

# SYSTEM DESIGN - 5

① SQL DBS  
↳ ACID  
Transactions  
B+ trees  
Indexes

② NoSQL  
↳ Key value  
↳ Doc  
↳ Column  
↳ Cassandra  
↳ MongoDB  
Managing process Kafka

③

- ① Defining MVP
- ② Estimating scale
- ③ Design goals
- ④ API design

↓  
Design

Google search  
typeahead

mich

michael jackson  
michelle obama

① MVP

min length = ③  
mich  
① Given prefix, keyword give back

mich  
 michael jackson

bob m  
 → bombay

top 5 suggestions

most frequently  
 searched

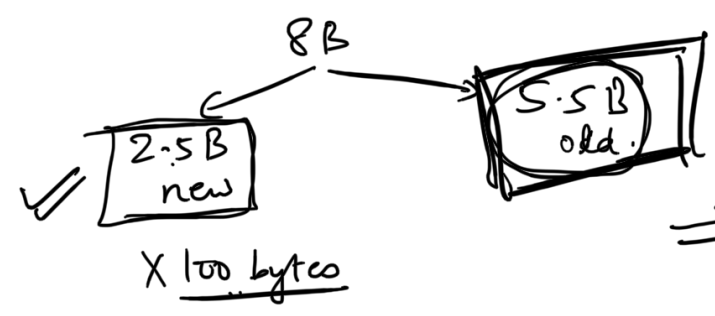
prefix key is strict  
 prefix of all  
 suggestions.

## ② Estimation of Scale

Search query → free

8 Billion queries / day

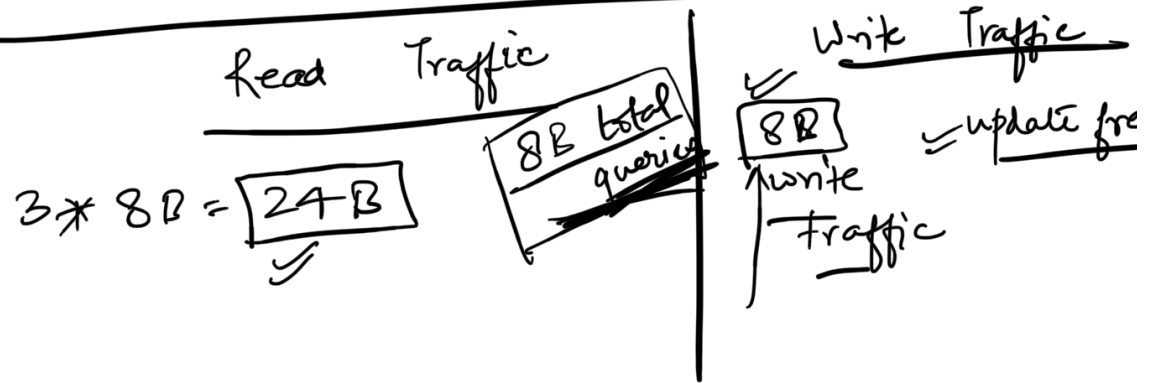
↳ 2.5 B queries which are new.



Sharding would  
 be needed

⇒ 1800 TB

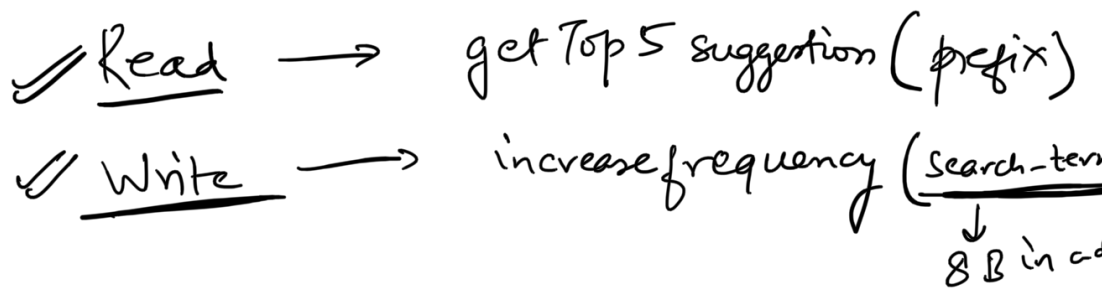
= 250 GB / day



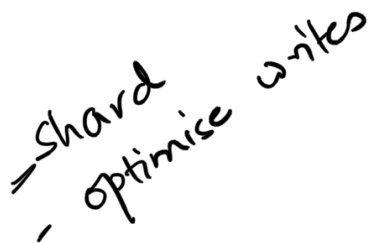
michael j  
 micr → 1 query

1 read + 1 read + 1 read + 1 write

8B



3-5 characters



① Available vs consistar

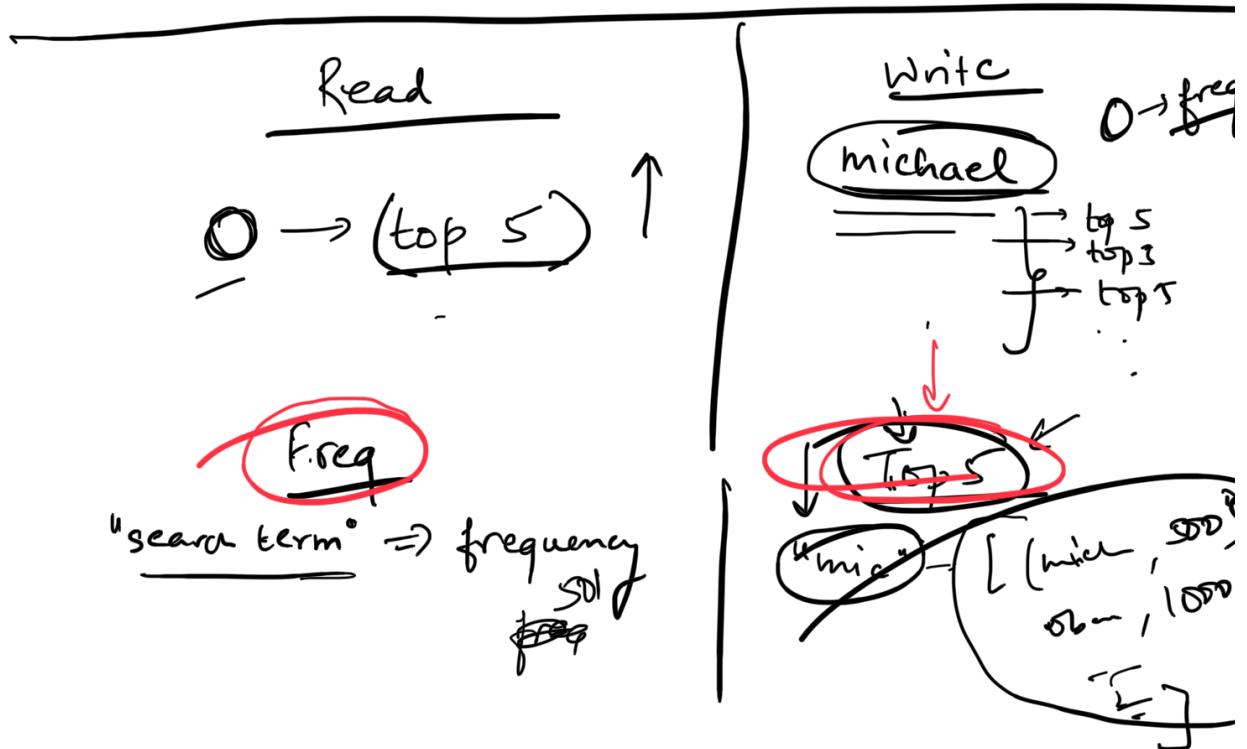
A

② Latency  $\rightarrow$  extremely low latency.

- ① get top 5 suggestion (prefix)
- ② increment frequency (search-term)

- ⑤ Energy 17/11/2020





- ① getTop5 (prefix)
- ② increment freq (michael jackson)  $\leftarrow$

michael

$\rightarrow$  m  
 $\rightarrow$  mi  
 $\rightarrow$  mic  
 $\rightarrow$  mich  
 $\rightarrow$  micha  
 $\rightarrow$  michael

Freq

"bat"  $\rightarrow$  20

"batman"  $\rightarrow$  50

"battle"  $\rightarrow$  60

ba  
bat

top 1

"ba"  $\rightarrow$  ["battle", 60]

"bat"  $\rightarrow$  ["batman", 61]

"battl"  $\rightarrow$  ["battle", 60]

batm  
batma  
batman

batman  
battle

search

~~key~~

Sharding Key  
prefix

↓  
WRITE

① "batman"

↳ update freq

② all prefixes

freq  
"batman"

batman  
~~100~~  
100

battle  
200  
101

freq | Top5  
se

① recent update  
"search" → count 100

↓ stale  
1 day

Batch update

②

"batman"

1000000

100000

"battle"

1000000

1.5M

"batman"

freq

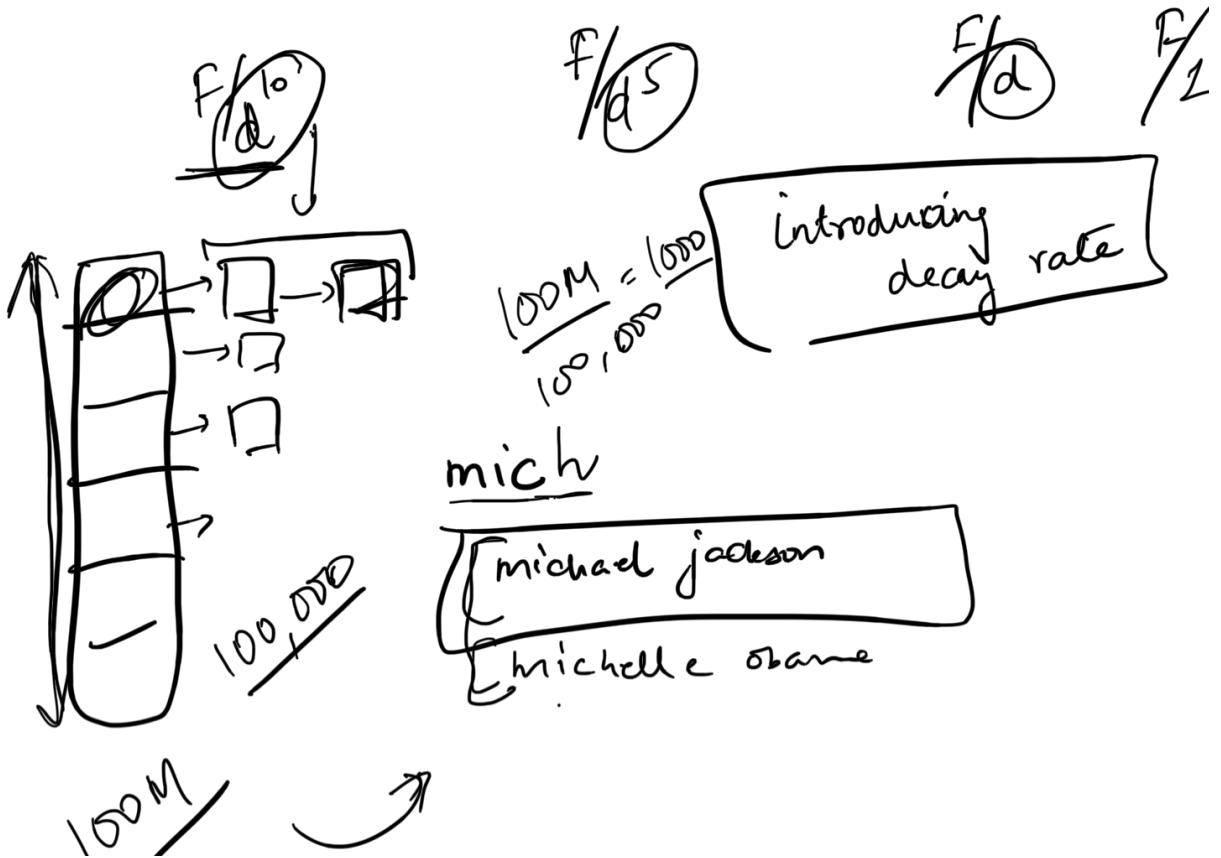
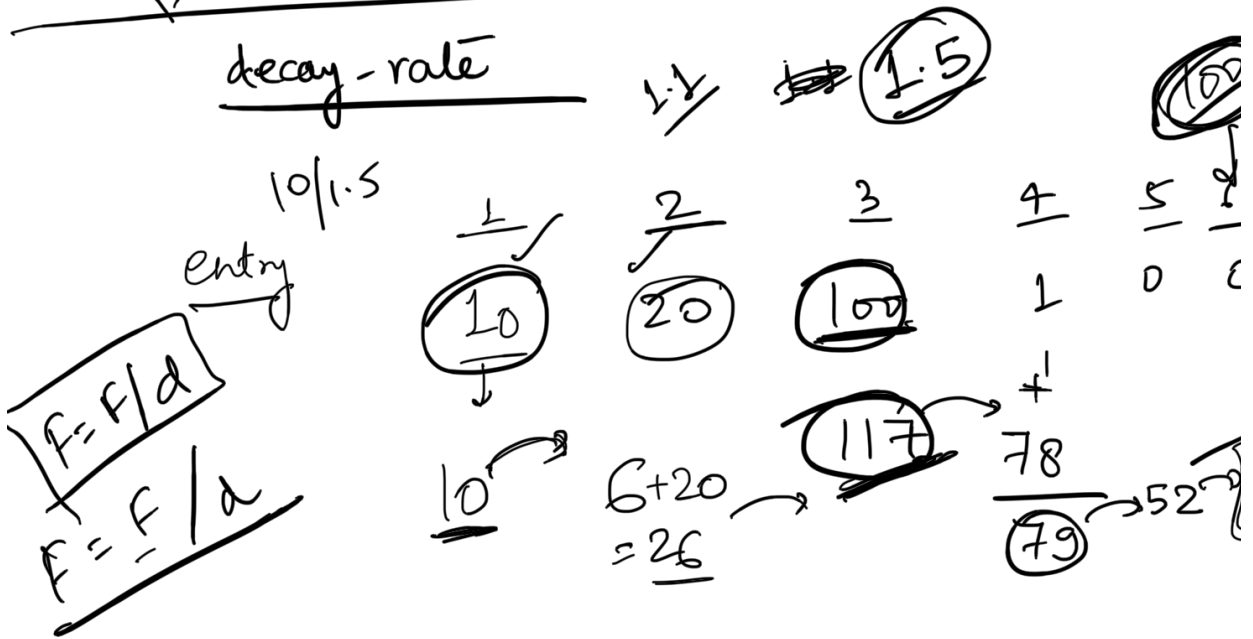
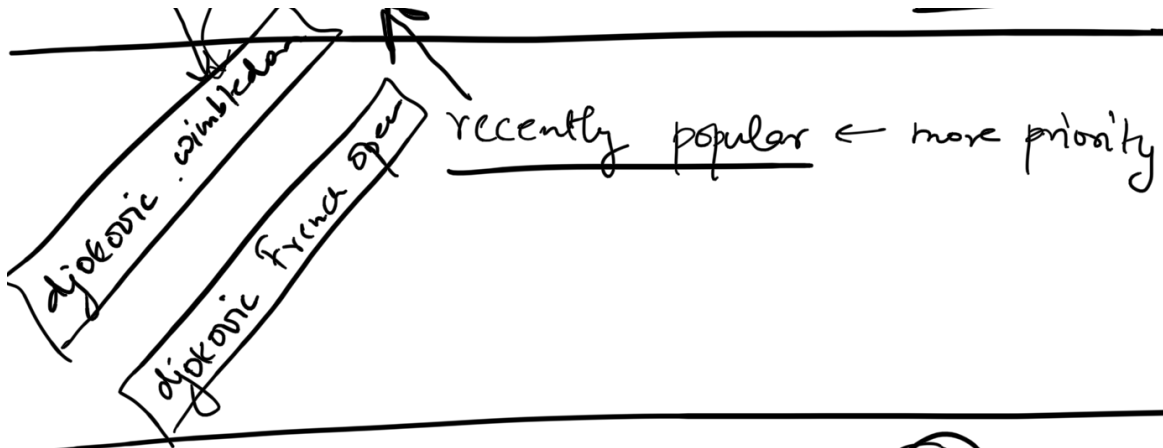
"batman" → (+1)

1000  
2000  
3000

reads  
Top5  
100000

(1/1000)

→ 1000000



✓