

## Databases & SQL for Analysts


### 3.8: Performing Subqueries

#### Answers 3.8

Step 1:

```
--This query to find out the average payment by the top 5 customers in top 10 cities.

SELECT ROUND(AVG(total_amount),2)AS avg_amount
  FROM (SELECT e.customer_id,
              first_name,
              last_name,
              city,
              country,
              SUM(amount)AS total_amount
        FROM country AS a
       INNER JOIN city AS b
          ON a.country_id=b.country_id
       INNER JOIN address AS c
          ON b.city_id=c.city_id
       INNER JOIN customer AS d
          ON c.address_id=d.address_id
       INNER JOIN payment AS e
          ON d.customer_id=e.customer_id
      WHERE city IN('Aurora',
                    'Shimoga',
                    'Aparecida de Goiania',
                    'Emeishan',
                    'Pontianak',
                    'Tarsus',
                    'Atlixco',
                    'Zalantun',
                    'Rio Claro',
                    'Tokat'))
 GROUP BY e.customer_id,
          first_name,
          last_name,
          city,
          country
 ORDER BY total_amount DESC
 LIMIT 5) AS subquery
```

	avg_amount numeric 
1	120.32

Step 2:

```

SELECT a.country,
       COUNT(DISTINCT d.customer_id)AS all_customer_count,
       COUNT(DISTINCT (a.country))AS top_customer_count
FROM country AS a
INNER JOIN city AS b
  ON a.country_id=b.country_id
INNER JOIN address AS c
  ON b.city_id=c.city_id
INNER JOIN customer AS d
  ON c.address_id=d.address_id
LEFT JOIN (SELECT e.customer_id,
                  first_name,
                  last_name,
                  city,
                  country,
                  SUM(amount)AS total_amount
            FROM country AS a
            INNER JOIN city AS b
              ON a.country_id=b.country_id
            INNER JOIN address AS c
              ON b.city_id=c.city_id
            INNER JOIN customer AS d
              ON c.address_id=d.address_id
            INNER JOIN payment AS e
              ON d.customer_id=e.customer_id
           WHERE city IN('Aurora',
                        'Shimoga',
                        'Aparecida de Goinia',
                        'Emeishan',
                        'Pontianak',
                        'Tarsus',
                        'Atlixco',
                        'Zalantun',
                        'Rio Claro',
                        'Tokat'))
  GROUP BY e.customer_id,
           first_name,
           last_name,
           city,
           country
ORDER BY total_amount DESC
LIMIT 5) AS top_5_customer
  ON a.country=top_5_customer.country
GROUP BY a.country,
         top_5_customer
ORDER BY all_customer_count DESC
LIMIT 5

```

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

### Step 3:

- There is no way to query the answer without the subquery method. The first one we asked about the average amount from the top 5 customers in top 10 cities, we need sum the amount through all customers then sorted them out as descending from the to the least and then limit 5. After that we need calculate the average amount through these 5 customers, using aggregation to aggregate another result is impossible appear in the same column, like `AVG(SUM(AMOUNT))`, that's means aggregate function cannot be nested. To get the result we must use subquery in these two questions.
- The conditions when we choose the subquery are:
  - 1) When the results of one query need to be used in another query
  - 2) Usually when we query the answer from a table that basically asked about the aggregate data value like average, maximum and so on to compare with the regular value, subquery is the best to get the answer.
  - 3) Subquery allows us to use inner query in the select statement and it helps us to create another column which we cannot retrieve from the main query.
  - 4) When there are a large number of rows in where condition to filter, using subquery will be more efficiency.