

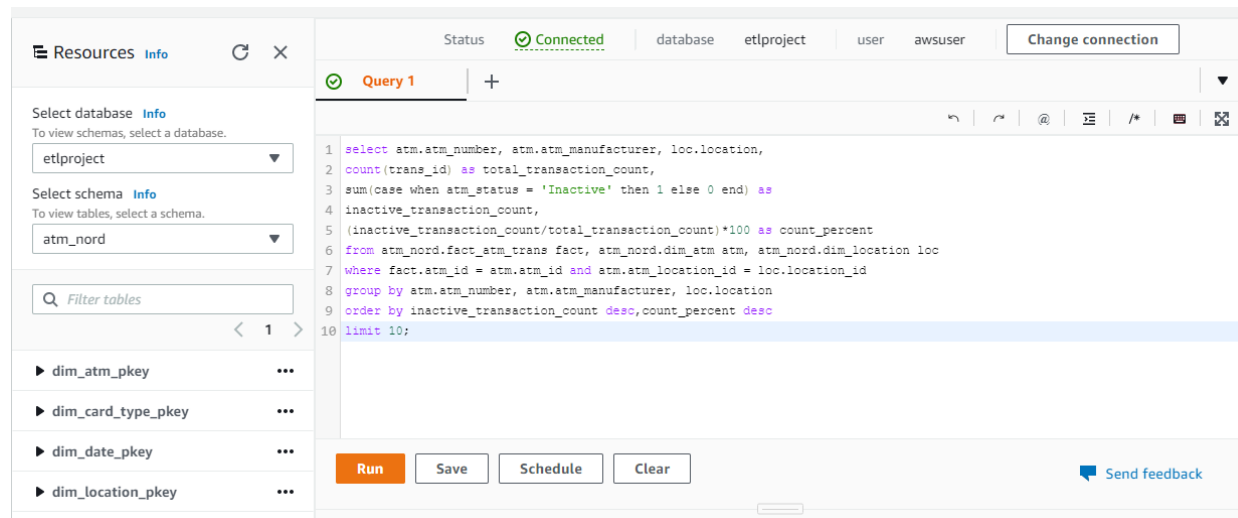
Solving analytical queries on Redshift Cluster

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

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.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_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
group by atm.atm_number, atm.atm_manufacturer, loc.location
order by inactive_transaction_count desc,count_percent desc
limit 10;
```



The screenshot shows the AWS Redshift Query Editor interface. On the left, the 'Resources' panel displays the database 'etlproject' and schema 'atm_nord'. The main editor area shows the SQL query for finding the top 10 ATMs with the most inactive transactions. The query is as follows:

```
1 select atm.atm_number, atm.atm_manufacturer, loc.location,
2 count(trans_id) as total_transaction_count,
3 sum(case when atm_status = 'Inactive' then 1 else 0 end) as
4 inactive_transaction_count,
5 (inactive_transaction_count/total_transaction_count)*100 as count_percent
6 from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
7 where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
8 group by atm.atm_number, atm.atm_manufacturer, loc.location
9 order by inactive_transaction_count desc,count_percent desc
10 limit 10;
```

Below the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also present.

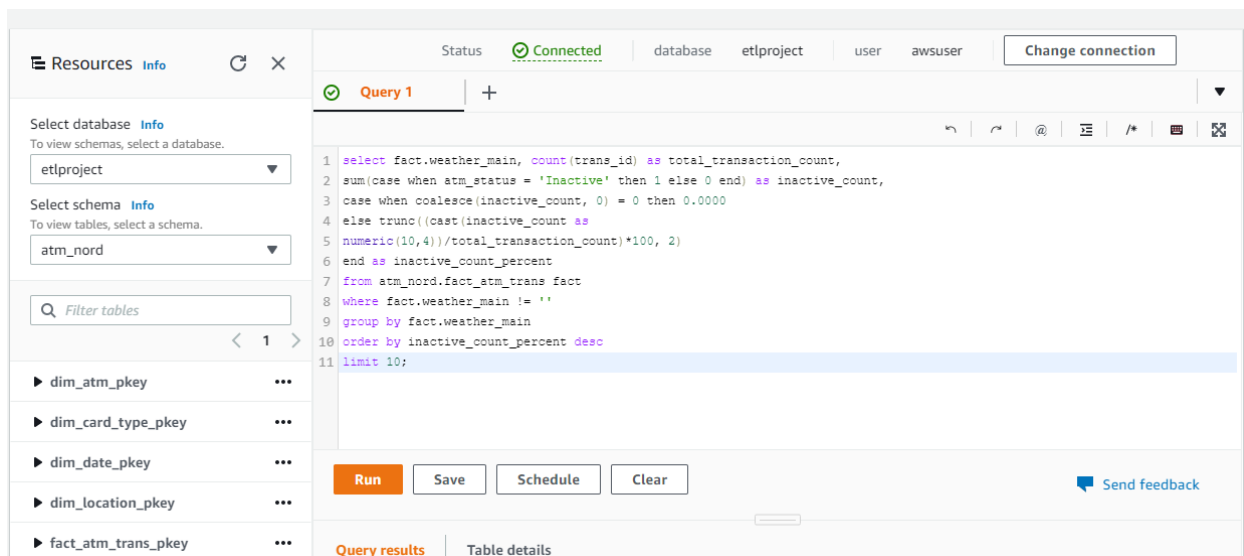
Screenshot of the resultant table

Search for services, features, blogs, docs, and more [Alt+S]				
atm_number	atm_manufacturer	location	total_transaction_count	inactive_transaction_count
16	NCR	Skive	44043	44043
12	NCR	Åfjølster, Åfjøl V Duus	33982	33982
2	NCR	Vejgaard	33725	33725
88	NCR	Storcenter indg. A	32183	32183
30	NCR	Nykjøl, bing Mors	30883	30883
52	NCR	Farsjøl,	27361	27361
50	NCR	Aarhus	23416	23416
29	NCR	Skelagervej 15	20773	20773
81	NCR	Spar Kjøl, bmand Tornhjel, j	20148	20148
102	NCR	Aalborg Storcenter Afd	18297	18297

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

Query:

```
select fact.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_nord.fact_atm_trans fact
where fact.weather_main != ''
group by fact.weather_main
order by inactive_count_percent desc
limit 10;
```



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Under 'Resources', it shows 'Select database' (etlproject) and 'Select schema' (atm_nord). Below that, there's a 'Filter tables' search bar and a list of tables: dim_atm_pkey, dim_card_type_pkey, dim_date_pkey, dim_location_pkey, and fact_atm_trans_pkey. The main area displays the SQL query for 'Query 1'. The query is:

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

Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. At the bottom, there are tabs for 'Query results' and 'Table details'.

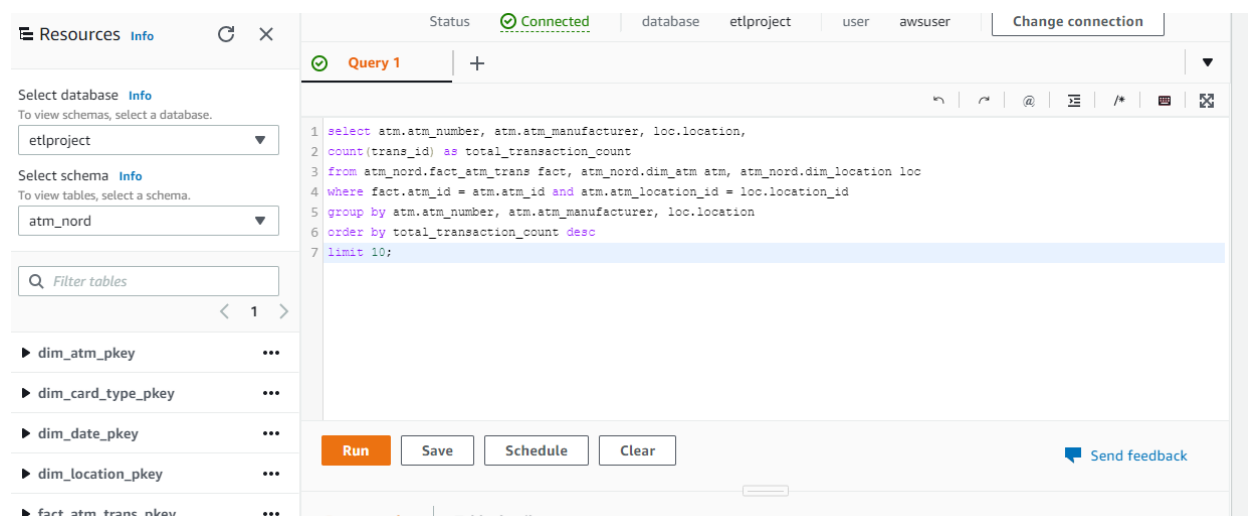
Screenshot of the resultant table:

Services	Search for services, features, blogs, docs, and more	[Alt+S]	N. Virginia	upgradpurvipadiya @ 9663-6614-31
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

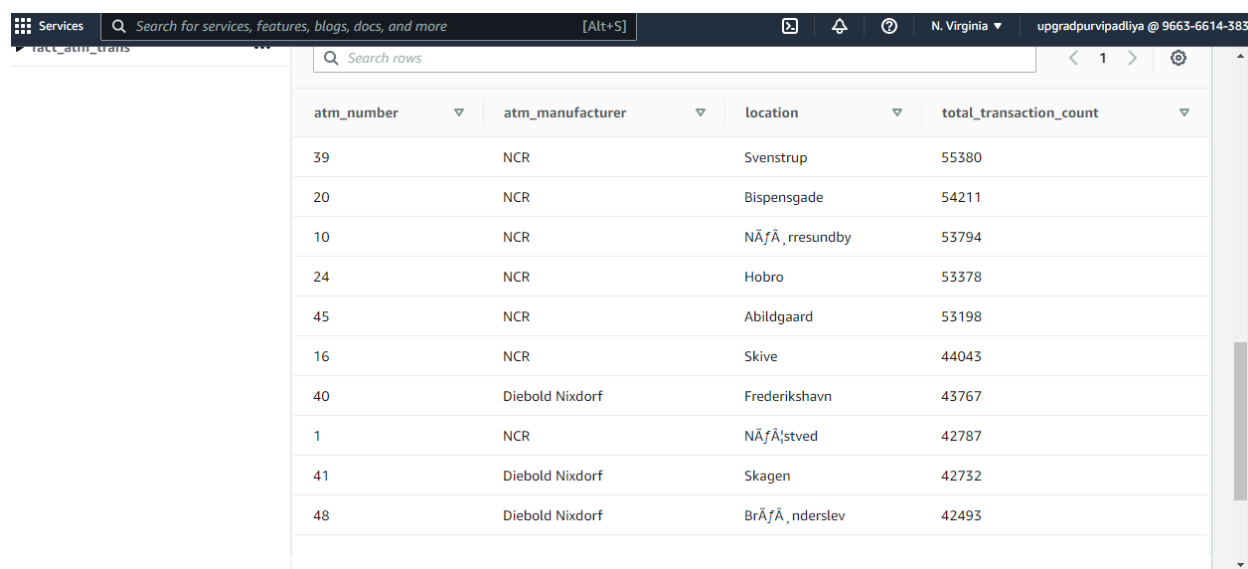
Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location,
count(trans_id) as total_transaction_count
from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
group by atm.atm_number, atm.atm_manufacturer, loc.location
order by total_transaction_count desc
limit 10;
```



The screenshot shows a database query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Below 'Resources', there's a 'Select database' dropdown set to 'etlproject' and a 'Select schema' dropdown set to 'atm_nord'. A search bar for tables is also present. The main area shows the SQL query for 'Query 1'. The query is:
 1 select atm.atm_number, atm.atm_manufacturer, loc.location,
 2 count(trans_id) as total_transaction_count
 3 from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
 4 where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
 5 group by atm.atm_number, atm.atm_manufacturer, loc.location
 6 order by total_transaction_count desc
 7 limit 10;
 Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also visible.

Screenshot of the resultant table:



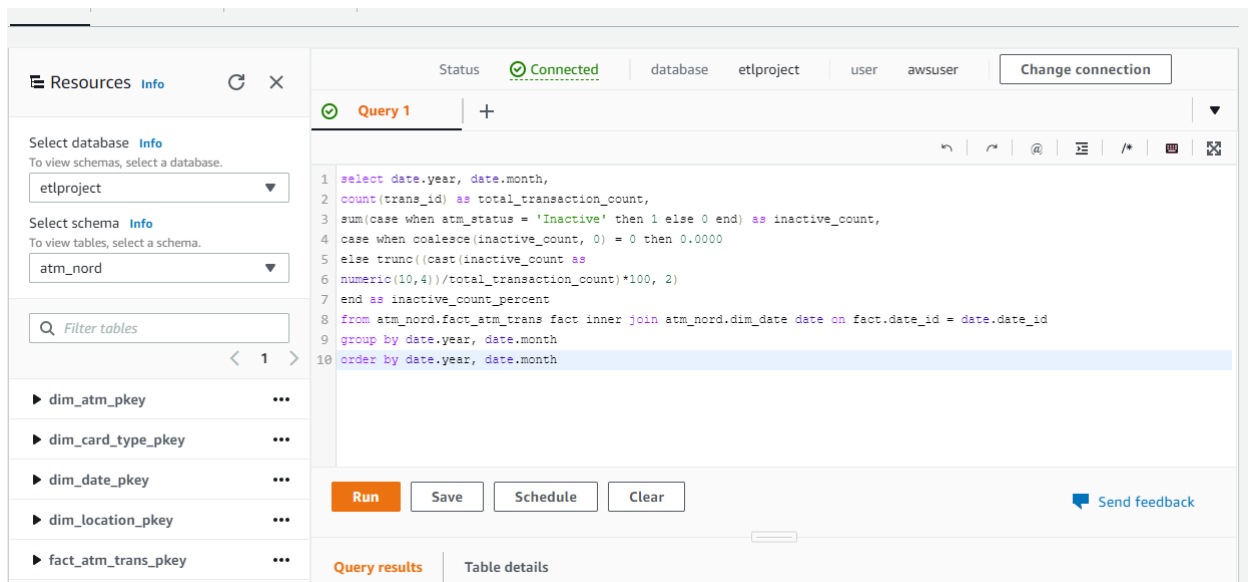
The screenshot shows a table viewer interface with a search bar and a table of results. The table has four columns: 'atm_number', 'atm_manufacturer', 'location', and 'total_transaction_count'. The results are as follows:

atm_number	atm_manufacturer	location	total_transaction_count
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

Query:

```
select date.year, date.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_nord.fact_atm_trans fact inner join atm_nord.dim_date date on fact.date_id =
date.date_id
group by date.year, date.month
order by date.year, date.month
```



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Below 'Resources', there's a 'Select database' dropdown set to 'etlproject' and a 'Select schema' dropdown set to 'atm_nord'. A 'Filter tables' search bar is also present. Below the search bar, a list of tables is shown: 'dim_atm_pkey', 'dim_card_type_pkey', 'dim_date_pkey', 'dim_location_pkey', and 'fact_atm_trans_pkey'. The main area displays the SQL query, which is the same as the one provided in the text. Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. At the bottom, there are tabs for 'Query results' and 'Table details'. The status bar at the top indicates 'Connected' to the 'etlproject' database using the 'awsuser' user.

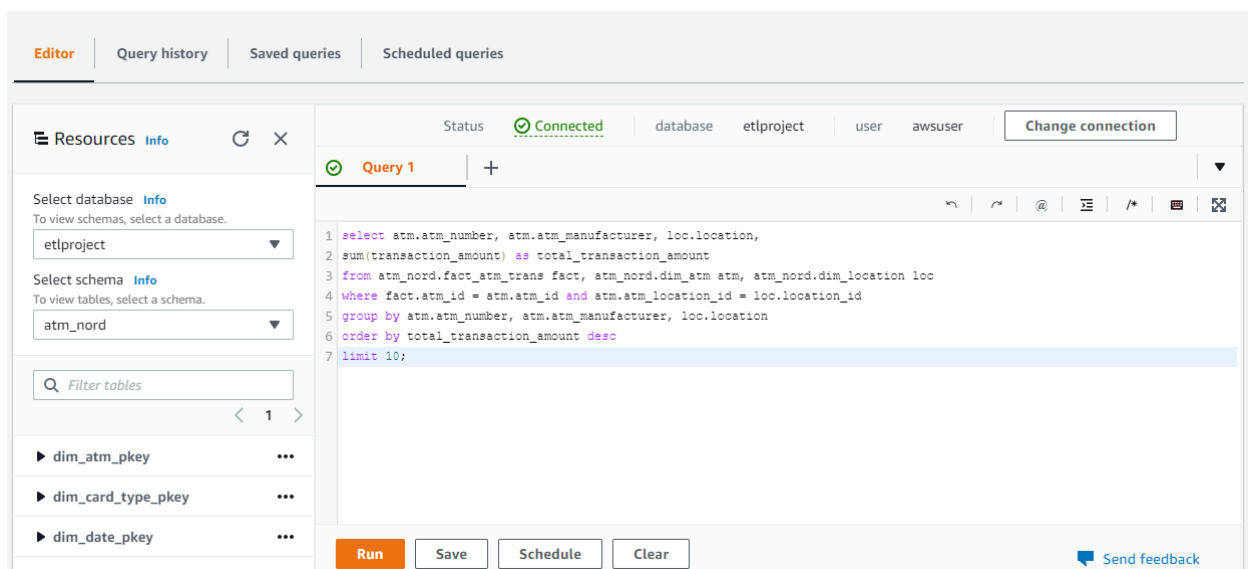
Screenshot of the resultant table:

Services					
Search for services, features, blogs, docs, and more					
[Alt+S]					
N. Virginia					
upgradpurvipadiya @ 9663-6614-3836					
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

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location,
sum(transaction_amount) as total_transaction_amount
from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
group by atm.atm_number, atm.atm_manufacturer, loc.location
order by total_transaction_amount desc
limit 10;
```



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Under 'Resources', it shows 'Select database' (etlproject) and 'Select schema' (atm_nord). Below that, there's a 'Filter tables' search bar and a list of tables: dim_atm_pkey, dim_card_type_pkey, and dim_date_pkey. The main area displays the SQL query for 'Query 1'. The query is:
 1 select atm.atm_number, atm.atm_manufacturer, loc.location,
 2 sum(transaction_amount) as total_transaction_amount
 3 from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc
 4 where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id
 5 group by atm.atm_number, atm.atm_manufacturer, loc.location
 6 order by total_transaction_amount desc
 7 limit 10;
 At the bottom of the editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also present.

Screenshot of the resultant table:

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

upgradpurvipadiya @ 9663-6614-3836

fact_atm_trans

Search rows

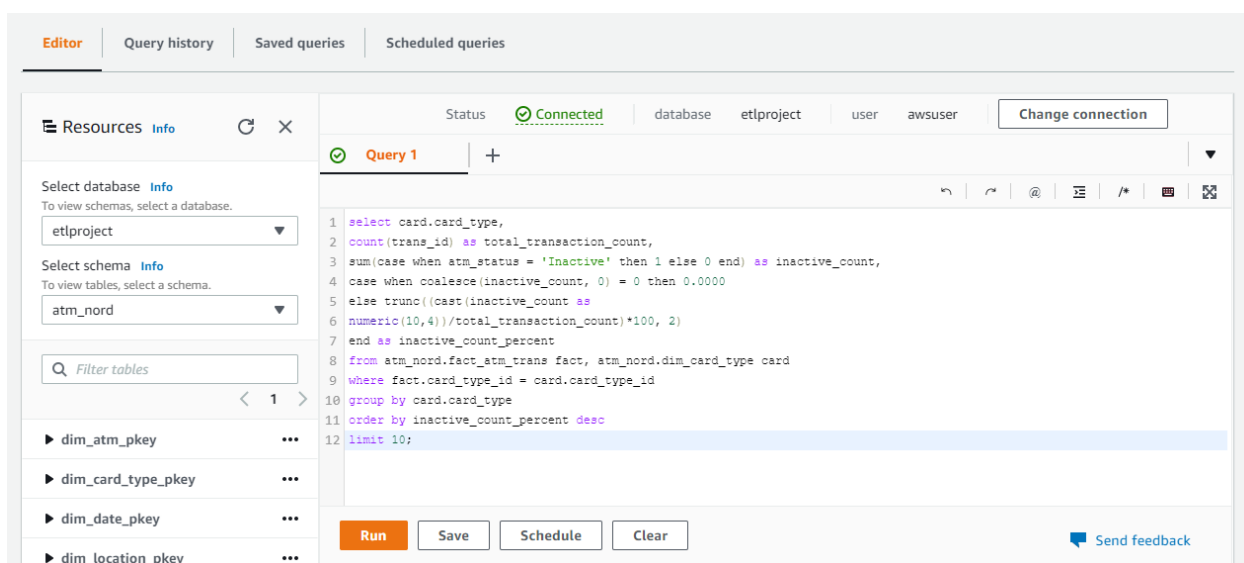
< 1 >

atm_number	atm_manufacturer	location	total_transaction_amount
39	NCR	Svenstrup	277097637
20	NCR	Bispensgade	271008803
24	NCR	Hobro	268289882
10	NCR	NÃfÃ, rresundby	267379103
45	NCR	Abildgaard	265639616
16	NCR	Skive	220677013
40	Diebold Nixdorf	Frederikshavn	219812287
41	Diebold Nixdorf	Skagen	214127315
1	NCR	NÃfÃstved	213721117
48	Diebold Nixdorf	BrÃfÃ, nderslev	212883099

6. Number of failed ATM transactions across various card types

Query:

```
select card.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_nord.fact_atm_trans fact, atm_nord.dim_card_type card
where fact.card_type_id = card.card_type_id
group by card.card_type
order by inactive_count_percent desc
limit 10;
```



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Under 'Resources', it shows the selected database 'etlproject' and schema 'atm_nord'. Below this, there's a list of tables: 'dim_atm_pkey', 'dim_card_type_pkey', 'dim_date_pkey', and 'dim_location_pkey'. The main area displays the SQL query for 'Query 1'. The query is:


```
1 select card.card_type,
2 count(trans_id) as total_transaction_count,
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_transaction_count)*100, 2)
7 end as inactive_count_percent
8 from atm_nord.fact_atm_trans fact, atm_nord.dim_card_type card
9 where fact.card_type_id = card.card_type_id
10 group by card.card_type
11 order by inactive_count_percent desc
12 limit 10;
```

 At the bottom of the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also present.

Screenshot of the resultant table:

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

upgradpurvipadiya @ 9663-6614-3

dim_date

dim_location

fact_atm_trans

Rows returned (10)

Export

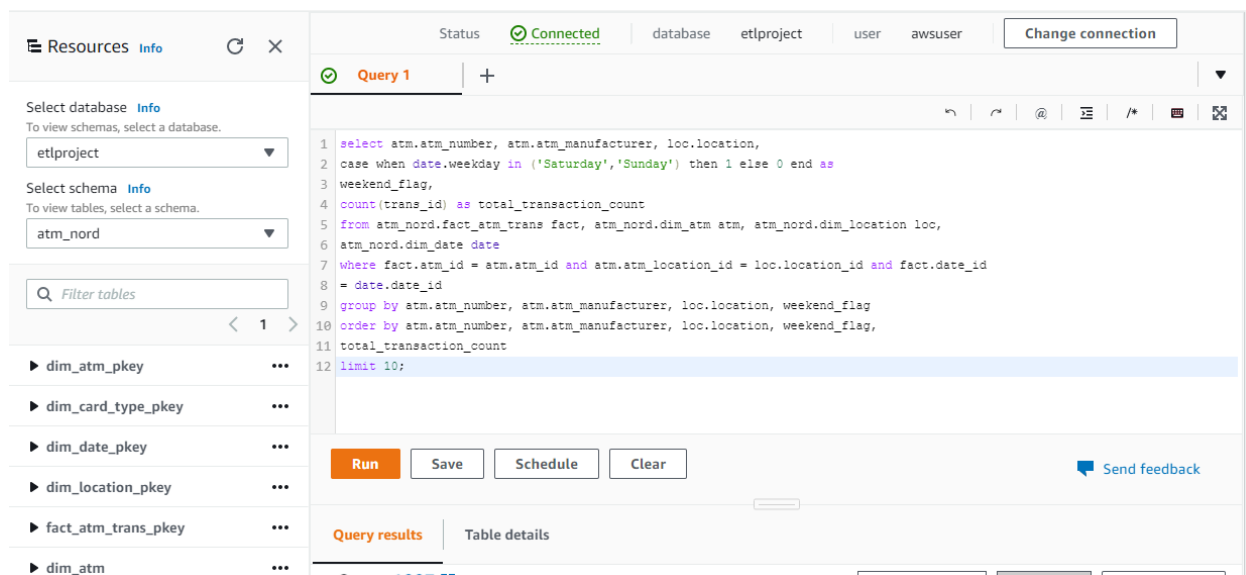
Search rows

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
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. 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 atm.atm_number, atm.atm_manufacturer, loc.location,
case when date.weekday in ('Saturday','Sunday') then 1 else 0 end as
weekend_flag,
count(trans_id) as total_transaction_count
from atm_nord.fact_atm_trans fact, atm_nord.dim_atm atm, atm_nord.dim_location loc,
atm_nord.dim_date date
where fact.atm_id = atm.atm_id and atm.atm_location_id = loc.location_id and fact.date_id
= date.date_id
group by atm.atm_number, atm.atm_manufacturer, loc.location, weekend_flag
order by atm.atm_number, atm.atm_manufacturer, loc.location, weekend_flag,
total_transaction_count
limit 10;
```



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Below 'Resources', there's a 'Select database' dropdown set to 'etlproject' and a 'Select schema' dropdown set to 'atm_nord'. A 'Filter tables' search bar is also present. The main area displays the SQL query from the previous block. Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also visible. At the bottom, there are tabs for 'Query results' and 'Table details'.

Screenshot of the resultant table:

Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia upgradpurvipadiya @ 9663-6614-3836

fact_atm_trans

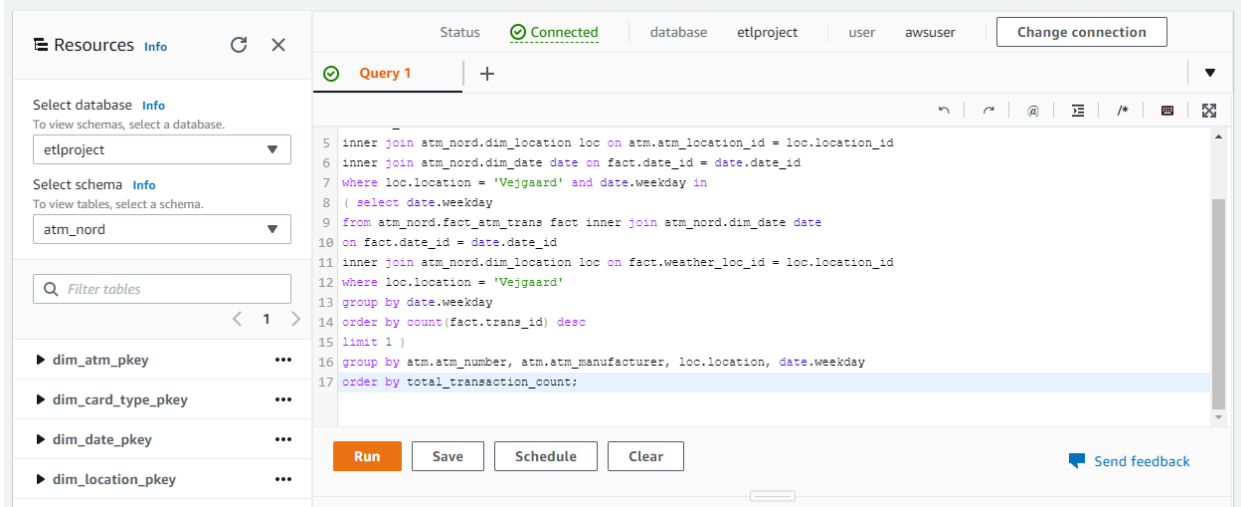
Search rows

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"

Query:

```
select atm.atm_number, atm.atm_manufacturer, loc.location, date.weekday,
count(trans_id) as total_transaction_count
from atm_nord.fact_atm_trans fact inner join atm_nord.dim_atm atm on fact.atm_id =
atm.atm_id
inner join atm_nord.dim_location loc on atm.atm_location_id = loc.location_id
inner join atm_nord.dim_date date on fact.date_id = date.date_id
where loc.location = 'Vejgaard' and date.weekday in
( select date.weekday
from atm_nord.fact_atm_trans fact inner join atm_nord.dim_date date
on fact.date_id = date.date_id
inner join atm_nord.dim_location loc on fact.weather_loc_id = loc.location_id
where loc.location = 'Vejgaard'
group by date.weekday
order by count(fact.trans_id) desc
limit 1 )
group by atm.atm_number, atm.atm_manufacturer, loc.location, date.weekday
order by total_transaction_count;
```



The screenshot shows a SQL query editor interface. On the left, there's a sidebar with 'Resources' and 'Info' tabs. Under 'Resources', it shows 'Select database' (etlproject) and 'Select schema' (atm_nord). Below that, there's a 'Filter tables' search bar and a list of tables: dim_atm_pkey, dim_card_type_pkey, dim_date_pkey, and dim_location_pkey. The main area displays the SQL query for 'Query 1'. The query is:


```
inner join atm_nord.dim_location loc on atm.atm_location_id = loc.location_id
inner join atm_nord.dim_date date on fact.date_id = date.date_id
where loc.location = 'Vejgaard' and date.weekday in
( select date.weekday
from atm_nord.fact_atm_trans fact inner join atm_nord.dim_date date
on fact.date_id = date.date_id
inner join atm_nord.dim_location loc on fact.weather_loc_id = loc.location_id
where loc.location = 'Vejgaard'
group by date.weekday
order by count(fact.trans_id) desc
limit 1 )
group by atm.atm_number, atm.atm_manufacturer, loc.location, date.weekday
order by total_transaction_count;
```

 At the bottom of the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also present.

Screenshot of the resultant table:

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

upgradpurvipadliya @ 9663-6614-383

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 1098

Execution

Data

Visualize

Completed, started on June 02, 2022 at 23:27:53

ELAPSED TIME: 00 m 03 s

Rows returned (2)

Export

Search rows

< 1 >

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