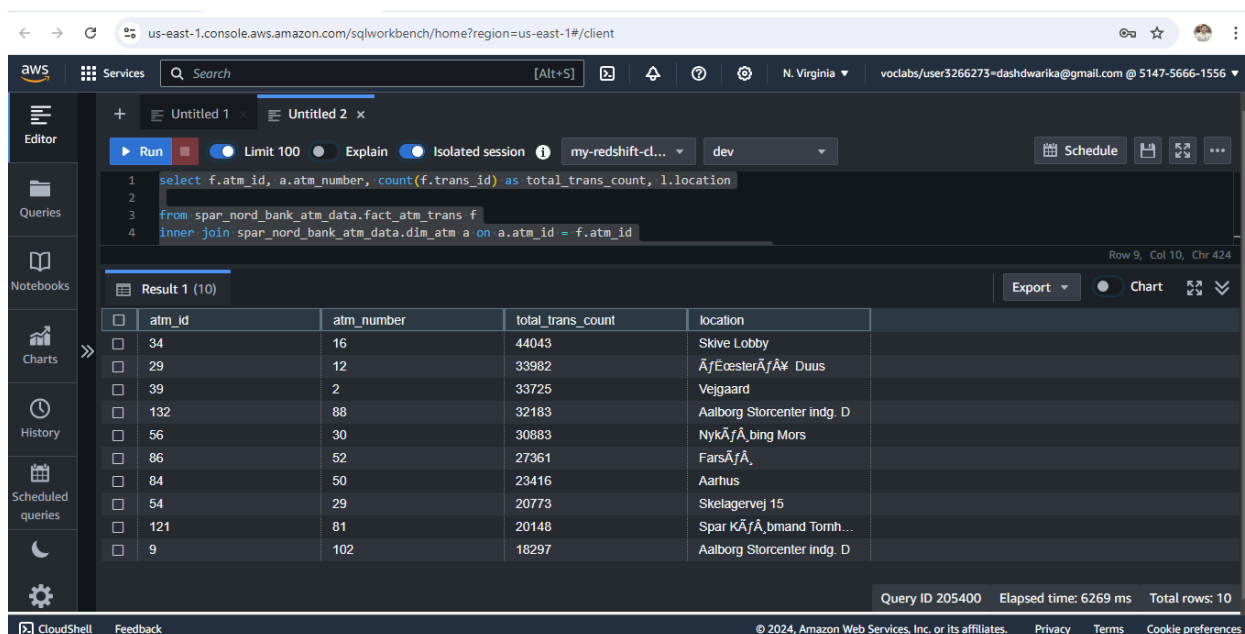


Solving analytical queries on Redshift Cluster

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 f.atm_id, a.atm_number, count(f.trans_id) as total_trans_count, l.location
from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_atm a on a.atm_id = f.atm_id
inner join spar_nord_bank_atm_data.dim_location l on a.atm_location_id =
l.location_id
group by f.atm_id, a.atm_number, l.location, f.atm_status
having f.atm_status = 'Inactive'
order by total_trans_count DESC
Limit 10;
```



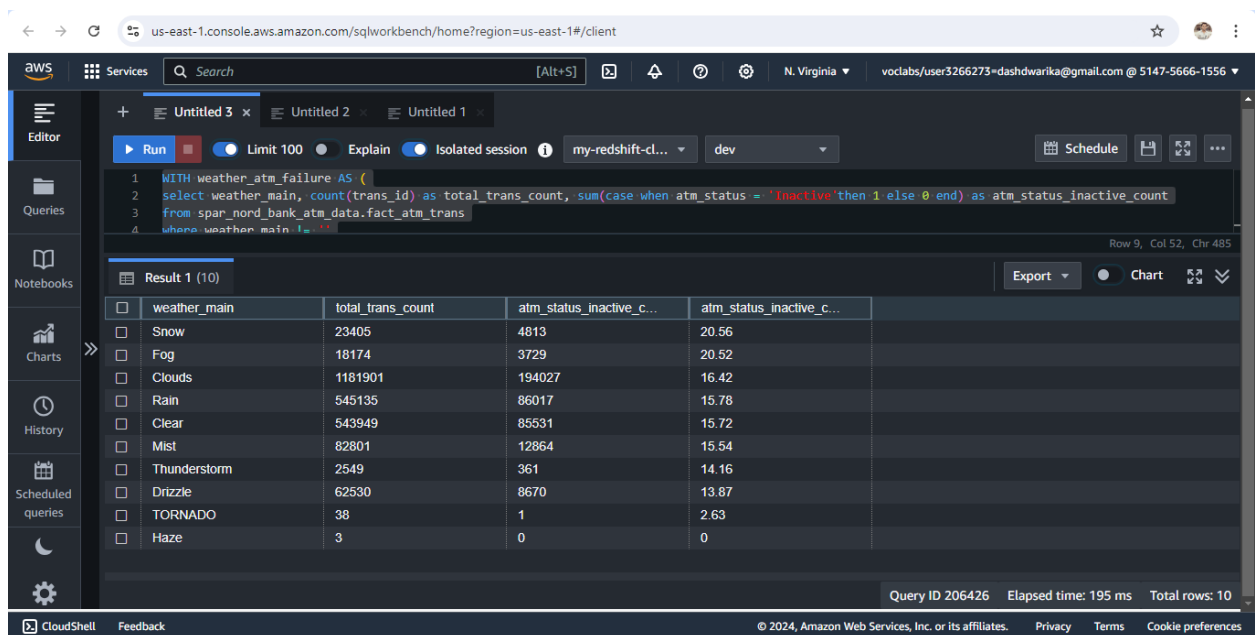
The screenshot shows the AWS Redshift Query Editor interface. The query is executed, and the results are displayed in a table. The table has 5 columns: atm_id, atm_number, total_trans_count, and location. The results are sorted by total_trans_count in descending order, showing the top 10 ATMs with the most transactions in the 'inactive' state.

atm_id	atm_number	total_trans_count	location
34	16	44043	Skive Lobby
29	12	33982	Århus
39	2	33725	Vejgaard
132	88	32183	Aalborg Storcenter indg. D
56	30	30883	Nykøbing Mors
86	52	27361	Farsø
84	50	23416	Aarhus
54	29	20773	Skelagervej 15
121	81	20148	Spar København Tønder
9	102	18297	Aalborg Storcenter indg. D

Query ID 205400 Elapsed time: 6269 ms Total rows: 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 total_trans_count, sum(case when
atm_status = 'Inactive' then 1 else 0 end) as atm_status_inactive_count
from spar_nord_bank_atm_data.fact_atm_trans
where weather_main != ''
group by weather_main
)
select *, round(((CAST(atm_status_inactive_count as numeric) / total_trans_count)
* 100), 2) AS atm_status_inactive_count_percentage
from weather_atm_failure
order by atm_status_inactive_count_percentage DESC;
```



The screenshot shows the AWS SQL Workbench interface with the query execution results displayed in a table. The table has 5 columns: weather_main, total_trans_count, atm_status_inactive_c..., atm_status_inactive_c..., and atm_status_inactive_c... (truncated). The results are sorted by atm_status_inactive_count_percentage in descending order.

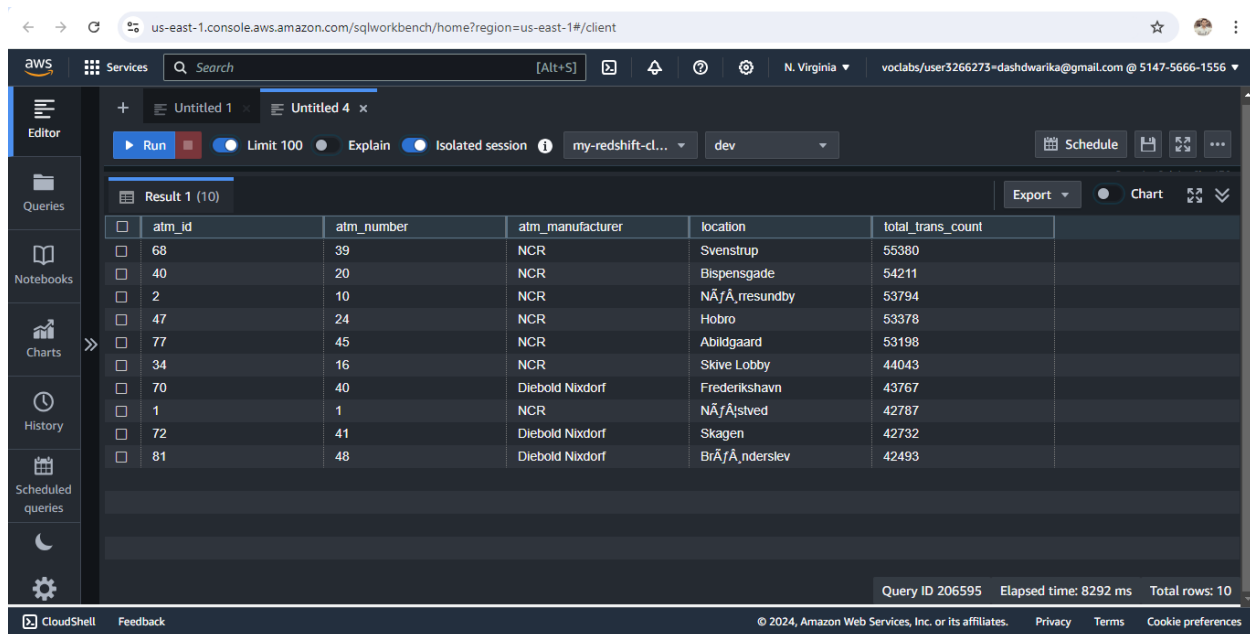
weather_main	total_trans_count	atm_status_inactive_c...	atm_status_inactive_c...	atm_status_inactive_c...
Snow	23405	4813	20.56	
Fog	18174	3729	20.52	
Clouds	1181901	194027	16.42	
Rain	545135	86017	15.78	
Clear	543949	85531	15.72	
Mist	82801	12864	15.54	
Thunderstorm	2549	361	14.16	
Drizzle	62530	8670	13.87	
TORNADO	38	1	2.63	
Haze	3	0	0	

Query ID 206426 Elapsed time: 195 ms Total rows: 10

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

```
select f.atm_id, a.atm_number, a.atm_manufacturer, l.location, count(f.trans_id)
as total_trans_count

from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_atm a on a.atm_id = f.atm_id
inner join spar_nord_bank_atm_data.dim_location l on a.atm_location_id =
l.location_id
group by f.atm_id, a.atm_number, a.atm_manufacturer, l.location, f.atm_status
order by total_trans_count DESC
Limit 10;
```



The screenshot shows the AWS SQL Workbench interface. The query results are displayed in a table with the following columns: atm_id, atm_number, atm_manufacturer, location, and total_trans_count. The results are ordered by total_trans_count in descending order.

atm_id	atm_number	atm_manufacturer	location	total_trans_count
68	39	NCR	Svenstrup	55380
40	20	NCR	Bispensgade	54211
2	10	NCR	NÃrÃresundby	53794
47	24	NCR	Hobro	53378
77	45	NCR	Abildgaard	53198
34	16	NCR	Skive Lobby	44043
70	40	Diebold Nixdorf	Frederikshavn	43767
1	1	NCR	NÃrÃstved	42787
72	41	Diebold Nixdorf	Skagen	42732
81	48	Diebold Nixdorf	BrÃfÃnderslev	42493

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

```
WITH inactive_count_month_wise AS (
select d.month, count(f.trans_id) as total_trans_count, sum(case when
f.atm_status = 'Inactive' then 1 else 0 end) as atm_status_inactive_count
from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_date d on d.date_id = f.date_id
group by d.month
)
select *, round(cast(atm_status_inactive_count as NUMERIC) / total_trans_count *
100, 2) as atm_status_inactive_count_percent
from inactive_count_month_wise
order by to_date(month, 'Month');
```

← → ↺

us-east-1.console.aws.amazon.com/sqlworkbench/home?region=us-east-1#/client

☆

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3266273=dashdwarika@gmail.com @ 5147-5666-1556

+

Untitled 1

Untitled 4

×

Run

Limit 100

Explain

Isolated session

my-redshift-cl...

dev

Schedule

1

2

3

ATTN: inactive count month wise AS OF

Result 1 (12)

Export

Chart

<input type="checkbox"/>	month	total_trans_count	atm_status_inactive_c...	atm_status_inactive_c...	
<input type="checkbox"/>	January	180195	35953	19.95	
<input type="checkbox"/>	February	182659	36656	20.07	
<input type="checkbox"/>	March	209586	41046	19.58	
<input type="checkbox"/>	April	218865	41830	19.11	
<input type="checkbox"/>	May	222418	37679	16.94	
<input type="checkbox"/>	June	225166	36789	16.34	
<input type="checkbox"/>	July	227682	38139	16.75	
<input type="checkbox"/>	August	217218	36713	16.9	
<input type="checkbox"/>	September	202101	28913	14.31	
<input type="checkbox"/>	October	191667	21780	11.36	
<input type="checkbox"/>	November	193967	21684	11.18	
<input type="checkbox"/>	December	197048	20476	10.39	

Query ID 207390

Elapsed time: 370 ms

Total rows: 12

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

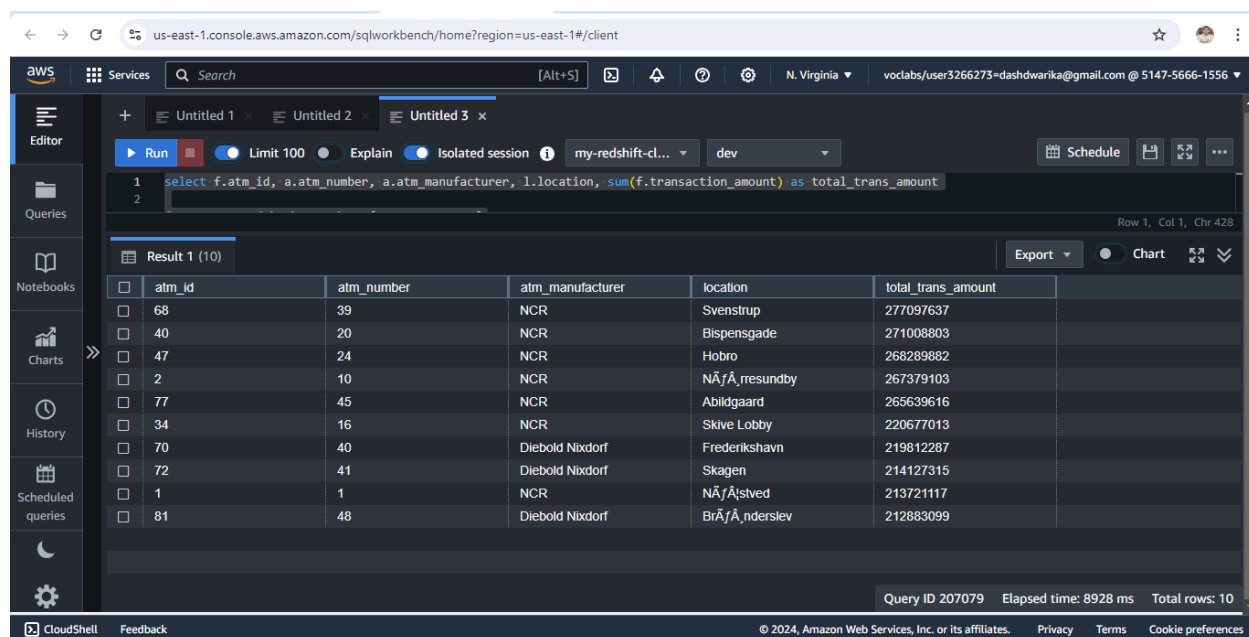
Terms

Cookie preferences

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

```
select f.atm_id, a.atm_number, a.atm_manufacturer, l.location,
sum(f.transaction_amount) as total_trans_amount

from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_atm a on a.atm_id = f.atm_id
inner join spar_nord_bank_atm_data.dim_location l on a.atm_location_id =
l.location_id
group by f.atm_id, a.atm_number, a.atm_manufacturer, l.location
order by total_trans_amount DESC
Limit 10;
```



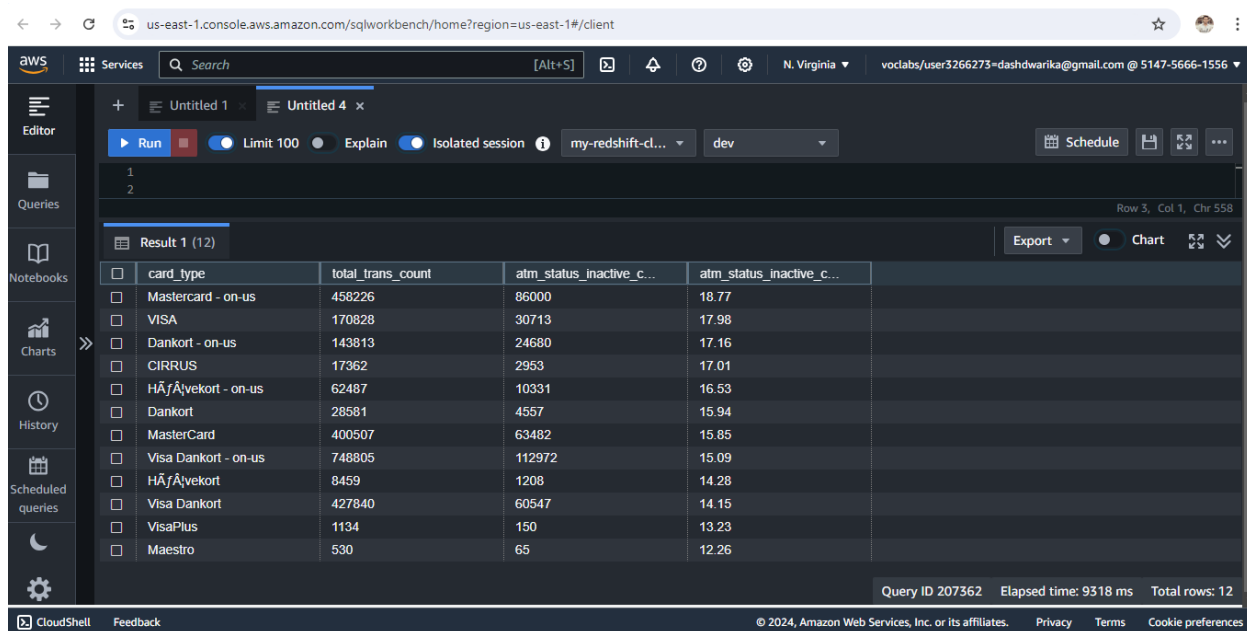
The screenshot shows the AWS SQL Workbench interface with the query results displayed in a table. The table has 6 columns: atm_id, atm_number, atm_manufacturer, location, and total_trans_amount. The results are sorted by total_trans_amount in descending order.

atm_id	atm_number	atm_manufacturer	location	total_trans_amount
68	39	NCR	Svenstrup	277097637
40	20	NCR	Blispensgade	271008803
47	24	NCR	Hobro	268289882
2	10	NCR	NÅfÅresundby	267379103
77	45	NCR	Abildgaard	265639616
34	16	NCR	Skive Lobby	220677013
70	40	Diebold Nixdorf	Frederikshavn	219812287
72	41	Diebold Nixdorf	Skagen	214127315
1	1	NCR	NÅfÅstved	213721117
81	48	Diebold Nixdorf	BrÅfÅnderslev	212883099

6. Number of failed ATM transactions across various card types

```
WITH inactive_count_card_wise AS (
select c.card_type, count(f.trans_id) as total_trans_count, sum(case when
f.atm_status = 'Inactive' then 1 else 0 end) as atm_status_inactive_count
from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_card_type c on c.card_type_id =
f.card_type_id
group by c.card_type
)
select *, round(cast(atm_status_inactive_count as NUMERIC) / total_trans_count *
100, 2) as atm_status_inactive_count_percent
from inactive_count_card_wise
```

```
order by atm_status_inactive_count_percent DESC;
```

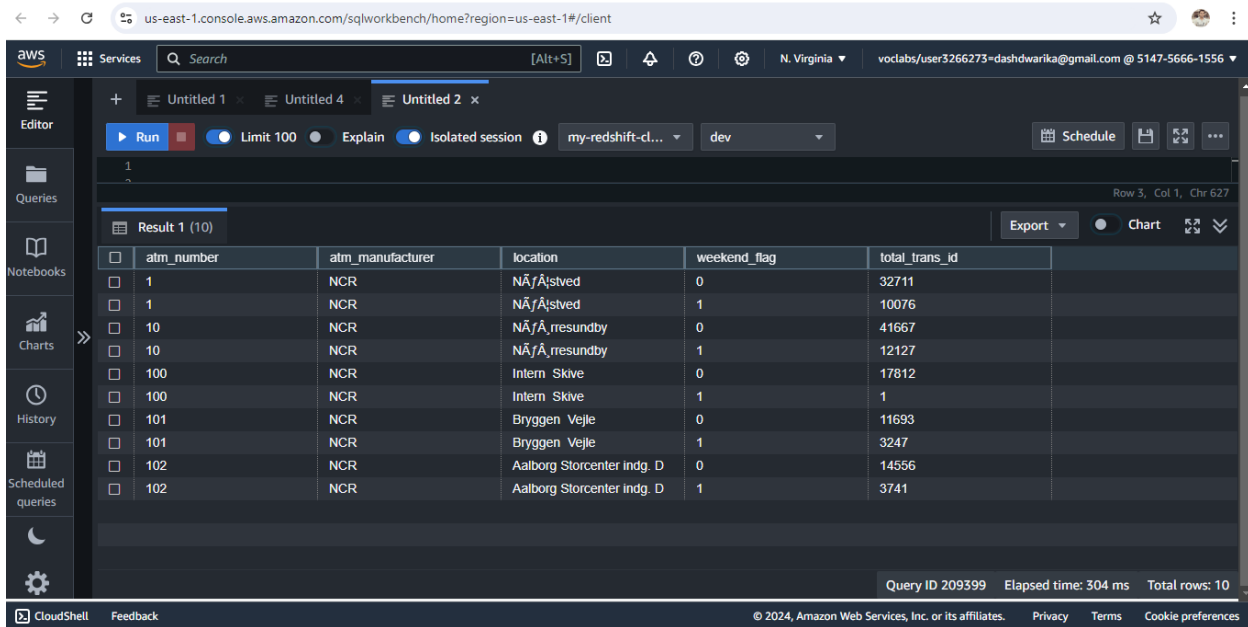


The screenshot shows the AWS SQL Workbench interface. The query result is displayed in a table with the following data:

card_type	total_trans_count	atm_status_inactive_c...	atm_status_inactive_c...
Mastercard - on-us	458226	86000	18.77
VISA	170828	30713	17.98
Dankort - on-us	143813	24680	17.16
CIRRUS	17362	2953	17.01
HÃ¶fvekort - on-us	62487	10331	16.53
Dankort	28581	4557	15.94
MasterCard	400507	63482	15.85
Visa Dankort - on-us	748805	112972	15.09
HÃ¶fvekort	8459	1208	14.28
Visa Dankort	427840	60547	14.15
VisaPlus	1134	150	13.23
Maestro	530	65	12.26

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(f.trans_id) as
total_trans_id
from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_atm a on a.atm_id = f.atm_id
inner join spar_nord_bank_atm_data.dim_location l on l.location_id =
a.atm_location_id
inner join spar_nord_bank_atm_data.dim_date d on d.date_id = f.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_trans_id DESC
LIMIT 10;
```



	atm_number	atm_manufacturer	location	weekend_flag	total_trans_id	total_trans_id
<input type="checkbox"/>	1	NCR	NÅrÅstved	0	32711	
<input type="checkbox"/>	1	NCR	NÅrÅstved	1	10076	
<input type="checkbox"/>	10	NCR	NÅrÅrresundby	0	41667	
<input type="checkbox"/>	10	NCR	NÅrÅrresundby	1	12127	
<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 indg. D	0	14556	
<input type="checkbox"/>	102	NCR	Aalborg Storcenter indg. D	1	3741	

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

```
WITH vejgaard_atm_trans_data AS (
select a.atm_id, a.atm_number, a.atm_manufacturer, l.location, count(f.trans_id)
as total_trans_id, d.weekday
from spar_nord_bank_atm_data.fact_atm_trans f
inner join spar_nord_bank_atm_data.dim_atm a on a.atm_id = f.atm_id
inner join spar_nord_bank_atm_data.dim_location l on l.location_id =
a.atm_location_id
inner join spar_nord_bank_atm_data.dim_date d on d.date_id = f.date_id
where l.location = 'Vejgaard'
group by a.atm_id, a.atm_number, a.atm_manufacturer, l.location, d.weekday
order by total_trans_id DESC
)
select * from vejgaard_atm_trans_data v1
where v1.total_trans_id = (select max(total_trans_id) from
vejgaard_atm_trans_data v2 where v1.atm_id = v2.atm_id);
```

us-east-1.console.aws.amazon.com/sqlworkbench/home?region=us-east-1#/client

Services Search [Alt+S] N. Virginia voclabs/user3266273-dashdwarika@gmail.com @ 5147-5666-1556

Editor

Run Limit 100 Explain Isolated session my-redshift-cl... dev Schedule

```

1
2
3 WITH vejgaard_atm_trans_data AS (
4   select a.atm_id, a.atm_number, a.atm_manufacturer, l.location, count(f.trans_id) as total_trans_id, d.weekday
5   from spar_nord_bank_atm_data.fact_atm_trans f
6   inner join spar_nord_bank_atm_data.dim_atm a on a.atm_id = f.atm_id

```

Row 3, Col 1, Chr 730

Result 1 (2) Export Chart

atm_id	atm_number	atm_manufacturer	location	total_trans_id	weekday
39	2	NCR	Vejgaard	6290	Friday
10	103	Diebold Nixdorf	Vejgaard	4757	Friday

Query ID 209099 Elapsed time: 10 ms Total rows: 2

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