13-07-2025

Agenda :—

→ Non-relational database
→ Statistics - I

RDBMS (postgreSQL) → structured data
→ predifined relationship
→ data integrity via ACID properties.

Why Non-relational Database ?

(1) issue : rigid schema
solution : flexible schema

Admission table

→ id, Name, Clay, DOB → 2024
2025 → + blood group

RDBMS → time consuming, complex operation.

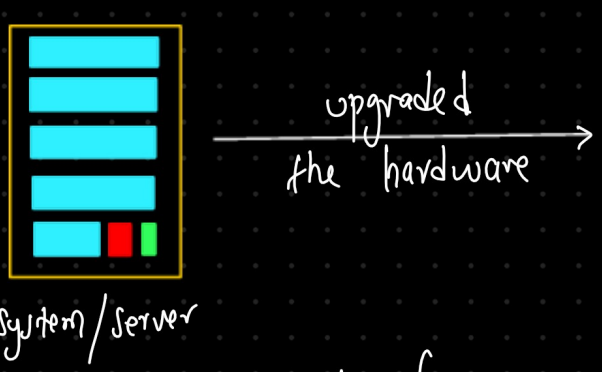
→ id, Name, Clay, DOB → 2024
2025 → + blood group

Non-relational →

(2) Scalability :—

issue :

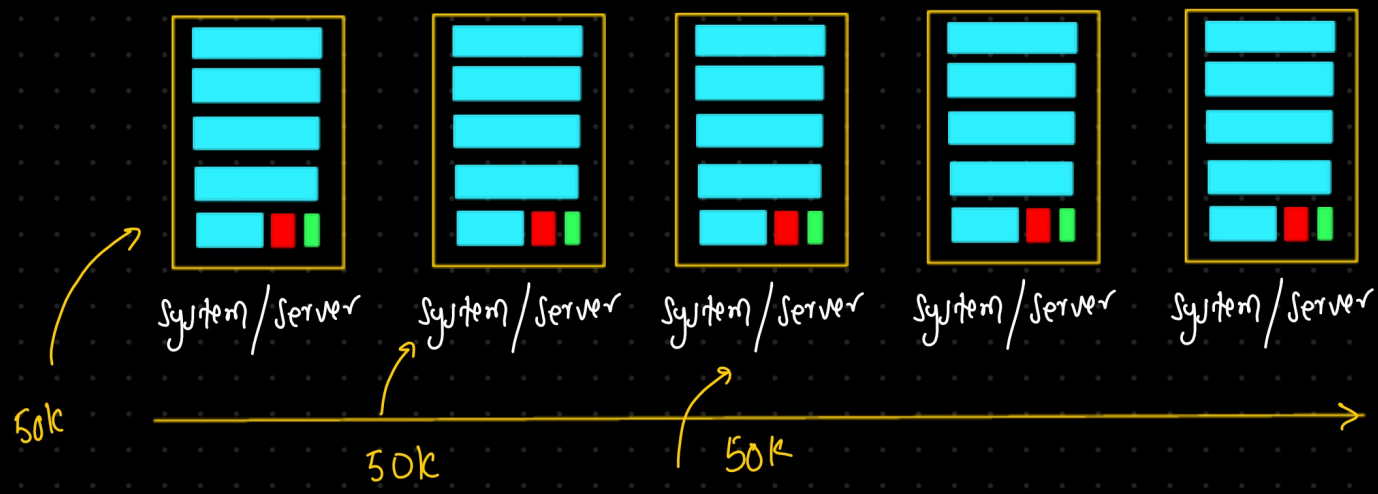
Total no of system : 1



postgres → 3gb → million of user
tool-2 → 3gb
tool-3 → 2gb
CPU → 4 core
ram → 8gb

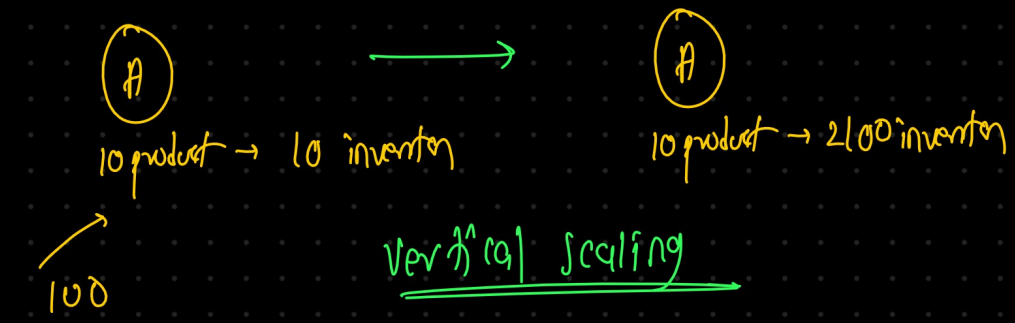
CPU → 12 core
ram → 32gb
postgres → 16gb
tool-1 →
tool-2 →

Vertical Scaling

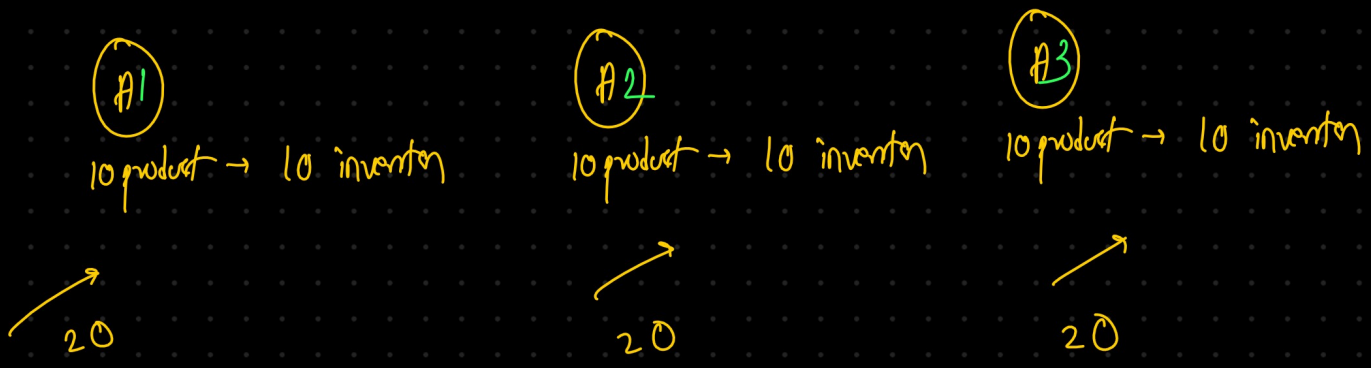


CPU → 4 core
ram → 8gb

Horizontal Scaling



Vertical Scaling



Horizontal Scaling

RDBMS : vertically (scale up)

Non-relation : They are designed to be horizontally scaled.

Disadvantage :—

(1) lacks standard Query language

Mongodb → package

Cassandra → CQL

Redis → Specific command

(2) Lack of strict schema →

→ no data integrity → Application level →

→ harder to debug

refers to the accuracy, consistency, and reliability of data throughout

garbage → AI → garbage

data → process → DB → Non-relational DB

data → DB → Database

update, delete is harder in NoSQL. transaction control like commit/rollback not supported

(3) Limited support for complex join & relationship :

T1:

D1 → get → 1000 row) 1000 row) 1000 row) ← D2 get

T2:

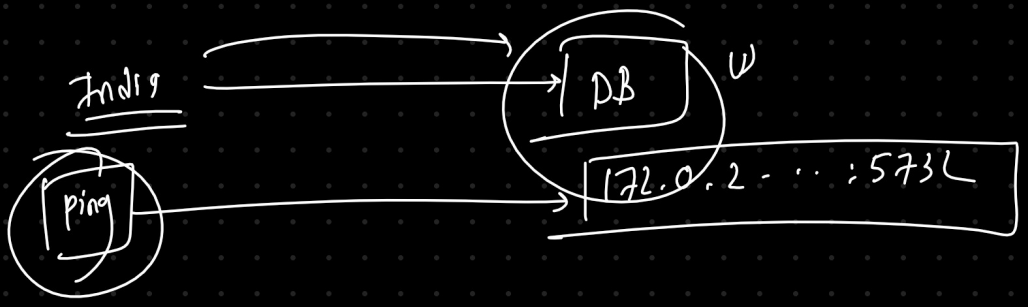
D1 → get → 1000 row) 1020 row 20 ← D2 insert

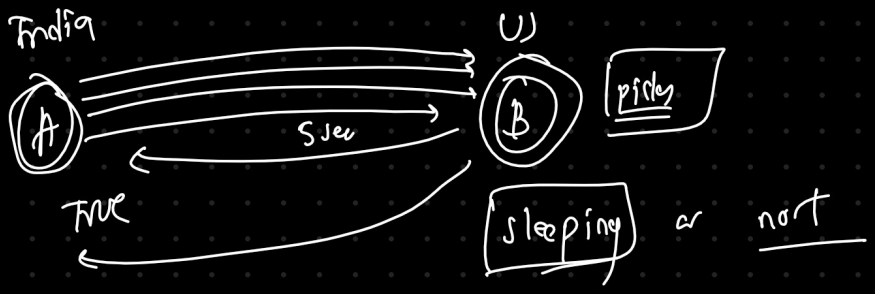
Mongo DB - atlas

project → cluster → load sample data → sample_mflix

db → collection_1, collection_2

RDBMS → db → table, table, table, ...





python → Driver → MongoDB ATLAS

collection.find(query)

① query = {} → find all

② = {"category": "smartphone"} → look for attribute category
↓
get all value with smartphone in category

③ = {"category": "smartphone",
"price": {"$gt": 800}}
↓
greater than

$gt $gte
$lt $lte
$eq
$in

deleteOne → delete the first match
deleteMany → delete all match

→ ~~Manaf~~
→ Manal

updateOne →

updateMany

all_data → get_data → update_operation → update

100 row

iphone → 30 row

100 million

1 million - iphone

5 million - pixel

10 milion → xyz

iphone → 1 million → iphone 16 → iphone 16_new

python numpy pandas DataVisualization Git

Database

Transformed data

CRUD

genAI

path 1 2

Next week - Starts - I → 8 PM IST - 11 PM IST, 1 hour - doub session

- Starts - II →

8 PM IST - 10 PM IST

10 PM IST - 12 PM IST →

* Think about a awesome python project
  - → cool!
  - → Threading (optional)
  - → Database (cloud)
  - → Data visualization (option)

* free to use any library
* Do not use (ML, AI function, opencv, API, UI)
* You can use Gemini, Chatgpt, or any LLM to create project

* Data Scraping library is fine but should not be LLM based.

* Solution acceptance criteria.
  - → provide short 5 minute presentation including code run.
  - → presentation should be max 4 page.
  - → No colab with members, self project