SQL Interview Questions and Answers

@bzlearnin

SQL Interview Questions and Answers

1. What is SQL?

SQL means Structured Query Language and is used to communicate with relational databases. It proposes a standardized way to interact with databases, allowing users to perform various operations on the data, including retrieval, insertion, updating, and deletion.

2. What are the different types of SQL commands?

- SELECT: Retrieves data from a database.
- INSERT: Adds new records to a table.
- UPDATE: Modifies existing records in a table.
- DELETE: Removes records from a table.
- CREATE: Creates a new database, table, or view.

- ALTER: Modifies the existing database object structure.
- DROP: Deletes an existing database object.

3. What is a primary key in SQL?

It is a unique identifier for each record in a table. It ensures that each row in the table has a distinct and non-null value in the primary key column. Primary keys enforce data integrity and create relationships between tables.

4. What is a foreign key?

It is a field in one table referencing the primary key in another. It establishes a relationship between the two tables, ensuring data consistency and enabling data retrieval across tables.

5. Explain the difference between DELETE and TRUNCATE commands.

The DELETE command is used by professionals to remove particular rows from a table based on a condition, allowing you to selectively delete records. TRUNCATE, on the other hand, removes all rows from a table without specifying conditions. TRUNCATE is faster and uses fewer system resources than DELETE but does not log individual row deletions.

6. What is a JOIN in SQL, and what are its types?

A JOIN operation merges information from two or more tables by utilizing a common column that links them together. Various types of JOINs exist, like INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN. These JOIN variations dictate the manner in which data from the involved tables is paired and retrieved.

7. What do you mean by a NULL value in SQL?

A NULL value in SQL represents the absence of data in a column. It is not the same as an empty string or zero; it signifies that the data is missing or unknown. NULL values can be used in columns with optional data or when the actual data is unavailable.

8. Define a Unique Key in SQL.

Often referred to as a unique constraint, a unique key guarantees that every value in a column (or a combination of columns) remains distinct and cannot be replicated within a table. In contrast to a primary key, a table has the flexibility to incorporate multiple unique keys.

9. What is a database?

A database is a systematically organized collection of data arranged into tables composed of rows and columns. The primary purpose of databases is to efficiently store, manage, and retrieve data.

10. Explain the differences between SQL and NoSQL databases.

SQL databases are characterized by their use of structured tables and strict adherence to a predefined schema, making them ideal for managing structured data with a strong focus on data consistency and transaction support. In contrast, NoSQL databases are non-relational and excel in handling unstructured or semi-structured data, frequently employed for scalable, distributed, and adaptable data storage solutions.

11. What is a table and a field in SQL?

In SQL, a table is a structured data collection organized into rows and columns. Each column in a table is called a field, representing a specific attribute or property of the data.

12. Describe the SELECT statement.

The SELECT statement serves the purpose of fetching data from one or multiple tables, enabling you to specify the desired columns to retrieve, apply filters through the WHERE clause, and manage the result's sorting using the ORDER BY clause.

13. What is a constraint in SQL? Name a few.

A constraint in SQL defines rules or restrictions that apply to data in a table, ensuring data integrity. Common constraints include:

- PRIMARY KEY: Ensures the values' uniqueness in a column.
- FOREIGN KEY: Enforces referential integrity between tables.
- UNIQUE: Ensures the uniqueness of values in a column.
- CHECK: Defines a condition that data must meet to be inserted or updated.
- NOT NULL: Ensures that there are no NULL values in a column.

14. What is normalization in SQL?

Normalization is the method used to streamline data storage within a database, reducing redundancy and enhancing data integrity. This approach entails dividing tables into more manageable, interrelated tables and establishing connections between them.

15. How do you use the WHERE clause?

The WHERE clause within SQL queries serves the purpose of selectively filtering rows according to specified conditions, thereby enabling you to fetch exclusively those rows that align with the criteria you define. For example:

```
SELECT * FROM employees WHERE department = 'HR';
```

16. What are indexes in SQL?

Indexes improve the data retrieval operations speed. They provide a quick way to locate specific rows in a table by creating a sorted data structure based on one or more columns. Indexes are essential for optimizing query performance.

17. Explain GROUP BY in SQL.

The GROUP BY clause organizes rows from a table into groups based on the values in one or more columns. It is commonly employed alongside aggregate functions like SUM, COUNT, AVG, MIN, and MAX to perform computations on data that has been grouped together.

18. What is an SQL alias?

An SQL alias serves as a transitory label bestowed upon either a table or a column within a query, with the primary purpose of enhancing the clarity of query outcomes or simplifying the process of renaming columns for improved referencing. For example:

```
SELECT first_name AS "First Name", last_name AS "Last Name" FROM employees;
```

19. Explain ORDER BY in SQL.

The ORDER BY clause is used to sort the result set of a query based on one or more columns. You can specify each column's sorting order (ascending or descending). For example:

```
SELECT * FROM products ORDER BY price DESC;
```

20. Describe the difference between WHERE and HAVING in SQL.

The WHERE clause is employed to restrict individual rows before they are grouped, such as when filtering rows prior to a GROUP BY operation. Conversely, the HAVING clause is utilized to filter groups of rows after they have been grouped, like filtering groups based on aggregate values.

21. What is a view in SQL?

An SQL view is essentially a virtual table that derives its data from the outcome of a SELECT query. Views serve multiple purposes, including simplifying intricate queries, enhancing data security through an added layer, and enabling the presentation of targeted data subsets to users, all while keeping the underlying table structure hidden.

22. What is a stored procedure?

A SQL stored procedure comprises precompiled SQL statements that can be executed together as a unified entity. These procedures are commonly used to encapsulate business logic, improve performance, and ensure consistent data manipulation practices.

23. What is a trigger in SQL?

An SQL trigger consists of a predefined sequence of actions that are executed automatically when a particular event occurs, such as when an INSERT or DELETE operation is performed on a table. Triggers are employed to ensure data consistency, conduct auditing, and streamline various tasks.

24. What are aggregate functions? Can you name a few?

Aggregate functions in SQL perform calculations on a set of values and return a single result.

- SUM: To calculate the sum of values in a column.
- COUNT: To count a column's number of rows or non-null values.
- AVG: To calculate the average of values in a column.
- MIN: To retrieve the minimum value in a column.
- MAX: To retrieve the maximum value in a column.

25. How do you update a value in SQL?

The UPDATE statement serves the purpose of altering pre-existing records within a table. It involves specifying the target table for the update, the specific columns to be modified, and the desired new values to be applied. For example:

```
UPDATE employees SET salary = 60000 WHERE department = 'IT';
```

Intermediate SQL Interview Questions and Answers

26. What is a self-join, and how would you use it?

A self-join is a type of join where a table is joined with itself. It is useful when creating relationships within the same table, such as finding hierarchical relationships or comparing rows with related data.

27. Explain different types of joins with examples.

- INNER JOIN: Gathers rows that have matching values in both tables.
- RIGHT JOIN: Gathers all rows from the right table and any matching rows from the left table.
- LEFT JOIN: Gathers all rows from the left table and any matching rows from the right table.
- FULL JOIN: Gathers all rows when there's a match in either table, including unmatched rows from both tables.

Example:

```
SELECT employees.name, departments.name

FROM employees

LEFT JOIN departments ON employees.department_id = departments.id;
```

28. What is a subquery? Provide an example.

A subquery refers to a query that is embedded within another query, serving the purpose of fetching information that will subsequently be employed as a condition or value within the encompassing outer query. For example, to find employees with salaries greater than the average salary:

```
SELECT name

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);
```

29. How do you optimize SQL queries?

SQL query optimization involves improving the performance of SQL queries by reducing resource usage and execution time. Strategies include using appropriate indexes, optimizing query structure, and avoiding costly operations like full table scans.

30. What is the difference between UNION and UNION ALL?

UNION merges the outcomes of two or more SELECT statements, removing duplicate rows, whereas UNION ALL merges the results without removing duplicates. While UNION ALL is faster, it may include duplicate rows.

31. What are correlated subqueries?

It is a type of subquery that makes reference to columns from the surrounding outer query. This subquery is executed repeatedly, once for each row being processed by the outer query, and its execution depends on the outcomes of the outer query.

32. Explain ACID properties in SQL.

ACID - Atomicity, Consistency, Isolation, and Durability. They are essential properties that ensure the reliability and integrity of database transactions:

- Atomicity (single, indivisible unit of transactions)
- Consistency (transactions bring the DB from one consistent state to another)
- Isolation (transactions are isolated from each other)
- Durability (committed transactions are permanent and survive system failures)

33. What is a transaction in SQL?

A transaction in SQL is a sequence of one or more SQL operations treated as a single unit of work. Transactions ensure that database operations are either completed successfully or rolled back entirely in case of failure.

34. How do you implement error handling in SQL?

Error handling in SQL is typically achieved using try-catch blocks (in SQL Server) or EXCEPTION blocks (in Oracle). These blocks allow you to handle and log errors gracefully to prevent application crashes.

35. What is a cursor, and how is it used?

In SQL, a cursor is a database element employed for the purpose of fetching and controlling data one row at a time from a result set. Cursors find frequent application within stored procedures or triggers when it becomes necessary to process data in a sequential manner.

36. Describe the data types in SQL.

SQL supports various data types, including numeric, character, date/time, and binary types. Common data types include INT, VARCHAR, DATE, and BLOB, among others. Data types define the kind of values a column can hold.

37. Explain normalization and denormalization.

Normalization is the method used to streamline data in a database, decreasing redundancy and enhancing data integrity. This procedure includes dividing large tables into smaller, interconnected ones to eliminate duplicated data. Conversely, denormalization is the deliberate act of introducing redundancy to enhance query performance.

38. What is a clustered index?

A clustered index in SQL determines the physical order of data rows in a table. Each table can have only one clustered index, which impacts the table's storage structure. Rows in a table are physically stored in the same order as the clustered index key.

39. How do you prevent SQL injection?

SQL injection represents a security flaw that arises when SQL queries mishandle untrusted data, posing a risk of unauthorized access or data tampering. To ward off SQL injection, employ techniques like parameterized queries, prepared statements, input validation, and the enforcement of stringent access controls.

40. What are the different types of triggers?

- DML triggers: These triggers fire in response to data manipulation language (DML) operations such as INSERT, UPDATE, or DELETE.
- DDL triggers: These triggers fire in response to data definition language (DDL) events, such as table or view creation.

41. Explain the concept of a database schema.

In SQL, a database schema functions as a conceptual container for housing various database elements, such as tables, views, indexes, and procedures. Its primary purpose is to facilitate the organization and segregation of these database elements while specifying their structure and interconnections.

42. How is data integrity ensured in SQL?

Data integrity in SQL is ensured through various means, including constraints (e.g., primary keys, foreign keys, check constraints), normalization, transactions, and referential integrity constraints. These mechanisms prevent invalid or inconsistent data from being stored in the database.

43. What is an SQL injection?

SQL injection is a cybersecurity attack method that involves the insertion of malicious SQL code into an application's input fields or parameters. This unauthorized action enables attackers to illicitly access a database, extract confidential information, or manipulate data.

44. How do you create a stored procedure?

You use the CREATE PROCEDURE statement to create a stored procedure in SQL. A stored procedure can contain SQL statements, parameters, and variables. Here's a simple example:

```
CREATE PROCEDURE GetEmployeeByID(@EmployeeID INT)
AS
BEGIN
  SELECT * FROM employees WHERE employee_id = @EmployeeID;
END;
```

45. What is a deadlock in SQL? How can it be prevented?

A deadlock in SQL occurs when two or more transactions cannot proceed because they are waiting for resources held by each other. Deadlocks can be prevented or resolved by using techniques such as locking hierarchies, timeouts, or deadlock detection and resolution mechanisms.

Advanced SQL Interview Questions

46. Explain different isolation levels in SQL.

Isolation levels define the visibility of data changes one transaction makes to other concurrent transactions. There are four commonly used isolation levels in SQL:

- READ UNCOMMITTED: At this isolation level, transactions are allowed to read changes made by other transactions even if those changes have not been committed. While this provides the highest level of concurrency, it also introduces the risk of encountering dirty reads.
- READ COMMITTED: In this level, transactions can only read committed data, avoiding dirty reads. However, it may still suffer from non-repeatable reads and phantom reads.
- REPEATABLE READ: Transactions at this level ensure that any data read during the transaction remains unchanged throughout the transaction's lifetime. It prevents non-repeatable reads but may still allow phantom reads.
- SERIALIZABLE: This represents the utmost isolation level, guaranteeing absolute isolation between transactions. While it eradicates all concurrency problems, it may exhibit reduced efficiency due to locking mechanisms.

47. How does a clustered index work and how is it different from a non-clustered index?

A clustered index defines the actual storage order of rows within a table, allowing for only one clustered index per table and directly influencing the on-disk data organization. Conversely, a non-clustered index does not impact the physical arrangement of data and can coexist with multiple indexes within the same table.

- Clustered Index: When you create a clustered index on a table, the table's rows are physically rearranged to match the order of the indexed column(s). This makes range queries efficient but may slow down insert/update operations.
- Non-clustered Index: Non-clustered indexes are separate data structures that store a copy of a portion of the table's data and point to the actual data rows. They improve read performance but come with some overhead during data modification.

48. Discuss SQL server reporting services.

SQL Server Reporting Services is a reporting tool provided by Microsoft for creating, managing, and delivering interactive, tabular, graphical, and free-form reports. SSRS allows users to design and generate reports from various data sources, making it a valuable asset for businesses needing comprehensive reporting capabilities.

49. What are ctes (Common table expressions)?

Common Table Expressions (CTEs) serve as momentary result sets that you can mention within SQL statements, typically found within SELECT, INSERT, UPDATE, or DELETE operations. They're established using the `WITH` keyword and are instrumental in streamlining intricate queries by dividing them into more digestible components.

50. Explain the MERGE statement.

The SQL MERGE statement is employed to execute insertions, updates, or deletions on a target table, guided by the outcomes of a source table or query. It consolidates the functionalities of several individual statements (INSERT, UPDATE, DELETE) into one comprehensive statement, rendering it particularly valuable for achieving data synchronization between tables.

51. How do you use a window function in SQL?

Window functions are employed to carry out computations on a group of table rows that are associated with the current row. They enable the generation of result sets containing aggregated data while retaining the distinct details of each row. Typical window functions encompass ROW_NUMBER(), RANK(), DENSE_RANK(), and SUM() OVER().

52. What is a pivot table and how do you create one in SQL?

A pivot table is a technique used to rotate or transpose rows into columns to better analyze and summarize data. You can create pivot tables in SQL using the `PIVOT` operator to convert row-based data into a column-based format.

53. Describe the process of database mirroring.

Database mirroring is a high-availability solution in SQL Server that involves creating and maintaining redundant copies of a database on separate servers. It ensures data availability and disaster recovery by automatically failing over to the mirror server in case of a primary server failure.

54. Explain the concept of table partitioning.

Partitioning a table involves the strategy of breaking down a sizable table into smaller, more easily handled segments known as partitions. This method can enhance query efficiency by permitting SQL Server to focus solely on pertinent partitions while executing queries. Typically, partitioning is carried out using a column characterized by a high cardinality, such as date or region.

55. How do you handle transactions in distributed databases?

Handling transactions in distributed databases involves ensuring the ACID (Atomicity, Consistency, Isolation, Durability) properties across multiple databases or nodes. This can be achieved through distributed transaction management protocols like Two-Phase Commit (2PC) or by using distributed database systems designed for this purpose.

56. What is the use of the explain plan?

The EXPLAIN plan is a valuable feature found in numerous relational database management systems. This tool offers a comprehensive view of the database engine's strategy for executing a query, encompassing details such as the selected execution plan, join techniques, index utilization, and projected costs. Database administrators (DBAs) and developers rely on EXPLAIN plans to enhance the performance of their queries.

57. Discuss SQL server integration services (SSIS).

Microsoft provides SQL Server Integration Services as a powerful ETL (Extract, Transform, Load) tool. It enables data integration from various sources, transformation of data as needed, and loading it into destination systems like data warehouses or databases.

58. What are indexed views?

Indexed views, or materialized views, are precomputed result sets stored as physical tables in the database. They improve query performance by allowing the database engine to access pre-aggregated or pre-joined data directly from the indexed view, reducing the need for complex query processing.

59. Explain the concept of database sharding.

Database sharding is a horizontal partitioning technique that distributes data across multiple database instances or servers. It's commonly used in large-scale systems to improve scalability and performance. Each shard contains a subset of the data, and a sharding strategy determines how data is distributed.

60. How do you manage large-scale databases for performance?

Managing large-scale databases for performance involves various strategies, including proper indexing, partitioning, query optimization, hardware optimization, and caching. Monitoring and fine-tuning the database is crucial to ensure optimal performance as data volumes grow.

61. What is a materialized view?

Materialized views are a type of database component designed to maintain the outcomes of a query in the form of a tangible table. These views undergo periodic updates to ensure that the stored data remains current. They are employed to enhance the efficiency of database queries, particularly for intricate or frequently executed ones.

62. Discuss the strategies for database backup and recovery.

Ensuring data availability and disaster recovery relies on the implementation of vital backup and recovery strategies. These strategies encompass various methods, such as full backups, differential backups, transaction log backups, and regular testing of restoration procedures.

63. What are the best practices for securing a SQL database?

Securing a SQL database involves implementing access controls, encryption, auditing, and regular security assessments. Best practices include using strong authentication, limiting permissions, and keeping database systems and software up to date.

64. Explain the concept of database replication.

It is the process of copying and synchronizing data from one database to another. It ensures data availability, load balancing, and disaster recovery. Common replication types include snapshot replication, transactional replication, and merge replication.

65. How do you monitor SQL server performance?

Monitoring SQL Server performance involves tracking key performance metrics, setting up alerts for critical events, and analyzing performance bottlenecks. Tools like SQL Server Profiler and Performance Monitor are commonly used for this purpose.

66. What is a database warehouse?

A database warehouse is a centralized repository that stores data from various sources for analytical and reporting purposes. It is optimized for querying and analysis and often contains historical data.

67. Explain the use of full-text search in SQL.

Full-text search in SQL allows users to search for text-based data within large text fields or documents. It uses advanced indexing and search algorithms to provide efficient and accurate text-searching capabilities.

68. How do you manage database concurrency?

Database concurrency involves a database system's capability to manage multiple concurrent transactions while upholding data integrity. To achieve this, various techniques such as locking mechanisms, optimistic concurrency control, and isolation levels are employed to oversee and regulate database concurrency.

69. What are the challenges in handling big data in SQL?

Handling big data in SQL involves dealing with large volumes of data that exceed the capabilities of traditional database systems. Challenges include data storage, processing, scalability, and efficient querying. Solutions may include distributed databases and big data technologies like Hadoop and Spark.

70. How do you implement high availability in SQL databases?

High availability in SQL databases ensures that the database remains accessible and operational despite failures. Techniques like clustering, replication, and failover mechanisms help achieve high availability.

71. Explain the use of XML data type in SQL server.

The XML data type allows to store, retrieve, and manipulate XML data. It provides support for querying XML documents using XQuery, and it's commonly used in applications that deal with XML data structures.

72. Discuss the concept of NoSQL databases and their interaction with SQL.

NoSQL databases are non-relational databases designed for handling large volumes of unstructured or semi-structured data. They interact with SQL databases through various integration methods, such as data pipelines, ETL processes, and API-based data transfers.

73. What is a spatial database?

A spatial database stores and queries geometric and geographic data, such as maps, GPS coordinates, and spatial objects. It provides specialized functions and indexing methods to support spatial queries and analysis.

74. How do you migrate a database from one server to another?

Database migration involves moving a database from one server or platform to another. This undertaking demands careful planning, data transfer, schema conversion, and thorough testing to ensure a smooth transition while minimizing the potential for data loss or system downtime.

75. Discuss advanced optimization techniques for SQL queries.

Advanced optimization techniques for SQL queries include using query hints, indexing strategies, query rewriting, and understanding the query execution plan. Profiling tools and performance monitoring are essential for identifying and resolving performance bottlenecks.