* **“users”** collection

{ "Location": "Seattle" }

* Message
* Getting message includes hello, not showing id, sort by timestamp

Filter : { content: /hello/i }

Project: { senderId: 1, content: 1, timestamp: 1, \_id: 0 }

Sort: { timestamp: -1 }

* **“likes”** collection, **“Aggregations”** tab.

Join likes with users, puts the matching user into a new array field called user

**$lookup**

{

from: "users",

localField: "likeeUserId",

foreignField: "\_id",

as: "user"

}

Turns the user array into a single sub‑document.

If a “like” had matched multiple users, get one pipeline document per matched user.

**$unwind**

"$user"

* Message table
* **Messages whose content contains “hello” (case‑insensitive)**

**$match**

{

"content": {

"$regex": "hello",

"$options": "i"

}

}

**Include sender and conversation IDs, plus the first 30 characters of the content**

**$project**

{

\_id: 0,

senderId: 1,

conversationId: 1,

snippet: {

$substr: ["$content", 0, 30]

}

}

* **Reverse‑chronological (newest first)**

**$sort**

{ timestamp: -1 }