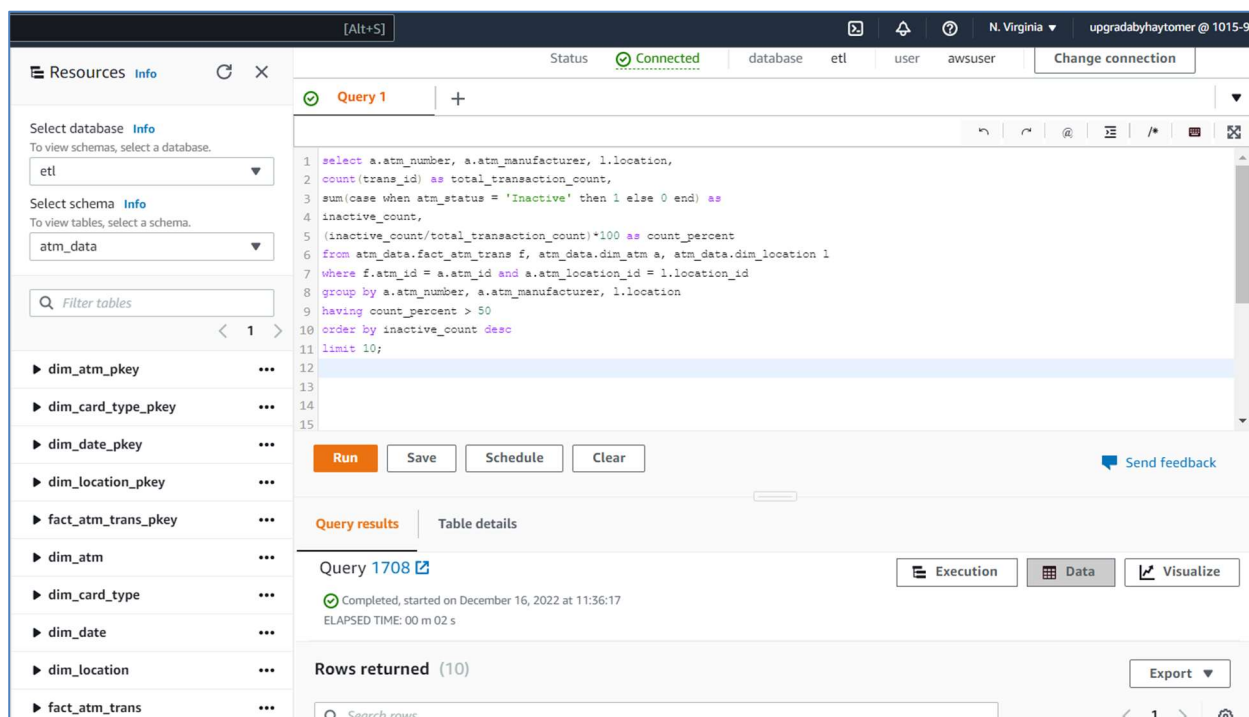


## Solving analytical queries on Redshift Cluster

Here, are the queries 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_count,
(inactive_count/total_transaction_count)*100 as count_percent
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
having count_percent > 50
order by inactive_count desc
limit 10;
```



The screenshot displays the AWS Redshift Query Editor interface. On the left, the 'Resources' panel shows the database 'etl' and schema 'atm\_data'. The main editor contains the SQL query for finding the top 10 ATMs with the highest percentage of inactive transactions. Below the query, the 'Run' button is visible. The 'Query results' tab is active, showing 'Query 1708' completed on December 16, 2022, at 11:36:17, with an elapsed time of 00 m 02 s. The results are displayed as a table with 10 rows returned. The table columns are: atm\_number, atm\_manufacturer, location, total\_transaction\_count, inactive\_count, and count\_percent. The results are ordered by inactive\_count in descending order.

atm_number	atm_manufacturer	location	total_transaction_count	inactive_count	count_percent
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

[Alt+S]		N. Virginia				upgradabyhaytomer @ 1015-972
► dim_location	...	Rows returned (10)				Export ▼
► fact_atm_trans	...	Search rows				< 1 > ⚙
atm_number ▼	atm_manufacture r ▼	location ▼	total_transaction_count ▼	inactive_coun t ▼	count_perce nt ▼	
16	NCR	Skive	44043	44043	100	
12	NCR	Åfjeller Åfjell Duus	33982	33982	100	
2	NCR	Vejgaard	33725	33725	100	
88	NCR	Storcenter indg. A	32183	32183	100	
30	NCR	Nykjeller, bing Mors	30883	30883	100	
52	NCR	Farsjeller,	27361	27361	100	
50	NCR	Aarhus	23416	23416	100	
29	NCR	Skelagervej 15	20773	20773	100	
81	NCR	Spar Kjeller, bmand Tornheller, j	20148	20148	100	
102	NCR	Aalborg Storcenter Afd	18297	18297	100	

## 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_transactions,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_transactions,
case when coalesce(inactive_transactions, 0) = 0 then 0.0000
else trunc((cast(inactive_transactions as
numeric(10,4))/total_transactions)*100, 2)
end as inactive_count_percent
from atm_data.fact_atm_trans f
where f.weather_main != ''
group by f.weather_main
order by inactive_count_percent desc
limit 10;

```

Resources Info

Select database Info

To view schemas, select a database.

etl

Select schema Info

To view tables, select a schema.

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

Query 1

```

1 select f.weather_main,
2 count(trans_id) as total_transactions,
3 sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_transactions,
4 case when coalesce(inactive_transactions, 0) = 0 then 0.0000
5 else trunc((cast(inactive_transactions as
6 numeric(10,4))/total_transactions)*100, 2)
7 end as inactive_count_percent
8 from atm_data.fact_atm_trans f
9 where f.weather_main != ''
10 group by f.weather_main
11 order by inactive_count_percent desc
12 limit 10;

```

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query 1868

Execution

Data

Visualize

Completed, started on December 16, 2022 at 11:49:11

ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export

dim\_location

fact\_atm\_trans

Rows returned (10)

Export

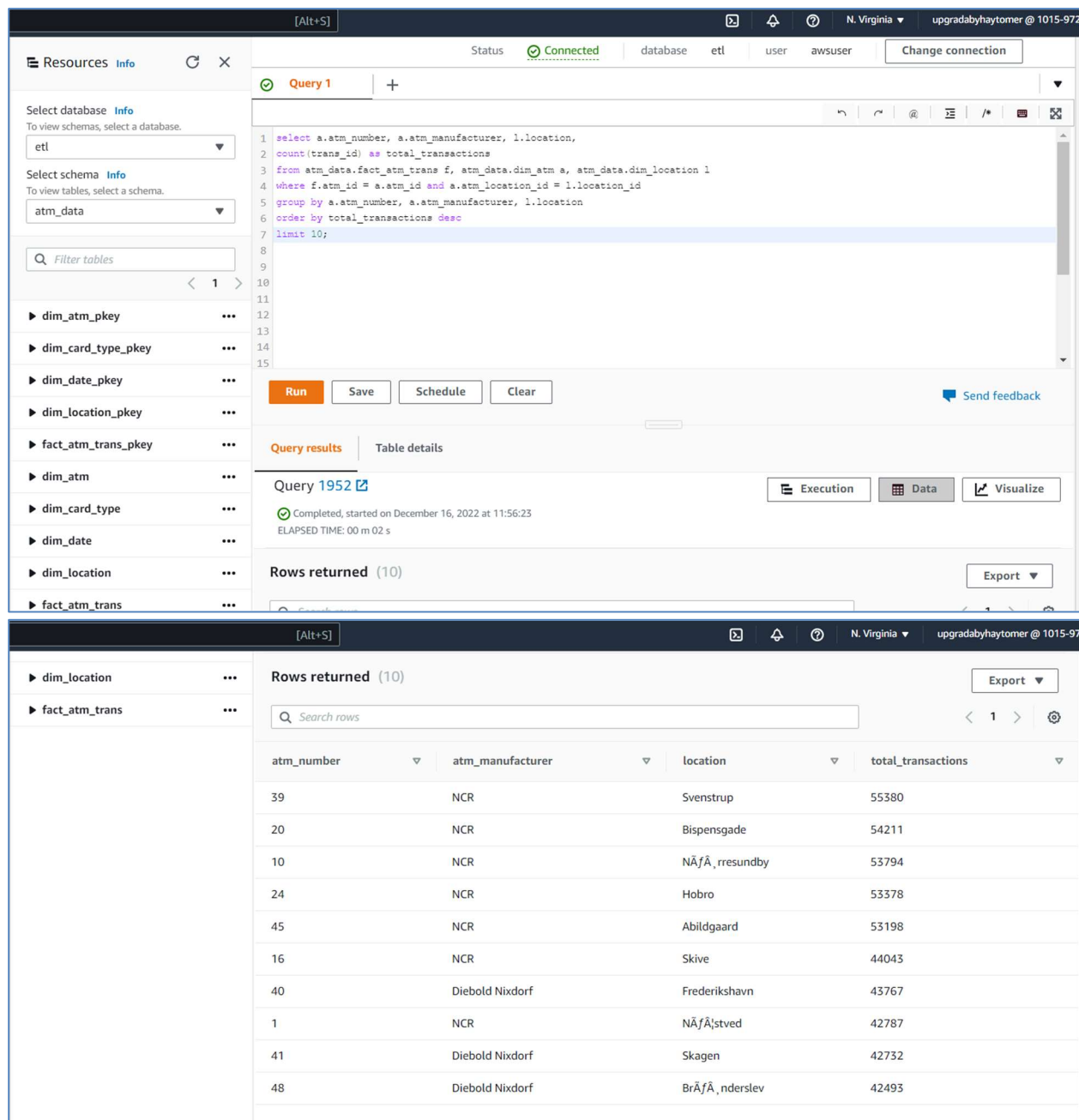
Search rows

1

weather_main	total_transactions	inactive_transactions	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

```
select a.atm_number, a.atm_manufacturer, l.location,
count(trans_id) as total_transactions
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_transactions desc
limit 10;
```



The screenshot displays a data analytics tool interface. The top section shows a SQL query being executed. The query is as follows:

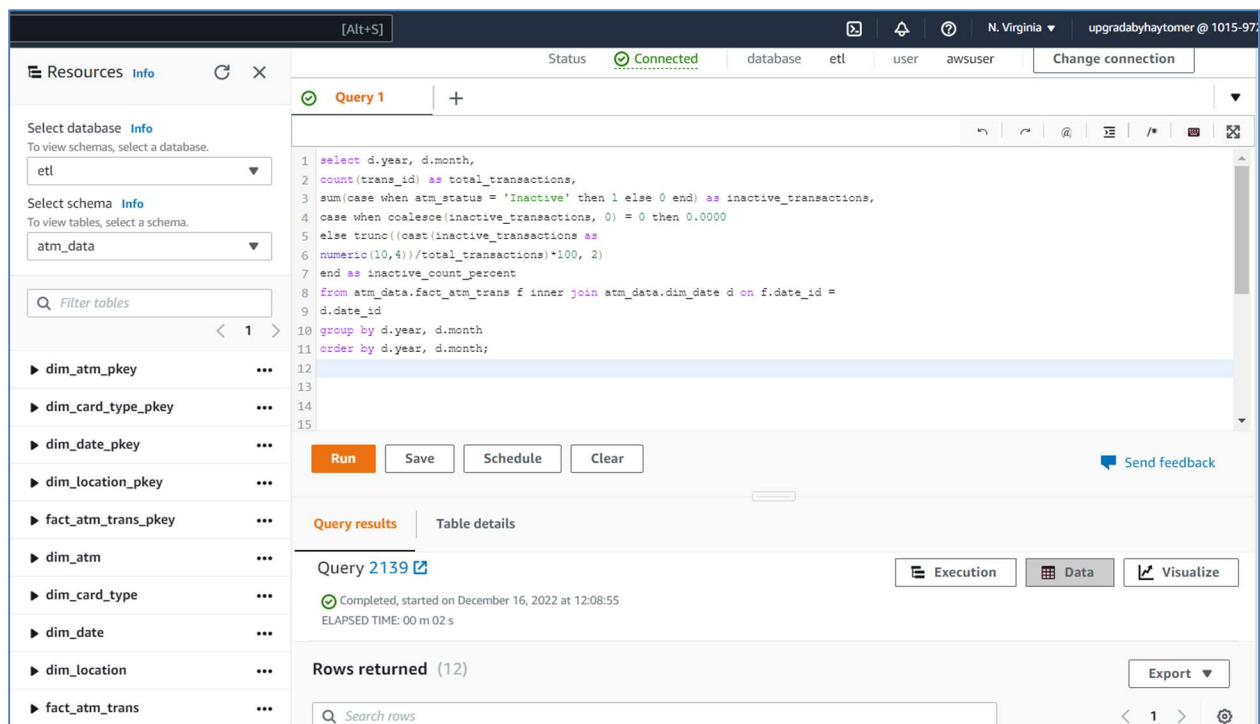
```
1 select a.atm_number, a.atm_manufacturer, l.location,
2 count(trans_id) as total_transactions
3 from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l
4 where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
5 group by a.atm_number, a.atm_manufacturer, l.location
6 order by total_transactions desc
7 limit 10;
```

The query is executed successfully, and the results are displayed in a table. The table has 4 columns: atm\_number, atm\_manufacturer, location, and total\_transactions. The results show the top 10 ATMs with the most transactions.

atm_number	atm_manufacturer	location	total_transactions
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÅfÅ, rresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÅfÅ, stved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÅfÅ, nderslev	42493

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

```
select d.year, d.month,
count(trans_id) as total_transactions,
sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_transactions,
case when coalesce(inactive_transactions, 0) = 0 then 0.0000
else trunc((cast(inactive_transactions as
numeric(10,4))/total_transactions)*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;
```



The screenshot shows a SQL query editor interface. On the left, there is a sidebar with 'Resources' and 'Info' tabs. Under 'Resources', there is a 'Select database' dropdown set to 'etl' and a 'Select schema' dropdown set to 'atm\_data'. Below these, there is a 'Filter tables' search bar and a list of 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 area displays the SQL query for 'Query 1'. The query is as follows:

```
1 select d.year, d.month,
2 count(trans_id) as total_transactions,
3 sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_transactions,
4 case when coalesce(inactive_transactions, 0) = 0 then 0.0000
5 else trunc((cast(inactive_transactions as
6 numeric(10,4))/total_transactions)*100, 2)
7 end as inactive_count_percent
8 from atm_data.fact_atm_trans f inner join atm_data.dim_date d on f.date_id =
9 d.date_id
10 group by d.year, d.month
11 order by d.year, d.month;
```

Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. To the right of these buttons is a 'Send feedback' link. The 'Query results' tab is active, showing 'Query 2139' with a status of 'Completed, started on December 16, 2022 at 12:08:55' and 'ELAPSED TIME: 00 m 02 s'. It indicates 'Rows returned (12)' and has an 'Export' button. At the bottom, there is a search bar for rows and navigation controls.

[Alt+S]

📄

🔔

🔍

N. Virginia ▾

upgradabyhaytomer @ 1015-97

► dim\_location ...

► fact\_atm\_trans ...

Rows returned (12)

Export ▾

🔍 Search rows

< 1 > ⚙

year ▾	month ▾	total_transactions ▾	inactive_transactions ▾	inactive_count_percent ▾
2017	April	209296	37750	18.0300
2017	August	217873	36006	16.5200
2017	December	201203	24847	12.3400
2017	February	179146	34212	19.0900
2017	January	197766	37016	18.7100
2017	July	231684	37046	15.9800
2017	June	203055	35008	17.2400
2017	March	199430	39415	19.7600
2017	May	229414	39361	17.1500
2017	November	196722	23281	11.8300
2017	October	197601	22359	11.3100
2017	September	205382	31357	15.2600

## 5. Top 10 ATMs with the highest total amount withdrawn 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;

```

Resources Info

Select database Info

To view schemas, select a database.

etl

Select schema Info

To view tables, select a schema.

atm\_data

Filter tables

1

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

Query 1

```

1 select a.atm_number, a.atm_manufacturer, l.location,
2 sum(transaction_amount) as total_transaction_amount
3 from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l
4 where f.atm_id = a.atm_id and a.atm_location_id = l.location_id
5 group by a.atm_number, a.atm_manufacturer, l.location
6 order by total_transaction_amount desc
7 limit 10;

```

Run Save Schedule Clear

Send feedback

Query results Table details

Query 2218

Execution Data Visualize

Completed, started on December 16, 2022 at 12:15:30

ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export

Search rows

1

dim\_location

fact\_atm\_trans

Rows returned (10)

Export

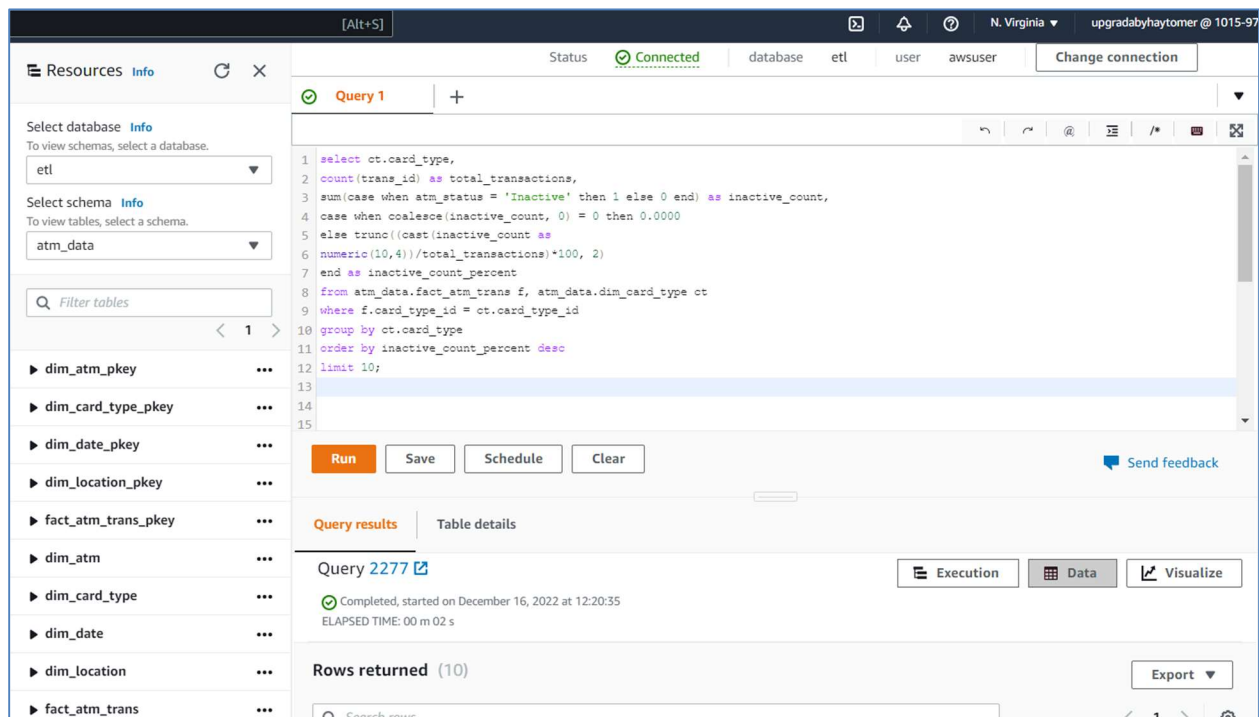
Search rows

1

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	BrÅfÅ_nderslev	212883099

## 6. Number of failed ATM transactions across various card types

```
select ct.card_type,
count(trans_id) as total_transactions,
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_transactions)*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;
```



The screenshot shows a SQL query editor interface. The query is as follows:

```
1 select ct.card_type,
2 count(trans_id) as total_transactions,
3 sum(case when atm_status = 'Inactive' then 1 else 0 end) as inactive_count,
4 case when coalesce(inactive_count, 0) = 0 then 0.0000
5 else trunc((cast(inactive_count as
6 numeric(10,4))/total_transactions)*100, 2)
7 end as inactive_count_percent
8 from atm_data.fact_atm_trans f, atm_data.dim_card_type ct
9 where f.card_type_id = ct.card_type_id
10 group by ct.card_type
11 order by inactive_count_percent desc
12 limit 10;
```

The interface shows the query is executed successfully. The results are displayed in a table with 10 rows. The columns are: card\_type, total\_transactions, inactive\_count, and inactive\_count\_percent. The results are sorted by inactive\_count\_percent in descending order.

Query 2277

Completed, started on December 16, 2022 at 12:20:35  
ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export



[Alt+S]

dim\_location

...

fact\_atm\_trans

...

Rows returned (10)

Export

Search rows

< 1 > ⚙

card_type	total_transactions	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
HÃfÃ\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
HÃfÃ\vekort	8459	1208	14.2800
Visa Dankort	427840	60547	14.1500

## 7. Top 10 records with the number of transactions ordered by the ATM\_number, ATM\_manufacturer, location, weekend\_flag and then total\_transaction\_count, on weekdays and on weekends throughout the year

```

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;

```

Resources Info

Select database Info

To view schemas, select a database.

etl

Select schema Info

To view tables, select a schema.

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

Query 1

```

1 select a.atm_number, a.atm_manufacturer, l.location,
2 case when d.weekday in ('Saturday','Sunday') then 1 else 0 end as
3 weekend_flag,
4 count(trans_id) as total_transaction_count
5 from atm_data.fact_atm_trans f, atm_data.dim_atm a, atm_data.dim_location l,
6 atm_data.dim_date d
7 where f.atm_id = a.atm_id and a.atm_location_id = l.location_id and f.date_id
8 = d.date_id
9 group by a.atm_number, a.atm_manufacturer, l.location, weekend_flag
10 order by a.atm_number, a.atm_manufacturer, l.location, weekend_flag,
11 total_transaction_count
12 limit 10;

```

Run Save Schedule Clear

Send feedback

Query results Table details

Query 2334

Execution Data Visualize

Completed, started on December 16, 2022 at 12:26:14

ELAPSED TIME: 00 m 02 s

Rows returned (10)

Export

Search rows

dim\_location

fact\_atm\_trans

Rows returned (10)

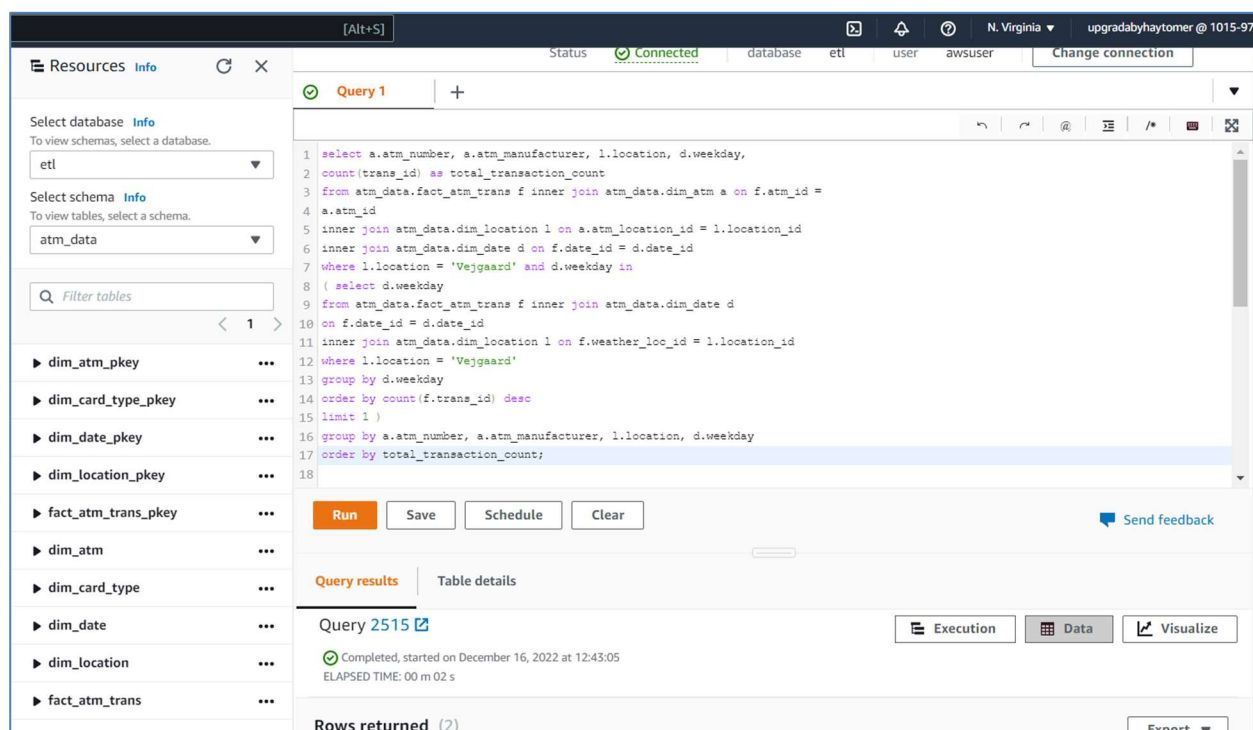
Export

Search rows

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃfÃstved	0	31448
1	NCR	NÃfÃstved	1	11339
10	NCR	NÃfÃ_rresundby	0	39935
10	NCR	NÃfÃ_rresundby	1	13859
100	NCR	Intern Skive	0	15092
100	NCR	Intern Skive	1	2721
101	NCR	Bryggen Vejle	0	11051
101	NCR	Bryggen Vejle	1	3889
102	NCR	Aalborg Storcenter Afd	0	13475
102	NCR	Aalborg Storcenter Afd	1	4822

## 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;
```



The screenshot shows a SQL query editor interface. On the left, there is a sidebar with a tree view of databases and schemas. The 'etl' database is selected, and the 'atm\_data' schema is chosen. Below this, a list of tables is shown, 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 area displays a SQL query (Query 1) that finds the most active day in each ATM from the location 'Vejgaard'. The query is as follows:

```
1 select a.atm_number, a.atm_manufacturer, l.location, d.weekday,
2 count(trans_id) as total_transaction_count
3 from atm_data.fact_atm_trans f inner join atm_data.dim_atm a on f.atm_id =
4 a.atm_id
5 inner join atm_data.dim_location l on a.atm_location_id = l.location_id
6 inner join atm_data.dim_date d on f.date_id = d.date_id
7 where l.location = 'Vejgaard' and d.weekday in
8 ( select d.weekday
9 from atm_data.fact_atm_trans f inner join atm_data.dim_date d
10 on f.date_id = d.date_id
11 inner join atm_data.dim_location l on f.weather_loc_id = l.location_id
12 where l.location = 'Vejgaard'
13 group by d.weekday
14 order by count(f.trans_id) desc
15 limit 1 )
16 group by a.atm_number, a.atm_manufacturer, l.location, d.weekday
17 order by total_transaction_count;
```

Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button is highlighted. To the right of the buttons is a 'Send feedback' link. Below the query, there is a section for 'Query results' and 'Table details'. The 'Query results' section shows 'Query 2515' and indicates that the query is completed, started on December 16, 2022 at 12:43:05, with an elapsed time of 00 m 02 s. The 'Table details' section shows 'Rows returned (2)' and an 'Export' button.

[Alt+S]		N. Virginia		upgradabyhaytomer @ 1015-97
► dim_date	...	Query 2515		
► dim_location	...	Execution Data Visualize		
► fact_atm_trans	...	Completed, started on December 16, 2022 at 12:43:05 ELAPSED TIME: 00 m 02 s		
		Rows returned (2)		
		Search rows		
		Export		
		< 1 > ⚙		
		atm_number	atm_manufacturer	location
		weekday	total_transaction_count	
		103	Diebold Nixdorf	Vejgaard
		Friday	3947	
		2	NCR	Vejgaard
		Friday	5012	