Task 3.8

1.

Query | Query History

```
SELECT AVG (total_amount_paid.total_amount) AS Average
From (SELECT a.customer_id,
       a.first_name,
       a.last_name,
       d.country,
       c.city,
       sum (e.amount) as 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 c.city IN ('Aurora','Tokat','Tarsus','Atlixco','Emeishan','Pontianak','Shimoga','Apar
GROUP BY a.customer_id,
         first_name,
         last_name,
         country,
         city
ORDER BY total_amount desc
limit 5) AS total_amount_paid
```

	average numeric
1	120.3220000000000000

2.

Query | Query History

```sql
SELECT d.country,
       COUNT(DISTINCT top_5_customers.customer_id) AS top_5_customer_count,
       COUNT(DISTINCT a.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
LEFT JOIN (SELECT a.customer_id,
       a.first_name,
       a.last_name,
       d.country,
       c.city,
       sum (e.amount) as 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 c.city IN ('Aurora','Tokat','Tarsus','Atlixco','Emeishan','Pontianak','Shimoga','Apa
GROUP BY a.customer_id,
         first_name,
         last_name,
         country,
         city
ORDER BY total_amount desc
limit 5 ) AS top_5_customers ON d.country = top_5_customers.country
GROUP BY d.country
ORDER BY top_5_customer_count desc
LIMIT 10
```

	country character varying (50)	top_5_customer_count bigint	all_customer_count bigint
1	Turkey	1	15
2	China	1	53
3	Mexico	1	30
4	United States	1	36
5	Indonesia	1	14
6	Armenia	0	1
7	Argentina	0	13
8	American Samoa	0	1
9	Bahrain	0	1
10	Anguilla	0	1

3.

- Do you think steps 1 and 2 could be done without using subqueries?

A/ Step one could be done without using subqueries by using the aggregate function and ordering the result descending, step 2 in the other hand wouldn't be possible to do without subqueries unless we create a new table which consume more time than using subqueries.

- When do you think subqueries are useful?

A/ They are useful when we have to make use of information that is constantly changing, creating a subquery would save us more time than running 2 queries and modifying the main one every time we need to use the data