PERFORMING SUBQUERIES

Ann Mariya Francis

Exercise 3.8

5 Feb 2025

---

1)

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

```
        JOIN city C ON B.city_id = C.city_id
        JOIN country D ON C.country_id = D.country_id
        GROUP BY D.country
        ORDER BY COUNT(A.customer_id) DESC
        LIMIT 10
    )
    GROUP BY D.country, C.city
    ORDER BY COUNT(A.customer_id) DESC
    LIMIT 10
)
GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
ORDER BY total_amount_paid DESC
LIMIT 5
) AS total_amount_paid;
```

Query Query History

```
SELECT AVG(total_amount_paid) AS average
FROM (
    SELECT B.customer_id,
           B.first_name,
           B.last_name,
           E.country,
           D.city,
           SUM(A.amount) AS total_amount_paid
    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 (E.country, D.city) IN (
        SELECT D.country, C.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
```

Data Output Messages Notifications

SQL Showing rows: 1 to 1

	average numeric
1	105.5540000000000000

2)

```
SELECT
    E.country,
```

```
    COUNT(DISTINCT B.customer_id) AS all_customer_count,
    COUNT(DISTINCT top_5_customers.customer_id) 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, E.country
    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 (E.country, D.city) IN (
        SELECT D.country, C.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 (
            SELECT D.country
            FROM customer A
            JOIN address B ON A.address_id = B.address_id
            JOIN city C ON B.city_id = C.city_id
            JOIN country D ON C.country_id = D.country_id
            GROUP BY D.country
            ORDER BY COUNT(A.customer_id) DESC
            LIMIT 10
        )
```

```
    GROUP BY D.country, C.city
    ORDER BY COUNT(A.customer_id) DESC
    LIMIT 10
  )
  GROUP BY B.customer_id, B.first_name, B.last_name, D.city, E.country
  ORDER BY SUM(A.amount) DESC
  LIMIT 5
) AS top_5_customers
ON E.country = top_5_customers.country
GROUP BY E.country
ORDER BY top_customer_count DESC;
```

Query | Query History

```
SELECT
    E.country,
    COUNT(DISTINCT B.customer_id) AS all_customer_count,
    COUNT(DISTINCT top_5_customers.customer_id) 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, E.country
    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
```

Data Output | Messages | Notifications

Showing rows: 1 to 108 | Page No: 1

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	1
2	India	60	1
3	China	53	1
4	United States	36	1
5	Japan	31	1
6	Argentina	13	0

3.1)

Yes, both steps can be done without subqueries by using JOINs and GROUP BY. Instead of using subqueries to find the top 5 customers, we can first calculate the total amount paid by each customer and then use ORDER BY and LIMIT to get the top 5. For counting top customers per country, we can join this result with the total customer count per country using a LEFT JOIN. This approach keeps the query simpler, more efficient, and easier to read without relying on subqueries.

3.2)

Subqueries are useful when you need to break down complex logic into smaller, manageable parts. They are great for filtering data based on aggregates like SUM() or COUNT(), especially when you can't easily access those aggregates in the outer query. Subqueries can also simplify queries by avoiding unnecessary joins or by returning scalar values, such as the maximum or minimum from a set of data.