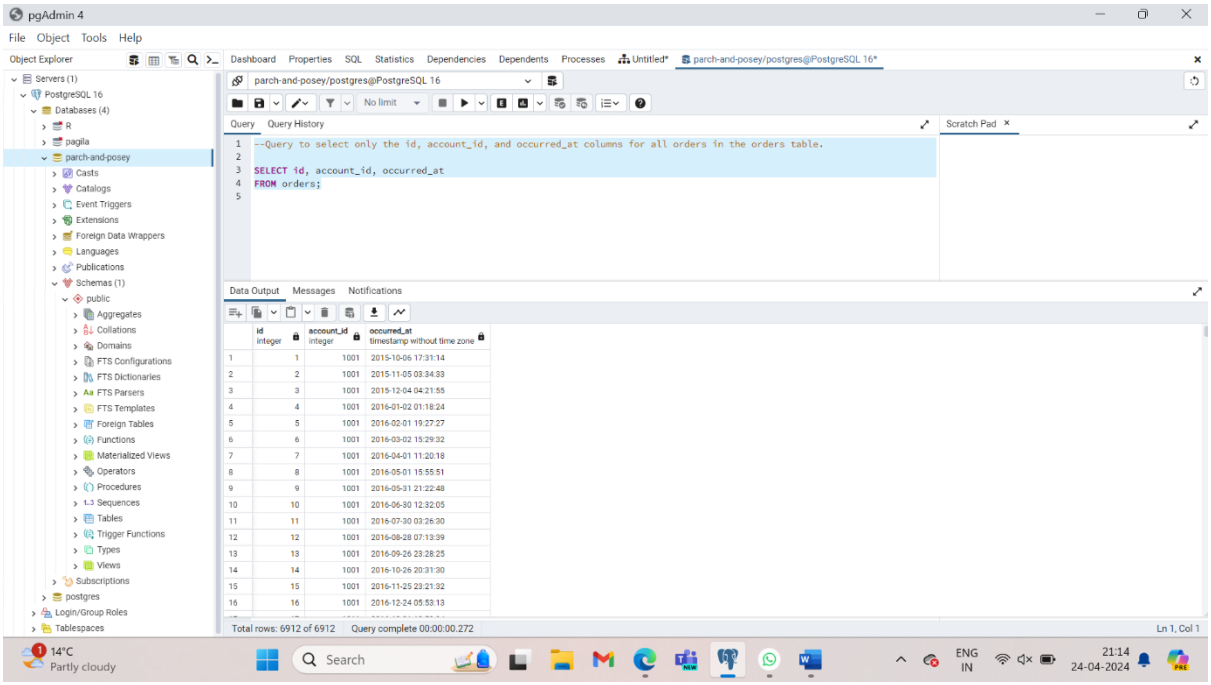
Basic SQL

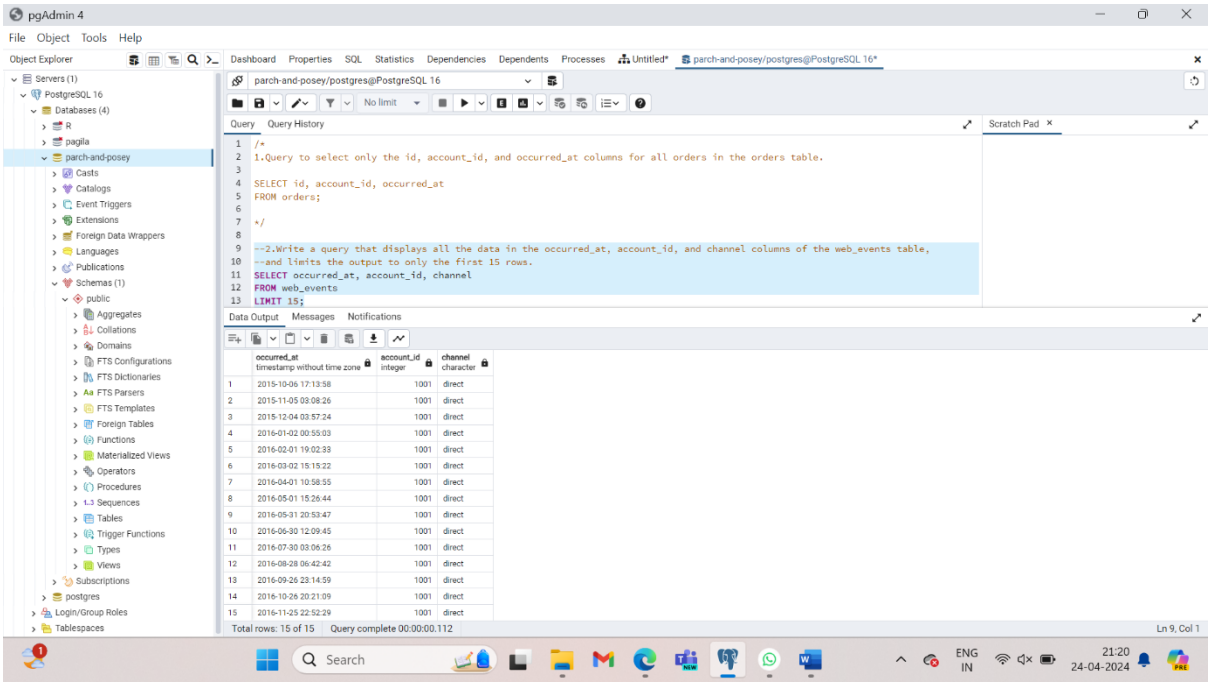
1. Query to select only the id, account_id, and occurred_at columns for all orders in the **orders** table.

```
SELECT id, account_id, occurred_at
FROM orders;
```



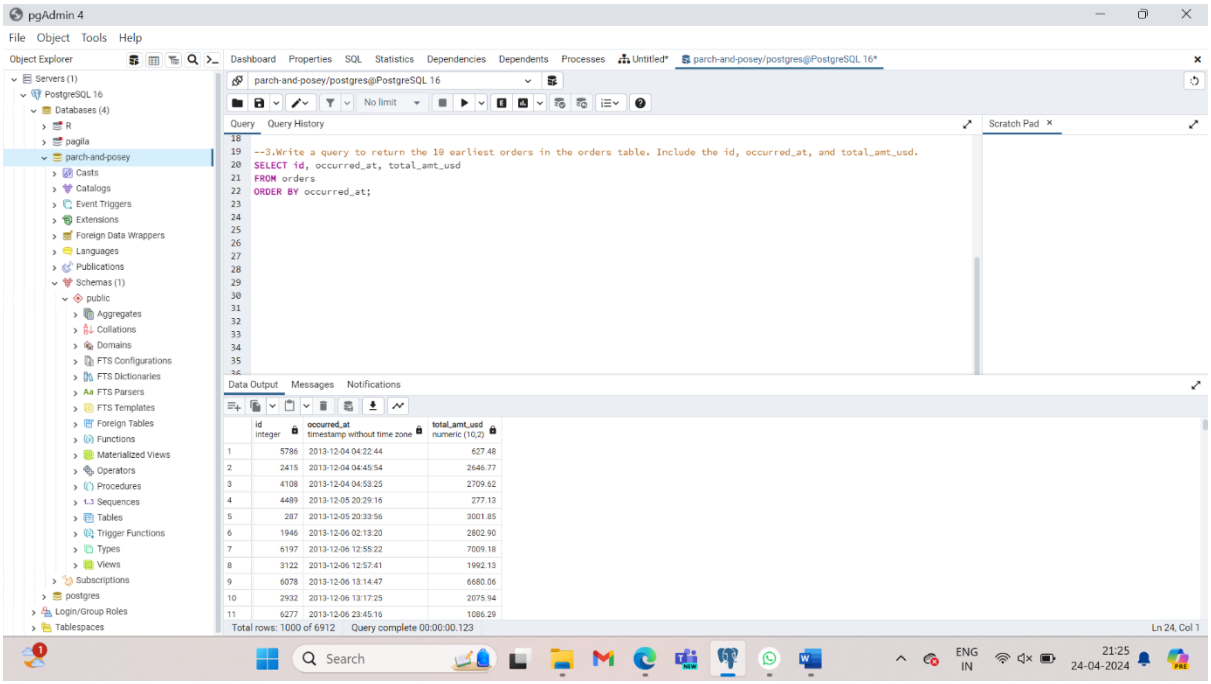
2. Write a query that displays all the data in the occurred_at, account_id, and channel columns of the web_events table, and limits the output to only the first 15 rows.

```
SELECT occurred_at, account_id, channel
FROM web_events
LIMIT 15;
```



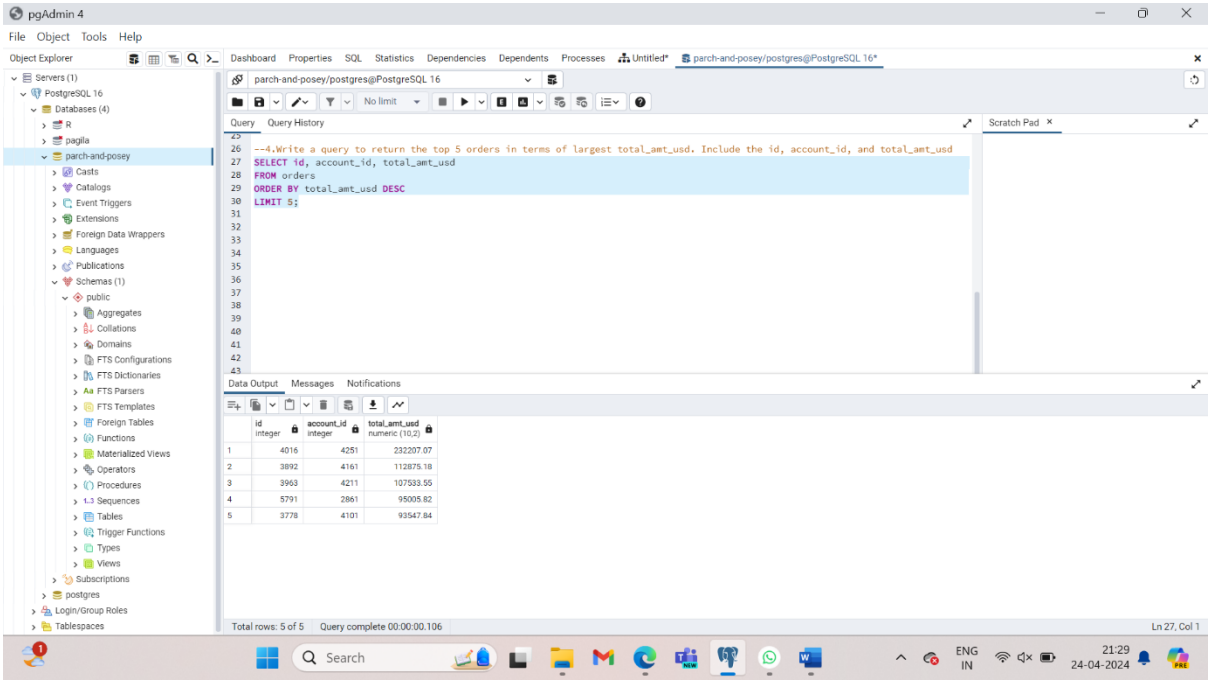
3. Write a query to return the 10 earliest orders in the **orders** table. Include the id, occurred_at, and total_amt_usd.

```
SELECT id, occurred_at, total_amt_usd
FROM orders
ORDER BY occurred_at;
```



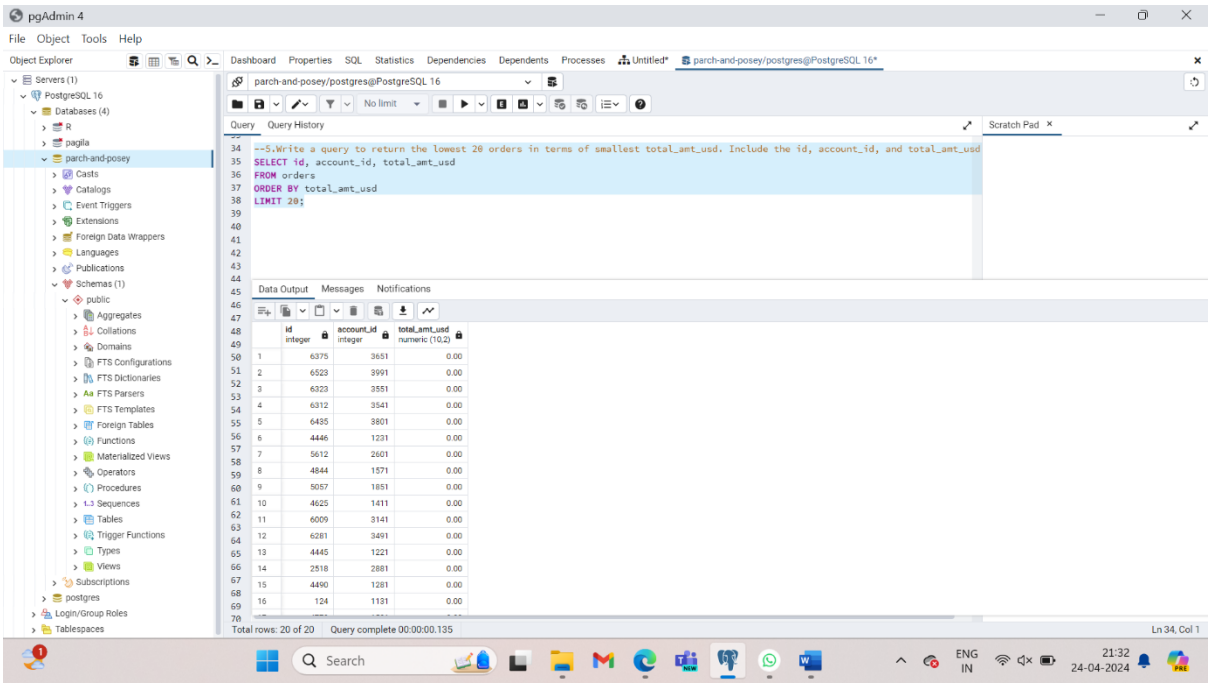
4. Write a query to return the top 5 **orders** in terms of largest total_amt_usd. Include the id, account_id, and total_amt_usd

```
SELECT id, account_id, total_amt_usd
FROM orders
ORDER BY total_amt_usd DESC
LIMIT 5;
```



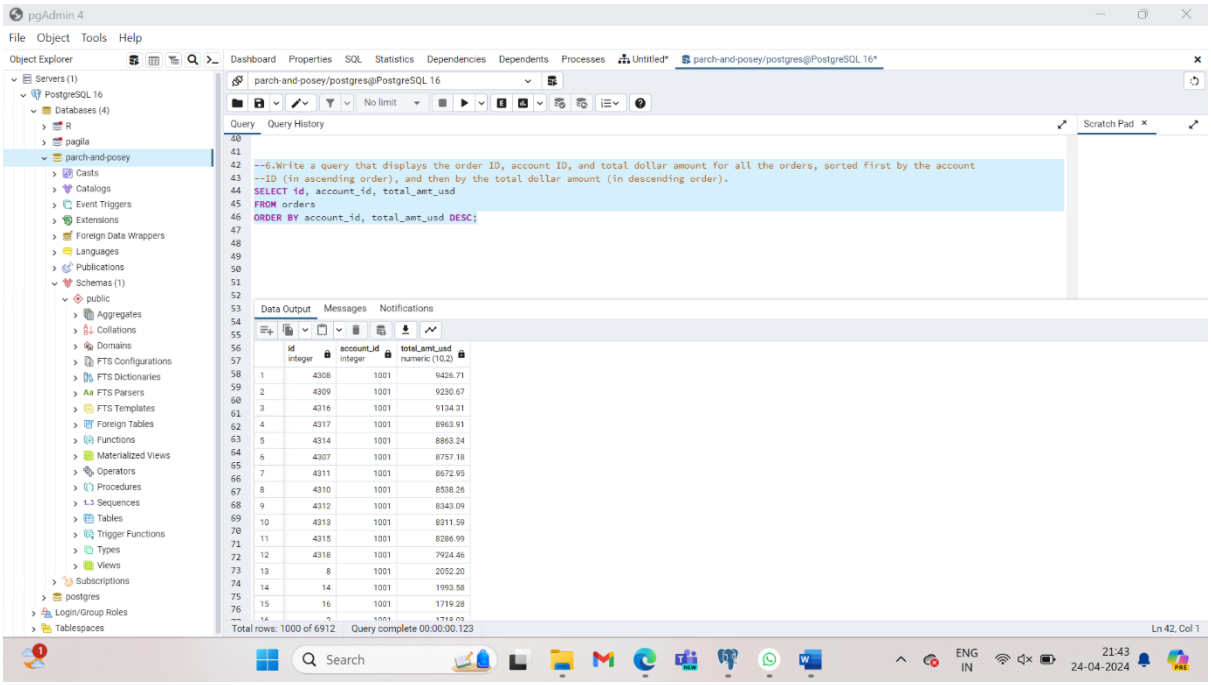
5. Write a query to return the lowest 20 **orders** in terms of smallest total_amt_usd. Include the id, account_id, and total_amt_usd.

```
SELECT id, account_id, total_amt_usd
FROM orders
ORDER BY total_amt_usd
LIMIT 20;
```

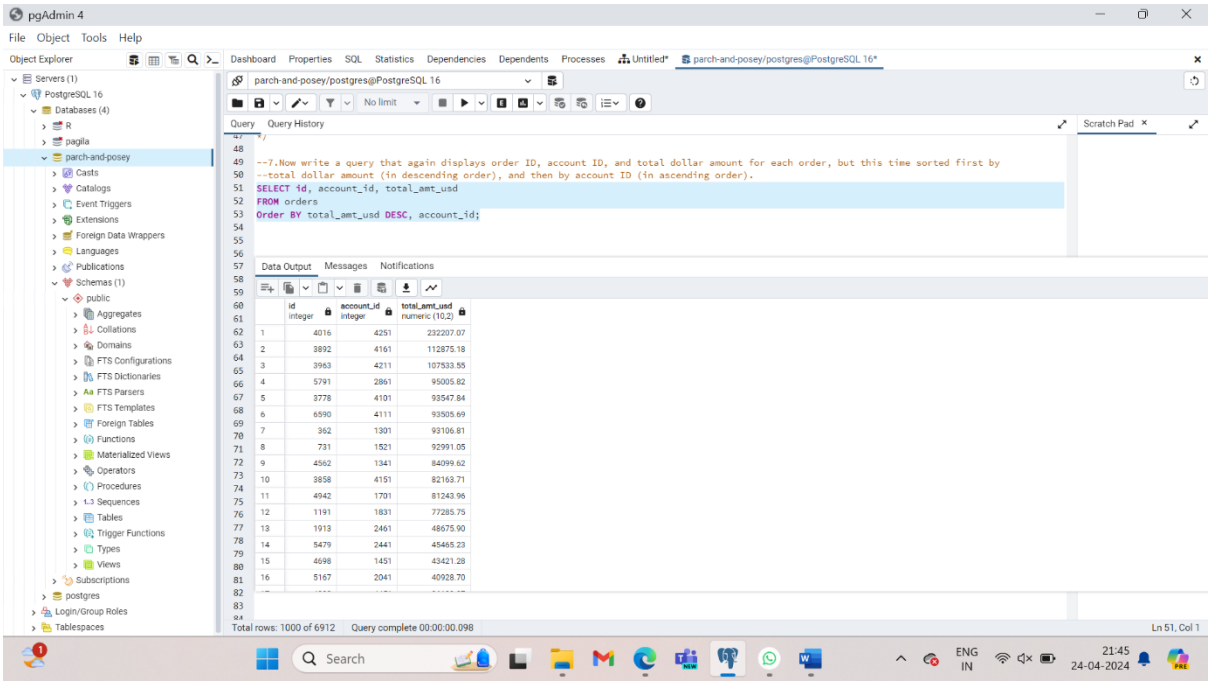


6. Write a query that displays the order ID, account ID, and total dollar amount for all the orders, sorted first by the account ID (in ascending order), and then by the total dollar amount (in descending order).

```
SELECT id, account_id, total_amt_usd
FROM orders
ORDER BY account_id, total_amt_usd DESC;
```

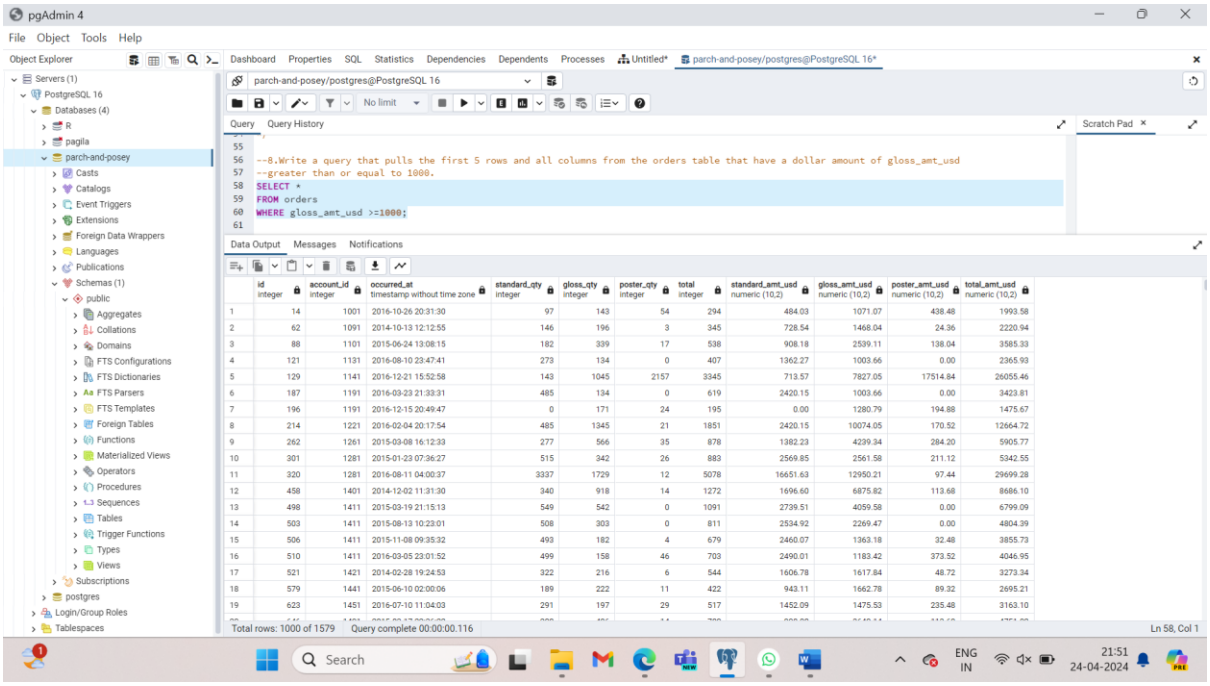


7. Now write a query that again displays order ID, account ID, and total dollar amount for each order, but this time sorted first by total dollar amount (in descending order), and then by account ID (in ascending order).
   SELECT id, account_id, total_amt_usd
   FROM orders
   Order BY total_amt_usd DESC, account_id;



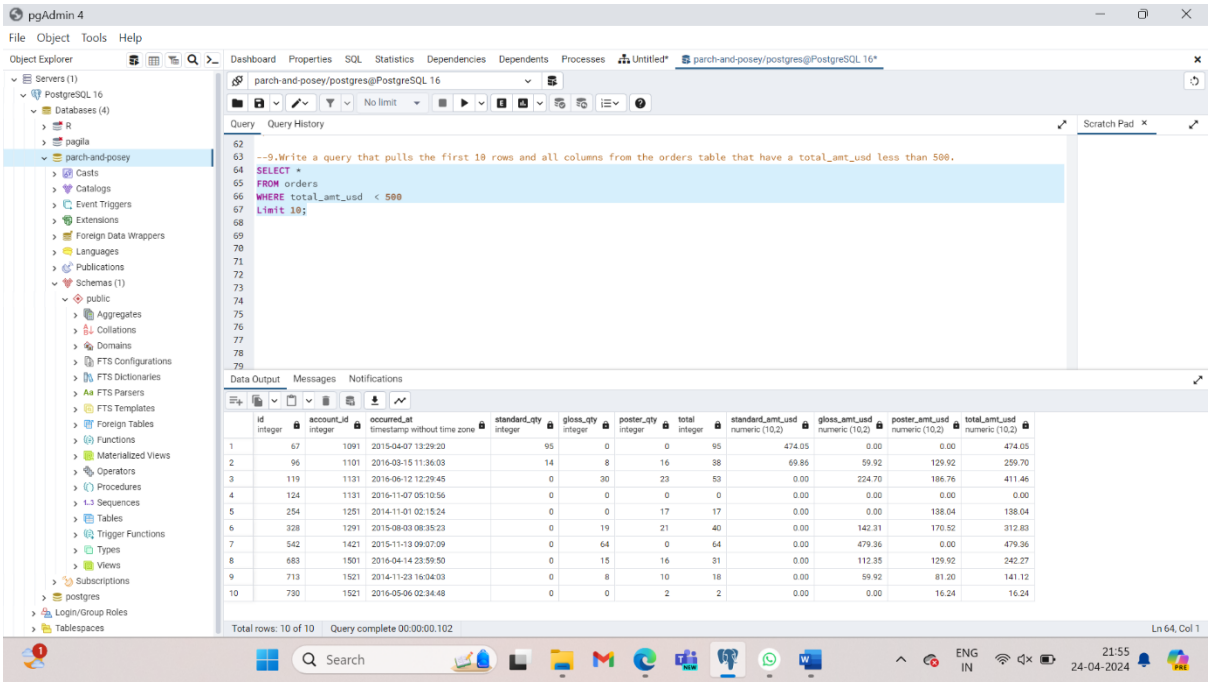
In query #6, all of the orders for each account ID are grouped together, and then within each of those groupings, the orders appear from the greatest order amount to the least. In query #7, since we sorted by the total dollar amount first, the orders appear from greatest to least regardless of which account ID they were from. Then they are sorted by account ID next. (The secondary sorting by account ID is difficult to see here, since only if there were two orders with equal total dollar amounts would there need to be any sorting by account ID.)

8. Write a query that pulls the first 5 rows and all columns from the **orders** table that have a dollar amount of gloss_amt_usd greater than or equal to 1000.
   SELECT *
   FROM orders
   WHERE gloss_amt_usd >=1000;



9. Write a query that pulls the first 10 rows and all columns from the **orders** table that have a total_amt_usd less than 500.

```
SELECT *
FROM orders
WHERE total_amt_usd  < 500
Limit 10;
```

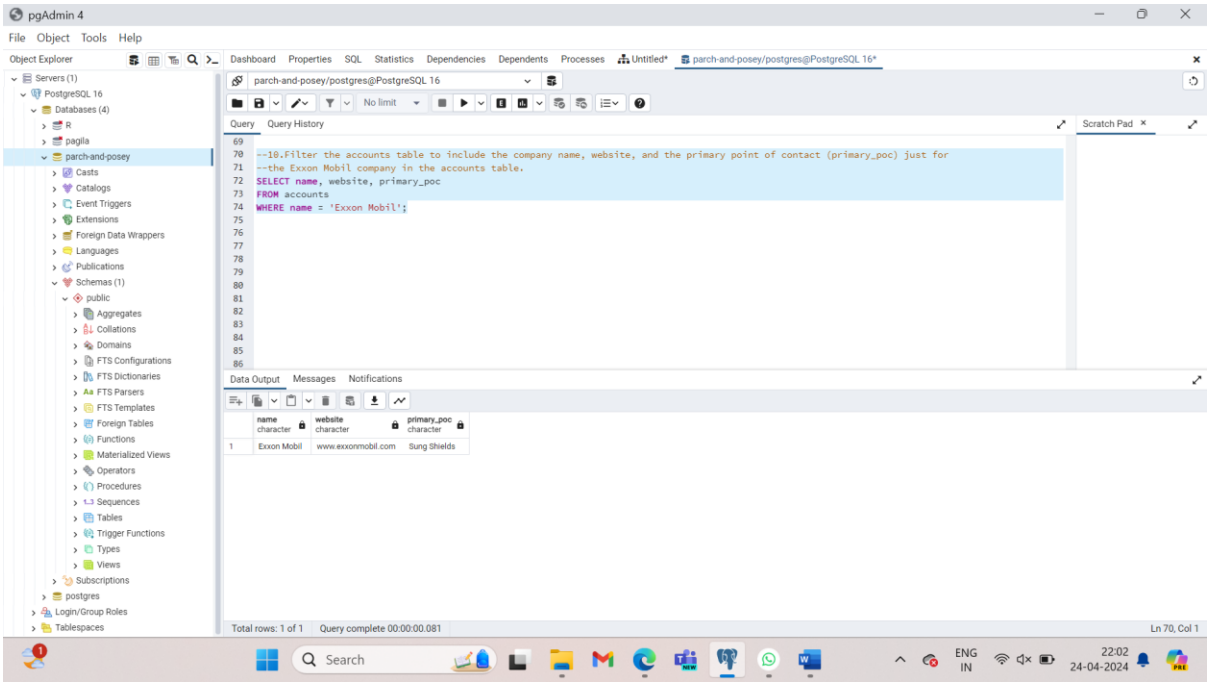


<u>When we are using **WHERE** with non-numeric data fields, we use the **LIKE**, **NOT**, or **IN** operators.</u>

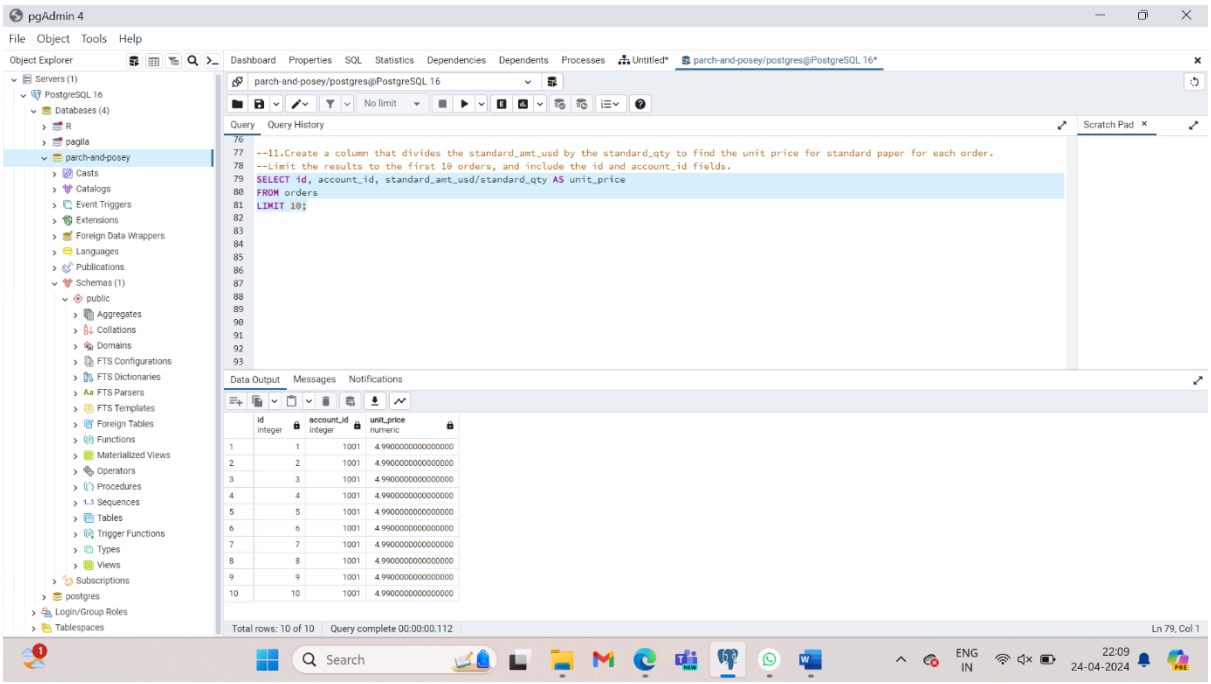
10. Filter the accounts table to include the company name, website, and the primary point of contact (primary_poc) just for the Exxon Mobil company in the **accounts** table.

```
SELECT name, website, primary_poc
```

FROM accounts
WHERE name = ‘Exxon Mobil’;

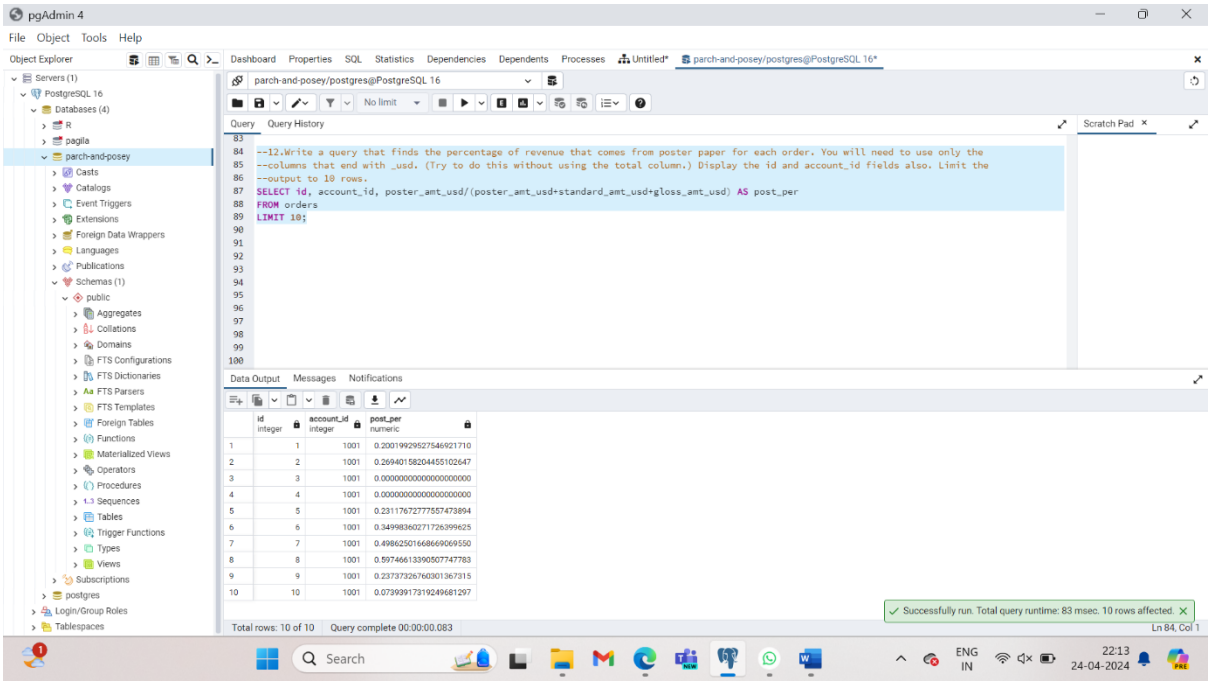


11. Create a column that divides the standard_amt_usd by the standard_qty to find the unit price for standard paper for each order. Limit the results to the first 10 orders, and include the id and account_id fields.
SELECT id, account_id, standard_amt_usd/standard_qty **AS** unit_price
FROM orders
LIMIT 10;



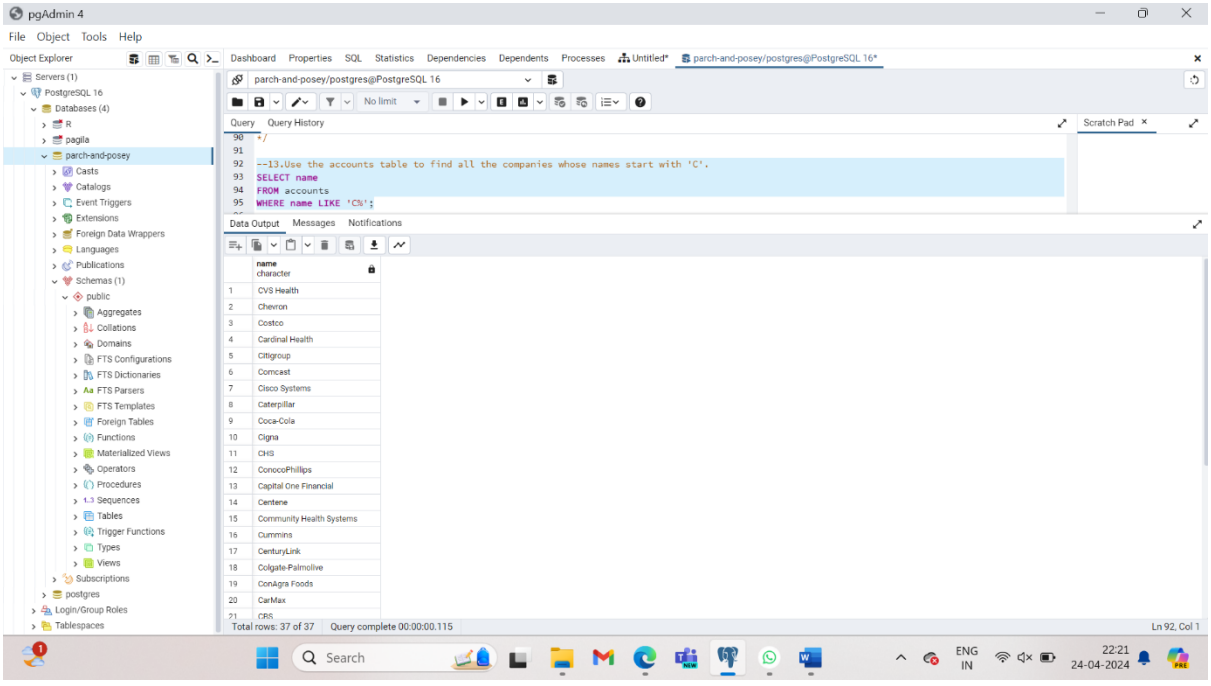
12. Write a query that finds the percentage of revenue that comes from poster paper for each order. You will need to use only the columns that end with _usd. (Try to do this without using the total column.) Display the id and account_id fields also. Limit the output to 10 rows.
SELECT id, account_id,
poster_amt_usd/(poster_amt_usd+standard_amt_usd+gloss_amt_usd) AS post_per

FROM orders
LIMIT 10;



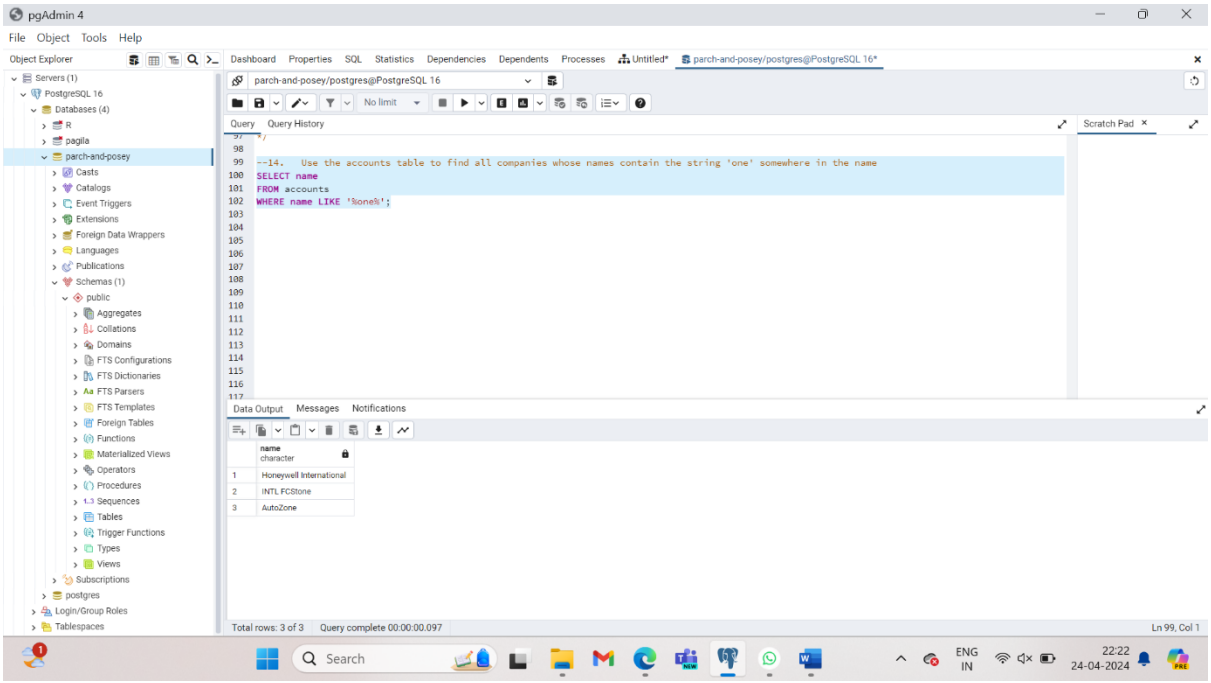
13. Use the **accounts** table to find all the companies whose names start with 'C'.

SELECT name
FROM accounts
WHERE name LIKE ‘C%’;

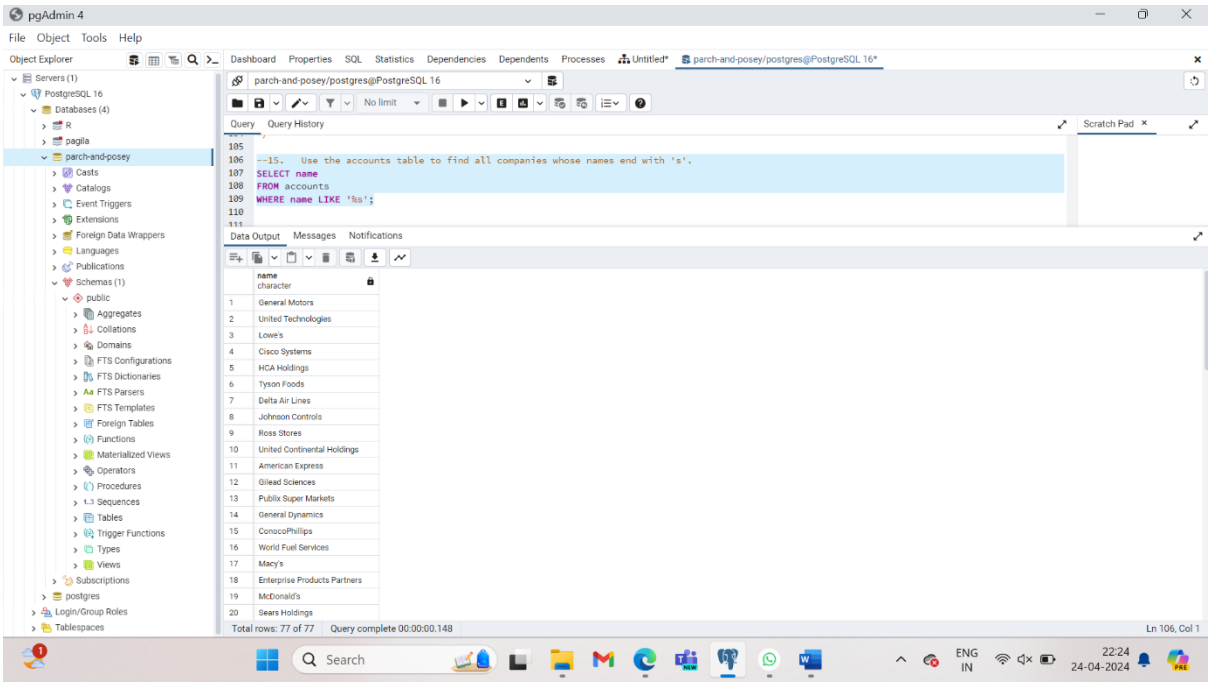


14. Use the **accounts** table to find all companies whose names contain the string 'one' somewhere in the name

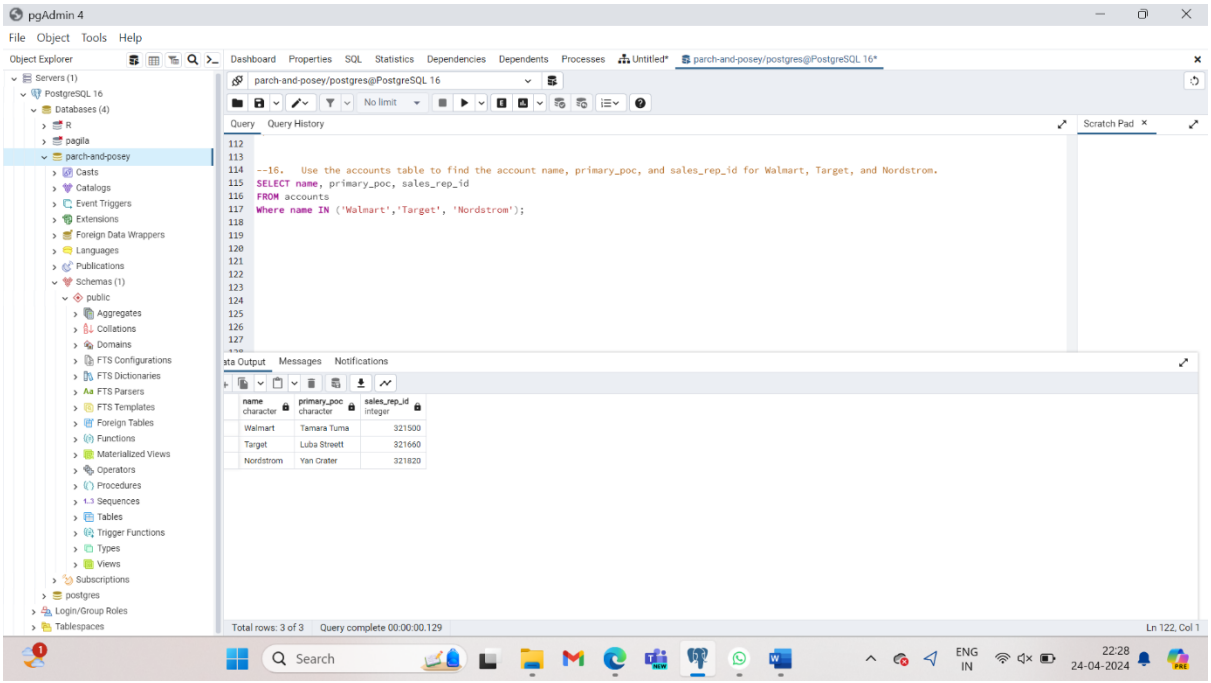
SELECT name
FROM accounts
WHERE name LIKE ‘%one%’;



15. Use the **accounts** table to find all companies whose names end with 's'.
    SELECT name
    FROM accounts
    WHERE name LIKE '%s';

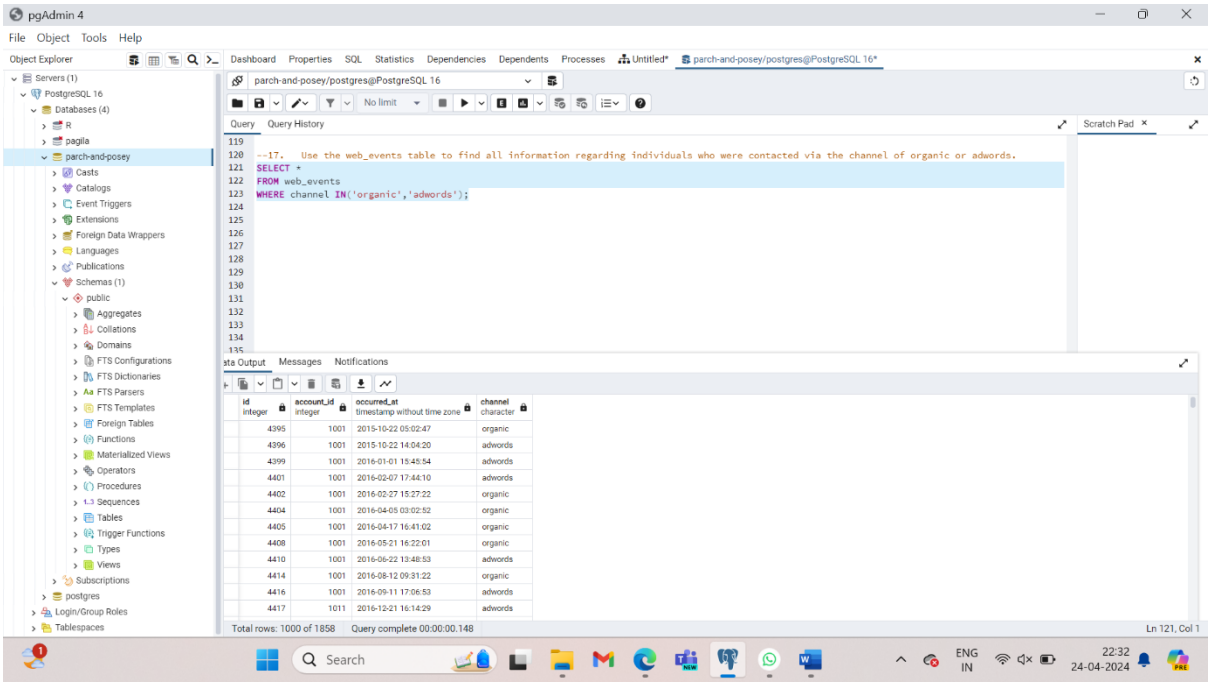


16. Use the **accounts** table to find the account name, primary_poc, and sales_rep_id for Walmart, Target, and Nordstrom.
    SELECT name, primary_poc, sales_rep_id
    FROM accounts
    Where name IN ('Walmart','Target', 'Nordstrom');



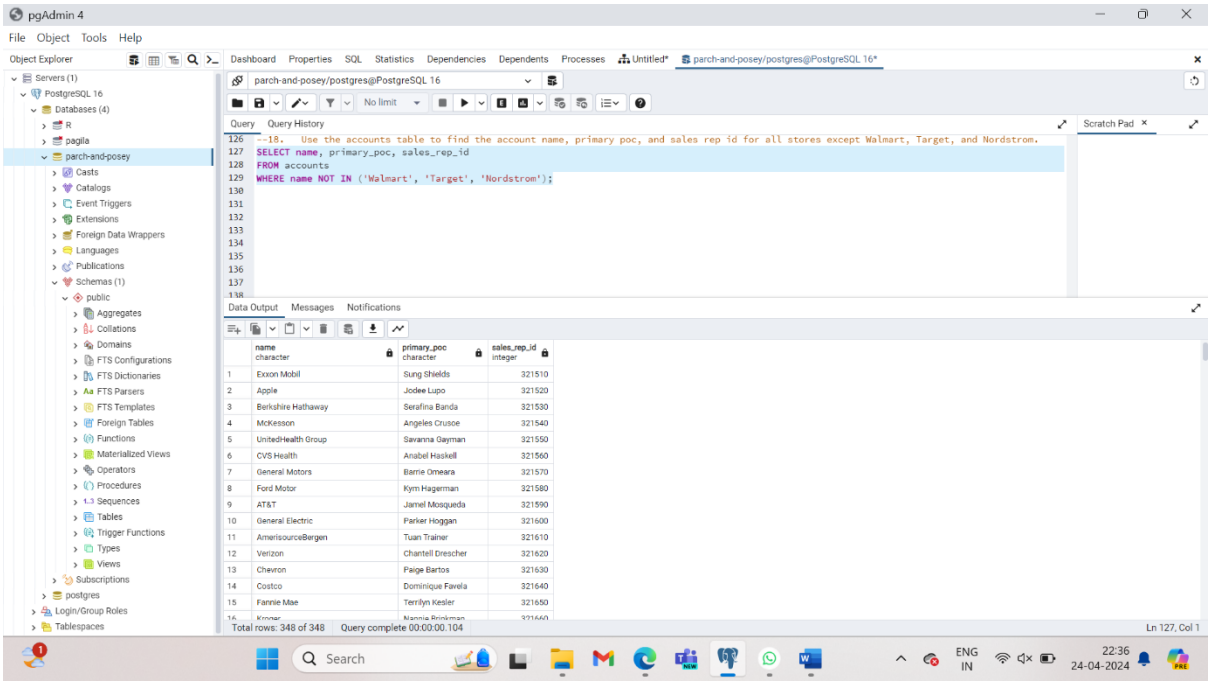
17. Use the **web_events** table to find all information regarding individuals who were contacted via the **channel** of organic or adwords.

```
SELECT *
FROM web_events
WHERE channel IN('organic','adwords');
```



18. Use the **accounts** table to find the account name, primary poc, and sales rep id for all stores except Walmart, Target, and Nordstrom.

```
SELECT name, primary_poc, sales_rep_id
FROM accounts
WHERE name NOT IN ('Walmart', 'Target', 'Nordstrom');
```



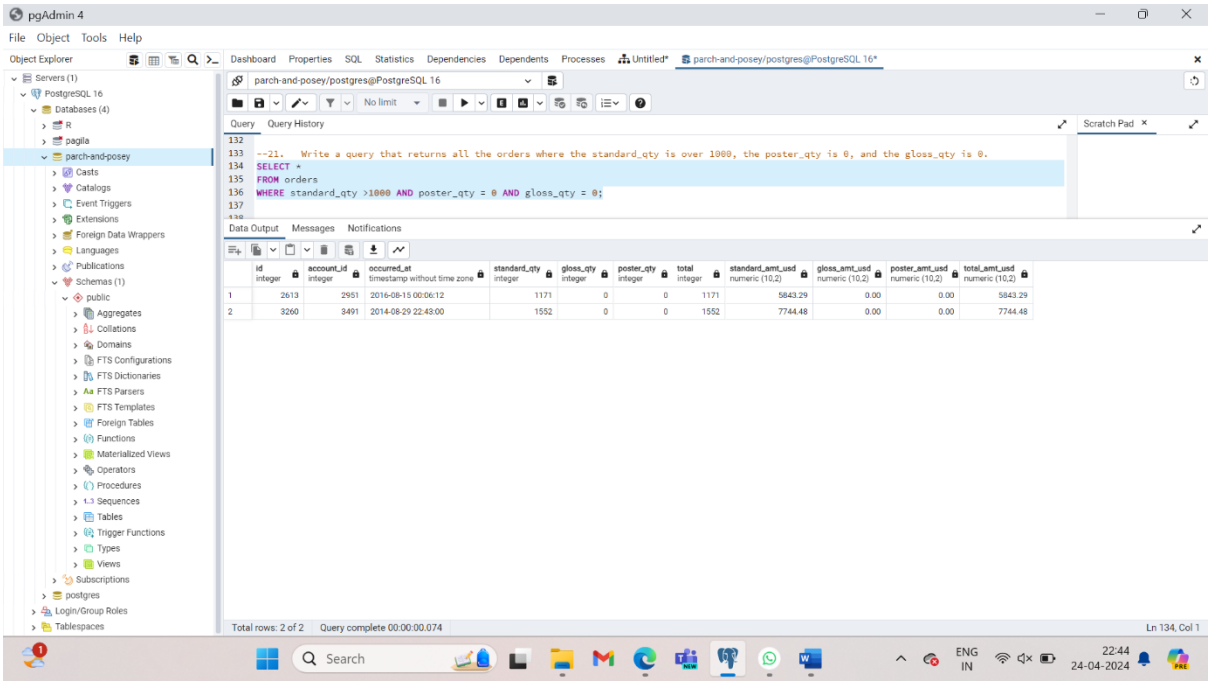
19. Use the **web_events** table to find all information regarding individuals who were contacted via any method except using organic or adwords methods.
    SELECT *
    FROM web_events
    WHERE channel NOT IN (‘organic’, ‘adwords’);

20. Use the **accounts** table to find all the companies whose names do not start with 'C'; all companies whose names do not contain the string 'one' somewhere in the name; all companies whose names do not end with 's'.
    SELECT name
    FROM accounts
    WHERE name NOT LIKE ‘C%’;

    SELECT name
    FROM accounts
    WHERE name NOT LIKE ‘%one%’;

    SELECT name
    FROM accounts
    WHERE name NOT LIKE ‘%s’;

21. Write a query that returns all the **orders** where the standard_qty is over 1000, the poster_qty is 0, and the gloss_qty is 0.
    SELECT *
    FROM orders
    WHERE standard_qty >1000 AND poster_qty = 0 AND ploss_qty = 0;

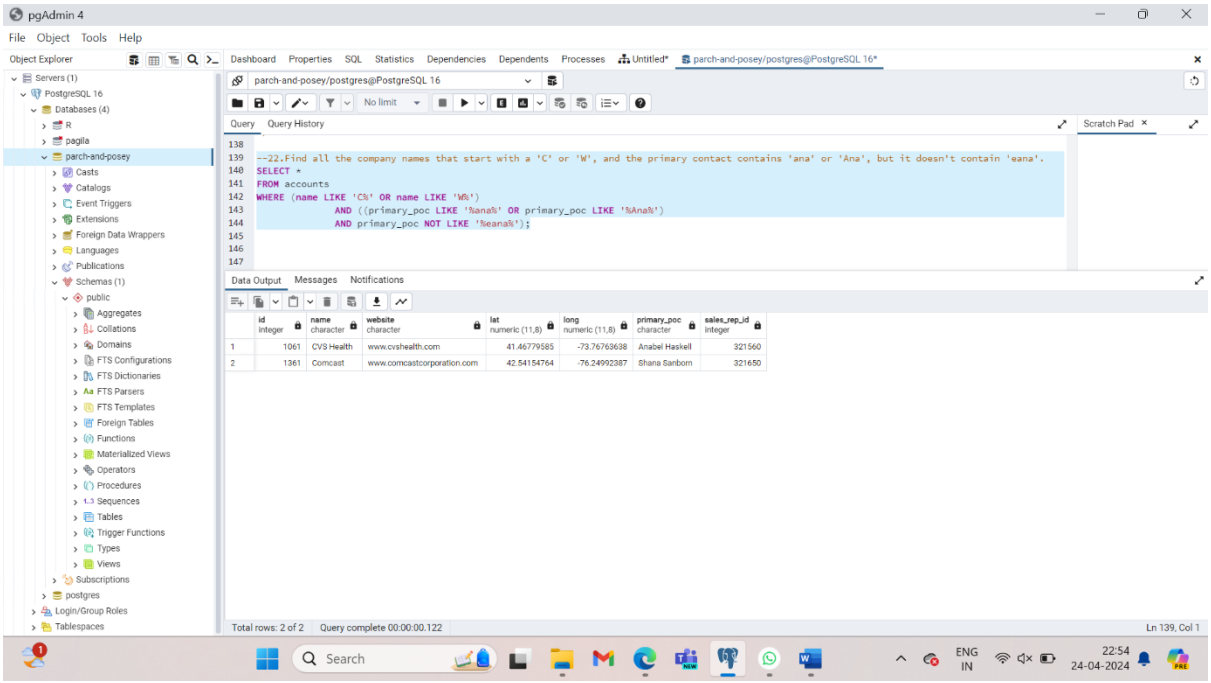


22. Use the **web_events** table to find all information regarding individuals who were contacted via the organic or adwords channels, and started their account at any point in 2016, sorted from newest to oldest.
    SELECT *
    FROM web_events
    WHERE channel IN('organic', 'adwords') AND occurred_at BETWEEN '2016-01-01' AND '2017-01-01'
    ORDER BY occurred_at DESC;

23. Find list of **orders** ids where either gloss_qty or poster_qty is greater than 4000. Only include the id field in the resulting table.
    SELECT id
    FROM orders
    WHERE gloss_qty > 4000 or poster_qty > 4000;

24. Write a query that returns a list of **orders** where the standard_qty is zero and either the gloss_qty or poster_qty is over 1000.
    SELECT *
    FROM orders
    WHERE standard_qty =0 AND (gloss_qty >1000 OR poster_qty >1000);

25. Find all the company names that start with a 'C' or 'W', and the primary contact **contains** 'ana' or 'Ana', but it doesn't contain 'eana'.
    SELECT *
    FROM accounts
    WHERE (name LIKE 'C%' OR name LIKE 'W%')
    AND ((primary_poc LIKE '%ana%' OR primary_poc LIKE '%Ana%')
    AND primary_poc NOT LIKE '%eana%');


pgAdmin 4
File Object Tools Help
Object Explorer
Dashboard Properties SQL Statistics Dependencies Dependents Processes Untitled* parch-and-posey/postgres@PostgreSQL 16*
Servers (1)
PostgreSQL 16
Databases (4)
R
pagila
parch-and-posey
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables
Trigger Functions
Types
Views
Subscriptions
postgres
Login/Group Roles
Tablespaces
parch-and-posey/postgres@PostgreSQL 16
No limit
Query Query History
Scratch Pad
138
139 --22.Find all the company names that start with a 'C' or 'W', and the primary contact contains 'ana' or 'Ana', but it doesn't contain 'eana'.
140 SELECT *
141 FROM accounts
142 WHERE (name LIKE 'C%' OR name LIKE 'W%')
143 AND ((primary_poc LIKE '%ana%' OR primary_poc LIKE '%Ana%')
144 AND primary_poc NOT LIKE '%eana%');
145
146
147
Data Output Messages Notifications
id integer
name character
website character
lat numeric (11,8)
long numeric (11,8)
primary_poc character
sales_rep_id integer
1 1061 CVS Health www.cvshealth.com 41.46779585 -73.76763638 Anabel Haskell 321560
2 1361 Comcast www.comcastcorporation.com 42.54154764 -76.24992387 Shana Sanborn 321650
Total rows: 2 of 2
Query complete 00:00:00.122
Ln 139, Col 1
Search
ENG
IN
22:54
24-04-2024