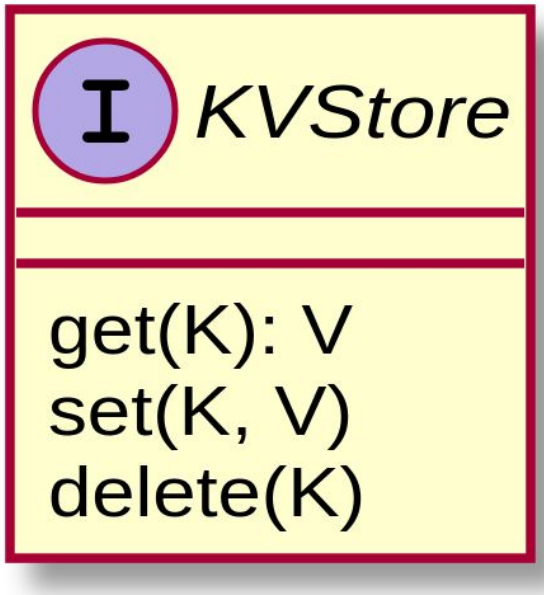
Key-Value Stores

2021

KV Store

1. Paprastas duomenų modelis
   a. Duomenų bazėje saugomos poros (raktas, reikšmė)
   b. Paieška galima tik pagal raktą
   c. Poros tarpusavyje nesusijusios
2. Paprastos operacijos
3. Gali turėti atskirus nepriklausomus porų rinkinius (nors tas nėra būtina)



Paprasta modeliuoti

- Galima sumodeliuoti daugelyje kitų duomenų bazių
  - Reliacinėje: stulpelis su raktu (pirminiu raktu) ir reikšme
  - Dokumentų bazėje (dviejų laukų dokumentas)
  - …
- Bet ribotas panaudojimas

Reikšmės

- Paprasčiausiu atveju reikšmė neturi duomenų tipo
  - T.y. negalima atlikti jokių operacijų su reikšme, kitų nei ją nustatyti ar keisti
- Reikšmės saugomos kaip baitų masyvas ar tekstas

Raktai

- Poros indeksuojamos pagal raktą
  - Maišos (hash) lentelėmis
  - B(+) medžiais
  - LSM medžiais

Surikiuoti raktai

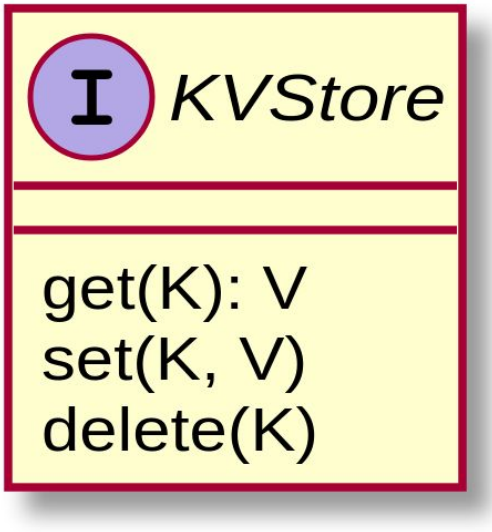
Duomenų bazės, kurios saugo raktus medyje ar sutvarkytus gali turėti papildomų operacijų.

- `getGreater(K)`
- `getGreaterOrEq(K)`
- `getLesser(K)`
- `getLesserOrEq(K)`
- `getBetween(`K_1`, `K_2`)`
- `getStartingWith(subK)`

...

Paprasta padalinti *(angl. “shard”)*

- Raktai nepriklausomi, taigi nesunku duomenis išskirstyti
- Kaip padalinti duomenis tarp N serverių?
  - Hash
    - Sutraukiame raktą į skaičių (pvz. maišos funkcija), raktas priklauso `h(K) mod n` serveriui
    - `h(K)` maišos funkcija raktui, `n` serverio numeris
    - Visose operacijose dalyvauja raktas, taigi paprasta nuspręsti kur siųsti operaciją
  - Ordered
    - Galima padalinti raktus į nepersidengiančius intervalus



Modeliavimas: pasirinkti raktą

- Atominis raktas
  - Asmens duomenys pagal asmens kodą
  - Adresų sąrašas pagal pašto kodą
  - Prekių krepšelis pagal vartotojo ID
- Sudėtinis raktas
  - Asmens informacija pagal vardą pavardę ir gimimo datą (pvz. Jonas-Jonaitis-1980-01-02, Petras-Petraitis-1985-05-19)
  - Forumo komentarai pagal forumo įrašo raktą ir datą (pvz. 27-2018, 27-2017, 42-2018)
    - Tarkime atvejis, kai tinklapis pirma rodo naujausius komentarus, bei leidžia užkrauti senesnius

Kada naudoti?

- Kuomet reikia greito priėjimo pagal konkretų raktą (ar fiksuotus intervalus)
- Atmintyje duomenis laikančios duomenų bazės gali būti panaudojamos greitai prieigai prie sesijos duomenų (vartotojo nustatymai, prekių krepšelis), kuomet yra keli aptarnaujantys serveriai ir sesijos negalima laikyti tiesiog atmintyje
- Atmintyje laikomos duomenų bazės taip pat naudojamos kaip podėlis (*angl. cache*) pasikartojančioms, bet iš disko lėtai aptarnaujamoms užklausoms

Kada nenaudoti?

- Reikalingos atominės operacijų su keliais raktais
- Egzistuoja ryšiai tarp reikšmių
- Užklausos priklauso ne tik nuo rakto, bet ir nuo duomenų
  - Kai tokių užklausų ne daug, galima pakartotinai įrašyti duomenis kitu raktu

Įdomūs atvejai: Berkeley DB

- Atsirado 1994, vis dar aktyviai vystoma
- Oracle NoSQL realizacijos pagrindas
- Turi kelis režimus
  - Paprastą
  - Concurrent - leidžia atlikti izoliuotą operaciją su keliais raktais
  - Transactional - leidžia transakcijas
  - High Availability - leidžia replikavimą

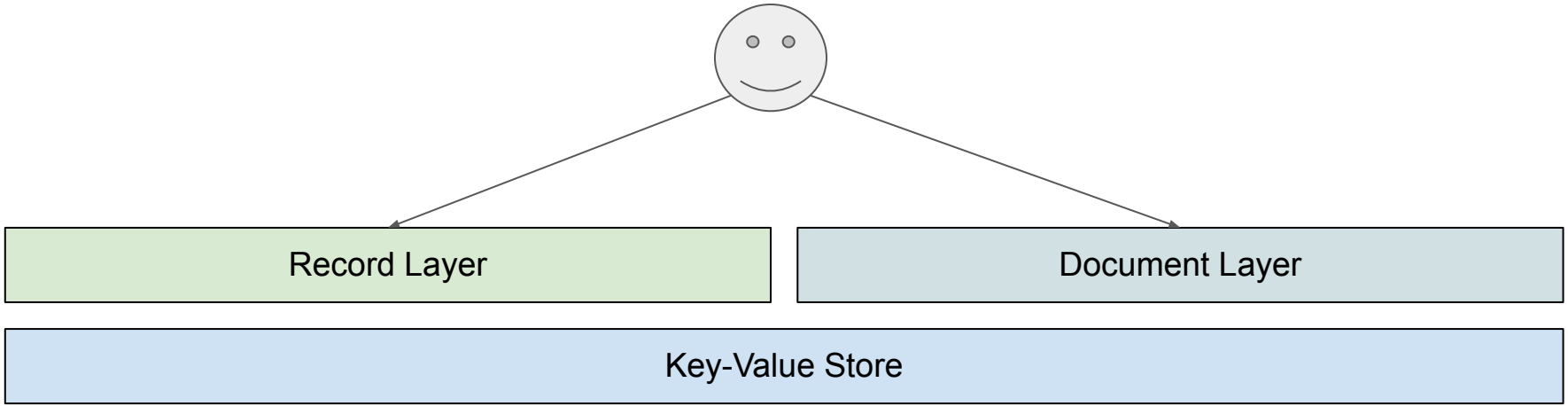
Redis

- Atmintyje laikanti duomenis duomenų bazė
  - Nors turi galimybę visą būseną rašyti į diską: įvykdžius komandą, nurodytais intervalais, su kiekviena operacija
- Ne tik key-value, bet ir data-structure store
  - Raktai gali būti duomenų struktūros - skaitliukai, sąrašai, aibės, žodynai, …
  - Palaiko atomines operacijas su duomenų struktūromis
  - Palaiko raktų galiojimo laiką
  - Palaiko CAS (check-and-set) tipo operacijas
  - Ir daug daugiau: Palaikomos komandos

FoundationDB

Sukurta Apple.

Automatiškai padalina raktus tarp mazgų.



Kiti “primityvai”

LevelDB, RocksDB, ...

Redis priemonės sinchronizacijai

Atominės operacijos

INCR, INCRBY, DECR, DECRBY, SET NX (GET), ...

Transakcijos

Atominės, garantuoja serializaciją

MULTI - pradėti transakciją

WATCH - stebėti reikšmę

EXEC - įvykdyti

DISCARD - atšaukti