

Building a Scalable Social Network Home Timeline

Introduction

You have been tasked with improving the **home timeline** feature for a simplified social network inspired by X (formerly Twitter). In this social network:

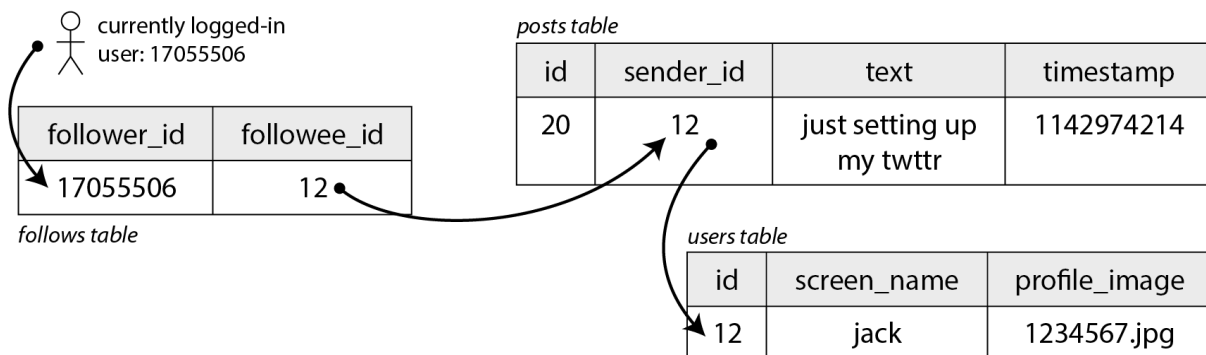
- Users can **post** short messages.
- Users can **follow** other accounts, creating a directed social graph.
- The **home timeline** displays recent posts made by people a user follows.

Despite its simplicity, the system must handle **massive scale**—potentially hundreds of millions of posts per day with spikes of extremely high throughput. The central challenge is ensuring new posts appear in followers' timelines **quickly** (e.g., within a few seconds), even under heavy load.

Below is the **current approach**:

1. Data Storage

- A **relational database** with three tables: **users**, **posts**, **follows**.



2. Querying the Home Timeline

- A SQL query joins the **posts** table with the **follows** table to retrieve posts made by all accounts the user follows, then sorts them by timestamp.
- Clients *poll* this query frequently (every 5 seconds) to remain up to date.

```
SELECT posts.*, users.* FROM posts
  JOIN follows ON posts.sender_id = follows.followee_id
  JOIN users   ON posts.sender_id = users.id
 WHERE follows.follower_id = current_user
 ORDER BY posts.timestamp DESC
 LIMIT 1000
```

You should assume:

- There can be **tens of millions** of users online concurrently.
 - Some users follow thousands, or even millions, of accounts.
 - Posting rates can vary drastically, from a steady background level to large spikes.
-

Your Task

Part 1: Analyze the Current Approach

Goal: Examine how the current system functions and identify where it might struggle.

1. System Representation

- Describe the basic layout of the relational schema and the typical client interaction.
- Where do you see the main *moving parts* in this design?
- Sketch the sequence diagram of showing the users' home timeline.

2. Points of Potential Failure

- Which aspects of the architecture or data model might become bottlenecks as the load increases?
- How might large follower counts or high posting rates affect performance?

3. Reflections on Scalability

- In your view, how well does this approach adapt to sudden spikes in usage?
 - Consider both read (timeline) and write (posting) operations. How might these demands compete or conflict?
-

Part 2: Identify Key Quality Attributes and Write Scenarios

Goal: Pin down the **non-functional requirements** that matter most for this timeline service, and describe how you'd measure success or failure.

1. Quality Attributes

- Based on the context of social media at scale, which **3–5** attributes do you believe are absolutely critical?
- Why are these attributes central to a home timeline experience?

2. Scenarios

- For each attribute, propose at least **one scenario** that tests the system's limits.
- Think about specific conditions (e.g., load spikes, heavy follow/followee relationships, user concurrency).
- Consider how you might verify whether the attribute is met—e.g., a maximum response time, a threshold for data freshness, a system's tolerance for partial failures, etc.

3. Potential Conflicts

- Reflect on whether optimizing for one attribute might compromise another.
 - Which trade-offs would you be prepared to accept, and which do you see as non-negotiable?
-

Part 3: Propose a New Architecture

Goal: Suggest a **revised** design that addresses the scale and responsiveness challenges surfaced in Parts 1 and 2.

1. Core Concepts

- How will your solution ensure that posts appear in a user's timeline quickly, even under high load?
- Are there any **changes** to the data model or approach to serving read queries that might reduce overhead or latency?

2. Architecture Outline

- Identify key **components** that could help.
- Show how **data flows** from the moment a user posts, to when followers see the new content on their home timelines.

- If you rely on new technologies, briefly explain **why** these choices fit your scenarios.

3. Evaluation Against Quality Attributes

- Revisit each scenario from Part 2.
- Argue how (or whether) your proposed design meets the required levels of you defined quality attributes, etc.
- Do you still see any trade-offs or concerns that remain unresolved?

Prepare:

- A **diagram** of your improved system architecture.
- Diagrams that detail how your proposal meets large-scale traffic demands.
- A short rationale linking design decisions to the quality attribute scenarios.

adapted from:

<https://learning.oreilly.com/library/view/designing-data-intensive-applications/9781098119058/ch02.html#id20>