

Seminarium dyplomowe

|  |  |  |  |
| --- | --- | --- | --- |
| **Kierunek studiów:** | *Informatyka* | **Rok studiów:** | *IV* |
| **Numer grupy:** | *L5* | | |
| **Rok akademicki:** | *2015/2016* | **Semestr:** | *VII* |

|  |  |
| --- | --- |
| Temat: |  |
| Sprawozdanie z realizacji zadań projektowych | |

|  |  |  |
| --- | --- | --- |
| **Lp.** | **Nr indeksu** | **Imię i nazwisko** |
| 1. |  | Damian Szymański |
| 2. |  | Radosław Szymański |

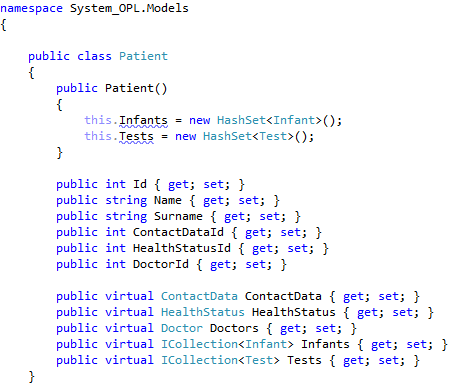
|  |  |  |
| --- | --- | --- |
| **Termin zajęć:** | | **Prowadzący:** |
| dzień: | *Poniedziałek* | Dr inż. Anna Bryniarska |
| godzina: | *14:50* |

1. **Umieszczanie rekordów w bazie danych**

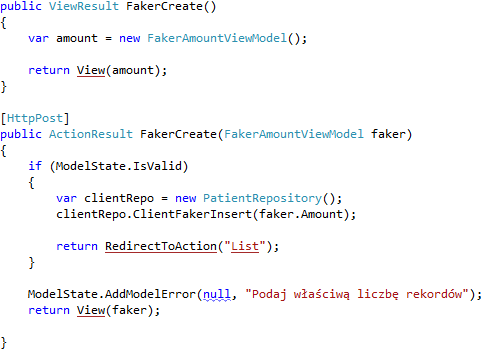
Celem zadania było wstrzykiwanie rekordów do wybranych tabel w bazie danych. Podczas projektowania funkcjonalności, założono, że będzie możliwe umieszczenie do 500 rekordów do 4 tabel jednocześnie. Przyjęto również, że dozwolone jest umieszczenie rekordów w tabelach powiązanych ze sobą.

Aplikacja została zrealizowany w środowisku .NET, zaimplikowało realizacje zadania w tej technologii. Podjęto decyzję o wykorzystaniu biblioteki Faker.NET, która umożliwia elastyczny i skalowalny proces realizacji zadania. Poniżej zamieszczono zrzuty ekranu, które przedstawiają kod, implementujący założoną fukncjonalność.

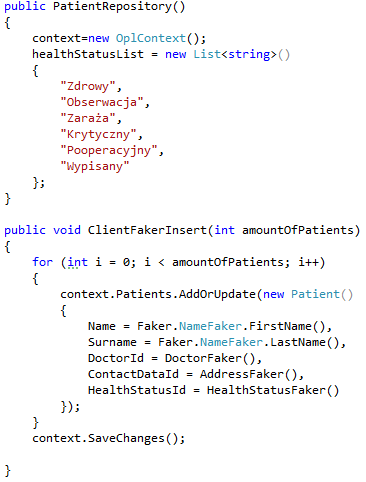
Poniżej przedstawiono reprezentację tabeli bazodanowej w programie. Tabela Patients została wybrana do realizacji zadania.

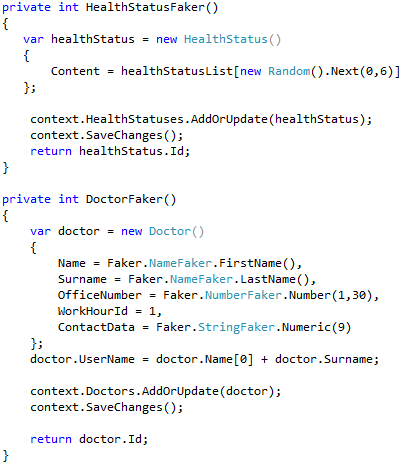


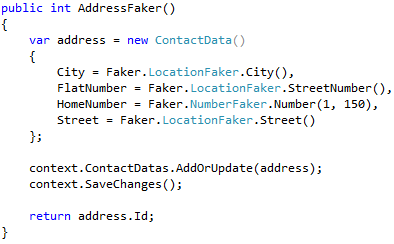
Poniżej przedstawiono miejsce wywołania metod, odpowiedzialnych za przeprowadzenie wstrzykiwania. Następuje to w kontrolerze, który zarządza akcjami CRUD pacjentów.



Poniżej przedstawiono ciało metod odpowiadających za realizje umieszczenia wybranej ilości rekordów w bazie danych.



****

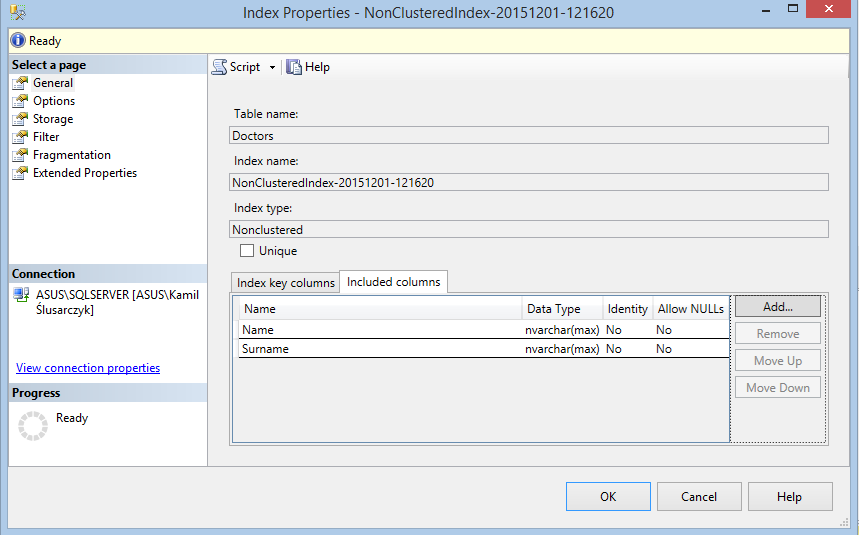
****

1. **Implementacja indeksów bazodanowych.**

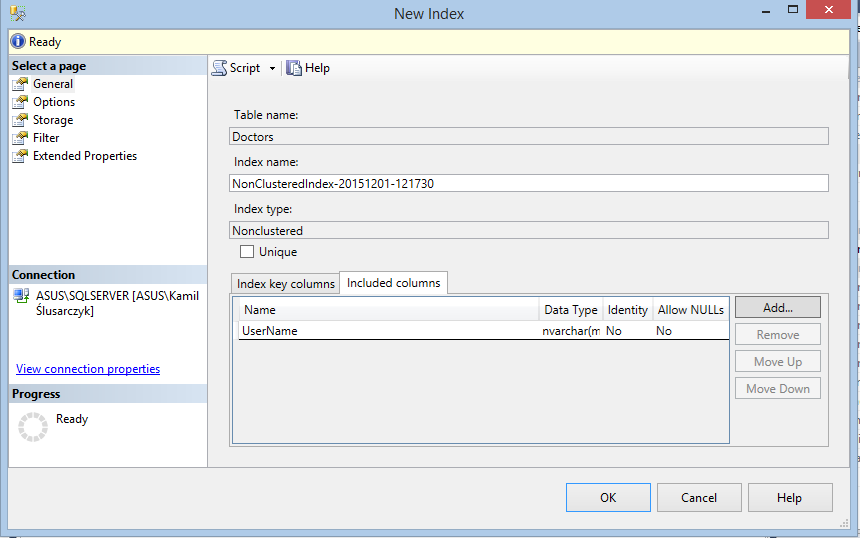
Celem zadania było utworzenie indeksów bazodanowych do wybranych tabel w bazie danych. W zadaniu należało ująć zarówno indeksy proste, jak i złożone.

Projekt został zrealizowany w środowisku .NET, co wymusiło konieczność realizacji zadania w programie MS SQL MANAGEMENT STUDIO.Poniżej przedstawiono zrzuty ekranu z realizacji zadania.

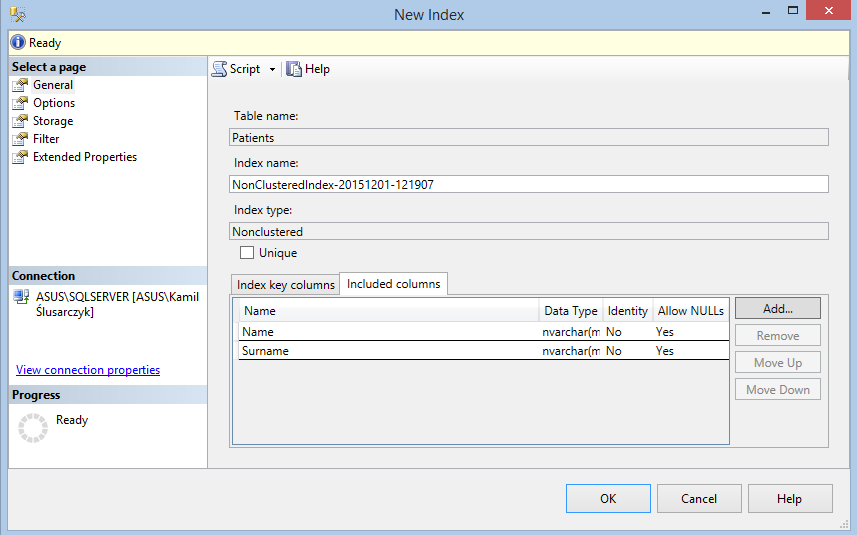
Poniżej przedstawiono właściwości indeksu złożonego, wykonanego w celu realizacji ćwiczenia. Jest to indeks przypisany do tabeli Doctors.



Następnie utworzono indeks do tabeli Doctors, który zawierał kolumnę UserName.



W celu realizacji ćwiczenia utworzono również indeks na tabeli Patients, który jest istotny ze względu na częste zastosowanie tej tabeli w logice aplikacji.

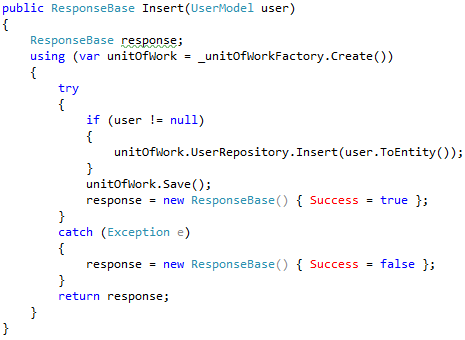


1. **Implementacja transakcji bazodanowych.**

Celem zadania zaimplementowanie transakcji bazodanowych. Założono realizację tego zadania w kodzie C# za pomocą wdrożenia wzorca projektowego Unit of Work.

Projekt został zrealizowany w środowisku .NET, co wymusiło konieczność realizacji zadania w tej samej technologii. Z tego powodu podjęto decyzję o wykorzystaniu języka C# i wyżej wspomnianej platformy.

Poniżej przedstawiono kod implementujący transakcję na przykładzie umieszczenia rekordu w bazie danych. W ramach dyrektywy *using* zostaje przeprowadzona operacje INSERT. Jeśli wszystkie dane są poprawne następuje zapisanie zmian w bazie i zwrócenie komunikatu systemowego o statusie operacji. W przeciwnym wypadku wyłapany zostaje wyjątek.



1. **Normalizacja bazy danych**

Zadanie normalizacji i optymalizacji bazy danych zostało przeprowadzone na przykładzie projektu, wykonanego w środowisku .NET. Fakt ten zaimplikował realizację powyższego zadania za pomocą MS SQL MANAGEMENT STUDIO.

Normalizacja została przedstawiona na dwóch przykładach. W obu przypadkach dzięki jej implementacji projekt zyskał na prostocie i elastyczności.

Proces normalizacji został zrealizowany na przykładzie tabeli Patients. Nadmiarowe kolumny zostały przeniesione do innych tabel. Kolumna HealthStatus oraz ContactData zostały zamienone na dwie tabele słownikowe. Dodano również klucze obce, co umożliwiło odwzorowanie zależności.

Poniżej przedstawiono tabelę Patients przed normalizacją:

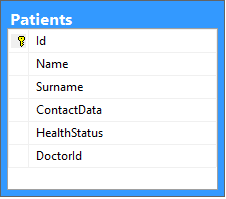
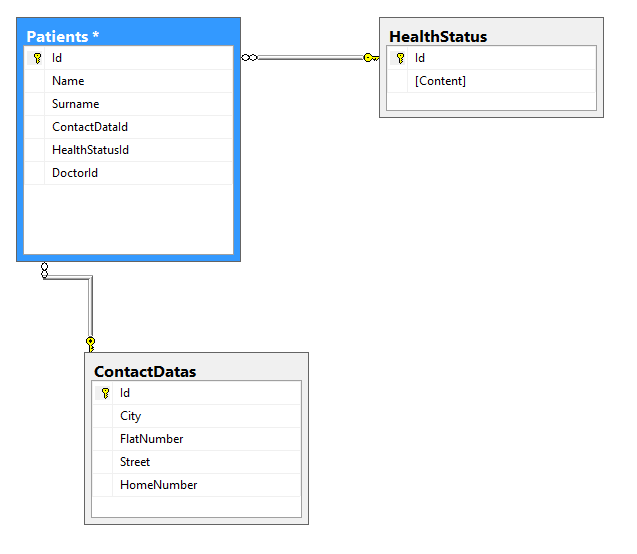


Tabela Patients po normalizacji:



Kolejnym przykładem jest normalizacji tabeli Doctors. Nadmiarowość kolumn ContactDate została rozwiązana, podobnie jak w poprzednim przykładzie, za pomocą utworzenia dodatkowej tabeli słownikowej.

Poniżej przedstawiono tabelę Doctors przed normalizacją:

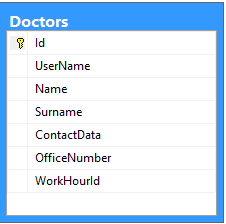


Tabela Doctors po normalizacji.

