BATCH : 146 - 149

LESSON : **SQL**

DATE : 25.07.2023

SUBJECT : **SQL - 2**

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ

TECHPRO EDUCATION

PostgreSQL

2. Ders

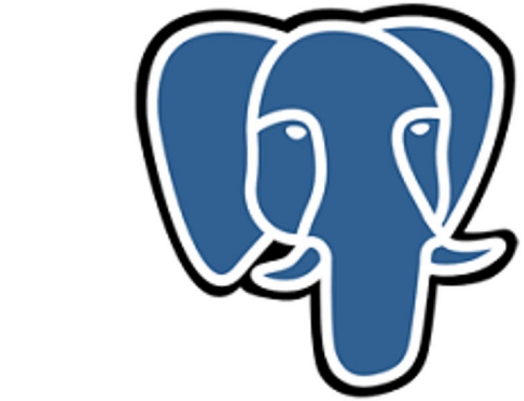
25.07.2023

B149 AWS & DevOps

B146 Cyber Security

Bugün ne yapıyoruz?

- CONSTRAINT



PostgreSQL

Constraint

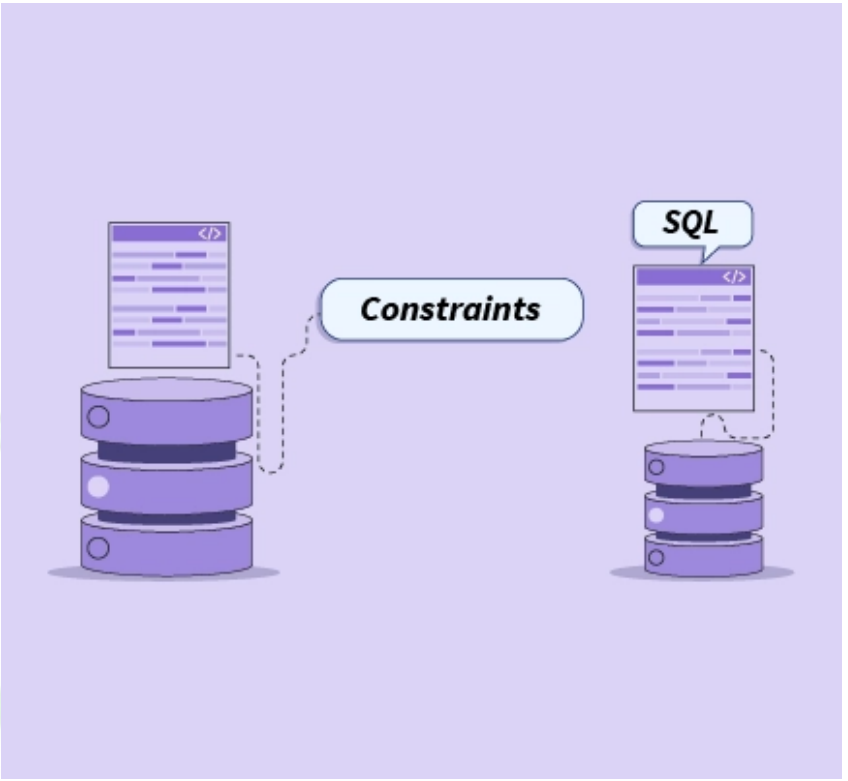
UNIQUE: Bir sütundaki tüm değerlerin Benzersiz/Tekrarsız yani tek olmasını sağlar.

NOT NULL: Bir sütunun NULL içermemesini sağlar. NOT NULL kısıtlaması için constraint ismi tanımlanmaz. Bu kısıtlama veri türünden hemen sonra yerleştirilir.

PRIMARY KEY: Bir sütunun NULL içermemesini ve sütundaki verilerin BENZERSİZ olmasını sağlar. (NOT NULL e UNIQUE)

FOREIGN KEY: Başka bir tablodaki Primary Key' i referans göstermek için kullanılır. Böylelikle, tablolar arasında ilişki kurulmuş olur.

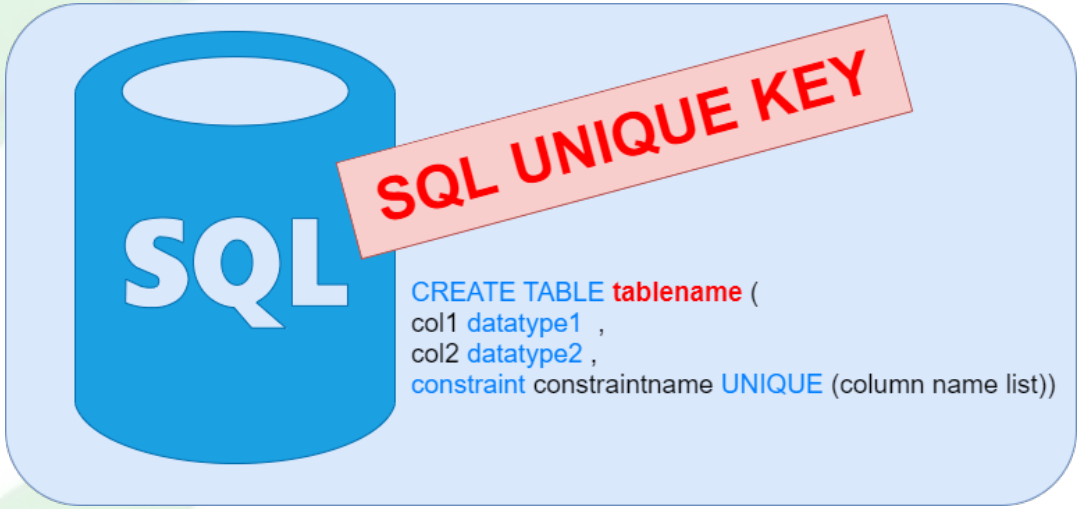
CHECK: Bir sütuna yerleştirilebilecek değer aralığını sınırlamak için kullanılır.



UNIQUE

Bir sütunun «tekrarsız» yapmak için, sütunun Data Type' dan sonra «UNIQUE» yazılır.

```
CREATE TABLE developers(
        id SERIAL,
        name VARCHAR(50),
        email VARCHAR(50) UNIQUE,
        salary REAL,
        prog_lang VARCHAR(20)
);
```

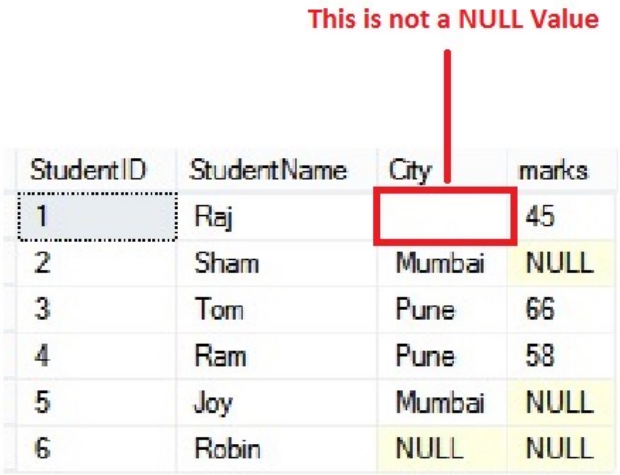


NOT NULL

Bir sütunun «boş bırakmamak» için, sütunun Data Type' dan sonra «NOT NULL» yazılır.

Önemli Not: Sütun değerinin '' (empty) ile NULL olması aynı şey değildir.

```
CREATE TABLE developers(
        id SERIAL,
        name VARCHAR(50) NOT NULL,
        email VARCHAR(50),
        salary REAL,
        prog_lang VARCHAR(20)
);
```

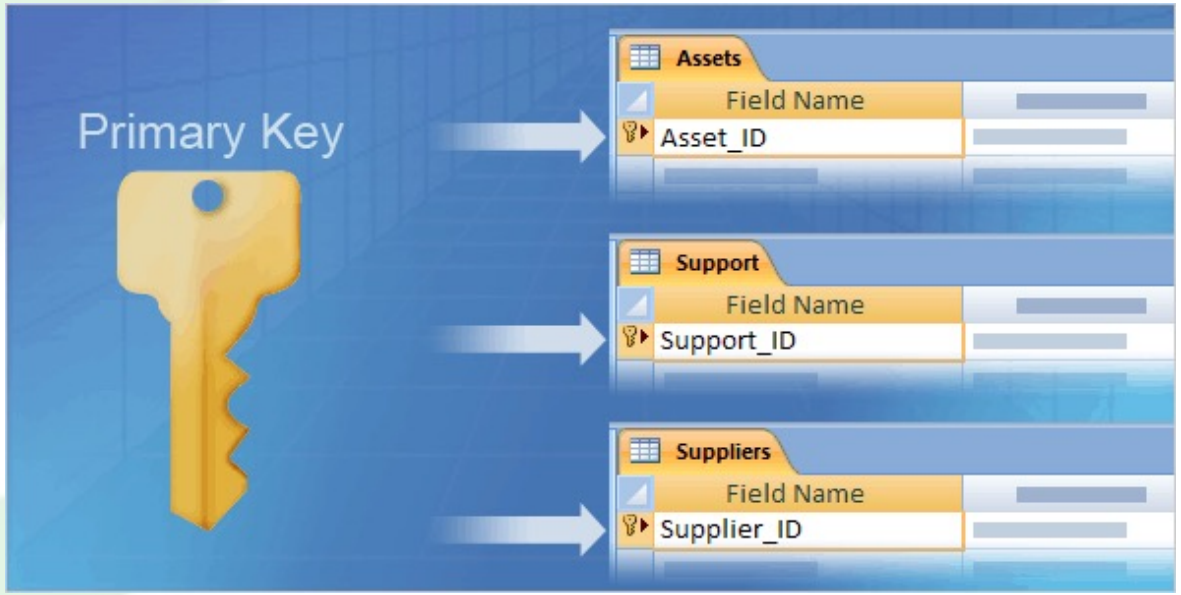


StudentID	StudentName	City	marks
1	Raj		45
2	Sham	Mumbai	NULL
3	Tom	Pune	66
4	Ram	Pune	58
5	Joy	Mumbai	NULL
6	Robin	NULL	NULL

PRIMARY KEY

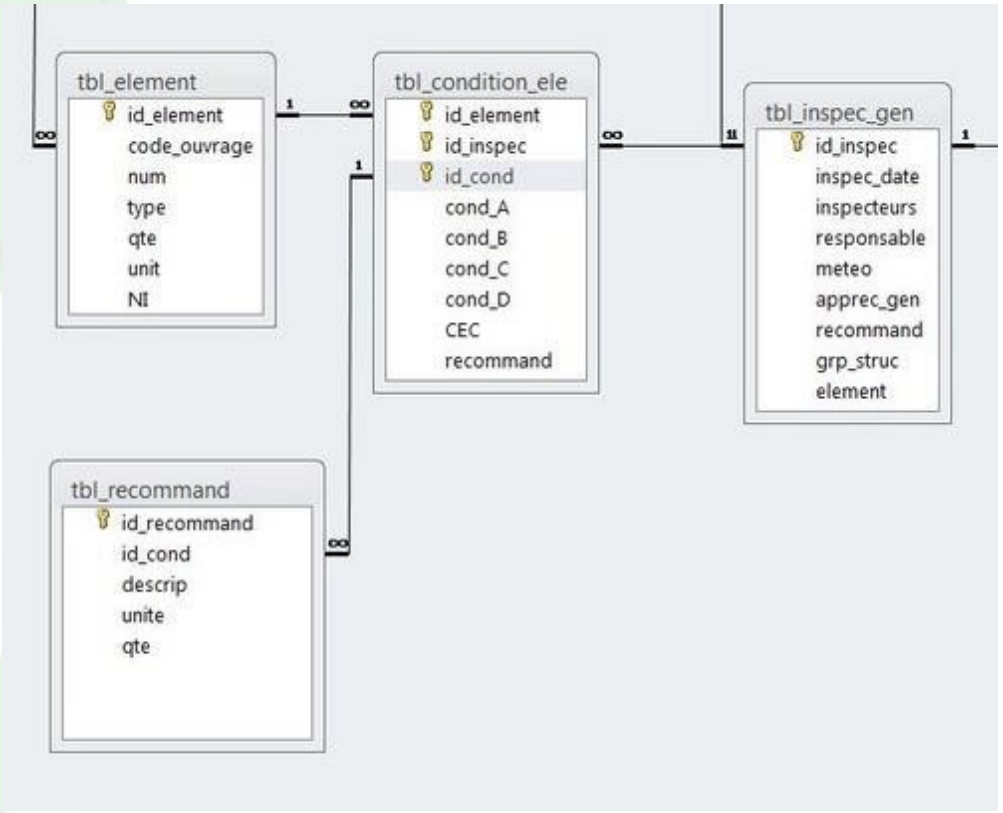
Bir veri tablosunuda yer alan her satır için bir vekil / tanımlayıcı görevi görür. «NOT NULL» ve «UNIQUE» olur. Sütunun Data Type' dan sonra «PRIMARY KEY» yazılır.

```
CREATE TABLE devops(
        id INT PRIMARY KEY,
        name VARCHAR(50),
        email VARCHAR(50),
        salary REAL,
        prog_lang VARCHAR(20)
);
```



Primary Key

- Primary Key değeri boş geçilemez ve **NULL** değer alamaz.
- Relational veri tabanlarında (relational database management system) mutlaka birincil anahtar olmalıdır.
- **Not** : Bir Tabloda **en fazla 1 tane** primary Key olabilir.
- **Not** : Primary Key benzersiz (**Unique**) olmalidir ama her unique data Primary Key değildir
- **Not** : Primary key her turlu datayi icerebilir. Sayi, String.
- **Not** : Her tabloda Primary Key olmasi **zorunlu değildir.**



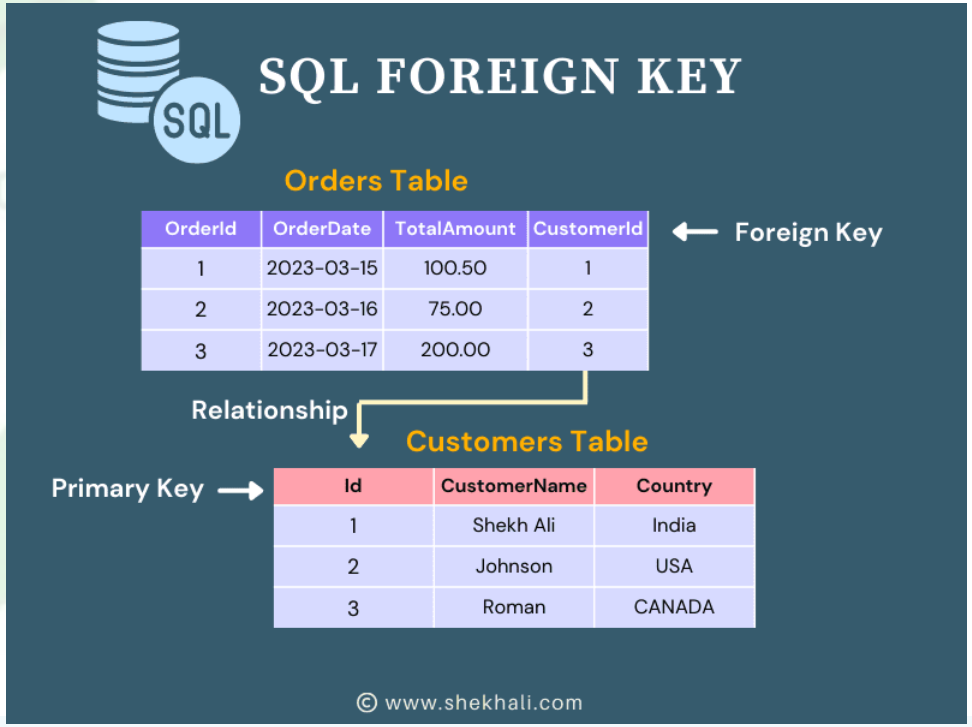
FOREIGN KEY

Bir veri tablosunuda yer alan her satır için bir vekil / tanımlayıcı görevi görür. «NOT NULL» ve «UNIQUE» olur. Sütunun Data Type' dan sonra «PRIMARY KEY» yazılır.

Not1: «Parent Table»' da olmayan bir id' ye sahip datayı «Child Table»' a ekleyemezsiniz.

Not2: «Child Table» silmeden «Parent Table» silemezsiniz. Önce «Child Table» silinir, sonra «Parent Table» silinir.

```
CREATE TABLE devops(
        id INT PRIMARY KEY,
        name VARCHAR(50),
        email VARCHAR(50),
        salary REAL,
        prog_lang VARCHAR(20)
);
```



CHECK

Bir sütunun değerlerini kontrol etmek için, sütunun Data Type' dan sonra «CHECK» yazılır.

```
CREATE TABLE developers(
        id SERIAL,
        name VARCHAR(50) CHECK(name <> ''),
        email VARCHAR(50),
        salary REAL,
        prog_lang VARCHAR(20)
);
```