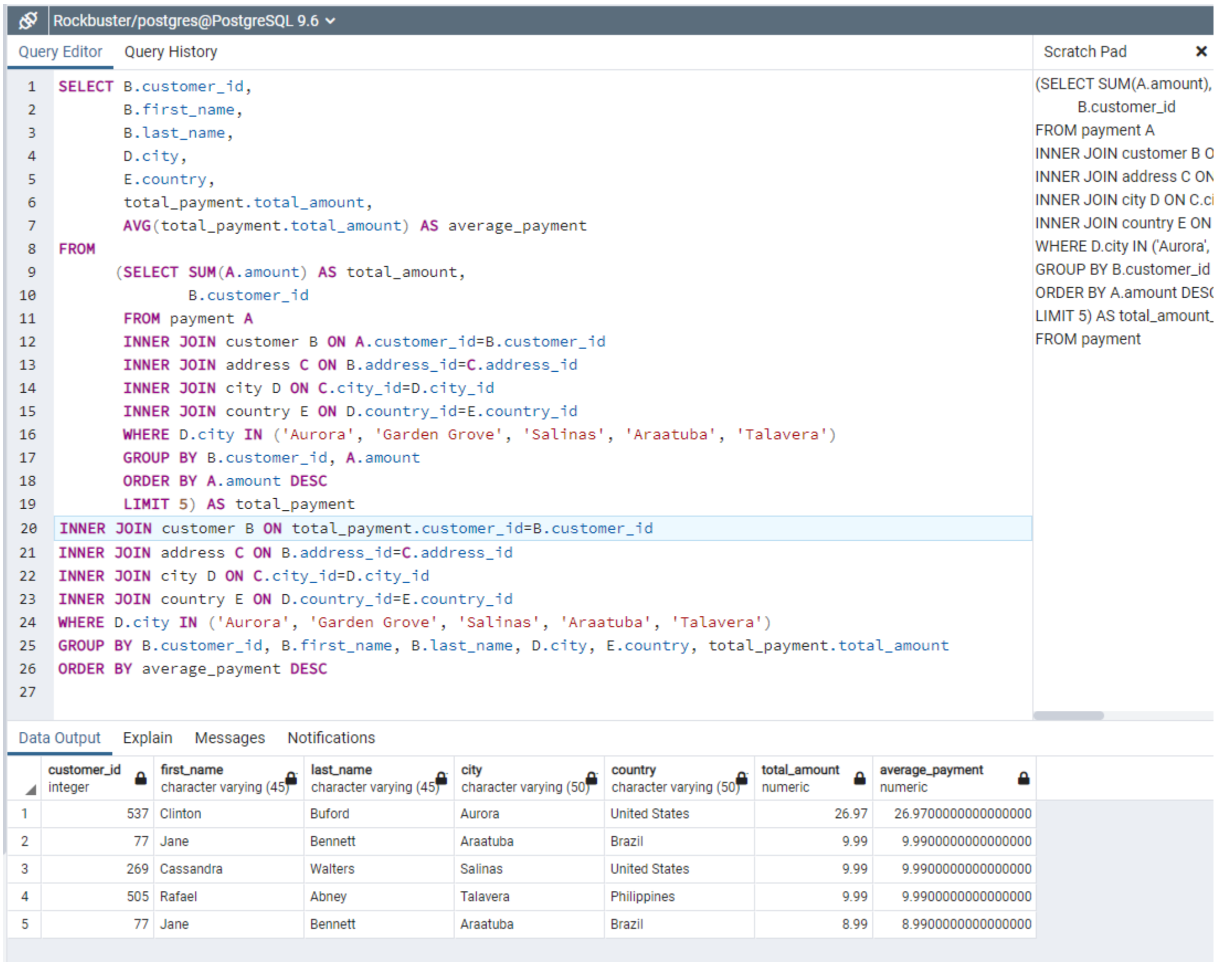
Alejandra Castro
30012025

3.8 Subqueries

1. **Find the average amount paid by the top 5 customers.**



```
SELECT B.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       total_payment.total_amount,
       AVG(total_payment.total_amount) AS average_payment
FROM
      (SELECT SUM(A.amount) AS total_amount,
              B.customer_id
       FROM payment A
       INNER JOIN customer B ON A.customer_id=B.customer_id
       INNER JOIN address C ON B.address_id=C.address_id
       INNER JOIN city D ON C.city_id=D.city_id
       INNER JOIN country E ON D.country_id=E.country_id
       WHERE D.city IN ('Aurora', 'Garden Grove', 'Salinas', 'Araatuba', 'Talavera')
       GROUP BY B.customer_id, A.amount
       ORDER BY A.amount DESC
       LIMIT 5) AS total_payment
INNER JOIN customer B ON total_payment.customer_id=B.customer_id
INNER JOIN address C ON B.address_id=C.address_id
INNER JOIN city D ON C.city_id=D.city_id
INNER JOIN country E ON D.country_id=E.country_id
WHERE D.city IN ('Aurora', 'Garden Grove', 'Salinas', 'Araatuba', 'Talavera')
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country, total_payment.total_amount
ORDER BY average_payment DESC
```

```
(SELECT SUM(A.amount),
      B.customer_id
FROM payment A
INNER JOIN customer B O
INNER JOIN address C ON
INNER JOIN city D ON C.ci
INNER JOIN country E ON
WHERE D.city IN ('Aurora',
GROUP BY B.customer_id
ORDER BY A.amount DES(
LIMIT 5) AS total_amount_
FROM payment
```

	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_amount numeric	average_payment numeric
1	537	Clinton	Buford	Aurora	United States	26.97	26.9700000000000000
2	77	Jane	Bennett	Araatuba	Brazil	9.99	9.9900000000000000
3	269	Cassandra	Walters	Salinas	United States	9.99	9.9900000000000000
4	505	Rafael	Abney	Talavera	Philippines	9.99	9.9900000000000000
5	77	Jane	Bennett	Araatuba	Brazil	8.99	8.9900000000000000

Script:

SELECT B.customer_id,

B.first_name,

B.last_name,

D.city,

E.country,

total_payment.total_amount,

AVG(total_payment.total_amount) AS average_payment

FROM

(SELECT SUM(A.amount) AS total_amount,

B.customer_id

FROM payment A

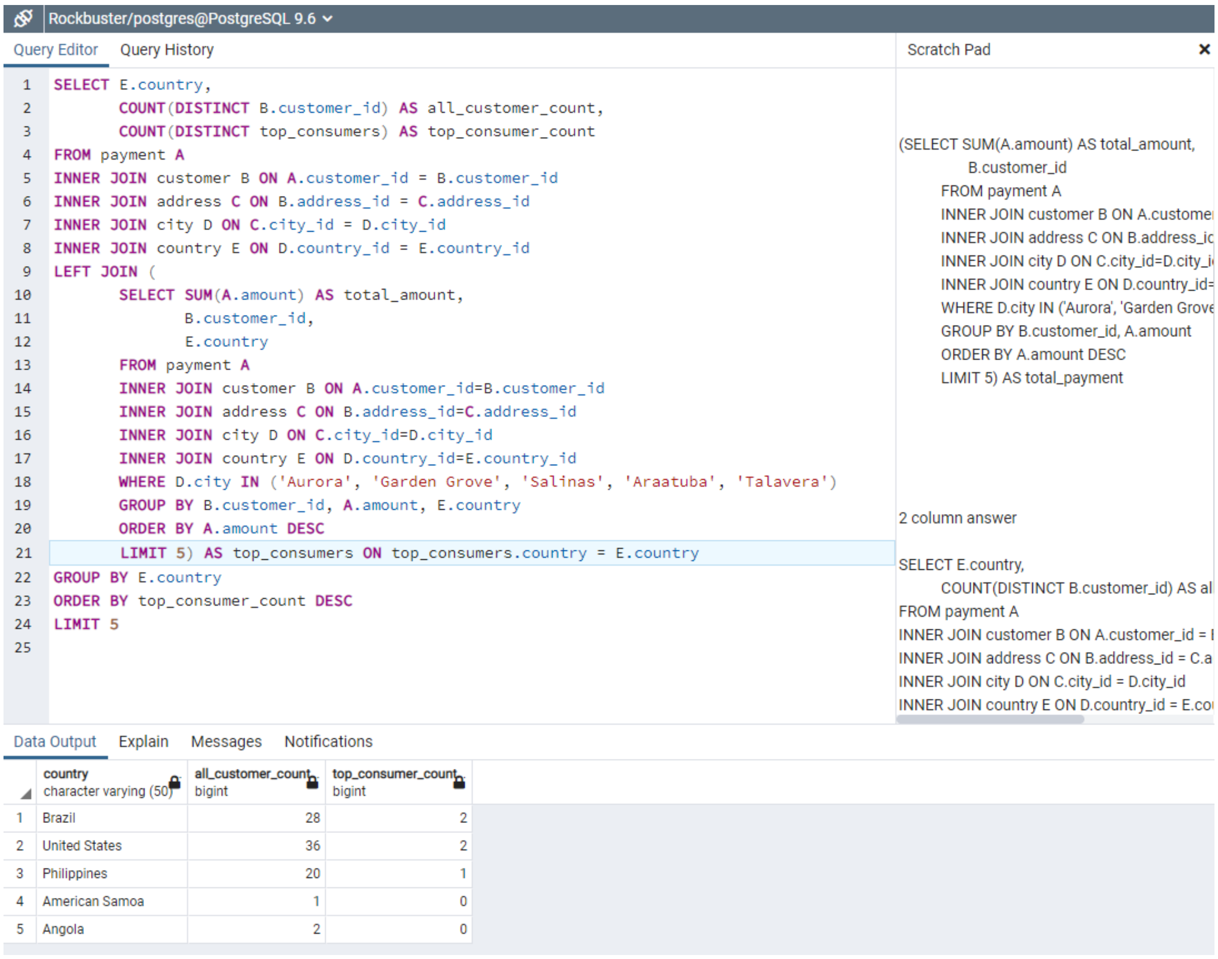
Alejandra Castro
30012025

```
		INNER JOIN customer B ON A.customer_id=B.customer_id
		INNER JOIN address C ON B.address_id=C.address_id
		INNER JOIN city D ON C.city_id=D.city_id
		INNER JOIN country E ON D.country_id=E.country_id
		WHERE D.city IN ('Aurora', 'Garden Grove', 'Salinas', 'Araatuba', 'Talavera')
		GROUP BY B.customer_id, A.amount
		ORDER BY A.amount DESC
		LIMIT 5) AS total_payment
INNER JOIN customer B ON total_payment.customer_id=B.customer_id
INNER JOIN address C ON B.address_id=C.address_id
INNER JOIN city D ON C.city_id=D.city_id
INNER JOIN country E ON D.country_id=E.country_id
WHERE D.city IN ('Aurora', 'Garden Grove', 'Salinas', 'Araatuba', 'Talavera')
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country,
total_payment.total_amount
ORDER BY average_payment DESC
```

NOTES: 1) It did not ask me for an alias. 2) I cannot figure out why it's not giving me the same customers as it gave me in Exercise 3.7

2. **Find out how many of the top 5 customers you identified in step 1 are based within each country.**

Alejandra Castro
30012025



Rockbuster/postgres@PostgreSQL 9.6

Query Editor | Query History

```
SELECT E.country,
       COUNT(DISTINCT B.customer_id) AS all_customer_count,
       COUNT(DISTINCT top_consumers) AS top_consumer_count
FROM payment A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.country_id
LEFT JOIN (
       SELECT SUM(A.amount) AS total_amount,
              B.customer_id,
              E.country
       FROM payment A
       INNER JOIN customer B ON A.customer_id=B.customer_id
       INNER JOIN address C ON B.address_id=C.address_id
       INNER JOIN city D ON C.city_id=D.city_id
       INNER JOIN country E ON D.country_id=E.country_id
       WHERE D.city IN ('Aurora', 'Garden Grove', 'Salinas', 'Araatuba', 'Talavera')
       GROUP BY B.customer_id, A.amount, E.country
       ORDER BY A.amount DESC
       LIMIT 5) AS top_consumers ON top_consumers.country = E.country
GROUP BY E.country
ORDER BY top_consumer_count DESC
LIMIT 5
```

Scratch Pad

```
(SELECT SUM(A.amount) AS total_amount,
        B.customer_id
      FROM payment A
      INNER JOIN customer B ON A.custome
      INNER JOIN address C ON B.address_ic
      INNER JOIN city D ON C.city_id=D.city_i
      INNER JOIN country E ON D.country_id=
      WHERE D.city IN ('Aurora', 'Garden Grove
      GROUP BY B.customer_id, A.amount
      ORDER BY A.amount DESC
      LIMIT 5) AS total_payment



2 column answer

SELECT E.country,
      COUNT(DISTINCT B.customer_id) AS al
FROM payment A
INNER JOIN customer B ON A.customer_id = I
INNER JOIN address C ON B.address_id = C.a
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.co
```

Data Output | Explain | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_consumer_count bigint
1	Brazil	28	2
2	United States	36	2
3	Philippines	20	1
4	American Samoa	1	0
5	Angola	2	0

Script:

SELECT E.country,

COUNT(DISTINCT B.customer_id) AS all_customer_count,

COUNT(DISTINCT top_consumers) AS top_consumer_count

FROM payment A

INNER JOIN customer B ON A.customer_id = B.customer_id

INNER JOIN address C ON B.address_id = C.address_id

INNER JOIN city D ON C.city_id = D.city_id

INNER JOIN country E ON D.country_id = E.country_id

LEFT JOIN (

SELECT SUM(A.amount) AS total_amount,

B.customer_id,

E.country

FROM payment A

```
        INNER JOIN customer B ON A.customer_id=B.customer_id
        INNER JOIN address C ON B.address_id=C.address_id
        INNER JOIN city D ON C.city_id=D.city_id
        INNER JOIN country E ON D.country_id=E.country_id
        WHERE D.city IN ('Aurora', 'Garden Grove', 'Salinas', 'Araatuba', 'Talavera')
        GROUP BY B.customer_id, A.amount, E.country
        ORDER BY A.amount DESC
        LIMIT 5) AS top_consumers ON top_consumers.country = E.country
GROUP BY E.country
ORDER BY top_consumer_count DESC
LIMIT 5
```

3. Write 1 to 2 short paragraphs
    a. Do you think steps 1 and 2 could be done without using subqueries?
    b. When do you think subqueries are useful?

I think they can be done without using subqueries, but this requires to use alternate methods that may be less effective in other circumstances. For example, I already knew the answer in step 2 before finishing the whole script because first I ran the outer query by itself:

```
SELECT E.country,
        COUNT(DISTINCT B.customer_id) AS all_customer_count
FROM payment A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_id = E.country_id
GROUP BY E.country
```

And with this, I already had arrived at the complete answer because the join showed the total count while I already knew that the top customers were in Brazil, the Philippines, and the US because of the results in step 1. However, this could not have been done with a larger dataset/a longer list of results.

Alejandra Castro
30012025

In terms of optimization, both steps could have been replaced by a CTE because it has better readability and maintainability in the long term, especially with longer scripts, whereas subqueries are a good solution when the task at hand is relatively simple.

Because of how costly they are, I think subqueries should only be used when it’s really justified, given that they can be replaced by other operations (like joins, CTEs, or even aggregated functions by themselves. They’re useful when the task to be done is simple and when you need to interact with tables that are constantly being modified. Otherwise, the script might become too nested and complex, making it more difficult to debug.