## Performing subqueries

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

```
(SELECT
    customer.customer_id,
      customer.first_name,
      customer.last_name,
      city.city,
      country.country,
      SUM(payment.amount) AS total_amount_paid
FROM country

INNER JOIN city ON country.country_id = city.country_id
INNER JOIN address ON city.city_id = address.city_id
INNER JOIN customer ON address.address_id = customer.address_id
INNER JOIN payment ON customer.customer_id = payment.customer_id

WHERE city.city IN
('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
AND country.country IN
('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil', 'Russian
Federation', 'Philippines', 'Turkey', 'Indonesia')

GROUP BY customer.customer_id, customer.first_name,
customer.last_name, city.city, country.country
ORDER BY total_amount_paid DESC
LIMIT 5;
) AS total_amount_paid
```

An outer statement to calculate the average amount paid with an alias:

```
SELECT AVG(total_amount_paid.total_amount_paid) AS average
FROM (
  SELECT
      customer.customer_id,
      customer.first_name,
      customer.last_name,
      city.city,
      country.country,
      SUM(payment.amount) AS total_amount_paid
  FROM country
  INNER JOIN city ON country.country_id = city.country_id
  INNER JOIN address ON city.city_id = address.city_id
  INNER JOIN customer ON address.address_id = customer.address_id
  INNER JOIN payment ON customer.customer_id =
payment.customer_id
  WHERE city.city IN
    ('Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
     'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo')
   AND country.country IN
    ('India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
     'Russian Federation', 'Philippines', 'Turkey', 'Indonesia')
  GROUP BY customer.customer_id, customer.first_name,
customer.last_name, city.city, country.country
  ORDER BY total_amount_paid DESC
  LIMIT 5
) AS total_amount_paid;
```

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

```
SELECT
  country.country,
  COUNT(DISTINCT customer.customer_id) AS all_customer_count,
  COUNT(DISTINCT top_5_customers.customer_id) AS
top_customer_count
FROM country
INNER JOIN city ON country.country_id = city.country_id
INNER JOIN address ON city.city_id = address.city_id
INNER JOIN customer ON customer.address_id = address.address_id

LEFT JOIN (
  -- paste the subquery here
  SELECT
    customer.customer_id,
    country.country
  FROM country
  INNER JOIN city ON country.country_id = city.country_id
  INNER JOIN address ON city.city_id = address.city_id
  INNER JOIN customer ON customer.address_id = address.address_id
  INNER JOIN payment ON customer.customer_id =
payment.customer_id
  WHERE city.city IN (
    'Aurora', 'Atlixco', 'Xintai', 'Adoni', 'Dhule (Dhulia)', 'Kurashiki',
    'Pingxiang', 'Sivas', 'Celaya', 'So Leopoldo'
  )
  AND country.country IN (
    'India', 'China', 'United States', 'Japan', 'Mexico', 'Brazil',
    'Russian Federation', 'Philippines', 'Turkey', 'Indonesia'
  )
  GROUP BY customer.customer_id, country.country
```

```
    ORDER BY SUM(payment.amount) DESC
    LIMIT 5
  ) AS top_5_customers
  ON top_5_customers.customer_id = customer.customer_id

  GROUP BY country.country
  ORDER BY all_customer_count DESC;
```

**Output:**

| "country" | "all_customer_count" | "top_customer_count" |
|---|---|---|
| "India" | 60 | 1 |
| "China" | 53 | 0 |
| "United States" | 36 | 1 |
| "Japan" | 31 | 0 |
| "Mexico" | 30 | 2 |
| "Brazil" | 28 | 0 |
| "Russian Federation" | 28 | 0 |
| "Philippines" | 20 | 0 |
| "Turkey" | 15 | 1 |
| "Indonesia" | 14 | 0 |
| "Nigeria" | 13 | 0 |
| "Argentina" | 13 | 0 |
| "South Africa" | 11 | 0 |
| "Taiwan" | 10 | 0 |
| "United Kingdom" | 9 | 0 |
| "Iran" | 8 | 0 |
| "Poland" | 8 | 0 |
| "Italy" | 7 | 0 |
| "Germany" | 7 | 0 |
| "Venezuela" | 7 | 0 |
| "Egypt" | 6 | 0 |
| "Ukraine" | 6 | 0 |
| "Vietnam" | 6 | 0 |
| "Colombia" | 6 | 0 |
| "Spain" | 5 | 0 |
| "Canada" | 5 | 0 |
| "Saudi Arabia" | 5 | 0 |
| "Netherlands" | 5 | 0 |
| "Pakistan" | 5 | 0 |
| "South Korea" | 5 | 0 |
| "Peru" | 4 | 0 |
| "France" | 4 | 0 |
| "Yemen" | 4 | 0 |
| "Israel" | 4 | 0 |
| "Algeria" | 3 | 0 |

| | | |
|---|---|---|
| "Switzerland" | 3 | 0 |
| "Tanzania" | 3 | 0 |
| "United Arab Emirates" | 3 | 0 |
| "Morocco" | 3 | 0 |
| "Bangladesh" | 3 | 0 |
| "Chile" | 3 | 0 |
| "Thailand" | 3 | 0 |
| "Malaysia" | 3 | 0 |
| "Austria" | 3 | 0 |
| "Paraguay" | 3 | 0 |
| "Mozambique" | 3 | 0 |
| "Ecuador" | 3 | 0 |
| "Dominican Republic" | 3 | 0 |
| "Sudan" | 2 | 0 |
| "Bolivia" | 2 | 0 |
| "Greece" | 2 | 0 |
| "Belarus" | 2 | 0 |
| "Bulgaria" | 2 | 0 |
| "Yugoslavia" | 2 | 0 |
| "Cambodia" | 2 | 0 |
| "Cameroon" | 2 | 0 |
| "Romania" | 2 | 0 |
| "Puerto Rico" | 2 | 0 |
| "Kazakstan" | 2 | 0 |
| "Kenya" | 2 | 0 |
| "Angola" | 2 | 0 |
| "Latvia" | 2 | 0 |
| "Azerbaijan" | 2 | 0 |
| "Congo, The Democratic Republic of the" | 2 | 0 |
| "Oman" | 2 | 0 |
| "Myanmar" | 2 | 0 |
| "French Polynesia" | 2 | 0 |
| "Zambia" | 1 | 0 |
| "American Samoa" | 1 | 0 |
| "Anguilla" | 1 | 0 |
| "Armenia" | 1 | 0 |
| "Bahrain" | 1 | 0 |
| "Brunei" | 1 | 0 |
| "Chad" | 1 | 0 |
| "Czech Republic" | 1 | 0 |
| "Estonia" | 1 | 0 |
| "Ethiopia" | 1 | 0 |
| "Faroe Islands" | 1 | 0 |
| "Finland" | 1 | 0 |
| "French Guiana" | 1 | 0 |

| | | |
|---|---|---|
| "Gambia" | 1 | 0 |
| "Greenland" | 1 | 0 |
| "Holy See (Vatican City State)" | 1 | 0 |
| "Hong Kong" | 1 | 0 |
| "Hungary" | 1 | 0 |
| "Iraq" | 1 | 0 |
| "Kuwait" | 1 | 0 |
| "Liechtenstein" | 1 | 0 |
| "Lithuania" | 1 | 0 |
| "Madagascar" | 1 | 0 |
| "Malawi" | 1 | 0 |
| "Moldova" | 1 | 0 |
| "Nauru" | 1 | 0 |
| "Nepal" | 1 | 0 |
| "New Zealand" | 1 | 0 |
| "North Korea" | 1 | 0 |
| "Runion" | 1 | 0 |
| "Saint Vincent and the Grenadines" | 1 | 0 |
| "Senegal" | 1 | 0 |
| "Slovakia" | 1 | 0 |
| "Sri Lanka" | 1 | 0 |
| "Sweden" | 1 | 0 |
| "Tonga" | 1 | 0 |
| "Tunisia" | 1 | 0 |
| "Turkmenistan" | 1 | 0 |
| "Tuvalu" | 1 | 0 |
| "Virgin Islands, U.S." | 1 | 0 |
| "Afghanistan" | 1 | 0 |

**Could steps 1 and 2 be done without using subqueries?**
Technically, yes—steps 1 and 2 could be done without using subqueries by writing more complex joins and filtering logic directly into a single query. However, that would make the query harder to read and maintain. Subqueries allow you to break a complex problem into logical steps. For example, isolating the top 5 customers first in a subquery makes it easier to join that smaller set of data with other tables in the next step.

**When are subqueries useful?**
Subqueries are especially useful when you need to perform intermediate calculations, filter down to a specific group of data, or compare results against aggregated values. They help keep the logic modular and readable—like creating a temporary dataset that you can

join or filter on later. This is particularly valuable in real-world scenarios where datasets are large and queries need to be optimized for clarity and performance.