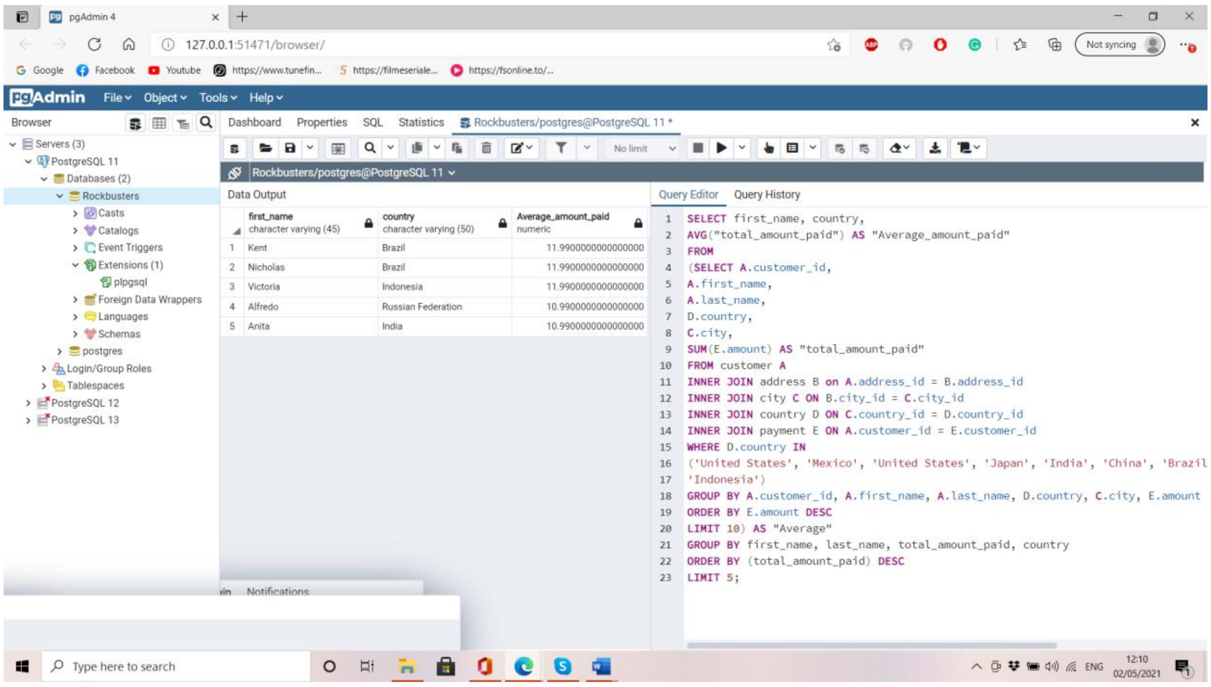
Step 1: Find the average amount paid by the top 5 customers.

- Copy the query you wrote in step 3 of the task from Exercise 3.7: Joining Tables of Data into the Query Tool. This will be your subquery, so give it an alias, “total_amount_paid,” and add parentheses around it.
- Write an outer statement to calculate the average amount paid.
- Add your subquery to the outer statement. It will go in either the SELECT, WHERE, or FROM clause. (Hint: When referring to the subquery in your outer statement, make sure to use the subquery’s alias, “total_amount_paid”.)
- If you have done everything correctly, pgAdmin 4 will require you to add an alias after the subquery. Go ahead and call it “average”.
- Copy-paste your queries and the final data output from pgAdmin 4 into your answers document.



Data Output

	first_name character varying (45)	country character varying (50)	Average_amount_paid numeric
1	Kent	Brazil	11.9900000000000000
2	Nicholas	Brazil	11.9900000000000000
3	Victoria	Indonesia	11.9900000000000000
4	Alfredo	Russian Federation	10.9900000000000000
5	Anita	India	10.9900000000000000

```
SELECT first_name, country,
AVG("total_amount_paid") AS "Average_amount_paid"
FROM
(SELECT A.customer_id,
A.first_name,
A.last_name,
D.country,
C.city,
SUM(E.amount) AS "total_amount_paid"
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
WHERE D.country IN
('United States', 'Mexico', 'United States', 'Japan', 'India', 'China', 'Brazil
'Indonesia')
GROUP BY A.customer_id, A.first_name, A.last_name, D.country, C.city, E.amount
ORDER BY E.amount DESC
LIMIT 10) AS "Average"
GROUP BY first_name, last_name, total_amount_paid, country
ORDER BY (total_amount_paid) DESC
LIMIT 5;
```

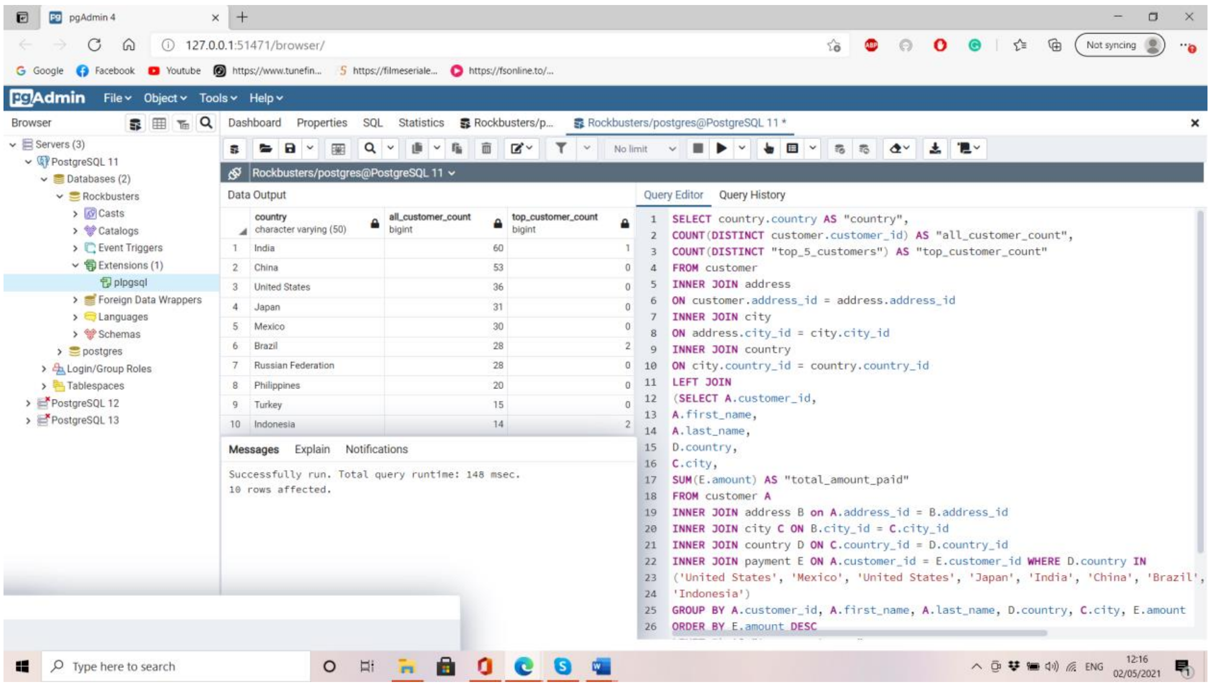
Step 2: Find out how many of the top 5 customers are based within each country.

- Your final output should include 3 columns:
  - "country"
  - "all_customer_count" with the total number of customers in each country,
  - "top_customer_count" showing how many of the top 5 customers live in each country.

- You will notice that this step is quite difficult. We have broken down each part and provided you with some helpful hints below:
  - Copy the query from step 3 of task 3.7 into the Query Tool and add parentheses around it. This will be your inner query. Write an outer statement that counts the number of customers living in each country. You will need to refer to your entity relationship diagram or data dictionary to do this. The information you need is in different tables, so you will have to use a join. To get the count for each country, use COUNT(DISTINCT) and GROUP BY. Give your second column the alias "all_customer_count" for readability.
  - Place your inner query in the outer query. Since you want to merge the entire output of the outer query with the information from your inner query, use a left join to connect the two queries on the "country" column.
  - Add a left join after your outer query, followed by the subquery in parentheses.
  - Give your subquery an alias so you can refer to it in your outer query, for example, "top_5_customers".
  - Remember to specify which columns to join the two tables on using ON. Both ON and the column names should follow the alias.
  - Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias "top_customer_count".
  - Copy-paste your query and the data output into your "Answers 3.8" document.

```
SELECT country.country AS "country",

COUNT(DISTINCT customer.customer_id) AS "all_customer_count",

COUNT(DISTINCT "top_5_customers") AS "top_customer_count"

FROM customer

INNER JOIN address

ON customer.address_id = address.address_id

INNER JOIN city
```

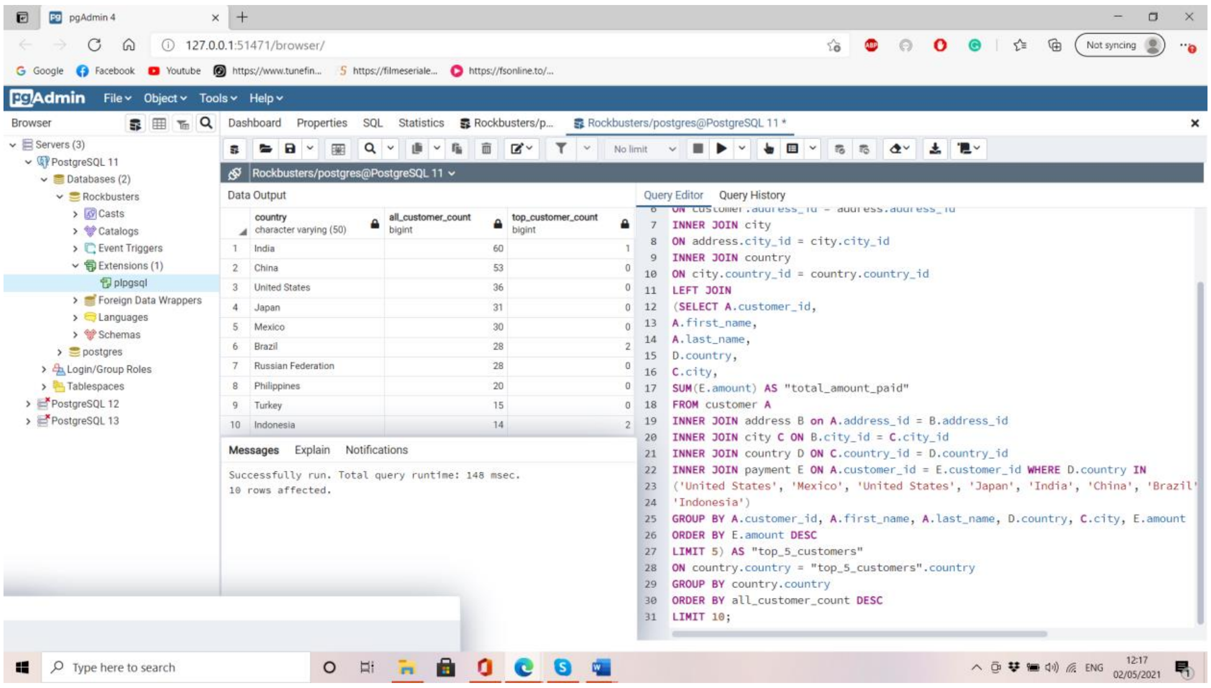
```
ON address.city_id = city.city_id

INNER JOIN country

ON city.country_id = country.country_id

LEFT JOIN

(SELECT A.customer_id,

A.first_name,

A.last_name,

D.country,

C.city,

SUM(E.amount) AS "total_amount_paid"

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

WHERE D.country IN

('United States', 'Mexico', 'United States', 'Japan', 'India', 'China', 'Brazil', 'Russian Federation',
'China',

'Indonesia')

GROUP BY A.customer_id, A.first_name, A.last_name, D.country, C.city, E.amount

ORDER BY E.amount DESC

LIMIT 5) AS "top_5_customers"

ON country.country = "top_5_customers".country

GROUP BY country.country

ORDER BY all_customer_count DESC

LIMIT 10;
```



	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	1
2	China	53	0
3	United States	36	0
4	Japan	31	0
5	Mexico	30	0
6	Brazil	28	2
7	Russian Federation	28	0
8	Philippines	20	0
9	Turkey	15	0
10	Indonesia	14	2

```
Successfully run. Total query runtime: 148 msec.
10 rows affected.
```

```sql
SELECT country.country AS "country",
COUNT(DISTINCT customer.customer_id) AS "all_customer_count",
COUNT(DISTINCT "top_5_customers") AS "top_customer_count"
FROM customer
INNER JOIN address
ON customer.address_id = address.address_id
INNER JOIN city
ON address.city_id = city.city_id
INNER JOIN country
ON city.country_id = country.country_id
LEFT JOIN
(SELECT A.customer_id,
A.first_name,
A.last_name,
D.country,
C.city,
SUM(E.amount) AS "total_amount_paid"
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 WHERE D.country IN
('United States', 'Mexico', 'United States', 'Japan', 'India', 'China', 'Brazil',
'Indonesia')
GROUP BY A.customer_id, A.first_name, A.last_name, D.country, C.city, E.amount
ORDER BY E.amount DESC
```



	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	India	60	1
2	China	53	0
3	United States	36	0
4	Japan	31	0
5	Mexico	30	0
6	Brazil	28	2
7	Russian Federation	28	0
8	Philippines	20	0
9	Turkey	15	0
10	Indonesia	14	2

```
Successfully run. Total query runtime: 148 msec.
10 rows affected.
```

```sql
INNER JOIN city
ON address.city_id = city.city_id
INNER JOIN country
ON city.country_id = country.country_id
LEFT JOIN
(SELECT A.customer_id,
A.first_name,
A.last_name,
D.country,
C.city,
SUM(E.amount) AS "total_amount_paid"
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 WHERE D.country IN
('United States', 'Mexico', 'United States', 'Japan', 'India', 'China', 'Brazil'
'Indonesia')
GROUP BY A.customer_id, A.first_name, A.last_name, D.country, C.city, E.amount
ORDER BY E.amount DESC
LIMIT 5) AS "top_5_customers"
ON country.country = "top_5_customers".country
GROUP BY country.country
ORDER BY all_customer_count DESC
LIMIT 10;
```

Step 3:

Write 1 to 2 short paragraphs on the following:

- Do you think steps 1 and 2 could be done without using subqueries?
    - ➢ Yes, I think step 1 and step 2 both could be done without subqueries. We already know which customers information need, so I would use the constrain command such as WHERE to specify the customers.

- When do you think subqueries are useful?
    - ➢ It is useful when I already obtain the query and need to extract information from the query. For example, when you must process data which your colleagues already built. Then you could use for further process by using query which the colleague built.