EXPLORING THE PARCH AND POSEY DATABASE (SQL ANALYSIS)

Data Analyst: Odudu Ochuko Richie

INTRODUCTION

In this analysis, I will be exploring the Parch and Posey database. I will carry out an analysis of this database to get insights that will be beneficial to the growth of the Parch and Posey company.

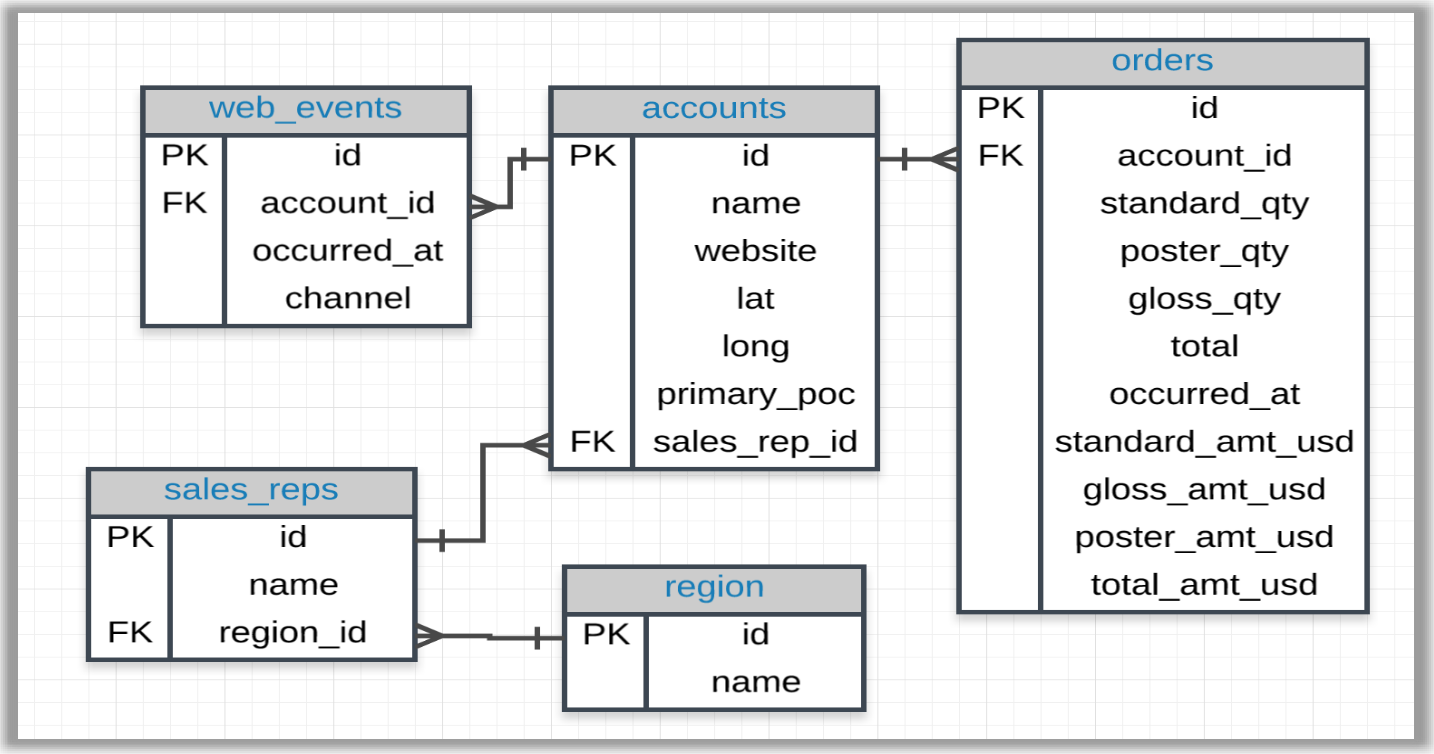
Firstly, Parch and Posey is a hypothetical company that sells paper. They have 50 sales reps spread across 4 regions in America. They sell 3 types of paper.

1. Standard
2. Gloss
3. Poster

Their customers are large companies they attract by advertising on Google, Facebook, and Twitter. In the Parch & Posey database, there are five tables web_events, accounts, orders, sales_reps, and region. The analysis will help understand the progress growth of Parch and Posey. In this analysis I will be looking at:

1. The account that placed the most orders
2. The account that placed the least orders
3. The account that placed the earliest order
4. The account that placed the latest order
5. Rank the total amount of paper ordered (from highest to lowest) for each account using a partition.
6. What kind of paper was ordered the most
7. Find the mean (average) amount spent per order on each paper type, as well as the mean amount of each paper type purchased per order.
8. The region for each sales rep along with their associated accounts. Sort the accounts alphabetically (A-Z) according to the account name.
9. Which year did Parch and Posey have the greatest sales in terms of total number of orders
10. Which month did Parch and Posey have the greatest sales in terms of total number of orders
11. The names of the sales reps in each region with the largest amount of sales (USD)
12. The region with the largest amount of total sales (USD), and how many total orders were placed?
13. The region with the smallest amount of total sales (USD), and how many total orders were placed?
14. The customer that spent the most (in total over their lifetime as a customer) in terms of total amount (USD). How many web events did they have for each channel?
15. Provide the name of the sales rep in each region with the largest amount of total amount (USD).
16. The account that purchased the most (in total over their lifetime as a customer) in terms of standard quantity paper, how many accounts still had more in total?

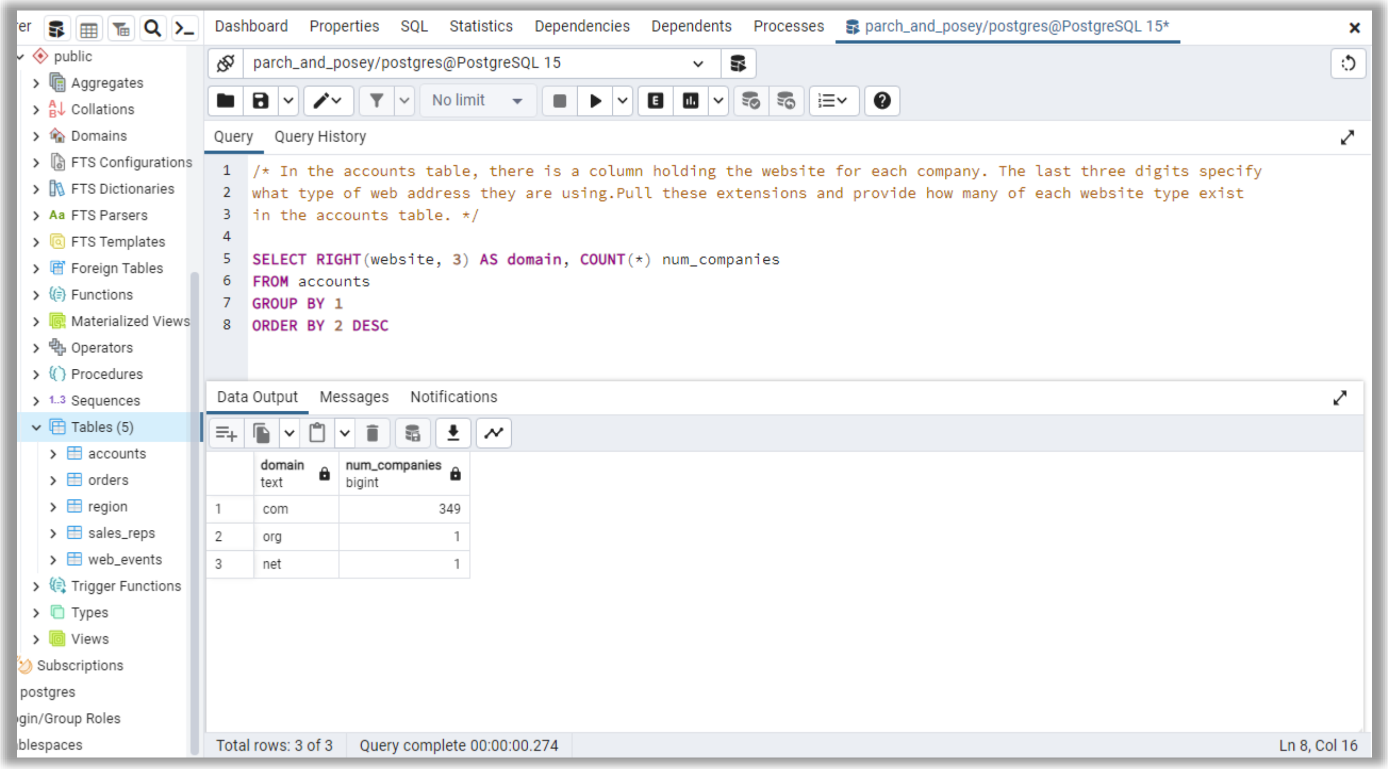
THE PARCH AND POSEY ERD DIAGRAM



DATA CLEANING

1. Clean and restructure messy data
2. Convert columns to different data types.

A. In the **accounts** table, there is a column holding the **website** for each company. The last three digits specify what type of web address they are using. Pull these extensions and provide how many of each website type exist in the **accounts** table.

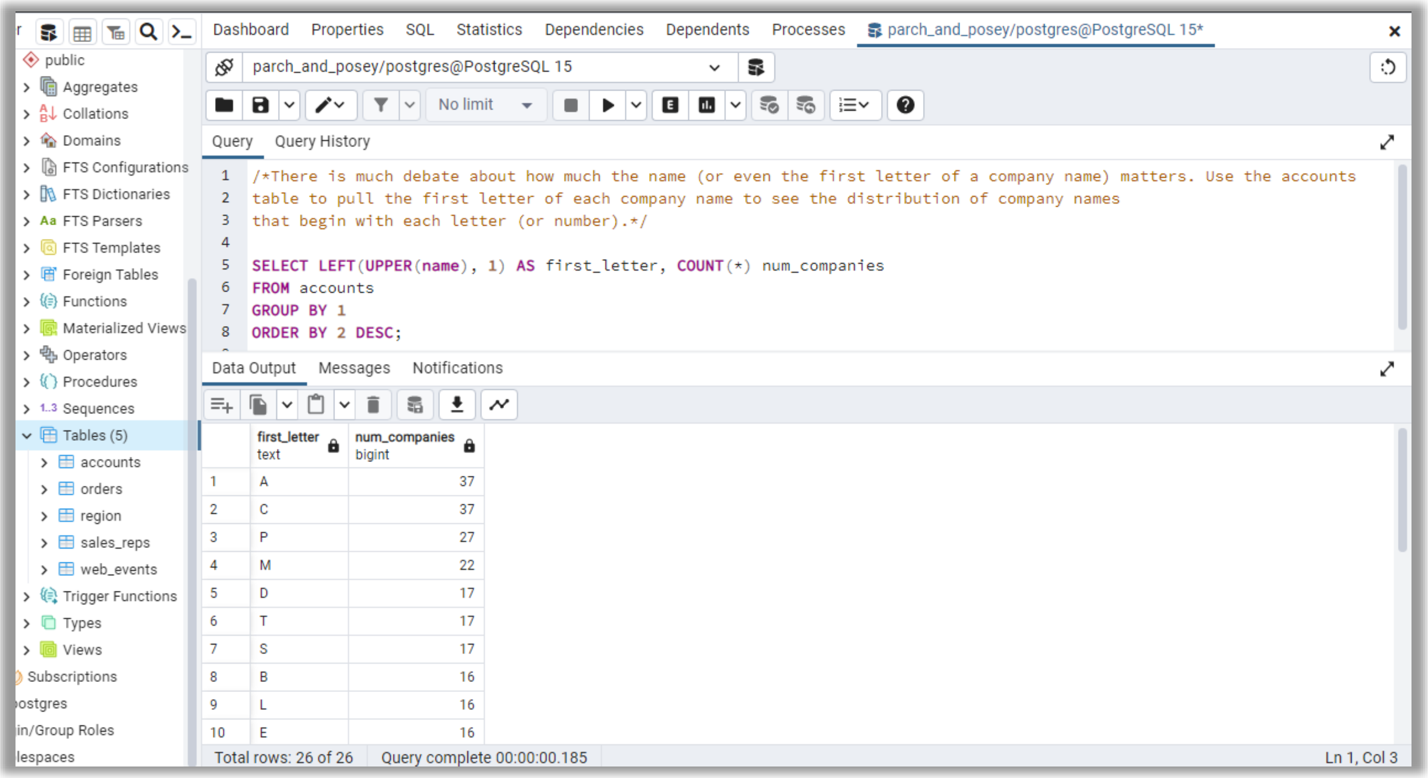


```
/* In the accounts table, there is a column holding the website for each company. The last three digits specify
what type of web address they are using.Pull these extensions and provide how many of each website type exist
in the accounts table. */

SELECT RIGHT(website, 3) AS domain, COUNT(*) num_companies
FROM accounts
GROUP BY 1
ORDER BY 2 DESC
```

	domain text	num_companies bigint
1	com	349
2	org	1
3	net	1

Total rows: 3 of 3   Query complete 00:00:00.274   Ln 8, Col 16

B. There is much debate about how much the name (or even the first letter of a company name) matters. Use the **accounts** table to pull the first letter of each company name to see the distribution of company names that begin with each letter (or number).

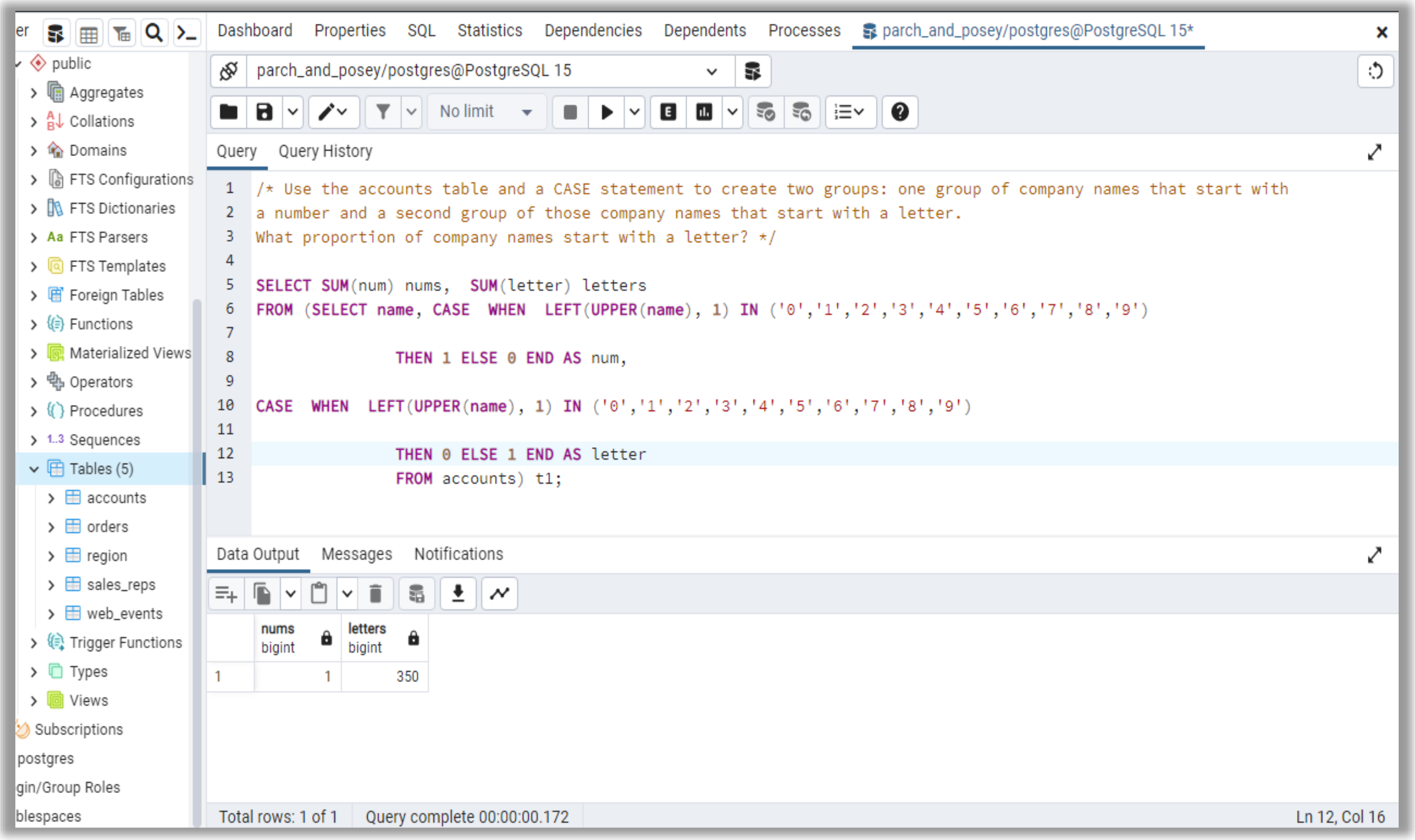


```
/*There is much debate about how much the name (or even the first letter of a company name) matters. Use the accounts
table to pull the first letter of each company name to see the distribution of company names
that begin with each letter (or number).*/

SELECT LEFT(UPPER(name), 1) AS first_letter, COUNT(*) num_companies
FROM accounts
GROUP BY 1
ORDER BY 2 DESC;
```

	first_letter text	num_companies bigint
1	A	37
2	C	37
3	P	27
4	M	22
5	D	17
6	T	17
7	S	17
8	B	16
9	L	16
10	E	16

Total rows: 26 of 26 Query complete 00:00:00.185 Ln 1, Col 3

C. Use the **accounts** table and a **CASE** statement to create two groups: one group of company names that start with a number and a second group of those company names that start with a letter. What proportion of company names start with a letter?

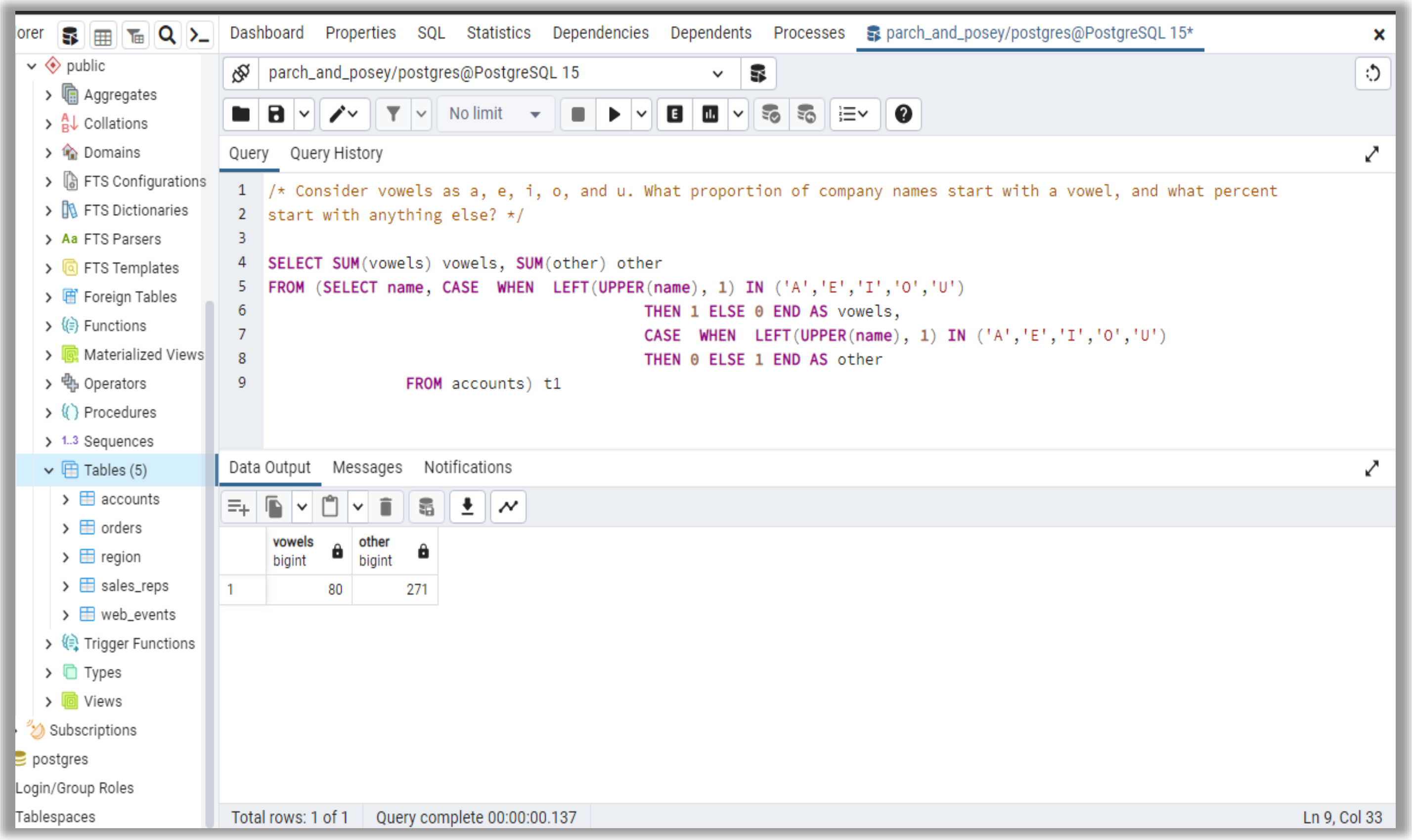


```
/* Use the accounts table and a CASE statement to create two groups: one group of company names that start with
a number and a second group of those company names that start with a letter.
What proportion of company names start with a letter? */

SELECT SUM(num) nums,  SUM(letter) letters
FROM (SELECT name, CASE  WHEN  LEFT(UPPER(name), 1) IN ('0','1','2','3','4','5','6','7','8','9')

                THEN 1 ELSE 0 END AS num,

CASE  WHEN  LEFT(UPPER(name), 1) IN ('0','1','2','3','4','5','6','7','8','9')

                THEN 0 ELSE 1 END AS letter
                FROM accounts) t1;
```

	nums bigint	letters bigint
1	1	350

Total rows: 1 of 1 Query complete 00:00:00.172 Ln 12, Col 16

D. Consider vowels as a, e, i, o, and u. What proportion of company names start with a vowel, and what percent start with anything else?

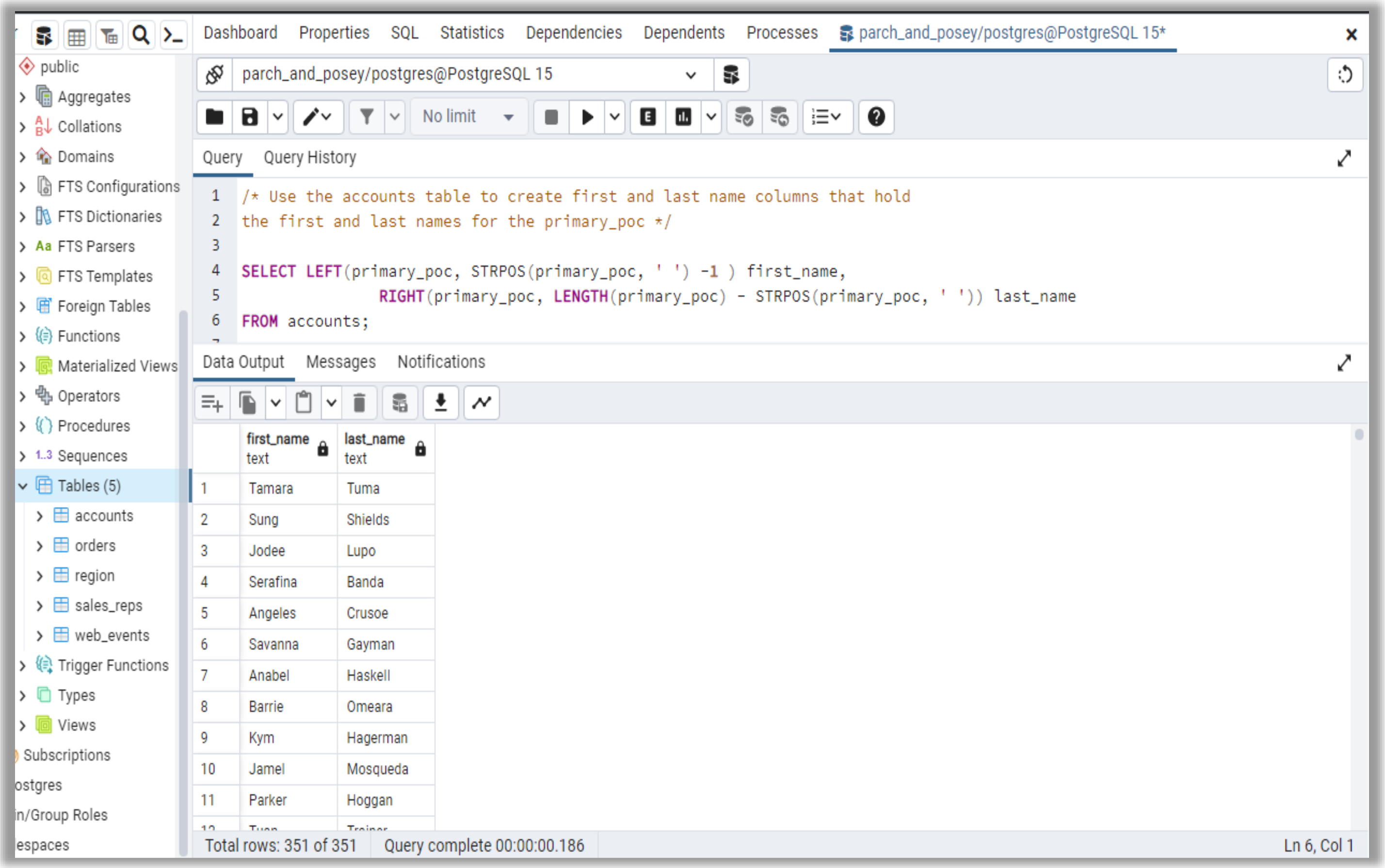


```
/* Consider vowels as a, e, i, o, and u. What proportion of company names start with a vowel, and what percent
start with anything else? */

SELECT SUM(vowels) vowels, SUM(other) other
FROM (SELECT name, CASE  WHEN  LEFT(UPPER(name), 1) IN ('A','E','I','O','U')
                                        THEN 1 ELSE 0 END AS vowels,
                                        CASE  WHEN  LEFT(UPPER(name), 1) IN ('A','E','I','O','U')
                                        THEN 0 ELSE 1 END AS other
              FROM accounts) t1
```

	vowels bigint	other bigint
1	80	271

Total rows: 1 of 1   Query complete 00:00:00.137

E. Use the **accounts** table to create first and last name columns that hold the first and last names for the **primary_poc**.

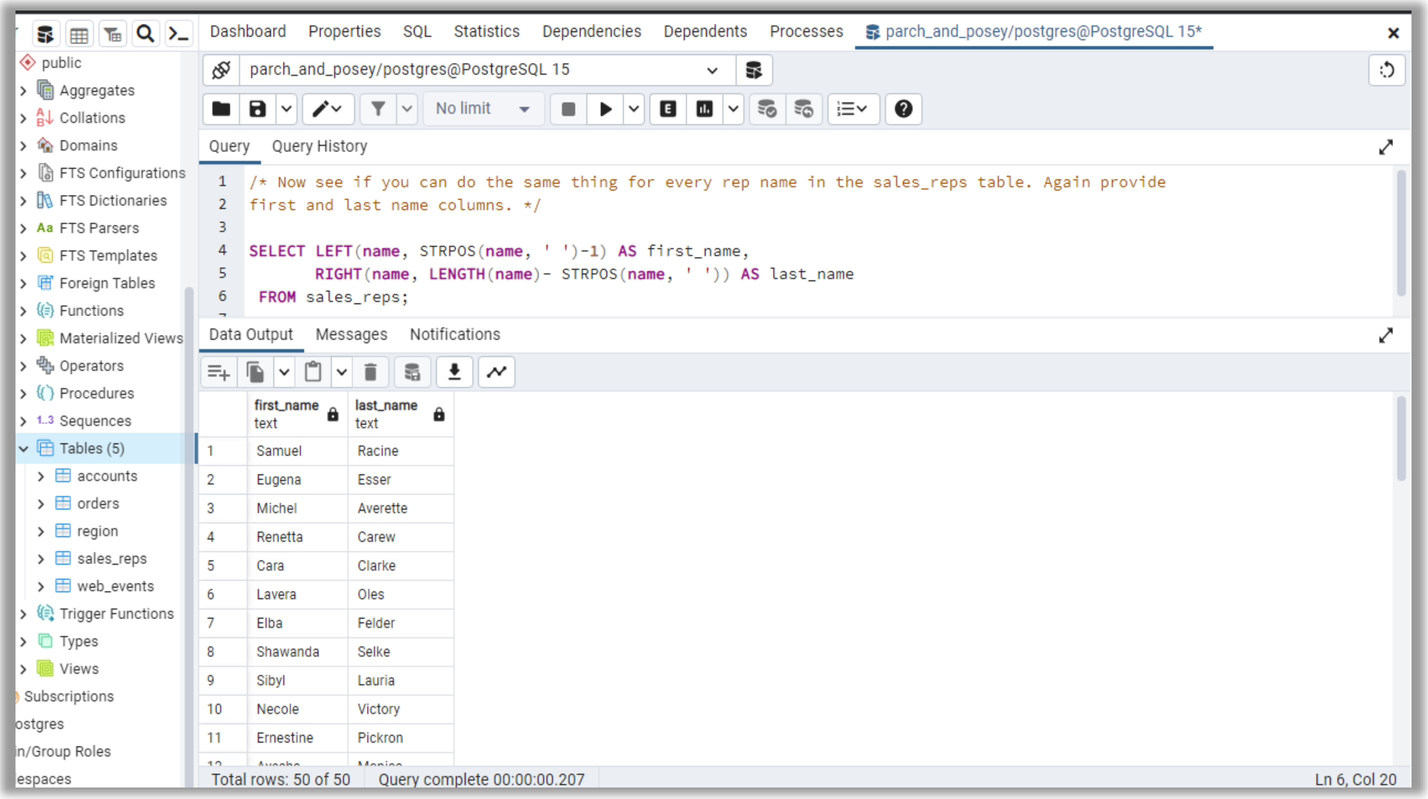


```
/* Use the accounts table to create first and last name columns that hold
the first and last names for the primary_poc */

SELECT LEFT(primary_poc, STRPOS(primary_poc, ' ') -1 ) first_name,
          RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name
FROM accounts;
```

	first_name text	last_name text
1	Tamara	Tuma
2	Sung	Shields
3	Jodee	Lupo
4	Serafina	Banda
5	Angeles	Crusoe
6	Savanna	Gayman
7	Anabel	Haskell
8	Barrie	Omeara
9	Kym	Hagerman
10	Jamel	Mosqueda
11	Parker	Hoggan
12	Tuan	Trainor

Total rows: 351 of 351   Query complete 00:00:00.186

F. Now see if you can do the same thing for every rep **name** in the **sales_reps** table. Again, provide first and last name columns.

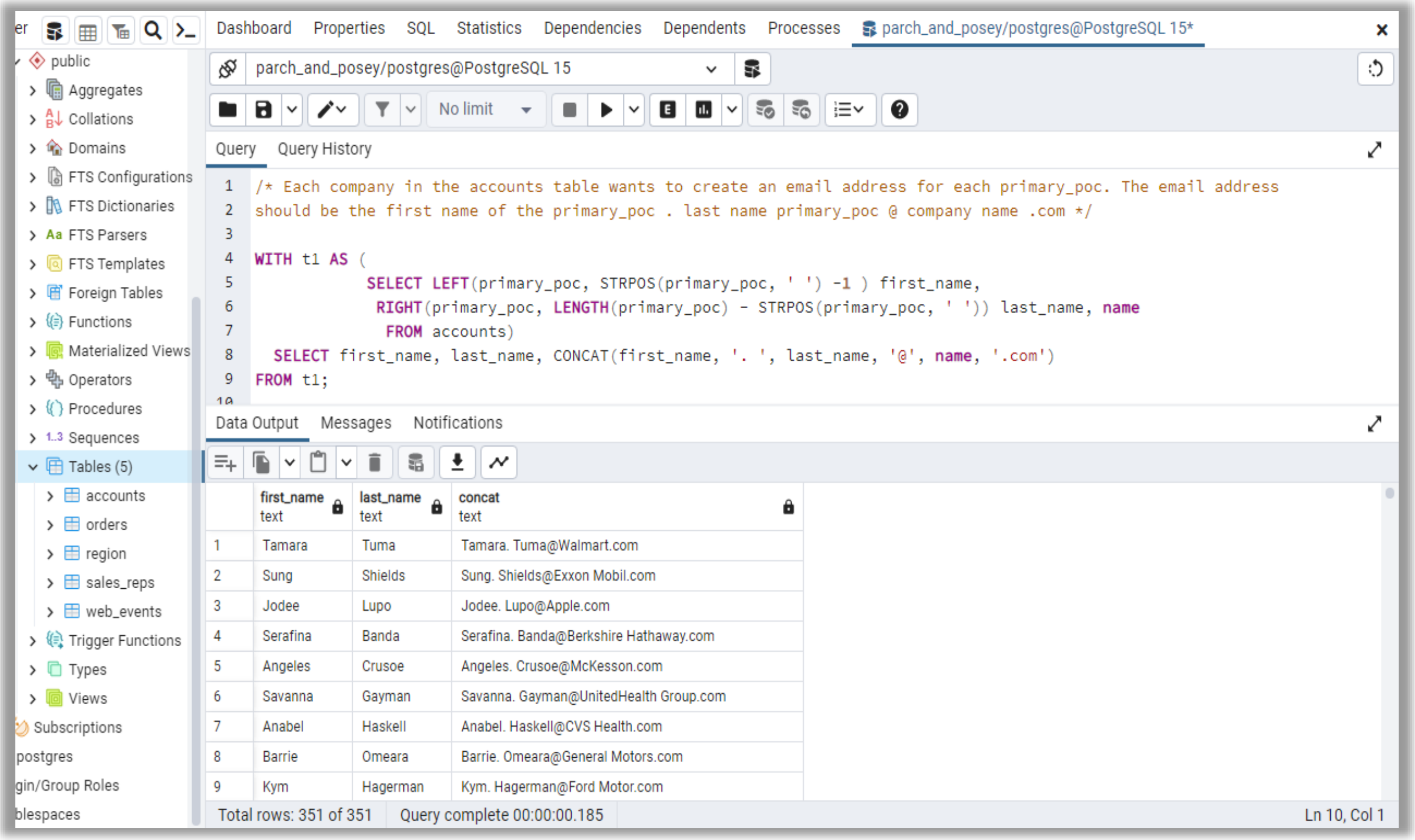


```
/* Now see if you can do the same thing for every rep name in the sales_reps table. Again provide
first and last name columns. */

SELECT LEFT(name, STRPOS(name, ' ')-1) AS first_name,
       RIGHT(name, LENGTH(name)- STRPOS(name, ' ')) AS last_name
 FROM sales_reps;
```

	first_name text	last_name text
1	Samuel	Racine
2	Eugena	Esser
3	Michel	Averette
4	Renetta	Carew
5	Cara	Clarke
6	Lavera	Oles
7	Elba	Felder
8	Shawanda	Selke
9	Sibyl	Lauria
10	Necole	Victory
11	Ernestine	Pickron

Total rows: 50 of 50 Query complete 00:00:00.207 Ln 6, Col 20

G. Each company in the accounts table wants to create an email address for each primary_poc. The email address should be the first name of the **primary_poc**. last name **primary_poc** @ company name .com.

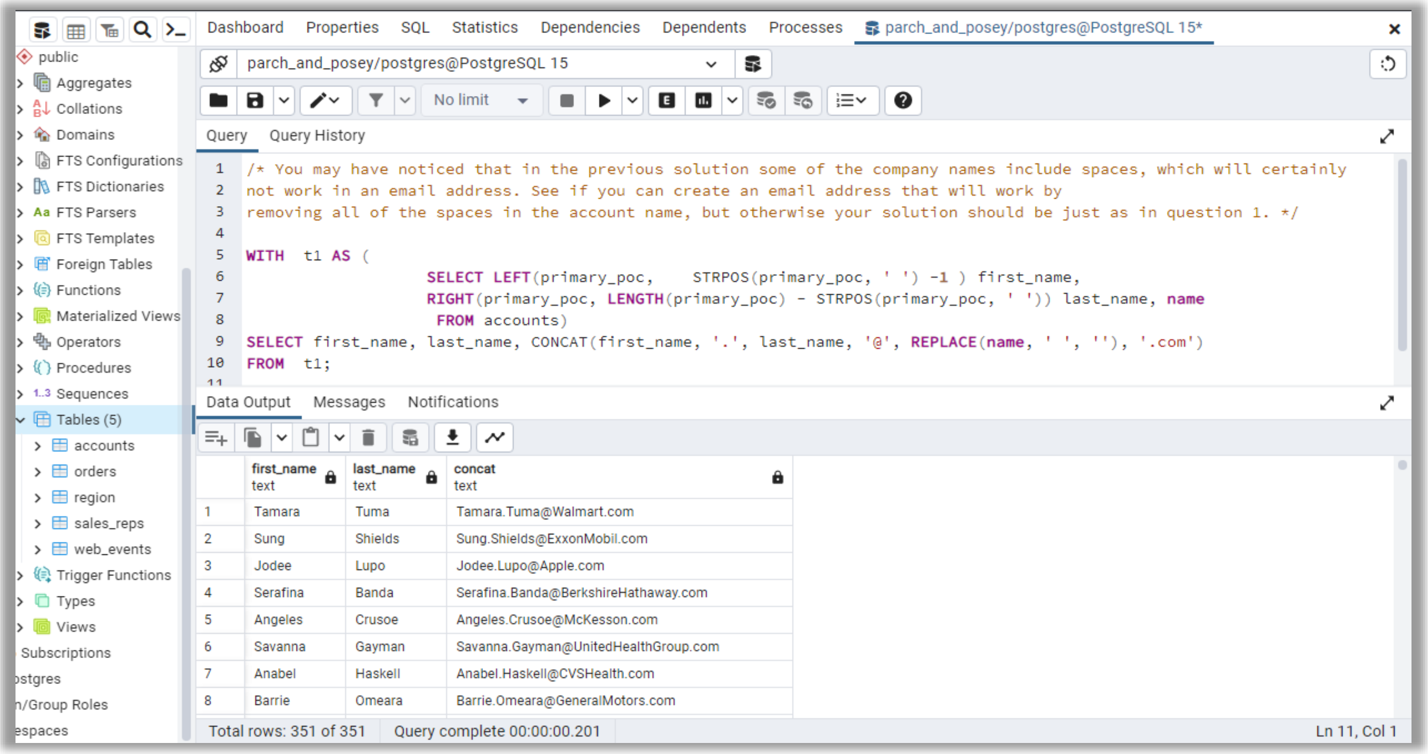


```
/* Each company in the accounts table wants to create an email address for each primary_poc. The email address
should be the first name of the primary_poc . last name primary_poc @ company name .com */

WITH t1 AS (
            SELECT LEFT(primary_poc, STRPOS(primary_poc, ' ') -1 ) first_name,
             RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name, name
              FROM accounts)
  SELECT first_name, last_name, CONCAT(first_name, '. ', last_name, '@', name, '.com')
FROM t1;
```

	first_name text	last_name text	concat text
1	Tamara	Tuma	Tamara. Tuma@Walmart.com
2	Sung	Shields	Sung. Shields@Exxon Mobil.com
3	Jodee	Lupo	Jodee. Lupo@Apple.com
4	Serafina	Banda	Serafina. Banda@Berkshire Hathaway.com
5	Angeles	Crusoe	Angeles. Crusoe@McKesson.com
6	Savanna	Gayman	Savanna. Gayman@UnitedHealth Group.com
7	Anabel	Haskell	Anabel. Haskell@CVS Health.com
8	Barrie	Omeara	Barrie. Omeara@General Motors.com
9	Kym	Hagerman	Kym. Hagerman@Ford Motor.com

Total rows: 351 of 351 Query complete 00:00:00.185 Ln 10, Col 1

H. You may have noticed that in the previous solution some of the company names include spaces, which will certainly not work in an email address. See if you can create an email address that will work by removing all of the spaces in the account name, but otherwise your solution should be just as in question 1

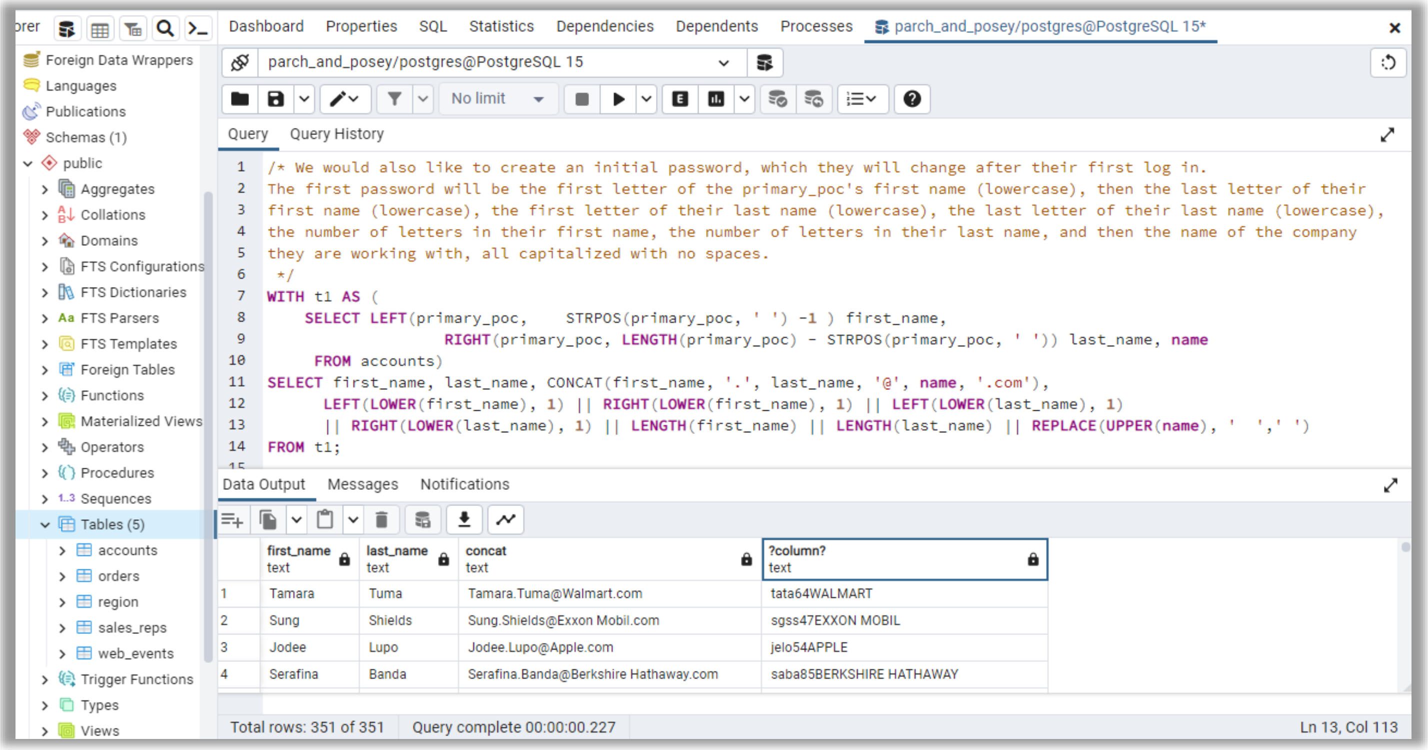


```
/* You may have noticed that in the previous solution some of the company names include spaces, which will certainly
not work in an email address. See if you can create an email address that will work by
removing all of the spaces in the account name, but otherwise your solution should be just as in question 1. */

WITH  t1 AS (
                    SELECT LEFT(primary_poc,    STRPOS(primary_poc, ' ') -1 ) first_name,
                    RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name, name
                     FROM accounts)
SELECT first_name, last_name, CONCAT(first_name, '.', last_name, '@', REPLACE(name, ' ', ''), '.com')
FROM  t1;
```

	first_name text	last_name text	concat text
1	Tamara	Tuma	Tamara.Tuma@Walmart.com
2	Sung	Shields	Sung.Shields@ExxonMobil.com
3	Jodee	Lupo	Jodee.Lupo@Apple.com
4	Serafina	Banda	Serafina.Banda@BerkshireHathaway.com
5	Angeles	Crusoe	Angeles.Crusoe@McKesson.com
6	Savanna	Gayman	Savanna.Gayman@UnitedHealthGroup.com
7	Anabel	Haskell	Anabel.Haskell@CVSHealth.com
8	Barrie	Omeara	Barrie.Omeara@GeneralMotors.com

Total rows: 351 of 351   Query complete 00:00:00.201   Ln 11, Col 1

I. We would also like to create an initial password, which they will change after their first log in. The first password will be the first letter of the primary_poc's first name (lowercase), then the last letter of their first name (lowercase), the first letter of their last name (lowercase), the last letter of their last name (lowercase), the number of letters in their first name, the number of letters in their last name, and then the name of the company they are working with, all capitalized with no spaces



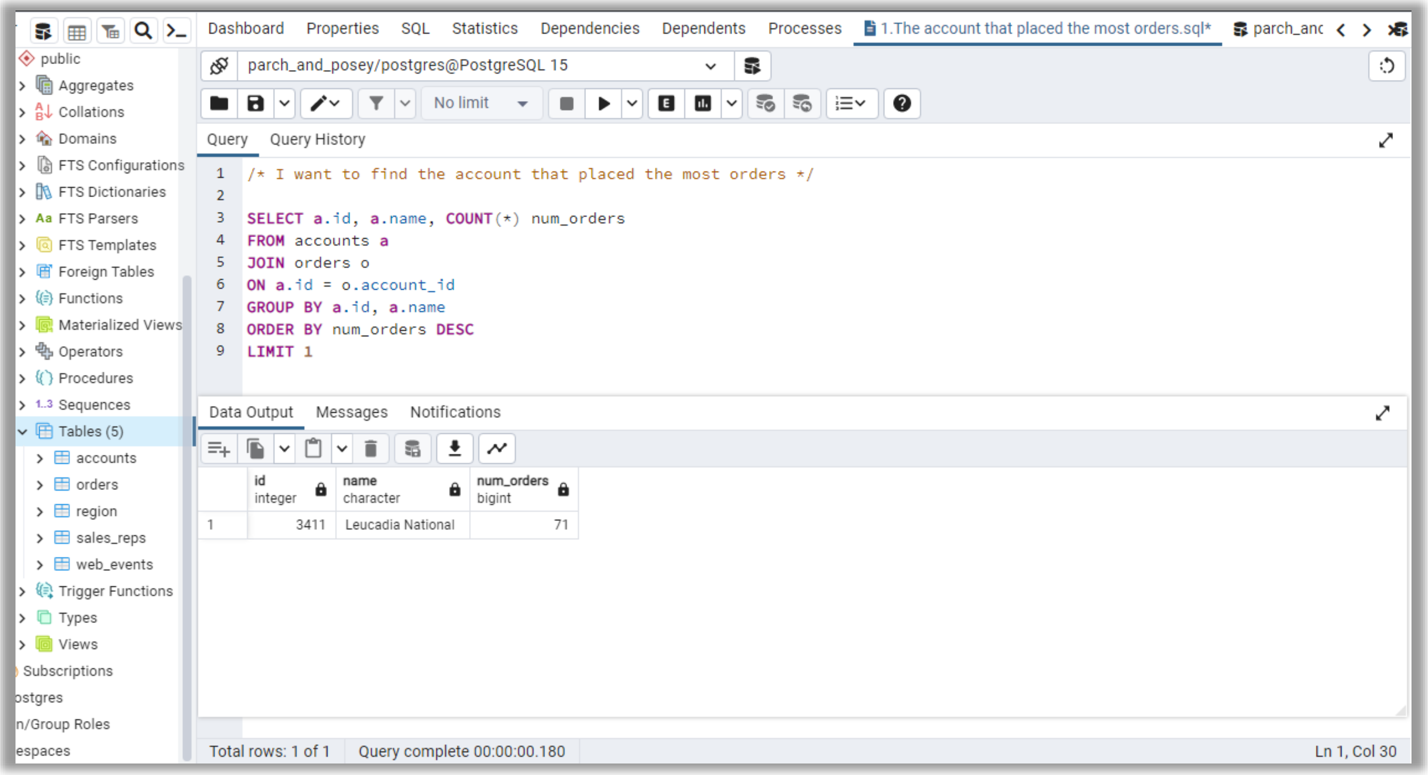
```
/* We would also like to create an initial password, which they will change after their first log in.
The first password will be the first letter of the primary_poc's first name (lowercase), then the last letter of their
first name (lowercase), the first letter of their last name (lowercase), the last letter of their last name (lowercase),
the number of letters in their first name, the number of letters in their last name, and then the name of the company
they are working with, all capitalized with no spaces.
 */
WITH t1 AS (
    SELECT LEFT(primary_poc,    STRPOS(primary_poc, ' ') -1 ) first_name,
                   RIGHT(primary_poc, LENGTH(primary_poc) - STRPOS(primary_poc, ' ')) last_name, name
     FROM accounts)
SELECT first_name, last_name, CONCAT(first_name, '.', last_name, '@', name, '.com'),
      LEFT(LOWER(first_name), 1) || RIGHT(LOWER(first_name), 1) || LEFT(LOWER(last_name), 1)
      || RIGHT(LOWER(last_name), 1) || LENGTH(first_name) || LENGTH(last_name) || REPLACE(UPPER(name), '  ',' ')
FROM t1;
```

	first_name text	last_name text	concat text	?column? text
1	Tamara	Tuma	Tamara.Tuma@Walmart.com	tata64WALMART
2	Sung	Shields	Sung.Shields@Exxon Mobil.com	sgss47EXXON MOBIL
3	Jodee	Lupo	Jodee.Lupo@Apple.com	jelo54APPLE
4	Serafina	Banda	Serafina.Banda@Berkshire Hathaway.com	saba85BERKSHIRE HATHAWAY

Total rows: 351 of 351   Query complete 00:00:00.227   Ln 13, Col 113

DATA ANALYSIS

1. The Account that Placed the Most Orders:



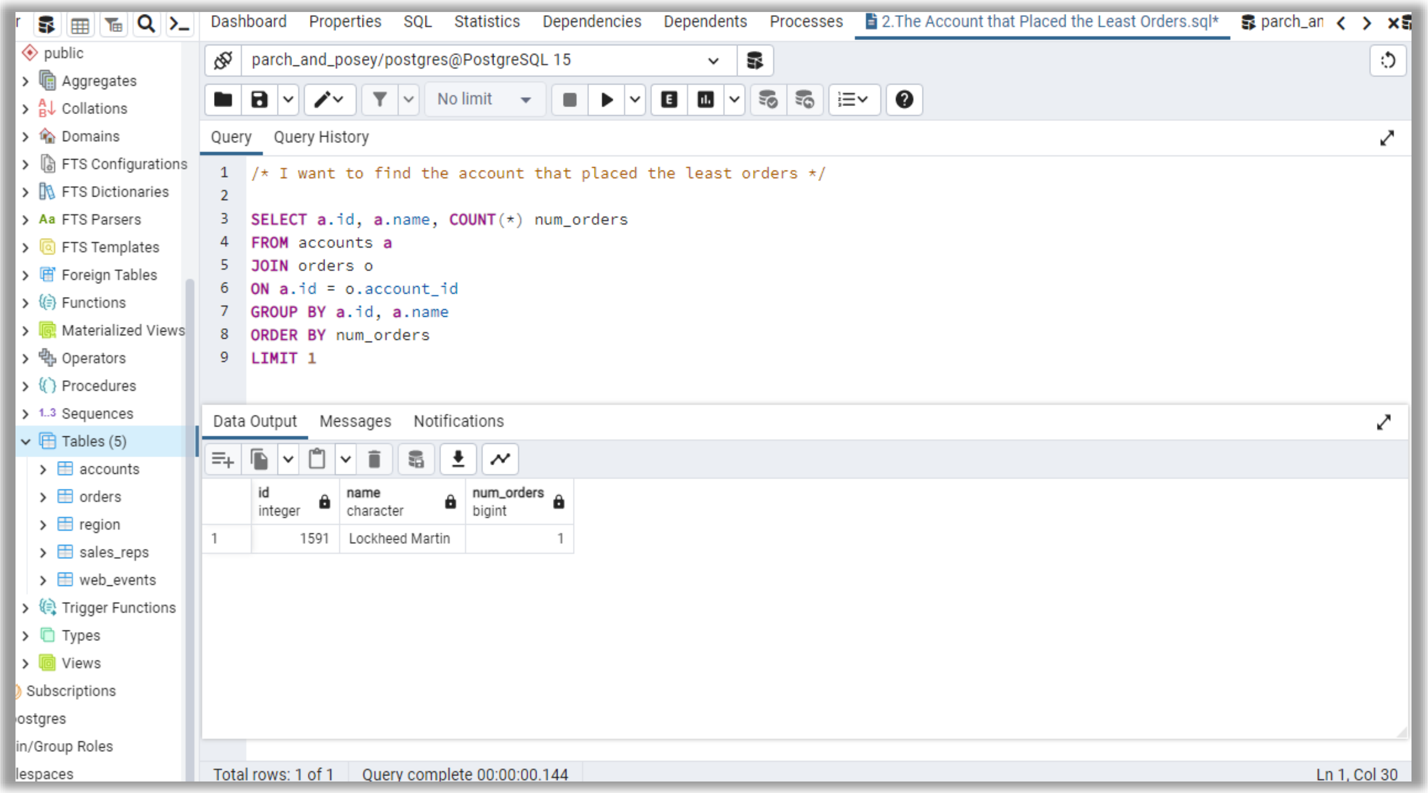
```sql
/* I want to find the account that placed the most orders */

SELECT a.id, a.name, COUNT(*) num_orders
FROM accounts a
JOIN orders o
ON a.id = o.account_id
GROUP BY a.id, a.name
ORDER BY num_orders DESC
LIMIT 1
```

	id integer	name character	num_orders bigint
1	3411	Leucadia National	71

Total rows: 1 of 1 Query complete 00:00:00.180 Ln 1, Col 30

"Leucadia National" placed the most orders. This account placed 71 orders in total.

2. The Account that Placed the Least Orders:



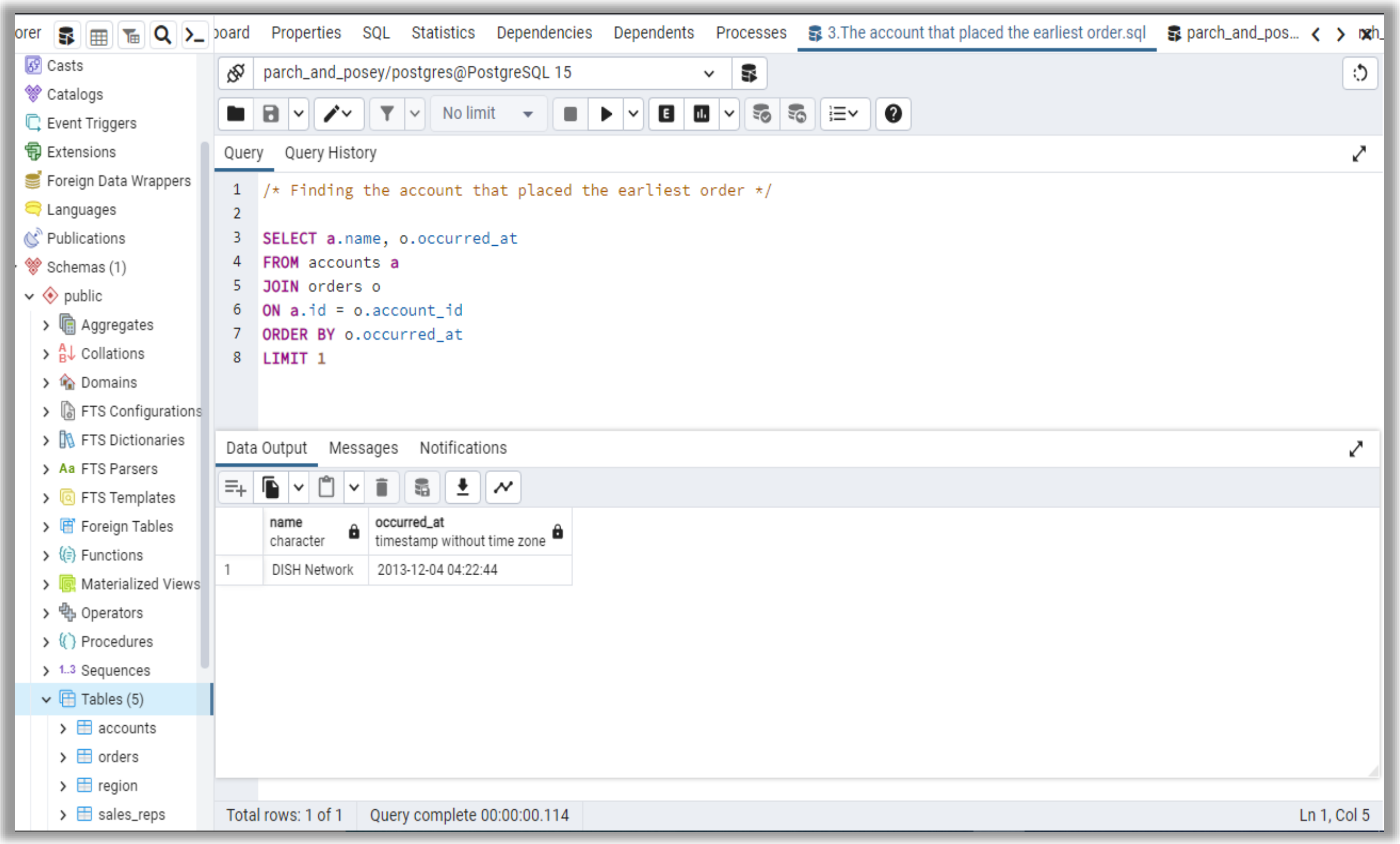
```sql
/* I want to find the account that placed the least orders */

SELECT a.id, a.name, COUNT(*) num_orders
FROM accounts a
JOIN orders o
ON a.id = o.account_id
GROUP BY a.id, a.name
ORDER BY num_orders
LIMIT 1
```

	id integer	name character	num_orders bigint
1	1591	Lockheed Martin	1

Total rows: 1 of 1 Query complete 00:00:00.144 Ln 1, Col 30

"Lockheed Martin" placed the least orders. This account placed 1 order in total.

3. The Account that Placed the Earliest Order:



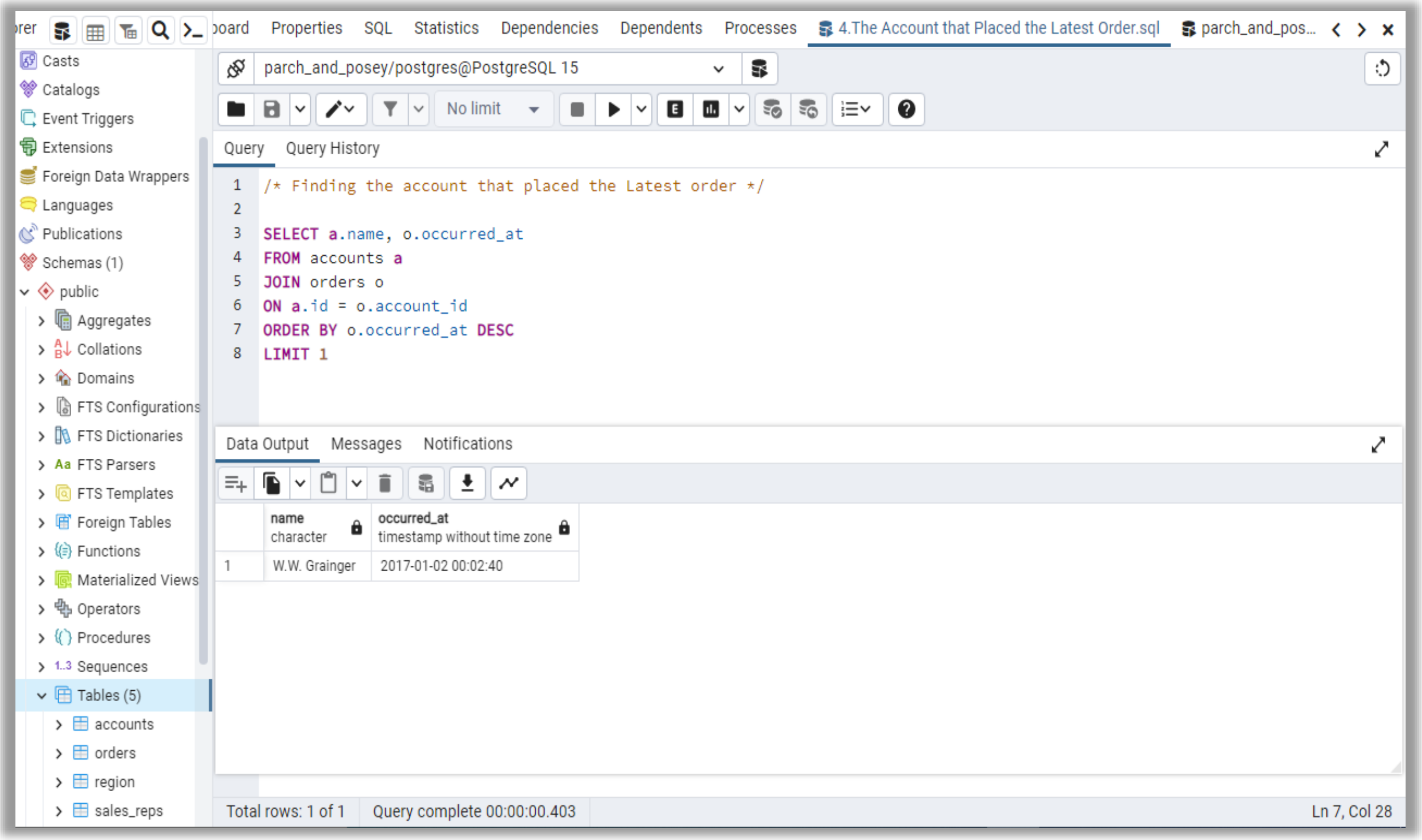
```sql
/* Finding the account that placed the earliest order */

SELECT a.name, o.occurred_at
FROM accounts a
JOIN orders o
ON a.id = o.account_id
ORDER BY o.occurred_at
LIMIT 1
```

	name character	occurred_at timestamp without time zone
1	DISH Network	2013-12-04 04:22:44

Total rows: 1 of 1 Query complete 00:00:00.114 Ln 1, Col 5

"DISH Network" placed the earliest order in **"2013-12-04 04:22:44"**

4. The Account that Placed the Latest Order:



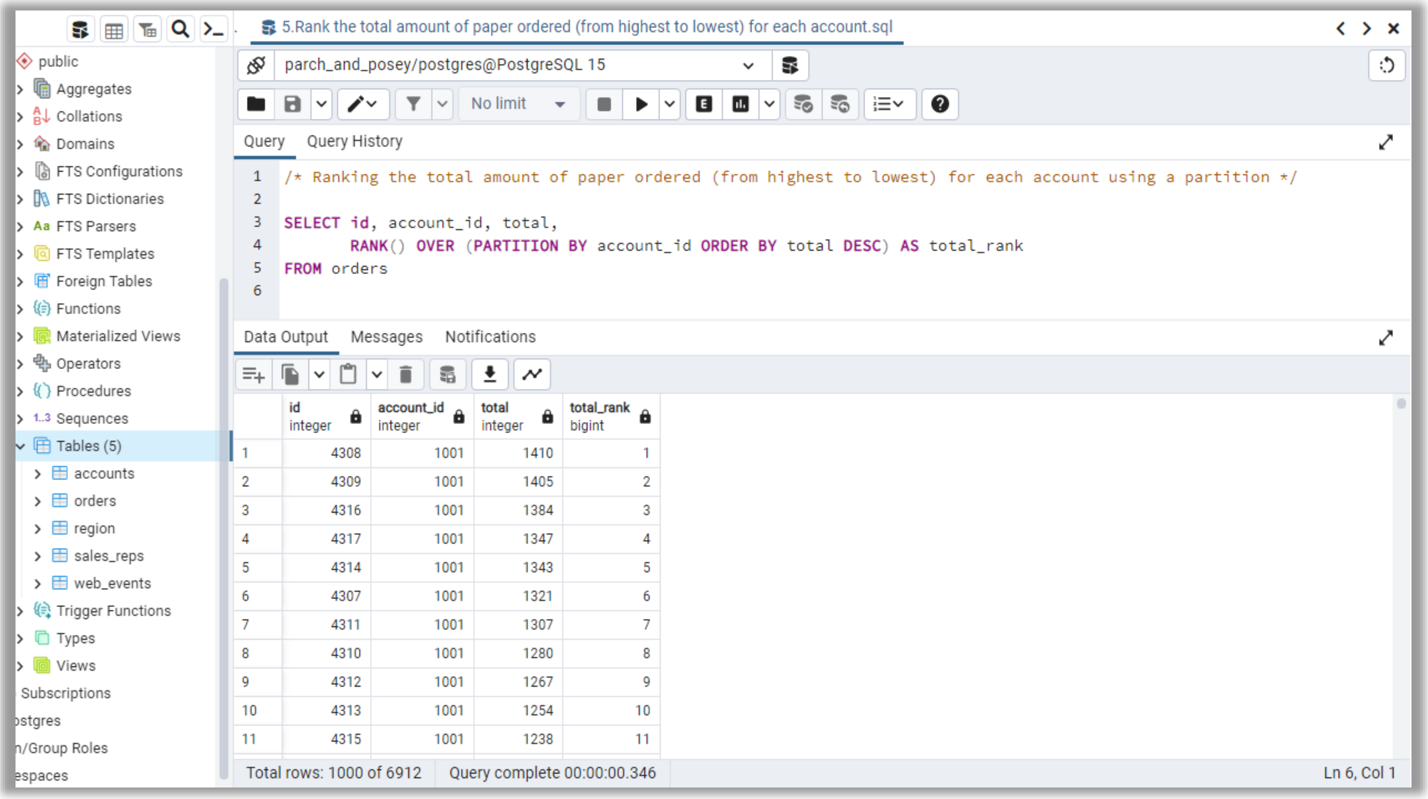
```sql
/* Finding the account that placed the Latest order */

SELECT a.name, o.occurred_at
FROM accounts a
JOIN orders o
ON a.id = o.account_id
ORDER BY o.occurred_at DESC
LIMIT 1
```

	name character	occurred_at timestamp without time zone
1	W.W. Grainger	2017-01-02 00:02:40

Total rows: 1 of 1 Query complete 00:00:00.403 Ln 7, Col 28

"W.W. Grainger" placed the latest order in **"2017-01-02 00:02:40"**

5. Rank the Total Amount of Paper Ordered (From Highest to Lowest) for Each Account using a Partition:

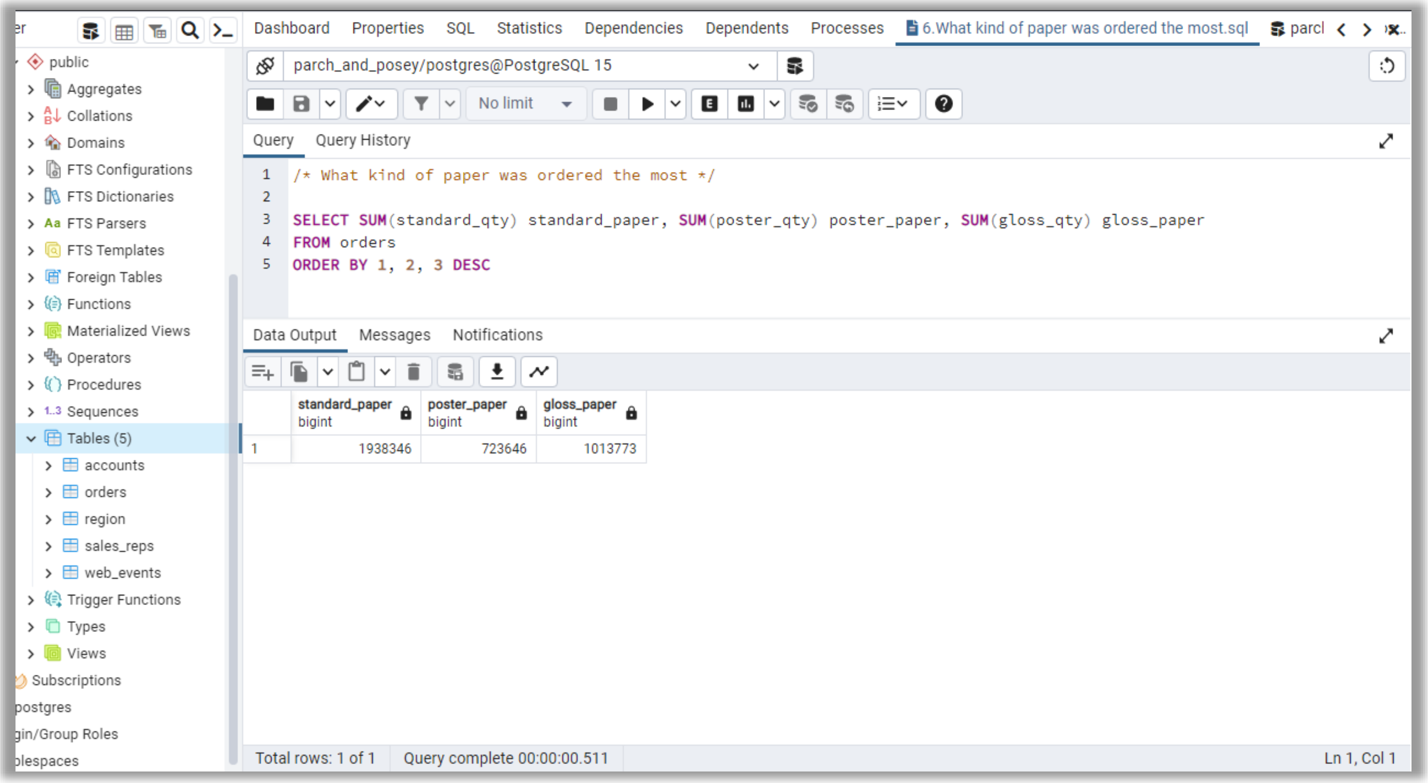


```
/* Ranking the total amount of paper ordered (from highest to lowest) for each account using a partition */

SELECT id, account_id, total,
       RANK() OVER (PARTITION BY account_id ORDER BY total DESC) AS total_rank
FROM orders

```

	id integer	account_id integer	total integer	total_rank bigint
1	4308	1001	1410	1
2	4309	1001	1405	2
3	4316	1001	1384	3
4	4317	1001	1347	4
5	4314	1001	1343	5
6	4307	1001	1321	6
7	4311	1001	1307	7
8	4310	1001	1280	8
9	4312	1001	1267	9
10	4313	1001	1254	10
11	4315	1001	1238	11

Total rows: 1000 of 6912 Query complete 00:00:00.346 Ln 6, Col 1

6. What Kind of Paper was Ordered the Most:



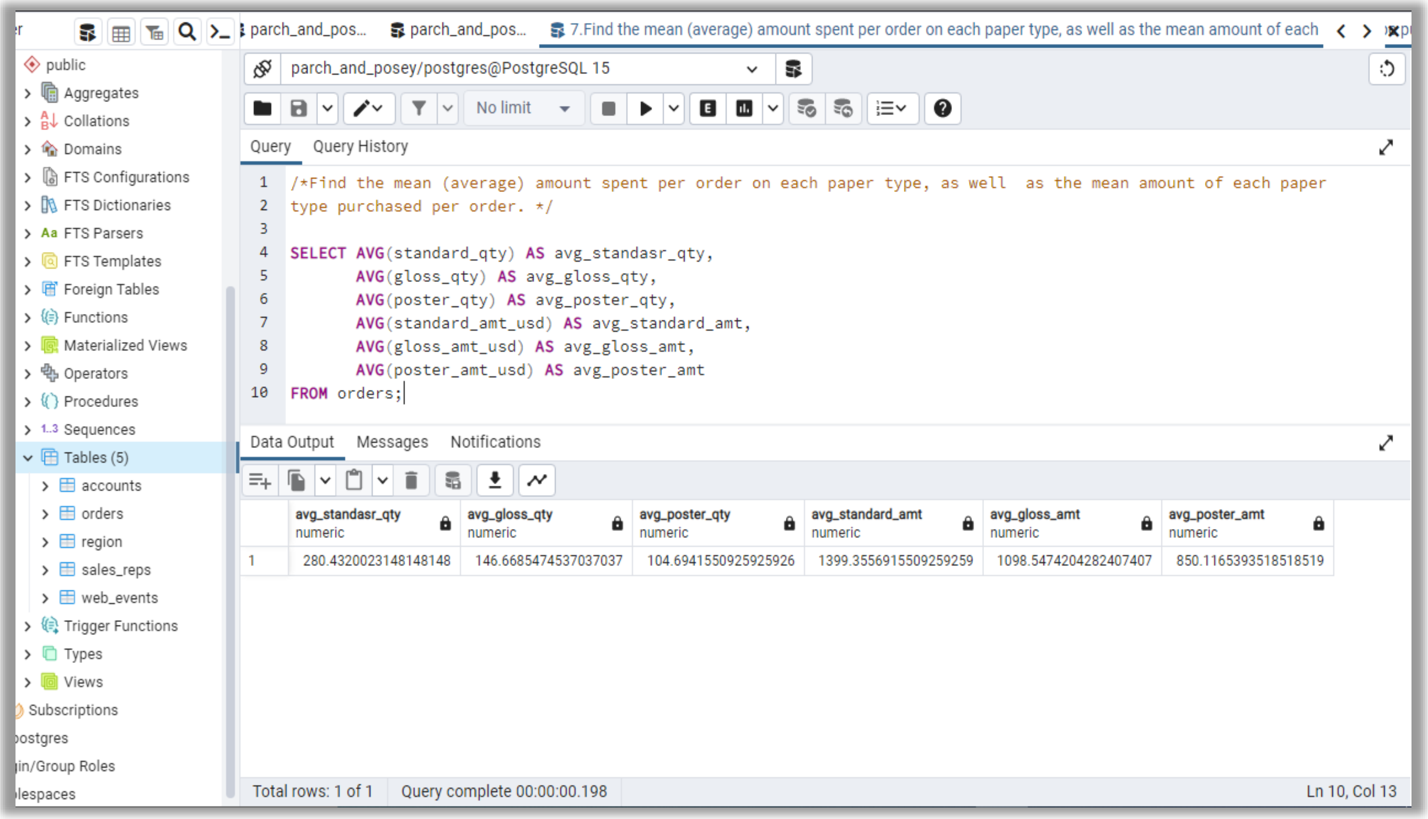
```
/* What kind of paper was ordered the most */

SELECT SUM(standard_qty) standard_paper, SUM(poster_qty) poster_paper, SUM(gloss_qty) gloss_paper
FROM orders
ORDER BY 1, 2, 3 DESC
```

	standard_paper bigint	poster_paper bigint	gloss_paper bigint
1	1938346	723646	1013773

Total rows: 1 of 1 Query complete 00:00:00.511 Ln 1, Col 1

Standard paper had **1938346** orders, **Post paper** had **723646** orders, **Gloss paper** had **1013773** orders. This analysis showed that **Standard paper** had the highest orders which was one million nine hundred thirty-eight thousand three hundred forty-six.

7. Find the Mean (Average) Amount Spent Per Order on Each Paper Type, as well as the Mean Amount of Each Paper type Purchased Per Order:

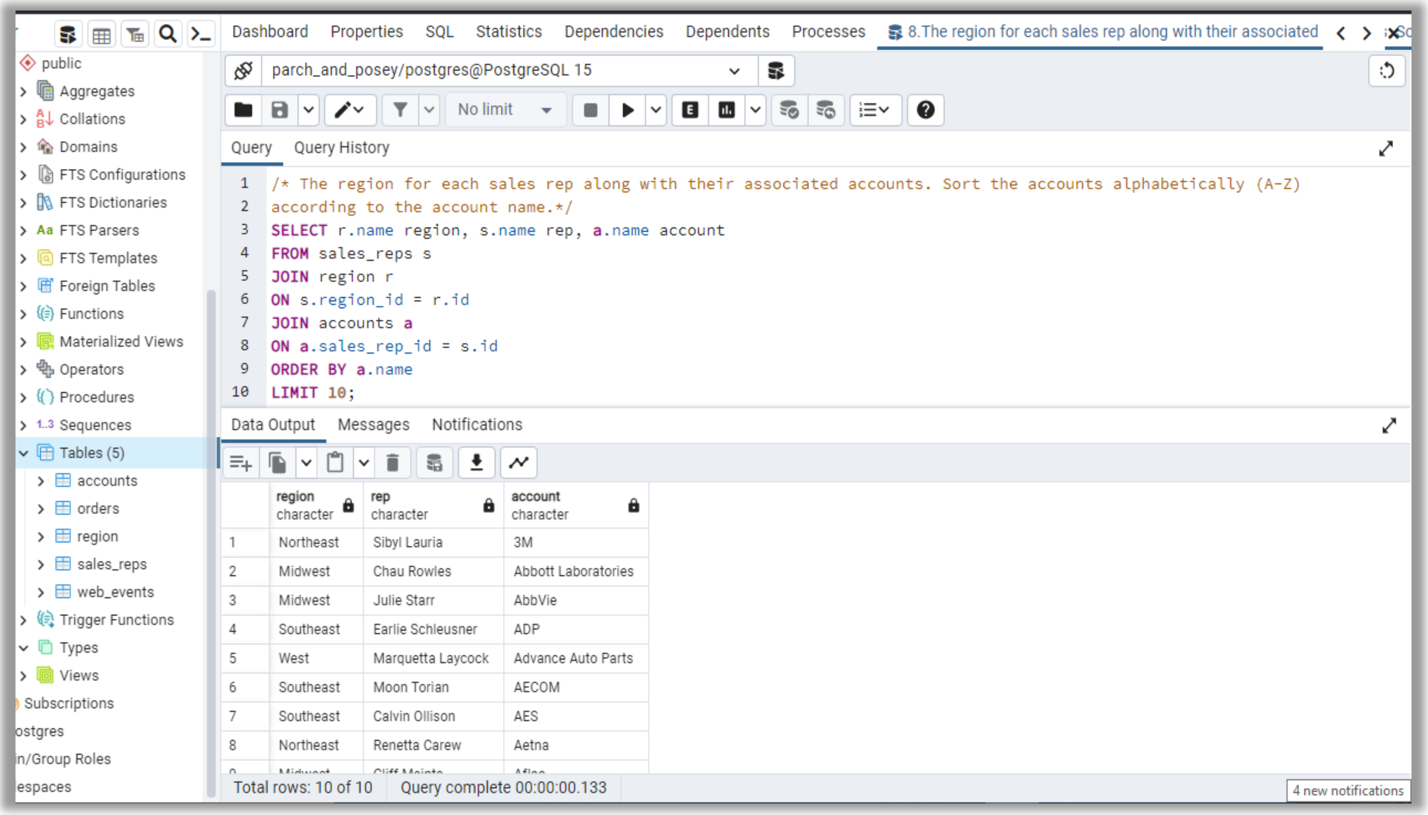


```
/*Find the mean (average) amount spent per order on each paper type, as well  as the mean amount of each paper
type purchased per order. */

SELECT AVG(standard_qty) AS avg_standasr_qty,
       AVG(gloss_qty) AS avg_gloss_qty,
       AVG(poster_qty) AS avg_poster_qty,
       AVG(standard_amt_usd) AS avg_standard_amt,
       AVG(gloss_amt_usd) AS avg_gloss_amt,
       AVG(poster_amt_usd) AS avg_poster_amt
FROM orders;
```

	avg_standasr_qty numeric	avg_gloss_qty numeric	avg_poster_qty numeric	avg_standard_amt numeric	avg_gloss_amt numeric	avg_poster_amt numeric
1	280.4320023148148148	146.6685474537037037	104.6941550925925926	1399.3556915509259259	1098.5474204282407407	850.1165393518518519

Total rows: 1 of 1   Query complete 00:00:00.198   Ln 10, Col 13

8. The Region for Each Sales Rep Along with their Associated Accounts. Sort the Accounts Alphabetically (A-Z) According to the Account Name:

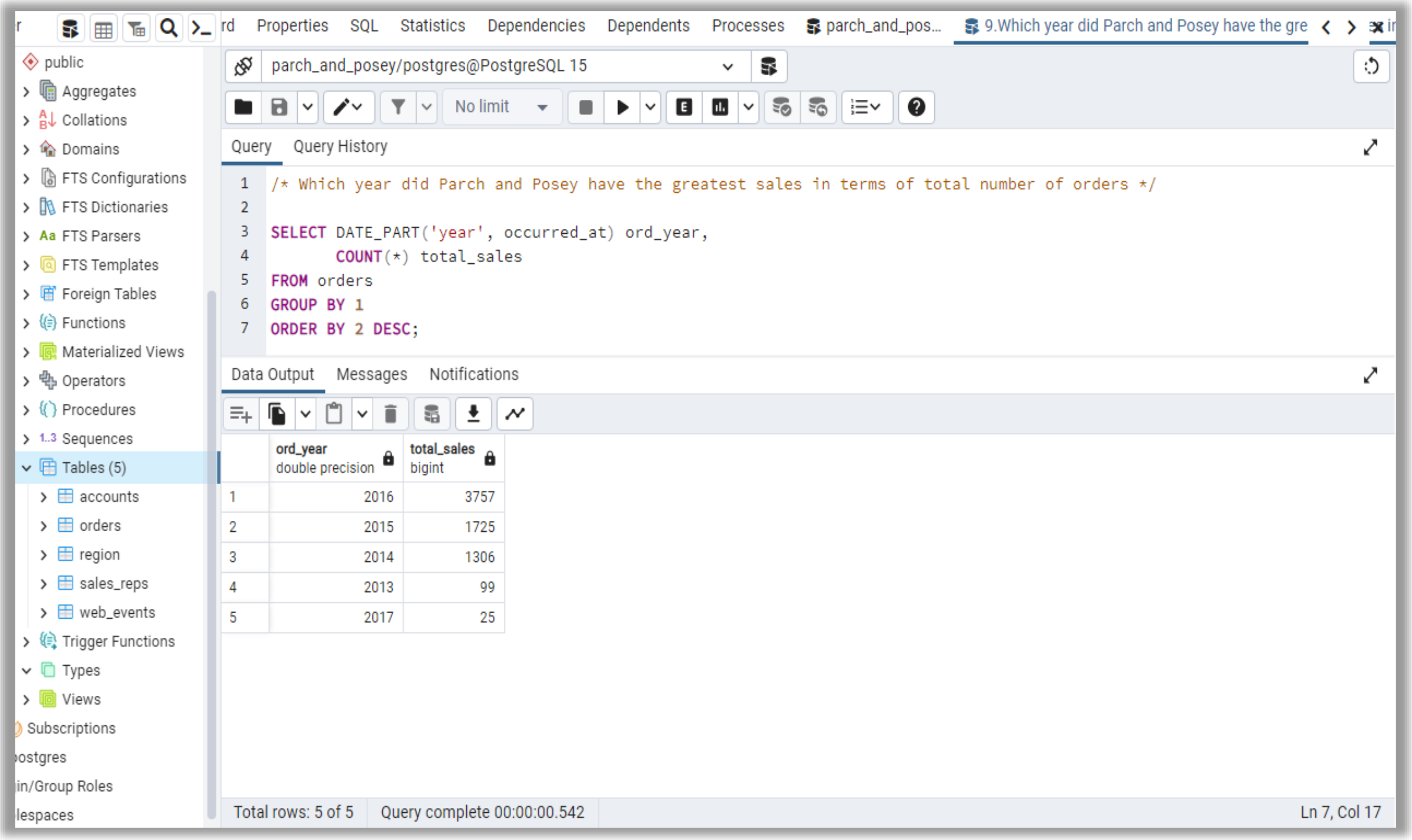


```
/* The region for each sales rep along with their associated accounts. Sort the accounts alphabetically (A-Z)
according to the account name.*/
SELECT r.name region, s.name rep, a.name account
FROM sales_reps s
JOIN region r
ON s.region_id = r.id
JOIN accounts a
ON a.sales_rep_id = s.id
ORDER BY a.name
LIMIT 10;
```

	region character	rep character	account character
1	Northeast	Sibyl Lauria	3M
2	Midwest	Chau Rowles	Abbott Laboratories
3	Midwest	Julie Starr	AbbVie
4	Southeast	Earlie Schleusner	ADP
5	West	Marquetta Laycock	Advance Auto Parts
6	Southeast	Moon Torian	AECOM
7	Southeast	Calvin Ollison	AES
8	Northeast	Renetta Carew	Aetna
9	Midwest	Cliff Meints	Aflac

Total rows: 10 of 10   Query complete 00:00:00.133

9. Which Year Did Parch and Posey have the Greatest Sales in Terms of Total Number of Orders:



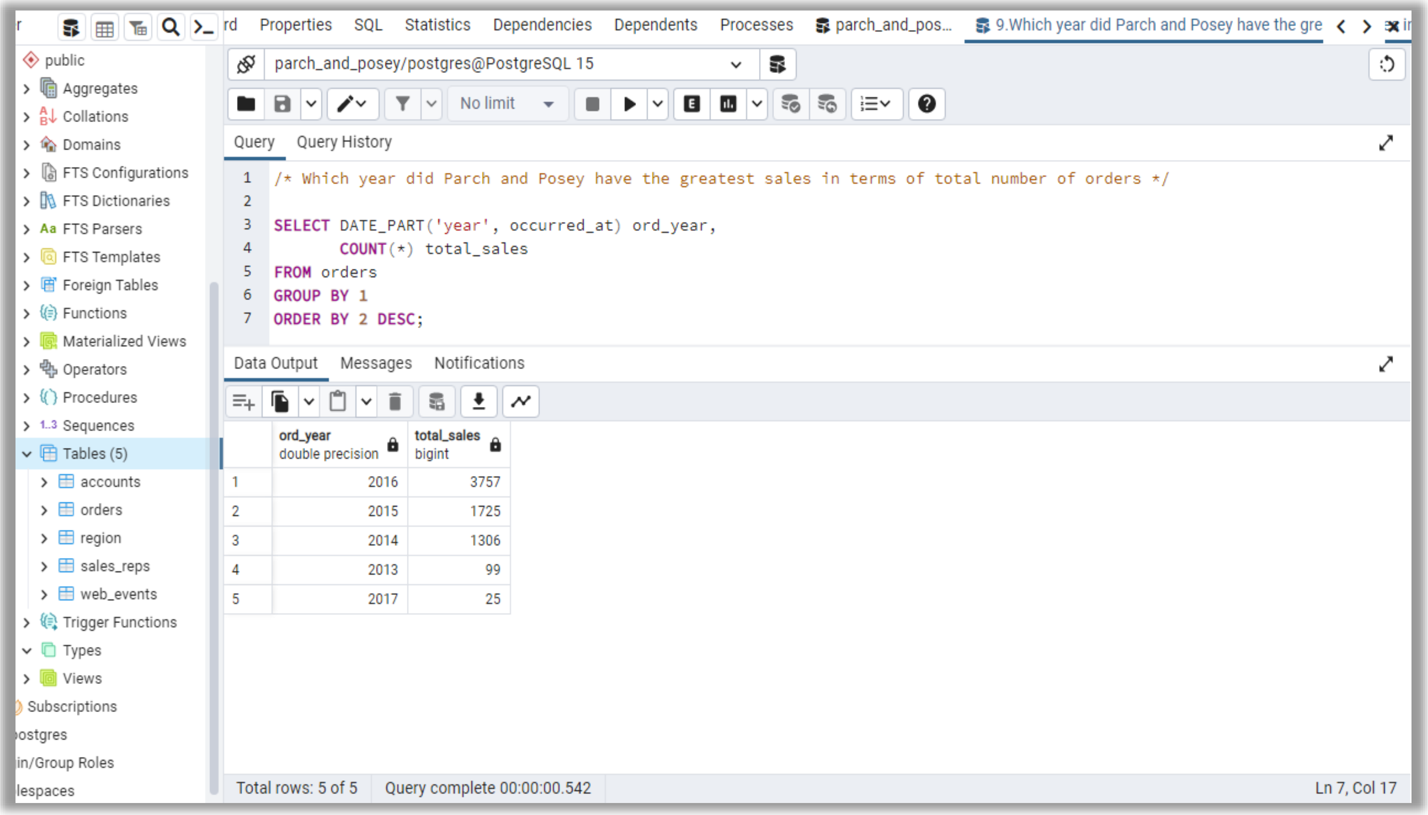
```
/* Which year did Parch and Posey have the greatest sales in terms of total number of orders */

SELECT DATE_PART('year', occurred_at) ord_year,
       COUNT(*) total_sales
FROM orders
GROUP BY 1
ORDER BY 2 DESC;
```

```
   ord_year          total_sales
   double precision  bigint
1              2016         3757
2              2015         1725
3              2014         1306
4              2013           99
5              2017           25
```

Total rows: 5 of 5   Query complete 00:00:00.542   Ln 7, Col 17

Parch and Posey had their greatest sales in **2016** with over **3757 total sales**.

10. Which Month did Parch and Posey have the Greatest Sales in Terms of Total Number of Orders:



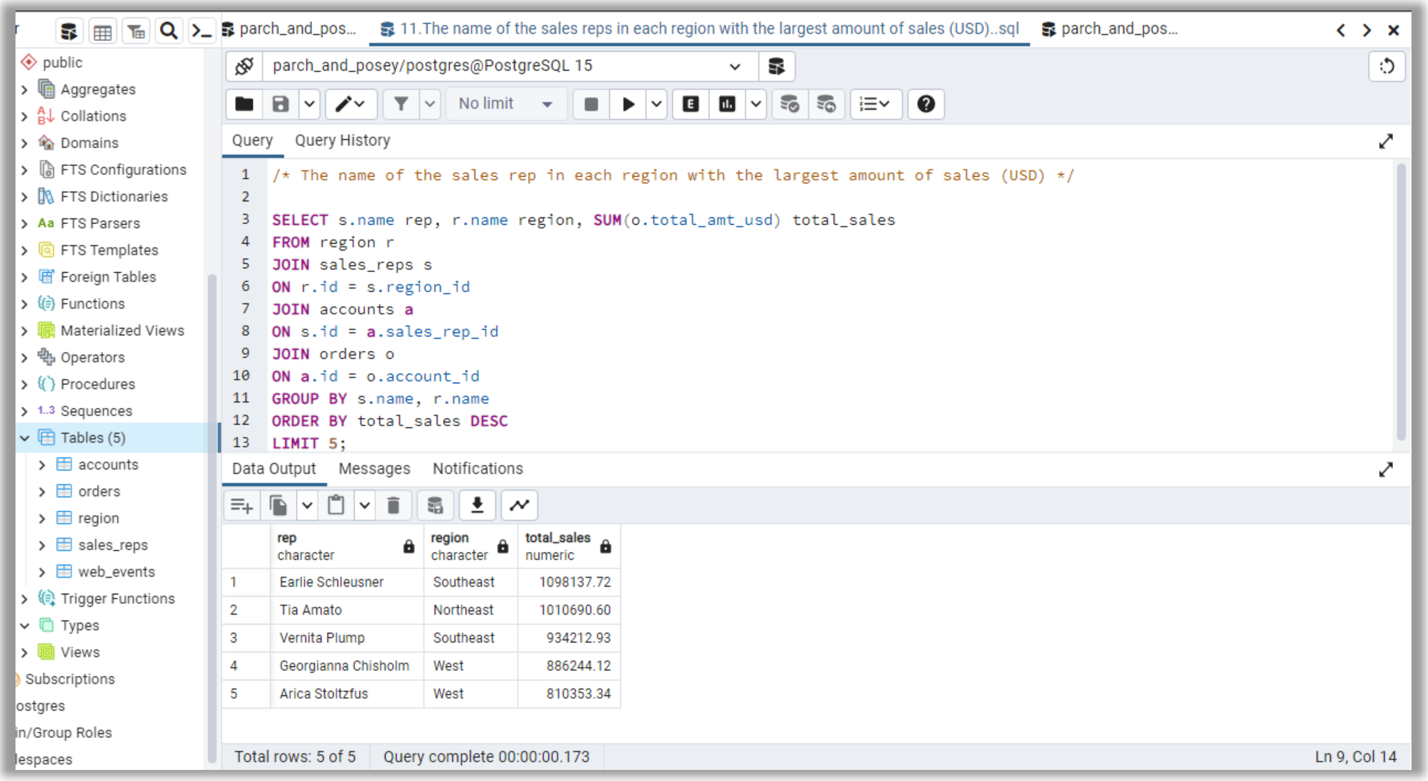
```
/* Which year did Parch and Posey have the greatest sales in terms of total number of orders */

SELECT DATE_PART('year', occurred_at) ord_year,
       COUNT(*) total_sales
FROM orders
GROUP BY 1
ORDER BY 2 DESC;
```

```
   ord_year          total_sales
   double precision  bigint
1              2016         3757
2              2015         1725
3              2014         1306
4              2013           99
5              2017           25
```

Total rows: 5 of 5   Query complete 00:00:00.542   Ln 7, Col 17

Parch and Posey had their greatest sales in the month of **December** with over **882 total sales**

11. The Names of the Sales Reps in each Region with the Largest Amount of Sales (USD):



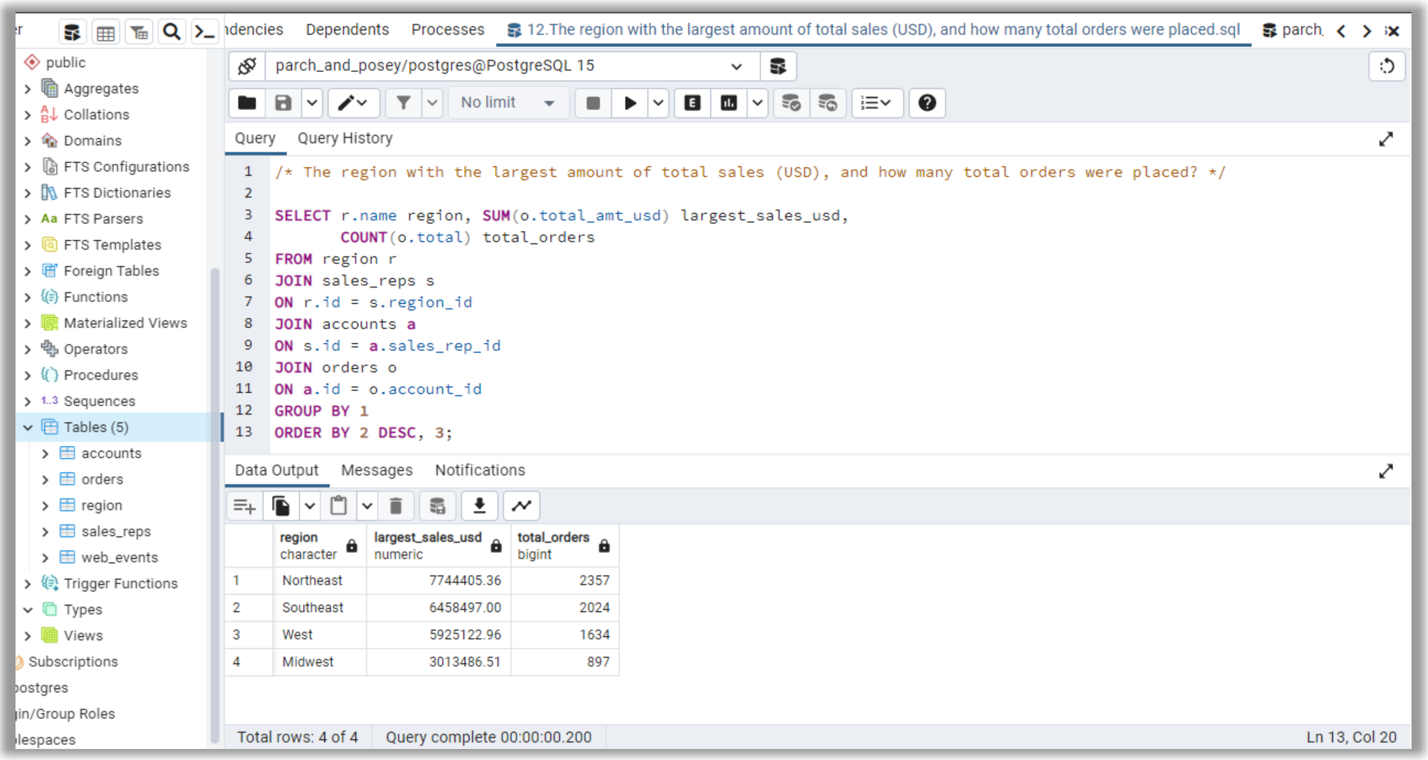
```sql
/* The name of the sales rep in each region with the largest amount of sales (USD) */

SELECT s.name rep, r.name region, SUM(o.total_amt_usd) total_sales
FROM region r
JOIN sales_reps s
ON r.id = s.region_id
JOIN accounts a
ON s.id = a.sales_rep_id
JOIN orders o
ON a.id = o.account_id
GROUP BY s.name, r.name
ORDER BY total_sales DESC
LIMIT 5;
```

	rep character	region character	total_sales numeric
1	Earlie Schleusner	Southeast	1098137.72
2	Tia Amato	Northeast	1010690.60
3	Vernita Plump	Southeast	934212.93
4	Georgianna Chisholm	West	886244.12
5	Arica Stoltzfus	West	810353.34

Total rows: 5 of 5 Query complete 00:00:00.173 Ln 9, Col 14

"Earlie Schleusner" from the **"Southeast"** region had the largest amount of sales **(1098137.72)**

12. The Region with the Largest Amount of Total Sales (USD), and How Many Total Orders were Placed:



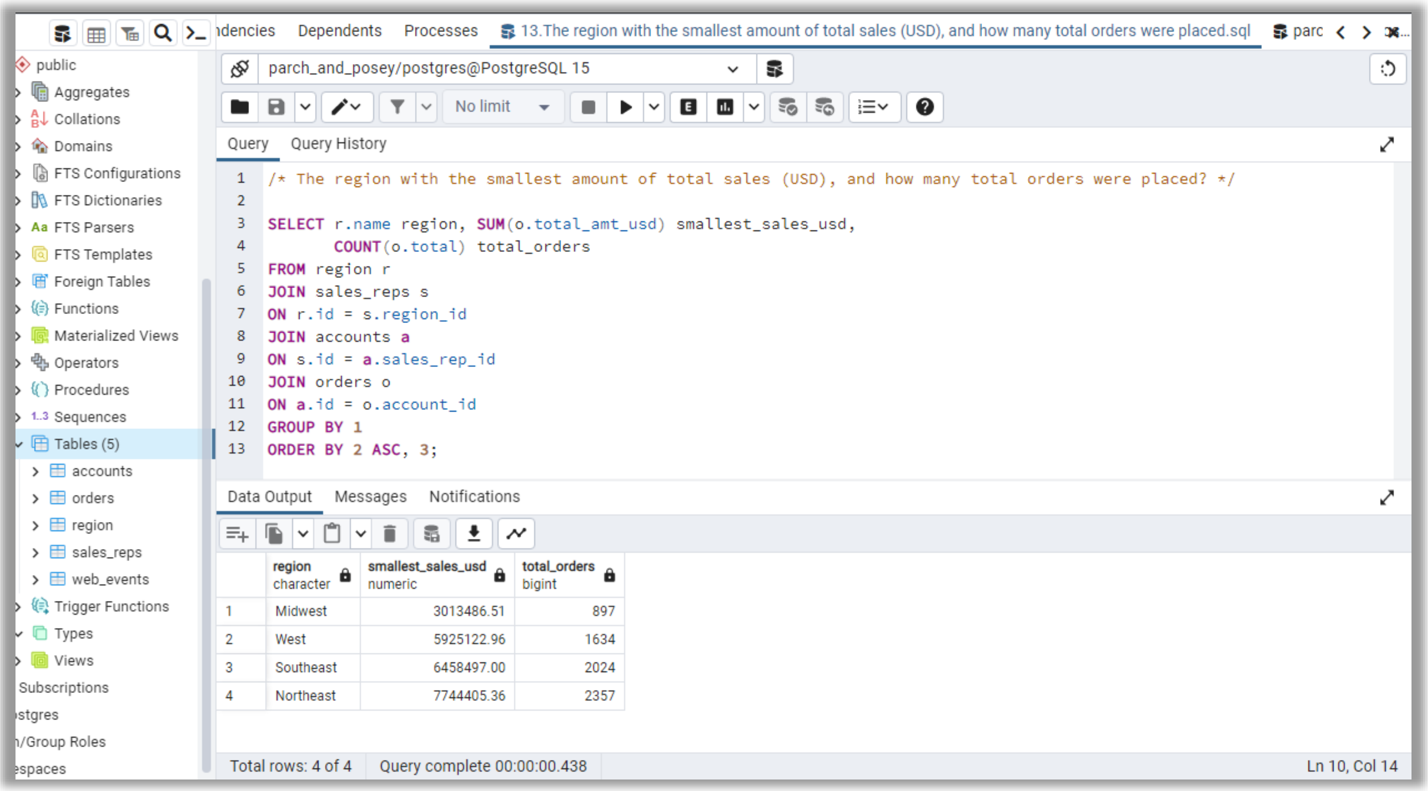
```sql
/* The region with the largest amount of total sales (USD), and how many total orders were placed? */

SELECT r.name region, SUM(o.total_amt_usd) largest_sales_usd,
       COUNT(o.total) total_orders
FROM region r
JOIN sales_reps s
ON r.id = s.region_id
JOIN accounts a
ON s.id = a.sales_rep_id
JOIN orders o
ON a.id = o.account_id
GROUP BY 1
ORDER BY 2 DESC, 3;
```

	region character	largest_sales_usd numeric	total_orders bigint
1	Northeast	7744405.36	2357
2	Southeast	6458497.00	2024
3	West	5925122.96	1634
4	Midwest	3013486.51	897

Total rows: 4 of 4 Query complete 00:00:00.200 Ln 13, Col 20

"Northeast" had the largest amount of total sales **(7744405.36)** in USD and **2357** orders were placed from that region.

13. The Region with the Smallest Amount of Total Sales (USD), and How Many Total Orders were Placed:



parch_and_posey/postgres@PostgreSQL 15

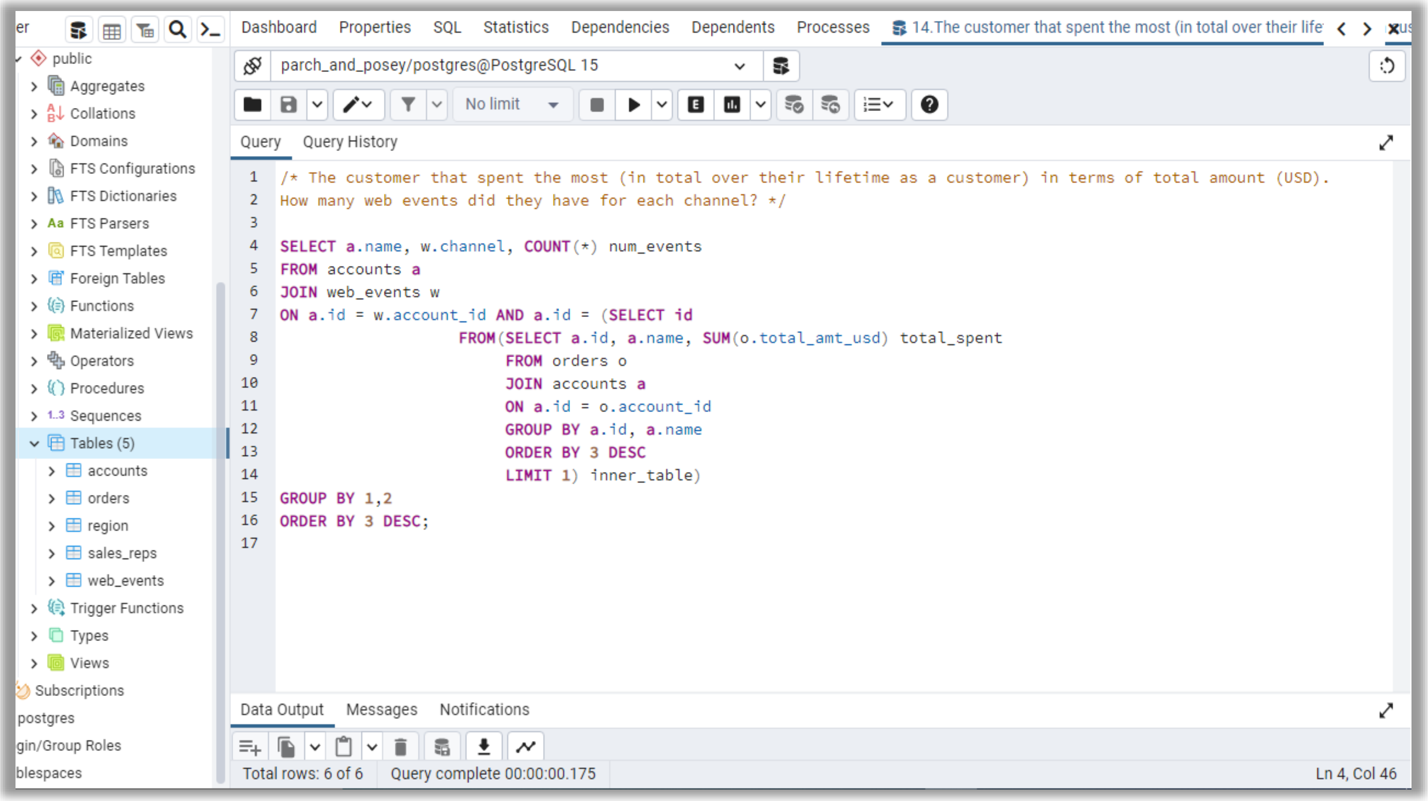
```sql
/* The region with the smallest amount of total sales (USD), and how many total orders were placed? */

SELECT r.name region, SUM(o.total_amt_usd) smallest_sales_usd,
       COUNT(o.total) total_orders
FROM region r
JOIN sales_reps s
ON r.id = s.region_id
JOIN accounts a
ON s.id = a.sales_rep_id
JOIN orders o
ON a.id = o.account_id
GROUP BY 1
ORDER BY 2 ASC, 3;
```

	region character	smallest_sales_usd numeric	total_orders bigint
1	Midwest	3013486.51	897
2	West	5925122.96	1634
3	Southeast	6458497.00	2024
4	Northeast	7744405.36	2357

Total rows: 4 of 4 Query complete 00:00:00.438 Ln 10, Col 14

"Midwest" had the smallest amount of total sales **(3013486.51)** in USD and **897** orders were placed from that region

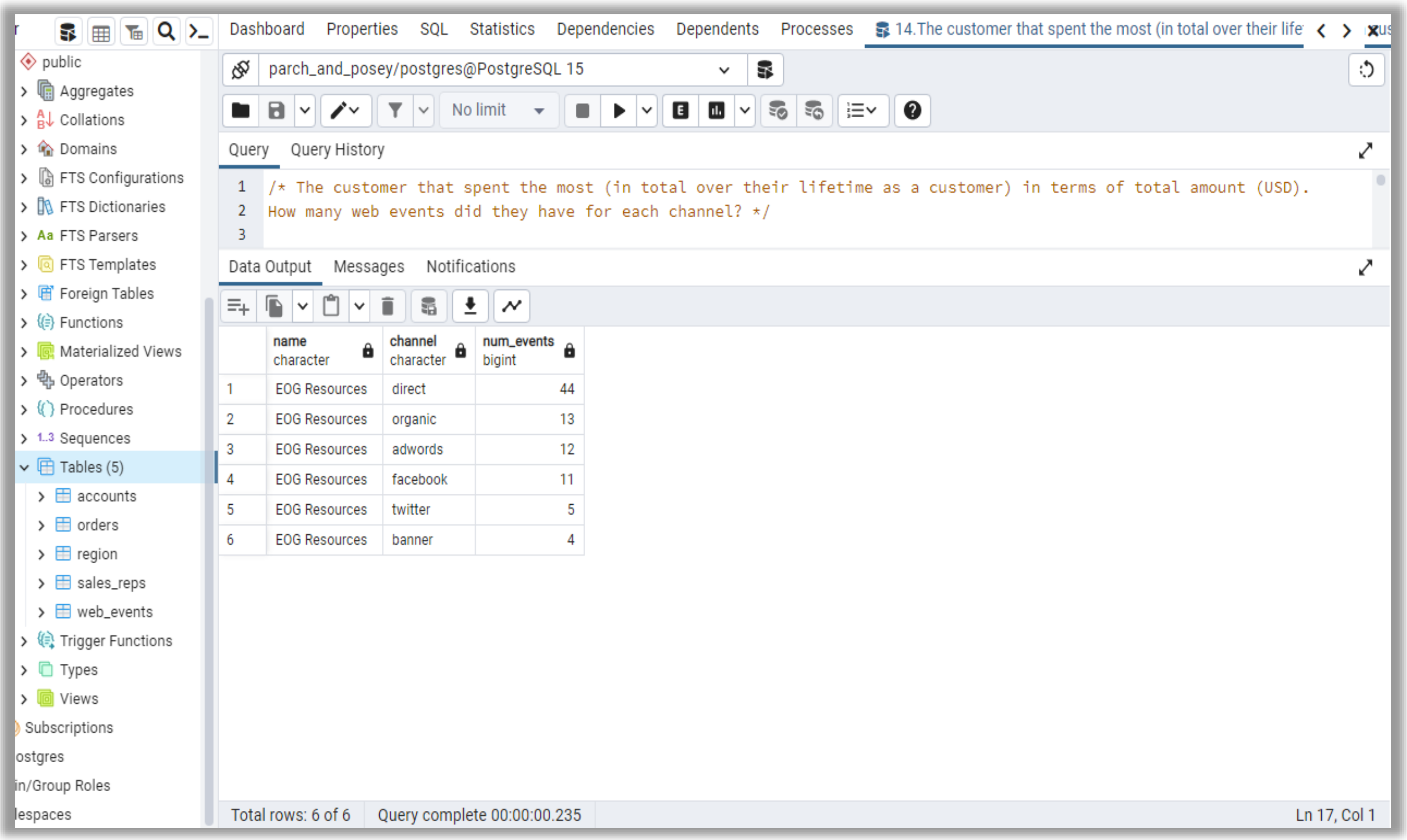
14. The Customer that Spent the Most (In Total Over their Lifetime as a Customer) in Terms of Total Amount (USD). How Many Web Events did they Have for Each Channel:



parch_and_posey/postgres@PostgreSQL 15

```sql
/* The customer that spent the most (in total over their lifetime as a customer) in terms of total amount (USD).
How many web events did they have for each channel? */

SELECT a.name, w.channel, COUNT(*) num_events
FROM accounts a
JOIN web_events w
ON a.id = w.account_id AND a.id = (SELECT id
                    FROM(SELECT a.id, a.name, SUM(o.total_amt_usd) total_spent
                         FROM orders o
                         JOIN accounts a
                         ON a.id = o.account_id
                         GROUP BY a.id, a.name
                         ORDER BY 3 DESC
                         LIMIT 1) inner_table)
GROUP BY 1,2
ORDER BY 3 DESC;
```

Total rows: 6 of 6 Query complete 00:00:00.175 Ln 4, Col 46



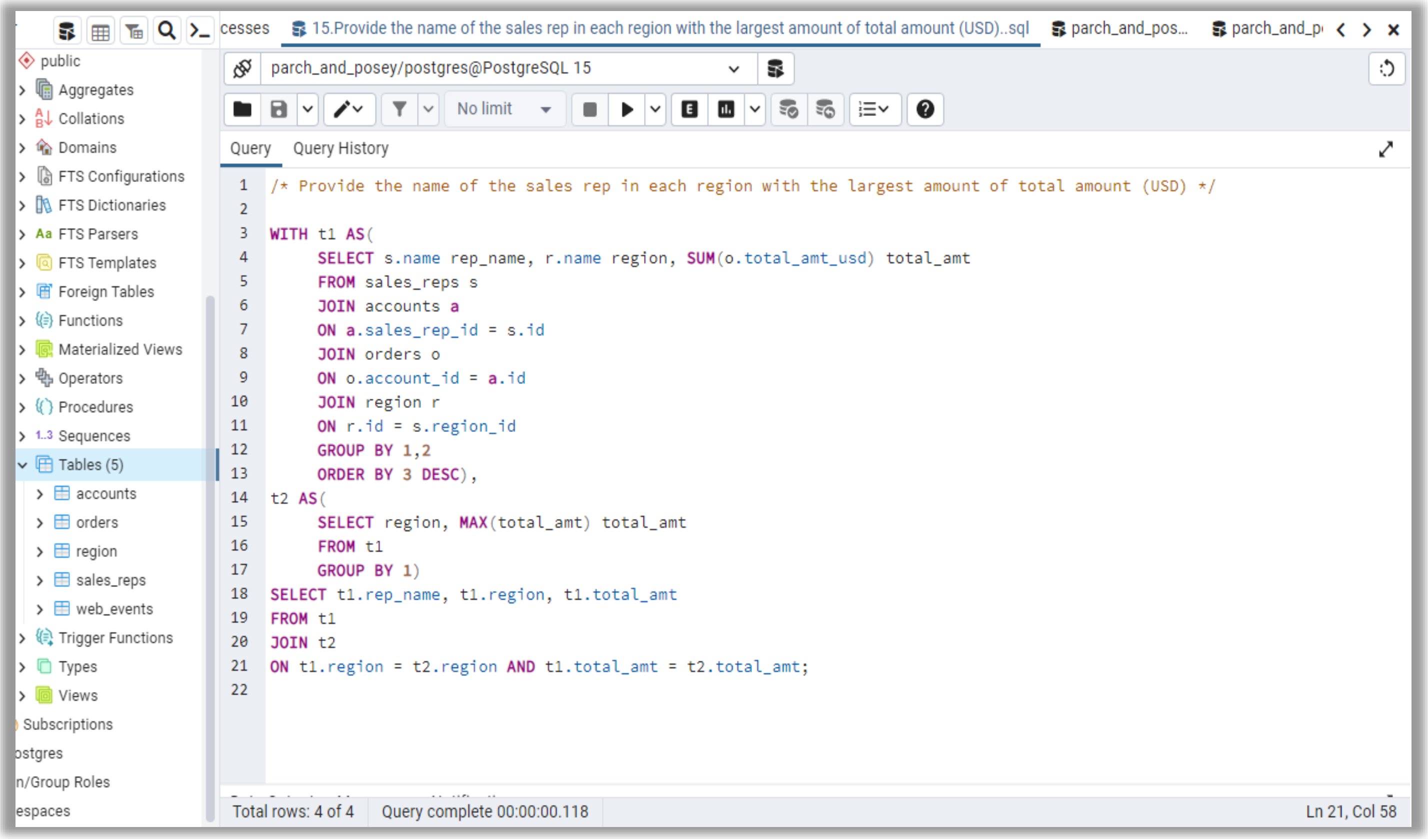
```
/* The customer that spent the most (in total over their lifetime as a customer) in terms of total amount (USD).
How many web events did they have for each channel? */
```

	name character	channel character	num_events bigint
1	EOG Resources	direct	44
2	EOG Resources	organic	13
3	EOG Resources	adwords	12
4	EOG Resources	facebook	11
5	EOG Resources	twitter	5
6	EOG Resources	banner	4

Total rows: 6 of 6 Query complete 00:00:00.235 Ln 17, Col 1

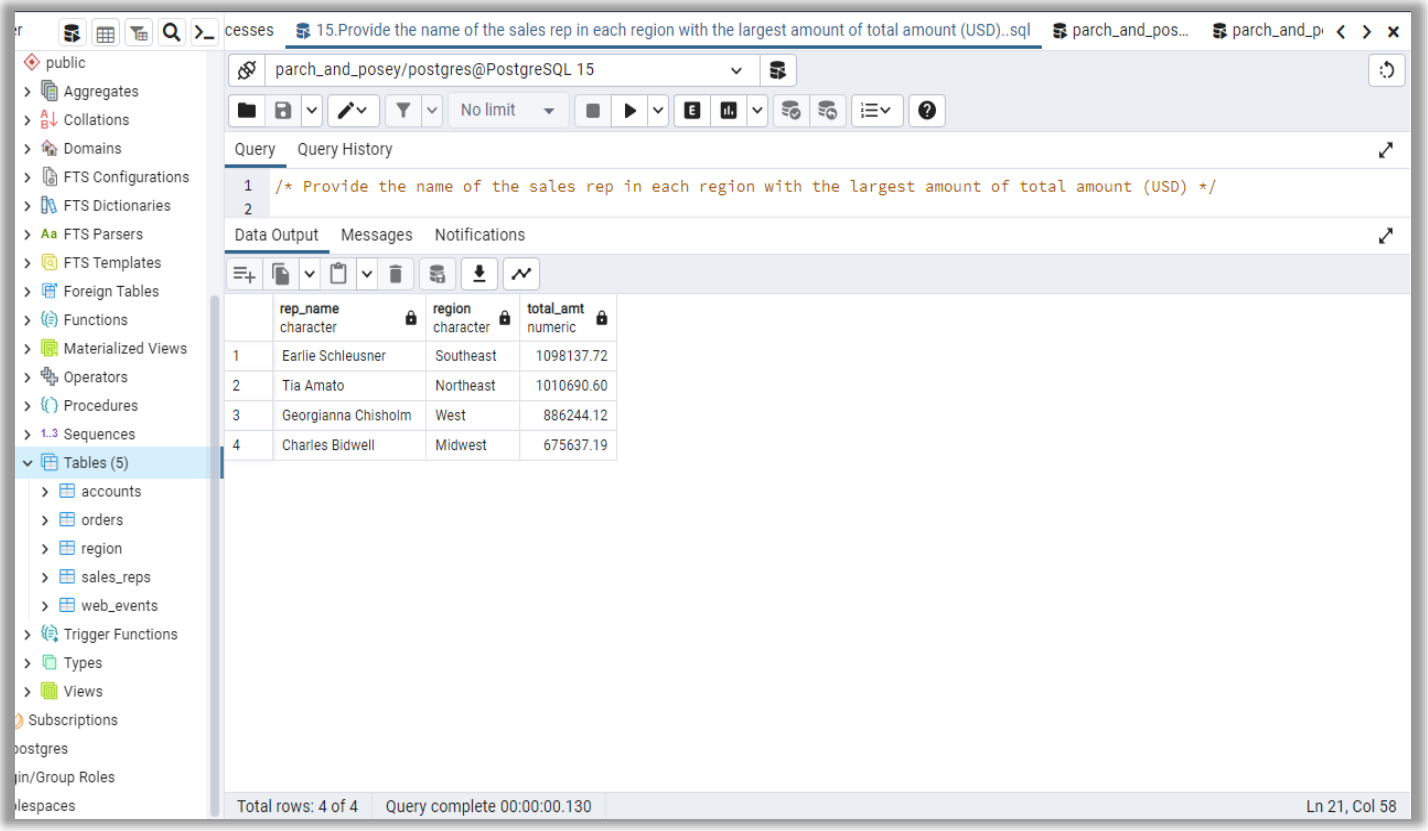
EOG Resources spent the most in total over their lifetime as a customer terms of total amount in us dollars, thy had 89 number of web events all together.

15. Provide the Name of the Sales Rep in Each Region with the Largest Amount of Total Amount (USD):



```
/* Provide the name of the sales rep in each region with the largest amount of total amount (USD) */

WITH t1 AS(
     SELECT s.name rep_name, r.name region, SUM(o.total_amt_usd) total_amt
     FROM sales_reps s
     JOIN accounts a
     ON a.sales_rep_id = s.id
     JOIN orders o
     ON o.account_id = a.id
     JOIN region r
     ON r.id = s.region_id
     GROUP BY 1,2
     ORDER BY 3 DESC),
t2 AS(
     SELECT region, MAX(total_amt) total_amt
     FROM t1
     GROUP BY 1)
SELECT t1.rep_name, t1.region, t1.total_amt
FROM t1
JOIN t2
ON t1.region = t2.region AND t1.total_amt = t2.total_amt;
```

Total rows: 4 of 4 Query complete 00:00:00.118 Ln 21, Col 58

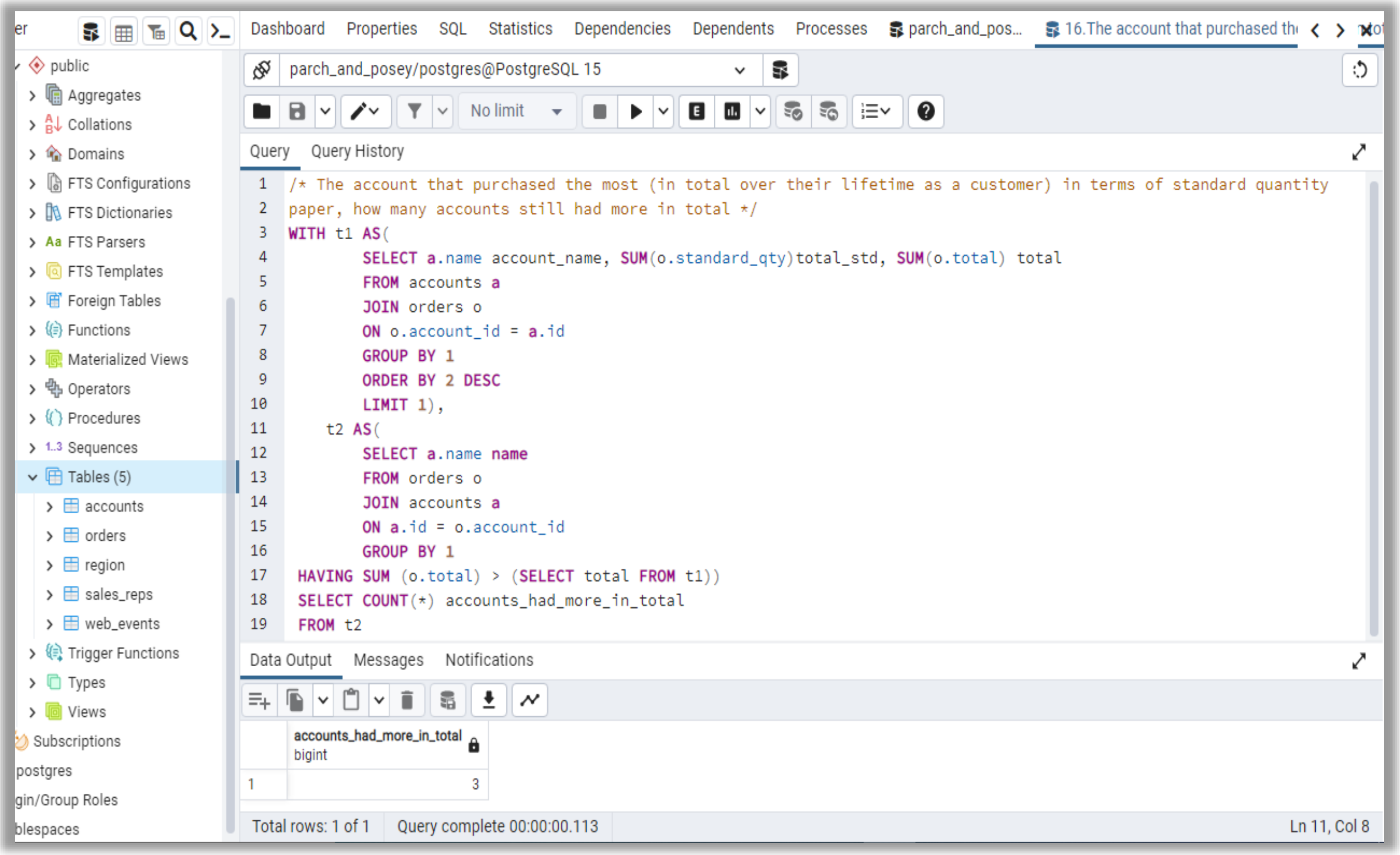


```
/* Provide the name of the sales rep in each region with the largest amount of total amount (USD) */
```

	rep_name character	region character	total_amt numeric
1	Earlie Schleusner	Southeast	1098137.72
2	Tia Amato	Northeast	1010690.60
3	Georgianna Chisholm	West	886244.12
4	Charles Bidwell	Midwest	675637.19

Total rows: 4 of 4 Query complete 00:00:00.130 Ln 21, Col 58

16. The Account that Purchased the Most (In Total Over their Lifetime as a Customer) in Terms of Standard Quantity Paper, How Many Accounts still had More in Total:



```
/* The account that purchased the most (in total over their lifetime as a customer) in terms of standard quantity
paper, how many accounts still had more in total */
WITH t1 AS(
        SELECT a.name account_name, SUM(o.standard_qty)total_std, SUM(o.total) total
        FROM accounts a
        JOIN orders o
        ON o.account_id = a.id
        GROUP BY 1
        ORDER BY 2 DESC
        LIMIT 1),
    t2 AS(
        SELECT a.name name
        FROM orders o
        JOIN accounts a
        ON a.id = o.account_id
        GROUP BY 1
 HAVING SUM (o.total) > (SELECT total FROM t1))
 SELECT COUNT(*) accounts_had_more_in_total
 FROM t2
```

	accounts_had_more_in_total bigint
1	3

Total rows: 1 of 1 Query complete 00:00:00.113 Ln 11, Col 8

In terms of standard paper, 3 accounts still had more in total.

CONCLUSIONS

- In my conclusion, I found out that **Leucadia National”** placed the most orders, and this account placed 71 orders in total.
- I also found that **"Lockheed Martin”** placed the least orders, and this account placed 1 order in total.
- **DISH Network** placed the earliest order in **2013-12-04 04:22:44** and **W.W. Grainger"** placed the latest order in **2017-01-02 00:02:40.**
- **Standard paper** had **1938346** orders, **Post paper** had **723646** orders, **Gloss paper** had **1013773** orders. This analysis showed that **Standard paper** had the highest orders which was one million nine hundred thirty-eight thousand three hundred forty-six.
- I found out that Parch and Posey had their greatest sales in **2016** with over **3757 total sales** and precisely had their greatest sales in the month of **December** with over **882 total sales**
- **Earlie Schleusner** from the **"Southeast"** region had the largest amount of sales **(1098137.72)**
- **Northeast** had the largest amount of total sales **(7744405.36)** in USD and **2357** orders were placed from that region.
- **Midwest** had the smallest amount of total sales **(3013486.51)** in USD and **897** orders were placed from that region
- **EOG Resources** spent the most in total over their lifetime as a customer in terms of total amount in US dollars, they had 89 number of web events all together.
- In terms of standard paper, 3 accounts still had more in total.