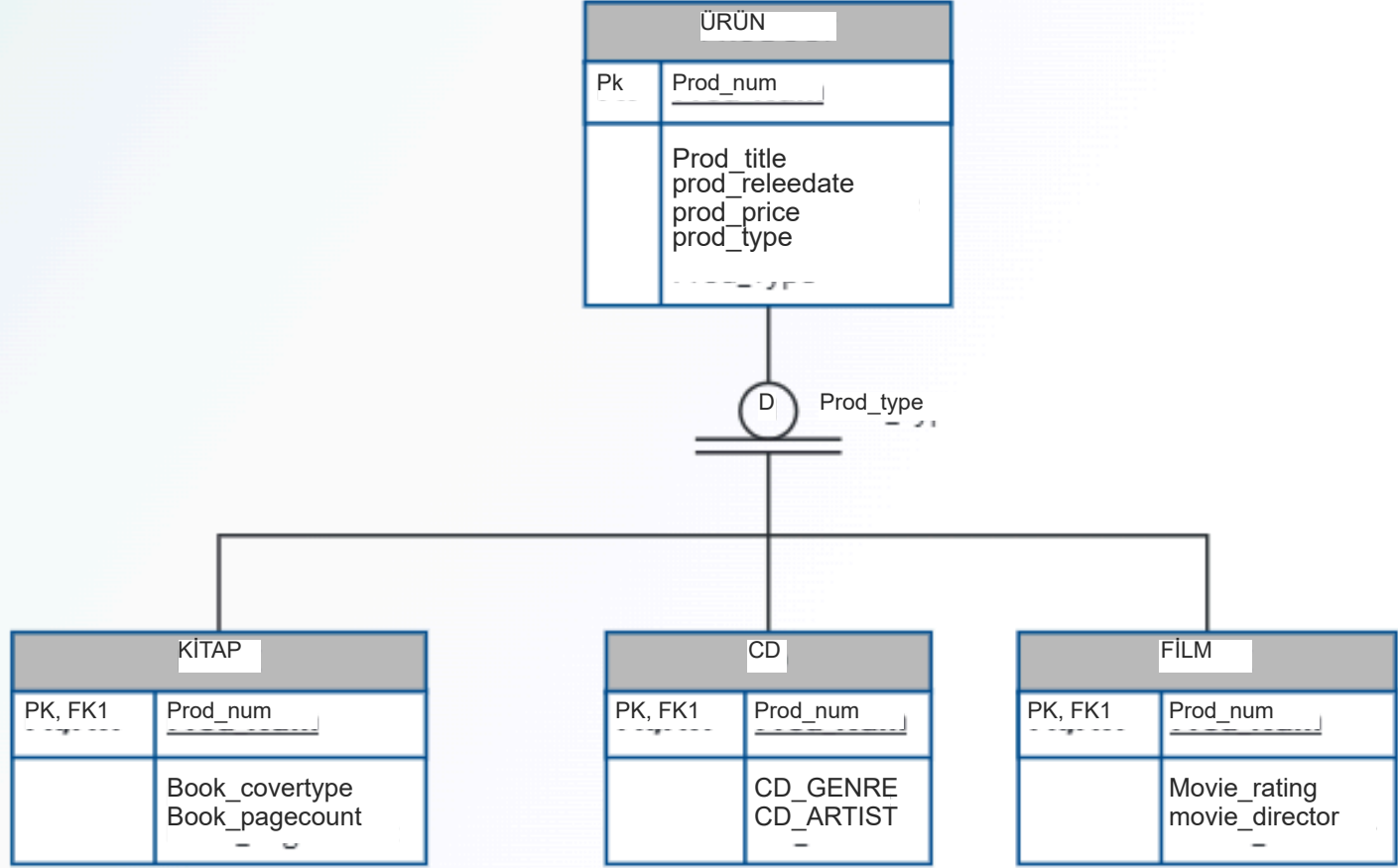


8-10. Sorular için bkz. Şekil Q5.8.

Şekil Q5.8 Ürün Veri Modeli



8. Bir filmin tüm özelliklerini listeleyin.

9. Veri modeline göre, ürün tablosundaki her varlık örneğinin CD tablosundaki bir varlık örneği ile ilişkili olması gerekiyor mu? Neden veya neden olmasın?

10. Bir kitabın ürün tablosunda görünmeden kitap masasında görünmesi mümkün müdür? Neden veya neden olmasın?

11. Bir varlık kümesi nedir ve kullanımından hangi avantajlar elde edilir?

12. Hangi birincil temel özellikler arzu edilebilir olarak kabul edilir? Her bir özelliğin neden arzu edildiğini açıklayın.

13. Hangi koşullar altında kompozit birincil anahtarlar uygundur?

14. Vekil birincil anahtar nedir ve ne zaman kullanırsınız?

15. 1: 1 ilişkisi uygularken, bir taraf zorunlu ve bir taraf isteğe bağlı ise yabancı anahtarı nereye yerleştirmelisiniz? Yabancı anahtar mandal mı yoksa isteğe bağlı mı olmalı?

16. Zamana göre değişken veriler nedir ve bu verilerle bir veritabanı tasarım açısından nasıl başa çıkarsınız? 17. En yaygın tasarım tuzağı nedir ve nasıl

meydana mı?

## Sorun

1. Aşağıdaki iş senaryosu göz önüne alındığında, uygunsa bir uzmanlık hiyerarşisi kullanarak bir karga ayağı erd oluşturun. İki bit Sondaj Şirketi, çalışanlar ve sigorta bağımlıları hakkında bilgi tutar. Her çalışanın bir çalışan numarası, adı, kiralama tarihi ve unvanı vardır. Bir çalışan bir müfettiş ise, o zaman sertifikasyon tarihi ve sertifika yenileme tarihi de sistemde kaydedilmelidir. Tüm çalışanlar için sosyal güvenlik numarası ve bağımlı isimler tutulmalıdır. Tüm bağımlılar bir ve sadece bir çalışanla ilişkilendirilmelidir. Bazı çalışanların bağımlıları olmaz, bazılarının birçok bağımlısı olacaktır.

2. Aşağıdaki iş senaryosu göz önüne alındığında, uygunsa bir uzmanlık hiyerarşisi kullanarak bir karga ayağı erd oluşturun. Tiny Hastanesi hastalar ve hastane odaları hakkında bilgi tutar. Sistem her hastaya hasta kimlik numarası atar. Ek olarak, hastanın adı ve doğum tarihi kaydedilir. Bazı hastalar en az bir tane geçiren yerleşik hastalardır

gece hastanede ve diğerleri tedavi edilen ve serbest bırakılan ayaktan hastalardır. Yerleşik hastalar bir odaya atanır. Her oda bir oda numarası ile tanımlanır. Sistem ayrıca oda türünü (özel veya yarı yarıya) ve oda ücretini depolar. Zamanla, her odada birçok hasta olacaktır. Her sakin hasta sadece bir odada kalacaktır. Her odada bir hastası olmalı ve her sakin hastada bir oda olmalı.

3. Aşağıdaki iş senaryosu göz önüne alındığında, uygunsa bir uzmanlık hiyerarşisi kullanarak bir karga ayağı erd oluşturun. Granit Satış Şirketi, çalışanlar ve çalıştıkları departmanlar hakkında bilgi tutar. Her departman için departman adı, dahili posta kutusu numarası ve ofis telefonu uzantısı saklanır. Bir departmanın atanmış birçok çalışanı olabilir ve her çalışan yalnızca bir departmana atanır. Çalışanlar maaş, saatlik veya sözleşme üzerinde çalışabilir. Tüm çalışanlara, çalışanın adı ve adresi ile birlikte tutulan bir çalışan numarası atanır. Saatlik çalışanlar için saatlik ücretler ve hedef haftalık çalışma saatleri saklanır; Örneğin, şirket bazı çalışanlar için haftada 40 saat, diğerleri için 32 ve diğerleri için 20 saat hedefleyebilir. Bazı maaşlı çalışanlar, temel maaşlarına ek olarak komisyon kazanabilecek satış görevlileridir. Tüm maaşlı çalışanlar için yıllık maaş tutarı sistemde kaydedilir. Satış görevlileri için, satışlar ve kardaki komisyon yüzdeleri sistemde depolanır. Örneğin, John, yılda 50.000 dolarlık bir taban maaşı ve yaptığı tüm satışlar için yüzde 2 komisyon ve bu satışların her birindeki kârın yüzde 5'ini daha fazla bir satış elemanıdır. Sözleşme çalışanları için, sözleşmelerinin başlangıç tarihi ve bitiş tarihi, saatleri için faturalandırma oranı ile birlikte saklanır.

4. Bölüm 4'te, “Bir Profesör birçok öğrenciye tavsiyede bulunabilir” ve “bir profesör bir departmana başkanlık edebilir” gibi iş kurallarını yansıtan küçük kolej veritabanı tasarımının oluşturulmasını gördünüz. Şekil 4.35'te gösterilen tasarımı bu iş kurallarını ekleyecek şekilde değiştirin:

- Bir çalışan personel, profesör veya yönetici olabilir. • Profesör de yönetici olabilir.

- Personel çalışanları, Seviye I veya Seviye II gibi iş düzeyinde bir sınıflandırmaya sahiptir. • Yalnızca profesörler bir departmana başkanlık edebilir. Bir bölüme sadece bir profesör başkanlık eder. • Sadece profesörler bir üniversitenin dekanı olarak görev yapabilir. Üniversitenin kolejlerinin her birine bir dekan tarafından servis edilir. • Bir profesör birçok dersi verebilir. • Yöneticilerin bir pozisyon başlığı vardır.

Bu bilgiler göz önüne alındığında, tüm birincil anahtarları, yabancı anahtarları ve ana özellikleri içeren tam ERD'yi oluşturun.

5. Tiny College, randevu tarihleri ve fesih tarihleri de dahil olmak üzere tüm idari randevularının tarihini takip etmek istiyor. (İpucu: zaman değişmez verileri iş başında.) Küçük kolej şansölyesi, 1 Ocak 1960 ve 1 Ocak 2022 arasında veya Eğitim Koleji Dekanı Kaç Dekan İşletme Koleji'nde çalıştığını bilmek isteyebilir. 1990'da. Bu bilgiler göz önüne alındığında, tüm birincil anahtarları, yabancı anahtarları ve ana özellikleri içeren tam ERD'yi oluşturun.

6. Bazı küçük üniversite personeli çalışanları bilgi teknolojisi (BT) personelidir. Bazı BT personeli akademik programlar için teknoloji desteği sağlar, bazıları teknoloji altyapısı desteği sağlar ve bazıları her ikisine de destek sağlar. BT personeli profesör değildir; Teknik uzmanlıklarını korumak için periyodik eğitim almaları gerekmektedir. Tiny College, tüm BT personel eğitimini tarih, tür ve sonuçlara göre izler (tamamlandı ve tamamlanmadı). Bu bilgiler göz önüne alındığında, tüm birincil anahtarları, yabancı anahtarları ve ana özellikleri içeren tam ERD'yi oluşturun.

7. FlyRight Company'nin (FRC) FlyRight Uçak Bakımı (FRAM) Bölümü, FRC'nin uçağı için tüm bakımı gerçekleştirir. Aşağıdaki iş kurallarını yansıtan bir veri modeli segmenti üretin:

- Tüm mekanikler FRC çalışanlarıdır. Tüm çalışanlar mekanik değildir. • Bazı mekanikler motor (EN) bakımında uzmanlaşmıştır. Diğerleri Uçak (AF) bakımında uzmanlaşmıştır

veya aviyonik (AV) bakımı. (Aviyonik, iletişim ve navigasyonda kullanılan bir uçağın elektronik bileşenleridir.) Tüm mekanikler, uzmanlık alanlarında güncel kalmak için periyodik tazeleme kursları alır. FRC, her bir tamirci tarafından alınan tüm kursları izler - Date, Ders Türü, Sertifikasyon (E/H) ve Performans.

- FRC, tüm mekaniğin istihdam geçmişini korur. Geçmiş, işe alınan tarihi, tanıtılan tarihi ve feshedilen tarihi içerir.

Bu gereksinimler göz önüne alındığında, karganın ayağı erd segmentini oluşturun.

# Dava

8. “Dövüş Sanatları R Us” (Maru) bir veritabanına ihtiyaç duyar. Maru, yüzlerce öğrencisi olan bir dövüş sanatları okuludur. Veritabanı, sunulan, her bir dersi öğretmek için atanan ve her sınıfa katılan tüm sınıfları takip etmelidir. Ayrıca, her öğrencinin ilerledikçe ilerlemesini izlemek önemlidir. Bu gereksinimler için eksiksiz bir karga ayağı oluşturun:

- Öğrencilere okula katıldıklarında bir öğrenci numarası verilir. Sayı, isimleri, doğum tarihi ve okula katıldıkları tarihle birlikte saklanır.
- Tüm öğretmenler de öğrencidir, ancak açıkça tüm öğrenciler öğretmen değildir. Normal öğrenci bilgilerine ek olarak, tüm öğretmenler için, öğretmen olarak çalışmaya başladıkları tarih, öğretmen statüsü (telafi edilmiş veya gönüllü) ile birlikte kaydedilmelidir.
- Bir öğretmen, herhangi bir sayıda sınıf öğretmek için atanabilir, ancak her sınıfta bir ve sadece bir atanmış öğretmen vardır. Bazı öğretmenler, özellikle gönüllü öğretmenler, herhangi bir sınıfa atanamaz.
- Belirli bir zamanda, haftanın gününde ve yerde belirli bir seviyeye sunulur. Örneğin, Pazartesi günleri saat 17: 00'de Oda 1'de öğretilen bir sınıf ara düzey bir sınıftır. Pazartesi günleri saat 18: 00'de Oda 1'de öğretilen bir başka sınıf yeni başlayan seviyede bir sınıftır. Oda 2'de Salı günleri 17: 00'de öğretilen üçüncü sınıf ileri düzey bir sınıftır.
- Öğrenciler her hafta uygun seviyenin herhangi bir sınıfına katılabilirler, bu nedenle herhangi bir öğrencinin belirli bir sınıf oturumuna katılacağı beklentisi yoktur. Bu nedenle, her bir sınıf toplantısına öğrencilerin katılımının izlenmesi gerekir.
- Bir öğrenci birçok farklı sınıf toplantısına katılacak ve her sınıf toplantısına normal olarak birçok öğrenci katılacak. Bazı sınıf toplantılarına herhangi bir öğrenci katılamaz. Yeni öğrenciler henüz herhangi bir sınıf toplantısına katılmamış olabilirler.
- Bir sınıfın herhangi bir toplantısında, atanan öğretmen dışındaki öğretmenler yardım için görünebilir. Bu nedenle, belirli bir sınıf toplantısında bir baş öğretmeni ve birçok yardımcı öğretmen olabilir, ancak her zaman en azından o sınıfa atanan bir öğretmen olacaktır. Her sınıf toplantısı için, sınıfın tarihi ve öğretmenlerin rolleri (baş öğretmen veya öğretmen yardımcısı) kaydedilmelidir. Örneğin, Bay Jones, Pazartesi, 17:00, Oda 1'deki ara sınıfı öğretmek üzere görevlendirildi. Bu sınıfın belirli bir toplantısında Bay Jones baş öğretmen oldu ve Bayan Chen yardımcı öğretmen olarak görev yaptı.
- Her öğrenci dövüş sanatlarında bir rütbeye sahiptir. Sıralama adı, kemer rengi ve sıralama gereksinimleri saklanır. Çoğu rütbenin sayısız sıralama gereksinimi vardır, ancak her gereksinim yalnızca belirli bir rütbe ile ilişkilidir. Beyaz kuşak dışındaki tüm rütbelerin en az bir gereksinimi vardır.
- Belirli bir rütbe birçok öğrenci tarafından tutulabilir. Bir öğrenciyi tek bir rütbeye sahip olarak düşünmek geleneksel olsa da, her öğrencinin ilerlemesini saflarda izlemek gerekir. Bu nedenle, bir öğrencinin elde ettiği her rütbe sistemde tutulur. Okula katılan yeni öğrencilere otomatik olarak beyaz kuşak rütbesi verilir. Bir öğrencinin her bir rütbe verildiği tarih, sistemde tutulmalıdır. Tüm rütbelerde bir zamanlar bu rütbeye ulaşmış en az bir öğrencisi vardır.

9. E-Ticaret Araştırma Bilgisi Dergisi prestijli bir Bilgi Sistemleri Araştırma Dergisidir. Yayın için el yazmalarını seçmek için bir akran görüntüleme işlemi kullanır. Dergiye gönderilen makalelerin sadece yüzde 10'u yayın için kabul edilmektedir. Her çeyrekte derginin yeni bir sayısı yayınlanıyor. Aşağıda açıklanan iş ihtiyaçlarını desteklemek için eksiksiz bir ERD oluşturun.

- İstenmeyen el yazmaları yazarlar tarafından sunulur. Bir makale alındığında, editör bir sayı atar ve makalenin başlığı, alındığı tarih ve “alınan” el yazması durumu da dahil olmak üzere sistemdeki bazı temel bilgiler kaydeder. Her yazarın adı, posta adresi, e -posta adresi ve bağlantı (yazarın okulu veya şirketi) dahil olmak üzere yazar (lar) hakkında bilgi de kaydedilmiştir. Her makalenin bir yazarı olmalıdır. Yalnızca sistemde el yazmaları gönderen yazarlar tutulur. Bir makalenin birkaç yazarı olması tipiktir. Tek bir yazar dergiye birçok farklı el yazması göndermiş olabilir. Ayrıca, bir makalenin birden fazla yazarı olduğunda, yazarların el yazması kredilerinde listelendiği sırayı kaydetmek önemlidir.

- En erken rahatlığı ile, editör, içeriklerinin dergi kapsamına girmesini sağlamak için makalenin konusunu kısaca gözden geçirecektir. İçerik dergi için uygun değilse, makalenin durumu “reddedilir” olarak değiştirilir ve yazar e -posta yoluyla bildirilir. İçerik dergi kapsamı içindeyse, editör makaleyi gözden geçirmek için üç veya daha fazla gözden geçiren seçer. Gözden geçirenler diğer şirketler veya teklilikler için çalışır ve bilimsel geçerliliklerini sağlamak için el yazmaları okurlar. Her gözden geçiren için sistem, bir gözden geçiren numarası, ad, e -posta adresi, bağlantı ve ilgi alanları kaydeder. İlgilenilen alanlar, gözden geçirenin belirttiği önceden tanımlanmış uzmanlık alanlarıdır. İlgilenilen bir alan AS kodu ile tanımlanır ve bir açıklama içerir (örneğin, IS2003 “veritabanı modellemesi” için koddur). Bir incelemenin birçok ilgi alanı olabilir ve bir ilgi alanı birçok gözden geçirenle ilişkilendirilebilir. Tüm gözden geçirenler en az bir ilgi alanı belirtmelidir. Derginin yorumcuları olmayan bir ilgi alanına sahip olmak olağandışı, ancak mümkündür. Editör, makalenin durumunu “Gözden Geçirme” olarak değiştirecek ve hangi gözden geçirenlerin makaleyi ve her bir gözden geçiriciye gönderildiği tarihi aldığını kaydedecektir. Bir yorumcu genellikle her yıl gözden geçirmek için birkaç el yazması alacaktır, ancak yeni gözden geçirenler henüz herhangi bir el yazması almamış olabilir.

- Gözden geçirenler makaleyi en kısa sürede okuyacak ve editöre geri bildirim sağlayacaktır. Her gözden geçirenden gelen geri bildirimler, makalenin uygunluk, netlik, metodoloji ve alana katkısı için 10 puanlık bir ölçekte derecelendirmeyi ve yayın için bir öneri (kabul veya reddetme) içerir. Editör, geri bildirim alındığı tarihle birlikte alınan her inceleme için sistemdeki tüm bu bilgileri kaydedecektir. Tüm gözden geçirenler değerlendirmelerini sağladıktan sonra, editör makaleyi yayınlamaya ve statüsünü “kabul edilen” veya “reddedilen” olarak değiştirip değiştirmeyeceğine karar verecektir. Makale yayınlanacaksa, kabul tarihi kaydedilir.

- Bir el yazması yayın için kabul edildikten sonra planlanmalıdır. Derginin her sayısı için yayın dönemi (sonbahar, kış, ilkbahar veya yaz), yayın yılı, cilt ve numara kaydedilir. Bir sorun birçok el yazması içerecektir, ancak sorun bu konuda hangi adamların yayınlanacağı bilinmeden önce sistemde oluşturulabilir. Derginin yalnızca bir sayısında kabul edilen bir el yazması görünür. Her el yazması, yazı tipleri, yazı tipi boyutu, satır aralığı, gerekçelendirme vb. Makale türü yapıldıktan sonra, sayfa sayısı sisteme kaydedilir. Editör daha sonra, kabul edilen her makalede hangi sorunun görüneceğine ve her sayıdaki el yazmalarının sırasına karar verecektir. Her makalenin sipariş ve başlangıç sayfası numarası sistemde saklanmalıdır. Makalenin bir sorun için planlandıktan sonra, makalenin durumu “planlanmış” olarak değiştirilir. Bir sorun yayınlandıktan sonra, sorunun yazdırma tarihi kaydedilir ve bu sayıdaki her makalenin durumu “yayınlanmış” olarak değiştirilir.

10. Global Birleşik Teknoloji Satışları (Guts), çalışan hesaplama için “kendi cihazınızı getir” (BYOD) modeline doğru ilerliyor. Çalışanlar ofislerinde geleneksel masaüstü bilgisayarları kullanabilirler. Ayrıca tabletler, akıllı telefonlar ve dizüstü bilgisayarlar gibi çeşitli kişisel mobil bilgi işlem cihazlarını da kullanabilirler. Yeni bilgi işlem modeli, Guts'un ele almaya çalıştığı bazı güvenlik riskleri getiriyor. Şirket, sunucularına bağlanan herhangi bir cihazın bilgi teknolojisi departmanı tarafından uygun şekilde kaydedilmesini ve onaylanmasını sağlamak istiyor. Aşağıdaki iş ihtiyaçlarını desteklemek için tam bir ERD oluşturun:

- Her çalışan, departman kodu, adı, posta kutusu numarası ve telefon numarası olan bir departmanda çalışır. En küçük departmanın şu anda 5 çalışanı var ve en büyük departmanın 40 çalışanı var. Bu sistem yalnızca bir çalışanın şu anda hangi departmanda çalıştığını izleyecektir. Çok nadiren, şirket içinde yeni bir departman oluşturulabilir. Bu zamanlarda, departman herhangi bir çalışan olmadan geçici olarak var olabilir. Her çalışan için, bir çalışan numarası ve adı (ilk, son ve orta başlangıç) sistemde kaydedilir. Her çalışanın unvanını korumak da gerekir.

- Bir çalışanın sisteme kayıtlı birçok cihaz olabilir. Her cihaza kayıtlı olduğunda bir kimlik numarası atanır. Çoğu çalışanın en az bir cihazı vardır, ancak yeni işe alınan çalışanların başlangıçta kayıtlı herhangi bir cihazı olmayabilir. Her cihaz için marka ve modelin kaydedilmesi gerekir. Yalnızca bir çalışana kayıtlı cihazlar sistemde olacaktır. Olası olmasa da, bir cihazın bir çalışandan diğerine aktarılması mümkündür. Ancak, bu olursa, yalnızca şu anda cihazın sahibi olan çalışan sistemde izlenir. Sisteme bir cihaz kaydedildiğinde, bu kaydın tarihinin kaydedilmesi gerekir.



- Cihazlar, bir şirket ofisinde veya mobil cihazlarda bulunan masaüstü sistemleri olabilir. Masaüstü cihazları şirket tarafından tipik olarak sağlanır ve şirket ağının kalıcı bir parçası olması amaçlanmıştır. Bu nedenle, her masaüstü aygıtına statik bir IP adresi atanır ve bilgisayar donanımı için MAC adresi sistemde tutulur. Bir masaüstü cihazı statik bir konumda tutulur (bina adı ve ofis numarası). Bu konum ayrıca sistemde tutulmalıdır, böylece cihaz tehlikeye girerse, BT departmanı sorunu çözmek için birini gönderebilir.

- Mobil cihazlar için, cihazın kullandığı işletim sistemini (OS) ve işletim sisteminin sürümünü de yakalamak önemlidir. BT departmanı ayrıca her mobil cihazın bir ekran kilidi etkin olduğunu ve veri için şifreleme etkinleştirildiğini doğrulamaktadır. Sistem, her mobil cihazın bu özelliklerin etkin olup olmadığı hakkında bilgi depolamayı desteklemelidir.

- Sisteme bir cihaz kaydedildikten ve bir mobil cihaz ise uygun özellikler etkinleştirildikten sonra, cihaz bir veya daha fazla sunucuya bağlantılar için onaylanabilir. Tüm cihazlar ilk başta onaylanacak gereksinimleri karşılamaz, bu nedenle cihaz herhangi bir sunucuya bağlanması onaylanmadan önce bir süre sistemde olabilir. Guts'un bir takım sunucuları vardır ve her sunucu için ayrı ayrı bir cihaz onaylanmalıdır. Bu nedenle, tek bir cihazın birkaç sunucu için onaylanması mümkündür, ancak tüm sunucular için onaylanmaz.

- Her sunucunun bir ad, marka ve IP adresi vardır. BT departmanının tesisleri içinde, fiziksel sunucuların bulunabileceği bir dizi iklim ile ilgili sunucu odası bulunmaktadır. Her sunucunun hangi odada olduğu da kaydedilmelidir. Ayrıca, her sunucuda hangi işletim sisteminin kullanıldığını izlemek gerekir. Bazı sunucular sanal sunuculardır ve bazıları fiziksel sunuculardır. Bir sunucu sanal bir sunucu ise, sistem hangi fiziksel sunucuyu çalıştırdığını izlemelidir. Tek bir fiziksel sunucu birçok sanal sunucuyu barındırabilir, ancak her sanal sunucu yalnızca bir fiziksel sunucuda barındırılır. Yalnızca fiziksel sunucular sanal bir sunucu barındırabilir. Başka bir deyişle, bir sanal sunucu başka bir sanal sunucuyu barındıramaz. Tüm fiziksel sunucular sanal bir sunucu barındırmaz.

- Bir sunucu normalde sunucuya erişmek için onaylanmış birçok cihaza sahip olacaktır, ancak henüz onaylanmış cihazları olmayan yeni sunucuların oluşturulması mümkündür. Bir cihaz bir sunucuya bağlantı için onaylandığında, bu onay tarihi kaydedilmelidir. Bir sunucunun onayını kaybetmesi için onaylanmış bir cihazın da mümkündür. Bu olursa, onayın kaldırıldığı tarih kaydedilmelidir. Bir cihaz onayını kaybederse, kaldırmaya yol açan koşulların çözülmesi durumunda bu onay daha sonraki bir tarihte geri kazanabilir.

- Sunucu, e -posta, sohbet, ödev yöneticileri ve diğerleri gibi birçok kullanıcı hizmeti sağlayabilir. Bir sunucudaki her hizmetin benzersiz bir kimlik numarası ve adı vardır. Bağışsıkların bu hizmeti sunmaya başladığı tarih kaydedilmelidir. Yeni sunucular başlangıçta herhangi bir hizmet sunmayabilir olsa da, her hizmet yalnızca bir sunucuda çalışır. İstemci tarafı hizmetleri bu sistemde izlenmez, bu nedenle her hizmet bir sunucu ile ilişkilendirilmelidir.

- Çalışanlar, kullanmadan önce bir hizmete erişme izni almalıdır. Çoğu çalışanın çok çeşitli hizmetler kullanma izinleri vardır, ancak yeni çalışanların herhangi bir hizmette izni olmayabilir. Her hizmet, onaylanmış birden fazla çalışanın kullanıcı olarak destekleyebilir, ancak yeni hizmetlerin ilk başta onaylanmış kullanıcıları olmayabilir. Çalışanın bir hizmet kullanmak için onaylandığı tarih, sistem tarafından izlenir. Bir çalışanın bir hizmete erişmesi için ilk kez onaylandığında, çalışanın bir kullanıcı adı ve şifre oluşturması gerekir. Bu, çalışanın sonunda onaylandığı her hizmet için kullanacağı kullanıcı adı ve şifre ile aynı olacaktır.

11. Global Bilgisayar Çözümleri (GCS), Amerika Birleşik Devletleri'nde birçok ofisi olan bir bilgi teknolojisi danışmanlığı şirkettir. Şirketin başarısı, kaynaklarını en üst düzeye çıkarma yeteneğine, yani yüksek vasıflı çalışanları bölgeye göre projelerle eşleştirme yeteneğine dayanmaktadır. Projelerini daha iyi yönetmek için GCS, GCS yöneticilerinin müşterilerini, çalışanlarını, projelerini, proje programlarını, ödevlerini ve faturaları takip edebilmeleri için bir veritabanı tasarlamak için sizinle iletişime geçti.

GCS veritabanı tüm GCS'nin operasyonlarını ve bilgi gereksinimlerini desteklemelidir. Ana varlıkların temel bir açıklaması şu:

- GCS çalışanlarının bir çalışan kimliği, bir soyadı, orta bir başlangıç, bir ad, bir bölge ve sistemde kaydedilen bir kiralama tarihi olmalıdır.

- Geçerli bölgeler aşağıdaki gibidir: Northwest (NW), Güneybatı (SW), Midwest North (MN), Midwest South (MS), Kuzeydoğu (NE) ve Güneydoğu (SE).
- Her çalışanın birçok becerisi vardır ve birçok çalışan aynı beceriye sahiptir. • Her becerinin bir beceri kimliği, açıklaması ve ödeme oranı vardır. Geçerli beceriler aşağıdaki gibidir: Veri Girişi I, Veri Girişi II, Sistemler

Analist I, Sistemler Analyst II, Veritabanı Tasarımcısı I, Veritabanı Tasarımcı II, Java I, Java II, C11 I, C11 II, Python I, Python II, Coldfusion I, Coldfusion II, ASP I, ASP II, ORACLE DBA, MS SQL Sunucu DBA, Ağ Mühendisi I, Ağ Mühendisi II, Web Yöneticisi, Teknik Yazar ve Proje Yöneticisi. Tablo P5.11a, beceri envanterinin bir örneğini göstermektedir.

Tablo P5.11a

Yetenek	Çalışan
Veri Girişi I	Seaton, Amy M.; Williams, Josh R.; Underwood, Trish L.
Veri Girişi II	Williams, Josh R.; Seaton, Amy M.
Sistem Analisti I	Craig, Brett M.; Sewell, Beth Z.; Robbins, Erin E.; Bush, Emily; Zebras, Steve
Systems Analyst II	Chandler, Joseph E.; Burklow, Shane C.; Robbins, Erin E.
DB Tasarımcı I	Yarbrough, Peter B.; Smith, Mary A.
DB Designer II	Yarbrough, Peter B.; Pascoe, Jonathan L.
Java içinde	Kattan, Chris N.; Ephanor, Victor; Summers, Anna; Ellis, Maria R.
Java II	Kattan, Chris N.; Ephanor, Victor; Batts, Melissa D.
C11 I	Smith, Jose F.; Rogers, Adam S.; Cope, Leslie R.
C11 II	Rogers, Adam S.; Bible, Hanah M.
Python'da	Zebras, Steve; Ellis, Maria R.
Python 2	Zebras, Steve; Newton, Christopher S.
Coldfusion I	Duarte, Miriam; Bush, Emily
Coldfusion II	Bush, Emily; Newton, Christopher S.
Asp I	Duarte, Miriam; Bush, Emily
ASP II	Duarte, Miriam; Newton, Christopher S.
Oracle DBA	Smith, Jose F.; Pascoe, Jonathan L.
SQL Server DBA	Yarbrough, Peter B.; Smith, Jose F.
Ağ Mühendisi I	Bush, Emily; Smith, Mary A.
Ağ Mühendisi II	Bush, Emily; Smith, Mary A.
Web Yöneticisi	Bush, Emily; Smith, Mary A.; Newton, Christopher S.
Teknik yazar	Kilby, Surgena T.; Bender, Larry A.
Proje Müdürü	Paine, Brad S.; Mudd, Roger T.; Kenyon, Tiffany D.; Connor, Sean

- GCS'nin birçok müşterisi var. Her müşterinin müşteri kimliği, adı, telefon numarası ve bölgesi vardır. • GCS projelerle çalışıyor. Bir proje, müşteri ve GC'ler arasında tasarlama, geliştirme ve

Bilgisayarlı bir çözüm uygulayın. Her projenin proje kimliği, projenin ait olduğu müşteri, kısa bir açıklama, bir proje tarihi (sözleşmenin imzalandığı tarih), tahmini bir proje başlangıç tarihi ve bitiş tarihi, tahmini bir proje bütçesi, bir Gerçek başlangıç tarihi, gerçek bir bitiş tarihi, gerçek bir maliyet ve projenin yöneticisi olarak atanan bir çalışan.

- Projenin gerçek maliyeti, o haftanın maliyeti gerçek maliyete ekleyerek her Cuma güncellenir. Haftanın maliyeti, her çalışanın bu beceri için ödeme oranı ile çalıştığı saatleri çarparak hesaplanır.

- Projenin yöneticisi olan çalışan, etkili bir tasarım ve geliştirme planı olan bir proje programını tamamlamalıdır. Proje programında (veya planında) yönetici, projeyi baştan sona götürmek için gerçekleştirilecek görevleri belirlemelidir. Her görevin bir görev kimliği, kısa bir görev açıklaması, başlangıç ve bitiş tarihleri, gerekli beceri türleri ve görevi tamamlamak için gereken çalışan sayısı (gerekli becerilerle) vardır. Genel görevler ilk görüşme, veritabanı ve sistem tasarımı, uygulama, kodlama, test ve nihai değerlendirme ve imzalama. Örneğin, GC'ler Tablo P5.11b'de gösterilen proje programına sahip olabilir.

**Tablo  
P5.11b**

Proje Kimliği: 1		Açıklama: Satış Yönetim Sistemi		
Şirket: Rocks'a bakın		Sözleşme Tarihi: 2/12/2022		Bölge: KB
Başlangıç Tarihi: 3/1/2022		Bitiş Tarihi: 7/1/2022		Bütçe: 15.500 \$
Başlangıç Tarihi	Bitiş tarihi	Görev Tanımı	Gerekli beceri (ler)	Gerekli miktar
3/1/22	3/6/22	İlk görüşme	Proje Müdürü	1
			Systems Analyst II	1
			DB Tasarımcı I	1
3/11/22	3/15/22	Veritabanı tasarımı	DB Tasarımcı I	1
3/11/22	4/12/22	Sistem tasarımı	Systems Analyst II	1
			Sistem Analisti I	2
3/18/22	3/22/22	Veritabanı uygulaması	Oracle DBA	1
3/25/22	5/20/22	Sistem Kodlama ve Test	Java içinde	2
			Java II	1
			Oracle DBA	1
3/25/22	6/7/22	Sistem belgeleri	Teknik yazar	1
6/10/22	6/14/22	Nihai Değerlendirme	Proje Müdürü	1
			Systems Analyst II	1
			DB Tasarımcı I	1
			Java II	1
6/17/22	6/21/22	Yerinde Sistem Çevrimiçi ve Veri Yükleme	Proje Müdürü	1
			Systems Analyst II	1
			DB Tasarımcı I	1
			Java II	1
7/1/22	7/1/22	Bitirmek	Proje Müdürü	1

- GCS tüm çalışanlarını bölgeye göre havuzlar; Bu havuzdan, çalışanlar proje yöneticisi tarafından planlanan belirli bir göreve atanır. Örneğin, ilk projenin programında, 3/1/22 ile 3/6/22 arasındaki dönem için bir System Analyst II, Veritabanı Tasarımcısı I ve Proje Yöneticisi'nin gerekli olduğunu biliyorsunuz. Proje yöneticisi proje oluşturulduğunda atanır ve proje süresince kalır. Bu bilgileri kullanarak GCS, müşteri ile aynı bölgede bulunan çalışanları arar, gerekli becerilerle eşleşir ve çalışanları proje görevine atar.

- Her proje programı görevinin kendisine atanmış birçok çalışanı olabilir ve belirli bir çalışan birden fazla proje görevi üzerinde çalışabilir. Ancak, bir çalışan aynı anda yalnızca bir proje görevi üzerinde çalışabilir. Örneğin, bir çalışan zaten 2/20/22 - 3/3/22 tarihleri arasında bir proje görevi üzerinde çalışmaya atanmışsa, mevcut ödev kapatılana (uçlar) kadar başka bir görev üzerinde çalışamaz. Bir ödevin kapatıldığı tarih, bir görev programın önünde veya arkasında bir görev tamamlanabilir çünkü proje programı görevinin bitiş tarihiyle eşleşmez.

- Önceki tüm bilgiler göz önüne alındığında, ödevin bir çalışanı proje programını kullanarak bir proje göreviyle ilişkilendirdiğini görebilirsiniz. Bu nedenle, ödevi takip etmek için, en azından aşağıdaki bilgilere ihtiyacınız vardır: ödev kimliği, çalışan, proje programı görevi, ödev başlangıç tarihi ve ödev bitiş tarihi. Bazı projeler programın önünde veya arkasında çalıştığı için bitiş tarihi herhangi bir tarih olabilir. Tablo P5.11c bir örnek atama formunu göstermektedir.

**Tablo**  
**P5.11c**

Proje Kimliği: 1			Açıklama: Satış Yönetim Sistemi			
Şirket: Rocks'a bakın			Sözleşme Tarihi: 2/12/2022			Ara: 03/29/22
Planlanmış				Gerçek ödevler		
Proje görevi	Başlangıç Tarihi	Bitiş tarihi	Yetenek	Çalışan	Başlangıç Tarihi	Bitiş tarihi
İlk görüşme	3/1/22	3/6/22	Proje Mgr.	101-Connor, S.	3/1/22	3/6/22
			Sys. Analyst II	102-Burklow, S.	3/1/22	3/6/22
			DB Tasarımcı I	103-Smith, M.	3/1/22	3/6/22
Veritabanı tasarımı	3/11/22	3/15/22	DB Tasarımcı I	104-Smith, M.	3/11/22	3/14/22
Sistem tasarımı	3/11/22	4/12/22	Sys. Analyst II	105-Burklow, S.	3/11/22	
			Sys. Analist I	106-Bush, E.	3/11/22	
			Sys. Analist I	107-Zebras, S.	3/11/22	
Veritabanı uygulaması	3/18/22	3/22/22	Oracle DBA	108-Smith, J.	3/15/22	3/19/22
Sistem Kodlama ve Test	3/25/22	5/20/22	Java içinde	109-Summers, A.	3/21/22	
			Java içinde	110-Ellis, M.	3/21/22	
			Java II	111-Okkanor, V.	3/21/22	
			Oracle DBA	112-Smith, J.	3/21/22	
Sistem belgeleri	3/25/22	6/7/22	Teknoloji. Yazar	113-Kilby, S.	3/25/22	
Nihai Değerlendirme	6/10/22	6/14/22	Proje Mgr. Sys. Analyst II DB Designer I Java II			
Yerinde Sistem Çevrimiçi ve Veri Yükleme	6/17/22	6/21/22	Proje Mgr. Sys. Analyst II DB Designer I Java II			
Bitirmek	7/1/22	7/1/22	Proje Mgr.			

(Not: Ödev numarası, çalışan adının bir öneki olarak gösterilmiştir - örneğin 101 veya 102.) Daha önce gösterilen ödevlerin bu tasarım tarihinden itibaren tek ödevler olduğunu varsayalım. Atama numarası, veritabanı tasarımınızla eşleşen herhangi bir sayı olabilir.

- Çalışan çalışma saatleri, belirli bir görevde çalışanlar tarafından yapılan gerçek saatlerin kaydını içeren bir çalışma günlüğünde tutulur. İş günlüğü, çalışanın her haftanın sonunda (Cuma) veya her ayın sonunda doldurduğu bir formdur. Form, bir Cuma günü düşmezse, ayın mevcut Cuma günü veya ayın son iş günü olan tarihi içeriyor. Form ayrıca ödev kimliğini, o hafta veya ay sonuna kadar çalışılan toplam saatleri ve çalışma-log girişinin tahsil edildiği fatura numarasını içerir. Açıkçası, her çalışma log girişi sadece bir fatura ile ilişkili olabilir. İlk örnek projesi için geçerli çalışma-log girişlerinin örnek listesi Tablo P5.11d'de gösterilmiştir.



- Son olarak, her 15 günde bir, bir fatura yazılır ve o dönemde proje üzerinde çalışan toplam saatler için müşteriye gönderilir. GCS bir fatura oluşturduğunda, faturanın bir parçası olan çalışma-log girişlerini güncellemek için fatura numarasını kullanır. Özetle, bir fatura birçok çalışma-log girişine başvurabilir ve her çalışma-log girişi yalnızca bir fatura ile ilişkili olabilir. GCS, ilk proje için 3/15/22 tarihinde bir fatura gönderdi (bkz. Kayalara bakınız), 3/1/22 ve 3/15/22 arasında çalıştı. Bu nedenle, bu tabloda sadece bir fatura olduğunu ve faturanın önceki formda gösterilen çalışma-log girişlerini kapsadığını güvenle varsayabilirsiniz.

**Tablo**  
**P5.11d**

Çalışan Adı	Hafta Sonu	Atama numarası	Saatler çalıştı	Fatura numarası
Burklow, S.	3/1/22	1-102	4	xxx
Connor, S.	3/1/22	1-101	4	xxx
Smith, M.	3/1/22	1-103	4	xxx
Burklow, S.	3/8/22	1-102	24	xxx
Connor, S.	3/8/22	1-101	24	xxx
Smith, M.	3/8/22	1-103	24	xxx
Burklow, S.	3/15/22	1-105	40	xxx
Bush, E.	3/15/22	1-106	40	xxx
Smith, J.	3/15/22	1-108	6	xxx
Smith, M.	3/15/22	1-104	32	xxx
Zebras, S.	3/15/22	1-107	35	xxx
Burklow, S.	3/22/22	1-105	40	
Bush, E.	3/22/22	1-106	40	
Ellis, M.	3/22/22	1-110	12	
Ephanor, V.	3/22/22	1-111	12	
Smith, J.	3/22/22	1-108	12	
Smith, J.	3/22/22	1-112	12	
Summers, A.	3/22/22	1-109	12	
Zebras, S.	3/22/22	1-107	35	
Burklow, S.	3/29/22	1-105	40	
Bush, E.	3/29/22	1-106	40	
Ellis, M.	3/29/22	1-110	35	
Ephanor, V.	3/29/22	1-111	35	
Kilby, S.	3/29/22	1-113	40	
Smith, J.	3/29/22	1-112	35	
Summers, A.	3/29/22	1-109	35	
Zebras, S.	3/29/22	1-107	35	

Not: xxx fatura kimliğini temsil eder. Veritabanınızdaki fatura numarasına uygun olanı kullanın.

Göreviniz, bu problemde açıklanan işlemleri yerine getiren bir veritabanı oluşturmaktır. Gerekli minimum varlıklar çalışan, beceri, müşteri, bölge, proje, proje programı, ödev, iş günlüğü ve faturadır. (Listelenmeyen ek gerekli varlıklar vardır.)

- Gerekli tüm tabloları ve gerekli ilişkileri oluşturun.
- Yetkililer birincil anahtarları kullanırken varlık bütünlüğünü korumak için gerekli dizinleri oluşturun.
- Örnek verilerinde ve formlarında belirtildiği gibi tabloları gerektiği gibi doldurun.

# 6

## bölüm

# Veritabanı tablolarının normalleştirilmesi

### Öğrenme Hedefleri

Bu bölümü tamamladıktan sonra şunları yapabilirsiniz:

- |     |   |     |  |
|-----|---|-----|--|
| 6-1 | Veritabanı tasarım sürecindeki normalleşmeyi ve rolünü açıklayın  | 6-4 | Tablo yapılarını değerlendirmek ve düzeltmek için normalleştirme kurallarını uygulayın                         |
| 6-2 | Normal formların her birini tanımlayın: 1NF, 2NF, 3NF, BCNF, 4NF ve 5NF   | 6-5 | Bilgi verimli bir şekilde üretmek için denormalizasyon gerektiren durumları belirleyin                         |
| 6-3 | Normal formların daha düşük normal formlardan daha yüksek normal formlara nasıl dönüştürülebileceğini açıklayın | 6-6 | ERD'nin bir dizi minimum gereksinimi karşıladığını kontrol etmek için bir veri modeli kontrol listesi kullanın |

Önizleme İyi veritabanı tasarımı iyi tablo yapılarıyla eşleştirilmelidir. Bu bölümde, veri fazlalıklarını kontrol etmek için iyi tablo yapılarını değerlendirmeyi ve tasarlamayı öğrenecek ve böylece veri anomalilerinden kaçınacaksınız. Bu tür arzu edilen sonuçları veren süreç normalleşme olarak bilinir.

İyi bir tablo yapısının özelliklerini tanımak ve takdir etmek için fakir bir yapıyı incelemek yararlıdır. Bu nedenle, bölüm zayıf bir tablo yapısının özelliklerini ve yarattığı sorunları inceleyerek başlar. Daha sonra tablo yapısını nasıl düzelteceğinizi öğrenirsiniz. Bu metodoloji önemli temettüler verecektir: İyi bir masa yapısının nasıl tasarlanacağını ve fakir bir yapının nasıl onarılacağını bileceksiniz.

Sadece veri anomalilerinin normalleşme yoluyla ortadan kaldırılabilirliğini değil, aynı zamanda düzgün normalleştirilmiş bir tablo yapıları setinin kullanımı kimlik dışı bir setten daha az karmaşık olduğunu keşfedeceksiniz. Buna ek olarak, normalleştirilmiş tablo yapıları setinin bir kuruluşun gerçek operasyonlarını daha sadakatle yansıttığını öğreneceksiniz.

## Veri dosyaları ve kullanılabilir biçimler

	MS erişimi	Kehanet	MS SQL	MySQL
Ch06_constructco	Evet	Evet	Evet	Evet
Ch06_eval	Evet	Evet	Evet	Evet
Ch06_service	Evet	Evet	Evet	Evet

Cengage.com adresinde bulunan veri dosyaları

## 6-1 Veritabanı Tabloları ve Normalleştirme

İyi ilişkisel veritabanı yazılımına sahip olmak, Bölüm 1, veritabanı sistemlerinde tartışılan veri yedekliliğinden kaçınmak için yeterli değildir. Veritabanı tabloları bir dosya sisteminde dosyalarımı gibi tasarlanmışsa, ilişkisel veritabanı yönetim sistemi (RDBMS) hiçbir zaman üstün veri engelleme özelliklerini gösterme şansına sahip değildir.

Tablo, veritabanı tasarımının temel yapı taşıdır. Sonuç olarak, tablonun yapısı büyük ilgi çekicidir. İdeal olarak, Bölüm 4, Varlık İlişkisi (ER) modellemesinde incelenen veritabanı tasarım süreci iyi tablo yapıları verir. Ancak, iyi bir veritabanı tasarımında bile zayıf tablo yapıları oluşturmak mümkündür. Zavallı bir masa yapısını nasıl tanıyorsunuz ve iyi bir masa nasıl üretiyorsunuz? Her iki sorunun cevabı normalleşmeyi içerir. Normalleştirme, veri fazlalıklarını en aza indirmek için tablo yapılarını değerlendirmek ve düzeltmek için bir süreçtir, böylece veri anomalileri olasılığını azaltmaktadır. Normalleştirme süreci, ilişkisel veritabanı modelinde, Bölüm 3'te öğrendiğiniz belirleme ve fonksiyonel bağımlılık kavramlarına dayanarak tablolara öznitelikler atamayı içerir.

Normalizasyon, normal formlar adı verilen bir dizi aşama aracılığıyla çalışır. İlk üç aşama birinci normal form (1NF), ikinci normal form (2NF) ve üçüncü normal form (3NF) olarak tanımlanır. Yapısal bir bakış açısından, 2NF 1NF'den daha iyidir ve 3NF 2NF'den daha iyidir. İş veritabanı tasarımında çoğu amaç için, 3NF norizasyon sürecine gitmeniz gerektiği kadar yüksektir.

Normalizasyon veritabanı tasarımında çok önemli bir bileşen olsa da, en yüksek normalleştirme seviyesinin her zaman en çok arzu edilen olduğunu varsaymamalısınız. Genel olarak, normal form ne kadar yüksek olursa, belirli bir çıktı üretmek için o kadar fazla ilişkisel birleştirme işlemleri gerekir. Ayrıca, veritabanı sistemi tarafından son kullanıcı sayısına yanıt vermek için daha fazla kaynak gereklidir. Başarılı bir tasarım, hızlı performans için son kullanıcı talebini de dikkate almalıdır. Bu nedenle, performans gereksinimlerini karşılamak için zaman zaman bir veritabanı tasarımının bazı bölümlerini denormelerine yönlendirmeniz gerekecektir. Denormalizasyon daha düşük bir normal form üretir; Yani, bir 3NF denormalizasyon yoluyla 2NF'ye dönüştürülecektir. Bununla birlikte, denormalizasyon yoluyla artan araç için ödediğiniz fiyat daha fazla veri artıklığıdır.

### Not

Kelime tablosu bu bölüm boyunca kullanılmasına rağmen, resmi olarak normalizasyon ilişkileriyle ilgilidir. Bölüm 3'te, tablo ve ilişkinin terimlerinin sıklıkla birbirinin yerine kullanıldığını öğrendiniz. Aslında, bir tablonun bazı belirli koşulları karşılayan mantıksal bir ilişkinin uygulama görünümü olduğunu söyleyebilirsiniz. (Bkz. Tablo 3.1.) Bununla birlikte, daha titiz olduğu için, matematiksel ilişki yinelenen tüpürlere izin vermez; oysa tablolarda bulunabilirler (bkz. Bölüm 6-5). Ayrıca, normalleştirme terminolojisinde, en azından bir anahtarın bir parçası olan herhangi bir özellik, daha önce tanıtilen daha yaygın terim anahtar özniteliği yerine ana öznitelik olarak bilinir. Tersine, prime olmayan bir öznitelik veya anahtar olmayan bir öznitelik, herhangi bir adayın anahtarının bir parçası değildir.

Normalleştirme Veri işten çıkarılmalarının azaltılması veya ortadan kaldırılması için varlıklara öznitelikler atayan bir süreç.

Denormalizasyon Bir tablonun, genellikle işleme hızını artırmak için daha yüksek seviyeli normal bir formdan daha düşük seviyeli normal bir forma dönüştürüldüğü bir işlem. Denormalizasyon potansiyel olarak veri anomalileri verir.

prime öznitelik bir anahtar öznitelik; Yani, bir anahtarın parçası olan veya tüm anahtar olan bir özellik. Ayrıca bkz. Anahtar özellikler.

Anahtar Özellikler Birincil anahtar oluşturan öznitelikler. Ayrıca bkz. Prime özniteliği.

Prime olmayan öznitelik Anahtarın parçası olmayan bir öznitelik.

Key Olmayan Özellik Bkz. Prime olmayan öznitelik.

## 6-2 Normalleştirme ihtiyacı

Normalizasyon tipik olarak önceki bölümlerde öğrendiğiniz varlık ilişkisi modellemesi ile birlikte kullanılır. Veritabanı tasarımcıları genellikle normalizasyonu iki oturumda kullanırlar. Son kullanıcıların iş gereksinimlerine dayalı yeni bir veritabanı yapısı tasarlarlarken, veritabanı tasarımcısı Crow's Foot Notation ERD'leri gibi bir teknik kullanarak bir veri modeli oluşturabilir. İlk tasarım tamamlandıktan sonra, tasarımcı her bir varlıktaki özellikler arasındaki ilişkileri analiz etmek ve yapının normalleşme yoluyla geliştirilip geliştirilemeyeceğini belirlemek için normalleştirmeyi kullanabilir. Alternatif olarak ve daha sık, veritabanı tasarımcılarından genellikle düz dosyalar, elektronik tablolar veya eski veritabanı yapıları şeklinde mevcut veri yapılarını değiştirmeleri istenir. Yine, veri yapısındaki öznitelikler veya alanlar arasındaki ilişkileri analiz ederek, veritabanı tasarımcısı mevcut veri yapısını geliştirmek ve uygun bir veritabanı tasarımı oluşturmak için normalleştirme işlemini kullanabilir. İster yeni bir veritabanı yapısı tasarlıyor olun, ister mevcut bir yapıyı değiştiriyor olun, normalleştirme işlemi aynıdır.

Sadece normalleştirme kullanarak tamamen yeni bir veritabanı tasarlamak çok nadirdir. Genellikle, iş kurallarını ve veri kısıtlamalarını tanımlayarak ve önceki bölümlerde öğrendiğiniz teknikleri kullanarak işlevsel bağımlılıkları, varlıkları ve nitelikleri tanımlayarak başlarsınız. Ardından, modeli doğrulamak ve daha da geliştirmek için normalleştirme kavramlarını uygularsınız.

Bu bölüm kitaptaki en kritik bölümlerden biridir, çünkü burada daha önce öğrendiğiniz kavramların, veri bütünlüğü kısıtlamalarını, kullanıcı raporlama ve performans gereksinimlerini karşılayan bir veritabanını modellemek için veritabanı tasarımında nasıl birlikte çalıştığını öğreneceksiniz. Bu kavramlar şunları içerir:

- İş Kurallarını Belirleme
- İş ve veri kısıtlamalarını tanımlamak ve tanımlamak
- İşlevsel bağımlılıkların tanımlanması
- Varlıkları ve ilişkileri tanımlamak
- Çok Değişken Özellikleri ortadan kaldırma

Normalizasyonun temel amacı, gereksiz veya istenmeyen veri fazlalıklarını ortadan kaldırarak veri anomalilerini ortadan kaldırmaktır. Veritabanı tasarımının daha önce belirtilen hedeflerini sağlamak için Normalizasyon, hangi özneliğin (veya öznelik kümesinin) diğer özellikleri belirlediğini belirlemek için işlevsel bağımlılık kavramını kullanır. Örnekler üzerinde çalışırken bunu aklınızda bulundurun.

Normalleştirme süreci hakkında daha iyi bir fikir edinmek için, çeşitli bina projelerini yöneten bir inşaat şirketinin basitleştirilmiş raporlama faaliyetlerini göz önünde bulundurun. Her projenin kendi proje numarası, adı, atanmış çalışanları vb. Vardır. Her çalışanın mühendis veya bilgisayar teknisyeni gibi bir çalışan numarası, adı ve iş sınıflandırması vardır.

Şirket, her sözleşmeye harcanan saatleri faturalandırarak müşterilerine ücret alır. Saatlik faturalandırma oranı, çalışanın iş sınıflandırmasına bağlıdır. Örneğin, bir saatlik bilgisayar teknisyeni zamanı bir saatlik mühendis zamanından farklı bir oranda faturalandırılır. Periyodik olarak, Tablo 6.1'de görüntülenen bilgileri içeren bir proje raporu oluşturulur. Bu rapor her proje için verileri özet bir formatta düzenler.

Bu durumda, bir danışman bu raporlama senaryosunu desteklemek için bir veritabanı oluşturmakla görevlendirilir. İlk adım, raporu oluşturmak için gerekli temel verilere odaklanmak olacaktır. Toplam ücretler, subtotals ve toplamaların hepsi türetilmiş verilerdir. Bölüm 4'ten türetilmiş verilerin veritabanında saklanabileceğini veya saklanamayacağını hatırlayın. İlk tasarım tamamlandıktan sonra, danışman hangi türetilmiş verilerin depolanacağı ve gerektiğinde hesaplanacağı konusunda tasarım kararlarını verebilir. Bu durumda, temel veriler Şekil 6.1'de gösterilmektedir.





## Şekil 6.1 Bir İnşaat Şirketi Raporu için Temel Veriler

Tablo Adı: RPT\_Format

Veritabanı adı: ch06\_constructco

PROJ_NUM	PROJECT_NAME	EMP_NUMBER	EMP_NAME	JOB_CLASS	CHARGE_HOUR	HOURS_BILLED
15	Evergreen	103,101,105, 106, 102	June E. Arbough, John G. News, Alice K. Johnson *, William Smithfield, David H. Senior	Elec. Engineer, Database Designer, Database Designer, Programmer, Systems Analyst	85.5, 105., 105., 35.75, 98.75	23.8, 19.4, 35.7, 12.6, 23.8
18	Amber Wave	114, 118, 104, 112	Annelise Jones, James J. Frommer, Anne K. Ramoras *, Darlene M. Smithson	Applications Designer, General Support, Systems Analyst, DSS Analyst	48.1, 18.36, 96.75, 45.95	25.6, 45.3, 32.4, 45.
22	Rolling Tide	105, 104, 113, 111, 106	Alice K. Johnson, Anne K. Ramoras, Delbert K. Joenbrood *, Geoff B. Wabash, William Smithfield	DB Designer, Systems Analyst, Applications Designer, Clerical Support, Programmer	105., 96.75, 48.1, 26.87, 35.75	65.7, 48.4, 23.6, 22., 12.8
25	Starflight	107, 115, 101, 114, 108, 118, 112	Maria D. Alonzo, Travis B. Bawangi, John G. News *, Annelise Jones, Ralph B. Washington, James J. Frommer, Darlene M. Smithson	Programmer, Systems Analyst, Database Design, Applications Designer, Systems Analyst, General Support, DSS Analyst	35.75, 96.75, 105., 48.1, 96.75, 18.36, 45.95	25.6, 45.8, 56.3, 33.1, 23.6, 30.5, 41.4

Orijinal durumunda kim olmayan veri ham verileri; Normalleştirilmiş veri ilişkilerinde bulunmayan gereksiz veri, çok değerli veriler ve/ veya diğer veri anomalileri içerebilir.

Şekil 6.1'deki temel veriler, raporun düzenlendiği gibi projeler etrafında düzenlenir ve her proje, o proje ile ilişkili verileri temsil etmek için tek bir satıra sahiptir. Temel veriler, bir projenin kendisine atanmış birden fazla çalışanı olduğunu göstermektedir. Şekil 6.1'deki verilerin, birkaç çok değerli veri ögesinin (EMP\_NUM, EMP\_NAME, JOB\_Class, Charge\_Hour, Saatler, Saatler) varlığı ile yansıtılan kimsiz veriler olduğunu unutmayın.

Ne yazık ki, Şekil 6.1'de gösterilen veri yapısı, Bölüm 3'te tartışılan ilişkisel tablo gereksinimlerine uymamaktadır (bkz. Tablo 3.1) ve bu nedenle veri güncellemelerini iyi ele almak için uygun değildir. Aşağıdaki eksiklikleri düşünün:

- Veri yapısı veri tutarsızlıklarını davet eder. Örneğin, Job\_class değeri “seç. Mühendis” “Elect.eng” olarak girilebilir. Bazı durumlarda, “El. Eng.” diğerlerinde ve hala diğerlerinde “ee”. Yapı, Evergreen Projesi'ndeki John G. News ve Alice K. Johnson'ın aynı iş sınıflandırmasına sahip olmalarına rağmen farklı oranlar almalarına izin verecekti.
- Veri yapısı, veri yönetimi görevlerini çok zorlaştıran birkaç çok değerli özellik içerir. Bir proje üzerinde çalışan tüm çalışanlar tek bir hücrede olduğundan, her çalışanı ayrı ayrı tanımlamak ve veritabanının “StarFlight projesinde kaç çalışan çalışıyor?” Gibi soruları cevaplaması zordur.
- Çalışan verileri tabloda gereksizdir, çünkü çalışanlar birden fazla projede çalışabilir. Verilerin eklenmesi, güncellenmesi ve silinmesi, bu yapıyı kullanarak çok hantal olacaktır. Örneğin, Alice K. Johnson için iş sınıflandırmasını değiştirmek en az iki satırın güncellenmesini gerektirecektir.

Açıkçası, bu veri yapısı veri tutarsızlıkları verir. Rapor, hangi veri anomalisinin meydana geldiğine bağlı olarak değişen sonuçlar verebilir. Örneğin, iş sınıflandırması “veritabanı tasarımcısı” tarafından faturalandırılan toplam saatleri göstermek için bir rapor yazdırmak istiyorsanız, bu rapor “DB tasarımcısı” ve “veritabanı tasarımcısı” veri girişleri için veri içermeyecektir - hatta mümkün olduğunu varsayarak Farklı iş sınıflandırmaları arasında ayırım yapmak için iş sınıflandırma sütununun her hücresindeki çoklu değerleri ayırıştırın. Bu tür raporlar anomaliler yöneticiler için çok sayıda soruna neden olur ve uygulama programlama yoluyla düzeltilemez.

Veritabanı tasarımı sırasında bu veri bütünlüğü, veri artıklığı ve veri tutarsızlık sorunları ele alınmalıdır. Bir sonraki bölüm, fazlalıkları en aza indirmek ve veri anomalilerini ortadan kaldırmak için kullanılan normalleştirme sürecinden geçer.

## 6-3 Normalizasyon Süreci

Bu bölümde, gerekli bilgileri oluşturmak için kullanılacak bir dizi normalleştirilmiş ilişkiyi (tablo) üretmek için normalleşmenin nasıl kullanılacağını öğrenirsiniz. Normalleştirmenin amacı, her tablonun iyi biçimlendirilmiş ilişkiler kavramına uygun olmasını sağlamaktır-başka bir deyişle, aşağıdaki özelliklere sahip tablolar:

- Her ilişki (Tablo) tek bir konuyu temsil eder. Örneğin, bir kurs tablosu yalnızca doğrudan kurslarla ilgili verileri içerecektir. Benzer şekilde, bir öğrenci tablosu yalnızca öğrenci verileri içerecektir.
- Her satır/sütun kavşağı, bir grup değer değil, yalnızca bir (tek) değer içerir.
- Hiçbir veri ögesi gereksiz yere birden fazla tabloda saklanmayacaktır (tabloların minimum kontrollü fazlalığı vardır). Bu gereksinimin nedeni, verilerin yalnızca bir yerde güncellenmesini sağlamaktır.
- Bir ilişkideki (Tablo) tüm prime olmayan nitelikler birincil anahtara bağlıdır - birincil anahtar ve birincil anahtardan başka bir şey. Bu gereksinimin nedeni, verilerin birincil anahtar değeri ile benzersiz bir şekilde tanımlanabilmesini sağlamaktır.
- Her ilişkinin (Tablo) verilerin bütünlüğünü ve tutarlılığını sağlayan ekleme, güncelleme veya silme anomalileri yoktur.

Bu hedeflere ulaşmak için, normalizasyon süreci sizi art arda daha yüksek normal formlara yol açan adımlardan geçirir. En yaygın normal formlar ve bunların temel özellikleri Tablo 6.2'de listelenmiştir. Bu normal formların detayları belirtilen bölümlerde verilmiştir.

**Tablo 6.2 Normal Formlar**

Normal form	Karakteristik	Bölüm
İlk Normal Form (1NF)	Tablo biçimi, tekrarlayan grup yok ve PK tanımlandı	6-3A
İkinci Normal Form (2NF)	1NF ve kısmi bağımlılık yok	6-3b
Üçüncü Normal Form (3NF)	2NF ve geçişli bağımlılık yok	6-3c
Boyce-Codd Normal Form (BCNF)	3NF ve her belirleyici bir aday anahtardır (3NF özel durum)	6-6A
Dördüncü Normal Form (4NF)	BCNF ve bağımsız çok değerli bağımlılıklar yok	6-6b
Beşinci Normal Form (5NF veya PJNF)	4NF ve daha küçük tablolara kayıpsız ayrışmaya sahip olamaz	6-6C

Anahtar kavramı normalleşme tartışmasının merkezinde yer almaktadır. Bölüm 3'ten bir aday anahtarının minimal (indirgenemez) bir süperey olduğunu hatırlayın. Birincil anahtar, tablodaki satırları tanımlamak için kullanılan birincil araç olarak seçilen aday anahtardır. Normalizasyon tipik olarak aday anahtarlar perspektifinden sunulsa da, bu ilk tartışma, her tablonun sadece bir aday anahtarına sahip olduğu sadelik uğruna varsayılmaktadır; Bu nedenle, bu aday anahtarı birincil anahtardır.

Veri modelleyicinin bakış açısından, normalleşmenin amacı, tüm tabloların en azından 3NF'de olmasını sağlamaktır. Üst düzey normal formlar bile mevcuttur. Bununla birlikte, beşinci normal form (5NF) ve alan-anahtar normal form (DKNF) gibi normal formların bir iş ortamında karşılaştırılması muhtemel değildir ve esas olarak teorik bir ilgi vardır. Bu tür daha yüksek normal formlar genellikle birleşimleri artırır, bu da veri yedekliliğinin ortadan kaldırılmasına herhangi bir değer eklemeden performansı yavaşlatır. İstatistiksel araştırma gibi bazı çok özel uygulamalar 4NF'nin ötesinde normalizasyon gerektirebilir, ancak bu uygulamalar çoğu iş operasyonunun kapsamı dışında kalmaktadır.

Tablo 6.3 Fonksiyonel Bağımlılık Kavramları

Kavram	Tanım
Fonksiyonel bağımlılık	B özniteliği, A'nın her değeri B'nin bir ve sadece bir değerini belirlerse, A özelliğine tamamen bağlıdır.  Örnek: proj_num $\rightarrow$ proj_name (proj_num olarak okundu Proj_name Proj_name'i işlevsel olarak belirler) Bu durumda, Proj_num özniteliği belirleyici öznitelik olarak bilinir ve Proj_Name özniteliği bağımlı öznitelik olarak bilinir.
Fonksiyonel Bağımlılık (Genelleştirilmiş Tanım)	Astribution A öznitelikini belirler (yani b, bir öznitelik için de aynı fikirde olan tablodaki satırların tüm (genelleştirilmiş tanım), B özniteliği için de bir değerde hemfikirse, işlevsel olarak A'ya bağlıdır)
Tamamen işlevsel bağımlılık (kompozit anahtar)	Eğer B özniteliği fonksiyonel olarak bir kompozit tuşuna bağlıysa, ancak bu kompozit anahtarın herhangi bir alt kümesinde değilse, B özniteliği tamamen işlevsel olarak A'ya bağlıdır.

Normalizasyon sürecini özetlemeden önce, Bölüm 3. bölümde ayrıntılı olarak ele alınan belirleme ve fonksiyonel bağımlılık kavramlarını gözden geçirmek iyi bir fikirdir. Tablo 6.3 ana kavramları özetlemektedir.

Bu kavramları anlamak çok önemlidir, çünkü belirli bir ilişki için fonksiyonel bağımlılıklar kümesini türetmek için kullanılırlar. Normalizasyon süreci her seferinde bir ilişki çalışır, bu ilişkiye olan bağımlılıkları belirler ve ilişkiyi normalleştirir. Aşağıdaki bölümlerde göreceğiniz gibi, normalizasyon, belirli bir ilişkinin bağımlılıklarını belirleyerek ve ilişkiyi (Tablo) aşamalı olarak tanımlanan bağımlılıklara dayanan bir dizi yeni ilişkiye (tablo) ayırarak başlar.

Normalleştirmeye özel ilgi duyan iki tür fonksiyonel bağımlılık, partial bağımlılıklar ve geçişli bağımlılıklardır. Kısmi bir bağımlılık, belirleyicinin birincil anahtarın sadece bir parçası olduğu işlevsel bir bağımlılık olduğunda vardır (bu tartışma için sadece bir aday anahtar olduğu varsayımını hatırlayın). Örneğin,  $(a, b) \rightarrow (c, d)$ ,  $b \rightarrow c$  ve  $(a, b)$  birincil anahtarsa, fonksiyonel bağımlılık  $B \rightarrow C$  kısmi bir bağımlılıktır, çünkü birincil anahtarın sadece bir kısmı ( B ) C'nin değerini belirlemek için gereklidir. Kısmi bağımlılıklar basit ve tanımlanması kolaydır.

$X \rightarrow y$ ,  $y \rightarrow z$  ve  $x$  birincil anahtar olduğu gibi fonksiyonel bağımlılıklar olduğunda geçişli bir bağımlılık vardır. Bu durumda,  $X \rightarrow Z$  bağımlılığı geçişli bir bağımlılıktır, çünkü  $x$ ,  $y$  üzerinden  $z$  değerini belirler. Kısmi bağımlılıklardan farklı olarak, geçişli bağımlılıkların bir dizi veri arasında tanımlanması daha zordur. Neyse ki, geçişli bağımlılıkları tanımlamanın etkili bir yolu vardır: sadece asal olmayan özellikler arasında fonksiyonel bir bağımlılık olduğunda ortaya çıkarlar. Önceki örnekte, gerçek geçişli bağımlılık  $x \rightarrow Z$ 'dir. Bununla birlikte,  $Y \rightarrow Z$  bağımlılığı, geçişli bir bağımlılığın var olduğuna işaret eder. Bu nedenle, normalizasyon sürecinin tartışılması boyunca, asal olmayan özellikler arasında fonksiyonel bir bağımlılığın varlığı, geçişli bir bağımlılığın bir işareti olarak kabul edilecektir. Geçişli bağımlılıklarla ilgili sorunları ele almak için, tablo yapısındaki değişiklikler, geçişli bağımlılığın varlığını işaret eden işlevsel bağımlılığa dayanarak yapılır. Bu nedenle, normalleşmenin açıklamasını basitleştirmek için, bu noktadan itibaren sinyal bağımlılığı geçiş bağımlılığı olarak adlandırılacaktır.

### 6-3A İlk Normal Forma Dönüşümü (1NF)

İlişkisel model verileri bir tablonun bir parçası olarak veya tüm anahtar değerlerin tanımlanması gereken bir tablo koleksiyonu olarak görüntülediğinden, Şekil 6.1'de gösterilen veriler gösterildiği gibi saklanmayabilir. Şekil 6.1'in tekrarlayan gruplar olarak bilinen şeyi içerdiğine dikkat edin. Tekrarlayan bir grup, adını, herhangi bir tek tuş özniteliği oluşumu için aynı veya birden fazla türden bir grubun birden fazla girişinin mevcut olması gerçeğinden türetir. Şekil 6.1'de, her bir proje numarasının (proj\_num) oluşumunun, çalışan numarası, çalışan adı, iş sınıflandırması ve saat başına sütunlardaki bir grup ilgili veriye başvurabileceğini unutmayın. Örneğin, Evergreen Projesi (Proj\_num 5 15) bu noktada bu özelliklerin her biri için beş değer içerir.

Kısmi bağımlılık Bir özniteliğin birincil anahtarın yalnızca bir kısmına (alt kümeye) bağlı olduğu bir koşul.

Geçişli Bağımlılık Bir özniteliğin birincil anahtarın bir parçası olmayan başka bir özelliğe bağlı olduğu bir koşul.

Bir ilişkide tekrarlayan grup, tek bir anahtar özniteliği oluşumu için aynı veya birden çok türden çoklu giriş grubunu tanımlayan bir karakteristik. Örneğin, bir araba üst, iç, alt, trim vb. için birden fazla renge sahip olabilir.



Tablo yapısının normalleştirilmesi veri fazlalıklarını azaltacaktır. Tekrarlayan gruplar varsa, her satırın tek bir varlık örneği tanımladığından ve her satır kolonu kavşağının yalnızca tek bir değeri olduğundan emin olarak ortadan kaldırılmalıdır. Ek olarak, normal formu teşhis etmek için bağımlılıklar tanımlanmalıdır. Normal formun tanımlanması, normalleştirme sürecinde nerede olduğunuzu bilmenizi sağlar. Normalizasyon basit bir üç adım prosedürle başlar.

## Not

Buradaki amaç, normalizasyon sürecini ve çeşitli normal formları göstermektir. Bölüm 4, yeni bir varlık oluşturarak çok değerli özelliklerle uğraşmak için daha sağlam bir çözüm sundu. Bu daha sağlam çözüm aslında normalleştirme sürecinden özellikle çok değerli niteliklerle ilgili birçok adım birleştirir. Bununla birlikte, bu çözüm, tekrarlayan tüm grup sorunları için mutlaka uygulanamaz, bu nedenle ilk normal formun normalleştirme hedeflerini anlamak önemlidir.

Adım 1: Tekrarlayan grupları ortadan kaldırın, verileri her hücrenin tek bir değeri olduğu ve tekrarlayan grupların bulunmadığı tablo bir formatta sunarak başlar. Tekrarlayan grupları ortadan kaldırmak için tabloyu bir proje odağından bir ödev odağına değiştirin. Bu, her bir projeye atanan her çalışan için çok değerli özellikleri tek değerli özelliklere dönüştürür. Bu değişiklik, Şekil 6.1'deki tabloyu Şekil 6.2'de gösterildiği gibi dönüştürür.

## Şekil 6.2 İlk normal formda bir tablo

Tablo adı: data\_org\_1nf

Veritabanı adı: ch06\_constructco

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
15	Evergreen	101	John G. News	Database Designer	105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	96.75	45.8
25	Starflight	101	John G. News *	Database Designer	105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	96.75	23.6
25	Starflight	118	James J. Frommer	General Support	18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Adım 2: Birincil anahtarı tanımlayın Şekil 6.2'deki düzen, sadece bir cosnetik değişimden daha fazlasını temsil eder. Sıradan bir gözlemci bile, proje numarası her satırı benzersiz bir şekilde tanımlamadığı için Proj\_Num'un yeterli bir ana anahtar olmadığını not edecektir. Örneğin, Proj\_num değeri 15, Evergreen projesinde çalışan çalışanları içeren beş satırdan herhangi birini tanımlayabilir. Herhangi bir öznitelik değerini benzersiz bir şekilde tanımlayacak uygun bir birincil anahtarı korumak için, yeni anahtar proj\_num ve emp\_num kombinasyonundan oluşmalıdır. Örneğin, Şekil 6.2'de gösterilen verileri kullanarak, Proj\_Num 5 15 ve EMP\_NUM 5 103 olduğunu biliyorsanız, Proj\_Name, EMP\_NAME, JOB\_Class, CHG\_HOUR ve SAATLERİTLERİN Girişleri Evergreen, Haziran E. Arbough, Secect. Mühendis, sırasıyla 84.50 \$ ve 23.8.

Adım 3: Tüm bağımlılıkları tanımlayın PK'nin tanımlanması, 2. adımda aşağıdaki bağımlılığı zaten tanımladığınız anlamına gelir:

Proj\_num, emp\_num → proj\_name, emp\_name, job\_class, chg\_hour, saatler

Yani, proj\_name, emp\_name, job\_class, chg\_hour ve saat değerleri, proj\_num ve emp\_num kombinasyonuna bağlıdır - belirlenirler.

1NF'ye ulaşmak, orijinal yapıda var olan tüm anomalileri ele almak için yeterli değildir. 1NF, tekrarlayan gruplarla uğraşmış ve tablomuzun Bölüm 3'te açıklandığı gibi ilişkisel bir tablo için gereksinimlere uygun olmasını sağlamıştır. Ancak anomaliler devam etmektedir. Örneğin, bir projeye başka bir çalışan her atandığında, bazı veri girişleri (Proj\_Name, EMP\_NAME ve CHG\_HOUR gibi) gereksiz yere tekrarlanır. 200 veya 300 tablo girişinin yapılması gerektiğinde veri girişi işini hayal edin! İdeal olarak, çalışan numarasının girişi Darlene M. Smithson'u, iş tanımını ve saatlik ücretini tanımlamak için yeterli olmalıdır. Yalnızca bir kişi 112 sayısı ile tanımlandığından, bir ödev yapıldığında veya güncellendiğinde bu kişinin özellikleri (isim, iş sınıflandırması vb.) Girilmesi gerekmemelidir.

Birincil anahtar bağımlılığına ek olarak ek bağımlılıklar olduğu için kalan anomaliler var. Örneğin, proje numarası proje adını belirler. Başka bir deyişle, proje adı proje numarasına bağlıdır. Bu bağımlılığı şu şekilde yazabilirsiniz:

Proj\_num → proj\_name

Ayrıca, bir çalışan numarası biliyorsanız, çalışanın adını, iş sınıflandırmasını ve saat başına ücretin de bilirsiniz. Bu nedenle, daha sonra gösterilen bağımlılığı tanımlayabilirsiniz:

Emp\_num → emp\_name, job\_class, chg\_hour

Daha basit bir şekilde, bir çalışanın aşağıdaki özellikleri vardır: bir sayı, bir isim, bir iş klişesi ve saatte bir ücret.

**Not** Bazı bağımlılıklar diğerlerinden daha açıktır. Örneğin, iş kuralı “her iş sınıflandırmasının saatte belirli bir ücreti vardır”, saat başına ücretin iş sınıflandırmasına bağlı olduğunu ima eder. Bununla birlikte, bu bölümdeki tartışmaların normalleştirme süreci bakış açısına dayandığını ve normalleşmenin iş kurallarını doğrulamaya nasıl yardımcı olabileceğini göstermeye hizmet ettiğini unutmayın.

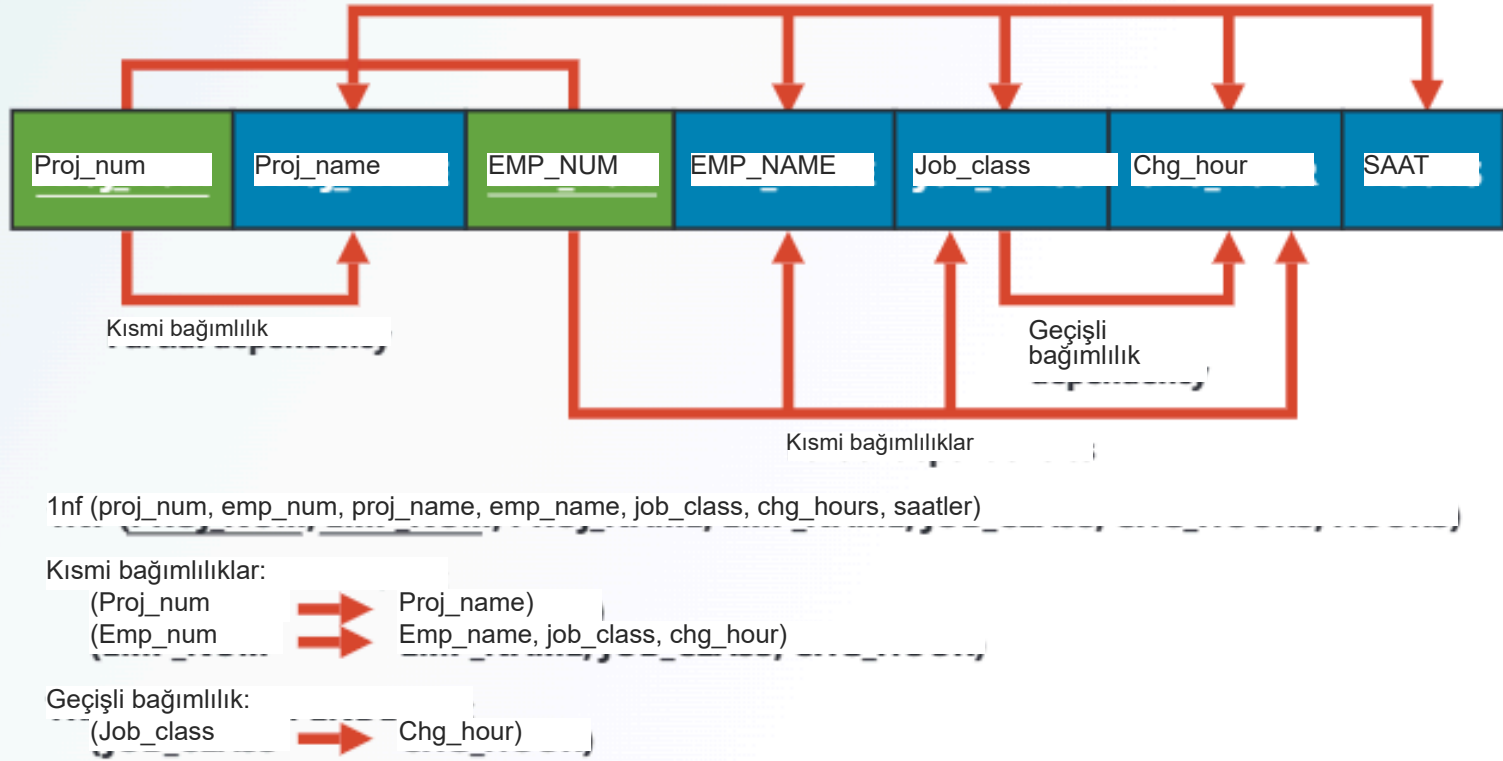
Şekil 6.2'deki verileri daha fazla inceleyerek, iş sınıflandırmasını bilmenin, bu iş sınıflandırması için saat başına ücret bilmek anlamına geldiğini görebilirsiniz. (Tüm sistem analisti veya programcı pozisyonlarının proje veya çalışandan bağımsız olarak saatte aynı ücrete sahip olduğuna dikkat edin.) Başka bir deyişle, saat başına ücret, çalışan değil, iş sınıflandırmasına bağlıdır. Bu nedenle, son bir bağımlılığı tanımlayabilirsiniz:

Job\_class → chg\_hour

Bununla birlikte, bu bağımlılık iki prim olmayan nitelik arasında mevcuttur; Bu nedenle, geçişli bir bağımlılığın var olduğuna dair bir işarettir ve buna geçişli bir bağımlılık olarak adlandıracağız. Yeni incelediğiniz bağımlılıklar, Şekil 6.3'te gösterilen şema yardımıyla da tasvir edilebilir. Böyle bir diyagram, belirli bir tablo yapısında bulunan tüm bağımlılıkları tasvir ettiğinden, bağımlılık diyagramı olarak bilinir. Bağımlılık diyagramları, bir tablonun özellikleri arasındaki tüm ilişkilerin kuşbakışı görünümünü elde etmede çok yararlıdır ve bunların kullanımları, önemli bir bağımlılığı göz ardı etme olasılığınızı daha az hale getirir.

**Bağımlılık Diyagramı** Bir tablo içindeki tüm veri bağımlılıklarının (birincil anahtar, kısmi veya geçişli) temsili.

## Şekil 6.3 İlk Normal Form (1NF) Bağımlılık Diyagramı



Şekil 6.3'ü incelerken, bir bağımlılık diyagramının aşağıdaki özelliklerini not edin:

1. Birincil anahtar özellikleri kalın, altı çizili ve farklı bir renktedir.
2. Özniteliklerin üzerindeki oklar, arzu edilen tüm bağımlılıkları- yani, birincil anahtara dayanan bağılırlar. Bu durumda, varlığın özelliklerinin proj\_num ve emp\_num kombinasyonuna bağlı olduğunu unutmayın.
3. Bağımlılık diyagramının altındaki oklar daha az arzu edilen bağımlılıkları gösterir. Bu tür bağımlılıkların iki türü vardır:
  - A. Kısmi bağımlılıklar. Proj\_name'i belirlemek için yalnızca proj\_num'u bilmeniz gerekir; Yani, proj\_name birincil anahtarın sadece bir kısmına bağlıdır. Ayrıca, emp\_name, job\_class ve chg\_hour'u bulmak için yalnızca emp\_num'u bilmeniz gerekir. Kompozit birincil anahtarın sadece bir kısmına dayanan bir bağımlılık kısmi bir bağımlılıktır.
  - B. Geçişli bağımlılıklar. CHG\_HOUR'un Job\_Class'a bağlı olduğunu unutmayın. Ne chg\_hour ne de job\_class ana bir özellik olmadığından, yani hiçbir öznitelik en azından bir anahtarın bir parçası değildir - durum geçişli bir bağımlılığın göstergesidir. Başka bir deyişle, sadece prim olmayan nitelikler arasında fonksiyonel bir bağımlılık olduğunda geçişli bir bağımlılık mevcuttur. Geçiş bağımlılıkları veri anomalileri verir.

## Not

İlk normal form (1NF) terimi, aşağıdaki ilişkisel bir tablonun tanımına uygun olan tablo formatını açıklar: • Anahtar özelliklerin tümü tanımlanmıştır.

• Masada tekrar eden grup yoktur. Başka bir deyişle, her satır/sütun kesimi bir değer kümesi değil, bir ve sadece bir değer içerir.

• Tüm özellikler birincil anahtara bağlıdır.

İlk Normal Form (1NF) Normalleştirme sürecinin ilk aşaması. Tekrarlayan gruplar ve birincil anahtar tanımlanmayan tablo formatında tasvir edilen bir ilişkiyi tanımlar. İlişkideki tüm anahtar olmayan özellikler birincil anahtara bağlıdır.

Şekil 6.3, 1NF'deki tablo için ilişkisel şemayı ve tanımlanan her bağımlılık için metin gösterimi içerir.

Tüm ilişkisel tablolar 1NF gereksinimlerini karşılamaktadır. Şekil 6.2'deki 1NF verileri, Şekil 6.1'deki kimsiz veriler üzerinde bir iyileşme olmasına rağmen, hala istenmeyen problemleri vardır. Örneğin, Şekil 6.2'de gösterilen ve Şekil 6.3'teki bağımlılık diyagramı ile temsil edilen 1NF tablo yapısı, daha önce keşfettiğimiz aynı veri anomalilerine neden olan kısmi bağımlılıklar ve geçiş bağımlılıkları içerir.

Her ne kadar kısmi bağımlılıklar bazen performans nedenleriyle kullanılsa da, kısmi bağımlılıklar içeren bir tablo hala veri fazlalıklarına ve dolayısıyla çeşitli anomalilere tabi olduğu için dikkatle kullanılmalıdır. Örneğimizin hala aşağıdaki anomalileri var:

- Anomalileri güncelleyin. Çalışan Annelise Jones için Job Class'ı değiştirmek birçok girişin güncellenmesini gerektirir; Aksi takdirde, veri tutarsızlıkları oluşturur.
- Ekleme anomalileri. Yeni bir çalışan eklemek, çalışanın bir projeye atanmasını ve dolayısıyla yinelenen proje bilgilerine girmesini gerektirir. Çalışan henüz bir projeye atanmamışsa, çalışan veri girişini tamamlamak için bir fantom projesi oluşturulmalıdır.
- Silme anomalileri. Diyelim ki yalnızca bir çalışanın belirli bir proje ile ilişkili olduğunu varsayalım. Bu çalışan silinirse, proje bilgileri de silinecektir.

Veri fazlalıkları, her satır girişinin verilerin çoğaltılmasını gerektirdiğinden oluşur. Bu tür çaba çoğaltılması çok verimsizdir ve veri anomalilerinin oluşturulmasına yardımcı olur; Hiçbir şey, kullanıcının çalışan adının, pozisyonunun veya saatlik ücretin biraz farklı sürümlerini yazmasını engellemez. Örneğin, EMP\_NUM 5 102 için çalışan adı Dave Senior veya D. olarak girilebilir. Proje adı da Evergreen olarak doğru bir şekilde girilebilir veya Evergeen olarak yanlış yazılabilir. Bu tür veri anomalileri ilişkisel veritabanının bütünlüğünü ve tutarlılık kurallarını ihlal eder.

## 6-3B İkinci Normal Forma Dönüşüm (2NF)

2NF'ye dönüşüm yalnızca 1NF'nin kompozit birincil tuşu olduğunda ortaya çıkar. 1NF'nin tek bir anahtar birincil tuşu varsa, tablo otomatik olarak 2NF cinsindendir. 1NF-2NF dönüşümü basittir. Şekil 6.3'te görüntülenen 1NF biçiminden başlayarak aşağıdaki adımları atarsınız:

**Adım 1:** Kısmi bir bağımlılıkta belirleyici görevi gören birincil anahtarın her bir bileşeni için kısmi bağımlılıkları ortadan kaldırmak için yeni tablolar yapın, bu bileşenin bir kopyasını birincil anahtar olarak içeren yeni bir tablo oluşturun. Bu bileşenler yeni tablolara yerleştirilirken, orijinal tabloda da kalmaları önemlidir. Belirleyiciler orijinal tabloda kalmalıdır, çünkü bu yeni tabloları orijinal tabloda ilişkilendirmek için gereken ilişkilerin yabancı anahtarları olacaktır. Gözden geçirilmiş bağımlılık diyagramını oluşturmak için, her anahtar bileşenini ayrı bir satıra yazın ve ardından son satıra orijinal (kompozit) tuşunu yazın. Örneğin:

```
Proj_num emp_num
proj_num emp_num
```

Her bileşen yeni bir tabloda anahtar olacaktır. Başka bir deyişle, orijinal tablo artık üç tabloya (proje, çalışan ve ödev) ayrılmıştır.

**Adım 2:** Karşılık gelen bağımlı nitelikleri yeniden atayın Kısmi bağımlılıklara bağlı nitelikleri belirlemek için Şekil 6.3'ü kullanın. Orijinal anahtar bileşenlerinin bağımlılıkları, gösterilen bağımlılık diyagramının altındaki oklar incelenerek bulunur



Şekil 6.3'te. Kısmi bir bağımlılıkta bağımlı olan özellikler orijinal tablodan çıkarılır ve bağımlılığın belirleyicisi ile yeni tabloya yerleştirilir. Kısmi bir bağımlılıkta bağımlı olmayan herhangi bir özellik orijinal tabloda kalacaktır. Başka bir deyişle, 2NF'ye dönüşümden kaynaklanan üç tabloya uygun isimler (proje, çalışan ve ödev) verilir ve aşağıdaki ilişkisel şemalar tarafından tanımlanır:

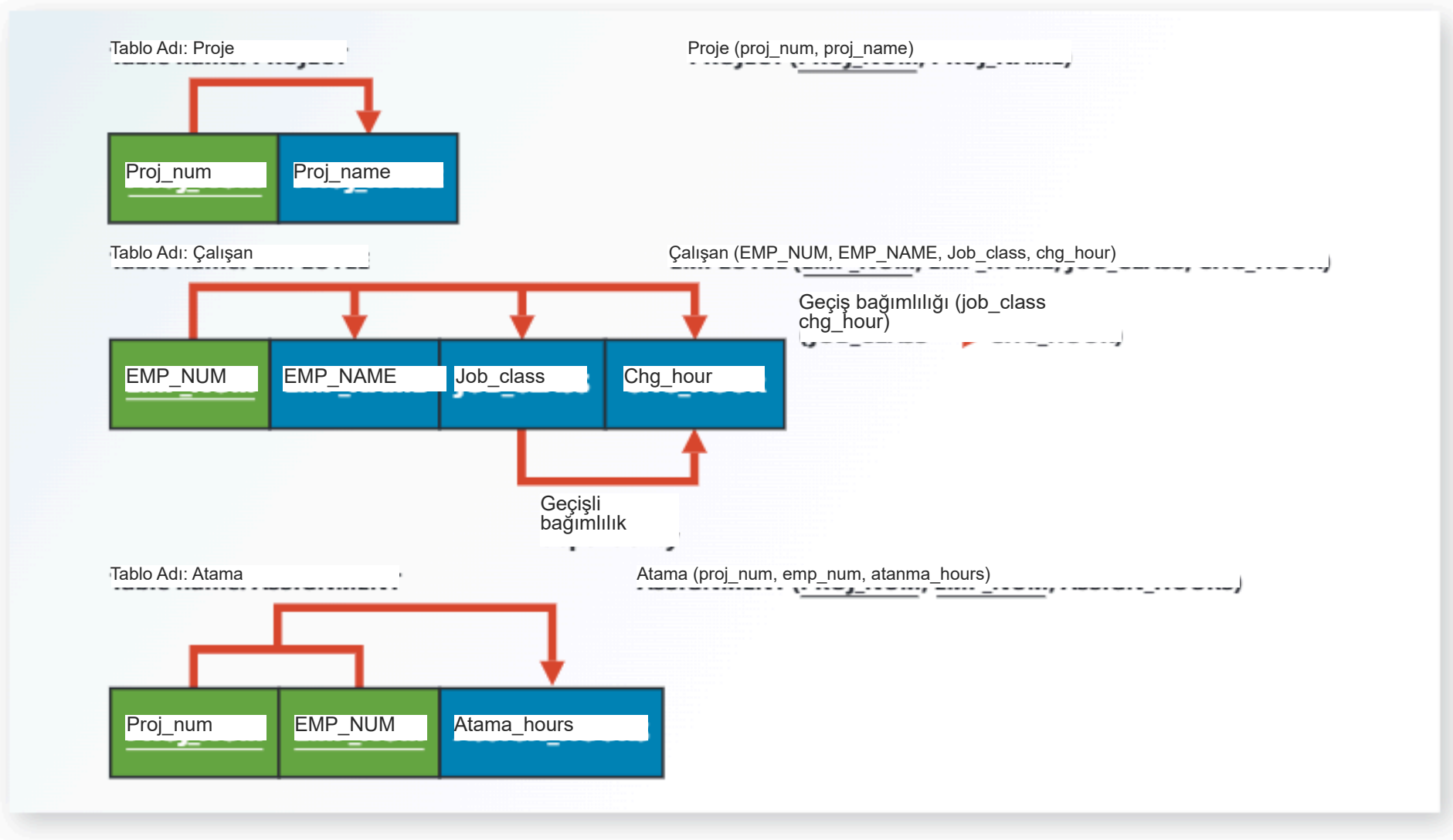
Project (proj\_num, proj\_name) çalışanı (emp\_num, emp\_name, job\_class, chg\_hour) ataması (proj\_num, emp\_num, atanma\_hours)

Her bir çalışan tarafından her bir projeye harcanan saat sayısı, atama tablosundaki hem Proj\_num hem de EMP\_NUM'a bağlı olduğundan, bu saatleri atama tablosunda atanma\_hours olarak bırakırsınız. Atama tablosunun, proj\_num ve emp\_num özniteliklerinden oluşan kompozit birincil anahtar içerdiğine dikkat edin. Ayrıca, belirleyicileri orijinal tabloda bırakarak ve onları yeni tabloların birincil anahtarları haline getirerek, birincil anahtar/yabancı anahtar ilişkileri oluşturulduğuna dikkat edin. Örneğin, çalışan tablosunda EMP\_NUM birincil anahtardır. Atama tablosunda EMP\_NUM, kompozit birincil anahtarın (proj\_num, emp\_num) bir parçasıdır ve çalışan tablosunu atama tablosuna ilişkin yabancı bir anahtardır.

1. ve 2. adımların sonuçları Şekil 6.4'te gösterilmektedir. Bu noktada, daha önce tartışılan anomalların çoğu ortadan kaldırılmıştır. Örneğin, şimdi bir proje kaydını eklemek, değiştirmek veya silmek istiyorsanız, yalnızca proje tablosuna gitmeniz ve değişikliği yalnızca bir satırda yapmanız gerekir.

Kısmi bir bağımlılık yalnızca bir tablonun birincil anahtarı birkaç özellikten oluştuğunda mevcut olabileceğinden, birincil anahtarı yalnızca tek bir öznitelikten oluşan bir tablo, 1NF'de olduğunda otomatik olarak 2NF'dedir.

#### Şekil 6.4 İkinci Normal Form (2NF) Dönüşüm Sonuçları



İkinci Normal Form (2NF)  
Normalleştirme sürecinde,  
bir ilişkinin 1NF'de olduğu ve  
kısımlı bağımlılıkların  
olmadığı ikinci aşama (

birincil  
anahtar).

Belirli bir satırdaki değeri,  
o satırdaki diğer değerleri  
doğrudan belirleyen  
herhangi bir özniteliği  
belirleyin.

## Not Bir tablo aşağıdaki durumlarda ikinci normal formda (2NF): • 1NF'de

ve • kısmi bağımlılık içermez; yani, hiçbir öznitelik sadece bir kısmına bağlı değildir

birincil anahtar. 2NF'deki bir masanın geçişli bağımlılık sergilemesi hala mümkündür. Yani, birincil anahtar, prime olmayan özellikler arasında fonksiyonel bir bağımlılıkla gösterildiği gibi, diğer prime olmayan nitelikleri işlevsel olarak belirlemek için bir veya daha fazla prime olmayan niteliklere güvenebilir.

Şekil 6.4 hala anomaliler üretebilen geçişli bir bağımlılığı göstermektedir. Örneğin, birçok çalışan tarafından tutulan bir iş sınıflandırması için saat başına ücret değişirse, bu çalışanların her biri için bu değişiklik yapılmalıdır. Saat başına ücretten etkilenen bazı çalışan kayıtlarını güncellemeyi unutursanız, aynı iş tanımına sahip farklı çalışanlar farklı saatlik ücretler üretecektir.

## 6-3C Üçüncü Normal Forma Dönüşümü (3NF)

Şekil 6.4'te gösterilen veritabanı kuruluşu tarafından oluşturulan veri anomalileri, aşağıdaki iki adımı tamamlayarak kolayca ortadan kaldırılabilir:

Adım 1: Her geçişli bağımlılık için geçişli bağımlılıkları ortadan kaldırmak için yeni tablolar yapın, belirleyicisinin bir kopyasını yeni bir tablo için birincil anahtar olarak yazın. Determinant, değeri arka arkaya diğer değerleri belirleyen herhangi bir özniteliktir. Üç farklı geçiş bağımlılığınız varsa, üç farklı belirleyiciniz olacaktır. 2NF'ye dönüşümde olduğu gibi, belirleyicinin yabancı bir anahtar olarak hizmet etmek için orijinal tabloda kalması önemlidir. Şekil 6.4, geçişli bir bağımlılık içeren sadece bir tabloyu göstermektedir. Bu nedenle, bu geçiş bağımlılığı için belirleyiciyi şu şekilde yazın:

Job\_class

Adım 2: Şekil 6.4'ü kullanarak karşılık gelen bağımlı nitelikleri yeniden atayın, Adım 1'de tanımlanan her bir belirleyiciye bağlı olan özleri tanımlayın. Bağımlı özneleri yeni tablolara belirleyicileriyle yerleştirin ve orijinal tablolarından çıkarın. Bu örnekte, çalışan tablosu bağımlılık tanımını şu şekilde bırakmak için Şekil 6.4'te gösterilen çalışan tablosundan CHG\_HOUR'u ortadan kaldırın:

Emp\_num → emp\_name, job\_class

1. ve 2. adımlarda tanımladığınız tüm tabloları göstermek için yeni bir bağımlılık diyagramı çizin. Tabloyu içeriğini ve işlevini yansıtacak şekilde adlandırın. Bu durumda, iş uygun görünüyor. Her tablonun bir belirleyiciye sahip olduğundan ve hiçbir tablonun uygunsuz bağımlılıklar içermediğinden emin olmak için tüm tabloları kontrol edin. Bu adımları tamamladığınızda, sonuçları Şekil 6.5'te göreceksiniz.

Başka bir deyişle, 3NF dönüşümü tamamlandıktan sonra, veritabanınız dört tablo içerecektir:

Project (proj\_num, proj\_name) çalışanı (emp\_num, emp\_name, job\_class) Job (job\_class, chg\_hour)

Atama (proj\_num, emp\_num, atanma\_hours)

Bu dönüşümün orijinal çalışan tablosunun geçişli bağımlılığını ortadan kaldırdığını unutmayın. Tabloların şimdi üçüncü normal formda (3NF) olduğu söyleniyor.