

Perfect 👍 Now let's make this **complete & exam-ready**:

I'll give you **definition + command + example** for **every SQL language command** under DDL, DML, DQL, DCL, TCL — all using the same Student theme.

---

## 📌 1. DDL (Data Definition Language)

👉 **Definition:** DDL commands are used to **define and manage database schema (structure)** such as tables, views, and indexes.

### **CREATE – create new objects**

```
CREATE TABLE Student (  
    roll_no INT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    dept VARCHAR(30),  
    age INT,  
    gpa DECIMAL(3,2)  
);
```

### **ALTER – modify existing objects**

```
ALTER TABLE Student ADD email VARCHAR(100);
```

### **DROP – delete database objects permanently**

```
DROP TABLE Student;
```

### **TRUNCATE – delete all rows but keep table structure**

```
TRUNCATE TABLE Student;
```

### **RENAME – rename a database object**

```
ALTER TABLE Student RENAME TO Students_Info;
```

---

## 📌 2. DML (Data Manipulation Language)

👉 **Definition:** DML commands are used to **insert, modify, and delete data** stored inside tables.

### **INSERT – add new records**

```
INSERT INTO Student (roll_no, name, dept, age, gpa)
```

```
VALUES (101, 'Rahul', 'CSE', 20, 8.2);
```

#### **UPDATE – modify existing records**

```
UPDATE Student
```

```
SET gpa = 8.5
```

```
WHERE roll_no = 101;
```

#### **DELETE – remove records**

```
DELETE FROM Student
```

```
WHERE roll_no = 101;
```

---

### **3. DQL (Data Query Language)**

 **Definition:** DQL is used to **query (retrieve) information** from the database.

*(Some books merge this into DML, but academically SELECT = DQL.)*

#### **SELECT – retrieve records**

```
SELECT * FROM Student;
```

#### **SELECT with condition – filter data**

```
SELECT name, dept, gpa
```

```
FROM Student
```

```
WHERE gpa > 8.0;
```

---

### **4. DCL (Data Control Language)**

 **Definition:** DCL commands are used to **control access and permissions** on database objects.

#### **GRANT – give permissions**

```
GRANT SELECT, INSERT ON Student TO user1;
```

#### **REVOKE – remove permissions**

```
REVOKE INSERT ON Student FROM user1;
```

---

### **5. TCL (Transaction Control Language)**

👉 **Definition:** TCL commands are used to **manage transactions** in a database, ensuring **ACID properties** (Atomicity, Consistency, Isolation, Durability).

**COMMIT – save changes permanently**

```
INSERT INTO Student (roll_no, name, dept, age, gpa)
VALUES (102, 'Priya', 'ECE', 21, 7.9);

COMMIT;
```

**ROLLBACK – undo uncommitted changes**

```
DELETE FROM Student WHERE dept = 'ECE';

ROLLBACK;
```

**SAVEPOINT – set a checkpoint in a transaction**

```
SAVEPOINT sp1;

UPDATE Student SET gpa = 9.0 WHERE roll_no = 101;

ROLLBACK TO sp1; -- undo update but keep earlier work
```

---

✅ **Final Recap (One-Line Definitions)**

Language Definition		Commands
<b>DDL</b>	Defines/manages schema (structure of DB objects)	CREATE, ALTER, DROP, TRUNCATE, RENAME
<b>DML</b>	Manipulates data inside tables	INSERT, UPDATE, DELETE
<b>DQL</b>	Retrieves data from database	SELECT
<b>DCL</b>	Manages access & permissions	GRANT, REVOKE
<b>TCL</b>	Manages transactions & ensures ACID	COMMIT, ROLLBACK, SAVEPOINT

---

👉 Do you want me to also add **output screenshots (sample table before & after each command)** so you can visualize how the commands affect the data?

### 1. Changing a Column's Data Type or Size:

Code

```
ALTER TABLE table_name  
ALTER COLUMN column_name new_data_type; -- SQL Server
```

Or, in some other database systems:

Code

```
ALTER TABLE table_name  
MODIFY COLUMN column_name new_data_type; -- MySQL, Oracle
```

Example: To change the data type of the Price column in the Products table to DECIMAL(10, 2):

Code

```
ALTER TABLE Products  
ALTER COLUMN Price DECIMAL(10, 2);
```

### 2. Adding a New Column:

Code

```
ALTER TABLE table_name  
ADD COLUMN column_name data_type [column_constraints];
```

Example: To add a new Email column with a VARCHAR(255) data type to the Customers table:

Code

```
ALTER TABLE Customers  
ADD COLUMN Email VARCHAR(255);
```

### 3. Dropping an Existing Column:

Code

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

Example: To remove the FaxNumber column from the Suppliers table:

Code

```
ALTER TABLE Suppliers  
DROP COLUMN FaxNumber;
```

### 4. Renaming a Column:

Code

```
ALTER TABLE table_name  
RENAME COLUMN old_column_name TO new_column_name; -- PostgreSQL, Oracle
```

Or, in SQL Server:

Code

```
EXEC sp_rename 'table_name.old_column_name', 'new_column_name', 'COLUMN';
```

Example: To rename the FName column to FirstName in the Employees table (PostgreSQL):

Code

```
ALTER TABLE Employees  
RENAME COLUMN FName TO FirstName;
```