

B Ö L Ü M

4

Normalizasyon



4.1 Normalizasyon Nedir?

TANIM

Normalizasyon (normalleştirme); ilişkisel veri tabanının tasarım aşamasında veri tekrarını, veri kaybını veya veri yetersizliğini önlemek için uygulanan işlemlerdir. En basit ifadeyle verideki **gereksiz tekrarları temizleme** ve veriyi **mantıklı parçalara bölme** işidir.

Normalleştirme aynı zamanda “ilk taslak” veri tabanı tasarımının üzerinde düzeltmeler yapmanın, taslağı son hâline yaklaştırmamanın yöntemidir. Veri tabanı tasarlandıktan sonra normalizasyon kurallarına göre yeniden yapılandırılır.

DOLAP BENZETMESİ

Bir dolabı düzenlediğinizi düşünün: çoraplar bir çekmecedeyken, tişörtler başka bir rafta olmalıdır. Her şeyi tek bir büyük çuvala doldurursanız bir şeyi bulmak veya değiştirmek kâbusa dönüşür. Normalizasyon, veriler için bu düzeni kurar.

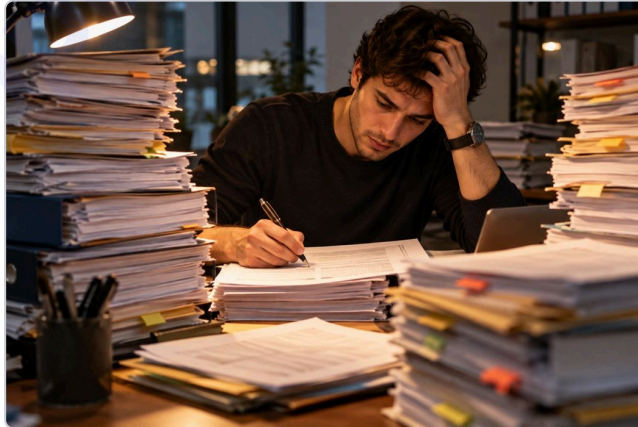
Normalizasyon Yapılmazsa: Üç Büyük Sorun

Ekleme sorunu

Yeni bir veri eklemek için ilgisiz başka verileri de girmek zorunda kalınır.

Güncelleme sorunu

Bir veri değiştirildiğinde aynı verinin geçtiği her yerin bulunup düzeltilmesi gerekir. Biri unutulursa veri tutarsızlaşır.



Güncelleme sorunu: aynı bilgi çok yerde duruyorsa her birinin tek tek düzeltilmesi gerekir

Silme sorunu

Bir satır silindiğinde, silinmek istenmeyen başka bir bilgi de kaybedilebilir.

4.2 Normalizasyonun Amaçları

Veri tekrarını azaltmak ve veri bütünlüğünü sağlamak	Veri gereksiz yere tekrarlanıyorsa bu kopyalar bir süre sonra birbirinden farklı değerler taşımaya başlar. Örneğin kişilerin adresi birden fazla tabloda tutuluyorsa zamanla aynı kişi için birden fazla adres oluşur. Adresin tek bir tabloda saklanması veri bütünlüğünü sağlar.
Uygulamadan bağımsızlık	İlişkisel model, kullanılacak uygulamaya göre değil verinin içeriğine göre kurulmalıdır. Böylece uygulama değişse bile veri modeli tutarlı ve sabit kalır.
Performansı artırmak	Tam normalleştirilmiş bir veri tabanında gereksiz kopyalanmış veri en aza iner. Daha az kopya, daha az saklama alanı ve daha kısa arama süresi demektir.

4.3 Anahtarlar ve Fonksiyonel Bağımlılık

Bileşik (kompozit) anahtar	İki veya daha fazla sütundan oluşan birincil anahtardır. Tek sütun satırı benzersiz yapamadığında kullanılır.
-----------------------------------	---

TANIM

Fonksiyonel (işlevsel) bağımlılık, bir sütunun değerinin başka bir sütuna bağlı olmasıdır. A sütunu bilindiğinde B sütunu kesin olarak bulunabiliyorsa “B, A’ya fonksiyonel bağımlıdır” denir ve $A \rightarrow B$ şeklinde gösterilir.

Örneğin müşteri numarası birincil anahtar ise kişinin adı soyadı bu anahtara fonksiyonel bağımlıdır: numarası bilinen müşterinin adı ve şehri bulunabilir. Başka bir niteliğin kendisine bağımlı olduğu niteliğe **belirleyici** denir.

Bağımlılık türü	Anlamı	Örnek
Tam bağımlılık	Anahtar olmayan sütun, birincil anahtarı oluşturan tüm sütunlara birlikte bağlıdır.	(Öğrenci No, Kurs Kodu) $\rightarrow$ Not
Kısmi bağımlılık	Sütun, bileşik anahtarın tamamına değil yalnızca bir parçasına bağlıdır.	Öğrenci No $\rightarrow$ Öğrenci Adı
Geçişli bağımlılık	Sütun, birincil anahtara değil anahtar olmayan başka bir sütuna bağlıdır; anahtara dolaylı bağlanır.	Öğrenci No $\rightarrow$ Posta Kodu $\rightarrow$ Şehir

**HATIRLATMA**

Kısmi bağımlılık ancak **bileşik anahtar** varsa ortaya çıkabilir. Birincil anahtar tek sütundan oluşuyorsa kısmi bağımlılık olamaz.

4.4 Normal Formlar

Normalizasyon yapılırken uyulması gereken kuralların her birine **normal form** denir: 1NF, 2NF, 3NF, BCNF, 4NF ve 5NF. Üst normal formlar alt normal formları kapsar: 3NF'deki bir tablo 1NF ve 2NF'ye de uygundur. İlk üç normal form çok kullanılır; normalizasyon genellikle 3NF'ye kadar yapılır ve bu seviye çoğu iş uygulaması için yeterince temiz bir yapı sunar.

Birinci Normal Form (1NF): Atomik Yapı

- ◆ Bir hücrede birden fazla değer olamaz; her hücre tek ve bölünemez (atomik) bir değer taşır.
- ◆ Bir alandaki bilgi virgül gibi özel karakterlerle ayrılarak tutulamaz.
- ◆ Tekrar eden sütun grupları olamaz.

1NF'ye uymayan tablo

OgrID	OgrAd	Telefon	Dersler	Notlar
101	Ali Kaya	555-111, 555-222	CS101, CS201, MT101	AA, BB, AA
102	Ayşe Er	555-333	MT101, MT201	BA, AA

Hücrelerde liste vardır; “CS101 dersini alanlar kimler?” diye sorgu yazılamaz. Çözüm, liste değerleri satırlara bölmektir:

1NF						
OgrID	PK	OgrAd	Telefon	DersKod	PK	Not
101		Ali Kaya	555-111	CS101		AA
101		Ali Kaya	555-111	CS201		BB
102		Ayşe Er	555-333	MT101		BA
102		Ayşe Er	555-333	MT201		AA

Artık her hücre atomiktir ve tablo sorgulanabilir. Ancak veri tekrarı sürmektedir; bunu sonraki adımlar çözer.

İkinci Normal Form (2NF): Kısmi Bağımlılık Yok

Tablo 1NF’de olmalı ve anahtar olmayan her sütun birincil anahtarın **tamamına** bağlı olmalıdır. Kural: “Bu bilgi kimin?” diye sor, o tabloya koy.

OgrID → OgrAd — yalnızca OgrID’ye bağlı: kısmi bağımlılık ✗

DersKod → DersAd — yalnızca DersKod’a bağlı: kısmi bağımlılık ✗

(OgrID, DersKod) → Not — ikisine birden bağlı: tam bağımlılık ✓

OGRENCİ			DERS		
OgrID	PK	OgrAd	BolumKod	FK	
101		Ali Kaya	CS		
102		Ayşe Er	MT		

DersKod	PK	DersAd	Kredi
CS101		Veri Yapıları	3
MT101		Calculus	4

KAYIT					
OgrID	PK, FK	DersKod	PK, FK	Not	Dönem
101		CS101		AA	2024-G
102		MT101		BA	2024-G

Üçüncü Normal Form (3NF): Geçişli Bağımlılık Yok

Tablo 2NF’de olmalı ve anahtar olmayan hiçbir sütun, anahtar olmayan başka bir sütuna bağlı olmamalıdır. Kural: “Bu bilgi başka bir bilgiyi mi belirliyor?” Evet ise ayrı tabloya.

3NF’ye uymayan tablo					
OgrID	PK	OgrAd	BolumKod	BolumAd	BolumTel
101		Ali Kaya	CS	Bilgisayar	0312-111
102		Ayşe Er	MT	Matematik	0312-222
103		Can Öz	CS	Bilgisayar	0312-111

Zincir: OgrID → BolumKod → BolumAd, BolumTel. Bölüm telefonu öğrencinin değil **bölümün** bilgisidir. Çözüm:

BOLUM			OGRENCI		
BolumKod PK	BolumAd	BolumTel	OgrID PK	OgrAd	BolumKod FK
CS	Bilgisayar	0312-111	101	Ali Kaya	CS
MT	Matematik	0312-222	102	Ayşe Er	MT
			103	Can Öz	CS

Özet

Form	Odak noktası	Kural	Yapılan işlem
1NF	Atomik veri	Her hücre tek değer taşır.	Liste değerleri satırlara böl.
2NF	Kısmi bağımlılık	Anahtar olmayan her sütun, birincil anahtarın tamamına bağlı olmalıdır.	Anahtarın yalnızca bir kısmına bağlı sütunları ayır.
3NF	Geçişli bağımlılık	Hiçbir sütun anahtara başka bir sütun üzerinden dolaylı bağlı olmamalıdır.	Anahtar olmayan bir sütuna bağlı olan sütunları ayır.

4.5 Uygulama: Kurs Kayıt Tablosu

Başlangıçta “kirli” ve verimsiz bir tablo vardır. Sütunları: ÖğrenciNo, ÖğrenciAdı, DersKodu, DersAdı, EğitimciAdı, EğitimciOda, Not.

Adım 0: Başlangıç (Normal Olmayan Form)

KURS KAYIT						
ÖğrenciNo	ÖğrenciAdı	DersKodu	DersAdı	EğitimciAdı	EğitimciOda	Not
101	Yunus Emre Demir	MAT1, FIZ1	Matematik, Fizik	Ayşe Fen, Can Bilim	A-101, B-202	80, 70

Sorun: Bir hücrede birden fazla ders ve not vardır. Veri tabanı bunu tek bir metin gibi görür; üzerinde matematiksel işlem yapılamaz.

Adım 1: 1NF (Hücreleri Parçala)

KURS KAYIT — 1NF						
ÖğrenciNo PK	ÖğrenciAdı	DersKodu PK	DersAdı	EğitimciAdı	EğitimciOda	Not
101	Yunus Emre Demir	MAT1	Matematik	Ayşe Fen	A-101	80
101	Yunus Emre Demir	FIZ1	Fizik	Can Bilim	B-202	70

Artık her hücre atomiktir. Ancak yalnızca öğrenci numarası satırı benzersiz yapmadığı için (aynı öğrenci iki kez geçiyor) **bileşik anahtar** kullanılır: (ÖğrenciNo, DersKodu).

Adım 2: 2NF (Kısmi Bağımlılığı Bitir)

ÖğrenciNo → ÖğrenciAdı — yalnızca öğrenciye bağlı ✗

DersKodu → DersAdı, EğitimciAdı, EğitimciOda — yalnızca derse bağlı ✗

(ÖğrenciNo, DersKodu) → Not — hem öğrenciye hem derse bağlı ✓

ÖĞRENCİ

ÖğrenciNo PK	ÖğrenciAdı
101	Yunus Emre Demir

KAYIT

ÖğrenciNo PK, FK	DersKodu PK, FK	Not
101	FIZ1	70
101	MAT1	80

DERS

DersKodu PK	DersAdı	EğitmenAdı	EğitmenOda
FIZ1	Fizik	Can Bilim	B-202
MAT1	Matematik	Ayşe Fen	A-101

Adım 3: 3NF (Geçişli Bağımlılığı Bitir)

DERS tablosunda EğitimciOda bilgisi anahtar olan DersKodu'na değil, anahtar olmayan EğitimciAdı'na bağlıdır: eğitimci değişirse oda da değişir. Zincir: DersKodu → Eğitimci → EğitimciOda. Eğitimci bilgileri ayrılır:

Ders (DersKodu, DersAdı, EğitimciID)

Eğitmen (EğitmenID, EğitimciAdı, EğitimciOda)

Sonuç: Dört Küçük ve Mantıklı Tablo

Tablo	Cevapladığı soru
Öğrenciler	Kim kimdir?
Eğitmenler	Hangi hoca nerede oturur?
Dersler	Hangi dersi kim verir?
Kayıtlar (Notlar)	Kim hangi dersten kaç aldı?

NEDEN YAPTIK?

Ayşe Hoca odasını değiştirirse yalnızca Öğitmenler tablosunda **tek bir satır** güncellenir. Eski yapıda, Ayşe Hoca'nın girdiği yüzlerce ders kaydının tek tek bulunup güncellenmesi gerekirdi.

Tabloların SQL Koduna Dönüştürülmesi

Önce bağımsız tablolar (Öğitmenler, Öğrenciler), sonra onlara bağlanan tablolar oluşturulur. Tablolar arasındaki ilişkiler yabancı anahtar ile kurulur.

```
CREATE DATABASE KursKayit
GO
USE KursKayit

CREATE TABLE Egitmenler (
    EgitmenID INT PRIMARY KEY,
    EgitmenAdi VARCHAR(100),
    EgitmenOdaNo VARCHAR(10)
);

CREATE TABLE Ogrenciler (
    OgrenciNo INT PRIMARY KEY,
    OgrenciAdi VARCHAR(100)
);

CREATE TABLE Dersler (
    DersKodu VARCHAR(10) PRIMARY KEY,
    DersAdi VARCHAR(100),
    EgitmenID INT,
    -- "Bu dersi kim veriyor?" sorusunun cevabı Egitmenler tablosuna bakar
    FOREIGN KEY (EgitmenID) REFERENCES Egitmenler(EgitmenID)
);

CREATE TABLE Kayitlar (
    OgrenciNo INT,
    DersKodu VARCHAR(10),
    Notu INT,
    PRIMARY KEY (OgrenciNo, DersKodu), -- bileşik birincil anahtar
    FOREIGN KEY (OgrenciNo) REFERENCES Ogrenciler(OgrenciNo),
    FOREIGN KEY (DersKodu) REFERENCES Dersler(DersKodu)
);
```

Kayıtlar tablosu çoğa-çok ilişkiyi yönetir: bir öğrenci birçok ders alabilir, bir dersi birçok öğrenci alabilir. Veriler de aynı sırayla eklenir: önce bağımsız veriler, sonra ilişkili veriler.

```

INSERT INTO Egitmenler VALUES (1, 'Ayşe Fen', 'A-101');
INSERT INTO Egitmenler VALUES (2, 'Can Bilim', 'B-202');
INSERT INTO Ogrenciler VALUES (101, 'Yunus Emre Demir');
INSERT INTO Ogrenciler VALUES (102, 'Veli Can');

INSERT INTO Dersler VALUES ('MAT1', 'Matematik', 1); -- Ayşe Fen veriyor
INSERT INTO Dersler VALUES ('FIZ1', 'Fizik', 2);      -- Can Bilim veriyor

INSERT INTO Kayitlar VALUES (101, 'MAT1', 80);
INSERT INTO Kayitlar VALUES (101, 'FIZ1', 70);
INSERT INTO Kayitlar VALUES (102, 'MAT1', 95);

```

JOIN ile Veriyi Yeniden Birleştirme

Normalizasyonun gücü buradadır: veri ayrı tutulur, ama raporlanırken tek bir tabloymuş gibi birleştirilir. Öğrencinin adı, aldığı dersin adı, notu ve dersi veren eğitmen tek sorguyla getirilir:

```

SELECT O.OgrenciAdi, D.DersAdi, K.Notu, E.EgitmenAdi
FROM Kayitlar K
JOIN Ogrenciler O ON K.OgrenciNo = O.OgrenciNo -- öğrenci adını getir
JOIN Dersler D ON K.DersKodu = D.DersKodu -- ders adını getir
JOIN Egitmenler E ON D.EgitmenID = E.EgitmenID; -- eğitmen adını getir

```

SONUÇ

OgrenciAdi	DersAdi	Notu	EgitmenAdi
Yunus Emre Demir	Matematik	80	Ayşe Fen
Yunus Emre Demir	Fizik	70	Can Bilim
Veli Can	Matematik	95	Ayşe Fen

4.6 Örnek: DAĞITIM Tablosu

DAĞITIM (müşteri_no, şehir_kodu, şehir_adı, gönderi_no, miktar) tablosu müşterilere yapılan gönderileri tutmaktadır. İlişkisel modelin temel kuralına göre bütün niteliklerin aldığı değerler atomik (tek ve basit) olmalıdır; aşağıdaki tablo bu kurala uymadığı için **normal değildir**.



DAĞITIM tablosu, müşterilere yapılan gönderileri ve miktarlarını tutar

DAĞITIM – normal olmayan form

müşteri_no	şehir_kodu	şehir_adı	gönderi_no	miktar
1	34	İstanbul	1, 2, 3, 4, 6	300, 200, 400, 200, 100
2	6	Ankara	1, 2	300, 400
3	6	Ankara	2	200
4	34	İstanbul	2, 4, 5	200, 300, 400

Birinci Normal Form

Çoklu değer içeren gönderi_no ve miktar alanları satırlara ayrıştırılır. gönderi_no tek başına satırları ayırt edemez (2 numaralı gönderi hem 1 hem 2 numaralı müşteride vardır); müşteri_no da tek başına yetmez. Bu yüzden bileşik anahtar tanımlanır: (**müşteri_no**, **gönderi_no**).

DAĞITIM – 1NF

müşteri_no PK	şehir_kodu	şehir_adı	gönderi_no PK	miktar
1	34	İstanbul	1	300
1	34	İstanbul	2	200
1	34	İstanbul	3	400
1	34	İstanbul	4	200
1	34	İstanbul	6	100
2	6	Ankara	1	300
2	6	Ankara	2	400
3	6	Ankara	2	200
4	34	İstanbul	2	200
4	34	İstanbul	4	300
4	34	İstanbul	5	400

Tablodaki işlevsel bağımlılıklar:

müşteri_no → **şehir_kodu**, **şehir_adı** — anahtarın yalnızca bir parçasına bağlı: kısmi bağımlılık ✗

(müşteri_no, gönderi_no) → **miktar** — anahtarın tamamına bağlı: tam bağımlılık ✓

şehir_kodu → **şehir_adı** — anahtar olmayan bir alana bağlı: geçişli bağımlılık ✗

Birinci Normal Formun Sorunları

1NF'deki tabloda şehir_kodu ve şehir_adı alanları her müşteri için tekrar etmektedir. Bu tekrarlar üç soruna yol açar:

Satır ekleme	Yeni bir müşterinin bilgilerinin (5, 35, İzmir) girilebilmesi için o müşteriye mutlaka bir dağıtım yapılmış, yani gönderi_no ve miktar değerlerinin girilmiş olması gerekir.
Satır silme	Bir müşteriye tek bir dağıtım yapıldıysa (3 numaralı müşteri), o dağıtım iptal edildiğinde yalnızca gönderi ve miktar değil, müşteri hakkındaki diğer bilgiler de yok olur.
Güncelleme	1 numaralı müşteri Ankara'ya taşınırsa bu müşteriyle ilgili tüm satırların güncellenmesi gerekir. Tablo çok büyükse küçük bir değişiklik binlerce kaydın güncellenmesini gerektirebilir.

İkinci Normal Form

şehir_kodu ve şehir_adı, bileşik anahtarın yalnızca müşteri_no parçasına bağlıdır. Bu nitelikler müşteri_no ile birlikte ayrı bir tablo oluşturur. miktar ise hem müşteri hem gönderi numarasına ihtiyaç duyduğu için (tam bağımlılık) yerinde kalır.

MÜŞTERİLER			MİKTARLAR		
müşteri_no PK	şehir_kodu	şehir_adı	müşteri_no PK, FK	gönderi_no PK	miktar
1	34	İstanbul	1	1	300
2	6	Ankara	1	2	200
3	6	Ankara	1	3	400
4	34	İstanbul	1	4	200
			1	6	100
			2	1	300
			...	...	...

11 satırın tamamı 1NF tablosundaki gibidir.

2NF'nin sorunları: Güncelleme sorunu ortadan kalkmıştır; ancak ekleme ve silme sorunları sürer. MÜŞTERİLER tablosuna yeni bir müşteri girilmedikçe yeni bir şehir tanımlanamaz (İzmir'i ekleyebilmek için İzmir'de bir müşteri gerekir). Bir müşteri silindiğinde, o şehirdeki tek müşteriye şehir bilgisi de yok olur.

Üçüncü Normal Form

MÜŞTERİLER tablosunda şehir_adı doğrudan müşteri_no'ya değil, anahtar olmayan şehir_kodu alanına bağlıdır (geçişli bağımlılık). Bu bağımlılık ayrı bir tabloya taşınır:

ŞEHİRLER		MÜŞTERİLER	
şehir_kodu PK	şehir_adı	müşteri_no PK	şehir_kodu FK
6	Ankara	1	34
34	İstanbul	2	6
35	İzmir	3	6
		4	34
		5	35

MİKTARLAR tablosu olduğu gibi kalır.

Tablo	Birincil anahtar (PK)	Yabancı anahtar (FK)
Şehirler	şehir_kodu	—
Müşteriler	müşteri_no	şehir_kodu
Miktarlar	müşteri_no + gönderi_no (bileşik)	müşteri_no

SONUÇ

Artık İzmir, orada hiç dağıtım yapılmamış olsa bile ŞEHİRLER tablosunda tek başına durabilir; 5 numaralı müşteri de gönderisi olmadan kaydedilebilir.

4.7 BCNF ve Dördüncü Normal Form

Boyce–Codd Normal Formu (BCNF)

TANIM

Bir tabloda **her belirleyici bir aday anahtar** ise tablo BCNF'dedir. (Belirleyici: başka bir niteliğin kendisine tam işlevsel bağımlı olduğu nitelik.)

ÖĞRENCİ (ÖğrNo, Bölüm, Danışman) tablosunda bir öğrenci birden çok bölüme kayıtlı olabilir ve her bölümde bir danışmanı vardır; her danışman tek bir bölümde görev yapar.

ÖĞRENCİ		
ÖğrNo	Bölüm	Danışman
123	Fizik	A. ERCAN
123	Kimya	M. AKINCI
456	Biyoloji	K. SÖNMEZ
789	Fizik	A. ERCAN
999	Kimya	B. ÖZKAN

(ÖğrNo, Bölüm) → Danışman — birincil anahtar

Danışman → Bölüm — Danışman anahtar değil ✗

Birincil anahtar: (ÖğrNo, Bölüm)

Aday anahtar: (ÖğrNo, Danışman)

Tablo 1NF (değerler atomik), 2NF (kısmi bağımlılık yok) ve 3NF (geçişli bağımlılık yok) koşullarını sağlar. Yine de sorun vardır: 456 numaralı öğrenci silinirse Biyoloji bölümü ve K. SÖNMEZ bilgisi de yok olur; bir öğrenci Matematik bölümüne kayıt olana kadar bu bölüm tabloda var olamaz. Nedeni, **Danışman** niteliğinin bir belirleyici olduğu hâlde anahtar olmamasıdır. Çözüm: belirleyicileri anahtar yapmak.

ÖĞRENCİ_DANIŞMAN		
ÖğrNo	Danışman	
123	A. ERCAN	
123	M. AKINCI	
456	K. SÖNMEZ	
789	A. ERCAN	
999	B. ÖZKAN	

DANIŞMAN_BÖLÜM	
Danışman	Bölüm
A. ERCAN	Fizik
M. AKINCI	Kimya
K. SÖNMEZ	Biyoloji
B. ÖZKAN	Kimya

Dördüncü Normal Form (4NF)

Bazı durumlarda BCNF'deki bir tabloda da sorun görülebilir. ÖĞRENCİ (ÖğrNo, Bölüm, Spor) tablosunda bir öğrenci birden çok bölüme kayıt olabilmekte ve birden çok spor etkinliğine katılabilmektedir. Bölüm ile Spor birbirinden bağımsızdır; ÖğrNo ile Bölüm ve ÖğrNo ile Spor arasındaki ilişkiler işlevsel bağımlılık değil, **çok değerli bağımlılıktır** (ÖğrNo → → Bölüm, ÖğrNo → → Spor).

ÖĞRENCİ		
ÖğrNo	Bölüm	Spor
123	Fizik	Kayak
123	Kimya	Kayak
123	Fizik	Tenis
123	Kimya	Tenis
999	Kimya	Tenis

ÖĞRENCİ_BÖLÜM	
ÖğrNo	Bölüm
123	Fizik
123	Kimya
999	Kimya

ÖĞRENCİ_SPOR	
ÖğrNo	Spor
123	Kayak
123	Tenis
999	Tenis

123 numaralı öğrencinin bir bölüme daha kayıt olması ya da bir spora daha katılması hâlinde tabloya iki kayıt birden eklenmesi gerekir. Bu yinelemeyi ortadan kaldırmak için tablo ikiye ayrılır ve **dördüncü normal form** elde edilir.

Form	Koşul
BCNF	Tablo 3NF'de ise ve her belirleyici bir aday anahtar ise
4NF	Tablo BCNF'de ise ve birbirinden bağımsız çok değerli bağımlılıklar ayrı tablolara ayrılmışsa

4.8 Çözümlü Sorular

ÖRNEK 1

OKUL (Ö.No, Ö.Ad, Ö.Soyad, Ders_No, Ders_Adı, Vize, Final, H.No, H.Ad, H.Soyad) tablosu 1NF'dedir (değerler atomik). Birincil anahtar (Ö.No, Ders_No) ikilisidir; H.No dersi veren hocanın numarasıdır. Tabloyu 3NF'ye getiriniz.

OKUL									
Ö.No	Ö.Ad	Ö.Soyad	Ders_No	Ders_Adı	Vize	Final	H.No	H.Ad	H.Soyad
2001001	Ahmet	Solmaz	202	Matematik 2	70	60	11	Özlem	UÇAR
2001001	Ahmet	Solmaz	203	Fizik 2	80	40	11	Özlem	UÇAR
2001005	Seyhan	Gülmez	202	Matematik 2	80	95	11	Özlem	UÇAR
2001002	Selim	Solmaz	702	Veri Tabanı Yönetimi	60	50	6	Altan	MESUT
2001003	Ahmet	Vardar	702	Veri Tabanı Yönetimi	60	60	6	Altan	MESUT

ÇÖZÜM

2NF: Kısmi bağımlılıklar kaldırılır. Öğrenci adı yalnızca Ö.No'ya; ders adı ve hoca bilgileri yalnızca Ders_No'ya bağlıdır. Vize ve Final ise ikisine birden bağlıdır.

NOTLAR (Ö.No, Ders_No, Vize, Final)

ÖĞRENCİLER (Ö.No, Ö.Ad, Ö.Soyad)

DERSLER (Ders_No, Ders_Adı, H.No, H.Ad, H.Soyad)

3NF: DERSLER tablosunda Ders_No → H.No → H.Ad, H.Soyad zinciri geçişli bağımlılıktır. Hoca bilgileri ayrılır.

NOTLAR (Ö.No, Ders_No, Vize, Final)

ÖĞRENCİLER (Ö.No, Ö.Ad, Ö.Soyad)

DERSLER (Ders_No, Ders_Adı, H.No)

HOCALAR (H.No, H.Ad, H.Soyad)

ÖRNEK 2

Bir hastanenin randevu tablosu aşağıdaki gibidir. Tabloyu 3NF'ye getiriniz.

RANDEVU

RanID	HastaID	HastaAd	HastaTel	DoktorID	DoktorAd	Uzmanlik	PolKlinik	PolKat	Tarih	Saat
R01	H01	Mehmet Ak	532-111	D01	Dr. Yılmaz	Kardiyoloji	Kardiyo. Pol.	3. Kat	10.03.2024	09:00
R02	H02	Fatma Şen	533-222	D02	Dr. Demir	Ortopedi	Ortopedi Pol.	2. Kat	10.03.2024	10:00
R03	H01	Mehmet Ak	532-111	D01	Dr. Yılmaz	Kardiyoloji	Kardiyo. Pol.	3. Kat	15.03.2024	11:00
R04	H03	Ali Çelik	534-333	D03	Dr. Can	Nöroloji	Nöroloji Pol.	4. Kat	11.03.2024	09:30

ÇÖZÜM

1NF: Her hücre atomik değer taşıyor; tablo zaten 1NF'dedir.



Hastane randevu sistemi: hasta, doktor, poliklinik ve randevu bilgileri

2NF: Birincil anahtar tek sütundur (RanID). Tek sütunlu anahtarda kısmi bağımlılık olamaz; tablo 2NF'dedir.

3NF: Veri tekrarının nedeni geçişli bağımlılıklardır:

RanID → **HastaID, DoktorID, Tarih, Saat** — randevunun kendi verisi ✓

RanID → **HastaID** → **HastaAd, HastaTel** — hasta bilgisi ✗

RanID → **DoktorID** → **DoktorAd, Uzmanlik** — doktor bilgisi ✗

DoktorID → **Poliklinik** → **PolKlinik, PolKat** — poliklinik bilgisi ✗

Hasta (HastaID, HastaAd, HastaTel)

Poliklinik (PolID, PolKlinik, PolKat)

Doktor (DoktorID, DoktorAd, Uzmanlik, PolID)

Randevu (RanID, HastaID, DoktorID, Tarih, Saat)

Bir büyük tablo dört tabloya ayrıldı. Hastanın telefonu değişirse tek satır güncellenir; poliklinik başka kata taşınırsa yine tek satır değişir.