TASK 3.8 – Performing Subqueries

Step 1:

```
--Average amount paid by the top 5 customers
SELECT AVG(total_amount_paid.total_amount_paid) AS average
FROM (SELECT A.customer_id,
             A.first_name,
             A.last_name,
             D.country,
             C.city,
             SUM(E.amount) AS total_amount_paid
    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 city IN ('Aurora','Acua','Citrus Heights','Iwaki','Ambattur','Shanwei','So Leopold
    GROUP BY A.customer_id,
        D.country,
        C.city
    ORDER BY total_amount_paid DESC
    LIMIT 5) AS total_amount_paid;
```

Data Output | Explain | Messages | Notifications

| | average numeric |
|---|---|
| 1 | 105.5540000000000000 |

Step 2:

```sql
--How many top 5 customers are based within each country
SELECT DISTINCT (A.country),
    COUNT (DISTINCT D.customer_id) AS all_customer_count,
    COUNT (DISTINCT A.country) AS top_customer_count
FROM Country A
INNER JOIN city B ON A.country_id = B.country_id
INNER JOIN address C ON B.city_id = C.city_id
INNER JOIN customer D ON C.address_id = D.address_id
LEFT JOIN (SELECT A.customer_id,
            A.first_name,
            A.last_name,
            C.city,
            D.country,
            SUM (E.amount) AS total_amount_paid
    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','Acua','Citrus Heights','Iwaki
    GROUP BY A.customer_id, C.city_id, D.country_id
    ORDER BY total_amount_paid DESC
    LIMIT 5) AS top_5_customers ON A.country = top_5_custome
GROUP BY A.country, top_5_customers
ORDER BY all_customer_count DESC
LIMIT 5;
```

Data Output | Explain | Messages | Notifications

| | country<br>character varying (50) | all_customer_count<br>bigint | top_customer_count<br>bigint |
|---|---|---|---|
| 1 | India | 60 | 1 |
| 2 | China | 53 | 1 |
| 3 | United States | 36 | 1 |
| 4 | Japan | 31 | 1 |
| 5 | Mexico | 30 | 1 |

STEP 3:

You could do step 1 without subqueries, but it would take multiple steps using aggregate functions. You need to use a subquery for step 2 cause you need to join the results of the inner to the outer and the take the average of that.

Subqueries are useful when the results you want are complex and require more than one query and the data is changing.