

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 X

```
1 (SELECT A.customer_id,
2     B.first_name,
3     B.last_name,
4     D.city,
5     E.country,
6     SUM(A.amount) AS total_amount_paid
7 FROM payment A
8 FULL JOIN customer B ON A.customer_id = B.customer_id
9 FULL JOIN address C ON B.address_id = C.address_id
10 FULL JOIN city D ON C.city_id = D.city_id
11 FULL JOIN country E ON D.country_id = E.country_id
12 WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
13                     'Russian Federation','Philippines','Turkey','Indonesia')
14 AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
15               'Pingxiang','Sivas','Celaya','So Leopoldo')
16 GROUP BY A.customer_id,
17          B.first_name,
18          B.last_name,
19          D.city,
20          E.country
21 ORDER BY total_amount_paid DESC
22 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 x

```
1 (SELECT A.customer_id,
2     B.first_name,
3     B.last_name,
4     D.city,
5     E.country,
6     SUM(A.amount) AS total_amount_paid
7 FROM payment A
8 INNER JOIN customer B ON A.customer_id = B.customer_id
9 INNER JOIN address C ON B.address_id = C.address_id
10 INNER JOIN city D ON C.city_id = D.city_id
11 INNER JOIN country E ON D.country_id = E.country_id
12 WHERE D.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
13                 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
14 GROUP BY A.customer_id,
15           B.first_name,
16           B.last_name,
17           D.city,
18           E.country
19 ORDER BY total_amount_paid DESC
20 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)	totalAmountPaid 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.

Query	Query History
1	SELECT AVG (amount)
2	FROM payment;

Data Output	Messages	Notifications
<div> </div>		
avg		
numeric		
1	4.2006056453822965	

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

```
1 SELECT AVG(total_amount_paid)
2 FROM (SELECT A.customer_id,
3         B.first_name,
4         B.last_name,
5         D.city,
6         E.country,
7         SUM(A.amount) AS total_amount_paid
8 FROM payment A
9 FULL JOIN customer B ON A.customer_id = B.customer_id
10 FULL JOIN address C ON B.address_id = C.address_id
11 FULL JOIN city D ON C.city_id = D.city_id
12 FULL JOIN country E ON D.country_id = E.country_id
13 WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
14                    'Russian Federation','Philippines','Turkey','Indonesia')
15 AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
16              'Pingxiang','Sivas','Celaya','So Leopoldo')
17 GROUP BY A.customer_id,
18          B.first_name,
19          B.last_name,
20          D.city,
21          E.country
22 ORDER BY total_amount_paid DESC
23 LIMIT 5) AS total_amount_paid
```

Data Output

Messages

Notifications



	avg	
	numeric	
1	107.354000000000000000	

or

INNER JOIN

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

```
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

```
1 SELECT AVG(total_amount_paid) AS average
2 FROM (SELECT A.customer_id,
3         B.first_name,
4         B.last_name,
5         D.city,
6         E.country,
7         SUM(A.amount) AS total_amount_paid
8 FROM payment A
9 FULL JOIN customer B ON A.customer_id = B.customer_id
10 FULL JOIN address C ON B.address_id = C.address_id
11 FULL JOIN city D ON C.city_id = D.city_id
12 FULL JOIN country E ON D.country_id = E.country_id
13 WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
14                    'Russian Federation','Philippines','Turkey','Indonesia')
15 AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
16               'Pingxiang','Sivas','Celaya','So Leopoldo')
17 GROUP BY A.customer_id,
18         B.first_name,
19         B.last_name,
20         D.city,
21         E.country
22 ORDER BY total_amount_paid DESC
23 LIMIT 5) AS total_amount_paid
```

Data Output

Messages

Notifications

	average	
	numeric	
1	107.3540000000000000	

or

INNER JOIN

Query









Query History

```
1 SELECT AVG(total_amount_paid) AS average
2 FROM (SELECT A.customer_id,
3         B.first_name,
4         B.last_name,
5         D.city,
6         E.country,
7         SUM(A.amount) AS total_amount_paid
8 FROM payment A
9 INNER JOIN customer B ON A.customer_id = B.customer_id
10 INNER JOIN address C ON B.address_id = C.address_id
11 INNER JOIN city D ON C.city_id = D.city_id
12 INNER JOIN country E ON D.country_id = E.country_id
13 WHERE D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
14                 'Pingxiang','Sivas','Celaya','So Leopoldo')
15 GROUP BY A.customer_id,
16          B.first_name,
17          B.last_name,
18          D.city,
19          E.country
20 ORDER BY total_amount_paid DESC
21 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

```
1 (SELECT A.customer_id,
2     B.first_name,
3     B.last_name,
4     D.city,
5     E.country,
6     SUM(A.amount) AS total_amount_paid
7 FROM payment A
8 FULL JOIN customer B ON A.customer_id = B.customer_id
9 FULL JOIN address C ON B.address_id = C.address_id
10 FULL JOIN city D ON C.city_id = D.city_id
11 FULL JOIN country E ON D.country_id = E.country_id
12 WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
13                    'Russian Federation','Philippines','Turkey','Indonesia')
14 AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
15              'Pingxiang','Sivas','Celaya','So Leopoldo')
16 GROUP BY A.customer_id,
17          B.first_name,
18          B.last_name,
19          D.city,
20          E.country
21 ORDER BY total_amount_paid DESC
22 LIMIT 5);
```

Scratch Pad X

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 x

```
1 (SELECT A.customer_id,
2     B.first_name,
3     B.last_name,
4     D.city,
5     E.country,
6     SUM(A.amount) AS total_amount_paid
7 FROM payment A
8 INNER JOIN customer B ON A.customer_id = B.customer_id
9 INNER JOIN address C ON B.address_id = C.address_id
10 INNER JOIN city D ON C.city_id = D.city_id
11 INNER JOIN country E ON D.country_id = E.country_id
12 WHERE D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
13                 'Pingxiang','Sivas','Celaya','So Leopoldo')
14 GROUP BY A.customer_id,
15           B.first_name,
16           B.last_name,
17           D.city,
18           E.country
19 ORDER BY total_amount_paid DESC
20 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)	totalAmountPaid 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

```
1 -- Write an outer statement that counts the number of customers living in each country
2 SELECT D.country,
3        COUNT(DISTINCT A.customer_id) AS all_customer_count -- To count the number of customers for each country, use COUNT(DISTINCT)
4 FROM customer A
5 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
6 INNER JOIN city C ON B.city_id = C.city_id
7 INNER JOIN country D ON C.country_id = D.country_id
8 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
1	-- Place your inner query in the outer query.
2	-- Since you want to merge the entire output of the outer query with the information from your inner query,
3	-- use a left join to connect the two queries on the “country” column
4	-- You’ll need to add a LEFT JOIN after your outer query, followed by the subquery in parentheses.
5	SELECT D.country,
6	COUNT(DISTINCT A.customer_id) AS all_customer_count
7	FROM customer A
8	INNER JOIN address B ON A.address_id = B.address_id
9	INNER JOIN city C ON B.city_id = C.city_id
10	INNER JOIN country D ON C.country_id = D.country_id
11	LEFT JOIN
12	(SELECT A.customer_id,
13	B.first_name,
14	B.last_name,
15	D.city,
16	E.country,
17	SUM(A.amount) AS total_amount_paid
18	FROM payment A
19	FULL JOIN customer B ON A.customer_id = B.customer_id
20	FULL JOIN address C ON B.address_id = C.address_id
21	FULL JOIN city D ON C.city_id = D.city_id
22	FULL JOIN country E ON D.country_id = E.country_id
23	WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
24	'Russian Federation','Philippines','Turkey','Indonesia')
25	AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
26	'Pingxiang','Sivas','Celaya','So Leopoldo')
27	GROUP BY A.customer_id,
28	B.first_name,
29	B.last_name,
30	D.city,
31	E.country
32	ORDER BY total_amount_paid DESC
33	LIMIT 5);
34	GROUP BY D.country

Moved GROUP BY to the bottom due to sequence order.

or

INNER JOIN

Query	Query History
1	-- Place your inner query in the outer query.
2	-- Since you want to merge the entire output of the outer query with the information from your inner query,
3	-- use a left join to connect the two queries on the "country" column
4	-- You'll need to add a LEFT JOIN after your outer query, followed by the subquery in parentheses.
5	SELECT D.country,
6	COUNT(DISTINCT A.customer_id) AS all_customer_count
7	FROM customer A
8	INNER JOIN address B ON A.address_id = B.address_id
9	INNER JOIN city C ON B.city_id = C.city_id
10	INNER JOIN country D ON C.country_id = D.country_id
11	LEFT JOIN
12	(SELECT A.customer_id,
13	B.first_name,
14	B.last_name,
15	D.city,
16	E.country,
17	SUM(A.amount) AS total_amount_paid
18	FROM payment A
19	INNER JOIN customer B ON A.customer_id = B.customer_id
20	INNER JOIN address C ON B.address_id = C.address_id
21	INNER JOIN city D ON C.city_id = D.city_id
22	INNER JOIN country E ON D.country_id = E.country_id
23	WHERE D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
24	'Pingxiang','Sivas','Celaya','So Leopoldo')
25	GROUP BY A.customer_id,
26	B.first_name,
27	B.last_name,
28	D.city,
29	E.country
30	ORDER BY total_amount_paid DESC
31	LIMIT 5);
32	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
1  -- Give your subquery an alias so you can refer to it in your outer query, for example, "top_5_customers"
2  SELECT D.country,
3         COUNT(DISTINCT A.customer_id) AS all_customer_count
4  FROM customer A
5  INNER JOIN address B ON A.address_id = B.address_id
6  INNER JOIN city C ON B.city_id = C.city_id
7  INNER JOIN country D ON C.country_id = D.country_id
8  LEFT JOIN
9  (SELECT A.customer_id,
10         B.first_name,
11         B.last_name,
12         D.city,
13         E.country,
14         SUM(A.amount) AS total_amount_paid
15  FROM payment A
16  FULL JOIN customer B ON A.customer_id = B.customer_id
17  FULL JOIN address C ON B.address_id = C.address_id
18  FULL JOIN city D ON C.city_id = D.city_id
19  FULL JOIN country E ON D.country_id = E.country_id
20  WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
21                     'Russian Federation','Philippines','Turkey','Indonesia')
22  AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
23               'Pingxiang','Sivas','Celaya','So Leopoldo')
24  GROUP BY A.customer_id,
25           B.first_name,
26           B.last_name,
27           D.city,
28           E.country
29  ORDER BY total_amount_paid DESC
30  LIMIT 5) AS top_5_customers
31  GROUP BY D.country
```

Removed ; next to the parenthesis of the inner query.

or

INNER JOIN

```
Query    Query History
1  -- Give your subquery an alias so you can refer to it in your outer query, for example, "top_5_customers"
2  SELECT D.country,
3         COUNT(DISTINCT A.customer_id) AS all_customer_count
4  FROM   customer A
5  INNER JOIN address B ON A.address_id = B.address_id
6  INNER JOIN city C ON B.city_id = C.city_id
7  INNER JOIN country D ON C.country_id = D.country_id
8  LEFT JOIN
9  (SELECT A.customer_id,
10         B.first_name,
11         B.last_name,
12         D.city,
13         E.country,
14         SUM(A.amount) AS total_amount_paid
15  FROM   payment A
16  INNER JOIN customer B ON A.customer_id = B.customer_id
17  INNER JOIN address C ON B.address_id = C.address_id
18  INNER JOIN city D ON C.city_id = D.city_id
19  INNER JOIN country E ON D.country_id = E.country_id
20  WHERE D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
21                 'Pingxiang','Sivas','Celaya','So Leopoldo')
22  GROUP BY A.customer_id,
23           B.first_name,
24           B.last_name,
25           D.city,
26           E.country
27  ORDER BY total_amount_paid DESC
28  LIMIT 5) AS top_5_customers
29  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

```
1  -- Remember to specify which columns to join the two tables on using ON.
2  -- Both ON and the column names should follow the alias.
3  SELECT D.country,
4         COUNT(DISTINCT A.customer_id) AS all_customer_count
5  FROM customer A
6  INNER JOIN address B ON A.address_id = B.address_id
7  INNER JOIN city C ON B.city_id = C.city_id
8  INNER JOIN country D ON C.country_id = D.country_id
9  LEFT JOIN
10 (SELECT A.customer_id,
11        B.first_name,
12        B.last_name,
13        D.city,
14        E.country,
15        SUM(A.amount) AS total_amount_paid
16 FROM payment A
17 FULL JOIN customer B ON A.customer_id = B.customer_id
18 FULL JOIN address C ON B.address_id = C.address_id
19 FULL JOIN city D ON C.city_id = D.city_id
20 FULL JOIN country E ON D.country_id = E.country_id
21 WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
22                    'Russian Federation','Philippines','Turkey','Indonesia')
23 AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
24               'Pingxiang','Sivas','Celaya','So Leopoldo')
25 GROUP BY A.customer_id,
26          B.first_name,
27          B.last_name,
28          D.city,
29          E.country
30 ORDER BY total_amount_paid DESC
31 LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
32 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

1

-- Remember to specify which columns to join the two tables on using ON.

2

-- Both ON and the column names should follow the alias.

3

SELECT D.country,

4

COUNT(DISTINCT A.customer_id) AS all_customer_count

5

FROM customer A

6

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

7

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

8

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

9

LEFT JOIN

10

(SELECT A.customer_id,

11

B.first_name,

12

B.last_name,

13

D.city,

14

E.country,

15

SUM(A.amount) AS total_amount_paid

16

FROM payment A

17

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

18

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

19

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

20

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

21

WHERE D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',

22

'Pingxiang','Sivas','Celaya','So Leopoldo')

23

GROUP BY A.customer_id,

24

B.first_name,

25

B.last_name,

26

D.city,

27

E.country

28

ORDER BY total_amount_paid DESC

29

LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country

30

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

```

1 -- Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias "top_customer_count".
2 SELECT D.country,
3        COUNT(DISTINCT A.customer_id) AS all_customer_count,
4        COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
5 FROM customer A
6 INNER JOIN address B ON A.address_id = B.address_id
7 INNER JOIN city C ON B.city_id = C.city_id
8 INNER JOIN country D ON C.country_id = D.country_id
9 LEFT JOIN
10 (SELECT A.customer_id,
11        B.first_name,
12        B.last_name,
13        D.city,
14        E.country,
15        SUM(A.amount) AS total_amount_paid
16 FROM payment A
17 FULL JOIN customer B ON A.customer_id = B.customer_id
18 FULL JOIN address C ON B.address_id = C.address_id
19 FULL JOIN city D ON C.city_id = D.city_id
20 FULL JOIN country E ON D.country_id = E.country_id
21 WHERE E.country IN ('India','China','United States','Japan','Mexico','Brazil',
22                    'Russian Federation','Philippines','Turkey','Indonesia')
23 AND D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
24               'Pingxiang','Sivas','Celaya','So Leopoldo')
25 GROUP BY A.customer_id,
26          B.first_name,
27          B.last_name,
28          D.city,
29          E.country
30 ORDER BY total_amount_paid DESC
31 LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
32 GROUP BY D.country
33 ORDER BY top_customer_count DESC
34 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

```
1 -- Count the top 5 customers for the third column using GROUP BY and COUNT (DISTINCT). Give this column the alias "top_customer_count".
2 SELECT D.country,
3        COUNT(DISTINCT A.customer_id) AS all_customer_count,
4        COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
5 FROM customer A
6 INNER JOIN address B ON A.address_id = B.address_id
7 INNER JOIN city C ON B.city_id = C.city_id
8 INNER JOIN country D ON C.country_id = D.country_id
9 LEFT JOIN
10 (SELECT A.customer_id,
11        B.first_name,
12        B.last_name,
13        D.city,
14        E.country,
15        SUM(A.amount) AS total_amount_paid
16 FROM payment A
17 INNER JOIN customer B ON A.customer_id = B.customer_id
18 INNER JOIN address C ON B.address_id = C.address_id
19 INNER JOIN city D ON C.city_id = D.city_id
20 INNER JOIN country E ON D.country_id = E.country_id
21 WHERE D.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki',
22                'Pingxiang','Sivas','Celaya','So Leopoldo')
23 GROUP BY A.customer_id,
24          B.first_name,
25          B.last_name,
26          D.city,
27          E.country
28 ORDER BY total_amount_paid DESC
29 LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
30 GROUP BY D.country
31 ORDER BY top_customer_count DESC
32 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.