1. What is DDL in SQL?

DDL stands for Data Definition Language. It is a set of SQL commands used to define, modify, and delete database structures like tables, schemas, indexes, and constraints. Unlike DML (Data Manipulation Language), DDL commands do not modify actual data in the database; instead, they modify the structure of the database.

2. Name all DDL commands in SQL.

The main DDL commands in SQL are:

- **CREATE** – Creates a new database, table, index, or view.
- **ALTER** – Modifies an existing table structure (e.g., adding a column).
- **DROP** – Deletes a database, table, or index permanently.
- **TRUNCATE** – Removes all records from a table but keeps its structure.
- **RENAME** – Renames a table or a column.

3. What is the purpose of the CREATE command?

The CREATE command is used to create new database objects, such as:

- Databases
- Tables
- Indexes
- Views

It defines the structure and properties of these objects.

Example: Creating a table

```
CREATE TABLE Employees (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
    Salary DECIMAL(10,2)
);
```

4. How do you create a database using SQL?

To create a database in SQL, use the CREATE DATABASE command.

Syntax:

CREATE DATABASE my_database;

This creates a new database named my_database. Some database systems require USE my_database; to start using the newly created database.

5. What is the difference between DROP and DELETE?

Feature	**DELETE**	**DROP**
Purpose	Removes specific records from a table	Deletes the entire table, including structure
Affects	Only data inside the table	Both data and structure
Rollback?	Yes (if inside a transaction)	No (cannot be undone)
Usage	Used when you want to remove only some records	Used when you no longer need the table

Example:

- DELETE FROM Employees WHERE Age > 50; → Deletes records of employees older than 50.
- DROP TABLE Employees; → Removes the entire Employees table from the database.

6. How can you modify a table's column name?

To rename a column, use the **ALTER TABLE... RENAME COLUMN** statement.

Syntax:

ALTER TABLE table_name RENAME COLUMN old_column_name TO new_column_name;

Example:

ALTER TABLE Employees RENAME COLUMN Name TO FullName;

7. What is the syntax to delete a column from a table?

To remove a column, use the **ALTER TABLE... DROP COLUMN** statement.

Syntax:

ALTER TABLE table_name DROP COLUMN column_name;

Example:

ALTER TABLE Employees DROP COLUMN Salary;

This removes the Salary column from the Employees table.

8. What happens if you use the DROP TABLE command?

When you execute **DROP TABLE**, it **completely removes** the table, including:

- All data stored in it
- The structure (columns, constraints, indexes)
- Associated metadata

Example:

DROP TABLE Employees;

This will permanently delete the Employees table from the database. This action cannot be undone.

9. What is a constraint in SQL?

A **constraint** in SQL is a rule applied to a column or table to enforce data integrity. It ensures that only valid data is entered.

Common constraints:

- **PRIMARY KEY** – Uniquely identifies each record.
- **UNIQUE** – Ensures values in a column are unique.
- **NOT NULL** – Prevents empty (NULL) values in a column.
- **CHECK** – Ensures values meet a specific condition.
- **FOREIGN KEY** – Ensures referential integrity between two tables.

Example:

```
CREATE TABLE Students (
   StudentID INT PRIMARY KEY,        // PRIMARY KEY ensures StudentID is unique.
   Name VARCHAR(50) NOT NULL,   // NOT NULL ensures Name cannot be empty.
   Age INT CHECK (Age >= 18)          // CHECK ensures Age is 18 or older.
);
```

10. Explain the difference between a Primary Key and a Unique Key.

Feature	**Primary Key**	**Unique Key**
Uniqueness	Ensures each row is unique	Ensures column values are unique
NULL values	**Cannot** be NULL	Can have **one** NULL value

Feature	Primary Key	Unique Key
Number of Keys	Only **one** per table	Can have **multiple** Unique Keys
Indexing	Creates a **clustered index**	Creates a **non-clustered index**

Example:

```
CREATE TABLE Customers (
   CustomerID INT PRIMARY KEY,     // Primary Key (must be unique and non-null)
   Email VARCHAR(100) UNIQUE  // Unique Key (ensures unique values but can be NULL).
);
```

11. How do you add a foreign key to an existing table?

To add a foreign key to an existing table, use the ALTER TABLE statement with ADD CONSTRAINT.

Syntax:

```
ALTER TABLE child_table
ADD CONSTRAINT fk_name
FOREIGN KEY (child_column)
REFERENCES parent_table (parent_column);
```

Example:
Let's say we have an Orders table and we want to link its CustomerID column to the Customers table's CustomerID column.

```
ALTER TABLE Orders
ADD CONSTRAINT fk_customer
FOREIGN KEY (CustomerID)
REFERENCES Customers(CustomerID);
```

12. Write an SQL query to add a CHECK constraint on a salary column (salary must be > 3000).

To enforce a **CHECK** constraint, use ALTER TABLE with ADD CONSTRAINT.

SQL Query:

```
ALTER TABLE Employees
ADD CONSTRAINT chk_salary
```

CHECK (Salary > 3000);

This ensures that no employee can have a salary less than or equal to 3000.

13. What happens when you try to insert duplicate values into a UNIQUE column?

If you attempt to insert a duplicate value into a column with a UNIQUE constraint, the database will return an error and reject the insertion.

Example:

```
CREATE TABLE Users (
   UserID INT UNIQUE,
   Name VARCHAR(50)
);
INSERT INTO Users (UserID, Name) VALUES (1, 'Alice');
INSERT INTO Users (UserID, Name) VALUES (1, 'Bob'); // Here, ERROR Occurred : Duplicate entry
```

14. How do you remove a foreign key constraint from a table?

To remove a foreign key, use the ALTER TABLE statement with DROP CONSTRAINT.

Syntax:

```
ALTER TABLE table_name
DROP CONSTRAINT foreign_key_name;
```

Example:

```
ALTER TABLE Orders
DROP CONSTRAINT fk_customer;
```

This removes the foreign key constraint named fk_customer from the Orders table.

15. What happens when you try to modify the datatype of an existing column?

- If the new datatype is **compatible** (e.g., INT → BIGINT), the change succeeds.
- If the new datatype is **incompatible** (e.g., VARCHAR → INT when text data exists), the query **fails**.
- Some databases require the column to be **empty** before changing the datatype.

Example:

ALTER TABLE Employees

MODIFY COLUMN Age BIGINT;

This changes Age from INT to BIGINT.

16. How do you ensure that an age column only accepts values between 18 and 60?

Use a **CHECK** constraint.

SQL Query:

ALTER TABLE Employees

ADD CONSTRAINT chk_age

CHECK (Age BETWEEN 18 AND 60);

This ensures that values outside **18-60** are not allowed.

17. How can you rename a table using SQL?

To rename a table, use the RENAME or ALTER TABLE command.

Syntax:

RENAME TABLE old_table_name TO new_table_name;

Example:

RENAME TABLE Employees TO Staff;

This renames the Employees table to Staff.

18. How do you delete an existing constraint from a table?

To remove a constraint from a table, use the ALTER TABLE command with DROP CONSTRAINT.

Syntax:

ALTER TABLE table_name

DROP CONSTRAINT constraint_name;

Example:
If we have a CHECK constraint named chk_age on the Employees table, we can remove it using:

ALTER TABLE Employees

DROP CONSTRAINT chk_age;

19. What is the purpose of the DEFAULT constraint?

The **DEFAULT** constraint automatically assigns a default value to a column if no value is provided during an INSERT operation.

Example:

CREATE TABLE Employees (ID INT PRIMARY KEY,

Name VARCHAR(50), Salary DECIMAL(10,2) DEFAULT 5000);

If we insert a new employee **without specifying Salary**, the default value 5000 is used:

INSERT INTO Employees (ID, Name) VALUES (1, 'Alice');

This record will have Salary = 5000.

20. How do you create an index to speed up queries on a table?

An index improves the speed of searches by allowing the database to quickly locate rows.

Creating an Index

Syntax:

CREATE INDEX index_name

ON table_name (column_name);

Example:

CREATE INDEX idx_employee_name ON Employees (Name);

This creates an index on the Name column of the Employees table, making queries like:

Creating a Unique Index

To ensure values are unique:

CREATE UNIQUE INDEX idx_unique_email ON Users (Email);

21. Design a Student Management System using CREATE TABLE with primary and foreign keys.

A Student Management System typically includes Students, Courses, and Enrollments.

For Queries and Table Info

// Refer Folder StudentMang

22. Create a Products table where ProductID is the primary key, and ProductName is unique.

CREATE TABLE Products (ProductID INT PRIMARY KEY,

ProductName VARCHAR(255) UNIQUE, Price DECIMAL(10,2), Quantity INT);

- ProductID is the **primary key**.
- ProductName is **unique**, preventing duplicate product names.

23. Write a query to drop a table only if it exists.

DROP TABLE IF EXISTS Products;

This prevents errors if the table doesn't exist.

24. Create a Library Management System with Books, Authors, and Borrowers tables.

For Queries and Table Info

// Refer Folder LibraryMang

- BookInfo table has a foreign key (AuthorID) linking it to Authors.
- BorrowerInfo have unique emails to prevent duplicate entries.

25. Modify a Customers table to change the column size of CustomerName to 150 characters.

ALTER TABLE Customers

MODIFY COLUMN CustomerName VARCHAR(150);

This changes CustomerName from its previous size to **150 characters**.

26. Create a Banking System database with proper constraints on transactions.

For Queries and Table Info

// Refer Folder BankSystem

- Balance cannot be negative (CHECK (Balance >= 0)).
- Transaction amount cannot be zero (CHECK (Amount <> 0)).

27. Write an SQL query to remove an existing UNIQUE constraint from a column.

ALTER TABLE Users

DROP CONSTRAINT unique_email;

This removes the UNIQUE constraint on the email column.

28. Design an E-commerce database with Orders, Customers, and Products.

For Queries and Table Info

// Refer Folder EcommData

- Customers have a **unique email**.

- Products have a **unique name**.
- Orders reference Customers via **foreign key**.

29. Create a table for Employee records that ensures no two employees can have the same email.

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100) UNIQUE,
    Position VARCHAR(50)
);
```

The UNIQUE constraint ensures no duplicate emails.

30. Explain the differences between DDL, DML, DCL, and TCL in SQL.

Category	Full Form	Description	Examples
DDL	Data Definition Language	Defines the structure of the database (schema, tables).	CREATE, ALTER, DROP
DML	Data Manipulation Language	Manages data inside tables.	INSERT, UPDATE, DELETE, SELECT
DCL	Data Control Language	Controls access and permissions.	GRANT, REVOKE
TCL	Transaction Control Language	Manages transactions in the database.	COMMIT, ROLLBACK, SAVEPOINT

- **DDL** changes the **structure** (e.g., creating/deleting tables).
- **DML** modifies **data** (e.g., adding, updating, deleting rows).
- **DCL** controls **user access** (e.g., who can access what).
- **TCL** handles **transactions** (e.g., save or undo operations).