

Solving analytical queries on Redshift Cluster

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

```
SELECT b.atm_number,  
       b.atm_manufacturer,  
       c.location,  
       Sum(CASE  
           WHEN atm_status = 'Inactive' THEN 1  
           ELSE 0  
       end) AS inactive_count  
FROM atm_dwh.fact_atm_trans a  
     LEFT JOIN atm_dwh.dim_atm b  
       ON a.atm_id = b.atm_id  
     LEFT JOIN atm_dwh.dim_location c  
       ON b.atm_location_id = c.location_id  
GROUP BY b.atm_number, b.atm_manufacturer, c.location  
ORDER BY inactive_count DESC  
LIMIT 10;
```

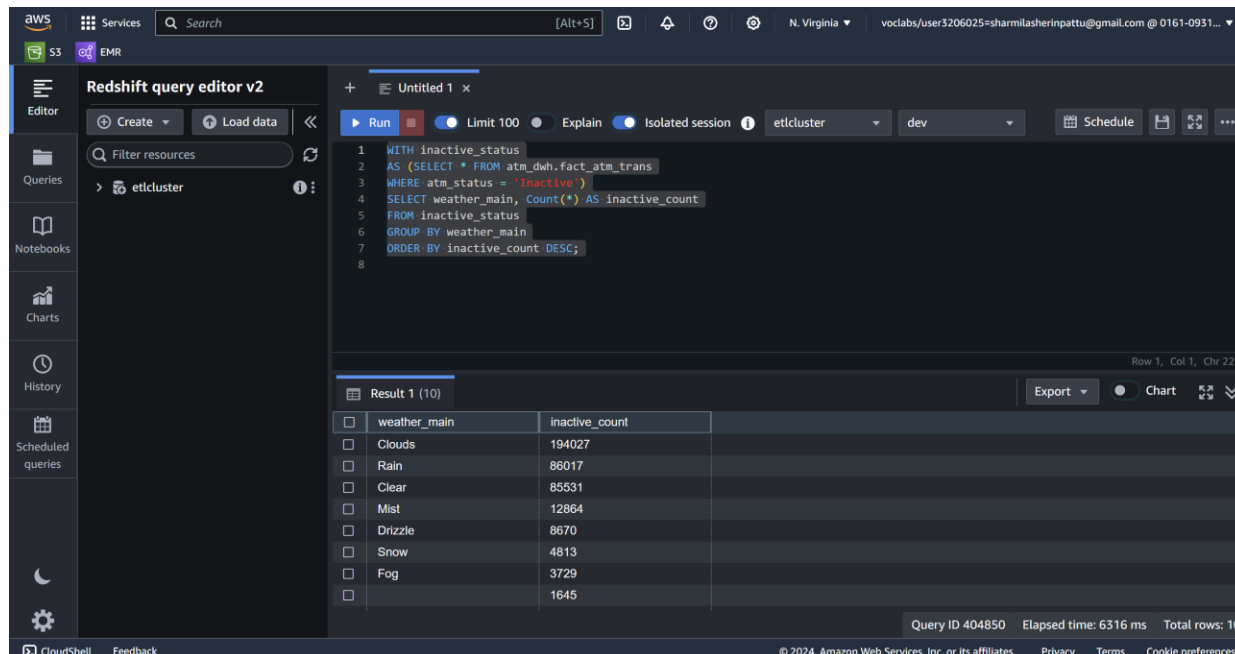
The screenshot displays the AWS Data Studio interface. At the top, the navigation bar includes the AWS logo, 'Services', a search bar with the text 'Search', a keyboard shortcut '[Alt+S]', and several icons for navigation and settings. The top right corner shows the user's profile information: 'N. Virginia' and 'voclabs/user3206025=sharmilasherinpattu@gmail.com @ 0161-0931...'. The left sidebar contains icons for 'Editor', 'Queries', 'Notebooks', 'Charts', and 'History'. The main workspace is titled 'Untitled 1' and shows a table with the caption 'Result 1 (10)'. The table has four columns: 'atm_number', 'atm_manufacturer', 'location', and 'transaction_count'. The data is as follows:

atm_number	atm_manufacturer	location	transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÅfÅ_resundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÅfÅstved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÅfÅnderslev	42493

FROM inactive status

GROUP BY weather_main

ORDER BY inactive_count DESC;



The screenshot shows the AWS Redshift query editor v2 interface. The SQL query being executed is:

```
1 WITH inactive_status
2 AS (SELECT * FROM atm_dwh.fact_atm_trans
3 WHERE atm_status = 'inactive')
4 SELECT weather_main, Count(*) AS inactive_count
5 FROM inactive_status
6 GROUP BY weather_main
7 ORDER BY inactive_count DESC;
```

The results are displayed in a table with 10 rows. The columns are weather_main and inactive_count.

weather_main	inactive_count
Clouds	194027
Rain	86017
Clear	85531
Mist	12864
Drizzle	8670
Snow	4813
Fog	3729
	1645

Query ID: 404850, Elapsed time: 6316 ms, Total rows: 10

3. Top 10 ATMs with the most number of transactions

SELECT b.atm_number,

b.atm_manufacturer,

c.location, Count(trans_id) AS transaction_count

FROM atm_dwh.fact_atm_trans a

LEFT JOIN atm_dwh.dim_atm b

ON a.atm_id = b.atm_id

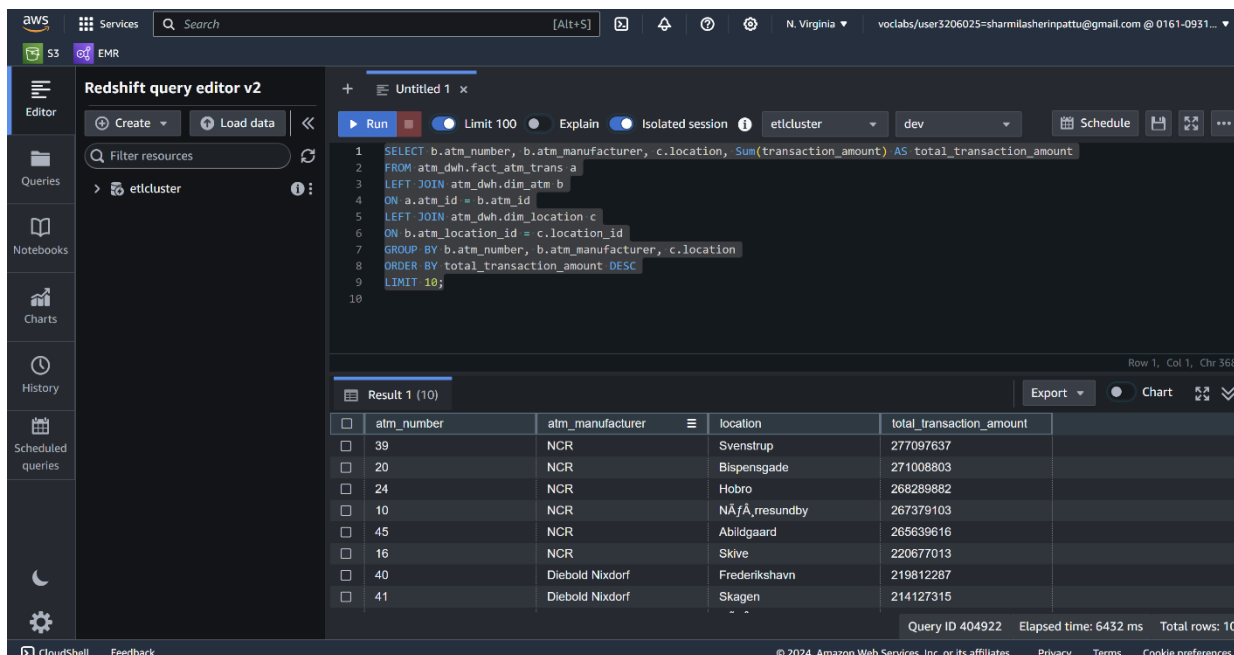
LEFT JOIN atm_dwh.dim_location c

ON b.atm_location_id = c.location_id

GROUP BY b.atm_number, b.atm_manufacturer, c.location

ORDER BY transaction_count DESC

LIMIT 10;



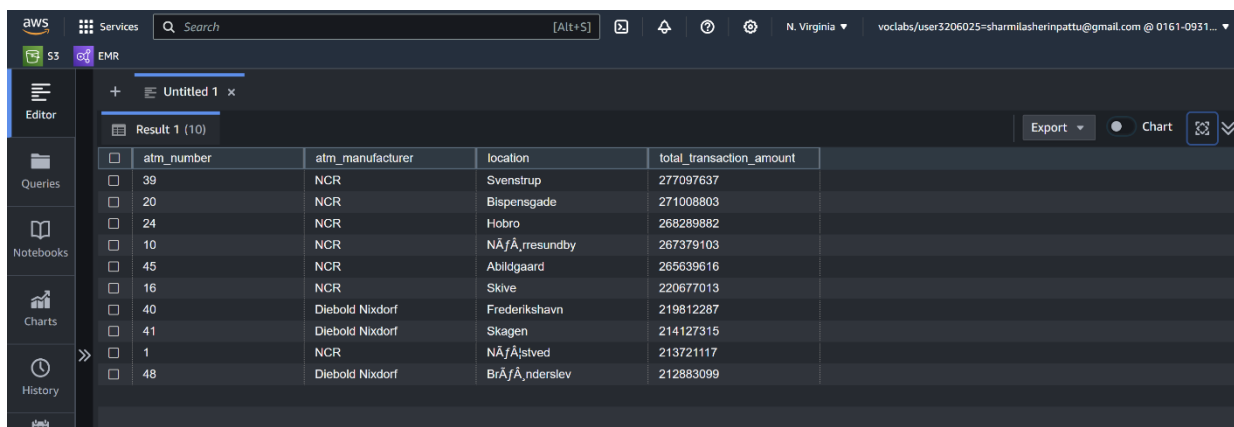
The screenshot shows the AWS Redshift query editor v2 interface. The query being executed is:

```
1 SELECT b.atm_number, b.atm_manufacturer, c.location, Sum(transaction_amount) AS total_transaction_amount
2 FROM atm_dwh.fact_atm_trans a
3 LEFT JOIN atm_dwh.dim_atm b
4 ON a.atm_id = b.atm_id
5 LEFT JOIN atm_dwh.dim_location c
6 ON b.atm_location_id = c.location_id
7 GROUP BY b.atm_number, b.atm_manufacturer, c.location
8 ORDER BY total_transaction_amount DESC
9 LIMIT 10;
```

The results are displayed in a table with the following columns: atm_number, atm_manufacturer, location, and total_transaction_amount. The results are sorted by total_transaction_amount in descending order.

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÄfÄ_resundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315

Query ID 404922 Elapsed time: 6432 ms Total rows: 10



The screenshot shows the AWS Redshift query editor v2 interface. The query being executed is:

```
1 SELECT year,
2 month,
3 Count(trans_id) AS total_inactive_count
4 FROM atm_dwh.fact_atm_trans a
```

The results are displayed in a table with the following columns: year, month, and total_inactive_count. The results are sorted by year and month.

year	month	total_inactive_count
2019	1	1
2019	2	1
2019	3	1
2019	4	1
2019	5	1
2019	6	1
2019	7	1
2019	8	1
2019	9	1
2019	10	1
2019	11	1
2019	12	1
2020	1	1
2020	2	1
2020	3	1
2020	4	1
2020	5	1
2020	6	1
2020	7	1
2020	8	1
2020	9	1
2020	10	1
2020	11	1
2020	12	1
2021	1	1
2021	2	1
2021	3	1
2021	4	1
2021	5	1
2021	6	1
2021	7	1
2021	8	1
2021	9	1
2021	10	1
2021	11	1
2021	12	1
2022	1	1
2022	2	1
2022	3	1
2022	4	1
2022	5	1
2022	6	1
2022	7	1
2022	8	1
2022	9	1
2022	10	1
2022	11	1
2022	12	1

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

SELECT year,

month,

Count(trans_id) AS total_inactive_count

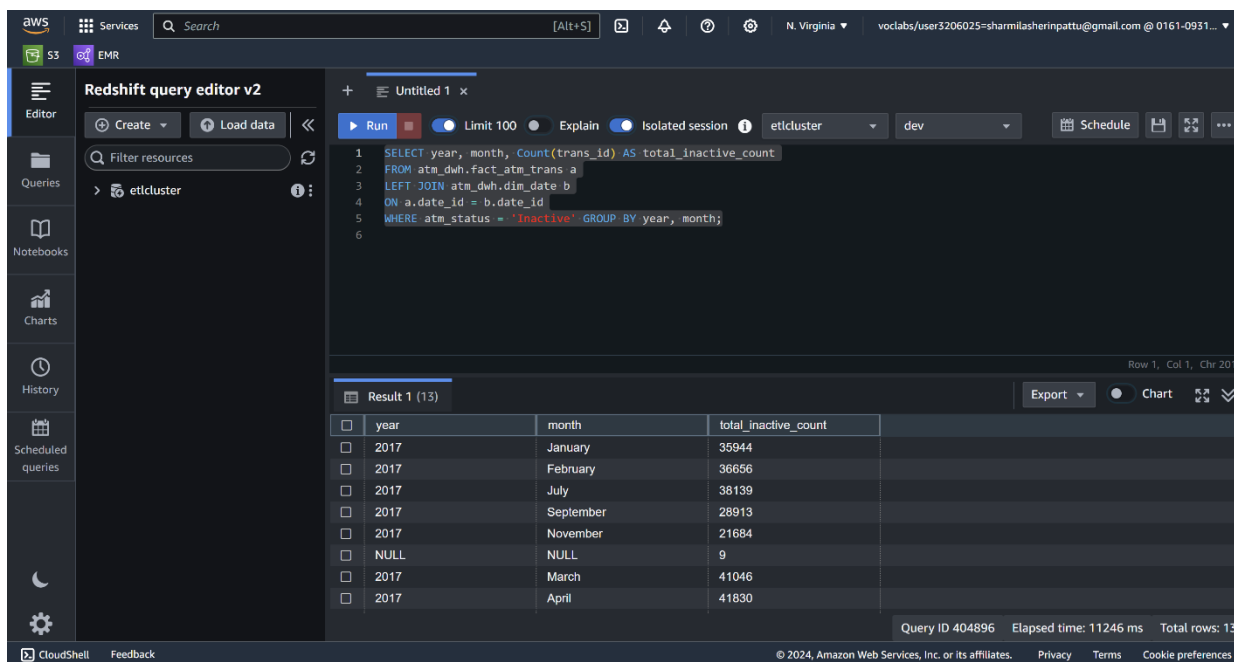
FROM atm_dwh.fact_atm_trans a

LEFT JOIN atm_dwh.dim_date b

ON a.date_id = b.date_id

WHERE atm_status = 'Inactive'

GROUP BY year, month;



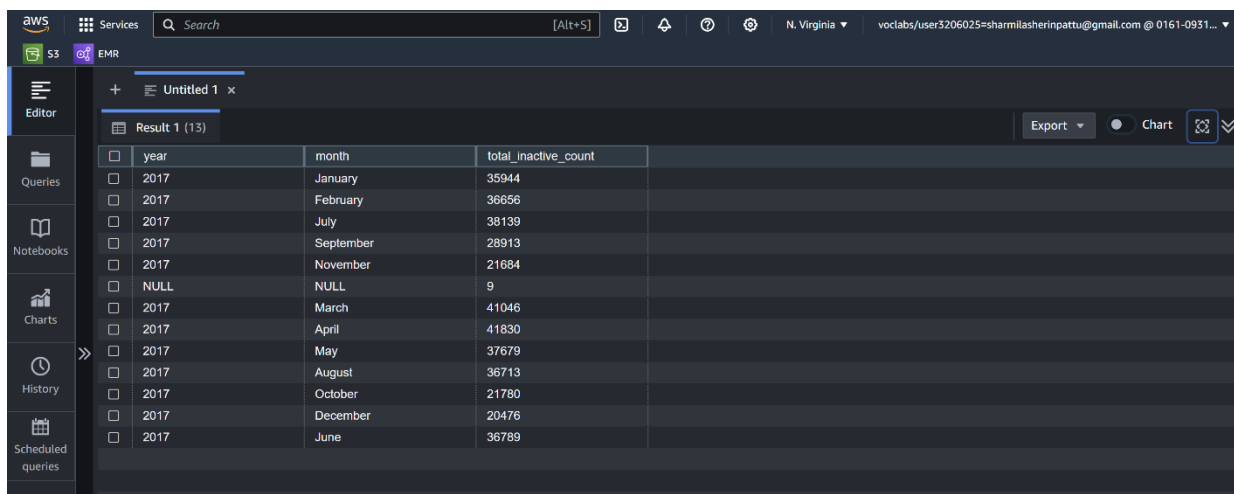
The screenshot shows the AWS Redshift query editor v2 interface. The SQL query is as follows:

```
1 SELECT year, month, Count(trans_id) AS total_inactive_count
2 FROM atm_dwh.fact_atm_trans a
3 LEFT JOIN atm_dwh.dim_date b
4 ON a.date_id = b.date_id
5 WHERE atm_status = 'Inactive' GROUP BY year, month;
6
```

The results are displayed in a table with the following data:

year	month	total_inactive_count
2017	January	35944
2017	February	36656
2017	July	38139
2017	September	28913
2017	November	21684
NULL	NULL	9
2017	March	41046
2017	April	41830

Query ID: 404896, Elapsed time: 11246 ms, Total rows: 13.

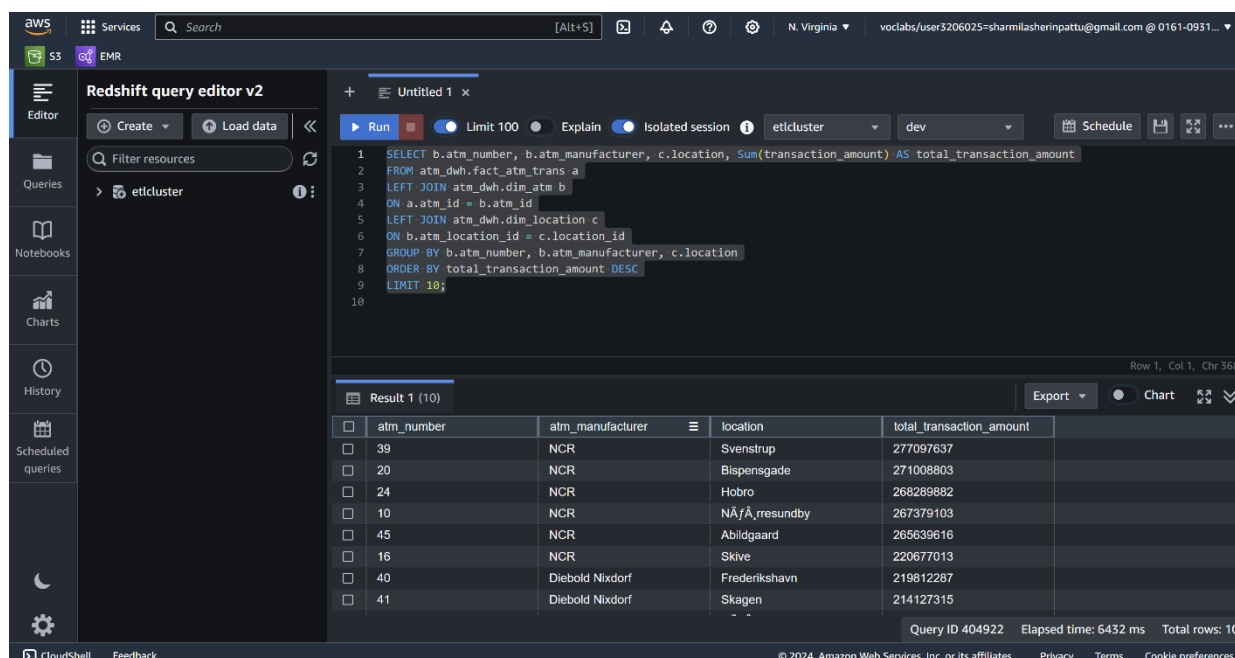


The screenshot shows the AWS Redshift query editor v2 interface with the full results of the SQL query. The results are displayed in a table with the following data:

year	month	total_inactive_count
2017	January	35944
2017	February	36656
2017	July	38139
2017	September	28913
2017	November	21684
NULL	NULL	9
2017	March	41046
2017	April	41830
2017	May	37679
2017	August	36713
2017	October	21780
2017	December	20476
2017	June	36789

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

```
SELECT b.atm_number,
       b.atm_manufacturer,
       c.location,
       Sum(transaction_amount) AS total_transaction_amount
FROM atm_dwh.fact_atm_trans a
LEFT JOIN atm_dwh.dim_atm b
      ON a.atm_id = b.atm_id
LEFT JOIN atm_dwh.dim_location c
      ON b.atm_location_id = c.location_id
GROUP BY b.atm_number, b.atm_manufacturer, c.location
ORDER BY total_transaction_amount DESC
LIMIT 10;
```



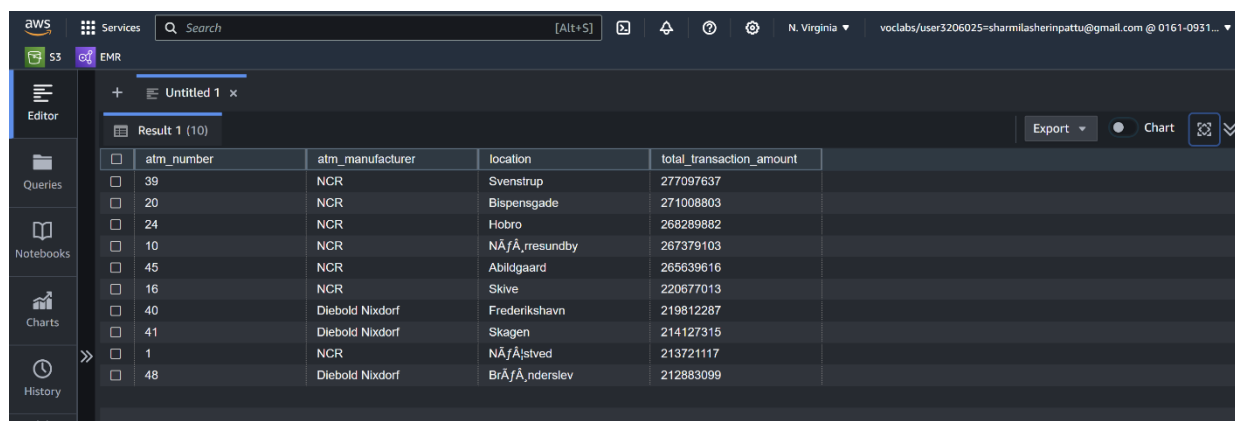
The screenshot shows the AWS Redshift Query Editor v2 interface. The SQL query is as follows:

```
1 SELECT b.atm_number, b.atm_manufacturer, c.location, Sum(transaction_amount) AS total_transaction_amount
2 FROM atm_dwh.fact_atm_trans a
3 LEFT JOIN atm_dwh.dim_atm b
4   ON a.atm_id = b.atm_id
5 LEFT JOIN atm_dwh.dim_location c
6   ON b.atm_location_id = c.location_id
7 GROUP BY b.atm_number, b.atm_manufacturer, c.location
8 ORDER BY total_transaction_amount DESC
9 LIMIT 10;
```

The results are displayed in a table with 5 columns: atm_number, atm_manufacturer, location, and total_transaction_amount. The table shows the top 10 results ordered by total_transaction_amount in descending order.

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÄfÅ_mesundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315

Query ID: 404922, Elapsed time: 6432 ms, Total rows: 10



atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÅfÅ_resundby	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	BrÅfÅnderslev	212883099

6. Number of failed ATM transactions across various card types

SELECT card_type,

Count(trans_id) AS failed_transaction_count

FROM atm_dwh.fact_atm_trans a

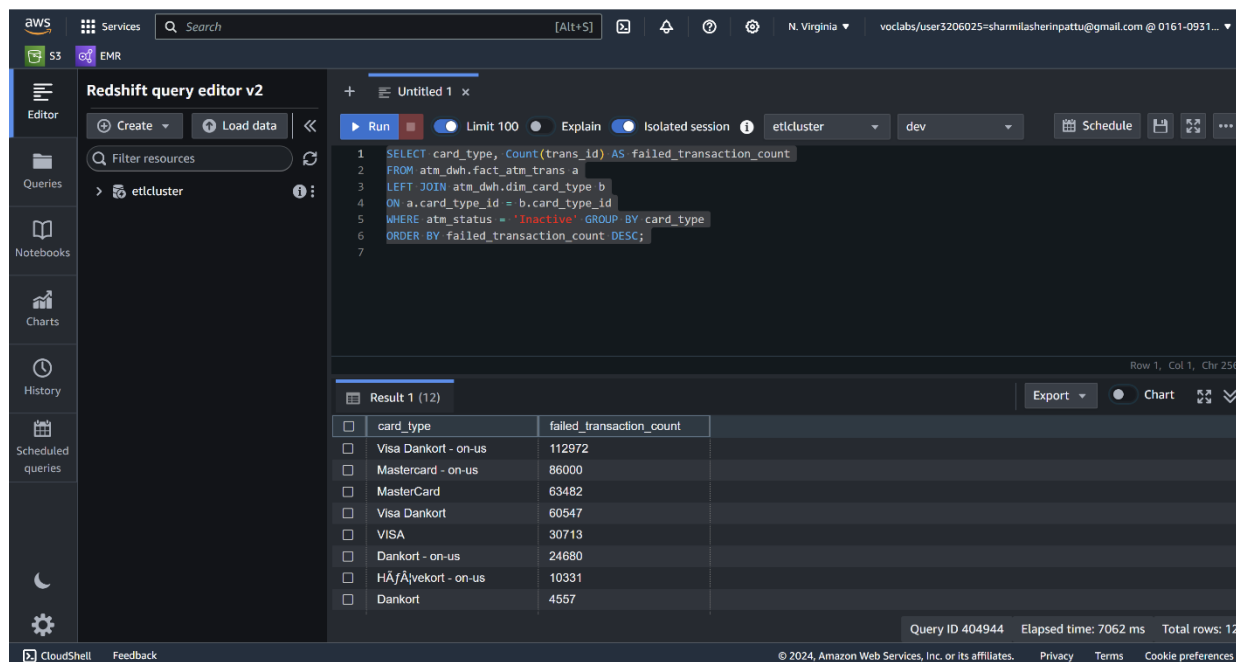
LEFT JOIN atm_dwh.dim_card_type b

ON a.card_type_id = b.card_type_id

WHERE atm_status = 'Inactive'

GROUP BY card_type

ORDER BY failed_transaction_count DES;



Redshift query editor v2

Filter resources: etcluster

```

1 SELECT card_type, Count(trans_id) AS failed_transaction_count
2 FROM atm_dwh.fact_atm_trans a
3 LEFT JOIN atm_dwh.dim_card_type b
4 ON a.card_type_id = b.card_type_id
5 WHERE atm_status = 'Inactive' GROUP BY card_type
6 ORDER BY failed_transaction_count DESC;
7

```

Result 1 (12)

card_type	failed_transaction_count
Visa Dankort - on-us	112972
Mastercard - on-us	86000
MasterCard	63482
Visa Dankort	60547
VISA	30713
Dankort - on-us	24680
HjÄrÅvekort - on-us	10331
Dankort	4557

Query ID 404944 Elapsed time: 7062 ms Total rows: 12

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

WITH weekday_weekend AS

(

SELECT date_id,

CASE

WHEN weekday IN ('Saturday',

'Sunday') THEN 1

ELSE 0

END AS weekday_weekend_flag

FROM atm_dwh.dim_date)

SELECT atm.atm_number,

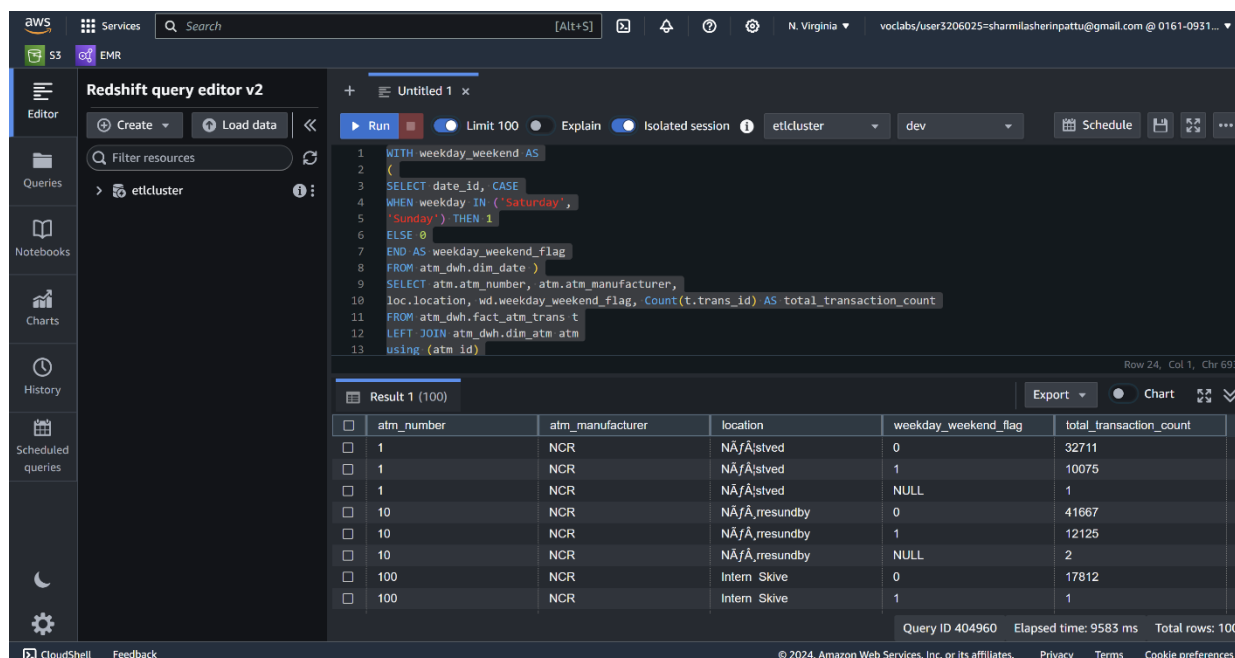
atm.atm_manufacturer,

loc.location,


```

wd.weekday_weekend_flag,
Count(t.trans_id) AS total_transaction_count
FROM atm_dwh.fact_atm_trans t
LEFT JOIN atm_dwh.dim_atm atm
    using (atm_id)
LEFT JOIN weekday_weekend wd
    using (date_id)
LEFT JOIN atm_dwh.dim_location loc
    ON loc.location_id= t.weather_loc_id
GROUP BY atm.atm_number, atm.atm_manufacturer, loc.location, wd.weekday_weekend_flag,
ORDER BY atm.atm_number, atm.atm_manufacturer, loc.location, wd.weekday_weekend_flag,
total_transaction_count;

```



The screenshot shows the AWS Redshift query editor v2 interface. The query editor displays a SQL query that calculates the total transaction count for each ATM location, categorized by weekday/weekend flag. The query is as follows:

```

1 WITH weekday_weekend AS
2 (
3 SELECT date_id, CASE
4 WHEN weekday IN ('Saturday',
5 'Sunday') THEN 1
6 ELSE 0
7 END AS weekday_weekend_flag
8 FROM atm_dwh.dim_date )
9 SELECT atm.atm_number, atm.atm_manufacturer,
10 loc.location, wd.weekday_weekend_flag, Count(t.trans_id) AS total_transaction_count
11 FROM atm_dwh.fact_atm_trans t
12 LEFT JOIN atm_dwh.dim_atm atm
13 using (atm_id)

```

The results are displayed in a table with the following columns: atm_number, atm_manufacturer, location, weekday_weekend_flag, and total_transaction_count. The table shows 100 rows of data.

atm_number	atm_manufacturer	location	weekday_weekend_flag	total_transaction_count
1	NCR	NÄfÄstved	0	32711
1	NCR	NÄfÄstved	1	10075
1	NCR	NÄfÄstved	NULL	1
10	NCR	NÄfÄ_resundby	0	41667
10	NCR	NÄfÄ_resundby	1	12125
10	NCR	NÄfÄ_resundby	NULL	2
100	NCR	Intern Skive	0	17812
100	NCR	Intern Skive	1	1

Query ID: 404960, Elapsed time: 9583 ms, Total rows: 100

aws Services Search [Alt+S] N. Virginia voclabs/user3206025-sharmilasherinpattu@gmail.com @ 0161-0931...

S3 EMR

Editor + Untitled 1 x

Result 1 (100)

Export Chart

	atm_number	atm_manufacturer	location	weekday_weekend_flag	total_transaction_count
<input type="checkbox"/>	1	NCR	NÅfÅstved	0	32711
<input type="checkbox"/>	1	NCR	NÅfÅstved	1	10075
<input type="checkbox"/>	1	NCR	NÅfÅstved	NULL	1
<input type="checkbox"/>	10	NCR	NÅfÅ, resundby	0	41667
<input type="checkbox"/>	10	NCR	NÅfÅ, resundby	1	12125
<input type="checkbox"/>	10	NCR	NÅfÅ, resundby	NULL	2
<input type="checkbox"/>	100	NCR	Intern Skive	0	17812
<input type="checkbox"/>	100	NCR	Intern Skive	1	1
<input type="checkbox"/>	101	NCR	Bryggen Vejle	0	11693
<input type="checkbox"/>	101	NCR	Bryggen Vejle	1	3247
<input type="checkbox"/>	102	NCR	Aalborg Storcenter Afd	0	14556
<input type="checkbox"/>	102	NCR	Aalborg Storcenter Afd	1	3741
<input type="checkbox"/>	103	Diebold Nixdorf	Vejgaard	0	18570
<input type="checkbox"/>	103	Diebold Nixdorf	Vejgaard	1	2607
<input type="checkbox"/>	104	NCR	Intern ÅfEøsterÅfÅ	0	21590
<input type="checkbox"/>	104	NCR	Intern ÅfEøsterÅfÅ	1	2
<input type="checkbox"/>	105	Diebold Nixdorf	NÅfÅ, resundby	0	8027
<input type="checkbox"/>	106	NCR	Intern Roskilde	0	3092
<input type="checkbox"/>	106	NCR	Intern Roskilde	1	1
<input type="checkbox"/>	107	Diebold Nixdorf	Kolding	0	3833
<input type="checkbox"/>	107	Diebold Nixdorf	Kolding	1	1

Query ID 404960 Elapsed time: 9583 ms Total rows: 100

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

WITH rank AS

(

SELECT DISTINCT atm_id,

atm_number, location,

weekday,

count(trans_id) AS

total_transaction_count,

row_number() OVER (partition BY atm_id ORDER BY count(trans_id)

DESC) AS high

FROM atm_dwh.fact_atm_trans f

LEFT JOIN atm_dwh.dim_atm

using (atm_id)

LEFT JOIN atm_dwh.dim_location al

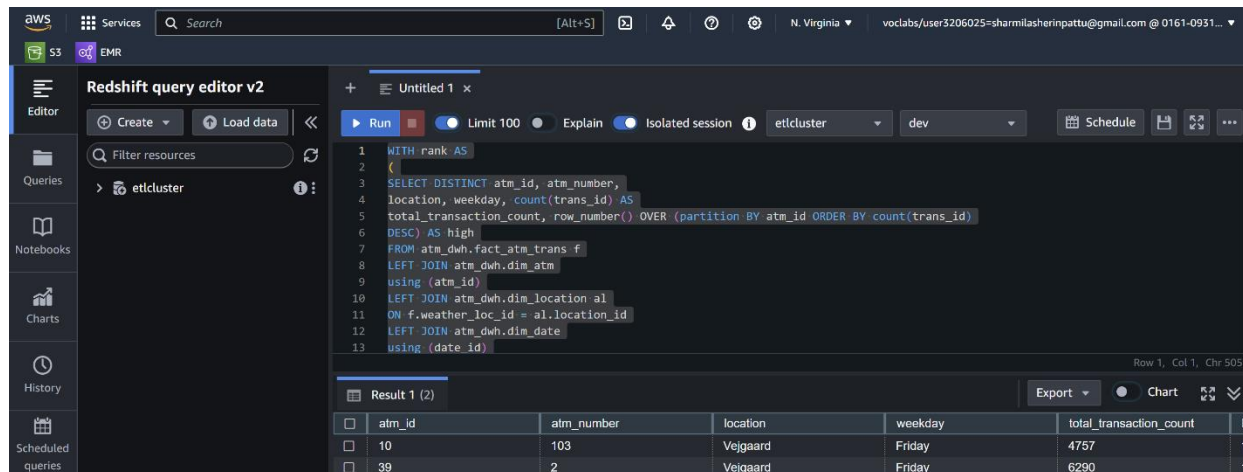
ON f.weather_loc_id = al.location_id

LEFT JOIN atm_dwh.dim_date

using (date_id) WHERE location = 'Veigaard'

GROUP BY weekday, atm_id, atm_number, location)

SELECT * FROM rank WHERE high=1;



The screenshot shows the AWS Redshift query editor v2 interface. The query editor is titled "Untitled 1" and contains the following SQL query:

```
1 WITH rank AS
2 {
3 SELECT DISTINCT atm_id, atm_number,
4 location, weekday, count(trans_id) AS
5 total_transaction_count, row_number() OVER (partition BY atm_id ORDER BY count(trans_id)
6 DESC) AS high
7 FROM atm_dwh.fact_atm_trans f
8 LEFT JOIN atm_dwh.dim_atm
9 using (atm_id)
10 LEFT JOIN atm_dwh.dim_location al
11 ON f.weather_loc_id = al.location_id
12 LEFT JOIN atm_dwh.dim_date
13 using (date_id)
```

The query is executed, and the results are displayed in a table with the following columns: atm_id, atm_number, location, weekday, total_transaction_count, and high. The results are as follows:

atm_id	atm_number	location	weekday	total_transaction_count	high
10	103	Veigaard	Friday	4757	1
39	2	Veigaard	Friday	6290	1