Here's how you can achieve that:

**1. Subquery to Find Customers with Above Average Order Value:**

SQL

SELECT customer\_id

FROM orders

GROUP BY customer\_id

HAVING AVG(total\_amount) > (

SELECT AVG(total\_amount)

FROM orders

);

**Explanation:**

* This query finds customer IDs where the average order value for each customer is greater than the overall average order value.
* The inner subquery calculates the overall average order value using AVG(total\_amount) FROM orders.
* The outer query groups orders by customer\_id and calculates the average order value for each customer using AVG(total\_amount).
* The HAVING clause then filters the results to include only customer IDs where the individual average is greater than the overall average.

**2. UNION Query to Combine Results:**

Here's how you can combine the above subquery with another SELECT statement (assuming you have another table named high\_value\_customers containing specific customer information) into a single result set:

SQL

SELECT customer\_id

FROM orders

GROUP BY customer\_id

HAVING AVG(total\_amount) > (

SELECT AVG(total\_amount)

FROM orders

)

UNION

SELECT customer\_id

FROM high\_value\_customers;

**Explanation:**

* This UNION query combines the results of two SELECT statements that have the same number of columns (in this case, just customer\_id).
* The first part remains the same, finding customers with above average order values.
* The UNION keyword combines the results of this subquery with the second SELECT statement that retrieves customer IDs directly from the high\_value\_customers table.
* This way, the final result set will include both customers with high average order values and any additional customers identified in the high\_value\_customers table.