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 atm.atm_number,count(1) as Inactive_Trans from etl_project.fact_atm_trans F
Inner join etl_project.atm on atm.atm_dim_id=f.atm_dim_id
where atm_status='Inactive'
group by atm.atm_number
Order by 2 desc limit 10;
```

<Screenshot of the resultant table>

Rows returned (10)

Search rows

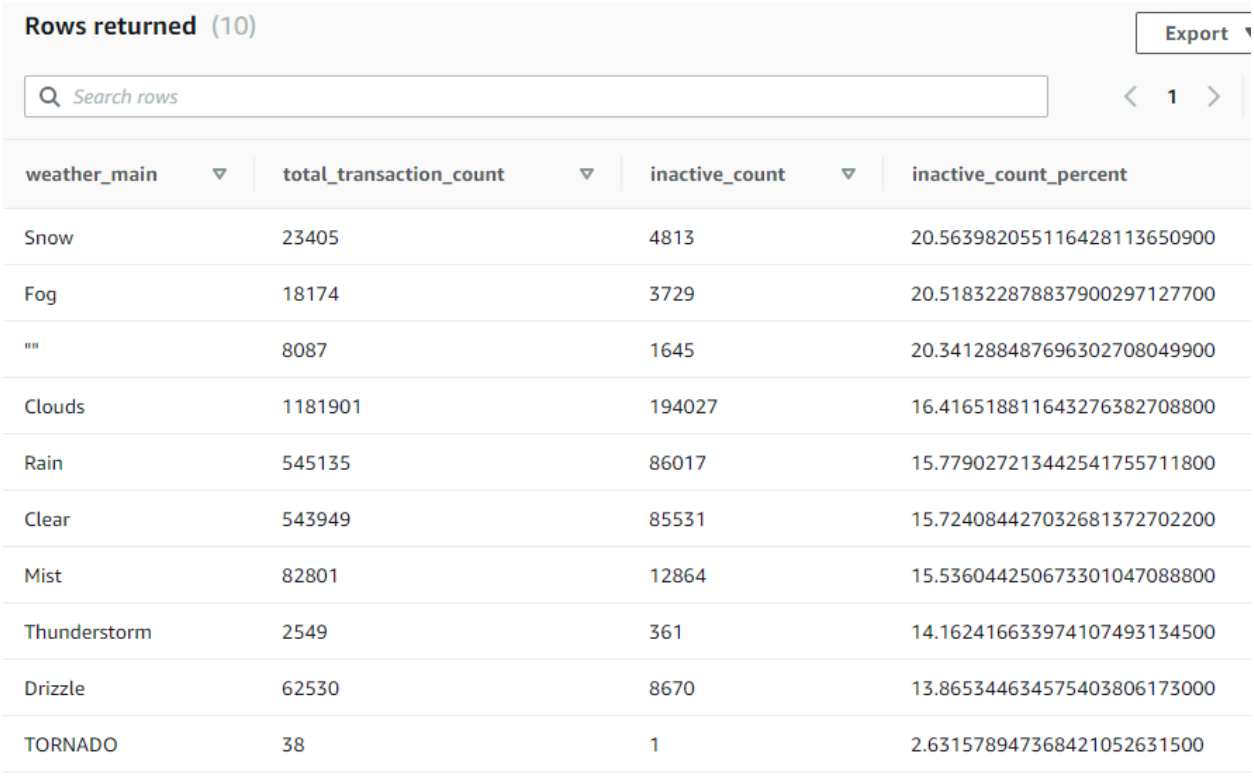
atm_number	inactive_trans
16	44043
12	33982
2	33725
88	32183
30	30883
52	27361
50	23416
29	20773
81	20148
102	18297

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

<Query>

```
select f.weather_main , count(f.trans_id) as total_transection_count ,
sum(case when f.atm_status = 'Inactive' then 1 else 0 end) as inactive_count ,
(cast(inactive_count as numeric(10,4))/total_transection_count)*100 as inactive_count_percent
from etl_project.fact_atm_trans as f
where LEN(weather_main) != 0
group by f.weather_main
order by inactive_count_percent desc
limit 10;
```

<Screenshot of the resultant table>



Rows returned (10)

Export

Search rows

1

weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.563982055116428113650900
Fog	18174	3729	20.518322878837900297127700
""	8087	1645	20.341288487696302708049900
Clouds	1181901	194027	16.416518811643276382708800
Rain	545135	86017	15.779027213442541755711800
Clear	543949	85531	15.724084427032681372702200
Mist	82801	12864	15.536044250673301047088800
Thunderstorm	2549	361	14.162416633974107493134500
Drizzle	62530	8670	13.865344634575403806173000
TORNADO	38	1	2.631578947368421052631500

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

<Query>

```
select d.year,atm.atm_number,count(1) as Number_Of_Trans from etl_project.fact_atm_trans F
Inner join etl_project.atm on atm.atm_dim_id=f.atm_dim_id
inner join etl_project.date d on d.date_id=f.date_id
group by atm.atm_number,d.year
Order by 3 desc limit 10;
```

<Screenshot of the resultant table>

Rows returned (10)

Search rows

year	atm_number	number_of_trans
2017	39	55380
2017	20	54211
2017	10	53794
2017	24	53378
2017	45	53198
2017	16	44043
2017	40	43767
2017	1	42787
2017	41	42732
2017	48	42493

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

<Query>

```
select d.year,d.Month,count(1) as ATM_Tran_Inactive from etl_project.fact_atm_trans F
Inner join etl_project.atm on atm.atm_dim_id=f.atm_dim_id
inner join etl_project.date d on d.date_id=f.date_id
where f.atm_status='Inactive'
group by d.year,d.Month
Order by 3 desc;
```

<Screenshot of the resultant table>

Rows returned (12)

Search rows

year	month	atm_tran_inactive
2017	April	41830
2017	March	41046
2017	July	38139
2017	May	37679
2017	June	36789
2017	August	36713
2017	February	36656
2017	January	35953
2017	September	28913
2017	October	21780

Rows returned (12)

Search rows

year	month	atm_tran_inactive
2017	November	21684
2017	December	20476

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

<Query>

```
select d.year,atm.atm_number,sum(transaction_amount) as withdrawn_amount  from
etl_project.fact_atm_trans F
Inner join etl_project.atm on atm.atm_dim_id=f.atm_dim_id
inner join etl_project.date d on d.date_id=f.date_id
group by atm.atm_number,d.year
Order by 3 desc limit 10;
```

<Screenshot of the resultant table>

Rows returned (10)

Search rows

year	atm_number	withdrawn_amount
2017	39	277097637
2017	20	271008803
2017	24	268289882
2017	10	267379103
2017	45	265639616
2017	16	220677013
2017	40	219812287
2017	41	214127315
2017	1	213721117
2017	48	212883099

6. Number of failed ATM transactions across various card types

<Query>

```
select c.card_type , count(f.trans_id) as total_transaction_count ,
sum(case when f.atm_status = 'Inactive' then 1 else 0 end) as inactive_count ,
(cast(inactive_count as numeric(10,4))/total_transaction_count)*100 as inactive_count_percent
from etl_project.fact_atm_trans as f
join etl_project.cardtype as c on c.cardtype_id = f.cardtype_id
group by c.card_type
order by inactive_count_percent DESC
limit 10;
```

<Screenshot of the resultant table>

Rows returned (10) Export

Search rows 1

card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.768031495375644332709100
VISA	170828	30713	17.978902755988479640339900
Dankort - on-us	143813	24680	17.161174580879336360412400
CIRRUS	17362	2953	17.008409169450524133164300
HÃƒÂ¦vekort - on-us	62487	10331	16.533038872085393761902400
Dankort	28581	4557	15.944158706833210874357000
MasterCard	400507	63482	15.850409605824617297575300
Visa Dankort - on-us	748805	112972	15.086971908574328429965000
HÃƒÂ¦vekort	8459	1208	14.280647830712850218701900
Visa Dankort	427840	60547	14.151785714285714285714200

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.atm_location ,
case WHEN d.weekday in ('Saturday' , 'Sunday') then 1 else 0 end AS weekend_flag ,
count(f.trans_id) as total_transaction_count
from etl_project.atm as a join etl_project.location as l
on a.atm_location_id = l.location_id
join etl_project.fact_atm_trans as f
on a.atm_dim_id = f.atm_dim_id
join etl_project.date as d on d.date_id = f.date_id
group by a.atm_number , a.atm_manufacturer , l.atm_location , weekend_flag
order by a.atm_number , a.atm_manufacturer , l.atm_location , weekend_flag ,
total_transaction_count
limit 10;
```

<Screenshot of the resultant table>

Rows returned (10) Export

Search rows 1

atm_number	atm_manufacturer	atm_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_number,atm_location,weekday,total_transaction_count
from ( select
atm_number,atm_location,weekday,total_transaction_count,max(total_transaction_count) over
(partition by atm_number) as max_count
from (
select a.atm_number,l.atm_location ,d.weekday,count(f.trans_id) as total_transaction_count
 from etl_project.Fact_Atm_Trans as f inner join etl_project.location as l
  on f.location_id=l.location_id
  inner join etl_project.atm as a on a.atm_dim_id=f.atm_dim_id
  inner join etl_project.date as d on f.date_id=d.date_id
  where l.atm_location='Vejgaard'
  group by a.atm_number,l.atm_location,d.weekday
)
          )
where total_transaction_count=max_count;
```

<Screenshot of the resultant table>

atm_number	atm_location	weekday	total_transaction_count
103	Vejgaard	Friday	4757
2	Vejgaard	Friday	6290