### 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 R.atm\_number, R.atm\_manufacturer,

S.location, Q.total\_transaction\_count, Q.inactive\_count, (Q.total\_transaction\_count\*100/Q.inactive\_count) as count\_percent

from ( select atm\_id, count(\*) total\_transaction\_count,

sum(case when atm\_status = 'Inactive' then 1 else 0 end) inactive\_count

from bankatm\_schema.atm\_trans

where atm\_status = 'Inactive'

group by atm\_id

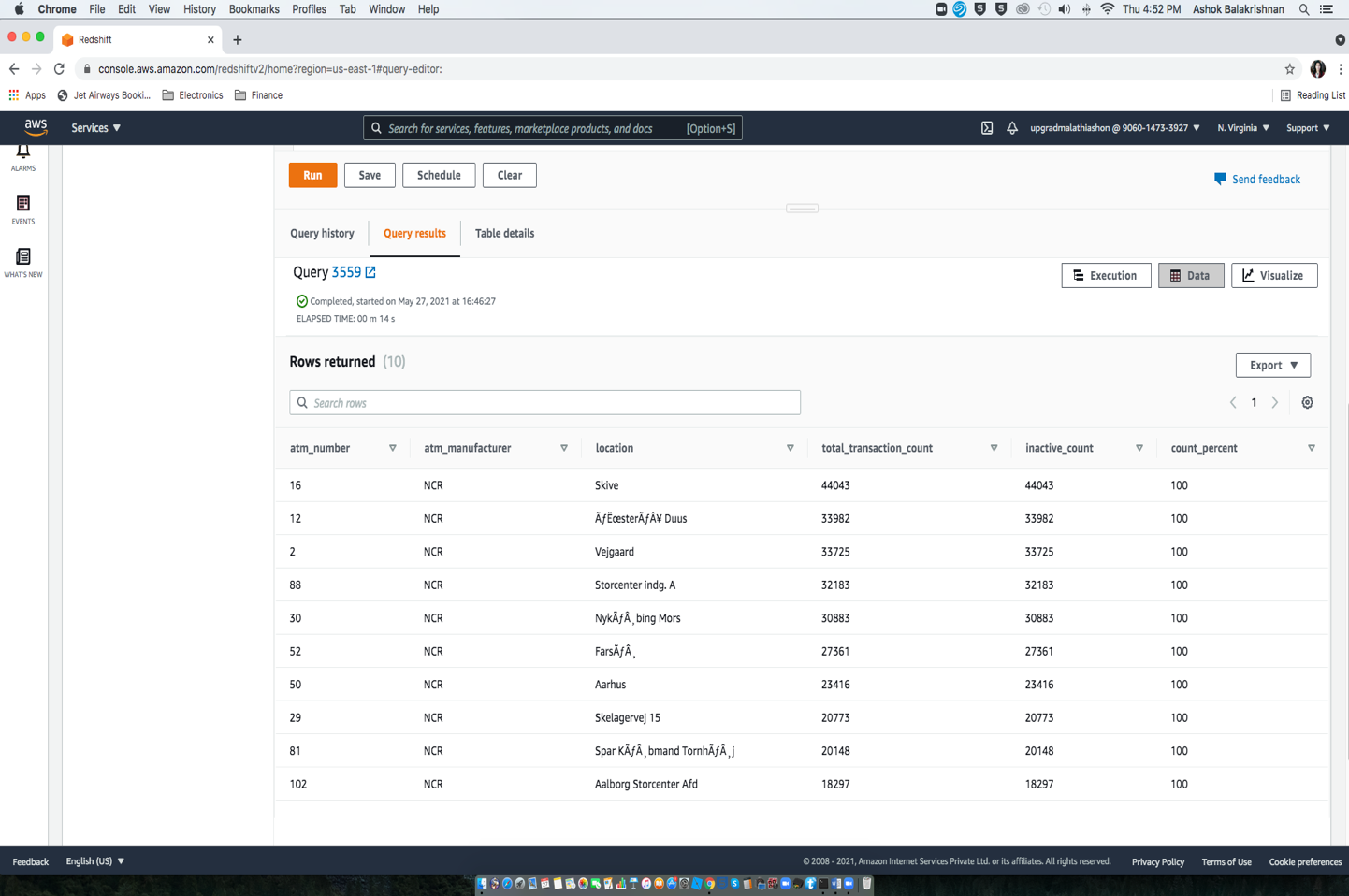
order by total\_transaction\_count desc limit 10) Q,

bankatm\_schema.atm R,

bankatm\_schema.location S

where Q.atm\_id = R.atm\_id and R.atm\_location\_id = S.location\_id

order by total\_transaction\_count desc;



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

select Q.weather\_main, count(\*) as total\_transaction\_count,

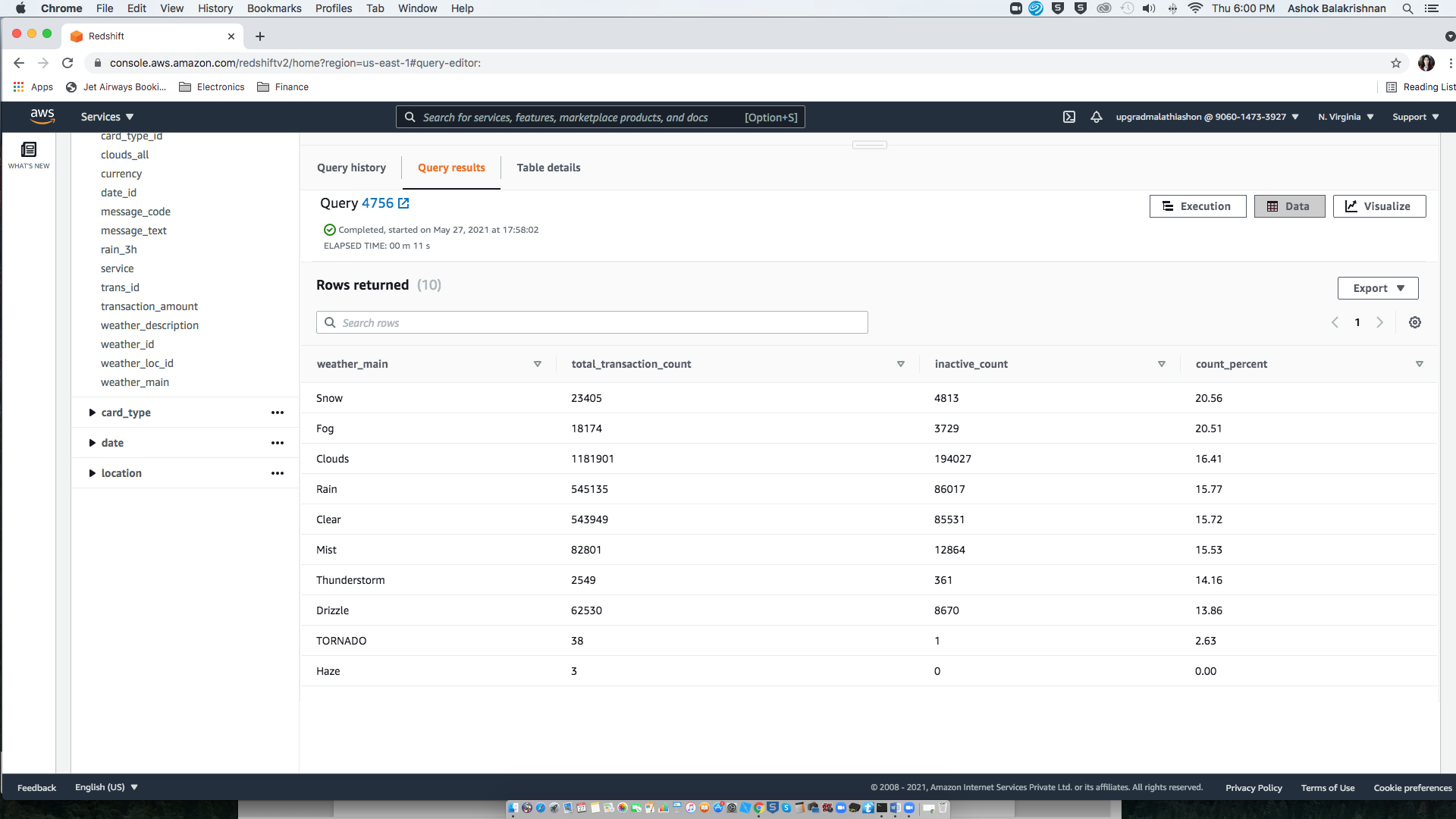
sum(case when Q.atm\_status = 'Inactive' then 1 else 0 end) as inactive\_count, CAST(inactive\_count\*100.0/total\_transaction\_count as decimal(5,2)) as count\_percent

from bankatm\_schema.atm\_trans Q, bankatm\_schema.atm R

where Q.atm\_id = R.atm\_id and Q.weather\_main != ''

group by Q.weather\_main

order by count\_percent desc;



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

select R.atm\_number, R.atm\_manufacturer, S.location,

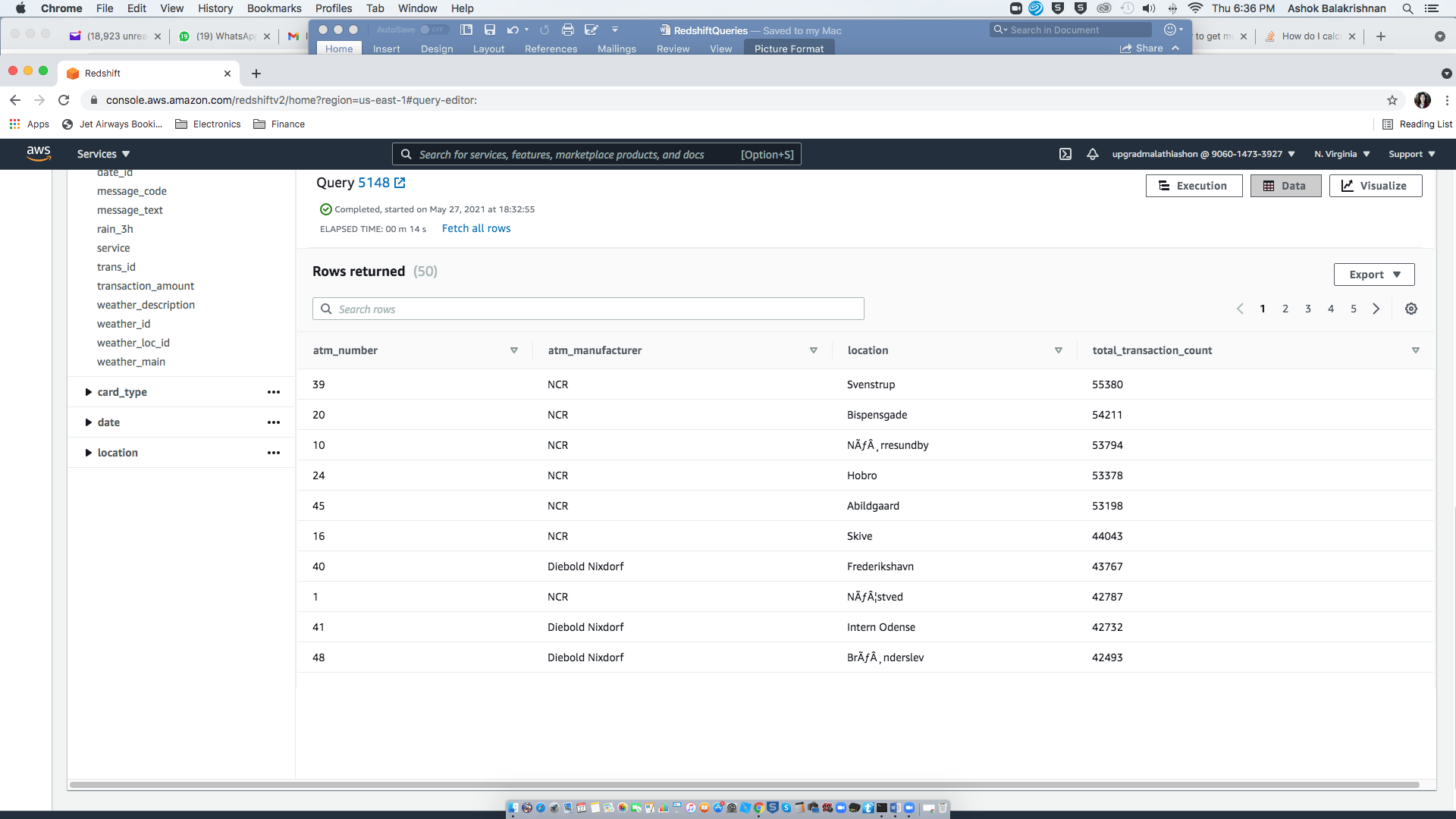
count(\*) total\_transaction\_count

from bankatm\_schema.atm\_trans Q, bankatm\_schema.atm R, bankatm\_schema.location S

where Q.atm\_id = R.atm\_id and R.atm\_location\_id = S.location\_id

group by R.atm\_number, R.atm\_manufacturer, S.location

order by total\_transaction\_count desc;



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

select Q.Year, Q.month, count(P.\*) total\_transaction\_count,

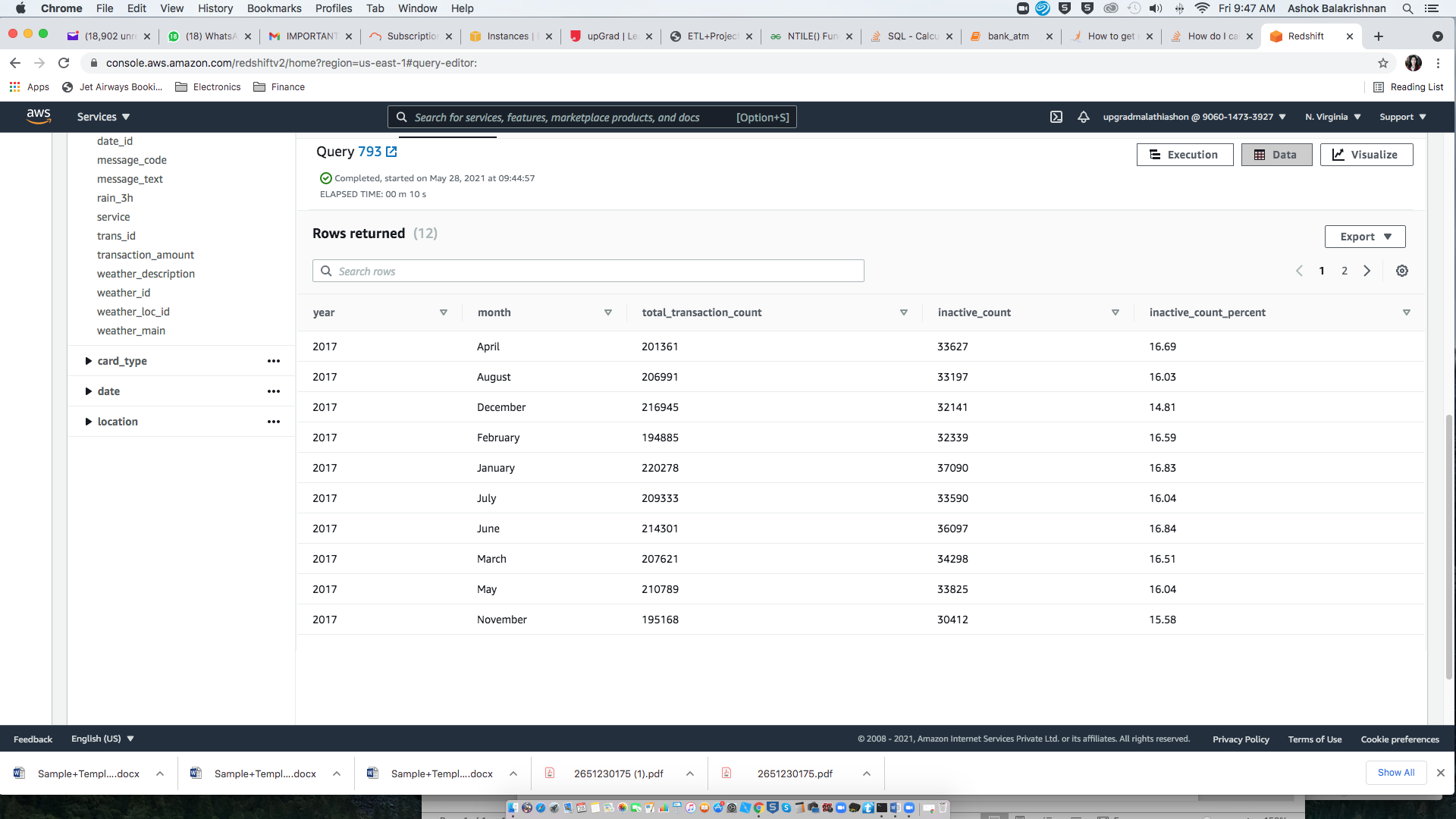
sum(case when P.atm\_status = 'Inactive' then 1 else 0 end) inactive\_count, CAST(inactive\_count\*100.0/total\_transaction\_count as decimal(5,2)) as inactive\_count\_percent

from bankatm\_schema.atm\_trans P, bankatm\_schema.date Q

where P.date\_id = Q.date\_id

group by Q.Year, Q.month

order by Q.Year, Q.month

****

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

select R.atm\_number, R.atm\_manufacturer, S.location, Q.total\_transaction\_amount

from (select atm\_id, sum(transaction\_amount) total\_transaction\_amount

from bankatm\_schema.atm\_trans

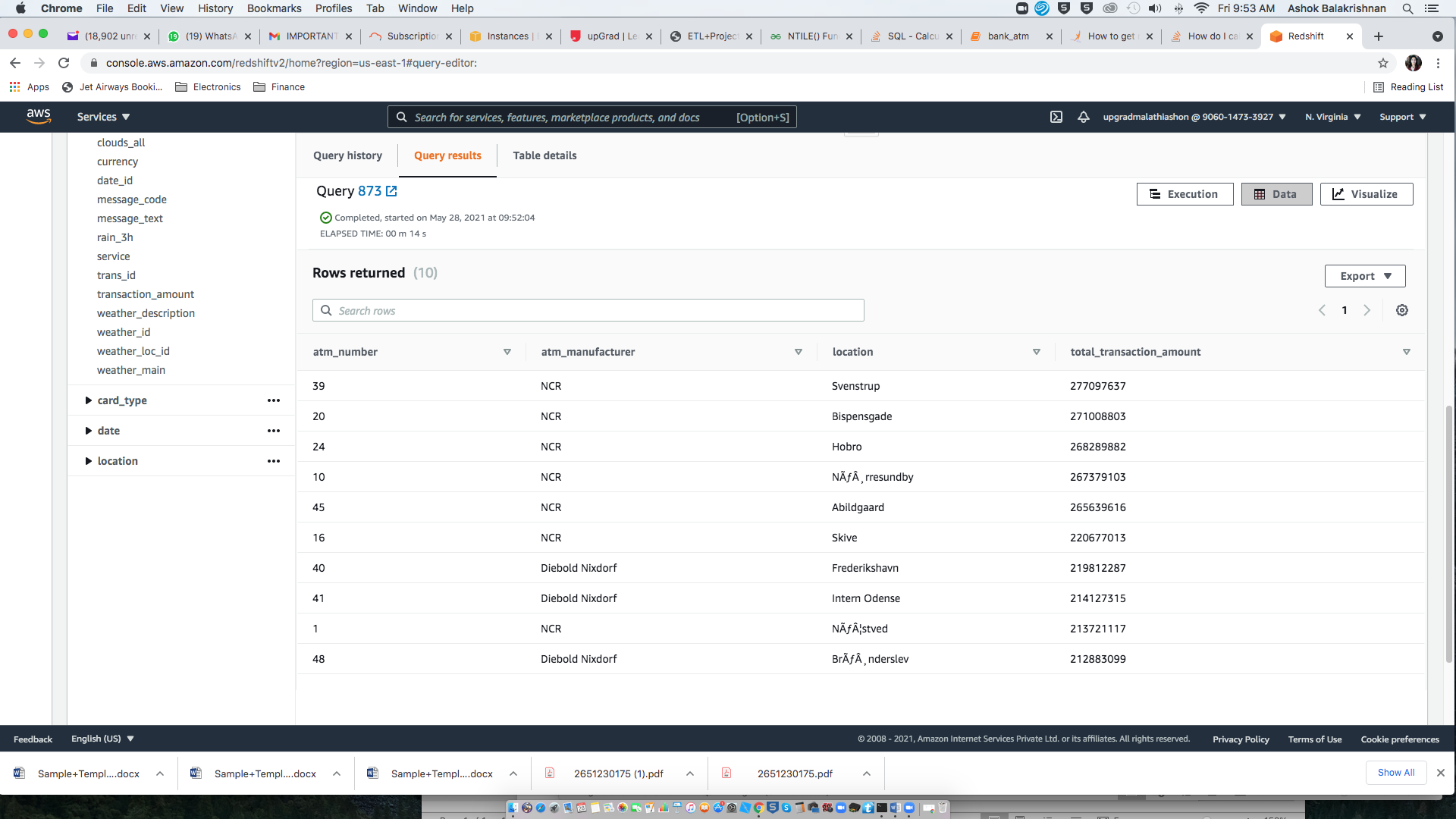
group by atm\_id

order by total\_transaction\_amount desc limit 10) Q, bankatm\_schema.atm R,

bankatm\_schema.location S

where Q.atm\_id = R.atm\_id and R.atm\_location\_id = S.location\_id

order by total\_transaction\_amount desc;



1. **Number of failed ATM transactions across various card types**

select Q.card\_type, count(P.trans\_id) total\_transaction\_count,

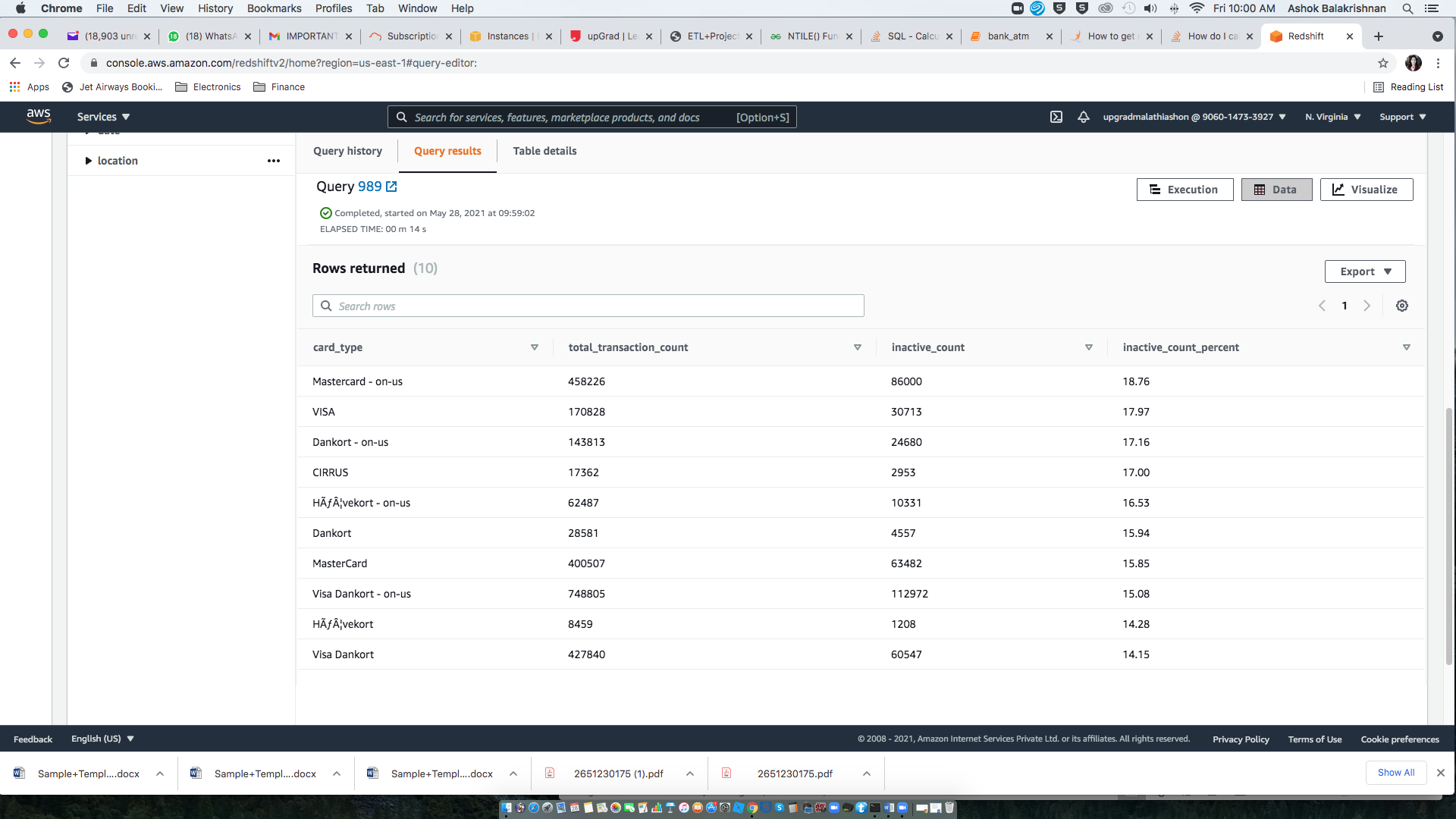
sum(case when P.atm\_status = 'Inactive' then 1 else 0 end) inactive\_count, CAST(inactive\_count\*100.0/total\_transaction\_count as decimal(5,2)) as inactive\_count\_percent

from bankatm\_schema.atm\_trans P, bankatm\_schema.card\_type Q

where P.card\_type\_id = Q.card\_type\_id

group by Q.card\_type

order by inactive\_count\_percent desc limit 10



1. **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 C.atm\_number, C.atm\_manufacturer, D.location, ( case when B.weekday = 'Saturday' or B.weekday = 'Sunday' then 1 else 0 end ) Weekend\_flag, count(trans\_id) transaction\_count

from bankatm\_schema.atm\_trans A, bankatm\_schema.date B, bankatm\_schema.atm C, bankatm\_schema.location D

where A.date\_id = B.date\_id and B.weekday in ('Saturday','Sunday') and A.atm\_id = C.atm\_id and A.weather\_loc\_id = D.location\_id

group by C.atm\_number, C.atm\_manufacturer, D.location, Weekend\_flag

order by C.atm\_number)

union

(select C.atm\_number, C.atm\_manufacturer, D.location, ( case when B.weekday != 'Saturday' or B.weekday != 'Sunday' then 0 else 1 end ) Weekend\_flag, count(trans\_id) transaction\_count

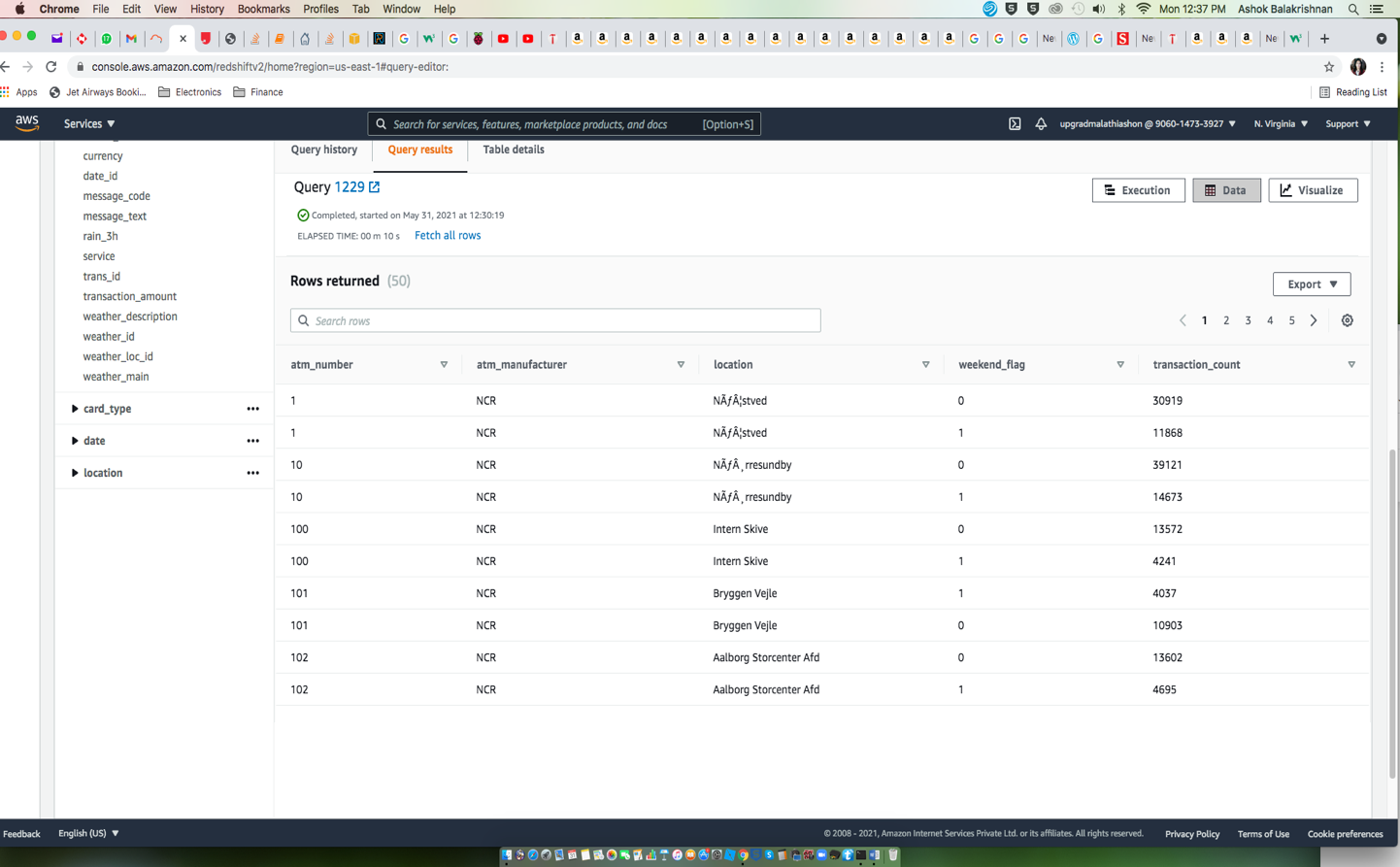
from bankatm\_schema.atm\_trans A, bankatm\_schema.date B, bankatm\_schema.atm C, bankatm\_schema.location D

where A.date\_id = B.date\_id and B.weekday not in ('Saturday','Sunday') and A.atm\_id = C.atm\_id and A.weather\_loc\_id = D.location\_id

group by C.atm\_number, C.atm\_manufacturer, D.location, Weekend\_flag

order by C.atm\_number)

order by atm\_number



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

WITH summary AS

(select Q.atm\_number, Q.atm\_manufacturer, R.location, S.weekday, count(\*) as total\_transaction\_count,

ROW\_NUMBER() OVER(PARTITION BY Q.atm\_number

ORDER BY total\_transaction\_count DESC) AS rank

from bankatm\_schema.atm\_trans P, bankatm\_schema.atm Q, bankatm\_schema.location R, bankatm\_schema.date S

where P.atm\_id=Q.atm\_id and R.location = 'Vejgaard' and R.location\_id = P.weather\_loc\_id and P.date\_id = S.date\_id

group by Q.atm\_number, Q.atm\_manufacturer, R.location, S.weekday

order by Q.atm\_number desc

)

SELECT atm\_number, atm\_manufacturer, location, weekday, total\_transaction\_count

FROM summary

WHERE rank = 1

order by total\_transaction\_count asc

