July 10, 2022

Data Immersion
Databases & SQL for Analysts
Joining Tables of Data
David Guillen Aroche

1. **Get the Top 10 countries where Rockbuster customers are based, so management team can focus on building a better image into those markets (Use GRPUP BY and ORDER BY).**

Query:

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

```
393  SELECT D.country_id,
394         D.country,
395         COUNT(A.customer_id) AS customers_count_by_country
396  FROM customer A
397  INNER JOIN address B ON A.address_id = B.address_id
398  INNER JOIN city C ON B.city_id = C.city_id
399  INNER JOIN country D ON C.country_id = D.country_id
400  GROUP BY D.country_id
401  ORDER BY customers_count_by_country DESC
402  LIMIT 10
```

Data output | Messages | Notifications

	country_id [PK] integer	country character varying (50)	customers_count_by_country bigint
1	44	India	60
2	23	China	53
3	103	United States	36
4	50	Japan	31
5	60	Mexico	30
6	15	Brazil	28
7	80	Russian Federation	28
8	75	Philippines	20
9	97	Turkey	15
10	45	Indonesia	14

Total rows: 10 of 10 | Query complete 00:00:00.076

I decided to join customer (A), address (B), city (C), and country (D) tables, and retrieve the information of the columns "D. *country_id"* and *"D.country",* and *"A.customer_id"* from the customer (A) table, as is the information we need. Then, I grouped the values accrding to countries, as is the Top 10 we are looking for. Finally, I ordered the values by the customers count by country in descendent order so the country with the most customers appears at the top and the country with the least customer appears at the bottom.

2. **Find the Top 10 cities within the Top 10 countries identified in the last query.**

Query:

```
SELECT  C.city,
    COUNT(A.customer_id) AS customers_count_by_city
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D ON C.country_id = D.country_id
WHERE D.country IN ('India',
                    'China',
                    'United States',
                    'Japan',
                    'Mexico',
                    'Brazil',
                    'Russian Federation',
                    'Philippines',
                    'Turkey',
                    'Indonesia')
GROUP BY (C.city)
ORDER BY customers_count_by_city DESC
LIMIT 10
```

Query Query History

```
406  SELECT  C.city,
407         COUNT(A.customer_id) AS customers_count_by_city
408  FROM customer A
409  INNER JOIN address B ON A.address_id = B.address_id
410  INNER JOIN city C ON B.city_id = C.city_id
411  INNER JOIN country D ON C.country_id = D.country_id
412  WHERE D.country IN ('India',
413                      'China',
414                      'United States',
415                      'Japan',
416                      'Mexico',
417                      'Brazil',
418                      'Russian Federation',
419                      'Philippines',
420                      'Turkey',
421                      'Indonesia')
422  GROUP BY (C.city)
423  ORDER BY customers_count_by_city DESC
424  LIMIT 10
```

Data output Messages Notifications

	city character varying (50)	customers_count_by_city bigint
1	Aurora	2
2	Tokat	1
3	Tarsus	1
4	Atlixco	1
5	Emeishan	1
6	Pontianak	1
7	Shimoga	1
8	Aparecida de Goinia	1
9	Zalantun	1
10	Taguig	1

Total rows: 10 of 10 Query complete 00:00:00.059

Following my logic, I decided to base this query on the query from point # 1. Before continuing, I must say, that at even though the instruction is simple, I was wondering at the beginning if I should look for the Top 10 cities per Country—resulting in 100 cities, or different Top 10s for each city—, so I decided to explore how the customers looked in each city with following queries:

```
--How many customers are there?
SELECT customer_id,
    first_name,
    last_name
```

```
FROM customer
ORDER BY customer_id ASC

--how many cities are in the database?
SELECT city_id,
    city,
    COUNT(city) AS city_count
FROM city
GROUP BY (city_id, city)
ORDER BY city_id ASC

--How many customers per city?
SELECT C.city_id,
    C.city,
    COUNT(A.customer_id) AS customers_count_by_city
FROM customer A
LEFT JOIN address B ON A.address_id = B.address_id
LEFT JOIN city C ON B.city_id = C.city_id
GROUP BY (C.city_id, C.city)
ORDER BY customers_count_by_city DESC
```

And found that:

- There are only 599 customers recorded in the data base,
- There are 600 different city IDs, from which two are repeated (312-London, and 313-London)
- There are 597 tuples from the count of customers per city

This is important, because even though we know that we need to get the Top 10 citites within the already queried Top 10 countries with the biggest customers base, it is important to explore the customers base from other perspectives, and ultimately make quality checks in terms of results.

I got to this conclusion because when I queried to answer the point #2, I requested to get the values from *"city_id, "city", "country_id", "country",* and *"customer_count_by_city",* the cities where different when grouped in three different ways:

- *GROUP BY city*
- *GrOUP BY (county_id, country, city)*
- *GROUP BY (country_id, country, city_id, city)*

Given the last information, I needed to know how the customers distribution was along cities; most of the cities only have one customer and two cities have two customers (Aurara with single ID, and London with two IDs).

For didactical purpose, I decided to group by "*city*" only. **Please note that only 8 countries are represented in here. References above.**

3. **Find the Top 10 cities who have paid the highest total amount to Rockbuster.**

Query:

```
SELECT A.customer_id,
    A.first_name,
    A.last_name,
    C.city,
    D.country,
    SUM(E.amount) AS payments_total_amount
FROM customer A
INNER JOIN address B ON A.address_id = B.address_id
INNER JOIN city C ON B.city_id = C.city_id
INNER JOIN country D ON C.country_id = D.country_id
INNER JOIN payment E ON A.customer_id = E.customer_id
WHERE city IN ('Aurora',
        'Tokat',
        'Tarsus',
        'Atlixco',
        'Emeishan',
        'Pontianak',
        'Shimoga',
        'Aparecida de Goinia',
        'Zalantun',
        'Taguig')
GROUP BY (A.customer_id, C.city, D.country)
ORDER BY payments_total_amount DESC
LIMIT 5
```

Query | Query History

```
488 FROM customer A
489 INNER JOIN address B ON A.address_id = B.address_id
490 INNER JOIN city C ON B.city_id = C.city_id
491 INNER JOIN country D ON C.country_id = D.country_id
492 INNER JOIN payment E ON A.customer_id = E.customer_id
493 WHERE city IN ('Aurora',
494                 'Tokat',
495                 'Tarsus',
496                 'Atlixco',
497                 'Emeishan',
498                 'Pontianak',
499                 'Shimoga',
500                 'Aparecida de Goinia',
501                 'Zalantun',
502                 'Taguig')
503 GROUP BY (A.customer_id, C.city, D.country)
504 ORDER BY payments_total_amount DESC
```

Data output | Messages | Notifications

	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	payments_total_amount numeric
1	566	Casey	Mena	Tokat	Turkey	130.68
2	84	Sara	Perry	Atlixco	Mexico	128.7
3	506	Leslie	Seward	Pontianak	Indonesia	123.72
4	389	Alan	Kahn	Emeishan	China	119.75
5	537	Clinton	Buford	Aurora	United States	98.76
6	216	Natalie	Meyer	Aparecida de Goinia	Brazil	90.78
7	72	Theresa	Watson	Taguig	Philippines	88.73
8	238	Nellie	Garrett	Shimoga	India	86.81
9	93	Phyllis	Foster	Zalantun	China	76.8
10	164	Joann	Gardner	Tarsus	Turkey	64.85
11	330	Scott	Shelley	Aurora	United States	60.82

Total rows: 11 of 11 | Query complete 00:00:00.137

This query was a little bit tricky to solve. I decided to use the same structure I used in point #2, changing WHERE, GROUP BY, and ORDER BY; found out that keeping the order of hierarchy was not going to be possible since grouping the payment table as the left-most table (A) was giving me an error stating that the table B was being repeated—so I checked on FROM, and the JOINS, but found no significant errors, typos, or mistakes. So the payment table was left at the right-most order side (E), and joined it with customer (A).

So I selected those columns stated in the question (*"custome_id", "first_name", "last_name", "city", "country", and "amount"* as an aggregate.

The joins started with the tables as follows:

- A, customer
- B, address
- C, city
- D, country,

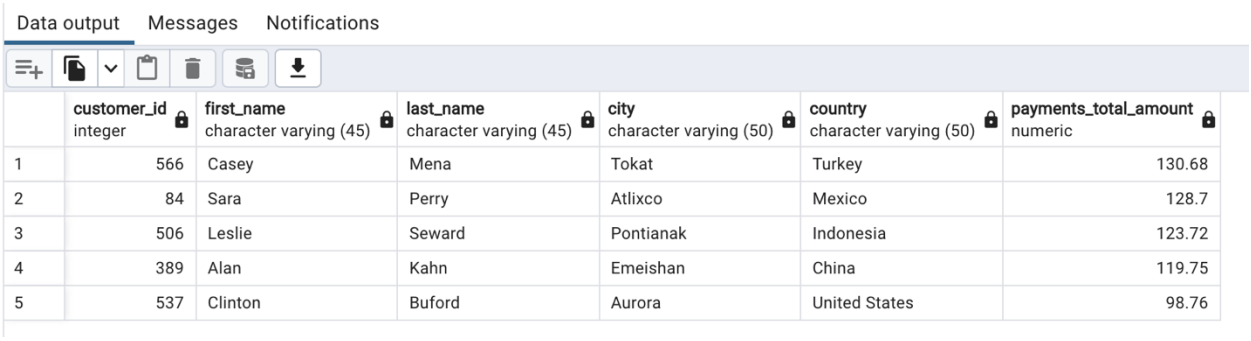
- E, payment.

Then I filtered the values according to the cities we got from point # 2:

1. Aurora, United States
2. Tokat, Turkey
3. Tarsus, Turkey
4. Atlixo, Mexico
5. Emeishan, China
6. Pontionak, Indonesia
7. Shimoga, India
8. Aparecida de Goinia, Brazil
9. Zalantun, China
10. Tugig, Philippines

Please reference to point #2 for information on repeated countries on the list.

Finally, the values where grouped by the total of the payments realized by each customer and ordered in DESC to have the highest value on top and the lowest at the bottom. Finally, the values to retrieve where limited to the five highest values.



Data output | Messages | Notifications

	customer_id integer	first_name character varying (45)	last_name character varying (45)	city character varying (50)	country character varying (50)	payments_total_amount numeric
1	566	Casey	Mena	Tokat	Turkey	130.68
2	84	Sara	Perry	Atlixco	Mexico	128.7
3	506	Leslie	Seward	Pontianak	Indonesia	123.72
4	389	Alan	Kahn	Emeishan	China	119.75
5	537	Clinton	Buford	Aurora	United States	98.76