

Veritabanı ve Yönetim Sistemleri

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi
Elmadağ Meslek Yüksekokulu

Ders İzlenesi

Hafta	Modüller/İçerik/Konular
1. Hafta	Temel Kavramlar
2. Hafta	
3. Hafta	
4. Hafta	
5. Hafta	
6. Hafta	
7. Hafta	
8. Hafta	
9. Hafta	
10. Hafta	
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

Veritabanı Nedir?

- **Veri tabanları** birbirleriyle ilişkili bilgilerin depolandığı alanlardır.
- **Bilgi artışıyla** birlikte bilgisayarda bilgi depolama ve bilgiye erişim konularında yeni yöntemlere **ihtiyaç** duyulmuştur.
- Veri tabanları; **büyük** miktardaki bilgileri depolamada **geleneksel yöntem** olan “dosya-işlem sistemine” alternatif olarak geliştirilmiştir.



Veritabanı Nedir?

- Birbiriyle ilişkisi olan verilerin tutulduğu,
- Kullanım amacına uygun olarak düzenlenmiş veriler topluluğunun,
- Mantıksal ve fiziksel olarak tanımlarının bulunduğu
- Belirli bir veri modeline göre çalışan
- Bilgi depolarıdır.



Veritabanı Nedir?

Veritabanı Örnekleri :

- **Üniversite-** Öğrenci İşleri Bilgi Sistemi
- **Hastane-**Hasta, doktor, tedavi, araç-gereç, mali bilgiler
- **Ticari bir şirket-** Müsteri, Ürün, Satış, Ödeme, Teslimat bilgileri
- **Banka-Müşteri,** mevduat, kredi kartı, kredi bilgileri



Veritabanı Yönetim Sistemi Nedir?

- Yeni bir veritabanı oluşturmak,
- Veritabanını düzenlemek
- Kullanmak,
- Geliştirmek
- Bakımını yapmak için
- Çeşitli karmaşık işlemlerin gerçekleştirildiği bir yazılım sistemidir.
- Belirli bir veri modeline göre çalışırlar
- VTYS, birbirleri ile ilişkili veri ve programlar topluluğundan oluşmaktadır.



Neden VTYS ?

- VTYS yaklaşımında veri girişi ve depolanması veriye erişen uygulama programlarından bağımsızdır.
- Klasik dosya kullanımında ise, kayıt desenleri ve dosya yapılarında ortaya çıkabilecek en ufak bir değişiklik bile uygulama programlarının değiştirilmesine neden olmaktadır.



Neden VTYS ?

- Gereksiz veri tekrarını önler.
 - Tüm uygulamaların gereksinim duyduğu veriler birbirleri ile bütünleşik yapıdadır. Farklı uygulamaların ortak kullanabileceği veri tabloları tasarlanarak veri tekrarı önlenmiş olur.
 - *Örneğin, Türkiye'deki il kodları ve isimleri hem personel alt sisteminde hem de pazarlama alt sisteminde ayrı ayrı tutulmaz.*

Neden VTYS ?

- Veri bütünlüğünü (data integrity) sağlar.
 - Veri bütünlüğü, verinin doğru ve tutarlı olmasıdır.
 - Veri bütünlüğü için bazı kısıtlamalar tanımlanabilir.
 - *Örneğin, çalışanların maaş bilgilerinin asgari ücretten az girilememesi sağlanabilir.*

Neden VTYS ?

- Verilerin güvenliğini sağlar.
 - Tüm verilere her kullanıcının kolayca erişebilmesi çoğu zaman istenmeyen bir durumdur.
 - Her kullanıcıya çeşitli yetkiler atanarak, bu kullanıcının erişebileceği, değiştirebileceği ve silebileceği veriler ayrı ayrı tanımlanabilir.
 - *Örneğin, pazarlama bölümünde çalışan bir kullanıcının diğer personelin özlük bilgilerine ulaşması engellenebilir.*

Neden VTYS ?

- Veriler üzerinde merkezi denetim sağlar.
 - Kullanıcılar işletim sistemi komutları ya da genel amaçlı programlama dilleri ile yazılmış uygulama programlarını kullanarak doğrudan veri tabanındaki verilere erişemezler ve bu verileri değiştiremezler.
 - Veri tabanına erişim yalnızca VTYS (DBMS - Data Base Management System) olarak adlandırılan yazılım sistemi aracılığıyla mümkündür.

Neden VTYS ?

- Veri tabanının bilgisayar belleklerdeki fiziksel yapısı kullanıcılardan gizlenir.
 - Kullanıcılara daha yalın mantıksal yapılar sunulur.
 - VTYS, bir anlamda yüksek düzeyli programlama dili derleyicisi gibi davranarak kullanıcının, soyut terimler kullanarak veri tabanı ile ilişki kurmasını sağlar.
 - Böylece kullanıcı, sistem tarafından kullanılan karmaşık veri gösterimleri ve algoritmaların ayrıntılarıyla uğraşmadan, neyin yapılmasını istediğini belirterek isteklerini ortaya koyabilir.

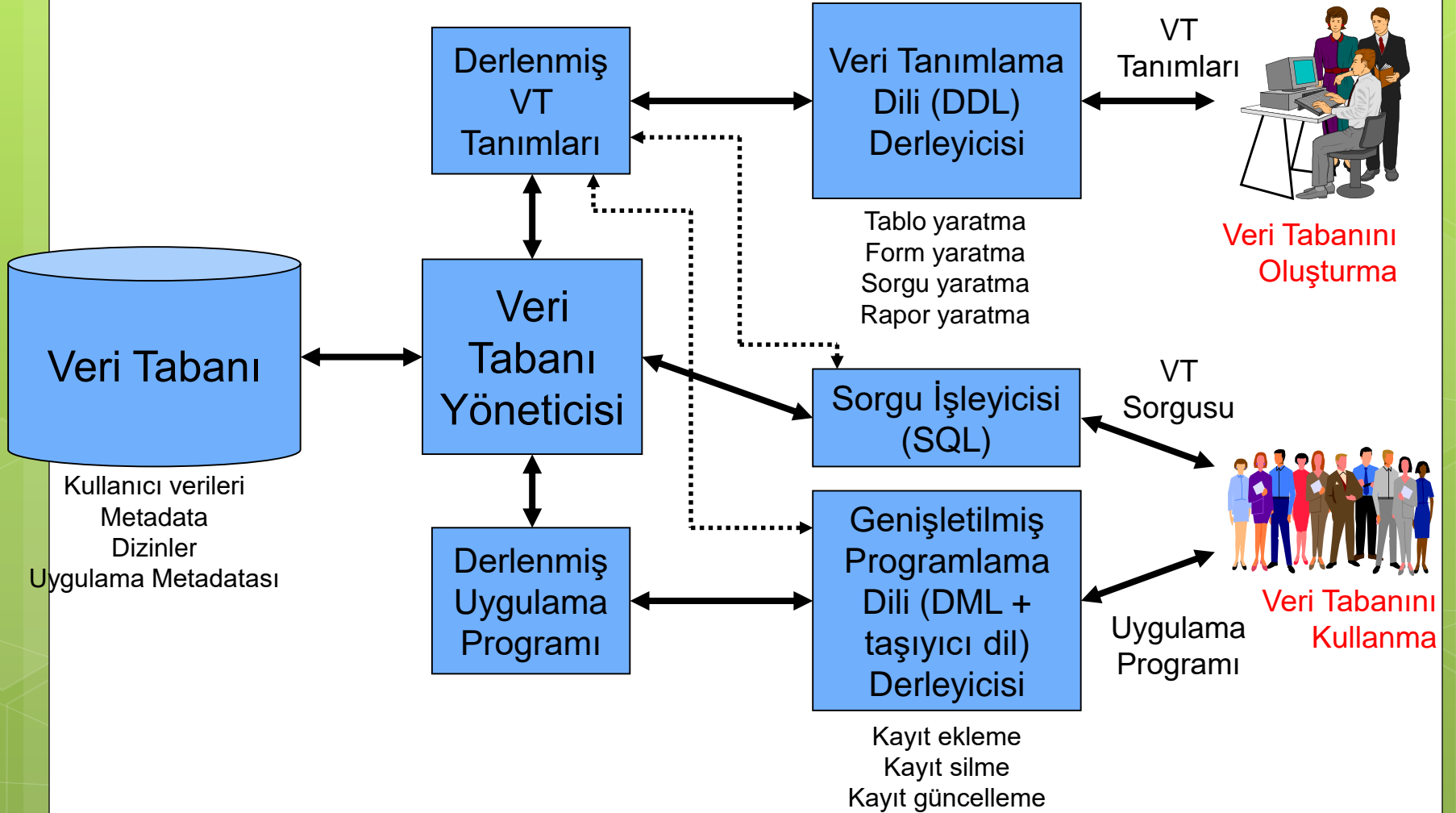
VTYS Çeşitleri

- **Oracle, MS SQL Server, IBM DB2, ...**
 - Büyük miktarda verinin güçlü bir sunucu üzerinde saklanması ve aynı anda çok sayıda kullanıcının bu veriler üzerinde işlem yapmasına izin veren güçlü VTYS'lerdir
 - Veriye erişim ve güvenlik ile ilgili kullanıcı yönetimi (user management) yapısı içerirler
- **MySQL**
 - Web tabanlı uygulamaların en çok kullandığı VTYS'lerden biridir.
 - Mobil cihazlarda kullanılan iOS ve Android işletim sistemleri üzerinde çalışabilir ve kullanıcı yönetimi yapısını içerir
- **Access**
 - Aynı anda çok sayıda kullanıcının aynı veriler üzerinde işlem yapması gerekmeyen durumlarda kullanılan, sisteme fazla yük getirmeyen VTYS'dir.
 - Kullanıcı yönetimi içermediği için veriye erişim ve güvenlik uygulama programı tarafından idare edilmelidir.

Nezaman VTYS Kullanmalıyız?

- VTYS kullanmak bilgisayara ek yük getirdiği için aşağıdaki gibi bazı durumlarda dosya kullanmak daha avantajlı olacaktır:
 - Saklayacağımız veri miktarı çok az ise
 - Sınırlı saklama kapasitesine sahip gömülü sistemlerde
 - Genellikle verinin tamamını sıralı olarak okuyup, yine tamamını sıralı olarak yazmamız gereken uygulamalarda (Büyük miktardaki verinin içinden restgele erişim ile küçük bir veri elde etmeye ihtiyacınız yoksa)

VTYS'nin Temel Bileşenleri (1)



Veri Tanımlama Dili (VTD)

- Veri tabanı tanımlarını VTYS'ye iletmek için kullanılan biçimsel dile Veri Tanımlama Dili (DDL: Data Definition Language) adı verilir.
- Veri Tanımlama Dili kullanılarak oluşturulan veri tabanı tanımları, VTYS'nin Veri Tanımlama Dili Derleyicisi olarak adlandırılabilen bir bileşeni tarafından çözümlenir, varsa eksik ve yanlışları bulunarak kullanıcıya iletilir.
- Yanlışlardan arınmış veri tabanı tanımları VTYS tarafından uygun yapılara dönüştürülerek saklanır.

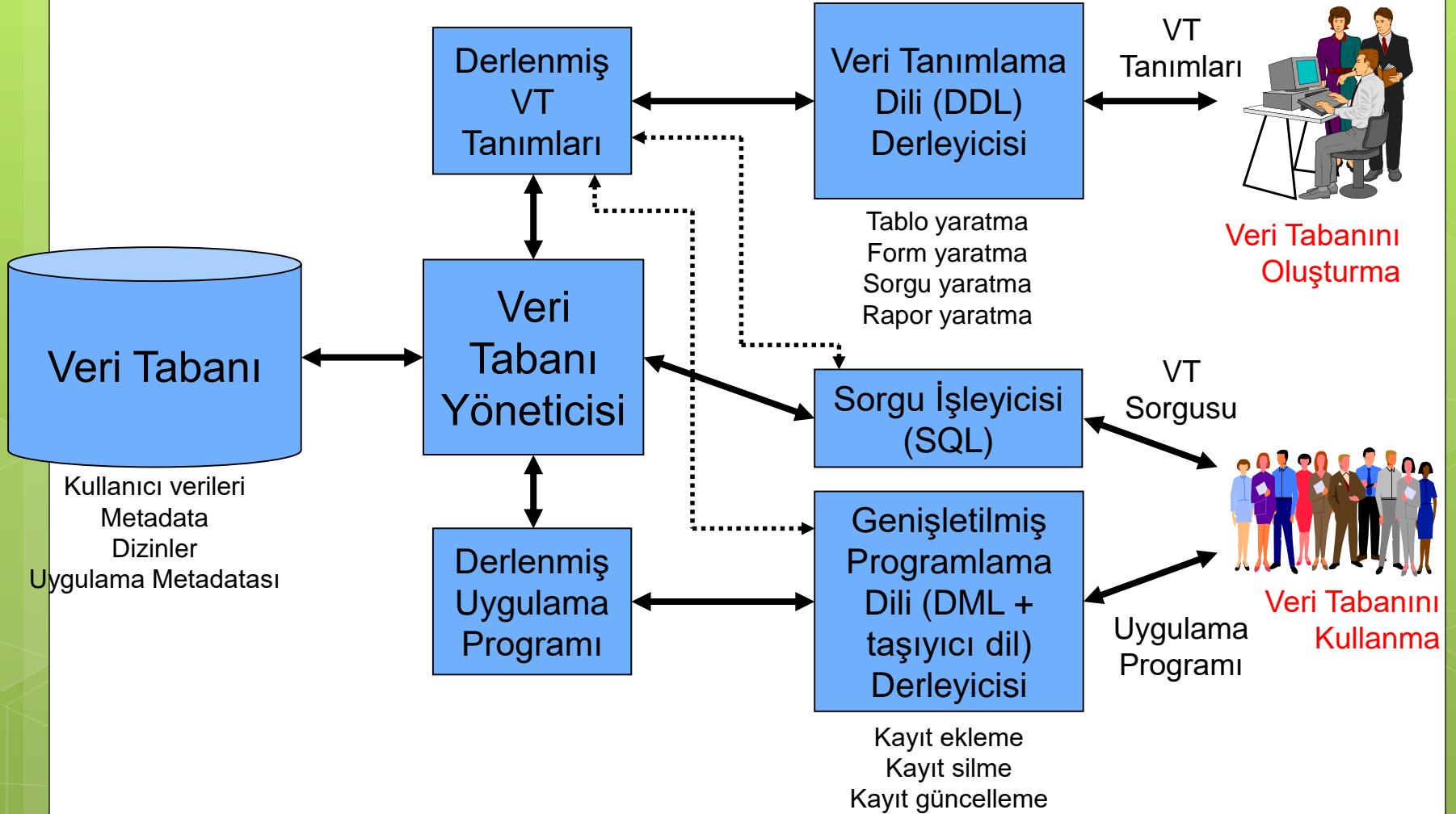
Veri Tanımlama Dili (VTD)

- Veritabanı tanımlarının VTYS tarafından derlenerek saklanması veritabanı yaklaşımının temel özelliklerinden biridir.
- Bu süreç sayesinde, veri tabanı tanımlarının yetkili kişiler tarafından bir kez yapılması, tanımların kalıcılığının sağlanması ve kullanıcıların bu tanımları kullanmaları ve bu tanımlara uygun işlem yapmaları sağlanmış olur.
- Programlama dili kullanılarak gerçekleştirilen dosya tabanlı uygulamalarda, bilgisayar belleklerinde yalnız veri saklanır. Uygulama programlarında, veri üzerinde yapılacak işlemler yanında verinin tanımı da yer alır.

Veri Tanımlama Dili (VTD)

- VTD ile yapılan tanımlarda veri tabanı tanımları içinde yer alan en önemli tanımlar şunlardır:
 1. Mantıksal düzeyde yapılan düzenlemeler oluşturulan yapılar ve her yapıda hangi verilerin yer aldığı.
 2. Her verinin türü, uzunluğu, varsa varsayılan değeri ve diğer özellikleri.
 3. Veriler arası ilişkiler ve her türlü kısıtlamalar.
 4. Fiziksel veri yapıları ile ilgili tercihler ve parametreler.
 5. Kullanıcı tanımları ve kullanıcıların hangi veriler üzerinde hangi işlemleri yapmaya yetkili olduklarına ilişkin tanımlar.

VTYS'nin Temel Bileşenleri (2)



Sorgu Dili ve Sorgu İşleyicisi

- Veri tabanı uygulamaları için kullanılan en yaygın araç sorgu dilidir.
- Kullanıcı, sorgu dili ile, gerçekleştirmek istediği işlemi yalın bir biçimde ifade eder.
- Kullanıcının oluşturduğu sorguda, neyin yapılmasının istendiği yer alır; bunun nasıl yapılacağı ise yer almaz.

Yapısal Sorgu Dili (SQL)

- IBM, deneysel bir ilişkisel veritabanı yönetim sistemi geliştirmek amacıyla System/R adlı bir proje başlatır.
- Bu sistem için SEQUEL (Structured English Query Language) adında bir sorgu dili geliştirilmeye başlanmıştır.
- 1979' da tamamlanan bu proje sırasında geliştirilen dilin adı SQL (Structured Query Language) olarak değiştirilmiştir.

ORACLE

- System/R projesini izleyen bir grup mühendis, ilişkisel veritabanındaki potansiyeli görmüş ve Relational Software, Inc. adında bir şirket kurmuşlardır (1979)
- Oracle adını verdikleri ilk ticari ilişkisel veri tabanı yönetim sistemini geliştirmişler ve SQL'i bu sistemde sorgu dili olarak kullanmışlardır.

SQL Örneği

- İngilizce diline çok benzeyen SQL sorgu dilinde oluşturulmuş bir sorgu aşağıda yer almaktadır.

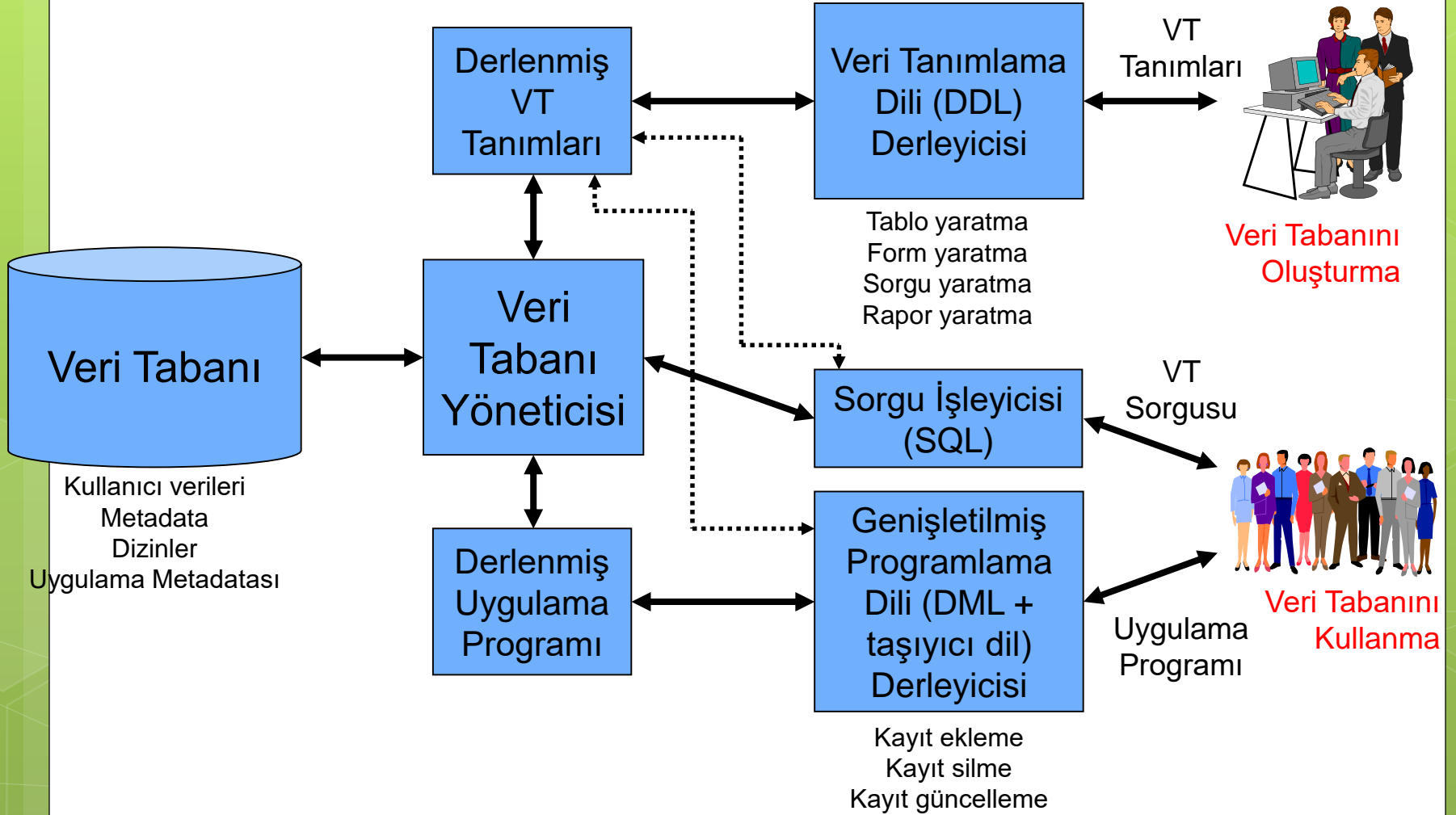
```
SELECT ADI, SOYADI, ADRESİ FROM  
PERSONEL WHERE BÖLÜMNO = 17 AND  
GÖREVİ = 'Sekreter';
```

- Yukarıdaki sorgu ile “17 numaralı bölümde çalışan sekreterlerin ad, soyad ve adreslerinin” bulunmak istendiği kolayca anlaşılmaktadır.

Sorgu işleyicisi

- VTYS'nin, sorguların işlenmesi ile ilgili görevleri gerçekleştiren bileşenine Sorgu İşleyici (Query Processor) adı verilir.
- Sorgu işleyicinin görevleri:
 - Sorgunun sözdizimsel ve anlamsal özümlemesini yapmak.
 - Kullanıcının verilen işlemi yapmaya yetkili olup olmadığını denetlemek.
 - Sorguyu işletmek için kullanılacak algoritmaları (işletim senaryolarını) belirlemek ve “Query Optimizer” alt bileşeni yardımıyla en iyisini seçmek.
 - Sorgunun işletimini gerçekleştirdikten sonra yanıtını oluşturup kullanıcıya iletmek.

VTYS'nin Temel Bileşenleri (3)



Veri İşleme Dili (DML)

- Veri tabanı üzerinde, veriyi değiştirme, silme ve güncelleme gibi sorgularla ifade edilemeyecek ya da sorgularla ifade edilmesi uygun olmayan işlemler de gerçekleştirilir.
- Bu işlemler için Veri İşleme Dili (DML: Data Manipulation Language) olarak adlandırılan bir dil kullanılır.

Genişletilmiş Programlama Dili

- Veri tabanı üzerindeki uygulamaları gerçekleştirmek için genişletilmiş bir dil oluşturulur.
- Bu dilde hem veri tabanı işlemlerini gerçekleştiren DML komutları, hem de diğer işlemleri gerçekleştiren PHP, C# ve Visual Basic gibi genel amaçlı programlama dillerinden bir ya da birkaçı ile yazılmış taşıyıcı dil komutları yer alır.
- Oluşturulan genişletilmiş dil (DML + taşıyıcı dil) ile hazırlanan uygulama programları, VTYS'nin bileşeni olan genişletilmiş derleyicilerle derlenerek saklanır ve gerektiğinde çalıştırılarak kullanılır.

Veritabanları

Önceki sunumlarda veritabanlarının avantajlarını ve çalışma modellerini gördük. Şimdi de veritabanlarının yapısını ve avantajlarını daha rahat anlayabilmek için **örnekler** üzerinden ilerleyelim.



Veritabanları

Veritabanını kullanmaksızın bir şeylerin kaydını tutmak için listeler kullanılır.

- Alışveriş listesi,
- Yapılacak işler listesi,
- Ödenen faturaların listesi

Veritabanları

Veritabanını kullanmaksızın bir şeylerin kaydını tutmak için listeler kullanılır.

- Alışveriş listesi,
- Yapılacak işler listesi,
- Ödenen faturaların listesi

Aslına bakarsanız günümüz veritabanlarının yapısı da bu listeleme mantığına dayanmaktadır.

Liste yapısını bir tablo üzerinde incelediğimizde bu yapısal benzerliği daha rahat göreceksinizdir.

Veritabanları

Örnek Liste

Müsteri adı	Mesleği	Firma	Adres	Fiyat
Ahmet	mimar	A	Çanakkale Cad. 43/5	1000 YTL
Sema	öğretmen	B	Kayabaşı mah. A Blok 8/4	250 YTL
Serdar	mühendis	C	Gazi Mah 6/7	350 YTL
Zerrin	emekli	B	Kayabaşı mah. A Blok 8/4	700 YTL
Mehmet	Öğretim elemanı	B	Kayabaşı mah. A Blok 8/4	1200 YTL
Defne	Doktor	D	Gazi Mah 6/7	100 YTL
Elif	avukat	A	Çanakkale Cad. 43/5	150 YTL

Veritabanları

- Listelerle ilgili problemler;
 - Örneğin B firmasının adresi deęiřti
 - 3 satırdaki adres bilgisi de deęiřmeli
 - Eksik kalırsa bilgi tutarsızlıęı oluşur
- Hem **zaman kaybına** hem de **hataya** neden olabilir.



Veritabanları

- Listelerle ilgili problemler;
 - Örneğin A firması artık sizin şirketinizle çalışmıyor,
 - Listedeki firmayla ilgili kayıtları silerseniz ürün satışı yapılan müşteri bilgisini ve firma adresi gibi bilgileri kaybedersiniz.



Veritabanları

- Paylaşılan verilerle ilgili problemler;
 - Bu bilgilerin tamamının tüm departmanlarla paylaşılması ise farklı nedenlerden dolayı sakıncalıdır
 - Güvenlik
 - Müşteri gizliliği Vb.



Veritabanları

- Listeler oluşturmanın en büyük sakıncası farklı temalarla ilgili bilgileri bir arada bulundurmasıyla ilgilidir.

Müsteri adı	Mesleği	Firma	Adres	Fiyat
Ahmet	mimar	A	Çanakkale Cad. 43/5	1000 YTL
Sema	öğretmen	B	Kayabaşı mah. A Blok 8/4	250 YTL
Serdar	mühendis	C	Gazi Mah 6/7	350 YTL
Zerrin	emekli	B	Kayabaşı mah. A Blok 8/4	700 YTL
Mehmet	Öğretim elemanı	B	Kayabaşı mah. A Blok 8/4	1200 YTL
Defne	Doktor	D	Gazi Mah 6/7	100 YTL
Elif	avukat	A	Çanakkale Cad. 43/5	150 YTL

Veritabanları

- Benzer şekilde farklı temalardaki bilgilerin farklı tablolara yerleştirilmesi süreci **normalizasyon** olarak adlandırılır.
- Biraz önceki liste için;
 - Müşteriler
 - Çalışılan firmalar
 - Satış bilgileri gibi

Veritabanları

- Müşteriler

Müşteri adı	Mesleği
Ahmet	mimar
Sema	öğretmen
Serdar	mühendis
Zerrin	emekli
Mehmet	Öğretim elemanı
Defne	Doktor
Elif	avukat

Veritabanları

- FirmaBilgileri

Firma	Adres
A	Çanakkale Cad. 43/5
B	Kayabaşı mah. A Blok 8/4
C	Gazi Mah 6/7
D	Gazi Mah 6/7

Veritabanları

- o SatisBilgileri

Fiyat
1000 YTL
250 YTL
350 YTL
700 YTL
1200 YTL
100 YTL
150 YTL

Veritabanları

- Normalizasyon yapmak veritabanlarında çok büyük öneme sahiptir.
- Farklı temalar farklı tablolara yerleştirildiğinde,
 - Değişen bilgiler,
 - Silinen bilgiler ve
 - Paylaşılan bilgilerle ilgili problemlerin çoğu ortadan kalkar.
- Farklı temalar farklı tablolarda yer aldığına;
 - Hangi müşteri hangi firmadan ürünü almış gibi soruları cevaplamak için ilişkilerin kurulması gereklidir.

Veritabanları

İlişkiler

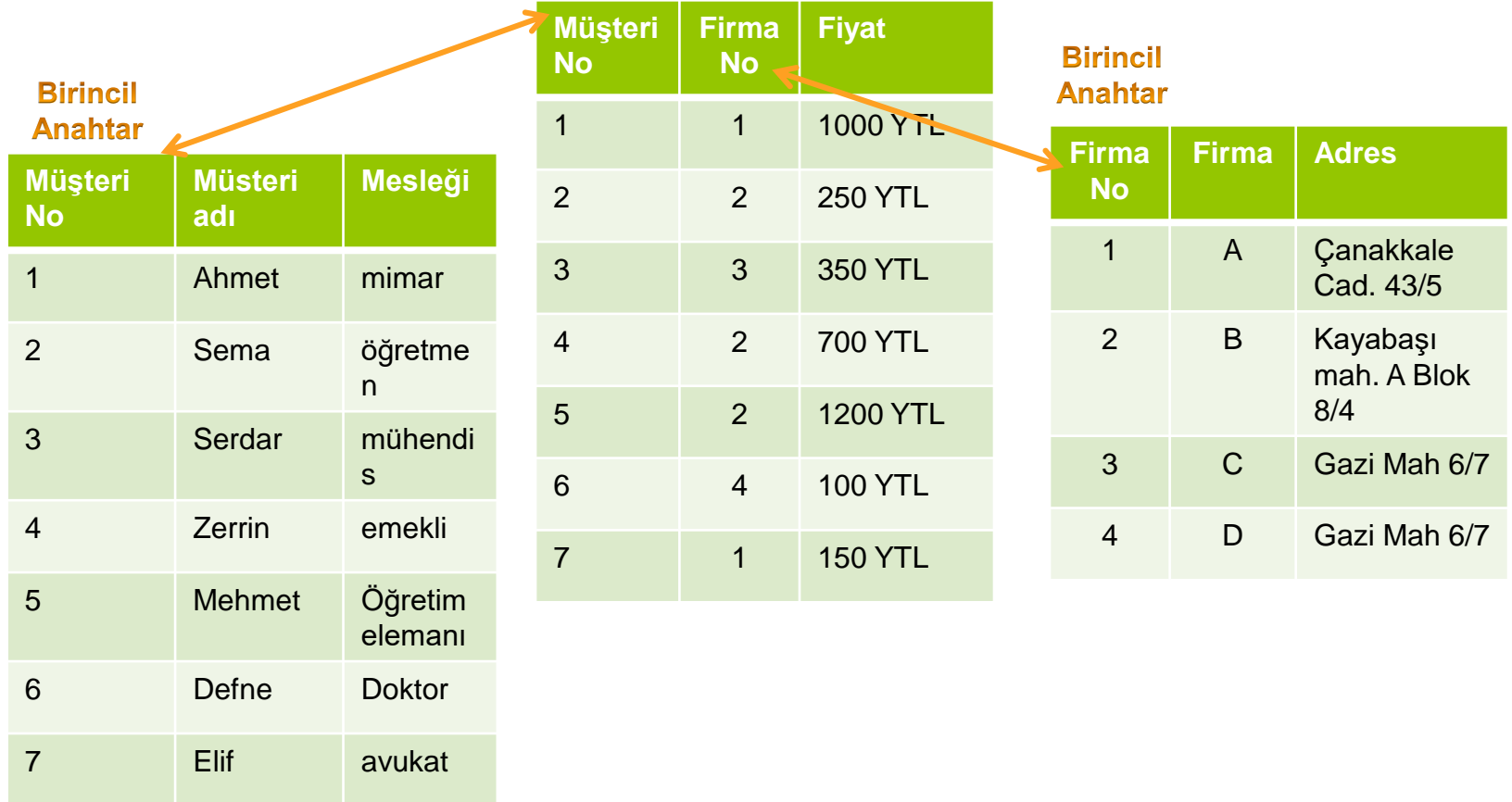
Müşteri No	Müşteri adı	Mesleği
1	Ahmet	mimar
2	Sema	öğretmen
3	Serdar	mühendis
4	Zerrin	emekli
5	Mehmet	Öğretim elemanı
6	Defne	Doktor
7	Elif	avukat

Müşteri No	Firma No	Fiyat
1	1	1000 YTL
2	2	250 YTL
3	3	350 YTL
4	2	700 YTL
5	2	1200 YTL
6	4	100 YTL
7	1	150 YTL

Firma No	Firma	Adres
1	A	Çanakkale Cad. 43/5
2	B	Kayabaşı mah. A Blok 8/4
3	C	Gazi Mah 6/7
4	D	Gazi Mah 6/7

Veritabanları

İlişkiler



Veritabanları

Tabloları birleştirmek

Peki baştaki bütüncül listeyi elde etmek için ne yapılır?



SQL dilinden faydalanılır.

Veri Modelleri

- Veri modeli, verileri mantıksal düzeyde düzenlemek için kullanılan yapılar, kavramlar ve işlemler topluluğudur.
- Her VTYS belirli bir veri modeli kullanır.
- Veri tabanını tasarlayan kişi, veri modelinin yapılarını ve kavramlarını kullanarak mantıksal düzeydeki düzenlemeleri oluşturur. Daha sonra o veri modelini kullanan bir VTYS üzerinde bu düzenlemelere göre veri tabanı yaratılır.
- Günümüzde en çok kullanılan veri modeli “ilişkisel veri modeli” dir.

KAYNAKLAR

- Yrd. Doç. Dr. Altan MESUT

<http://altanmesut.trakya.edu.tr/vtys1/>

- Öğr. Gör. Dr. Sibel SOMYÜREK

http://sibelsomyurek.com/veritabani/ders_notlari.html