

# Трансакције

Ненад Митић

Математички факултет  
nenad.mitic@matf.bg.ac.rs

# Трансакције

Како се извршавају операције у РСУБП?

- пример могуће операције: корисник улази у банку и жели да подигне новац са једног рачуна и уплати га на други
- Како обезбедити да нпр. у случају нестанка струје после подизања новца, а пре уплате не дође до "нестанка" новца корисника?

Све операције у РСУБП се обављају у облику **трансакција**

# Транзакције-примери

```
BEGIN TRANSACTION ;

UPDATE ACC 123 { BALANCE := BALANCE - $100 } ;
IF any error occurred THEN GO TO UNDO ; END IF ;

UPDATE ACC 456 { BALANCE := BALANCE + $100 } ;
IF any error occurred THEN GO TO UNDO ; END IF ;

COMMIT ; /* successful termination */
GO TO FINISH ;
UNDO :
ROLLBACK ; /* unsuccessful termination */
FINISH :
RETURN
```

# Увод

## Структура трансакције је

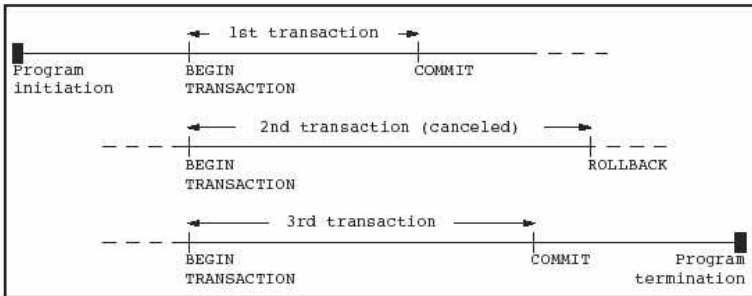
```
BEGIN TRANSACTION;  
/* niz operacija aplikacije */  
.  
.  
COMMIT; (ili ROLLBACK;)  
/* signal uspesnog ili neuspesnog zavrsetka */
```

## Кључне речи

- BEGIN TRANSACTION – при инцијализацији програма
- COMMIT – при нормалном завршетку
- ROLLBACK – при прекиду програма
- могу бити и имплцитно позване

## Увод

Програм се извршава као низ трансакција  
COMMIT/ROLLBACK завршавају трансакцију али  
не прекидају извршавање програма



## Увод

- Трансакција је логичка јединица посла
- Може да садржи низ операција
- Са гледишта корисника, извршавање трансакције је атомско
  - наредбе се извршавају над скуповима торки и, у случају грешке у средини наредбе, није дозвољено да над неким торкама наредба буде извршена а над неким не
  - систем гарантује да је извршавање наредби над базом атомско

# ACID особине

## Трансакција поседује особине

- 1 Атомичност (енг. *Atomicity*). Трансакције су атомске
- 2 Конзистентност (енг. *Consistency*). Трансакције чувају конзистентност базе
- 3 Изолованост (енг. *Isolation*). Трансакције су изоловане једна од друге приликом извршавања.
- 4 Трајност (енг. *Durability*). По потврђивању трансакције промене остају у бази, чак и у случају пада система

# Особине транзакција

- Транзакција оставља базу у конзистентном стању
  - унутар транзакције, при извршавању оперција које чине транзакцију конзистентност може да буде нарушена
  - конзистентно стање не значи да је садржај базе коректан; данашњи СУБП не врше проверу коректности већ подразумевају да су транзакције коректне
- Могуће угнеждење транзакција једних у друге; није подржано од већине РСУБП



# Тачке чувања (*savepoints*)

- места унутар трансакције на које се долази поништавањем урађених активности које следе после њих
- видљиве су једино унутар трансакције
- нису исто што и COMMIT; извршавањем ROLLBACK-а се поништавају све тачке чувања унутар трансакције

# Особине трансакција - наставак

- Управљач трансакцијама
  - (eng. *transaction manager, transaction processing monitor, TP monitor*)
  - управља трансакцијама, обезбеђује да се не изгубе, да не буду делимично извршене или да не буду извршене више пута
- Паралелно извршава операције трансакције
  - укида потребу за COMMIT, ROLLBACK, BEGIN TRANSACTION
  - није у потпуности подржано од стране савремених СУБП

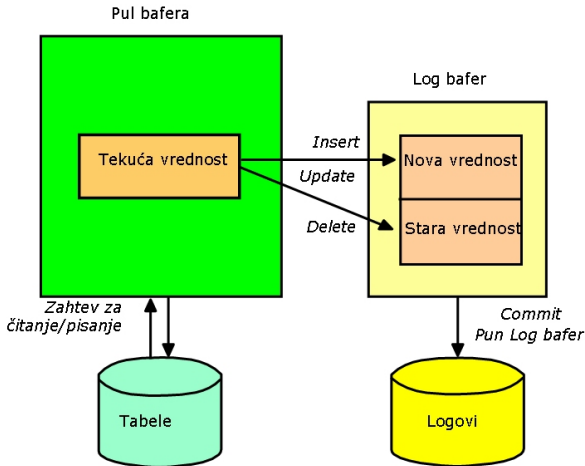
# Увод

Опоравак трансакција подразумева активности које предузима управљач трансакцијама ради повратка базе у конзистентно стање у случају да је нека трансакција поништена, или је њено извршавање прекинуто неким спољашњим утицајем (нестанак струје, пад система, грешка на медијуму, ...)

Принцип на коме је опоравак заснован је постојање редувантних података на физичком нивоу које одржава РСУБП

## Uvod

Редундантни подаци се чувају у **log** датотекама



# COMMIT и ROLLBACK

## COMMIT

- 1 сигнализира успешан крај трансакције
- 2 све промене постају сталне
- 3 до тада су промене сматране само као намера која је могла да се поништи у случају појаве грешке

## ROLLBACK

- 1 сигнализира неуспешан крај трансакције
- 2 база може да буде у неконзистентном стању
- 3 све промене учињене у трансакцији морају да буду поништене (на основу садржаја лог датотеке)

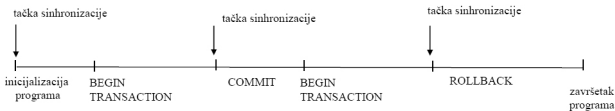
# Лог протокол *писање унапред*

- 1 Упис промена у базу и упис промена у лог су две различите операције
- 2 Између њих може да се деси грешка
- 3 Пре писања слога у физичку базу мора прво да се упише слог у лог датотеку
- 4 Пре потврде трансакције сви лог слогови морају да се упишу у физички лог



# Тачка синхронизације

- 1 Налази се између две трансакције. База је у конзистентном стању
- 2 У њу се долази или иницијализацијом програма или извршавањем COMMIT/ROLLBACK
- 3 Када се успостави тачка синхронизације





# Тачка синхронизације

- све промене настале од претходне тачке синхронизације се потврђују (COMMIT) или поништавају (ROLLBACK)
- сви курсори отворени у програму се затварају, сем уколико нису декларисани са WITH HOLD опцијом
- кључеви над објектима базе се ослобађају (зависи и од начина везивања програма)

# Опоравак

Критичан ресурс: садржај бафера односно главне меморије који је изгубљен

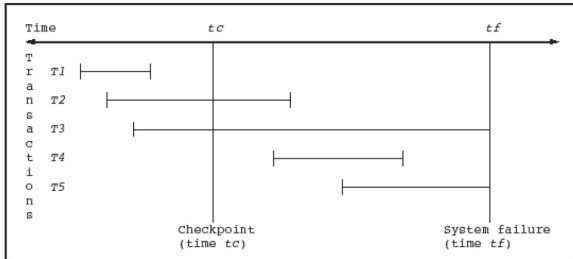
## Опоравак

- поништити ефекте трансакција које су биле активне у време настанка грешке
- поновити извршавање трансакција које су успешно завршиле до тог тренутка али нису стигле да ефекат извршавања запишу на физички диск

# Опоравак - наставак

- Како систем зна које трансакције треба поновити?
- Узимањем пресека стања система (енг. *checkpoint*) у регуларним временским интервалима. Ово узимање укључује:
  - записивање садржаја бафера ван физичког простора који припада бази података
  - записивањем посебног контролног слога (енг. *checkpoint record*) у физичку датотеку која садржи лог. Контролни слог садржи листу трансакција које су биле активне у тренутку узимања пресека стања

# Опоравак - наставак



- У случају на слици трансакције T2 и T4 морају да буду поновљене а ефекат извршавања трансакција T3 и T5 поништен. На трансакцију T1 није утицао пад система
- У општем случају систем одлучује које ће трансакције поново извршити а чије дејство ће поништити

# Алгоритми оправка

## Поступак код старијих реализација РСУБП-а

- Формирају се две листе трансакција, Поновљене и Поништене
- У листу поништених иницијално се уписују све трансакције које су постојале у последњем узетом контролном слогу
- Претражује се лог датотека, почевши од последњег контролног слога
- Ако се нађе `BEGIN TRANSACTION` за трансакцију `T`, тада се она додаје у листу поништених.

# Алгоритми оправка

## Поступак .... (наставак)

- Ако је COMMIT пронађен за трансакцију T, тада се она помера из листе поништених у листу поновљених
- Доласком до краја лог датотеке листе поновљених и поништених садрже трансакције које треба поновити односно чије ефекте треба поништити.
- Систем се враћа унатраг кроз лог поништавајући ефекат извршења трансакција из које се налазе у листи поништених
- Систем се креће ка крају лог датотеке понављајући садржај трансакција из листе поновљених

# ARIES алгоритам

## ARIES (*Algorithms for Recovery and Isolation Exploiting Semantics*)

Због ефикасности савремени системи прво врше понављање извршавања трансакција а затим поништавање ефеката на следећи начин:

- 1) **Анализа:** направе се листе Поновљени и Поништени
- 2) **Понављање:** крене се од позиције у лог датотеци која је одређена у фази анализе и рестаурира се база тако да њено стање одговара стању у тренутку пада система

# ARIES алгоритам - наставак

- 3) Поништавање: поништи се ефекат трансакција које нису потврђене

Како ARIES уписује у лог све промене настале у фази поништавања ефеката трансакција то омогућује коректан рестарт базе у случају да се при поништавању деси нова грешка



# Двофазни COMMIT

- Неопходан у случају да апликација ради са независним управљачима ресурса (нпр. IMS и DB2)
- Непоходан у случају дистрибуираних база
- Обезбеђује коректно извршавање у случају појаве грешке на било којој компоненти
- "Глобални" (односи се на све системе) COMMIT или ROLLBACK
- Координатор - системска компонента која координира рад управљача ресурса

# Двофазни COMMIT - наставак

- Припрема:
  - координатор шаље свим управљачима ресурса да се припреме, тј. да упишу у своје физичке лог датотеке све лог слоге које се односе на трансакцију и да пошаљу поруку о успешности те операције
- Потврда:
  - када координатор прими (потврдне) поруке од свих управљача ресурса он уписује слоге у свој физички лог. Ако су све операције успешне онда је трансакција потврђена, а ако нису онда је поништена.

# Двофазни COMMIT - наставак

- Потврда (наставак)
  - Сваки од учесника у трансакцији мора да потврди или поништи трансакцију у складу са поруком координатора

У случају грешке система прво се гледа лог датотека координатора и на основу њеног садржаја се шаљу поруке учесницима у процесу

# SQL подршка

- START TRANSACTION - код већине савремених РСУБП имплицитно
- COMMIT [WORK] [AND [NO] CHAIN];
- ROLLBACK [WORK] [TO SAVEPOINT [ime]];
  - AND CHAIN узрокује почетак нове трансакције
  - AND NO CHAIN је предефинисано
- SAVEPOINT ....
- WITH HOLD у декларацији курсора