**Solving analytical queries on RedShift Cluster**

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

select da.atm\_number, da.atm\_manufacturer, dl.location,

count(da.atm\_id) as total\_transaction\_count, count(tf.atm\_status) as inactive\_count,

round(inactive\_count \* 100.0 / count(\*), 1) as inactive\_count\_percent

from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_atm da on tf.atm\_id = da.atm\_id

join atm\_new\_trans.dim\_location dl on da.location\_id = dl.location\_id

where tf.atm\_status = 'Inactive'

group by da.atm\_number, da.atm\_manufacturer, dl.location

order by inactive\_count desc limit 10;

Graphical user interface, application

Description automatically generated

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

select dt1.weather\_main, dt1.total\_counts,

case

when dt2.inactive\_counts is null then 0 else dt2.inactive\_counts

end as inactive\_counts,

case

when round(dt2.inactive\_counts \* 100.0 / dt1.total\_counts, 4) is null then 0 else round(dt2.inactive\_counts \* 100.0 / dt1.total\_counts, 4)

end as inactive\_count\_percent

from (select weather\_main, count(\*) as total\_counts from atm\_new\_trans.transaction\_fact where weather\_main <> '' group by weather\_main) dt1

left join (select weather\_main, count(\*) as inactive\_counts

from atm\_new\_trans.transaction\_fact where atm\_status = 'Inactive' and weather\_main <> ''

group by weather\_main) dt2 on dt1.weather\_main = dt2.weather\_main order by inactive\_count\_percent desc;

Graphical user interface

Description automatically generated with medium confidence

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

select da.atm\_id, da.atm\_manufacturer, dl.location, count(\*) as transaction\_count from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_atm da on tf.atm\_id = da.atm\_id

join atm\_new\_trans.dim\_location dl on dl.location\_id = da.location\_id

group by da.atm\_id, da.atm\_manufacturer, dl.location order by transaction\_count desc limit 10;

Graphical user interface, text, application, email

Description automatically generated

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

month

select dd.year, dd.month, count(\*) as total\_transaction\_count,

count(case

when tf.atm\_status = 'Inactive' then 1

end) as inactive\_count,

round(inactive\_count \* 100.0 / total\_transaction\_count, 4) as inactive\_count\_percent

from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_atm da on da.atm\_id = tf.atm\_id

join atm\_new\_trans.dim\_date dd on tf.date\_id = dd.date\_id

group by dd.year, dd.month

order by dd.month;

Table

Description automatically generated

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

select da.atm\_id, da.atm\_manufacturer, dl.location, sum(tf.transaction\_amount) as total\_trans\_amount from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_atm da on tf.atm\_id = da.atm\_id

join atm\_new\_trans.dim\_location dl on da.location\_id = dl.location\_id

group by da.atm\_id, da.atm\_manufacturer, dl.location order by total\_trans\_amount desc limit 10;

Graphical user interface, application

Description automatically generated

1. Number of failed ATM transactions across various card types

select ct.card\_type, count(\*) as total\_transaction\_count,

count(case

when tf.atm\_status = 'Inactive' then 1

end) as inactive\_count,

round(inactive\_count \* 100.0 / total\_transaction\_count, 4) as inactive\_count\_percent

from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_card\_type ct on tf.card\_type\_id = ct.card\_type\_id

group by ct.card\_type order by inactive\_count\_percent desc;

Table

Description automatically generated with low confidence

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

select da.atm\_number, da.atm\_manufacturer, dl.location,

case

when dd.weekday in ('Saturday', 'Sunday') then 1 else 0

end as weekend\_flag,

count(\*) as total\_transaction\_count

from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_atm da on tf.atm\_id = da.atm\_id

join atm\_new\_trans.dim\_date dd on tf.date\_id = dd.date\_id

join atm\_new\_trans.dim\_location dl on da.location\_id = dl.location\_id

group by da.atm\_id, da.atm\_number, da.atm\_manufacturer, dl.location, weekend\_flag

order by da.atm\_number, weekend\_flag, total\_transaction\_count;

Graphical user interface

Description automatically generated with medium confidence

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

drop view if exists atm\_new\_trans.weekday\_trans;

create view atm\_new\_trans.weekday\_trans as

select dd.weekday, count(\*)

from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_date dd on dd.date\_id = tf.date\_id

join atm\_new\_trans.dim\_atm da on tf.atm\_id = da.atm\_id

join atm\_new\_trans.dim\_location dl on dl.location\_id = da.location\_id

where dl.location = 'Vejgaard' group by dd.weekday;

select da.atm\_id, da.atm\_manufacturer, dd.weekday,

dl.location, count(\*) as total\_transaction\_count

from atm\_new\_trans.transaction\_fact tf

join atm\_new\_trans.dim\_date dd on dd.date\_id = tf.date\_id

join atm\_new\_trans.dim\_atm da on da.atm\_id = tf.atm\_id

join atm\_new\_trans.dim\_location dl on dl.location\_id = da.location\_id

where dl.location = 'Vejgaard' and dd.weekday = (select weekday

from atm\_new\_trans.weekday\_trans

where count = (select max(count) from atm\_new\_trans.weekday\_trans))

group by da.atm\_id, da.atm\_manufacturer, dd.weekday, dl.location

order by total\_transaction\_count;

Graphical user interface, application, Word

Description automatically generated