



Veritabanı Kavramları

Uzm. Murat YAZICI



Veritabanı Kavramları

Tablo

- Veritabanı içerisinde tutulacak verileri taşımak için kullanılır. Tablo, satır ve sütunlardan oluşan verilerin depolandığı veritabanı elemanıdır.
- Bir veritabanı içerisinde birden fazla tablo kullanılabilir.
- Örneğin, öğrenci bilgilerinin tutulduğu bir veritabanındaki öğrenci tablosu öğrencilerin **no**, **ad**, **soyad** gibi bilgilerini tutacaktır. Buradaki **no**, **ad** ve **soyad** bilgilerinin her biri bir sütunla gösterilecektir.

Veritabanı Kavramları

- **Satır** ise her bir öğrencinin tüm bilgileridir. **Satır** ifade yerine **kayıt** ifadesi de kullanılmaktadır.

Personel tablosu			
per_no	ad	soyad	gorev
20111001	Murat	Yazıcı	Şube Müdürü
20111002	Mustafa	Birinci	Personel Şefi
20111003	Hakan	Erkan	Mühendis

Tablo içerisinde her **satır** bir kaydı temsil eder. Örneğin, 20111001 nolu personele ait bilgiler.

Tablo içerisinde her **sütun** alanları gösterir. Tablomuzda, **per_no**, **ad**, **soyad**, **gorev** olmak üzere 4 adet alanımız vardır.



Veritabanı Kavramları

Veri Tipleri

- Oluşturulan veritabanında tutulan verilerin hepsi aynı türde değildir. Tutulacak veriye göre değişiklik gösterir. Örneğin, isim **karakter**, no bilgisi sadece **sayı**, doğum tarihi bilgisi **tarih** bilgisini içerir.
- Bir veritabanı oluşturulurken önce tablolar sonra da tablolardaki alanların veri tipleri tanımlanmak zorundadır.



SQL' de Veri Tipleri

SQL Veri Tipleri

String	CHAR (n)	Sabit uzunluklu karakter. Alfa numerik verileri tutar.
	VARCHAR (n)	Değişken uzunluklu karakter
Numeric	INT	-2147483648 ile +2147483647 arasındaki tam sayılar
	SMALLINT	-32768 ile +32767 arasındaki tamsayılar
	DECIMAL (x,y)	Ondalık sayı
	REAL	"
Date and Time	DATETIME	Tarih türündeki veri tipi



Veritabanı Kavramları

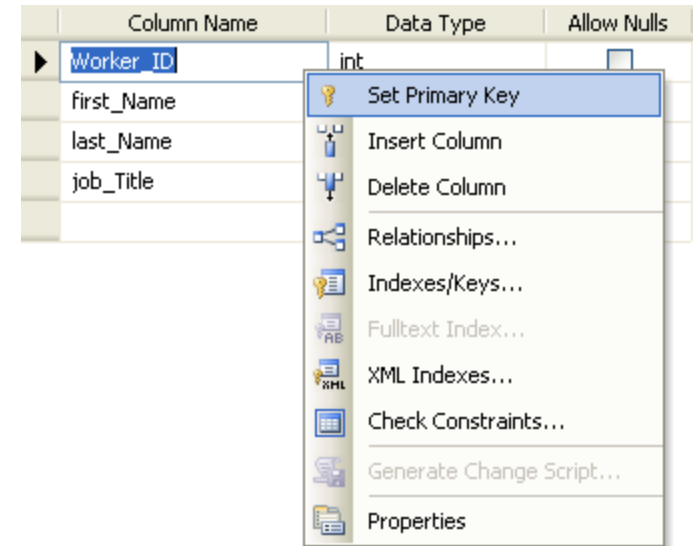
Primary Key (Birincil Anahtar)

- Tabloda tutulan verilerden benzersiz yani aynı değeri iki kez içermeyecek olan sütun birincil anahtar olarak belirlenir.
- Öğrenci bilgilerinin tutulduğu bir tabloda öğrencilerin birbirinden ayırt edilmesini sağlayan ne öğrencinin adı ne de sınıfıdır. Bizim için burda benzersiz (unique) olan bilgi **öğrenci numarası**dır.
- Öğrenci tablosu için **Primary Key öğrenci numarası** olabilir.

Primary Key (Birincil Anahtar)

- Tablolarda anahtar sütun tanımlama zorunluluğu yoktur, ama kullanılması tavsiye edilmektedir.
- Birincil anahtar olan sütun NULL (boş) değerleri veya birbirinin aynı olan değerleri içeremez.

Column Name	Data Type	Allow Nulls
Worker_ID	int	<input type="checkbox"/>
first_Name		
last_Name		
job_Title		



The screenshot shows a table with columns: Worker_ID (int, Allow Nulls:) and first_Name, last_Name, job_Title. A context menu is open over the Worker_ID column, with 'Set Primary Key' selected. Other options include: Insert Column, Delete Column, Relationships..., Indexes/Keys..., Fulltext Index..., XML Indexes..., Check Constraints..., Generate Change Script..., and Properties.



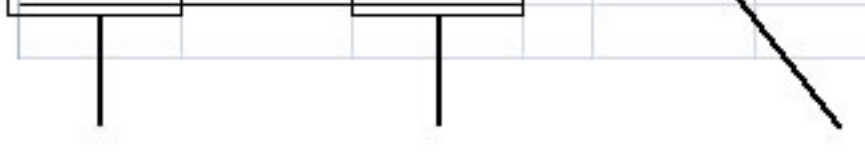
Veritabanı Kavramları

Foreign Key (Yabancı Anahtar)

- Bir tabloda olan bir anahtar alan başka bir tabloda da yer alıyorsa, yani tablodaki bir anahtar alan başka bir tablodaki anahtar alanı gösteriyorsa buna **“yabancı anahtar”** denir.
- Yabancı anahtar genelde diğer tablolarla ilişki kurmak için kullanılır.
- Birbiri arasında ilişki bulunan iki tablodan birisinden diğer tablodaki birincil anahtara başvuran sütun **yabancı anahtar** olarak adlandırılır.

Foreign Key (Yabancı Anahtar)

ogrenci			bolum	
No	Adı	Böl. Kodu	Kodu	Adı
53	Murat	4	2	Bilgisayar Programcılığı
34	Mustafa	4	4	Elektrik
28	Ahmet	2	5	İnşaat
21	Ayhan	2		
49	Kadir	5		



Primary Key

Foreign Key

Primary Key



Veritabanı Kavramları

İlişkilendirme (Joining)

- Tek sorgu ile birden fazla tablodan bilgi alma işlemine **ilişkilendirme** denilir.
- Bazı veritabanı yazılımlarında sorgu kullanmadan ilişkilendirme yapılabilir. (Örneğin, MS Access veritabanı)
- Hazırlanan projede programlama dili kullanılıyorsa ilişkilendirme işlemi sorgu ile yapılmalıdır.



Veritabanı Kavramları

İlişkilendirme (Joining)

- Öğrenci ve bölüm bilgilerinin ayrı ayrı tablolarda tutulduğu bir veritabanında öğrenci bilgileri ve öğrencinin kayıtlı olduğu bölümün adını öğrenmek için ilişki kurulması gerekmektedir.
- İlişkilendirme ayrıca veri tutarlılığını sağlamak için de kullanılır. Örneğin, bölüm tablosunda olmayan bölüm kodunu öğrenci tablosunda kullanılmasının engellenmesi, öğrenci tablosundan öğrenci silindiğinde diğer tablolarda o öğrenciye ait tüm bilgilerin silinmesi veya değiştirilmesi için kullanılabilir.

İlişkilendirme (Joining)

MURAT-PC.test - dbo.ogrenci				
	ogrno	ad	soyad	fakulte
▶	125213	MURAT	YAZICI	1
	125214	HAKAN	ERKAN	1
	125220	CUMALİ	ÇELİK	2
	125224	HÜSEYİN	ERTEKİN	3

MURAT-PC.test - dbo.bolum		
	id	fakulteAdi
▶	1	MÜHENDİSLİK FAKÜLTESİ
	2	TIP FAKÜLTESİ
	3	FEN-EDEBİYAT FAKÜLTESİ
	4	EĞİTİM FAKÜLTESİ
	5	İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ

İlişkilendirme (Joining)

SQLQuery2.sql - ...-PC\MURAT (52))*

```
SELECT ogrenci.ogrno, ogrenci.ad, ogrenci.soyad, bolum.fakulteAdi  
FROM ogrenci, bolum  
WHERE ogrenci.fakulte=bolum.id
```

Results Messages

	ogrno	ad	soyad	fakulteAdi
1	125213	MURAT	YAZICI	MÜHENDİSLİK FAKÜLTESİ
2	125214	HAKAN	ERKAN	MÜHENDİSLİK FAKÜLTESİ
3	125220	CUMALİ	ÇELİK	TIP FAKÜLTESİ
4	125224	HÜSEYİN	ERTEKİN	FEN-EDEBİYAT FAKÜLTESİ