

Zarządzanie transakcjami

Badane poziomy izolacji transakcji:

1. READ COMMITTED
2. SERIALIZABLE
3. READ ONLY
4. blokada FOR UPDATE

Poziomy izolacji READ UNCOMMITTED oraz REPEATABLE READ nie są implementowane

1. READ COMMITTED

a) dirty read & non repeatable-read

T1	T2
set transaction isolation level read committed name 'test1'; select * from accounts where account_NUMBER = 11110002; 10000 update accounts set account_balance = 10002 where account_number = 11110002;	
	set transaction isolation level read committed name 'read committed test'; select * from ACCOUNTS where ACCOUNT_NUMBER = 11110002; 10000
commit;	
	select * from ACCOUNTS where ACCOUNT_NUMBER = 11110002; 10000

b) phantom read

T1	T2
set transaction isolation level read committed name 'read committed test'; select * from ACCOUNTS;	
	set transaction isolation level read committed name 'test2'; INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES (11110001, 10000); commit;
update accounts set account_balance = account_balance+100; select * from ACCOUNTS; Wynik 1.1	

Wynik 1.1

11110002	10102
11110003	14100
11110004	12100
11110005	16100
11110006	20100
11110007	19100
11110008	11100
11110009	13100
11110010	15100
11110011	17100
11110001	10100

Wnioski:

Izolacja typu READ COMMITTED chroni przed anomalią brudnego zapisu.

Transakcje na tym poziomie prawdopodobnie pracują na tabeli w czasie rzeczywistym, a operacje są przechowywane w buforze i zapisywane do tabali w momencie wywołania polecenia COMMIT.

2.SERIALIZABLE

a) lost update/dirty write

T1	T2
set transaction isolation level serializable name 'serializable test';	
	set transaction isolation level serializable name 'serializable test';
update accounts set account_balance = 10000 where account_number = 11110002;	
	update accounts set account_balance = 11000 where ACCOUNT_NUMBER = 11110002;
commit;	
	Error 2.1

Error 2.1

Error starting at line : 8 in command -

update accounts set account_balance = 11000 where ACCOUNT_NUMBER = 11110002

Error report -

ORA-08177: nie można szeregować dostępu dla tej transakcji

b) dirty read & non repeatable-read

T1	T2
set transaction isolation level serializable name 'serializable test'; select * from accounts where account_NUMBER = 11110002; 10000	
	set transaction isolation level serializable name 'serializable test'; select * from accounts where account_NUMBER = 11110002; 10000
update accounts set account_balance = 12000 where account_number = 11110002;	
	select * from accounts where account_NUMBER = 11110002; 10000
commit;	
	select * from accounts where account_NUMBER = 11110002; 10000

c) phantom read

T1	T2
set transaction isolation level serializable name 'serializable test';	
	set transaction isolation level serializable name 'serializable test'; select * from accounts; Wynik 2.1
INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES (11110000, 10000); commit;	
	Error 2.2

Wynik 2.1

11110002	12000
11110003	14100
11110004	12100
11110005	16100
11110006	20100
11110007	19100
11110008	11100
11110009	13100
11110010	15100

11110011 17100
11110001 10100

Error 2.2

update accounts set account_balance = account_balance+100;

Error report -

ORA-08177: nie można szeregować dostępu dla tej transakcji

Wnioski:

Poziom izolacji SERIALIZABLE zabezpiecza przed wystąpieniem każdej z badanych anomalii. Można wywnioskować, że transakcje izolowane na poziomie SERIALIZABLE otrzymują coś na kształt kopii tabeli, która w żadnym momencie sesji nie jest aktualizowana, oraz zestaw flag do informowania, o aktualizacji danych przez inną transakcję do ochrony przed anomaliami lost update oraz phantom read.

3. READ ONLY

a) anomalia lost update nie występuje, ze względu na brak możliwości modyfikacji tabeli w transakcji read only

b) dirty read/non repeatable read

T1	T2
set transaction read only name 'read only test';	
	set transaction isolation level read committed name 'test1'; select * from accounts where account_NUMBER = 11110002; 12000 update accounts set account_balance = 11000 where account_number = 11110002;
select * from accounts where account_NUMBER = 11110002; 11000	
	commit;
select * from accounts where account_NUMBER = 11110002; 11000	

c) phantom read

T1	T2
set transaction read only name 'read only test'; select * from ACCOUNTS; Wynik 3.1	
	set transaction isolation level read committed name 'test1'; INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES (11109999, 10000); select * from accounts; Wynik 3.2 commit;
select * from ACCOUNTS; Wynik 3.3	

Wynik 3.1

11110002	12000
11110003	14100
11110004	12100
11110005	16100
11110006	20100
11110007	19100
11110008	11100
11110009	13100
11110010	15100
11110011	17100
11110001	10100
11110000	10000

Wynik 3.2

11110002	12000
11110003	14100
11110004	12100
11110005	16100
11110006	20100
11110007	19100
11110008	11100
11110009	13100
11110010	15100
11110011	17100
11110001	10100
11110000	10000
11109999	10000

Wynik 3.3

11110002	12000
11110003	14100
11110004	12100
11110005	16100
11110006	20100
11110007	19100
11110008	11100
11110009	13100
11110010	15100
11110011	17100
11110001	10100
11110000	10000

Wnioski:

Poziom izolacji READ ONLY wydaje się działać na takich samych zasadach, jak izolacja na poziomie SERIALIZABLE, z zastrzeżeniem, że zezwala transakcji jedynie na polecenia odczytu. Chroni przed wszystkimi testowanymi anomaliami.

4. blokada FOR UPDATE

a) lost update

T1	T2
set transaction isolation level read committed; select * from accounts where account_number = 11110002 for update; wynik: oczekiwanie na wykonanie kwerendy	
	set transaction isolation level read committed; select * from accounts where account_number = 11110002 for update; wynik: oczekiwanie na wykonanie kwerendy

Brak reakcji na kolejne kwerendy

Wnioski:

Blokada FOR UPDATE nie działa tak jak na prezentacji.