SubQuery

```sql
USE MyDatabase;


-- Show customers whose score is higher than the average score.
SELECT
      first_name,
      score
FROM customers
WHERE score > (SELECT AVG(score) FROM customers)
ORDER BY score DESC;




-- Find the customer(s) with the highest score.
SELECT
      id,
      first_name,
      country,
      score
FROM customers
WHERE score = (SELECT MAX(score) FROM customers);


-- Find customers whose score is lower than the average score.
SELECT
      id,
      first_name,
      country,
      score
FROM customers
WHERE score < (SELECT AVG(score) FROM customers)
ORDER BY score DESC;


-- Show customers who live in the same country as "Ahmed".
SELECT
      first_name,
      country
FROM Customers
WHERE country = (
    SELECT country
    FROM Customers
    WHERE first_name = 'Ahmed'
);



-- Find customers who have the minimum score.
SELECT
      id,
      first_name,
      country,
      score
FROM customers
```

```sql
WHERE score = (SELECT MIN(score) FROM customers);


-- Show customers whose score is higher than the average score of customers from
Egypt.
SELECT
       id,
       first_name,
       country,
       score
FROM customers
WHERE score > (
       SELECT AVG(score)
       FROM customers
       WHERE country LIKE 'egypt'
);


-- Find customers who are NOT from the same country as "Sara".
SELECT
       first_name,
       country
FROM customers
WHERE country <> (
       SELECT country
       FROM customers
       WHERE first_name = 'Sara'
);

-- Show customers whose score is greater than the lowest score.
SELECT
       id,
       first_name,
       country,
       score
FROM customers
WHERE score > (
       SELECT
       MIN(score)
       FROM customers)
ORDER BY score DESC;


-- Show customers whose score is equal to the average score.
SELECT
       id,
       first_name,
       country,
       score
FROM customers
WHERE score = (
       SELECT
       AVG(score)
       FROM customers)
```

```sql
ORDER BY score DESC;



-- Find customers whose score is higher than the minimum score.
SELECT
       id,
       first_name,
       country,
       score
FROM customers
WHERE score > (
       SELECT
       MIN(score)
       FROM customers)
ORDER BY score DESC;



-- Show customers whose score is lower than the highest score.
SELECT
       id,
       first_name,
       country,
       score
FROM customers
WHERE score < (
       SELECT
       MAX(score)
       FROM customers)
ORDER BY score DESC;



/*
       ===============================================================
       == This is called nested subquery (subquery inside subquery) ==
       ===============================================================
*/
-- Find customers who live in the same country as the customer with the highest
score.
SELECT
       id,
       first_name,
       country,
       score
FROM Customers
WHERE country = (
    SELECT country
    FROM Customers
    WHERE score = (
        SELECT MAX(score)
        FROM Customers
    ))
ORDER BY score DESC;
```

```sql
-- Show customers whose score is higher than the average score of customers from
USA.
SELECT first_name, score
FROM Customers
WHERE score > (
    SELECT AVG(score)
    FROM Customers
    WHERE country = 'USA'
);



-- Find customers who have the same score as "Ahmed".
SELECT first_name, score
FROM Customers
WHERE score = (
    SELECT score
    FROM Customers
    WHERE first_name = 'Ahmed'
);



-- Show customers whose score is NOT equal to the average score.
SELECT first_name, score
FROM Customers
WHERE score <> (
    SELECT AVG(score)
    FROM Customers
);



-- Find customers whose score is higher than the score of "Sara".
SELECT first_name, score
FROM Customers
WHERE score > (
    SELECT score
    FROM Customers
    WHERE first_name = 'Sara'
);


-- Show customers whose country is the same as the customer with the lowest score.
SELECT first_name, country
FROM Customers
WHERE country = (
    SELECT country
    FROM Customers
    WHERE score = (
        SELECT MIN(score)
        FROM Customers
    )
);



-- Find customers whose score is between the minimum and maximum score.
```

```sql
SELECT first_name, score
FROM Customers
WHERE score BETWEEN (
    SELECT MIN(score) FROM Customers
) AND (
    SELECT MAX(score) FROM Customers
);



-- Find customers who live in the same country as any customer with a score above
90.
SELECT first_name, country, score
FROM Customers
WHERE country IN (
    SELECT country
    FROM Customers
    WHERE score > 90
);



-- Show customers whose country is not the same as any customer with score < 50.
SELECT first_name, country, score
FROM Customers
WHERE country NOT IN (
    SELECT country
    FROM Customers
    WHERE score < 50
);



-- Find customers whose score is equal to any score in the "USA".
SELECT first_name, country, score
FROM Customers
WHERE score = ANY (
    SELECT score
    FROM Customers
    WHERE country = 'USA'
);



-- Show customers whose score is higher than all scores in the "Egypt" department.
SELECT first_name, country, score
FROM Customers
WHERE score > ALL (
    SELECT score
    FROM Customers
    WHERE country = 'Egypt'
);



-- Find customers who live in the same countries as "Ahmed" or "Sara".
SELECT first_name, country
FROM Customers
```

```sql
WHERE country IN (
    SELECT country
    FROM Customers
    WHERE first_name IN ('Ahmed', 'Sara')
);



-- Show customers who do not live in the countries of the top 3 highest scoring
customers.
SELECT first_name, country, score
FROM Customers
WHERE country NOT IN (
    SELECT country
    FROM Customers
    ORDER BY score DESC
    LIMIT 3
);



-- Find customers whose score matches any other customer.
SELECT first_name, score
FROM Customers
WHERE score IN (
    SELECT score
    FROM Customers
    GROUP BY score
    HAVING COUNT(*) > 1
);



/*

      ----------------------------------------------
      -- Correlated Subquery - Simple Explanation --
      ----------------------------------------------
      - A correlated subquery is a subquery that depends on the outer query.
      - It runs once for each row of the outer query.
      - Unlike normal subqueries, it references columns from the outer query.
*/



-- Find customers whose score is higher than the average score of their country.
SELECT c1.first_name, c1.country, c1.score
FROM Customers c1
WHERE c1.score > (
    SELECT AVG(c2.score)
    FROM Customers c2
    WHERE c2.country = c1.country
);



-- Show customers whose score is the highest in their country.
SELECT c1.first_name, c1.country, c1.score
```

```sql
FROM Customers c1
WHERE c1.score = (
    SELECT MAX(c2.score)
    FROM Customers c2
    WHERE c2.country = c1.country
);



-- Find customers whose score is below the average score of customers from the same
country.
SELECT c1.first_name, c1.country, c1.score
FROM Customers c1
WHERE c1.score < (
    SELECT AVG(c2.score)
    FROM Customers c2
    WHERE c2.country = c1.country
);



-- Find customers whose score is less than the maximum score of their country.
SELECT c1.first_name, c1.country, c1.score
FROM Customers c1
WHERE c1.score < (
    SELECT MAX(c2.score)
    FROM Customers c2
    WHERE c2.country = c1.country
);



-- Find customers whose score is higher than the average score of customers with
lower scores in the same country.
SELECT c1.first_name, c1.country, c1.score
FROM Customers c1
WHERE c1.score > (
    SELECT AVG(c2.score)
    FROM Customers c2
    WHERE c2.country = c1.country AND c2.score < c1.score
);



-- Show customers who live in a country where all other customers have lower scores.
SELECT c1.first_name, c1.country, c1.score
FROM Customers c1
WHERE c1.score >= ALL (
    SELECT c2.score
    FROM Customers c2
    WHERE c2.country = c1.country
);



-- Find customers whose score is below the average score of their country, but above
the average of the entire table.
```

```sql
SELECT c1.first_name, c1.country, c1.score
FROM Customers c1
WHERE c1.score < (
    SELECT AVG(c2.score)
    FROM Customers c2
    WHERE c2.country = c1.country
)
AND c1.score > (
    SELECT AVG(score)
    FROM Customers
);
```