SUBQUERIES AND TEMPORARY TABLES

Subqueries allow you to answer more complex questions than you can with a single database table.

We’d like to know which channels send the most traffic per day on average to Parch and Posey.

In order to do that we’ll need to aggregate events by channel, by day and then average them.

SELECT channel,AVG(EVENT\_COUNT) AS avg\_event\_count

FROM

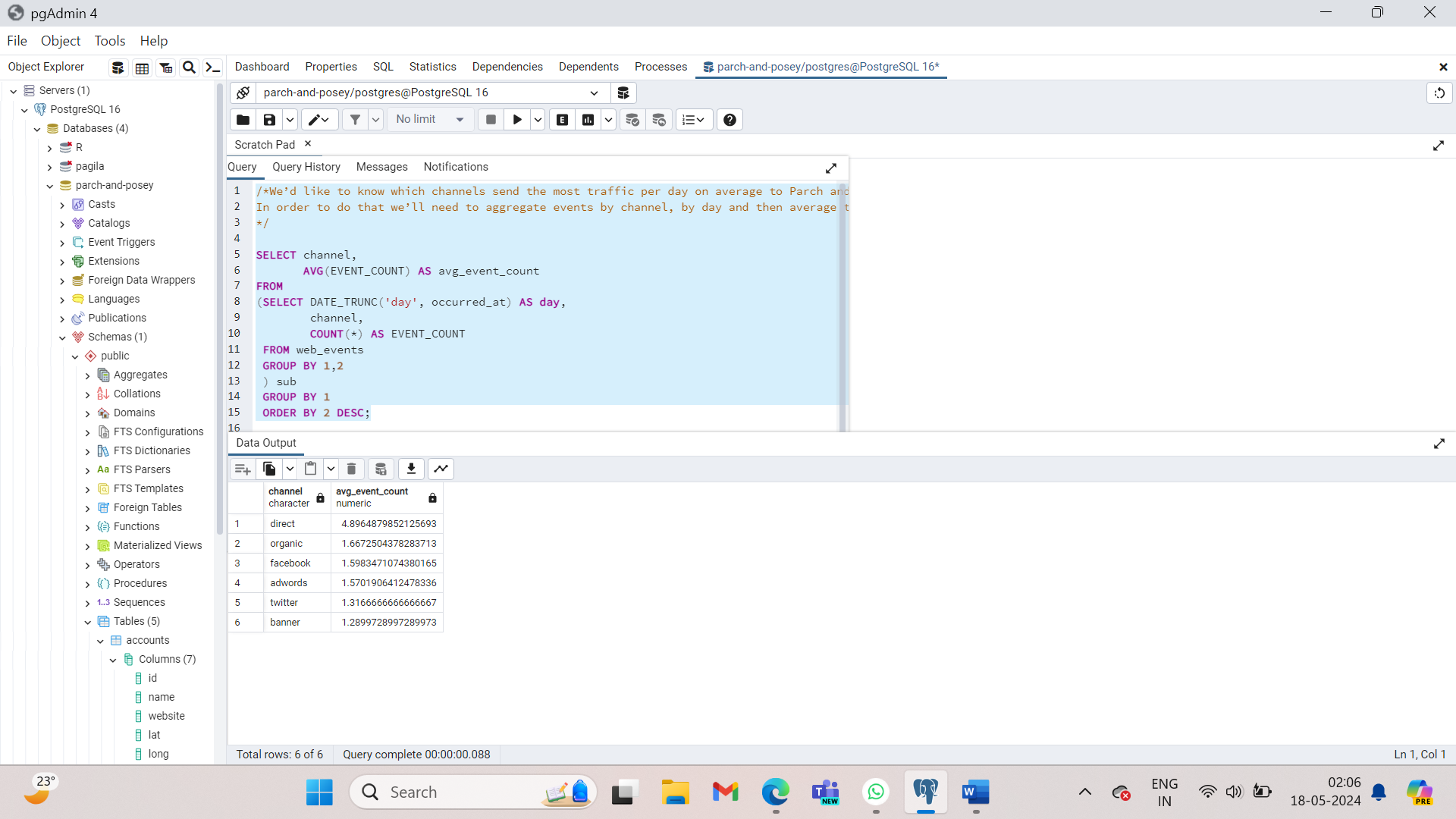
(SELECT DATE\_TRUNC('day', occurred\_at) AS day, channel, COUNT(\*) AS EVENT\_COUNT

FROM web\_events

GROUP BY 1,2) sub

GROUP BY 1

ORDER BY 2 DESC;



SELECT \*

FROM orders

WHERE DATE\_TRUNC('month', occurred\_at) =

(SELECT DATE\_TRUNC('month', MIN(occurred\_at))AS min\_month

FROM orders)

ORDER BY occurred\_at;A screenshot of a computer

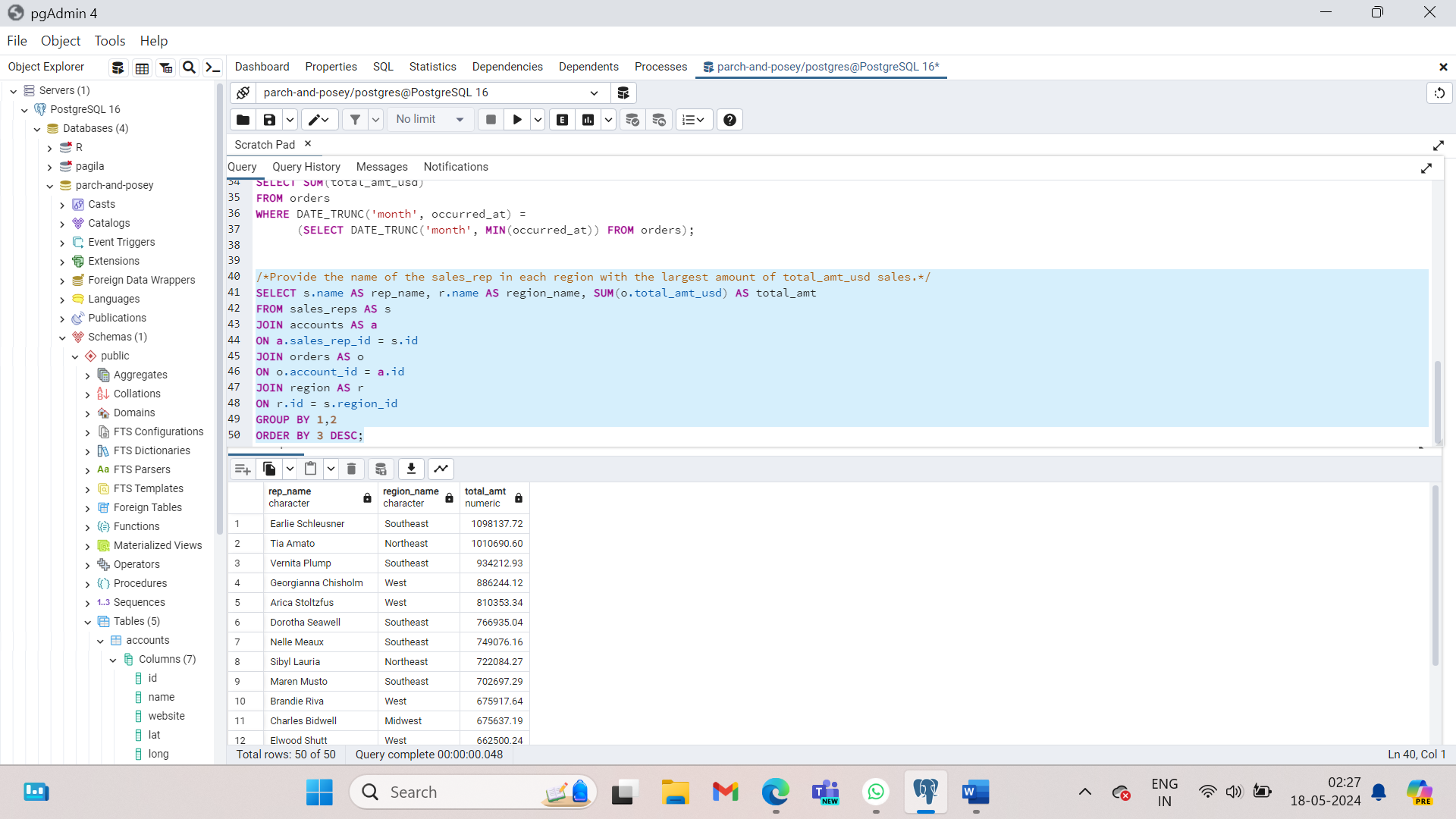
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

/\*We’d like to know which channels send the most traffic per day on average to Parch and Posey.

In order to do that we’ll need to aggregate events by channel, by day and then average them.

\*/

SELECT channel,

AVG(EVENT\_COUNT) AS avg\_event\_count

FROM

(SELECT DATE\_TRUNC('day', occurred\_at) AS day,

channel,

COUNT(\*) AS EVENT\_COUNT

FROM web\_events

GROUP BY 1,2

) sub

GROUP BY 1

ORDER BY 2 DESC;

SELECT \*

FROM orders

WHERE DATE\_TRUNC('month', occurred\_at) =

(SELECT DATE\_TRUNC('month', MIN(occurred\_at))AS min\_month

FROM orders)

ORDER BY occurred\_at;

SELECT AVG(standard\_qty) avg\_std, AVG(gloss\_qty) avg\_gls, AVG(poster\_qty) avg\_pst

FROM orders

WHERE DATE\_TRUNC('month', occurred\_at) =

(SELECT DATE\_TRUNC('month', MIN(occurred\_at)) FROM orders);

SELECT SUM(total\_amt\_usd)

FROM orders

WHERE DATE\_TRUNC('month', occurred\_at) =

(SELECT DATE\_TRUNC('month', MIN(occurred\_at)) FROM orders);

/\*Provide the name of the sales\_rep in each region with the largest amount of total\_amt\_usd sales.\*/

SELECT s.name AS rep\_name, r.name AS region\_name, SUM(o.total\_amt\_usd) AS total\_amt

FROM sales\_reps AS s

JOIN accounts AS a

ON a.sales\_rep\_id = s.id

JOIN orders AS o

ON o.account\_id = a.id

JOIN region AS r

ON r.id = s.region\_id

GROUP BY 1,2

ORDER BY 3 DESC;

/\*Next, pull the max for each region, and then we can use this to pull those rows in our final result.\*/

SELECT region\_name, MAX(total\_amt) as Total\_amt

FROM (SELECT s.name AS rep\_name, r.name AS region\_name, SUM(o.total\_amt\_usd) AS total\_amt

FROM sales\_reps AS s

JOIN accounts AS a

ON a.sales\_rep\_id = s.id

JOIN orders AS o

ON o.account\_id = a.id

JOIN region AS r

ON r.id = s.region\_id

GROUP BY 1,2)t1

GROUP BY 1;

/\*Essentially, this is a JOIN of these two tables, where the region and amount match.\*/

SELECT t3.rep\_name, t3.region\_name, t3.total\_amt

FROM(SELECT region\_name, MAX(total\_amt) total\_amt

FROM(SELECT s.name rep\_name, r.name region\_name, 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) t1

GROUP BY 1) t2

JOIN (SELECT s.name rep\_name, r.name region\_name, 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) t3

ON t3.region\_name = t2.region\_name AND t3.total\_amt = t2.total\_amt;

/\*1 Provide the name of the sales\_rep in each region with the largest amount of total\_amt\_usd sales.

1(a) First, I wanted to find the total\_amt\_usd totals associated with each sales rep, and I also wanted the region in which they were located.

The query below provided this information.

\*/

SELECT s.name rep\_name, r.name region\_name, 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;

/\* 1(b) Next, I pulled the max for each region, and then we can use this to pull those rows in our final result.\*/

SELECT region\_name, MAX(total\_amt) total\_amt

FROM(SELECT s.name rep\_name, r.name region\_name, 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) t1

GROUP BY 1;

/\* 1(c) Essentially, this is a JOIN of these two tables, where the region and amount match.\*/

SELECT t3.rep\_name, t3.region\_name, t3.total\_amt

FROM(SELECT region\_name, MAX(total\_amt) total\_amt

FROM(SELECT s.name rep\_name, r.name region\_name, 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) t1

GROUP BY 1) t2

JOIN (SELECT s.name rep\_name, r.name region\_name, 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) t3

ON t3.region\_name = t2.region\_name AND t3.total\_amt = t2.total\_amt;

/\* 2. For the region with the largest sales total\_amt\_usd, how many total orders were placed? \*/

SELECT r.name region\_name, 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 r.name;

/\* Then we just want the region with the max amount from this table. There are two ways I considered getting this amount. One was to pull the max using a subquery.

Another way is to order descending and just pull the top value.\*/

SELECT MAX(total\_amt)

FROM (SELECT r.name region\_name, 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 r.name) sub;

/\* Finally, we want to pull the total orders for the region with this amount:\*/

SELECT r.name, COUNT(o.total) total\_orders

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 r.name

HAVING SUM(o.total\_amt\_usd) = (

SELECT MAX(total\_amt)

FROM (SELECT r.name region\_name, 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 r.name) sub);

/\* 3. How many accounts had more total purchases than the account name which has bought the most

standard\_qty paper throughout their lifetime as a customer?

First, we want to find the account that had the most standard\_qty paper. The query here pulls that account, as well as the total amount:\*/

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;

/\* Now, I want to use this to pull all the accounts with more total sales:\*/

SELECT a.name

FROM orders o

JOIN accounts a

ON a.id = o.account\_id

GROUP BY 1

HAVING SUM(o.total) > (SELECT total

FROM (SELECT a.name act\_name, SUM(o.standard\_qty) tot\_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) sub);

/\* This is now a list of all the accounts with more total orders. We can get the count with just another simple subquery.\*/

SELECT COUNT(\*)

FROM (SELECT a.name

FROM orders o

JOIN accounts a

ON a.id = o.account\_id

GROUP BY 1

HAVING SUM(o.total) > (SELECT total

FROM (SELECT a.name act\_name, SUM(o.standard\_qty) tot\_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) inner\_tab)

) counter\_tab;

/\* 4. For the customer that spent the most (in total over their lifetime as a customer) total\_amt\_usd, how many web\_events did they have for each channel?

Here, we first want to pull the customer with the most spent in lifetime value.\*/

SELECT a.id, a.name, SUM(o.total\_amt\_usd) tot\_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;

/\* Now, we want to look at the number of events on each channel this company had, which we can match with just the id.\*/

SELECT a.name, w.channel, COUNT(\*)

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) tot\_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;

/\* 5. What is the lifetime average amount spent in terms of total\_amt\_usd for the top 10 total spending accounts?

First, we just want to find the top 10 accounts in terms of highest total\_amt\_usd.\*/

SELECT a.id, a.name, SUM(o.total\_amt\_usd) tot\_spent

FROM orders o

JOIN accounts a

ON a.id = o.account\_id

GROUP BY a.id, a.name

ORDER BY 3 DESC

LIMIT 10;

/\* Now, we just want the average of these 10 amounts.\*/

SELECT AVG(tot\_spent)

FROM (SELECT a.id, a.name, SUM(o.total\_amt\_usd) tot\_spent

FROM orders o

JOIN accounts a

ON a.id = o.account\_id

GROUP BY a.id, a.name

ORDER BY 3 DESC

LIMIT 10) temp;

/\*6. What is the lifetime average amount spent in terms of \*\*total\_amt\_usd\*\*, including only the companies

that spent more per order, on average, than the average of all orders.

First, we want to pull the average of all accounts in terms of \*\*total\_amt\_usd\*\*: \*/

SELECT AVG(o.total\_amt\_usd) avg\_all

FROM orders o

/\* Then, we want to only pull the accounts with more than this average amount.\*/

SELECT o.account\_id, AVG(o.total\_amt\_usd)

FROM orders o

GROUP BY 1

HAVING AVG(o.total\_amt\_usd) >

(SELECT AVG(o.total\_amt\_usd) avg\_all

FROM orders o);

/\*Finally, we just want the average of these values.\*/

SELECT AVG(avg\_amt)

FROM (SELECT o.account\_id, AVG(o.total\_amt\_usd) avg\_amt

FROM orders o

GROUP BY 1

HAVING AVG(o.total\_amt\_usd) >

(SELECT AVG(o.total\_amt\_usd) avg\_all

FROM orders o)) temp\_table;

The **WITH** statement is often called a **Common Table Expression or CTE**. Though these expressions serve the exact same purpose as subqueries, they are more common in practice, as they tend to be cleaner for a future reader to follow the logic.

WITH events AS (SELECT DATE\_TRUNC(‘day’, occurred\_at) AS day,

Channel,

COUNT(\*) AS event\_count

FROM web\_events

GROUP BY 1,2)

SELECT channel,

AVG(event\_count) AS avg\_event\_count

FROM events

GROUP BY 1

ORDER BY 2 DESC;

/\* WITH Solutions

Below, you will see each of the previous solutions restructured using the WITH clause. This is often an easier way to read a query.

Provide the name of the sales\_rep in each region with the largest amount of total\_amt\_usd sales.\*/

WITH t1 AS (

SELECT s.name rep\_name, r.name region\_name, 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\_name, MAX(total\_amt) total\_amt

FROM t1

GROUP BY 1)

SELECT t1.rep\_name, t1.region\_name, t1.total\_amt

FROM t1

JOIN t2

ON t1.region\_name = t2.region\_name AND t1.total\_amt = t2.total\_amt;

/\*For the region with the largest sales total\_amt\_usd, how many total orders were placed?\*/

WITH t1 AS (

SELECT r.name region\_name, 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 r.name),

t2 AS (

SELECT MAX(total\_amt)

FROM t1)

SELECT r.name, COUNT(o.total) total\_orders

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 r.name

HAVING SUM(o.total\_amt\_usd) = (SELECT \* FROM t2);

/\*For the account that purchased the most (in total over their lifetime as a customer) standard\_qty paper, how many accounts still had more in total purchases?\*/

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

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(\*)

FROM t2;

/\*For the customer that spent the most (in total over their lifetime as a customer) total\_amt\_usd, how many web\_events did they have for each channel?\*/

WITH t1 AS (

SELECT a.id, a.name, SUM(o.total\_amt\_usd) tot\_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)

SELECT a.name, w.channel, COUNT(\*)

FROM accounts a

JOIN web\_events w

ON a.id = w.account\_id AND a.id = (SELECT id FROM t1)

GROUP BY 1, 2

ORDER BY 3 DESC;

/\*What is the lifetime average amount spent in terms of total\_amt\_usd for the top 10 total spending accounts?\*/

WITH t1 AS (

SELECT a.id, a.name, SUM(o.total\_amt\_usd) tot\_spent

FROM orders o

JOIN accounts a

ON a.id = o.account\_id

GROUP BY a.id, a.name

ORDER BY 3 DESC

LIMIT 10)

SELECT AVG(tot\_spent)

FROM t1;

/\* 6. What is the lifetime average amount spent in terms of \*\*total\_amt\_usd\*\*, including only the companies that spent more per order, on average,

than the average of all orders.\*/

WITH t1 AS (

SELECT AVG(o.total\_amt\_usd) avg\_all

FROM orders o

JOIN accounts a

ON a.id = o.account\_id),

t2 AS (

SELECT o.account\_id,

AVG(o.total\_amt\_usd) avg\_amt

FROM orders o

GROUP BY 1

HAVING AVG(o.total\_amt\_usd) > (SELECT \* FROM t1))

SELECT AVG(avg\_amt) FROM t2;