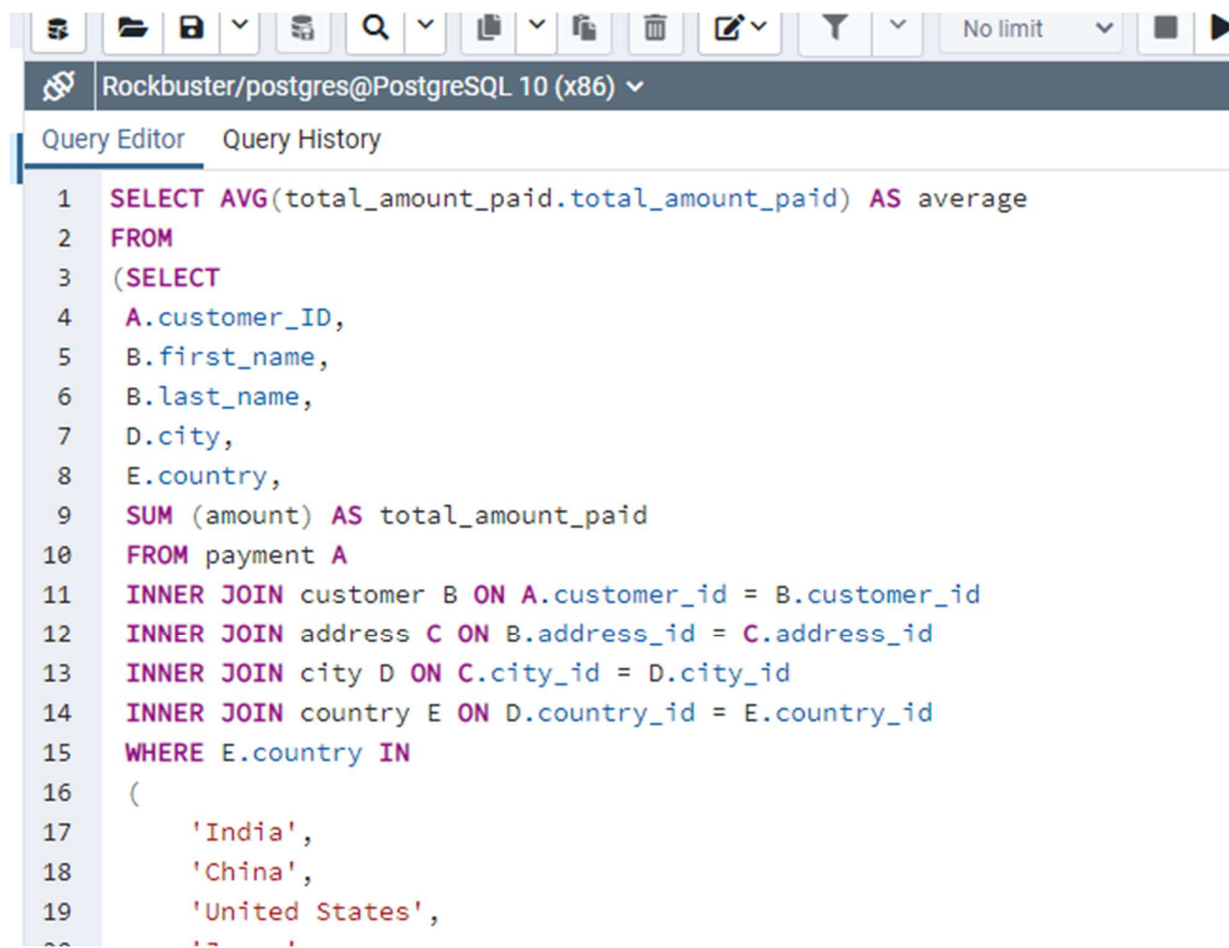


Performing Subqueries



The screenshot shows a PostgreSQL query editor window titled "Rockbuster/postgres@PostgreSQL 10 (x86)". The interface includes a toolbar with icons for file operations, search, and execution. Below the toolbar, there are tabs for "Query Editor" and "Query History". The "Query Editor" tab is active, displaying a SQL query. The query is a SELECT statement that calculates the average total amount paid for customers in India, China, and the United States. It uses a subquery to select customer details and their total amount paid from the payment table, joined with customer, address, city, and country tables. The subquery results are then filtered by country and averaged.

```
1 SELECT AVG(total_amount_paid.total_amount_paid) AS average
2 FROM
3 (SELECT
4   A.customer_ID,
5   B.first_name,
6   B.last_name,
7   D.city,
8   E.country,
9   SUM (amount) AS total_amount_paid
10  FROM payment A
11  INNER JOIN customer B ON A.customer_id = B.customer_id
12  INNER JOIN address C ON B.address_id = C.address_id
13  INNER JOIN city D ON C.city_id = D.city_id
14  INNER JOIN country E ON D.country_id = E.country_id
15  WHERE E.country IN
16  (
17    'India',
18    'China',
19    'United States',
20    'United Kingdom')
```

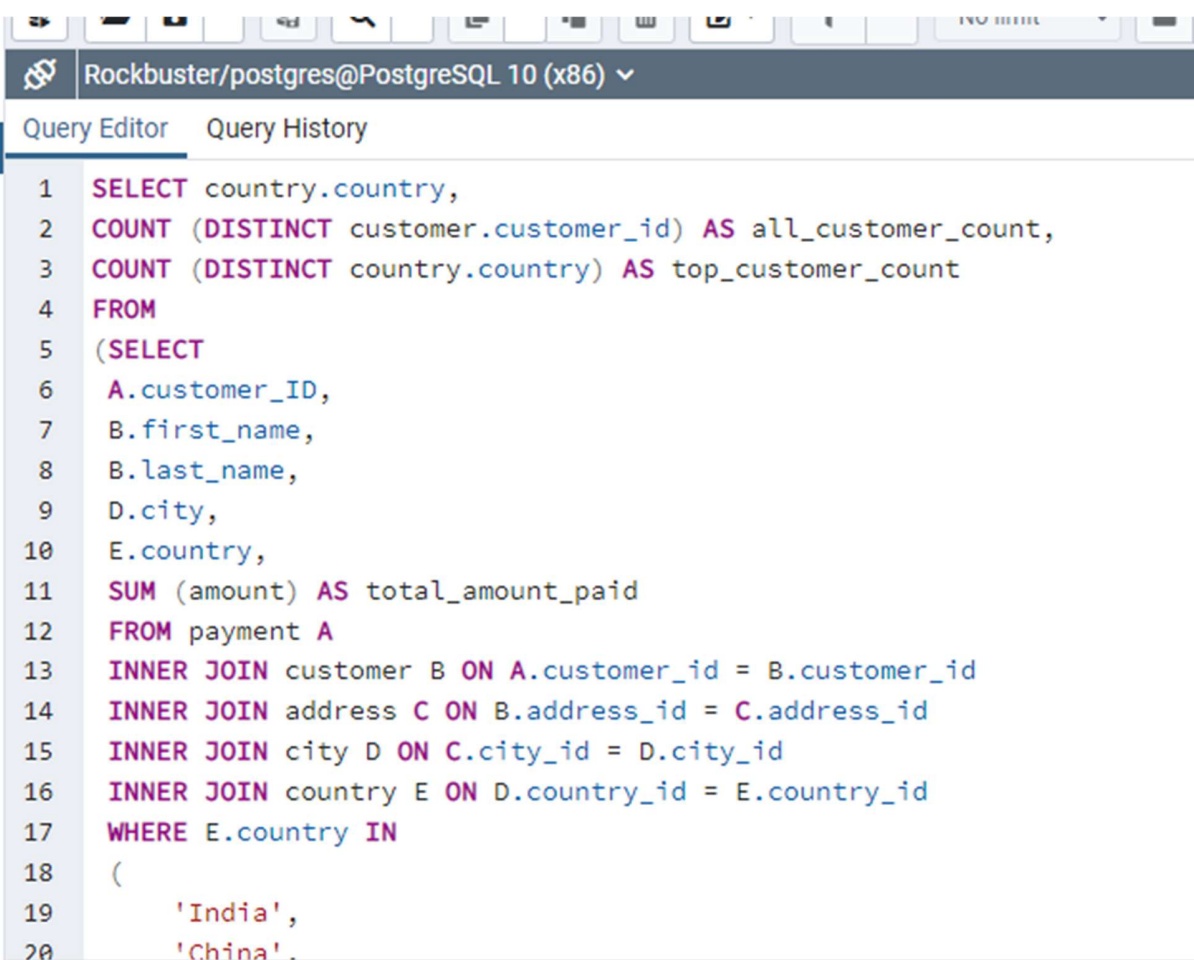
```

20      'Japan',
21      'Mexico',
22      'Brazil',
23      'Russian Federation',
24      'Philippines',
25      'Turkey',
26      'Indonesia'
27    )
28    AND D.city in
29    (
30      'Aurora',
31      'Atlixco',
32      'Xintai',
33      'Adoni',
34      'Dhule (Dhulia)',
35      'Kurashiki',
36      'Pingxiang',
37      'Sivas',
38      'Celaya',
39      'So Leopoldo'
40    )
41    GROUP BY
42    A.customer_ID,
43    B.first_name,
44    B.last_name,
45    D.city,
46    E.country
47    ORDER BY total_amount_paid DESC
48    LIMIT 5) AS total_amount_paid
49

```

Data Output Explain Messages Notifications

	average numeric	
1	107.3540000000000000	



The image shows a screenshot of a PostgreSQL Query Editor window. The title bar at the top reads "Rockbuster/postgres@PostgreSQL 10 (x86)". Below the title bar, there are two tabs: "Query Editor" and "Query History". The "Query Editor" tab is active, displaying a SQL query. The query is a complex SELECT statement that joins several tables (payment, customer, address, city, country) and calculates aggregate counts and sums. The query is as follows:

```
1 SELECT country.country,  
2 COUNT (DISTINCT customer.customer_id) AS all_customer_count,  
3 COUNT (DISTINCT country.country) AS top_customer_count  
4 FROM  
5 (SELECT  
6 A.customer_ID,  
7 B.first_name,  
8 B.last_name,  
9 D.city,  
10 E.country,  
11 SUM (amount) AS total_amount_paid  
12 FROM payment A  
13 INNER JOIN customer B ON A.customer_id = B.customer_id  
14 INNER JOIN address C ON B.address_id = C.address_id  
15 INNER JOIN city D ON C.city_id = D.city_id  
16 INNER JOIN country E ON D.country_id = E.country_id  
17 WHERE E.country IN  
18 (  
19     'India',  
20     'China',
```

```
20      'China',
21      'United States',
22      'Japan',
23      'Mexico',
24      'Brazil',
25      'Russian Federation',
26      'Philippines',
27      'Turkey',
28      'Indonesia'
29  )
30  AND D.city in
31  (
32      'Aurora',
33      'Atlixco',
34      'Xintai',
35      'Adoni',
36      'Dhule (Dhulia)',
37      'Kurashiki',
38      'Pingxiang',
39      'Sivas'.
```

```

40         'Celaya',
41         'So Leopoldo'
42     )
43     GROUP BY
44     A.customer_ID,
45     B.first_name,
46     B.last_name,
47     D.city,
48     E.country
49     ORDER BY total_amount_paid DESC
50     LIMIT 5) AS top_5_customers
51     LEFT JOIN customer ON customer.customer_id = customer.customer_id
52     LEFT JOIN address ON customer.address_id = address.address_id
53     LEFT JOIN city ON address.city_id = city.city_id
54     LEFT JOIN country ON city.country_id = country.country_id
55     GROUP BY country.country
56     ORDER BY COUNT (country.country) DESC
57

```

Data Output Explain Messages Notifications

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

- Step 1 can be done without a subquery function whilst with Step 2 we will need a subquery because we have to compare other tables.
- Subqueries are useful for comparing different tables.