

Bazy danych i systemy informacyjne

laboratorium – Lista 2

Piotr Syga

21 października 2019

1. Utwórz bazę danych **Hobby**. Utwórz użytkownika `'U'@'localhost'` (lub `'U'@'%'`), gdzie *U* jest twoim imieniem, skonkatenowanym z dwoma ostatnimi cyframi indeksu. Ustaw dla tego użytkownika hasło będące twoim numerem indeksu czytany od tyłu. Nadaj utworzonemu użytkownikowi uprawnienia do Selectowania, wstawiania i zmieniania danych w tabeli, jednak nie do tworzenia usuwania i modyfikowania tabel.
2. Wewnątrz utworzonej bazy, utwórz tabele
 - **osoba**(id:int, imię:varchar(20), dataUrodzenia: date, plec: char(1)),
 - **sport**(id: int, nazwa: varchar(20), typ: enum(indywidualny, drużynowy, mieszany), lokacja: varchar(20)),
 - **nauka**(id:int, nazwa: varchar(20), lokacja: varchar(20)),
 - **inne**(id:int, nazwa:varchar(20), lokacja: varchar(20), towarzysze:bool),
 - **hobby**(osoba: int, id: int, typ: enum(sport, nauka, inne)).

Uwagi:

- Podkreślone atrybuty lub zbiory atrybutów ustaw jako klucz główny.
- Klucze będące samym ID powinny być automatycznie inkrementowane.
- Wszystkie osoby w bazie są pełnoletnie. (Zaimplementuj ograniczenia, sprawdź czy działają).
- Żadna z kolumn poza lokacją (we wszystkich tabelach, w których występuje) nie może przyjmować wartości NULL.
- W przypadku typu sportu wartością domyślną jest *drużynowy*. W przypadku kolumny towarzysze, w tabeli inne, wartością domyślną jest prawda.

3. Utwórz tabelę **zwierzak**, a następnie eksportuj do niej dane z tabeli **pet** w bazie **menagerie**. (Uwaga: schematy baz muszą sobie odpowiadać). Pobierz dane odnośnie właścicieli, uzupełnij brakujące atrybuty wartościami losowymi, a następnie eksportuj je do tabeli **osoba**.
4. Zmodyfikuj tabelę **osoba**, dodając kolumnę **nazwisko** (**varchar(50)**), która może przyjmować wartość **NULL**. Zmodyfikuj tabelę **zwierzak** poprzez usunięcie kolumny zawierającej imię właściciela, a dodając kolumnę zawierającą jego **ID**.
5. Zmodyfikuj tabelę **zwierzak** oraz **hobby**, tak by odniesienia do osoby lub hobby (odpowiednie **ID**) były kluczami obcymi.
6. Zmodyfikuj kolumnę **ID** w tabeli **inne**, by numeracja zaczynała się od 7000.
7. Napisz procedurę, przyjmującą dwa argumenty wejściowe: nazwę tabeli **@name** oraz liczbę rekordów **@num**, która po wywołaniu dodaje do tabeli **@name**, **@num** losowych rekordów. Zadbaj o spełnienie odpowiednich ograniczeń. Uruchom procedurę dla (**osoba**, 1000), (**sport**, 300), (**nauka**, 300), (**inne**, 550), (**hobby**, 1300).
8. Za pomocą konstrukcji **PREPARE statement** przygotuj zapytanie, które zwraca listę nazw wszystkich hobby. **ID** osoby oraz kategoria hobby powinny być podawane dopiero podczas wywołania **EXECUTE**.
9. Napisz funkcję lub procedurę, która po podaniu **ID** osoby wypisze jej wszystkie hobby. Problem z wyborem przeglądanej tabeli możesz rozwiązać przez **PREPARE statement**.
10. Zmodyfikuj procedurę (lub funkcję) zwracającą informacje o hobby użytkownika tak, by posiadane zwierzęta również liczyły się jako hobby. Jako nazwę hobby wpisz gatunek posiadanego zwierzęcia (informacja ta nie powinna się powtarzać).
11. Napisz trigger, który przy wstawianiu nowego hobby, dodaje informacje do odpowiednich tabel, tak by dodać nieistniejące wcześniej **ID** (wartości pozostałych atrybutów mogą być dowolne).
12. Napisz trigger (lub zmodyfikuj istniejący), który po usunięciu wpisu z tabeli **sport**, usuwa odpowiednie informacje z tabeli **hobby**.
13. Napisz trigger (lub zmodyfikuj istniejący), które przy usunięciu lub modyfikacji nazwy rekordu z tabeli **nauka** usuwa odpowiednie informacje z tabeli **hobby**.
14. Napisz trigger lub zmodyfikuj istniejący, w taki sposób by po usunięciu z tabeli osoby, wszystkie jego hobby były usuwane, a posiadane zwierzęta, przypisywane do innych osób.

15. Sprawdź czy wszystkie triggerzy z listy mogą istnieć w bazie jednocześnie. Zastanów się nad powodem.
16. Napisz widok, który zawiera informacje o każdym hobby oraz liczbie osób zajmujących się nim.
17. Napisz widok, który zawiera informacje o każdym użytkowniku oraz jego hobby i posiadanych zwierzętach.
18. Napisz procedurę bez argumentów wejściowych, z jednym argumentem wyjściowym (lub funkcję zwracającą), która wróci imię oraz wiek osoby posiadającej największą liczbę hobby.
19. Sprawdź, które polecenia można wykonać za pomocą użytkownika utworzonego w zadaniu 1.