

**SQL INTRODUCTION: DDL, DML, AND DATABASE STRUCTURE**

1. SQL Command Types

In SQL, commands are grouped into types. Two important groups are:

- Data Definition Language (DDL)**- These commands define or change the structure of the database: databases, tables, columns, and constraints.- They control how the database is built and organized.- Examples: CREATE, ALTER, TRUNCATE, DROP, RENAME.
- Data Manipulation Language (DML)**- These commands work with the data inside the tables.- They control how data is added, changed, removed, and retrieved.- Examples: INSERT, UPDATE, DELETE, SELECT.

**Analogy:-** DDL is like building or modifying a shop in Lagos: walls, shelves, counters.- DML is like putting goods on shelves, changing prices, removing old stock.

2. Practical Use Case (Nigerian Context)Imagine a small fintech startup in Lagos that manages agent banking transactions. They want to store:- Agents (POS agents).- Customers.- Transactions.We will design part of this database and use it to illustrate DDL and key concepts.Database structure (simplified view):- Database: FintechDB- Tables: agents, customers, transactionsExample table structures (conceptually):- agents: agent\_id, name, phone, lga, state- customers: customer\_id, full\_name, phone, date\_of\_birth, state- transactions: txn\_id, agent\_id, customer\_id, amount, txn\_type, txn\_time, status

3. Structure of a Database

At a high level, a relational database has:-

- A Database: A container that holds multiple related tables.
- Tables: Like Excel sheets, each with rows and columns.
- Columns: Fields such as agent\_id, name, state.
- Rows: Individual records (one agent, one customer, one transaction). For example, in FintechDB:-
- The agents table holds all POS agents.
- The customers table holds all customers.
- The transactions table holds all transaction records.

4. DDL Basics: Auto-commit Behavior  
DDL commands are usually auto-committed. This means:- When you run a DDL command (CREATE, ALTER, TRUNCATE, DROP, RENAME), the change is immediately saved to the database.- You usually cannot roll back these operations with a simple ROLLBACK. Therefore, use DDL carefully, especially TRUNCATE and DROP, because they can remove data or whole tables permanently.

5. DDL Commands Explained

### 5.1 CREATE: Creating Tables

The CREATE TABLE command defines the structure of a new table.

Example: create an agents table for POS agents in Nigeria:

```
CREATE TABLE agents (
    agent_id INT PRIMARY KEY,
    name      VARCHAR(100) NOT NULL,
    phone     VARCHAR(20)
    UNIQUE,
    lga       VARCHAR(100),
    state    VARCHAR(50));
```

Explanation:-

- agent\_id: integer, primary key (unique identifier for each agent).
- name: text up to 100 characters, cannot be NULL.
- phone: text up to 20 characters, must be unique.
- lga: text for local government area.
- state: state in Nigeria, stored as text.

5.2 CREATE: Another Example with Foreign KeysCreate a transactions table that links to agents and customers:CREATE TABLE transactions ( txn\_id INT PRIMARY KEY, agent\_id INT, customer\_id INT, amount DECIMAL(12, 2) NOT NULL, txn\_type VARCHAR(20) NOT NULL, txn\_time TIMESTAMP NOT NULL, status VARCHAR(20) NOT NULL, FOREIGN KEY (agent\_id) REFERENCES agents(agent\_id), FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id));Here transactions is a child table that depends on agents and customers.

5.3 ALTER: Changing Existing Tables

The ALTER TABLE command modifies an existing table structure.

Example A: add a new column email to customers: ALTER TABLE customers ADD COLUMN email VARCHAR(150);

Example B: change the data type of phone in customers to make it longer: ALTER TABLE customers MODIFY COLUMN phone VARCHAR(30);

Example C: rename a column from phone to phone\_number: ALTER TABLE customers RENAME COLUMN phone TO phone\_number;

5.4 TRUNCATE: Remove All Rows, Keep Table  
TRUNCATE TABLE removes all records from a table very quickly but keeps the table structure.  
Example: clear all data from customers but keep the table definition:  
`TRUNCATE TABLE customers;`  
Notes:- DDL, auto-committed. Cannot easily undo.- Faster than DELETE without WHERE for large tables.

5.5 DROP: Delete the Entire Table  
DROP TABLE completely removes a table and its data from the database.  
Example: drop the customers table:  
DROP TABLE customers;  
Notes:- The table structure and data are both removed.- Any foreign keys that reference this table will be affected.- Use with extreme caution.

5.6 RENAME: Renaming Tables

The RENAME TABLE command changes the name of a table.

Example:  
rename customers to bank\_customers;RENAME TABLE customers TO bank\_customers;This is useful during refactoring or when improving naming conventions.

6. Relationships in the ExampleRelationships describe how tables are connected.In our example:- Each transaction belongs to one agent.- Each transaction belongs to one customer.This creates one-to-many relationships:- One agent to many transactions.- One customer to many transactions.Implemented with foreign keys:FOREIGN KEY (agent\_id) REFERENCES agents(agent\_id),FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id);Here:- agents and customers are parent tables.- transactions is the child table.

7. Constraints in the ExampleConstraints are rules that the database enforces on the data. They improve data quality and consistency. Common constraints used in the examples:  
PRIMARY KEY- Uniquely identifies each row in a table.- Example: agent\_id INT PRIMARY KEY.  
FOREIGN KEY- Ensures a value in one table exists in another table.- Example: FOREIGN KEY (agent\_id) REFERENCES agents(agent\_id).  
NOT NULL- Ensures a column cannot be left empty.- Example: name VARCHAR(100) NOT NULL.  
UNIQUE- Ensures all values in a column are different.- Example: phone VARCHAR(20) UNIQUE.  
These constraints help prevent issues such as duplicate agents, transactions with non-existent agents, or missing critical fields like amount.

8. Data Types in the ExampleData types control what kind of data a column can store.  
INT- Whole numbers (no decimal).- Used for IDs like agent\_id, customer\_id, txn\_id.  
VARCHAR(n)- Variable-length text up to n characters.- Used for names, phone numbers, LGA, state, statuses.  
DATE- Stores a calendar date (year, month, day).- Example: date\_of\_birth.  
TIMESTAMP (or DATETIME)- Stores a full date and time.- Example: txn\_time for transaction time.  
DECIMAL(p, s)- Exact numeric type with p total digits and s decimal places.- Example: DECIMAL(12, 2) for amounts like 1500000.50.  
Choosing correct data types is critical, especially for money and dates in financial systems.

9. Practice Exercises with Answers

Exercise 1: Create a customers table

Write a CREATE TABLE statement for customers with:-

- customer\_id: INT, primary key.
- full\_name: VARCHAR(150), not null.
- phone: VARCHAR(20), unique.
- date\_of\_birth: DATE.
- state: VARCHAR(50).

Sample Answer: CREATE TABLE customers (customer\_id INT PRIMARY KEY, full\_name VARCHAR(150) NOT NULL, phone VARCHAR(20) UNIQUE, date\_of\_birth DATE, state VARCHAR(50));

Exercise 2: Add an email column to customers

ALTER TABLE customers ADD COLUMN email VARCHAR(150);

Exercise 3: Rename phone to phone\_number

ALTER TABLE customers RENAME COLUMN phone TO phone\_number;

Exercise 4: Clear all data from customers but keep the table

TRUNCATE TABLE customers;

Exercise 5: Drop the customers table completely

DROP TABLE customers;

Exercise 6: Rename customers to bank\_customers

RENAME TABLE customers TO bank\_customers;