1. **What are the different types of commands in SQL?**
   * SQL commands are divided into several types:
     + **DDL (Data Definition Language):** Commands such as CREATE, ALTER, DROP, and TRUNCATE define and modify the structure of database objects like tables.
     + **DML (Data Manipulation Language):** Commands such as SELECT, INSERT, UPDATE, and DELETE manipulate data within tables.
     + **DCL (Data Control Language):** Commands such as GRANT and REVOKE control access to data in the database.
     + **TCL (Transaction Control Language):** Commands such as COMMIT, ROLLBACK, and SAVEPOINT manage transactions within the database.
2. **What are Constraints in SQL? What are its types?**
   * Constraints enforce rules on data columns to ensure the integrity and accuracy of the data. Types of constraints include:
     + **NOT NULL:** Ensures that a column cannot have a NULL value.
     + **UNIQUE:** Ensures that all values in a column are unique.
     + **PRIMARY KEY:** Uniquely identifies each record in a table and combines NOT NULL and UNIQUE constraints.
     + **FOREIGN KEY:** Ensures referential integrity by linking two tables.
     + **CHECK:** Ensures that all values in a column satisfy a specific condition.
     + **DEFAULT:** Sets a default value for a column when no value is specified.
3. **What is a Primary Key?**
   * A primary key is a column, or a set of columns, that uniquely identifies each row in a table. It ensures that each record can be uniquely identified and does not allow NULL values. Example:

CREATE TABLE Students (

    StudentID INT PRIMARY KEY,

    Name VARCHAR(100)

);

1. **What is a Foreign Key?**
   * A foreign key is a column or set of columns that establish a link between data in two stables by referencing the primary key of another table. This helps maintain referential integrity. Example:

CREATE TABLE Orders (

    OrderID INT PRIMARY KEY,

    StudentID INT,

    FOREIGN KEY (StudentID) REFERENCES Students(StudentID)

);

1. **What is a Join?**
   * A join is used to combine rows from two or more tables based on a related column between them. Example:

SELECT Orders.OrderID, Students.Name

FROM Orders

JOIN Students ON Orders.StudentID = Students.StudentID;

1. **What is a Subquery?**
   * A subquery is a query nested inside another query, used to perform operations that depend on the results of another query. Example:  
     SELECT Name

FROM Students

WHERE StudentID IN (SELECT StudentID FROM Orders WHERE Amount > 100);

1. **What are some common clauses used with SELECT query in SQL?**
   * Common clauses include:
     + **WHERE:** Filters records based on conditions.
     + **GROUP BY:** Groups records with identical values.
     + **HAVING:** Filters groups based on conditions.
     + **ORDER BY:** Sorts the result set.
     + **LIMIT:** Limits the number of rows returned.
2. **What is the difference between ENUM and SET data types?**
   * **ENUM:** A string object with a value chosen from a list of permitted values. Example:

CREATE TABLE Colors (

    Color ENUM('Red', 'Green', 'Blue')

);

**SET:** A string object that can have zero or more values, each of which must be chosen from a list of permitted values. Example:

CREATE TABLE Permissions (

    Permission SET('Read', 'Write', 'Execute')

);

1. **What is an Alias in SQL?**
   * An alias is a temporary name given to a table or column for the purpose of a particular SQL query. Example:  
     SELECT Name AS StudentName FROM Students;
2. **What are the TRUNCATE, DELETE and DROP statements?**
   * **TRUNCATE:** Removes all rows from a table without logging individual row deletions. Cannot be rolled back. Example:

TRUNCATE TABLE Students;

* **DELETE:** Removes specified rows from a table and can be rolled back. Example:  
  DELETE FROM Students WHERE StudentID = 1;
* **DROP:** Removes a table or database entirely. Example:  
  DROP TABLE Students;

1. **What is the difference between DELETE and TRUNCATE statements?**
   * **DELETE:** Removes specific rows based on a condition, can be rolled back, and triggers may be activated.
   * **TRUNCATE:** Removes all rows from a table, cannot be rolled back, and is faster because it does not log individual row deletions.
2. **What is the difference between single row (scalar) functions and multiple rows (aggregate / group) functions?**
   * **Single Row (Scalar) Functions:** Operate on each row and return a single value per row. Example:

SELECT UPPER(Name) FROM Students;

* **Multiple Rows (Aggregate/Group) Functions:** Operate on a set of rows and return a single value for the set. Example:  
  SELECT AVG(Score) FROM Students;

1. **What is the difference between IN and BETWEEN operators?**
   * **IN:** Specifies multiple values in a WHERE clause. Example:  
     SELECT \* FROM Students WHERE StudentID IN (1, 2, 3);
   * **BETWEEN:** Selects values within a given range. Example:  
     SELECT \* FROM Students WHERE Score BETWEEN 50 AND 100;
2. **How do we use the DISTINCT statement? What is its use?**
   * The DISTINCT keyword is used to remove duplicate records from the result set. Example:  
     SELECT DISTINCT Name FROM Students;
3. **What is the difference between the WHERE and HAVING clauses?**
   * **WHERE:** Filters rows before grouping them. Example:  
     SELECT \* FROM Students WHERE Score > 50;
   * **HAVING:** Filters groups after grouping has been performed. Example:  
     SELECT Class, AVG(Score) FROM Students GROUP BY Class HAVING AVG(Score) > 50;
4. **How many Aggregate functions are available in SQL?**
   * Common aggregate functions include:
     + **COUNT():** Counts the number of rows.
     + **SUM():** Calculates the sum of a set of values.
     + **AVG():** Calculates the average of a set of values.
     + **MIN():** Finds the minimum value.
     + **MAX():** Finds the maximum value. Example:  
       SELECT COUNT(\*), SUM(Score), AVG(Score), MIN(Score), MAX(Score) FROM Students;
5. **What is the use of the LIKE operator in SQL?**
   * The LIKE operator is used to search for a specified pattern in a column. Example:  
     SELECT \* FROM Students WHERE Name LIKE 'A%';
6. **What is inner join, outer join, equi join, non-equi join and self join in SQL?**
   * **Inner Join:** Returns records with matching values in both tables. Example:  
     SELECT \* FROM Students INNER JOIN Orders ON Students.StudentID = Orders.StudentID;
   * **Outer Join:** Returns records with matching values and all records from one or both tables.
     + **Left Join:** Returns all records from the left table and matched records from the right table. Example:  
       SELECT \* FROM Students LEFT JOIN Orders ON Students.StudentID = Orders.StudentID;
     + **Right Join:** Returns all records from the right table and matched records from the left table. Example:  
       SELECT \* FROM Students RIGHT JOIN Orders ON Students.StudentID = Orders.StudentID;
     + **Full Join:** Returns all records when there is a match in either the left or right table. Example:  
       SELECT \* FROM Students FULL JOIN Orders ON Students.StudentID = Orders.StudentID;
   * **Equi Join:** A type of inner join using equality operator. Example:  
     SELECT \* FROM Students, Orders WHERE Students.StudentID = Orders.StudentID;
   * **Non-Equi Join:** Uses a non-equality operator. Example:  
     SELECT \* FROM Employees E, Salaries S WHERE E.Salary < S.Salary;
   * **Self Join:** Joins a table to itself. Example:  
     SELECT A.StudentID, B.Name FROM Students A, Students B WHERE A.StudentID = B.StudentID;
7. **What are the types of the Subquery?**
   * **Single-row subquery:** Returns a single row. Example:  
     SELECT \* FROM Students WHERE StudentID = (SELECT MAX(StudentID) FROM Orders);
   * **Multi-row subquery:** Returns multiple rows. Example:  
     SELECT \* FROM Students WHERE StudentID IN (SELECT StudentID FROM Orders);
   * **Correlated subquery:** References a column from the outer query. Example:  
     SELECT \* FROM Students S WHERE EXISTS (SELECT \* FROM Orders O WHERE O.StudentID = S.StudentID);
8. **What is the use of IN, ANY and ALL operators in Subquery?**
   * **IN:** Checks if a value matches any value in a list or subquery. Example:  
     SELECT \* FROM Students WHERE StudentID IN (SELECT StudentID FROM Orders);
   * **ANY:** Compares a value to each value in a list or subquery and returns true if any comparison is true. Example:  
     SELECT \* FROM Students WHERE Score > ANY (SELECT Score FROM Students WHERE Class = '10');
   * **ALL:** Compares a value to all values in a list or subquery and returns true if all comparisons are true. Example:  
     SELECT \* FROM Students WHERE Score > ALL (SELECT Score FROM Students WHERE Class = '10');