Data Immersion
Databases & SQL for Analysts
3.8: Performing Subqueries
David Guillen Aroche

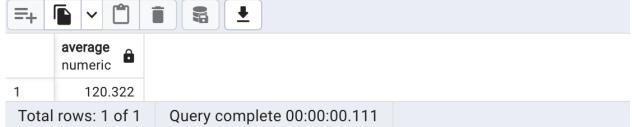
Step 1: Find the average amount paid by the top 5 customers.

1. Copy query wrote on step 3 from task 3.7 (Joining Tables of Data) into the query tool; this will be the subquery, with "total_amount_paid" as its alias.

```
Inner Query (Inner Statement):
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')
AND country IN('India',
               'China',
               'United States',
               'Japan',
               'Mexico',
               'Brazil',
               'Russian Federation',
               'Philippines',
               'Turkey',
               'Indonesia')
GROUP BY (A.customer_id, C.city, D.country)
```

```
Inner S. within Outer S.:
SELECT AVG(total amount paid.total amount paid) AS average
FROM (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')
                  AND country IN('India',
                                  'China',
                                  'United States',
                                   'Japan',
                                   'Mexico',
                                   'Brazil',
                                   'Russian Federation',
                                   'Philippines',
                                   'Turkey',
                                   'Indonesia')
                   GROUP BY (A.customer id, C.city, D.country)
                   ORDER BY payments_total_amount DESC
           LIMIT 5) AS total_amount_paid
```

```
SELECT AVG(total_amount_paid.total_amount_paid) AS average
626
627
     FROM (SELECT A.customer_id,
628
                 A.first_name,
629
                 A.last_name,
630
                 C.city,
631
                 D.country,
632
                 SUM(E.amount) AS total_amount_paid
633
          FROM customer A
         INNER JOIN address B ON A.address_id = B.address_id
634
         INNER JOIN city C ON B.city_id = C.city_id
635
636
         INNER JOIN country D ON C.country_id = D.country_id
637
          INNER JOIN payment E ON A.customer_id = E.customer_id
         WHERE city IN ('Aurora',
638
639
                          'Tokat',
                          'Tarsus',
640
                          'Atlixco'.
641
642
                          'Emeishan',
                          'Pontianak',
643
644
                          'Shimoga',
                          'Aparecida de Goinia',
645
646
                          'Zalantun',
647
                          'Taguig')
         AND country IN('India',
648
649
                          'China',
                          'United States',
650
651
                          'Japan',
652
                          'Mexico',
653
                          'Brazil',
654
                          'Russian Federation',
Data output
                      Notifications
            Messages
```



2. Find out how many of the Top 5 Customers are vases within each country. flnal output should include 3 columns: "country", "all_customer_count", and "top_customer_count"

Outer Query (counting the number of customers living in each country):

SELECT DISTINCT D.country,

COUNT(DISTINCT customer_id) AS all_customer_count

FROM customer A

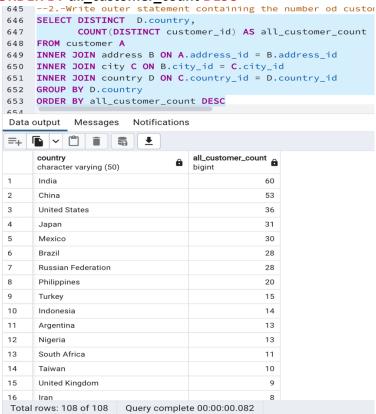
INNER JOIN address B ON A.address id = B.address id

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

ORDER BY all customer count DESC



SELECT DISTINCT D.country,

COUNT(DISTINCT A.customer_id) AS all_customer_count,

COUNT(DISTINCT D.country) AS top_customer_count

FROM customer A

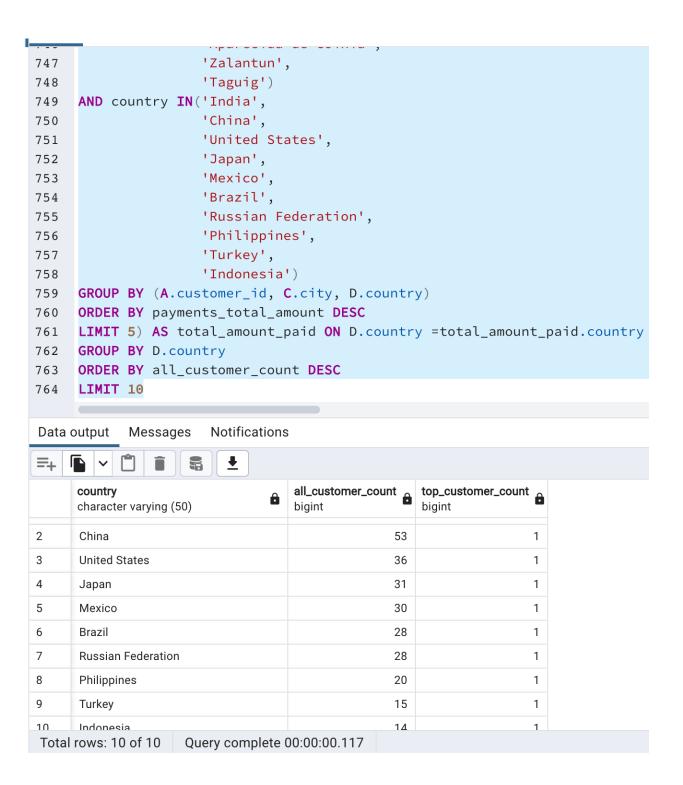
INNER JOIN address B ON A.address_id = B.address_id

INNER JOIN city C ON B.city id = C.city id

INNER JOIN country D ON C.country_id = D.country_id

LEFT JOIN (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')
                  AND country IN('India',
                                  'China',
                                  'United States',
                                   'Japan',
                                   'Mexico',
                                   'Brazil',
                                   'Russian Federation',
                                   'Philippines',
                                   'Turkey',
                                  'Indonesia')
                  GROUP BY (A.customer_id, C.city, D.country)
                   ORDER BY payments total amount DESC
                  LIMIT 5) AS total_amount_paid ON D.country = total_amount_paid.country
GROUP BY D.country
ORDER BY all_customer_count DESC
LIMIT 10
```



3. Comment on the following:

Could we have achieved results without using subqueries in both steps?

In **STEP 1** we could have had results without using a query, either specifying the IDs from customers with WHERE or using HAVING and specifying the values.

In **STEP 2** it was necessary to have/use the generated table total_amount_paid, not possible to retrieve any results if that subquery element is not present.

O When do you think subqueries are useful?

When we need to determine several different sets of data points (or single data points) that are within the database but is not necessarily explicit, or when we need to perform operations or other functions that need information that has to be processes or calculated first. In general, when we have to answer several questions which content or information is not necessarily related and/or needs to be factorized in order to use it with other set of columns/tables.