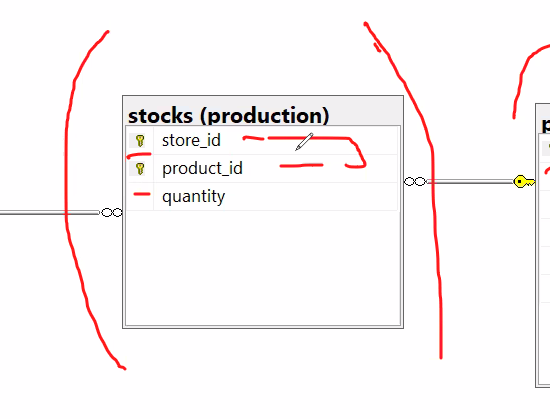
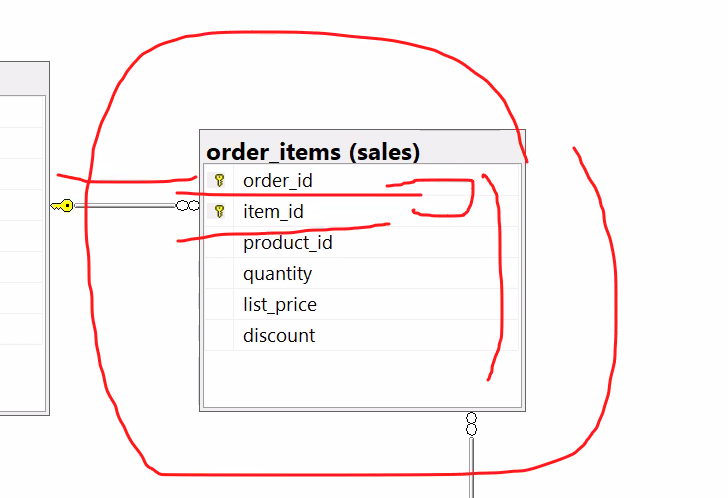
---------------------------------------------------------------------------------------

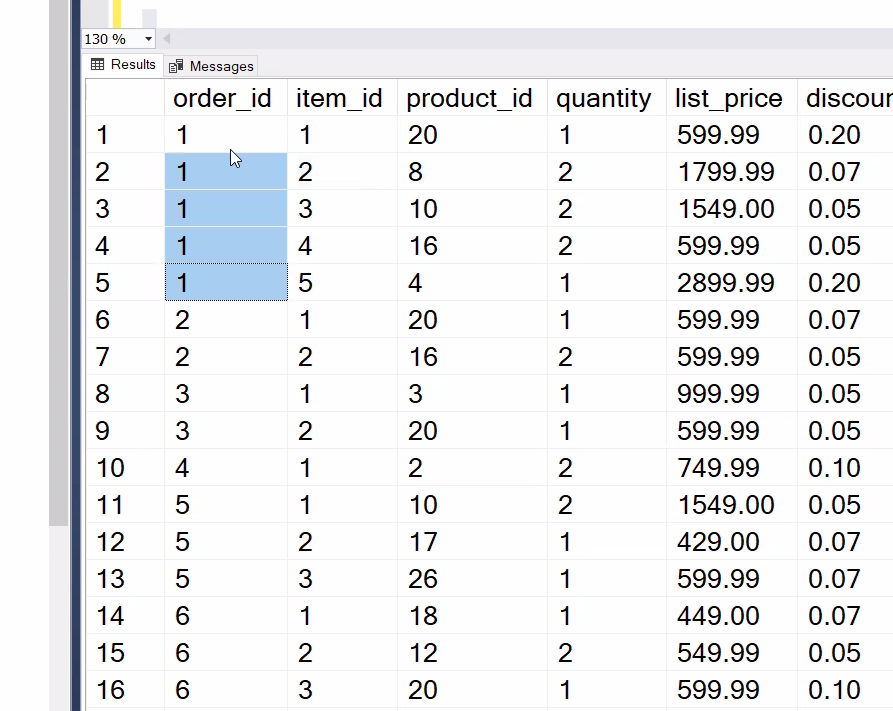


Burada product\_id tek başına primary key olamıyor ancak store\_id ile birlikte olabiliyor. O yüzden başka bir tablodan gelen store\_id ile yine başka bir tablodan gelen product\_id i beraber composite key yaptık. iki sütun birden primary key oldu.



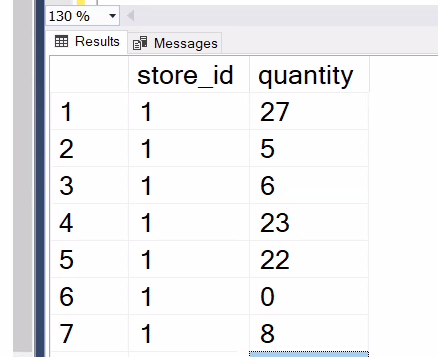
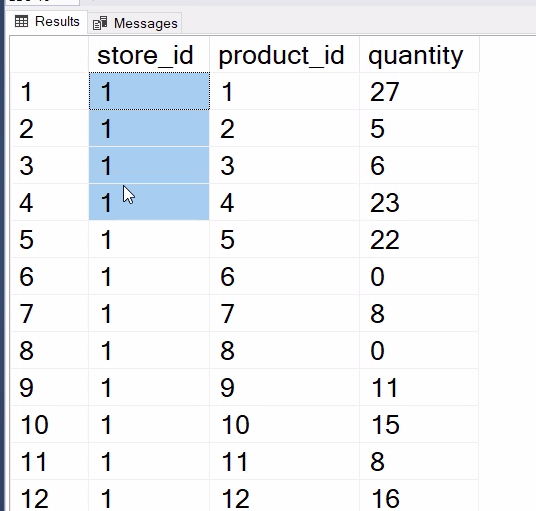
Tek başına order\_id, order\_items tablosunun tüm alanlarını identify edemiyor. Bu yüzden item\_id ile birlikte primary key yapıyoruz.

---------------------------------------------------------------------------------------------------



**Primary key bir tabloyu identify etme yeteneğine sahip keydir.** Ama örneğin burada bakın (order\_id’lerden) 1 çoklanmış. 2. satırda farklı bir item var o yüzden ona farklı bir order\_id verilemiyor çünkü o item\_no da aynı siparişe (order\_id’ye) ait. 3,4 ve 5. Satırda da aynı durum söz konusu. Bu yüzden order\_id ve item\_id birlikte kullanıyoruz ve bunlar beraber bir primary key oluşturuyor.

----------------------------------------------------------------------------------------------

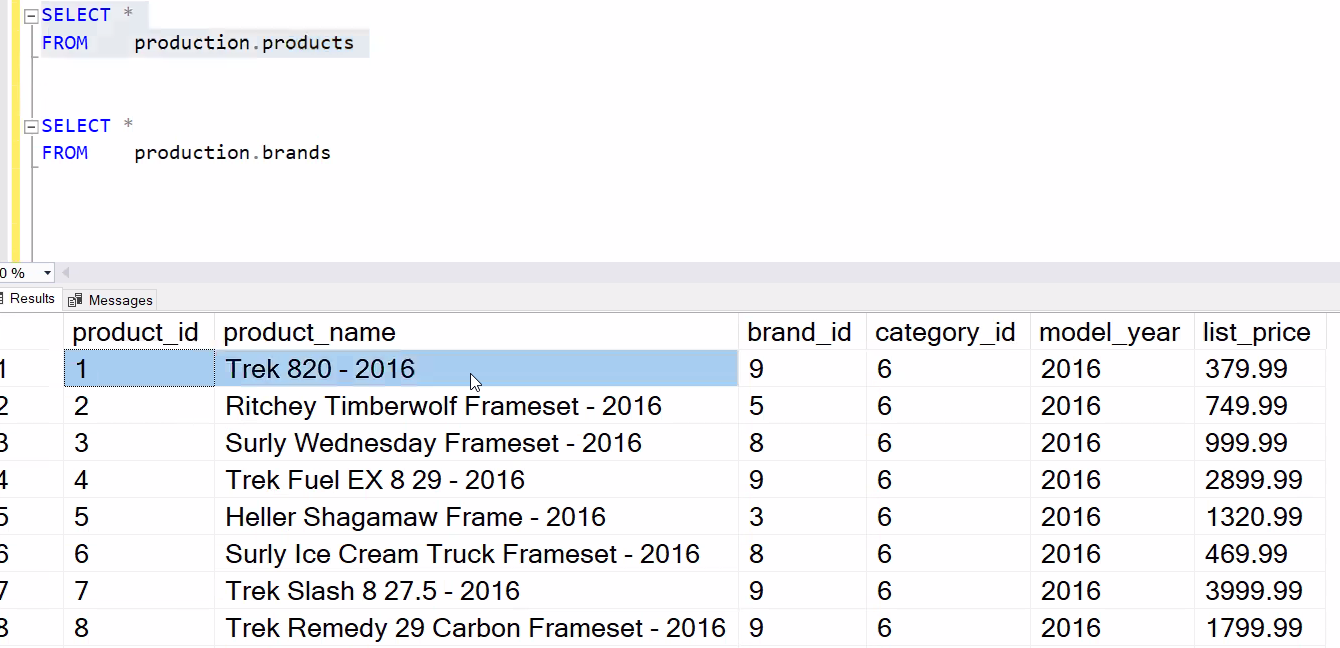


Bu tabloda hangi mağazada hangi üründen kaçar tane var bilgileri mevcut. o yüzden store id bilgisi olacak, product id olacak ve ürün sayısı bilgisi olacak.

Burada product\_id sütunu olmasaydı bu tablo bize bir şey ifade eder miydi? Hayır, hiç bir şey ifade etmezdi. Demek ki Store\_id bu tabloyu tek başına bize anlatamıyor, tanımlayamıyor. Burada başka bir feature daha olması lazım. O da product\_id.

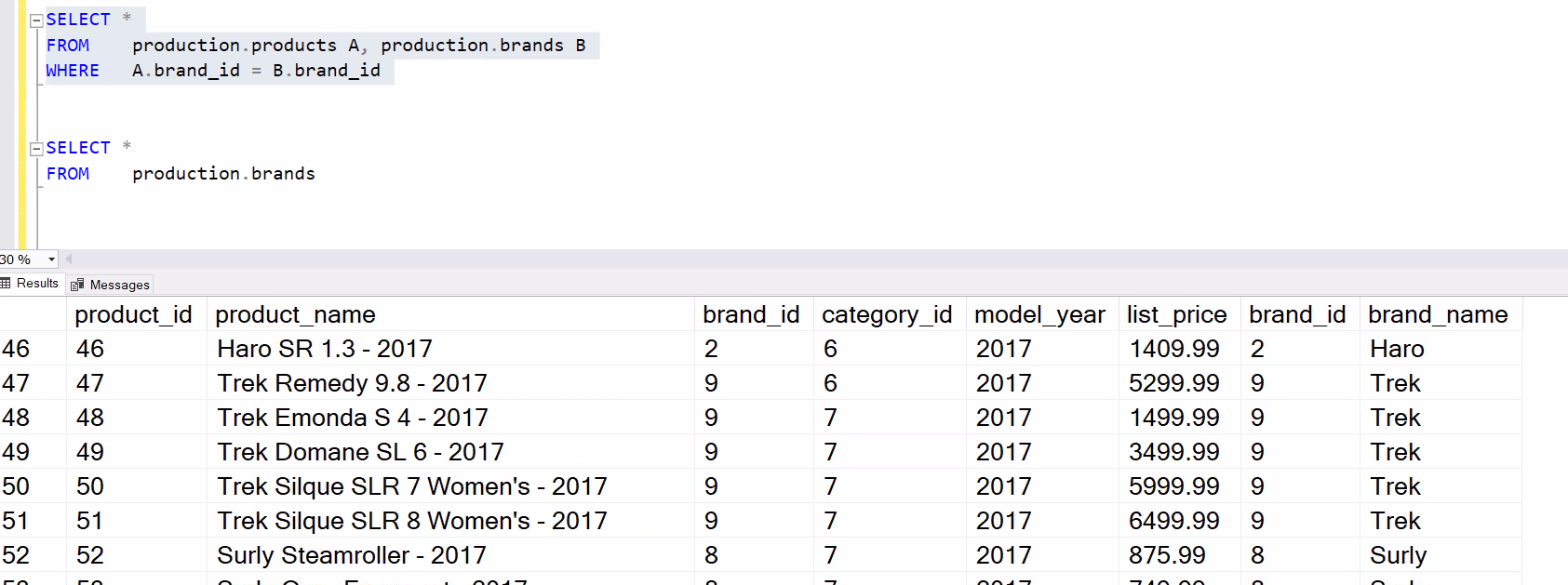
Burada realational entity gerekiyor.

--------------------------------------------------------------------------------------



Bu ürünün hangi bran’de ait olduğunu (brand\_id ile) ve hangi category ye ait olduğunu (category\_id ile) anlatıyor.

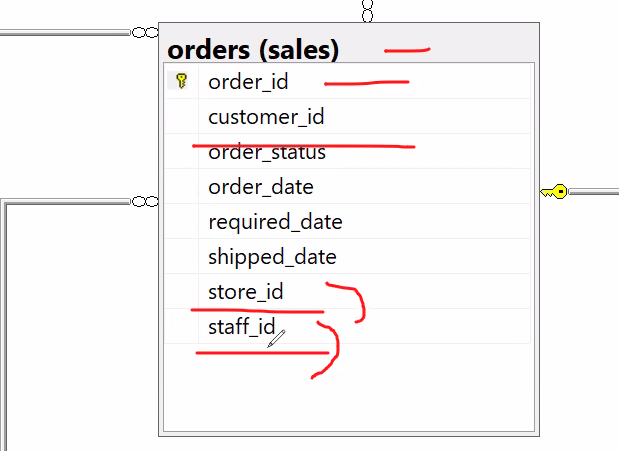
Bütün bilgileri product tablosunda tutsaydık çoklamalar olacaktı. Aşağıda gördüğün gibi brand\_name’ler çoklamış.



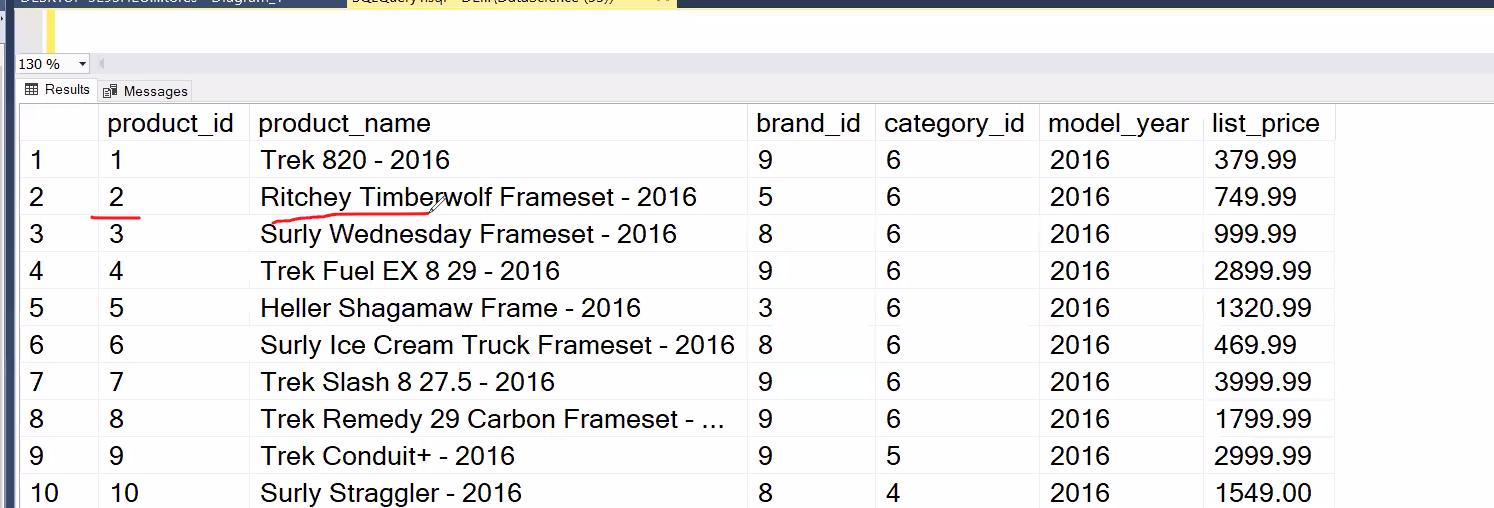
Bu alanlar (sütunlar) çoğaldıkça tekrarlarla tablom şişecekti. Değişiklik yapmak istediğimde bu bana zorluklar çıkaracaktı. Bir tabloda bütün özelliklerin olması gereksiz olacaktır performansı da düşürecektir.

İşte bunlara ANOMALİ deniyor. Ve normalizasyonu bunun için yapıyoruz. mevcut olan bütün entitileri bölüyoruz, bilgileri atomize ediyoruz (parçalara ayırıyoruz)

Aşağıdaki tabloda store\_id ve staff\_id alanlarına gerek yok ama olmasının da zararı yok. Bunları asıl staffs.sales tablosunda tutuyor.

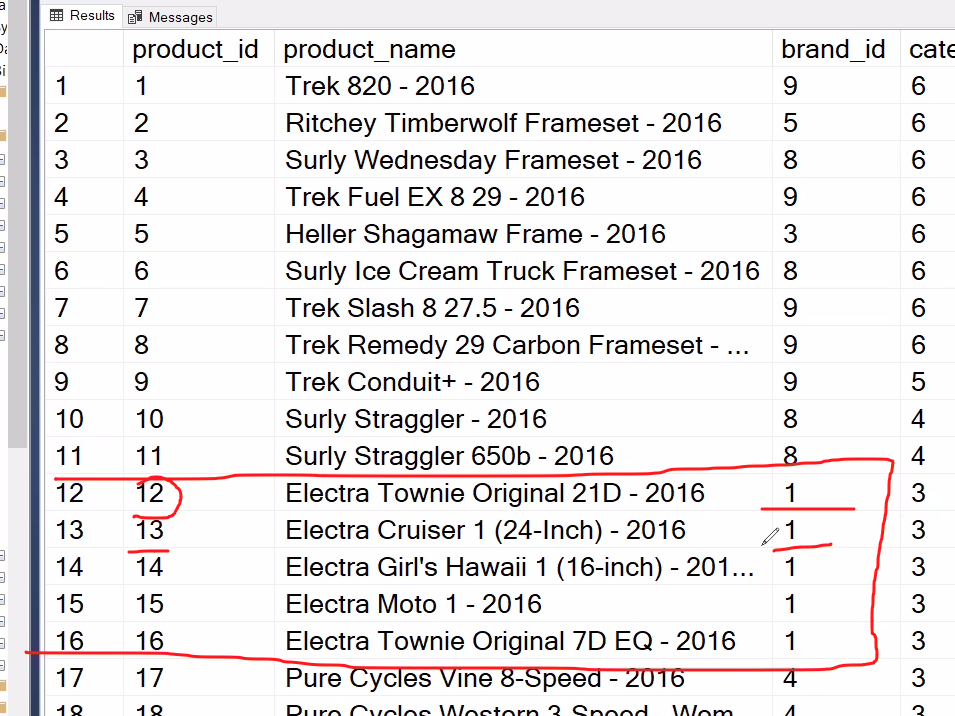


--------------------------------------------------------------------------------------



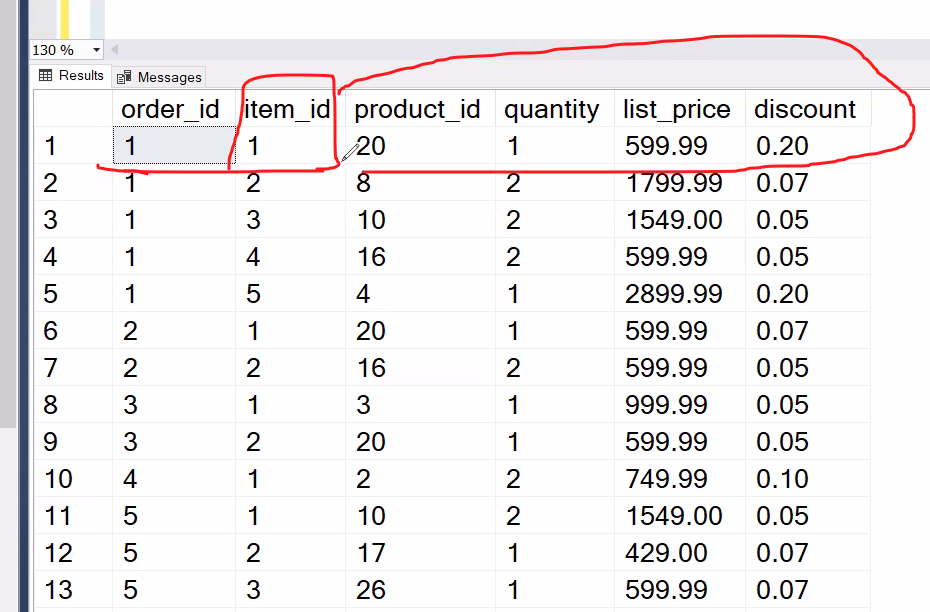
Primary key bir tabloyu temsil ediyor, tam olarak tanımlıyor olmalı ve bir satırı

Product\_name, product\_id’ye functional olarak bağlı. Yani product\_id, product\_name’i identify ediyor.

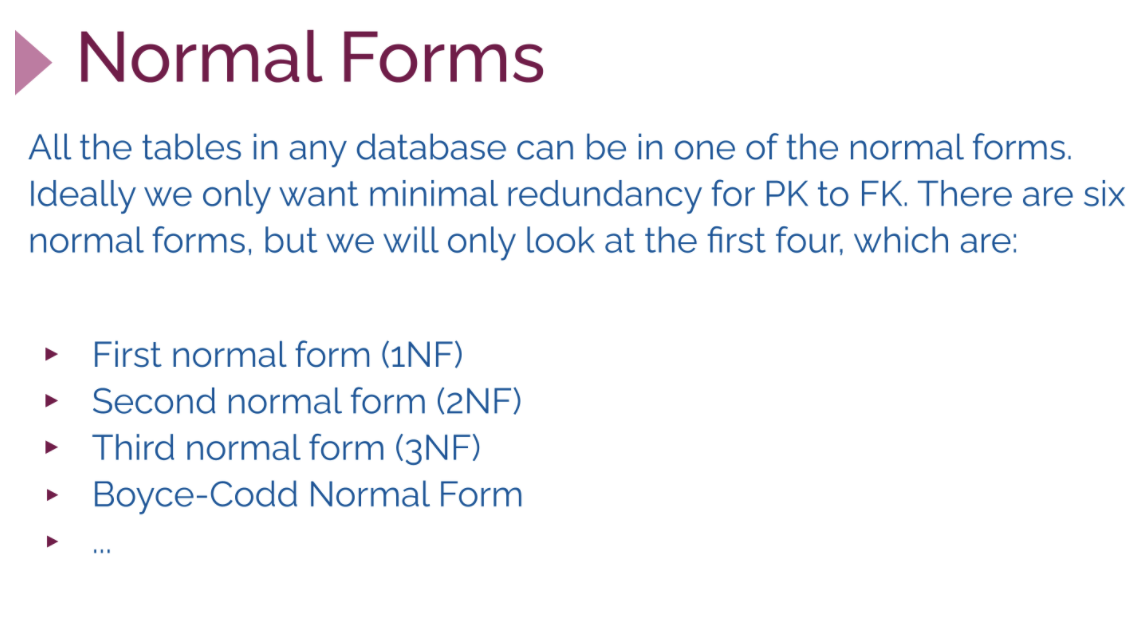


Peki product\_id, brand\_id yi identify ediyor mu? Hayır edemiyor. O zaman brand\_id fonksiyonel olarak product\_id’ye bağlı değil. Bir product\_id, bir brand\_id yi getiremiyor. 12 numaralı product\_id 1 numaralı brand\_id’ye karşılık gelirken 13 numara da yine 1 e karşılık geliyor. Bir sütunun başka bir sütunu identify etmesi demek, her bir değerine ait diğer sütunda tek bir değere işaret etmesi demek. Buradaki örnekte 12 no lu product\_id sadece bir brand\_id’yi identify etmesi lazımken edemiyor. Demek ki

---------------------------------------------------



Tek başına order\_id, o satırı temsil edemiyor. Ya da item\_id tek başına o satırı temsil edemiyor. İkisi birlikte temsile etmek zorunda. Buna composite key diyoruz.



1. Normal form.

Database deki entitileri (primary key ile) ayırıyoruz.

Her bir entitiye uygun bir primary key veriyoruz.

Her bir sütun atomik olduğunda tek bir bilgiyi ifade edebiliyor olmalı.

Mümkün olduğunda tekrar eden sütun olmayacak. Ve tekrar eden satır olmayacak.

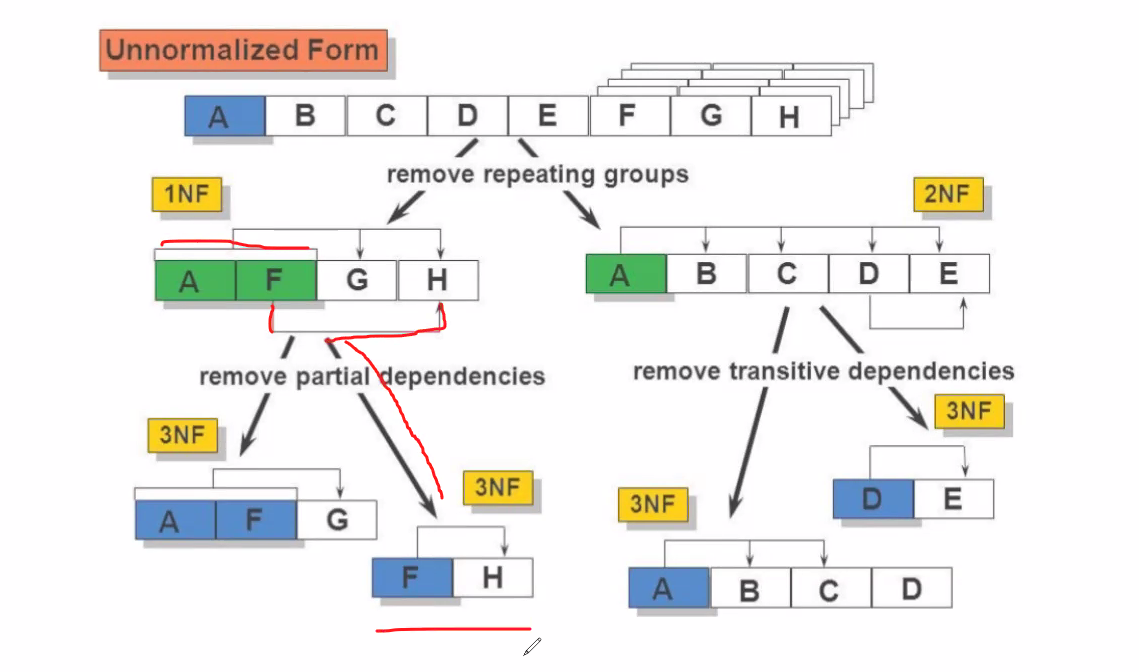
2. normal form.

Komposit key yoksa otomatik olarak ikinci normal form sağlanmış olacak. Ama komposit key varsa

3. normal form:

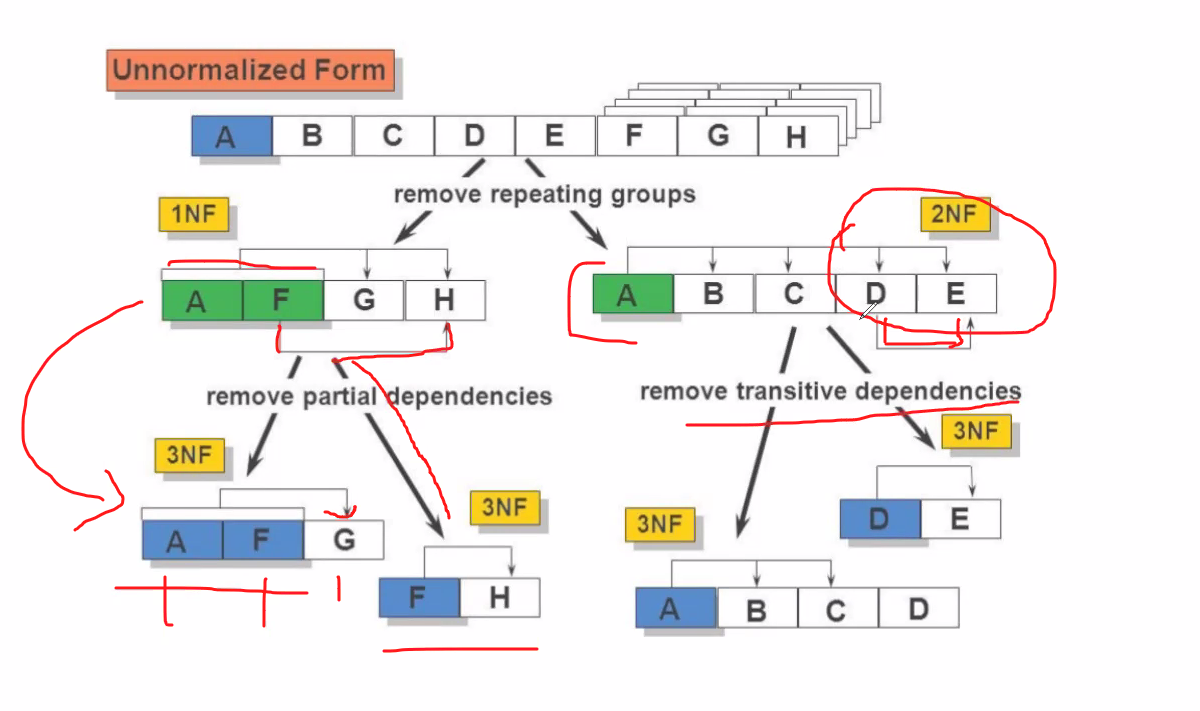
Primary key i bir kenara ayırırsak primary key olmayan diğer sütunlarda bir sütun diğer sütunlara bağlı ise bu bilgiyi bir arada tutamazsın. Bir sütun bir şeye bağlı olacaksa bu primary key olmalı. primary key olmayan başka sütunlara bağlı olmamalı.

-------------------------------------------------------------------------



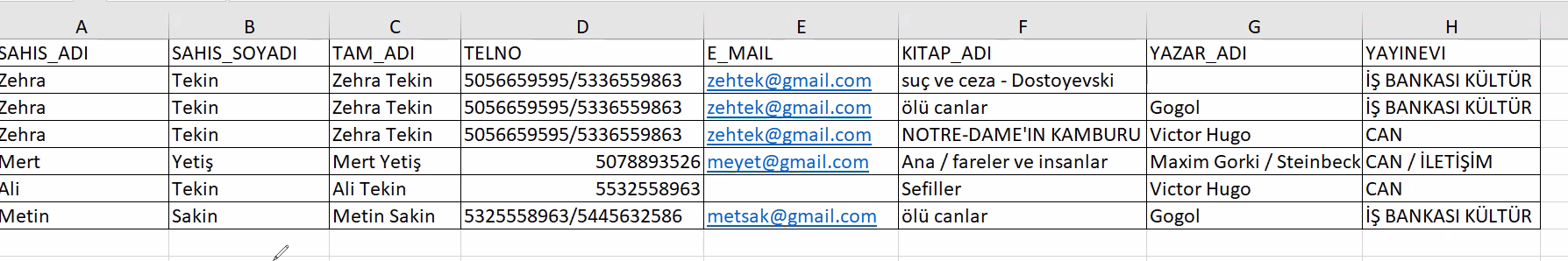
Burada 1NF tablosunda:

A ve F beraber o tabloyu (geri kalan sütunları) identify edebiliyor. Ama F tek başına H sütununu da identify edebiliyor. Fakat bütün sütunlar yalnız bir tek (primary key olan) sütuna fonksiyonel olarak bağlı olabileceğinden) Bu durumda diyoruz ki F sütunu sen ayrıl ve başka bir tabloda (3NF tablosunda) H’yi identify et. Unnormalize formda olan tabloyu normalize etmiş oluyoruz.



Burada A sütununun primary key olduğu en üst tablodaki D ve E sütunlarına dikkat ediyoruz. Ve E’nin D’ye functional dependent olduğunu görüyoruz (yani fonksiyonel olarak bağlı, D sütunu E’yi identify ediyor.) Bu duruma transitive dependency diyoruz.. Bu da istenmeyen bir durumdur. Çünkü bütün sütunlar yalnızca primary keye funcional dependency olarak bağlı olabilir. Bunlara diyoruz ki siz gidin D’nin primary key olduğu ve E’yi identify ettiği başka bir sütun oluşturun. Bu işleme de normalizasyon diyoruz

---------------------------------------------------------------------------



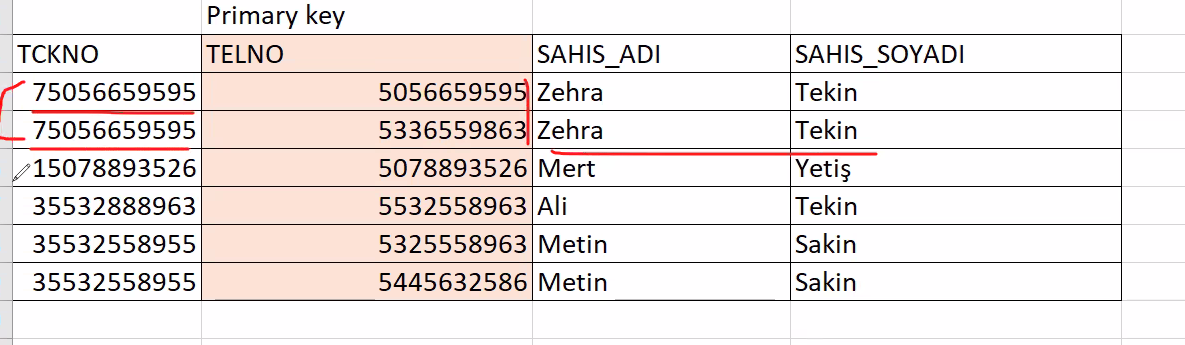
Bu database den relational database oluşturacağız. Fakat tekrarlar var, boş satırlar var. bunu first, second, thirth formlara göre normalize etmem lazım.

Önce entitileri belirliyorum. Telefon entitisi, ad-soyad entitisi, e-mail entitisi gibi.

Telefon entitisini alıp değerlendiriyorum. Telefon bilgisi bana tek bir tablo oluşturabilir mi? Telefon numaraları mevcut ama bunlar tek başına bir şey ifade etmiyor. Bunlara kime ait bunu bilmeliyim. Yeni bir tablo oluşturuyorum. Bu yeni tabloda telefon numaraları ve onların kimeler ait olduğu yer alacak.

1NF kapsamında yukardaki tablodaki tekrarlayan değerler içeren sütunların ayrı bir tabloda ifade edilmesi gerektiğinden

Telefon entitisi tablosunda primary key ne olmalı?



Primary key unique olmalı ve tüm sütunları identify edebilmeli.

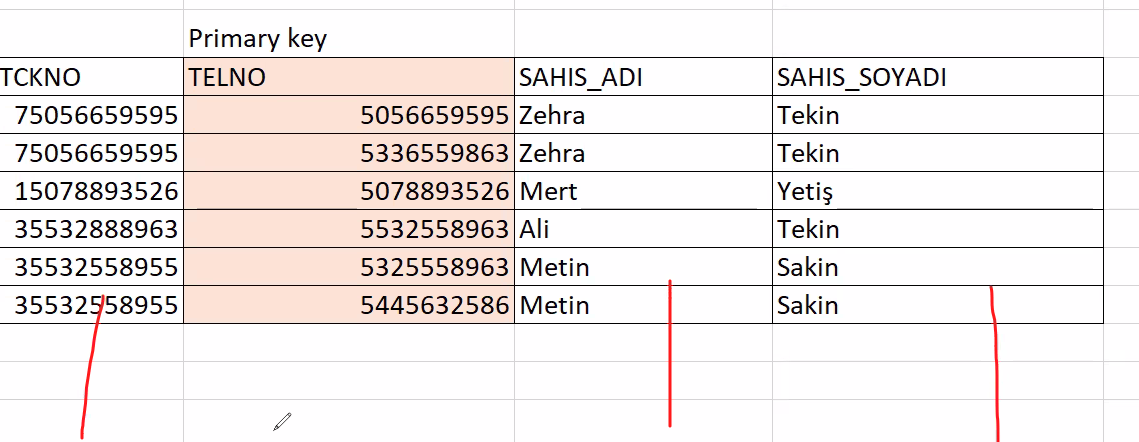
Ama Zehra Tekine ait TC kimlik numarası sütununa baktığımda bir TCno farklı iki Tel.no’yu temsil ediyor. Ama primary key olabilmesi için bir satırın yalnızca bir değeri ifade etmesi gerekiyor. Bu yüzden TC no lar primary key olamaz.

Her satırda farklı bir telefon numarası mevcut. Demek ki tel.no sütunu benim primary key özelliklerini taşıyor. Telno sütununu primary key yapıyorum.

1. Normal form:

Bu tabloda kompozit pk. Var mı? Yok. O zaman ikinci normal formu atlayıp 3. Normal forma geçebiliriz.

1. Normal form:
2. Ve 2. Normal formların gerekliliklerini sağlaması ve transitive ….o lması gerekir



Primary key dışındaki sütunların birbirine fonksiyonel olarak bağlı olmaması lazım. Ama Şahıs\_adı ve Sahıs\_soyadı sütunları TCKNolar sütununa fonksiyonel olarak bağlıdır. Bu istenmeyen durumdur. Bunlarla başka bir tablo oluşturmam ve bu yeni tablonun primary key’i TCKno lar olmalı.

Bunları ayırdığımda elimde sadece Telno’nun primary key olduğu bir tablo olarak TCKno ve Telno sütunları kalır. Normalize etmiş oldum.

------



Kitap adı sütunu yazar sütununu identify ediyor mu? Eğer identify ediyorsa tek kitap adına karşılık yalnızca bir tek yazar adı olmalı. Evet her bir kitabı bir yazar yazmıştır. Burada da her kitabın karşısında farklı bir yazar ismi yer alıyor. Kitap\_adı primary key olabilir. Fakat benim kütüphanemde iki tane ölü canlar kitabı var ama yazarı aynı. Yani satır tekrarlamış.

Peki tam tersinden bakarsak; kitap adı sütunu yazar sütununa fonksiyonel olarak bağlı değil çünkü bir yazara karşılık birden fazla kitap olabiliyor.

-----------------------------

İkinci örnekte partial dependency’i inceleyelim:



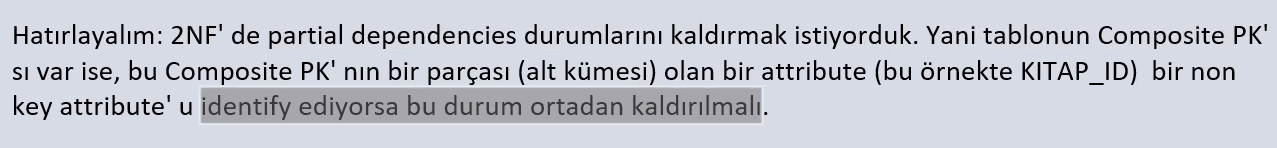
Hangi kitabı hangi kişi ödünç almış durumunu gösteren bir tablo.

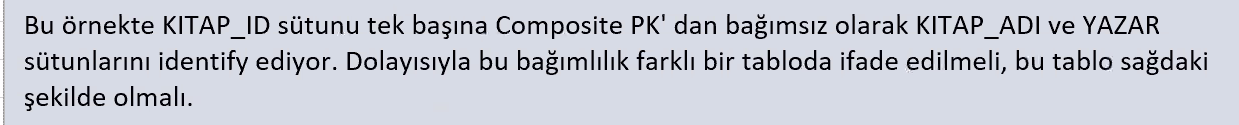
Bakalım TCKNO sütunu diğer sütunları identify edebiliyor mu? Her bir TCKNO, bir tek kitap id’yi adresleyemiyor. Farklı farklı kitapları adreslediğinde TCKNO primary key olamaz.

Tersten baktığımızda; kitap adlarından TCKNO ya baktığımızda, ölü canlar kitabı farklı zamanlarda farklı iki TCKNO ya ödünç verilmiş. Demek ki her kitap adı tek bir TCKNO’yu adresleyemiyor yani identify edemiyor. Bu durumda KITAP\_ADI sütunu primary key olamaz.

Bu durumda ben TCKNO ve KITAP\_ID sütunarını komozit key yaparım.

Kompozit primary key varsa şunlara bakalım





Kitap ID alanı, kitap\_adı alanı ile bire bir eşleşerek onu identify etmiş.

Bu durumda compozit PK’nin bir alt kümesi (bu örnekte KITAP\_ID), non primary key olan bir sütunu (KITAP\_ADI sütununu) identify ettiğinden onları ayırmam lazım!!