

Rozdział 17. Zarządzanie współbieżnością – zadania

Transakcja DML

1. Uruchom narzędzie Oracle SQL Developer i przyłącz się do bazy danych. Następnie rozpocznij nową transakcję, zmieniając pracownikowi o nazwisku MATYSIAK etat na ADIUNKT.
2. W drugim poleceniu usuń z relacji PRACOWNICY wszystkie informacje o pracownikach na etacie ASYSTENT.
3. Sprawdź (wykonaj zapytanie), czy wprowadzone przez Ciebie w punktach 1. i 2. modyfikacje rzeczywiście miały miejsce.
4. Zakończ transakcję z wycofaniem efektów wszystkich operacji, jakie miały miejsce w transakcji. Sprawdź, czy zmiany, wprowadzone przez polecenia z punktów 1. i 2., zostały anulowane.

Transakcja DDL


1. Rozpocznij nową transakcję, wykonując polecenie zwiększenia płacy podstawowej wszystkim adiunktom o 10%. Sprawdź, czy operacja została poprawnie wykonana (wykonaj odpowiednie zapytanie).
2. Wykonaj kolejne polecenie, które dokona modyfikacji typu kolumny `placa_dod` w relacji PRACOWNICY na `number(7,2)`.
3. Spróbuj anulować zmiany, wprowadzone w punktach 1. i 2., wykonując polecenie `rollback`. Sprawdź efekty wykonania tego polecenia.

Punkty bezpieczeństwa transakcji

1. Rozpocznij nową transakcję poleceniem, które pracownikowi MORZY doda do płacy dodatkowej 200 złotych. Następnie utwórz punkt bezpieczeństwa S1.
2. Daj Pracownikowi BIALY 100 złotych płacy dodatkowej i utwórz punkt bezpieczeństwa S2.
3. Usuń pracownika o nazwisku JEZIERSKI.
4. Wycofaj transakcję do punktu S1 i zobacz zawartość relacji PRACOWNICY.
5. Spróbuj wycofać transakcję do punktu S2. Czy polecenie zakończyło się sukcesem?

6. Wycofaj całą transakcję.

Współbieżność, blokady

1. Uruchom drugą sesję (sesja B) pracy z bazą danych. W narzędziu Oracle SQL Developer możesz to zrobić, klikając na przycisk  lub kombinacją klawiszów *Ctrl + Shift + N*. Sprawdź, czy rzeczywiście pracujesz w dwóch różnych sesjach (czy też nowo otwarta zakładka nie należy do tej samej sesji co zakładka, w której do tej pory pracowałeś/eś). W tym celu w obu sesjach wykonaj poniższe zapytanie, odczytujące unikalny identyfikator sesji (ang. *sid*):

```
select sys_context('USERENV', 'SID') from dual;
```

Jeśli uzyskałeś/eś dwie różne liczby, oznacza to, że pracujesz w dwóch różnych sesjach. Zapamiętaj (najlepiej zapisz!) identyfikator każdej sesji. Będą potrzebne w kolejnych ćwiczeniach.

2. W sesji A rozpocznij nową transakcję poleceniem, które podniesie płacę podstawową pracownikowi o nazwisku HAPKE o 100 zł. Sprawdź, jakie blokady zostały założone przez tą transakcję. W tym celu wykonaj poniższe zapytanie i zanalizuj otrzymane wyniki:

```
select * from table(sbd.blokady);
```

Blokady jest funkcją potokową, umieszczoną w pakiecie SBD. Funkcja ta zwraca wynik w postaci zbioru rekordów, pokazujących:

- *typ blokady (tabeli – TM lub rekordu – TX), jakie uzyskała bądź których żąda transakcja w sesji – kolumna typ,*
- *jakie blokady uzyskała transakcja w sesji – kolumna blokada_uzyskana,*
- *jakich blokad żąda transakcja w sesji – kolumna blokada_zadana,*
- *nazwa i właściciel obiektu, którego dotyczą informacje – kolumna obiekt,*
- *czy założona blokada blokuje inną transakcję – kolumna czy_blokuje_inna.*

Jeśli uruchomisz funkcję bez parametru, wówczas zaprezentowane informacje będą dotyczyły bieżącej sesji (tej, w której została uruchomiona funkcja). Jeśli przekazesz do funkcji parametr, będący identyfikatorem sesji, wyniki będą opisywały blokady założone lub żądane we wskazanej sesji.

3. W sesji B rozpocznij nową transakcję. Najpierw odczytaj wartość płacy podstawowej pracownika o nazwisku HAPKE. Czy zaobserwowałeś/eś zmiany, wprowadzone przez aktywną transakcję w sesji A? Następnie spróbuj pracownikowi HAPKE podnieść płacę dodatkową o 50 zł. Co teraz się dzieje z sesją? Czy obserwujesz efekt „zawieszenia” narzędzia?
4. Wróć do sesji A. Wykonaj ponownie zapytanie wyświetlające blokady dla sesji A. Czy widzisz różnicę (zwróć uwagę na kolumnę *czy_blokuje_inna*)? Wykonaj ponownie

zapytanie o blokady, tym razem jako parametr podaj identyfikator sesji B. Na uzyskanie jakiej blokady czeka sesja B?

5. Wycofaj transakcję w sesji A. Co się dzieje z sesją B? Odczytaj, jakie blokady posiada teraz transakcja w sesji B.
6. Wycofaj transakcję w sesji B.

Współbieżność, poziomy izolacji

1. W sesji A rozpocznij nową transakcję i określ jej poziom izolacji na `read committed`. Następnie odczytaj w tej transakcji wartość płacy podstawowej pracownika o nazwisku KONOPKA. Zapamiętaj odczytaną wartość.
2. W sesji B również rozpocznij nową transakcję z poziomem izolacji `read committed` a następnie odczytaj wartość płacy podstawowej pracownika KONOPKA. W kolejnym poleceniu transakcji ustaw płacę podstawową pracownika KONOPKA na wartość większą o 300 zł od wartości odczytanej przez zapytanie. Zakończ transakcję zatwierdzeniem wprowadzonych zmian.
3. W sesji A wykonaj polecenie, które ustawi pracownikowi KONOPKA płacę podstawową na wartość mniejszą o 200 zł od wartości odczytanej w punkcie 1. Czy polecenie zakończyło się powodzeniem? Zakończ transakcję zatwierdzeniem wprowadzonych zmian.
4. Określ, jaką anomalię zasymulowały operacje w p. 1, 2 i 3. Jaka jest aktualna płaca podstawowa pracownika KONOPKA? Jaka byłaby wartość płacy tego pracownika w sytuacji sekwencyjnego wykonania obu transakcji?
5. Wykonaj ponownie ćwiczenie, tym razem określając poziom izolacji transakcji w sesji A na `serializable`. Skomentuj zaobserwowane wyniki.
6. Spróbuj doprowadzić do wystąpienia anomalii skrośnego zapisu na poziomie izolacji `serializable`.

Zakleszczenie

1. Wygeneruj zakleszczenie trzech transakcji (musisz do sesji A i B dodać trzecią sesję, sesję C).

Blokady a klucze obce

1. W sesji A rozpocznij nową transakcję. Wykonaj w niej polecenie, które doda do relacji ZESPOLY nowy zespół o identyfikatorze 70 i nazwie „Zespół testowy”. Zatwierdź transakcję.

2. W sesji B rozpocznij nową transakcję poleceniem, które pracownikowi o nazwisku KONOPKA podwyższy płacę podstawową o 200 zł. Pozostaw transakcję aktywną.
3. W sesji A rozpocznij kolejną transakcję poleceniem, które spróbuje zmodyfikować identyfikator zespołu, dodanego w p. 1, na wartość 80. Co zaobserwowałeś/eś?
4. W sesji B wyświetl blokady, założone i żądane przez obie transakcje (z sesji A i sesji B). Skomentuj otrzymane wyniki.
5. Wycofaj transakcję w sesji B.
6. Usuń w sesji A dodany w p. 1. zespół. Zatwierdź transakcję.
7. Powtórzysz teraz zadanie, ale z istniejącym na kluczu obcym z relacji PRACOWNICY do ZESPOLY indeksem. Utwórz w sesji A indeks poniższym poleceniem:

```
create index test_fk on pracownicy(id_zesp);
```

8. Powtórz zadanie od p. 1. do 6. Co zaobserwowałeś/eś?
9. Usuń indeks, założony w p. 7., poniższym poleceniem:

```
drop index test_fk;
```

Zadania dodatkowe (dla chętnych)

1. Odroczone ograniczenia integralnościowe. Utwórz relację TEST z jednym atrybutem id typu number(5). Zdefiniuj na atrybucie id klucz podstawowy o nazwie TEST_PK, którego weryfikacja ma być przeprowadzana po zakończeniu transakcji modyfikującej relację TEST. Wstaw do relacji TEST kilka rekordów z tą samą wartością atrybutu id. Spróbuj zatwierdzić transakcję. Co zaobserwowałeś/aś? Odczytaj bieżącą zawartość relacji TEST. Zmień definicję ograniczenia w taki sposób, aby było weryfikowane natychmiast. Spróbuj teraz wstawić kilka rekordów z tą samą wartością atrybutu id.
2. Przetestuj działanie transakcji autonomicznych. W tym celu utwórz dwie tabele: TAB1 i TAB2 o identycznym schemacie: id number(1). Wstaw do obu tabel po dwa rekordy z dowolnymi wartościami kolumny id. Następnie utwórz procedurę ProcAut, która w ramach transakcji autonomicznej wstawi do TAB2 jeden rekord, policzy liczby rekordów w obu tabelach, wypisze tę informację na konsoli i dokona zatwierdzenia transakcji. W kolejnym kroku przygotuj kod anonimowego bloku PL/SQL, który wykona po kolei następujące czynności:
 - wstawi jeden rekord to tabeli TAB1,
 - policzy liczby rekordów w obu tabelach i wypisze tę informację na konsoli,
 - wywoła utworzoną wcześniej procedurę ProcAut,
 - znowu policzy liczby rekordów w obu tabelach i wypisze tę informację na konsoli,
 - wycofa transakcję,
 - ostatecznie policzy liczby rekordów w obu tabelach i wypisze tę informację na konsoli.

Uruchom blok (nie zapomnij przedtem ustawić wartość zmiennej `SERVEROUTPUT` na `ON!`) i zanalizuj otrzymane wyniki.