

## Veri Tabanı Sistemleri Bağıntı Modeli

H. Turgut Uyar Şule Öğüdücü

2002-2012

1 / 101

## License



©2002-2012 T. Uyar, Ş. Öğüdücü

You are free:

- ▶ to Share – to copy, distribute and transmit the work
- ▶ to Remix – to adapt the work

Under the following conditions:

- ▶ Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- ▶ Noncommercial – You may not use this work for commercial purposes.
- ▶ Share Alike – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Legal code (the full license):

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

2 / 101

## Konular

### Bağıntı Modeli

- Giriş
- Veri Tipleri
- Bağıntı Yönetimi
- Veriyi Değiştirme
- Başvuru Bütünlüğü

### SQL

- Veri Tipleri
- Tablo Yönetimi
- Veriyi Değiştirme
- Başvuru Bütünlüğü

3 / 101

## Veri Modelleri

- ▶ önceki modeller:
  - ▶ ters liste
  - ▶ hiyerarşik
  - ▶ ağ
- ▶ bağıntı modeli:
  - ▶ Dr. E. F. Codd, 1970
- ▶ sonraki modeller:
  - ▶ nesne
  - ▶ nesne / bağıntı

4 / 101

## Bağıntı Modeli

- ▶ veri **bağıntılar** şeklinde modellenir:  
 $\alpha \subseteq A \times B \times C \times \dots$
- ▶ bağıntılar **bağıntı değişkenlerine** atanır
- ▶ bağıntının her elemanı bir **çoklu**
- ▶ elemanların her verisi bir **nitelik**
- ▶ bağıntılar tablolarla temsil edilir
  - ▶ kullanıcı bütün veriyi tablolar şeklinde *algılamalı*
  - ▶ bağıntı → tablo, çoklu → satır, nitelik → sütun

5 / 101

## Bağıntı Örneği

Örnek (film bağıntısı)

MOVIE

TITLE	YEAR	DIRECTOR	SCORE	VOTES
Usual Suspects	1995	Bryan Singer	8.7	3502
Suspiria	1977	Dario Argento	7.1	1004
Being John Malkovich	1999	Spike Jonze	8.3	13809
...	...	...	...	...

- ▶ bağıntı değişkeninin adı MOVIE
- ▶ (Usual Suspects, 1995, Bryan Singer, 8.7, 3502) film bağıntısının bir çoklusu
- ▶ YEAR, film bağıntısının bir niteliği

6 / 101

## Bağıntı Yapısı

### bağıntı başlığı

- bağıntının nitelikleri kümesi
- veri tanımlama dili komutlarından etkilenir

### bağıntı gövdesi

- bağıntıdaki çoklular kümesi
- veri işleme dili komutlarından etkilenir

7 / 101

## Bağıntı Yüklemi

### Tanım

**bağıntı yüklemi:** bağıntının "anlamını" ifade eden cümle

- her çoklu yükleme göre *Doğru* ya da *Yanlış* değerini alır

8 / 101

## Bağıntı Yükleme Örneği

### Örnek (film bağıntısının yüklemi)

- TITLE başlıklı film YEAR yılında, DIRECTOR tarafından çekilmiştir; verilen VOTES oyun ortalaması SCORE'dur.
- (Suspiria, 1977, Dario Argento, 1004, 7.1) çoklusu doğrudur
- (Suspiria, 1978, Dario Argento, 1004, 7.1) çoklusu yanlıştır

9 / 101

## Çoklu Sırası

- çokluların sırası önemsizdir

### Örnek

- şu iki bağıntı eşdeğerlidir:

TITLE	...
Usual Suspects	...
Suspiria	...
Being John Malkovich	...

TITLE	...
Suspiria	...
Being John Malkovich	...
Usual Suspects	...

10 / 101

## Nitelik Sırası

- niteliklerin sırası önemsizdir

### Örnek

- şu iki bağıntı eşdeğerlidir:

TITLE	YEAR	...
Usual Suspects	1995	...
Suspiria	1977	...
Being John Malkovich	1999	...

YEAR	TITLE	...
1995	Usual Suspects	...
1977	Suspiria	...
1999	Being John Malkovich	...

11 / 101

## Nitelik Değerleri

- niteliklere verilen değerler tek boyutlu olmalıdır
  - dizi, liste, kayıt gibi değerlere izin verilmez

### Örnek (birden fazla yönetmen)

TITLE	...	DIRECTORS	...
...	...	...	...
Matrix	...	Andy Wachowski, Lana Wachowski	...
...	...	...	...

12 / 101

## Boş Değer

- ▶ çoklu için o niteliğin değeri bilinmiyor
- ▶ çoklu o nitelik için bir değer taşıyor

### Örnek

- ▶ filmin yönetmeni bilinmiyor
- ▶ film için oy kullanılmamış, o yüzden SCORE yok

13 / 101

## Varsayılan Değer

- ▶ boş değer yerine varsayılan bir değer kullanılabilir
  - ▶ niteliğin alabileceği geçerli değerlerden biri olmamalı

### Örnek

- ▶ SCORE niteliği 1.0 ile 10.0 arasında değer alıyorsa varsayılan değeri 0.0 seçilebilir

14 / 101

## Eş Çoklular

- ▶ bir bağıntıda birbirinin eşi çoklular bulunamaz
  - ▶ her çoklu diğerlerinden ayırt edilebilmelidir

### Örnek



TITLE	YEAR	DIRECTOR	SCORE	VOTES
Usual Suspects	1995	Bryan Singer	8.7	3502
Suspiria	1977	Dario Argento	7.1	1004
Being John Malkovich	1999	Spike Jonze	8.3	13809
...	...	...	...	...
Suspiria	1977	Dario Argento	7.1	1004
...	...	...	...	...

15 / 101

## Anahtarlar

- ▶  $B$  bağıntının bütün nitelikleri kümesi olsun ve  $A \subseteq B$  olsun
- ▶  $A$ 'nın bir anahtar adayı olabilmesi için şu koşullar sağlanmalı:
  - ▶ **eşsizlik**: herhangi iki çoklu  $A$ 'da yer alan bütün nitelikler için aynı değeri taşımazlar
  - ▶ **indirgenemezlik**:  $A$ 'nın hiçbir altkümesi eşsizlik özelliğini sağlamaz
- ▶ her bağıntının en az bir anahtar adayı vardır

16 / 101

## Anahtar Adayı Örneği

### Örnek (film bağıntısı için anahtar adayları)

- ▶ {TITLE}
- ▶ {TITLE, YEAR}
- ▶ {TITLE, DIRECTOR}
- ▶ {TITLE, YEAR, DIRECTOR}

17 / 101

## Anahtar Eşdeğeri

- ▶ bir **doğal anahtar** bulunamıyorsa bir **anahtar eşdeğeri** tanımlanabilir
- ▶ kimlik niteliği
  - ▶ değerinin ne olduğunun önemi yok
  - ▶ sistem tarafından üretilabilir

18 / 101

## Anahtar Eşdeğeri Örneği

### Örnek

MOVIE#	TITLE	YEAR	DIRECTOR	SCORE	VOTES
...	...	...	...	...	...
6	Usual Suspects	1995	Bryan Singer	...	...
1512	Suspiria	1977	Dario Argento	...	...
70	Being John Malkovich	1999	Spike Jonze	...	...
...	...	...	...	...	...

- ▶ {MOVIE#} anahtar adaydır
- ▶ {MOVIE#, TITLE} anahtar adayı değildir

19 / 101

## Birincil Anahtar

- ▶ bir bağıntının birden fazla anahtar adayı varsa:
  - ▶ biri **birincil anahtar** seçilir
  - ▶ diğerleri **anahtar seçeneği** olur
- ▶ her bağıntının bir birincil anahtarı bulunması zorunludur
- ▶ birincil anahtarın parçası olan hiçbir niteliğin değeri hiçbir çokluda boş olamaz

20 / 101

## Birincil Anahtar Örneği

- ▶ birincil anahtarı oluşturan nitelikler altı çizili gösterilir

### Örnek

<u>MOVIE#</u>	<u>TITLE</u>	<u>YEAR</u>	<u>DIRECTOR</u>	<u>SCORE</u>	<u>VOTES</u>
...	...	...	...	...	...
6	Usual Suspects	1995	Bryan Singer	...	...
1512	Suspiria	1977	Dario Argento	...	...
70	Being John Malkovich	1999	Spike Jonze	...	...
...	...	...	...	...	...

21 / 101

## Veri Tipleri

- ▶ aynı niteliğe ilişkin değerler aynı tanım kümesinden seçilmeli
  - ▶ karşılaştırma işlemi ancak aynı tanım kümesinden seçilmiş değerler arasında anlamlıdır
- ▶ pratikte veri tipleri kullanılır

22 / 101

## Tanım Kümesi Örneği

### Örnek

- ▶ TITLE başlıklar kümesinden, YEAR yıllar kümesinden, DIRECTOR yönetmenler kümesinden, ...
- ▶ veri tipi kullanılırsa:  
TITLE katar, YEAR tamsayı, DIRECTOR katar, ...
  - ▶ DIRECTOR niteliğine "Usual Suspects" değerini vermek veri tipi açısından doğru, yüklem açısından yanlış
  - ▶ YEAR ve VOTES değerleri birer tamsayı ancak bunları karşılaştırmak anlamlı değil

23 / 101

## Tutorial D Veri Tipleri

- ▶ INTEGER
- ▶ RATIONAL
- ▶ BOOL
- ▶ CHAR

24 / 101

## Tip Tanımlama

### Komut

```
TYPE type_name POSSREP {  
    field_name field_type  
    [, ...]  
};
```

### Tip Silme

```
DROP TYPE type_name;
```

25 / 101

## Tip Tanımlama Örnekleri

### Örnek

```
TYPE MOVIE# POSSREP { VALUE INTEGER };
```

```
TYPE YEAR POSSREP { VALUE INTEGER };
```

```
TYPE SCORE POSSREP { VALUE RATIONAL };
```

26 / 101

## Tip İşlemleri

- ▶ türetilen tip için değer üretme:  
type\_name(base\_value [, ...])!
- ▶ bir alanın değerini alma: THE\_ işlemleri  
THE\_field\_name(variable\_name)
- ▶ tip dönüşümü: CAST\_AS\_ işlemleri  
CAST\_AS\_target\_type(value)
- ▶ bir niteliğin adını değiştirmek:  
RENAME (attribute\_name AS new\_name)

27 / 101

## Tip İşlemleri Örnekleri

### Örnek

- ▶ bir SCORE değeri üretme:  
SCORE(8.7)
- ▶ bir SCORE değişkeninin VALUE alanını alma:  
THE\_VALUE(SCORE)
- ▶ bir tamsayı VOTES değerini bir RATIONAL değere dönüştürme:  
CAST\_AS\_RATIONAL(VOTES)
- ▶ MOVIE# niteliğinin adını değiştirmek:  
RENAME (MOVIE# AS MOVIE\_NO)

28 / 101

## Değer Kısıtlamaları

### Komut

```
TYPE type_name POSSREP {  
    field_name field_type  
    [, ...]  
    CONSTRAINT condition  
};
```

29 / 101

## Değer Kısıtlaması Örneği

### Example

- ▶ SCORE değerleri 1.0 ile 10.0 arasında olmalı

```
TYPE SCORE POSSREP {  
    VALUE RATIONAL  
    CONSTRAINT  
        (VALUE >= 1.0) AND (VALUE <= 10.0)  
};
```

30 / 101

## Bağıntı Tanımlama

### Komut

```
RELATION
{ attribute_name attribute_type
  [, ...] }
KEY { attribute_name [, ...] }
```

31 / 101

## Bağıntı Tanımlama Örneği

### Örnek

```
RELATION
{ MOVIE# MOVIE#,
  TITLE CHAR,
  YEAR YEAR,
  DIRECTOR CHAR,
  SCORE SCORE,
  VOTES INTEGER }
KEY { MOVIE# }
```

32 / 101

## Taban Bağıntı Değişkeni Yaratma

### Komut

```
VAR relvar_name BASE RELATION
{ ... }
KEY { ... };
```

### Değişken Silme

```
DROP VAR relvar_name;
```

33 / 101

## Taban Bağıntı Değişkeni Yaratma Örneği

### Örnek

```
VAR MOVIE BASE RELATION
{ MOVIE# MOVIE#,
  TITLE CHAR,
  YEAR YEAR,
  DIRECTOR CHAR,
  SCORE SCORE,
  VOTES INTEGER }
KEY { MOVIE# };
```

34 / 101

## Çoklu ve Bağıntı Üretme

### Çoklu Üretme

```
TUPLE {
  attribute_name attribute_value
  [, ...]
}
```

### Bağıntı Üretme

```
RELATION {
  TUPLE { ... }
  [, ...]
}
```

35 / 101

## Bağıntı Değişkenine Atama

- bağıntıların bağıntı değişkenlerine atanması:  
relvar\_name := RELATION { ... };

36 / 101

## Bağıntı Değişkenine Atama Örneği

### Örnek

```
MOVIE := RELATION {  
  TUPLE { MOVIE# MOVIE#(6),  
    TITLE "Usual Suspects",  
    YEAR YEAR(1995), DIRECTOR "Bryan Singer",  
    SCORE SCORE(8.7), VOTES 35027 },  
  TUPLE { MOVIE# MOVIE#(70),  
    TITLE "Being John Malkovich",  
    YEAR YEAR(1999), DIRECTOR "Spike Jonze",  
    SCORE SCORE(8.3), VOTES 13809 }  
};
```

37 / 101

## Çoklu Ekleme

### Komut

```
INSERT relvar_name RELATION {  
  TUPLE { ... }  
  [, ...]  
};
```

38 / 101

## Çoklu Ekleme Örneği

### Örnek

```
INSERT MOVIE RELATION {  
  TUPLE { MOVIE# MOVIE#(6),  
    TITLE "Suspiria",  
    YEAR YEAR(1977),  
    DIRECTOR "Dario Argento",  
    SCORE SCORE(7.1), VOTES 1004 }  
};
```

39 / 101

## Çoklu Silme

### Komut

```
DELETE relvar_name  
[ WHERE condition ];
```

- koşul belirtilmezse bütün çoklular silinir

40 / 101

## Çoklu Silme Örneği

### Örnek

- puanı 3.0'dan düşük, oy sayısı 4'den fazla olan filmleri sil

```
DELETE MOVIE  
WHERE ((SCORE < SCORE(3.0))  
  AND (VOTES > 4));
```

41 / 101

## Çoklu Güncelleme

### Komut

```
UPDATE relvar_name  
[ WHERE condition ]  
( attribute_name := attribute_value  
  [, ...] );
```

- koşul belirtilmezse bütün çoklular güncellenir

42 / 101

## Çoklu Güncelleme Örneği

### Örnek

- "Suspiria" filmi için yeni verilen bir oyu (9) işle

```
UPDATE MOVIE
WHERE (TITLE = "Suspiria") (
  SCORE := SCORE(
    (THE_VALUE(SCORE)
     * CAST_AS_RATIONAL(VOTES)
     + CAST_AS_RATIONAL(9))
    / CAST_AS_RATIONAL(VOTES + 1)
  ),
  VOTES := VOTES + 1
);
```

43 / 101

## Tek Boyutlu Değerler

- değerlerin tek boyutlu olması kuralına uyabilmek için çokluların yinelenmeleri gerekebilir

### Örnek (oyuncular nasıl tutulacak?)

MOVIE#	TITLE	...	ACTORS
6	Usual Suspects	...	Gabriel Byrne
...	...	...	...
70	Being John Malkovich	...	<del>Cameron Diaz, John Malkovich</del>
...	...	...	...

MOVIE#	TITLE	...	ACTOR
6	Usual Suspects	...	Gabriel Byrne
...	...	...	...
70	Being John Malkovich	...	Cameron Diaz
70	Being John Malkovich	...	John Malkovich
...	...	...	...

44 / 101

## Tek Boyutluluk Örneği

### Örnek (filmler ve oyuncular)

MOVIE#	TITLE	...
6	Usual Suspects	...
1512	Suspiria	...
70	Being John Malkovich	...
...	...	...

ACTOR#	NAME
308	Gabriel Byrne
282	Cameron Diaz
503	John Malkovich
...	...

MOVIE#	ACTOR#	ORD
6	308	2
70	282	2
70	503	14
...	...	...

45 / 101

## Tek Boyutluluk Örneği

### Örnek (yönetmenler nasıl tutulacak?)

MOVIE#	TITLE	...	DIRECTOR#
6	Usual Suspects	...	639
1512	Suspiria	...	2259
70	Being John Malkovich	...	1485
...	...	...	...

PERSON#	NAME
308	Gabriel Byrne
1485	Spike Jonze
639	Bryan Singer
282	Cameron Diaz
2259	Dario Argento
503	John Malkovich
...	...

MOVIE#	ACTOR#	ORD
6	308	2
70	282	2
70	503	14
...	...	...

46 / 101

## Dış Anahtarlar

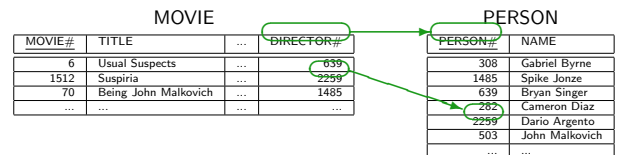
### Tanım

**dış anahtar:** bir bağıntının bir niteliğinin başka bir bağıntının anahtar adayı olması

47 / 101

## Dış Anahtar Örnekleri

### Örnek (MOVIE bağıntı değişkeninin DIRECTOR# niteliği)



48 / 101



## Dış Anahtar Örnekleri

Örnek (film veri tabanındaki dış anahtarlar)



49 / 101

## Başvuru Bütünlüğü

### Tanım

**başvuru bütünlüğü:**

dış anahtar niteliğinin aldığı bütün değerler  
başvurulan bağıntının ilgili niteliğinde bulunmalı

- ▶ başvuru bütünlüğünü bozacak işlemler engellenmeli

50 / 101

## Başvuru Bütünlüğü Örneği

Örnek

MOVIE				PERSON	
MOVIE#	TITLE	...	DIRECTOR#	PERSON#	NAME
...	...	...	...	...	...
1512	Suspiria	...	2259	2259	Dario Argento
...	...	...	...	...	...

- ▶ (2259, Dario Argento) çoklusunu sil
- ▶ (2259, Dario Argento) çoklusunu  
(2871, Dario Argento) olarak değiştir

51 / 101

## Dış Anahtar Tanımlama

### Komut

```
CONSTRAINT constraint_name
referencing_relvar_name
{ attribute_name }
<= referenced_relvar_name
{ attribute_name };
```

- ▶ iki bağıntıdaki nitelik isimlerinin aynı olması gerek
  - ▶ değilse nitelik isimleri değiştirilmeli

52 / 101

## Dış Anahtar Tanımlama Örnekleri

Örnek (CASTING bağıntı değişkeninin MOVIE# niteliği)

```
CONSTRAINT CASTING_FKEY_MOVIE
CASTING { MOVIE# }
<= MOVIE { MOVIE# };
```

53 / 101

## Dış Anahtar Tanımlama Örnekleri

Örnek (MOVIE bağıntı değişkeninin DIRECTOR# niteliği)

```
CONSTRAINT MOVIE_FKEY_DIRECTOR
MOVIE { DIRECTOR# }
RENAME (DIRECTOR# AS PERSON#)
<= PERSON { PERSON# };
```

54 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (tipler)

```
TYPE MOVIE# POSSREP { VALUE INTEGER };
TYPE YEAR POSSREP { VALUE INTEGER };
TYPE SCORE POSSREP { VALUE RATIONAL
  CONSTRAINT (VALUE >= 1.0)
    AND (VALUE <= 10.0) };
TYPE PERSON# POSSREP { VALUE INTEGER };
```

55 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (MOVIE bağıntı değişkeni)

```
VAR MOVIE BASE RELATION
{ MOVIE# MOVIE#, TITLE CHAR, YEAR YEAR,
  SCORE SCORE, VOTES INTEGER,
  DIRECTOR# PERSON# }
KEY { MOVIE# };
```

56 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (PERSON bağıntı değişkeni)

```
VAR PERSON BASE RELATION
{ PERSON# PERSON#, NAME CHAR }
KEY { PERSON# };
```

57 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (CASTING bağıntı değişkeni)

```
VAR CASTING BASE RELATION
{ MOVIE# MOVIE#, ACTOR# PERSON#,
  ORD INTEGER }
KEY { MOVIE#, ACTOR# };
```

58 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (MOVIE bağıntı değişkeninin dış anahtarları)

```
CONSTRAINT MOVIE_FKEY_DIRECTOR
MOVIE { DIRECTOR# }
  RENAME (DIRECTOR# AS PERSON#)
<= PERSON { PERSON# };
```

59 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (CASTING bağıntı değişkeninin dış anahtarları)

```
CONSTRAINT CASTING_FKEY_MOVIE
CASTING { MOVIE# } <= MOVIE { MOVIE# };

CONSTRAINT CASTING_FKEY_ACTOR
CASTING { ACTOR# }
  RENAME (ACTOR# AS PERSON#)
<= PERSON { PERSON# };
```

60 / 101

## Veri Tipleri

- ▶ INTEGER
  - ▶ SMALLINT
- ▶ NUMERIC (precision, scale)
  - ▶ precision: toplam hane sayısı
  - ▶ scale: noktadan sonraki hane sayısı
  - ▶ eşanlamlısı: DECIMAL (precision, scale)
- ▶ FLOAT (p)
  - ▶ p: kabul edilebilir en düşük duyarlık
- ▶ BOOLEAN

61 / 101

## Katar Veri Tipleri

- ▶ CHARACTER [VARYING] (n)
  - ▶ CHARACTER (n) tipinde, katar n simgeden kısaysa sona boşluk eklenir
- ▶ kısaltmalar:
  - ▶ CHARACTER (n) yerine CHAR (n)
  - ▶ CHARACTER VARYING (n) yerine VARCHAR (n)

62 / 101

## Tarih - Zaman Veri Tipleri

- ▶ DATE
  - ▶ değer örneği: 2005-09-26
- ▶ TIME
  - ▶ değer örneği: 11:59:22.078717
- ▶ TIMESTAMP
  - ▶ değer örneği: 2005-09-26 11:59:22.078717
- ▶ INTERVAL
  - ▶ değer örneği: 3 days

63 / 101

## Büyük Nesne Veri Tipleri

- ▶ rasgele uzunluklu nesneler
- ▶ sorgulamada kullanılamaz
- ▶ metin: CHARACTER LARGE OBJECT (n)
  - ▶ CLOB
- ▶ ikili: BINARY LARGE OBJECT (n)
  - ▶ BLOB
  - ▶ resim, ses v.b.

64 / 101

## Tanım Kümesi Yaratma

### Komut

```
CREATE DOMAIN domain_name [ AS ] base_type
[ DEFAULT default_value ]
[ { CHECK ( condition ) } [, ...] ]
```

### Tanım Kümesi Silme

```
DROP DOMAIN domain_name [, ...]
```

65 / 101

## Tanım Kümesi Örneği

Örnek (geçerli SCORE değerleri için bir tanım kümesi)

```
CREATE DOMAIN SCORES AS FLOAT
CHECK ((VALUE >= 1.0) AND (VALUE <= 10.0))
```

66 / 101

## Tablo Yaratma

### Komut

```
CREATE TABLE table_name (  
    { column_name data_type }  
    [, ... ]  
)
```

### Tablo Silme

```
DROP TABLE table_name [, ... ]
```

67 / 101

## Boş ve Varsayılan Değerler

### Komut

```
CREATE TABLE table_name (  
    { column_name data_type  
        [ NULL | NOT NULL ]  
        [ DEFAULT default_value ] }  
    [, ... ]  
)
```

- ▶ NULL: niteliğin boş bırakılmasına izin var (varsayılan)
- ▶ NOT NULL: niteliğin boş bırakılmasına izin yok

68 / 101

## Birincil Anahtar Tanımlama

### Komut

```
CREATE TABLE table_name (  
    { column_name data_type  
        [ NULL | NOT NULL ]  
        [ DEFAULT default_value ] }  
    [, ... ]  
    [ PRIMARY KEY ( column_name [, ...] ) ]  
)
```

- ▶ birincil anahtar tek bir sütundan oluşuyorsa, doğrudan sütun tanımında belirtilebilir:  
  
column\_name data\_type PRIMARY KEY

69 / 101

## Tablo Yaratma Örneği

### Örnek

```
CREATE TABLE MOVIE (  
    ID INTEGER,  
    TITLE VARCHAR(80) NOT NULL,  
    YR NUMERIC(4),  
    DIRECTOR VARCHAR(40),  
    SCORE FLOAT,  
    VOTES INTEGER DEFAULT 0,  
    PRIMARY KEY (ID)  
)
```

70 / 101

## Tablo Yaratma Örneği

### Örnek

```
CREATE TABLE MOVIE (  
    ID INTEGER PRIMARY KEY,  
    ...  
    VOTES INTEGER DEFAULT 0  
)
```

71 / 101

## Kendiliğinden Artırılan Değerler

- ▶ ürüne özel tanımlar
  - ▶ PostgreSQL: SERIAL veri tipi
  - ▶ MySQL: AUTO\_INCREMENT özelliği

72 / 101

## Kendiliğinden Artırılan Değer Örnekleri

### Örnek (PostgreSQL)

```
CREATE TABLE MOVIE (  
  ID SERIAL PRIMARY KEY,  
  ...  
)
```

### Örnek (MySQL)

```
CREATE TABLE MOVIE (  
  ID INTEGER PRIMARY KEY AUTO_INCREMENT,  
  ...  
)
```

73 / 101

## Değer Kısıtlamaları

### Komut

```
CREATE TABLE table_name (  
  ...  
  [ { CHECK ( condition ) }  
  [, ...] ]  
  ...  
)
```

74 / 101

## Değer Kısıtlaması Örneği

### Örnek

- SCORE değerleri 1.0 ile 10.0 arasında olmalı

```
CREATE TABLE MOVIE (  
  ID INTEGER PRIMARY KEY,  
  ...,  
  SCORE FLOAT,  
  VOTES INTEGER DEFAULT 0,  
  CHECK ((SCORE >= 1.0) AND (SCORE <= 10.0))  
)
```

75 / 101

## Eşsizlik Tanımı

### Komut

```
CREATE TABLE table_name (  
  ...  
  [ { UNIQUE ( column_name [, ...] ) }  
  [, ...] ]  
  ...  
)
```

- eşsizlik kısıtlaması tek bir sütundan oluşuyorsa, doğrudan sütun tanımında belirtilebilir:

```
column_name data_type UNIQUE
```

76 / 101

## Eşsizlik Tanımı Örneği

### Örnek

- {TITLE} ve {DIRECTOR, YR} eşsiz

```
CREATE TABLE MOVIE (  
  ID INTEGER PRIMARY KEY,  
  TITLE VARCHAR(80) UNIQUE NOT NULL,  
  YR NUMERIC(4),  
  DIRECTOR VARCHAR(40),  
  SCORE FLOAT,  
  VOTES INTEGER DEFAULT 0,  
  UNIQUE (DIRECTOR, YR)  
)
```

77 / 101

## Tablo Adı Değiştirme

### Komut

```
ALTER TABLE table_name  
  RENAME TO new_name
```

### Örnek

```
ALTER TABLE MOVIE  
  RENAME TO FILM
```

78 / 101

## Sütun Ekleme

Komut

```
ALTER TABLE table_name
  ADD [ COLUMN ] column_name data_type
      [ NULL | NOT NULL ]
      [ DEFAULT default_value ]
```

Örnek

```
ALTER TABLE MOVIE
  ADD COLUMN RUNTIME INTEGER
```

79 / 101

## Sütun Silme

Komut

```
ALTER TABLE table_name
  DROP [ COLUMN ] column_name
```

Örnek

```
ALTER TABLE MOVIE
  DROP COLUMN RUNTIME
```

80 / 101

## Sütun Adı Değiştirme

Komut

```
ALTER TABLE table_name
  RENAME [ COLUMN ] column_name TO new_name
```

Örnek

```
ALTER TABLE MOVIE
  RENAME COLUMN TITLE TO NAME
```

81 / 101

## Sütun Varsayılan Değeri

Sütun Varsayılan Değeri Değiştirme

```
ALTER TABLE table_name
  ALTER [ COLUMN ] column_name
  SET DEFAULT default_value
```

Sütun Varsayılan Değeri Silme

```
ALTER TABLE table_name
  ALTER [ COLUMN ] column_name
  DROP DEFAULT
```

82 / 101

## Kısıtlama Ekleme

Komut

```
ALTER TABLE table_name
  ADD [ CONSTRAINT constraint_name ]
  constraint_definition
```

- var olan çoklular ne olacak?

Kısıtlama Silme

```
ALTER TABLE table_name
  DROP [ CONSTRAINT ] constraint_name
```

83 / 101

## Kısıtlama Ekleme Örneği

Örnek

- YR değerleri 1888'den küçük olmasın

```
ALTER TABLE MOVIE
  ADD CONSTRAINT MINIMUM_YEAR
  CHECK (YR >= 1888)
```

- minimum yıl kısıtlamasını kaldır

```
ALTER TABLE MOVIE
  DROP CONSTRAINT MINIMUM_YEAR
```

84 / 101

## Satır Ekleme

### Komut

```
INSERT INTO table_name
[ ( column_name [, ...] ) ]
VALUES ( column_value [, ...] )
```

- değer sırası sütun adı sırasına uymalıdır
- sütun adları belirtilmezse sütun değerleri tablo yaratılırken verilen sırayla yazılmalıdır
- belirtilmeyen sütunlara varsayılan değerleri atanır
- otomatik üretilecek sütunları belirtmemek gerekir

85 / 101

## Satır Ekleme Örnekleri

### Örnek

```
INSERT INTO MOVIE VALUES (
6,
'Usual Suspects',
1995,
'Bryan Singer',
8.7,
35027
)
```

86 / 101

## Satır Ekleme Örnekleri

### Örnek

```
INSERT INTO MOVIE (YR, TITLE) VALUES (
1995,
'Usual Suspects'
)
```

- ID değeri otomatik üretilir

87 / 101

## Satır Silme

### Komut

```
DELETE FROM table_name
[ WHERE condition ]
```

88 / 101

## Satır Silme Örneği

### Örnek

- puanı 3.0'dan düşük, oy sayısı 4'den fazla olan filmleri sil

```
DELETE FROM MOVIE
WHERE ((SCORE < 3.0) AND (VOTES > 4))
```

89 / 101

## Satır Güncelleme

### Komut

```
UPDATE table_name
SET { column_name = column_value } [, ...]
[ WHERE condition ]
```

90 / 101

## Satır Güncelleme Örneği

### Örnek

- "Suspiria" filmi için yeni verilen bir oyu (9) işle

```
UPDATE MOVIE
SET SCORE = (SCORE * VOTES + 9)
           / (VOTES + 1),
    VOTES = VOTES + 1
WHERE (TITLE = 'Suspiria')
```

91 / 101

## Dış Anahtar Tanımlama

### Komut

```
CREATE TABLE table_name (
    ...
    [ { FOREIGN KEY ( column_name [, ...] )
      REFERENCES table_name
        [ ( column_name [, ...] ) ]
      [ ON DELETE option ]
      [ ON UPDATE option ] } [, ...] ]
    ...
)
```

92 / 101

## Başvuru Bütünlüğü Seçenekleri

- işleme izin verme: RESTRICT, NO\_ACTION
- işlemi etkilenen çoklulara yansıt: CASCADE
- boş değer ata: SET NULL
  - boş değer olmasına izin varsa
- varsayılan değer ata: SET DEFAULT

93 / 101

## Bütünlük Kısıtlamaları Örnekleri

### Örnek (silerken kısıtla)

- PERSON bağlantısından silinmek istenen yönetmenin ID niteliği değeri MOVIE bağlantısının DIRECTORID niteliği değerleri arasında geçiyorsa silmeye izin verme

### Örnek (güncellerken yansıt)

- bir yönetmenin ID niteliği değeri PERSON bağlantısında güncellendiğinde MOVIE bağlantısının DIRECTORID niteliği değerlerinde geçtiği çoklularda da güncelle

94 / 101

## Bütünlük Kısıtlamaları Örnekleri

### Örnek (silerken yansıt)

- kişiyi PERSON bağlantısından sil
- CASTING bağlantısının ACTORID niteliğinde silinen kişinin ID niteliği değerinin geçtiği bütün çokluları sil
- kişinin ID niteliği değerinin MOVIE bağlantısının DIRECTORID niteliğinde geçtiği bütün çokluları MOVIE bağlantısından sil
- önceki adımda silinen her bir MOVIE çoklusu için CASTING bağlantısının MOVIEID niteliğinde silinen filmin ID niteliği değerinin geçtiği bütün çokluları sil

95 / 101

## Dış Anahtar Tanımlama Örneği

### Örnek (MOVIE tablosunda DIRECTORID dış anahtarı)

```
CREATE TABLE MOVIE (
    ID INTEGER PRIMARY KEY,
    ...
    DIRECTORID INTEGER,
    FOREIGN KEY DIRECTORID
    REFERENCES PERSON (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
)
```

96 / 101



## Dış Anahtar Tanımlama

- ▶ dış anahtar, başvuru tablosunda birincil anahtarla eşleşiyorsa REFERENCES kısmında belirtilmesi zorunlu değildir
- ▶ dış anahtar tek bir sütundan oluşuyorsa, sütun tanımında belirtilebilir:

```
column_name data_type  
REFERENCES table_name [ ( column_name ) ]
```

97 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (MOVIE tablosunun yaratılması)

```
CREATE TABLE MOVIE (  
  ID INTEGER PRIMARY KEY,  
  TITLE VARCHAR(80) NOT NULL,  
  YR NUMERIC(4),  
  SCORE FLOAT,  
  VOTES INTEGER DEFAULT 0,  
  DIRECTORID INTEGER REFERENCES PERSON  
)
```

98 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (PERSON tablosunun yaratılması)

```
CREATE TABLE PERSON (  
  ID INTEGER PRIMARY KEY,  
  NAME VARCHAR(40) UNIQUE NOT NULL  
)
```

99 / 101

## Örnek Veri Tabanının Yaratılması

### Örnek (CASTING tablosunun yaratılması)

```
CREATE TABLE CASTING (  
  MOVIEID INTEGER REFERENCES MOVIE,  
  ACTORID INTEGER REFERENCES PERSON,  
  ORD INTEGER,  
  PRIMARY KEY (MOVIEID, ACTORID)  
)
```

100 / 101

## Kaynaklar

### Okunacak: Date

- ▶ Chapter 3: An Introduction to Relational Databases
  - ▶ 3.2. [An Informal Look at the Relational Model](#)
  - ▶ 3.3. [Relations and Relvars](#)
- ▶ Chapter 6: [Relations](#)
- ▶ Chapter 9: Integrity
  - ▶ 9.10. [Keys](#)
  - ▶ 9.12. [SQL Facilities](#)

101 / 101