

# Ağ (Network) Temelleri

## TCP/IP Nedir? Özellikleri Nelerdir?

TCP/IP, internetin ve çoğu ağın temel iletişim protokolleri kümesidir. Bilgisayarların ve cihazların ağ üzerinden veri alışverişini tanımlar.

### Temel Özellikleri:

- **Katmanlı Mimari:** Belirli işlevleri yerine getiren katmanlardan oluşur: Uygulama, Taşıma, İnternet ve Ağ Arayüzü katmanları.
- **Paket Anahtarlama:** Veri, paketlere bölünür, bağımsız yönlendirilir ve hedefte birleştirilir.
- **Adresleme:** Her cihaza benzersiz bir IP adresi atanır.
- **Yönlendirme:** Veri paketlerinin en uygun yolu belirlenir (router'lar tarafından).
- **Güvenilirlik (TCP):** Veri iletiminde güvenliği ve sıralı teslimi sağlar.
- **Bağlantısız İletişim (UDP):** Hızın öncelikli olduğu durumlar için kullanılır.
- **Standartlaştırma ve Açıklık:** Farklı cihazların uyumlu çalışmasını sağlar.
- **Esneklik ve Ölçeklenebilirlik:** Küçük yerel ağlardan küresel internete kadar uyum sağlar.
- **Platform Bağımsızlığı:** Farklı işletim sistemleri ve donanımlarda çalışır.

TCP/IP, günümüz internetinin çalışmasını sağlayan temel kurallar bütünüdür.

## OSI Katmanları Nelerdir? Her Katmanın Görevi Nedir?

OSI (Open Systems Interconnection) modeli, ağ iletişiminin nasıl gerçekleştiğini tanımlayan 7 katmanlı kavramsal bir çerçevedir. Her katman belirli bir görevi yerine getirir.

### OSI Katmanları ve Görevleri:

1. **Fiziksel Katman (Katman 1):** Verinin fiziksel ortam üzerinden bitler halinde iletimini tanımlar (kablolar, sinyaller vb.). Örnekler: Ethernet kabloları, Wi-Fi sinyalleri.
2. **Veri Bağlantı Katmanı (Katman 2):** Fiziksel katman üzerinden hatasız veri transferinden sorumludur. Veriyi çerçevelere böler, MAC adreslerini yönetir, hata tespiti yapar. İki alt katmanı vardır: LLC ve MAC. Örnekler: Ethernet, Wi-Fi, switch'ler.

3. **Ağ Katmanı (Katman 3):** Farklı ağlar arasında veri paketlerinin yönlendirilmesinden (routing) sorumludur. Mantıksal adreslemeyi (IP adresleri) yönetir, en iyi yolu belirler. Örnekler: IP, yönlendiriciler (router).
4. **Taşıma Katmanı (Katman 4):** Uçtan uca güvenilir veya güvenilirsiz veri iletimini sağlar. Veriyi segmentlere böler, akış ve hata kontrolü sunar. Örnekler: TCP, UDP.
5. **Oturum Katmanı (Katman 5):** Uygulamalar arasındaki iletişim oturumlarını kurar, yönetir ve sonlandırır. Örnekler: NetBIOS, RPC.
6. **Sunum Katmanı (Katman 6):** Verinin uygulamalar tarafından anlaşılabilir formatta sunulmasını sağlar. Formatlama, şifreleme/şifre çözme, sıkıştırma işlemlerini gerçekleştirir. Örnekler: JPEG, GIF, SSL/TLS.
7. **Uygulama Katmanı (Katman 7):** Kullanıcının doğrudan etkileşimde bulunduğu veya ağ servislerine erişim sağlayan protokolleri içerir. Örnekler: HTTP, FTP, SMTP, DNS.

OSI modeli, ağ iletişimi kavramlarını anlamak için önemli bir referans noktasıdır.

### IP Adresi Nedir? Türleri Nelerdir?

IP adresi, ağa bağlı cihazları benzersiz şekilde tanımlayan sayısal bir etikettir. Verilerin doğru hedefe yönlendirilmesini sağlar.

#### IP Adresi Türleri:

- **Kullanılan Protokol Sürümüne Göre:**
  - **IPv4:** 32 bitlik adres (örn: 192.168.1.1). Adres sıkıntısı yaşanmaktadır.
  - **IPv6:** 128 bitlik adres (örn: 2001:0db8::). Daha geniş adres havuzu ve gelişmiş özellikler sunar.
- **Erişilebilirlik ve Kapsama Alanına Göre:**
  - **Genel (Public) IP Adresi:** İnternet üzerinde benzersizdir, İSS tarafından atanır ve dünya çapında erişilebilirliği sağlar.
  - **Özel (Private) IP Adresi:** Yerel ağ içinde kullanılır, internette doğrudan yönlendirilemez. NAT ile internete erişir.
- **Atanma Yöntemine Göre:**
  - **Dinamik IP Adresi:** DHCP sunucusu tarafından otomatik olarak atanır, yönetimi kolaydır. Zamanla değişebilir.
  - **Statik IP Adresi:** Cihaza kalıcı olarak atanır, değişmez. Sunucular ve belirli ağ cihazları için gereklidir.

- **Kullanım Amacına Göre:**

- **Paylaşımlı (Shared) IP Adresi:** Birden fazla servisin aynı IP'yi paylaşmasıdır.
- **Atanmış (Dedicated) IP Adresi:** Tek bir servis için ayrılmış IP'dir.

IP adresleri, ağ iletişiminin temel taşlarındandır.

### **Router (Yönlendirici) Nedir? Ne İşe Yarar?**

Router, bilgisayar ağları arasında veri paketlerini ileten bir ağ donanım cihazıdır. Gelen paketlerin hedefine ulaşması için en uygun yolu belirler ve yönlendirir.

#### **Temel İşlevleri:**

- **Ağları Birbirine Bağlamak:** Farklı ağları (LAN, WAN gibi) birbirine bağlar.
- **Yönlendirme (Routing):** Hedef IP adresini okuyarak yönlendirme tablosuna göre paketi iletir.
- **Veri Paketlerini Filtreleme:** Belirli kriterlere göre paketleri filtreleyebilir.
- **Ağ Adresi Çevirisi (NAT):** Özel IP'li cihazların tek genel IP üzerinden internete çıkmasını sağlar.
- **DHCP Sunucusu:** Cihazlara otomatik IP ataması yapar.
- **Güvenlik Duvarı:** Temel güvenlik özellikleri sunar.
- **Kablosuz Erişim Noktası (WAP):** Wi-Fi özelliği sunarak kablosuz cihazların bağlanmasını sağlar.
- **QoS (Quality of Service) Yönetimi:** Ağ trafiğini önceliklendirebilir.

Router, farklı ağlar arasında köprü görevi görerek veri paketlerinin verimli şekilde ulaşmasını sağlar.

### **Switch Nedir? Router ile Farkı Nedir?**

Switch (Ağ Anahtarı), bir yerel alan ağı (LAN) içinde cihazları birbirine bağlayan donanımdır. Veri paketlerini (çerçeveleri) alıp yalnızca hedeflenen cihaza iletir.

#### **Switch'in Temel İşlevleri:**

- **Cihazları Birbirine Bağlamak:** LAN üzerindeki cihazlar için fiziksel bağlantı noktası sağlar.
- **MAC Adreslerini Öğrenme ve Kullanma:** Bağlı cihazların MAC adreslerini öğrenir.

- **Hedefe Yönelik Veri İletimi (Frame Switching):** Hedef MAC adresine göre veriyi ilgili porta yönlendirir.
- **Çarpışma Alanlarını Azaltma:** Her port ayrı bir çarpışma alanıdır.
- **Yayın Alanları Oluşturma:** Genellikle tek bir yayın alanı oluşturur.

#### Router ile Switch Arasındaki Farklar:

Özellik	Switch (Ağ Anahtarı)	Router (Yönlendirici)
OSI Katmanı	Genellikle Katman 2 (Veri Bağlantı)	Katman 3 (Ağ)
Temel İşlevi	LAN içi cihazları bağlama ve trafiği yönetme	Farklı ağları bağlama ve paketleri yönlendirme
Adresleme Türü	MAC adresleri	IP adresleri
Çalışma Alanı	Genellikle tek bir ağ içinde (LAN)	Birden fazla ağ arasında (LAN'lar arası, LAN-WAN)
Yönlendirme	Genellikle yapmaz (Layer 2)	Temel görevi yönlendirmedir
Veri Birimi	Çerçeveler (Frames)	Paketler (Packets)

Switch, aynı yerel ağdaki cihazları bağlarken, Router farklı ağları birbirine bağlar ve yönlendirir.

#### Server (Sunucu) Nedir?

Sunucu, ağlarda diğer bilgisayarlara veya kullanıcılara (istemcilere) çeşitli hizmetler sunan özel bir bilgisayar veya yazılımdır. İstemcilerden gelen istekleri karşılar. Genellikle güçlü donanıma sahiptir ve kesintisiz çalışır.

#### Sunucu Çeşitleri (Amaçlarına Göre):

- **Web Sunucuları:** Web sitelerinin dosyalarını depolar ve sunar.
- **Dosya Sunucuları:** Dosya depolama, paylaşma ve erişime olanak tanır.
- **Veritabanı Sunucuları:** Veritabanlarını barındırır ve yönetir.
- **E-posta Sunucuları:** E-posta iletişimini yönetir.
- **Uygulama Sunucuları:** Belirli uygulamaları çalıştırır ve istemcilere erişim sağlar.
- **Oyun Sunucuları:** Çevrimiçi oyunlar için ortam sağlar.

- **Yazdırma Sunucuları:** Ağ üzerindeki yazıcıları yönetir.

Sunucular fiziksel, sanal veya bulut tabanlı olabilir. Modern bilgi teknolojilerinin temel taşlarıdır.

### Client (İstemci) Nedir?

İstemci, ağlarda sunuculardan hizmet talep eden ve kullanan bilgisayar, cihaz veya yazılımdır. Genellikle kullanıcıların etkileşimde bulunduğu arayüzü sağlar.

İstemci-sunucu modelinde, istemci talep eden taraftır. Örneğin, web tarayıcısı bir web sunucusundan sayfa talep eden istemcidir.

### İstemci Çeşitleri:

- **Cihazlar:** Bilgisayarlar, akıllı telefonlar vb..
- **Yazılımlar:** Web tarayıcıları, e-posta programları vb..

İstemcilerin temel görevi, sunucularla iletişim kurarak bilgi almak, veri göndermek veya işlem yapmak için isteklerde bulunmaktır.

### Port Nedir? Hangi Portlar Ne İçin Kullanılır?

Port, bir ağdaki cihaz üzerinde çalışan belirli bir uygulama veya hizmeti tanımlayan yazılımsal bir numardır. Fiziksel bir bağlantı noktası değildir. Gelen verinin hangi uygulamaya gideceğini belirler.

Port numaraları 0 ile 65535 arasında değişir ve kategorilere ayrılır:

- **İyi Bilinen Portlar (0-1023):** Yaygın hizmetler için ayrılmıştır.
- **Kayıtlı Portlar (1024-49151):** Belirli uygulamalar için tescil edilebilir.
- **Dinamik/Özel Portlar (49152-65535):** İstemci uygulamalar tarafından geçici iletişim için atanır.

### Yaygın Kullanılan Portlar:

- **20 & 21 (TCP):** FTP (Dosya Transfer Protokolü)
- **22 (TCP):** SSH (Güvenli Uzaktan Erişim)
- **23 (TCP):** Telnet (Şifresiz Uzaktan Erişim)
- **25 (TCP):** SMTP (E-posta Gönderme)
- **53 (TCP/UDP):** DNS (Alan Adı Çözümleme)
- **80 (TCP):** HTTP (Şifresiz Web Erişimi)
- **110 (TCP):** POP3 (E-posta İndirme)

- **143 (TCP):** IMAP (E-posta Yönetimi)
- **443 (TCP):** HTTPS (Güvenli Web Erişimi)
- **3389 (TCP):** RDP (Uzak Masaüstü)

Güvenlik için gereksiz portların kapalı tutulması önemlidir.

### Socket (Soket) Nedir? Hangi Amaçla Kullanılır?

Ağ iletişimde Soket, farklı cihazlardaki veya aynı cihaz üzerindeki süreçler arasında veri alışverişi için kullanılan bir iletişim bitiş noktasıdır. Bir IP adresi ve port numarasının birleşimidir.

#### Kullanım Amaçları:

- **İstemci-Sunucu Uygulamaları:** Çoğu internet uygulamasının temelidir.
- **Gerçek Zamanlı Uygulamalar:** Oyunlar, mesajlaşma, video konferans gibi.
- **Dağıtık Sistemler:** Bileşenler arası iletişimi sağlar.
- **Özel Ağ Protokolleri:** İhtiyaca özel iletişim kuralları için kullanılır.

Soketler, ağdaki cihazlar ve uygulamalar arasında kapılar görevi görerek verinin doğru yere ulaşmasını sağlar.

### Request (İstek) Nedir?

İstemci-sunucu iletişimde İstek, bir istemcinin (tarayıcı, mobil uygulama vb.) sunucuya belirli bir eylemi gerçekleştirmesi veya kaynak sağlaması için gönderdiği mesajdır. İletişim döngüsünün ilk adımıdır.

#### Bir HTTP İsteği Temel Bileşenleri:

- **İstek Metodu:** İstemcinin beklediği eylemi belirtir (GET, POST, PUT, DELETE vb.).
- **URL:** Erişilmek istenen kaynağın adresidir.
- **Başlıklar (Headers):** İstek hakkında ek bilgiler içerir.
- **Gövde (Body):** Sunucuya gönderilen asıl veri içeriğini taşır (özellikle POST, PUT, PATCH'de).

İstekler, istemcinin sunucu ile etkileşim kurabilmesi için temel rol oynar.

### Response (Yanıt) Nedir?

Yanıt, bir sunucunun istemci tarafından gönderilmiş bir isteğe karşılık olarak gönderdiği mesajdır. Sunucunun isteği işledikten sonraki sonucunu veya talep edilen kaynağı içerir.

#### Bir HTTP Yanıtı Ana Bileşenleri:

- **Durum Satırı:** HTTP versiyonu, durum kodu ve açıklaması (örn: 200 OK, 404 Not Found).
- **Başlıklar (Headers):** Yanıt hakkında ek bilgiler içerir.
- **Gövde (Body):** Yanıtın ana içeriğidir (talep edilen kaynak veya işlem sonucu).

Yanıt, istemcinin isteğe karşılık sunucudan aldığı geri bildirimdir.

### FTP (File Transfer Protocol) Nedir? Ne İşe Yarar?

FTP, ağ üzerinden istemci ve sunucu arasında dosya transferi için kullanılan standart protokoldür.

#### Ne İşe Yarar:

- **Dosya Yükleme (Upload):** Yerel dosyaları sunucuya göndermek için kullanılır.
- **Dosya İndirme (Download):** Sunucudaki dosyalara erişmek ve indirmek için kullanılır.
- **Dosya Yönetimi:** Sunucu üzerindeki dosyalar üzerinde temel işlemler yapılabilir.

FTP, genellikle iki ayrı bağlantı kullanır: kontrol (Port 21) ve veri bağlantısı. Standart FTP veriyi şifrelemez, bu nedenle hassas bilgiler için FTPS veya SFTP gibi daha güvenli alternatifler tercih edilir.

## Alan Adı Sistemleri ve İletişim

### DNS (Domain Name System) Nedir?

DNS, internete bağlı kaynakların isimlerini (alan adlarını) IP adreslerine çeviren hiyerarşik ve dağıtılmış bir isimlendirme sistemidir. İnternetin "telefon rehberi" gibidir.

#### Ne İşe Yarar:

- **Alan Adı Çözümleme:** Alan adını (örn: [www.google.com](http://www.google.com)) karşılık gelen IP adresine çevirir.
- **İnternette Gezinmeyi Kolaylaştırma:** Kullanıcıların karmaşık IP adresleri yerine alan adları kullanmasını sağlar.
- **E-posta Yönlendirme:** E-posta sunucusunun IP adresini bulmaya yardımcı olur.
- **Yük Dengeleme ve Yönlendirme:** Trafiği farklı sunuculara yönlendirebilir.

DNS, internetin düzenli ve kullanılabilir olmasını sağlayan hayati bir servistir.

### Domain (Alan Adı) Nedir?

Domain, internet üzerindeki bir kaynağı tanımlayan, insanlar tarafından okunabilir adreslerdir. IP adreslerinin aksine kelimelerden oluşur.

### Ne İŖe Yarar:

- **EriŖim Kolaylıđı:** Kullanıcıların hatırlanması kolay isimler kullanarak kaynaklara ulaŖmasını sađlar.
- **MarkalaŖma ve Kimlik:** Çevrimiçi varlık için benzersiz bir kimlik oluŖturur.
- **DNS ile İliŖki:** DNS ile birlikte çalışarak alan adını IP adresine çevirir.
- **E-posta Adresleri:** E-posta adreslerinde kullanılır (kullaniciadi@firmaadi.com).

Alan adları hiyerarŖik bir yapıya sahiptir (TLD, SLD, subdomain).

### DHCP (Dynamic Host Configuration Protocol) Nedir?

DHCP, ađdaki cihazlara otomatik olarak IP adresleri ve diđer ađ yapılandırma bilgilerini atamak için kullanılan bir ađ yönetim protokolüdür.

### Ne İŖe Yarar:

- **Otomatik IP Ataması:** Cihazlar ađa bađlandığında otomatik IP alırlar.
- **IP Adresi Yönetiminin KolaylaŖması:** IP adresleri merkezi havuzdan yönetilir.
- **Ađ Yapılandırma Hatalarının Azalması:** Manuel hataları önler.
- **Cihaz Hareketliliđi:** Farklı ađlara bađlanan cihazlar kolayca IP alabilir.
- **Zaman ve Maliyet Tasarrufu:** Ađ yöneticilerinin iŖ yükünü azaltır.

DHCP, ađdaki cihazların hızlı ve hatasız bađlanması için kritik bir protokoldür.

### WWW (World Wide Web) Nedir?

WWW, internet üzerinde çalışan ve birbirine bađlı hipermetin belgelerinden oluŖan bir bilgi sistemidir. İnternetin kendisi üzerinde çalışan bir servistir.

### Ne İŖe Yarar:

- **Bilgiye EriŖim:** Kullanıcıların web sitelerine ve sayfalara eriŖmesini sađlar.
- **Hipermetin Bađlantıları:** Sayfalar arası geçiŖi sađlayan linkler içerir.
- **Web Siteleri:** Birbiriyle ilişkili web sayfalarının bir araya geldiđi alanları oluŖturur.
- **Evrensel Bilgi PaylaŖımı:** Dil veya platformdan bađımsız bilgiye eriŖim ve paylaŖım sađlar.

WWW, bilgiye eriŖimi ve gezinmeyi kolaylaŖtıran, internetin küresel bilgi ađı olarak kullanılmasını sađlayan bir sistemdir.

### A Kaydı (Address Record) DNS'te Ne Anlama Gelir?



DNS içerisinde A Kaydı, bir alan adını veya alt alan adını doğrudan bir IPv4 adresiyle eşleyen temel kayıt türüdür. "A" harfi "Address" kelimesini temsil eder.

### **Ne Anlama Gelir ve Ne İşe Yarar:**

Bir alan adının hangi IPv4 adresine karşılık geldiğini DNS sistemine bildirir. Kullanıcılar bir alan adı yazdığında, DNS bu alan adının A kaydını sorgular ve ilgili IPv4 adresini döndürür.

A kaydı:

- Bir alan adını bir IPv4 adresiyle ilişkilendirir.
- Kullanıcıların alan adları aracılığıyla kaynaklara erişmesini sağlar.
- DNS çözümleme sürecinin temelini oluşturur.

Bir alan adının çalışabilmesi için genellikle en az bir A kaydı olmalıdır. IPv6 adresleri için AAAA kaydı kullanılır.

## **Veri Tabanı Temelleri**

### **NoSQL Veri Tabanı Nedir? Özellikleri Nelerdir?**

NoSQL Veritabanı, geleneksel ilişkisel (SQL) veritabanlarından farklı olarak verileri tablo ve ilişkisel model dışında depolayan ve yöneten veritabanı türleridir. "Not only SQL" veya "Non-relational" olarak adlandırılırlar. Büyük hacimli, hızla değişen verilere esnek ve ölçeklenebilir çözümler sunar.

#### **Özellikleri:**

- **Esnek Şema:** Önceden tanımlanmış katı bir şema gerektirmez.
- **Yatay Ölçeklenebilirlik:** Sisteme ek sunucular ekleyerek kolayca genişleyebilir.
- **Çeşitli Veri Modelleri:** Tek bir modele bağlı değildir, farklı modeller sunar: Anahtar-Değer, Belge Tabanlı, Sütun Ailesi Tabanlı, Graf Tabanlı.
- **Dağıtılmış Hesaplama ve Veri Dağıtımı:** Verileri birden fazla sunucuya dağıtarak yüksek erişilebilirlik sağlar.
- **Nihai Tutarlılık:** Anlık tutarlılık yerine zamanla tutarlı hale gelmeyi tercih eder.
- **Yüksek Performans ve Erişilebilirlik:** Belirli iş yüklerinde yüksek performans sunar.

NoSQL veritabanları, büyük veri analizi, gerçek zamanlı web uygulamaları gibi alanlarda tercih edilir.

### **SQL Server Veri Tipleri Nelerdir?**

SQL Server, veritabanında depolanacak farklı türdeki bilgileri tanımlamak için çeşitli veri tipleri sunar.

### Yaygın Veri Tipleri ve Örnekleri:

- **Tam Sayısal Veri Tipleri:** Tam sayıları depolar (TINYINT, SMALLINT, INT, BIGINT). DECIMAL/NUMERIC (sabit ondalık), MONEY/SMALLMONEY (para birimi), BIT (Boolean).
- **Yaklaşık Sayısal Veri Tipleri:** Ondalık sayıları yaklaşık değerleriyle depolar (REAL, FLOAT).
- **Tarih ve Saat Veri Tipleri:** Tarih ve saat bilgilerini saklar (DATE, TIME, DATETIME, SMALLDATETIME, DATETIME2, DATETIMEOFFSET).
- **Karakter Dizisi Veri Tipleri:** Metin verilerini saklar (CHAR, VARCHAR, VARCHAR(MAX)). TEXT eski kullanımdır.
- **Unicode Karakter Dizisi Veri Tipleri:** Farklı dillerdeki karakterleri saklar (NCHAR, NVARCHAR, NVARCHAR(MAX)). NTEXT eski kullanımdır.
- **İkili Veri Tipleri:** İkili veriyi saklar (BINARY, VARBINARY, VARBINARY(MAX)). IMAGE eski kullanımdır.
- **Diğer Veri Tipleri:** Özel amaçlı tipler (UNIQUEIDENTIFIER (GUID), XML, GEOMETRY/GEOGRAPHY (mekansal), ROWVERSION (satır sürümü)).

### Örnekler:

#### 1. Tam Sayısal Veri Tipleri (Exact Numerics)

Tam sayıları depolamak için kullanılır. Farklı boyutlarda ve değer aralıklarında seçenekler sunarlar.

- **TINYINT:** 0 ile 255 arasındaki tam sayıları saklar (1 byte).

SQL

```
CREATE TABLE Urunler (  
    UrunID INT PRIMARY KEY,  
    StokAdedi TINYINT  
);
```

```
INSERT INTO Urunler (UrunID, StokAdedi) VALUES (1, 50);
```

- **SMALLINT:** -32,768 ile 32,767 arasındaki tam sayıları saklar (2 bytes).

SQL

```
CREATE TABLE Siparisler (  
    SiparisID INT PRIMARY KEY,  
    MusteriSayisi SMALLINT  
);
```

```
INSERT INTO Siparisler (SiparisID, MusteriSayisi) VALUES (101, 150);
```

- **INT:** -2,147,483,648 ile 2,147,483,647 arasındaki tam sayıları saklar (4 bytes). Genellikle en yaygın tam sayı tipidir.

SQL

```
CREATE TABLE Kullanicilar (  
    KullaniciID INT PRIMARY KEY,  
    Yas INT  
);
```

```
INSERT INTO Kullanicilar (KullaniciID, Yas) VALUES (1001, 30);
```

- **BIGINT:** Çok büyük veya çok küçük tam sayıları saklar (-9,223,372,036,854,775,808 ile 9,223,372,036,854,775,807 arası) (8 bytes).

SQL

```
CREATE TABLE BuyukSayilar (  
    ID INT PRIMARY KEY,  
    Nufus BIGINT  
);
```

```
INSERT INTO BuyukSayilar (ID, Nufus) VALUES (1, 800000000);
```

- **DECIMAL / NUMERIC:** Sabit ondalık sayıları saklar. Duyarlılık (precision) ve ölçek (scale) belirtilebilir. Para birimleri gibi hassas değerler için idealdir.

SQL

```
CREATE TABLE Finans (
```

IslemID INT PRIMARY KEY,  
Tutar DECIMAL(10, 2) -- Toplam 10 basamak, virgülden sonra 2 basamak  
);

INSERT INTO Finans (IslemID, Tutar) VALUES (201, 1234.56);

- **MONEY / SMALLMONEY:** Para birimi değerlerini saklar. MONEY daha geniş bir aralığa sahiptir.

SQL

CREATE TABLE Satislar (  
SatisID INT PRIMARY KEY,  
Fiyat MONEY  
);

INSERT INTO Satislar (SatisID, Fiyat) VALUES (301, 199.99);

- **BIT:** Boolean (doğru/yanlış) değerleri saklar (0, 1 veya NULL).

SQL

CREATE TABLE Ayarlar (  
AyarID INT PRIMARY KEY,  
Aktif BIT  
);

INSERT INTO Ayarlar (AyarID, Aktif) VALUES (1, 1); -- 1 doğruyu temsil eder

## 2. Yaklaşık Sayısal Veri Tipleri (Approximate Numerics)

Ondalık sayıları yaklaşık değerleriyle saklar. Bilimsel hesaplamalar veya çok büyük/küçük kayan noktalı sayılar için kullanılır, ancak hassasiyet garantisi vermez.

- **REAL:** Tek duyarlıklı kayan noktalı sayılar (4 bytes).
- **FLOAT:** Çift duyarlıklı kayan noktalı sayılar (8 bytes). Duyarlılık parantez içinde belirtilebilir (float(n)), bu da depolama boyutunu etkileyebilir.

SQL

```
CREATE TABLE Olcumler (  
    OlcumID INT PRIMARY KEY,  
    Deger FLOAT  
);
```

```
INSERT INTO Olcumler (OlcumID, Deger) VALUES (401, 3.14159);
```

### 3. Tarih ve Saat Veri Tipleri (Date and Time)

Tarih ve saat bilgilerini saklamak için kullanılır.

- **DATE:** Yalnızca tarih bilgisini saklar (YYYY-MM-DD) (3 bytes).

SQL

```
CREATE TABLE Etkinlikler (  
    EtkinlikID INT PRIMARY KEY,  
    EtkinlikTarihi DATE  
);
```

```
INSERT INTO Etkinlikler (EtkinlikID, EtkinlikTarihi) VALUES (501, '2025-12-31');
```

- **TIME:** Yalnızca saat bilgisini saklar (HH:MM:SS[.kesir]) (3-5 bytes).

SQL

```
CREATE TABLE Toplantilar (  
    ToplantID INT PRIMARY KEY,  
    ToplantSaati TIME(7) -- Saniyenin kesir kısmı için duyarlılık  
);
```

```
INSERT INTO Toplantilar (ToplantID, ToplantSaati) VALUES (601,  
'14:30:00.1234567');
```

- **DATETIME:** Tarih ve saat bilgisini saklar (1753-01-01 ile 9999-12-31 arası) (8 bytes). Daha düşük hassasiyete sahiptir.

SQL

```
CREATE TABLE Kayitlar (
```

```
KayitID INT PRIMARY KEY,  
KayitZamani DATETIME  
);
```

```
INSERT INTO Kayitlar (KayitID, KayitZamani) VALUES (701, '2025-05-08 19:00:00');
```

- **SMALLDATETIME:** Daha küçük bir tarih ve saat aralığını saklar (1900-01-01 ile 2079-06-06 arası) (4 bytes). Hassasiyeti dakikadır.

SQL

```
CREATE TABLE Loglar (  
LogID INT PRIMARY KEY,  
LogZamani SMALLDATETIME  
);
```

```
INSERT INTO Loglar (LogID, LogZamani) VALUES (801, '2025-05-08 19:05');
```

- **DATETIME2:** Daha geniş bir tarih ve saat aralığını ve daha yüksek hassasiyeti saklar (0001-01-01 00:00:00.0000000 ile 9999-12-31 23:59:59.9999999 arası) (6-8 bytes).

SQL

```
CREATE TABLE Islemler (  
IslemLogID INT PRIMARY KEY,  
IslemZamani DATETIME2(7)  
);
```

```
INSERT INTO Islemler (IslemLogID, IslemZamani) VALUES (901, '2025-05-08 19:06:19.1234567');
```

- **DATETIMEOFFSET:** Tarih, saat ve saat dilimi offset bilgisini saklar (8-10 bytes).

SQL

```
CREATE TABLE GlobalEtkinlikler (  
EtkinlikID INT PRIMARY KEY,
```

EtkinlikZamani DATETIMEOFFSET(7)

);

```
INSERT INTO GlobalEtkinlikler (EtkinlikID, EtkinlikZamani) VALUES (1001, '2025-05-08 19:06:19.1234567 +03:00');
```

#### 4. Karakter Dizisi Veri Tipleri (Character Strings)

ASCII karakterlerini içeren metin verilerini saklar.

- **CHAR(n):** Belirtilen uzunlukta (n) sabit uzunluklu karakter dizisi saklar (1 ile 8000 arası). Belirtilen uzunluktan kısa veri girilirse boşluklarla doldurulur.

SQL

```
CREATE TABLE Kodu (
```

```
    UlkeKodu CHAR(2) PRIMARY KEY
```

```
);
```

```
INSERT INTO Kodu (UlkeKodu) VALUES ('TR');
```

- **VARCHAR(n):** Belirtilen maksimum uzunlukta (n) değişken uzunluklu karakter dizisi saklar (1 ile 8000 arası). Depolama boyutu girilen verinin uzunluğuna göre değişir.

SQL

```
CREATE TABLE KullaniciBilgileri (
```

```
    KullaniciID INT PRIMARY KEY,
```

```
    Ad VARCHAR(50)
```

```
);
```

```
INSERT INTO KullaniciBilgileri (KullaniciID, Ad) VALUES (1, 'Ahmet');
```

- **VARCHAR(MAX):** 2<sup>31</sup>-1 byte'a kadar değişken uzunluklu karakter dizisi saklar. Büyük metin blokları için kullanılır (örn: makale içerikleri).

SQL

```
CREATE TABLE Makaleler (
```

```
    MakaleID INT PRIMARY KEY,
```

```
Icerik VARCHAR(MAX)  
);
```

```
INSERT INTO Makaleler (MakaleID, Icerik) VALUES (1, 'Bu çok uzun bir makale içeriği...');
```

- **TEXT:** VARCHAR(MAX)'in eski karşılığıdır ve ileriki sürümlerde kaldırılması planlanmaktadır. Yeni geliştirmelerde kullanılması önerilmez.

## 5. Unicode Karakter Dizisi Veri Tipleri (Unicode Character Strings)

Farklı dillerdeki karakterleri (Unicode) içeren metin verilerini saklar. Her karakter genellikle 2 byte yer kaplar.

- **NCHAR(n):** Belirtilen uzunlukta (n) sabit uzunluklu Unicode karakter dizisi saklar (1 ile 4000 arası).

SQL

```
CREATE TABLE Diller (  
    DilKodu NCHAR(2) PRIMARY KEY  
);
```

```
INSERT INTO Diller (DilKodu) VALUES (N'TR'); -- Unicode sabitleri için 'N' ön eki kullanılır
```

- **NVARCHAR(n):** Belirtilen maksimum uzunlukta (n) değişken uzunluklu Unicode karakter dizisi saklar (1 ile 4000 arası).

SQL

```
CREATE TABLE Ceviriler (  
    CeviriID INT PRIMARY KEY,  
    Metin NVARCHAR(100)  
);
```

```
INSERT INTO Ceviriler (CeviriID, Metin) VALUES (1, N'Merhaba Dünya');
```

- **NVARCHAR(MAX):**  $2^{31}-1$  byte'a kadar değişken uzunluklu Unicode karakter dizisi saklar. Büyük Unicode metin blokları için kullanılır.



SQL

```
CREATE TABLE Kitaplar (  
    KitapID INT PRIMARY KEY,  
    Ozet NVARCHAR(MAX)  
);
```

INSERT INTO Kitaplar (KitapID, Ozet) VALUES (1, N'Bu kitabın uzun özeti...');

- **NTEXT:** NVARCHAR(MAX)'in eski karşılığıdır ve ileriki sürümlerde kaldırılması planlanmaktadır. Yeni geliştirmelerde kullanılması önerilmez.

## 6. İkili Veri Tipleri (Binary Strings)

İkili veriyi (bayt dizileri) saklamak için kullanılır. Resimler, belgeler veya diğer dosya içerikleri için kullanılabilir.

- **BINARY(n):** Belirtilen uzunlukta (n) sabit uzunluklu ikili veri saklar (1 ile 8000 arası).
- **VARBINARY(n):** Belirtilen maksimum uzunlukta (n) değişken uzunluklu ikili veri saklar (1 ile 8000 arası).

SQL

```
CREATE TABLE Dosyalar (  
    DosyaID INT PRIMARY KEY,  
    Dosyaicerigi VARBINARY(MAX) -- Büyük ikili veriler için MAX kullanılır  
);
```

-- Dosya içeriği genellikle uygulama tarafından ikili veri olarak sağlanır.

- **IMAGE:** VARBINARY(MAX)'in eski karşılığıdır ve ileriki sürümlerde kaldırılması planlanmaktadır. Yeni geliştirmelerde kullanılması önerilmez.

## 7. Diğer Veri Tipleri

Yukarıdaki kategorilere girmeyen özel amaçlı veri tipleri.

- **UNIQUEIDENTIFIER:** Küresel olarak benzersiz tanımlayıcı (GUID) saklar (16 bytes).

SQL

```
CREATE TABLE KayitDefteri (  
    LogID UNIQUEIDENTIFIER PRIMARY KEY DEFAULT NEWID(), -- Otomatik GUID  
    olusturur  
    IslemAciklama VARCHAR(100)  
);
```

```
INSERT INTO KayitDefteri (IslemAciklama) VALUES ('Kullanıcı girişi');
```

- **XML:** XML verilerini saklar.

SQL

```
CREATE TABLE UrunDetay (  
    UrunID INT PRIMARY KEY,  
    Detay XML  
);
```

```
INSERT INTO UrunDetay (UrunID, Detay) VALUES (1,  
'<urun><renk>Kırmızı</renk><beden>M</beden></urun>');
```

- **GEOMETRY / GEOGRAPHY:** Mekansal (konumsal) veri saklar. GEOMETRY düzlem koordinat sistemlerini, GEOGRAPHY ise yuvarlak dünya koordinat sistemlerini (enlem/boylam) temsil eder.

SQL

```
CREATE TABLE Lokasyonlar (  
    LokasyonID INT PRIMARY KEY,  
    Konum GEOGRAPHY  
);
```

-- Örnek ekleme (enlem ve boylam ile bir nokta oluşturma)

```
-- INSERT INTO Lokasyonlar (LokasyonID, Konum) VALUES (1,  
geography::Point(41.0082, 28.9784, 4326)); -- Örnek WKT formatı ve SRID
```

- **ROWVERSION:** Bir tablodaki satırların sürümünü belirten otomatik olarak oluşturulan ikili sayıdır. Genellikle eşzamanlılık kontrolü için kullanılır. Eski adı TIMESTAMP'tir, ancak tarih/saat ile ilgisi yoktur.

SQL

CREATE TABLE Envanter (

UrunID INT PRIMARY KEY,

UrunAdi VARCHAR(50),

SonGuncelleme ROWVERSION -- Otomatik olarak güncellenir

);

### CRUD İşlemleri Nedir? (Create, Read, Update, Delete)

CRUD, kalıcı depolama üzerindeki dört temel işlemi temsil eden bir kısaltmadır.

#### CRUD İşlemleri:

1. **Create (Oluşturma):** Veritabanına yeni veri eklemek için kullanılır. SQL'de INSERT komutu ile yapılır. Örnek: Yeni kullanıcı kaydı oluşturma.
2. **Read (Okuma / Listeleme):** Mevcut veriyi almak veya görüntülemek için kullanılır. SQL'de SELECT komutu ile yapılır. Örnek: Ürün listesi görüntüleme.
3. **Update (Güncelleme):** Mevcut veriyi değiştirmek için kullanılır. SQL'de UPDATE komutu ile yapılır. Örnek: Kullanıcı bilgisi güncelleme.
4. **Delete (Silme):** Mevcut veriyi silmek için kullanılır. SQL'de DELETE komutu ile yapılır. Örnek: Kullanıcı hesabı silme.

CRUD işlemleri, bir uygulamanın verilerle etkileşimde bulunabilmesi için temel yapı taşlarıdır.

### Foreign Key Nedir? Nasıl Kullanılır?

Foreign Key (Yabancı Anahtar), bir tablodaki sütunun, başka bir tablodaki birincil veya benzersiz anahtara referans veren kısıtlamadır.

#### Ne İşe Yarar:

- **Tablolar Arası İlişki Kurulur:** İki tablo arasında mantıksal bağlantı oluşturur.
- **Veri Tutarlılığı Sağlanır:** Alt tablodaki değerlerin ana tabloda var olmasını zorunlu kılar.

- **"Yetim" Kayıtlar Önlenir:** Ana tablodan kayıt silindiğinde/güncellendiğinde ilgili alt tablo kayıtlarının ne olacağı belirlenir.

### Nasıl Kullanılır:

Tablo oluşturulurken (CREATE TABLE) veya değiştirilirken (ALTER TABLE) tanımlanır.

**ON UPDATE ve ON DELETE Eylemleri:** Ana tablodaki kayıt güncellendiğinde/silindiğinde alt tablodaki kayıtlar için eylemler belirlenebilir (NO ACTION, CASCADE, SET NULL, SET DEFAULT).

Foreign Key'ler, ilişkisel veritabanı tasarımının önemli bir parçasıdır ve veri bütünlüğünü sağlar.

### Join İşlemleri Nedir? Çeşitleri Nelerdir? (Inner Join, Left Join, vb.)

JOIN işlemleri, aralarında ilgili bir sütun bulunan iki veya daha fazla tablodaki satırları birleştirerek tek bir sonuç kümesi elde etmek için kullanılır.

### JOIN Çeşitleri:

1. **INNER JOIN:** Birleştirme koşulunu sağlayan eşleşen satırları getirir.
2. **LEFT JOIN (Sol Dış Birleştirme):** Sol tablodaki tüm satırları ve sağ tablodan eşleşenleri getirir. Eşleşmeyenler için sağ sütunlar NULL döner.
3. **RIGHT JOIN (Sağ Dış Birleştirme):** Sağ tablodaki tüm satırları ve sol tablodan eşleşenleri getirir. Eşleşmeyenler için sol sütunlar NULL döner.
4. **FULL OUTER JOIN (Tam Dış Birleştirme):** Her iki tablodaki tüm satırları getirir, eşleşenler birleştirilir, eşleşmeyenler için ilgili sütunlar NULL döner.
5. **CROSS JOIN (Çapraz Birleştirme):** İki tablonun tüm satırlarının Kartezyen Çarpımını döndürür, koşul kullanılmaz.

Farklı JOIN türleri, ilişkili verileri sorgularken farklı ihtiyaçlara yönelik esneklik sağlar.