

AĞ (NETWORK) TEMELLERİ

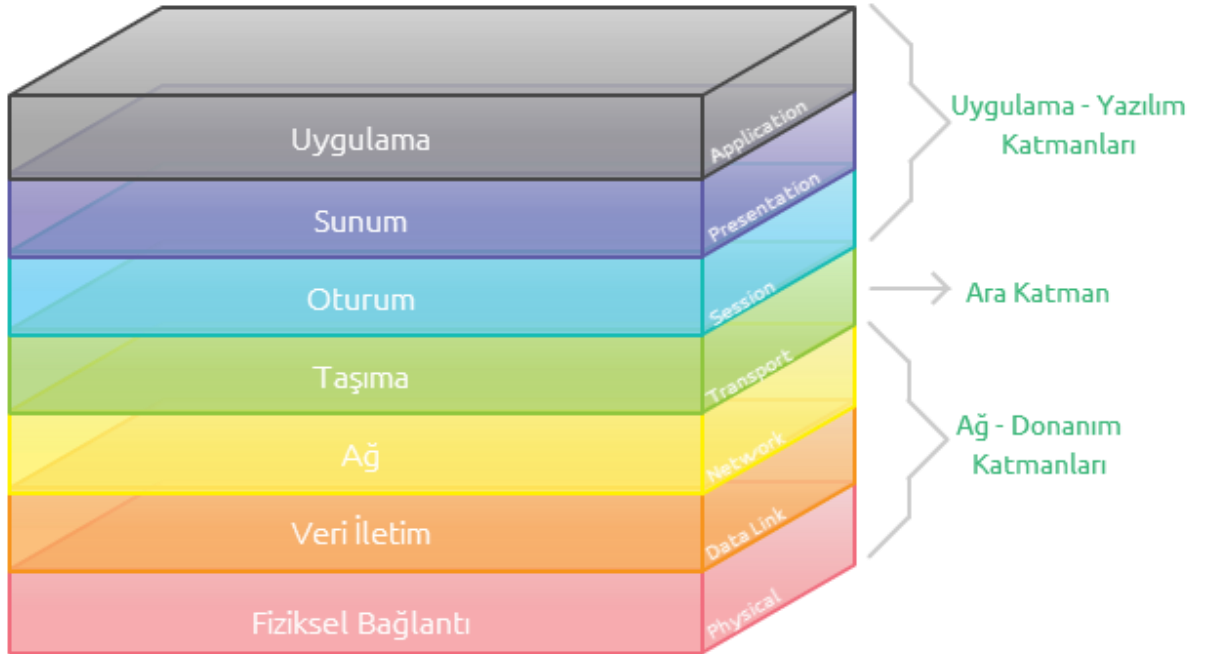
TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol), bilgisayarlar arasında iletişimi sağlayan bir dizi ağ protokolüdür. Ağ protokolleri, ağ iletişimini yöneten kurallar veya standartlardır. Bir ağdaki iki cihazın birlikte iletişim kurması gerekiyorsa, ortak ağ protokolleri setini kullanmaları gerekir.

TCP/IP Özellikleri :

- **Katmanlı Yapı:** Uygulama, Taşıma, İnternet ve Ağ Erişim katmanlarından oluşur.
- **Standartlaştırılmıştır:** Farklı donanım ve sistemlerde çalışabilir.
- **Adresleme:** IP adresiyle cihazlar tanımlanır.
- **Güvenilirlik (TCP):** Veriler sıralı ve hatasız iletilir.
- **Yönlendirme (IP):** Veriler en uygun yol üzerinden iletilir.
- **Bağlantı türü:** TCP bağlantı tabanlıdır; IP bağlantısızdır.

OSI Katmanları



OSI Katmanları (Open Systems Interconnection Model), bilgisayar ağlarındaki iletişimi 7 katmana ayırarak açıklar. Her katmanın kendine özgü görevi vardır ve bu yapı sayesinde sistemler arası veri iletişimi daha anlaşılır ve standart hale gelir.

1. Fiziksel Katman (Physical Layer):

Görevi: Verinin elektriksel, mekanik veya optik sinyallere çevrilip fiziksel ortamda iletilmesini sağlar.

✦ Kablo, priz, voltaj seviyesi gibi donanımlar bu katmandadır.

2. Veri Bağlantı Katmanı (Data Link Layer):

Görevi: Aynı ağ içindeki cihazlar arasında **hatasız veri iletimi** sağlar. Çerçeve (frame) oluşturur ve MAC adresleriyle çalışır.

✦ Anahtarlar (switch), MAC adresi, hata kontrolü bu katmandadır.

3. Ağ Katmanı (Network Layer):

Görevi: Verinin farklı ağlar arasında yönlendirilmesini sağlar. **IP adresi** bu katmanda kullanılır.

✦ Yönlendiriciler (router), IP adresleme, paket yönlendirme burada işler.

4. Taşıma Katmanı (Transport Layer):

Görevi: Uçtan uca bağlantı kurar ve verinin **güvenli, sıralı** iletimini sağlar. TCP ve UDP bu katmandadır.

✦ Bağlantı kontrolü, hata düzeltme ve akış kontrolü yapar.

5. Oturum Katmanı (Session Layer):

Görevi: Uygulamalar arasında oturum (session) kurar, yönetir ve sonlandırır.

✦ Oturum senkronizasyonu ve yönetimi sağlar.

6. Sunum Katmanı (Presentation Layer):

Görevi: Veriyi sistemler arası ortak formata çevirir. **Şifreleme, sıkıştırma, veri biçimi dönüşümü** bu katmanda olur.

✦ Örneğin: JPEG ↔ Bitmap dönüşümü veya verinin şifrelenmesi.

7. Uygulama Katmanı (Application Layer):

Görevi: Kullanıcıya en yakın katmandır. Uygulamaların ağa erişmesini sağlar (web, e-posta vb.).

✦ HTTP, FTP, SMTP gibi protokoller burada yer alır.

IP Adresi Nedir?

IP Adresi (Internet Protocol Address), internet veya yerel ağ üzerindeki cihazların birbirini tanımasını ve veri alışverişi yapmasını sağlayan sayısal bir tanımlayıcıdır.

Örnek: 192.168.1.1

IP Adresi Türleri

◆ **1. Yapısına Göre IP Adresleri**

- **IPv4 (Internet Protocol version 4):**

- 32 bit uzunluğundadır.
- Nokta ile ayrılmış 4 sayıdan oluşur.
- Örnek: 192.168.0.1
- En yaygın kullanılan türdür.

- **IPv6 (Internet Protocol version 6):**

- 128 bit uzunluğundadır.
- Nokta yerine : ile ayrılmış 8 bloktan oluşur.
- Örnek: 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- IPv4 yetersiz kaldığı için geliştirilmiştir.

◆ **2. Kullanım Amacına Göre IP Adresleri**

- **Özel IP (Private IP):**

- Yerel ağlarda (ev, okul, işyeri gibi) kullanılır.
- İnternete doğrudan çıkamaz; NAT kullanılır.
- Örnek aralıklar:
 - 192.168.0.0 – 192.168.255.255
 - 10.0.0.0 – 10.255.255.255

- **Genel IP (Public IP):**

- İnternete doğrudan çıkabilen, servis sağlayıcıdan alınan IP'dir.
- Web siteleri, sunucular bu adresleri kullanır.

◆ 3. Bağlantı Türüne Göre IP Adresleri

- **Statik IP (Sabit):**

- Değişmeyen, sabit atanmış IP adresidir.
- Genellikle sunucular ve güvenlik sistemlerinde kullanılır.

- **Dinamik IP (Değişken):**

- İnternet servis sağlayıcı tarafından geçici olarak atanır.
- Ev kullanıcıları genellikle dinamik IP kullanır.

Router

Router, bilgisayar ağları arasında veri paketlerini almak, analiz etmek ve iletmek için tasarlanmış fiziksel veya sanal bir ağlar arası aygıttır. **Router** cihazları, dosyalar, iletişimler ve web etkileşimleri gibi basit aktarımlar gibi çeşitli veri türlerini içeren paketleri kullanarak ağ verilerini yönlendirmeye yarar. **Router**, izinsiz bir şekilde girilen ağa karşı ilk güvenlik hattıdır. Yönlendiricide en yüksek güvenlik düzeyini etkinleştirmek, güvenlik duvarı gibi özellikleri açar. Bilgisayar sisteminizi ve bilgilerinizi saldırılara karşı korumanın en iyi yollarından biridir.

Switch

Switch (Anahtar), bir ağ cihazıdır ve yerel ağ (LAN) içindeki bilgisayarları birbirine bağlar. Görevi, gelen veriyi yalnızca hedef cihaza iletmektir (MAC adresine göre).

- Her cihaza özel iletişim sağlar → Çakışma (collision) azaltılır.
- Veri iletiminde çerçeve (frame) kullanır.
- 2. katmanda (Veri Bağlantı Katmanı) çalışır.

Özellik	Switch	Router
Çalıştığı Katman	2. Katman (Data Link)	3. Katman (Network)
Görev	Cihazları yerel ağda bağlar	Ağlar arası veri yönlendirme yapar
Adres Türü	MAC adresi kullanır	IP adresi kullanır
Kullanım Alanı	İç ağ (LAN)	İç ağ ↔ Dış ağ (LAN ↔ WAN)
İnternete Bağlantı	Sağlayamaz	Sağlar

Server (Sunucu) Nedir?

Başka cihazlara hizmet sunan, veri sağlayan sistemdir.

- ✦ Örnek: Web sunucusu, veri tabanı sunucusu, e-posta sunucusu.
- Sürekli açık kalır ve istemcilerden gelen isteklere yanıt verir.

Client (İstemci) Nedir?

Sunucudan hizmet alan bilgisayar ya da uygulamadır.

- ✦ Örnek: Tarayıcı (Chrome), e-posta programı, mobil uygulama.

Port Nedir?

Aynı IP üzerindeki farklı hizmetlerin ayrılmasını sağlayan **sanal kapılardır**.

- ✦ IP adresi posta koduysa, port numarası daire numarası gibidir.
- ✦ Port aralığı: 0 – 65535

Sık Kullanılan Portlar:

Port Protokol Açıklama

20/21	FTP	Dosya transferi
22	SSH	Güvenli uzaktan bağlantı
25	SMTP	E-posta gönderme
53	DNS	Alan adı çözümleme
80	HTTP	Web (şifresiz)
443	HTTPS	Web (şifreli)
3306	MySQL	Veritabanı bağlantısı

Socket (Soket) Nedir?

IP adresi + Port numarasından oluşan iletişim uç noktasıdır.

- ✦ Ağ üzerindeki iki cihazın veri alışverişi yapabilmesi için kullanılır.
- ✦ Yazılım geliştirirken, istemci ve sunucu arasında bağlantı kurmak için kullanılır.

Request (İstek) Nedir?

İstemciden sunucuya gönderilen veri veya işlem talebidir.

- ✦ Örnek: Bir web sitesini açmak için tarayıcının yaptığı GET isteği.

Response (Yanıt) Nedir?

Sunucunun, istemcinin isteğine verdiği cevaptır.

✦ Örnek: Web sayfasının HTML kodları sunucudan gelen yanıtla gelir.

FTP (File Transfer Protocol) Nedir?

Dosya transferi için kullanılan ağ protokolüdür.

✦ Kullanımı:

- Web sitelerine dosya yüklemek
- Uzaktan dosya indirmek/göndermek
 - ✦ Port: 21
 - ✦ FTP istemcileri: FileZilla, WinSCP vb.

ALAN ADI SİSTEMLERİ VE İLETİŞİM

DNS (Domain Name System) nedir?

DNS, internet üzerindeki alan adlarını (örneğin: www.google.com) IP adreslerine (örneğin: 142.250.190.78) çeviren sistemdir. İnsanların kolayca hatırlayabildiği alan adlarını, bilgisayarların anlayabileceği IP adreslerine dönüştürür. Bu sayede tarayıcına alan adı yazdığında doğru sunucuya yönlendirilir.

Domain (Alan Adı) nedir?

Domain, internet üzerindeki bir web sitesinin adıdır. Kullanıcıların bir web sitesine erişmek için kullandığı adres biçimindedir (örnek: example.com). IP adreslerinin yerine daha kolay hatırlanabilir bir yöntem sunar.

DHCP (Dynamic Host Configuration Protocol) nedir?

DHCP, bir ağdaki cihazlara otomatik olarak IP adresi, alt ağ maskesi, ağ geçidi (gateway) ve DNS sunucusu gibi bilgileri atayan protokoldür. Böylece cihazlar elle ayar yapılmadan ağa bağlanabilir.

WWW (World Wide Web) nedir?

WWW, internet üzerindeki web sitelerini oluşturan sayfaların, bağlantıların ve içeriklerin bütünüdür. Web tarayıcıları aracılığıyla erişilen, HTML sayfalarından oluşan bir sistemdir. İnternetin bir hizmetidir (diğerleri e-posta, FTP gibi).

A Kaydı (Address Record) DNS'te ne anlama gelir?

A kaydı, bir alan adının hangi **IPv4** adresine karşılık geldiğini tanımlar. Örneğin example.com alan adı için bir A kaydı, bu alan adının IP adresini belirtir. Böylece DNS sunucusu, bu adresi çözümleyebilir.

Veri Tabanı Temelleri

NoSQL Veritabanları

NoSQL, büyük veri tabanı yönetim sistemlerinin genel adıdır. İlişkisel olmayan veritabanı türlerini ifade eder. Bu tablolar, verileri ilişkisel tablolardan daha farklı bir formatta depolama yapar. Büyük veri ile birlikte ölçekleme ihtiyacından doğmuştur ve klasik RDBMS'lerin (ilişkisel veritabanı yönetim sistemi) yetersiz kaldığı alanlarda tercih edilmektedir.

Özellikleri:

- Şemaya bağlı değildir (esnek veri yapısı).
- Yüksek performans sağlar.
- Dağıtık mimari ile çalışır.
- Büyük veriyi yatay olarak ölçekleyebilir.
- JSON, BSON, XML gibi formatları destekler.

Popüler NoSQL türleri ve örnekleri:

- **Belge tabanlı:** MongoDB (veriler JSON gibi saklanır)
- **Anahtar-değer:** Redis
- **Sütun tabanlı:** Cassandra
- **Graf:** Neo4j

SQL Server Veri Tipleri

1. Sayısal (Numerik) Veri Tipleri

1. INT

- Açıklama: Tamsayı veri tipi (± 2 milyar arası).
- Örnek: INT Yas = 25

2. SMALLINT

- Açıklama: Daha küçük tamsayı aralığı ($\pm 32,767$).
- Örnek: SMALLINT Seviye = 1200

3. TINYINT

- Açıklama: 0 ile 255 arası pozitif sayı.
- Örnek: TINYINT Puan = 100

4. BIGINT

- Açıklama: Çok büyük tamsayılar (8 bayt).
- Örnek: BIGINT ToplamTıklama = 9876543210

5. DECIMAL(p,s) veya NUMERIC(p,s)

- Açıklama: Kesin ondalıklı sayılar. p: Toplam basamak, s: virgülden sonra.
- Örnek: DECIMAL(6,2) Fiyat = 1299.99

6. FLOAT

- Açıklama: Yaklaşık ondalıklı sayı, daha fazla hassasiyet.
- Örnek: FLOAT Sıcaklık = 36.6

7. REAL

- Açıklama: Daha az hassas float türü.
- Örnek: REAL Oran = 0.75

8. BIT

- Açıklama: 0 veya 1 tutar (true/false için).
- Örnek: BIT AktifMi = 1



2. Metin (Text) Veri Tipleri

1. CHAR(n)

- Açıklama: Sabit uzunlukta karakter dizisi. Her zaman n karakter yer kaplar.
- Örnek: CHAR(5) Kod = 'A12 '

2. VARCHAR(n)

- Açıklama: Değişken uzunlukta metin. Ne kadar yazarsan o kadar yer kaplar.
- Örnek: VARCHAR(50) Ad = 'Ahmet'

3. TEXT (eski)

- Açıklama: Çok uzun metin. Artık VARCHAR(MAX) önerilir.
- Örnek: TEXT Açıklama = 'Ürün açıklaması...'

4. NCHAR(n)

- Açıklama: Unicode sabit uzunlukta metin. Türkçe, Arapça vs. diller için.
- Örnek: NCHAR(10) = N'İstanbul'

5. NVARCHAR(n)

- Açıklama: Unicode değişken uzunlukta metin.
- Örnek: NVARCHAR(100) = N'Türkiye'

6. NTEXT (eski)

- Açıklama: Unicode uzun metin, yerine NVARCHAR(MAX) kullanılır.
- Örnek: NTEXT İçerik = N'...'

3. Tarih ve Zaman (Date/Time) Veri Tipleri

1. DATE

- Açıklama: Sadece tarih bilgisi (YYYY-MM-DD).
- Örnek: '2025-05-06'

2. TIME

- Açıklama: Sadece saat bilgisi (hh:mm:ss).
- Örnek: '14:30:00'

3. DATETIME

- Açıklama: Tarih + saat birlikte tutulur.
- Örnek: '2025-05-06 14:30:00'

4. SMALLDATETIME

- Açıklama: Daha az hassas versiyonu, dakikaya kadar.
- Örnek: '2025-05-06 14:00'

5. DATETIME2

- Açıklama: Daha hassas, saniyenin altı basamağına kadar.
- Örnek: '2025-05-06 14:30:00.1234567'

6. DATETIMEOFFSET

- Açıklama: Zaman dilimi bilgisi de içerir (+03:00 gibi).
- Örnek: '2025-05-06 14:30:00 +03:00'

4. Diğer Veri Tipleri

1. UNIQUEIDENTIFIER

- Açıklama: Global benzersiz kimlik (GUID).
- Örnek: '6F9619FF-8B86-D011-B42D-00C04FC964FF'

2. XML

- Açıklama: XML veri tutar.
- Örnek: <Kisi><Ad>Ali</Ad></Kisi>

3. VARBINARY(n)

- Açıklama: Değişken uzunlukta ikili veri (resim, dosya vs.).

- Örnek: VARBINARY(1000)

4. VARBINARY(MAX)

- Açıklama: Büyük boyutlu ikili veri.
- Örnek: Video, ses, resim

CRUD İşlemleri

CRUD, veritabanı işlemlerinin 4 temel adımını temsil eder:

- **C – Create (Oluştur)**
Yeni veri ekleme işlemi.

INSERT INTO Ogrenciler (Ad, Soyad, Yas) VALUES ('Ali', 'Yılmaz', 20);

- **R – Read (Oku)**
Veriyi görüntüleme, sorgulama işlemi.

SELECT * FROM Ogrenciler WHERE Yas > 18;

- **U – Update (Güncelle)**
Mevcut veriyi değiştirme.

UPDATE Ogrenciler SET Yas = 21 WHERE Ad = 'Ali';

- **D – Delete (Sil)**
Veriyi silme işlemi.

DELETE FROM Ogrenciler WHERE Ad = 'Ali';

FOREIGN KEY

Foreign Key, bir tablodaki sütunun başka bir tablodaki **Primary Key (birincil anahtar)** ile ilişkili olduğunu gösteren anahtardır.

Amaç:

- Tablolar arasında **ilişki kurmak**,
- **Veri tutarlılığı sağlamak** (örneğin olmayan bir bölüm koduna öğrenci yazılmasını engellemek).

Foreign key oluşturma

Foreign key veri tabanı tasarımı sırasında, [tablo oluşturma](#) veya daha sonra ihtiyaç halinde bir tabloya eklenebilir.

Foreign key oluşturma – 1

```
CREATE TABLE tablo_adi (  
    sutun_adi veritipi kısıtlama,  
    ...  
    ...  
    FOREIGN KEY (sutun_adi) REFERENCES diger_tablo_adi(diger_sutun_adi)  
        ON UPDATE <EYLEM>  
        ON DELETE <EYLEM>  
);
```

Foreign key oluşturma – 2

```
CREATE TABLE tablo_adi (  
    sutun_adi veritipi kısıtlama,  
    ...  
    ...  
    CONSTRAINT FK_adi FOREIGN KEY (sutun_adi) REFERENCES  
diger_tablo_adi(diger_sutun_adi)  
        ON UPDATE <EYLEM>  
        ON DELETE <EYLEM>  
);
```

Foreign key kısıtlamasına eklenen ON UPDATE ve ON DELETE özellikleri isteğe bağlıdır.

ON UPDATE ve ON DELETE özelliği **diger_tablo_adi** içerisindeki **diger_sutun_adi** ile belirtilen sütunda bir değişiklik olduğunda <EYLEM> ile belirtilen işlemi yapar.

Foreign key eylemleri

Foreign key kısıtlamasına ait eylemler kullanılan VTYS göre değişmektedir.

CASCADE: REFERENCES ile belirtilen sütunda bir eylem (UPDATE, DELETE) olduğunda Foreign key ile belirtilen ilişkili sütunda benzer eylemi yapar.

NO ACTION: Foreign key sütunu ve REFERENCES ile belirtilen sütunda bir ilişki varsa, REFERENCES ile belirtilen sütunda bir eylem (UPDATE, DELETE) izin vermez.

SET NULL: REFERENCES ile belirtilen sütunda bir eylem (UPDATE, DELETE) olduğunda Foreign key ile belirtilen ilişkili sütunu NULL yapar.

NOT: Foreign key sütununda NOT NULL kısıtlaması varsa hata verir.

SET DEFAULT: REFERENCES ile belirtilen sütunda bir eylem (UPDATE, DELETE) olduğunda Foreign key ile belirtilen ilişkili sütuna DEFAULT değerini verir.

NOT: Foreign key sütununda DEFAULT kısıtlaması yoksa hata verir.

RESTRICT: MySQL VTYS içerisinde bulunan bu eylem **NO ACTION** ile aynı işleve sahiptir.

JOIN işlemleri

JOIN işlemleri, SQL'de iki veya daha fazla tablodan veri almak için kullanılan bir işlemdir. Bu işlem, farklı tablolarda bulunan ilişkili verileri birleştirmek için kullanılır. JOIN işlemleri, genellikle birincil anahtar (primary key) ve yabancı anahtar (foreign key) ilişkisi kurarak verileri ilişkilendirir.

- **INNER JOIN:** Sadece eşleşen kayıtları getirir.
- **LEFT JOIN:** Sol tablodaki tüm kayıtları, sağ tablodan eşleşenleri getirir.
- **RIGHT JOIN:** Sağ tablodaki tüm kayıtları, sol tablodan eşleşenleri getirir.
- **FULL JOIN:** Her iki tablodan da tüm kayıtları, eşleşmeyenleri NULL olarak getirir.
- **CROSS JOIN:** Her iki tablodaki tüm kombinasyonları döndürür.
- **SELF JOIN:** Aynı tabloyu kendisiyle birleştirir.

