


# DATABASES


→ types  
→ usecase


→ Pros & Cons  
→ examples

Database → collection of information or data in a computer, often managed using database management system DBMS.

→ Organized collection of data that can be organized and managed ~~easy~~ and accessed easily.

Eg  Theatre Mall

 Bank

 Hotel







Suppose we have diff building  
mall, hotel + bank

Peoples are like data

Buildings → are like database

Database

Data

Eg each building has diff char

Mall → large char  
water, beer, bags

} according to the service they provide.

Different database fulfil different requirement

## Types of database

Mainly 2 types (1) Relational (2) Non Relational (3) cloud db.

### (1) Relational Database

→ Ques On which factor you will decide to choose Rel<sup>n</sup> db

Ans Schema & acid properties

#### 1.1 SCHEMA IN RELATIONAL DATABASE

It defines how our data is organized and stored and their relationship between these tables. Complex data can be stored in table using relational database.



Eg we have 3 table Employee, department, account

id	name	age	acc-id	dep-id
1	Sayli	20	10	30
2	Ekta	21	20	40
3	Kanchan	22	30/null	50

Emp table

pk = id

foreign key -  
acc-id, dep-id  
that points to dep table

id	name	DOB	
30	Sayli		
40	Ekta		
50	Kan		

department table

pk = id

id	name	card
10	Sayli	
20	Ekta	
	Kan	

acc table

account id and dep id from employee table define  
that the employee is from which department &  
account

~~ITAEID~~ Proper Adv

① In emp table our schema constraints are  
that empid & depid cannot be null so  
if there is any null for an emp then that  
record would be removed  
Adv is → we donot get any garbage value  
in our table

② We can represent complex data in simple  
format using relational db

A → Atomicity  
 C → Consistency  
 I → Isolation  
 D → Durability

Atomicity → A transaction happens completely or it does not happen at all.

Eg

Account A	500
	↓
Account B	500

~~Bal~~ A Sends 500 to B  
 it shouldn't happen that money is deducted from A & didn't received to B

Transaction should get deducted from one account and get transferred to another account (All trans should be completed or do not happen at all)

Consistency

At any pt of time the state of data should be consistent

Eg Acc A → 500 Bal <sup>sec 1</sup>  
 Acc A → 600 Bal <sup>sec 2</sup>

while reading the balance it shouldn't read old data.

2 read operations should return same value at same time

Isolation

Bal = 500  
 write +100 = 600  
 read ⇒ 600

In actual the op<sup>n</sup> performed is 1st write & then read so

So after the order should be maintained  
 The read should not happen before write

Durability → it ensures all the operations that are happening are locked properly & all data is persistent.



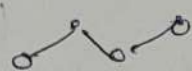
where does relational database  
→ can be used when no of col are fixed / min  
add<sup>n</sup> or subtraction  
→ when we don't know the no of columns in future  
needed to be add then use ~~not~~ non relational  
database

## ② NON RELATIONAL DB

it doesn't use labels (row, col)

→ key / value pairs

→ graphs



→ Document...

① key value. Eg DynamoDB Reddis

→ a unique key is paired with each collection of value.

Benefit → we don't have to worry about complex query bec system knows where the data is stored as our key is mapped

② Document Eg MongoDB

stores data in document Eg JSON

Each doc can have collect<sup>n</sup> & nested collect<sup>n</sup> of data

↳ { "id" : 123

name : "Ehts"

↳

JSON - Java script notation lang

Simple way of storing & exchanging data  
easy for human & machine to understand