

MPivec_Exercise 3.8: Performing Subqueries

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

a. Inner Statement

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

i.

b. Outer Statement

i. SELECT AVG(total_amount_paid) AS average
FROM (Inner Statement) AS total_amount_paid;

c. Final Statement (with rounding included for dollar value)

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

Data output Messages Notifications

	average numeric
1	107.35

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

Query Query History

```

1 SELECT DISTINCT(D.country),
2     COUNT(DISTINCT A.customer_id) AS all_customer_count,
3     COUNT(DISTINCT D.country) AS top_customer_count
4 FROM country D
5 INNER JOIN city C ON D.country_id = C.country_id
6 INNER JOIN address B ON C.city_id = B.city_id
7 INNER JOIN customer A ON B.address_id = A.address_id
8 LEFT JOIN (SELECT A.customer_id,
9     A.first_name,
10    A.last_name,
11    C.city,
12    D.country,
13    SUM(E.amount) AS total_amount_paid
14 FROM customer A
15 INNER JOIN address B ON A.address_id = B.address_id
16 INNER JOIN city C ON B.city_id = C.city_id
17 INNER JOIN country D ON C.country_id = D.country_id
18 INNER JOIN payment E ON A.customer_id = E.customer_id
19 WHERE C.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule(Dhulia)','Kurashiki','Pingxiang','Sivas','Celaya'
20 GROUP BY A.customer_id, A.first_name, A.last_name, C.city, D.country
21 ORDER BY total_amount_paid DESC
22 LIMIT 5) AS top_5_customers
23     ON D.country=top_5_customers.country
24 GROUP BY D.country, top_5_customers
25 ORDER BY all_customer_count DESC
26 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

a.

3. Write 1-2 short paragraphs on the following

a. Do you think steps 1 and 2 could have been done without using subqueries?

i. I don't think these steps could have been done without using subqueries. The outer statement is based on the inner statement, which was created in the first place because the data is always changing - the top 5 customers and the average amount are always changing based on current spend.

b. When do you think subqueries are useful?

i. Subqueries are useful in a few situations. The SELECT clause can be used to create new columns in your data output, which is helpful for comparing different data points. The WHERE clause allows us to filter results based on ever changing data, which is helpful when you need real time information. The FROM clause is useful when you need to analyze results of an already complex query.