

SQL Interview Questions

Download PDF

Print

Curated questions for beginner, experienced, and scenario-based interviews. Use the filter to jump to what you need.

Filter (e.g., joins, window functions, indexing, normalization)...

Showing 246 of 246

Top SQL Interview Questions for Entry-Level Candidates

Fundamentals such as SELECT/WHERE, joins, aggregations, GROUP BY/HAVING, constraints, and basic indexing/normalization.

1. What is SQL?

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

2. What are the different types of SQL commands?

There are several types of SQL commands, including: DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), and TCL (Transaction Control Language).

3. What is the difference between a primary key and a foreign key?

A primary key is a unique identifier for each record in a table. A foreign key is a field in one table that refers to the primary key of another table, establishing a relationship between the two tables.

4. What is a join in SQL?

A join in SQL is a way to combine rows from two or more tables based on a related column between them. The result is a new table that includes columns from all joined tables.

5. What is a subquery in SQL?

A subquery in SQL is a query nested inside another query. It is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

6. What is a transaction in SQL?

A transaction in SQL is a sequence of one or more SQL statements that are executed as a single unit of work. Transactions ensure that the database remains in a consistent state even if some of the statements fail to execute.

7. What is an index in SQL?

An index in SQL is a database object that provides a fast, optimized way to look up data based on the values in one or more columns. Indexes help speed up query performance by reducing the number of rows that need to be scanned.

8. What is a view in SQL?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views can be used to simplify the complexity of a query, or to restrict access to specific columns or rows of a table.

9. What is a stored procedure in SQL?

A stored procedure in SQL is a precompiled collection of one or more SQL statements that can be executed by calling the stored procedure name. Stored procedures are used to encapsulate complex business logic and improve database performance.

10. What is a trigger in SQL?

A trigger in SQL is a special type of stored procedure that automatically executes when an event such as an INSERT, UPDATE, or DELETE occurs on a specified table. Triggers are used to enforce complex business rules or to automatically maintain data consistency.

11. What is the difference between a subquery and a join?

A subquery is a query nested inside another query, while a join is a way to combine rows from two or more tables based on a related column between them. Subqueries are used to retrieve data that will be used in the main query, while joins are used to combine data from multiple tables into a single result set.

12. What is a correlated subquery in SQL?

A correlated subquery in SQL is a subquery that references a column from the outer query. Correlated subqueries are executed repeatedly for each row in the outer query, and their results are used to further restrict the data that is returned by the main query.

13. What is a primary key in SQL?

A primary key in SQL is a column or set of columns in a table that uniquely identifies each row in the table. The primary key is used to enforce data integrity and to create relationships with other tables through foreign keys.

14. What is a foreign key in SQL?



A foreign key in SQL is a column or set of columns in one table that refers to the primary key of another table. Foreign keys are used to enforce referential integrity and to create relationships between tables.

15. What is a unique constraint in SQL?

A unique constraint in SQL is a rule that restricts the values in a column or set of columns to be unique across all rows in a table. Unique constraints are used to enforce data integrity and to ensure that each value in a column is distinct.

16. What is an index in SQL?

An index in SQL is a database object that is used to improve the performance of queries by allowing the database management system to quickly locate the rows that match a specific condition. Indexes work by creating a data structure that maps the values in one or more columns of a table to the location of the corresponding rows in the table.

17. What is a clustered index in SQL?

A clustered index in SQL is a type of index that physically reorders the rows of a table to match the order of the index. There can be only one clustered index per table, as it determines the physical order of the data. The clustered index is used to optimize the performance of queries that retrieve ranges of data.

18. What is a non-clustered index in SQL?

A non-clustered index in SQL is a type of index that stores a separate copy of the indexed columns and a pointer to the corresponding row in the table. Non-clustered indexes are used to improve the performance of queries that retrieve individual rows based on the values in the indexed columns.

19. What is a view in SQL?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views are used to simplify queries by providing a predefined set of columns and rows, or to restrict access to specific columns or rows in a table.

20. What is a stored procedure in SQL?

A stored procedure in SQL is a pre-compiled set of one or more SQL statements that can be executed with a single call. Stored procedures are used to encapsulate complex business logic, to improve the performance of repetitive tasks, and to promote code reuse.

21. What is a transaction in SQL?

A transaction in SQL is a sequence of one or more database operations that are executed as a single unit of work. Transactions ensure the consistency and integrity of data by either committing all changes or rolling back all changes in the event of an error or failure.

22. What is a trigger in SQL?

A trigger in SQL is a special type of stored procedure that is automatically executed in response to specific changes in the data, such as insertions, updates, or deletions. Triggers are used to enforce business rules, to maintain data integrity, and to enforce security.

23. What is a cursor in SQL?

A cursor in SQL is a database object that is used to retrieve and manipulate data one row at a time. Cursors are used to process large result sets, to perform complex calculations, and to maintain data integrity when updating data in multiple tables.

24. What is a temporary table in SQL?

A temporary table in SQL is a table that is created and used for a single session or transaction and is automatically dropped when the session or transaction ends. Temporary tables are used to store intermediate results for processing, to improve performance, and to simplify complex queries.

25. What is a dynamic SQL in SQL?

Dynamic SQL in SQL is a type of SQL statement that is generated and executed at runtime, rather than being pre-compiled and stored in the database. Dynamic SQL is used to dynamically construct and execute SQL statements based on input parameters or changing conditions.

26. What is a normalization in SQL?

Normalization in SQL is a database design technique that reduces data redundancy and improves data integrity by dividing a database into two or more separate tables and defining relationships between the tables. Normalization is used to minimize data duplication and ensure that data is stored in a consistent and organized manner.

27. What is a join in SQL?

A join in SQL is a query operation that combines rows from two or more tables based on related columns between the tables. There are several types of joins, including inner join, left join, right join, and full outer join, and each type returns a different combination of matching and non-matching rows.

28. What is a subquery in SQL?

 A subquery in SQL is a query that is nested inside another query and is used to return data that will be used in the main query. Subqueries can be used to simplify complex queries, to perform calculations, and to retrieve data from multiple tables.

29. What is a self-join in SQL?

A self-join in SQL is a join of a table to itself. Self-joins are used to compare rows within a single table, to identify relationships between rows, or to split a table into groups.

30. What is a UNION in SQL?

A UNION in SQL is a query operator that combines the results of two or more SELECT statements into a single result set. UNION is used to combine data from multiple tables or to return the distinct values from multiple SELECT statements.

31. What is a view in SQL?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views are used to simplify complex queries, to enforce security and data privacy, and to provide a simplified representation of data for users.

32. What is a stored procedure in SQL?

A stored procedure in SQL is a pre-compiled collection of SQL statements that are stored in the database and can be executed by multiple users. Stored procedures are used to encapsulate complex logic, to improve performance, and to simplify the management of data.

33. What is a constraint in SQL?

A constraint in SQL is a rule that is used to enforce data integrity and to ensure that data is entered in a consistent and correct manner. Constraints can be applied to columns, tables, or the entire database, and they can be used to enforce data types, ranges, uniqueness, referential integrity, and other data rules.

34. What is a data type in SQL?

A data type in SQL is a classification of the type of data that can be stored in a column of a table. Data types determine the type of data that can be stored, the size of the data, and the operations that can be performed on the data.

35. What is a primary key in SQL?

A primary key in SQL is a unique identifier for each row in a table that is used to enforce data integrity and to ensure that data is entered in a consistent and correct manner. The primary key is used to enforce referential integrity, to index the table for faster search and retrieval, and to enforce uniqueness of data.

36. What is an index in SQL?

An index in SQL is a database object that is used to improve the speed of data retrieval operations on a table. An index provides a fast and efficient way to search and retrieve data based on the values in specific columns, and it can improve the performance of complex queries and join operations.

37. What is a transaction in SQL?

A transaction in SQL is a sequence of one or more SQL statements that are executed as a single unit of work. Transactions are used to ensure the atomicity, consistency, isolation, and durability (ACID) properties of data operations, and they are used to coordinate multiple statements that must be executed as a single operation.

38. What is a trigger in SQL?

A trigger in SQL is a database object that is used to automatically execute a set of SQL statements when an event occurs, such as an insert, update, or delete. Triggers are used to enforce business rules, to implement complex data validation and integrity checks, and to maintain audit trails and other data history.

39. What is a foreign key in SQL?

A foreign key in SQL is a column or set of columns in a table that refers to the primary key of another table. Foreign keys are used to enforce referential integrity, which ensures that data is entered in a consistent and correct manner, and they are used to create relationships between tables and to maintain data consistency.

40. What is a Cursor in SQL?

A cursor in SQL is a database object that is used to retrieve data from a result set one row at a time. Cursors are used to process data row by row, to perform complex calculations, and to manipulate data in a way that is not possible with a standard SELECT statement.

41. What is a subquery in SQL?

A subquery in SQL is a query that is nested inside another query. Subqueries are used to retrieve data that is used as input to another query, and they can be used to return a single value, a set of values, or a table that is used as part of a larger query.

42. What is a join in SQL?



A join in SQL is an operation that combines rows from two or more tables into a single result set based on a related column between the tables. Joins are used to retrieve data from multiple tables as a single result set, and they can be used to implement complex relationships and data manipulations.

43. What is a union in SQL?

A union in SQL is an operation that combines the result sets of two or more SELECT statements into a single result set. Unions are used to combine data from multiple tables into a single result set, and they can be used to perform complex data manipulations and to simplify complex queries.

44. What is a self join in SQL?

A self join in SQL is a join that is performed on a single table, where the table is joined to itself. Self joins are used to retrieve data from a single table based on relationships within the data, and they can be used to implement complex data manipulations and to simplify complex queries.

45. What is a outer join in SQL?

An outer join in SQL is a join that returns all the rows from one table and the matching rows from another table, and where the non-matching rows from the second table are filled with NULL values. Outer joins are used to retrieve data from multiple tables, and they can be used to implement complex relationships and data manipulations.

46. What is a stored procedure in SQL?

A stored procedure in SQL is a pre-compiled collection of SQL statements that can be executed multiple times as a single unit of work. Stored procedures are used to encapsulate complex business logic, to improve performance by reducing network round trips, and to improve security by hiding implementation details.

47. What is a view in SQL?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views are used to simplify complex queries, to provide security by hiding implementation details, and to simplify data maintenance by encapsulating data transformations and manipulations.

48. What is a primary key in SQL?

A primary key in SQL is a column or set of columns in a table that uniquely identifies each row in the table. Primary keys are used to enforce referential integrity, to improve performance by creating an index, and to ensure the uniqueness of data in the table.

49. What is a unique constraint in SQL?

A unique constraint in SQL is a restriction that enforces the uniqueness of data in a column or set of columns in a table. Unique constraints are used to enforce data integrity and to ensure that data is entered in a consistent and correct manner.

50. What is a default constraint in SQL?

A default constraint in SQL is a rule that specifies a default value for a column in a table. Default constraints are used to ensure that data is entered in a consistent and correct manner, and to provide default values for columns when no other values are specified.

51. What is a transaction in SQL?

A transaction in SQL is a unit of work that consists of multiple SQL statements that are executed as a single unit. Transactions are used to ensure the consistency and integrity of data, by allowing multiple statements to be executed as a single unit, and by allowing the transaction to be rolled back if any errors occur.

52. What is a trigger in SQL?

A trigger in SQL is a set of SQL statements that are automatically executed in response to an event, such as the insertion, update, or deletion of data in a table. Triggers are used to enforce business logic, to maintain data integrity, and to perform complex data manipulations and updates.

53. What is a constraint in SQL?

A constraint in SQL is a restriction that is placed on the data in a table, to ensure that data is entered in a consistent and correct manner. Constraints can be used to enforce referential integrity, to ensure the uniqueness of data, and to specify default values for columns.

54. What is an index in SQL?

An index in SQL is a database object that is used to improve the performance of queries. Indexes are used to improve the speed of data retrieval operations by allowing the database to search for data more quickly and efficiently. Indexes can be created on one or more columns in a table, and they can be used to improve the performance of queries that retrieve data based on the values in these columns.

55. What is a normalization in SQL?

Normalization in SQL is a process of organizing data in a database to reduce redundancy and to improve data integrity. Normalization involves dividing a database into two or more tables and defining relationships between the tables to ensure that data is stored in a consistent and correct manner.

56. What is a join in SQL?



A join in SQL is an operation that combines rows from two or more tables into a single result set based on a related column between the tables. Joins are used to combine data from multiple tables and to retrieve data based on the relationships between the tables.

57. What is a subquery in SQL?

A subquery in SQL is a query that is nested inside another query. Subqueries are used to perform complex data manipulations and to retrieve data based on the results of other queries.

58. What is a self join in SQL?

A self join in SQL is a join in which a table is joined with itself. Self joins are used to retrieve data from a single table based on relationships between the rows in the table.

59. What is a UNION in SQL?

A UNION in SQL is an operation that combines the results of two or more SELECT statements into a single result set. The UNION operation returns only unique rows, and it removes duplicates from the result set.

60. What is a CASE statement in SQL?

A CASE statement in SQL is an expression that is used to perform conditional logic in a SELECT, UPDATE, or DELETE statement. The CASE statement allows you to perform different actions based on different conditions, and it can be used to return different values based on different conditions.

61. What is a stored procedure in SQL?

A stored procedure in SQL is a pre-compiled collection of SQL statements that are stored in the database. Stored procedures are used to encapsulate complex business logic, to improve performance by reducing the amount of data that is transferred between the database and the client, and to improve security by hiding the underlying SQL code from the client.

62. What is a view in SQL?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views are used to encapsulate complex data structures, to simplify the structure of a database, and to provide a level of security by hiding the underlying data from unauthorized users.

63. What is a Cursor in SQL?

A cursor in SQL is a database object that is used to retrieve data from a result set one row at a time. Cursors are used to perform complex data manipulations and to retrieve data in a specific order.

64. What is a primary key in SQL?

A primary key in SQL is a unique identifier that is used to identify each row in a table. The primary key is used to enforce referential integrity, to ensure the uniqueness of data, and to provide a fast way to access data in the table.

65. What is a foreign key in SQL?

A foreign key in SQL is a field in one table that is used to establish a relationship with another table. Foreign keys are used to enforce referential integrity, to ensure that data is entered in a consistent and correct manner, and to provide a fast way to access data in related tables.

66. What is an index in SQL?

An index in SQL is a database object that is used to improve the performance of queries by allowing the database to quickly locate the rows that match a specific condition. An index contains a copy of the data in a table, but it is organized in a way that makes it faster to retrieve the data that you want.

67. What is a transaction in SQL?

A transaction in SQL is a sequence of one or more database operations that are treated as a single unit of work. Transactions are used to ensure that data is entered into the database in a consistent and correct manner, even if there are errors or problems with individual operations in the transaction.

68. What is a trigger in SQL?

A trigger in SQL is a database object that is used to automatically execute specific actions when certain events occur in the database. Triggers are used to enforce business rules, to maintain data integrity, and to automate complex data manipulations.

69. What is a normalization in SQL?

Normalization in SQL is the process of organizing data in a database to minimize data redundancy and to eliminate data anomalies. Normalization is achieved by dividing a database into two or more tables and defining relationships between the tables.

70. What is a constraint in SQL?

 A constraint in SQL is a rule that is used to enforce data integrity in a database. Constraints are used to ensure that data is entered in a consistent and correct manner, to enforce business rules, and to prevent data anomalies.

71. What is indexing in SQL?

Indexes are special lookup tables that need to be used by the database search engine to speed up data retrieval. An index is simply a reference to data in a table. A database index is similar to the index in the back of a journal. It cannot be viewed by the users and just used to speed up the database access. Note: Updating a table with indexes takes more time than updating a table without (because the indexes also need an update).

Top SQL Interview Questions for Experienced Candidates

Advanced topics like window functions, CTEs/recursion, query planning, performance tuning, partitioning, transactions/isolation, and security.

1. What is SQL and what does it stand for?

SQL stands for Structured Query Language and is used for managing and manipulating relational database management systems.

2. Explain the difference between a primary key and a foreign key.

A primary key is a unique identifier for each record in a table and ensures data integrity by preventing duplicate values. A foreign key is a field in a table that refers to the primary key of another table, establishing a relationship between the two tables.

3. What is normalization in database management, and why is it important?

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity by splitting large tables into smaller, related tables. It is important for maintaining data accuracy, reducing data duplication, and improving database performance.

4. How do you create a table in SQL?

To create a table in SQL, you use the "CREATE TABLE" command followed by the name of the table, the names and data types of columns, and any constraints such as primary keys or foreign keys.

5. What is a join in SQL, and what are the different types of joins?

A join is a method for combining data from two or more tables based on a related column between them. The different types of joins in SQL are Inner Join, Left Join, Right Join, and Full Outer Join.

6. Explain the difference between a clustered and a non-clustered index.

A clustered index is a type of index that physically reorders the rows of a table based on the values in the indexed column, making it faster to retrieve data based on that column. A non-clustered index is a type of index that stores the indexed values in a separate structure from the original table, allowing for faster searching but slower data modification operations.

7. How do you retrieve data from a table in SQL?

To retrieve data from a table in SQL, you use the "SELECT" statement followed by the columns you want to retrieve and the "FROM" clause indicating the table name. You can also add conditions to the query using the "WHERE" clause to filter the results.

8. What is an SQL subquery?

An SQL subquery is a query within a query. It is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

9. What is the purpose of the GROUP BY clause in SQL?

The GROUP BY clause in SQL is used to group rows that have the same values into summary rows, like finding the total sum of a particular column for each unique value in another column.

10. What is the difference between the LIKE and IN operators in SQL?

The LIKE operator is used to search for a specified pattern in a column, while the IN operator is used to specify a list of values to match against a column.

11. What is an SQL transaction and why is it important?

An SQL transaction is a sequence of one or more SQL statements that are executed as a single unit of work. Transactions are important for ensuring the integrity and consistency of data in the database, as either all the statements in a transaction are executed or none are executed.

12. What is a view in SQL and why is it used?

A view in SQL is a virtual table based on the result of an SQL statement. Views are used for several purposes such as security, data abstraction, and simplifying complex queries.

13. What is a stored procedure in SQL?



A stored procedure in SQL is a precompiled set of SQL statements that can be executed multiple times by calling the stored procedure's name. Stored procedures are used to encapsulate a series of SQL statements into a single re-usable routine, for better code reuse and improved performance.

14. What is a trigger in SQL and what is it used for?

A trigger in SQL is a special type of stored procedure that is automatically executed in response to certain events, such as data modification, in a database. Triggers are used to enforce business rules and to maintain data integrity.

15. What is the purpose of the HAVING clause in SQL?

The HAVING clause in SQL is used in combination with the GROUP BY clause to filter groups based on a specified condition. It is used to restrict the results of a query based on aggregate values calculated for each group.

16. What is the difference between a clustered and a non-clustered index?

A clustered index is a type of index that physically reorders the rows of a table based on the values in the indexed column, making it faster to retrieve data based on that column. A non-clustered index is a type of index that stores the indexed values in a separate structure from the original table, allowing for faster searching but slower data modification operations.

17. What is the UNION operator in SQL and when is it used?

The UNION operator in SQL is used to combine the result sets of two or more SELECT statements into a single result set. It removes duplicate rows from the final result set.

18. What is normalization in database design and why is it important?

Normalization in database design is the process of organizing data into separate tables to reduce data redundancy and improve data integrity. Normalization is important for avoiding data inconsistencies and anomalies that can occur when data is updated in a poorly designed database.

19. What is the purpose of the DISTINCT keyword in SQL?

The DISTINCT keyword in SQL is used to return only unique values from a SELECT statement. It filters out duplicate rows from the result set.

20. What is an inner join in SQL and when is it used?

An inner join in SQL is a type of join that returns only the matching rows from both tables. It is used to combine rows from two or more tables based on a related column between them.

21. What is a left join in SQL and when is it used?

A left join in SQL is a type of join that returns all the rows from the left table and the matching rows from the right table. If there is no match, the result will contain NULL values for the right-side columns. It is used to combine data from two tables and return all the rows from the left table, even if there is no match in the right table.

22. What is the difference between a primary key and a foreign key in SQL?

A primary key in SQL is a unique identifier for each record in a table. It is used to enforce data integrity and to ensure that no two records in a table have the same value. A foreign key in SQL is a column or set of columns in a table that refers to the primary key of another table, creating a relationship between the two tables.

23. What is a subquery in SQL and when is it used?

A subquery in SQL is a query nested within another query. It is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

24. What is a transaction in SQL and why is it important?

A transaction in SQL is a sequence of one or more SQL statements that are executed as a single unit of work. Transactions are important for maintaining data consistency and for ensuring that either all or none of the statements are executed, even in case of errors or system failures.

25. What is a view in SQL and what is it used for?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views are used to encapsulate complex query logic, to simplify data access and to provide a level of security by limiting access to certain columns or rows in the underlying tables.

26. What is a cursor in SQL and when is it used?

A cursor in SQL is a database object used to retrieve data from a result set one row at a time. Cursors are used to perform operations on a row-by-row basis, where other methods such as a SELECT statement are not suitable.

27. What is the difference between the INNER JOIN and OUTER JOIN in SQL?



An INNER JOIN in SQL returns only the matching rows from both tables, while an OUTER JOIN returns all the rows from one table and the matching rows from the other table. If there is no match, the result will contain NULL values for the non-matching columns. OUTER JOINS are further classified as LEFT JOIN, RIGHT JOIN and FULL OUTER JOIN, depending on which table's non-matching rows are returned.

28. What is the difference between DELETE and TRUNCATE in SQL?

DELETE in SQL is used to delete specific rows from a table, while TRUNCATE is used to delete all the rows from a table and release the table's space to the operating system. TRUNCATE is faster and more efficient than DELETE, but it is not possible to roll back a TRUNCATE operation.

29. What is the difference between WHERE and HAVING clause in SQL?

The WHERE clause in SQL is used to filter rows from a table based on specified conditions, before the data is grouped and aggregated by the SELECT statement. The HAVING clause is used to filter groups of rows after the data has been grouped and aggregated, based on specified conditions.

30. What is a stored procedure in SQL and what is it used for?

A stored procedure in SQL is a pre-compiled collection of SQL statements that can be executed with a single call. Stored procedures are used to encapsulate complex business logic, to improve performance by reducing network traffic and to provide a level of security by controlling access to the underlying data.

31. What is an index in SQL and what is its purpose?

An index in SQL is a database object that provides fast access to data in a table. An index speeds up the query process by allowing the database to quickly locate the rows that match the conditions specified in the WHERE clause, without having to scan the entire table.

32. What is a trigger in SQL and what is it used for?

A trigger in SQL is a special type of stored procedure that automatically executes in response to specified events, such as INSERT, UPDATE or DELETE operations on a table. Triggers are used to enforce business rules, to maintain data consistency and to audit changes to the data.

33. What is normalization in SQL and why is it important?

Normalization in SQL is the process of organizing the data in a database into separate tables based on the relationships between the data. Normalization helps to minimize data redundancy, improve data consistency and simplify the process of updating and querying the data.

34. What is a primary key in SQL and why is it important?

A primary key in SQL is a column or set of columns in a table that uniquely identifies each row in the table. A primary key is important for ensuring data integrity, as it is used to enforce referential integrity constraints and to define relationships between tables.

35. What is a foreign key in SQL and what is its purpose?

A foreign key in SQL is a column or set of columns in one table that refers to the primary key of another table. Foreign keys are used to enforce referential integrity constraints, to ensure that the data in the related tables is consistent and to define relationships between tables.

36. What is a UNION operator in SQL and when is it used?

The UNION operator in SQL is used to combine the results of two or more SELECT statements into a single result set. The UNION operator only returns distinct rows from the combined result set, eliminating duplicates.

37. What is a JOIN operation in SQL and what are the different types of JOINS?

A JOIN operation in SQL is used to combine rows from two or more tables based on a related column between them. The different types of JOINS in SQL are INNER JOIN, LEFT JOIN, RIGHT JOIN and FULL OUTER JOIN, which differ in the way they combine and return the rows from the related tables.

38. What is a subquery in SQL and when is it used?

A subquery in SQL is a query that is nested within another query. Subqueries are used to perform operations in the context of the data retrieved by the outer query. Subqueries can be used in a variety of scenarios, such as to retrieve data from multiple tables or to perform complex calculations.

39. What is a transaction in SQL and why is it important?

A transaction in SQL is a sequence of one or more SQL statements that are executed as a single unit of work. Transactions are important for ensuring data consistency, as they allow you to perform multiple operations as a single, atomic operation, ensuring that either all the operations succeed or all are rolled back in case of an error.

40. What is the difference between a clustered and non-clustered index in SQL?



A clustered index in SQL determines the physical order of the data in a table, while a non-clustered index provides a fast way to look up data in a table without affecting the physical order of the data. In other words, a table can have only one clustered index, but multiple non-clustered indexes.

41. What is a view in SQL and what is it used for?

A view in SQL is a virtual table that is based on the result of a SELECT statement. Views are used to simplify complex queries, to enforce security by limiting access to specific columns or rows of a table, and to encapsulate business logic for easy reuse.

42. What is the difference between a WHERE and ON clause in SQL?

The WHERE clause in SQL is used to filter rows from a table before the data is returned by the SELECT statement. The ON clause in SQL is used to define conditions for joining tables, specifying the relationships between the data in the tables.

43. What is a stored procedure in SQL and what is it used for?

A stored procedure in SQL is a pre-compiled collection of SQL statements that perform a specific task. Stored procedures are used to encapsulate complex logic, to reuse code, to improve security by hiding the underlying tables and to optimize performance by executing complex logic on the database server rather than on the client.

44. What is a trigger in SQL and what is it used for?

A trigger in SQL is a special type of stored procedure that is automatically executed in response to specific changes in the data, such as INSERT, UPDATE or DELETE operations. Triggers are used to enforce complex business rules, to maintain data integrity and consistency, and to perform cascading updates or deletes in related tables.

45. What is a cursor in SQL and what is it used for?

A cursor in SQL is a database object that is used to retrieve data from a result set one row at a time. Cursors are used to iterate through a result set and perform operations on each row, such as updating or deleting the data, and to perform complex data operations that cannot be done using a single SQL statement.

46. What is an index in SQL and what is it used for?

An index in SQL is a database object that provides a fast and efficient way to retrieve data from a table based on the values in one or more columns. Indexes are used to speed up the querying process by allowing the database to quickly locate the rows that match a specific condition, rather than having to scan the entire table.

47. What is a constraint in SQL and what is it used for?

A constraint in SQL is a rule that is applied to the data in a table to ensure data integrity and consistency. Constraints can be used to enforce business rules, such as setting the data type and length of a column, or to enforce referential integrity between related tables, such as by defining primary and foreign keys.

48. What is the difference between a primary key and a unique key in SQL?

A primary key is a unique identifier for a record in a table and it cannot contain null values. Only one primary key can be defined for a table. A unique key, on the other hand, is a unique identifier for a record in a table and can contain null values. Multiple unique keys can be defined for a single table.

49. What is the difference between a join and a union in SQL?

A join operation in SQL combines rows from two or more tables based on a related column between them. A union operation, on the other hand, combines the result sets of two or more SELECT statements into a single result set, but removes any duplicate rows.

50. What is a subquery in SQL and what is it used for?

A subquery in SQL is a SELECT statement that is nested inside another SELECT, INSERT, UPDATE or DELETE statement. Subqueries are used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

51. What is a self-join in SQL and what is it used for?

A self-join in SQL is a regular join, but the table is joined with itself. A self-join is used to compare rows within the same table, such as finding employees who have the same manager or finding products that have the same price.

52. What is a transactional consistency in SQL and how is it ensured?

Transactional consistency in SQL refers to the requirement that data must be in a consistent state before and after a transaction. This consistency is ensured through the use of transactions, which are a sequence of one or more SQL statements that are executed as a single unit of work. Transactions can be rolled back if an error occurs, ensuring that the data remains in a consistent state.

53. What is the difference between a clustered and a non-clustered index in SQL?

A clustered index in SQL determines the physical order of data in a table and only one clustered index can be created for a table. A non-clustered index, on the other hand, provides a fast access path to the rows in a table, but does not affect the physical order of the data.

54. What is a view in SQL and what is it used for?



A view in SQL is a virtual table that is based on the result of a SELECT statement. A view can be used to simplify the complexity of a query, or to hide sensitive information by only exposing the columns and rows that are required.

55. What is a stored procedure in SQL and what is it used for?

A stored procedure in SQL is a precompiled program that can be executed to perform specific tasks. Stored procedures can be used to encapsulate complex business logic, improve performance, and increase security by reducing the amount of SQL code that is executed directly against the database.

56. What is a trigger in SQL and what is it used for?

A trigger in SQL is a special type of stored procedure that is automatically executed in response to an INSERT, UPDATE, or DELETE event on a table. Triggers can be used to enforce business rules, update dependent tables, or log changes to the data.

57. What is a cursor in SQL and what is it used for?

A cursor in SQL is a database object that is used to retrieve a set of rows from a result set, one row at a time. Cursors can be used to process large amounts of data row by row, or to perform complex updates on the data.

58. What is the difference between a primary key and a unique key in SQL?

A primary key in SQL is a unique identifier for each row in a table and is used to enforce referential integrity in a database. A primary key must contain unique values and cannot contain null values. A unique key, on the other hand, is also a unique identifier for each row in a table, but can contain null values.

59. What is a self join in SQL and when would you use it?

A self join in SQL is a regular join, but the table is joined with itself. A self join is used to combine rows from the same table based on related data within the table.

60. What is a subquery in SQL and when would you use it?

A subquery in SQL is a SELECT statement that is nested within another SELECT statement. A subquery can be used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

61. What is a transaction in SQL and why are they used?

A transaction in SQL is a sequence of one or more database operations that are executed as a single unit of work. Transactions are used to ensure that the database remains in a consistent state even if one of the operations fails. Transactions also allow multiple operations to be executed as a single atomic operation, either all or nothing.

62. What is a constraint in SQL and why are they used?

A constraint in SQL is a rule that is used to restrict the values that can be stored in a column or a set of columns in a table. Constraints are used to enforce business rules, maintain data integrity, and ensure that the data stored in the database is consistent and meaningful.

63. What is an index in SQL and why is it used?

An index in SQL is a database object that is used to improve the performance of queries by providing a faster way to look up rows in a table. An index allows the database management system to quickly locate the required data pages in a table without having to scan the entire table.

64. What is a join in SQL and what are the different types of joins?

A join in SQL is a query that combines rows from two or more tables based on a related column between them. The different types of joins are: INNER JOIN, LEFT JOIN (or LEFT OUTER JOIN), RIGHT JOIN (or RIGHT OUTER JOIN), and FULL OUTER JOIN.

65. What is a stored procedure in SQL and what are its benefits?

A stored procedure in SQL is a precompiled collection of SQL statements that can be executed one or more times as a single unit. The benefits of using stored procedures include improved security, improved performance, reduced network traffic, and ease of maintenance.

66. What is a trigger in SQL and why is it used?

A trigger in SQL is a set of instructions that are automatically executed in response to a specific event, such as an INSERT, UPDATE, or DELETE statement. Triggers are used to enforce business rules, maintain data integrity, and perform additional actions, such as auditing or cascading updates, whenever data is modified in the database.

67. What is normalization in SQL and why is it important?

Normalization in SQL is the process of organizing a database into two or more tables to minimize data redundancy and improve data integrity. Normalization is important because it helps to ensure that the data stored in the database is accurate, consistent, and free of anomalies.

68. What is a subquery in SQL and how is it used?

X A subquery in SQL is a query that is nested inside another query and is used to return data that will be used in the main query as a condition. Subqueries can be used in a variety of ways, such as to return data for use in a SELECT statement, to calculate aggregate values, or to return data for use in a WHERE clause.

69. What is the difference between a primary key and a foreign key in SQL?

A primary key in SQL is a unique identifier for a record in a table, used to enforce referential integrity. A foreign key in SQL is a field in one table that refers to the primary key of another table and is used to enforce referential integrity between the two tables.

70. What is a transaction in SQL and what are its properties?

A transaction in SQL is a sequence of one or more database operations that are executed as a single unit of work. The properties of transactions are atomicity, consistency, isolation, and durability (ACID). Atomicity means that either all of the operations in a transaction are executed or none of them are. Consistency means that the database is in a consistent state after a transaction is executed. Isolation means that transactions are executed as if they are the only transactions executing on the database. Durability means that the results of a transaction are permanent, even in the event of a system failure.

71. What is a view in SQL and what is it used for?

A view in SQL is a virtual table that is based on the result of a SELECT statement. A view is used to simplify the data structure of a database by presenting a subset of the data in a table or by combining data from multiple tables into a single virtual table. Views can be used to restrict access to sensitive data, to simplify complex queries, or to provide a common interface for different user groups.

72. What is a cursor in SQL and what is it used for?

A cursor in SQL is a database object that is used to traverse through the rows of a result set one row at a time. Cursors are used to perform row-by-row operations, such as processing each row of a result set for a complex calculation or updating data in a table based on the values of a result set.

73. What is the difference between a left join and a right join in SQL?

A left join in SQL returns all records from the left table (table1), and the matching records from the right table (table2). The result set will contain NULL values for non-matching records from the right table. A right join in SQL returns all records from the right table (table2), and the matching records from the left table (table1). The result set will contain NULL values for non-matching records from the left table.

74. What is a self join in SQL and how is it used?

A self join in SQL is a regular join, but the table is joined with itself. Self joins are used to compare records within a single table, such as finding employees who are managers of other employees in an employees table.

75. What is the difference between a clustered and a non-clustered index in SQL?

A clustered index in SQL is a special type of index that reorders the rows in a table to match the order of the index. A clustered index is the only type of index that can be created on a primary key. A non-clustered index in SQL is a separate object from the table that contains a copy of the data from the columns specified in the index, along with a pointer to the actual row. Non-clustered indexes do not reorder the rows in a table.

76. What is a stored procedure in SQL and what is it used for?

A stored procedure in SQL is a precompiled collection of SQL statements that are stored in the database and can be executed repeatedly. Stored procedures are used to encapsulate a set of operations that are performed on the database, to reduce network traffic between the application and the database, and to improve the security of data operations by encapsulating sensitive operations within the database.

77. What is a trigger in SQL and what is it used for?

A trigger in SQL is a special type of stored procedure that is automatically executed in response to an INSERT, UPDATE, or DELETE statement on a specified table. Triggers are used to enforce business rules, to perform complex calculations, or to maintain summary data.

Scenario-Based SQL Interview Questions

Hands-on problems with sample solutions and SQL snippets. Try writing a solution before expanding the provided answer.

1. What is the syntax for SELECT statement in SQL?

The syntax for SELECT statement in SQL is as follows:

```
SELECT column1, column2, ...  
FROM table_name;
```

2. What is the use of the WHERE clause in SQL?

 The WHERE clause in SQL is used to filter rows from a table based on specific conditions.

3. How to find the second highest salary of an employee in a table?

The following query can be used to find the second highest salary of an employee in a table:

```
SELECT MAX(salary)
FROM employee
WHERE salary NOT IN (SELECT MAX(salary)
FROM employee);
```

4. What is the difference between INNER JOIN and LEFT JOIN in SQL?

INNER JOIN returns only the rows from both tables where there is a match.
LEFT JOIN returns all the rows from the left table and matching rows from the right table. If there is no match, NULL values will be displayed for the right table's columns.

5. How to find the total number of rows in a table?

The following query can be used to find the total number of rows in a table:

```
SELECT COUNT(*)
FROM table_name;
```

6. How to find the average salary of employees in a table?

The following query can be used to find the average salary of employees in a table:

```
SELECT AVG(salary)
FROM employee;
```

7. How to find the minimum and maximum salary of employees in a table?

The following query can be used to find the minimum and maximum salary of employees in a table:

```
SELECT MIN(salary), MAX(salary)
FROM employee;
```

8. How to sort the data in a table in ascending or descending order?

The following query can be used to sort the data in a table in ascending or descending order:

```
SELECT *
FROM table_name ORDER BY column_name;
SELECT *
FROM table_name ORDER BY column_name DESC;
```

9. How to group the data in a table based on a column?

The following query can be used to group the data in a table based on a column:

```
SELECT column1, SUM(column2)
FROM table_name GROUP BY column1;
```

10. How to update data in a table?

The following query can be used to update data in a table:

```
UPDATE table_name SET column1 = value1, column2 = value2, ...
WHERE some_column = some_value;
```

11. How to delete data from a table?

The following query can be used to delete data from a table:

```
DELETE
FROM table_name
WHERE some_column = some_value;
```

12. How to insert data into a table?

The following query can be used to insert data into a table:

```
INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);
```

13. What is the use of the LIKE operator in SQL?

The LIKE operator in SQL is used to search for a specified pattern in a column. The following query can be used to search for a specified pattern in a column:



```
SELECT *
FROM table_name
WHERE column_name LIKE 'pattern';
```

14. What is the use of the IN operator in SQL?

The IN operator in SQL is used to specify multiple values in a WHERE clause. The following query can be used to search for multiple values in a column:

```
SELECT *
FROM table_name
WHERE column_name IN (value1, value2, value3, ...);
```

15. What is the use of the DISTINCT keyword in SQL?

The DISTINCT keyword in SQL is used to return unique values from a SELECT statement. The following query can be used to return unique values from a column:

```
SELECT DISTINCT column_name
FROM table_name;
```

16. What is the use of the BETWEEN operator in SQL?

The BETWEEN operator in SQL is used to select values within a specified range. The following query can be used to select values within a specified range:

```
SELECT *
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

17. What is the use of the ALIAS in SQL?

The ALIAS in SQL is used to give a temporary name to a column or table. The following query can be used to give a temporary name to a column:

```
SELECT column_name AS alias_name
FROM table_name;
The following query can be used to give a temporary name to a table:
SELECT column_name
FROM table_name AS alias_name;
```

18. How to join multiple tables in SQL?

The following query can be used to join multiple tables in SQL:

```
SELECT column1, column2, ...
FROM table1 JOIN table2 ON table1.column = table2.column JOIN table3 ON table2.column = table3.column ...;
```

19. What is the use of the UNION operator in SQL?

The UNION operator in SQL is used to combine the result-set of two or more SELECT statements into a single result-set. The following query can be used to combine the result-set of two SELECT statements into a single result-set:

```
SELECT column_name
FROM table1
UNION
SELECT column_name
FROM table2;
```

20. What is the use of the GROUP BY clause in SQL?

The GROUP BY clause in SQL is used to group rows that have the same values into summary rows, like finding the total count for each group. The following query can be used to group rows based on the values in a column:

```
SELECT column_name, aggregate_function(column_name)
FROM table_name GROUP BY column_name;
```

21. What is the use of the HAVING clause in SQL?

The HAVING clause in SQL is used to restrict the groups of returned rows from the GROUP BY clause. The following query can be used to restrict the groups of returned rows based on aggregate functions:

```
SELECT column_name, aggregate_function(column_name)
FROM table_name GROUP BY column_name HAVING aggregate_function(column_name) operator value;
```

22. What is the use of the LIMIT clause in SQL?

The LIMIT clause in SQL is used to specify the number of records to return in a SELECT statement. The following query can be used to return a specified number of records:

```
SELECT column_name
FROM table_name LIMIT number_of_records;
```

23. What is the use of the INNER JOIN in SQL?

The INNER JOIN in SQL is used to combine rows from two or more tables based on the values of the joined columns, and only returns the rows that have matching values in both tables. The following query can be used to combine rows from two tables based on the values of the joined columns:

```
SELECT column1, column2, ...
FROM table1
INNER JOIN table2 ON table1.column = table2.column;
```

24. What is the use of the LEFT JOIN in SQL?

The LEFT JOIN in SQL is used to combine all rows from the left table and the matching rows from the right table, and returns NULL values for non-matching rows in the right table. The following query can be used to combine all rows from the left table and the matching rows from the right table:

```
SELECT column1, column2, ...
FROM table1
LEFT JOIN table2 ON table1.column = table2.column;
```

25. What is the use of the RIGHT JOIN in SQL?

The RIGHT JOIN in SQL is used to combine all rows from the right table and the matching rows from the left table, and returns NULL values for non-matching rows in the left table. The following query can be used to combine all rows from the right table and the matching rows from the left table:

```
SELECT column1, column2, ...
FROM table1
RIGHT JOIN table2 ON table1.column = table2.column;
```

26. What is the use of the FULL OUTER JOIN in SQL?

The FULL OUTER JOIN in SQL is used to combine all rows from both tables, and returns NULL values for non-matching rows in either the left or right table. The following query can be used to combine all rows from both tables:

```
SELECT column1, column2, ...
FROM table1
FULL OUTER JOIN table2 ON table1.column = table2.column;
```

27. What is the use of the IS NULL operator in SQL?

The IS NULL operator in SQL is used to test for NULL values. The following query can be used to select records with NULL values in a column:

```
SELECT column_name
FROM table_name
WHERE column_name IS NULL;
```

28. What is the use of the IS NOT NULL operator in SQL?

The IS NOT NULL operator in SQL is used to test for non-NULL values. The following query can be used to select records with non-NULL values in a column:

```
SELECT column_name
FROM table_name
WHERE column_name IS NOT NULL;
```

29. What is the use of the IN operator in SQL?

The IN operator in SQL is used to match values against a set of values. The following query can be used to select records with matching values in a column:

```
SELECT column_name
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

30. What is the use of the NOT IN operator in SQL?

The NOT IN operator in SQL is used to exclude values that match a set of values. The following query can be used to select records that do not match the specified values in a column:

```
SELECT column_name
FROM table_name
WHERE column_name NOT IN (value1, value2, ...);
```

31. What is the use of the LIKE operator in SQL?

The LIKE operator in SQL is used to search for a specific pattern in a column. The following query can be used to select records that contain the specified pattern in a column:

```
SELECT column_name
FROM table_name
WHERE column_name LIKE 'pattern';
```

32. What is the use of the NOT LIKE operator in SQL?

The NOT LIKE operator in SQL is used to exclude records that contain a specific pattern in a column. The following query can be used to select records that do not contain the specified pattern in a column:

```
SELECT column_name
FROM table_name
WHERE column_name NOT LIKE 'pattern';
```

33. What is the use of the COUNT function in SQL?

The COUNT function in SQL is used to count the number of rows in a column. The following query can be used to count the number of rows in a column:

```
SELECT COUNT(column_name)
FROM table_name;
```

34. What is the use of the SUM function in SQL?

The SUM function in SQL is used to calculate the sum of values in a column. The following query can be used to calculate the sum of values in a column: SELECT SUM(column_name) FROM table_name;

35. What is the use of the AVG function in SQL?

The AVG function in SQL is used to calculate the average of values in a column. The following query can be used to calculate the average of values in a column:

```
SELECT AVG(column_name)
FROM table_name;
```

36. What is the use of the MIN function in SQL?

The MIN function in SQL is used to find the minimum value in a column. The following query can be used to find the minimum value in a column:

```
SELECT MIN(column_name)
FROM table_name;
```

37. What is the use of the MAX function in SQL?

The MAX function in SQL is used to find the maximum value in a column. The following query can be used to find the maximum value in a column:

```
SELECT MAX(column_name)
FROM table_name;
```

38. What is the use of the GROUP BY clause in SQL?

The GROUP BY clause in SQL is used to group rows that have the same values into summary rows, like finding the total number of items sold by each salesperson. The following query can be used to group rows by one or more columns:

```
SELECT column1, SUM(column2)
FROM table_name GROUP BY column1;
```

39. What is the use of the DISTINCT keyword in SQL?

The DISTINCT keyword in SQL is used to return only unique values from a column in a table. The following query can be used to return unique values from a column:

```
SELECT DISTINCT column_name
FROM table_name;
```

40. What is the use of the INNER JOIN clause in SQL?

The INNER JOIN clause in SQL is used to combine rows from two or more tables based on a related column between them. The following query can be used to combine rows from two tables based on a related column:

```
SELECT column1, column2
FROM table1
INNER JOIN table2 ON table1.column = table2.column;
```

41. What is the use of the LEFT JOIN clause in SQL?



The LEFT JOIN clause in SQL is used to combine all rows from the left table and matching rows from the right table. If there's no match, NULL values will be displayed for right table's columns. The following query can be used to combine all rows from the left table and matching rows from the right table:

```
SELECT column1, column2
FROM table1
LEFT JOIN table2 ON table1.column = table2.column;
```

42. What is the use of the RIGHT JOIN clause in SQL?

The RIGHT JOIN clause in SQL is used to combine all rows from the right table and matching rows from the left table. If there's no match, NULL values will be displayed for left table's columns. The following query can be used to combine all rows from the right table and matching rows from the left table:

```
SELECT column1, column2
FROM table1
RIGHT JOIN table2 ON table1.column = table2.column;
```

43. What is the use of the FULL OUTER JOIN clause in SQL?

The FULL OUTER JOIN clause in SQL is used to combine all rows from both tables. If there's no match, NULL values will be displayed for the columns of the table without a match. The following query can be used to combine all rows from both tables:

```
SELECT column1, column2
FROM table1
FULL OUTER JOIN table2 ON table1.column = table2.column;
```

44. How can you retrieve the first N records from a table in SQL?

You can retrieve the first N records from a table in SQL using the LIMIT keyword in the SELECT statement. For example, the following query will retrieve the first 5 records from the 'employees' table:

```
SELECT *
FROM employees LIMIT 5;
```

45. How can you group data in SQL?

You can group data in SQL using the GROUP BY clause. The GROUP BY clause groups data based on one or more columns in a table. For example, the following query will group data in the 'employees' table by 'department' column:

```
SELECT department, SUM(salary)
FROM employees GROUP BY department;
```

46. How can you sort data in SQL?

You can sort data in SQL using the ORDER BY clause. The ORDER BY clause sorts data in ascending or descending order based on one or more columns in a table. For example, the following query will sort data in the 'employees' table in ascending order based on the 'name' column:

```
SELECT *
FROM employees ORDER BY name ASC;
```

47. What is the use of the HAVING clause in SQL?

The HAVING clause in SQL is used to filter data after it has been grouped by the GROUP BY clause. The HAVING clause operates on the results of the grouping rather than on individual rows. For example, the following query will group data in the 'employees' table by 'department' column and returns only those departments with a total salary greater than 10,000:

```
SELECT department, SUM(salary)
FROM employees GROUP BY department HAVING SUM(salary) > 10000;
```

48. What is the use of the COUNT function in SQL?

The COUNT function in SQL is used to return the number of rows in a table or the number of rows that meet a specified condition. For example, the following query will return the number of rows in the 'employees' table:

```
SELECT COUNT(*)
FROM employees;
```

49. What is the use of the AVG function in SQL?

The AVG function in SQL is used to return the average value of a numeric column in a table. For example, the following query will return the average salary of employees in the 'employees' table:



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

50. What is the use of the MAX function in SQL?

The MAX function in SQL is used to return the maximum value of a numeric column in a table. For example, the following query will return the maximum salary of employees in the 'employees' table:

```
SELECT MAX(salary)
FROM employees;
```

51. What is the use of the MIN function in SQL?

The MIN function in SQL is used to return the minimum value of a numeric column in a table. For example, the following query will return the minimum salary of employees in the 'employees' table:

```
SELECT MIN(salary)
FROM employees;
```

52. What is the use of the SUM function in SQL?

The SUM function in SQL is used to return the sum of values of a numeric column in a table. For example, the following query will return the total salary of employees in the 'employees' table:

```
SELECT SUM(salary)
FROM employees;
```

53. What is a subquery in SQL?

A subquery in SQL is a query nested inside another query. The result of the subquery is used in the main query. For example, the following query will retrieve the name of the employee with the maximum salary in the 'employees' table: SELECT name FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);

54. What is the use of the LIKE operator in SQL?

The LIKE operator in SQL is used to search for a specified pattern in a column. It uses wildcard characters such as '%' and '_' to match any number of characters or a single character respectively. For example, the following query will retrieve all employees whose name starts with 'J':

```
SELECT *
FROM employees
WHERE name LIKE 'J%';
```

55. What is the use of the IN operator in SQL?

The IN operator in SQL is used to determine if a value is present in a list of specified values. For example, the following query will retrieve all employees who work in departments with department id 1, 2, or 3:

```
SELECT *
FROM employees
WHERE department_id IN (1, 2, 3);
```

56. What is the use of the NOT IN operator in SQL?

The NOT IN operator in SQL is used to determine if a value is not present in a list of specified values. For example, the following query will retrieve all employees who do not work in departments with department id 1, 2, or 3:

```
SELECT *
FROM employees
WHERE department_id NOT IN (1, 2, 3);
```

57. What is the use of the BETWEEN operator in SQL?

The BETWEEN operator in SQL is used to determine if a value is within a specified range. For example, the following query will retrieve all employees with a salary between 50000 and 80000:

```
SELECT *
FROM employees
WHERE salary BETWEEN 50000 AND 80000;
```

58. What is the use of the NOT BETWEEN operator in SQL?

The NOT BETWEEN operator in SQL is used to determine if a value is not within a specified range. For example, the following query will retrieve all employees with a salary not between 50000 and 80000:

```
SELECT *
FROM employees
WHERE salary NOT BETWEEN 50000 AND 80000;
```

59. What is the use of the AVG function in SQL?



The AVG function in SQL is used to calculate the average value of a set of numbers. For example, the following query will retrieve the average salary of all employees:

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

60. What is the use of the SUM function in SQL?

The SUM function in SQL is used to calculate the sum of a set of numbers. For example, the following query will retrieve the total salary of all employees:

```
SELECT  SUM(salary)
FROM    employees;
```

61. What is the use of the MIN function in SQL?

The MIN function in SQL is used to find the minimum value in a set of values. For example, the following query will retrieve the minimum salary of all employees:

```
SELECT  MIN(salary)
FROM    employees;
```

62. What is the use of the MAX function in SQL?

The MAX function in SQL is used to find the maximum value in a set of values. For example, the following query will retrieve the maximum salary of all employees:

```
SELECT  MAX(salary)
FROM    employees;
```

63. What is the use of the COUNT function in SQL?

The COUNT function in SQL is used to count the number of rows in a table. For example, the following query will retrieve the number of employees in the employees table:

```
SELECT  COUNT(*)
FROM    employees;
```

64. What is the use of the GROUP BY clause in SQL?

The GROUP BY clause in SQL is used to group rows with identical values into summary rows. For example, the following query will group employees by department and retrieve the total salary of each department:

```
SELECT  department, SUM(salary)
FROM    employees GROUP BY department;
```

65. What is the use of the DISTINCT keyword in SQL?

The DISTINCT keyword in SQL is used to remove duplicates from the results of a query. For example, the following query will retrieve a list of all unique departments:

```
SELECT  DISTINCT department
FROM    employees;
```

66. What is the use of the UNION operator in SQL?

The UNION operator in SQL is used to combine the results of two or more SELECT statements into a single result set, removing duplicates in the process. For example, the following query will retrieve a list of all unique departments from two tables, employees and departments:

```
SELECT  department
FROM    employees UNION SELECT  department_name
FROM    departments;
```

67. What is the use of the INTERSECT operator in SQL?

The INTERSECT operator in SQL is used to combine the results of two SELECT statements and retrieve only the rows that appear in both results. For example, the following query will retrieve a list of departments that are common to both the employees and departments tables:

```
SELECT  department
FROM    employees INTERSECT SELECT  department_name
FROM    departments;
```

68. What is the use of the EXCEPT operator in SQL?

The EXCEPT operator in SQL is used to combine the results of two SELECT statements and retrieve only the rows that appear in the first result but not in the second. For example, the following query will retrieve a list of departments that are in the employees table but not in the departments table:



```
SELECT department
FROM employees EXCEPT SELECT department_name
FROM departments;
```

69. What is the use of the INNER JOIN clause in SQL?

The INNER JOIN clause in SQL is used to combine rows from two or more tables based on a related column between them. For example, the following query will retrieve a list of all employees and their departments:

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments ON employees.department_id = departments.id;
```

70. What is the use of the LEFT JOIN clause in SQL?

The LEFT JOIN clause in SQL is used to combine rows from two or more tables based on a related column between them, including all rows from the left table and matching rows from the right table. If there is no match, the result will contain NULL values for the right table columns. For example, the following query will retrieve a list of all employees and their departments, including employees who do not have a department:

```
SELECT employees.name, departments.department_name
FROM employees
LEFT JOIN departments ON employees.department_id = departments.id;
```

71. What is the use of the RIGHT JOIN clause in SQL?

The RIGHT JOIN clause in SQL is used to combine rows from two or more tables based on a related column between them, including all rows from the right table and matching rows from the left table. If there is no match, the result will contain NULL values for the left table columns. For example, the following query will retrieve a list of all departments and their employees, including departments that do not have any employees:

```
SELECT employees.name, departments.department_name
FROM employees
RIGHT JOIN departments ON employees.department_id = departments.id;
```

72. What is the use of the FULL OUTER JOIN clause in SQL?

The FULL OUTER JOIN clause in SQL is used to combine rows from two or more tables based on a related column between them, including all rows from both tables and matching rows. If there is no match, the result will contain NULL values for the missing columns. For example, the following query will retrieve a list of all employees and their departments, including employees who do not have a department and departments that do not have any employees:

```
SELECT employees.name, departments.department_name
FROM employees
FULL OUTER JOIN departments ON employees.department_id = departments.id;
```

73. How to retrieve the top N rows in SQL?

To retrieve the top N rows in SQL, you can use the LIMIT clause. For example, the following query will retrieve the top 10 employees ordered by their salary in descending order:

```
SELECT *
FROM employees ORDER BY salary DESC LIMIT 10;
```

74. How to update multiple rows in SQL?

To update multiple rows in SQL, you can use the UPDATE statement in conjunction with a WHERE clause that specifies the conditions for which rows should be updated. For example, the following query will increase the salary of all employees by 10%:

```
UPDATE employees SET salary = salary * 1.10
WHERE department_id = 1;
```

75. How to delete multiple rows in SQL?

To delete multiple rows in SQL, you can use the DELETE statement in conjunction with a WHERE clause that specifies the conditions for which rows should be deleted. For example, the following query will delete all employees with a salary less than \$50,000:

```
DELETE
FROM employees
WHERE salary < 50000;
```

76. How to create a new table in SQL?

X To create a new table in SQL, you can use the CREATE TABLE statement along with the desired column names and data types. For example, the following query will create a new table named departments with three columns: id, department_name, and location:

```
CREATE TABLE departments (id INT PRIMARY KEY, department_name VARCHAR(50), location VARCHAR(50));
```

77. **How to alter an existing table in SQL?**

To alter an existing table in SQL, you can use the ALTER TABLE statement. For example, the following query will add a new column named budget to the departments table:

```
ALTER TABLE departments ADD COLUMN budget INT;
```

78. **How to drop a table in SQL?**

To drop a table in SQL, you can use the DROP TABLE statement. For example, the following query will drop the departments table:

```
DROP TABLE departments;
```

79. **How to insert data into a table in SQL?**

To insert data into a table in SQL, you can use the INSERT INTO statement along with the values for each column. For example, the following query will insert a new department into the departments table:

```
INSERT INTO departments (id, department_name, location) VALUES (1, 'Sales', 'New York');
```

80. **How to join two tables in SQL?**

To join two tables in SQL, you can use the JOIN clause. For example, the following query will retrieve the names of all employees and the name of their department by joining the employees and departments tables on the department_id column:

```
SELECT employees.name, departments.department_name
FROM employees JOIN departments ON employees.department_id = departments.id;
```

81. **How to group data in SQL?**

To group data in SQL, you can use the GROUP BY clause along with aggregate functions such as SUM, AVG, MIN, MAX, or COUNT. For example, the following query will group employees by department and retrieve the average salary for each department:

```
SELECT department_id, AVG(salary)
FROM employees GROUP BY department_id;
```

82. **How to use a subquery in SQL?**

To use a subquery in SQL, you can nest a SELECT statement within another SELECT statement. For example, the following query will retrieve the names of all employees who earn more than the average salary for their department:

```
SELECT name
FROM employees
WHERE salary > (SELECT AVG(salary)
FROM employees
WHERE department_id = employees.department_id);
```

83. **How to use a case statement in SQL?**

To use a case statement in SQL, you can use the CASE statement along with the WHEN, THEN, and END clauses. For example, the following query will retrieve the names of all employees along with a salary grade based on their salary:

```
SELECT name, CASE WHEN salary <= 50000 THEN 'A' WHEN salary <= 100000 THEN 'B' ELSE 'C' END AS salary_grade
FROM employees;
```

84. **How to use a self join in SQL?**

To use a self join in SQL, you join a table to itself by using an alias for the same table. For example, the following query will retrieve the names of all employees who have the same manager:

```
SELECT e1.name, e2.name
FROM employees e1 JOIN employees e2 ON e1.manager_id = e2.id
WHERE e1.id != e2.id;
```

85. **How to use a left join in SQL?**

To use a left join in SQL, you can use the LEFT JOIN clause to include all the rows from the left table and matching rows from the right table. If there is no match, the right side's columns will be filled with NULL values. For example, the following query will retrieve the names of all departments along with the number of employees in each department:

```
SELECT departments.department_name, COUNT(employees.id)
FROM departments
LEFT JOIN employees ON departments.id = employees.department_id GROUP BY departments.department_name;
```

86. **How to use a right join in SQL?**

⌘ To use a right join in SQL, you can use the RIGHT JOIN clause to include all the rows from the right table and matching rows from the left table. If there is no match, the left side's columns will be filled with NULL values. For example, the following query will retrieve the names of all employees along with the names of the departments they work in:

```
SELECT employees.name, departments.department_name
FROM employees
RIGHT JOIN departments ON employees.department_id = departments.id;
```

87. **How to use a full outer join in SQL?**

To use a full outer join in SQL, you can use the FULL OUTER JOIN clause to include all the rows from both tables. If there is no match, the non-matching side's columns will be filled with NULL values. For example, the following query will retrieve the names of all employees along with the names of the departments they work in, including departments that have no employees:

```
SELECT employees.name, departments.department_name
FROM employees
FULL OUTER JOIN departments ON employees.department_id = departments.id;
```

88. **How to use a cross join in SQL?**

To use a cross join in SQL, you can use the CROSS JOIN clause to return all possible combinations of rows from the two tables. For example, the following query will retrieve the names of all employees along with the names of all departments:

```
SELECT employees.name, departments.department_name
FROM employees CROSS JOIN departments;
```

89. **How to use the DISTINCT keyword in SQL?**

To use the DISTINCT keyword in SQL, you can include it in the SELECT clause to remove duplicates from the result set. For example, the following query will retrieve the names of all departments, with no duplicates:

```
SELECT DISTINCT department_name
FROM departments;
```

90. **How to use the GROUP BY clause in SQL?**

To use the GROUP BY clause in SQL, you can include it in the query to group rows that have the same values in specified columns. You can then use aggregate functions such as COUNT, SUM, AVG, MIN, MAX, etc. on the grouped data. For example, the following query will retrieve the names of all departments along with the number of employees in each department:

```
SELECT department_name, COUNT(id)
FROM employees GROUP BY department_name;
```

91. **How to use the HAVING clause in SQL?**

To use the HAVING clause in SQL, you can include it in the query after the GROUP BY clause to filter groups based on a specified condition. For example, the following query will retrieve the names of all departments with more than 5 employees:

```
SELECT department_name, COUNT(id)
FROM employees GROUP BY department_name HAVING COUNT(id) > 5;
```

92. **How to use the UNION operator in SQL?**

To use the UNION operator in SQL, you can combine the result sets of two or more SELECT statements into a single result set, with duplicates removed. The number of columns and the data types of the columns must be the same in all SELECT statements. For example, the following query will retrieve the names of all employees from the employees table and all department names from the departments table:

```
SELECT name
FROM employees UNION SELECT department_name
FROM departments;
```

93. **How to use the INTERSECT operator in SQL?**

To use the INTERSECT operator in SQL, you can combine the result sets of two or more SELECT statements into a single result set, with only the rows that appear in all SELECT statements included. The number of columns and the data types of the columns must be the same in all SELECT statements. For example, the following query will retrieve the names of employees who work in both the 'Sales' and 'Marketing' departments:

```
SELECT name
FROM employees
WHERE department_name = 'Sales' INTERSECT SELECT name
FROM employees
WHERE department_name = 'Marketing';
```

94. **How to use the MINUS operator in SQL?**

X To use the MINUS operator in SQL, you can subtract the result set of one SELECT statement from the result set of another SELECT statement. The number of columns and the data types of the columns must be the same in both SELECT statements. For example, the following query will retrieve the names of employees who work in the 'Sales' department but not in the 'Marketing' department:

```
SELECT name
FROM employees
WHERE department_name = 'Sales' MINUS SELECT name
FROM employees
WHERE department_name = 'Marketing';
```

95. **How to use the JOIN clause in SQL?**

To use the JOIN clause in SQL, you can combine the rows from two or more tables based on a related column between them. There are several types of JOINS such as
INNER JOIN,
LEFT JOIN,
RIGHT JOIN,
FULL OUTER JOIN, etc. For example, the following query will retrieve the names of employees along with the names of their departments:

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments ON employees.department_id = departments.id;
```

96. **How to use the INNER JOIN clause in SQL?**

To use the
INNER JOIN clause in SQL, you can combine only the rows from two or more tables that have matching values in the related columns. For example, the following query will retrieve the names of employees along with the names of their departments:

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments ON employees.department_id = departments.id;
```

97. **How to use the LEFT JOIN clause in SQL?**

To use the
LEFT JOIN clause in SQL, you can combine all the rows from the left table (the first table in the query) and the matching rows from the right table (the second table in the query). If there is no matching row in the right table, NULL values will be displayed for the columns from the right table. For example, the following query will retrieve the names of all departments and the names of employees in each department:

```
SELECT departments.department_name, employees.name
FROM departments
LEFT JOIN employees ON departments.id = employees.department_id;
```

98. **How to use the RIGHT JOIN clause in SQL?**

To use the
RIGHT JOIN clause in SQL, you can combine all the rows from the right table (the second table in the query) and the matching rows from the left table (the first table in the query). If there is no matching row in the left table, NULL values will be displayed for the columns from the left table. For example, the following query will retrieve the names of all employees and the names of departments in each employee works:

```
SELECT employees.name, departments.department_name
FROM employees
RIGHT JOIN departments ON employees.department_id = departments.id;
```

Tip: When explaining queries, mention indexes used, estimated row counts, and trade-offs (e.g., window vs. subquery) for performance clarity.

SEARCH

Search topics...

Go

INTERVIEW TOPICS

[Java](#)

Q&A

[Java 8](#)

Q&A

[Spring](#)

Q&A

Hibernate	Q&A
React JS	Q&A
SQL	Q&A
MongoDB	Q&A
JavaScript	Q&A
Kafka	Q&A
Design Patterns	Q&A
Microservices	Q&A
REST API	Q&A
Behavioral	Q&A

QUICK ACTIONS

- Start System Design
- Practice Coding
- Earn a Certificate

RECOMMENDED READING

- [Learn Java 17 Programming](#)
- [Learn JavaScript Quickly](#)
- [SQL QuickStart Guide](#)
- [System Design Interview](#)

[See all →](#)

SPONSORED

Ads help keep content free.

NEED HELP?

Have a topic request or correction? We'd love to hear from you.

Contact Us



THANK YOU!

Your comments and complements are always the inspiration that keeps us motivated.

ABOUT

[PROJECTS](#)

[ABOUT US](#)

[BLOG](#)

[AWARDS](#)

ADDRESS

 Sterling, Virginia, 20166, US

 info@techventurellc.com

 641 451 3463

 641 451 3463

FOLLOW US

   

© 2022 Copyright: techventurellc.com