



Cap.5. Gestiunea tranzactiilor

CUPRINS

1. Tranzactii
2. Anomaliile de acces concurent la BD
3. Proprietatile tranzactiilor
4. Operatiile efectuate de tranzactii
5. Starile tranzactiilor
6. Planificarea tranzactiilor
7. Tehnici de control al concurenței
8. Tehnici de refacere a BD

1



1. Tranzactii

DEFINITII

Tranzacție (transaction): o unitate logică de prelucrare indivizibilă (atomică) a datelor unei BD.

O tranzacție trebuie să asigure consistența BD în diferite situații:

- tranzacția se execută individual sau concurent cu alte tranzacții
- apar defecte ale sistemului în cursul execuției tranzacției

Tranzacție : operație indivizibilă de acces la BD care fie :

- execută cu succes toate acțiunile și se termină cu o validare** a modificărilor efectuate asupra BD (commit)
- nu poate executa** (din diferite motive) **toate acțiunile și este abandonată și anulată** (abort, rollback)



1. Tranzactii

DEFINITII

Acces concurent la BD: un sistem SGBD deservește mai mulți utilizatori, care accesează **concurent** datele din tabele.

Execuția concurentă a mai multor procese poate avea loc în :

- sistem uniprocessor:** prin partajarea timpului de execuție al procesorului între mai multe procese (**multiprogramare**)
- sistem multiprocessor:** mai multe procese pot fi executate în mod real simultan, pe mai multe procesoare ale sistemului (**multiprocesare**)



2. Anomaliile de acces concurent la BD

Operațiile de acces la un articol X al bazei de date pot fi:

- read(X):** citește articolul X din BD într-o variabilă a programului;
- write(X):** scrie variabila de program în articolul X al BD.



Tranzacțiile lansate de diferiți utilizatori se pot executa concurent și este posibil să actualizeze aceleași articole ale BD

Dacă execuția concurentă a tranzacțiilor este necontrolată, este posibil ca BD să ajungă într-o stare inconsistentă (incorectă), chiar dacă:

- fiecare tranzacție în partea fost executată corect
- nu au apărut defecte de funcționare ale sistemului



2. Anomalii de acces concurrent la BD

Motive care pot cauza anomalii:

- Actualizare pierduta** (lost update): o tranzactie foloseste valoarea initiala a unui articol care intre timp a fost modificata de o alta tranzactie
- Citire improprie** (dirty read): una din tranzacții este abandonată, iar altă tranzacție concurrentă a utilizat articolele modificate, înainte de readucerea acestora la valoarea inițială.
- Citire irepetabilă** (nonrepetable read): o tranzacție citește un articol de 2 ori, iar între cele 2 citiri, o altă tranzacție a modificat chiar acel articol
- Citire fantomă** (phantom read): o tranzacție prelucrează un set de linii rezultat al unei interogări și în timpul acestei prelucrări o altă tranzacție inserează sau șterge o linie



3. Proprietatile tranzactiilor

DEFINITII

Atomicitatea (atomicity): **proprietatea unei tranzacții de a reprezenta o unitate de execuție indivizibilă** (adică de a executa “totul sau nimic”)

Consistența (consistency): **proprietatea unei tranzacții de a efectua modificări corecte ale BD:**

- o tranzacție transformă BD dintr-o stare consistentă în altă stare consistentă
- starea unei BD este consistentă dacă respectă toate constrângerile de integritate implicite sau explicite

Izolarea (isolation): **proprietatea unei tranzacții de a face vizibile modificările efectuate numai după ce a fost validată** (committed)

Durabilitatea (durability): **după validarea unei tranzacții, modificările efectuate în BD nu vor mai fi pierdute datorită unor defectări ulterioare a sistemului**



4. Operatiile efectuate de tranzactii

DEFINITII

START TRANSACTION

[transaction_characteristic...]

BEGIN [WORK]

COMMIT [WORK] [AND [NO] CHAIN] [[NO] RELEASE]

ROLLBACK [WORK] [AND [NO] CHAIN] [[NO] RELEASE]

SET autocommit = {0 | 1}

END

Pornesc o noua tranzactie

Ruleaza
tranzactia curenta
si face
schimbarile
permanente

Activeaza
autocommit ptr.
Sesiunea curenta

Anuleaza
tranzactia si
modificarile
facute



4. Operatiile efectuate de tranzactii

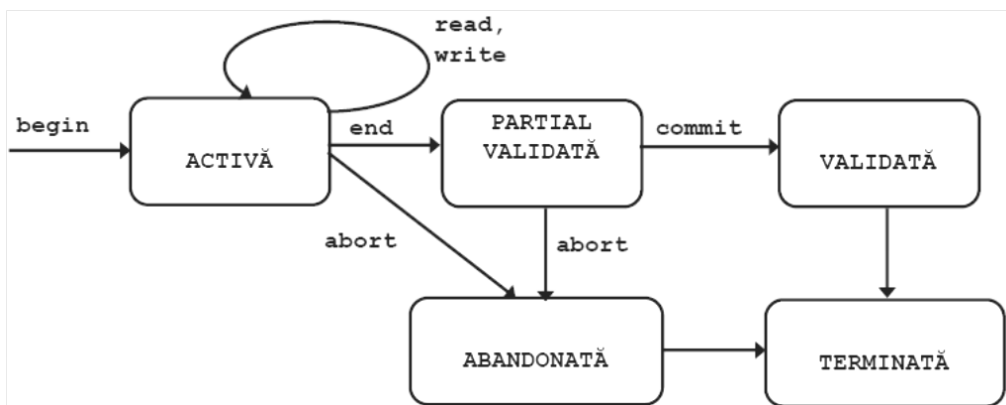
Operațiile efectuate de o tranzacție (înregistrate de administratorul de refacere):

- begin**: începutul execuției unei tranzacții
- read** sau **write**: operații de citire sau scriere a articolelor din BD
- end**: marchează terminarea operațiilor de scriere/citire din BD; uneori sunt necesare operații de verificare înainte de validarea (commit) tranzacției.
- commit**: terminarea cu succes a tranzacției, validarea tuturor modificărilor efectuate în BD și vizibilitatea modificărilor efectuate pentru alte tranzacții;
- rollback (abort)**: tranzacția a fost abandonată și orice efect pe care tranzacția l-a avut asupra BD trebuie să fie anulat (printr-o “rulare înapoi” a operațiilor).
- undo**: operație similară operației rollback, dar se aplică unei singure operații, nu unei întregi tranzacții.
- redo**: specifică faptul că unele operații ale unei tranzacții trebuie să fie executate din nou pentru a se putea valida întreaga tranzacție.



5. Starile tranzactiilor

DIAGRAMA DE STARE A UNEI TRANZACTII



EXEMPLE TRANZACTII IN MYSQL

```

//Ex. Baza de date test cu tabela useri
mysql> use test;
Database changed
mysql> Select * from useri;
Empty set (0.00 sec)
mysql> Start transaction;
Query OK, 0 rows affected (0.00 sec)
mysql> Insert into useri(user_name) values ('utilizator1');
Query OK, 1 row affected (0.00 sec)
mysql> Insert into useri(user_name) values ('utilizator2');
Query OK, 1 row affected (0.00 sec)
mysql> Insert into useri(user_name) values ('utilizator3');
Query OK, 1 row affected (0.00 sec)
mysql> rollback;
Query OK, 0 rows affected (0.01 sec)
mysql> Select * from useri;
  
```

Care este rezultatul tranzactiei?

user_id	user_name
1	utilizator1
2	utilizator2
3	utilizator3

Gresit !

```

// Empty set (0.00 sec)
//tabela useri nu va contine inregist-
//trile deoarece s-a renuntat la
//acestea, prin rollback.
  
```



EXEMPLE TRANZACTII IN MYSQL

```
//Ex. Baza de date test cu tabela useri
mysql> Start transaction;
Query OK, 0 rows affected (0.00 sec)
mysql> Insert into useri(user_name) values ('utilizator4');
Query OK, 1 row affected (0.00 sec)
mysql> Insert into useri(user_name) values ('utilizator5');
Query OK, 1 row affected (0.00 sec)
mysql> commit;
Query OK, 0 rows affected (0.03 sec)
mysql> Select * from useri;
```

Care este rezultatul tranzactiei?

```
2 rows in set (0.00 sec)
+-----+-----+
| user_id | user_name |
+-----+-----+
|      4 | utilizator4 |
|      5 | utilizator5 |
+-----+-----+
```



REFACEREA UNEI BD

Fișier jurnal (log file): în care SGBD memorează operațiile efectuate de fiecare tranzacție, identificată printr-un identificator unic generat de sistem

Mod de utilizare:

- fișierul jurnal este memorat pe disc și nu este afectat de erori de execuție, cu excepția unei defectări catastrofice a discului,
- este salvat periodic pe un suport auxiliar (bandă magnetică).



6. Planificarea tranzacțiilor

Planificare (schedule) sau **istorie** (history)= ordonarea tranzacțiilor ;

Operatii conflictuale (conflicting operations) : 2 operații dintr-o planificare sunt conflictuale dacă aparțin unor tranzacții diferite, accesează același articol și cel puțin una dintre operații este operație de scriere

Planificări seriale (serial schedules): o planificare este serială dacă pentru orice tranzacție participantă în planificare, toate operațiile se execută consecutiv ; altfel, **planificarea** se numește **neserială**

Planificări serializabile: se utilizează în cazul SBD cu utilizatori multipli, care admit concurența, asigurând în același timp consistența BD; O planificare a n tranzacții se numește serializabilă, dacă este echivalentă cu o planificare serială a celor n tranzacții



7. Tehnici de control al concurenței

Controlul concurenței se realizează prin tehnici bazate pe **blocarea accesului la date prin**:

- zăvoare** (locks)
- mărci de timp** (timestamps)

Protocoalele de control al concurenței sunt implementate de SGBD-uri, a.i. programatorii de aplicații :

- nu operează explicit cu zăvoare sau mărci de timp
- stabilesc opțiunile prin care sistemul SGBD adoptă tehnici de control



BLOCAREA ACCESULUI LA DATE PRIN ZĂVOARE

Zăvor (lock): variabilă asociată unui articol al BD care descrie starea acelui articol în raport cu operațiile care i se pot aplica

Tipuri de zăvoare utilizate în SGBD-uri:

- zăvoare binare
- zăvoare cu stări multiple

Zăvor binar (binary lock): poate avea două stări:

- liber (neblocaț = free, unlocked) și ocupat (blocaț = busy, locked) sau
- 1 și 0

Operații asupra unui zăvor binar :

- operația de blocare, lock(X)
- operația de eliberare, unlock(X)



BLOCAREA ACCESULUI LA DATE PRIN ZĂVOARE

Zăvor cu stări multiple (multiple-mode lock): poate fi într-una din următoarele stări:

- liber** (neblocaț, unlocked): zăvorul nu este deținut de nici o tranzacție și prima tranzacție care lansează o operație de blocare îl poate obține
- blocaț pentru citire** (read-locked): oricâte tranzacții pot deține zăvorul respectiv și pot efectua operații de citire a articolului protejat de acesta, dar nici o tranzacție nu poate scrie în acel articol
- blocaț pentru scriere** (write-locked): o singură tranzacție poate deține zăvorul și poate citi sau scrie în articolul protejat de acesta, nici o altă tranzacție neputând accesa articolul respectiv, nici pentru scriere nici pentru citire



CONTROLUL CONCURENTEI PRIN MARCI DE TIMP

Marcă de timp (timestamp): identificator unic al unei tranzacții, creat de SGBD bazat pe timpul de start al tranzacției

Moduri de creare a unei marci de timp, utilizand:

- valoarea curentă a ceasului sistemului de operare
- numărător incrementat la fiecare asignare a unei noi mărci, în ordinea de lansare a tranzacțiilor

Serializabilitatea planificărilor: prin impunerea unor condiții ordinii de accesare a articolelor de mai multe tranzacții concurente, în funcție de mărcile de timp ale acestora



8. Tehnici de refacere a BD

Refacerea unei BD după producerea unui defect (database recovery): aducerea BD într-o stare precedentă corectă, din care, eventual, se poate reconstrui o nouă stare corectă și cât mai apropiată de momentul apariției defectului

Tehnicile de refacere a BD: sunt în general integrate cu tehnicile de control al concurenței și depind de SGBD. Pentru operațiile de refacere se folosește fișierul jurnal (log file), și (sau) o copie de rezervă a BD (database backup) stocată în general pe bandă magnetică sau alte dispozitive externe.



8. Tehnici de refacere a BD

Elemente ale fișierului jurnal :

- Punct de validare** (commit point): punctul atins de o tranzacție care a executat cu succes toate operațiile și le-a înregistrat în fișierul jurnal
- Punct de control** (checkpoint): este înscris în fișierul jurnal atunci când se scriu în fișierele BD toate rezultatele operațiilor de scriere ale tranzacțiilor validate

Administratorul de refacere al SGBD (recovery manager): decide la ce interval de timp (sau după câte tranzacții) introduce un nou punct de control



REFACEREA DATELOR DUPA DEFECTE NECATASTROFICE

Defecte necatastrofice: BD nu este distrusă fizic, dar a devenit inconsistentă datorită unui defect.

Strategie de refacere dupa defect necatastrofic:

- anularea modificărilor** care au produs inconsistența (prin operații **undo**):
- executarea din nou a modificărilor care s-au pierdut** (prin operații **redo**)

Nu este necesară copia de rezervă, se folosește starea actuală a BD și fișierul jurnal

Tehnici de refacere a datelor dupa defecte necatastrofice:

- refacerea cu actualizare amânată** (deferred update)
- refacerea cu actualizare imediată** (immediate update)



REFACEREA DATELOR DUPA DEFECTE NECATASTROFICE

Refacerea cu actualizare amanata

- se parcurge fișierul jurnal în sens invers, începând de la ultima înregistrare, până se întâlnește primul punct de control
- se construiește o listă a tranzacțiilor validate și a tranzacțiilor nevalidate, din acest punct de control
- se execută reluarea (REDO) tuturor operațiilor de scriere ale tranzacțiilor validate, în ordinea în care apar în fișierul jurnal, iar tranzacțiile nevalidate sunt relansate



REFACEREA DATELOR DUPA DEFECTE NECATASTROFICE

Refacerea cu actualizare imediată

- atunci când o tranzacție lansează o comandă de actualizare a BD, actualizarea este efectuată imediat, fără să se mai aștepte ajungerea la un punct de validare
- în majoritatea acestor tehnici se impune regula ca modificarea să fie mai întâi memorată în fișierul jurnal (pe disc), înainte de a fi aplicată BD (“protocol de scriere în avans a fișierului jurnal”(write-ahead log protocol))
- dacă se admite actualizarea imediată, atunci la refacere trebuie să se poată anula (prin undo) modificările efectuate de o tranzacție, dacă aceasta eșuează ulterior și astfel se efectuează rularea înapoi (rollback) a tranzacției



REFACEREA DATELOR DUPA DEFECTE CATASTROFICE

Defecte catastrofice: BD a fost puternic distrusă datorită unei defectări serioase a SBD (ex.: hard disc deteriorat).

Strategie de refacere după defect catastrofic :

- se restaurează starea BD din copia de rezervă a BD (database backup)
- tranzacțiile efectuate de la ultima operație de salvare până în momentul apariției defectului sunt considerate pierdute
- se folosește fișierul jurnal pentru a reface toate tranzacțiile validate existente în copia salvată a fișierului jurnal