

# 100 SQL interview questions along with their answers

## Basic SQL Questions

### 1. What is SQL?

- SQL (Structured Query Language) is a standard programming language for managing and manipulating relational databases.

### 2. What are the different types of SQL commands?

- The main types are DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), and TCL (Transaction Control Language).

### 3. What is a primary key?

- A primary key is a unique identifier for a record in a table, ensuring that no two rows have the same key.

### 4. What is a foreign key?

- A foreign key is a field (or a collection of fields) in one table that uniquely identifies a row in another table, establishing a relationship between the two.

### 5. What is a unique key?

- A unique key ensures that all values in a column are different from one another, allowing NULL values.

### 6. What is a join? Explain different types of joins.

- A join is used to combine rows from two or more tables based on a related column. Types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

### 7. What is a subquery?

- A subquery is a query nested inside another query, used to perform operations that depend on the result of the outer query.

### 8. What is a view?

- A view is a virtual table based on the result of a SELECT query, which can simplify complex queries.

### 9. What is an index?

- An index is a database object that improves the speed of data retrieval operations on a database table.

### 10. What is a stored procedure?

- A stored procedure is a prepared SQL code that can be saved and reused, allowing for modular programming.

### 11. What is a trigger?

- A trigger is a set of SQL statements that automatically execute in response to certain events on a particular table.

### 12. What is normalization?

- Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity.

### 13. What are the different normal forms?

- The normal forms are 1NF (First Normal Form), 2NF (Second Normal Form), 3NF (Third Normal Form), and BCNF (Boyce-Codd Normal Form).

#### 14. What is denormalization?

- Denormalization is the process of intentionally introducing redundancy into a database for the purpose of improving query performance.

#### 15. What is a constraint?

- A constraint is a rule applied to a column or table to enforce data integrity.

#### 16. What is a composite key?

- A composite key is a combination of two or more columns in a table that uniquely identifies a row.

#### 17. What is a candidate key?

- A candidate key is a column, or a set of columns, that can qualify as a potential primary key.

#### 18. What is the difference between DELETE and TRUNCATE?

- DELETE removes rows one at a time and can be rolled back; TRUNCATE removes all rows in a table without logging individual row deletions.

#### 19. What is the difference between WHERE and HAVING clauses?

- WHERE is used to filter records before aggregation, while HAVING is used to filter records after aggregation.

#### 20. What is the difference between UNION and UNION ALL?

- UNION combines the results of two queries and removes duplicates; UNION ALL combines the results without removing duplicates.

### Intermediate SQL Questions

#### 21. What is a self-join?

- A self-join is a join in which a table is joined with itself.

#### 22. What is a cross join?

- A cross join returns the Cartesian product of two tables, combining every row of the first table with every row of the second.

#### 23. What is a correlated subquery?

- A correlated subquery is a subquery that references columns from the outer query.

#### 24. What is an alias in SQL?

- An alias is a temporary name given to a table or column for the duration of a query.

#### 25. What is the difference between COUNT(\*) and COUNT(column\_name)?

- COUNT(\*) counts all rows, including duplicates and NULLs, while COUNT(column\_name) counts only non-NULL values in that column.

## 26. What are aggregate functions?

- Aggregate functions perform a calculation on a set of values and return a single value (e.g., SUM, AVG, COUNT).

## 27. What are scalar functions?

- Scalar functions return a single value based on the input value (e.g., UPPER, LOWER, ROUND).

## 28. What is a cursor?

- A cursor is a database object used to retrieve, manipulate, and navigate through a result set one row at a time.

## 29. What is a transaction?

- A transaction is a sequence of operations performed as a single logical unit of work, which is either fully completed or fully rolled back.

## 30. What are the ACID properties?

- ACID stands for Atomicity, Consistency, Isolation, and Durability, ensuring reliable processing of database transactions.

## 31. What is the difference between ROLLBACK and COMMIT?

- ROLLBACK undoes changes made in the current transaction, while COMMIT saves all changes made during the transaction.

## 32. What is a sequence in SQL?

- A sequence is a database object that generates a sequential number, often used for primary keys.

## 33. What is the difference between CHAR and VARCHAR?

- CHAR is a fixed-length string data type, while VARCHAR is a variable-length string data type.

## 34. What is a temporary table?

- A temporary table is a short-lived table that exists temporarily during a session or transaction.

## 35. What is a common table expression (CTE)?

- A CTE is a temporary result set that can be referenced within a SELECT, INSERT, UPDATE, or DELETE statement.

## 36. What is the difference between RANK() and DENSE\_RANK()?

- RANK() provides a rank number with gaps for ties, while DENSE\_RANK() provides consecutive rank numbers without gaps.

## 37. What is the GROUP BY clause used for?

- The GROUP BY clause is used to group rows that have the same values in specified columns for aggregation.

## 38. What is the difference between INNER JOIN and OUTER JOIN?

- INNER JOIN returns rows with matching values in both tables, while OUTER JOIN returns all rows from one table and matched rows from the other.

## 39. What are the different types of indexes?

- Types of indexes include unique indexes, composite indexes, full-text indexes, and clustered and non-clustered indexes.

**40. What is a clustered index?**

- A clustered index determines the physical order of data in a table; there can be only one clustered index per table.

**41. What is a non-clustered index?**

- A non-clustered index does not alter the physical order of the table and can have multiple instances in a table.

**42. What is a bitmap index?**

- A bitmap index uses bit arrays for indexing, suitable for columns with a limited number of distinct values.

**43. What is the difference between OLTP and OLAP?**

- OLTP (Online Transaction Processing) focuses on transaction-oriented tasks, while OLAP (Online Analytical Processing) is used for data analysis and reporting.

**44. What is a data warehouse?**

- A data warehouse is a centralized repository that stores large amounts of historical data for analysis and reporting.

**45. What is a data mart?**

- A data mart is a subset of a data warehouse, focused on a specific business line or team.

**46. What is ETL?**

- ETL (Extract, Transform, Load) is the process of extracting data from different sources, transforming it into a suitable format, and loading it into a data warehouse.

**47. What is a star schema?**

- A star schema is a type of database schema that organizes data into fact and dimension tables, resembling a star shape.

**48. What is a snowflake schema?**

- A snowflake schema is a more normalized version of the star schema, where dimension tables are further broken down into sub-dimensions.

**49. What is data integrity?**

- Data integrity refers to the accuracy and consistency of data over its lifecycle.

**50. What is referential integrity?**

- Referential integrity ensures that relationships between tables remain consistent, particularly with foreign keys.

## Advanced SQL Questions

### 51. What is a pivot table in SQL?

- A pivot table summarizes data from a detailed table, allowing users to analyze it in a multidimensional format.

### 52. How do you create a pivot table in SQL?

- You can create a pivot table using the PIVOT function, which allows you to transform rows into columns.

### 53. What is the difference between PIVOT and UNPIVOT?

- PIVOT rotates data from rows to columns, while UNPIVOT converts columns back into rows.

### 54. What are window functions?

- Window functions perform calculations across a set of table rows related to the current row, without collapsing the result set.

### 55. What is the ROW\_NUMBER() function?

- ROW\_NUMBER() assigns a unique sequential integer to rows within a partition of a result set.

### 56. What is the NTILE() function?

- NTILE() divides a result set into a specified number of groups and assigns a group number to each row.

### 57. What is the LAG() function?

- LAG() accesses data from a previous row in the same result set without the need for a self-join.

### 58. What is the LEAD() function?

- LEAD() accesses data from the next row in the same result set, similar to LAG() but forward-looking.

### 59. What is the difference between a table and a view?

- A table stores actual data, while a view is a virtual table that provides a representation of the data from one or more tables.

### 60. What is the difference between a materialized view and a regular view?

- A materialized view stores the result of a query physically on disk, allowing for faster access, while a regular view does not store data.

### 61. What is database normalization and why is it important?

- Normalization organizes data to reduce redundancy and dependency, improving data integrity and efficiency.

### 62. Explain 1NF, 2NF, 3NF, and BCNF.

- 1NF (First Normal Form) requires atomicity of values. 2NF (Second Normal Form) eliminates partial dependencies. 3NF (Third Normal Form) removes transitive dependencies. BCNF (Boyce-Codd Normal Form) is an enhancement of 3NF that resolves anomalies caused by functional dependencies.

### 63. What is indexing and how does it improve query performance?

- Indexing creates a data structure that improves the speed of data retrieval operations by providing quick access paths.

### 64. What are the pros and cons of indexing?

- Pros: Faster data retrieval. Cons: Slower data modification (insert/update/delete) and increased storage requirements.

### 65. What is query optimization?

- Query optimization is the process of making a query run more efficiently by choosing the best execution plan.

### 66. What are the different methods to optimize a SQL query?

- Methods include using indexes, avoiding SELECT \*, optimizing joins, reducing subqueries, and analyzing execution plans.

### 67. How do you use EXPLAIN or EXPLAIN PLAN in SQL?

- EXPLAIN or EXPLAIN PLAN provides information about how a SQL query will be executed, including the access methods and join types used.

### 68. What are the different types of subqueries?

- Types include single-row subqueries, multi-row subqueries, and correlated subqueries.

### 69. What is a recursive query?

- A recursive query refers to a query that references itself, often used for hierarchical data.

### 70. What is the difference between EXCEPT and NOT IN?

- EXCEPT returns distinct rows from the first query that are not found in the second, while NOT IN checks for exclusion of specific values from a list.

## Practical SQL Questions

### 71. Write a SQL query to find the second highest salary from a table named `Employees`.

```
SELECT MAX(Salary) AS SecondHighestSalary FROM Employees  
WHERE Salary < (SELECT MAX(Salary) FROM Employees);
```

### 72. Write a SQL query to find the names of employees who have joined in the last 30 days.

```
SELECT Name FROM Employees WHERE JoinDate >= DATEADD(DAY, -30, GETDATE());
```

### 73. Write a SQL query to retrieve the top N records from a table.

```
SELECT TOP N * FROM Employees;
```

### 74. Write a SQL query to find duplicate records in a table.

```
SELECT Name, COUNT(*) FROM Employees GROUP BY Name HAVING COUNT(*) > 1;
```

**75. Write a SQL query to delete duplicate records from a table.**

WITH CTE AS

(SELECT \*,ROW\_NUMBER() OVER (PARTITION BY Name ORDER BY Id) AS RowNumFROM Employees)

DELETE FROM CTE WHERE RowNum > 1;

**76. Write a SQL query to find the nth highest salary from a table.**

SELECT DISTINCT Salary FROM Employees ORDER BY Salary DESC OFFSET N-1 ROWS

FETCH NEXT 1 ROW ONLY;

**77. Write a SQL query to find the common records between two tables.**

SELECT \* FROM Employees INTERSECT SELECT \* FROM Managers;

**78. Write a SQL query to find the records that are present in one table but not in another.**

SELECT \* FROM Employees EXCEPT SELECT \* FROM Managers;

**79. Write a SQL query to retrieve the first 10 characters of a string**

SELECT LEFT(Name, 10) AS First10Characters FROM Employees;

**80. Write a SQL query to concatenate two strings.**

SELECT CONCAT(FirstName, ' ', LastName) AS FullName FROM Employees;

**81. Write a SQL query to convert a string to uppercase.**

SELECT UPPER(Name) AS UppercaseName FROM Employees;

**82. Write a SQL query to find the length of a string.**

SELECT LEN(Name) AS NameLength FROM Employees;

**83. Write a SQL query to find the position of a substring in a string.**

SELECT CHARINDEX('substring', Name) AS SubstringPosition FROM Employees;

**84. Write a SQL query to replace a substring in a string.**

SELECT REPLACE(Name, 'old', 'new') AS NewName FROM Employees;

**85. Write a SQL query to remove leading and trailing spaces from a string.**

SELECT LTRIM(RTRIM(Name)) AS TrimmedName FROM Employees;

**86. Write a SQL query to find the difference between two dates.**

SELECT DATEDIFF(DAY, StartDate, EndDate) AS DateDifference FROM Projects;

**87. Write a SQL query to add a number of days to a date.**

SELECT DATEADD(DAY, 10, StartDate) AS NewDate FROM Projects;

**88. Write a SQL query to extract the year from a date.**

SELECT YEAR(JoinDate) AS JoinYear FROM Employees;

**89. Write a SQL query to extract the month from a date.**

```
SELECT MONTH(JoinDate) AS JoinMonth FROM Employees;
```

**90. Write a SQL query to extract the day from a date.**

```
SELECT DAY(JoinDate) AS JoinDay FROM Employees;
```

**91. Write a SQL query to convert a string to a date.**

```
SELECT CONVERT(DATE, '2024-01-01') AS ConvertedDate;
```

**92. Write a SQL query to find the number of employees in each department.**

```
SELECT Department, COUNT(*) AS EmployeeCount FROM Employees GROUP BY Department;
```

**93. Write a SQL query to find the total salary of each department.**

```
SELECT Department, SUM(Salary) AS TotalSalary FROM Employees GROUP BY Department;
```

**94. Write a SQL query to find the average salary of each department.**

```
SELECT Department, AVG(Salary) AS AverageSalary FROM Employees GROUP BY Department;
```

**95. Write a SQL query to find the maximum salary of each department.**

```
SELECT Department, MAX(Salary) AS MaxSalary FROM Employees GROUP BY Department;
```

**96. Write a SQL query to find the minimum salary of each department.**

```
SELECT Department, MIN(Salary) AS MinSalary FROM Employees GROUP BY Department;
```

**97. Write a SQL query to find the department with the highest number of employees.**

```
SELECT TOP 1 Department, COUNT(*) AS EmployeeCount FROM Employees  
GROUP BY Department ORDER BY EmployeeCount DESC;
```

**98. Write a SQL query to find the department with the highest total salary.**

```
SELECT TOP 1 Department, SUM(Salary) AS TotalSalary FROM Employees  
GROUP BY Department ORDER BY TotalSalary DESC;
```

**99. Write a SQL query to find the employees who earn more than the average salary.**

```
SELECT * FROM Employees WHERE Salary > (SELECT AVG(Salary) FROM Employees);
```

**100. Write a SQL query to find the employees who earn the highest salary in each department.**

```
SELECT e.Department, e.Name, e.Salary FROM Employees e  
JOIN (SELECT Department, MAX(Salary) AS MaxSalary FROM Employees GROUP BY Department) AS  
maxSalaries ON e.Department = maxSalaries.Department AND e.Salary = maxSalaries.MaxSalary;
```

**Thank you! Connect with me on LinkedIn for more updates: [@Aditya Raja](#)**

