

## A. VERİTABANI NEDİR?

Veritabanı, bir yada birkaç uygulamada kullanmak için, gereksiz yinlemelerden arınmış olarak, düzenli biçimde bilgisayar belleklerinde veya saklama ortamlarında saklanan birbirleriyle ilişkili veriler topluluğudur.

Veritabanının ne olduğunu ifade etmek için sadece yukarıdaki tanımlar yeterli olmayabilir. Çünkü her veri topluluğu veritabanı olmayabilir. Dolayısıyla veritabanının özelliklerinden de bahsetmek gerekir:

1. Bir veritabanı bir çok uygulamada kullanılabilen, birbirleriyle ilişkili, işletilebilen verilerden oluşur. Bu uygulamalar bir bankanın, bir şirketin, bir hastanenin olabilir. Bu veriler bu kuruluşlarda sürekli bulunurlar ve bir çok program tarafından ortak kullanılabilirler.
2. Veritabanındaki veriler gereksiz tekrarlardan arındırılmış olmalıdır. Gereksiz yinlemeler bir süre sonra, özellikle büyük kuruluşlarda ve internet üzerinde yoğun bilgi giriş çıkışının olduğu yerlerde çökmelere, yavaşlamalara sebep olabilir.
3. Veritabanında saklanan veriler değişmeyen statik veriler değildir. Ekleme, silme ve güncelleme işlemleri ile veriler değiştirilebilir. Bunun yanında sorgulama ve raporlama gibi işlemler de yapılabilir.
4. Veritabanındaki veriler üzerinde merkezi bir denetim vardır. Bu denetim, birer yazılım olan Veritabanı Yönetim Sistemi aracılığıyla mümkün olmaktadır. VTYS aracılığıyla veritabanının bilgisayardaki fiziksel yapısı kullanıcılardan gizlenir ve kullanıcının rahatlıkla veritabanına hükmedebileceği kullanışlı arabirimler sunar. Bazı bilinen VTYS araçları şunlardır: MS Access, MS SQLServer, Oracle, MySQL, PostgreSQL, SQLite, InterBase, FireBird, Clipper, SyBase, Paradox, FoxPro, dBase, ...

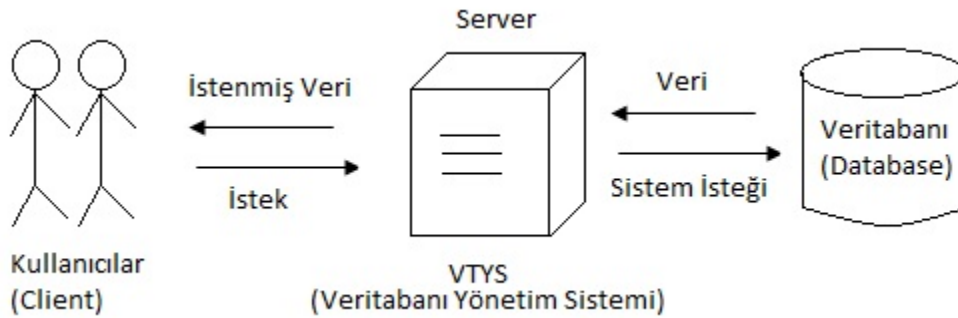
## B. VTYS' lerin Çeşitleri

### I. Database' e Erişim ve Saklama Ortamı Açısından

#### 1. Yerel Veritabanı Sistemleri

Tek bir bilgisayar üzerinde bulunan ve yalnızca bir kullanıcının erişebildiği DB' dir. Masaüstü DB adı da verilebilir. Ucuzdurlar, sadedirler ama tek kullanıcıdırlar, gelişmiş nesneleri, özellikleri yoktur, güvenlik açıkları vardır. En ünlüleri dBase, Lotus, Paradox, FoxPro, Clipper, FileMakerPro.

#### 2. İstemci / Sunucu (Client / Server) Veritabanı Sistemleri



Bu sistemlerde, DB bir sunucu üzerinde bulunur ve veriler bilgisayarda saklanır. Bu bilgisayar sunucu (server) kısmıdır. Bu bilgisayardaki veritabanına başka bilgisayarlar üzerinden erişen

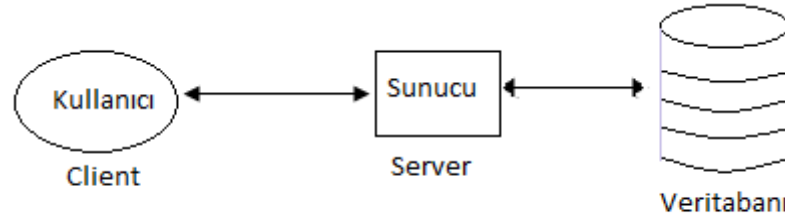
kullanıcılar ise client(istemci)dir. Bir ağ ortamı sözkonusudur. Bu tür VTYS' den bazıları; SQL Server, Oracle, MySQL, InterBase, FireBird, PostgreSQL,...

Bu sistemler esnek, performansı yüksek, gelişmiş nesneleri barındıran, güvenlik özellikleri olan sistemlerdir.

Dezavantajları: özel bilgi ve eğitim gerektirmesi ve pahalı olmaları (Örn,Oracle,SQLServer)

## II. Veriler Üzerinde İşlem Açısından VTYS

1. **Tek Katmanlı VTYS'ler:** Kayıtlar üzerinde herhangi bir değişikliğin anında meydana geldiği veritabanıdır. Hem DB hem de program aynı bilgisayarda olduğu için birbirlerine direk bağlıdır. Yerel veritabanları bu sınıftadırlar. FoxPro, Clipper dibi...
2. **İki Katmanlı VT'leri:** Burada bir istemci uygulama, veritabanı sürücülerini kullanarak veritabanı sunucusuna erişim sağlar. Örneğin Delphi' de yazılan bir uygulama Database Desktop aracılığı ile Paradox veritabanına bağlanabilir. Arada bir sunucu yoktur.
3. **Çok Katmanlı VT'ler:** Bu modelde client(istemci) veritabanına doğrudan değil de aradaki uygulama sunucusu üzerinden bağlanır.



Client Uygulama bir veya birden fazla application server (uygulama sunucusu) ile bağlantı kurar. 3 bölüme ayrılmıştır. Bu yüzden 3-katmanlı da denilebilir.

Client Application: Kullanıcı bilgisayarındaki arabirim

Application Server: Ağ ortamında Clientların erişebileceği merkezi bir konumdadır ve clientler için veri servisi sunar.

Remote DB Server (Uzaktan Erişimli DB Sunucu): İlişkisel veritabanı yönetimi sistemini sunar.

## C. Veritabanı Yaklaşımının Yararları

Geleneksel Yaklaşım: Bu yaklaşımda her uygulama ile ilgili veriler için bir yada birkaç dosya oluşturulur. Uygulamaları gerçekleştirmek için bir programlama dili kullanılır.

Veritabanı Yaklaşımı: Bu yaklaşımda öncelikle bir VTYS seçilir. Veritabanı tasarlanır ve VTYS'nde tanımı yapılır. Daha sonra VTYS' de sorgulama imkanları da kullanılarak uygulama programı ile veriler işlenir.

Yukarıdaki yaklaşım farkından yola çıkılarak veritabanı yaklaşımının sağladığı yararları ve kolaylıkları aşağıdaki gibi sıralayabiliriz:

- a) **Veri tekrarının önlenmesi:** Verilerin normalize edilmiş ilişkisel hali tutulduğundan dolayı gereksiz bellek ve yer israfı engellenmiş ve performans artmış olur.
- b) **Veri bütünlüğünün sağlanması(data integrity):** Veriler bu sayede doğru ve tutarlı hale getirilebilir. VTYS' de bulunan kısıtlamalar(constraints) ile verilerin tutarlı olması sağlanabilir. Örneğin öğrenci notlarının 0..100 aralığında olması gerektiği gibi, "bir öğrencinin aldığı derslerin saati 40'ı geçemez" kısıtlaması gibi.

- c) Veri paylaşımının sağlanması: Her uygulamanın ayrı bir verisi yoktur. Veritabanındaki veriler tüm uygulamalar tarafından ortak kullanılır. Bu sayede uygulama programından bağımsızdırlar.
- d) Veriler daha güvenli bir ortamda saklanmaktadır.
- e) Verilerin yedeklenmesi ve onarılması çok daha kolaydır.
- f) Veritabanı üzerinde sorgulama dilinin(SQL) kullanılabilmesi
- g) Standartların Uygulanabilmesi: VT yaklaşımında veriler üzerinde merkezi bir denetim uygulanmaktadır. Veritabanı sorumlusu(Database Admin) veritabanının işletiminden sorumludur ve veritabanındaki yapılar, veriler ve uygulamalarla ilgili standartlar kurumsal uygulamalarda önem arz etmektedir.

## D. Temel Veritabanı Kavramları ve Bazı Gelişmiş VTYS Özellikleri

### Tablolar

Saklanan verilerin satırlar (row) ve sütunlar (fields) olarak düzenlenmesi ile oluşan en temel VT yapılarıdır.

Tablo Örneği

Markakod	Markaadı	Renk	Model
100	Renult	Kırmızı	2000

Satır →

Sütun(alan)

Bir veri tabanında çok sayıda tablo bulunabilir. Örneğin bir hastanenin veritabanında hastalar, personel, servisler, birimler için birçok tablo bulunur. Veritabanı tasarımı yapılırken büyük ölçüde tabloların içereceği bilgiler göz önünde bulundurulur ve birbirleriyle ilişkili bir çok tablo söz konusudur.

Ogrencino	ad	soyad	.....	bolumkod

bolumkod	bolumad	fakkod

fakkod	fakadı

Yukarıdaki 3 tablo, birbirleriyle anahtarlar yardımıyla ilişkilendirilmiş tablolardır. Ve öğrenci veritabanının sadece bir kısmı oluşturur.

### Veri Türleri

Bir tabloda saklanan verilerin türünü belirlemekte kullanılır. Örneğin bir maaş bilgisi sayısal bir bilgidir ve o tablo tanımlanırken o sütun sayısal olarak tanımlanmalıdır. Bir ad soyad sütunu karakter türünden olmalıdır. Bu şekilde, her tablonun sütunları, hangi tür verileri tutacaksa ona göre veri tipi tanımlanmalıdır. Veri tipleri VTYS'ne göre çeşitlilik gösterir. Örneğin SQL Server'ın veri tipleri şöyledir:

char (uzunluk) : Sayısal olmayan veriler içindir. Uzunluk 256 ya kadardır. char(20) : en fazla 20 karakterlik veri saklanabilir demektir.

nchar (uzunluk) : unicode türünden chardır 256 ile sınırlı değil 65536 ya kadardır.

varchar (uzunluk) : Uzunluk sınırı veri girince belli olan türdür. Esnektir. Gereksiz yer kaplamaz.

nvarchar (uzunluk) : unicond varchardardır.

text : Çok büyük açıklamaları saklamak için kullanılır. 2GB' a kadar veri tutabilir.

image: Resim tutan alanlardır.

int: Tamsayı tutar -2147483648.....2147483647 arası tamsayıları tutar.  
bigint : -9223372036854775808..... 9223372036854775808 arası tamsayı değerlerini tutar.  
smallint : -32768.....32768 arası tamsayıları tutar.  
tinyint : 0 ile 255 arası tamsayıları tutar.  
decimal, numeric : Ondalıkli sayıları tutar.  
real : floattan daha büyük  
money, smallmoney: Parasal değerleri tutar.  
datetime, smalldatetime: Tarih saat türü veri tutar.  
binary : İkilik sayıları tutar.  
bit: 0 ve 1 gibi sadece iki seçenekli verileri tutar.

SQL Serverda bir tablo tanımlanırken bu veri tipleri bize seçenek olarak sunulur:

Table- dbo.Tablet1\*

Column Name	DataType	AllovNulls
ogmo	int	<input type="checkbox"/>
ad	char(20)	<input type="checkbox"/>
soyad	<div> <div>char(10)</div> <div>datetime</div> <div>decimal</div> <div>float</div> <div>int</div> <div>money</div> </div>	<input type="checkbox"/>

Tablo tanımı yapılırken  
bir görünüm

### **Kısıtlamalar (Constraints)**

Tablo alanlarına veri girişini kontrol etmeyi amaçlayan düzenlemelerdir. Bunlar, kullanıcıların yapabilecekleri muhtemel hatalar açısından önemlidir. Tanımlanan constraintler verilerin doğru girilmesini garantiler, hataları önler. Örneğin öğrencilere ait not bilgisini tutan alana girilecek değer için 0 ile100 arası girilmesi yönünde bir kısıtlama yapılırsa, kullanıcı bu aralığın dışında not giremez. Örneğin MS Access’de

Alan Adı	Veri türü
VİZE	SAYI

Şeklinde bir kısıtlama yapılmıştır. Artık VİZE alanı bir aralıkta olacaktır

Geçerlilik Kuralı	$\geq 0$ and $\leq 100$
-------------------	-------------------------

### Birincil Anahtar ( Primary Key)

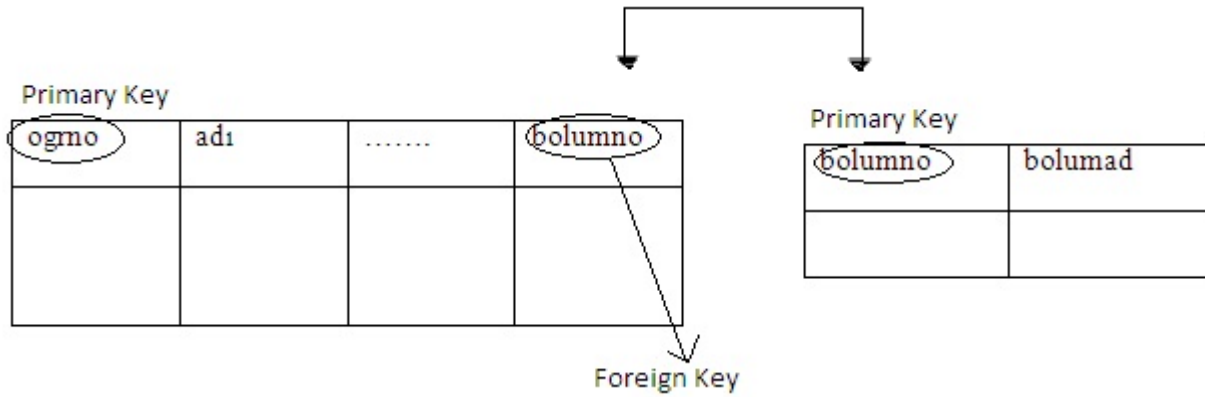
Primary Key özelliğine sahip alandaki verilerin birbirleriyle farklı olması gerektiği verilmiş olur. Örneğin, öğrenci numaralarının tutulacağı alan Primary Key olmalıdır. Veya TCKimlikNo alanı gibi. Primary Key alanı boş olamaz (Not NULL). Bir tabloda en çok bir tane Primary Key olabilir ve bu özelliğe sahip olan, tablolar arasında ilişki kurulmasını sağlayan en güçlü yapıdır.

### Unique Kısıtlaması(Benzersiz, Tek)

Tanımlandığı alandaki verilerin benzersiz ve tek olmasını sağlar. Primary Key kısıtlamasına benzer. Ancak bu kısıtlamada NULL olmasına izin verilir ve bir anahtar özelliği taşımaz.

### Dış Anahtar (Foreign Key)

Tablolar arası ilişkileri tanımlamaya yarayan bir kısıtlamadır. Genelde Primary Key ile Foreign Key birbirine bağlanır. Örneğin



### Default (Varsayılan) Kısıtlaması

Tanımlandığı alana bir değerin girilmemesi durumunda alana sabit, varsayılan bir değerin atanması sağlayan kısıtlamadır. Tanımlanan alana değer girilmediğinde NULL olmasını (değersiz kalmasını) engellemek için kullanılır. Değer girilmediğinde otomatik olarak tanımlanan default bir değer girilir.

### Not NULL Kısıtlaması

Tanımlandığı alana veri girişinin zorunlu olduğunu belirtir. Alanın değersiz (NULL) olamayacağını belirtir. Örnek bir SQL cümlesi:

## CREATE TABLE

( Personelno integer NOT NULL, ad char(20), soyad char(20), ...)

Bu tanıma göre personel tablosundaki personelno alanı NULL olamaz.

### **Tablo ve Sütun İsimlendirme Kuralları**

1. Tablo ve sütun isimleri tek kelimeden oluşmalıdır. Boşluk bırakılmamalıdır.

personel no (yanlış)

personel-no (doğru)

personelno (doğru)

personel kayıtları (yanlış)

2. Türkçe karakter kullanılmamalıdır. (ç, ş, İ, ğ, Ş, Ç, ü, Ğ, ı)

İsim (yanlış)

maaş (yanlış)

adı (yanlış)

adi (doğru)

ilaç (yanlış)

3. Tablo ve sütun isimleri rakam ve sayısal değerlerle başlamamalıdır.

5.tablo (yanlış)

tablo5 (doğru)

### **Görünümler (Views)**

Sanal Tablolardır. Bir tablonun tüm verilerini yada bir kısmını kullanıcılara sunar. Güvenlik açısından yarar sağlar. Zira, tablonun kendisinden sanal bir görünümü daha güvenlidir.

### **Stored Procedure' ler**

Stored Procedure' ler gelişmiş VTYS'de uygulama programının yükünü hafifletip, sistemdeki bilgi alışverişini sunucu tarafına biraz daha görev vererek düzenleyen bir yapıdır. Stored Procedure kullanılmayan bir yapıda tüm sorgular, tüm istekler uygulama programı tarafından yapılır. Ancak Stored Procedure'lerin tanımının yapıldığı sunucuları kullanan uygulama programları ise sadece o Procedure'leri çalıştır komutu verir ve sunucuda çalıştırarak sonuçları verir. S.P. olmadığı durumlarda sorguları çalıştırma işlemi de uygulama programı tarafından yapılır. Buda fazladan yük demektir. S.P. ler içerisinde sorgu gruplarının olduğu bir nevi alt program gibi düşünülebilir. Örneğin öğrencileri memleketlerine göre gruplayan bir sorgumuz var ve bunu Stored Procedure içerisine yazarsınız, ardından programda o S.P.' nin adını vererek çağırırsınız; sunucuda bunu çalıştırarak sonuçları yollar. S.P.' nin kullanılmadığı yöntemde ise o sorgu grubu program tarafında çalıştırılarak sunucudan istenir.

### **Tetikleyiciler(Triggers)**

Bir olaya baęlı olarak bir takım işler yapan SQL kodlarıdır. Örneęin öğrenciler tablosundan bir öğrenci silindiğinde, öğrenciye ait diğer tablolardaki bilgiler trigger ile silinebilir.

Triggerlar otomatiktir. Tablodaki bir olayın öncesinde ve sonrasında çalıştırılırlar. Çok karmaşık işlemler, birden fazla SQL işlemi trigger haline getirilip veritabanının daha kolay yönetilmesini sağlayabilir.

### **Sorgular (Queries)**

Belirtilen kriterlere göre veritabanından bilgi çekmek ya da veritabanının yapısıyla ilgili işlemler (tanımlama, silme, ekleme,...) yapan komutlardır. Bu komutlar SQL (Structural Query Language) adını olarak standart olmuştur. Bazı VTYS' lerde de T-SQL(Transact SQL) şeklinde daha ileri seviyede SQL kullanımı da söz konusudur. Örnek bir SQL cümlesi:

```
SELECT * FROM personel WHERE maas > 5000
```

Yukarıdaki sorgunun anlamı: personel tablosundan maaş 5000 den yukarı olanları listele