TIPURI DE CONCURENȚĂ VIZ-A-VIZ DE MODIFICAREA DATELOR UNEI APLICAȚII

Tema 2 DATC

Concurența, în contextul aplicațiilor software, se referă la manipularea mai multor utilizatori care încearcă să acceseze aceleași date în același timp.

Dacă luăm drept exemplu un sistem de procesare a comenzilor care permite mai multor utilizatori să adauge sau să editeze comenzi, se poate afirma că adăugarea de noi comenzi de către mai mulți utilizatori în același timp nu reprezintă o problemă, pentru că înregistrările noi inserate nu interacționează unele cu celelalte. Pe de altă parte, editarea unor comenzi poate conduce la o problemă, în sensul că doi utilizatori finali ar putea edita și transmite către server o înregistrare în același timp. Cum este manipulată acestă situație depinde de tipul de concurență care este implementată în cadrul aplicației.

Tipuri de concurență:

- Optimistă
  În modelul concurenței optimiste utilizatorilor le este permis întotdeuna să citească datele și chiar să le actualizeze. Când un utilizator încearcă să actualizeze anumite informații, sistemul realizează totuși o verificare pentru a determina dacă vreun alt utilizator a realizat vreo modificare asupra datelor de când acestea au fost preluate de utilizatorul actual. Dacă este determinata vreo neconcordanță, încercarea de actualizare este respinsă. Un avantaj al acestui tip de concurență este reprezentat de faptul că toate datele sunt disponibile către toți utilizatorii mereu. Dezavantajul surprinde timpul pierdut cu realizarea unor modificări asupra informațiilor conținute de aplicație, modificări care nu vor fi acceptate, astfel încât trebuie reluat procesul.
- Pesimistă
  Acest model de concurență realizează blocarea datelor. Dacă un utilizator are deschisă o înregistrare pentru editare, atunci aceasta va fi blocată și oricare alt utilizator care încearcă să acceseze aceleași date pentru a le actualiza va primi un mesaj de informare că datele sunt în curs de modificare. Utilizarea acestei concurențe prezintă un mare avantaj în cazurile aplicațiilor în care sunt foarte probabile aparițiile unor încercari simultate de modificare a datelor. Dezavantajul este reprezentat de faptul că toți ceilalți utilizatori nu au acces la date în timpul în care un utilizator le editează.

- Ultimul câștigă
  În cazul acestui model de concurență, nicio verificare nu este făcută când se încearcă modificarea unor informații de către un utilizator. Când mai mulți utilizatori încercă să actualizeze aceleași date, acestea vor putea fi suprascrise de fiecare în parte, sistemul păstrând ceea ce ultimul dintre utilizatori a modificat.

Conflicte de concurență la utilizarea bazelor de date în viața reală

Cele mai multe aplicații care utilizează baze de date sunt aplicații multi-utilizator, ceea ce înseamnă că la orice moment de timp mai multe persoane ar putea executa operații de citire sau de scriere în baza de date.

Având în vedere aceasta, este foarte probabilă apariția unei situații în care doi sau mai mulți utilizatori vor încerca să execute operații asupra aceleiași înregistrări. O operație completă de modificare a unor informații presupune trei pași: citirea în memorie, actualizarea în memorie, scrierea în baza de date. Existența unei concurențe de tipul "ultimul câștigă" va determina ca în baza de date să fie salvate doar ultimele modificări încărcate.

Acest tip de probleme determinate de execuții concurențe pot apărea atât în cazul proceselor umane, cât și în cazul celor automate sau combinații de ambele. În cazul proceselor automate este mai greu de rezolvat o astfel de problemă decât în cazul utilizatorilor umani care pot fi întrebați dacă vor să suprascrie datele actualizate de alt utilizator. Rezolvarea acestor situații se poate face prin implementarea concurenței pesimiste sau a celei optimiste.

O situație reală poate referii un transfer bancar între conturile a două persoane. Putem lua în considerare ce s-ar afișa în cazul în care persoana care realizează transferul și persoana căreia îi sunt transferați banii fac verificări ale soldului disponibil, sau ce s-ar întâmpla dacă persoana căreia i se realizează transferul vrea să facă o retragere de bani în timp ce tranzacția are loc. Aceste sunt exemple de tranzacții interdependente (depind unele de celelalte).

Un alt exemplu tipic poate referii situația în care o înregistrare a fost ștearsă de un utilizator, în timp ce un altul a realizat o actualizare a acesteia. Pentru a astfel de conflicte sunt implementate blocări asupra bazelor de date sau doar asupra unor înregistrări. Este recomandat ca tranzacțiile să fie cât mai scurte posibil pentru a păstra timpul în care resursele sunt blocate cât mai mic.

Un alt aspect care surprinde aplicațiile cu baze de date în viața reală se referă la faptul că nu se urmărește doar adăugarea sau modificarea unei singure înregistrări într-o tabelă, ci actualizarea mai multor înregistrări, tabele sau, uneori, chiar a mai multor baze de date.

Astfel, aplicațiile pot avea două tipuri de date:
- date simple, care actualizează o singură linie într-o tabelă
- date complexe, care realizează modificări asupra mai multor linii/tabele

Testarea apariției unor abateri ale concurenței optimiste

Exită mai multe metode de testare:

→ una implică includerea unei coloane noi în baza de date care conține un timestamp care conține data și ora ultimei actualizări a înregistrării. Astfel, modificarea unei înregistrări presupune și actualizarea coloanei corespunzătoare timestamp-ului. Când se încearcă realizarea unei actualizări se compară valoarea timestamp-ului din baza de date cu valoarea timestamp-ului din înregistrarea modificată. Dacă aceste sunt egale, are loc actualizarea urmată de înlocuirea timestamp-ului cu ora actuala.

→ o altă metodă presupune verificarea dacă anumite valori din coloanele unei înregistrări se potrivesc cu cele din baza de date:

```
SELECT Col1, Col2, Col3 FROM Table1
```

```
UPDATE Table1 Set Col1 = @NewCol1Value,
              Set Col2 = @NewCol2Value,
              Set Col3 = @NewCol3Value
WHERE Col1 = @OldCol1Value AND
      Col2 = @OldCol2Value AND
      Col3 = @OldCol3Value
```

Iusco Cosmina – Mihaela
An 4, IS, grupa 3.1