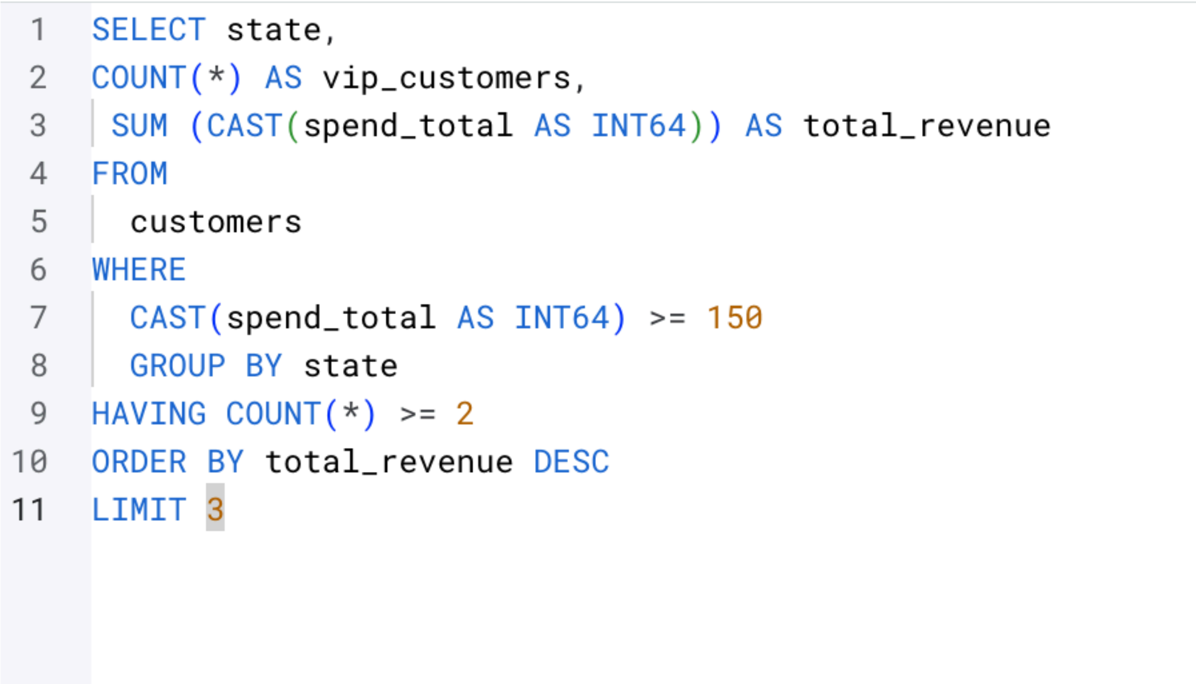
Scenario:
A manager asks:
“We want to identify the top 3 states with the highest revenue from VIP customers, but also show the average spend of VIPs in those states.”
Definitions:

- VIP customer = `spend_total ≥ 150`
- Return:
  - state
  - vip_customers (count)
  - total_revenue (sum of spend)
  - `avg_vip_spend` (average spend per VIP)
- Only include states with at least 2 VIPs
- Sort by total_revenue descending
- Limit to **top 3 states**

Result:
ORDER BY + LIMIT

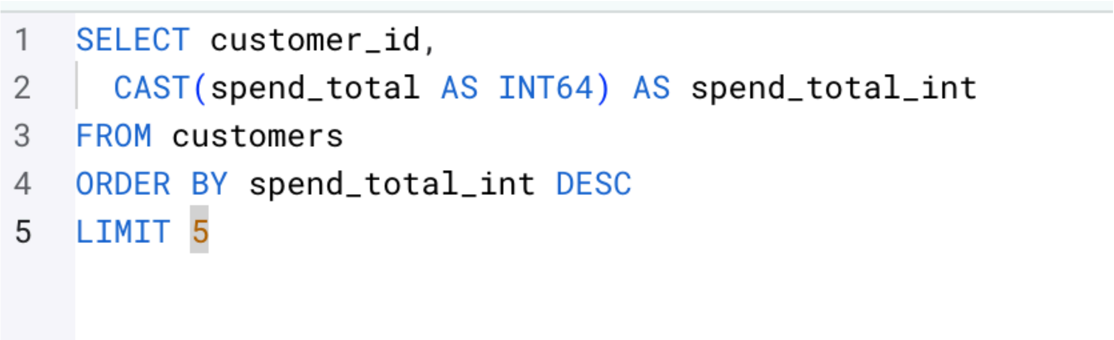


```
SELECT state,
COUNT(*) AS vip_customers,
 SUM (CAST(spend_total AS INT64)) AS total_revenue
FROM
  customers
WHERE
  CAST(spend_total AS INT64) >= 150
  GROUP BY state
HAVING COUNT(*) >= 2
ORDER BY total_revenue DESC
LIMIT 3
```

Which **top 5 customers** have spent the most overall?
Return:

- customer_id
- spend_total

Sorted correctly.
Result



```
SELECT customer_id,
  CAST(spend_total AS INT64) AS spend_total_int
FROM customers
ORDER BY spend_total_int DESC
LIMIT 5
```

Interview-Style Question

You have two tables:

customers

- customer_id
- state
- signup_date

orders

- order_id
- customer_id
- order_total

Goal

Find **states** that:

- have at least 5 customers, and
- whose customers made at least 10 total orders combined

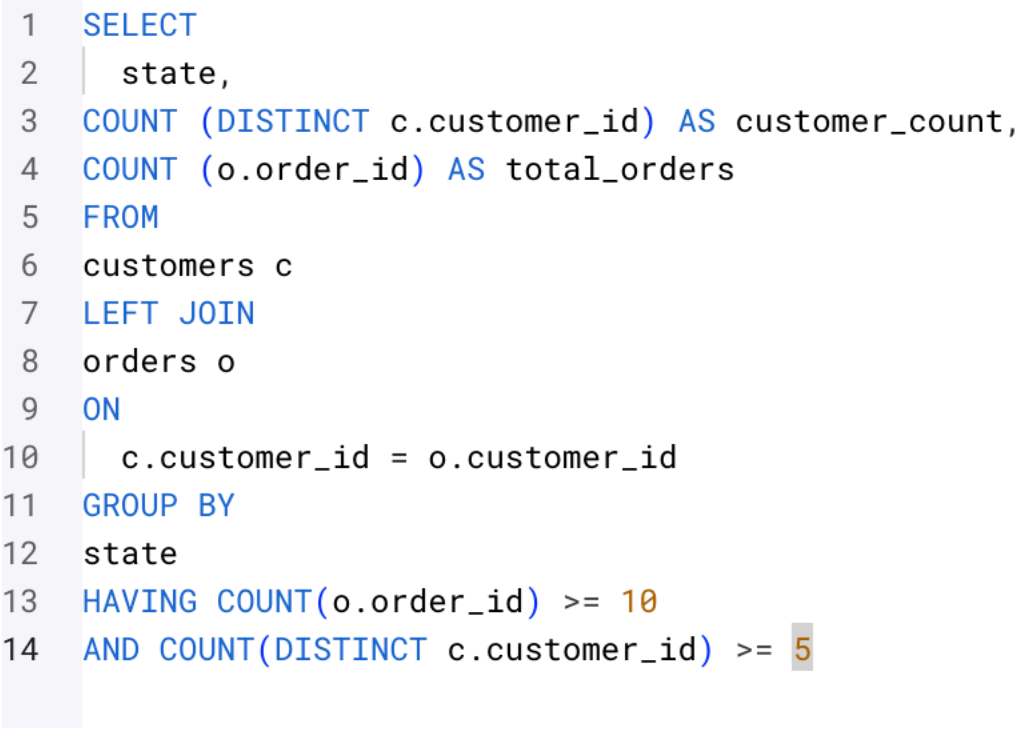
Return:

- state
- customer_count
- total_orders

Rules (like a real interview)

- You must use a JOIN
- You must use GROUP BY
- You must use HAVING
- Customers with zero orders should still count toward customer_count

Result



```
SELECT
    state,
COUNT (DISTINCT c.customer_id) AS customer_count,
COUNT (o.order_id) AS total_orders
FROM
customers c
LEFT JOIN
orders o
ON
    c.customer_id = o.customer_id
GROUP BY
state
HAVING COUNT(o.order_id) >= 10
AND COUNT(DISTINCT c.customer_id) >= 5
```