

# Programowanie pod Windows

## Zestaw A

Komunikacja z bazą danych

2024-05-07

Liczba punktów do zdobycia: **11/84**

Zestaw ważny do: 2024-05-21

1. (**1p**) Zainstalować SQL Server w dowolnej wersji dla programisty (Developer, Express, LocalDB). Zorientować się w dokumentacji czym charakteryzują się poszczególne wersje.
2. (**1p**) Przygotować bazę danych Microsoft SQL Server zawierającą dane osobowe i adresy przykładowej grupy studentów.

Model bazy danych zawiera trzy tabele

- tabelę **Student** z polami ID, Imię, Nazwisko, DataUrodzenia
- tabelę **Adres** z polami ID, Ulica, NrDomu, NrMieszkania, KodPocztowy, ID\_MIEJSCOWOSC
- tabelę **Miejscowosc** z polami ID, Nazwa

Tabele **Student** i **Adres** powinny być połączone relacją wiele-wiele (to wymaga pomocniczej tabeli!). Tabele **Adres** i **Miejscowosc** powinny być połączone relacją jeden-wiele

3. (**2p**) Użyć ADO.NET do połączenia do bazy danych, pobierania danych, dodawania, modyfikacji i usuwania. Ściśle - pokazać pracę z bazą danych za pomocą obiektów **SqlConnection**, **SqlCommand**, **SqlDataReader**.

W przypadku dodawania danych - oprogramować scenariusz dodawania studenta z adresem i miejscowością. Formalnie - przygotować funkcję o takiej liczbie parametrów jaka jest niezbędna do rejestracji pełnych danych (Imię, Nazwisko, DataUrodzenia, Ulica, NrDomu, ..., Nazwa):

- funkcja sprawdza czy osoba o podanym imieniu, nazwisku i dacie urodzenia już istnieje w bazie, jeśli tak to pomija dodawanie
  - funkcja sprawdza czy w tabeli Miejscowości znajduje się miejscowość o podanej nazwie. Jeśli nie - dodaje, jeśli tak, pobiera identyfikator
  - funkcja używa identyfikatora miejscowości żeby zarejestrować adres
  - funkcja dodaje dane osobowe studenta i powiązanie studenta z nowo dodanym adresem
4. (**1p**) W poprzednim zadaniu kryje się pewien problem - w sytuacji kiedy funkcja wykonuje się równolegle dwa lub więcej razy, mimo sprawdzenia może zdarzyć się, że w bazie pojawiają się duplikaty (te same dane w dwu różnych rekordach).

W jakich przypadkach mogłoby zdarzyć się że mimo sprawdzenia, te same dane zostaną zarejestrowane wiele razy? Jak najlepiej zabezpieczyć się przed dodawaniem takich duplikatów?

Pokazać stosowną modyfikację kodu aplikacji lub struktury bazy danych, która uniemożliwia powstawanie duplikatów.

5. **(1p)** Inny problem kryjący się w kodzie który wykonuje zmiany na kilku różnych tabelach, to problem odczytu przez inny wątek takich danych, których proces dodawania jeszcze się nie zakończył (na przykład już dodano adres ale jeszcze nie dodano studenta). W celu wykluczenia możliwości wystąpienia takich sytuacji, baza danych ma możliwość używania tzw. transakcji.  
Pokazać w jaki sposób rozpoczynać, kończyć lub anulować (BEGIN, COMMIT, ROLLBACK) transakcje z poziomu kodu C#.
6. **(1p)** Zadanie 3 powtórzyć w technologii Dapper (używając tych samych zapytań do danych)
7. **(1p)** Zadanie 3 powtórzyć w technologii Linq2SQL
8. **(1p)** Zadanie 3 powtórzyć w technologii Dapper.SimpleCRUD (nie używając żadnych zapytań tylko polegając na automatycznym tworzeniu zapytań przez Dapper.SimpleCRUD).
9. **(1p)** Zadanie 3 powtórzyć w technologii Entity Framework 6 lub Entity Framework Core.
10. **(1p)** Pokazać jak zarządzać migracjami w Entity Framework. Formalnie - wygenerować migrację początkową. Pokazać jak baza danych tworzy się automatycznie na podstawie definicji migracji. Zmienić model obiektowy (na przykład dodać jakieś pole w klasie). Wygenerować migrację aktualizującą strukturę bazy danych.

Wiktor Zychla