

Linux Kullanıcıları Derneđi  
Düzenli Seminerleri

# VERİTABANLARINA GİRİŞ

*Haziran, 2002  
ANKARA  
Güncelleme : Şubat 2006*

**Devrim GÜNDÜZ**  
LKD, Kivi Bilişim Teknolojileri

<http://www.gunduz.org>  
[devrim@gunduz.org](mailto:devrim@gunduz.org)



# Veritabanı nedir?

<http://www.m-w.com> adresindeki Merriam-Webster sözlüğünde bir veritabanı :

*“a usually large collection of data organized especially for rapid search and retrieval (as by a computer)”*

olarak tanımlanır.

Bir veritabanı yönetim sistemi (DBMS: DataBase Management System), kitaplıklar uygulamalar ve yardımcı programların birleşmesinden oluşur ve verilerin saklanması ve yönetilmesi ile ilgili konulardaki ayrıntılardan veritabanı yöneticilerini kurtarır. Aynı zamanda, kayıtların güncellenmesi ve kayıtlar üzerinde araştırma yapılması da mümkündür.

Veritabanı yönetim sistemleri zamanla çok daha fazla özellikler kazanmaktadırlar.

- Belirli bir tarzda organize edilmiş bilgi “koleksiyon”udur.
- En az bir tablodan oluşmak zorundadır.
- Veritabanının içindeki tablolar ise veri alanlarından oluşur (data field).
- Kitaplıklar, uygulamalar ve yardımcı programların birleşmesinden oluşur.
- Verilerin saklanması ve yönetilmesi ile ilgili konulardaki ayrıntılardan veritabanı yöneticilerini kurtarır.
- Kayıtların güncellenmesi ve kayıtlar üzerinde araştırma yapılması da mümkündür.

## Veritabanı türleri

- a. Hiyerarşik Veritabanı
- b. İlişkisel Veritabanı (İlişkisel Type)
- c. Nesnesel Veritabanı

Bu yazıda, günümüzde en yaygın kabul edilen ilişkisel veritabanı modeli anlatılacaktır.

# İlişkisel Veritabanı (Relational Database) Modeli

Veritabanı yönetim sistemleri teorisi 1970'de "A İlişkisel Model of Data for Large Shared Data Banks", (E. F. Codd) yazısı ile büyük bir ilerleme kaydetmiştir. Bu devrim niteliğindeki yazıda, Codd ilişki mantığını tanıtmış ve tabloların gerçek dünyadaki nesnelere göstermekte, nitelendirmekte nasıl kullanılabileceğini belirtmiştir.

İlişkisel veritabanılarını tanımlayan bazı önemli kurallar vardır:

Öncelikle, tablolardaki kayıtlar matematiksel açıdan tuple olarak tanımlanırlar. Bir tuple tanımlanmış bir veri tipi olan bileşenlerden oluşan sıralı grup olarak tanımlanır. Tüm tuplelar aynı sayıda ve tipte bileşenlerden oluşur. Örnek vermek gerekirse,

{ "10", "Veritabanlarına Giriş", "2002-06-12" }  
{ "11", "Progress Veritabanı Sunucusu", "2002-06-26" }

Örneğimizdeki her bir tuple da 3 bileşen bulunmaktadır:

- Ankara'daki 2002 yılındaki kaçıncı seminer olduğu (integer)
- Seminerin konusu (char)
- Seminerin tarihi (timestamp)

İlişkisel veritabanlarında bu "küme" ya da tabloya eklenen tüm kayıtlar aynı yapıda olmalıdırlar, dolayısıyla aşağıdakilere izin verilmez:

- { "Veritabanlarına Giriş", "2002-06-12" }
  - *eksik bileşen*
- { "10", "Veritabanlarına Giriş", "2002-06-12", "Devrim GÜNDÜZ" }
  - *fazla bileşen*
- { "2002-06-12", "Veritabanlarına Giriş", "10" }
  - *yanlış bileşen tipleri (yanlış sırada)*

Bu aşamada, row ve record tanımını yapmak yerinde olacaktır. Row (record, kayıt) , bir tabloda, üstte gördüğümüz her bir satıra, kümeyle verilen addır.

Ayrıca tuple lardan oluşan bir tabloda aynı veriler bulunmaz.(no duplicate record). Dolayısıyla ilişkisel veritabanlarındaki herhangi bir tabloda birbiriyle tamamen aynı iki kayıt (row or record) bulunamaz.

Bu, çok gereksiz bir sınırlama olarak görünebilir. Örnek vermek gerekirse, aynı kullanıcının aynı malı iki kez sipariş etmesi görünürde engellenmiştir. Bunu da tabloya bir bileşen ekleyerek çözebilirsiniz.

Bir kayıttaki her bir bileşen “atomik”, yani bir veri olmalıdır; başka bir kayıt ya da diğer bileşenlerin listesi olamaz. Aynı zamanda, daha önce de belirttiğimiz gibi, tablodaki bileşenlerin veri tipleri de üsttekilerle ve dolayısıyla tablo tanımlarındakiyle aynı olmalıdır. Diğer bir deyişle, veritabanı tarafından desteklenen veri tiplerinden biri olmalıdır.

Birbiriyle eş kayıtları ayırmak için kullanılan bileşenlere **key** denir.

Tablodaki bir kaydı diğer tüm kayıtlardan ayırmak için kullandığımız bileşene, **primary key (birincil anahtar)** adı verilir. Primary key, o kaydı tekil (unique) yapar. Tüm ilişkisel veritabanlarında her bir tablo ya da ilişkide mutlaka primary key olmalıdır.

İlişkisel veritabanı yapısını belirleyen son bir kural da referential integrity (bütünlük denetimi). Veritabanında uygulama geliştiren programcılar yazdıkları kodun veritabanının bütünlüğünü bozmamasına dikkat etmelidirler. Uygun bir örnek bir müşteri kaydının silinmesi olabilir. Eğer müşteri tablosundan bir kayıt silinecekse, o kişinin sipariş tablosundaki siparişlerinin de silinmesi gerekir. Aksi takdirde olmayan bir müşteriye ait siparişler veritabanında kalacaktır.

## Tabloların yapılandırılması

- Aynı konu ile ilgili olan bilgilerin belirlenmelidir.
- Olası olan en yüksek seviyede yapısal bir şekilde tabloların oluşturulması sağlanmalıdır
- Aynı konu ile ilgili alanların kendi tablolarında aynı tabloda toplanması sağlanmalı
- Veri tekrarı olmamalı
- Gereksiz alanlar kullanılmamalı
- Alanlar basite indirgenmelidir.

# SQL Nedir?

SQL (Structured Query Language), bir veritabanı dilidir. Program geliştiricileri ve Veritabanı kullanıcıları, bir veritabanına veri eklerken, silerken, güncellerken veya sorgularken bu dili kullanırlar. Hem ANSI hem de ISO standardı olmasına rağmen, çoğu veritabanı programı standart dillere ekleme yaparak bu dili kullanmaktadırlar.

SQL'in kullandığımız bir kaç sorgulama temel komutu vardır. Bunların en basitleri:

```
SELECT (veri seçerken kullanılır)
DELETE(veri silerken kullanılır)
UPDATE (veri güncellerken kullanılır)
INSERT (veri girerken kullanılır)
```

## Örnekler:

```
SELECT * FROM tablo1 ; // tablo1 adlı tablodaki tüm verileri seçer.
SELECT * FROM tablo1 WHERE adi='Devrim'; // tablo1 tablosundan, adi
kısmında Devrim yazan tüm kayıtları getirir.
SELECT * FROM tablo1 WHERE adi LIKE '%evrim'; // tablo1 tablosunda,
adının baş kısmı ne olursa olsun, sonu evrim olanları listeler.
DELETE FROM tablo1; // tablo1 tablosundaki tüm verileri siler.
DELETE FROM tablo1 WHERE adi='Devrim'; // Adı alanında Devrim olan
tüm kayıtları siler.
UPDATE tablo1 SET adi='Anıl'; // tablo1 tablosundaki tüm kayıtların adi
alanını Anıl ile değiştirir.
UPDATE tablo1 SET adi='Anıl' WHERE adi='Devrim'; // tablo1
tablosundaki adı kısmında Devrim yazan tüm kayıtları Anıl ile değiştirir.
INSERT INTO tablo1 (adi,soyadi) VALUES ('Devrim','Gündüz'); // tablo1
tablosuna, Devrim Gündüz satırını ekler ( {'Devrim', 'Gündüz'} )
```

## Neden Veritabanı?

Gerçekten veritabanına gereksinmeniz var mı? En başta bu soruya yanıt vermeniz gerekir. Veritabanları, üstte de belirtildiği gibi, verilerin saklanması ve yönetilmesi için kullanılmalıdır. Küçük bilgiler için metin dosyaları yeterli olacaksa bunun için veritabanı kullanılması gerekemeyebilir. Bir kurumun tüm belgeleri üzerinde arama/sorgu yapılacaksa bunun için de veritabanı kullanmanız gerekemeyecektir, bir web indeksleme programı da bunun için yeterli olabilir. Ancak, bir e-ticaret sitesinde müşterilerin bilgilerinin tutulması söz konusu ise, ya da bir kurumun değişik birimlerine kullanıcı/müşteri/personel giriş/çıkış bilgileri tutulacaksa bunun için verinin

büyüklüğüne ve çalışacağınız platforma uygun bir veritabanı sunucusunu kullanmak gerekecektir.

Aşağıdaki sorulara vereceğiniz yanıtlar size belirli konularda ışık tutacaktır:

<b>Veri sadece bir konuyu içeren bir listenin içinde mi?</b>	“flat file” olarak adlandırılırlar. Eğer sadece bir konu başlığınız varsa bu yazıda bahsedilen “ilişkisel veritabanlarına” bakın ve sadece bir tablo kullanın.
<b>Sorun karmaşık mı?</b>	Değişik konulardaki veriler birden fazla tablolarda saklanabilir. Bunun için ilişkisel veritabanı gereklidir. Bunlardan birini seçebilirsiniz.
<b>İstatiksel bir analiz mi yapmak istiyorsunuz?</b>	Bunun için en güçlü teknik veritabanını veriyi kontrol etmek ve yönetmek için kullanırken, SPSS gibi bir yazılımı da veritabanının yanında analiz için kullanmaktır. Örnek vermek gerekirse, SPSS, Access tablo ya da sorgularını okuyacaktır.
<b>Bir yönetim mi yapacaksınız?</b>	Yönetim amaçları için konular biraz farklıdır – ancak genel prensipleri uygulayabilirsiniz.
<b>Metinsel veritabanları</b>	FileMaker Pro gibi yazılımlar, metinsel bilgileri daha rahat toparlayabilirler.
<b>Kullanım kolaylığı</b>	Bir veritabanının kullanım kolaylığı
<b>Bilimsel formüllere gereksinmeniz olacak mı?</b>	Veriyi veritabanında tutmak, ancak bunu analiz etmek için başka programlar kullanmak isteyebilirsiniz.
<b>Veriyi paylaşma gereksinmeniz olacak mı?</b>	Verinizi bir ağ üzerinde sunacak mısınız? Veriyi aynı zamanda webde de sunmak mümkündür.
<b>Veriyi webde sunacak mısınız?</b>	İnternet üzerinden veritabanına ulaşma, hızla ilerleyen bir alandır. Hemen hemen tüm çözümler programlama becerileri gerektirirler. Ancak, bunun yanında bazı programlar bu konuda kullanıcıya yardımcı olabilmektedir. Bu yazılımlarla verinizi geliştirebilir ve web ucu ile verinizi webde sunabilirsiniz.

# Veritabanı çeşitleri

Kendinize bir veritabanı seçmeden önce, bu veritabanı ile ne yapacağınıza karar vermelisiniz. Bunun için aşağıdaki basit soruları yanıtlamakla işe başlayabilirsiniz:

1. Bu veritabanı ile neler yapacaksınız? Küçük bir şirket çalışanlarının özel bilgileri mi tutulacak, yoksa büyük bir şirketin binlerce müşterilerinin bilgileri mi?
2. Sitenizi günde kaç kişi ziyaret edecek?
3. Aynı anda kaç işlem yapılacak?
4. Güvenlik ne ölçüde olacak?
5. Verilerinizin güvenliği ne ölçüde olacak?

Genel, genel olduğu kadar da yanlış bir kanı vardır: *“Paralı ürünler iyidir, ücretsiz ürünler iyi değildir!”* Linux, bu tezi çürüten, bilgisayar sektöründeki son yıllardaki en iyi konudur. Dolayısıyla, bir veritabanının ücretsiz olup olmamasından çok işinizi görüp görmeyeceği önemlidir.

Birkaç veritabanını inceleyelim:

## 1. Microsoft Access

Microsoft firmasının Office paketi içinden çıkan Access, paralı veritabanları arasında nispeten ucuz olarak göze çarpar. Küçük ölçekli uygulamalardaki gereksinimlerinizi karşılayabilir. Eğer bir web sitesinde veri miktarı ve aynı anda yapılan işlem sayıları az ise, Access kullanabilirsiniz. Tek bir veri tablosunda 2 GB a kadar veri depolayabilir ve aynı anda 255 bağlantıya izin verebilirsiniz. Access, MS Windows sistemlerinde kullanılamamakta, bu da yaygınlaşmasını engellemektedir. “Transaction locking” özelliğine sahiptir, ancak “trigger” ve “stored procedure” özelliklerine sahip değildir.

## 2. MySQL

MySQL Inc. Tarafından kodlanan MySQL, Access ile karşılaştırıldığında daha güvenlidir. Windows’un yanı sıra Linux, OS/2, Solaris, AIX ve birçok işletim sistemini desteklemesi nedeniyle çok yaygındır.. Ev kullanıcıları tarafından, kolay kurulumu ve gelen kurulum paketleri nedeniyle sıkça tercih edilmektedir. Tablo başına 8 TB veri depolayabilmektedir. MySQL’ in en büyük dezavantajlarından biri ücretsiz olmasından kaynaklanan destek eksikliğidir. Özellikle web uygulamaları için çok hızlıdır. Transaction’ ları desteklemediği için alabildiğine yalındır ve transaction desteği olmadan gerçekleştirilebilecek web uygulamaları için çok hızlı bir alternatiftir. Oldukça fazla yazılım desteği bulunmaktadır.

### 3. **IBM DB2**

IBM firmasının ürünü olan DB2, Access ve MySQL e göre daha performanslı, ancak küçük işletmelere göre daha yüksek maliyete sahiptir. Windows ve \*nix sistemlerinde çalışabilir.

“Transaction locking”, “trigger” ve “stored procedure” özelliklerine sahiptir.

### 4. **Informix**

Ücretli ve güçlü bir veritabanıdır. Orta ölçekli işletmelerin yükünü kaldırabilecek kapasitededir. 1994'deki Postgres kodundan geliştirilmeye başlanmıştır.

### 5. **Microsoft SQL Server**

Yine Microsoft firmasının bir ürünü olan Microsoft SQL Server (MSSQL), iyi bir performansa sahiptir. En büyük dezavantajı, sadece Windows üzerinde çalışabilmesidir. Kullanım kolaylığı, güvenilirliği ve işlem gücüyle dikkat çekmektedir. Maliyeti diğer veritabanlarına göre yüksektir. Tablo başına 4 TB veri depolayabilmektedir. “Transaction locking”, “trigger” ve “stored procedure” özelliklerine sahiptir.

### 6. **PostgreSQL**

PostgreSQL, veritabanları için yukarıda bahsedilmiş olan ilişkisel modeli kullanan ve SQL standart sorgu dilini destekleyen bir veritabanı yönetim sistemidir.

PostgreSQL aynı zamanda iyi performans veren, güvenli ve geniş özellikleri olan bir DBMS'tir. Hemen hemen tüm UNIX ya da Unix türevi (Linux, FreeBSD gibi) işletim sistemlerinde çalışır. Ayrıca NT çekirdekli tüm Windows sistemlerde de çalıştırılabilir. Ücretsiz ve açık kodludur.

PostgreSQL diğer ticari ya da açık kodlu veritabanlarında bulabileceğiniz özelliklerin hemen hemen hepsini (ya da daha fazlasını) kapsar.

PostgreSQL özellikleri (PostgreSQL FAQ'da listelendiği gibi):

- Transactions
- Subselects
- Views
- Foreign key referential integrity
- Inheritance
- User-defined types
- Rules



## 7. Oracle

Oracle, dünyanın en güçlü ve güvenilir veritabanı olarak gösterilmektedir, ancak çok yüksek maliyeti nedeniyle sadece büyük kurumların tercih edebilecekleri bir veritabanıdır. Windows ve Unix sistemlerinde kullanılabilir. Oracle, sınırsız sayıda tabloları desteklemektedir. Çok büyük yazılım desteği vardır.

## 8. Interbase

Linux platformu üzerinde çalışan, ücretsiz bir veritabanı sunucusudur. Yaygın değildir. Ücretsizdir.

## 9. Progress

Linux ve Windows platformu üzerinde çalışan çok kuvvetli bir veritabanı sunucusudur.

# Hangi veritabanını seçmeli?

Yukarıda da görüldüğü gibi, her veritabanının farklı özellikleri bulunmaktadır. Eğer, küçük yoğunlukta trafiğe sahip bir web siteniz varsa, bunun için maliyeti diğerlerine göre daha düşük olan Access kullanılabilir. Web uygulamaları için, amacınız veritabanı oluşturmak değil de, veritabanını kullanmak ise, MySQL bunun için biçilmiş kaftandır. Daha büyük ve orta ölçekli uygulamalar içinse, kullanım kolaylığı açısından Progress ya da MS SQL kullanılabilir. MS SQL'in Windows ortamında sınırlı bir kullanım alanı olduğunu da unutmamak gerekir. Bu işletmeler, Linux üzerinde PostgreSQL kullanırlarsa hem sıfır maliyetle (teknik açıdan) hem de daha iyi performansla işlemlerini yönetebilirler. Oracle ise çok yüksek güvenilirlik ve işlem gücü gerektiğinde tercih edilen bir veritabanı sunucusudur. PostgreSQL burada yine düşük maliyetli sistemler gerektiğinde, özellikle yurt dışında tercih edilmektedir.

## Veritabanları konusunda Türkiye’de nasıl destek alabilirim?

Ülkemizde ücretli destek çok fazla gelişmemekle beraber, bu amaçla gönüllü insanların üye olduğu e-posta listeleri kurulmuştur. Bunlardan ikisi aşağıda verilmiştir:

- [Linux-programlama@liste.linux.org.tr](mailto:Linux-programlama@liste.linux.org.tr) (Veritabanları için tartışma listesi (üye olmak için, <http://liste.linux.org.tr> web arayüzünü kullanabilirsiniz.)
- [pgsql-tr-genel@postgresql.org](mailto:pgsql-tr-genel@postgresql.org) (PostgreSQL Türkiye E-posta listesi. PostgreSQL hakkında her türlü konu konuşulabilir.)

### Bazı linkler:

- Tüm veritabanları için: <http://foundries.sourceforge.net/databases>  
(Bu sitede, veritabanları konusunda yardımcı olabilecek kişiler bulunmaktadır.)
- PostgreSQL : <http://www.PostgreSQL.org> (resmi sayfa)  
<http://gborg.PostgreSQL.org> (projeler)  
<http://techdocs.PostgreSQL.org> (belgeler)  
<http://techdocs.oper.metu.edu.tr> (resmi yansıması)
- MySQL : <http://www.mysql.com>
- Oracle : <http://www.oracle.com/>  
<http://otn.oracle.com/>

### Belgeye katkıda bulunanlar

Bu belgenin oluşmasındaki ve güncellenmesindeki katkılarından dolayı, Doruk FİŞEK ([dfisek@fisek.com.tr](mailto:dfisek@fisek.com.tr)) ve Burak DAYIOĞLU’na ([dayioglu@metu.edu.tr](mailto:dayioglu@metu.edu.tr)) teşekkür ederim.

### Belgenin en güncel hali

Belgenin en güncel haline, <http://www.gunduz.org/seminer/vtgiris> adresinden ulaşabilirsiniz.

**Belgenin son güncelleme tarihi : 09.02.2006**