2. What is the DISTINCT statement and how do you use it?
The SQL DISTINCT statement returns only distinct (different) values from a table.
It is used when a column contains many duplicate values, and you only want to list
the different (distinct) values.
71. What is the difference between DELETE and TRUNCATE in SQL?
The DELETE and TRUNCATE commands in SQL are used to remove data from a table, but they differ in functionality, performance, and behavior.
The DELETE command allows you to remove specific rows from a table using a WHERE clause.
For example, DELETE FROM Employees WHERE Department = 'HR'; removes
only the rows where the department is "HR". If no WHERE clause is specified, all rows in the table will be deleted.
On the other hand, the TRUNCATE command is used to quickly remove
all rows from a table without allowing any conditions. For example,
TRUNCATE TABLE Employees; clears the entire table.
Triggers are not executed when using TRUNCATE.
DELETE is more flexible as it allows filtering rows, but it is slower due to logging each row's deletion and executing triggers. TRUNCATE is faster and ideal for quickly clearing all data in a table but lacks flexibility and resets identity
columns. Use DELETE for selective row removal and TRUNCATE for clearing entire tables efficiently.
72. What is the difference between UNION and UNION ALL?
The UNION operator combines the results of the queries and removes duplicate rows from the final result set.
It performs a distinct operation to ensure that each row in the output is unique.
However, this additional step of eliminating duplicates makes UNION slower compared to UNION ALL, especially when working
with large datasets. For example, the query SELECT City FROM Customers
UNION SELECT City FROM Suppliers; returns a list of unique cities from both the Customers and Suppliers tables.
On the other hand, the UNION ALL operator also combines the results of two or more SELECT queries,
but it does not remove duplicates. It returns all rows from the queries, including any duplicates.
Since it skips the distinct operation, UNION ALL is faster and more efficient than UNION.
For instance, the query SELECT City FROM Customers UNION ALL SELECT City FROM Suppliers; returns all cities from both tables,
even if some are repeated.
73. What is the difference between the HAVING clause and the WHERE clause?
The HAVING clause and the WHERE clause in SQL are both used to filter data, but they serve different purposes and are used in different contexts within a query.
The WHERE clause is used to filter rows before any grouping or aggregation takes place. It operates on individual rows and specifies the conditions
that the rows must meet to be included in the result set. For example, in the query SELECT * FROM Employees WHERE Department = 'HR';,
the WHERE clause filters rows where the Department is "HR". The WHERE clause cannot be used to filter data based on aggregate
functions like SUM, COUNT, or AVG.
The HAVING clause, on the other hand, is used to filter the results after the grouping or aggregation has been performed.
It operates on grouped data and is typically used with aggregate functions. For example, in the query SELECT Department,
COUNT(*) AS EmployeeCount FROM Employees GROUP BY Department HAVING COUNT(*) > 10;,
the HAVING clause filters groups (departments) that have more than 10 employees.

Unlike WHERE, the HAVING clause can filter based on aggregate functions because it processes the data after aggregation.
74. What is a transaction in SQL?

A transaction in SQL is a sequence of one or more operations (such as INSERT, UPDATE, DELETE, etc.) that are executed as a single, logical unit of work. Transactions ensure that the database remains in a consistent state,
even in the event of errors, system failures, or concurrent access by multiple users.For example, consider transferring money between two bank accounts.
The transaction involves two steps: deducting money from one account and adding it to another. Without a transaction, if one step is completed and the other fails, the database will be in an inconsistent state.
Using a transaction ensures both operations are executed together or not at all.
75. What is a deadlock?
A deadlock in SQL happens when two or more transactions block each other because they are waiting for resources that the other transactions are holding.
This causes a situation where none of the transactions can move forward.

How Deadlocks Happen:
Imagine two transactions:

Transaction 1 locks Table A and tries to lock Table B.
Transaction 2 locks Table B and tries to lock Table A.
Now, both transactions are stuck because:

Transaction 1 is waiting for Transaction 2 to release Table B.
Transaction 2 is waiting for Transaction 1 to release Table A.
This is called a deadlock, as neither can continue.
76. What is the difference between a database and a schema?
The difference between a database and a schema lies in their scope, purpose, and role in organizing and managing data within a database management system (DBMS).

A database is a structured collection of data that is stored, managed, and accessed electronically. It serves as the main container that holds the data and the structures necessary for organizing and accessing it.
It is the highest-level container in a DBMS. A single database can contain multiple schemas.
A schema is a logical container or namespace within a database. It organizes database objects like tables, views, stored procedures, and indexes into separate groups for better management and access control.
A schema is part of a database. Multiple schemas can exist within a single database, and each schema can group related objects. Schemas are used to logically separate and organize database objects.
They can also be used to implement security by controlling user access at the schema level.
77. What is the difference between a temporary table and a table variable?
A temporary table is a physical table stored in the tempdb database and is used to store temporary data.Created using the CREATE TABLE or SELECT INTO syntax, similar to regular tables.
Use temporary tables when working with larger datasets or when explicit indexing and full transaction support are required.Fully supported within transactions.Slightly slower than table variables
because they are written to the tempdb disk.
A table variable is a memory-optimized variable that stores temporary data within the scope of a batch, stored procedure, or function.
Declared using the DECLARE keyword. Does not support explicit indexing but can use primary keys or unique constraints.
Limited transaction support; changes to table variables are not rolled back during transaction rollbacks. Faster for small datasets since they are primarily stored in memory.
78. What is the purpose of the GROUP BY clause?
The GROUP BY clause in SQL is used to group rows with the same values in specified columns into summary rows, such as totals, averages, counts, or other aggregate results. It is often used in combination
with aggregate functions like COUNT(), SUM(), AVG(), MAX(), and MIN() to perform operations on each group of rows.
It categorizes rows that share the same value in specified columns into groups.
It is used to generate summarized or aggregated results, such as sales totals by

region or average scores by department.
79. What is the difference between CHAR and VARCHAR data types?
CHAR (Fixed-Length): This data type always reserves a fixed amount of space. Even if the stored string is shorter than the defined length,
it will be padded with spaces to meet the required length. It is best suited for data that is consistently the same length, such as zip codes or country codes.
For example, if you define a CHAR(10) column, it will always use 10 characters of storage, even if you store a 3-character string like "ABC" (the remaining 7 characters will be padded with spaces).
VARCHAR (Variable-Length): Unlike CHAR, VARCHAR only uses the space necessary to store the actual data. It does not pad shorter strings with spaces and is best suited for data that can vary in length,
such as names or descriptions. For example, if you define a VARCHAR(10) column, it will store a string like "John" using only 4 characters, without adding any extra space.
80. What is a stored procedure?
At its core, a stored procedure is a precompiled collection of one or more SQL statements saved within the database.
We can think of it as a function for a database: we define it once, and then we can call it whenever we need, passing parameters if necessary.
Stored procedures have several important features that distinguish them from simple queries:

they are precompiled, which means the database compiles and optimizes them at the time of creation
they can be parameterized
their reusability makes them an invaluable tool for maintaining consistent and efficient database operations
81. What is a subquery?
Also called an inner query, a query placed inside another query, or an outer query. A subquery may occur in the clauses such as SELECT, FROM, WHERE, UPDATE, etc.
It's also possible to have a subquery inside another subquery.
The innermost subquery is run first, and its result is passed to the containing query (or subquery).
You can use SQL subqueries in SELECT, INSERT, UPDATE, and DELETE statements.
Specifically, you can nest a subquery in the SELECT, FROM, WHERE, JOIN, and HAVING SQL clauses.
Also, you can adopt SQL queries in conjunction with several SQL operators, such as =, <, >, >=, <=, IN, NOT IN, EXISTS, NOT EXISTS, and more.
82. What is a view?
A view is a virtual table whose contents are defined by a query. Like a table, a view consists of a set of named columns and rows of data.
Views simplify complex queries by encapsulating them, making it easier to retrieve specific data without repeatedly writing the same query logic.
Views with aggregations, joins, or certain functions cannot be updated directly.
It acts as a virtual table that dynamically retrieves data from one or more underlying tables, making it a useful tool for abstraction and security in database management.
83. What is the difference between a cross join and an inner join?
A Cross Join and an Inner Join are two different types of SQL joins that determine how rows from two tables are combined, but they serve distinct purposes.
A Cross Join produces the Cartesian product of two tables, meaning every row from the first table is combined with every row from the second table.
An Inner Join returns only the rows that have matching values in both tables based on a specified condition (typically using a ON clause).
In summary, a Cross Join generates all possible combinations of rows from two tables without any matching condition,
while an Inner Join only returns rows where there is a match between the tables based on a specified condition.
84. What is the purpose of the COMMIT statement?
The COMMIT statement in SQL is used to permanently save changes made to the database during the current transaction. When a transaction is committed, all

the changes (such as INSERT, UPDATE, DELETE)
that have been performed during that transaction are finalized and written to the database. It marks the successful completion of a transaction, guarantees data integrity, and ensures that changes are reflected in the database.

85. What is the purpose of the ROLLBACK statement?

The ROLLBACK statement in SQL is used to undo changes made during the current transaction. It reverses all the modifications (such as INSERT, UPDATE, DELETE) that have been performed since the beginning of the transaction or since the last SAVEPOINT within the transaction.

A ROLLBACK is typically issued when there is an error or when the changes need to be discarded for any other reason.

86. What is the purpose of the NULL value in SQL?

In SQL, the NULL value represents missing or unknown data. It is used to indicate that a particular field does not have a value, either because the value is not available, not applicable, or has not been assigned yet. NULL is not the same as an empty string ('') or a zero (0);
it is a distinct value that represents the absence of any value.

The NULL value in SQL is used to represent the absence or unknown status of data in a database. It is a distinct value, different from empty strings or zeros, and requires special handling in SQL queries.
 It plays an important role in managing missing or incomplete data within relational databases.

87. What is the purpose of the DISTINCT keyword?

The DISTINCT keyword in SQL is used to remove duplicate rows from the result set of a query. When used in a SELECT statement,
it ensures that the result only includes unique values for the specified columns, eliminating any repeated data. The DISTINCT keyword is essential for filtering out duplicate data from query results, ensuring that only unique rows are returned.

It is especially useful when you want to get a list of unique values from a column or a combination of columns in a database table.

88. What is the difference between the IN and EXISTS operators?

The IN operator is used to check whether a specified value matches any value in a list or a subquery result. It is commonly used with a list of static values or with a subquery that returns a set of values.

The IN operator can sometimes be slower than EXISTS when the subquery returns a large number of rows, as it needs to match each value in the list against the column.

The EXISTS operator is used to test whether a subquery returns any rows. It is typically used when you are checking for the existence of records in a subquery, rather than comparing values.

EXISTS can be more efficient than IN when the subquery returns a large number of rows. Since EXISTS stops processing once it finds a matching row, it often performs better when dealing with large datasets.

89. What is the purpose of the TRIGGER statement?

A TRIGGER in SQL is a special type of stored procedure that automatically executes (or "fires") when a specific event occurs in the database. These events can include actions such as inserting, updating, or deleting records in a table.

Triggers are used to enforce business rules, automate tasks, or ensure data integrity without requiring manual intervention.

BEFORE Trigger: Executes before the data modification event (e.g., before an INSERT, UPDATE, or DELETE operation). It allows you to modify the data or validate it before the change is applied.

AFTER Trigger: Executes after the data modification event. It is typically used to perform actions such as logging changes, updating related tables, or enforcing referential integrity.

90. What is the difference between a unique constraint and a unique index?

A unique index ensures that the values in the index key columns are unique. A unique constraint guarantees that no duplicate values can be inserted into the column(s) on which the constraint is created.

When a unique constraint is created a corresponding unique index is automatically created on the column(s).

91. What is the purpose of the TOP or LIMIT clause?

The TOP (in SQL Server and some other databases) or LIMIT (in MySQL, PostgreSQL, and others) clause is used to restrict the number of rows returned by a query. This is particularly useful when you want to retrieve
only a subset of the result set, such as the first few records or the most recent entries, instead of the entire table.
TOP is used in SQL Server and MS Access, while LIMIT is used in MySQL, PostgreSQL, SQLite, and others.
These clauses are especially useful for retrieving a subset of records, improving performance, and controlling the size of the result set, such as fetching the top N records or a specific range of records.
92. What is the difference between the UNION and JOIN operators?
The UNION operator is used to combine the result sets of two or more SELECT queries into a single result set. It returns rows from multiple queries and removes duplicate rows by default.
The JOIN operator is used to combine rows from two or more tables based on a related column between them. A clause used to combine and retrieve records from two or multiple tables.
SQL tables can be joined based on the relationship between the columns of those tables.
93. What is the purpose of the CASE statement?
The CASE statement in SQL is used to add conditional logic to your queries. It allows you to evaluate expressions and return different values based on certain conditions.
The CASE statement is similar to an "IF-THEN-ELSE" construct found in programming languages. Whether used for data categorization, conditional formatting, or complex calculations,
it enhances the flexibility of your SQL queries.
94. What is the purpose of the ROW_NUMBER() function?
ROW_NUMBER function is a SQL ranking function that assigns a sequential rank number to each new record in a partition.
When the SQL Server ROW NUMBER function detects two identical values in the same partition, it assigns different rank numbers to both.
It is often used with ORDER BY to define the sequence in which rows are numbered.
ROW_NUMBER() is commonly used in scenarios like pagination, ranking, or when you need to assign a unique identifier to rows for further analysis.
95. What is the difference between the EXISTS and NOT EXISTS operators?
The EXISTS and NOT EXISTS operators in SQL are used to test for the presence (or absence) of rows returned by a subquery.
They are often employed in conjunction with subqueries to filter data based on whether certain conditions are met in related tables.
Similar to EXISTS and IN, NOT EXISTS and NOT IN are often comparable in functionality. However, NOT IN can yield unexpected results when NULL values are involved.
NOT IN will return an empty set if any NULL values are part of the subquery result, regardless of other values. NOT EXISTS doesnâ ™t get tripped up by NULL values and behaves as expected.
96. What is a self-join?
A Self Join is a type of a JOIN query used to compare rows within the same table. Unlike other SQL JOIN queries that join two or more tables, a self join joins a table to itself.
To use a self join, a table must have a unique identifier column, a parent column, and a child column.
In conclusion, a Self Join is used to establish the parent-child or hierarchy relationships between different rows in the same table and extract meaningful insights.
97. What is an ALIAS command?
A temporary name given to a table (or a column in a table) while executing a certain SQL query. Aliases are used to improve the code readability and make the code more compact.
98. Why are SQL functions used?
A database object representing a set of SQL statements frequently used for a certain task. A function takes in some input parameters, performs calculations or other manipulations on them, and returns the result.

Functions help improve code readability and avoid repetition of the same code snippets.SQL functions are essential for transforming, analyzing, and managing data in a database.
By offering a wide range of operationsâ€"from simple string manipulation to complex data aggregationâ€"they empower users to write efficient and effective queries that address various business and technical requirements.
99. What is SQL?
It stands for Structured Query Language, and it's a programming language used for interaction with relational database management systems (RDBMS).
This includes fetching, updating, inserting, and removing data from tables.Retrieve, insert, update, and delete data from tables using commands like SELECT, INSERT, UPDATE, and DELETE.
Create, modify, and delete database structures such as tables, indexes, and views using commands like CREATE, ALTER, and DROP.
Manage transactions to ensure data integrity using commands like COMMIT, ROLLBACK, and SAVEPOINT.Work with relationships between tables using joins, constraints, and foreign keys.
100. What is a database?
A database is an organized collection of data that is stored, managed, and accessed electronically.
It is designed to efficiently store large amounts of information, make it easily retrievable, and ensure data integrity and security.
Databases are used in a wide range of applications, from small personal systems to large enterprise systems.
Data is stored in a structured format, such as tables, rows, and columns, in relational databases. Databases provide query languages (like SQL) to fetch specific data quickly.Constraints and rules ensure the accuracy and consistency of data.
Access controls restrict unauthorized access to sensitive data.Ensures that data is stored efficiently to minimize duplication.
101. What is a primary key?
A column (or multiple columns) of a table to which the PRIMARY KEY constraint was imposed to ensure unique and non-null values in that column.
In other words, a primary key is a combination of the NOT NULL and UNIQUE constraints.
The primary key uniquely identifies each record of the table. Each table should contain a primary key and can't contain more than one primary key.
A primary key cannot contain NULL values because it is used to uniquely identify rows.
102. What is a foreign key?
A column (or multiple columns) of a table to which the FOREIGN KEY constraint was imposed to link this column to the primary key in another table (or several tables).
The purpose of foreign keys is to keep connected various tables of a database.The foreign key constraint
ensures that the value in the foreign key column matches a value in the primary key column of the referenced table or is NULL.
103. What is the difference between a primary key and a unique key?
A primary key serves as a unique identifier for each row in a table.
A unique key uniquely determines a row that isnâ€™t necessarily the primary key.
A primary key cannot accept NULL values.
A unique key can accept NULL values.
Only one primary key can be defined per table.
Multiple unique keys can be defined in a table.
A primary key creates a clustered index.
A unique key creates a non-clustered index.
A primary key supports auto-increment values, which is commonly used for generating unique identifiers.
A unique key does not support auto-increment values.
104. What is a join in SQL?
SQL joins are a fundamental concept in relational database systems, used to combine records from two or more tables in a database.
A join is performed whenever two or more tables are listed in an SQL statement and is based on the relationship between the columns of these tables.

The most commonly used JOINs are INNER JOIN, LEFT JOIN, and RIGHT JOIN, each serving different purposes for data retrieval.
The purpose of using a JOIN is to retrieve meaningful and related information from different tables by matching the values of columns.
105. What is normalization?
Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity.
It involves structuring tables and their relationships to ensure that each piece of data is stored in only one place.
The primary goal of normalization is to eliminate data anomalies and maintain consistency in the database.
Goals of Normalization
Minimize Data Redundancy: Avoid storing the same data in multiple places.
Ensure Data Integrity: Maintain accurate and consistent data across the database.
Simplify Maintenance: Make the database easier to update and manage.
106. What is the difference between a clustered and a non-clustered index?
While a clustered index defines the physical order of records of a table and performs data searching based on the key values, a non-clustered index keeps the order of records that do not match the physical order of the actual data on the disk.
 A table can have only one clustered index but many non-clustered ones.
Clustered Index:
Directly affects the physical storage of data.
There can be only one per table.
Ideal for columns frequently used for sorting or range queries.
Non-Clustered Index:
Does not alter physical data storage.
Multiple non-clustered indexes can be created on a table.
Useful for quick lookups and filtering.
107. What is the difference between a left join and a right join?
In SQL, LEFT JOIN and RIGHT JOIN are used to combine data from two tables based on a related column. While they serve similar purposes,
their behavior differs in terms of which table's rows are retained when no matching rows are found in the other table.
LEFT JOIN (or LEFT OUTER JOIN)
A LEFT JOIN retrieves all rows from the left table and the matching rows from the right table.
If no match is found in the right table, the result will include the row from the left table with NULL values for columns from the right table.
This join is commonly used when you want to retain all data from the left table, regardless of whether it has corresponding rows in the right table.

RIGHT JOIN (or RIGHT OUTER JOIN)
A RIGHT JOIN retrieves all rows from the right table and the matching rows from the left table.
If no match is found in the left table, the result will include the row from the right table with NULL values for columns from the left table.
This join is used when you want to retain all data from the right table, regardless of whether it has corresponding rows in the left table.
108. What is the difference between a left join and a full outer join?
LEFT JOIN (or LEFT OUTER JOIN)
A LEFT JOIN retrieves all rows from the left table and the matching rows from the right table.
If no match is found in the right table, the result will include the row from the left table with NULL values for columns from the right table.
This join is commonly used when you want to retain all data from the left table, regardless of whether it has corresponding rows in the right table.

FULL (OUTER) JOIN â “ returns all records from both (or all) tables. It can be considered as a combination of left and right joins.
If a row in the left table has no match in the right table, its columns from the right table will be NULL.
Similarly, if a row in the right table has no match in the left table, its

columns from the left table will be NULL.
This join provides a complete view of data from both tables.
109. What is the difference between a right join and an inner join?
RIGHT JOIN (or RIGHT OUTER JOIN)
A RIGHT JOIN retrieves all rows from the right table and the matching rows from the left table.
If no match is found in the left table, the result will include the row from the right table with NULL values for columns from the left table.
This join is used when you want to retain all data from the right table, regardless of whether it has corresponding rows in the left table.
INNER JOIN
An INNER JOIN retrieves only the rows that have matching values in both tables.
Rows without a match in either table are excluded from the result.
110. What is the difference between a primary key and a foreign key?
Primary Key
A primary key is a unique identifier for each record in a table.
It ensures that no two rows in the table have the same value for the primary key column(s).
A table can have only one primary key.
It cannot contain NULL values.
Automatically creates a clustered index (in most database systems).

Foreign Key
A foreign key is a column or set of columns in one table that establishes a link to the primary key in another table.
It ensures referential integrity by enforcing that the value in the foreign key column must exist in the referenced primary key column or be NULL.
A table can have multiple foreign keys.
It can contain duplicate values if the relationship is one-to-many.