

2.Constraint in SQL

- ***Constraint**

Các ràng buộc được sử dụng để giới hạn loại dữ liệu có thể đi vào bảng, điều này đảm bảo tính chính xác và độ tin cậy của dữ liệu trong bảng. Nếu có bất kì vi phạm nào về ràng buộc, hành động đó sẽ bị hủy bỏ.

Các ràng buộc thường dùng trong SQL:

- NOT NULL: Đảm bảo mỗi cột ko thể có giá trị NULL.
- UNIQUE: Đảm bảo rằng tất cả các giá trị trong 1 cột là khác nhau
- PRIMARY KEY: Là sự kết hợp của NOT NULL và UNIQUE. Xác định duy nhất từng hàng trong bảng.
- FOREIGN KEY: Là khóa ngoại, ngăn chặn hành động phá hủy liên kết giữa các bảng.
- CHECK: Đảm bảo rằng các giá trị trong 1 cột thỏa mãn điều kiện cụ thể nào đấy.
- DEFAULT: Đặt giá trị mặc định cho 1 cột nếu ko có giá trị nào được chỉ định.
- CREATE INDEX: Được dùng để tạo và lấy dữ liệu từ cơ sở dữ liệu 1 cách nhanh chóng

1. NOT NULL

Theo mặc định, một cột có thể chứa giá trị NULL.

Các NOT NULL hạn chế thực thi 1 cột để không chấp nhận giá trị NULL.

Điều này bắt buộc trường luôn chứa giá trị, nghĩa là bạn ko thể chèn bản ghi mới vào giá trị mà ko thêm giá trị vào trường này.

Eg:

```
CREATE TABLE Persons(  
    ID INT NOT NULL,
```

```
    LastName VARCHAR(255) NOT NULL,  
    FirstName VARCHAR(255) NOT NULL,  
    Age int  
);  
ALTER TABLE Persons  
MODIFY Age int NOT NULL;
```

2. UNIQUE

Các UNIQUE hạn chế để đảm bảo rằng tất cả các giá trị trong 1 cột là khác nhau. Cả ràng buộc UNIQUE và PK đều đảm bảo tính duy nhất cho 1 cột or tập hợp các cột.

1 PRIMARY KEY ràng buộc tự động có 1 UNIQUE ràng buộc.

Có thể có nhiều UNIQUE ràng buộc trên mỗi bảng nhưng chỉ có 1 PRIMARY KEY ràng buộc cho mỗi bảng.

Eg:

```
CREATE TABLE Persons(  
    ID INT NOT NULL,  
    LastName VARCHAR(255) NOT NULL,  
    FirstName VARCHAR(255) NOT NULL,  
    Age int  
    UNIQUE(ID)  
);  
SELECT * FROM Persons;
```

Để tạo UNIQUE ràng buộc trên cột Age khi bảng đã được tạo:

```
ALTER TABLE Persons  
ADD UNIQUE(Age)
```

Để loại bỏ một UNIQUE ràng buộc:

```
ALTER TABLE Persons
```

```
DROP INDEX Age.
```

3. PRIMARY KEY

Các PRIMARY KEY hạn chế nhận dạng duy nhất mỗi bản ghi trong 1 bảng.

Khóa chính phải chứa DUY NHẤT và không được chứa giá trị NULL.

Một bảng chỉ có thể có 1 khóa chính, và trong bảng, khóa chính này có thể bao gồm một or nhiều cột.

Eg:

```
CREATE TABLE Persons(
```

```
    ID INT NOT NULL,
```

```
    LastName VARCHAR(255) NOT NULL,
```

```
    FirstName VARCHAR(255) NOT NULL,
```

```
    Age int
```

```
    PRIMARY KEY(ID)
```

```
);
```

```
ALTER TABLE Persons
```

```
ADD PRIMARY KEY(Age)⇒ lỗi
```

Để loại bỏ 1 PRIMARY KEY:

```
ALTER TABLE PRIMARY KEY
```

```
DROP PRIMARY KEY;
```

4. FOREIGN KEY

Các FOREIGN KEY hạn chế được dùng để ngăn chặn những hành động sẽ phá hủy các liên kết giữa các bảng.

FOREIGN KEY sẽ là 1 trường(tập hợp các trường) trong 1 bảng, tham chiếu đến PK của 1 bảng khác.

Bảng có khóa ngoại gọi là bảng con và bảng có khóa chính gọi là bảng tham chiếu or bảng cha.

Persons Table			
PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

Orders Table		
OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

```
CREATE TABLE Persons(  
    PersonID INT NOT NULL,  
    LastName VARCHAR(255) NOT NULL,  
    FirstName VARCHAR(255) NOT NULL,  
    Age INT,  
    PRIMARY KEY(ID)  
);;
```

```
CREATE TABLE Orders(  
    OrderID INT NOT NULL,  
    OrderNumber INT NOT NULL,  
    PersonID INT,  
    PRIMARY KEY (Order ID),
```

```
FOREIGN KEY (PersonID) REFERENCES Persons( PersonID)
);
```

Cách thêm FK:

```
ALTER TABLE Orders
```

```
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

5. CHECK

CHECK được sử dụng để giới hạn phạm vi giá trị có thể được đặt trong một cột.

Nếu bạn xác định 1 cột CHECK ràng buộc trên 1 cột, nó sẽ chỉ cho phép một số giá trị nhất định cho cột này.

Nếu bạn xác định CHECK ràng buộc trên 1 bảng, nó có thể giới hạn các giá trị trong các cột nhất định trong các cột khác trong hàng.

Eg:

```
CREATE TABLE Persons(
```

```
    ID INT NOT NULL,
```

```
    LastName VARCHAR(255) NOT NULL,
```

```
    FirstName VARCHAR(255) NOT NULL,
```

```
    Age int
```

```
    CHECK(Age ≥ 18)
```

```
);
```

```
SELECT * FROM Persons;
```

```
INSERT INTO Persons(ID,LastName, FirstName,Age)
```

```
VALUE(1,"Tung", "Bui",20); ⇒ oke
```

```
INSERT INTO Persons(ID,LastName, FirstName,Age)
```

```
VALUE(1,"Tung", "Bui",16); ⇒ Error.
```

6. DEFAULT

DEFAULT được sử dụng để đặt một giá trị mặc định cho một cột.

Giá trị mặc định sẽ được thêm vào tất cả các bản ghi mới, nếu không có giá trị nào khác được chỉ định.

Eg:

```
CREATE TABLE Persons(  
    ID INT NOT NULL,  
    LastName VARCHAR(255) NOT NULL,  
    FirstName VARCHAR(255) NOT NULL,  
    Age int  
    City VARCHAR(255) DEFAULT "Sandnes"  
);  
  
INSERT INTO Persons(ID, LastName, FirstName, Age)  
VALUES(1,"Tung","Bui",20);
```

⇒ Không có cột city nhưng mặc định là Sandnes

Với bảng đã được tạo, để tạo DEFAULT

```
ALTER TABLE Persons
```

```
ALTER City SET DEFAULT "Sandness";
```

Loại bỏ DEFAULT:

```
ALTER TABLE Persons
```

```
ALTER City DROP DEFAULT;
```

7. CREATE INDEX STATEMENT

Các chỉ mục được sử dụng để lấy dữ liệu từ cơ sở dữ liệu nhanh hơn so với cách khác. Người dùng không thể nhìn thấy các chỉ mục, chúng chỉ được sử dụng để tăng tốc độ tìm kiếm, truy vấn.

Eg:

```
SELECT * FROM Persons;
```

```
CREATE INDEX idx_lastname  
ON Persons(LastName);  
CREATE INDEX idx_pname  
ON Persons(LastName, FirstName);
```

DROP INDEX: Xóa 1 chỉ mục trong bảng

```
ALTER TABLE Persons  
DROP INDEX index_name;
```