JOINING(Tabloları Birlikte Sorgulama)

İlişkisel veritabanı demek, kayıtları az yer kaplayacak kadar parçalama ve hiç parçalanmamış gibi bir arada kullanabilmek demektir.

Bu kısımda kullanılan tabloların ilişkisel yapılarını anlamak için bkz. (Diyagram_1.jpg)
İlişkisel veritabanının en temelinde birden fazla tablo üstünde birlikte işlem yapabilmek yatar. Bu sayede verilerin tekrarlaması önlenmiş olur ve sonuçta veri yönetimi kolaylaşır.

Örneğin, Kitap tablomuzda, Kitapın bir tekil numara ile listesini tutuyoruz ve odunc listesinde de bu Kitapın kim tarafından alındığının, geri getirilip getirilmediğinin kaydını tutuyoruz.
Bazen, bu iki tablodaki bilgilere de bir tek sorgu sonucu olarak ihtiyaç duyarız. Örneğin elimizde öyle bir sonuç olmalıdır ki, hangi kitabın kim tarafından ödünç alındığını bir listede görmemiz gereksin.

Bu iki tablo birbirine, kitapNo alanı ile bağlıdır. Çünkü, ödünç verilen bir kitap hakkında detaylı bilgi edinmek istediğimizde, odunc listesinden kitap numarasını alırız, daha sonra Kitap tablosundan aynı numarayı bulur ve karşılığındaki kitap hakkındaki detayları görebiliriz.

İPUCU:
Tabloların birbiri ile arasında bir ilişki tanımlanmadan ilişkisel sorgu yapılırsa, kartezyen çarpımı bulunur. Bu, '1.Tablodaki her bir kayıt ile 2. Tablodaki her bir kayıt ilişkilidir' demektir ve ilişkisel veritabanın doğasına terstir.

Örnek-44:

```
SELECT *
FROM Kitap, odunc;
```

Sorgusunun sonucunda, Kitap tablosunda 50, odunc tablosunda 100 kayıt varsa, (50*100=5000) 5000 satırlık bir bu iki tablo için kartezyen çarpım sonucu bulunur.
Klasik Join
İki tabloyu birlikte sorgulamanın en kolay yolu, asıl tablodaki birincil anahtar ile ikinci tablodaki yabancı anahtarı birbirine eşitlemektir.

```
SELECT tablo1.alan1|tablo2.alan1[tablo1.alan2,...]
FROM tablo1, tablo2
WHERE tablo1.birincilAnahtar=Tablo2.yabanciAnahtar [AND diğer şartlar]
```

Örnek-45:
Şu anda dışarıda olan Kitapın listesini alalım:

```
SELECT Kitap.*
FROM Kitap,odunc
WHERE Kitap.kitapNo=Odunc.kitapNo AND odunc.geldiMi=0;
```

kitapNo	kitapAdi	ISBNNo	turu	sayfaSayisi	kitapOzeti
1	Visual Basic.NET	0-672-32203-X	Programlama	204	Visual Basic.NET konusunda temel geçis bilgilerini içerien bir kitap.
2	Telkin ve Hipnoz ile ögrenme Teknikleri	975-6700-15-7	Kisisel Gelisim	274	ögrenme ve sinav konusunda hipnoz teknigi
3	Yatirim Plani Yapma	975-381-263-9	Is dünyasi	74	Yatirim yapmak için nereden baslamali, neler yapmali...

Tabloya Takma Ad(Alice)

Birden fazla tablo üstünde işlem yaparken, sık sık tablo isimleri kullanılır. Bu da bir kelime kalabalığı olarak karşımıza çıkar. Bunun önüne geçmek için tabloların yerine takma ad verip o adları kullanabiliriz.

İPUCU:
Takma adlar tabloların ismini kısaltmanın yanı sıra başka işlerde de kullanılır. Genellikle tablolar için takma ad seçilirken kısa olması açısından 2 veya tek harf lakaplar tercih edilir.

Örnek-46:
Dışarıda olan Kitapların adlarını,hangi tarihte alındığını ve kimde olduklarını bulalım:

```
SELECT K.kitapAdi,O.vermeTarihi,u.Adi,U.Soyadi
FROM Kitap K, odunc O,Uye U
WHERE K.kitapNo = O.kitapNo AND O.UyeNo=U.uyeNo AND O.geldiMi=0;
```

Bir Tabloyu Kendisi ile İlişkilendirme

Bir tabloyu kendisi ile ilişkilendirerek sorgulamamız gerektiğinde aynı tabloya farklı iki takma adı atayıp, farklı iki tablo üstünde ilişkili sorgu yapıyormuş gibi ilişkili sorgu yapılabilir. İyi tasarlanmış bir veritabanında bu tür ilişki pek kullanılmaz.Genellikle verilerin doğruluğunu (veya tutarsızlığını) test etmede kullanılır.

ÖRNEK-47:
Kitap tablomuzda, bazı Kitapın adları hatalı okunduğu için aynı ISBN numarası ile farklı kitap adları girilmiş olup olmadığını test etmek ve sonucu görmek istersek:

```
SELECT K1.KitapAdi,K1.ISBNNo,K2.kitapAdi,K2.ISBNNO
FROM Kitap K1, Kitap K2
WHERE K1.ISBNNo=K2.ISBNNo AND K1.KitapAdi<>K2.KitapAdi;

>>0 kayıt seçildi.
```

Çünkü ISBN numaralarında bir hata yok.
Left [Outer] Join
İki tablo arasında ilişkili sorgu yapılırken, LEFT OUTER JOIN kullanıldığında, birinci tablodaki tüm kayıtlar getirilir. 2. Tabloda ise sadece ilişkiye göre uygun kayıtlar sağ tarafına eklenir.
Genel kullanımı,
...

```
FROM tablo1 LEFT JOIN tablo2
ON tablo1.alan1 karşılaştırma tablo2.alan2;
```

Şeklindedir.

Örnek-48:

Tüm üyelerin bir listesini, yanına da ödünç aldıkları kitap varsa, kitap numarasını sıralayalım

```
SELECT uye.Adi,odunc.KitapNo
FROM uye LEFT JOIN odunc ON odunc.uyeNo=uye.uyeNo
```

Right [Outer] Join

İki tablo arasında ilişkili sorgu yapılırken, RIGHT OUTER JOIN kullanıldığında, ikinci tablodaki tüm kayıtlar getirilir. 1. Tabloda ise sadece ilişkiye göre uygun kayıtlar sağ tarafına eklenir.

Genel kullanımı,

```
...

FROM tablo1 RIGHT JOIN tablo2
ON tablo1.alan1 karşılaştırma tablo2.alan2;
```

Şeklindedir.

Örnek-49:
Ödünç alınmış Kitaplardan dışarıda olanların listesini bulunuz:

```
SELECT Kitap.kitapAdi
FROM Kitap
RIGHT JOIN odunc ON odunc.KitapNo=Kitap.KitapNo
WHERE odunc.geldiMi=0;
```

[Inner] Join

INNER JOIN en çok kullanılan tablo birleştirme yöntemidir. İki tablo arasında birleştirme yaparken, tablolardan her ikisinde de yer alan değerler seçilir, tek tabloda yer alan değerler seçilmez.

İPUCU:
LEFT JOIN ve RIGHT JOIN, aslında birer OUTER JOIN ifadeleridir. Bir çok sistem OUTER ifadesinin yazılmasını isterken, MS temelli sistemlerde kod yazıyorsanız, böyle davranmak zorunda değilsiniz.

Aslında, Klasik Join ile INNER JOIN aynı şeydir. Sadece birleştirmede kullanılan teknik farklıdır.

```
...

FROM tablo1 INNER JOIN tablo2
ON tablo1.alan1 karşılaştırma tablo2.alan2;
```

Örnek-50:
LEFT JOIN örneğindeki sorguyu INNER JOIN ile çalıştıralım. Bu durumda, hiç kitap ödünç almayan üyelerin sonuçta yer almadığına dikkat edin.

```
SELECT uye.Adi,odunc.KitapNo
FROM uye INNER JOIN odunc ON odunc.uyeNo=uye.uyeNo
```

Adi	KitapNo
Irem	1
Yusuf	5
Irem	1
Tugay	6
Irem	1
Tugay	2
Yusuf	3

Sadece iki tabloyu JOIN edilebilir diye bir kısıtlama yoktur. İkiden fazla tabloyu da ihtiyacımıza göre JOIN edebiliriz.

Örnek-51:
Bu güne kadar kimin hangi kitabı ödünç aldığının dökümünü bulmak istersek:

```
SELECT Kitap.kitapAdi,uye.Adi,uye.Soyadi
FROM (odunc INNER JOIN Kitap ON Kitap.KitapNo=Odunc.kitapNo)
INNER JOIN uye ON uye.UyeNo = odunc.uyeNo;
```

kitapAdi	Adi	Soyadi
Visual Basic.NET	Irem	AKYOL
Hayat Yolunda zorluklarla Mücadele	Yusuf	GÖZÜDELI
Visual Basic.NET	Irem	AKYOL
Is Hayatinda Motivasyon	Tugay	ERYATAN
Visual Basic.NET	Irem	AKYOL
Telkin ve Hipnoz ile ögrenme Teknikleri	Tugay	ERYATAN
Yatirim Plani Yapma	Yusuf	GÖZÜDELI

Bazı sistemlerde kodlama yaparken, INNER ifadesi yer almamalıdır. Aksi halde sorgunuz çalışmayabilir. Bu farklılığın nedeni, JOIN ON ifadesi ANSI-SQL'in bir parçası değildir. Ancak VTYS üreticileri, WHERE şartında yer alan diğer koşullar ile, tablo ilişkilendirme koşullarının bir birinden kolayca ayırt edilmesi için standartlarda yer alan klasik join ile birlikte bu türden ifadeleri de desteklemektedirler.