



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



**Katedra
Informatyki i Automatyki**
Politechnika Rzeszowska

BAZY DANYCH

PROJEKT

pt.: „Salon fryzjerski”

Data wykonania: 21.01.2020

Grupa: L3
Mateusz Tłusty

Spis treści

Spis treści.....	2
1. Określenie projektu.....	4
1.1. Temat projektu.....	4
1.2. Określenie funkcji bazy danych.....	4
1.3. Narzędzia wykorzystane do realizacji projektu.....	5
1.4. Odnosnik do repozytorium z projektem.....	3
2. Diagramy i tabele.....	4
2.1. Diagram bazy danych.....	4
2.2. Opis tabel.....	5
3. Przedstawienie SQL.....	6
3.1. Tworzenie tabel baz danych i relacji.....	6
3.2. Uzupełnienie tabel przykładowymi danymi.....	7
3.3. Przykładowe funkcje DB, które może realizować SQL.....	9

1. Określenie projektu

1.1. Temat projektu

Tematem projektu jest baza danych zakładu fryzjerskiego. Zaprojektowana jest do przechowywania informacji niezbędnych do funkcjonowania salonu fryzjerskiego oraz umawiania spotkań. Przechowywane są również dane fryzjerów pracujących w zakładzie oraz dane klientów.

1.2. Określenie funkcji bazy danych

Baza danych zostanie wykorzystana jako baza systemu umożliwiającego umawianie spotkań w salonie fryzjerskim do konkretnego fryzjera i na konkretny rodzaj strzyżenia. System ten będzie przechowywał dane o klientach, łączenie z danymi kontaktowymi i danymi do logowania, dane fryzjerów oraz harmonogram spotkań.

1.3. Narzędzia wykorzystane do realizacji projektu

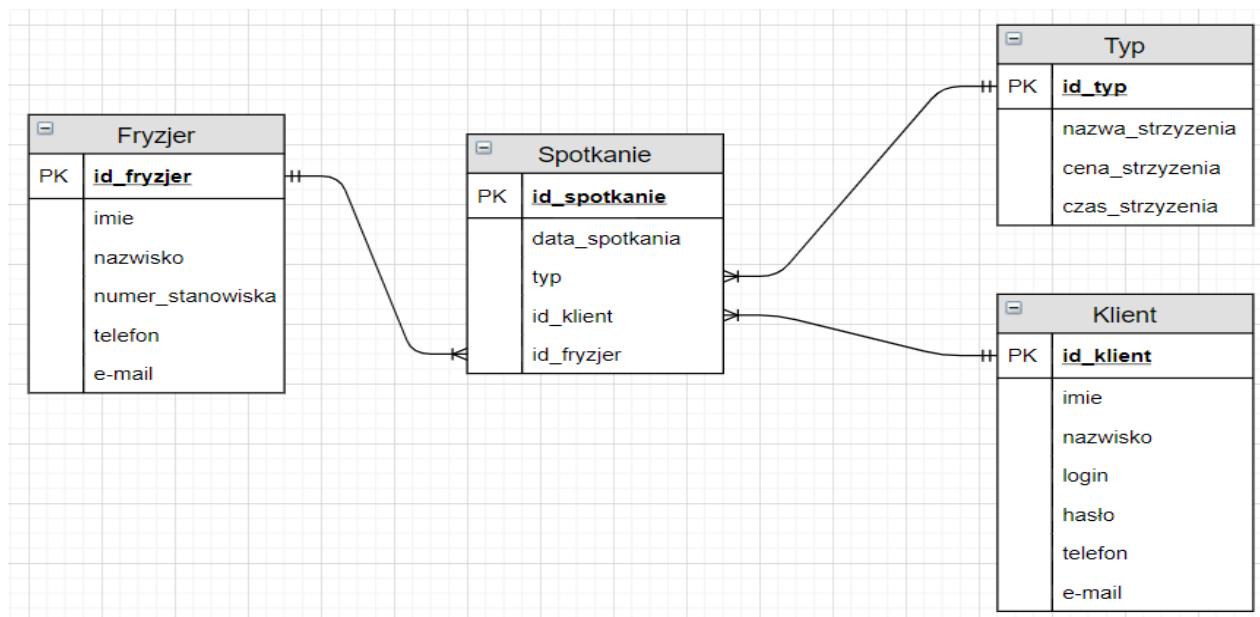
Do realizacji projektu zostały wykorzystane programy : SQL Server Management Studio 2019 w wersji 15.0.18206.0 oraz SQL Server Profiler v18.4

1.4. Odnośnik do repozytorium z projektem

<https://github.com/mateusztlusty/DB-Salon-Fryzjerski>

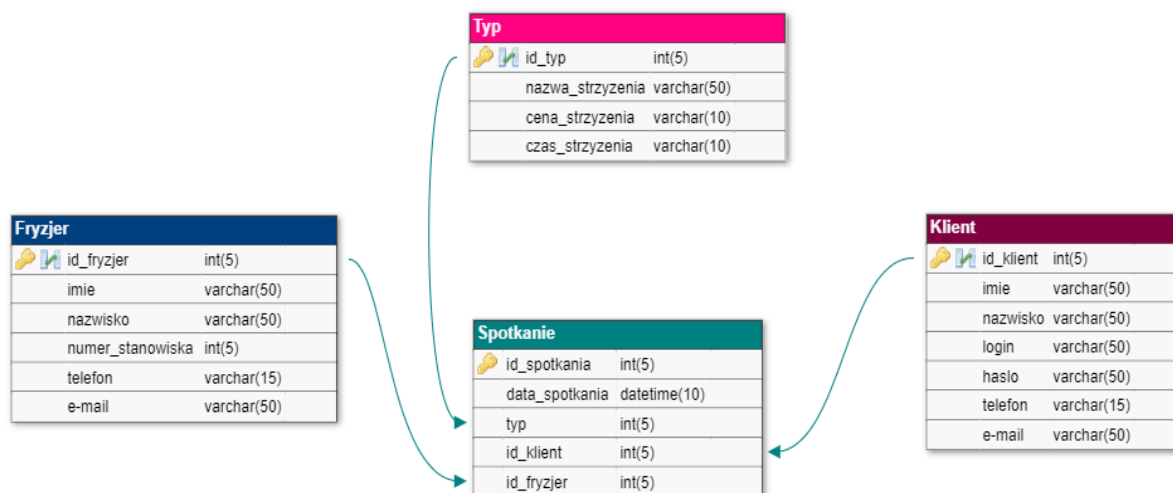
2. Diagramy i tabele

2.1 Diagram bazy danych



Rys. Prosty diagram ERD

dbdesigner.net



Rys. Diagram ERD



2.2 Opis tabel

- Tabela „Klient” przechowuje informacje o klientach. Tabela zawiera spis wszystkich klientów odwiedzających salon. Każdy klient posiada swój własny ID. Imię, nazwisko, login, hasło, telefon i e-mail jest typu VARCHAR. Numer telefonu jest typu VARCHAR a nie INTEGER aby zachować swobodę przy wprowadzaniu danych, przez co można wpisać numer telefonu np: +48 666 000 666 a nie ograniczać się do numeru kierunkowego w postaci 0048.
- Tabela „Fryzjer” przechowuje informacje o fryzjerach. Tabela zawiera spis wszystkich fryzjerów zatrudnionych w salonie, „id_fryzjer” jest tzw. kluczem głównym, co oznacza, że nie będzie innego fryzjera w tabeli o takim samym ID. Imię, nazwisko, telefon i e-mail jest typu VARCHAR, numer_stanowiska jest typu INTEGER.
- Tabela „Spotkanie” przechowuje informacje o spotkaniach. Każde spotkanie posiada własny ID, datę spotkania (datetime), typ strzyżenia podczas wizyty oraz ID klienta i fryzjera przypisanych do spotkania (INTEGER).
- Tabela „Typ” przechowuje informacje o typach strzyżenia. Każdy rodzaj strzyżenia posiada własny ID oraz jest opisany w polu nazwa_strzyżenia (np. trwała / loki / farbowanie itp.), posiada informacje o czasie i cenie usługi.

3. Przedstawienie SQL

3.1 Tworzenie tabel baz danych i relacji

```
CREATE TABLE Fryzjer (  
    id_fryzjer INT NOT NULL PRIMARY KEY,  
    imie VARCHAR(50) NOT NULL,  
    nazwisko VARCHAR(50) NOT NULL,  
    numer_stanowiska INT NOT NULL,  
    telefon VARCHAR(15) NOT NULL,  
    e_mail VARCHAR(50) NULL );  
  
CREATE TABLE Typ (  
    id_typ INT NOT NULL PRIMARY KEY,2  
    nazwa_strzyzenia VARCHAR(50) NOT NULL,  
    cena_strzyzenia VARCHAR(50) NOT NULL,  
    czas_strzyzenia VARCHAR(50) NOT NULL );  
  
CREATE TABLE Spotkanie (  
    id_spotkania INT NOT NULL PRIMARY KEY,  
    data_spotkania DATETIME NOT NULL,  
    typ INT NOT NULL,  
    id_klient INT NOT NULL,  
    id_fryzjer INT NOT NULL );  
  
CREATE TABLE Klient (  
    id_klient INT NOT NULL,  
    imie VARCHAR(50) NOT NULL,  
    nazwisko VARCHAR(50) NOT NULL,  
    llogin VARCHAR(50) NOT NULL,  
    haslo VARCHAR(50) NOT NULL,  
    telefon VARCHAR(15) NOT NULL,  
    e_mail VARCHAR(15) NULL );  
  
ALTER TABLE Fryzjer  
ADD FOREIGN KEY (id_fryzjer) REFERENCES Spotkanie(id_fryzjer);  
  
ALTER TABLE Klient  
ADD FOREIGN KEY (id_klient) REFERENCES Spotkanie(id_klient);  
  
ALTER TABLE Typ  
ADD FOREIGN KEY (id_typ) REFERENCES Spotkanie(typ);
```

3.2 Uzupełnienie tabel przykładowymi danymi

- Fryzjer

```
INSERT INTO Fryzjer VALUES (1, 'Tomasz', 'Lenart', 1, '+48 666 000 666', 'tlenart@gg.pl');
INSERT INTO Fryzjer VALUES (2, 'Karol', 'Palka', 3, '+48 666 888 666', 'kpalka@gg.pl');
INSERT INTO Fryzjer VALUES (3, 'Mateusz', 'Pazdzioch', 4, '+48 666 000 777', 'pazdzioch@gg.pl');
INSERT INTO Fryzjer VALUES (4, 'Karolina', 'Majcher', 2, '+48 666 777 666', 'kmajcher@gg.pl');
SELECT * from Fryzjer;
```

id_fryzjer	imie	nazwisko	numer_stanowiska	telefon	e_mail
1	Tomasz	Lenart	1	+48 666 000 666	tlenart@gg.pl
2	Karol	Palka	2	+48 666 888 666	kpalka@gg.pl
3	Mateusz	Pazdzioch	3	+48 666 000 777	pazdzioch@gg.pl
4	Karolina	Majcher	4	+48 666 777 666	kmajcher@gg.pl

- Typ

```
INSERT INTO Typ VALUES (1, 'Farbowanie', '120 ZL', '150 MIN');
INSERT INTO Typ VALUES (2, 'Cieniowanie', '120 ZL', '120 MIN');
INSERT INTO Typ VALUES (3, 'Loki', '80 ZL', '100 MIN');
INSERT INTO Typ VALUES (4, 'Strzyzenie krótkie', '20 ZL', '15 MIN');
INSERT INTO Typ VALUES (5, 'Strzyzenie półdługie', '35 ZL', '25 MIN');
INSERT INTO Typ VALUES (6, 'Strzyzenie długie', '50 ZL', '35 MIN');
INSERT INTO Typ VALUES (7, 'Grzywka', '50 ZL', '30 MIN');
INSERT INTO Typ VALUES (8, 'Effilage', '100 ZL', '80 MIN');
SELECT * from Typ;
```

id_typ	nazwa_strzyzenia	cena_strzyzenia	czas_strzyzenia
1	Farbowanie	120 ZL	150 MIN
2	Cieniowanie	120 ZL	120 MIN
3	Loki	80 ZL	100 MIN
4	Strzyzenie krótkie	20 ZL	15 MIN
5	Strzyzenie półdługie	35 ZL	25 MIN
6	Strzyzenie długie	50 ZL	35 MIN
7	Grzywka	50 ZL	30 MIN
8	Effilage	100 ZL	80 MIN

- Klient**

```
INSERT INTO Klient VALUES (1, 'Krystian', 'Grzesik', 'kgrzesik23', 'papapa23', '+48 543 345 543', null );
INSERT INTO Klient VALUES (2, 'Monika', 'Faja', 'fajaa43', 'palopa23', '+48 543 999 000', null );
INSERT INTO Klient VALUES (3, 'Władysława', 'barszcz', 'bbaarr', 'krupnik', '+48 576 543 765', 'barszczyk@on.pl' );
INSERT INTO Klient VALUES (4, 'Kasia', 'Bauman', 'kasiab23', 'lalalaland', '+48 545 345 678', null );
INSERT INTO Klient VALUES (5, 'Brylant', 'Gradz', 'gradzdd', 'tratata52', '+48 345 345 543', 'gradz@on.pl' );
INSERT INTO Klient VALUES (6, 'Krystian', 'Kuper', 'kkuper2', 'ddada', '+48 547 345 543', null );
INSERT INTO Klient VALUES (7, 'Mariola', 'Banas', 'banasiowa', 'partyssa', '+48 900 877 655', null );
SELECT * from Klient;
```

1	Krystian	Grzesik	kgrzesik23	papapa23	+48 543 345 543	NULL
2	Monika	Faja	fajaa43	palopa23	+48 543 999 000	NULL
3	Władysława	barszcz	bbaarr	krupnik	+48 576543 765	barszczyk@on.pl
4	Kasia	Bauman	kasiab23	lalalaland	+48 545 345 678	NULL
5	Brylant	Gradz	gradzdd	tratata52	048 345 345 543	gradz@on.pl
6	Krystian	Kuper	kkuper2	ddada	+48 547 345 543	NULL
7	Mariola	Banas	banasiowa	partyssa	+48 900 877 655	NULL

- Spotkanie**

```
INSERT INTO Spotkanie VALUES (1, '20191111 11:00:00 AM', 4, 1, 3);
INSERT INTO Spotkanie VALUES (2, '20191111 11:30:00 AM', 8, 2, 4);
INSERT INTO Spotkanie VALUES (3, '20191111 11:45:00 AM', 4, 3, 1);
INSERT INTO Spotkanie VALUES (4, '20191111 12:30:00 AM', 5, 6, 2);
INSERT INTO Spotkanie VALUES (5, '20191111 12:40:00 AM', 1, 4, 3);
SELECT * from Spotkanie;
```

id_spotkania	data_spotkania	typ	id_klient	id_fryzjer
1	2019-11-11 11:00:00.000	4	1	3
2	2019-11-11 11:30:00.000	8	2	4
3	2019-11-11 11:45:00.000	4	3	1
4	2019-11-11 00:30:00.000	5	6	2
5	2019-11-11 00:40:00.000	1	4	3

3.3 Przykładowe funkcje DB, które może realizować SQL

- sortowanie zatrudnionych fryzjerów po numerze stanowiska

```
SELECT imie, nazwisko FROM Fryzjer ORDER BY numer_stanowiska;
```

	imie	nazwisko
1	Tomasz	Lenart
2	Karol	Palka
3	Mateusz	Pazdzioch
4	Karolina	Majcher

- procedura dodająca nowego fryzjera

```
CREATE PROCEDURE [DodajFryzjera] @a INT, @b VARCHAR(50), @c VARCHAR(50), @d INT, @e VARCHAR(15), @f VARCHAR(50)
as
    insert into Fryzjer values (@a, @b, @c, @d, @e, @f);

exec DodajFryzjera '5', 'Karol', 'Swiderski', '5', '+48 888 999 000', 'karol@karol.pl';

SELECT * from Fryzjer;
```

id_fryzjer	imie	nazwisko	numer_stanowiska	telefon	e_mail
1	Tomasz	Lenart	1	+48 666 000 666	tlenart@gg.pl
2	Karol	Palka	2	+48 666 888 666	kpalka@gg.pl
3	Mateusz	Pazdzioch	3	+48 666 000 777	pazdzioch@gg.pl
4	Karolina	Majcher	4	+48 666 777 666	kmajcher@gg.pl
5	Karol	Swiderski	5	+48 888 999 000	karol@karol.pl



- procedura usuwająca fryzjera po podaniu numeru ID

```
} CREATE PROCEDURE [UsunFryzjera] @i INT  
as  
    delete from Fryzjer where id_fryzjer = @i;  
  
    exec UsunFryzjera 5;  
  
    SELECT * from Fryzjer;
```