

## Database fundamentals

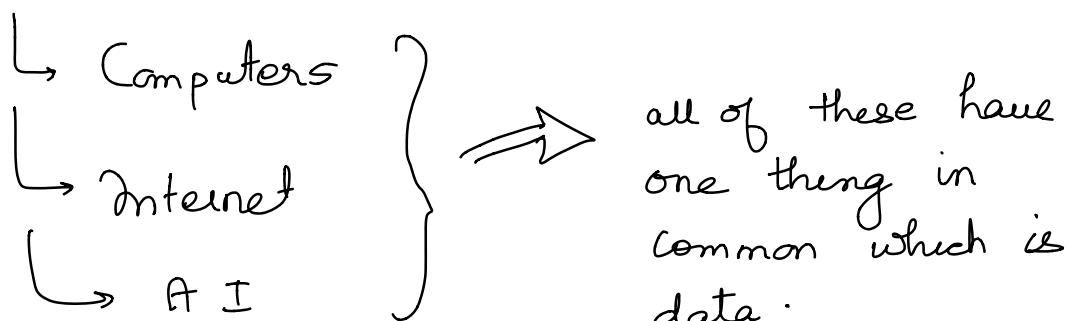
### ① { What is data? }

data is the new oil

जिसके लिए जब तक बड़े उड़ाने की जरूरत हो तब उसके लिए ज्यादा डेटा चाहिए, वह ज्यादा पैसा लगायेगा।

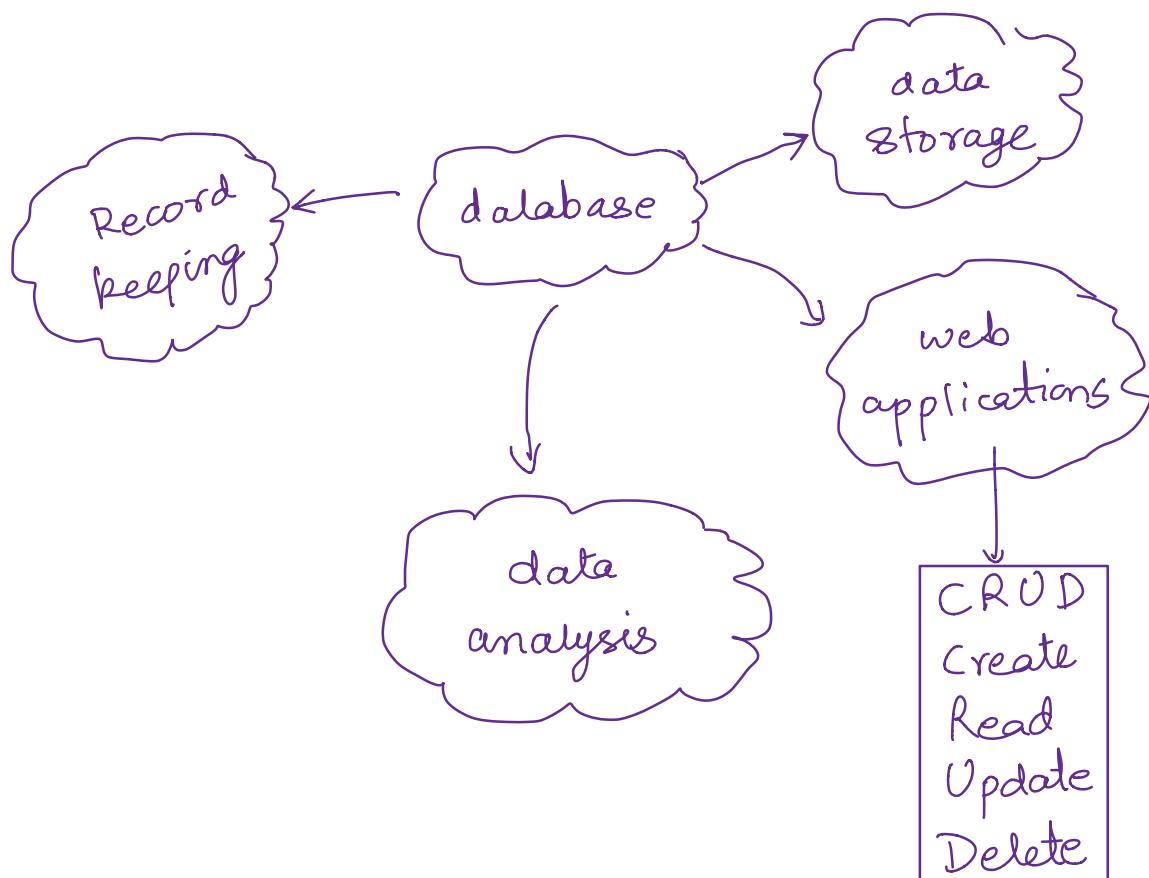
### ② { Revolution in Businesses }

The 3 major revolutions that has happened in the world are:-



### ③ { But what is a database? }

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



## ④ { properties of an ideal database }

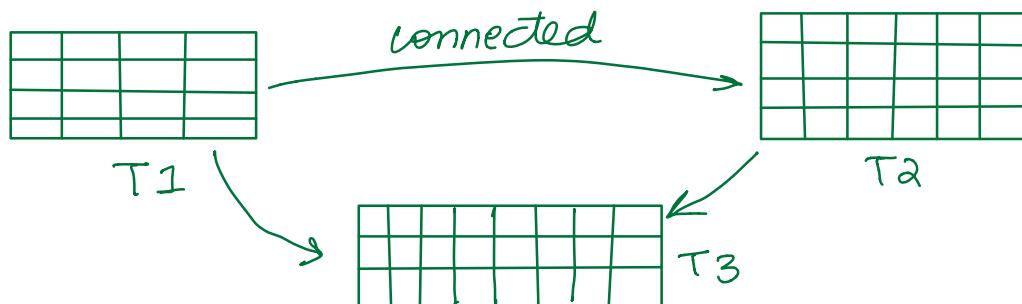
- ↳ Integrity
  - { accuracy + consistency }
- ↳ Availability
  - { A database must be accessible 24 x 7 }
- ↳ Security
  - { A database must be highly secure }
- ↳ Independent of application
  - { We must have same database for website as well as mobile application }
- ↳ Concurrency
  - { A database must be able to handle multiple concurrent user. It should serve data parallelly to be able }

to handle millions of users simultaneously }

## ⑤ { Types of databases }

### ↳ Relational databases

{ also called SQL database. Data is stored in the form of rows and columns }



### ↳ NoSQL databases { Not Only SQL databases }

{ These databases are used to handle large amounts of unstructured or semi-structured data such as documents, images or videos }

Examples of NoSQL databases : - MongoDB

### ↳ Column databases

{ These databases store data in columns rather than rows. Used for OLAP. used for analytic purposes by data analyst }

Some Examples are :- RedShift (By Amazon)  
BigQuery (By Google)

### ↳ Graph databases

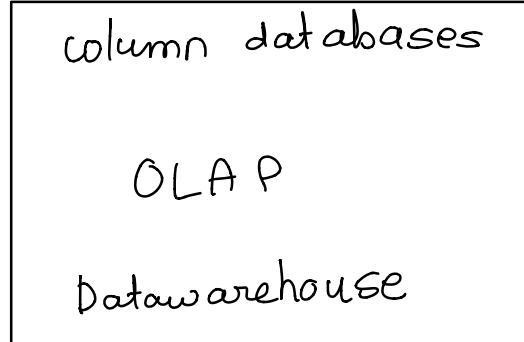
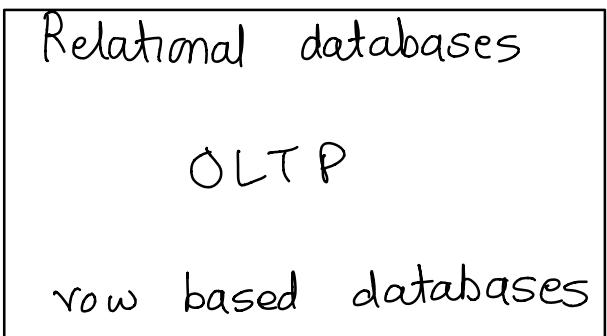
{ used for storing graph-structured data, such as social networking connections or recommendation systems }

Some examples are :- Neo4j, Amazon Neptune

### ↳ key-value databases

{ used for storing data as a collection of key-value pairs making them well suited for caching and simple data storage needs }

Some examples are :- Redis, Amazon DynamoDB



↑  
website

↑  
used for analytic  
purposes by data  
analyst.

## ⑥ { Relational Databases }

cardinality  $\triangleq$  # of rows

degree  $\triangleq$  # of columns

relation  $\triangleq$  table

rows are also called as records and tuples.

columns are also called as attributes and fields.

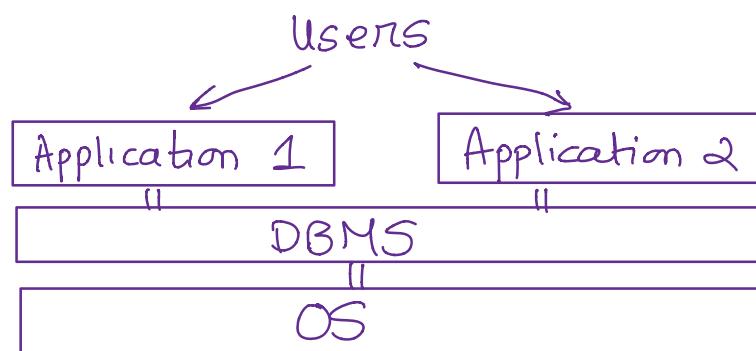
| sid | name | college | branch |
|-----|------|---------|--------|
| .   | .    | .       | .      |
| '   | .    | .       | .      |
| '   | .    | .       | .      |

A relation is represented as

Student ( sid, name, college  
, branch )

⑦ { What is DBMS ? }

A DBMS is a system software that provides the interfaces and tools needed to store, organize and manage data in a database.



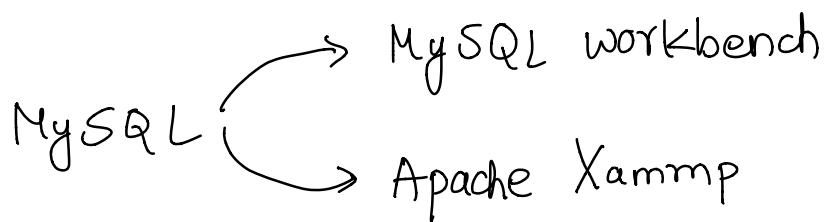
But OS ~~not~~ ~~not~~ Role ~~is~~ ?

At the end of the day, data to h HDD, SSD  
is stored there. That is the reason  
OS ~~not~~ ~~not~~ beech me.

## ③ { functions of DBMS }

- ↳ Data Management { Store, retrieve & modify data }
- ↳ Integrity { maintain accuracy & consistency of data }
- ↳ Concurrency { simultaneous data access for multiple users }
- ↳ Transaction { modification in the database must either be successful or must NOT happen at all }
- ↳ Security { Access must be given to authorized users only }
- ↳ Utilities { Data import / export, logging, backup, user management }

## ⑨ { Practical }



## ⑩ { Database Keys }

key  $\triangleq$  a column or a set of columns that uniquely identifies each 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.

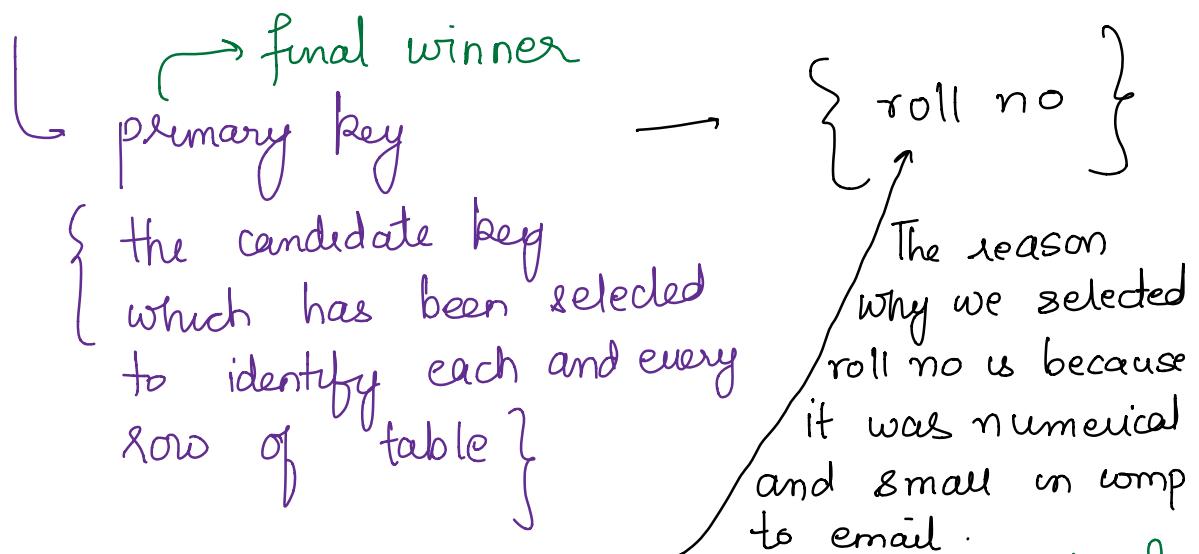
## ⑪ { Types of keys }

| roll no | name     | branch | email         |
|---------|----------|--------|---------------|
| 1       | abhushek | MCE    | samus@gmail-  |
| 2       | abhalash | EEE    | abhi46@gmail. |
| 3       | prachita | CA     | pra.56@gmail  |

↳ Super keys → {  
 roll no  
 roll no + name  
 roll no + branch  
 roll no + email  
 roll no + name + branch  
 roll no + name + email  
 roll no + email + branch  
 roll no + name + branch  
     + email  
 email  
 email + name  
 email + branch  
 ; email + name + branch

↳ candidate keys → {  
 { minimal    super }  
                     keys  
 ↴  
 which have minimum  
 # of columns

{ roll no  
 email



General criteria to select primary key?

- \* no NULL values
- \* no duplicate values
- \* numerical
- \* small

Good to have

↳ Alternate key → { email }

{ candidate keys which lost the battle to become pk }

↳ Composite key  
 { A pk having two or more attributes }

↓ Example

Students  $\longleftrightarrow$  Books

{ This is a many to many relationship  
i.e a student can buy many  
books and similarly a book can  
be sold to multiple students }

So we need to create 3 tables in these  
kind of cases - A table for students  
a table for books , and a third  
table which stores which student  
bought which book .

| sid | name | email | phone |
|-----|------|-------|-------|
|     |      |       |       |

| bid | name | price | author |
|-----|------|-------|--------|
|     |      |       |        |

| sid | bid | date | Payment |
|-----|-----|------|---------|
|     |     |      |         |

combination  
of sid and  
bid is pk.

← 3<sup>rd</sup>  
table



Thus it is a composite key.

### L Surrogate key

{ Sometimes we need to add an extra column in our table as no column or collection of columns qualifies as pk }.

{ In that case, this extra column is then used as pk. and called as surrogate key (sk) }.

### Example:-

| name | branch | college |
|------|--------|---------|
|      |        |         |

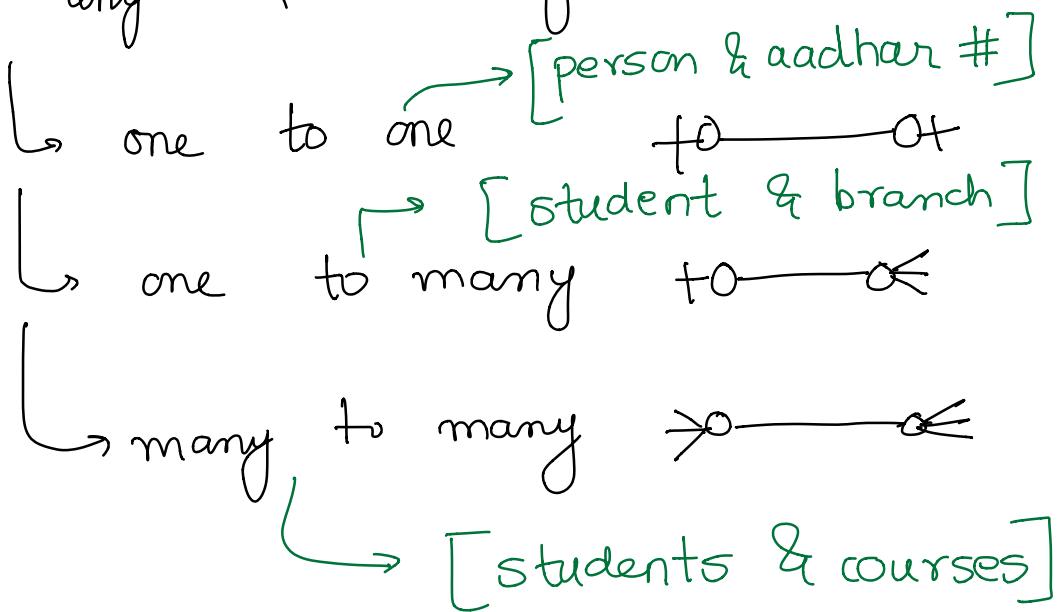
we need to add an extra column so that we have a pk.

↳ foreign key  
{ A fk is a pk of another table }

It is very very important to establish relationships between table.

## ⑫ { cardinality of relationship }

We have 3 kinds of relationship between any 2 real life entities.



for 1-1 relationship, only 1 table is needed

for 1-many relationship, 2 tables are req.

for many-many relationship, 3 tables are  
req.