

# Introduction (SQL-DDL-DML)

## 1. يعني إيه SQL؟

(RDBMS) هي اللغة التي بنستخدمها عشان ندير قواعد البيانات العلائقية (SQL (Structured Query Language). يعني تقدر من خلالها تنشئ، تعدل، تدير، وتتعلم عن البيانات في أي قاعدة بيانات.

أنواع أوامر SQL المشهورة:

- ♦ DDL (Data Definition Language) → أوامر خاصة بتصميم الجداول والقواعد.
- ♦ DML (Data Manipulation Language) → أوامر خاصة بإضافة وتعديل ومسح البيانات.
- ♦ DCL (Data Control Language) → أوامر التحكم في الصلاحيات والأذونات.
- ♦ TCL (Transaction Control Language) → أوامر خاصة بإدارة العمليات داخل قواعد البيانات.

## 2. لغة تعريف البيانات (DDL - Data Definition Language)

دي مجموعة أوامر بنستخدمها عشان نحدد شكل وهيكل قواعد البيانات، زي إنشاء الجداول، تعديلها، أو حتى حذفها بالكامل.

أوامر DDL المهمة:

- ✓ CREATE → لإنشاء جداول أو قواعد بيانات جديدة.
- ✓ ALTER → لتعديل الجداول أو الأعمدة.
- ✓ DROP → لحذف الجداول أو قواعد البيانات نهائيًا.
- ✓ TRUNCATE → لمسح كل البيانات من الجدول بس من غير ما نحذف الجدول نفسه.

مثال عملي:

```
CREATE TABLE Students (  
  ID INT PRIMARY KEY,  
  Name VARCHAR(50),  
  Age INT );
```

الكود ده بيعمل جدول اسمه Students فيه 3 أعمدة:

- ♦ ID (المفتاح الأساسي)
- ♦ Name (الاسم)
- ♦ Age (العمر)

### 3. لغة معالجة البيانات (DML - Data Manipulation Language)

الأوامر دي خاصة بالتعامل مع البيانات جوه الجداول، يعني إضافة بيانات جديدة، تعديلها، مسحها، أو استرجاعها.

#### أوامر DML المهمة:

- ✓ INSERT → لإضافة بيانات جديدة.
- ✓ UPDATE → لتعديل البيانات الموجودة.
- ✓ DELETE → لحذف بيانات معينة.
- ✓ SELECT → لاسترجاع البيانات من الجدول.

#### أمثلة عملية:

إضافة طالب جديد: 📄

```
INSERT INTO Students (ID, Name, Age) VALUES (1, 'Ali', 20);
```

📌 ده بيدخل طالب اسمه علي عنده 20 سنة في جدول Students.

✏ تعديل بيانات الطالب:

```
UPDATE Students  
SET Age = 21  
WHERE ID = 1;
```

📌 ده بيعدل عمر الطالب اللي رقمه 1 ويخليه 21 سنة بدل 20.

🗑 مسح بيانات الطالب:

```
DELETE FROM Students WHERE ID = 1;
```

📌 ده بيحذف الطالب اللي رقمه 1 من الجدول.

### 4. الفرق بين DDL و DML

المقارنة	DDL	DML
بيتعامل مع؟	هيكله البيانات والجداول	البيانات نفسها

المقارنة	DDL	DML
بيئات على؟	تصميم قاعدة البيانات	البيانات جوه الجداول
أمثلة	CREATE TABLE , ALTER TABLE , DROP TABLE	INSERT , UPDATE , DELETE

## 5. حاجات إضافية ممكن متكونش في الملف، بس مهمة جدًا:

### الفهارس (Indexes) في SQL:

- بتساعد في تسريع عمليات البحث والاستعلام عن البيانات.
- بنعمل فهرس على عمود معين عشان تحسين الأداء.

### مثال على إنشاء فهرس:

```
CREATE INDEX idx_name ON Students(Name);
```

ده بيعمل فهرس على عمود Name في جدول Students عشان الاستعلام يبقى أسرع.

### المفاتيح (Keys) في SQL:

- ♦ **Primary Key** (المفتاح الأساسي) → بيحدد كل سجل في الجدول يكون فريد.
- ♦ **Foreign Key** (المفتاح الخارجي) → بيربط جدولين ببعض.

## أنواع البيانات (Data Types) في SQL

في SQL، لازم نحدد نوع البيانات لكل عمود في الجدول عشان نضمن إدخال بيانات صحيحة ومنظمة. كل قاعدة بيانات (مثل MySQL، SQL Server، PostgreSQL) ممكن يكون ليها اختلافات بسيطة في الأنواع، بس معظم الأنواع الأساسية متشابهة.

### 1 الأنواع النصية (String Data Types)

تستخدم لتخزين النصوص مثل الأسماء والعناوين

النوع	الوصف	الحجم
CHAR(n)	نص ثابت الطول (Fixed Length)	يستهلك نفس الحجم دائمًا حتى لو النص أصغر
VARCHAR(n)	نص متغير الطول (Variable Length)	يستهلك حجم البيانات المدخلة فقط

النوع	الوصف	الحجم
TEXT	نص طويل جدًا	يُستخدم لتخزين نصوص كبيرة (مثل المقالات)

### أمثلة:

```
CREATE TABLE Users (
  Username VARCHAR(50), -- نص متغير الطول بحد أقصى 50 حرفًا
  Bio TEXT                -- نص طويل جدًا
);
```

## 2 الأعداد الصحيحة (Integer Data Types)

💡 تُستخدم لتخزين الأرقام الصحيحة بدون كسور

النوع	الوصف	الحجم
TINYINT	عدد صغير (0 إلى 255 أو -128 إلى 127)	1 بايت
SMALLINT	عدد صغير (أكبر من TINYINT)	2 بايت
INT أو INTEGER	عدد قياسي	4 بايت
BIGINT	عدد كبير جدًا	8 بايت

### أمثلة:

sql

CopyEdit

```
CREATE TABLE Orders (
  OrderID INT PRIMARY KEY, -- رقم الطلب
  Quantity SMALLINT        -- كمية المنتج
);
```

## 3 الأعداد العشرية (Decimal & Floating Point Data Types)

💡 تُستخدم للأرقام التي فيها كسور عشرية مثل الأسعار أو الحسابات الدقيقة

النوع	الوصف	الحجم
DECIMAL(p, s) أو NUMERIC(p, s)	عدد عشري بدقة عالية، p عدد الأرقام الكلية، s عدد الأرقام بعد الفاصلة	يعتمد على الحجم
FLOAT	عدد عشري لكنه أقل دقة عند الأرقام الكبيرة جدًا	4 بايت
REAL	مثل FLOAT لكنه أكثر دقة	8 بايت

## أمثلة:

```
CREATE TABLE Products (
Price DECIMAL(10, 2) -- أرقام إجمالية، 2 بعد العلامة العشرية 10
);
```

## 4 التواريخ والأوقات (Date & Time Data Types)

💡 تُستخدم لتخزين التواريخ والأوقات مثل تاريخ الميلاد أو وقت الطلب

النوع	الوصف
DATE	تاريخ فقط (YYYY-MM-DD)
TIME	وقت فقط (HH:MI:SS)
DATETIME	تاريخ ووقت معًا (YYYY-MM-DD HH:MI:SS)
TIMESTAMP	مثل DATETIME لكنه محسوب بناءً على التوقيت العالمي UTC

## أمثلة:

```
`CREATE TABLE Employees (
BirthDate DATE, -- تاريخ الميلاد
HireTime TIME, -- وقت التوظيف
CreatedAt TIMESTAMP -- تاريخ ووقت التسجيل
);
```

## 5 القيم المنطقية (Boolean Data Type)

💡 نستخدم لتخزين القيم "صح" أو "خطأ" (1 أو 0)

النوع	الوصف
BOOLEAN	في بعض الأنظمة مثل PostgreSQL
BIT	يستخدم في SQL Server لتخزين 0 أو 1

أمثلة:

```
CREATE TABLE Users (  
  IsActive BOOLEAN -- هل الحساب نشط؟ (True/False)  
);
```

## 6 الأنواع الثنائية (Binary Data Types)

💡 نستخدم لتخزين البيانات الثنائية مثل الصور أو الملفات

النوع	الوصف
BLOB	يستخدم لتخزين بيانات كبيرة مثل الصور
VARBINARY(n)	يستخدم لتخزين بيانات ثنائية متغيرة الطول

أمثلة:

```
CREATE TABLE Files (  
  FileData BLOB -- تخزين ملف ثنائي  
);
```

💡 متى نستخدم كل نوع؟

نوع البيانات	متى يُستخدم؟
VARCHAR(n)	لو عندك نصوص قصيرة متغيرة مثل أسماء المستخدمين
TEXT	لو بتخزن نصوص طويلة جداً مثل المقالات

نوع البيانات	متى يُستخدم؟
INT	لو بتخزن أرقام صحيحة مثل أرقام الطلبات
DECIMAL(10,2)	لو بتخزن أسعار أو بيانات مالية
DATETIME	لو عاوز تخزن تاريخ ووقت معًا
BOOLEAN	لو عاوز تخزن بيانات "صح" أو "خطأ"