3.8 data Immersion

Katie Goyal

STEP 1:

Query | Query History

```
1  SELECT AVG(total_spent) AS average_paid_by_top_5_customers FROM
2  (
3      SELECT
4          A.customer_id, B.first_name, B.last_name,
5          E.country, D.city,
6          SUM(A.amount) AS total_spent, B.email
7      FROM payment A
8      INNER JOIN customer B ON A.customer_id = B.customer_id
9      INNER JOIN address C ON B.address_id = C.address_id
10     INNER JOIN city D ON C.city_id = D.city_id
11     INNER JOIN country E ON D.country_id = E.country_id
12     WHERE city IN
13         ('Aurora', 'Atlixco', 'Xintai',
14         'Adoni', 'Dhule (Dhulla)', 'Khurasaki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
15     GROUP BY
16         A.customer_id, A.customer_id, B.first_name, B.last_name,
17         A.amount, B.email, C.address, D.city, E.country
18     ORDER BY total_spent DESC
19     LIMIT 5
20 )
21
```

Data Output | Messages | Notifications

	average_paid_by_top_5_customers numeric
1	34.9300000000000000

Step 2:

Query | Query History

```
SELECT A.country, B.all_customer_count, COUNT(A.country) AS top_5_customer_count
FROM
    (SELECT
        E.country,
        COUNT(DISTINCT B.customer_id) AS all_customer_count
    FROM payment A
    INNER JOIN customer B ON A.customer_id = B.customer_id
    INNER JOIN address C ON B.address_id = C.address_id
    INNER JOIN city D ON C.city_id = D.city_id
    INNER JOIN country E ON D.country_id = E.country_id
    GROUP BY E.country) B
LEFT JOIN
    (SELECT country
     FROM payment A
     INNER JOIN customer B ON A.customer_id = B.customer_id
     INNER JOIN address C ON B.address_id = C.address_id
     INNER JOIN city D ON C.city_id = D.city_id
     INNER JOIN country E ON D.country_id = E.country_id
     WHERE D.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulla)', 'Khurasaki', 'Pingxiang', 'Sivas', 'Celaya', 'Sc
     GROUP BY E.country
     ORDER BY SUM(A.amount) DESC
     LIMIT 5) A ON B.country = A.country
```

Data Output | Messages | Notifications

	country character varying (50)	all_customer_count bigint	top_5_customer_count bigint
1	India	60	1
2	China	53	1
3	United States	36	1
4	[null]	31	0
5	Mexico	30	1
6	[null]	28	0
7	[null]	20	0
8	Turkey	15	1
9	[null]	14	0
10	[null]	13	0
11	[null]	11	0
12	[null]	10	0
13	[null]	9	0
14	[null]	8	0
15	[null]	7	0

Total rows: 21 of 21 | Query complete 00:00:00.066

Step 3:

No I believe it could not have been done without subqueries they are essential for running this code because it helps with complex filtering with in the data. Subqueries can also can also improve the performance of a query which I believe that's what it did here.