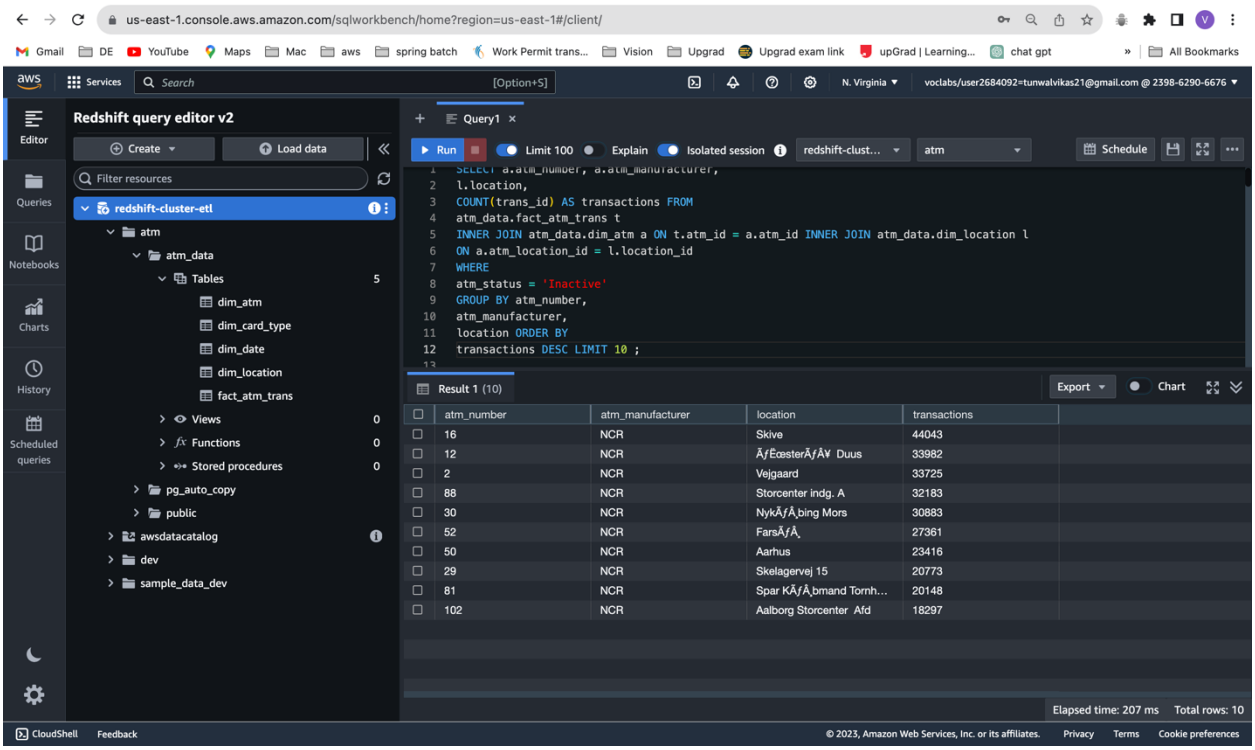
Solving analytical queries on Redshift Cluster

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.

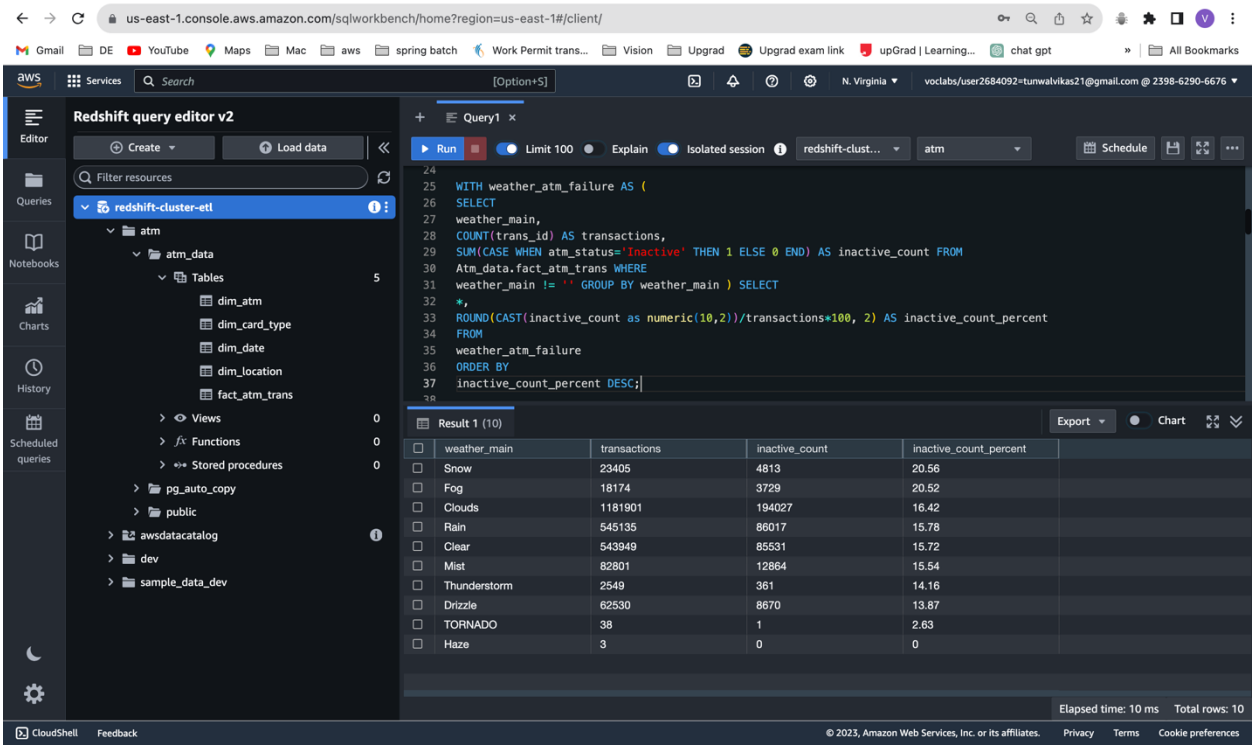
1. Top 10 ATMs where most transactions are in the 'inactive' state

```
SELECT a.atm_number,
a.atm_manufacturer,
l.location,
COUNT(trans_id) AS transactions FROM
atm_data.fact_atm_trans t
INNER JOIN atm_data.dim_atm a ON t.atm_id = a.atm_id
INNER JOIN atm_data.dim_location l ON a.atm_location_id = l.location_id
WHERE atm_status = 'Inactive'
GROUP BY atm_number, atm_manufacturer, location
ORDER BY transactions DESC
LIMIT 10 ;
```



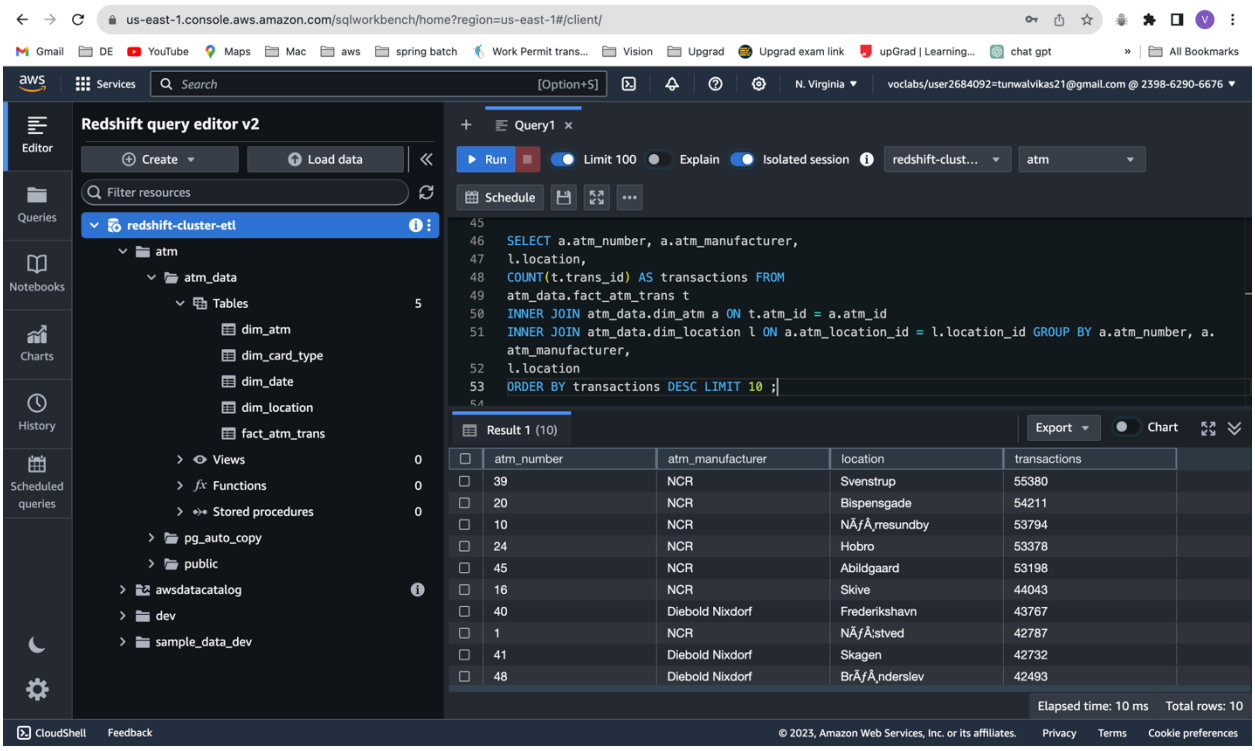
2. Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions

```
WITH weather_atm_failure AS (
SELECT
weather_main,
COUNT(trans_id) AS transactions,
SUM(CASE WHEN atm_status='Inactive' THEN 1 ELSE 0 END) AS inactive_count
FROM
Atm_data.fact_atm_trans WHERE
weather_main != '' GROUP BY weather_main ) SELECT
*,
ROUND(CAST(inactive_count as numeric(10,2))/transactions*100, 2) AS
inactive_count_percent
FROM
weather_atm_failure
ORDER BY
inactive_count_percent DESC;
```



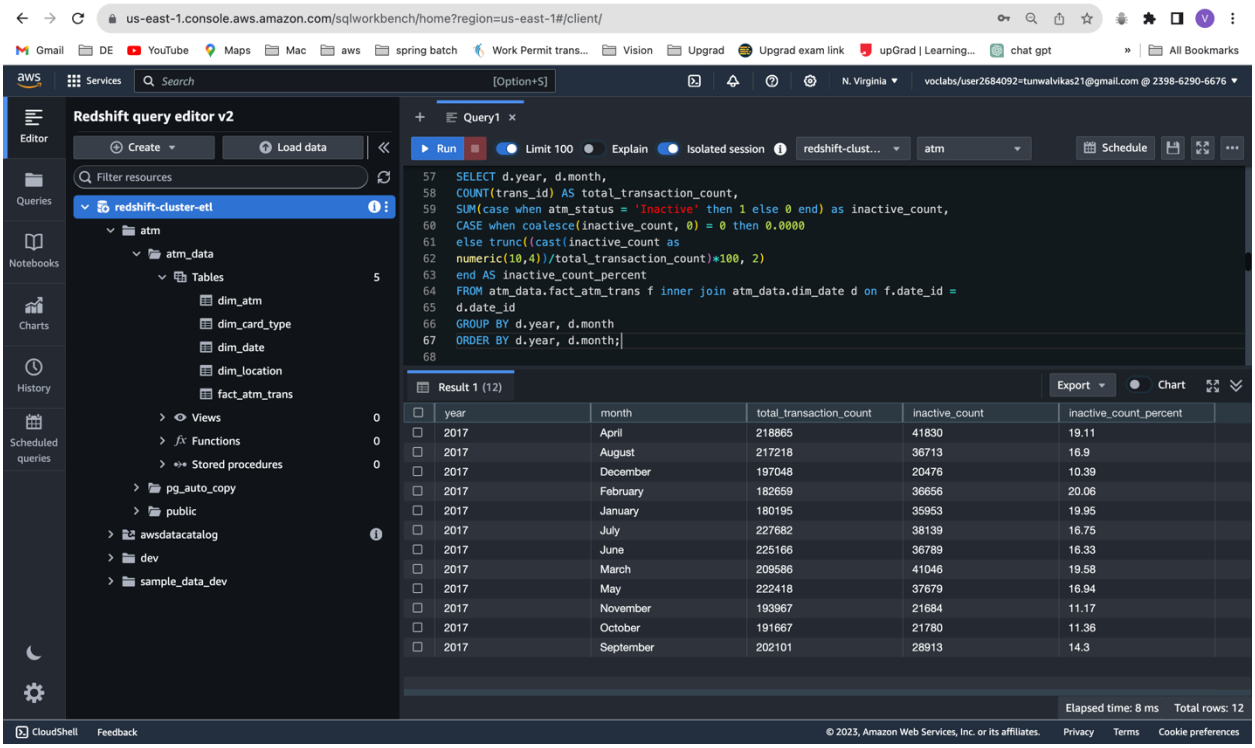
3. Top 10 ATMs with the most number of transactions throughout the year

```
SELECT a.atm_number,
a.atm_manufacturer,
l.location,
COUNT(t.trans_id) AS transactions
FROM
atm_data.fact_atm_trans t
INNER JOIN atm_data.dim_atm a ON t.atm_id = a.atm_id
INNER JOIN atm_data.dim_location l ON a.atm_location_id = l.location_id
GROUP BY a.atm_number, a.atm_manufacturer, l.location
ORDER BY transactions DESC LIMIT 10 ;
```



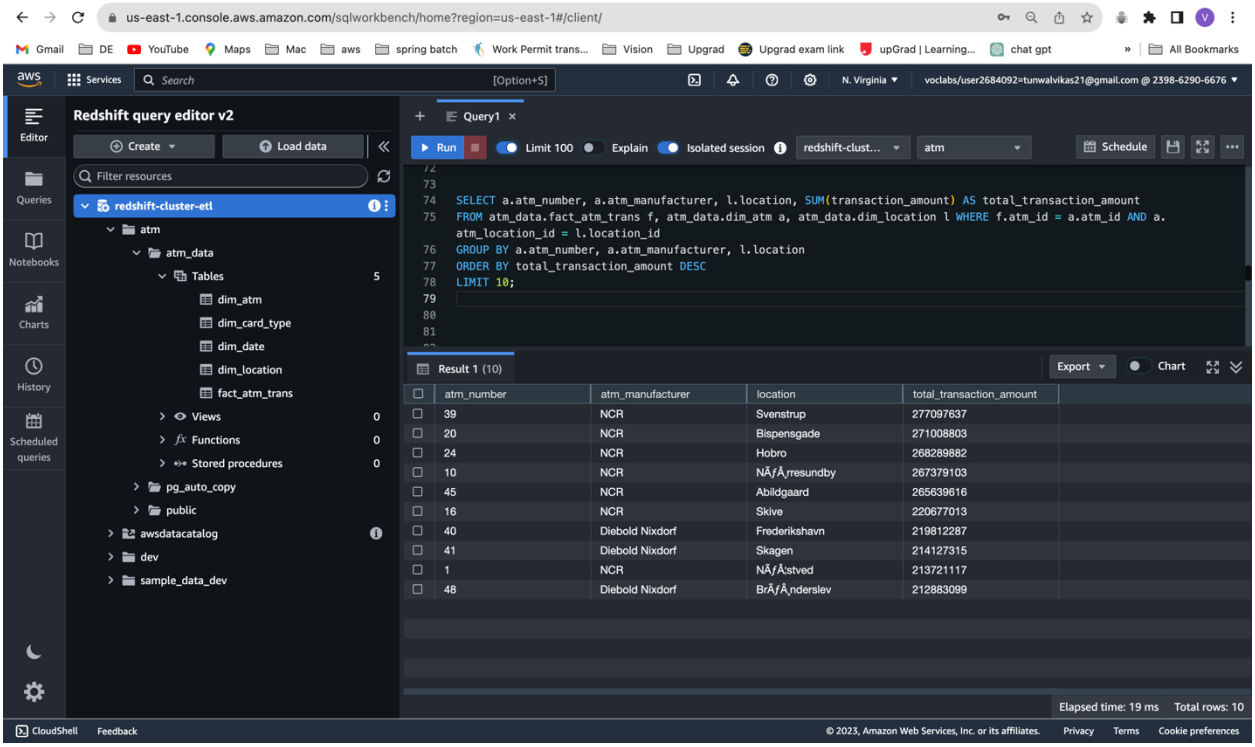
4. Number of overall ATM transactions going inactive per month for each month

```
SELECT d.year, d.month,
COUNT(trans_id) AS total_transaction_count,
SUM(case when atm_status = 'Inactive' then 1 else 0 end) as
inactive_count,
CASE when coalesce(inactive_count, 0) = 0 then 0.0000
else trunc((cast(inactive_count as
numeric(10,4))/total_transaction_count)*100, 2)
end AS inactive_count_percent
FROM atm_data.fact_atm_trans f inner join atm_data.dim_date d on f.date_id
=
d.date_id
GROUP BY d.year, d.month
ORDER BY d.year, d.month;
```



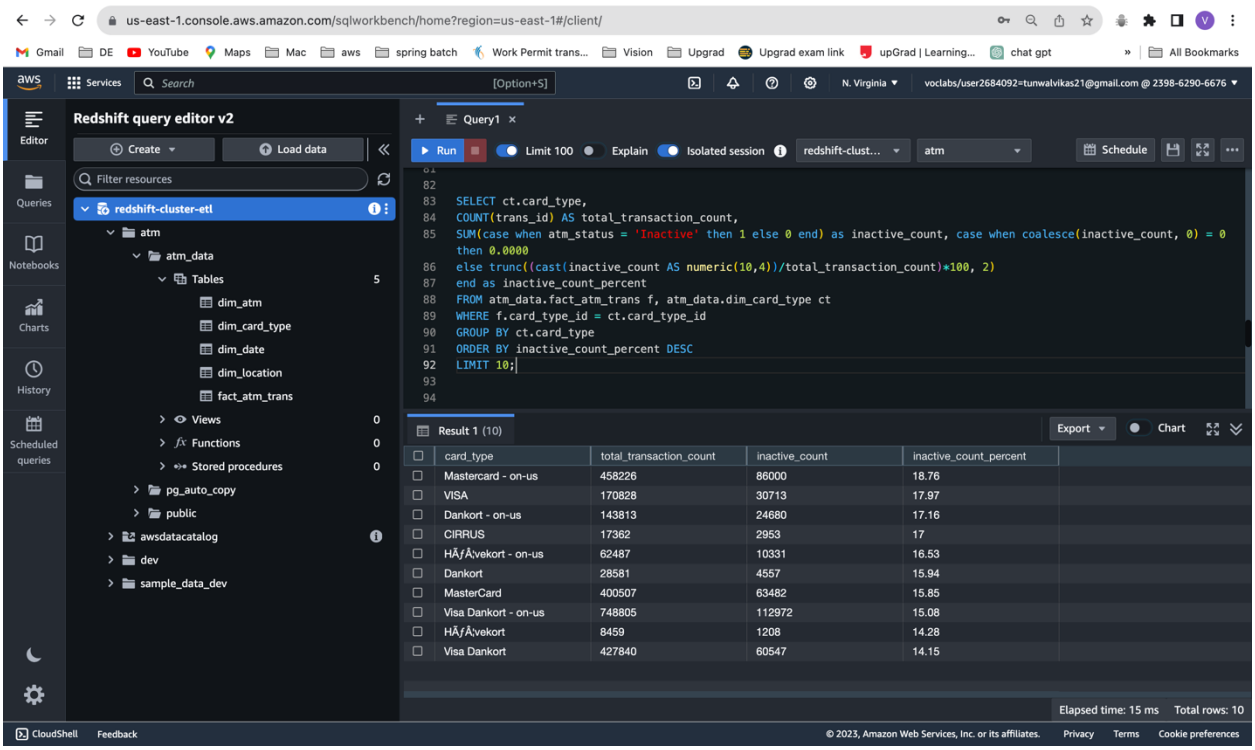
5. Top 10 ATMs with the highest total withdrawn amount throughout the year

```
SELECT a.atm_number, a.atm_manufacturer, l.location,
SUM(transaction_amount) AS total_transaction_amount
FROM atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location
l WHERE f.atm_id = a.atm_id AND a.atm_location_id = l.location_id
GROUP BY a.atm_number, a.atm_manufacturer, l.location
ORDER BY total_transaction_amount DESC
LIMIT 10;
```



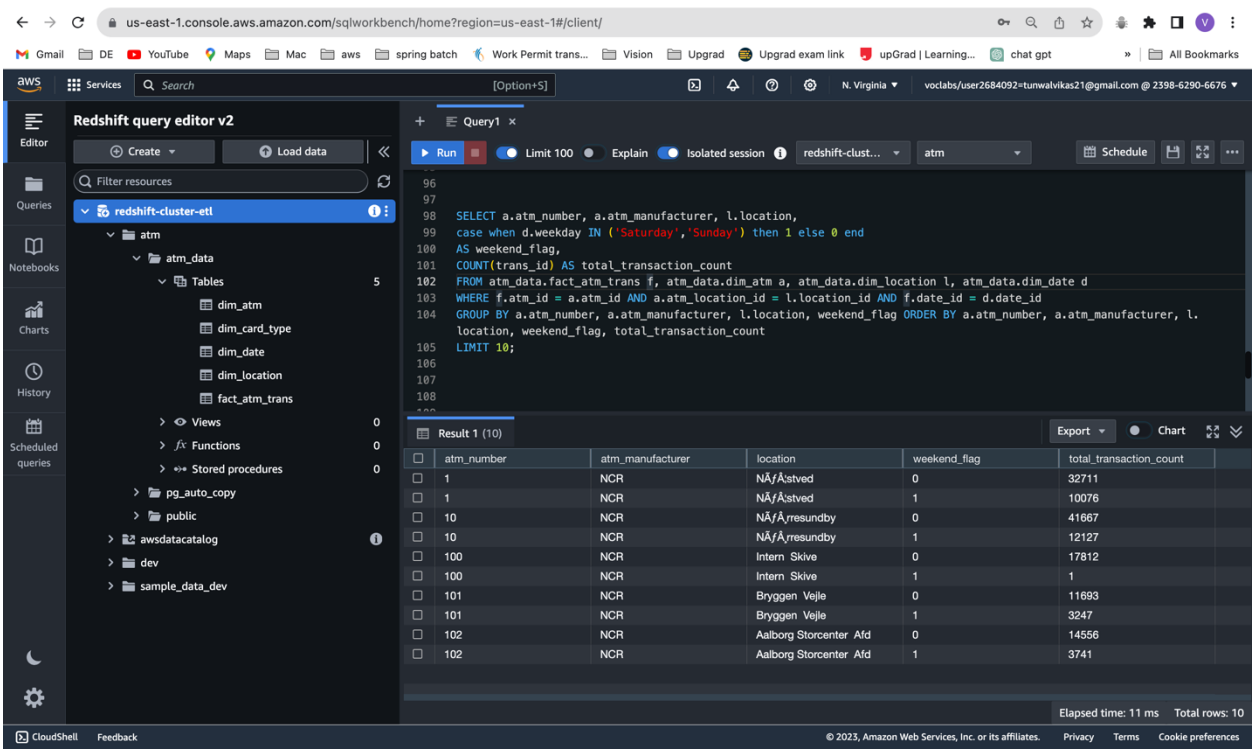
6. Number of failed ATM transactions across various card types

```
SELECT ct.card_type,
COUNT(trans_id) AS total_transaction_count,
SUM(case when atm_status = 'Inactive' then 1 else 0 end) as
inactive_count, case when coalesce(inactive_count, 0) = 0 then 0.0000
else trunc((cast(inactive_count AS
numeric(10,4))/total_transaction_count)*100, 2)
end as inactive_count_percent
FROM atm_data.fact_atm_trans f, atm_data.dim_card_type ct
WHERE f.card_type_id = ct.card_type_id
GROUP BY ct.card_type
ORDER BY inactive_count_percent DESC
LIMIT 10;
```



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 a.atm_number, a.atm_manufacturer, l.location,
case when d.weekday IN ('Saturday','Sunday') then 1 else 0 end
AS weekend_flag,
COUNT(trans_id) AS total_transaction_count
FROM atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location
l, atm_data.dim_date d
WHERE f.atm_id = a.atm_id AND a.atm_location_id = l.location_id AND
f.date_id = d.date_id
GROUP BY a.atm_number, a.atm_manufacturer, l.location, weekend_flag ORDER
BY a.atm_number, a.atm_manufacturer, l.location, weekend_flag,
total_transaction_count
LIMIT 10;
```



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

```
SELECT a.atm_number, a.atm_manufacturer, l.location, d.weekday,
COUNT(trans_id) AS total_transaction_count
FROM atm_data.fact_atm_trans f inner JOIN atm_data.dim_atm a ON f.atm_id =
a.atm_id
inner JOIN atm_data.dim_location l ON a.atm_location_id = l.location_id
inner JOIN atm_data.dim_date d ON f.date_id = d.date_id
WHERE l.location = 'Vejgaard' AND d.weekday IN
( SELECT d.weekday
FROM atm_data.fact_atm_trans f inner JOIN atm_data.dim_date d
ON f.date_id = d.date_id
inner JOIN atm_data.dim_location l ON f.weather_loc_id = l.location_id
WHERE l.location = 'Vejgaard'
GROUP BY d.weekday
ORDER BY COUNT(f.trans_id) DESC
LIMIT 1 )
GROUP BY a.atm_number, a.atm_manufacturer, l.location, d.weekday
ORDER BY total_transaction_count;
```