

# 1. VERİ TABANI YÖNETİM SİSTEMİ KAVRAMI

## Birlikte Düşünelim

1. Dosya nedir? Temel dosya tipleri nelerdir?
2. Neden Veri Tabanı Yönetim Sistemleri bu kadar sık kullanılmaktadır?
3. Veri modeli nedir? Başlıca veri modelleri nelerdir?
4. Klasik veri saklama biçimi nasıl olmaktadır?
5. Sıralı dosyalar hangi durumlarda yararlıdır?
6. İndeksli dosyalarla hesaba dayalı dosyalar arasındaki temel fark hangisidir?

## Başlamadan Önce

Bu bölümde dosya yapıları ve Veri Tabanı Yönetim Sistemleri hakkında genel bilgiler verilecektir. Veri Tabanı Yönetim Sistemleri'ni konu alan derslerde klasik dosya yapılarının da anlatıldığı görülmektedir. Bunun sebebi okuyucuların Veri Tabanı Yönetim Sistemleri'nden önceki yapıları öğrenerek Veri Tabanı Yönetim Sistemleri'ne neden gereksinim duyulduğunu anlamasını sağlamaktır. Klasik dosya yapılarının eksiklikleri fark edilerek Veri Tabanı Yönetim Sistemleri'nin bu eksiklikleri nasıl giderdiği anlaşıldığında Veri Tabanı Yönetim Sistemleri konusundaki kavramları anlamak da daha kolay bir hale gelmektedir. Bu bilgilerden hareketle de bu bölümde önce Veri Tabanı Yönetim Sistemleri kavramları ele alınacak daha sonra sık kullanılan Veri Tabanı Yönetim Sistemleri anlatılacaktır.

### 1.1. Veri Tabanı Yönetim Sistemleri Kavramları

Veri Tabanı Yönetim Sistemleri, yazılımlarda kullanılan verilerin saklanması, kaydedilmesi, filtrelenmesi, sıralanması, sorgulanması ve kullanılması amacıyla geliştirilen özel bir yazılım türüdür. Yazılımların az miktarda veri içerdiği eski dönemlerde bu verilerin düz dosyalar halinde saklanması ve elle ya da küçük kod parçaları kullanılarak sorgulanması yeterli olmaktaydı. Ancak yazılımların gelişmesi, kullanılan veri ve dosya sayılarının artması sonucunda karmaşık dosyalara ve dosyalar arasında çoklu ilişkilere ihtiyaç duyulmuş ve Veri Tabanı Yönetim Sistemleri (VTYS - kısaca veri tabanı olarak da adlandırılır) ortaya çıkmıştır. Veri Tabanı Yönetim Sistemi kendi içerisinde, klasik yazılım yapılarından farklı yapılara sahiptir ve Veri Tabanı Yönetim Sistemi mantığının anlaşılabilmesi için öncelikle bunların anlaşılması gerekmektedir. Bu yapılar kayıt, alan, dosya yapıları ve Veri Tabanı Yönetim Sistemi olarak ifade edilebilir. Dosya yapıları da kendi içerisinde klasik dosyalar, sıralı dosyalar, dizinli dosyalar, hesaba dayalı dosyalar olarak sınıflandırılabilir.

#### 1.1.1. Kayıt ve Alan Kavramları

Veri Tabanı Yönetim Sistemleri'ndeki ve diğer veri saklama birimlerindeki veri toplulukları olan dosyalarda farklı özellikteki veriler bulunabilmektedir. Burada farklı özellikten kast edilen bir verinin sayısal, metinsel, tarih-saat vb. şeklinde olmasıdır.

Verilerin bu özelliklerine alan, her veri grubuna ise kayıt adı verilir. Dosyalar birçok kayıt, kayıtlar ise

birçok alan içerebilir. Her kayıt ve alan disk üzerinde yer anlamına gelmektedir. Bir Veri Tabanı Yönetim Sistemi'ne kayıt ya da alan eklendiğinde kapsadığı alan büyüyecek, kayıt ya da alan silindiğinde kapsadığı alan küçülecektir. Bu sebeple mümkün olduğunca az alan ve kayıt kullanmakta fayda bulunmaktadır. Öğrencilerin bilgilerini tutan bir dosya ele alındığında dosyadaki her kayıta Öğrenci No, Ad-Soyad, Baba Adı, Doğum Yeri ve Doğum Tarihi gibi bilgilerin bulunması beklenir. Bahsedilen bu tablonun adının ÖĞRENCİ Tablosu olması halinde kayıt – alan yapısı Şekil 1.1'deki gibi olacaktır.

### ÖĞRENCİ Tablosu

Öğrenci No	Ad – Soyad	Baba Adı	Doğum Yeri	Doğum Tarihi	Kilo	Geçti/Kolalı
1306	Rüya	İsmail	Ankara	1983	andalık	True / False
1308	Eylem		Kayseri	1979		1, 0
1310	Selçuk	Orhan		1978		Evet / Hayır
1320	Zeynep			1979		
1312	Sibel	Ali	İstanbul			
1330	Emel	Osman		1977		
1313	Özgür	Kadir	Rize	1982		

Şekil 1.1. Kayıt – Alan Yapısı Örneği

Bir veri tabanı tablosunda Şekil 1.1'de de görüldüğü gibi her kayıta yer alan her alanın doldurulması zorunlu değildir. Hangi alanların doldurulmasının zorunlu olduğu, Veri Tabanı Yönetim Sistemi'nin kullanıldığı uygulamaya ve ihtiyaca göre değişmektedir.

### 1.1.2. Dosya Yapıları

Araç-gereç, teçhizat, ulaşım vb. konulardaki tüm teknolojiler zaman geçtikçe insanların ihtiyaçlarından ötürü geliştirilmiştir. Aynı şekilde yazılımlarda kullanılan veriler konusunda da insanların ihtiyaçları artmış ve kullanılan verileri saklamak, depolamak için daha farklı yapılar kullanılması gerekmiştir. Bunlar arasındaki en temel yapılardan biri, birbirinden bağımsız elemanlarda kullanılmak üzere hazırlanan ve sıralı ya da doğrudan erişim yöntemleri kullanılarak işlenen klasik dosya yapılarıdır. Sıralı erişim, bir dosyada bir kayda ulaşmak için dosyadaki ilk kayıttan itibaren tüm kayıtların taranması; doğrudan erişim ise, kayıtlara tek tek bakılmayıp istenen kayda direkt olarak erişilmesi olarak ifade edilebilir. Sıralı dosyalar (ardışık dosyalar olarak da adlandırılır), sıralı erişim kullanan dosyalardır. Bu tür dosyaların kayıtlarına ardışık olarak erişilebilmesine karşılık, kayıtlar fiziksel olarak (bilgisayar belleğinde) ardışık gözlerde yer almak zorunda değildir. Kayıtların fiziksel olarak ardışık olması kimi zaman faydalı kimi zaman zararlı olabilir. Örneğin, bir okuldaki öğrencilerin tümü listelenecek ise, dosyanın sıralı olarak tasarlanmış olması faydalı olabilecektir, ancak sadece birkaç öğrencinin listelenmesi isteniyorsa, bu yöntem uygun bir yöntem olmayacaktır. Böyle bir durumda söz konusu öğrenciye ya da öğrencilere doğrudan erişilmesi en uygun yoldur. Diğer bir dosya çeşidi olan dizinli dosyalarda (indeksli dosyalar olarak da adlandırılır) verilerin bulunduğu dosyaya ek olarak ayrı bir dizin dosyası oluşturulur. Dizin kavramı, anahtarlar ve anahtarların bilgisayar üzerindeki fiziksel adresi içerir. Anahtar ise, her kayıt için özgün nitelik taşıyan, tekrarlanmayan alan olarak ifade edilebilir. Örnek olarak, öğrenci bilgilerini içeren yukarıdaki gibi bir öğrenci dosyasında, aynı ad ve soyada sahip, aynı baba adına sahip, aynı doğum yerine sahip ve aynı doğum tarihine sahip öğrencilerin olması olası bir durumdur. Ancak aynı öğrenci numarasına sahip öğrenci olma ihtimali yoktur. Bu sebeple öğrenci numarasının anahtar olarak kullanılması en uygun yoldur. Genellikle dosyalarda sayısal ifadeler (öğrenci no, kimlik no, pasaport no vb.) anahtar olarak kullanılır çünkü bu sayısal ifadeler kayıtlara özgü olarak ve unik bir şekilde verilmektedir. Dosyaların herhangi bir kaydına doğrudan doğruya erişebilmek istendiğinde hash fonksiyonları ile oluşturulan bir hesaplama algoritması kullanılır.

### 1.1.3. Veri Tabanı Yönetim Sistemleri

Geliştirilen yazılımlardaki veri sayısının ve çeşidinin artması ile bir süre sonra dosyalar arasındaki ilişki miktarı da artmış ve çok sayıda dosya arası ilişki ve kullanıcıların dosyalara erişimi söz konusu olduğunda geleneksel dosya sistemlerinin yetersiz kaldığı görülmüştür. Bu sorunu çözmek üzere veriyi saklama ve erişim konusunda yeni yazılım teknolojiler araştırılmaya ve geliştirilmeye başlanmış ve ortaya Veri Tabanı Yönetim Sistemleri çıkmıştır. Geleneksel dosya yapılarında, dosyalar ile kullanılan yazılımlar arasında bağlantı ve ilişki bulunduğundan dosya yapılarında ortaya çıkabilecek en ufak bir değişiklik bile yazılımların/uygulama programlarının değişmesine ve yeniden derlenmesine/çalıştırılmasına neden olmaktadır. Hâlbuki Veri Tabanı Yönetim Sistemleri'nde verilerin girişi ve depolanması, veriye erişen yazılımlardan/uygulama programlarından bağımsızdır ve Veri Tabanı Yönetim Sistemleri'ndeki bir değişiklik yazılımda/uygulama programlarında değişikliğe neden olmayacağı gibi yazılımda/uygulama programlarında değişiklik de Veri Tabanı Yönetim Sistemleri'nde değişikliğe neden olmaz. Bu da yeni geliştirilen yazılımlarda, sıklıkla aranmakta olan taşınabilirlik, uyumluluk, bağımsızlık, fonksiyonellik gibi özelliklerin oluşmasına yardımcı olabilen bir durumdur.

VTYS → Yazılım  
Uygulama

Veri Tabanı Yönetim Sistemleri, günümüzde yazılımların önemli bir bileşenidir.

Her yazılım çeşitli veriler kullanmaktadır ve veriler üzerinde gerekli işlemleri başarılı bir şekilde yapamayan bir yazılımın başarılı olma şansı çok düşüktür. Veri Tabanı Yönetim Sistemi kullanımı, geleneksel dosya kullanımına göre birçok yönden üstünlük ve fayda sağlamaktadır. Veri Tabanı Yönetim Sistemleri'nin faydalarından bazıları aşağıda verilmiş ve açıklanmıştır:

- Verilerin tekrarlanması önler.
- Verilerin tutarlı olmasını sağlar.
- Verilerin güvenliğini sağlar.

**Verilerin Tekrarlanması Önler:** Geleneksel dosya sistemlerinin kullanıldığı yazılımlarda, özellikle yazılım boyutu büyükse, yazılımın hantallaşmasını önlemek için yazılımlar genellikle alt sistemlere bölünür ve her bir alt sistemin kendi veri dosyaları bulunurdu. Bu da her alt-yazılım için farklı veri kümeleri bulunması anlamına gelmekte idi. Fakat bu veriler esasında kökte aynı yazılıma hitap ettiğinden büyük oranda tekrarlı veri olmakta idi. Örneğin Şekil 1.1'de verilen bir ÖĞRENCİ Tablosu, bir öğrenci işleri bürosunda bir öğrencinin hem kayıt işlemlerinde, hem ders işlemlerinde, hem de diğer işlemlerinde kullanılacaktır. Dolayısıyla eğer öğrencilerin kayıt işlemleri ve ders işlemleri ayrı birer alt-yazılım olarak düşünülürse bu durumda her alt-yazılımda bu veri tabanı tablosunun tekrar kullanımı söz konusu olacaktır. Veri Tabanı Yönetim Sistemleri'nde ise bu tekrarlara gerek olmaz ve tüm alt-yazılımlar aynı dosyaya kolaylıkla erişebilir. Veri Tabanı Yönetim Sistemleri, yazılımların bir bütün olarak tasarlanmasını, yazılımların sahip olduğu alt-yazılımlar arasında gerekli ilişkilerin kurulmasını ve birden çok alt-yazılımda ve hatta ana-yazılımda verilerin aynı Veri Tabanı Yönetim Sistemi içinde ortak olarak kullanılmasını sağlar.

**Verilerin Tutarlı Olmasını Sağlar:** Veri Tabanı Yönetim Sistemi'ndeki verilerin doğruluğunu ve tutarlı bir bütün oluşturmasını sağlamak için bir takım kısıtlamalara yer verilebilir. Örneğin bir ÖĞRENCİ veri tabanı tablosuna öğrenci bilgileri girilirken, öğrenci numarasının, öğrencinin bağlı bulunduğu fakülte ve bölümün formatına göre girilmesi sağlanabilir. Öğrenci numarasında ilk iki hane fakülteyi, ikinci iki hane bölümü, sonraki dört hane öğrencinin giriş yılını ifade ediyorsa, yazılımcı ve kullanıcıların da bu şekilde girmesine izin verilip, başka bir formatta girmesine izin verilmemesi sağlanabilir. Yine vize, final, bütünleme gibi sınav notlarının girilmesi esnasında sadece 0-100 arasında bir değer girmelerine izin verilip, farklı bir değere izin verilmemesi sağlanabilir. Bir öğrencinin yıl içerisinde aldığı notların toplamının %100 (veya öğretim üyesinin izin verdiği başka bir değer) olması gerekmektedir, Veri Tabanı Yönetim Sistemi bunu kontrol edip başka bir değere izin verilmemesini sağlayabilir. Bu örnekleri çoğaltmak mümkündür.

**Verilerin Güvenliğini Sağlar:** Özellikle devlet verileri, kişisel veriler, sağlık verileri, malî veriler gibi daha hassas verilerin kullanıldığı bazı yazılımlarda, yazılımların ürettiği verilerin güvenliğini sağlamak çok önemli bir konudur. Veri Tabanı Yönetim Sistemleri’nde farklı seviyede kullanıcılar bulunmaktadır ve her kullanıcının, Veri Tabanı Yönetim Sistemi’nin içerdiği tüm bilgilere kolayca erişmesi istenmez. Örneğin öğrencilerin harç işlemleri ile ilgilenen bir memurun öğrencinin ders notlarına erişme ve değiştirme yetkisine sahip olmasına gerek yoktur. Bu ve buna benzer örnekler veri güvenliği konusunda açıklar ortaya çıkarabilir. Bunu çözebilmek için kullanıcılar arasında hiyerarşik bir düzen bulunmalı ve her kullanıcının yetkisi ayrı olmalıdır. Diğer bir deyişle her kullanıcının erişebileceği veriler ayrı ayrı tanımlanmalıdır. Veri Tabanı Yönetim Sistemleri, erişimlerin kısıtlanması için geniş olanaklar sunduğundan bu güvenliğini sağlamaktadır.

Veri Tabanı Yönetim Sistemleri yöneticilerinin sıklıkla karşılaştığı zorluklar ise şu şekilde ifade edilebilir:

- Veri hacmindeki artışlardan etkilenmemek
- Veri güvenliğini sağlamak
- Talebe ayak uydurmak
- Veri Tabanı Yönetim Sistemi ve altyapı yönetimini sağlamak ve bakım yapmak
- Ölçeklenebilirlik sınırlarını kaldırmak

### Veri hacmindeki artışlardan etkilenmemek:

Günümüzde birçok farklı noktadan elde edilen ve miktarı olağanüstü şekilde artan veriler, Veri Tabanı Yönetim Sistemi yöneticilerinin, şirketlerinin verilerini verimli bir şekilde yönetmek ve organize etmek üzere sürekli olarak mücadele etmesini gerektirmektedir.

### Veri güvenliğini sağlamak:

Günümüzde herhangi bir yerde ve zaman diliminde, veri ihlali yaşanabilmekte ve bilişim suçu işlenmektedir. Hacker'ların da her geçen gün daha yaratıcı bir hale geldiği düşünülecek olursa veri güvenliğinin garanti altına alınması ve aynı zamanda kullanıcıların verilere kolayca erişebilmesi, her zamankinden daha önemli bir hale gelmiştir ve bu durum Veri Tabanı Yönetim Sistemi yöneticilerinin önemli görevlerinden biri olmuştur.

### Talebe ayak uydurmak:

Günümüzün hızlı değişkenlik gösteren iş, akademi, sektör ortamlarında şirketlerin, karar alma süreçlerini zamanında desteklemek ve yeni fırsatlardan faydalanmak için verilere gerçek zamanlı olarak erişebilmesi gereklidir. Bu da Veri Tabanı Yönetim Sistemi yöneticilerinin bu konuda hızlı ve etkin çözümler bulmasını gerektirir.

### Veri Tabanı Yönetim Sistemi ve altyapı yönetimini sağlamak ve bakım yapmak:

Veri tabanı yöneticileri, sürekli olarak sorunlar açısından Veri Tabanı Yönetim Sistemi'ni izlemeli, önleyici bakım işlemleri gerçekleştirmeli ve gerekli olduğu takdirde yazılım yükseltmeleri ve yamaları uygulamalıdır.

### Ölçeklenebilirlik sınırlarını kaldırmak:

Günümüzde işletmeler, sektörde rekabet edebilmek ve ayakta kalabilmek için sürekli büyümelidir. Tabi ki bu büyüme ile birlikte veri yönetimi çözümleri de genişlemelidir. Veri Tabanı Yönetim Sistemi yöneticilerinin şirketin ne kadar kapasiteye ihtiyaç duyacağını tahmin etmesi oldukça zor bir işlemdir ve bu konuda sürekli olarak çalışmalıdır.

## Şekil 1.2. Veri Tabanı Yönetim Sistemleri Yöneticilerinin Sıklıkla Karşılaştığı Zorluklar

Her Veri Tabanı Yönetim Sistemi belirli bir veri modeli kullanır. Bir veri tabanı yapısının temelini, kullandığı veri modeli oluşturmaktadır. Verilerin artmaya ve Veri Tabanı Yönetim Sistemleri'nin geliştirilmeye başladığı dönemlerden itibaren pek çok veri modeli geliştirilmiştir. Bu modeller 4 ana grupta toplanabilir:

- Sıradüzensel/Hiyerarşik Veri Modeli
  - Ağ Veri Modeli
  - İlişkisel Veri Modeli \*
  - Nesneye Yönelik Veri Modeli
- } Kullanılmıyor.

Bu veri modellerinin ilk ikisi günümüzde kullanılmamaktadır. En yaygın kullanılan ise İlişkisel Veri Modeli'dir. Günümüzde kullanılan Veri Tabanı Yönetim Sistemleri'nin hemen hemen tümü İlişkisel Veri Modeli'ne dayalıdır. Son zamanlarda ortaya çıkan Nesneye Yönelik Veri Modeli de yine İlişkisel Veri Modeli ile birlikte bazı Veritabanı Yönetim Sistemleri'nde kullanılmaktadır.



## 1.2. Sık Kullanılan Veri Tabanı Yönetim Sistemleri

Bir yazılımdaki verilerin hangi Veri Tabanı Yönetim Sistemi ile modellenmesi gerektiği pek çok kritere bağlı olarak değişmektedir. Bu kriterlerden bazıları: Veri Tabanı Yönetim Sistemi yapısının, geliştirilen yazılım yapısına uygunluğu, Veri Tabanı Yönetim Sistemi'nde kullanılan veri ve veri türlerinin, geliştirilen yazılımdaki veri ve veri türlerine uygunluğu, Veri Tabanı Yönetim Sistemi'ne verilen teknik destek, daha önceki tecrübeler, geliştirilecek olan tablo yapılarının Veri Tabanı Yönetim Sistemi'ne uygunluğu şeklindedir. Günümüzde birçok firmanın geliştirdiği çok sayıda Veri Tabanı Yönetim Sistemi bulunmaktadır. Bunlar arasında en sık kullanılanlar aşağıda tanıtılmıştır:

- Microsoft (MS) SQL Server
- Oracle
- Sybase
- MySQL
- PostgreSQL
- Microsoft (MS) Access

Açık Kaynak

**MS SQL Server:** Microsoft firması tarafından geliştirilen, orta ve büyük ölçekli bir Veri Tabanı Yönetim Sistemi'dir. Sürekli olarak güncellemesi gerçekleştirilen bir Veri Tabanı Yönetim Sistemi olduğundan günümüzde pek çok firmada sıklıkla kullanılmaktadır. T-SQL'i destekleyen MS SQL Server sürümleri geçmişten günümüze Tablo 1.1'de gösterilmektedir.

**Tablo 1.1.** Geçmişten Günümüze MS SQL Server Sürümleri

Versiyon	Yıl	Sürüm Adı	Kod Adı
1.0 (OS/2)	1989	SQL Server 1.0 (16bit)	-
1.1 (OS/2)	1991	SQL Server 1.1 (16bit)	-
4.21 (WinNT)	1993	SQL Server 4.21	SQLNT
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7.0	1998	SQL Server 7.0	Sphinx
-	1999	SQL Server 7.0 OLAP Tools	Plato
8.0	2000	SQL Server 2000	Shiloh
8.0	2003	SQL Server 2000 64-bit Edition	Liberty
9.0	2005	SQL Server 2005	Yukon
10.0	2008	SQL Server 2008	Katmai
10.25	2010	SQL Azure	Matrix (nam-ı diğer CloudDB)
10.5	2010	SQL Server 2008 R2	Kilimanjaro (nam-ı diğer KJ)
11.0	2012	SQL Server 2012	Denali

**MS SQL** Server, farklı amaçlara hizmet eden bir Veri Tabanı Yönetim Sistemi olduğundan farklı versiyonları bulunmaktadır. **Temel versiyonlar**, **Datacenter**, **Enterprise**, **Standard**, **Web**, **Workgroup**, **Express**; **özelleştirilmiş versiyonlar** ise **Azure**, **Compact**, **Developer**, **Embedded**, **Evaluation**, **Fast Track** ve **Parallel Data Warehouse** şeklindedir.

- **Datacenter:** MS SQL Server Datacenter versiyonu, veri merkezleri için tasarlanmış, yüksek seviye

uygulama desteği ve ölçeklenebilirlik ihtiyaçlarını gideren, 256 mantıksal işlemciyi ve neredeyse sınırsız hafızayı destekleyen bir versiyondur.

- **Enterprise:** MS SQL Server Enterprise versiyonu, SQL Server kümelerini yaratmak ve düzenlemek için gerekli araçları içeren bir versiyondur, çekirdek Veri Tabanı Yönetim Sistemi motoru ve add-on servislerini desteklemektedir. 524 PB Veri Tabanı Yönetim Sistemi yönetebilmekte, 2 TB hafıza içermekte ve 8 fiziksel işlemci desteklemektedir.
- **Standard:** MS SQL Server Standard versiyonu, çekirdek Veri Tabanı Yönetim Sistemi motorunu, bağımsız servislerle birlikte içermektedir. Enterprise versiyonundan farklı olarak daha az aktif kümedeki ağ sayısı destekler, sunucu çalışırken hafıza eklenebilmesi gibi yüksek süreklilik fonksiyonlarını ve paralel indeksleri içermez.
- **Web:** MS SQL Server Web versiyonu, Web hosting için toplamda sahip olma maliyeti bir seçenektir.
- **Workgroup:** MS SQL Server Workgroup versiyonu, çekirdek Veri Tabanı Yönetim Sistemi fonksiyonlarını içermekte ancak ek servisleri içermemektedir.
- **Express:** MS SQL Server Express versiyonu, çekirdek veri tabanı motorunu içeren, ücretsiz bir versiyondur. Veri tabanı ve kullanıcı sayısında bir sınırlama yoktur, ancak 1 işlemci, 1 GB hafıza ve 4 GB Veri Tabanı Yönetim Sistemi dosya limiti bulunmaktadır.
- **Azure:** MS SQL Server Azure versiyonu, MS SQL Server'ın bulut bazlı versiyonudur. Azure Servisleri Platform'unda servis olarak yazılım olarak sunulmuştur.
- **Compact:** MS SQL Server Compact versiyonu, gömülü bir Veri Tabanı Yönetim Sistemi versiyonudur. Diğer MS SQL Server versiyonlarının aksine, SQL CE, SQL Mobile (başlangıçta mobil cihazlar için tasarlanmıştır) üzerine kuruludur ve aynı ikili değerleri paylaşmamaktadır. Boyutunun küçüklüğüne ek olarak, özellik setleri belirgin olarak azaltılmıştır, 4 GB Veri Tabanı Yönetim Sistemi limiti bulunmaktadır.
- **Developer:** MS SQL Server Developer versiyonu, MS SQL Server Datacenter versiyonuyla aynı özellikleri içermektedir, ancak sadece geliştirme (development) ve test sistemi olarak kullanılabilir.
- **Embedded:** MS SQL Server Embedded versiyonu, MS SQL Server Express Veri Tabanı Yönetim Sistemi'nin, sadece belli Windows Servisleri tarafından ulaşılabilen, özel olarak düzenlenmiş bir halidir.
- **Evaluation:** MS SQL Server Evaluation versiyonu, Enterprise versiyonunun tüm özelliklerini içermektedir ancak sadece 180 günle sınırlıdır ve bu sebeple Deneme Sürümü olarak bilinir.
- **Fast Track:** MS SQL Server Fast Track versiyonu, özel olarak, ticarî kurum bazında, veri depolama ve zekâ işleme (intelligence processing) işlemlerinde kullanılan bir MS SQL Server versiyonudur.
- **Parallel Data Warehouse:** MS SQL Server Parallel Data Warehouse sürümü, yüzlerce terabaytlık veri depolama işlemleri için optimize edilmiş versiyondur.

**Oracle:** Çoğunlukla daha yüksek ölçekli uygulamalarda tercih edilen bir Veri Tabanı Yönetim Sistemi'dir. Oracle Veri Tabanı Yönetim Sistemi verilerin tümünü güvenli bir biçimde tutup on binlere aktarabilen bir yapı olduğundan bankacılık gibi yüksek güvenliğe ihtiyaç duyan uygulamalarda kullanılır. Oracle Veri Tabanı Yönetim Sistemi üzerinden uygulamalar yönetebildiği gibi yeni uygulamalar geliştirmek de mümkündür.

Bu noktada Oracle Application Express (Oracle Apex) büyük ölçüde ihtiyaç duyulacak bir destektir ve hazır uygulamalarla, kullanıcıların ihtiyaç duyacağı düzenlemeleri yapmayı sağlar. Veri Tabanı Yönetim Sistemi merkezli uygulama geliştirmelerinde ve web tabanlı yazılımlarda kullanılır. Veri Tabanı Yönetim Sistemi üzerinde çalışan ve ayrıca bir uygulama sunucu kurulumuna gerek duymaz. Oracle Veri Tabanı

Yönetim Sistemi da MS SQL Server gibi farklı standartlar altında kullanıcılara sunulur. Bunlar arasında kullanıcının, ihtiyaç duyduğu özellikler ve dolayısıyla seçeceği Veri Tabanı Yönetim Sistemi paketinde var olan özellikler fiyat yönünden belirleyici unsur olur. Örneğin bir işlemcinin kullanılabildiği, düşük kullanım alanı sunan ve çeşitli uygulama izinleri bulunmayan bir Oracle Veri Tabanı Yönetim Sistemi versiyonu ücretsiz olarak denenebilir. Ancak daha güçlü, daha işlevsel ve daha büyük verileri işleyecek bir Veri Tabanı Yönetim Sistemi'ne ihtiyaç varsa Oracle resmi sayfasından Veri Tabanı Yönetim Sistemi türlerini incelemek mümkündür. Ayrıca alınan Veri Tabanı Yönetim Sistemi'nin lisans süresi de değişebilmektedir. 1 yıl ile 5 yıl arasında lisans almak mümkün olabilmektedir. Oracle'da ANSI SQL'e eklentiler yapmak için PL-SQL adlı dil geliştirilmiştir.

**Sybase:** SAP tarafından 2010 yılında geliştirilen orta ve büyük ölçekli bir Veri Tabanı Yönetim Sistemi'dir. ANSI SQL'e eklentiler yazmak için T-SQL komutlarını destekler. Ülkemizde daha çok bankacılık ve kamusal alanlarda tercih edilmektedir. MS SQL Server ile komutlar açısından benzerlik göstermektedir.

**MySQL:** Genellikle Unix - Linux temelli web uygulamalarında tercih edilen bir Veri Tabanı Yönetim Sistemi'dir. Açık kaynak kodlu (open source) bir yazılımdır. Küçük-orta ölçeklidir. MySQL, SQL tabanlı bir açık kaynak ilişkisel Veri Tabanı Yönetim Sistemi'dir. Web uygulama yazılımları için tasarlanıp optimize edilmiştir ve tüm platformlarda çalıştırılabilir. İnternet ile yeni ve farklı gereksinimlerin ortaya çıkmasıyla birlikte MySQL, web geliştiricileri ve web tabanlı uygulama yazılımları için tercih edilen seçenek hâline gelmiştir. Milyonlarca sorguyu ve binlerce işlemi gerçekleştirmek üzere tasarlandığından MySQL, çoklu para transferlerini yönetmesi gereken e-ticaret işletmeleri için popüler bir tercih olmuştur. MySQL Airbnb, Uber, LinkedIn, Facebook, Twitter ve YouTube gibi dünya genelindeki başlıca web sitelerinin ve web tabanlı uygulama yazılımlarının ardındaki Veri Tabanı Yönetim Sistemi olarak karşımıza çıkmaktadır.

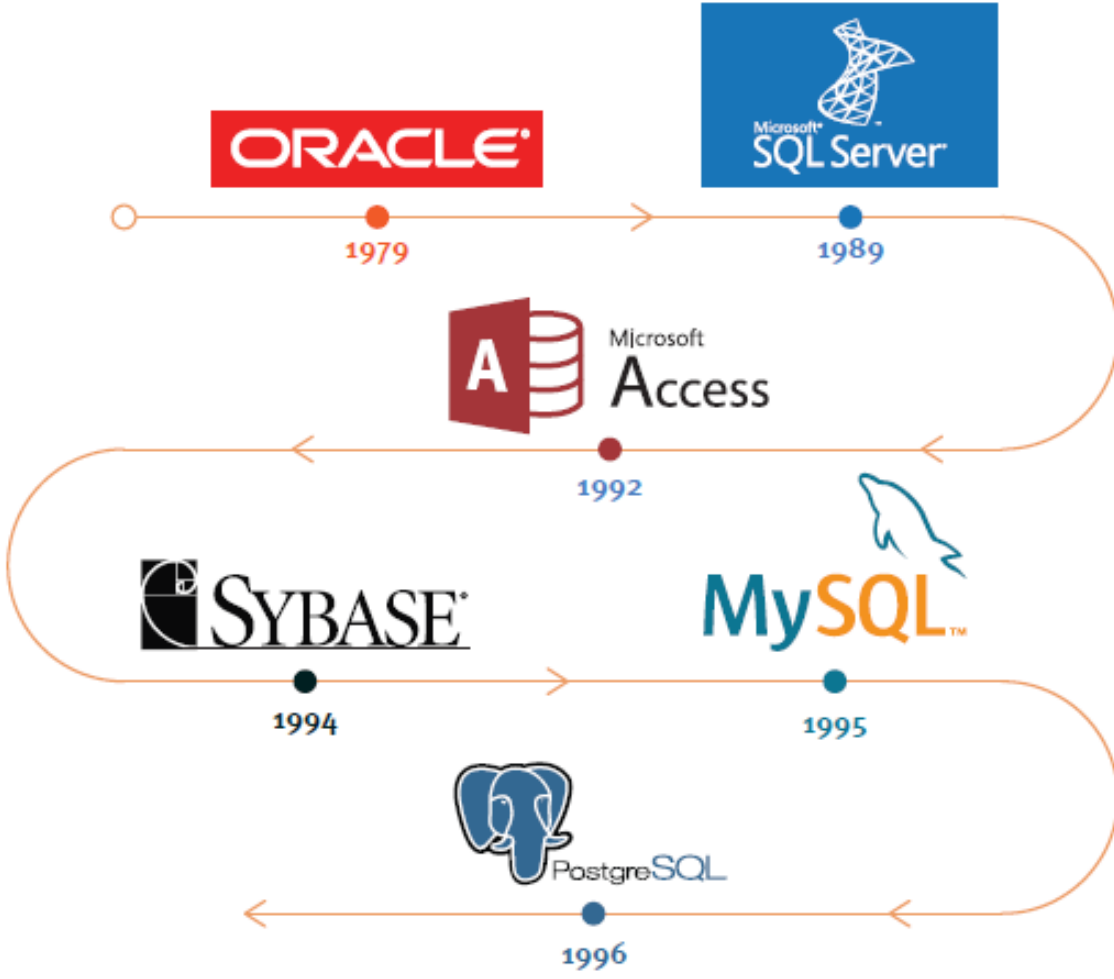
## PostgreSQL

**PostgreSQL:** MySQL gibi açık kaynak koda sahip bir Veri Tabanı Yönetim Sistemi'dir. PostgreSQL, güvenilirlik, özellik sağlamlığı ve performans açısından kendisine güçlü bir itibar kazandıran 30 yılı aşkın aktif geliştirme sürecine sahip güçlü, açık kaynaklı bir nesne ilişkisel Veri Tabanı Yönetim Sistemi olarak ifade edilmektedir.

**MS Access:** MS SQL Server gibi Microsoft tarafından geliştirilen bir Veri Tabanı Yönetim Sistemi'dir. Ancak çoklu kullanıcı desteği olmadığı, işletim sisteminin sağladığı güvenlik seçeneklerini kullandığı, belli sayıda kayda kadar (1 milyon civarı) ya da belli bir boyutun (yaklaşık 25 MB) altına kadar bir sorun çıkarmadan kullanılabileceği için ancak küçük çaplı uygulamalarda kullanılabilir.

Hiçbir programlama dili için diğerinden iyi ya da kötü denilemeyeceği gibi, hiçbir Veri Tabanı Yönetim Sistemi için de diğerinden iyi ya da kötü denilemez. Kullanıcı, yazılım geliştirici ve diğer paydaşlar, yazılımın, kullanılan verilerin, projedeki kaynakların durumuna göre o projeye en uygun Veri Tabanı Yönetim Sistemi'ni seçmektedir. En çok kullanılan bu Veri Tabanı Yönetim Sistemleri'nin özellikleri özet halinde Tablo 1.2'de verilmiştir.





Şekil 1.3. En Çok Kullanılan Bu Veri Tabanı Yönetim Sistemleri

## Bölüm Özeti

- Yazılımlar bu kadar büyümeden, kullanılan verilerin miktarı ve türü artmadan önce verileri düz, klasik dosyalar halinde saklamak yeterliydi ancak karmaşık dosya yapıları ve çok sayıda dosyalar arası ilişki oluşmaya başlayınca geleneksel dosya sistemi yetersiz kalmış ve dolayısıyla farklı yaklaşımlar aranmaya başlanmıştır. Veri Tabanı Yönetim Sistemleri yaklaşımı bu amaçla ortaya çıkmıştır.
- Veri Tabanı Yönetim Sistemleri, birçok yazılımda kullanılan verilerin tutulması, saklanması, kaydedilmesi, gerektiğinde filtrelenmesi, sorgulanması ve kullanılması amacıyla geliştirilen özel bir yazılım türüdür.
- Veri Tabanı Yönetim Sistemleri’ni doğru bir şekilde kullanmak, yazılımların da başarısını arttıracığından büyük önem arz etmektedir. Ancak Veri Tabanı Yönetim Sistemleri’ni başarılı bir şekilde kullanabilmek için birçok kavramı iyi bir biçimde bilmek gereklidir.
- Veri Tabanı Yönetim Sistemi sistemlerinin temelini oluşturan kayıt, alan, sıralı dosyalar, dizinli dosyalar gibi temel kavramları anlamak gerekmektedir.
- Veri Tabanı Yönetim Sistemleri’nde veriler tablolar halinde tutulur. Bu tablolara eklenen her bir elemana kayıt, bu kayıtların özelliklerini ifade eden her sütuna ise alan adı verilmektedir.

• Bir Veri Tabanı Yönetim Sistemi’ndeki tabloda tablonun içeriği ile ilgili alanlar ve kayıtlar olması beklenir. Örneğin ÖĞRENCİ isimindeki bir tabloda alan olarak SGK No bulunması beklenmez çünkü ÖĞRENCİ isimli bir tabloda kaydedilecek kişilerin çalışan özelliği değil öğrenci özelliği taşıması beklenir. Bu sebeple de o tabloda SGK No değil Öğrenci No alanı bulunması beklenmektedir. Buna göre bir Veri Tabanı Yönetim Sistemi’ni tasarlarken bu konulara dikkat edilmeli ve uygun alanlar seçilmelidir. Bu sayede başarılı sorgular oluşturulabilir ve başarılı bir yazılım kullanımı sağlanabilir.

## Kaynakça

Kadir Çamoğlu, “Programlama ve Veritabanı Mantığı”, Kodlab, 1. Baskı, 2009

Rıfat Çölkesen, “Veri Yapıları ve Algoritmalar”, Papatya Yayıncılık, 9. Basım, 2014

Yalçın Özkan, “Veritabanı Sistemleri”, Papatya Yayıncılık, 4. Basım, 2015

WEB: Kadir Peker, <https://koraypeker.com/2019/03/16/modern-veri-tabanlari/>

WEB: <https://www.mertmekatronik.com/en-cok-kullanilan-veritabanlaridatabase>