Database Normalization

Database normalizasyonu bir tablodaki(data) tekrarları azaltmak için o tabloyu birden fazla tabloya ayırmayı ifade eder.

Bir film kiralama şirketi açtık. 1.günümüzde yaptığımız satışları aşağıdaki gibi kaydettik:

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

`MOVIES RENTED` sütununda birden fazla değer var ve bunlar virgül ile ayrılmış Yani biz iki kişiye iki film satmışız fakat aynı satırda göstermişiz. Bu bir problem. Bu kişiler yeni filmler aldıkça bu sütun uzayacak. Diğer bir problem ise Robert Phil isminde 2 müşterimiz var. Bu kişiler farklı kişiler mi, yoksa adres değişikliği mi oldu?

Database normalization bu tarz problemleri ortadan kaldırmak için tabloları birden fazla tabloya ayırma işlemine deniyor.

Böyle bir tablo olduğunda şu tarz problemler ortaya çıkıyor genelde ve database normalization bu sorunlara çözüm oluyor:

1. Duplicate rows Bazı satırlar tekrar edebiliyor. Bu örnekte tekrar eden satırlar yok. Ancak olabilirdi.
2. More than one value in a cell Bu tabloda bu problem var. Bunun çözülmesi gerekir.
3. Each Row is not uniquely identified Bu tabloda Robert Phil satırları kafa karışıklığı yaratıyor. Aynı kişi mi bu?
4. Tek tablo Birden fazla tablo yapmamız lazım

1.Adım

1. Tekrarlayan satırları(duplicate rows) sil. Tekrar eden satırlar bir bilgi taşımıyor.
2. Birden çok değer taşıyan hücreleri (multivariate cells) ayır. Bunları satır olarak ekle.
3. Her satırı tekil olarak identify eden bir sütun bul. Yoksa ekle.

İlk iki adımı uygulayınca tablomuz şu hale geliyor:

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

Verimizi bu hale getirdikten sonra 3. adıma geçelim. Bir PK bulmamız gerekli. Her bir satırda tekrar etmeyen, etme ihtimali bulunmayan bir sütun var mı? Sırasıyla bakalım:

- Full names:

  Birden fazla satırda aynı full namesi görebilir miyim? Evet. Çünkü kişiler aynı isim ve soyisimi paylaşabilir. **Dolayısıla Full names her bir satır için tek bir değere sahip olmayabilir, bazen tekrar edebilir** Bizim tablomuzda bu durum var. O zaman Full names primary key olamaz. Bunu anlamanın bir diğer yolu da şunu sormaktır: bana sadece full names verilse diğer tüm sütunlar için değerleri bilebilir miyim? Yani mesela Janet Jones verilse bana, bu kişinin pysical adress, movies rented, ve salutationımı bilebilir miyim. Evet. Ama robert için bu durum geçerli değil. Robert phil bilindiğinde adres hangisi? belirli değil. PK olamaz.

- Pyhsical adress:

  Bir satırda gördüğüm pysical adresi başka satırda da görebilir miyim? Yani kimi müşteriler aynı adresi paylaşabilir mi? Evet: yani pysical adres satırı tekrar edebilir. Aynı evde yaşayan iki kişi bizden film kiralamaya gelebilir. Pysical adress primary key olamaz. Yine şöyle düşünebiliriz: phsical adress bilindiğinde diğer sütunlar tanımlanabiliyor mu? Yani 3rd streeti bildiğimde kişinin ne kiraladığını isminin ne olduğunu bilebiliyor muyum? Evet. Ancak şöyle bir problem olabilir: aynı evde yaşayan iki müşterimiz olduğunun düşünelim. İsimleri farklı. Adresi bildiğimizde isimi bilemeyiz. PK olamaz.

- Movies rented:

  Bir film başka müşteriler tarafından da kiralanabilir mi? Evet. PK olamaz. Çünkü tekrar edebilir satırlarda.

- Saluta: Her satırda tekrar edebilir: PK olamaz.

Mevcut halinde verimizde bir PK yok. Ancak şunu farkedebiliriz: full names ve pysical adres tek başlarına birer Pk olmasalar da aslında birlikte her bir satırı tekil olarak identify edebiliyorlar. Yani kişinin Full namesi ve pysical adresi birlikte bilindiğinde kiraladıkları film ve salutaion bilinebiliyor. Bu tarz 2 sütunun bir PK olduğu duruma composite key deniliyor. Bu composite keyleri kullarak bir PK yaratabiliriz. ,

Yani Full names ve pysical adress bileşimlerini tanımlayan bir sütun yaratsam bir PK olur elimde. Peki yaratacağım sütunun adı ne olsun? Neyi tanımlıyorum: müşteri adını ve adresini. CustomerID mantıklı geliyor. Bunu yaratalım, tabloya ekleyelim.

CustomerID	FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
1	Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
1	Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
2	Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
2	Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
3	Robert Phil	5th Avenue	Clash of the Titans	Mr.

Şimdi bakalım `CustomerID` bilinirse tüm satırlar bilinebilir mi?

- Customer id 1 ise kişinin adını biliyorum, adresini biliyorum, kiraladığı filmi ve hitabı biliyorum.
- Customer id 2, ve 3 için de durum aynı.

Datamız bu hale geldikten sonra 2. adıma geçiyoruz.

2.adım

Bu adımda mevcut tablomuzu birden çok tabloya ayırma işlemi yapacağız. Ayırma işlemi functional dependecye göre yapılır. Functional dependent olanlar birlikte tutulur, olmayanların her biri için bir

tablo yapılır.

Functional dependency bir sütunun diğer sütun üzerinde belirleyici olmasını ifade eder. Matematiksel olarak düşünürsek $x = y + 2$ de $x = f(y)$ dir: x y ye functionally dependent'tır. Yani y nin her bir değerine karşılık gelen tek bir x değeri vardır.

Veri setimiz üzerinden giderek functional dependencyleri bulalalım.Bunun için tek tek sütunlara gidiyoruz ve soruyoruz: bu sütun diğer sütunun belirleyicisi midir? Ya da diğer sütun bu sütuna bağlı mıdır?

- `CustomerID`

  CustomerID Full namesi belirler mi? Evet. Pysical Adresi belirler mi? Evet. Customer ID 1 ise full names ve pysical adres belirlidir. Başka bir değer alamaz. Peki CusotmerID movies rentedı belirleyebilir mi? Customer IDnin 1 olması movies rented için belirli bir değer oluştur mu? hayır. Customer id 1 iken de 2 iken de aynı movies rented değeri olabilir. Customer ID salutaionı belirler mi? **Dolaylı yoldan evet**. Her bir customer id ye denk gelen bir salutaion var mı? Evet. Yani customer id 1 ise salutaion Ms dir; 2 ise Mr dir. Burada bir functional dependency var. Ancak bir detay var: salutationın temel belirleyeni customer id değil full namestir. Full names değişirse salutation değişir. Dolayısıyla customer id salutaionı full names üzerinden belirler. Buna transitive functional dependency denir. Bu adımda önemli değil ancak bilmemiz lazım. , Nihai olarak `CustomerID` ile full names, pysical adress ve salutaion arasında functional dependency bulduk. Bu şu demek: bunlar birlikte aynı tabloda yer alacak.

`MOVIES RENTED` bu sütunun hiçbir sütun ile dependecysi yok. Dolayısıyla tek başına bir tablo olacak.

Şimdi ayırma işlemine geçelim.

CusomerID, Fullnames, pysical adress ve salutationı bir tabloda, movies rentedı ayrı bir tabloda gösterelim.

Customers table diyelim buna:

CustomerID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Diğeri de movies table olsun:

MOVIES RENTED
Pirates of the Caribbean
Clash of the Titans
Forgetting Sarah Marshal
Daddy's Little Girls

Şu durumda tablolarımız var ancak aralarında bir relation yok. Yani customerlara ilişkin tabloda movies ile ilgili bir veri yok. Movieslerde ise hangi customerın kiraladığına ilişkin bir veri yok.

Bu durumu iki tablo arasında bir relation yaratarak çözebiliriz. Customers tablosuna movies rentedları girersek ilk tablomuzun aynısı olacak, mantıklı değil. Bu nedenle movies tablosuna customer bilgilerini girebiliriz. Peki hangi bilgileri girmemiz gerek? Full names i pysical adresi ve salutationı girersek yine ilk tablo ortaya çıkacak. Bunları girmeyeceğiz, bunun yerine zaten bunları temsil eden CustomerID verilerini gireceğiz. CustomerID yi yaratmanın anlamı da buydu zaten, tüm bu bilgileri tek bir sayı ile özetlemek.

Dolayısıla Movies tablosuna customerid leri gireceğiz. Dikkat etmemiz gereken şey ise bu girişleri yaparken ilk tabloya mutlaka bakmamız. Movies rentedın ilk değeri için hangi customer idler var: yani hangi customerlar prites of the cariibean kiralamış. Tüm değerler için movies tabledaki bunları yazacağız:

İlk tabloya bakalım

CustomerID	FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
1	Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
1	Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
2	Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
2	Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
3	Robert Phil	5th Avenue	Clash of the Titans	Mr.

CustomerID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
3	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls

Dolayısyla artık tablolarımız bağlantılı oldu.

Tekrar bakalım tablolarımıza:

Customers table:

CustomerID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Movies table:

CustomerID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
3	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls

CustomerID in Customers table is PK: it uniquely identifies each row.

CusomerID in Movies table is a Foreign Key. It is a reference for Customers table. The PK in Movies table is Movies Rented.

3.adım

Bu adım transitional functional dependencylerin kaldırılmasını içeririr. Bu adımda tüm dependencyler primary key ile olmalı.

Mevcut halde Primary keyimiz customer id ile salutation arasında bir functional dependency var. Ancak salutaion ile full name areasında da bir dependency var: full name salutationı belirliyor.

3.adım'da bu dependencynin kaldırılması gerekir. Bunu yapabilmek için de dependent olan ayrı bir tabloya alınır.

Yani salutatin için bir tablo yapıyoruz bu adımda:

SalutationID	Salutation
1	Ms.
2	Mr.

1. Bu tabloya neden customerID referansı vermedik? Neden salutaionID koyduk?

Şöyle: Salutation ancak iki değer alabiliyor. Eğer bu tabloya customerID leri koysaydık her bir customerID için bir satırımız olacaktı. Oldukça uzun bir salutation tablomuz olacaktı. Bunun temel nedenlerinden biri ise salutationın yalnızca 2 değer alabilmesi. Eğer 100 farklı değer alabiliyor olsaydı bu çok problem olmazdı.

Bir alternatifini düşünelim. Customers tablosunda salutationa referans verelim? Bunu yababilmek için Salutation ID ye ihtiyacımız var. Onu yarattıktan sonra customers table da gösterebiliriz:

CustomerID	FULL NAMES	PHYSICAL ADDRESS	SalutationID
1	Janet Jones	First Street Plot No 4	1
2	Robert Phil	3rd Street 34	2
3	Robert Phil	5th Avenue	2

Neden böyle göstermek daha iyi? Düşünün: 100 tane customer var. Her biri için bir ID var. Customer tablosu 100 satırdan oluşuyor dolayısıyla.

Şimdi eğer ilk yolu yapsaydık salutation tablosunda 100 satırımız olacaktı, customersda 100 satırımız olacaktı.

İkinci yolda ise salutationsı customers tablosunda gösterdik. Böylelikle zaten 100 satır olan customers tablosuna ek satır eklemedik. Salutation tablomuz da 2 satır kaldı.

Her şey bitti. Normalize oldu tablomuz.

Gerçekleştirdiğimiz bu 3 adıma sırasıyla 1NF, 2NF ve 3NF deniyor:

- 1NF: First Normal Form

  Duplicateların silinmesi, her hücrede tek bilginin yer alması, PK tanımlanması

- 2NF: Second Normal Form

  Tabloların ayrılması: functional dependecy

- 3NF: Third Normal Form

  Transitional functional dependencyden kurtulunması.

Example

Imagine you have the following data

Name	Address	Gender	T-Shirt Order
Joe Bloggs	37 Buttercup Avenue	Male	Large
Jane Smith	64 Francisco Way	Female	Small
Jane Smith	64 Francisco Way	Female	Medium
Chris Columbus	5 Mayflower Street	Male	Medium
Alex Johnson	123 Main Street	Male	Small
Maria Garcia	78 West Avenue	Female	Large
Sam Lee	90 East Drive	Male	Extra Large
Olivia Brown	12 South Road	Female	Small
Ethan Wright	45 North Crescent	Male	Large
Ava Taylor	67 Queen Boulevard	Female	Medium

Steps:

1. Does this table has duplicate rows or multiple values in cells? If so handle them. Then ask yourself: Is there a primary key right now? If so identify it. If not generate one.
2. What are the functional dependencies? Seperate this table to multiple tables
3. Is there a transitional dependency? If so remove it. If not, well.. there is!