Bazy Danych Projekt

Tomasz Polit

168160

Przykładowe zapytanie SELECT:

```
USE royal_family_tree;
  2 •
        SELECT
  3
            p.First_Name,
            p.Regnal_Number,
  4
  5
            t.Title Name,
            r.Relation_Type
  6
        FROM persons p
  9
        JOIN title_person tp
            ON p.Person_ID = tp.Person_ID
 10
 11
        JOIN titles t
            ON tp.Title_ID = t.Title_ID
 12
        JOIN primary_person_in_relation ppir
 13
 14
            ON ppir.PPIR_Person_ID = p.Person_ID
        JOIN secondary_person_in_relation spir
 15
 16
            ON spir.SPIR Person ID = p.Person ID
 17
        JOIN relations r
 18
            ON ppir.PPIR_Relation_ID = r.Relation_ID
 19
            OR spir.SPIR Relation ID = r.Relation ID
 20
     AND (r.Relation_Type LIKE '%fathe_' OR r.Relation_Type LIKE '%Fathe_'))
 21
 22
 23
        GROUP BY p.First Name, p.Regnal Number
        ORDER BY p.First_Name, p.Regnal_Number
 24
Export: Wrap Cell Content: 1A
   First_Name
            Regnal_Number
                                                                    Relation_Type
  Charles
                         King of the United Kingdom and other Commonwealth realms
                         King of the United Kingdom and other Commonwealth realms Great-Great-Grandfather
  George
                         King of the United Kingdom and other Commonwealth realms Great-Grandfather
  George
```

Wykorzystuję w nim wszystkie tablice. Poszukujemy imienia, numeru porządkowego, nazwy tytułu, typu relacji wszystkich osobób, które są dla Wiliama Windsora ojcem, dziadkiem, pradziadkiem itd., które posiadały tytuł króla Anglii.

Przykładowy UPDATE:

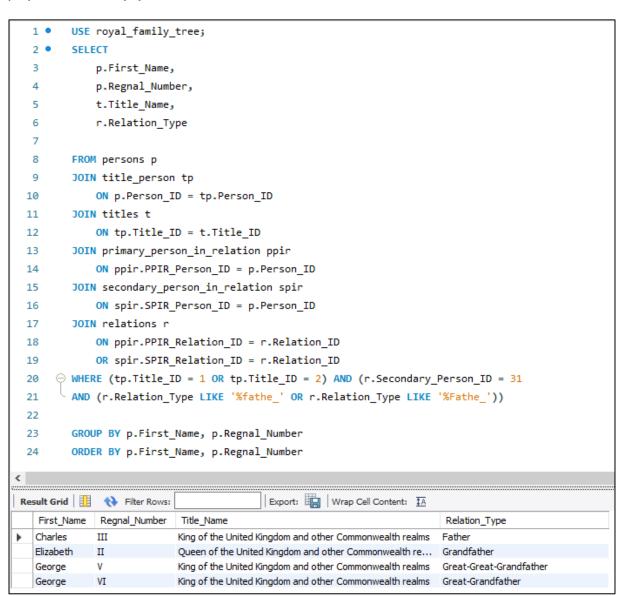
Zmienimę relację pomiędzy królową Elżbietą II oraz księciem Walii Wiliamem w taki sposób, że Elżbieta zostanie nazwana dziadkiem a nie babcią Wiliama.

```
USE royal_family_tree;

UPDATE relations SET Relation_Type = 'Grandfather'

WHERE Relation_ID = 851 AND Primary_Person_ID = 12 AND Secondary_Person_ID = 31
```

Urzywając widodznego powyrzej zapytania UPDATE zmienię wynik jaki pokarze poprzednio urzyty SELECT.

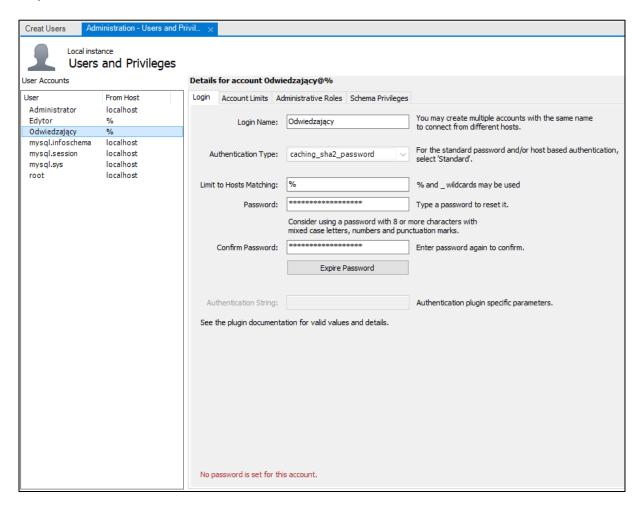


Jak widać na zdęciu powyrzej Elżbieta pojawiła się w wyniku jako dziadek Wiliama.

Utworzenie użytkownika zapytaniem SQL:

```
1 •
       CREATE USER 'Administrator'@'localhost'
           IDENTIFIED BY 'password';
 2
 3 •
       GRANT ALL
           ON *.*
4
 5
           TO 'Administrator'@'localhost';
 6
7 •
      CREATE USER 'Edytor'@'%'
8
           IDENTIFIED BY 'password';
       GRANT SELECT, INSERT, UPDATE, DELETE
9 •
           ON royal_family_tree.*
10
           TO 'Edytor'@'%';
11
12
13 •
       CREATE USER 'Odwiedzający'@'%';
14 •
       GRANT SELECT
15
           ON royal_family_tree.*
           TO 'Odwiedzający'@'%';
16
```

W ramach projektu w bazie danych utworzeni zostali urzytkownicy o odpowiedznich rolach.



Wynikiem zapytania są utworzeni użytkownicy, co widzimy na zdjęciu powyżej.

Baza danych posiada trzech użytkowników:

Administrator

Jego rola polega na zarządzaniu działaniem baz danych. Nie jest on ograniczony do żadnej bazy danych czy konkretnego zapytania. Rola ta jest bardzo ważna i jest przez to chroniona hasłem.

Edytora

Rola ta polega na wprowadzaniu niezbędnych zmian i poprawek do istniejącej bazy danych. Jest on jednak ograniczony tylko do niezbędnych do edycji zapytań wyłącznie wewnątrz bazy royal_family_tree. Rola ta jest chroniona hasłem.

Odwiedzający

Jest to rola, która jest przyznawana każdemu nowemu użytkownikowi. Nie jest ona chroniona hasłem, lecz jest ograniczona ściśle do zapytania SELECT tylko bazy danych royal_family_tree.

Funkcja duplikująca:

```
CREATE SCHEMA `new_royal_family_tree`;
2 •
       USE new_royal_family_tree;
3
4 •
      CREATE TABLE New_Persons LIKE royal_family_tree.persons;
      INSERT New_Persons SELECT * FROM royal_family_tree.persons;
7 • CREATE TABLE New_Relations LIKE royal_family_tree.relations;
      INSERT New_Relations SELECT * FROM royal_family_tree.relations;
9
10 • CREATE TABLE New_Primary_Person_in_Relation LIKE royal_family_tree.primary_person_in_relation;
11 • INSERT New_Primary_Person_in_Relation SELECT * FROM royal_family_tree.primary_person_in_relation;
12
13 • CREATE TABLE New_Secondary_Person_in_Relation_LIKE royal_family_tree.secondary_person_in_relation;
14 • INSERT New_Secondary_Person_in_Relation SELECT * FROM royal_family_tree.secondary_person_in_relation;
15
16 • CREATE TABLE New_Title_Person LIKE royal_family_tree.title_person;
17 •
      INSERT New_Title_Person SELECT * FROM royal_family_tree.title_person;
18
19 •
      CREATE TABLE New_Titles LIKE royal_family_tree.titles;
      INSERT New_Titles SELECT * FROM royal_family_tree.titles;
```

Powyższe komendy tworzą duplikat bazy danych, który może posłużyć, jako wariant testowy.