Quora

Thursday, 9 December 2021 8:59 PM

MVP
Estimation of Scale
Design Goals
API + Design

MVP

1) Post Question
2) Post Answer
3) Upvoting / Downvoting Answer & Question
4) User Profile
5) News Feed of Q&A ✓ ←
6) Question page with answer in order.
7) Topics for questions
8) Searching Questions → (V1) ✓

Estimation of Scale

DAU = 50 M
= 10K questions per day

Sharding req?
Read/Write Heavy?

Question (text), Answer (text) > high data → 1 day — ≃ 10K questions ✓
1 question — on average 5 answers. ✓
≃ 50K Answers in 1 Day

User Profile, Topics List > less data

Question ≃ 500 Bytes per question
Answer ≃ 2000 Bytes per answer

Data Stored in 1 year → Questions → ~~500~~ B * ~~10~~K * 365 = 1.8 GB
→ Answers → ~~2000~~ B * 50K * 365 = 36.5 GB
≃ 38 GB

KB

10 Years → 380 GB → can be stored in 1 machine.

∴ No sharding is required. ✓

Read Heavy ✓

DAU ≃ 50 M ✓

= 10K questions
+50K
60K

Design choice

Available ✓ or consistent | Latency: low latency is imp. but not critical. ✓

APIs

1) getNewsFeed (userId, num_results, offset) ✓
2) searchQuestion (userId, topic, num_res, offset) ← ✓
3) getQuestionPage (userId, questionId, num_res, offset) ✓
4) AddQuestion (userId, text, ts, questionId) ✓
5) AddAnswer (userId, questionId, text, ts, ansId) ✓
6) upvoteQuestion (userId, questionId, isPositive) ✓
7) upvoteAnswer (userId, answerId, isPositive) ✓

Que —
9:PM 18 Nov.
A1 —
A2 —
A3 —

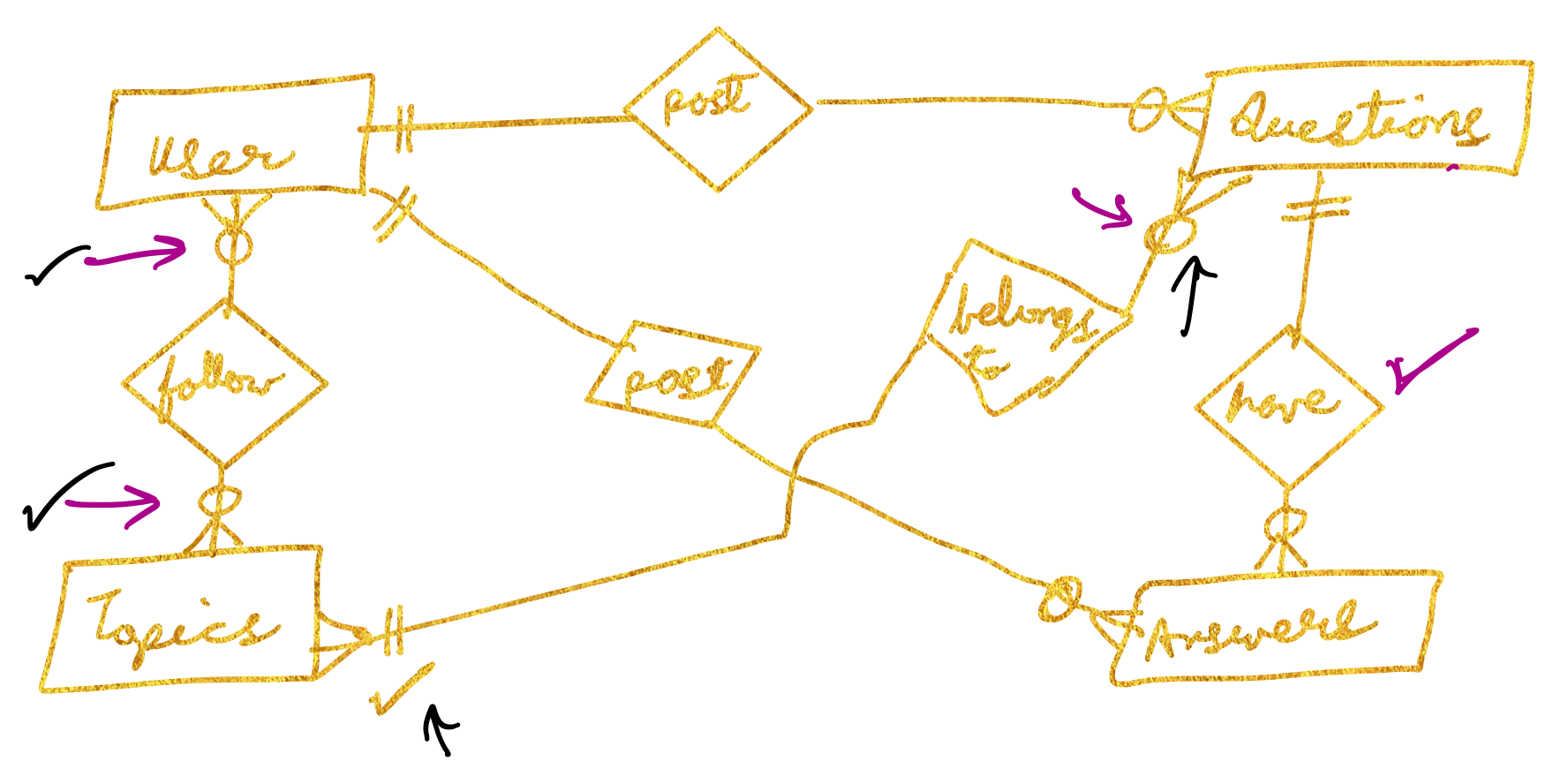
client ts
IP address
random no.

Idempotent Systems ✓

Design

MySQL (Read Heavy)

ER Diagram



user

id	name	email - -

Questions

id	text	ts	userId

Answers

id	text	ts	userId	qid

Topics

id	name	authorid	ts (extra)

Question_topic

qid	tid

user_topic

uid	tid

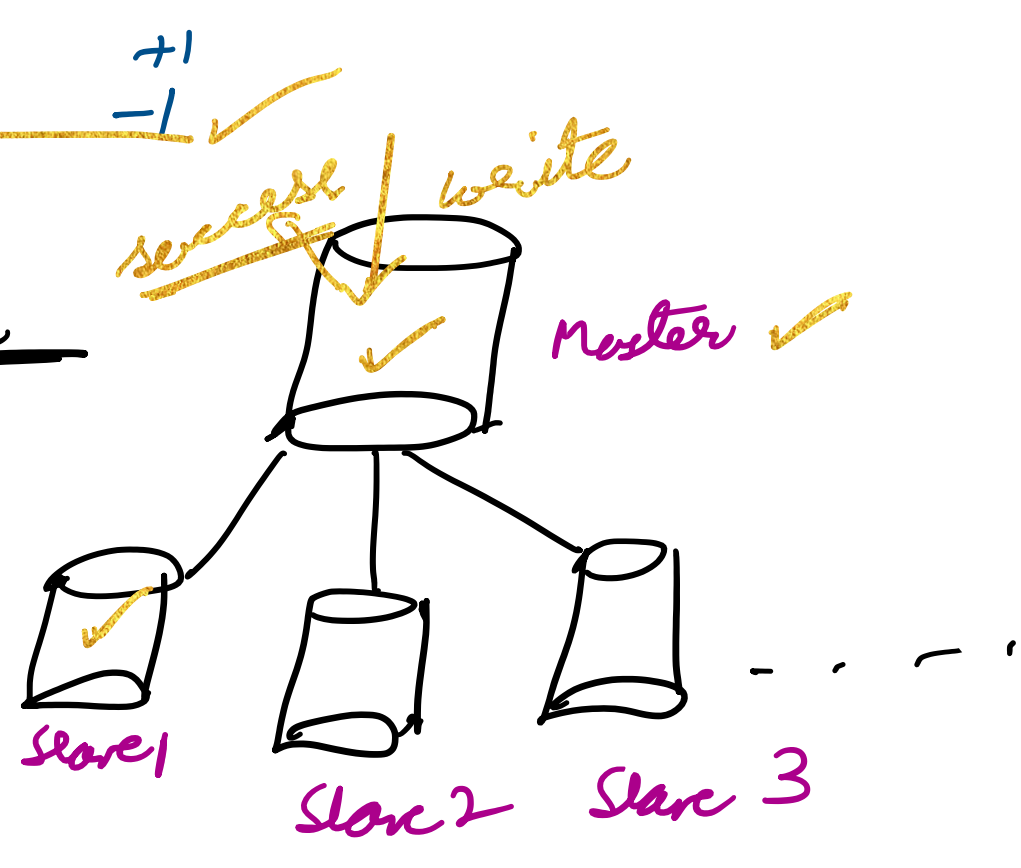
question_vote

qid	userid	isPositive
		+1
		-1

answer_vote

aid	userId	is Positive
		+1
		-1

✓ Replication



Read to all Slaves

getNewsFeed (userId, num_res, offset)

user → topics → 20 most popular questions for a topic → most popular answer

upvotes/downvotes + recency ⇒ decay factor for votes 1.1 per day

⇒ # Sum of followers of upvoted authors

Day 1	Day 2	Day 3

Question → 100 upvotes → 1 year ago
→ 10 upvotes → 1 day ago

upvotes → $\frac{100}{(1.1)^2} + \frac{50}{1.1} + \frac{60}{}$ = 188 ✓

100 + 50 + 60 = 210

Cache (Redis) (Replicas)

[Topic → Top 20 most popular question & Answers] ← 100K ← 50 + votes

50K in total

50K * 20 * 2 = 2000 K entries
≃ 2 M entries ✓

User → followed topics

Update Cache? ✓

How frequent → If the # upvotes is increased by a threshold (50 or 100) ✓

Cache ✓ decay factor (Replicas)

Question_id → # upvotes
Answerid → # upvotes

Top 20
Q1 → 200
Q2 → 150
Q3 → ~~150~~ 200
Q4 → 50

Q3 +50 votes

No. of Questions & Answers
≃ 10K + 50K in 1 days ←
= 60K * 365 in 1 year = 21.9M ✓
10 years → 21.9 * 10 = 219 M ✓

int int int
4B + 4B + 4B = 12B

getQuestionPage (user Id, Question Id, num_res, offset)
→ get all answers ✓

Attributes to consider to make answer relevant

1) upvote/downvote. ✓
2) recency.
3) user can see his/her answer on Top.
4) sum of followers of upvoted authors ✓
5) answers by followed users
6) format/content of answer ✓
→ media
→ only text
→ some keywords related to questions
→ links

A/B testing

different set of users get different order

based on weights of attributes.