

## Solving analytical queries on Redshift Cluster

### 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,
(inactive_transaction_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_transaction_count desc limit 10;
```

Completed, started on October 04, 2022 at 21:16:38  
ELAPSED TIME: 00 m 02 s

Rows returned (10) Export ▼

Search rows

atm_number ▼	atm_manufacturer ▼	location ▼	total_transaction_count ▼	inactive_transaction_count ▼	count_percent ▼
16	NCR	Skive	44043	44043	100
20	NCR	Bispensgade	33982	33982	100
1	NCR	NÃfÃstved	33725	33725	100
8	NCR	GlyngÃfÃ, re	32183	32183	100
41	Diebold Nixdorf	Skagen	30883	30883	100
74	NCR	Jebjerg	27361	27361	100
50	NCR	Aarhus	23416	23416	100
98	NCR	Odense	20773	20773	100
81	NCR	Spar KÃfÃ, bmand TornhÃfÃ, 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_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 atm_data.fact_atm_trans f where f.weather_main != '' group by f.weather_main
order by inactive_count_percent desc limit 10;
```

Rows returned (10)				Export
Search rows				
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

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

Query 1908				Execution	Data	Visualize
Completed, started on October 04, 2022 at 21:20:38 ELAPSED TIME: 00 m 03 s						
Rows returned (10)				Export		
Search rows				< 1 > ⚙		
atm_number	atm_manufacturer	location	total_transaction_count			
35	NCR	Aabybro	55380			
86	NCR	Hillerød Idrætsscenter	54211			
10	NCR	Næstved, resundby	53794			
24	NCR	Hobro	53378			
45	NCR	Abildgaard	53198			
16	NCR	Skive	44043			
40	Diebold Nixdorf	Frederikshavn	43767			
6	NCR	Fredericia	42787			
99	NCR	Intern Vejle	42732			
48	Diebold Nixdorf	Brønderslev	42493			

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

```
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 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
```

Query results

Table details

Query 74

Completed, started on August 18, 2021 at 23:25:17

ELAPSED TIME: 00 m 02 s

Execution

Data

Visualize

Rows returned (12)

Export

Q Search rows

< 1 > ⌂

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
2017	October	191667	21780	11.3600
2017	September	202101	28913	14.3000

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 Top 10 ATMs with the highest total withdrawn amount 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

Query 162 [🔗](#)

Completed, started on August 18, 2021 at 23:33:06  
ELAPSED TIME: 00 m 08 s

Execution Data Visualize

Rows returned (10)

Export

Search rows

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_nderlev	212883099

## 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 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;
```

Query 258 [Query](#)

Completed, started on August 18, 2021 at 23:43:16  
ELAPSED TIME: 00 m 09 s

Rows returned (10)

Search rows

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
MASTERCARD - on-us	62487	10331	16.5300
Dankort	28581	4557	15.9400
MasterCard	400507	63482	15.8500
Visa Dankort - on-us	748805	112972	15.0800
MASTERCARD	8459	1208	14.2800
Visa Dankort	427840	60547	14.1500

Feedback English (US) © 2008 - 2021 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preference

**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 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;
```

Query 359

Completed, started on August 18, 2021 at 23:53:40  
ELAPSED TIME: 00 m 13 s

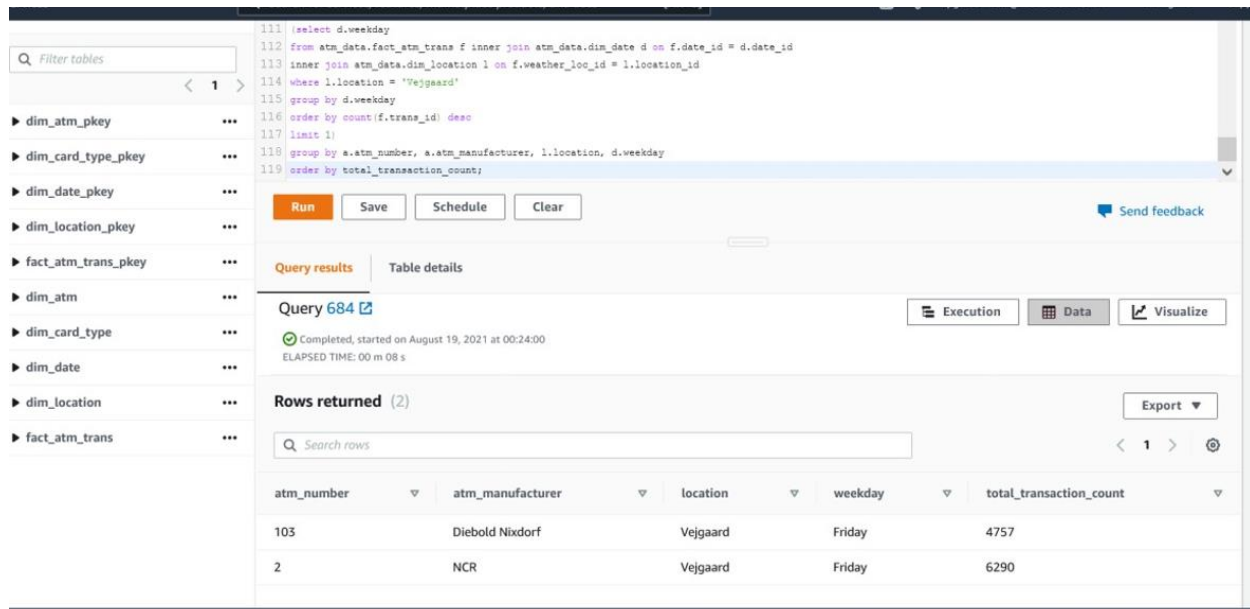
Rows returned (10)

Search rows

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃfÃstved	0	32711
1	NCR	NÃfÃstved	1	10076
10	NCR	NÃfÃ_rresundby	0	41667
10	NCR	NÃfÃ_rresundby	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"

```
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.location_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 execution interface. The query is as follows:

```
111 select d.weekday
112 from atm_data.fact_atm_trans f inner join atm_data.dim_date d on f.date_id = d.date_id
113 inner join atm_data.dim_location l on f.location_id = l.location_id
114 where l.location = 'Vejgaard'
115 group by d.weekday
116 order by count(f.trans_id) desc
117 limit 1)
118 group by a.atm_number, a.atm_manufacturer, l.location, d.weekday
119 order by total_transaction_count;
```

The query results are displayed in a table with the following columns: atm\_number, atm\_manufacturer, location, weekday, and total\_transaction\_count. The results show two rows:

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