Bazy Danych II: Projekt Aplikacji Konsolowej z Drzewem Genealogicznym zaimplementowanym przy użyciu typu HIERARARCHYID

Karolina Klimek

Czerwiec 2023

Spis treści

1 Wstęp

Projekt zakłada implementację interfejsu API umożliwiającego tworzenie drzewa genealogicznego. Do tego celu wykorzystuje się typ Hierarchyid dostępny w SQL Server. Funkcjonlność zaimplementowano poprzez stworzenie tablic i procedur w języku tsql. Do zaprezentowania tych funkcjonalności została napisana aplikacja konsolowa w języku C#.

Typ hierarchii w SQL Server jest specjalnym typem danych, który pozwala na reprezentację danych w postaci hierarchicznej struktury drzewa. Ten typ umożliwia tworzenie struktur danych, które mają relacje rodzic-dziecko, na przykład organizacji, struktury plików lub struktury kategorii produktów. Typ hierarchii składa się z jednej kolumny, w której przechowywana jest struktura hierarchiczna drzewa. Kolumna ta może przechowywać hierarchiczne wartości jako ciągi znaków lub wartości binarne.

Aby operować na hierarchiach w SQL Server, można użyć specjalnych funkcji, takich jak GetRoot(), GetParent(), GetChild(), GetDescendant() lub GetAncestor(). Typ hierarchii jest szczególnie przydatny, gdy przechowujemy dane związane z hierarchicznymi strukturami oraz gdy wykonujemy zapytania operujące na takich strukturach danych.

W celu realizacji projektu tworzona jest tabela "Osoba", która zawiera kolumny takie jak imię, nazwisko, płeć, data urodzenia oraz gen typu Hierarchyid, które wskazuje na położenie w rodzinie tej osoby względem jej najstarszego męskiego członka, którego id przechowywane jest w kolumnie family_id. Istenie również kolumna mid będąca odnośnikiem do id matko osoby. Dzięki temu możliwe jest tworzenie powiązań między rekordami w tabeli.

Użytkownik aplikacji będzie miał możliwość tworzenia drzewa genealogicznego i dodawania członków jako rekordów w tabeli "Osoba". W celu zapewnienia funkcjonalności aplikacji, konieczne jest dodawanie rodziców dla poszczególnych członków w celu ustalenia relacji między nimi. Następnie użytkownik będzie mógł wyświetlać raporty dla wybranych rekordów, takie jak raport zawierający przodków danej osoby. Użytkownik będzie również mógł usuwać osoby z drzewa genealogicznego."

2 Baza danych

Projekt ten obejmuje tworzenie tabeli "Person" oraz implementację procedur, które umożliwiają dodawanie osób, pobieranie osób, aktualizowanie powiązań rodzinnych oraz usuwanie osób i relacji zrealizowane w SQL Server.

2.1 Struktura tabeli "Person"

Tabela "Person" składa się z następujących kolumn:

- **id** (INT, PRIMARY KEY) Unikalny identyfikator osoby.
- **mid** (INT) Identyfikator matki osoby - z założonym ograniczeniem: wartość tutaj wpisana musi nawiązywać do istniejącego już id innej osoby.
- **family_id** (INT) Identyfikator rodziny, do której osoba należy.
- **firstname** (VARCHAR(50)) Imię osoby.
- **lastname** (VARCHAR(50)) Nazwisko osoby.
- **dateofbirth** (DATE) Data urodzenia osoby.
- **gender** (VARCHAR(7)) Płeć osoby.
- **gen** (HIERARCHYID): Hierarchiczny identyfikator osoby.

2.2 Procedury

Tabela "Person" składa się z następujących kolumn:

- **GetProperId** Ta funkcja zwraca najniższy dostępny identyfikator, który może być użyty podczas dodawania nowej osoby do tabeli "Person".
- **AddPerson** Procedura "AddPerson" służy do dodawania nowej osoby do tabeli "Person". Przyjmuje parametry: @fname (VARCHAR(50)) - imię osoby, @lname (VARCHAR(50)) - nazwisko osoby, @dob (VARCHAR(20)) - data urodzenia osoby w formacie "YYYY-MM-DD", @gender (VARCHAR(7)) - płeć osoby.
- **GetAllPeople** Procedura "GetAllPeople" zwraca wszystkie osoby z tabeli "Person" w postaci identyfikatora, imienia i nazwiska.
- **GetPerson** Procedura "GetPerson" zwraca szczegółowe informacje o osobie o określonym identyfikatorze. Przyjmuje parametr @id (INT) - identyfikator osoby.

- **UpdateHidDescendants** Procedura "UpdateHidDescendants" aktualizuje hierarchiczne identyfikatory (gen) i identyfikatory rodzinne (family_id) dla wszystkich potomków danej osoby o określonym identyfikatorze. Przyjmuje parametry: @id (INT) - identyfikator osoby, @newhid (VARCHAR(50)) - nowy hierarchiczny identyfikator (gen), @newfid (INT) - nowy identyfikator rodziny.

- **AddParent** Procedura "AddParent" służy do dodawania rodzica dla osoby. Przyjmuje parametry: @idp (INT) - identyfikator rodzica, @idc (INT) - identyfikator dziecka.

- **RemovePerson** Procedura "RemovePerson" usuwa osobę o określonym identyfikatorze wraz z jej wszystkimi potomkami. Przyjmuje parametr @id (INT) - identyfikator osoby.

- **RemoveRelationship** Procedura "RemoveRelationship" usuwa relację pomiędzy rodzicem a dzieckiem. Przyjmuje parametry: @idp (INT) - identyfikator rodzica, @idc (INT) - identyfikator dziecka.

- **GetDescendants** Procedura "GetDescendants" zwraca potomków z linii męskiej osoby o określonym identyfikatorze. Przyjmuje parametr @id (INT) - identyfikator osoby.

- **GetAncestors** Procedura "GetAncestors" zwraca przodków osoby o określonym identyfikatorze z linii męskiej oraz ich matki. Przyjmuje parametr @id (INT) - identyfikator osoby.

3 Aplikacja konsolowa

Aplikacja konsolowa została napisana w języku C# i łączy się z bazą danych poprzez interfejs ADO.NET.

3.1 API

API zaimplementowane zostało w pliku DBconnect.cs i obsługuje wywoływanie procedur zarządzających drzewem genealogicznym z bazy danych.

Klasa "DBconnect" posiada pole "connectionString", które przechowuje informacje potrzebne do połączenia z bazą danych. Konstruktor klasy przyjmuje nazwę serwera i tworzy łańcuch połączenia.

3.1.1 Metody klasy DBconnect

- **bool addPerson(string fname, string lname, string gender, string birth) - publiczna** służy do dodawania nowej osoby do bazy danych. Przyjmuje parametry takie jak imię, nazwisko, płeć i data urodzenia osoby. Metoda wykonuje procedurę składowaną w bazie

danych AddPerson, aby dodać osobę do tabeli "Person". Zwraca prawdę jeśli udało się wykonać procedurę i fałsz jeśli nie.

- **int getNextId() - publiczna** zwraca następne dostępne ID dla nowej osoby. Wykonuje zapytanie do bazy danych, aby uzyskać wartość za pomocą procedury składowanej "GetProperId".
- **DataSet getPersonsData(int id) - publiczna** pobiera dane osoby o podanym identyfikatorze z bazy danych. Wykonuje procedurę składowaną "GetPerson" z id podanym jako argument i zwraca zestaw danych w formie DataSet.
- **void getPerson(int id) - publiczna** pobiera dane osoby o podanym identyfikatorze z bazy danych i wyświetla je na konsoli. Wykorzystuje metodę "getPersonsData" i "printPersonData".
- **bool removePerson(int id) - publiczna** usuwa osobę o podanym identyfikatorze z bazy danych. Wykonuje procedurę składowaną "RemovePerson". Zwraca prawdę jeśli udało się wykonać procedurę i fałsz jeśli nie.
- **void printPersonData(DataSet dataSet) - publiczna** wyświetla dane osoby, takie jak ID, imię, nazwisko, płeć i data urodzenia otrzymane w formacie DataSet.
- **void printPersonData2(DataRow dataSet) - publiczna** wyświetla dane osoby w formacie ID, imię i nazwisko otrzymane ze zmiennej DataRow.
- **void printAncestorsWithGens(DataRow dt) - publiczna**wyświetla dane przodka z określoną relacją na podstawie numeru pokolenia i płci, które otrzymała z danych typu DataRow otrzymanych wcześniej przez wykonanie procedury GetAncestors w bazie danych.
- **void printDescendantsWithGens(DataRow dt) - publiczna** wyświetla dane potomka osoby, które otrzymała z danych typu DataRow otrzymanych wcześniej przez wykonanie procedury GetAncestors w bazie danych.
- **string determineRelationshipAnc(int num, string gender) - publiczna** określa relację przodka na podstawie numeru pokolenia i płci.
- **void printAllPeople() - publiczna** pobiera dane wszystkich osób z bazy danych i wyświetla je na konsoli.
- **DataSet getEveryone() - publiczna** pobiera dane wszystkich osób z bazy danych poprzez wykonanie procedury GetEveryone i zwraca zestaw danych w formie DataSet.

- **bool isIdInDb(int id) - publiczna** sprawdza, czy podane ID istnieje w bazie danych i zwraca prawdę jeśli tak, a fałsz jeśli nie.
- **bool removeRelationship(int idp, int idc) - publiczna** usuwa relację między dwiema osobami na podstawie ich identyfikatorów. Pierwszym argumentem jest id rodzica a drugim id dziecka. Zwraca prawdę jeśli procedurę RemoveRelationship udało się wykonać, a fałsz jeśli nie.
- **DataSet getPersonData(int id) - publiczna** pobiera dane osoby o podanym identyfikatorze z bazy danych i zwraca zestaw danych w postaci DataSet
- **bool modifyRelationship(int id1, int id2, string rel) - publiczna** modyfikuje relację między dwiema osobami na podstawie ich identyfikatorów i podanej relacji poprzez wywołanie metody AddPerson. Zwraca wartość otrzymaną
- **addParent(int idc, int idp) - prywatna** dodaje relację rodzica do dziecka na podstawie identyfikatorów poprzez wykonanie procedury AddParent. Pierwszy identyfikator, który przyjmuje należy do dziecka, a drugi do rodzica. Zwraca prawdę jeśli procedurę udało się wykonać, a fałsz jeśli nie.
- **void findAncestors(int id) - publiczna** pobiera dane zwrócone przez getAncestors z argumentem identyfikującym osobę w bazie i wyświetla je na konsoli.
- **DataSet getAncestors(int id) - publiczna** pobiera dane przodków osoby o podanym identyfikatorze z bazy danych poprzez wykonanie procedury getAncestors i zwraca zestaw danych.
- **void findDescendants(int id) - publiczna** pobiera dane zwrócone przez getDescendant z argumentem identyfikującym osobę w bazie i wyświetla je na konsoli.
- **DataSet getDescendants(int id) - publiczna** pobiera dane przodków osoby o podanym identyfikatorze z bazy danych poprzez wykonanie procedury getDescendants i zwraca zestaw danych.

3.2 Obsługa zapytań użytkownika

Aby wygodnie zarządzać swoją bazą danych, użytkownik ma do dyspozycji aplikację konsolową.

3.2.1 Program.cs

W pliku Program.cs znajduje się klasa MainRun, która pełni funkcję punktu wejścia do programu. W niej znajduje się statyczne wywołanie funkcji Main, która wywołuje instancję klasy GenealogyUI.

3.2.2 FamilyTreeUI.cs

Klasa GenealogyUI jest odpowiedzialna za obsługę interfejsu użytkownika i interakcji z aplikacją FamilyTree. Zapewnia ona menu wiersza poleceń, w którym użytkownik może wykonywać różne operacje związane z zarządzaniem drzewem genealogicznym, takie jak dodawanie osób, modyfikowanie relacji, pobieranie przodków i potomków, usuwanie osób i relacji oraz pobieranie danych osoby.

Pola

- (db (DBconnect)) Instancja klasy DBconnect używana do nawiązywania połączenia z bazą danych i wykonywania operacji na bazie danych.

Konstruktory

- **GenealogyUI()** Inicjalizuje nową instancję klasy GenealogyUI i nawiązuje połączenie z bazą danych.

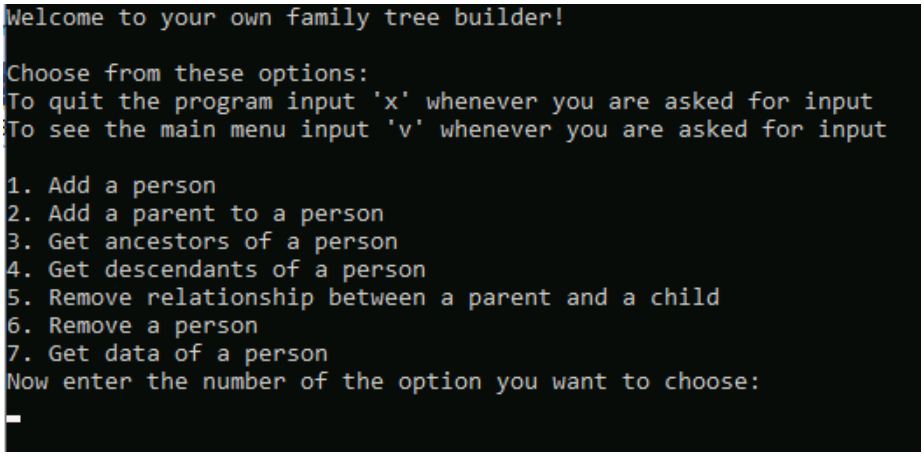
Metody

- **void choicesMenu() - publiczna** Wyświetla główne menu i obsługuje wprowadzanie użytkownika w celu wybrania różnych opcji.
- **globalChoicesListing() - publiczna** Wyświetla opcje wyjścia z programu lub powrotu do głównego menu.
- **void optionToQuit(string opt) - publiczna** - Wyświetla opcje wyjścia z programu lub powrotu do głównego menu.
- **void getAncestors() - publiczna** Wyświetla listę wszystkich osób w bazie danych i prosi użytkownika o podanie ID osoby, dla której mają zostać pobrani przodkowie.
- **void getDescendants() - publiczna** Wyświetla listę wszystkich osób w bazie danych i prosi użytkownika o podanie ID osoby, dla której mają zostać pobrani potomkowie.
- **void getPersonsData() - prywatna** Wyświetla listę wszystkich osób w bazie danych i prosi użytkownika o podanie ID osoby, dla której mają zostać pobrane dane.
- **void addPerson() - prywatna** Prosi użytkownika o wprowadzenie szczegółów nowej osoby i dodaje ją do bazy danych.
- **string getDate() - prywatna** Prosi użytkownika o wprowadzenie daty urodzenia w formacie "rrrr-mm-dd" i zwraca wprowadzoną datę jako ciąg znaków.
- **void printRelationshipTypes() - prywatna** Wyświetla rodzaje relacji do modyfikacji relacji między dwiema osobami.

- **string inputRelationship() - prywatna** Prosi użytkownika o wprowadzenie rodzaju relacji i zwraca wprowadzoną relację jako ciąg znaków.
- **string inputGender() - prywatna** Prosi użytkownika o wprowadzenie płci (f - kobieta, m - mężczyzna) i zwraca wprowadzoną płeć jako ciąg znaków.
- **int getIdFromUser(int id) - prywatna** Prosi użytkownika o wprowadzenie ID i zwraca wprowadzone ID jako liczbę całkowitą.
- **void modifyRelationship() - prywatna** Wyświetla listę wszystkich osób w bazie danych i prosi użytkownika o wprowadzenie ID dwóch osób, dla których mają zostać zmodyfikowane relacje.
- **string establishRelationship(int id1, int id2)** Wyświetla nazwy dwóch osób i prosi użytkownika o wprowadzenie rodzaju relacji między nimi.
- **void removeRelationship() - prywatna** Wyświetla listę wszystkich osób w bazie danych i prosi użytkownika o wprowadzenie ID rodzica i dziecka, dla których mają zostać usunięte relacje.
- **bool confirmation() - prywatna** Prosi użytkownika o potwierdzenie (y - tak, n - nie) i zwraca true lub false w zależności od wprowadzonego wyboru.

3.3 Przykłady użycia

Po włączeniu aplikacji użytkownikowi ukazuje się ekran menu z instrukcjami.



Rysunek 1: Główne menu.

Użytkownik może wpisać liczbę z listy. Jeśli input będzie niepoprawny, to aplikacja poinformuje o tym i poprosi o ponownie wpisanie wartości. Użytkownik może zakończyć działanie aplikacji w każdym momencie, w którym jest poproszony o wpisanie wartości poprzez naciśnięcie "x" i Enter. Wpisanie "v" i Enter spowoduje ponowne wyświetlenie głównego menu.

3.3.1 Dodawanie osoby do bazy

W menu dodawania osoby użytkownik zostaje zapytany o imię, nazwisko, płeć i datę urodzenia. Jeśli dane zostaną wprowadzone poprawnie i udało się nawiązać połączenie z bazą to osoba zostanie dodana. Wpisanie błędnego formatu płci poskutkuje prośbą o ponowienie wpisu. Wpisanie niepoprawnego formatu daty spowoduje pojawieniem się informacji o niepowodzeniu dodania osoby do bazy.

```
-> Add a person
CREATE A PERSON:
ENTER FIRST NAME:

Anna

ENTER LAST NAME:

Czerwinska

ENTER GENDER     f - female, m - male:

f
Enter date of birth in format yyyy-mm-dd
2001-06-02
First name: Anna
Last name: Czerwinska
Gender: f
Birthday: f
Successfully added a person
```

Rysunek 2: Menu dodawania osoby.

3.3.2 Dodawanie relacji między osobami

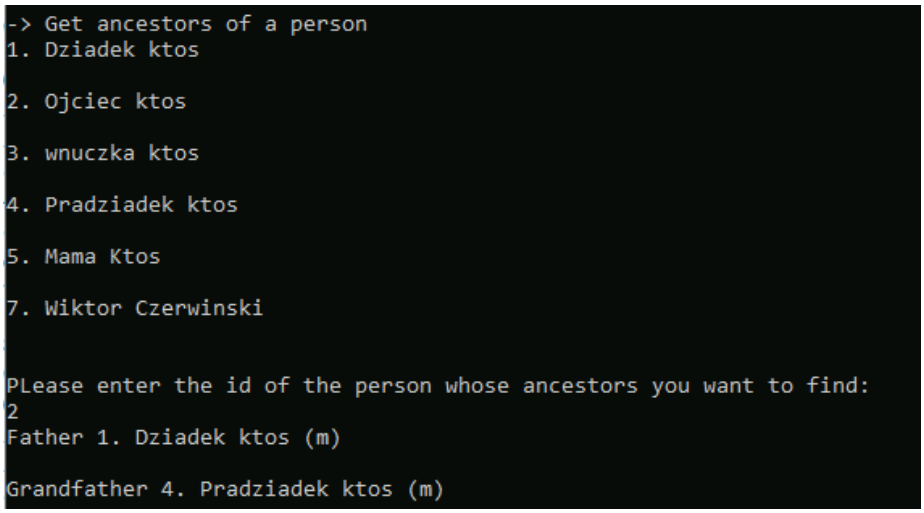
W menu dodawania relacji między osobami użytkownikowi zostanie wyświetlona lista wszystkich osób z bazy. Potem zostanie poproszony o podanie dwóch id, jednego dla rodzica, a drugiego dla dziecka, między którymi chce utworzyć relacje. W ostatnim kroku zostanie zapytany o typ relacji między osobami. "p" - oznacza, że osoba pierwsza ma być rodzicem dla drugiej. "c" - osoba pierwsza ma być dzieckiem drugiej. Aplikacja poinformuje o powodzeniu lub niepowodzeniu modyfikacji.

```
-> Modify relationship between two people
1. Dziadek ktos

2. Ojciec ktos

3. wnuczka ktos

4. Pradziadek ktos

5. Mama Ktos

6. Anna Czerwinska

7. Wiktor Czerwinski


Enter the id number of the first person
6
6
Is this the person you were thinking of? if yes type 'y', if no type 'n'
y

Enter the id number of the second person
7
7
Is this the person you were thinking of? if yes type 'y', if no type 'n'
y
6. Anna Czerwinska (f)
date of birth:  02.06.2001 00:00:00

7. Wiktor Czerwinski (m)
date of birth:  20.05.1980 00:00:00

Who is Anna Czerwinska to Wiktor Czerwinski?
'p' - a is a parent of b
'c' - a is a child of b
c
Modyfying successful
```

Rysunek 3: Menu dodawania relacji między osobami.

3.3.3 Wypisywanie przodków

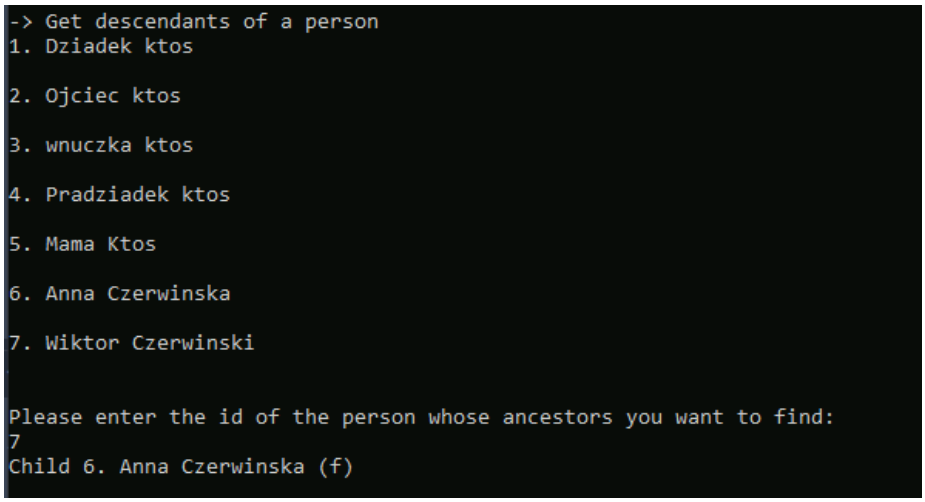
W menu wyświetlania przodków użytkownikowi zostanie wyświetlona lista wszystkich osób z bazy. Użytkownik zostaje poproszony o wpisanie id osoby, kórej przodków z linii męskiej chce zobaczyć. Jeśli w bazie znajduje się matka osoby, to również zostanie wyświetlona. Obok nazwiska osoby spokrewnionej znajduje się informacja o relacji jaką ma z osobą, której przodków wyświetlamy.



Rysunek 4: Menu wyświetlania przodków.

3.3.4 Wypisywanie potomków

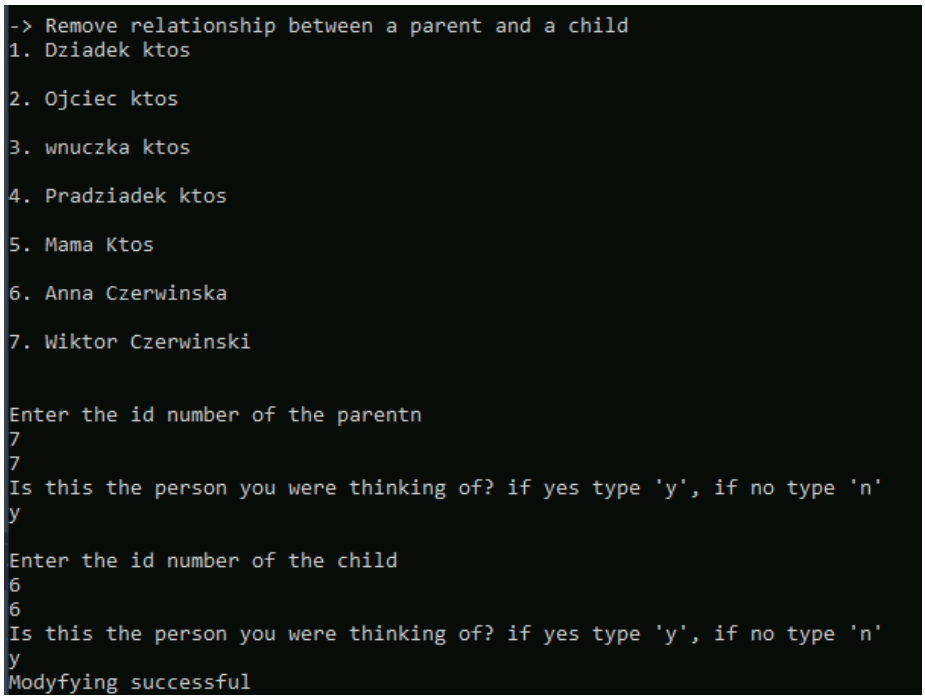
W menu wyświetlania potomków użytkownikowi zostanie wyświetlona lista wszystkich osób z bazy. Użytkownik zostaje poproszony o wpisanie id osoby, kórej potomków z linii męskiej chce zobaczyć. Zostaną również wyświetlone wszystkie kobiety, które jako rodzica mają członka bezpośredniej linii męskiej. Obok nazwiska osoby spokrewnionej znajduje się informacja o relacji jaką ma z osobą, której przodków wyświetlamy.



Rysunek 5: Menu wyświetlania potomków.

3.3.5 Usuwanie relacji

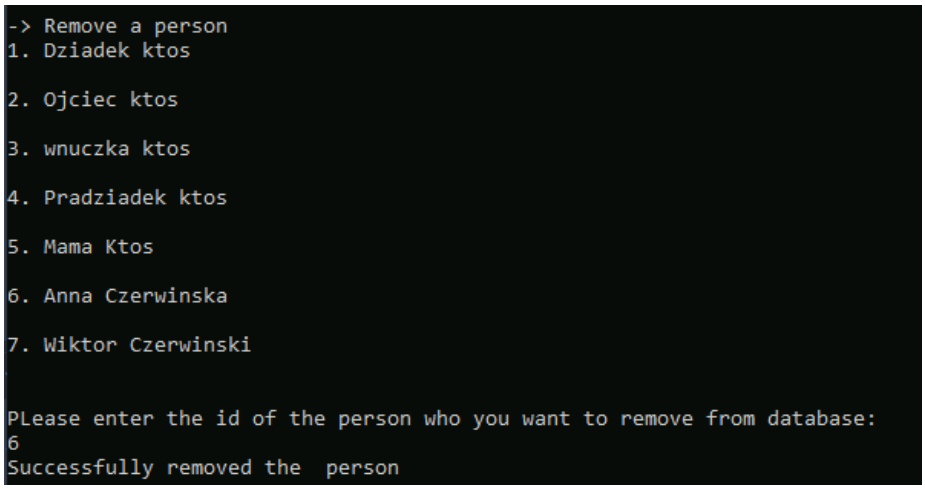
W menu wyświetlania potomków użytkownikowi zostanie wyświetlona lista wszystkich osób z bazy. Użytkownik poproszny zostanie o wpisanie id rodzica, a następnie id dziecka, między którymi tę relację chce usunąć. Aplikacja poinformuje o powodzeniu lub niepowodzeniu modyfikacji.



Rysunek 6: Menu usuwania relacji.

3.3.6 Usuwanie osoby

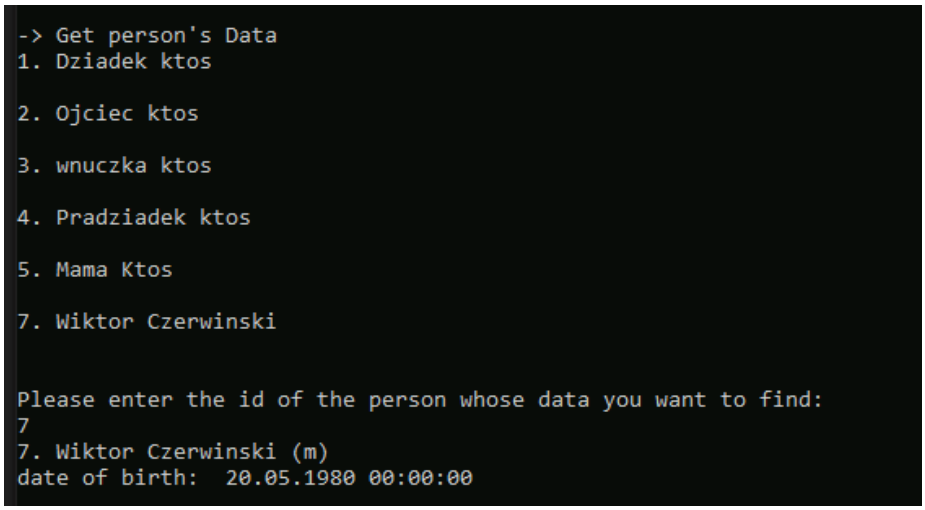
W menu usuwania osoby zostanie wyświetlona lista wszystkich osób z bazy. Użytkownik poproszny zostanie o wpisanie id osoby, o której informacje chce zobaczyć. Aplikacja poinformuje o powodzeniu lub niepowodzeniu usuwania.



Rysunek 7: Menu usuwania osoby.

3.3.7 Wyświetlanie informacji o osobie

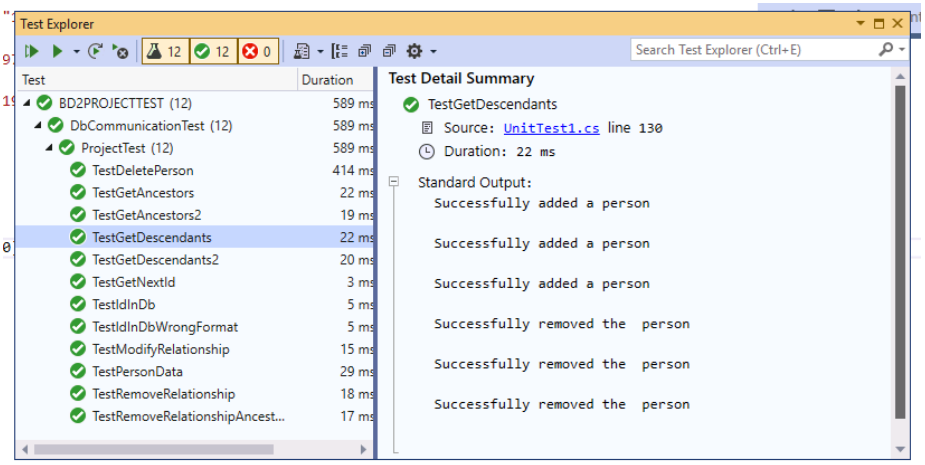
W menu wyświetlania informacji o osobie użytkownikowi zostanie wyświetlona lista wszystkich osób z bazy. Użytkownik poproszny zostanie o wpisanie id osoby, o której informacje chce zobaczyć. Zostanie wyświetlone imię, nazwisko, płeć i data urodzenia osoby o wprowadzonym identyfikatorze.



Rysunek 8: Menu usuwania osoby.

4 Test api

Na podstawie projektu zawartego w BD2projekt, który zawiera aplikację konsolową oraz API, został wygenerowny plik DLL, który dodano do zależności dla projektu BD2PROJECTETST, który testuje fukncjonalność API.



Rysunek 9: Wszystkie testy jednostkowe.

W pliku UnitTests1.cs znajduje się dwanaście testów jednostkowych, które kolejno tesują poszczególne metody z klasy DBconnect, będącym biblioteką dla tego projektu. Wszystkie testy są zdawane z powodzeniem.
Testowane są metody getNextId(), addPerson(), isInDb(), removePerson(int), modifyRelationship(int, int, string), getAncestors(int), getDescendants(int), removeRelationship(int, int).

5 Uruchamianie projektu

Projekt zrealizowany jest na platformmie udostępnionej na zajęciach. Aby uruchomić całość należy wykonać skrypt "proj.sql" w aplikacji Microsoft SQL Server Management Studio jako admin.
Plik wykonawczy aplikacji nosi nazwę "BD2Project". W folderze BD2Project znajduje się kod źródłowy tego projektu.
Kod źródłowy testów znajduje się w folderze BD2PROJECTTEST.