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

1. **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.**

   FULL JOIN

Query | Query History | Scratch Pad

```
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	customer_id smallint	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_amount_paid numeric
1	84	Sara	Perry	Atlixco	Mexico	128.70
2	518	Gabriel	Harder	Sivas	Turkey	108.75
3	587	Sergio	Stanfield	Celaya	Mexico	102.76
4	537	Clinton	Buford	Aurora	United States	98.76
5	367	Adam	Gooch	Adoni	India	97.80

or

INNER JOIN

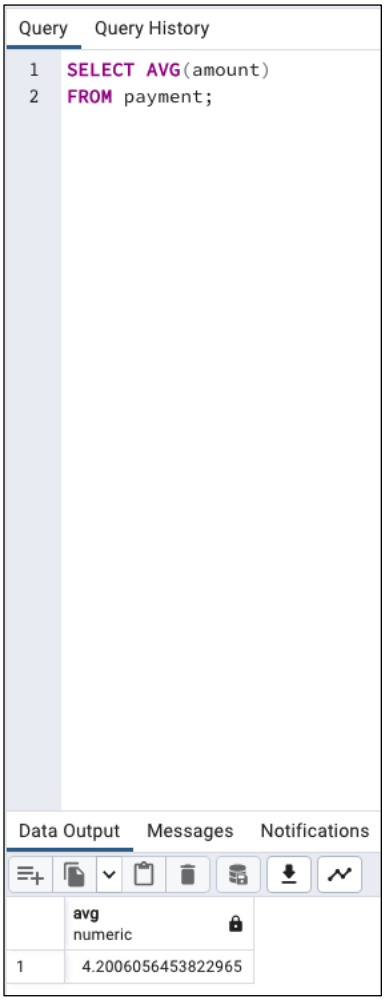
Query | Query History | Scratch Pad

```
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
                 'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	customer_id smallint	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_amount_paid numeric
1	84	Sara	Perry	Atlixco	Mexico	128.70
2	518	Gabriel	Harder	Sivas	Turkey	108.75
3	587	Sergio	Stanfield	Celaya	Mexico	102.76
4	537	Clinton	Buford	Aurora	United States	98.76
5	367	Adam	Gooch	Adoni	India	97.80

2. **Write an outer statement to calculate the average amount paid.**



3. **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".)**

FULL JOIN

Query | Query History

```
SELECT AVG(total_amount_paid)
FROM (SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                 'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
             'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	avg numeric
1	107.3540000000000000

or

INNER JOIN

Query | Query History

```
SELECT AVG(total_amount_paid)
FROM (SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
              'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	avg numeric
1	107.3540000000000000

4. **If you've done everything correctly, pgAdmin 4 will require you to add an alias after the subquery. Go ahead and call it "average".**
5. **Copy-paste your queries and the final data output from pgAdmin 4 into your answers document.**

FULL JOIN

Query | Query History

```
SELECT AVG(total_amount_paid) AS average
FROM (SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	average numeric
1	107.3540000000000000

or

INNER JOIN

Query | Query History

```
SELECT AVG(total_amount_paid) AS average
FROM (SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
             'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Data Output | Messages | Notifications

	average numeric
1	107.3540000000000000

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'll notice that this step is quite difficult. We've broken down each part and provided you with some helpful hints below:

1. **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.**

   FULL JOIN

Query | Query History | Scratch Pad

```
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5);
```

Data Output | Messages | Notifications

	customer_id smallint	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_amount_paid numeric
1	84	Sara	Perry	Atlixco	Mexico	128.70
2	518	Gabriel	Harder	Sivas	Turkey	108.75
3	587	Sergio	Stanfield	Celaya	Mexico	102.76
4	537	Clinton	Buford	Aurora	United States	98.76
5	367	Adam	Gooch	Adoni	India	97.80

or

INNER JOIN

Query | Query History | Scratch Pad

```
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
             'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5);
```

Data Output | Messages | Notifications

	customer_id smallint	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	total_amount_paid numeric
1	84	Sara	Perry	Atlixco	Mexico	128.70
2	518	Gabriel	Harder	Sivas	Turkey	108.75
3	587	Sergio	Stanfield	Celaya	Mexico	102.76
4	537	Clinton	Buford	Aurora	United States	98.76
5	367	Adam	Gooch	Adoni	India	97.80

2. **Write an outer statement that counts the number of customers living in each country. You'll need to refer to your entity relationship diagram or data dictionary in order to do this. The information you need is in different tables, so you'll 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.**

Query | Query History

```sql
-- Write an outer statement that counts the number of customers living in each country
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count -- To count the number of customers for each country, use COUNT(DISTINCT)
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id -- Use INNER JOIN for MULTIPLE JOINS since we only need customer_id and country
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D ON C.country_id = D.country_id
GROUP BY D.country -- Add every variable that is not the aggregated value
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint
1	Afghanistan	1
2	Algeria	3
3	American Samoa	1
4	Angola	2
5	Anguilla	1
6	Argentina	13
7	Armenia	1
8	Austria	3
9	Azerbaijan	2
10	Bahrain	1
11	Bangladesh	3
12	Belarus	2
13	Bolivia	2
14	Brazil	28
15	Brunei	1
16	Bulgaria	2
17	Cambodia	2
18	Cameroon	2

Total rows: 108 of 108 | Query complete 00:00:00.090

3. **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. You'll need to add a LEFT JOIN after your outer query, followed by the subquery in parentheses.**

FULL JOIN

Query | Query History

```sql
-- 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
-- You'll need to add a LEFT JOIN after your outer query, followed by the subquery in parentheses.
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5);
GROUP BY D.country
```

Moved GROUP BY to the bottom due to sequence order.

or

INNER JOIN

Query | Query History

```sql
-- 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
-- You’ll need to add a LEFT JOIN after your outer query, followed by the subquery in parentheses.
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
                 'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5);
GROUP BY D.country
```

4. **Give your subquery an alias so you can refer to it in your outer query, for example, "top_5_customers".**

FULL JOIN

Query | Query History

```
-- Give your subquery an alias so you can refer to it in your outer query, for example, "top_5_customers"
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers
GROUP BY D.country
```

Removed ; next to the parenthesis of the inner query.

or

INNER JOIN

Query | Query History

```sql
-- Give your subquery an alias so you can refer to it in your outer query, for example, “top_5_customers”
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
             'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers
GROUP BY D.country
```

5. **Remember to specify which columns to join the two tables on using ON. Both ON and the column names should follow the alias.**

FULL JOIN

Query | Query History

```sql
-- Remember to specify which columns to join the two tables on using ON.
-- Both ON and the column names should follow the alias.
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
GROUP BY D.country
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint
1	Afghanistan	1
2	Algeria	3
3	American Samoa	1
4	Angola	2
5	Anguilla	1
6	Argentina	13
7	Armenia	1
8	Austria	3
9	Azerbaijan	2
10	Bahrain	1
11	Bangladesh	3
12	Belarus	2
13	Bolivia	2
14	Brazil	28

Total rows: 108 of 108 | Query complete 00:00:00.113

or

INNER JOIN

Query | Query History

```sql
-- Remember to specify which columns to join the two tables on using ON.
-- Both ON and the column names should follow the alias.
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
             'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
GROUP BY D.country
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint
1	Afghanistan	1
2	Algeria	3
3	American Samoa	1
4	Angola	2
5	Anguilla	1
6	Argentina	13
7	Armenia	1
8	Austria	3
9	Azerbaijan	2
10	Bahrain	1
11	Bangladesh	3
12	Belarus	2
13	Bolivia	2
14	Brazil	28
15	Brunei	1
16	Bulgaria	2
17	Cambodia	2
18	Cameroon	2
19	Canada	5
20	Chad	1
21	Chile	3

Total rows: 108 of 108 | Query complete 00:00:00.052

6. **Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias "top_customer_count".**

7. **Copy-paste your query and the data output into your "Answers 3.8" document.**

FULL JOIN

Query | Query History

```sql
-- Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias "top_customer_count".
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count,
       COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
FROM payment A
FULL JOIN customer B ON A.customer_id = B.customer_id
FULL JOIN address C ON B.address_id = C.address_id
FULL JOIN city D ON C.city_id = D.city_id
FULL JOIN country E ON D.country_id = E.country_id
WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
                    'Russian Federation','Philippines','Turkey','Indonesia')
AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
               'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
GROUP BY D.country
ORDER BY top_customer_count DESC
LIMIT 5;
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	2
2	United States	36	1
3	India	60	1
4	Turkey	15	1
5	American Samoa	1	0

or

INNER JOIN

Query | Query History

```sql
-- Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias “top_customer_count”.
SELECT D.country,
       COUNT(DISTINCT A.customer_id) AS all_customer_count,
       COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
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
LEFT JOIN
(SELECT A.customer_id,
       B.first_name,
       B.last_name,
       D.city,
       E.country,
       SUM(A.amount) AS total_amount_paid
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','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
             'Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY A.customer_id,
         B.first_name,
         B.last_name,
         D.city,
         E.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
GROUP BY D.country
ORDER BY top_customer_count DESC
LIMIT 5;
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	2
2	United States	36	1
3	India	60	1
4	Turkey	15	1
5	American Samoa	1	0

Step 3:

1. **Write 1 to 2 short paragraphs on the following:**

- **Do you think steps 1 and 2 could be done without using subqueries?**

  Yes, through JOINs. JOINs can create an execution plan that is better for your query and can predict what data should be loaded to be processed and save time. JOINs are faster than subqueries and it is rare for a subquery to be faster. If your report needs data that is from more than one table, then you must perform a join. Whenever multiple tables (or views) are listed in the FROM clause, those tables become joined.

- **When do you think subqueries are useful?**

  A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved. Subqueries are useful when the result that you want requires more than one query and each subquery provides a subset of table involved in the query. For example, if a membership question is asked, then a subquery is used. If the

query requires a NOT EXISTS condition, then you must use a subquery because NOT EXISTS operates only in a subquery.