

BAZY DANYCH

Uwagi do zadania II

Dodatek do instrukcji laboratoryjnej

Opracowanie:

Agnieszka Landowska
Wojciech Waloszek
Michał Zawadzki

Gdańsk, 2012

WPROWADZENIE

Bazy danych przechowują zazwyczaj bardzo duże ilości danych. W ogromnej większości zastosowań spójność danych ma krytyczne znaczenie, jak np. w zastosowaniach bankowych. Spójność ta powinna być zachowana gdy z systemu korzysta wielu użytkowników jak również, gdy wystąpią np. awarie systemu. Sposobem na zachowanie spójności danych w bazach danych są transakcje.

TRANSAKCJE

Transakcja jest logiczną jednostką składającą się z jednego lub więcej wyrażeń SQL-owych wydawanych przez pojedynczego użytkownika.

Do zilustrowania sposobu funkcjonowania mechanizmu transakcji posłużymy się przykładem. Kiedy klient przelewa pieniądze z rachunku oszczędnościowego na rozliczeniowy, realizuje transakcję składającą się z trzech części: obciążenia rachunku oszczędnościowego, zdeponowania pieniędzy na rachunku rozliczeniowym i zarejestrowania transakcji w dzienniku transakcji:

```
BEGIN TRANSACTION;

UPDATE rachunek_oszczednosciowy
  SET saldo = saldo - 500
  WHERE rachunek = 3209;

UPDATE rachunek_rozliczeniowy
  SET saldo = saldo + 500
  WHERE rachunek = 3208;

INSERT INTO dziennik VALUES
  (dziennik_seq.nextval, '1b',
   3209, 3208, 500);

COMMIT;
```

Baza danych musi zagwarantować wykonanie wszystkich trzech faz transakcji – w przeciwnym razie klient byłby narażony na utratę podjętych pieniędzy. Jeśli któreś z poleceń wchodzących w skład transakcji nie może być wykonane (np. z powodu awarii sprzętowej), wówczas cała transakcja musi ulec odwołaniu. Jeden błąd oznacza anulowanie całej sekwencji poleceń.

Zatwierdzanie i odwoływanie transakcji

Zmiany wprowadzone przez wyrażenia SQL, wchodzące w skład transakcji, mogą być zatwierdzone lub odwołane. Po sfinalizowaniu bieżącej transakcji (przez jej zatwierdzenie lub odwołanie) następne wyrażenie SQL rozpocznie nową transakcję.

Zatwierdzenie transakcji powoduje utrwalenie wszystkich zmian wprowadzonych przez transakcję SQL wchodzące w jej skład. Wszelkie zmiany będą widoczne dla wyrażeń innych użytkowników dopiero po zatwierdzeniu ich macierzystej transakcji.

Odwołanie transakcji oznacza cofnięcie wszystkich wprowadzonych przez nią zmian. Po odwołaniu transakcji dane pozostają niezmiennione, tak jak gdyby transakcja w ogóle nie została rozpoczęta.

ANOMALIE WSPÓLBIEŻNEGO DOSTĘPU

Współbieżny dostęp do danych (wielu użytkowników, wiele transakcji) może powodować zagrożenia spójności danych. Wyróżniamy cztery anomalie współbieżnego dostępu:

Utracona modyfikacja (ang. lost update)

Anomalia występuje gdy dwie transakcje jednocześnie odczytują i modyfikują te same dane – jedna transakcja nadpisuje zmiany dokonane przez drugą:

Transakcja 1	Transakcja 2
...	READ(x)
READ(x)	...
WRITE(x)	...
...	WRITE(x)
COMMIT	...
...	COMMIT

Brudny odczyt (ang. dirty read)

Brudny odczyt występuje gdy pierwsza transakcja odczytuje niezatwierdzone zmiany dokonane przez drugą transakcję. Jeśli druga transakcja zostanie wycofana, dane odczytane przez pierwszą transakcję będą nieprawidłowe, ponieważ wycofanie usunęło wprowadzone zmiany. Pierwsza transakcja nie będzie świadoma, iż odczytane przez nią dane są nieprawidłowe:

Transakcja 1	Transakcja 2
READ(x)	...
WRITE(x)	...
...	READ(x)
ROLLBACK	...
...	WRITE(x)
...	COMMIT

Niepowtarzalny odczyt (ang. non-repeatable read, fuzzy read)

Niepowtarzalny odczyt występuje, gdy wielokrotne odczytanie tych samych danych w ramach tej samej transakcji daje różne rezultaty – dane nie są chronione przed modyfikacją w ramach innej transakcji:

Transakcja 1	Transakcja 2
...	READ(x)
READ(x)	...
...	UPDATE/DELETE(x)
...	COMMIT
READ(x)	...
COMMIT	...

Fantomy (ang. phantom read)

Fantomy występują gdy nowe dane dodane do bazy danych są dostępne w transakcjach, które rozpoczęto przed operacją dodawania. Wyniki zapytania będą zawierały dane dodane przez inne transakcje rozpoczęte już po rozpoczęciu danej transakcji:

Transakcja 1	Transakcja 2
READ(x)	...
...	INSERT(x)
...	COMMIT
READ(x)	...
COMMIT	...

POZIOMY IZOLACJI TRANSAKCJI

Dla zwiększenia współbieżności definiuje się poziomy izolacji transakcji, świadomie godząc się

z występowaniem niektórych anomalii. Anomalia utraconej modyfikacji nigdy nie może się pojawić. Nie występuje ona na żadnym poziomie izolacji. Zgodnie ze standardem ANSI/ISO SQL zdefiniowane są poziomy izolacji:

- **Read Uncommitted** - Transakcja może odczytywać niezatwierdzone dane (dane zmienione przez inną transakcję będącą jeszcze w toku).
- **Read Committed** - Transakcja nie może odczytywać niezatwierdzonych danych zmienionych przez inną transakcję będącą jeszcze w toku.
- **Repeatable Read** - Transakcja nie może zmienić danych, które są czytane przez inną transakcję.
- **Serializable** - Transakcja ma wyłączne prawa do odczytu i zapisu danych – inne transakcje nie mają dostępu do tych samych danych. W praktyce oznacza to, że transakcje korzystające z tych samych danych wykonywane są po sekwencyjnie, a nie współbieżnie.

	Lost update	Dirty read	Fuzzy read	Phantom read
READ UNCOMMITTED	nie występuje	możliwy	możliwy	możliwy
READ COMMITTED	nie występuje	nie występuje	możliwy	możliwy
REPEATABLE READ	nie występuje	nie występuje	nie występuje	możliwy
SERIALIZABLE	nie występuje	nie występuje	nie występuje	nie występuje

Baza MS SQL Server implementuje wszystkie wymienione poziomy izolacji transakcji.

POLECENIA MS SQL Serwera

```
SET IMPLICIT_TRANSACTIONS OFF;
```

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
```

```
BEGIN TRANSACTION;
COMMIT TRANSACTION;
ROLLBACK TRANSACTION;
```

UWAGA! Ze względu na potencjalny konflikt (nadanie kolejnego numeru) z inną transakcją w ramach zadania II zamiast `IDENT_CURRENT()` należy używać `SCOPE_IDENTITY()`