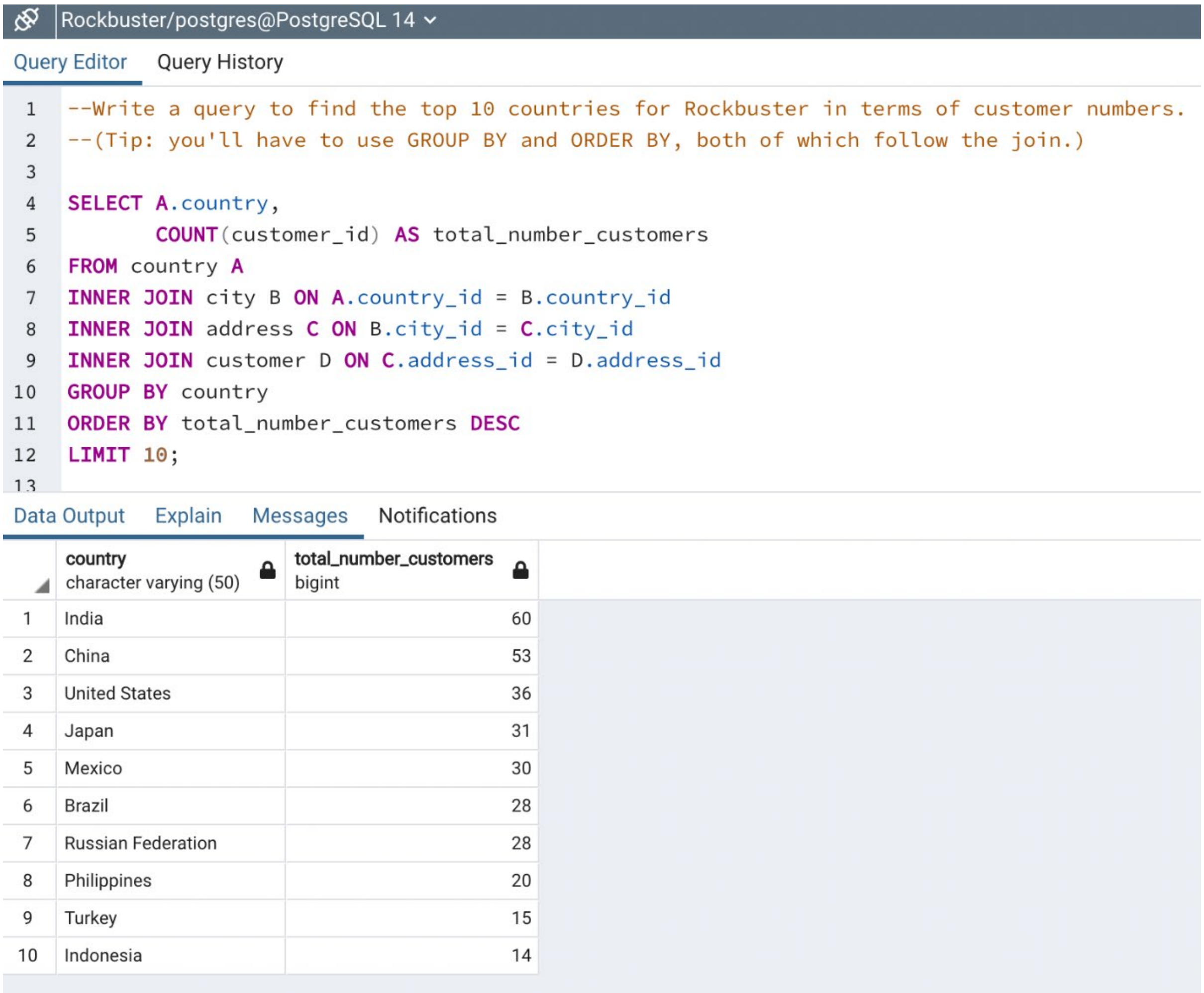
Juan Ignacio Galvalisi

Exercise 3.7: Joining Tables of Data

1. Write a query to find the top 10 countries for Rockbuster in terms of customer numbers. (Tip: you'll have to use GROUP BY and ORDER BY, both of which follow the join.)



```sql
--Write a query to find the top 10 countries for Rockbuster in terms of customer numbers.
--(Tip: you'll have to use GROUP BY and ORDER BY, both of which follow the join.)

SELECT A.country,
       COUNT(customer_id) AS total_number_customers
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
GROUP BY country
ORDER BY total_number_customers DESC
LIMIT 10;

```

	country character varying (50)	total_number_customers bigint
1	India	60
2	China	53
3	United States	36
4	Japan	31
5	Mexico	30
6	Brazil	28
7	Russian Federation	28
8	Philippines	20
9	Turkey	15
10	Indonesia	14

First of all, it is a good practice to look into our Entity Relationship Diagram (ERD) because it will give us a good point of view of which tables we have to retrieve to find out what countries are the largest in the number of customers around the world. From there, we started to join from the country table to the customer table.

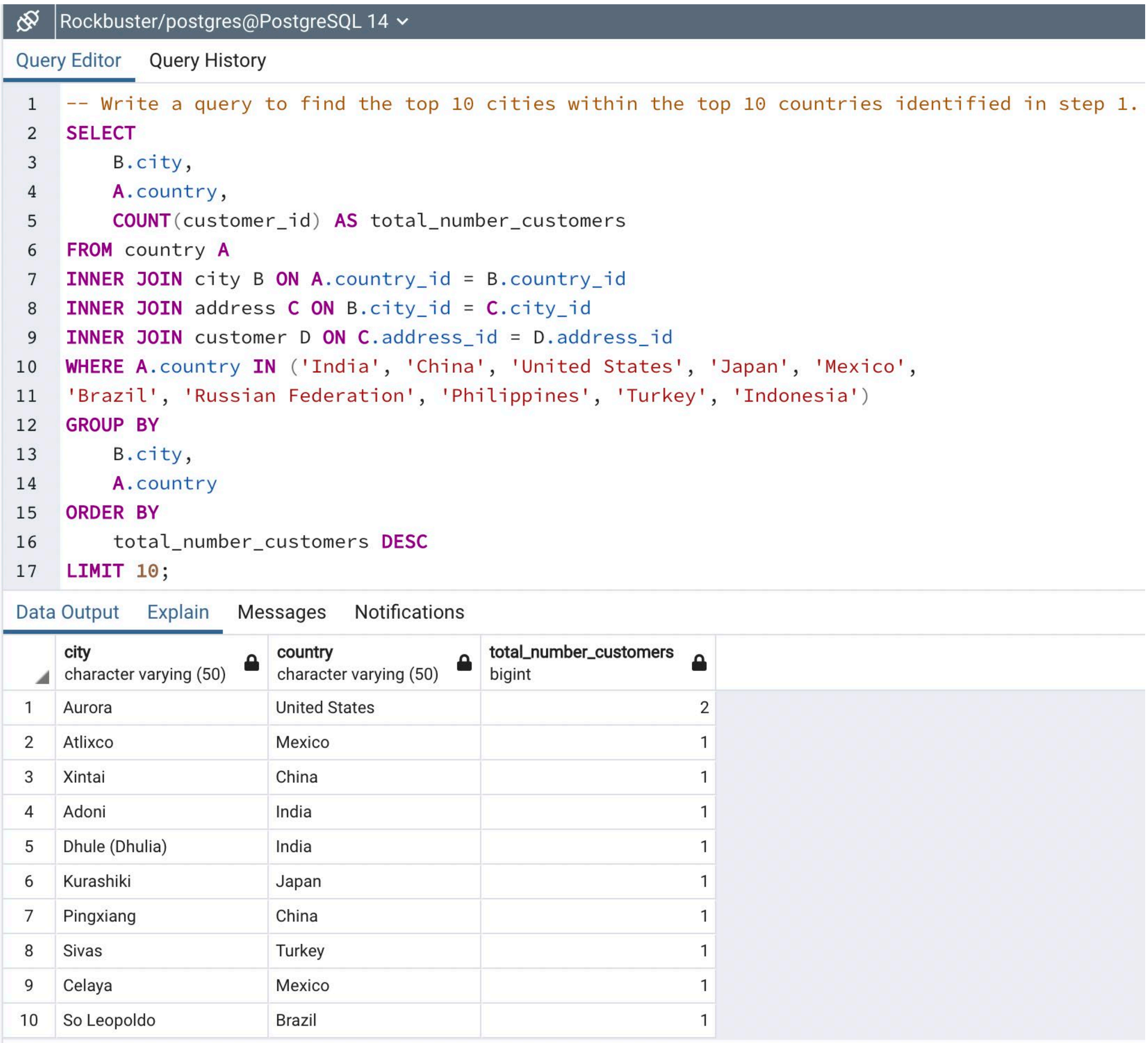
We had four tables in total: country, city, address, and customer. In this case, inner joins are our best alternative because we only need specific information from our two main tables: country and customer. In order to perform the relationships we want, we use the INNER JOIN statement with their corresponding aliases to better understanding.

I used the COUNT statement to count each country's total number of customers and the GROUP BY to isolate our query by country. Moreover, I employed the ORDER BY and LIMIT statements with the aim of retrieving the top 10 countries with more customers in descending order (the largest first).

The following sequence was carried out in order to perform the corresponding joins:

- country_id for country and city tables.
- city_id for city and address tables.
- address_id for address and customer tables.

2. Write a query to find the top 10 cities within the top 10 countries identified in step 1.

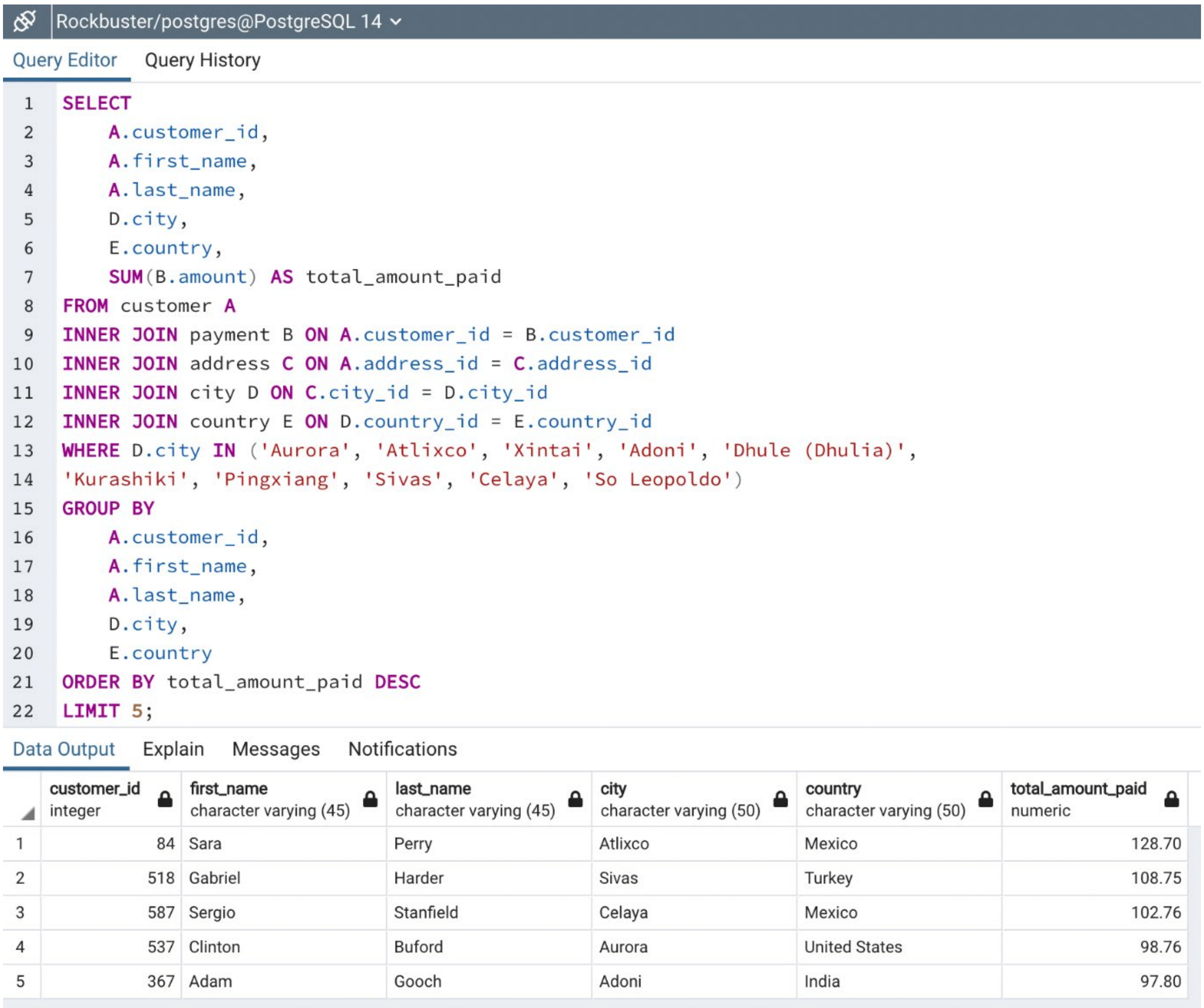


```sql
-- Write a query to find the top 10 cities within the top 10 countries identified in step 1.
SELECT
    B.city,
    A.country,
    COUNT(customer_id) AS total_number_customers
FROM country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
WHERE A.country IN ('India', 'China', 'United States', 'Japan', 'Mexico',
'Brazil', 'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
GROUP BY
    B.city,
    A.country
ORDER BY
    total_number_customers DESC
LIMIT 10;
```

	city character varying (50)	country character varying (50)	total_number_customers bigint
1	Aurora	United States	2
2	Atlixco	Mexico	1
3	Xintai	China	1
4	Adoni	India	1
5	Dhule (Dhulia)	India	1
6	Kurashiki	Japan	1
7	Pingxiang	China	1
8	Sivas	Turkey	1
9	Celaya	Mexico	1
10	So Leopoldo	Brazil	1

For this occasion, we used the same query in step 1 but with some aggregations. Since we have to find cities within our top 10 countries, the WHERE and IN statements are helpful because they allow us the filter we want. In order to see these new cities, we added another column with the SELECT statement. The table could acquire other complementary orders beyond the total number of customers, such as, for example, the alphabetical order by city. For this case, we have not used any additional orders apart from the total number of clients.

3. Write a query to find the top 5 customers in the top 10 cities who have paid the highest total amounts to Rockbuster. The customer team would like to reward them for their loyalty!



```sql
SELECT
    A.customer_id,
    A.first_name,
    A.last_name,
    D.city,
    E.country,
    SUM(B.amount) AS total_amount_paid
FROM customer A
INNER JOIN payment B ON A.customer_id = B.customer_id
INNER JOIN address C ON A.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,
    A.first_name,
    A.last_name,
    D.city,
    E.country
ORDER BY total_amount_paid DESC
LIMIT 5;
```

	customer_id integer	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

As we did in exercise number 2, we took the query from the last task and added some new functions to retrieve our 5 most loyal customers. In detail, we had to add the payout table to our set of tables because only there we have the information about how much money our customers have paid. Through the INNER JOIN statement, we carried out our query with the aim of finding the top 5 customers in the top 10 cities who have paid the highest total amounts to Rockbuster.

We can see that the most loyal customers are at the top because of the ORDER BY tool, oriented in descending order (most money paid first). Since we've already limited our top 10 cities with WHERE + IN, we only need to filter out the top 5 customers with the LIMIT statement at the end of our query.