

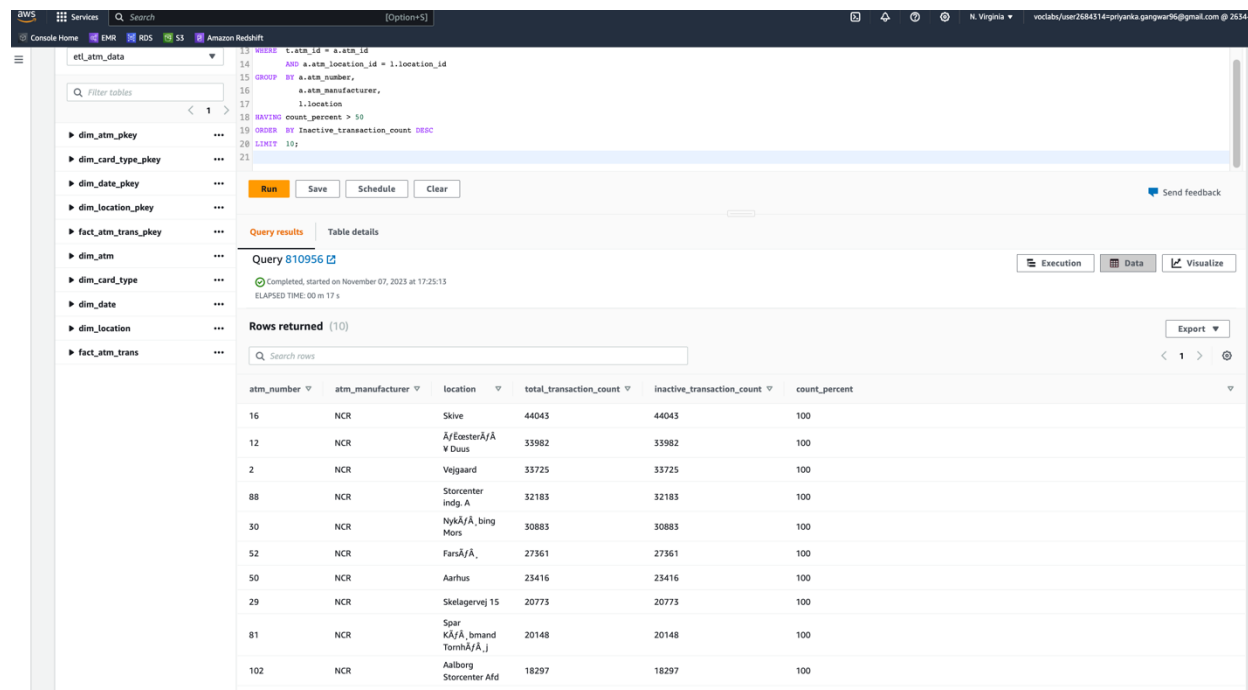
Solving analytical queries on Redshift Cluster

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

<Query>

```
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,
    (Inactive_transaction_count / total_transaction_count ) * 100 AS count_percent
FROM
    etl_atm_data.fact_atm_trans t,
    etl_atm_data.dim_atm a,
    etl_atm_data.dim_location l
WHERE
    t.atm_id = a.atm_id AND
    a.atm_location_id = l.location_id
GROUP BY
    a.atm_number,
    a.atm_manufacturer,
    l.location
HAVING
    count_percent > 50
ORDER BY
    Inactive_transaction_count DESC
LIMIT 10;
```

<Screenshot of the query results>



The screenshot shows the Amazon Redshift console interface. On the left, there is a sidebar with a search bar and a list of tables including dim_atm_pikey, dim_card_type_pikey, dim_date_pikey, dim_location_pikey, fact_atm_trans_pikey, dim_atm, dim_card_type, dim_date, dim_location, and fact_atm_trans. The main area displays a SQL query for Query 810956. The query is as follows:

```

14  SELECT t.atm_id = a.atm_id
15  FROM etl_atm_data
16  GROUP BY a.atm_location_id = 1.location_id,
17         a.atm_manufacturer,
18         1.location
19  HAVING count_percent > 50
20 ORDER BY inactive_transaction_count DESC
21 LIMIT 10;

```

Below the query, the results are shown in a table with 6 columns: atm_number, atm_manufacturer, location, total_transaction_count, inactive_transaction_count, and count_percent. The results are as follows:

atm_number	atm_manufacturer	location	total_transaction_count	inactive_transaction_count	count_percent
16	NCR	Skive	44043	44043	100
12	NCR	År/EaesterÅrÅ V Daus	33982	33982	100
2	NCR	Vejgaard	33725	33725	100
88	NCR	Storcenter Indg. A	32183	32183	100
30	NCR	NykÅrÅ, bing Mors	30883	30883	100
52	NCR	FarsÅrÅ,	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
81	NCR	Spar KÅrÅ, bmand TornhÅrÅ J	20148	20148	100
102	NCR	Aalborg Storcenter Ård	18297	18297	100

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

<Query>

```

SELECT
    t.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_atm_data.fact_atm_trans t
WHERE
    t.weather_main != ''
GROUP BY

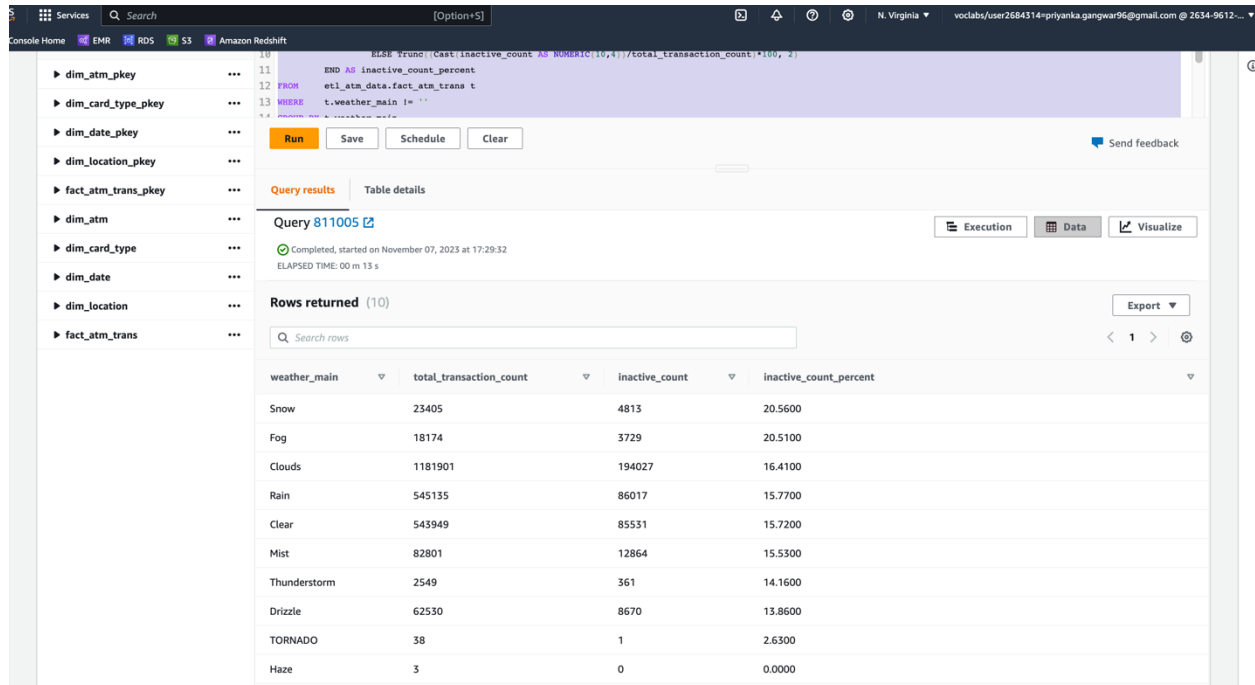
```

```

t.weather_main
ORDER BY
    inactive_count_percent DESC
LIMIT 10;

```

<Screenshot of the query results>



Query 811005

Completed, started on November 07, 2023 at 17:29:32
ELAPSED TIME: 00 m 13 s

Rows returned (10)

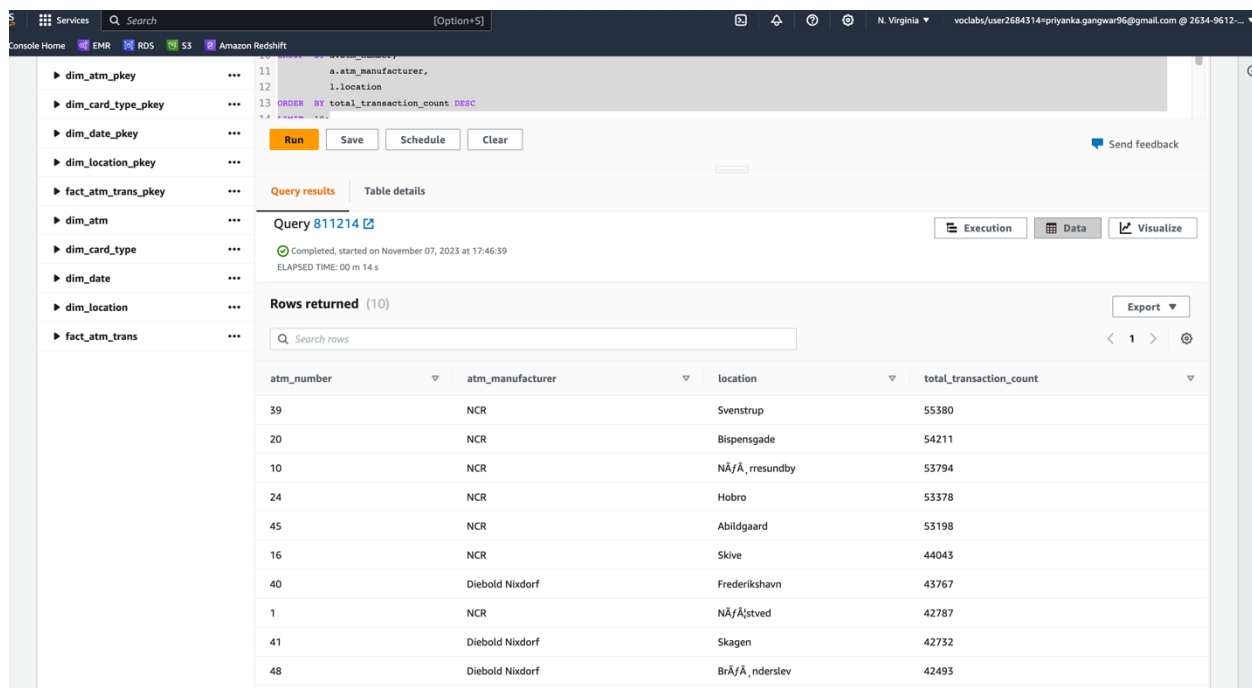
weather_main	total_transaction_count	inactive_count	inactive_count_percent
Snow	23405	4813	20.5600
Fog	18174	3729	20.5100
Clouds	1181901	194027	16.4100
Rain	545135	86017	15.7700
Clear	543949	85531	15.7200
Mist	82801	12864	15.5300
Thunderstorm	2549	361	14.1600
Drizzle	62530	8670	13.8600
TORNADO	38	1	2.6300
Haze	3	0	0.0000

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

<Query>

```
SELECT
    a.atm_number,
    a.atm_manufacturer,
    l.location,
    Count(trans_id) AS total_transaction_count
FROM
    etl_atm_data.fact_atm_trans t,
    etl_atm_data.dim_atm a,
    etl_atm_data.dim_location l
WHERE
    t.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;
```

<Screenshot of the query results>



The screenshot shows the Amazon Redshift console interface. On the left, there is a sidebar with a list of database objects including dimension and fact tables. The main area displays the SQL query that was executed, followed by buttons for 'Run', 'Save', 'Schedule', and 'Clear'. Below the query, the 'Query results' tab is active, showing 'Query 811214' which was completed on November 07, 2023. The results are presented as a table with 10 rows, sorted by 'total_transaction_count' in descending order. The columns are 'atm_number', 'atm_manufacturer', 'location', and 'total_transaction_count'.

atm_number	atm_manufacturer	location	total_transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÅ/Å, resundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÅ/Åstved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÅ/Å, nderslev	42493

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

<Query>

```
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
    etl_atm_data.fact_atm_trans t
INNER JOIN
    etl_atm_data.dim_date d
ON
    t.date_id = d.date_id
GROUP BY
    d.year,
    d.month
ORDER BY
    d.year, d.month;
```

<Screenshot of the query results>

▶ dim_date_pkey

...

▶ dim_location_pkey

...

▶ fact_atm_trans_pkey

...

▶ dim_atm

...

▶ dim_card_type

...

▶ dim_date

...

▶ dim_location

...

▶ fact_atm_trans

...

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query 811285

Execution

Data

Visualize

Completed, started on November 07, 2023 at 17:52:12

ELAPSED TIME: 00 m 15 s

Rows returned (12)

Export

Search rows

< 1 2 > ⌕

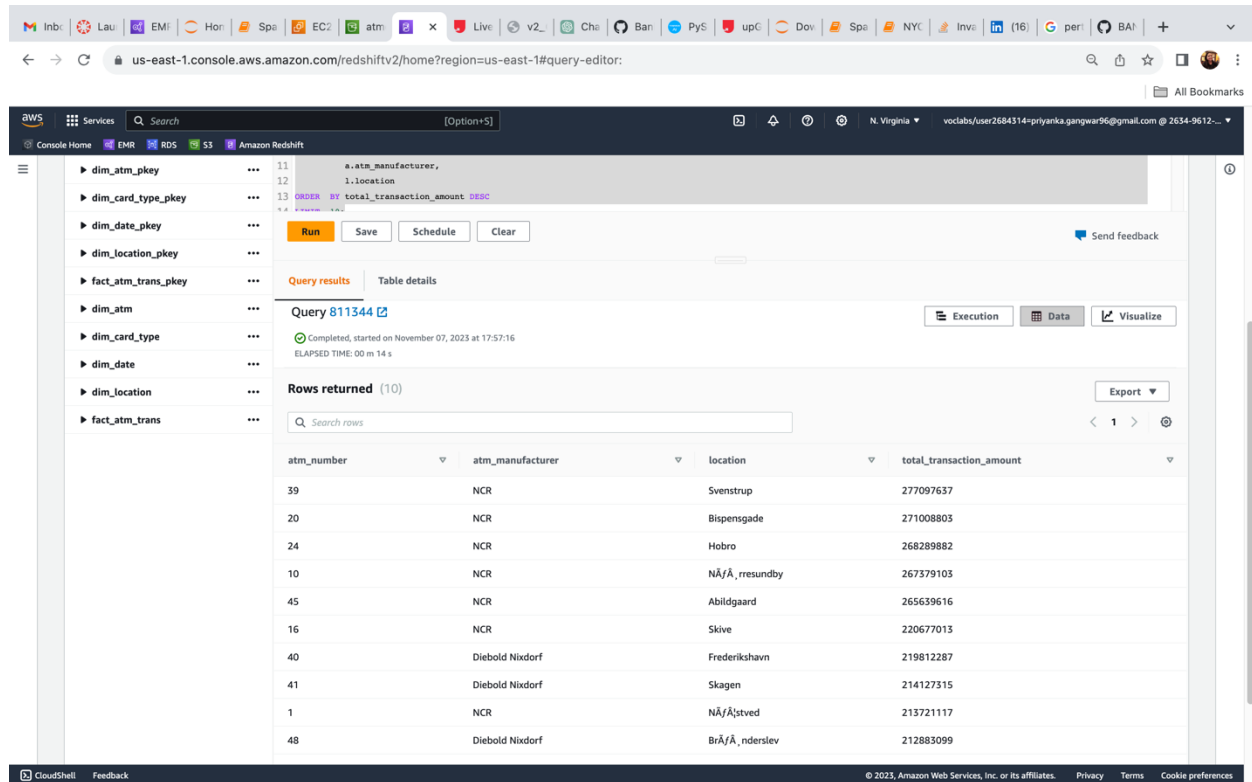
year	month	total_transaction_count	inactive_count	inactive_count_percent
2017	April	218865	41830	19.1100
2017	August	217218	36713	16.9000
2017	December	197048	20476	10.3900
2017	February	182659	36656	20.0600
2017	January	180195	35953	19.9500
2017	July	227682	38139	16.7500
2017	June	225166	36789	16.3300
2017	March	209586	41046	19.5800
2017	May	222418	37679	16.9400
2017	November	193967	21684	11.1700

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

<Query>

```
SELECT
    a.atm_number,
    a.atm_manufacturer,
    l.location,
    Sum(transaction_amount) AS total_transaction_amount
FROM
    etl_atm_data.fact_atm_trans t,
    etl_atm_data.dim_atm a,
    etl_atm_data.dim_location l
WHERE
    t.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;
```

<Screenshot of the query results>



The screenshot displays the AWS Redshift console interface. On the left, a sidebar lists various database schemas and tables, including 'dim_atm_pkey', 'dim_card_type_pkey', 'dim_date_pkey', 'dim_location_pkey', 'fact_atm_trans_pkey', 'dim_atm', 'dim_card_type', 'dim_date', 'dim_location', and 'fact_atm_trans'. The main panel shows the 'Query results' tab for 'Query 811344'. The query is a SELECT statement that filters for 'location' and orders the results by 'total_transaction_amount' in descending order. The query is completed, and the results are displayed in a table with 10 rows. The table has four columns: 'atm_number', 'atm_manufacturer', 'location', and 'total_transaction_amount'. The data shows various ATM locations and their corresponding transaction amounts.

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÅfÅ_resunday	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	NÅfÅstved	213721117
48	Diebold Nixdorf	BrÅfÅ_nderlev	212883099

6. Number of failed ATM transactions across various card types

<Query>

```
SELECT
    t.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_atm_data.fact_atm_trans f,
    etl_atm_data.dim_card_type t
WHERE
    f.card_type_id = t.card_type_id
GROUP BY
    t.card_type
ORDER BY
    inactive_count_percent DESC
LIMIT 10;
```

<Screenshot of the query results>

▶ dim_atm_pkey

...

▶ dim_card_type_pkey

...

▶ dim_date_pkey

...

▶ dim_location_pkey

...

▶ fact_atm_trans_pkey

...

▶ dim_atm

...

▶ dim_card_type

...

▶ dim_date

...

▶ dim_location

...

▶ fact_atm_trans

...

11

END AS inactive_count_percent

12

FROM

13

etl_atm_data.fact_atm_trans f,

14

etl_atm_data.dim_card_type t

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query 811379

Execution

Data

Visualize

Completed, started on November 07, 2023 at 18:00:02

ELAPSED TIME: 00 m 03 s

Rows returned (10)

Export

Search rows

< 1 > ⌕

card_type	total_transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.7600
VISA	170828	30713	17.9700
Dankort - on-us	143813	24680	17.1600
CIRRUS	17362	2953	17.0000
HÃ¶f&vekort - on-us	62487	10331	16.5300
Dankort	28581	4557	15.9400
MasterCard	400507	63482	15.8500
Visa Dankort - on-us	748805	112972	15.0800
HÃ¶f&vekort	8459	1208	14.2800
Visa Dankort	427840	60547	14.1500

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

<Query>

```
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_atm_data.fact_atm_trans t,
    etl_atm_data.dim_atm a,
    etl_atm_data.dim_location l,
    etl_atm_data.dim_date d
WHERE
    t.atm_id = a.atm_id AND
    a.atm_location_id = l.location_id AND
    t.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;
```

dim_atm_pkey

...

dim_card_type_pkey

...

dim_date_pkey

...

dim_location_pkey

...

fact_atm_trans_pkey

...

dim_atm

...

dim_card_type

...

dim_date

...

dim_location

...

fact_atm_trans

...

```

11      etl_atm_data.dim_location l,
12      etl_atm_data.dim_date d
13 WHERE t.atm_id = a.atm_id
14 
```

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query **811403**

Execution

Data

Visualize

Completed, started on November 07, 2023 at 18:01:58

ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export

Search rows

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÅfÅstved	0	32711
1	NCR	NÅfÅstved	1	10076
10	NCR	NÅfÅ, resundby	0	41667
10	NCR	NÅfÅ, resundby	1	12127
100	NCR	Intern Skive	0	17812
100	NCR	Intern Skive	1	1
101	NCR	Bryggen Vejle	0	11693
101	NCR	Bryggen Vejle	1	3247
102	NCR	Aalborg Storcenter Afd	0	14556
102	NCR	Aalborg Storcenter Afd	1	3741

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

<Query>

```

SELECT
    a.atm_number,
    a.atm_manufacturer,
    l.location,
    d.weekday,
    Count(trans_id) AS total_transaction_count
FROM
    etl_atm_data.fact_atm_trans t
INNER JOIN
    etl_atm_data.dim_atm a ON t.atm_id = a.atm_id
INNER JOIN
    etl_atm_data.dim_location l ON a.atm_location_id = l.location_id
INNER JOIN
    etl_atm_data.dim_date d ON t.date_id = d.date_id
WHERE
    l.location = 'Vejgaard' AND
    d.weekday

    IN (SELECT
        d.weekday
        FROM
            etl_atm_data.fact_atm_trans t
        INNER JOIN
            etl_atm_data.dim_date d ON t.date_id = d.date_id
        INNER JOIN
            etl_atm_data.dim_location l ON t.weather_loc_id = l.location_id
        WHERE
            l.location = 'Vejgaard'
        GROUP BY
            d.weekday
        ORDER BY
            Count(t.trans_id) DESC
        LIMIT 1)
GROUP BY
    a.atm_number,
    a.atm_manufacturer,
    l.location,
    d.weekday
ORDER BY
    total_transaction_count;

```

<Screenshot of the query results>

Query 1 x Query 2 x Query 3 x +

Select database info
To view schemas, select a database.
redshiftdb

Select schema info
To view tables, select a schema.
etl_atm_data

Filter tables

dim_atm_pkey
dim_card_type_pkey
dim_date_pkey
dim_location_pkey
fact_atm_trans_pkey
dim_atm
dim_card_type
dim_date
dim_location
fact_atm_trans

```

1 SELECT a.atm_number,
2       a.atm_manufacturer,
3       l.location,
4       d.weekday,
5       Count(trans_id) AS total_transaction_count
6 FROM   etl_atm_data.fact_atm_trans t
7       INNER JOIN etl_atm_data.dim_atm a
8               ON t.atm_id = a.atm_id
9       INNER JOIN etl_atm_data.dim_location l
10              ON a.atm_location_id = l.location_id
11       INNER JOIN etl_atm_data.dim_date d
12               ON t.date_id = d.date_id
13 WHERE  l.location = 'Vejsgaard'

```

Run Save Schedule Clear

Send feedback

Query results Table details

Query 811457

Completed, started on November 07, 2023 at 18:06:39
ELAPSED TIME: 00 m 14 s

Rows returned (2)

Search rows

atm_number	atm_manufacturer	location	weekday	total_transaction_count
103	Diebold Nixdorf	Vejsgaard	Friday	4757
2	NCR	Vejsgaard	Friday	6290