



SQL QUESTIONS

BASIC TO ADVANCED INTERVIEW QUESTION



©vaibhavsaini

1. Q: What is a SQL query?

A: A SQL query is a request for data or information from a database. It is used to retrieve, manipulate, and manage data stored in relational databases.

2. Q: What is the SELECT statement in SQL?

A: The SELECT statement is used to retrieve data from one or more tables in a database. It allows you to specify the columns to retrieve and the conditions to filter the data.

Example:

`, ` ,`

SELECT column1, column2

FROM table_name

WHERE condition;

`, ` ,`

3. Q: What is the difference between WHERE and HAVING clauses in SQL?

A: The WHERE clause is used to filter rows before grouping and aggregation, while the HAVING clause is used to filter groups after grouping and aggregation.

Example:

`, ` ,`

SELECT column1, COUNT(*)

FROM table_name

WHERE condition

GROUP BY column1

HAVING COUNT(*) > 5;

`, ` ,`

4. Q: How do you sort data in SQL?

A: You can use the ORDER BY clause to sort data in ascending (ASC) or descending (DESC) order based on one or more columns.

Example:

`, ` ,`

SELECT column1, column2

FROM table_name

ORDER BY column1 ASC, column2 DESC;

`, ` ,`

5. Q: What is the difference between INNER JOIN and OUTER JOIN in SQL?

A: INNER JOIN returns only the matching rows between two tables, while OUTER JOIN returns all rows from both tables, including non-matching rows.

Example:

`, ` ,`

SELECT column1, column2

FROM table1

INNER JOIN table2

ON table1.column = table2.column;

`, ` ,`

6. Q: How do you perform a self-join in SQL?

A: A self-join is performed when a table is joined with itself. It is useful when you want to compare rows within the same table.

Example:

\ \ \

```
SELECT e1.employee_name, e2.employee_name
FROM employees e1
INNER JOIN employees e2
ON e1.manager_id = e2.employee_id;
```

\ \ \

7. Q: What is the difference between UNION and UNION ALL in SQL?

A: UNION combines the result sets of two or more SELECT statements, removing duplicate rows. UNION ALL also combines result sets but includes all rows, including duplicates.

Example:

\ \ \

```
SELECT column1, column2
FROM table1
UNION
SELECT column1, column2
FROM table2;
```

\ \ \

8. Q: How do you perform a subquery in SQL?

A: A subquery is a query nested within another query. It is used to retrieve data based on the results of another query.

Example:

\ \ \

```
SELECT column1, column2
FROM table1
WHERE column1 IN (SELECT column1 FROM table2 WHERE condition);
```

\ \ \

9. Q: What is the difference between GROUP BY and ORDER BY in SQL?

A: GROUP BY is used to group rows based on one or more columns, while ORDER BY is used to sort the result set based on one or more columns.

Example:

\ \ \

```
SELECT column1, COUNT(*)
FROM table_name
GROUP BY column1
ORDER BY column1 ASC;
```

\ \ \

10. Q: How do you perform a conditional query in SQL?

A: You can use the CASE statement to perform conditional queries in SQL. It allows you to specify different conditions and return different values based on those conditions.

Example:

`, ``,

```
SELECT column1, column2,
CASE
  WHEN condition1 THEN value1
  WHEN condition2 THEN value2
  ELSE value3
END AS new_column
FROM table_name;
```

`, ``,

11. Q: How do you retrieve the top N rows in SQL?

A: You can use the LIMIT or TOP clause to retrieve the top N rows in SQL, depending on the database system you are using.

Example (MySQL):

`, ``,

```
SELECT column1, column2
FROM table_name
LIMIT N;
```

`, ``,

Example (SQL Server):

`, ``,

```
SELECT TOP N column1, column2
FROM table_name;
```

`, ``,

12. Q: How do you calculate the average, sum, minimum, and maximum values in SQL?

A: You can use aggregate functions like AVG, SUM, MIN, and MAX to perform calculations on a column or a group of columns.

Example:

`, ``,

```
SELECT AVG(column1), SUM(column2), MIN(column3), MAX(column4)
FROM table_name;
```

`, ``,

13. Q: How do you retrieve distinct values in SQL?

A: You can use the DISTINCT keyword to retrieve unique values from a column or a combination of columns.

Example:

`, ``,

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

`, ``,

14. Q: How do you perform a wildcard search in SQL?

A: You can use the LIKE operator with wildcard characters (%) to perform a wildcard search in SQL.

Example:

`, ``, `

```
SELECT column1, column2
FROM table_name
WHERE column1 LIKE 'abc%';
`, ``, `
```

15. Q: How do you retrieve data from multiple tables in SQL?

A: You can use JOIN operations like INNER JOIN, LEFT JOIN, RIGHT JOIN, or FULL JOIN to retrieve data from multiple tables based on common columns.

Example:

`, ``, `

```
SELECT column1, column2
FROM table1
INNER JOIN table2
ON table1.column = table2.column;
`, ``, `
```

16. Q: How do you perform a case-insensitive search in SQL?

A: You can use the UPPER or LOWER function to convert the column value and the search term to uppercase or lowercase, respectively, and then perform the search.

Example:

`, ``, `

```
SELECT column1, column2
FROM table_name
WHERE UPPER(column1) = 'SEARCH_TERM';
`, ``, `
```

17. Q: How do you retrieve the current date and time in SQL?

A: You can use the GETDATE or CURRENT_TIMESTAMP function to retrieve the current date and time in SQL.

Example:

`, ``, `

```
SELECT GETDATE();
`, ``, `
```

18. Q: How do you calculate the difference between two dates in SQL?

A: You can use the DATEDIFF function to calculate the difference between two dates in SQL.

Example:

`, ``, `

```
SELECT DATEDIFF(day, start_date, end_date) AS date_difference
FROM table_name;
`, ``, `
```

14. Q: How do you perform a wildcard search in SQL?

A: You can use the LIKE operator with wildcard characters (%) to perform a wildcard search in SQL.

19. Q: How do you perform a NULL check in SQL?

A: You can use the IS NULL or IS NOT NULL operator to check if a column value is NULL or not.

Example:

\ \ \

SELECT column1, column2

FROM table_name

WHERE column1 IS NULL;

\ \ \

20. Q: How do you update data in SQL?

A: You can use the UPDATE statement to modify existing data in a table based on specified conditions.

Example:

\ \ \

UPDATE table_name

SET column1 = value1, column2 = value2

WHERE condition;

\ \ \

21. Q: Consider two tables, "Customers" and "Orders". Write a SQL query to retrieve the names of customers who have placed orders.

A: SELECT Customers.name FROM Customers JOIN Orders ON Customers.customer_id = Orders.customer_id;

22. Q: Given three tables, "Employees", "Departments", and "Employee_Departments", write a SQL query to retrieve the names of employees along with their corresponding department names.

A: SELECT Employees.name, Departments.department_name FROM Employees JOIN Employee_Departments ON Employees.employee_id = Employee_Departments.employee_id JOIN Departments ON Employee_Departments.department_id = Departments.department_id;

23. Q: Consider two tables, "Students" and "Courses". Write a SQL query to retrieve the names of students who have enrolled in the course "Mathematics".

A: SELECT Students.name FROM Students JOIN Courses ON Students.student_id = Courses.student_id WHERE Courses.course_name = 'Mathematics';

24. Q: Given two tables, "Books" and "Authors", write a SQL query to retrieve the titles of books along with their corresponding author names.

A: SELECT Books.title, Authors.author_name FROM Books JOIN Authors ON Books.author_id = Authors.author_id;

25. Q: Consider three tables, "Customers", "Orders", and "Products". Write a SQL query to retrieve the names of customers who have placed orders for the product "Laptop".

A: SELECT Customers.name FROM Customers JOIN Orders ON Customers.customer_id = Orders.customer_id JOIN Products ON Orders.product_id = Products.product_id WHERE Products.product_name = 'Laptop';

26. Q: What is normalization and its different forms?

A: Normalization is the process of organizing data in a database to eliminate redundancy and dependency issues. There are different normal forms, including 1NF, 2NF, 3NF, BCNF, and 4NF. Each normal form has specific rules to ensure data integrity and minimize data duplication.

27. Q: What is the difference between a database and a data warehouse?

A: A database is a structured collection of data that is designed for efficient storage, retrieval, and management. It is typically used for transactional processing. On the other hand, a data warehouse is a large, centralized repository of data that is used for reporting, analysis, and decision-making. It is optimized for read-intensive operations and supports complex queries.

28. Q: How do you handle database transactions?

A: Database transactions ensure the atomicity, consistency, isolation, and durability (ACID) properties of data operations. Transactions are typically managed using the BEGIN TRANSACTION, COMMIT, and ROLLBACK statements. For example:

```
-- \n\nBEGIN TRANSACTION;\n-- Perform database operations\nCOMMIT; -- If successful\nROLLBACK; -- If an error occurs\n-- \n
```

29. Q: What is the purpose of an index in a database?

A: An index is a data structure that improves the speed of data retrieval operations on database tables. It allows for faster searching, sorting, and filtering of data. Indexes are created on specific columns to facilitate efficient data access. For example:

```
-- \n\nCREATE INDEX idx_name ON table_name (column_name);\n-- \n
```

29. Q: Explain the concept of data integrity and how it is maintained.

A: Data integrity ensures the accuracy, consistency, and reliability of data. It is maintained through various mechanisms such as constraints, validations, and referential integrity. Constraints like primary key, unique key, and foreign key enforce data integrity rules. Validations ensure that data meets specific criteria. Referential integrity ensures the consistency of relationships between tables.

30. Q: How do you optimize database performance?

A: Database performance can be optimized by various techniques such as:

- Proper indexing of tables
- Optimizing SQL queries
- Partitioning large tables
- Caching frequently accessed data
- Regular database maintenance (e.g., updating statistics)
- Optimizing hardware resources (e.g., memory, disk I/O)

31. Q: What is the difference between a clustered and non-clustered index?

A: A clustered index determines the physical order of data in a table. There can be only one clustered index per table. In contrast, a non-clustered index is a separate structure that contains a copy of the indexed columns and a pointer to the actual data. A table can have multiple non-clustered indexes.

32. Q: What is the role of a database administrator (DBA)?

A: A DBA is responsible for managing and maintaining the database system. Their role includes tasks such as database installation, configuration, security management, performance tuning, backup and recovery, and ensuring data integrity. They also handle user access, database monitoring, and troubleshooting.

33. Q: How do you handle database backups and recovery?

A: Database backups are essential for data protection. Common backup strategies include full backups, incremental backups, and differential backups. Recovery involves restoring the database from a backup in case of data loss or system failure. DBAs use techniques like point-in-time recovery, transaction logs, and backup validation to ensure data integrity and minimize downtime.

34. Q: Consider a table "Employees" with columns "EmployeeID" and "Salary". Write a SQL query to retrieve the names of employees who have a salary greater than the average salary.

A: `SELECT Name FROM Employees WHERE Salary > (SELECT AVG(Salary) FROM Employees);`

35. Q: Consider two tables, "Customers" and "Orders". Write a SQL query to retrieve the names of customers who have placed more than 3 orders.

A: `SELECT Name FROM Customers WHERE CustomerID IN (SELECT CustomerID FROM Orders GROUP BY CustomerID HAVING COUNT(*) > 3);`

36. Q: Consider a table "Products" with columns "ProductID" and "Price". Write a SQL query to retrieve the names of products that have a price higher than the maximum price of products in a specific category.

A: `SELECT Name FROM Products WHERE Price > (SELECT MAX(Price) FROM Products WHERE Category = 'SpecificCategory');`

37. Write a SQL query to find the second highest salary from an "Employees" table.

A:

`-- sql`

```
SELECT MAX(salary) AS second_highest_salary
FROM Employees
WHERE salary < (SELECT MAX(salary) FROM Employees);
--
```

38. Write a SQL query to find the most common word in a "Text" column of a table.

A:

`-- sql`

```
SELECT word, COUNT(*) AS frequency
FROM (
    SELECT REGEXP_SPLIT_TO_TABLE(Text, E'\\s+') AS word
    FROM YourTable
) AS words
GROUP BY word
ORDER BY frequency DESC
LIMIT 1;
--
```


39. Write a SQL query to find the top N customers with the highest total order amounts.

A:

```
` `` `sql
SELECT customer_id, SUM(order_amount) AS total_order_amount
FROM Orders
GROUP BY customer_id
ORDER BY total_order_amount DESC
LIMIT N;
` `` `
```

40. Write a SQL query to find the average time between consecutive log entries in a "Logs" table.

A:

```
` `` `sql
SELECT AVG(time_diff) AS average_time_diff
FROM (
  SELECT EXTRACT(EPOCH FROM (log_time - lag(log_time) OVER (ORDER BY log_time))) AS time_diff
  FROM Logs
) AS time_diffs;
` `` `
```

41. Q: What is PL/SQL?

A: PL/SQL (Procedural Language/Structured Query Language) is a procedural extension of SQL used for writing stored procedures, functions, triggers, and packages in Oracle Database.

42. Q: How do you declare a variable in PL/SQL?

A: Variables in PL/SQL are declared using the DECLARE keyword, followed by the variable name, data type, and optional initial value. For example:

```
DECLARE
my_variable VARCHAR2(50) := 'Hello';
BEGIN
-- Code goes here
END;
```

43. Q: What is a stored procedure in PL/SQL?

A: A stored procedure is a named PL/SQL block that performs a specific task. It can accept input parameters, execute SQL statements, and return output parameters or a result set.

44. Q: How do you call a stored procedure in PL/SQL?

A: To call a stored procedure in PL/SQL, you use the EXECUTE or EXEC keyword followed by the procedure name and any required parameters. For example:

```
EXECUTE my_procedure(param1, param2);
```

45. Q: What is a trigger in PL/SQL?

A: A trigger in PL/SQL is a named PL/SQL block that is automatically executed in response to a specific event, such as an INSERT, UPDATE, or DELETE operation on a table.

46. Q: What are the types of triggers in PL/SQL?

A: There are two types of triggers in PL/SQL: row-level triggers and statement-level triggers. Row-level triggers are executed once for each affected row, while statement-level triggers are executed once for each triggering statement.

47. Q: How do you create a trigger in PL/SQL?

A: To create a trigger in PL/SQL, you use the CREATE TRIGGER statement followed by the trigger name, event, and trigger body. For example:

```
CREATE TRIGGER my_trigger
AFTER INSERT ON my_table
FOR EACH ROW
BEGIN
    -- Trigger logic goes here
END;
```

48. Q: What is a package in PL/SQL?

A: A package in PL/SQL is a named collection of related procedures, functions, variables, and cursors. It provides a way to organize and encapsulate related code for easier maintenance and reusability.

49. Q: How do you create a package in PL/SQL?

A: To create a package in PL/SQL, you use the CREATE PACKAGE statement followed by the package name and the package specification and body. The specification contains the public interface, while the body contains the implementation details.

50. Q: What is the difference between a stored procedure and a function in PL/SQL?

A: The main difference between a stored procedure and a function in PL/SQL is that a stored procedure does not return a value, while a function returns a value. Functions can be used in SQL statements, while stored procedures cannot.

In summary, the journey through 50 SQL questions, spanning from basic to advanced topics, equips you with the tools to query, manipulate, and optimize databases. From foundational SELECT statements to intricate joins, subqueries, and normalization, you gain the skills to handle complex data scenarios, enhancing your data management capabilities.



THANKYOU