**1️⃣ DDL – Data Definition Language**

DDL commands are used to **define or modify the database structure**. Common commands:

* CREATE – create database or table
* ALTER – modify table structure
* DROP – delete table or database

**Example: Create a College Database**

-- Create Database

CREATE DATABASE CollegeDB;

-- Use the database

USE CollegeDB;

-- Create Department Table

CREATE TABLE Department (

DeptID INT PRIMARY KEY AUTO\_INCREMENT,

DeptName VARCHAR(50) NOT NULL UNIQUE,

Location VARCHAR(50)

);

-- Create Student Table

CREATE TABLE Student (

StudentID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(50) NOT NULL,

Gender CHAR(1) CHECK (Gender IN ('M','F')),

DOB DATE NOT NULL,

DeptID INT,

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(15),

FOREIGN KEY (DeptID) REFERENCES Department(DeptID)

);

✅ **Explanation:**

* CREATE DATABASE – creates CollegeDB
* CREATE TABLE – defines table structure
* PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK – constraints
* AUTO\_INCREMENT – automatically increases ID

**2️⃣ DML – Data Manipulation Language**

DML commands are used to **manipulate data in tables**. Common commands:

* INSERT – add records
* UPDATE – modify records
* DELETE – remove records
* SELECT – retrieve data

**Example: Insert and Update Data**

-- Insert departments

INSERT INTO Department (DeptName, Location)

VALUES ('Computer Science', 'Block A'),

('Mechanical', 'Block B'),

('Electronics', 'Block C');

-- Insert students

INSERT INTO Student (Name, Gender, DOB, DeptID, Email, Phone)

VALUES ('Mayuri Kasetty', 'F', '2000-05-10', 1, 'mayuri@example.com', '9876543210'),

('Ravi Kumar', 'M', '1999-08-15', 2, 'ravi@example.com', '9123456780');

-- Update a student record

UPDATE Student

SET Phone = '9998887776'

WHERE Name = 'Ravi Kumar';

-- Delete a student record

DELETE FROM Student

WHERE Name = 'Ravi Kumar';

✅ **Explanation:**

* INSERT INTO – adds rows to table
* UPDATE – modifies existing row(s)
* DELETE – removes row(s)

**3️⃣ DCL – Data Control Language**

DCL commands are used to **control access to the database**. Common commands:

* GRANT – give privileges
* REVOKE – remove privileges

**Example: Grant and Revoke Privileges**

-- Grant SELECT privilege to a user

GRANT SELECT, INSERT ON CollegeDB.Student TO 'student\_user'@'localhost';

-- Revoke privilege

REVOKE INSERT ON CollegeDB.Student FROM 'student\_user'@'localhost';

✅ **Explanation:**

* GRANT allows a user to perform certain actions
* REVOKE removes that ability

**4️⃣ TCL – Transaction Control Language**

TCL commands are used to **control transactions**. Common commands:

* COMMIT – save changes permanently
* ROLLBACK – undo changes
* SAVEPOINT – set intermediate points to rollback

**Example: Transaction Control**

-- Start transaction

START TRANSACTION;

-- Insert student

INSERT INTO Student (Name, Gender, DOB, DeptID, Email, Phone)

VALUES ('Anil Sharma', 'M', '2001-02-20', 3, 'anil@example.com', '9001122334');

-- Update student

UPDATE Student

SET Phone = '9009988776'

WHERE Name = 'Mayuri Kasetty';

-- Rollback if something goes wrong

ROLLBACK;

-- Insert another student and commit

INSERT INTO Student (Name, Gender, DOB, DeptID, Email, Phone)

VALUES ('Sita Reddy', 'F', '2002-03-12', 1, 'sita@example.com', '9012345678');

COMMIT;

✅ **Explanation:**

* START TRANSACTION begins a transaction
* ROLLBACK undoes uncommitted changes
* COMMIT makes changes permanent
* SAVEPOINT can be used to rollback to an intermediate state

**5️⃣ Full Flow Summary**

| **Category** | **Commands** | **Example** |
| --- | --- | --- |
| **DDL** | CREATE, ALTER, DROP | CREATE TABLE Student... |
| **DML** | INSERT, UPDATE, DELETE, SELECT | INSERT INTO Student ... |
| **DCL** | GRANT, REVOKE | GRANT SELECT ON Student ... |
| **TCL** | COMMIT, ROLLBACK, SAVEPOINT | COMMIT; |

If you want, I can **draw a complete diagram showing all tables and sample data** for this CollegeDB, which will make it easier to visualize how DDL/DML/DCL/TCL work together.

Do you want me to do that?

**You said:**

rename truncate

**ChatGPT said:**

Ah! You’re asking about **RENAME** and **TRUNCATE** in SQL. Let’s clarify both carefully.

**1️⃣ RENAME**

The RENAME command is used to **change the name of a table or a column**.

**Example: Rename Table**

-- Rename the Student table to Learner

RENAME TABLE Student TO Learner;

**Example: Rename Column (MySQL syntax)**

-- Rename column Phone to ContactNumber in Learner table

ALTER TABLE Learner RENAME COLUMN Phone TO ContactNumber;

✅ **Explanation:**

* RENAME TABLE → changes the table name
* ALTER TABLE ... RENAME COLUMN → changes column name
* No data is lost; only names change

**2️⃣ TRUNCATE**

The TRUNCATE command is used to **delete all data from a table**, but **does not remove the table structure**.

**Example: Truncate Table**

-- Remove all data from Learner table

TRUNCATE TABLE Learner;

✅ **Explanation:**

* Faster than DELETE because it **doesn’t log individual row deletions**
* Resets auto-increment counters (in most DBMS)
* Cannot use WHERE clause; it removes **all rows**
* Table structure, indexes, and constraints remain intact

**Difference Between DELETE and TRUNCATE**

| **Feature** | **DELETE** | **TRUNCATE** |
| --- | --- | --- |
| Removes rows | Yes (with condition possible) | All rows |
| WHERE clause | Yes | No |
| Transaction log | Logs each row deletion | Minimal logging |
| Speed | Slower for large tables | Faster |
| Resets AUTO\_INCREMENT | No | Yes (in most DBMS) |