

# **Dokumentacja Procesu Przebudowy Systemu Bazodanowego Dla Polskiego Związku Wędkarskiego**

---

**Autor: Kacper Duda**

*Link do repozytorium: [https://github.com/KPZ231/Projekt\\_Bazodanowy](https://github.com/KPZ231/Projekt_Bazodanowy)*

## **Spis Treści:**

---

# Temat Pracy:

---

Dokumentacja przedstawia, przebudowę systemu bazodanowego Polskiego Związku Wędkarskiego, finansowana przez główny zarząd związku.

Proces przebudowy systemu bazodanowego PZW będzie obejmować:

- Kompletną przebudowę systemu, wraz ze zmianą z MS SQL, na MySQL
- Zmianę nazw tabel, wraz z rekordami
- Stworzenie nowych tabel
- Zawiązanie relacji pomiędzy tabelami
- Dodanie nowych rekordów do istniejących już tabel

## Opis PZW (*Polskiego Związku Wędkarskiego*)

---

PZW to instytucja zrzeszająca polskich wędkarzy, zajmują się dbaniem o wody wchodzące w skład ich zbiorników, zarybianie wód, oraz kontrola warunków na wodach oraz wędkarzy dokonujących połowy na tych wodach.



Logo Instytucji

## Potrzeby PZW W Zakresie Systemów Bazodanowych

---

Potrzeby:

- Rekonstrukcja aktualnego systemu bazodanowego
- Zwiększenie bezpieczeństwa dla użytkowników
- Zwiększenie wydajności oraz wygody dla użytkowników i pracowników informatycznych

## Opis Tabel Znajdujących Się W Procesie Przebudowy

---

**Tabela Członkowie**

| Kod Tworzenia Tabel:

```

CREATE TABLE czlonkowie (
    id_czlonka INT AUTO_INCREMENT PRIMARY KEY,
    imie VARCHAR(120) NOT NULL,
    nazwisko VARCHAR(120) NOT NULL,
    pesel VARCHAR(11) NOT NULL UNIQUE,
    email VARCHAR(255) UNIQUE,
    nr_tel VARCHAR(9) UNIQUE,
    wiek INT NOT NULL,
    id_kola INT NOT NULL,
    FOREIGN KEY (id_kola) REFERENCES kolo(id_kola)
);

CREATE TABLE kolo (
    id_kola INT AUTO_INCREMENT PRIMARY KEY,
    nazwa VARCHAR(120) NOT NULL UNIQUE,
    miejscowosc VARCHAR(255) NOT NULL UNIQUE,
    nr_kola INT NOT NULL UNIQUE,
    prct_skladki FLOAT NOT NULL
);

CREATE TABLE zbiornik (
    id_zbiornika INT AUTO_INCREMENT PRIMARY KEY,
    nazwa VARCHAR(120) NOT NULL UNIQUE,
    nr_kola INT NOT NULL,
    typ_zbiornika ENUM('staw', 'jezioro', 'zbiornik zaporowy',
    typ_zbiornika_pzw ENUM('zwykly', 'specialny', 'chroniony'),
    FOREIGN KEY (nr_kola) REFERENCES kolo(nr_kola)
);

CREATE TABLE okrog (
    id_okregu INT AUTO_INCREMENT PRIMARY KEY,
    nazwa VARCHAR(120) NOT NULL UNIQUE,
    nr_okregu INT NOT NULL UNIQUE,
    miejscowosc VARCHAR(255) NOT NULL UNIQUE
);

```

```
CREATE TABLE skladki (  
    id_skladki INT AUTO_INCREMENT PRIMARY KEY,  
    stala_skladki FLOAT NOT NULL DEFAULT 300  
);
```

Opis:

Oto opis tabel zawartych w powyższym kodzie SQL:

1. Tabela **czlonkowie** :

- Nazwa: czlonkowie
- Zakres przechowywanych danych:
  - id\_czlonka: identyfikator członka (liczba całkowita)
  - imie: imię członka (łańcuch znaków o maksymalnej długości 120)
  - nazwisko: nazwisko członka (łańcuch znaków o maksymalnej długości 120)
  - pesel: numer PESEL członka (łańcuch znaków o długości 11, unikalny)
  - email: adres e-mail członka (łańcuch znaków o maksymalnej długości 255, unikalny)
  - nr\_tel: numer telefonu członka (łańcuch znaków o długości 9, unikalny)
  - wiek: wiek członka (liczba całkowita)
  - id\_kola: identyfikator koła, do którego należy członek (liczba całkowita)
- Pola i dobrane do nich typy danych oraz atrybuty:
  - id\_czlonka: INT, AUTO\_INCREMENT, PRIMARY KEY
  - imie: VARCHAR(120), NOT NULL
  - nazwisko: VARCHAR(120), NOT NULL
  - pesel: VARCHAR(11), NOT NULL, UNIQUE
  - email: VARCHAR(255), UNIQUE
  - nr\_tel: VARCHAR(9), UNIQUE

- wiek: INT, NOT NULL
- id\_kola: INT, NOT NULL

## 2. Tabela **koło**:

- Nazwa: koło
- Zakres przechowywanych danych:
  - id\_kola: identyfikator koła (liczba całkowita)
  - nazwa: nazwa koła (łańcuch znaków o maksymalnej długości 120, unikalna)
  - miejscowosc: miejscowość, w której znajduje się koło (łańcuch znaków o maksymalnej długości 255, unikalna)
  - nr\_kola: numer koła (liczba całkowita, unikalna)
  - prct\_skladki: procentowa składka (liczba zmiennoprzecinkowa, niepusta)
- Pola i dobrane do nich typy danych oraz atrybuty:
  - id\_kola: INT, AUTO\_INCREMENT, PRIMARY KEY
  - nazwa: VARCHAR(120), NOT NULL, UNIQUE
  - miejscowosc: VARCHAR(255), NOT NULL, UNIQUE
  - nr\_kola: INT, NOT NULL, UNIQUE
  - prct\_skladki: FLOAT, NOT NULL

## 3. Tabela **zbiornik**:

- Nazwa: zbiornik
- Zakres przechowywanych danych:
  - id\_zbiornika: identyfikator zbiornika (liczba całkowita)
  - nazwa: nazwa zbiornika (łańcuch znaków o maksymalnej długości 120, unikalna)
  - nr\_kola: numer koła, do którego przynależy zbiornik (liczba całkowita, unikalna)

- typ\_zbiornika: typ zbiornika (ENUM: 'staw', 'jezioro', 'zbiornik zaporowy', 'rzeka', 'strumyk', niepuste)
- typ\_zbiornika\_pzw: typ zbiornika według Polskiego Związku Wędkarskiego (ENUM: 'zwykly', 'specjalny', 'chroniony')
- Pola i dobrane do nich typy danych oraz atrybuty:
  - id\_zbiornika: INT, AUTO\_INCREMENT, PRIMARY KEY
  - nazwa: VARCHAR(120), NOT NULL, UNIQUE
  - nr\_kola: INT, NOT NULL, UNIQUE
  - typ\_zbiornika: ENUM('staw', 'jezioro', 'zbiornik zaporowy', 'rzeka', 'strumyk'), NOT NULL
  - typ\_zbiornika\_pzw: ENUM('zwykly', 'specjalny', 'chroniony')

#### 4. Tabela **okrog** :

- Nazwa: okrog
- Zakres przechowywanych danych:
  - id\_okregu: identyfikator okręgu (liczba całkowita)
  - nazwa: nazwa okręgu (łańcuch znaków o maksymalnej długości 120, unikalna)
  - nr\_okregu: numer okręgu (liczba całkowita, unikalna)
  - miejscowosc: miejscowość, w której znajduje się okręg (łańcuch znaków o maksymalnej długości 255, unikalna)
- Pola i dobrane do nich typy danych oraz atrybuty:
  - id\_okregu: INT, AUTO\_INCREMENT, PRIMARY KEY
  - nazwa: VARCHAR(120), NOT NULL, UNIQUE
  - nr\_okregu: INT, NOT NULL, UNIQUE
  - miejscowosc: VARCHAR(255), NOT NULL, UNIQUE

#### 5. Tabela **skladki** :

- Nazwa: skladki

- Zakres przechowywanych danych:
  - id\_skladki: identyfikator składki (liczba całkowita)
  - stala\_skladki: stała składka (liczba zmiennoprzecinkowa, niepusta, domyślnie 300)
- Pola i dobrane do nich typy danych oraz atrybuty:
  - id\_skladki: INT, AUTO\_INCREMENT, PRIMARY KEY
  - stala\_skladki: FLOAT, NOT NULL, DEFAULT 300

## Opis Zbiorów Danych Przechowywanych W Bazie

---

### 1. Członkowie:

- Dane osobowe członków, takie jak imię, nazwisko, numer PESEL, adres e-mail, numer telefonu oraz wiek.
- Informacja o przynależności członka do konkretnego koła wędkarskiego.

### 2. Koła wędkarskie:

- Informacje dotyczące poszczególnych kół wędkarskich, takie jak nazwa, miejscowość, numer identyfikacyjny oraz procentowa składka.

### 3. Zbiorniki wodne:

- Charakterystyka zbiorników wodnych, obejmująca nazwę, numer identyfikacyjny, typ zbiornika (np. staw, jezioro, rzeka) oraz specjalny typ zbiornika według Polskiego Związku Wędkarskiego.

### 4. Okręgi:

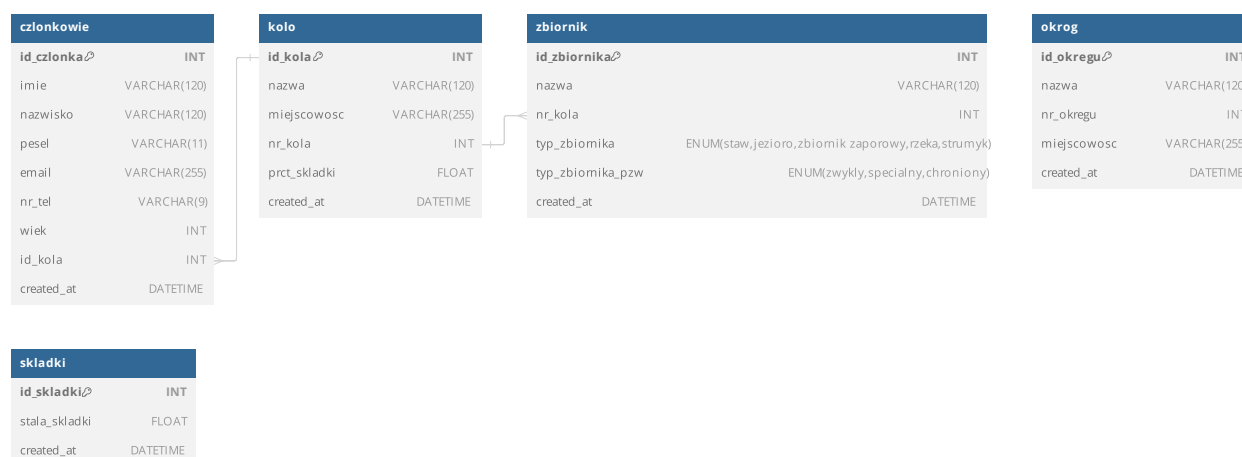
- Informacje o okręgach, w których działają kółka wędkarskie, takie jak nazwa, numer identyfikacyjny oraz miejscowość.

### 5. Składki:

- Dane dotyczące składek członkowskich, w tym stała składka, która może być modyfikowana w przypadku zmian regulaminowych.



# Diagram ERD



## Opis Utworzonych Relacji

- Relacja pomiędzy tabelami "czlonkowie" i "kolo":
  - Pole kluczowe w tabeli "czlonkowie": id\_kola
  - Pole kluczowe w tabeli "kolo": id\_kola
  - Typ relacji: wiele do jednego (wiele członków może należeć do jednego koła)
- Relacja pomiędzy tabelami "zbiornik" i "kolo":
  - Pole kluczowe w tabeli "zbiornik": nr\_kola
  - Pole kluczowe w tabeli "kolo": nr\_kola
  - Typ relacji: wiele do jednego (wiele zbiorników może należeć do jednego koła)
- Nie zostały utworzone żadne relacje pomiędzy tabelą "okrog" a innymi tabelami w bazie danych.
- Nie zostały utworzone żadne relacje pomiędzy tabelą "skladki" a innymi tabelami w bazie danych.

Ogólnie rzecz biorąc:

- W tabelach "czlonkowie" i "zbiornik" pole nr\_kola odnosi się do kolumny nr\_kola w tabeli "kolo", tworząc relacje wiele do jednego z tabelą "kolo".
- W tabeli "czlonkowie" pole id\_kola jest kluczem obcym odnoszącym się do tabeli "kolo", co oznacza, że każdy członek jest przypisany do jednego koła.

## Opis osób mających dostęp do bazy danych

---

### 1. Użytkownik standardowy:

- Funkcja: Zwykły użytkownik aplikacji.
- Uprawnienia:
  - Może wykonywać operacje odczytu i zapisu na swoim własnym profilu.
  - Ma dostęp do podstawowych funkcji aplikacji, takich jak przeglądanie zawartości strony głównej.

### 2. Moderator:

- Funkcja: Osoba odpowiedzialna za moderację treści.
- Uprawnienia:
  - Może wykonywać operacje odczytu i zapisu na wszystkich profilach użytkowników.
  - Ma uprawnienia do usuwania nieodpowiednich treści.
  - Zarządza komentarzami i innymi interakcjami użytkowników z aplikacją.

### 3. Administrator:

- Funkcja: Osoba odpowiedzialna za administrację systemem.
- Uprawnienia:
  - Posiada pełny dostęp do bazy danych.
  - Ma możliwość zarządzania wszystkimi profilami użytkowników.

- Może usuwać konta użytkowników w razie potrzeby.
- Zarządza uprawnieniami dostępu i wykonuje wszelkie operacje administracyjne wymagane przez system.