101 MySQL interview questions that you should ask to hire top engineers

Questions

- 1. What is SQL and why do we use it?
- 2. Can you explain the difference between DELETE and TRUNCATE commands?
- 3. What are the different types of joins in SQL?
- 4. How do you sort the results of a SQL query?
- 5. What is a primary key, and why is it important?
- 6. What is a foreign key and how does it relate to other tables?
- 7. How do you filter data in SQL? Give some examples.
- 8. What is the purpose of the GROUP BY clause?
- 9. How do you use aggregate functions like COUNT, SUM, AVG, MIN, and MAX?
- 10. What is a subquery? Can you give an example of when you might use one?
- 11. Explain the difference between WHERE and HAVING clauses.
- 12. How do you combine the results of two or more SELECT statements?
- 13. What are indexes and why are they used?
- 14. How can you avoid SQL injection vulnerabilities?
- 15. Describe the concept of database normalization.
- 16. What are the different normal forms in database design?
- 17. How do you back up and restore a database?
- 18. What are transactions in SQL? Why are they important?
- 19. Explain the ACID properties of a database transaction.
- 20. What is a stored procedure?
- 21. What are views in SQL?
- 22. Explain the difference between clustered and non-clustered indexes.
- 23. How can you optimize a slow-running SQL query?
- 24. What is the purpose of the EXISTS operator in SQL?
- 25. Describe a scenario where you might use a self-join.

26. Explain the difference between CHAR and VARCHAR data types. When would you choose one over the other?

27. What is the purpose of using window functions in SQL, and can you provide an example of a scenario where they would be beneficial?

28. Describe the use of common table expressions (CTEs). How do they simplify complex queries?

29. How can you optimize a slow-running SQL query? Mention at least three techniques.

30. Explain the concept of normalization in database design and why it is important.

31. What are the different types of joins available in SQL? Describe their differences with examples.

32. How do you handle NULL values in SQL queries? Explain the IS NULL and IS NOT NULL operators.

33. Explain the difference between clustered and non-clustered indexes.

34. What are aggregate functions in SQL? Give examples and explain how to use GROUP BY clause with them.

35. Describe the purpose and usage of subqueries in SQL. Give an example scenario.

36. How can you prevent SQL injection attacks? What are parameterized queries?

37. Explain the ACID properties of a database transaction. Why are they important?

38. What is the difference between DELETE, TRUNCATE, and DROP statements in SQL?

39. How would you design a database schema for a simple e-commerce application?

40. Explain the use of the HAVING clause in SQL. How does it differ from the WHERE clause?

41. Describe the use of the UNION and UNION ALL operators in SQL. What is the key difference between them?

42. How do you create a stored procedure in SQL? What are the benefits of using stored procedures?

43. Explain the concept of database triggers. Provide an example of when you might use a trigger.

44. What is the purpose of using the EXPLAIN statement before a SQL query? How can it help with optimization?

45. How do you handle concurrency issues in a database environment? Explain concepts like locking.

46. Explain how to optimize a slow-performing SQL query. What are the key steps you'd take?

47. Describe the difference between clustered and non-clustered indexes in SQL.

48. How do you handle deadlocks in SQL Server? Explain with example scenarios.

49. What are the advantages and disadvantages of using stored procedures?

50. Explain window functions in SQL and provide examples of how they can be used.

51. Describe different isolation levels in SQL Server and how they affect concurrency.

52. How can you implement pagination in SQL queries?

53. Explain the purpose of query hints and when you might use them.

54. How would you design a SQL database schema for an e-commerce website?

55. Describe common SQL injection vulnerabilities and how to prevent them.

56. What are common table expressions (CTEs) and how can they be helpful?

57. How would you optimize a SQL query that involves multiple joins?

58. Explain the concept of normalization and denormalization in database design.

59. Describe the different types of SQL joins and their use cases.

60. How do you handle errors and exceptions in SQL stored procedures?

61. Explain data warehousing concepts and their importance.

62. How can you improve the performance of full-text search in SQL Server?

- 63. Describe the use of triggers in SQL Server and potential drawbacks.
- 64. How would you design a database schema to store time-series data?
- 65. Explain the concept of data partitioning in SQL Server.

66. How would you implement auditing in a SQL Server database?

67. Describe the role of the SQL Server Agent and its use cases.

68. How can you monitor SQL Server performance and identify bottlenecks?

69. Explain different methods of backing up and restoring a SQL Server database.

70. How can you implement role-based access control in SQL Server?

71. Describe the purpose of the SQL Server profiler and how to use it.

72. How can you migrate data between different SQL Server databases?

73. Explain the concept of data encryption in SQL Server and its benefits.

74. How would you optimize a slow-running SQL query, detailing the steps from identification to implementation?

75. Explain the concept of a 'covering index' and how it can improve query performance. Give a practical example.

76. Describe a scenario where using a correlated subquery is more efficient than using a join, and explain why.

77. How do you handle data skew in a database, and what are the potential performance implications?

78. Explain the different isolation levels in SQL and their impact on concurrency and data integrity. Provide examples of when to use each level.

79. Describe the process of database sharding and its benefits and drawbacks. How would you implement it in a real-world scenario?

80. Explain the difference between clustered and non-clustered indexes, and when would you choose one over the other?

81. How can you use window functions to solve complex analytical problems? Provide a specific example.

82. What are some strategies for dealing with deadlocks in a database system, and how can you prevent them?

83. Explain the concept of query hints and when they should (and shouldn't) be used.

84. How would you design a database schema to efficiently store and query time-series data?

85. Describe the difference between optimistic and pessimistic locking, and explain when each is appropriate.

86. How would you implement a full-text search capability in a SQL database?

87. Explain the concept of normalization and denormalization in database design. When would you choose to denormalize a database?

88. How would you audit changes to data in a SQL database?

89. Describe how you would handle versioning of data in a SQL database.

90. How do you diagnose and resolve performance bottlenecks related to disk I/O in a database system?

91. Explain the role of the query optimizer in SQL Server. How can you influence its behavior?

92. Describe how you would implement a data warehouse using SQL. What are the key considerations?

93. How do you handle large object (LOB) data types in SQL databases efficiently?

94. Explain how to optimize SQL queries that involve complex joins across multiple tables.

95. How do you handle data migration between different database systems with minimal downtime?

96. Describe how you would implement a geospatial data solution using SQL.

97. Explain the concept of database partitioning and its advantages and disadvantages.

98. How do you ensure data consistency across multiple databases in a distributed environment?

99. Describe how you would implement a recommendation system using SQL.