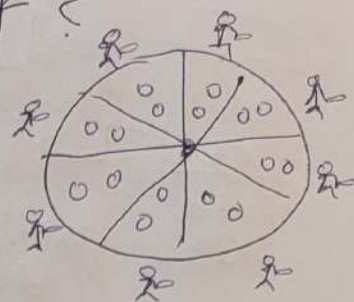


## 7) lecture ) What is Sharding?

eg we have a pizza and I cannot finish it completely. (8 slices)  
→ So I call my 7 friends to eat it



Sharding → method of distribution of data among across different machines

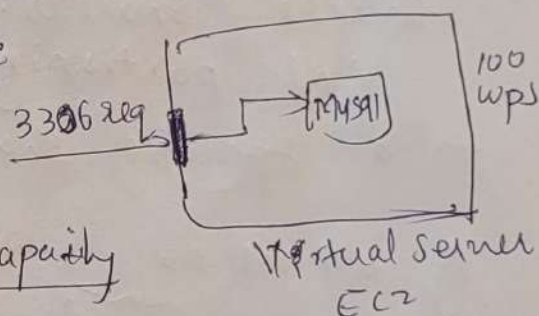
partitioning → splitting subset of instance within same machine.

How data is scaled? Horizontal & vertical.

eg we have a EC2 machine, there we install mysql server & use it

eg now user wants to access the data. so he will send request to EC2 port

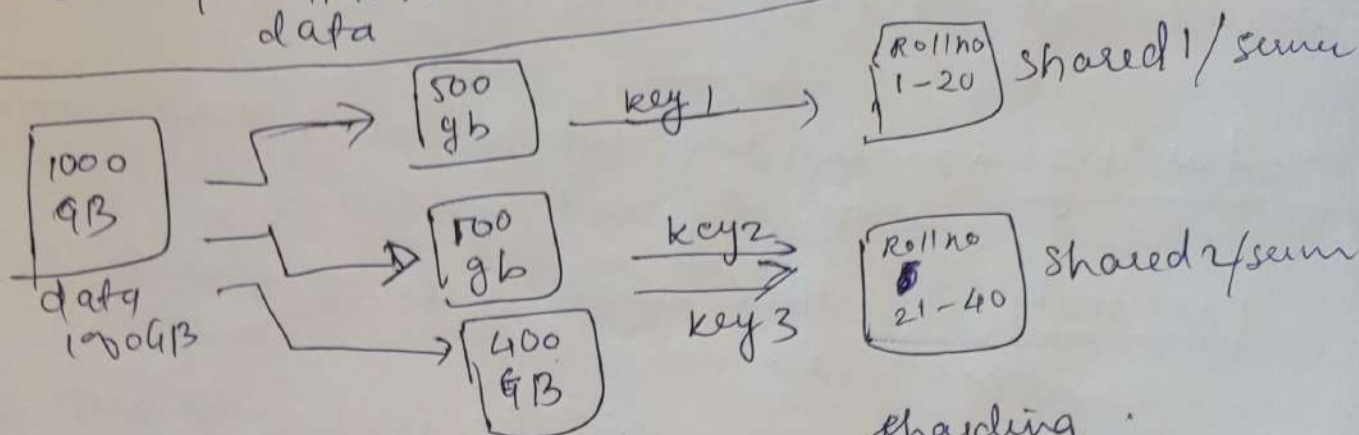
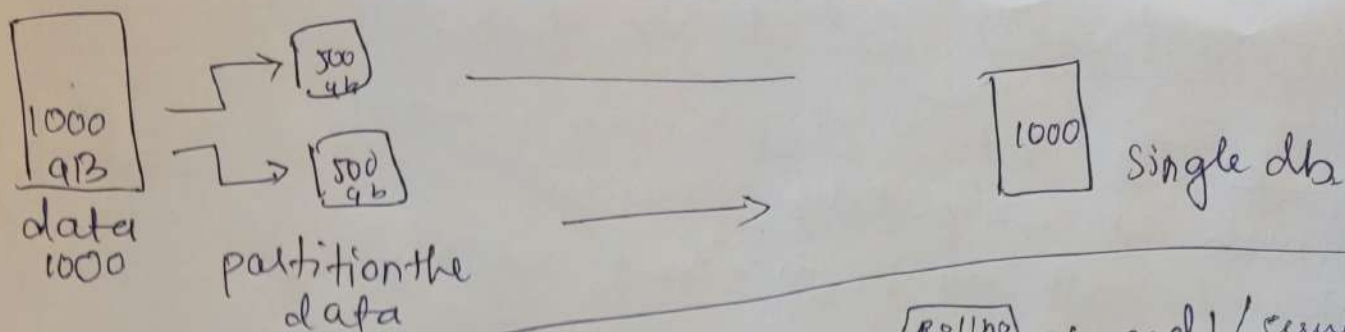
→ if no no read per sec & write per sec RPS & WPS since our actual server is not able to handle so we will inc the no of cpu capacity & RAM



But in Actual EC2 provides only 1000 wps max capacity. In that case we would add more servers.

→ This is nothing but sharding.

SHARDING - Split large db into smaller ~~chunks~~ chunks or db's / shards and use them across multiple servers.



sharding

Splitting the database

Sharding → partitioning database into multiple smaller chunks / shards.

And partitioning the data and mapping each data to a specific shard using specific key.

Eg all roll no 1-20 → are in shared 1  
21-40 → are in shared 2

key is used to map data to specific shard.

Here each shard / smaller server will have chunk of data.

Sharding should have Adv

consistency → ~~what~~ what data we put, we should read it

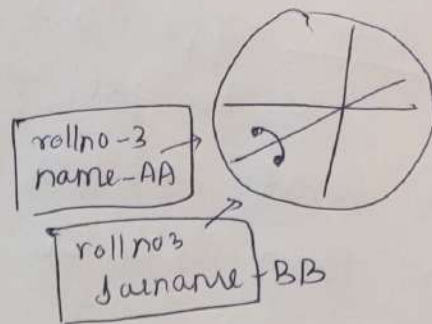
availability → should be available  
db should not crash

synchronization → if and updates done in data, it should be updated on that point



## Dis Adv

① Joins - if data is across 2 shards then query needs to go to 2 shards join them & then give output



② As we have already sharded our db so no. of shards are fixed.

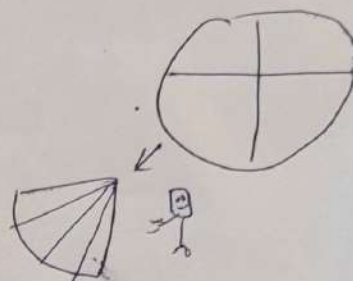
Shards are not flexible - as ek bar pizza slices kar diye to use kam ya jada nahi kar sakte

→ to overcome this, use consistent hashing to have our db flexible in no. of MEM CACHED

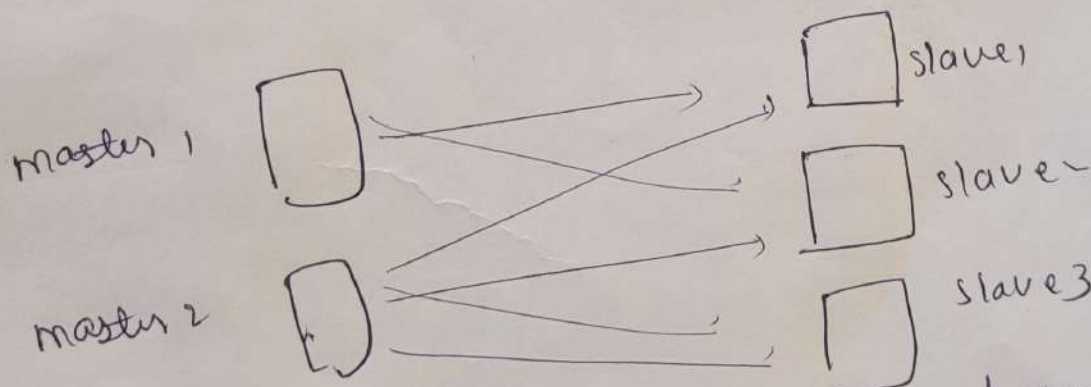
③ we have fix no. of shards → to overcome this

~~we~~ use hierarchical sharding

each shard again split into smaller chunks & there would be a manager who would help in mapping data to smaller chunks



Adv uses Master Slave → increases our performance



all write operations

all read operations

Some server/shard → will be master → perform only write oprn  
Some shard → will be slave → perform only read oprn  
in case all 2 masters are crashed/dead then one slave would be automatically act as master & vice versa.

Ques efficient way to query this db  
ie to get read write op fast in interview que

→ first answer → using Indexing

→ then answer → sharding

→ consistent hashing.