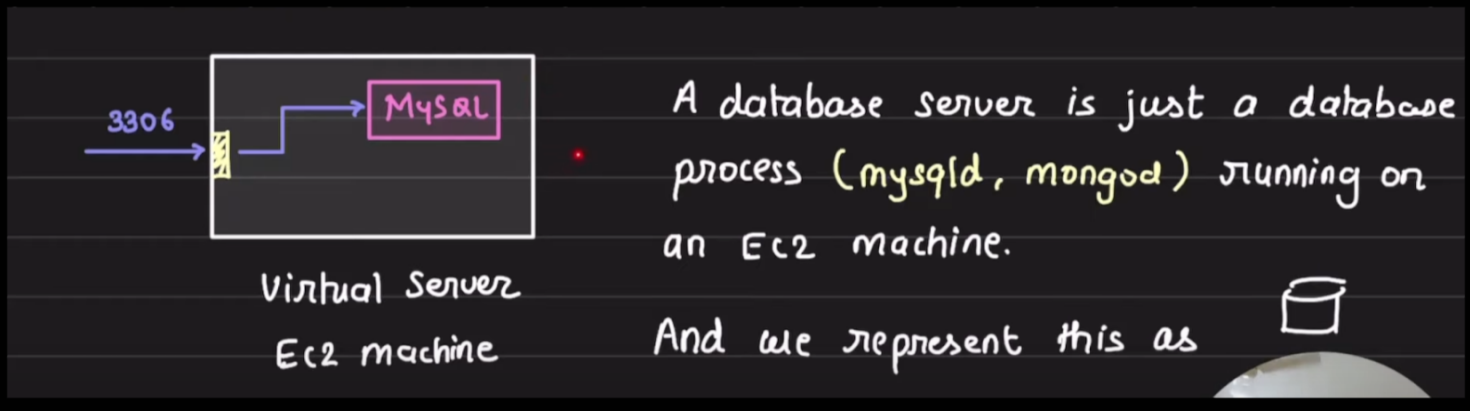
Database

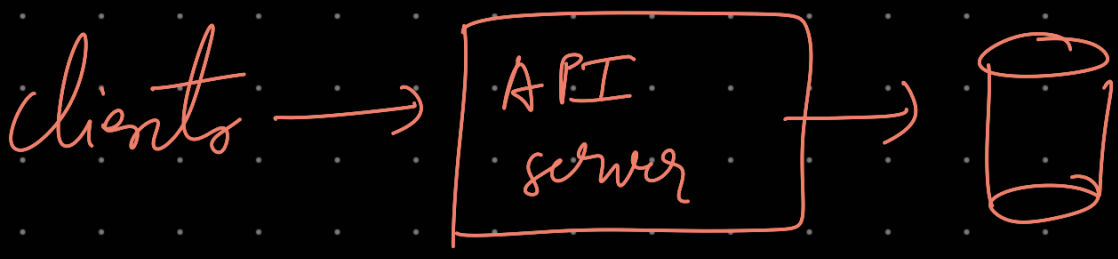
↳ It's just a process running on some port in a physical server Machine



⇓

Let say I wrote a S/W with a Database and now I put it on production using some cloud provider

⇓



100 write per second (WPS)

⇓

We started getting more users

⇓

Database metrics drops down

* CPU ↑ RAM ↑
* Query time increased.

⇓

Vertically scale

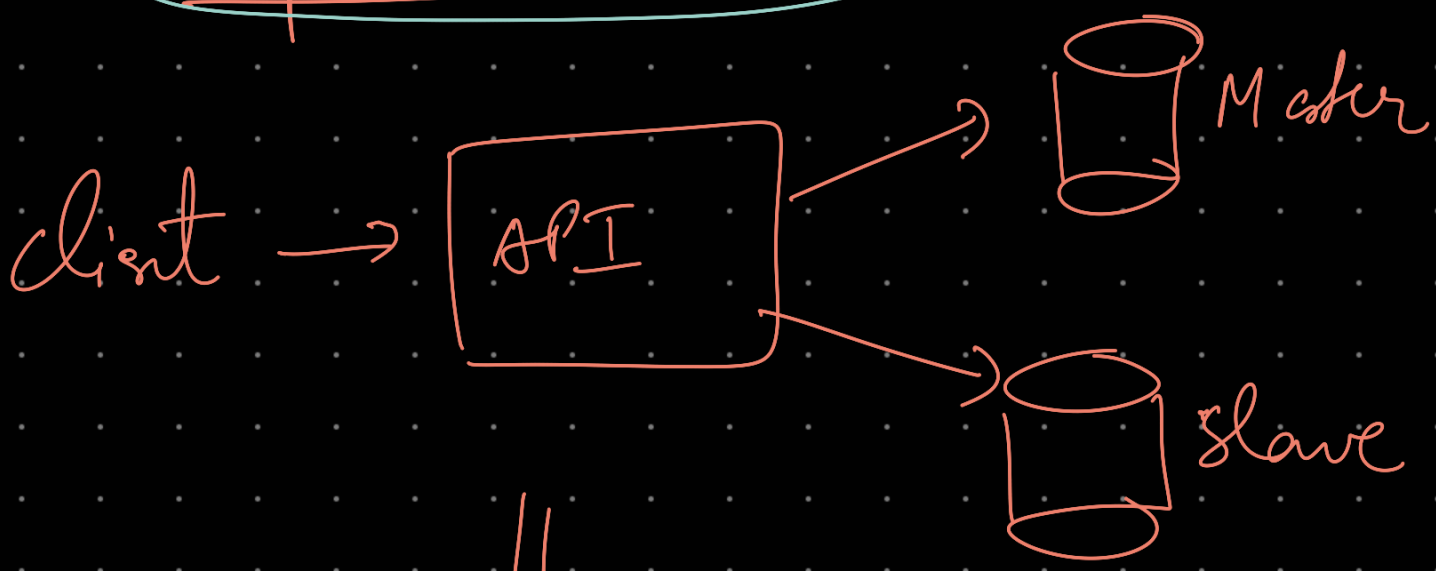
(Add more RAM, CPU, Disk)

⇓

Single Machine reaches Max Limit

⇓

Replicate Data



Horizontally scaling

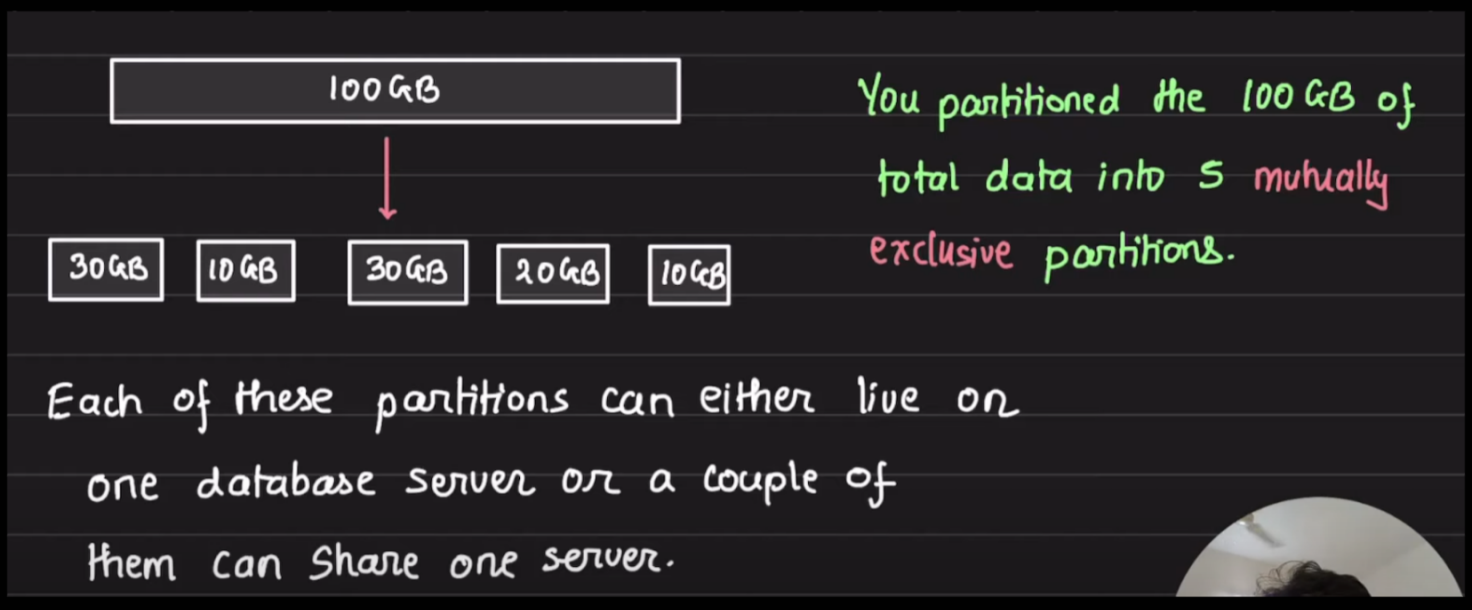
We also split data

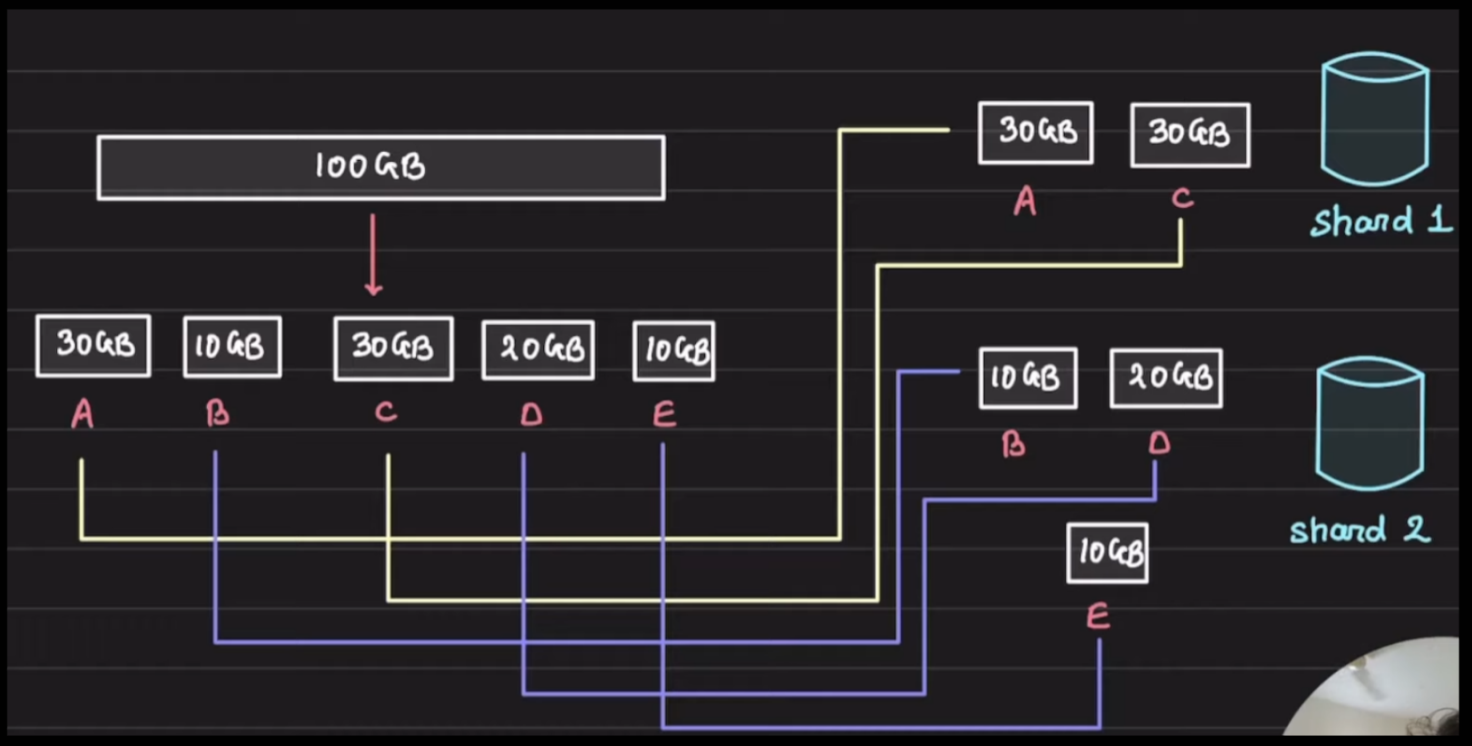
→ partitioning

Data level → partitioning

Machine level or Database level → sharding

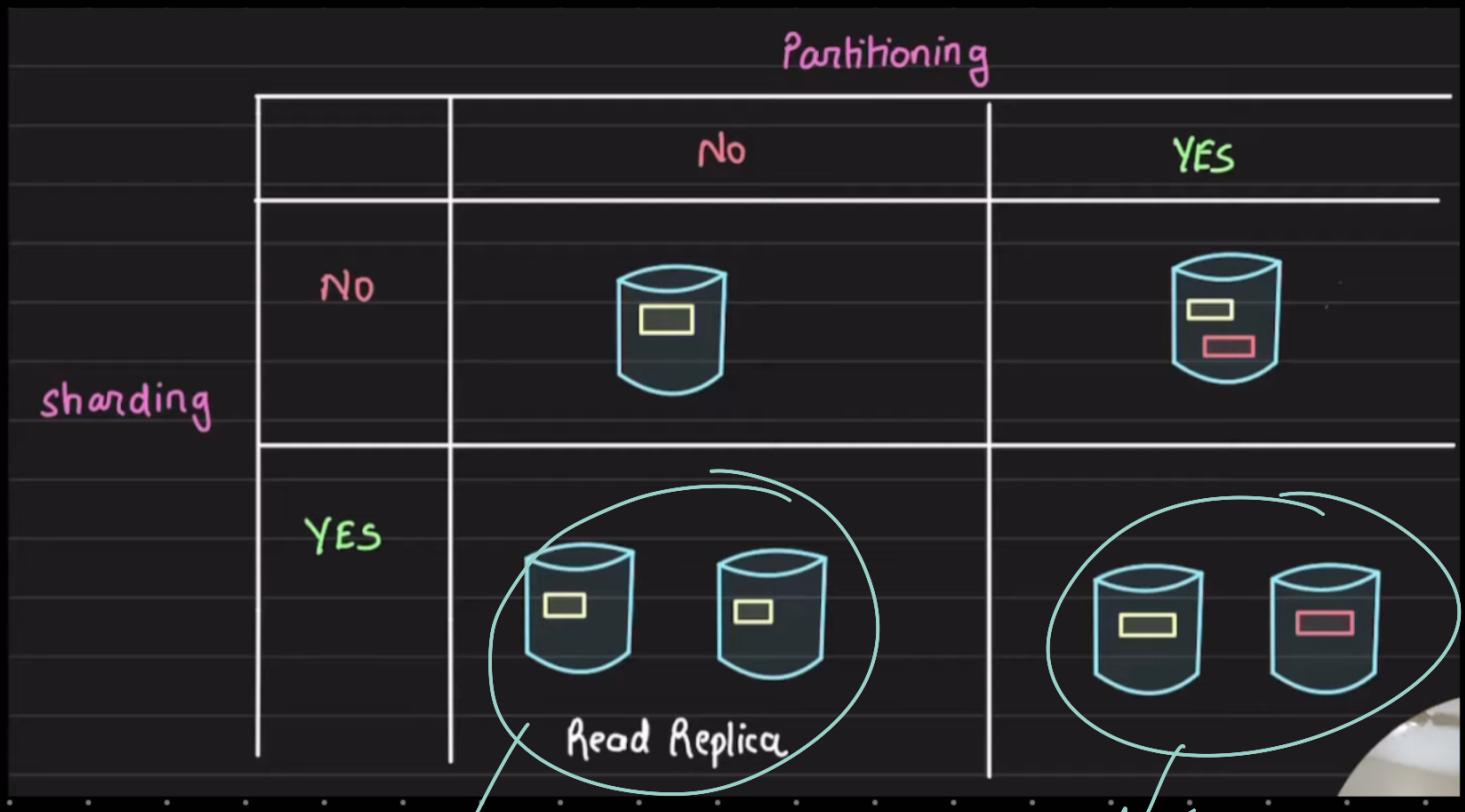
* Each database is a shard and we partitioned the data





Partitioning

- Horizontal
  - Row/document splitting
- Vertical
  - column splitting



High Reads

High reads + writes

Advantages of sharding

- Handle large Reads and Writes
- Increase overall storage capacity
- Higher availability

Disadvantages of sharding

- operationally complex
- cross-shard queries expensive

* In case of JOINS across shards/Machine, networking is required → Latency increased

Steps

1. Check slow reads/write
2. Analyze slowest queries
3. Create index
4. Introduce Cache
5. Replicate DB
6. Scale vertically

⇓

Exhausted all options, then

↓

Horizontal scaling + sharding

Compromise on ACID