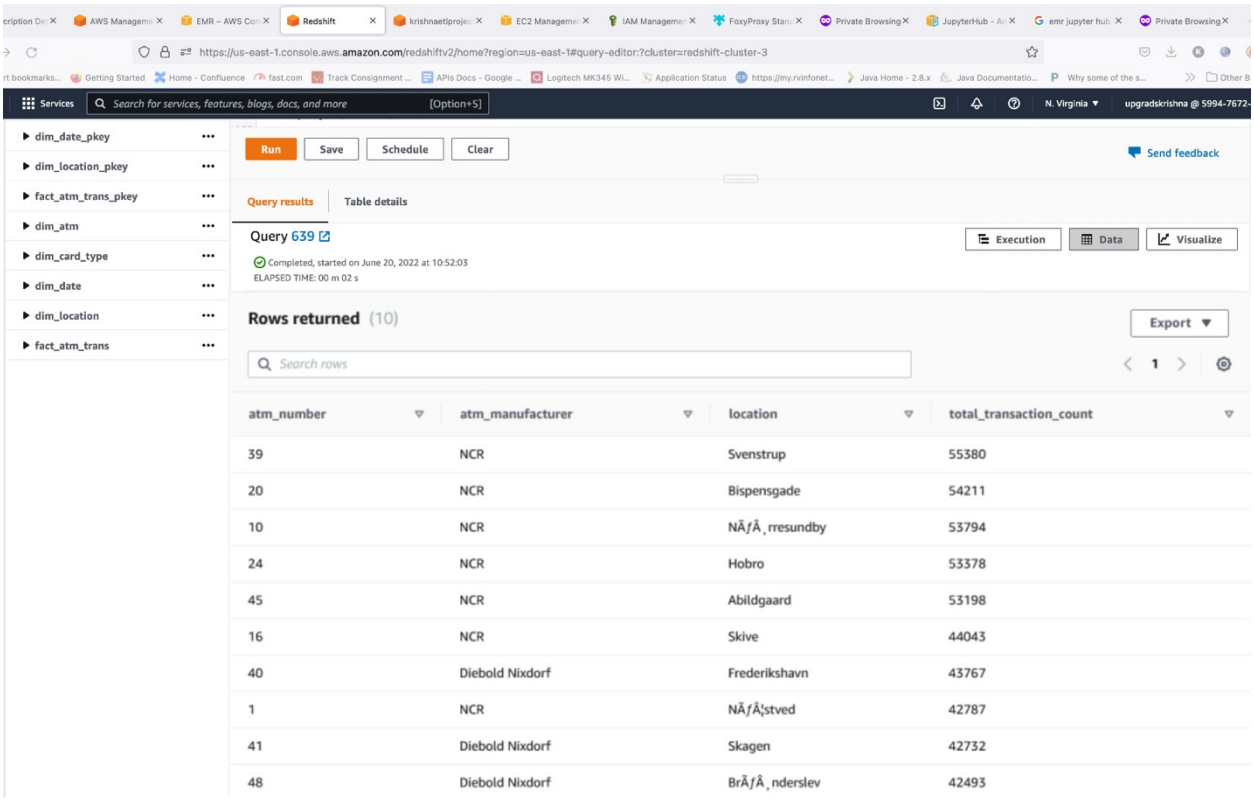
2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

```
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;
```



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

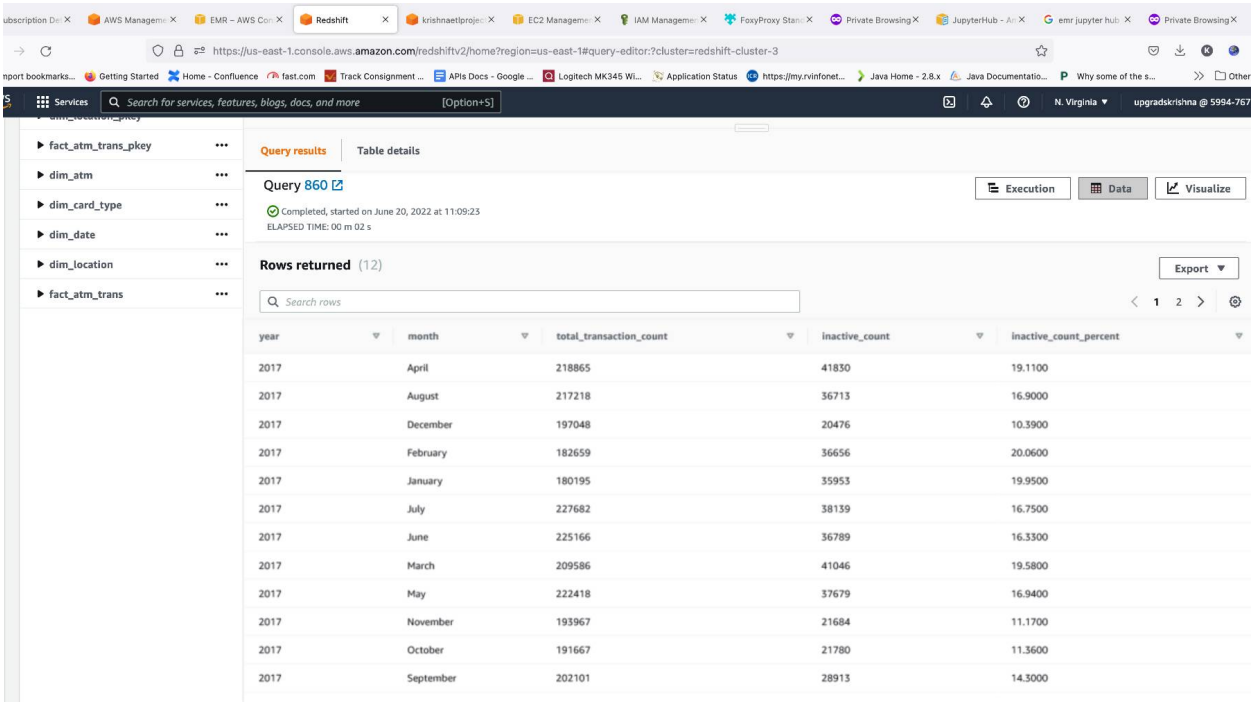
```
--3. Top 10 ATMs with the most number of transactions throughout the year
select
    a.atm_number,
    a.atm_manufacturer,
    l.location,
    count(trans_id) as total_transaction_count
from
    atm_data.fact_atm_trans f,
    atm_data.dim_atm a,
    atm_data.dim_location l
where
    f.atm_id = a.atm_id
    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;
```



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

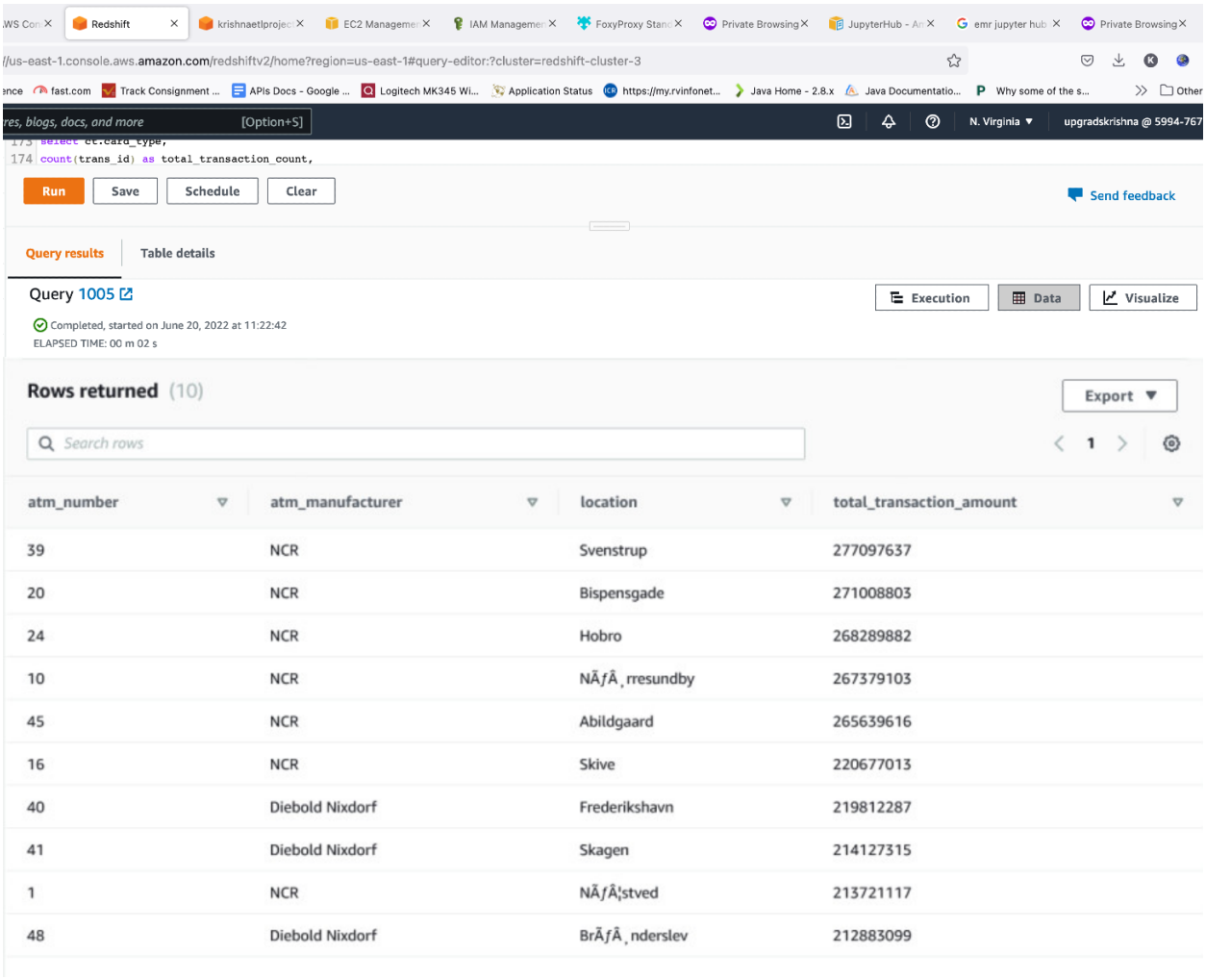
```
select
    d.year,
    d.month,
    count(trans_id) as total_transaction_count,
    sum(
        case
            when atm_status = 'Inactive' then 1
            else 0
        end
    ) as inactive_count,
    case
        when coalesce(inactive_count, 0) = 0 then 0.0000
        else trunc(
            (
                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;
```



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

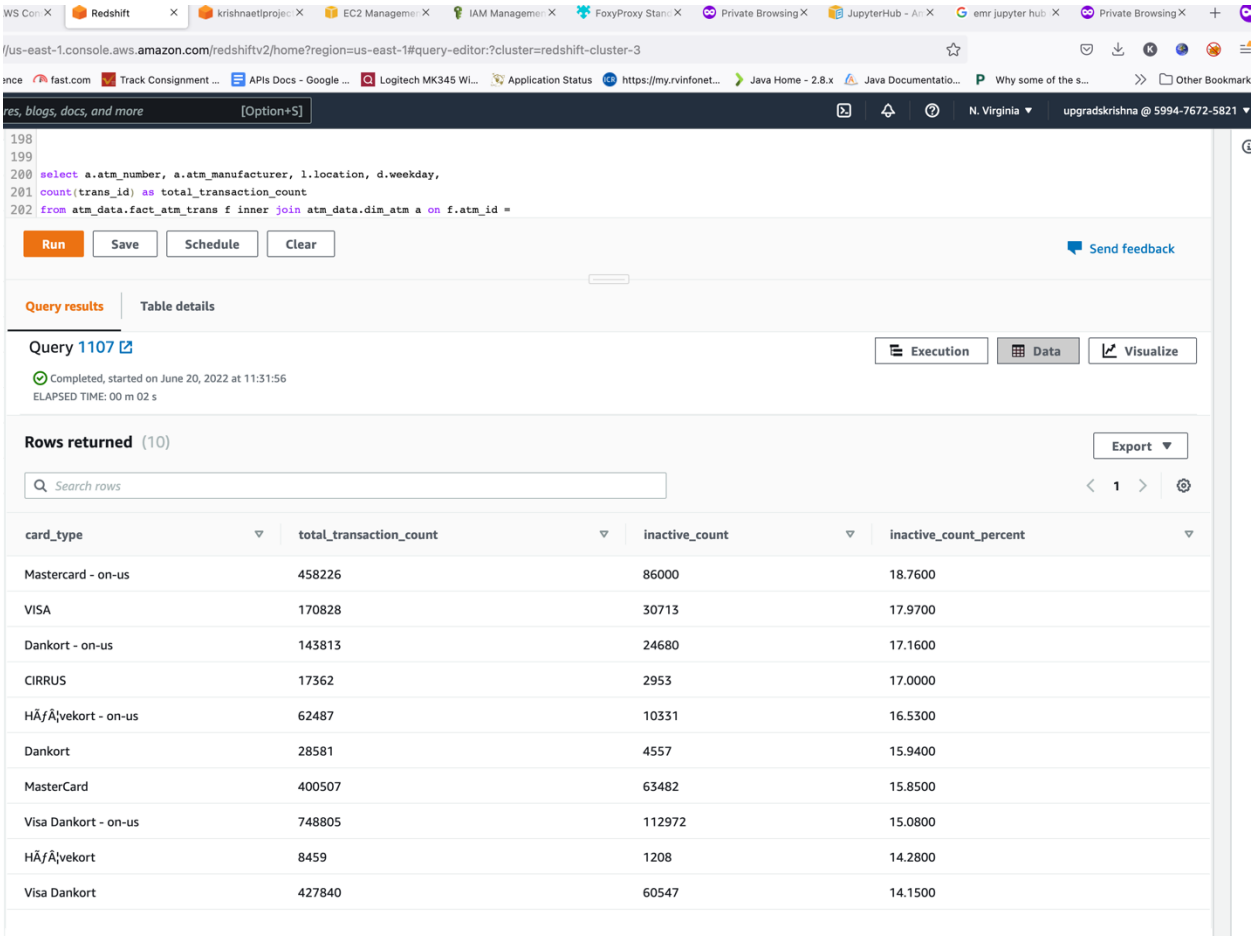
```sql
select
    a.atm_number,
    a.atm_manufacturer,
    l.location,
    sum(transaction_amount) as total_transaction_amount
from
    atm_data.fact_atm_trans f,
    atm_data.dim_atm a,
    atm_data.dim_location l
where
    f.atm_id = a.atm_id
    and a.atm_location_id = l.location_id
group by
    a.atm_number,
    a.atm_manufacturer,
    l.location
order by
    total_transaction_amount desc
limit 10;
```



6. Number of failed ATM transactions across various card types

```
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
```

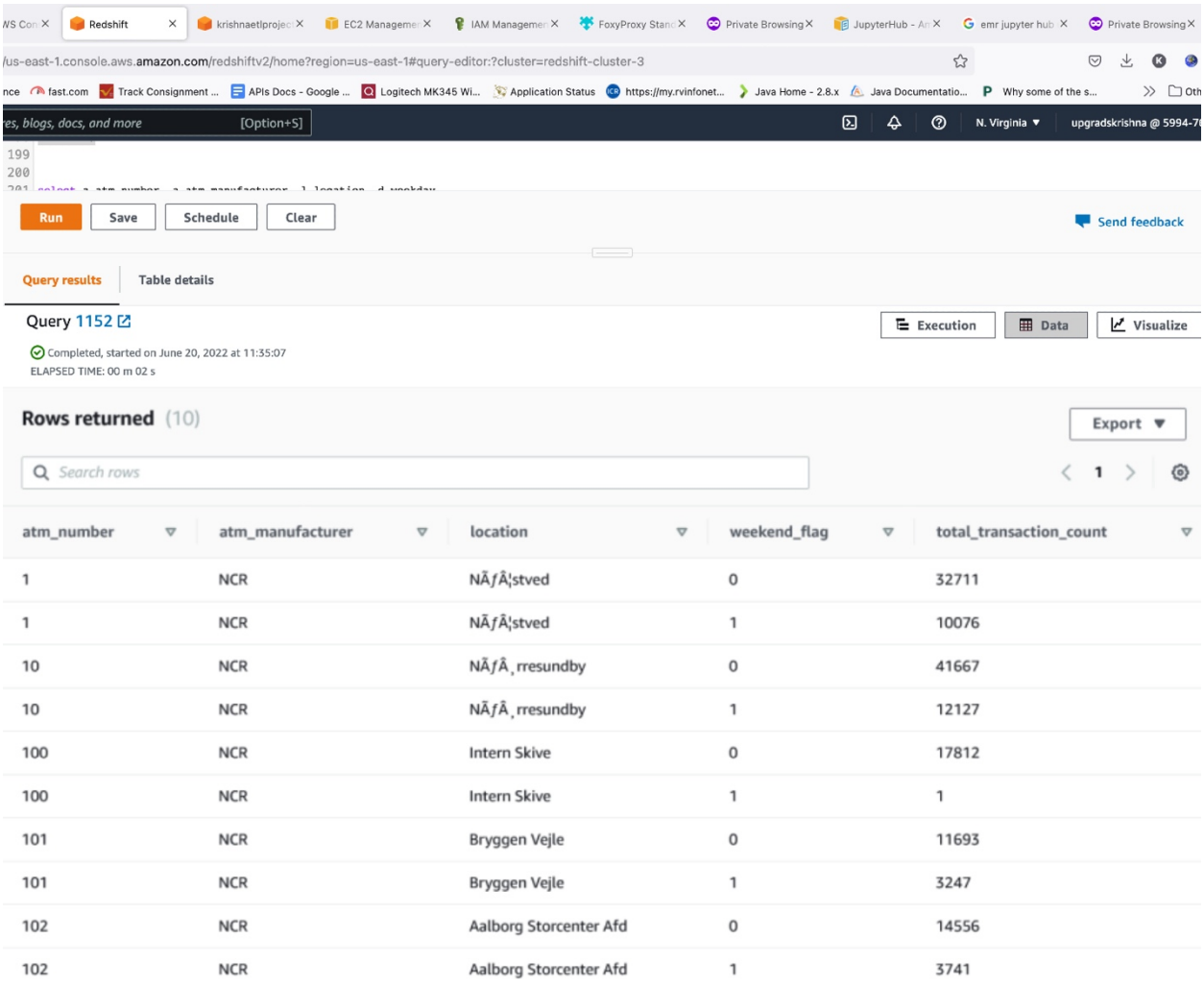
```
order by
    total_transaction_amount desc
limit 10;
```



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;
```



atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃƒÂ¦stved	0	32711
1	NCR	NÃƒÂ¦stved	1	10076
10	NCR	NÃƒÂ¸rresundby	0	41667
10	NCR	NÃƒÂ¸rresundby	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 Afd	0	14556
102	NCR	Aalborg Storcenter Afd	1	3741

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

```sql
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;
```