



# Veritabanı Dersi

## Teoriden Pratięe

Çaęıltay N.E., Tokdemir G.

Veritabanı Sistemleri Dersi -Bölüm XI: IV. Aşama: Fiziksel Model  
Çaęıltay, N., Tokdemir, G.

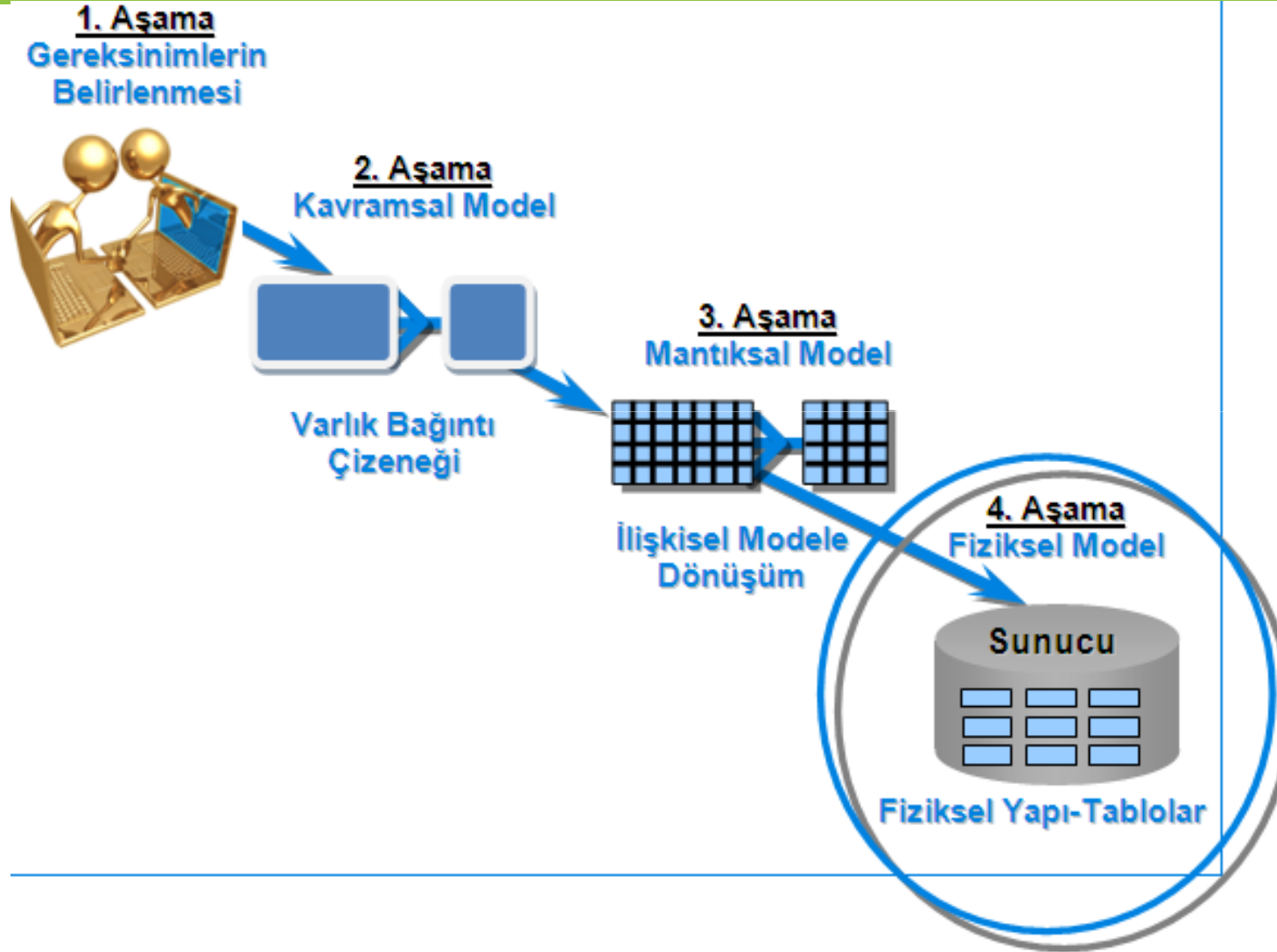
# BÖLÜM 11

## IV. Aşama: Fiziksel Model

- Tablo
- Veri Tipleri
- Tablo Oluşturma
- Tablo Kısıtları
- Örnek Veritabanı



# IV. Aşama: Fiziksel Model





# Tablo

**Dikeç**

kullanıcı			
	kod	ad	soyad
+	Ayşe	Ayşe	Dalgıç
+	erika	Erdem	Şenulmaz
+	gonca	Gonca	Gökay
+	Hakan	Hakan	Dalgıç
+	k_1960	Kumru	Aytan
+	Kursat	Kürşat	Çalışkan
+	mustafa	Mustafa	Karadağlı

**Dizeç**



# Veri Tipleri: Karakter

Sabit Uzunlukta

CHAR (*uzunluk*)

ad CHAR (15)

A	l	i												
---	---	---	--	--	--	--	--	--	--	--	--	--	--	--



# Veri Tipleri: Karakter

Değişken Uzunlukta

`VARCHAR (uzunluk)`

`ad VARCHAR (15)`

A	l	i
---	---	---



# Veri Tipleri: Nümerik/Sayısal

NUMBER

not NUMBER

NUMBER (*uzunluk, ondalık*)

ödeme NUMBER (10, 2)



# Veri Tipleri: Tarih

doğumTarihi DATE





# Tablo Oluşturma

```
CREATE TABLE [veritabanıAdı.[sahip].|sahip.] tabloAdı
( { dikeçAdı veriTipi [ DEFAULT varsayılanAnlatım ]
  [ dikeçKısıtı [ ... ] ] |
  tabloKısıtı
  [ CONSTRAINT kısıtAdı ]
  { NOT NULL | NULL | UNIQUE | PRIMARY KEY | CHECK (anlatım)
  CONSTRAINT kısıtAdı } { UNIQUE ( dikeçAdı [, ... ] )
  | PRIMARY KEY ( dikeçAdı [, ... ] ) | CHECK (anlatım) |
  FOREIGN KEY ( dikeçAdı [, ... ] )
  REFERENCES referansTablosu [ ( refDikeç [, ... ] ) ] ]
  [ ON DELETE eylem] [ ON UPDATE eylem ] }
```



# Tablo Kısıtları

## Birincil Anahtar Kısıtı

```
CREATE TABLE kitap (  
    isbn      NUMBER,  
    başlık  CHAR(50),  
    CONSTRAINT PK_kitap PRIMARY KEY (isbn));
```

```
CREATE TABLE kitap2 (  
    isbn      NUMBER PRIMARY KEY,  
    başlık  CHAR(50));
```

```
CREATE TABLE şahıs (  
    ad        CHAR(15),  
    soyAd    CHAR(15),  
    dTarih    DATE,  
    adres     VARCHAR(50),  
    tel       CHAR(15),  
    CONSTRAINT PK_şahıs PRIMARY KEY (ad, soyAd));
```



# Tablo Kısıtları

## Boş Geçmez Kısıtı

```
CREATE TABLE il (  
    ilKodu NUMBER,  
    ilAdı CHAR(30)  
    CONSTRAINT ilAdı_null NOT NULL,  
    CONSTRAINT PK_il PRIMARY KEY (ilKodu) );
```

```
CREATE TABLE il2 (  
    ilKodu NUMBER PRIMARY KEY,  
    ilAdı CHAR(30) NOT NULL);
```



# Tablo Kısıtları

## Varsayılan Değer Kısıtı

```
CREATE TABLE kitap3 (  
    isbn      NUMBER,  
    başlık  CHAR(50),  
    fiyat    NUMBER DEFAULT 10,  
    CONSTRAINT PK_kitap PRIMARY KEY (isbn));
```



# Tablo Kısıtları

## Benzersiz Kısıtı

```
CREATE TABLE il3 (  
    plakaNo NUMBER PRIMARY KEY,  
    ad VARCHAR(40) UNIQUE );
```



# Tablo Kısıtları Denetim Kısıtı

```
CREATE TABLE il4 (  
    plakaNo NUMBER PRIMARY KEY,  
    CONSTRAINT ilPlakaNo CHECK (plakaNo BETWEEN 1 AND 81),  
    ad VARCHAR(40) UNIQUE );
```

```
CREATE TABLE malzeme (  
    kod NUMBER PRIMARY KEY,  
    durum CHAR(1) DEFAULT A  
    CONSTRAINT malzemeDurumNULL NOT NULL, CHECK (durum IN  
    ('A', 'X')));
```



# Tablo Kısıtları

## Yabancı Anahtar Kısıtı

```
CREATE TABLE şahıs2 (  
    ad          CHAR(15),  
    soyAd       CHAR(15),  
    dTarih      DATE,  
    dYeri        NUMBER,  
    adres        VARCHAR(50),  
    tel          CHAR(15),  
    CONSTRAINT PK_şahıs2 PRIMARY KEY (ad, soyAd),  
    CONSTRAINT FK_şahıs2_dYeri FOREIGN KEY (dYeri)  
        REFERENCES il5(ilKodu));
```



# Tablo Kısıtları Sil İşlem Kısıtı

```
CREATE TABLE şahıs3 (  
    ad          CHAR(15),  
    soyAd       CHAR(15),  
    dTarih      DATE,  
    dYeri        NUMBER,  
    adres        VARCHAR(50),  
    tel          CHAR(15),  
    CONSTRAINT PK_şahıs3 PRIMARY KEY (ad, soyAd),  
    CONSTRAINT FK_şahıs3_dYeri FOREIGN KEY(dYeri)  
        REFERENCES il(ilkodu) ON DELETE NO ACTION);
```





# Tablo Kısıtları

## Değiştir İşlem Kısıtı

```
CREATE TABLE şahıs4 (  
    ad          CHAR(15),  
    soyAd      CHAR(15),  
    dTarih     DATE,  
    dYeri      NUMBER,  
    adres      VARCHAR(50),  
    tel        CHAR(15),  
    CONSTRAINT PK_şahıs PRIMARY KEY (ad, soyAd),  
    CONSTRAINT FK_şahıs_dYeri FOREIGN KEY(dYeri)  
        REFERENCES il(ilKodu) ON UPDATE NO ACTION);
```



# Tablo Kısıtları

## Değeri Sil

```
CREATE TABLE şahıs5 (  
    ad          CHAR(15),  
    soyAd       CHAR(15),  
    dTarih      DATE,  
    dYeri        NUMBER DEFAULT 6,  
    adres       VARCHAR(50),  
    tel         CHAR(15),  
    CONSTRAINT PK_şahıs5 PRIMARY KEY (ad, soyAd),  
    CONSTRAINT FK_şahıs5_dYeri FOREIGN KEY (dYeri)  
        REFERENCES il (ilkodu) ON DELETE SET NULL);
```



# Tablo Kısıtları İlgili Diğer Kayıtları da Sil

```
CREATE TABLE şahıs6 (  
    ad          CHAR(15),  
    soyAd       CHAR(15),  
    dTarih      DATE,  
    dYeri        NUMBER DEFAULT 6,  
    adres        VARCHAR(50),  
    tel          CHAR(15),  
    CONSTRAINT PK_şahıs6 PRIMARY KEY (ad, soyAd),  
    CONSTRAINT FK_şahıs6_dYeri FOREIGN KEY (dYeri)  
        REFERENCES İL(ilKodu) ON DELETE CASCADE);
```



# Tablo Kısıtları Silinemez Kılma Sil

```
CREATE TABLE kitap4 (  
    isbn      NUMBER,  
    başlık  CHAR(50),  
    fiyat    NUMBER DEFAULT 10 DISABLE,  
    CONSTRAINT PK_kitap4 PRIMARY KEY (isbn));
```



# Örnek Veritabanı Fiziksel Modeli

```
soruTipleri (kod, açıklama)
```

```
CREATE TABLE soruTipleri (  
    kod        NUMBER,  
    açıklama  VARCHAR(150),  
    CONSTRAINT PK_soruTipleri PRIMARY KEY (kod));
```



# Örnek Veritabanı Fiziksel Modeli

```
okumalar (kod, ad, dosyaAdı, ağırlık)
```

```
CREATE TABLE okumalar (  
    kod NUMBER,  
    ad CHAR(250),  
    dosyaAdı VARCHAR(50),  
    ağırlık NUMBER,  
    CONSTRAINT PK_okumalar PRIMARY KEY (kod) );
```



# Örnek Veritabanı Fiziksel Modeli

`sorular` (`kod`, `sıra`, `ağırlık`, `dosyaAdı`, `soruTipi` (*soruTipleri* tablosuna referans verir), `okumaKodu` (*okumalar* tablosuna referans verir) )

```
CREATE TABLE sorular (  
    kod NUMBER,  
    sıra NUMBER,  
    ağırlık NUMBER,  
    dosyaAdı VARCHAR(50),  
    soruTipi NUMBER,  
    okumaKodu NUMBER,  
    CONSTRAINT FK_sorular PRIMARY KEY (kod),  
  
    CONSTRAINT FK_sorularSoruTipi FOREIGN KEY (soruTipi)  
        REFERENCES soruTipleri,  
    CONSTRAINT FK_sorularOkumaKod FOREIGN KEY (okumaKodu)  
        REFERENCES okumalar);
```



# Örnek Veritabanı Fiziksel Modeli

```
kullanıcı(kod, eposta, şifre, faks, dTarihi, açıklama, adres, ad, soyAd)
```

```
CREATE TABLE kullanıcı (  
    kod CHAR(8),  
    eposta CHAR(20),  
    şifre CHAR(15),  
    faks CHAR(25),  
    dTarihi DATE,  
    açıklama VARCHAR(150),  
    ad CHAR(20),  
    soyAd CHAR(20),  
    adres VARCHAR(50),  
    CONSTRAINT PK_kullanıcı PRIMARY KEY (kod));
```





# Örnek Veritabanı Fiziksel Modeli

```
ofis(kod, tel, tipi)
```

```
CREATE TABLE ofis(  
    kod        NUMBER,  
    tel        CHAR(25),  
    tipi       CHAR(20),  
    CONSTRAINT PK_ofis PRIMARY KEY (kod));
```



# Örnek Veritabanı Fiziksel Modeli

```
soruSeçenekleri (soruKodu, kod, (soruKodu, sorular tablosuna referans verir), puan, metin, anahtarKelime)
```

```
CREATE TABLE soruSeçenekleri (  
    soruKodu NUMBER,  
    kod NUMBER,  
    puan NUMBER,  
    metin VARCHAR(200),  
    anahtarKelime VARCHAR(200),  
    CONSTRAINT PK_soruSeçenekleri  
        PRIMARY KEY (soruKodu, kod),  
    CONSTRAINT FK_soruSeçenekleri FOREIGN KEY (soruKodu)  
        REFERENCES sorular);
```



# Örnek Veritabanı Fiziksel Modeli

*öğretmen (kullanıcıKodu (kullanıcı tablosuna referans verir), ofis (ofis tablosuna referans verir), zümreBşk (öğretmen tablosuna referans verir) )*

```
CREATE TABLE öğretmen (  
    kullanıcıKodu CHAR(8),  
    ofis NUMBER,  
    zümreBşk CHAR(8),  
    CONSTRAINT PK_öğretmen PRIMARY KEY (kullanıcıKodu),  
    CONSTRAINT FK_öğretmenKod FOREIGN KEY (kullanıcıKodu)  
        REFERENCES kullanıcı,  
    CONSTRAINT FK_öğretmenOfis FOREIGN KEY (ofis)  
        REFERENCES ofis,  
    CONSTRAINT FK_öğretmenZümreBşk FOREIGN KEY (zümreBşk)  
        REFERENCES öğretmen);
```



# Örnek Veritabanı Fiziksel Modeli

öğrenci (kullanıcıKodu (*kullanıcı tablosuna referans verir*))

```
CREATE TABLE öğrenci (  
    kullanıcıKodu CHAR(8),  
    CONSTRAINT PK_öğrenci PRIMARY KEY (kullanıcıKodu),  
    CONSTRAINT FK_öğrenciKod FOREIGN KEY (kullanıcıKodu)  
        REFERENCES kullanıcı);
```



# Örnek Veritabanı Fiziksel Modeli

`sınavSoruları` (`soruKodu`, `sınavKodu` (*soruKodu sorular tablosunu referans verir, sınavKodu sınavlar tablosunu referans verir*))

```
CREATE TABLE sınavSoruları (  
    soruKodu          NUMBER,  
    sınavKodu        NUMBER,  
    CONSTRAINT PK_sınavSoruları  
        PRIMARY KEY (soruKodu, sınavKodu),  
    CONSTRAINT FK_sınavSorularıSoru  
        FOREIGN KEY (soruKodu) REFERENCES sorular,  
    CONSTRAINT FK_sınavSorularıSınav  
        FOREIGN KEY (sınavKodu) REFERENCES sınavlar);
```



# Örnek Veritabanı Fiziksel Modeli

`sınavOkumaları` (`sınavKodu`, `okumaKodu` (*`sınavKodu` sınavlar tablosunu referans verir, `okumaKodu` okumalar tablosunu referans verir*), `sıra`)

```
CREATE TABLE sınavOkumaları (  
    sınavKodu          NUMBER,  
    okumaKodu         NUMBER,  
    sıra              NUMBER,  
    CONSTRAINT PK_sınavSoruları  
        PRIMARY KEY (sınavKodu, okumaKodu),  
    CONSTRAINT FK_sınavOkumalarıSınav  
        FOREIGN KEY (sınavKodu) REFERENCES sınavlar,  
    CONSTRAINT FK_sınavOkumalarıOkuma  
        FOREIGN KEY (okumaKodu) REFERENCES okumalar);
```



# Örnek Veritabanı Fiziksel Modeli

*öğrenciSınavları (sınavKodu, öğrenciKodu (sınavKodu sınavlar tablosunu referans verir, öğrenciKodu öğrenci tablosunu referans verir), açıklama, süre, tarih, tdoğru, tyanlış, sonuç)*

```
CREATE TABLE öğrenciSınavları (  
    sınavKodu          NUMBER,  
    öğrenciKodu       NUMBER,  
    açıklama          VARCHAR(100),  
    süre              NUMBER,  
    tarih             DATE,  
    tdoğru            NUMBER,  
    tyanlış           NUMBER,  
    sonuç             NUMBER,  
    CONSTRAINT PK_öğrenciSınavları  
        PRIMARY KEY (sınavKodu, öğrenciKodu),  
    CONSTRAINT FK_öğrenciSınavlarıSınav  
        FOREIGN KEY (sınavKodu) REFERENCES sınavlar,  
    CONSTRAINT FK_öğrenciSınavlarıÖğrenci  
        FOREIGN KEY (öğrenciKodu) REFERENCES öğrenci);
```



# Örnek Veritabanı Fiziksel Modeli

`hazırlayan` (`sınavKodu`, `öğretmenKodu` (*`sınavKodu` sınavlar tablosunu referans verir, `öğretmenKodu` öğretmen tablosunu referans verir*), `tarikh`)

```
CREATE TABLE hazırlayan(  
    sınavKodu NUMBER,  
    öğretmenKodu CHAR(8),  
    tarih DATE,  
    CONSTRAINT PK_hazırlayan  
        PRIMARY KEY (sınavKodu, öğretmenKodu),  
    CONSTRAINT FK_hazırlayanSınav  
        FOREIGN KEY (sınavKodu) REFERENCES sınavlar,  
    CONSTRAINT FK_hazırlayanÖğretmen  
        FOREIGN KEY (öğretmenKodu) REFERENCES öğretmen);
```





# Örnek Veritabanı Fiziksel Modeli

`cevaplar` (`sınavKodu`, `öğrenciKodu`, `soruKodu`, `seçenekKodu`  
(*{`sınavKodu`, `öğrenciKodu`}* *öğrenciSınavları* tablosunu referans verir, *{`soruKodu`, `seçenekKodu`}* *soruSeçenekleri* tablosunu referans verir), `puan`, `değer`)

```
CREATE TABLE cevaplar(  
    sınavKodu NUMBER,  
    öğrenciKodu CHAR(8),  
    soruKodu NUMBER,  
    seçenekKodu NUMBER,  
    puan NUMBER,  
    değer NUMBER,  
    CONSTRAINT PK_cevaplar PRIMARY KEY  
        (sınavKodu, öğrenciKodu, soruKodu, seçenekKodu),  
    CONSTRAINT FK_cevaplarSınav  
        FOREIGN KEY (sınavKodu, öğrenciKodu)  
        REFERENCES öğrenciSınavları,  
    CONSTRAINT FK_cevaplarSeçenek  
        FOREIGN KEY (soruKodu, seçenekKodu)  
        REFERENCES soruSeçenekleri);
```



# Örnek Veritabanı Fiziksel Modeli

```
kullanıcıTel (kullanıcıKodu, tel (kullanıcıKodu kullanıcı tablosunu referans verir))
```

```
CREATE TABLE kullanıcıTel(  
    kullanıcıKodu    CHAR(8),  
    tel              VARCHAR(50),  
    CONSTRAINT PK_tel  
        PRIMARY KEY (kullanıcıKodu, tel),  
    CONSTRAINT FK_telKullanıcı  
        FOREIGN KEY (kullanıcıKodu) REFERENCES kullanıcı);
```