

Dokumentacja techniczna

Projekt nr 2

Implementacja mechanizmów kontroli dostępu do baz danych

Wybrane środowisko uruchamiania programów: Windows

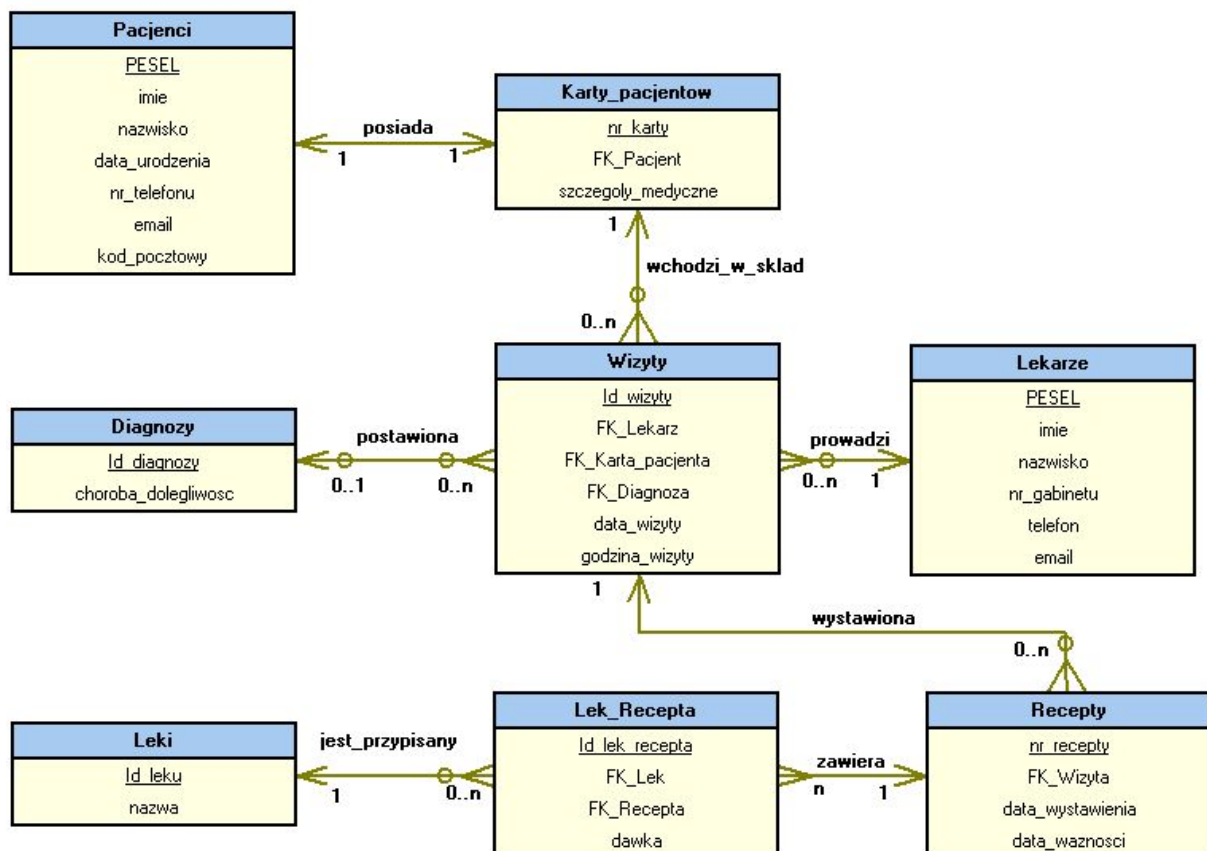
Użyty język programowania: C#

Użyty RDBMS: MS-SQL

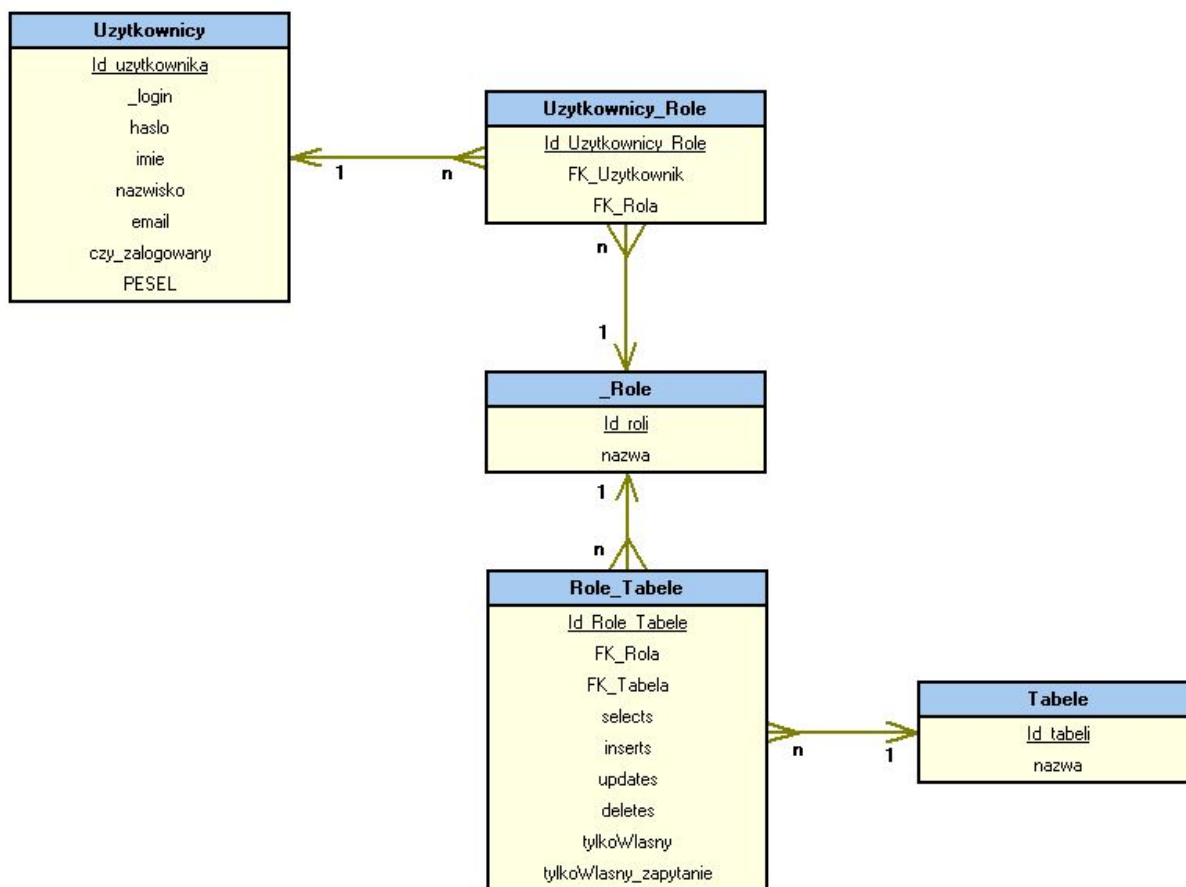
Model kontroli dostępu: RBAC

Schemat bazy danych:

a) Baza danych użytkownika



b) Baza danych administratora



Problem:

Dostęp do bazy danych w gabinecie lekarskim. Wyraźny podział ról na Pacjenta, Recepcjonistę, Lekarza oraz Administratora

	Nazwa tabeli	Pacjent	Recepcjonista	Lekarz
0	Pacjenci	select, update (tylko dla siebie)	select, insert, update, delete	brak praw
1	Karty_pacjentow	select (tylko dla siebie)	brak praw	select, insert, update, delete
2	Wizyty	select (tylko dla siebie)	select, insert, update, delete	select (tylko dla siebie)
3	Lekarze	select	select, insert, update, delete	select, update (tylko dla siebie)

4	Diagnozy	select (tylko dla siebie)	brak praw	select, insert, update, delete
5	Recepty	select (tylko dla siebie)	select	select, insert, update, delete
6	Leki	select (tylko dla siebie)	brak praw	select, insert, update, delete
7	Lek_Recepta	select (tylko dla siebie)	select	select, insert, update, delete
8	Uzytkownicy	brak praw	brak praw	brak praw
9	_Role	brak praw	brak praw	brak praw
10	Uzytkownicy_Role	brak praw	brak praw	brak praw
11	Tabele	brak praw	brak praw	brak praw
12	Role_Tabel	brak praw	brak praw	brak praw

Administrator ma pełne prawa do wszystkich tabel.

Przykładowe dane początkowe:

stworzenie konta (w tym przypadku konta dla przyszłego administratora)

```
INSERT INTO Uzytkownicy VALUES(0,'admin','d033e22ae348aeb5660fc2140aec35850c4da997',
'Admin','Adminowski','admin@student.pg.edu.pl','false','');
```

zdefiniowanie ról

```
INSERT INTO Role VALUES(0,'Administrator');
INSERT INTO Role VALUES(1,'Pacjent');
INSERT INTO Role VALUES(2,'Recepcjonista');
INSERT INTO Role VALUES(3,'Lekarz');
```

połączenie użytkowników i ról

```
INSERT INTO Uzytkownicy_Role VALUES(0,0,0);
INSERT INTO Uzytkownicy_Role VALUES(1,0,1);
INSERT INTO Uzytkownicy_Role VALUES(2,0,2);
INSERT INTO Uzytkownicy_Role VALUES(3,0,3);
```

zdefiniowanie tabel

```
INSERT INTO Tabele VALUES(0,'Pacjenci');
INSERT INTO Tabele VALUES(1,'Karty_pacjentow');
INSERT INTO Tabele VALUES(2,'Wizyty');
INSERT INTO Tabele VALUES(3,'Lekarze');
INSERT INTO Tabele VALUES(4,'Diagnozy');
INSERT INTO Tabele VALUES(5,'Recepty');
INSERT INTO Tabele VALUES(6,'Leki');
INSERT INTO Tabele VALUES(7,'Lek_Recepta');
```

połączenie ról i tabel (implementacja wcześniejszej tabeli podziału ról) dla recepcjonisty

```
----Recepcjonista
INSERT INTO Role_Tabele VALUES(26,2,0,'true','true','true','true','false','')
INSERT INTO Role_Tabele VALUES(27,2,1,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(28,2,2,'true','true','true','true','false','')
INSERT INTO Role_Tabele VALUES(29,2,3,'true','true','true','true','false','')
INSERT INTO Role_Tabele VALUES(30,2,4,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(31,2,5,'true','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(32,2,6,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(33,2,7,'true','false','false','false','false','')
--
INSERT INTO Role_Tabele VALUES(34,2,8,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(35,2,9,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(36,2,10,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(37,2,11,'false','false','false','false','false','')
INSERT INTO Role_Tabele VALUES(38,2,12,'false','false','false','false','false','')
```

Zakładane warunki pracy:

- gabinet lekarski - serwer, urządzenia desktopowe
- domy pacjentów - urządzenia desktopowe

Wykorzystane technologie:

- Cała aplikacja zbudowana została w Windows Form
- Wykorzystujemy klienta SqlConnection (System.Data.SqlClient)
- Baza danych MS-SQL.

Funkcjonalności:

- Logowanie się użytkowników
- Możliwość wyboru tabeli z dostępnych dla danego użytkownika o danej roli
- Podstawowe operacje na bazie danych (Select, Insert, Update, Delete)
- Kontrola dostępu użytkowników
- Statyczna separacja ról podmiotu

Interfejs użytkownika:

1. Miejsce na login użytkownika
2. Miejsce na hasło użytkownika
3. Przycisk logowania
4. Wybór roli użytkownika
5. Potwierdzenie roli
6. Lista tabel dostępnych dla użytkownika
7. Obszar roboczy
8. Kolumna wyboru wierszy do usunięcia
9. Przycisk odświeżania
10. Przycisk do tworzenia nowego, pomocniczego okna
11. Przycisk wylogowania
12. Konsola programu
13. Przycisk wprowadzenia danych do bazy danych
14. Przycisk zaktualizowania danych w bazie danych
15. Przycisk usunięcia wybranych (patrz punkt 8) wierszy z obszaru roboczego oraz w bazie danych.

The first screenshot shows a login form with the following elements:

- Login field (1)
- Password field (2)
- Login button (3)
- Profile dropdown (4)
- Select button (5)

The second screenshot shows the same login form with the following elements:

- Login field (1)
- Password field (2)
- Login button (3)
- Profile dropdown (4)
- Select button (5)

The third screenshot shows the main panel with the following elements:

- Pacjenci dropdown (6)
- Odśwież button (9)
- Nowe okno button (10)
- Wyloguj button (11)
- Table with columns: PESEL, imie, nazwisko, data_urodzenia, nr_telefonu, email, kod_pocztowy
- Table data:

	8	PESEL	imie	nazwisko	data_urodzenia	nr_telefonu	email	kod_pocztowy
<input type="checkbox"/>	6666666666	Bartosz	Korzyniewski	1996-07-20	+48876543210	barkor@test.pl	65-315	
<input checked="" type="checkbox"/>	7777777777	Pawel	Adamski	1996-07-20	+48876543210	pawada@test.pl	65-315	
<input type="checkbox"/>	8888888888	Sebastian	Mikolajczak	1996-07-20	+48876543210	sebmik@test.pl	65-315	
<input type="checkbox"/>	*							

- 7
- Console: Success, loaded selected table (12)
- Insert button (13)
- Update button (14)
- Delete button (15)

Testy:

Wczytywanie Tabeli		
Role_Tabele	Tabele	Pacjenci
7	4	3
3	6	3
7	4	6
4	7	7
7	3	4
3	4	2
4	3	3
6	7	3
4	4	4
3	3	3
4.8	4.5	3.8

Wykonano pomiary długości obsługi zapytań (w milisekundach) przez bazę danych podczas ładowania tabel Role_Tabele (9 kolumn x 52 wiersze), Tabele (3 kolumn x 13 wiersze) i Pacjenci (7 kolumn x 3 wiersze).

Obsługa zapytań dla tych zapytań trwa od 4 do 5 milisekund. Zapytania lepiej sobie radzą z tabelami o większej ilości kolumn niż większej ilości wierszy.

Bibliografia:

- **Beginners guide to accessing SQL Server through C#**
<https://www.codeproject.com/Articles/4416/Beginners-guide-to-accessing-SQL-Server-through-C>