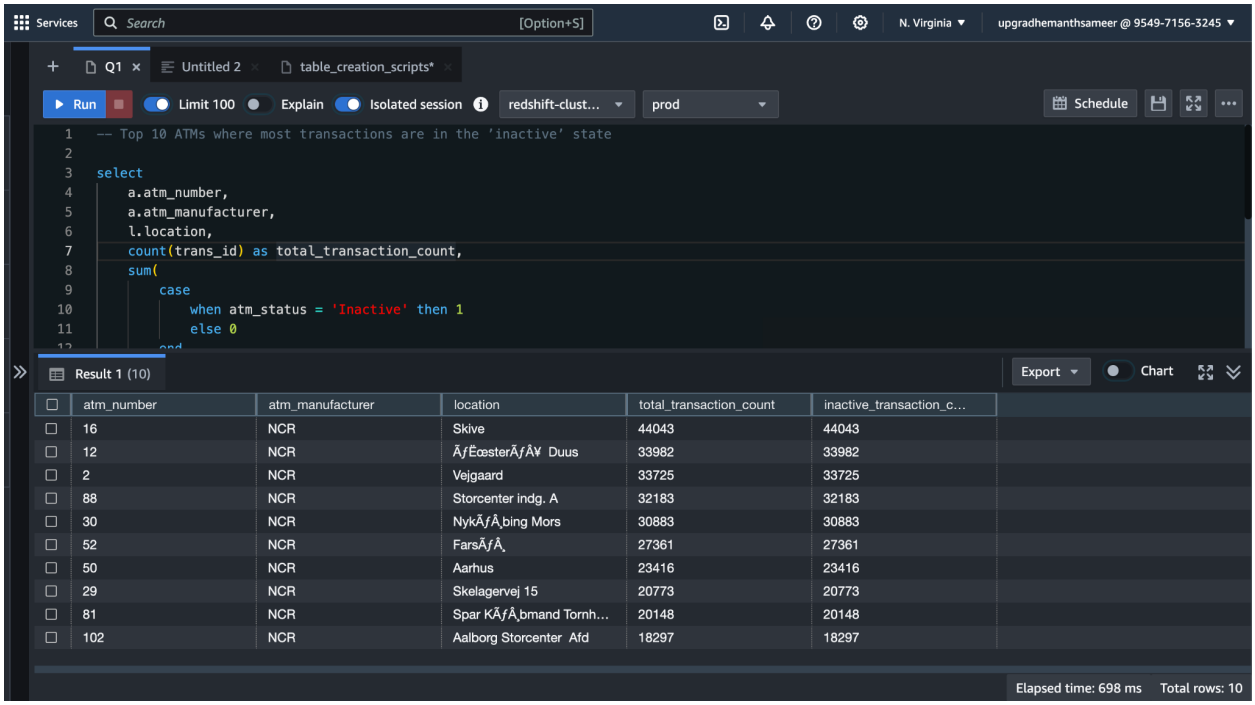
Solving analytical queries on Redshift Cluster

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.

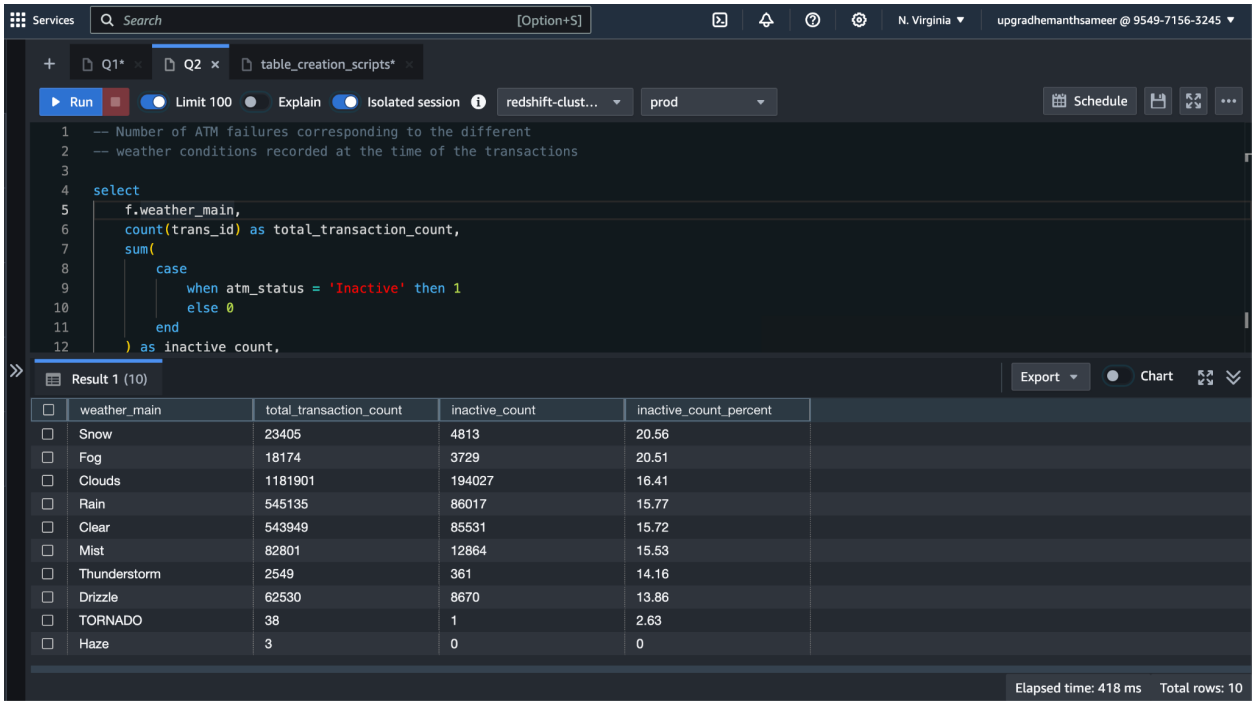
1. **Top 10 ATMs where most transactions are in the 'inactive' state**

```
select
   a.atm_number,
   a.atm_manufacturer,
   l.location,
   count(trans_id) as total_transaction_count,
   sum(
       case
           when atm_status = 'Inactive' then 1
           else 0
       end
   ) as inactive_transaction_count
from
   etl_bankatm_data.fact_atm_trans f,
   etl_bankatm_data.dim_atm a,
   etl_bankatm_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
   inactive_transaction_count desc
limit 10;
```



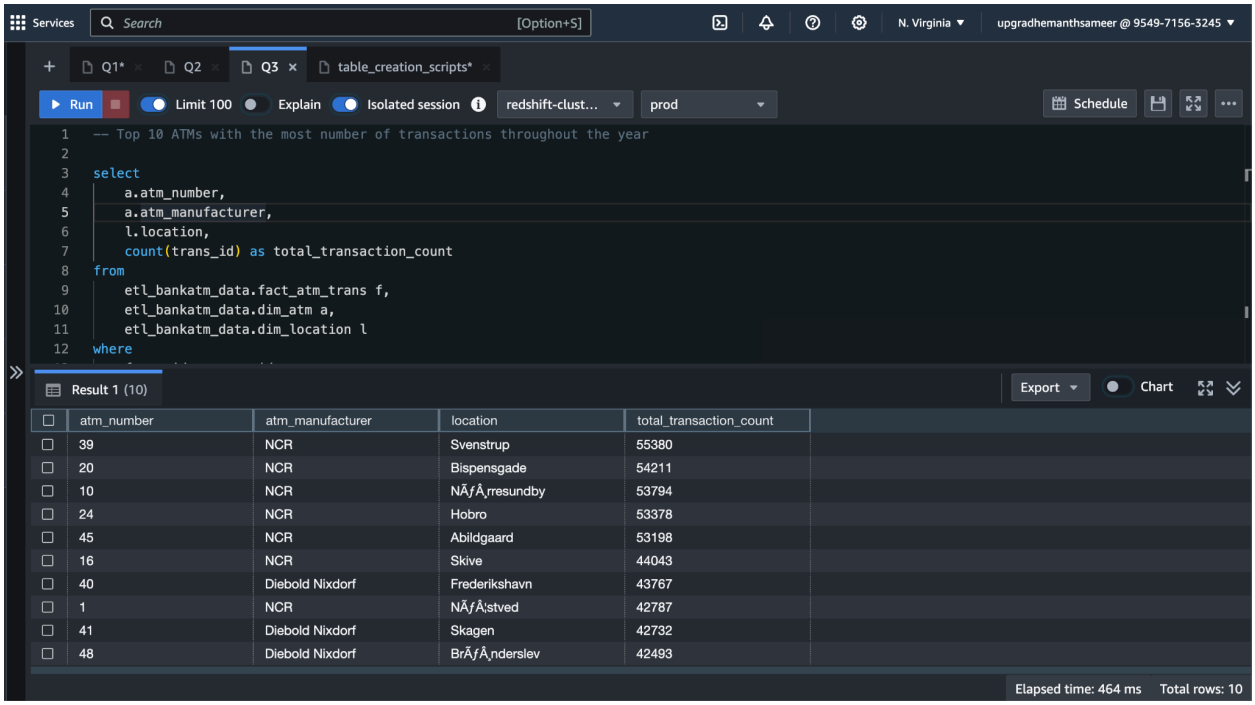
2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

```
select
   f.weather_main,
   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
   etl_bankatm_data.fact_atm_trans f
where f.weather_main != ''
group by f.weather_main
order by
   inactive_count_percent desc
limit 10;
```



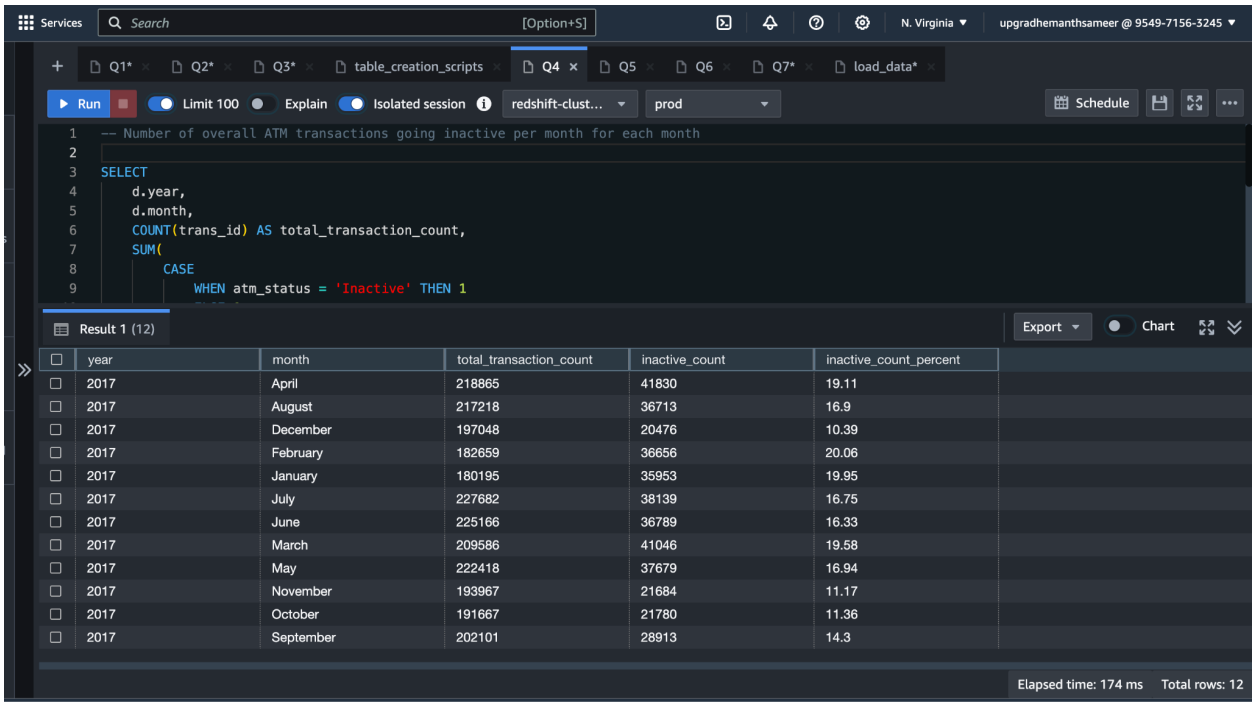
3. Top 10 ATMs with the most number of transactions throughout the year

```sql
select
   a.atm_number,
   a.atm_manufacturer,
   l.location,
   count(trans_id) as total_transaction_count
from
   etl_bankatm_data.fact_atm_trans f,
   etl_bankatm_data.dim_atm a,
   etl_bankatm_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_count desc
limit 10;
```



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

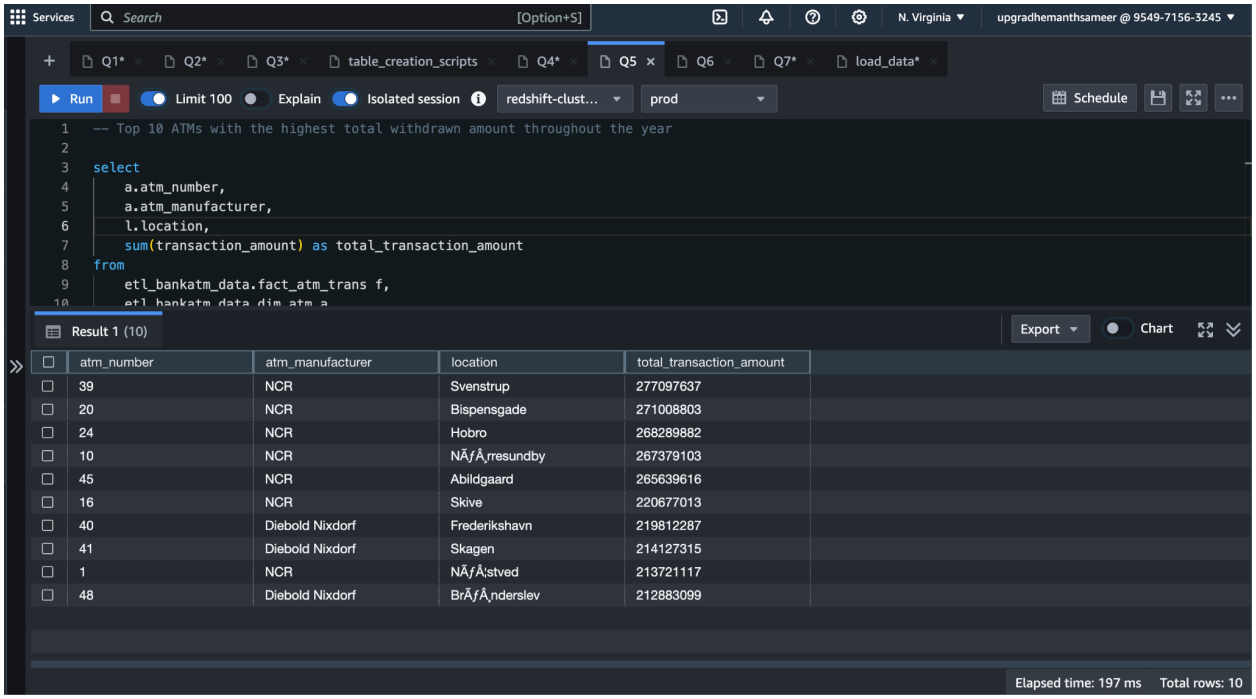
```
select
   c.year,
   c.month,
   c.transaction_count,
   d.inactive_count,
   CAST(
       trunc(
           100.0 * d.inactive_count / c.transaction_count,
           2
       ) AS NUMERIC(10, 4)
   ) as inactive_count_percent
from (
       select
           a.year,
           a.month,
           count(b.trans_id) as transaction_count
       from
           etl_bankatm_data.dim_date a,
           etl_bankatm_data.FACT_ATM_TRANS b
       where a.date_id = b.date_id
       group by a.month, a.year
   ) c
   left join (
       select
           a.year,
           a.month,
           count(b.atm_status) as inactive_count
       from
           etl_bankatm_data.dim_date a,
           etl_bankatm_data.FACT_ATM_TRANS b
       where
           a.date_id = b.date_id
           and b.atm_status = 'Inactive'
       group by
           a.month,
           a.year
   ) d on c.year = d.year
   and c.month = d.month
order by c.year, c.month;
```



year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.11
2017	August	217218	36713	16.9
2017	December	197048	20476	10.39
2017	February	182659	36656	20.06
2017	January	180195	35953	19.95
2017	July	227682	38139	16.75
2017	June	225166	36789	16.33
2017	March	209586	41046	19.58
2017	May	222418	37679	16.94
2017	November	193967	21684	11.17
2017	October	191667	21780	11.36
2017	September	202101	28913	14.3

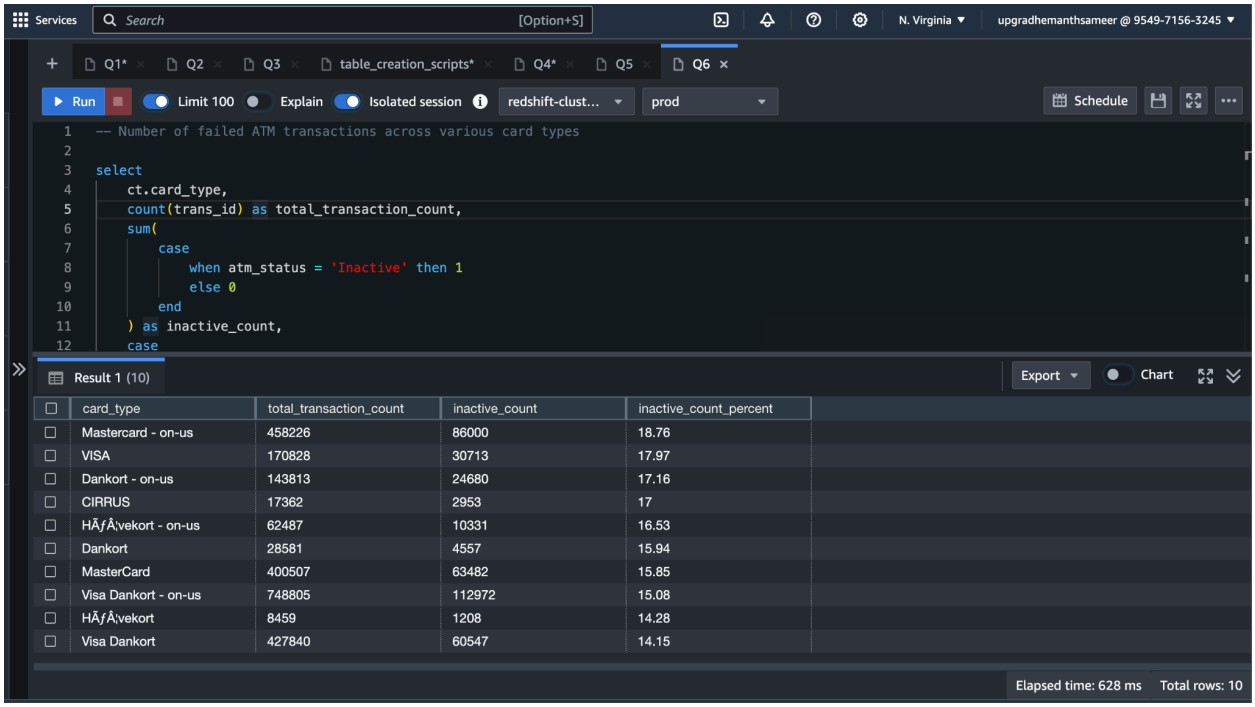
5. **Top 10 ATMs with the highest total withdrawn amount throughout the year**

```sql
select
   a.atm_number,
   a.atm_manufacturer,
   l.location,
   sum(transaction_amount) as total_transaction_amount
from
   etl_bankatm_data.fact_atm_trans f,
   etl_bankatm_data.dim_atm a,
   etl_bankatm_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**

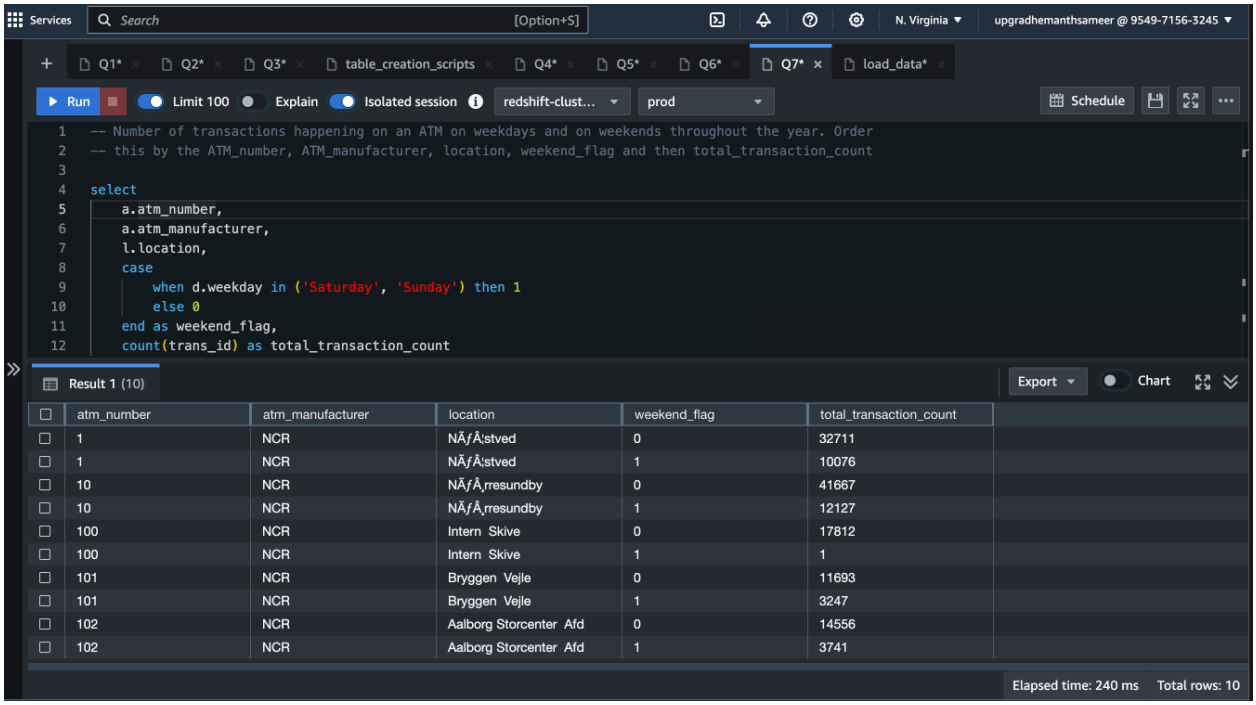
```
select
   ct.card_type,
   count(trans_id) as total_transaction_count,
   sum(
       case
           when atm_status = 'Inactive' then 1
           else 0
       end
   ) as inactive_count,
   case
       when coalesce(inactive_count, 0) = 0 then 0.0000
       else trunc( (
               cast(
                   inactive_count as numeric(10, 4)
               ) / total_transaction_count
           ) * 100,
           2
       )
   end as inactive_count_percent
from
   etl_bankatm_data.fact_atm_trans f,
   etl_bankatm_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;
```



card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.76
VISA	170828	30713	17.97
Dankort - on-us	143813	24680	17.16
CIRRUS	17362	2953	17
HÃƒÂ¦vekort - on-us	62487	10331	16.53
Dankort	28581	4557	15.94
MasterCard	400507	63482	15.85
Visa Dankort - on-us	748805	112972	15.08
HÃƒÂ¦vekort	8459	1208	14.28
Visa Dankort	427840	60547	14.15

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
   etl_bankatm_data.fact_atm_trans f,
   etl_bankatm_data.dim_atm a,
   etl_bankatm_data.dim_location l,
   etl_bankatm_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
   etl_bankatm_data.fact_atm_trans f
   inner join etl_bankatm_data.dim_atm a on f.atm_id = a.atm_id
   inner join etl_bankatm_data.dim_location l on a.atm_location_id = l.location_id
   inner join etl_bankatm_data.dim_date d on f.date_id = d.date_id
where
   l.location = 'Vejgaard'
   and d.weekday in (
       select d.weekday
       from
           etl_bankatm_data.fact_atm_trans f
           inner join etl_bankatm_data.dim_date d on f.date_id = d.date_id
           inner join etl_bankatm_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;
```