

Ayrıştırılabilirlik Yitimsizlik Sınaması

Bir ilişki ikiden fazla ilişkiye ayrıştırıldığında bu yöntem kullanılır.

ÖRNEK: $R(A, B, C, D, E)$

$F: A \rightarrow BC$
 $D \rightarrow B$
 $E \rightarrow AC$
 $CD \rightarrow E$

$R_1(A, B, C)$

$R_2(A, E)$

$R_3(C, D, E)$

} Yitimsiz
midir?

Çözüm: Çizelgenin i. satır. j. sütundaki elemana;

R_1 ilişkisinde A_j niteliği varsa $\rightarrow a_j$ yaz

R_1 ilişkisinde A_j " yoksa $\rightarrow b_{ij}$ yaz.

	1 A	2 B	3 C	4 D	5 E
R_1	a_1	a_2	a_3	b_{14}	b_{15}
R_2	a_1	b_{22}	b_{23}	b_{24}	a_5
R_3	b_{31}	b_{32}	a_3	a_4	a_5

$\rightarrow A \rightarrow BC$ kullanılarak R_1 ve R_2 'nin
A kolonundaki değerler eşit old. için
B ve C deki de eşitler.

$b_{22} = a_2$ $b_{23} = a_3$ olur.

\rightarrow Komple 1 satır a olduğunda bırak
yitimsizdir.

$\rightarrow E \rightarrow A$ kullanılarak R_2 ve R_3 E kolonundaki değerler eşit old için
A kolonu da eşitler $b_{31} = a_1$ olur.

$\rightarrow A \rightarrow BC$ kullanılarak tablonun son haline baktığımızda $b_{32} = a_2$ olur.
 R_3 satırı bütünüyle a olduğu için ayrıştırma yitimsiz

Ayrıştırılabilirlik İstisnai Bağımlılıkları Koruması

Her istisnai b. bir bütünlük kısıtlamasıdır. Veri tabanının bütünlüğünün ve tutarlılığının korunması için yapılan her güncelleme işleminde istisnai bağımlılıkların sağlandıklarının denetlenmesi gerekir. Bu denetimlerin her birinin, birden çok ilişkinin birleştirilmesini gerektirmeden tek bir ilişki üzerinde yapılabilmesi gerekir.

ÖRNEK: Sayfının yukarıdaki örneğin istisnai b. koruyup korumadığını bakalım

Çözüm: F^+ $A \rightarrow BC$
 $D \rightarrow B$
 $E \rightarrow AC$
 $CD \rightarrow E$

F^+ ların ilere göre indirgimlerini ve bunların küme birleşimini bulalım

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

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

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

$G = F_1 \cup F_2 \cup F_3$

$G = \{A \rightarrow BC, E \rightarrow A, CD \rightarrow E\}$

F'deki istisnai b. bir ($D \rightarrow B$) G'de yoktur ve G'deki ist. bağımlılıklardan türemez. Yitimsiz olan bu ayrıştırma istisnai bağımlılıkları korumaz
Geçerli bir ayrıştırma değildir.

ÖRNEK: $R = \text{KURS}(\text{ÖNO}, \text{OKODU}, \text{ÖĞRETMEN}, \text{NOTU})$ $F = \text{ÖĞRETMEN} \rightarrow \text{OKODU}$
 $\text{ÖNO}, \text{OKODU} \rightarrow \text{NOTU}$

$\text{ÖNO}, \text{ÖĞRETMEN} - \text{anahtar}$

$\rightarrow \text{INF}^+$

Bir öğretmen bir derse giriyor, OKODU anahtarı kısmi bağımlı olduğundan 2NF değil

2NF olmadığından 3NF değil.

İlişkilerin Ayrıştırılması

* Ayrıştırma BCNF ve 3NF olmayan veritabanlarını BCNF ve 3NF yapmaktır. Ayrıştırmanın geçerli bir ayrıştırma olması için

- \rightarrow Yitimsiz - birleştirme ayrıştırması
 - \rightarrow İşlevsel bağımlılıkları korunması
- } gerekli ve yeterlidir.

Yitimsiz - Birleştirme Ayrıştırması

\rightarrow İkili Bir Ayrıştırma için Yitimsizlik Koşulu

R 'nin $\{R_1, R_2\}$ ikili ayrıştırması eğer aşağıdaki koşullar sağlanıyorsa yitimsiz

1) R_1 ve R_2 de ortak nitelik ya da nitelikler bulunmalıdır.

$\hookrightarrow R(x, y, z)$ ise $R_1(x, y)$ $R_2(x, z)$ olması

2) R_1 ve R_2 'deki ortak nitelikler R_1 ve R_2 'den en az birinin anahtarı olması

$\hookrightarrow x \rightarrow R_1$ ya da $x \rightarrow R_2$ işl. b. en az biri F^+ da bulunmalıdır.

ÖRNEK: $R(A, B, C, D, E)$ $F: A \rightarrow BC$ $R_1(A, B, C)$ $R_2(A, D, E)$ \rightarrow Bu ayrıştırma yitimsiz midir?

$D \rightarrow B$
 $E \rightarrow A$
 $CD \rightarrow E$

$\rightarrow R_1$ ve R_2 de ortak nitelik A 'dır.

$A^+ = ABC \rightarrow R_1$ in anahtarı old. için yitimsizdir.

\hookrightarrow Bu kural sadece ikili ayrıştırmalar için geçerlidir.

2'den fazla ayrıştırma yapılacaksa bu yöntem kullanılamaz.

İşlevsel B. Türetme Kuralları

(2)

1) Dönüşlülük

$$Y \subseteq X \quad X \rightarrow Y$$

2) Arttırma

$$X \rightarrow Y \quad XZ \rightarrow Y$$

3) Geçirgenlik

$$\begin{matrix} X \rightarrow Y \\ Y \rightarrow Z \end{matrix} \Rightarrow X \rightarrow Z$$

Amstrong aksiyomları

4) Birleşim

$$\begin{matrix} X \rightarrow Y \\ X \rightarrow Z \end{matrix} \Rightarrow X \rightarrow YZ$$

5) Ayrılma

$$\begin{matrix} X \rightarrow YZ \\ X \rightarrow Y, X \rightarrow Z \end{matrix}$$

6) Sözcük geçirgenlik

$$\begin{matrix} X \rightarrow Y \\ YZ \rightarrow W \end{matrix} \Rightarrow XZ \rightarrow W$$

Kanonic Örtü

$R(A, B, C, D)$ kümesi üzerinde tanımlı aşağıdaki işl. b. düşünelim

$$F: B \rightarrow C$$

$$C \rightarrow A$$

$$BC \rightarrow A$$

$$B \rightarrow A$$

$$F_c: B \rightarrow C$$

$$C \rightarrow A$$

$$F^+ = F_c^+ \rightarrow F \text{ 'den türeyenler } F_c \text{ 'den türeyenlerle aynıdır.}$$

ARTIKLIK ALGORİTMASI

→ Bir kümedeki işlevsel b. bir değer işlevsel bağımlılıklardan türetiliyorsa bu işlevsel b. kümede artıktır.

ÖRNEK: $R(A, B, C, D)$
E, G

$$F: A \rightarrow BCDE$$

$$G \rightarrow BD$$

$$BC \rightarrow E$$

$$CG \rightarrow A$$

$$BDE \rightarrow ACG$$

buradan bakılıyor?

C bulunduğundan için artıktır

$$A \rightarrow B$$

$$G \rightarrow B$$

$$BDE \rightarrow A$$

$$A \rightarrow C$$

$$G \rightarrow D$$

$$BDE \rightarrow C$$

$$A \rightarrow D$$

$$BC \rightarrow E$$

$$BDE \rightarrow G$$

$$A \rightarrow E$$

$$CG \rightarrow A$$

$$* A \rightarrow B \text{ artıktır mı? } T = \{A, C, D, E\} \text{ hayır}$$

$$* A \rightarrow C \text{ artıktır mı? } T = \{A, B, D, E, A, C, G\} \text{ evet}$$

$$F = F - \{A \rightarrow C\} \Rightarrow \{A, B, D, E, G\}$$

AGC'ye gidiyor
oyutdan artıktır

$$* A \rightarrow D \text{ artıktır mı?}$$

$$T = \{A, B, E\} \text{ (C'yi çıkardık)}$$

Bütün
kümeyi
bililebiliriz

$$* BDE \rightarrow A \text{ artıktır mı? } T = \{B, D, E, G, C\} \times$$

$$* BDE \rightarrow C \text{ artıktır mı? } T = \{B, D, E, G, A\} \times$$

$$* BDE \rightarrow G \text{ artıktır mı? } T = \{B, D, E, A, C\} \times$$

$$F_c: A \rightarrow BDE$$

$$G \rightarrow BD$$

$$BC \rightarrow E$$

$$BDE \rightarrow ACG$$

$$* A \rightarrow E \text{ artıktır mı? } T = \{A, B, D\} \text{ hayır}$$

$$* G \rightarrow B \text{ artıktır mı? } T = \{G, D\} \text{ hayır}$$

$$* G \rightarrow D \text{ artıktır mı? } T = \{G, B\} \text{ hayır}$$

$$* BC \rightarrow E \text{ artıktır mı? } T = \{B, C\} \text{ hayır}$$

$$* CG \rightarrow A \text{ artıktır mı? } T = \{C, G, B, D, E, A\} \text{ evet}$$

$$F_2 = F_1 - \{CG \rightarrow A\}$$

Türetilirlik Algoritması

Bir kümedeki istersel bağımlılıklardan t istersel b türetilip türetilmeyeceğini bulan algoritmadır.

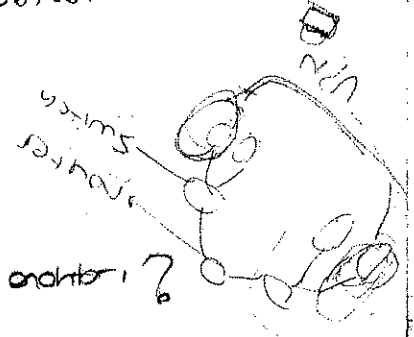
* $AC \rightarrow BDE$ $AC \rightarrow B \rightarrow$ Sağ tarafta oynatılır. Artıklik algoritmasına çok benzer.
 $AC \rightarrow D$
 $AC \rightarrow E$

Bir Nitelik Kümesinin Kapanışı

ÖRNEK: $F: A \rightarrow BCDE$
 $G \rightarrow BD$
 $BC \rightarrow E$
 $CG \rightarrow A$
 $BDE \rightarrow ACG$

Kapanışları?

Süper anahtar?



- * $A^+ = (AB CDE G)$ * $(AB)^+ = AB CDE G$
- * $B^+ = B$ * $(BC)^+ = BC E$
- * $C^+ = C$ * $(CG)^+ = AB CDE G \rightarrow CG \rightarrow A$ ya A nereye? onları yaz.
- * $G^+ = BDG$ * $(CDE)^+ = CDE$
- * $E^+ = E$ * $(DEG)^+ = DEG B A C$
- * $D^+ = D$ * $(BCDE)^+ = AB CDE G$ *

İlişki Anahtarları

Süper Anahtar: Eğer bir nitelik alt kümesi (K) tüm istersel nitelikleri belirliyorsa, K 'nin kapanışı R 'ye eşitse süper anahtar.

Anahtar: Eğer bir nitelik alt kümesi (K) ilişkideki tüm nitelikleri istersel belirliyorsa ($K^+ = R$) ise ve K 'nin hiç bir alt kümesi tüm nitelikleri belirlemiyorsa, K anahtardır.

* Eğer K R 'nin süper anahtarı ise ve K 'nin hiçbir \emptyset alt kümesi R 'nin süper anahtarı değilse K R 'nin anahtardır.

* Her ilişki anahtarı bir nitelik kümesi old. göre, her anahtarın her üst kümesi süper anahtardır.

* Her anahtar aynı zamanda süper anahtardır.

(5)

ÖR: $R(A, B, C, D)$ $F: A \rightarrow B$ $R_1 = (A, B)$ Bu ayrıştırma
işlevsel b. korur mu? $B \rightarrow C$ $R_2 = (B, C)$ $C \rightarrow D$ $R_3 = (C, D)$ $D \rightarrow A$

Korur

 $F^+ : A^+ \rightarrow BCD$ $F_1 = \{A \rightarrow B, B \rightarrow A\}$ $B^+ \rightarrow CDA$ $F_2 = \{B \rightarrow C, C \rightarrow B\}$ $C^+ \rightarrow DAB$ $F_3 = \{C \rightarrow D, D \rightarrow C\}$ $D^+ \rightarrow ABC$ $G = F_1 \cup F_2 \cup F_3 \rightarrow$ $A \rightarrow B$ $A \rightarrow B$ $B \rightarrow A$ $B \rightarrow A$ $B \rightarrow C$ $B \rightarrow C$ $C \rightarrow B$ $C \rightarrow B$ $C \rightarrow D$ $C \rightarrow D$ $D \rightarrow C$ $D \rightarrow C$ $D \rightarrow C$ işlevsel bağımlılığı F 'de yoktur

ama

 $B \rightarrow A$ $C \rightarrow B$ $D \rightarrow C$ $D \rightarrow A$ türetilebiliyor ✓

BCNF Ayrıştırma Algoritması

Bu alg. ile ayrıştırma yaptırır olur ancak işl. bağımlılıkların korunma güvencesi yoktur.

 $R_1 = R, k=1, T=R_1$ F^+ hesapla T 'de BCNF olmayan her R_i için F^+ daki işl. b. R_i 'de tanımlı önemli her $X \rightarrow Y$ işl. b. için X R_i 'de ~~çok~~ anahtarı değilse ($X \rightarrow R_i$ F^+ da yoksa R_i den Y 'deki nitelikler çıkar, k 'yı 1 artır T 'ye $R_k(X, Y)$ ilişki ekle T 'deki tüm ilişkiler BCNF olana kadar tekrarla.ÖR: $R(A, B, C, D, E)$ $F: A \rightarrow CDE$ $E \rightarrow B$ $CD \rightarrow E$ R ilişkisinin anahtarı A 'dır ve BCNF değildir. $F^+ = A \rightarrow BCDE$ $R_1(E, B)$ $F_1: E \rightarrow B$ $E \rightarrow B$ $R_2(C, D, E)$ $F_2: CD \rightarrow E$ $CD \rightarrow BE$ $R_3(A, C, D)$ $F_3: A \rightarrow CD$ $A \rightarrow CD$ $CD \rightarrow E$ $\rightarrow A \rightarrow E$ oluşturulabilir. E, CD, A anahtardırBCNF 'tir ve işlevsel bağımlılıkları korur.

ÖRN: $R(A, B, C)$ $F = \{AB \rightarrow C, C \rightarrow A\}$

AB ve BC anahtardır. Tüm özellikler oral 3NF, anahtar olmayan belirleyen old için BCNF değil.

$F^+ = AB \rightarrow C$ $BCNF$ ayırtırma $\rightarrow R_1(B, C)$ $R_2(C, A)$ $\left. \begin{array}{l} AB \rightarrow C \text{ yitirildi} \\ \text{yitimsiz} \\ \text{isl. bağ. korunmadı.} \end{array} \right\}$
 $C \rightarrow A$

3NF Ayırtırma

Bu ayırtırma ile hem yitimsiz hemde işlevsel b. koruyon ayırtırma elde edilir.

ÖRN: $R(A, B, D, C, E, G)$ $F: AB \rightarrow CD$
 $AC \rightarrow E$
 $DE \rightarrow G$

Tek anahtar AB

geçişli bağımlılıklar nedeniyle 3NF 'dir. F zaten kanonik biçimdedir.

3NF Ayırtırma $\rightarrow R_1(A, B, C, D)$
 $R_2(A, C, E)$
 $R_3(D, E, G)$

BCNF ise $F^+ = AB \rightarrow CDEG$
 $AC \rightarrow E$
 $DE \rightarrow G$
 $ACD \rightarrow G$

1. Ayırtırma

$R_1(ABCD)$
 $R_2(A, C, E)$
 $R_3(DEG)$

isl. b. korun

2. Ayırtırma

$R_1(AB, CD)$
 $R_2(A, C, E)$
 $R_3(A, B, G)$

$\hookrightarrow DE \rightarrow G$ yitirildi

$AB \rightarrow C$
 $AB \rightarrow D$
 $AC \rightarrow E$
 $DE \rightarrow G$

$T = \{ABD\}$ hayır
 $T = \{ABCE\}$ hayır
 $T = \{AC\}$ hayır
 $T = \{DE\}$ hayır.

$AB \rightarrow CD$
 $AC \rightarrow E$
 $DE \rightarrow G$

$R_1(A, B, C, D)$
 $R_2(A, C, E)$
 $R_3(DEG)$

$F^+ = AB \rightarrow CD$
 $AC \rightarrow E$
 $DE \rightarrow G$
 $ACD \rightarrow G$

$R_1(ABCD)$ ✓
 $R_2(A, C, E)$ ✓
 $R_3(DEG)$ ✓
 ~~$R_4(ACDG)$~~

isl. korun.



(3)

ÖRNEK: $R(A, B, C, D, E)$ $F = A \rightarrow G$ $B \rightarrow D$ $D \rightarrow AE$ $CE \rightarrow B$ İlişkinin 4 anahtar
veya anahtar adayı vardır
 $\rightarrow B, D, AE, CE$ $B^+ = BDAEC \rightarrow$ anahtar $D^+ = DAECB \rightarrow$ anahtar $AE^+ = AECLB \rightarrow$ anahtar $CE^+ = \underline{CEBDA} \rightarrow$ anahtar
 R $B, AB, ABC, DC, ADE, ABCDE$
ilişkinin süper anahtarlarıdır.

ADECE → ADECE

İlişkiler için Normal Biçimler

Sorunsuz 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, matris ya da karmaşık değer olmaması gerekir.

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

 \rightarrow Ders niteliği yalın bir nitelik olmadığı için 1NF değil!ÖĞRENCİ (ÖNO, ÖADI, DERS, DADI, NOTU) \rightarrow 1NF

2- İkinci Normal Biçim (2NF)

 \rightarrow Asal Nitelik: İlişki anahtarlarının en az birinde yer alan nitelikler \rightarrow Asal Olmayan Nitelik: İlişki anahtarlarının hiç birinde yer olmayan niteliklerdir.* 2NF \rightarrow Bir ilişki 1NF ise ve asal olmayan niteliklerden hiçbirisi anahtarlardan hiç birine ^{kısmi} işlevsel bağımlı değilse 2NF'tir.

ÖRNEK: SATICI (ÜKODU, FNO, FADI, FADRESİ, SFİYATI)

 \rightarrow Tüm nitelikler tek değerli yalın nitelikler old. için 1NF \rightarrow İlişkinin tek anahtarı FNO, ÜKODU'dur.* 2NF \rightarrow Asal olmayan tüm nitelikler tüm anahtarlara tam işl. b. olacak \rightarrow ÜKODU ve FNO asal nitelikler, FAD, FADRESİ, SFİYATI asal olmayan nit. \rightarrow FADI ve FADRESİ anahtarlara kısmi işlevsel bağımlı old. için

2NF değildir.

ÜKODU, FNO \rightarrow FADI \rightarrow kısmiÜKODU, FNO \rightarrow FADRESİ \rightarrow kısmiÜKODU, FNO \rightarrow SFİYATI \rightarrow tamNOT: Eğer tek anahtar varsa
hepsi tam işlevsel b. dir.

3- Üçüncü Normal Biçim (3NF)

Eğer bir ilişki 2NF ve asal olmayan hiçbir nitelik hiçbir anahtara geçişli bağımlı değilse 3NF'tir

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

- Tüm nitelikler yalın olduğundan 1NF
- Tek anahtar PLAKANO'dur. Bunun dışındakiler asal olmayan nitelik. Nitelik hepsi anahtara tam işlevsel bağımlıdır. 2NF'dir.
- Zaten tek nitelikten oluşan bir anahtara niteliklerin kısmi b. old. düşünülmez.
- Asal dm. nitelikler birbirinden bağımsız değildir.
MARKA, MODEL → AĞIRLIK işlevsel b. mevcuttur. Bu yüzden 3NF değil.

Boyce Codd Normal Biçimi (BCNF)

Eğer bir ilişki 1NF ve tüm belirleyiciler ilişkinin anahtarı ise BCNF

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

$B \rightarrow F$ ise F 'in sol tarafının tamamı anahtar ise BCNF

ÖRNEK: ÖĞRETS (ÖĞRNO, DKODU, DADI, NOTU) $F = DKODU \rightarrow DADI$

$DADI \rightarrow DKODU$

$(DKODU, ÖGRNO)^+ = DKODU, ÖGRNO, DADI, NOTU = ÖGRETS$

$DKODU, ÖGRNO \rightarrow NOTU$

$(DADI, ÖGRNO)^+ = DADI, ÖGRNO, DKODU, NOTU = ÖGRETS$

$DADI, ÖGRNO \rightarrow NOTU$

→ anahtarlar

→ 1NF'tir.

→ 2NF'tir

→ Tek asal olmayan nitelik NOTU old. için 3NF'tir.

Baska asal " " varsa anahtar ile onlarında ilişki olup olmadığına bakılır ona göre karar verilir.

→ BCNF değildir.

İşlevsel B. Türleri

1- Kısmi işli B. \rightarrow Gereksiz nitelik var.

$\text{ÖNO}, \text{ÖADI} \rightarrow \text{BNO}$ (BNO'yu öğrenmem için ÖADI bilgisine gerek yok)

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

2- Tam işlevsel B. \rightarrow Gereksiz nitelik yok.

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

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

3- Önemli işli B \rightarrow Sağ taraf sol tarafın alt kümesi veya aynısı

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

$\text{DKODU} \rightarrow \text{DKODU}$

4- Önemli işli B \rightarrow Sağ taraf sol tarafın aynı veya alt kümesi değil.

$\text{ÖNO} \rightarrow \text{ÖADI}$

5- Geçişli işli B

$X \rightarrow Y$

$Y \rightarrow Z$

$X \rightarrow Z$ yi belirtir.

İşlevsel B. ilgili Tanım, Önerme ve Algoritmalar

1) İşlevsel B kümesinin Kapanışı (F^+)

F'deki işlevsel b. türetilebilecek tüm kısmi ve geçişli bağılıkları içeren işlevsel b. kümesine F'nin kapanışı denir. (F^+)

$R(A, B, C, D), F = A \rightarrow B$

$BC \rightarrow D$

$A \rightarrow AB$

$B \rightarrow B$

$C \rightarrow C$

$D \rightarrow D$

$AB \rightarrow AB$

$AC \rightarrow ABCD$

$AD \rightarrow ABCD$

$BC \rightarrow BCD$

$BD \rightarrow BD$

$CD \rightarrow CD$

$ABC \rightarrow ABCD$

$ABD \rightarrow BD$

$ACD \rightarrow ABCD$

$BCD \rightarrow BCD$

$ABCD \rightarrow ABCD$

Bütünlük Kısıtlamaları ve İlişkisel Tasarım

1- Bütünlük Kısıtlamaları: Yanlış, eksik, tutarsız veri içermemesi istenir.

2- Alan Kısıtlamaları: Her niteliğe bir alan eslenmesi ve niteliğin alabileceği değerlerin bu alandaki değerlerle sınırlanması ile ilgilidir. 3'e ayrılır.

* Alan türleri → tamsayı, kesirli s. vs

* Değer sınırları → Alt sınır değeri, üst sınır değeri vs

* Boş (null) değer → Niteliklerin boş olup olmayacağı

3- Referans Kısıtlamaları: Bir ilişkideki çoklukların varlığının, bir başka ilişkideki belirli çoklukların varlığına bağlı olmasıdır.

ÖRÜN (Üno, ^{Primary}Tarih) → primary
 MAĞAZA (MNO, Tarih)
 SATIŞ (Süno, ^{Foreign key}Smno)

4- Nitelikler Arası

İşlevsel Bağımlılık

2 bir ilişki seması X ve Y alt kümeleri olsun.

Eğer X nitelik değerleri aynı olan tüm çokluklarda Y nitelik değerleri de aynıysa $X \rightarrow Y$ işlevsel belirtir. (Y X 'e işlevsel bağımlıdır.)

ÖRNEK: TAŞIT (PLAKANO, MARKA, MODEL, YIL, AĞIRLIK, RENK)

İşlevsel b. göster?

PLAKANO → MARKA, MODEL, YIL, AĞIRLIK, RENK

MARKA, MODEL → AĞIRLIK (marka ve model bildikten sonra ağırlığa gerek yok)

ÖRNEK 2 (A, B, C, D) ilişki semasına göre oluşturulmuş r örneği verilmiştir.

r :

A	B	C	D
1	a	x	e
2	a	y	b
3	b	x	c
4	c	x	c
5	a	x	e

$D \rightarrow C$ ilişkisi vardır

$C \rightarrow D$ " yoktur (x hem e hem c'ye gittiği için)

$AB \rightarrow CD$

$AC \rightarrow BD$

$AD \rightarrow BC$

$BC \rightarrow D$ 'ye

İşlemler Başgimlilik

2 bitlik işlemci X ve Y ise niteliklerin işi alt kümeler
→ Eğer X nitelik değeri aynı olan tüm cokluklarda Y nitelik
değerinin de aynı olması gerekiyorsa X Y'yi işlemler
belirler $X \rightarrow Y$

1) Kuvamsal Düzeyde

Ö: SATICI (ÜKODU, FNO, FAD1, FADRESİ, SEYHATI)
* FNO → FAD1, FADRESİ
ÜKODU, FNO → SEYHATI

2) Ağ Düzeyinde

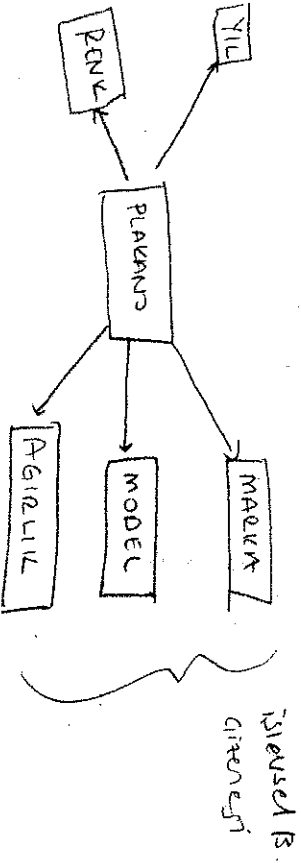
İstikide o anda mevcut cokluklara göre belirli işlemler başgimlilikler
sağlar.

Ö: SATICI (ÜKODU, FNO, FAD1, FADRESİ, SEYHATI)

* FNO → FAD1, FADRESİ
FAD1 → FNO, FADRESİ
ÜKODU, FNO → SEYHATI
FADRESİ → FNO, FAD1

Ö: TAAIT (PLAKANO, MARKA, MODEL, YIL, AGİRLİK, RENK)

* PLAKANO → MARKA, MODEL, YIL, AGİRLİK, RENK
MARKA, MODEL → AGİRLİK



İşlemler Başgimlilik Tanı

Tan İşlemler B. Eğer X A'yı belirliyorsa ve X'in hiçbir öalt
kümesi A'yı belirlemiyorsa tanıdır.

Önemli İşlemler B. Eğer X A'yı belirliyorsa ve A X'in bir altkümesi
ise önemli

Geciklik İşlemler B. Eğer X Y'yi Y de Z'yi belirliyorsa
(X → Y → Z) X → Z işlemler b. geciktiricidir.

Önemli işl. B. X A'yı belirliyorsa ve A X'in bir altkümesi
değilse

Ö: 2 (ÖNO, ÖAD1, BNO, BADI, FAKNO, OKODU, ÖAD1, KOD, NOTU)

* Tan İşlemler B. Ö → ÖNO → ÖAD1, BNO

BNO → FAKNO
ÖNO, OKODU → NOTU

* Geciklik işl. B. Ö

→ ÖNO → FAKNO (ÖNO → BNO, BNO → FAKNO)

→ ÖNO → BADI (ÖNO → BNO, BNO → BADI)

* Önemli işl. B. Ö → ÖNO, ÖAD1 → ÖAD1

BNO, ÖAD1, FAKNO → BADI, FAKNO

* Önemli işl. B. Ö → ÖNO → ÖAD1

ÖNO, ÖAD1, OKODU → BNO, KOD
ÖNO, OKODU → NOTU.

HARAKET KAVRAMI

İki komutun gelişmesi:

$\left. \begin{array}{l} \text{Read}_1(A), \text{write}_2(A) \\ \text{Read}_2(A), \text{write}_1(A) \\ \text{write}_1(A), \text{write}_2(A) \\ \text{Read}_1(A), \text{Read}_2(A) \end{array} \right\} \text{Gelişen} \\ \text{komut çiftleri}$