Answers 3.8: Performing Subqueries

Questions

Step 1

Find the average amount paid by the top 5 customers.

1.
```sql
SELECT AVG (total_amount_paid) AS avg_amount_paid_top_five_customers
FROM
(SELECT SUM(pmt.amount) AS total_amount_paid
FROM payment AS pmt
INNER JOIN customer AS cust1 ON pmt.customer_id = cust1.customer_id
INNER JOIN address AS addr1 ON cust1.address_id = addr1.address_id
INNER JOIN city AS cty1 ON addr1.city_id = cty1.city_id
INNER JOIN country AS cnt1 ON cty1.country_id = cnt1.country_id
WHERE cty1.city IN
(SELECT cty2.city
FROM customer AS cust2
INNER JOIN address AS addr2 ON cust2.address_id = addr2.address_id
INNER JOIN city AS cty2 ON addr2.city_id = cty2.city_id
INNER JOIN country AS cnt2 ON cty2.country_id = cnt2.country_id
WHERE cnt2.country IN
(SELECT cnt3.country
FROM customer AS cust3
INNER JOIN address AS addr3 ON cust3.address_id = addr3.address_id
INNER JOIN city AS cty3 ON addr3.city_id = cty3.city_id
INNER JOIN country AS cnt3 ON cty3.country_id = cnt3.country_id
GROUP BY cnt3.country
ORDER BY COUNT(cust3.customer_id) DESC
LIMIT 10)
GROUP BY cty2.city
ORDER BY COUNT(cust2.customer_id) DESC
LIMIT 10)
GROUP BY cust1.customer_id
ORDER BY total_amount_paid DESC
LIMIT 5) AS average;
```

	avg_amount_paid_top_five_customers numeric
1	120.3220000000000000

Step 2

Find out how many of the top 5 customers you identified in step 1 are based within each country.

1.
```sql
SELECT cnt1.country,
COUNT(DISTINCT cust1.customer_id) AS all_customer_count,
COUNT (DISTINCT top_five_customers.country) AS top_customer_count
FROM customer AS cust1
INNER JOIN address AS addr1 ON cust1.address_id = addr1.address_id
INNER JOIN city AS cty1 ON addr1.city_id = cty1.city_id
INNER JOIN country AS cnt1 ON cty1.country_id = cnt1.country_id
LEFT JOIN
(SELECT cust2.customer_id,cty2.city,cnt2.country,
SUM(pmt.amount) AS total_amount_paid
FROM payment AS pmt
INNER JOIN customer AS cust2 ON pmt.customer_id = cust2.customer_id
INNER JOIN address AS addr2 ON cust2.address_id = addr2.address_id
INNER JOIN city AS cty2 ON addr2.city_id = cty2.city_id
INNER JOIN country AS cnt2 ON cty2.country_id = cnt2.country_id
WHERE cty2.city IN
(SELECT cty3.city
FROM customer AS cust3
INNER JOIN address AS addr3 ON cust3.address_id = addr3.address_id
INNER JOIN city AS cty3 ON addr3.city_id = cty3.city_id
INNER JOIN country AS cnt3 ON cty3.country_id = cnt3.country_id
WHERE cnt3.country IN
(SELECT cnt4.country
FROM customer AS cust4
```

```
INNER JOIN address AS addr4 ON cust4.address_id = addr4.address_id
INNER JOIN city AS cty4 ON addr4.city_id = cty4.city_id
INNER JOIN country AS cnt4 ON cty4.country_id = cnt4.country_id
GROUP BY cnt4.country
ORDER BY COUNT(cust4.customer_id) DESC
LIMIT 10)
GROUP BY cty3.city
ORDER BY COUNT(cust3.customer_id) DESC
LIMIT 10)
GROUP BY cust2.customer_id, cty2.city,cnt2.country
ORDER BY total_amount_paid DESC
LIMIT 5) AS top_five_customers ON top_five_customers.country = cnt1.country
GROUP BY cnt1.country
ORDER BY top_customer_count DESC;
```

	country character varying (50)	all_customer_count bigint	top_customer_count bigint
1	Mexico	30	1
2	Turkey	15	1
3	China	53	1
4	United States	36	1
5	Indonesia	14	1

Step 3

Write 1 to 2 short paragraphs on the following:

1. Do you think steps 1 and 2 could be done without using subqueries?
2. They could be. The average payment could be found and calculated manually after the top five customers were found. The location data of the top five customers can be calculated separately without needing to add a subquery. However, nesting the query reduces the search cost and ensures that the query is pulling the right information in each part (query and subquery). More individual queries means more points to make mistakes.
3. When do you think subqueries are useful?
    a. Nesting the query allows us to show our work and ensure that each part (query and subquery) are looking at the right tables / pulling the right information. You would want to use subqueries when working with large tables that are constantly updated. Shipping ledgers are a good example.