

# 1. Veri Tabanı Sistemine Giriş

Dr. Hidayet Takçı

VTYS' den önce verilerin tutulması için dosyalama sistemleri kullanılmaktaydı. Dosyalama sistemlerinde başta veriler arasındaki ilişkilerin kurulması olmak üzere bir dizi problemler bulunmaktaydı. Dosyalar arasındaki ilişkilerin kurulamadığı gibi dosya verilerinin paylaşımı da imkansız denecek derecede zordu. Bu arada bir başka önemli problem de dosyalarda tutulan verilerin yapısı yazılıma bağlıydı. Yani dosyadaki kayıtların yapısını değiştirmek için programı değiştirmek gereği bulunmaktaydı.

Veritabanı yönetim sistemleri kullanılmadan önceki döneme ait problemlerden bazıları şunlardır:

- Her dosya farklı bir sırada işlenmektedir.
  - Bir dosyanın işi bitmeden aynı anda başka bir dosyayla çalışmak mümkün değildir.
- Dosyanın kayıt yapısı değiştiğinde ilgili dosyaya erişen programların da değişmesi gerekmektedir.
  - Dosya yapıları yazılım ile tanımlanmakta ve yine yazılım yoluyla güncellenmekteydi. Dolayısıyla eğer bir dosyanın yapısı değiştirilecek ise onun tek yolu ilgili yazılım kodunu değiştirmek idi.
- Veriler birden fazla kez girilebildiğinden veri tekrarı ve buna bağlı olarak veri tutarsızlığı meydana gelebilmektedir.
  - Bir verinin birden fazla kopyasının tutulması genellikle veriler arasında uyumsuzluk ve güncel veri problemlerine yol açar. Örneğin, bir kişinin hayatı boyunca farklı dosyalarda kimlik bilgileri tutulmakta ve kimlik bilgileri genellikle değişim göstermektedir. Bütün dosyaların aynı anda güncellenmesi ise imkansız denecek kadar zordur. O yüzden dosyalama sisteminde veri tutarsızlığı önemli bir problemdir.
- Dosyalarda tutulan verilere izinsiz erişimleri denetleyen bir mekanizma bulunmamakta ve bu da güvenlik problemini ortaya çıkarmaktadır.
  - Henüz dosya bazında güvenlik sağlamak mümkün olmadığı için dosyalama sistemlerinin güvenlik problemi bulunmaktadır.
- Verilerin işlenmesinin maliyeti çok yüksek olabilir.
  - Dosyalarda bulunan verilerin sorgulanması ve işlenmesi için yeni prosedürler yazmak ihtiyacı vardır. Veri yapısı değiştikçe veri işleme yazılımlarını da değiştirmek gerektiğinden veri işleme maliyeti çok yüksektir.
- Yazılımların etkinliği düşük olur.

Dosyalama sistemlerindeki problemler için en uygun çözüm Veritabanı kullanımıdır. Veritabanı (VT) birbiriyle ilişkili veriler topluluğudur. Veritabanında yer alan bilgiler kimi zaman bir öğrencinin kimlik bilgileri kimi zamanda bir kitabın bibliyografik bilgileridir. Nesnelerin almış olduğu özellik değerleri veritabanlarında tutulurlar. Veritabanları da dosyalar gibi bilgi tutmakla birlikte kapasiteleri dosyaların kapasitesinden oldukça fazladır. Tuttuğu veriler arasındaki ilişkileri kurma ve bu ilişkileri modelleme konusunda oldukça başarılıdırlar. Örneğin, Ahmet isimli bir kullanıcı ÖĞRENCİ nesnesinin bir örneği olup KİTAP nesnesinin bir örneği olan

“Data Mining“ isimli bir kitabı ödünç alabilir ve ÖĞRENCİ ile KİTAP nesneleri arasındaki bu ilişki Ahmet, Data Mining kitabını ödünç almıştır şeklinde bir ilişki ile sunulabilir. Burada ÖĞRENCİ ve KİTAP nesnelere, ödünç alma ise ilişkinin adıdır.

Veritabanını bir veri havuzu gibi düşünecek olursak bu veri havuzundan herkes izni doğrultusunda faydalanabilir. Veritabanının kullanıcıları sadece yetkili oldukları kısımlarda, yetkileri nispetinde işlem yapabilirler. Örneğin, kullanıcıların bir kısmı veritabanı nesnelere bir kısmında bir kısım yetkilere sahip olabilir. Veritabanı her bir kullanıcı için farklı bir görünüme sahiptir aslında ve kullanıcıların her birinin veritabanına bakışına bir veritabanı görünümü adı verilir.

Veritabanındaki yetkilendirmeler, yasaklamalar ve yetkilerin kullanımı gibi işlevleri kontrol eden bir yapı bulunmaktadır. Bu yapıya veri tabanı yönetim sistemi (VTYS) adı verilir. VTYS, veritabanı ile ilişkili işlevleri yöneten yazılımlar topluluğudur.

**Veri Tabanı Yönetim Sistemi (VTYS)** genel amaçlı bir yazılım sistemi olup kullanıcıya değişik uygulamalar için veri tabanlarının tanımlanmasına, kurulmasına ve işlenmesine olanak verir.

Veritabanında bulunan veriler ile veritabanının yönetim işlevlerini sağlayan VTYS ikisi birlikte **Veri Tabanı Sistemini** meydana getirir.

## 2. VTYS Neden Kullanılır?

- Öncelikle veritabanları bir verinin sadece bir kez girilmesini sağlar. Çünkü veritabanında bir veri sadece bir kez girildiği takdirde verilerde tutarsızlık meydana gelmez ve bellekten tasarruf sağlanır.
- Veritabanının bir başka kullanım nedeni verilerin paylaşımıdır. VTYS veritabanındaki verilere aynı anda farklı kullanıcıların erişmesine imkan sağlar.
- Verilerin güvenliği de veritabanı kullanımı için sebeplerden biridir. Kullanıcı hakları ve kimlik denetimleri güvenliği sağlayıcı mekanizmalardır.
- Sorgu dili imkanları ile veritabanlarında bulunan verilere hemen her ortamdan erişmek mümkündür. Özellikle ODBC benzeri ara katman yazılımlarda veritabanına çeşitli yollardan erişilebilir.
- Dosyalarla sağlanamayacak derecede karmaşık ilişkiler VTYS ile sağlanabilmektedir.
- Veritabanına doğruluk kriterleri konularak hatalı veri girişi engellenebilmektedir. Bu kriterler iki şekilde kullanılabilir. Birinci tip kriter ile; sayısal veri girilecek yere sayısal veriden başkası kabul edilmeyebilir. İkinci tip kriter için bir örnek ise olmayan üyeye ödünç kitap verilmemesi olabilir.
- VTYS aynı zamanda yedekleme konusunda da çözümler sunmaktadır. Bazı VTYS lerde veriler iki yerde kayıt edilirler.
- Veriler ile uygulamalar birbirinden bağımsızdır.
- Veritabanı merkezi olarak yönetilebilir.

### 3. Veri Tabanları Konusunu Niçin Çalışmalıyız?

- Bilgisayar teknolojisi hesaplama dan bilgiye doğru bir yöneliş halindedir.
- Veri kümelerinin hacmi artmakta ve çeşitlenmektedir. Bilimsel verilerden askeri verilere kadar birçok alanda veritabanı kullanımı artmıştır. Ayrıca, İnternet veritabanlarını bütün kullanıcılara açık hale getirdiği için kullanımı en yüksek seviyeye çıkmıştır.
- Kısacası içinde VT olmayan hemen hiçbir ciddi uygulama kalmamıştır.

### 4. Veri Modelleri ve Sınıflandırılması

*Model* kelimesi ise; isim, sıfat ve fiil olarak ve her birinde oldukça farklı çağrışımlar yapacak şekilde kullanılmaktadır. İsim olarak “model”, bir temsili ifade eder. Bu temsil; bir mimarın, bir binanın küçük ölçekli modeli veya bir fizikçinin bir atomun büyük ölçekli modelini oluşturması anlamındadır. Sıfat olarak “model”, mükemmeliyetin veya idealin ölçüsünü ifade eder. “Model ev”, “model öğrenci” ve “model eş” ifadelerinde olduğu gibi. Fiil olarak “model” ise, bir şeyin nasıl olduğunu ispat etmek, açıklamak, göstermek anlamındadır.

Bilimsel modeller bütün bu çağrışımları bünyelerinde bulundururlar. Onlar; durumların, nesnelere ve olayların temsilleridir.<sup>50, 51</sup> Gerçeklerden daha az karmaşık ve böylece araştırma amacıyla kullanılmaları daha kolay olduğundan, bu anlamda ideal hale getirilmişlerdir. Gerçek durumlarla karşılaştırıldıklarında, modellerin basitliğinin sebebi, gerçeklerin sadece uygun özelliklerini temsil etmelerinden kaynaklanmaktadır. Örneğin, yeryüzünün bir kısmının modeli olan bir yol haritasında, bitki örtüsü gösterilmez. Çünkü bu durum, o haritanın bir yol haritası olarak kullanımı açısından uygun değildir. Güneş sisteminin bir modelinde, gezegenleri temsil eden topların, gezegenlerle aynı maddeden yapılmış olmaları veya aynı sıcaklığa sahip olmalarına ihtiyaç yoktur.

Bilimsel modellerden, gerçeklerin farklı boyutları hakkındaki bilgiyi artırmak ve birbirleri ile ilişkilendirmek için faydalanılır. Modeller, gerçeği ortaya çıkarmak ve bundan daha fazla olarak, geçmiş ve şimdiki durumu açıklamak ve geleceği tahmin ve kontrol etmek için kullanılır. Modeller uygulanarak, gerçekler üzerinde bilimin kontrolü sağlanır. Modeller gerçeğin tanımı ve açıklamasıdır. Bir bilimsel model, aslında, gerçek hakkında bir veya bir dizi ifadelerdir. Bu ifadeler olaylara dayanan, kanun benzeri ya da teorik olabilir.

Bilimde, sıradan işlemlerde olduğu gibi, değişik tipte modeller kullanılır: Simgesel Model, Benzetim Modeli ve Sembolik Model.

*Simgesel Modeller*, durumların büyük veya küçük ölçekli temsilleridir. Gerçek şeylerin uygun özelliklerini temsil ederler. Şekilleri, temsil ettikleri şeylere benzer. Yol haritaları, hava fotoğrafları bu tip modellere örnek verilebilir. Bazı durumlarda ise; haritada yükseltiler, yol genişlikleri gibi özellikleri belirtmek gerekebilir. O zaman, renkler ve kontur çizgileri gibi bir takım açıklayıcı özelliklere ihtiyaç duyulur. Bu tip modeller *Benzetim Modelleri* olarak isimlendirilir.

*Sembolik Modellerde*, temsil edilen şeylerin özellikleri sembollerle ifade edilir. Böylece, bir grafik ile gösterilen ilişki (benzetim modeli), bir eşitlik olarak da ifade edilebilir. Bu tip modellere *Matematiksel Modeller* de denilmektedir.

Bu üç tip modelden benzetim modeli, soyut ve geneldir. Matematiksel model ise en soyut ve en genel modeldir. Üzerinde düzenleme yapılabilmesi daha kolaydır. Simgesel modellerin ise anlaşılması diğerlerine göre daha kolaydır. Bilişim sistemlerinin oluşturulması için kullanılan veri modelleri, benzetim modelleri ve sembolik modellerdir.<sup>52</sup>

Bir bilişim sisteminin kullanıcısı, özellikle bir karar verici, kendisini sonsuz denebilecek boyutta bilgi karşısında bulur. Bir bilişim sistemi modeli, gerçek bilgi kümesinin alt kümesini oluşturur ve onun daha basit bir şeklidir. Bu şekil, işlenebilmeye imkân verir ve bunu kullanarak elde edilen çözüm veya cevap, gerçek hayatta uygulanmaya çalışılır. Model, var olan bilgi yığınının bir düzen getirmeyi, hatta bir yapı oluşturmayı amaçlar. Tek bir model yoktur. Var olan bilgi yığınının, uygulanan farklı modeller doğal olarak farklı yorumlar getirir.

## Veri Modeli

Veri modeli; veritabanının yapısını tanımlayan kavramların bir kümesidir. Veritabanının modeli; veri tipleri, işlemler ve kısıtlamalardan oluşur. Veri modeli veritabanının nasıl oluşturulacağı, nasıl kullanılacağı ve hangi sınırlar içerisinde tanımlı olduğu ile ilgilidir. Veri modeli içerisinde yer alan işlemler kümesi veritabanından seçim yapma ve veritabanını güncelleme ile ilgilidir.

## Veri Modellerinin Sınıflandırılması

1. **Yüksek Seviyeli Veri Modelleri:** Bu modeller varlıklar (entities), özellikler (attributes) ve ilişkiler (relationships) gibi kavramlardan oluşur. Bu modeller için en bilinen örnek Varlık-İlişki veri modelidir.
2. **Mantıksal Veri Modelleri:** Bu modeldeki kavramlar verilerin bilgisayarda fiziksel olarak organize olma biçimine çok benzerler. Bunlara bir örnek ilişkiyel veri modelidir. İlişkiyel veri modelinde iki boyutlu tablolar kullanılır. Tabloların her bir satırında bir kişiye ait değerler, her bir sütununda ise bir özelliğe ait değerler bulunur.
3. **Fiziksel Veri Modelleri:** Bu modeller verilerin bilgisayarda nasıl tutulduklarını detaylı olarak gösteren konseptlere sahiptirler. Bunlar kayıtların biçimiyle, kayıtların sırasıyla, ulaşma yollarıyla ilgili bilgiler içerirler.

## 5. Veri Tabanı Tanımı (Şema)

Veritabanı şeması aslında veritabanında bulunan dosyaların kayıt yapısıdır. Bir veritabanında ne kadar tablo varsa veritabanı şemasında o kadar tablonun kayıt yapısı tanımlanır.

### Örnek (Kütüphane Veritabanı):

YAZARTIPLERİ(YAZARTIPI, YAZARTIPADI)  
YAZAR(YAZARNO, YAZARADI, YAZARTIPI, DNO)  
NOTLAR(NOTNO, NOTLAR, DNO)  
KONUBASLIK(KONUBNO, KONUB, DNO)  
YAPITADI(YAPITNO, YAPITADI, DNO)  
FIRMA(FIRMANO, FIRMAADI)  
DILLER(DILNO, DILADI)

KITAP(DNO, KADI, ORNEKAD, BASIM, YY, YE, YT, ISBN, OZET, BOLUMNO, UYENO, FOZELIK, DIZI, TASNIFNO, FIYAT, FIRMANO, GELISTAR, DILNO, ADET)  
BOLUM(BOLUMNO, BOLUMADI)  
GRUP(GRUPNO, GRUPADI)  
UNVAN(UNVANNO, UNVANADI)  
UYE(UYENO, ADSOYAD, BOLUM, GRUP, UNVAN, ADRES, MAIL, TEL, GSM)  
ODUNC(UYENO, DNO, ALISTAR, IADETAR)  
IADE(UYENO, DNO, GELISTAR)

**Veritabanı örneği:** Veritabanında bir andaki veriyi içerir.

**Örnek:**

ID	HNO	BNO	DNO
001	1047	104	BİL 102
001	10415	104	BİL 314

## 6. Veri Tabanı Dilleri

### Veri Tanımı Dilleri(VTD-DDL)

***Bu diller VT'nin şema tanımlarını yerine getirmek için kullanılırlar. Bu diller sayesinde tabloların oluşturulması, değiştirilmesi gibi işlemler yerine getirilir.***

#### YAZARTIPLERI Tablosu

```
CREATE TABLE YAZARTIPLERI  
(  
YAZARTIPI INTEGER NOT NULL,  
YAZARTIPADI CHAR (20),  
PRIMARY KEY(YAZARTIPI)  
)
```

#### YAZAR Tablosu

```
CREATE TABLE YAZAR  
(  
YAZARNO INTEGER NOT NULL,  
YAZARADI CHAR (50),  
YAZARTIPI INTEGER NOT NULL,  
DNO INTEGER NOT NULL,  
PRIMARY KEY(YAZARNO),  
FOREIGN KEY (YAZARTIPI) REFERENCES YAZARTIPLERI  
FOREIGN KEY (DNO) REFERENCES KITAP,  
)
```

### Veri İşleme Dilleri(VİD-DML)

***VT'nin işlenmesinde kullanılırlar.***

**Tek başına kullanılan bu tür diller VT'nin komplike operasyonlarını bütün olarak belirtebilmektedir. Bunlar ya direk komut olarak girilebilmekte ya da genel amaçlı bir programlama dili içinde tanımlanabilmektedir.**

```
SELECT uyeno, adsoyad, bolum FROM uye ORDER BY bolum ASC
```

```
SELECT * FROM kitap WHERE fiyat >5.000.000
```

**Veri tabanı yöneticisinin (VTY) temel sorumlulukları:**

1. Mantıksal ve fiziksel şemaların tasarımı
2. Emniyetin sağlanması ve izin verilme işlemlerinin yapılması
3. Verilere istenildiğinde ulaşılabilmesi ve VT'nin gereken durumlarda kurtarılması.
4. VT' nin ihtiyaçları değiştikçe ayarlamaların yapılması.

Yani, VTY VTYS' nin nasıl çalıştığını anlamak zorundadır.

## 7. VTYS'lerin Sınıflandırılması

### Veri Modeline Göre

- İlişkisel VTYS'ler – Sybase, Oracle 7.6, SQL Server
- Seviyeli VTYS'ler – IMS
- Ağ VTYS'ler – DBTG
- İlişkisel nesne modelli VYYS'ler – Oracle 8, DB2, Informix
- Nesneye-yönelik VTYS'ler – GemStone, O2, ObjectStore
- Genişletilmiş İlişkisel VTYS'ler – Starburst
- Çıkarımcı VTYS'ler – Postgres, Educe, Prosql
- Çıkarımcı Nesneye Yönelik VTYS'ler

### Kullanıcı Sayısına Göre

- Tek kullanıcı sistemler
- Çok kullanıcı sistemler

### Yer Sayısına Göre

- Merkezi VTYS'ler
- Dağıtık VTYS'ler
- Homojen VTYS'ler – bunlar çok yerde aynı VTYS yazılımını kullanırlar.
- Heterojen VTYS'ler – bunlar federal VTYS'lere dönüşürler ve diğer bir isimle de çokluveritabanı sistemi olarak adlandırılırlar.

## Haftanın Özeti

Bu kısımda;

- VTYS'nin genel yapısını,
- VTYS'nin önemini,
- VTYS'nin kullanım nedenlerini gördük.

Genel olarak VTYS'ler için aşağıdakileri söyleyebiliriz:

- VTYS'ler büyük veri kümelerinin sorgulanması ve tutulması için kullanılırlar.