

NORMALİZASYON

TEMEL KAVRAMLAR

FONKSİYONEL (İŞLEVSEL) BAĞIMLILIK

- ❖ *Bir niteliğin değeri, diğer niteliği kesin olarak belirliyorsa aralarında fonksiyonel bağımlılık vardır.*
- ❖ Fonksiyonel bağımlılık gerçek hayatta **nitelikler arası ilişkilerden çıkarılır.**

TEMEL KAVRAMLAR

FONKSİYONEL (İŞLEVSEL) BAĞIMLILIK

- X ile Y nitelik kümelerini göstermek üzere fonksiyonel bağımlılık $X \rightarrow Y$ şeklinde verilir.

Örneğin: **sicilNo** \rightarrow *personel_ad, personel_soyad*

tcNo \rightarrow *ad, soyad*

Burada **sicilNo** ve **tcNo** belirleyici öznitelik, diğer alanlar ise bağımlı öznitelikler olarak adlandırılır.

ÖRNEK

id	isim	sehir
123	Ahmet	Ankara
324	Derya	Ankara
574	Derya	İstanbul

Kişiler tablosu ile ilgili neler söyleyebiliriz?

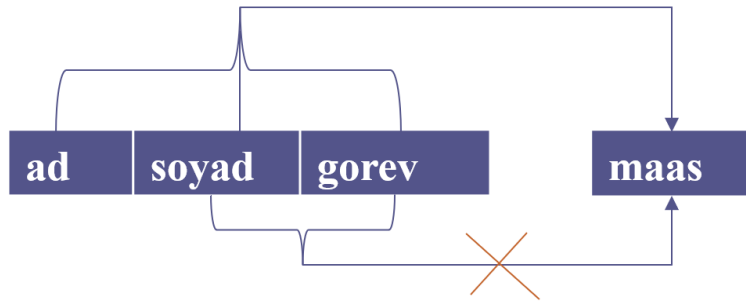
1. “Eğer ID numarasını biliyorsam, ismini de biliyorum”
2. ID numarası ismi belirlemektedir.
3. İSİM niteliği, ID’ye fonksiyonel bağımlıdır. ($ID \rightarrow İSİM$)

TEMEL KAVRAMLAR

Tam Fonksiyonel (İşlevsel) Bağımlılık:

A, B bir R tablosunun öznitelikleri kümesi olsun.

B, A'ya işlevsel bağımlı ve A'nın herhangi bir C öz alt kümesi için B, C'ye işlevsel bağımlı olmuyorsa **B**, A'ya tam işlevsel bağımlıdır diyoruz.



Tam fonksiyonel bağımlılık

* **Bileşik anahtar (Composite Key) alanlar**, tam fonksiyonel bağımlılık özelliği gösterir.

TEMEL KAVRAMLAR

Kısmi Bağımlılık

Kısmi bağımlılık; belirleyici alanlardan bir tanesi, fonksiyonel bağımlılığı sağlamada yeterli olması durumunda gözlenir.

- Belirleyici bir aday anahtar kullanılır.
- Basittir.
- Tanımlaması kolaydır.

TEMEL KAVRAMLAR

Kısmi Bağımlılık

Örneğin, *aday anahtarlar* (A, B) ve *alanlar* (C, D) olsun.

Burada birincil anahtar olarak **B** seçilsin.

B \rightarrow **C** işlevsel bağımlılığı kısmi olacaktır.

Çünkü **C** alanını belirlemek için anahtar alanların yalnızca bir kısmı (**B**) yeterlidir.

TEMEL KAVRAMLAR

Dolaylı Bağımlılık (Transitive Dependency)

A,B,C niteliklilerini içeren 3 nitelikli bir tabloda

A, B, ve C için:

- $A \rightarrow B$ ve $B \rightarrow C$ işlevsel bağımlılıkları bulunmakta ise,
- **C, A 'ya B aracılığı ile dolaylı bağımlıdır.**

ÖRNEK

Çalışan

<u>Emp-No</u>	Emp-Name	Dept	Manager
005	Smith	Marketing	Jones
007	Bond	Accounts	Bloggs
009	King	Info Systems	Hume
010	Holt	Accounts	Bloggs

Emp2

Not: **Dept.** ve **Manager** alanları **Çalışan** (Employee) varlığının direk olarak özneliği değildir. Geçişli bağımlılıklar, veri anomalileri üretirler.

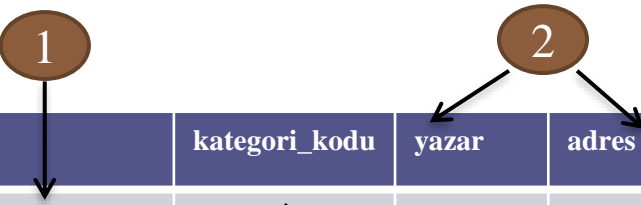
Manager alanı anahtar olmayan *Dept.* alanına bağlıdır.
Manager, *Emp_No*'ya *Dept.* anahtar alanı ile dolaylı bağımlıdır.

NORMALİZASYON

- ❑ Normalizasyon, verilen bir nitelik kümesini sadece doğrudan ilişkili verileri içererek daha küçük tablolara (ilişkilere) aşama aşama bölmektir.
- ❑ İlişikler veri modeli tasarlanırken
 - *veri tekrarını* ortadan kaldırmak,
 - *veri üzerindeki işlemleri kolay ve esnek hale getirmek*amaçlanır.

NORMAL FORMA UYMAYAN (UNF) ÖRNEK 1

- ❑ Bir kayıttaki her nitelik de sadece kendisine izin verilen özelliklerdeki **tek (atomic) veriyi saklamalıdır**.
- ❑ Bir tabloda bir varlığı ihtiva eden kayıtların olması gerekir.



urun_no	urun_adi	kategori	kategori_kodu	yazar	adres
1	Bilgisayara Giriş	Elektronik, bilişim, Bilim	ELE, BİL	Ahmet SERT	Kadıköy, İstanbul
2	Fizik	Bilim, Sayısal	BİM, SAY	Kemal SAY	GAZİ Sokak, Ankara
3	Deneme	Sosyal	SOS	Ayşe DEMİR	Avcılar, İstanbul

1. Bir sütunda birden fazla veri saklanması (atomik olmayan)
2. Birden fazla varlığı ihtiva eden kayıtların olması

NORMAL FORMA UYMAYAN (UNF) ÖRNEK 2

❑ Bir tabloda tekrarlı kayıtlar ve bunun sonunca boş değerlerin olmaması gerekir.

urun_no	urun_adi	kategori	kategori_kodu	yazar	adres
1	Bilgisayara Giriş	Elektronik	ELE	Ahmet	İstanbul
		Bilişim	BİL	Ahmet	İstanbul
2	Fizik	Bilim	BİM	Kemal	Ankara
		Sayısal	SAY	Kemal	Ankara
3	Deneme	Sosyal	SOS	Ayşe	İstanbul

Boş (Null) Alanlar

3

4

Tekrar kayıtlar eden

Tekrar kayıtlar eden

NORMAL FORMA UYMAYAN (UNF) ÖRNEK 3

❑ Bir tabloda tekrarlı sütunlar olmaması gerekir.

Ogr_no	Bolum_kodu	Bolum	Ders_kodu	Sınav
009001	BTP	Bilgisayar	B1,B2,B3	75,85,45
009002	BTP	Bilgisayar	B2,B3,B4	25,60,44
009003	ELK	Elektrik	E1,E4,E9	45,66,88
009004	ELK	Elektrik	E5,E3,E8	66,78,45

↓ 5 Tekrarlı sütunlar ↓

Ogr_no	Bolum_kodu	Bolum	Ders_kodu1	Ders_kodu2	Ders_kodu3	Sınav1	Sınav2	Sınav3
009001	BTP	Bilgisayar	B1	B2	B3	75	85	45
009002	BTP	Bilgisayar	B2	B3	B4	25	60	44
009003	ELK	Elektrik	E1	E4	E9	45	66	88
009004	ELK	Elektrik	E5	E3	E8	66	78	45

NORMALİZASYON İŞLEMLERİNİN SAĞLADIĞI AVANTAJLAR

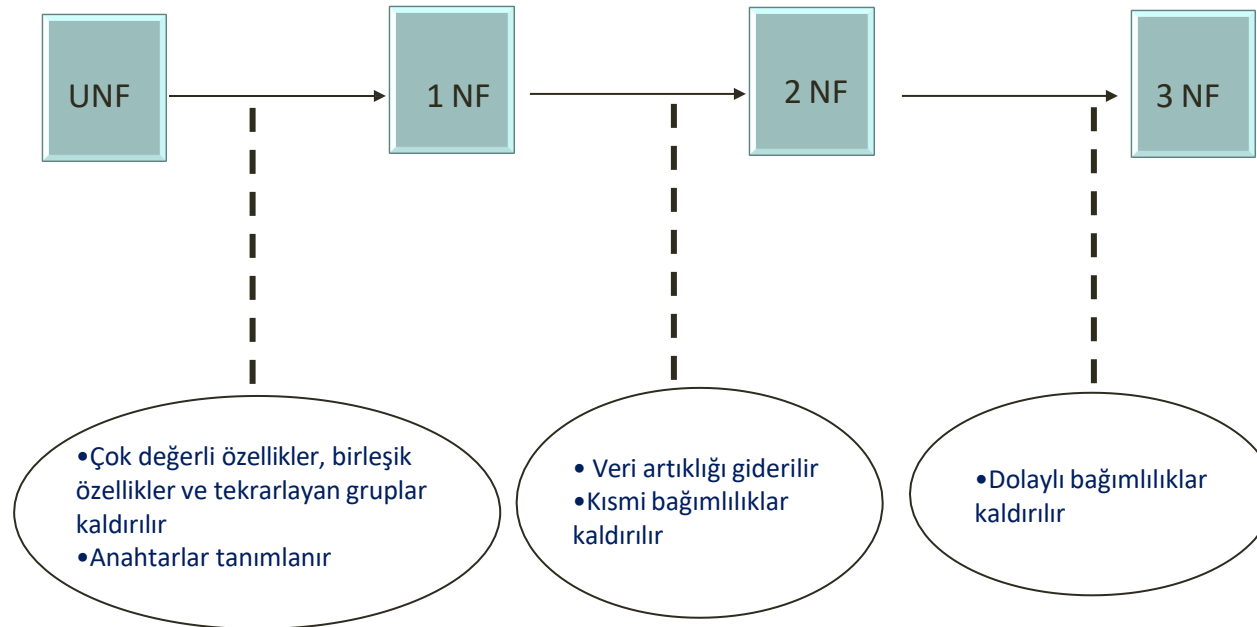
Veri
bütünlüğünü
sağlamak,

Veri tekrarını
engellemek

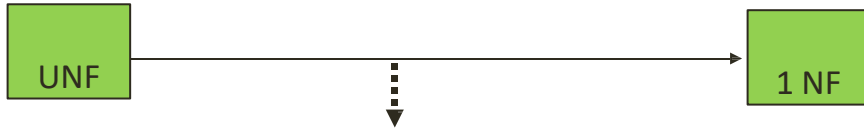
Uygulamadan
bağımsızlık
kazandırmak

Performansı
arttırmak

Normalizasyon aşamaları

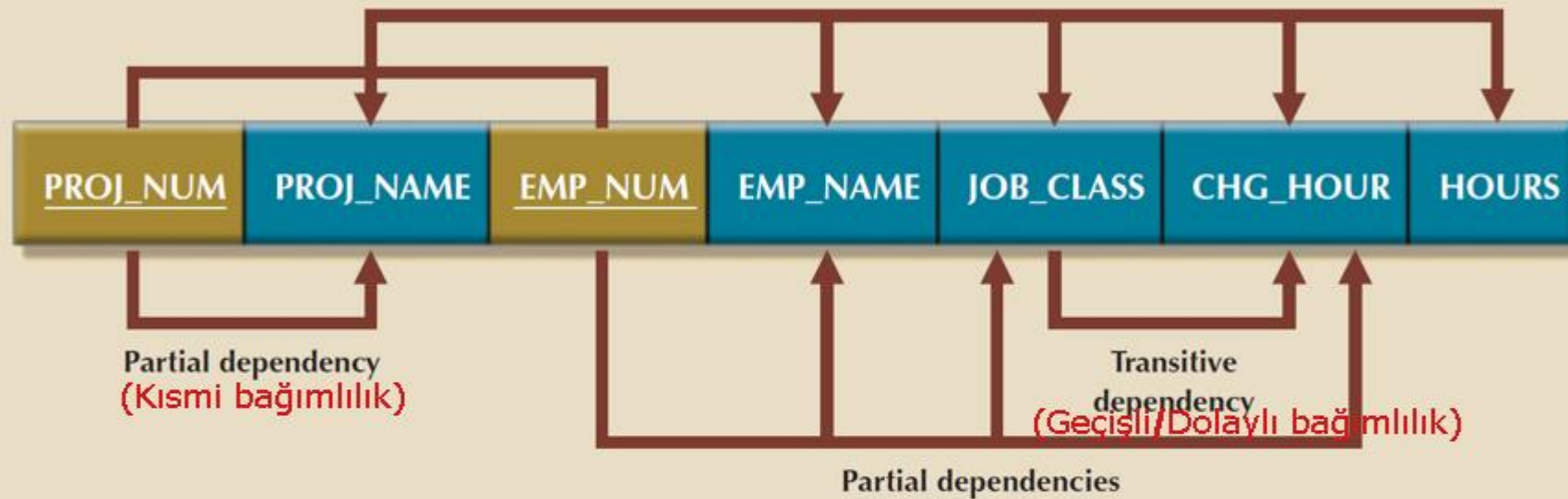


Birinci (First) Normal Form (1NF)



- ❑ **Adım 1:** Yinelenen Grupları Ortadan Kaldırma
Boşları kayıtları yok etme
- ❑ **Adım 2:** Birincil Anahtarın belirlenmesi
Özellik değerini benzersiz olarak tanımlamalıdır.
Yeni anahtar oluşturulmalıdır
- ❑ **Adım 3:** Tüm Bağımlılıkların Belirlenmesi
Bir diyagram ile gösterilen bağımlılıklar

1. NF FORMDA BAĞIMLILIK DİYAGRAMI



1NF (PROJ_NUM, EMP_NUM, PROJ_NAME, EMP_NAME, JOB_CLASS, CHG_HOURS, HOURS)

PARTIAL DEPENDENCIES:

(PROJ_NUM \Rightarrow PROJ_NAME)

(EMP_NUM \Rightarrow EMP_NAME, JOB_CLASS, CHG_HOUR)

TRANSITIVE DEPENDENCY:

(JOB_CLASS \Rightarrow CHG_HOUR)

1. NF FORMA UYAN BİR ÖRNEK

Table name: DATA_ORG_1NF

Database name: 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

SORUNLAR

❖ Satır Ekleme

- Yeni bir çalışan eklemek için çalışanın bir projeye atanması ve bu nedenle yinelenen proje bilgileri girmesi gerekir.
- Çalışan henüz bir projeye atanmamışsa, çalışan veri girişini tamamlamak için bir hayalet proje oluşturulmalıdır.

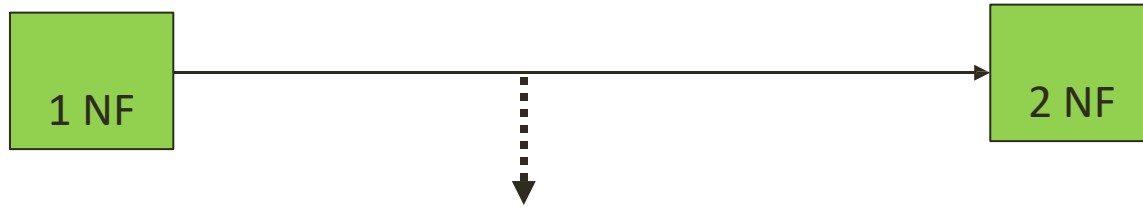
❖ Satır Silme

- Belirli bir projeye yalnızca bir çalışanın ilişkili olduğunu varsayılırsa; çalışan silinirse, proje bilgileri de silinir.

❖ Satır Güncelleme

- Annelise Jones çalışanı için **JOB_CLASS**'ın değiştirilmesi birçok girişi güncellemesi yapılması gerekir. Aksi takdirde veri tutarsızlıkları oluşturur.

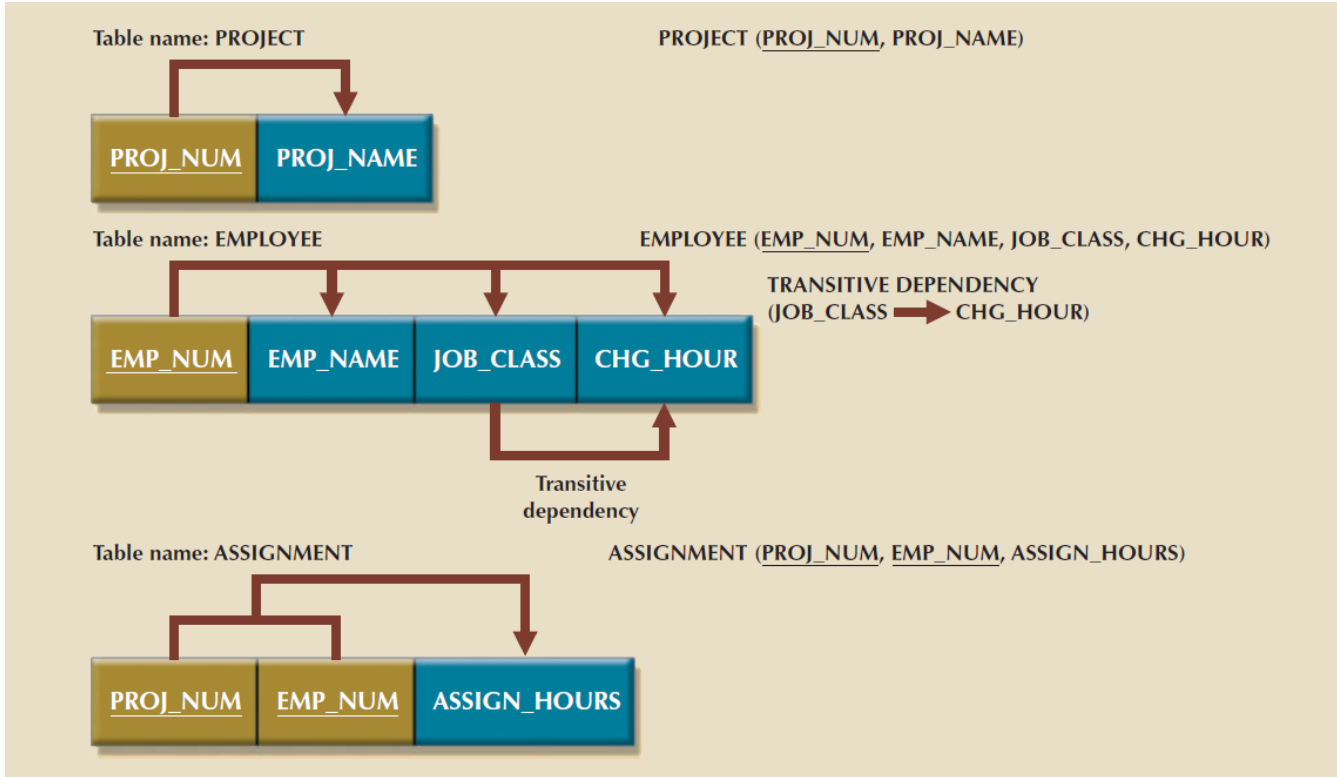
2. NORMAL FORM



Kurallar:

- Tablo 1. NF da olmalı
- **Fonksiyonel bağımlılık** göz önünde bulundurularak **tablonun başka tablolara ayrıştırılmasıdır.**
- Kayıtlar bir tablonun birincil anahtarı dışında bir öğeye bağımlı olmamalıdır.

2. NF FORMA DÖNÜŞTÜRME DİYAGRAMI



SROUNLAR

Anomaliler oluşturabilecek geçişli bir bağımlılık vardır.

Örnek:

- ❖ Birçok çalışan tarafından tutulan bir iş sınıflandırması için saat başına ücret (CHG_HOUR) değişebilir.
- ❖ Bu değişiklik çalışanların her biri için yapılmalıdır.
- ❖ Saat başına ücret değişikliğinden etkilenen bazı çalışan kayıtlarını güncellemeyi unutulursa; aynı iş tanımına (JOB_CLASS) sahip farklı çalışanlar farklı saatlik ücretler oluşturur.

Üçüncü Normal Form (3NF)

- ❑ Tablo 2 NF formunda olmalıdır.
- ❑ Geçişli bağımlılıklar ortadan kaldırılır.
- ❑ Geçişli bağımlılıkları ortadan kaldırmak için yeni tablolar oluşturulmalıdır.

3. NF FORMA DÖNÜŞTÜRME DİYAGRAMI

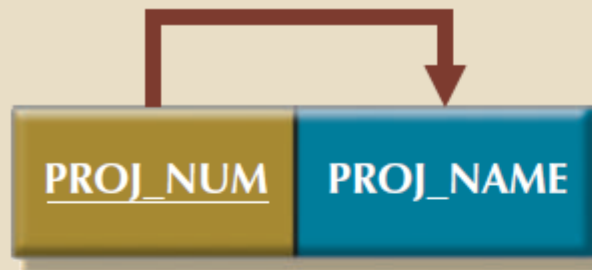


Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

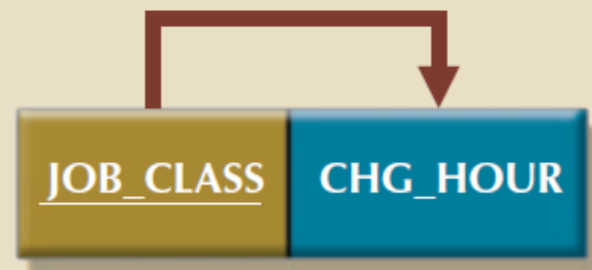


Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

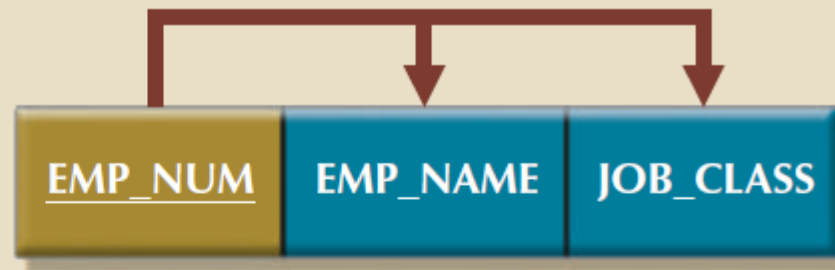


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

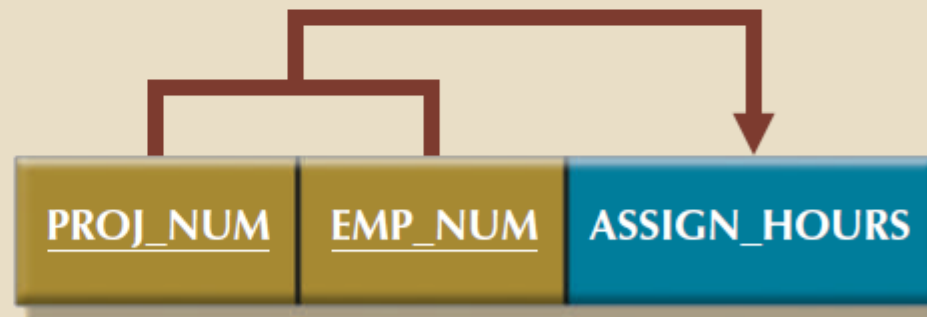


Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

Veritabanı Tasarımının İyileştirilmesi



Veritabanlarının bilgi sağlama yeteneğinin ve operasyonel özelliklerin iyi olması gerekir.



Normalleştirmenin kendi başına iyi tasarımlar yapmayı garantilememektedir.

Veritabanı Tasarımının İyileştirilmesi

1. Birincil anahtar alanların (PK) değerlendirilmesi
2. Adlandırma kurallarını yeniden gözden geçirilmesi
3. Atomik özniteliklere arındırma
4. Yeni özniteliklerin belirlenmesi

Veritabanı Tasarımının İyileştirilmesi

5. Yeni ilişkilerin belirlenmesi

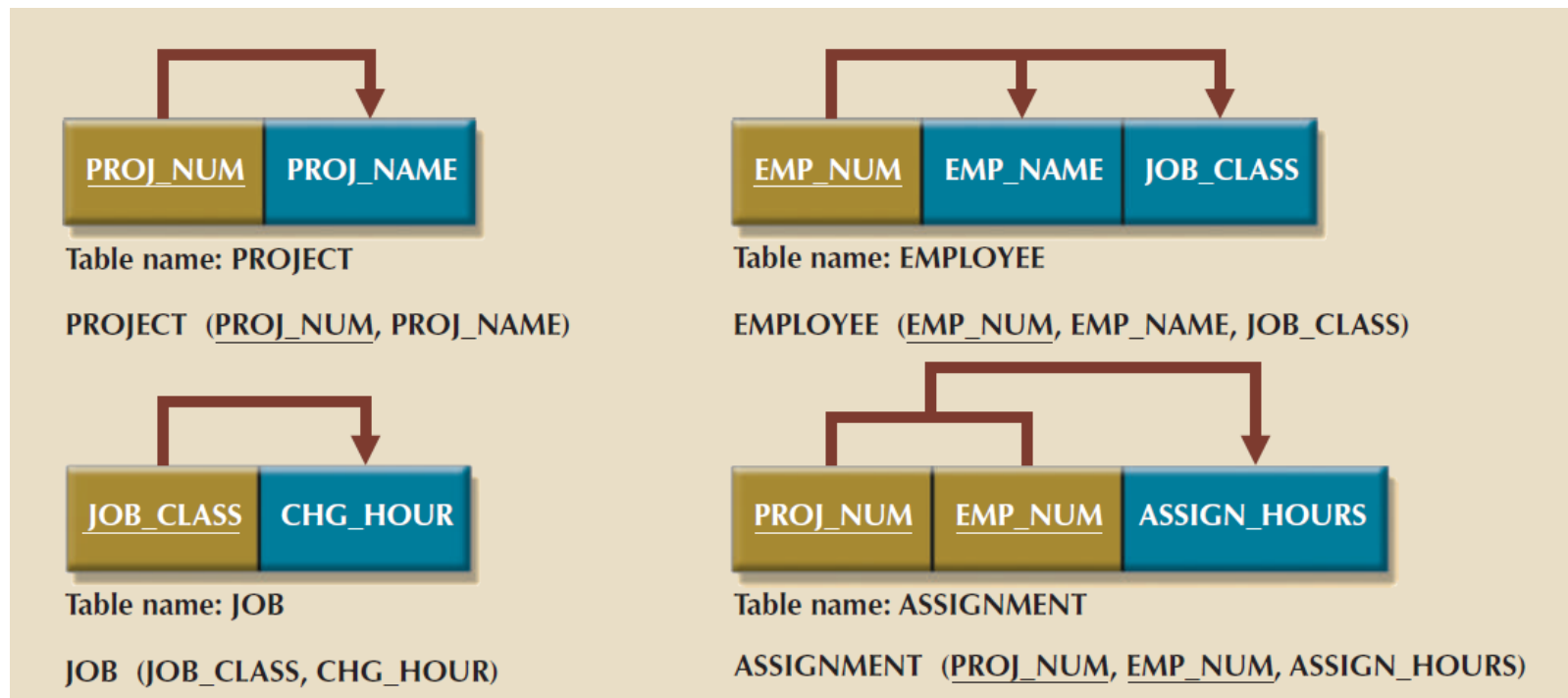
6. Veri ayrıntı düzeyi için anahtar alanların daraltılması

7. Geçmiş Kayıtların Sağlanması

8. Türetilmiş Nitelikler Kullanarak Değerlendirilmelerin Yapılması

Veritabanı Tasarımının İyileştirilmesi-ÖRNEK

- **PROJE** (PROJECT), **ÇALIŞAN** (EMPLOYEE), **İŞ** (JOB), **GÖREV** (ASSIGNMENT) tabloları üzerinden veritabanı tasarımının adım adım iyileştirilmesi yapılsın.



1. Birincil anahtar alanların (PK) değerlendirilmesi

Örneğin, EMPLOYEE tablosundaki **JOB_CLASS** özneliği için *Veritabanı Tasarımcısı* yerine **DBA** girilebilir.

Bu nedenle, benzersiz bir tanımlayıcı oluşturmak için bir **JOB_CODE** özneliği eklemek daha iyi olur.

JOB_CODE → **JOB_CLASS**, **CHG_HOUR**

1. Birincil anahtar alanların (PK) değerlendirilmesi

Daha önceki bağımlılık **JOB_CLASS** → **CHG_HOUR** şeklindeydi.

Bu durumda tabloda **JOB_CODE** ve **JOB_CLASS** olmak üzere iki aday anahtar alan vardır.

JOB_CODE alanının birincil anahtar alan olarak seçilmesi daha doğru olacaktır.

JOB_CODE, yedek (Surrogate) anahtar alan özelliğine sahiptir.

Yedek (Surrogate) anahtar alanlar



Genellikle sayısaldır.



Genellikle DBMS tarafından otomatik olarak oluşturulurlar.



Anlamsal içerikten arındırılmışlardır (özel bir anlamı yoktur).



Genellikle son kullanıcılardan gizlenirler.

2. Adlandırma kurallarının yeniden gözden geçirilmesi



Veri Modellerinde belirtilen adlandırma kurallarına uymak daha isabetli olacaktır.



Örnek olarak **1NF**'den **2NF**'ye dönüşümde **HOURS**'un **ASSIGN_HOURS** olarak değiştirilmiştir.

Bu değişiklik, çalışma saatlerini **ASSIGNMENT** tablosuyla ilişkilendirmenize olanak tanır.

2. Adlandırma kurallarının yeniden gözden geçirilmesi



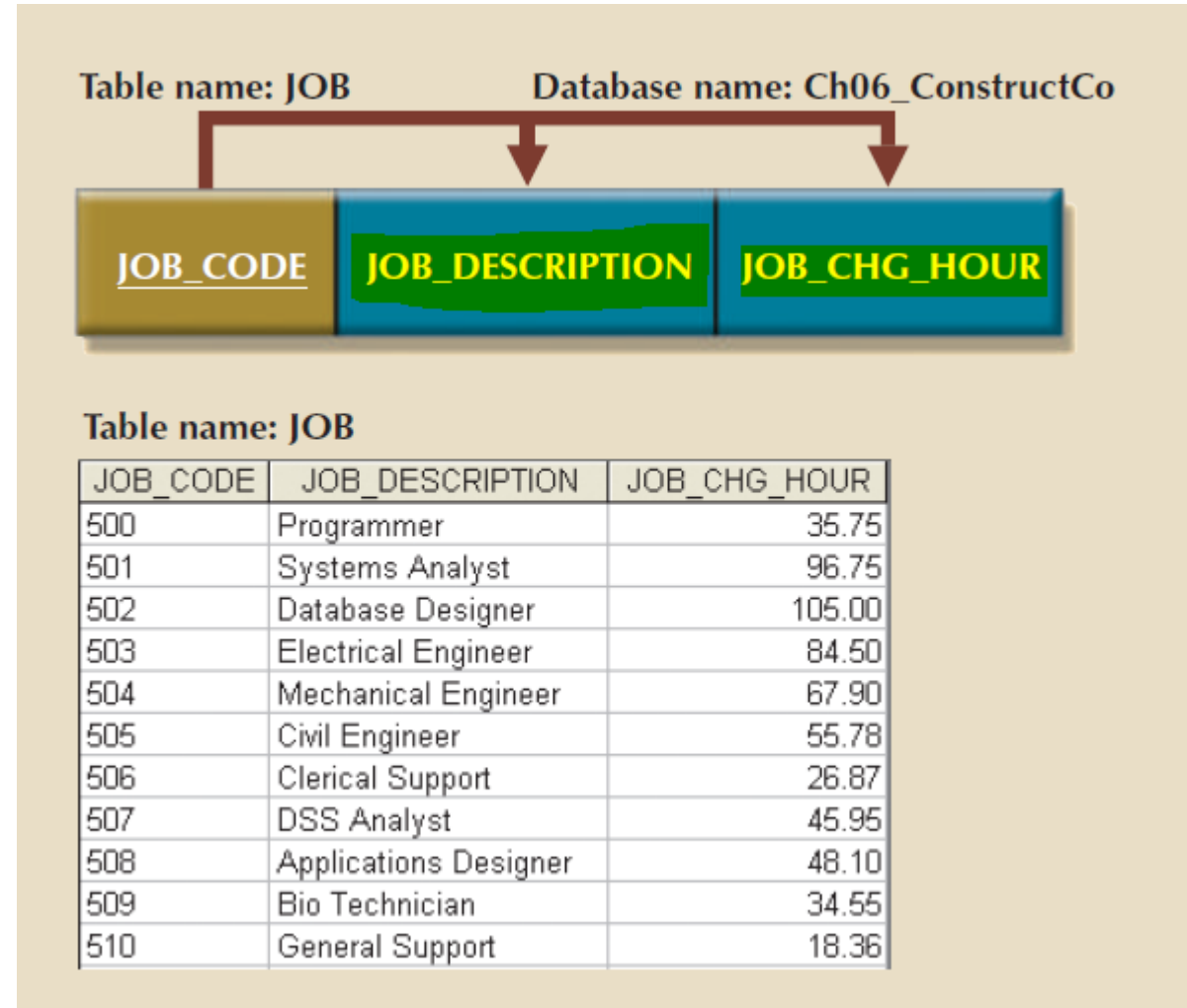
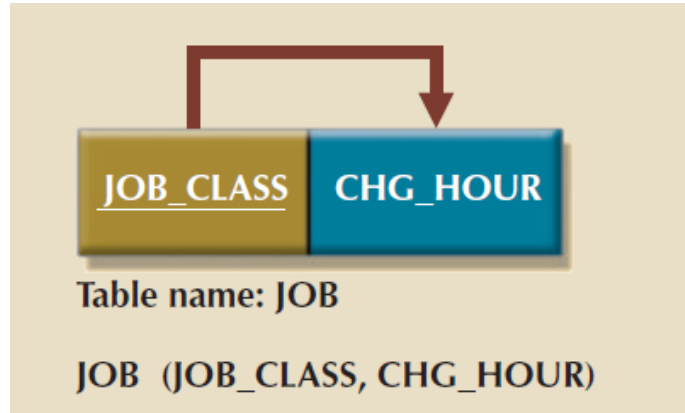
Dolayısıyla **CHG_HOUR**, **JOB_ CHG_HOUR** olarak değiştirilmelidir.



JOB_CLASS özneliği, *Sistem Analisti*, *Veritabanı Tasarımcısı* vb. değerleri tanımlamada yetersiz kalmaktadır.

Bunun yerine **JOB_DESCRIPTION** ismi daha iyi tanımlama olacaktır.

2. Adlandırma kurallarının yeniden gözden geçirilmesi



3. Atomik özniteliklere arındırma

- **Atomik özellik**, daha fazla alt bölümlere ayrılamayan bir nitelikdir.
 - Genel olarak, tasarımcılar iş kuralları ve işleme gerekliliklerinde belirtildiği gibi *basit, tek değerli* özellikleri kullanmayı tercih ederler.
- **EMP_NAME**'nin **EMPLOYEE** tablosunda kullanımını atomik değildir.
 - EMP_NAME bir soyadı, adı ve ikinci bir ad şeklinde ayrılabilir. Örneğin, *EMP_LNAME*, *EMP_FNAME* ve *EMP_INITIAL* şeklinde ayrılabilir.

3. Atomik özniteliklere arındırma

- Atomiklik derecesinin arttırılması veritabanı sorgulamalarında esneklik kazandırır.

Örneğin **soyadı**, **adı** ve **ikinci isimleri** sıralayarak telefon listelerini kolayca oluşturabilirsiniz.

Ad bileşenleri tek bir öznitelik içindeyse böyle bir görev çok zor olurdu.

3. Atomik özniteliklere arındırma

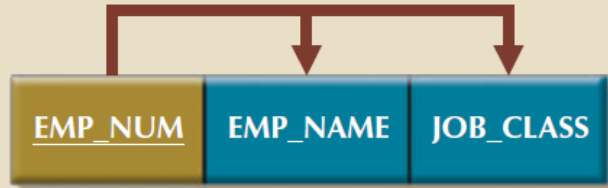


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)



Table name: EMPLOYEE

Database name: Ch06_ConstructCo

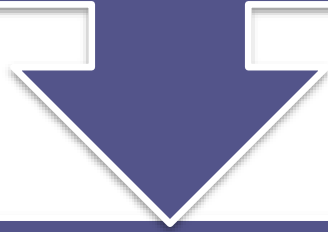


Table name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joebrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

4. Yeni özniteliklerin belirlenmesi

Veritabanının tasarımının iyileştirilmesinde yeni öznitelikler belirlenebilir.



Örneğin **EMPLOYEE** tablosunda, *yılbaşından bugüne brüt maaş ödemeleri, Sosyal Güvenlik ödemeleri ve sağlık ödemeleri yapılması* istenebilir.

4. Yeni özniteliklerin belirlenmesi

- ❑ Bir çalışanın *işe alım tarihi* *özniteliği* (**EMP_HIREDATE**), bir çalışanın iş çalışma süresini izlemek için kullanılabilir.
- ❑ Uzun vadeli çalışanlara *ikramiye verilmesi* ve *diğer ödüllendirme ödemeleri* için bir temel oluşturabilir.

Aynı prensip tasarımınızdaki diğer tüm tablolara da uygulanmalıdır.

4. Yeni öz niteliklerin belirlenmesi

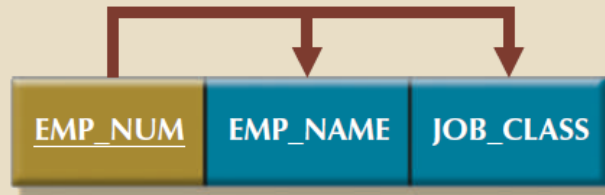


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)



Table name: EMPLOYEE

Database name: Ch06_ConstructCo



Table name: EMPLOYEE

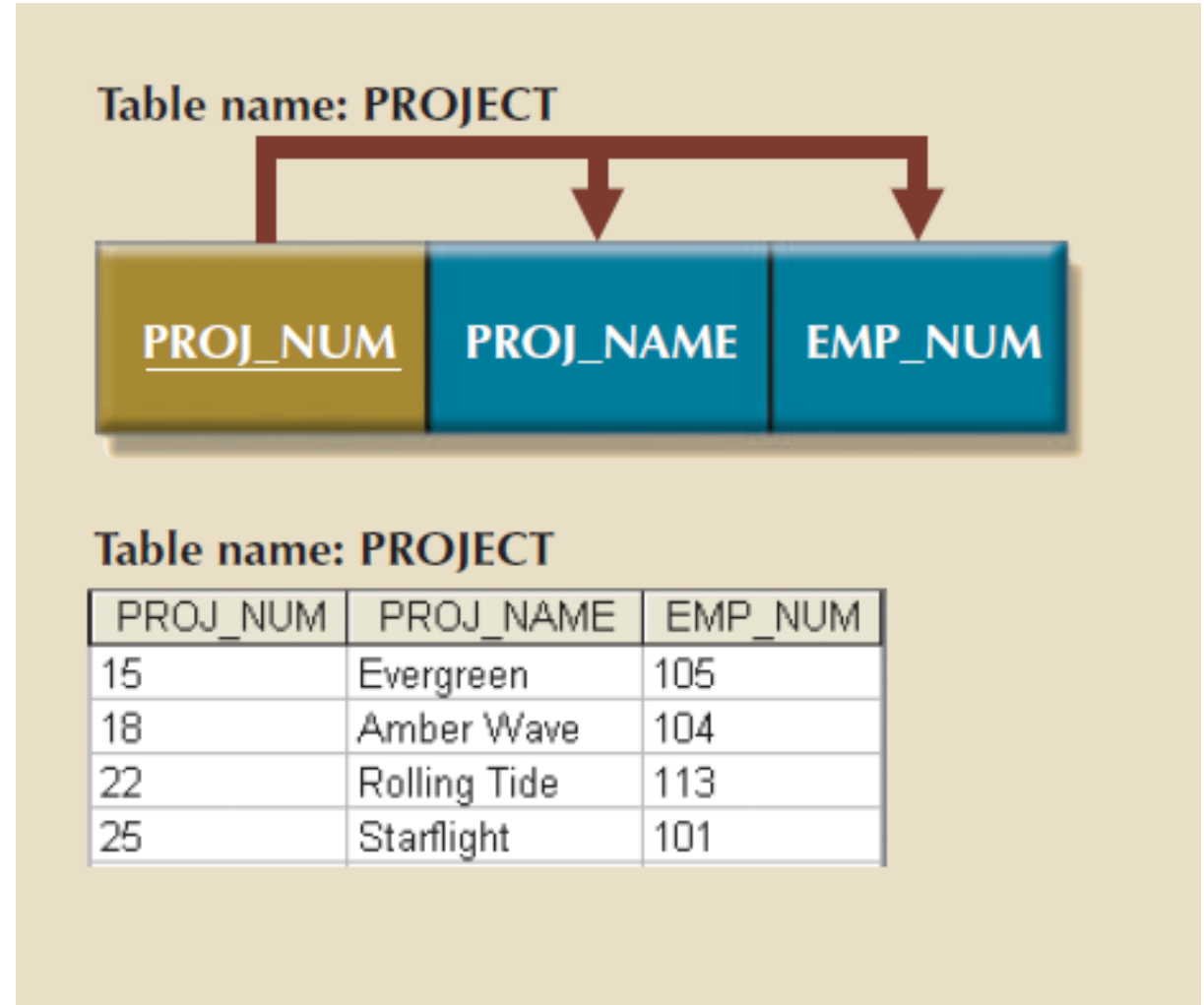
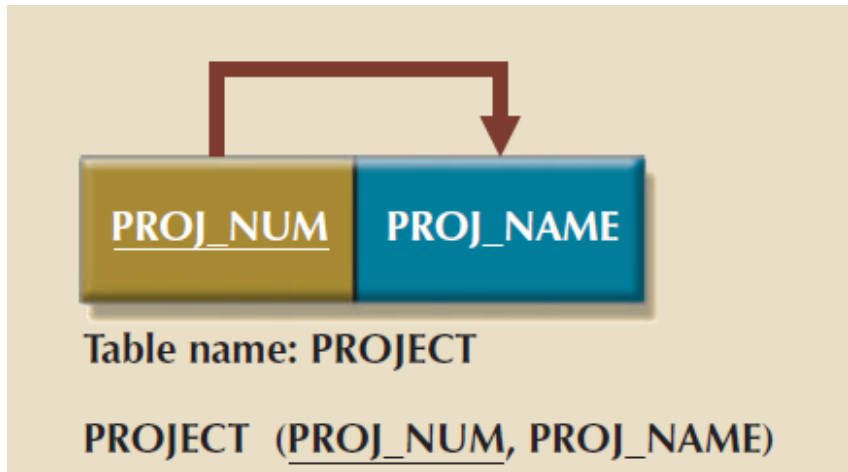
EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joenbrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

5. Yeni ilişkilerin belirlenmesi

□ Her projenin bir yöneticisi olabilir.

- Dolayısıyla, **ÇALIŞAN (EMPLOYEE)** ve **PROJE (PROJECT)** tabloları arasında proje yöneticisi hakkında bilgi sağlanması için bir ilişki kurulması gerekir.
- Bunun için **PROJE** tablosunda yabancı anahtar alan tanımlanmalıdır.
- Bu işlem, **gereksiz ve istenmeyen veri kopyaları üretmeden** her **PROJE'nin yönetici verilerinin ayrıntılarına erişilmesini sağlar.**

5. Yeni ilişkilerin belirlenmesi



6. Veri ayrıntı düzeyi için anahtar alanların ayrıştırılması

□ **Ayrıntı düzeyi (Granularity)**, tablonun satırında depolanan değerlerle temsil edilen detay düzeyini ifade eder.

- *En düşük ayrıntı düzeyinde saklanan verilerin **atomik veriler** olduğu söylenir.*

6. Veri ayrntı düzeyi için anahtar alanların ayrıştırılması

- Ancak, bu değerler en düşük ayrntı düzeylerinde kaydediliyor mu?
- Başka bir deyişle, **ASSIGN_HOURS**; günlük, haftalık, aylık veya yıllık toplamı mı temsil ediyor?
- **ASSIGN_HOURS** daha dikkatli bir tanımlanması gerekir.

6. Veri ayrıntı düzeyi için anahtar alanların ayrıştırılması



EMP_NUM ve **PROJ_NUM** birleşiminin **ASSIGNMENT** tablosunda kabul edilebilir (bileşik) bir birincil anahtar olabilir.

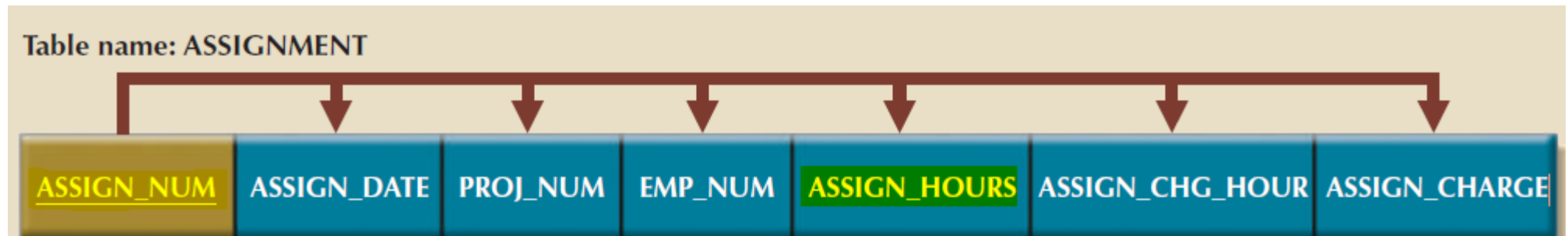
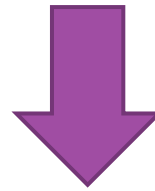
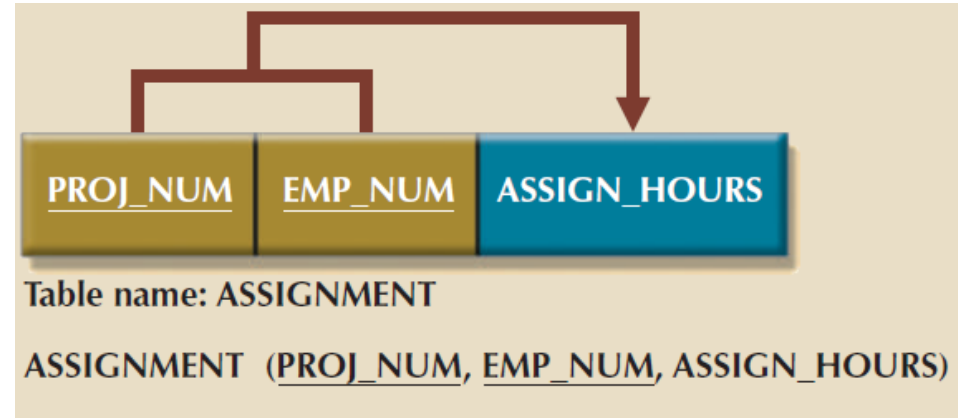


Fakat bu birincil anahtar, bir çalışanın projede yalnızca toplam çalışma saatini temsil etmekte kullanılabilir.



ASSIGN_NUM gibi bir yedek birincil anahtar (Surrogate) kullanmak daha düşük ayrıntı düzeyi sağlar ve daha fazla esneklik kazandırır.

6. Veri ayrıntı düzeyi için anahtar alanların ayrıştırılması



7. Geçmiş Kayıtların Sağlanması

Saatte iş ücretini ASSIGNMENT (GÖREV) tablosuna yazmak, tablonun verilerinin geçmiş doğruluğunu korumak için çok önemlidir.



Bu özelliği ASSIGN_CHG_HOUR olarak adlandırmak uygun olur. Saatlik iş ücretinin zamanla değişeceğini varsaymak mantıklıdır.

8. Türetilmiş Nitelikler Kullanarak Değerlendirilmelerin Yapılması

Bir projeye yapılan fiili masrafı saklamak için ASSIGNMENT (GÖREV) tablosunda türetilmiş bir öznitelik kullanabilir.



ASSIGN_CHARGE (TOPLAM ÜCRET) adlı türetilmiş öznitelik, ASSIGN_HOURS (GÖREVDE BULUNDUĞU SAAT) ile ASSIGN_CHG_HOUR (GÖREV BAŞI SAAT ÜCRETİ) çarpımının sonucu elde edilir.

8. Türetilmiş Nitelikler Kullanarak Değerlendirilmelerin Yapılması



Sistem işlevselliği açısından, bu tür türetilmiş öznitelik değerleri, **rapor veya fatura yazmak** gerektiğinde hesaplanabilir.



Türetilmiş özniteliğin tabloda depolanması, istenen sonuçları üretmek için uygulama yazılımının yazılmasını kolaylaştırır.



Birçok işlemin raporlanması ve / veya özetlenmesi gerekiyorsa, türetilen özelliğin kullanılabilirliği raporlama süresinden tasarruf sağlayacaktır.

8. Türetilmiş Nitelikler Kullanarak Değerlendirilmelerin Yapılması

Table name: ASSIGNMENT

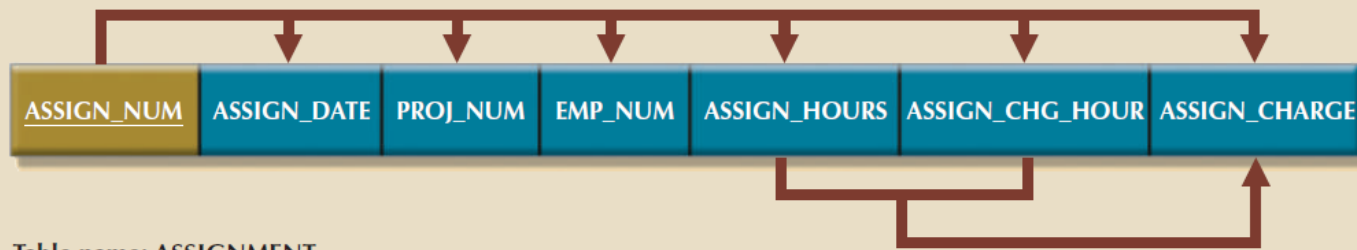


Table name: ASSIGNMENT

ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHG_HOUR	ASSIGN_CHARGE
1001	04-Mar-18	15	103	2.6	84.50	219.70
1002	04-Mar-18	18	118	1.4	18.36	25.70
1003	05-Mar-18	15	101	3.6	105.00	378.00
1004	05-Mar-18	22	113	2.5	48.10	120.25
1005	05-Mar-18	15	103	1.9	84.50	160.55
1006	05-Mar-18	25	115	4.2	96.75	406.35
1007	05-Mar-18	22	105	5.2	105.00	546.00
1008	05-Mar-18	25	101	1.7	105.00	178.50
1009	05-Mar-18	15	105	2.0	105.00	210.00
1010	06-Mar-18	15	102	3.8	96.75	367.65
1011	06-Mar-18	22	104	2.6	96.75	251.55
1012	06-Mar-18	15	101	2.3	105.00	241.50
1013	06-Mar-18	25	114	1.8	48.10	86.58
1014	06-Mar-18	22	111	4.0	26.87	107.48
1015	06-Mar-18	25	114	3.4	48.10	163.54
1016	06-Mar-18	18	112	1.2	45.95	55.14
1017	06-Mar-18	18	118	2.0	18.36	36.72
1018	06-Mar-18	18	104	2.6	96.75	251.55
1019	06-Mar-18	15	103	3.0	84.50	253.50
1020	07-Mar-18	22	105	2.7	105.00	283.50
1021	08-Mar-18	25	108	4.2	96.75	406.35
1022	07-Mar-18	25	114	5.8	48.10	278.98
1023	07-Mar-18	22	106	2.4	35.75	85.80

Daha Yüksek Seviye Normal Formlar

- **3NF düzeyinde tablolar, ticari işlem veritabanlarında uygun performans gösterirler.**
- **Ancak daha yüksek düzeydeki normal formlar bazı durumlarda faydalı özellikler gösterirler.**

Daha Yüksek Düzeydeki Normal Formlar :

- **Boyce-Codd Normal Form (BCNF)**
- **Dördüncü Normal Form (4NF)**

Boyce-Codd Normal Form (BCNF)

BCNF, 3NF Özel bir hali olarak düşünülebilir.

BCNF, tablo **yalnızca bir aday anahtarı içerdiğinde** 3NF'ye eşdeğerdir.

BCNF için aşağıdaki şartlar sağlanmalıdır.

- **Tablo 3NF düzeyinde olmalıdır.**
- Tablodaki **her belirleyici aday anahtar** olmalıdır.

Veritabanı tasarım kurallarına uyulduğunda genellikle 3NF ve BCNF formuna uygun tablolar olacaktır.

Bir tabloyu BCNF'ye dönüştürmek için tablo ayrı tablolara bölünmektedir.

ÖRNEK 1

- Fonksiyonel bağımlılıklar:

$$A + B \rightarrow C, D$$

$$A + C \rightarrow B, D$$

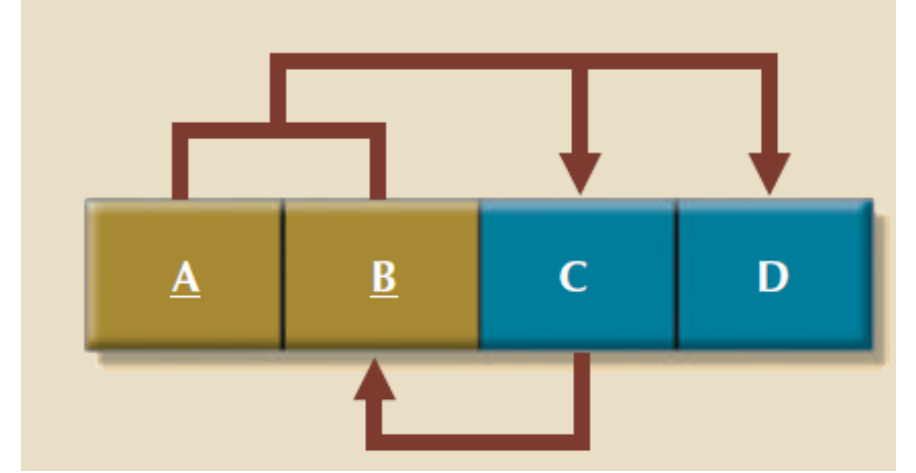
$$C \rightarrow B$$

olsun.

Bu tabloda aday anahtarlar (A+B, A+C) birden fazladır.

C belirleyicisi aday anahtar özelliği göstermemektedir.

3NF FAKAT BCNF DEĞİL

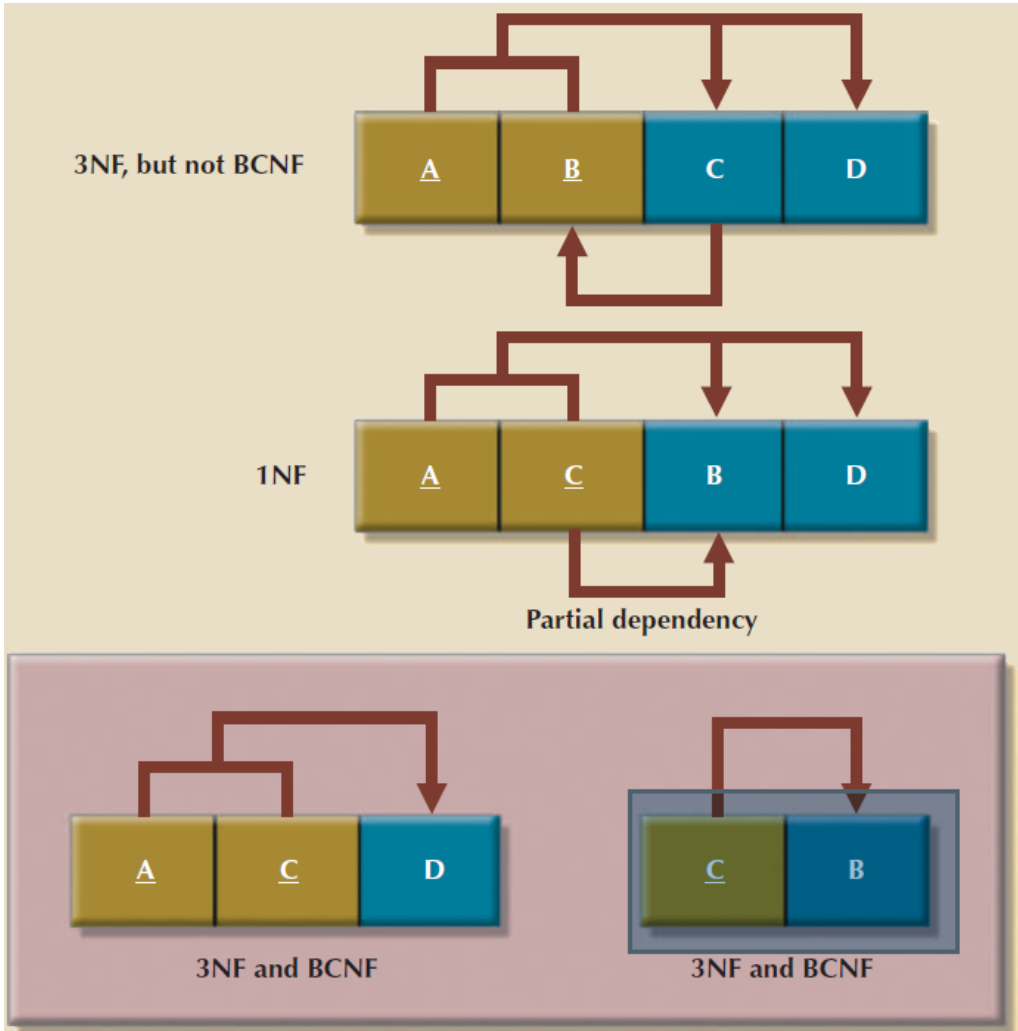


Tabloyu BCNF'ye dönüştürmek için ilk önce anahtar alan belirlenir.

Burada (A+C) birincil anahtar alan olarak değiştirilsin.

- Bu değişiklik uygundur çünkü $C \rightarrow B$ bağımlılığı, B'nin C'nin bir alt kümesi olduğu anlamına gelir.
- Ayrıca tablo bölündüğünde aday anahtar olmayan bağımlılıklar, aday anahtar olma durumuna getirilecektir.

ÖRNEK 1



■ Tablo ikiye bölünerek 3NF ve BCNF özelliği kazandı.

□ $C \rightarrow B$ kısmi bağımlılık kazanmış oldu.

□ C aday anahtar değilken anahtar alan özelliği kazandı.

Örnek 2

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

* BCNF özelliği çoğa-çok ilişkilerde oluşturulan ara tablolarda sağlanmayabilir.

- **Ders/kurs kodu(CLASS_CODE)** , asal olarak (uniquely) dersleri tanımlar.
- **Öğrenci**, birden fazla ders alır.
Örneğin **numarası (STU_ID) 125** olan öğrenci, **21334** ve **32456 ders koduna** sahip dersleri almıştır.
Bunlardan sırasıyla **A ve C dereceli**, notlar almıştır.
- **Bir öğretim elemanı** birden fazla ders verebilir.
Örneğin **personel numarası (STAFF_ID) 20** olan öğretim üyesi, **21334** ve **32456** ders koduna sahip dersleri vermiştir.

Fonksiyonel Bağımlılıklar:

STU_ID + STAFF_ID → **CLASS_CODE, ENROLL_GRADE**

CLASS_CODE → **STAFF_ID**

- Bir ders bir öğretim elemanı tarafından verildiğine göre tabloda ders koduna bakarak öğretim elemanına erişebiliriz.
- **Tüm belirleyiciler aday anahtar mı?**
- CLASS_CODE bu tabloda aday anahtar olamaz.

Örnek 2

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

Bu tabloda derslere personel atama ve öğrencileri kayıt bilgilerini tanımlamada problem oluşturacaktır. Bunlar:

❑ Güncelleme Sorunu

- **32456** ders koduna (CLASS_CODE) farklı bir personel atanırsa; iki satırı güncelleme gerektirir.

Bu nedenle bir güncelleme anomaliği oluşacaktır.

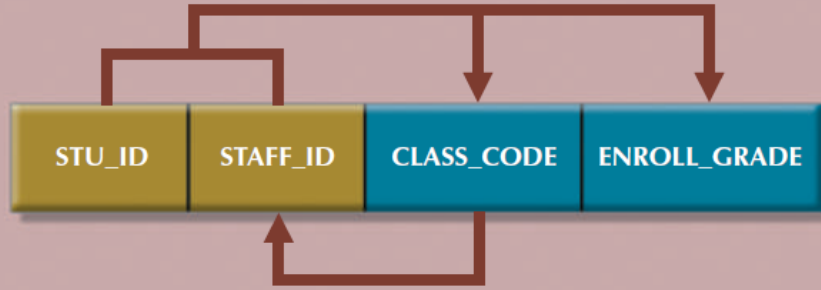
❑ Silme Sorunu

- **135** numaralı (STU_ID) öğrenci kaydı kaldırılırsa **28458** numaralı derste silinecektir. Dolayısıyla bu ders kim tarafından verildiği bilgisi de ortadan kalkacaktır.

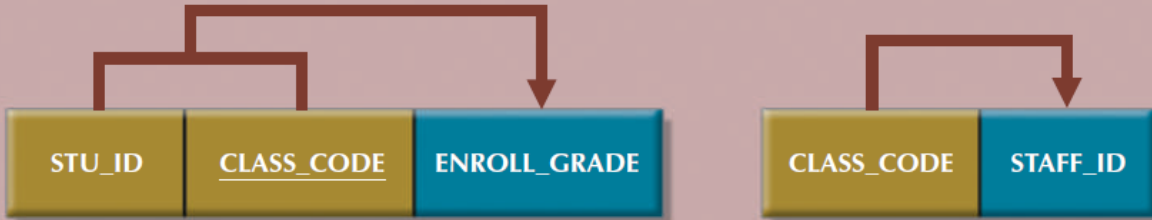
Bu tabloyu 3NF ve BCNF'ye uyumlu yapmak için tablo iki kısma bölünmüştür.

Örnek 2

Panel A: 3NF, but not BCNF



Panel B: 3NF and BCNF



Fonksiyonel Bağımlılık

$STU_ID + STAFF_ID \rightarrow CLASS_CODE, ENROLL_GRADE$

$CLASS_CODE \rightarrow STAFF_ID$ (Aday anahtar değil)

Dolayısıyla **CLASS_CODE** ve **STAFF_ID** alanları ayrı bir tablo olarak oluşturulur.

Böylece **CLASS_CODE** aday anahtar olmuş olur.

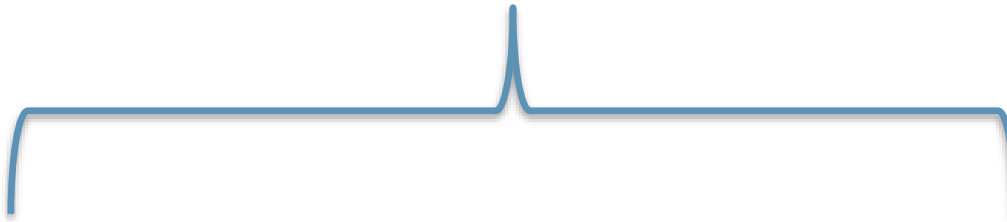
Yukarıdaki tablo ayrıştırılarak hem 3NF hem de BCNF'ye uyumlu olmuştur.

Oluşturulan iki tablo birbirleriyle ders kodu (CLASS_CODE) alanı ile ilişkilendirilmiştir. Yani bağlantıları kaybolmamıştır.

Fakat, bir tablo BCNF'ye uyumlu hale getirildiğinde her zaman bağlantılarının korunma garantisi yoktur.

Örnek 2

STU_ID	STAFF_ID	CLASS_CODE	ENROLL_GRADE
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B



STU_ID	CLASS_CODE	ENROLL_GRADE
125	21334	A
125	32456	C
135	28458	B
144	27563	C
144	32456	B

* Silme anomaliliği ortadan kaldırıldı.

* **135** öğrenci numarası silindiğinde **28458** ders koduna sahip dersi veren **öğretim üyesi** bilgisi kaybolmadı.

CLASS_CODE	STAFF_ID
21334	25
32456	20
28458	20
27563	25

* Güncelleme anomaliliği ortadan kaldırıldı.

32456 ders koduna sahip dersi veren **öğretim üyesi** güncelleneceği zaman **çift satırın güncellenmesi** problemi ortadan kaldırıldı.

4NF

Kötü tasarlanmış veritabanları ile karşılaşabiliriz. Bu nedenle tabloları 4NF düzeyine getirmemiz gerekebilir.

4NF’de bulunması gereken şartlar:

- Tablo, 3NF düzeyinde olmalıdır.
- Tabloda çoklu (birden fazla) değer bağımlılığı olmamalıdır.

4NF

Kurallar:

- Tüm öznitelikler **birincil anahtara bağlı olmalı**, ancak birbirlerinden bağımsız olmalıdır.
- Hiçbir satır, bir varlık hakkında iki veya daha fazla **çok değerli gerçekler** içermemelidir.

4NF-ÖRNEK

Çoklu değer bağımlılığı (multivalued dependencies), çoğa-çok ilişkilerde oluşturulan ara tablolarda karşımıza çıkabilmektedir.

Örneğin, bir çalışan birden fazla görev yapma olasılığı ve birden fazla organizasyonlarda gönüllü olarak yer alma durumu olabilir. Bu durum çok farklı yollarla nasıl kaydedilebileceği aşağıda gösterilmiştir.

FIGURE 6.11 TABLES WITH MULTIVALUED DEPENDENCIES

Database name: Ch06_Service

Table name: VOLUNTEER_V1

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

EMP_NUM: Çalışan numarası

ORG_CODE: Organizasyon numarası

ASSIGN_NUM: Görev numarası

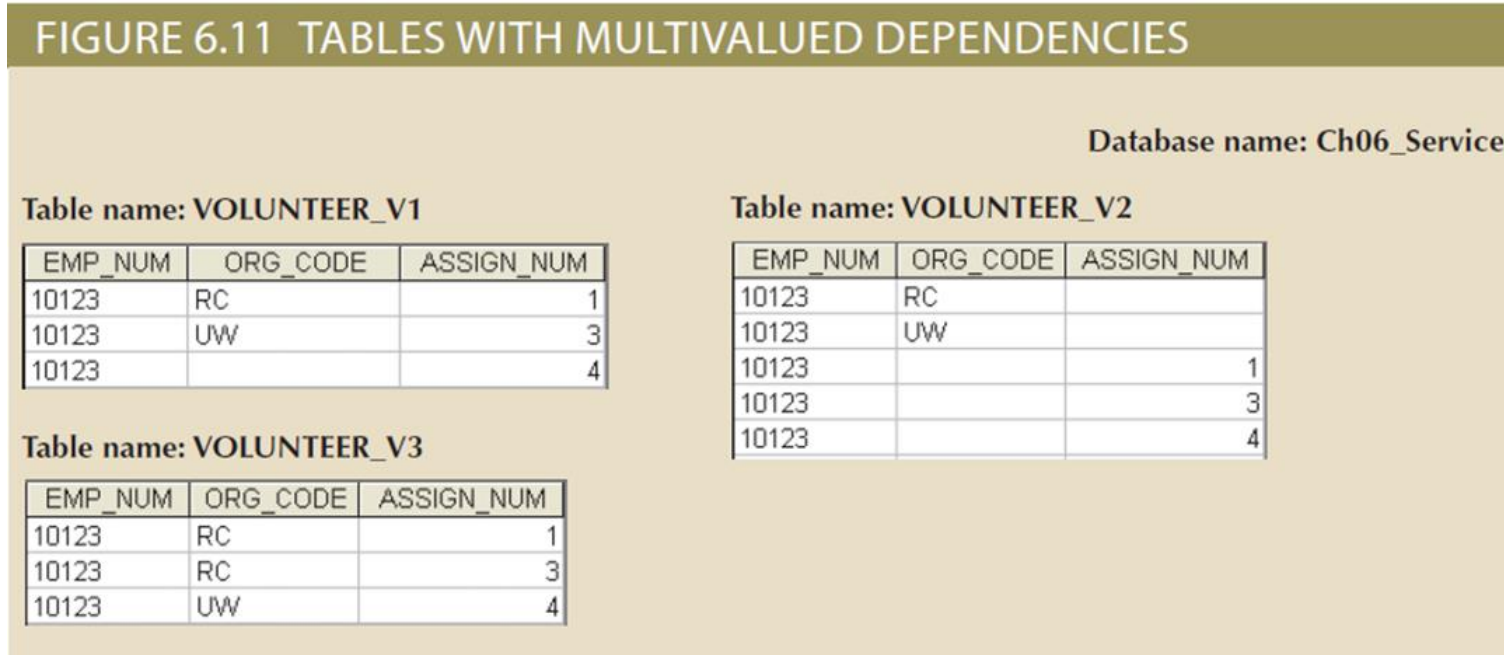
4NF-ÖRNEK

Tablolardan da görüldüğü gibi çalışan ile ilgili çoklu değerli gerçekler vardır.

Başka bir deyişle **EMP_NUM**; **ORG_CODE**'un ve **ASSIGN_NUM**'un birden çok değerini belirlemektedir.

Ancak **ORG_CODE** ve **ASSIGN_NUM** birbirinden bağımsızdır.

4NF-ÖRNEK



Tablolarda **birincil anahtar** olarak seçilecek bir alan yoktur.

Not: EMP_NUM kendine başına **birincil anahtar alan** olamamaktadır.

Tablo 1 ve 2'de EMP_NUM ile kombinasyon oluşturacak bir alan seçilip PK belirlenmeye çalışılsa bile boş kayıtlar olduğu için böyle bir imkan olamayacaktır.

4NF-ÖRNEK

Çok değerli bağımlılığın neden olduğu sorunlar ortadan kaldırmak için tablo 4NF dönüştürülür.

4NF'de çok değerli bağımlılığın bileşenleri için yeni tablolar oluşturulur.

4NF-ÖRNEK

FIGURE 6.11 TABLES WITH MULTIVALUED DEPENDENCIES

Database name: Ch06_Service

Table name: VOLUNTEER_V1

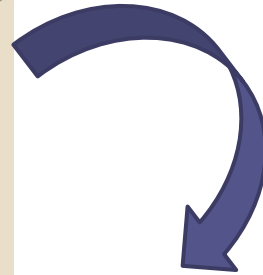
EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4



Database name: CH06_Service

Table name: PROJECT

PROJ_CODE	PROJ_NAME	PROJ_BUDGET
1	BeThere	1023245.00
2	BlueMoon	20198608.00
3	GreenThumb	3234456.00
4	GoFast	5674000.00
5	GoSlow	1002500.00

Table name: EMPLOYEE

EMP_NUM	EMP_LNAME
10121	Rogers
10122	O'Leery
10123	Panera
10124	Johnson

Table name: ORGANIZATION

ORG_CODE	ORG_NAME
RC	Red Cross
UW	United Way
WF	Wildlife Fund

Table name: ASSIGNMENT

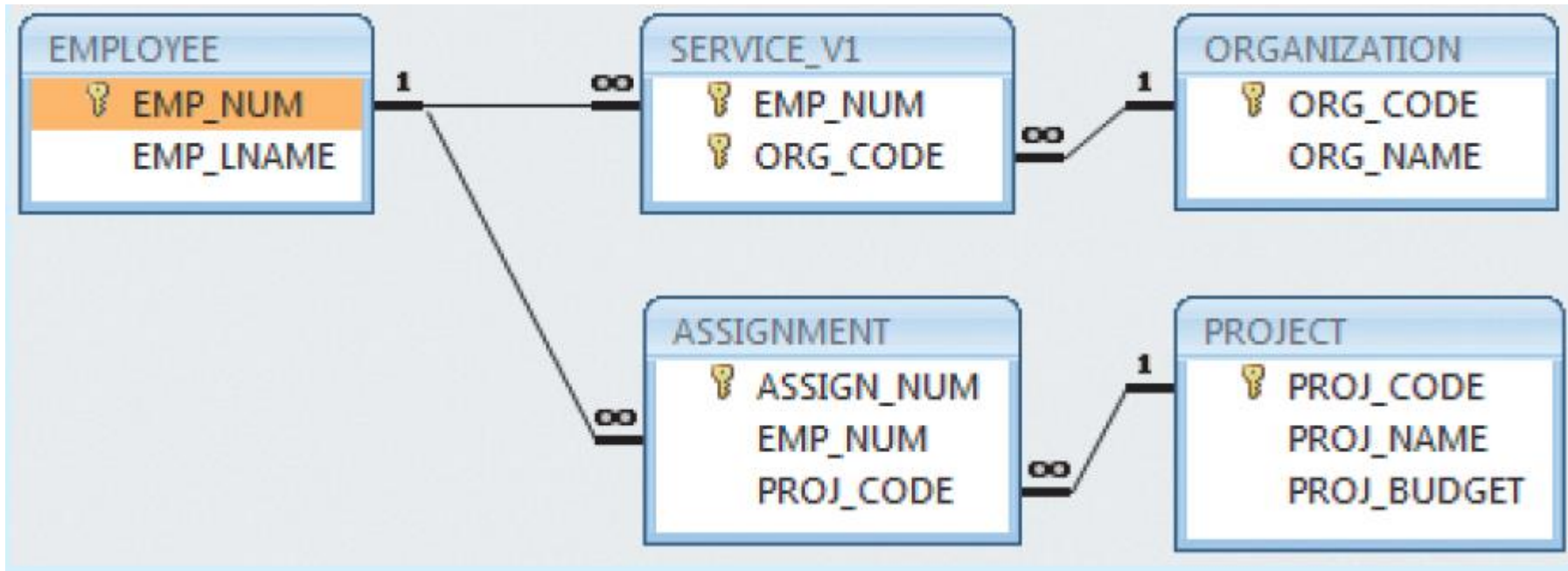
ASSIGN_NUM	EMP_NUM	PROJ_CODE
1	10123	1
2	10121	2
3	10123	3
4	10123	4
5	10121	1
6	10124	2
7	10124	3
8	10124	5

Table name: SERVICE_V1

EMP_NUM	ORG_CODE
10123	RC
10123	UW
10123	WF

4NF-ÖRNEK

İLİŞKİ DİYAGRAMI



Kaynakça

- Yalçın ÖZKAN, Veritabanı Sistemleri, Papatya Yayıncılık Eğitim, 4. Baskı
- Dr. Ali NİZAM, Veritabanı Teorisi ve Uygulamaları, Papatya Bilim, 2016.
- Dr. Ali NİZAM, Veritabanı İlişkisel Veri Modeli ve Uygulamaları, Papatya Bilim, 2011.
- Turgut Özseven, Veritabanı Yönetim Sistemleri-1 , Ekin Basın Yayın Dağıtım 6. Baskı
- Turgut Özseven, Veritabanı Yönetim Sistemleri-2 , Ekin Basın Yayın Dağıtım 5. Baskı