

# Using Constraints

الـ **Constraints** هي قواعد (Rules) بنطبقها على الأعمدة (Columns) والجداول (Tables) في قاعدة البيانات بهدف ضمان سلامة البيانات (**Data Integrity**) ومنع إدخال بيانات غير صحيحة أو غير منطقية.

## ✓ NOT NULL

- المعنى: العمود لا يقبل قيم فارغة (NULL).
- الهدف: التأكد من أن كل سجل يحتوي قيمة في هذا العمود.

مثال:

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    Name VARCHAR(50) NOT NULL  
);
```

هنا العمود Name لازم يحتوي قيمة، وممنوع يكون فارغ.

## ✓ UNIQUE

- المعنى: القيم في العمود لازم تكون مميزة (لا تتكرر).
- الهدف: منع التكرار في البيانات.

مثال:

```
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    Email VARCHAR(100) UNIQUE  
);
```

هنا Email لازم يكون مختلف لكل موظف.

## ✓ DEFAULT

- المعنى: لو ما اتحددش قيمة للعمود عند الإدخال، يتم وضع قيمة افتراضية.

- الهدف: توفير قيمة تلقائية بدون الحاجة للإدخال اليدوي دائمًا.

مثال:

```
CREATE TABLE Orders (  
  OrderID INT PRIMARY KEY,  
  Status VARCHAR(20) DEFAULT 'Pending'  
);
```

لو ما اتحددش حالة الطلب، تلقائيًا بتكون Pending .

## ✓ PRIMARY KEY

- المعنى: عمود/مجموعة أعمدة بتمييز كل سجل في الجدول بشكل فريد.
- الخصائص:
- **Unique** (لا يتكرر)
- **NOT NULL** (لا يقبل NULL)
- الهدف: ضمان أن كل سجل له معرف مميز.

مثال:

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

العمود StudentID هو المفتاح الأساسي.

## ✓ FOREIGN KEY

- المعنى: عمود/أعمدة في جدول تشير إلى **Primary Key** في جدول آخر.
- الهدف: فرض **Referential Integrity** (سلامة العلاقات بين الجداول).

مثال:

```
CREATE TABLE Orders (  
  OrderID INT PRIMARY KEY,  
  CustomerID INT,
```

```
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

العمود CustomerID في جدول Orders لازم يكون موجود في جدول Customers.

---

## ◆ Integrated Example

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Age SMALLINT,
    Email VARCHAR(255) UNIQUE,
    GPA DECIMAL(3,2) DEFAULT 0.00
);
```

```
CREATE TABLE Courses (
    CourseID INT PRIMARY KEY,
    Title VARCHAR(150) NOT NULL
);
```

```
CREATE TABLE StudentCourses (
    StudentID INT NOT NULL,
    CourseID INT NOT NULL,
    CourseGPA DECIMAL(3,2) CHECK (CourseGPA >= 0 AND CourseGPA <= 4.00),
    PRIMARY KEY (StudentID, CourseID),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE
    CASCADE,
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) ON DELETE CASCADE
);
```

---

## ◆ Best Practices

- Name constraints ( CONSTRAINT fk\_orders\_customers ) for clarity.
- Use CHECK for value ranges (e.g., GPA, age).
- Remember: UNIQUE creates an index automatically.
- Don't confuse NOT NULL (mandatory value) with DEFAULT (auto value if missing).

- Be cautious with `ON DELETE CASCADE` — it can delete large amounts of data unintentionally.
  - Use transactions when altering constraints for safety.
-