Achievement 3.8
Javiera Quezada

Performing Subqueries.

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

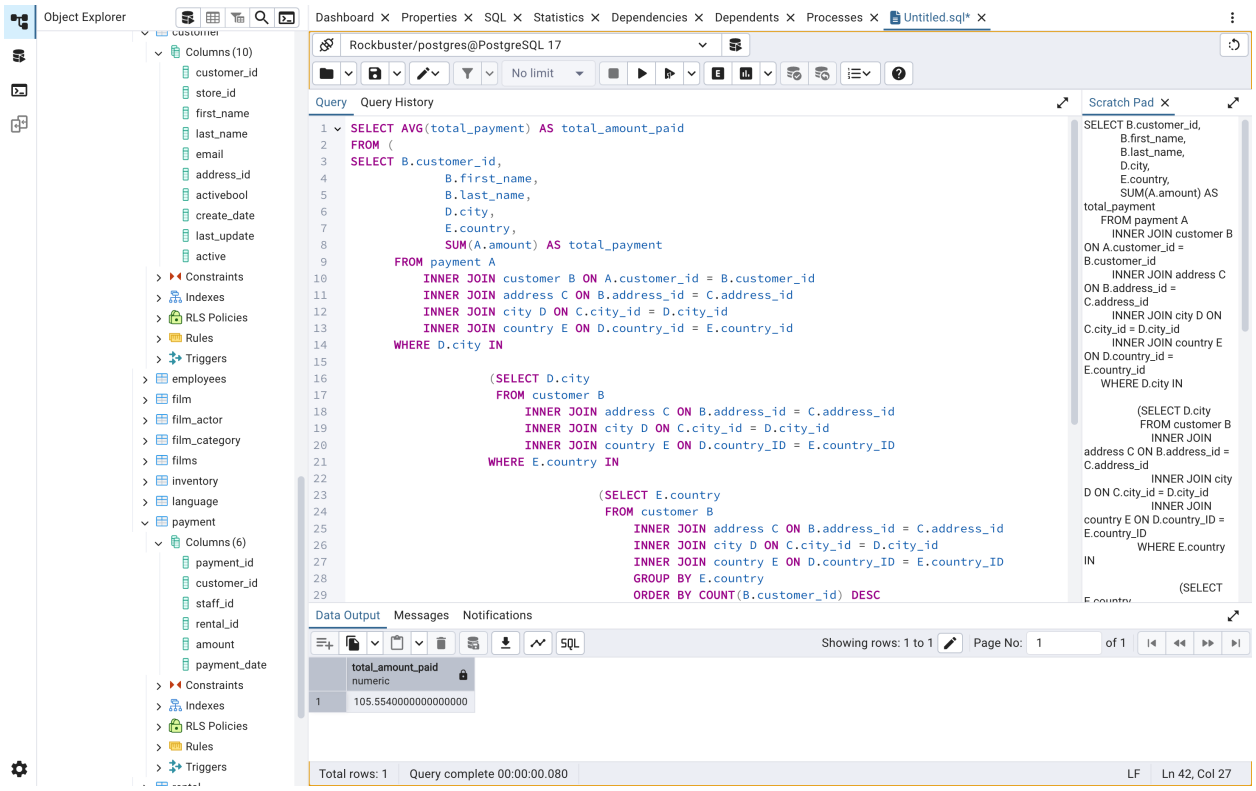
```
SELECT AVG(total_payment) AS total_amount_paid
FROM (
	SELECT B.customer_id,
		B.first_name,
		B.last_name,
		D.city,
		E.country,
		SUM(A.amount) AS total_payment
	 FROM payment A
		INNER JOIN customer B ON A.customer_id = B.customer_id
		INNER JOIN address C ON B.address_id = C.address_id
		INNER 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

	  (SELECT D.city
	   FROM customer B
		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 E.country IN

			(SELECT E.country
			 FROM customer B
				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
					GROUP BY E.country
					ORDER BY COUNT(B.customer_id) DESC
					LIMIT 10)

		GROUP BY E.country, D.city
		ORDER BY COUNT(B.customer_id) DESC
		LIMIT 10)

	  GROUP BY E.country,   D.city,  B.customer_id,  B.first_name,  B.last_name
	  ORDER BY SUM(A.amount) DESC
	   LIMIT 5)
```



Output 3.8.S1.1

total_amount_paid
105.5540000000000000

Step 2: Find out how many of the top 5 customers you identified in step 1 are based within each country.

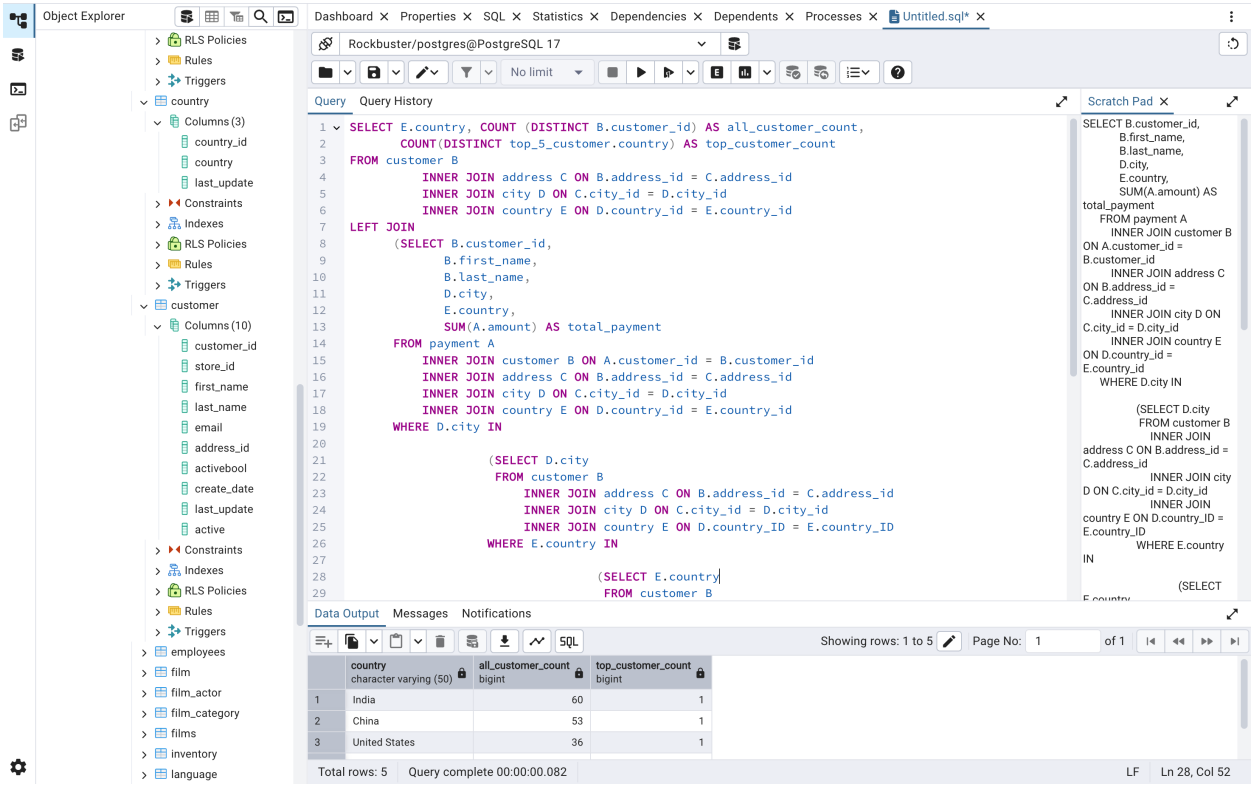
```
SELECT E.country, COUNT (DISTINCT B.customer_id) AS all_customer_count,
    COUNT(DISTINCT top_5_customer.country) AS top_customer_count
FROM customer B
      INNER JOIN address C ON B.address_id = C.address_id
      INNER JOIN city D ON C.city_id = D.city_id
      INNER JOIN country E ON D.country_id = E.country_id
LEFT JOIN
       (SELECT B.customer_id,
                  B.first_name,
                  B.last_name,
                  D.city,
                  E.country,
                  SUM(A.amount) AS total_payment
        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

                  (SELECT D.city
                   FROM customer B
                          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 E.country IN

                           (SELECT E.country
                            FROM customer B
                                   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
                            GROUP BY E.country
                            ORDER BY COUNT(B.customer_id) DESC
                                                          LIMIT 10)

                      GROUP BY
                       E.country,
                       D.city
                      ORDER BY COUNT(B.customer_id) DESC
                                                    LIMIT 10)

           GROUP BY
                        E.country,
                        D.city,
                        B.customer_id,
                        B.first_name,
                                B.last_name
             ORDER BY SUM(A.amount) DESC
                                       LIMIT 5) AS top_5_customer ON top_5_customer.country = E.country
GROUP BY E.country
ORDER BY all_customer_count DESC
LIMIT 5;
```



Output A.3.8.S.2

country	all_customer_count	top_customer_count
India	60	1
China	53	1
United States	36	1
Japan	31	1
Mexico	30	1

Step 3:

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

When we need to isolate steps with filtering and aggregating data, correlate values row by row and encapsulate results with temporary tables for reuse.

I think it made easier to this case because of following a way of getting different specific answers to different questions in the same common root. Probably if we would need more general answers or we were making exploratory analysis, we could have come to the same with a wider view. This saved us time, and allowed us to go straight to the point.