

İLİŞKİSEL CEBİR VE İLİŞKİSEL HESAPLAMA

İlişkisel Veri Tabanı Yönetim Sistemlerinde kullanılan ilişkisel diller kuramsal temelleri açısından ikiye ayrılır:

1. **İlişkisel Cebir:** İlişkiler üzerinde cebirsel işlemler uygulanıp yeni ilişkilere elde ederek çeşitli sorguların yanıtlarını elde etmek için kullanılır. Kullanıcının hangi işlemlerin hangi sırayla uygulanacağını yazarak istediği bilginin nasıl elde edileceğini belirtmesi gerekir.
2. **İlişkisel Hesaplama:** İstenilen kayıtlara ulaşmak için gerekli sorgular söz konusu kayıtların sağlaması gereken niteliklerle ilgili önermeler oluşturularak tanımlanır. Kullanıcı kayıtlarla ilgili istediği önermeleri oluşturur, ancak bu kayıtların nasıl elde edileceğine ilişkin bilgiyi vermez.

İLİŞKİSEL CEBİR İŞLEMLERİ

İlişkisel cebir, ilişkilere uygulanan yüksek düzeyli şu işlemlerden oluşur:

- Seçme İşlemi (Select Operation)
- İzdüşüm İşlemi (Project Operation)
- Aktarma İşlemi ve Ara İlişkilerin İsimlendirilmesi (Rename Operation)
- Birleşim İşlemi (Union Operation)
- Kesişim İşlemi (Intersection Operation)
- Fark işlemi (Minus Operation)
- Kartezyen Çarpım İşlemi (Cartesian Product İşlemi)
- Birleştirme İşlemi (Join Operation)

- Doğal Birleştirme İşlemi (Natural Join Operation)
- Bölme İşlemi (Division Operation)

Seçme İşlemi:

Bir ilişki / tablo içerisinde belli bir koşulu sağlayan kayıtların / satırların seçilmesi işlemidir.

$\sigma_{\langle \text{seçme koşulu} \rangle}$ (**R**) biçiminde yazılır.

Burada σ işlem işaretini ve **R** işlemin uygulanacağı ilişkinin adını ifade eder.

Örnek:

2000 YTL'den fazla maaş alan çalışanları seçmek için:

$\sigma_{\text{MAAŞ} > 2000}$ (**ÇALIŞAN**)

4 No'lu departmanda çalışıp 2000 YTL'den fazla maaş alan veya 5 No'lu departmanda çalışıp 1500 YTL'den fazla maaş alan çalışanları seçmek için:

$\sigma_{(\text{DNO}=4 \wedge \text{MAAŞ} > 2000) \vee (\text{DNO}=5 \wedge \text{MAAŞ} > 1500)}$ (**ÇALIŞAN**)

İzdüşüm İşlemi:

Bir ilişki / tablo içerisinde belli bir koşulu sağlayan niteliklerin / sütunların seçilmesi işlemidir.

$\pi_{\langle \text{nitelik listesi} \rangle}$ (**R**) biçiminde yazılır.

Örnek:

Çalışan tablosundaki kayıtların sadece maaş ve personel numarası bilgilerini elde etmek için:

$\pi_{\text{MAAŞ, PNO}}(\text{ÇALIŞAN})$

2000 YTL'den fazla maaş alan çalışanların personel numaralarını elde etmek için:

$\pi_{\text{PNO}}(\sigma_{\text{MAAŞ} > 2000}(\text{ÇALIŞAN}))$

Aktarma İşlemi ve Ara İlişkilerin İsimlendirilmesi:

Tüm ilişkisel cebir işlemleri parantezler kullanılarak tek bir işlem satırında yapılabilir olsalar da, çoğu zaman olası karışıklıkları önlemek adına deyimler parçalanarak yazılırlar. Ayrıca, farklı işlemlerle elde edilen aynı nitelikleri içeren kayıtların sonraki işlemlerde karıştırılmadan kullanılabilmesi önemlidir. Bu durumlar sözkonusu olduğunda, problemleri ortadan kaldırmak için aktarma işlemi uygulanmaktadır.

P [yeni nitelik isimleri] ← ilişkisel cebir deyimini biçiminde yazılır.

Burada **P** ara ilişkinin ismini, **yeni nitelik isimleri** ara ilişkinin niteliklerine verilen yeni isimleri ve **←** aktarma işaretini temsil eder.

Örnek:

4 No'lu departmanda çalışanları isimlendirmek için:

DEP4_Ç $\leftarrow \sigma_{DNO=4}$ (ÇALIŞAN)

Birleşim İşlemi:

İki ilişkinin / tablonun küme birleşiminin alınması işlemidir. Bu işlemin uygulanabilmesi için iki ilişkinin derecelerinin (nitelik sayılarının) aynı olması ve aynı sıradaki niteliklerin benzer yapıda veri içermesi (tarih, string, vb.) gerekir.

$R_1 \cup R_2$ biçiminde yazılır.

Örnek:

ÖĞRENCİ	Adı	Soyadı
	Defne	Opal
	Serçin	Metin
	İlke	Doğruöz
	Murat	Uslu
	Alev	Sezgin
	Kutlu	Esendal

ASİSTAN	İsim	Soyisim
	Kerem	Özgür
	Yusuf	Elekçi
	Serçin	Metin
	Alev	Sezgin

ÖĞRENCİ \cup ASİSTAN	Adı	Soyadı
	Defne	Opal
	Serçin	Metin
	İlke	Doğruöz
	Murat	Uslu
	Alev	Sezgin
	Kutlu	Esendal
	Kerem	Özgür
	Yusuf	Elekçi

Kesişim İşlemi:

İki ilişkinin / tablonun küme kesişiminin alınması işlemidir. Bu işlemin uygulanabilmesi için iki ilişkinin derecelerinin (nitelik sayılarının) aynı olması ve aynı sıradaki niteliklerin benzer yapıda veri içermesi (tarih, string, vb.) gerekir.

$R_1 \cap R_2$ biçiminde yazılır.

Örnek:

ÖĞRENCİ	Adı	Soyadı
	Defne	Opal
	Serçin	Metin
	İlke	Doğruöz
	Murat	Uslu
	Alev	Sezgin
	Kutlu	Esental

ASİSTAN	İsim	Soyisim
	Kerem	Özgür
	Yusuf	Elekçi
	Serçin	Metin
	Alev	Sezgin

ÖĞRENCİ \cap ASİSTAN	Adı	Soyadı
	Serçin	Metin
	Alev	Sezgin

Fark işlemi :

İki ilişkinin / tablonun küme farkının alınması işlemidir. Bu işlemin uygulanabilmesi için iki ilişkinin derecelerinin (nitelik sayılarının) aynı olması ve aynı sıradaki niteliklerin benzer yapıda veri içermesi (tarih, string, vb.) gerekir.

$R_1 - R_2$ biçiminde yazılır. ($R_1 - R_2 \neq R_2 - R_1$)

Örnek:

ÖĞRENCİ	Adı	Soyadı
	Defne	Opal
	Serçin	Metin
	İlke	Doğruöz
	Murat	Uslu
	Alev	Sezgin
	Kutlu	Esendal

ASİSTAN	İsim	Soyisim
	Kerem	Özgür
	Yusuf	Elekçi
	Serçin	Metin
	Alev	Sezgin

ÖĞRENCİ – ASİSTAN	Adı	Soyadı
	Defne	Opal
	İlke	Doğruöz
	Murat	Uslu
	Kutlu	Esendal

ASİSTAN – ÖĞRENCİ	İsim	Soyisim
	Kerem	Özgür
	Yusuf	Elekçi

Kartezyen Çarpım İşlemi :

İki ilişkinin / tablonun küme çarpımının bulunması işlemidir. Sonuçta elde edilen ilişki / tablo, işleme giren ilişkilerdeki kayıtların ardarda eklenmesiyle oluşan tüm kayıtları içerir.

$$R_1 \times R_2 = \{ \overline{a_1 a_2} : a_1 \in R_1, a_2 \in R_2 \}$$

R_1 ve R_2 ilişkilerinin dereceleri sırasıyla n_1 ve n_2 , kayıt sayıları ise k_1 ve k_2 ise kartezyen çarpım işleminin sonucunda elde edilen ilişkinin derecesi $n_1 + n_2$, kayıt sayısı ise $k_1 \times k_2$ olarak bulunur.

Örnek:

ÖĞRENCİ	Adı	Soyadı
	Defne	Opal
	Serçin	Metin
	Alev	Sezgin
	Kutlu	Esendal

DERS	Dadı	Kredi
	Matematik	4
	Fizik	3

ÖĞRENCİ x DERS	Adı	Soyadı	Dadı	Kredi
	Defne	Opal	Matematik	4
	Serçin	Metin	Matematik	4
	Alev	Sezgin	Matematik	4
	Kutlu	Esendal	Matematik	4
	Defne	Opal	Fizik	3
	Serçin	Metin	Fizik	3
	Alev	Sezgin	Fizik	3
	Kutlu	Esendal	Fizik	3

Birleştirme İşlemi :

İki ilişkide bulunan kayıtlar arasından birbiriyle ilgili olanların tek kayıt haline getirilmesi işlemidir. İlişkiler arası ilişkileri kontrol edebilmemize olanak sağlaması açısından çoklu ilişkiler içeren veri tabanları için önemli bir işlemdir.

$R_1 \bowtie_{\langle \text{birleştirme koşulu} \rangle} R_2$ biçiminde yazılır.

Burada R_1 ve R_2 birleştirilen ilişkiler \bowtie ise birleştirme işaretini temsil eder.

Birleştirme işleminde en çok kullanılan aritmetik karşılaştırma koşulu eşitliktir. Sadece “=” işlecini kullandığı bu birleştirme işlemleri **Eşit Birleştirme (Equijoin)** olarak adlandırılır.

Örnek:

20KY3 kodlu ürünü üreten firmaların numaraları ile bu ürüne ilişkin üretim adeti + 20KY3 kodlu ürünün 2. parti ürün teslimindeki satış miktarı:

$$\pi_{FNO, ADET} (\sigma_{ÜKODU='20KY3' (ÜRETİM)}) \bowtie$$

$$ADET > ADET$$

$$\pi_{ADET} (\sigma_{(ÜKODU='20KY3') \wedge (PNO=2)} (SATIŞ))$$

Doğal Birleştirme İşlemi :

Birleştirilen ilişkilerde / tablolarda aynı adı taşıyan en az bir nitelik varsa, doğal birleştirme işlemi uygulanarak ilişkiler aynı adı taşıyan nitelikler üzerinden birleştirilir ve tekrarlı niteliklerden birer tanesi atılır.

$R_1 \bowtie R_2$ veya $R_1 * R_2$ biçiminde yazılır.

Örnek:

DEPARTMAN	Dadı	Dno
	Üretim	1
	Arge	2

PROJE	Pkodu	Dno
	A34R5G	2
	JK89L7	1
	JK76H6	1
	A25L7J	2

DEPT_PROJE	Dadı	Dno	Pkodu
	Arge	2	A34R5G
	Üretim	1	JK89L7
	Üretim	1	JK76H6
	Arge	2	A25L7J

Bölme İşlemi :

İlişkisel cebir işlemleri arasında en karmaşık olanıdır. İşlemi uygulanabilmesi için ilişkilerde / tablolarda aynı adı taşıyan en az bir niteliğin bulunması gerekir.

$R_1(Z) \div R_2(X)$ biçiminde yazılır. ($X \subseteq Z$)

Örnek:

$T \leftarrow R \div S$

R	A	B
	a ₁	b ₁
	a ₂	b ₁
	a ₃	b ₁
	a ₄	b ₁
	a ₁	b ₂
	a ₃	b ₂
	a ₂	b ₃
	a ₃	b ₃
	a ₄	b ₃
	a ₁	b ₄
	a ₂	b ₄
	a ₃	b ₄

S	A
	a ₁
	a ₂
	a ₃
T	B
	b ₁
	b ₄

İLİŞKİSEL HESAPLAMA İŞLEMLERİ

İlişkisel hesaplama, ifade gücü açısından ilişkisel cebire denk olmasına rağmen biçimsel olarak farklılıklar gösterir. İlişkisel cebirde sorunun çözümü için takip edilecek prosedür verilirken ilişkisel hesaplamanın soru çözümünde prosedür belirleyici herhangi bir yaklaşımı yoktur. Hesaplamalar niceleme mantığına (predicate logic) dayalıdır.

İlişkisel Hesaplama Deyimleri

Genel olarak ilişkisel hesaplama kullanılan deyimler şu formdadır:

$$\{t_1.A_j, t_2.A_k, \dots, t_n.A_m \mid \text{KOŞUL}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$$

Burada $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}$ kayıt değerlerini, her A_i sözkonusu kayıt için geçerli bir niteliği ve KOŞUL ise ilişkisel hesaplamanın gerçekleştirilme koşulunu / formülünü ifade eder.

Formüllerdeki önermeler şu şekillerde oluşturulabilir:

1. $R(t_i) \rightarrow R$ ilişkisindeki t_i kaydının içerdiği değişkenleri ifade eder.
2. $t_i.A \text{ op } t_j.B \rightarrow t_i$ kaydının A niteliği ile t_j kaydının B niteliğinin herhangi bir karşılaştırma operatörü ($\equiv, \leq, \geq, \neq, <, >$) kullanılarak karşılaştırılmasını ifade eder.
3. $t_i.A \text{ op } c / c \text{ op } t_j.B \rightarrow t_i$ kaydının A niteliğinin ya da t_j kaydının B niteliğinin herhangi bir karşılaştırma operatörü ($\equiv, \leq, \geq, \neq, <, >$) kullanılarak sabit bir sayıyla karşılaştırılmasını ifade eder.

Bu önermeler ve mantık operatörleri (\wedge (ve), \vee (veya), \neg (değil)) kullanılarak yaratılan ilişkisel hesaplama formülleri için şu kurallar geçerlidir:

1. Her önerme bir formüldür.
2. F_1 ve F_2 formül ise şu ifadeler de formüldür:
 - a. $F_1 \wedge F_2$
 - b. $F_1 \vee F_2$
 - c. $\neg(F_1)$ ve $\neg(F_2)$

Formüllerin doğruluk değerleri içerdikleri önermelerin doğruluk değerlerine bağlı olarak aşağıdaki tabloda belirtildiği gibi hesaplanır (T=TRUE/DOĞRU, F=FALSE/YANLIŞ).

F_1	F_2	$F_1 \wedge F_2$	$F_1 \vee F_2$	$\neg(F_1)$	$\neg(F_2)$
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

Varlıksal ve Tümel Niceleyiciler

İlişkisel hesaplama formüllerinde sıklıkla kullanılan iki çeşit niceleyici vardır:

1. Varlıksal Niceleyici (\exists) \rightarrow vardır / bulunur (there exists) anlamını içerir.
2. Tümel Niceleyici (\forall) \rightarrow tümü / herbiri (for all / for each) anlamını içerir.

Niceleyicilerle ilgili temel eşitlikler şu şekilde sıralanabilir:

1. $\neg(\forall x)(F_x) \equiv (\exists x)(\neg F_x)$
2. $\neg(\exists x)(F_x) \equiv (\forall x)(\neg F_x)$
3. $(\forall x)(F_x) \equiv \neg(\exists x)(\neg F_x)$
4. $(\exists x)(F_x) \equiv \neg(\forall x)(\neg F_x)$
5. $(\forall x)(F_x \wedge G_x) \equiv [(\forall x)(F_x)] \wedge [(\forall x)(G_x)]$
6. $(\exists x)(F_x \vee G_x) \equiv [(\exists x)(F_x)] \vee [(\exists x)(G_x)]$
7. $(\exists x)(F_x \wedge G_x) \Rightarrow [(\exists x)(F_x)] \wedge [(\exists x)(G_x)]$
8. $[(\forall x)(F_x) \vee (\forall x)(G_x)] \Rightarrow (\forall x)[(F_x) \vee (G_x)]$

Örnek İlişkisel Hesaplama Sorguları:

SORGU 1:

“Araştırma departmanında çalışan personellerin adı ve adresi”

$$\{t.PADI, t.PADRES \mid PERSONEL(t) \wedge (\exists d) \\ (DEPARTMAN(d) \wedge d.DADI='Araştırma' \wedge d.PNO=t.PNO)\}$$

SORGU 2:

“Ankarada’ki bütün projelerin proje numarası ve projeyi kontrol eden departmanın numarası, yöneticisinin adı, doğum tarihi ve adresi”

$$\{p.PNO, p.DNO, m.PADI, m.PDTAR, m.PADRES \mid PROJE(p) \wedge \\ PERSONEL(m) \wedge p.ŞEHİR = 'Ankara' \wedge (\exists d) \\ (DEPARTMAN(d) \wedge p.DNO = d.DNO \wedge d.YPNO=m.PNO)\}$$

SORGU 3:

“5 No’lu departman tarafından kontrol edilen tüm projelerde çalışan personellerin adı”

$$\{p.PADI \mid PERSONEL(p) \wedge ((\forall x)(\neg PROJE(x)) \vee (\neg (x.DNO=5)) \vee ((\exists w) (\text{ÇALIŞTIĞI_PROJE}(w) \wedge w.PNO=p.PNO \wedge x.PROJENO=w.PROJENO))\}$$

Bu sorguyu temel bileşenlerine bölebiliriz:

$$\{p.PADI \mid PERSONEL(p) \wedge F'\}$$

$$F' = \{((\forall x)(\neg PROJE(x)) \vee F_1\}$$

$$F_1 = \{\neg (x.DNO=5) \vee F_2\}$$

$$F_2 = \{((\exists w)$$

$$(\text{ÇALIŞTIĞI_PROJE}(w) \wedge w.PNO=p.PNO \wedge x.PROJENO=w.PROJENO))\}$$

Daha doğal bir dille ifade etmek istersek, bu sorgu için;

“PROJE ilişkisinde Departman numarası 5 olan bütün x kayıtları için, ÇALIŞTIĞI_PROJE ilişkisinde $w.PNO=p.PNO$ ve $x.PROJENO=w.PROJENO$ eşitliklerini sağlayacak bir w kaydı mutlaka bulunmalıdır”

tanımlamasını yapabiliriz.