

Sprawozdanie laboratorium 9.

1. Poziom izolacji READ_COMMITED

a) Dirty reads:

T1	T2
COMMIT	COMMIT
BEGIN	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (10000)	
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 15000 WHERE ACCOUNT_NUMBER = 111100002	
	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (10000)
ROLLBACK	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (10000)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (10000)

Przykład pokazuje, że na poziomie izolacji READ_COMMITED nie występują tzw. brudne odczyty (w tym przypadku transakcja T2 ma dostęp do zaktualizowanego stanu konta dopiero w momencie, w którym transakcja T1 wykona operację COMMIT).

b) Non – repeatable reads

T1	T2
BEGIN	BEGIN
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002 (15000)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (15000)
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 17500 WHERE ACCOUNT_NUMBER = 111100002;	
COMMIT	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002 (17500)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002 (17500)

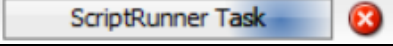
Wg przykładu przedstawionego powyżej, w poziomie izolacji READ COMMITTED występują anomalie nazywane niepowtarzalnymi odczytami. Podczas trwania transakcji T2 wykonywane są 2 zapytania sprawdzające stan tego samego konta. Dają one jednak różne wyniki, ponieważ w międzyczasie dokonana została aktualizacja stanu konta przez transakcję T1. Aktualizacja ta została zatwierdzona.

c) Phantoms

T1	T2
BEGIN	BEGIN
SELECT SUM(ACCOUNT_BALANCE), COUNT(*) FROM ACCOUNTS; (338555, 8)	
	SELECT SUM(ACCOUNT_BALANCE), COUNT(*) FROM ACCOUNTS; (338555, 8)
	INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES(SEQ_ACCOUNT_NUMBER.NEXTVAL, 22000);
	COMMIT
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = ACCOUNT_BALANCE + 1000;	
COMMIT	
SELECT SUM(ACCOUNT_BALANCE), COUNT(*) FROM ACCOUNTS; (369555, 9);	

Przykład przedstawia występowanie fantomów. Transakcja T1 dokonuje operacji aktualizacji stanów kont dla wartości innych, niż odczytała w poprzednim kroku. Dzieje się tak, ponieważ w międzyczasie transakcja T2 wprowadziła do tablicy nowy rekord.

d) Lost updates

T1	T2
BEGIN	
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 277145 WHERE ACCOUNT_NUMBER = 111100005; (277145)	BEGIN
	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 365 WHERE ACCOUNT_NUMBER = 111100005; 
COMMIT	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100005; (277145)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100005; (365)

Na poziomie izolacji READ COMMITED niemożliwe jest jednoczesne wykonanie aktualizacji tego samego rekordu przez dwie transakcje bez zatwierdzenia transakcji która jako pierwsza wykonała operację aktualizacji. Jednakże, po zatwierdzeniu, druga transakcja natychmiastowo nadpisuje rekord, nie zachowując zmienionych przez transakcję T1 zmian.

2. Poziom izolacji SERIALIZABLE

a) Dirty reads

T1	T2
COMMIT	COMMIT
BEGIN	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003; (9500)	
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 2500 WHERE ACCOUNT_NUMBER = 111100003	
	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003; (9500)
ROLLBACK	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (9500)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (9500)

Powyższy przykład potwierdza, że w poziomie izolacji SERIALIZABLE również nie występują niepotwierdzone odczyty.

b) Non – repeatable reads

T1	T2
BEGIN	BEGIN
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003 (9500)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003; (9500)
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 2500 WHERE ACCOUNT_NUMBER = 111100003;	
COMMIT	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003 (2500)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003 (9500)
	COMMIT
	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003; (2500)

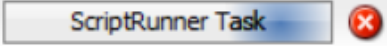
Poziom izolacji SERIALIZABLE radzi sobie również z występowaniem niepowtarzalnych odczytów. Jak widać na przykładzie powyżej, stan konta jaki „widzi” transakcja T2 ulega zaktualizowaniu dopiero po operacji COMMIT wykonanym przez transakcję T2.

c) Phantoms

T1	T2
BEGIN	BEGIN
SELECT SUM(ACCOUNT_BALANCE), COUNT(*) FROM ACCOUNTS; (362555, 8)	
	SELECT SUM(ACCOUNT_BALANCE), COUNT(*) FROM ACCOUNTS; (362555, 8)
	INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES(SEQ_ACCOUNT_NUMBER.NEXTVAL, 22000);
	COMMIT
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = ACCOUNT_BALANCE + 1000; (Error report - ORA-08177: nie można szeregować dostępu dla tej transakcji)	

Poziom izolacji **SERIALIZABLE** nie pozwala na występowanie fantomów podczas przeprowadzania transakcji. Podczas gdy transakcja T1 chce wykonać operację aktualizacji stanów kont napotyka informację o błędzie o braku możliwości szeregowania dostępu do transakcji. Transakcja T2 w międzyczasie dodała bowiem nowy rekord do tablicy, na którym operacja T1 miałaby również wykonać operację aktualizacji.

d) Lost updates

T1	T2
BEGIN	
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 10500 WHERE ACCOUNT_NUMBER = 111100006; (10500)	BEGIN
	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100006; (2900)
	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 13000 WHERE ACCOUNT_NUMBER = 111100006; 
COMMIT	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100006; (10500)	Error report - ORA-08177: nie można szeregować dostępu dla tej transakcji

Przykład demonstruje, że niemożliwe jest wystąpienie anomalii utraconych aktualizacji przy użyciu izolacji SERIALIZABLE. Po zatwierdzeniu aktualizacji transakcji T1, transakcja T2 natychmiastowo napotyka błąd.

3. Poziom izolacji READ ONLY

a) Dirty reads

T1	T2 (READ ONLY)
COMMIT	COMMIT
BEGIN	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (19500)	
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 10000 WHERE ACCOUNT_NUMBER = 111100002	
	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (19500)
ROLLBACK	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (19500)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (19500)

Poziom izolacji READ ONLY również nie pozwala na występowanie niepotwierdzonych odczytów.

b) Non – repeatable reads

T1	T2 (READ ONLY)
BEGIN	BEGIN
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002 (19500)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100002; (19500)
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 1000 WHERE ACCOUNT_NUMBER = 111100003;	
COMMIT	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003 (1000)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003 (19500)
	COMMIT
	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100003; (1000)

c) Phantoms / lost updates

W poziomie izolacji READ ONLY nie występują fantomy ani utracone aktualizacje, ponieważ w tym trybie nie można dokonywać modyfikacji tabel (CREATE, UPDATE, DELETE)

Analizując przykłady z punktów 2. i 3. można stwierdzić, że transakcje przeprowadzane w poziomie izolacji READ ONLY zabezpieczone są przed występowaniem tych samych typów anomalii co przy poziomie izolacji SERIALIZABLE.

4. Kwerenda FOR UPDATE (Poziom izolacji COMMITED READ)

T1	T2
BEGIN	
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100006; FOR UPDATE (10500)	BEGIN
UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 500 WHERE ACCOUNT_NUMBER = 111100006;	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100006; FOR UPDATE (10500)
COMMIT	
	UPDATE ACCOUNTS SET ACCOUNT_BALANCE = 42195 WHERE ACCOUNT_NUMBER = 111100006;
	COMMIT
SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100006; (42195)	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER = 111100006; (42195)

Przykład zastosowania klauzuli FOR UPDATE jest próbą uniknięcia anomalii utraty aktualizacji. Klauzula FOR UPDATE blokuje możliwość aktualizacji wskazanego przez transakcję T1 rekordu.