Veritabanı Tasarımı

Dr. Hidayet TAKÇI

Tekrarlama Problemleri

- İlişkisel şemalardaki problemlerin temelinde verilerde oluşabilecek tekrarlanma vardır.
- Tutarlılık kısıtları, özellikle fonksiyonel bağımlılıklar şemalarda oluşabilecek bu problemleri tespit etmede kullanılır.

Ayrıştırma (Normalleştirme)

- Veritabanı problemlerinin üstesinden gelebilmenin en iyi yöntemi ayrıştırmadır.
- Yani ABCD niteliklerinden oluşan bir şema AB ve ACD niteliklerinden oluşan iki farklı şemaya ayrıştırılabilir.
- Ayrıştırmanın tipi önemli bir konudur.

İlişkisel Şema Tasarımı Ölçütleri

- Niteliklerin anlamları olmalı: her bir niteliğin bir diğerleriyle olan ilgileri ve nasıl yorumlanmaları gerektiği belirgin olmalıdır.
- Veri giriş ve güncelleme problemleri engellenmeli: aynı verinin iki kez girilmesi, olmayan verinin silinmeye çalışılması veya güncellenmeye çalışılması gibi hatalar engellenmelidir.
- Boş değerler minimize edilmeli: veri girişlerinde mümkün olduğu kadar boş geçmemeye dikkat edilmeli.
- İlave satırlar engellenmeli: tabloların birleştirilmesi sonucu bazen gereksiz satırlar meydana gelir. İlave satırların engellenmesi için şart cümlecikleri doğru yazılmalıdır. (where cümleciğinden sonrası)

Fonksiyonel Bağımlılık

- X ile Y nitelik kümelerini göstermek üzere fonksiyonel bağımlılık X → Y şeklinde verilir ve X ten Y ye kısıtlama şartı belirtir. y=f(x), x bağımsız değişken, y ise bağımlı değişkendir.
- Fonksiyonel bağımlılık kuralları
  - {X→Y} |= XZ→ YZ // çoğaltma kuralı
  - {X→Y, Y→Z} |=X→Z // geçişlilik kuralı
  - {X→YZ} |=X→ Y // X→Y // ayrıştırma kuralı
  - {X→Y, X→Z} |=X→YZ // toplama kuralı

Aday Anahtar

- X' in R ilişkisi için aday anahtar olması X → R şeklinde verilir. Yani, eğer R ilişkisi {XSNMLK} ise ve X bir anahtar ise X → SNMLK şeklinde verilir.
- Aday anahtarlardan bir tanesi birincil anahtar olarak seçilir. Bu anahtar değeri hiç bir zaman NULL değeri alamaz.

SQL ‘de Birincil Anahtar Tanımı

- Birincil anahtar SQL ile şu şekilde verilir.

```
Create table ogrenci
    (
        o_no integer not null,
        o_isim char(15) not null,
        yas integer,
        primary key (o_no),
        unique (o_isim)
    )
```

o_isim niteliğinin aday anahtar olacağına karar verdi isek UNIQUE sözcüğü ile onu VTYS’ ye veririz.

Yabancı Anahtar

- Yabancı anahtarı (foreign key) VTYS' ye nasıl tanıtacağımızı aşağıdaki örnekte görmek mümkündür.

```
Create table ogrenci
    (
        o_no integer not null,
        o_isim char(15) not null,
        d_no integer,
        yas integer,
        primary key (o_no),
        unique (o_isim),
        foreign key (d_no) references dersler
    )
```

Tekrarlama ve güncelleme problemi

- Satırlardaki tekrarlanmış bilgi ve güncelleme problemlerine örnek olarak kartezyen çarpım ile birleştirme operasyonları sonucu elde edilen satırlar verilebilir.
- Fazlalık satırlar kartezyen çarpımlar sonucu oluşurken, fazlalık sütunlar hatalı birleştirme operasyonları sonucu oluşur. Bu problemi gidermek için doğal birleştirme gibi operasyonlar kullanılır.

Diğer Problemler

- Giriş Problemi:
  - Eğer şirket-bölüm gibi bir ilişkiye yeni bir kayıt ekleyeceksek ya tüm niteliklerin değerini girmeli ya da bazılarını boş bırakmalıyız.
  - Eğer şirket ve bölüm bilgilerini ayrı ayrı girecek olursak o zaman problem olmadığı halde şirket-bölüm ilişkisinde bir çalışanın bölümünün başındaki elemanı yanlış biliyorsak o zaman giriş problemi yapmış oluruz.
  - Bu problem kütüphanede bir zamanlar sıkça meydana geliyordu, üye tablosuna bölüm bilgisi girilirken hatalı girişler yapılıyor sonra da bölümlere göre üyeleri bulmak imkansız hale geliyordu. Bu problemin çözümü için yabancı anahtar kısıtlamasından faydalandık.

Silme Problemi

- Eğer şirket-bölüm ilişkisindeki kayıtları silerken ilişkideki son kaydı silersek o zaman bölüm bilgilerini kaybetmiş oluruz. Fakat şirket ve bölüm gibi iki farklı ilişkimiz olsaydı o zaman böyle bir problemimiz olmayacaktı.
- Bir başka silme problemi de bir zamanlar kütüphanede yaşanmaktaydı. Öğrenci olan üyelerimiz asistan oldukları zaman üye numaraları değişiyor ve eğer üzerindeki ödünç verilmiş kitap bilgileri transfer edilmemişse bu bilgiler kayıp ediliyordu.

Güncelleme Problemi

- Şirket-bölüm ilişkisinde bir bölüm hakkında değişiklik yapılacağı zaman bu değişiklik bütün bölüm için yapılmalıdır.
- Yine kütüphane veritabanı üzerinde konuyu açıklamak gerekirse;
  - Üye tablosunda her bir üyenin bağlı olduğu bölüm bir numara ile tutulmaktadır. Eğer bölüm tablosundaki bölüm numarası değişecek olursa bu değişiklik bütün üye tablosunda yapılmalıdır. Aksi taktirde veri tutarsızlığı meydana gelecektir.

Ayrıştırma

- Eğer bir ilişki belirli bir normal formda ise (2NF, 3NF) zaten bazı problemlerin oluşması engellenmiştir. Bu yüzden onların daha fazla ayrıştırılmalarına gerek yoktur.
- R ilişkisi A1, ..., An niteliklerini içermek üzere; R ilişkisinin ayrıştırılması R ilişkisinin birden fazla ilişki ile yer değiştirmesidir. Ayrıştırılan ilişkiler şu özellikleri gösterir:
  - her bir yeni ilişki şeması R ilişkisinin niteliklerinin bir alt kümesini içerir. R de olmayan hiçbir nitelik bu yeni şemalarda yer alamaz.
  - R deki her bir nitelik ayrıştırılan ilişkilerden birine ait bir niteliktir.

Normal Formlar

- Normal formlar normalleştirmenin derecelerini veren formlar olup 1NF, 2NF, 3NF,BCNF, 4NF ve 5NF şeklindedir.
- En dışta 5NF ve en içerde 1NF olmak üzere her üst form aynı zamanda alt formun özelliklerini de taşır.

Birinci Normal Form (1NF)

- Birinci normal formda ilişkiyi oluşturan bütün nitelikler ayrı etki alanlarından değer alırlar. Ancak bu halde yeni bir değer eklenmek veya çıkarılmak istendiği zaman satırın tamamı üzerinde çalışmak gerekir. Bu sakıncayı ortadan kaldırmak için ilişkimizi, nitelikleri birbiri ile fonksiyonel bağımlı alt ilişkilere bölmek gerekmektedir.

Örnek

- Örneğin kitap tablosunda, birden fazla yazarı olan kitap için yazar1, yazar2, yazar3 diye alanlar açsaydık, bu kurala uymamış olurduk. Böyle bir durumda, ayrıca yazarlar tablosu da oluşturarak kuralı çiğnememiş oluruz.

Genellikle yapılan hata:
Verileri virgül veya bir başka karakter ile ayırıp aynı alana girmek. Daha sonra program içerisinde split ile bu değerleri ayırmak. Ancak bu ilişkisel veritabanının doğasına terstir.

Birinci Normal Form

- Örneğin, öğrencilerin katıldığı aktiviteler (kayak, yüzme, futbol) ve bu aktivitelerin ücretlerinin tutulduğu bir ilişki birinci normal formda bir ilişkidir.

Aktivite şeması=(ogr_no, aktivite, ucret)
Anahtar(key): ogr_no
Veri örneği(data instance)

Ogr_no	Aktivite	Ucret
65	Kayak	6.000.000
76	Yüzme	15.000.000
83	Futbol	15.000.000
92	Yüzme	15.000.000

Birinci Normal Form

- 65 numaralı öğrenci tablodan silinecek olursa kayak aktivitesine ait ücret bilgisi kaybolacaktır. Ayrıca, ücreti 5.000.000 olan folklor aktivitesi de o aktiviteye bir öğrenci yazılmadığı için tabloda bulunmamaktadır.
- Bu sakıncayı gidermek için öğrenci_aktivite ve aktivite_ucret isimli iki yeni alt ilişki kurulabilir.
  - öğrenci_aktivite şeması =(**ogr_no**, aktivite)
  - aktivite_ucret şeması =(**aktivite**, ucret)
- Şeklinde iki ilişkiye ayırma işlemi ile ikinci normal forma geçmek mümkün olacaktır.

Birinci Normal Form

- Ayrıştırma sonucu ilk baştaki ilişki yandaki gibi iki alt ilişkiye bölünür.

Ogr_no	Aktivite
65	Kayak
76	Yüzme
83	Futbol
92	yüzme

Aktivite	Ücret
Kayak	6.000.000
Yüzme	15.000.000
Futbol	15.000.000
Folklor	15.000.000

İkinci Normal Form

- İkinci normal form (2NF) kendisi anahtar olmayan bütün niteliklerin hepsinin anahtara bağlanması halidir. Böylece bir özelliği anahtar olan bütün ilişkiler ikinci normal formda bulunurlar.
- Eğer R' deki herhangi bir birincil olmayan nitelik (A), R' nin hiçbir anahtar niteliğine kısmi fonksiyonel bağımlı değilse bu ilişki şeması 2NF'tedir.

Örnek

- Bir tablo için, anahtar olmayan her alan, birincil anahtar olarak tanımlı tüm alanlara bağlı olmak zorundadır.
  - Mesela, Ödünç tablosuna Kitap_Adı diye bir alan eklese idik, bu sadece ödünç verilen kitap ile ilgili bir bilgi olacaktı ve Ödünç_No ya bağlı bir nitelik olmayacaktı. Bunu çözmek için, kitap adları ayrı bir tabloda tutulmuştur.
- Yada anahtar alanın birden fazla alandan oluştuğu tablolarda, anahtar alanlardan sadece birine bağlı veriler, tabloda yer almamalı, ayrı bir tabloya taşınmalıdır.
- Bunun tersi de geçerlidir. Yani iki yada daha fazla tablonun birincil anahtarı aynı olamaz. Şayet böyle ise, bu iki tablo tek tabloya indirgenmelidir.

İkinci Normal Form

- İkinci normal formdaki bir ilişkide de anormallikler bulunabilir.
  - Örneğin, bir öğrenci yurdundaki A,B,C bloklarının oda ücretleri birbirinden farklı olabilir. Yerleşim ilişkisi Ogr_no, blok ve ücret niteliklerinden oluşmak üzere aşağıdaki verilerle bu ilişki 2NF‘ tedir.

Yerleşim şeması=(ogr_no, blok, ucret)

Anahtar(key): ogr_no

Veri örneği (data instance)

Fonksiyonel bağımlılıklar: blok → ucret

Ogr_no	Blok	Ucret
65	A blok	250.000.000
76	B blok	100.000.000
83	A blok	250.000.000
92	C blok	100.000.000
98	A blok	250.000.000

İkinci Normal Form

- Bu örnekte, bir öğrenci belirli bir blokta kalmakta ve her bloktaki odaların ücreti aynı olmaktadır, başka bir deyimle öğr_no →blok ve blok→ücret şeklinde fonksiyonel bağımlılık ve dolaylı olarak ta öğr_no→blok→ücret halinde geçişli bağımlılık kurulmuştur. Ancak, 76 numaralı öğrencinin ayrılması halinde, b bloğun ücreti de silinmiş olacaktır. D bloğa yeni bir öğrenci yazılıncaya kadar da, bu bloğun ücreti görülemeyecektir. Bu anormalliği önlemek üzere, ilişki; yandaki ilişkilere bölünür.

Öğr-yerleşim (ogr_no, blok)
Anahtar: ogr-no

Öğr-no	Blok
65	A blok
76	B blok
83	A blok
92	C blok
98	A blok

oda-ucret (blok, ucret)
Anahtar: ogr-no

Blok	Ücret
A blok	250.000.000
B blok	100.000.000
C blok	100.000.000
D blok	150.000.000

Üçüncü Normal Forma Geçiş

- Bağımlılığın fonksiyonel değil de geçişli olması halinde, geçişli bağımlılığın elenerek fonksiyonel bağıntı kurulması ile üçüncü normal form sağlanmaktadır.
- Öğr-no ile ücret arasındaki bağımlılık öğr-no → aktivite → ücret şeklinde geçişli bir bağımlılıktır. Bu ilişkiyi iki ayrı ilişkiye ayırmakla ikinci normal forma dönüştürüldüğü gibi ayrıca öğr-no→aktivite ve aktivite→ ücret şeklinde fonksiyonel bağımlılık kurmuş olmakla üçüncü normal formda sağlanmaktadır.

Üçüncü normal form

- Bir tablo için, anahtarı olmayan bir alan, anahtarı olmayan başka hiç bir alana bağlı olamaz. Örneğin, kitaplarımız için cilt tipi adında bir alan ekleyip burada da karton kapak için K, deri cilt için D, spiral cilt için S yazsaydık, bu kodlama, kitap tablosunun birincil anahtarı olan Kitap_No alanına bağlı bir kodlama olmayacaktı. Çünkü bu kodlama bir başka anahtarı olmayan alana bağlıdır. Bunun sonucunda da veritabanımızda, karşılığı olmayan bir kodlama yer almış olacaktır. Cilt tipi bilgisini kodlu olarak tutan alan aslında cilt tipi açıklaması olan başka bir alana bağlıdır. Bu ilişki başka bir tabloda tutulmalıdır.
- Bu durumda, Cilt_Sekli adında bir tablo açmamız gerekir. Bu tablonun alanları da Cilt_Tip_Kodu ve Cilt_Sekli olmalıdır. Ancak bundan sonra, kitaplar tablosunda Cilt_Tipi adında bir sütun açıp buraya da D,S,K gibi kodları yazabiliriz.

Sonuç

- Veritabanı Normalleştirme kuralları, bir ilişkisel veritabanının tasarlanma aşamalarını değil de ilişkisel veritabanında yer alacak kayıtların ilişkisel veritabanı ile uyumlu olup olmadığını denetlemeye yöneliktir.