4. FONKSİYONEL BAĞIMLILIK

Birlikte Düşünelim

Bir nitelikte yer alan bilgi, dolaylı olarak başka nitelik altında da bulunuyor olabilir mi?

Aynı veriyi farklı biçimlerde veri tabanı içerisinde tutmak bir hata mıdır?

Veri tabanlarında performans artışı için neler yapılabilir?

Hatalı nitelik gruplaması yapıldığında oluşabilecek riskler nelerdir?

Hatalı tasarım, çalışan bir sistem üzerinde nasıl düzeltilebilir?

Başlamadan Önce

İlişkisel veri tabanlarında tablolar arasındaki ilişkiden öte nitelikler arasındaki ilişkiler de oldukça önemlidir. Eldeki veri kümesinde yer alan nitelikleri kullanarak bir ilişkisel veri tabanı hazırlama sürecinde hangi niteliklerin ne şekilde gruplanacağı ve tablolar oluşturulacağı, nitelikler arasındaki ilişkilerin belirlenmesi ile mümkündür. Nitelikler arasındaki ilişkilerin tanımlanması ise fonksiyonel bağımlılıkların belirlenmesiyle mümkündür.

Fonksiyonel bağımlılık, nitelikler arasında hangi nitelik grubunun hangi nitelik grubu tarafından temsil edilebildiğini gösteren bir ifadedir. Fonksiyonel bağımlılığın tanımlanması, niteliklerin birbirleri arasındaki ilişkileri daha kolay keşfetmeyi ve bu sayede daha doğru veri tabanı tabloları oluşturmayı sağlamaktadır. Bu sebeple fonksiyonel bağımlılık tanımlama becerisi her veri tabanı tasarımcısı tarafından mutlaka içselleştirilmesi gereken bir süreçtir.

Bu bölümde fonksiyonel bağımlılığın ne olduğu, nasıl gösterildiği ve nasıl kurulduğu gösterilecek, bununla ilgili beceriler kazandırılmaya çalışılacaktır. Bu bölümün yoğunluklu amacı bu konuda bilgi sunmak değil, beceriyi kazandırmaktır. Bu sebeple bölüm boyunca kapsamlı ve karmaşık alıştırmalarla karşılaşacaksınız.

Dersin bu bölümünden yeterince faydalanabilmek için mutlaka alıştırmalara öncelikle siz yorum getirmeli, sonrasında kitap içerisinde sunulan yorum ve çözüm önerilerini incelemelisiniz. Daha uğraştıracağı kesin olan bu bölüm, sonucunda size bu beceriyi kazandırmış olacak ve dersin ilerleyen bölümlerinde gerçekleştirilmesi gereken faaliyetleri için hazır oluş durumunuzu yükseltecektir.

İlişkisel veri tabanı denildiğinde akla öncelikle tablolar arasındaki ilişkiler gelmektedir. Hem veri tabanı tasarımında kullanılan varlık-ilişki (ER) diyagramları, hem de verinin tablolara bölünmüş yapısı “ilişki” kelimesinin doğrudan tablolar arasındaki bağlantıya atfedilmesine sebep olmaktadır. Ancak ilişkisel veri tabanlarında tablolar arasındaki ilişki, zaten niteliklerin gruplandırılmasından sonra ortaya çıkan bir süreçtir. Tablolar arasındaki ilişkiden çok daha önem arz eden, nitelikler arasındaki ilişkilerin ortaya çıkartılması ve tanımlanmasıdır. Hangi niteliklerin birbiriyle ilişkili olduğu, aynı zamanda hangilerinin birlikte gruplanması gerektiğini göstermekte; bu da veri tabanı tablolarının oluşturulmasına olanak sağlamaktadır.

Fonksiyonel bağımlılık, bir nitelik grubu içerisinde yer alan bir alt nitelik kümesinin başka bir nitelik kümesi için temsil edici nitelikte olduğunun belirlenmesidir. Diğer bir deyişle nitelikler arasındaki temsil özelliğinin ortaya çıkartılmasıdır. Günlük hayatta sıkça karşımıza çıkan ID kelimesi, kayıtların işaret edilmesi için kullanılan bir yapay niteliktir. Biz de gerçekleştirdiğimiz tabloların tümünde ID niteliği kullanarak anahtar seçimimizi kolaylaştıracak ve riskimizi en aza indireceğiz. Ancak fonksiyonel bağımlılıkların belirlenmesi, niteliklerin gruplanması ve veri kümesine daha hakim olmamızı sağlayarak daha iyi bir veri tabanı tasarımı gerçekleştirmeyi mümkün hale getirdiği için önemli bir yer tutmaktadır.

Fonksiyonel bağımlılığın ne olduğuna dair tartışmak, bir noktadan sonra ilerlemeyi zorlaştıracaktır. Bu sebeple dersin bu bölümünde fonksiyonel bağımlılığı örnekler üzerinden öğrenerek devam edeceğiz.

Veri tabanı tasarımcılığında deneyimin önemini vurgulamıştık. Fonksiyonel bağımlılığı öğrenmek ve ihtiyaç halinde uygulayabilmek bu sürecin temelinde yer alıyor. Bir nitelik kümesi gördüğümüzde bu niteliklere aralarındaki ilişkilerin tanımlanması ve gerekli şekilde gruplanması gözüyle bakmaya başladığınızda veri tabanı tasarımcılığında farklı bir aşamaya erişmiş olacaksınız.

Fonksiyonel bağımlılık konusunda bilgi ve deneyim sahibi olmak, sizi dersin ilerleyen başlıklarında veri kümesini daha iyi inceleyen, veri tabanı tasarımını gerçekleştirme konusunda sezgi sahibi, hem öznel hem de nesnel kararları alma konusunda hızlı ve isabetli veri tabanı tasarımcıları haline getirecektir. Lütfen bölüm boyunca yer alan alıştırmalar için, yapılan açıklamaları incelemeden önce kendiniz bir yorum getiriniz. Ardından yapılan yorumu okuyarak karşılaştırmalı değerlendirme yapınız.

Bir önceki başlıkta fonksiyonel bağımlılığın gösterim şekline değinmiştik. Hatırlatmak gerekirse;

$X \rightarrow Y$

gösteriminde veri kümesine ait Y nitelik alt kümesi, X alt kümesine fonksiyonel bağımlıdır denir (Vardi, 1985; Riordan, 2005). Ayrıca X alt kümesi, Y alt kümesi için temsil edici niteliktedir. Y'nin herhangi bir değeri, benzersiz bir X ile temsil edilir. Böylece ilgili X değeri kullanıldığında Y değerine denk geldiği kesin olarak bilinir.

Bu bölümde, fonksiyonel bağımlılığı keşfetme becerisini elde ederek, nitelikler arasındaki ilişkileri ön görebileceksiniz. Bunu yapabilmek için çok kez alıştırma yapmak ve farklı örnekler incelemek gerekiyor. Bu bölümün alt başlıkları farklı alıştırmalar içermektedir. Her bir alıştırma ile yeni ayrıntılar fark edeceksiniz. Her bir alıştırma için öncelikle kendinizin bir çalışma gerçekleştirmesi faydalı olacaktır. Basit bir alıştırma ile başlayalım.

Kitap: adi, yayinEvi, isbn, yil, baski, yazar

Yukarıda tarif edilmiş bir veri kümesine sahip olduğumuzu düşünelim. Bu gösterim bize Kitap adında bir veri kümemiz olduğunu, bu veri kümesi içerisinde adi, yayinEvi, isbn, yil, baski ve yazar adlarında nitelikler bulunmaktadır. Amacımız hangi niteliğin ya da nitelik grubunun diğerleri için temsil edici nitelikte olduğunu belirlemektir. Tüm nitelikleri kısaca ele alalım: adi niteliği kitabın adını göstermektedir. İki sebeple bu niteliği temsil edici olarak kullanamamaktayız. Birincisi tüm kayıtlı kitaplar göz önüne alındığında aynı ada sahip birden fazla kitap olma ihtimali oldukça yüksektir. Kitabın adı temsil edici kabul edildiğinde bir ad bilgisi yüksek ihtimalle birden fazla kaydı karşımıza getireceği için tercih etmemeliyiz. İkinci sebep ise bu niteliğin uzun metinler içeriyor olması. Uzun metin içeren ifadeler temsil edici kabul edilmezler. Bunu uç bir örnekle açıklayalım. Bir blok yazısı blog başlığı ve metinden oluşuyor olsun. Blog başlığı tekrarlanabilir. Ancak içerik tekrarlanmaz. Her bir blog içerik metni tüm bloglar için mutlaka farklı olur. Ama ele alınan son derece uzun bir metin temsil edici kullanılabilir mi? Bu pratik bir çözüm mü? İlgili blog kaydına ulaşmak için tüm blog metninin kullanılması, bu metne sahip blog kaydına ulaşmak istenilmesi gerekiyor. Bu da pratik bir çözüm olmayacaktır. Kitap adı da benzer bir özelliğe sahip. Bir niteliğin içerdiği metin miktarı arttıkça pratik olarak temsil edici olması zorlaşmaktadır. Bu sebeplerle adi niteliğini geçiyoruz. yayinEvi niteliği hakkında karar vermesi oldukça kolay olan bir nitelik. Bir yayın evine ait çok sayıda kitap olacağı aşikardır. Dolayısıyla bir kitap kaydının işaret edilmesi için yayin evi bilgisi kullanılmamalıdır. isbn, kitapların kimlik koduna benzer bir değerdir. Her bir kitap farklı bir isbn bilgisine sahip olacaktır. Dolayısıyla temsil edici özelliği kuvvetlidir ve tek başına bile bir kitabın temsil edilmesi için kullanılabilir. yil değeri yine çok fazla kitabı temsil edecektir. Aynı yılda basılan birçok kitap olacağı için bu niteliği tercih etmek uygun değildir. baski niteliği kitabın kaçıncı baskı olduğu bilgisini içerir. Bu nitelik; 1, 2, 3, … gibi benzer değerler alacaktır ve temsil ediciliği düşüktür. yazar niteliği de yine bir yazarın birden fazla kitabı olması mümkün olacağı için temsil edici değildir. Tüm nitelikler arasında isbn niteliği bariz şekilde temsil edici özelliktedir. Bu sebeple ele alınan veri kümesi için fonksiyonel bağımlılığı şu şekilde tanımlayabiliriz:

$\text{isbn} \rightarrow \{\text{adi, yayinEvi, yil, baski, yazar}\}$

4.1. Alıştırma 1

Bir önceki örnek, yaklaşımımızı göstermek üzere oldukça kolay seçilmişti. Şimdi çok benzer ama üzerine daha fazla düşünmeyi gerektiren bir örnek üzerine çalışacağız.

Kitap: adi, yayinEvi, yil, baski, yazar

Bir önceki örnekten farklı olarak bu veri kümesi isbn gibi bariz bir temsil edici nitelik içermemektedir. Tüm nitelikler için açıklamamızı zaten yapmıştık ve hiçbirinin tatmin edici düzeyde bir temsil edici olmadığını biliyoruz. Bu durumda bir kesişim kümesi oluşturmak ve bundan faydalanmak önemlidir. Diğer kayıtların fonksiyonel bağımlı olduğu en az sayıdaki niteliği tercih ederek bağımlılığı ortaya çıkarabiliriz. Eldeki tüm nitelikler, birden fazla kayıtta karşımıza çıkabilecek niteliklerdir. Seçeceğimiz nitelik grubu yani kayıt kesişimi tek bir kaydı temsil edebilmelidir. Amaç, en az sayıda kayıtla eşleşen nitelikleri gruplayarak en nadir durumu elde etmektir. Bunu yapmanın yolu, bir niteliğe ait benzersiz kaç kaydolduğunu tahmin etmektir. Yine uç bir örnekle inceleyelim. Eğer isbn niteliği hala söz konusu olsaydı, 1000 kayıt içeren bir veri kümesi için 1000 farklı değer almasını beklerdik. Bu da her bir kayıt için farklı bir isbn değeri olduğunun kanıtı oldurdu. Şimdi adi niteliğini inceleyelim. Yine 1000 kayıt içeren bir veri kümesinde kaç tane aynı ada sahip kitap olabilir?

Bu noktada tahmine dayalı bir süreci işlediğini fark etmiş olmalısınız. Önceki bölümlerde veri tabanı tasarımının riskleri ön görmek ve öznel süreçler sebebiyle deneyimin başarıya etkisini ele almıştık. Bu aşamada gerçekte karşılaşılacak, belirsizlik durumlarında karar alma süreçlerinin bir benzerini elde etmekteyiz. Bu sebeple verilen alıştırmalar, çeşitli belirsizlikler içerecek şekilde tasarlanmıştır. Dolayısıyla bir nitelik hakkında görüş üretirken biraz hayali, riskleri öngören, her zaman geçerli olmayabilecek planlamalar yapabiliriz.

Şimdi soruya geri dönelim. Bin kayıt içeren bir veri kümesinde aynı ada sahip kaç kitap olabilir? Ya da şu şekilde de sorabiliriz: Bin kayıt içerisinde aynı adda birden fazla kez kaç kitap yer alabilir? Bu sorunun elbette net bir cevabı yok. Elimizde veri kümesi bulunuyor olsaydı kolaylıkla bu sayıyı belirleyebilirdik ancak bu durumda tahmini bir yaklaşımda bulunmamız gerekmektedir.

Ele aldığımız örnekte tek bir nitelikle fonksiyonel bağımlılık ifadesini kuramadığımız için mümkünse 2 nitelik kullanarak bunu yapmaya çalışacağız. adi niteliği metin içerdiği için kullanmayı tercih etmediğimizi belirtmiştik. Bir yayın evine ait çok fazla kitap olabileceği için yayinEvi niteliğinin temsil edici özelliği düşüktür. yil değişkeni aynı yılda çıkan kitapları kategorize etmek için iyi bir seçim olabilir. baski, baskı sayısını içeren bir nitelik olduğu için kitap kayıtlarının çoğu; 1, 2, 3… gibi kayıtlar içerecektir. Bu sebeple temsil edici özelliği en düşük niteliktir. Kitabın yazarı ise, aynı yazara ait kitap kayıtlarının olması durumunda ayırt edici olacaktır. Dersin eğitmeni olarak benim çözüm önerim yazar ve yil niteliklerinin birlikte kullanılmasıdır. Bir yazarın aynı yıl yayınlanan kitabının birden fazla olması durumunda bu iki nitelik birden fazla kayda denk gelebilir. Ancak bunun dışında yazar ve yil niteliklerinin birlikte kullanılması durumunda temsil edici nitelik olmaları mümkündür. Aynı yazara ait, aynı yıl yayımlanmış birden fazla kitap olması durumu dışında bu tercih herhangi bir risk barındırmamaktadır. Bu en iyi çözüm olmayabilir. Ancak kişisel olarak benim, en düşük riskli olarak öngördüğüm çözüm budur. Yakın başka çözümler de önerilebilir ve elde gerçekten bir veri kümesi olmadığı müddetçe hangi çözümün daha iyi olduğunu ispat etmek oldukça zordur. Bu durumda öngörülere göre karar vererek çözümü ele almak gerekmektedir. Elde ettiğimiz çözüm şudur:

$$\{yazar, yil\} \rightarrow \{adi, yayinEvi, baski\}$$

En iyi çözümün ne olduğunu söylemek zor olsa da yanlış çözümün ne olduğunu söylemek genellikle daha basittir. Kitabın adı temsil edici nitelikte olmamalıdır. Yayın evi çok fazla kitap kaydıyla eşleşebilir. Ayrıca baskı niteliği genellikle aynı değerleri içerdiği için yine temsil edici nitelikte olmamalıdır. Alıştırmalar boyunca doğru çözümü bulmanın yanında neyin kesinlikle yanlış olacağını da tayin edebilmek oldukça mühimdir.

4.2. Alıştırma 2

Müşteri kayıtlarının tutulduğu Musteri adında bir veri kümesi için fonksiyonel bağımlılığı araştıralım.

Musteri: ad, soyad, telefon, ePosta, dogumTarihi, il, ilce, kayitTarihi, sonSiparisTarihi

Ele alınan tablo müşteriye ait kayıtları içeren bir veri kümesidir. Alıştırmada, verilen niteliklerden hangisinin müşteri kayıtları için temsil ediciliğe en uygun olduğunu belirlemek gerekmektedir. Bu örneği zihinde canlandırmak, diğer birçok örneğe göre çok daha kolay olacaktır. Bir müşteri kaydını temsil edecek niteliği elde etmek, aslında doğrudan bir kişiyi temsil eden niteliği belirlemekle benzerdir. Dolayısıyla bir kişiyi temsil etmek için en kullanışlı niteliği tercih etmek gerekmektedir. Eldeki nitelikleri bu bağlamda inceleyelim.

Bir veri kümesinde aynı ad değerine sahip birden fazla kayıt bulunması oldukça mümkündür. Benzer şekilde soyad, dogumTarihi, il ve ilce değerleri de aynı kayıtlarla eşleşecek değerler içerebilirler. telefon ve ePosta nitelikleri müşteri kayıtlarını ayırt etmek için kullanılabilecek iyi seçeneklerdendir. il ve ilce nitelikleri, alacağı değerler belli olan niteliklerdedir. Belki 1000 kayıtlık bir veri kümesinde daha seyrek olabilir ancak çok daha fazla kayıt içeren bir veri kümesi ile karşılaşıldığında il ve ilce niteliklerinin aldığı değerler benzer olacağı için veri tekrarı ile daha sık karşılaşılacaktır.

kayitTarihi ve sonSiparisTarihi nitelikleri tarih bilgisi içeren niteliklerdir. Tarih bilgisi içeren bir niteliğin temsil edici olması pratikte pek mümkün değildir. Kayıt sayısının az olması durumunda, her bir kaydın farklı tarih ile ilişkilendirilmesi bu niteliklerin seçilmesini mümkün hale getirse de kayıt sayısı artınca aynı tarih farklı kayıtlarla eşleşeceği için temsil edici özelliği kaybolacaktır.

Elimizde 2 çok iyi seçenek var: telefon ve ePosta. İki nitelik de bir kişiyi temsil etmek için kullanılabilecek telefonlarda. Her bir müşteri kaydının bir telefon numarası ve e-posta bilgisine sahip olduğunu söyleyebiliriz. Ayrıca bir telefon numarası bilgisi ve e-posta bilgisi ayrı ayrı bir kaydı temsil edebilecek özelliktedir. Bu niteliklerden herhangi birini tercih etmek bu alıştırma için yanlış yanıt olmayacaktır. Yine de mutlaka bir niteliği tercih etmek için zorlarsak şu yorumu yapabiliriz: Telefon numaraların sahipleri zamanla değişebilir. Bir telefon numarası çok uzun süre kullanılmadığında kapatılır ve başka biri için tekrar hizmete alınır. Ancak bir e-posta adresi için bu geçerli değildir. E-posta adresleri daima bir kişiye ait olur. Bu sebeple çok düşük bir olasılık olsa da telefon numarasının birden fazla kişiyi temsil etmesi mümkün olabilir. Bu sebeple benim tercihim ePosta niteliğini kullanmaktan yanadır. Buna göre fonksiyonel bağımlılık ifadesini yazalım.

$$ePosta \rightarrow \{ad, soyad, telefonNumarasi, dogumTarihi, il, ilce, kayitTarihi, sonSiparisTarihi\}$$

4.3. Alıştırma 3

Bir sonraki alıştırmayı ele alalım. Bir bilgisayarda çeşitli fotoğrafların saklanıldığını ve bu fotoğrafların bir kayıt listesinin tutulduğunu düşünelim. İlgili veri kümesini aşağıdaki gibi tarif edebiliriz.

Fotograflar: ad, boyut, dosyaTuru, w, h, olusturulmaTarihi, degistirilmeTarihi

Veri kümesindeki fonksiyonel bağımlılığı belirlemek için nitelikleri teker teker ele alalım. ad niteliği fotoğrafın adını belirten, metin değerler alan bir nitelik. Metin nitelikleri temsil edici olarak seçmekten kaçındığımızı belirtmiştik. Bu sebeple bu alıştırmada da ad niteliğini tercih etmeyeceğimiz niteliklerden biri olarak belirleyerek bir kenara bırakıyoruz.

boyut niteliği tartışmamız gereken, ilginç sayılabilecek bir nitelik. Öncelikle şunu söylemek gerekir ki sayısal nitelikler temsil edici olma konusunda en sık tercih edilenlerdir. Az bir bellek kullanarak benzersiz şekilde kayıt temsil etmek, çok fazla kombinasyona sahip olmak ve kolay yönetilebilir olması sayısal niteliklerin temsil edici olmasını kolaylaştıran etmenlerdendir. Daha önceki bölümlerde sayısal veri ve metin içeren niteliklerin bellekte saklanılmasıyla ilgili ayrıntıları vermiştik. Sayısal bir değeri saklamak ve işlemek, bilgisayar ortamında daha az kaynak gerektiren bir karar olacaktır. Benzersiz sayısal değerler üretmek, rastgele sayı üretmek ya da sıralı sayı verme gibi yöntemler sayesinde kolaylık sağlamaktadır. Bu sayede sayısal değerleri temsil edici kullanmak, bu değerleri daha kolay yönetebildiğimiz için daha uygundur. Sayısal değerlerin sıralı kullanımında, ulaşılan sayı kadar çok kombinasyon kullanılması mümkün olduğu için çeşitlilik anlamında da fayda sağlamaktadır. Hatta 10 sayı tabanından daha yüksek başka bir sayı tabanına ulaşmak daha az karakterle daha fazla kombinasyon elde etmemizi sağlayacaktır. İşin özü; sayısal değerler her zaman bizim için bir adım öndedir.

Şimdi boyut niteliğine geri dönelim. Dosya boyutunu hangi mertebede aldığımız oldukça önemli. Örneğin fotoğraflar için konuştuğumuzda megabayt mertebesinde her fotoğraf 5, 6, 7 gibi benzer değerler alacaktır.

Dosya boyutundaki farklar, megabayt mertebesine ulaşırken yuvarlanacağı için tüm dosyalar aynı boyutta gibi görülecektir. Ancak kilobayt ya da bayt mertebesinde her dosya farklı değerlerde boyutlara sahip olacaktır. Bu sebeple boyut niteliğini kilobayt ya da bayt mertebesinde kabul ederek boyut niteliğini tercih edilebilecek olanlar listemize ekleyerek diğer nitelikleri incelemeye devam edebiliriz.

dosyaTuru niteliği, dosyanın hangi formatta kaydedildiği bilgisini içermektedir. Benzer dosyalar aynı dosya türüne sahip olacağı için temsil edici nitelikte olmayacaktır. Farklı kaynaklar birleştirilse bile görsel dosyaların olabilecekleri biçimlerin sayısı oldukça sınırlıdır (jpg, png, gif, vb). Bu sebeplerden dolayı bu niteliğe fonksiyonel bağımlılık da mümkün gözükmemektedir.

w ve h nitelikleri genişlik (weight) ve yükseklik (height) anlamındadır. Genellikle benzer kalitede fotoğraflar aynı çözünürlüğe sahip olur. Örneğin Full HD olarak bilinen görüntü türü için 1920 piksel genişlik, 1080 piksel yükseklik kullanıldığı görülmektedir. Her ne kadar w ve h nitelikleri de sayısal olsa da genellikle aynı değeri alan nitelikler olacağı için temsil edicilik özellikleri düşük olacaktır.

Son 2 niteliğimiz; olusturulmaTarihi ve degistirilmeTarihi. Tarih içeren niteliklerle ilgili ayrıntılardan daha önce bahsetmiştik. Tarih içeren nitelikler hem aynı tarihin kayıtlarda tekrar yer alması sebebiyle hem de metin içerikli nitelikler olduğu için anahtar olarak kullanılmaması gereken niteliklerdendir.

Tüm bu ayrıntıları incelediğimizde en iyi seçenek boyut niteliği gibi gözükmektedir. Peki boyut niteliği fonksiyonel bağımlılık tanımlaması için tek başına yeterli midir? Eğer dosya boyutu mertebesi, daha hassas sayı sunan bir ölçekteyse her kayıt için yeterince farklı değer elde etmek mümkün olacaktır. Birebir aynı boyuta sahip iki dosyaya sahip olma olasılığımız oldukça düşük olmakla birlikte bunu engelleyecek ikinci bir nitelik öneremiyoruz.

$$\text{boyut} \rightarrow \{\text{ad}, \text{dosyaTuru}, \text{w}, \text{h}, \text{olusturulmaTarihi}, \text{degistirilmeTarihi}\}$$

olarak yanıtımızı kayıt altına alabiliriz.

4.4. Alıştırma 4

Bir teknik servis tarafından tutulan kayıtlarla ilgili bir veri kümesini ele alalım. Bu veri kümesi aşağıdaki gibi tanımlanmış olsun:

TeknikServisKayitlari: urunTuru, marka, model, ariza, teknisyen, müşteri, iletişim

Yine tüm nitelikleri tek tek ele alıp değerlendirmemiz gereken bir problem. Bunu, hem doğru sonucu belirleyebilmek için hem de yanlış ¸yanıt verenlere gerekçe sunabilmek için yapıyorum. urunTuru niteliği birkaç değerden birini alan, dolayısıyla çok sayıda kayıtta kendini sürekli tekrar edebilecek bir nitelik. Aynı marka ya da modele sahip birçok cihazın servise gelmesi de muhtemel olduğu için tercih etmek mantıklı olmayacaktır. ariza çeşitli değerler alabilecek olsa da çok sayıda kayıtta yine kendini tekrar etmesi muhtemel bir nitelik. Serviste çalışan teknisyen sayısı sınırlı olacağı için teknisyen de kendini tekrar eden kayıtlara sahip olacak ve bu sebeple işaret edici olmayacaktır. Bir müşterinin aynı servise birden fazla kez uğraması mümkün olduğundan musteri ve buna bağlı bir nitelik olan iletisim de işaret edicilikten uzaktır. Görüldüğü üzere tüm nitelikler için çeşitli bahanelerimiz var. Çözüm için kafa yorarken sizin de problem yaşadığınızı düşünüyorum. Bu zor durumu kurtarmak için birden fazla nitelikten faydalanmamız gerektiği açık. Dikkat etmemiz gereken nokta riski en aza indirmektir.

Tüm niteliklere tekrar göz gezdirerek en az tekrar eden nitelikleri belirlememiz gerekiyor. En az tekrar eden iki niteliği seçmek, bu ikisinin kesişiminin tekrar etme olasılığını ciddi şekilde düşüreceği için bize güzel bir çözüm getirebilir. Burada tercih, problemi hayal etmenize ve tüm riskleri düşünmenize bağlı olarak değişebilir. Eğer zorlanıyorsanız şu yöntem faydalı olacaktır: Bu veri setinde 10.000 kayıt olduğunu düşünün ve her bir niteliğin bu kadar kayıtta kaç farklı değer alacağını tahmin etmeye çalışın. En az tekrar eden ikiliyi inceleyin.

Tamamen kendi hayal gücüme güvenerek, 10.000 kayıt için niteliklerin benzersiz kaç kayıt içerdikleri konusunda s ¸öyle bir tahmin yaptım: urun- Turu: 10, marka: 3, model: 50, ariza: 500, teknisyen: 20, musteri: 8.000, iletisim: 8.500. Bir müşterinin iletişim bilgisini güncelleyeceğini düşünerek iletisimi biraz daha fazla

tuttum. Bu tahminlere göre musteri ve iletisimi birlikte kullanmak harika bir çözüm gibi görünüyor ancak bu iki değer çoğunlukla birlikte hareket ettiği için mantıklı da olmayacaktır. Dolayısıyla bunlardan biri ve bence daha sağlıklısı olan iletisim niteliği kullanılmalıdır.

Riski en aza indirmek için seçilebilecek ikinci nitelik ise sayılar gözetildiğinde arizadır. Bununla birlikte çözüm önerimiz şöyle şekillenmiştir:

{iletisim, ariza → {urunTuru, marka, model, teknisyen, musteri}

Buradan çıkan sonuç; bir iletişim bilgisi ve arızanın ne olduğu bilgileri birlikte kullanılarak en iyi yaklaşım elde edilebilir, diğer niteliklerin bu iki niteliğe fonksiyonel bağımlı olduğu söylenebilir. Bu çözümün de mükemmel olmadığını söylemeye gerek olmayacaktır ancak riski tahminlerimiz doğrultusunda en aza indirmeyi başardık. Sizin ürettiğiniz çözüm, yine hayal ettiğiniz durum ve tahminleriniz doğrultusunda farklılık gösterebilir. Bu sebeple farklı yanıtlara doğrudan hatalı demek mümkün değildir ancak yine de verdiğiniz yanıt farklıysa ikinci kez göz gezdirmek ve yorumlamak faydalı olacaktır.

4.5. Alıştırma 5

Sınavlardan alınan puanların kaydedildiği bir veri kümesini ele alalım. Veri kümesi tanımlamamız aşağıdaki gibi olsun:

Sinav: ogrenci, ders, not, yil, donem

Üzerine düşünülmesi kolay olmayan bir alıştırma ile karşı karşıyayız. Problem; tek bir nitelikle çözüme ulaşılamayacak, niteliklerin bir kombinasyonunu gerektiren yapıdadır. Hangi nitelikleri seçeceğimizi belirlemeden önce tüm nitelikleri kısaca yorumlayalım.

öğrenci niteliğini doğrudan metin içeren bir nitelik olarak da görebiliriz ancak bunun yanında her bir kaydı bir kişiyi işaret eden bir nitelik olarak da görebiliriz. Yani, bir kayıt dosyasında bir öğrenciye ait kayıt tekrarlanma durumu ne sıklıkla yaşanılır sorusuna yanıt aramak gerekiyor. Her öğrenci için her bir dönem, her bir ders ve dönem için sınav notu kaydı yapılacaktır. Bu sebeple öğrenci kaydının çok fazla kez tekrar ettiğini söyleyebiliriz.

ders niteliği, aynı dersi alan her öğrenci için ve farklı yıllarda tekrar tekrar anılacağı için çok kez tekrar edecektir ve temsil niteliği yoktur. not niteliği, öğrencinin dersin sınavından aldığı notu ifade etmektedir. 0 ile 100 aralığında bir değer alacaktır. 100'den fazla kayıtla ilgileniyor olduğumuz her durumda kendini tekrar edeceği kesindir. yil niteliği ilgili sınav notunun alındığı yılı gösterir. Bir yıl içerisinde çok sayıda kayıt tutulacağı için yine temsil edici bir niteliğe sahip olmayacaktır. donem niteliği de çoğunlukla yalnızca 2 farklı değer alan, sınav notunun alındığı dönemi gösteren niteliktir. Bu nitelik de temsil edici özellikte sayılamaz.

Tüm nitelikleri eledik. Peki en iyi çözüm ne olabilir? Hangi nitelik(ler) hangi nitelik(ler)e fonksiyonel bağımlıdır? Fonksiyonel bağımlılığı tanımlamak için bir nitelik grubu tanımlanması gerekmektedir. En belirleyici iki niteliği seçmeye çalışalım. ogrenci ve ders nitelikleri, kesişim kümeleri sebebiyle en belirleyici 2 nitelik olarak gözüküyor. Belli bir yılın bir dönemindeki not nitelikleri, bir öğrenci ve ders niteliklerine fonksiyonel olarak bağımlı sayılabilir. Bu durumda;

{ogrenci, ders}→{yil, donem, not}

gösterimi uygundur. Burada bir risk ile karşı karşıyayız. Eğer bir öğrenci farklı yıllarda aynı dersi tekrar alır ve bunun için yeni kayıt oluşturulursa mevcut çözüm önerimiz hatalı kalabilir. Daha iyi bir çözüm ancak şu şekilde elde edilebilir:

{ogrenci, ders, yil, donem}→{not}

Veri kümesindeki not niteliği; ogrenci, ders, yil ve donem niteliklerine fonksiyonel bağımlıdır. Diğer bir deyişle not verisinin alınabilmesi için ogrenci, ders, yil ve donem bilgilerine ihtiyaç duymaktayız.

4.6. Alıştırma 6

Bir ülkede yer alan radyo kanallarıyla ilgili bir veri kümesini inceliyor olalım. Veri kümesi şu şekilde tanımlanmış olsun:

RadyoKanallari: ad, frekans, sehir

Burada ad niteliğinin metin bir nitelik olması sebebiyle zorunlu olmadıkça temsil edici nitelik olarak seçmeyeceğimizi biliyoruz. frekans ise radyo kanalının yayın yaptığı frekans değeridir ve sayısal içeriği bulunmaktadır. Bu sebeple mantıklı bir seçim olarak karşımıza çıkar. sehir ise daha önce öğrenci niteliği için yaptığımız yorumu hak eder. sehir niteliği metin içeren bir niteliktir ancak aslında bir kategori gösterir. Yani hiçbir zaman uzunca bir metin içermeyecektir. Dahası, şehrin adı yerine plaka kodu ya da kısaltması gibi daha kısa belirteçler içeriyor da olabilir. Kişiyi, şehri, ülkeyi ve benzeri olguları ifade eden, kategori içeren değişkenler de temsil edici nitelikle karşımıza çıkabilirler. Peki bu örnekte sehir niteliğini kullanmaya ihtiyacımız var mı? frekans niteliğini fonksiyonel bağımlılık ifadesinin sol tarafında, ad niteliğinin ise sağ tarafında olduğunu belirledik. Peki sehir ve ad niteliklerinin frekans niteliğine fonksiyonel bağımlı olması mı daha doğrudur yoksa ad niteliğinin frekans ve sehir niteliklerine fonksiyonel bağımlı olması mı? Radyo kanallarının frekans bilgileri şehirden şehire farklılık göstermektedir. Bu sebeple bir radyo kanalını temsil edecek en iyi değer sadece bu radyo kanalının frekansı değildir. Hangi şehirde olduğunun bilgisinin de alınması gerekmektedir.

{frekans, sehir}→ad

Yukarıda yer alan fonksiyonel bağımlılık ifadesi, belirlediğimiz çözüm için uygundur.

4.7. Alıştırma 7

Araç takibi cihazı üreten bir firmanın, cihazlardan topladığı bilgiyi kaydettikleri veri kümesini ele alalım. Bu veri kümesi aşağıdaki gibi tanımlanmış olsun:

AracTakip: plaka, kullanici, mesafe, ilkKm, sonKm, tarih

Bir aracı temsil edebilecek en net nitelik oldukça kolay göze çarpıyor. plaka niteliği, bir aracı kolayca tanımlayabilecek, benzersiz, fonksiyonel bağımlılık ifadesinin sol tarafında yer alabilecek bir niteliktir. Bunu kesinlikle kullanmamız gerektiğini bir kenara not edebiliriz.

Peki plaka niteliği bu veri kümesi için yeterli midir? Eğer veri kümemiz Arac adında, araçların listelendiği bir veri kümesi olsaydı muhtemelen yeterli olacaktı. Ancak araç takibinden bahsediyoruz. ilkKm ve sonKm değerleri aracın farklı km aralıkları için takip edildiğini göstermektedir. Bu sebeple plaka tek başına yeterli olmayabilir. Bununla birlikte hangi takip sürecinin kayda alındığını da belirtmek gerekecektir. Bunun için de en kullanışlı nitelik tarih niteliğidir. Böylece çözümümüzü aşağıdaki gibi gösterebiliriz.

{plaka, tarih}→{kullanici, mesafe, ilkKm, sonKm}

Bölüm Özeti

Fonksiyonel bağımlılık, bir nitelik grubunda yer alan bir alt nitelik kümesinin bir diğer alt nitelik grubu tarafından temsil edilebilirliğini gösteren yapıdır. Fonksiyonel bağımlılık sayesinde bir nitelik grubunda hangi nitelik ya da niteliklerin temsil edici olduğu belirlenebilir. Temsil edilen nitelikler için temsil eden niteliklere fonksiyonel bağımlıdır denir. Bu ilişkinin tanımlanması niteliklerin gruplandırılması açısından önemlidir. Aralarında fonksiyonel bağımlılık olan nitelikler gruplandırılır ve bir veri tabanı tablosu olarak tanımlanır. Bu sebeple dersin ilerleyen bölümlerinde yer alan veri tabanı tasarlama süreçlerinde bu tanımlamaların yapılması, yanlış bir veri tabanı tasarımı yapılmaması açısından önemlidir.

Fonksiyonel bağımlılık konusu çok fazla bilgi içeren bir konu değildir. Ancak bu ilişkileri tanımlamak deneyim gerektirmektedir. Bu sebeple bu bölümde bilgiden çok alıştırmalara ağırlık verilmiş, bu sayede ilgili

becerinin kazandırılması hedeflenmiştir.

Fonksiyonel bağımlılık, hangi nitelik ya da nitelik grubunun temsil edici özellikte olduğunun belirlenmesi sürecidir. Bu süreç boyunca mümkün olan en az sayıda niteliği temsil edici olarak tercih etmek gerekmektedir. Ancak bazı durumlarda hiçbir nitelik tek başına iyi bir temsil edici olmayabilir. Bu durumda bir nitelik grubunu tercih etmek gerekir.

Tercih edilen niteliklerin ne tür veri içerdikleri de önemlidir. Sayısal veri içeren nitelikler öncelikli tercih sebebidir. Metin içeren nitelikler ise her seferinde benzersiz içeriğe sahip olsalar bile pratikte kullanışlı bir temsil edici olmayacakları için tercih edilmezler. Bir de bu iki durumun ortası var. Bir nitelik metin içerik alıyor ancak bu metin içerikler bir kategoriyi işaret ediyor olabilirler. Şehir bilgisi içeren nitelikler bu duruma bir örnektir. Şehir adları bir metin içeriği olsa da gerçekte bir kategoriyi ya da olguyu işaret ederler. Kişileri belirten nitelikler için de aynı durum geçerlidir. Kişi adıyla ilgili bir nitelik, bir kişinin adını ve soyadını içerdiği için metin içeriklidir. Ancak gerçekte bir kişiyi işaret etmektedir. Şöyle netleştirebiliriz; kişi ile ilgili nitelik içerisinde kişinin ad ve soyad bilgisini silip kimlik numarası bilgisini ekleyebiliriz. Böylece veri bütünlüğü bozulmaz. Biz hala ilgili kişiyi işaret ediyor oluruz. Bu sebeple bu tarz kategori belirten nitelikleri de temsil edici özellikte kullanmamız mümkündür.

Fonksiyonel bağımlılık belirlemenin önemli özelliklerinden biri zaman zaman herkes için geçerli doğru bir yanıt bulunmamasıdır. Elimizde yalnızca bir nitelik grubu varsa ve bu niteliklerin alacağı değerler konusunda varsayımlarda bulunmak zorundaysak, her veri tabanı tasarımcısı farklı varsayımlarla farklı kararlar alabilir. Bu sebeple bölüm içerisinde veriler çözümlere benzer ancak farklı çözümler üretmiş olabilirsiniz. Bu bir hata değildir. Ancak şuna dikkat etmek gerekiyor; her zaman kesin doğru olmayabilir ancak her zaman kesin yanlışlar vardır. Uzun metin içeriklerini temsil edici olarak kullanmak ya da bir nitelik yeterliyken ikinci bir niteliği de temsil edici tarafta kullanmak hatalı görülebilecek kararlardandır.

Fonksiyonel bağımlılıklar konusunda iyi çözüm önerileri sunmak, dersin ilerleyen kısımlarında da başarılı tasarımlar gerçekleştirmenizde fayda sağlayacaktır.

Kaynakça

Akadal, E. 2020. Veritabanı Tasarlama Atölyesi. Türkmen Kitabevi, İstanbul.

Vardi, M. Y. 1985. Fundamentals of dependency theory. IBM Thomas J. Watson Research Division.

Riordan, R. M. 2005. Designing effective database systems. Addison- Wesley Professional.