Lab 4 Baza danych RockMusic

Mateusz Wiśniewski
Politechnika Śląska

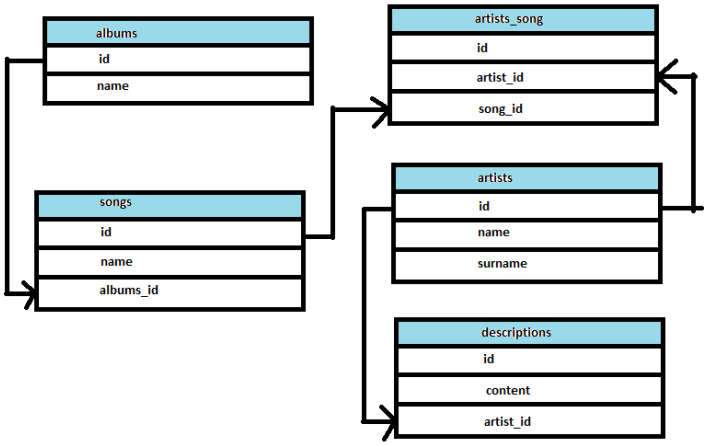
Zadanie polegało na zaprojektowaniu i stworzeniu aplikacji, która wyświetli wszystkie dane ze stworzonej bazy danych RockMusic w trybie połączeniowym oraz bezpołączeniowym.

Aplikacja została stworzony za pomocą języka C# z wykorzystaniem technologii WPF oraz środowiska Microsoft Visual Studio 2017.

Pierwszym krokiem było stworzenie bazy danych za pomocą języka SQL.

```sql
create database rockmusic;
use RockMusic

CREATE TABLE `albums` (
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `name` text COLLATE utf8_bin
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE `artists` (
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `name` text COLLATE utf8_bin,
  `surname` text COLLATE utf8_bin
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

INSERT INTO `artists` (`id`, `name`, `surname`) VALUES
(1, 'Ryan', 'Lewis'),
(2, 'Filip', 'Marcinek')
(3, 'Tomasz', 'Chada');

CREATE TABLE `artists_songs` (
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `artist_id` int(11) NOT NULL,
  `song_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE `descriptions` (
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `content` text COLLATE utf8_bin,
  `artist_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

CREATE TABLE `songs` (
  `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `name` text COLLATE utf8_bin,
  `albums_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

ALTER TABLE `artists_songs`
  ADD KEY `artist_id` (`artist_id`),
  ADD KEY `song_id` (`song_id`);

ALTER TABLE `descriptions`
  ADD KEY `artist_id` (`artist_id`);

ALTER TABLE `songs`
  ADD KEY `albums_id` (`albums_id`);

ALTER TABLE `artists_songs`
  ADD CONSTRAINT `artists_songs_ibfk_1` FOREIGN KEY (`artist_id`) REFERENCES `artists` (`id`),
  ADD CONSTRAINT `artists_songs_ibfk_2` FOREIGN KEY (`song_id`) REFERENCES `songs` (`id`);

ALTER TABLE `descriptions`
  ADD CONSTRAINT `descriptions_ibfk_1` FOREIGN KEY (`artist_id`) REFERENCES `artists` (`id`);

ALTER TABLE `songs`
  ADD CONSTRAINT `songs_ibfk_1` FOREIGN KEY (`albums_id`) REFERENCES `albums` (`id`);
```

Diagram bazy



Łączenie się do bazy w aplikacji

```
class DBConnection
{
    public static MySqlConnection Connection()
    {
            MySqlConnectionStringBuilder connectionStringBuilder = new MySqlConnectionStringBuilder
            {
                Port = uint.Parse("3306"),
                Server = "localhost",
                UserID = "root",
                Password = "",
                Database = "RockMusic"
            };
            MySqlConnection connection = new MySqlConnection(connectionStringBuilder.ToString());
            return connection;
    }
}
```

Reader

```
public void ShowAllArtistsWithReader()
{
    using (MySqlConnection conn = DBConnection.Connection())
    {
        try
        {
            conn.Open();
            using (MySqlCommand command = new MySqlCommand("SELECT * FROM artists", conn))
            {
                using (MySqlDataReader dataReader = command.ExecuteReader())
                {
                    while (dataReader.Read())
                    {
                        int id = (int)dataReader[0];
                        string name = (string)dataReader[1];
                        string surname = (string)dataReader[2];
                        Console.WriteLine($"Id: {id}, Imie: {name}, Nazwisko: {surname}");
                    }
                }
            }
        }
        catch
        {
            Console.WriteLine("Błąd reader");
        }
    }
```

Adapter

```
public void ShowAllArtistsWithAdapter()
{
    using (MySqlConnection conn = DBConnection.Connection())
    {
        try
        {
            conn.Open();
            string query = "SELECT * FROM artists";
            using (MySqlDataAdapter dataAdapter = new MySqlDataAdapter(query, conn))
            {
                dataAdapter.Fill(this.customers);
                foreach (DataTable dt in this.customers.Tables)
                {
                    for (int curCol = 0; curCol < dt.Columns.Count; curCol++)
                    {
                        Console.Write(dt.Columns[curCol].ColumnName.Trim() + "\t");
                    }
                    Console.WriteLine();
                    for (int curRow = 0; curRow < dt.Rows.Count; curRow++)
                    {
                        for (int curCol = 0; curCol < dt.Columns.Count; curCol++)
                        {
                            Console.Write(dt.Rows[curRow][curCol].ToString().Trim() + "\t");
                        }
                        Console.WriteLine();
                    }
                }
            }

        }
        catch
        {
            Console.WriteLine("Błąd adapter");
```