

# VERİ TABANI YÖNETİM SİSTEMLERİ - I



YRD. DOÇ. DR. ZEHRA ALAKOÇ BURMA

## İÇİNDEKİLER

VERİ TABANI YÖNETİM SİSTEMLERİ .....	2
1. TEMEL VERİ TABANI KAVRAMLARI .....	2
1.1. Veri Nedir? .....	2
1.2. Veri Tabanı Nedir ? .....	3
1.3. Veri Modeline Göre Veritabanı Yönetim Sistemleri .....	4
1.4. Neden Veritabanı Kullanılır? .....	6
1.5. Veri Tabanı Yönetim Sistemlerinin Sağladığı Yararlar .....	7
1.6. Bilinen VTYS Programları .....	7
1.7. Proje ve VTYS arasındaki ilişki.....	8
2. VERİ ve VERİ MODELLERİ .....	10
2.1. Model Nedir?.....	10
2.2. Veri Kavramı .....	11
2.2.1. Veri Güvenliği.....	12
2.2.2. Veri Tekrarı ve Veri Bütünlüğü .....	12
2.3. Veri Modeli .....	13
2.3.1. Yapılar.....	13
2.3.2. Kısıtlar .....	14
2.3.3. İşlemler .....	14
2.4. Başlıca Veri Modelleri.....	15
2.4.1. Basit Veri Modelleri .....	15
2.4.2. Geliştirilmiş Veri Modelleri .....	16
3. VERİ TABANI TEMEL KAVRAMLARI .....	22
3.1. Tablo ve Elemanları .....	22
3.2. Veri Tipi (Data Type) .....	23
3.2.1. Access Veri Tabanı Veri Tipleri .....	23
3.2.2. MySQL Veri Tabanı Veri Tipleri.....	24
3.2.3. Oracle Veri Tabanı Veri Tipleri .....	24
3.3. Zorlayıcı (Constraint).....	25
3.4. Anahtar (Key) .....	26
3.5. Index (İndeks) .....	27
3.6. View (Görüntü) .....	27
3.7. Joining (ilişkilendirme).....	28
4. VERİ TABANI TASARIMI ve NORMALİZASYONU .....	29
4.1. Veri Tabanı Tasarımı.....	29
4.2. Veri Tabanı Normalizasyonu .....	33
4.3. İlişkisel Veri Tabanı Yönetim Sistemleri .....	35
5. ÖRNEK BİR VERİ TABANI TASARIMI ve NORMALİZASYONU .....	37
5.1. Örnek Personel Projesi Veri Tabanı Tasarımı .....	37
6. SQL VERİ İŞLEME DİLİ .....	41
6.1. SQL Nedir? .....	41
6.2. Veri Tabanı Programlarında ve SQL de Değişken Tanımlama .....	42
6.3. SQL Programı Çalıştırılması ve Yazım Kuralları .....	43

7.	TEMEL SQL KOMUTLARI-I .....	48
7.1.	CREATE (Yarat) Komutu .....	48
7.2.	ALTER (Düzenle) TABLE Komutu.....	51
7.3.	DROP (Sil) TABLE Komutu.....	52
7.4.	DESCRIBE Komutu.....	52
7.5.	INSERT (Ekle) Komutu .....	53
8.	TEMEL SQL KOMUTLARI-II .....	60
8.1.	SELECT (Seç) Komutu .....	60
8.1.1.	SQL Operatörleri .....	62
8.1.2.	Order By.....	65
8.1.3.	Distinct .....	67
8.2.	UPDATE (Güncelle) Komutu .....	68
8.3.	DELETE (Sil) Komutu.....	70
9.	SQL 'de FONKSİYONLAR .....	71
9.1.	SUM (Topla) Fonksiyonu.....	71
9.2.	AVG (Ortalama) Fonksiyonu .....	71
9.3.	MAX (En Büyük) Fonksiyonu .....	72
9.4.	MIN (En Küçük) Fonksiyonu.....	72
9.5.	COUNT (Say) Fonksiyonu.....	73
10.	SQL 'de GRUPLANDIRMA .....	75
10.1.	GROUP BY (Gruplandır) Deyimi.....	75
10.2.	HAVING (Sahip) Deyimi .....	77
11.	SQL 'de JOIN (BİRLEŞTİRME) İŞLEMİ .....	79
11.1.	JOIN (Birleştirme) İşlemi .....	79
11.2.	JOIN İşlemine Ait Örnekler .....	83
12.	SQL 'de YÖNETİMSEL FONKSİYONLAR.....	87
12.1.	VIEWS ( Tablo Görünümü).....	87
12.2.	CREATE TABLESPACE (Tablo Uzayı = Veri Alanı).....	89
12.3.	CREATE USER (Kullanıcı) .....	89
12.4.	CREATE ROLE .....	90
12.5.	CREATE INDEX .....	90
12.6.	CREATE SEQUENCE .....	91
12.7.	GRANT .....	91
12.8.	REVOKE.....	91
	KAYNAKLAR.....	92

## VERİ TABANI YÖNETİM SİSTEMLERİ

Veritabanı kavramı ilk olarak 1980’li yıllarda ortaya atılmış olmasına rağmen; günümüzde hemen hemen tüm veri kullanılan alanlarda Veritabanı Yönetim Sistemleri (VTYS) olmadan hiçbir şey yapılamaz hale gelmiştir. Basit bir Web uygulamasından, devasa kuruluşların ağır verilerine kadar, günümüzde bir çok alanda veritabanı uygulamalarına ihtiyaç duyulmaktadır. İşletim sistemlerinden sonra en popüler ve en çok gelir getiren yazılımlar Veritabanı Yönetim Sistemi Yazılımlarıdır.

Günümüzde, bir çok alandaki veri işlemlerinde pek çok Veri Tabanı Yönetim Sistemleri programları yaygın olarak kullanılmaktadır. Birbirinden farklı isimler adı altında anılan bu programlar için bir çok nesne birbiri ile aynı temel işlevi yerine getirmekte olup, yaklaşık olarak aynı teorilere dayanarak çalışırlar.

Veri tabanı, bir kuruluşun uygulama programlarının kullandığı operasyonel verilerin bütünüdür. Veritabanı Yönetim Sistemleri, verilerin fiziksel hafızadaki durumlarını, kullanıcıların erişimlerini düzenleyen sistemlerdir. İlişkisel VTYS’ler günümüzde yaygın olarak kullanılmaktadır.

### 1. TEMEL VERİ TABANI KAVRAMLARI

#### 1.1. Veri Nedir?

Bilgi (information) kavramı yeni biçimlenmeye başlayan bir kavramdır; üzerindeki düşünce ekolleri henüz yeterince gelişmediğinden, İngilizce’de bile, bir çok anlama çekilmektedir. Bu kavram, daha önce de değinildiği gibi, çeşitli açılardan başlıca şu şekillerde açıklanmaktadır:

- Bilgi, bir nesne veya olayda veya bunlara ilişkin raporlarda ortaya çıkan mesaj ile ilgilidir. Bu açıdan ele alındığında, sadece kaynağın bir fonksiyonu olma özelliği taşır ve bazen veri olarak da ifade edilir.
- Diğer bir açıdan bilgi, mesajın iletilmesini açıklayan bir kavramdır. Bu açıdan ele alındığında bilgi, mesajın iletilmesi ile ilgili ihtimal hesaplarına dayanan, belirsizliğin azaltılması için gerekli olan bir kavramdır. Bir başka ifadeyle, bu anlamda bilgi iletişim kanalının da bir fonksiyonudur.
- Diğer bir açıdan ise, bilgi bir alıcı tarafından kazanılan anlam ile ilgilidir. Bu anlamdaki bilgi, hem iletişim kaynağının hem de alıcının bir fonksiyonudur.

Bu görüşlerin birincisi, literatürde fazla benimsenmeyen bir görüştür. Genellikle veri ile bilgi arasında farklılık olduğu ve verinin bilgi elde etmeye yarayan işlenmemiş ham malzeme olduğu kabul edilir. Kişi; bilmek, öğrenmek istedikten ve veriyi kullanmaya başladıktan sonra bilgi ortaya çıkar.

İkinci görüş en yaygın olanıdır. Buna göre bilgi anlamlı biçimde derlenen ve birleştirilen veridir ve şimdiki zamanda ve gelecekte verilecek kararlar için varolan gerçek bir değerdir. Bir başka ifade ile, bir kaynaktan, bir alıcıya iletilen mesajın içeriğidir. Bu anlamda bilgi, karar verme ile bağlantılıdır ve dolayısıyla veriye göre daha etkin bir kavramdır.

Veri kelimesinin tekil hali (datum) Latince'den gelmektedir. Sözlük anlamı “gerçek” tir. Fakat, veri her zaman somut gerçekleri göstermez. Bazen, kesin değildirler veya hiç olmamış şeyleri, örneğin bir fikri tarif etmek için kullanılırlar. Burada bahsedildiği şekliyle veri, bir kişinin formülleştirmeye veya kayıt etmeye değer bulduğu her türlü olay ve fikir anlamındadır. Bilgisayarda veri depolanacağı zaman, çoğunlukla veri tabanı yönetim sistemleri kullanılarak gerçekleştirilir. Çünkü bu tip sistemlerde yanlış verinin depolanmasını ve/veya verinin istenmeyen kişilerin kullanımına sunulmasını engelleyen bir takım imkân bulunmaktadır.

## **1.2. Veri Tabanı Nedir ?**

Veritabanı en genel tanımıyla, kullanım amacına uygun olarak düzenlenmiş veriler topluluğudur. Birbirleriyle ilişkileri olan verilerin tutulduğu, mantıksal ve fiziksel olarak tanımlarının olduğu bilgi depolarıdır. Veritabanları gerçekte var olan ve birbirleriyle ilişkisi olan nesneleri ve ilişkileri modeller.

Veri tabanı, bir kuruluşun uygulama programlarının kullandığı operasyonel verilerin bütünüdür (toplamıdır). Burada; “kuruluş”, bir okul, üniversite, banka, bir üretim şirketi, hastane, devlet kuruluşu, vb. olabilir. “Operasyonel veri” bir kuruluşun çalışabilmesi, işleyebilmesi için kullanılan çok çeşitli verilerdir. Ticari bir şirket için müşteri bilgileri, satış bilgileri, ürün bilgileri, ödeme bilgileri, vb., okul için öğrenci bilgileri, açılan dersler, kimlerin kaydolduğu, öğretmen bilgileri, boş ve dolu derslikler, sınav tarihleri, vb., hastane için hasta bilgileri, doktor bilgileri, yatakların doluluk boşluğu, teşhis-tedavi bilgileri, mali bilgileri, vb ...

Belirli bir konu hakkında toplanmış veriler bir veritabanı programı altında toplanır. İstenildiğinde toplanan bilgilerin tümü veya istenilen özelliklere uyanları görüntülenebilir, yazdırılabilir hatta bilgilerden yeni bilgiler üretilerek bunlar çeşitli amaçlarla kullanılabilir.

Veri tabanı yönetim sistemi(VTYS), yeni bir veritabanı oluşturmak, veri tabanını düzenlemek, geliştirmek ve bakımını yapmak gibi çeşitli karmaşık işlemlerin gerçekleştirildiği birden fazla programdan oluşmuş bir yazılım sistemidir. Veri tabanı yönetim sistemi, kullanıcı ile veri tabanı arasında bir arabirim oluşturmaktadır ve veri tabanına her türlü erişimi sağlar. Veri tabanının tanımlanması: veri tabanını oluşturan verilerin tip ve uzunluklarının belirlenmesidir. Veri tabanını oluşturulması ise veri için yer belirlemesi ve saklama ortamına verilerin yüklenmesini ifade eder. Veri tabanı üzerinde işlem yapmak; belirli bir veri üzerinde sorgulama yapmak, meydana gelen değişiklikleri yansıtmak için veri tabanının güncellenmesi ve rapor üretilmesi gibi işleri temsil eder. Ayrıca veri tabanı yönetim sistemi, verinin geri çağırılmasını sağlar. Veri tabanına yeni kayıt eklemek, eskileri çağırarak ve gerekli düzeltmeleri yapmak yoluyla, verinin bakımını ve sürekliliğini gerçekleştirir, kayıtlara yeni veri eklemek ve yeni kayıtlar oluşturmakla, veri tabanını genişletir.

Bir veritabanından beklenen özellikler, verileri koruması, onlara erişilmesini sağlaması ve başka verilerle ilişkilendirilmesi gibi işlemleri yapabilmesidir. Veritabanı kullanılarak, verilerden daha kolay yararlanılabilir, istenilen veriye çok kolay erişilebilir, çeşitli sorunların çözümünde yardımcı olacak yeni bilgiler üretilir. En önemlisi veriler bir merkezde toplanabilir, herkesin bu verilere yetkileri ölçüsünde erişmesi, düzeltilmesi, silmesi veya görebilmesi sağlanabilir. Böylece veri girişinde ve veriye erişimde etkinlik ve güvenilirlik sağlanır.

Veri tabanı kullanıldığı zaman bir kuruluşa ait tüm operasyonel veriler merkezi bir yerde ve merkezi kontrol altında tutulmuş olur.

Veri tabanlarını kurmayı, yaratmayı, tanımlamayı, işletmeyi ve kullanmayı sağlayan programlar topluluğuna “**veri tabanı sistemi**” ya da “**veri tabanı yönetim sistemi (VTYS) – data base management system (DBMS)**” denir.

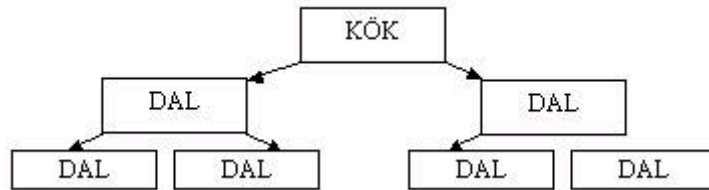
Bir veritabanı üzerinde birden fazla veritabanı bileşeni vardır; bu bileşenler, saklanmak istenen ham bilginin, belli bir formatta alınarak, veri haline gelmesi işleminde etkin rol oynarlar.

VTYS’ler fiziksel hafızayı ve veri tiplerini kullanıcılar adına şekillendirip denetleyen ve kullanıcılarına standart bir SQL arayüzü sağlayarak onların dosya yapıları, veri yapısı, fiziksel hafıza gibi sorunlarla ilgilenmek yerine veri giriş-çıkışı için uygun arayüzler geliştirmelerine olanak sağlayan yazılımlardır. VTYS’de verileri tutmak üzere bir çok türde nesne ve bu nesnelere erişimleri düzenlemek üzere kullanıcılar, roller ve gruplar yer alır. Her bir kullanıcının belli hakları vardır. Bu haklar, kısıtlanabilir. Örneğin bir tablo ya da programcı bir kullanıcı kullanabilirken bir başkasının hakları veritabanı yöneticisi tarafından kısıtlanmış olabilir.

### 1.3. Veri Modeline Göre Veritabanı Yönetim Sistemleri

Yapısal olarak bütün veri tabanları bir değildir. Veri tabanları verileri saklama ve onlara erişme bakımından farklı tiplere ayrılır.

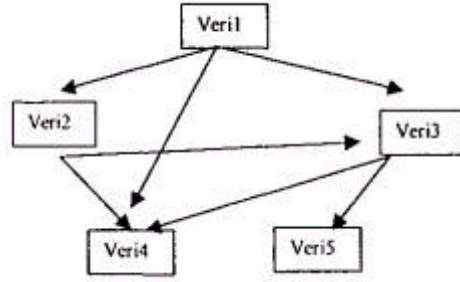
**Hiyerarşik Veri Tabanları :** Bu veritabanı tipi, ana bilgisayar ortamlarında çalışan yazılımlar tarafından kullanılmaktadır. Bu türde en çok kullanılan yazılım, IBM tarafından çıkarılan IMS’ dir. Uzun bir geçmişe sahip olmasına rağmen, PC ortamına uyarlanan hiyerarşik veri tabanları yoktur. Hiyerarşik veri tabanları, bilgileri bir ağaç (tree) yapısında saklar. Kök (Root) olarak bir kayıt ve bu köke bağlı dal (Branch) kayıtlar bu tip veritabanının yapısını oluşturur. Aşağıda böyle bir veri tabanının yapısı gösterilmektedir.



Şekil 1.3.1. Hiyerarşik Veri Tabanı Yapısı

**Ağ Veri Tabanları :** Hiyerarşik veri tabanlarının yetersiz kalmasından dolayı bilim adamlarının ortak çalışması sonucu ortaya konulmuş bir veri tabanı türüdür. Ağ veri tabanları verileri ağaçların daha da gelişmiş hali olan graflar (ağacın kendisinde özel bir graftır.) şeklinde saklarlar. Bu yapı en karışık yapılardan biridir.

**İlişkisel Veri Tabanları :** E.F. Codd Tarafından Geliştirilmiştir. Bu sistemde veriler tablo şeklinde saklanır. Bu veri tabanı yönetim sisteminde; veri alış verişi için özel işlemler kullanılır. Bu işlemlerde tablolar operandlar olarak kullanılır. Tablolar arasında ilişkiler belirtilir. Bu ilişkiler matematiksel bağıntılarla (ilişkilerle) temsil edilir. Günümüzde hemen hemen tüm veri tabanı yönetim sistemleri ilişkisel veri modelini kullanırlar. İlişkisel modeli 1970 yılında Codd önermiştir. Bu model, matematikteki ilişki teorisine ("the relational theory") dayanır. İlişkisel veri modelinde (Relational Data Model) veriler basit tablolar halinde tutulur. Tablolar, satır ve sütunlardan oluşur



**Şekil 1.3.2. İlişkisel Veri Tabanı Yapısı**

**Nesneye Yönelik Veri Tabanları :** Günümüzde nesne kavramı her yerde kullanılmaktadır. Pek çok kelime işlemci ve hesap tablosu programlarının alıştığımız görünümüne artık bir de nesneler eklenmiştir. Ancak bu gerçek anlamda bir nesneye yönelik yazılım demek değildir. Yüzde yüz nesneye yönelik bir yazılımın tamamen nesneye temelli çalışması gerekir. Yazılımın mutlaka nesneye yönelik bir dilde yazılmış olması beklenir. Fakat Windows gibi işletim sistemi üzerinde çalışan yazılımlar bu özelliklere tümüyle sahip değildir. Sadece nesne kavramını kullanarak bazı ek özellikler sunarlar. Nesneye yönelik veri tabanı da , C++ gibi nesneye dayalı bir dille (OOPL) yazılmış olan ve yine C++ gibi nesneye dayalı (OOPL) bir dille kullanılan veri tabanı anlamına gelir. Günümüz teknolojisinde yüzde yüz nesneye yönelik bir veri tabanı yaygın olarak kullanıma sunulmuş değildir. Ancak nesneye yönelik veri tabanlarının bazı üstünlükleri olacağından söz ediliyor. İlişkisel veri tabanları ile karşılaştırıldığında; nesneye yönelik veri tabanlarının sahip olması gereken üstünlükler şunlardır:

1. Nesneler, bir tabloda yer alan bir kayıttan çok daha karmaşık yapıya sahiplerdir ve daha esnek bir yapıda çok daha kullanışlı düzenlenebilirler.
2. Nesneye dayalı bir veri tabanında, yapısı gereği arama işlemleri çok hızlı yapılabilir. Özellikle büyük tablolarla uğraşırken ilişkisel veri tabanlarından çok daha hızlı sonuca ulaşırlar. Ancak çalışma mantığı tümüyle değişir



Tüm bu özellikler tamamen nesneye yönelik olan veri tabanları için geçerlidir. Bazı ilişkisel veri tabanları ile çalışan yazılımlarda da nesnelerin bazı özellikleri ni kullanırlar, ama nesneye yönelik veri tabanı bunu kendini ilişkisel veri tabanı kurallarına uydurarak gerçekleştirebilir

#### **1.4. Neden Veritabanı Kullanılır?**

Bilgisayar ortamında veri saklama ve erişiminde geçmişten günümüze değişik yöntemler ve yaklaşımlar kullanılmıştır. Bunlardan Geleneksel Yaklaşım (Dosya - İşlem Sistemi) verileri ayrı ayrı dosyalarda gruplamaya dayanır. Verileri saklamak için programlama dillerinde kullanılan sıralı (Sequential) ve rastgele (Random) dosyalama sistemleri gibi. Birbiriyle ilgili olan ve aynı gruba dahil olan veriler bir dosyada, bir başka gruba dahil olan veriler de başka bir dosyada tutulurdu. Geleneksel Yaklaşımın birçok sakıncası vardır ve bu sakıncaların beraberinde getirdiği sorunların üstesinden gelebilmek için de Veri Tabanı Yaklaşımı zamanla Geleneksel Yaklaşımın yerini almıştır. Günümüzde veriler artık Veri Tabanı Yaklaşımı ilkesine göre VTYS' lerde tutulmakta ve işlenmektedir.

#### **Geleneksel Yaklaşımın (Dosya - İşlem Sistemi) Sakıncaları**

- Veri tekrarı ve veri tutarsızlığı
- Verinin paylaşılabilmesi
- Uygulamalardaki her yeni gereksinimin ve değişikliğin yalnız uzman kişiler tarafından karşılanabilmesi
- Veriye erişim ve istenen veriyi elde etme güçlükleri
- Karmaşık veri saklama yapıları ve erişim yöntemlerini bilme zorunluluğu
- Bütünlük (integrity) sorunları
- Güvenlik, gizlilik sorunları
- Tasarım farklılıkları, standart eksikliği
- Yedekleme, yeniden başlatma, onarma gibi işletim sorunları

#### **Veri Tabanı Yaklaşımının Yararları**

- Ortak verilerin tekrarının önlenmesi; verilerin merkezi denetiminin ve tutarlılığının sağlanması
- Veri paylaşımının sağlanması
- Fiziksel yapı ve erişim yöntemi karmaşıklıklarının, çok katmanlı mimarilerle kullanıcılardan gizlenmesi
- Her kullanıcıya yalnız ilgilendiği verilerin, alışıktığı kolay, anlaşılır yapılarda sunulması
- Sunulan çözümleme, tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmenin kolaylaşması.
- Veri bütünlüğü için gerekli olanakların sağlanması, mekanizmaların kurulması
- Güvenlik ve gizliliğin istenilen düzeyde sağlanması
- Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına çözüm getirilmesi



### 1.5. Veri Tabanı Yönetim Sistemlerinin Sağladığı Yararlar

- Aynı veri değişik kişilerin PC'lerinde veya değişik bilgisayarlarda tekrar tekrar tutulmaz; **veri tekrarı ("data redundancy")** azaltılır ya da yok edilir.
- **Veri tutarlılığı ("data consistency")** : Aynı verinin değişik yerlerde birkaç kopyasının bulunması "bakım" zorluğu getirir: bir yerde güncellenen bir adres bilgisi başka yerde güncellenmeden kalabilir ve bu durum veri tutarsızlığına ("data inconsistency") yol açar.
- **Veri paylaşımı / Eşzamanlılık ("concurrency")** : Veri tabanı yönetim sistemi (VTYS) kullanılmadığı durumlarda veriye sıralı erişim yapılır. Yani birden çok kullanıcı aynı anda aynı veriye erişemez. Bir VTYS'de ise verinin tutarlılığını ve bütünlüğünü bozmadan aynı veritabanlarına saniyede yüzlerce, binlerce erişim yapılabilir.
- **Veri bütünlüğü ("data integrity")**: Bir tablodan bir öğrenci kaydı silinirse, öğrenci varolduğu diğer tüm tablolardan silinmelidir.
- **Veri güvenliği ("data security")** : Verinin isteyerek ya da yanlış kullanım sonucu bozulmasını önlemek için çok sıkı mekanizmalar mevcuttur. Veri tabanına girmek için kullanıcı adı ve şifreyle korumanın yanı sıra kişiler sadece kendilerini ilgilendiren tabloları ya da tablo içinde belirli kolonları görebilirler.
- **Veri Bağımsızlığı ("data independence")** : Programcı, kullandığı verilerin yapısı ve organizasyonu ile ilgilenmek durumunda değildir. VERİ BAĞIMSIZLIĞI, VTYS'lerinin en temel amaçlarındandır.

### 1.6. Bilinen VTYS Programları

**MS SQL Server:** Bir orta ve büyük ölçekli VTYS'dir. ANSI SQL'e eklentiler yazmak için T-SQL'i destekler.

**Oracle:** Daha çok yüksek ölçekli uygulamalarda tercih edilen bir VTYS'dir. ANSI SQL'e eklentiler yazmak için PL/SQL geliştirilmiştir.

**Sybase:** Bir orta ve büyük ölçekli VTYS'dir. ANSI SQL'e eklentiler yazmak için T-SQL komutlarını destekler. Ülkemizde daha çok bankacılık ve kamusal alanlarda tercih edilmektedir.

**Informix:** Bir orta ve büyük ölçekli VTYS'dir.

**MySQL:** Genellikle Unix-Linux temelli Web uygulamalarında tercih edilen bir VTYS'dir. Açık kod bir yazılımdır. Küçük-orta ölçeklidir. Özellikle Web için geliştirilmiş bir VTYS'dir denilebilir.

**PostgreSQL:** Bu da MySQL gibi açık kod bir VTYS'dir.

**MS Access:** Çoklu kullanıcı desteği yoktur. İşletim sisteminin sağladığı güvenlik seçeneklerini kullanır. Bunun yanında belli sayıda kayda kadar (1000000 civarı) ya da belli bir boyutun (yaklaşık 25MB) altına kadar bir sorun çıkartmadan kullanılabilecek bir küçük ölçekli VTYS'dir.

**Advantage:** Türk programcılar tarafından geliştirilen bir orta ve büyük ölçekli VTYS'dir.

**DB/2:** IBM'in framework'lere yönelik büyük ölçekli VTYS'dir.

Bu ders notunda popüler olan VTYS programlarının isimleri yazılmıştır. Bunların dışında daha bir çok VTYS programı mevcuttur. VTYS'lerin Avrupa genelindeki pazar payları yaklaşık olarak aşağıda listelenmiştir. En büyük Pazar payı IBM(DB/2) ile Oracle arasındadır. Hemen arkasından MS SQL Server, Informix ve Sybase gelmektedir. Yeni başlayanlar için; hangi VTYS'yi öğrenmem en iyisi olur sorusunu yanıtlamak gerekebilir. Ülkemizde insan kaynakları açısından en çok kalifiye elaman aranan VTYS Oracle ve arkasından da MS SQL Server gelmektedir.

IBM .....	%37.8
Oracle .....	%26.3
Microsoft .....	%15.4
Informix .....	%3.2
Sybase .....	%3
Digerleri .....	%14.3

**2001 yılında bir araştırmaya göre Avrupa çapında VTYS'lerinin pazar payları [kaynak: Gartner]**

VTYS'lerin bir çoğu ANSI SQL'in karşılayamadığı durumlarda kullanılmak üzere ek programlama komutları barındırırlar. Bu iş için MS SQL Server ve Sybase SQL Server Transact SQL (T-SQL) denilen komut takımlarını içerir. Oracle ise PL/SQL ile bu işe çözüm getirmiştir. Bu diller sayesinde, bu konu içerisinde öğrenmeyeceğimiz Stored Procedure (saklı prosedürler), Trigger, Fonksiyon gibi veritabanları için vazgeçilmez olan nesneler yazılabilmektedir.

### **1.7. Proje ve VTYS arasındaki ilişki**

Herhangi bir veritabanı programında çalışmaya başlanılmadan önce, yapılacak işe uygun veri tabanı tasarımı yapılmalıdır. Bu işin en önemli aşamasıdır. Başlangıçta iyi tasarlanamayan bir veritabanı, ileride geriye dönüşü olmayan verimsiz bir bilgi yığınına dönüşebilir. En basit hali ile veritabanı tasarımında; hangi tabloların olacağı, bu tablolarda hangi alanların olacağı, tablolar arasındaki alan ilişkilerinin neler olacağı ve alanlara ait özelliklerin tanımlanması yapılır. Alan özelliklerinde alan adı, alan tipi, alanın uzunluğu, alanın varsayılan değeri, bu alana yazılacak verilerin geçerlilik koşullarının başlangıçta tasarlanması gerekir.

Bir projede hangi veritabanının seçileceği, projenin çapı ile ilgili bir karardır. Aşağıdaki sorulara verilecek cevaplar projenin çapı konusunda karar vermede yardımcı olurlar.

- Projede kaç tablo kullanılacak?
- Her bir tabloda en fazla kaç satır yer alabilir? (tablodaki bilgi sayısıdır)
- Projeye aynı anda en fazla kaç kullanıcı bağlanacak?
- Proje günlük kaç transaction (INSERT-DELETE-UPDATE) gerçekleştirecek?
- Proje en fazla ne kadarlık yer kaplayacak ne kadarlık bir veritabanı dosyasına ihtiyaç duyulacak?
- Proje için güvenlik ne derece önemli? Ancak bir VTYS kullanılarak proje geliştirilecekse, hangisinin seçilmesi gerektiğinin dışında, hangi sürümlerinin kullanılacağı ya da hangi donanımlar üstünde çalıştırılacağı da önemlidir.