

Tranzacții

O tranzacție este o unitate logică de lucru a SQL Server constând dintr-o secvență de comenzi care trebuie să se execute ca un întreg pentru a menține consistența bazei de date. Tranzacțiile ne asigură că o operație de modificare a datelor fie este îndeplinită complet, fie nu este efectuată deloc.

Pentru a controla tranzacțiile definite de utilizator, se utilizează comenzile BEGIN TRANSACTION (început tranzacție), COMMIT TRANSACTION (tranzacție efectuată cu succes) și ROLLBACK TRANSACTION (refacerea contextului anterior începerii tranzacției).

Identificatorul *Transaction* poate fi prescurtat cu *Tran*

Exemplul 1

```
select codStd,nume,prenume, from tStudenti
begin tran
  delete tStudenti where localitate!='Pitesti'
  select codStd,nume,prenume, localitate from tStudenti
rollback tran
select codStd,nume,prenume, localitate from tStudenti
```

codStd	nume	prenume	localitate
s1	Marinescu	Bogdan	Pitesti
s2	Radu	Ionel	Pitesti
s3	Pop	Andreea	Buc
s4	Radu	Bogdan	NULL
s5	Popescu	Ion	NULL
s6	Ionescu	Ion	Pitesti
s7	Puiu	Victor	NULL

(7 row(s) affected)

(4 row(s) affected)

codStd	nume	prenume	localitate
s1	Marinescu	Bogdan	Pitesti
s2	Radu	Ionel	Pitesti
s6	Ionescu	Ion	Pitesti

(3 row(s) affected)

codStd	nume	prenume	localitate
s1	Marinescu	Bogdan	Pitesti
s2	Radu	Ionel	Pitesti
s3	Pop	Andreea	Buc
s4	Radu	Bogdan	NULL
s5	Popescu	Ion	NULL
s6	Ionescu	Ion	Pitesti
s7	Puiu	Victor	NULL

(7 row(s) affected)

Exemplul 2

```
begin tran tr1 - putem sa atribuim un identificator tranzactiei
select codStd, codCurs,nota from tRezEx
  update tRezEx set nota=nota+1 where codCurs='c1'
if (select max(nota) from tRezEx where codCurs ='c1') >9
  rollback tran tr1
else
  commit tran tr1
select codStd, codCurs,nota from tRezEx
```

Limbajul Transact-SQL permite imbricarea tranzacțiilor prin generarea de comenzi Begin Transaction imbricate. Variabila globală @@TRANCOUNT ne furnizează nivelul de imbricare al tranzacțiilor – 0 indică absența imbricării, 1 indică imbricare cu un nivel, 2 imbricare cu 2 nivele etc

Exemplu de tranzacții imbricate:

```
select * from tClienti
```

codClient	denumire	codJudet	localitate	restAdresa
C1	S.C. Vulturul SRL	Ag	Pitesti	NULL
C2	SC Ana SRL	Ag	Pitesti	NULL
C3	S.C. Viitorul	DB	Targoviste	NULL
C4	S.C. Soft Nou SRL	VL	Rm. Valcea	NULL
C5	S.C. X SRL	VL	Rm.Valcea	NULL
C6	cocolino	ag	pitesti	NULL

```
BEGIN TRAN Tranzactia1
  UPDATE tClienti
    SET denumire = 'S.C. Viitorul SRL'
    WHERE codClient = 'c3'
  BEGIN TRAN Tranzactia2
    UPDATE tClienti
      SET denumire = 'S.C. ALFA SRL'
      WHERE codClient = 'c5'
    COMMIT TRAN Tranzactia2
  ROLLBACK TRAN Tranzactia1
```

```
select * from tClienti
```

codClient	denumire	codJudet	localitate	restAdresa
C1	S.C. Vulturul SRL	Ag	Pitesti	NULL
C2	SC Ana SRL	Ag	Pitesti	NULL
C3	S.C. Viitorul	DB	Targoviste	NULL
C4	S.C. Soft Nou SRL	VL	Rm. Valcea	NULL
C5	S.C. X SRL	VL	Rm.Valcea	NULL
C6	cocolino	ag	pitesti	NULL

In acest exemplu, observăm că, în ciuda comenzii imbricate *COMMIT TRAN Tranzactia2*, comanda *ROLLBACK TRAN Tranzactia1* exterioară anulează și efectele comenzii UPDATE corespunzătoare tranzacției interioare *Tranzactia2*.

```
BEGIN TRANSACTION
BEGIN TRY

    select * from tNote
    update tNote set nota=Nota+1
    select * from tNote
    declare @NrNoteMari int
    set @NrNoteMari=(select count(case when nota>7 then 1 end) from tNote)
    print @NrNoteMari
    if @NrNoteMari>8      RAISERROR('Prea multe note mari',12,2)

    COMMIT TRAN
    select * from tNote

END TRY

BEGIN CATCH
    print 'Eroare '+str(@@ERROR)
    print ERROR_MESSAGE()
    ROLLBACK TRAN
    select * from tNote
END CATCH
```