

# Veri Tabanı Sistemleri

## Bağıntı Modeli

H. Turgut Uyar Şule Öğüdücü

2002-2016

1 / 104

## License



© 2002-2016 T. Uyar, Ş. Öğüdücü

You are free to:

- Share – copy and redistribute the material in any medium or format
- Adapt – remix, transform, and build upon the material

Under the following terms:

- Attribution – You must give appropriate credit, provide a link to the license, and indicate if changes were made.
- NonCommercial – You may not use the material for commercial purposes.
- ShareAlike – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

For more information:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Read the full license:

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

2 / 104

## Konular

### Bağıntı Modeli

Giriş  
Anahtarlar  
Başvuru Bütünlüğü  
TutorialD

### SQL

Veri Tipleri  
Veri Tanımlama  
Veriyi Değiştirme  
Başvuru Bütünlüğü

3 / 104

## Bağıntı Modeli

- Dr. E. F. Codd, 1970
- 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
- bağıntı → tablo, çoklu → satır, nitelik → sütun

4 / 104

## Bağıntı Örneği

### MOVIE

TITLE	YEAR	DIRECTOR	SCORE	VOTES
The 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

5 / 104

## Bağıntı Yapısı

- ▶ bağıntı başlığı: bağıntının nitelikleri kümesi
- ▶ veri tanımlama dili komutlarından etkilenir
- ▶ bağıntı gövdesi: bağıntıdaki çoklular kümesi
- ▶ veri işleme dili komutlarından etkilenir

6 / 104

## Bağıntı Yükleme

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

### örnek

- ▶ "TITLE başlıklı film YEAR yılında çekilmiştir. DIRECTOR tarafından yönetilmiştir. Verilen VOTES oyun ortalaması SCORE'dur."

7 / 104

## Çoklu Değerleri

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

### örnek: MOVIE bağıntısı

- ▶ (Suspiria, 1977, Dario Argento, 1004, 7.1) çoklusu doğrudur
- ▶ (Suspiria, 1978, Dario Argento, 1004, 7.1) çoklusu yanlıştır

8 / 104

## Çoklu Sırası

- çokluların sırası önemsizdir

### örnek

- şu iki bağıntı eşdeğerlidir:

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

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

9 / 104

## Nitelik Sırası

- niteliklerin sırası önemsizdir

### örnek

- şu iki bağıntı eşdeğerlidir:

TITLE	YEAR	...
The Usual Suspects	1995	...
Suspiria	1977	...

YEAR	TITLE	...
1995	The Usual Suspects	...
1977	Suspiria	...

10 / 104

## Eş Çoklular

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

### örnek

TITLE	YEAR	DIRECTOR	SCORE	VOTES
The 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
...	...	...	...	...

11 / 104

## Tanım Kümeleri

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

12 / 104

## Tanım Kümesi Örneği

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

13 / 104

## Nitelik Değerleri

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

örnek: birden fazla yönetmen

TITLE	...	DIRECTORS	...
...	...	...	...
The Matrix	...	<del>Andy Wachowski, Lana Wachowski</del>	...
...	...	...	...

14 / 104

## Boş Değer

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

örnek

- ▶ filmin yönetmeni bilinmiyor

örnek

- ▶ film için oy kullanılmamış, o yüzden SCORE yok

15 / 104

## 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

16 / 104

## Anahtarlar

- ▶  $B$  bağıntısının 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

17 / 104

## Anahtar Adayı Örneği

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

18 / 104

## Anahtar Eşdeğeri

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

19 / 104

## Anahtar Eşdeğeri Örneği

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

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

20 / 104

## Birincil Anahtar

- ▶ bir bağlantının birden fazla anahtar adayı varsa biri **birincil anahtar** seçilir
- ▶ diğerleri anahtar seçeneği olur
- ▶ birincil anahtarı oluşturan nitelikler altı çizili gösterilir
- ▶ birincil anahtarın parçası olan hiçbir niteliğin değeri hiçbir çokluda boş olamaz
- ▶ her bağlantının bir birincil anahtarı bulunması zorunludur

21 / 104

## Birincil Anahtar Örneği

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

22 / 104

## Tek Boyutluluk Örneği

- ▶ oyuncular nasıl tutulacak?

MOVIE

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

23 / 104

## Tek Boyutluluk Örneği

- ▶ Tek boyutluluk için çokluların tekrar edilmesi gerekir

MOVIE

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

24 / 104

## Tek Boyutluluk Örneği

MOVIE		
<u>MOVIE#</u>	TITLE	...
6	The Usual Suspects	...
1512	Suspiria	...
70	Being John Malkovich	...
...	...	...

ACTOR	
<u>ACTOR#</u>	NAME
308	Gabriel Byrne
282	Cameron Diaz
503	John Malkovich
...	...

CASTING		
<u>MOVIE#</u>	<u>ACTOR#</u>	ORD
6	308	2
70	282	2
70	503	14
...	...	...

25 / 104

## Tek Boyutluluk Örneği

MOVIE			
<u>MOVIE#</u>	TITLE	...	DIRECTOR#
6	The Usual Suspects	...	639
1512	Suspiria	...	2259
70	Being John Malkovich	...	1485
...	...	...	...

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

CASTING		
<u>MOVIE#</u>	<u>ACTOR#</u>	ORD
6	308	2
70	282	2
70	503	14
...	...	...

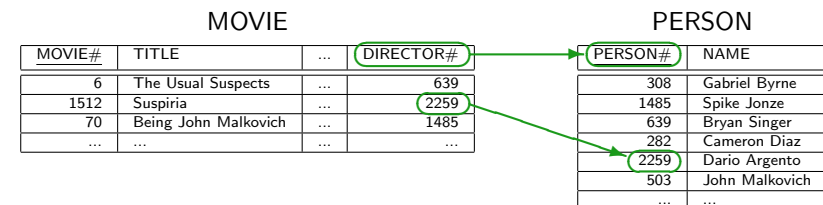
26 / 104

## Dış Anahtarlar

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

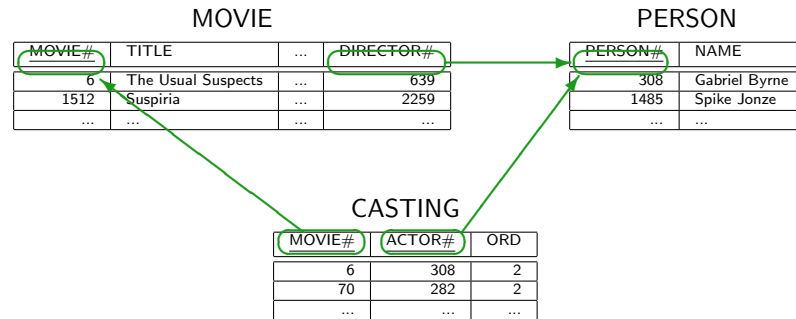
27 / 104

## Dış Anahtar Örnekleri



28 / 104

## Dış Anahtar Örnekleri



29 / 104

## Başvuru Bütünlüğü

- ▶ **başvuru bütünlüğü:**  
dış anahtar niteliğinin aldığı bütün değerler başvuru bağıntısının ilgili niteliğinde bulunmalı
- ▶ bir işlem isteği başvuru bütünlüğünü bozuyorsa:
- ▶ izin verme
- ▶ işlemi etkilenen çoklulara yansıt
- ▶ boş değer ata
- ▶ varsayılan değer ata

30 / 104

## Başvuru Bütünlüğü Örnekleri

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

31 / 104

## Tutorial D Veri Tipleri

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

32 / 104



## Tip Tanımlama

- yeni bir tip tanımlama:

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

- tip silme:

```
DROP TYPE type_name;
```

33 / 104

## Tip Tanımlama Örnekleri

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

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

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

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

34 / 104

## Tip İşlemleri

- türetilen tip için değer üretme:

```
type_name(base_value [, ...])
```

### example

- bir SCORE değeri üretme:

```
SCORE(8.7)
```

35 / 104

## Tip İşlemleri

- bir alanın değerini alma: **THE\_** işlemleri

```
THE_field_name(variable_name)
```

### örnek

- bir SCORE değişkeninin VALUE alanını alma:

```
THE_VALUE(SCORE)
```

36 / 104

## Tip İşlemleri

- tip dönüşümü: `CAST_AS_` işleçleri  
`CAST_AS_target_type(value)`

### örnek

- bir tamsayı VOTES değerini bir RATIONAL değere dönüştürme:  
`CAST_AS_RATIONAL(VOTES)`

37 / 104

## Bağıntı Tanımlama

- yeni bir bağıntı tanımlama:

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

38 / 104

## Bağıntı Tanımlama Örneği

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

39 / 104

## Bağıntı Değişkenleri

- yeni bir bağıntı değişkeni tanımlama

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

- bağıntı değişkeni silme:

```
DROP VAR relvar_name;
```

40 / 104

## Bağıntı Değişkeni Örnekleri

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

41 / 104

## Bağıntı Değişkeni Örnekleri

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

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

42 / 104

## Çoklu Üretme

- ▶ çoklu üretme:

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

43 / 104

## Çoklu Üretme Örnekleri

```
TUPLE
{ MOVIE# MOVIE#(6),
  TITLE "The Usual Suspects",
  YEAR YEAR(1995),
  DIRECTOR# PERSON#(639),
  SCORE SCORE(8.7),
  VOTES 35027 }
```

```
TUPLE
{ PERSON# PERSON#(639),
  NAME "Bryan Singer" }
```

44 / 104

## Bağıntı Üretme

- bağıntı üretme:

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

- bağıntıların bağıntı değişkenlerine atanması:

```
relvar_name := RELATION { ... };
```

45 / 104

## Bağıntı Değişkenine Atama Örneği

```
MOVIE := RELATION
```

```
{ TUPLE
  { MOVIE# MOVIE#(6), TITLE "The Usual Suspects",
    YEAR YEAR(1995), DIRECTOR# PERSON#(639),
    SCORE SCORE(8.7), VOTES 35027 },
  TUPLE
  { MOVIE# MOVIE#(70), TITLE "Being John Malkovich",
    YEAR YEAR(1999), DIRECTOR# PERSON#(1485),
    SCORE SCORE(8.3), VOTES 13809 } };
```

46 / 104

## Çoklu Ekleme

- çokluları ekleme:

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

47 / 104

## Çoklu Ekleme Örneği

```
INSERT MOVIE RELATION
```

```
{ TUPLE
  { MOVIE# MOVIE#(1),
    TITLE "Star Wars",
    YEAR YEAR(1977),
    DIRECTOR# PERSON#(360),
    SCORE SCORE(8.8),
    VOTES 53567 } };
```

48 / 104

## Çoklu Silme

- ▶ çokluları silme:

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

- ▶ koşul belirtilmezse bütün çoklular silinir

49 / 104

## Çoklu Silme Örneği

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

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

50 / 104

## Çoklu Güncelleme

- ▶ çokluları güncelleme:

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

- ▶ koşul belirtilmezse bütün çoklular güncellenir

51 / 104

## Çoklu Güncelleme Örneği

- ▶ "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  
);
```

52 / 104

## Nitelik Adı Değiştirme

- bir niteliğin adını değiştirme:

```
RENAME { attribute_name AS new_name }
```

### örnek

- DIRECTOR# niteliğinin adını değiştirme:

```
RENAME { DIRECTOR# AS PERSON# }
```

53 / 104

## Dış Anahtar Tanımlama

- dış anahtar tanımlama:

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

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

54 / 104

## Dış Anahtar Örnekleri

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

55 / 104

## Dış Anahtar Tanımlama Örnekleri

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

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

56 / 104

## Veri Tipleri

- ▶ INTEGER
- ▶ NUMERIC (precision, scale)
  - ▶ precision: toplam hane sayısı
  - ▶ scale: noktadan sonraki hane sayısı
  - ▶ eşanlamlısı: DECIMAL (precision, scale)
- ▶ FLOAT
- ▶ BOOLEAN

57 / 104

## Katar Veri Tipleri

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

58 / 104

## 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

59 / 104

## Büyük Nesne Veri Tipleri

- ▶ rasgele uzunluklu nesneler
- ▶ ikili: BINARY LARGE OBJECT (n)
- ▶ BLOB
- ▶ metin: CHARACTER LARGE OBJECT (n)
- ▶ CLOB
- ▶ sorgulamada kullanılamaz

60 / 104

## Tanım Kümesi Yaratma

- tanım kümesi yaratma:

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

- tanım kümesi silme:

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

61 / 104

## Tanım Kümesi Örneği

- geçerli SCORE değerleri için bir tanım kümesi:

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

62 / 104

## Tablo Yaratma

- tablo yaratma:

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

- tablo silme:

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

63 / 104

## Tablo Yaratma Örneği

- tanım kümesi kullanarak:

```
CREATE TABLE MOVIE (  
  ID INTEGER,  
  TITLE VARCHAR(80),  
  YR NUMERIC(4),  
  DIRECTORID INTEGER,  
  SCORE FLOAT,  
  VOTES INTEGER  
)
```

```
CREATE TABLE MOVIE (  
  ID INTEGER,  
  TITLE VARCHAR(80),  
  YR NUMERIC(4),  
  DIRECTORID INTEGER,  
  SCORE SCORES,  
  VOTES INTEGER  
)
```

64 / 104



## Boş ve Varsayılan Değerler

- ▶ NULL değeri alabilen sütun ve varsayılan değer tanımlama:

```
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

65 / 104

## Tablo Yaratma Örneği

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

66 / 104

## Değer Kısıtlamaları

- ▶ değerler üzerinde kısıt tanımlama:

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

67 / 104

## Değer Kısıtlaması Örneği

- ▶ SCORE değerleri 1.0 ile 10.0 arasında

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

68 / 104

## Birincil Anahtarlar

- birincil anahtar tanımlama:

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

69 / 104

## Birincil Anahtar Tanımlama Örneği

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

70 / 104

## Birincil Anahtarlar

- birincil anahtar tek bir sütundan oluşuyorsa, doğrudan sütun tanımında belirtilebilir:

column\_name data\_type PRIMARY KEY

örnek

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

71 / 104

## Kendiliğinden Artırılan Değerler

- kendiliğinden artırılan değer tanımlamada standart yok
- PostgreSQL: SERIAL data type  
ID SERIAL PRIMARY KEY
- MySQL: AUTO\_INCREMENT property  
ID INTEGER PRIMARY KEY AUTO\_INCREMENT
- SQLite: AUTOINCREMENT property  
ID INTEGER PRIMARY KEY AUTOINCREMENT

72 / 104

## Eşsizlik

- ▶ eşsiz sütun tanımlama:

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

- ▶ boş değerler dikkate alınmaz

73 / 104

## Eşsizlik Tanımı Örneği

- ▶ (başlıklar) ve (yönetmen, yıl) eşsiz:

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

74 / 104

## Eşsizlik

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

column\_name data\_type **UNIQUE**

örnek: kişi isimleri eşsiz

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

75 / 104

## Dizinler

- ▶ dizin yaratma:

```
CREATE [ UNIQUE ] INDEX index_name  
    ON table_name (column_name [, ...])
```

- ▶ aramaları hızlandırır
- ▶ ekleme ve güncellemeleri yavaşlatır

örnek: yıl sütunda dizin yaratma

```
CREATE INDEX MOVIE_YEAR ON MOVIE (YR)
```

76 / 104

## Tablo Adı Değiştirme

- bir tablonun adını değiştirme:

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

örnek

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

77 / 104

## Sütun Ekleme

- varolan bir tabloya sütun ekleme:

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

örnek

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

78 / 104

## Sütun Silme

- bir tablodan sütun silme:

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

örnek

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

79 / 104

## Sütun Adı Değiştirme

- sütunun adını değiştirme:

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

örnek

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

80 / 104

## 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
```

81 / 104

## Kısıtlama Ekleme

- ▶ bir tabloya yeni bir kısıt ekleme:

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

- ▶ tablodan bir kısıt silme:

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

- ▶ kısıt eklendiğinde var olan çoklular ne olacak?

82 / 104

## Kısıtlama Ekleme Örneği

- ▶ YR değerleri 1887'den küçük olmasın

```
ALTER TABLE MOVIE  
  ADD CONSTRAINT MINIMUM_YEAR  
  CHECK (YR >= 1887)
```

- ▶ minimum yıl kısıtlamasını kaldır

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

83 / 104

## Satır Ekleme

- ▶ bir tabloya bir satır ekleme:

```
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 üretilcek sütunları belirtmemek gerekir

84 / 104

## Satır Ekleme Örnekleri

```
INSERT INTO MOVIE VALUES (  
    6,  
    'The Usual Suspects',  
    1995,  
    639,  
    8.7,  
    35027  
)
```

85 / 104

## Satır Ekleme Örnekleri

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

- ▶ ID değeri otomatik üretilir

86 / 104

## Satır Silme

- ▶ satır silme:

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

- ▶ koşul belirtilmezse bütün satırlar silinir

87 / 104

## Satır Silme Örneği

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

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

88 / 104

## Satır Güncelleme

- ▶ satır güncelleme:

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

- ▶ koşul belirtilmezse bütün satırlar güncellenir
- ▶ sütun sıralamasının önemi yoktur

89 / 104

## Satır Güncelleme Örneği

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

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

90 / 104

## Dış Anahtarlar

- ▶ dış anahtar tanımlama:

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

91 / 104

## Dış Anahtar Tanımlama Örneği

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

92 / 104

## Dış Anahtarlar

- ▶ dış anahtar tek bir sütundan oluşuyorsa, sütun tanımında belirtilebilir:

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

### örnek

```
CREATE TABLE MOVIE (  
  ID SERIAL PRIMARY KEY,  
  ...  
  DIRECTORID INTEGER REFERENCES PERSON (ID),  
  ...  
)
```

93 / 104

## Dış Anahtarlar

- ▶ dış anahtar, başvuru yapılan tabloda birincil anahtarla eşleşiyorsa REFERENCES kısmında belirtilmesi zorunlu değildir

### example

```
CREATE TABLE MOVIE (  
  ID SERIAL PRIMARY KEY,  
  ...  
  DIRECTORID INTEGER REFERENCES PERSON,  
  ...  
)
```

94 / 104

## Başvuru Bütünlüğü Seçenekleri

- ▶ yapılan işlem bütünlük kısıtlamasını bozarsa ne olacak?
- ▶ işleme izin verme: RESTRICT, NO\_ACTION
- ▶ işlemi etkilenen çoklulara yansıt: CASCADE
- ▶ boş değer ata: SET NULL
- ▶ varsayılan değer ata: SET DEFAULT

95 / 104

## Dış Anahtarlar

- ▶ bütünlük kısıtlamaları seçenekleri:

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

96 / 104



## Dış Anahtar Örneği

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

97 / 104

## Bütünlük Kısıtlamaları Örnekleri

MOVIE

ID	TITLE	...	DIRECTORID
6	The Usual Suspects	...	639
70	Being John Malkovich	...	1485
107	Batman & Robin	...	105

PERSON

ID	NAME
308	Gabriel Byrne
1485	Spike Jonze

- ▶ MOVIE.DIRECTORID: ON DELETE RESTRICT
- ▶ PERSON tablosundan "Spike Jonze" sil: izin verilmez
- ▶ PERSON tablosundan "Gabriel Byrne" sil: izin verilir

98 / 104

## Bütünlük Kısıtlamaları Örnekleri

MOVIE

ID	TITLE	DIRECTORID
6	The Usual Suspects	639
70	Being John Malkovich	1485
107	Batman & Robin	105
112	Three Kings	1070

PERSON

ID	NAME
308	Gabriel Byrne
1485	Spike Jonze

CASTING

MOVIEID	ACTORID	ORD
6	308	2
70	282	2
112	1485	4

- ▶ MOVIE.DIRECTORID: ON DELETE CASCADE
- ▶ CASTING.MOVIEID: ON DELETE CASCADE
- ▶ CASTING.ACTORID: ON DELETE CASCADE
- ▶ PERSON tablosundan "Spike Jonze" sil: hangi satırlar silinir?

99 / 104

## Bütünlük Kısıtlamaları Örnekleri

MOVIE

ID	TITLE	DIRECTORID
6	The Usual Suspects	639
70	Being John Malkovich	1485
107	Batman & Robin	105
112	Three Kings	1070

PERSON

ID	NAME
308	Gabriel Byrne
1485	Spike Jonze

CASTING

MOVIEID	ACTORID	ORD
6	308	2
70	282	2
112	1485	4

- ▶ MOVIE.DIRECTORID: ON DELETE RESTRICT
- ▶ CASTING.MOVIEID: ON DELETE CASCADE
- ▶ CASTING.ACTORID: ON DELETE CASCADE
- ▶ PERSON tablosundan "Spike Jonze" sil: hangi satırlar silinir?

100 / 104

## Örnek Veri Tabanı

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

101 / 104

## Örnek Veri Tabanı

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

102 / 104

## Örnek Veri Tabanı

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

103 / 104

## 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

104 / 104