VERİTABANI YÖNETİM SİSTEMLERİ
DERS NOTLARI

Öğr. Gör. Erkan Kaynak

İçindekiler

BÖLÜM 1: GİRİŞ VE TEMEL KAVRAMLAR

Veritabanı Nedir?

Veritabanı, bilgilerin düzenli bir şekilde saklandığı, yönetildiği ve gerektiğinde hızlıca erişilebildiği bir yapıdır. Genellikle bilgisayar sistemlerinde kullanılır ve özellikle büyük miktarda verinin güvenli, tutarlı ve erişilebilir bir biçimde tutulması için kullanılır.

Veritabanlarının Kullanım Alanları Nelerdir?

- Bankacılık sistemleri
- E-ticaret siteleri
- Sosyal medya platformları
- Hastane bilgi sistemleri
- Oyunlar (oyuncu verileri, skorlar vb.)

Veritabanı Türleri Nelerdir?

- İlişkisel Veritabanı (Relational Database)
  - Veriler tablolar halinde tutulur. (Örn: MySQL, SQL Server, Oracle)
- Nesne Tabanlı Veritabanı (Object-Oriented Database)
  - Veriler nesne olarak saklanır.
- NoSQL Veritabanı
  - Esnek veri yapıları kullanılır. (Örn: MongoDB, Firebase)

Veritabanı Yönetim Sistemi Nedir?

Veritabanı Yönetim Sistemi (DBYS ya da İngilizce kısaltmasıyla DBMS – Database Management System), veritabanlarını oluşturmak, düzenlemek, yönetmek ve kullanmak için geliştirilmiş yazılımlardır.

VTYS'nin Temel Görevleri Nelerdir?

- Veri Tanımlama:
  - Veritabanında hangi tür verilerin saklanacağını belirlemek (tablo, alan türleri vb.).
- Veri Saklama:
  - Verileri diskte organize bir şekilde saklar.
- Veri İşleme:
  - SQL gibi dillerle verileri ekleme, silme, güncelleme ve sorgulama işlemleri yapılır.
- Veri Güvenliği:
  - Kimlerin hangi verilere erişebileceği kontrol edilir.
- Veri Yedekleme ve Kurtarma:
  - Olası hata ya da sistem çökmesi durumlarında verilerin korunmasını sağlar.
- Çoklu Kullanıcı Desteği:
  - Aynı anda birden fazla kullanıcının veritabanına erişmesini sağlar.

Popüler VTYS Yazılımları Nelerdir?

DBMS Adı	Türü	Açıklama
MySQL	Açık kaynak	Web uygulamaları için yaygın
Microsoft SQL Server	Ticari	Microsoft'un kurumsal çözümü
Oracle Database	Ticari	Büyük ölçekli işletmeler için
PostgreSQL	Açık kaynak	Gelişmiş özellikleriyle bilinir
SQLite	Hafif DBMS	Mobil ve küçük uygulamalarda yaygın
MongoDB	NoSQL	Doküman tabanlı veritabanı sistemi

VTYS Neden Önemli?

VTYS ler olmasaydı:

- Verileri metin dosyalarında elle tutmak zorunda kalırdık.
- Karmaşık sorgular yapmak çok zor olurdu.
- Veri bütünlüğü ve güvenliği sağlanamazdı.
- Aynı anda birden fazla kişinin veriyle çalışması imkânsız olurdu.

BÖLÜM 2: VERİTABANI MODELLERİ

Veritabanı Modelleri Nedir?

Veritabanı modelleri, verilerin nasıl organize edileceğini, nasıl saklanacağını ve veriler arasındaki ilişkilerin nasıl tanımlanacağını belirleyen yapısal taslaklardır. Başka bir deyişle, verilerin mantıksal düzenini ifade eder.

Neden Veritabanı Modeli Kullanılır?

- Veriyi daha düzenli ve anlamlı bir şekilde saklamak için
- Veriler arası ilişkileri kurmak için
- Sorgulama ve veri yönetimini kolaylaştırmak için
- Veri bütünlüğünü ve güvenliğini sağlamak için

Başlıca Veritabanı Modelleri Nelerdir?

1. Hiyerarşik Model (Hierarchical Model)
    - Veriler ağaç yapısı şeklinde düzenlenir.
    - Her veri biriminin yalnızca bir üst verisi (ebeveyni) vardır.
    - Ana–alt ilişkisi vardır (Parent–Child).
    - Eski IBM sistemlerinde kullanılmıştır.
    - Avantaj: Hızlı veri erişimi
    - Dezavantaj: Esneklik az, ilişkiler sınırlı
2. Ağ Modeli (Network Model)
    - Hiyerarşik modele benzer, ancak her düğüm birden fazla ebeveyne sahip olabilir.
    - Çoktan çoğa (N:N) ilişkileri destekler.
    - Karmaşık yapılarda kullanılır.
    - Avantaj: Esnek yapı
    - Dezavantaj: Yapısı karmaşıktır, yönetimi zordur
3. İlişkisel Model (Relational Model) ⭐ (En yaygın model)
    - Veriler tablolar (relations) halinde saklanır.
    - Tablolar arasında anahtarlar aracılığıyla ilişki kurulur.
    - SQL dili ile yönetilir.
    - Avantaj: Kolay sorgulama, esneklik, güçlü veri bütünlüğü
    - Dezavantaj: Çok büyük verilerde performans sıkıntısı olabilir
4. Nesneye Yönelik Model (Object-Oriented Model)
    - Veriler, nesneler olarak temsil edilir (OO programlamaya benzer).
    - Veri ile birlikte, o veriye ait işlemler (metotlar) da saklanır.
    - Avantaj: Gerçek dünya modellerine uygun
    - Dezavantaj: Karmaşık, kullanımı yaygın değil
5. NoSQL Modeli (Not Only SQL)
    - Büyük veri ve esnek yapılar için kullanılır.
    - Tablo yapısı yerine belge, anahtar-değer, grafik veya sütun yapıları kullanılır.
    - Genellikle ilişkisel olmayan yapılarda tercih edilir.
    - Avantaj: Büyük veri ve yüksek performans
    - Dezavantaj: Veri bütünlüğü sınırlı, karmaşık ilişkiler zayıf
    - Alt türleri:
        - Doküman tabanlı: MongoDB
        - Anahtar-Değer tabanlı: Redis
        - Graf tabanlı: Neo4j
        - Sütun tabanlı: Cassandra

BÖLÜM 3: İLİŞKİSEL VERİTABANI MODELLERİ

İlişkisel Veritabanı Nedir?

İlişkisel Veritabanı (Relational Database), verilerin tablolar (table) şeklinde saklandığı ve bu tablolar arasında ilişkiler (relationships) kurularak verilerin organize edildiği bir veritabanı modelidir.

Temel Kavramlar Nelerdir?

Terim	Açıklama
Tablo (Table)	Verilerin tutulduğu yapıdır (örnek: Ogrenciler tablosu).
Satır (Row / Record)	Her satır bir veri kaydını temsil eder (örnek: bir öğrencinin bilgileri).
Sütun (Column / Field)	Verinin niteliğini belirtir (örnek: Ad, Soyad, Numara).
Birincil Anahtar (Primary Key)	Her satırı benzersiz şekilde tanımlayan sütun (örnek: ÖğrenciNo).
Yabancı Anahtar (Foreign Key)	Başka bir tablodaki birincil anahtara referans verir. İlişki kurar.

Normalizasyon Nedir?

Normalizasyon, veritabanı tasarımında verilerin doğru ve verimli bir şekilde organize edilmesini sağlayan bir işlemdir. Amaç; veri tekrarını (redundancy) azaltmak, veri tutarlılığını sağlamak ve veritabanının performansını artırmaktır.

Neden Normalizasyon Yapılır?

- Veri tekrarını önlemek
- Veri bütünlüğünü sağlamak
- Güncelleme, ekleme ve silme işlemlerindeki tutarsızlıkları önlemek
- Veritabanını daha anlaşılır ve yönetilebilir hale getirmek

Normalizasyon Aşamaları Nelerdir?

- 1NF (Birinci Normal Form) :
    - Her sütun atomik (bölünemez) olmalı. Çoklu değerler olmamalı.
- 2NF (İkinci Normal Form) :
    - 1NF + Tablodaki her alan, birincil anahtara tamamen bağlı olmalı.
- 3NF (Üçüncü Normal Form) :
    - 2NF + Tabloda, anahtar olmayan alanlar birbirine bağımlı olmamalı.

Sorgu (View) Nedir?

View (Görünüm / Sanal Tablo), bir veya birden fazla tablodan SQL sorgusu ile elde edilen sanal bir tablodur. Gerçekte fiziksel olarak veritabanında veri tutmaz, ancak kullanıcıya sanki bir tabloymuş gibi veri sunar.

Sorguların Özellikleri Nelerdir?

- Veritabanında fiziksel veri içermez.
- Güncel veriyi gösterir (kullanıldığı anda tablolardan çeker).
- SQL sorguları ile oluşturulur.
- Kullanıcıya karmaşık veriyi sadeleştirilmiş şekilde sunar.
- Güvenlik amacıyla bazı verileri gizlemek için de kullanılır.

Sorguların Kullanım Amaçları Nelerdir?

- Karmaşık sorguları sadeleştirmek
- Veriye erişimi sınırlamak (bazı sütunları gizlemek)
- Tekrar kullanılabilir sorgular oluşturmak
- Raporlama ve analizlerde kolaylık sağlamak

MS ACCESS

- Microsoft tarafından geliştirilen bir veritabanı yönetim sistemidir (DBMS).
- Küçük ve orta ölçekli veritabanı uygulamaları için uygundur.
- Hem veri saklama hem de arayüz tasarımı yapılabilir (formlar, raporlar vb.)
- Dosya uzantısı *.mdb ve *.accdb dir.
- Sınırlamaları:
    - Büyük ve çok kullanıcılı sistemler için uygun değildir.
    - Web tabanlı uygulamalar için sınırlı destek sunar.
    - Performansı yüksek veritabanlarına göre düşüktür (örneğin: SQL Server).
- Avantajları:
    - Kullanımı kolay, arayüzü görseldir.
    - Küçük ölçekli projeler için idealdir.
    - SQL bilgisi olmadan da sorgular oluşturulabilir.
    - Excel gibi Office uygulamalarıyla entegre çalışabilir.

BÖLÜM 4: SQL - STRUCTURED QUERY LANGUAGE

SQL Nedir?

SQL (Structured Query Language), yani Yapılandırılmış Sorgu Dili, veritabanları ile iletişim kurmak için kullanılan standart bir dildir. SQL, ilişkisel veritabanlarındaveri oluşturmak, okumak, güncellemek ve silmek amacıyla kullanılır.

SQL'in Temel Amaçları Nelerdir:

1. Veri tanımlama (DDL – Data Definition Language) :
   Veritabanı ve tablo oluşturma, değiştirme, silme işlemleri için kullanılır.

2. Veri işleme (DML – Data Manipulation Language) :
   Veri ekleme, güncelleme, silme ve listeleme işlemleri yapılır.

3. Veri sorgulama (en sık kullanılan kısım) :
   Veritabanındaki verileri filtreleyerek, sıralayarak, gruplandırarak alma işlemleridir.

4. Yetkilendirme ve güvenlik (DCL – Data Control Language) :
   Kullanıcılara izin verme ya da izinleri kaldırma işlemleri yapılır.

SQL 'de Veri Sorgulama İşlemleri Nasıl Yapılır?

Veri sorgulama işlemleri, SQL'de en çok kullanılan işlemlerdir ve genellikle SELECT komutu ile gerçekleştirilir. Bu komut, veritabanındaki tablolardan istenilen verileri okumak ve listelemek için kullanılır.

1. **Temel Sorgu:**

   *SELECT * FROM Ogrenciler;*

   Ogrenciler tablosundaki tüm kayıtları ve sütunları listeler.

2. **Belirli Sütunları Listeleme**

   SELECT adi, Soyadi FROM Ogrenciler

   Sadece adi ve soyadi sütunlarını listeler.

3. **Şartlı (Filtreli) Sorgu – WHERE**

   *SELECT * FROM Ogrenciler WHERE adi = 'Ali'*

   Sadece adı "Ali" olan öğrencileri listeler.

4. **Karşılaştırma Operatörleri**

   - = (eşittir)
   - != veya <> (eşit değil)
   - <, >, <=, >= (küçük, büyük, küçük eşit, büyük eşit)
   - BETWEEN ... AND ... (iki değer arasında)
   - IN (...) (belirtilenler içinde)
   - LIKE (benzerlik arama)

```
SELECT * FROM Ogrenciler WHERE OgrenciID > 5;  --    Öğrenci ID değeri 5 den büyük olanlar.
SELECT * FROM Ogrenciler WHERE Ad LIKE 'A%';    -- Adı A ile başlayanlar.
SELECT * FROM Ogrenciler WHERE Soyad IN ('Yılmaz', 'Demir'); -- Soyadı Yılmaz veya Demir olanlar.
```

5. **Sıralama – Order By**

   *SELECT * FROM Ogrenciler ORDER BY Ad ASC;*

   Öğrenciler tablsıundaki kayıtları Ad'a göre sıralar.

   ASC: Artan Sıra, DESC: Azalan Sıra

6. **Tekrarlayanları Kaldırma – Distinct**

   SELECT DISTINCT Sehir FROM Ogrenciler;

   Aynı şehir ismini tekrar etmeden listeler.

7. **Sütunlara İsim Verme**

   SELECT Ad AS Isim, Soyad AS Soyisim FROM Ogrenciler;

   Ad ve Soyad yerine sütun başlığı olarak İsim ve Soyisim kullanır.

8. **Fonksiyonlar**
   - Count() : Kayıt sayısı.
   - SUM() : Verilen sütunun tüm kayıtlardaki toplamı.
   - Max(): Verilan sütunun tüm kayıtlardaki en yüksek değeri.
   - Min: Verilen sütunun tüm kayıtlardaki en küçük değeri.
   - Avg(): Verilen sütünün tüm kayıtlardaki ortalaması.

   SELECT Count(*) From Ogrenciler; -- Öğrenciler tablosundaki kayıt sayısı.

   SELECT Avg(yas) FFROM Ogrenciler; -- Öğrencilerin yaş ortalaması.

9. **Gruplama**

   SELECT Sehir, COUNT(*) AS OgrenciSayisi FROM Ogrenciler GROUP BY Sehir;

   Öğrenciler tablosundaki Şehirleri ve o şehirde bulunan kayıt sayılarını listeler.

10. **Kayıt Sayısını Sınırlama – Limit, Top**

    *SELECT Top 5 * FROM Ogrenciler;* -- SQL Server

    *SELECT * FROM Ogrenciler LIMIT 5;* -- My SQL

    Öğrenciler tablosundaki ilk 5 kaydı listeler.

11. **Birleştirmeler – Join**

    İki veya daha fazla tabloyu ortak bir sütun üzerinden birleştirir. En yaygın kullanımı yabancı anahtar (foreign key) ilişkisi olan tablolardır.

    SELECT Ogrenci.AdiSoyadi, Fakulte.FakulteAdi
    FROM Ogrenci
    INNER JOIN Fakulte ON Ogrenci.FK_FakulteID = Fakulte.ID

    Öğrenci ve Fakülte tablolarında eşleşen kayıtları bulur, öğrencinin adı soyadı ve fakültenin adı bilgilerini listeler.

    SELECT Ogrenci.AdiSoyadi, Fakulte.FakulteAdi
    FROM Ogrenci
    LEFT JOIN Fakulte ON Ogrenci.FK_FakulteID = Fakulte.ID

    Öğrenci tablosundaki tüm kayıtları listeler. Eğer bir fakülte kaydı ile eşleşme varsa o fakültenin adını da getirir.

    SELECT Ogrenci.AdiSoyadi, Fakulte.FakulteAdi
    LEFT FROM Ogrenci
    RIGHT JOIN Fakulte ON Ogrenci.FK_FakulteID = Fakulte.ID

    Fakülte tablosundaki tüm kayıtları listeler. Eğer bir öğrenci ile eşleşiyorsa o öğrencinin adını soyadını da getirir.

SQL de Veri İşleme (DML) İşlemleri Nasıl Yapılır?

SQL'de Veri İşleme (DML - Data Manipulation Language) komutları, veritabanındaki veriler üzerinde işlem yapmak için kullanılır. Bu işlemler veri ekleme (INSERT), güncelleme (UPDATE), silme (DELETE) ve okuma (SELECT) işlemleridir. Ancak DML terimi genellikle veri ekleme, silme ve güncelleme işlemleri için kullanılır (SELECT ise genellikle sorgulama olarak ayrı incelenir).

1. **Veri Ekleme – INSERT**

   INSERT INTO Ogrenciler (ID, Ad, Soyad) VALUES (1, 'Ali', 'Yılmaz');

   Alan sırası ve değerlerin sıraları aynı olmalıdır.

   Birden fazla kayıt eklemek için değerler art arda girilebilir.

   INSERT INTO Ogrenciler (ID, Ad, Soyad) VALUES (1, 'Ali', 'Yılmaz'), (2, ‘Ayşe’, ‘Suna’);

2. **Veri Güncelleme – UPDATE**

   UPDATE Ogrenciler SET Ad = 'Ahmet' WHERE ID = 1;

   Öğrenciler tablosunda ID değeri 1 olan kaydın Ad değeri Ahmet olarak değiştirilir.

   Dikkat: WHERE komutu kullanılmazsa tablodaki tüm kayıtlar güncellenir.

   Çoklu sütun değeri tek seferde güncellenebilir.

   UPDATE Ogrenciler SET Ad = 'Ahmet', Soyadi=’Kara’ WHERE ID = 1;

3. **Veri Silme – DELETE**

   DELETE FROM Ogrenciler WHERE ID = 1;

   Öğrenciler tablosunda, ID değeri 1 olan kayıt silinir.

   Dikkat: WHERE komutu kullanılmazsa tüm kayıtlar silinir.

SQL de Veri Tanımlama (DDL) İşlemleri Nasıl Yapılır?

SQL’de veri tanımlama işlemleri (DDL – Data Definition Language), veritabanı nesnelerini oluşturmak, değiştirmek veya silmek için kullanılır. Bu işlemlerle tablo, sütun, veritabanı gibi yapılar tanımlanır ve yönetilir.

1. **Veritabanı Oluşturma – Create Database**

   CREATE DATABASE OkulVeritabani;

2. **Tablo Oluşturma – Create Table**

```
CREATE TABLE Ogrenciler (
   ID INT PRIMARY KEY,
   Ad VARCHAR(50),
   Soyad VARCHAR(50),
   DogumTarihi DATE
);
```

   Veritabanında Öğrenciler isminde bir tablo yaratır.

   ID Tamsayı ve Birincil Anahtar, Ad ve Soyad 50 karakterlik metin, DogumTarihi ise Tarih tipinde alanlar olarak yaratılır.

   Otomatik Sayı olarak atamak için IDENTITY (SQL Server) veya AUTO_INCREMENT (MySQL) seçenekleri kullanılır.

3. **Tabloya Yeni Bir Sütun Ekleme – ALTER, ADD**

   ALTER TABLE Ogrenciler ADD Telefon VARCHAR(20);

   Öğrenciler tablosuna Telefon adında ve 20 karakter uzunluğunda yeni bir sütun ekler.

4. **Var Olan Sütunu Değiştirme – Alter, Alter Column**

   ALTER TABLE Ogrenciler ALTER COLUMN Ad VARCHAR(100);

   Öğrenciler tablosundaki Ad sütununun maksimum karakter uzunluğunu 100 karaktere çıkartır.

5. **Sütun Silme – Alter, Drop Column**

   ALTER TABLE Ogrenciler DROP COLUMN Telefon;

   Öğrenciler tablosundaki Telefon sütununu kaydedilen verilerle beraber siler.

6. **Tablo Silme – Drop Table**

   DROP TABLE Ogrenciler;

   Öğrenciler tablosunu ve içindeki verileri veri tabanından siler.

BÖLÜM 5: İLİŞKİSEL VERİ TABANLARI

İlişkisel Veritabanı, verileri tablolar (relations) halinde tutan ve aralarındaki ilişkileri tanımlayabilen bir veritabanı modelidir.

Temel Kavramlar

Terim	Açıklama
Tablo (Table)	Verilerin tutulduğu temel yapıdır. Satır ve sütunlardan oluşur.
Satır (Row - Record)	Her bir veri kaydıdır.
Sütun (Column - Field)	Veri alanının adını ve veri türünü belirler.
Birincil Anahtar (Primary Key)	Her satırı benzersiz şekilde tanımlar.
Yabancı Anahtar (Foreign Key)	Başka bir tablodaki birincil anahtara referans verir.
İlişki (Relation)	Tablolar arasındaki bağlantıdır.

Şema (Schema)

- Veritabanındaki tabloların, sütunların, veri türlerinin ve ilişkilerin tanımıdır.
- İlişkisel veritabanlarında katı şema yapısı vardır:
- Veri türleri ve alanlar önceden tanımlanır.

ACID

ACID, veritabanı işlemlerinin güvenilirliğini ve bütünlüğünü sağlamak için kullanılan dört temel ilkedir.

Harf	Açılımı	Türkçesi	Açıklama
A	Atomicity	Atomiklik	Bir işlem ya tamamen gerçekleşir ya da hiç gerçekleşmez.
C	Consistency	Tutarlılık	İşlem tamamlandığında veritabanı geçerli bir durumda kalmalıdır.
I	Isolation	Yalıtım	Aynı anda çalışan işlemler birbirini etkilemeden çalışmalıdır.
D	Durability	Kalıcılık	İşlem tamamlandığında yapılan değişiklikler kalıcıdır, sistem çökse bile kaybolmaz.

Popüler İlişkisel Veritabanları

Veritabanı	Açıklama
SQL Server	Microsoft'un ilişkisel veritabanıdır. Kurumsal uygulamalarda sık kullanılır.
MySQL	Açık kaynak kodlu, web projelerinde popülerdir.
PostgreSQL	Gelişmiş özelliklere sahip, açık kaynak bir veritabanıdır.
Oracle Database	Büyük kurumsal sistemlerde tercih edilen güçlü bir RDBMS'tir.
SQLite	Hafif yapısıyla mobil ve gömülü sistemlerde tercih edilir.

SQL SERVER

Microsoft SQL Server, Microsoft tarafından geliştirilen bir ilişkisel veritabanı yönetim sistemidir. Verilerin tablolar halinde düzenlenmesini sağlar ve SQL dili ile veriler üzerinde sorgular gerçekleştirilir.

Geliştirici: Microsoft
Platform: Windows ağırlıklı, ancak Linux desteği de sunuyor.
Avantaj: Kurumsal projeler için uygundur, .NET ile iyi entegre olur.
Kullanım Alanları: Bankacılık, ERP sistemleri, büyük şirket uygulamaları.

SQL Server Management Studio

SQL Server Management Studio (SSMS), Microsoft SQL Server için geliştirilmiş resmi grafiksel yönetim aracıdır. Hem yeni başlayanlar hem de ileri düzey kullanıcılar için SQL Server'ı yönetmeyi, sorgu yazmayı ve veri tabanlarını görsel olarak kontrol etmeyi kolaylaştırır.

SQL Server Nasıl Çalışır?

SQL Server, temel olarak şu bileşenlerle çalışır:

Bileşen	Açıklama
Database Engine	Veritabanı işlemlerini (sorgular, transaction'lar, depolama) yöneten ana motor
SQL Server Agent	Zamanlanmış görevleri çalıştıran servis
SQL Browser	Sunucu ve instance'lar arasında bağlantı yönetimini kolaylaştırır
SSMS (İstemci Aracı)	Yönetici tarafından sorguların yazıldığı ve verilerin görüntülendiği arayüz
Protocol Stack	SQL Server'a TCP/IP, Named Pipes, Shared Memory gibi yollarla bağlanmayı sağlar

Bulut Platformları (Azure SQL)

SQL Server, Microsoft'un bulut hizmeti olan **Azure** üzerinde de barındırılabilir.

Seçenekler:

- **Azure SQL Database**: Tamamen yönetilen veritabanı (sunucusuz, PaaS).
- **Azure SQL Managed Instance**: Sunucuya daha yakın, yüksek uyumluluklu yapı.
- **Azure VM Üzerinde SQL Server**: Geleneksel sunucu kurulumu, ama sanal ortamda.

Avantajlar:

- Yedekleme, güncelleme, yama gibi işlemler Microsoft tarafından yapılır.
- Ölçeklenebilirlik ve yüksek erişilebilirlik özellikleri otomatik gelir.
- Dünya genelinde erişim sağlanabilir.

Stored Procedure

Stored Procedure, veritabanında saklanan ve gerektiğinde çalıştırılabilen **önceden tanımlı SQL komutları kümesidir**. Genellikle tekrar eden işlemleri otomatikleştirmek ve kod tekrarını azaltmak için kullanılır.

Amaç	**Açıklama**
Tekrarlanan işlemleri topluca yürütme	Sipariş oluşturma işlemi gibi birden fazla SQL komutunu bir arada çalıştırır.
Güvenlik sağlar	Kullanıcıya tabloya doğrudan erişim vermeden işlem yaptırabilir.
Performans iyileştirmesi	Derlenmiş olarak saklanır, tekrar çalıştırıldığında daha hızlıdır.
Kodun yönetimini kolaylaştırır	Tüm işlemler merkezi bir kodda toplanır, değişiklikler daha kolay yapılır.

Örn:

```
CREATE PROCEDURE Ogrenci_Bul
   @OgrenciNo INT,
AS
BEGIN
   SELECT * FROM Ogrencier WHERE OgrenciNo = @OgrenciNo;
END;
```

Çalıştırmak için:

```
EXEC Ogrenci_Bul @OgrenciNo = 2025001
```

MYSQL

MySQL, açık kaynak kodlu, ilişkisel bir veritabanı yönetim sistemidir. Verileri tablolar halinde saklar ve yönetir. SQL (Structured Query Language) dilini kullanır. Özellikle web tabanlı uygulamalarda (örneğin WordPress, Joomla, PHP projeleri) çok yaygın olarak kullanılmaktadır. Şu anda Oracle Corporation tarafından geliştirilmektedir.

Temel Özellikleri

Özellik	Açıklama
Açık kaynak	Ücretsizdir, topluluk ve ticari sürümleri vardır.
İlişkisel veritabanı	Veriler tablolar halinde saklanır, ilişkiler kurulabilir.
Hızlı ve hafif	Web projeleri ve küçük-orta ölçekli sistemler için çok uygundur.
Kullanıcı yönetimi ve güvenlik	Yetkilendirme, şifreleme, kullanıcı bazlı erişim gibi özellikler sunar.
Çok platformlu	Windows, Linux, macOS üzerinde çalışabilir.
Uyumluluk	PHP, Python, Java, .NET gibi birçok dille entegre edilebilir.

MySQL Bağlantı ve Yönetim

MySQL veritabanına bağlanmak ve yönetmek için hem grafik arayüzlü araçlar hem de komut satırı seçenekleri mevcuttur.

Araç	Açıklama
MySQL Workbench	Grafik arayüzle veritabanı yönetimi
phpMyAdmin	Tarayıcı üzerinden MySQL yönetimi (özellikle PHP projeleriyle birlikte)
Komut Satırı (CLI)	mysql -u root -p komutu ile sunucuya bağlanılır

SQLite

SQLite, sunucuya ihtiyaç duymayan, gömülü (embedded) ve ilişkisel bir veritabanı yönetim sistemidir.

Veriler bir tek dosyada saklanır ve uygulama ile birlikte çalışır.

Temel Özellikleri

Özellik	Açıklama
Dosya tabanlı	Tüm veritabanı bir .sqlite, .db veya .sqlite3 dosyasında saklanır.
Sunucusuz	MySQL veya SQL Server gibi ayrı bir sunucuya ihtiyaç duymaz.
Gömülü yapı	Veritabanı motoru, uygulamaya doğrudan gömülür (özellikle mobil ve masaüstü uygulamalarda).
SQL destekli	Standart SQL dilinin büyük kısmını destekler.
Hafif ve hızlı	Küçük-orta ölçekli projeler için idealdir.

Nerelerde Kullanılır?

Alan	Açıklama
Mobil uygulamalar	Android ve iOS uygulamalarında en yaygın kullanılan veritabanıdır.
Masaüstü uygulamaları	Python, C#, Java ile geliştirilen basit uygulamalarda kullanılır.
Tarayıcılar	Firefox, Chrome gibi tarayıcılar SQLite ile veri saklar.
Gömülü sistemler	IoT cihazları, akıllı cihazlar gibi sistemlerde tercih edilir.

DB Browser

Kullanımı kolay, grafik ara yüzlü bir programdır. Tablolar, kayıtlar ve SQL komutları görsel olarak yönetilebilir.

Avantajları Nelerdir?

- Sunucu kurulumu gerekmez, kurulumu sıfırdır
- Hızlı ve hafiftir
- Tek dosyalık yapısıyla taşınabilir
- Açık kaynak ve ücretsiz
- Küçük projeler, prototipler ve testler için idealdir

Dezavantajları Nelerdir?

- Çok kullanıcılı sistemlerde uygun değildir (tek kullanıcılı/tek bağlantılı senaryolar için idealdir)
- Gelişmiş veritabanı özellikleri (çoklu kullanıcı yönetimi, roller vb.) sınırlıdır
- Büyük veritabanları için uygun değildir (terabaytlarca veri işlemek için tasarlanmamıştır)

BÖLÜM 6: İLİŞKİSEL OLMAYAN VERİTABANLARI

NoSQL geleneksel ilişkisel veri tabanlarının dışında, farklı veri modellerini destekleyen veri tabanlarıdır. Özellikle büyük veri, yüksek hız ve esneklik gereken uygulamalarda tercih edilir.

Temel Özellikeri

Özellik	Açıklama
Şemasız (Schema-free)	Veriler önceden belirlenmiş katı yapıya bağlı kalmaz, esnek veri modelleri kullanılır.
Yatay Ölçeklenebilirlik	Veri tabanı, çok sayıda sunucuya (node) yayılabilir ve bu sayede büyük veri kolayca yönetilir.
Yüksek Performans	Büyük veri üzerinde hızlı okuma-yazma işlemleri için optimize edilmiştir.
Esnek Veri Modelleri	Belge, anahtar-değer, sütunlu, grafik gibi farklı veri tiplerini destekler.
ACID yerine BASE	Katı ACID kuralları yerine daha esnek tutarlılık ilkeleri uygular (BASE: Basically Available, Soft state, Eventually consistent).

NoSQL Veritabanlarının Avantajları Nelerdir?

- Esnek şema sayesinde veri yapıları hızlı değişebilir
- Büyük veri ve gerçek zamanlı uygulamalarda yüksek performans
- Yatay ölçeklenebilirlik ile sistem kolayca büyütülebilir
- Karmaşık ilişki ve hiyerarşi gerektirmeyen veri yapıları için ideal
- Bulut ve dağıtık sistemlerle uyumludur

NoSQL Veritabanlarının Dezavantajları Nelerdir?

- Geleneksel SQL sorguları ve ilişkisel işlemler desteklenmez veya sınırlıdır
- ACID garantileri genellikle yoktur, bu da veri tutarlılığı açısından risk oluşturabilir
- Karmaşık sorgu işlemleri ve çok tablolu ilişkiler zor olabilir
- Bazı NoSQL sistemler olgunluk ve standartlaşma açısından RDBMS kadar gelişmiş değildir

NoSQL Veritabanları Nerelerde Kullanılır?

- Sosyal medya platformları (Facebook, Twitter)
- Büyük veri ve analiz uygulamaları
- Gerçek zamanlı veri işleme (örneğin oyunlar, IoT)
- İçerik yönetim sistemleri (CMS)
- E-ticaret ürün katalogları, öneri sistemleri

İlişkisel ve NoSQL Veritabanları Karşılaştırılması

Özellik	İlişkisel Veritabanları (RDBMS)	NoSQL Veritabanları
Veri Yapısı	Tablo (satır ve sütun)	Esnek yapılar: belge, anahtar-değer, grafik, sütun
Veri İlişkileri	Tablolar arasında ilişkiler (foreign key) ile	Genellikle ilişkiler yoktur veya zayıftır
Şema (Schema)	Katı şema yapısı	Şemasız veya esnek şema
Sorgu Dili	SQL (Structured Query Language)	Kendi özel sorgulama sistemleri
ACID Uyumu	Evet	Genellikle BASE prensibi (esneklik ön planda)
Ölçeklenebilirlik	Dikey ölçeklenir (+donanım)	Yatay ölçeklenir (+sunucu)
Performans (B.V.)	Karmaşık ilişkilerde güçlü	Yüksek hacimli, hızlı okuma-yazma için güçlü
Veri Tipi Desteği	Sayılar, metin, tarih vb.	JSON, BSON, XML, binary, metin vb.
Kullanım Alanı	Bankacılık, muhasebe, stok yönetimi	Sosyal medya, IoT, büyük veri, içerik yönetimi
Popüler Sistemler	SQL Server, MySQL, PostgreSQL, Oracle	MongoDB, Redis, Cassandra, CouchDB, Neo4j

İlişkisel Veritabanı	NoSQL Veritabanı (MongoDB)

-- Öğrenci tablosu
CREATE TABLE Ogrenci (
OgrID INT PRIMARY KEY,
Ad VARCHAR(50),
Soyad VARCHAR(50)
);
 	
-- Not tablosu
CREATE TABLE Notlar (
NotID INT PRIMARY KEY,
OgrID INT,
Puan INT,
FOREIGN KEY (OgrID) REFERENCES Ogrenci(OgrID)
);
	{
	"_id": 1,
	"Ad": "Ali",
	"Soyad": "Yılmaz",
	"Notlar": [
	{ "Ders": "Matematik", "Puan": 85 },
	{ "Ders": "Fizik", "Puan": 78 }
	]
	}

Tabloar arasında ikişki vardır. Veri yapısı sabittir.	Veriler JSON benzeri bir belge içinde tutulur, ilişkili veri aynı belgeye gömülür. Şema zorunlu değildir.

Hangi Durumda Hangisi Tercih Edilmeli?

Durum / İhtiyaç	Önerilen Veritabanı Türü
Veri tutarlılığı kritikse (ACID)	İlişkisel Veritabanı
Karmaşık ilişkili veriler varsa	İlişkisel Veritabanı
Büyük veri, esneklik, hız gerekiyorsa	NoSQL
Esnek veri yapısı gerekiyorsa	NoSQL
Gerçek zamanlı analiz, içerik sistemi	NoSQL
Geleneksel iş uygulamaları	İlişkisel Veritabanı

Mongo DB

MongoDB, belge (document) tabanlı, açık kaynak kodlu, NoSQL türü bir veritabanıdır. JSON benzeri yapıları (BSON) kullanarak esnek ve şemasız veri saklama imkânı sunar.

MongoDB, yüksek performans, ölçeklenebilirlik ve kolay kullanım odaklı bir veritabanı sistemidir.

Temel Özellikleri

Özellik	Açıklama
Belge Tabanlı	Veriler JSON benzeri belgeler (documents) olarak saklanır.
Şema (Schema) zorunluluğu yok	Aynı koleksiyondaki belgeler farklı yapıda olabilir.
Yüksek performanslı	Büyük veri ve gerçek zamanlı uygulamalar için idealdir.
Yatay ölçeklenebilirlik	Veriler birden fazla sunucuya kolayca yayılabilir (sharding).
Zengin sorgulama dili	Belgeler üzerinde güçlü sorgular yapılabilir.
BSON kullanır	JSON'ın binary hali olan BSON formatı sayesinde hızlı işlem yapılır.

Nerelerde Kullanılır?

- Gerçek zamanlı analiz sistemleri
- İçerik yönetim sistemleri
- E-ticaret ürün katalogları
- Mobil uygulamalar
- IoT sistemleri
- Oyun geliştirme sunucuları

Temel Yapılar

- Veritabanı (Database): MongoDB sunucusu içinde birçok veritabanı olabilir.
- Koleksiyon (Collection): Tabloya benzeyen yapılardır, ama şemasızdır.
- Belge (Document): Asıl verinin tutulduğu yerdir. JSON benzeri yapıdadır.

Örnek Belge:

```
{
  "_id": 1,
  "ad": "Ahmet",
  "yas": 25,
  "meslek": "Yazılımcı"
}
```

Popüler MongoDB Araçları

- MongoDB Atlas: MongoDB'nin resmi bulut çözümüdür.
- Compass: MongoDB için grafiksel kullanıcı arayüzü (GUI).
- Mongoose: Node.js ile MongoDB kullanımı için popüler bir kütüphane.