

12.ACID, Seriabilitate

Pentru ca o tranzactie sa fie bine definita ea trebuie sa indeplineasca niste criterii de corectitudine care au fost sintetizate prin abrevierea ACID. Aceasta semnifica

- A – Atomicitate
- C – Consistenta
- I – Izolare
- D – Durabilitate.

Atomicitate

O tranzactie trebuie sa fie atomica in sensul ca fie toate modificarile efectuate de ea in baza de date sunt comise fie sunt toate revocate.

```
INSERT INTO STUD ...  
UPDATE SPEC ...  
DELETE FROM STUD ...
```

Aceste modificari trebuie ori comise ori revocate impreuna. Faptul ca de exemplu doar inserarea si stergerea sunt comise si nu si actualizarea este o incalcare a acestei reguli. Sistemul de gestiune este cel care trebuie sa puna la dispozitie mecanismele prin care sa se asigura atomicitatea tranzactiilor inclusiv in cazul unor incidente hardware si software care pot interveni in timpul executiei unei tranzactii.

Consistenta

O tranzactie care incepe sa lucreze pe o baza de date consistenta trebuie sa o lase la final tot intr-o stare consistenta.

In acest sens o tranzactie nu poate incalca restrictiile existente la nivelul bazei de date. In cele mai multe cazuri aceste restrictii sunt modelate sub forma constrangerilor de integritate (NOT NULL, PRIMARY KEY, etc).

Daca o tranzactie contine o operatie care violeaza o constrangere de integritate atunci toate modificarile efectuate de tranzactie vor fi revocate. Mecanismele de pastrare a consistentei trebuie asigurate de SGBD.

Izolare

O tranzactie trebuie sa se comporte ca si cand operatiile efectuate de ea sunt izolate, independente de operatiile efectuate de alta tranzactie.

Nici o alta tranzactie nu trebuie sa citeasca date intermediare scrise de tranzactia respectiva. De exemplu daca o tranzactie contine succesiunea de operatii:

```
READ (A)    -- citeste valoarea veche a lui A: 5  
A = A + 1  
WRITE (A)   -- scrie valoarea noua a lui A:6
```

```

READ (B)  -- citeste valoarea veche a lui B:10
B = B - 1
WRITE (B) -- scrie valoarea noua a lui B:9

```

Atunci nici o alta tranzactie nu poate citi pentru A si B doua valori dintre care una este actualizata si cealalta nu (adica 6 pentru A si 10 pentru B).

Inconsistentele prezentate in paragraful anterior erau datorate incalcarii acestui criteriu de corectitudine. Din punct de vedere al modului de asigurare a izolarii tranzactiilor tot sistemul de gestiune trebuie sa asigure mecanismele necesare.

Izolarea este obiectul controlului accesului concurrent, prezentat in paragrafele urmatoare.

Durabilitate

O data comise cu succes modificarile efectuate de catre o tranzactie ele vor persista si nu mai pot fi revocate. Inclusiv in cazul unui incident hardware si software efectele tranzactiilor comise sunt regasite la recuperarea dupa incident. Din acest punct de vedere fiecare sistem de gestiune trebuie sa contina mecanisme prin care efectele tuturor tranzactiilor comise sa fie inregistrate si in jurnalele sistemului pentru a fi restaurate in caz de incident.

6.3. Serializabilitate

Asa cum s-a specificat planificarile seriale nu duc la inconsistente. In practica insa in cazul unor sisteme incarcate planificarile contin pasi intercalati ai diverselor tranzactii. Rezultatul va fi totusi corect daca efectul executiei planificarii respective este acelasi cu al uneia dintre planificarile seriale posibile ale acelorasi tranzactii. O astfel de planificare se numeste planificare serializabila.

6.3.1. Planificari serializabile

Definitie: O planificare este serializabila daca produce aceleasi efecte in baza de date cu o planificare seriala.

Exemplu: O planificare serializabila si planificarea seriala echivalenta:

Planificare serializabila:

T1	T2	T3
READ A		
READ B		
		READ C
		READ D
WRITE A		
		WRITE C
	READ A	
	WRITE A	

Planificare seriala echivalenta (se putea si T1, T2, T3):

T1	T2	T3
		READ C
		READ D
		WRITE C
READ A		
READ B		
WRITE A		
	READ A	
	WRITE A	

Exemplul 2: Planificare neserializabila si cele doua planificari seriale echivalente: Sa luam exemplul anterior cu doua tranzactii care rezerva si anuleaza o rezervare:

T1	T2	A în BD
READ A		10
	READ A	10
A = A + 1		10
	A = A + 1	10
WRITE A		11
	WRITE A	11

Planificare seriala 1:

Planificare seriala 2:

T1	T2	A în BD
READ A		10
A = A + 1		10
WRITE A		11
	READ A	11
	A = A + 1	11
	WRITE A	12

T1	T2	A în BD
	READ A	10
	A = A + 1	10
	WRITE A	11
READ A		11
A = A - 1		11
WRITE A		12

Se observa ca exista doar doua planificari seriale diferite pentru T1 si T2 si nici una nu produce acelasi rezultat cu planificarea initiala.

6.3.2. Planificari conflict-serializabile

Exista si o alta abordare a serializabilitatii bazata pe conflictele care pot sa apara intre pasii a doua tranzactii dintr-o planificare.

Definitie: Intre doua operatii apartinand unei planificari exista un conflict daca:

- Apartin unor tranzactii diferite
- Sunt pe acelasi obiect
- Una dintre operatii este o scriere
- Cele doua operatii sunt succesive in sensul ca intre ele nu exista o operatie cu care vreuna dintre ele este in conflict.

Rezulta ca exista 3 tipuri de situatii conflictuale: Fiind date doua tranzactii T1 si T2 pot exista conflicte de tipurile R1-W2, W1-W2 si W1-R2.

Definitie: Doua planificari sunt conflict-echivalente daca:

- Contin aceleasi operatii ale acelorasi tranzactii
- Fiecare pereche de operatii conflictuale apare in aceeași ordine in cele doua planificari.

Aceasta definitie nu spune ca nu pot sa apara anomalii in executia celor doua planificari ci ca apar aceleasi anomalii in ambele.

Definitie: O planificare este conflict-serializabila daca este conflict-echivalenta cu o planificare serial.

Observatie: Reformuland putem spune ca o planificare este conflict-serializabila daca poate fi transformata intr-o planificare seriala prin interschimbari ale operatiilor consecutive care nu sunt in conflict din doua tranzactii.

Exemplu:

1. T1	T2
Read A	
Write A	
	Read A
Read B	
	Write A
Write B	
	Read B
	Write B

3. T1	T2
Read A	
Write A	
Read B	
	Read A
Write B	
	Write A
	Read B
	Write B

2. T1	T2
Read A	
Write A	
Read B	
	Read A
	Write A
Write B	
	Read B
	Write B

4. T1	T2
Read A	
Write A	
Read B	
Write B	
	Read A
	Write A
	Read B
	Write B

Prin trei interschimbari de operatii neconflictuale s-a obtinut o planificare seriala.

Test de conflict-serializabilitate:

- Se construiește graful de dependenta astfel:
 - Nodurile sunt tranzactii
 - Pentru orice pereche de operatii aflate in conflict O_i si O_j , cu O_i in T_i si O_j in T_j , avem un arc de la nodul T_i la T_j daca O_i apare in planificare inaintea lui O_j .
- Daca acest graf nu contine cicluri planificarea este conflict-serializabila altfel nu este conflict-serializabila

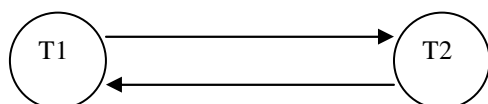
Exemplu:

T1	T2
READ A	
	WRITE A
WRITE A	

Avem 2 conflicte:

- T1-READ A cu T2-WRITE A
- T2-WRITE A cu T1-WRITE A

Rezulta ca graful are doua noduri si doua arce:



Cum exista un ciclu planificarea nu este conflict-serializabila.

Observatie: exista planificari serializabile care nu sunt conflict-serializabile. De exemplu sa presupunem ca in planificare de mai sus tranzactia T2 scrie in A exact valoarea citita de T1. Planificarea nu e conflict-serializabila dar e serializabila, avand acelasi efect cu planificarea seriala "T2 urmata de T1":

T1	T2
	WRITE A
READ A	
WRITE A	

Acest fapt se datoreaza inasa doar coincidentei intre valoarea scrisa de T2 si cea citita de T1. Cum sistemul de gestiune nu face astfel de judecati pentru el potential planificarea este periculoasa putand sa duca la inconsistente. De aceea in judecarea planificarilor se considera ca la o scriere o tranzactie poate scrie orice valoare si nu doar o valoare particulara.

6.3.2. Planificari v-serializabile (view-serializability)

Exista de asemenea o a treia abordare (mai slaba) a serializabilitatii:

Definitie: Doua planificari S1 si S2 sunt v-echivalente daca pentru orice articol A:

- Daca T_i citeste valoarea initiala a lui A in S1 atunci ea face acelasi lucru si in S2
- Daca T_i citeste o valoare a lui A scrisa de T_j in S1, atunci face acelasi lucru si in S2.
- Daca T_i scrie valoarea finala a lui A in S1 atunci ea face acelasi lucru si in S2

Definitie: O Planificare este v-serializabila daca este v-echivalenta cu o planificare seriala.

Exemplu: O planificare v-serializabila si planificarea seriala v-echivalenta:

T1	T2	T3
READ A		
	WRITE A	
WRITE A		
		WRITE A

T1	T2	T3
READ A		
WRITE A		
	WRITE A	
		WRITE A

Aceasta definitie permite planificarile de tranzactii conflict-serializabile si planificari care contin tranzactii care scriu date fara sa citeasca ceva din baza de date. Din acest punct de vedere planificarea de la punctul 6.3.2. nu e conflict-serializabila dar este v-serializabila.

Incluziunea intre diverse tipuri de planificari este urmatoarea:

