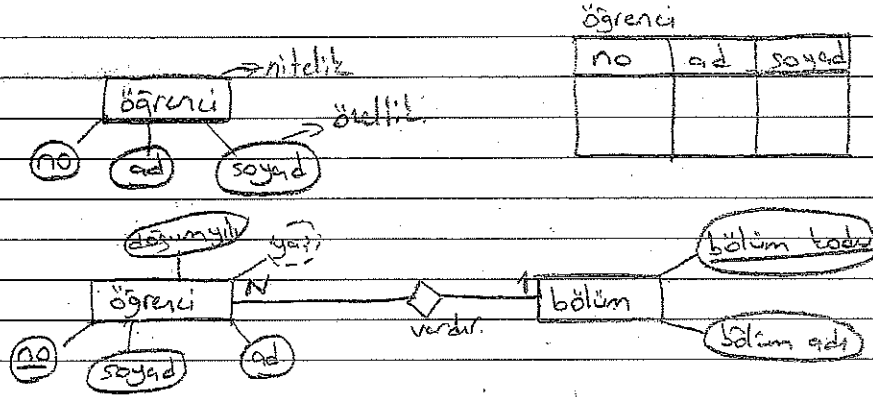


VERİTABANI

HELİN

~ ÖRNEKLER ~

1-)



* Öğrenci ile bölüm arasında ilişki vardır.

* Öğrencinin bir bölümü vardır.

NOT

Eğer N:M ilişkisi olursa tablo şu şekilde olmalıdır.

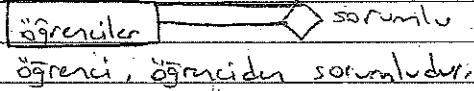
öğrenci

yas	ad	ad	soyad

bölüm kodu	bölüm adı

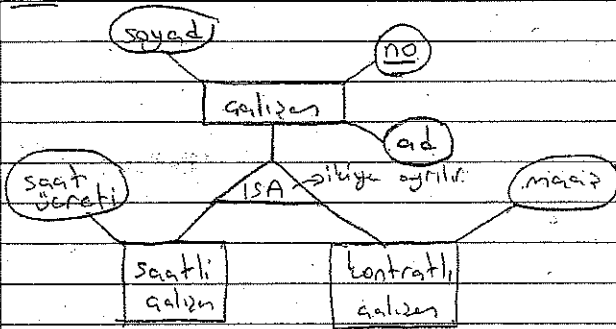
no	bölüm kodu

NOT



öğrenci, öğrenciden sorumludur.

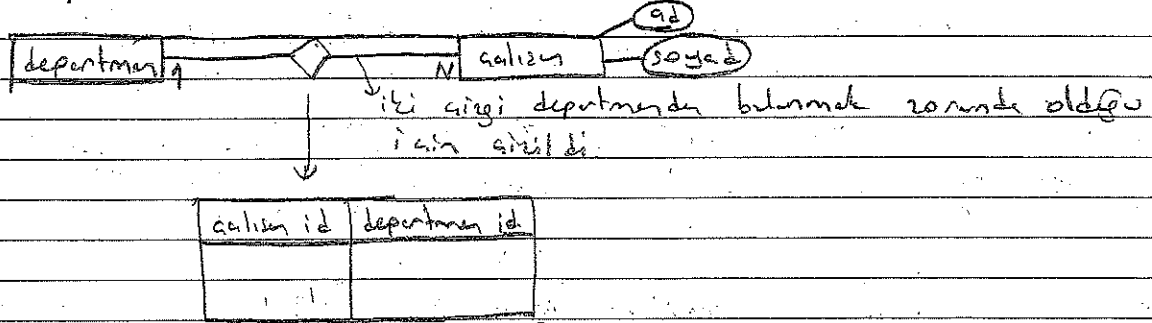
NOT



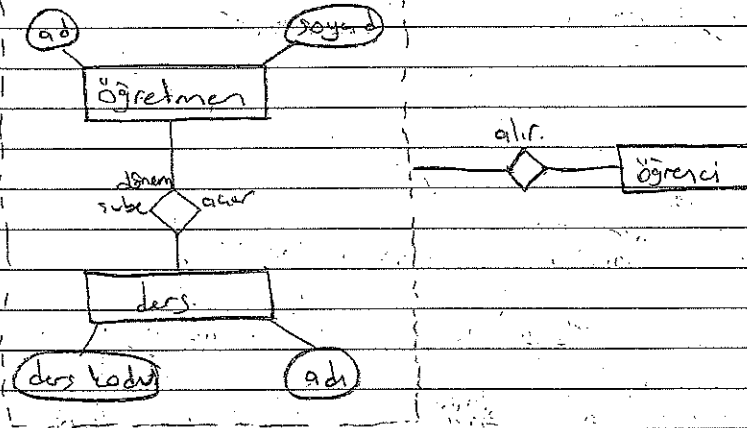
Yandaki şekil için üç tablo girilir. Saatli çalışan, kontratlı çalışan, galiyenin no'su verilir.

Örnek 1

- * 1 departmanda 1'den fazla çalışan
- * 1 çalışan ancak ve ancak bir departmana sahip
- * 1 departmanda en az 1 çalışan bulunur

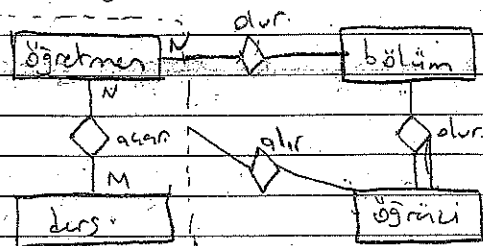


Örnek 1



Örnek 1

- * 1 öğretmen birden fazla ders verebilir.
- * 1 ders birden fazla öğretmen tarafından verilebilir.
- * Bölümde ders vardır, verilen ders öğrenci alır.
- * Öğrencilerin ve öğretmenlerin bölümleri vardır.
- * Bütün öğrenciler en az bir bölüme kayıtlı olmalıdır.



Örneği

Mağazada satılan her ürünün ürün no, markası, modeli, özellikleri, birim fiyatı, garanti süresi, stok miktarı tutulmaktadır.

* Mağazada satılan her ürün belli bir ürün grubundadır.

* Bir ürün grubunda birden fazla ürün bulunabilir.

* Mağazada satılmak üzere beyaz eşya firmalarından ürün satın alınır.

* Ürün satılan beyaz eşyanın adı, adresi, telefon, firma no belirlenir.

* 1 firmadan 1'den fazla ürün satın alınabilir.

* 1 üründe 1'den fazla farklı firmalardan satın alınabilir.

* Ürünün alım fiyatı, alım adedi, alım tarihi belirlenmektedir.

* Mağazadan satın alan müşteri bilgileri tutulmaktadır.

* Müşterinin kimlik no, ad, soyad, adres, telefonu belirlenir.

* Müşteri ürün aldığı anda fatura verilir.

* Faturadan fatura adı ve fatura bilgileri bulunmaktadırlar.

* 1 müşteri için birden çok fatura düzenlenir.

* 1 faturadan 1'den çok ürün bulunabilir.

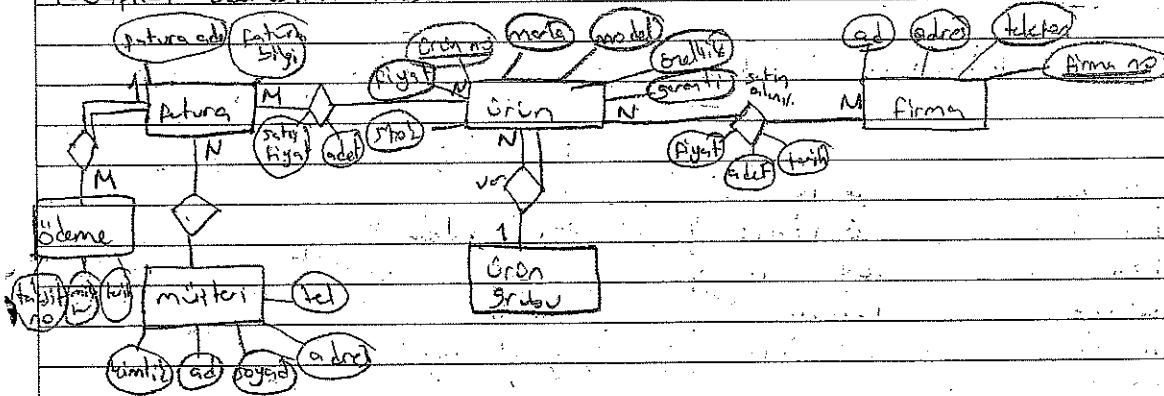
* 1 üründe 1'den çok fatura bulunabilir.

* Her ürünün satış fiyatı ve adedi belirlenir.

* Faturadaki ürün ödemesi peşin veya taksit yapılabilir.

* Bu durumda faturanın birden çok ödemesi olur.

* Yapılan ödemenin taksit numarası miktarı ve tarihi bulunur.



Ödevi

Proje amacı = Bir kombi servisi kombi bakım ve onarımını yaptığı müşterilerin bilgilerini veritabanında tutmak istiyor.

1-) Serviste çalıştığı personelin sicil no, ad, soyad, telefon ve adres bilgileri tutulmaktadır.

2-) Müşterilerin müşteri no, adı, soyadı, adresi ve telefon bilgileri bulunmaktadır.

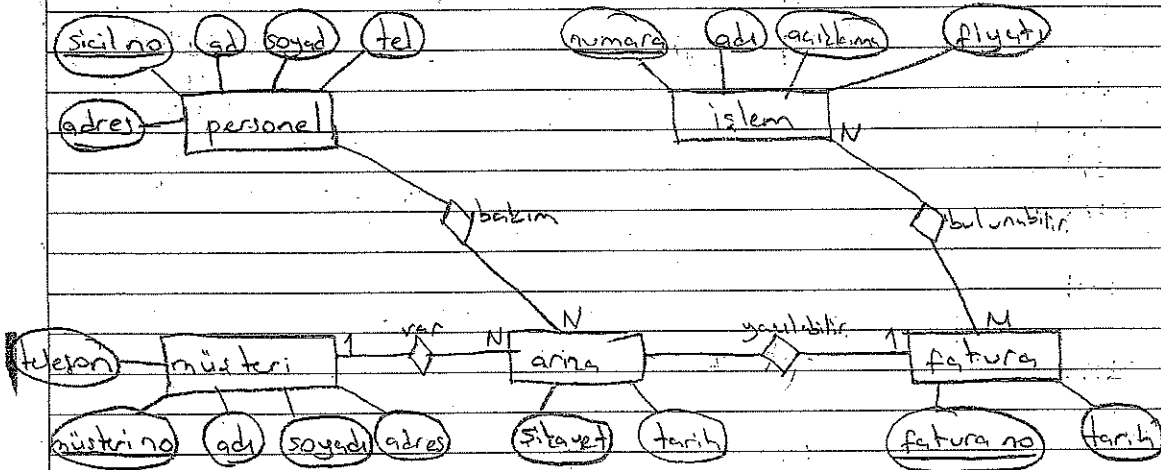
3-) Müşterinin kombisiyle arıza şikayeti almaktadır.

4-) Arıza bilgilerin yanında arıza talebin tarihi ve şikayet bilgileri vardır.

5-) Bir müşteri kombisinin 1'den çok arızası olabilir.

6-) Bir arızayı giderebilmek için servis elemanı müşteriye gönderilir.

- 7) Bir servis elemanı 1'den çok arızaya bakabilir.
 - 8) Bakım yapıldıktan sonra bakım ile ilgili fatura düzenlenir. Fatura sadece fatura no ve tarih bilgileri bulunmazdır.
 - 9) 1 arızaya 1 fatura yazılabilir.
 - 10) Faturada arızayı gidermek için yapılan her işlemin bir numarası, adı, ağırlığı ve işlem fiyatı bulunmalıdır.
 - 11) 1 faturada 1'den çok işlem bulunabilir.
 - 12) 1 işleminde 1'den çok fatura bulunabilir.
- Bu bilgilere göre müşteri firmasının İAR servisini ve tablalarını oluşturunuz.



~ İLİŞKİSEL CEBİR ~

P:		Q:		Birişim	
Id		Id		PUQ:	
İsim		İsim		İsim	
101	James	103	Smith	101	James
103	Smith	104	Lalende	103	Smith
104	Lalende	106	Byran	104	Lalende
107	Evan	110	Drew	106	Byran
110	Drew			107	Evan
112	Smith			110	Drew
				112	Smith
					(personel)

Fark		Kesişim	
P-Q:		P∩Q	
Id		Id	
İsim		İsim	
101	James	103	Smith
107	Evan	104	Lalende
112	Smith	110	Drew

Seçme (Selection) (σ)

Id < 105 (personel)

Id	Isim
101	James
103	Smith
104	Lalende

Proje (projection) (π)

TIsim (personel)

Isim
James
Smith
Lalende
Byran
Evan
Drew

* Birleştirme isleminde

nikillilerden en az bir-
nin ortak olması gerekir

Birleştirme (join): (⋈)

Personel-1

Maas:

Personel-1 ⋈ Maas

Id	Isim	Id	Maas	Id	Isim	Maas
101	James	101	67	101	James	67
103	Smith	103	55	103	Smith	55
104	Lalende	104	75	104	Lalende	75
107	Evan	107	80	107	Evan	80

Bölme: (÷)

P:

Q:

A	B	B	A
a1	b1	b1	a1
a1	b2	b2	a5
a2	b1		
a3	b1	B	A
a4	b2	b1	a1
a5	b1		a2
a5	b2		a3
			a5
		B	A
		b1	
		b2	
		b3	
		B	A
			a1
			a2
			a3
			a4
			a5

Örnek:



Personel:		Proje:			Atama:	
Id	İsim	Proje No	Proje-Adı	Sev	Proje No	Id
101	Jones	231	Pascal	102	453	101
103	Smith	278	Personel Verisi	110	354	103
104	Kalende	353	Veritabanı	102	343	104
106	Byran	354	İşletim Sistemi	104	354	106
107	Evren	453	Veritabanı	101	231	106
110	Drew				278	106
112	Smith				353	106
					354	106
					453	106
					231	107
					353	107
					278	110
					353	112
					354	112

- Soru 1: 353 nolu projede çalışan personelin Id'sini bulun.
- Soru 2: 353 nolu projede çalışan personelin Id'si ve İsmi bulun.
- Soru 3: Veritabanı projesinde çalışan personelin Id ve İsmi bulun.
- Soru 4: 353 ve 354 nolu projelerin birliğinde çalışan personelin Id ve İsmi bulun.
- Soru 5:

Cevap 1: $\Pi_{Id} (\sigma_{projeNo = 353} (Atama))$

Id
106
107
112

Cevap 2: $Personel \bowtie \Pi_{Id} (\sigma_{projeNo = 353} (Atama))$

Id	İsim
106	Byran
107	Evren
112	Smith

Cevap 3:

Personel \bowtie Π_{Id} (Atama \bowtie $\Pi_{projeno}$ ($\Delta_{projeno} = Veritabanı(projeno)$))

Id	İsim
101	Jane
106	Byron
107	Eve
112	Smith

353

353

Cevap 4:

Personel \bowtie (Atama \div $\Pi_{projeno}$ ($\Delta_{projeno} = 353 \vee Projeno = 354$ (A

Id	İsim
106	Byron
112	Smith

projeno
353
354

ER Şeması :

Soru \Rightarrow Yayınevi ilişkisinde yayınevinin numaraları, adları, ülkeleri ve adresleri bulunmaktadır.

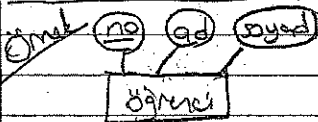
Yayınevi ilişkisinde kitap yazarlarının no, ad, ülke ve adres verileri bulunmaktadır.

Kitap ilişkisinde her kitabın ISBN numarası, adı, dili ve yayınevinin numarası yer almaktadır.

KTYAZ ilişkisinde her kitabın hangi yazarlar tarafından yapıldığını ve kitabın yazarlarının

~ SQL ~

Tablo Oluşturma:



Yandaki ER şemasının tablosu :

ogrenci

no	ad	soyad	aldigi ders kodu

CREATE TABLE ogrenci (

no int,

ad varchar(20),

soyad varchar(20),

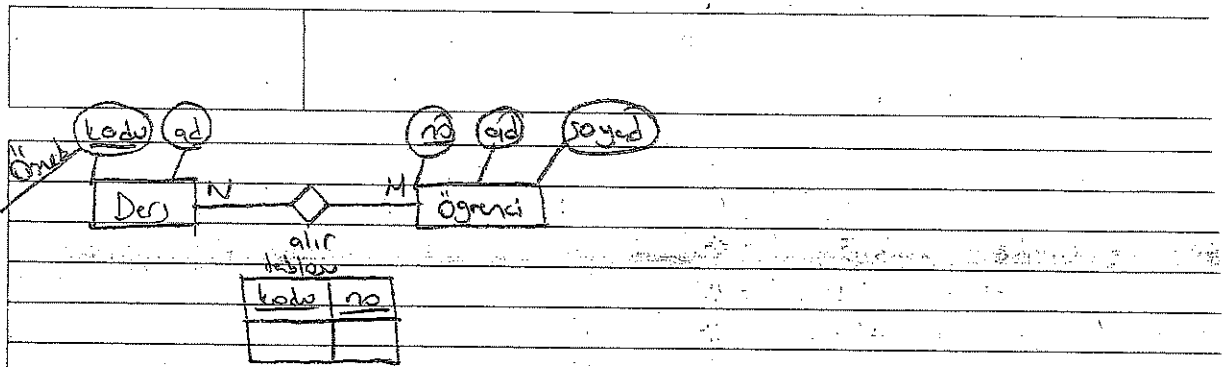
primary key (no)

aldigiderskodu char(25),

FOREIGN KEY aldigiderskodu REFERENCES Ders(kodu))

no int primary key

(tek primary key
olduğu için böylede
yazılabilir)



```

CREATE TABLE Ders (
  kod char(5) primary key,
  ad varchar(20)
)
  
```

```

CREATE TABLE Alır (
  kod char(5),
  no int,
  PRIMARY KEY(kod, no)
)
  
```

```

FOREIGN KEY kod REFERENCES Ders(kod),
FOREIGN KEY no REFERENCES Öğrenci(no)
  
```

Örnek

Öğrenci

no	ad	soyad

Eğer ad boş girilmemeli deniliyorsa şöyle yapılır.

```

CREATE TABLE Öğrenci (
  ad varchar(20) NOT NULL,
  ad varchar(20) UNIQUE
)
  
```

UNIQUE ⇒ aynı adden iki tane olamaz demektir.

NOT

* char ⇒ mutlak uzunluğu belirtir.

varchar ⇒ maksimum uzunluğu belirtir.

* Eğer 1'e 1'e ilişkisi varsa olduğu ders kodu UNIQUE yapılır.

Veri Ekleme

Örnek Öğrenci

no	ad	soyad	ders kodu
128	Helin	Cirit	BMÜ-318

```

INSERT INTO Öğrenci (no, ad, soyad, ders kodu)
VALUES (128, 'Helin', 'Cirit', 'BMÜ318')
  
```

--	--

Örnek

Çalışanlar					
NO	Ad	Soyad	Maas	Departman	
150	Oran	Kara	15000	Sist.	
151	Ufuk	Demir	20000	Sist	
152	Hakan	Yetiş	4050	Asist	
185	Osman	Top	300	Stajyer	
1	Hikmet	Tombul	65000	Patron	

ad,soyad yanlıyda ad ve soyad sütununu alırdı.

SELECT * Çalışanlar

SELECT DISTINCT Departman FROM Çalışanlar

→ tekrar eden yapılar bir defa gösterilir.

* Tablodan yıllık maaşları seçmek istiyorsak,

SELECT Ad, Maas*12 FROM Çalışanlar

* İsmi Oran olanın maaşının yıllık miktarını getir derseniz,

SELECT Maas*12 FROM Çalışanlar WHERE ad='Oran'

* Maaşı 5000'den fazla olanın Ad'ı ve yıllık maaşını getir derseniz,

SELECT Ad, Maas*12 FROM Çalışanlar WHERE Maas > 5000

* Maaşı olmayanın adını getir derseniz,

SELECT Ad FROM Çalışanlar WHERE Is NULL

* Maaşı boş olmayanı getir derseniz,

SELECT Ad FROM Çalışanlar WHERE Is NOT NULL

* Departmanı sist olan ve maaşı 17000'den yukarı olanların listesini getir.

SELECT * FROM Çalışanlar WHERE Departman='sist' AND Maas > 17000

* Departmanı sist veya stajyer olan ve maaşı 17000'den yukarı olanların listesini getir.

SELECT * FROM Çalışanlar (WHERE Departman='sist' OR Departman='stajyer' AND Maas > 17000)

* Maaşı 15000 ile 25000 arasında olan

SELECT * FROM Çalışanlar WHERE Maas BETWEEN 15000 AND 25000

* Departman **IN** ('sist', 'asist', 'stajyer')

veya anlamı katar

* Soyismi T ile başlayanları getirir dersen,

SELECT * FROM Calisanlar WHERE Soyad = 'T%'

* = '%T%' → içinde T geçenler

* = '%T' → T ile başlayanlar

* = '---T%' → 2. harfi T olsun

* SELECT * FROM Calisanlar ORDER BY Departman, Maaş
Önce departman göre sıralar (kocanın boyuna göre)

* Calisanların ad, soyad, maaş bilgisini ad, soyad sırasına göre getiren kod;

SELECT Ad, soyad, Maaş FROM Calisanlar WHERE ORDER BY
Ad, Soyad

* Tabloda kaç eleman olduğunu alır.

SELECT COUNT(*) FROM Calisanlar

↓
5 getirir.

* SELECT COUNT(*) AS ToplamCalisan FROM Calisanlar

Çıktı →

ToplamCalisan
5

* SELECT COUNT(DISTINCT Departman) AS ToplamFarkliDep FROM Calisanlar
Kaç farklı departman olduğunu belirtir.

* SELECT MIN(Maaş) AS minimummaaş FROM Calisanlar

↳ En az maaşın bulunduğu bilgisi getirir.

MAX → maksimum getirir.

AVG → ortalamayı verir.

SUM → toplam maaşı verir.

Departman	ToplamMaaş
Patron	
Asist	
Stajyer	
Sist	

Yandaki tabloyu isteseydik,

SELECT Depart SUM(maaş) AS ToplamMaaş FROM Calisanlar GROUP BY
Departman

* WHERE ToplamMaaş > 30000 yazılmaz. Bunun yerine
HAVING ToplamMaaş > 30000 yazılır.

* Birde cdk calisanı olan departman kodu, departmanındaki calisan eleman sayisi, departmandaki toplam maaşı sırayla getirir.

Departman	Toplam Calisan	Toplam Maaş
sist	2	65000

```
SELECT Departman Count(no) AS ToplamCalisan, Sum(Maaş) AS
Toplam Maaş FROM Calisanlar GROUP BY Departman HAVING
Count(No) > 1 ORDER BY Departman
```

* Maaşların ortalamasını bul. Hangi calisan bu ortalamağa ne kadar yakın veya ne kadar uzaktır.

```
SELECT Maaş - 1000 AS ebisimaaş FROM Calisanlar
```

↓

```
SELECT AVG(maaş) FROM Calisanlar
```

Ebisimaaş
14000
13000
3050
44000

LIKE KOMUTU :

* İsmi A ile başlayan soyadı B ile biten kimsede sorgu yapılır.
SELECT * FROM Calisanlar WHERE ad LIKE 'A*';

INNER-JOIN KOMUTU :

* İki farklı tablodaki kayıtları eşleştirmek amacıyla INNER JOIN kull

* Tartışma bölümünde sorulan soruları ve bu sorulara verilen cevapları listelemek için :

```
SELECT tartışma.soru.soru, tartışma.cavap.cavap FROM tartışma.soru
INNER JOIN tartışma.cavap ON tartışma.soru.id = tartışma.cavap.no
```

Microsoft SQL Server Management Studio

matlasyonitim.com/vtys

+ ÜRÜN (Ürün ID, Ürün adı, satış fiyat, alış fiyat, firma ID, Kategori ID)

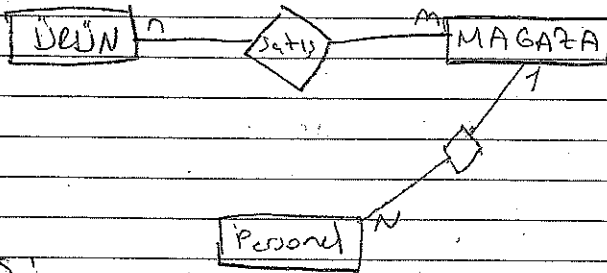
+ FİRMA (Firma ID, Adı, Firma adı, Tel)

+ KATEGORİ (KDV oranı, kategori adı, ID)

SİPARİŞ (ID, Tarih, Müşteri ID) ?

GERALDİĞİ (Sipariş ID, Ürün ID, Adet)

+ MÜŞTERİ (ID, Adres, Adı Soyadı, Tel)



DEFS:

İstisnai Bağımlılık!

X	Y
a	b
a	b

$X \rightarrow Y$ istisnai bağımlılık denir.

Ölçü: 0 anında verilen kapsayıcı türü olmaktadır.

Örnek 4.5 R(A,B,C,D) ilişkisi sonucu - -

$D \rightarrow C$ ilişkisi var Ama $C \rightarrow D$ ilişkisi yoktur

C'de x D'de e 'ye gidiyor. İlişiminde olduğu için istisnai bağımlıdır. b 'y'ye gidiyor. Tek türü olduğu için türü bozuyor. $AB \rightarrow CD$ istisnai bağımlıdır.

$B \rightarrow C$ ölçü ilişkisini bozan a hem x'e hem y'ye gittiği için.

İşlevsel Bağımlılık Türleri:

Kısmi işlevsel: Gerekli nitelik var,

Örnek 4.6

ÖADI \rightarrow gerekirse bir ifadedir. ÖNO bilimsel BNO bilimsel Bütün bilgiye gerek yok

ÖNO, ÖKODU \rightarrow KAD ÖNO gerekir ifadedir

Tam işlevsel bağımlılık: Gerekli nitelik yok

Örnek 4.6

Gerekli nitelik alınınca tam işlevsel olur.

ÖNO \rightarrow ÖADI, BNO

Önemli İşlevsel Bağımlılık:

ÖNO, ÖADI \rightarrow ÖNO sağ taraf sol tarafın alt kısmı veya aynıysa önemli dir.

Önemli İşlevsel Bağımlılık:

Sağ taraf sol tarafın alt kısmı değil.

Genel İşlevsel Bağımlılık:

$X \rightarrow Y$

$Y \rightarrow Z$

$X \rightarrow Z$ is belirlenir.

İşlevsel Bağımlılıkla ilgili kimi Kurum, Önerme
Temel Kurallar:

1-) Dönüşlülük:

Sağ taraf sol tarafın alt kısmı veya eşitli olduğu için

$X \rightarrow Y$ dışbilim

$A \rightarrow A$

$AB \rightarrow A$

2) Antinomi:

$X \rightarrow Y$ veya $XZ \rightarrow Y$ yapılabilir.

Yasaklıdır.

3- Geçerlilik:

4- Birliklik:

5- Ayırıcılık:

$X \rightarrow Y$

Sol tarafta sadece değişken yazılabilir Ayırıcılık

6- Sıra geçmezlik:

$X \rightarrow D$
 $Y \rightarrow W$ } $XZ \rightarrow W$ olur

Örnek:

1- $DG \rightarrow E$ (artırma) $D \rightarrow E$ olur

2- $A \rightarrow E$ $A \rightarrow BD$ artırılabilir
 $A \rightarrow B$
 $A \rightarrow D$
 $D \rightarrow E$ } $A \rightarrow E$ olur

3- $AC \rightarrow EG$ $A \rightarrow BD$ } $A \rightarrow B$
 $BC \rightarrow EG$
 AD } $AC \rightarrow EG$ olur

Kanonic Örtü:

$F: \{$ $F_c: \{$ $F^+ \equiv F_c^+$
10 tane üye 4 tane üye F^+ 'den türeyenler F_c^+ 'den türeyenler ağıdır

Artık Algoritması:

Örnek 1.10: $R(A, B, C, D, E, G)$

$X \rightarrow Y$
 $F: A \rightarrow B \rightarrow T = ACDE$ $Y \notin T$ $B \notin T$ B burda olmadığından artık değil
 $A \rightarrow C \rightarrow T = ABDECG$ $C \in T$ $A \rightarrow C$ artırılabilir
 $A \rightarrow D \rightarrow T = ABE$ $D \notin T$ artık değil
 $A \rightarrow E$
 $G \rightarrow B$
 $G \rightarrow D$
 $BC \rightarrow E \rightarrow T = BE$ $E \notin T$ artık değil

$CG \rightarrow A \rightarrow T-CG-BDEA \quad A \in T \quad A \neq T$

$BDE \rightarrow A$

$BDE \rightarrow C$

$BDE \rightarrow G$

$f_c: A \rightarrow BDE$

$G \rightarrow BD$

$BC \rightarrow E$

$BDE \rightarrow ACG$

BETİM NİTELİKLERİ

BÜTÜNLÜK KISITLAMALARI VE İLİŞKİSEL TASARIM

1-) Bütünlük Kısıtlamaları:

Yanlış, eksik, birbirleriyle çelişen, tutarsız veri içermemesi isteri

2-) Alan Kısıtlamaları:

Her niteliğe bir alan ataması ve niteliğin alabileceği değerlerin bu alandaki değerlerle sınırlanması ile ilgilidir. 3'e ayrılır

Alan türleri, değer sınırları, boş değer

b-) Referans Kısıtlaması:

Bir ilişkiadaki gözlemlerin varlığının, bir başka ilişkiadaki belirli gözlemlerin varlığına bağımlı olmasıdır

ÖRNEK (Dino, Tarih) ^{primary}

Mağaza (Mia, Tarih) ^{primary}

SATIS (S.Dino, S.Mag)

foreign foreign

c-) Nitelik Arası Bağımlılıklar:

İşlevsel Bağımlılık:

R bir ilişki seması, X ve Y alt kümeleri olsun.

Eğer X nitelik değerleri aynı olan tüm gözlemlerde, Y nitelik değerlerinin de aynı olması gerçekleşirse X-Y'yi işlevsel belirler ya da Y-X'e işlevsel bağımlıdır denir.

Örneği:

TASIT (PLAKANO, MARKA, MODEL, YIL, AĞIRLIK, RENK) işlevsel bağımlılıklarını gösterin.

PLAKANO \rightarrow MARKA, MODEL, YIL, AĞIRLIK, RENK

MARKA, MODEL \rightarrow AĞIRLIK

marka ve modeli bildikten sonra ağırlık'a gerek kalmaz.

Örnek 4.5: $R(A, B, C, D)$ ilişkisi semasına göre oluşturulmuş aşağıdaki \rightarrow ilişkisi olgusu veriliyor:

A	B	C	D	
1	a	x	e	$D \rightarrow C$ ilişkisi var. Ama $C \rightarrow D$ ilişkisi yok.
2	a	y	b	$AB \rightarrow CD$ ilişkisi bulunabilir.
2	b	x	c	$B \rightarrow C$ olgu ilişkisini bolar a hem
3	c	x	c	x'e hem y'ye gittiği için
4	a	x	e	

1-) Kısmi İstisnai Bağımlılık:
Gereksiz nitelik vardır.

Örnek 4.6:

$R(\text{ÖNO}, \text{ÖADI}, \text{BNO}, \text{BADI}, \text{FAKNO}, \text{DKODU}, \text{DADI}, \text{KED}, \text{NOTU})$ kısmi istisnai niteliklerini bulun.

$\text{ÖNO}, \text{ÖADI} \rightarrow \text{BNO}$; BNO 'yu öğrenmek için ÖADI bilginine gerek yok.

$\text{ÖNO}, \text{DKODU} \rightarrow \text{KED}$

$\text{ÖNO}, \text{DKODU}, \text{DADI} \rightarrow \text{NOTU}$

2-) Tam İstisnai Bağımlılık:
Gereksiz nitelik yoktur.

Örnek 4.6:

$\text{ÖNO} \rightarrow \text{ÖADI}, \text{BNO}$

$\text{BNO} \rightarrow \text{FAKNO}$

$\text{ÖNO}, \text{DKODU} \rightarrow \text{NOTU}$

3-) Önemli İstisnai Bağımlılık:

Örnek 4.6:

$\text{DKODU} \rightarrow \text{DKODU}$ sağ taraf sol tarafın alt kümesi veya

$\text{ÖNO}, \text{ÖADI} \rightarrow \text{ÖADI}$ aynıysa önemlidir.

4-) Önemli İstisnai Bağımlılık:

Örnek 4.6:

$\text{ÖNO} \rightarrow \text{ÖADI}$ sağ taraf sol tarafın alt kümesi değil.

5-) Gerçekli İstisnai Bağımlılık:

Örnek 4.6:

$X \rightarrow Y$

$Y \rightarrow Z$

$X \rightarrow Z$ y'ye bağlıdır.

İstisnai Bağımlılıklarla İlgili Kimi Tanım, Önerme ve Algoritmalar

1-) Bir istisnai bağımlılık kümesinin kapanışı:

F 'deki istisnai bağımlılıklardan türetilen tüm küme ve gerçekte bağımlı olan içeren istisnai bağımlılık kümesine F 'nin kapanışı denir (F^+)

İstemsel Bağımlılıkların Türetme Kuralları :

Temel Kurallar (Armstrong aksiyamları)

1-) Dönüştürülebilirlik :

$$A \subseteq B$$

$$B \rightarrow A$$

2-) Artırma :

$$X \rightarrow Y$$

$$XZ \rightarrow Y$$

3-) Geçirilebilirlik :

$$X \rightarrow Y$$

$$Y \rightarrow Z$$

$$X \rightarrow Z$$

4-) Birleştirme :

$$X \rightarrow Y$$

$$X \rightarrow Z$$

$$X \rightarrow YZ$$

5-) Ayrıştırma :

$$X \rightarrow YZ$$

$$X \rightarrow Y$$

$$X \rightarrow Z$$

6-) Sırdaki Geçirilebilirlik :

$$X \rightarrow Y$$

$$XZ \rightarrow W$$

$$YZ \rightarrow W$$

Kanonik Özet :

Örnek 4.9

$R(A, B, C, D)$ nitelik kümesi üzerinde tanımlı aşağıdaki istemsel bağımlılık kümesi düşünelim

$$F: \begin{array}{l} B \rightarrow C \\ C \rightarrow A \end{array}$$

$$F_c: \begin{array}{l} B \rightarrow C \\ C \rightarrow A \end{array}$$

$$F^+ = F_c^+$$

$$BC \rightarrow A$$

$$B \rightarrow A$$

F 'den türeyenler F_c 'den türeyenlerle aynıdır.

Antikip Algoritması :

Örnek 4.10

$R(A, B, C, D, E, G)$ nitelik kümesi üzerinde tanımlı aşağıdaki istemsel bağımlılık kümesi verilmiş olsun.

F:

F: $A \rightarrow BCDE$

$G \rightarrow BD$

$BC \rightarrow E$

$CG \rightarrow A$

$BDE \rightarrow ACG$

Çözümleri:

$A \rightarrow B$ CDE \Rightarrow artık değil

$A \rightarrow C$ BDEACG $\Rightarrow A \rightarrow C$ artıktır.

$A \rightarrow D$ BE \Rightarrow artık değil

$A \rightarrow E$ BD \Rightarrow artık değil

$G \rightarrow B$ D \Rightarrow artık değil

$G \rightarrow D$ B \Rightarrow artık değil

7. Türetilirlik Algoritması

- Bir F islemsel bağımsızlık kümesi verildiğinde, bu kümedeki islemsel bağımlılıklardan $f: X \rightarrow Y$ islemsel bağımlılığın türetilip türetilmeyeceğini bulan bir algoritmadır.

Örnek:

$AC \rightarrow BDE$ islemsel bağımlılığı türetilenir. Açıklaması:

$AC \rightarrow B$

$AC \rightarrow D$

$AC \rightarrow E$

} sağ taraf birden sol nitelikten oluşan islemsel bağımlılık açıklanmalıdır.

Bir Nitelik Kümesinin Kapanması:

Örnek: F: $A \rightarrow BCDE$

$G \rightarrow BD$

$BC \rightarrow E$

$CG \rightarrow A$

$BDE \rightarrow ACG$

Yukarıdaki ifadeye göre kimi nitelik alt kümelerinin kapanışlarını veriniz.

$A^+ = ABCDEG$

$(AB)^+ = ABCDEG$

$B^+ = B$

$(BC)^+ = BCE$

$C^+ = C$

$G^+ = BDG$

İLİSKİ ANAHTARLARI :

Süper Anahtar \Rightarrow Eğer bir nitelik alt kümesi (K) ilişkideki tüm nitelikleri işlevsel belirliyorsa yani K 'nin kapama, R 'ye eşitse, bu nitelik alt kümesi ilişkinin süper anahtarıdır.

Anahtar \Rightarrow Eğer bir nitelik alt kümesi (K) ilişkideki tüm nitelikleri işlevsel belirliyorsa ($K^+ = R$ ise) ve de K 'nin hiçbir alt kümesi tüm nitelikleri belirlemediyse, K ilişkinin anahtarıdır.

Eğer K , R 'nin bir süper anahtarı ise ve K 'nin hiçbir öz alt kümesi R 'nin süper anahtarı değilse K , R 'nin anahtarıdır.

Her ilişki anahtarı bir nitelik kümesi olduğuna göre, her anahtarı her üst kümesi süper anahtardır.

Her anahtar aynı zamanda süper anahtardır.

Örnek 1 (4.12)

$R(A, B, C, D, E)$ ilişkisi şeması üzerinde tanımlı aşağıdaki işlevsel bağımlılık kümesi veriliyor.

$F = \{ A \rightarrow C, B \rightarrow D, D \rightarrow AE, CE \rightarrow B \}$ ilişkinin 4 anahtarı vardır.

$B^+ = \underline{BDAEC} \rightarrow$ anahtar
 R

$B, AB, ABC, DC, ADE, ABCDE$
ilişkinin süper anahtarlarıdır.

$D^+ = \underline{DAECB} \rightarrow$ anahtar
 R

$AE^+ = \underline{AECBD} \rightarrow$ anahtar
 R

$CE^+ = \underline{CEBDA} \rightarrow$ anahtar
 R

İLİSKİLER İÇİN NORMAL BİCİMLER :

Soruşur ilişkiler oluşturabilmek amacıyla ilişkiler için bir dizi normal biçim tanımlanmıştır.

1- Birinci Normal Biçim (1NF)

1NF olabilmesi için niteliklerin tek değerli yalın nitelikler olması, hiçbir niteliğin hiçbir değerinin bir dizi, bir matris ya da karmaşık bir değer olmaması gerekir.

Örnek 1

ÖĞRENCİ (ÖNO, ÖADI, DERS (DADI, NOTU))

\Rightarrow DERS niteliği yalın bir nitelik olmadığından 1NF değildir.

ÖĞRENCİ (ÖNO, ÖADI, DERS, DADI, NOTU)

\Rightarrow 1NF'dir.

2-) İkinci Normal Biçim (2NF)

Asıl Nitelik \Rightarrow İlişkili anahtarlarının en az birinde yer alan niteliklere asıl nitelik denir.

Asıl Olmayan Nitelik \Rightarrow İlişkili anahtarlarının hiçbirinde yer almayan niteliklere denir.

2NF \Rightarrow Bir ilişki 1NF ise ve asıl olmayan niteliklerden hiçbirini kısmi işlevsel bağımlı değilse 2NF'dir.

Örneği:

SATICI (ÜKODU, FNO, FADI, FADRESİ, SFIYATI)

\Rightarrow Tüm nitelikler yalın olduğun 1NF'dir.

\Rightarrow İlişkinin tek anahtarı ÜKODU, FNO'dur. Bu yüzden asıl nitelikler

FADI, FADRESİ, SFIYATI asıl olmayan niteliklerdir. FADI ve FADRESİ anahtara kısmi işlevsel bağımlı olduğu için 2NF değildir.

~~Örneği:~~

ÜKODU, FNO \rightarrow FADI \rightarrow kısmi

ÜKODU, FNO \rightarrow FADRESİ \rightarrow kısmi

ÜKODU, FNO \rightarrow SFIYATI \rightarrow tam

} 2 kısmi old.
2NF değildir

NOT

Eğer tek anahtar varsa hepsi tam işlevsel bağımlıdır denir.

3-) Üçüncü Normal Biçim (3NF)

Eğer bir ilişki 2NF ise ve asıl olmayan hiçbir nitelik hiçbir anahtara geçişli bağımlı değilse bu ilişki 3NF'dir.

Örneği:

TASIT (PLAKANO, MARKA, MODEL, YIL, AĞIRLIK, RENK)

\Rightarrow Tüm nitelikler yalın olduğun 1NF'dir.

\Rightarrow PLAKANO anahtardır. Bunun dışında hiçbir asıl olmayan nitelikler. Hepsi anahtara tam işlevsel bağımlıdır.

⇒ Tek anahtar old. için tüm işlevsel bağımlı olması söz konusu olamaz. Bu yüzden 2NF'dir.

⇒ MARKA, MODEL → AĞIRLIK işlevsel bağımlıdır. Bu nedenle 3NF değildir.

4-) BCNF Normal Biçimi !

Eğer bir ilişki 1NF ise ve tüm belirleyici ilişkinin anahtarı ise BCNF'dir.

BCNF olabilmesi için, önemli her işlevsel bağımlılığın sol tarafında yer alan her nitelik ilişkisel tüm nitelikleri belirlemesi, dolayısıyla ilişkinin anahtarı olması gerekir.

$B \rightarrow F$ ise F 'nin sol tarafın tamamı anahtar ise BCNF'dir.

Örnek:

4.12'deki $R(A, B, C, D, E)$ ilişkisinin anahtarı old. biliyoruz ve anahtarları B, D, AE, CE 'dir. Bu ilişkinin tüm nitelikleri asal nitelikler. Asal olmayan nitelik olmadığından 3NF'dir.

$A \rightarrow C$ işlevsel bağımlılığı vardır. A anahtar olmadığından BCNF değildir.

NOT

= Asal olmayan nitelik yoksa anahtar olmasını bozan durum olmadığından 2NF'dir. Aynı zamanda 3NF'dir. ~~Fakat BCNF değildir~~

Örnek: (4.13)

ÖĞRETERLER (ÖĞRNO, DKODU, DADI, NOTU) nitelikler arasında aşağıdaki işlevsel bağımlılıkların bulunduğu varsayalım.

$DKODU \rightarrow DADI$

$DADI \rightarrow DKODU$

$DKODU, ÖĞRNO \rightarrow NOTU$

$DADI, ÖĞRNO \rightarrow NOTU$

⇒ Bu ilişkinin anahtarları DKODU, ÖĞRNO DADI, ÖĞRNO 'dur.

Asal olmayan nitelik NOTU'dur.

NOTU anahtarlara tam bağımlıdır. (Asıl olmayan tek nitelik bulunursa geçici bağımlılık söz konusu değildir). 3NF'dir.

DKODU, ÖGRNO \rightarrow NOTU \rightarrow Tam } 2NF'dir
DADI, ÖGRNO \rightarrow NOTU \rightarrow Tam }

Ancak anahtar olmayan DKODU ve DADI birer belirleyici olduğu için ilişki BCNF değildir.

İLİŞKİLERİN AYRISTIRILMASI:

Ayrıştırma BCNF, 3NF uyumayan veritabanlarını BCNF ve 3NF yapma. Ayrıştırmanın geçerli bir ayrıştırma olması için:

- 1) Yitimsiz - birleştirme ayrıştırması olması;
 - 2) İstisnai bağımlılıktan korunması;
- genelli ve yeterlidir.

Yitimsiz - Birleştirme Ayrıştırması \Rightarrow Eğer bir ilişki BCNF değilse bir takım aykırılıklara yol açabileceğini gördük. Bu tür bir ilişkiyi belirli sayıda BCNF ilişkiye ayırarak aykırılıklara yol açmaması, oluşturulması ve bakımı kolay bir semaya elde edebiliriz. Bu işleme ilişkinin normalleştirilmesi denir.

İkili Bir Ayrıştırma İçin Yitimsizlik Kuralı \Rightarrow

Eğer R 'nin $\{R_1, R_2\}$ ikili ayrıştırması, eğer aşağıdaki koşullar sağlanırsa yitimsizdir.

* R_1 ve R_2 'de ortak nitelik ya da nitelikler bulunmalıdır.

\rightarrow Eğer $R(X, Y, Z)$ ise $R_1(X, Y)$ ve $R_2(X, Z)$ olmalıdır.

* R_1 ve R_2 'deki ortak nitelikler R_1 ve R_2 'den en az birinin anahtarı olmalıdır.

$\rightarrow X \rightarrow R_1$ ya da $X \rightarrow R_2$ istisnai bağımlılıklardan en az biri F^+ bulunmalıdır.

Örnek: 4.16

$R(A, B, C, D, E)$ ilişkisi seması ve nitelikler arası aşağıdaki istisnai bağımlılık kümesi verilmiştir.

$F: A \rightarrow BC$

$D \rightarrow B$

$E \rightarrow A$

$CD \rightarrow E$

Bu ilişki $R_1(A, B, C)$ ve $R_2(A, D, E)$ ilişkisine ayrıştırılıyor.
Bu ayrıştırma yitimsiz midir?

Cevap:

R_1 ve R_2 'de ortak nitelik A 'dır. A R_1 ilişkisinin anahtarı olduğu için yitimsizdir.

$A^+ \rightarrow ABC$ (R_1 'in anahtarı sırası)

Ayrıştırmanın Yitimsizlik Sınaması \Rightarrow A2 önceki konu yalnız belli ayrıştırmanın yitimsizliğini sınamak için kullanılabilir. Bir ilişki birden fazla ilişkiye ayrıştırıldığında, bu yöntem kullanılamaz.

Örnek:

$R(A, B, C, D, E)$ ilişki şeması ve nitelikler arası ağdaki ilişkisel bağımlılık kümesi veriliyor.

$F: A \rightarrow BC$ $R_1(A, B, C)$
 $D \rightarrow B$ $R_2(A, E)$
 $E \rightarrow A$ $R_3(C, D, E)$ } \rightarrow ayrıştırmanın yitimsiz olup olmadığını belirler.
 $CD \rightarrow E$

Cevap:

Gizliğin i. satır, j. kolonundaki elemanına:

\rightarrow Eğer R_i ilişkisinde A_j niteliği varsa : a_j yaz.

\rightarrow Eğer R_i ilişkisinde A_j niteliği yoksa : b_{ij} yaz.

	1	2	3	4	5
	A	B	C	D	E
R_1	a ₁	a ₂	a ₃	b ₁₄	b ₁₅
R_2	a ₁	b ₂₂ a ₂	b ₂₃ a ₃	b ₂₄	a ₅
R_3	b ₃₁ a ₁	b ₃₂ a ₂	a ₃	a ₄	a ₅

* $A \rightarrow BC$ kullanılarak, R_1 ve R_2 satırlarının A kolonundaki değerler eşit olduğu için, bu satırların B ve C kolonundaki değerlerde eşitlik (b₂₂ yerine a₂, b₂₃ yerine a₃ yazılır).

* $E \rightarrow A$ kullanılarak R_2 ve R_3 satırlarının E kolonundaki değerler eşit olduğu için bu satırların A kolonundaki değerlerde eşitlik (b₃₁=a₁ olur).

* $A \rightarrow BC$ kullanılarak, R_1 ve R_3 satırlarının A kolonundaki değerler eşit olduğu için B kolonundaki değerlerde eşitlik (b₃₂=a₂ olur). C aynı oldu. İtina değeriyle yapılmaz - D.

R_3 satırı tüm a 'lardan oluştuğu için yitmemektedir.

(Çinleme değeriyle olduğu sürece, satırlardan biri tüm a 'lar
den oluşuncaya kadar 3. adımı tekrarlama)

Ayrıştırmanın İzlenel Bağımlılıkları Korunması
Örnek (4.16)

$R(A, B, C, D, E)$ ilişkisi semesi ve nitelikler arası aşağıdaki izlenel
bağımlılık kasesi veriliyor.

$F: A \rightarrow BC$	$R_1(A, B, C)$	Bu ayrıştırmanın izlenel bağımlılıkları korunup korunma- dığını araştıracağız.
$D \rightarrow B$	$R_2(A, E)$	
$E \rightarrow A$	$R_3(C, D, E)$	
$CD \rightarrow E$		

Çözümü:

$F^+:$ $A \rightarrow BC$
 $D \rightarrow B$
 $E \rightarrow ABC$
 $CD \rightarrow AE$
 ~~$AE \rightarrow E$~~

F^+ 'lerin R_i 'lere göre indirgenimini ve bunların kasma birleşimini bulalım.

$R_1(A, B, C)$ $F_1: A \rightarrow BC$

$R_2(A, E)$ $F_2: E \rightarrow A$

$R_3(C, D, E)$ $F_3: CD \rightarrow E$

$G = F_1 \cup F_2 \cup F_3 = \{A \rightarrow BC, E \rightarrow A, CD \rightarrow E\}$

F 'deki izlenel bağımlılıklardan biri ($D \rightarrow B$) G 'de yoktur ve G 'deki
izlenel bağımlılıklardan türetilemez. Bu nedenle yitmiştir olan bu
ayrıştırma izlenel bağımlılıkları korumamaktadır. Geçerli bir ayrış-
tırma değildir.

Örnek (4.17)

$R(A, B, C, D)$ ilişkisi semesi ve nitelikler arası izlenel bağımlılık kasesi
veriliyor.

$F: A \rightarrow B$	R ilişkisinin	$R_1(A, B)$	ayrıştırılabilir yitimsiz
$B \rightarrow C$		$R_2(B, C)$	bir ayrıştırma değildir.
$C \rightarrow D$		$R_3(C, D)$	işlevsel bağımlılıkları koruyor mu?
$D \rightarrow A$			

Cevap:

F^+ : $A \rightarrow BCD$
 $B \rightarrow CDA$
 $C \rightarrow DAB$
 $D \rightarrow ABC$

$R_1(A, B)$ $F_1: A \rightarrow B, B \rightarrow A$
 $R_2(B, C)$ $F_2: B \rightarrow C, C \rightarrow B$
 $R_3(C, D)$ $F_3: C \rightarrow D, D \rightarrow C$

$G = F_1 \cup F_2 \cup F_3$

$B \rightarrow A$

$C \rightarrow B$

$D \rightarrow C$

$D \rightarrow A$ üretiliyor. Bu yüzden G bütün işlevsel bağımlılıkları koruyor. Yitimsizlik verilmiş ve işlevsel bağımlılıkları koruyor.

BCNF Ayrıştırma Algoritması:

BCNF ayrıştırma algoritması ile elde edilen ayrıştırma yitimsiz bir ayrıştırma olur. Ancak ayrıştırmanın işlevsel bağımlılıkları koruma güvencesi yoktur.

Örnek: (4.18)

$R(A, B, C, D, E)$ ilişkisi semasi ve nitelikler arası aşağıdaki işlevsel bağımlılıklarla tanımlanıyor.

$F: A \rightarrow CDE$

$E \rightarrow B$

$CD \rightarrow E$

Cevap:

R ilişkisinin anahtarı A 'dır. ve BCNF değildir.

F^+ : $A \rightarrow BCDE$

$E \rightarrow B$

$CD \rightarrow BE$

~~İşlem aşaması~~

BCNF ayrıştırma:

- ① $E \rightarrow B$ $R_1(E, B)$ $F_1: E \rightarrow B$
② $CD \rightarrow E$ $R_2(C, D, E)$ $F_2: CD \rightarrow E$
③ $A \rightarrow CDE$ $R_3(A, C, D)$ $F_3: A \rightarrow CD$ ($A \rightarrow CD$
 $CD \rightarrow E$)

E, CD, A anahtar $A \rightarrow E$ oluşturdular.

~~İşlem aşaması~~

BCNF ayrıştırma:

$R_1(A, C, D)$

$R_2(C, D, E)$

$R_3(E, B)$

$$G = F_1 \cup F_2 \cup F_3$$

Bu ayrıştırma yitimsizdir. Ayrıca istenmiş bağımlılıklar korunur.

Örnek 1.18

$R(A, B, C, D, E, G)$ ilişkili seması ve nitelikler arası istenmiş bağımlılık kümesi veriliyor.

- $F: \textcircled{1} AB \rightarrow CD$ $R_1(D, E, G)$ $F_1: DE \rightarrow G$
② $AC \rightarrow E$ $R_2(A, C, E)$ $F_2: AC \rightarrow E$
③ $DE \rightarrow G$ $R_3(A, B, C, D)$ $F_3: AB \rightarrow CD$

bu işlem kaybolmaması için
seçiliyor.

~~İşlem aşaması~~

Örnek 1.19

$R(A, B, C)$ ilişkili seması ve nitelikler arası $F = \{AB \rightarrow C, C \rightarrow A\}$ istenmiş bağımlılık kümesi veriliyor.

~~Çözüm~~

Anahtarı AB, BC dir. 3NF'dir. BCNF değildir.

F^+ olup BCNF ayrıştırma algoritması uygulanır.

$F^+:$ $AB \rightarrow C$
 $C \rightarrow A$

BCNF ayrıştırma:

$R_1(B, C)$

$R_2(C, A)$

Bu ayrıştırma yitimsizdir. $AB \rightarrow C$ bağımlılığı yitirilmiştir. Yitimsizden
Ama istenmiş bağımlılıkları korumam.

3NF Ayrıştırma Algoritması: Tek bir bağımlılık olsun.
BCNF ayrıştırma algoritması ile her zaman istenil bağımlılıklar,
koruyan bir ayrıştırma bulmak mümkün değildir, 3NF ayrıştırma
algoritması ile her zaman istenil bağımlılıklar koruyan ay-
rıştırma bulmak mümkün.

Örnek: (4.20)

$R(A, B, C, D, E, G)$ ilişkisi semer ve niteliler arası istenil bağımlılık
kumesi veriliyor.

$F: AB \rightarrow CD$

$AC \rightarrow E$

$DE \rightarrow G$

İlişkinin tek anahtarı AB 'dir ve geçisti bağımlılıklar nedeniyle de
ilişkinin bütünü 2NF old. görür.

$(F_c = F)$ Kanıtı.

3NF ayrıştırma:

$R_1(A, B, C, D)$

$R_2(A, C, E)$

$R_3(D, E, G)$

R için BCNF ayrıştırması bulmak istediğimizde, F^+ y. hesaplayıp,
algoritmayı uygulayalım.

F^+ : $AB \rightarrow CDEG$

$AC \rightarrow E$

$DE \rightarrow G$

$ACD \rightarrow G$

BCNF Ayrıştırma:

1. Ayrıştırma

$R_1(A, B, C, D)$

$R_2(A, C, E)$

$R_3(D, E, G)$

2. Ayrıştırma

$R_1(A, B, C, D)$

$R_2(A, C, E)$

$R_3(A, B, G)$

1. si istenil bağımlılıklar korumaktadır. 2. si de $DE \rightarrow G$ istenil
bağımlılığı getirilmektedir.

--	--

~ HAREKET KAVRAMI ~

Bir banka hesabından diğerine para atılma, stok miktarı artırılma gibi işlemlerdir.

Hareketlerden birini veritabanı işleminin bir araya gelmesi ile oluşan program parçasıdır.

Kurtarılabılır İsketim Planı :

Çizim 9.10

P10

H8

H9

Read(A);

Write(A);

Read(A);

Commit;

Read(B);

Write(B);

Commit;

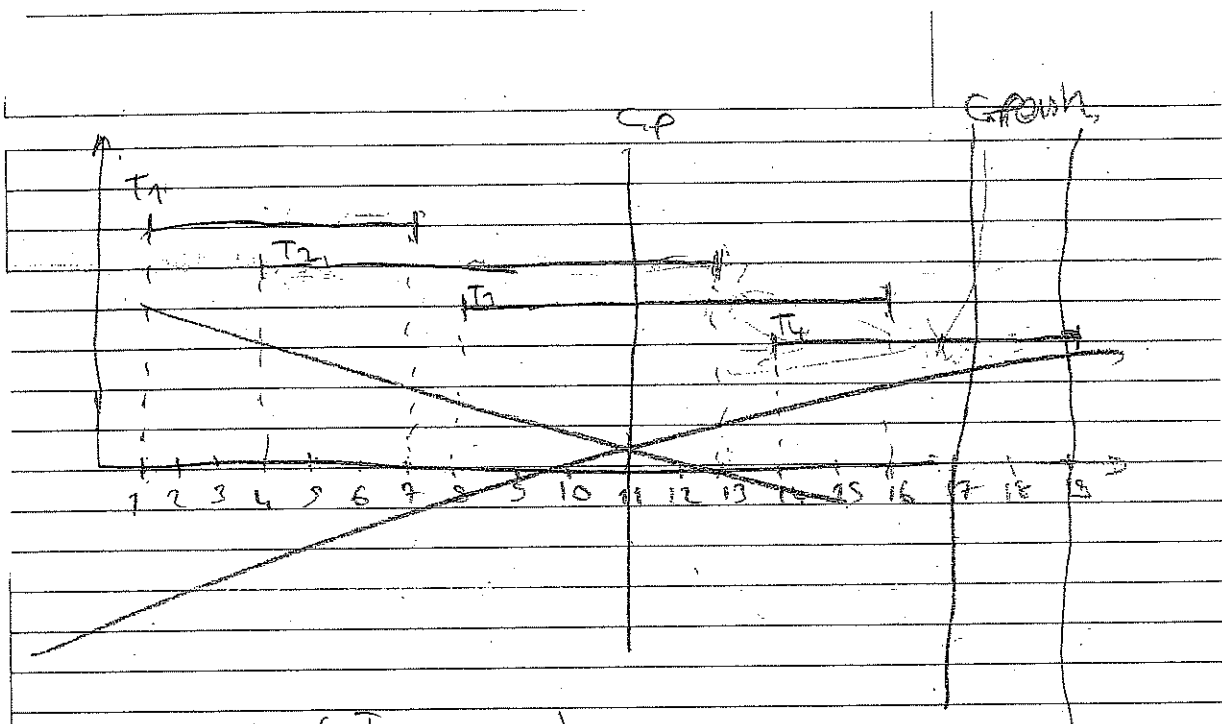
Eğer H9 hareketi Read(A) komutunun işletiminden hemen sonra sonlandırılırsa (commit işlemi uygulandıysa) sorunlar yaşanabilir. Çünkü H9 tarafından okunan A değeri, yazan H8 hareketi henüz tamamlanmamıştır.

Eğer H8 hareketi, örneğin Read(B) komutunun işletiminden hemen sonra başarısız olursa ve H8 için abort işlemi uygulanırsa, H8 işi kurtarma işlemi gerçekleştirilerek ve işletimi tamamlanmış komutların etkisi geriye alınacaktır. Bu durumda, H9 tarafından okunan değerde geçerliliğini yitirecektir, bu nedenle H9'da kurtarılabılır.

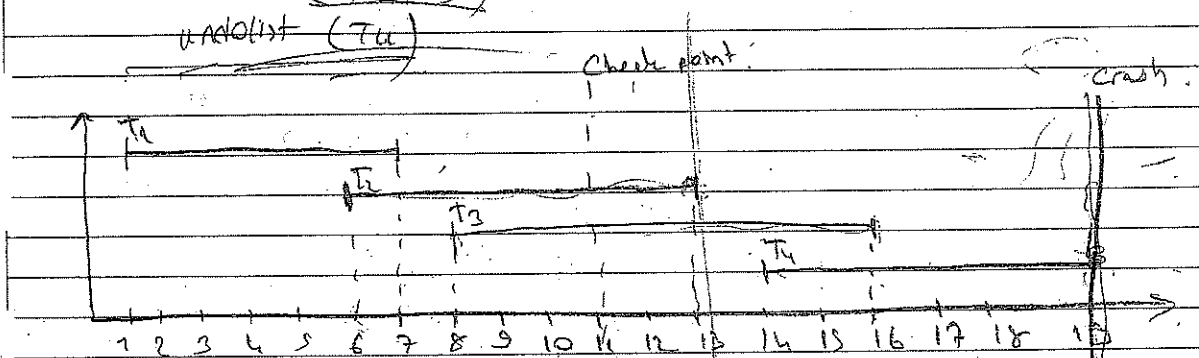
Tamamlanmış bir hareketin geriye alınmasının da işi karıştırmaz. Bu sorunun nedeni P10 planın kurtarılabılır bir işletim planı olmamasından kaynaklanmaktadır.

NOT

Kurtarma Stratejisi, tutarlılığa neden olan işlemleri geri almak (undo) dayanır. Bu arada veri tabanını tutarlı bir duruma getirmek için işlemlerden bir kısmını yeniden işletmek (redo) gerekebilir.



~~undo (T_2, T_3, T_4)~~
~~redo (T_2, T_3)~~
~~undolint (T_4)~~



$X=50$
 $Y=40$
 $Z=45$
 $U=60$
 $V=80$
 $W=100$

$X=50$
 $Y=40$
 $Z=45$
 $U=50$
 $V=50$

~~UNDO (T_2, T_3, T_4)~~
~~REDO (T_2, T_3)~~
~~UNDOLINT (T_4)~~