



Veri Tabanı Yönetim Sistemleri

Bölüm 2 - İlişkisel (Relational) Model

Dr. Öğr. Üyesi Elvan DUMAN

Bölüm 2

İlişkisel (Relational) Model

Bölüm 2 Amaçlar



- ▶ **İlişkisel Modelin Temel Kavramları**
- ▶ **İlişkisel Modeldeki İlişkiler**
 - Matematiksel küme teorisi ve ilişkisel cebir ile bağlantısı vardır.
- ▶ **Veri tabanı ilişkilerinin Özellikleri**
 - Birincil anahtar ve diğer anahtarlar
- ▶ **Bütünlük Kuralları**
- ▶ **Görünümler (Views)**

İlişkisel Modelin Temel Kavramları

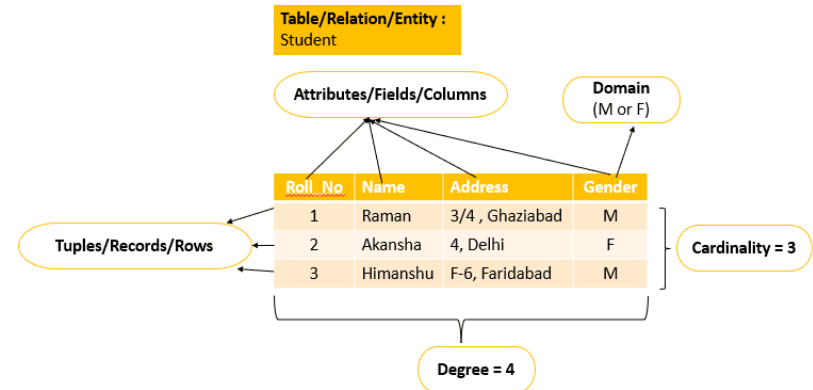


► İlişki (Relation):

- Bir tablo, satırlar (tuple) ve sütunlar (attribute) içerir.
- Sadece veritabanının mantıksal yapısını temsil eder, fiziksel yapıyı değil.

► Nitelik (Attribute):

- Bir tablodaki sütunları ifade eder.



► Alan (Domain):

- Bir veya daha fazla niteliğin alabileceği geçerli değerler kümesidir.

İlişkisel Modelin Temel Kavramları

► Çoklu (Tuple):

- Bir tablonun (ilişkinin) bir satırıdır.

► Derece (Degree):

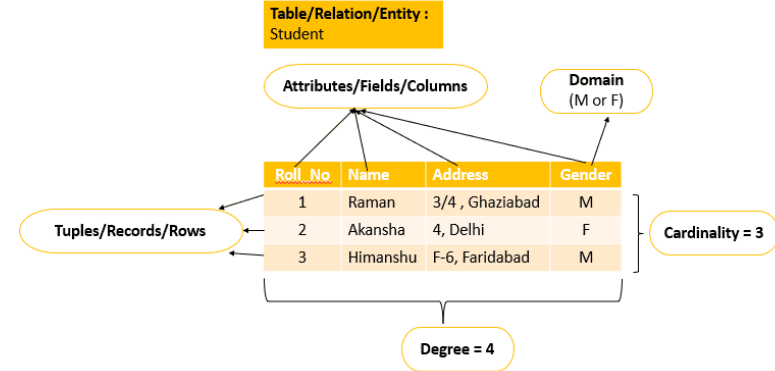
- Bir ilişkide bulunan niteliklerin (sütunların) sayısıdır.
- Örneğin, 5 sütunlu bir tablo için derece = 5.

► Eleman Sayısı (Cardinality):

- Bir ilişkideki satırların (kayıtların) sayısıdır.
- Örneğin, 100 satırlı bir tablo için eleman sayısı = 100.

► İlişkisel Veritabanı:

- Normalize edilmiş ilişkilerden oluşan bir koleksiyondur.
- Farklı tablo adları ile tanımlanır.



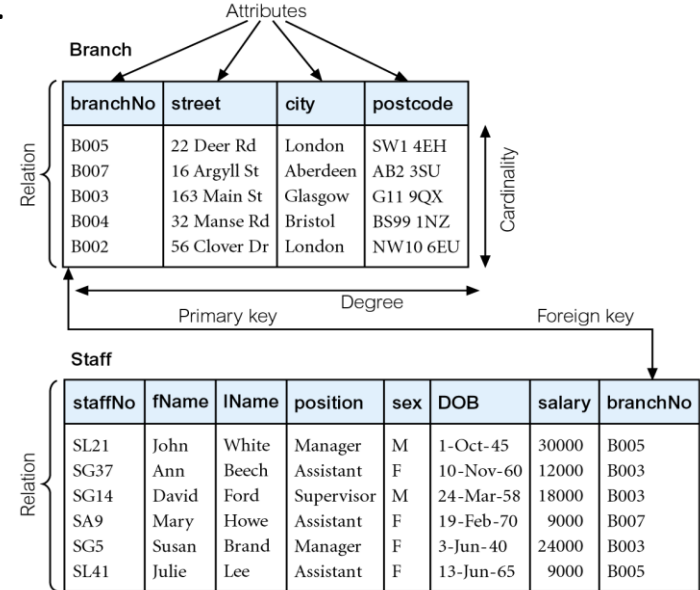
Şube ve Personel İlişkisi (Tablo) Örneği

► Branch (Şube) Tablosu

- **Nitelikler (Attributes):** branchNo, street, city, postcode.
- **Birincil Anahtar (Primary Key):** branchNo, (her şubeyi benzersiz olarak tanımlar.)
- **Derece (Degree):** 4 (çünkü 4 sütun var).
- **Eleman Sayısı (Cardinality):** 5 (çünkü 5 satır var).

► Staff (Personel) Tablosu:

- **Nitelikler:** staffNo, fName, lName, position ...
- **Birincil Anahtar (Primary Key):** staffNo.
- **Yabancı Anahtar (Foreign Key):** branchNo, (personelin hangi şubeye bağlı olduğunu gösterir)
- **Derece:** 8 (8 sütun)
- **Eleman Sayısı :** 6 (6 satır).



Nitelik Alanı (Domain) Kavramı ve Örnekleri



- ▶ Alan (Domain) veri bütünlüğünü sağlar.
- ▶ Veri türü, uzunluk ve kapsam (range) gibi kısıtlamalar içerir.

Nitelik (Attribute)	Alan Adı (Domain Name)	Anlamı	Alan Tanımı (Domain Definition)
branchNo	BranchNumbers	Şube numaraları kümesi	4 karakter, B001–B999 aralığı
street	StreetNames	Cadde adları kümesi	Maksimum 25 karakter
city	CityNames	Şehir adları kümesi	Maksimum 15 karakter
postcode	Postcodes	Posta kodları kümesi	Maksimum 8 karakter
sex	Sex	Kişinin cinsiyeti	1 karakter, M veya F
DOB	DatesOfBirth	Doğum tarihleri kümesi	dd-mmm-yy formatında tarih
salary	Salaries	Personel maaşlarının değerleri	7 basamaklı, 6000.00–40000.00 aralığında parasal değer

İlişkisel Model için Alternatif Terimler



- ▶ Farklı veri tabanı sistemlerinde bu terimler değişiklik gösterebilir.
- ▶ Alternatif 1, genellikle ilişkisel veritabanı terminolojisinde yaygın olarak kullanılır.
- ▶ Alternatif 2, daha eski dosya tabanlı sistemlerde ve bazı veri işleme yaklaşımlarında karşılaşılabılır.

Resmi Terim	Alternatif 1	Alternatif 2
İlişki (Relation)	Tablo (Table)	Dosya (File)
Çoklu (Tuple)	Satır (Row)	Kayıt (Record)
Nitelik (Attribute)	Sütun (Column)	Alan (Field)

İlişkinin Matematiksel Tanımı



- ▶ **İlişki (Relation)**, $R(A_1, A_2, \dots, A_n)$ şeklinde gösterilir ve $A_1, A_2, \dots, A_n$ tabloda bulunan nitelikleri (sütunları) ifade eder.
- ▶ **Matematiksel olarak**, bir ilişki, $A_1 \times A_2 \times \dots \times A_n$ kümesinin bir alt kümesidir.
- ▶ Bir ilişkinin derecesi (degree), içerdiği niteliklerin sayısıdır.
- ▶ Bir Öğrenciler tablosu olsun: Öğrenci (ÖğrenciNo, Ad, Soyad, Yas)

Bu durumda

$A_1 = \text{ÖğrenciNo}$, $A_2 = \text{Ad}$, $A_3 = \text{Soyad}$, $A_4 = \text{Yas}$

Derece: 4 (nitelik sayısı)

İlişkinin Matematiksel Tanımı



ÖğrenciNo (A_1)	Ad (A_2)	Soyad (A_3)	Yas (A_4)
1001	Ayşe	Yılmaz	21
1002	Mehmet	Demir	22
1003	Zeynep	Çelik	20
1004	Ahmet	Kaya	23

- Bu tablo, $R(\text{ÖğrenciNo}, \text{Ad}, \text{Soyad}, \text{Yas})$ ilişkisini gösterir ve:
- Derece (Degree) = 4 (çünkü 4 nitelik var).
- Eleman sayısı (Cardinality) = 4 (çünkü 4 satır var).

İlişkinin Matematiksel Tanımı



Kartezyen Çarpım

- ▶ İki ilişkinin Kartezyen çarpımı ($R \times S$), birinci ilişkideki tüm satırların, ikinci ilişkideki tüm satırlarla birleşmesiyle oluşur.
- ▶ Sonuç kümesi, iki tablonun satırlarının tüm kombinasyonlarını içerir.
- ▶ Sonuç tablosunun derecesi (sütun sayısı), iki tablodaki toplam nitelik (sütun) sayısına eşittir.
- ▶ Sonuç tablosunun eleman sayısı, ilk tablodaki satır sayısı $\times$ ikinci tablodaki satır sayısıdır.
- ▶ **Örnek**
 - $D_1 = \{2, 4\}$
 - $D_2 = \{1, 3, 5\}$ **Kartezyen Çarpım ($D_1 \times D_2$)?**
 - $D_1 \times D_2 = \{ (2,1), (2,3), (2,5), (4,1), (4,3), (4,5) \}$

İlişkinin Matematiksel Tanımı



Kartezyen Çarpım ve Alt Küme İlişkileri

- Kartezyen çarpımın **herhangi bir alt kümesi bir ilişkidir**. Belirli koşullara göre hangi çiftlerin dahil olacağını seçebiliriz.

$D_1 = \{2,4\}$ ve $D_2 = \{1,3,5\}$ olsun

- **Örnek 1:** Belirli Çiftler İçeren İlişki $R = \{(2,1), (4,1)\}$
- **Örnek 2:** Belirli Bir Şarta Göre Seçilen İlişki
 - İkinci eleman 1 olmalıdır şartı seçilsin
 - $R = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } y = 1\}$
 - Sonuç: $R = \{(2,1), (4,1)\}$
- **Örnek 3:** İlk Eleman İkincinin 2 Katı Olmalıdır
 - $S = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } x = 2y\}$
 - Sonuç: $S = \{(2,1)\}$

İlişkinin Matematiksel Tanımı

Kartezyen Çarpım ve Alt Küme İlişkileri



► $D_1 = \{1, 3\}$, $D_2 = \{2, 4\}$, $D_3 = \{5, 6, 7\}$ olacak şekilde 3 küme ele alınsın. O halde **kartezyen çarpım**

► $D_1 \times D_2 \times D_3 =$
 $\{(1,2,5), (1,2,6), (1,2,7), (1,4,5), (1,4,6), (1,4,7), (3,2,5),$
 $(3,2,6), (3,2,7), (3,4,5), (3,4,6), (3,4,7)\}$

olarak hesaplanır.

Bu sıralı üçlülerin herhangi bir alt kümesi bir ilişkidir.

D_1	D_2	D_3
1	2	5
1	2	6
1	2	7
1	4	5
1	4	6
1	4	7
3	2	5
3	2	6
3	2	7
3	4	5
3	4	6
3	4	7

İlişkinin Matematiksel Tanımı



n Kümenin Kartezyen Çarpımı Tanımı:

- ▶ $D1 \times D2 \times \dots \times Dn = \{(d1, d2, \dots, dn) \mid d1 \in D1, d2 \in D2, \dots, dn \in Dn\}$
- ▶ Bu ifade, her elemanın sırasıyla $D_1, D_2, \dots, D_n$ kümelerinden seçildiği tüm sıralı n -lilerin kümesini tanımlar.
- ▶ Genellikle şu şekilde gösterilir: Denklemi buraya yazın.

$$\prod_{i=1}^n D_i$$

- ▶ Bu Kartezyen çarpımdan herhangi bir n çoklu kümesi, n küme üzerinde bir ilişkidir.

İlişki Şeması ve İlişkisel Veritabanı Şeması



► İlişki Şeması

- Nitelik ve alan adı çiftlerinin bir kümesiyle tanımlanan ilişkiyi temsil eder.
- Bir tabloyu oluşturan sütunların (niteliklerin) yapısını belirler.
- **Örnek:** Ogrenci(OgrenciNo, Ad, Soyad, Yas)

► İlişkisel Veritabanı Şeması

- İlişki şemalarının kümesidir ve her biri ayrı bir adla tanımlanır.
- Veritabanındaki tüm tabloların şemalarının bir bütün olarak gösterimidir.
- **Örnek:**
 - {Ogrenci(OgrenciNo, Ad, Soyad, Yas), Ders(DersNo, DersAdi)}

Bu şemalar, veritabanı tasarımının temel yapı taşlarını oluşturur ve her ilişki kendi içinde bir anahtar (primary key) barındırabilir.

İlişkilerin (Tablo) Özellikleri

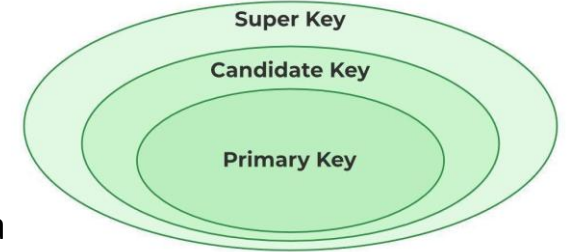


- ▶ **Sırasızdır:** Satırların (tuple) belirli bir sıralaması yoktur.
- ▶ **Tekrarsızdır:** Aynı satır (tuple) ilişkide iki kez bulunamaz; çift (duplicate) satır olamaz.
- ▶ **Benzersiz ad:** Her nitelik (sütun) benzersiz bir ada sahiptir.
- ▶ **Nitelikler sırasızdır:** Sütunların (niteliklerin) soldan sağa hangi sırada yazıldığı sonucu değiştirmez; önemli olan sütun adları ve değerleridir.
- ▶ **Her hücre tek bir değer içerir:** Her hücrede yalnızca tek (atomik/bölünemez) değer bulunur.
- ▶ **Alan (domain) birliği:** Her sütun için geçerli değerler bir alan (domain) ile tanımlanır; sütundaki değerler bu alana uymalıdır (tip/kısıt).
- ▶ **Her sütun aynı veri tipindedir:** Bir sütundaki tüm değerler aynı türdedir (ör. sayı, metin, tarih).

İlişkisel Anahtarlar



- ▶ **Süper Anahtar (Super Key):** Bir tuple'ı benzersiz tanımlayan herhangi bir nitelik veya nitelik kombinasyonudur. Aday anahtarlar süper anahtarın bir alt kümesidir. Tcno veya (TCno, isim)
- ▶ **Aday Anahtar (Candidate Key):** Bir tuple'ı (satırı) benzersiz şekilde tanımlayan nitelik veya nitelik grubudur. Bir nitelik gruptan çıkarılınca benzersizlik bozulur. Bir ilişkide birden fazla aday anahtar olabilir.
- ▶ **Birincil Anahtar (Primary Key):** Seçilen aday anahtardır. Her tuple için benzersizdir ve boş değer içeremez.
- ▶ **Yabancı Anahtar (Foreign Key):** Bir başka tablodaki birincil anahtara referans veren niteliklerdir. Tablolar arasında bağlantı kurulmasını sağlar.



Günümüzde

- ▶ PK olarak surrogate key kullanılır: id BIGINT auto-increment ya da
- ▶ Doğal iş anahtarları (ör. email, tckn, sku, phone) PK yapılmaz, ama UNIQUE ile korunur.

İlişkisel Anahtarlar



Neden çoğu ekip surrogate (sanal) PK tercih ediyor?

- ▶ **Değişmeyen kimlik:** İş alanları değişebilir (email değişir, telefon değişir, SKU revize edilir). PK değişirse referanslar zor.
- ▶ **Join ve indeks maliyeti:** Tek, dar (8 byte gibi) bir PK ile join yapmak çoğu DB’de verimli.
- ▶ **Dağıtık sistemler:** UUID/ULID gibi anahtarlar farklı servislerde çakışmadan üretilebilir.
- ▶ **ORM/uygulama kolaylığı:** Çoğu framework id üzerinden ilişki kurmayı kolaylaştırır.

Bugünkü doğru pratik: PK = surrogate, aday anahtarlar = UNIQUE

Örnek: Users

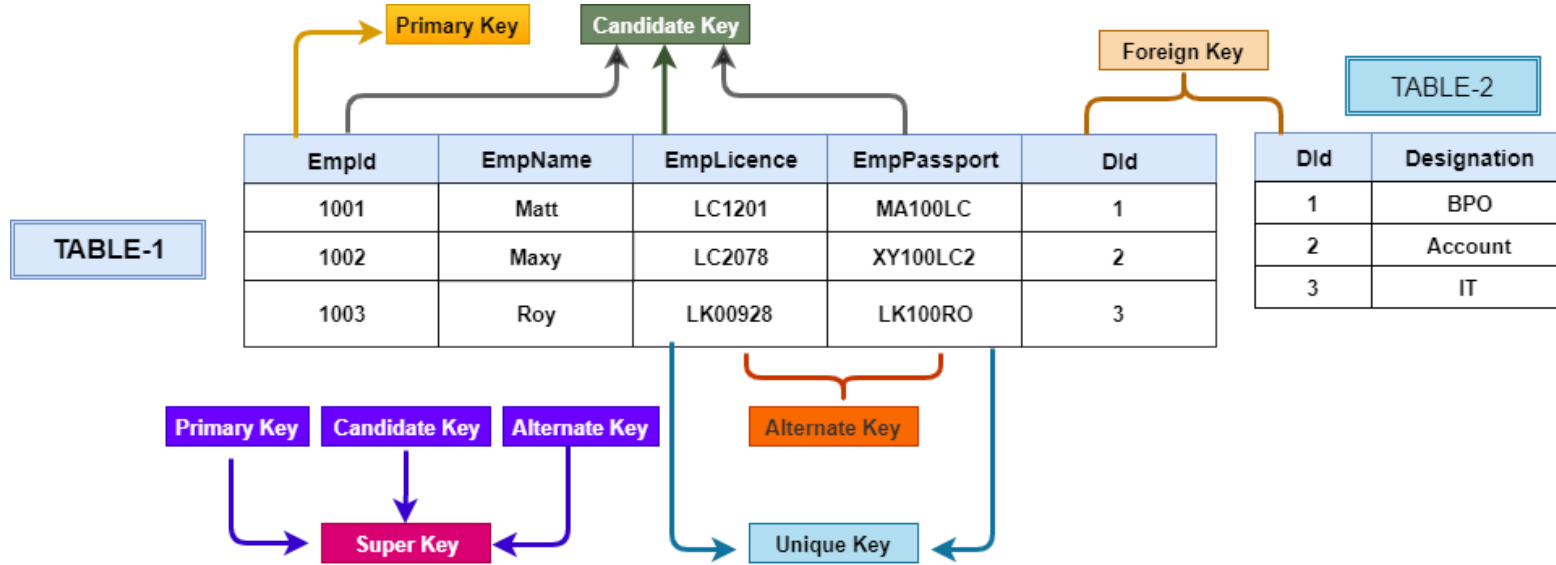
PK: id (auto increment / UUID)

Candidate key (iş kuralı): email (benzersiz olmalı)

Uygulama:

id → PK, email → UNIQUE

İlişkisel Anahtarlar



Bütünlük Kısıtları



Varlık Bütünlüğü (entity integrity):

- ▶ Birincil anahtarın boş (NULL) olamaz ve her değeri benzersiz olmalıdır.
- ▶ Her tablonun bir birincil anahtarı olmalıdır.

Referansiyel Bütünlük(referential integrity):

- ▶ **Bir ilişkide yabancı anahtar (foreign key) varsa, aşağıdaki koşullardan birini sağlamalıdır:**
 - Yabancı anahtar değeri, hedef ilişkideki bazı tuple'ların bir aday anahtar (candidate key) değeri ile eşleşmelidir.
 - Yabancı anahtar NULL olabilir, yani hiçbir ilişkiye bağlanmayabilir.

Bu bütünlük kısıtları, veri tutarlılığını koruyarak hatalı veri girişlerini önlemeye yardımcı olur.

Bütünlük Kısıtları



Genel Kısıtlar

- ▶ İşletmenin bazı yönlerini tanımlamak veya sınırlamak amacıyla belirlenen ilave kurallardır.
- ▶ Kullanıcılar veya veri tabanı yöneticileri tarafından tanımlanır.
- ▶ İş mantığını desteklemek için kullanılır.
- ▶ **Örnekler:**
 - Bir çalışanın maaşı asgari ücretin altında olamaz.
 - Bir öğrencinin aldığı ders sayısı 10'u geçemez.
 - Bir sipariş, yalnızca kayıtlı bir müşteri tarafından verilebilir.

Bu kısıtlar, verinin doğruluğunu ve iş süreçleriyle uyumluluğunu sağlamak için kullanılır.

Görünümler (Views)



► Temel İlişki:

- Tuple'ları fiziksel olarak veritabanında saklanan kavramsal şemadaki bir varlığa karşılık gelir.
- Gerçek verilerin bulunduğu fiziksel tabloları ifade eder.

► Görünüm

- Başka bir ilişkiyi üretmek için temel ilişkiler üzerinde çalışan bir veya daha fazla ilişkisel işlemin dinamik sonucudur.
- Verileri daha güvenli ve esnek bir şekilde sunmak için kullanılır.

► Görünüm Özellikleri

- Veri tabanında fiziksel olarak saklanmaz, sadece bir sorgu sonucu oluşturulur.
- Karmaşık sorguları basitleştirir ve kullanıcıya özelleştirilmiş veri sunar.
- Güvenliği artırır, belirli kullanıcıların yalnızca ilgili verileri görmesini sağlar.

Görünümlerin Güncellenmesi



- ▶ **Görünümler üzerinde yapılabilecek değişiklikler belirli kısıtlamalara tabidir.**
- ▶ **Güncellemelere izin verilen durumlar:**
 - Sorgu tek bir temel ilişki içeriyorsa ve bu ilişkinin aday anahtarı varsa güncellenebilir.
- ▶ **Güncellemelere izin verilmeyen durumlar:**
 - Görünüm birden fazla temel ilişkiyi kapsıyorsa.
 - Görünüm grup veya küme işlemleri (GROUP BY, HAVING, SUM, AVG gibi) içeriyorsa.
 - Vb.

Bu kısıtlamalar, veri tutarlılığını korumak ve karmaşık sorguların yanlışlıkla değiştirilmesini önlemek amacıyla uygulanır. Detayları konusu geldiğinde anlatılacaktır.

Görünümlerin Güncellenmesi



- ▶ Görünümler üzerinde yapılabilecek değişiklikler belirli kısıtlamalara tabidir.
- ▶ Güncellemelere izin verilen durumlar:

<pre>CREATE VIEW BilgisayarOgrencileri AS SELECT OgrenciNo, Ad, Soyad FROM Ogrenci WHERE Bolum = 'Bilgisayar';</pre>	<pre>UPDATE BilgisayarOgrencileri SET Ad = 'Mehmet' WHERE OgrenciNo = 101;</pre>
--	--

- ▶ Güncellemelere izin verilmeyen durumlar:

<pre>CREATE VIEW OgrenciNotlar AS SELECT Ogrenci.Ad, Ogrenci.Soyad, Notlar.DersAdi, Notlar.Not FROM Ogrenci JOIN Notlar ON Ogrenci.OgrenciNo = Notlar.OgrenciNo;</pre>	<pre>UPDATE OgrenciNotlar SET Not = 90 WHERE Ad = 'Ali';</pre>
--	--

Bölüm sonu- Teşekkürler.

Dr. Öğr. Üyesi Elvan DUMAN – eduman@mehmetakif.edu.tr