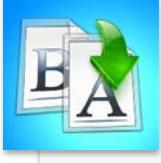


VERİ TABANI VE NORMALİZASYON



- HEDEFLER
 - İlişkisel Veri Tabanı Tasarımı
 - İlişkisel Veri Tabanının Kavramsal Tasarımı
 - Varlık İlişki Modeli
 - Varlık kümesi
 - Zayıf Varlık Kümesi
 - Veri Tabanı Normalizasyon Kuralları
 - Bağımlılıklar
 - Normal Formlar

İÇİNDEKİLER



- Bu üniteyi çalıştıktan sonra;
 - Veri tabanı verileri arasındaki ilişkiyi kavrayabilecek,
 - Veri tabanı yapısını anlayabilecek,
 - Veri tabanlarında yapılabilecek, normalizasyon türlerini ve kullanım alanlarını kavrayabileceksiniz.

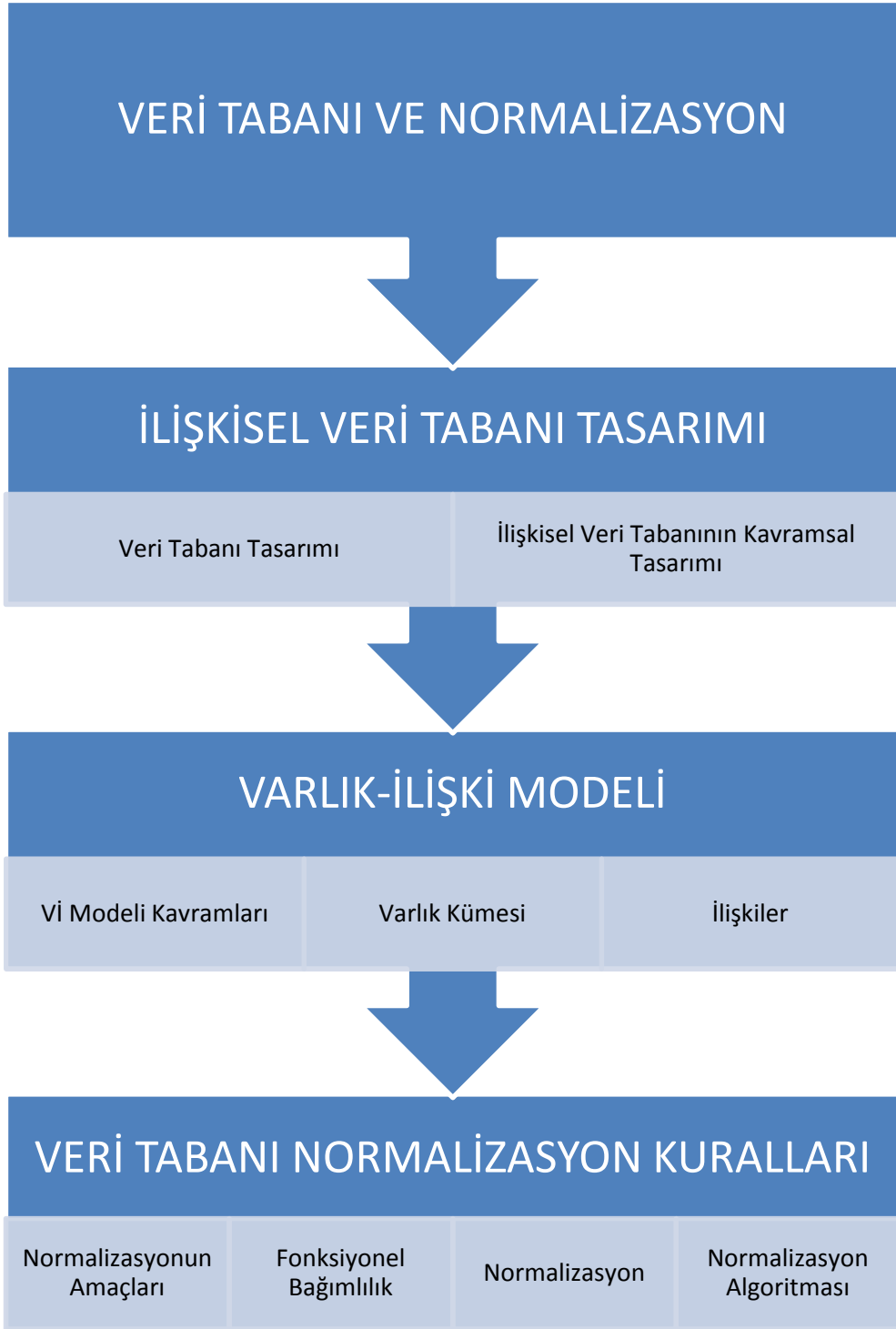
HEDEFLER



Atatürk Üniversitesi
Açıköğretim Fakültesi

VERİ TABANI YÖNETİM SİSTEMLERİ Doç. Dr. Abdulkadir ÖZDEMİR

ÜNİTE 2



GİRİŞ

Veri tabanları verilerin organize edildiği ve sonradan erişilip üzerinde güncelleme gibi işlemler yapıldığından erişim hızı ve doğru veri güncellemenin önemli olduğu yapılarıdır.

Verilerin hızlı bir şekilde erişilip üzerlerindeki işlemlerin doğru ve tek bir yerde yapılabilmesi için ise veri tabanının iyi tasarlanmış ve tablolarında normalizasyon işlemlerinin yapılmış olması gereklidir.

Veri tabanı tasarımı verileri saklanacak konuya uygun bir veri yapısı tasarımı ile başlar. Veri yapısının uygun olarak seçilmemiş olması verilerin istenilen ayrıntıda kaydedilmesine engel olduğu gibi sonrasında veri tabanında yapısal değişikliklerin yapılmasına yol açacaktır.

Veri tabanları tasarlanırken bazen gerçek uygulamadaki veri yapıları ve ilişkilere göre tasarlanır. Bunun sonucu olarak ta veri tabanlarında performans kaybına ve hatta veri bütünlüğünün bozulması nedeniyle veri kaybına yol açabilir.

Bu gibi durumların önüne geçmek için de veri tabanları tasarlandıktan sonra normalizasyon kurallarına göre yeniden yapılandırılır. Bu yapılandırmalar veri tabanı tablolarının durumuna bir adımda veya durumuna göre birden fazla adımda yapılabilir. Normalizasyon 5 ana ve bir alt işlemde oluşur ve bunlar *normal form* olarak isimlendirilir. Birçok veri tabanı için ilk üç normal form yeterli olurken çok sayıda ve karışık yapılarda veriler barındıran veri tabanları için diğer normal formların işletilmesine ihtiyaç duyulabilir.

Veri tabanı tasarımcıları tecrübe kazandıkça normal formlara uygun tasarımlar yapmaya başlayacaklarından normalizasyona ihtiyacı olmayan tasarımlarda yapabilmektedir. Ancak tasarım konusunda tecrübesi olmayanların, normalizasyon kurallarına göre veri tabanı yapılarını düzeltmeleri gerekir.

İLİŞKİSEL VERİ TABANI TASARIMI

Veri tabanı tasarımında yetenek, bilgi ve tecrübe önemlidir. Ancak veri tabanı sistemi ilişkileri ve normalizasyon kuralları da bilinmelidir. Bu kurallar veri tabanının hangi şartlara göre tasarlanması gerektiğini belirler ve veri tekrarını, veri kaybını veya veri yetersizliğini önler. Bu kurallar depolanan veri miktarı artıkça önem kazanmaktadır.

Veri Tabanı Tasarımı

Veri tabanı projenin başında tasarlandığından, çok iyi tasarlanmalıdır ki daha sonraki uygulamalarda iş sürecini aksatmasın. (Kaya, 2007)

Aşağıda veri tabanı tasarımı yapılırken izlenecek adımlar verilmiştir:

- **Depolanacak Veriler:** Veri tabanı içerisinde tutmak istediğiniz bilgiler belirlenerek bunlar gruplandırılmalıdır.
- **Tabloların Oluşturulması:** Belirlenen veri grupları ve sütunlar doğrultusunda tablolar oluşturulur.

- **Anahtar Sütunların Belirlenmesi:** Kayıtların birbirinden ayırt edilebilmesi için anahtar sütun oluşturulur. Anahtar sütunun tanımlanma zorunluluğu yoktur ama verilere daha çabuk ulaşmak ve tekrar eden kayıtların önlenmesi için kullanılabilir.
- **Tabloları Bölme:** Tabloda tekrar eden kayıtlarla karşılaşılacaksa tekrar eden sütun için yeni tablo oluşturulur.
- **İlişkilerin Kurulması:** Projelerin büyük kısmında tablolar arasında kesinlikle ilişki oluşturulmaktadır.

İlişkisel Veri Tabanının Kavramsal Tasarımı

Kavramsal tasarım, veri tabanında tutulacak verilerin daha üst seviyede gösterilmesi için kullanılır. Kavramsal tasarım için en çok kullanılan model Entity Relationship (ER) [Varlık İlişki (VI)] modelidir.

Varlık-ilişki modeli kavramsal tasarım için kullanılan en popüler modeldir. Bu model kullanılarak VTYS'den bağımsız modelleme yapılarak ve ilişkiler tanımlanarak herhangi bir VTYS veri tabanına dönüştürülebilir.

Varlık-ilişki modelinde kullanılan şekiller veri tabanlarının şematik olarak tasarlanması için kullanılır.

Varlık-ilişki modelinde temel üç öge vardır. Bunlar (Özseven, 2012):

- **Varlık:** Modelin en temel ögesidir. Var olan ve benzerlerinden ayırt edilebilen her şey varlıktır. Örneğin, kitap, öğrenci, araba birer varlıktır. Birden fazla varlığın oluşturduğu kümeye varlık kümesi denilir. Veri tabanı olarak düşünülürse her bir tablo bir varlık kümesidir.
- **Nitelik:** Varlıkların her bir özelliği bir nitelik olarak ifade edilir. Örneğin, öğrencinin numarası ve bölümü öğrenci varlığının nitelikleridir. Nitelik bağlı olduğu varlığa düz bir çizgi ile birleştirilir. Veri tabanı olarak düşünülürse tablonun her bir sütunu bir varlığı gösterir. Bir niteliğin değeri her bir varlık için farklıysa bu nitelik anahtar nitelik olarak belirlenir. Anahtar nitelik şema içerisinde niteliğin altı çizilerek gösterilir. Örneğin, her bir öğrenci varlığı farklı öğrenci numarası niteliklerine sahip olacağı için öğrenci no niteliği anahtar nitelik olarak belirlenebilir.
- **Domain (Etki Alanı):** Niteliklerin alabileceği değer aralığıdır. Örneğin, öğrenci notlarını içeren sınav niteliği için alacağı değerleri 0 ile 100 arasında belirlemek etki alanı oluşturmaktadır. Etki alanı VI şeması içerisinde gösterilmez.
- **İlişki:** Farklı varlıklar arasındaki ilişkileri ifade eder. Örneğin, öğrenci ve dersler ayrı varlık kümeleridir ama öğrenciler ders almak zorunda olduğu için iki varlık arasında ders alma ilişkisi vardır. Baklava dilimi ilişkili olduğu varlıklarla düz çizgi ile bağlanır. Tablolar arasında kurulan ilişkiler (1-n, 1-1, n- m) model içerisinde ilişki olarak geçmektedir. İki varlık kümesi arasında birden fazla ilişki bulunabilir.



Varlık İlişki Modeli,
veri tabanı tasarım
aşaması için
önemlidir.

VARLIK-İLİŞKİ MODELİ

Veri tabanı tasarım işlemi birkaç aşamadan oluşur (Özseven, 2012):

- İlk olarak veri tabanı kurulacak kuruluşun ihtiyacı olan bilgilerin analizi yapılır.
- Veriler toplanıp sistem analizi yapıldıktan sonra, ilişkisel veri modeline göre veri tabanı şeması oluşturulur. Bu şema, veri tabanı kullanıcılarının ihtiyaç duydukları verilerin kısa açıklamasıdır.
- Bu şema içerisinde varlık tipleri ve ilişkilerin açıklaması bulunur. Daha sonra bu şemaya göre tablolar oluşturulur.

İlişkisel veri modeli türlerinden olan ER (Entity-Relationship) veya Türkçedeki kullanımı ile Varlık-İlişki (Vİ) modeli günümüzde yaygın olarak kullanılmaktadır.

Vİ Modeli Kavramları

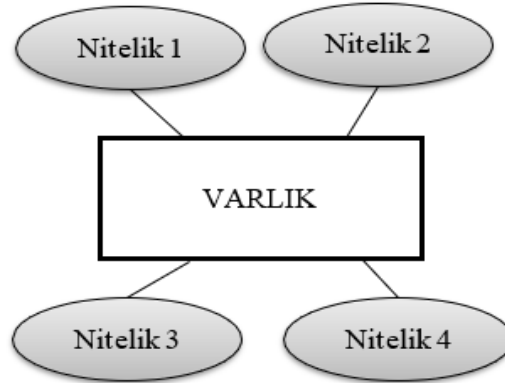
Varlık ve nitelikleri

Gerçek hayatta diğerlerinden ayırt edilebilen nesnelere varlık denir. Bir varlık, kişi, araba, ev veya çalışan gibi fiziksel nesnelere olabileceği gibi, şirket, iş veya ders gibi fiziksel olmayan nesnelere de olabilir. Her varlık kendini tanımlayan kendine has özelliklere sahiptir. Örneğin, bir çalışan varlığı; çalışan adı, yaşı, adresi, maaşı ve görevi özellikleri ile tanımlanır. Her varlığın niteliğinin bir değeri olur.

Varlık Kümesi

Veri tabanında benzer varlıklar ve nitelik değerlerinden oluşan kümeye varlık kümesi denir.

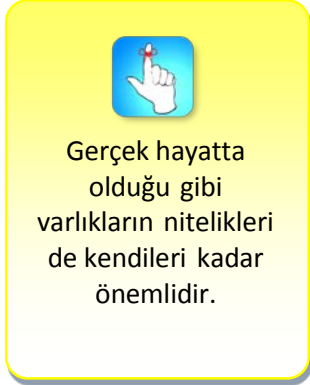
Vİ şemasında varlık kümeleri dikdörtgen içinde belirtilir. Nitelikler ise oval bir daire içinde belirtilerek ilgili varlık kümesine çizgi ile bağlanır. Şekil 2.1.de bir varlığa ait Vİ gösterilmiştir. Aşağıdaki örnekte varlığa ait 4 nitelik olduğu gösterilmiştir.



Şekil 2-1. Varlık İlişki Diyagramı örneği

İlişkiler

İki veya daha fazla varlık kümesi arasında kurulan anlamlı bağıntılara ilişki denir. İlişkiler Vİ şemasında eşkenar dörtgen ile gösterilir. Eşkenar dörtgen içine

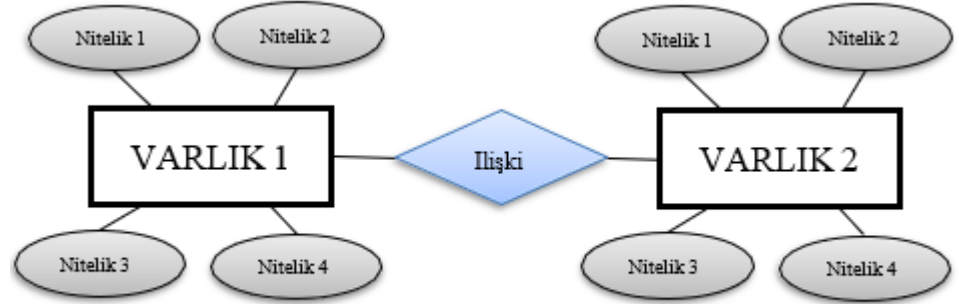


ilişkinin adı yazılır. İki varlık kümesi arasında birden fazla ilişki de olabilir.

İlişki derecesi

İlişkide bulunan varlık kümelerinin sayısı ilişkinin derecesini oluşturur.

Genelde ilişkiler iki varlık kümesi arasında yapılır. Ancak bazı durumlarda, ilişkide ikiden fazla küme yer alabilir. İlişkilerin gösterildiği şema Şekil 2.2.de gösterilmiştir.



Şekil 2.2. İlişki Diyagramı

İlişki türleri

Varlık kümeleri aralarında 3 türde ilişki kurulabilir.

Bire-bir (1-1) ilişki

Bir varlık kümesinin bir elemanının başka bir varlık kümesindeki bir elemanla ilişkisi olarak tanımlanabilir. Buna örnek olarak bir kişinin sadece bir kişi ile evlenmesi gösterilebilir.



Şekil 2.3. Bire-Bir İlişki

Bire-çok (1-n) ilişki

Bir varlığın başka bir kümedeki farklı varlıkla birden çok ilişkisi buna örnek olarak verilebilir. Bir kişinin birden çok kitabı kütüphaneden ödünç alması buna örnek olarak verilebilir.



Şekil 2.4. Bire Çok İlişki

Çoka-çok (n-n ya da n-m) ilişki

Bir varlığın başka varlıkla iki yönlü veya çoklu ilişkisidir. Çok sayıda kişinin ödünç kitap alması ve kitapların çok sayıda kişiye ödünç verilmesi buna örnek olarak verilebilir.

Varlıklar arasında üç tür ilişkiden bahsedilebilir.



Şekil 2.5. Çoktan Çokla ilişki

Zayıf Varlık Kümeleri

Bir varlık kümesi anahtar niteliğe sahip değilse zayıf varlık kümesi olarak adlandırılır. Zayıf varlık kümeleri güçlü varlık kümeleri ile ilişkilendirilerek kullanılır. Yani zayıf varlık kümelerinin güçlü varlık kümelerine var olma bağımlılığı vardır. Zayıf varlık kümeleri çift çizgili dörtgen ile gösterilir.

Tablo 2.1. Varlık İlişki Sembolleri

Sembol	Açıklama
	Varlık kümesi
	Nitelik
	Anahtar Nitelik
	İlişki
	Çok Değerli Nitelik
	Zayıf Varlık Kümesi



Örnek

- Bir üniversitenin personellerine ait araç plakalarının tutulduğu tablodaki bilgiler doğrudan ilgili personele bağlı olduğundan, yani personelin üniversitede olmaması halinde plakanın ilişkilendirileceği kayıt olmaması nedeniyle plaka verileri zayıf varlık kümesi olarak isimlendirilir.

VERİ TABANI NORMALİZASYON KURALLARI

Normalizasyon, veri tabanının tasarım aşamasında veri tekrarını, veri kaybını veya veri yetersizliğini önlemek için gerçekleştirilen işlemlerdir. (Özseven, 2012)

Tabloların oluşturulması, tablolar arasında artık veri ve tutarsız ilişkilerin ortadan kaldırılması normalizasyonun temel amacıdır.

Veri tabanı normalleştirilmesi için birkaç kural bulunmaktadır. Her kurala "normal form" adı verilir. İlk kural kullanılıyorsa, veri tabanının "ilk normal formda" olduğu söylenir. İlk üç kural kullanılıyorsa, veri tabanı "üçüncü normal

formda" olarak nitelendirilir. Başka normalleştirme düzeyleri de kullanılabilmele birlikte, çoğu uygulama için en yüksek düzey üçüncü normal formdur ama daha üst seviye olan dört ve beşinci normalizasyon kuralları da vardır.

Genel olarak normalleştirme için ek tablolar gerekir ve çoğunlukla bunun ek yük getirdiğini düşünülebilir. Normalleştirmenin ilk üç kuralından biri ihlal edildiği takdirde, uygulamalarda artık verilerin oluşturabileceği sorunlara karşı hazır olunması gerekir.

Bu kurallar, depolanacak verilerin neler olacağını belirler ve kayıtlar arasında veri tekrarını, veri kaybını ve yetersizlikleri engeller. Normalizasyon kurallarına uygun tasarım yapılmadığında kayıt güncelleme, kayıt bulma ve yeni kayıt işlemlerinde sorunlarla karşılaşılabilir.

Üst normalizasyon kuralları alt normalizasyon kurallarını kapsamaktadır. Yani, 2NF kuralları 1NF kurallarını ve 3NF kuralları 1NF, 2NF kurallarını kapsar.

Normalizasyonun Amaçları

- **Veri Bütünlüğünü Sağlamak:** Eğer bir sütun için gereksiz veri tekrarı varsa bu sütun bir süre sonra birbirinden farklı değerler içermeye başlayacaktır. Örneğin, bir veri tabanı içerisinde kişilerin adresleri birden fazla tabloda kullanılmışsa ve veri bütünlüğü sağlanmazsa zamanla aynı kişi için birden fazla adres bilgisi oluşacaktır. Böyle bir durumla karşılaşmamak için normalizasyon kuralları adresin tek bir tabloda saklanması ile veri bütünlüğü sağlanabilir.
- **Uygulamadan Bağımsızlık:** Hazırlanacak ilişkisel model kullanılacak uygulamaya göre değil saklanacak veriye göre hazırlanmalıdır. Böylece, kullanılan uygulama değişse de veri modeli tutarlı olarak çalışmaya devam edecektir.
- **Performansı Arttırmak:** Gerekli normalleştirilme işlemleri yapılmış veri tabanı, veri tekrarını en aza indireceği için veri tabanı boyutunu azaltacağından disk alanından da tasarruf sağlanmasına neden olacaktır. Bu ise veri tabanı içerisinde gerçekleştirilen arama ve veri güncelleme işlemlerinin daha kısa sürede gerçekleşmesini sağlayacaktır.

Fonksiyonel bağımlılık

R'nin ilişkiyi (relation) A ve B'nin bir özelliği ya da özellik kümesini temsil ettiği kabul edilir. Eğer R ilişkisinde her bir A değeri, tam olarak bir B değerine işaret ediyorsa B, A ya fonksiyonel olarak bağımlıdır denebilir. Bu durum sembolik olarak da $A \rightarrow B$ şeklinde gösterilebilir (A fonksiyonel olarak B'yi tanımlar.). Aşağıdaki örnek tabloda "M.No" birincil anahtar ise kişi "Ad-Soyadı" bu anahtara fonksiyonel olarak bağımlıdır denir. Bu tablodan hareketle her M.No'su bilinen kişinin adı ve şehrini elde etmek mümkündür.



Üst normalizasyon kuralları alt normalizasyon kurallarını kapsar.

Tablo 2.2. Fonksiyonel Bağımlılık Örneği

M. No	Ad-Soyadı	Şehir
125	Ali Okur	Erzurum
126	Eda Koş	Ankara
130	Veli Yaz	Erzurum

Kısmi bağımlılık

Anahtar olmayan alan, birleşik anahtarın sadece bir kısmı ile belirlenebiliyorsa buna kısmi bağımlılık denir. Tablo 2.2.deki kişiler tablosunun müşterilere ait olduğu ve bu tablodan ayrı olarak birde aşağıdaki gibi satışlar tablosu olduğunu düşünürsek burada M. No'nun tekrar ettiğini ve bu kayıtlardan birinin olmaması durumunda veri bütünlüğü bozulmayacağından kısmi bağımlılıktan bahsedilebilir.

Tablo 2.3. Kısmi Bağımlılık Örneği

M. No	Ad-Soyadı	ÜrünKodu	Adet
125	Ali Okur	M0125	1
126	Eda Koş	M0225	1
130	Veli Yaz	M0125	2
125	Ali Okur	M0225	1

Normalizasyon

Normalizasyon yapılırken uyulması gereken kurulların her birine normal form adı verilir. 5 tip normalleştirme kuralından bahsedilebilir.

- Birinci Normal Form (1NF)
- İkinci Normal Form (2NF)
- Üçüncü Normal Form (3NF)
- Dördüncü Normal Form (4NF)
- Beşinci Normal Form (5NF)

Bu Normal formlardan ilk üçü çok kullanılırken son ikisinin kullanımı daha azdır. İlk üç normal form kayıt güncelleme, kayıt silme ve kayıt bulmada kolaylık sağlar.

3NF'da olan tablolar 1NF ve 2NF'ye uygundur. 2NF'da olan tablolar ise 1NF'ye uygundur.

Birinci Normal Form (1NF)

Bu formda veri tabanında bulunan tablolar ilişkilendirilebilir bir şekilde tasarlanmalıdır.

- Her sütunda sadece bir veri tutulmamalıdır yani birden fazla bilgi tek bir sütunda olmamalıdır.
- Bir alan içerisindeki bilgi özel karakterlerle ayrılarak tutulmuş olmamalıdır. Bir veri tabanı yukarıda belirtilen kurallara uyuyorsa 1NF yani birinci normal form kuralı yerine getirilmiş denir.



Veri tabanı normalizasyonunda ilk üç normal form daha çok kullanılır.



Her sütunda sadece bir veri saklanmalıdır.

Tablo 2.4. Örnek Tablo

Ogr_no	Ad-Soyad	Bolum_kodu	Bolum	Ders_kodu	Sinav_S
110012	Hasan Oku	BLGP	Bilgisayar Prog.	B101, B102, B103	70, 85, 45
110013	Eda Boz	BLGP	Bilgisayar Prog.	B102, B103, B104	25, 60, 55
110014	Ahmet Yaz	ELK	Elektrik	E201, E205, E204	45, 66, 74
110015	Ayşe Koş	ELK	Elektrik	E205, E204, E208	60, 78, 75

Bu tabloyu 1NF kurallarına göre inceleyecek olursak:

Ders_konu ve Sinav_S sütunlarındaki her bir alanda **birden çok veri** tutulmaktadır, veriler ise virgüllerle ayrılmış durumdadır. Bu ise yukarıdaki iki kuralı ihlal anlamına gelmektedir.

Yukardaki tabloya ilk normalizasyon kuralı uygulandığında kurala uymayan değerlerle tablo aşağıdaki gibi olacaktır:

Tablo 2.4. 1NF Uygulanmış tablo

Ogr_no	Bolum_kodu	Ad-Soyad	Bolum	Ders_kodu	Sinav_S
110012	BLGP	Hasan Oku	Bilgisayar Prog.	B101	70
110012	BLGP	Hasan Oku	Bilgisayar Prog.	B102	85
110012	BLGP	Hasan Oku	Bilgisayar Prog.	B103	45
110013	BLGP	Eda Boz	Bilgisayar Prog.	B102	25
110013	BLGP	Eda Boz	Bilgisayar Prog.	B103	60
110013	BLGP	Eda Boz	Bilgisayar Prog.	B104	55
110014	ELK	Ahmet Yaz	Elektrik	E201	45
110014	ELK	Ahmet Yaz	Elektrik	E205	66
110014	ELK	Ahmet Yaz	Elektrik	E204	74
110015	ELK	Ayşe Koş	Elektrik	E205	60
110015	ELK	Ayşe Koş	Elektrik	E204	78
110015	ELK	Ayşe Koş	Elektrik	E208	75

1NF sonucu elde edilecek tablonun yeni halinde tekrarlı satırlara sahip olacağı için bu tabloya satır ekleme, satır silme ve satır güncellemede sorunlar yaşanabilir.

İkinci Normal Form (2NF)

1NF'de karşılaşılan sorunlardan güncelleme sorununu çözmek için tablo 2NF kuralına uygun hâle getirilir. 2NF, nitelikler arasındaki fonksiyonel bağımlılıktan yararlanılarak tabloların birden fazla tabloya dönüştürülmesi ile sağlanır.

Dönüştürmede uygulanacak kurallar ise şöyledir.

Kurallar:

- Bir tablo içinde tanımlı ancak anahtar olmaya uygun sütunlar anahtar olarak tanımlanmalı ve tanımlı **birincil anahtar** sütunlara bağlanmalıdır. Anahtar sütunun ihtiyaç duyduğu bilgileri içermelidir. Örneğin, öğrenci bilgilerinin tutulduğu bir tabloda **not ve ders bilgisinin** olması gereksizdir. Çünkü notlarla ilgili asıl erişim noktası öğrenci numarası olacaktır. Bölüm veya ad gibi bilgiler ayrıntı olarak kalacaktır. Bunu çözmek için **öğrenci** bilgileri ve **not** bilgileri ayrı tablolarda tutulmalıdır.
- Anahtar sütun birden fazla sütunun birleşiminden oluşuyorsa tabloda yer alacak veriler iki sütuna da bağımlı olmalıdır. Tek sütuna bağımlı ise ayrı



Tablolarda aynı kayda ait tekrar eden veriler ayrı bir tabloda tutulmalıdır.

bir tabloda tutulmalıdır. Örneğin, bir ders farklı bölümlerde veriliyorsa ders kodu bölüm kodu ile birlikte bir anahtar olarak tanımlanmalıdır.

Tablo 2.5.e 2NF uygulandıktan sonra tablolarımız şu şekli alacaktır. Buradan da görüldüğü gibi öğrencilerin bilgilerini tutan bir özlük tablosu oluşturulmuştur. Bu tabloda Ogr_no birincil anahtar saha olarak tanımlanmalı ve dersler tablosundaki yabancı anahtar sahası olarak tanımlanacak olan Ogr_no sütunu özlük tablosundaki Ogr-No alanı ile ilişkilendirilmelidir.

Tablo 2.5. 2NF Sonrası Öğrenci Özlük Tablosu

Ogr_no	Ad-Soyad	Bolum_kodu	Bolum
110012	Hasan Oku	BLGP	Bilgisayar Prog.
110013	Eda Boz	BLGP	Bilgisayar Prog.
110014	Ahmet Yaz	ELK	Elektrik
110015	Ayşe Koş	ELK	Elektrik

Tablo 2.6. 2NF Sonrası Dersler Tablosu

Ogr_n	Ders kodu	Sınav S
110012	B101	70
110012	B102	85
110012	B103	45
110013	B102	25
110013	B103	60
110013	B104	55
110014	E201	45
110014	E205	66
110014	E204	74
110015	E205	60
110015	E204	78
110015	E208	75

İkinci Normal Form sonucu güncelleme sorunu çözülmüştür ancak kayıt ekleme ve kayıt silme sorunu devam etmektedir.

Üçüncü Normal Form (3NF)

2NF'de karşılaşılan sorunları çözmek için geçişli bağımlılıkları da ortadan kaldırmak gerekmektedir. 2NF'de sadece anahtar sütunlara göre bağımlılıklar kullanılmıştı, 3NF'de ise bir tablo içerisinde anahtar olmayan bir sütun, başka bir tablonun anahtar sütunu veya bulunduğu tablonun sütunlarıyla ilgili olmalıdır veya bir tablo için anahtar olmayan bir sütun anahtar olmayan başka hiçbir sütuna bağımlı olmamalıdır. Ayrıca, veri tabanındaki ilişkiler 2NF kuralına uymalıdır.

Öğrenci tablosu için Bolum → Bolum_kodu geçişli bağımlılığı mevcuttur. Öğrenci tablosundaki sorunlardan kurtulmak için bu bağımlılığın da kaldırılması gereklidir. Bağımlılığı kaldırmak için bölümler ayrı bir tablo olarak oluşturulmalıdır.

Bu düzenlemeden sonra Öğrenci Özlük tablosunu Öğrenci tablosu ve bölümler tablosu olarak 2 ayrı tablo hâline getirebiliriz. Tablolar arasındaki ilişkiler ise Bolum_kodu ve Bolum_k arasında oluşturulacak ilişki ile veri bütünlüğü sağlanmış olur.

Birçok yazılım uygulamasında üçüncü normal form ile normalizasyon işlemi tamamlanmış olmaktadır. Ancak çok kapsamlı ve karmaşık veri yapılarında üçüncü



Tablolar arasında anahtar alanlar ile bağımlılıklar belirlenir.

normal form yeterli olmayacağından diğer normal form işlemleri de yapılmalıdır.

Boyce-Codd Normal Form(BCNF)

Bir tablonun BCNF olup olmadığını anlamak için tablonun tüm belirleyicileri tespit edilip, her birinin aday anahtar özelliği taşıyıp taşımadığını kontrol edilir.

Veri tabanı tabloları 1NF, 2NF ve 3NF işlemine uygun olarak düzeltildikten sonra bazı tablolarda tüm alanların aday anahtar olarak kaldığı görülebilir. Bu durumda BCNF işlemine göre tablo veya tabloların parçalanması gerekir.

Veri tabanı tasarımında $A \rightarrow B$ şeklinde bir fonksiyonel bağımlılık bulunuyorsa bu bağımlılıktaki B birincil anahtar olmak zorundadır. 3NF tasarımında A anahtarı bir aday anahtar (candidate key) olmak zorunda değildir. Ancak BCNF’de bunun tersine $A \rightarrow B$ şeklindeki bir fonksiyonel bağımlılık durumunda A bir aday anahtar olmalıdır.

Tablo 2.6.daki Eda Boz silindiğinde Tablo 2.7.deki Eda Boz’a ait dersler yok olacaktır. Ayrıca yeni bir öğrenci kayıt olana kadar başka bir ders ortaya çıkmayacaktır.

Dördüncü Normal Form (4NF)

Birincil anahtar olan sütunlar ile anahtar olmayan sütunlar arasında birden fazla bağımsız 1-n ilişkiye izin verilmez. 4NF sağlamak için her bağımsız 1-n ilişki için ayrı tablo oluşturmak gerekir.

4NF’yi açıklamak için dersleri veren hoca tablosunu ele alalım. Bu tabloda her hocanın bir dersi verdiği öngörülmüştür.

Tablo 2.9. 4NF Uygulanacak Tablo

HocaNo	Hoca Ad-Soyad	Bölümü	Ders
125	Hakan Bilen	BLGP	Veri Tabanı
130	Ayşe Soykan	ELK	Devreler
140	Ahmet Kul	BLGP	Programlama

Tablo incelendiğinde ilk olarak herhangi bir sorun olmadığı görülür. Bu tablo için birincil anahtar HocaNo sütunudur. Bu tabloya göre her öğretim elemanı sadece bir derse girebilir. Aynı öğretim elemanı için iki veya daha fazla ders girilmek istendiğinde birincil anahtardan dolayı aynı HocaNo kodu kullanılmayacak dolayısıyla da yeni ders girilemeyecektir ve 1-n ilişki söz konusu olmayacaktır. 4NF kuralını ve 1-n ilişkiyi sağlamak için mevcut tablo iki ayrı tabloya bölünür. Aşağıda 4NF kuralına uygun tablolar verilmiştir.



Kayıtlar arasında birden çoğa ilişkiler varsa bunlar 4NF ile ortadan kaldırılır.

Tablo 2.10. 4NF Uygulamasından Sonraki Hoca Tablosu

HocaNo	Hoca Ad-Soyad	Bölümü
125	Hakan Bilen	BLGP
130	Ayşe Soykan	ELK
140	Ahmet Kul	BLGP

Tablo 2.11. Dersler Tablosu

HocaNo	Ders
125	Veri Tabanı
130	Devreler
140	Programlama
125	Donanım
125	Sistem Analizi
130	Tasarım

Görüldüğü gibi hocaların verebileceği dersler için ayrı bir tablo oluşturulmuş olup, burada her hocaya birden fazla ders girmek mümkün olmuştur.

Tekrarları önlemek için her tabloyu mümkün olduğunca küçük parçalara bölmek gerekir. Aslında ilk 4 kural bu işlemi gerçekleştirir ama bu kurallar kapsamında olmayan tekrarlamalar da beşinci normalizasyon kuralı ile giderilir.

Beşinci normal formda olan bir kayıt aynı zamanda dördüncü, üçüncü, ikinci ve birinci normal formlardadır. Beşinci normal form dördüncü normal formdan simetrik bir kısıtlama olmadığı sürece farklı değildir. Böyle bir kısıtlamanın yokluğunda, dördüncü normal formdaki bir kayıt tipi her zaman beşinci normal formdadır.

Tekrarlamaları ortadan kaldırmak için her bir tablonun mümkün olduğunca küçük parçalara bölünmesi gerektiğinden daha önce bahsedilmişti. İlk dört normal formda olmayan tekrarlamalar beşinci normal formlarla giderilebilir.

Örneğin bir firma için stok kaydını tutması istenilen bir veri tabanı programı hazırlandığında, ilgili firma daha sonraki bir zamanda ürünün stoktan çıkış sebebini de sisteme girmek istediğini belirttiğinde bu sütuna girilecek olan bilgiler bellidir.



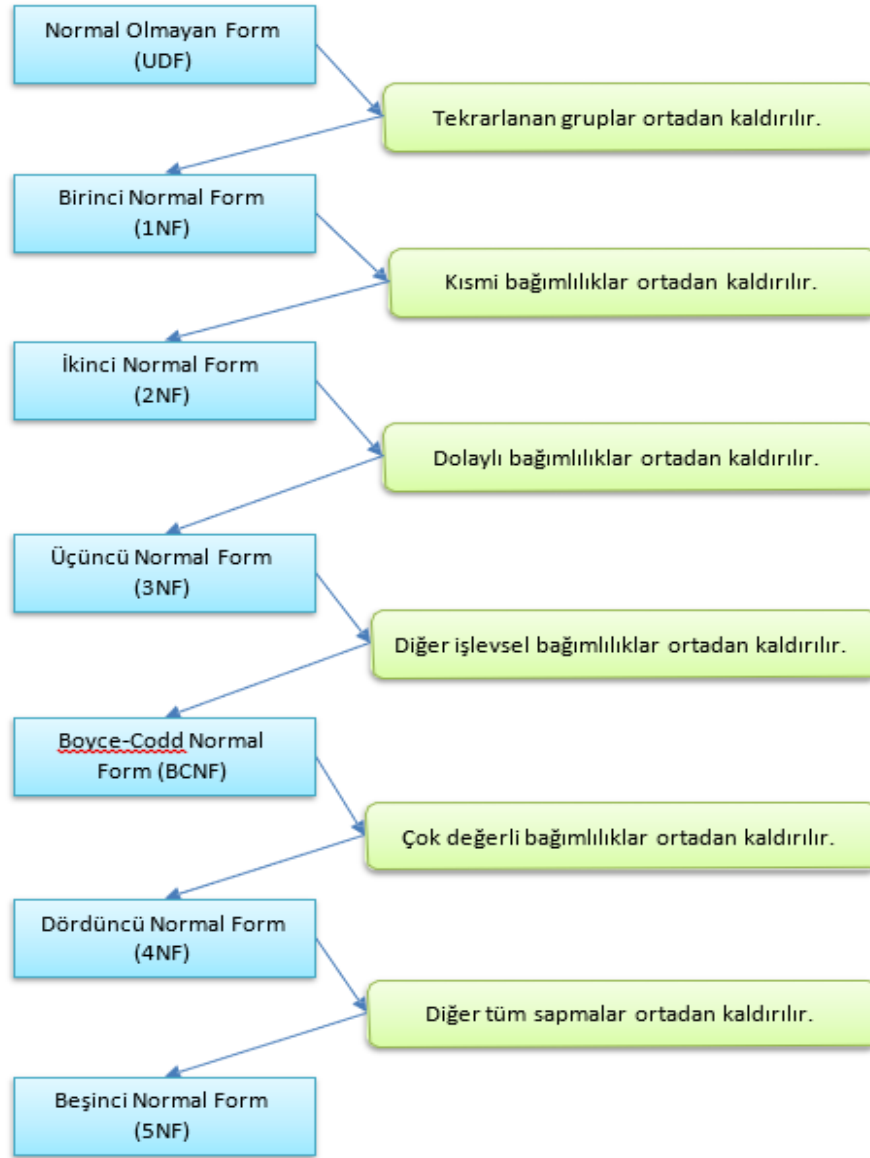
Bireysel Etkinlik

- Normalizasyon yapılmayan tablolarda verilerin işlenmesi daha zordur.
- Normalizasyon tablo sayısını artırsa da sonuçta tablolara erişim ve veri tekrarını ortadan kaldırması nasıl bir tasarruf sağlar.

Normalizasyon Algoritması

Öğrenciler için hazırlanan ilk tabloda normalizasyon yapılmadığından verilerle ilgili sorunlar vardı. Bu sorunları ortadan kaldırmak ve veri tabanına erişimi kolaylaştırmak için normalizasyon kuralları uygulandı. Bu kuralları şekilsel olarak göstermek gerekirse aşağıdaki akış şeması ortaya çıkar. Bunu ise

normalizasyon algoritması olarak kullanabiliriz.



Şekil 2.6. Normalizasyon Akış Şeması



Özet

• VERİ TABANLARI

•Veri tabanları çok sayıda güncel ve erişilebilen verinin saklandığı ve istenildiğinde de kullanıcılara veya sistemlere sunulduğu yazılımlardır.

•İLİŞKİSEL VERİ TABANI TASARIMI

•Veri tabanları zaman içerisinde farklı veri modellerine göre tasarlanmış olsalar da günümüzde en yaygın kullanım alanı bulan ilişkisel veri tabanlarıdır.

•Veri Tabanı Tasarımı

•Veri tabanı tasarımındaki ilk aşama varlık ilişki diyagramlarıdır.

•Bu diyagramlar sayesinde varlıkların verileri arasındaki ilişkiler ortaya konulduğundan veri tabanının mantıksal tasarımı da kolaylıkla yapılabilmektedir.

•İlişki türleri

•Varlık ilişki diyagramlarında üç tür ilişki bulunabilir. Bire-bir, Bire-çok veya çoğa-çok ilişki. Bu ilişki türüne göre mantıksal tasarım şekillenir.

•Veri tabanları tasarlanırken yazılımlarda kullanılacak veya saklanmak istenilen verilere göre tasarım yapılır.

•Veri tabanı tasarımında depolanacak veri türleri, kullanılacak tablo sayıları, tablolar arasındaki ilişkiler göz önüne alınarak tasarım yapılmalıdır.

•Bu tasarıma kavramsal tasarım denir ve veri tabanlarının kullanımında ortaya çıkabilecek eksiklik, tekrar ve veri kayıplarını en aza indirmek açısından önem taşır.

•Veri tabanlarında yer alan ve en önemli veri saklama ortamı olan tablolarda aynı özellikteki verilerin bir sütunda saklanmasına dikkat edilir.

•Tablolarda saklanan verilerde birbirleriyle ilişkili başka bir ifade ile aynı varlığa ait verilerin aynı satırda tutulması kuralı uygulanır.

•Bu nedenle de bu tür veri tabanlarına yapısal veri tabanları denir. Çünkü veriler belli bir yapıda saklanmakta ve kullanılmaktadır.

•Normalizasyon

•Yukarıda sayılan işlemler yapılsa dahi çoğunlukla veri tabanı tablolarında birçok tekrarlı veri olduğu görülebilir.

•Bu tür verilerin atılması verilerde anlam kaybına neden olacağından veriler atılamaz, ancak bu tür veriler farklı tablolarda saklanarak veri tekrarları ortadan kaldırılır.

•Tekrarlı veya aynı varlığa ait olup kaç kez tekrar edeceği belli olmayan türdeki verilerin veri tabanı mantığını bozmadan saklanmasına yönelik düzenlemeye normalizasyon denilmektedir.

•Normalizasyon da amaç veri bütünlüğünü sağlamak, uygulama bağımsızlığını oluşturmak ve veri erişiminde performans artışını sağlamak şeklinde de özetlenebilir.

•Normalizasyon kuralları tablolara uygulandığında tablolarda sorgulama ve güncellemelerde daha dikkatli olmak gerekir.

•Normalizasyon işlemlerinde fonksiyonel ve kısmi bağımlılığında göz önüne alınması gerekir.

•Normalizasyon işlemleri normal formlar şeklinde isimlendirilmektedir.

•Birden beşe kadar normal formlar vardı. Genellikle bunlardan ilk üçü uygulandıktan sonra diğerlerine ihtiyaç duyulmaz ve veri tabanında normalizasyon sağlanmış olunur.



Özet (devamı)

- **Birinci Normal Form**
- Birinci normal form ile verilerde yer alan tekrarlar ortadan kaldırılır.
- **İkinci Normal Form**
- İkinci normal form ile kısmi bağımlılıklar ortadan kaldırılır.
- **Üçüncü Normal Form**
- Üçüncü normal form ile dolaylı bağımlılıklar ortadan kaldırılır.
- **Dördüncü Normal Form**
- Çok karmaşık veri yapılarında ise dördüncü ve beşinci normal formlar uygulanabilir ancak genellikle ilk üç normal form ile ihtiyaç duyulan normalizasyon sağlanmış olmaktadır.
- Bir varlığa ait veriler birden çok tabloda saklandığından ilgili tüm tablolara da gerekli işlemlerin yapılmasında özen gösterilmelidir.

DEĞERLENDİRME SORULARI

1. Veri tabanı tasarlanırken aşağıdakilerden hangisi kavramsal tasarımda yapılıır?
 - a) Varlık İlişki Modeli
 - b) Birinci Normalizasyon Formu
 - c) Beşinci Normalizasyon Formu
 - d) Bire-Bir İlişki
 - e) Bire-Çok İlişki
2. Veri tabanındaki varlıkların özelliklerine ne denir?
 - a) İlişki
 - b) Nitelik
 - c) Etki Alanı
 - d) Varlık
 - e) Domain
3. Varlık ilişki modelinde varlıklar aşağıdakilerden hangisi ile gösterilir?
 - a) Elips
 - b) Kare
 - c) Silindir
 - d) Çizgi
 - e) Dikdörtgen
4. Bir dersi birden çok öğrencinin alması aşağıdaki varlık ilişkilerinden hangisidir?
 - a) Bire-bir
 - b) Çoğa-çok
 - c) Bire-çok
 - d) İkiye-bir
 - e) İkiye-iki
5. Varlık ilişki diyagramlarında anahtar nitelik nasıl gösterilir?
 - a) Kare içinde
 - b) Çift çizgili elips
 - c) Tek çizgili elips
 - d) Altı çizilerek
 - e) Koyu yazılarak
6. Veri tabanında veri bütünlüğünü sağlamak, uygulama bağımsızlığı ve performansını artırmak için aşağıdakilerden hangisi yapılmalıdır?
 - a) Varlık ilişki diyagramı
 - b) Tekilleştirme
 - c) Anahtar saha belirleme
 - d) Normalizasyon
 - e) Sıkıştırma

7. Anahtar olmayan sahaların birleştirilerek anahtar saha olarak kullanılması aşağıdakilerden hangisidir?
 - a) Fonksiyonel bağımlılık
 - b) Birincil anahtar saha
 - c) Kısmi bağımlılık
 - d) İkincil anahtar saha
 - e) Varlık ilişki diyagramı

8. Genel olarak tablolarda normalizasyon adımlarının hangisi yeterli olmaktadır?
 - a) 3NF
 - b) 2NF
 - c) 5NF
 - d) 4NF
 - e) 1NF

9. Birinci normal formda aşağıdaki işlemlerden hangisi yapılır?
 - a) Sütunları birleştirmek
 - b) Uygun sütunları anahtar alan yapmak
 - c) Tablolar arasında ilişkiler tanımlamak
 - d) Bir sütundaki farklı verileri farklı tabloya aktarmak
 - e) Bir tabloda tekrar eden verileri başka bir tabloya kaydetmek

10. İkinci normal formda aşağıdaki işlemlerden hangisi yapılır?
 - a) Sütunları birleştirmek
 - b) Uygun sütunları anahtar alan yapmak
 - c) Tablolar arasında ilişkiler tanımlamak
 - d) Bir sütundaki farklı verileri farklı tabloya aktarmak
 - e) Bir tabloda tekrar eden verileri başka bir tabloya kaydetmek

Cevap Anahtarı

1.a, 2.b, 3.e, 4.c, 5.d, 6.e, 7.c, 8.a, 9.d, 10.b

YARARLANILAN KAYNAKLAR

Kaya, Y. Ve Tekin, R. (2007). Veritabanı Uygulamaları. Papatya Yayınları, İstanbul.

Özseven, T. (2012). Veri Tabanı Yönetim Sistemleri-1. Murathan Yayınları,
Trabzon.