Database Systems CS-220

Saniya Ashraf

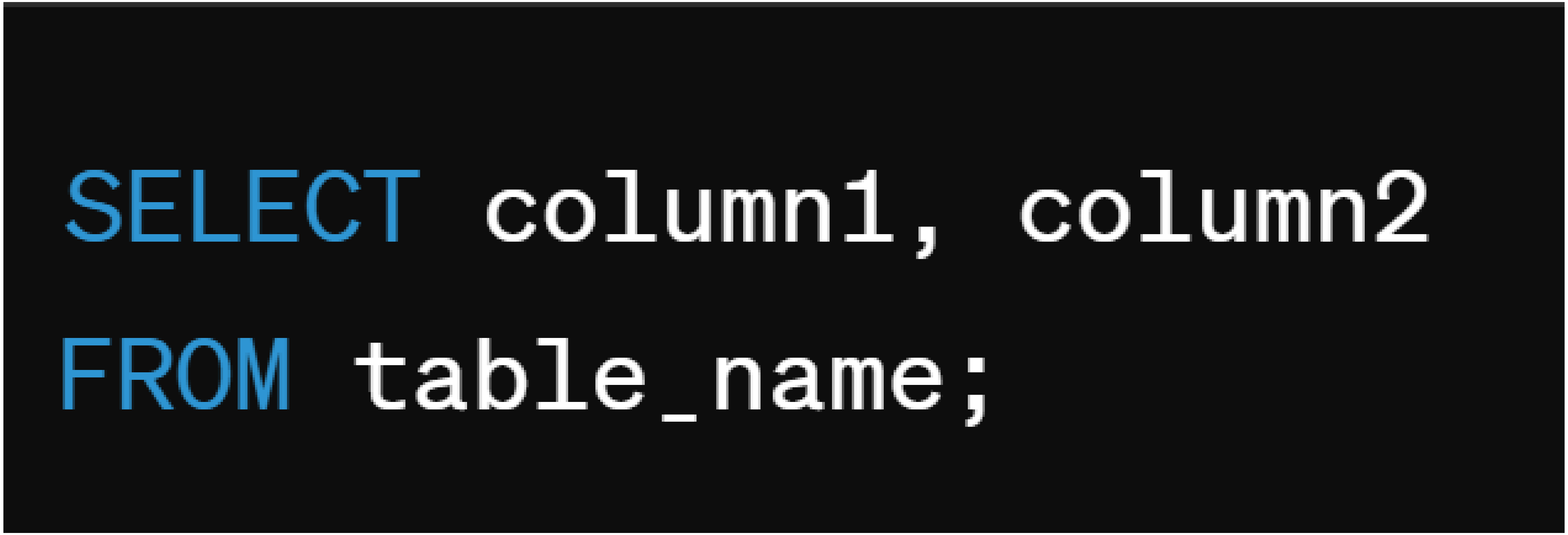
Lecture 5

Retrieving data with SELECT and aggregation

- We retrieve data using the SELECT command and Filter it using the WHERE command
- Retrieving data with SELECT and aggregation involves fetching information from a database and summarizing it using functions like COUNT, SUM, AVG, MAX, or MIN. These functions help condense large datasets into meaningful insights

SELECT

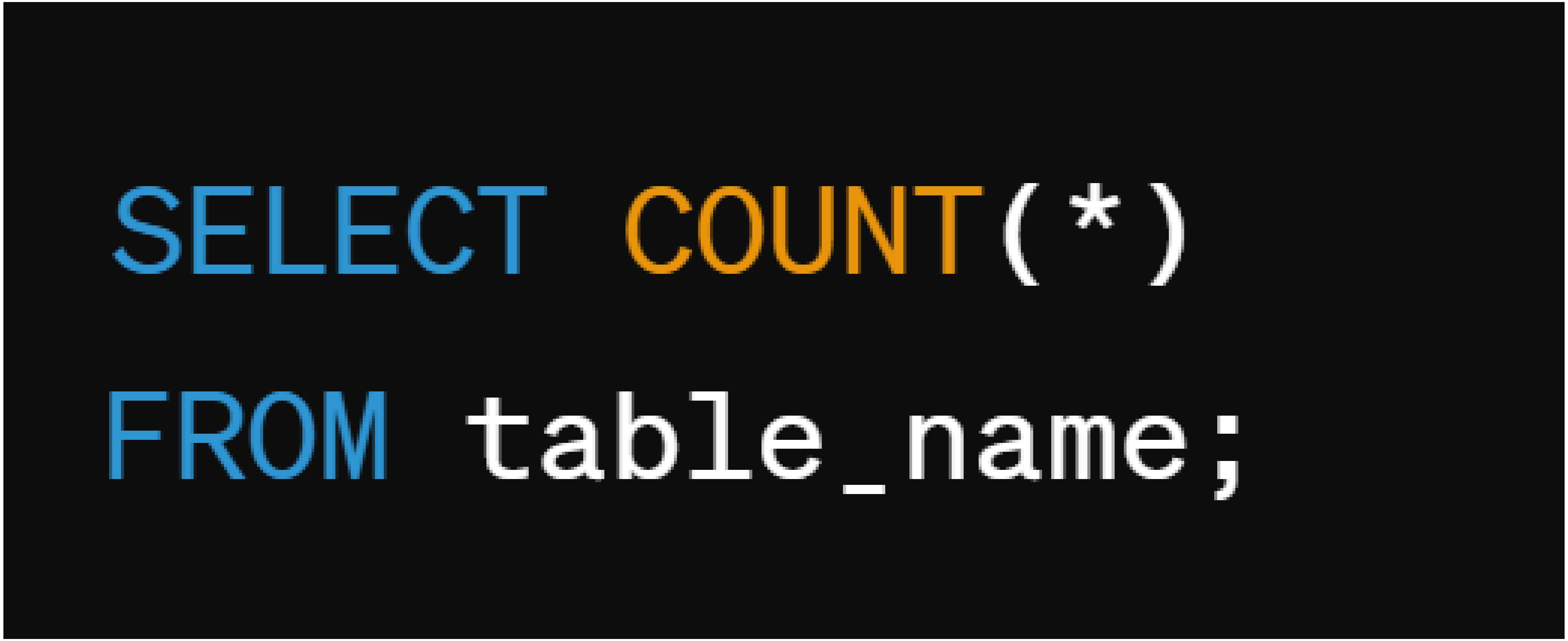
- **SELECT statement**: This is the primary SQL command used to retrieve data from a database table. It specifies which columns you want to include in the result set. You can select specific columns or use the wildcard (*) to select all columns.
- Example:



```sql
SELECT column1, column2
FROM table_name;
```

Aggregation functions

- These functions operate on a set of values and return a single value. They are often used with the SELECT statement to summarize data.
- **COUNT**: Counts the number of rows in a result set.
- **SUM**: Calculates the sum of values in a column.
- **AVG**: Computes the average of values in a column.
- **MAX**: Finds the maximum value in a column.
- **MIN**: Finds the minimum value in a column.



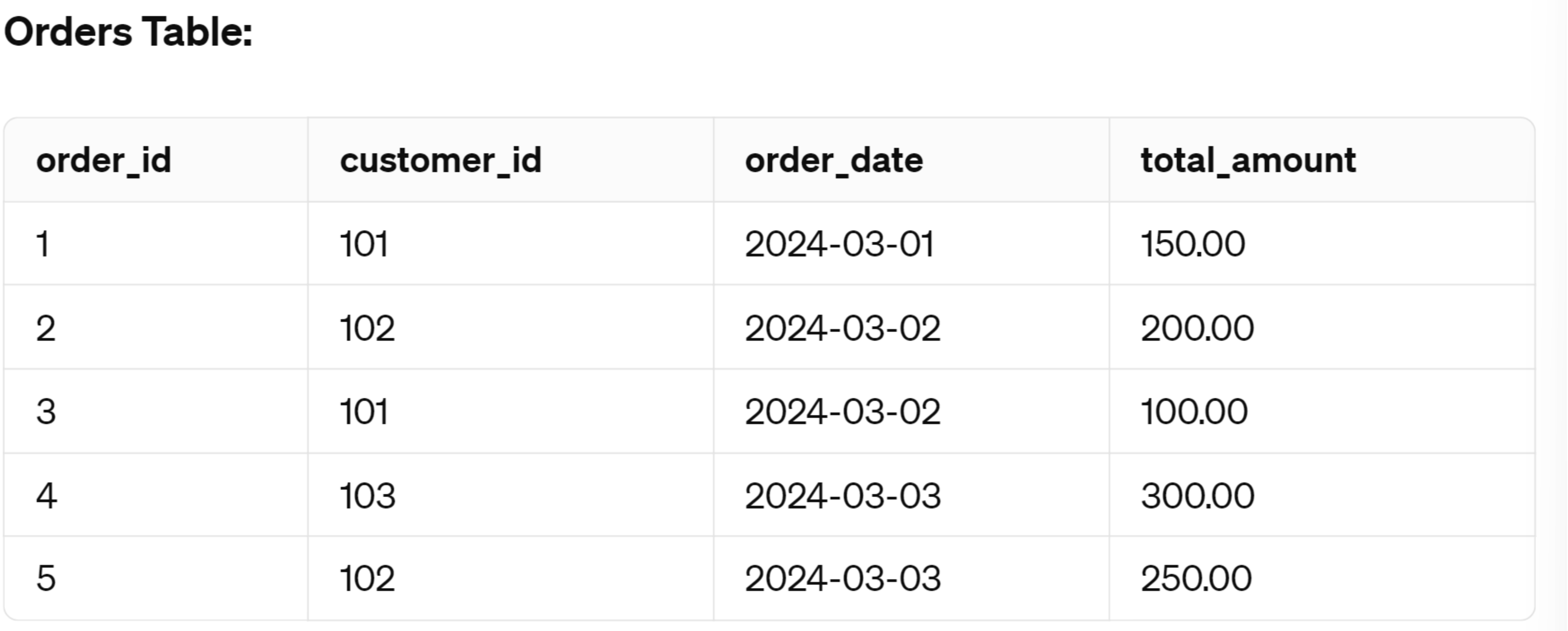
```sql
SELECT COUNT(*)
FROM table_name;
```

GROUP BY clause

- When using aggregation functions, you often want to group the results by one or more columns.
- The GROUP BY clause divides the rows into groups based on the specified column(s), and the aggregation functions are applied to each group separately.
- **Grouping Columns:** You specify one or more columns in the **GROUP BY** clause.
- These columns determine the groups into which the rows are divided.
- Rows with the same values in these columns are grouped together.

GROUP BY example’

- Consider the table

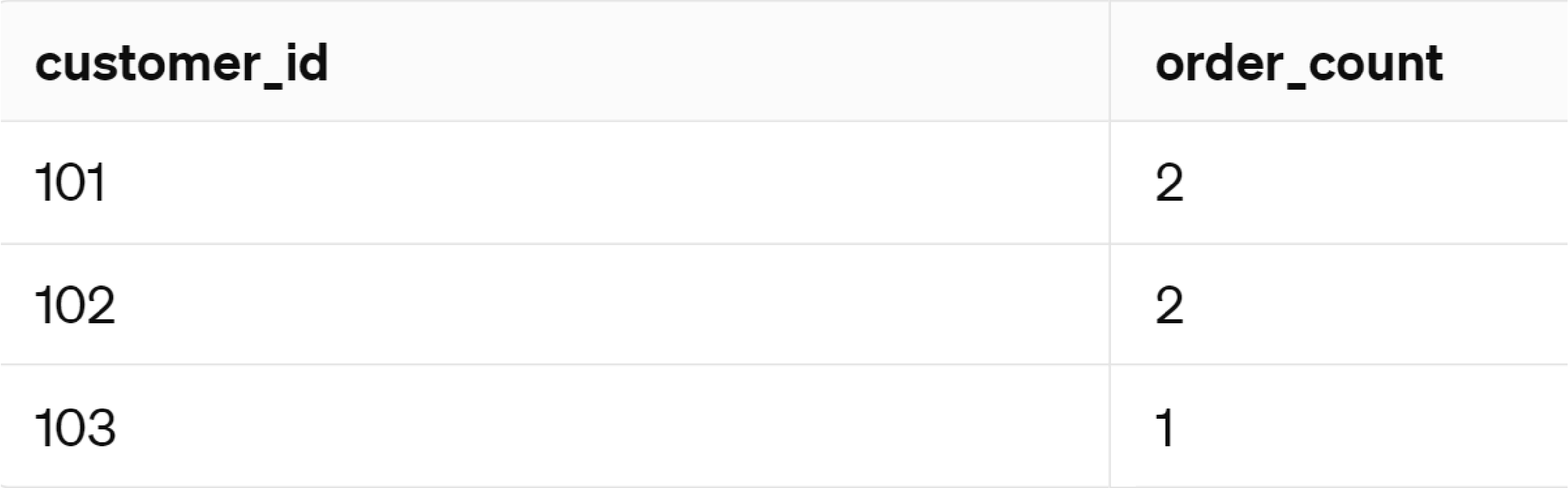


Orders Table:

order_id	customer_id	order_date	total_amount
1	101	2024-03-01	150.00
2	102	2024-03-02	200.00
3	101	2024-03-02	100.00
4	103	2024-03-03	300.00
5	102	2024-03-03	250.00

Grouping by a Single Column

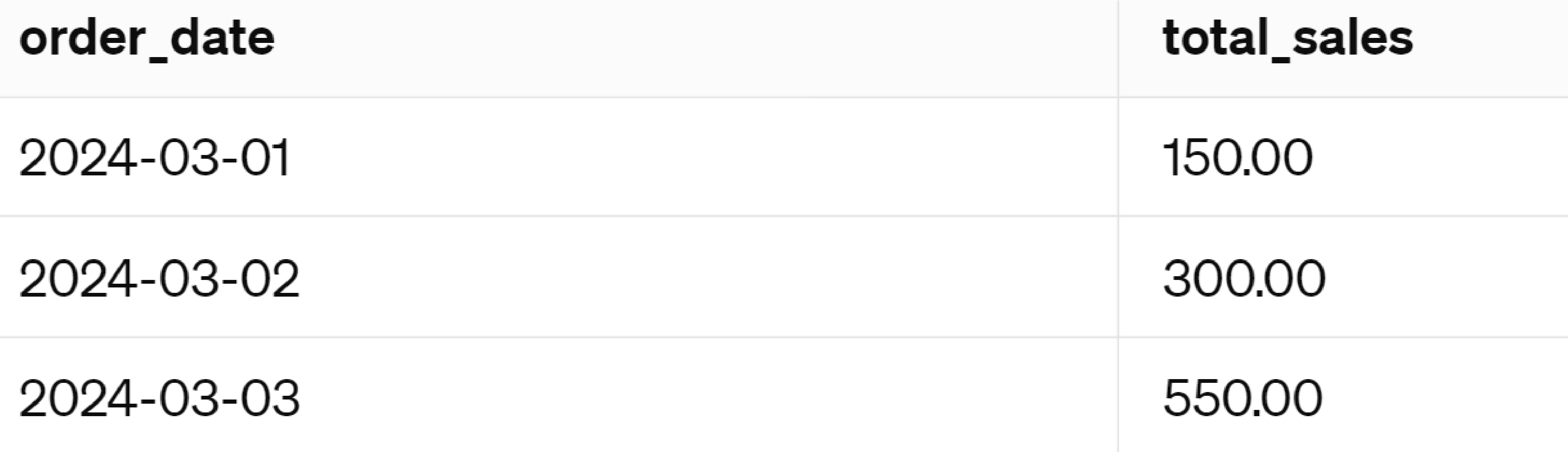
- SELECT customer_id, COUNT(*) AS order_count
- FROM orders
- GROUP BY customer_id;
- **Result:**



customer_id	order_count
101	2
102	2
103	1

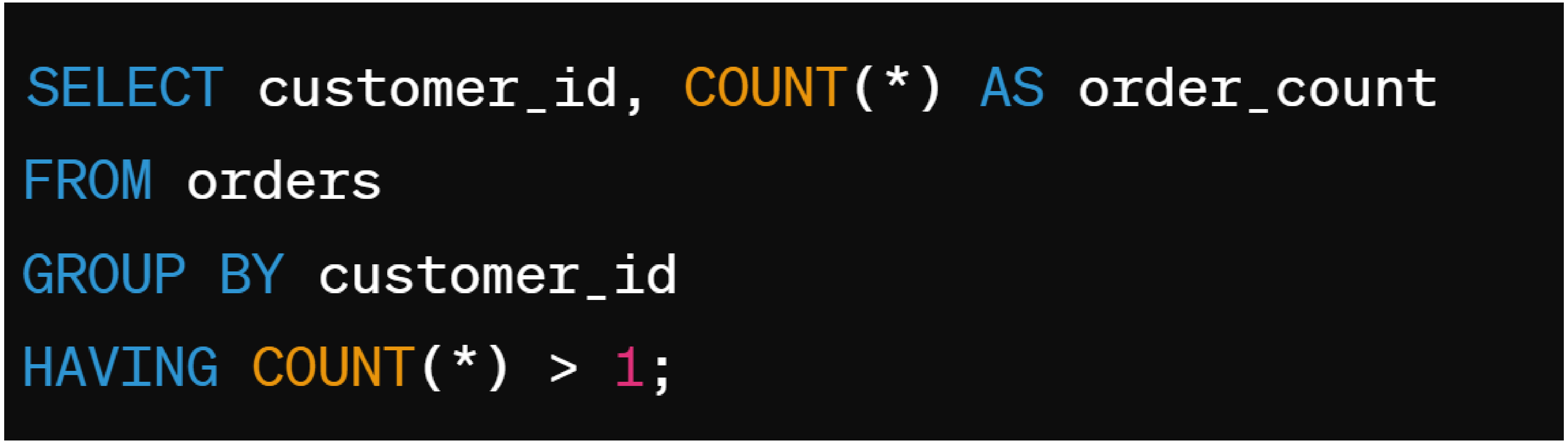
Grouping by Multiple Columns

- SELECT order_date, SUM(total_amount) AS total_sales
- FROM orders
- GROUP BY order_date;



order_date	total_sales
2024-03-01	150.00
2024-03-02	300.00
2024-03-03	550.00

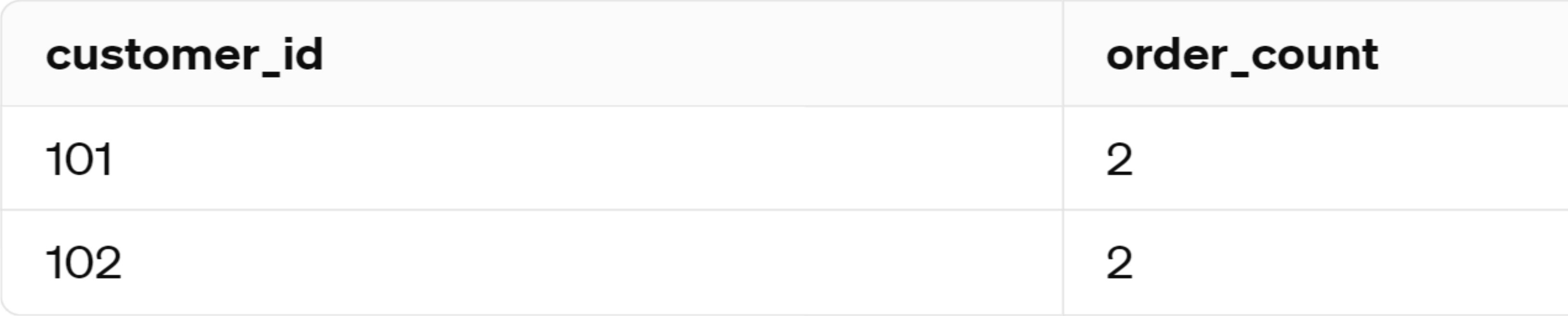
Filtering Groups with HAVING



```sql
SELECT customer_id, COUNT(*) AS order_count
FROM orders
GROUP BY customer_id
HAVING COUNT(*) > 1;
```

This query groups the orders by customer_id and selects only those customers who have placed more than one order.

Result:



customer_id	order_count
101	2
102	2