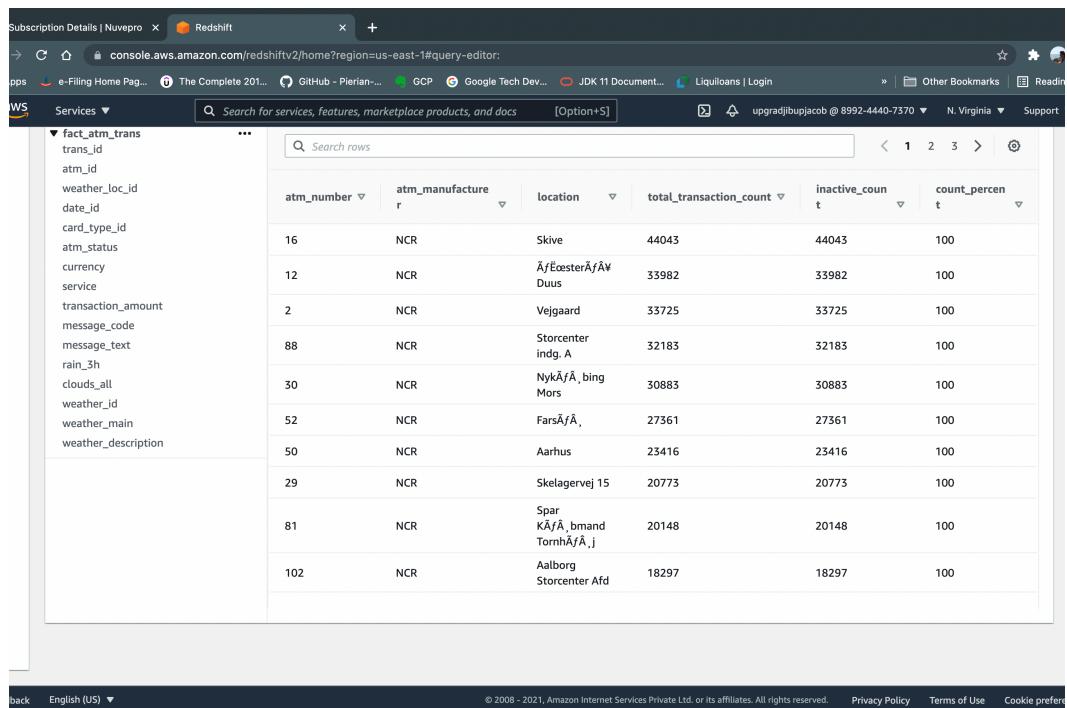


# Solving analytical queries on RedShift Cluster

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

```
select atm.atm_number,atm.atm_manufacturer,loc.location
    ,count(trans.trans_id) as total_transaction_count,
    count(trans.atm_status) as inactive_count,
    inactive_count/total_transaction_count * 100 as count_percent
from atm_etl.dim_atm atm , atm_etl.dim_location loc ,atm_etl.fact_atm_trans trans
where atm.atm_location_id = loc.location_id
and atm.atm_id = trans.atm_id
and loc.location_id = trans.weather_loc_id
and trans.atm_status = 'Inactive'
group by atm.atm_number,atm.atm_manufacturer,loc.location
order by total_transaction_count desc;
```



The screenshot shows the AWS Redshift Query Editor interface. The query has been executed, and the results are displayed in a table. The table has columns: atm\_number, atm\_manufacturer, location, total\_transaction\_count, inactive\_count, and count\_percent. The data shows the top 10 ATMs with the highest percentage of inactive transactions. The results are as follows:

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
16	NCR	Skive	44043	44043	100
12	NCR	Århus	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
88	NCR	Storcenter Indg. A	32183	32183	100
30	NCR	Nykøbing Mors	30883	30883	100
52	NCR	Farsø,	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
81	NCR	Spar Købmand Tornhøj	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

```

with total_trans as (select trans.weather_main,count(trans.trans_id) as total_trans_count
                     from atm_etl.fact_atm_trans trans
                     where trans.weather_main !="
group by trans.weather_main),
inactive_trans as (select trans.weather_main,count(trans.atm_status) as inactive_count
                     from atm_etl.fact_atm_trans trans
                     where trans.atm_status='Inactive'
                     and trans.weather_main != "
                     group by trans.weather_main)
select total_trans.weather_main,total_trans_count
      ,nvl(inactive_count::int,0) as total_inactive_count ,
      round(100.0000 * total_inactive_count / total_trans_count,4) inactive_count_percent
from total_trans left join inactive_trans
on total_trans.weather_main = inactive_trans.weather_main
group by total_trans.weather_main,total_trans_count,inactive_count
order by inactive_count_percent desc;

```

The screenshot shows the AWS Redshift Query Editor interface. The table has the following columns:

weather_main	total_trans_count	total_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

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

The screenshot shows the AWS Redshift query results page. The results are displayed in a table titled 'Rows returned (10)'. The table has columns: atm\_number, atm\_manufacturer, location, and total\_transaction\_count. The data is as follows:

atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÃrresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÃrsted	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÃ¥fjÃ¶nderslev	42493

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

```
with total_trans as ( select date.year,date.month,count(trans.trans_id) as total_transaction_count
                      from atm_etl.dim_date as date , atm_etl.fact_atm_trans trans
                     where trans.date_id = date.date_id
                     group by date.year,date.month),
inactive_trans as ( select date.year,date.month,count(trans.trans_id) as inactive_count
                      from atm_etl.dim_date as date , atm_etl.fact_atm_trans trans
                     where trans.date_id = date.date_id
                     and trans.atm_status = 'Inactive'
                     group by date.year,date.month)
select
total_trans.year,total_trans.month,total_trans.total_transaction_count ,inactive_trans.inactive_count
,cast (trunc(100.0000 * inactive_trans.inactive_count /total_trans.total_transaction_count,2)as
numeric(10,4)) inactive_count_percent
from total_trans left outer join inactive_trans
on total_trans.year = inactive_trans.year
and total_trans.month = inactive_trans.month
order by total_trans.month;
```

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

```
select atm.atm_number,atm.atm_manufacturer,loc.location,sum(trans.transaction_amount) as
total_transaction_amount
from atm_etl.dim_atm atm, atm_etl.dim_location loc , atm_etl.fact_atm_trans trans
where atm.atm_id          =      trans.atm_id
and   atm.atm_location_id =      loc.location_id
and   loc.location_id     =      trans.weather_loc_id
```

```
group by atm.atm_number,atm.atm_manufacturer,loc.location  
order by total_transaction_amount desc  
limit 10
```

The screenshot shows the AWS Redshift Query Editor interface. On the left, there is a sidebar with a tree view of database schema, including 'dim\_location', 'fact\_atm\_trans', and various fact tables like 'weather\_loc\_id', 'date\_id', etc. The main area displays a table titled 'Rows returned (10)' with the following columns: atm\_number, atm\_manufacturer, location, and total\_transaction\_amount. The data is as follows:

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispengsgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÃfÃ¸resundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	NÃfÃ'stved	213721117
48	Diebold Nixdorf	BrÃfÃ¸nderslev	212883099

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

```
with total_trans as ( select card.card_type,count(trans.trans_id) total_transaction_count
                      from atm_etl.dim_card_type card , atm_etl.fact_atm_trans trans
                     where card.card_type_id = trans.card_type_id
                       group by card.card_type),
inactive_trans as ( select card.card_type,count(trans.trans_id) inactive_count
                      from atm_etl.dim_card_type card , atm_etl.fact_atm_trans trans
                     where card.card_type_id = trans.card_type_id
                       and trans.atm_status = 'Inactive'
                       group by card.card_type)
select total_trans.card_type,total_trans.total_transaction_count,inactive_trans.inactive_count ,
cast(trunc(100.0000 * inactive_count/total_transaction_count,2) as numeric(10,4))
inactive_count_percent
from total_trans left outer join inactive_trans
on total_trans.card_type = inactive_trans.card_type
order by inactive_count_percent desc
```

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
H&A 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
H&A 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 atm.atm_number , atm.atm_manufacturer , loc.location ,
case date.weekday
    when 'Sunday' then '1'
    when 'Saturday' then '1'
    else '0' end as weekend_flag,
count(trans.trans_id) total_transaction_count
from atm_etl.dim_atm atm , atm_etl.dim_location loc, atm_etl.dim_date date,
atm_etl.fact_atm_trans trans
where atm.atm_id = trans.atm_id
and atm.atm_location_id = loc.location_id
and loc.location_id = trans.weather_loc_id
and trans.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

```

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	Næstved	0	32711
1	NCR	Næstved	1	10076
10	NCR	Næsby	0	41667
10	NCR	Næsby	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"**

```

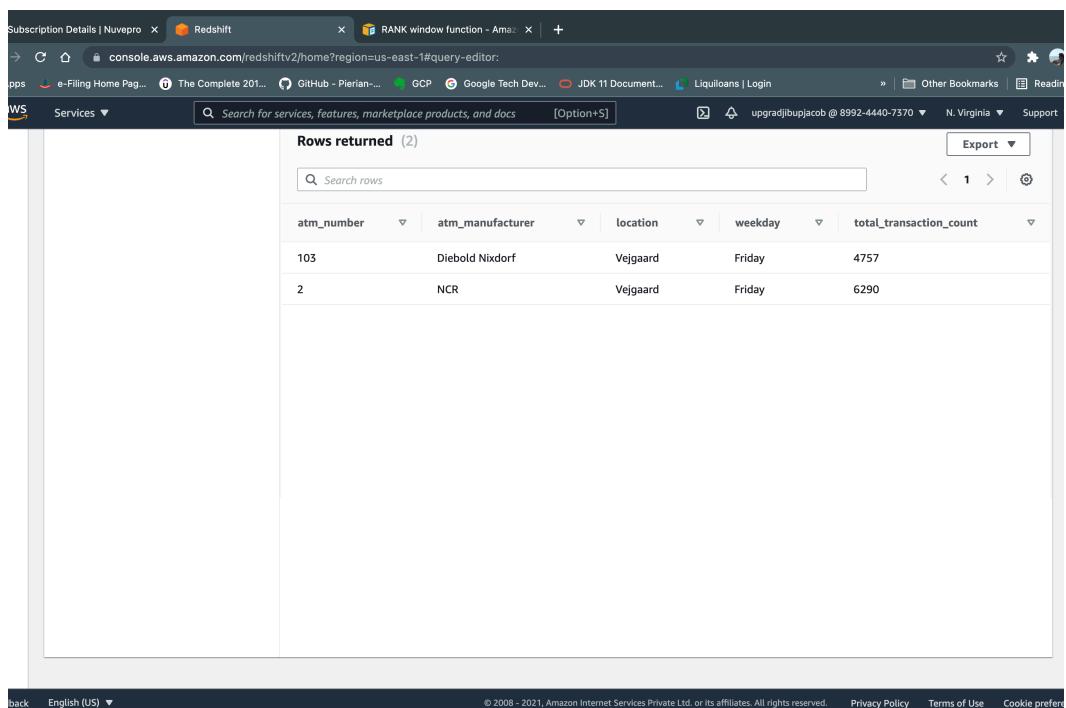
select atm_number,atm_manufacturer,location,weekday,total_transaction_count
from
(select atm.atm_number , atm.atm_manufacturer , loc.location ,
date.weekday,

```

```

count(trans.trans_id) total_transaction_count,
rank() over (partition by atm_number order by total_transaction_count desc) as rank
from atm_etl.dim_atm atm , atm_etl.dim_location loc, atm_etl.dim_date date,
atm_etl.fact_atm_trans trans
where atm.atm_id = trans.atm_id
and atm.atm_location_id = loc.location_id
and loc.location_id = trans.weather_loc_id
and trans.date_id = date.date_id
and loc.location = 'Vejgaard'
group by atm.atm_number , atm.atm_manufacturer , loc.location,date.weekday
order by atm.atm_number , atm.atm_manufacturer ,
loc.location,date.weekday,total_transaction_count)
where rank =1;

```



The screenshot shows a browser window with the AWS Lambda console URL: <https://console.aws.amazon.com/redshiftv2/home?region=us-east-1#query-editor>. The query results are displayed in a table format.

**Rows returned (2)**

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejgaard	Friday	4757
2	NCR	Vejgaard	Friday	6290