

Veri Bütünlüğü
Veri Normalizasyonu
Veri Tabanı Tasarlama

VERİ BÜTÜNLÜĞÜNÜ

- *Veri bütünlüğü adından da çıkartabileceğimiz gibi verilerin eksiksiz ve fazlalıksız şekilde tutulması.*
- *Eksiksiz olması veri kaybına karşılık düşünülebilirken fazlalıksız olması silinen bir verinin ilişkili olan diğer kısımlarının gereksiz yere tablolarda bulunmaması şeklinde düşünülebilir.*
- *Kısaca veri bütünlüğü ne eksik(veri kaybı) ne fazla(gereksiz-çöp) veri bulunmamasıdır.*

- *Veri bütünlüğünün ilk adımı veri tabanı tasarımı sırasında belirlenen kurallar(kısıtlamalar ve yönlendirilmeler) ile atılır.*
- *Bir veri birden fazla yerde olmamalı yani aynı veri başka yerlerde tekrar etmemeli.*
- *Veri farklı şekillerde olmamalı.*
- *Veri bütünlüğünün sağlanmasındaki en önemli kısımlardan biri tablolar arasındaki birincil anahtarlar ile yapılan ilişkilendirilmenin doğru yapılması.*

Veri Bütünlüğünde Kısıtlamalar

- Veri girişleri sırasında bir takım kurallar(kısıtlama) ile verilerin doğru şekilde girilmesi sağlanmalıdır.
- Bunun için;
 - bir takım kontroller(check-harf karakter uzunluk veri tipi gibi e-posta adresinde @ işaretinin olması gibi)
 - Aynı veriden başka olmaması benzersizlik(unique- aynı kullanıcı adından iki tane olamaz)
 - Boş geçilemez alanların olması(Not null-bazı kritik bilgilerin mutlaka olması gibi)

İşte bu durumlara dikkat ederek veri bütünlüğü sağlanabilir

Tabloda Unique(Benzersiz) Alan Oluşturma

- *Tabloda bir alandaki değerlerin birbirinden farklı olması istendiğinde bu alan Unique olarak belirlenir.*

```
CREATE TABLE Musteri
(
    No INT PRIMARY KEY NOT NULL,
    Ad VARCHAR(100) NOT NULL,
    Soyad VARCHAR(100) NOT NULL,
    KullaniciAdi VARCHAR(20) NOT NULL UNIQUE
);
```

```
CREATE TABLE Musteri
(
    No INT PRIMARY KEY NOT NULL,
    Ad VARCHAR(100) NOT NULL,
    Soyad VARCHAR(100) NOT NULL,
    KullaniciAdi VARCHAR(20) NOT NULL UNIQUE
);
```

Messages
Commands completed successfully.

Hazırladığımız tabloya aynı KullanıcıAdı atamaya çalışalım

- `INSERT INTO Musteri VALUES('1', 'Ayşe', 'DAĞ', 'Ayşe');`

No	Ad	Soyad	KullanıcıAdı
1	Ayşe	DAĞ	Ayşe
NULL	NULL	NULL	NULL

- İlk olarak aynı sorguyu çalıştırdığımızda No alanı için primary key hatası verdi aynı olamaz diye

Messages
Msg 2627, Level 14, State 1, Line 224
Violation of PRIMARY KEY constraint 'PK__Musteri__3214D4A8AE72F56B'. Cannot insert the statement has been terminated.

Tekrar No alanı değiştirdiğimizde bu sefer kullanıcı adı kısmı için Unique key hatası veriyor

```
INSERT INTO Musteri VALUES('2', 'Ayşe', 'DAĞ', 'Ayşe');
```

119 %

Messages
Msg 2627, Level 14, State 1, Line 224
Violation of UNIQUE KEY constraint 'UQ__Musteri__5BAE6A755B9595C'. The statement has been terminated.

PRIMARY KEY

- *Tablolarda benzersiz olarak her kayda verilen koda Primary key diyorduk.*
- *Primary key'i tablo oluştururken de tanımlayabiliriz sonradan da tanımlama şansımız var.*
- *Birincil anahtar oluşturmanın iki yöntemi vardır.*
 - *SQL sorguları ile*
 - *Management Studio ile*

SQL Sorguları ile Primary Key Oluşturma

- *Burada tablo oluştururken No alanına primary key tanımladık.*
- *Primary key tanımladığımız sütun için genelde beraberinde Otomatik artan ve boş geçilemez ifadeleri eklenir.*

```
CREATE TABLE Musteri
(
    No INT IDENTITY PRIMARY KEY NOT NULL,
    Ad VARCHAR(100) NOT NULL,
    Soyad VARCHAR(100) NOT NULL,
    KullaniciAdi VARCHAR(20) NOT NULL UNIQUE
);
```

Var olan tabloya sonradan Primary Key Ekleme

- *Tabloyu oluşturduktan sonra bir alana primary key ilave etmek gerekebilir.*
- *Bu durumda ALTER ifadesi ile değişiklik yapabiliriz.*

```
ALTER TABLE Personel /*personel tablosunda değişiklik yap*/
```

```
ADD CONSTRAINT PersonelNoPk PRIMARY KEY(PersonelNo)/*Ekle yeni bir kısıtlama ekle,  
kısıtlamaya isim verip bu değişikliğin yapılacağı sütun*/
```

Management Studio ile Primary Key Oluřturma

Tabloya Kontrol(Check) Kısıtlamaları(Constraint) Ekleme

- *Mesela şifre parola alanına girilecek olan karakterin sayısının en az 7 karakter en fazla 9 karakter olması istiyorsak;*

```
ALTER TABLE Personel
```

```
ADD CONSTRAINT CkSifreUzunluk CHECK(LEN(Sifre)>6 AND (Sifre)<10)
```

- *E-posta alanında @ işaretinin olup olmadığının kontrolü*

```
ALTER TABLE Personel
```

```
ADD CONSTRAINT CkSifreUzunluk CHECK(CHARINDEX('@',Eposta)>0 OR Eposta IS NULL
```

- *E posta alanında @ işaretinin sayısı 0 dan büyük olmalı ve boş geçilmemeli.*

Default Kısıtlama Ekleme

- Kullanıcıya bırakmadan tasarı sırasında eklenen kısıtlamalardır.
- Bu alanlara değer girişi yapılmaz her kayıt eklendiğinde sistem direk olarak değer atar.
- Mesela her kayıt sırasında tarih bilgisinin eklenmesi;

KayıtTarihi DATETIME NOT NULL DEFAULT GETDATE()

CREATE TABLE Musteri

(

No INT IDENTITY PRIMARY KEY NOT NULL,

Ad VARCHAR(100) NOT NULL,

Soyad VARCHAR(100) NOT NULL,

KullaniciAdi VARCHAR(20) NOT NULL UNIQUE

KayıtTarihi DATETIME NOT NULL DEFAULT GETDATE()

);

Veri Tabanı Tasarlama

- *İlk olarak veri tabanındaki öncelikler belirlenmeli;*
 - *Güvenlik, Hız-Performans, veri güvenliği mi önemli?*
 - *Bu başlıklar birbirini etkileyen faktörler veri güvenliği önemli diyorsak kontrolleri çok tutacağız buda performansı etkileyecektir*

- *Veri tabanının birlikte çalışacağı uygulama(yazılım) tarafı nedir?*
 - *Bu yazılım kısmında kullanılacak dil, sağlayacak olduğu kontroller ve hizmetler neler olacak.*
 - *Bunları bilmek iş bölümü anlamında kolaylık sağlar gereksiz kısıtlamaları ve uygulamaların önüne geçer. Yazılım tarafında kısıtlama ve kontrol koymak tüm sistemin performansı için daha iyidir.*

Yazılım Geliştirme Süreci

- *Microsoft temelli yazılım geliştirme süreci veri tabanı tasarımını üç safha halinde ele alır*
 - *Kavramsal Model*
 - *Mantıksal Model*
 - *Fiziksel Model*

Kavramsal Model

- *Projenin uygulanacağı sahadaki kişilerle birebir temasa geçip istekleri, gereksinimler, işleyişleri(iş süreçleri) hakkında analizler yapılır.*
- *Veritabanından ziyade kullanıcının dili ile ifade edilmiş gereksinimler(ihtiyaçlar) ortaya konur.*
- *Daha önceden kullanılan bir veritabanı varsa gözden geçirilir.*

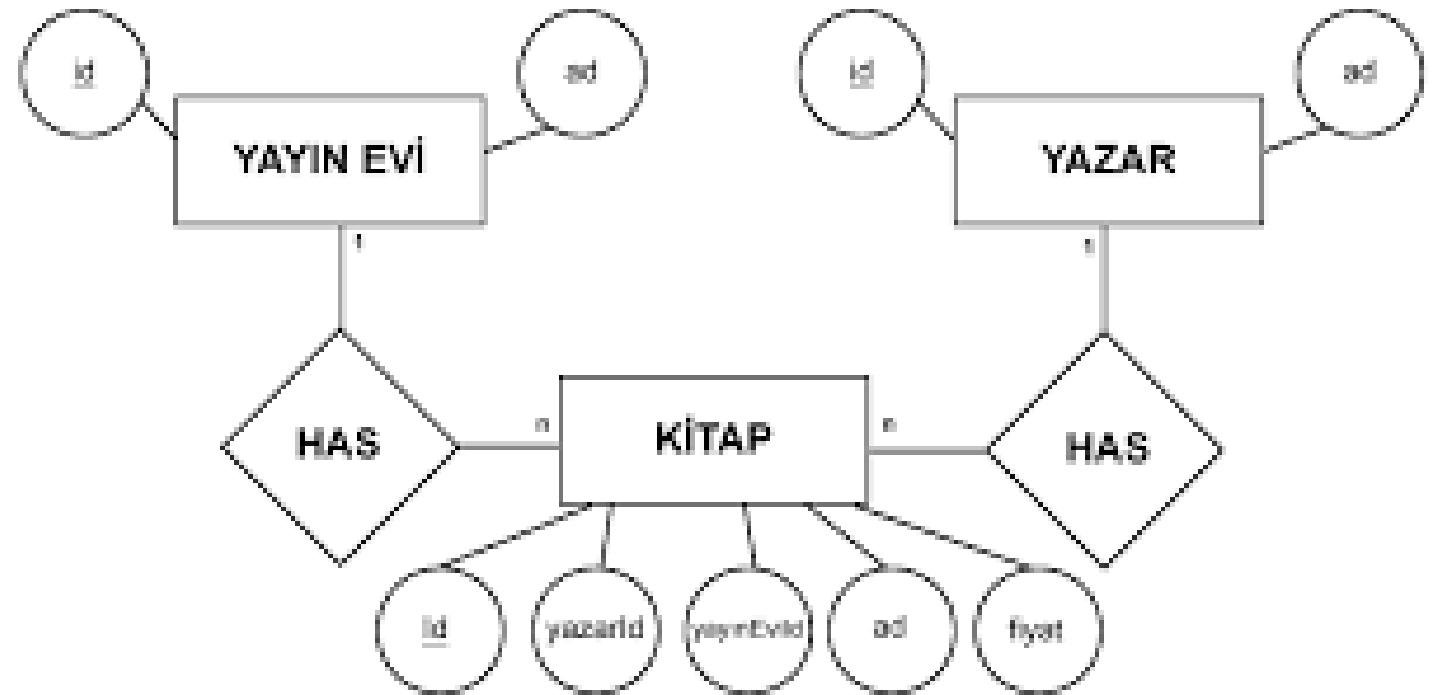
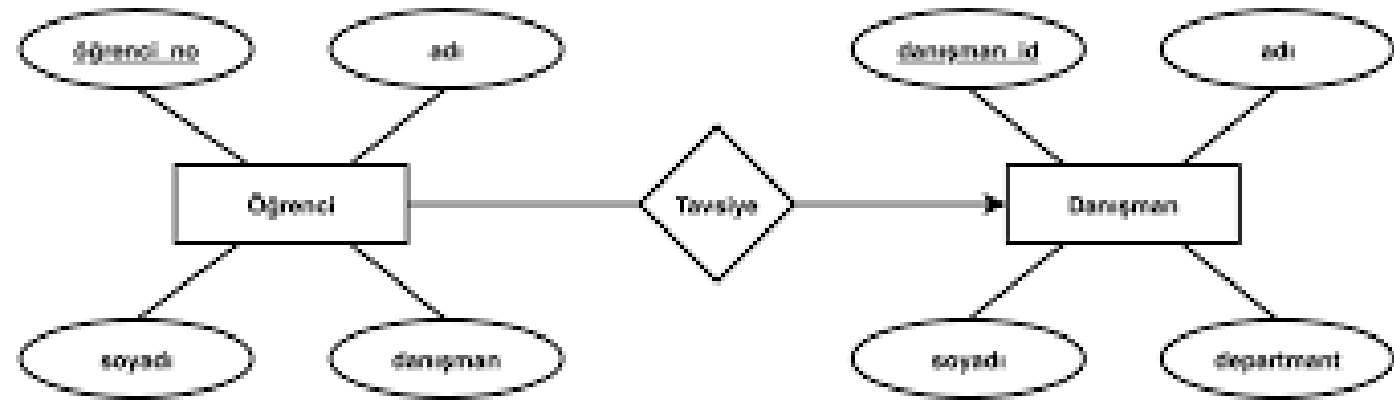
- *Bu seviyedeki bütün model, veri tabanından anlamayan, sadece işine odaklanmış şekilde düşünen işverenin bakış açısıyla ifade edilir.*
- *Her hangi bir teknik dil ve söylem yoktur sadece ihtiyaç tespiti ve işin işleyişi anlaşılmaya çalışılır.*

Mantıksal Model

- *Analizler sonucu elde edilen kavramsal modelin , uygulama geliştirenlerin bakış açısına veri tabanı tekniğine ve bilimsel kurallara uygun olarak yeniden düzenlenerek oluşturulan veri tabanı projesidir.*
- *Basit ve orta seviyeli veri tabanlarında kağıt ve kalem ile mantıksal model oluşturulabilirken büyük boyutlu veri tabanlarında yardımcı yazılımlardan destek alınmalı.*
- *Mantıksal model oluşturulurken en çok kullanılan modelleme yöntemlerinden biri olan ER(Entitit Relationship) diyagramı kullanılır*

ER ile Modelleme Aşamaları

- *Gereksinimleri göz önüne alarak varlıkları (tablolar vb), nitelikleri ve ilişkileri belirlenir.*
- *Bağılılıkları ve kısıtlayıcıları belirleyin*
- *Genel ve özel bilgi tiplerini belirlenir.*
- *Kavramsal bir ER diyagramı oluşturulur.*



Fiziksel Model

- *Bu aşamada, teorilere dayanarak tasarladığımız veri tabanını gerçek ortamda oluşturuyoruz.*
- *Bu aşamada çeşitli veri tabanı yönetim sistemi programlarından yararlanabiliriz.*
 - *Ms SQL*
 - *My Swl*
 - *Oracle*
 - *Sybase*
 - *Postrage*

VERİ NORMALİZASYONU

- *Verileri normalize etmek, birden fazla nesneye ait veriyi orta bir yol bularak, bir tabloda saklanabilir hale getirmek demektir.*
- *Normalizasyon kurallarımın çeşitli seviyeleri vardır. İlk üç kural bir çok problemin çözümü için yeterlidir.*
- **Veri normalizasyonun amacı;**
 - *Tekrarları en aza indirmek ☐*
 - *Veri tutarlılığını sağlamak*
 - *Yönetimi merkezileştirmek = kolaylaştırmak*

Normalizasyon Kurallarına Uymayan Tasarımlarda Ortaya Çıkabilecek Sorunlar

- ***Tekrarlı Veriler:*** Bazı bilgiler birden fazla tekrarlar.
- ***Güncelleme Sıkıntıları:*** Bir bilgiyi güncellemek için birden fazla yerde bulma ve değiştirme yapmak gerekir.
- ***Veri Ekleme Problemleri:*** Bir veriyi eklemek için birden fazla tabloya veri eklemek gerekir.
- ***Silme Problemleri:*** Bir veriyi silmek için birden fazla tabloda bulup silmek gerekir.

Normalizasyon Kurallarına Uymayan Tasarımlarda Ortaya Çıkabilecek Sorunlar-Özet

- *Normalizasyon kuralları tasarım aşamasında bir veriyi sadece bir yerde tutarak veri girişi, güncelleme ve silme işlemlerinde tek bir noktadan güvenli, hızlı ve tutarlı olmasını sağlar.*
- *Özet olarak normalizasyon kurallarına uymayan tasarımlarda veri tabanı kullanılmaz hale gelebilir ve yeni bir veri tabanı tasarımı ihtiyacı doğabilir.*

Veri Normalizasyonu-1

- Her sütunda bir varlığa ait bir özellik saklanmalıdır.

| Örnek: Ad ve soyad ayrı alanlarda

	Musteri	Telefon
	Ahmet ERYATAN	3121233231
	Ali FINDIK	3121233231
	Mehmet ERYATAN	3121233231
	Musa ER	3121233231
►		



	MusteriAd	musteriSoyad	Telefon
	Ahmet	ERYATAN	3121233231
	Ali	FINDIK	3121233231
	Mehmet	ERYATAN	3121233231
	Musa	ER	3121233231
►			

Bilgiler tekrarlayan gruptan oluşmamalıdır. Örneğin bir alan içerisindeki bilgi özel karakterler ile ayrılarak tutulmamalıdır

Öğr.Gör. Erkut TEKELİ

Ogr_No	BolumKod	BolumAd	Ders_Kodu	Sinav
2012688001	1	Bilgisayar	BPR201, BPR202, BPR203	75, 85, 45
2012690005	2	Muhasebe	MVU202, MVU204, MVU206	25, 60, 55

Tablo 4. Ders_Kodu ve Sinav sütunları birden fazla değer içermektedir.

Ogr_No	BolumKod	BolumAd	Ders_Kodu	Sinav
2012688001	1	Bilgisayar	BPR201	75
2012688001	1	Bilgisayar	BPR202	85
2012688001	1	Bilgisayar	BPR203	45
2012690005	2	Muhasebe	MVU202	25
2012690005	2	Muhasebe	MVU204	60
2012690005	2	Muhasebe	MVU206	55

Tablo 5. Tablo 4'te verilen tablonun normalize edilmiş hali

Veri Normalizasyonu-2

- Her tablonun bir birincil anahtarı olmalıdır.

	MusteriAd	musteriSoyad	Telefon
	Ahmet	ERYATAN	3121233231
	Ali	FINDIK	3121233231
	Mehmet	ERYATAN	3121233231
	Musa	ER	3121233231
	Ali	FINDIK	3129999999
►			



	musteriKod	MusteriAd	musteriSoyad	Telefon
	100	Ahmet	ERYATAN	3121233231
	101	Ali	FINDIK	3121233231
	102	Mehmet	ERYATAN	3121233231
	103	Musa	ER	3121233231
	104	Ali	FINDIK	3129999999
►				

Veri Normalizasyonu-3

- Her Bir Tablo, Bir Varlığa Ait Verileri Tutmalıdır

urun

Barkod	urunAdi	marka	Fiyat
8000	SQL Server 2000	Microsoft	5000
8001	SQL Server 2000 ile Veritabani Programlama	Seçkin	10
8002	Celeron 2500	Intel	150
8003	Athlon 3000+	AMD	160
8004	Celeron 3000	Intel	190
8005	ASP.NET	Seçkin	10

3.NF

urun

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

1.NF

marka

markaKod	marka
1	Microsoft
2	Seçkin
3	Intel
4	AMD

Veri Normalizasyonu-4

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

Veri Normalizasyonu-5

- Tablolar, Tekrarlayan satırlar veya 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	İşlemci

reyon



urun

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

urunReyon

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

reyon

reyonKod	ReyonAd
1000000	Bilgisayar
1000100	Kitap
1000200	İşlemci

Veri tabanı Tasarlama

- 1. Nesneler tanımlanır*
- 2. Her nesne için bir tablo oluşturulur.*
- 3. Her tablo için bir anahtar alan seçilir*
- 4. Nesnelerin her bir özelliği için tabloya sütun eklenir*
- 5. Tekrarlayan nesne özellikleri için ek tablolar oluşturulur.*
- 6. Tablo ile doğrudan ilişkili olamayan alanlar belirlenir.*
- 7. Tablolar arasındaki ilişkiler tanımlanmalıdır.*

Veri tabanı Tasarlama

1. Nesneler tanımlanır

Kütüphane sistemi: kitap, üyeler, türler, ödünç hareketleri

Veri tabanı Tasarlama

2. Her nesne için bir tablo oluşturulur.

- *kitap,*
- *uyeler,*
- *turler,*
- *odunc_hareketleri*

Veri tabanı Tasarlama

3. Her tablo için bir anahtar alan(primarykey) seçilir

Kitap tablosu: kitapno

Üyeler tablosu: uyeno

Veri tabanı Tasarlama

4. Nesnelerin her bir özelliği için tabloya sütun eklenir

Kitap tablosu: kitapno, yılı, yazarı, adı, ilgili olduğu alan

Üyeler tablosu: uyeno,adı, soyadı,doğumtarihi,üyelik

tarihi,adres,telefon

Veri tabanı Tasarlama

5. Tekrarlayan nesne özellikleri için ek tablolar oluşturulur.

Ödünç tablosu

İstek tablosu

uyeno	İstek _tarihi	Kitap_adi	Kitap_yili	Kitap_yazari	ilgili _alan

Veri tabanı Tasarlama

6. Tablo ile doğrudan ilişkili olmayan alanlar belirlenir.

Ödünç hareketleri tablosunda kitabı ödünç alan üyenin adresi doğrudan bu tablo ile ilişkili değildir.

Bu veri üye bilgilerinin tutulduğu uyeler tablosunda yer almalıdır.

Veri tekrarını önlemek için bu önemlidir.

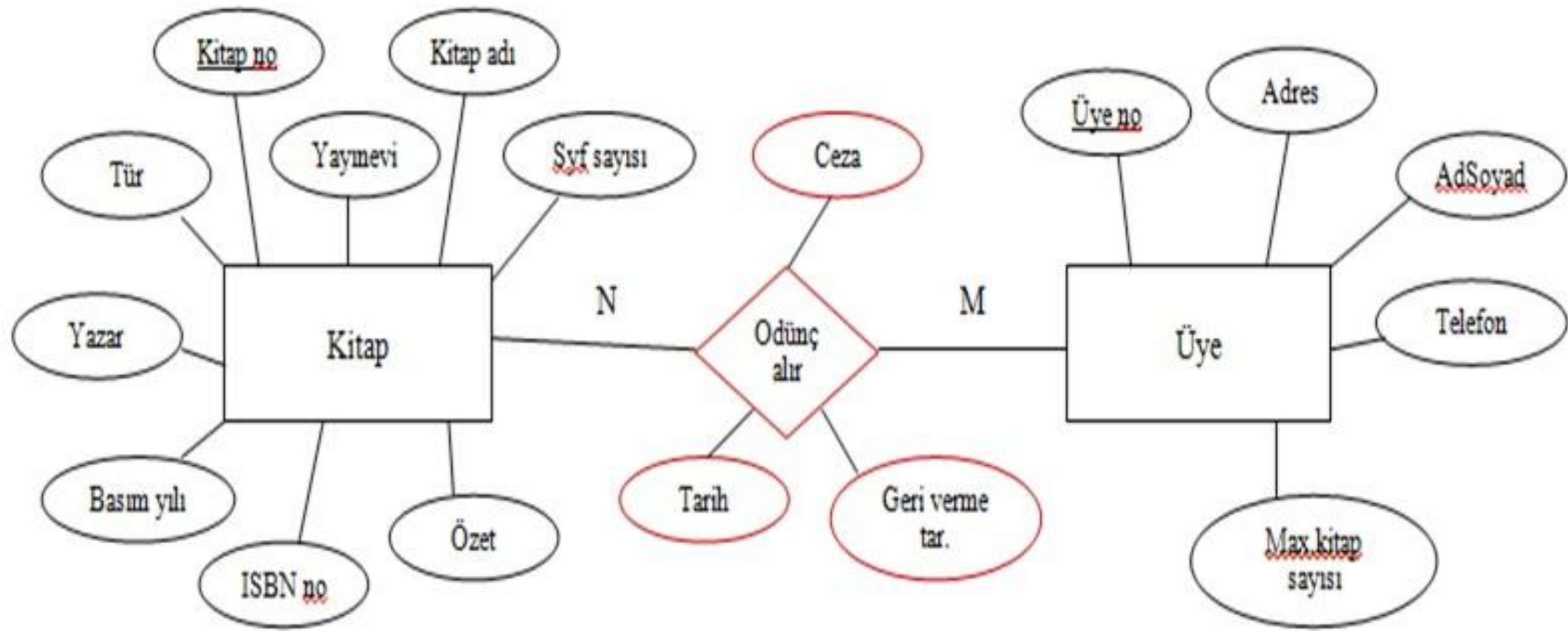
Veri tabanı Tasarlama

7. Tablolar arasındaki ilişkiler tanımlanmalıdır.

Tanımlanan tablodaki alanların birbiri ile ilişkisi tanımlanır.

*Örneğin uyerler tablosundaki uyeno ile odunç_tablosundaki uyeno alanı
ilişkilendirilmelidir.*

KÜTÜPHANE BİLGİ SİSTEMİ



Kaynaklar

- <http://www.ikslab.com/veritabani-kavramsal-tasarimi>
- Veritabanı Dersi Ders Notları – 4 Öğr.Gör. Erkut TEKELİ