



# 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 C.atm_number, C.atm_manufacturer, D.location, total_transaction_count, inactive_count,
inactive_count/total_transaction_count*100.0 as count_percent
from (
       select atm_id, count(atm_id) as total_transaction_count
       from etl_bank_schema.fact_atm_trans
       group by atm_id
       order by total_transaction_count desc
) A, (
       select atm_id, count(atm_id) as inactive_count
       from etl_bank_schema.fact_atm_trans
       where atm_status = 'Inactive'
       group by atm_id
       order by inactive_count desc limit 10
) B,
etl_bank_schema.dim_atm C,
etl_bank_schema.dim_location D
where A.atm_id = B.atm_id
AND B.atm_id = C.atm_id
AND C.atm_location_id = D.location_id
ORDER BY inactive_count desc;
```





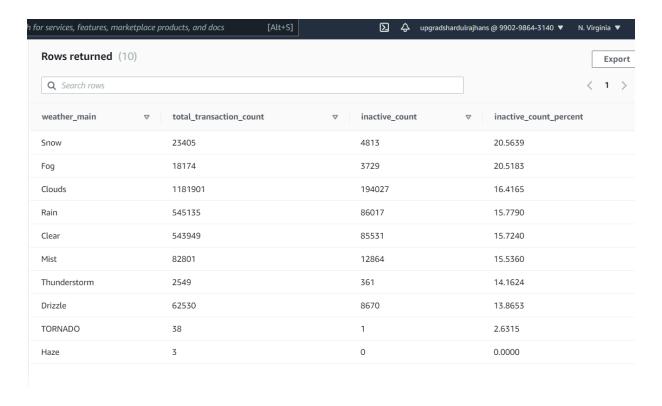
n for services, features, r	marketplace products, and doc	s [Alt+S]	🔈 🗘 upgradshardulrajhan	s @ 9902-9864-3140 ▼	N. Virginia ▼ Support ▼
Rows returned (10)					Export ▼
Q Search rows					⟨ 1 ⟩ ⊚
atm_number ▽	atm_manufacturer ▽	location $\nabla$	total_transaction_count ▽	inactive_count ▽	count_percent ▽
16	NCR	Skive	44043	44043	100.0
12	NCR	$ ilde{A} f  ilde{E}  ilde{cester}  ilde{A} f  ilde{A}  ilde{Y}  ext{ Duus}$	33982	33982	100.0
2	NCR	Vejgaard	33725	33725	100.0
88	NCR	Storcenter indg. A	32183	32183	100.0
30	NCR	Nyk $ ilde{A} f \hat{A}$ , bing Mors	30883	30883	100.0
52	NCR	Fars $ ilde{A}f\hat{A}$ ,	27361	27361	100.0
50	NCR	Aarhus	23416	23416	100.0
29	NCR	Skelagervej 15	20773	20773	100.0
81	NCR	Spar K $ ilde{A}f\hat{A}$ , bmand Tornh $ ilde{A}f\hat{A}$ , j	20148	20148	100.0
102	NCR	Aalborg Storcenter Afd	18297	18297	100.0

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

```
select A.weather_main, total_transaction_count, convert(decimal(10,0),COALESCE(inactive,0.0)) as
inactive_count, convert(decimal(10,4),100.0*inactive_count/total_transaction_count) as
inactive_count_percent
from (
       select weather_main, count(*) as total_transaction_count from
etl_bank_schema.fact_atm_trans
       group by weather_main
       having weather_main != "
LEFT JOIN (
       select weather_main, count(*) as inactive
       from etl_bank_schema.fact_atm_trans
       where atm_status = 'Inactive'
       group by weather_main having weather_main != "
) B
ON A.weather_main = B.weather_main
order by inactive_count_percent desc;
```



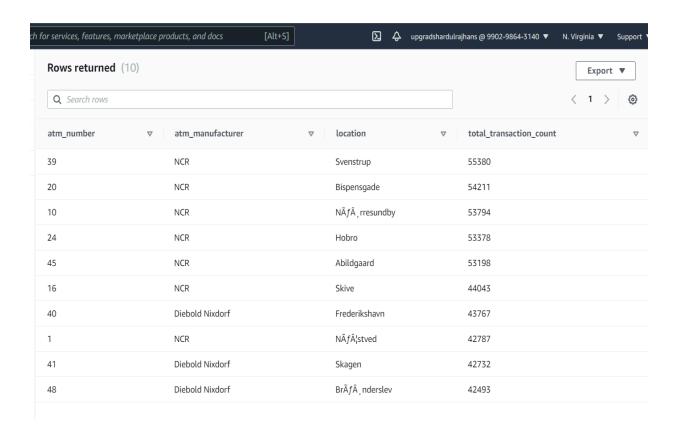








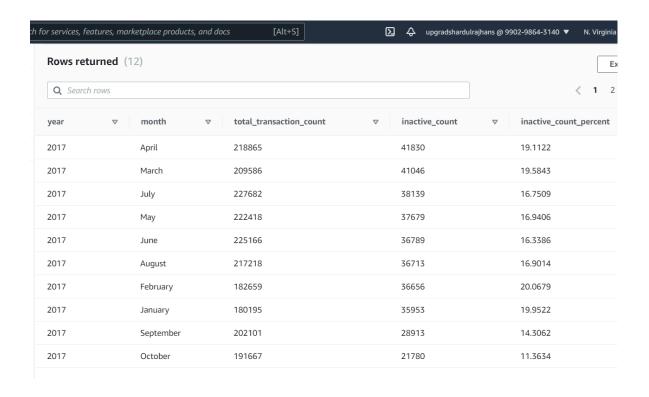
#### 3. Top 10 ATMs with the most number of transactions throughout the year







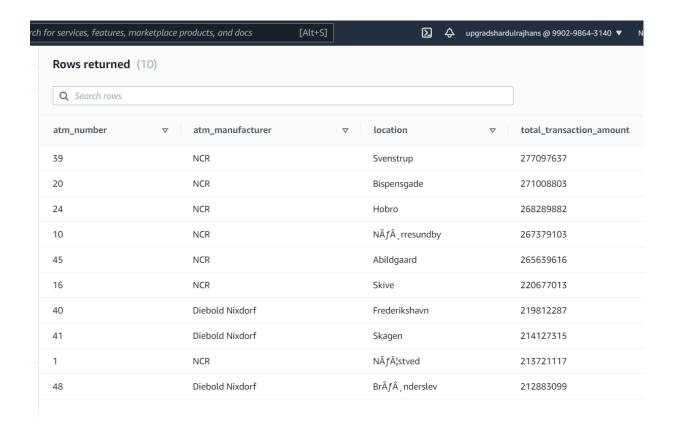
#### 4. Number of overall ATM transactions going inactive per month for each month







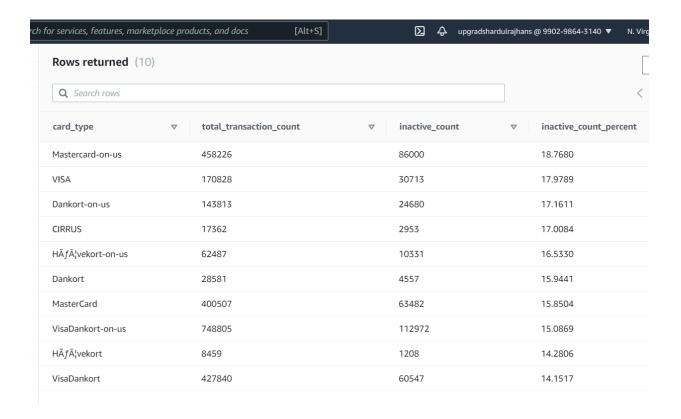
## 5. Top 10 ATMs with the highest total withdrawn amount throughout the year







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





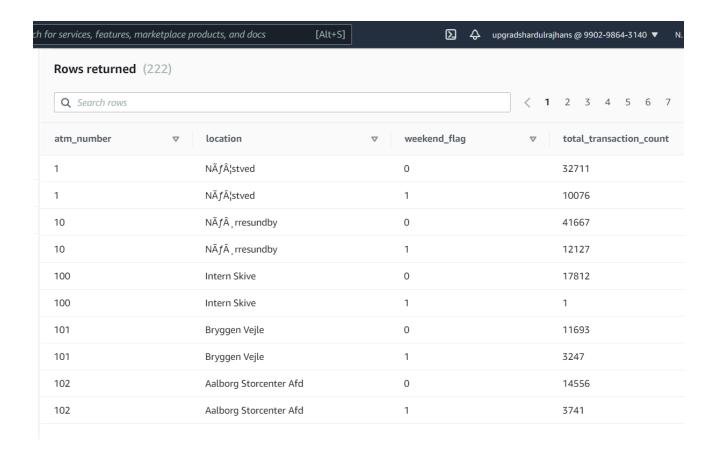


7. Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order this by the ATM\_number, ATM\_manufacturer, location, weekend\_flag and then total\_transaction\_count

select atm.atm\_number, loc.location,

CASE(date.weekday) WHEN 'Saturday' THEN '1' WHEN 'Sunday' THEN '1' ELSE '0' END as weekend\_flag,
count(\*) as total\_transaction\_count

from etl\_bank\_schema.fact\_atm\_trans fact,
etl\_bank\_schema.dim\_atm atm,
etl\_bank\_schema.dim\_location loc,
etl\_bank\_schema.dim\_date date
where fact.atm\_id = atm.atm\_id
and fact.weather\_loc\_id = loc.location\_id and fact.date\_id = date.date\_id
group by atm\_number,location,weekend\_flag
order by atm\_number,location,weekend\_flag;







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

```
select final.atm_number, final.atm_manufacturer, final.location, final.weekday,
final.total_transaction_count
from (
       select factor.atm_number, factor.atm_manufacturer, factor.location, factor.weekday,
factor.total_transaction_count, DENSE_RANK() OVER (partition by factor.location, factor.atm_number,
factor.atm_manufacturer order by factor.total_transaction_count desc) as RANK
       from (
       select atm.atm_number, atm.atm_manufacturer, loc.location, date.weekday,
       count(*) as total_transaction_count
       from etl_bank_schema.fact_atm_trans fact,
       etl_bank_schema.dim_atm atm,
       etl_bank_schema.dim_location loc,
       etl_bank_schema.dim_date date
       where fact.atm_id = atm.atm_id and fact.weather_loc_id = loc.location_id
       and fact.date_id = date.date_id
       group by loc.location, atm.atm_number, atm.atm_manufacturer, date.weekday
       having loc.location = 'Vejgaard'
       ) factor
) final
where RANK = 1
ORDER BY final.total_transaction_count;
```

