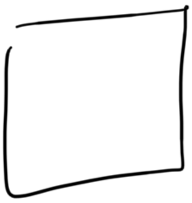
NOSQL - II



Inside a machine

(1) Unlike SQL, NoSQL is unstructured and has no fixed size

Update → row

xyz → "abc: xyz"

xyz → " ______ "

(2) SQL

- B+ trees
- write: log(N)
- read: log(N)

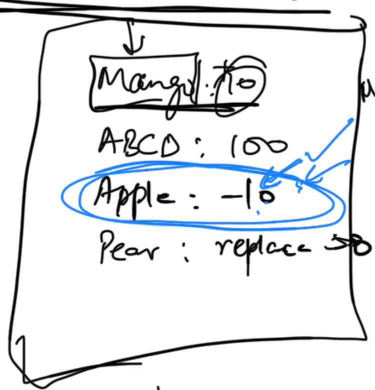
write-heavy ← write more optimise

Key : value

(1) Immutable

↳ update → create a new copy.

(2)



RAM

Mango : 5

ABCD : 1

Apple : 10

Banana : 20

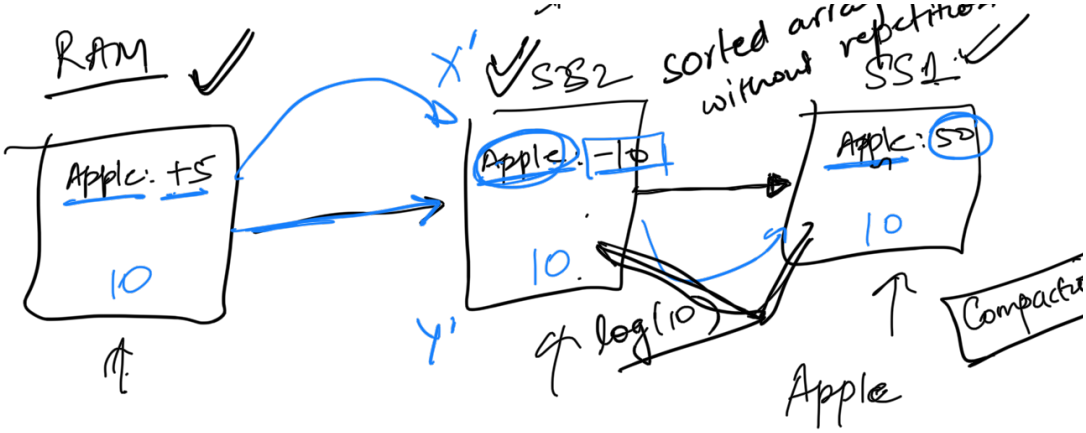
Mango : 50

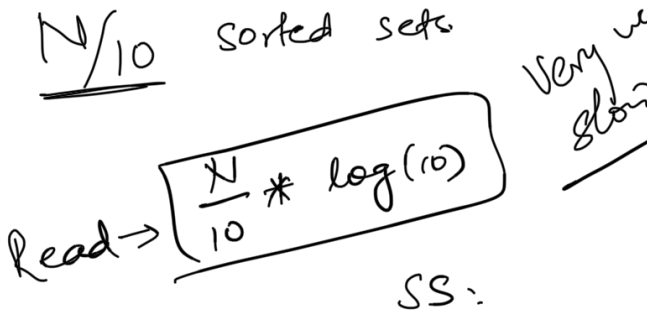
Pear : 30

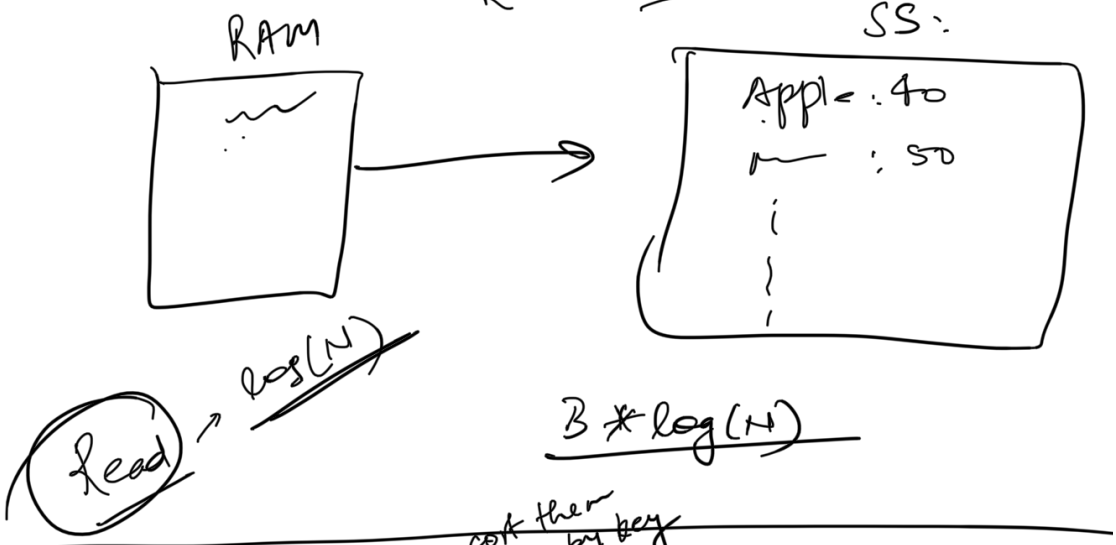
: 40

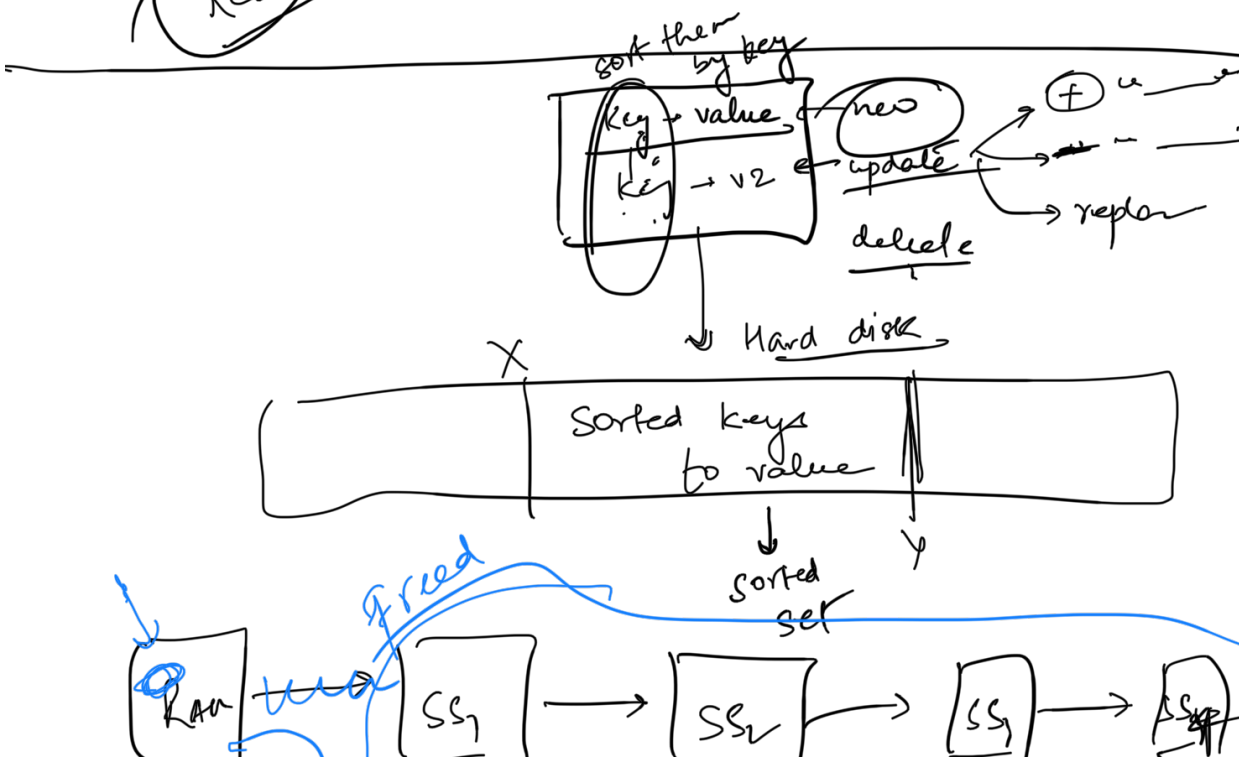
+5 -10 +50

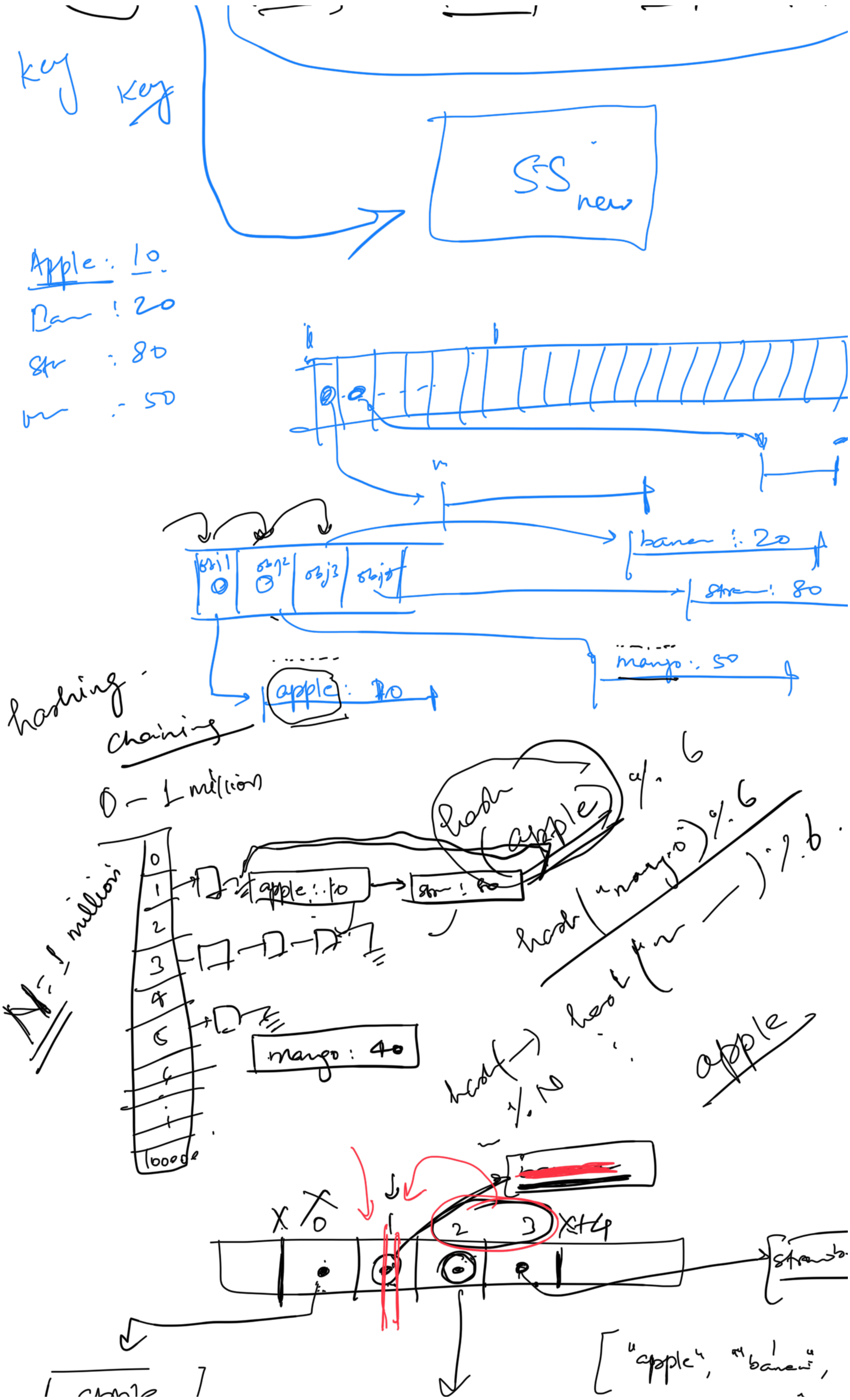
= 45


RAM
SS2 sorted array without repetition
SS1
Apple: +5
10
X'
Apple: -10
10
Apple: 50
10
Y'
log(10)
Apple
Compaction

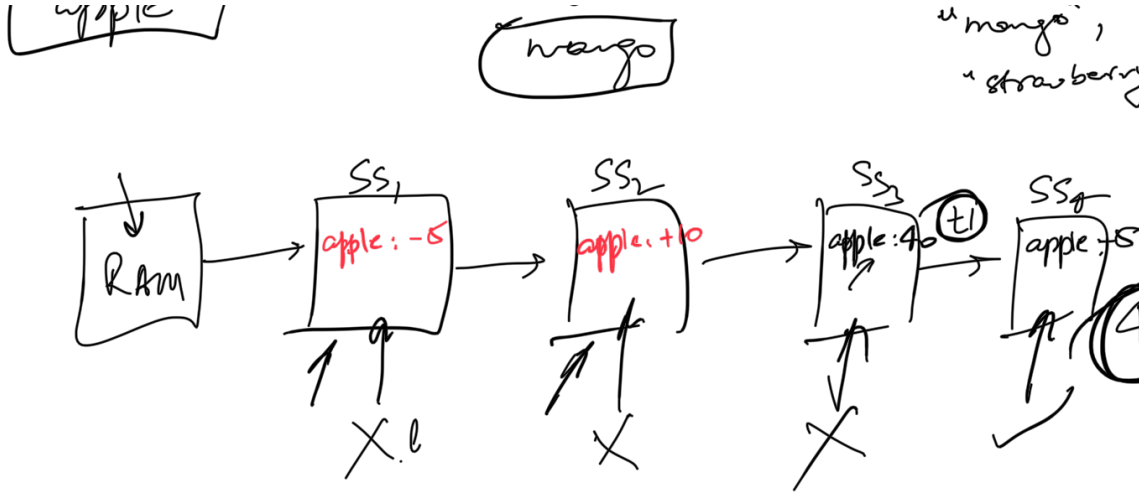

N/10 sorted sets
Very
Read→
N/10 * log(10)

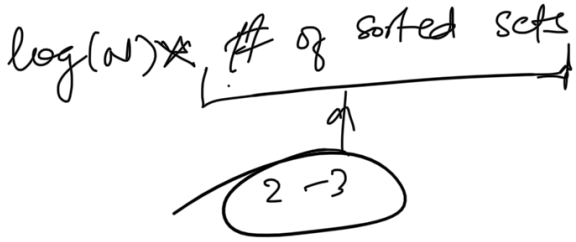

RAM
SS:
Apple: 40
: 50
Read
log(N)
B * log(N)

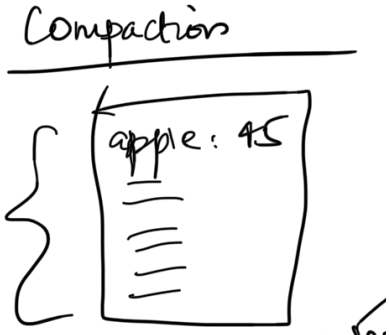

sort them by key
key → value
key → v2
new
update
delete
replace
Hard disk
X
Y
Sorted keys to value
sorted set
Read
RAM
SS1
SS2
SS3

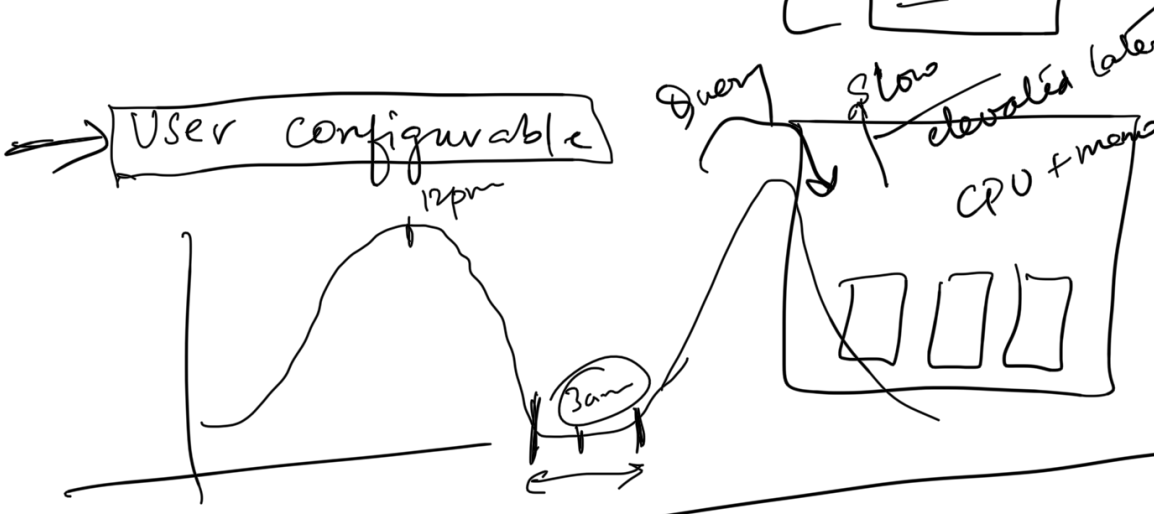


key
key
SS new
Apple : 10
: 20
: 80
: 50
obj1
obj2
obj3
obj4
banan : 20
: 80
mango : 50
apple : 10
hashing
chaining
0 – 1 million
hash(apple) % 6
% 6
mango : 40
apple : 10
% N
apple
X+4
["apple", "banana",

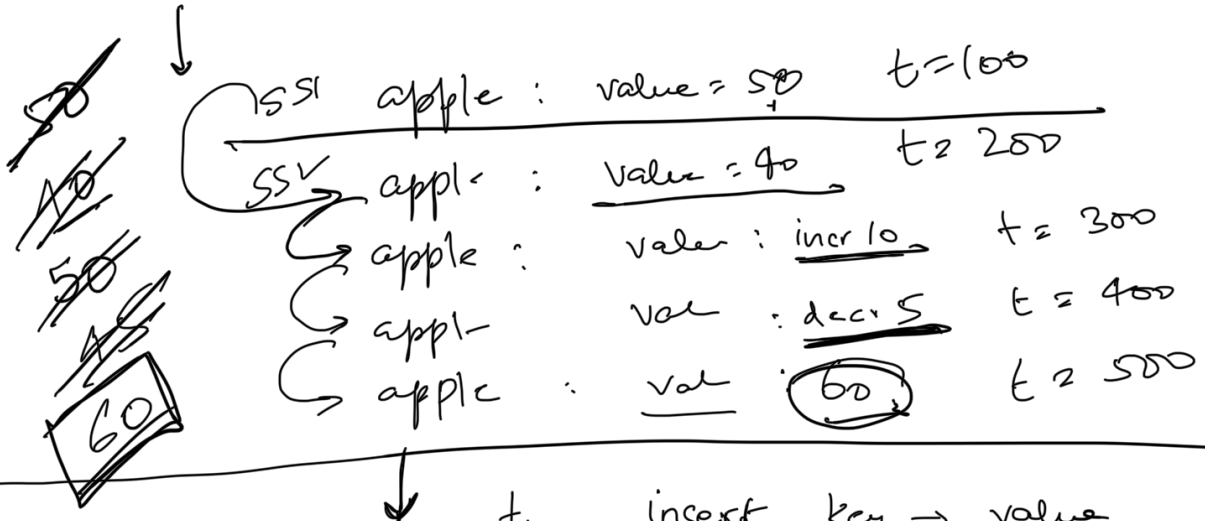




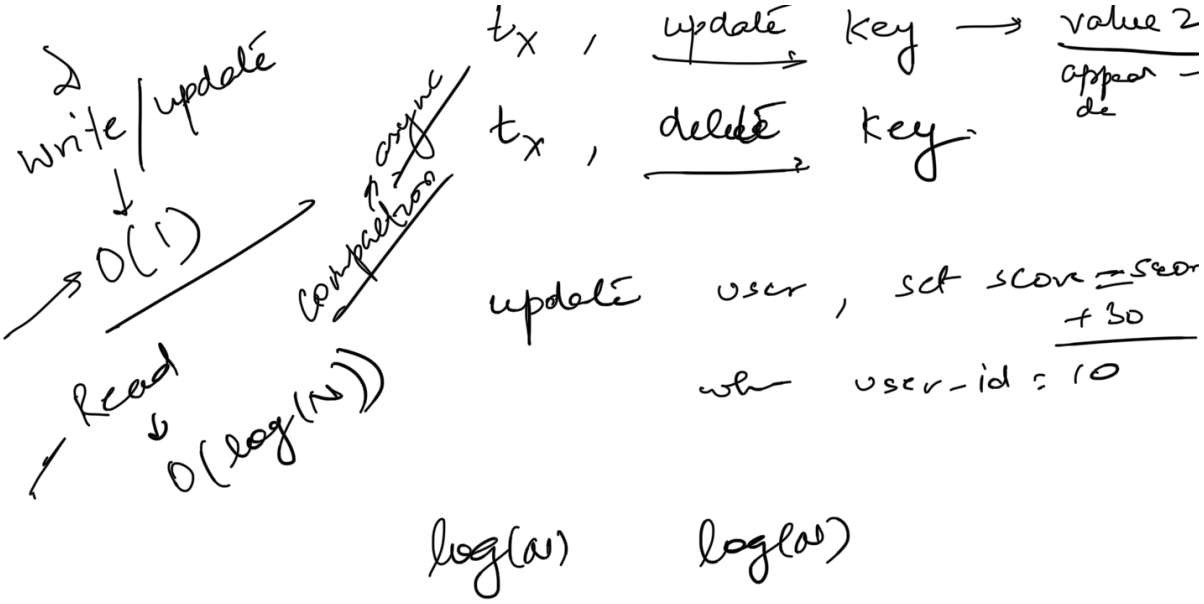


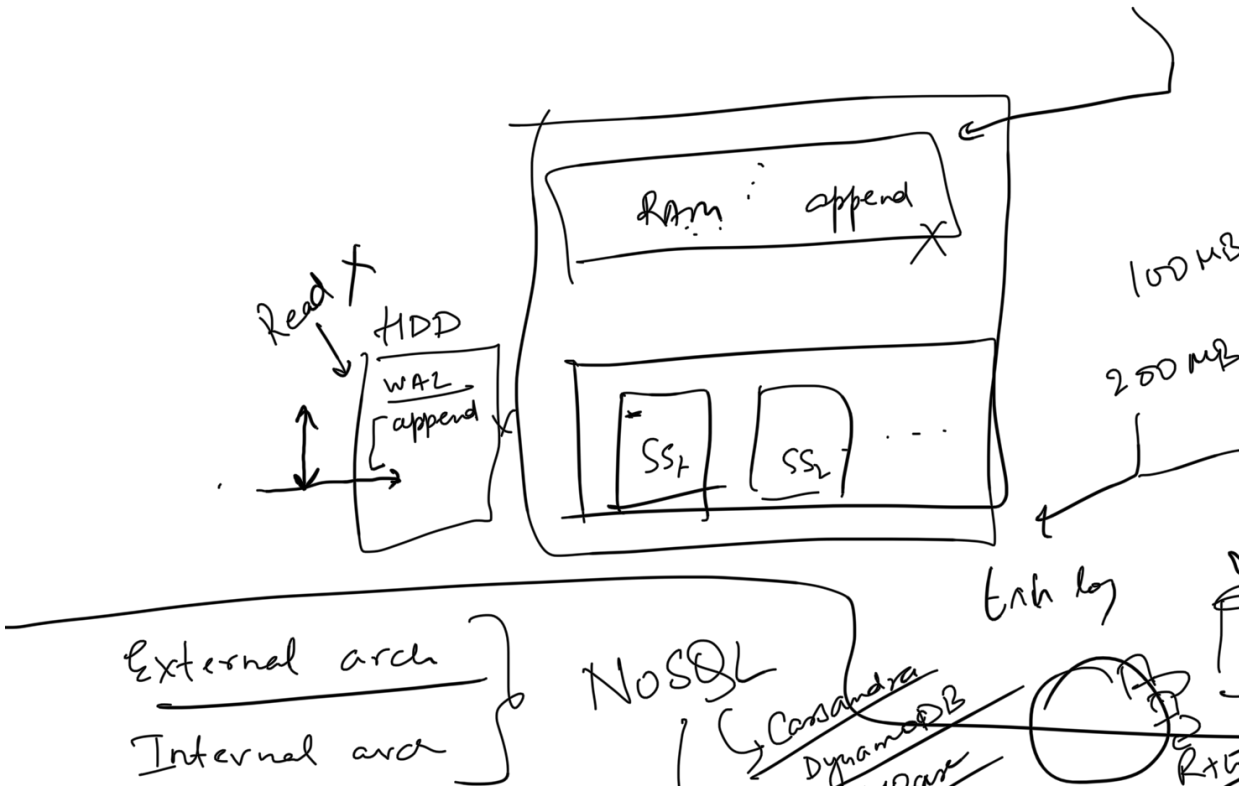


Transaction logs.



t_x, insert key → value


write / update
O(1)
Read
O(log(N))
async compaction
t_x, update key → value 2
t_x, delete key
update user, set score = score + 30 where user_id = 10
log(n) log(n)


Ram
append
Read
HDD
WAL
append
SS1
SS2
100 MB
200 MB
Disk log
External arch
Internal arch
NoSQL
Cassandra
DynamoDB

Sharding data / distributed dat

SCHEMA

Gmail

Messaging

Key : value → Messages

userid → " conversation + messages "

key:value — Redis, Aerospike, Oracle NoSQL

get (key)

set (key, value)

Column family stores — HBase, Cassandra, DynamoDB

Docum... store — Mongo..., Couch...

row_key → CF₁ (Table in SQL) | CF₂

Conversations | Messages

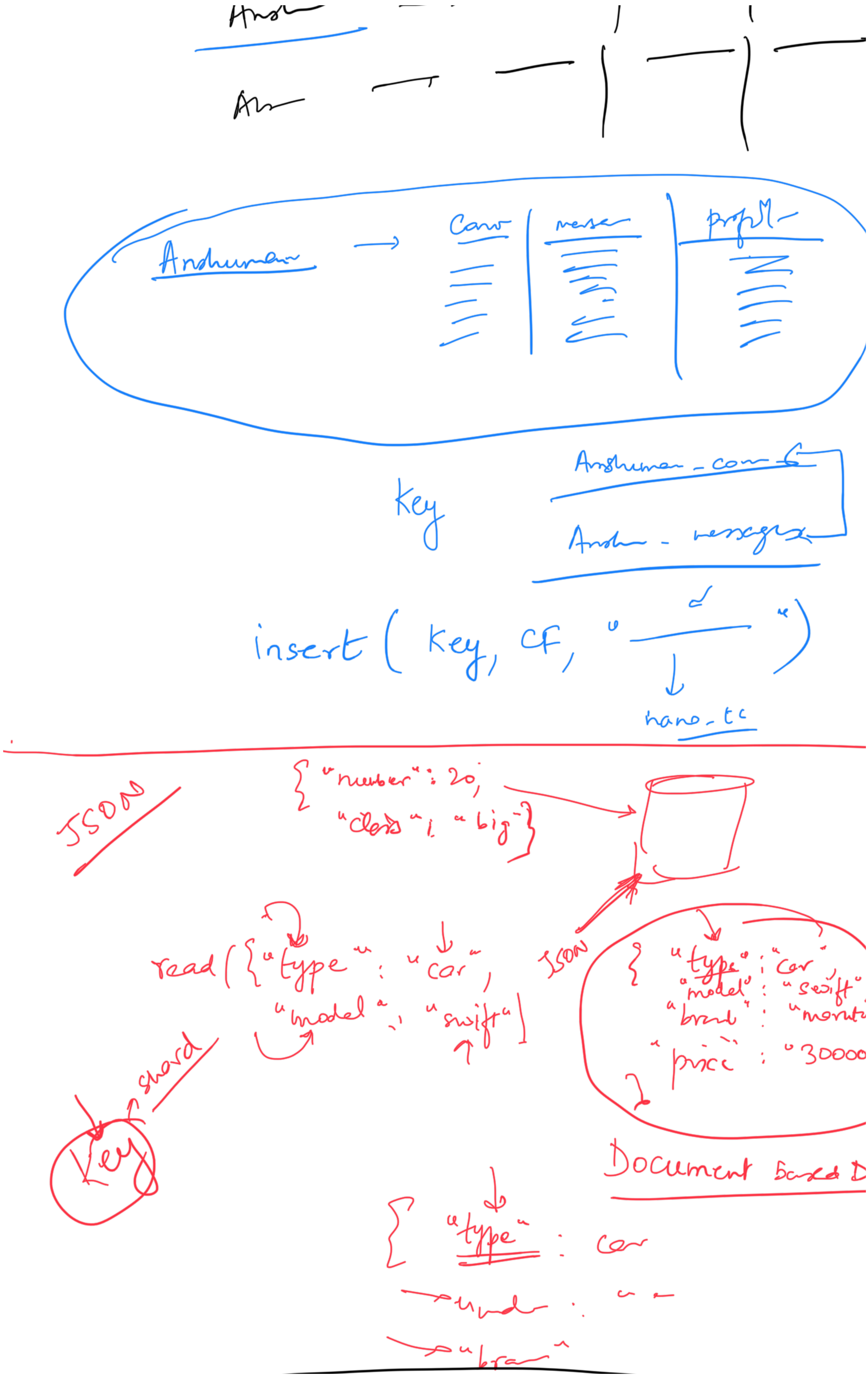
sorted by ts convid

key → CF₁ → most recent X entries

get (key, CF, X, offset)

row-key → Conv | messages | profile

Conv | — | profile


Anshuman
Cars
messages
profile
Anshuman_cars
Anshuman_messages
Key
insert (key, CF, " ")
nano_tc
JSON
{ "number" : 20,
"class" : "big" }
read ({ "type" : "car",
"model" : "swift" }
shard
Key
JSON
{ "type" : "car",
"model" : "swift"
"brand" : "maruti"
"price" : "30000
}
Document based D
{ "type" : car
"model" : " "
"brand"

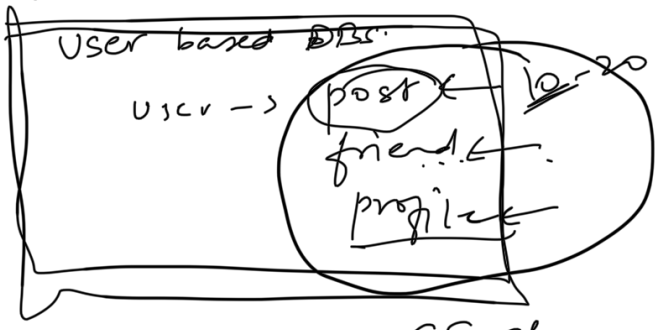
mailbox → CF

pages

pagerank sc

FB case

URL → score

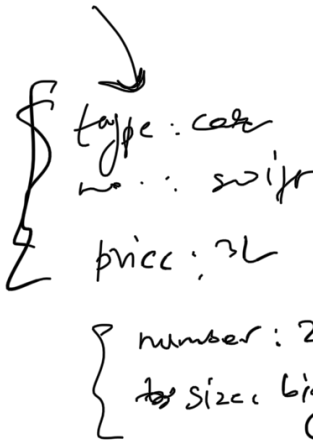


Key → value

CF → Ove

Do

CF store



Google search typehead