Solving analytical queries on Redshift Cluster

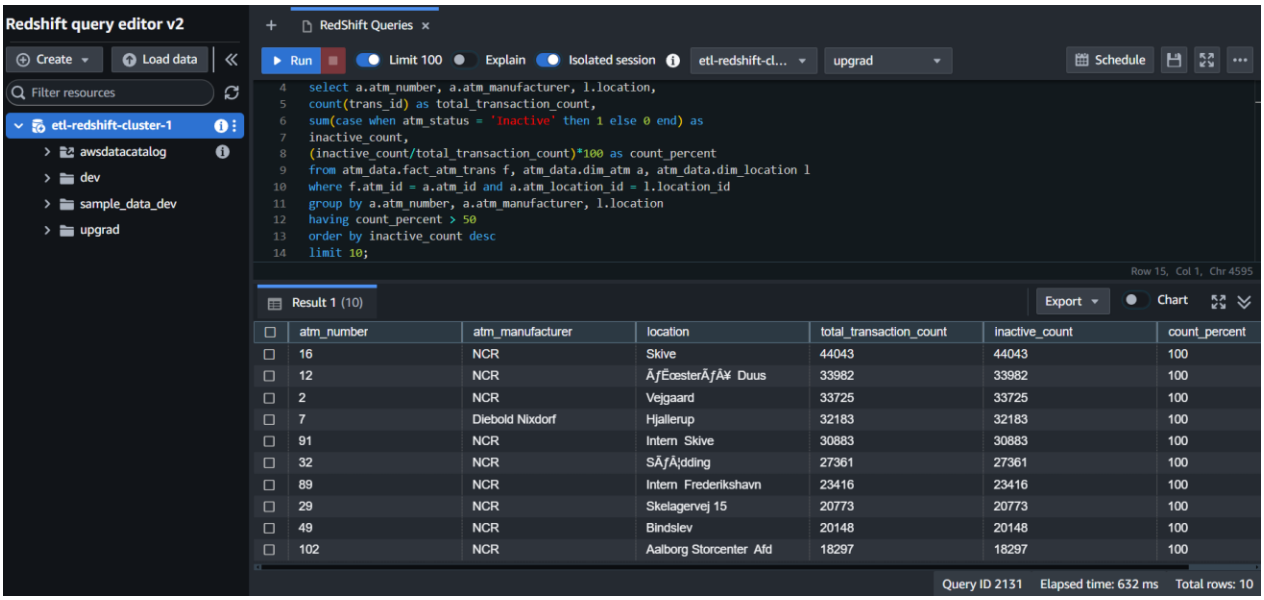
Here, you have to write the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS Redshift Query editor UI.

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

<Query>

```
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_count,
(inactive_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_count desc
limit 10;
```

<Screenshot of the resultant table>

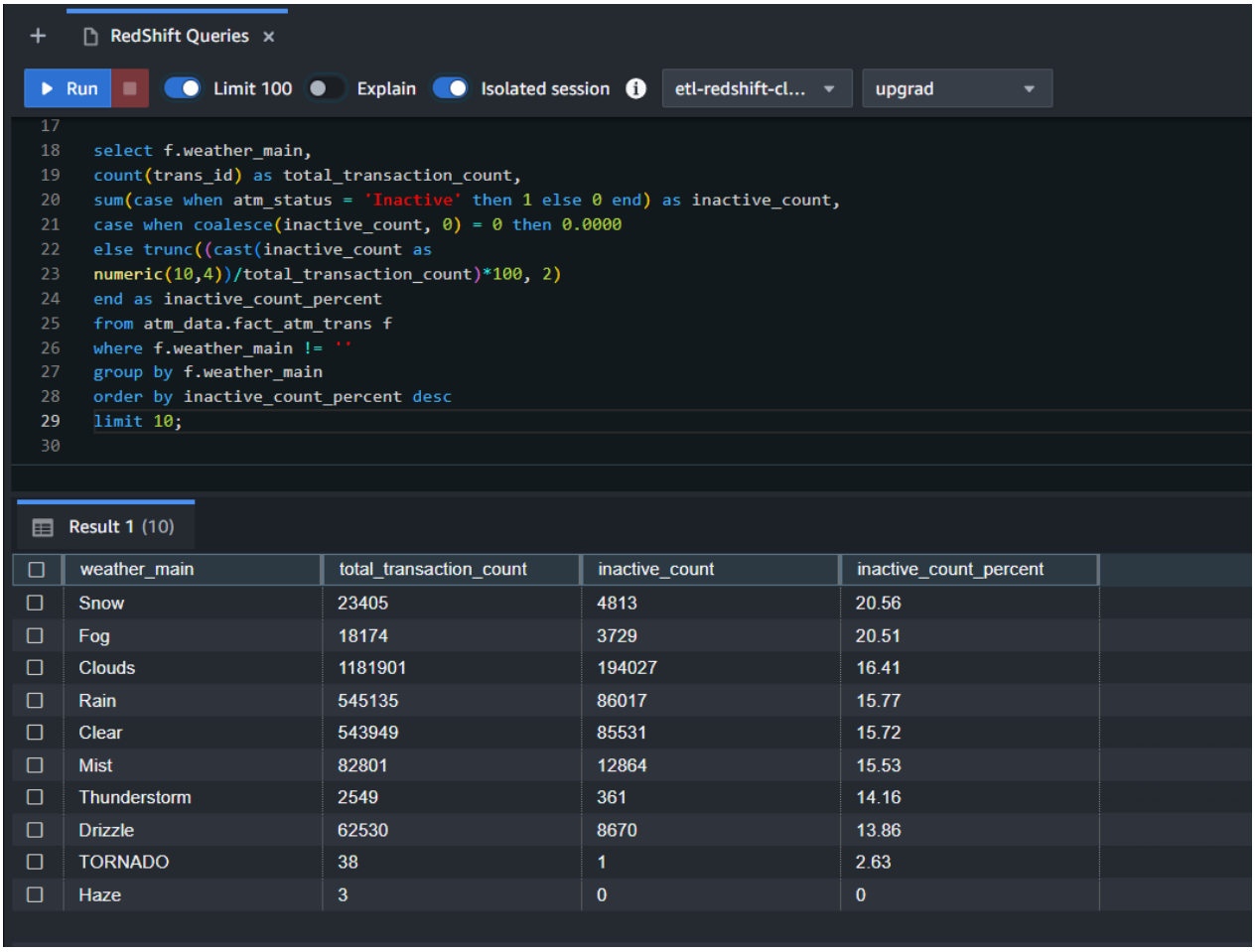


2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

<Query>

```
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;
```

<Screenshot of the resultant table>\



weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.56
Fog	18174	3729	20.51
Clouds	1181901	194027	16.41
Rain	545135	86017	15.77
Clear	543949	85531	15.72
Mist	82801	12864	15.53
Thunderstorm	2549	361	14.16
Drizzle	62530	8670	13.86
TORNADO	38	1	2.63
Haze	3	0	0

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

<Query>

```
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;
```

<Screenshot of the resultant table>

```
31  --Q3: Top 10 ATMs with the most number of transactions throughout the year
32
33  select a.atm_number, a.atm_manufacturer, l.location,
34  count(trans_id) as total_transaction_count
35  from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l
36  where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
37  group by a.atm_number, a.atm_manufacturer, l.location
38  order by total_transaction_count desc
39  limit 10;
40
41  --Q4: Number of overall ATM transactions going inactive per month for each month
```

Result 1 (10)

atm_number	atm_manufacturer	location	total_transaction_count
88	NCR	Storcenter indg. A	55380
20	NCR	Bispensgade	54211
10	NCR	NÃƒÂ¸rresundby	53794
90	NCR	HolbÃƒÂ¦k	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
53	NCR	SÃƒÂ¦by Syd	43767
1	NCR	NÃƒÂ¦stved	42787
88	NCR	Aalborg Storcenter Afd	42732
39	NCR	Svenstrup	42493

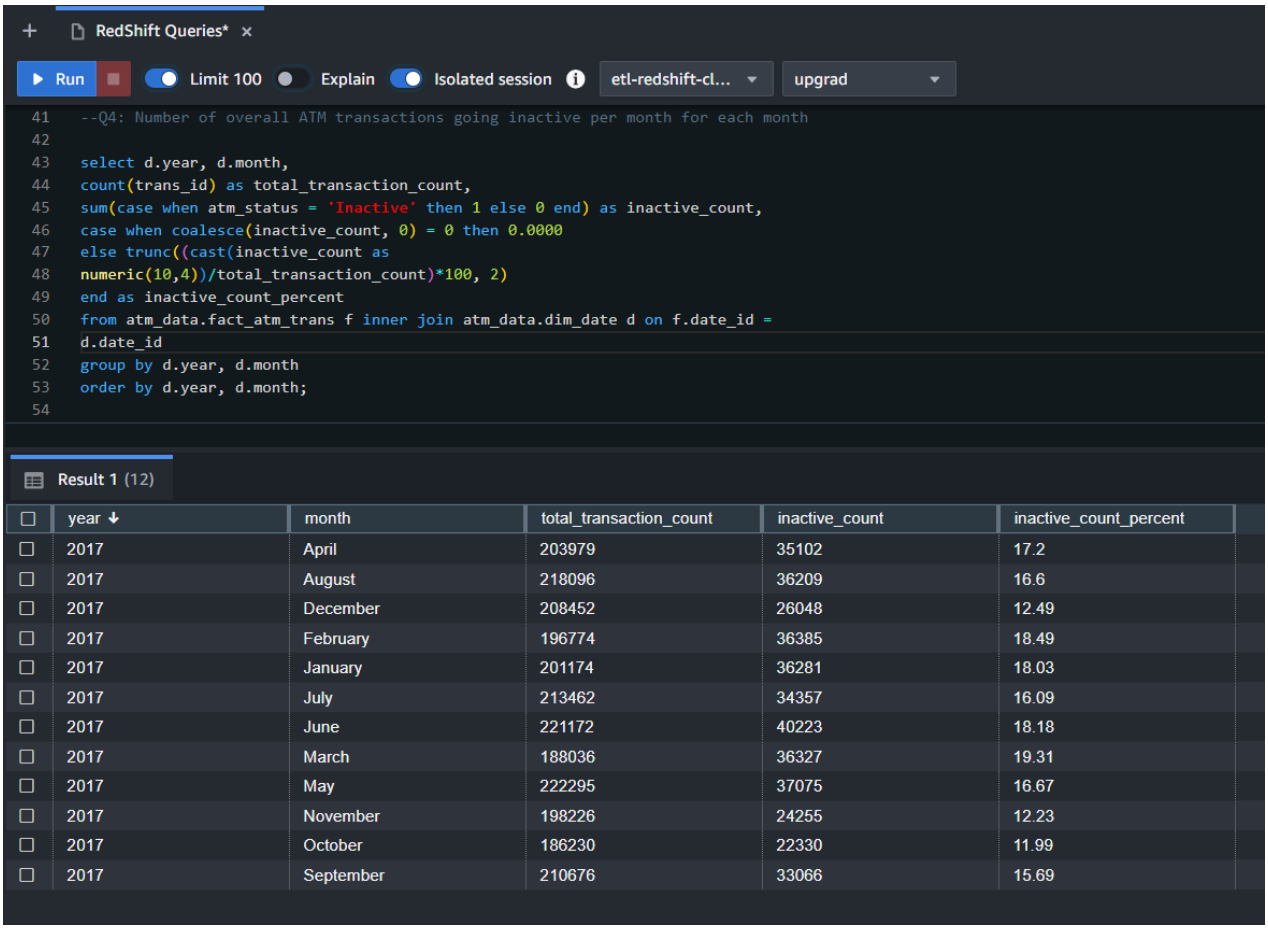
4. Number of overall ATM transactions going inactive per month for each month

<Query>

```
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
```

<Screenshot of the resultant table>



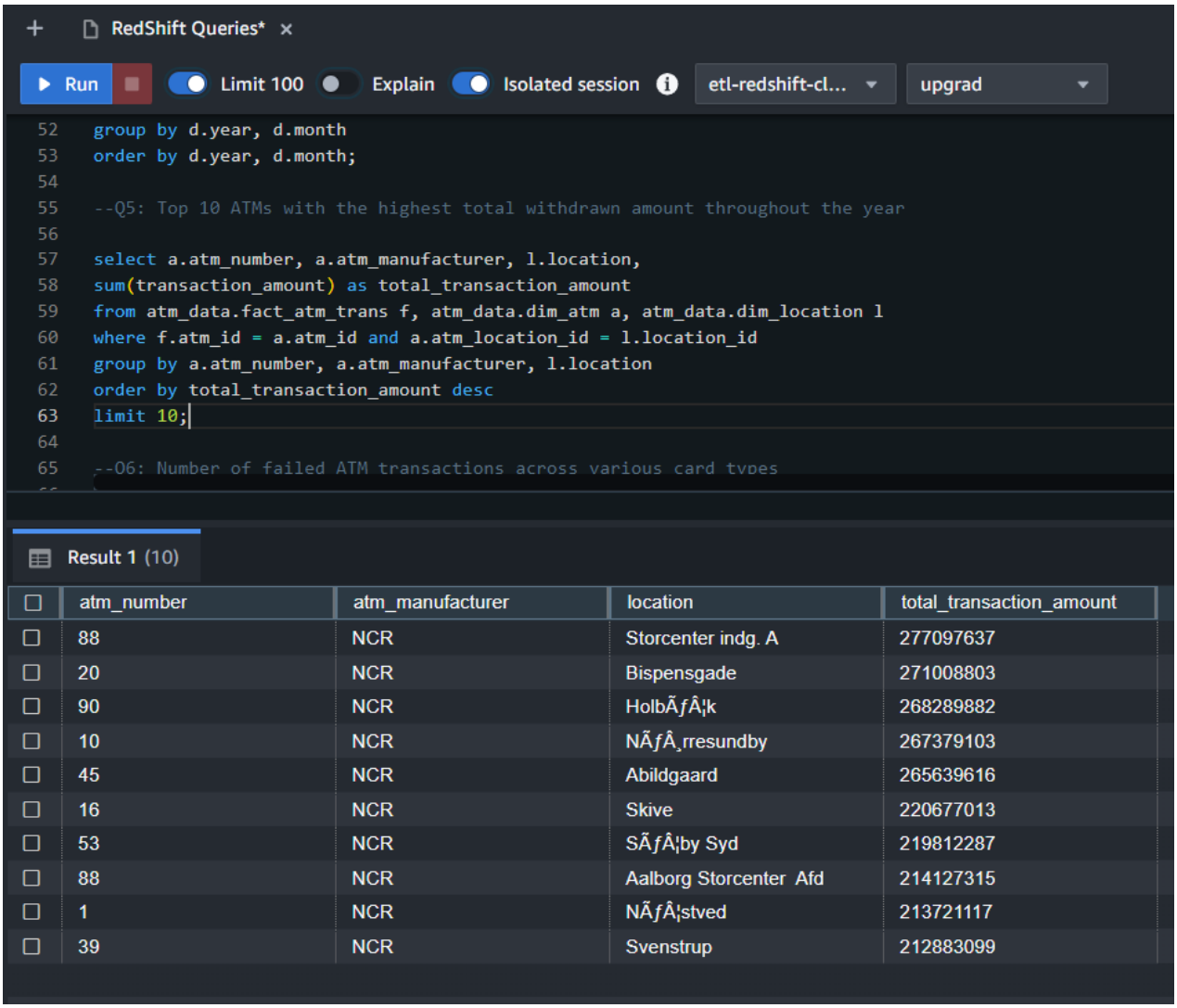
year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	203979	35102	17.2
2017	August	218096	36209	16.6
2017	December	208452	26048	12.49
2017	February	196774	36385	18.49
2017	January	201174	36281	18.03
2017	July	213462	34357	16.09
2017	June	221172	40223	18.18
2017	March	188036	36327	19.31
2017	May	222295	37075	16.67
2017	November	198226	24255	12.23
2017	October	186230	22330	11.99
2017	September	210676	33066	15.69

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

<Query>

```
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;
```

<Screenshot of the resultant table>



atm_number	atm_manufacturer	location	total_transaction_amount
88	NCR	Storcenter indg. A	277097637
20	NCR	Bispensgade	271008803
90	NCR	HolbÃƒÂ¦k	268289882
10	NCR	NÃƒÂ¸rresundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
53	NCR	SÃƒÂ¦by Syd	219812287
88	NCR	Aalborg Storcenter Afd	214127315
1	NCR	NÃƒÂ¦stved	213721117
39	NCR	Svenstrup	212883099

6. Number of failed ATM transactions across various card types

<Query>

```
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;
```

<Screenshot of the resultant table>

RedShift Queries*

Run | Limit 100 | Explain | Isolated session | etl-redshift-cl... | upgrad

```
65  --Q6: Number of failed ATM transactions across various card types
66
67  select ct.card_type,
68  count(trans_id) as total_transaction_count,
69  sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,
70  case when coalesce(inactive_count, 0) = 0 then 0.0000
71  else trunc((cast(inactive_count as
72  numeric(10,4))/total_transaction_count)*100, 2)
73  end as inactive_count_percent
74  from atm_data.fact_atm_trans f, atm_data.dim_card_type ct
75  where f.card_type_id = ct.card_type_id
76  group by ct.card_type
77  order by inactive_count_percent desc
78  limit 10;
79
80  --Q7: Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order
```

Result 1 (10)

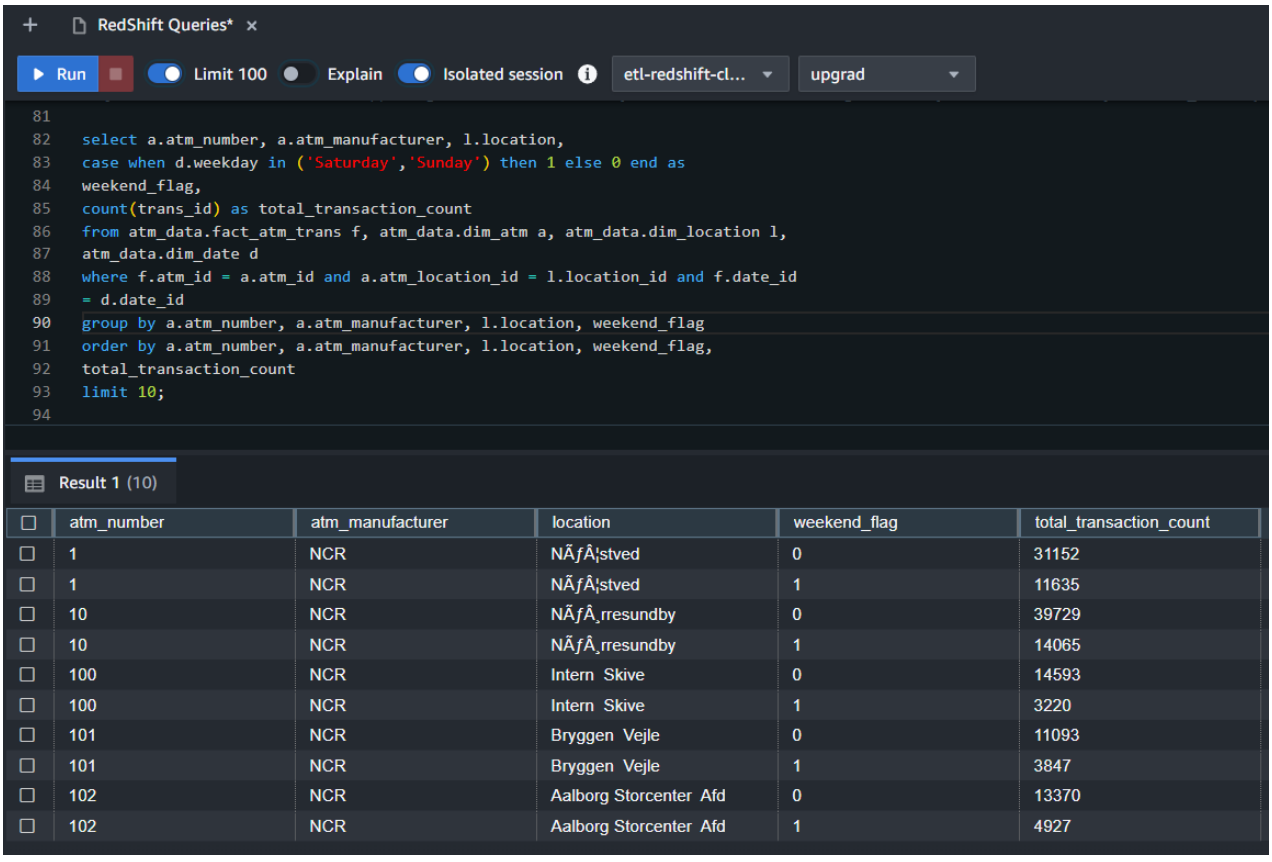
card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.76
VISA	170828	30713	17.97
Dankort - on-us	143813	24680	17.16
CIRRUS	17362	2953	17
HÃƒÂ¦vekort - on-us	62487	10331	16.53
Dankort	28581	4557	15.94
MasterCard	400507	63482	15.85
Visa Dankort - on-us	748805	112972	15.08
HÃƒÂ¦vekort	8459	1208	14.28
Visa Dankort	427840	60547	14.15

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 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;
```

<Screenshot of the resultant table>



atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃƒÂ¦stved	0	31152
1	NCR	NÃƒÂ¦stved	1	11635
10	NCR	NÃƒÂ¸rresundby	0	39729
10	NCR	NÃƒÂ¸rresundby	1	14065
100	NCR	Intern Skive	0	14593
100	NCR	Intern Skive	1	3220
101	NCR	Bryggen Vejle	0	11093
101	NCR	Bryggen Vejle	1	3847
102	NCR	Aalborg Storcenter Afd	0	13370
102	NCR	Aalborg Storcenter Afd	1	4927

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

<Query>

```
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;
```

<Screenshot of the resultant table>