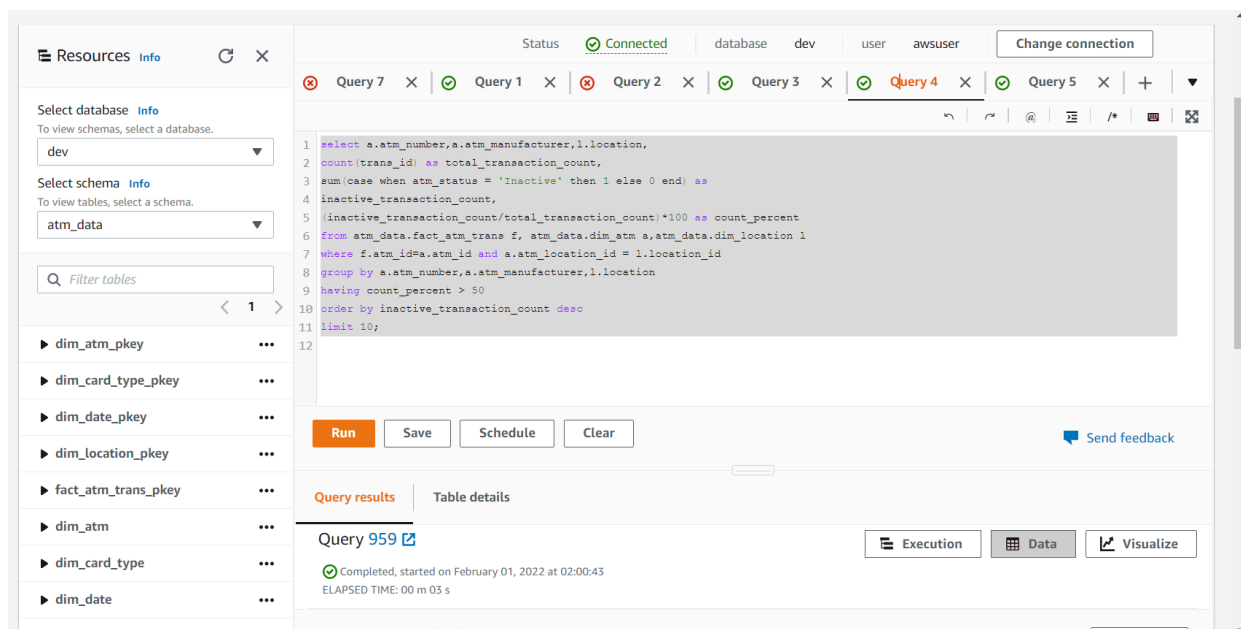


Solving analytical queries on Redshift Cluster

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 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;
```



The screenshot displays the AWS Redshift Query Editor interface. On the left, the 'Resources' panel shows the database 'dev' and schema 'atm_data'. The main editor area contains a SQL query (Query 4) that identifies the top 10 ATMs with the highest percentage of inactive transactions. The query is as follows:

```
1 select a.atm_number,a.atm_manufacturer,l.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_data.fact_atm_trans f, atm_data.dim_atm a,atm_data.dim_location l
7 where f.atm_id=a.atm_id and a.atm_location_id = l.location_id
8 group by a.atm_number,a.atm_manufacturer,l.location
9 having count_percent > 50
10 order by inactive_transaction_count desc
11 limit 10;
```

Below the query editor, the 'Run' button is highlighted. The 'Query results' tab is active, showing the execution status: 'Completed, started on February 01, 2022 at 02:00:43' and 'ELAPSED TIME: 00 m 03 s'. The results are displayed in a table format, with options for 'Execution', 'Data', and 'Visualize'.

▶ dim_date...

▶ dim_location...

▶ fact_atm_trans...

ELAPSED TIME: 00 m 03 s

Rows returned (10)

Export ▼

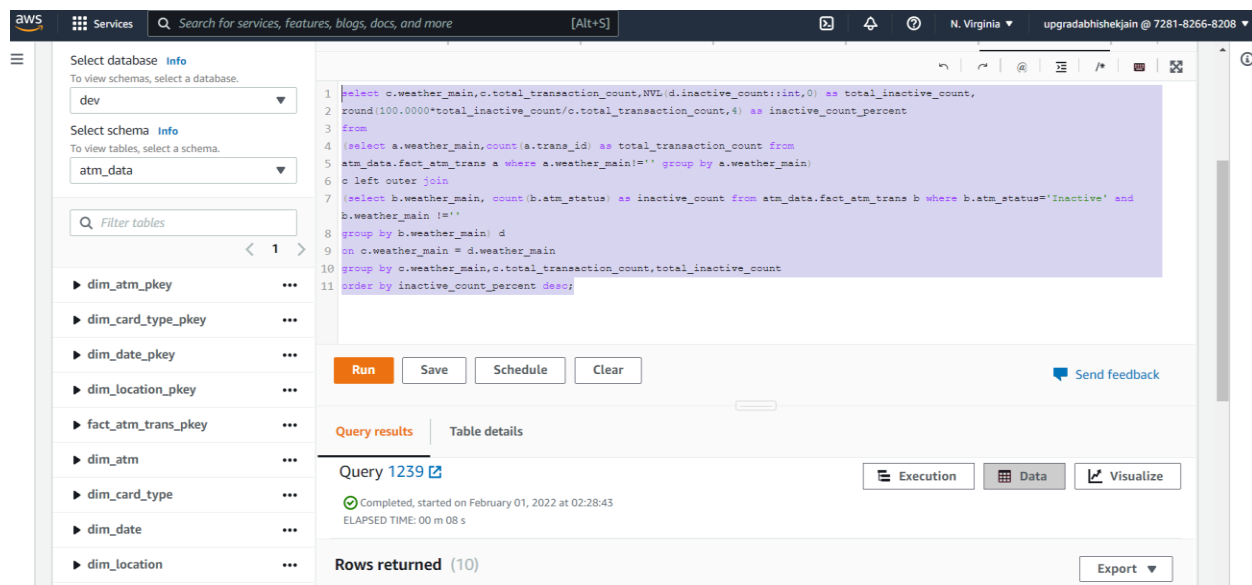
Search rows

< 1 > ⚙

atm_number ▼	atm_manufacture r ▼	location ▼	total_transaction_coun t ▼	inactive_transaction_count ▼	count_pe rcent
16	NCR	Skive	44043	44043	100
68	NCR	Intern Vejle	33982	33982	100
1	NCR	NÃfÃstved	33725	33725	100
8	NCR	GlyngÃfÃ, re	32183	32183	100
30	NCR	NykÃfÃ, bing Mors	30883	30883	100
52	NCR	FarsÃfÃ,	27361	27361	100
50	NCR	Aarhus	23416	23416	100
29	NCR	Skelagervej 15	20773	20773	100
28	NCR	LÃfÃ, gstÃfÃ, r	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

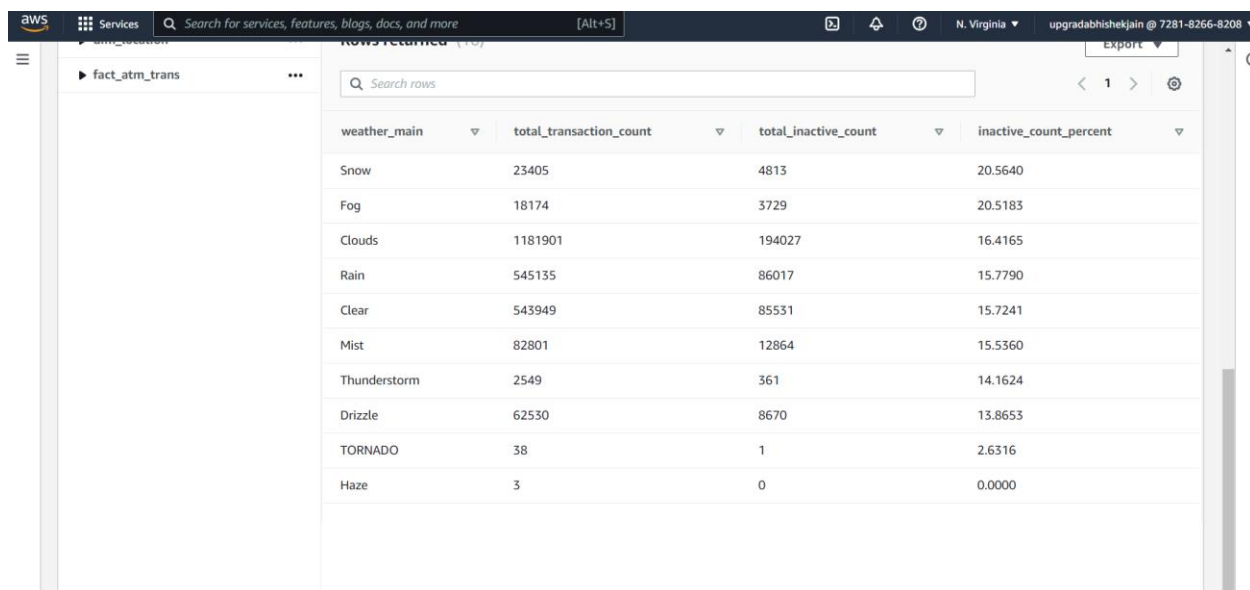
```
select c.weather_main,c.total_transaction_count,NVL(d.inactive_count::int,0) as
total_inactive_count,
round(100.0000*total_inactive_count/c.total_transaction_count,4) as inactive_count_percent
from
(select a.weather_main,count(a.trans_id) as total_transaction_count from
atm_data.fact_atm_trans a where a.weather_main!='' group by a.weather_main)
c left outer join
(select b.weather_main, count(b.atm_status) as inactive_count from atm_data.fact_atm_trans b
where b.atm_status='Inactive' and b.weather_main !=''
group by b.weather_main) d
on c.weather_main = d.weather_main
group by c.weather_main,c.total_transaction_count,total_inactive_count
order by inactive_count_percent desc;
```



The screenshot shows the AWS Redshift console interface. On the left, the 'Select database' dropdown is set to 'dev' and the 'Select schema' dropdown is set to 'atm_data'. Below these, a list of tables is visible, 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', and 'dim_location'. The main area displays a SQL query that calculates the percentage of ATM failures (inactive status) for each weather condition. The query is as follows:

```
1 select c.weather_main,c.total_transaction_count,NVL(d.inactive_count::int,0) as total_inactive_count,
2 round(100.0000*total_inactive_count/c.total_transaction_count,4) as inactive_count_percent
3 from
4 (select a.weather_main,count(a.trans_id) as total_transaction_count from
5 atm_data.fact_atm_trans a where a.weather_main!='' group by a.weather_main)
6 c left outer join
7 (select b.weather_main, count(b.atm_status) as inactive_count from atm_data.fact_atm_trans b where b.atm_status='Inactive' and
8 b.weather_main !=''
9 group by b.weather_main) d
10 on c.weather_main = d.weather_main
11 group by c.weather_main,c.total_transaction_count,total_inactive_count
12 order by inactive_count_percent desc;
```

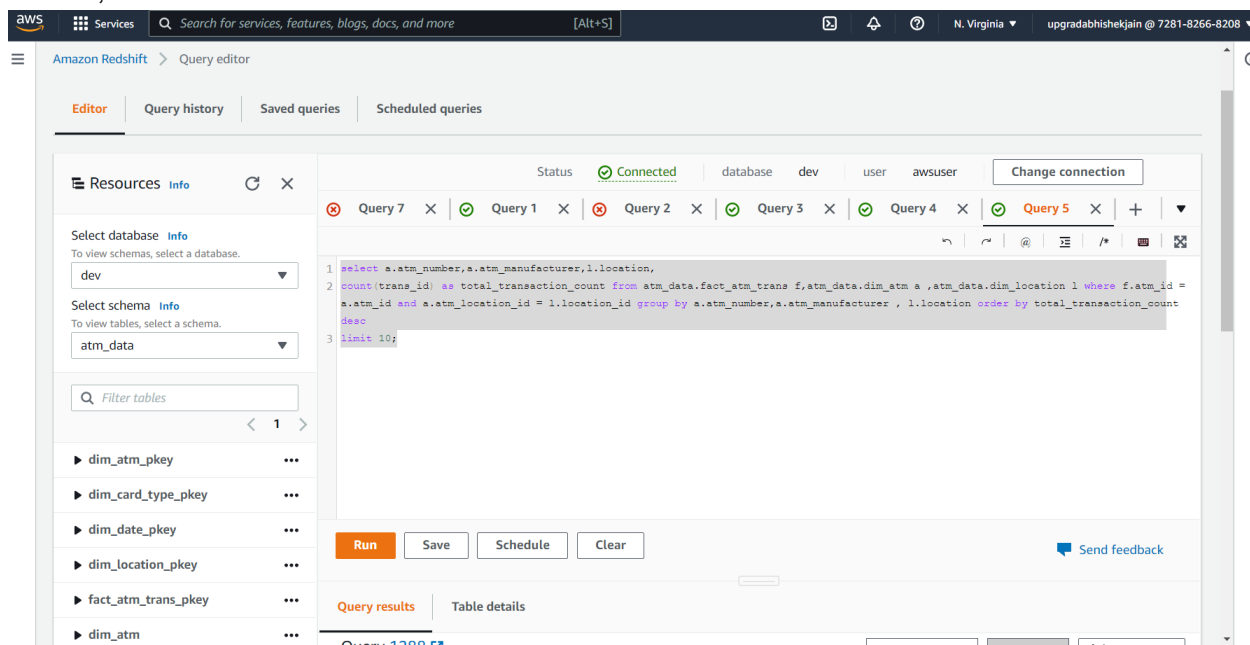
Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button has been clicked, and the 'Query results' tab is active. The results show that the query was completed successfully on February 01, 2022, at 02:28:43, with an elapsed time of 00 m 08 s. The query ID is 1239. The results are displayed in a table with 10 rows returned. There are also buttons for 'Execution', 'Data', 'Visualize', and 'Export'.



weather_main	total_transaction_count	total_inactive_count	inactive_count_percent
Snow	23405	4813	20.5640
Fog	18174	3729	20.5183
Clouds	1181901	194027	16.4165
Rain	545135	86017	15.7790
Clear	543949	85531	15.7241
Mist	82801	12864	15.5360
Thunderstorm	2549	361	14.1624
Drizzle	62530	8670	13.8653
TORNADO	38	1	2.6316
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;
```



Amazon Redshift Query Editor interface showing the SQL query and its execution status.

Resources:

- Select database: dev
- Select 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

Query Editor:

Status: Connected | database: dev | user: awsuser | Change connection

Query 7 | Query 1 | Query 2 | Query 3 | Query 4 | Query 5

```
1 select a.atm_number,a.atm_manufacturer,l.location,
2 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 =
3 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;
```

Run | Save | Schedule | Clear

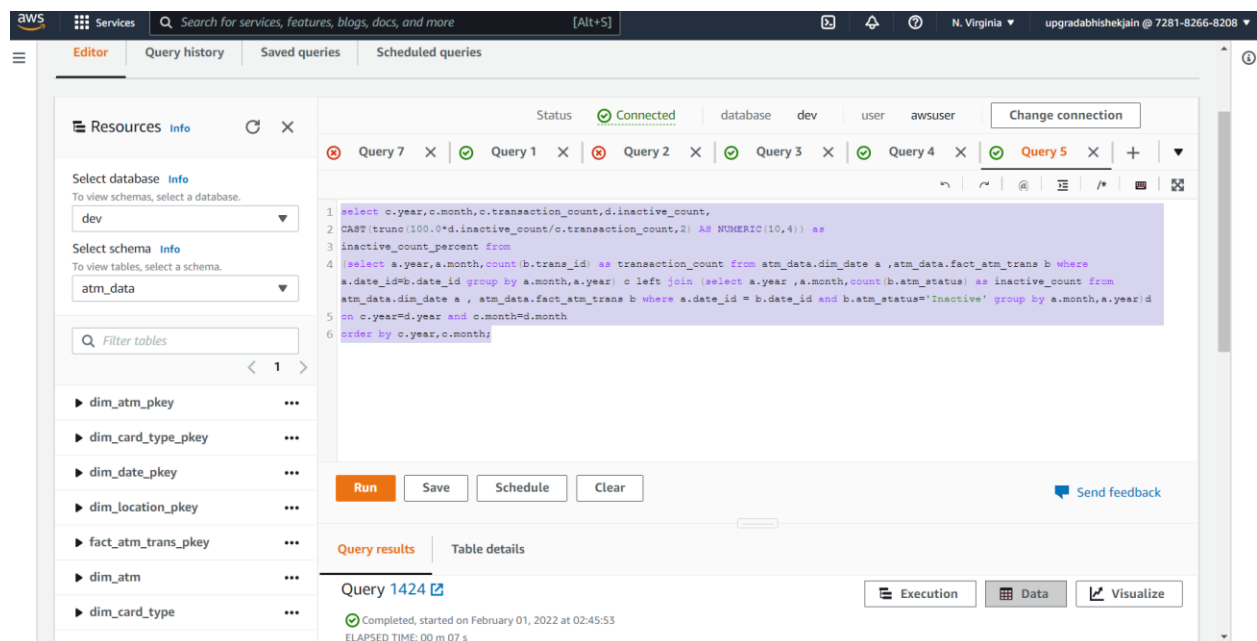
Query results | Table details

Query 1288.02

© Copyright. upGrad Education Pvt. Ltd. All rights reserved

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


```
select c.year,c.month,c.transaction_count,d.inactive_count,
CAST(trunc(100.0*d.inactive_count/c.transaction_count,2) AS NUMERIC(10,4)) as
inactive_count_percent from
(select a.year,a.month,count(b.trans_id) as transaction_count from atm_data.dim_date a
,atm_data.fact_atm_trans b where a.date_id=b.date_id group by a.month,a.year) c left join
(select a.year ,a.month,count(b.atm_status) as inactive_count from atm_data.dim_date a ,
atm_data.fact_atm_trans b where a.date_id = b.date_id and b.atm_status='Inactive' group by
a.month,a.year)d
on c.year=d.year and c.month=d.month
order by c.year,c.month;
```



The screenshot shows the AWS Data console interface. On the left, the 'Resources' panel is open, showing the 'dev' database and 'atm_data' schema. The main editor area contains a SQL query (Query 5) that calculates the percentage of inactive ATM transactions per month. The query is as follows:

```
1 select c.year,c.month,c.transaction_count,d.inactive_count,
2 CAST(trunc(100.0*d.inactive_count/c.transaction_count,2) AS NUMERIC(10,4)) as
3 inactive_count_percent from
4 (select a.year,a.month,count(b.trans_id) as transaction_count from atm_data.dim_date a ,atm_data.fact_atm_trans b where
5 a.date_id=b.date_id group by a.month,a.year) c left join (select a.year ,a.month,count(b.atm_status) as inactive_count from
6 atm_data.dim_date a , atm_data.fact_atm_trans b where a.date_id = b.date_id and b.atm_status='Inactive' group by a.month,a.year)d
on c.year=d.year and c.month=d.month
order by c.year,c.month;
```

Below the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button is highlighted. At the bottom, the 'Query results' tab is active, showing 'Query 1424' as 'Completed, started on February 01, 2022 at 02:45:53' with an 'ELAPSED TIME: 00 m 07 s'.



Services

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

[Alt+S]

N. Virginia

upgradabhishekjain @ 7281-8266-8208

dim_atm

dim_card_type

dim_date

dim_location

fact_atm_trans

Query 1424

Completed, started on February 01, 2022 at 02:45:53

ELAPSED TIME: 00 m 07 s

Execution

Data

Visualize

Rows returned (12)

Export

Search rows

<

1

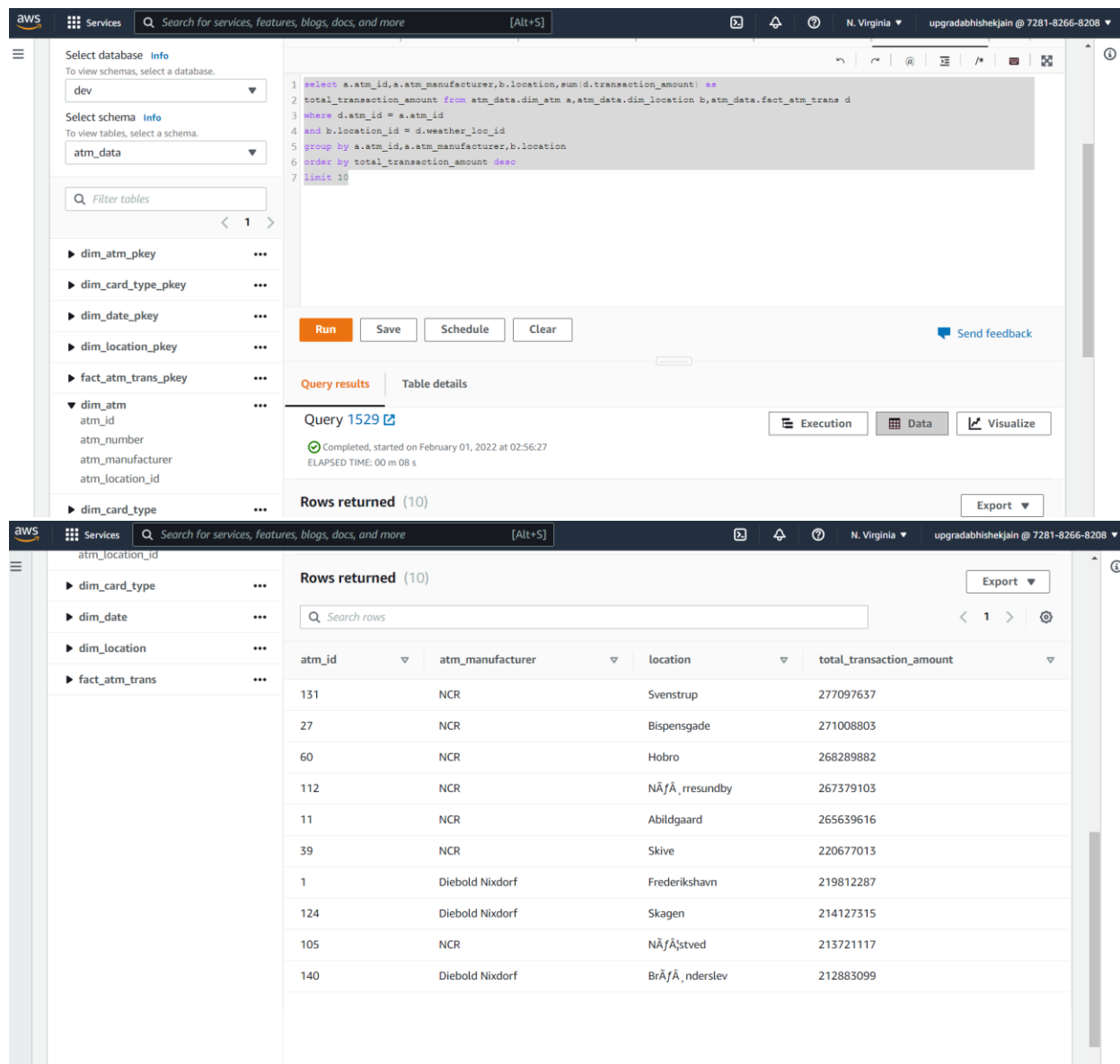
2

>

year	month	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

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

```
select a.atm_id,a.atm_manufacturer,b.location,sum(d.transaction_amount) as
total_transaction_amount from atm_data.dim_atm a,atm_data.dim_location
b,atm_data.fact_atm_trans d
where d.atm_id = a.atm_id
and b.location_id = d.weather_loc_id
group by a.atm_id,a.manufacturer,b.atm_location
order by total_transaction_amount desc
limit 10
```



The screenshot shows the AWS Glue console interface. On the left, the 'Select database' dropdown is set to 'dev' and the 'Select schema' dropdown is set to 'atm_data'. Below these, 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_location_id', 'dim_date', 'dim_location', and 'fact_atm_trans'. The 'dim_atm' table is expanded, showing columns: 'atm_id', 'atm_number', 'atm_manufacturer', and 'atm_location_id'.

The main area displays a SQL query in a text editor:

```
1 select a.atm_id,a.atm_manufacturer,b.location,sum(d.transaction_amount) as
2 total_transaction_amount from atm_data.dim_atm a,atm_data.dim_location
3 b,atm_data.fact_atm_trans d
4 where d.atm_id = a.atm_id
5 and b.location_id = d.weather_loc_id
6 group by a.atm_id,a.manufacturer,b.atm_location
7 order by total_transaction_amount desc
8 limit 10
```

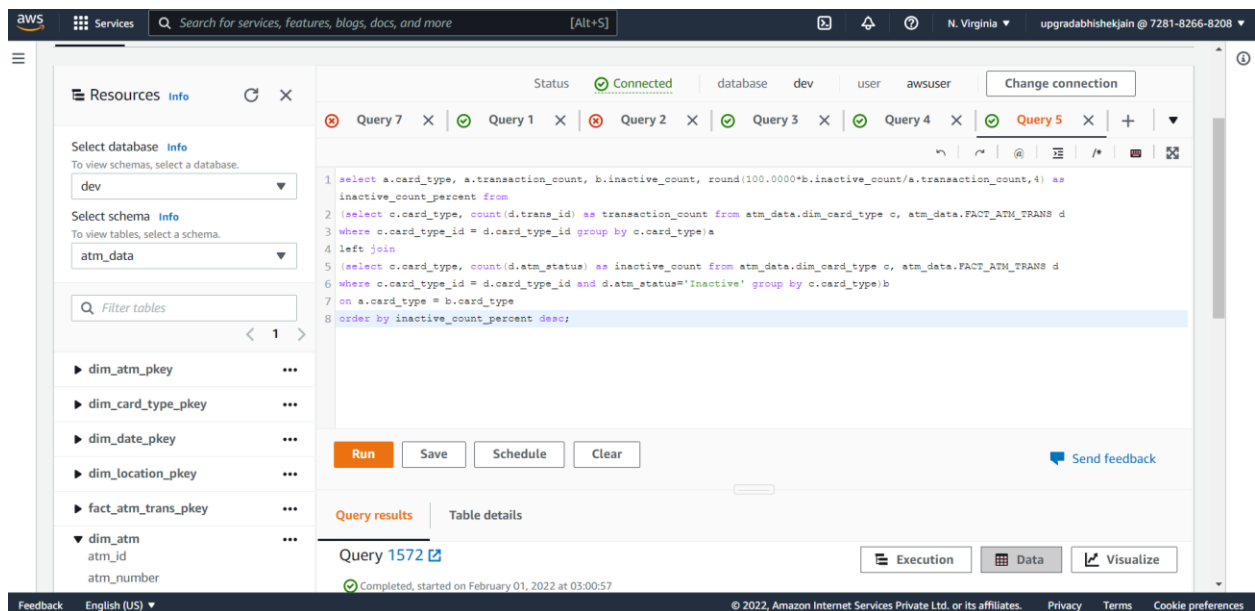
Below the query editor, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. A 'Send feedback' link is also present.

The 'Query results' tab is active, showing 'Query 1529' with a status of 'Completed, started on February 01, 2022 at 02:56:27' and 'ELAPSED TIME: 00 m 08 s'. The 'Rows returned' are 10. Below this, there are tabs for 'Execution', 'Data', and 'Visualize'. The 'Data' tab is selected, showing a table with 10 rows and 4 columns: 'atm_id', 'atm_manufacturer', 'location', and 'total_transaction_amount'.

atm_id	atm_manufacturer	location	total_transaction_amount
131	NCR	Svenstrup	277097637
27	NCR	Bispensgade	271008803
60	NCR	Hobro	268289882
112	NCR	NÅfÅ, rresundby	267379103
11	NCR	Abildgaard	265639616
39	NCR	Skive	220677013
1	Diebold Nixdorf	Frederikshavn	219812287
124	Diebold Nixdorf	Skagen	214127315
105	NCR	NÅfÅ'stved	213721117
140	Diebold Nixdorf	BrÅfÅ, nderslev	212883099

6. Number of failed ATM transactions across various card types

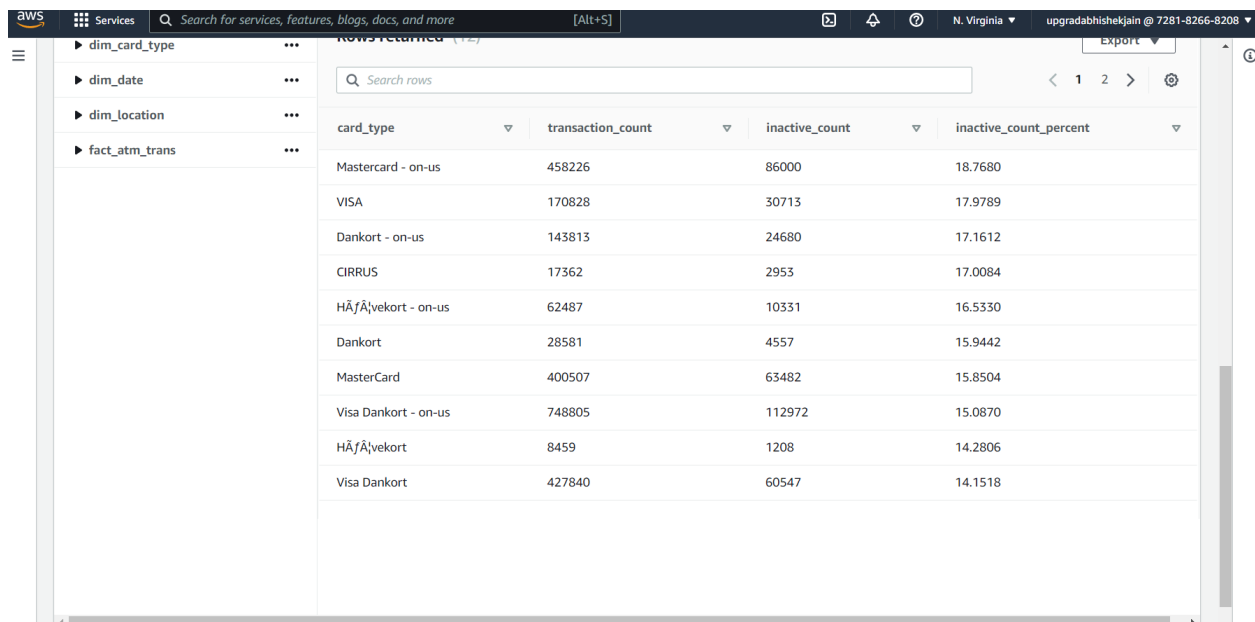
```
select a.card_type, a.transaction_count, b.inactive_count,
round(100.0000*b.inactive_count/a.transaction_count,4) as inactive_count_percent from
(select c.card_type, count(d.trans_id) as transaction_count from atm_data.dim_card_type c,
atm_data.FACT_ATM_TRANS d
where c.card_type_id = d.card_type_id group by c.card_type)a
left join
(select c.card_type, count(d.atm_status) as inactive_count from atm_data.dim_card_type c,
atm_data.FACT_ATM_TRANS d
where c.card_type_id = d.card_type_id and d.atm_status='Inactive' group by c.card_type)b
on a.card_type = b.card_type
order by inactive_count_percent desc;
```



The screenshot displays the AWS Glue console interface. On the left, there's a sidebar with 'Resources' and a list of tables including 'dim_atm_pkey', 'dim_card_type_pkey', 'dim_date_pkey', 'dim_location_pkey', 'fact_atm_trans_pkey', and 'dim_atm' (with sub-tables 'atm_id' and 'atm_number'). The main area shows a SQL query editor with the following query:

```
1 select a.card_type, a.transaction_count, b.inactive_count, round(100.0000*b.inactive_count/a.transaction_count,4) as
   inactive_count_percent from
2 (select c.card_type, count(d.trans_id) as transaction_count from atm_data.dim_card_type c, atm_data.FACT_ATM_TRANS d
3  where c.card_type_id = d.card_type_id group by c.card_type)a
4 left join
5 (select c.card_type, count(d.atm_status) as inactive_count from atm_data.dim_card_type c, atm_data.FACT_ATM_TRANS d
6  where c.card_type_id = d.card_type_id and d.atm_status='Inactive' group by c.card_type)b
7 on a.card_type = b.card_type
8 order by inactive_count_percent desc;
```

Below the query editor are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button is highlighted. Below these buttons, it says 'Query results' and 'Table details'. At the bottom, it indicates 'Query 1572' and 'Completed, started on February 01, 2022 at 03:00:57'. The footer shows '© 2022, Amazon Internet Services Private Ltd. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.



card_type	transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.7680
VISA	170828	30713	17.9789
Dankort - on-us	143813	24680	17.1612
CIRRUS	17362	2953	17.0084
HÃfÃ\vekort - on-us	62487	10331	16.5330
Dankort	28581	4557	15.9442
MasterCard	400507	63482	15.8504
Visa Dankort - on-us	748805	112972	15.0870
HÃfÃ\vekort	8459	1208	14.2806
Visa Dankort	427840	60547	14.1518

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, b.location,
case when c.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 d, atm_data.dim_atm a, atm_data.dim_location b,
atm_data.dim_date c
where d.atm_id = a.atm_id and a.atm_location_id = b.location_id and d.date_id
= c.date_id
group by a.atm_number, a.atm_manufacturer, b.location, weekend_flag
order by a.atm_number, a.atm_manufacturer, b.location, weekend_flag,
total_transaction_count
limit 10;

```

Resources

Select database

dev

Select schema

atm_data

Filter tables

dim_atm_pkey

dim_card_type_pkey

dim_date_pkey

dim_location_pkey

fact_atm_trans_pkey

dim_atm

atm_id

atm_number

atm_manufacturer

Status

Connected

database

dev

user

awsuser

Change connection

Query 7

Query 1

Query 2

Query 3

Query 4

Query 5

```

1 select a.atm_number, a.atm_manufacturer, b.location,
2 case when c.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 d, atm_data.dim_atm a, atm_data.dim_location b,
6 atm_data.dim_date c
7 where d.atm_id = a.atm_id and a.atm_location_id = b.location_id and d.date_id
8 = c.date_id
9 group by a.atm_number, a.atm_manufacturer, b.location, weekend_flag
10 order by a.atm_number, a.atm_manufacturer, b.location, weekend_flag,
11 total_transaction_count
12 limit 10;

```

Run

Save

Schedule

Clear

Send feedback

Query results

Table details

Query 1665

Execution

Data

Visualize

Completed, started on February 01, 2022 at 03:10:12

ELAPSED TIME: 00 m 02 s

atm_location_id

dim_card_type

dim_date

dim_location

fact_atm_trans

Rows returned (10)

Export

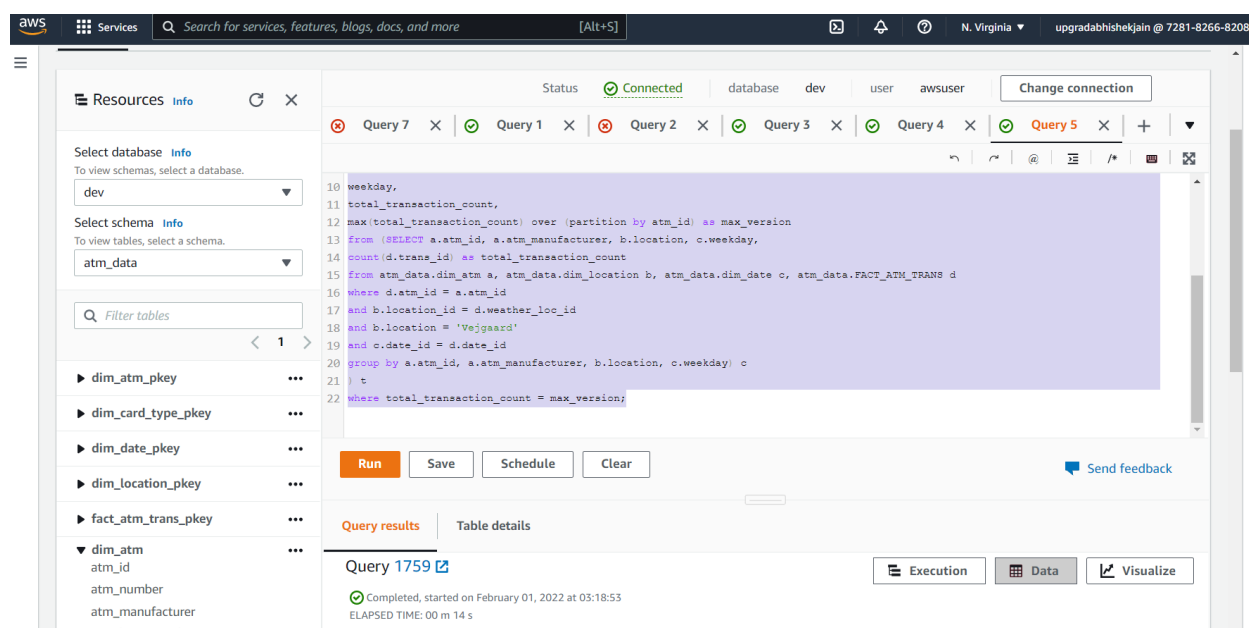
Search rows

1

atm_number	atm_manufacturer	location	weekend_flag	total_transaction_count
1	NCR	NÃfÃstved	0	25527
1	NCR	NÃfÃstved	1	8198
10	NCR	NÃfÃ, rresundby	0	41667
10	NCR	NÃfÃ, rresundby	1	12127
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
102	NCR	Aalborg Storcenter indg. D	0	5218
102	NCR	Aalborg Storcenter indg. D	1	1008

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

```
SELECT atm_id,
atm_manufacturer,
location,
weekday,
total_transaction_count
FROM (
select atm_id,
atm_manufacturer,
location,
weekday,
total_transaction_count,
max(total_transaction_count) over (partition by atm_id) as max_version
from (SELECT a.atm_id, a.atm_manufacturer, b.location, c.weekday,
count(d.trans_id) as total_transaction_count
from atm_data.dim_atm a, atm_data.dim_location b, atm_data.dim_date c,
atm_data.FACT_ATM_TRANS d
where d.atm_id = a.atm_id
and b.location_id = d.weather_loc_id
and b.location = 'Vejgaard'
and c.date_id = d.date_id
group by a.atm_id, a.atm_manufacturer, b.location, c.weekday) c
) t
where total_transaction_count = max_version;
```



The screenshot shows the AWS Glue console interface. On the left, the 'Resources' panel displays a list of tables under the 'atm_data' schema, including 'dim_atm_pkey', 'dim_card_type_pkey', 'dim_date_pkey', 'dim_location_pkey', 'fact_atm_trans_pkey', and 'dim_atm'. The main panel shows a SQL query being executed, which is the same query as provided in the text above. The query is labeled 'Query 1759'. Below the query, there are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. The 'Run' button is highlighted. At the bottom, the 'Query results' tab is active, showing the status 'Completed, started on February 01, 2022 at 03:18:53' and 'ELAPSED TIME: 00 m 14 s'.

© Copyright. upGrad Education Pvt. Ltd. All rights reserved