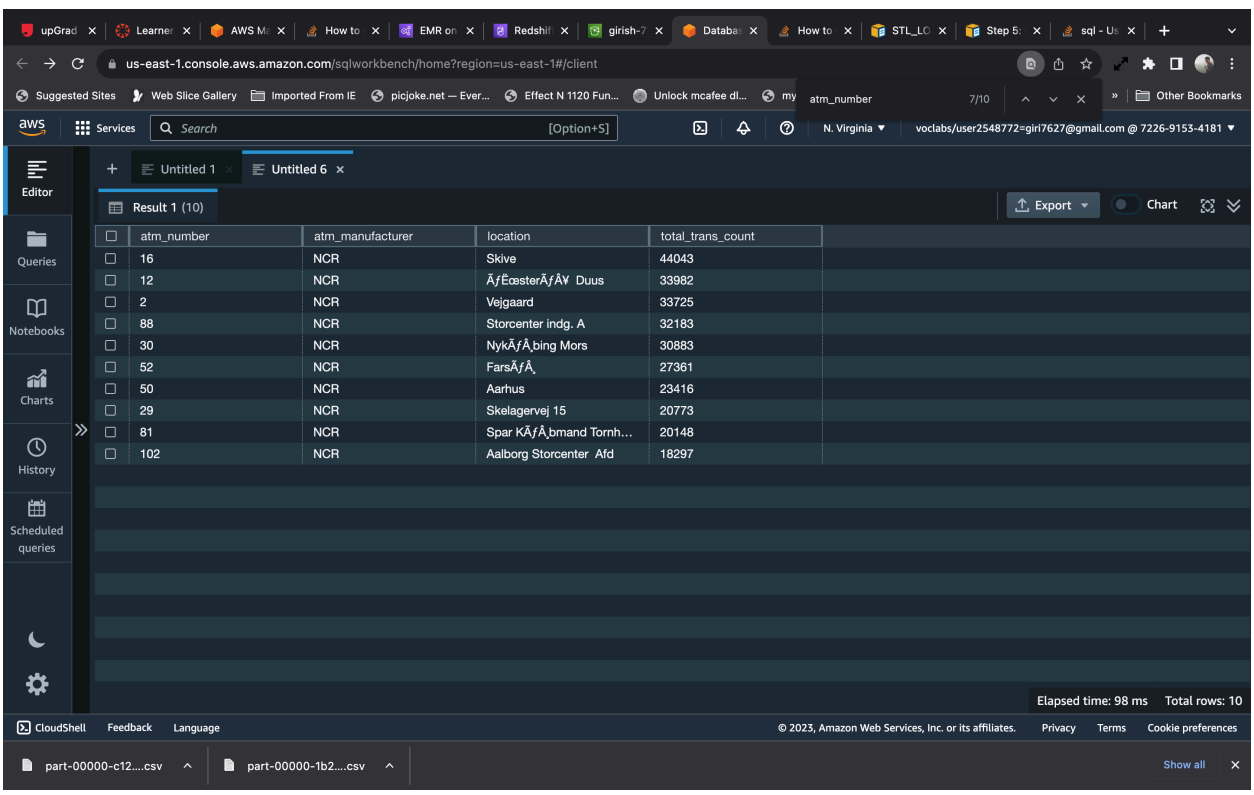
Solving analytical queries on Redshift Cluster

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 atm_number, atm_manufacturer,location, count(trans_id) as
total_trans_count from atm_data.fact_atm_trans as fact_atm_trans
inner join atm_data.dim_atm as dim_atm using ("atm_id")
inner join atm_data.dim_location as  dim_location
on dim_atm.atm_location_id=dim_location.location_id
where atm_status='Inactive'
group by atm_number, atm_manufacturer,location
order by count(trans_id) desc limit 10
```



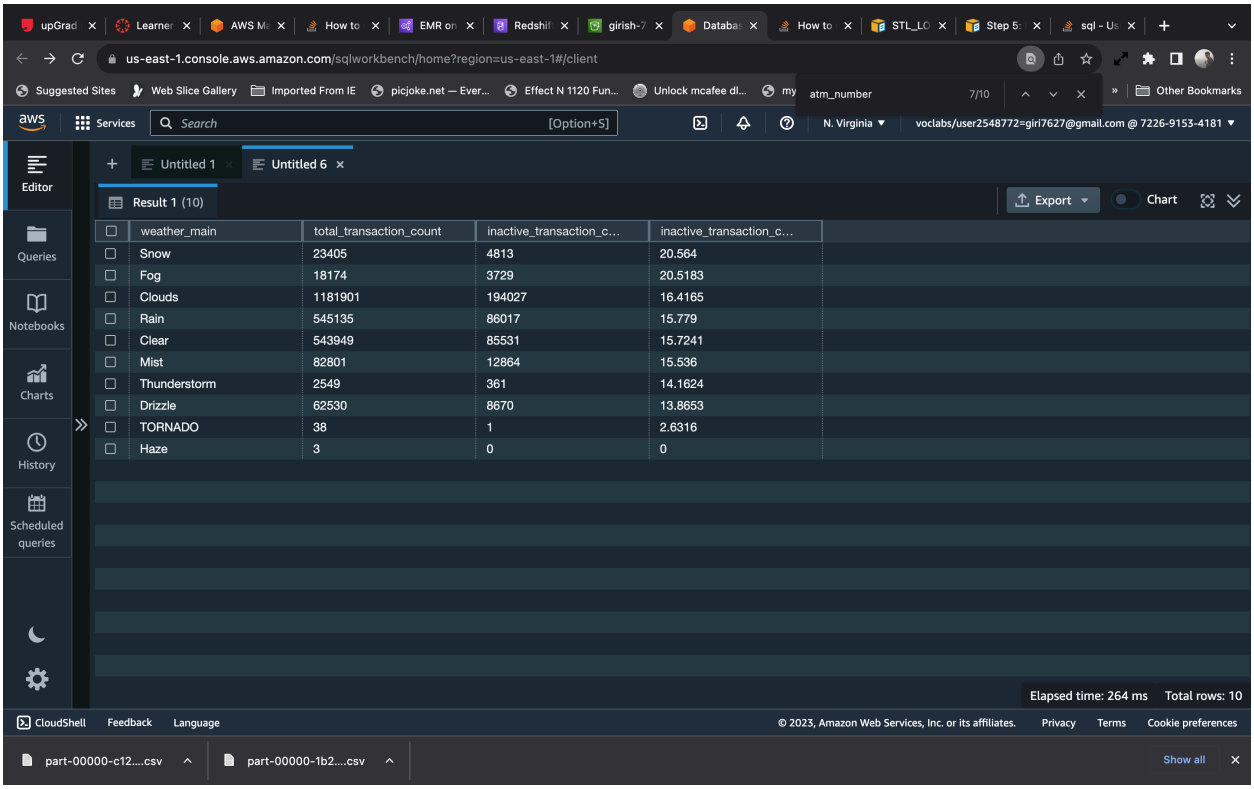
Result 1 (10)

atm_number	atm_manufacturer	location	total_trans_count
16	NCR	Skive	44043
12	NCR	ÃƒÈœsterÃƒÂ¥ Duus	33982
2	NCR	Vejgaard	33725
88	NCR	Storcenter indg. A	32183
30	NCR	NykÃƒÂ¸bing Mors	30883
52	NCR	FarsÃƒÂ¸	27361
50	NCR	Aarhus	23416
29	NCR	Skelagervej 15	20773
81	NCR	Spar KÃƒÂ¸bmand Tornh...	20148
102	NCR	Aalborg Storcenter Afd	18297

Elapsed time: 98 ms Total rows: 10

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

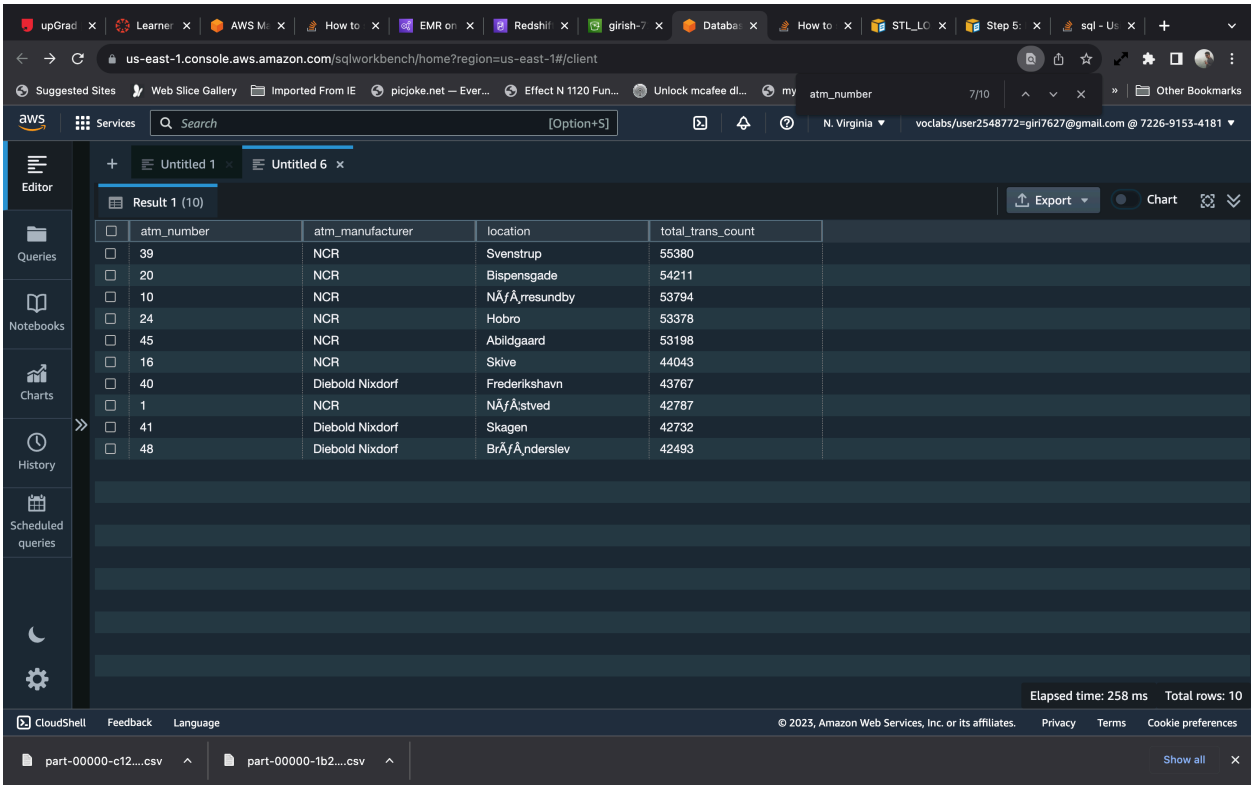
```
with active_counts as
(select weather_main, count(trans_id) as total_transaction_count from
atm_data.fact_atm_trans group by weather_main),
inactive_counts as
(select weather_main, count(trans_id) as inactive_transaction_count from
atm_data.fact_atm_trans where atm_status='Inactive' group by weather_main)
select weather_main,  total_transaction_count,
case when inactive_transaction_count is not null then
inactive_transaction_count else 0 end as inactive_transaction_count,
round(case when inactive_transaction_count is not null then
(inactive_transaction_count*1.0000/total_transaction_count)*100.0000 else
0.0000 end, 4)
as inactive_transaction_count_percent from
active_counts left outer join inactive_counts using (weather_main)
where weather_main!=''
order by case when inactive_transaction_count is not null then
(inactive_transaction_count*1.0000/total_transaction_count)*100.0000 else
0.0000 end  desc;
```



weather_main	total_transaction_count	inactive_transaction_c...	inactive_transaction_c...
Snow	23405	4813	20.564
Fog	18174	3729	20.5183
Clouds	1181901	194027	16.4165
Rain	545135	86017	15.779
Clear	543949	85531	15.7241
Mist	82801	12864	15.536
Thunderstorm	2549	361	14.1624
Drizzle	62530	8670	13.8653
TORNADO	38	1	2.6316
Haze	3	0	0

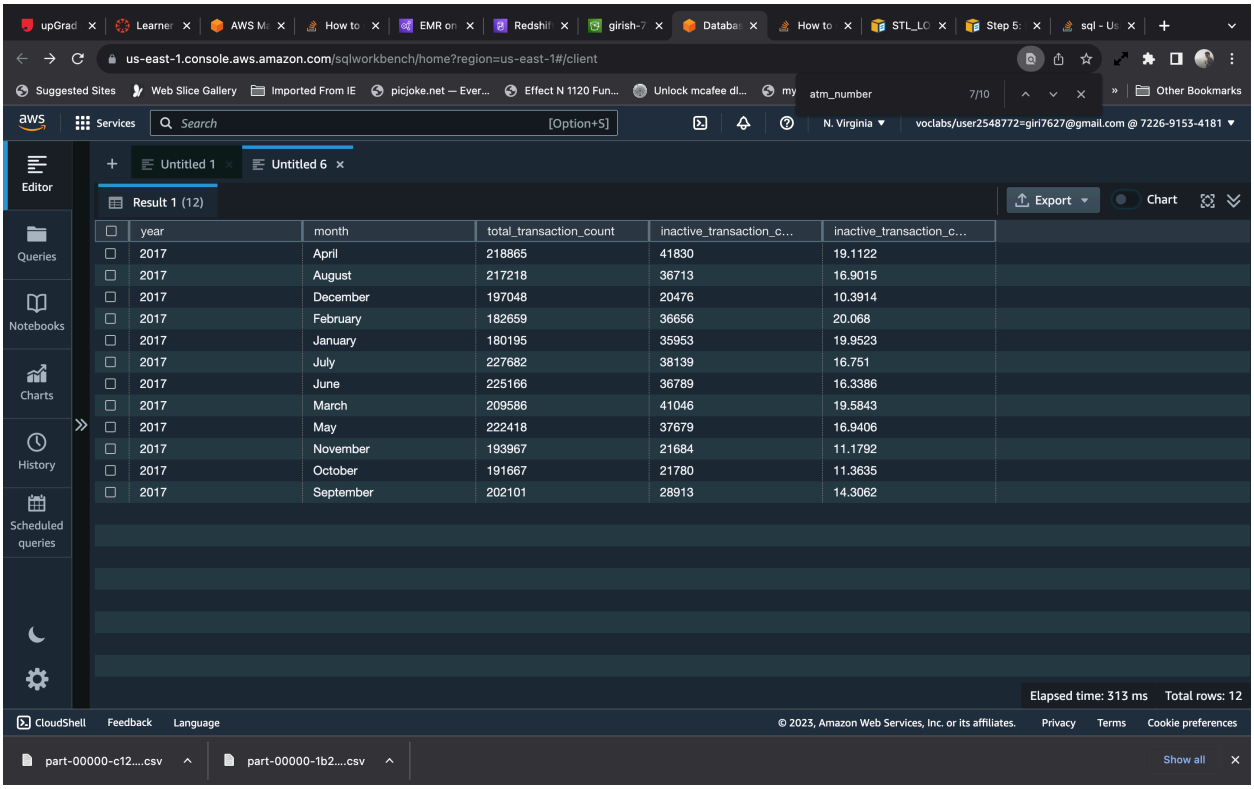
3. Top 10 ATMs with the most number of transactions throughout the year

```
select atm_number, atm_manufacturer,location, count(trans_id) as
total_trans_count from atm_data.fact_atm_trans as fact_atm_trans
inner join atm_data.dim_atm as dim_atm using ("atm_id")
inner join atm_data.dim_location as  dim_location
on dim_atm.atm_location_id=dim_location.location_id
group by atm_number, atm_manufacturer,location
order by count(trans_id) desc limit 10
```



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

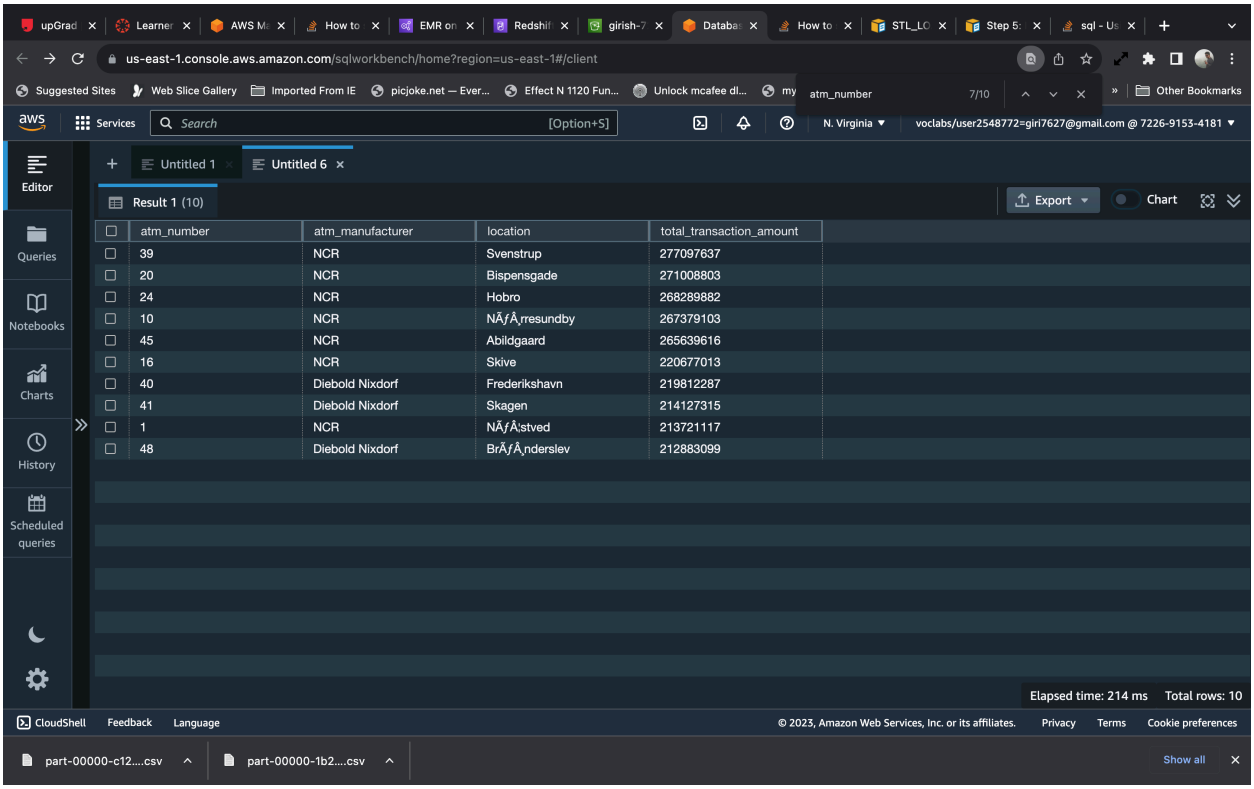
```
with active_counts as
(select year, month, count(trans_id) as total_transaction_count from
atm_data.fact_atm_trans inner join atm_data.dim_date
using ("date_id") group by year, month),
inactive_counts as
(select year, month, count(trans_id) as inactive_transaction_count from
atm_data.fact_atm_trans inner join atm_data.dim_date
using ("date_id") where atm_status='Inactive' group by year, month)
select year, month, total_transaction_count,
case when inactive_transaction_count is not null then
inactive_transaction_count else 0 end as inactive_transaction_count,
round(case when inactive_transaction_count is not null then
(inactive_transaction_count*1.0000/total_transaction_count)*100.0000 else
0.0000 end, 4)
as inactive_transaction_count_percent from
active_counts left outer join inactive_counts using (year, month)
order by month asc;
```



year	month	total_transaction_count	inactive_transaction_c...	inactive_transaction_c...
2017	April	218865	41830	19.1122
2017	August	217218	36713	16.9015
2017	December	197048	20476	10.3914
2017	February	182659	36656	20.068
2017	January	180195	35953	19.9523
2017	July	227682	38139	16.751
2017	June	225166	36789	16.3386
2017	March	209586	41046	19.5843
2017	May	222418	37679	16.9406
2017	November	193967	21684	11.1792
2017	October	191667	21780	11.3635
2017	September	202101	28913	14.3062

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

```
select atm_number, atm_manufacturer,location, sum(transaction_amount) as
total_transaction_amount from atm_data.fact_atm_trans as fact_atm_trans
inner join atm_data.dim_atm as dim_atm using ("atm_id")
inner join atm_data.dim_location as  dim_location
on dim_atm.atm_location_id=dim_location.location_id
group by atm_number, atm_manufacturer,location
order by sum(transaction_amount) desc limit 10
```

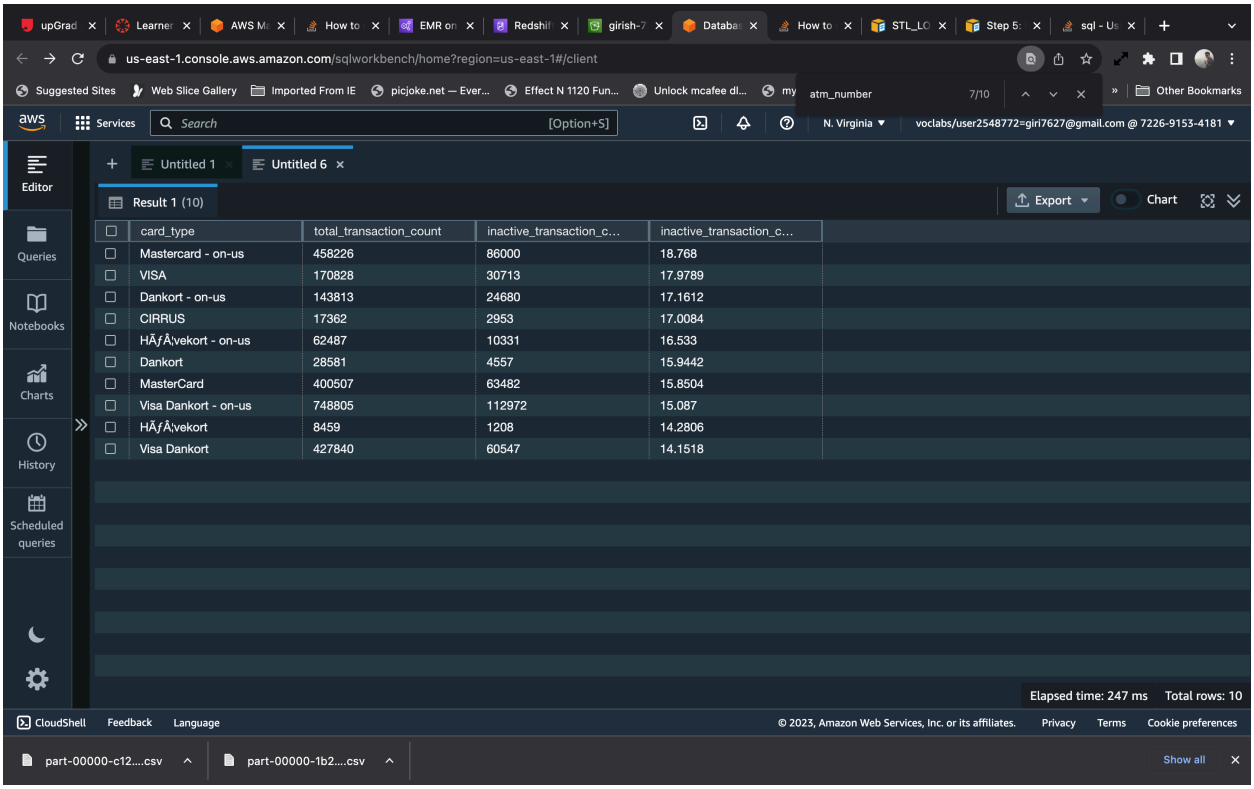


atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÃƒÂ¸resundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	NÃƒÂ¦stved	213721117
48	Diebold Nixdorf	BrÃƒÂ¸nderslev	212883099

Elapsed time: 214 ms Total rows: 10

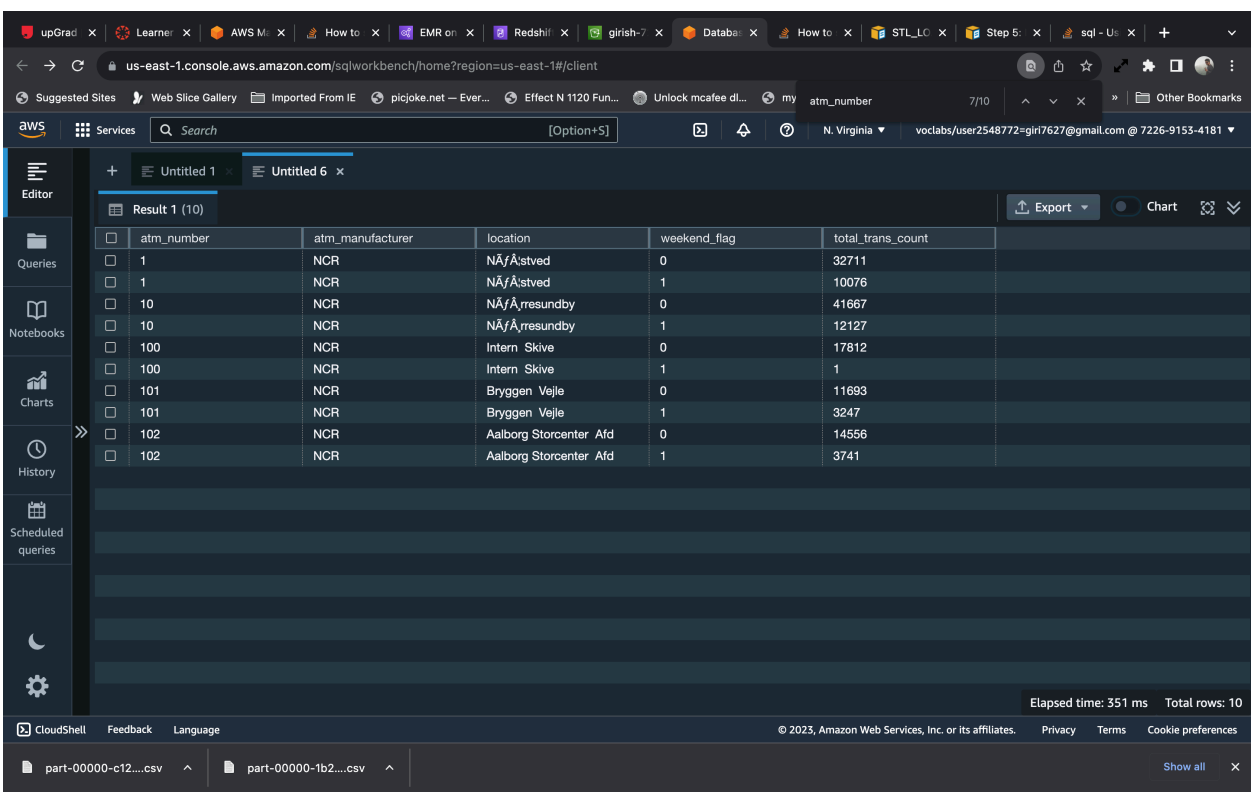
6. **Number of failed ATM transactions across various card type**

```
with active_counts as
(select card_type, count(trans_id) as total_transaction_count from
atm_data.fact_atm_trans inner join atm_data.dim_card_type
using (card_type_id) group by card_type),
inactive_counts as
(select card_type, count(trans_id) as inactive_transaction_count from
atm_data.fact_atm_trans inner join atm_data.dim_card_type
using (card_type_id) where atm_status='Inactive' group by card_type)
select card_type, total_transaction_count,
case when inactive_transaction_count is not null then
inactive_transaction_count else 0 end as inactive_transaction_count,
round(case when inactive_transaction_count is not null then
(inactive_transaction_count*1.0000/total_transaction_count)*100.0000 else
0.0000 end, 4)
as inactive_transaction_count_percent from
active_counts left outer join inactive_counts using (card_type)
order by round(case when inactive_transaction_count is not null then
(inactive_transaction_count*1.0000/total_transaction_count)*100.0000 else
0.0000 end, 4) desc limit 10;
```



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 atm_number, atm_manufacturer,location,
case when weekday in ('Sunday', 'Saturday') then 1 else 0 end as
weekend_flag,
count(trans_id) as total_trans_count from atm_data.fact_atm_trans as
fact_atm_trans
inner join atm_data.dim_atm as dim_atm using ("atm_id")
inner join atm_data.dim_location as  dim_location
on dim_atm.atm_location_id=dim_location.location_id
inner join atm_data.dim_date as dim_date using ("date_id")
group by atm_number, atm_manufacturer,location,
case when weekday in ('Sunday', 'Saturday') then 1 else 0 end
order by atm_number, atm_manufacturer,location,
case when weekday in ('Sunday', 'Saturday') then 1 else 0
end,count(trans_id)  asc limit 10
```



atm_number	atm_manufacturer	location	weekend_flag	total_trans_count
1	NCR	NÃƒÂ¦stved	0	32711
1	NCR	NÃƒÂ¦stved	1	10076
10	NCR	NÃƒÂ¸resundby	0	41667
10	NCR	NÃƒÂ¸resundby	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"**

```
with total_transactions_weekday as (
    select atm_number, atm_manufacturer,location, weekday,
count(trans_id) as total_trans_count
from atm_data.fact_atm_trans as fact_atm_trans
inner join atm_data.dim_atm as dim_atm using ("atm_id")
inner join atm_data.dim_location as  dim_location
on dim_atm.atm_location_id=dim_location.location_id
inner join atm_data.dim_date as dim_date using ("date_id")
where location='Vejgaard'
group by atm_number, atm_manufacturer,location, weekday
order by count(trans_id) desc),
max_transactions as (
    select atm_number, atm_manufacturer,location,
max(total_trans_count) as max_trans_count
from total_transactions_weekday group by atm_number,
atm_manufacturer,location)
select b.atm_number, b.atm_manufacturer, b.location, b.weekday,
b.total_trans_count
from max_transactions a inner join total_transactions_weekday b using
(atm_number, atm_manufacturer,location)
where total_trans_count=max_trans_count;
```