

VERI MODELI

GİRİŞ



Verileri, tasarımcılar, programcılar ve son kullanıcılar farklı şekillerde görür.



Aynı verinin farklı görünümleri,
organizasyonun işleyişini yansıtmayan
tasarımlara yol açar.

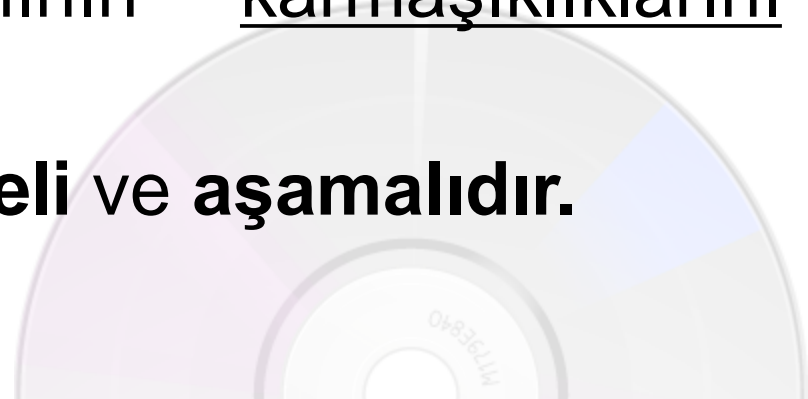


Veri modelleme, veritabanı tasarımının karmaşıklığını azaltır.

Veri Modelleme ve Veri Modelleri

Veri modelleri

- Karmaşık gerçek dünyadaki **veri yapılarının nispeten basit temsillerini sağlar.**
 - Genellikle grafiksel olarak gösterilir.
- **Model:** Gerçek dünyadaki *bir nesnenin veya olayın bir soyutlaması*
 - Gerçek dünya ortamının karmaşıklıklarını anlamada faydalıdır.
 - Veri modelleme **yinelemeli ve aşamalıdır.**



Veri Modellerinin Önemi

Veriler farklı kişiler tarafından farklı şekillerde görülür.

- ❑ Örneğin, **envanter yöneticisi** envanter seviyeleriyle daha fazla ilgilenir.
- ❑ **Satınalma müdürü**, daha çok kalemlerin **maliyeti** ve bu kalemlerin **tedarikçileri** ile olan ilişkilerle ilgilenir.
- ❑ **Uygulama programcıları**, *veri konumu, biçimlendirme* ve *özel raporlama* gereksinimleriyle ilgilenir.
- ❑ ...

Veri Modellerinin Önemi

- ❑ **Veri modeli**, çeşitli kullanıcılar için verileri düzenler.
- ❑ **Veri modeli tasarımı**, *uygulama programcısı* ve *son kullanıcı* arasındaki etkileşimi kolaylaştırır.
- ❑ **Veri modeli bir soyutlamadır.**
 - Gerekli veriler, veri modelinden bağımsız olarak çizilemez.

Veri Modeli Temel Yapı Taşları

1. Varlık (Entity): Hangi verilerin toplanacağı ve saklanacağı hakkında herhangi bir şeydir.

- **Varlık 3 ana bölüme ayrılabilir:**

- A. Genel (Generals)

- B. Kavramlar (Concepts)

- C. Olaylar (Events)

Veri Modeli Temel Yapı Taşları

A. Genel (Generals)

- **Kişisel bilgiler:** öğrenci, çalışan, öğretmen, doktor vb.
- **Yerler:** restoran, şirket, hastane, hayvanat bahçesi, sınıf
- **Nesneler:** makine, araba, kitap

B. Kavramlar (Concepts)

- konu, fakülte, bölüm

C. Olaylar

- kayıt, ödünç alma, iade vb.

Veri Modellerinin Önemi

2. Nitelik: Bir varlığın özelliğidir.

Örneğin, bir **MÜŞTERİ** varlığı, müşterinin *soyadı*, müşterinin *adı*, müşteri *telefon numarası*, müşteri *adresi* ve müşteri *kredi limiti* gibi özelliklerle tanımlanır.

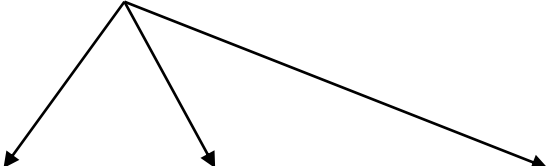
❖ **Nitelikler, dosya sistemlerindeki alanların eşdeğeridir.**

Veri Modellerinin Önemi

Entity 1 (Varlık): AUTHOR

YazarID	Yazar Adı	Yazar Soyadı	Yazar Telefon No
1	Robert	Adams	(222) 335 25 22
2	Marvin	Bittinger	(222) 114 22 52
3	Laurence	Hoffmann	(333) 322 22 22
4	Gerald	Bradley	(333) 554 44 44
5	Gerald	Folland	(555) 222 22 22
6	Susan	Colley	(233) 225 22 55
7	Frank	Beatrous	(255) 352 22 55
8	Deborah	Hughes	(333) 558 55 55
9	Michael	Robert	(333) 558 55 55
10	George	Thomas	(255) 555 44 66
11	Ross	Finney	(255) 557 44 77
12	Charles	Edwards	(322) 444 55 66
13	David	Penney	(333) 555 77 88
14	Richard	Madsen	(444) 111 88 11

Attribute (Öznitelikler)



YayıneviID	Yayınevi Adı	Yayınevi Telefon No
1	Addison Wesley	(222) 320 19 11
2	McGraw Hill	(322) 255 55 55
3	Prentice Hall	(445) 222 55 44
4	John Wiley & Sons	(444) 211 11 55

Entity 2 : PUBLISHER

Veri Modeli Temel Yapı Taşları

3. İlişki (Relationship): Varlıklar arasındaki ilişkiyi tanımlar.

- **Bire bir (1:1) ilişki**
- **Bire çok (1:M) ilişki**
- **Çoktan çoğa (M:N veya N:M) ilişkisi**

Bire bir (1:1) İlişki

- İlgili varlıklarda **yalnızca bir kayıt olabilir.**
 - Örneğin, bir kullanıcının bir parolası vardır.
 - Bir telefon numarası bir kişiye ait olabilir.
- Bire bir ilişkilerde, ilişkinin yönü bağımsız bir varlıktan bağımlı bir varlığa doğrudur.
- Örneğin, **PERSONEL** varlık bağımsız bir varlık ve **MAAŞ** varlığı bağımlı bir varlık olarak görülebilir.
- İki nesne bağımsızsa, herhangi bir yön seçilir.

Bire bir (1:1) İlişki

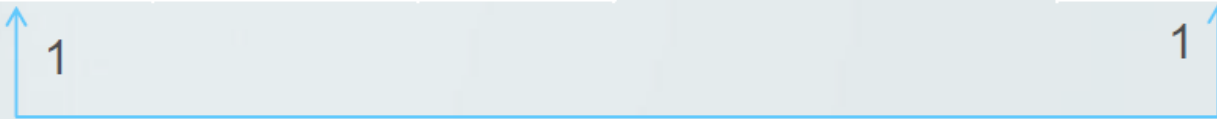
- İlişkilerin katılım özelliği de iki yönlü olarak incelenmiştir.
- Örneğin, *MÜŞTERİ* ve *ADDRES* varlığı arasındaki ilişki:
 - *Müşterilere bir adres atanabilir.*
 - *Her adres bir müşteriye atanmalıdır.*
 - Müşteri-adres ilişkisinde müşteri varlığı için herhangi bir kısıtlama yoktur.
 - Adres varlığına baktığınızda her adrese en az bir kişi atanmalıdır.

Bire bir (1:1) İlişki

Örneğin, bir veritabanında iki tablo arasındaki birebir ilişki gösterilir.

KimlikBilgileri		
TcNo	Ad	Soyad
121	Ayşe	Berk
243	Kemal	Kurt
982	Musa	Tufan

Sifre	
TcNo	Sifre
121	sifre1
243	asd123
982	111111



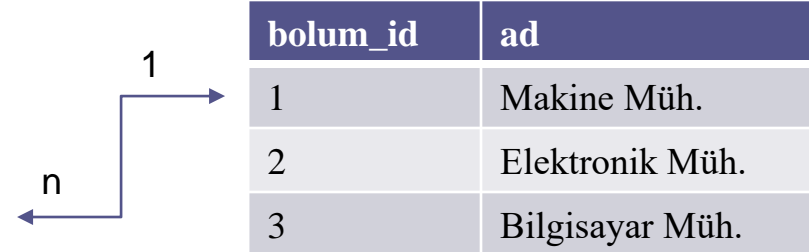
Bire çok (1:M) ilişki

- Bağımsız varlığın bir kaydı varken, bağımlı varlıktaki kayıt sayısı $1, \dots, N$ olabilir.
 - Örneğin, veritabanı tasarımcıları, “**Ressam**” *boyar* “**Tablo**” ilişkisini 1:M olarak etiketler.
 - Örneğin, “**MÜŞTERİ**” *oluşturur* “**FATURA**” ilişkisi de 1:M olarak etiketlenir.
- Bire-çoğa ilişki yönü, bağımsız (ana) bir varlıktan bağımlı (detay) bir varlığa doğrudur.

Bire çok (1:M) ilişki

- Örneğin, iki tablo arasındaki bire çok ilişki bir veritabanında gösterilir.

ogrNo	ad_soyad	adres	dtarih	bolum_id
1236	Temel SERT	Mimar Sinan Mah.	15.06.1995	1
6789	Fadime AKSU	Erikli Mah.	14.03.1989	1
1234	Dursun AKSU	Esenevler Mah.	12.05.1987	1
1456	Cemal MEŞE	Çayağzı Mah.	08.09.1992	3
4567	Erhan KIRMIZI	Tepe Mah.	23.05.1998	2



Çoktan çoğa (M:N veya N:M) ilişkisi

- İki varlık, birden fazla kaydı karşılıklı olarak eşleştirebilir.
 - Örneğin, bir çalışan birçok iş becerisini öğrenebilir ve her bir iş becerisi, birçok çalışan tarafından öğrenilebilir.
- Veritabanı tasarımcıları, “ÇALIŞAN” *öğrenir* “BECERİYİ” ilişkisini M:N olarak etiketler.
- Benzer şekilde “ÖĞRENCİ” *alır* “DERS” ifadesi M:N olarak etiketlenebilir.
- Çoktan çoğa ilişkilerde herhangi bir yön seçilir.

Çoktan çoğa (M:N veya N:M) ilişkisi

- Örneğin, iki tablo arasındaki bire-çok ilişki bir veritabanında gösterilir.



Veri Modeli Temel Yapı Taşları – Kısıtlama

- **Kısıtlamalar (Constraint)**, önemlidir çünkü veri bütünlüğünü sağlamaya yardımcı olurlar.
- **Kısıtlamalar, normalde kurallar şeklinde ifade edilir:**
 - *Bir çalışanın maaşı, 6.000 ile 350.000 arasında değerlere sahip olmalıdır.*
 - *Bir öğrencinin genel not ortalaması 0.00 ile 4.00 arasında olmalıdır.*
 - *Her bir dersin sadece bir öğretmeni olmalıdır.*
 - *Öğrencilerin alabileceği ders sayısında bir sınırlama olmalıdır.*

İş kuralları (Business Rules)



Verilerin toplanması, ancak uygun şekilde tanımlanmış iş kurallarını yansıttığında anlamlı hale gelir.



Varlıkları, nitelikleri, ilişkileri ve kısıtlamaları tanımlamak için uygun şekilde yazılmış iş kuralları kullanılır.

İş kuralları (Business Rules)

- ❑ İş kurallarının anlaşılması kolay ve geniş çapta dağıtılmış (yayınlanmış) olmalıdır.
 - ❑ Organizasyondaki her kişi, iş kurallarının ortak bir yorumunu paylaşır.
- ❑ Şirket tarafından görüntülenen verilerin karakteristik özellikleri, tanımlanması gerekir.
 - ❑ Bu özelliklerinin herhangi bir kuruluşa (büyük veya küçük - bir işletme, bir devlet birimi, bir araştırma laboratuvarı vs.) uygulayın.

İş kuralları (Business Rules)

- İş kuralları, yazılı hale getirilmeli ve kuruluşun operasyonel ortamındaki herhangi bir değişikliği yansıtacak şekilde güncellenmelidir.
- Örnekler:
 - Bir müşteri birçok fatura oluşturabilir.
 - Bir fatura yalnızca bir müşteri tarafından oluşturulur.
 - Her yönetici en fazla bir departmanı yönetir.
 - Her departman bir yönetici tarafından yönetilir.
 - ...

İş Kurallarını Keşfetmek

İş kurallarının kaynakları:

- Şirket yöneticileri
- Kural koyucular
- Departman yöneticileri
- Yazılı belgeler
 - Prosedürler
 - Standartlar
- Operasyon kılavuzları
- Son kullanıcılarla yapılan doğrudan görüşmeler



İş Kurallarını Keşfetmek (2)

İş kurallarını; **belirleme** ve **belgeleme süreci**, birkaç nedenden dolayı veritabanı tasarımı için gereklidir:

- ❑ Şirketin veri görünümünü, standartlaştırmaya yardımcı olur.
- ❑ Kullanıcılar ve tasarımcılar arasında bir iletişim aracı olabilir.
- ❑ Tasarımcının verilerin doğasını, rolünü ve kapsamını anlamasını sağlar.

İş Kurallarını Keşfetmek (2)

- ❑ Tasarımcının iş süreçlerini anlamasını sağlar.
- ❑ Tasarımcının uygun **ilişki katılım kuralları** ve **kısıtlamaları geliştirmesine** ve **doğru bir veri modeli oluşturmaya** olanak tanır.

İş Kurallarını Veri Modeli Bileşenlerine Çevirme

- Genellikle **isimler**, *varlıklara* çevrilir.
- **Fiiller**, varlıklar arasındaki *ilişkilere* dönüşür.

İş kuralını veri modeline çevirme örneği:

“Bir müşteri birçok fatura üretebilir”.

Bu cümle, *iki isim* (müşteri ve faturalar) ve *bir fiil* (üret) içerir.

Varlıklar: müşteri ve fatura

İlişki: üretir.

İş Kurallarını Veri Modeli Bileşenlerine Çevirme

❑ İlişkiler çift yönlüdür.

Örneğin, iş kuralı:

- “Bir müşteri birçok fatura üretebilir”.
- “Bir fatura yalnızca bir müşteri tarafından oluşturulur.”

Bu durumda, ilişki birden çoğa (1:M)'dir.

Müşteri, “1” taraf ve fatura “çoklu” tarafı ifade etmektedir.

İş Kurallarını Veri Modeli Bileşenlerine Çevirme

**İlişki türünü doğru bir şekilde belirlemek
için genellikle iki taraflı soru sormalısınız:**

- B'nin kaç örneği, A'nın bir örneğiyle ilişkilidir?
- A'nın kaç örneği B'nin bir örneğiyle ilişkilidir?

İş Kurallarını Veri Modeli Bileşenlerine Çevirme

Örneğin:

- Bir öğrenci kaç kursa kayıt olabilir?
 - Cevap: birçok kurs.
- Bir kursa kaç öğrenci kayıt yaptırabilir?
 - Cevap: birçok öğrenci.
- Bu nedenle, **öğrenci** ve **kurs** arasındaki ilişki çoktan çoğadır (M:N).

Adlandırma Kuralları



**VERİ MODELİ
TASARIMCISI**



**UYGULAMA
PROGRAMCISI**



**SON
KULLANICI**

* Uygun bir adlandırmaların kullanılması, yukarıda belirtilen kişiler arasında iletişimi kolaylaştıracaktır.

Adlandırma Kuralları

❖ Varlık adları:

- İş ortamındaki nesneleri açıklayıcı olmalıdır.
- Kullanıcıların aşina olduğu terminolojiyi kullanmalıdır.

❖ Öznitelik adları:

- Öznitelik tarafından temsil edilen verileri de açıklayıcı olmalıdır.

Örneğin, bir kuruluştaki alıcılar, **MÜŞTERİ**, müşterinin kredi limiti, **MUS_KREDI_LIMIT** olarak adlandırılabilir.

KAYNAKLAR

- Carlos Coronel, Steven Morris, DATABASE SYSTEMS, Design, Implementation, and Management, Cengage Learning, 13. edition