

Tehnici Avansate de Programare

Baze de date și Gestiunea sarcinilor de lucru

Petru Rebeja, Marius Apetrii

16 Aprilie 2020

Facultatea de Matematică

Universitatea Alexandru Ioan Cuza, Iași

Introdurre

Recapitulare

- **Unit of Work** — un șablon care ne permite să executăm toate modificările aferente bazei de date într-o singură tranzacție.
- **Dependency Injection** — o modalitate de a-i da unei instanțe variabilele de care aceasta are nevoie separând astfel crearea de instanțe de utilizarea lor.
- **Test-Driven Development** — un stil de dezvoltare software în care mai întâi se scriu testele pentru un anumit aspect iar mai apoi implementarea propriu-zisă.

Agenda

- Baze de date
- Istoricul schemei bazei de date relaționale
- Gestiunea sarcinilor de lucru într-un proiect

Baze de date

Bază de date

O bază de date este o colecție organizată de date care sunt stocate și accesate de pe un calculator¹.

¹<https://en.wikipedia.org/wiki/Database>

Sistem de Gestiune al Bazelor de Date

Sistemul de Gestiune a Bazelor de Date este un sistem software care le permite utilizatorilor să definească, să creeze, să întrețină și să controleze accesul la baza de date².

²Connolly, Thomas M.; Begg, Carolyn E. (2014). Database Systems – A Practical Approach to Design Implementation and Management (6th ed.). Pearson. p. 64. ISBN 978-1292061184.

Schema bazei de date

Schema bazei de date este structura logică a bazei de date descrisă într-un limbaj formal suportat de SGBD³ sau o reprezentare vizuală a acesteia⁴

³http://en.wikipedia.org/wiki/Database_schema

⁴<https://www.techopedia.com/definition/30601/database-schema>

Tipuri de baze de date

Există două tipuri de baze de date:

- Relaționale (SQL) și
- Non-relaționale (No-SQL)

Baze de date relaționale

- Sunt bazate pe tabele și modelează relația dintre ele,
- Au o schemă predefinită,
- Suportă interogări complexe,

Baze de date relaționale (cont.)

- Suportă creșterea pe verticală (`vertical scaling`) prin adăugarea de memorie RAM, spațiu pe disk etc.,
- Pun accentul pe proprietățile ACID:
 - `Atomicity` — modificări atomice,
 - `Consistency` — impune consistența datelor
 - `Isolation` — modificările se fac în izolare unele față de altele
 - `Durability` — modificările sunt salvate pe disk.

- Sunt bazate pe documente, grafuri, perechi cheie-valoare etc.
- Nu au o schemă predefinită,
- Au suport limitat pentru interogări complexe,

- Suportă creșterea pe orizontală (horizontal scaling) prin adăugarea de noduri noi,
- Aplică teorema CAP⁵ — în orice moment oferă două proprietăți din următoarele:
 - Consistency — orice scriere primește cele mai recente date sau o eroare,
 - Availability — fiecare cerere primește un răspuns dar pot să nu fie cele mai recente,
 - Partition tolerance — sistemul continuă să funcționeze în ciuda pierderii unor mesaje.

⁵https://en.wikipedia.org/wiki/CAP_theorem

Evoluția bazei de date

- Baza de date evoluează (de obicei) împreună cu aplicația,
- Modificările bazei de date fac parte din ciclul de dezvoltare.

- Schema bazei de date trebuie păstrată în sistemul de management al istoricului⁶ pentru a asigura sincronizarea între modificările aplicației și a bazei de date,
- Aplicarea modificărilor trebuie sincronizată,
- Altfel întregul sistem software devine inutilizabil.

⁶[https:](https://www.troyhunt.com/10-commandments-of-good-source-control/)

[//www.troyhunt.com/10-commandments-of-good-source-control/](https://www.troyhunt.com/10-commandments-of-good-source-control/)

Gestiunea sarcinilor de lucru

Dezvoltarea unui sistem software implică existența următoarelor componente:

- Lista de cerințe/defecte (product backlog),
- Fluxul de lucru în dezvoltare,
- Documentația,
- Ciclul de lansare.

Product backlog

Este o listă de cerințe și defecte ale sistemului software, ordonată după valoarea adăugată, risc și prioritate.

- Product Backlog Item = o cerință sau un defect din listă.

- Totalitatea stărilor prin care trece o sarcină. Ex. (Open, In Progress, Closed).
- Sarcinile sunt afișate pe o tablă separată în coloane cu o coloană pentru fiecare stare.
- La trecerea dintr-o stare în alta sarcina este mutată în coloana respectivă.

- Trebuie păstrată la minimul necesar și ajustată o dată cu modificările aplicației.
- Trebuie păstrată în sistemul de gestiune al istoricului:
 - Fiind aproape de cod este (mai) ușor de întreținut,
 - Trebuie să fie într-un format care să permită revizuirea modificărilor (ex: `markdown`).

- Poate fi periodic sau sporadic,
- Necesită planificarea sarcinilor și modificărilor pentru fiecare versiune.
- Sarcinile planificate sunt puse într-o listă nouă sau etichetate pentru a putea fi filtrate.

Demonstrații

Înceiere

Recapitulare — baze de date

- **Baza de date** este o colecție organizată de date care pot fi manipulate prin intermediul unui SGBD.
- **SGBD** = Sistem de Gestiune al Bazelor de Date; permite manipulearea datelor și întreținerea bazelor de date.
- **Schema bazei de date** este reprezentarea structurii bazei de date și trebuie păstrată în sistemul de gestiune al istoricului alături de codul-sursă al aplicației.
- Folosiți Database project din Visual Studio pentru modificarea schemei bazei de date.

- Păstrați o listă a sarcinilor de lucru a proiectului,
- Definiți un flux de lucru,
- Păstrați un minim necesar de documentație,
- Concentrați-vă asupra sarcinilor planificate pentru versiunea nouă.

Succes la evaluare!