

3.8: Performing Subqueries

1: Find the average amount paid by the top 5 customers.

Query Editor

Query History

```
1 SELECT AVG(total_amount_paid)
2 FROM
3 (SELECT A.customer_id, A.first_name, A.last_name, C.city, D.country,
4      SUM(E.amount) AS total_amount_paid
5 FROM customer A
6 INNER JOIN address B ON A.address_id = B.address_id
7 INNER JOIN city C ON B.city_id = C.city_id
8 INNER JOIN country D ON C.country_ID = D.country_ID
9 INNER JOIN payment E ON A.customer_id = E.customer_id
10 WHERE city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
11               'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
12 GROUP BY A.customer_id, C.city, D.country
13 ORDER BY total_amount_paid DESC
14 LIMIT 5) AS average
```

Data Output

Explain

Messages

Notifications

	avg numeric	
1	107.354	

2: Find out how many of the top 5 customers are based within each country.

Query Editor

Query History

```

1  SELECT D.country, COUNT (DISTINCT A.customer_id) AS all_customer_count,
2         COUNT (DISTINCT top_5_customers) AS top_customer_count
3  FROM customer A
4  INNER JOIN address B ON A.address_id = B.address_id
5  INNER JOIN city C ON B.city_id = C.city_id
6  INNER JOIN country D ON C.country_id = D.country_id
7  LEFT JOIN (SELECT A.customer_id, A.first_name, A.last_name, C.city, D.country,
8              SUM(E.amount) AS total_amount_paid
9            FROM customer A
10           INNER JOIN address B ON A.address_id = B.address_id
11           INNER JOIN city C ON B.city_id = C.city_id
12           INNER JOIN country D ON C.country_ID = D.country_ID
13           INNER JOIN payment E ON A.customer_id = E.customer_id
14          WHERE D.country IN ('India', 'China', 'United States', 'Japan', 'Mexico',
15                              'Brazil', 'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
16          AND C.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)',
17                        'Kurashiki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
18         GROUP BY A.customer_id, C.city, D.country
19         ORDER BY total_amount_paid DESC LIMIT 5) AS top_5_customers ON D.country = top_5_customers.country
20 GROUP BY D.country
21 ORDER BY top_customer_count DESC

```

Data Output

Explain

Messages

Notifications

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	2
2	Turkey	15	1
3	India	60	1
4	United States	36	1

3. Step 1 can be completed without a subquery by using an aggregate function. You need a subquery for step two because there are multiple tables you need to look at and the subquery allows you to do that without creating an additional table. Subqueries are useful when working with data that is continuously updated or tables that are reliant on each other with one of them, the inner, continuously being updated. You save time by having the subquery instead of having to run two separate queries.