

Databázové systémy

DBS 2018

Úvod do predmetu

Prednášky

- Michal Barla
- <https://dbs-fiit.github.io>
(resp. <http://fiit.stuba.sk/DBS>)
- michal.barla@stuba.sk
- ~~3.03~~ (ozvite sa mailom, nie som full-time FIIT)
- 12 x 2 hodiny

Q&A?

[www.slido.com#DBS01](https://www.slido.com/#DBS01)

<https://askalot.fiit.stuba.sk>

Cvičenia

- Ing. Tomáš Kramár, PhD. (ext)
 - Miroslav Demovič, MBA (ext)
 - Ing. Kristián Košťál
 - Ing. Samuel Pecár
 - Ing. Matúš Pikuliak
 - A ja (ext)
-
- Nebojte sa **včas** eskalovať prípadné problémy

Náplň predmetu

- budeme sa venovať DBS z pohľadu návrhára, vývojára a používateľa databázovej aplikácie
 - ktorá rieši perzistenciu pomocou databázy

Náplň predmetu

- budeme sa venovať DBS z pohľadu návrhára, vývojára a používateľa databázovej aplikácie
 - ktorá rieši perzistenciu pomocou relačnej databázy

Motivácia

- Každý program pracuje iba s údajmi načítanými v pamäti
- Potrebujeme riešiť perzistenciu údajov
- Medzi behmi programu
- Medzi reštartmi počítača (servera)

Jednoduché riešenie

- Pri štarte načítať zo súboru
 - Pri ukončení zapísať do súboru
 - Pri výpadku elektriny...?
 - Budeme zapisovať po každej zmene?
-
- Ako dlho to bude štartovať a končiť?
 - Zmestí sa to do pamäte?
-
- Škálovanie?

Jednoduché riešenie pohľad vývoja

- V akom formáte to budem ukladať?
- Musím si napísať “obslužný kód”
(serializér/deserializér, ...)
- V každom projekte?
- Spravím si vlastnú libku?
- A príde nový programátor do firmy...

Motivácia - čo potrebujeme

- efektívne
- spoľahlivé
- vhodné
- bezpečné
- viac-používateľske
- ukladanie a prístup k veľkému množstvu perzistentných dát

Relačné databázy to riešia

- veľké množstvo dát – gigabajty (to **nebudete** mať v pamäti)
- perzistencia – dáta prežijú reštart aplikácie
- bezpečne – dáta sa nestratia, budú konzistentné a odolajú zlyhaniu (sw, hw, používatelia)
- viac-používateľské – používatelia si neprepisujú dáta, zvládajúce paralelné prístupy
- vhodné/pohodlné – nezaujíma nás fyzické uloženie/reprezentácia dát, vysoko-úrovňové dopytovanie (deklaratívne dopytovanie)
- efektívne – brutálne rýchle
- spoľahlivé – nepadajú po každom treťom dopyte

Kľúčové pojmy

- Dátový model
- schéma vs. dáta
 - typy a premenné
- jazyk na manipuláciu schémy
 - data definition language (DDL)
- jazyk na manipuláciu premenných
 - data manipulation language (DML)

Čomu sa teda budeme venovať

- Dátové modelovanie
- Relačný model, relačná algebra
- Jazyk SQL (DDL, DML)
- O/R mapovanie
- Normalizácia dátových modelov
- RDMS – ako to funguje vo vnútri
- Indexy, optimalizácia dopytov
- ACID, Transakcie, Paralelizmus

Relačné databázy to riešia?

- veľké množstvo dát – gigabajty (to nebudete mať v pamäti)
- perzistencia – dáta prežijú reštart aplikácie
- bezpečne – dáta sa nestratia, budú konzistentné a odolajú zlyhaniu (sw, hw, používatelia)
- viac-používateľské – používatelia si neprepisujú dáta, zvládajúce paralelné prístupy
- vhodné/pohodlné – nezaujíma nás fyzické uloženie/**reprezentácia dát**, vysoko-úrovňové dopytovanie (deklaratívne dopytovanie)
- efektívne – brutálne rýchle
- spoľahlivé – nepadajú po každom treťom dopyte

Čomu sa teda budeme venovať

- Dátové modelovanie
- Relačný model, relačná algebra
- Jazyk SQL (DDL, DML)
- O/R mapovanie
- Normalizácia dátových modelov
- RDMS – ako to funguje vo vnútri
- Indexy, optimalizácia dopytov
- ACID, Transakcie, Paralelizmus
- **Nerelačné databázové systémy (NoSQL)**

Ako budú vyzerat' cvičenia

- riešenie malých príkladov na témy z prednášok
- Mini-testy
- konzultácie a odovzdanie projektu
- Mini-testy
 - Počas niektorých (nie všetkých) cvičení
 - Spolu za 15 bodov
 - krátke úlohy/príklady na témy z cvičení, za 2-3 body
 - **Je potrebné z nich získať v súčte aspoň 6 bodov**

Projekt

- za 30 bodov, riešený vo dvojici
- databázová aplikácia
 - Postgres, MySQL alebo MS-SQL (pre C#istov)
 - ľubovoľný programovací jazyk/framework
 - Ale prosím, nie taký, s ktorým nemáte skúsenosti
 - musí mať GUI
- Štyri checkpointy, tri hodnotené, dva povinné
 - Model, dáta + analytický report – 10 bodov, povinne
 - Jeden scenár – 4 body, nepovinne
 - Kompletná appka – 16 bodov, povinne

Trošku k architektúre

- ~~Krieda?~~

Projekt – náplň

- Vami zvolená doména
 - Nutné odkonzultovať s cvičiacim
- Implementácia kompletného dátového modelu
- Dáta
- Report o dátach
- Implementácia definovaných scenárov
 - Napr. “ako vedúci ZOO chcem pridať nové zviera do evidencie”
- Dokumentácia
 - Čo by ste chceli vedieť o projekte, v ktorom máte pokračovať vo vývoji?

Model

- Min. 7 tabuliek, bez zarátania číselníkov
 - Ideálne nie viac ako 9
- Naplnený dátami
 - Veľa dát, milióny riadkov
 - Ľubovoľne, ale vierohodne
- Analytický report
 - Sumarizácia numerických atribútov
 - Distribúcia nominálnych atribútov
 - Bonus: histogramy, boxploty, ...

Aplikácia – scenáre všeobecne

- Zobrazenie prehľadu všetkých záznamov (rozumne stránkovaných, ak je to potrebné), kde v každom riadku figuruje aj číslo, pre ktorého výpočet sa využíva GROUP BY,
 - extend podscenár: Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom.
- Zobrazenie konkrétneho záznamu - detail,
- Vytvorenie nového záznamu,
- Aktualizácia existujúceho záznamu,
- Vymazanie záznamu.
- Odkonzultovanie a schválenie cvičiacim

Aplikácia - implementácia

- Funkčnosť aplikácie je nutná,
 - ale nie postačujúca podmienka
- Dôraz na efektívnosť práce s databázou
 - Z pohľadu využitia SQL
 - Z pohľadu softvérového návrhu
- Použiteľnosť aplikácie
 - Použiteľné rozhranie
- Vaša aplikácia
 - musí obsahovať JOINy (namiesto cyklenia cez záznamy)
 - musí použiť agregáčnú funkciu a GROUP BY
 - musí používať transakcie

Projekt - odovzdanie

- Jednotlivé checkpointy do AIS
 - Lebo archivácia
 - Lebo tam vieme riešiť deadline
- Zip v AIS obsahuje referenciu na git commit
 - Pozor na to, 23:59 by som to neriešil
- Pushnutý na GitHub
- Otvorený pull request
- Potrebujeme k tomu samostatný seminár?

Projekt – neskoré odovzdanie?

- Nepovinné checkpointy
 - Bez možnosti neskorého odovzdania
- Povinné checkpointy
 - Za každý deň oneskorenia dole o švrtinu
 - 0.75, 0.5, 0.25, F_x

Prehľad

- Priebežné mini testy ==> 15 bodov
- Projekt ==> 30 bodov
- Max. zápočet ==> 45 bodov
- Min. zápočet ==> 20 bodov
 - **Z testov min. 6 bodov**
- Skúška (test cez AIS) ==> 55 bodov
- Spolu klasických 100 bodov