

Veritabanı Yönetim Sistemleri

Giriş, Temel Kavramlar

Öğr. Gör. Dr. Yasemin Topuz

Yıldız Teknik Üniversitesi



Ders İşleyışı

- Teorik Ders: Cuma 14:00 – 16:50
- Pratik Ders: Cuma 17:00 – 18:50 / Asistan arkadaşlar ilan edecek (5 asistan)
- Duyurular, Bilgilendirmeler ve Ders Notları
 - Online Sistem Ders Duvarı, OBS
 - Google Classroom: [2yyr3qrb](#)

Ders İşleyişi

- Database system concepts. 7th Editon. Silberschatz, A., Korth, H. F., & Sudarshan, S. New York: McGraw-hill.
- Veri tabanı Sistemleri, Ünal Yarımağan, Akademi Kitapevi.
- Fundamentals of database systems. 7th Editon. Elmasri, R., & Navathe, S. B.

Etkinlikler	Sayı	Katkı Payı
Laboratuvar	12	15
Ödev	0	0
Projeler	1	10
Ara Sınavlar	2	35
Final	1	40

#	Hafta	Konular	Kaynak
1	3 Ekim 2025	GİRİŞ VE TEMELLER: Dosya sistemi ve Veritabanı Kavramları; VT Sistem Genel Yapısı, DBMS bileşenleri - türleri	Silberschatz Ch1
2	10 Ekim 2025	İLİŞKİSEL MODEL Varlık, ilişki, anahtarlar (PK, FK, CHECK), İlişkisel Cebir (Relational Algebra) ve Kısıtlar	Silberschatz Ch2
3	17 Ekim 2025	SQL'E GİRİŞ DDL/DML; temel SELECT/WHERE/ORDER BY/LIMIT; basit JOIN.	Silberschatz Ch3
4	24 Ekim 2025	ORTA DÜZEY SQL Çoklu JOIN, Aggregation, GROUP BY/HAVING; alt sorgular (IN/EXISTS).	Silberschatz Ch4
5	31 Ekim 2025	İLERİ SQL CTE (WITH), pencere fonksiyonları, VIEW/materialized view, performans ipuçları.	Silberschatz Ch5
6	7 Kasım 2025	VERİTABANI PROGRAMLAMA Embedded SQL, JDBC, Saklı Yordamlar (Stored Procedures),	Silberschatz Ch5
7	14 Kasım 2025	SQL İLE VERİ BÜTÜNLÜĞÜ VE GÜVENLİK Kısıtlamalar (Constraints), Tetikleyiciler (Triggers), SQL Injection	Silberschatz Ch5
8	17-22 Kasım 2025	VİZE SINAVI - I	
9	28 Kasım 2025	KAVRAMSAL TASARIM ER ile Veritabanı Kavramsal Tasarımı	Silberschatz Ch6
10	5 Aralık 2025	KAVRAMSAL TASARIM EER ve UML ile Veritabanı Kavramsal Tasarımı (UML sınırlı)	Silberschatz Ch6
11	12 Aralık 2025	NORMALİZASYON Anomaliler; 1NF–3NF–BCNF; işlevsel bağımlılıklar.	Silberschatz Ch7
12	19 Aralık 2025	KARMAŞIK VERİ YAPILARI Yarı Yapısal Veri Modelleri, XML, JSON-Relational, XQuery	Silberschatz Ch8
13	26 Aralık 2025	VİZE SINAVI – II	
14	2 Ocak 2026	FİZİKSEL TASARIM Depolama düzenleri; B+ ağaç ve hash; kümelenmiş/kümelenmemiş; kaplama indeks.	Silberschatz Ch14
15	9 Ocak 2026	NOSQL AİLESİ, DAĞITIK & VEKTÖR VT TEMELLERİ MongoDB/Redis/Cassandra; Replikasyon, sharding, CAP; VT & RAG temelleri.	Silberschatz Ch20
16	12-22 Ocak 2026	FİNAL SINAVI	

Dersin Kazanımları

- Veritabanı yönetim sistemlerinin temel kavramlarını anlama
- İlişkisel sorgu dili SQL (Structured Query Language)'i öğrenme ve uygulama
- Bir ilişkisel veritabanı yönetim sistemi kullanarak;
 - Veritabanı tasarımı (Diyagramlar, çizimler)
 - Veritabanı gerçekleştirmesi (indekslemeler, veriye hızlı erişim yöntemleri)
 - Veritabanı üzerinde sorgulama yapılması becerisi kazanma
- Takım çalışması ve iletişim becerilerini geliştirme

Neler konuşacağız?

- Veri, Bilgi ve Veritabanı (VT) Nedir?
- Veritabanı Yönetim Sistemi (VTYS) Nedir?
- VTYS Olanakları ve Dosya İşleme Sisteminden Farkları
- Veri Modeli Nedir?
- Örnek Bir Veritabanı: Üniversite
- VTYS Uygulamalarının Bileşenleri
- VTYS Ana İşlevsel Modülleri
- VTYS Dilleri
- VTYS Türleri
- VTYS 3-Şema Mimarisi ve Veri Bağımsızlığı

Veri, Bilgi ve Veritabanı Kavramları

- **Veri (Datum):** Gerçek dünyadaki olguları temsil eden ham semboller, ölçümler ya da kayıtlardır.
 - **Termometre:** 36.7°C;
 - **Hava durumu:** İstanbul, 18°C, %70 nem;
 - **Navigasyon:** 12 km, ort. hız 20 km/s
 - **Toplu taşıma kartı:** Bakiye 28,60 TL
- **Bilgi:** Verinin anlaşılmış, yorumlanmış hâli (bağlam eklenir, karar verdirir).
 - **Termometre:** Ateşim normal
 - **Hava durumu:** Üşütmez, hafif mont yeter.
 - **Navigasyon:** Yoğun trafik; 35 dk'da varırım
 - **Toplu taşıma kartı:** Bir biniş bir de aktarma hakkı var; sonra yükleme yapmalıyım.

Veri, Bilgi ve Veritabanı Kavramları

- **Veritabanı:** Verinin düzenli biçimde saklandığı ve Veritabanı Yönetim Sistemi (VTYS) ile yönetildiği yapıdır.
 - Online alışveriş
 - Gün bazında toplam kaç alışveriş yapıldığı, satılan ürünlerin toplam bedeli, en çok satılan ürün, vb.
 - Bu veriler neden bir txt dosyasına yazılmıyor?
 - Bu veriler birbirleriyle ne kadar ilişkili olursa bu verileri indekslemek ve bu verilere erişmek o kadar kolaylaşır.

Veri, Bilgi ve Veritabanı Kavramları

- **Veritabanı:** Verinin düzenli biçimde saklandığı ve Veritabanı Yönetim Sistemi (VTYS) ile yönetildiği yapıdır.
 - Birden çok uygulama tarafından kullanılabilen,
 - Gereksiz yinelemelerden arınmış,
 - Düzenli bir şekilde saklanan,
 - Birbirleriyle ilişkili (uyumlu olarak),
 - Sürekli ancak statik olmayan,
 - Belirli bir amaç için bir araya getirilmiş,
- VERİ TOPLULUĞU** (küçük bir dünya)'dur.

Veritabanı Uygulamaları



Banking



Library
Systems



Payroll
Systems



Healthcare



Social Media



Online
Shopping

Database Applications



Education



Human
Resources



Research



Inventory MS

Veritabanı

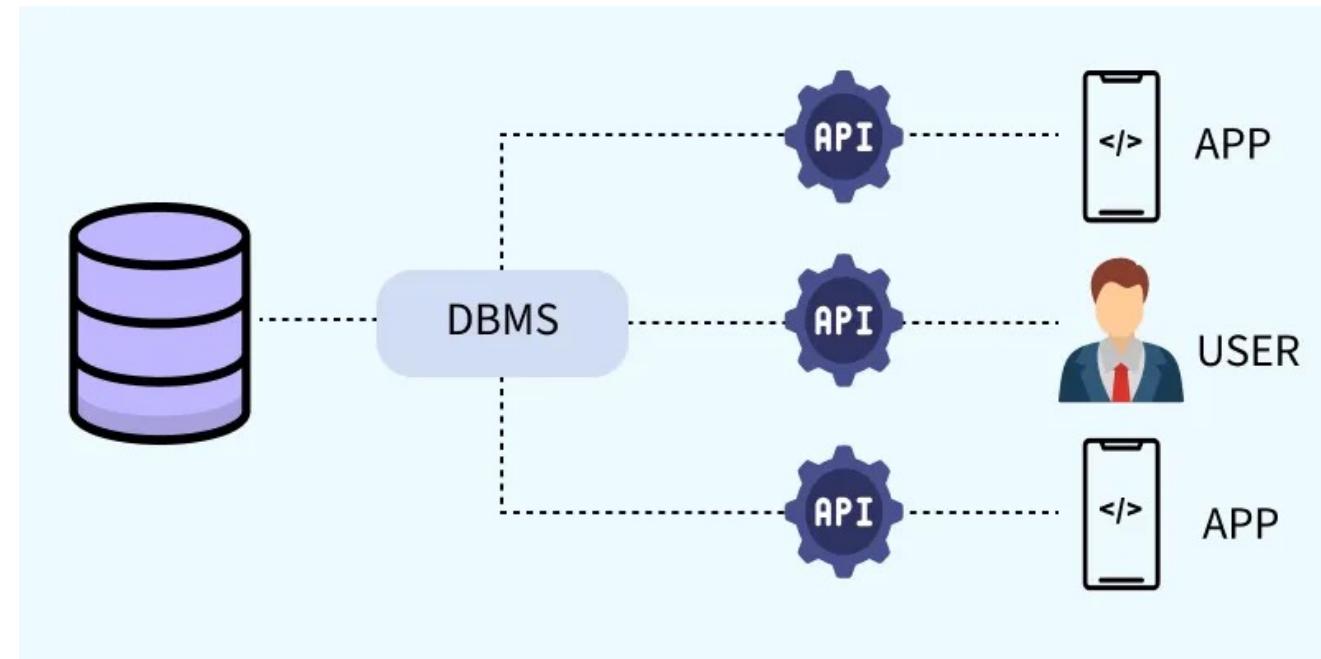
- **Veritabanı:** Verinin düzenli biçimde saklandığı ve Veritabanı Yönetim Sistemi (VTYS) ile yönetildiği yapıdır.
 - VT gerçek zamanlı çalışmalıdır.
 - VT büyüklüğü için bir kısıtlama yoktur.
 - Örnek: Facebook (2019)
 - $300 \text{ petaB} (\sim 3 \times 10^{17} \text{ B})$
 - Günlük 4 petaB
 - 2.45 milyar aktif kullanıcı
 - 30k sunucu (2010)

Geleneksel Dosya Tabanlı Sistemlerdeki Sorunlar

- Modern VTYS ortaya çıkmadan önce, veriler sabit disklerdeki temel dosya sistemleri kullanılarak yönetiliyordu.
- Bu yaklaşım, kullanıcıların dosyaları gerektiğinde depolamasına, erişmesine ve güncellemesine olanak sağlasa da, beraberinde birçok zorluk barındırır.
 - **Veri Yedekliliği:** Dosyalar arasında yinelenen girişler
 - **Tutarsızlık:** Çelişkili veya güncellliğini yitirmiş bilgiler
 - **Zor Erişim:** Manuel dosya araması gerekli
 - **Zayıf Güvenlik:** Veri erişimi üzerinde kontrol yok
 - **Tek Kullanıcı Erişimi:** İşbirliği desteği yok
 - **Yedekleme/Kurtarma Yok:** Veri kaybı genellikle kalıcıdır

VTYS (Database Management System) Nedir?

- VTYS, verileri **yapılandırılmış** bir formatta etkin bir şekilde **yöneten, depolayan ve geri çağırın** bir yazılım sistemidir.
- Kullanıcıların veritabanlarını etkili bir şekilde **oluşturmasına, güncellemesine ve sorgulamasına** olanak tanır.
- Birden fazla kullanıcı ve uygulamada **veri bütünlüğünü, tutarlığını ve güvenliğini** sağlar.
- Merkezi kontrol sayesinde **veri fazlalığını ve tutarsızlığını azaltır**.
- **Eşzamanlı** veri erişimini, işlem yönetimini ve **otomatik yedeklemeleri** destekler.



VTYS'nin Sağladığı Olanaklar

- Veritabanı (VT) tanımlanması, gerçekleştirilmesi (oluşturulması), kullanımı ve paylaşımı
- Kontrollü veri tekrarı,
- Sorgu işlemede verimli erişim metotları kullanımı,
- Çoklu kullanıcılı hizmet, veri kurtarma ve yedekleme imkanı,
- Farklı kullanıcı arayüzleri imkanı,
- Üst seviyeli karmaşık iş kısıtlamalarının tanımlanması, gerçekleştirilmesi ve sağlanması
- Güvenlik tanımlamaları ve sağlanması.

VT sistemine, gerek işletim sistemi gerek diğer kullanıcılar (uygulama programları gibi) **doğrudan erişemez**; ancak VTYS üzerinden erişilebilir.

Neden VYTS?

- 2025 yılı girişli öğrencileri bulalım.

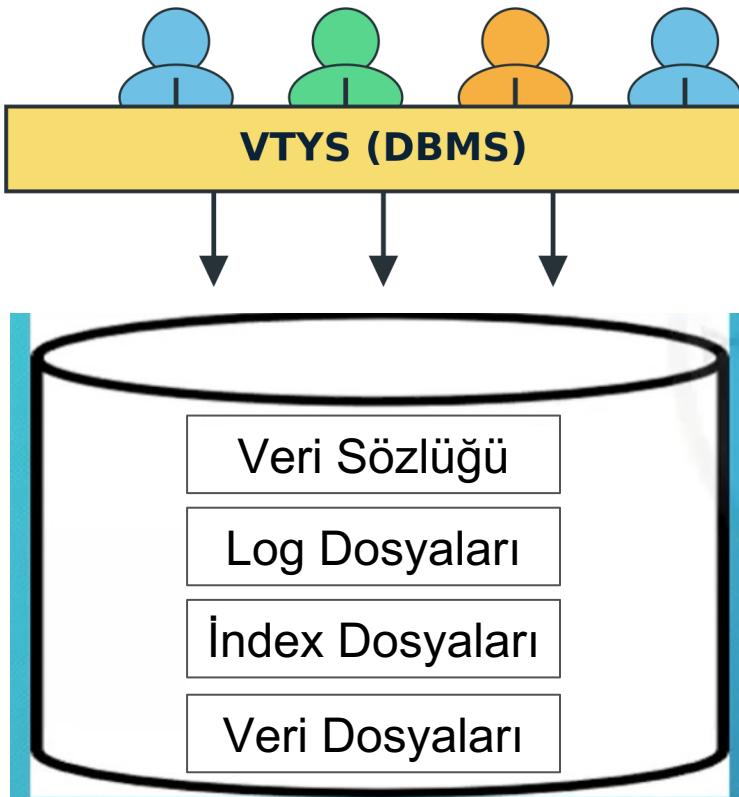
```
public static List<String> getStudents1997() {  
    List<String> result = new ArrayList<String>();  
    FileReader rdr = new FileReader("students.txt");  
    BufferedReader br = new BufferedReader(rdr);  
    String line = br.readLine();  
    while (line != null) {  
        String[] vals = line.split("\t");  
        String gradyear = vals[2];  
        if (gradyear.equals("2025"))  
            result.add(vals[1]);  
        line = br.readLine();  
    }  
    return result;  
}
```

Select Sname from STUDENT where GradYear=2025

Dosya Sistemi vs VTYS – Karşılaştırmalı Özeti

Kriter	Dosya Sistemi	VTYS
Yapı	Dosyaları düzenlemeye yaklaşımı	Veriyi yöneten yazılım; şema/index/katalog
Veri Yedekliliği & Tutarlılık	Gereksiz kopyalar oluşur; tutarlılık düşük	Normalizasyon ve kısıtlarla yüksek tutarlılık (ACID)
Sorgu İşleme & Anlam	Yerleşik sorgu motoru yok	SQL + sorgu eniyileyici (optimizer)
Üst Veri (Metadata)	Sınırlı/dağıtık	Katalogda şema, görünümler, kısıtlar, yetkiler
Yedekleme & Kurtarma	Elle/araçlara bağımlı	Günlükleme, snapshot, point-in-time recovery
Eşzamanlılık & Paylaşım	Aynı anda az kullanıcı; paylaşım zayıf	Kilitleme/MVCC; transaction, rollback
Güvenlik	Dosya izinlerine dayanır	Rol/yetki; satır-sütun düzeyi denetim, denetim izi
Bütünlük Kısıtları	Uygulama koduya, zor	Birincil/yabancı anahtar, check, tetikleyici
Veri Bağımsızlığı & Görünümler	Yok denecek kadar az	Mantıksal-fiziksel bağımsızlık; multiple views
Kayıt Saklama & Erişim	İndeks/tampon yönetimi kullanıcında	İndeksler ve buffer havuzu sistemde
Karmaşıklık & Maliyet	Basit/ucuz; büyündükçe kırılgan	Kurulum maliyeti yüksek; güvenilir/ölçeklenir
Kullanım Örneği	Tek kullanıcı, küçük/geçici veri	Çok kullanıcılı, kurallı ve kritik sistemler

VT / VTYS Genel Yapısı



- **Veri Sözlüğü:** Veritabanının tanımı (metadata) ve kurallarıdır: şemalar, tablolar, sütun tipleri, anahtarlar, kısıtlar, görünümler, roller/yetkiler. Kim neye, nasıl erişir ve veri nasıl doğrulanır bilgisini sistem düzeyinde tutar.
- **Log Dosyaları:** Verinin doğruluğu ve kurtarılması amaçlı dosyalardır. Veritabanında yapılan tüm işlemler, hangi öğrenci hangi dersi eklemiş silmiş vb.
- **Index Dosyaları:** Veriye erişimi hızlandıran arama yapılarıdır. Sorgu maliyetini düşürür, özellikle filtreleme/sıralama/join operasyonlarında büyük kazanç sağlar.
- **Veri Dosyaları:** Tabloların gerçek satır/veri içerikleri burada tutulur.

VTYS – Bazı Kavramlar

- **Uygulama programı:** Veri için istek veya sorgu göndererek VT'ye erişim yapan program.
- **Sorgu:** VT'deki verinin alınmasına (retrieve) neden olan işlem.
- **Transaction:** Bazı verilerin VT'den okunmasına ve bazı verilerin VT'ye yazılmasına neden olan hareket, iş.

Veri Modeli

- Gerçek dünya verilerini kavramsal ve mantıksal seviyede düzenlemek için kullanılan yapı ve kavamlar bütünü olarak tanımlanır.
- Bu sayede veriler arası ilişkiler ve veritabanının uyacağı kısıtlamalar tanımlanabilir.
- Görselleştirmek için; E-R modeli, (E)ER, UML,
- İlişkisel model (RM), Object model, Object-Relational Model, XML
- Genel VT tasarımı:
 - Kavramsal Tasarım
 - Mantıksal Düzenleme, Varlık ve Bağıntıların Belirlenmesi
 - Veri Tipleri, Değer Aralığı, Uzunluk belirlenmesi
 - Veri bütünlüğü kısıtlamalarının belirlenmesi
 - Fiziksel tasarım tercihleri (index)
 - Kullanıcıların belirlenmesi ve güvenlik ayarları

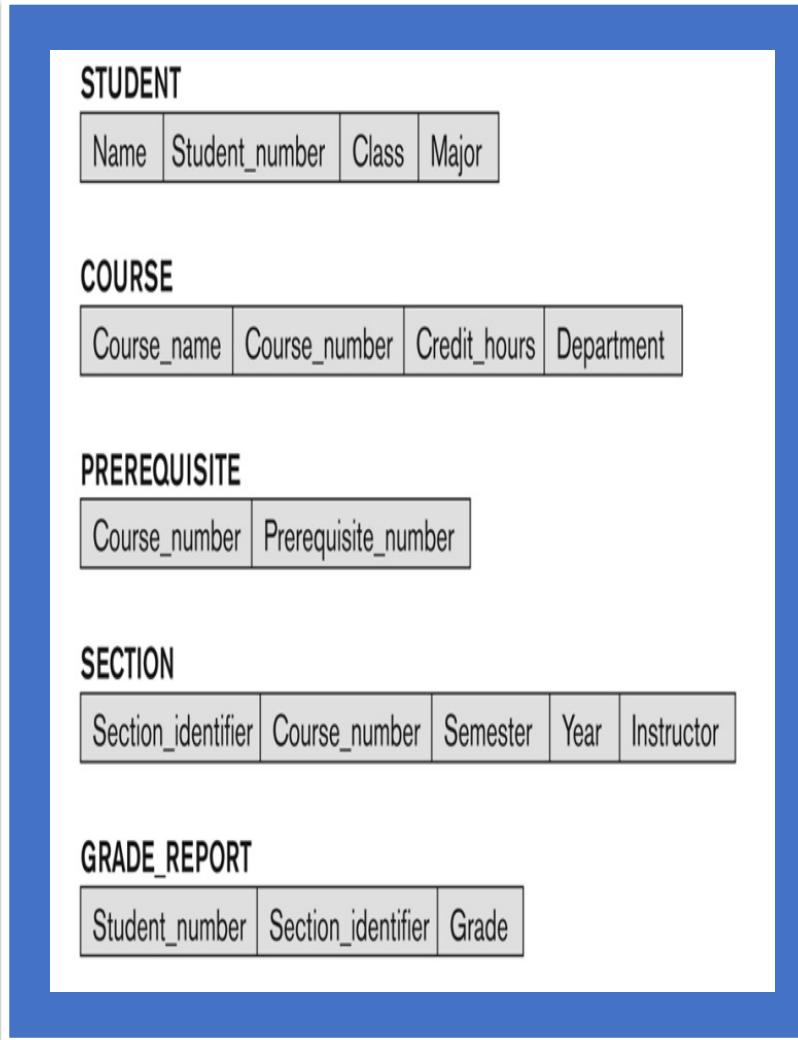
Veri Modellerinin Kategorilendirilmesi

- **Kavramsal Modelleme:**
 - Kullanıcıların veriye erişim konsepti tanımlanır.
- **Mantıksal Modelleme**
 - Veriler ve aralarındaki ilişkilerin modellenmesidir.
- **Fiziksel Modelleme**
 - Verilerin boyutuyla beraber değerlendirilerek modellenmesidir.

Şemalar (Schemas) vs Durum (State)

- **Veritabanı Şemaları (Database Schema)**
 - Veritabanın bir tanımıdır.
 - Veritabanı yapısı, veri tipleri ve veriler arasındaki kısıtları gösterir.
- **Şema Diyagram (Schema Diagram)**
 - Veritabanı şemalarının açıklayıcı ve basitleştirilmiş bir görüntüsüdür.
- **Şema Elemanları (Schema Construct)**
 - Şema içerisindeki nesnelerdir. Örnek Öğrenci, Ders, Sınıf vb.
- **Veritabanı Durumu (Database State)**
 - Belirli bir zamanda veritabanı içerisinde depolanan veriler bütünüdür.
 - Farklı zamanlarda farklı neticeler üretmesi beklenir.

Şemalar (Schemas) vs Durum (State)



COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

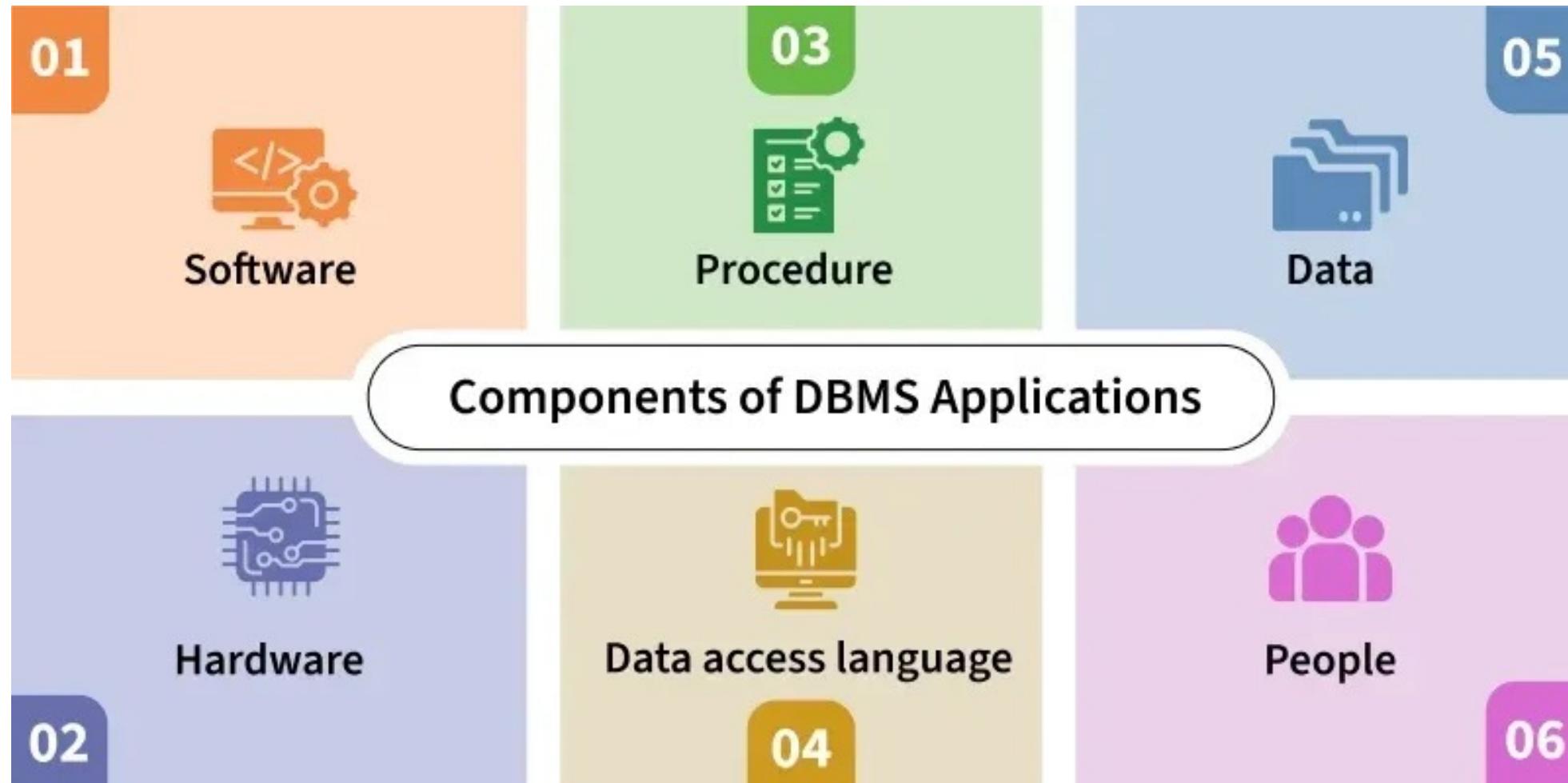
Şemalar (Schemas) vs Durum (State)

- Üniversite ortamının bir parçası (Mini-dünya)
- Bazı mini-dünya varlıklar (Varlık-bağıntı modeli ile)
 - ÖĞRENCİLER
 - DERSLER
 - BÖLÜMLER
 - HOCALAR
- Bazı mini-dünya ilişkileri (Varlık-bağıntı modeli ile)
 - ÖĞRENCİLER DERSLERİ *alır.*
 - Bazı DERSLERİN önkoşul DERSLERİ *vardır.*
 - HOCALAR DERSLERİ *verir.*
 - BÖLÜMLER DERSLERİ *açar.*
 - ÖĞRENCİLER BÖLÜMLERE *kayıtlıdır.*

Şemalar (Schemas) vs Durum (State)

- **Sorgu örnekleri:**
 - Numarası girilen öğrencinin transkriptini getir.
 - Veritabanı Yönetimi dersini 2025 Güz döneminde alan öğrencileri getir.
 - Veritabanı Yönetimi dersinin ön koşullarını listele.
- **Güncelleme örnekleri:**
 - Ali'nin not ortalamasını 2.97'ye güncelle
 - Veritabanı Yönetimi dersinin kontenjanını 250 yap.
 - Ali'nin Veritabanı Yönetimi dersi notunu AA olarak işle.
 - Veritabanı Yönetimi dersini veren hoca listesine Elçin'i ekle.

VTYS Uygulamalarının Bileşenleri



<https://www.geeksforgeeks.org/dbms>

VTYS Uygulamalarının Bileşenleri

1. Donanım

- Sunucular, diskler, giriş-çıkış aygıtları (klavye, monitör, yazıcı) gibi fiziksel aygıtlar.
- Verileri depolar ve işler; gerçek dünya girdileri ile dijital sistemler arasında arayüz oluşturur.
- Örnekler: Kişisel bilgisayar sabit diski, RAM, VTYS işlemleri için kullanılan ağ aygıtları.

2. Yazılım

- MySQL, Oracle, PostgreSQL gibi VTYS yazılımları.
- Veritabanı motoru, işletim sistemi, ağ yazılımı ve uygulama araçlarını içerir.
- Veritabanı erişim dillerini operasyonlara çevirir.

VTYS Uygulamalarının Bileşenleri

3. Veri

- Yapılandırılmış veya yapılandırılmamış formatlarda saklanan ham bilgiler.
- Operasyonel Veriler: Gerçek kullanıcı verileri (örneğin, isim, yaşı).
- Meta Veri: Veriler hakkındaki nitelikler (örneğin, depolama süresi, boyut, veri türü).
- VTYS'nin varoluşunun temel nedeni, verileri verimli bir şekilde yönetmek ve depolamaktır.

4. Prosedürler

- VTYS'yi etkili bir şekilde kullanmak için talimatlar ve kurallar.
- Kurulum, giriş/çıkış, veri doğrulama, yedekleme, erişim kontrolü ve rapor oluşturma konularını kapsar.
- Sistemin tutarlı ve güvenli bir şekilde kullanılmasını sağlamaya yardımcı olur.

VTYS Uygulamalarının Bileşenleri

5. Veritabanı Erişim Dili

- Veritabanıyla etkileşimde bulunmak (veri oluşturmak, okumak, güncellemek, silmek) için kullanılır.
- Örnekler: SQL, MyAccess, Oracle PL/SQL.
- DDL (Veri Tanımlama Dili) – CREATE, ALTER,DROP
- DML (Veri İşleme Dili) – INSERT, UPDATE,DELETE

6. İnsanlar

- VTYS ile farklı düzeylerde etkileşimde bulunan kullanıcılar:
- Veritabanı Yöneticileri (DBA) – Güvenliği, performansı ve kullanıcı erişimini yönetin.
- Geliştiriciler – Veritabanını kullanarak uygulamalar oluştururlar.
- Son Kullanıcılar – Veritabanına erişmek için uygulamaları kullanın (örneğin, öğrenciler, çalışanlar).

VT Kullanıcıları

VT Sorumlusu (Admin, DBA)

- VT tasarımından işletimine kadar her şeyden sorumlu kişi(ler)
- VT erişim yetkilerini tanımlama ve kullanımını kontrol
- Sistem için gerekli yazılım ve donanım desteğini belirler.
- Güvenlik için yedekleme ve kurtarma işlemleri,
- VT kullanımını ve başarısını takip etmek, gerekli değişiklikleri yapmak.

Tasarımcı (Designer)

- Verinin her aşamada modellenmesi(yapısı, içeriği ve kısıtlamaları ile).
- Gerçekleme öncesi aşamalardan sorumludur.
- VT kullanıcıları ile haberleşir gereksinim analizi yapar.
- Veritabanı üzerinde fonksiyon ve işlemleri tanımlar.
- Genel olarak bütün kullanıcılar ile yakın temas.
- Hangi veri saklanacak, şema tanımları
- Veriler için uygun yapıları belirleme.

VT Kullanıcıları

Sistem Çözümleyici (System analysts):

- Son kullanıcıların gereksinimlerini belirler.

VT uygulama yazılımcısı (Application Programmer)

- VT erişimi yapacak uygulamaları tasarlama ve gerçekleştirmeye

Son kullanıcılar (End user)

- VT'deki bilgilere sorular ve raporlar aracılığıyla erişirler.
- Veritabanı içeriğini güncellerler.
- Uygulama programı kullanıcıları
- SQL dili kullanıcıları
- Rastgele kullanıcılar

VTYS Dilleri

Type of DBMS Language

Data Definition Language

- Create
- Drop
- Truncate
- Rename

Data Manipulation Language

- Select
- Insert
- Delete
- Update

Data Control Language

- Revoke
- Grant

Transaction Control Language

- Rollback
- Commit

VTYS Dilleri

1. Veri Tanımlama Dili (Data Definition Language - DDL): DDL, veritabanı şemaları ve açıklamaları ile verilerin veritabanında nasıl yer alması gerektiğini ele alan Veri Tanımlama Dili'nin kısa adıdır.

- **CREATE:** Bir veritabanı ve onun nesnelerini (tablo, dizin, görünümler, depolama prosedürü, fonksiyon ve tetikleyiciler) oluşturmak için kullanılır.
- **ALTER:** Mevcut veritabanının yapısını değiştirir.
- **DROP:** Nesneleri veritabanından siler.
- **TRUNCATE:** Kayıtlar için ayrılmış tüm boşluklar da dahil olmak üzere bir tablodan tüm kayıtları kaldırır.
- **COMMENT:** Veri sözlüğüne yorum ekler.
- **RENAME:** Bir nesneyi yeniden adlandırır.

VTYS Dilleri

2. Veri İşleme Dili (Data Manipulation Language - DML): DML, veritabanında saklanan verileri düzenlemeye odaklanarak kullanıcıların verileri almasını, eklemesini, güncellemesini ve silmesini sağlar.

- **SELECT:** Bir veritabanından veri alır.
- **INSERT:** Bir tabloya veri ekler.
- **UPDATE:** Bir tablodaki mevcut verileri günceller.
- **DELETE:** Bir veritabanı tablosundan kayıtları siler.
- **MERGE:** UPSERT işlemi (ekleme veya güncelleme)
- **CALL:** PL/SQL veya Java alt programını çağırır.
- **EXPLAIN PLAN:** Veri erişim yolunun yorumlanmasıdır.
- **LOCK TABLE:** Eşzamanlılık Denetimi

VTYS Dilleri

3. Veri Kontrol Dili (Data Control Language - DCL): DCL komutları erişim izinlerini yönetir ve veritabanında belirli eylemleri kimlerin gerçekleştirebileceğini kontrol ederek veri güvenliğini sağlar.

- **GRANT:** Kullanıcıya belirli ayrıcalıklar sağlar (örneğin, SELECT, INSERT).
- **REVOKE:** Bir kullanıcıya daha önce verilen izinleri kaldırır.

4. İşlem Kontrol Dili (Transaction Control Language - TCL): TCL komutları, tutarlılığı, güvenilirliği ve atomikliği (ya hep ya hiç kuralı) korumak için işlemler verileri denetler.

- **ROLLBACK:** Bir işlem sırasında yapılan değişiklikleri geri alır.
- **COMMIT:** Bir işlem sırasında yapılan tüm değişiklikleri kaydeder.
- **SAVEPOINT:** Bir işlem içerisinde daha sonra geri dönülebilecek bir nokta belirler.

VTYS Dilleri

5. Veri Sorgulama Dili (Data Query Language - DQL): DQL , DML'nin bir alt kümeleridir ve özellikle veri almaya odaklanmıştır.

- **SELECT:** Veritabanının yapısını veya içeriğini değiştirmeden veri sorgulamak için kullanılan birincil DQL komutu.

VTYS Türleri

Her biri farklı veri yapılarına, ölçeklenebilirlik gereksinimlerine ve uygulama ihtiyaçlarına göre uyarlanmış çeşitli Veritabanı Yönetim Sistemleri (VTYS) türleri mevcuttur.

En yaygın türler:

1. İlişkisel Veritabanı Yönetim Sistemi (RDBMS)

- Verileri satır ve sütunlardan oluşan tablolara (ilişkilere) düzenler.
- Satırları benzersiz şekilde tanımlamak için birincil anahtarları ve tablolar arasında ilişkiler kurmak için yabancı anahtarları kullanır.
- Sorgular , verimli veri işleme ve alma olanağı sağlayan SQL (Yapılandırılmış Sorgu Dili) ile yazılır .
- Örnekler: MySQL, Oracle, Microsoft SQL Server ve PostgreSQL.

VTYS Türleri

2. NoSQL VTYS

- Büyük ölçekli verileri işlemek ve ilişkisel modellerin kısıtlayıcı olabileceği senaryolarda yüksek performans sağlamak üzere tasarlanmıştır.
- Verileri anahtar-değer çiftleri, belgeler, grafikler veya sütunlar gibi çeşitli ilişkisel olmayan formatlarda depolarlar.
- Bu esnek veri modelleri hızlı ölçeklendirmeye olanak tanır ve yapılandırılmış veya yarı yapılandırılmış veriler için oldukça uygundur.
- Örnekler: MongoDB, Cassandra, DynamoDB ve Redis.

VTYS Türleri

3. Nesne Yönelimli VTYS (OODBMS)

- Nesne yönelimli programlama kavramlarını veritabanı ortamına entegre ederek, verilerin nesne olarak saklanmasına olanak tanır.
- Karmaşık veri türlerini ve ilişkilerini destekler, bu da onu gelişmiş veri modelleme ve gerçek dünya simülasyonları gerektiren uygulamalar için ideal hale getirir.
- Örnekler: ObjectDB, db4o.

VTYS Türleri

4. Hiyerarşik Veritabanı

- Verileri, her kaydın (düğümün) tek bir üst ögesi ve birden fazla alt ögesi olacak şekilde ağaç benzeri bir yapıda düzenler.
- Bu model klasörler ve alt klasörlerden oluşan bir dosya sistemine benzer.
- Organizasyon şemaları veya dosya dizinleri gibi verilerin net bir hiyerarşije sahip olarak depolanmasında etkilidir.
- Sabit yapı sayesinde navigasyon hızlı ve öngörülebilirdir.
- Esneklikten yoksundur ve karmaşık çoktan çoga ilişkileri yeniden yapılandırmak veya yönetmek zordur.
- Örnek: IBM Bilgi Yönetim Sistemi (IMS).

VTYS Türleri

5. Ağ Veritabanı

- Varlıklar arasında daha karmaşık ilişkilere izin vermek için grafik benzeri bir model kullanır.
- Hiyerarsık modelden farklı olarak, her çocuğun birden fazla ebeveyne sahip olmasına izin verir ve bu da çoktan çoğa ilişkilere olanak tanır.
- Veriler, ilişkileri tanımlayan kayıtlar ve kümeler kullanılarak temsil edilir.
- Hiyerarsık modele göre daha esnekdir ve karmaşık veri bağlantıları olan uygulamalar için daha uygundur.
- Örnek: Entegre Veri Deposu (IDS), TurboIMAGE.

VTYS Türleri

6. Bulut Tabanlı Veritabanı

- Bunlar AWS, Azure veya Google Cloud gibi bulut bilişim platformlarında barındırılır.
- İsteğe bağlı ölçüklenebilirlik, yüksek erişilebilirlik, otomatik yedeklemeler ve uzaktan erişilebilirlik sunarlar.
- Bu veritabanları ilişkisel (SQL) veya ilişkisel olmayan (NoSQL) olabilir ve bulut hizmeti sağlayıcıları tarafından yönetilir, bu da yönetimsel yükü azaltır.
- Dağıtık erişim ve gerçek zamanlı analizler de dahil olmak üzere modern uygulama gereksinimlerini desteklerler.
- Örnek: Amazon RDS (SQL için), MongoDB Atlas (NoSQL için), Google BigQuery.

VTYS Türleri

Database Category		
SQL	NoSQL	Specialized

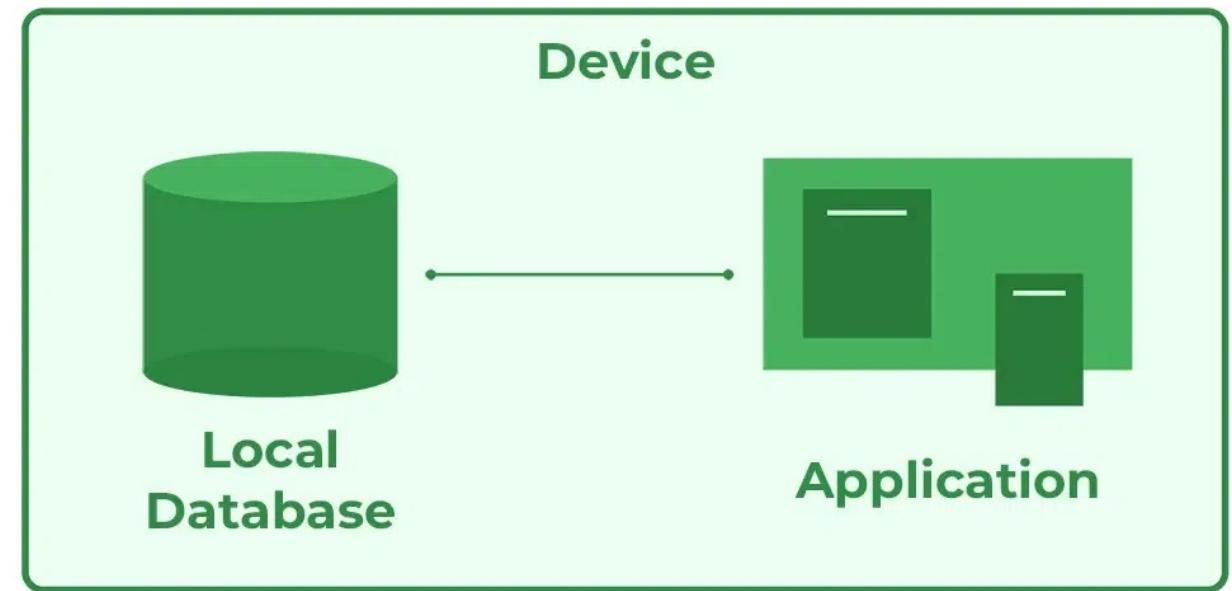
Type	Structure	Strengths	Best Use	Examples
Relational (RDBMS)	Tables (rows & columns)	Strong consistency, ACID compliance	Structured data, ERP	PostgreSQL, MySQL
Document Store	JSON-like documents	Flexible schema, scalable	Content management	MongoDB, CouchDB
Key-Value Store	Key-value pairs	Ultra-fast lookups	Caching, session storage	Redis, DynamoDB
Wide-Column Store	Column families	High write throughput	Time-series, logs	Cassandra, HBase
Graph Database	Nodes and edges	Efficient relationship queries	Social networks	Neo4j, Amazon Neptune
Time-Series DB	Time-stamped entries	Optimized for time-based queries	IoT, monitoring	InfluxDB, TimescaleDB
Object-Oriented DB	Objects (OOP)	Natural mapping to code	CAD, simulations	ObjectDB, db4o
Multi-Model DB	Multiple models	Versatility	Hybrid applications	ArangoDB, OrientDB
Hierarchical DB	Tree-like structure	Fast for hierarchical data	Legacy systems	IBM IMS
Network DB	Graph-like sets	Complex relationships	Legacy enterprise	IDS, TurboIMAGE
Columnar DB	Columns	Fast analytical queries	OLAP, warehousing	ClickHouse, Amazon Redshift
Vector DB	Vectors (embeddings)	Semantic search	AI/ML, similarity search	Pinecone, FAISS

VTYS Mimarisi

- Bir VTYS mimarisi, kullanıcıların bilgileri okumak, yazmak veya güncellemek için veritabanıyla nasıl etkileşim kurduğunu tanımlar.
- İyi tasarlanmış bir mimari ve şema (tabloları, alanları ve ilişkileri ayrıntılı olarak açıklayan bir taslak), veri tutarlığını sağlar, performansı artırır ve verileri güvende tutar.
- İhtiyaçlara göre kullandığımız çeşitli VTYS Mimarisi türleri bulunmaktadır.
 - 1 Katmanlı Mimari
 - 2 Katmanlı Mimari
 - 3 Katmanlı Mimari

VTYS Mimarisi – 1 Katmanlı

- 1 Katmanlı Mimari'de, kullanıcı aynı sistemdeki veritabanıyla doğrudan çalışır.
- Bu, istemci, sunucu ve veritabanının tek bir uygulamada bulunduğu anlamına gelir.
- Kullanıcı, ayrı bir sunucu veya ağ bağlantısına ihtiyaç duymadan uygulamayı açabilir, verilerle etkileşim kurabilir ve görevlerini gerçekleştirebilir.
- Yaygın bir örnek Microsoft Excel'dir.
- Kullanıcı arayüzünden mantığa ve veri depolamaya kadar her şey aynı cihazda gerçekleşir.
- Kullanıcı verileri girer, hesaplamalar yapar ve dosyaları doğrudan bilgisayarına kaydeder.



<https://www.geeksforgeeks.org/dbms>

VTYS Mimarisi – 1 Katmanlı

AVANTAJLAR

Basit Mimari: 1 Katmanlı Mimar, kurulumu en basit mimaridir, çünkü onu sürdürmek için yalnızca tek bir makine gereklidir.

Maliyet Etkin: 1 Katmanlı Mimar'yi uygulamak için ek bir donanıma ihtiyaç duyulmaz, bu da onu maliyet etkin kılar.

Kolay Uygulanabilir: 1 Katmanlı Mimar kolayca dağıtılabılır ve bu nedenle çoğunlukla küçük projelerde kullanılır.

DEZAVANTAJLAR

Tek Kullanıcıyla Sınırlıdır: Uygulamayı aynı anda yalnızca bir kişi kullanabilir. Birden fazla kullanıcı veya ekip çalışması için tasarlanmamıştır.

Zayıf Güvenlik: Her şey aynı makinede olduğundan, biri sisteme eriştiğinde hem verilere hem de uygulamaya kolayca erişebilir.

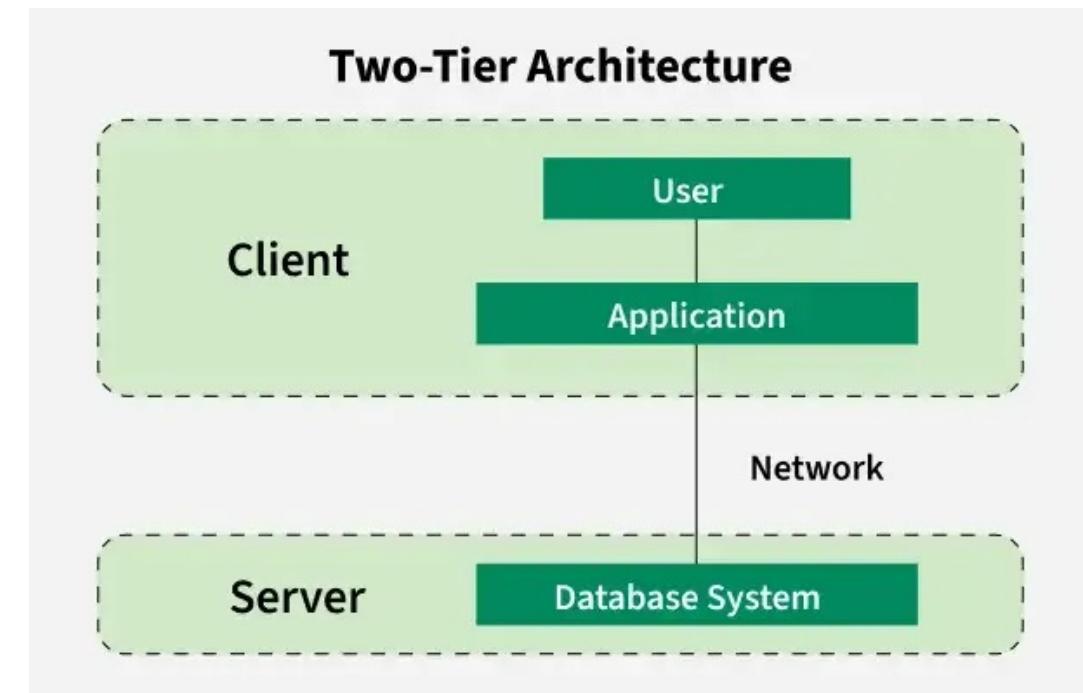
Merkezi Kontrol Yok: Veriler yerel olarak depolandığından, merkezi bir veritabanı yoktur. Bu durum, verileri birden fazla cihazda yönetmeyi veya yedeklemeyi zorlaştırır.

Veri Paylaşımı Zor: Kullanıcılar arasında veri paylaşımı zordur çünkü her şey tek bir bilgisayarda saklanır.

VTYS Mimarisi – 2 Katmanlı

- 2 katmanlı mimari, temel bir istemci-sunucu modeline benzer.
- İstemci tarafındaki uygulama, sunucu tarafındaki veritabanıyla doğrudan iletişim kurar.
- Bu etkileşim için ODBC ve JDBC gibi API'ler kullanılır. Sunucu tarafı ise sorgu işleme ve işlem yönetimi işlevlerini sağlamaktan sorumludur.
- İstemci tarafında, kullanıcı arayüzleri ve uygulama programları çalıştırılır. İstemci tarafındaki uygulama, VTYS ile iletişim kurmak için sunucu tarafıyla bağlantı kurar.

Örneğin: Okullarda veya küçük kuruluşlarda kullanılan bir Kütüphane Yönetim Sistemi, iki katmanlı mimarinin klasik bir örneğidir.



VTYS Mimarisi – 2 Katmanlı

- **İstemci Katmanı (1. Katman):** Bu, kütüphane personelinin veya kullanıcıların etkileşim kurduğu kullanıcı arayüzüdür. Örneğin, kitap aramak, kitap vermek veya teslim tarihlerini kontrol etmek için bir masaüstü uygulaması kullanabilirler.
- **Veritabanı Katmanı (2. Katman):** Veritabanı sunucusu, kitap detayları, kullanıcı bilgileri ve işlem günlükleri gibi tüm kütüphane kayıtlarını depolar.
- İstemci katmanı, veritabanı katmanına bir istek (bir kitap arama gibi) gönderir ve veritabanı katmanı isteği işleyip sonucu geri gönderir.
- Bu ayrılmış, istemcinin kullanıcı arayüzüne odaklanmasını sağlarken, sunucu veri depolama ve alma işlemlerini gerçekleştirir.

VTYS Mimarisi – 2 Katmanlı

AVANTAJLAR

Kolay Erişim: 2 Katmanlı Mimari, veritabanına kolay erişim sağlar, bu da hızlı geri alma imkanı sağlar.

Ölçeklenebilir: İstemci ekleyerek veya donanımı yükselterek veritabanını kolayca ölçeklendirebiliriz.

Düşük Maliyet: 2 Katmanlı Mimari, 3 Katmanlı Mimari ve Çok Katmanlı Mimari'den daha ucuzdur .

Kolay Dağıtım: 2 Katmanlı Mimari, 3 Katmanlı Mimari'ye göre dağıtımını daha kolaydır.

Basit: 2 Katmanlı Mimari, yalnızca iki bileşendenoluştugu için hem anlaşılır hem de basittir.

DEZAVANTAJLAR

Sınırlı Ölçeklenebilirlik: Kullanıcı sayısı arttıkça, sunucu çok fazla istekle aşırı yüklediğinden sistem performansı yavaşlayabilir.

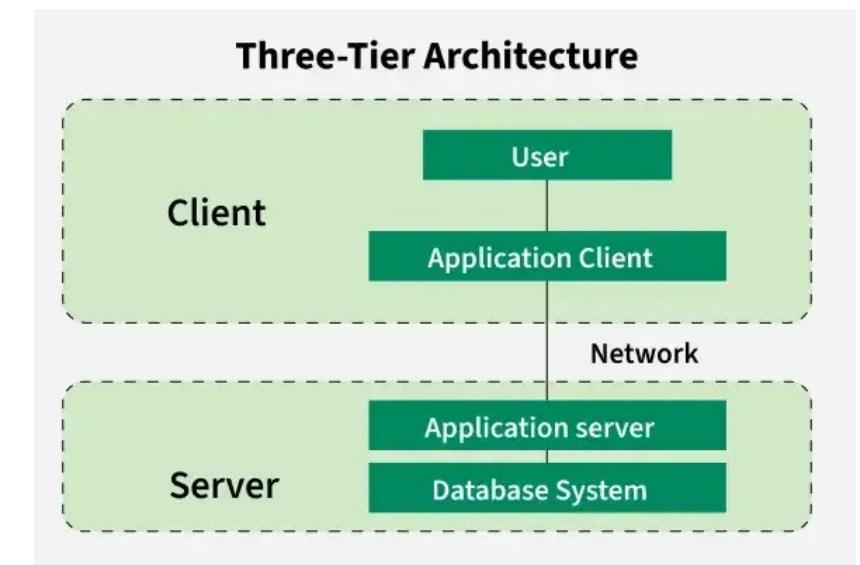
Güvenlik Sorunları: İstemciler doğrudan veritabanına bağlanır, bu da sistemi saldırlıara veya veri sızıntılarına karşı daha savunmasız hale getirebilir.

Sıkı Bağlantı: İstemci ve sunucu birbirine sıkı sıkıya bağlıdır. Veritabanı değiştiğinde, istemci uygulamasının da sıkılıkla güncellenmesi gereklidir.

Zorlu Bakım: Kullanıcı veya sistem sayısı arttıkça güncellemeleri yönetmek, hataları düzeltmek veya özellik eklemek daha da zorlaşır.

VTYS Mimarisi – 3 Katmanlı

- 3 Katmanlı Mimari'de, istemci ve sunucu arasında başka bir katman daha bulunur. İstemci, sunucuya doğrudan iletişim kurmaz. Bunun yerine, veritabanı sistemiyle iletişim kuran bir uygulama sunucusuyla etkileşime girer ve ardından sorgu işleme ve işlem yönetimi gerçekleşir.
- Bu ara katman, sunucu ve istemci arasında kısmen işlenmiş verilerin alışverişesi için bir ortam görevi görür. Bu mimari türü, büyük web uygulamalarında kullanılır.
- **Kullanıcı:** Bir online mağazayı ziyaret ediyorsunuz, bir ürün arıyorsunuz ve sepetinize ekliyorsunuz.
- **İşleme:** Sistem ürünün stokta olup olmadığını kontrol eder, toplam fiyatı hesaplar ve varsa indirimleri uygular.
- **Veritabanı:** Ürün detayları, sepetiniz ve sipariş geçmişiniz gelecekte referans olması amacıyla veritabanında saklanır.



VTYS Mimarisi – 3 Katmanlı

AVANTAJLAR

Gelişmiş Ölçeklenebilirlik: Uygulama sunucularının dağıtılmış dağılımı sayesinde ölçeklenebilirlik artırılmıştır. Artık istemci ve sunucu arasında ayrı ayrı bağlantı kurulmasına gerek yoktur.

Veri Bütünlüğü: 3 Katmanlı Mimar, Veri Bütünlüğünü korur. İstemci ve sunucu arasında bir ara katman bulunduğuundan, veri bozulması önlenebilir/kaldırılabilir.

Güvenlik: 3 Katmanlı Mimar Güvenliği Artırır. Bu tür bir model, istemcinin sunucuya doğrudan etkileşimiini önleyerek yetkisiz verilere erişimi azaltır.

DEZAVANTAJLAR

Daha Karmaşık: 3 Katmanlı Mimar, 2 Katmanlı Mimariye kıyasla daha karmaşıktır. İletişim Noktaları da 3 Katmanlı Mimaride iki katına çıkar.

Etkileşimin Zor Olması: Ara katmanların varlığı nedeniyle bu tür etkileşimin gerçekleşmesi zorlaşır.

Daha Yavaş Tepki Süresi: İstek ekstra bir katmandan (uygulama sunucusu) geçtiği için, 2 Katmanlı sistemlere kıyasla yanıt almak daha uzun sürebilir.

Daha Yüksek Maliyet: Üç ayrı katmanın (istemci, sunucu ve veritabanı) kurulumu ve bakımı daha fazla donanım, yazılım ve yetenekli personel gerektirir. Bu da maliyeti artırır.



YTU

YILDIZ TEKNİK
UNİVERSİTESİ

Yasemin Topuz

Yıldız Teknik Üniversitesi



ytopuz@yildiz.edu.tr

