

Kurs programowania pod Oracle i Java

lista zadań nr 5

Dane są tabele:¹

```
Ksiazka( IdKsiazki, Autor, Tytul, IleStron )
Katalog( Sygn, IdKsiazki )
Czytelnik( Pesel, Imie, Nazwisko )
Wypozyczenie( Sygn, Pesel, DataOd, DataDo, Oddana )
Rezerwacja( IdKsiazki, Pesel, DataOd, DataDo )
Audyty( Id, Tabela, Operacja, Kiedy, Kto )
```

1. Napisz wyzwalacz, który momencie usuwania pozycji z tabeli **katalog** sprawdza, czy są jeszcze jakieś inne pozycje usuwanego tytułu i, jeśli nie ma, usuwa odpowiadającą książkę z tabeli **książka**.
[1p]
2. Napisz wyzwalacz, który przy wprowadzaniu pozycji do tabeli **wypozyczenie** sprawdza, czy wypożyczenie jest realizacją rezerwacji i, jeśli tak, usuwa odpowiadającą pozycję z tabeli **rezerwacja**, a jeśli nie, sprawdza, ile dana osoba ma już wypożyczonych książek i w przypadku, gdy jest to już ponad 10, daje jednodniowy czas wypożyczenia (interpretujemy to jako tylko do czytelnika)².
[2p]
3. Zrób na tabelach **książka**, **katalog**, i **czytelnik** wyzwalacze, które będą w tabeli **audyt** rejestrować wszystkie operacje **INSERT**, **UPDATE** i **DELETE** wykonywane na tych tabelach. Należy w rozwiązaniu tego zadania skorzystać z predykatów **INSERTING**, **UPDATING** i **DELETING**.
[1p]
4. Przygotować zestawienie w postaci tabelki, która w kolejnych wierszach będzie zawierać rodzaj operacji, czyli element z {**SELECT**, **INSERT**, **UPDATE**, **DELETE**}, a kolumnach element z {**AFTER**, **BEFORE**} × × {**ROW**, **STATEMENT**}. Na przecięciu wiersza i kolumny stawiamy znak „x” wtedy, gdy można wykonać daną operację podczas „mutowania” tabeli (należy rozważyć dwa przypadki: wykonanie zapytania na wszystkich wierszach oraz wykonanie zapytania tylko na wierszach, na których operacja uruchamiająca wyzwalacz operuje).
[2p]
5. Dane są tabele **Klub**(**Nazwa**, **Adres**) oraz **Zespol**(**Nazwa**, **IloscCzlonkow**), które są połączone relacją wiele–do–wielu **Koncert**(**NazwaKlubu**, **NazwaZespołu**, **Data**). Tworzymy widok **Koncerty**(**NazwaKlubu**, **AdresKlubu**, **NazwaZespołu**, **IloscCzlonkowZespołu**, **Data**). Poprzez utworzenie odpowiednich wyzwalaczy typu „zamiast” zdefiniuj operacje **INSERT**, **UPDATE** i **DELETE**, które powinny działać następująco:
 - **INSERT**. Przy wstawieniu wiersza do widoku najpierw należy sprawdzić, czy krotka już nie istnieje. Jeśli tak, kończymy z odpowiednim komunikatem błędu, natomiast jeśli nie, przechodzimy dalej i wykonujemy następujące czynności:
 - Sprawdzamy, czy istnieją klub i zespół o podanych nazwach. Jeśli tak, sprawdzamy ich dane i w przypadku, gdy coś się nie zgadza, kończymy operację z odpowiednim komunikatem błędu.
 - W przypadku, gdy wstawiana krotka zawiera dane klubu lub zespołu, którego w tabelach bazowych dla widoku jeszcze nie ma, uzupełniamy odpowiednie tabele bazowe.
 - Na końcu uzupełniamy tabelę **Koncert** reprezentującą relację wiele–do–wielu.
 - **UPDATE**. Przy modyfikacji musimy najpierw sprawdzić, czy nie występuje próba zmiany na coś, co już bazie istnieje, a następnie należy nanieść odpowiednie zmiany na tabele bazowe dla widoku.

¹Tabele do zadań 1–3.

²W obecnym kształcie jest nie do końca eleganckie — w części poświęconej programowaniu w Javie dopiszemy kod, który sprawi, że całość będzie już porządnym kawałkiem aplikacji

- DELETE. Przy usuwaniu najpierw usuwamy krotkę z relacji **Koncert**, następnie sprawdzamy, czy klub i zespół z usuniętej krotki występują jeszcze gdzieś w tabeli **Koncert**. Jeśli nie, usuwamy odpowiednie krotki z tabel **Klub** i **Zespół**.

[2p]

6. Dana jest tabela **Towar**(Id, Nazwa, NrMagazynu), gdzie NrMagazynu przyjmuje wartości A–J. Utwórz dwie sekwencje, gdzie

- pierwsza sekwencja będzie służyć do generowania identyfikatorów,
- druga sekwencja wraz funkcją DECODE będą służyć do tego, żeby kolejno wstawianym towarom nadawać cyklicznie kolejne magazyny A–J.

Napisz odpowiedni wyzwalacz dla operacji wstawiania oraz przygotuj wsad, który wstawi 140 towarów o losowych nazwach.

[2p]

Paweł Rajba