Bazy danych i systemy informacyjne

laboratorium – Lista 2

Piotr Syga

25 października 2020

1. Utwórz bazę danych `Music`. 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 czytanym 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

   - `zespol` (id:int, nazwa:varchar(90), liczbaAlbumow: int),
   - `album`(tytul: varchar(90), gatunek: varchar(30), zespol:int),
   - `utwor`(id:int, tytul: varchar(90), czas: int, album: varchar(90), zespol: int).

   Uwagi:

   - Podkreślone atrybuty lub zbiory atrybutów ustaw jako klucz główny.
   - Klucze będące samym ID powinny być automatycznie inkrementowane.
   - Czas trawania utworów wyrażony jest w sekundach i musi być nieujemny. (Zaimplementuj ograniczenia, sprawdź czy działają).
   - LiczbaAlbumów początkowo wynosi `NULL` dla każdego z zespołów, w ramach zadań kolumna zostanie uzupełniona.
   - W tablie `utwor` zaimplementuj odpowiednie klucze obce, tak by powiązać każdy wiersz z albumem i zespołem.

3. Korzystając z bazy danych chinook, uzupełnij informacje w utworzonych tabelach. Nie importuj danych na temat filmów i seriali.

4. Napisz procedurę, która dla każdego artysty podliczy liczbę jego albumów w bazie danych i uzupełni odpowiednią kolumnę.

5. Napisz trigger, który za każdym razem gdy zmieni się zawartość tabeli `album`, odpowiednio zaktualizuje kolumnę `liczbaAlbumow`.

6. Napisz procedurę, która będzie sprawdzała czy w kolumnie `gatunek` są jedynie gatunki uwzględnione w bazie chinook. W przypadku niezgodności, zmień typ na taki, który w bazie chinook ma najmniejszy GenreId. W wersji rozszerzonej, takie sprawdzenie (i ew. korekta) powinno odbywać się za pomocą zapytania do bazy chinook.

7. Napisz procedurę, która jako parametr wejściowy będzie przyjmowała liczbę albumów do wygenerowania k, a następnie uzupełni wszystkie trzy tabele o odpowiednie informacje. Tytuły albumów mogą być losowe, np. *Album*<liczba>, każdy powinien zawierać nie mniej niż 6 utworów i nie więcej niż 20. Utworzone albumy mogą być przydzielane istniejącym już w bazie zespołom lub zespołom nowym, jednak nie powinna zachodzić sytuacja (dla $k > 2$), gdy wszystkie albumy przydzielane są do tego samego zespołu ani, że dla każdego albumu tworzymy nowy zespół.
   Zadbaj o weryfikację wartości k przed rozpoczęciem generowania. Możesz wykorzystać pętle, ogranicz natomiast korzystanie z dodatkowych plików i języków programowania.

8. Napisz funkcję lub procedurę, która dla podanej nazwy zespołu wypisze tytuł najdłuższego albumu tego zespołu.
   W wersji podstawowej zadania – załóż, że nazwa zespołu zawsze podana jest prawidłowo. W wersji rozszerzonej – w razie braku zespołu, zwróć informację o błędzie (informacja nie powinna być wyświetlana, jeśli zespół istnieje, ale nie wydał żadnego albumu).

9. Napisz widok zawierający informacje o nazwie zespołu, tytułach ich albumów oraz gatunku tych albumów.

10. Napisz trigger (lub zmodyfikuj istniejące), który po usunięciu albumu, usuwa z bazy wszystkie utwory z tego albumu. Zadbaj o to, by procedura usuwania przebiegła pomyślnie, pamiętaj o kolumnie `liczbaAlbumów`.

11. Napisz procedurę lub funkcję, która przyjmować będzie podciąg *patt*, a następnie usunie z bazy danych wszystkie albumy zespołów, w których nazwach występuje *patt*. Zadbaj by podany podciąg nie był pusty.
    Zastanów się jakie niebezpieczeństwa dla bazy (zarówno zawartych w niej danych, jak i struktury) może mieć umożliwienie użytkownikowi korzystanie z takiej procedury.

12. Utwórz index dla kolumny gatunek – zastanów się jaki jest odpowiedni typ indeksu. Podaj przykład zapytań odnoszących się do tej kolumny, które wykorzystują indeks oraz takich, które nie mogą z niego skorzystać.

13. Utwórz widok, który, dla każdego zespołu, będzie zawierał informację o liczbie jego utworów dłuższych niż 120s.

14. Za pomocą konstrukcji `PREPARE statement` przygotuj zapytanie, które zwraca listę wszystkich utworów konkretnego zespołu, zawartych na albumie o podanym gatunku. Nazwa zespołu oraz nazwa gatunku powinny być podane dopiero przy wywołaniu `EXECUTE`.

15. Za pomocą konstrukcji `PREPARE statement` przygotuj zapytanie, które najdłuższy utwór zamieszczony na albumie o podanym gatunku, który nie przekracza *thresh* sekund. Ograniczenie na długość oraz nazwa gatunku powinny być podane dopiero przy wywołaniu `EXECUTE`.

16. Korzystając z `DESCRIBE` lub `SHOW` napisz procedurę, która jako parametry wejściowe będzie przyjmowała nazwę tabeli t oraz nazwę kolumny k, a następnie zwróci ostatnią w porządku rosnącym wartość w tej tabeli, w podanej kolumnie. W wersji podstawowej wykorzystaj instrukcje warunkowe, które sprawdzą odpowiednie warunki. W wersji rozszerzonej wykorzystaj wskazane polecenia oraz `PREPARE statement`.

17. Wykorzystując CTE oraz rekursję napisz zapytanie, które dla podanych nieujemnych liczby całkowitych $n \geqslant k$ obliczą liczbę Stirlinga I rodzaju przy założeniach brzegowych $\begin{bmatrix} n \\ n \end{bmatrix} = 1$, $\begin{bmatrix} n \\ 0 \end{bmatrix} = 0$, $\begin{bmatrix} a \\ b \end{bmatrix} = 0$, dla $b > a$ oraz zależności rekurencyjnej $\begin{bmatrix} n \\ k \end{bmatrix} = (n-1)\begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}$.

18. Wykorzystując CTE oraz rekursję napisz zapytanie, które dla podanej nieujemnej liczby n obliczą liczbę Bella B_n. Sprawdź dla jakich n jesteś w stanie otrzymać rozwiązanie.

19. Sprawdź czy wszystkie triggery z listy mogą istnieć w bazie jednocześnie. Jeśli tak – uzasadnij, jeśli nie – zastanów się nad powodem.

20. Sprawdź, które polecenia można wykonać za pomocą użytkownika utworzonego w zadaniu 1.