Solving analytical queries on RedShift Cluster

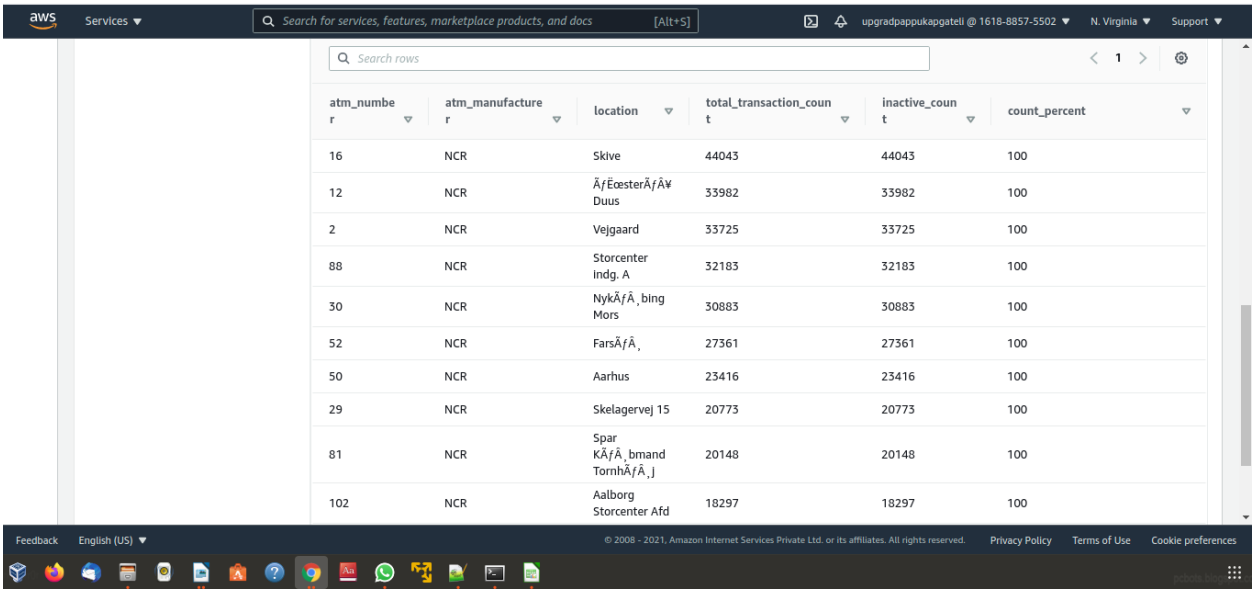
Here, you have to write the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS RedShift Query editor UI.

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

<u>Query</u>

select da.atm_number, da.atm_manufacturer, dl.location, count(fat.trans_id) as total_transaction_count, count(fat.atm_status) as inactive_count, ((total_transaction_count*100) /inactive_count) as count_percent from atm_schema.dim_location as dl, atm_schema.dim_atm as da, atm_schema.fact_atm_trans as fat where fat.atm_status = 'Inactive' and fat.atm_id = da.atm_id and fat.weather_loc_id = dl.location_id group by da.atm_number, da.atm_manufacturer,dl.location order by total_transaction_count desc limit 10;

<u>Screenshot of the resultant table</u>



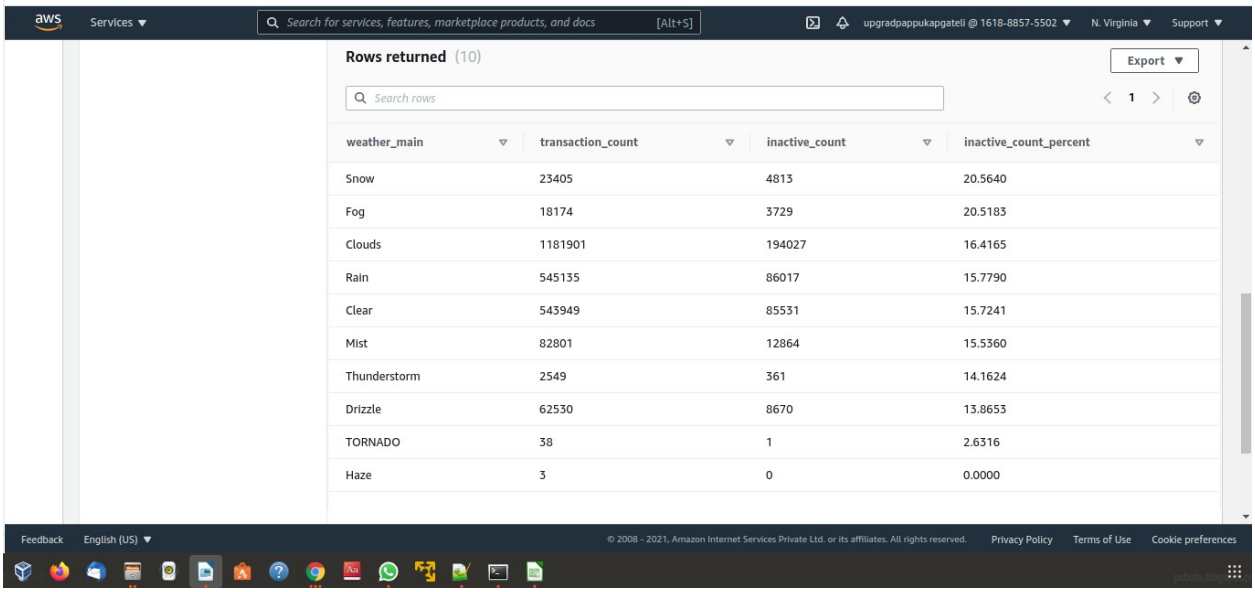
atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
16	NCR	Skive	44043	44043	100
12	NCR	ÃƒÈœsterÃƒÂ¥ Duus	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
88	NCR	Storcenter indg. A	32183	32183	100
30	NCR	NykÃƒÂ¸bing Mors	30883	30883	100
52	NCR	FarsÃƒÂ¸	27361	27361	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 fat.weather_main as weather_main, count(fat.trans_id) as transaction_count,
cast(sum(case when fat.atm_status !='Active'
then 1 else 0 end) as decimal(10,0)) as inactive_count,
round(inactive_count*100.0/transaction_count,4) as inactive_count_percent
from atm_schema.fact_atm_trans as fat
where fat.weather_main !=''
group by fat.weather_main order by inactive_count_percent desc;
```

Screenshot of the resultant table



Rows returned (10)

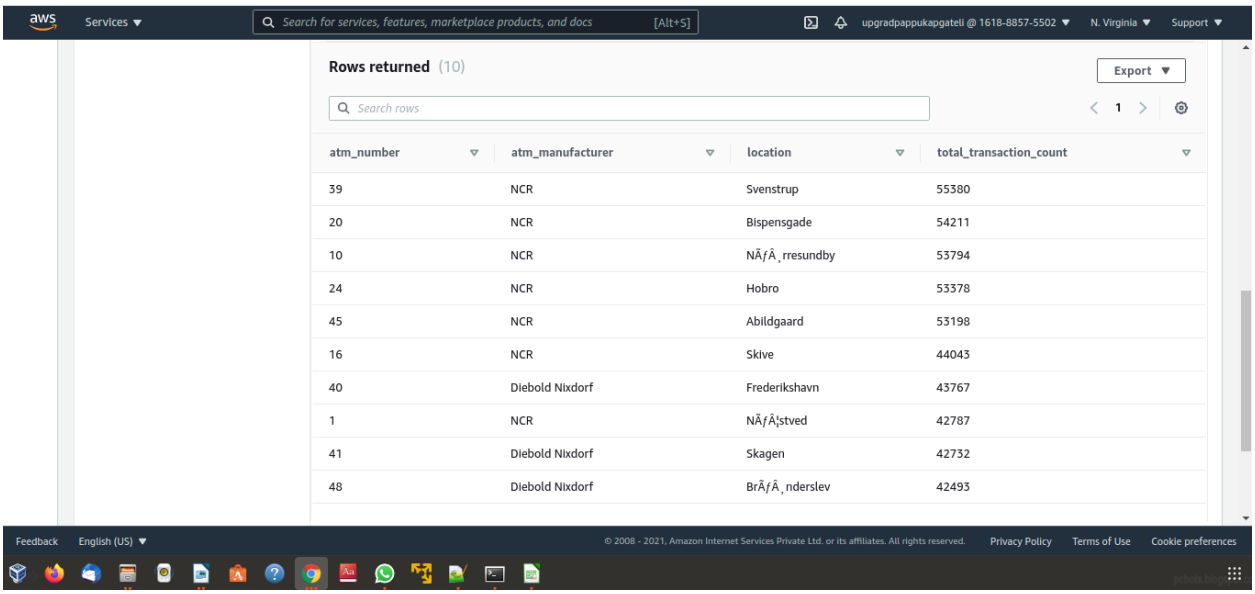
weather_main	transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.5640
Fog	18174	3729	20.5183
Clouds	1181901	194027	16.4165
Rain	545135	86017	15.7790
Clear	543949	85531	15.7241
Mist	82801	12864	15.5360
Thunderstorm	2549	361	14.1624
Drizzle	62530	8670	13.8653
TORNADO	38	1	2.6316
Haze	3	0	0.0000

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

Query

select da.atm_number, da.atm_manufacturer, dl.location, count(fat.trans_id) as total_transaction_count from atm_schema.dim_location as dl, atm_schema.dim_atm as da, atm_schema.fact_atm_trans as fat where fat.atm_id = da.atm_id and fat.weather_loc_id = dl.location_id group by da.atm_number, da.atm_manufacturer,dl.location order by total_transaction_count desc limit 10;

Screenshot of the resultant table



Rows returned (10)

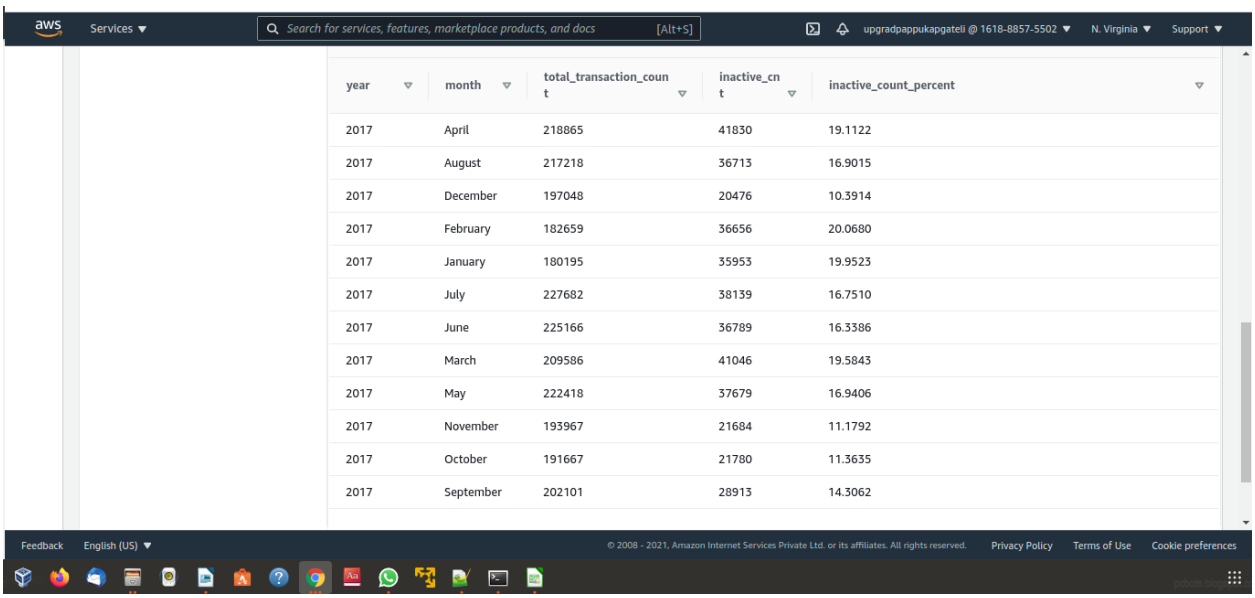
atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÃƒÂ¸rresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÃƒÂ¦stved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold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

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

Query

```
select dd.year as year, dd.month as month, count(fat.trans_id) as total_transaction_count,
cast(sum(case when fat.atm_status != 'Active' then 1 else 0 end) as decimal(10,0)) as
inactive_cnt,
round((inactive_cnt * 100.0)/total_transaction_count, 4) as inactive_count_percent
from atm_schema.fact_atm_trans as fat
left join
atm_schema.dim_date as dd on fat.date_id = dd.date_id
group by year, month
order by month;
```

Screenshot of the resultant table



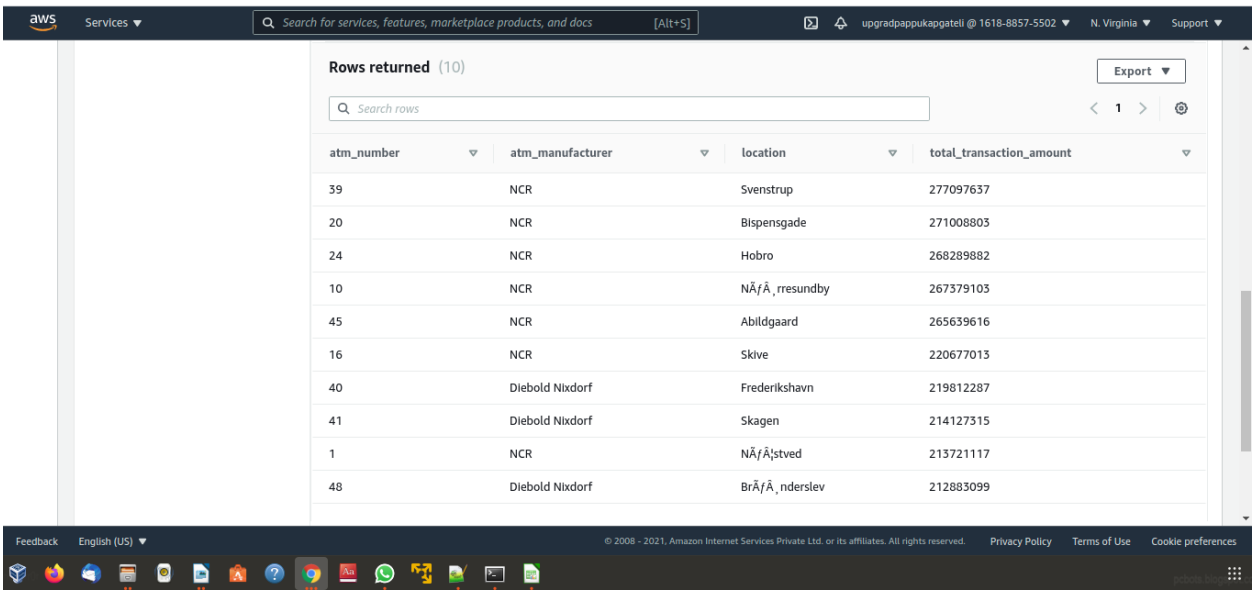
year	month	total_transaction_count	inactive_cnt	inactive_count_percent
2017	April	218865	41830	19.1122
2017	August	217218	36713	16.9015
2017	December	197048	20476	10.3914
2017	February	182659	36656	20.0680
2017	January	180195	35953	19.9523
2017	July	227682	38139	16.7510
2017	June	225166	36789	16.3386
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**

Query

select da.atm_number, da.atm_manufacturer, dl.location, sum(fat.transaction_amount) as total_transaction_amount from atm_schema.dim_location as dl, atm_schema.dim_atm as da, atm_schema.fact_atm_trans as fat where fat.atm_id = da.atm_id and fat.weather_loc_id = dl.location_id group by da.atm_number, da.atm_manufacturer,dl.location order by total_transaction_amount desc limit 10;

Screenshot of the resultant table



Rows returned (10)

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÃƒÂ¸rresundby	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

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

Query

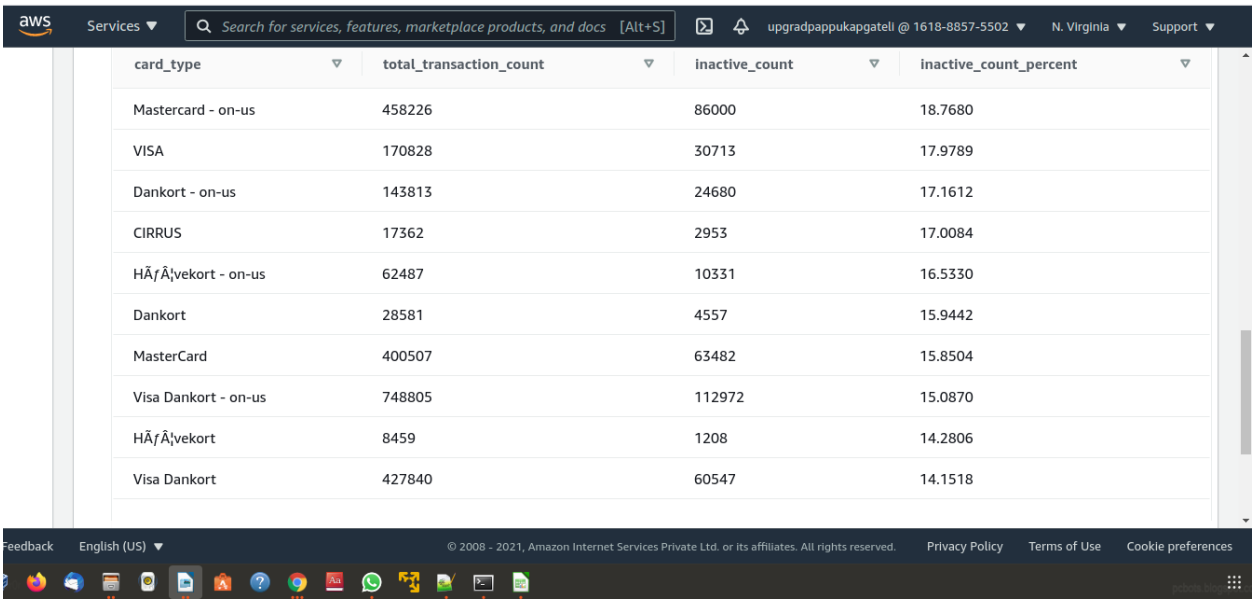
```
select total_card_type.card_type, total_active_inactive_trans_card.total_transaction_count,
atm_status_inactive.inactive_count,round(atm_status_inactive.inactive_count*100.0/
total_active_inactive_trans_card.total_transaction_count,4) as inactive_count_percent
from (

(select fat.card_type_id, count(*) as inactive_count from atm_schema.fact_atm_trans as fat
where fat.card_type_id is not null and fat.atm_status != 'Active' group by fat.card_type_id) as
atm_status_inactive

join
(select fat.card_type_id, count(*) as total_transaction_count from atm_schema.fact_atm_trans
as fat where fat.card_type_id is not null and fat.atm_status != '' group by fat.card_type_id) as
total_active_inactive_trans_card
on atm_status_inactive.card_type_id = total_active_inactive_trans_card.card_type_id

join
(select dct.card_type, dct.card_type_id from atm_schema.dim_card_type as dct group by
dct.card_type, dct.card_type_id) as total_card_type
on
atm_status_inactive.card_type_id = total_card_type.card_type_id ) order by
inactive_count_percent desc;
```

Screenshot of the resultant table



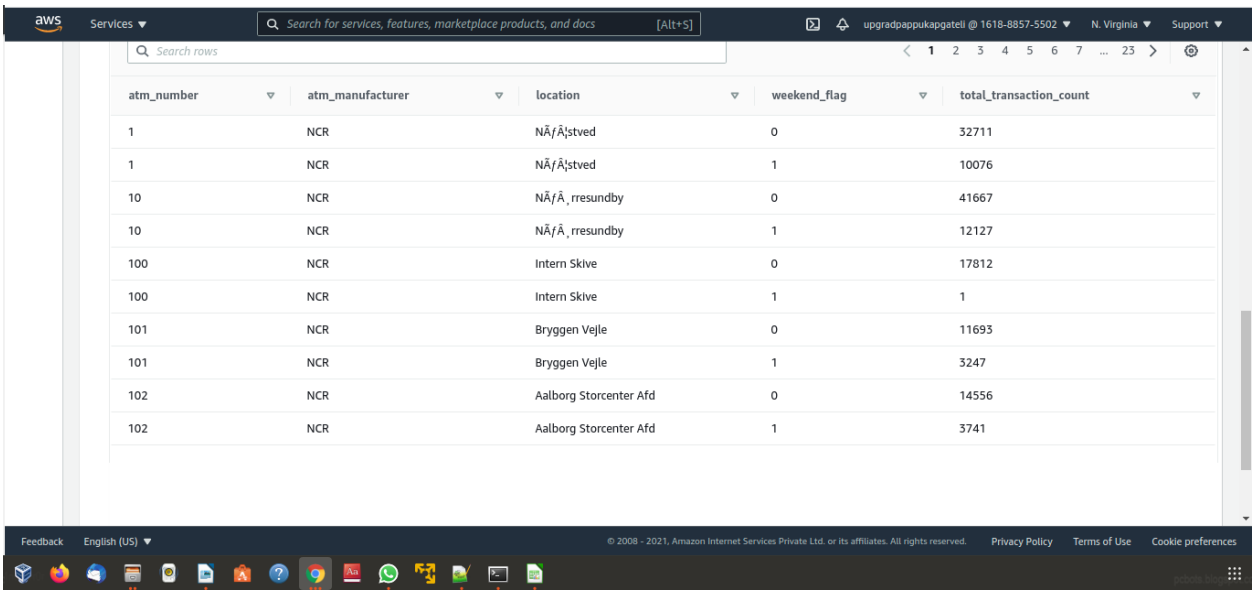
card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.7680
VISA	170828	30713	17.9789
Dankort - on-us	143813	24680	17.1612
CIRRUS	17362	2953	17.0084
HÃƒÂ¦vekort - on-us	62487	10331	16.5330
Dankort	28581	4557	15.9442
MasterCard	400507	63482	15.8504
Visa Dankort - on-us	748805	112972	15.0870
HÃƒÂ¦vekort	8459	1208	14.2806
Visa Dankort	427840	60547	14.1518

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 da.atm_number as atm_number, da.atm_manufacturer as atm_manufacturer, dl.location
as location, cast((case when dd.weekday in ('Saturday','Sunday') then 1 else 0 end) as
decimal(10,0)) as weekend_flag,
count(fat.trans_id) as total_transaction_count from atm_schema.fact_atm_trans as fat
left join
atm_schema.dim_date as dd on fat.date_id =  dd.date_id
left join
atm_schema.dim_atm as da on fat.atm_id = da.atm_id
left join
atm_schema.dim_location as dl on fat.weather_loc_id = dl.location_id
group by atm_number, atm_manufacturer, location, weekend_flag
order by atm_number, atm_manufacturer, location, weekend_flag;
```

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_rank_trans_data.atm_number,atm_rank_trans_data.atm_manufacturer,atm_rank_trans_da
ta.location,atm_rank_trans_data.transaction_count
from
(
select da.atm_number,da.atm_manufacturer,dl.location,dd.weekday,count(fat.trans_id) as
transaction_count,
rank() over(partition by da.atm_number,da.atm_manufacturer, dl.location order by
transaction_count desc) as atm_trans_rank
from atm_schema.fact_atm_trans as fat
inner join
atm_schema.dim_atm as da
on fat.atm_id = da.atm_id

inner join
atm_schema.dim_location as dl
on dl.location_id = da.atm_location_id

inner join
atm_schema.dim_date as dd
on dd.date_id = fat.date_id

where location = 'Vejgaard'

group by da.atm_number, da.atm_manufacturer, dl.location,dd.weekday) as
atm_rank_trans_data
where atm_trans_rank = 1 ;
```

Screenshot of the resultant table