71 SQL Queries Interview Questions to Ask Candidates

Questions

1. Can you explain the difference between INNER JOIN and OUTER JOIN?
2. How would you find duplicate entries in a database table?
3. What is the purpose of the GROUP BY clause in SQL?
4. How do you optimize a slow-running query?
5. What are the differences between SQL and NoSQL databases?
6. Can you describe what a subquery is and provide an example?
7. How would you handle NULL values in SQL?
8. What are ACID properties in a database transaction, and why are they important?
9. How do you create an index in SQL, and why would you use one?
10. Can you explain the difference between DELETE and TRUNCATE commands?
11. Can you explain the difference between a primary key and a foreign key?
12. What is normalization, and why is it important in database design?
13. How would you handle a situation where a database is running out of space?
14. What steps would you take to back up a database?
15. Can you describe what a view is and its purpose in SQL?
16. How would you approach troubleshooting a database connection issue?
17. What is a transaction in SQL, and why are transactions important?
18. How do you use the CASE statement in SQL?
19. What is a stored procedure, and how do you create one?
20. Explain the difference between a UNION and a UNION ALL statement.
21. How would you retrieve the top N records from a table?
22. Can you explain what a window function is and give an example?
23. Describe the difference between HAVING and WHERE clauses.
24. What are common table expressions (CTEs), and how do you use them?
25. How would you implement a recursive query in SQL?
26. Explain the concept of a cursor and provide a use case.
27. What are the different types of indexes in SQL?
28. How would you perform a full-text search in an SQL database?
29. Describe the process of database partitioning and its benefits.
30. How do you use the COALESCE function in SQL?
31. What is the difference between RANK() and DENSE_RANK() functions?
32. Explain the role of a trigger in SQL and how to create one.
33. Can you explain the concept of database sharding and its benefits?
34. How would you approach performance tuning for a heavily loaded SQL database?
35. What is data warehousing, and how does it differ from a traditional database?
36. How do you ensure data integrity in a database?
37. Can you describe the role of database normalization and its impact on performance?
38. How do you handle database migration from an on-premise setup to the cloud?
39. What strategies do you use for disaster recovery and ensuring high availability in a database system?
40. How do you ensure secure access to a database?
41. Can you explain how a LEFT JOIN works and provide an example?
42. What is a CROSS JOIN and in what scenario would you use it?
43. How do you combine data from multiple tables using a RIGHT JOIN?
44. Describe a situation where you would use a SELF JOIN.
45. Explain the concept of a FULL OUTER JOIN and when it would be useful.
46. How would you perform an EQUI JOIN and what are its benefits?
47. Can you differentiate between a NATURAL JOIN and an INNER JOIN?
48. What is an ANTI JOIN and how would you implement it in SQL?
49. How do you handle joining tables with no direct relationship?
50. Describe how to use a JOIN with a subquery.
51. What are the potential pitfalls of joining large tables and how can you mitigate them?
52. How can you optimize a query that involves multiple joins?
53. How would you identify and resolve a query that's causing excessive table scans?
54. Can you explain the concept of query plan caching and its impact on performance?
55. What strategies would you use to optimize a query with multiple joins on large tables?
56. How does proper indexing affect query performance, and what factors do you consider when creating indexes?
57. Can you describe a situation where denormalization might improve query performance?
58. What tools or techniques do you use to analyze and optimize slow-running queries?
59. How would you approach optimizing a query that uses a correlated subquery?
60. Can you explain the concept of query parallelism and how it can be leveraged for performance?
61. What are some common anti-patterns in SQL that can lead to poor query performance?
62. How do you balance the trade-offs between query optimization and maintainability in complex database systems?
63. How would you approach optimizing a query that's causing performance issues during peak traffic periods?
64. Describe a scenario where you had to recover a database after a critical failure. What steps did you take?
65. How would you design a database schema for a new application with rapidly changing data requirements?
66. Explain a time when you had to migrate a database from one SQL server to another. What challenges did you face, and how did you overcome them?
67. What methods would you use to monitor database performance and why?
68. Describe a situation where you had to troubleshoot a deadlock issue. What was your approach?
69. How would you handle a data corruption issue in a production database?
70. What strategies would you implement to ensure data security in a database system?