Answers 3.8 Performing Subqueries

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

Query

```
SELECT AVG(total_amount_paid.total_amount_paid) AS average
FROM
(SELECT cu.customer_id, cu.first_name, cu.last_name, ci.city, co.country, SUM(p.amount) AS
total_amount_paid
FROM customer cu
INNER JOIN address a on a.address_id = cu.address_id
INNER JOIN city ci on a.city_id = ci.city_id
INNER JOIN country co on ci.country_id = co.country_id
INNER JOIN payment p on cu.customer_id = p.customer_id
WHERE ci.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule
(Dhulia)','Kurashiki','Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY cu.customer_id, ci.city, co.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS total_amount_paid
```

Output

average
107.354

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

Query

```
SELECT co.country, COUNT(DISTINCT(cu.customer_id)) AS all_customer_count,
COUNT(DISTINCT(top_5_customers.customer_id)) AS top_customer_count
FROM COUNTRY co
INNER JOIN city ci on ci.country_id = co.country_id
INNER JOIN address a on a.city_id = ci.city_id
INNER JOIN customer cu on cu.address_id = a.address_id
```

```
LEFT JOIN (SELECT cu.customer_id, cu.first_name, cu.last_name, ci.city, co.country, SUM(p.amount) AS
total_amount_paid
FROM customer cu
INNER JOIN address a on a.address_id = cu.address_id
INNER JOIN city ci on a.city_id = ci.city_id
INNER JOIN country co on ci.country_id = co.country_id
INNER JOIN payment p on cu.customer_id = p.customer_id
WHERE ci.city IN ('Aurora','Atlixco','Xintai','Adoni','Dhule
(Dhulia)','Kurashiki','Pingxiang','Sivas','Celaya','So Leopoldo')
GROUP BY cu.customer_id, ci.city, co.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_5_customers ON top_5_customers.country = co.country
GROUP BY co.country
HAVING COUNT(DISTINCT(top_5_customers.customer_id)) <> 0
order by top_customer_count DESC
```

Output

country	all_customer_count	top_customer_count
Mexico	30	2
India	60	1
Turkey	15	1
United States	36	1

3. Write 1 to 2 short paragraphs on the following:
    a. Do you think steps 1 and 2 could be done without using subqueries?
    b. When do you think subqueries are useful?

Based on my limited understanding of Common Table Expressions, both queries can probably be completed without using subqueries by using common table expressions instead to create a temporary result sets that can be referenced within SELECT statements. I attempted to use a nested aggregate AVG(SUM()) instead of a subquery for step 1 and discovered that this isn't permitted. With a CTE, I could create the same dataset created with the subquery and calculate the average referencing the CTE. A CTE could also be used to create a temporary result set of the top 5 customers for step 2, which could then be called on in a SELECT statement to find the number of top 5 customs in each country. Subqueries are useful when the output you want requires more than one query and the subquery provides the results that can be used by the outer query. It saves you from having to write separate queries and manually

insert the results. This is also useful when the results of the subquery are constantly changing as it ensures that accurate and up-to-date results are used.