

# Veritabanı Yönetim Sistemleri

Yapısal Sorgulama Dili (SQL) I



## Hedefler

- SQL standardizasyonu hakkında bilgi sahibi olmak
- SQL ile veritabanı oluşturup silebilmek
- SQL ile yeni tablolar oluşturabilmek ve silebilmek
- SQL ile tablolar üzerinde değişiklik yapabilmek

## İçindekiler

1. SQL Standardizasyonu
2. SQL Komut Tipleri
3. Veritabanı Oluşturma ve Silme
4. Tablo Oluşturma
5. Tablo Yapısını Değiştirme
6. Tablo Silme

## 1. SQL Standardizasyonu


SQL dili oluşturulduktan sonra SQL dilinin kullanımındaki farklılıkları ortadan kaldırmak ve bu konuda bir standart yakalamak için 1983 yılında ANSI ve ISO bir araya gelerek bir çalışma başlatmış ve 1987 yılında resmi olmayan ilk SQL standardı olan SQL1'i ortaya çıkarmıştır. Bu standardın yetersiz kalmasının ardından 1989 yılında SQL89 adında yine aynı kurumlar tarafından yeni bir standart geliştirilmiştir. Daha sonra 1992 yılında SQL2 diye bir standart çıkarılmıştır.


Yeni gelişmeler karşısında SQL diline birçok yeni özellikler (özellikle nesneye yönelik olmak üzere) eklenmiş ve yeni uygulamaların ihtiyaçlarını karşılamak için yeni eklentiler yapılmıştır. Böylece resmi olmamakla beraber SQL3 ortaya çıkmış oldu.

## 2. SQL Komut Tipleri

SQL, isminin belirttiği gibi sadece bir veri tabanı sorgulamak ve onun verisini idare etmek için değil onu tanımlamak için de kullanılır.

SQL komutları iki ana grupta incelenebilir.

 Veri tabanı ve tabloların oluşturulması ve yapısının değiştirilmesi için kullanılan komutlar

 Veri ekleme/güncelleme/silme için kullanılan komutlar

## 3. Veritabanı Oluşturma ve Silme

Bir veritabanı oluşturmak için aşağıdaki komut kullanılır.

```
CREATE DATABASE VeritabaniAdi;
```

Var olan bir veritabanını silmek için ise aşağıdaki komut kullanılır.

```
DROP DATABASE VeritabaniAdi;
```

## 4. Tablo Oluşturma

Tablo oluşturma işlemleri genellikle veri tabanı yönetim sisteminin sunduğu ara yüzler kullanılarak oluşturulur. Ancak bu ara yüzler arka planda gerekli SQL sorgularını hazırlayıp çalıştırarak yeni tabloları oluşturur.


Veri tabanı yöneticisi, tablo oluşturmak için direkt olarak kendi hazırladığı SQL sorgularını kullanabilir. Ayrıca programcılar da kendi programlarının içerisinde SQL sorguları çalıştırarak gerekli durumlarda yeni tablolar oluşturabilirler.


Veritabanında yeni tablolar oluşturmak için CREATE TABLE kullanılır. Kullanım şekli aşağıdaki gibidir.

```
CREATE TABLE <tablo adı> (<alanın adı> <alanın tipi> <alanın kısıtlaması>, ...);
```

CREATE TABLE komutu kullanılırken tablonun her alanı (sütunu) için veri tipi ve başka bazı parametreleri vardır. Bazı alanlar (sütunlar) için kısıtlamalarının gösterilmesi gereklidir.

Bu kısıtlamalardan bazıları aşağıdaki şekilde sayılabilir.

 NOT NULL, alan için bir değer mutlaka girilmesi gerektiğini gösterir. (Bu parametre daha çok birincil anahtarlar ve eşsiz alanlar için kullanılır.)

 DEFAULT, alan için bir başlangıç değeri verilmesinde kullanılır. Bu değer sabit bir değer olabileceği gibi, NULL değeri de olabilir.

### ÖRNEK

Örnek olarak, personel isimli bir tablo oluşturalım.

**SicilNo:** Sicil numarası için integer tipinde boş değer girilemeyecek şekilde birincil anahtar olacak bir alan oluşturalım.

**Ad:** Personelin adını tutmak için 8 karakterli boş değer girilemeyecek bir alan oluşturalım.

**Soyad:** Personelin soyadını tutmak için 10 karakterli boş değer girilemeyecek bir alan oluşturalım.

**EvliMi:** Personelin medeni halini tutmak için bir alan oluşturalım.

**Maas:** Personelin aldığı maaşı tutmak için bir alan oluşturalım.

```
CREATE TABLE Personel (
    SicilNo integer NOT NULL PRIMARY KEY,
    Ad char(8) NOT NULL,
    Soyad char(10) NOT NULL,
    EvliMi smallint,
    Maas money
);
```

Bu sorgu çalıştırıldığında veritabanında aşağıdaki görünüme sahip bir tablo oluşturulacaktır.

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas

### ÖRNEK

Örnek olarak, adres isimli başka bir tablo oluşturalım. Bu tabloda, birincil anahtar iki alanın (Ad ve Soyad) birleştirilmesiyle oluşturulsun.

```
CREATE TABLE Adres (  
    Ad char(8) NOT NULL,  
    Soyad char(10) NOT NULL,  
    PRIMARY KEY (Ad, Soyad),  
    Adres char(30),  
    Telefon char(12),  
    SicilNo integer  
);
```

Bu sorgu çalıştırıldığında veritabanında aşağıdaki görünüme sahip bir tablo oluşturulacaktır.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo

## 5. Tablo Yapısını Değiştirme

Tabloların yapısını değiştirmek için ALTER TABLE komutu kullanılır.

### Örnek

Örnek olarak, Ogrenci isimli tablodan DogumTarihi alanını silmek için aşağıdaki SQL komutu kullanılır.

```
ALTER TABLE Ogrenci DROP COLUMN DogumTarihi;
```

## 6. Tablo Silme

Veritabanındaki bir tabloyu silmek için de DROP TABLE komutu kullanılır.

### Örnek

Örnek olarak, Ogrenci tablosunu veritabanından silmek için aşağıdaki SQL komutu kullanılır.

```
DROP TABLE Ogrenci;
```