

3.8: Performing Subqueries.

1. Copy the query from step 3 of task 3.7 into the Query Tool and add parentheses around it. This will be your inner query
2. Write an outer statement that counts the number of customers living in each country. You'll need to refer to your entity relationship diagram or data dictionary in order to do this. The information you need is in different tables, so you'll have to use a `JOIN`. To get the count for each country, use `COUNT (DISTINCT)` and `GROUP BY`. Give your second column the alias "`all_customer_count`" for readability.
3. Place your inner query in the outer query. Since you want to merge the entire output of the outer query with the information from your inner query, use a left join to connect the two queries on the "country" column. You'll need to add a `LEFT JOIN` after your outer query, followed by the subquery in parentheses.
4. Give your subquery an alias so you can refer to it in your outer query, for example, "`top_5_customers`".
5. Remember to specify which columns to join the two tables on using `ON`. Both `ON` and the column names should follow the alias.
6. Count the top 5 customers for the third column using `GROUP BY` and `COUNT (DISTINCT)`. Give this column the alias "`top_customer_count`".

```
SELECT
    D.country,
    COUNT(DISTINCT A.customer_id) AS all_customer_count,
    COUNT(DISTINCT top_5_customers.customer_id) 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,
        B.address,
        C.city,
        D.country,
        SUM(E.amount) AS total_amount_paid
    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', 'Acua', 'Citrus Heights', 'Iwaki', 'Ambattur', 'Shanwei', 'So Leopoldo', 'Teboksary', 'Tianjin', 'Cianjur')
    GROUP BY A.customer_id, A.first_name, A.last_name, B.address, C.city, D.country
    ORDER BY total_amount_paid DESC
    LIMIT 5
) AS top_5_customers
ON A.customer_id = top_5_customers.customer_id
GROUP BY D.country
ORDER BY top_customer_count DESC
LIMIT 5
```

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

1. Write 1 to 2 short paragraphs on the following:

- Do you think steps 1 and 2 could be done without using subqueries?
- When do you think subqueries are useful

Yes, I think, it would be possible to create several queries and use the outcome of each one as input for the next one. The problem is that it would take quite long and everything would need to be done manually. Furthermore, the information would need to be updated manually every time.

When data are updated frequently and a result requires more than one query, subqueries are very useful. They are an excellent way to make code more efficient and to avoid unnecessary lines of code.