OS - CN - DBMS

- CN
- DBMS
- OS

→ Course Expectation

↳ Theory

Computer Network ⟶ Network Model

↓ TCP/IP

Computer Networks

Network → It is a group or system of interconnected people or items. Computers connected with each other with cables or wireless is called computer networks.

Internet → In a nutshell, internet is a network of computer networks. Complex web of interconnected computer networks.

History Of Internet

1957 → ARPA (advanced research project agency) → 1960's-70's → 1980's → 1990's

1957
↓
Soviet
↓
Sputnik

ARPA
↓
US

1960's-70's
↓
Communication System for ARPA's Computers to talk
↓
1969
ARPANET

1980's
TCP/IP
↓
Internet

1990's
CERN
↓
hyperlink based doc
1990
Tim Berners Lee
WWW
↓
MOSAIC Netscape

Protocols

Network

- Protocols are a set of rules and regulations setup to communicate and share information over a network.

Ex → HTTP, UDP, TCP, SMTP etc

Packets

In order to share data, we can't send big chunk of data over the network. So we divide the data in smaller chunks, these small chunks are called as packets.

Address

Sending messages over the networks requires the destination details. This detail uniquely identify the end system is called as address.

Ports

Any machine could be running many network apps.
In order to distinguish these apps for receiving messages we use ports. (port number)

IP-address + Port

Socket

Port helps you get the packets to specific process on the host

↳ Every process has 16 bit port number

$0 - 2^{16} = 65535$

↓

range of port no.

★ 0-1023 → well known ports

↳ Ex → Port 80 → HTTP

port 443 → HTTPS

* 1024 – 49152 → Registered ports

They are used by specific, potentially proprietary apps / process that are known but not system defined.

Sql server → 1433

mongo → 27017

★ 49152 – 65535 → Dynamic ports

Access Networks

These are media using which end systems connect to the internet.

Network Interface Adapter → It enables a computer to attach to a network. As there are different types of networks, it acts as a single suite to connect to any network

DSL (Digital Subscriber Line)

→ DSL uses the existing telephone groundwork lines for internet connection. Generally DSL is provided by same company which supplies telephone service.

↳ ISP (Internet Service Provider) → It is just a company that provides end users internet. Ex → AT&T

Network Protocol Stack

OSI (7 layer)

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

TCP/IP (5 layer)

- Application
- Transport
- Network
- Data Link
- Physical

Application layer → email server, chat server, browser

Presentation → Presentation of data, compression encryption

Session → User session managed

Transport → Divides big chunk of data coming from above to small chunks. & mgt these chunks.

Network → how route of packets will be done in the network

Data link layer → error / flow control, multiplexing & demultiplexing, handle address

Physical

Application layer

Roles →

1) Writing / providing data off to the network
2) Reading the data from user
3) Contains applications that helps users to interact on the network
4) Error handling & recovery can also be done.

Where it exists??

↳ End systems

↳ Instant messaging

↳ www

↳ VoIP

↳ email

Client - Server Architecture →

↳ It is a 2- level architecture

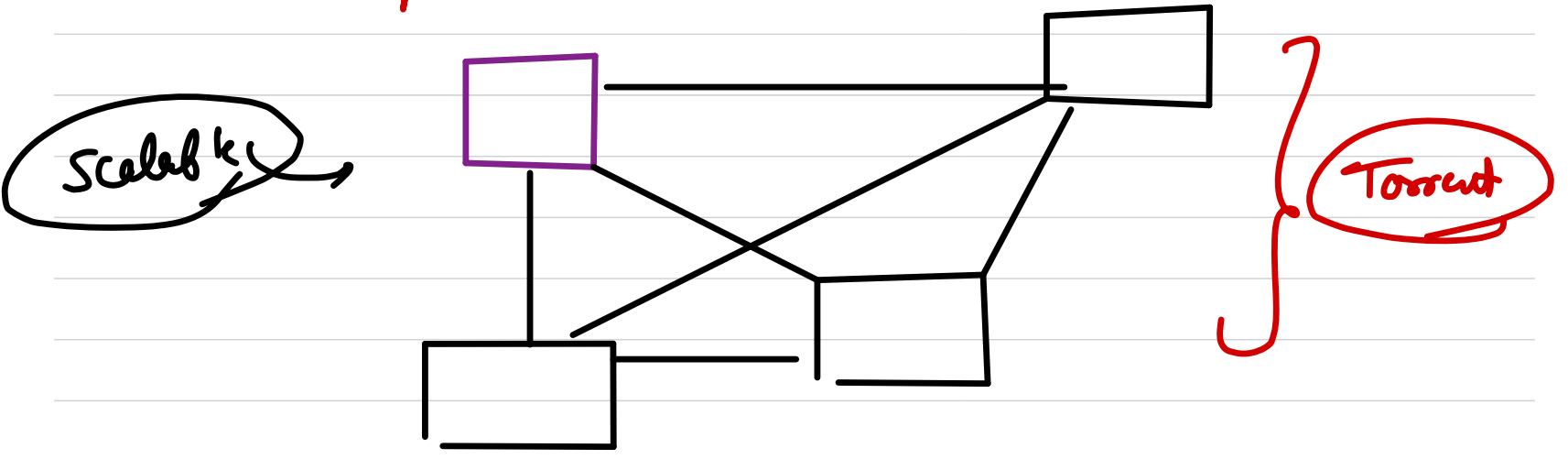
Client-side Server side

Servers → this process controls access to a centralized resource or service such as a website / webapp

Client → frontend where user interacts.

P2P architecture

(Peer to peer)



Hybrid → combination of client-server & p2p architecture

HTTP

HTML

It stands for hyper text transfer protocol

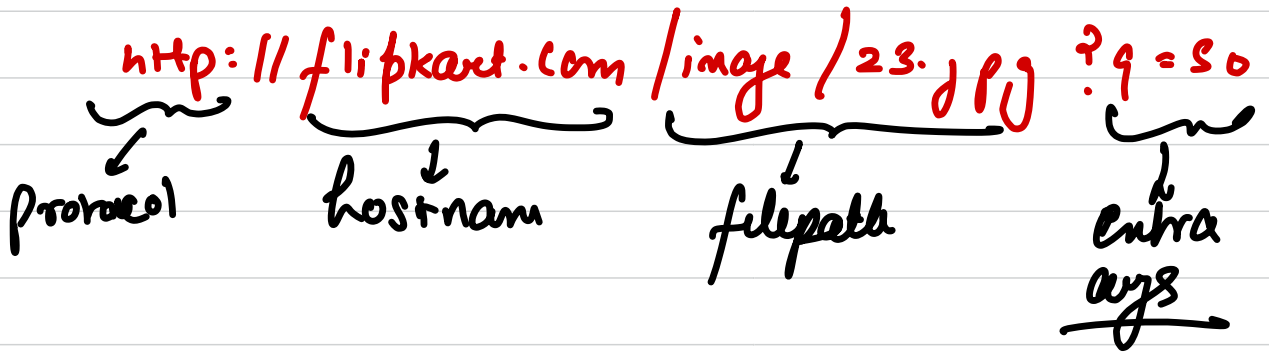
↳ **Objects** → web pages are the main objects that contains other objects.

Some other objects can be mp3 file, png, jpg etc

Every objects has a URL

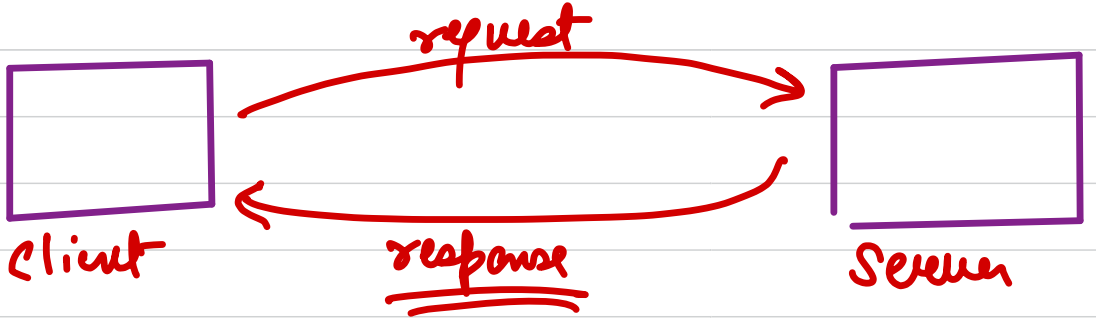
URL (uniform resource locator)

- protocol
- hostname
- location of file / object
- arguments


http://flipkart.com/image/23.jpg?q=50
protocol
hostname
filepath
extra args

HTTP

It defines the whole procedure on how client & server will interact.



The first msg → http request
The second msg → http response

] http is also called as request-response protocols

→ http is stateless protocol. (servers donot store any information about the client)

So, a lot of application layer protocols depend on lower level protocols of transport layer

→ In transport layer there are 2 main protocols

① TCP ② UDP

→ HTTP depends on TCP

There are two type of HTTP connections -

1) persistent http
2) non- persistent http

HTTP req and response msgs

HTTP Req Msg →

any http message are plain ASCII text

- host
- method
- status code
- Referer-policy
- ⋮

Method	URL
Get	www.booky.com

There are multiple http methods :

1) Get → request some data
2) Post → put some data on the server
3) Put
4) Patch
   → update data on server
5) Delete → delete an object at a given url

* user-agent → It specifies the client. Useful when server has different web pages that exist for different devices.

* Accept-Language → specifies the preferred language

* Connection : close

HTTP Status code:

Convention → SOAP

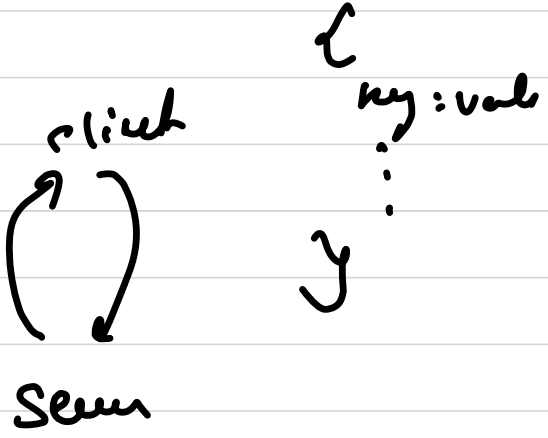
REST

GRPC

XML

JSON

protocol buffers



deal everything as a resource

deal everything as an action

Torrent

DNS

Cookie

SMTP

Cookies

↳ These are mainly concerned towards privacy.

↳ HTTP is a stateless protocol, & a lot of time user session is reqd.

Q→ How cookies work ??

cookies are unique identifier strings. These are set by the server through http header.

as soon as a cookies is shared, it is sent along with subsequent http req to the same server. this allows server to know who is contacting it and hence serve the content accordingly.

★ Set-Cookie header →

When a server wants to set a cookie it includes "Set-Cookie : value" in the http response.

this value is stored in the cookie folder of browser

Email: SMTP

⇒ For executing the functionality of email, SMTP (Simple Mail Transfer Protocol) is used. One more protocol named POP3 is used in combination with SMTP. One is used to send emails that are stored in the user's inbox & other is used to retrieve emails sent to a user.

SMTP also used TCP protocol from transport layer.

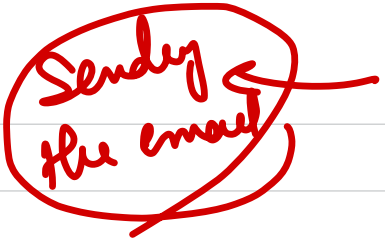
Connection for SMTP is setup on port 25.

Mail clients gives the actual UI for end users to send and receive mail. viz gmail, outlook etc

How SMTP works??

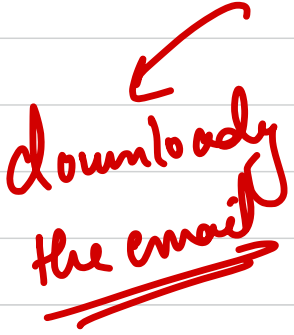
* When an Email is sent, it is sent to the Sender's SMTP server using SMTP protocol (Also the SMTP server is configured in the mail clients)
* The SMTP server places the email on a message queue.

* then SMTP server initiates a connection with receiver's SMTP server and conducts an initial SMTP handshake.
* then finally it sends the email to receiver's SMTP server.
* the email is downloaded from receiver's SMTP server & then the client shows the mail.



SMTP → push protocol

POP3/IMAP → pull protocols



If receipient SMTP server is offline, the sender SMTP server tries again & again after some delta mins. There is a set threshold after which it stops sending the email & marks it not delivered.

POP3

POP (Post office protocol)

It downloads emails in 4 phases.

1) Connect
2) Authorize
3) Transaction
4) Update

Two modes of POP
- → download & keep
- → download & delete

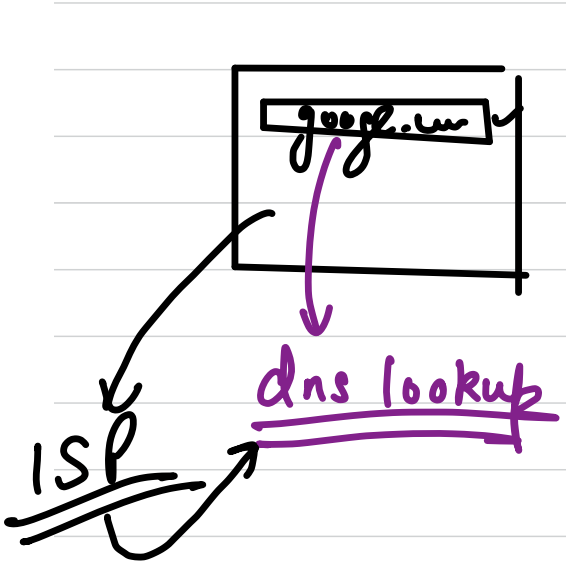
IMAP (Internet message access protocol)

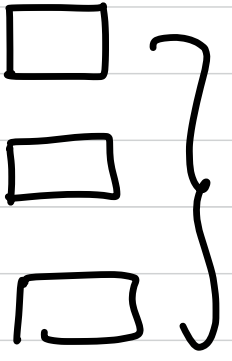
→ Emails are kept on the Server & not deleted

→ local copies of the emails are cached on each device

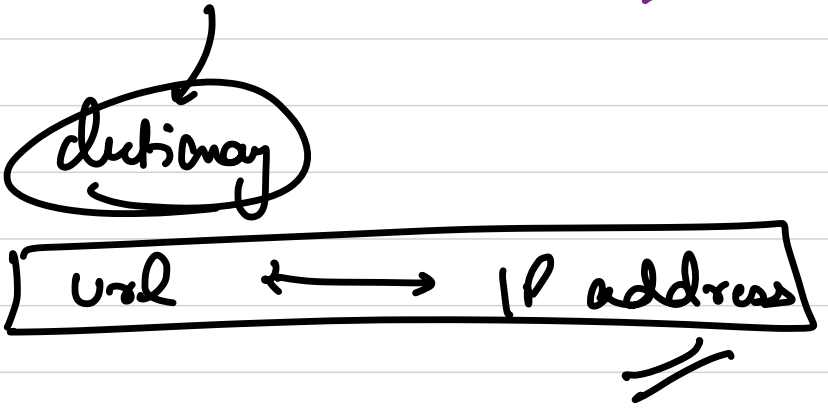
→ If an email is deleted by user manually then only it gets deleted from server.

What happens when you write www.google.com on the browser ??





DNS (domain name server)



http :// cs144.scs.stanford.edu / labs / sc.html

application protocal

host → IP address

file / resource

HOSTS.txt (URL - IP)

all the hosts were in a file hosts.txt, maintained by a network information center

→ using FTP → they first downloads new copy of hosts.txt

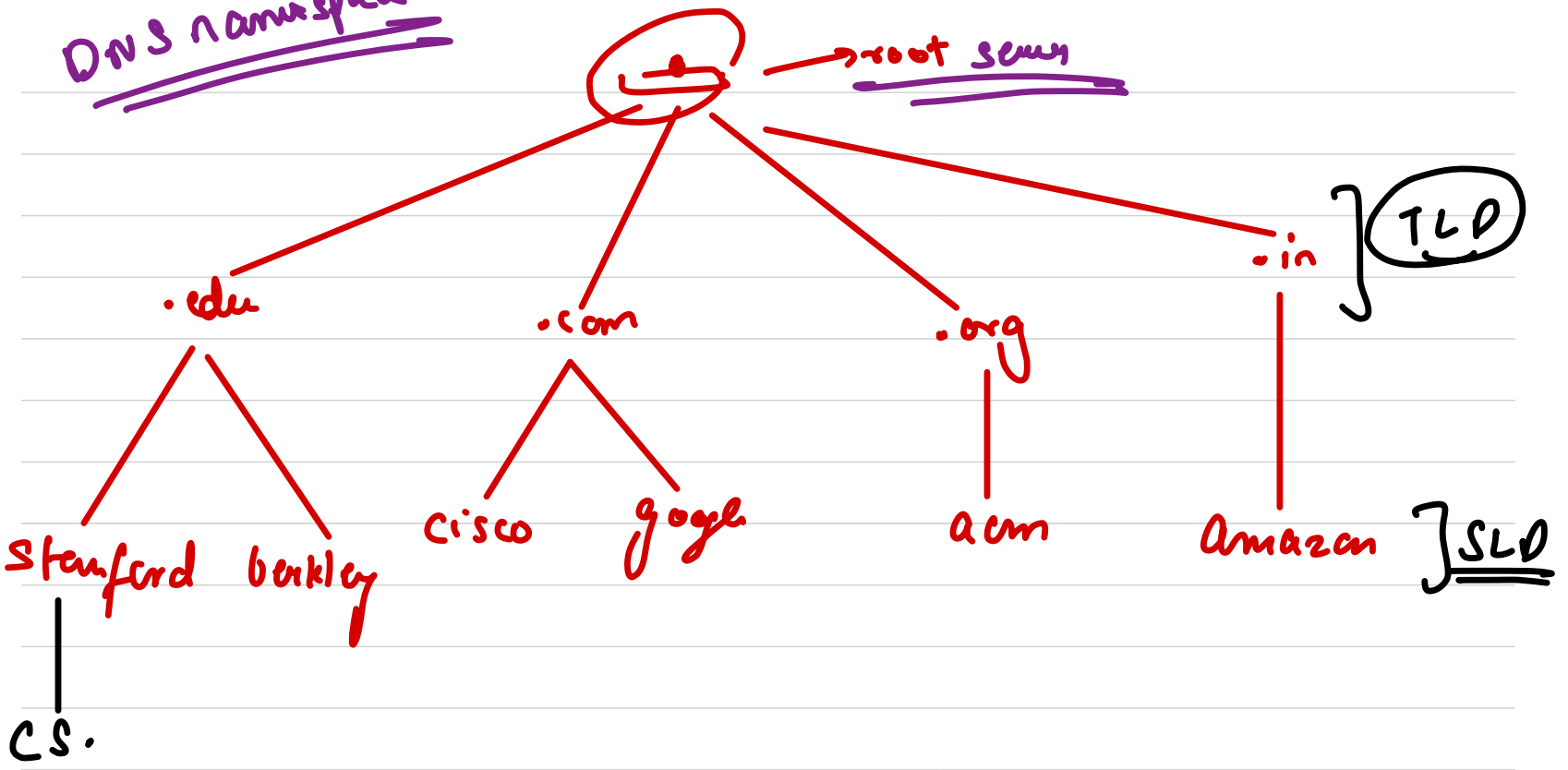
Domain Name System (DNS)

↳ maps names to address

↳ was able to handle huge records.

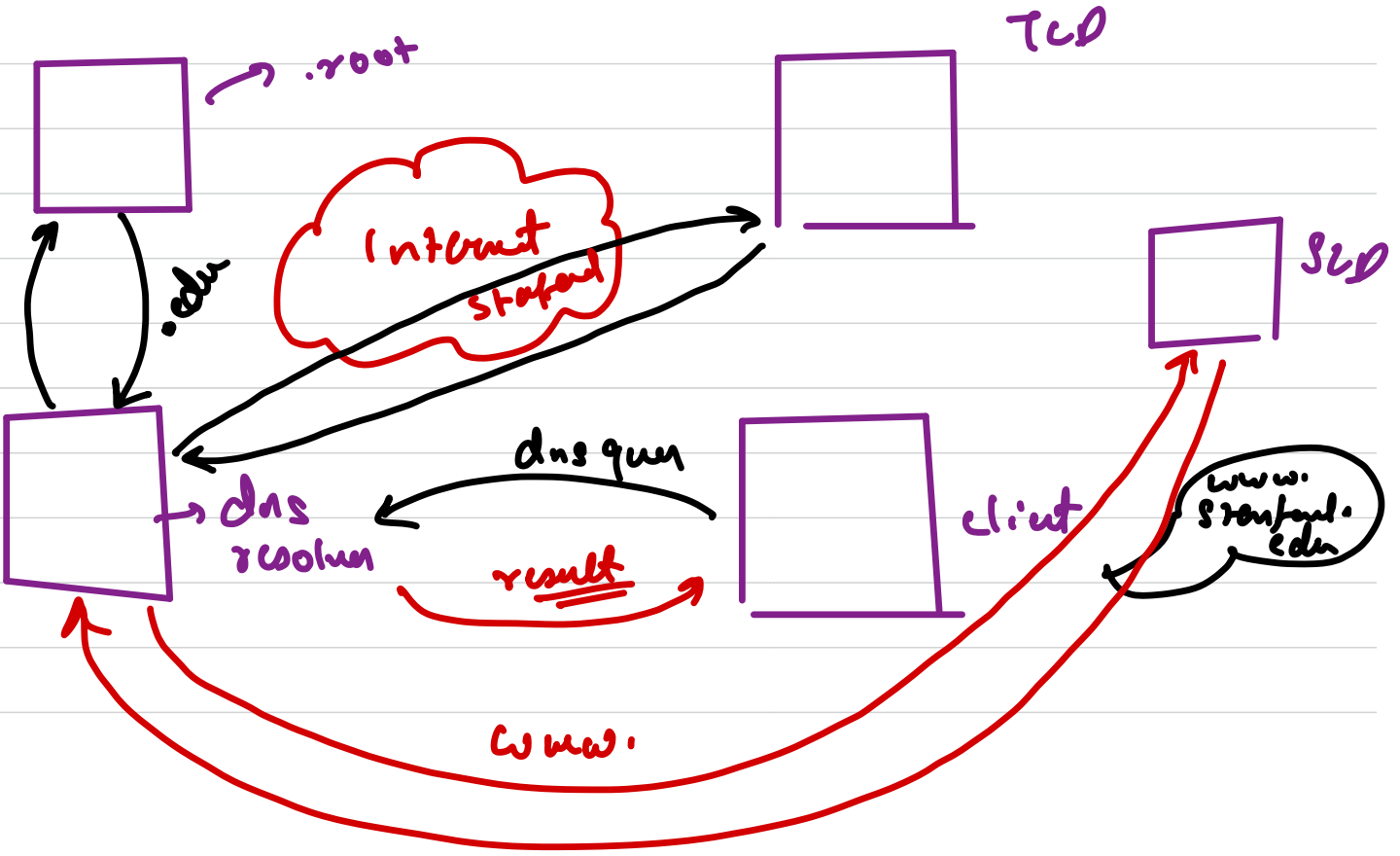
↳ Robust to failures.

It is a tree based structure


DNS namespace
.
root servers
.edu
.com
.org
.in
TLD
stanford
berkley
cisco
gogle
acm
amazon
SLD
CS.

DNS Query

read only



TORRENT

→ It is a protocol for P2P file sharing. A bit Torrent client is an application that uses this protocol.

→ actually it not only follows P2P but instead follows hybrid architecture

→ Allows easy access for share large files.

→ A bit torrent client requests files from multiple clients in parallel.

→ Small chunks of data is called pieces.

→ If a client succesfully downloads a piece, torrent tells all other clients, that this piece is now also available in this new client & can be downloaded from this client too.

→ The collection of collaborating clients are called "Swarms"

Torrent file

→ A client joins a swarm by downloady a .torrent file

→ Gives info about the file they shared, like how big it is, size of it's pieces, how to start interacting with other clients etc

→ Gives info about TRACKER

→ When a client, joins the swarm, it req a list of clients from the tracker, & starts communication with these clients over TCP (initially acts as a leecher)

→ When the size of swarm increases, we can also use trackers (ces torrents. (distributed HT)

Q. What exactly torrent does??

(1) It breaks the file they shared into N pieces.

→ for better perf → 256 kb – 1MB

→ It uses TCP so the sharing of pieces is reliable.

(2) To ensure pieces integrity → torrent attaches SHA-1 hash to each piece

→ Peers exchange whatever pieces they have.

Peers download the rarest piece first

if any piece is unavailable in all peers, no one can download it.

This is called as "Rarest first policy"

TFT policy (Tit for tat)

↳ You send data to peers, who send you data. The peers who contribute more can download faster, this creates incentives for seeders.

TRASPORT LAYER

Transport layer, takes messages from network to applications & vice-a-versa.

It segments the data in small manageable pieces called as "Segments" or "Datagrams".

It allows logical app-app delivery,

It multiplexes & demultiplexes data.

Where it exists

↳ Transport layer also exists on end systems.

In transport layer there are 2 main protocols

1) TCP → Transfer Control protocol
2) UDP → User datagram protocol

TCP

→ While using TCP small pieces of data are called segments

Ensures reliable and in-order delivery of segments

Detects modifications on the packets during delivery & corrects them.

Slower than UDP

Ex → HTTP, Email, FTP

UDP

While using UDP small pieces of data are called datagrams

Does not ensure reliable delivery of datagrams.

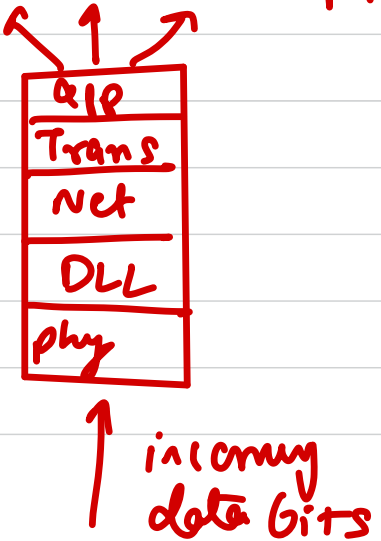
It also detects modifications but does not correct them.

Faster than TCP

Ex → VOIP, live streaming

Multiplexing And Demultiplexing

* Demultiplexing → It is the process of delivering the correct packets to correct apps from one stream.



* Multiplexing → It allows data to be sent to more than one dest. host via a single medium.

How all of this is managed ??

↳ Port no.'s help in multiplexing & demultiplexing.

Congestion

When more packets than the network bandwidth is sent through it, then the performance drops & loss of data occurs. This is called as congestion

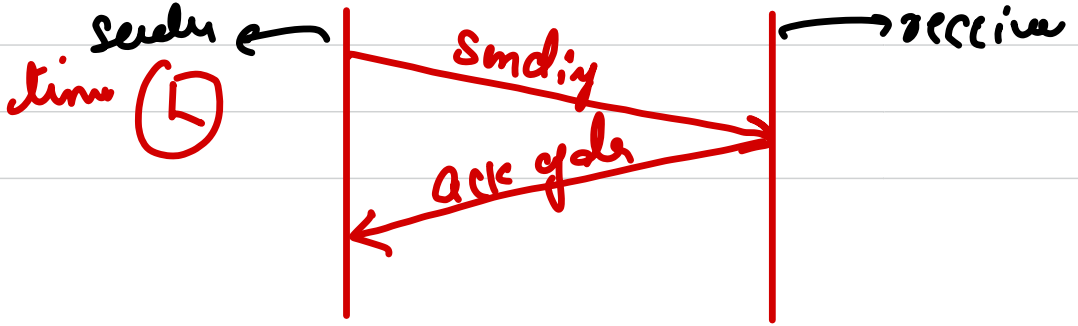
Imperfections in Network Layer.

↳ Segments can be corrupted / lost / damaged / reordered / duplicated etc

Checksums → It is an error-detection mechanism

We can keep an arithmetic sum of all bytes of segment. After sending we can match the checksum on receiver's end with the one on sender's end.

Retransmission timer →



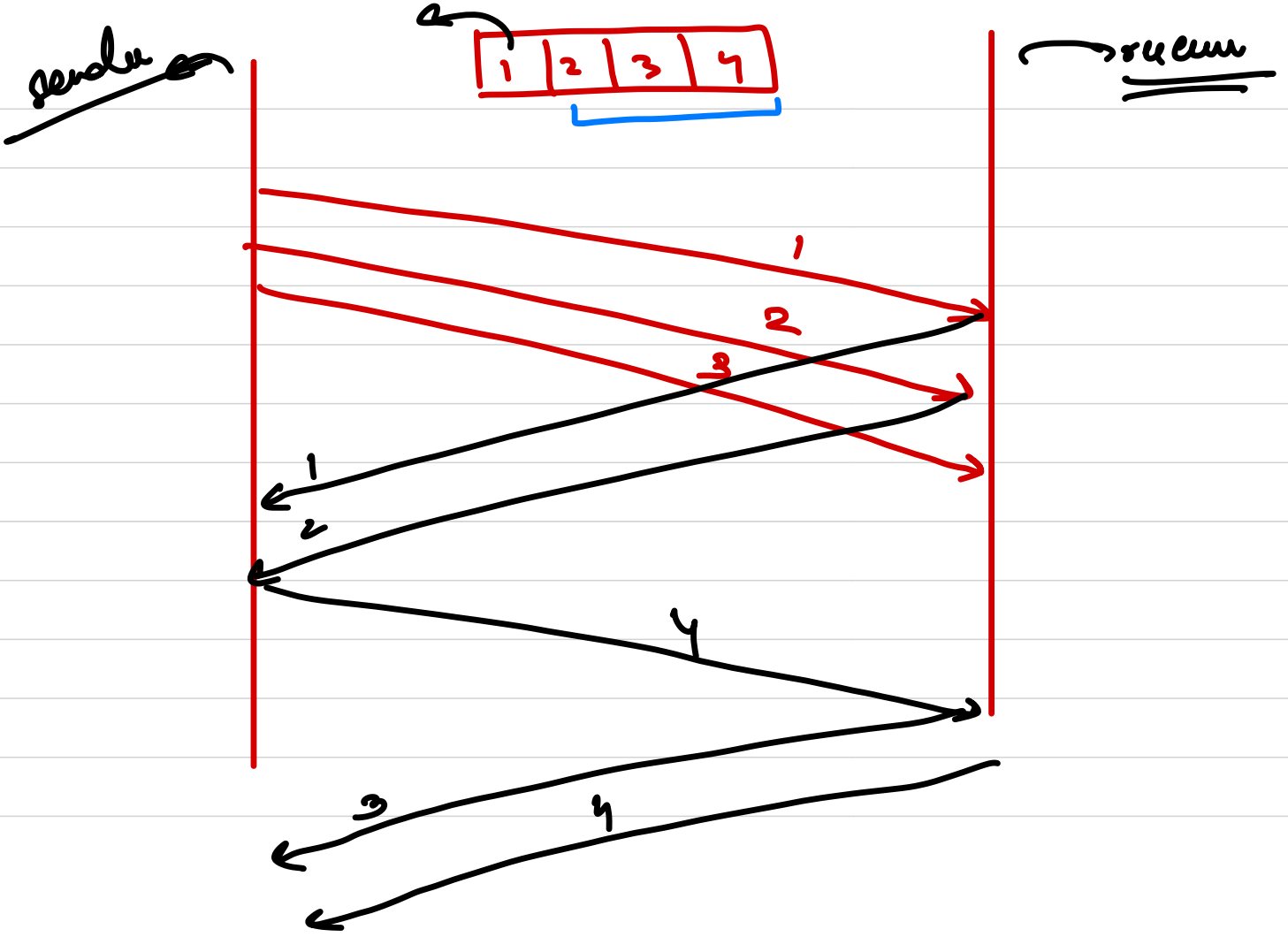
Sequence Number → unique id no for every segment

Pipelining → apps can generate data at a much higher rate than expected by network.

for TCP → reliable message comm.

* Sliding Window

→ It is the set of consecutive no.'s that the sender can use when transmitting segment without being forced to wait for ack.


sender
1
2
3
4
receiver
1
2
3
4
1
2
4
3
4

Detection & re-transmission of packets cannot be handled just by Sliding window protocol.

Go-Back N

It is defined at two ports in a network

1) Sender

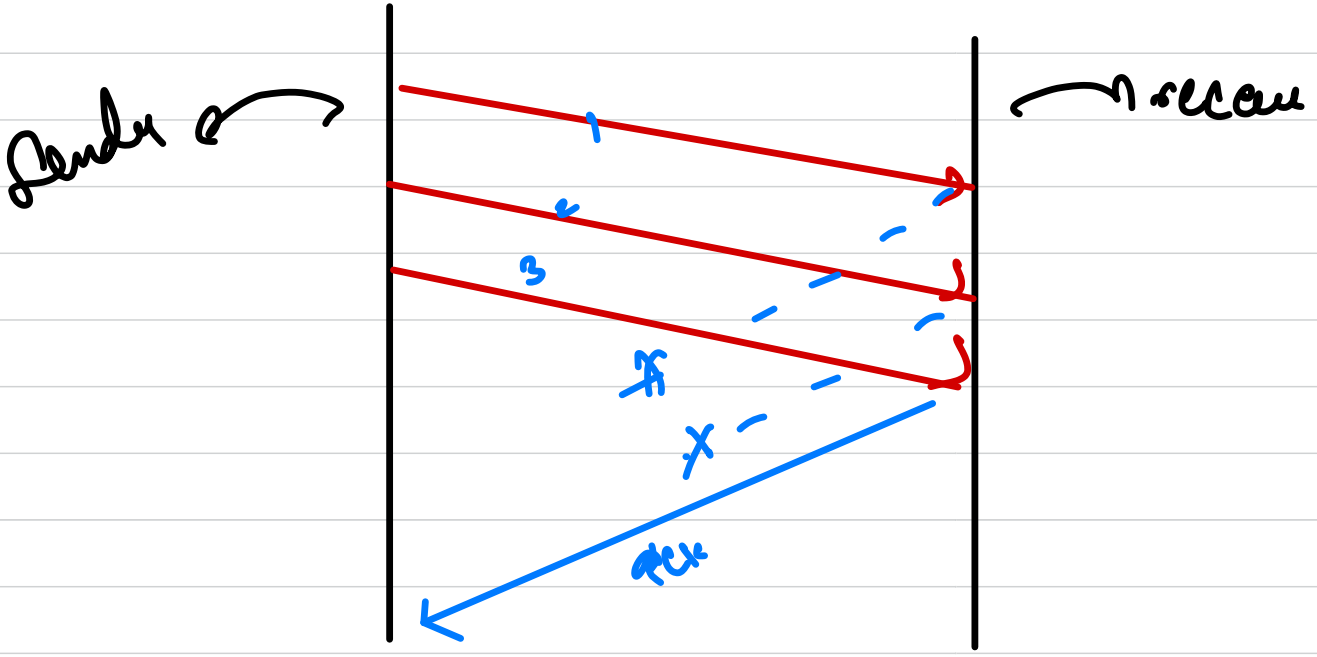
2) Receiver

→ Receiver →

↳ It only accepts that packets that come in seq

↳ any out of sequence packet is discarded.

↳ when it receives a data segment, it sends an ack containg seq no of last in-seq segment.


Sender
Receiver
1
2
3
4
x
4x

→ Sender → Segments are sent based on sliding window

Sender must wait for ack if the buffer is full

When the sender receives an ack it removes all

previous acknowledged segments.

Selective Repeat

→ This also uses sliding window like go-back-n.

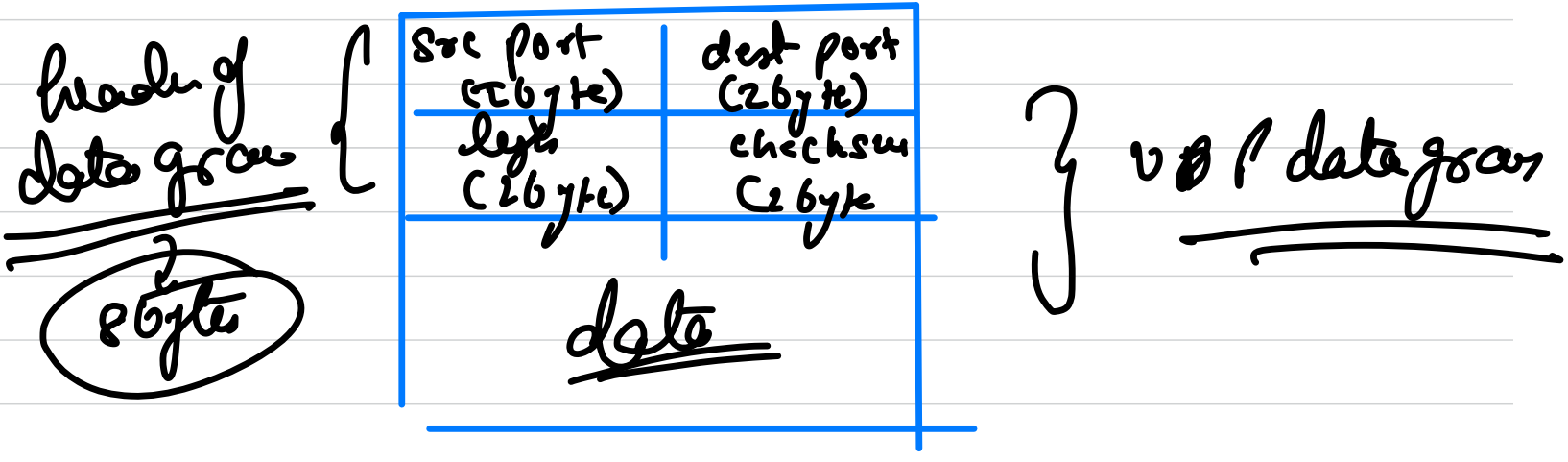
→ Window size should be less than or equal to half of the seq no.

Sender retransmits unack packets after a timeout.

Receiver ack all correct packets.

UDP (user data gram protocal)

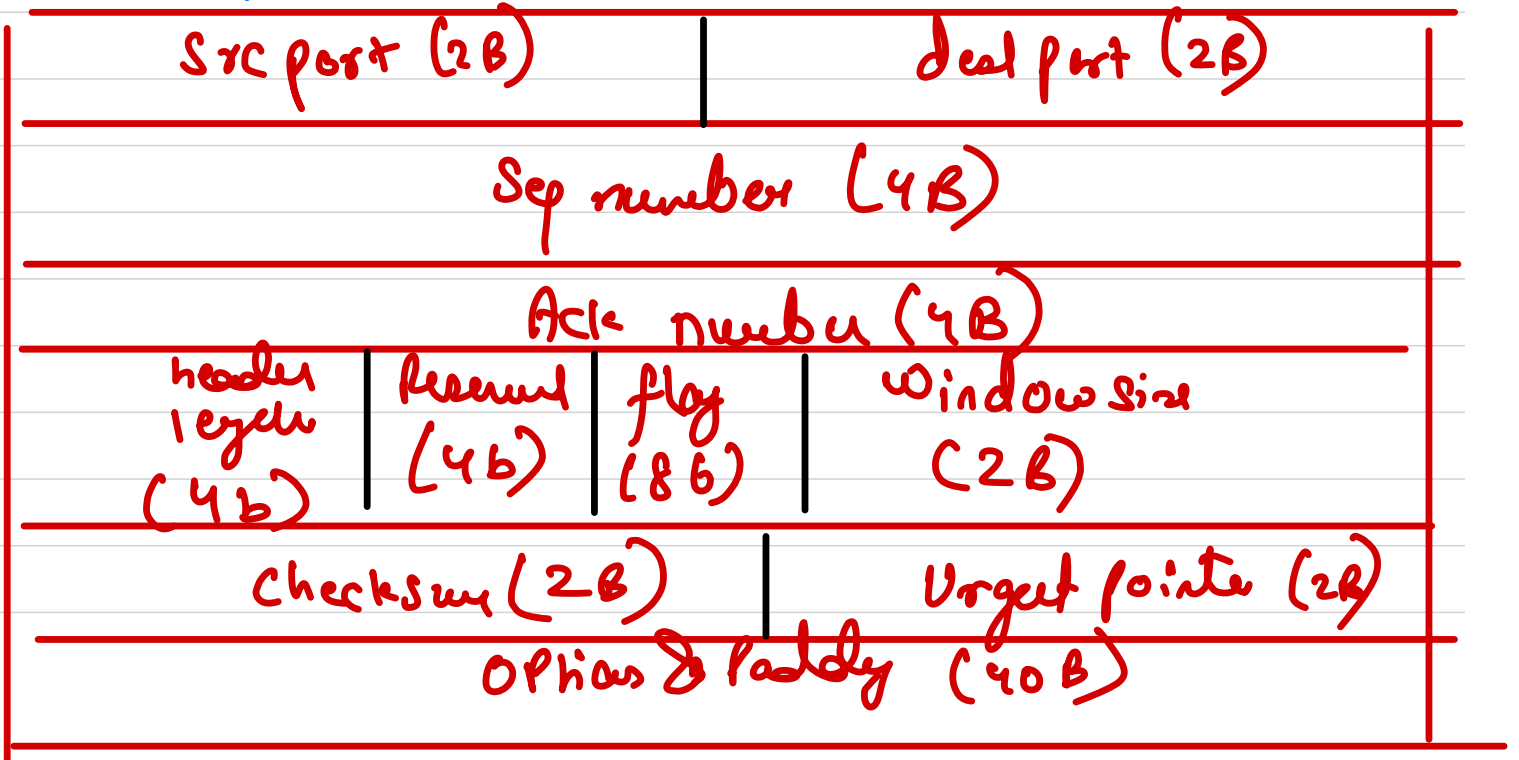
→ Non reliable

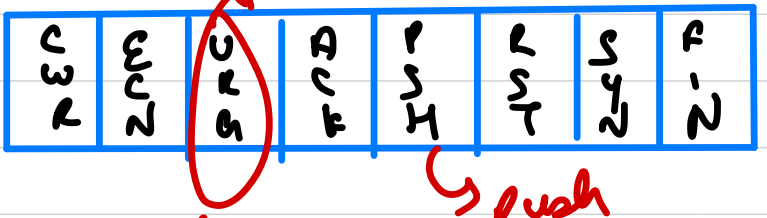


TCP (transmission control protocol)

→ reliable

→ header of TCP





ACK → acknowledgement

RST → reset

SYN → Synchronization

FIN → finish

TCP

Reliable

Connection Oriented

Error Control

flow control

full duplex

How to setup a TCP Connection

Three way handshake →

1) Initiating a Connection →

SYN flag set
Seq no → random value

} → send by sender

2) Responding to initial connection msg

SYN flag set

ACK flag is set
ACK no. is set

} SYN+ACK Segment

3) Connection established: ACK → Sender sends to receiver.

↓
SYN is not set
ACK is set

Network Layer

Main objective is to allow end systems to exchange info through intermediate systems called as Routers. The unit of info in network layer is called packet.

Services of Network layer →

1) Unreliable Connectionless service →

2) Connection oriented and reliable

The main responsibility of Network layer is routing of packets.

Static Routing

Dynamic Routing

Static Routing → Here we have manually computed routes which manually managed in routing table. It is good for small network. It does not adapt automatically to changes in network.

1 Dynamic Routing → They have adaptable routing mechanism & routing tables.

There are 2 types of dynamic routing -

1. Link state
2. Distance vector


Routing
Static
Dynamic
link state
dist. vector

Database Management System

In the earlier times → file based systems

Problems

1) Data redundancy
2) Data inconsistency
3) Security issues
4) Concurrent access
5) Difficult data access

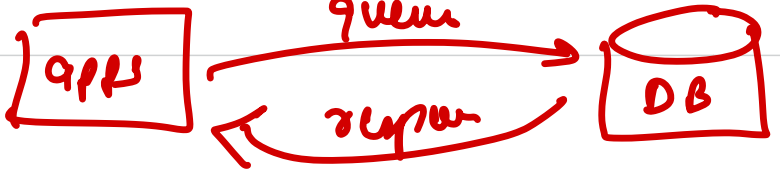
* Databases It is a shared collection of data

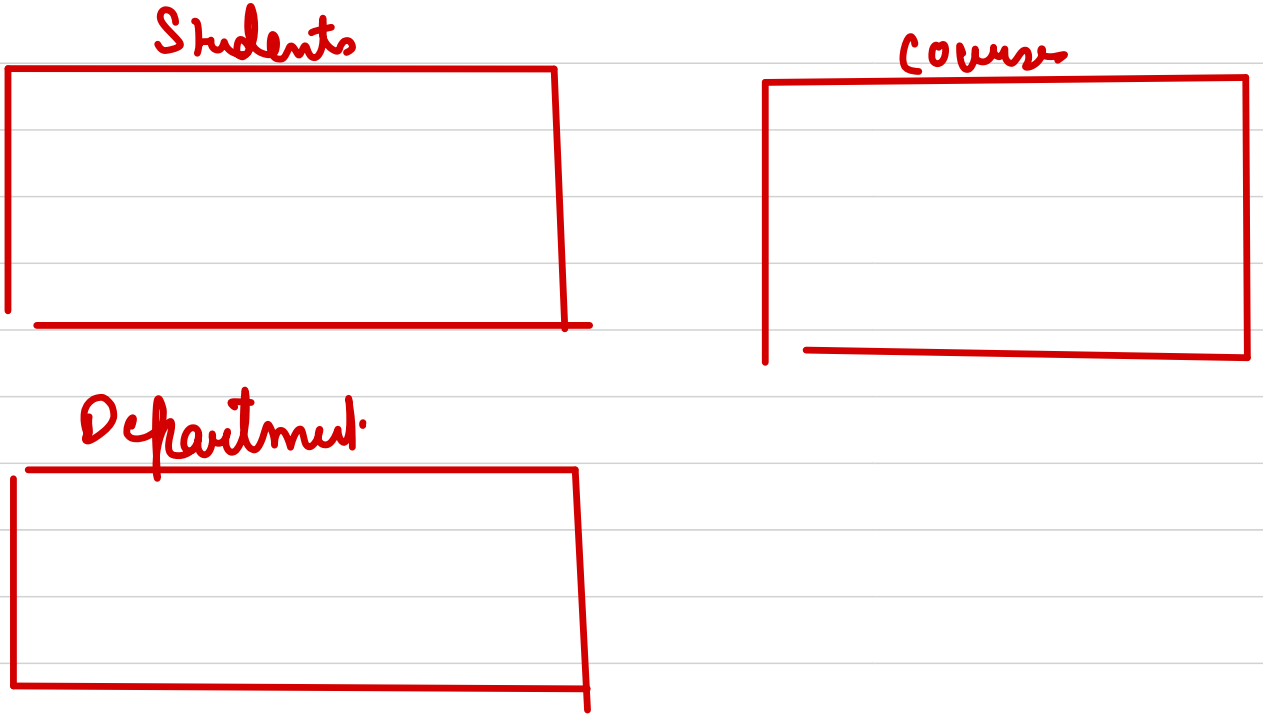
Properties of Databases

DB represents real world entities and relationships, which is also called as universe of discourse
There is some logical coherency in the data
DB are created for specific usecases.

facilities provided by database:

1) Define the data in logical manner
2) Easy manipulation of data
3) Easy sharing of data between multiple people & apps
4) Smooth way of adding or deleting data.




Students
Courses
Department

Data Models

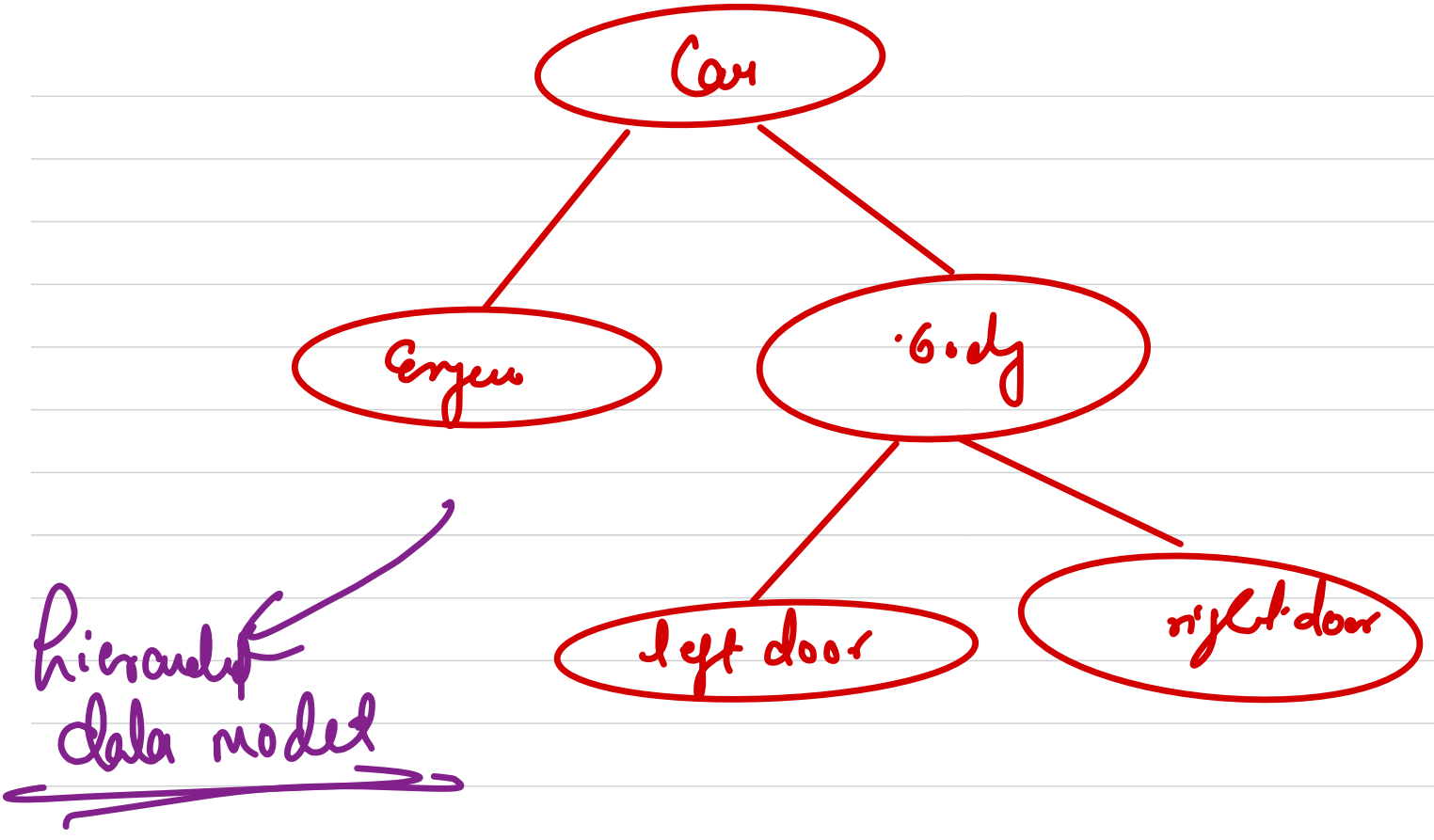
To hide the implementation details of a database we use data models. which helps us in achieving data abstraction.

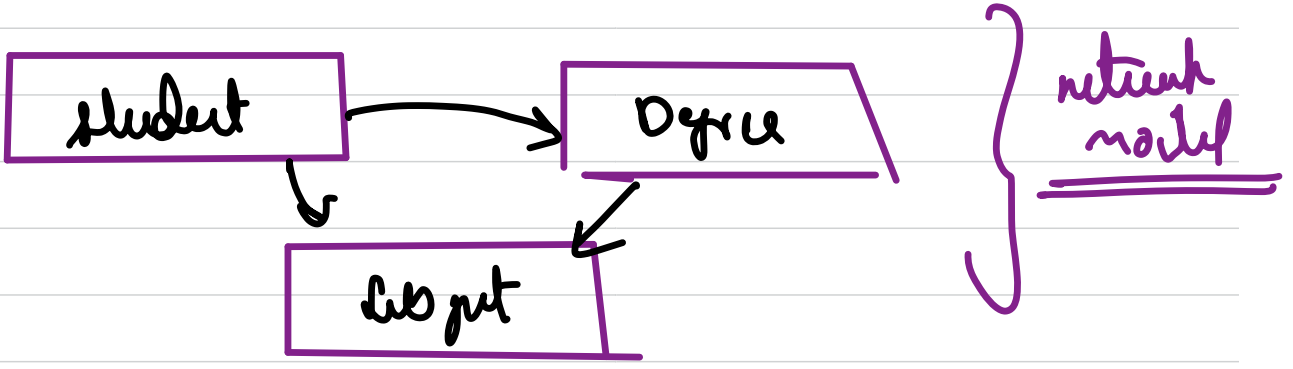
data model is a collection of notations for describing the data, it's relationships & constraints

Types of data models →

1) High level conceptual data model ⇒ We use ER diagram (Entity relationship diagram)

2) Record based data model →
   - ↳ Hierarchical
   - ↳ Network
   - ↳ Relational


Car
Engine
Body
left door
right door
hierarchy
data model



Physical data model

Database Schema → It is the blueprint of the database

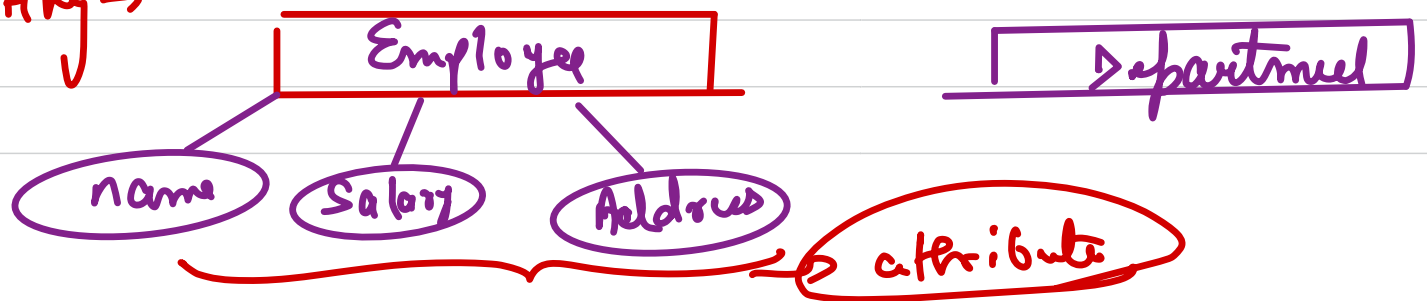
Diff → Data Model | Data Schema

E-R Model

Entity → these are real world objects

Relationship → this is how the entities are related.

Entity →



* Attributes → these are properties that define the database.

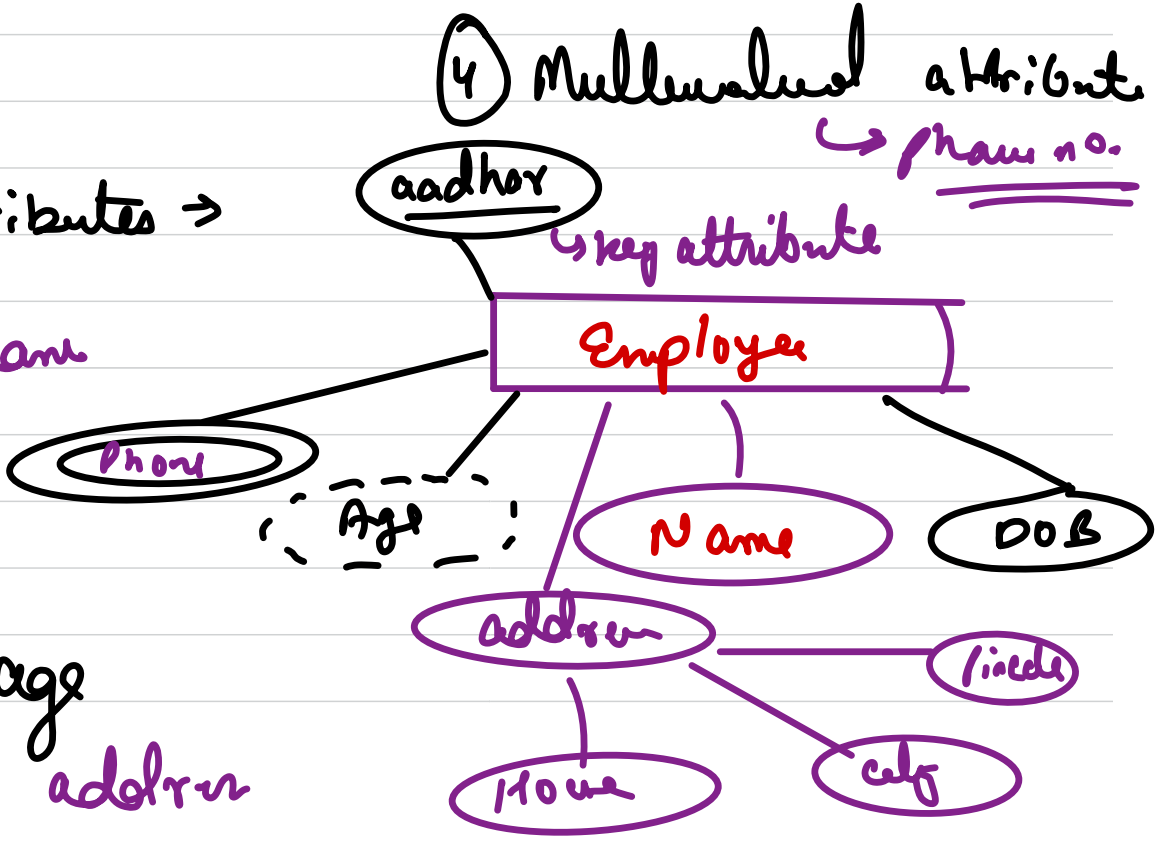
types of attributes →

(1) Simple → name

(2) Derived → age

(3) Composite → address

(4) Multivalued attribute → phone no.

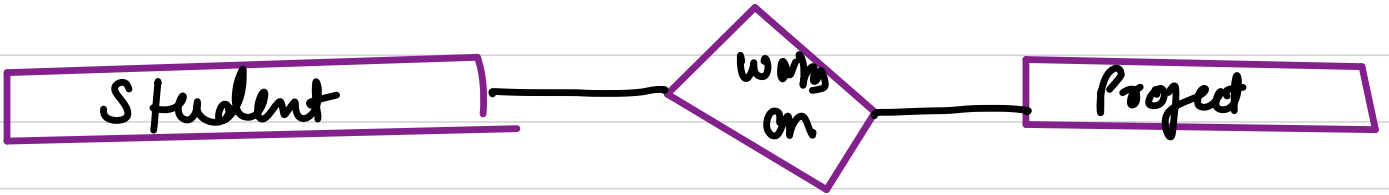


keys → In an ER diagram, an imp constraint on the entities is the KEY or uniqueness attribute. This unique attribute is also called as primary key.

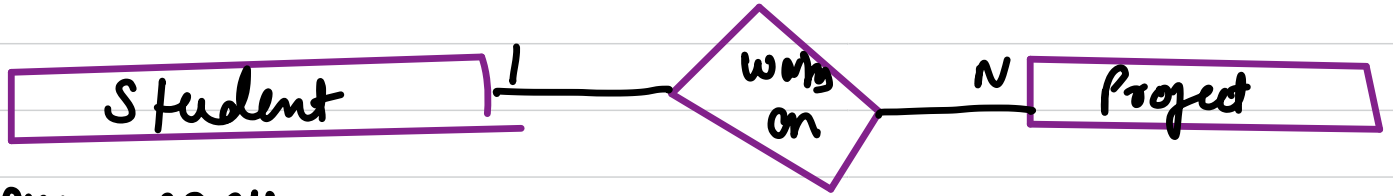
But, sometimes we need 2 or more than 2 attributes to uniquely identify a record. Then these type of attributes are called composite keys.

Relationship

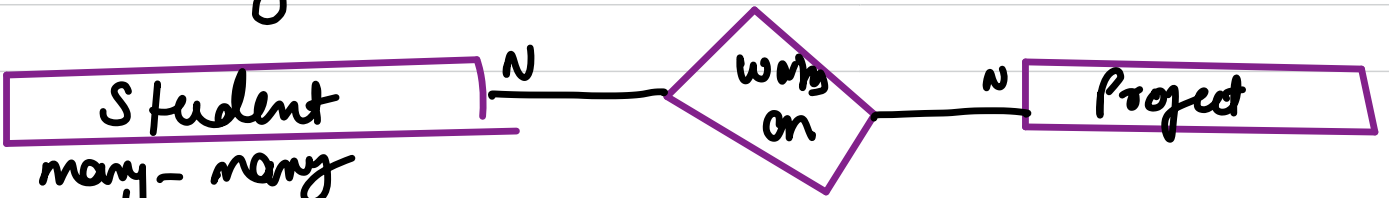
Binary relationships



One-one relation

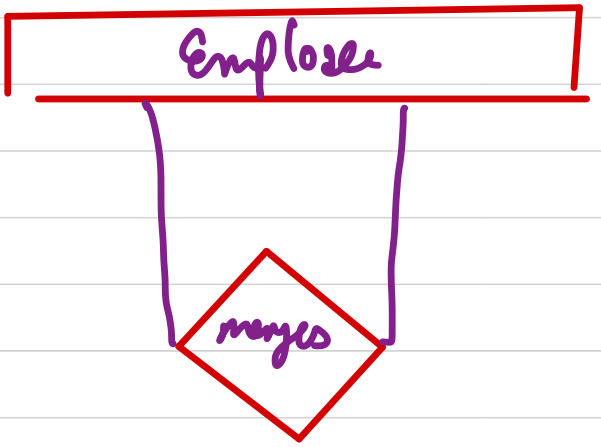


one-many

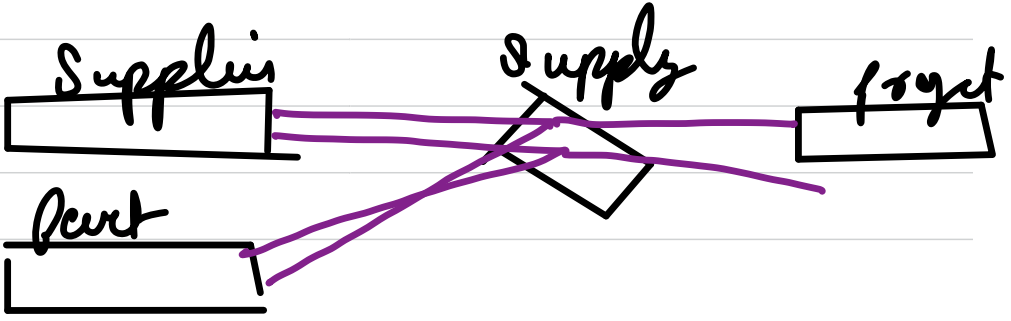


many-many

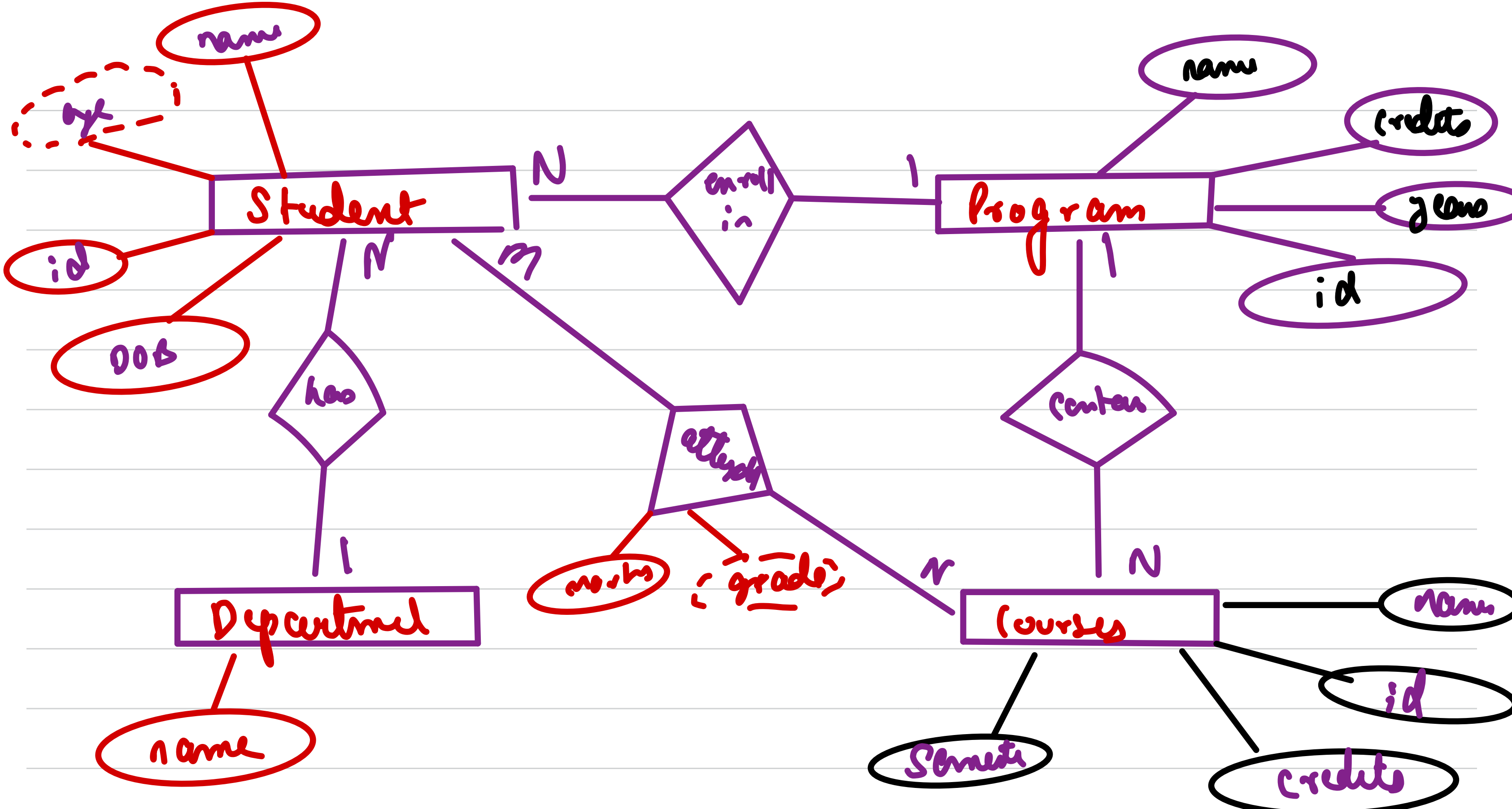
Unary Relationship



Binary Relationship

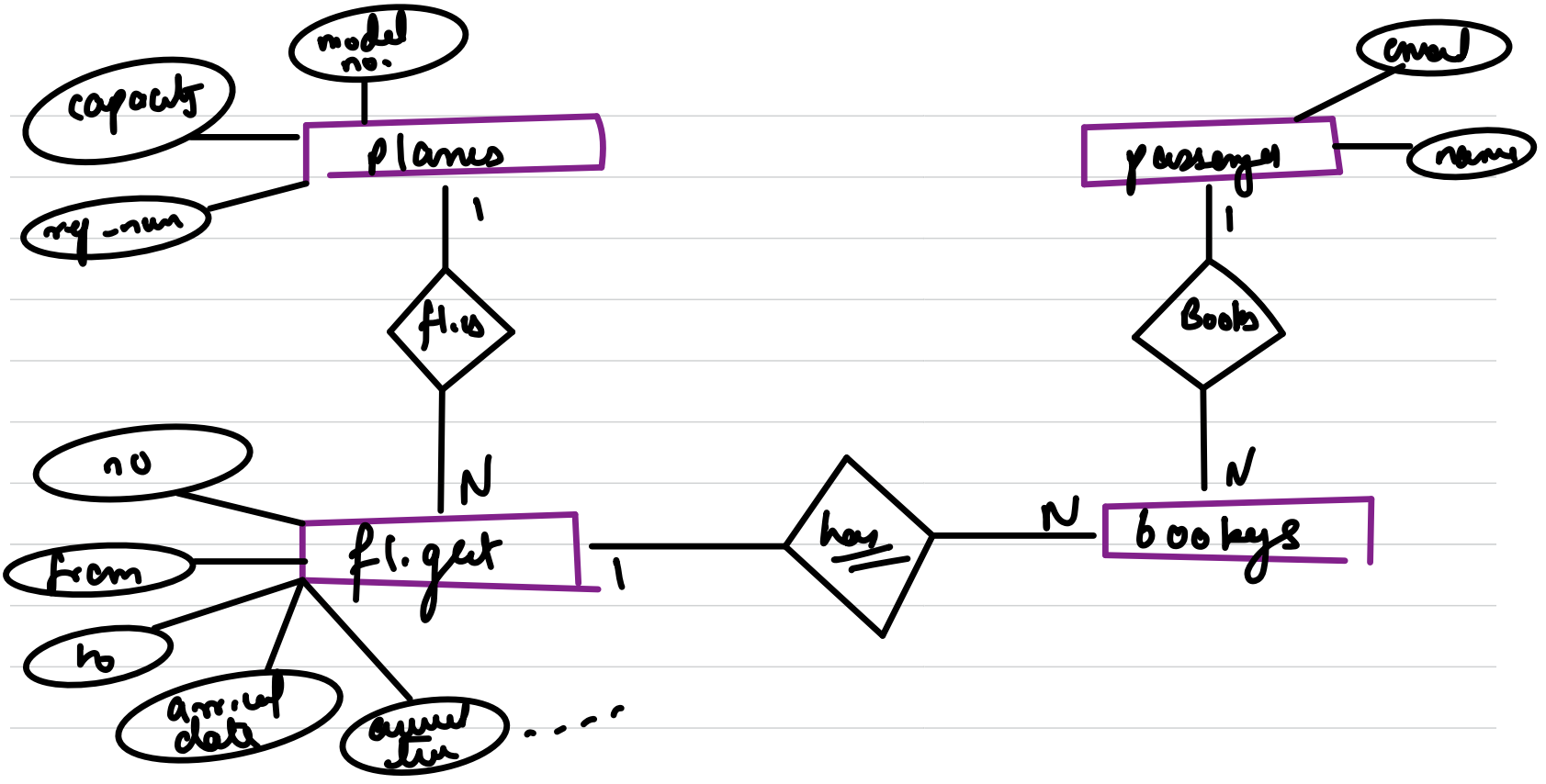


Try to create ER diagram of university db.



name
age
Student
id
DOB
N
M
has
1
Department
name
enroll in
N
1
Program
name
credits
id
1
contains
N
attends
marks
{grade}
N
Courses
name
id
Semester
credits

Draw an ER diagram of flight database →


model no.
capacity
planes
reg_num
1
N
H.is
no
From
fl. get
to
arrival date
1
has
N
N
Books
1
N
passengers
email
name
bookings

DBMS

Relational Data Model

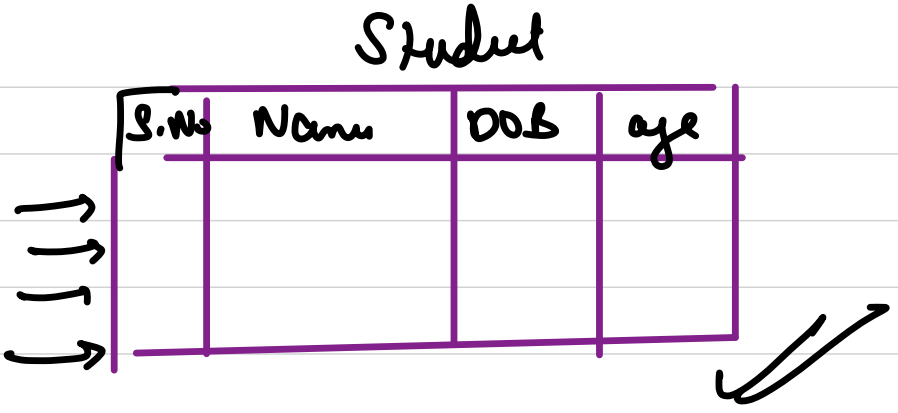
In actual entities are represented as tables & data as values (rows) and attributes of the entities are represented by columns.

RDBMS → MySql, PgSQL, Oracle db etc

columns → attributes
rows → Tuples

properties →

1) Each row is unique
2) We try to store as atomic value as possible.
3) Sequence of column is not relevant.
4) Every col has unique - name



→ RDBMS mostly support schemas

→ # Primary key → Single attribute to uniquely identify a record

Composite key →

Super key → It's a set of attributes that has the potential to uniquely identify a record

Candidate key → Minimal set of attributes from super key that uniquely identifies data.

alternate key → Candidate key other than primary key.

foreign key → a group of attributes that establishes link btw 2 tables.

Student

S_id	nam	phn
1	a	---
2	b	---

Cours

C-id	name	dept
1	c	1
2	d	2

Student_cours

S-id	C-id
1	1
1	2
2	1