

Projektowanie aplikacji z bazami danych

lista zadań nr 5

Poniższa lista ma za zadanie wypracować podstawowe umiejętności programowania w SQL i nie jest częścią budowanej w ramach zajęć aplikacji. Do realizacji zadań niezbędna będzie struktura biblioteki dołączona do przykładów wykład *T-SQL: podstawy*.

1. Utworzyć w T-SQL funkcję zwracającą w wyniku tabelę zawierającą sito Erastotenesa, przy czym zrobić dwie wersje: wykorzystującą w maksymalny sposób relacyjność SQL-a oraz natywną nie korzystającą z tej własności. Funkcję tą należy też zaimplementować w swoim ulubionym języku wyższego poziomu (np. C#, Java, Python). Przeprowadzić testy porównujące szybkość działania wszystkich implementacji.

[1p]

2. Napisz procedurę, która jako parametr bierze liczbę naturalną $n \geq 2$ (domyślną wartością powinna być liczba 15). W wyniku działania procedura ma drukować ładnie sformatowane pierwsze n wierszy trójkąta pascala. Wydrukowany trójkąt ma reprezentować trójkąt równoramienny, ale nie prostokątny. W tworzeniu procedury należy wykorzystać zmienne tabelowe. Pierwszych pięć wierszy ma wyglądać następująco:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

[1p]

3. Utworzyć tabele *imiona(id, imie)*, *nazwiska(id, nazwisko)* oraz *dane(imie, nazwisko)*. Następnie utworzyć procedurę, która dla zadanego parametru n do tabeli *dane* wstawi n losowo dobranych par (imie, nazwisko) uprzednio usuwając jej zawartość. Jeśli n będzie większe od połowy wszystkich możliwych kombinacji, należy to zakomunikować poprzez odpowiednie wywołanie instrukcji THROW. Klucz główny tabeli *dane* tworzą kolumny (imie,nazwisko), dlatego trzeba zadbać o to, żeby podczas generowania danych każdą parę (imie,nazwisko) wygenerować co najwyżej raz.

[2p]

4. Zaimplementować procedurę do tworzenia nowego czytelnika przyjmującą odpowiednie parametry. W ramach procedury należy zaimplementować odpowiednie walidacje: poprawność formatu PESEL, nazwisko z wielkiej litery i co najmniej dwuliterowe, poprawną datę urodzenia, zgodną z PESELelem. Wszystkie niezgodności należy zakomunikować poprzez odpowiednie wywołanie instrukcji THROW.

[1p]

5. Utworzyć procedurę, która jako parametr bierze tabelę (*czytelnik.id*) identyfikatorów czytelników, a jako wyniku zwraca tabelę o dwóch kolumnach (*czytelnik.id,suma_dni*), gdzie dla każdego przekazanego czytelnika dołączona jest sumaryczna liczba dni, na którą dany czytelnik wypożyczył książki. Zadanie należy zrobić w oparciu o bazę *biblioteka*.

[1p]

6. Przygotować raport opisujący sposoby przekazywania danych i odbierania danych przy wywoływaniu procedur i funkcji, czyli przykładowo, jak przy wywołaniu procedury wewnątrz procedury odebrać dane tabelaryczne do dalszego przetwarzania. Należy rozpatrzyć wszystkie możliwe przypadki i przygotować małe ilustrujące przykłady.

[2p]

7. Zaimplementować zadanie 6 z poprzedniej listy za pomocą kilku prostych instrukcji SQL z użyciem tabel tymczasowych. Przeprowadzić testy porównujące wydajność obu rozwiązań, tj. bieżącego i tego będącego jednym zapytaniem.

[1p]

8. Przygotować przykład pokazujący dostępność i czas życia tabel tymczasowych lokalnych i globalnych.

[1p]