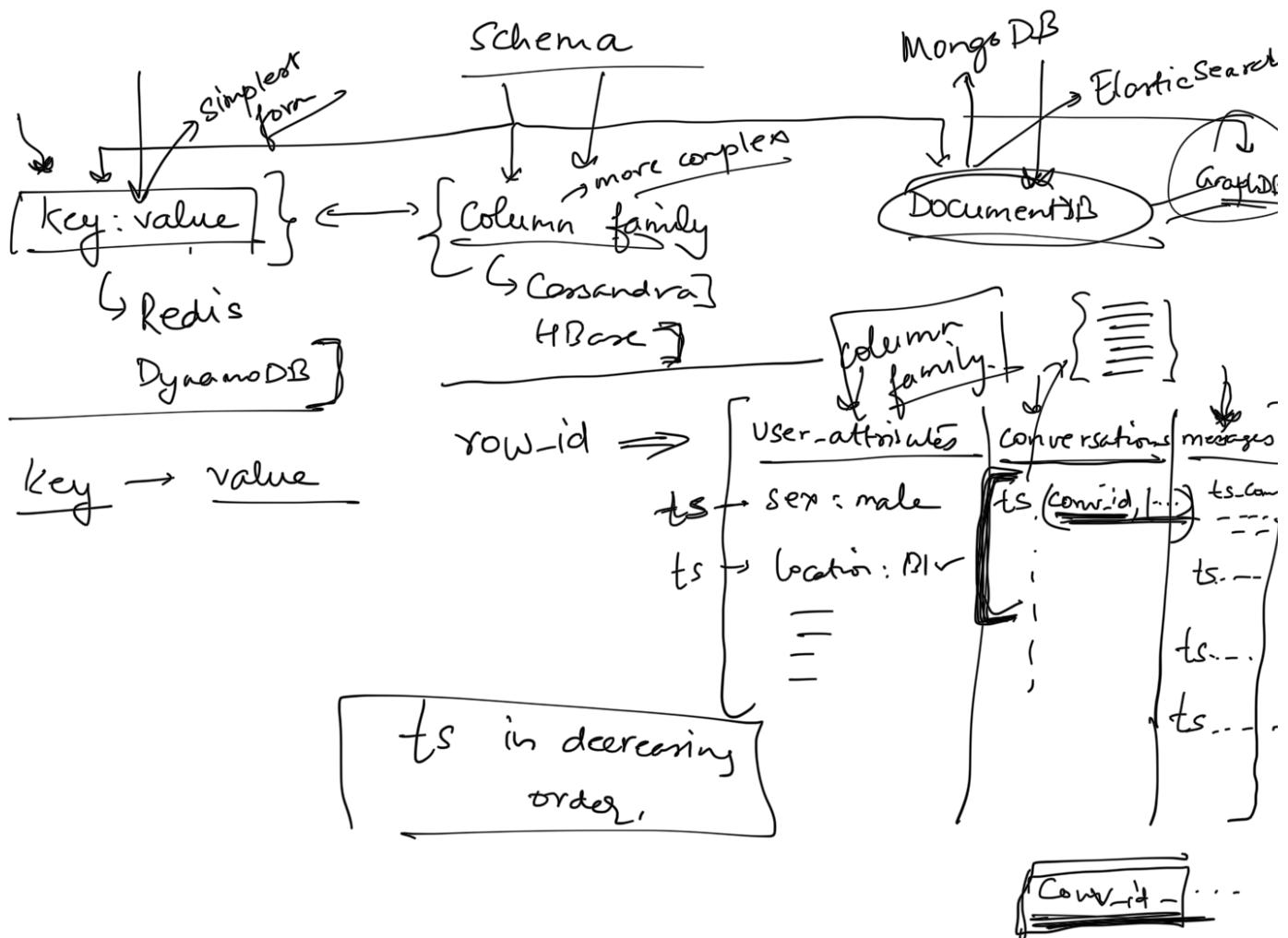


NoSQL Databases

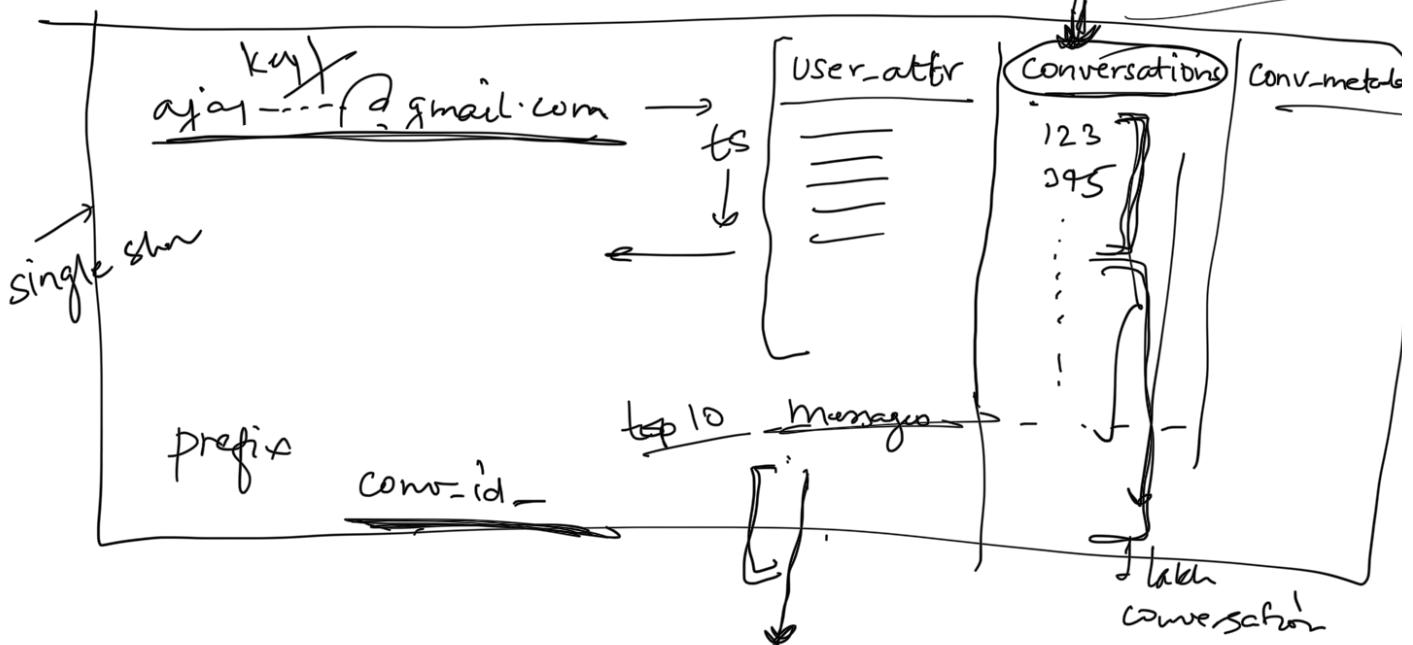
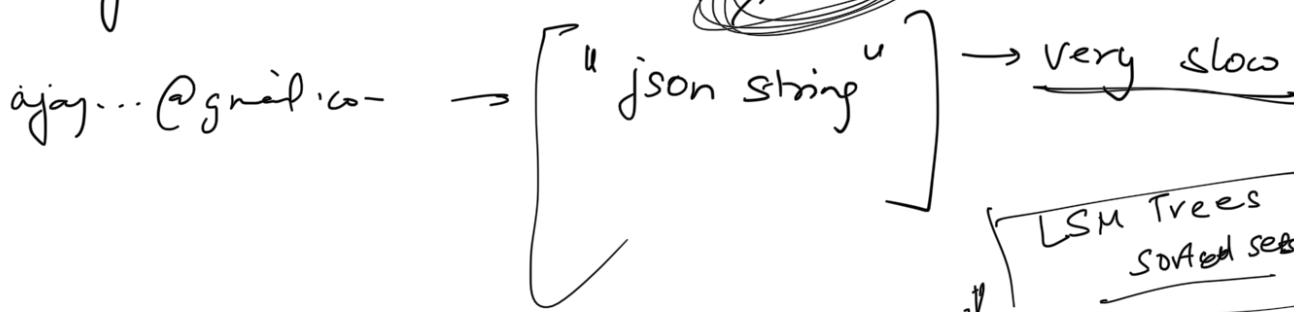
- Intro ✓
- Types of storage ✓
- Sharding key ✓
- Basic NoSQL architecture
 - some problems
- Multi-master

NoSQL {
① Sharding support
② Schema-less → fields of diff. size,
→ variable # of fields.



JULY

Key → value



Documents

↳ JSON

city \Rightarrow bangalore

uid. 1

"city" : "bangalore"

"address": { "pincode": "560100" },

"society": "xy2"

"apt no": - - -

{ email : user_id ,

conversations : { - - - - } }

meo

"key" : "value"

1) id

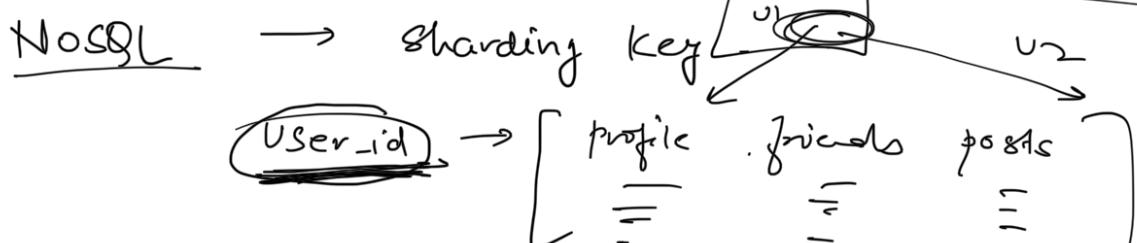
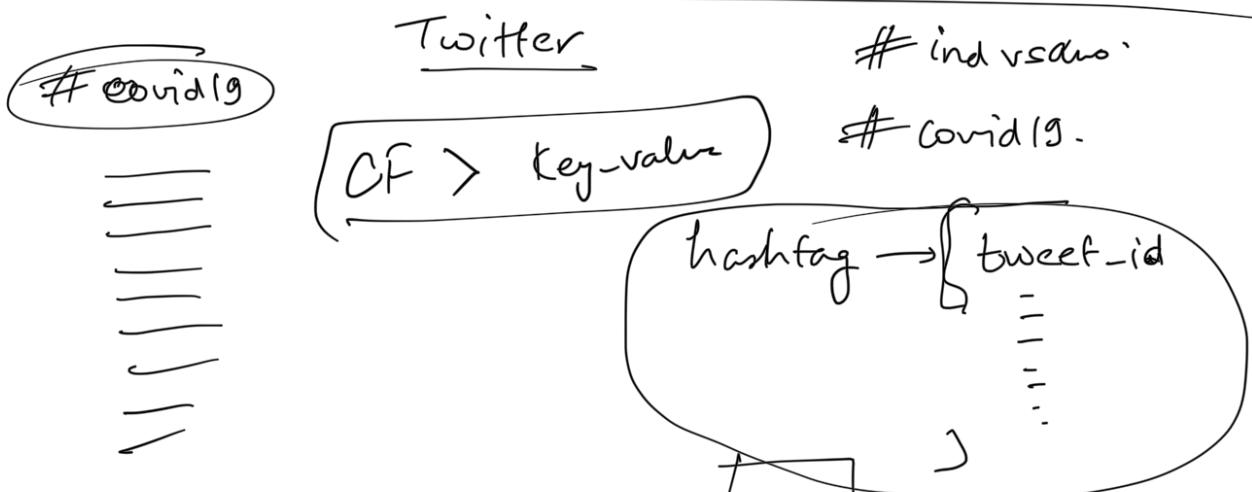
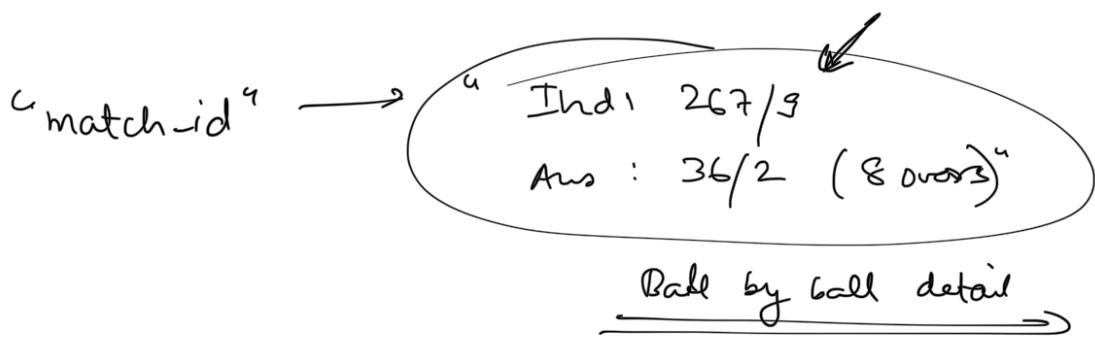
db.store () { "type": "taxi",
 "no": "MH12-1234",
 "model": "Maruti Swift" } ,

db.store ({ "id": "500",
 "name": "Ajay" }) ,

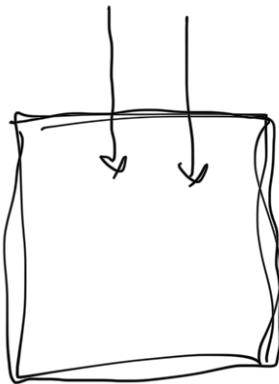
db.find ("type" = "taxi")

~~db~~ db.sharding-key ("id")

Live scores



How to choose sharding-key



① List most frequent usecases:

② sharding-key ~~to~~ **entity**

which entity



1 machine

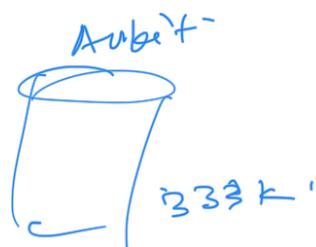
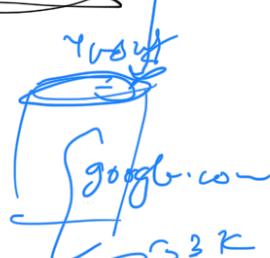
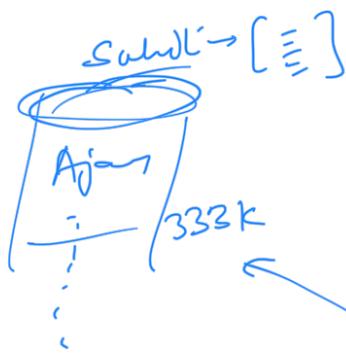
Delicious.vc.

- ①
- add bookmark (user-id, site-url, title)
 - Get All Bookmarks (Userid)

User-id

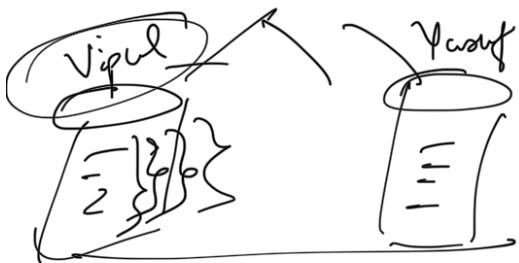
~~X~~ site-url ~~X~~

Title



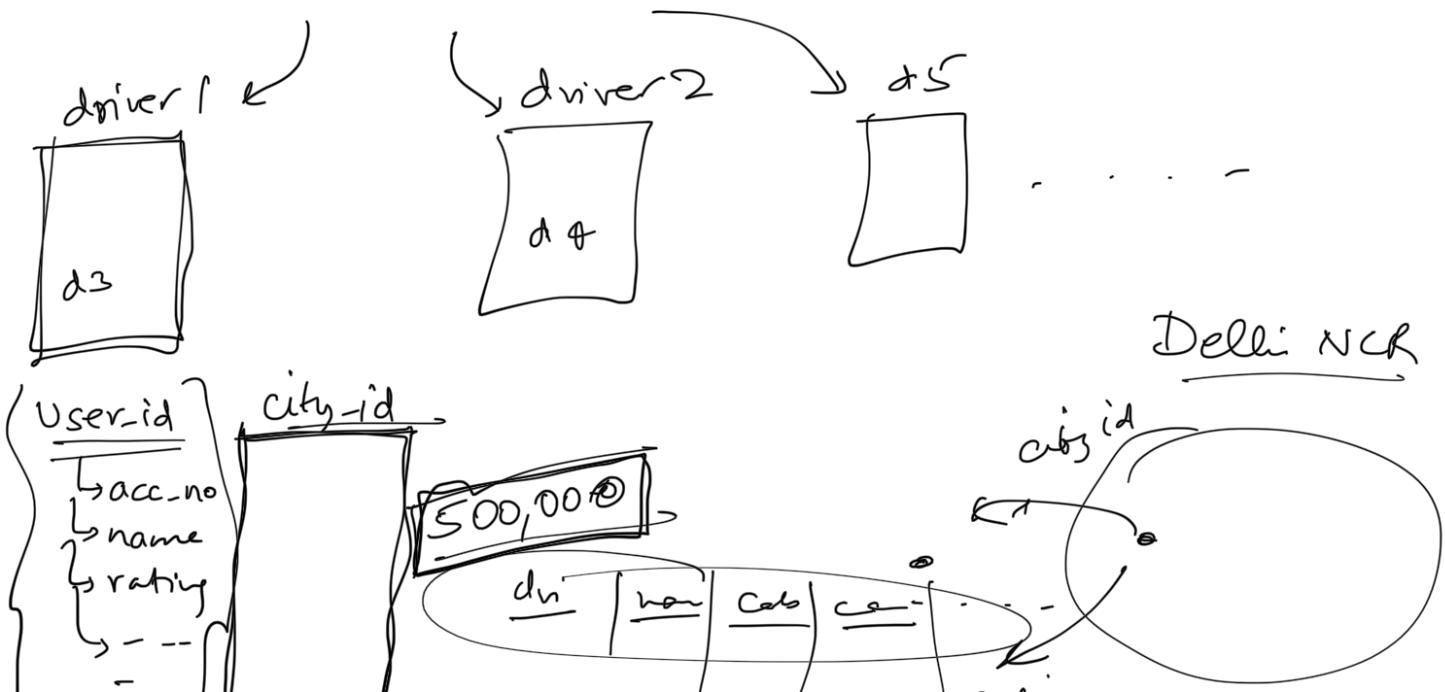
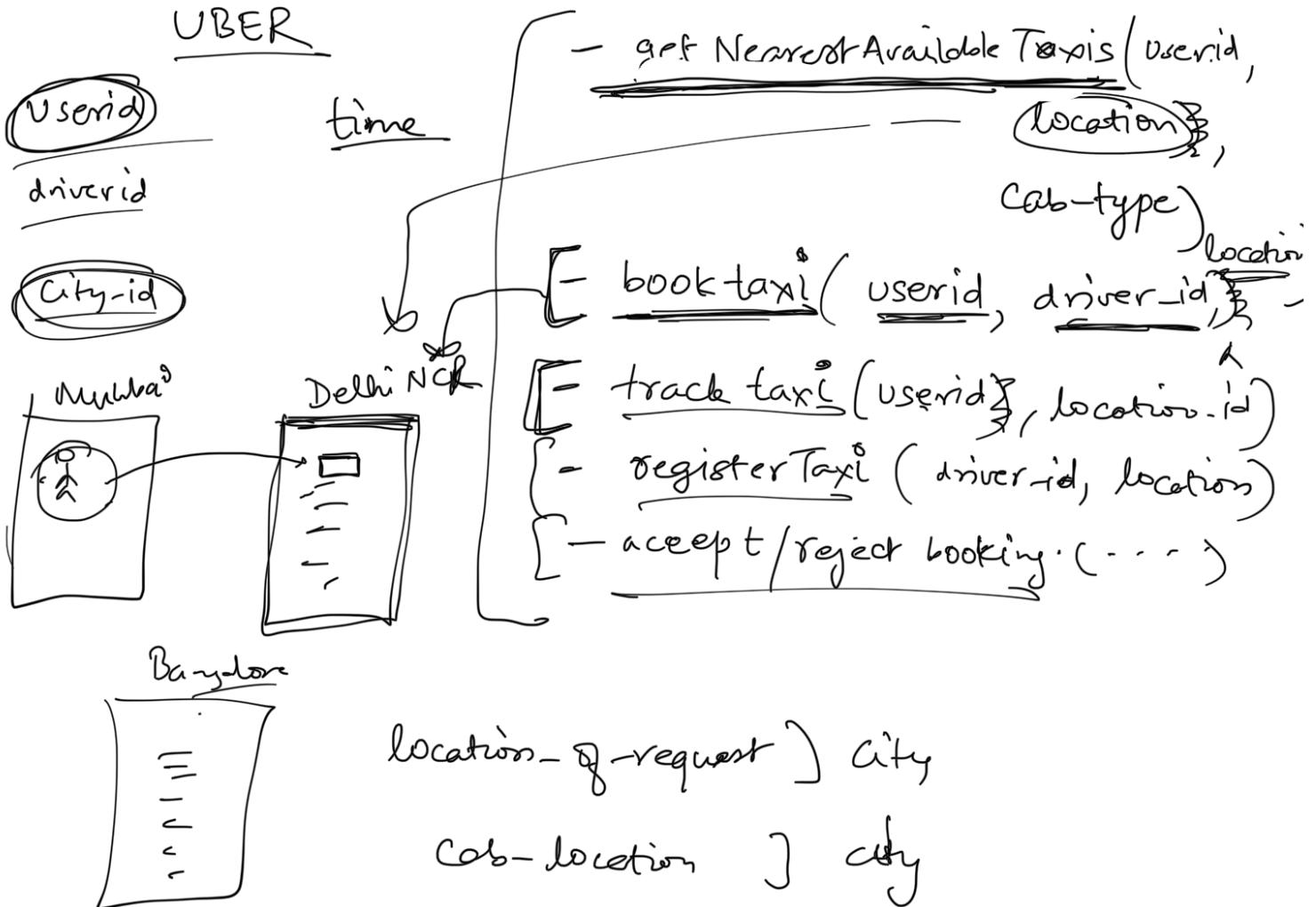
{Closed data }
leaderboard

Scalr

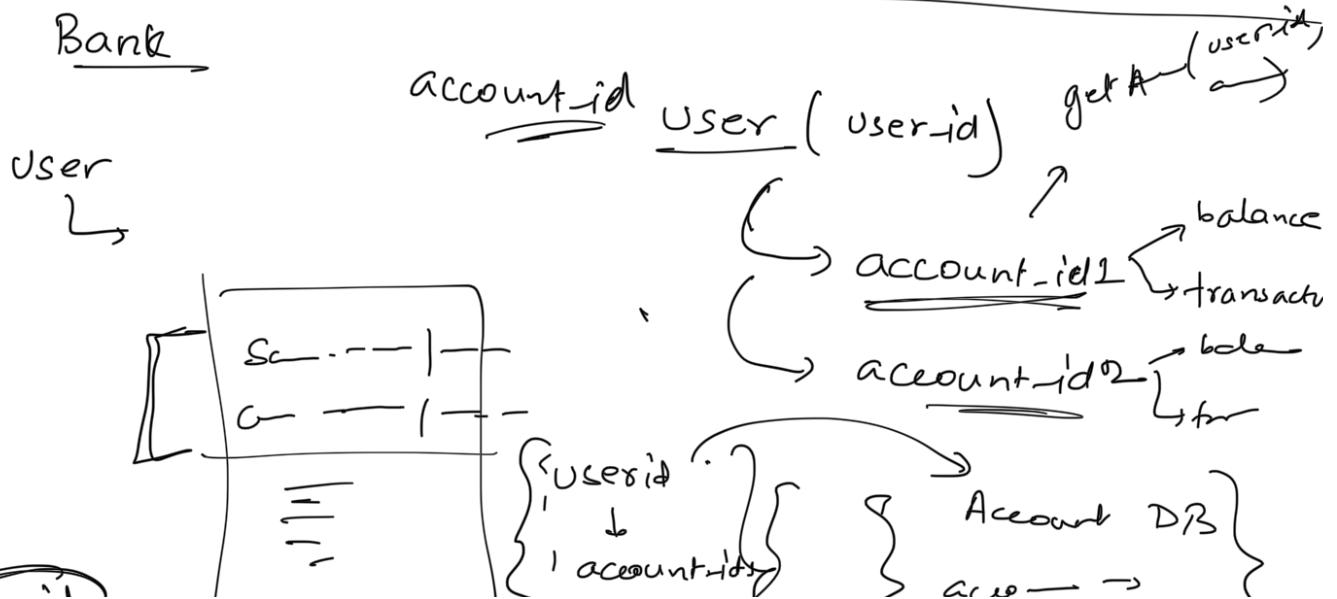
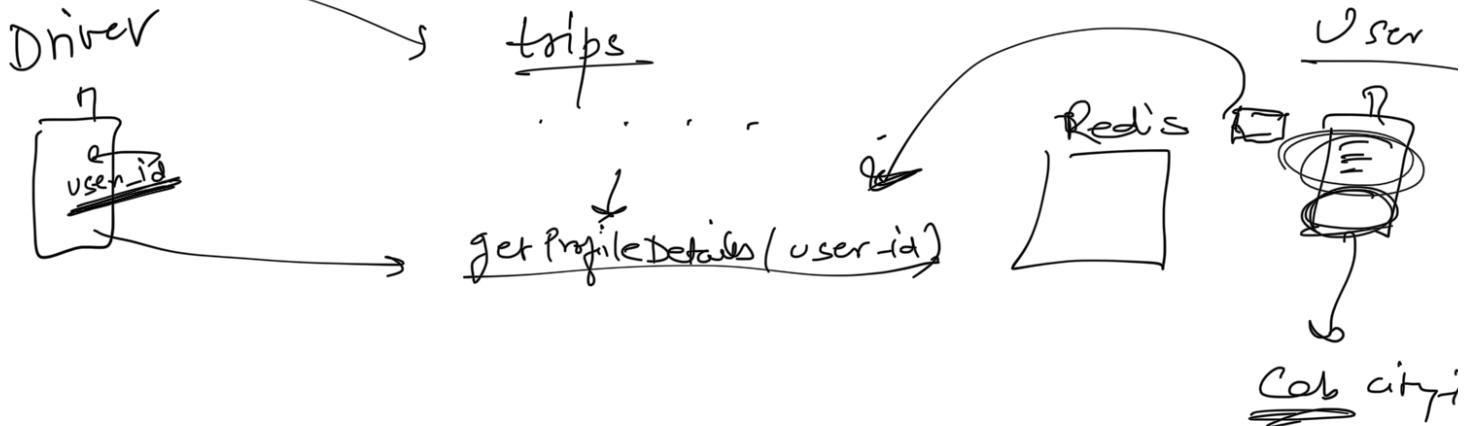
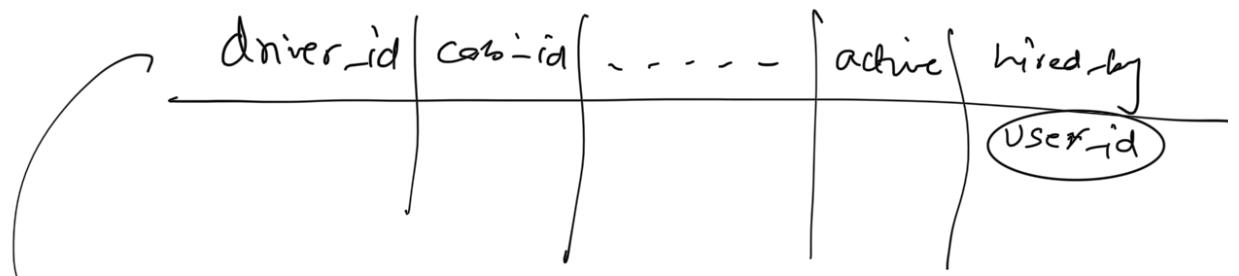


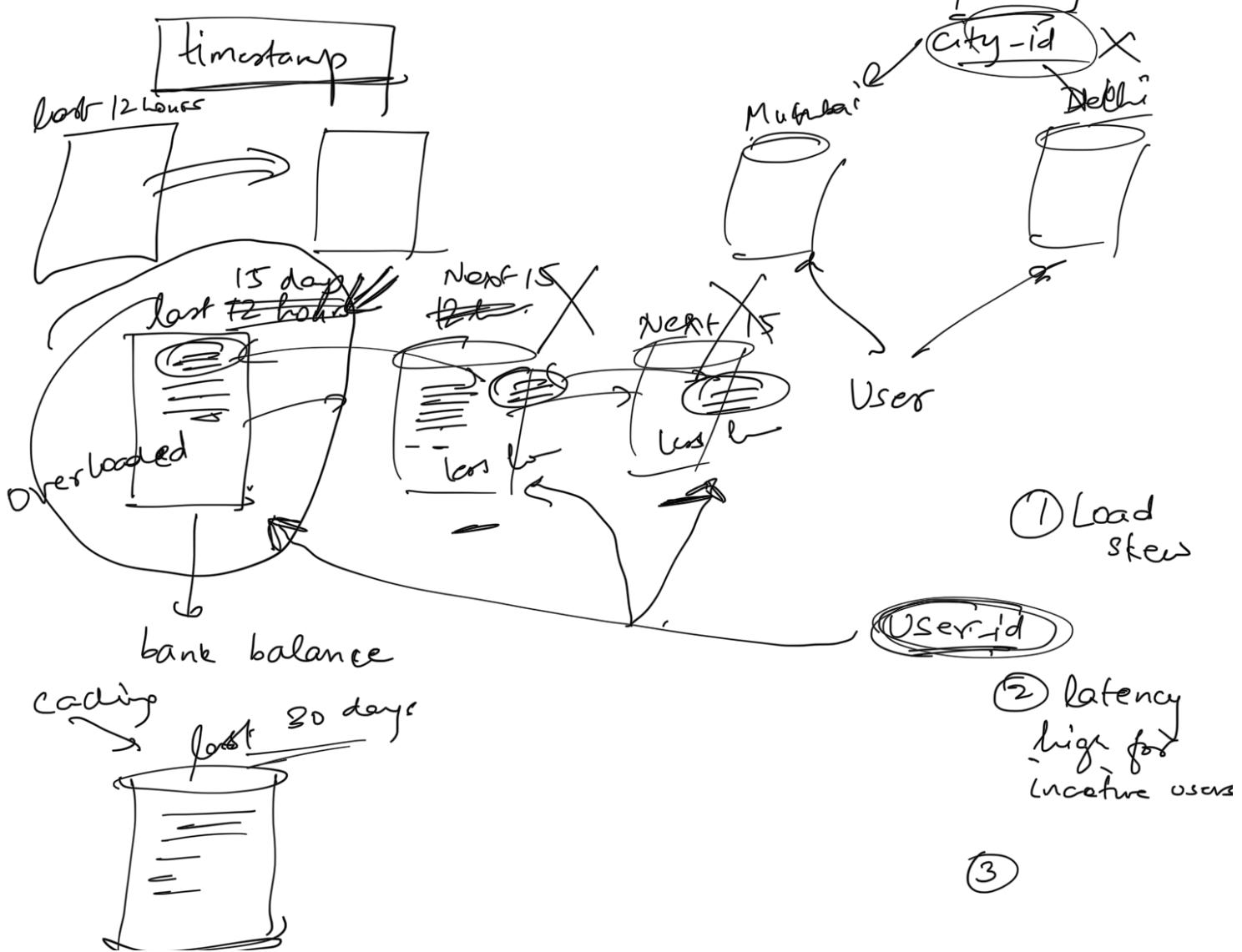
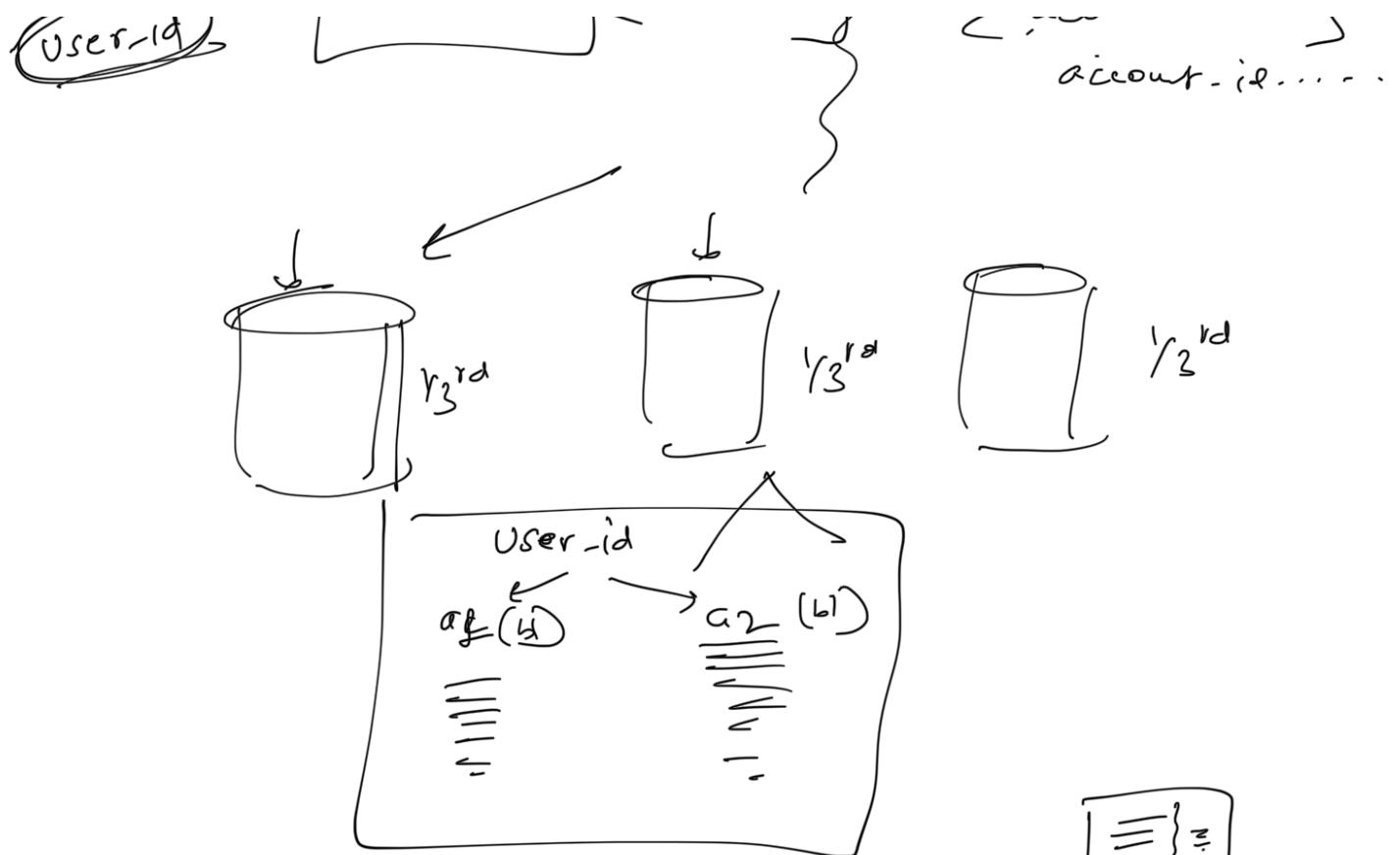
- get closed (userid)
 get unsolved prn (userid)
 get problems (userid, ~~class-id~~)
 get streak (userid)
 get PSP (userid)
 get leaderboard (userid) batch

DB



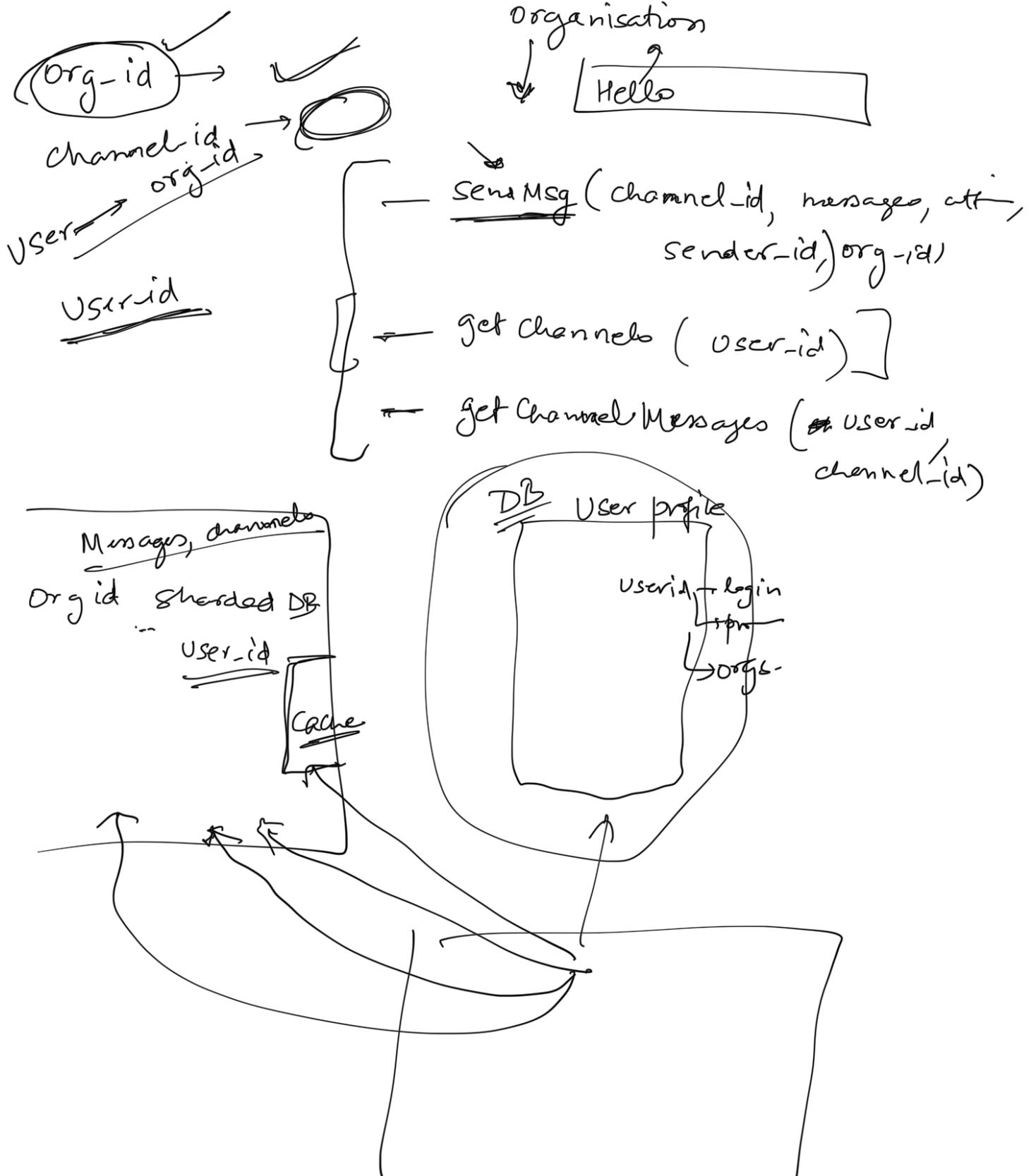
L : J L | active

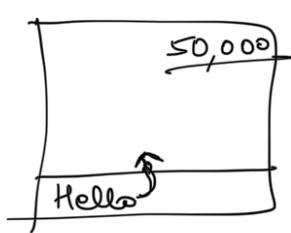
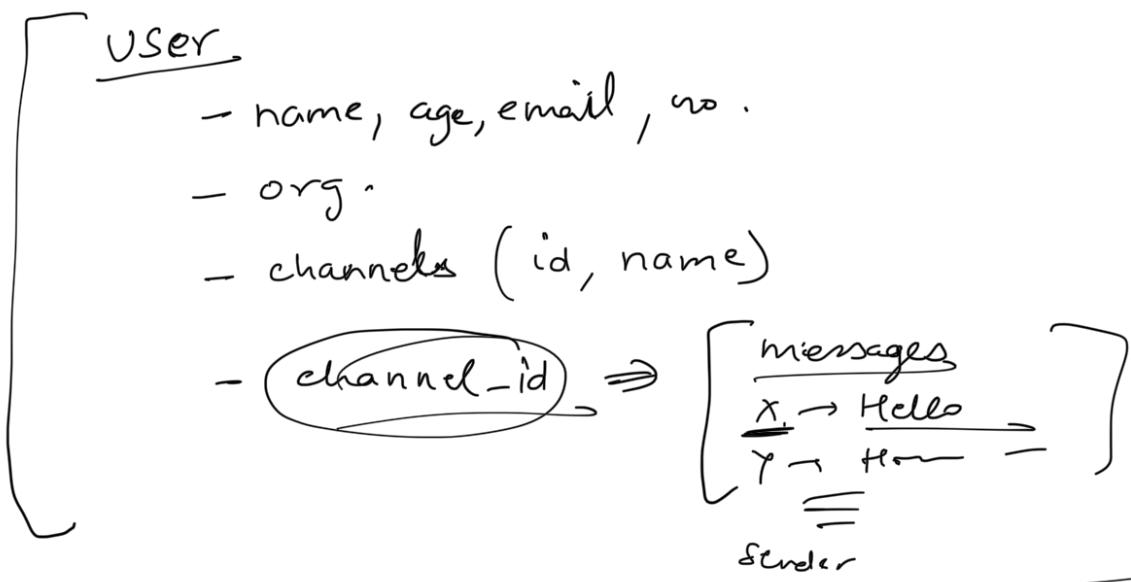
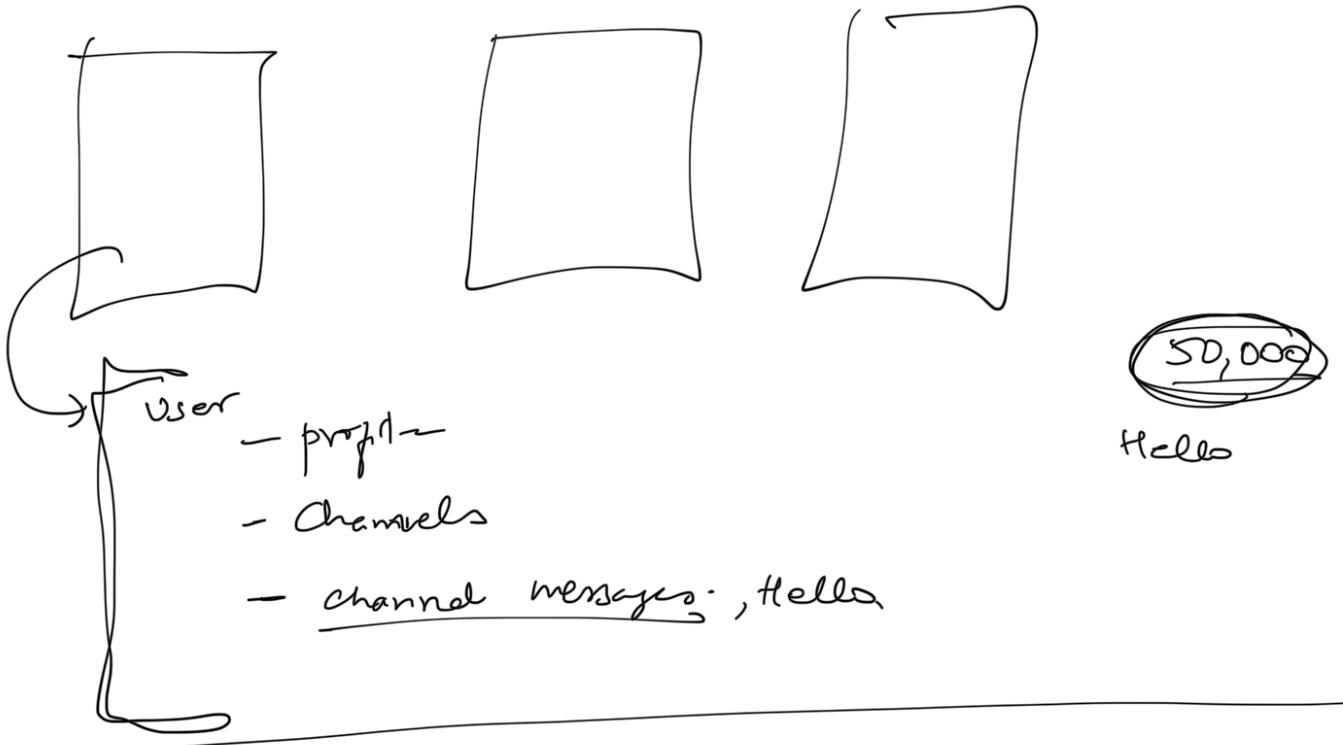


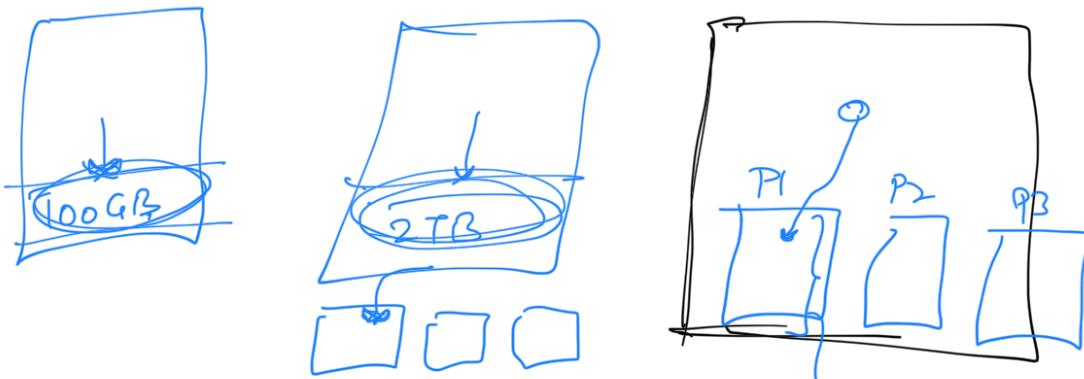
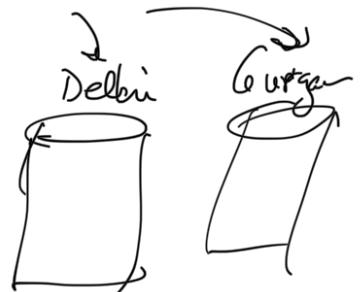
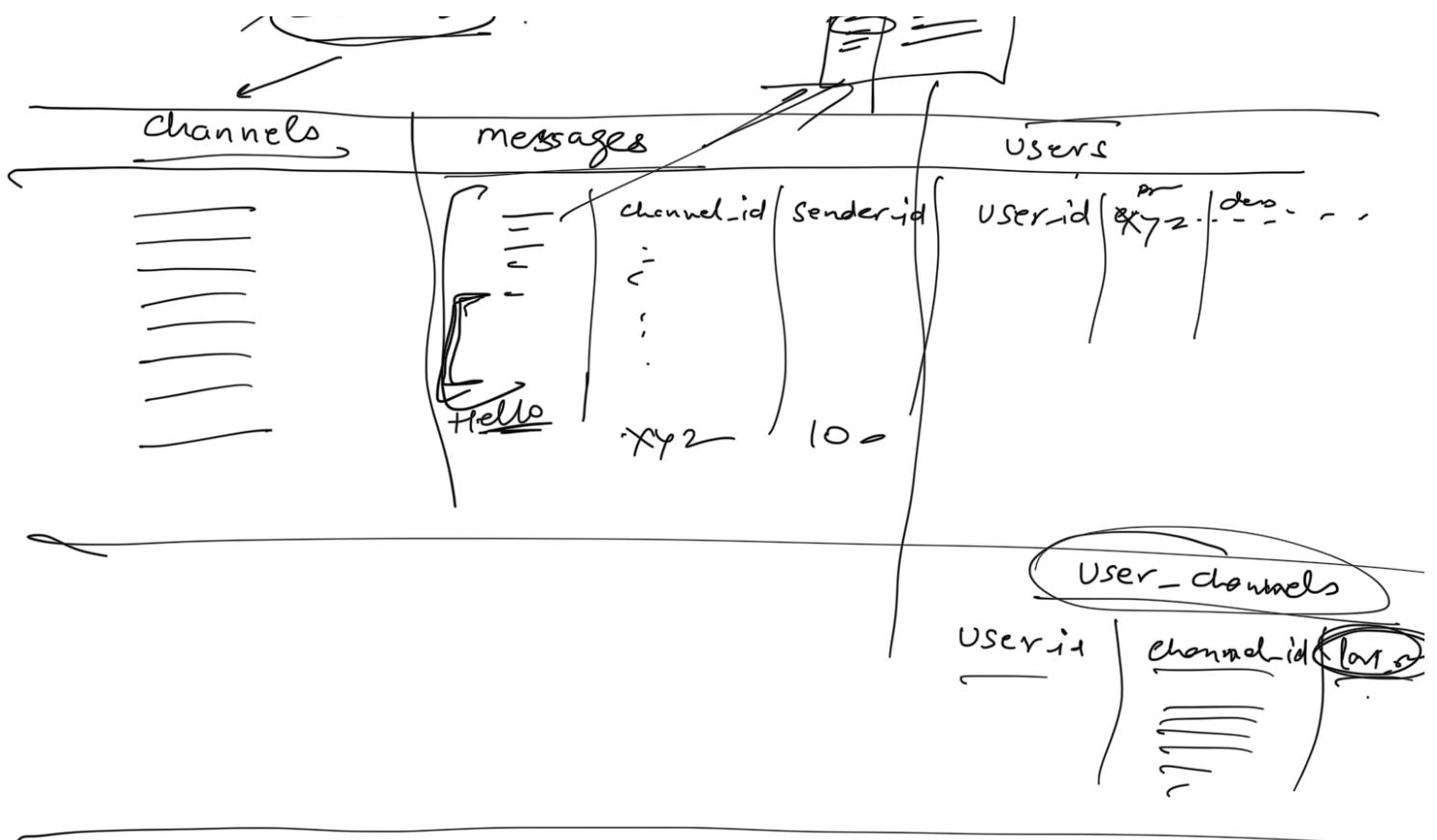


Slack

→ #channel → 10,000





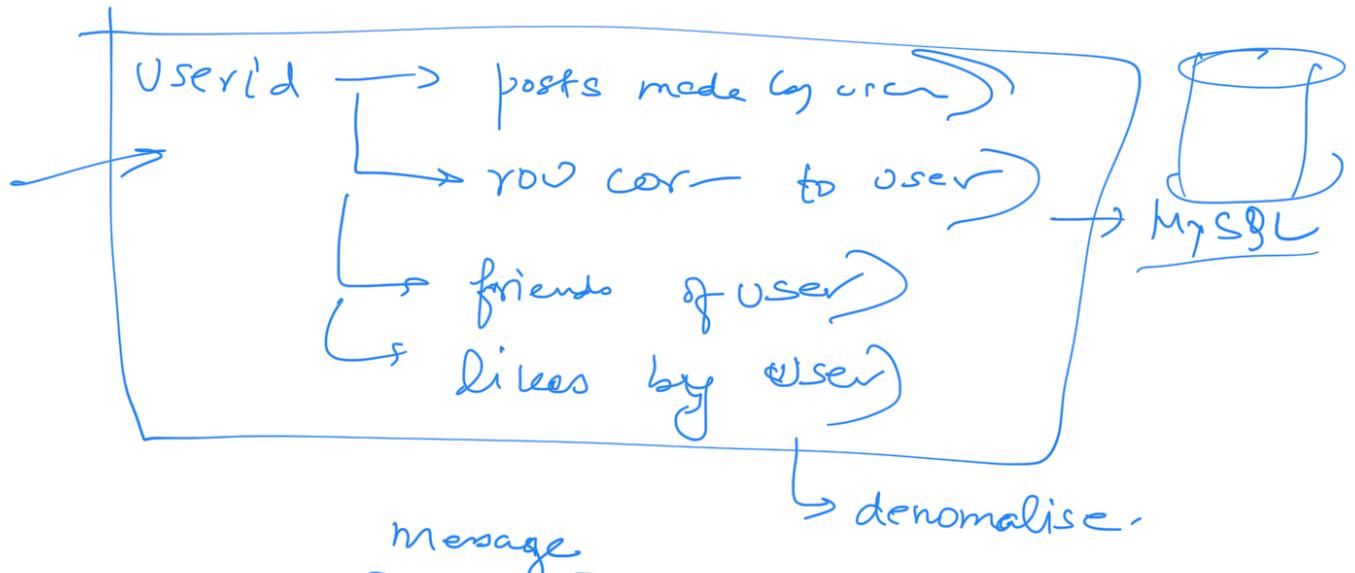


① partitioning
↳ sharding

② MySQL,
partitioning.

posts

more



A → Hello → B.

user id

100

Hello	100	110
-------	-----	-----

message

message	sender	rec
Hello	100	110

110

Hello	100	110
-------	-----	-----

partition → shard

Row id

CF1

CF2 CF3 CF4

Database

DB1

DB2



