

## Sprawozdanie z laboratorium: Zarządzanie transakcjami

W sprawozdaniu przetestowano występowanie wybranych anomalii dla poziomów izolacji występujących w Oracle. W tym celu uruchomiono program sql developer i w dwóch oddzielnych sesjach symulowano działanie różnych transakcji, sprawdzając czy dany poziom izolacji chroni przed daną anomalią.

Każda sesja rozpoczyna się ustawieniem poziomu izolacji poleceniem:

```
ALTER SESSION SET ISOLATION_LEVEL= (poziom izolacji);
```

Powyższe polecenie nie zostało wpisane do tabel z kodem.

Dodatkowo, dla poziomów izolacji READ COMMITTED oraz SERIALIZABLE właściwy kod pisany jest czarną czcionką. Dla poziomu izolacji READ COMMITTED z klauzulą FOR UPDATE uwzględnia się czcionkę czarną oraz czerwoną.

### 1) Niezatwierdzony odczyt

	Transakcja 1	Transakcja 2
1	COMMIT;	COMMIT;
2	SET TRANSACTION READ WRITE NAME 'DIRTY READ 1';	SET TRANSACTION READ WRITE NAME 'DIRTY READ 2';
3	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004;	
4	UPDATE ACCOUNTS SET ACCOUNT_BALANCE=2000 WHERE ACCOUNT_NUMBER=11110004;	
5		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110002 <b>FOR UPDATE;</b>
6	ROLLBACK;	

#### 1.1. Poziom izolacji: READ COMMITTED

Ta izolacja CHRONI przed niezatwierdzonym odczytem. Wartość ACCOUNT\_BALANCE po wykonaniu się operacji w wierszu 3 wynosi 1000, w wierszu 4 wynosi 2000, a w wierszu 5 również wynosi 1000.

#### 1.2. Poziom izolacji: READ COMMITTED with FOR UPDATE

Jeżeli typ transakcji jest ustawiony na READ ONLY nie da się zastosować klauzuli FOR UPDATE. W przypadku ustawienia transakcji na READ WRITE ta izolacja CHRONI, operacja z wiersza 5 czeka na COMMIT bądź ROLLBACK transakcji 1.

#### 1.3. Poziom izolacji: SERIALIZABLE

Ta izolacja CHRONI przed niezatwierdzonym odczytem, dokładnie tak jak READ COMMITTED.

Wnioski: Otrzymane wyniki są zgodne z teorią.

## 2. Niepowtarzalny odczyt

	Transakcja 1	Transakcja 2
1	COMMIT;	COMMIT;
2	SET TRANSACTION READ WRITE NAME 'non-repetable read 1';	SET TRANSACTION READ WRITE NAME 'non-repetable read 2';
3	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004;	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004 FOR UPDATE;
4	UPDATE ACCOUNTS SET ACCOUNT_BALANCE=80 WHERE ACCOUNT_NUMBER=11110004;	
5	COMMIT;	
6		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004 FOR UPDATE;
7		COMMIT;

### 2.1. Poziom izolacji: READ COMMITTED

Ta izolacja NIE CHRONI przed tą anomalią. Wartość ACCOUNT\_BALANCE po wykonaniu się operacji w wierszu 3 wynosi 1000, w wierszu 4 wynosi 80, a w wierszu 6 również wynosi 80. Jest to niespełnienie warunku wymagającego, aby operacje SELECT tych samych rekordów transakcji 2 miały tą samą wartość

### 2.2. Poziom izolacji: READ COMMITTED with FOR UPDATE

Ta izolacja NIE CHRONI przed tą anomalią. Operacja z wiersza 3 czeka na COMMIT bądź ROLLBACK transakcji 1, jednak wartości z wiersza 3 oraz 6 są różne.

### 2.3. Poziom izolacji: SERIALIZABLE

Ta izolacja CHRONI przed tą anomalią. Wartości z kolumn 3 i 6 są równe 1000.

Wnioski: Otrzymane wyniki są zgodne z teorią.

### 3. Utracone aktualizacje

	Transakcja 1	Transakcja 2
1	COMMIT;	COMMIT;
2	SET TRANSACTION READ WRITE NAME 'Lost updates 1';	SET TRANSACTION READ WRITE NAME 'Lost updates 2';
3	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004 <b>FOR UPDATE;</b>	
4		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004 <b>FOR UPDATE;</b>
5	UPDATE ACCOUNTS SET ACCOUNT_BALANCE=100 WHERE ACCOUNT_NUMBER=11110004;	
6		UPDATE ACCOUNTS SET ACCOUNT_BALANCE=200 WHERE ACCOUNT_NUMBER=11110004;
7	COMMIT;	
8		COMMIT;
9	SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004;	
10		SELECT ACCOUNT_BALANCE FROM ACCOUNTS WHERE ACCOUNT_NUMBER=11110004;

#### 3.1. Poziom izolacji: READ COMMITTED

Ta izolacja NIE CHRONI przed tą anomalią. Wartość ACCOUNT\_BALANCE po wykonaniu się operacji w wierszach 3 i 4 wynosi 1000. Wartości otrzymane w wierszach 8 i 9 są równe 200, co jest efektem niepożądanym.

#### 3.2. Poziom izolacji: READ COMMITTED with FOR UPDATE

Ta izolacja CHRONI przed tą anomalią. Operacja z wiersza 3 wykonuje się normalnie, zaś operacja z wiersza 4 czeka na COMMIT bądź ROLLBACK transakcji 1. Tak więc dostęp do zmiany wartości rekordu ma tylko jedna transakcja w danym czasie.

#### 3.3. Poziom izolacji: SERIALIZABLE

W tej izolacji, wiersze 3 oraz 4 zwracają tę samą wartość równą 1000. Wiersz 5 wykonuje się bez komplikacji, natomiast transakcja czeka na wierszu 6. Po wykonaniu operacji z wiersza 7, pojawia się błąd: ORA-08177: nie można szeregować dostępu dla tej transakcji.

Zgodnie z teorią poziom izolacji SERIALIZABLE nie chroni powinien chronić przed tą anomalią, jednak z niewiadomego powodu sql developer nie pokrywa się z teorią.

Wnioski: Wszystkie otrzymane wyniki są zgodne z teorią. Poziom izolacji SERIALIZABLE nie chroni tej anomalii, dlatego błąd jest odpowiedni.

#### 4. Odczyt fantomowy

	Transakcja 1	Transakcja 2
1	COMMIT;	COMMIT;
2	SET TRANSACTION READ WRITE NAME 'Phantom READ 1';	SET TRANSACTION READ WRITE NAME ' Phantom READ 1'';
3	SELECT COUNT (*) FROM ACCOUNTS FOR UPDATE;	
4		INSERT INTO ACCOUNTS (ACCOUNT_NUMBER, ACCOUNT_BALANCE) VALUES (SEQ_ACCOUNT_NUMBER.NEXTVAL, 1000);
5		COMMIT;
6	SELECT COUNT (*) FROM ACCOUNTS FOR UPDATE;	
7	COMMIT;	

##### 4.1. Poziom izolacji: READ COMMITTED

Ta izolacja NIE CHRONI przed tą anomalią. Wartość operacji 3 jest równa 10, zaś po dodaniu kolejnego rekordu za pomocą operacji z wiersza 4 i operacji commit z wiersza 5, operacja z wiersza 6 zwraca wartość 11.

##### 4.2. Poziom izolacji: READ COMMITTED with FOR UPDATE

Ta izolacja NIE CHRONI przed tą anomalią. Dla operacji z wiersza 3 klauzula FOR UPDATE jest niedozwolona.

##### 4.3. Poziom izolacji: SERIALIZABLE

Ta izolacja CHRONI przed tą anomalią. Operacje 3 oraz 6 zwracają identyczną wartość, mimo że w wierszu 4 został dodany nowy wiersz do tabeli.

Wnioski: Otrzymane wyniki są zgodne z teorią. Operacja READ COMMITTED with FOR UPDATE jest niepotrzebna, gdyż READ COMMITTED chroni przed tą anomalią.