

Baze de date

- Lucrare de laborator -

TRANZACTII

Obiective

1. Definirea unei tranzactii
2. Folosirea declaratiilor, formularilor, instructiunilor de genul BEGIN TRAN (inceperii unei activitati), COMMIT WORK (realizarii unei activitati) si ROLLBACK (revenirii asupra unei activitati, reluarea ei) in cadrul unei tranzactii

Definirea tranzactiei

Gestiunea tranzactiilor, se refera la capacitatea unui SGBD relational de a efectua tranzactii intr-o baza de date.

O tranzactie este o multime de instructiuni care se executa in grup. Daca oricare din instructiune nu poate fi incheiata, toate celelalte actiuni trebuiesc refacute. Tranzactiile sunt unitati de prelucrare(program) care trebuie efectuate fie in intregime, fie deloc. Prin unitate de prelucrare intelegem ca o tranzactie are un inceput si un sfarsit.

Prin urmare o tranzactie are intotdeauna un inceput si un sfarsit. Server-ul bazei de date garanteaza ca operatiile realizate in limitele unei tranzactii vor fi fie complet si perfect salvate pe disc, sau daca acest lucru nu este posibil, baza de date va reveni in stadiul in care era inainte ca tranzactia sa fi inceput.

Daca ceva gresit se petrece intimpul unei tranzactii, intreaga unitate de prelucrare poate fi anulata, dupa dorinta utilizatorului. Daca totul este corect, unitatea de prelucrare poate fi salvata in intregime in baza de date

Tranzactie bancara

Transferul banilor din contul de economii in contul curent este un bun exemplu de tranzactie.

Actiuniile implicate in aceasta tranzactie sunt:

- Actualizarea contului de economii prin indicarea unei retrageri.
- Actualizarea contului curent pentru a pune in evidenta un depozit.

Aceste actiuni trebuiesc amandoua incheiate cu succes inainte ca transferul sa fie terminat. Daca ceva nu merge, cu oricare dintre cele doua actiuni, tranzactia esueaza. De exemplu daca ati incercat sa transferati, in contul curent, mai multi bani decat aveti in contul de economii transferul nu va avea loc si balanta conturilor dumneavoastra va ramane la fel cum a fost inainte ca dumneavoastra sa incercati sa realizati transferul.

Logarea unei tranzactii

Multe lucruri, ca de exemplu o defectiune software sau hardware, pot duce la esecul unei tranzactii. Din aceasta cauza server-ul bazei de date trebuie sa aiba o modalitate de urmarire a modificarilor realizate in timpul tranzactiei. Acest lucru este posibil prin intermediul logarii unei tranzactii. Server-ul foloseste logarea unei tranzactii pentru a inregistra fiecare modificare facuta bazei de date in timpul tranzactiei. Daca tranzactia nu poate fi realizata, server-ul foloseste automat datele din jurnalul(logarea) tranzactiei pentru a anula modificarile.

Bazele de date nu sunt prevazute cu un sistem de jurnalizare automata a tranzactiilor. Administratorul bazei de date trebuie sa decida daca realizeaza sau nu o baza de date prevazuta cu un sistem de logare a tranzactiilor. Daca baza de date nu este prevazuta cu un sistem de logare tranzactiile nu sunt valabile.

Reluarea activitatii (ROLLBACK)

In unele cazuri poate veti dori ca programul dvs. sa anuleze o tranzactie, in mod deliberat, in functie de unele circumstante. Pentru aceasta se foloseste instructiunea ROLLBACK (reluarea sau revenirea la un anumit punct in cadrul activitatii). Instructiunea ROLLBACK face ca server-ul bazei de date sa anuleze tranzactia curenta si sa anuleze si modificarile facute. In exemplul dat in pseudo-cod declaratia ROLLBACK este executata doar daca @v este setat pe **zero** inainte ca tranzactia sa fie realizata.

BEGIN TRANSACTION

```
Select @v=1
update economii set sold=sold-100
If sold < 0          (este balanta negativa?)
  Then
    @v=0
  else
    update cont-curent set sold=sold+100
end
If @v=0 then
  ROLLBACK    -- una sau mai multe actiuni au esuat
else
  COMMIT     -- toate actiunile s-au incheiat cu succes
end
```

Exercitii propuse pentru laborator

Completati exercitiul descris mai jos. Intrebarile(interogari) ar trebui executate separat pentru a va permite observarea comportamentului fiecarei interogari.

1. Incepeti o tranzactie.
2. Inserati un nou client numit *Albu Ion* in tabelul **clienti** .
3. Selectati toti clientii numiti *Albu* din tabelul **clienti** .
Intrebare: A fost inregistrat clientul *Fred Flinstone* in tabelul **clienti**?
4. Reluati tranzactia.
5. Selectati toti clientii numiti *Fred* din tabelul **clienti** . In ce mod difera SELECT-ul primei interogari fata de SELECT-ul celei de-a doua interogari.

Nu uitati sa includeti un place holder pentru numarul coloanei clientului in instructiunile INSERT. De asemenea nu uitati sa separati instructiunile prin intrmediul ';

Solutie

```
BEGIN TRANSACTION
INSERT INTO clienti (customer_nu, fname, lname)
VALUES (0,'Fred','Flinstone');
SELECT * FROM clienti WHERE fname='Fred'
ROLLBACK WORK
SELECT * FROM clienti WHERE fname='Fred'
```

Primul SELECT returneaza doi clienti inclusiv noul client inserat. Al doilea SELECT returneaza doar un singur client deoarece instructiunea INSERT a fost reluata.

TRANZACTII (SQL-SERVER)

Inceperea unei tranzactii

```
BEGIN TRAN [ SACTION] [ nume_tranzactie | @tran_nume_variabila  
[ WITH MARK [ 'description' ] ] ]
```

nume_tranzactie Este numele asignat tranzactiei care trebuie sa se conformeze regulilor pentru identificatori care nu trebuie sa aiba mai mult de 32 de caractere.

@tran_nume_variabila

Este numele unei variabile definite de utilizator ce contine numele unei tranzactii valide. Variabila trebuie sa fie declarata de tipul **char**, **varchar**, **nchar**, sau **nvarchar**.

WITH MARK ['description']

Specifica faptul ca tranzactia este marcata in fisierul de log. *description* este un sir de caractere care descrie marcarea.

Daca WITH MARK este folosit, numele tranzactiei trebuie sa fie specificat. WITH MARK permite pentru restaurarea fisierului de log de la numele marcat.

Terminarea unei tranzactii

COMMIT [TRAN [SACTION] [*nume_tranzactie* | @tran_nume_variabila]

Comanda COMMIT salveaza toate modificarile efectuate in cursul unei tranzactii. Deseori este mai bine sa executati o instructiune COMMIT inainte de a incepe o tranzactie noua

Anularea unei tranzactii

**ROLLBACK [TRAN [SACTION]
[*nume_tranzactie* | @tran_nume_variabila
nume-punct-de salvare | @variabila-punct-de salvare]]**

nume-punct-de salvare

Este numele punctului de salvare pentru un enunt SAVE TRANSACTION.

Folositi *nume-punct-de salvare* cand o anulare a tranzactiei trebuie sa afecteze doar o parte din tranzactie.

@variabila-punct-de salvare

Este numele unei variabile definite de utilizator ce contine numele punct de salvare valid. Variabile trebuie sa fie declarata de tipul **char**, **varchar**, **nchar**, sau **nvarchar**.

Dupa inceperea unei tranzactii, este efectuata de obicei o procedura de verificare a erorilor pentru a determina daca tranzactia a fost executata cu succes pana in momentul respectiv. Instructiunea ROLLBACK readuce tranzactia la forma initiala. Aceasta inseamna ca starea bazei de date este refacuta in forma care exista inainte de inceperea tranzactiei.

Folosirea punctelor de salvare in tranzactii

SAVE TRAN [SACTION] { *nume-punct-de salvare* | @variabila-punct-de salvare }

Am mentionat ca operatia de readucere a unei tranzactii la punctul de inceput inseamna de fapt anularea intregii tranzactii. Cu toate acestea, in unele cazuri, ati dori sa executati doar o parte din tranzactie adica o parte din instructiunile din cadrul tranzactiei. Pentru a realiza acest lucru, trebuie sa folositi un punct de salvare. Dincolo de acest punct, daca apare o comanda de revenire, tranzactia este adusa la starea din punctul de salvare. Toate instructiunile care au fost executate pana in acest punct de salvare sunt salvate.

```

-----
BEGIN TRANSACTION
    INSERT INTO catalog VALUES('222','3',8,'17-06-2005')
    COMMIT TRANSACTION
GO

SELECT * FROM Catalog

-----
BEGIN TRANSACTION tran1
    INSERT INTO catalog VALUES('222','3',10,'17-06-2005')
    COMMIT TRANSACTION tran1
GO

SELECT * FROM Catalog

-----
DECLARE @TranName VARCHAR(20)
SELECT @TranName = 'TransactiaMea'

BEGIN TRANSACTION @TranName
GO
USE dbStud
GO
UPDATE catalog
SET Nota = Nota +1
WHERE NrLeg LIKE '22%'
GO

COMMIT TRANSACTION TransactiaMea
GO

SELECT * FROM Catalog

-----
BEGIN TRANSACTION TranModif
    WITH MARK 'Modificarea Valorilor'
GO
USE dbStud
GO
UPDATE Catalog
    SET Nota = Nota +2
    WHERE NrLeg LIKE '11%'
GO
COMMIT TRANSACTION TranModif
GO

```

```

SELECT * FROM Catalog

-----
BEGIN TRANSACTION AddStudNou
INSERT INTO Student
VALUES('210','Traian','I','Titi','M','10-09-
1980','N','10303', NULL)
IF EXISTS (SELECT * FROM Student WHERE NrLeg='210')
    BEGIN
        BEGIN TRANSACTION AddNota
        INSERT INTO Catalog VALUES('210', '2',5,'4-9-2001')
        INSERT INTO Catalog VALUES('210', '4',7,'12-9-2001')
    END
ELSE
    ROLLBACK TRANSACTION
COMMIT TRANSACTION

GO
SELECT * FROM Student
GO
SELECT * FROM Catalog

-----
BEGIN TRANSACTION Tran1
    UPDATE Student
        SET NrLeg_sot = '104'
        WHERE NrLeg = '210'

    UPDATE Student
        SET NrLeg_sot = '210'
        WHERE NrLeg = '104'
SAVE TRANSACTION savepoint1

UPDATE Catalog
    SET nota = Nota +1
    WHERE NrLeg='210'
SELECT s.NrLeg, Nume, Prenume, Nota, Cod_materie, NrLeg_sot
FROM Student s, Catalog n
WHERE s.NrLeg=n.NrLeg

ROLLBACK TRANSACTION savepoint1
COMMIT TRANSACTION

```