

İLİŞKİSEL CEBİR

İlişkisel cebir (Relational Algebra) biçimsel bir sorgu dili olarak tanımlanabilir. Bu dil yardımıyla birçok sorgulama işlemi tanımlanabilir. Örneğin, belirli kayıtların seçilmesi, belirli sütunların sorgu sonuçlarına dahil edilmesi gibi işlemler ilişkisel cebir ifadeleriyle kolayca ortaya konabilir.

İlişkisel cebir, bir veya iki ilişkiyi girdi olarak alıp, sonuç olarak yeni bir ilişki üreten bir dizi işlemden oluşur. Ancak bu sorgulamalar sadece biçimsel olarak yapılır. Yani bu sorgulama dili için bir yorumlayıcı veya derleyici yoktur.

İlişkisel cebirin şu temel işlemleri bulunmaktadır: Seçim (Selection), Atma (Projection), Kartezyen Çarpım (Cartesian Product), Birleşim (Union), Kesişim (Intersection), Fark (Difference), Doğal Birleştirme (Join), Bölme (Division).

Seçim İşlemi

Belirli bir ilişkiden bazı kayıtları seçerek alma işlemidir.

$\sigma_{\text{seçim kriteri}} (TABLO)$

Seçim kriterinde karşılaştırma işlemleri kullanılır: =, ≠, <, >, ≤, ≥.

Mantıksal işlemler için : “∧” ve, “∨” veya kullanılır.

Örnek

Müşteri tablosu

Müşteri adı	No	İlçe	Şehir	Bakiye
Ahmet	2520	Beşiktaş	İstanbul	100
Faruk	6345	Beşiktaş	İstanbul	150
Ali	6755	Şişli	İstanbul	250
Veli	3054	Kızılay	Ankara	100
Zeki	2521	Ulus	Ankara	500

Beşiktaş ilçesindeki müşterileri seçmek :

$\sigma_{\text{İlçe="Beşiktaş"}} (MÜŞTERİ)$

Sorgu sonucu :

Müşteri adı	No	İlçe	Şehir	Bakiye
Ahmet	2520	Beşiktaş	İstanbul	100
Faruk	6345	Beşiktaş	İstanbul	150

İlçesi Beşiktaş ve bakiye miktarı 100'den büyük olan müşterileri seçmek :

$\sigma_{İlçe="Beşiktaş" \wedge Bakiye>100} (MÜŞTERİ)$

Sorgu sonucu :

Müşteri adı	No	İlçe	Şehir	Bakiye
Faruk	6345	Beşiktaş	İstanbul	150

Atma İşlemi

Belirli bir ilişkiden bazı sütunları seçerek alma işlemidir.

$\Pi_{sütun isimleri} (TABLO)$

Örnek

Sadece müşteri adı ve şehir bilgileri isteniyor.

$\Pi_{müşteri adı, şehir} (MÜŞTERİ)$

Sorgu sonucu :

Müşteri adı	Şehir
Ahmet	İstanbul
Faruk	İstanbul
Ali	İstanbul
Veli	Ankara
Zeki	Ankara

Bakiyesi 200'den fazla olan müşterilerin adı ve bakiyesi?

$\Pi_{\text{müşteri adı, bakiye}} (\sigma_{\text{Bakiye} > 200} (\text{MÜŞTERİ}))$

Müşteri adı Bakiye

Ali 250

Zeki 500

Bakiyesi 150-250 arasında olan veya Kızılay'da oturan müşterilerin adı, bakiyesi, ilçesi?

$\Pi_{\text{müşteri adı, bakiye, ilçe}} (\sigma_{\text{Bakiye} \geq 150 \wedge \text{Bakiye} \leq 250 \vee \text{ilçe} = \text{"Kızılay"}} (\text{MÜŞTERİ}))$

Müşteri adı Bakiye İlçe

Faruk 150 Beşiktaş

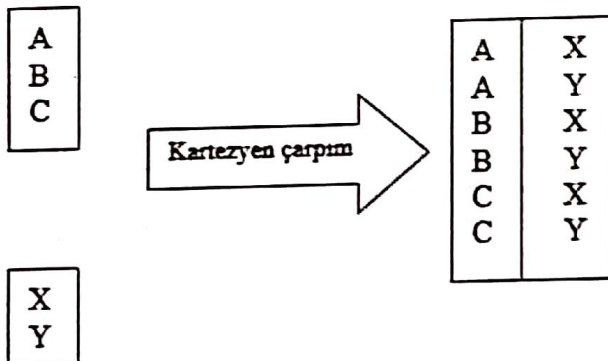
Ali 250 Şişli

Veli 100 Kızılay

Çarpma İşlemi

Belirli iki ilişkiden mümkün olabilecek tüm çiftleri elde ederek tek bir ilişki biçiminde göstermek için kartezyen çarpım oluşturulur.

Tablo1 x Tablo2



Örnek

ÖĞRENCİ tablosu

<u>Öğrenci</u>	<u>Bölüm</u>
Burak	Bilgisayar
Begüm	İktisat
Dilay	İktisat
Selin	İşletme
Seray	Hukuk

DERSLER tablosu

<u>Ders</u>	<u>Saat</u>
Matematik	3
İstatistik	2

ÖĞRENCİ x DERSLER
ikilileri ifade eder)

(Bu ifade Öğrenci ve Dersler tablolarından elde edilebilecek tüm

<u>Öğrenci</u>	<u>Bölüm</u>	<u>Ders</u>	<u>Saat</u>
Burak	Bilgisayar	Matematik	3
Begüm	İktisat	Matematik	3
Dilay	İktisat	Matematik	3
Selin	İşletme	Matematik	3
Seray	Hukuk	Matematik	3
Burak	Bilgisayar	İstatistik	2
Begüm	İktisat	İstatistik	2
Dilay	İktisat	İstatistik	2
Selin	İşletme	İstatistik	2
Seray	Hukuk	İstatistik	2

İktisat bölümünde okuyan öğrencilerin tüm dersleri alması durumunda ilgili sorgu : $\sigma_{Bölüm="İktisat"} (\text{ÖĞRENCİ} \times \text{DERSLER})$

<u>Öğrenci</u>	<u>Bölüm</u>	<u>Ders</u>	<u>Saat</u>
Begüm	İktisat	Matematik	3

Dilay	İktisat	Matematik	3
Begüm	İktisat	İstatistik	2
Dilay	İktisat	İstatistik	2

İktisat bölümünde okuyan öğrencilerin tüm dersleri alması durumunda yalnızca adları ve aldıkları derslerin adlarını seçmek için :

$\Pi_{\text{Öğrenci.Ders}} (\sigma_{\text{Bölüm} = \text{"İktisat"}} (\text{ÖĞRENCİ} \times \text{DERSLER}))$

<u>Öğrenci</u>	<u>Ders</u>
Begüm	Matematik
Dilay	Matematik
Begüm	İstatistik
Dilay	İstatistik

Birleşim İşlemi

İki ilişkiden birinde veya her ikisinde birden bulunan kayıtların seçilmesi için yapılan işlemdir. \cup işareti ile gösterilir.

Örnek

KREDİ tablosu

<u>Müşteri</u>	<u>Bakiye</u>	<u>İlçe</u>
Burak	100	Beşiktaş
Begüm	200	Şişli
Sezin	150	Ulus

MEVDUAT tablosu

<u>Müşteri</u>	<u>Bakiye</u>	<u>İlçe</u>
Burak	50	Beşiktaş

Selin	125	Beşiktaş
Sezin	300	Ulus
Begüm	100	Ulus

Bankanın Beşiktaş şubesinde mevduat ve/veya kredi hesabı bulunan kişileri seçmek ve sadece isimlerini sunmak için :

$$\Pi_{Kredi.Müşteri}(\sigma_{İlçe="Beşiktaş"}(KREDİ)) \cup \Pi_{Mevduat.Müşteri}(\sigma_{İlçe="Beşiktaş"}(MEVDUAT))$$

Müşteri

Burak

Selin

Kesişim İşlemi

İki ilişkide de bulunan kayıtların belirlenmesi için kullanılır. \cap işareti ile gösterilir.

Örnek

Bankanın Beşiktaş şubesinde hem mevduat hem de kredi hesabı bulunan kişileri seçmek ve sadece isimlerini sunmak için :

$$\Pi_{Kredi.Müşteri}(\sigma_{İlçe="Beşiktaş"}(KREDİ)) \cap \Pi_{Mevduat.Müşteri}(\sigma_{İlçe="Beşiktaş"}(MEVDUAT))$$

Müşteri

Burak

Fark İşlemi

İki ilişkiden birinde bulunup diğerinde bulunmayan kayıtların seçilmesi işlemidir. - işareti ile gösterilir.

Örnek

Bankanın Ulus şubesinde mevduatı olup bu şubede kredisi olmayan müşterilerin isimlerini sunmak için :

$$\Pi_{Mevduat.Müşteri}(\sigma_{İlçe="Ulus"}(MEVDUAT)) - \Pi_{Kredi.Müşteri}(\sigma_{İlçe="Ulus"}(KREDİ))$$

Müşteri

Begüm

Doğal Birleştirme İşlemi

(A,B) ve (B,C) niteliklerine sahip iki ayrı ilişkinin (A,B,C) niteliklerine sahip tek bir ilişki haline dönüştürülmesine "doğal birleştirme" adı verilir. Birleştirme ilişkili sütunlar üzerinden gerçekleştirilir.

Tablo1 \bowtie Birleştirme sonucu Tablo2

A	X
B	X
C	Y

Doğal Birleştirme

A	X	1
B	X	1
C	Y	2

X	1
Y	2
Z	3

Örnek

KREDİ tablosu

Adı	Bakive	Şube
Burak	100	Beşiktaş
Begüm	200	Şişli

MÜŞTERİ tablosu

Müş adı	İlçe	İl
Burak	Beşiktaş	İstanbul
Begüm	Kadıköy	İstanbul
Selin	Bakırköy	İstanbul
Sezin	Şişli	İstanbul

Dilay Kızılay Ankara

Kredi almış müşterilerin adı, bakiyesi, oturduğu il?

$\Pi_{\text{Kredi Adı Kredi Bakiye Müşteri İl}} (KREDİ \bowtie_{\text{Ad=Müş adı}} MÜŞTERİ)$

Adı	Bakiye	İl
Burak	100	İstanbul
Begüm	200	İstanbul

Kadıköy'de oturan müşterilerin adı, kredi bakiyesi, kredi şubesi?

$\Pi_{\text{Müş adı Bakiye Şube (İlçe="Kadıköy" (MÜŞTERİ) \bowtie_{\text{Müş adı=Ad} KREDİ)$

Müş adı	Bakiye	Şube
Begüm	200	Şişli

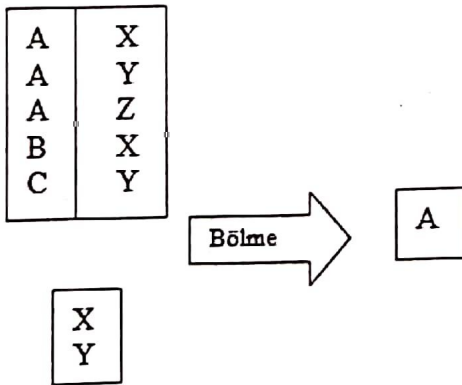
Kredi hesabı olan ve aynı ilçede oturan tüm müşterilerin adları ve ilçeleri?

$\Pi_{\text{Kredi Adı Müşteri İlçe}} (KREDİ \bowtie_{\text{Ad=Müş adı Şube=İlçe}} MÜŞTERİ)$

Adı	İlçe
Burak	Beşiktaş

Bölme İşlemi

İkili ve birli iki ayrı ilişkiyi karşılaştırarak, birli olan ilişkiye eş olan ikinci ilişkinin değerlerinden oluşan yeni bir ilişki oluşturulabilir. : işareti ile gösterilir.



Örnek

Bir bankanın İstanbul ilindeki bütün şubelerinde mevduat hesabı olan müşterilerini öğrenmek.

SUBE tablosu

<u>Şube adı</u>	<u>İl</u>
Beşiktaş	İstanbul
Şişli	İstanbul
Yayla	İstanbul
Kızılay	Ankara
Ulus	Ankara

İstanbul'da yer alan tüm şubeler:

$$\pi = \Pi_{\text{Şube adı} (Q_{II} = \text{"İstanbul"} (ŞUBE))}$$

Şube adı
Beşiktaş
Şişli
Yayla

MEVDUAT tablosu

<u>Müşteri</u>	<u>Bakiye</u>	<u>İlçe</u>
Burak	50	Beşiktaş
Selin	125	Beşiktaş
Sezin	300	Ulus
Dilay	200	Kızılay
Burak	100	Şişli
Burak	200	Yayla

$$\rho = \Pi_{\text{Müşteri, İlçe} (MEVDUAT)}$$

<u>Müşteri</u>	<u>İlçe</u>
Burak	Beşiktaş
Selin	Beşiktaş
Sezin	Ulus
Dilay	Kızılay
Burak	Şişli
Burak	Yayla

$$\rho : \pi \longrightarrow \begin{array}{c} \text{Müşteri} \\ \text{Burak} \end{array}$$