

## Hakemli Yazılar

## Refereed Articles

### Veri Tabanı Tasarımının Önemi ve Normalizasyon Süreci

#### The Importance of Database Design and Normalization Process

Tülay Fenerci\*

#### Öz

*Veri tabanı yönetim sistemleri kuruluşların işlemlerinin bilgisayar aracılığı ile yönetilmesini sağlayan önemli araçlardır. Ancak verimli kullanılabilmeleri için veri tabanı tablolarının iyi organize edilmesi gerekir. Tablo tasarımı yapılmadan veriler sisteme yüklenirse veri tekrarı ve yer kaybı gibi sorunlar yaşanabilir. Bu tür sorunların ortadan kaldırılması için tablolar üzerinde normalizasyon süreci uygulanır.*

*Normalizasyon özelliklerin bir araya getirilmesi ile oluşturulan ilişkilerin, rafine edilmesini sağlar. Dolayısıyla elde edilen taban tablolarda tekrar, gizli bağımlılık ve bileşik ilişkilerin bir arada gösterilmesinden kaynaklanan sorunlar çözümlenmiş olur. Bu açıdan veri tabanı yönetim sisteminin verimli kullanımında tasarım önemli bir etki unsurudur.*

#### Abstract

*Database management systems are major tools which provide management of organizations' operational data via computers. However their efficient usage depends on how well database tables are organized. When undesigned data is sto-*

\* Yrd. Doç. Dr. Tülay Fenerci, Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Kütüphanecilik Bölümü Öğretim Üyesidir.

*red in the database there probably occur some problems such as data repeating and waste of storage space. In order to prevent such problems a normalization process is applied to tables.*

*Normalization is a process of refining relations where relations are formed by gathering attributes. Hence problems such as repeating of data, dependency and combined relationships are solved with normalized base tables.*

*Therefore database design is an important fact in efficiency of database management systems.*

## **Anahtar Kelimeler**

*Veri Tabanı Tasarımı, Normalizasyon*

## **Keywords**

*Database Design, Normalization*

## **Giriş**

Günümüzün evrensel gerçeği, enformasyon teknolojisinin insan yaşamının her noktasına girdiği ve onu çepeçevre kuşattığıdır. Birçok toplum için öncelikli hedef, enformasyon teknolojisi ürünlerinin tam kapasite ile yaygın kullanımını sağlamak ve ürünlerden en üst düzeyde yararlanmaktır. Dolayısıyla enformasyon teknolojisi, bilginin bireysel ve kurumsal düzeyde üretim-tüketim sürecini yaşadığı toplumlarda bir altyapı unsuru olarak yerini almıştır. Söz konusu toplumlar kurumsal yaşamda iyi yönetim, doğru karar verme, nitelikli mal ve hizmet üretimi, verimliliği artırma gibi konularda enformasyon teknolojisinin rolünü kavramışlardır. Özellikle kar, düşük maliyet ve uzun vadeli rekabet gücü gibi hedeflere ulaşma ve başarı elde etmek için etkinliklerini enformasyon teknolojisine dayandırma çabaları bunun en önemli göstergesidir.

Bu açıdan bakıldığında tüm kuruluşlar için ön plana çıkan ürünlerden biri, işlemlerin bilgisayar ortamında daha hızlı ve doğru yapılmasını sağlayan, farklı mekanlardaki veya coğrafi bölgelerdeki kullanıcılarla bilgi paylaşımına olanak veren yazılımlardır. Sözü edilen yazılım veri tabanı yönetim sistemi(VTYS).dir.

Veri tabanı yönetim sistemi tüm kuruluş için gereksinilen verileri işleyen ve kullanıcıların işlemleri yürütmelerinin yanı sıra sorunlarına yanıt arayan bir sistemdir. Benzer bir tanım, ilişkisel veri tabanları konusundaki uzmanlığı ile tanınan ve IBM'in ilişkisel ürünleri(SQL/DS veDB2) için teknik planlama görevini yürütmüş olan J.C. Date tarafından verilmektedir. O'na göre veri tabanı özellikle önceden tahmin edilemeyen sorular (ad hoc queries) için düzenlenmiş bir veri

topluluğu olup sorgulama, veri güvenliği ve onarım desteği veren veri tabanı yönetim sistemi ile çalışır(Koffman,1992:522). Bir başka tanıma göre veri tabanı yönetim sistemi birçok kullanıcının paylaşacağı düşünülerek tasarlanmış standart formatta depolanmış olan veri topluluğu(veri tabanı)nu tanımlayan, sorgulayan, rapor üreten ve veri giriş ekranları yaratan bir yazılımdır (Post,1999:3). Dolayısıyla VTYS kuruluş içinde veriyi işleme ve gereksinimler doğrultusunda bilgi üretme işlevi olan önemli bir üründür.

Bundan dokuz yıl önce kendisi ile yapılan bir görüşmede Date'in "2000'li yıllarda veri tabanlarında radikal değişiklikler olacak mı?" yönündeki soruya "...birkaç yıl içinde veri tabanları daha büyük ve hızlı olacaklar. Karşılıklı iletişim kapasiteleri artacak. Formatlı kayıtlar, yapılandırılmış metin, resim, ses verisi gibi her tür farklı veriyi bir arada tutabilecekler. Etkileşimli video arayüzleri bir olasılık..." biçiminde verdiği yanıtta bir öngörü olarak sunduğu görüşler, günümüzde VTYS'lerinin kazandığı yeni beceriler olarak karşımızda durmaktadır(Koffman,1992:523).

Ancak VTYS'lerinin işlerliği ve verimli kullanımı, sistem için gereksinilen verilerin doğru ve eksiksiz olarak saptanması ve VTYS'nin verilerini sağlayacağı taban tablolarının iyi organize edilerek ilişkilendirilmesi ile mümkündür. Bunun da ötesinde veri tekrarını ve buna bağlı olarak veri tutarsızlığını olabildiğince önlemek ve veri silme, ekleme, değiştirme gibi işlemlerde karşılaşılabilecek sorunları ortadan kaldırmak için çeşitli teknikler (örneğin normalizasyon) kullanılabilir.

Bu çalışmada amaç normalizasyon tekniğini bir örnek yardımıyla açıklamaktır. Dayanılan temel düşünce ise mesleklerinin gereği olarak materyal tanımlama ve ortak konular aracılığı ile ilişkilendirme işlevi olan kataloglama ve sınıflama gibi temel birtakım işleri yerine getiren kütüphanecilerin söz konusu işlemlerde izlenen mantık ve düşünce sistemini yani bilgiyi organize etme sürecini tasarım sürecine uygulayabileceğidir.

Bu doğrultuda çalışmada, veri tabanlarının yapılandırılma biçimini yansıtan veri modelleri gelişim ve özellikleri ile kısaca ele alınmış ve normalizasyon süreci üniversite kütüphanelerinde sipariş işlemine dayalı bir örnekle açıklanmaya çalışılmıştır.

Çalışmada sipariş işlemi için kullanılan formlar incelenmiş, farklılıklar göz ardı edilerek genel bir yaklaşım benimsenmiştir.

## Veri Modelleri

Veri tabanı yönetim sistemlerinin verimli kullanımı daha önce de belirtildiği gibi sistem için gereksinilen verilerin doğru ve eksiksiz olarak saptanması ile mümkündür. Anılan amaçları gerçekleştirebilecek bir sistem tasarımı yapabilmek için

öncelikle şu iki soruyu yanıtlamak gerekir. Kullanıcılar kimdir? Bir enformasyon sistemi onlara nasıl yardım edebilir? Doğal olarak tasarımın ilk adımı kullanıcıları tanımlamak ve gereksinimlerini saptamaktır. Bu konuda yeterli bilgiyi toplamak için tasarımcı kullanıcılarla görüşme yapmalıdır. Bir sonraki aşama, yaratılacak olan enformasyon sistemi için işin gereklerini bütünüyle anlamaktır. Tasarımcı işletmedeki çalışmaları gözleyerek ve işlemlerle ilgili bilgi sağlayan rapor, form gibi dokümanları derleyerek tasarım için gereken verileri toplamaya çalışır. Ön tasarım için gereksinilen bilgi üç temel başlık altında gruplandırılabilir:

- Toplanması gereken veri;
- Veri tipi;
- Veri miktarı.

Söz konusu bilgiler verilerin bir modelini yaratmak için kullanılır. Veri tabanı verilerinin organizasyonu ve yönetimi konusunda birçok yaklaşım vardır. Bunlar veri modelleridir.

**Veri modeli** gerçek bir dünya sisteminin basitleştirilmiş bir özü olup aynı zamanda veri tabanının yapılandırılma biçimini yansıtır. Veri modeli bir kuruluştaki varlıklar, olaylar, etkinlikler ve bunlar arasındaki ilişkilerin soyut temsilidir. Dolayısıyla kuruluşun kendisini temsil eder (McFadden and Hoffer, 1985: 165).

Veri modellerini şu üç ana grupta toplamak olasıdır:

**1-Nesneye dayalı mantıksal modeller:** Kavramsal ve görünüm düzeylerinde veri betimlemesi için kullanılırlar. Örneğin: Varlık-ilişki modeli, nesne uyarlı model ve semantik veri modeli (Korth and Silberschatz, 1986: 6).

**2-Kayda dayalı mantıksal modeller:** Çeşitli tipteki sabit formatlı kayıtlar olarak kavramsal ve görünüm düzeylerinde veri betimlemesinde kullanılırlar. Örneğin: Hiyerarşik, ağ ve ilişki model (Korth and Silberschatz, 1986: 8).

**3-Fiziksel veri modelleri:** Veriyi en alt düzeyde temsil eden modellerdir. Az sayıda olmalarına karşın en bilinenleri birleştirici (unifying) model ve çerçeve bellek (frame memory) modelidir. (Korth and Silberschatz, 1986:10) ,

Veri modellerinin tarihsel gelişimi incelendiğinde temel hedefin, gerçek dünya sisteminin bir model aracılığı ile bilgisayar ortamında temsil edilebilmesi ve bu amaç doğrultusunda semantik unsurun modelde yansıtılabilmesi olduğu görülmüştür. Programlama dilleri ve veri tanımlama biçimlerindeki gelişmeler de belirtilen amacı destekler yönde olmuş ve yeni veri modellerinin ortaya çıkışına öncülük etmiştir. Örneğin: 1960'lı yıllarda CODASYL (Conference On DATA Systems and Languages) tarafından geliştirilen COBOL, VTYS'lerinin öncüleri olan ve sıralama, güncelleme ve rapor üretme gibi ortak birtakım işlemleri dosyaların verilerinden bağımsız olarak gerçekleştirebilen dosya yönetim sistemlerine oranla veri tabanı sistemlerinde verilerin daha iyi temsil edilebilmesini sağlamıştır. Daha sonraki yıllarda CODASYL'in atamış olduğu DBTG (Data Base Task Group) tarafından veri tabanlarının niteliklerine uygun olarak geliştirilen COBOL'da veri tanımlama ve veri yönetim dilleri tanımlanmış ve bu gelişim ağ veri modelinin

oluşumuna yol açmıştır(Khoshafian and Abnous,1990: 17).

Ağ veri modeli varlıkların, kayıt tipleri ve aralarındaki ilişkilerin kümeler ile ifade edilmesi esasına dayanır. Böylelikle küme aracılığı ile birden fazla kayıt tipine erişim olanağı yaratılmıştır. Bu modele dayalı olarak üretilen kimi ürünler IDMS (Cullinet,1970), IDS-2 (Honeywell Information Systems,1975) gibidir(Khoshafian and Abnous, 1990: 17; ayr.bkz. Fenerci, 1996: 59-64).

Benzer bir oluşum 1970'li yıllarda yaşanmıştır. E.F.Codd tarafından ortaya atılan ve 1990'lı yıllara değin popülerliğini sürdüren ilişkisel model, erişim yoluna bağımlı olan önceki modellere esneklik getirmiştir.

İlişkisel modelde varlıklar tablo veri yapısıyla ifade edilir. Her tablo bir kayıt tipini, tablonun sıraları ise kayıt tipinin elemanları(kayıtlar)nı temsil eder. Varlıklar arası ilişkiler ortak alanlar aracılığı ile gerçekleştirilmektedir. Ancak ilişkisel modeli daha anlaşılır ve kullanıcı uyarlı kılan en önemli özellik SQL(Structured Query Language)'in getirdiği kolaylıklardır. Bu sayede kullanıcı, veriye **nasıl** ulaşacağı yerine **hangi** veriye ulaşacağı üzerine odaklanabilmektedir. Bu modelle dayalı ticari ürünlere System/R, DB2 (IBM), Ingres (Relational Technologies) ve Oracle(Oracle Cooperation) örnek verilebilir(Khoshafian and Abnous,1990: 21).

Öte yandan 1980'li yıllara doğru semantik unsur ağır basmış ve yaygınlıkla kullanılan veri modellerinde semantik unsurun yansıtılabilmesi için alternatif bir takım modeller üretilmeye başlanmıştır. Abrial'in semantik veri modeli, Chen'in varlık-ilişki modeli, Kroenke'nin semantik nesne modeli ve Nijssen'in NI-AM(Nijssen's Information Analysis Methodology) bu tür modellere örnek gösterilebilir (Semantik veri modelleri için ayr.bkz. Khoshafian and Abnous,1990: 264-268; Betra,1997: 3). Bu modeller daha çok laboratuvar çalışmaları olarak kalmalarına karşılık özellikle ilişkisel modele semantik unsurun aktarımında birer tasarım aracı olarak işlev görmüş ve daha önemlisi günümüzde popülerliğini sürdüren nesne uyarlı modelin öncülüğünü yapmıştır. Nesne uyarlı model, nesne uyarlı programlamaya uygun hale getirilmiş bir ilişkisel model olarak tanımlanabilir. Model, nesne uyarlı varlıklar ve varlıklar üzerine tanımlanmış yöntemlerden oluşur. Model birçok yönden ilişkisel modele benzemektedir. Örneğin: İlişkisel modeldeki varlık/ilişki/tablo, özellik ve özelliklerin tanım kümesi nesne uyarlı modelde sırasıyla sınıf, sınıf değişkenleri ve değişken sınırlılıklarına karşılık gelir (McFadden and Hoffer, 1985: 273). Nesne uyarlı modelin ortaya çıkış ve gelişimi nesne uyarlı programlama dilleri(C++, Smaltalk gibi)ndeki gelişmeler ile paralellik gösterir. GemStone (Servio), Gbase(Graphael) ve Vbase (Ontologic) gibi ticari ürünlerle piyasaya çıkmış ve yaygın kullanım alanı bulmuştur. (Khoshafian and Abnous,1990: 21). ticari nesne uyarlı veri tabanı sistemleri için ayr.bkz.age.270-273).

## Normalizasyon

Bir mantıksal tasarımda, veri tabanında yer alacak varlıklar, varlıklara ait özellikler ve varlıklar arası ilişkiler belirtilir. Amaç, gereksinilen bilginin bütünleşik olarak temsil edilebilmesidir. Özellikle ilişkisel veri tabanları için birçok mantıksal tasarım yöntemi önerilmiştir. Örneğin: Nijssen'nin enformasyon analiz yöntemi, Chen'in varlık-ilişki modeli mantıksal tasarımın gerçekleştirilmesinde ve ilişkisel modelin tablolarına dönüştürülmesinde yararlanılan tasarım yöntemleridir (Batra,1997: 2). Normalizasyon ise özellikle ilişkisel veri tabanının mantıksal tasarımında yaygın olarak kullanılan bir tekniktir (Nijssen and Halpin, 1989: 273).

Normalizasyon güçlü ve iyi yapılandırılmış bir veri tabanı yaratmada veya hangi veri modeli kullanılmış olursa olsun (Kroenke and Dolan,1990: 132) tasarım sonrası denetlemeyi sağlamada yararlanılan bir süreç (Khan,1995:155) bir ilkel kümesi (Batra,1997: 2) veya yardımcı bir araç olarak da düşünülebilir.

Normalizasyonun amacı şöyle özetlenebilir:

- Veri tabanını bütünlüğünü artırmak;
- Tekrar ve tutarsızlığı en aza indirmek;
- İstem dışı veri silinmesini önlemek;
- İstenen bilginin temsil edilebilirliğini artırmak(Fenerci,1995: 103)

İlişkilerin/tabloların kullanıcı gereksinimlerini eksiksiz ve yeterli bir biçimde yansıtması, iyi yapılandırıldığı anlamına gelmez. İyi bir veri tabanı tasarımının başlıca amacı veri tekrarını önlemektir. Bu amacı gerçekleştirmek için bilgi, daha küçük tablolara bölünür. Böylelikle **tek bir yerde, tek bir gerçek** ilkesi korunur(Riccardi,1994: 285) Bunun yanı sıra veri tekrarını önleyerek veri tutarsızlığı engellenir ve depolama alanında tasarruf sağlanmış olur.

Tabloların ayrıştırılması aynı zamanda tabloların rafine edilerek güncelleme anormalliklerinden arındırılması anlamına gelir. **Güncelleme anormalliği**, bir güncelleme işlemi (veri girişi, silme, değiştirme) gerçekleştiğinde ortaya çıkan problemdir (Nijssen and Halpin,1989: 274). Bu tür problemler, genellikle taban tablolardaki veri tekrarından kaynaklanır. Temel işlevi veri tekrarını önleyerek güncellemeden kaynaklanan anormallikleri ortadan kaldırmak olan normalizasyon için tabloların daha küçük tablolara ayrıştırılmasını sağlamak amacıyla bir takım kurallar ortaya konmuştur. Her bir kural, normalizasyonda bir aşamayı belirtir. Aşamalar genelde **normal biçim** olarak ifade edilir. Normalizasyona ilişkin ilk üç kural(1, 2 ve 3. normal biçimler) 1970 yılında E.F.Codd tarafından belirlenmiştir. Daha sonra Codd, Boyce ile Boyce-Codd normal biçimi belirlemiş ve bunu 4. ve 5. normal biçimler izlemiştir. 1981'de ise Fagin Tanım Kümesi Anahar Kelime Normal Biçim (TKAKNB) olarak adlandırdığı yeni bir normal biçim ortaya atmıştır(Kroenke and Dolan,1990:137). Daha önce de belirtildiği gibi veri tekrarını önlemek ve dolayısıyla güncellemeden kaynaklanan anormal durumlardan arınmak için tablolar yeniden yapılandırılır. Yeniden yapılandırma genel-

likle tabloların daha küçük tablolara bölünmesi yani belli ilkeler doğrultusunda tabloların ayrıştırılması ile gerçekleşir. Ancak normalizasyonu anlayabilmek için öncelikle fonksiyonel bağımlılık, anahtar, ilişki anahtarı ve tablolar arası ilişkiler (tekliden tekliye, tekliden çokluya ve çokludan çokluya) anlamak gerekir.

**Fonksiyonel bağımlılık:** Özellik/alan değerleri arasındaki ilişkidir. Bir özellik değeri bir başka özellik değerini bilmemizi sağlıyorsa fonksiyonel bağımlılıktan söz edilebilir (Kroenke and Dolan, 1990: 138).

**Anahtar:** Bir tablonun bir sırasını (bir kaydı) tek başına tanımlayan bir veya birkaç özelliktir. Her ilişkinin/tablonun en az bir anahtarı vardır. Genel olarak anahtar bir veya birkaç özelliğin fonksiyonel olarak anahtar olmayan özellikleri belirlemesidir (Kroenke and Dolan, 1990: 139).

**İlişki anahtarı:** Tablolar arası ilişkilerin kurulması ortak alanlar aracılığı ile olur. Ancak alanların, yer aldıkları tablolardaki işlevleri farklıdır. Bir tabloda anahtar (primary key) olan özellik/alan ve doğrudan o tablonun bir sırasına erişmeyi sağlarken diğer bir tabloda ilişki anahtarı (foreign key) olarak tanımlanır. Ortak alanın ismi farklı olsa da veri yapıları ve değer aralıkları aynı olmak zorundadır.

**İlişkiler:** Tablolar arası ortak alanlar ile kurulan ilişkiler veri tabanında veri tekrarı olup olmadığını işaret eder. Tablolar arası ilişkiler tekliden tekliye, tekliden çokluya ve çokludan çokluya biçiminde kendini gösterir.

Tekliden tekliye ilişkide bir tabloda yer alan bir alan değerine karşılık diğer tabloda da tek bir değer yer almasıdır.

Tekliden çokluya ilişkide bir tabloda yer alan bir alan değerine karşılık diğer bir tabloda birden fazla değer gözükür. Bu durumda tekliliğin olduğu tablodaki anahtar çoklu tabloya ilişki anahtarı olarak yerleştirilir.

Çokludan çokluya ilişkilerde ise ilişkinin kurulabilmesi için ortak alanlardan oluşan üçüncü bir tablo yaratma ihtiyacı vardır. Bu yolla tablolar arası ilişki tekliden çokluya biçimine dönüştürülür (Riccardi, 1994: 285).

## Normalizasyon Süreci ve Bir Örnek

Çalışmanın bu bölümünde normalizasyon süreci, üniversite kütüphanelerinde gerçekleştirilen kitap sipariş işlemi üzerine uygulanmıştır. Sipariş işlemi için gereksinilen bilgi, formlar aracılığı ile elde edilmiştir.

Bu bağlamda bir sipariş işleminin gerçekleştirilebilmesi için genel hatlarıyla şu bilgilere gereksinim vardır:

Üniversite kütüphanesi satın alma öncesi fakülte ve bölümlerde görevli öğretim üyelerine gereksinim duydukları yayınların listesini hazırlayıp kütüphaneye bildirmelerini ister. Listelerde isteğin kimden geldiği, gereken durumlarda iletişim kurulabilmesi için kişinin telefon numarası, e-posta adresi vb., isteğin hangi

bölümden yapıldığı ve istenen yayına ilişkin kimlik bilgileri(yazar adı, eser adı, yayın tarihi, bası, yayınlayan gibi genel bilgiler)nin yer alması beklenir.

Kütüphane derlediği listeleri inceleyerek dublikasyonları saptar, yayınlara ilişkin eksik bilgileri tamamlar ve bir siparişteki listesi oluşturur. Siparişteki listesinde her yayına ilişkin kimlik bilgisinin yanı sıra eserin sipariş no'su, sipariş edildiği firma, hangi tarihte sipariş edildiği ve kim tarafından sipariş edildiği bilgileri yer alır. Ayrıca kütüphane satın alma işinde işbirliği yaptığı firma bilgilerini tuttuğu bir dosya da oluşturmak durumundadır. Bu dosyada firmaya ilişkin ad, adres, kiminle kontak kurulacağı ve firma ile ilgili özel düşünceler gibi bilgiler tutulur. Öte yandan kütüphane, bütçe dengelerini korumak için ne kadar ödenek ayırabileceği, satın alma harcamalarının tutarı, ödemenin zamanında yapılıp yapılmadığı, ödemelerin hangi firmalara yapılacağı bilgilerine sahip olmak durumundadır.

Derlenen bilgi rehberliğinde **sipariş istek formu, firma ve firmadan gelen fatura** olmak üzere üç temel varlık saptanmıştır. Ancak burada öncelikle şunu belirtmek gerekir ki başlangıçta veri tabanında yer alacak tabloların tümünü eksiksiz saptamak her zaman mümkün olmayabilir. Bir sonraki adım varlıklara ait işlemin gerektirdiği özelliklerin belirlenmesi ve ilişki modelin tabloları biçiminde ifade edilmesidir. Başlangıç tabloları ve özellikleri şöyle sıralanabilir:

**Sipariş İstek Formu** (**kişino**, ad, soyad, unvan, bölüm, tel, e-posta, **birimno**, eser adı, yazar adı, yayın yeri, yayınlayan, yayın tarihi, basım, cilt, ISBN, fiyat, adet)

**Sipariş Edilen Firma** (**kod**, ad, adres, tel, e-posta,, faks, teslim süresi, indirim oranı, kontak kişi)

**Firmadan Gelen Fatura** (**faturano**, **firkamod**, gönderilme tarihi, geliş tarihi, ödeme tarihi, tutar, ödenen fon)

Son aşama yapılandırmanın niteliğini ölçmek için normalizasyon sürecinin uygulanmasıdır. Bir normalizasyon sürecinde izlenecek adımlar, sırasıyla şöyledir:

- Formların değerlendirilmesi;
- Özelliklerin belirlenmesi;
- Tekrarlayan grupların bulunması;
- Potansiyel anahtarların bulunması;
- Hesaplanabilir alanların bulunması(Post,1999: 64).

Formların değerlendirilmesi ve özelliklerin belirlenmesi aşamaları gerçekleştirilmiştir. Bu bağlamda yukarıda verilen başlangıç tabloları üzerine normalizasyonun daha önce belirtilen kuralları uygulanarak aşama aşama tablolar rafine edilecektir.

**Birinci Normal Biçim(1NB):** Veri tabanında tekrarlayan alanların, taban tablolar ile birlikte depolanması taban verinin her giriş için tekrarlanmasına neden olur. Örneğin: Sipariş İstek Formu tablosunda kişi, istekte bulunduğu her yayın



için kimlik bilgilerini tekrarlamak durumundadır. Ayrıca kitap ve kişi bilgileri bir arada verildiğinden birine ilişkin veri silindiğinde diğerine ait veriler de silinir ve bilgi kaybı meydana gelir. Bu nedenle tablo ayrıştırılarak tekrarlayan gruplar ortadan kaldırılmalı ve tablo, (diğer tablolar gibi) 1NB'e getirilmelidir. Ancak ayrıştırmada dikkat edilmesi gereken konu, anahtarın diğer tablolara da taşınmasıdır. Bu durumda yeni tablolar şöyle olacaktır (Post, 1999: 74).

**A-SiparişEdenKişi** (**kişino**, ad, soyad, unvan, bölüm, e-posta, tel)

**B-SiparişEdilenKitap** (**kişino**, **birimno**, eser adı, yazar adı, yayın yeri, yayınlayan, yayın tarihi, basım, cilt, ISBN, fiyat, adet)

**C-SiparişEdilenFirma** (**kod**, ad, adres, tel, e-posta., faks, teslim süresi, indirim oranı, kontak kişi)

**D-FirmadanGelenFatura** (**faturano**, **firkamakod**, gönderilme tarihi, geliş tarihi, ödeme tarihi, tutar, ödenen fon)

**İkinci Normal Biçim(2NB)**: Tablo içinde yer alan özelliklerin anahtarın tümüne bağımlı olması gerekir. Eğer bu koşul sağlanıyorsa tablo 2NB'dedir. Özellikle bileşik anahtarlar için geçerlidir (Kroenke and Dolan, 1990: 142).

A tablosunda bütün alanlar anahtara bağımlıdır. B tablosunda eserle ilgili alanlar fonksiyonel olarak kişino'ya bağımlı değildir. Bu nedenle tablo ayrıştırılmalı ve anahtarın bir kısmına bağımlı olan alanlar, bağımlı oldukları anahtar ile başka bir tabloya taşınmalıdır.

**B1. Kişi-Birim** (**kişino**, **birimno**)

**B2.Birim** (**birimno**, eser adı, yazar adı, yayın yeri, yayınlayan, yayın tarihi, basım, cilt, ISBN, fiyat, adet,)

D tablosunda anahtar dışındaki tüm alanların anahtarın bir kısmını oluşturan faturano'ya bağımlı olduğu görülür. Bu durumda yeni tablolar:

**D1.Firma-Fatura** (**faturano**, **firkamakod**)

**D2.Fatura** (**faturano**, gönderilme tarihi, geliş tarihi, ödeme tarihi, tutar, ödenen fon)

Bu aşamaya kadar elde edilen tablolar gözden geçirilecek olursa, sipariş işleminin gereksindiği bilgileri veren bir tablonun henüz yaratılmamış olduğu görülür. Yeni tablo şöyle olmalıdır.

**E-Sipariş** (**sipno**, siptarihi, adet, **birimno**, fiyat, ISBN, **firkamakod**, geliş tarihi, statü)

Yeni tabloya yukarıda kuralları açıklanan 1NB ve 2NB uygulandığında ortaya çıkan yeni tablolar şöyle olacaktır:

**E1.Birim-Sipariş** (**birimno**, **sipno**, adet, birimfiyat, ISBN) B2 ve E1 tablolarında tekrar edilen adet, fiyat, ISBN gibi ortak alanlar veri tekrarını önlemek amacıyla ve özelliklerin sipariş edilen birime ait olduğunu gösterebilmek için E1 tablosuna taşınmıştır.

**E2.Sipariş** (**sipno**, siptarihi, geliş tarihi, statü)

**E3.Sipariş-Firma** (**sipno**, **firkamakod**).

**D2** tablosunda yer alan fon alanı sıklıkla tekrarlayacak bir alandır. Bu nedenle ayrı bir tabloda fon bilgilerinin tutmak daha akılcıdır. Böylelikle fon bilgisi bir kez taban tablosunda yer alacak ve fatura tablosu fon bilgisini ait olduğu tablodan kullanacaktır.

**F.Ödenecek fon (kod, isim, miktar)**

**Üçüncü Normal Biçim (3NB):** İşe bağlı olarak ilişkilerin ortaya konduğu aşamadır. Bir tablonun 3NB’de olabilmesi için bütün alanların anahtara bağımlı olması ve geçişli bağımlılığın bulunmaması gerekir (Kroenke and Dolan, 1990: 143). Bu durumda tabloların her biri 3NB’dedir.

**A-SiparişEdenKişi (kişino, ad, soyad, unvan, bölüm, e-posta, tel)**

**B1. Kişi-Birim (kişino, birimno)**

**B2.Birim (birimno, eser adı, yazar adı, yayın yeri, yayınlayan, yayın tarihi, basım, cilt, ISBN, fiyat, adet)**

**C-SiparişEdilenFirma (kod, ad, adres, tel, e-posta, faks, teslim süresi, indirim oranı, kontak kişi)**

**D1.Firma-Fatura (faturano, firmakod)**

**D2.Fatura (faturano, gönderilme tarihi, geliş tarihi, ödeme tarihi, tutar, ödenen fon)**

**E1.Birim-Sipariş (birimno, sipno, adet, birimfiyat, ISBN)**

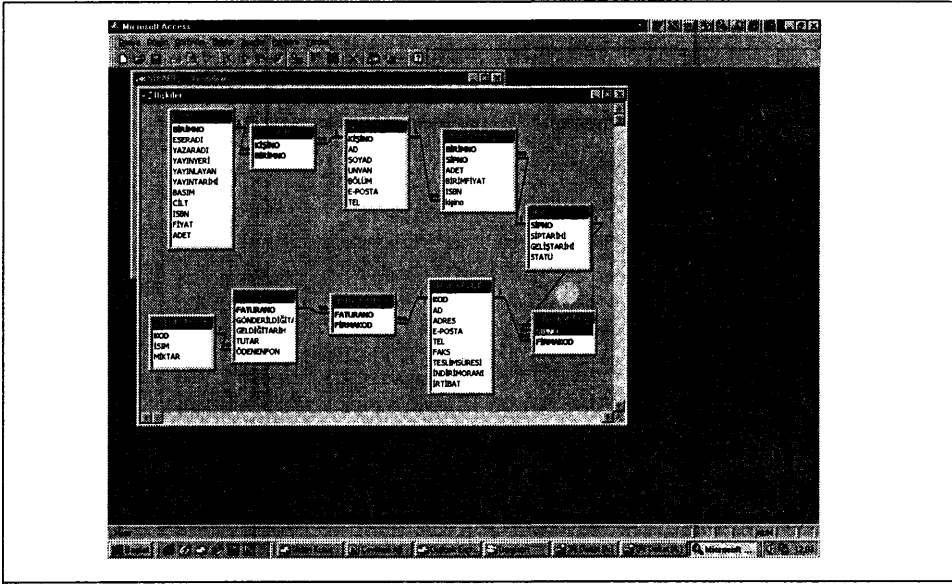
**E2.Sipariş (sipno, siptarihi, geliş tarihi, statü)**

**E3.Sipariş-Firma (sipno, firmakod).**

**Boyce-Codd Normal Biçim (BCNB):** Bir tablodaki sütunlar arasında ikincil bir ilişki varsa dublikasyon ve veri kaybı gibi sorunlar ortaya çıkar (Post,1999; 85). Bu durum genellikle belirleyici olan özelliğin aday anahtar olmamasından kaynaklanır. Örneğin: Bir yayının birden fazla firmaya sipariş edilmesi söz konusuysa birimno, sipno ve firmakod alanlarından oluşacak T gibi bir tabloda böyle bir sorun ortaya çıkacaktır. Çünkü aday anahtarlar olan (birimno, sipno)→firmakod’unu ve (birimno, firmakod)→sipno’yu belirlemesine rağmen determinant olan firmakod aday anahtar değildir ve bu nedenle tablo BCNB’de bulunmamaktadır. Çözüm, tablonun T1 (birimno, sipno) ve T2 (birimno, firmakod) biçiminde ayrıştırılmasındadır. Böylelikle tablo BCNB’e gelir.

**Dördüncü Normal Biçim (4NB):** Bir tablo içerisinde iki ayrı ilişki olduğunda ve tasarımcı bunu bileşik bir ilişki gibi gösterdiğinde sorun ortaya çıkar. Bu durum genellikle en az üç özelliği olan tablolarda karşımıza çıkar ve çok değerli bağımlılık olarak adlandırılır. Örneğin:T(a, b, c) gibi bir tabloda eğer, a çoklu b değeri veriyorsa, a→çoklu c değeri veriyorsa ve eğer, b ve c birbirinden bağımsız ise çok değerli bağımlılık vardır. Bu sorun giderildiğinde tablo 4NB’de olur (Kroenke and Dolan,1990: 147). Tablolarımız bu açıdan incelendiğinde 4NB sorunlarının bulunmadığı görülür.. Tablolar otomatik olarak 4NB’dedir.

Tabloların normalize edilmiş ilişkileri Access 97 yazılımında şöyledir:



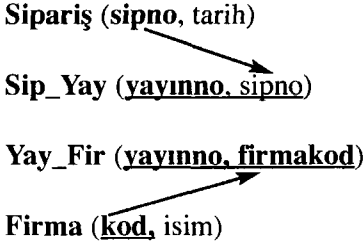
**Tanım Kümesi Anahtar Kelime Normal Biçim (TKAKNB):** Normalizasyon sürecine 1981’de Fagin tarafından eklenen bu adım, bazı tabloların hiçbir zaman TKAKNB’de dönüştürülemez olmasına rağmen, uygulama geliştirenler için önem taşır. Tabloların TKAKNB’de olduğunu anlamak için işe 3NB ile başlanır. Tüm iş kuralları listesine sahip olup olunmadığı kontrol edilir ve kuralların tanım kümesi (özelliklerin alabileceği değerler) sınırlılıkları ve anahtar ilişkileri ile sağlanıp sağlanmadığına bakılır. Anahtarların teklifi ve tüm çokludan çokluya ilişkilerin saptanıp saptanmadığı denetlenir. Gizli kural ya da bağımlılıkların elimine edildiğinden emin olunur. İlişki anahtarları konularak var olan kurallar güçlendirilir ve diğer tablo verileri ile uyumu sağlanır (Kroenke and Dolan,1990:149-150).

Kitapların gruplar hakkında tek bir sipariş no’su ile bir firmadan istenmesi halinde ve belirli bir kitabın hangi tarihte hangi firmaya sipariş edildiği bilgisine gereksinim duyulduğunda tablolar şöyle olabilir:

**Sipariş** (sipno, tarih, yayıno, firmakod)

**Firma** (kod, isim)

Sipno bir grup yayını temsil ettiğinden teklifi sağlamıyor; firma, ancak belli oranda alım yapan kütüphanelere indirim uyguluyor. Bu sınırlılıklar doğrultusunda yeni tablolar TKAKNB’de göre şöyle olacaktır:



## Sonuç

Bir VTYS, gereksinilen bilginin yapılandırılmış olarak tutulduğu veri tabanını kullanarak, bir kuruluşun işlemlerini hızlı, kolay ve doğru bir biçimde gerçekleştirir ve gereksinilen bilgileri zamanında üretebilir. Ancak VTYS'nin verimli çalışabilmesi verilerini sağladığı veri tabanının iyi yapılandırılmasına bağlıdır. Veri tabanını oluşturan taban tablolarında veri tekrarının bulunması güncellemede problem yaratır ve dolayısıyla veri tutarsızlığına neden olur. Bunun yanı sıra depolama ortamında yer sorunu yaratır. Sözü edilen sorunlardan kaçınmak için geliştirilen yöntemlerden biri normalizasyondur. Normalizasyon, bir tabloda tek bir ilişkinin ifade edilmesini amaçlar. Bu doğrultuda tablolarda alanlar arası fonksiyonel bağımlılıkların varlığını saptamaya çalışır. Her bir tablonun doğrudan teklisi sağlayan ve diğer bütün alanların kendisine bağımlı olduğu bir anahtar özelliğinin bulunması ilkesi ve tablolar arası bağlantıların ortak özellikler aracılığı ile kurulması ve tablolar arasındaki çokludan çokluya ilişkilerin tekliden çokluya ilişkilere dönüştürülmesi esasına dayanır.

Bu bağlamda çalışmada seçilen “üniversite kütüphanelerinde kitap siparişi” örneği üzerinde normalizasyonun aşamalarını belirleyen kurallar uygulanmıştır. Başlangıç tabloları öncelikle tekrarlayan gruptan arındırılmış, alanlar arası fonksiyonel bağımlılıklar saptanmış ve tabloların tek bir anahtara ya da anahtarın tümüne bağımlı alanlardan oluşması sağlanmıştır. Tek bir ilişkiyi ifade eden taban tabloların yanı sıra tablolar arası ilişkilerin ortak alanlar uyarınca ve tekliden çokluya biçiminde kurulmasını sağlayan ara tablolar yaratılmıştır. Böylelikle istenen verinin tam olarak temsil edilmesi gerçekleştirilmiş ve bilgi kaybına neden olabilecek anormal durumlar ortadan kaldırılmıştır. Daha önce de vurgulandığı gibi VTYS'nden en üst düzeyde yararlanma ancak iyi yapılandırılmış taban tabloları ile mümkündür. Bu bakımdan tasarım çok büyük önem taşımaktadır. Aksi halde VTYS çözüm üreten değil sorun yaratan bir teknolojik ürün haline dönüşebilir.

**KAYNAKÇA**

- Batra, Dinesh.(1997). "A method for easing normalization of user views", *Journal of Management Information Systems*,14(1): 215-234
- Fenerci,Tülay.(1995).*Veri tabanı kuramı ve bir kütüphane uygulaması*.(Yayımlanmamış doktora tezi) Ankara: A.Ü. Dil ve Tarih Coğrafya Fakültesi Kütüphanecilik Bölümü
- Khan, Philippe.(1995). "Better database design", *Windows Magazine* 2:155-161
- Khoshafian, Setrag and Razmik Abnous.(1990) *Object orientation: Concepts,languages, databases, user interfaces*.New York: John Wiley.
- Koffman, Elliot B.(1992).*Pascal; problem solving and program design*. 4th ed. Reading ,Massachusetts: Addison-Wesley
- Kroenke, David M.and Kathleen A. Dolan.(1990). *Database processing; fundamentals, design, implementation*. 3<sup>rd</sup> ed. Chicago: Science Research Associates
- McFadden, Fred and Jeffrey A. Hoffer.(1985). *Database management*. California: The Benjamin Commings Pub. Com.
- Nijssen,G.M. and T.A. Halpin.(1989). *Conceptual schema and relational database design; a fact oriented approach*. New York: Prentice Hall.
- Post, Gerald V.(1999). *Database management systems; designing and building business applications*. Boston: Irwin McGrow Hill..
- Ricciardi, Sal.(1994). "Database design; redundancy and normalization", *PC Magazine*,13(2): 285-287.