SQL Module Curriculum

algonex.co.in
91 - 9959789424
Marathahalli,
Bangalore Karnataka

CURRICULUM

Chapter 1: Introduction to Databases and SQL

- Understanding Databases
- What is SQL? Relational Databases
- The History of SQL
- Setting Up SQL Environment

Chapter 2: SQL Basics

- SQL Syntax SELECT Statement
- Filtering Data with WHERE
- Sorting Data with ORDER BY
- Filtering and Sorting Data Exercises

Chapter 3: Data Manipulation

- INSERT, UPDATE, and DELETE Statements
- Modifying Data Transaction Management
- Data Manipulation Exercises

Chapter 4: Retrieving Data with SQL

- Retrieving Data from a Single Table
- DISTINCT Keyword Limiting Results with
- LIMIT Retrieving Data from Multiple Tables
- (JOINs) Retrieving Data Exercises

Chapter 5: Data Filtering and Sorting

- The WHERE Clause
- Logical Operators
- Sorting Data with ORDER BY Advanced Filtering and Sorting Exercises

Chapter 6: Aggregating Data

- Aggregate Functions (COUNT, SUM, AVG, MAX, MIN)
- GROUP BY Clause, HAVING Clause
- Aggregating Data Exercises

Chapter 7: Subqueries

- Subqueries in WHERE Clause
- Subqueries in SELECT Clause
- Correlated Subqueries
- Subqueries Exercises

Chapter 8: Data Modification and Transactions

- Transactions in SQL
- COMMIT and ROLLBACK
- SAVEPOINT
- Transaction Exercises

Chapter 9: Working with Dates and Times

- Date and Time Data Types Date
- and Time Functions Date and
- Time Exercises

Chapter 10: String Manipulation

- String Functions
- Concatenation
- String Manipulation Exercises

Chapter 11: Views and Indexes

- Creating and Using Views
- Indexes and Performance
- View and Index Exercises

Chapter 12: Stored Procedures and Functions

- Creating and Calling Stored Procedures
- Creating and Calling User-Defined Functions
- Stored Procedures and Functions Exercises

Chapter 13: Advanced Query Techniques

- Common Table Expressions (CTEs)
- Window Functions
- Pivoting Data
- Advanced Query Techniques Exercises

Chapter 14: Security and Permissions

- User Authentication and Authorization
- GRANT and REVOKE Statements
- Security Best Practices
- Security and Permissions Exercises

Chapter 15: Database Design and Normalization

- Basics of Database Design
- Normalization
- Denormalization
- Database Design and Normalization Exercises

SQL Interview Pattern

Made By:

algonex.co.in

91-9959789424

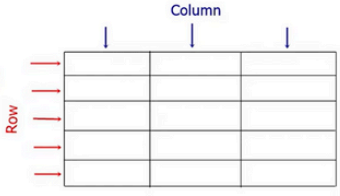
Marathahalli,Bangalore
Karnataka -560037

Database Management System (or DBMS)

Software that enables users to interact with databases

an interface to define, create, and manage databases

MySQL PostgreSQL ORACLE Microsoft SQL Server SQLite



Relational Database

organizes data into tables with **rows** and **columns**

Table → Entity

Row → Record
or an instance of that entity

Column → Attributes
or properties of that entity

- To create Database

```
CREATE DATABASE databasename;
```

- To create Table in a database

```
CREATE TABLE table_name (
column1 datatype,
column2 datatype,
....
);
```

CRUD Operations

Create is used to insert new records into a table
Read is used to retrieve data from a table
Update is used to modify existing records
Delete is used to remove records from a table

SQL Queries

commands or instructions you give to a database to **ask for, add, modify, or delete information**

SELECT,
INSERT,
UPDATE,
DELETE, and
JOIN

Transactions

(ALL succeed or ALL fail)

sequences of one or more SQL operations treated as a **single** unit

```
BEGIN TRANSACTION;
-- SQL statements (INSERT, UPDATE, DELETE, etc.) go here
COMMIT; -- to save changes permanently
ROLLBACK; -- to undo the changes
```

Reliability & correctness

Normalization

organizing data in a database to eliminate redundancy and improve data integrity

defining relationships between them

SQL Interview Pattern

Made By:

algonex.co.in

91-9959789424

Marathahalli,Bangalore
Karnataka -560037

Advanced SQL Interview Questions from Top Tech Companies

1. Query Optimization

- How would you optimize a slow SQL query? What steps do you take to identify bottlenecks?
- How does indexing affect query performance? When can too many indexes be a problem?

2. Joins and Aggregations

- Write a query to fetch the top 3 highest salaries from an employees table.
- Explain the difference between INNER JOIN, LEFT JOIN, and FULL OUTER JOIN with examples.
- How do you join three or more tables in a single query?

3. Window Functions

- What are window functions? Write a query to assign a rank to employees based on their salary within each department.
- What is the difference between RANK() and DENSE_RANK()?

4. Subqueries and CTEs

- What is a Common Table Expression (CTE), and how does it differ from a subquery?
- Write a query to find employees who earn more than the average salary in their department (correlated subquery).

5. Data Transformation and Pivoting

- Write a query to pivot table rows to columns (e.g., monthly sales as columns).
- How do you handle NULLs in aggregations?

6. Indexes and Performance

- What is the difference between a clustered and a non-clustered index?
- When should you avoid using indexes?

7. Transactions and ACID Properties

- What are the ACID properties in SQL transactions? Give an example of each.
- How do you avoid deadlocks in SQL?

8. Views and Materialized Views

- What is a view? When would you use a materialized view instead of a regular view?

100 SQL Interview QnA

Made By:
algonex.co.in
91-9959789424
Marathahalli,Bangalore
Karnataka -560037

1. What is SQL?
   a. SQL (Structured Query Language) is a programming language used for managing relational databases. It allows users to store, manipulate, and retrieve data from databases.
2. What are the different types of SQL statements?
   a. SQL statements can be categorized into three types:
      i. Data Definition Language (DDL): Used for creating, altering, and dropping database objects.
      ii. Data Manipulation Language (DML): Used for querying, inserting, updating, and deleting data.
      iii. Data Control Language (DCL): Used for controlling access to the database, granting or revoking privileges.
3. What is a primary key?
   a. A primary key is a column or a set of columns that uniquely identifies each record in a table. It ensures data integrity and allows efficient retrieval of data.
4. What is a foreign key?
   a. A foreign key is a column or a set of columns in a table that refers to the primary key of another table. It establishes a relationship between the two tables.
5. What is a composite key?
   a. A composite key is a primary key composed of two or more columns. Together, these columns uniquely identify each record in a table.
6. What is the difference between DELETE and TRUNCATE?
   a. DELETE is a DML statement used to remove specific rows from a table, whereas TRUNCATE is a DDL statement used to remove all rows from a table, effectively resetting the table
7. What is a subquery?
   a. A subquery is a query nested within another query. It can be used to retrieve data from one table based on values from another table or perform complex calculations.
8. What is the difference between a subquery and a join?

a. A subquery is a query nested within another query, whereas a join is used to combine rows from two or more tables based on related columns.

9. What is a self-join?

a. A self-join is a join operation where a table is joined with itself. It is useful when you want to compare rows within the same table.

10. What are the different types of JOIN operations?

a. The different types of JOIN operations are:

i. INNER JOIN: Returns only the matching rows from both tables.

ii. LEFT JOIN: Returns all rows from the left table and matching rows from the right table.

iii. RIGHT JOIN: Returns all rows from the right table and matching rows from the left table.

iv. FULL JOIN: Returns all rows from both tables.

11. What is normalization in SQL?

a. Normalization is the process of organizing data in a database to eliminate redundancy and dependency issues. It involves splitting tables into smaller, more manageable entities.

12. What are the different normal forms in database normalization?

a. The different normal forms are:

i. First Normal Form (1NF): Eliminates duplicate rows and ensures atomicity of values.

ii. Second Normal Form (2NF): Ensures that each non-key column depends on the entire primary key.

iii. Third Normal Form (3NF): Ensures that each non-key column depends only on the primary key and not on other non-key columns.

iv. Fourth Normal Form (4NF): Eliminates multi-valued dependencies.

v. Fifth Normal Form (5NF): Eliminates join dependencies.

13. What is an index?

a. An index is a database structure that improves the speed of data retrieval operations operations on database tables. It allows faster searching, sorting, and filtering of data.

14. What is a clustered index?

a. A clustered index determines the physical order of data in a table. Each table can have only one clustered index, and it is generally created on the primary key column(s).

15. What is a non-clustered index?

a. A non-clustered index is a separate structure from the table that contains a sorted list of selected columns. It enhances the performance of searching and filtering operations.

- **Real-World Query Example (Google) - Don't worry try it**

Given a table employee_managers(employee_id, employee_name, manager_id, manager_name), write a query to find employees who have more than one manager

Query to find employees with more than one manager:

We have a table:

```
employee_managers(
  employee_id,
  employee_name,
  manager_id,
  manager_name
)
```

```
SELECT
   employee_id,
   employee_name,
   COUNT(DISTINCT manager_id) AS manager_count
FROM
   employee_managers
GROUP BY
   employee_id, employee_name
HAVING
   COUNT(DISTINCT manager_id) > 1;
```

- **Complex Query Example (Amazon/Meta)**

Between two dates, find the customer with the highest overall order cost. If a customer placed multiple orders on the same day, calculate the order expenses daily. Output their name, total cost, and date.

We want:

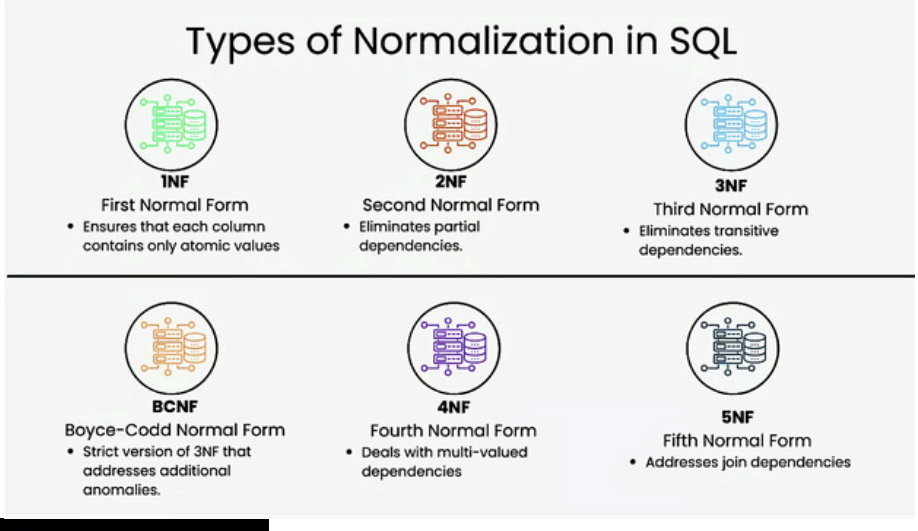
- Between two given dates
- Find the customer with the highest overall order cost
- If customer placed multiple orders on the same day, we must sum them at daily level first
- Output → customer_name, total_cost, order_date

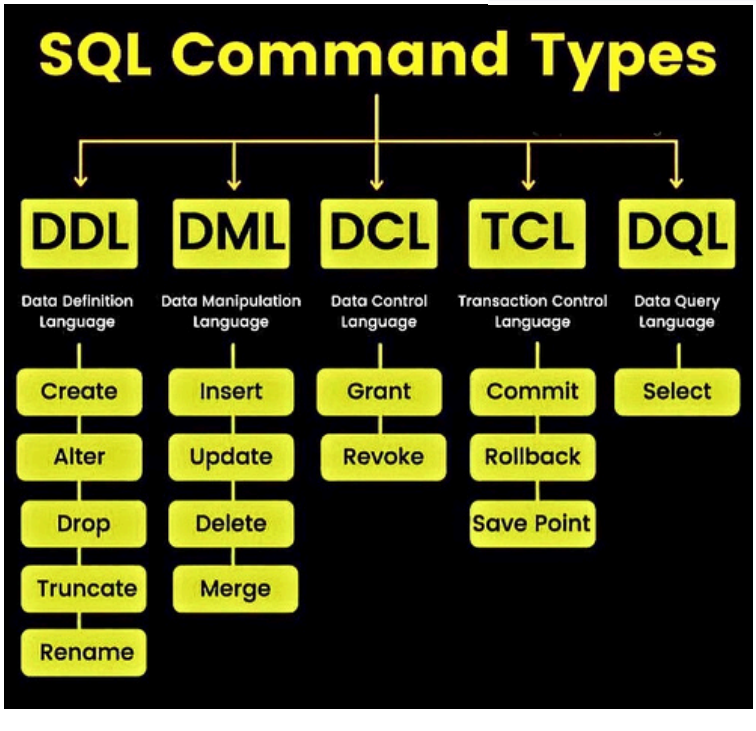
```
WITH daily_totals AS (
   SELECT
      o.customer_id,
      c.customer_name,
      o.order_date,
      SUM(o.order_cost) AS daily_total
   FROM
      orders o
   JOIN
      customers c ON o.customer_id = c.customer_id
   WHERE
      o.order_date BETWEEN '2025-01-01' AND '2025-01-31'
   GROUP BY
      o.customer_id, c.customer_name, o.order_date
),
```

```
ranked_customers AS (
 SELECT
 customer_id,
 customer_name,
 order_date,
 daily_total,
 RANK()
 OVER (ORDER BY daily_total DESC) AS rnk
 FROM
 daily_totals
)
SELECT
 customer_name,
 daily_total AS total_cost,
 order_date
FROM
 ranked_customers
WHERE
 rnk = 1;
```

Explanation:

1. **daily_totals CTE**
   - Groups orders by customer and date, summing their costs.
   - Ensures multiple same-day orders are combined into one daily expense.
2. **ranked_customers CTE**
   - Applies RANK() (or ROW_NUMBER() if you want a single winner even in ties)
   - Orders results by daily_total (highest first)
3. **Final SELECT**
   - Returns only the top customer(s) with max spend





16. **What is the difference between a primary key and a unique key?**
    a. A primary key is a column or a set of columns that uniquely identifies each record in a table and cannot contain NULL values. A unique key, on the other hand, allows NULL values and enforces uniqueness but does not automatically define the primary identifier of a table.
17. **What is ACID in database transactions?**
    a. ACID stands for Atomicity, Consistency, Isolation, and Durability. It is a set of properties that ensure reliability and integrity in database transactions.
18. **What is the difference between UNION and UNION ALL?**
    a. UNION combines the result sets of two or more SELECT statements and removes duplicates, whereas UNION ALL combines the result sets without removing duplicates.
19. **What is a view?**
    a. A view is a virtual table derived from one or more tables. It does not store data but provides a way to present data in a customized or simplified manner.
20. **What is a stored procedure?**
    a. A stored procedure is a precompiled set of SQL statements that performs a specific task. It can be called and executed multiple times with different parameters.
21. **What is a trigger?**
    a. A trigger is a set of SQL statements that are automatically executed in response to a specific event, such as INSERT, UPDATE, or DELETE operations on a table.
22. **What is a transaction?**
    a. A transaction is a logical unit of work that consists of one or more database operations. It ensures that all operations within the transaction are treated as a single unit, either all succeeding or all failing.
23. **What is a deadlock?**
    a. A deadlock is a situation where two or more transactions are unable to proceed because each is waiting for a resource held by another transaction. This can result in a perpetual wait state.
24. **What is the difference between CHAR and VARCHAR data types?**
    a. CHAR is a fixed-length character data type that stores a specific number of characters, while VARCHAR is a variable-length character data type that stores a varying number of characters.
25. **What is the difference between a function and a stored procedure?**
    a. A function returns a value and can be used in SQL statements, whereas a stored procedure does not return a value directly but can perform various actions.

26. **What is the difference between GROUP BY and HAVING clauses?**
    a. GROUP BY is used to group rows based on one or more columns, while HAVING is used to filter grouped rows based on specific conditions.
27. **What is the difference between a database and a schema?**
    a. A database is a collection of related data that is stored and organized. A schema, on the other hand, is a logical container within a database that holds objects like tables, views, and procedures.
28. **What is a data warehouse?**
    a. A data warehouse is a large repository of data collected from various sources, structured and organized to support business intelligence and reporting.
29. **What is the difference between OLTP and OLAP?**
    a. OLTP (Online Transaction Processing) is used for day-to-day transactional operations and focuses on real-time processing. OLAP (Online Analytical Processing) is used for complex analytical queries and focuses on historical data analysis.
30. **What is a correlated subquery?**
    a. A correlated subquery is a subquery that references columns from the outer query. It is executed for each row of the outer query, making it dependent on the outer query's results.
31. **What is the difference between a temporary table and a table variable?**
    a. A temporary table is a physical table that is created and used temporarily within a session or a specific scope, whereas a table variable is a variable with a structure similar to a table and exists only within the scope of a user-defined function or a stored procedure.
32. **What is the difference between UNION and JOIN?**
    a. UNION combines rows from two or more tables vertically, while JOIN combines columns from two or more tables horizontally based on related columns.
33. **What is the difference between WHERE and HAVING clauses?**
    a. WHERE is used to filter rows before grouping in a query, while HAVING is used to filter grouped rows after grouping.
34. **What is the difference between a database and a data warehouse?**
    a. A database is a collection of related data organized for transactional purposes, while a data warehouse is a large repository of data organized for analytical purposes.
35. **What is the difference between a primary key and a candidate key?**
    a. A candidate key is a column or a set of columns that can uniquely identify each record in a table. A primary key is a chosen candidate key that becomes the main identifier for the table.

36. What is the difference between a schema and a database?

a. A database is a collection of related data, while a schema is a logical container within a database that holds objects like tables, views, and procedures.

37. What is a self-join?

a. A self-join is a join operation where a table is joined with itself. It is used when you want to compare rows within the same table.

38. What is a recursive SQL query?

a. A recursive SQL query is a query that refers to its own output in order to perform additional operations. It is commonly used for hierarchical or tree-like data structures.

39. What is the difference between a correlated subquery and a nested subquery?

a. A correlated subquery is a subquery that references columns from the outer query, while a nested subquery is a subquery that is independent of the outer query.

40. What is the difference between a natural join and an equijoin?

a. A natural join is a join operation that automatically matches columns with the same name from both tables, whereas an equijoin is a join operation that explicitly specifies the join condition using equality operators.

41. What is the difference between an outer join and an inner join?

a. An inner join returns only the matching rows from both tables, whereas an outer join returns all rows from one table and matching rows from the other table(s).

42. What is the difference between a left join and a right join?

a. A left join returns all rows from the left table and matching rows from the right table, whereas a right join returns all rows from the right table and matching rows from the left table.

43. What is a full outer join?

a. A full outer join returns all rows from both tables, including unmatched rows, and combines them based on the join condition.

44. What is a self-referencing foreign key?

a. A self-referencing foreign key is a foreign key that references the primary key of the same table. It is used to establish hierarchical relationships within a single table.

45. What is the purpose of the GROUP BY clause?

a. The GROUP BY clause is used to group rows based on one or more columns. It is typically used with aggregate functions to perform calculations on each group.

46. What is the purpose of the HAVING clause?

a. The HAVING clause is used to filter grouped rows based on specific conditions. It operates on the results of the GROUP BY clause.

47. What is the purpose of the ORDER BY clause?

a. The ORDER BY clause is used to sort the result set based on one or more columns in ascending or descending order.

48. What is the purpose of the DISTINCT keyword?

a. The DISTINCT keyword is used to retrieve unique values from a column in a result set, eliminating duplicate rows.

49. What is the purpose of the LIKE operator?

a. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column. It allows wildcard characters like % (matches any sequence of characters) and _ (matches any single character).

50. What is the purpose of the IN operator?

a. The IN operator is used in a WHERE clause to check if a value matches any value in a list or a subquery.

51. What is the purpose of the BETWEEN operator?

a. The BETWEEN operator is used in a WHERE clause to check if a value lies within a specified range of values, inclusive of the endpoints.

52. What is the purpose of the EXISTS operator?

a. The EXISTS operator is used in a WHERE clause to check if a subquery returns any rows. It returns true if the subquery result set is not empty.

53. What is the purpose of the COUNT() function?

a. The COUNT() function is used to count the number of rows or non-null values in a column.

54. What is the purpose of the SUM() function?

a. The SUM() function is used to calculate the sum of values in a column.

55. What is the purpose of the AVG() function?

a. The AVG() function is used to calculate the average value of a column.

56. What is the purpose of the MAX() function?

a. The MAX() function is used to retrieve the maximum value from a column.

57. What is the purpose of the MIN() function?

a. The MIN() function is used to retrieve the minimum value from a column.

58. What is the purpose of the GROUP_CONCAT() function?

a. The GROUP_CONCAT() function is used to concatenate values from multiple rows into a single string, grouped by a specific column.

59. What is the purpose of the JOIN keyword?

a. The JOIN keyword is used to combine rows from two or more tables based on related columns.

60. What is a self-referencing table?

a. A self-referencing table is a table that has a foreign key column referencing its own primary key. It is used to represent hierarchical relationships within a single table.

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

a. UNION combines the result sets of two or more SELECT statements and removes duplicate rows, whereas UNION ALL combines the result sets without removing duplicates.

62. What is the purpose of the ROW_NUMBER() function?

a. The ROW_NUMBER() function assigns a unique sequential number to each row within a result set. It is often used for pagination or ranking purposes.

63. What is the purpose of the RANK() function?

a. The RANK() function assigns a rank to each row within a result set based on a specified criteria, such as ordering by a column. It allows you to identify the ranking of each row.

64. What is the purpose of the DENSE_RANK() function?

a. The DENSE_RANK() function is similar to the RANK() function but assigns consecutive ranks to rows without gaps. If two rows have the same rank, the next rank is skipped.

65. What is the purpose of the LAG() function?

a. The LAG() function is used to access the value of a previous row within a result set based on a specified column. It allows you to compare values across adjacent rows.

66. What is the purpose of the LEAD() function?

a. The LEAD() function is used to access the value of a subsequent row within a result set based on a specified column. It allows you to compare values across adjacent rows.

67. What is the purpose of the COALESCE() function?

a. The COALESCE() function is used to return the first non-null value from a list of expressions. It is often used to provide a default value when a column value is null.

68. What is the purpose of the CASE statement?

a. The CASE statement is used to perform conditional logic within a SQL statement. It allows you to evaluate multiple conditions and return different values based on the result.

69. What is the purpose of the TRUNCATE TABLE statement?

a. The TRUNCATE TABLE statement is used to remove all rows from a table, while keeping the table structure intact. It is faster than deleting all rows using the DELETE statement.

70. What is the purpose of the CONSTRAINT keyword?

a. The CONSTRAINT keyword is used to define rules and relationships on columns within a table. It ensures data integrity and enforces business rules.

71. What is the purpose of the PRIMARY KEY constraint?

a. The PRIMARY KEY constraint is used to uniquely identify each record in a table. It ensures that the primary key column(s) have unique values and cannot contain null values.

72. What is the purpose of the FOREIGN KEY constraint?

a. The FOREIGN KEY constraint is used to establish a relationship between two tables based on a common column. It ensures referential integrity by enforcing that values in the foreign key column exist in the referenced table's primary key.

73. What is the purpose of the INDEX keyword?

a. The INDEX keyword is used to create an index on one or more columns of a table. It improves query performance by allowing faster data retrieval based on the indexed columns.

74. What is the purpose of the CASCADE keyword in a FOREIGN KEY constraint?

a. The CASCADE keyword is used to specify that changes made to the primary key values in the referenced table should be propagated to the foreign key values in the referring table. This ensures that the relationship remains valid.

75. What is the purpose of the UPDATE statement?

a. The UPDATE statement is used to modify existing records in a table. It allows you to change the values of one or more columns based on specified conditions.

76. What is the purpose of the DELETE statement?

a. The DELETE statement is used to remove one or more records from a table. It allows you to delete rows based on specified conditions.

77. What is the purpose of the COMMIT statement?

a. The COMMIT statement is used to permanently save all changes made within a transaction to the database. Once committed, the changes are visible to other users.

78. What is the purpose of the ROLLBACK statement?

a. The ROLLBACK statement is used to undo all changes made within a transaction and restore the database to its previous state. It is typically used when an error occurs or when the transaction needs to be canceled.

79. What is the purpose of the SAVEPOINT statement?

a. The SAVEPOINT statement is used to define a specific point within a transaction to which you can roll back. It allows you to undo changes up to a specific savepoint without rolling back the entire transaction.

80. What is the purpose of the CONSTRAINT keyword in the ALTER TABLE statement?

a. The CONSTRAINT keyword in the ALTER TABLE statement is used to add, modify, or drop constraints on columns within an existing table.

81. What is the purpose of the DISTINCT keyword in the SELECT statement?

a. The DISTINCT keyword in the SELECT statement is used to retrieve unique values from a column in the result set, eliminating duplicate rows.

82. What is the purpose of the AS keyword in the SELECT statement?

a. The AS keyword in the SELECT statement is used to assign an alias to a column or a table. It allows you to refer to the column or table by the assigned alias in subsequent parts of the query.

83. What is the purpose of the ORDER BY clause in the SELECT statement?

a. The ORDER BY clause in the SELECT statement is used to sort the result set based on one or more columns in ascending or descending order.

84. What is the purpose of the GROUP BY clause in the SELECT statement?

a. The GROUP BY clause in the SELECT statement is used to group rows based on one or more columns. It is typically used with aggregate functions to perform calculations on each group.

85. What is the purpose of the HAVING clause in the SELECT statement?

a. The HAVING clause in the SELECT statement is used to filter grouped rows based on specific conditions. It operates on the results of the GROUP BY clause.

86. What is the purpose of the LIMIT clause in the SELECT statement?

a. The LIMIT clause in the SELECT statement is used to restrict the number of rows returned by a query. It allows you to specify the maximum number of rows to be retrieved.

87. What is the purpose of the OFFSET clause in the SELECT statement?

a. The OFFSET clause in the SELECT statement is used in conjunction with the LIMIT clause to skip a specified number of rows before starting to return the result set.

88. What is the purpose of the JOIN keyword in the SELECT statement?

a. The JOIN keyword in the SELECT statement is used to combine rows from two or more tables based on related columns. It allows you to retrieve data from multiple tables in a single query.

89. What is the purpose of the INNER JOIN?

a. The INNER JOIN is a join operation that returns only the matching rows from both tables based on the specified join condition. It combines rows that have matching values in the joined columns.

90. What is the purpose of the LEFT JOIN?

a. The LEFT JOIN is a join operation that returns all rows from the left table and the matching rows from the right table based on the specified join condition. If no match is found, null values are returned for the right table columns.

91. What is the purpose of the RIGHT JOIN?

a. The RIGHT JOIN is a join operation that returns all rows from the right table and the matching rows from the left table based on the specified join condition. If no match is found, null values are returned for the left table columns.

92. What is the purpose of the FULL OUTER JOIN?

a. The FULL OUTER JOIN is a join operation that returns all rows from both tables, including unmatched rows, and combines them based on the join condition. If no match is found, null values are returned for the respective columns.

93. What is the purpose of the UNION operator?

a. The UNION operator 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.

94. What is the purpose of the UNION ALL operator?

a. The UNION ALL operator is used to combine the result sets of two or more SELECT statements into a single result set, including duplicate rows.

95. What is the purpose of the LIKE operator in the WHERE clause?

a. The LIKE operator is used in the WHERE clause to search for a specified pattern in a column. It allows wildcard characters like % (matches any sequence of characters) and _ (matches any single character).

96. What is the purpose of the IN operator in the WHERE clause?

a. The IN operator is used in the WHERE clause to check if a value matches any value in a list or a subquery.

97. What is the purpose of the EXISTS operator in the WHERE clause?

a. The EXISTS operator is used in the WHERE clause to check if a subquery returns any rows. It returns true if the subquery result set is not empty.

98. What is the purpose of the GROUP BY clause in the SELECT statement?

a. The GROUP BY clause in the SELECT statement is used to group rows based on one or more columns. It is typically used with aggregate functions to perform calculations on each group.

99. What is the purpose of the ORDER BY clause in the SELECT statement?

a. The ORDER BY clause in the SELECT statement is used to sort the result set based on one or more columns in ascending or descending order.

100. What is the purpose of the DISTINCT keyword in the SELECT statement?

- The DISTINCT keyword in the SELECT statement is used to retrieve unique values from a column in the result set, eliminating duplicate rows.