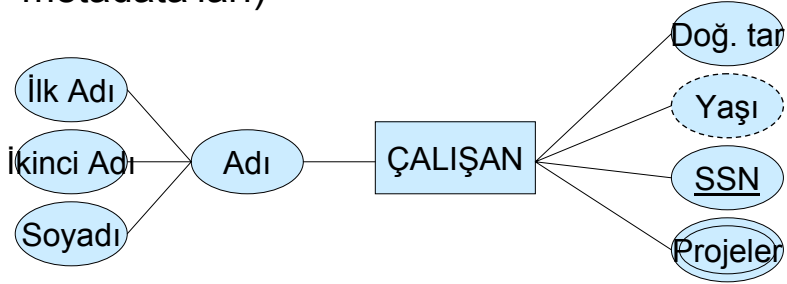
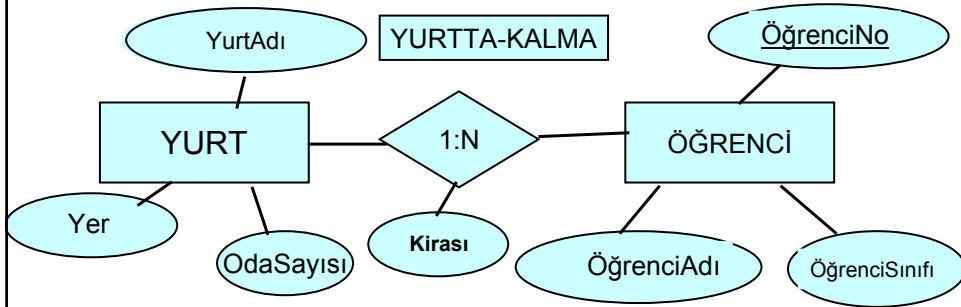


## Varlık-İlişki Şemaları

- Bir varlığı belirlemeye yarayan, o varlıkla etkileşim kurmak ya da o varlığı kullanmak için gerekli önemli özellikleri (yani varlıkların metadata'ları)



## Varlık-İlişki Şemaları



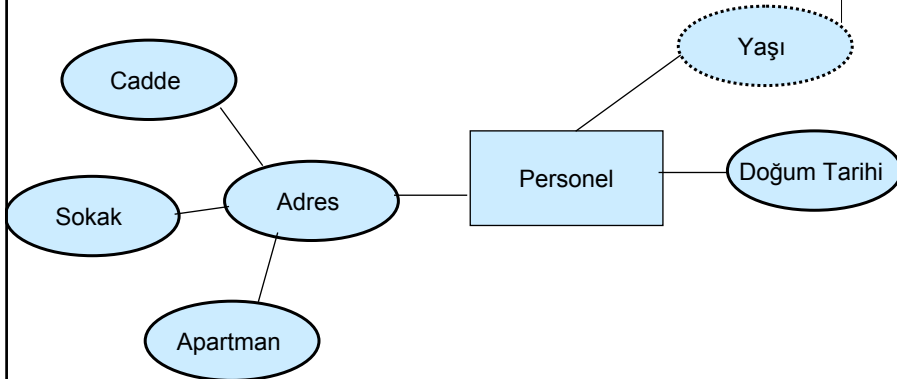


## Varlık-İlişki Şemaları

- Bir personel varlığının aşağıda belirtilen özelliklere sahip olduğu varsayılınsın
- Adı
- Cadde
- Sokak
- Apartman
- Doğum Tarihi
- Cadde, sokak ve apartman nitelikleri adres ile birleştirilecektir.
- Yaş doğum tarihinden elde edilecektir.



## Varlık-İlişki Şemaları



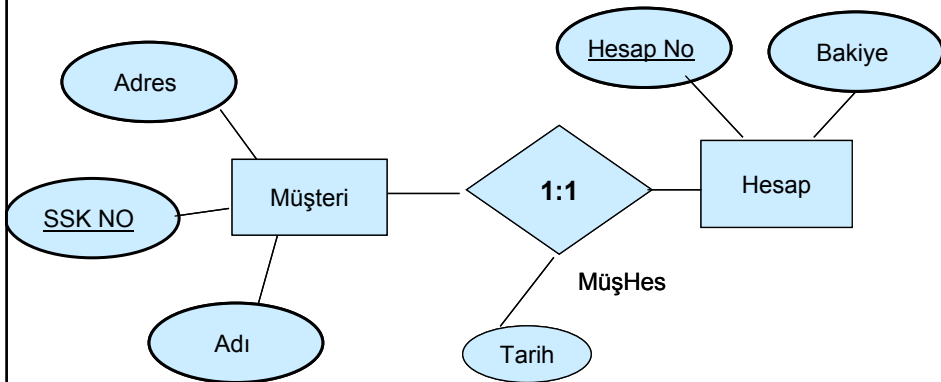


## Varlık-İlişki Şemaları

- “Müşteri” ve “Hesap” isimli iki varlık kümesinin nitelikleri aşağıdaki gibidir:
- Müşteri: Adı, SskNo, , Adres
- Hesap: Hesap No, Bakiye
- Bu veriler ile varlık-ilişki şemasını oluşturunuz.



## Varlık-İlişki Şemaları



## Varlık-ilişki şemalarının tablo haline dönüştürülmesi



Müşteri={SskNo, adı,adres}

Adı	<u>Ssk No</u>	Adres

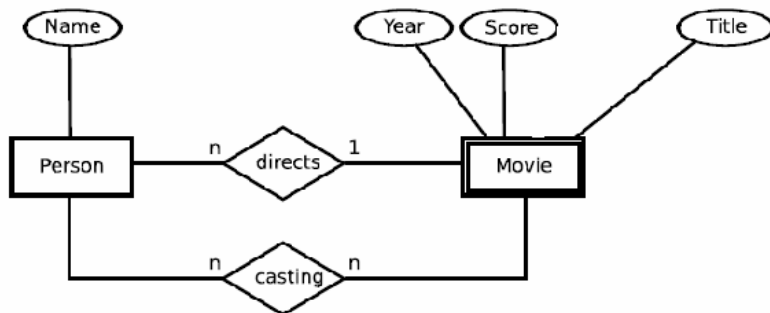
Hesap={Hesap No, Bakiye}

Hesap No	Bakiye

Müşteri Hesap ilişkisi için={SSkNo, Hesap No, Tarih}

<u>Ssk No</u>	<u>Hesap No</u>	Tarih

## Varlık-İlişki Şemaları

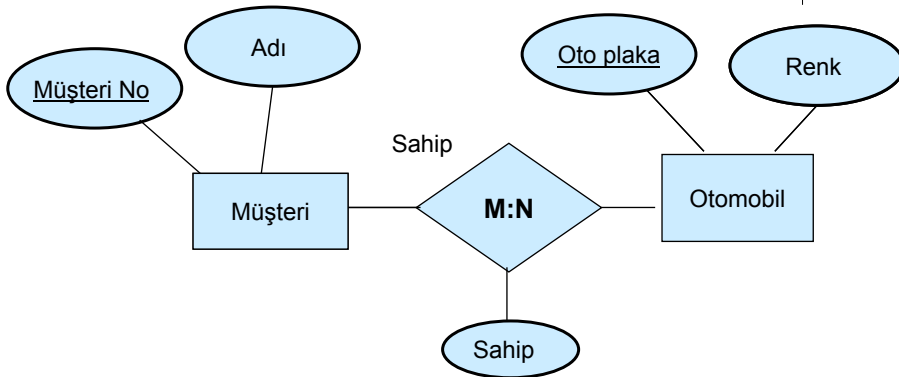




## Varlık-İlişki Şemaları

- Bir müşteri birden fazla otomobile sahip olabilir ve her otomobil modeline birden fazla müşteri sahip olabilir.
- Bu durumda otomobiller ve müşteriler arasındaki ilişki çoktan-çoğa biçimindedir.
- Bu ilişkinin varlık-ilişki şemasını çiziniz.

## Varlık-İlişki Şemaları



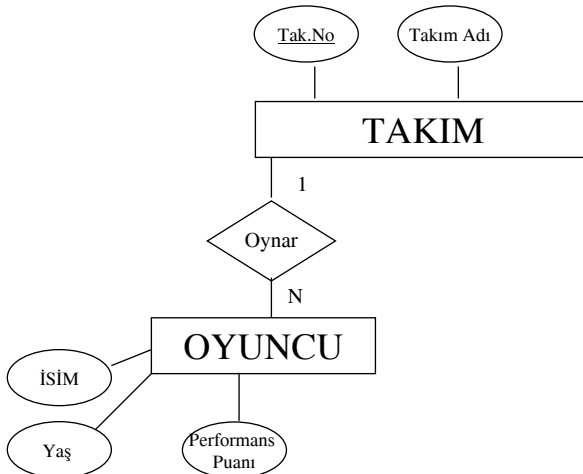
Müşteri ve otomobil varlıkları ve sahip ilişkisi için varlık-ilişki şeması

## Varlık-İlişki Şemaları

- Müşteri={müşteri no, adı}
- Otomobil={oto model, renk}
- Sahip={müşteri no, oto plaka, tarih}
- Sahip isimli ilişkiye dayanarak aşağıdaki tablo oluşturulabilir.

Müşteri no	oto plaka	Tarih
345	34 GF 67	12.12.2005
346	45 HN 34	15.11.2002
347	36 BN 67	12.12.2005
348	34 AV 45	15.11.2002

## Varlık-İlişki Şemaları





## İlişkisel Model

- İlişkisel model, günümüzde en yaygın biçimde kullanılan bir modeldir.
- İlişkisel model varlıklar arasındaki bağlantının içerdiği değerlere göre sıralanması esasına dayanır.
- Bu yaklaşımda veri tabanındaki tüm ilişkiler tablolar biçiminde tanımlanmaktadır.



## İlişkisel Veri Tabanı

- İlişkisel veri tabanı, her biri özel isimlere sahip tablolardan oluşur.
- Burada her bir tablo bir varlığa veya bir ilişkiye karşılık gelmektedir.
- Tablonun sütunları nitelikleri, satırları ise bu niteliklerin değerlerini ifade eder.
- Her bir satır bir “kayıt” olarak da düşünülebilir.
- Anahtar alan tablonun tanımlayıcısıdır.



## Tablonun özellikleri

- Tablolar sütunlardan oluşur.
- Her bir sütunun ayrı bir adı vardır.
- Her bir sütun aynı niteliğin tanımladığı aynı etki alanının belirlediği değerleri içerir.
- Satırların ve sütunların sırası önemsizdir.
- Her bir satır birbirinden farklıdır.

## Soyutsal Katmanlar (Levels of Abstraction)



- **Fiziksel Katman (Physical level):** Bir kaydın nasıl saklanacağını tanımlar(Örneğin,müşteri).
- **Mantıksal Katman (Logical level):** Bir verinin nasıl veritabanında saklanacağını ve veriler arasındaki ilişkileri tanımlar.

**tip (type) *musteri* = kayıt (record)**

```
musteri_id : string;  
musteri_adi : string;  
musteri_sokak : string;  
musteri_il : integer;  
bitiş (end)
```

- **Görüntü Katmanı (View level):** Tasarımı kullanıcıdan saklar( Örneğin veri tipi veya hangi bilgilerin görüntüleneceği)



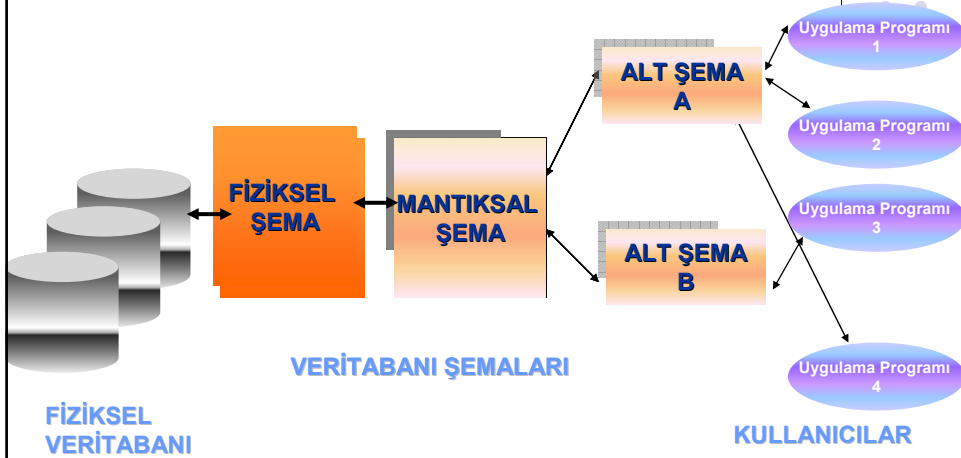


# Nesneler ve Şemalar

- Programlama dillerindeki tip ve değişkenlere benzerler
- **Şema (Schema)** – Veritabanının mantıksal yapısı
  - Örnek: Veritabanı, Müşteri ve hesap bilgileri ile bunlar arasındaki ilişkiyi barındıran bir kümedir.
  - **Fiziksel Şema (Physical schema)**: Fiziksel düzeyde veritabanı tasarımı (Dosyanın sabit diskteki yeri, büyüklüğü..)
  - **Mantıksal Şema (Logical schema)**: Mantıksal düzeyde veritabanı tasarımı (veri alanları, ilişkiler)
- **Nesneler (Instance)** – Zamandaki herhangi bir noktadaki veritabanı içerisindeki içerik.
  - Bir değişkenin değerine benzerdir.
- **Fiziksel Veri Bağımsızlığı (Physical Data Independence)** – Mantıksal şemayı değiştirmeden fiziksel şemayı değiştirme kabiliyeti
  - Uygulamalar mantıksal şemaya bağlıdır.
  - Genelde , değişik katmanlar ve bileşenler arasındaki arabirimler öyle tanımlanmalıdır ki bazı bölümlerin değiştirilmesi diğerlerini fazla etkilememeli.



# VERİTABANI ÖRNEĞİ



# VERİ MODELLERİ



- Asağıdakilerini tanımlayan araç topluluğu
  - Veri (Data)
  - Veri İlişkileri (Data relationships)
  - Veri Kısıtlamaları (Data constraints)
- İlişkisel Model (Relational model)
- Varlık-ilişki veri modeli (Entity-Relationship data model) (Coğunlukla veri tabanı dizaynı için)

# İlişkisel Veritabanı



- İlişkisel veritabanı ilişkisel model bazlıdır.
- Veri etrafındaki bilgi ve ilişkiler tablolar tarafından gösterilir

Öznitelikler (Attributes)

<i>customer_id</i>	<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>	<i>account_number</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-101
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-201
677-89-9011	Hayes	3 Main St.	Harrison	A-102
182-73-6091	Turner	123 Putnam St.	Stamford	A-305
321-12-3123	Jones	100 Main St.	Harrison	A-217
336-66-9999	Lindsay	175 Park Ave.	Pittsfield	A-222
019-28-3746	Smith	72 North St.	Rye	A-201

## İlişki Gösterimi



- İlişkinin o anki değerleri (*relation instance*) bir tablo tarafından gösterilir.
- $r$  deki bir  $t$  elemanı bir değerdir ve tablodaki bir satır (row) ile gösterilir.
- Değerlerin sırası önemli değildir. (Değerler keyfi sırada olabilir)

Öznitellikler (veya sütunlar)  
attributes  
(or columns)

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

Değerler  
(yada satırlar)  
tuples  
(or rows)

*customer*

21

## Veritabanı



- Bir veritabanı birden fazla ilişkiye sahiptir.
- Bir şirketin bilgisi birden fazla parçaya bölünmüştür, her parça bilginin bazı bölgelerini ilişkilendirir

*account* : hesaplar hakkındaki bilgiyi tutar.

*depositor* : hangi müşterinin hangi hesabı tuttuğunu gösteren bilgiyi saklar

*customer* : müşteri hakkındaki bilgileri tutar

- Bütün bilgilerin tek bir ilişkide saklanması örneğin  
*bank(account\_number, balance, customer\_name, ..)*  
aşağıdaki sonuçlara yol açabilir
  - Bilginin tekrarlanması (repetition of information)
    - Örneğin iki müşteri tek hesaba sahip (ne tekrarlar?)
  - Boş değerlerin ihtiyacı
    - Örneğin hesabı olmayan müşterinin gösterimi
- Normalizasyon teorisi (Normalization theory) ilişkisel veri tabanının tasarımından bahsetmektedir

22

## TEMEL KAVRAMLAR



- **Alan(Field)** : Veritabanı tabloları içerisinde saklanacak verinin içeriğine göre, fiziksel tipi belirlenen (Sayı,String vb.) ve isimlendirilen bölümlere denir.(Örnek : Bir field içerisinde bir iş yerindeki personele ait "isim" bilgisi saklanacak ise, programın kontrolü açısından alan(field) isminde içerik ile ilgili seçilmesi tercih edilir yani "Personellsim" veya benzer bir alan(field) ismi seçilmelidir.
- **Tablo(Table)** : İçeriklerine göre ayrıştırılmış alan(field) topluluklarına tablo denir, tablolar veri tabanlarını oluşturan bilgi depolarıdır.
- Örnek= Bir işyeri veritabanını ele alalım, iş yerinin bünyesinde barındırdığı departman adedince veri tabanının içerisinde tablo oluştururuz tablo isimlerinin de departmanlar ile ilişkili olarak isimlendiririz (Personel Müdürlüğü departmanı için "personel" tablo ismi ,
- Muhasebe departmanı için "muhasebe" vb.) daha sonra bu tabloların içerisine alanlar oluştururuz.(Personel tablosu içerisine
- Personellsim", "PersonelYas" vb. isimleri taşıyan alanlar açmamız mümkündür.)

## BİRİNCİL VE YABANCI ANAHTAR



- **Birincil Anahtar(Primary Key)** : Üzerinde işlem yapılan tabloya ait kayıtları benzersiz olarak tanımlayan alanlardır.
- Örneğin bir okulu ele alalım burda öğrencileri benzersiz biçimde tanımlayabilen en önemli öge şüphesiz ki öğrenci numarasıdır.Bir okulda,İsim,soyisim gibi kimlik bilgileri aynı olabilecek bir çok öğrenci mevcut olabilir fakat,hiç bir öğrencinin, o öğrenciyi tanımlayan, öğrenci numarası aynı olamaz benzer bir mantık ile telefon numaraları da düşünülebilir.
- **Yabancı Anahtarlar(Foreign Keys)** : Tablo içerisindeki verilerin birbirleri ile iletişim kurabilmeleri amacı ile kullanılan benzersiz olması gerekmeyen alanlardır.
- Örneğin içerisinde "Ogrenci\_No" birincil anahtarını barındıran "Ogrenciler" isimli tablonun var olduğunu varsayalım ayrıca "Notlar" isimli bir tablonun içerisinde, aynı "Ogrenci\_No" alanını çeşitli defalar yabancı anahtar olarak kullanmamız gerekebilir (Çünkü, genellikle bir öğrencinin birden fazla dersi ve dolayısıyla "Notlar" isimli tabloya işlenmesi gereken birden fazla sınav notu olacaktır.)
- Bağımsız tablolarda bu şekilde organize edilmiş veriye "**ilişkisel(Relational)**" bu veriyi içeren veritabanına ise "**ilişkisel veritabanı**" ismi verilir.Veritabanlarındaki verinin okunması ve yönetilmesi için kullanılan ortak sorgulama diline **Yapısal Sorgulama Dili (Structured Query Language (SQL))** denir.

# BİRİNCİL VE YABANCI ANAHTAR



BÖLÜM	
B_no	isim
1	muhasabe
2	insan kaynakları
3	IT

BİRİNCİL ANAHTAR

ÇALIŞANLAR		
Calisan_no	B_no	isim
1	2	Nora Edwards
2	3	Ajay Patel
3	2	Ben Smith
4	1	Brian Burnett
5	3	John O'Leary
6	3	Julia Lenin

YABANCI ANAHTAR

25

## Fonksiyonel Bağımlılık



- R'nin ilişkiyi(relation) ;
- A ve B'nin bir attribute yada attribute setini temsil ettiğini düşünelim.
- Eğer R ilişkisinde her bir A değeri , tam olarak bir B değerine işaret ediyorsa ;
- B, A ya fonksiyonel olarak bağımlıdır diyebiliriz
- A -> B (A fonksiyonel olarak B'yi tanımlar.)

26

## Fonksiyonel Bağımlılık



ID İSİM ŞEHİR

123 Ahmet Ankara

324 Derya Ankara

574 Derya İstanbul

- Kişiler tablosu ile ilgili neler söyleyebiliriz?
- “Eğer ID numarasını biliyorsam, ismini de biliyorum”
- ID numarası ismi belirlemektedir.
- İSİM niteliği, ID’ye fonksiyonel bağımlıdır.

## Fonksiyonel Bağımlılık



A -> B A fonksiyonel olarak B’yi tanımlar.)

ogr_no	ogr_name	bolum	Sinif	kimlik_no
001213082	Erdem Akyıldız	Bilgisayar Mühendisliği	4	4424543212
991213068	Aslı Yılmaz	İletişim	3	4044540002

Yukarıdaki öğrenci tablosunu ele aldığımız zaman aşağıdaki bağımlılıklardan söz edebiliriz.

ogr\_no -> ogr\_name

ogr\_no -> ogr\_name , bolum , sinif , kimlik\_no

kimlik\_no -> ogr\_no , ogr\_name , bolum , sinif

## Normalleştirme



- İlişkisel veritabanı oluşturmak için normalleştirmeyi bilmek çok önemlidir.
- Normalleştirme veri kayıplarını engellemek, verinin tekrarını azaltmak, silme, güncelleme eklemede çıkan zorlukları en aza indirmek için yapılan operasyonlar toplamıdır
- Amac veritabanına etkinlik kazandırmaktır.
- Herhangi bir tablonun tekrarlı veriler içerdiği duruma birinci normal form denir.
- Birinci normal formdaki bir tabloda tekrarlayan sütunlar bulunmamalıdır.

## Birinci Normal Form



- Dikkat edilecek olursa belirli alanlarda tekrarlar yer almaktadır.

MusteriD	Adı	Soyadı	TelNo	Adres	Şehir	PlakaKodu	AlınanÜrün	Miktar	Fiyat
1	Mustafa	Kaya	2222222	Deneme Adresi	Ankara	06	Disket	5	10
2	Ahmet	Şeker	3333333	Ahmetin Adresi	Istanbul	34	CD-R	50	100
3	Ahmet	Şeker	3333333	Ahmetin Adresi	Istanbul	34	DVD-R	10	50
4	Ahmet	Şeker	3333333	Ahmetin Adresi	Istanbul	34	Mouse	2	2
5	Ahmet	Şeker	3333333	Ahmetin Adresi	Istanbul	34	CD-R	25	50
6	Ahmet	Şeker	3333333	Ahmetin Adresi	Istanbul	34	CD-R	50	100
7	Mustafa	Kaya	2222222	Deneme Adresi	Ankara	06	DVD-R	10	50
8	Gökmen	Imamoğlu	5555555	Ahmetin Adresi	Izmir	35	CD-R	50	100
9	Selim	Dener	4444444	Selimin Adresi	Ankara	06			
10	Selim	Dener	4444444	Selimin Adresi	Ankara	06			
11	Selim	Dener	4444444	Selimin Adresi	Ankara	06			
*	(AutoNumber)							0	0

Record: 9 of 11



## Birinci Normal Form

- **Satır Ekleme Sorunu:** Böyle bir tabloda ancak yeni bir ürün satın alındığında yeni bir müşteri eklenir eğer alış veriş yoksa mantıksal olarak müşteri eklemek boş kayıt eklemekle aynı anlama gelir.
- **Güncelleştirme Sorunu:** Eğer Müşterilerden birinin oturma adresi değişirse bu tablodaki bu müşteri ile ilgili tüm kayıtlar güncellenmelidir örneğimizde 5 kayıt olmasına rağmen büyük bir veri tabanında böyle küçük bir değişiklik için bile binlerce kaydı güncellemek gerekir.



## İkinci Normal Form

- Birinci normal formda tekrarların ne tür sorunlara neden olduğunu öğrendik şimdide bu sorunları çözmek için birinci normal formdaki tabloyu başka normal tablolara bölmeliyiz.
- Fonksiyonel bağımlılık göz önünde bulundurularak tablolar bölünmelidir.
- Bölünen tablolardan birinin birincil anahtarı ile bölünen diğer tablodaki birincil olmayan bir alan arasında bağımlılık varsa buna tam bağımlılıkta denir;bu duruma ikinci normal form denilir.





Adı	Soyadı	TelNo	Adres	Şehir	PlakaKodu
Mustafa	Kaya	2222222	Deneme Adresi	Ankara	06
Ahmet	Şeker	3333333	Ahmetin Adresi	İstanbul	34
Ahmet	Şeker	3333333	Ahmetin Adresi	İstanbul	34
Ahmet	Şeker	3333333	Ahmetin Adresi	İstanbul	34
Ahmet	Şeker	3333333	Ahmetin Adresi	İstanbul	34
Mustafa	Kaya	2222222	Deneme Adresi	Ankara	06
Gökmen	İmamoğlu	5555555	Ahmetin Adresi	İzmir	35
Selim	Dener	4444444	Selmin Adresi	Ankara	06
Selim	Dener	4444444	Selmin Adresi	Ankara	06
Selim	Dener	4444444	Selmin Adresi	Ankara	06

Adı	Soyadı	AlınanÜrün	Miktar	Fiyat
Mustafa	Kaya	Disket	5	10
Ahmet	Şeker	CD-R	50	100
Ahmet	Şeker	DVD-R	10	50
Ahmet	Şeker	Mouse	2	2
Ahmet	Şeker	CD-R	25	50
Ahmet	Şeker	CD-R	50	100
Mustafa	Kaya	DVD-R	10	50
Gökmen	İmamoğlu	CD-R	50	100
Selim	Dener			
Selim	Dener			
Selim	Dener			

ikinci normal formda da satir ekleme ve silme sorunları ortaya çıkıyor.

### Satir ekleme sorunu:

Müşteri tablosuna bir il eklemek istediğinizde mantıken bir müşteri eklemek gerekir.

### Satir Silme Sorunu:

Müşteri silindiğinde (Örn: Gökmen İmamoğlu) müşterinin oturduğu şehir ve şehirle ilgili plaka verisi de silinebilir

- Bu sorunları Düzeltmek için Üçüncü Normal form uygulanır.
- Üçüncü Normal Form, İkinci Normal Form yapısındaki Tabloların Fonksiyonel bağımlılık içermese de alt tablolara bölünmesinden elde edilir.

33

# Üçüncü Normal Form



- Üçüncü normal formda tablonun, **Primary Key** ile direk ilişkisi bulunmayan, ancak diğer alanlara bağlı alanlar bulunur.
- Örneğin müşteri tablosunda şehir ismi ve plaka alanı kişi ile doğrudan bağlantısı yoktur. Bu alanların ayrı bir tabloya alınması üçüncü derece normalizasyondur.

34

## Üçüncü Normal Form



MusteriID	Adı	Soyadı	TelNo	Adres	ŞehirID
1	Mustafa	Kaya	2222222	Deneme Adresi	1
2	Ahmet	Şeker	3333333	Ahmet'in Adresi	2
3	Gökmen	İmamoğlu	5555555	Ahmet'in Adresi	3
4	Selin	Dener	4444444	Selin'in Adresi	1
(AutoNumber)					0

ŞehirID	Şehir	PlakaKodu
1	Ankara	06
2	İstanbul	34
3	İzmir	35
(AutoNumber)		

AliyeveceID	MusteriID	AlinanÜrün	Miktar	Fiyat
1	1	1 Disket	5	10
2	2	2 CD-R	50	100
3	3	2 DVD-R	10	50
4	2	2 Mouse	2	2
5	2	2 CD-R	25	50
6	2	2 CD-R	50	100
7	1	1 DVD-R	10	50
8	3	3 CD-R	50	100
(AutoNumber)	0		0	0

35

## Dördüncü Normal Form



- Tablolarda, Bos Geçilen Sütunlardan Kaçınılmalıdır.
- Verilerin yönetimi ve tutarlılığı açısından NULL geçilebilir alan sayısını az tutmaya çalışmak faydalı olacaktır

36

# Beşinci Normal Form



- Tablolar, Tekrarlayan satırlar veya Tekrarlayan Sütunlar içermemelidir.

Barkod	urunAdi	marka	Fiyat	reyon	reyon2
8000	SQL Server 2000	1	5000	1000000	<NULL>
8001	SQL Server 2000 ile	2	10	1000100	1000000
8002	Celeron 2500	3	150	1000200	<NULL>
8003	Athlon 3000+	4	160	1000200	<NULL>
8004	Celeron 3000	3	190	1000200	<NULL>
8005	ASP.NET	2	10	1000100	1000000

urun

reyonKod	ReyonAd
1000000	Bilgisayar
1000100	Kitap
1000200	İslemci

reyon

Barkod	urunAdi	marka	Fiyat
8000	SQL Server 2000	1	5000
8001	SQL Server 2000 ile	2	10
8002	Celeron 2500	3	150
8003	Athlon 3000+	4	160
8004	Celeron 3000	3	190
8005	ASP.NET	2	10

urun

barkod	ReyonKod
8000	1000000
8001	1000100
8005	1000100
8003	1000200
8002	1000200
8004	1000200
8001	1000000
8005	1000000

urunReyon

reyonKod	ReyonAd
1000000	Bilgisayar
1000100	Kitap
1000200	İslemci

reyon

37

# ÖRNEK (1NF)



## Çalışanlar (1NF)

isci_no	isim	Bölüm_no	Bölüm_adi	Bildiği Programlar
1	Kevin Jacobs	201	R&D	C
1	Kevin Jacobs	201	R&D	Perl
1	Kevin Jacobs	201	R&D	Java
2	Barbara Jones	224	IT	Linux
2	Barbara Jones	224	IT	Mac
3	Jake Rivera	201	R&D	DB2
3	Jake Rivera	201	R&D	Oracle
3	Jake Rivera	201	R&D	Java

38

## 2NF



Çalışanlar (1NF)				
isci_no	isim	Bölüm_no	Bölüm_adi	Bildiği Programlar
1	Kevin Jacobs	201	R&D	C
1	Kevin Jacobs	201	R&D	Perl
1	Kevin Jacobs	201	R&D	Java
2	Barbara Jones	224	IT	Linux
2	Barbara Jones	224	IT	Mac
3	Jake Rivera	201	R&D	DB2
3	Jake Rivera	201	R&D	Oracle
3	Jake Rivera	201	R&D	Java

Çalışanlar (2NF)			
isci_no	isim	Bölüm	Bölüm_adi
1	Kevin Jacobs	201	R&D
2	Barbara Jones	224	IT
3	Jake Rivera	201	R&D

Beceriler (2NF)	
isci_no	Bildiği Progra
1	C
1	Perl
1	Java
2	Linux
2	Mac
3	DB2
3	Oracle
3	Java

39

## 3NF



Çalışanlar(2NF)			
isci_no	isim	Bölüm	Bölüm_adi
1	Kevin Jacobs	201	R&D
2	Barbara Jones	224	IT
3	Jake Rivera	201	R&D

Çalışanlar(3NF)		
isci_no	isim	Bölüm_n
1	Kevin Jacobs	201
2	Barbara Jones	224
3	Jake Rivera	201

Bölüm (3NF)	
Bölüm_	Bölüm_adi
201	R&D
224	IT

40



# Örnek



## Normalize edilmemiş tablo

ÖğrenciNo Danışman Danış.Odası Ort1 Ort2 Ort3

-----  
1022 Ahmet 412 60 70 67  
4123 Derya 216 56 80 70

Ortalama alanında olduğu gibi tekrarlayan alanlar olmamalıdır.

## 1NF

Öğrenci No	Danışman	Danışman Odası	Ort
1022	Ahmet	412	60
1022	Ahmet	412	70
1022	Ahmet	412	67
4123	Derya	216	56
4123	Derya	216	80

41



## 2NF



- Tekrarlayan veriler tabloda yer almamalıdır.

ÖğrenciNo	Danışman	Danışman Odası
1022	Ahmet	412
4123	Derya	216

ÖğrenciNo	Ort
1022	60
1022	70
1022	67
4123	56
4123	80
4123	70

42

## 3NF

- Bir anahtara bağılı olmayan veri tablodan çıkartılmalıdır.
- Danışman odası alanı, Danışman alanına fonksiyonel bağımlıdır.
- İstenirse bu alan öğrenci tablosundan çıkartılabilir

Property Number	Property Address	Rent Start	Rent Finish	Rent	Owner Number	Owner Name
PG4	6 Lawrence St, Glasgow	1-Jul-94	31-Aug-96	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	1-Sep-96	1-Sep-98	450	CO93	Tony Shaw



Customer\_Rental Table

Customer_No	CName	Property_No	PAddress	RentStart	RentFinish	Rent	Owner_No	OName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-94	31-Aug-96	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-96	1-Sep-98	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-92	10-June-94	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-94	1-Dec-95	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Jan-96	10-Aug-96	450	CO93	Tony Shaw

1NF

Customer\_Rental Relation

Müş.No	İşlemNo	Müş.Adı	Adres	KiraBaşl.	KiraBitiş	Kira	SahipNo	SahipAdı
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-94	31-Aug-96	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-96	1-Sep-98	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-92	10-Jun-94	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-94	1-Dec-95	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Jan-96	10-Aug-96	450	CO93	Tony Shaw

45

2NF



Customer Relation

Müş.No	Müş.Adı
CR76	John Kay
CR56	Aline Stewart

Rental Relation

Müş.No	İşlemNo	KiraBaşl.	KiraBitiş
CR76	PG4	1-Jul-94	31-Aug-96
CR76	PG16	1-Sep-96	1-Sep-98
CR56	PG4	1-Sep-92	10-Jun-94
CR56	PG36	10-Oct-94	1-Dec-95
CR56	PG16	1-Jan-96	10-Aug-96

Property\_Owner Relation

İşlem No	Adres	Kira	SahipNo	SahipAdı
PG4	6 Lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93	Tony Shaw

46



## 3NF



Property\_for\_Rent Relation

İşlem No	Adres	Kira	SahipNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner Relation

SahipNo	SahipAdı
CO40	Tina Murphy
CO93	Tony Shaw

47



## 3NF(TÜM TABLOLAR)



Customer Relation

Müş.No	Müş.Adı
CR76	John Kay
CR56	Aline Stewart

Rental Relation

Müş.No	İşlemNo	KiraBaşl.	KiraBitiş
CR76	PG4	1-Jul-94	31-Aug-96
CR76	PG16	1-Sep-96	1-Sep-98
CR56	PG4	1-Sep-92	10-Jun-94
CR56	PG36	10-Oct-94	1-Dec-95
CR56	PG16	1-Jan-96	10-Aug-96

Property\_for\_Rent Relation

İşlem No	Adres	Kira	SahipNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner Relation

SahipNo	SahipAdı
CO40	Tina Murphy
CO93	Tony Shaw

48