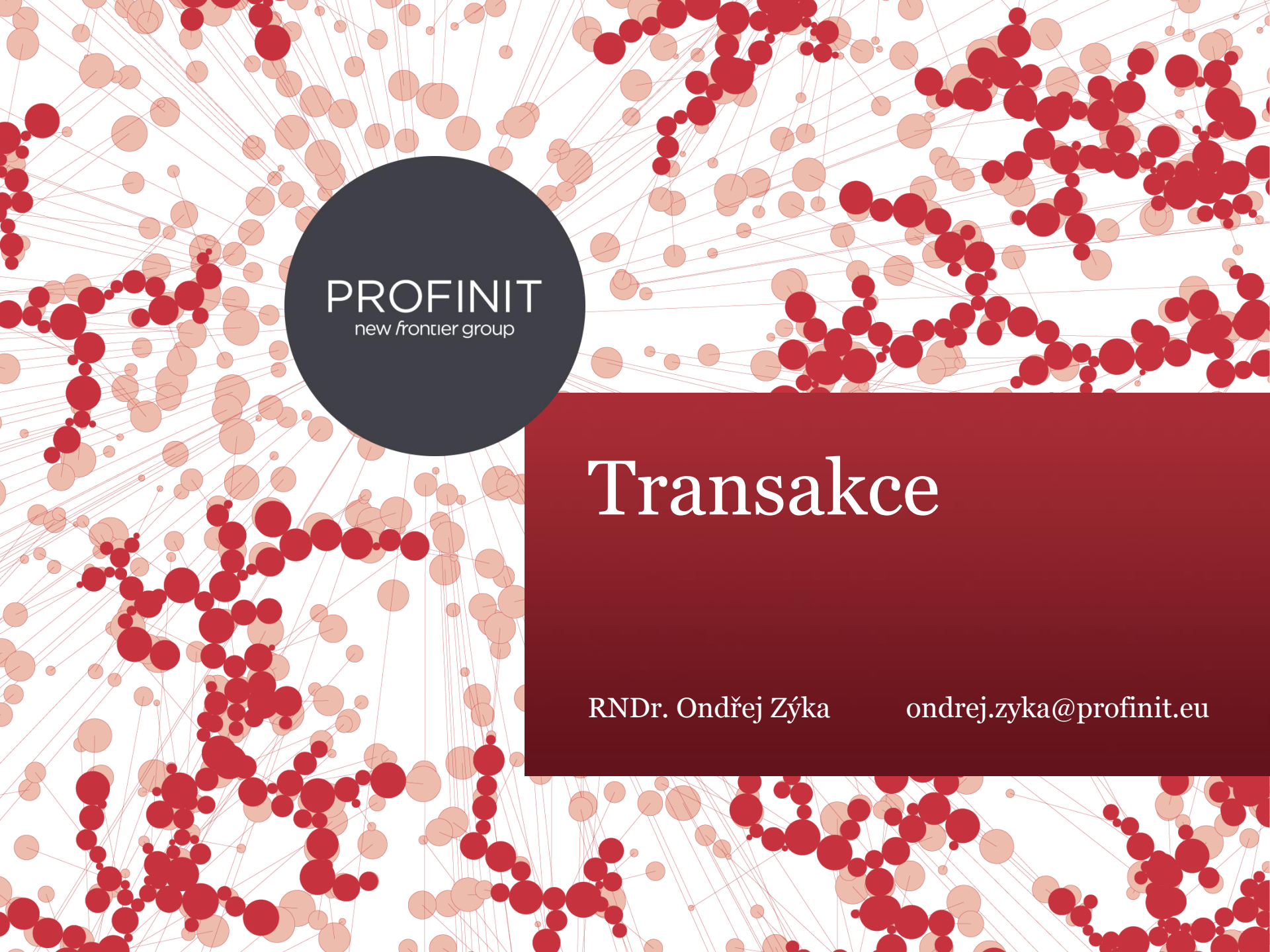


Transakce

RNDr. Ondřej Zýka ondrej.zyka@profinit.eu

Obsah

Transakce

- Množina operací s daty, které splňují podmínku ACID
- Atomicity
- Consistency
- Isolation
- Durability

Transakce

PROFINIT
new frontier group

- Atomicity
  - Změna musí být provedena celá (nebo vůbec)
  - I v případě chyby hardware, chyby software, chyby aplikace, chyby operačního systému.
  - Uživatel musí být informován, zda se transakce uskutečnila a je ukončena.

- Consistency – po konci transakce musí být všechny požadavky na konzistenci databáze splněny
  - null values
  - foreign key
  - unique constraint
  - deferred…

Transakce

PROFINIT
new frontier group

- Isolation - Neukončené změny nejsou viditelné pro ostatní uživatele.
  - Uživatel provádějící změnu vidí i vlastní nekomitované změny.

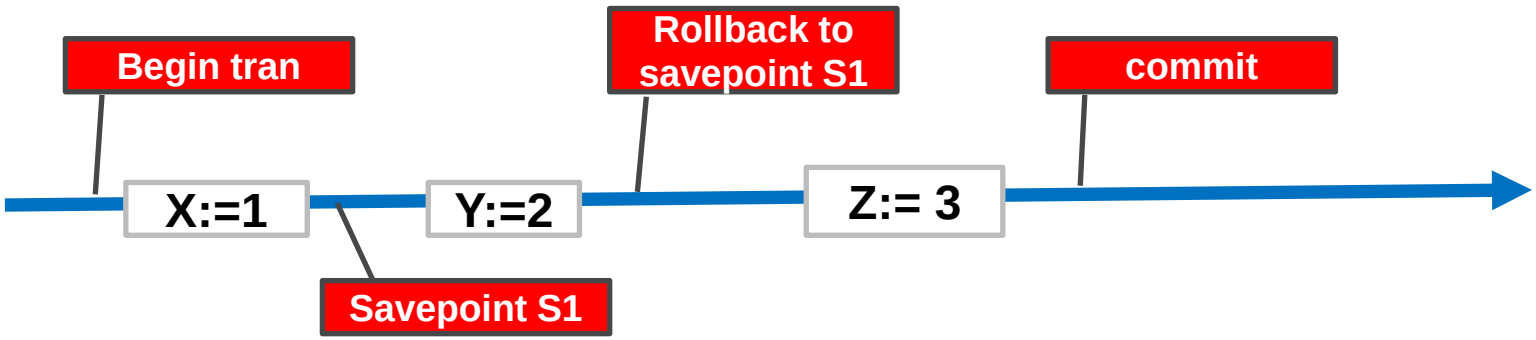
- Durability – Commitovaná data jsou trvale uložena v databázi.
  - Commitovaná znamená, že uživatel dostal informaci o ukončení commitu. (Příkaz commit byl ukončen a server předal řízení uživateli).
  - Transakace přežije jakoukoliv systémovou chybu.

Transakce v jiných významech

- Aplikační transakce
  - Vytvoření objednávky (desítky databázových transakcí),
  - Přepočet ohodnocení skladu (desítky minut).
- Transakce na podnikové úrovni
  - Schválení půjčky – několik aplikačních transakcí, workflow zasahující několik oddělení.
- Long-running transaction
  - transakční model podporující
    - persistenci transakcí při restartu systémů,
    - interakci s uživateli,
    - opravné bloky kódu spouštěné při neúspěchu operací.

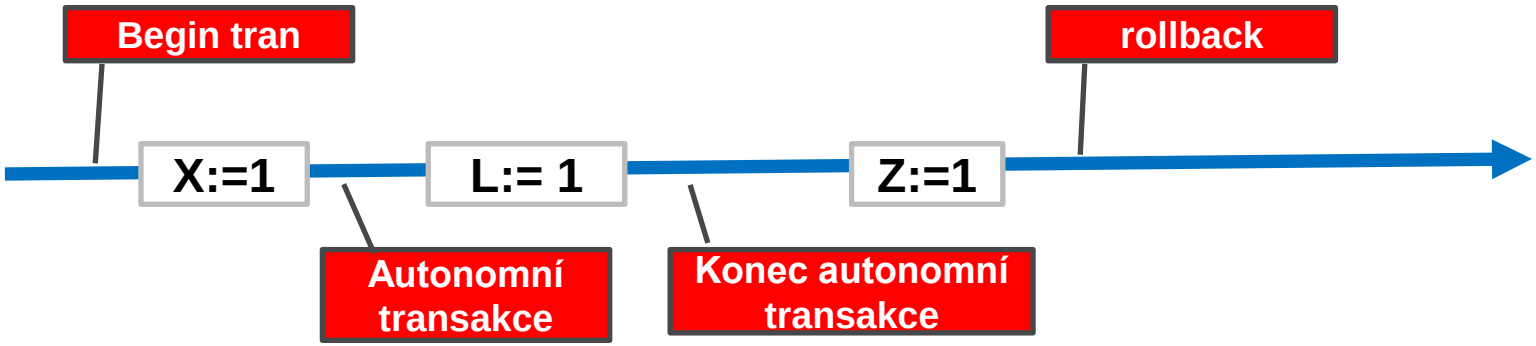
Savepoint

- Každé napojení do databáze – maximálně jedna transakce.
- Savepoint, rollback to savepoint
- V transakci se dají zrušit poslední provedené změny
- **Nejdou** zrušit pouze změny ze začátku transakce



Autonomní transakce

- Změna v datech mimo transakci
- Například zápis do logu
- Oracle
- Na úrovní procedury



Konzistentní stav databáze

PROFINIT
new frontier group

- Dva logické přístupy
- Pokud A dělá změnu, data jsou nekonzistentní a B musí počkat na konzistentní stav.
  - Implementace pomocí zamykání.
- I když A dělá změnu, existuje konzistentní stav - poslední konzistentní stav před změnou A.
  - Multiversion concurency control (snapshots)

Transakční mód

- Chained
  - Začátek transakce
    - První příkaz (ne každý)
    - Insert, update, delete, select for update/holdlock, …
  - Konec transakce
    - Commit, rollback
- Unchained
  - Každý příkaz autonomní transakce
  - Začátek transakce
    - Explicitně begin tran
  - Konec transakce
    - Commit, rollback

Transakční mód

- Oracle
  - Chained mode
- MS SQL server, Sybase
  - Unchained i chained mód
- Simulace (v klientských nástrojích)
  - Chained módu
    - Commit begin tran, rollback begin tran
    - Není ekvivalentní chained módu
  - Unchained módu
    - Commit za každým příkazem

Porušení izolace transakcí

PROFINIT
new frontier group

- **Dirty write**
  - Tranakce T1 změní data, která jsou měněna v probíhající transakci T2.
- **Dirty read**
  - Transakce T1 načte data změněná transakcí T2 ještě před tím, než transakce T2 provedla commit.
- **Fuzzy read (Non-repeatable read)**
  - Během transakce T1 se dvakrát načte řádek a pokaždé se vrátí jiná hodnota (transakce T2 modifikovala řádek mezi dvěma čteními).
- **Phantom**
  - Během transakce T1 se provede dvakrát stejný dotaz a pokaždé vrátí jiný výsledek (transakce T2 přidala nebo ubrala řádek použitý v dotazu transakce T1).

ANSI isolation level

PROFINIT
new frontier group

	Dirty write	Dirty read	Fuzzy read	Phantom
Read uncommited	Not possible	Possible	Possible	Possible
Read commited	Not possible	Not possible	Possible	Possible
Repeatable read	Not possible	Not possible	Not possible	Possible
Serializable	Not possible	Not possible	Not possible	Not possible

Implementace pomocí zámků

PROFINIT
new frontier group

- Typy zámků
  - Shared – transkce čte objekt, zámek končí po načtení objeku nebo trvá do konce transakce
  - Exclusive – transakce mění objekt, zámek trvá do konce transakce
  - Update – objekt zřejmě bude změněn (cursor)
  - Zámek na data, zámek na indexy
  - Doba uvolnění zámků
- Rozsah zámků
  - Řádek
  - Stránka
  - Tabulka
  - Databáze
- Intend zámek
  - Intend table shared – na tabulce existuje shared zámek na nějakém řádků nebo stránce

Kombinace zámků

- Microsoft® SQL Server 2012

Požadované zámky

Existující zámky

	IS	S	U	IX	SIX	X	Sch-S	SCH-M	BU
Intent shared	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
Shared	Yes	Yes	Yes	No	No	No	Yes	No	No
Update	Yes	Yes	No	No	No	No	Yes	No	No
Intent exclusive	Yes	No	No	Yes	No	No	Yes	No	No
Shared with intent exclusive	Yes	No	No	No	No	No	Yes	No	No
Exclusive	No	No	No	No	No	No	Yes	No	No
Schema stability	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Schema modify	No	No	No	No	No	No	No	No	No
Bulk update	No	No	No	No	No	No	Yes	No	Yes

Implementace pomocí zámků

PROFINIT
new frontier group

- Problematické vlastnosti
  - Pokud transakce dlouho a často exklusivně zamykají řádek, nedá se řádek číst.
  - Pokud mnoho transakcí dlouhodobě zamyká řádek pro čtení, nedá se upravit.
  - Uzamknutí stránky nebo tabulky omezí přístup i na data, která transakce nepoužívá.
  - Režie s množstvím zámků
    - Zámky vyžadují zdroje datového serveru
    - Eskalace zamykání
  - Absolutní nutnost používat krátké transakce

Speciální konstrukce

PROFINIT
new frontier group

- Nečekat na uvolnění zámku nebo čekat maximálně určenou dobu

```
select * from author for update nowait;
select * from author for update wait 10;
```

- Číst jenom z neuzamčených oblastí

```
select * from author readpast;
```

- Manuálně uzamknout tabulku

```
lock table table_name
in {share | exclusive } mode
[ wait [ numsecs ] | nowait ]
```

Optimistické schéma zamykání

PROFINIT
new frontier group

- Ke konfliktům dochází zřídka
- Načtení hodnot
- Zpracování hodnot
- Při zápisu kontrola, že použité hodnoty jsou správné
- Pokud ne, proveď opravu
  - Znovu načti data a proveď opakuj výpočet
  - Informuj uživatele

Optimistické schéma zamykání

PROFINIT
new frontier group

```
select @old_column=column from table where condition
(condition vybere jeden řádek)

Použití:
   @old_column
Natavení:
   @new_column

update table
   set column = @new_column
   where column = @old_column
   and condition

Update jednoho řádku – hodnota se nezměnila -> commit
Update žádného řádku – hodnota se změnila -> oprava
```

Pesimistické schéma zamykání

PROFINIT
new frontier group

- Ke konfliktům dochází často
- Načtení hodnot a jejich uzamčení
  - Možné čekání na možnost uzamčení nebo chybový stav
- Zpracování hodnot
- Commit
- Ostatní čekají nebo obdrží chybovou hlášku

Pesimistické schéma zamykání

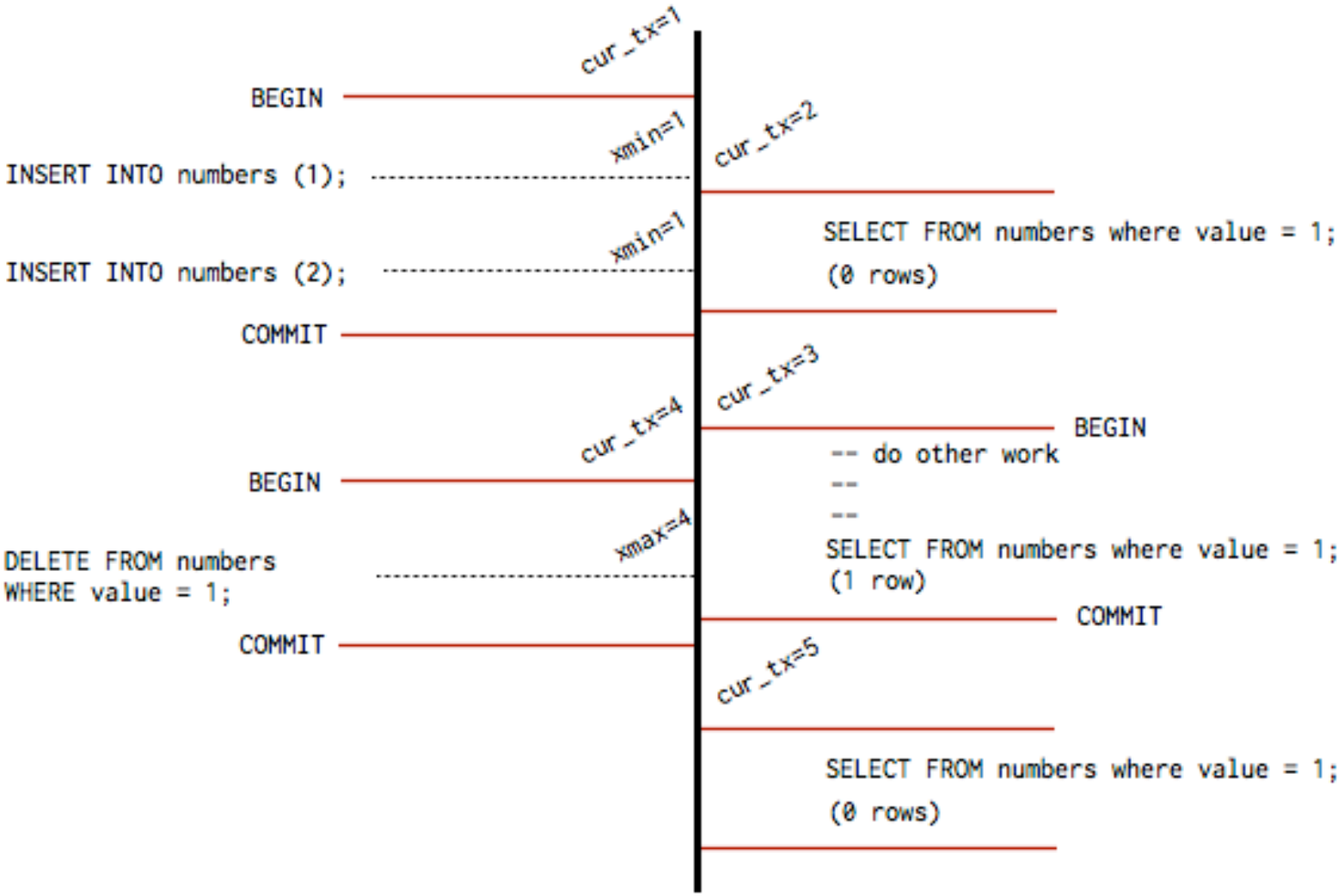
```
select @old_column=column from table where condition
   with holdlock
(řádek je uzamčen)

Použití:
   @old_column
Natavení:
   @new_column

update table
   set column = @new_column
   where condition

commit
```

MVCC (PosgreSQL)

PROFINIT
new frontier group

- Každá transakce má rostoucí timestamp - XID
- Verze na úrovni řádků (obecně několik verzí)
- xmin – timestamp vytvoření
  - timestamp transakce, která řádek vytvořila
- xmax – timestamp ukončení
  - timestamp transakce, která řádek zrušila
  - update zruší starý a vytvoří nový řádek
- Transakce s timestampem x vidí řádky
  - Svoje změny
  - xmin < x, xmax neexistuje a transakce s xmin je komitovaná
  - xmin < x, xmax < x a transakce s xmax je nekomitovaná
- Podmínky pro změny (podle izolačních úrovní) – viz dokumentace.

MVCC



Izolační úrovně

PROFINIT
new frontier group

- Microsoft

```
SET TRANSACTION ISOLATION LEVEL
  { READ UNCOMMITTED
  | READ COMMITTED
  | REPEATABLE READ
  | SNAPSHOT
  | SERIALIZABLE
}
[ ; ]
```

Izolační úrovně

- Oracle
  - Read Committed (Default)
    - konzistence na úrovni příkazu
  - Serializable Transactions
    - konzistence na úrovni transakce
  - Read-only

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

SET TRANSACTION ISOLATION LEVEL READ ONLY;
```

- Oracle
  - Zámky pouze pro měněná data
  - Zámky na úrovni řádků – neexistuje/není potřeba eskalace
  - Transakce libovolně dlouhé – commit zatěžuje server

- Dlouhá Serializable transakce nevidí změny v datech, chyba v serializable módu se objeví až při příkazu commit.

Deadlock

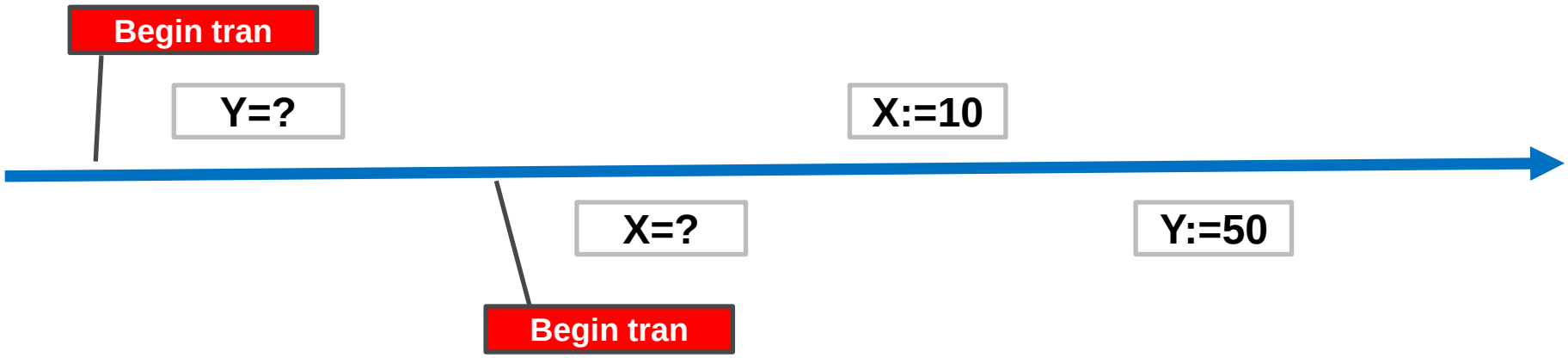
PROFINIT
new frontier group

- Dvě transakce si navzájem blokují zdroje a čekají na jejich uvolnění.
- Situace může nastat v kterémkoliv systému, kde se více uživatelů dělí o zdroje.
- Řešení musí provést nějaká vnější autorita – datový server

Deadlock

Server dokáže deadlock identifikovat

Server zruší jednu z transakcí



Deadlock

PROFINIT
new frontier group

```
update publisher
   set name = 'Aldata
   Infosystems'
   where pub_id = '1389';
```

```
update publisher
   set name = 'New Age Books'
   where pub_id = '0736';
```

```
update publisher
   set name = 'New Age Books'
   where pub_id = '0736';
```

```
update publisher set name =
   'Aldata Infosystems' where
   pub_id = '1389';
```

ORA-00060

Coffmanovy podmínky

PROFINIT
new frontier group

- Dvě transakce si navzájem blokují zdroje a čekají na jejich uvolnění.
- Situace může nastat v kterémkoliv systému, kde se více procesů dělí o prostředky.
- Přesněji: Může nastat v každém systému, kde mohou být splněny tzv. Coffmanovy podmínky:
  - Mutual Exclusion – Prostředek může v jednom okamžiku vlastnit pouze jeden proces.
  - Hold and wait – Proces může žádat o další prostředky, i když má nějaké přiděleny.
  - No preemtion – Pokud proces prostředek vlastní, nelze mu ho bezpečně odejmout (musí ho vrátit sám).
  - Circual wait – Je možné uzavřít cyklus procesů vzájemně čekajících na zdroje svého předchůdce.

Předcházení deadlockům

PROFINIT
new frontier group

- Oslabením některé z Coffmanových podmínek
- Nastavením všech zámků na začátku transakce
- Zamykání tabulek ve stejném pořadí
- Použití krátkých transakcí
- Řešení provede externí autorita – datový server
- Zruší zablokovaný proces
- Přinutí proces aby nějaké zdroje uvolnil

Non serializable

```
set transaction isolation
   level serializable;

update publisher
   set name = 'Aldata
   Infosystems'
   where pub_id = '1389';
```

```
update publisher
   set name = 'XXX' where
   pub_id = '0736';
commit;
```

```
update publisher
   set name = name || '!'
   where pub_id = '0736';

commit;
```

Non serializable

```
update publisher
   set name = 'Aldata
   Infosystems'
   where pub_id = '1389';
```

```
update publisher
   set name = 'XXX' where
   pub_id = '0736';
commit;
```

```
update publisher
   set name = name || '!'
   where pub_id = '0736';

commit;

ORA-08177
```

Závěry

- Složitost problematiky se projeví při zvýšení počtu uživatelů (paralelních transakcí) nikoliv při vývoji.
- Nárůst problémů je exponenciální s počtem paralelních transakcí.
- Nutno počítat s tím, že transakce bude z nějakého důvodu zrušena.
- Je nutné znát přesné možnosti a chování konkrétního serveru (verze)

Co si zapamatovat

PROFINIT
new frontier group

- Co to je transakce
- K čemu slouží savepoint
- Co to je autonomní transakce
- Jaké existují transakční módy a v čem se liší
- Definice isolační úrovně podle ANSI normy
- Jak se implementují isolační úrovně pomocí mechanismu zámků
- Jak se implementují isolační úrovně pomocí snapshotů
- Co to je optimistické a pesimistické schéma zamykání
- Co to je deadlock, jak se dá deadlockům předcházet

PROFINIT
new frontier group

Diskuse