

# Veritabanı Yönetim Sistemleri



## **Hedefler**

- İlişkisel veri tabanlarında kullanılan tablolar, alanlar, kayıtlar, anahtarlar ve ilişkiler ile ilgili genel bilgi sahibi olmak



## İçindekiler

1. İlişkisel Veritabanı
  - 1.1. Tablo
  - 1.2. Alan
  - 1.3. Kayıt
  - 1.4. Anahtar
    - 1.4.1. Birincil anahtar (Primary key - Ana anahtar)
    - 1.4.2. Aday anahtar (İndeks)
    - 1.4.3. Yabancı anahtar (Foreign key)
  - 1.5. İlişki
    - 1.5.1. Bire bir ilişki
    - 1.5.2. Bire çok ilişki
    - 1.5.3. Çoka çok ilişki

## 1. İlişkisel Veritabanı

İlişkisel veri tabanı modeli ilk defa 1970 yılında Dr. E.F. Codd tarafından ortaya atılmıştır. Model, veri tabanına kaydedilmiş bilgilerin belli kurallara uymasını kapsar. İlişkisel veri modeli, ilişkisel veri tabanı dillerini ortaya çıkarmıştır.

İlişkisel veri tabanlarının iki önemli özelliği vardır:

-  Bu sistemde veriler tablolar halinde tutulurlar.
-  Veri alışverişi tabloları kullanarak özel işlemlerle yapılır.

İlişkisel veri tabanını iyi bir şekilde tasarlamak için temel özelliklerinin iyi bilinmesi gerekir.

### 1.1. Tablo

Bir ilişkisel veri tabanında bilgiler tablolar halinde saklanır. Yani veri tabanının alt yapısı tablolardan oluşur. Tablo belli bir amaç ve belli bir konudaki verileri içeren bir taşıyıcıdır. Mesela tablolar, öğrenci bilgileri, müşteri bilgileri, stok bilgileri, sipariş bilgileri, vb. gibi verileri saklayan bir taşıyıcıdır. Bir veri tabanında birden fazla tablo bulunabilir.

Bir ilişkisel veri tabanı tasarımında en önemli kısım tabloların tasarımıdır. Tabloları tasarlarken onlara verdiğimiz özellikler ve yaptığımız sınıflandırma veri tabanı performansı ve ilerde alınacak olan raporlar ve sorguları da etkiler. Verilere erişimi de doğrudan etkiler. Bunun için tabloların tasarımı sırasında tablolarda kullanılacak olan alanların birbiriyle ilgili olmasına dikkat etmemiz gerekir.

Bir tablonun tasarımı yapılırken veri tekrarı olmayacak şekilde tasarlanmasına dikkat edilmelidir. Bu veri tekrarı, verilerin saklanması etkinliğini azalttığı gibi veri hatalarına da yol açar. Bir tablo içerisinde bütün bilgileri toplamaya gerek yoktur. Mümkünse veriler birbirleriyle olan ilgilerine göre küçük küçük tablolara bölünür. Mesela; öğrenci, ders ve bölüm bilgilerini bir tabloda tutup gereksiz veri girişi arttırmaktansa bu verilerin her birini ayrı ayrı tablolarda tutarak hem veri tekrarıdan kurtulmuş oluruz hem de daha tutarlı bir yapıya ulaşmış oluruz.

Örnek olarak daha önceden oluşturulmuş olan mobilya sipariş tablosunu inceleyelim.

Siparişler						
Sipariş No	Mobilya Cins	Mobilya Ebat	Mobilya Renk	Mobilya Fiyat	Müşteri Ad	Müşteri Adres
1201	Masa	80x60x80	Kahverengi	400	Selim	İstanbul
1245	Sandalye	40x80x70	Siyah	100	Selim	İstanbul
1254	Masa	80x60x80	Kahverengi	400	Meltem	Gaziantep

Yukarıdaki oluşturulmuş olan bu tablo hatalı bir sipariş tablosudur. Müşterinin özellikleri ve mobilyanın özellikleri her sipariş kaydında tekrar etmiştir. Bu tekrarı ortadan kaldırmak için bu tablo mobilya ve müşteri tablosu diye iki tabloya ayrılmıştır. Böylece veri tekrarını azaltmış ve verilerin etkin bir şekilde kullanımı sağlanmıştır.

Mobilyalar				
Mobilya Kod	Mobilya Cins	Mobilya Ebat	Mobilya Renk	Mobilya Fiyat
002	Masa	80x60x80	Kahverengi	400
005	Sandalye	40x80x70	Siyah	100

Müşteriler		
Müşteri Kod	Müşteri Ad	Müşteri Adres
A74	Selim	İstanbul
A47	Meltem	Gaziantep

Artık sipariş tablosu olması gerektiği gibi oluşturulabilir. Yukarıdaki iki tablodan yararlanarak aşağıdaki doğru sipariş tablosunu oluştururuz.

Siparişler		
Sipariş No	Mobilya Kod	Müşteri Kod
1201	002	A74
1245	005	A74
1254	002	A47

## 1.2. Alan

Veri tabanında ilgili bilgiler tablolara göre tanımlandıktan sonra bu tabloda kullanılacak bilgilerin özellikleri tanımlanır. Tablolar dikey olarak alanlardan oluşmuştur.

Tabloların alanlarının tanımlanması için ilk önce tabloların özellikleri ve tablodan beklentilerin belirlenmesi gerekmektedir. Tablonun alanları tabloda yer alan her bir kaydın ilgili kısmının tutulduğu yerdir. Bu nedenle burada saklanacak veri düşünülerek karar verilmelidir.

Tablonun her bir alanının, mutlaka tablo ile mantıksal olarak ilgili olması gerekmektedir. Tabloda gereksiz alanlara yer verilmemelidir. Alanlara verinin en temel hali konulur. Hesaplamalar burada yapılmaz. Bu yüzden tasarım yaparken dikkatli olmak veri tabanının performansını artırır.

### 1.3. Kayıt

Tablolar yatay olarak kayıtlardan oluşmuşlardır. Kayıtlar alanlarda tanımlanan özelliktedirler. Verinin asıl temsil edildiği ve verinin saklandığı yer burasıdır. Sorgulamalarda ve aramalarda buranın içeriğine bakılarak karar verilir. Örnek olarak tablo alan ve kayıtları aşağıdaki gibidir.

	Alan 1	Alan 2	Alan 3	Alan 4
Kayıt 1	Kayıt 1 Alan 1	Kayıt 1 Alan 2	Kayıt 1 Alan 3	Kayıt 1 Alan 4
Kayıt 2	Kayıt 2 Alan 1	Kayıt 2 Alan 2	Kayıt 2 Alan 3	Kayıt 2 Alan 4

### 1.4. Anahtar

Bir veri tabanının etkin bir şekilde çalışabilmesi için tablonun alanlarından bir veya kayıtları tek başına temsil edebilecek birkaç alan kullanılarak anahtarlar oluşturulur. Bu anahtarlar kayıtların sıralanmasında, aranmasında ve ilişkilerin kurulmasında yardımcı olur. Bunun dışında anahtarlar alanlara çift bilgi girilmesini de önlerler.

Bir sistemin gücü, içerisindeki bilgilerine doğru ve hızlı bir şekilde erişimiyle ve farklı tablolardaki bilgileri bir araya getirmesiyle görülür. İlişkisel veri tabanları bunu tanımlanmış olan anahtarlarıyla gerçekleştirir. Anahtarlar yukarıda anlatılan görevlerini yerine getirebilmek için veri tabanı içinde ayrı bir biçimde saklanır.

Mobilyalar				
Mobilya Kod	Mobilya Cins	Mobilya Ebat	Mobilya Renk	Mobilya Fiyat
002	Masa	80x60x80	Kahverengi	400
008	Askılık	150x15x20	Siyah	300
011	Sandalye	70x80x70	Kahverengi	120
001	Masa	80x120x160	Siyah	1500
005	Sandalye	40x80x70	Siyah	100

Yukarıdaki tabloda görüldüğü gibi, mobilya kayıtları içinde bir mobilyayı diğerlerinden ayıran en önemli özellik mobilya kodu olduğundan, birincil anahtar mobilya kodudur. Numarasını öğrendiğimiz bir mobilyanın diğer özelliklerini de buradan öğrenebiliriz.

Mobilya kodu diğer alanlardan farklı olarak hiçbir şekilde tekrar edilmemektedir. Yani tek (**unique**) bir veridir. Ama diğer alanlar ise bir veya daha fazla sayıda tekrar edilebilmektedir. Bu yüzden tek başlarına birincil anahtar olamazlar. İki veya daha fazlası bir araya gelerek birincil anahtarlar oluşturabilirler fakat pek kullanılan bir yöntem değildir.

Eğer bir tabloda belli bir alana göre sıralama yapmak istiyorsak, mesela; az önceki örnekteki gibi mobilya genel tablosunda kayıtların mobilyanın cinsine göre sıralama yapılmak isteniyorsa anahtar alan buna göre düzenlenmelidir. Anahtar alan üzerindeki bilgiye daha çabuk ulaşılır. Veri tabanlarında bir kere bir anahtar seçtikten sonra her seferinde anahtar tanımlamamıza gerek yoktur.

Eğer bir tabloda, bir alana göre sıralatma yapıldıktan sonra başka bir alana göre de sıralatma yapılmak isteniyorsa, bunun için yeni anahtarlar tanımlayabiliriz. Bir anahtar seçmişsek veri tabanına girdiğimiz veriler bu veriler arasında sıralanmış şekilde görülür.

Fakat veri tabanı verileri bu sıralamada saklamaz ve sadece biz görmek istediğimizde bize bu şekilde gösterir. Yani en son girdiğimiz bir kaydı en altta göremeyebiliriz. Bir anahtar girmemişsek verileri girdiğimiz sırada görürüz.

#### 1.4.1. Birincil anahtar (Primary key - Ana anahtar)

Birincil anahtar, tablodaki kaydı tek başına temsil eden bir veridir. Eğer bir alan belli bir alana göre sıralanacaksa, çift girişler önlenmek isteniyorsa ve verilere daha çabuk ulaşmak için birincil anahtar oluşturulur. Diğer tabloların alanlarıyla ilişkilendirmeler yapmak içinde birincil anahtarlar kullanılır.

**Bir tabloda sadece bir tane anahtar olmayabilir. Fakat birincil anahtar bir tanedir.**

Birincil anahtar birden fazla alanın birleştirilmesiyle oluşturulmuş olabilir. Mesela Mobilya genel tablosundaki "Mob Kod" ile "Mob Cins" alanları birlikte tek bir birincil anahtar olarak seçilebilir. Bu şekilde oluşturulmuş anahtarlara "Birleştirilmiş Anahtar" denir. Bu yöntem genelde bir tabloda kayıtları tek başına temsil eden bir alan bulunmadığı zaman kullanılır.

Aslında birincil anahtarların bu sıralama işinden daha önemli işleri vardır. Çünkü ilişkisel veri tabanları devamlılıklarını sürdürebilmek için bazı kuralları sağlamak zorundadırlar. Bu kuralları da anahtarlar yardımıyla gerçekleştirirler. Birincil anahtarların bazı özellikleri vardır.

Anahtar alanda yer alan veri, tablonun o alanı içinde tek (unique) olması gerekmektedir. İlişkisel veri tabanlarının en önemli özelliklerinden biri tekrarlanan kayıtların bulunmamasıdır. Aynı kaydın tabloya tekrar girişini önlemek için birincil anahtarlar kullanılır.

Anahtar alan içinde veri tekrar etmiyor fakat istisnai durumlar varsa bu durumda en iyi çözüm esas anahtar yerine geçecek olan bir birleşik anahtar oluşturmaktır. Bir tabloda bazı alanları boş bırakabiliriz, fakat birincil anahtarı boş olarak geçemeyiz, çünkü böyle yaparak birincil anahtarın tek olma kuralını bozmuş oluruz.

#### 1.4.2. Aday anahtar (İndeks)

Daha önce de bahsedildiği gibi, bir tabloda birincil anahtar sadece bir tanedir. Aday anahtar sayısı ise birden fazla olabilir. Aday anahtar bir alanın istenen kaydına daha çabuk erişmek ve kaydı sıralı bir şekilde görmek için kullanılır. Birincil anahtara göre yapılan tüm

işlemler aday anahtar için de geçerlidir. Fakat aday anahtarlarda veri tekrarı olabilir. Aday anahtar kullanmak sıralamayı kolaylaştırırken veri girişini bir miktar yavaşlatır.

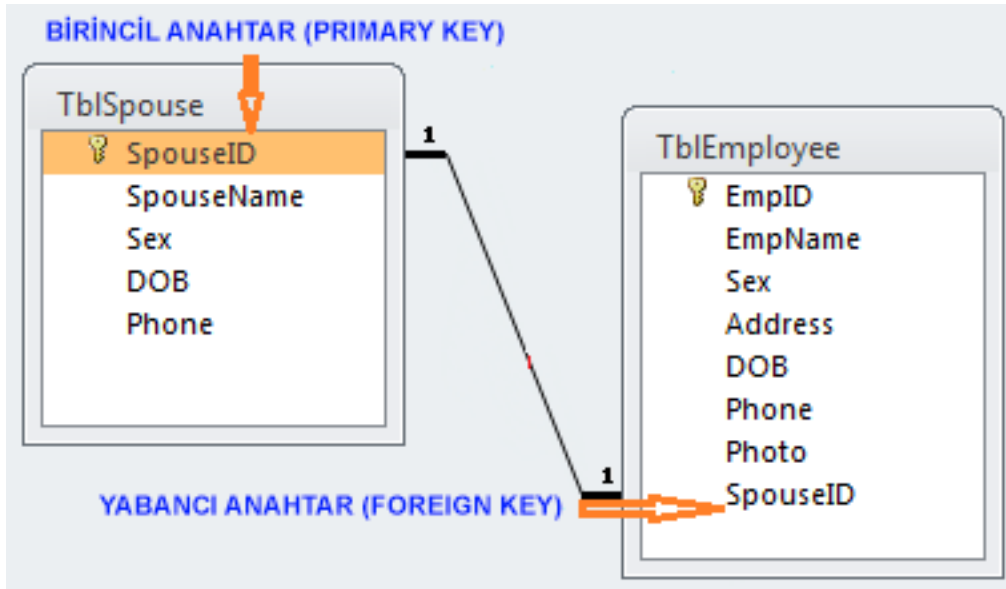
Çok sık kullanılan alanlar için aday anahtar oluşturmak her zaman yararlıdır. Fakat fazla sayıda aday anahtar oluşturarak veri tabanını yavaşlatmanın da bir anlamı yoktur.

### 1.4.3. Yabancı anahtar (Foreign key)

Bir tabloda olan bir anahtar alan başka bir tabloda da yer alıyorsa, yani tablodaki bir alan başka bir tablodaki anahtar alanı gösteriyorsa buna "yabancı anahtar" denir. Yabancı anahtar genelde diğer tablolarla ilişki kurmak için kullanılır.

Bir yabancı anahtarın, diğer tablolardaki birincil anahtarlarla aynı özelliklere sahip olması gerekmektedir. Farklı iki tablodaki iki ayrı alanın aynı adı ve aynı özellikleri taşıması yabancı anahtar olacak anlamına gelmez.

Bir tablodaki birincil anahtarda meydana gelen bir değişiklik onunla eşleşmiş olan yabancı anahtarların da değişmesine neden olur. Bu sayede veri tutarlığı sağlanmış olur.



### 1.5. İlişki

Tablolar arası bağlantı ilişkiler aracılığıyla sağlanır. İlişkiler iki tabloda bulunan ortak alanlarla yapılır. İki tabloyu ilişkilendirmek ilişkisel veri tabanının en temel çalışma şeklidir. Böylece farklı tablolardaki bilgilerin birbiriyle etkili biçimde kullanılması sağlanmış olur. Ayrıca herhangi bir tablodaki yapılan değişiklik diğer tabloları da anında etkiler.

Veri tabanı içindeki ilişkilerin tanımlanması tabloların birçoğundan yararlanarak veri bütünlüklerinin oluşturulması sağlar. Birçok rapor ve veri giriş ekranında birden çok tablodan yararlanılır. İlişkiler kurularak tablolar arasındaki verilere kolaylıkla erişilir. Tablolar arasındaki verileri bağlamak için çeşitli ilişki türleri vardır.

### 1.5.1. Bire bir ilişki

Bire bir ilişkilerde bir tablodaki bir kayda diğer tabloda sadece bir kayıt karşılık gelir. Aslında bu tür ilişkiler, tablolar arası ilişkilerden daha çok bir tablo diğer tablonun devamı gibidir.

Bire bir ilişkiler çok sık kullanılan bir ilişki türü değildir. Güvenlik dışında bir neden yoksa bu iki tabloyu birleştirmek mümkündür. Böyle bir ilişkide bir tablo diğerinin içindeymiş gibi kullanılır.

Örnek olarak mobilya bilgileri için oluşturulan tabloları inceleyelim.

Mobilyalar				
Mobilya Kod	Mobilya Cins	Mobilya Çeşit	Mobilya Ebat	Mobilya Renk
002	Masa	Çalışma	80x60x80	Kahverengi
008	Askılık	Çalışma	150x15x20	Siyah
011	Sandalye	Oturma	70x80x70	Kahverengi
001	Masa	Çalışma	80x120x160	Siyah
005	Sandalye	Oturma	40x80x70	Siyah

Mobilya Fiyatları		
Mobilya Kod	Mobilya Satış Fiyatı	Mobilya KDV Oranı
002	400	%18
008	300	%18
011	120	%8
001	1500	%18
005	100	%8

Yukarıdaki iki tablo arasında “Mobilya Kod” alanı sayesinde bire bir ilişki kurulabilir.

### 1.5.2. Bire çok ilişki

Bire çok ilişkiler, ilişkisel veri tabanında en çok kullanılan ilişki çeşididir. Bu ilişki temelde bire bir ilişkiye benzer fakat burada bir tablonun bir kaydına diğer tabloda birden fazla kayıt karşılık gelir.

Bire çok ilişkide klasik olarak bir kaydın ona bağlı sahip olduğu alt kayıtları tutarlı biçimde kontrol etmeye karşılık gelen bir yapıdır. Bu tür ilişkide bir tablonun birincil anahtarına karşılık diğer tablodaki eşdeğer yabancı anahtar denk gelir.



Bire çok ilişkiyi bir örnek üzerinden inceleyelim. Bir fakülteye bağlı birden fazla bölüm olabilir. Ancak bir bölüm birden fazla fakülteye bağlı olamaz, sadece bir fakülteye bağlı olabilir.

Bu tür durumlarda iki tablo arasında bire çok ilişki kurulur.

Fakülteler	
Fakülte Kod	Fakülte Ad
01	Mühendislik Fakültesi
02	İşletme Fakültesi
03	Bilgisayar ve Bilişim Bilimleri Fakültesi

Bölümler		
Bölüm Kod	Fakülte Kod	Bölüm Ad
101	03	Bilgisayar Mühendisliği
102	03	Yazılım Mühendisliği
103	01	İnşaat Mühendisliği
104	01	Makine Mühendisliği
105	02	Yönetim ve Bilişim Sistemleri
106	01	Elektrik-Elektronik Mühendisliği
107	02	İnsan Kaynakları Yönetimi

Yukarıdaki iki tablo arasında “Fakülte Kod” alanı sayesinde bire çok ilişki kurabiliriz.

### 1.5.3. Çoka çok ilişki

Bu tür ilişkide birinci tabloda yer alan çok sayıda kayda ikinci tabloda da yine çok sayıda kayıt karşılık gelir.

Çoka çok ilişkiler bire bir ya da bire çok ilişkilerde olduğu gibi doğrudan değil, dolaylıdır. Bu yüzden anlaşılması diğer iki ilişki türüne göre biraz daha zor olabilir.

Bu ilişki dolaylıdır çünkü iki tablo arasında çoka çok ilişki kurmak için üçüncü bir tablo oluşturmak gerekir. Birinci tablo ve ikinci tablo ile bu yeni oluşturulan tablo arasında bire çok ilişkiler kurulur.

Aslında bakıldığında iki tane bire çok ilişki kurulmuştur, ancak bu sayede dolaylı yoldan çoka çok ilişki kurulmuş olur.

Çoka çok ilişkiler örnek üzerinden daha kolay anlaşılacaktır.

Bir okul veritabanında Öğrenciler ve Veliler adında iki tablonun bulunduğunu varsayalım. Bir öğrencinin birden fazla velisi (annesi, babası, vs.) olabilir. Aynı zamanda bir

veli birden fazla öğrencinin velisi olabilir. Böyle bir durumda bu veritabanındaki Öğrenciler ve Veliler tabloları arasında çokla çok ilişki olmalıdır.

Aşağıdaki tablolardan da görülebileceği gibi, bu iki tabloda ortak hiçbir alan bulunmamaktadır. İlişkiler ortak alanlar üzerinden kurulabildiğine göre, zaten üçüncü bir tablo oluşturmadan bu iki tabloyu birbiriyle ilişkilendirmenin bir yolunun olmadığı görülecektir.

Öğrenciler		
Öğrenci Kod	Öğrenci Ad	Öğrenci Soyad
1	Yasemin	Kaya
2	Fatma	Deviren
3	Neşe	Saydam
4	Hatice	Saydam
5	Ayşe	Kökmen

Veliler		
Veli Kod	Veli Ad	Veli Soyad
1	Ali	Kökmen
2	Serhat	Kaya
3	Hakan	Saydam
4	Mehmet	Deviren
5	Elif	Deviren

Bu iki tabloyu ilişkilendirmek için velilerle öğrencilerin ilişkisini gösteren aşağıdaki üçüncü tablo oluşturulmalıdır.

Öğrenci Velileri		
Kod	Öğrenci Kod	Veli Kod
1	1	2
2	2	4
3	2	5
4	3	3
5	4	3
6	5	1

**Kaynaklar**

[http://www.worldbestlearningcenter.com/index\\_files/Access-2010-table-relationship-relationships.htm](http://www.worldbestlearningcenter.com/index_files/Access-2010-table-relationship-relationships.htm)