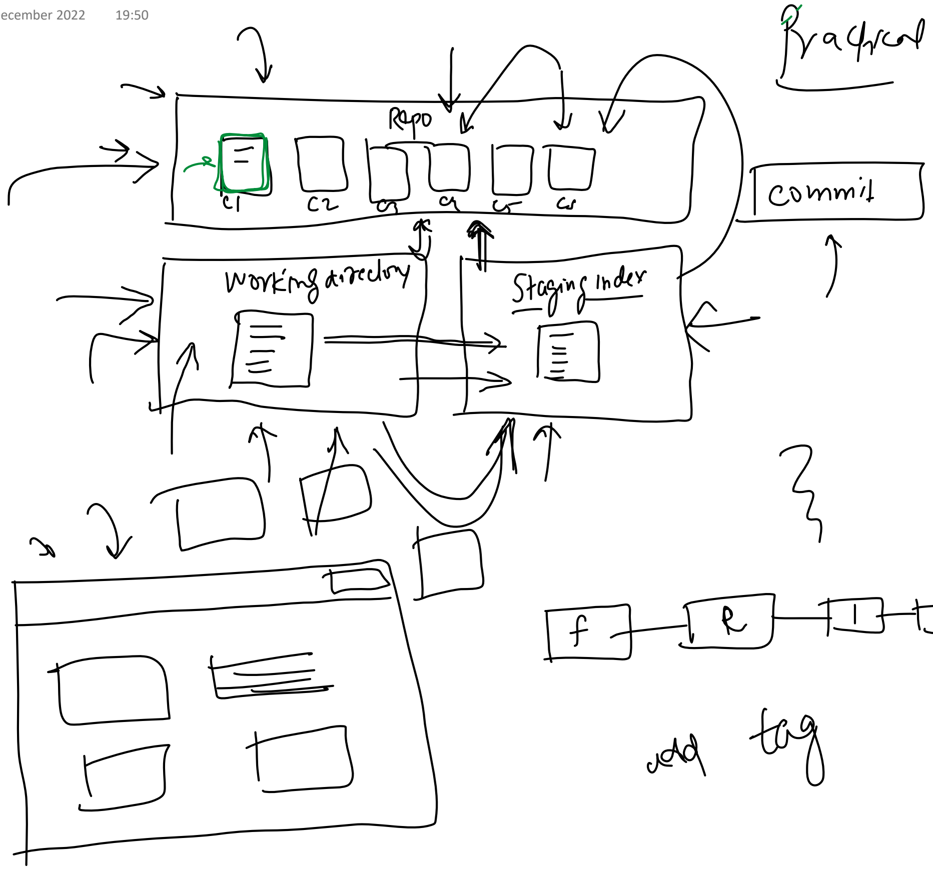
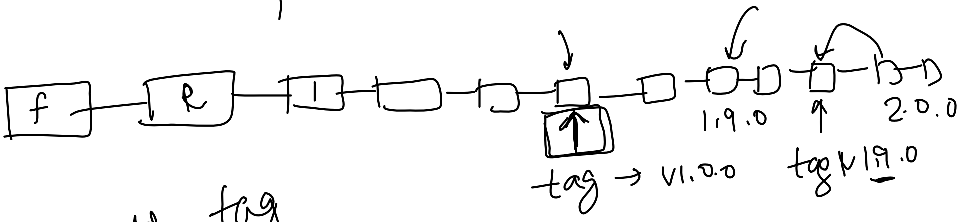
Git & Github

26 December 2022 19:50

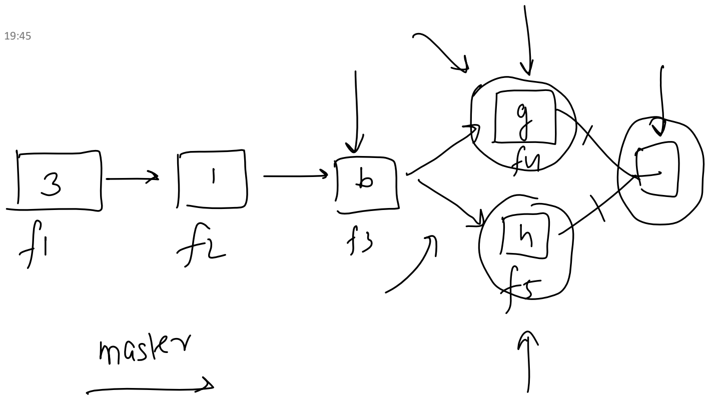
Practical

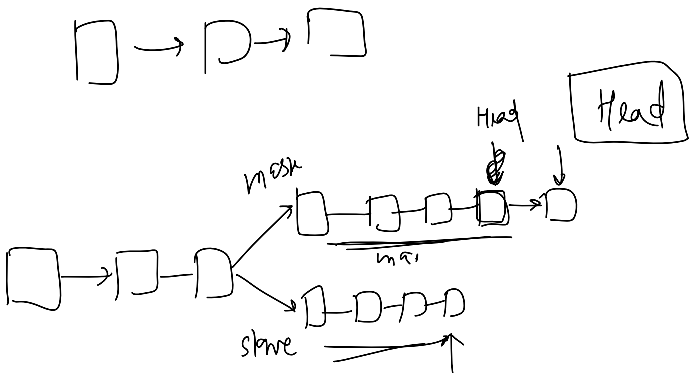
install git

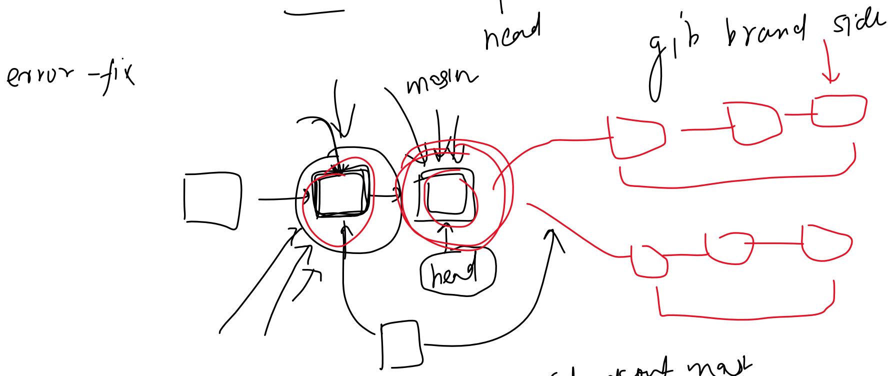


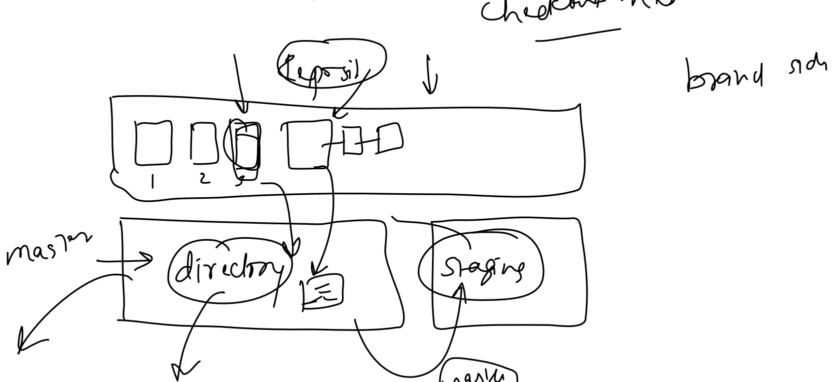


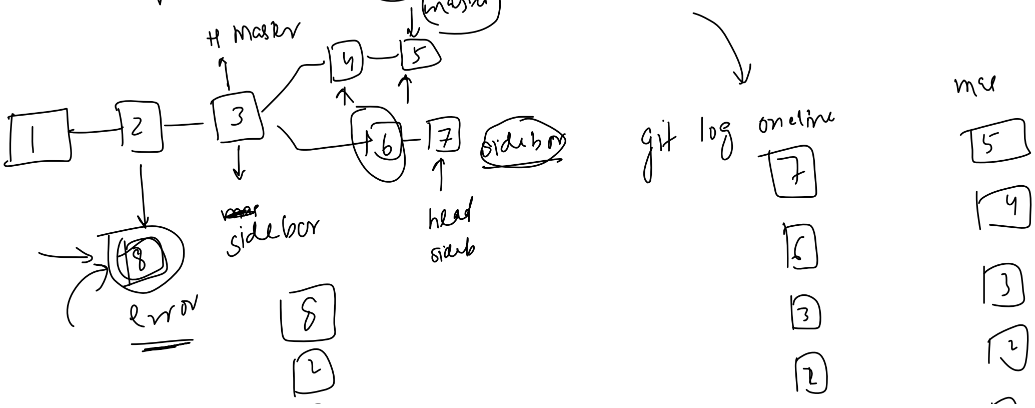
add tag

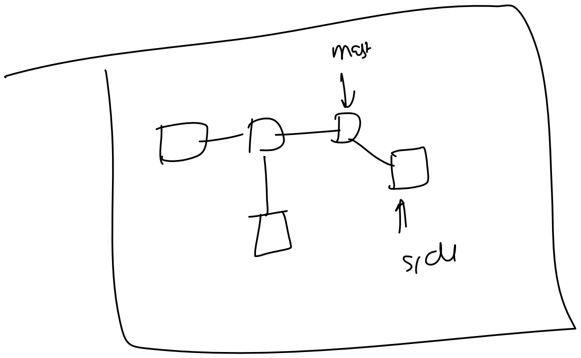

3
1
b
g
h
f1
f2
f3
f4
f5
masker

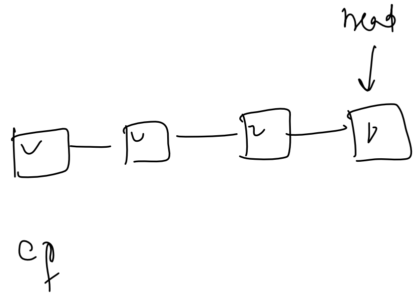

Head
Head
mesh
mas
slave
head

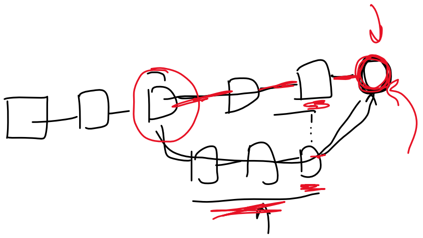

error -fix
mesin
git brand side
head


Checkout mast
Reposit
brand side
1
2
3
master
directory
staging

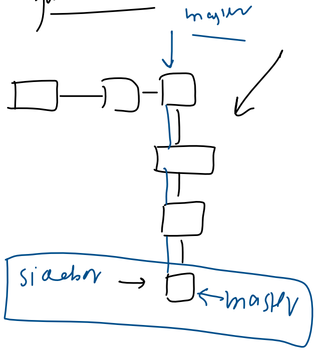

H master
master
4
5
1
2
3
6
7
sidebar
~~mas~~ sidebar
head
sideb
8
error
8
2
git log oneline
7
6
3
2
mer
5
4
3
2

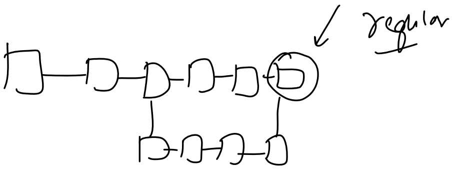


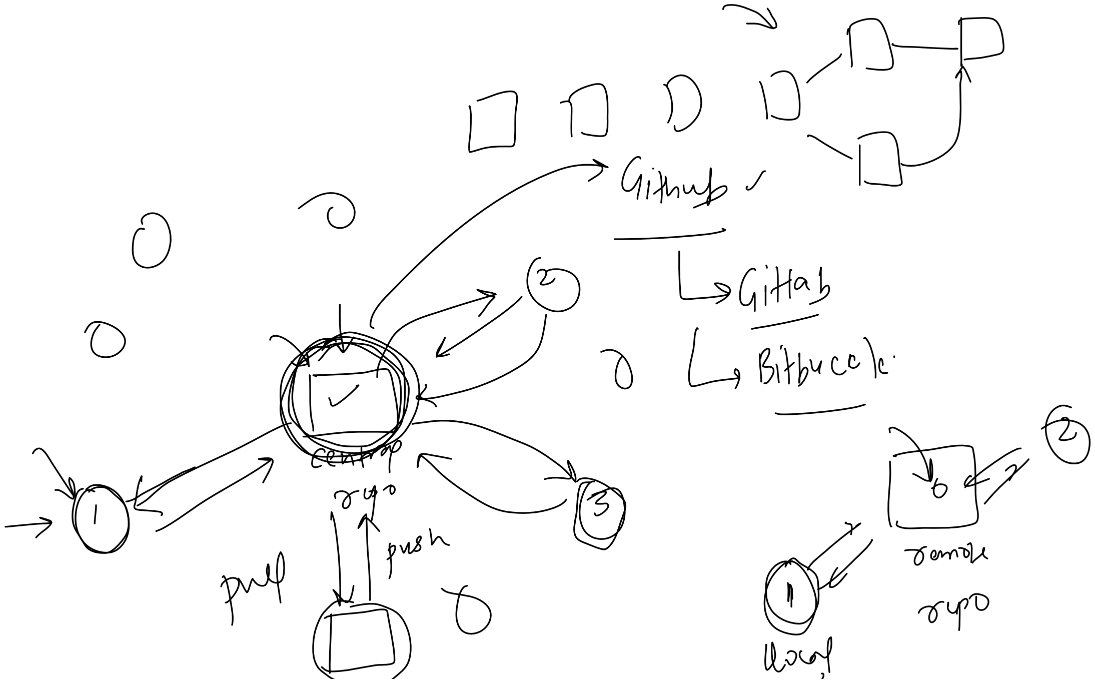


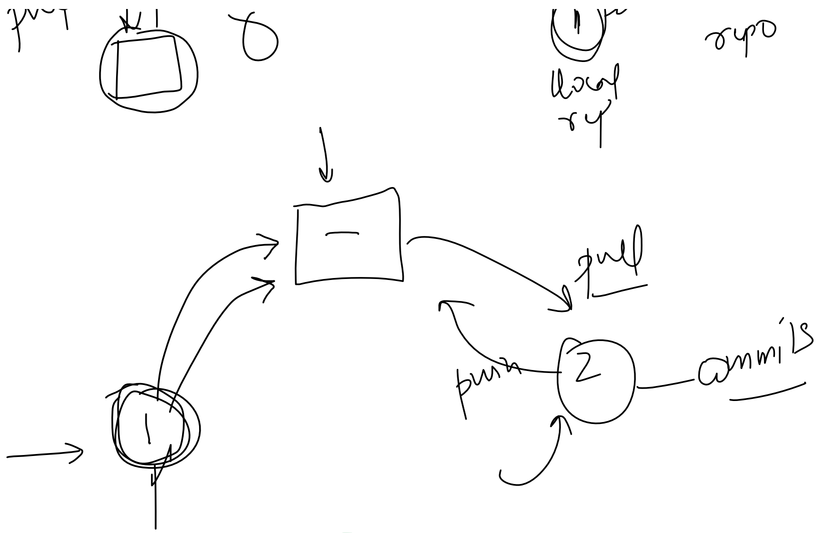


fast forward









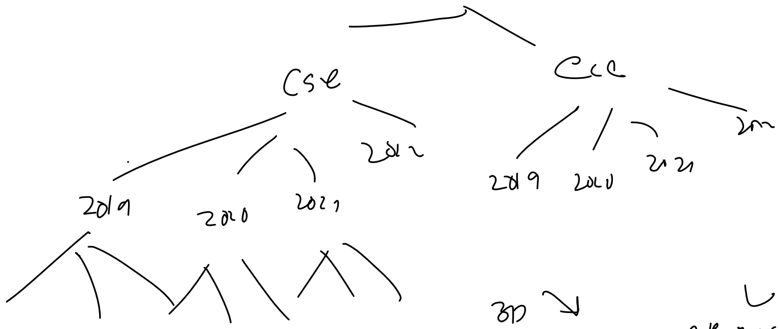
series (1D)

name	
nitish	70
ankit	80
rahul	90

Dataframe (2D)

name	wt	height
nil	—	—
ankit	100	—
rahul	—	—

branch	year	
cse	2019	120
ec	2020	140
:		160
ece	2019	200
ece	2020	



3D

index

3D → 2D

branch	year	avg pack	student
cse	2019	4.5	120
cse	2020	6.1	100
ece	2022	3.9	130

$2 \times 4 = 8 \rightarrow \boxed{8 \times 2}$

index	col1	col2	col3
0			
1			
2			
3			
4			

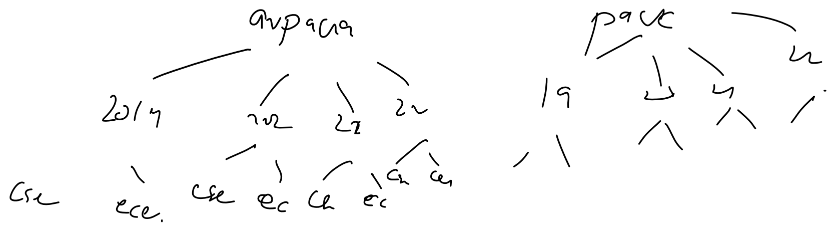
index + cols

MI Df → row

↓

MI Df → cols

df → series



~~fo~~ long

country	date	confirm	death
USA	3 Jan	3000	30
India	3 Jan	1000	10
India	2 Jan		

date	sum
6	42
7	37
8	89
9	101
.	
.	

→ ds

→ 6 — □ 37.5

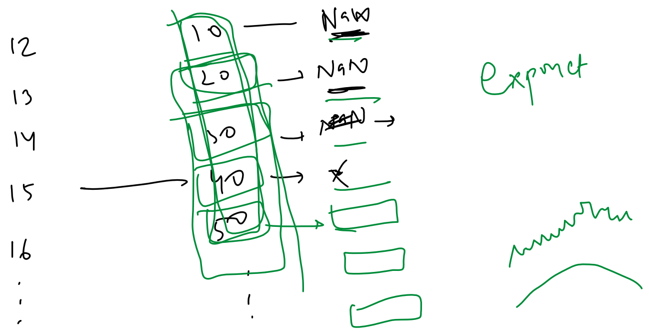
13 —

20 —

27 —

rolling avg → moving

=3 ⑤



reviews | sentiment

reviews

---

sentiment
+ve
-ve

model → new review
↓
+ve/-ve

→ 0) Data gathering ✓

✓ 1) Cleaning

✓ 2) Preprocess

- tokenization → [I, am, Nitish]
- stopword removal → and I have come
- stemming → dance danced dancing

dance dance dance

imdb → I am Nitish

@

night

textblob

#words

r1	32
r2	132
r3	64
r4	266

✓ 3) EDA →

✓ 4) make features

✓ 5) Vectorization → ML → numbers → BoW
↳ TfIdf
↳ Word2vec

6) modelling
7) Evaluation
8) Deploy
9) Monitor

bigrams

I am nitish my weight is 70

Unigrams

→ [I am] [am nitish] [nitish my] [my wt] [wt is] [is 70]

→ [I am Nitish] [am Nitish my] [nitish my wt] [my wt is] [wt is 70]

Bag of word

reviews → vocab

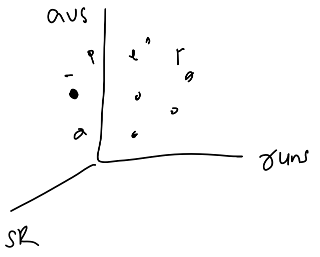
w1 w2 w3 w4
w3 w4 w5 w6
w1 w4 w6 w4

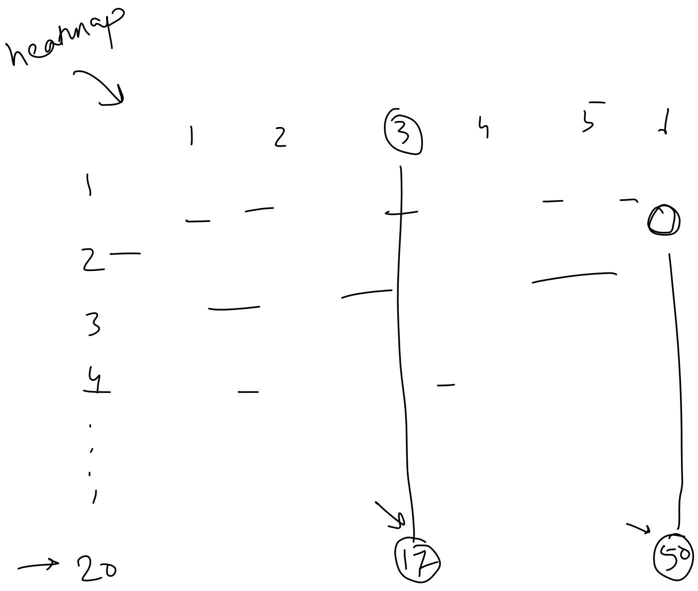
w1	w2	w3	w4	w5	w6
2	1	2	4	1	2

	w4	w3	w1	w6
r1	1	1	1	0

row ↘ 10000,5000

r1 →	1	1	1	0
r2 →	1	1	0	1
r3 →	2	0	1	1





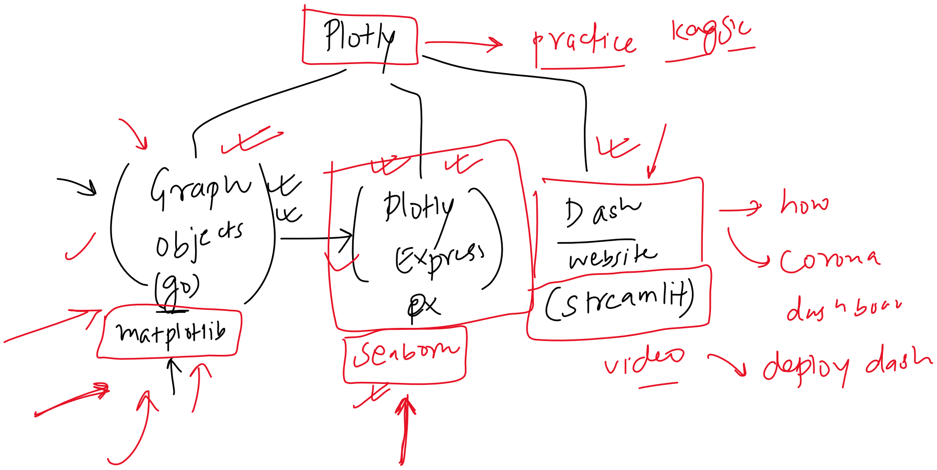
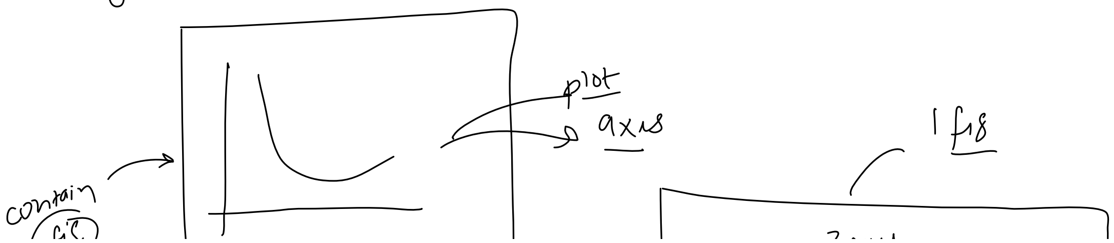
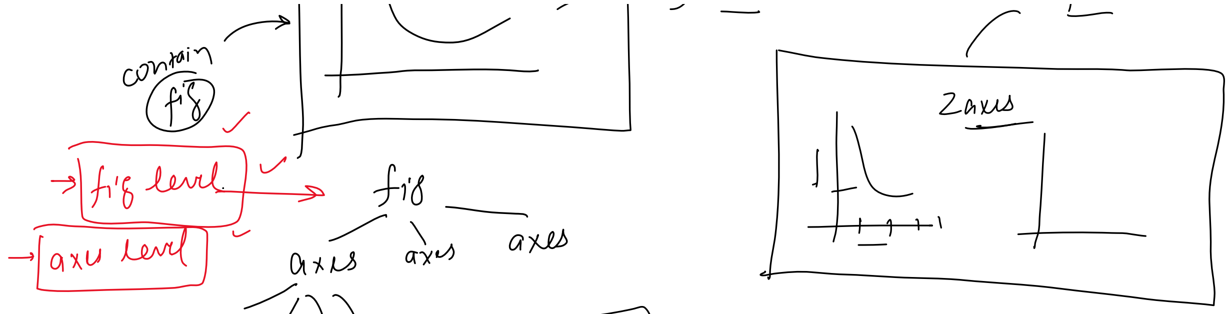
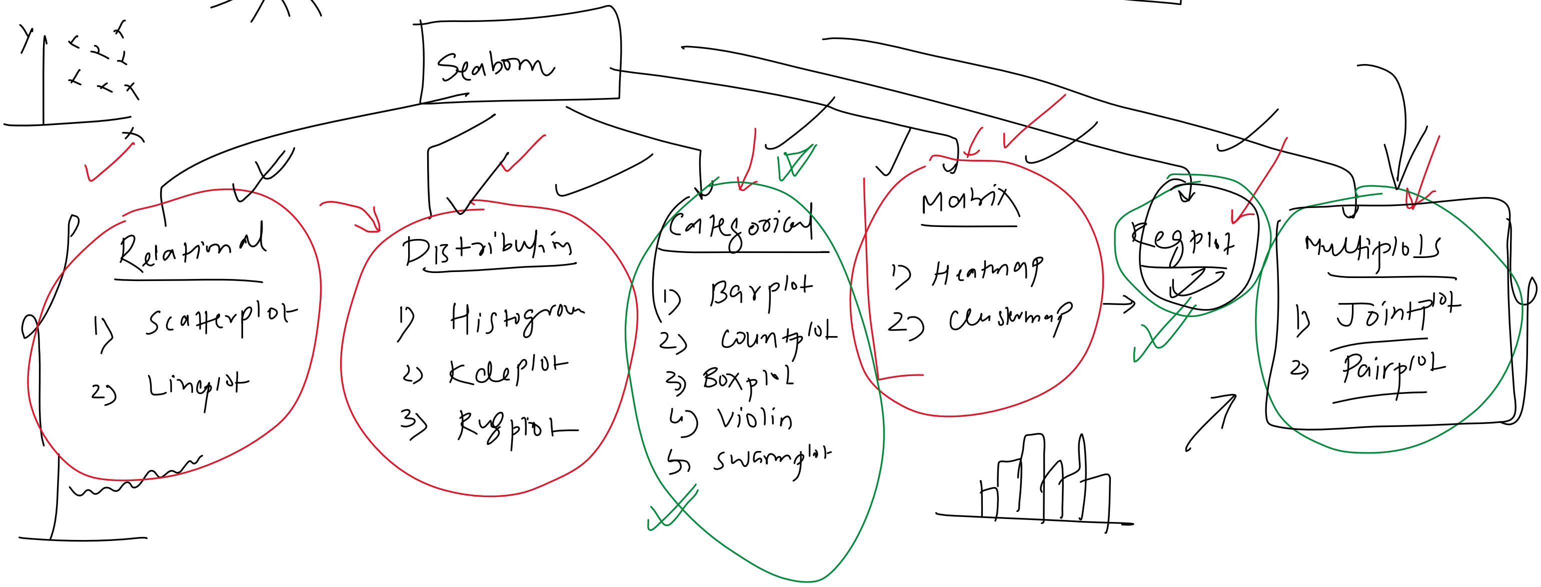


fig , axes

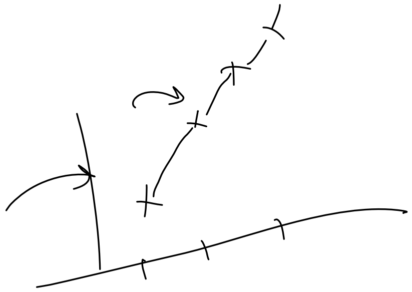




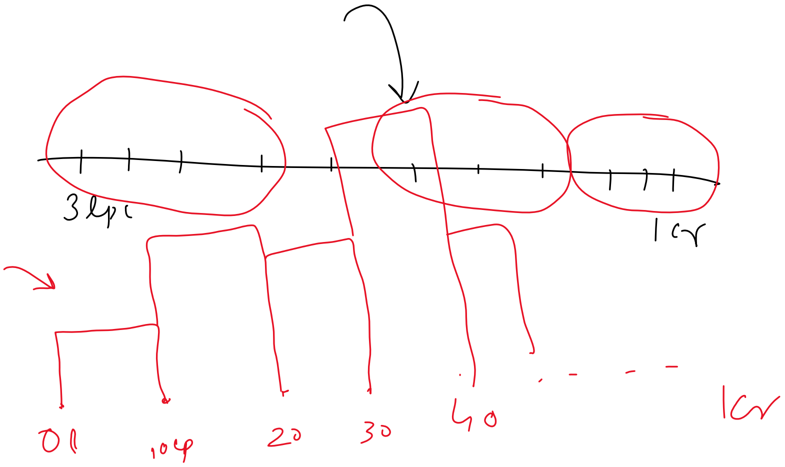


exp 4 5 6 7

sal 400 500· 6··



3lpa
4lpa
10 lpa
-
-
-
-

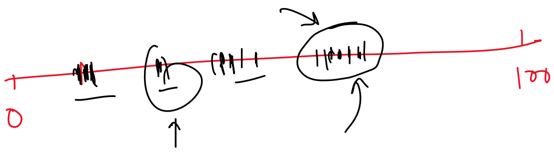


-3

4

5

15

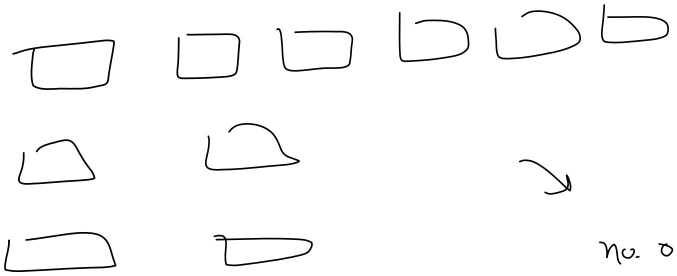


1952 57 62 67 . - - 2017

Afg

India

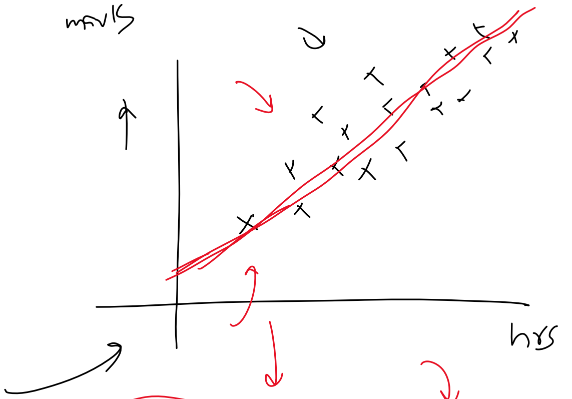
USA



no. of count × hr 0/ye

200 × 10 →

hrs of study (cm)	marks
80	6.5
100	7.5
200	8.1
30	6.6



$y = mx + b$

marks

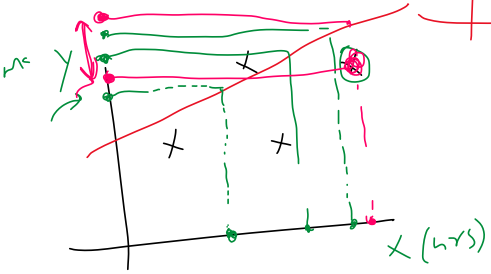
$\boxed{\text{marks}} = m \times \text{hrs} + \boxed{b}$

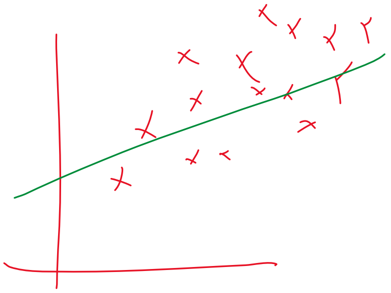
input → 75

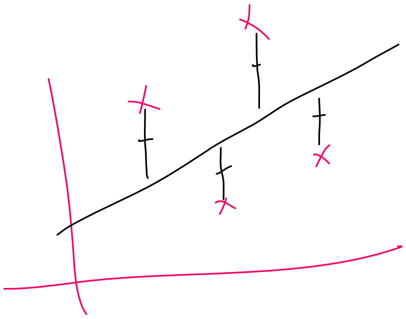
↓

marks

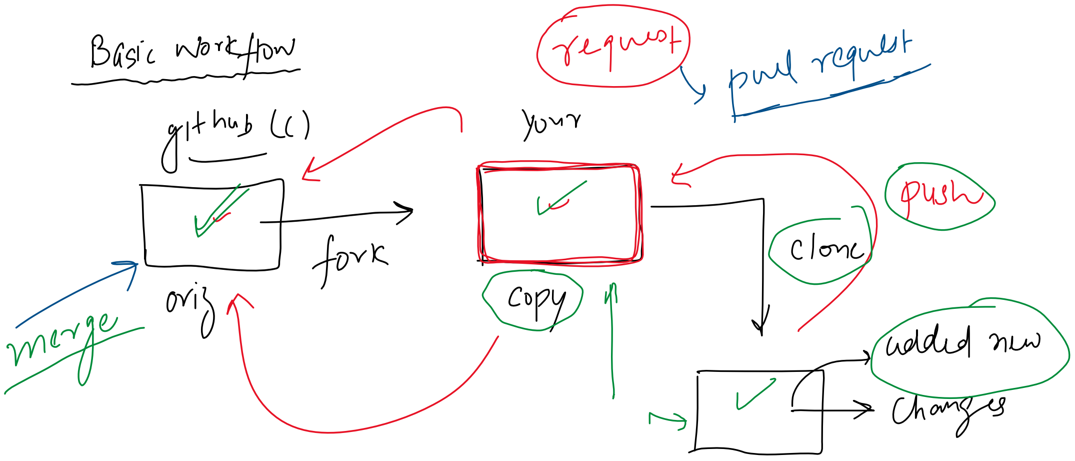
reg plot



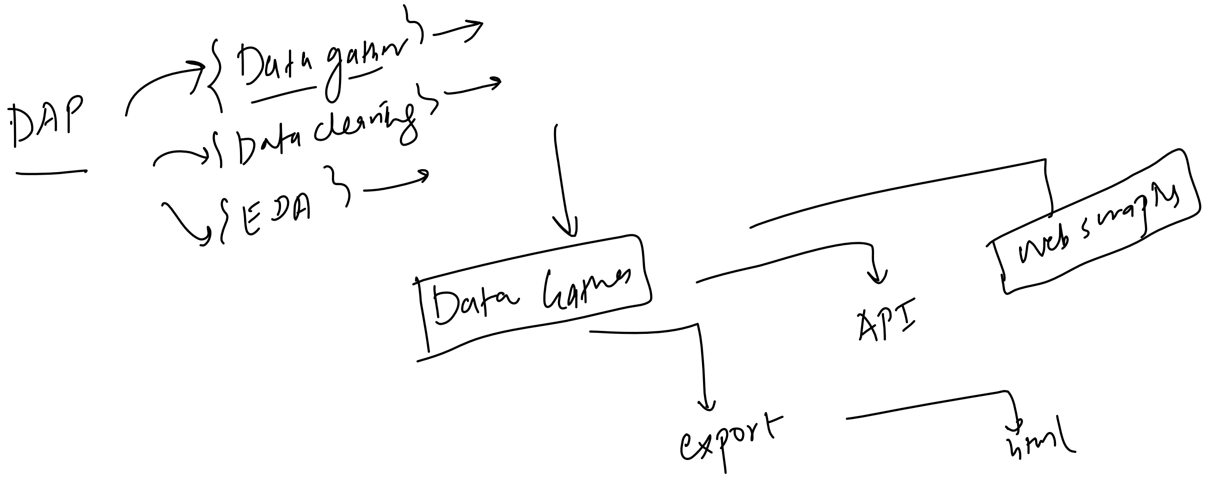


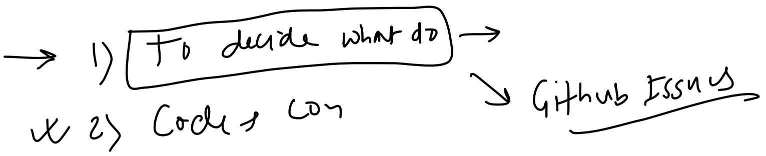


Basic workflow



23 January 2023 14:01





GitHub issues is a interface for asking questions about the project to a project maintainer in an open way that lets everyone see what's being done with the project.

"issues" doesn't mean that there's actually a bug, it can just be any change that needs to be made to the project.

Each issue can:

- have a label or multiple labels applied to it
- can be assigned to an individual
- can be assigned a milestone (for example the issue will be resolved by the next major release)

One of the most important aspects of the issue tracker is that each issue can have its own comments, so a conversation can form around the issue.

Another thing that's nice about issues is:

- they let you subscribe to an issue so you'll be notified of new

Another thing that's nice about issues is:

- they let you subscribe to an issue so you'll be notified of new comments and code changes
- you can communicate back and forth with a project maintainer on a specific change

GitHub issues support Markdown.

You can also create your own issues. Make sure you write good descriptive issues.

Next, it's a good idea to look at the GitHub issues for the project

- look at the existing issues to see if one is similar to the change you want to contribute
- if necessary create a new issue
- communicate the changes you'd like to make to the project maintainer in the issue

Contribution.MD

When you start developing, commit all of your work on a topic branch:

- do not work on the master branch
- make sure to give the topic branch clear, descriptive name

As a general best practice for writing commits:

- make frequent, smaller commits
- use clear and descriptive commit messages
- update the README file, if necessary

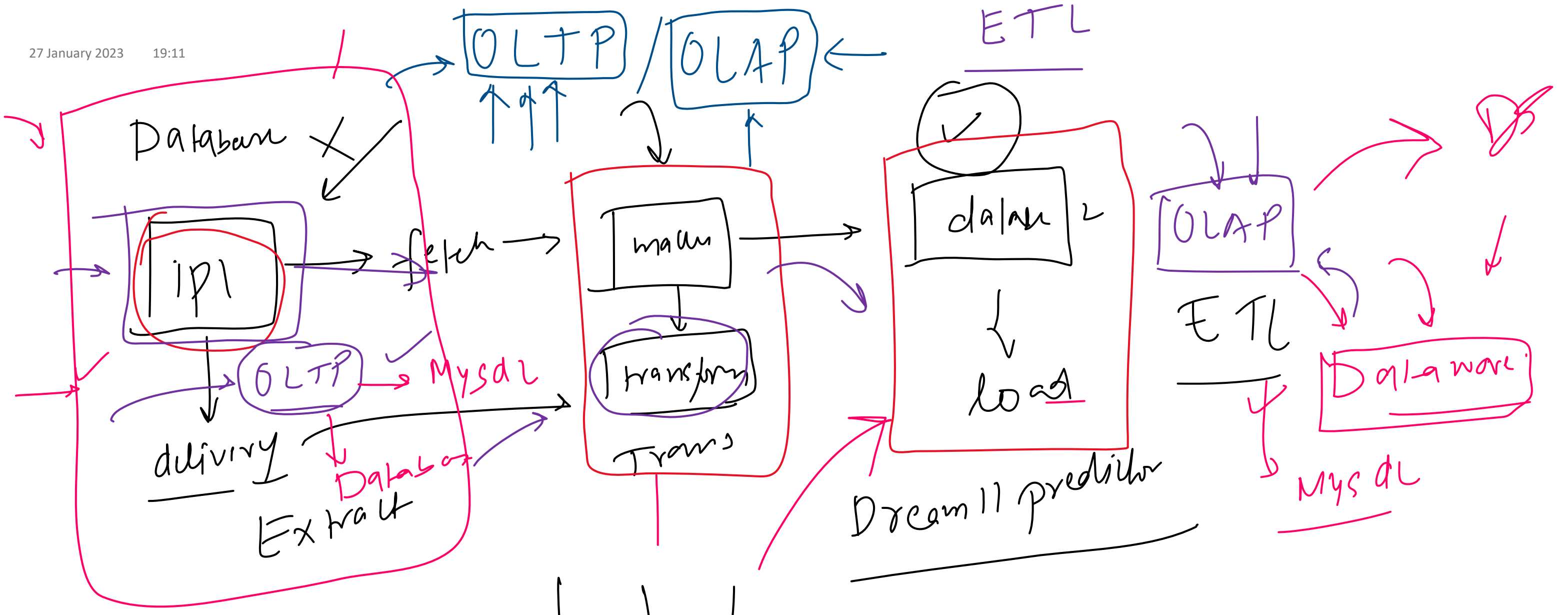
messy data

year	China	India	pakistan
1910	—	—	—
1970	—	—	—
1950	—	—	—
1990	—	—	—
2000	—	—	—

tidy data

year	country	popu
1910 —	china	—
1910 —	india	—
1960 —	pak	—

{
{
{

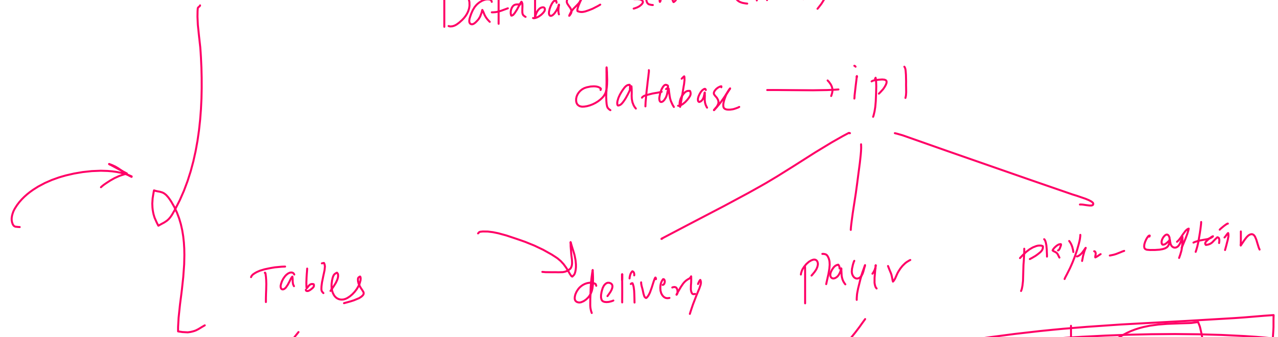


match	player	dream11 points
1	Kohli	57
1	Rohit	122
1	Bumrah	92
2	Dhoni	120

→ AWS

Database server (AWS) → database 1

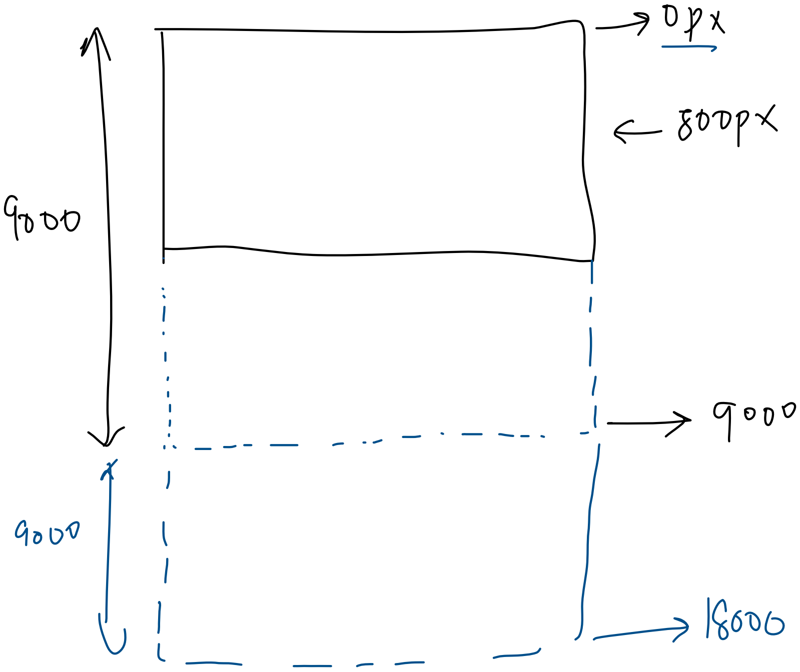
database → ipl



match id	player	runs	4s	6s	balls	iscaptain
1	Kohli	57	4	3	43	1
1	Rohit	63	5	2	47	0
		—	—	—		

Rohit

28 January 2023 19:49

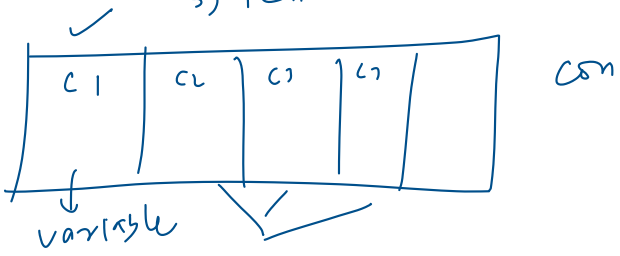


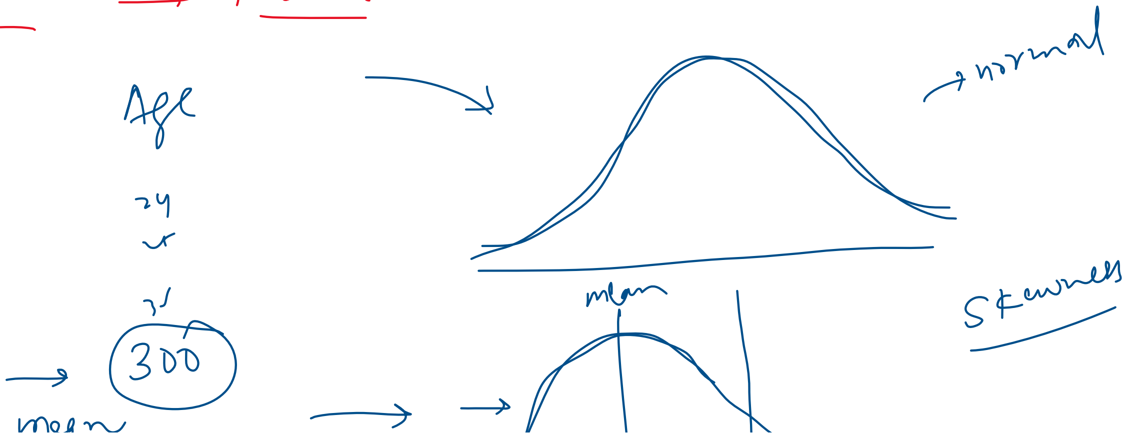
DAP ✓

1) Asking questions
2) Data wrangling
 ↳ Gathering
 ↳ Assessing ✓
 ↳ Clean
3) EDA → today
4) Conclusions
5) Presentation → Tableau

→ EDA

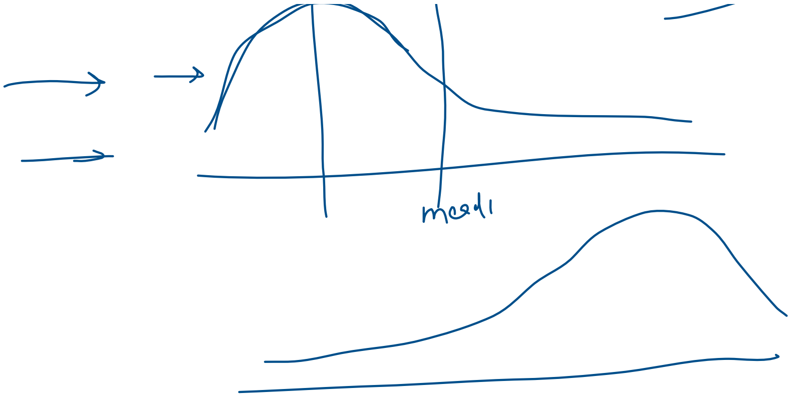
1) label col → num, cat, mixed
→ 2) Univariate
3) Bivariate
4) multivariate
5) Feature engg → outlier → missing





→ $\boxed{300}$

mean



03 February 2023 19:44

brand	~~samsung~~	apple	one
samsung	→ 0	0	
apple	→ 1	0	
oneplus	→ 0	1	

$$no = n \downarrow \boxed{n-1}$$

1. Before starting

06 February 2023 16:36

Python → Numpy → Pandas → Dat vz → DAP

↓

→ SQL → 9-feb

↑

→ SQL → run database

↓

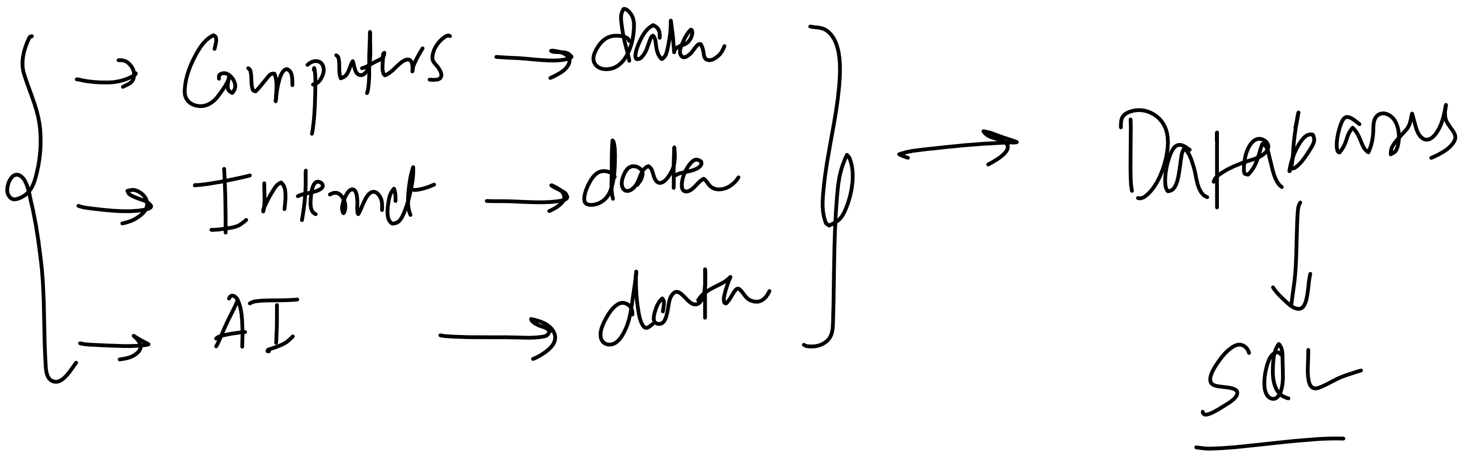
Databases

→ Practice

→ Data Analyst /Software

2. Importance of Data

06 February 2023 16:36



3. What are Databases?

06 February 2023 16:37

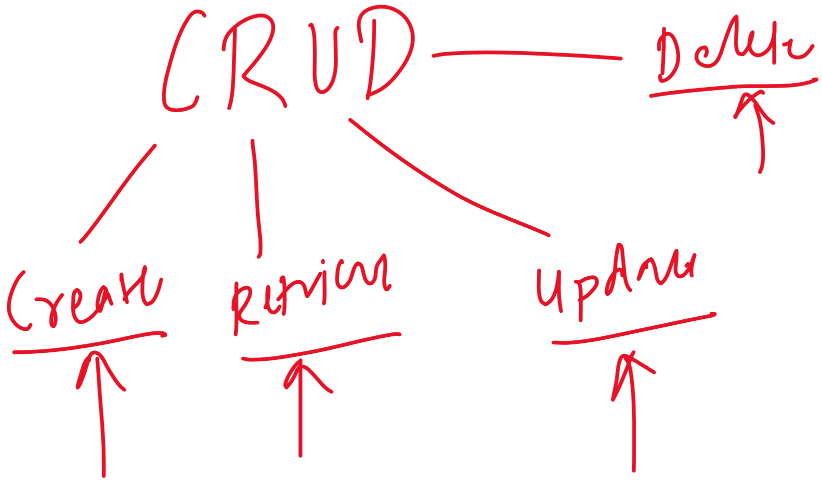
A Database is a shared collection of logically related data and description of these data, designed to meet the information needs of an organization

Data Storage: A database is used to store large amounts of structured data, making it easily accessible, searchable, and retrievable.

Data Analysis: A database can be used to perform complex data analysis, generate reports, and provide insights into the data.

Record Keeping: A database is often used to keep track of important records, such as financial transactions, customer information, and inventory levels.

Web Applications: Databases are an essential component of many web applications, providing dynamic content and user management.



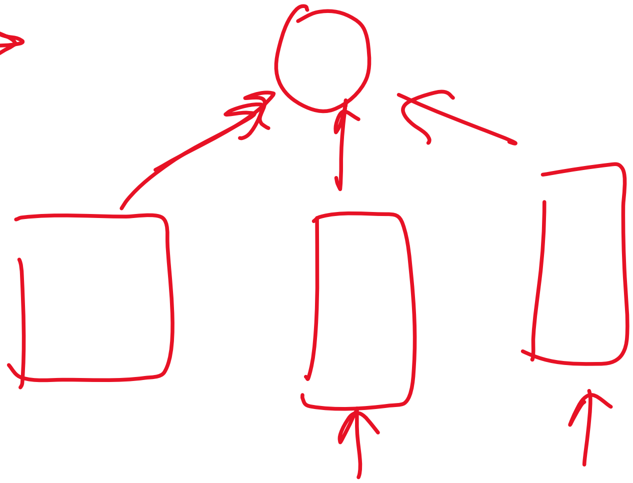
4. Properties of an Ideal Database

06 February 2023 16:37

accurate + consistency

1. Integrity
2. Availability
3. Security
4. Independent of Application
5. Concurrency

24x7

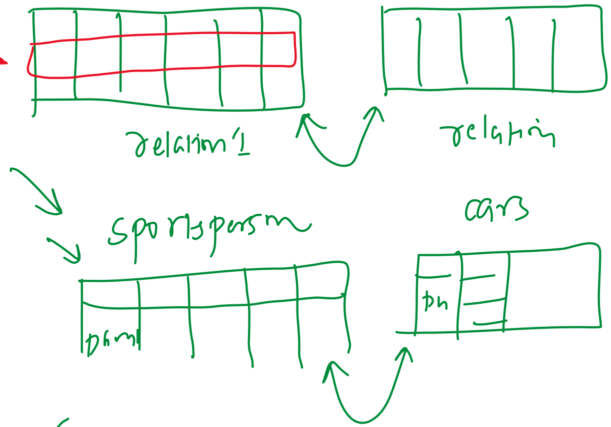


5. Types of Databases

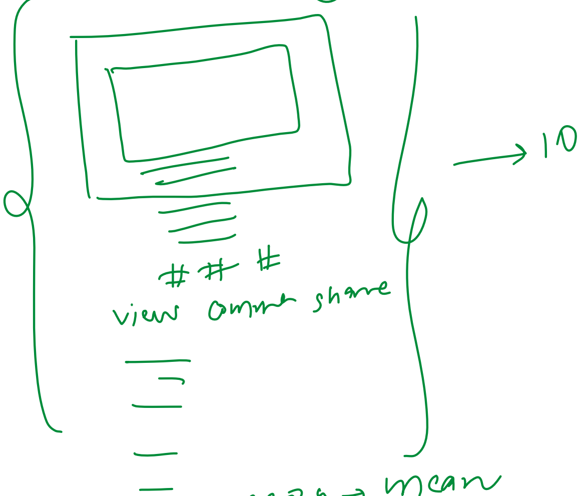
06 February 2023 16:42

sql server, oracle → MySQL, Postgre, Microsoft Access

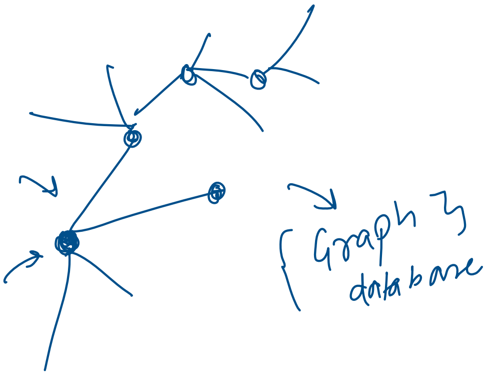
1. **Relational Databases -** (row)
   Also known as SQL databases, these databases use a relational model to organize data into tables with rows and columns.



2. **NoSQL Databases -**
   These databases are designed to handle large amounts of unstructured or semi-structured data, such as documents, images, or videos. (MongoDB)
3. **Column Databases -**
   These databases store data in columns rather than rows, making them well-suited for data warehousing and analytical applications. (Amazon Redshift, Google BigQuery)
4. **~~Graph~~ Databases -**
   These databases are used to store and query graph-structured data, such as social network connections or recommendation systems. (Neo4j, Amazon Neptune)
5. **~~Key~~-value databases -**
   These databases store data as a collection of keys and values, making them well-suited for caching and simple data storage needs (Redis and Amazon DynamoDB)

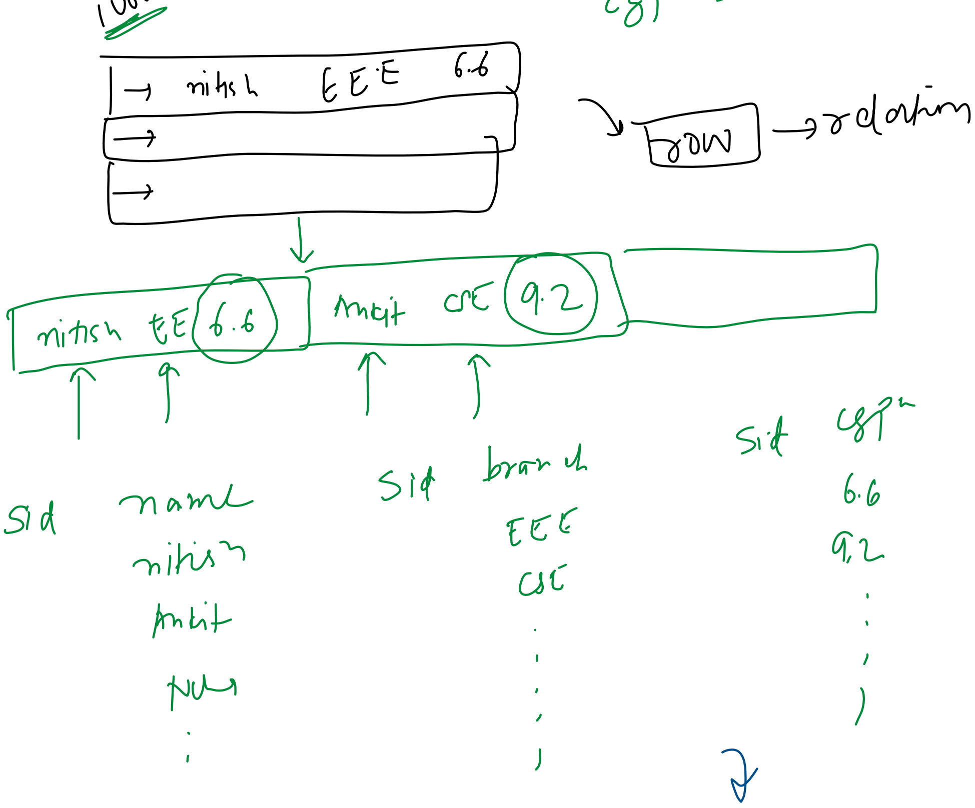


Which one should you use?

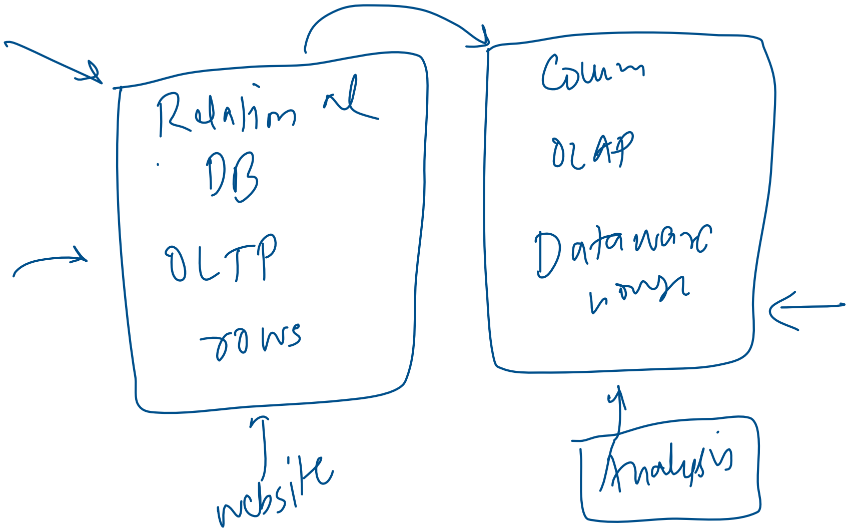


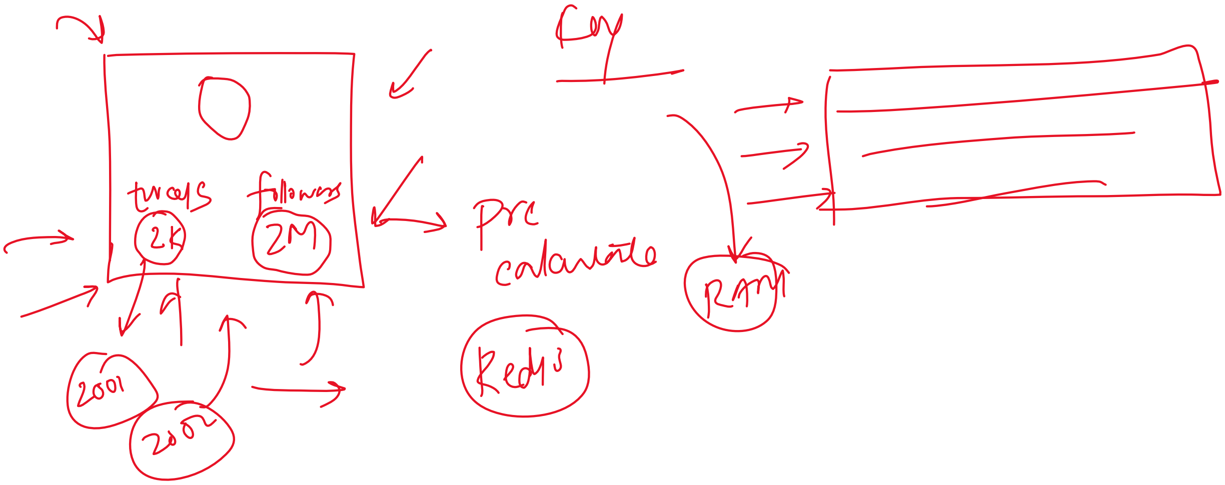
1000 students

cgpa → mean



nitish anni now . . EE CSE . . . - | 66 9,2 . . . .





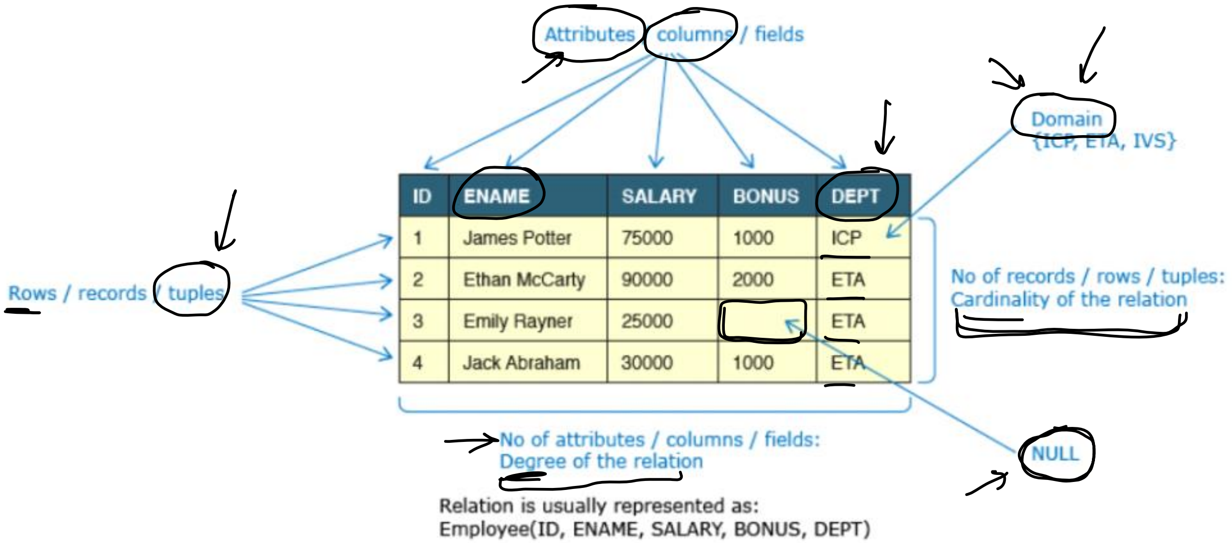
6. Relational Databases

06 February 2023 16:42

Also known as SQL databases, these databases use a relational model to organize data into tables with rows and columns.

table → relation

4 →


Attributes / columns / fields
Domain
{ICP, ETA, IVS}
Rows / records / tuples
No of records / rows / tuples:
Cardinality of the relation
NULL
No of attributes / columns / fields:
Degree of the relation

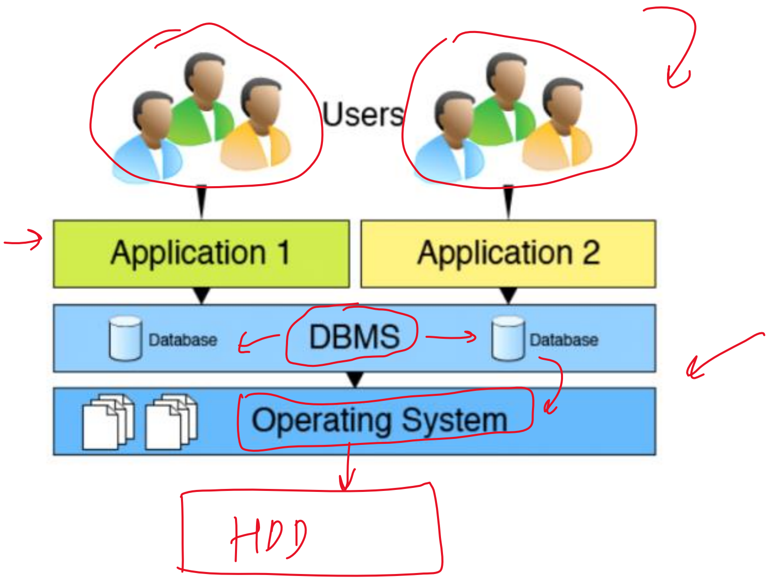
ID	ENAME	SALARY	BONUS	DEPT
1	James Potter	75000	1000	ICP
2	Ethan McCarty	90000	2000	ETA
3	Emily Rayner	25000		ETA
4	Jack Abraham	30000	1000	ETA

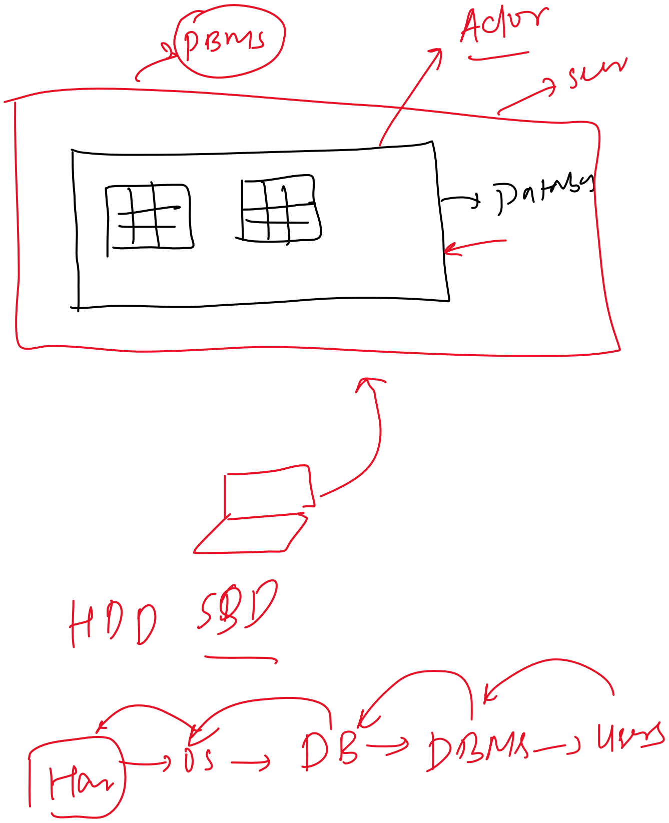
Relation is usually represented as:
Employee(ID, ENAME, SALARY, BONUS, DEPT)

7. What is a DBMS

06 February 2023 16:41

A database management system (DBMS) is a software system that provides the interfaces and tools needed to store, organize, and manage data in a database. A DBMS acts as an intermediary between the database and the applications or users that access the data stored in the database.





8. Core Functionalities of a DBMS

06 February 2023 16:41

Functions of DBMS

CRUD

Data Management - Store, retrieve and modify data

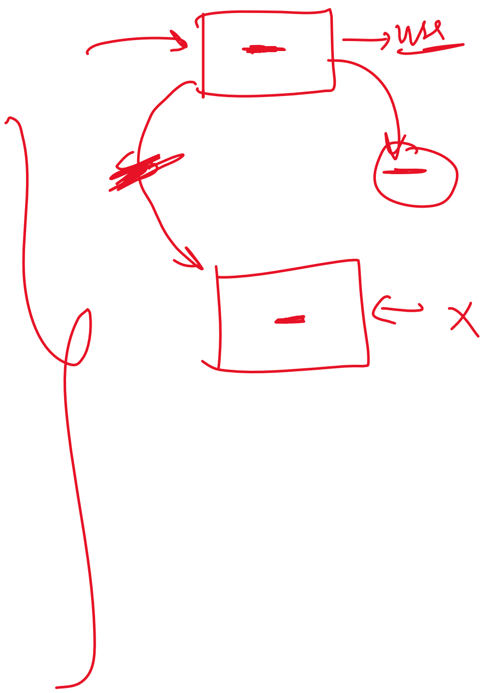
~~**Integrity** - Maintain accuracy of data~~

Concurrency - Simultaneous data access for multiple users

Transaction - Modification to database must either be successful or must not happen at all

Security - Access to authorized users only

Utilities - Data import/export, user management, backup, logging



9. Practical

06 February 2023 17:04

MySQL → MySQL Workbench

↳ Apachexamp

10. Database Keys

06 February 2023 17:07

A key in a database is an attribute or a set of attributes that uniquely identifies a tuple (row) in a table. Keys play a crucial role in ensuring the integrity and reliability of a database by enforcing unique constraints on the data and establishing relationships between tables.

1. **Super Key** - → janta
   A Super key is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table
2. **Candidate key** - → ummeadw
   A candidate key is a minimal Super key, meaning it has no redundant attributes. In other words, it's the smallest set of attributes that can be used to uniquely identify a tuple (row) in the table
3. **Primary Key** - vijeta
   A primary key is a unique identifier for each tuple in a table. There can only be one primary key in a table, and it cannot contain null values.
4. **Alternate Key** -
   An alternate key is a candidate key that is not used as the primary key.
5. **Composite Key** -
   A composite key is a primary key that is made up of two or more attributes. Composite keys are used when a single attribute is not sufficient to uniquely identify a tuple in a table.
6. **Surrogate Key** -
7. **Foreign Key** -
   A foreign key is a primary key from one table that is used to establish a relationship with another table.

Roll no	Name	Branch	Email
1	Nitish Singh	CSE	nitish@gmail.com
2	Ankit Sharma	EEE	ankit@gmail.com
3	Neha Verma	ME	neha@gmail.com

- roll no ✓
- email ✓ → roll no
- → roll + Name ✗
- → roll + branch ✗
- → roll + email ✗
- → Name + bran + email ✗
- → roll no + branc + name ✗
- → roll no + branch + name + email ✗

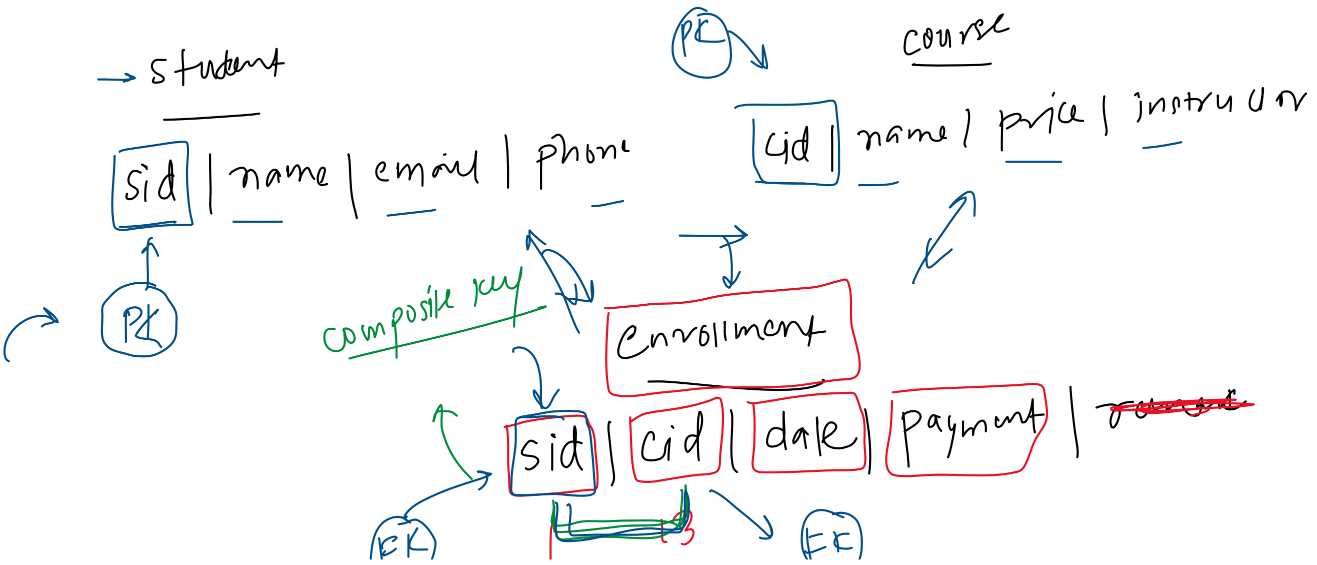
mando

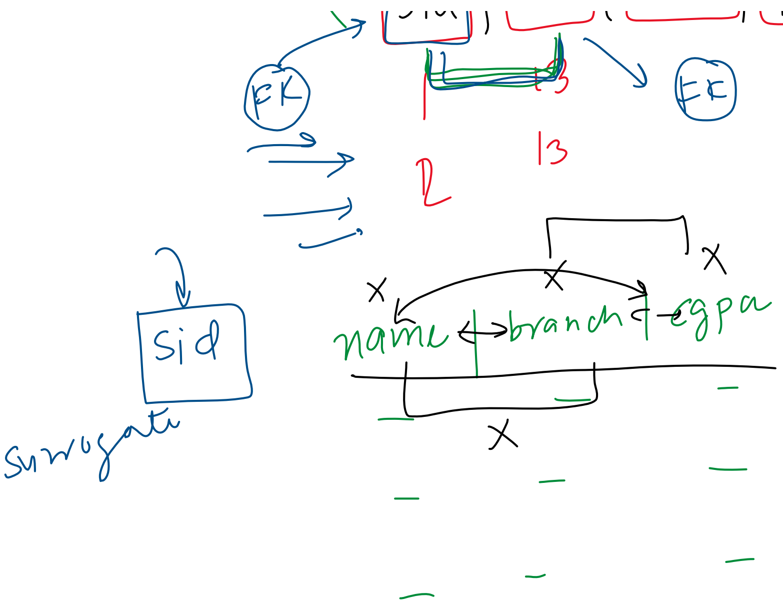
- → NULL →
- → repeat → duplicate →

good to have

- → numerical →
- → small →
- → constant

CK − PK = AK


PK
course
→ student
sid | name | email | phone
cid | name | price | instructor
PK
composite key
enrollment
sid | cid | date | payment |
FK
FK


FK
FK
13
13
2
sid
Surrogati
X
X
X
name
branch
cgpa
X

11. Cardinality of Relationships

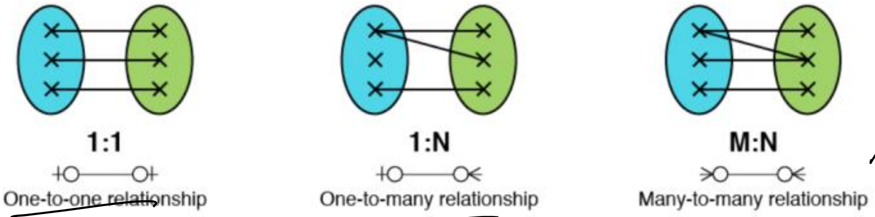
06 February 2023 16:43

entity

↓

table

Cardinality in database relationships refers to the number of occurrences of an entity in a relationship with another entity. Cardinality defines the number of instances of one entity that can be associated with a single instance of the related entity.



1 table

2tables

3tables

name	DLno

Examples

1. Person -> Driving License Number
2. Student -> college branch
3. Restaurants -> orders
4. Restaurants -> menu
5. Students -> courses

branch

bid	name	hod
1	cse	—
2	eee	—

Student

sid	name	bid FK
1	Nitish	1
2	Ankit	1

sid | name

cid | course | Dna

sid | cid | date

12. Drawbacks of Databases

06 February 2023 16:39

Complexity: Setting up and maintaining a database can be complex and time-consuming, especially for large and complex systems.

Cost: The cost of setting up and maintaining a database, including hardware, software, and personnel, can be high.

Scalability: As the amount of data stored in a database grows, it can become more difficult to manage, leading to performance and scalability issues.

Data Integrity: Ensuring the accuracy and consistency of data stored in a database can be a challenge, especially when multiple users are updating the data simultaneously.

Security: Securing a database from unauthorized access and protecting sensitive information can be difficult, especially with the increasing threat of cyber attacks.

Data Migration: Moving data from one database to another or upgrading to a new database can be a complex and time-consuming process.

Flexibility: The structure of a database is often rigid and inflexible, making it difficult to adapt to changing requirements or to accommodate new types of data.