

Answers 3.8

Step 1)

a) After taking the query obtained in the step 3 from Exercise 3.7, I wrapped the query in parentheses and give it an alias "total_amount_paid".

Updated Query (as a Inner query) is :

Query	Query History
1	(
2	SELECT c.customer_id,
3	c.first_name,
4	c.last_name,
5	co.country AS country_name,
6	ci.city AS city_name,
7	SUM(p.amount) AS total_amount_paid
8	FROM customer c
9	INNER JOIN address a ON c.address_id = a.address_id
10	INNER JOIN city ci ON a.city_id = ci.city_id
11	INNER JOIN country co ON ci.country_id = co.country_id
12	INNER JOIN payment p ON c.customer_id = p.customer_id
13	WHERE ci.city IN ('Aurora', 'Atxlixo', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
14	'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
15	GROUP BY c.customer_id, c.first_name, c.last_name, co.country, ci.city
16	ORDER BY total_amount_paid DESC
17	LIMIT 5
18) AS total_amount_paid
19	

b) The outer query should calculate the average of the total_amount_paid from the subquery, so we have to use the subquery as a virtual table in the FROM clause.

Full query (outer query + subquery)

Query	Query History
1	SELECT AVG(total_amount_paid.total_amount_paid) AS average
2	FROM (
3	SELECT c.customer_id,
4	c.first_name,
5	c.last_name,
6	co.country AS country_name,
7	ci.city AS city_name,
8	SUM(p.amount) AS total_amount_paid
9	FROM customer c
10	INNER JOIN address a ON c.address_id = a.address_id
11	INNER JOIN city ci ON a.city_id = ci.city_id
12	INNER JOIN country co ON ci.country_id = co.country_id
13	INNER JOIN payment p ON c.customer_id = p.customer_id
14	WHERE ci.city IN ('Aurora', 'Atxlixo', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
15	'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
16	GROUP BY c.customer_id, c.first_name, c.last_name, co.country, ci.city
17	ORDER BY total_amount_paid DESC
18	LIMIT 5
19) AS total_amount_paid;

c) The subquery was added in the FROM clause of the outer query with the alias total_amount_paid, allowing the outer query to reference its results.

d) The alias average was added to the outer query to clearly define the output of the calculation.

Average total amount paid by the top 5 customers:

average
100.37200000000000

Step 2)

a) inner query: identifies the top 5 customers based on their total payments, by wrapping it in parentheses and alias it as top_5_customers.

```
Query  Query History
1  (
2      SELECT c.customer_id,
3             c.first_name,
4             c.last_name,
5             co.country AS country_name,
6             ci.city AS city_name,
7             SUM(p.amount) AS total_amount_paid
8      FROM customer c
9      INNER JOIN address a ON c.address_id = a.address_id
10     INNER JOIN city ci ON a.city_id = ci.city_id
11     INNER JOIN country co ON ci.country_id = co.country_id
12     INNER JOIN payment p ON c.customer_id = p.customer_id
13     WHERE ci.city IN ('Aurora', 'Atxlixo', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
14                      'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
15     GROUP BY c.customer_id, c.first_name, c.last_name, co.country, ci.city
16     ORDER BY total_amount_paid DESC
17     LIMIT 5
18 ) AS top_5_customers;
```

b) We need total customer count grouped by country query using COUNT(DISTINCT) and Alias the result as all_customer_count.

```
SELECT co.country AS country_name,
       COUNT(DISTINCT c.customer_id) AS all_customer_count
FROM customer c
INNER JOIN address a ON c.address_id = a.address_id
INNER JOIN city ci ON a.city_id = ci.city_id
INNER JOIN country co ON ci.country_id = co.country_id
GROUP BY co.country;
```

c)

This Combined Query is made by total customer count (outer query) with the top 5 customers (inner query) using a LEFT JOIN:

```
SELECT country_data.country_name,
       country_data.all_customer_count,
       COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
FROM (
    SELECT co.country AS country_name,
           COUNT(DISTINCT c.customer_id) AS all_customer_count
    FROM customer c
```

```

INNER JOIN address a ON c.address_id = a.address_id
INNER JOIN city ci ON a.city_id = ci.city_id
INNER JOIN country co ON ci.country_id = co.country_id
GROUP BY co.country
) AS country_data
LEFT JOIN (
    SELECT c.customer_id,
           co.country AS country_name
    FROM customer c
    INNER JOIN address a ON c.address_id = a.address_id
    INNER JOIN city ci ON a.city_id = ci.city_id
    INNER JOIN country co ON ci.country_id = co.country_id
    INNER JOIN payment p ON c.customer_id = p.customer_id
    WHERE ci.city IN ('Aurora', 'Atlixo', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
                     'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
    GROUP BY c.customer_id, co.country
    ORDER BY SUM(p.amount) DESC
    LIMIT 5
) AS top_5_customers
ON country_data.country_name = top_5_customers.country_name
GROUP BY country_data.country_name, country_data.all_customer_count;

```

d) In the combined query above:

- The first subquery is aliased as country_data (total customer count per country).
- The second subquery is aliased as top_5_customers.

These aliases make it easier to reference the data during the LEFT JOIN.

e) The join is performed on the country name because it is the common column between the two subqueries:

```

ON country_data.country_name = top_5_customers.country_name

```

f) The complete query including the calculation for top_customer_count is:

```

SELECT country_data.country_name,
       country_data.all_customer_count,
       COUNT(DISTINCT top_5_customers.customer_id) AS top_customer_count
FROM (
    SELECT co.country AS country_name,
           COUNT(DISTINCT c.customer_id) AS all_customer_count
    FROM customer c
    INNER JOIN address a ON c.address_id = a.address_id
    INNER JOIN city ci ON a.city_id = ci.city_id
    INNER JOIN country co ON ci.country_id = co.country_id
    GROUP BY co.country
) AS country_data
LEFT JOIN (
    SELECT c.customer_id,
           co.country AS country_name
    FROM customer c
    INNER JOIN address a ON c.address_id = a.address_id

```

```

INNER JOIN city ci ON a.city_id = ci.city_id
INNER JOIN country co ON ci.country_id = co.country_id
INNER JOIN payment p ON c.customer_id = p.customer_id
WHERE ci.city IN ('Aurora', 'Atlixo', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
                  'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY c.customer_id, co.country
ORDER BY SUM(p.amount) DESC
LIMIT 5
) AS top_5_customers
ON country_data.country_name = top_5_customers.country_name
GROUP BY country_data.country_name, country_data.all_customer_count;

```

Top 5 customers for the third column using GROUP BY and COUNT (DISTINCT), give this column the alias "top_customer_count"

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

Step 3)

- Yes, steps 1 and 2 could theoretically be done without using subqueries, but it would make the SQL queries much more complex and harder to read. Without subqueries, we would need to create multiple joins and filters in a single query to replicate the same logic, which would result in a very long and difficult-to-maintain SQL statement. For example, identifying the top 5 customers would require embedding all filtering, grouping, and ordering operations into one main query, rather than isolating them in manageable subqueries.
- Subqueries are particularly useful when you need to filter, aggregate, or compare data before performing other operations. They allow us to modularize the logic by creating temporary tables or values that simplify the main query. For instance, in Step 1, the subquery identified the top 5 customers, which the outer query then used to calculate the average payment. Similarly, the subquery isolated the top 5 customers' data, which was then joined with the total customer count per country.