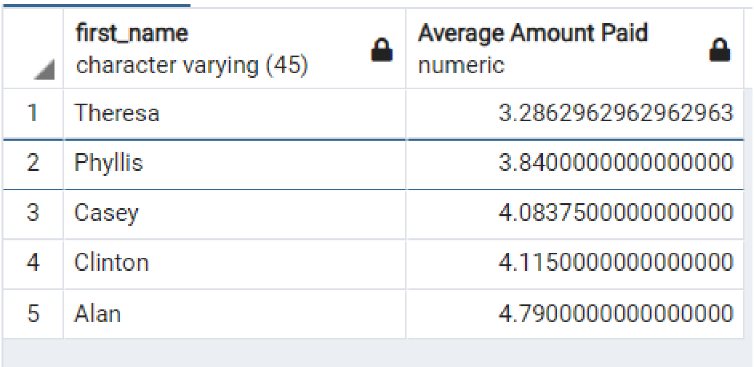
Task 3.8

Step 1:

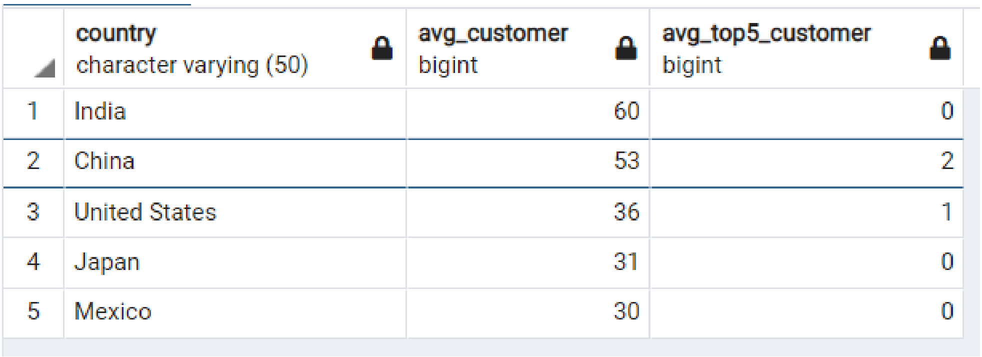
```
SELECT B.first_name, AVG (A.amount) AS "Average Amount Paid"
FROM payment A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C ON B.address_id = C.address_id
INNER JOIN city D ON C.city_id = D.city_id
INNER JOIN country E ON D.country_ID = E.country_ID
RIGHT JOIN
        (SELECT A.customer_id, B.first_name, B.last_name, E.country, D.city,
                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','Tokat', 'Tarsus', 'Atlixco', 'Emeishan', 'Pontianak', 'Shimoga',
        'Aparecida de Goinia', 'Zalantun', 'Taguig')
        GROUP BY A.customer_id, B.first_name, B.last_name, E.country, D.city, A.amount
        ORDER BY total_amount_paid DESC
        LIMIT 5) AS Z
ON B.customer_id = Z.customer_id
GROUP BY B.first_name
ORDER BY "Average Amount Paid"
LIMIT 5
```



	first_name character varying (45)	Average Amount Paid numeric
1	Theresa	3.2862962962962963
2	Phyllis	3.8400000000000000
3	Casey	4.0837500000000000
4	Clinton	4.1150000000000000
5	Alan	4.7900000000000000

Step 2:

```
SELECT E.country, COUNT (DISTINCT B.customer_id) as "avg_customer", COUNT (DISTINCT
Z.customer_id) as "avg_top5_customer"

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 A.customer_id, B.first_name, B.last_name, E.country, D.city,

                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','Tokat', 'Tarsus', 'Atlixco', 'Emeishan', 'Pontianak', 'Shimoga',
        'Aparecida de Goinia', 'Zalantun', 'Taguig')

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

        ORDER BY total_amount_paid DESC
        LIMIT 5) AS Z

ON E.country = Z.country

GROUP BY E.country

ORDER BY "avg_customer" DESC, "avg_top5_customer" DESC

LIMIT 5
```



	country character varying (50)	avg_customer bigint	avg_top5_customer bigint
1	India	60	0
2	China	53	2
3	United States	36	1
4	Japan	31	0
5	Mexico	30	0

Step 3:

Both Step 1 and Step 2 could’ve been done without a suquerie, however, for the sake of the exercise, we went ahead and performed the extra steps. By writing out the customer id’s individually we would’ve skipped the subquerie altogether. Subqueries are most useful when analyzing data that changes consistently. In order to keep up with the newer data and having to write two different queries, we can use the subqueries to keep the data up to date.