

Class - 02

What is DDL?

Data Definition Language (Define and manage the database structure.)

Command	Description
CREATE	Creates database objects (like tables)
ALTER	Modifies database objects (adds/removes columns)
DROP	Deletes database objects permanently.
TRUNCATE	Deletes all rows from a table quickly

What is DML?

Data Manipulation Language (modify data inside tables)

Command	Description
INSERT	Adds new records to a table.
UPDATE	Modifies existing records.
DELETE	Removes records from a table.
SELECT	Retrieves data (not modifying, but part of DML).

What is DCL?

Data Control Language (Commands manage user permissions)
✓ Typically used by database administrators.

Command	Description
GRANT	Gives a user privileges (read, write, etc.).
REVOKE	Removes a user's privileges.

What is TCL?

Transaction Control Language (Commands manage database transactions.)
✓ Ensures data consistency and rollback if needed.

Command	Description
COMMIT	Saves all changes permanently.
ROLLBACK	Undoes changes in the current transaction.
SAVEPOINT	Creates a checkpoint to roll back to.

Create Table

The diagram illustrates the SQL syntax for creating a table, with annotations pointing to specific parts of the code:

- Table Name:** Points to the `employees` table name in the `CREATE TABLE` statement.
- Column Name:** Points to the `employee_id` column name.
- Data Type:** Points to the `SERIAL` data type for the `employee_id` column.
- Constraints:** Points to the `PRIMARY KEY` constraint for the `employee_id` column.

```
CREATE TABLE employees (  
  employee_id SERIAL PRIMARY KEY, -- Auto-incrementing unique ID  
  first_name TEXT NOT NULL, -- Mandatory column  
  last_name TEXT NOT NULL,  
  email TEXT UNIQUE NOT NULL, -- Ensures email is unique  
  age INTEGER CHECK (age >= 18), -- Restricts age to 18+  
  salary NUMERIC(10,2) CHECK (salary > 0), -- Salary must be positive  
  department TEXT DEFAULT 'General', -- Assigns default department if not provided  
  hire_date DATE DEFAULT CURRENT_DATE -- Auto-fills current date  
);
```

Data Types of PostgreSQL

Category	Data Type
Numeric	SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE PRECISION
Auto-increment	SERIAL, BIGSERIAL
Text	CHAR(n), VARCHAR(n), TEXT
Date & Time	DATE, TIME, TIMESTAMP, TIMESTAMPTZ, INTERVAL
Boolean	BOOLEAN

Constraints

Constraint	Purpose
PRIMARY KEY	Ensures each row is uniquely identified.
FOREIGN KEY	Maintains relationships between tables.
UNIQUE	Prevents duplicate values.
NOT NULL	Ensures column values cannot be NULL.
CHECK	Restricts column values with conditions.
DEFAULT	Assigns a default value when none is provided.
ON DELETE CASCADE	Deletes dependent rows automatically.
COMPOSITE PRIMARY KEY	Uses multiple columns as a unique identifier.

Unnormalized Data

Order ID	Customer Name	Customer Phone	Product	Category	Price	Quantity	Supplier	Supplier Phone
101	John Doe	1234567890	Laptop	Electronics	1000	1	Dell	9876543210
102	Alice Smith	0987654321	Mouse	Accessories	50	2	Logitech	9123456789
103	John Doe	1234567890	Keyboard	Accessories	80	1	Logitech	9123456789

Why Do We Need to Normalize?

1. Avoid Data Redundancy → "John Doe" appears multiple times.
2. Improve Data Consistency → Supplier details are repeated and might cause inconsistency.
3. Better Data Integrity → Helps maintain relationships without duplication.
4. Efficient Storage → Reducing redundancy saves space.
5. Easier Updates → Changing a supplier's phone number requires updates in multiple rows in an unnormalized table.

Normalized Data

Orders Table

OrderID	CustomerID	ProductID	Quantity
101	C001	P001	1
102	C002	P002	2
103	C001	P003	1

Customers Table

CustomerID	CustomerName	CustomerPhone
C001	John Doe	1234567890
C002	Alice Smith	0987654321

Products Table

ProductID	ProductName	CategoryID	Price	SupplierID
P001	Laptop	CAT1	1000	S001
P002	Mouse	CAT2	50	S002
P003	Keyboard	CAT2	80	S002

Categories Table

CategoryID	CategoryName
CAT1	Electronics
CAT2	Accessories

Suppliers Table

SupplierID	SupplierName	SupplierPhone
S001	Dell	9876543210
S002	Logitech	9123456789

Types of Keys in a Database

Primary Key

A Primary Key is a unique identifier for each record in a table. It cannot be **NULL** and must be **unique**.

CustomerID (PK)	CustomerName	Email	PhoneNumber
1	Alice	alice@email.com	555-1234
2	Bob	bob@email.com	555-5678

Foreign Key

A Foreign Key establishes a relationship between two tables. It refers to a **Primary Key** in another table.

OrderID (PK)	CustomerID (FK)	OrderDate
101	1	2024-02-28
102	2	2024-02-29

Types of Keys in a Database

Unique Key

A Unique Key ensures that values in a column remain unique, but unlike a Primary Key, it **can** contain **NULL** values.

EmployeeID (PK)	EmployeeName	Email
1	John Doe	john@email.com
2	Jane Doe	jane@email.com

Composite Key

A Composite Key consists of two or more columns combined to uniquely identify a row.

StudentID	CourseID	EnrollmentDate
1	101	2024-02-28
1	102	2024-02-29