DAY 9 – SQL GROUP BY, HAVING & AGGREGATE FUNCTIONS

Topics Covered

• GROUP BY
• HAVING
• Aggregate Functions: SUM, COUNT, AVG, MIN, MAX

Aggregate Functions

SUM() – Calculates total value
COUNT() – Counts number of rows
AVG() – Finds average value
MIN() – Finds minimum value
MAX() – Finds maximum value

GROUP BY

GROUP BY groups rows that have the same values in specified columns.

HAVING

HAVING is used to filter aggregated data. WHERE cannot be used with aggregate functions.

Sample Table: sales

Columns:
order_id, customer_name, region, product, quantity, price, order_date

20 Practice SQL Queries

1. SELECT COUNT(*) FROM sales;

2. SELECT SUM(quantity) FROM sales;

3. SELECT AVG(price) FROM sales;

4. SELECT SUM(quantity * price) FROM sales;

5. SELECT region, COUNT(*) FROM sales GROUP BY region;

6. SELECT region, SUM(quantity * price) FROM sales GROUP BY region;

7. SELECT product, SUM(quantity) FROM sales GROUP BY product;

8. SELECT product, AVG(price) FROM sales GROUP BY product;

9. SELECT customer_name, COUNT(*) FROM sales GROUP BY customer_name;

10. SELECT customer_name, SUM(quantity * price) FROM sales GROUP BY customer_name;

11. SELECT region, SUM(quantity * price) FROM sales GROUP BY region HAVING SUM(quantity * price) > 100000;

12. SELECT product, SUM(quantity) FROM sales GROUP BY product HAVING SUM(quantity) > 3;

13. SELECT customer_name, COUNT(*) FROM sales GROUP BY customer_name HAVING COUNT(*) > 1;

14. SELECT product, AVG(price) FROM sales GROUP BY product HAVING AVG(price) > 30000;

15. SELECT product, region, SUM(quantity * price) FROM sales GROUP BY product, region;

16. SELECT region, COUNT(*) FROM sales GROUP BY region HAVING COUNT(*) > 2;

17. SELECT product, MAX(price) FROM sales GROUP BY product;

18. SELECT product, MIN(price) FROM sales GROUP BY product;

19. SELECT MONTH(order_date), SUM(quantity * price) FROM sales GROUP BY MONTH(order_date);

20. SELECT customer_name, SUM(quantity * price) FROM sales GROUP BY customer_name HAVING SUM(quantity * price) > 50000;

Interview Key Points

• WHERE cannot be used with aggregate functions
• HAVING filters grouped data
• Execution Order: FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY