

Day 1 — Complete SQL Quiz Set with Answers

Q1. Difference between WHERE and HAVING?

Answer: WHERE filters rows before grouping; HAVING filters groups after aggregation.

Q2. What does COUNT(DISTINCT customer_id) return?

Answer: It returns the number of unique customers (duplicates removed).

Q3. What is a correlated subquery?

Answer: A subquery that depends on the outer query and runs once per row. Example:
`SELECT e1.name FROM employees e1 WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department = e2.department);`

Q4. Find second-highest salary without LIMIT.

Answer:
`SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees);`

Q5. Purpose of PARTITION BY?

Answer: It divides rows into groups so window functions apply independently in each group.

Q6. Highest salary in each department.

Answer:
`SELECT department, MAX(salary) FROM employees GROUP BY department;`

Q7. Running total revenue per day.

Answer:
`SELECT order_date, SUM(amount) day_revenue, SUM(SUM(amount)) OVER (ORDER BY order_date) running_total FROM orders GROUP BY order_date;`

Q8. Second-highest order amount per customer.

Answer:
`SELECT customer_id, amount FROM (SELECT customer_id, amount, ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY amount DESC) rn FROM orders) t WHERE rn = 2;`

Q9. Customers with more than 3 orders.

Answer:
`SELECT customer_id, COUNT(*) total_orders FROM orders GROUP BY customer_id HAVING COUNT(*) > 3;`

Q10. Difference between INNER JOIN and LEFT JOIN?

Answer: INNER JOIN returns matching rows; LEFT JOIN returns all left rows with NULLs for missing right matches.

Q11. What is ROW_NUMBER()?

Answer: It assigns unique, sequential numbers to each row in a partition.

Q12. What is RANK()?

Answer: RANK() gives same rank for ties but skips the next rank. Example: 1, 1, 3.

Q13. Difference between ROW_NUMBER and RANK?

Answer: ROW_NUMBER is unique; RANK allows ties and skips numbers.

Q14. Write query for 3rd highest salary (no LIMIT).

Answer:

```
SELECT salary FROM (SELECT salary, ROW_NUMBER() OVER (ORDER BY salary DESC) rn
FROM employees) t WHERE rn = 3;
```

Q15. What is a window function?

Answer: A function that performs calculations across related rows without reducing row count.

Q16. Query to find unmatched rows?

Answer:

```
SELECT c.* FROM customers c LEFT JOIN orders o ON c.customer_id=o.customer_id WHERE
o.customer_id IS NULL;
```

Q17. Difference between UNION and UNION ALL?

Answer: UNION removes duplicates; UNION ALL keeps duplicates.

Q18. Query to get customers who spent above average?

Answer:

```
SELECT customer_id, SUM(amount) total FROM orders GROUP BY customer_id HAVING
SUM(amount) > (SELECT AVG(amount) FROM orders);
```

Q19. Query to get top 2 salaries per department?

Answer:

```
SELECT department, name, salary FROM (SELECT *, ROW_NUMBER() OVER (PARTITION BY
department ORDER BY salary DESC) rn FROM employees) t WHERE rn <= 2;
```

Q20. What is INDEX? Why is it used?

Answer: An index speeds up searches on columns used in WHERE, JOIN, ORDER BY. It reduces scan time.