Solving analytical queries on Redshift Cluster

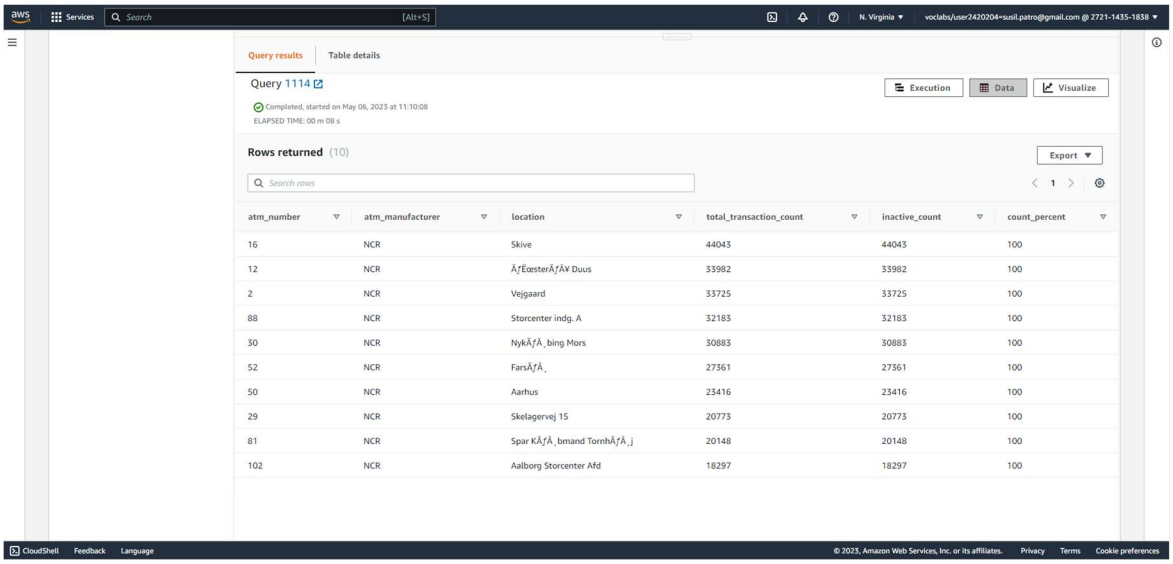
Following are the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS Redshift Query editor UI:

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_info.fact_atm_trans f, atm_info.dim_atm a, atm_info.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 inactive_count desc limit 10;
```

Screenshot of the resultant table:



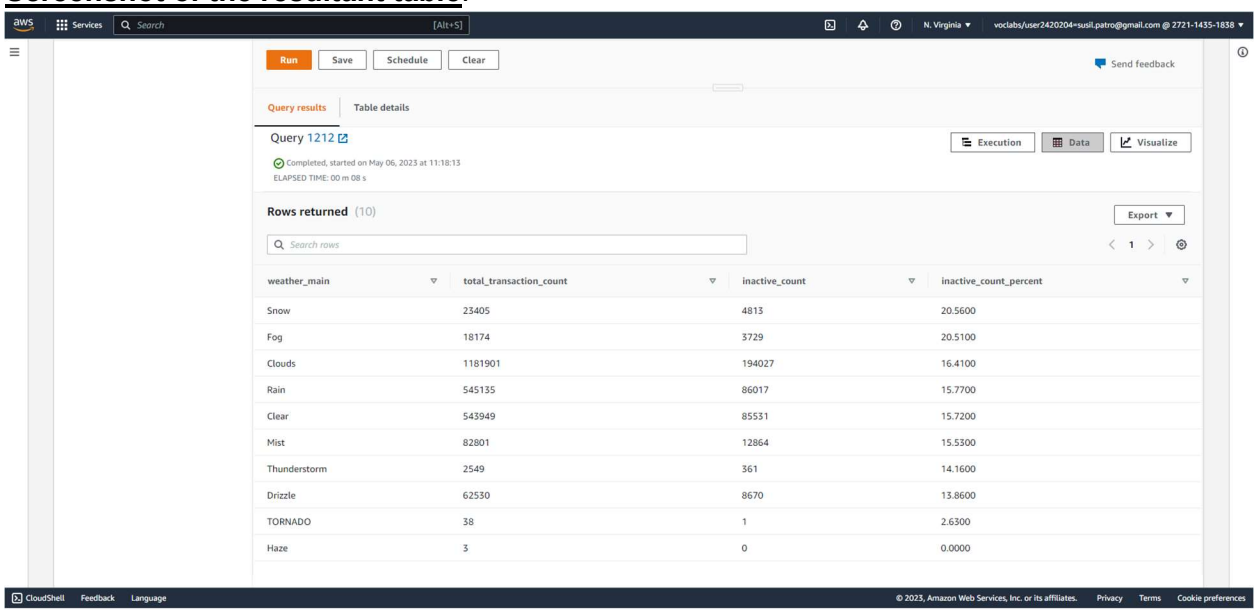
atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
16	NCR	Skive	44043	44043	100
12	NCR	ÃƒÆœsterÃƒÂ¥ Duus	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
88	NCR	Storcenter indg. A	32183	32183	100
30	NCR	NykÃƒÂ¸bing Mors	30883	30883	100
52	NCR	FarsÃƒÂ¸	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
81	NCR	Spar KÃƒÂ¸bmand TornhÃƒÂ¸j	20148	20148	100
102	NCR	Aalborg Storcenter Afd	18297	18297	100

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 cast(0 as numeric(10,4))
          else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2)
         end as inactive_count_percent
from atm_info.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:



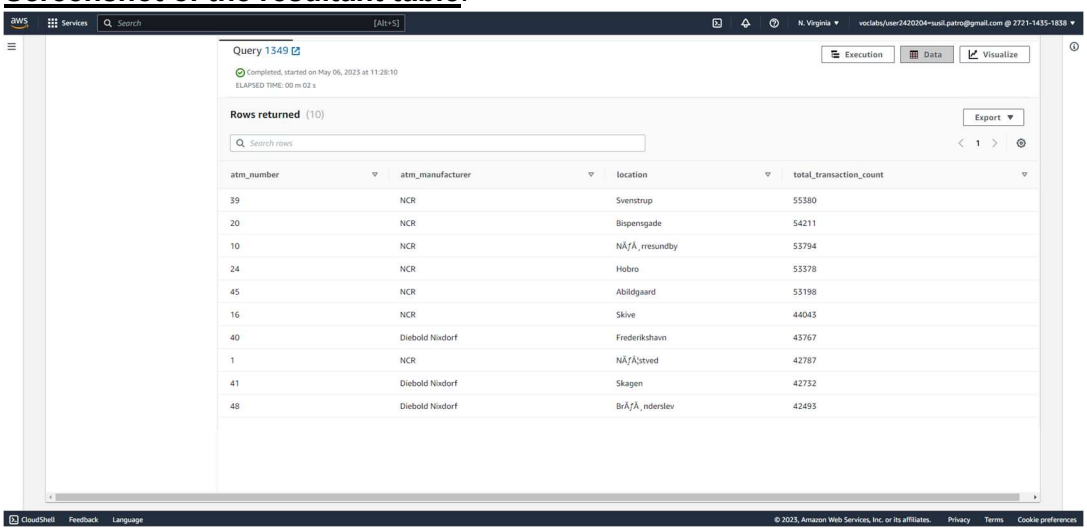
weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.5600
Fog	18174	3729	20.5100
Clouds	1181901	194027	16.4100
Rain	545135	86017	15.7700
Clear	543949	85531	15.7200
Mist	82801	12864	15.5300
Thunderstorm	2549	361	14.1600
Drizzle	62530	8670	13.8600
TORNADO	38	1	2.6300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

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

Query:

```
select  a.atm_number,
        a.atm_manufacturer,
        l.location, count(trans_id) as total_transaction_count
from atm_info.fact_atm_trans f
                INNER JOIN atm_info.dim_atm a ON f.atm_id = a.atm_id
                INNER JOIN atm_info.dim_location l on 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:



Query 1349

Completed, started on May 06, 2023 at 11:28:10
ELAPSED TIME: 00 m 02 s

Execution | Data | Visualize

Rows returned (10)

Export

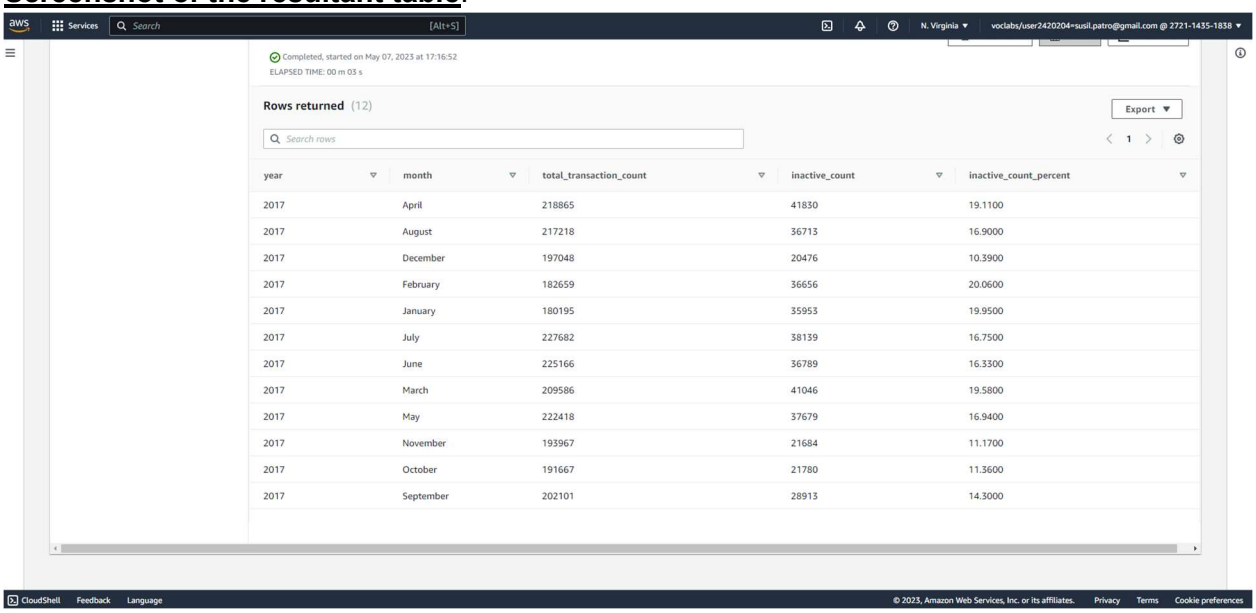
atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÃƒÂ rresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÃƒÂ stved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold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

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

Query:

```
select 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 cast(0 as numeric(10,4))
        else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2)
                        end as inactive_count_percent
from atm_info.fact_atm_trans f
inner join atm_info.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:



Completed, started on May 07, 2023 at 17:16:52
ELAPSED TIME: 00 m 03 s

Rows returned (12)

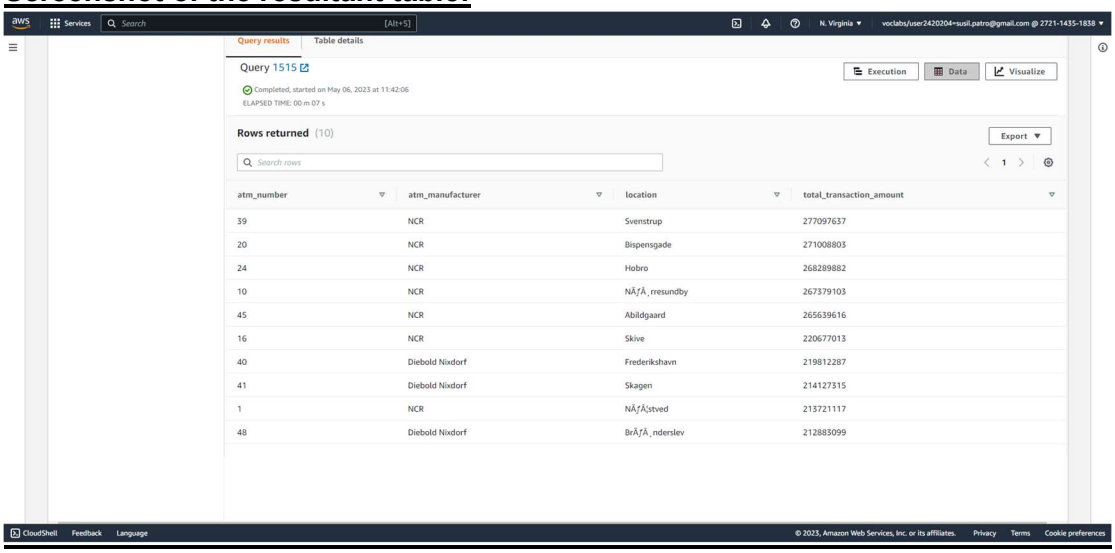
year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.1100
2017	August	217218	36713	16.9000
2017	December	197048	20476	10.3900
2017	February	182659	36656	20.0600
2017	January	180195	35953	19.9500
2017	July	227682	38139	16.7500
2017	June	225166	36789	16.3300
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  a.atm_number,
        a.atm_manufacturer,
        l.location, sum(transaction_amount) as total_transaction_amount
from atm_info.fact_atm_trans f
                INNER JOIN atm_info.dim_atm a ON f.atm_id = a.atm_id
                INNER JOIN atm_info.dim_location l ON 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:

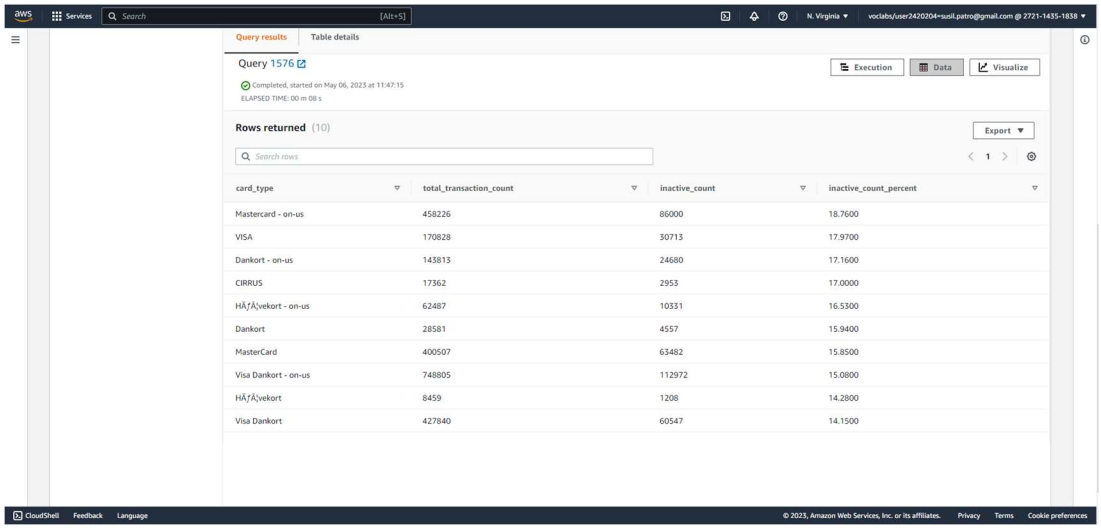


6. **Number of failed ATM transactions across various card types**

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 cast(0 as numeric(10,4))
           else trunc((cast(inactive_count as numeric(10,4))/total_transaction_count)*100, 2) end
as inactive_count_percent
from atm_info.fact_atm_trans f
         INNER JOIN atm_info.dim_card_type ct ON 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:



Query results | Table details

Query 1576

Completed, started on May 06, 2023 at 11:47:15

ELAPSED TIME: 00 m 08 s

Execution | Data | Visualize

Rows returned (10)

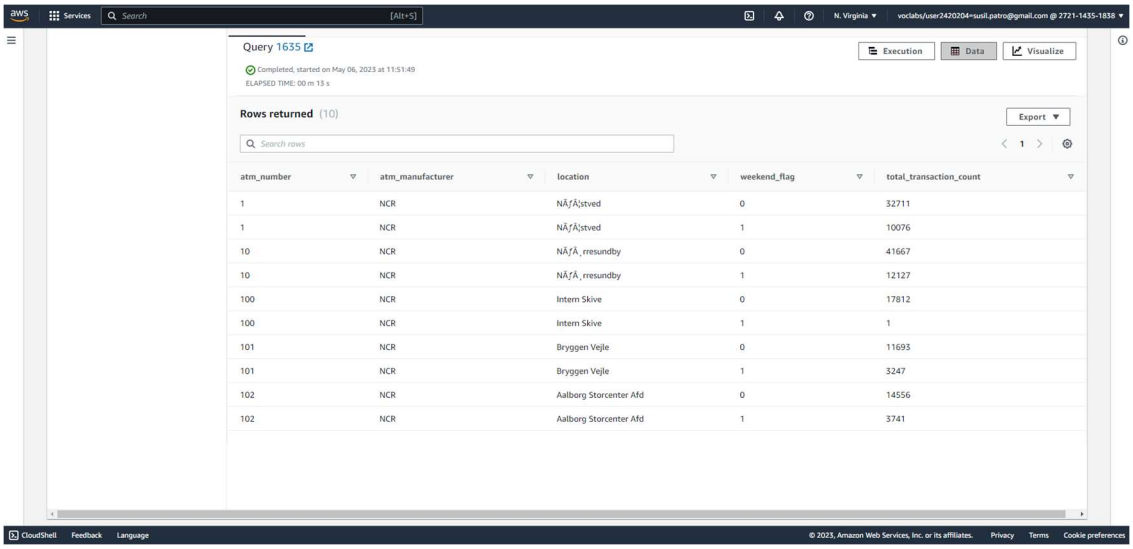
Export

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_info.fact_atm_trans f
                        INNER JOIN atm_info.dim_atm a ON f.atm_id = a.atm_id
                        INNER JOIN atm_info.dim_location l ON a.atm_location_id = l.location_id
          INNER JOIN atm_info.dim_date d ON 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:

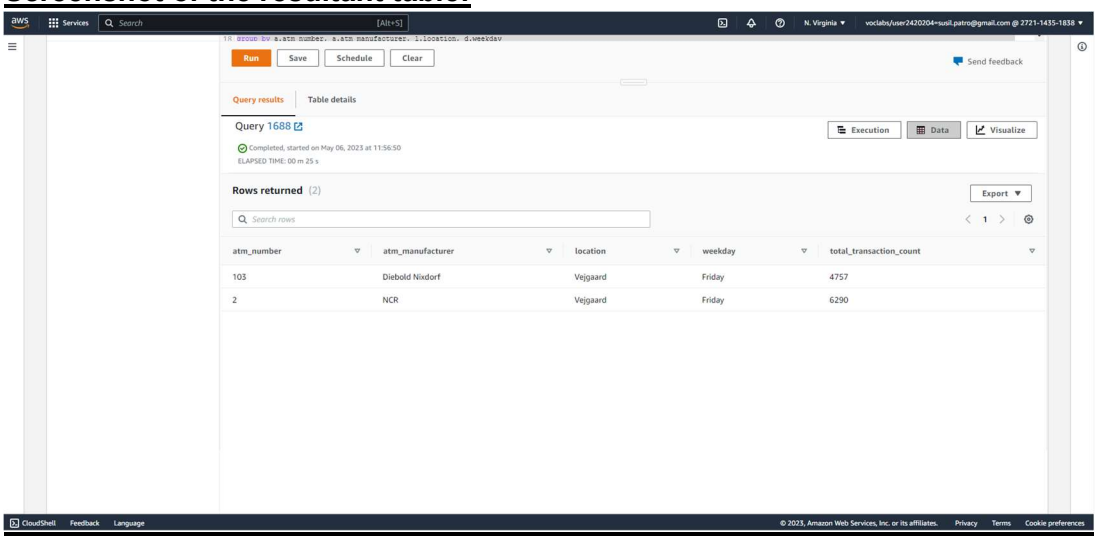


8. **Most active day in each ATMs from location "Vejgaard"**

Query:

```
select  a.atm_number,
        a.atm_manufacturer,
        l.location,
        d.weekday,
        count(trans_id) as total_transaction_count
from atm_info.fact_atm_trans f
inner join atm_info.dim_atm a on f.atm_id = a.atm_id
inner join atm_info.dim_location l on a.atm_location_id = l.location_id
inner join atm_info.dim_date d on f.date_id = d.date_id
where  l.location = 'Vejgaard'
and d.weekday in ( select d.weekday
                                        from atm_info.fact_atm_trans f
                                        inner join atm_info.dim_date d on f.date_id = d.date_id
                                        inner join atm_info.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:



Case Study submitted by:
Susil Patro, Vivek Agrawal & Harshal Sapkade