SQL Exercise Report

Immersion 3.8: Performing Subqueries

Prepared by Alexandru Cojocari

May 2025

SQL Exercise Report - Alexandru Cojocari

Immersion 3.8: Performing Subqueries

1. Introduction

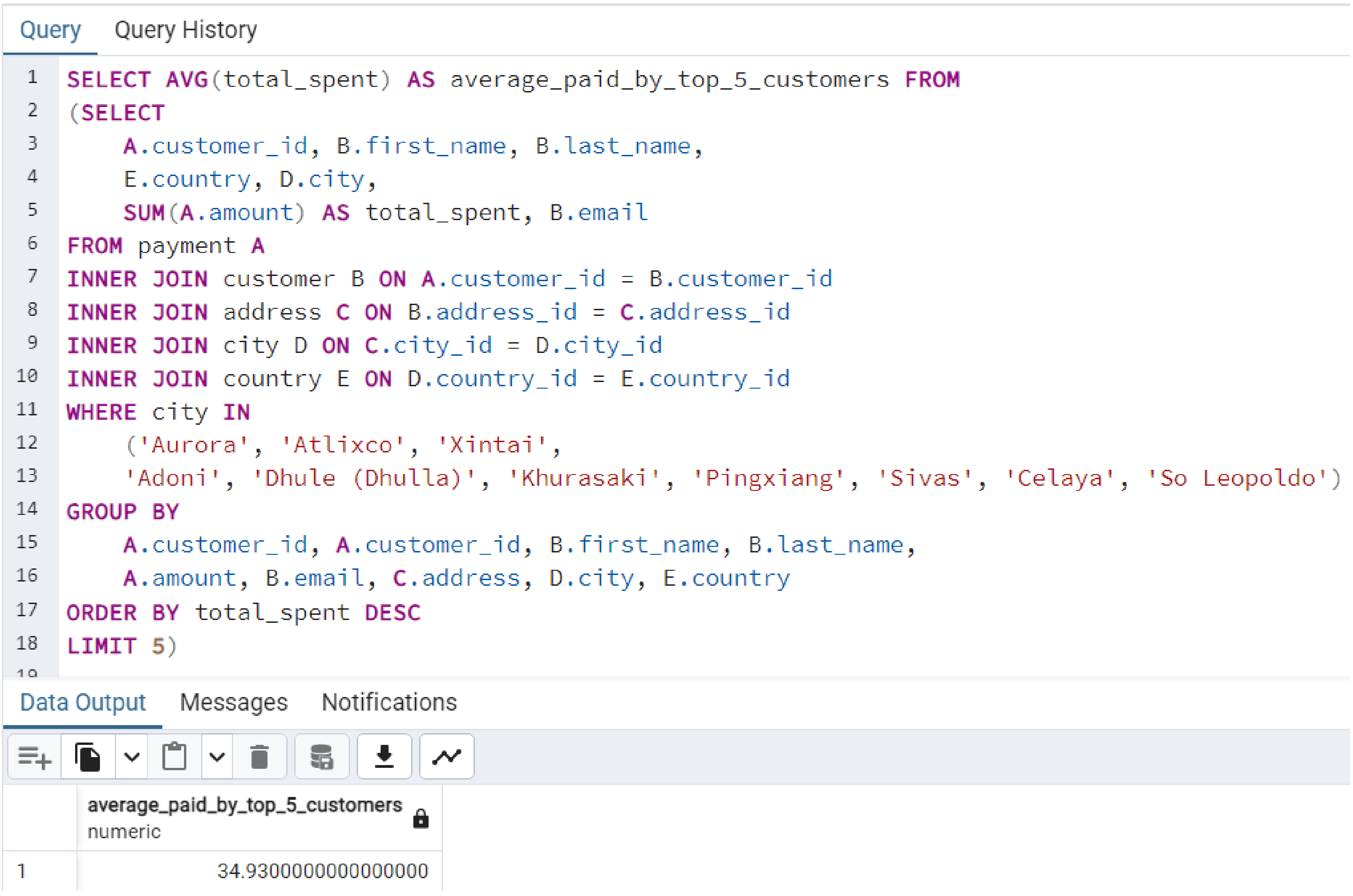
Subqueries are a powerful feature in SQL, allowing a query to be nested within another. They provide structured access to intermediate results and enable complex analytical workflows. This report evaluates top customer spending and country-based comparisons using subqueries.

2. Query Breakdown

Query 1: Average Spending of Top 5 Customers

```
SELECT AVG(total_spent) AS average_paid_by_top_5_customers FROM (
  SELECT
    A.customer_id, B.first_name, B.last_name,
    E.country, D.city,
    SUM(A.amount) AS total_spent, B.email
  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 city IN
    ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulla)', 'Khurasaki',
     'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
  GROUP BY A.customer_id, B.first_name, B.last_name,
           B.email, D.city, E.country
  ORDER BY total_spent DESC
  LIMIT 5
)
```

Figure 1: SQL for calculating average spending of top 5 customers



```
SELECT AVG(total_spent) AS average_paid_by_top_5_customers FROM
(SELECT
    A.customer_id, B.first_name, B.last_name,
    E.country, D.city,
    SUM(A.amount) AS total_spent, B.email
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 city IN
    ('Aurora', 'Atlixco', 'Xintai',
    'Adoni', 'Dhule (Dhulla)', 'Khurasaki', 'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
GROUP BY
    A.customer_id, A.customer_id, B.first_name, B.last_name,
    A.amount, B.email, C.address, D.city, E.country
ORDER BY total_spent DESC
LIMIT 5)
```

	average_paid_by_top_5_customers numeric
1	34.9300000000000000

Query 2: Top 5 Customer Count by Country

```
SELECT A.country, B.all_customer_count, COUNT(A.country) AS top_5_customer_count
FROM (
  SELECT
    A.customer_id, B.first_name, B.last_name,
    E.country, D.city,
    SUM(A.amount) AS total_spent, B.email
  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 city IN
    ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulla)', 'Khurasaki',
     'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
  GROUP BY A.customer_id, B.first_name, B.last_name, B.email, D.city, E.country
  ORDER BY total_spent DESC
  LIMIT 5
) A
LEFT JOIN (
  SELECT country, COUNT(country) 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 country
```

- Diverse city selection supports regional comparison
- Top customers provide key revenue insights
- Useful for targeted marketing strategy

4. Reflections & Best Practices

Subqueries are flexible but can become complex when deeply nested. Views or temporary tables might be clearer. While subqueries help encapsulate logic, they can also reduce readability and increase maintenance costs in larger systems.

Updated SQL Queries

Query 1: Average Spending of Top 5 Customers

```
SELECT AVG(total_spent) AS average_paid_by_top_5_customers
FROM (
  SELECT
    A.customer_id,
    B.first_name,
    B.last_name,
    E.country,
    D.city,
    SUM(A.amount) AS total_spent,
    B.email
  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 city IN ('Aurora', 'Linares', 'Gdynia', 'So Leopoldo', 'Sivas',
                 'Xintai', 'Adoni', 'Celaya', 'Tebessa', 'Tianjin',
                 'Changzhi', 'Dhule (Dhulla)', 'Pingxiang', 'Khursak')
  GROUP BY A.customer_id, B.first_name, B.last_name, B.email, D.city, E.country
  ORDER BY total_spent DESC
  LIMIT 5
) AS top_customers;
```

Query 2: Top 5 Customer Count by Country

```
-- Step 1
SELECT *
FROM (
  SELECT
    A.customer_id,
    B.first_name,
    B.last_name,
    D.city,
    E.country,
    SUM(A.amount) AS total_payment
  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', 'Linares', 'Gdynia', 'So Leopoldo', 'Sivas',
                   'Xintai', 'Adoni', 'Celaya', 'Tebessa', 'Tianjin',
                   'Changzhi', 'Dhule (Dhulla)', 'Pingxiang', 'Khursak')
  GROUP BY A.customer_id, B.first_name, B.last_name, B.email, D.city, E.country
  ORDER BY total_payment DESC
  LIMIT 5
) AS top_customers

-- Step 2
LEFT JOIN (
  SELECT
    E.country,
    COUNT(*) 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
) AS all_customers
```