

B Ö L Ü M

1

Veri Tabanına Giriş ve Temel Kavramlar



1.1 Veri Tabanı Nedir?

TANIM

Veri tabanı; birbiriyle ilişkisi olan verilerin tutulduğu, kullanım amacına uygun olarak düzenlenmiş veriler topluluğunun mantıksal ve fiziksel olarak tanımlarının bulunduğu bilgi depolarıdır.

Aynı kavram farklı açılardan da tanımlanabilir. Aşağıdaki ifadelerin hepsi veri tabanını doğru biçimde anlatır:

- ◆ Sistematik erişim imkânı olan, yönetilebilir, güncellenebilir, taşınabilir ve birbirleri arasında tanımlı ilişkiler bulunabilen bilgiler kümesidir.
- ◆ Genellikle bir bilgisayar sisteminde elektronik olarak depolanan, yapılandırılmış bilgi veya veriden oluşan düzenli bir koleksiyondur.
- ◆ Bir veri tabanı sistemi donanım ve yazılıma dayanır, veriyi işler, kullanıcılarla etkileşim kurar ve prosedürlerle düzenlenir.

Veri tabanının amacı, insanların ve organizasyonların bir şeyleri takip edebilmesine yardımcı olmaktır. Basit bir web uygulamasından uluslararası kuruluşların büyük ve karmaşık verilerine kadar pek çok alanda veri tabanı uygulamalarına ihtiyaç duyulur.

Veri Tabanı Örnekleri

Üniversite	Öğrenci işleri bilgi sistemi
Hastane	Hasta, doktor, tedavi, araç-gereç ve mali bilgiler
Ticari şirket	Müşteri, ürün, satış, ödeme ve teslimat bilgileri
Banka	Müşteri, mevduat, kredi kartı ve kredi bilgileri

Temel Kavramlar

Veri yönetimi	Verilerin uygun şekilde üretilmesine, depolanmasına ve alınmasına odaklanan bir disiplindir. Herhangi bir işletme, devlet kurumu veya hizmet kuruluşu için temel niteliğinde bir etkinliktir.
Yapısal veri	Bilgilerin saklanması, kullanılmasını ve üretilmesini kolaylaştırmak için biçimlendirilmiş verilerdir.
Yapılandırılmamış veri	Orijinal, ham durumunda olan verilerdir.



Veri tabanları günlük hayatın her yerindedir: öğrenci işleri, hastane, market ve banka

Metadata (üst veri)

Veri tabanındaki tabloların, sütunların, veri türlerinin, ilişkilerin, anahtarların ve diğer nesnelerin özelliklerini açıklayan yapısal verilerdir.

VTYS

Veri tabanlarını saklamak ve yönetmek amacıyla tasarlanmış bir yazılım paketidir (Veri Tabanı Yönetim Sistemi).

1.2 Neden Veri Tabanı Kullanılır?

Veri tabanı kullanılmadığında bir şeylerin kaydını tutmak için **listeler** kullanılır: alışveriş listesi, yapılacak işler listesi, ödenen faturaların listesi gibi. Verilerin tutulması, saklanması ve erişilmesinde geleneksel yaklaşım da verilerin ayrı ayrı dosyalarda gruplanmasına dayanır. Veriler arttıkça, verilere aynı anda erişme ve düzenleme ihtiyacı doğdukça bu yaklaşım yetersiz kalır.



Solda dağınık listeler, sağda aynı bilgilerin düzenli bir tabloda tutulması

Aşağıdaki liste bir şirketin müşterilerini, çalıştığı firmaları ve satışlarını tek bir yerde tutmaktadır:

SATIŞ LİSTESİ				
Müşteri adı	Mesleği	Firma	Adres	Fiyat
Hasan	İşçi	A	Abc Cad. 43/5	100 TL
Selim	Öğretmen	B	Abc Mah. A Blok 8/4	2050 TL
Serkan	Mühendis	C	Abc Mah. 6/7	3500 TL
Rıza	Öğretmen	B	Abc Mah. A Blok 8/4	70 TL
Salih	Öğretim elemanı	B	Abc Mah. A Blok 8/4	140 TL
Ali	Emekli	D	Abc Mah. 6/7	190 TL
Elif	Avukat	A	Abc Cad. 43/5	1050 TL

Listelerle İlgili Sorunlar

Güncelleme sorunu

B firmasının adresi değişirse üç satırdaki adres bilgisi de değişmelidir. Biri eksik kalırsa **bilgi tutarsızlığı** oluşur; bu hem zaman kaybına hem de hataya yol açar.

Silme sorunu

A firması artık şirketle çalışmıyorsa ve firmaya ait kayıtlar silinirse, ürün satışı yapılan müşteri bilgisi ve firma adresi gibi bilgiler de kaybedilir.

Paylaşım sorunu

Farklı departmanlar listenin farklı bölümlerine ihtiyaç duyar: iletişim departmanı firma ve adresi, pazarlama departmanı firma ve fiyatı, müşteri hizmetleri müşteri adı, mesleği ve firmayı görmek ister. Bilgilerin tamamının herkesle paylaşılması güvenlik ve müşteri gizliliği gibi nedenlerle sakıncalıdır.

Listelerin en büyük sakıncası, **farklı temalarla ilgili bilgileri bir arada bulundurmasıdır**. Yazılı bir metinde farklı konular için farklı paragraflar açıldığı gibi, farklı temalardaki bilgiler de farklı tablolara yerleştirilmelidir. Bu sürece **normalizasyon** denir (ayrıntısı 4. bölümde).

Çözüm: Temaları Ayırmak ve ilişkilendirmek

Liste; müşteriler, çalışılan firmalar ve satış bilgileri olmak üzere üç tabloya ayrılır. Hangi müşterinin hangi firmadan ürün aldığı gibi soruları cevaplayabilmek için tablolar arasında **ilişki** kurulur: her müşteriye ve firmaya bir numara verilir, satış tablosu bu numaraları kullanır.

MÜŞTERİ			SATIŞ		
No PK	Müşteri adı	Mesleği	Müşteri No FK	Firma No FK	Fiyat
1	Hasan	İşçi	1	1	100 TL
2	Selim	Öğretmen	2	2	2050 TL
3	Serkan	Mühendis	3	3	3500 TL
4	Rıza	Öğretmen	4	2	70 TL
5	Salih	Öğretim elemanı	5	2	140 TL
6	Ali	Emekli	6	4	190 TL
7	Elif	Avukat	7	1	1050 TL

FİRMA		
No PK	Firma	Adres
1	A	Abc Cad. 43/5
2	B	Abc Mah. A Blok 8/4
3	C	Abc Mah. 6/7
4	D	Abc Mah. 6/7

Müşteri ve Firma tablolarındaki **No** alanları birincil anahtardır (PK). Satış tablosundaki **Müşteri No** ve **Firma No** alanları ise bu tablolara bağlanan yabancı anahtarlardır (FK). Artık B firmasının adresi tek bir yerde durur; değiştiğinde yalnızca bir satır güncellenir.

SONUÇ

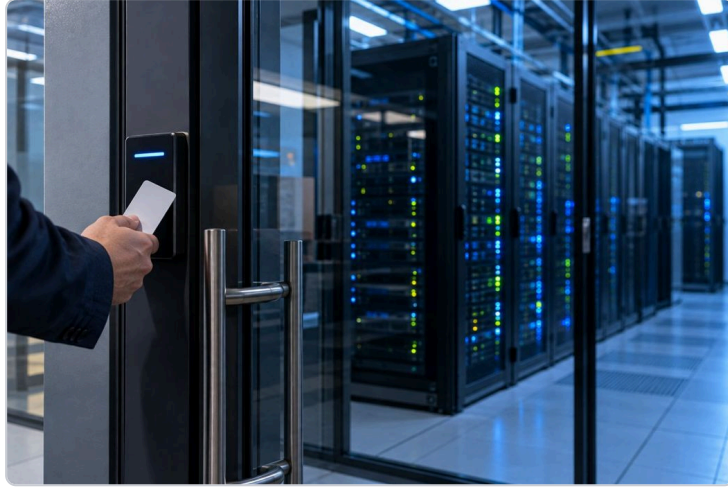
Farklı temalar farklı tablolara yerleştirildiğinde değişen, silinen ve paylaşılan bilgilerle ilgili sorunların çoğu ortadan kalkar. Baştaki bütün listeyi yeniden elde etmek için tablolar **SQL** dili ile birleştirilir.

1.3**Veri Tabanı Yaklaşımının Avantajları**

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

ÖZET

Veri tabanı sistemlerinin temel özellikleri şöyle özetlenebilir: tekrarlı ve tutarsız veri kontrolünün sağlanması, verilere yetkili olmayan kişilerin erişiminin engellenmesi, veriler arasında karmaşık bağlantıların gösterilebilmesi, **etkin sorgu yapılarının kullanılması**, programlar ile verilerin birbirinden ayrılması.



Verilere yetkili olmayan kişilerin erişimi engellenir

1.4 Veri Tabanı Yönetim Sistemi (VTYS)

TANIM

Veri tabanı yönetim sistemi; yeni bir veri tabanı oluşturmak, veri tabanını düzenlemek, kullanmak, geliştirmek ve bakımını yapmak için çeşitli karmaşık işlemlerin gerçekleştirildiği bir yazılım sistemidir.

Yaygın kullanılan veri tabanı yönetim sistemlerinden bazıları: Oracle Database, IBM DB2, Microsoft SQL Server, Microsoft Access, MySQL, PostgreSQL, SQLite, Informix, Teradata, Adaptive Server Enterprise, Microsoft Visual FoxPro, Progress.

Veri Modeline Göre Sınıflandırma

Hiyerarşik

Veri tabanları için kullanılan ilk modeldir. Bilgiler bir **ağaç yapısında** saklanır.

Ağ

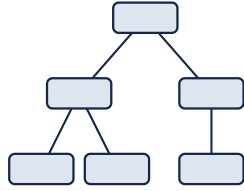
Hiyerarşik veri tabanları yetersiz kalınca 1960'ların sonunda ortaya çıkmıştır. Veriler, ağaçların daha gelişmiş hâli olan **graflar** şeklinde saklanır.

İlişkisel

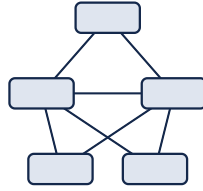
1970'lerin başında geliştirilmiştir. Veriler **tablo** şeklinde saklanır; tablolar arasındaki bağlantılar matematiksel ilişkilerle gösterilir. Günümüzdeki hemen hemen tüm veri tabanı programları bu yapıdadır.

Nesneye yönelik

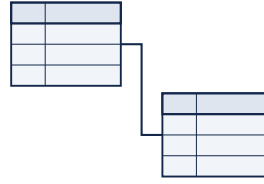
Nesneye yönelik bir dille (örneğin C++) oluşturulan ve yine bu tarz bir dille kullanılan veri tabanıdır. Kelime işlemci ve hesap tablosu programlarında kullanılan nesneler veri tabanlarında da kullanılır.



Hiyerarşik
ağaç yapısı



Ağ
graf yapısı



İlişkisel
tablolar ve ilişkiler



Nesneye Yönelik
nesneler

Veri modeline göre veri tabanı yönetim sistemleri

HATIRLATMA

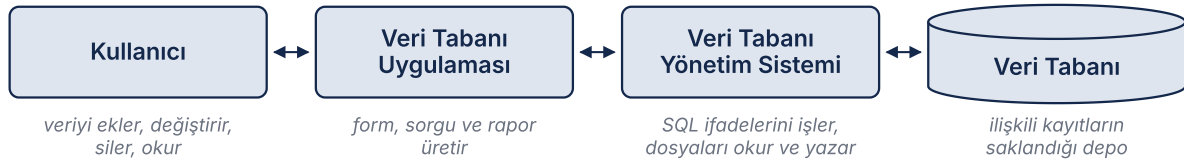
Birbiriyle ilişkili, yani **PK-FK yoluyla** birbirine bağlanan tablolardan oluşan veri tabanı modeline **ilişkisel veri tabanı modeli** denir. Bu derste kullanılan Microsoft SQL Server ilişkisel bir veri tabanı yönetim sistemidir.

Kullanıcı Sayısına Göre Sınıflandırma

Veri tabanı yönetim sistemleri kullanıcı sayısına göre **tek kullanıcı** ve **çok kullanıcı** olarak ikiye ayrılır.

1.5 Veri Tabanı Sisteminin Bileşenleri

Bir veri tabanı sistemi dört bileşenden oluşur: kullanıcı, veri tabanı uygulaması, veri tabanı yönetim sistemi ve veri tabanı.



Veri tabanı sisteminin bileşenleri

Kullanıcı

İşlerini gerçekleştirmek için veri tabanını kullanır: yeni veri ekler, mevcut verileri değiştirir, verileri siler; formlar, sorgular ya da raporlar yoluyla verileri okur.



Kullanıcı formu doldurur; veri tabanı uygulaması bu bilgiyi VTYS'ye iletir

Veri tabanı uygulaması

VTYS ile kullanıcı arasında iletişimi sağlayan bir ya da birden fazla bilgisayar programıdır. Kullanıcıdan verileri alır ya da kullanıcıya gönderir; kullanıcının davranışlarını VTYS'ye iletilecek isteklere dönüştürür.

VTYS

Uygulamadan gelen istekleri alır ve veri tabanı dosyaları üzerinde okuma ya da yazma yaparak gerçekleştirir. SQL cümlelerini okur ve bunları işletim sisteminin dosyalar üzerinde uygulayacağı yönergelere dönüştürür.

Veri tabanı

İlişkili kayıtlardan oluşan veri deposudur.

Uygulamanın ve VTYS'nin İşlevleri

Veri tabanı uygulaması	Veri tabanı yönetim sistemi
Formlar oluşturur ve formları işler.	Veri tabanını ve tabloları oluşturur.
Sorgular oluşturur ve VTYS'ye iletir (genellikle SQL ile).	Veri tabanından veri okur ve verileri günceller.
Raporlar oluşturur; sorgu sonuçlarını rapor biçiminde sunar.	Veri değerlerine ilişkin sınırlamaları (kısıtları) uygular.
Uygulama mantığını gerçekleştirir.	Bir kullanıcının işleminin diğer kullanıcıyı engellemesini önler.
Uygulamayı kontrol eder.	Kullanıcıların yetkileri ölçüsünde işlem yapmasına izin verir.
	Veri tabanındaki verileri yedekler.

UYGULAMA MANTIĞI ÖRNEĞİ

Kullanıcı 10 birimlik bir sipariş verdi, ancak stokta 8 birim var. Ne olacağına **uygulama programı** karar verir; uygun mantığın gerçekleştirilmesi uygulamanın görevidir. Süreçte bir hata oluşursa kullanıcıya gerekli mesajı gösteren de yine uygulamadır.

1.6 Veri Tabanının Yapısı: Tablo, Kayıt, Alan

Bir veri tabanı, tablolarda saklanan verilerden oluşur. **Tablo**, verilerin satırlar ve sütunlar hâlinde düzenlenmesiyle oluşan veri grubudur. Tablodaki her bir satır **kayıt**, her bir sütun **alan** olarak adlandırılır.

Örneğin ders içeriği ve öğrenci bilgilerini saklamak için iki tablo oluşturulur:

Oğrenci_bilgileri ve **icerik**. Öğrenci bilgileri tablosunda öğrenci numarası, adı soyadı, doğum tarihi, doğum yeri ve e-posta adresi yer alacaksa tablo aşağıdaki gibi olur:

ALAN (SÜTUN)
↓

Ogr_no	Ad_soyad	d_tarih	d_yeri	e-mail
1	Ayşe Türk	01.11.1999	Karaman	ayse@ghj.edu.tr
2	Sema Gül	24.05.1995	Antep	sema@dfg.edu.tr
3	Serdar Efe	06.06.1993	Adıyaman	serdar@xyz.edu.tr
4	Mehmet Yaman	11.02.1998	Nevşehir	mehmet@abc.edu.tr
5	Zerrin Işık	22.08.1990	Elazığ	zerrin@abc.edu.tr
6	Ulviye Polat	12.12.1994	İzmir	ulviye@xyz.edu.tr

KAYIT (SATIR) →

TABLO: Oğrenci_bilgileri

Tablo, kayıt (satır) ve alan (sütun)

Her alan için önceden bir **veri türü** seçilir: sayısal, tarih-saat, metinsel gibi. Örneğin personel sicil numarası tam sayı olmalı, ad soyad harflerden oluşmalıdır. Veri türleri 5. bölümde ele alınmıştır.

Anahtar (Key)

Anahtar, bir veya birden fazla alanın bir satır için niteleyici olmasını zorunlu kılan bir kısıtlayıcıdır. İki çeşit anahtar vardır:

Birincil anahtar (Primary Key, PK)

Bir kayda ulaşmayı sağlayan anahtar veridir. Öğrenciler arasında iki Ahmet varsa, aranan Ahmet'i bulabilmek için her öğrenciye özel bir numara olmalıdır: öğrenci numarası. Birden fazla alan birlikte de birincil anahtar olabilir.

**Yabancı anahtar
(Foreign Key, FK)**

Bir tabloya girilebilecek kayıtları, başka bir tablonun belli alanındaki verilerle **sınırlandırmaya ve ilişkilendirmeye** yarar. Örneğin not tablosundaki her satıra, öğrenci bilgileri tablosundaki bir öğrenci numarası ile eşleşen değer girilmelidir.

1.7 Veri Tabanında Neler Bulunur?

Veri tabanı yalnızca kullanıcı verilerinden oluşmaz. İlişkili kayıtların yanında şu yapıları da barındırır:

Üst Veri (Metadata)

Veri tabanının yapısı hakkındaki verilere **metadata** denir: tablo isimleri, sütun isimleri, tablo ve sütunların özellikleri gibi. 1.2'deki örnek veri tabanı için metadata şöyle görünür:

TABLolar

Tablo no	Tablo adı	Sütun sayısı	Satır sayısı
1	Müşteriler	3	7
2	Firmalar	3	4
3	Satışlar	3	7

SÜTUNLAR

Sütun no	Sütun adı	Veri türü	Uzunluğu	Tablo no
1	id	int	4	1
2	Mus_adi	char	50	1
3	meslegi	char	50	1

Bazı veri tabanları **uygulama üst verisi** de içerir; bunlar formlar ve raporlar gibi uygulama bileşenlerini tanımlar.

İndeks

İndeksler, veri tabanının performansını artırmak için kullanılır. Hangi kayıtların nerede bulunduğunu gösterir; mantığı kitapların başındaki ya da sonundaki dizine benzer.



İndeks, bir kitabın dizini gibidir: aranan bilginin yerini gösterir

| Saklı Yordam (Stored Procedure)

Saklı yordamlar, bir tabloya bağlı olmaksızın veri tabanı içinde tanımlanan ve belirli bir işi yapan, **derlenmiş SQL cümlecikleridir**. Yazıldıkları anda derlendikleri için optimize edilmiştir ve en hızlı şekilde çalışmaya hazırdır. Örnek: bir tablodaki verilerin yedeğini alan ya da bir yıldan eski verileri temizleyen saklı yordam.

| Tetikleyici (Trigger)

Bir tablo üzerinde belirli bir olaya bağlı olarak tetiklenip çalışan SQL kodlarına tetikleyici denir. Tetikleyen olaylar **INSERT, UPDATE ve DELETE** olaylarıdır. Tipik kullanım yeri: stok hareketleri sonucunda stok miktarının kendiliğinden azaltılması veya artırılması.



Stok hareketi: ürün depoya girdiğinde ya da çıktığında stok miktarı tetikleyici ile güncellenir

NOT

Saklı yordamlar da tetikleyiciler de veri tabanını sunan **sunucu üzerinde** çalışır. Bu sayede veriler istemci ile sunucu arasında gidip gelmez; sunucudan istemciye en az veri çekilir. İstemci-sunucu mimarisinin güçlü bileşenleridir (Oracle, MS SQL Server, Sybase, Firebird vb. tarafından desteklenir).

Veri Tutarlılığı (Referential Integrity)

PERSONEL tablosunda kişinin bölümü **BOLUM_NO** alanında, bölümün adı ise BOLUM tablosunda tutulsun. 1 numaralı bölüm herhangi bir personelde kullanılmışsa, BOLUM tablosundan BOLUM_NO değeri 1 olan kayıt **silinmemelidir**. Bu tür denetimlerle veri bütünlüğünün korunmasına veri tutarlılığı denir. Veri tutarlılığını sağlamak için yabancı anahtar kısıtları ve tetikleyiciler kullanılır.

1.8 Ölçeğine Göre Veri Tabanı Sistemleri

Ölçek	Örnek	Öne çıkan ihtiyaç
Tek kişi	Boyacı: Kimin evi, ne zaman, ne kadar boyandı? Hangi renk ve stiller kullanıldı? Kim kime referans oldu?	Tek kullanıcı; kişinin kendi işini takip etmesi yeterlidir.
Küçük işletme	Müzik enstrümanı kiralama: Hangi enstrüman kaç kiralandı? En çok hangisi kiralaniyor? Kiralamayı kim yaptı?	Çok kullanıcıdır; aynı enstrüman aynı anda iki farklı satıcı tarafından seçilememelidir.
Büyük kurum	Sürücü lisansı ve kayıt bürosu: 52 merkez; kazalar, trafik ihlalleri, lisans yenileme ve kısıtlamalar.	Yüzlerce kullanıcı (kayıt, hukuk, maliye personeli); haftada 7 gün 24 saat erişilebilirlik.



Tek kişi: boyacı



Küçük işletme: enstrüman kiralama



Büyük kurum: çok sayıda gişe ve kullanıcı