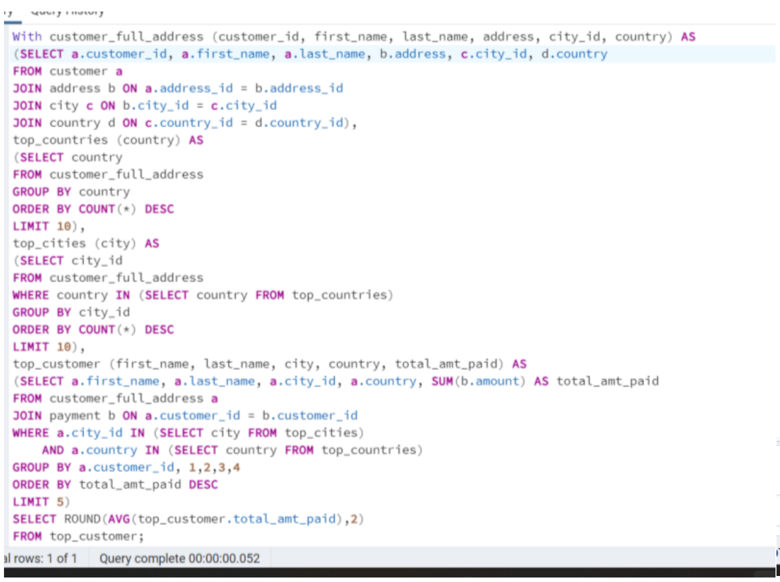
Answers 3.9 Common Table Expressions

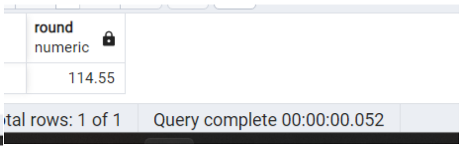
Step 1: Answer the business questions form step 1 and 2 of task 3.8 using CTEs

- Average amount paid by top 5 customers (from top-10 countries, from top-10 cities)
    - This CTE starts with a large, joined table of customer information including address, city, and country.

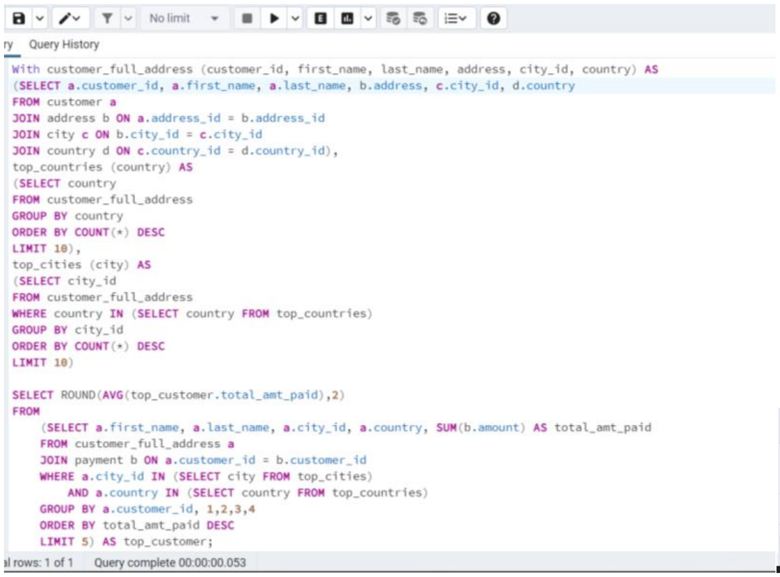


```
With customer_full_address (customer_id, first_name, last_name, address, city_id, country) AS
(SELECT a.customer_id, a.first_name, a.last_name, b.address, c.city_id, d.country
FROM customer a
JOIN address b ON a.address_id = b.address_id
JOIN city c ON b.city_id = c.city_id
JOIN country d ON c.country_id = d.country_id),
top_countries (country) AS
(SELECT country
FROM customer_full_address
GROUP BY country
ORDER BY COUNT(*) DESC
LIMIT 10),
top_cities (city) AS
(SELECT city_id
FROM customer_full_address
WHERE country IN (SELECT country FROM top_countries)
GROUP BY city_id
ORDER BY COUNT(*) DESC
LIMIT 10),
top_customer (first_name, last_name, city, country, total_amt_paid) AS
(SELECT a.first_name, a.last_name, a.city_id, a.country, SUM(b.amount) AS total_amt_paid
FROM customer_full_address a
JOIN payment b ON a.customer_id = b.customer_id
WHERE a.city_id IN (SELECT city FROM top_cities)
    AND a.country IN (SELECT country FROM top_countries)
GROUP BY a.customer_id, 1,2,3,4
ORDER BY total_amt_paid DESC
LIMIT 5)
SELECT ROUND(AVG(top_customer.total_amt_paid),2)
FROM top_customer;
```

```
al rows: 1 of 1    Query complete 00:00:00.052
```



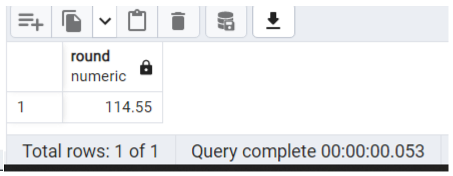
```
round
numeric
    114.55
tal rows: 1 of 1    Query complete 00:00:00.052
```



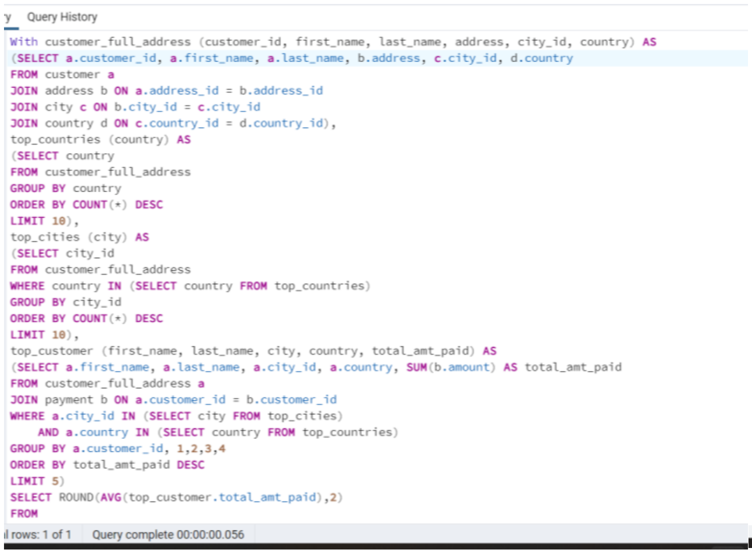
```
With customer_full_address (customer_id, first_name, last_name, address, city_id, country) AS
(SELECT a.customer_id, a.first_name, a.last_name, b.address, c.city_id, d.country
FROM customer a
JOIN address b ON a.address_id = b.address_id
JOIN city c ON b.city_id = c.city_id
JOIN country d ON c.country_id = d.country_id),
top_countries (country) AS
(SELECT country
FROM customer_full_address
GROUP BY country
ORDER BY COUNT(*) DESC
LIMIT 10),
top_cities (city) AS
(SELECT city_id
FROM customer_full_address
WHERE country IN (SELECT country FROM top_countries)
GROUP BY city_id
ORDER BY COUNT(*) DESC
LIMIT 10)

SELECT ROUND(AVG(top_customer.total_amt_paid),2)
FROM
    (SELECT a.first_name, a.last_name, a.city_id, a.country, SUM(b.amount) AS total_amt_paid
    FROM customer_full_address a
    JOIN payment b ON a.customer_id = b.customer_id
    WHERE a.city_id IN (SELECT city FROM top_cities)
        AND a.country IN (SELECT country FROM top_countries)
    GROUP BY a.customer_id, 1,2,3,4
    ORDER BY total_amt_paid DESC
    LIMIT 5) AS top_customer;
```

```
al rows: 1 of 1    Query complete 00:00:00.053
```



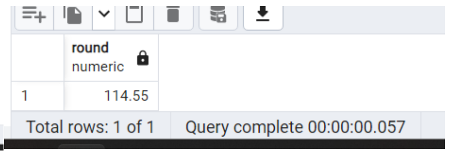
```
   round
   numeric
1     114.55
Total rows: 1 of 1    Query complete 00:00:00.053
```



y Query History

```
With customer_full_address (customer_id, first_name, last_name, address, city_id, country) AS
(SELECT a.customer_id, a.first_name, a.last_name, b.address, c.city_id, d.country
FROM customer a
JOIN address b ON a.address_id = b.address_id
JOIN city c ON b.city_id = c.city_id
JOIN country d ON c.country_id = d.country_id),
top_countries (country) AS
(SELECT country
FROM customer_full_address
GROUP BY country
ORDER BY COUNT(*) DESC
LIMIT 10),
top_cities (city) AS
(SELECT city_id
FROM customer_full_address
WHERE country IN (SELECT country FROM top_countries)
GROUP BY city_id
ORDER BY COUNT(*) DESC
LIMIT 10),
top_customer (first_name, last_name, city, country, total_amt_paid) AS
(SELECT a.first_name, a.last_name, a.city_id, a.country, SUM(b.amount) AS total_amt_paid
FROM customer_full_address a
JOIN payment b ON a.customer_id = b.customer_id
WHERE a.city_id IN (SELECT city FROM top_cities)
    AND a.country IN (SELECT country FROM top_countries)
GROUP BY a.customer_id, 1,2,3,4
ORDER BY total_amt_paid DESC
LIMIT 5)
SELECT ROUND(AVG(top_customer.total_amt_paid),2)
FROM
```

l rows: 1 of 1   Query complete 00:00:00.056



```
   round
   numeric
1  114.55
Total rows: 1 of 1    Query complete 00:00:00.057
```

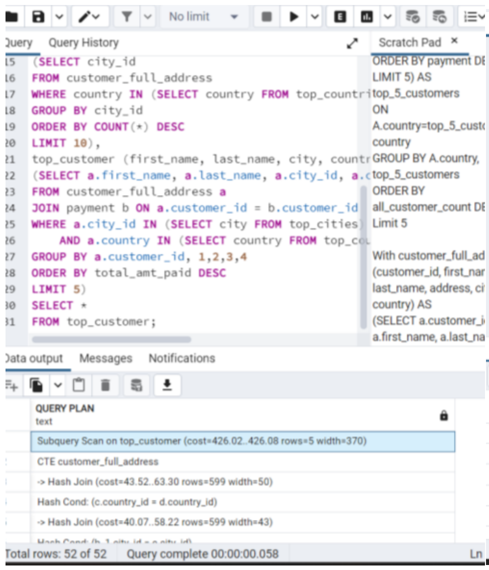
They all came up with the same result (114.55), but you can't see the Query complete time increased. And the cost of adding the payment info at the very beginning proved to be higher that the EXPLAIN

Count of the top 5 customers based in each country

- I used a third iteration, joining payment information at the beginning to simplify the main query and increase readability.
- I also used LEFT JOIN to link the top 5 customers to the list of all customers and the aggregated country, counting the customers in the whole set and in the top 5.

Step 2: Compare the proformance of your CTEs and subqueries.

- Average amount paid by top 5 customers (from top 10 countries, from top 10 cities)



Query   Query History   Scratch Pad

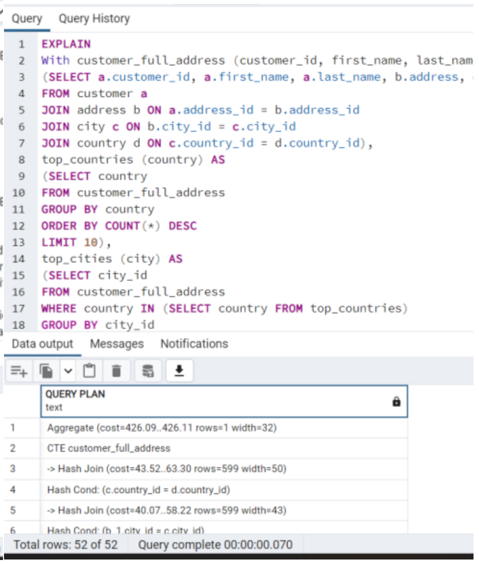
```
(SELECT city_id
FROM customer_full_address
WHERE country IN (SELECT country FROM top_countri
GROUP BY city_id
ORDER BY COUNT(*) DESC
LIMIT 10),
top_customer (first_name, last_name, city, countr
(SELECT a.first_name, a.last_name, a.city_id, a.c
FROM customer_full_address a
JOIN payment b ON a.customer_id = b.customer_id
WHERE a.city_id IN (SELECT city FROM top_cities
    AND a.country IN (SELECT country FROM top_co
GROUP BY a.customer_id, 1,2,3,4
ORDER BY total_amt_paid DESC
LIMIT 5)
SELECT *
FROM top_customer;
```

```
ORDER BY payment DE
LIMIT 5) AS
top_5_customers
ON
A.country=top_5_custo
country
GROUP BY A.country,
top_5_customers
ORDER BY
all_customer_count DE
Limit 5

With customer_full_ad
(customer_id, first_nar
last_name, address, ci
country) AS
(SELECT a.customer_i
a.first_name, a.last_na
```

Data output   Messages   Notifications

```
QUERY PLAN
text
Subquery Scan on top_customer (cost=426.02..426.08 rows=5 width=370)
CTE customer_full_address
-> Hash Join (cost=43.52..63.30 rows=599 width=50)
Hash Cond: (c.country_id = d.country_id)
-> Hash Join (cost=40.07..58.22 rows=599 width=43)
Total rows: 52 of 52    Query complete 00:00:00.058
```

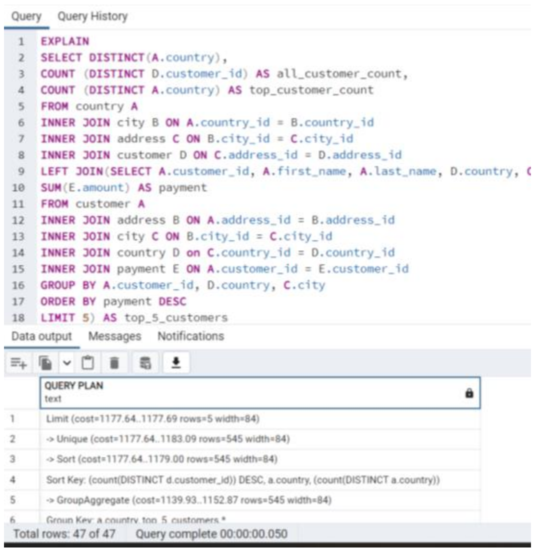


Query   Query History

```
EXPLAIN
With customer_full_address (customer_id, first_name, last_nam
(SELECT a.customer_id, a.first_name, a.last_name, b.address,
FROM customer a
JOIN address b ON a.address_id = b.address_id
JOIN city c ON b.city_id = c.city_id
JOIN country d ON c.country_id = d.country_id),
top_countries (country) AS
(SELECT country
FROM customer_full_address
GROUP BY country
ORDER BY COUNT(*) DESC
LIMIT 10),
top_cities (city) AS
(SELECT city_id
FROM customer_full_address
WHERE country IN (SELECT country FROM top_countries)
GROUP BY city_id
```

Data output   Messages   Notifications

```
   QUERY PLAN
   text
1  Aggregate (cost=426.09..426.11 rows=1 width=32)
2  CTE customer_full_address
3  -> Hash Join (cost=43.52..63.30 rows=599 width=50)
4  Hash Cond: (c.country_id = d.country_id)
5  -> Hash Join (cost=40.07..58.22 rows=599 width=43)
6  Hash Cond: (b_1.city_id = c.city_id)
Total rows: 52 of 52    Query complete 00:00:00.070
```



```sql
EXPLAIN
SELECT DISTINCT(A.country),
COUNT (DISTINCT D.customer_id) AS all_customer_count,
COUNT (DISTINCT A.country) AS top_customer_count
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
LEFT JOIN(SELECT A.customer_id, A.first_name, A.last_name, D.country,
SUM(E.amount) AS payment
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D on C.country_id = D.country_id
INNER JOIN payment E ON A.customer_id = E.customer_id
GROUP BY A.customer_id, D.country, C.city
ORDER BY payment DESC
LIMIT 5) AS top_5_customers
```

```
   QUERY PLAN
   text
1  Limit (cost=1177.64..1177.69 rows=5 width=84)
2  -> Unique (cost=1177.64..1183.09 rows=545 width=84)
3  -> Sort (cost=1177.64..1179.00 rows=545 width=84)
4  Sort Key: (count(DISTINCT d.customer_id)) DESC, a.country, (count(DISTINCT a.country))
5  -> GroupAggregate (cost=1139.93..1152.87 rows=545 width=84)
6  Group Key: a.country, top_5_customers.*
Total rows: 47 of 47    Query complete 00:00:00.050
```

It doesn't surprise me that joining the payment table combined with customer data increased the cost. When joining the customer table with city and country only 2 columns were added. Whereas the payment table added to customer table becomes much larger. Doing an aggregation with a larger tale cost more.

Step 3:

I wasn't sure if I wrote multiple CTEs if they would relate to one another. Meaning if they would work as two different commands or if the second one would refer to the first one. It goes to show the power of , ; punctuation. I also tried putting subqueries into CTEs and it worked. Similar to subqueries, it's unnecessary to use ORDEY BY unless you use a LIMIT, which is best for finding top and bottom records.