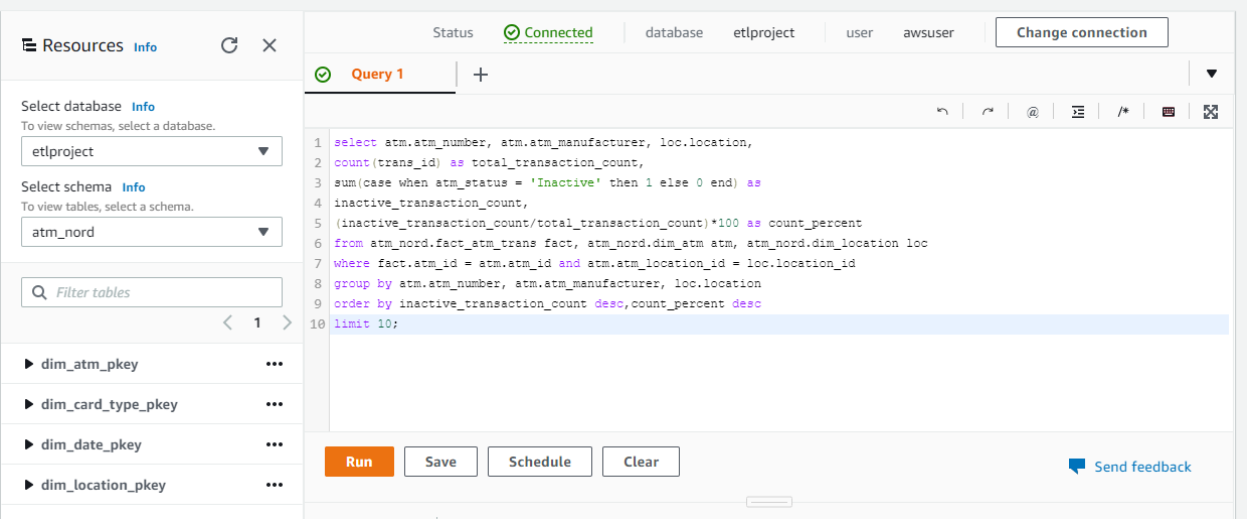
Solving analytical queries on Redshift Cluster

Queries 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.

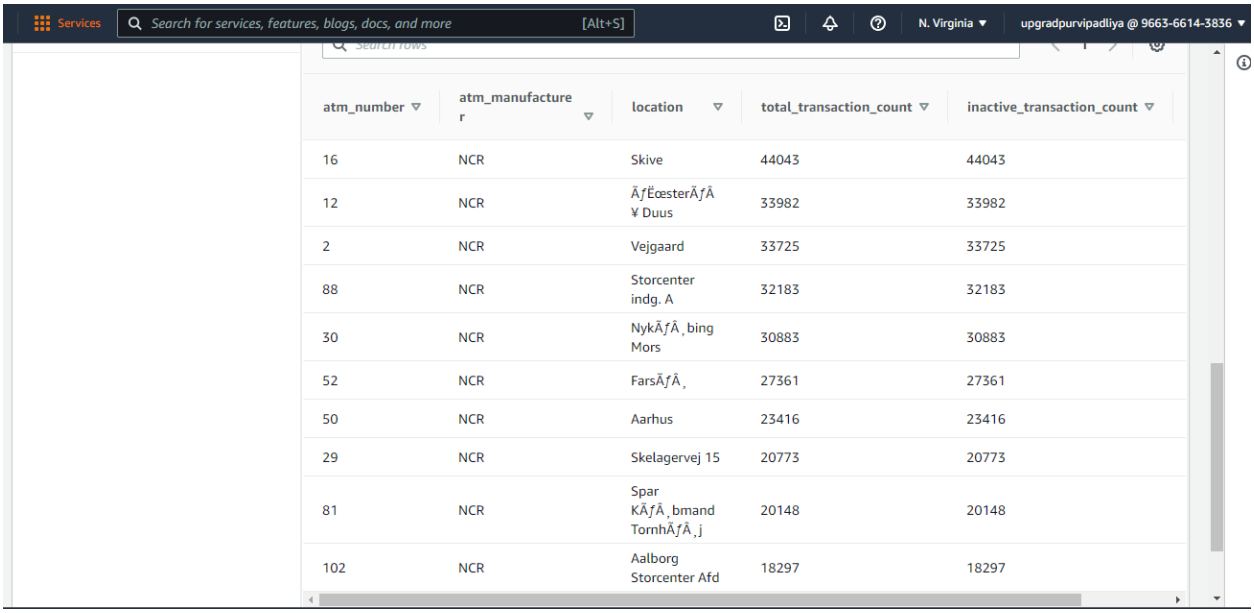
1. **Top 10 ATMs where most transactions are in the 'inactive' state**

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.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_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
group by atm.atm_number, atm.atm_manufacturer, loc.location
order by inactive_transaction_count desc,count_percent desc
limit 10;
```



Screenshot of the resultant table

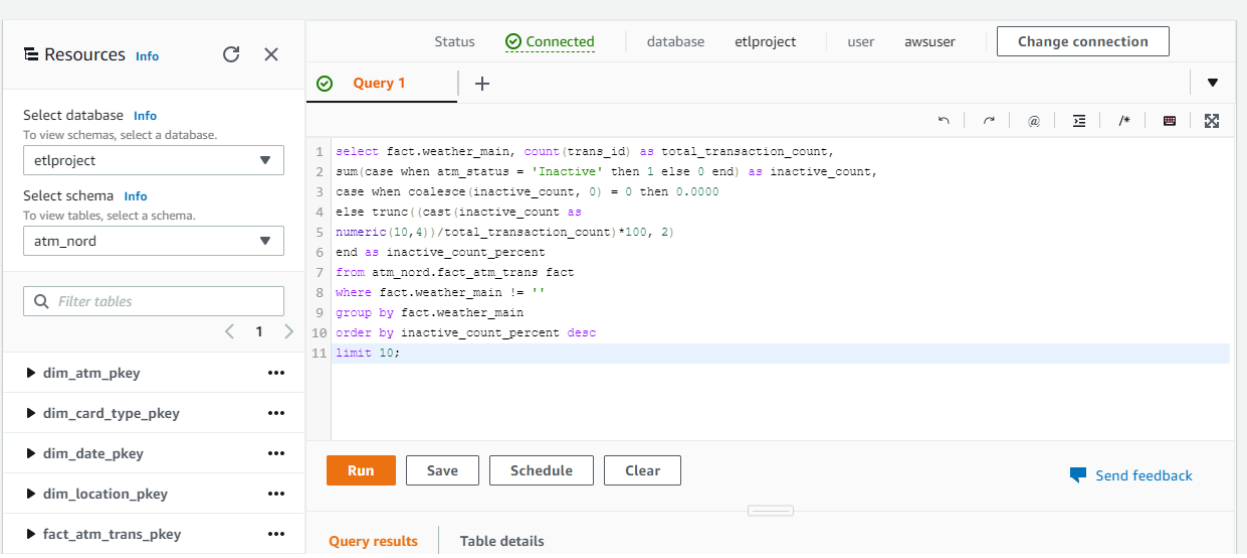


atm_number	atm_manufacturer	location	total_transaction_count	inactive_transaction_count
16	NCR	Skive	44043	44043
12	NCR	ÃƒÆ’Ã¦sterÃƒÂ¥ Duus	33982	33982
2	NCR	Vejgaard	33725	33725
88	NCR	Storcenter indg. A	32183	32183
30	NCR	NykÃƒÂ¸bing Mors	30883	30883
52	NCR	FarsÃƒÂ¸	27361	27361
50	NCR	Aarhus	23416	23416
29	NCR	Skelagervej 15	20773	20773
81	NCR	Spar KÃƒÂ¸bmand TornhÃƒÂ¸j	20148	20148
102	NCR	Aalborg Storcenter Afd	18297	18297

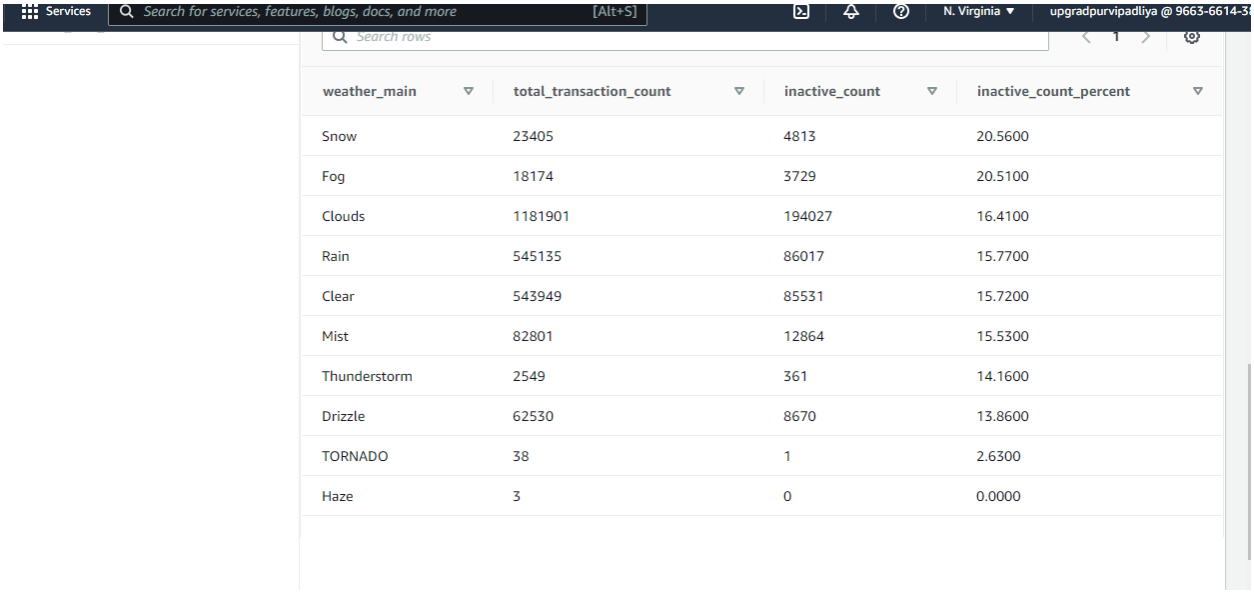
2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

Query:

```
select fact.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_nord.fact_atm_trans fact
where fact.weather_main != ""
group by fact.weather_main
order by inactive_count_percent desc
limit 10;
```



Screenshot of the resultant table:

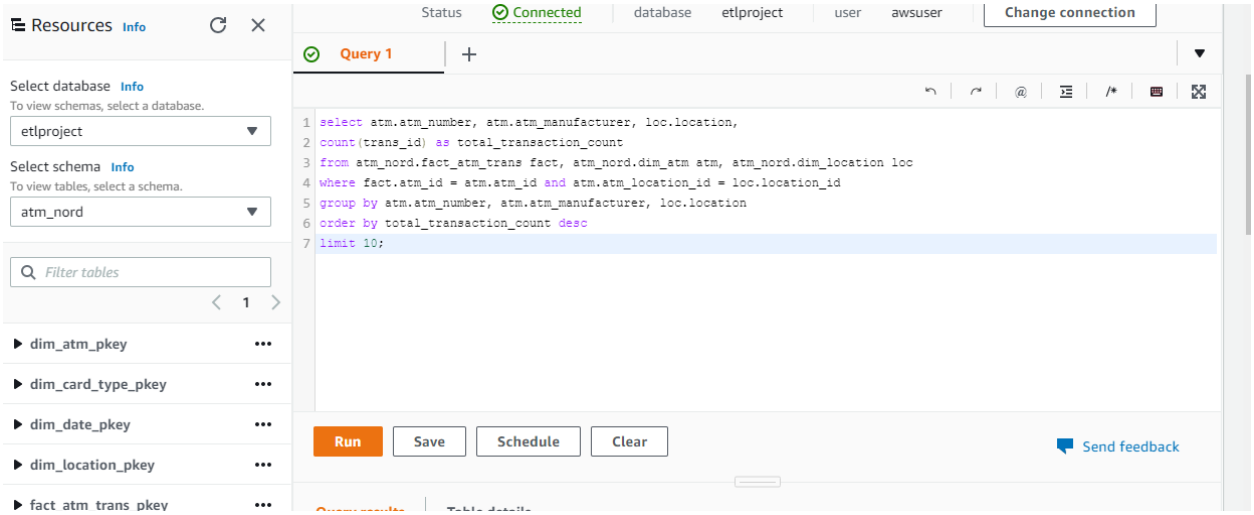


weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.5600
Fog	18174	3729	20.5100
Clouds	1181901	194027	16.4100
Rain	545135	86017	15.7700
Clear	543949	85531	15.7200
Mist	82801	12864	15.5300
Thunderstorm	2549	361	14.1600
Drizzle	62530	8670	13.8600
TORNADO	38	1	2.6300
Haze	3	0	0.0000

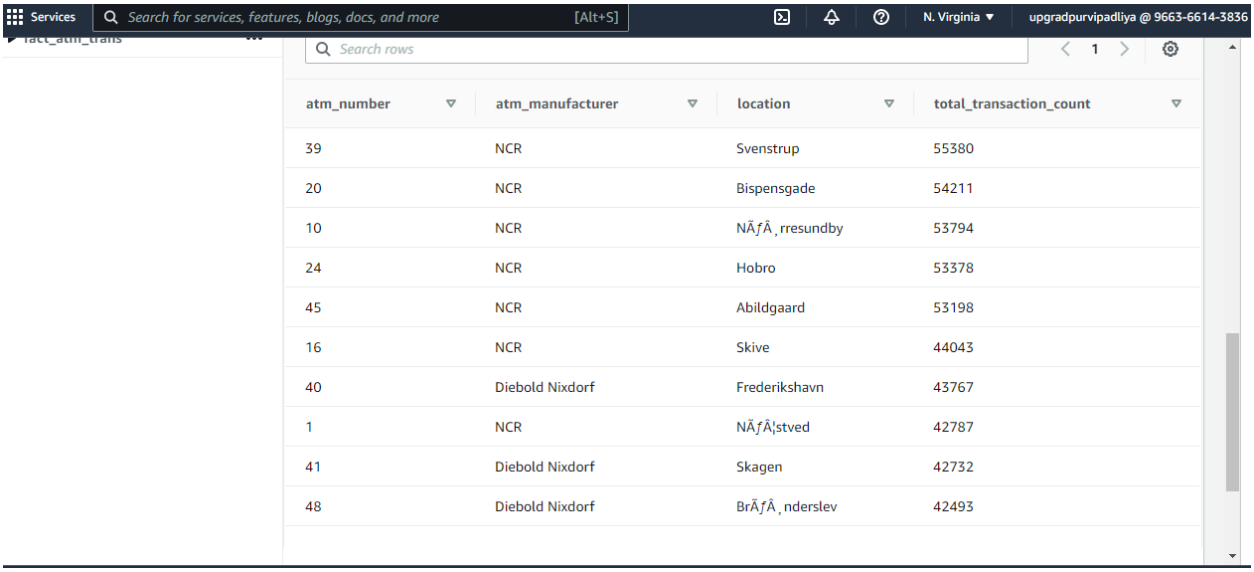
3. Top 10 ATMs with the most number of transactions throughout the year

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location,
count(trans_id) as total_transaction_count
from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
group by atm.atm_number, atm.atm_manufacturer, loc.location
order by total_transaction_count desc
limit 10;
```



Screenshot of the resultant table:

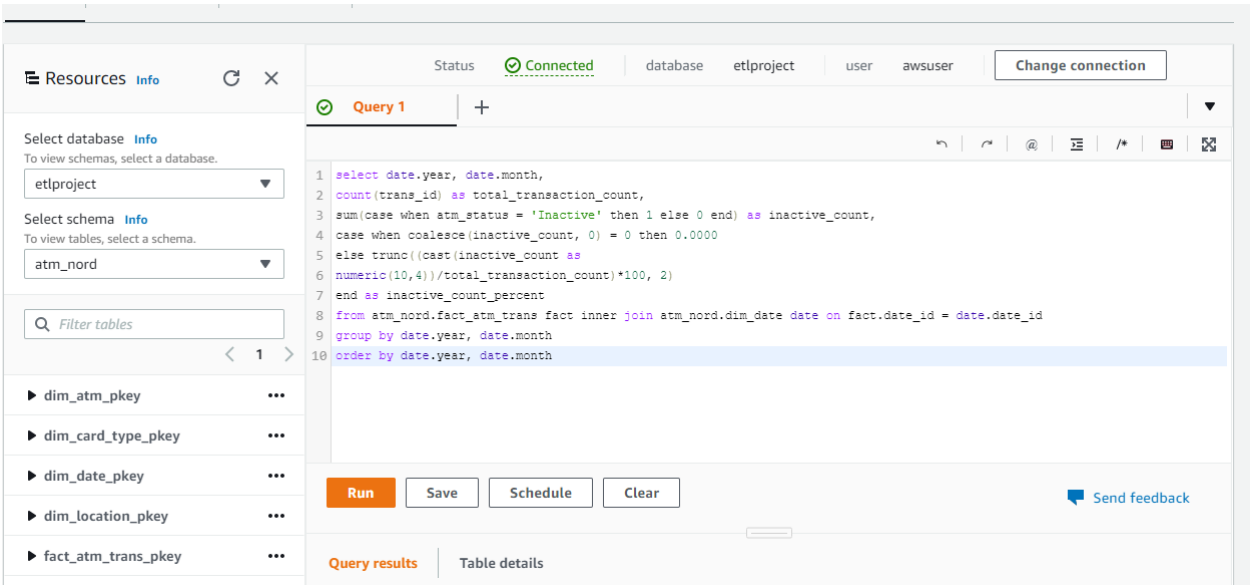


atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÃƒÂ¸rresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÃƒÂ¦stved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÃƒÂ¸nderslev	42493

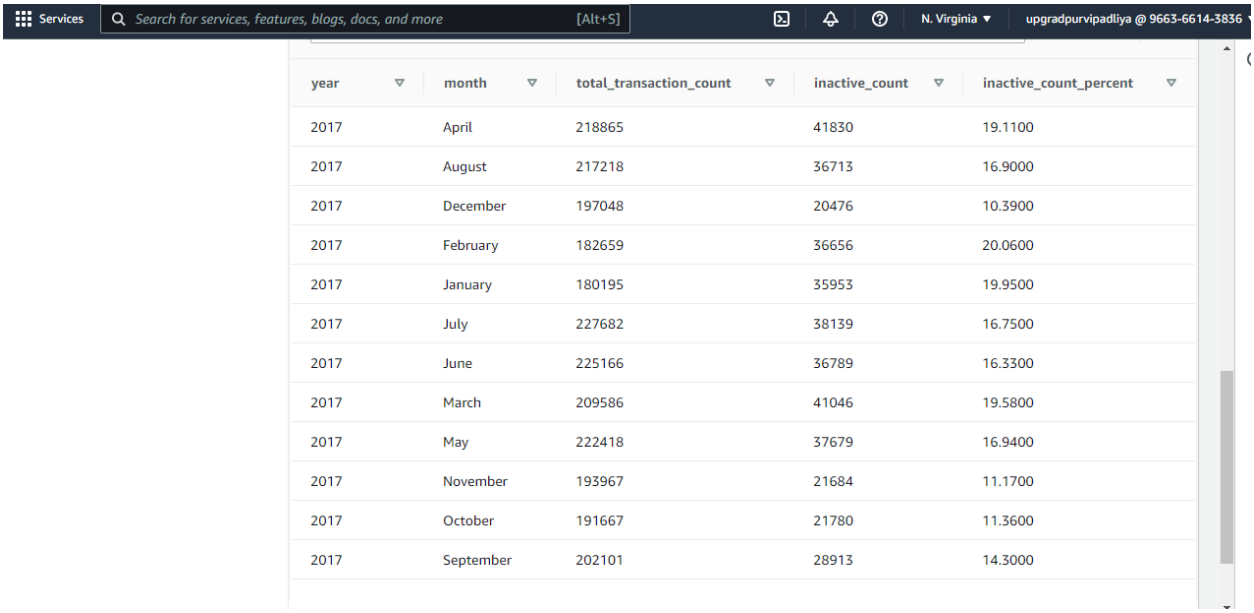
4. **Number of overall ATM transactions going inactive per month for each month**

Query:

```
select date.year, date.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_nord.fact_atm_trans fact inner join atm_nord.dim_date date on fact.date_id =
date.date_id
group by date.year, date.month
order by date.year, date.month
```



Screenshot of the resultant table:

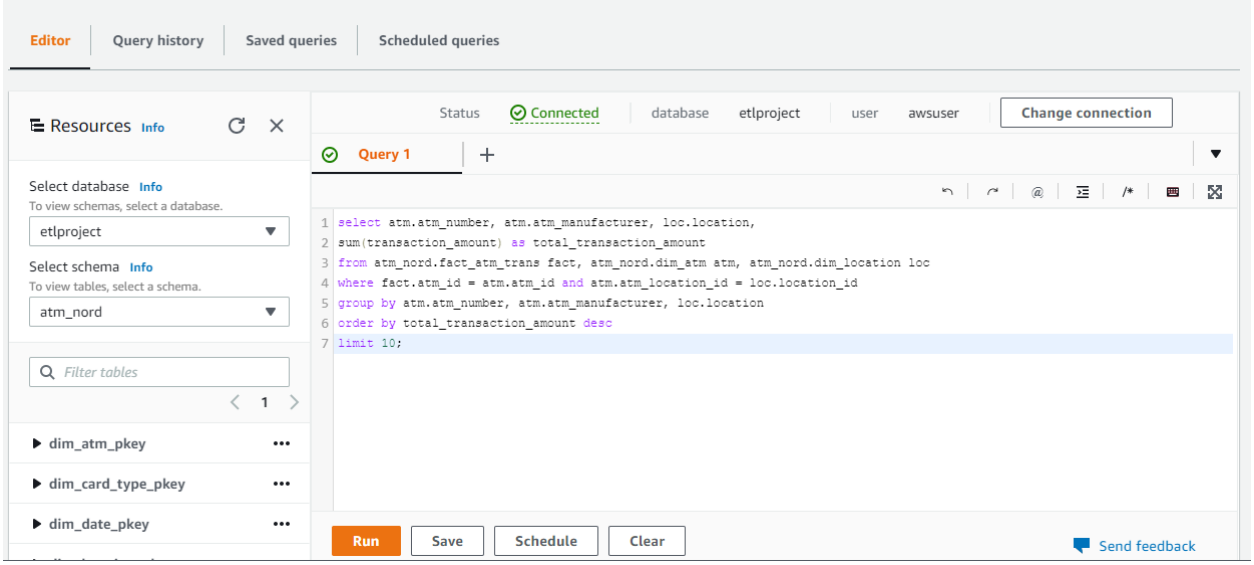


year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.1100
2017	August	217218	36713	16.9000
2017	December	197048	20476	10.3900
2017	February	182659	36656	20.0600
2017	January	180195	35953	19.9500
2017	July	227682	38139	16.7500
2017	June	225166	36789	16.3300
2017	March	209586	41046	19.5800
2017	May	222418	37679	16.9400
2017	November	193967	21684	11.1700
2017	October	191667	21780	11.3600
2017	September	202101	28913	14.3000

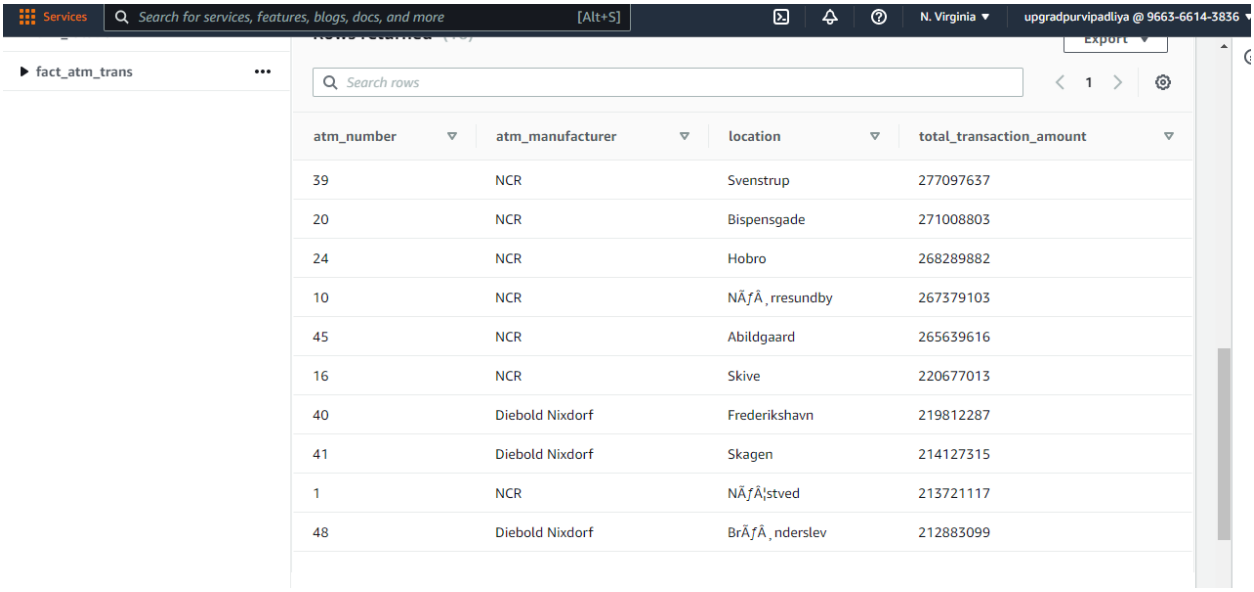
5. **Top 10 ATMs with the highest total withdrawn amount throughout the year**

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location,
sum(transaction_amount) as total_transaction_amount
from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
group by atm.atm_number, atm.atm_manufacturer, loc.location
order by total_transaction_amount desc
limit 10;
```



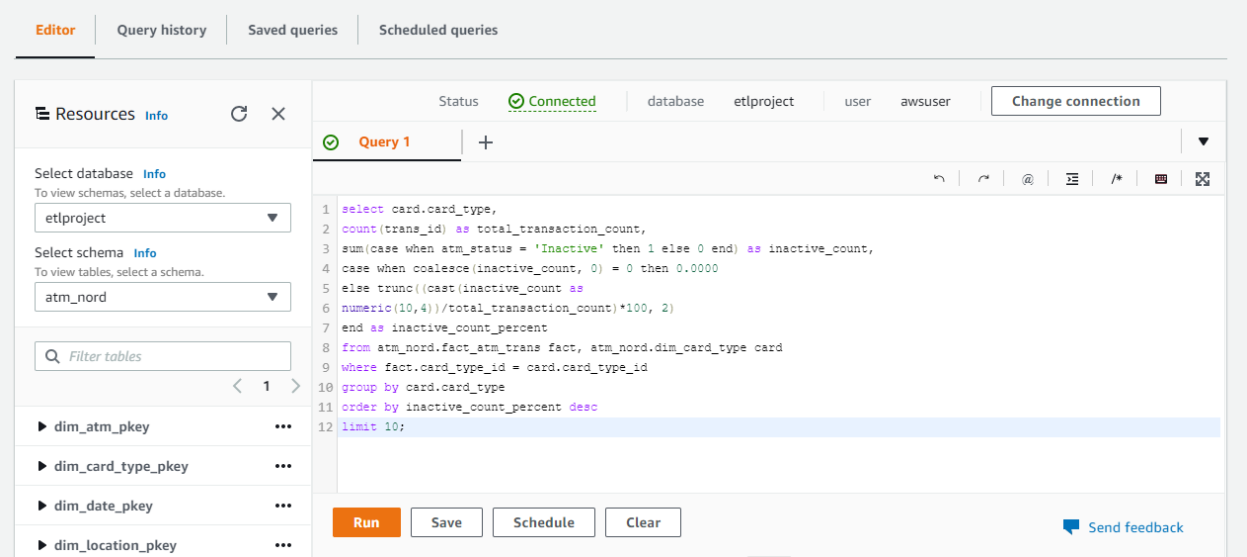
Screenshot of the resultant table:



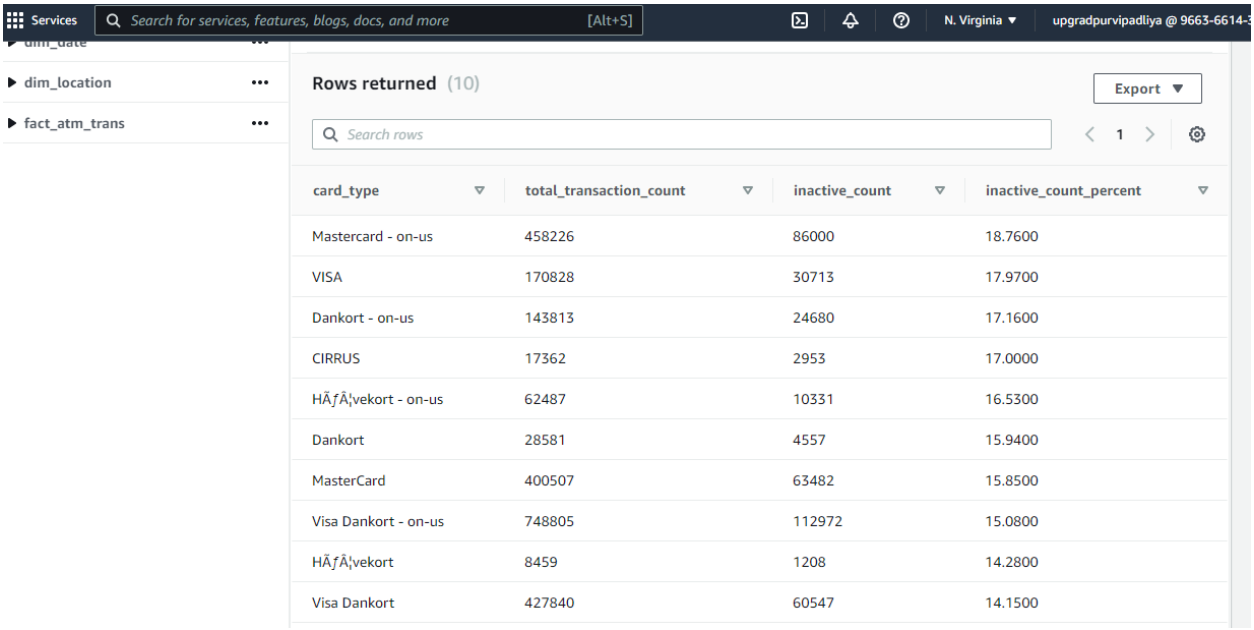
6. Number of failed ATM transactions across various card types

Query:

```
select card.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_nord.fact_atm_trans fact, atm_nord.dim_card_type card
where fact.card_type_id = card.card_type_id
group by card.card_type
order by inactive_count_percent desc
limit 10;
```



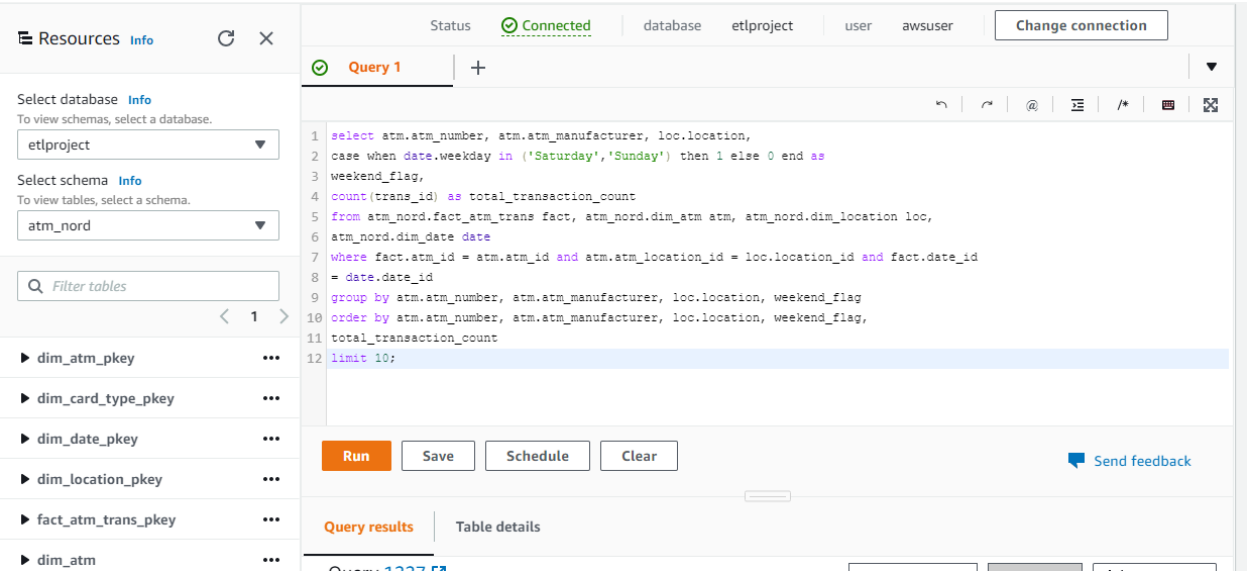
Screenshot of the resultant table:



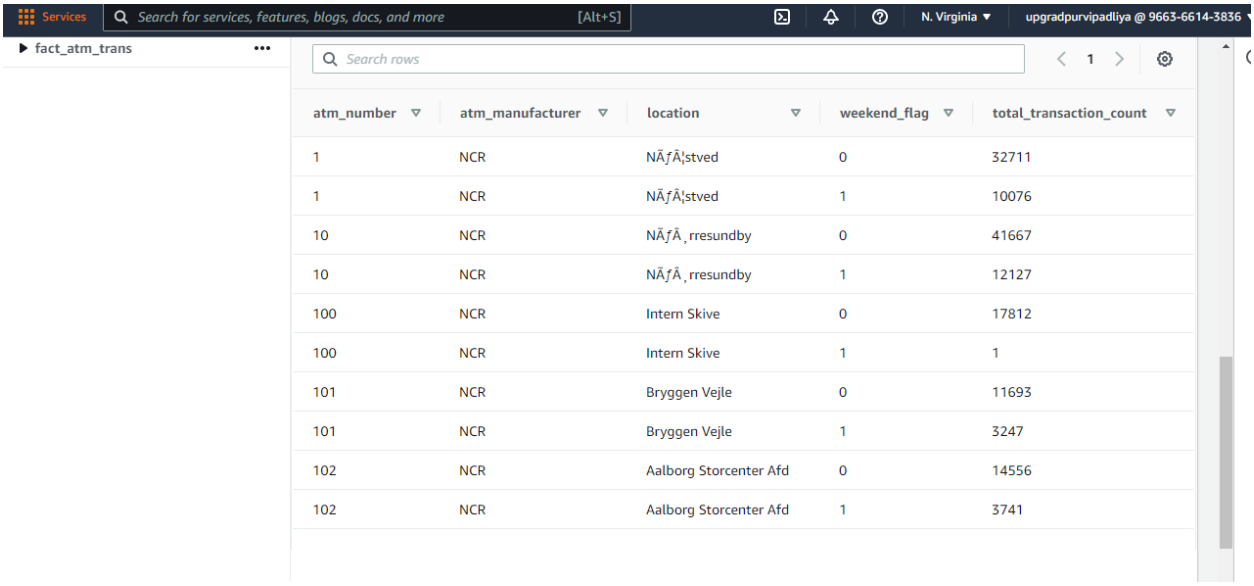
7. **Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order this by the ATM_number, ATM_manufacturer, location, weekend_flag and then total_transaction_count**

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location,
case when date.weekday in ('Saturday','Sunday') then 1 else 0 end as
weekend_flag,
count(trans_id) as total_transaction_count
from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc,
atm_nord.dim_date date
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id and fact.date_id
= date.date_id
group by atm.atm_number, atm.atm_manufacturer, loc.location, weekend_flag
order by atm.atm_number, atm.atm_manufacturer, loc.location, weekend_flag,
total_transaction_count
limit 10;
```



Screenshot of the resultant table:

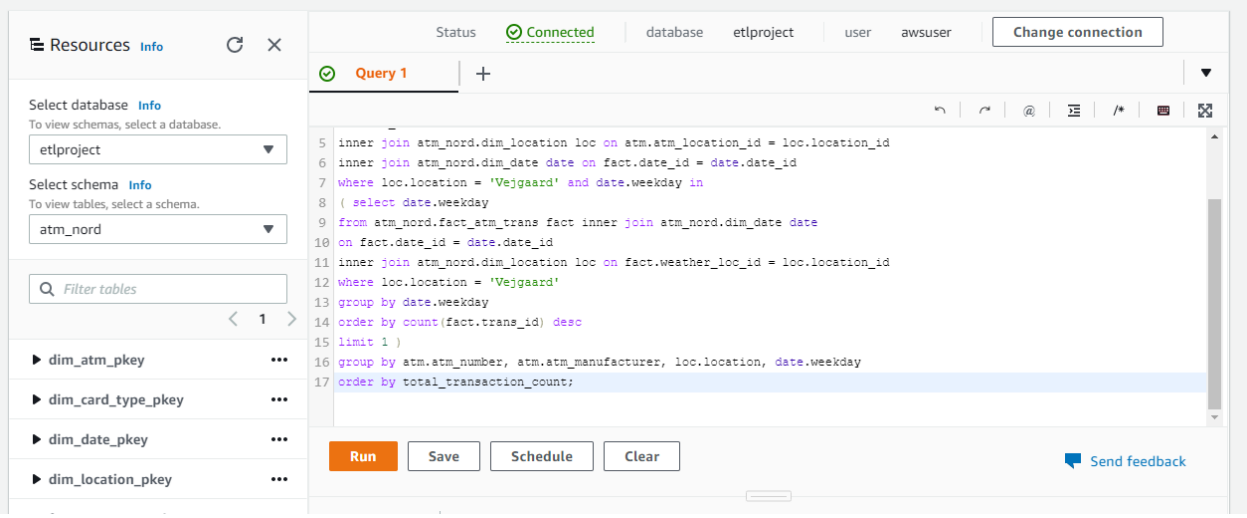


atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃƒÂ¦stved	0	32711
1	NCR	NÃƒÂ¦stved	1	10076
10	NCR	NÃƒÂ¸rresundby	0	41667
10	NCR	NÃƒÂ¸rresundby	1	12127
100	NCR	Intern Skive	0	17812
100	NCR	Intern Skive	1	1
101	NCR	Bryggen Vejle	0	11693
101	NCR	Bryggen Vejle	1	3247
102	NCR	Aalborg Storcenter Afd	0	14556
102	NCR	Aalborg Storcenter Afd	1	3741

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

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location, date.weekday,
count(trans_id) as total_transaction_count
from atm_nord.fact_atm_trans fact inner join atm_nord.dim_atm atm on fact.atm_id =
atm.atm_id
inner join atm_nord.dim_location loc on atm.atm_location_id = loc.location_id
inner join atm_nord.dim_date date on fact.date_id = date.date_id
where loc.location = 'Vejgaard' and date.weekday in
( select date.weekday
from atm_nord.fact_atm_trans fact inner join atm_nord.dim_date date
on fact.date_id = date.date_id
inner join atm_nord.dim_location loc on fact.weather_loc_id = loc.location_id
where loc.location = 'Vejgaard'
group by date.weekday
order by count(fact.trans_id) desc
limit 1 )
group by atm.atm_number, atm.atm_manufacturer, loc.location, date.weekday
order by total_transaction_count;
```



Screenshot of the resultant table: