

Управление на Транзакции

Транзакции

- ▶ Транзакция = серия от четене и писане на обекти от базата данни.

T2

R(A)

W(A)

R(B)

W(B)

Commit



ACID свойствата на транзакциите

- ▶ Атомарност
- ▶ Всяка транзакция в базата данни се разглежда като единица работа.
- ▶ Така за да бъде успешна една транзакция, всяка операция съдържаща се в нея също трябва да бъде успешна.
- ▶ Ако една от операциите не е успешна цялата транзакция се смята за неуспешна и всички промени направени от транзакцията до момента се премахват(Abort/Rollback).
- ▶ Ако пък всички операции са извършени успешно, транзакцията се смята за успешна и преминава в базата(Commit).

► Консистентност

- Всяка транзакция в базата данни трябва да довежда базата данни от едно консистентно състояние в друго консистентно състояние.

Т.е. Ако имаме база данни със заплатите на служители, техните заплати не могат да бъдат отрицателни. Следователно ако опитаме да зададем на някой служител отрицателна заплата това би нарушило консистентността на базата. Подобни транзакции не бива да бъдат допускани.

- ▶ Изолация
- ▶ Всяка транзакция в базата трябва да работи така, все едно е единствената транзакция която се извършва в момента.

Това означава, че тя не бива да бъде повлиявана от други транзакции. Колкото и транзакции да се извършват едновременно, резултатът на всяка една от тях не трябва да се повлиява от останалите.

► Устойчивост

- Цялата информация която се съдържа в базата трябва да се запазва на постоянна памет.(хард диск)
- В случай на срыв на системата, базата трябва да е в състояние да се възстанови напълно.

Поредност на Транзакциите (schedule)

- ▶ Всяка транзакция от гледна точка на базата е серия от писане и четене на информация от базата. Тези операции се следят в списък. Този списък определя реда на изпълнение на всяка транзакция и нейните операции.
- ▶ За да се постигне по-доброто представяне на базата, транзакциите не се обработват една след друга, а успоредно.

<i>T1</i>	<i>T2</i>
<i>R(A)</i>	
<i>W(A)</i>	
	<i>R(B)</i>
	<i>W(B)</i>
<i>R(C)</i>	
<i>W(C)</i>	

Конфликти

- ▶ Read-Write conflict (unrepeatable read) - четене на промени от незавършени транзакции
- ▶ Write-Read conflict (dirty read) - промяна на информация вече прочетена от транзакция, която още се извършва
- ▶ Write-Write conflict - презаписване на промени

<i>T1</i>	<i>T2</i>
<i>R(A)</i> <i>W(A)</i>	<i>R(A)</i> <i>W(A)</i> <i>R(B)</i> <i>W(B)</i> Commit
<i>R(B)</i> <i>W(B)</i> Commit	

Успоредна работа

- ▶ Протокол за заключване
- ▶ 2 Phase Locking

За да се избегнат подобни конфликти се използват протоколи за заключване на определени ресурси от базата.

Shared lock(read)

Exclusive lock(write)

$T1$	$T2$
$S(A)$	
$R(A)$	
	$S(A)$
	$R(A)$
	$X(B)$
	$R(B)$
	$W(B)$
	Commit
$X(C)$	
$R(C)$	
$W(C)$	
Commit	

Възстановяване след срыв

- ▶ Всяка операция която се извършва върху базата се записва хронологично в списък(Log).
- ▶ Системата се грижи този списък да уцелее дори при срыв. След което, използвайки този списък започва да се възстановява състоянието на базата.

