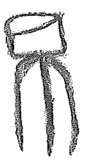
TRANSISTORS : Bilgisayarın yapıtaşı



Bir devrimdir.

Silikondan yapılmıştır.

Belli bir sıcaklığın üstünde iletken,
altında yalıtkan olur.

Mantık kapıları kurarlar.

2'li sistemdeki mantık kurgularını yaparlar.

Bilye düzeneğindeki mantığı kurmayı sağlar.

Çok ucuz ve küçüktür.

Bozulmaz, dayanıklıdır.

Çok az enerjiyle çalışır.

CPU : Central Processing Unit.

Brain of computer.

4.6 GHz = 4,6 Billion cycles per second.

⇓

Saniyede 4.6 milyar kere tetikleme yapıyor (Bilye örneği)

Tetiklemeyi sağlar.

RAM : Random Access Memory

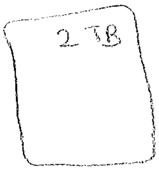
Üzerinde işlem yapılır, depolanmaz.

Bilgisayar çalıştığı sürece çalışır

(Temporary
Fast
Expensive)

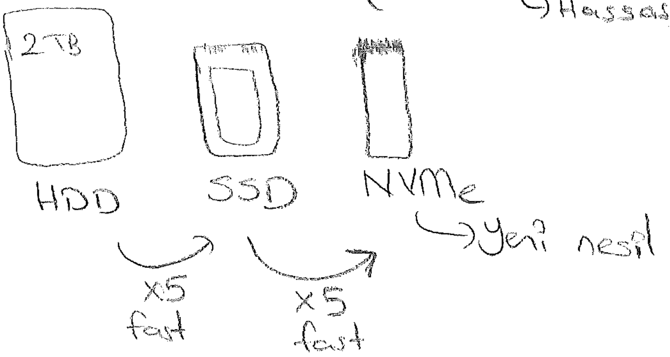
Temporarily stores the data that you are currently using.

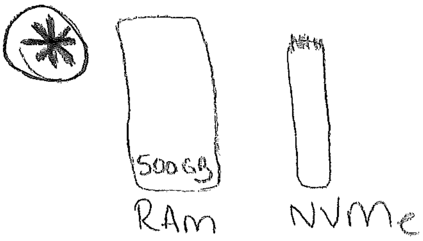
## HARD DRIVE (→HDD) : Elektrik olsa da olmasa da verileri depol



- Permanent
- Moving parts
- Slow
- Cheap
- Vulnerable
  ↳ Hassas

En ucuz
En yavaş





RAM → Depolayamaz
Sadece işlem yapar.
NVMe'den daha hızlı.

NVMe → Depolar.

## GPU : Graphical Processing Unit

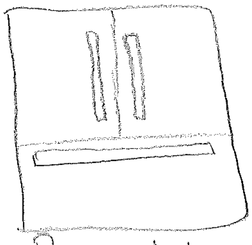
Oyun bilgisayarları, bit coin için kullanılır.

Ekranımızda ne görünmesi gerektiğini tarif eder.

Oyunu sürdürür. Çünkü oyunlar 3 boyutludur ve zordur.

Grafik tasarımları için de gereklidir.

2

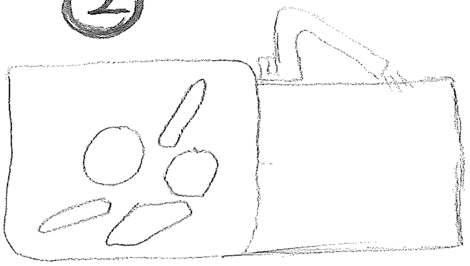


Buzdolabı

⇓

Hard Drive

(Önce aldıklarımızı burda depolarız)

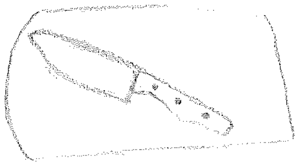


Yiyecekler

⇓

RAM

(Yemek yapacağımız zaman tezgahın üzerinde hazırlarız.)



Kesme tahtası

⇓

CPU

(CPU'nun hızlı hızlı tetikleme yapması gibi

INPUT : Keyboard
Mouse
Touching screen

Siri
Cortana } INPUT & OUTPUT

OUTPUT : Printer
Hoparlor

HARDWARE ⇒ The physical part of computer

How do hardware and software work together?

Hardware makes up the computer & software runs/maintains it.

Which of the following is NOT the purpose of the computer processor?
işlemci

- Run programs
- Display data
- Controls the operations
- Send and receive signals from devices

! Hardware components are connected via ______?

Bus system

! Which is NOT storage device?

- Universal Serial Bus flash drives
- [✓ Random Access Memory]
- Optical Disc Drives
- Magnetic Disc Drives

! If I want to reach my data more faster which type of hard drive should i use?

SSD

! Which part deals with running the games?

GPU

! Which one is NOT a hardware?

- BIOS CHIP
- [✓ KERNEL]
- CMOS
- POWER SUPPLY

! Where do we storage the data permanently?

HDD

## OPERATING SYSTEM (İşletim Sistemi) = YAZILIM

Sadece 1 ve 0 rakamlarından anlayan bir sistemle iletişim kurabilmek için bir ara yüz lazım. Bunu sağlayan sisteme işletim sistemi denir.

Donanım - İnsan arasındaki ilişkiyi kurar.

→ An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs require an operating system to function.

Mac
Chrome
Android
Windows
IOS
Linux
Debian
} İşletim sistemi

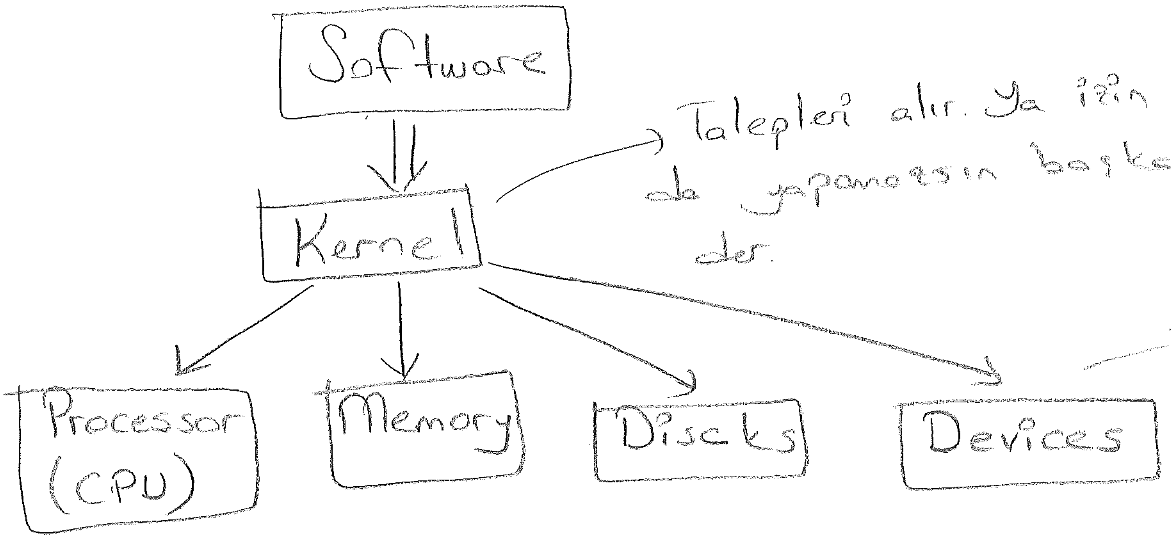
## KERNEL (Körnıl) (Çekirdek)

↓ Kaynakların tek elden yönetilmesi

: Bilgisayarın kahyası

Kaynakları yönetir.

Biri hoparlörü kullanmak istiyor diyelim, Kernel bana soracaksınız der. Çünkü bir hoparlör var. Konuşma yapıyorsan aynı anda hoparlörle müzik dinleyemezsin. Bu yüzden her şeyin tek elden kontrol edilmesi gerekiyor. Önceliğe göre sırayı verir. Mesela müzik dinlerken telefon gelirse, müziği durdurup telefonu çalar. Çünkü öyle bir önceliklendirme



Talepleri alır. Ya izin verir ya da yapamazsın başkası kullanıyor der.

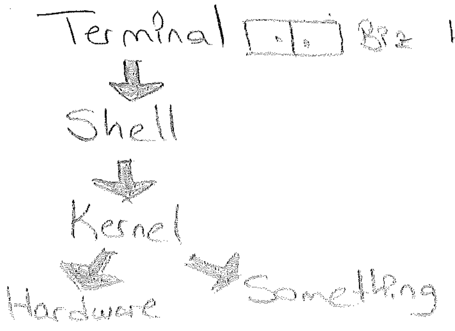
Kernel'in yönettiği kaynaklar

<u>Kernel</u>: It connects applications to the actual processing of data. It also manage s all communications between software and hardware components to ensure usability and reliability.

<u>BIOS</u>: Basic Input Output System

→ It is firmware (sabit yazılım) used to perform hardware initalization during the booting process (power-on startup) and to provide runtime services for operating systems and programs.

ULAK : Sen düğmeye basınca bilgisayar açılıyor. Her şeyi uyandırır. Herkes çalışıyor mu test eder.
Önce ilk sinyali alır
Bilgisayarı bizden komut bekler hale getirir.



Biz buna söylüyoruz. Çünkü alttakilerin dilini bilmiyoruz.

# SOFTWARE AND PROGRAMING LANGUAGES

## SOFTWARE

Computers do not understand human language. They do understand 1s and 0s, bits and bytes. The computer programs fill this gap. A computer program, also called software, is a set of commands that a computer should execute one by one.

Software is a set of instructions, data or programs used to operate computers and execute spesific tasks. Opposite of hardware, which describes the pysical aspects of a computer, the software is a generic term used to refer to applications, scripts and programs that run on a device. The software can be thought of as the variable part of a computer and hardware the invariable part.

Hardware ⇒ Invariable part

Software ⇒ Variable part

## Machine Language

The language that the computers understand, is called
the machine language. As mentioned, this language has
only two letters which are 0 and 1. The binary code
has special meaning for compu-
sequence of these numbers
ters. Different type of computers have different types of
machine language. Because, it's hard to understand
and use for people, other types of languages are develope
high-level
These are Assembly Language (low level) and
→ montaj
programing languages.

## Assembly Language

Assembly language uses short descriptive words,
know as mnemonic to represent each of the machine
language commands. For example, mnemonic add means to
add two numbers. Adding numbers 2 and 3 and getting the
result in assembly code is as such:

```
add 2, 3, result
```

Assembly language is not useful for computers. For
that reason, it should be translated into machine language.
The translator is called "assembler."

Assembly language is not much different than the machine language. And knowing the commands is not just enough. You have to know how the CPU and memory work. So it is also not much use for humans. Fortunately, this type of languages are no longer used. There fore high-level languages are developed in place of this low-level language.

An assembler is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code.

What is assembler?

## HIGH LEVEL LANGUAGES

High-level languages emerged in the 1950s. These languages are similar to human-level languages. Much easier to understand than low level and machine language.

The commands in high level programming languages are called statements. There are many programming languages, each of which has upsides and downsides.

A program in a high-level language is called source program or source code.

As in the assembly language, the source code also should be translated to machine code before it get executed. High-level languages should use tools called compiler or interpreter to make this translation. An interpreter or a compiler reads one statement from the source code and turns it into machine code Then the execution happens right away.

⊛ What is the difference between compiler and interpreter?

A compiler takes entire program and converts it into object code which is typically stored in a file The object code is also refereed as binary code and can be directly executed by the machine after linking.

An interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.

# PACKAGES

## Packages and Package Managers

Software Package is a set of software that
fulfills a particular function, for example,
installation on the desktop. Packages automatically
manage software and **libraries** on your system.
You can install packages from source (from web
site / DVD or USB) or package managers.

**Package Managers** are the tool used to manage
Software and Libraries (together called Packages)
in your OS. They install, upgrade and
uninstall packages easily. There are several
types of package managers. The most popular
ones are Linux System package managers and
programming language package managers.

(*) How can you install software packages
for your computer?

USB, DVD/CD, Package's Official web site,
Package Manager.

# LIBLARIES

Real-life libraries give you a book for "prepared" spesific topic information. You don't have to write again the same book its already prepared by someone and you can't use it, you can do whatever you want with this information. Software libraries and real-life libraries are similar. A software library generally consist of pre-written code, classes, procedures, scripts, configuration data and more. Typically, a developer might manually add a software library to a program to achieve more functionally or to automate a process without writing code for it. For example, when developing a mathematical program or application, a developer may add a mathematics software library to the program to eliminate the need for writing complex functions. All of the available functions within a software library can just be called / used within the program body without defining them explicity. Similarly, a compiler might automatically add a related software library to a program on run time.

(*) What is a software library used ~~for~~?

A software library generally consist of pre-written code, classe procedures, scripts, configuration data and more. So software libraries save programmers' time from writing routine processes' codes everytime. For example, when developing a mathematical program or application, a developer may add a mathematics software library to the program to eliminate the need for writing complex functions.

# Frameworks

→ The father of software libraries

Frameworks are meant to make the life of a developer easier by supplying tools and design patterns to accomplish a task in an expressive and relatively simple way. Frameworks are like the father of software libraries. Frameworks are collections of libraris, classes, functions and constants designed to complete a task easier. They give you a collection of different tasks codes and avoid you from reinventing the wheel.

Frameworks give you a working enviroment. For example, you want to open a coffee shop, you can buy another coffee shop and operate it or you can build it from zero. If you buy another coffee shop, you don't have to pay attention to accessories, coffee shops interior architecture etc. Frameworks supply pre-written software working environments.

For instance, while you are thinking to write your own mobile phone application with Javascript programming language, the React-Native framework gives you the code that every mobile application necessary functions such as visulation to your application at a smartphone.

Frameworks dictate a spesific look to an application and limit the design choices a developer can make in favor of making the code easier to read and write.

# FRAMEWORKS

## Model-View-Controller (MVC)

The Model-View-Controller (MVC) is an architectural pattern that seperates and application into three main logical components: the model, the view and the controller. Each of these components is built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

Model, the data represented in some way, usually from a database.

View, what the user sees. for example the web page you look at.

Controller, the code that manipulates the data in the database. for example insert/delete/update user information to your web site.

## BACKEND vs. FRONTEND

Frontend and Backend development terms are related to web and application development. Before you build your web site you should first develope functionalities that what web site should do and second you shoul visualize it very well.

Fronted and Backend are the two most popular terms used in web development. These terms are very crucial for web development but are quite different from each other. Each side needs to communicate and operate effectively with the other as a single unit to improve the website's functionally.

The frontend of a website or application is what you see and interact with on your device screen. Also referred to as "client-side", it includes everything the user experiences directly: from text and colors to buttons, images and navigation menus.

The backend (or "server-side") is the portion of the website you don't see. It's responsible for storing and organizing data, and ensuring everything on the client-side actually works. The backend communicates with the front-end, sending and receiving information to be displayed as a web page.

## BACKEND

Backend Development refers to the server-side (we will explain later on the Servers section) of development where you are primarily focused on how the site works. Making updates and changes in addition on monitoring the functionality of the site will be your primary responsibilities.

This type of web development usually consist of three parts: a server, an application and a database. Code written by backend developers is what communicates the database information to the web browser. Web browser is your web pages "window application" like Google chrome, Opera, Safari etc. It just prints web site. Anything you can't see easily with the eye such as databases and servers is the work of a backend developer. Backend developer positions are often called programmers or web developers.

## FRONTED

Fronted development manages everything that users visually see first their web page or applications. Fronted developers are responsible for the look and feel of a site. fronted developers primarily use three languages: HTML, CSS, Javacript.

Web designer ⇒ Designs websites with different tools such as photoshop, fireworks or wordpress.

User Interface (UI) Designer ⇒ Is basically a visual designer and is generally focused on design.

görsel

User Experience (UX) Designers ⇒ Work in the front-end, studying and researching how people use the sites. Then they make changes through a lot of testing.

ⓐ LESSON 12.06.2021

⊛ What is algorithm?

An algorithm is a specific procedure for solving a well-defined computational problem.

⊛ What is pseucode?

Algoritmanın taslak çalışması. (Kod yazmadan önce)

⊛ High-level ⇒ İnsan diline ~~daha~~ yakındır.

⊛ Interpreter ⇒ Satır satır çalıştırır (Perakende)

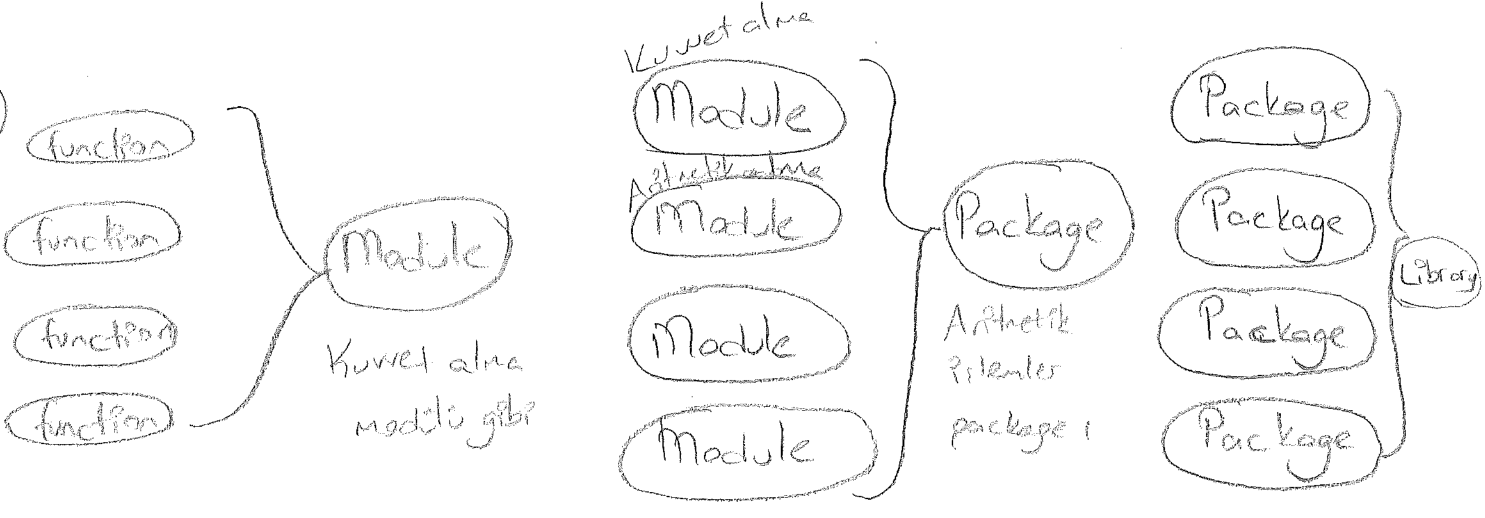
Compiler ⇒ Toptan çalıştırır.

⊛ BUG ⇒ Hata.

↙ Syntax error (prnt)

↘ Logic error (Mantık hatası)

⊛



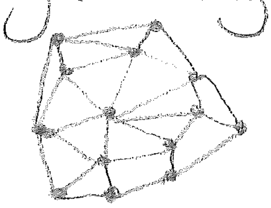
⊛ FRAMEWORK ⇒ Phyton dili kullanarak çeşitli programlar yapabileceğimiz program.

⊛ Backend ⇒ Arka plan

Frontend ⇒ Vitrin (Görsellik, sunum, hız önemli)

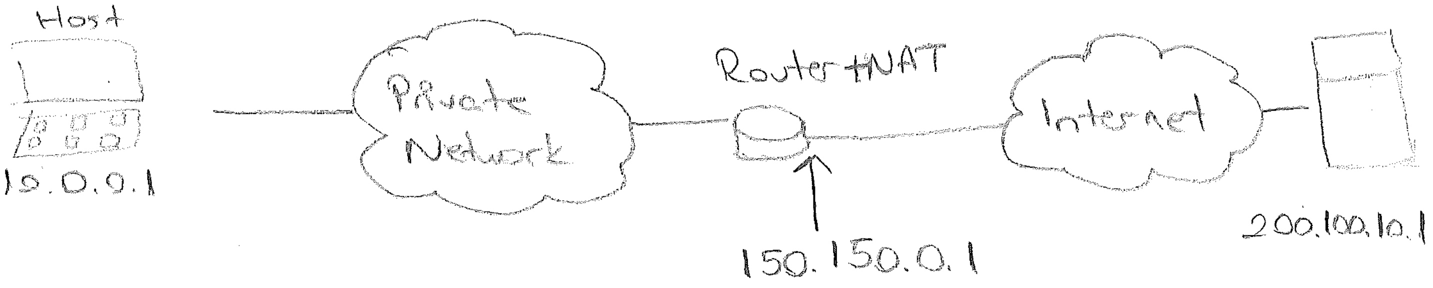
⊛ İnternet ağları birbiriyle hangi sistemle bağlanıyor.

Distributed



⊛ Çok fazla bilgisayar olduktan sonra kurallar koymamız lazım demişler. Bu protokol ⇒ TCP / IP

⊛ ROUTER → özel IP'ler için geçerli ⇒ Sen bu IP ile hedefe gidemezsin deyip yeni bir IP veriyor ve bununla gidiyor. (Public IP veriyor.)



⊛ Bilgisayarlarımızın public IP'si yok.

⊛ İnternete çıktığın adres aynı olduğu sürece hangi cihazdan bağlanırsan bağlan hep aynı IP çıkar

⊛ İki çeşit IP var.

Public IP

Private IP

⊛ Router'ın bir özelliği modem olması.

⊛ IP elektrik gidip-gelmesiyle değişebilir.

bi

13.06.2021

(*) TCP ⇒ Gönderilen tüm paketlerin karşıya ulaşıp ulaşmadığını, doğru sıraya ulaştığını kontrol eder.

## DNS (DOMAIN NAME SYSTEM)

(*) Bizim yazdığımız domain (web) adresini bilgisayarın anlayacağı dile çevirir.

$$\underbrace{\text{https://}}_{\text{Protokol (scheme)}}\ \underbrace{\text{www.}}_{\text{sub-domain}}\ \underbrace{\text{example.}}_{\text{Domain name}}\ \underbrace{\text{com}}_{\text{Top level domain (TLD)}}$$

! DNS = Bilinmeyen Numaralar Servisi

(*) ara çubuğuna cmd yazdık.
→ Sinyal hangi yolları izliyor. Bakabiliyoruz.
(Router yolu)

* IP = Internet Protocol

! Domain Name Server ⇒ Internetin phonebook'u

* WAN ⇒ Geniş ağları bağlar

LAN ⇒ Local ağları bağlar.

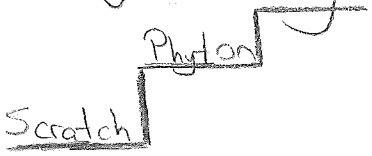
# SCRATCH

⊛ Scratch is a programming language.

⊛ Görselliği çoktur.

⊛ A free block-based programming tool developed by MIT

⊛ Kodlama mantığını kolay anlamak için anlatılacak

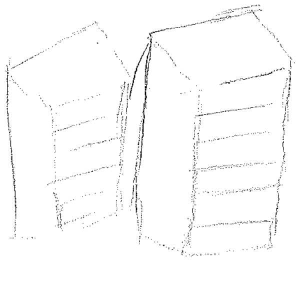


⊛ Açık kaynaktır, ücretsizdir, kolaydır.

C 16.06.2021

## SERVER

→ Sanayi tipli bilgisayar.



Web Server
Email Server
FTP Server
File Server
⋮

Serverlar data senter denen yerlerde bulunur.

- Blade Server
- Rack Server
- Tower Server

*Bir odanın içinde bir sürü çekmece gibi oluyo ya işte onlar serverlar. Taleplere cevap verirler.

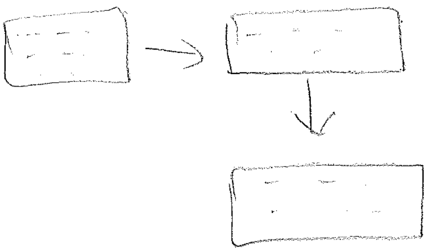
## DATABASES

Çok önemli

Her programın bir bagaja ihtiyacı var. Verilerin istiflendiği gerektiğinde çağrıldığı yerdir. Kalıcı işler. Dataların toplandığı veri tabanı

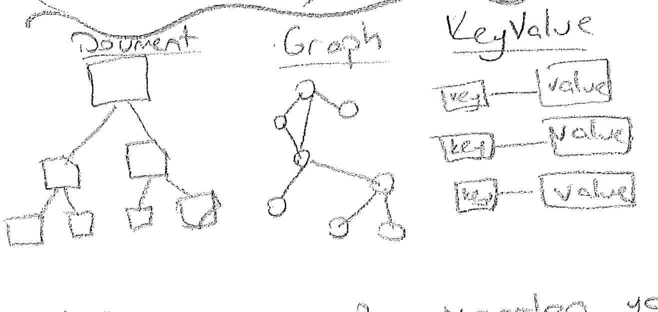
### Relational Database (SQL)

Unique'dir. (Eşsiz, tek)



Bir tablodan aldığını diğerinde kullanılır. Bir kod 1 tanedir.

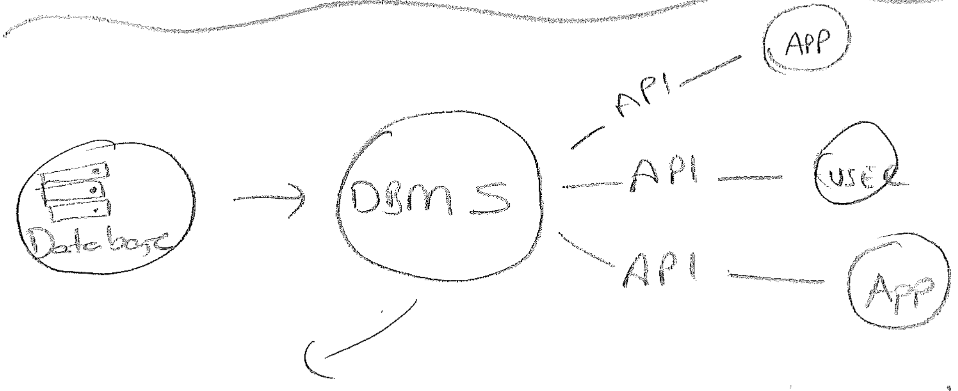
### Non-Relational Database (NoSQL)



Veri akarken bir yandan yazar, bir yandan depolar

→ Bu artık yetmediği için son zamanlarda bu tip database çıktı.

# DBMS (Data Base Management System)



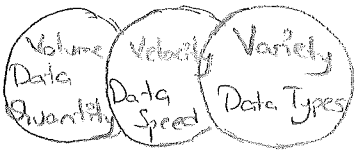
Elektrik kesintisi, deprem gibi büyük olaylarda veri kaybı olmasın diye kayıt yapar.

## BIG DATA

→ Çok büyük bir veri ve çok hızlı akıyor. Büyük şirketler bu veriyi kullanıyor ve bizi takip ediyorlar.

Çok büyük veriler var ve çok hızlı akıyor.

3 V's of Big Data and How They Sum Up the WHOLE Big Data Schematic



Dünyada örneğin aynı anda milyonlarca kişi tweet atıyor. Bunlar aynı ~~anda~~ analiz ediliyor, data tarafından.

Ayakkabı baktığında, ayakkabı arayanlar grubuna giriyorsun. Bu yüzden önüne böyle reklam çıkarıyor. Örneğin attığı tweete göre kitlelerin siyasi görüşlerini ortaya çıkarıyorlar.

Sen Ankaradasın kredi kartın İstanbul'da kullanılmış. Hemen banka sana mesaj atıyor. Bunu sen mi kullandın diye. (Konum servislerin açık olduğu için data biliyor.)

16.06.2021

## CLOUDS

CLOUD COMPUTING ?

Bir işlem kaynağı olması lazım. Bunlar bir havuzda toplanıyor.

Bulut teknolojisini kullanmamızı sağlayan ⇒ VIRTUALIZATION

Sanallaştırma
Depolama alanı sağlar.
Kaynak havuzunun olmasını sağlar.

Kaynakları paylaştırır.

VIRTUALIZATION ⇒ Katmanlara ayırarak küçük paketler yapar. Kasayla değil de kiloyla verir bize.

Public Cloud
Cloud deyince akla ilk gelen.

Community Cloud
Ortak

Hybrid Cloud

Prive Cloud

## WEB BROWSERS

Browser ⇒ Pencere

Kullandığım browser ⇒ Firefox

Web ⇒ Veri okyanusu

HTML ⇒ Structure

CSS ⇒ Style

Javascript ⇒ Behavior (Dinamik)

Kuru iskelet

Stil katar

WORLD WIDE WEB ⇒ INTERNET

HTTP ⇒ Hypertext Transfer Protocol

↳ Burda işlem yapmanın kuralları var.

Client ⟶ Server

COOKIES ⟶ Çerezler

Siteyle ilgili tercihlerin tutulduğu yer.

"BEN SENİN TERCİHLERİNİ, VERİLERİNİ KAYDEDİYORUM" demek

Girince tekrar kullanıcı adı sormuyor.

CYBER ATTACKS

Physical Security → Kapıların kapalı tutulması

Software Security → Yazılım güvenliği

Network Security → Ağ güvenliği

} 3 katmanlı

The Gaining Acces Process

Identification → Sen kimsin

Authentication → Kimliği ıspatla.

Authorization → Tamam sen osun ama yetkin var mı?

Phishing ⇒ E-dolandırıcılık

Malware ⇒ Kötü amaçlı yazılım

SQL Injection ⇒ Sen "gömlek" yazıyorsun. O onu arıyormuş gibi kod yazıyor sistem farkına varmıyor.
(SQLİ)

XSS: Cross-Site Scripting

CSRF: Cross-Site Request Forgery ⇒ Hacker kullanıcıya yaptırıyor farkında olmadan e

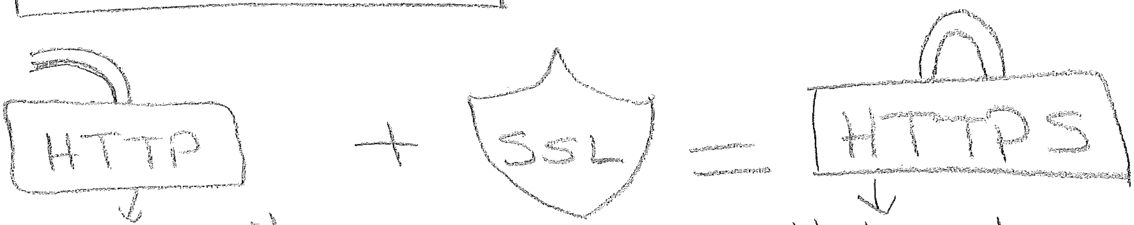
ENCRYPTION ⇒ a dersem b algıla veya ~~tersten~~ oku

3 harf ileri aldım son 3 harf geri al.

Mesela Caesar Cypher gibi

Şifreleme

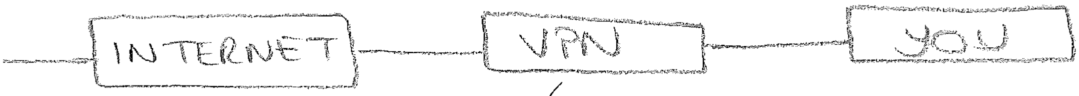
DECRYPTION → Çözümleme



Yeterli değil Çünkü herkes okuyabilir

Herkes okuyamasın diye bir sertifika koydular. Bu protokolle gizlilik sağlandı.

VPN → Virtual Private Network



Şifreli haberleşme. Güvenli bir ağ kuruyorsun

API → Application Programming Interface

Programlar arası aracılık yapar.

Mesela facebook'ta bir arkadaş bulma yeri var orda başka programlar var.