



# Relacyjne Bazy Danych

Andrzej M. Borzyszkowski  
PJATK/ Gdańsk

materiały dostępne elektronicznie  
<http://szuflandia.pjwstk.edu.pl/~amb>

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Poziomy izolacji w/g ANSI/ISO

- Najwyższym poziomem jest założenie, że transakcja jest jedyną wykonywaną w danym momencie, tzn. wiele transakcji musi się uszeregować w kolejności (szeregowalność)
- Może to być zbyt mocne założenie, zbyt ograniczające wydajność bazy danych
- Standard ANSI/ISO wprowadza cztery poziomy izolacji
  - READ UNCOMMITTED
  - READ COMMITTED
  - REPEATABLE READ
  - SERIALIZABLE
- PostgreSQL domyślnie przyjmuje drugi poziom, można ustawić czwarty

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

3

## Poziomy izolacji ANSI/ISO

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

## Poziomy izolacji: odczyt na brudno

- Odczyt na brudno (*dirty read*): odczyt danych jeszcze nie zatwierdzonych przez transakcję piszącą
  - transakcja być może będzie wycofana (ROLLBACK), należy przyjąć, że dane te nigdy nie istniały
- Poziom ANSI/ISO: READ UNCOMMITTED
- PostgreSQL: nie dopuszcza do odczytu na brudno
  - *gdyby dopuszczał*: np. zmiana wartości konta z 100 na 200

czas	transakcja 1	transakcja 2
0 min	SET konto=200	
1 min		SELECT konto... odczyt 200
2 min	ROLLBACK	
3 min		SELECT konto... odczyt 100

Relacyjne Bazy Danych © Andrzej M. Borzyszkowski

4

## Poziomy izolacji: odczyt niepowtarzalny

- Odczyt niepowtarzalny (nonrepeatable read): odczyt danych nie dający się powtórzyć w ramach jednej transakcji
  - tzn. pozwolenie, by inna transakcja zmieniła odczytane dane
- Poziom ANSI/ISO: READ COMMITTED
- PostgreSQL: domyślnie *dopuszcza* do odczytu niepowtarzalnego
  - np. wpłata na konto przez każdego z użytkowników
  - czyli możliwa jest niespójna analiza
  - oraz utracona modyfikacja

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

5

## Odczyt niepowtarzalny, c.d.

- Np. wpłata na konto przez każdego z użytkowników

czas	użytkownik 1	użytkownik 2
0 min	BEGIN WORK	BEGIN WORK
1 min	czyta stan konta, wynik 50	
2 min		czyta stan konta, wynik 50
3 min	pisze wartość konta 110	
4 min		nie może zmienić stanu konta
5 min	COMMIT WORK	
6 min	(gdyby jeszcze raz czytał, byłaby wartość 110)	pisze wartość konta 125
7 min		COMMIT WORK

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

6

## Poziomy izolacji: odczyt widmo

- Odczyt widmo (phantom): odczyt danych nie istniejących wcześniej w danej transakcji
  - tzn. pozwolenie, by inna transakcja wstawiła wiersz do przeczytanej tabeli
- Poziom ANSI/ISO: REPEATABLE READ
- PostgreSQL: domyślnie *dopuszcza* do odczytu widm

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

7

czas	transakcja 1	transakcja 2
0 min	BEGIN	BEGIN
1 min	UPDATE towar SET cena=1	
2 min		INSERT INTO item VALUES ...
3 min	SELECT cena FROM towar	
4 min	COMMIT	

- nowa wartość nie podległa globalnej zmianie w transakcji 1

## Poziomy izolacji

Poziom izolacji	odczyt brudny	niepowt.	widmo
UNCOMMITTED	możliwy	możliwy	możliwy
COMMITTED	niedopuszcz.	możliwy	możliwy
REPEATABLE	niedopuszcz.	niedopuszcz.	możliwy
SERIALIZABLE	niedopuszcz.	niedopuszcz.	niedopusz

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

8

PostgreSQL domyślnie przyjmuje drugi poziom, można ustawić czwarty

**SET TRANSACTION ISOLATION LEVEL SERIALIZABLE**

# Blokady i inne narzędzia zarządzania współbieżnością

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

9

## Blokady binarne

- Obiekt X może być zablokowany
  - transakcja ma prawo blokować tylko obiekt nieblokowany
  - może być utworzona kolejka transakcji do blokowania X
  - transakcja może odblokować obiekt, który zablokowała
  - protokół wzajemnego wykluczania – co najwyżej jedna transakcja blokuje obiekt
- Blokowanie jeszcze nie gwarantuje szeregowalności przebiegu
  - $l1(Y); r1(Y); ul1(Y); l2(X); r2(X); ul2(X); l2(Y); r2(Y); w2(Y); ul2(Y); l1(X); r1(X); w1(X); ul1(X)$
- te blokady niczego nie dają, są za wcześnie zwalniane
  - $r1(Y); r2(X); r2(Y); w2(Y); r1(X); w1(X)$  wymaga  $1 < 2 < 1$
  - blokowanie binarne być może blokuje za dużo

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

11

# Cele i narzędzia zarządzania współbieżnością

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

10

- Szeregowalność
  - dopuszczenie współbieżności jest pożądane/konieczne z powodu wydajności
  - sprawdzanie szeregowalności nie jest słuszne
  - przebieg nie jest znany z góry, wynika ze stanu obliczeń
  - raczej należy gwarantować szeregowalność przebiegu transakcji
- Cel: ten sam wynik zmian w bazie danych
- Narzędzia
  - blokady
  - znaczniki czasu
  - wielowersyjność
  - protokoły optymistyczne

## Blokady czytania i zapisu

- blokada współdzielona (shared) – blokada do odczytu
  - blokada wyłączna (exclusive) – blokada do zapisu
  - Transakcja czytająca dane musi założyć blokadę współdzieloną, wiele transakcji może założyć taką blokadę, nie można jej założyć, jeśli jest już blokada wyłączna
  - Transakcja zapisująca dane musi założyć blokadę wyłączną, nie można jej założyć, jeśli jest już założona jakakolwiek blokada
- | nowa/dotychczasowa | X   | S   | brak |
|--------------------|-----|-----|------|
| X                  | nie | nie | tak  |
| S                  | nie | tak | tak  |
- Różne wersje: jedno zwolnienie blokady, zwolnienie blokady zapisu pozostawiając blokadę odczytu, brak możliwości podnoszenia stopnia blokady

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

12

## Współbieżność vs. blokady

- Niespójna analiza:

czas	użytkownik 1	użytkownik 2
0 min	blokada czytania konta A wynik 100	
1 min		próba blokady zapisu A nieudana, trzeba czekać
2 min	blokada czytania konta B wynik 100	
3 min	zwolnienie blokad	
4 min		blokada zapisu konta A i dalszy ciąg transakcji

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

13

## Współbieżność vs. blokady, c.d.

- Utracona modyfikacja:

czas	użytkownik 1	użytkownik 2
0 min	blokada czytania konta A	
1 min		blokada czytania konta A
2 min	blokada zapisu konta A nieudana, trzeba czekać	
3 min		blokada zapisu konta A nieudana, trzeba czekać

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

- znowu zakleszczenie (*deadlock*)

15

## Współbieżność vs. blokady

- Niespójna analiza w innej kolejności:

czas	użytkownik 1	użytkownik 2
0 min	blokada czytania konta A wynik 100	
1 min		blokada zapisu konta B dodaje 50 do konta B
2 min		blokada zapisu konta A nieudana, trzeba czekać
3 min	blokada czytania konta B nieudana, trzeba czekać	

- te transakcje czekają na siebie nawzajem - zakleszczenie

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

14

## Zakleszczenie, rozwiązanie 1

- Rozwiązanie 1: timeout
  - system zarządzania bazą danych wycofuje transakcję, która zbyt długo oczekiwała na zwolnienie blokady
  - czy transakcja będzie powtórzona?
  - w PostgreSQL nieautomatycznie, może to zrobić aplikacja działająca w pętli aż do pozytywnego zakończenia transakcji
- Wersje:
  - zero czekania – transakcja żądająca niemożliwej blokady jest natychmiast wycofywana
- Problem zagłodzenia
  - nie ma gwarancji, że wycofana transakcja doczeka się wykonania
  - konieczne są metody promocji transakcji wycofywanych

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

16

## Zakleszczenie, rozwiązanie 2,3

- Rozwiązanie 2: analiza grafu oczekiwań
  - SZBD analizuje graf wzajemnych oczekiwań na zwolnienie blokady i wycofuje jedną z transakcji
  - np. najnowszą, lub najstarszą, lub najmniejszą, lub najmniej ważną, lub .....
  - problem zagłodzenia jest obecny, ta sama transakcja nie powinna być ciągle wycofywana
- Rozwiązanie 3: wszystkie blokady powinny być zakładane w tej samej kolejności – wówczas nie będzie zakleszczeń

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

17

## Blokowanie dwufazowe (2PL)

- Protokół blokowanie dwufazowego:
  - faza 1: transakcja zakłada potrzebne blokady (rozszerzanie)
  - faza 2: transakcja zwalnia blokady (kurczenie)
- Twierdzenie: jeśli wszystkie transakcje przestrzegają protokołu blokowania dwufazowego, to dowolny przebieg jest szeregowalny
  - ale zwiększa to niebezpieczeństwo zakleszczenia

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

19

## Zakleszczenie, rozwiązanie 4

- Rozwiązanie 4: znaczniki czasu dla transakcji
  - wersja „czekaj albo zgiń”: transakcja starsza może czekać, młodsza jest wycofywana i powtórnie wykonana z tym samym znacznikiem
  - wersja „zabij albo czekaj”: transakcja starsza powoduje wycofanie młodszej ( i wykonanie z tym samym znacznikiem ), transakcja młodsza czeka
  - na pewno nie będzie zakleszczenia
  - może dość do zagłodzenia
  - wycofywane są transakcje, które być może nie powodują w ogóle zakleszczenia

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

18

## Blokowanie dwufazowe – wersje

- Blokowanie statyczne – transakcja z góry określa swoje blokady
  - jeśli nie może założyć wszystkich, to czeka
  - na pewno nie wystąpi zakleszczenie
  - protokół mało praktyczny, nie zawsze znane są potrzeby
  - transakcja może nigdy nie doczekać się wykonania
- Blokowanie ścisłe – transakcja zwalnia blokady zapisu dopiero na końcu
  - blokowanie rygorystyczne – transakcja zwalnia wszystkie blokady dopiero na końcu
  - gwarantowany jest przebieg ścisły (szeregowalność, łatwe odtwarzanie)
  - możliwość zakleszczeń

© Andrzej M. Borzyszkowski  
Relacyjne Bazy Danych

20

## Blokady a SQL

- SQL nie przewiduje w ogóle jawnego zakładania i zwalniania blokad
  - blokada czytania wierszy będzie założona jeśli użyje się instrukcji **SELECT 1 FROM \_\_\_\_\_ WHERE \_\_\_\_\_** zmuszając system do czytania wierszy
  - blokada zapisu wierszy będzie założona jeśli użyje się instrukcji **SELECT 1 FROM \_\_\_\_\_ WHERE \_\_\_\_\_ FOR UPDATE** anonsując chęć zapisu w wierszach
- PostgreSQL dopuszcza jeszcze instrukcję **LOCK TABLE \_\_\_\_\_** nie należy tej możliwości nadużywać, bo ma poważne konsekwencje dla wydajności
- SQL nie daje możliwości blokowania poszczególnych atrybutów

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

21

## Inne narzędzia rozwiązywania problemów współbieżności

- Znaczniki czasu
  - każda transakcja ma swój znacznik czasu
  - każdy obiekt ma zapisany czas ostatniego odczytu i zapisu przez transakcje jeszcze nie zatwierdzone
  - różne algorytmu wycofujące transakcje, które mogłyby zagrozić pojęciu szeregowalności
- Wielowersyjność
  - SZBD utrzymuje wiele wersji bazy danych dla niezatwierdzonych transakcji, odczyty i zapisy dotyczą odpowiednich wersji
- Techniki optymistyczne
  - transakcje są wykonywane bez przeszkód
  - przy zatwierdzaniu transakcji zaczyna się sprawdzanie, czy mogło dojść do naruszenia spójności bazy danych

© Andrzej M. Borzyszkowski

Relacyjne Bazy Danych

22