

Tema: Anomalii de execuție concurentă a tranzacțiilor.

Blocarea relațiilor și gestiunea tranzacțiilor.

Interacțiunea necontrolată a două sau mai multe tranzacții poate duce la apariția unor stări inconsistente ale BD și la producerea unor rezultate eronate. Două tranzacții T_i și T_j sunt susceptibile de interferență (*capabile să se suprapună*) dacă rezultatul execuției lor concurente poate fi diferit de rezultatul execuției seriale.

Între două tranzacții poate să apară o interferență dacă acestea efectuează operații asupra unor date comune. Dacă aceste două tranzacții sunt executate în mod concurent, atunci spunem că sunt conflictuale.

Deci, două tranzacții T_i și T_j sunt conflictuale dacă sunt concurente și susceptibile de interferență.

În funcție de natura operațiilor pe care le efectuează asupra datelor comune, între două tranzacții pot să apară mai multe tipuri de interferențe care provoacă anomalii de interferență:

anomalia de actualizare pierdută (lost update)

anomalia de citire improprie (dirty read)

anomalia de citire irepetabilă (nonrepeatable read)

Anomalia de actualizare pierdută

Correspunde unui conflict de tip scriere-scriere și constă în faptul că rezultatul actualizării efectuate de o tranzacție se pierde ca urmare a reactualizării aceleiași date de către o altă tranzacție, fără ca reactualizarea să fie influențată de rezultatul primei actualizări.

Exemplu: Fie următoarea execuție concurentă a două tranzacții T_1 și T_2 .

Observație: în acest capitol vom nota simbolic operațiile de citire și scriere a unui atribut X din BD prin Read X , respectiv Write X .

T1	T2
	Read A
Read A	
$A=A+5$	
Write A	
	$A=A+10$
	Write A

În urma acestor tranzacții valoarea lui A apare mărită cu 10, nu cu 15 așa cum ar fi de așteptat. Este ca și cum tranzacția T_1 nici nu s-ar fi executat.

Anomalia de citire improprie

Corespunde unui conflict de tip scriere-citire și apare atunci când o tranzacție surprinde o stare temporar inconsistentă a BD.

Exemplu: Fie următoarea execuție concurentă a două tranzacții T1 și T2.

T1	T2
Read A	
$A=A-10$	
Write A	
	Read A
	Read B
	$C=A+B$
	Write C
Read B	
$B=B+10$	
Write B	

În urma acestor tranzacții valoarea sumei $A + B$ este neschimbată. Intenția era de a reține în C valoarea acestei sume, dar datorită interferenței, valoarea din C este cu 10 mai mică decât cea reală.

Anomalia de citire irepetabilă

Corespunde unui conflict de tip citire-scriere și apare atunci când aceeași tranzacție găsește valori diferite la citiri repetate ale aceluiași date.

Exemplu: Fie următoarea execuție concurentă a două tranzacții T1 și T2.

T1	T2
ReadA	
$B=A$	
Write B	
	Read A
	$A=A+10$
	Write A
ReadA	

C=A	
Write C	

Se observă că, deși valorile rezultate pentru B și C ar trebui să fie egale în urma execuției tranzacției T1, ele sunt diferite din cauza interferenței cu tranzacția T2.

Blocarea relațiilor și gestiunea tranzacțiilor

Dacă drepturile de *inserție*, *suprimare* sau *modificare* ale unei relații le posedă mai mulți utilizatori sau dacă un utilizator inițiază mai multe sesiuni de lucru asupra BD, pot apărea operații simultane asupra unuia și aceluiași articol de date. Aceste modificări simultane pot duce la apariția unor fenomene nedorite, la Inconsistența datelor. **SGBD-urile** Avansate au posibilitatea de a realiza operații asupra datelor în mod concurent, cu un grad clar de paralelism. Dacă comparațiile sunt de tipul **SELECT**, adică nu se modifică conținutul relației, ele se pot realiza fără probleme.

Drept normă, se consideră starea în care o operație de inserare, modificare sau suprimare a tuplurilor blochează relația, împiedicând accesul altor utilizatori asupra ei până operația nu se termină. Blocările pot fi predefinite, fixate automat de sistem sau definiție explicit, declarate de utilizator.

Blocarea poate fi realizată la nivel de tupluri sau la nivel de relații în două moduri: *exclusiv*(**EXCLUSIVE**) sau *partajat* (**SHARE**).

Folosind instrucțiunea **LOCK TABLE**, un utilizator poate bloca o relație(apelând la ea pe nume sau sinonim). Blocarea se elimină, folosind instrucțiunea **UNLOCK TABLE**.

LOCK TABLE<nume relatie>**IN**<mod> **MODE**;
UNLOCK TABLE <nume relatie>;

În limbajul SQL, este posibilă, în mod explicit, alegerea nivelului de protecție contra incoerenței ce poate rezulta în urma accesului concurențial la date. Comportamentul predefinit este cel de asigurare a serializabilitatii și restabilirii stricte, Dar acest mod poate provoca o încetinire a fluxului tranzacțional pentru aplicațiile care nu au voie de un control atât de strict.

Cea mai simplă opțiune este cea de specificare a tranzacției că este numai de citire. În aceste condiții, se poate garanta că ea nu va provoca nici o problemă de concurență și SGBD-ul poate economisi efortul de fixare a blocajelor.

Instrucțiunea SQL este:

SET TRANSACTION READ ONLY;

Astfel, sunt interzise comenzile *INSERT*, *UPDATE*, *DELETE* până la imediat următorul *COMMIT* sau *ROLLBACK*. Sistemul va respinge aceste comenzi. Dar există și o consecință neplăcută: două lecturi succesive ale aceleași instrucțiuni *SELECT* pot genera rezultate diferite, deoarece între cele două lecturi datele pot fi actualizate de alte tranzacții.

Opțiunea predefinită este cea de citire și scriere. această opțiune poate fi specificată, în mod explicit, de instrucțiunea:

SET TRANSACTION READ WRITE;