

NoSQL, Indexy, subdotazy

## Databáze z 9. přednášky

<http://is.muni.cz/el/1421/jaro2016/VIKBB68/um/62920643/eshop--db.sqlite>

*zahřívací dotaz*

**Kolik knih napsal autor se jménem *Nasim* nebo *Alfonso*? Kniha nesmí mít v názvu písmeno “L” a vyřadte autory, kteří napsali více než 10 knih. Vypište jméno autora, příjmení autora a počet jím napsaných knih.**

# Relační VS nerelační DB

## Vlastnosti relačních databází

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

## Nevýhody

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

# Nerelační databáze

- + Not Only SQL
- + specifické využití
  - stránkování v pornoprůmyslu (<http://blog.togo.io/redisphere/how-youporn-uses-redis-sfw-edition/>)
  - real-time hledání v textu
- + jednoduché škálování
- + data jsou “nestructurovaná”
  
- každá má svůj “jazyk” pro dotazování
- redundantní data
- základní typy: documents, key-value, grafové, column oriented

# Redis

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

# Elasticsearch

- <https://www.zdrojak.cz/clanky/elasticsearch-vyhledavame-cesky/>
- Documented base
- Data v reálném čase (doopravdy v reálném)
- Škálovatelnost
- REST API
  - Webová služba
  - Multiplatformní (protože to běží na webu)
- Umí skvěle hledat i česky
  - hledání založeno na převedení textu na slova => tokeny dle jazyka

# Zpátky k SQL - Subdotazy

- Vložíme dotaz do dotazu, často ho tím velmi zjednodušíme
- Poddotaz musí vracet pouze jeden sloupec (atribut) nad kterým budeme provádět porovnávání pomocí IN / NOT IN

**Vypište zákazníky, kteří si nic neobjednali.**

- standardně
- pomocí IN/NOT IN

Vypište počet nakladatelství bez prodané knihy

Vypište nakladatelství s prodanými alespoň 150 knihami a počet prodaných knih.



# UNION

[http://www.tutorialspoint.com/sqlite/sqlite\\_unions\\_clause.htm](http://www.tutorialspoint.com/sqlite/sqlite_unions_clause.htm)

- Umí slepit dva dotazy dohromady a vypsát pod sebou výsledky ... praktické uplatnění ale není moc valné

**Vypiš knihy začínající na j a pod to knihy začínající na l pomocí UNION**

# Indexy

- [https://cs.wikipedia.org/wiki/Index\\_%28datab%C3%A1ze%29](https://cs.wikipedia.org/wiki/Index_%28datab%C3%A1ze%29)
- Bez indexů bychom ve velkých databázích čekali na výsledky třeba i celou věčnost
- Index je v databázi vždy nad
  - Primární klíč
  - Cizí klíč
  - Unikátní klíč
  - Fulltext
  - Vyhledávání (např. pokud víme že se bude často vyhledávat podle jména)
- Nevýhodou je větší čas zápisu a editace záznamu (index se musí přeskládat)
  - Proto nemůžeme zaindexovat všechny sloupce. Vyhledávali bychom rychle, ale u velkých databází by naopak mohlo trvat věčnost přidat/změnit záznam

# Pohledy, materializované pohledy, EXPLAIN

- VIEW
  - Virtuální tabulka vytvořená “nějakým” dotazem
  - [http://www.w3schools.com/sql/sql\\_view.asp](http://www.w3schools.com/sql/sql_view.asp)
  - Při každém zobrazení view se spustí definovaný dotaz

**Pohled nad knihami začínajícími na “A”. Po vytvoření si musíte v SQLite manageru najít v levém menu záložku VIEWS**

```
CREATE VIEW knihy_zacinajici_na_a AS SELECT id, nazev FROM knihy WHERE nazev LIKE "a%"
```

- MATERIALIZED VIEW
  - Umí jen některé databáze, data v materializovaném pohledu se nemění aktuálně se změnami v databázi, je to taková cache

# EXPLAIN

- Pokud před dotaz přidáme, můžeme se dozvědět více o tom jak je dotaz spuštěn, jaké využívá indexy, případně jak pracuje s námi zadaným dotazem

```
EXPLAIN SELECT z.id_zakaznik, z.jmeno, z.prijmeni FROM zakaznici z  
WHERE z.jmeno LIKE "%a%" OR (z.jmeno LIKE "%b%" AND z.jmeno NOT  
LIKE "%a%")
```

# Google query language

- Dokumentace

<https://developers.google.com/chart/interactive/docs/querylanguage>

- Testovací data

<http://jdem.cz/b8s5v6>

Vybrat rozsah:

```
=QUERY(ceniks!G5:G6; "SELECT AVG(G)");
```

Agregační funkce

```
=QUERY(ceniks!A:Z; "SELECT AVG(G), SUM(G), COUNT(G)");
```

GROUP BY

```
=QUERY(ceniks!A:Z; "SELECT R, COUNT(G) GROUP BY R ORDER BY COUNT(G) DESC");
```

PIVOT

```
=QUERY(ceniks!A:Z; "SELECT SUM(G) PIVOT R ");
```

GRAFY

```
=QUERY(ceniks!A:Z; "SELECT R, COUNT(G) GROUP BY R ORDER BY COUNT(G) DESC  
LIMIT 25");
```