

## Relační VS nerelační databáze

- relace
- tabulky
- normální formy
  - duplicitní data => redundance
- integritní omezení (cizí klíče => prázdná relace, číslo ...)
- kopírují chápání tabulky jako abstraktního modelu

## **Nevýhody relačních databází**

- data ve skutečném světě nebývají  
nerenundatní
- škálovatelnost (horizontální a vertikální)
- výkon
- složitá abstrakce dat

## Nerelační databáze

- + Not Only SQL
  - + specifické využití
  - + jednoduché škálování
  - + data jsou “nestrukturovaná”
- 
- každá má svůj “jazyk” pro dotazování
  - redundantní data
  - základní typy: documents, key-value, grafové, column oriented

# Hledejte fulltextově česky, pomocí Lucene a ElasticSearch

- + documented base
- + data v reálném čase
- + škálovatelnost (nodes + shards)
- + REST API (CRUD)
  - + webová služba
  - + lehce multiplatformní

## - Mapping

- můžeme definovat, nebo nechat na databázi (definujeme datové typy, analyzéry, boostování)
- třeba dobře rozvrhnout čím a jak budeme vyhledávat

## - Index

- definuje jednotlivé dokumenty (může obsahovat více typů)
- podobně jako u SQL databázi má něco jako primární klíč

## - Analyzátor (včetně českého)

### - <http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/analysis-intro.html>

- filtry (htmlstrip, & => and)
- tokenizér (úprava řetězce znaků na terms)
- token filtry (úprava termu )
  - hledání kořene nebo základu slova
  - odstranění stopwords (a, nebo, že)
  - hledání synonym
  - přídavná jména

## Vyhledávání

- **query**
  - GET požadavek
  - sofistikovanější JSON dotaz
  - boost
  - must / should / must not
- **filter**
  - vyhledávní nad přesným zněním
- Aleph VS Smartlib

## Key-value databáze - Redis

<http://no.gd/redis-presentation.pdf>

- uložena v operační paměti
- můžeme si představit jako slovník
- atomické operace
- vhodné pro stránkování (pornoservery)
- Ďábelsky rychlé
  - Více než 100k GET/s
  - MySQL max 6k/s

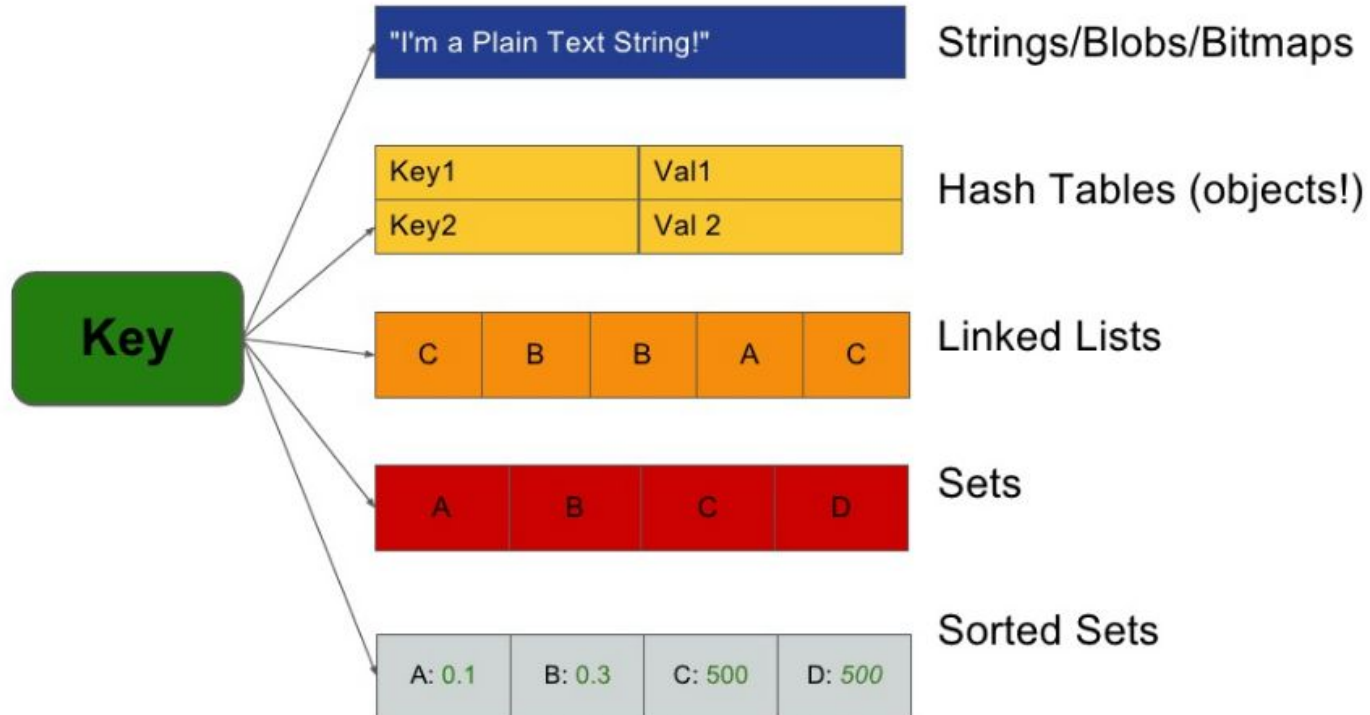
- <http://try.redis.io/>

## **Datové typy**



# Key => { Data Structures }

---



# Klíč - Hodnota

- student:395616:name - Tomáš Hrabal
- books:000995650:title - Nevídaní akademikové
- příkaz - keys \*

## Strings

<http://redis.io/commands#string>

- set jmeno "tomas hrabal"
- get jmeno

## String jako počítadlo

- set pocitadlo 0
- incrby pocitadlo 1
- get pocitadlo

## Expirace klíče

- set hlasovani:395616 1
- get hlasovani
- expire hlasovani 60
- ttl hlasovani:395616
- exists hlasovani:395616

## Seznamy

<http://redis.io/commands#list>

- lpush seznam a
- lpush seznam b
- lpush seznam c
- lrange 0 -1
- lpop seznam
- lrange 0 -1

## Sety

<http://redis.io/commands#set>

SADD ovoce jablko

SADD ovoce hruska

SADD ovoce jablko

SMEMBERS ovoce

SADD zelenina mrkve

SADD zelenina jablko

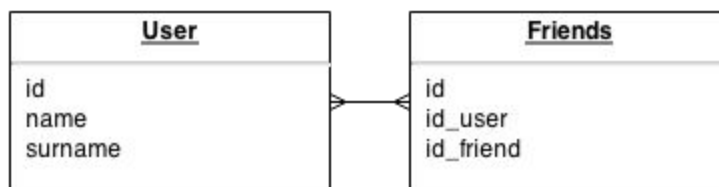
SDIFF ovoce zelenina

SINTER ovoce zelenina

## Hash - něco jak objekt

<http://redis.io/commands#hash>

### Aplikace kamarádi



## **Seznam uživatelů**

- sadd users 1
- sadd users 2
- sadd users 3
- smembers users

## **Detail uživatelů**

- hset user:1 jmeno tom
- hset user:1 prijmeni hrabal
- hset user:2 jmeno jan
- hset user:2 prijmeni mayer
- hset user:3 jmeno jan
- hset user:3 prijmeni pospisil
- hgetall user:1
- hgetall user:2

## **Přátelství**

- sadd user:1:friends 2 (tom kamaradi s jan pospisil)
- sadd user:1:friends 3 (tom kamaradi s jan mayer)
- sadd user:3:friends 1 (jan pospisil kamaradi s tom)
- sadd user:2:friends 1 (tom kamaradi s jan pospisil)

-

## **Společní přátelé:**

- sinter user:1:friends user:2:friends
- 1 .. 3, 2 .. 3

## **Lidi kteří se neznají**

- sdiff

## **Všichni přátele dvou lidí, které zná tom**

- sunion

