### 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 A.\*,(inactive\_count/total\_transaction\_count)\*100 as count\_percent from

(select distinct DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location,

count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location) as total\_transaction\_count,

count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location) as inactive\_count

from redshift\_etlcasestudy.FACT\_ATM\_TRANS

join redshift\_etlcasestudy.DIM\_ATM DA on

DA.atm\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.atm\_id

join redshift\_etlcasestudy.DIM\_LOCATION DL on

DL.location\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.weather\_loc\_id

where atm\_status = 'Inactive'

order by count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location) desc limit 10) A

<Screenshot of the resultant table>

**A picture containing text, computer, computer, screenshot

Description automatically generated**

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

<Query>

<Screenshot of the resultant table>

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

<Query>

select distinct DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location,

count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location, DD.year) as total\_transaction\_count

from redshift\_etlcasestudy.FACT\_ATM\_TRANS

join redshift\_etlcasestudy.DIM\_ATM DA on

DA.atm\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.atm\_id

join redshift\_etlcasestudy.DIM\_LOCATION DL on

DL.location\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.weather\_loc\_id

join redshift\_etlcasestudy.DIM\_DATE DD on

DD.date\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.date\_id

order by count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location, DD.year) desc limit 10

<Screenshot of the resultant table>

A picture containing text, computer, computer, screenshot

Description automatically generated

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

<Query>

<Screenshot of the resultant table>

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 **Top 10 ATMs with the highest total withdrawn amount throughout the year**

<Query>

select distinct DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location,

sum(transaction\_amount) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location) as total\_transaction\_count

from redshift\_etlcasestudy.FACT\_ATM\_TRANS

join redshift\_etlcasestudy.DIM\_ATM DA on

DA.atm\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.atm\_id

join redshift\_etlcasestudy.DIM\_LOCATION DL on

DL.location\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.weather\_loc\_id

join redshift\_etlcasestudy.DIM\_DATE DD on

DD.date\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.date\_id

order by sum(transaction\_amount) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location) desc limit 10

<Screenshot of the resultant table>

A picture containing text, computer, computer, indoor

Description automatically generated

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

<Query>

<Screenshot of the resultant table>

1. **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 distinct DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location,

case when DD.weekday in ('Saturday','Sunday')

then 1

else 0

end weekend\_flag,

count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location, case when DD.weekday in ('Saturday','Sunday')

then 1

else 0

end) as total\_transaction\_count

from redshift\_etlcasestudy.FACT\_ATM\_TRANS

join redshift\_etlcasestudy.DIM\_ATM DA on

DA.atm\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.atm\_id

join redshift\_etlcasestudy.DIM\_LOCATION DL on

DL.location\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.weather\_loc\_id

join redshift\_etlcasestudy.DIM\_DATE DD on

DD.date\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.date\_id

order by DA.atm\_number,atm\_manufacturer ,atm\_location, case when DD.weekday in ('Saturday','Sunday')

then 1

else 0

end,

count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location, case when DD.weekday in ('Saturday','Sunday')

then 1

else 0

end)

desc limit 10

<Screenshot of the resultant table>

A picture containing text, computer, screenshot, computer

Description automatically generated

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

<Query>

WITH B AS (

select A.\*,

RANK() OVER(PARTITION BY A.atm\_number order by A.total\_transaction\_count desc) as Rank

from

(select distinct

DA.atm\_number , DA.atm\_manufacturer , DL.atm\_location, DD.weekday,

count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location, DD.weekday) as total\_transaction\_count

from redshift\_etlcasestudy.FACT\_ATM\_TRANS

join redshift\_etlcasestudy.DIM\_ATM DA on

DA.atm\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.atm\_id

join redshift\_etlcasestudy.DIM\_LOCATION DL on

DL.location\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.weather\_loc\_id

join redshift\_etlcasestudy.DIM\_DATE DD on

DD.date\_id = redshift\_etlcasestudy.FACT\_ATM\_TRANS.date\_id

where DL.atm\_location = 'Vejgaard'

order by count(trans\_id) over (partition by DA.atm\_number, DA.atm\_manufacturer , DL.atm\_location, DD.weekday) desc) A

)

SELECT atm\_number, atm\_manufacturer , atm\_location, weekday, total\_transaction\_count

FROM B

where Rank = 1;

<Screenshot of the resultant table>

Graphical user interface, text, application, email

Description automatically generated