

# SQL Interview Questions



can you give me SQL also

Absolutely! Since you're preparing for interviews and like things organized by topic, I'll create **SQL interview questions by topic**, with **100 questions per topic** (like you're doing with Python and DSA). Here's a suggested breakdown of topics—let me know if you want to customize it:

## ✓ SQL Interview Topics (100 Questions Each)

1. **Basic SQL Queries**
2. **SQL Joins (INNER, LEFT, RIGHT, FULL)**
3. **SQL Aggregations (GROUP BY, HAVING, COUNT, etc.)**
4. **SQL Subqueries and Nested Queries**
5. **SQL Set Operations (UNION, INTERSECT, EXCEPT)**
6. **SQL Window Functions (RANK, DENSE\_RANK, etc.)**
7. **SQL Data Definition Language (DDL)**
8. **SQL Data Manipulation Language (DML)**
9. **SQL Indexes and Performance Tuning**
10. **Advanced SQL Scenarios (Case Studies & Real-world Queries)**

Which topic would you like to start with? I can give you **Topic 1: Basic SQL Queries (100 questions)** right away if that works for you.



Topic 1: Basic SQL Queries (100 questions)

give me each topic 50 questions

Perfect! Here are the **first 50 questions for Topic 1: Basic SQL Queries**:

## 📖 SQL Interview Questions: Topic 1 - Basic SQL Queries (Part 1 of 2)

1. What is SQL and why is it used?
2. What is the difference between `SELECT` and `SELECT DISTINCT`?
3. How do you retrieve all the columns from a table?
4. How do you select specific columns from a table?

5. How do you filter results using a `WHERE` clause?
6. How do you write a query to find rows with NULL values?
7. How do you use comparison operators like `=`, `>`, `<`, `!=` in a SQL query?
8. How do you sort the result set using `ORDER BY`?
9. What is the default sort order in `ORDER BY`?
10. How do you sort data in descending order?
11. What is the use of `LIMIT` in SQL?
12. What is the difference between `LIMIT` and `TOP`?
13. How do you rename a column in the result set?
14. What is the use of `AS` in SQL?
15. How do you use `BETWEEN` in a SQL query?
16. How do you use `IN` in a SQL query?
17. What is the difference between `IN` and `=`?
18. How do you use `NOT` with `IN`, `LIKE`, or `BETWEEN`?
19. How do you use pattern matching with `LIKE`?
20. What do the `%` and `_` wildcards mean in `LIKE`?
21. How do you combine conditions using `AND` and `OR`?
22. How do parentheses affect logical condition evaluation in `WHERE`?
23. How do you retrieve unique values from a column?
24. How do you retrieve rows with non-null values in a column?
25. What is the difference between `IS NULL` and `= NULL`?
26. How do you fetch the first N records from a table?
27. How do you get the count of rows in a table?
28. How do you get the number of distinct values in a column?
29. How do you use `SELECT` with an arithmetic operation?
30. How do you concatenate strings in SQL?
31. What is a column alias? How is it useful?
32. What is the difference between `WHERE` and `HAVING`?
33. Can you use column aliases in the `WHERE` clause?
34. How do you retrieve rows where a column starts with a specific letter?
35. How do you retrieve rows where a column ends with a specific letter?
36. How do you retrieve rows where a column contains a specific substring?
37. How do you retrieve the top 3 highest values in a column?
38. How do you retrieve the bottom 5 lowest values in a column?
39. How do you check for even or odd values in a column?

40. How do you write a query to retrieve rows that are not in another list?
41. What happens if you don't use `WHERE` in a `DELETE` query?
42. Can you run a `SELECT` query without a `FROM` clause?
43. What is the result of a `SELECT 2 + 2;` query?
44. What is the default name of a column when no alias is given to an expression?
45. Can you use functions like `UPPER()` or `LOWER()` in `SELECT`?
46. How do you apply a condition on the result of a function in a column?
47. What is the difference between `= NULL` and `IS NULL`?
48. How do you count rows that have a specific condition?
49. How do you use a subquery inside a `WHERE` clause?
50. What happens if the subquery returns more than one value in a single-value comparison?

---

Let me know when you're ready for **Part 2 (questions 51–100)** or if you want to start the next topic!



## Part 2 (questions 51–100)

Awesome! Here's **Part 2 of Topic 1: Basic SQL Queries** with questions **51–100**:

---

### SQL Interview Questions: Topic 1 - Basic SQL Queries (Part 2 of 2)

51. How do you retrieve records where one column value is greater than another column value?
52. How can you alias a table name in a query?
53. What is the difference between `*` and explicitly listing column names in `SELECT`?
54. How do you calculate the length of a string in SQL?
55. How do you convert a string to uppercase or lowercase in SQL?
56. How do you extract a substring from a column in SQL?
57. How do you round numeric values in a `SELECT` query?
58. How do you find the current date or time in SQL?
59. How do you retrieve records between two dates?
60. How do you filter data based on a year or month from a date column?
61. How do you perform calculations using multiple columns in `SELECT`?
62. How do you replace NULL values in the result set?
63. How do you use `CASE` for conditional logic in `SELECT`?
64. How do you write a query to show a new column based on conditional logic?
65. What is the difference between `NULL`, 0, and empty string in SQL?
66. How do you fetch every nth row from a table (e.g., every 5th record)?

67. How do you filter rows based on a condition applied to a computed column?
68. How do you find duplicate values in a column?
69. How do you retrieve rows with duplicate values only?
70. How do you retrieve rows without any duplicates?
71. How do you get the maximum and minimum values in a column?
72. How do you calculate the average of a numeric column?
73. How do you group results by one or more columns?
74. How do you filter grouped results using ``HAVING``?
75. How do you count the number of records in each group?
76. How do you find the sum of a column's values per group?
77. How do you find the number of employees per department (group example)?
78. How do you display the highest salary in each department?
79. How do you sort each group by a specific column?
80. How do you combine multiple filtering conditions with grouping?
81. How do you retrieve only the rows where a column has no duplicate values?
82. How do you find the second highest salary from a table?
83. How do you fetch records that occur more than once in a column?
84. How do you avoid duplicates using ``DISTINCT`` with multiple columns?
85. What happens if you use ``GROUP BY`` on all selected columns?
86. How do you select rows that exist in one table but not in another?
87. How do you check if a table has any data (non-empty)?
88. How do you write a query that checks if a value exists in a column?
89. How do you get a list of all table names in a database?
90. How do you get a list of all column names for a given table?
91. How do you check the data type of a column?
92. How do you write a query that returns no rows intentionally?
93. How do you use the ``EXISTS`` keyword in SQL?
94. How do you use the ``ALL`` and ``ANY`` operators in SQL?
95. How do you use the ``COALESCE()`` function?
96. How do you prevent SQL errors due to NULL values in expressions?
97. How do you check if a column contains only numeric values?
98. What is the difference between ``TRIM()``, ``LTRIM()``, and ``RTRIM()``?
99. How do you return a fixed value in all rows using SQL?
100. How do you write a query that returns a constant column and a calculated column?

Ready to move to **Topic 2: SQL Joins** or want to revise this topic with answers and explanations?



## Topic 2: SQL Joins

Great! Here are **50 interview questions for Topic 2: SQL Joins (Part 1 of 2)**.

### SQL Interview Questions: Topic 2 – SQL Joins (Part 1 of 2)

1. What is a SQL JOIN?
2. What are the different types of JOINS in SQL?
3. What is an INNER JOIN and how does it work?
4. What is a LEFT JOIN and how does it work?
5. What is a RIGHT JOIN and how does it work?
6. What is a FULL OUTER JOIN?
7. What is the difference between INNER JOIN and LEFT JOIN?
8. What happens if you use a JOIN without an ON condition?
9. What is a CROSS JOIN?
10. How does a CROSS JOIN differ from an INNER JOIN?
11. When should you use a LEFT JOIN instead of an INNER JOIN?
12. How do you perform a self join?
13. What is a self join used for?
14. Can you join a table with itself? Give an example.
15. How do NULLs affect JOIN operations?
16. How do you join more than two tables?
17. How do you write a JOIN using table aliases?
18. Can you use JOINS in combination with GROUP BY?
19. Can you use JOINS inside subqueries?
20. How do you filter joined data using a WHERE clause?
21. How do you filter joined data using an ON clause?
22. What's the difference between ON and WHERE when used in JOINS?
23. What is an equi join?
24. What is a non-equi join?
25. How do you perform a non-equi join in SQL?
26. How can you write a JOIN using `USING()` instead of `ON`?
27. What are natural joins and how do they work?

28. Why should you be cautious using NATURAL JOIN?
29. Can you use JOINS with aggregate functions?
30. How do you write a query that joins a table with a derived table (subquery)?
31. What is the difference between a JOIN and a subquery?
32. How do you handle duplicate column names after a JOIN?
33. How do you join tables with different column names?
34. How do you join tables based on multiple conditions?
35. What is a theta join?
36. How do you simulate FULL OUTER JOIN in MySQL (which lacks native support)?
37. Can JOINS affect query performance? How?
38. How can you optimize queries involving JOINS?
39. How does indexing affect JOIN performance?
40. How do you write a query to join 3 tables with different keys?
41. How do you get unmatched rows from the left table only?
42. How do you get unmatched rows from both tables?
43. Can you use ORDER BY after a JOIN?
44. Can you join on expressions like `ON a.id = b.id + 1`?
45. How do you join on non-primary and non-foreign keys?
46. What is the difference between `JOIN` and `UNION`?
47. How do you detect cartesian products in a JOIN?
48. What are the best practices for writing JOIN queries?
49. How do you write a JOIN to display all employees and their managers (self join example)?
50. How do you write a JOIN to show customers who placed orders and those who didn't?

---

Let me know when you're ready for **Part 2 (questions 51–100)** or want to jump to another topic!



Part 2 (questions 51–100)

---

Here's **Part 2 of Topic 2: SQL Joins** — questions **51–100**:

## **SQL Interview Questions: Topic 2 – SQL Joins (Part 2 of 2)**

51. How do you join two tables based on a date range condition?
52. How do you perform a LEFT JOIN and filter only unmatched rows?
53. How do you perform an INNER JOIN with a condition on a third column?
54. How do you join tables with composite keys?

55. How can you prevent duplicate rows when using JOINS?
56. What is a semi-join, and how is it different from a regular JOIN?
57. What is an anti-join, and how can you implement it in SQL?
58. How do you find records in one table that do not exist in another using JOINS?
59. How do you simulate a FULL OUTER JOIN using UNION?
60. How can you write a query to join a table and include a default value for NULL matches?
61. What's the result of joining a table to itself without aliasing?
62. How do table aliases help when writing joins?
63. How do you join a table with a result from a GROUP BY query?
64. What is a derived table in the context of JOINS?
65. How do you use window functions alongside JOINS?
66. Can you use JOINS inside CTEs (Common Table Expressions)?
67. How do JOINS behave differently in MySQL vs PostgreSQL?
68. What happens when you use aggregate functions in JOINS without GROUP BY?
69. What is a chained JOIN and how do you write one?
70. How do you debug unexpected row counts in JOIN queries?
71. How do you write a query to get students and the courses they are not enrolled in?
72. How do you write a query to list customers and the products they never bought?
73. How do JOINS work when one of the tables has no rows?
74. What is a star schema and how are JOINS used in it?
75. How do JOINS work in snowflake schema?
76. How do you join a table with a pivoted table?
77. How do you use COALESCE with JOINS?
78. How do you join a lookup table to add readable labels to numeric codes?
79. Can you use aggregate filters like `HAVING COUNT(*) > 1` after a JOIN?
80. How do you find mismatched values between two tables using JOIN?
81. How do you write a query that joins employee data with department data?
82. How can you combine JOINS with CASE expressions?
83. How do you join and simultaneously filter with WHERE and ON clauses?
84. Can you use an ORDER BY clause inside a subquery in a JOIN?
85. How do you perform a many-to-many JOIN?
86. What are bridge tables and how do you JOIN them?
87. How do you retrieve the most recent record using JOIN?
88. How do you handle JOINS with denormalized data?
89. What's the difference between using JOINS and window functions for ranking?

90. How do you write a query to get each employee's department and department head?
91. How do you join a large transactional table with a small lookup table efficiently?
92. What tools or query hints can help optimize JOIN performance in SQL Server?
93. How do temporary tables help in simplifying JOIN queries?
94. What's the role of EXPLAIN (or EXPLAIN PLAN) in understanding JOIN performance?
95. How do you avoid using JOINS in performance-sensitive queries?
96. When is denormalization preferred over complex JOINS?
97. What is hash join vs nested loop join vs merge join?
98. How do database engines internally handle JOINS?
99. What is join elimination in SQL optimizers?
100. How do you track the execution time of JOIN queries in your RDBMS?

Ready to move to **Topic 3: SQL Aggregations (GROUP BY, HAVING, etc.)** or want solutions/explanations for any of these?



Topic 3: SQL Aggregations (GROUP BY, HAVING, etc.)

Awesome! Here's **Topic 3: SQL Aggregations (GROUP BY, HAVING, etc.) – Part 1 (Questions 1–50):**



## SQL Interview Questions: Topic 3 – Aggregations (Part 1 of 2)

1. What are aggregate functions in SQL?
2. Name the most commonly used aggregate functions.
3. What does the `COUNT()` function do?
4. How does `COUNT(*)` differ from `COUNT(column_name)`?
5. What does `SUM()` do in SQL?
6. What does `AVG()` do in SQL?
7. How does `MIN()` function work?
8. How does `MAX()` function work?
9. Can you use aggregate functions without `GROUP BY`?
10. How does `GROUP BY` work in SQL?
11. Can you use multiple columns in a `GROUP BY` clause?
12. What's the difference between `WHERE` and `HAVING`?
13. Why can't you use aggregate functions in `WHERE`?
14. How do you filter grouped records using `HAVING`?
15. How do you get the number of employees in each department?



16. How do you find the average salary per department?
17. How do you find the departments with more than 10 employees?
18. How do you get the highest salary per department?
19. How do you use `GROUP BY` with `ORDER BY`?
20. How do you sort groups based on aggregated values?
21. Can you use column aliases in `GROUP BY` or `HAVING`?
22. Can you use `GROUP BY` on an expression (e.g., `YEAR(hire_date)`)?
23. How do you group by calculated fields?
24. How do you get the count of NULL values in a column?
25. How do you exclude NULL values from an aggregate function?
26. What happens if you group by a column and it has NULLs?
27. How do you write a query that shows the total sales per product?
28. How do you calculate the percentage contribution of each group to the total?
29. Can you use `DISTINCT` inside aggregate functions?
30. What's the difference between `COUNT(DISTINCT col)` and `COUNT(col)`?
31. How do you write a query that counts employees hired each year?
32. How do you get the average order value per customer?
33. How do you write a query to find the customer who spent the most?
34. How do you group and filter using aliases in the `HAVING` clause?
35. How do you use `CASE` with aggregate functions?
36. How do you aggregate conditionally using `CASE WHEN`?
37. How do you calculate the running total using `SUM()`?
38. Can you use aggregate functions in the `SELECT` clause along with regular columns?
39. How does SQL treat rows with NULL values in aggregation?
40. How do you write a query to show the top 3 departments with the highest average salary?
41. How do you write a query to show the second highest average order value per region?
42. How do you use `LIMIT` with `GROUP BY`?
43. How do you get the minimum and maximum values in one query per group?
44. Can you use `HAVING` without `GROUP BY`?
45. What is the default grouping behavior of SQL when `GROUP BY` is omitted?
46. How do you count the number of orders per customer and list only those with more than 3 orders?
47. How do you find the total revenue per category, ordered by revenue descending?
48. How do you use `ROLLUP` to produce subtotals in a grouped query?
49. How do you use `CUBE` to create cross-tab summaries?
50. What's the difference between `ROLLUP`, `CUBE`, and `GROUPING SETS`?

Let me know when you're ready for **Part 2 (questions 51–100)** or want to jump to another topic!



## Part 2 (questions 51–100)

Here's **Part 2 of Topic 3: SQL Aggregations (GROUP BY, HAVING, etc.)** — questions **51–100**:



### SQL Interview Questions: Topic 3 – Aggregations (Part 2 of 2)

51. How do you use the `GROUPING()` function to identify subtotal rows in `ROLLUP` or `CUBE`?
52. How do you filter out subtotal rows after using `ROLLUP`?
53. Can you use aggregate functions in subqueries?
54. How do you compare aggregate results across groups using subqueries?
55. How do you calculate a difference between two aggregate values?
56. How do you use aggregate functions with `JOINS`?
57. How do you calculate the average number of items per order?
58. How do you group by date parts like year, month, and day?
59. How do you group by week or quarter in SQL?
60. How do you format grouped date output for better readability?
61. How do you get the most recent order per customer using aggregation?
62. How do you combine aggregated data from multiple tables?
63. How do you find departments where the average salary is above company average?
64. How do you use aggregate results in a `CASE WHEN` condition?
65. How do you use `HAVING` with multiple aggregate conditions?
66. How do you calculate ratios between different aggregate values (e.g., male/female ratio)?
67. How do you group by derived values like age buckets or salary ranges?
68. How do you implement percentile calculations in SQL (if supported by RDBMS)?
69. How do you get the first and last records per group using aggregation?
70. How do you group by calculated rankings or classifications?
71. How do you filter out small groups based on row counts?
72. How do you find customers who placed only one order using aggregation?
73. How do you combine `GROUP BY` with `UNION`?
74. How do you group by a CASE expression result?
75. How do you show zero-counts for groups that don't exist in the data?
76. How do you pivot data using aggregation and `GROUP BY`?
77. How do you use conditional aggregation with `SUM(CASE...)`?

78. How do you group and sort by a calculated ratio (e.g., revenue per order)?
79. How do you find the product with the most sales in each category?
80. How do you compute aggregate metrics over a sliding time window?
81. How do you return both detailed and aggregated results in a single query?
82. How do you group by a concatenated or derived string (e.g., "Dept-Year")?
83. How do you exclude certain groups from the final result based on aggregate filters?
84. How do you track the top-N values per group using aggregation?
85. How do you group by a boolean condition (e.g., premium vs. non-premium users)?
86. How do you group by transformed column values (e.g., `TRIM()`, `LOWER()`)?
87. How do you find duplicate records using `GROUP BY`?
88. How do you apply aggregation in a recursive CTE (if supported)?
89. How do you group and get the count of unique values in a group?
90. How do you write a query that groups records and returns only the highest value row per group?
91. How do you optimize queries with heavy `GROUP BY` usage?
92. What indexing strategies help `GROUP BY` performance?
93. How does data distribution affect aggregation performance?
94. How do you handle aggregations on massive datasets (partitioning, sharding, etc.)?
95. How does the SQL optimizer treat aggregation queries?
96. How do you use `GROUP BY` in views?
97. How do you chain or nest aggregation queries?
98. Can you use aggregation in a materialized view?
99. How do you document and test complex aggregation queries?
100. How do different databases (e.g., MySQL vs PostgreSQL vs SQL Server) implement and optimize `GROUP BY`?

Let me know when you're ready for **Topic 4: SQL Subqueries & CTEs**, or if you want solutions or explanations for any of the above!



Topic 4: SQL Subqueries & CTEs,

Awesome! Here's **Topic 4: SQL Subqueries & CTEs – Part 1 (Questions 1–50):**

## SQL Interview Questions: Topic 4 – Subqueries & CTEs (Part 1 of 2)

### Subqueries (Basic to Intermediate)

1. What is a subquery in SQL?

2. What are the different types of subqueries?
3. What is the difference between a correlated and non-correlated subquery?
4. Can subqueries be used in the `SELECT` clause?
5. How do you use a subquery in the `WHERE` clause?
6. How do you use a subquery in the `FROM` clause?
7. What is a scalar subquery?
8. What happens if a scalar subquery returns more than one row?
9. How do you use subqueries to filter based on aggregated results?
10. How do you use subqueries with `IN`, `NOT IN`?
11. How do you use subqueries with `EXISTS`, `NOT EXISTS`?
12. What's the difference between `IN` and `EXISTS` in subqueries?
13. How do subqueries affect query performance?
14. How do you write a query to find employees with salaries greater than the department average (using a subquery)?
15. How do you use subqueries in `CASE` expressions?
16. Can you use `ORDER BY` in a subquery?
17. How do you limit the number of rows in a subquery?
18. How do correlated subqueries work row by row?
19. How do you use subqueries inside `UPDATE` statements?
20. How do you use subqueries inside `DELETE` statements?
21. How do you find the second highest salary using subqueries?
22. How do you write a query to find customers who placed the most orders (subquery version)?
23. Can you use aggregate functions inside subqueries?
24. Can a subquery return a table instead of a single value?
25. How can you rewrite a query using `JOIN` instead of a subquery?
26. Which is faster — JOIN or subquery — and why?
27. How do you find rows in one table that don't have a match in another using subqueries?
28. How do you use subqueries to calculate running totals?
29. How do you return top-N rows per group using subqueries?
30. What's the difference between using subqueries in `SELECT` vs `FROM`?

## CTEs (Common Table Expressions)

31. What is a CTE (Common Table Expression)?
32. How is a CTE different from a subquery?
33. How do you declare a CTE using `WITH` clause?

34. Can you use multiple CTEs in a single query?
35. Can CTEs be nested?
36. How do you chain CTEs?
37. What are the advantages of using CTEs over subqueries?
38. How do you use a CTE in combination with `JOIN`?
39. How do you write a recursive CTE?
40. What are typical use cases for recursive CTEs?
41. How do you find hierarchical relationships (e.g., org chart) using CTEs?
42. Can you use CTEs in `INSERT`, `UPDATE`, or `DELETE` statements?
43. How do you debug complex queries using CTEs?
44. Can you use aggregate functions inside a CTE?
45. How do you apply a `ROW_NUMBER()` function in a CTE?
46. Can you filter CTEs with a `WHERE` clause?
47. How do you optimize a query that uses multiple CTEs?
48. How do you use CTEs to simplify queries with repeated logic?
49. Can CTEs be used in views?
50. Are CTEs materialized or inlined by the SQL engine?

---

Let me know when you're ready for **Part 2 (Questions 51–100)** or want to move to the next topic!



Part 2 (Questions 51–100)

---

Here's **Part 2 of Topic 4: SQL Subqueries & CTEs** — questions **51–100**:

---

## SQL Interview Questions: Topic 4 – Subqueries & CTEs (Part 2 of 2)

### Advanced Subqueries

51. How do you simulate window functions using correlated subqueries?
52. How can you write a query to compare each row with the overall average using a subquery?
53. How do you identify duplicates using subqueries?
54. How do you fetch the latest record for each group using subqueries?
55. How do subqueries affect query optimization?
56. How do you rewrite a subquery to improve performance?
57. What is a lateral subquery, and when is it used?
58. How is a derived table different from a subquery in the `FROM` clause?

59. How do you filter groups based on subquery logic?
  60. How do you pass parameters into subqueries in stored procedures?
  61. How do correlated subqueries behave differently from joins?
  62. How do you combine a subquery with `GROUP BY`?
  63. How do you avoid `NOT IN` subquery pitfalls with NULL values?
  64. How does a subquery interact with indexes?
  65. How do you structure subqueries for clarity and readability?
  66. How do you use subqueries to find employees who earn more than their managers?
  67. How do you handle errors when subqueries return multiple rows in a scalar context?
  68. Can you alias a subquery result?
  69. How do subqueries behave in different RDBMSs (MySQL, PostgreSQL, SQL Server)?
  70. How can subqueries be combined with window functions?
- 

## Advanced CTEs & Recursive CTEs

71. How do recursive CTEs work internally?
  72. How do you limit recursion in a recursive CTE?
  73. How do you find the factorial of a number using recursive CTEs?
  74. How do you generate a series of numbers using a CTE?
  75. How do you flatten hierarchical data using a recursive CTE?
  76. How do you prevent infinite loops in recursive CTEs?
  77. How do you use the `SEARCH` and `CYCLE` clauses in recursive CTEs (in supported RDBMSs)?
  78. Can you use `ROW_NUMBER()` with a recursive CTE?
  79. How do you find parent-child relationships in a table using recursive CTEs?
  80. How do you calculate the depth of a tree using a recursive CTE?
  81. How do you combine multiple recursive paths into one CTE result?
  82. Can recursive CTEs return duplicate values? How do you handle them?
  83. What are performance considerations with large recursive CTEs?
  84. How do recursive CTEs compare to hierarchical queries in Oracle?
  85. Can you use CTEs to replace temporary tables?
  86. How do you use multiple CTEs to build a reporting pipeline in a single query?
  87. What's the difference between temporary views and CTEs?
  88. How do you use CTEs for pivoting or unpivoting data?
  89. Can you perform DML operations (INSERT, UPDATE, DELETE) on a CTE?
  90. What are limitations of recursive CTEs in different RDBMSs?
-

## Best Practices, Optimization, and Use Cases

91. When should you use a subquery vs. a CTE?
92. When should you avoid using a subquery?
93. When should you avoid using a CTE?
94. How do you use CTEs to break down complex business logic?
95. How do you test and debug queries using CTEs step-by-step?
96. What are the best naming conventions for CTEs?
97. How do you document queries that use multiple subqueries and CTEs?
98. How do query planners optimize CTEs vs subqueries?
99. Can a CTE or subquery be reused in the same query?
100. What's the impact of query execution order on subqueries and CTEs?

---

Let me know when you're ready for **Topic 5: SQL Window Functions**, or if you'd like detailed solutions/examples for any of these!



Topic 5: SQL Window Functions dont dived parts give me at a time

Perfect! Here's the full **Topic 5: SQL Window Functions (100 Questions)** in one go – no parts, just straight through:

---

## 📄 SQL Interview Questions: Topic 5 – Window Functions (100 Questions)

---

### ♦ Basics & Definitions

1. What is a window function in SQL?
  2. How is a window function different from an aggregate function?
  3. What is the `OVER()` clause used for?
  4. What is a window frame?
  5. Can you use `WHERE` with window functions?
  6. What are some common window functions in SQL?
  7. What is the default window frame in SQL?
  8. What happens if you omit `PARTITION BY` in a window function?
  9. What's the difference between `PARTITION BY` and `GROUP BY`?
  10. Can you use window functions in the `SELECT` clause?
- 

### ♦ Ranking Functions

11. What does the `ROW_NUMBER()` function do?
  12. How is `RANK()` different from `ROW_NUMBER()`?
  13. What does the `DENSE_RANK()` function do?
  14. When would you use `NTILE(n)`?
  15. What's the difference between `RANK()` and `DENSE_RANK()` in case of ties?
  16. How do you assign row numbers per group using `ROW_NUMBER()`?
  17. How can you get the top N records per category using `ROW_NUMBER()`?
  18. What happens when you don't `ORDER BY` in a ranking function?
  19. Can ranking functions return duplicate values?
  20. How do you remove duplicate rows but keep one using window functions?
- 

#### ◆ Value-based Functions

21. What does the `LAG()` function do?
  22. What does the `LEAD()` function do?
  23. What is the purpose of `FIRST_VALUE()`?
  24. What is the use of `LAST_VALUE()`?
  25. How do `LAG()` and `LEAD()` handle NULLs?
  26. How can you provide a default value to `LAG()` and `LEAD()`?
  27. How do you calculate the difference between current and previous rows?
  28. How can you access the 3rd next value using `LEAD()`?
  29. How can you calculate the moving average using window functions?
  30. How does `FIRST_VALUE()` behave with changing window frames?
- 

#### ◆ Aggregates over Windows

31. Can you use `SUM()` as a window function?
  32. How do you use `AVG()` over a partitioned window?
  33. How do you calculate running totals in SQL?
  34. How do you compute a cumulative percentage?
  35. How do you calculate rolling averages?
  36. How do you reset running totals when a partition changes?
  37. How do you find the max value per window frame?
  38. How do you include current row in running total or not?
  39. How does `ROWS BETWEEN` vs `RANGE BETWEEN` work?
  40. What's the default window frame for aggregate functions?
-



## ◆ Frame Clauses

41. What does `ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW` mean?
  42. What does `RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW` mean?
  43. How do you specify a frame that looks ahead?
  44. What does `CURRENT ROW` refer to in a frame?
  45. What happens when you use `RANGE BETWEEN` on non-numeric values?
  46. How do frame clauses affect `FIRST_VALUE()` and `LAST_VALUE()`?
  47. What's the difference between `ROWS` and `RANGE` when sorting on `ORDER BY id`?
  48. How can you define custom frames using `ROWS BETWEEN`?
  49. Can you mix window frames with ranking functions?
  50. What's the performance impact of large window frames?
- 

## ◆ Practical Applications

51. How do you calculate the day-over-day change in revenue?
  52. How do you compare each employee's salary to the department average using window functions?
  53. How do you compute percent of total using window functions?
  54. How do you create a rank column without gaps in ranking?
  55. How do you filter to only the top-ranked record per partition?
  56. How do you find the first and last order per customer?
  57. How do you find customers whose purchase amount increased month over month?
  58. How do you write a query to show current vs. previous row values side-by-side?
  59. How do you build a leaderboard query using window functions?
  60. How do you generate cumulative revenue grouped by category?
- 

## ◆ Advanced Patterns

61. How do you detect gaps in a sequence using `LAG()` or `LEAD()`?
62. How do you calculate the difference between first and last values in a group?
63. How do you find streaks or consecutive values using window functions?
64. How can you use window functions to deduplicate data?
65. How do you rank values within a date range (rolling window)?
66. How do you pivot data using window functions?
67. How do you calculate conversion funnels (step-by-step drop-off)?
68. How do you compare metrics across time periods (week-over-week, year-over-year)?
69. How do you detect changes or churn between states using `LAG()`?

70. How do you find the last non-null value in a partition?

---

### ♦ Window Functions + Joins, CTEs, Subqueries

- 71. Can window functions be used in a CTE?
  - 72. Can window functions be used inside a subquery?
  - 73. How do you use a window function with a `JOIN`ed table?
  - 74. How do you use `ROW_NUMBER()` in a subquery to filter data?
  - 75. Can you use window functions in combination with `HAVING`?
  - 76. How do you apply multiple window functions in one query?
  - 77. Can you nest window functions inside each other?
  - 78. How do you use a window function to get the earliest record per group via subquery?
  - 79. How do you use window functions with filters on aggregate values?
  - 80. How can you simulate window functions in databases that don't support them?
- 

### ♦ Window Function Syntax & Behavior

- 81. What happens if `ORDER BY` is missing in a window function?
  - 82. Can you alias the result of a window function?
  - 83. Can you reuse a window definition with `WINDOW` clause?
  - 84. What's the difference between inline vs. named windows?
  - 85. How do you order by multiple columns in `OVER()`?
  - 86. How do you use a different window frame for each function in a query?
  - 87. Can you write a `CASE WHEN` inside a window function?
  - 88. How do you assign percentile rankings with window functions?
  - 89. Can you use window functions inside `UPDATE` or `DELETE`?
  - 90. What's the result if your partition is empty?
- 

### ♦ Performance, Optimization, and Differences

- 91. What are the performance considerations when using window functions?
- 92. How can indexes affect window function performance?
- 93. When should you use window functions vs. subqueries or joins?
- 94. What's the impact of memory usage with large window frames?
- 95. How do different RDBMSs (MySQL, PostgreSQL, SQL Server) handle window functions?
- 96. Can you parallelize queries with window functions?
- 97. What are the limitations of window functions in your database engine?

98. How do you test and debug queries with window functions?
99. Can window functions be used in views or materialized views?
100. How would you explain window functions to someone new to SQL?

Let me know when you're ready for **Topic 6: SQL Set Operations (UNION, INTERSECT, etc.)**, or if you'd like any examples or answers!



## Topic 6: SQL Set Operations (UNION, INTERSECT, etc.)

Here's **Topic 6: SQL Set Operations (UNION, INTERSECT, etc.) – 100 Interview Questions** in one go:

### SQL Interview Questions: Topic 6 – Set Operations (100 Questions)

#### ◆ Basic Understanding

1. What are set operations in SQL?
2. Name all major SQL set operations.
3. What is the purpose of `UNION` in SQL?
4. How is `UNION ALL` different from `UNION`?
5. What does `INTERSECT` do?
6. What does `EXCEPT` (or `MINUS`) do?
7. What are the differences between `EXCEPT` and `NOT IN`?
8. When would you use `UNION` instead of `JOIN`?
9. Do set operations remove duplicates by default?
10. Which set operation is most performant?

#### ◆ UNION & UNION ALL

11. What are the requirements for using `UNION` between two tables?
12. What happens if the number of columns doesn't match in `UNION`?
13. Does `UNION` preserve ordering?
14. How do you sort results from a `UNION` query?
15. Can you use column aliases in `UNION` queries?
16. What are some real-world use cases for `UNION`?
17. How do NULLs behave in `UNION` queries?
18. How does `UNION ALL` improve performance?
19. When would you choose `UNION ALL` over `UNION`?

20. Can `UNION` be used on different data types?
  21. What happens when using `UNION` with `DISTINCT`?
  22. What's the impact of `ORDER BY` in the first query of a `UNION`?
  23. How do you `UNION` results from more than two queries?
  24. Can you filter individual parts of a `UNION` query separately?
  25. How do you write a query that combines customer data from two systems using `UNION`?
  26. What does the error "number of columns must be the same" mean in `UNION`?
  27. How do you remove duplicates across multiple tables using `UNION`?
  28. How do you handle column mismatches when writing a `UNION`?
  29. Can you perform `JOIN` operations before applying `UNION`?
  30. Can you use `GROUP BY` within individual queries in a `UNION`?
- 

#### ♦ INTERSECT

31. What is the purpose of `INTERSECT` in SQL?
  32. Which databases support `INTERSECT`?
  33. How does `INTERSECT` treat NULL values?
  34. Can `INTERSECT` be used with `ORDER BY`?
  35. How does `INTERSECT` handle duplicates?
  36. What happens if column order is different in two `INTERSECT` queries?
  37. What's the difference between `INTERSECT` and `INNER JOIN`?
  38. How do you use `INTERSECT` to find common employees in two tables?
  39. How do you simulate `INTERSECT` in MySQL (which doesn't support it directly)?
  40. Can `INTERSECT` be combined with `GROUP BY`?
  41. What is the performance impact of using `INTERSECT`?
  42. How do you compare results from two different subqueries using `INTERSECT`?
  43. What data type compatibility rules apply for `INTERSECT`?
  44. Can `INTERSECT` work with expressions (like `SELECT 1`)?
  45. How can you debug unexpected results from an `INTERSECT` query?
- 

#### ♦ EXCEPT / MINUS

46. What is the purpose of `EXCEPT` (or `MINUS`)?
47. What is the difference between `EXCEPT` and `NOT EXISTS`?
48. How does `EXCEPT` handle NULL values?
49. Which databases support `EXCEPT` and `MINUS`?

50. Can `EXCEPT` be used to compare subqueries?
  51. How do you find records in one table but not in another using `EXCEPT`?
  52. What happens when columns don't match in an `EXCEPT` query?
  53. How do you simulate `EXCEPT` in MySQL?
  54. Can you use `EXCEPT` to clean data?
  55. What happens if `EXCEPT` queries use different column names?
  56. Can you use `GROUP BY` inside an `EXCEPT` query?
  57. Can `EXCEPT` be used in conjunction with `JOIN`?
  58. What's the difference between `EXCEPT` and `LEFT JOIN WHERE IS NULL`?
  59. When should you use `EXCEPT` instead of `NOT IN`?
  60. What's the performance cost of `EXCEPT`?
- 

#### ♦ Combining Set Operations

61. Can you use `UNION`, `INTERSECT`, and `EXCEPT` in one query?
  62. How do you control precedence when combining set operations?
  63. What are the limitations of combining set operations?
  64. Can set operations be nested?
  65. How do parentheses affect set operation logic?
  66. How do you combine results from multiple APIs (as tables) using set operations?
  67. How do set operations interact with `DISTINCT` and `GROUP BY`?
  68. How do you find customers present in one country but not in another?
  69. Can you perform aggregate calculations after a set operation?
  70. How do you alias columns in multi-set queries?
- 

#### ♦ Performance & Optimization

71. Which set operation is the most performant?
72. How do indexes impact set operation queries?
73. How can you reduce duplication in `UNION` queries?
74. How do temporary tables affect set operation performance?
75. What query plans are typically used for set operations?
76. Can set operations be parallelized?
77. What's more performant: `EXCEPT` or `NOT EXISTS`?
78. What are common bottlenecks when using `UNION` with large datasets?
79. Can materialized views speed up set operations?

80. How do execution plans differ between `UNION` and `JOIN`?

### ◆ Advanced Use Cases

- 81. How do you combine rows from multiple years into one result set using `UNION`?
- 82. How do you find overlapping data using `INTERSECT`?
- 83. How do you eliminate overlapping data using `EXCEPT`?
- 84. How do you simulate `INTERSECT` and `EXCEPT` in databases that lack them?
- 85. How do you combine customer datasets from different platforms using `UNION`?
- 86. How do you generate a report combining different sources of transactions with `UNION`?
- 87. How do you create a differential report using `EXCEPT`?
- 88. How do you find common IDs in two logs using `INTERSECT`?
- 89. How do you find items that changed between snapshots using set operations?
- 90. How do set operations help in data warehousing?

### ◆ Debugging & Pitfalls


- 91. What causes the "number of columns doesn't match" error in set operations?
- 92. Why does `UNION` remove rows unexpectedly?
- 93. How do NULLs affect set operation logic?
- 94. How do you find hidden duplicates in `UNION ALL`?
- 95. How do you trace performance issues in large `UNION` queries?
- 96. How do data types impact result consistency in set operations?
- 97. What are common mistakes when writing `INTERSECT` queries?
- 98. What are edge cases with empty sets in `EXCEPT`?
- 99. How do you verify that set operation queries return correct results?
- 100. How do you write test cases to validate set operation logic?

Let me know when you're ready for **Topic 7: SQL Constraints & Keys**, or if you'd like detailed solutions or explanations for any of these!



Topic 7: SQL Constraints & Keys,

Here's **Topic 7: SQL Constraints & Keys – 100 Interview Questions** in one go:

 **SQL Interview Questions: Topic 7 – Constraints & Keys (100 Questions)**

## ◆ Basic Concepts

1. What is a constraint in SQL?
  2. Name the types of constraints in SQL.
  3. What is a primary key?
  4. What is a foreign key?
  5. What is a unique constraint?
  6. What is a check constraint?
  7. What is a default constraint?
  8. What is a `NOT NULL` constraint?
  9. What is a composite key?
  10. Can a table have multiple primary keys?
- 

## ◆ Primary Key

11. What is the purpose of a primary key?
  12. Can a primary key contain NULL values?
  13. How do you define a primary key on a table?
  14. Can a primary key be modified after creation?
  15. How do you define a composite primary key?
  16. What's the difference between `PRIMARY KEY` and `UNIQUE NOT NULL`?
  17. Can a table have a primary key and a unique constraint on the same column?
  18. Can you drop and recreate a primary key?
  19. How do primary keys relate to indexing?
  20. What's the default behavior when inserting duplicate primary key values?
- 

## ◆ Foreign Key

21. What does a foreign key do in SQL?
22. Can a foreign key reference a non-primary key column?
23. What is referential integrity?
24. What happens if a foreign key is violated?
25. Can a foreign key be NULL?
26. How do you define a foreign key during table creation?
27. How do you add a foreign key after table creation?
28. What are `ON DELETE CASCADE` and `ON UPDATE CASCADE`?
29. What's the difference between `RESTRICT`, `CASCADE`, and `SET NULL`?

- 30. Can a table have multiple foreign keys?
  - 31. What happens when the parent row is deleted and foreign key constraint exists?
  - 32. How do you check existing foreign key constraints on a table?
  - 33. How can you disable and re-enable a foreign key constraint?
  - 34. What are circular references in foreign keys?
  - 35. Can a foreign key reference multiple columns?
- 

#### ◆ **Unique Constraints**

- 36. What is the purpose of a `UNIQUE` constraint?
  - 37. Can a unique column have NULL values?
  - 38. How does `UNIQUE` differ from `PRIMARY KEY`?
  - 39. Can you define a multi-column `UNIQUE` constraint?
  - 40. Can a table have more than one `UNIQUE` constraint?
  - 41. Can you apply `UNIQUE` to existing columns?
  - 42. What error will you get when violating a `UNIQUE` constraint?
  - 43. Can `UNIQUE` constraints be used with foreign keys?
  - 44. What's the impact of a `UNIQUE` constraint on performance?
  - 45. How do `UNIQUE` constraints affect data modeling?
- 

#### ◆ **Check Constraints**

- 46. What is a `CHECK` constraint used for?
  - 47. Can `CHECK` constraints reference other tables?
  - 48. What's an example of a `CHECK` constraint?
  - 49. How do you modify an existing `CHECK` constraint?
  - 50. What happens if a row violates a `CHECK` constraint?
  - 51. How do `CHECK` constraints handle NULL values?
  - 52. Can you create complex conditions with `CHECK`?
  - 53. How do you drop a `CHECK` constraint?
  - 54. Can you have multiple `CHECK` constraints on a column?
  - 55. Can you enforce business rules with `CHECK` constraints?
- 

#### ◆ **Default Constraints**

- 56. What is a default constraint in SQL?
- 57. How do you define a default value for a column?



- 58. Can a column have both `DEFAULT` and `NOT NULL`?
  - 59. What's the behavior if no value is passed for a column with a default?
  - 60. Can you change the default value of a column?
  - 61. What data types support default values?
  - 62. Can the default be a function (e.g., `GETDATE()`)?
  - 63. What's the behavior of `DEFAULT` in an `INSERT INTO ... SELECT`?
  - 64. Can `DEFAULT` and `CHECK` be combined?
  - 65. How do you remove a default constraint?
- 

#### ♦ **NOT NULL Constraints**

- 66. What does `NOT NULL` mean?
  - 67. Can a primary key column be null?
  - 68. How do you add `NOT NULL` to an existing column?
  - 69. What error do you get if inserting NULL into a `NOT NULL` column?
  - 70. Can you remove a `NOT NULL` constraint?
- 

#### ♦ **Constraint Management & Metadata**

- 71. How do you list constraints on a table?
  - 72. How do you check if a constraint exists?
  - 73. How do you name constraints explicitly?
  - 74. Can you rename a constraint?
  - 75. Can you disable a constraint temporarily?
  - 76. How are constraints stored in the system catalog?
  - 77. How do constraints affect query performance?
  - 78. What's the effect of constraints on `INSERT` performance?
  - 79. Can you copy constraints from one table to another?
  - 80. How do you export/import constraints in backup/restore processes?
- 

#### ♦ **Composite & Multi-Column Constraints**

- 81. What is a composite key?
- 82. Can `UNIQUE` constraints span multiple columns?
- 83. What's the difference between single-column and multi-column constraints?
- 84. How do you enforce that a combination of values is unique?
- 85. Can a composite foreign key reference a composite primary key?

86. How do you define a composite key during table creation?
87. What's the impact of multi-column constraints on indexing?
88. Can you apply `CHECK` constraints to multiple columns at once?
89. How do you manage composite constraints in normalization?
90. How do you write queries against composite key constraints?

#### ♦ Advanced Topics & Scenarios

91. Can a constraint enforce case sensitivity?
92. Can constraints be applied to views?
93. What's the difference between declarative constraints and triggers?
94. How do constraints relate to data normalization?
95. Can you enforce constraints conditionally?
96. What are deferrable constraints?
97. What's the difference between `DEFERRABLE INITIALLY DEFERRED` and `IMMEDIATE`?
98. Can you apply constraints in temporary tables?
99. What's the role of constraints in transactional integrity?
100. How would you explain constraints to a non-technical person?

Let me know when you're ready for **Topic 8: SQL Indexes & Performance Tuning** or want explanations or examples for any of the questions above!



#### Topic 8: SQL Indexes & Performance Tuning

Here's **Topic 8: SQL Indexes & Performance Tuning – 100 Interview Questions** in one go:

### SQL Interview Questions: Topic 8 – Indexes & Performance Tuning (100 Questions)

#### ♦ Index Basics

1. What is an index in SQL?
2. How do indexes improve query performance?
3. What are the types of indexes in SQL?
4. What is the difference between clustered and non-clustered indexes?
5. Can a table have multiple clustered indexes?
6. What is a composite index?

7. What is a unique index?
  8. What is a full-text index?
  9. What is a bitmap index?
  10. What are spatial indexes used for?
- 

#### ♦ **Creating & Managing Indexes**

11. How do you create an index in SQL?
  12. How do you create a unique index?
  13. How do you create a composite index on multiple columns?
  14. How do you drop an index?
  15. Can you rename an index?
  16. How do you check if an index exists on a table?
  17. How do you list all indexes on a table?
  18. What is the syntax to create an index with a specific name?
  19. How do you create an index on a view?
  20. Can indexes be created on temporary tables?
- 

#### ♦ **Indexing Strategy**

21. When should you use an index?
  22. What columns are good candidates for indexing?
  23. When should you **not** use an index?
  24. What is index selectivity?
  25. How does low selectivity affect index performance?
  26. How do composite indexes work with queries that use only some columns?
  27. What is a covering index?
  28. How do you decide between single-column and composite indexes?
  29. What are filtered indexes?
  30. How do indexes interact with joins?
- 

#### ♦ **Clustered Indexes**

31. What is the clustered index structure?
32. How many clustered indexes can a table have?
33. What is the default clustered index in most databases?
34. How do clustered indexes affect row storage?

35. What are the pros and cons of using a clustered index?
  36. Can you change the clustered index of a table?
  37. What happens when you drop a clustered index?
  38. What is the impact of clustered indexes on insert and update operations?
  39. How does clustering affect range queries?
  40. Can clustered indexes be created on VARCHAR columns?
- 

#### ◆ **Non-Clustered Indexes**

41. What is a non-clustered index?
  42. How is a non-clustered index stored?
  43. How does a non-clustered index affect data retrieval?
  44. Can a table have both clustered and non-clustered indexes?
  45. What is a bookmark lookup?
- 

#### ◆ **Performance Tuning Concepts**

46. What is query optimization?
  47. What is a query execution plan?
  48. How do you view the execution plan in SQL Server/PostgreSQL/MySQL?
  49. What are the key components of an execution plan?
  50. What is a full table scan?
  51. What is an index scan vs. index seek?
  52. What is a table scan, and when does it happen?
  53. How do you avoid table scans?
  54. What is a slow query log?
  55. What are the most common reasons for slow queries?
- 

#### ◆ **Tuning Techniques**

56. How can indexes improve SELECT performance?
57. Can indexes slow down INSERT/UPDATE/DELETE operations?
58. What's the trade-off between read and write performance with indexes?
59. How do you identify missing indexes?
60. What tools are available to analyze query performance?
61. How do you reduce the cost of JOIN operations?
62. What is query caching?

- 63. How does normalization impact performance?
  - 64. When is denormalization helpful for performance?
  - 65. What are materialized views?
- 

#### ♦ **Advanced Indexing Concepts**

- 66. What is an index hint?
  - 67. When would you use index hints?
  - 68. What is an invisible index?
  - 69. What is index fragmentation?
  - 70. How do you rebuild or reorganize indexes?
  - 71. How often should indexes be rebuilt?
  - 72. How do you monitor index usage?
  - 73. What's the difference between a rebuild and a reorganize?
  - 74. What is fill factor in indexing?
  - 75. How does index compression affect performance?
- 

#### ♦ **Indexing for Specific Queries**

- 76. How do indexes help with `WHERE` clauses?
  - 77. How do indexes work with `ORDER BY` clauses?
  - 78. How do indexes help with range queries (`BETWEEN`, `<`, `>`, etc.)?
  - 79. How do indexes help with `LIKE` and wildcards?
  - 80. Why might a query not use an index even when it exists?
  - 81. How do you optimize queries involving multiple joins?
  - 82. What indexing strategies are useful for analytical queries?
  - 83. How do you speed up aggregate queries with indexes?
  - 84. What's the impact of functions on indexed columns in queries?
  - 85. How do you index columns used in WHERE and JOIN conditions?
- 

#### ♦ **Index & Database Size**

- 86. How do indexes affect disk space?
- 87. How do you calculate the size of an index?
- 88. Can too many indexes hurt performance?
- 89. How do you find unused indexes?
- 90. How do you remove redundant or duplicate indexes?

## ♦ Real-World Scenarios

91. How would you optimize a report query that takes too long?
92. What do you do when a query is fast one day and slow the next?
93. How do you find which queries are using the most resources?
94. What do you do when you see high CPU usage on the DB server?
95. How do you test index impact before deploying to production?
96. What's the best approach to indexing in a write-heavy application?
97. How do you balance OLTP and OLAP index needs?
98. How would you structure indexing in a highly normalized schema?
99. How does indexing differ across MySQL, PostgreSQL, and SQL Server?
100. What are best practices for indexing in large databases?

Let me know if you want **Topic 9: SQL Transactions & Concurrency**, or want explanations/examples for any of these!



Topic 9: SQL Transactions & Concurrency,

Here's **Topic 9: SQL Transactions & Concurrency – 100 Interview Questions** in one go:

## **SQL Interview Questions: Topic 9 – Transactions & Concurrency (100 Questions)**

### ♦ Transactions – Basics

1. What is a transaction in SQL?
2. What are the properties of a transaction (ACID)?
3. What does atomicity mean in the context of transactions?
4. What is consistency in transactions?
5. What is isolation in transactions?
6. What is durability in transactions?
7. How do you start a transaction in SQL?
8. What does `COMMIT` do?
9. What does `ROLLBACK` do?
10. What is `SAVEPOINT` used for?

### ♦ Working with Transactions

11. How do you rollback to a savepoint?
  12. What happens if you forget to commit a transaction?
  13. Can you nest transactions?
  14. What is autocommit mode?
  15. How do you disable autocommit?
  16. How do you check if a transaction is active?
  17. What happens if there's a crash during a transaction?
  18. What is the difference between implicit and explicit transactions?
  19. Can a transaction span multiple SQL statements?
  20. How do transactions behave in stored procedures?
- 

#### ◆ Isolation Levels – Concepts

21. What are isolation levels in SQL?
  22. Name the four standard isolation levels.
  23. What is `READ UNCOMMITTED`?
  24. What is `READ COMMITTED`?
  25. What is `REPEATABLE READ`?
  26. What is `SERIALIZABLE` isolation?
  27. What is the default isolation level in PostgreSQL?
  28. What is the default isolation level in MySQL?
  29. What is the default in SQL Server?
  30. How do isolation levels affect locking?
- 

#### ◆ Isolation Levels – Effects

31. What is a dirty read?
  32. What is a non-repeatable read?
  33. What is a phantom read?
  34. Which isolation level prevents dirty reads?
  35. Which isolation level prevents phantom reads?
  36. What isolation level guarantees full consistency?
  37. Which level provides the best performance?
  38. What's the trade-off of using `SERIALIZABLE`?
  39. Can isolation levels be set per transaction?
  40. How do isolation levels affect concurrency?
-

## ♦ Locks – Basics

41. What is a lock in SQL?
  42. What are shared and exclusive locks?
  43. What is a table-level lock?
  44. What is a row-level lock?
  45. What is an intent lock?
  46. What is a deadlock?
  47. What is a blocking transaction?
  48. How do you detect a deadlock?
  49. How can you avoid deadlocks?
  50. What happens when a deadlock is detected?
- 

## ♦ Locks – Management

51. How do you view active locks in SQL Server/PostgreSQL?
  52. What is lock escalation?
  53. How does lock escalation impact performance?
  54. How can you control locking behavior in SQL?
  55. What is optimistic locking?
  56. What is pessimistic locking?
  57. What's the difference between optimistic and pessimistic locking?
  58. How do you implement optimistic locking with version columns?
  59. What are row-versioning-based isolation levels?
  60. What is MVCC (Multi-Version Concurrency Control)?
- 

## ♦ MVCC (Multi-Version Concurrency Control)

61. How does MVCC work?
  62. Which databases use MVCC (e.g., PostgreSQL)?
  63. What are the benefits of MVCC?
  64. How does MVCC handle concurrent writes?
  65. Does MVCC eliminate the need for locks?
- 

## ♦ Concurrency – Basics

66. What is concurrency in SQL databases?
67. Why is concurrency control important?



- 68. What issues arise without proper concurrency control?
  - 69. What is the lost update problem?
  - 70. How do isolation levels relate to concurrency?
- 

#### ◆ **Concurrency – Control Techniques**

- 71. What is a transaction log?
  - 72. How do write-ahead logs (WAL) support transactions?
  - 73. What is snapshot isolation?
  - 74. What is two-phase locking?
  - 75. How does SQL Server handle concurrency differently from PostgreSQL?
  - 76. What is serialization anomaly?
  - 77. How do databases prevent serialization anomalies?
  - 78. What is a concurrency-safe update query?
  - 79. What is row-level versioning?
  - 80. How can you minimize lock contention?
- 

#### ◆ **Concurrency in Practice**

- 81. What happens when two users update the same row simultaneously?
  - 82. How do you prevent update conflicts?
  - 83. What is the best isolation level for analytics workloads?
  - 84. How do different isolation levels impact throughput?
  - 85. How do transactions affect connection pooling?
- 

#### ◆ **Real-World Transaction Scenarios**

- 86. How do transactions work in financial applications?
  - 87. How would you handle a failed payment rollback?
  - 88. What is transactional integrity?
  - 89. How do you handle partial commits?
  - 90. What is compensating transaction?
- 

#### ◆ **Testing & Debugging Transactions**

- 91. How can you test transaction behavior?
- 92. How do you simulate concurrency issues in development?
- 93. What are tools to trace transaction performance?

94. How do you log transaction activity?
95. How do you roll back a transaction after a stored procedure fails?

### ◆ Best Practices

96. When should you commit a transaction?
97. How can you minimize transaction duration?
98. Why is it important to close transactions explicitly?
99. What happens if a transaction remains open too long?
100. What are best practices for handling transactions in high-concurrency environments?

Let me know when you're ready for **Topic 10: SQL Views, Stored Procedures & Functions** or want explanations, examples, or practice problems for this topic!



### Topic 10: SQL Views, Stored Procedures & Functions

Here's **Topic 10: SQL Views, Stored Procedures & Functions – 100 Interview Questions** all at once:

## SQL Interview Questions: Topic 10 – Views, Stored Procedures & Functions (100 Questions)

### ◆ Views – Basics

1. What is a SQL view?
2. How do you create a view?
3. What is the syntax for creating a view with `AS SELECT`?
4. What are the benefits of using views?
5. Can views simplify complex queries?
6. What is an updatable view?
7. When is a view **not** updatable?
8. What's the difference between a view and a table?
9. Can you use a `JOIN` in a view definition?
10. Can you use `GROUP BY` in a view?

### ◆ Views – Advanced Concepts

11. What is a materialized view?

12. How is a materialized view different from a regular view?
  13. How do you refresh a materialized view?
  14. What are indexed views?
  15. Can you filter data in a view using `WHERE`?
  16. How do you drop a view?
  17. Can you alter a view?
  18. How do you check the definition of a view?
  19. What happens if a base table changes in a view?
  20. Can you have nested views?
- 

#### ◆ **Stored Procedures – Basics**

21. What is a stored procedure?
  22. What are the benefits of using stored procedures?
  23. How do you create a stored procedure?
  24. What are input parameters in stored procedures?
  25. What are output parameters in stored procedures?
  26. How do you call a stored procedure in SQL?
  27. Can you return multiple result sets from a stored procedure?
  28. Can a stored procedure return a value?
  29. What is the difference between a stored procedure and a function?
  30. Can stored procedures call other procedures?
- 

#### ◆ **Stored Procedures – Control Flow**

31. What control statements can be used in procedures (`IF`, `WHILE`, etc.)?
  32. How do you use `IF...ELSE` in a stored procedure?
  33. How do you use `CASE` inside a stored procedure?
  34. How do you handle exceptions in stored procedures?
  35. What is `TRY...CATCH` in SQL Server?
  36. What is `EXCEPTION` block in PL/pgSQL?
  37. How do you use a loop in a stored procedure?
  38. How do you log errors from stored procedures?
  39. Can you use cursors in stored procedures?
  40. What are the drawbacks of using cursors?
-

## ◆ **Stored Procedures – Security & Management**

41. How do you grant execution permission on a stored procedure?
  42. What is the difference between `WITH ENCRYPTION` and normal procedure?
  43. Can stored procedures be scheduled?
  44. How do you debug a stored procedure?
  45. How do you alter a stored procedure?
  46. How do you drop a stored procedure?
  47. Can you have temporary stored procedures?
  48. How can stored procedures be version-controlled?
  49. How do stored procedures impact performance?
  50. What are best practices for writing stored procedures?
- 

## ◆ **User-Defined Functions (UDFs) – Basics**

51. What is a SQL function?
  52. How do you create a scalar function?
  53. What's the difference between scalar and table-valued functions?
  54. How do you return a table from a function?
  55. What's the syntax for an inline table-valued function?
  56. What's the difference between deterministic and non-deterministic functions?
  57. What is a system function vs. a user-defined function?
  58. Can you call a function in a `SELECT` clause?
  59. Can functions be used in constraints?
  60. Can you use DML operations (`INSERT`, `UPDATE`, `DELETE`) in functions?
- 

## ◆ **Functions – Use Cases & Management**

61. When should you use a function vs. a stored procedure?
62. Can functions return multiple values?
63. How do you pass default arguments in functions?
64. Can you overload functions in SQL?
65. How do you drop a function?
66. How do you modify a function?
67. Can you nest functions?
68. How do you handle exceptions inside a function?
69. Can you use transactions inside a function?

70. How do you test a SQL function?

---

◆ **Performance & Optimization**

- 71. Do views improve performance?
  - 72. When should you avoid using views?
  - 73. How do stored procedures affect query plans?
  - 74. Can stored procedures be cached?
  - 75. How do indexes interact with views and functions?
  - 76. Can functions be inlined into queries?
  - 77. Do scalar functions slow down performance in large queries?
  - 78. How do you profile a stored procedure's performance?
  - 79. How do you refactor a large stored procedure?
  - 80. How does using views or functions affect maintainability?
- 

◆ **Real-World Scenarios & Best Practices**

- 81. How would you use a view in a reporting dashboard?
  - 82. How would you use a stored procedure for data validation?
  - 83. What are some common anti-patterns with stored procedures?
  - 84. How do you track changes to a procedure?
  - 85. How would you secure sensitive logic in a stored procedure?
  - 86. What is the role of views/functions in a multi-tenant system?
  - 87. How would you build reusable functions for date calculations?
  - 88. How would you manage stored procedures in version-controlled deployments?
  - 89. What are the risks of excessive abstraction with views/functions?
  - 90. How would you document a large set of views and procedures?
- 

◆ **Cross-Platform & Vendor-Specific Features**

- 91. How do stored procedures differ between SQL Server and PostgreSQL?
- 92. How do views work differently in MySQL vs. PostgreSQL?
- 93. Can you write functions in languages like PL/pgSQL or T-SQL?
- 94. How do you execute dynamic SQL inside procedures/functions?
- 95. What is the equivalent of a stored procedure in MySQL?
- 96. What are Oracle's packages and how are they different from standalone procedures?
- 97. How does SQL Server handle CLR functions?

98. Can you integrate external logic (Python/Java) in stored procedures?
99. What are limitations of stored procedures across different RDBMSs?
100. What are best practices for writing portable SQL procedures/functions?

Let me know if you'd like **examples**, **practice problems**, or **explanations** for any of these!



examples, practice problems, or explanations

Awesome! Since you'd like **examples, practice problems, and explanations** for **SQL Views, Stored Procedures & Functions**, I'll break it down by category for clarity.

## 1. SQL Views

### Explanation:

A *view* is a virtual table based on the result of a SQL query. It does **not store data**; it simply shows data from other tables.

### Example:

```
sql

`CREATE VIEW active_customers AS
SELECT customer_id, name, email
FROM customers
WHERE status = 'active';`
```

Now you can query the view like a table:

```
sql

`SELECT * FROM active_customers;`
```

### Practice Problem:

Create a view called `high_salary_employees` that displays `id`, `name`, and `salary` from an `employees` table where salary is over 100000.

## 2. Stored Procedures

### Explanation:

A *stored procedure* is a block of SQL that can be executed multiple times. It can take input/output parameters, and include control logic like `IF`, `WHILE`, etc.

## 💡 Example (SQL Server):

```
sql

`CREATE PROCEDURE GetEmployeeById
    @emp_id INT
AS
BEGIN
    SELECT * FROM employees WHERE id = @emp_id;
END;`
```

To call it:

```
sql

`EXEC GetEmployeeById @emp_id = 101;`
```

## 🧠 Practice Problem:

Write a stored procedure `GetOrdersByCustomer` that takes a customer ID and returns all orders from an `orders` table for that customer.

---

## 🔄 3. User-Defined Functions (UDFs)

### ✅ Explanation:

A *function* returns a value and can be used inside SQL expressions. Functions can return scalars or tables.

## 💡 Example (Scalar Function – SQL Server):

```
sql

`CREATE FUNCTION GetFullName(@first NVARCHAR(50), @last NVARCHAR(50))
RETURNS NVARCHAR(100)
AS
BEGIN
    RETURN @first + ' ' + @last;
END;`
```

Usage:

```
sql

`SELECT dbo.GetFullName('Jane', 'Doe');`
```

## 🧠 Practice Problem:

Create a scalar function `GetAge` that takes a birthdate as input and returns the age in years.

---

## Mini-Project Challenge

Imagine you have these tables:

- ``employees(id, name, department_id, salary)``
- ``departments(id, name)``
- ``projects(id, name, budget)``
- ``employee_projects(employee_id, project_id)``

Try the following:

1. Create a **view** that shows all employees with their department names.
2. Write a **stored procedure** that returns all projects assigned to a given employee.
3. Write a **function** that returns the total budget of projects an employee is assigned to.

---

Would you like me to give you **solutions** to any of the practice problems or mini-project steps above? Or would you prefer I quiz you and review your answers?