

# 3. İLİŞKİSEL VERİ TABANI YÖNETİM SİSTEMİ

## Birlikte Düşünelim

1. İlişkisel veri modeli nedir?
2. İlişkisel veri tabanı özellikleri nelerdir?

## Başlamadan Önce

Önceki bölümde varlık-ilişki modeli konusu ele alınmıştı. Daha evvel değinildiği üzere günümüzde veriler ve bunları kullanan yazılımlar çeşitlendiği gibi, yazılımları kullanan Veri Tabanı Yönetim Sistemleri de çeşitlenmiştir. Veri Tabanı Yönetim Sistemleri arasındaki en önemli ayrımlardan biri ilişkisel Veri Tabanı Yönetim Sistemleri olup olmamalarıdır. İlişkisel bir Veri Tabanı Yönetim Sistemi mi yoksa ilişkisel olmayan bir Veri Tabanı Yönetim Sistemi mi kullanmaya karar verebilmek için bu iki çeşit Veri Tabanı Yönetim Sistemi'nin ayrıntılarını bilmek gerekmektedir. Bu bölümde bu iki tür Veri Tabanı Yönetim Sistemi yapısının ayrıntıları açıklanacaktır.

## 3.1. İlişkisel Veri Tabanı Yönetim Sistemleri

Daha önceki bölümlerde de ifade edildiği gibi yazılımlardaki verileri modellerken farklı model yapıları kullanılmaktadır. İlişkisel model, günümüzde en yaygın kullanılan modeldir. İlişkisel model, varlıklar arasındaki bağlantının içerdiği değerlere göre sağlanması esasına dayanır ve varlıklar arasında oluşan karmaşık ilişkileri basite indirgemek amacıyla geliştirilmiştir. Bu yaklaşımda, Veri Tabanı Yönetim Sistemi'ndeki tüm ilişkiler tablolar biçiminde belirtilmektedir. Ticarî Veri Tabanı Yönetim Sistemleri'nin neredeyse tümünde bu model kullanılmaktadır. Şekil 3.1'de bir fabrikadaki durumu gösteren Fabrika isimli bir Veri Tabanı Yönetim Sistemi'nde bulunan ve personellerin özelliklerini gösteren PERSONEL isimli tablo ile departmanların özelliklerini gösteren DEPARTMAN isimli tablo arasındaki ilişki gösterilmiştir.

**PERSONEL Tablosu**

Departman No	Personel No	Adı	Soyadı

**DEPARTMAN Tablosu**

Departman No	Departman Adı

**Şekil 3.1.** Veri Tabanı Yönetim Sistemi İlişkileri

Şekil 3.1’den de anlaşıldığı gibi hem PERSONEL Tablosu hem de DEPARTMAN Tablosu Departman No isimli bir alan içermektedir. Bu alan, DEPARTMAN Tablosu’ndaki departman isimlerine karşılık gelen özgün numarayı gösterdiğinden bu iki tablo arasında kolaylıkla bir ilişki kurulmasını sağlar. İlişkisel Veri Tabanı Yönetim Sistemi ise, her biri özel isimlere sahip tablolardan oluşan Veri Tabanı Yönetim Sistemi yapısıdır. Burada her tablo bir varlığa veya bir ilişkiye karşılık gelmektedir. Tablonun sütunları, nitelikleri; satırlar ise bu niteliklerin değerlerini (sayısal, metinsel vb.) ifade eder. Her bir satır bir “kayıt”, anahtar alan ise tablonun tanımlayıcısı olarak düşünülebilir. Şekil 3.2’de bir Veri Tabanı Yönetim Sistemi tablosunun satırları (kayıt) ve sütunları (alan) gösterilmektedir.

Sütunlar (Alanlar)				Satırlar (Kayıtlar)
Departman No	Personel No	Adı	Soyadı	
10	13	Rüya	Şamlı	
10	23	Eylem	Yüce	
20	35	Selçuk	Sevilen	

**Şekil 3.2.** Tablonun Satırları ve Sütunları

İlişkisel Veri Tabanı Yönetim Sistemi içinde yer alan her tablo belirtilen özelliklere sahiptir.

- Tablolar sütunlardan oluşur.
- Her bir sütunun ayrı, özgün bir adı vardır.
- Her bir sütun, aynı niteliğin tanımladığı aynı etki alanının belirlediği değerleri içerir. Bu değerler satırlar halindedir.
- Her bir satır birbirinden farklıdır.

- Sütunların sırası önemsizdir.
- Satırların sırası önemsizdir.

Örneğin yukarıdaki PERSONEL Tablosu’nda yer alan Departman No, Personel No, Adı ve Soyadı alanlarının yerleri değiştirildiğinde de tablonun özellikleri ve diğer tablolarla olan ilişkileri değişmez. Şekil 3.3’te Personel Tablosu’nun sütunlarının yer değiştirilmesiyle oluşan yeni tablo görülmektedir. Bu tablo yalnızca görsel olarak farklıdır, içerik ve özellik olarak bir farklılığı yoktur.

Personel No	Soyadı	Adı	Departman No
13	Şamlı	Rüya	10
23	Yüce	Eylem	10
35	Sevilen	Selçuk	20

**Şekil 3.3.** PERSONEL Tablosu’nda Sütunların Yer Değiştirilmesi

Yine aynı şekilde yukarıdaki PERSONEL Tablosu’nda yer alan kayıtların yerleri değiştirildiğinde de tablonun özellikleri ve diğer tablolarla olan ilişkileri değişmez. Şekil 3.4’te PERSONEL Tablosu’nun satırlarının yer değiştirilmesiyle oluşan yeni tablo görülmektedir. Bu tablo yalnızca görsel olarak farklıdır, içerik ve özellik olarak bir farklılığı yoktur.

Departman No	Personel No	Adı	Soyadı
10	23	Eylem	Yüce
10	13	Rüya	Şamlı
20	35	Selçuk	Sevilen

**Şekil 3.4.** PERSONEL Tablosu’nda Sütunların Yer Değiştirilmesi

Buna göre Şekil 3.5’te verilen tüm tablolar temelde aynı niteliği ve içeriği taşıyan PERSONEL Tablosu’nun görsel olarak değişik versiyonlarıdır.

Departman No	Personel No	Adı	Soyadı
10	13	Rüya	Şamlı
10	23	Eylem	Yüce
20	35	Selçuk	Sevilen

Personel No	Soyadı	Adı	Departman No
13	Şamlı	Rüya	10
23	Yüce	Eylem	10
35	Sevilen	Selçuk	20

Departman No	Personel No	Adı	Soyadı
10	23	Eylem	Yüce
10	13	Rüya	Şamlı
20	35	Selçuk	Sevilen

**Şekil 3.5.** PERSONEL Tablosu’nun Farklı Versiyonları

Veri Tabanı Yönetim Sistemi’nin mantıksal tasarımı “veri tabanı şeması” olarak isimlendirilir. Tablolar ve nitelikleri Veri Tabanı Yönetim Sistemi şemasını oluşturur.

Veri Tabanı Yönetim Sistemi şemaları Fiziksel Şema ve Kavramsal Şema olarak gruplandırılabilir. Fiziksel Şema, Veri Tabanı Yönetim Sistemi’nin fizikî çevresi ile ilgili tanımları içerir. Örneğin, Veri

Tabanı Yönetim Sistemi'nin bilgisayar belleğindeki karşılığı olan disk dosyasının disk üzerindeki adresi ve özellikleri ile ilgili tanımlar fiziksel şemayı oluşturur. Kavramsal şema ise, tüm Veri Tabanı Yönetim Sistemi'nin mantıksal tasarımı olarak ifade edilebilir ve Veri Tabanı Yönetim Sistemi'nde olmasına karar verilen veriler arasındaki mantıksal ilişkiler yapısının saptanması için oluşturulur. Bu şemada, Veri Tabanı Yönetim Sistemi'nde veri alanları kayıtlar dosyalar vb. gibi ne tür veri elemanlarının bulunacağı, veri elemanları arasındaki ilişkiler ve Veri Tabanı Yönetim Sistemi'nin yapısı hakkında bilgiler yer alır. Önünde bir sıfat kullanılmadan, direkt Veri Tabanı Yönetim Sistemi şeması dendiğinde kavramsal şema anlaşılmaktadır. Kavramsal şema tasarlandıktan sonra her bir uygulama için alt-şemalar hazırlanır. Örneğin bir üniversite Veri Tabanı Yönetim Sistemi için öğrenci işleri uygulaması bir alt şemadır. Çünkü bu uygulama Veri Tabanı Yönetim Sistemi'nin tümü ile ilgilenmez. Alt şema, anlaşıldığı üzere Veri Tabanı Yönetim Sistemi şemasının herhangi bir uygulama programı tarafından gereksinim duyulan alt-düzye mantıksal görünümüdür. Uygulama programlarının Veri Tabanı Yönetim Sistemi'nin her noktasına erişmesi gerekmez. Çoğunlukla kendisi ile ilgili bölümlere yani alt-şemaya ulaşması yeterlidir. Bu uygulamalar Veri Tabanı Yönetim Sistemi'nin diğer kaynakları ile ilgilenmez.

Veri Tabanı Yönetim Sistemi'nin herhangi bir andaki durumuna “Veri Tabanı Yönetim Sistemi Örneği” adı verilir. Veri Tabanı Yönetim Sistemi'nin doğru ve tutarlı bir biçimde çalışması ve işlemleri yerine getirmesi gerekir. Verinin doğru ve tutarlı olmasına veri bütünlüğü adı verilmektedir. Veri bütünlüğünün sağlanması sonucunda, Veri Tabanı Yönetim Sistemi'ni eksik, yanlış, tutarsız ve çelişkili olmaması sağlanır. Bu da Veri Tabanı Yönetim Sistemi'nin hizmet ettiği yazılımın da başarılı olması anlamına gelmektedir. Veri Tabanı Yönetim Sistemi'nde veri bütünlüğünü sağlamakta kullanılan pek çok yol vardır ve en önemlisi, “bütünlük sınırlamaları” olarak adlandırılır. Bütünlük sınırlamaları Veri Tabanı Yönetim Sistemi üzerinde gerçekleştirilen çeşitli sınırlamalardır ve sistem veya uygulama programları tarafından tanımlanmaktadır. Bu sınırlamalar, kullanıcı tarafından yapılması gereken ekleme, silme ve güncelleştirme işlemlerinden önce bir denetim gerçekleştirilmesini sağlar. Denetim sonucunda, sınırlamalara uyan işlemlerin yapılmasına izin verilirken diğerleri reddedilir bu da gerçekleştirilen işlemlerin belirli bir kritere uyması anlamına gelmektedir. Veri Tabanı Yönetim Sistemi sınırlamaları temel olarak aşağıdaki gibi ifade edilebilir:

- Anahtar Sınırlamaları
- Veri Değerleri Sınırlamaları
- İşletme Kuralları Sınırlaması
- Nitelikler Arası Bağımlılıklar

**Anahtar Sınırlamaları:** Bütünlük sınırlamalarının sağlanmasında ciddi etkisi olan anahtarların türü belirlenerek, bu sınırlamaların Veri Tabanı Yönetim Sistemi tarafından otomatik yapılması sağlanır. Bu sınırlamalar, birincil anahtar (primary key) veya dış anahtar (foreign key) sınırlamaları şeklinde gruplandırılabilir.

**Birincil Anahtar Sınırlamaları:** İlişkisel Veri Tabanı Yönetim Sistemi'nde bir tablonun benzer değerler içermeyen, özgün bir sütunu ya da birkaç sütunu birlikte “birincil anahtar” olarak tanımlanabilir. Birincil anahtar, aslında bir aday anahtardır ve söz konusu varlığın kayıtlarını en iyi biçimde karakterize eder. Birincil anahtar tanımlandığında, bu anahtarın NULL ya da birbirinin aynı değerleri içermemesi gibi bir sınırlandırma konulmuş olur. Şekil 3.6'da Personel Tablosu için bir birincil anahtar oluşturulmuş ve gösterilmiştir.

## PERSONEL Tablosu

Departman No	Personel No	Adı	Soyadı
10	13	Rüya	Şamlı
10	23	Eylem	Yüce
20	35	Selçuk	Sevilen

**Şekil 3.6.** PERSONEL Tablosu İçin Birincil Anahtar

Buradaki gibi PERSONEL Tablosu’nda her bir personelin bağlı olduğu departmanı belirten bir numara, personele verilmiş olan bir numara, personelin adı ve personelin soyadı alanları bulunmaktadır. Personele verilen numara bireysel olacağı ve tekrar kullanılmayacağı için birincil anahtar Personel No olarak seçilmiştir. Diğer alanlar birincil anahtar olarak uygun değildir. Zira bir departmanda birçok personel bulunup hepsi için Departman No aynı değere sahip olacaktır. Nitekim burada da Rüya Şamlı ile Eylem Yüce aynı departman numarasına sahiptir. Bu tabloda örneği verilmemekle beraber birçok personelin adının, soyadının hatta

ad – soyad birleşiminin aynı değere sahip olması da olasıdır. Bu yüzden Adı, Soyadı alanları birincil anahtar olarak kullanılamaz. Burada şunu belirtmek gerekir ki, veri tabanlarından anahtar seçilirken sayısal değerlerden faydalanmak daha uygundur, çünkü genellikle sayısal değerleri işlemesi daha kolaydır ve sayısal işlemlerin özgünlük ihtimali daha yüksektir.

Bu tabloya değerleri sırasıyla Departman No: 25, Personel No: 13, Adı: Sibel, Soyadı: Şenkan olan bir kayıt eklenmek istendiğinde, Veri Tabanı Yönetim Sistemi ilk olarak birincil anahtar değerini kontrol edecek ve Sibel Şenkan’ın birincil anahtar değeri olan 13, Rüya Şamlı’nın birincil anahtar değeri olan 13 ile aynı değere sahip olduğundan bu kaydın eklenmesine izin vermeyecektir. Birincil anahtar tek bir sütundan oluşabileceği gibi, birden fazla sütunun birleşiminden de oluşabilir. Bu durumda “birleşik birincil anahtar”dan söz edilir. Birleşik birincil anahtarın içerdiği değer “tek” olmalıdır. Ancak birleşik anahtarı oluşturan sütunların her biri çift değerler içerebilir. Bununla birlikte birincil anahtarı oluşturan sütunların hiçbirisi yine aynı şekilde NULL değeri içeremez. Şekil 3.7’de PERSONEL Tablosu için bir birleşik anahtar örneği görülmektedir.

### PERSONEL Tablosu

Departman No	Personel No	Adı	Soyadı
10	13	Rüya	Şamlı
10	23	Eylem	Yüce
20	35	Selçuk	Sevilen

**Şekil 3.7.** PERSONEL Tablosu İçin Birleşik Birincil Anahtar

Şekilde görülen birincil anahtarda Departman ve Personel No bir arada alınmıştır. Departman No, birçok kez değerde olacağı için tek başına bir birincil anahtar olamaz ancak tek başına da bir birincil anahtar özelliğine sahip Personel No alanını kuvvetlendirerek onunla beraber bir anahtar oluşturabilir.

**Dış Anahtar Sınırlamaları:** Bir dış anahtar, tek bir sütundan veya çok sayıdaki sütunların birleşiminden oluşur. Dış anahtar, aynı tablo ya da bir başka tabloda yer alan birincil anahtarla eşleştirilir. Dış anahtarın değeri bir birincil anahtar değeriyle eşleşmeli veya NULL olmalıdır. Eğer dış anahtar birincil anahtarın bir kısmını oluşturuyorsa, doğal olarak NULL değer içeremez. Şekil 3.8’de PERSONEL ve DEPARTMAN tabloları için bir dış anahtar örneği gösterilmiştir. Burada anahtar alan Departman No alanıdır. İki tablo birbirleriyle dış anahtar kullanılarak ilişkilendirildikten sonra, bu sınırlamalara dayanarak, iki tablo arasında silme ve güncelleştirme işlemleri otomatik olarak yerine getirilebilir. Örneğin PERSONEL ve DEPARTMAN isimli iki tabloyu incelendiğinde Personel tablosunun “Departman No” isimli sütunu, DEPARTMAN Tablosu’na ilişkin bir dış anahtar olarak tanımlandığı görülebilir. Bu tür bir tanım beraberinde ilişkisel bütünlük özelliklerini getirir. DEPARTMAN isimli tablodan bir satırı, örneğin “10” numaralı bölümü silmek istendiğinde, bu satır silindiğinde, PERSONEL Tablosu’nda da aynı satır ile ilgili

tüm kayıtlar otomatik olarak yok olacaktır. Çünkü bu durumda Departman No'su 10 olan herhangi bir kaydın bulunmaması gerekmektedir. Şekil 3.9'da bu durum görsel olarak açıklanmaya çalışılmıştır.

**PERSONEL Tablosu**

Departman No	Personel No	Adı	Soyadı
10	13	Rüya	Şamlı
10	23	Eylem	Yüce
20	35	Selçuk	Sevilen

**DEPARTMAN Tablosu**

Departman No	Departman Adı
10	Muhasebe
20	Satış
25	Üretim

**Şekil 3.8. Dış Anahtar Sınırlaması**

**DEPARTMAN Tablosu**

Departman No	Departman Adı
10	Muhasebe
20	Satış
25	Üretim

**PERSONEL Tablosu**

Departman No	Personel No	Adı	Soyadı
10	13	Rüya	Şamlı
10	23	Eylem	Yüce
20	35	Selçuk	Sevilen

**Şekil 3.9.** Departman Tablosundan Bir Kayıt Silindiğinde, Personel Tablosu'nda Ona Bağlı Tüm Kayıtlar Silinir

Tabloların bu silme işleminden sonraki hali Şekil 3.10'da verilmiştir.

**DEPARTMAN Tablosu**

Departman No	Departman Adı
20	Satış
25	Üretim

**PERSONEL Tablosu**

Departman No	Personel No	Adı	Soyadı
20	35	Selçuk	Sevilen

**Şekil 3.10.** Silme İşleminden Sonra Tabloların Durumları

**Veri Değerleri Sınırlamaları:** Tablonun herhangi bir sütununda yer alacak verilerin hangi değerlere sahip olabileceği konusunda sınırlamalar getirebilir. Eğer herhangi bir tabloda boş olması istenmeyen bir

alan varsa bu alan giriři ile ilgili NULL olmaması gerektiđi belirtilebilir. Bu duruma verilebilecek diđer örneklerden bazıları ařađıda belirtilmiřtir:

- řehir ii telefon numarasını ieren bir alana girilecek deđerin 7 haneli olması, sadece rakamlardan oluřması ve bařında 0 olmaması,
- řehirler arası telefon numarasını ieren bir alana girilecek deđerin 11 haneli olması, sadece rakamlardan oluřması ve bařında 0 olması,
- e-mail adreslerini ieren bir alana girilecek deđerin mutlaka @ iřaretini ieren bir metin olması ancak bu @ iřaretinin ilk karakter olmaması,
- T.C. Kimlik No ieren bir alana girilecek deđerin 11 haneli olması, sadece rakamlardan oluřması, son hanesinin mutlaka ift bir sayı olması,
- đrenci numaralarını ieren bir alana girilecek deđerin o üniversite, fakólte ve bölüme verilen đrenci numaraları ile aynı formata sahip olması,
- đrenci notlarını ieren bir alana girilecek deđerin 0-100 aralıđında sayısal bir deđer olması,
- tarih ieren bir alanın hangi tarih formatı kullanılıyorsa ona uygun olması, örneđin Gün–Ay–Yıl řeklinde kullanılıyorsa Gün deđerinin 1-31; Ay deđerinin ise 1-12 arasında olması,
- Türke metin ieren herhangi bir alana girilecek deđerin Ğ ile bařlamaması, W, X, Q gibi harfler iermemesi,

Burada řunu önemle belirtmek gerekir ki bu sınırlama, girilen verilerin dođru ve geerli olmasını garantilemezken, sadece formata uygun olmasını garantiler. Verilerin dođru ve geerli olduđunu garantilemek iin yazılımın kendisine uygun kodlar eklenmeli ve bu kodlarda bu sınırlamalar ifade edilmelidir.

- řehir ii telefon numarasını ieren bir alana girilecek deđerin 7 haneli olması, sadece rakamlardan oluřması ve bařında 0 olmaması kurallarına uyan her telefon numarasının var olduđu ve o řehre ait bir telefon numarası olduđunu garantilemek bu gibi sınırlamalar ile mümkün deđildir. Örneđin 111 11 11 řeklindeki bir telefon numarası deđer bu sınırlamaya uymasına rađmen gerekte böyle bir telefon numarası bulunmayabilir.
- řehirler arası telefon numarasını ieren bir alana girilecek deđerin 11 haneli olması, sadece rakamlardan oluřması ve bařında 0 olması kurallarına uyan her telefon numarasının var olduđu, bařındaki kodun gerekten var olan geerli bir kod olduđunu garantilemek bu gibi sınırlamalar ile mümkün deđildir. Örneđin 0111 111 11 11 řeklindeki bir telefon numarası bu sınırlamaya uymaktadır ancak ölkemizde alan kodu 0111 olan bir řehir bulunmamaktadır, buna göre bu numara geerli bir telefon numarası deđildir.
- e-mail adreslerini ieren bir alana girilecek deđerin mutlaka @ iřaretini ieren bir metin olması ancak bu @ iřaretinin ilk karakter olmaması kurallarına uyan her e-mail adresinin geerli bir e-mail adresi olduđunu garantilemek bu gibi sınırlamalar ile mümkün deđildir. Örneđin x@y.com řeklindeki bir e-mail adresi bu sınırlamaya uymaktadır ancak böyle bir e-mail adresi mevcut olmayabilir.
- T.C. Kimlik No ieren bir alana girilecek deđerin 11 haneli olması, sadece rakamlardan oluřması, son hanesinin mutlaka ift bir sayı olması kurallarına uyan her T.C. Kimlik No deđerinin gerekten Türkiye vatandaşlarından birine ait olan bir T.C. Kimlik No olduđunu garantilemek bu gibi sınırlamalar ile mümkün deđildir. Örneđin 12345678900 řeklindeki bir T.C. Kimlik No deđer bu sınırlamaya uymaktadır ancak gerekte böyle bir numara hiçbir Türk vatandaşına ait olmayabilir.

- Öğrenci numaralarını içeren bir alana girilecek değerin o üniversite, fakülte ve bölüme verilen öğrenci numaraları ile aynı formata sahip olması kurallarına uyan bir öğrenci numarası değerinin gerçekten o bölümdeki bir öğrenciye ait olduğunu garantilemek bu gibi sınırlamalar ile mümkün değildir. Örneğin 1306 şeklindeki bir öğrenci numarası söz konusu bölüm formatına uygun olabilir ancak gerçekte o bölümdeki hiçbir öğrencinin öğrenci numarası 1306 olmayabilir.
- Öğrenci notlarını içeren bir alana girilecek değerin 0-100 aralığında sayısal bir değer olması kurallarına uyan öğrenci notlarının geçerliliğini garantilemek, bu gibi sınırlamalar ile mümkün değildir. Örneğin bir öğrencinin bir dersindeki bir sınav için girilen 63 değeri bu sınırlamaya uygun bir değerdir ancak gerçekte o sınav 20 soruluk bir testten oluşuyorsa notların 5 ve 5'in katları şeklinde olması gerektiğinden 63 şeklinde bir not olması mümkün olmayabilir.
- Öarih içeren bir alanın hangi tarih formatı kullanılıyorsa ona uygun olması, örneğin Gün–Ay–Yıl şeklinde kullanılıyorsa Gün değerinin 1-31; Ay değerinin ise 1-12 arasında olması kurallarına uyan bir tarih değerinin gerçek bir tarih değeri olduğunu garantilemek bu gibi kısıtlamalar ile mümkün değildir. Örneğin 31.02.2021 şeklindeki bir tarih değeri bu sınırlamalara uygun görünmektedir ancak hiçbir yıl içerisinde 31 Şubat şeklinde bir tarih bulunmamaktadır.
- Türkçe metin içeren herhangi bir alana girilecek değerin Türkçe metin içeren herhangi bir alana girilecek değerin Ğ ile başlamaması, W, X, Q gibi harfler içermemesi, kurallarına uyan tüm metinlerin Türkçe'ye uygun bir metin olduğunu garantilemek bu gibi kısıtlamalar ile mümkün değildir. Örneğin çkışmp değeri yukarıdaki sınırlamaya uymaktadır fakat Türkçe'de böyle bir kelime bulunmamaktadır.

**İşletme Kuralları Sınırlaması:** Tablolar üzerinde, işletme kuralları baz alınıp sınırlamalar yapılarak veri bütünlüğü sağlanabilir. Örneğin, muhasebe bölümünde “programcı” kadrosuna sahip eleman çalıştırmak tutarlı değildir, hatta hatalara, yetkisiz erişimlere vb. sebep olacağından önlenmesi gerekmektedir. Veri Tabanı Yönetim Sistemi'nde bilgi girişleri esnasında muhasebe bölümüne yeni personel kaydı eklemek gerektiğinde, bu kural, uygunsuz veri girişlerini engelleyecektir.

**Nitelikler Arası Bağımlılıklar:** Veri bütünlüğünün ve tutarlılığının sağlanmasında, nitelikler arasındaki bağımlılıkların kullanılması yararlı olabilir. Niteliklerin birbirleriyle olan bağımlılığı göz önüne alınmadan Veri Tabanı Yönetim Sistemi tasarımının yapılması sorunlara neden olacaktır. Nitelikler arasındaki fonksiyonel bağımlılıklar belirlenerek, şema üzerinde düzenlemeler yapılır.

## 3.2. İlişkisel Olmayan Veri Tabanı Yönetim Sistemleri

İlişkisel olmayan (Not Only SQL – Sadece SQL Değil – NoSQL) Veri Tabanı Yönetim Sistemi kavramı; ilk olarak 1998 yılında öne sürülmüş ve ilişkisel Veri Tabanı Yönetim Sistemleri'ne alternatif bir çözüm olarak ortaya çıkmıştır. NoSQL Veri Tabanı Yönetim Sistemleri, belirli veri modelleri için özel olarak tasarlanmıştır ve modern uygulamalar oluşturmaya yönelik esnek şemalara sahiptir. NoSQL Veri Tabanı Yönetim Sistemleri uygun ölçekte geliştirme kolaylığı, işlevselliği ve performansı ile geniş çaplı olarak kabul görmüştür. Dünya'da NoSQL örneklerine bakıldığında çok geniş bir kullanım alanı olduğu görülmektedir. Bunun nedeni ilişkisel Veri Tabanı Yönetim Sistemleri'ne ilişkin ölçek sorunu problemine, tüm çözümlerin içinde en iyi cevap vereni NoSQL'dir. Günlük işlem hacimleri terabaytlar ve üzerinde olan Twitter ve Facebook gibi birçok firma, çok büyük verilerin depolanması ve yazılmasında ilişkisel Veri Tabanı Yönetim Sistemleri'nin eksik kaldığı hususlarda, yatay ölçekleme yapan dağıtık NoSQL çözümleri geliştirilmiştir. İlişkisel Veri Tabanı Yönetim Sistemleri kullanıcıları esneklik, ölçekleme, performans, maliyet gibi kriterler dolayısıyla NoSQL veri tabanlarını tercih edebilmektedir. İlişkisel olmayan Veri Tabanı Yönetim Sistemleri'nin en önemli özellikleri aşağıdaki şekilde ifade edilebilir:



- 
- Kolay Ulaşılabilirlik (Basically Availability):**  
Veri erişim sorunlarını ortadan kaldırmak için verilerin kopyaları kullanılır ve paylaşılmış ya da bölümlenmiş veriyi birçok sunucudan alarak kolay erişim sağlar.
  - Esnek Durum (Soft state):**  
İlişkisel Veri Tabanı Yönetim Sistemleri'nde olmazsa olmaz bir gereklilik olarak düşünülen veri tutarlılığı, NoSQL sistemlerde yerini tutarsız ve süreksiz verilerin barınmasına da izin veren sistemlere bırakmıştır.
  - Eninde Sonunda Tutarlı (Eventually consistent):**  
Uygulamalar anlık tutarlılıkla ilgili olmasına rağmen, NoSQL sistemlerin gelecekte bir zamanda tutarlı olacağı farz edilir, diğer bir deyişle anlık tutarlılıklar önemsenmez, sistemin komple tutarlılığı ile ilgilenilir.

**Şekil 3.11.** İlişkisel Olmayan Veri Tabanı Yönetim Sistemleri'nin En Önemli Özellikleri

NoSQL Veri Tabanı Yönetim Sistemleri, ilişkisel veri tabanlarına kıyasla çok daha yeni bir kavramdır. Ancak e-ticaret, internet arama motorları ve sosyal ağlar gibi büyük ölçekli internet uygulamaları için güvenilirliğini kanıtlamıştır. Pek çok kayıt saklama teknolojisi kendilerini NoSQL ürünü olarak sınıflandırmaktadır ve her biri kendilerine özgü karakteristiklere sahiptir. En çok bilinen ve kullanılan NoSQL ürünleri MongoDB, CouchDB, Riak, Redis, Voldemort, Cassandra ve Hbase'dir. NoSQL veri tabanlarında, verilere erişmek ve onları yönetmek için çeşitli veri modelleri kullanılır. Bu tür veri tabanları, özellikle büyük veri hacmi, düşük gecikme süresi ve esnek veri modelleri gerektiren uygulamalar için optimize edilmiştir. Bu gereksinimler, diğer Veri Tabanı Yönetim Sistemleri'nin veri tutarlılığı kısıtlamalarının bir kısmı esnetilerek karşılanır. Basit bir kitap Veri Tabanı Yönetim Sistemi'nin şeması modelleme örneğini ele alalım ve ilişkisel ve NoSQL Veri Tabanı Yönetim Sistemi yapıları ile değerlendirmeye çalışalım. İlişkisel bir Veri Tabanı Yönetim Sistemi'nde genellikle bir kitap kaydı gizlenerek (veya "normalleştirilerek") ayrı tablolarda depolanırken, ilişkiler birincil ve yabancı anahtar kısıtlamaları tarafından tanımlanır. Bu örnekte Kitaplar Tablosu'nun alanlarının ISBN, Kitap Başlığı ve Baskı Sayısı şeklinde olduğu; Yazarlar Tablosu'nun alanlarının Yazar No ve Yazar Adı şeklinde olduğu; Yazar-ISBN Tablosu'nun alanlarının ise Yazar No ve ISBN şeklinde olduğu planlanır. Bu ilişkisel model, Veri Tabanı Yönetim Sistemi'nin yedekliliği ve tekrarlılığı azaltacak şekilde normalleştirilmiş ve genel olarak depolama için optimize edilmiş tablolar arasında gerçekleşmesini sağlayacak şekilde tasarlanır. Şekil 3.11'de bu Veri Tabanı Yönetim Sistemleri yapısı görülmektedir.

## KİTAPLAR Tablosu

ISBN	Kitap Başlığı	Baskı Sayısı

## YAZARLAR Tablosu

Yazar No	Yazar Adı

## YAZAR-ISBN Tablosu

Yazar No	ISBN

Şekil 3.12. Kitap İlişkisel Veri Tabanı Yönetim Sistemi Modeli

Bir NoSQL Veri Tabanı Yönetim Sistemi'nde, kitap kaydı genellikle bir JSON belgesi olarak saklanır. Her kitap için öge, ISBN, Kitap Başlığı, Baskı Sayısı, Yazar Adı ve Yazar No bilgileri tek bir belgede öznitelikler olarak depolanır. Bu modelde, veriler sezgisel yazılım geliştirme ve yatay ölçeklenebilirlik için optimize edilir. Kullanılan NoSQL Veri Tabanı Yönetim Sistemi'nin yapısına göre farklı yapılarda ifade edilebilir. Belli başlı NoSQL Veri Tabanı Yönetim Sistemi türleri şunlardır:



Şekil 3.13. Belli Başlı NoSQL Veri Tabanı Yönetim Sistemi Türleri

**Anahtar-Değer Veri Tabanı Yönetim Sistemleri:** Anahtar-Değer Veri Tabanı Yönetim Sistemleri yüksek oranda bölümlendirilebilir ve diğer Veri Tabanı Yönetim Sistemleri'nin ulaşamayacağı ölçeklerde yatay ölçeklendirmeye imkân tanır. Anahtar-değer veri modeli için özellikle de oyun, reklam teknolojisi ve IoT gibi kullanım örnekleri uygundur. Örneğin Amazon DynamoDB her ölçekte iş yükü için tutarlı,

milisaniye cinsinden tek basamaklı gecikme süresi sunacak şekilde tasarlanmıştır.

**Belge Veri Tabanı Yönetim Sistemleri:** Geliştiriciler için verimli ve sezgisel bir veri modeli olduğundan veriler, uygulama kodunda genellikle bir nesne veya JSON benzeri bir belgeyle temsil edilir. Belge Veri Tabanı Yönetim Sistemleri, geliştiricilerin uygulama kodlarında kullandıkları aynı belge modeli biçimini kullanarak bir Veri Tabanı Yönetim Sistemi'nde veri depolamasını ve sorgulamasını kolaylaştırır. Belgelerin ve belge Veri Tabanı Yönetim Sistemleri'nin esnek, yarı yapılandırılmış ve hiyerarşik yapısı sayesinde geliştiriciler, uygulamaların gereksinimleri doğrultusunda geliştirme yapabilirler. Bu belge modeli; kataloglar, kullanıcı profilleri ve her belgenin hem benzersiz olduğu hem de zamanla geliştiği içerik yönetim sistemleri ile birlikte oldukça kullanışlıdır. MongoDB, esnek ve yinelemeli geliştirme için hem güçlü hem de sezgisel API'ler sağlayan popüler bir belge Veri Tabanı Yönetim Sistemi'dir.

**Grafik Veri Tabanı Yönetim Sistemleri:** Bu sistemin amacı, yüksek oranda bağlı veri kümeleriyle çalışan uygulamalar oluşturup çalıştırmayı kolaylaştırmaktır. Grafik Veri Tabanı Yönetim Sistemleri'nin yaygın olan kullanım örnekleri arasında dolandırıcılık algılama, sosyal ağlar, öneri altyapıları, ve bilgi grafikleri yer alır. Giraph ,Amazon Neptune, Neo4j bu sistemlere örnek verilebilir.

**Bellek İçi Veri Tabanı Yönetim Sistemleri:** Oyun ve reklam teknolojisi uygulamalarının puan tabloları, oturum depoları ve gerçek zamanlı analitik gibi mikrosaniye düzeyinde yanıt süreleri gerektiren ve her an ani trafik artışları yaşanabilen kullanım örnekleri vardır. Bu gibi durumlar için bellek içi veri tabanları kullanılır.

**Arama Veri Tabanı Yönetim Sistemleri:** Pek çok uygulama, yazılım geliştiricilere sorun gidermede yardımcı olmak için günlük oluşturur. Özellikle yüksek seviyede veri kullanan sistemlerde arama Veri Tabanı Yönetim Sistemi yapıları sıklıkla karşımıza çıkmaktadır. Amazon Elasticsearch Service (Amazon ES), yarı yapılandırılmış günlükleri ve ölçümleri dizine ekleyerek, biriktirerek ve bunlarda arama yaparak makineler tarafından oluşturulan veriler için neredeyse gerçek zamanlı görselleştirmeler ve analitik sağlama amacına özel olarak tasarlanmıştır.

### 3.3. İlişkisel ve İlişkisel Olmayan Veri Tabanlarının Karşılaştırması

2000'lerin ortasına kadar uygulama geliştirme için çoğunlukla, Oracle, DB2, SQL Server, MySQL ve PostgreSQL gibi ilişkisel veri tabanları ve ilişkisel veri modeli kullanılmaktaydı. Bu tarihlerden itibaren ise ilişkisel olmayan veya "NoSQL" olarak adlandırılan veri modeli kullanılmaya başlanmıştır. NoSQL veri tabanlarının değişken özelliklere sahip birçok türü olsa genel bazı ortak özellikleri bulunmaktadır. Tablo 3.1'de ilişkisel ve ilişkisel olmayan veri tabanları arasındaki farklardan bazıları gösterilmiştir.

**Tablo 3.1.** İlişkisel ve İlişkisel Olmayan Veri Tabanı Yönetim Sistemleri Arasındaki Farklar

	<b>İlişkisel Veri Tabanı Yönetim Sistemleri</b>	<b>İlişkisel Olmayan Veri Tabanı Yönetim Sistemleri</b>
Veri modeli	Verileri satır/sütun tabloları halinde ifade eder. Tablolar, satırlar, sütunlar, ilişkiler ve diğer öğeler şema ile tanımlanır.	Anahtar-değer, belge ve grafik gibi çeşitli veri modelleri sağlar.
ACID özellikleri	Bölünmezlik, bir işlemin ya tamamen yürütülmesi ya hiç yürütülmemesidir.	
(atomicity, consistency,	Tutarlılık, bir işlem gönderildiğinde verilerin Veri Tabanı Yönetim Sistemi'ne uygun olmasıdır.	Yatay olarak ölçeklendirilebilen daha esnek bir veri modeli sağlamak için genellikle ilişkisel veri tabanlarının bazı ACID özelliklerini esneterek bunlardan ödün verir.

isolation, durability – bölünmezlik, tutarlılık, yalıtım ve dayanıklılık)	Yalıtım, eş zamanlı işlemlerin birbirinden bağımsız olarak yürütülmesini gerektirir.	Bu, tek bir bulut sunucusunun ulaşamayacağı derecede yatay ölçeklendirme gerektiren yüksek performanslı, düşük gecikme süreli kullanım örnekleri için mükemmel bir seçim olmasını sağlar.
En uygun iş yükleri	İşlemsel ve güçlü tutarlılığa sahip çevrimiçi işlem gerçekleştirme uygulamaları için tasarlanmıştır ve çevrimiçi analitik işlem için uygundur.	Düşük gecikme süreli uygulamaları içeren çeşitli veri erişimi desenleri için tasarlanmıştır.
Performans	Performans genellikle disk alt sistemine bağlıdır. En üst düzey performans için genellikle sorguların, dizinlerin ve tablo yapısının optimize edilmesi gerekir.	Performans genel olarak temel donanımın küme boyutu, ağ gecikme süresi ve çağrı yapan uygulama gibi etmenlerin birleşimine bağlıdır.
Ölçek	Genellikle donanımın işlem kapasitesini artırarak ölçeği artırır veya salt okunur iş yüklerine yönelik replikalar ekleyerek ölçeği genişletir.	Erişim desenleri aktarım hızını artırmak için neredeyse sınırsız ölçekte tutarlı performans sağlayan dağıtılmış mimariyi kullanarak ölçeği genişletebildiğinden, genellikle bölümlendirilebilen veri tabanlarıdır.
API'ler	Veri depolama ve alma istekleri, SQL uygun sorgular kullanılarak iletilir. Bu sorgular Veri Tabanı Yönetim Sistemi tarafından ayrıştırılır ve yürütülür.	Nesne tabanlı API'ler, veri yapılarını kolayca depolar.

## Bölüm Özeti

- Veri modellemesi yapılırken farklı model yapıları kullanılmaktadır. Güncel olarak en yaygın kullanılan model ise İlişkisel Model'dir.
- İlişkisel model, varlıklar arasındaki bağlantının içerdiği değerlere göre sağlanması esasına dayanır ve varlıklar arasında oluşan karmaşık ilişkileri basite indirmek amacıyla geliştirilmiştir. Bu yaklaşımda, Veri Tabanı Yönetim Sistemi'ndeki tüm ilişkiler tablolar biçiminde tanımlanmaktadır.
- İlişkisel Veri Tabanı Yönetim Sistemi içinde yer alan her tablo belirtilen özelliklere sahip olmalıdır. Buna göre; tablolar sütunlardan (kolonlardan) oluşur, her bir sütunun ayrı bir adı vardır, her bir sütun, aynı niteliğin tanımladığı aynı etki alanının belirlediği değerleri içerir, her bir satır birbirinden farklıdır, sütunların ve satırların sırası önemsizdir.
- Veri Tabanı Yönetim Sistemi'nin mantıksal tasarımına "Veri Tabanı Yönetim Sistemi şeması" adı verilir. Tablolar ve nitelikleri Veri Tabanı Yönetim Sistemi şemasını oluşturur. Veri Tabanı Yönetim Sistemi şemaları Fiziksel Şema ve Kavramsal Şema olarak gruplandırılabilir.
- Fiziksel şema, Veri Tabanı Yönetim Sistemi'nin fiziki çevresi ile ilgili tanımları içerir.
- Kavramsal şema tüm Veri Tabanı Yönetim Sistemi'nin mantıksal tasarımıdır. Veri Tabanı Yönetim Sistemi'nde bulunmasına karar verilen veriler arasındaki mantıksal ilişkiler yapısının saptanması için oluşturulur. Bu şemada, Veri Tabanı Yönetim Sistemi'nde veri alanları kayıtlar dosyalar vb. gibi ne tür veri elemanlarının bulunacağı, veri elemanları arasındaki ilişkiler ve Veri Tabanı Yönetim Sistemi'nin

yapısı hakkında bilgiler yer alır. Veri Tabanı Yönetim Sistemi şeması dendiğinde genellikle kavramsal şema anlaşılmaktadır.

- Veri Tabanı Yönetim Sistemi'nde veri bütünlüğünü sağlamak için Anahtar Sınırlamaları, Veri Değerleri Sınırlamaları, İşletme Kuralları Sınırlaması ve Nitelikler Arası Bağımlılıklar olmak üzere çeşitli sınırlamalar bulunmaktadır.

## Kaynakça

Kadir Çamoğlu, “Programlama ve Veritabanı Mantığı”, Kodlab, 1. Baskı, 2009

Ömer Coşkun, “İlişkisel ve İlişkisel Olmayan Veri Tabanı Yönetim Sistemlerinin Karşılaştırılması ve Performans Analizi”, İstanbul Üniversitesi-Cerrahpaşa, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Programı, Yüksek Lisans Tezi, 2019

Rıfat Çölkesen, “Veri Yapıları ve Algoritmalar”, Papatya Yayıncılık, 9. Basım, 2014

Yalçın Özkan, “Veritabanı Sistemleri”, Papatya Yayıncılık, 4. Basım, 2015

WEB: <https://e-bergi.com/y/iliskisel-veritabanlari/>

WEB: S. Davaz, “NoSQL Nedir Avantajları ve Dezavantajları Hakkında Bilgi”  
<https://blog.kodcu.com/2014/03/nosql-nedir-avantajlari-ve-dezavantajlari-hakkinda-bilgi>

WEB: G. Vaish, “Getting Started With NoSQL”, Packt Publishing, United Kingdom, 2013

WEB: M. Otey, “NoSQL? No Way!”, SQL Server Magazine, Sayfa: 5, 2010

WEB: “Oracle NoSQL Database”,  
<http://www.oracle.com/technetwork/database/nosqldb/learnmore/nosql-database-498041.pdf>

WEB: N. Rozanski, E. Woods, “Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives”, Pearson Education Inc., USA, 2012.

WEB: G. Burd, “NoSQL”, Sysadmin Journal, Vol:36 No. 5 ss:5-12 ABD, 2011.

WEB: <https://aws.amazon.com/tr/nosql/>